AppVeil User Manual

Contents

1. Introduction

AppVeil is an Android application which lets users improve privacy on Android systems by controlling what sensitive data and features an app can access. It patches the target apps to alter some system API calls. Users can choose options for the patches, controlling which features should be altered.

1.1 Revisions

2015-04-19: Initial creation of this manual.

2. Installation

This software is provided as an APK file which may be installed on Android devices via sideloading from a PC. We recommend using the ADB utility from Android's SDK Tools.

2.1 Prerequisites

The Android device you wish to install this application to must be running Android version 4.0 or higher and must support developer options. We recommend having at least 100MB free space on your Android device. Initial installation of this application requires using a PC that meets the system requirements for Android SDK (linked below).

Android SDK Requirements: https://developer.android.com/sdk/index.html#Requirements

2.2 Device Preparation

Source: https://developer.android.com/tools/help/adb.html

> In order to use adb with a device connected over USB, you must enable **USB debugging** in the device system settings, under **Developer options**.
>
> On Android 4.2 and higher, the Developer options screen is hidden by default. To make it visible, go to **Settings > About phone** and tap **Build number** seven times. Return to the previous screen to find **Developer options** at the bottom.
>
> On some devices, the Developer options screen may be located or named differently.

2.3 Installing Android SDK Tools

Instructions: https://developer.android.com/sdk/installing/index.html?pkg=tools
Additional instructions: https://developer.android.com/tools/device.html#setting-up
See the section for setting up your system to detect your device.

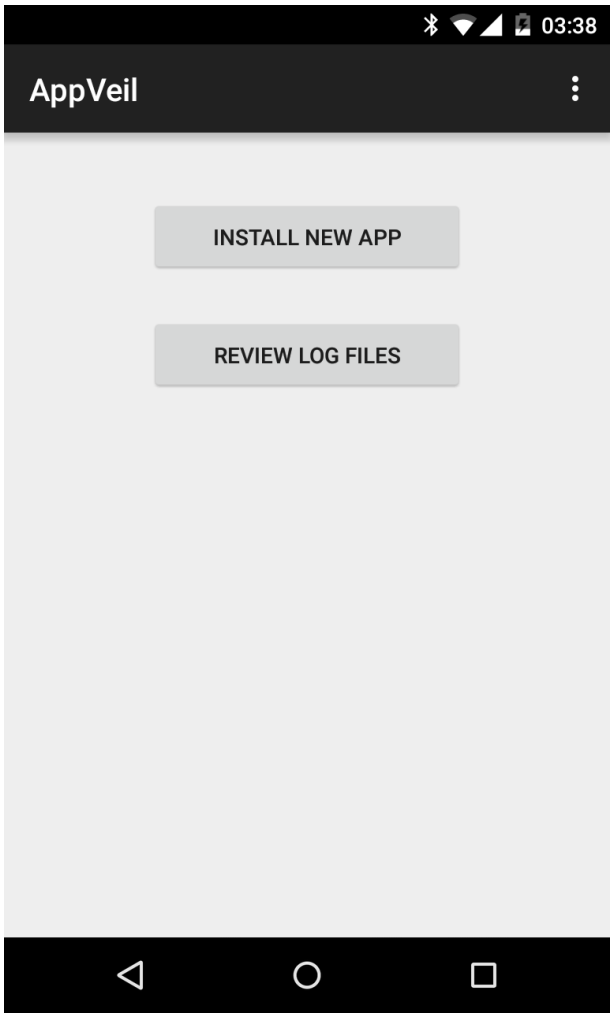
2.4 Installing from ADB

1. Open a terminal or command prompt on your PC
2. Connect your Android device to your PC with a USB cable
3. Run the command:
   `adb install /path/to/appveil.apk`
4. If your device's screen shows a prompt for debug access, grant it approval.

