**Android Malware Detection Based on Factorization Machine**

**Name1:** Amit Waizman 314635756

**Name2**: Rivka Buskila 206701187

**The feature set that the paper uses:**

Our article includes 7 types of features from the source code and the manifest file, of which four types are extracted from an application's manifest file and three additional types of features from the unpacked application's source code.

**features from manifest file**

**App components :**

The components declared in the manifest file defines the different interfaces that exist between app and end-user as well as the app and the

larger Android OS as a whole. The names of these

components are collected to help identify variants of

well-known malware.

**: Hardware features**

If an application wants to request access to the hardware components of the device, such as its camera, GPS or sensors, then those features must be declared in the manifest file. Requesting certain hardware components or pairs of components may have security implications.

**: Permissions**

Android uses a permission mechanism to protect the privacy of users. An app must request permission to access sensitive data (e.g. SMS), system features (e.g. camera) and restricted APIs. Malware usually tends to request a specific set of permissions. In this respect, this is similar to how we handle hardware features.

**: Intent filter**

Intent filters declared within the declaration of components in the manifest file are important tools for inter-component and inter-application communication. Intent filters define a special entry point for a component as well as the application. Intent filters can be used for eavesdropping specific intents. Malware is sensitive to a special set of system events. Thus, intent filters can serve as vital features.

**features from manifest file:**

**: Restricted APIs**

In the Android system, some special APIs related to sensitive data access are protected by permissions. If an app calls these APIs without requesting corresponding permissions, it may be a sign of root exploits.

**: Suspicious APIs**

We should be aware of a special set of APIs that can lead to malicious behavior without requesting permissions. For example, cryptography functions in the Java library and some math functions need no permission to be used. However, these functions can be used by malware for code obfuscation. Thus, attention should be paid to the unusual usage of these functions. We mark these types of functions as suspicious APIs

**Used permissions:**

We first extract all API calls from the app source code, and use this to build a set of permissions that are actually used in the app by looking up a predefined dictionary that links an API to its required permission(s).

. The type of classifier that you analyze static