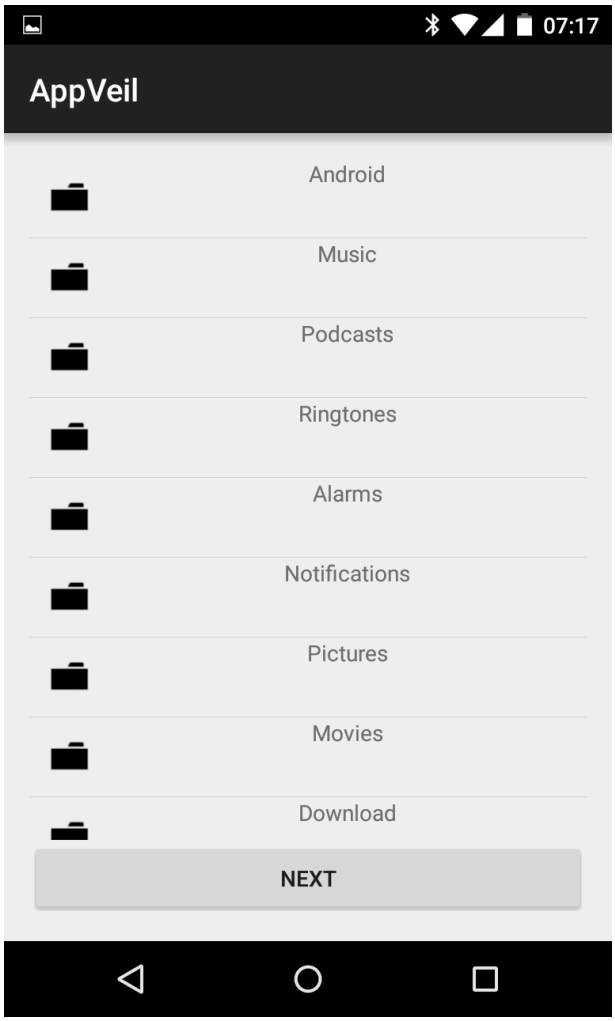
3. User Interface

3.1 Main Menu

This is what you first see when you launch AppVeil. This menu currently gives you the option to either install a new app or review logs that have been recorded from previously patched apps.



3.2 File Browser



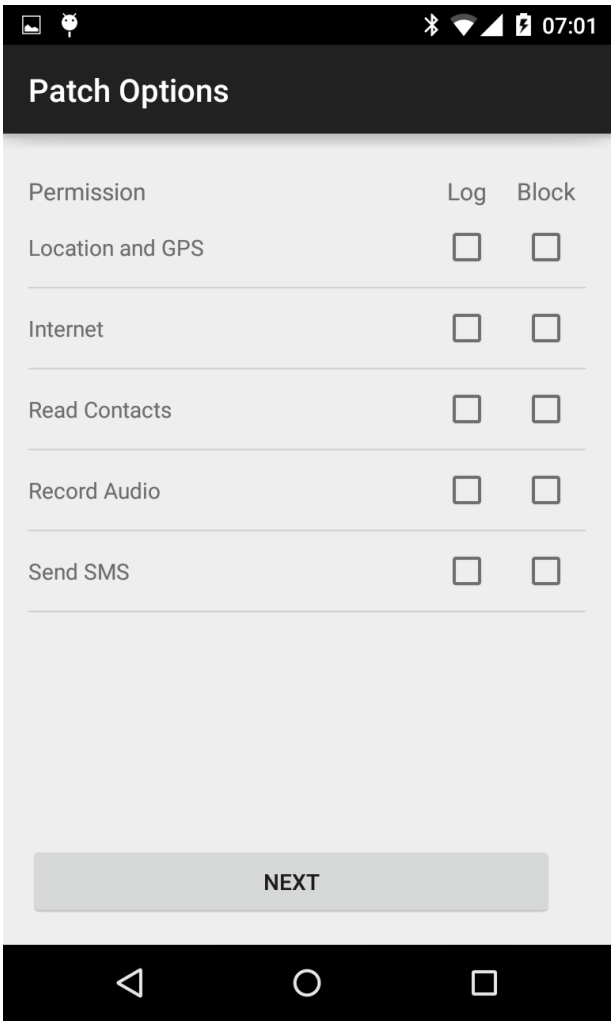
This is the first screen you see when you want to install a new app. In this menu, you can browse the files on your device. Tap a subdirectory to descend into it and view its contents. After descending, press the back button to return to the previous directory. When you're at the top directory, press back again to return to the main menu. Swipe up or down to scroll through long lists of files. To proceed, select an APK file of the app you wish to patch or install and press the **Next** button.

The root directory this file browser starts at is your device's "external storage" directory. If you have an SD card, it's likely that. Otherwise, don't worry, that "external" is a misnomer. This is the directory on any Android device that is shared between almost all apps, which is normally used to store all photos, music, and many other files. Most devices let you access this directory on a PC when you connect a USB cable. In the SDK Tools, you can usually access this directory as **/sdcard** (which may be a symbolic link).

Some of the likely locations you might find your APK files at:

- the starting directory: If you copied over an APK file from your computer
- **Download**: If you downloaded an APK file from a web site, using a web browser on your Android device.
- **backups/apps**: If you use ES File Explorer to backup your device's apps, the APK files of those backups are stored here.
- nowhere: Just having an app installed doesn't mean you can find it here. See section **5. Obtaining Target Apps** for more information.

3.3 Patch Options



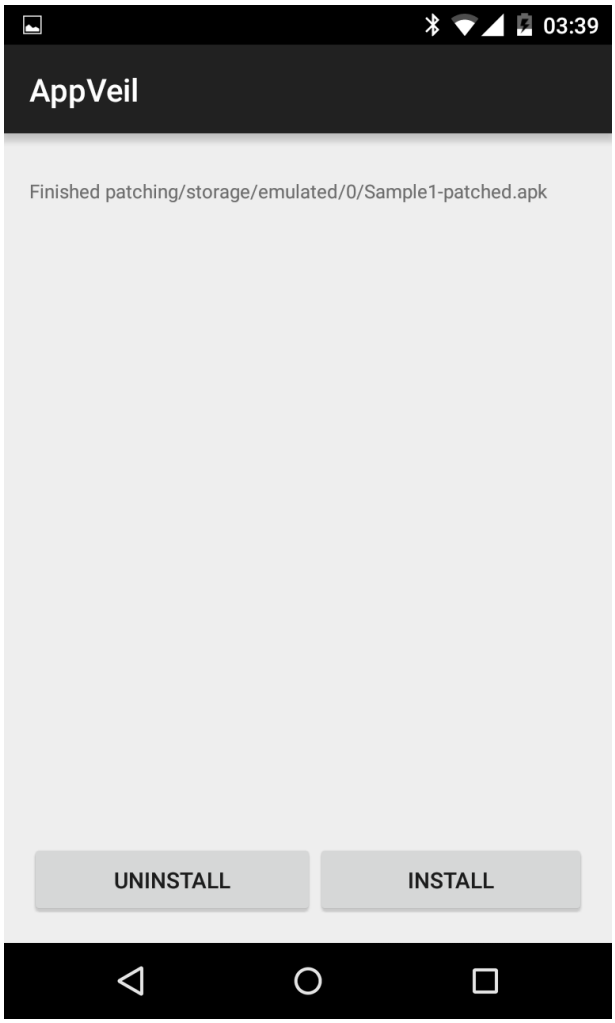
This screen shows you the available patch options for the app you selected in the file browser. The available options are based on what permissions that app uses. You may install the original app without any changes by leaving all of the options unchecked and pressing the button at the bottom. To patch the app, select the options you want and press the button.

Every possible set of options is valid:

- If you choose to **Block** any items, the patched app will be unable to access those features.
- If you choose to **Log** any items, the patched app will keep records of when it accesses those features.
- If you choose to both **Block** and **Log** the same item, the patched app cannot access the feature, but every attempt to access it will be recorded.
- If you choose neither, all API calls related to that item will be left unchanged.

3.4 Patch Status

As soon as you finish selecting the patch options, this screen will show you the progress of the patching process. When the patching completes, you may install the patched app. If you previously installed an unpatched version of this app, you must uninstall it first.



4. Patches

4.1 Terminology

- **APK File:** Basically, a file that contains an app. More accurately, this is an Android package file which contains the code, resources, and metadata needed by Android to install an app.
- **Source code:** The Java code that developers write in a plain text human-readable format, which may or may not be kept private and never released to users
- **Bytecode:** The format that an Android app's code is in when it's distributed for installation on users' devices. This is a binary format that is generated from source code by Android's build tools and is meant for internal use by the Android system.
- **Instruction:** A single operation in an app's bytecode
- **DEX File**: A file contained within an APK file which includes all of the code for an app and its bundled libraries
- **Target app:** An app that you wish to provide to AppVeil as input
- **Patched app:** The resulting app produced by AppVeil after the patching process. This result depends on both the original target app and the user-selected patch options.
- **Android System API**: All of the possible ways that the code within an app can invoke code outside the app to access features provided by the Android system
- **API method**: A single piece of the Android System API which an app can invoke directly from Java code
- **API call**: An instruction which invokes an API method
- **Permission**: A set of API methods that correspond to a feature relevant to user privacy concerns. The "permissions" used in AppVeil don't perfectly match the "permissions" managed by Android.

4.2 Technical Overview

AppVeil works by accepting any user-provided APK files and patching them to fulfil the changes requested by the user. The patcher searches an app's bytecode for every API call that corresponds to a permission selected by the user. These API calls are all altered to invoke newly-inserted code instead of invoking an API method. The inserted code imitates the original API method and, depending on the patch options, may save a record of the call attempt and/or may call the original intended API method.

4.3 Limitations

- You cannot have an original/unpatched app and a patched version of it installed on a single Android device at the same time. You must uninstall one before installing the other.
- All settings and private data belonging to an unpatched app cannot be transferred to a patched version and vice-versa.
- AppVeil's patches currently operate on only the bytecode from Java source. Some apps may have system-dependent native code which directly accesses sensitive

system features, but AppVeil cannot modify that code. Most typical apps don't use this, but because of this limitation, we cannot promise reliable privacy protection and detection.

5. Obtaining Target Apps

In order to use AppVeil, you must provide APK files for it to patch and install. There are various ways you can obtain the APK files of apps you desire. **Keep in mind the copyright license of all APK files you download.** We don't recommend redistributing any patched apps you produce with AppVeil, but if you really want to, make sure to obey the original app's license restrictions. We're not lawyers and cannot make any legal promises, but we believe it is safe to use AppVeil to patch and install any app you have access to for private, noncommercial use. Consult your country's laws if you're concerned about your particular use-case. When in doubt, the safest target apps to work with include ones licensed under any GPL, MIT, BSD, or Apache licenses.

5.1 Direct download and Sideload

Some app developers freely release current APK file downloads of their apps that you can download from the web. Try searching online for the app you want to find any download links that might exist. We do not recommend using any unofficial download of an app that was not uploaded by the developer of that app. While AppVeil can assist with some privacy concerns, we do not promise any reliable protection against malware.

If you used a PC to download an APK file, you can copy it to your device by connecting it with a USB cable. You can either use your operating system's file manager or the **adb push** command to send the file over USB.

If you downloaded an APK file directly from a web browser on your Android device, it should be in your **Download** folder and ready for AppVeil to use.

5.2 F-Droid

Disclaimer: AppVeil is not affiliated with F-Droid. This is just one repository we know of which may be useful to our users.

F-Droid is a fairly large repository of exclusively free open-source apps. Its web interface provides easily accessible download links for APK files built and signed by the F-Droid Project for all apps.
Link: https://f-droid.org/repository/browse/

5.3 Backups

Disclaimer: AppVeil is not affiliated with the developers of ES File Explorer. This is just one tool we know of which may be useful to our users.

If you already installed an app on your device, it is possible to retrieve the APK file of that app for backup purposes. For at least some devices (we tested on Nexus 4, Nexus 5, and Nexus 7), this does not require root permissions. There are probably many Android apps which let you obtain these backups. ES File Explorer (available from Google Play) is one that we know of and have successfully used. One procedure which you can use for this (as of the time this manual was written) is:

1. Install ES File Explorer from Google Play onto your device.
2. Use any method (such as the Google Play Store) to install the target apps you wish to use onto the same device. Note that doing this requires accepting the app's full list of permission requests for now.
3. Open ES File Explorer.
4. Open the navigation drawer, expand the "Library" section, and select the "APP" item.
5. In the app list, find at least one installed app you wish to use.
6. Long-press on that app's icon to select it. If you want to backup more apps right now, tap on each of their icons as well to select them all.
7. Press the Backup button.
8. Open AppVeil.
9. In the file browser, navigate into **backup** and then **apps**. All the apps you backed up should be listed here.
10. Patch and use the apps however you like.