DroidEagle: Seamless Detection of Visually Similar Android Apps

Mingshen Sun, Mengmeng Li and John C.S. Lui

Department of Computer Science and Engirneering
The Chinese University of Hong Kong

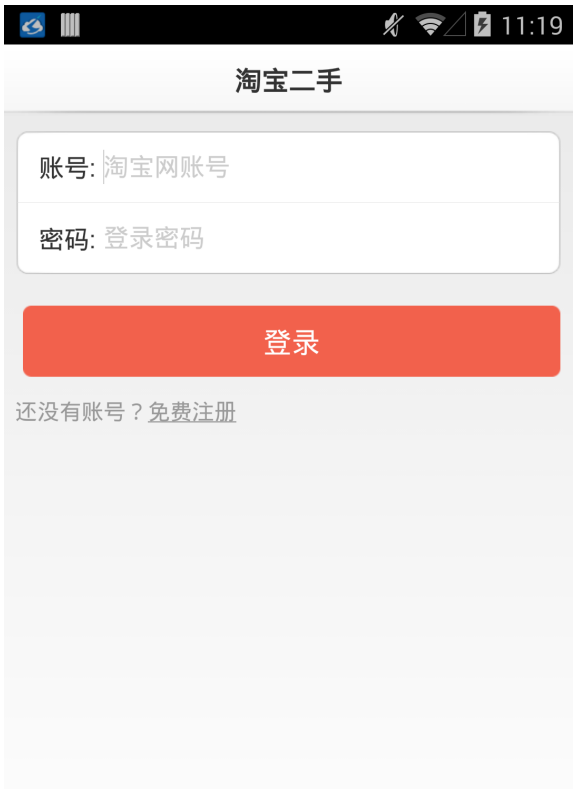
June 25, 2015

Introduction – Android Malware is Coming

Introduction – Android Malware

- Android malware samples accounted for **98%** of all mobile threats
- **99%** came from many third-party markets
- trojan, fake and phishing apps: can you tell which one is real

Taobao, one of the largest online shopping platform in China?



Introduction – Repackaging Technique

- **86%** of Android malware are using the *repackaging technique*
  - disassemble a legitimate app using some well-known tools
  - hackers can add or modify logics of the original apps, and then assemble it back
  - distribute them in third-party markets
- malicious functions using repackaging technique
  - crack paid apps to bypass payment functions
  - replace developers' advertisement IDs
  - acquire sensitive information: account password and credit card number

Introduction – Related Work

Related Work

- instruction sequences
  - fuzzy hashing
  - sensitive to instruction sequence obfuscation
- semantic information
  - call reference graph
  - hacking tricks to bypass existing disassembling tools

Introduction – DroidEagle

Observations:

- repackaged apps *should* have **similar appearance** as original one
- phishing malware *relies on* **similar appearances** as banks or shopping apps to deceive users
- by comparing *visual similarity*, one can quickly determine potential repackaged malware or phishing malware

DroidEagle is based on visual characteristics to detect similar Android apps.

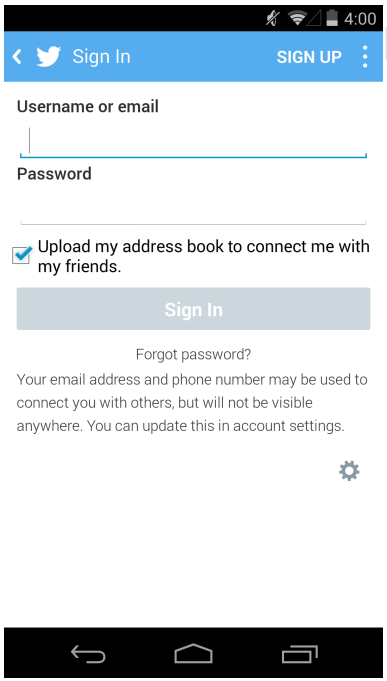
- detect visually similar apps in app repository and Android device respectively
- RepoEagle and HostEagle implementation

Background – Android App User Interface

Android app user interface

- `View`: objects on the screen which can interact with users and display objects (e.g., `ImageView`, `EditText`)
- `ViewGroup`: define the layout arrangement of its elements (e.g., `ScrollView`, `LinearLayout`)

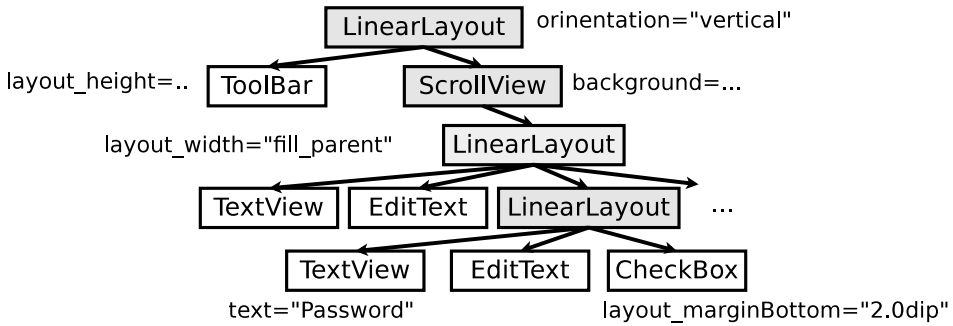
- layout files in `/res/layout*` directory




```
<?xml version="1.0" ... ?>
<LinearLayout orinentation=...>
  <ToolBar ... />
  <!-- The tool bar in the top. -->
  <ScrollView background=...>
    <LinearLayout ...>
      <TextView text=... />
      <!-- Username or email. -->
      <EditText inputType=... />
      <LinearLayout ...>
        <TextView ... />
        <!-- Password field -->
        <EditText ... />
        <CheckBox ... />
        <!-- "Upload my ..." -->
      </LinearLayout>
      ...
    </RelativeLayout>
</LinearLayout>
```

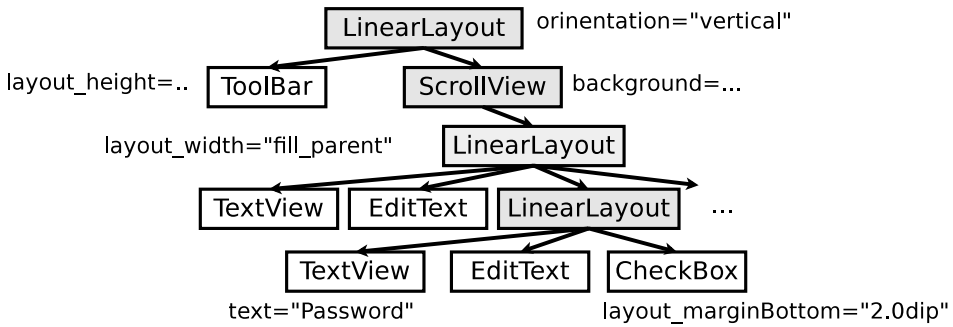
Detection Methodologies – Layout Tree

- Overview
  - accuracy, efficiency, scalability and flexibility
  - visual resources in an app: layout files and drawable images
- A layout tree is a tree data structure over a layout file where:
  - A node in the layout tree represents an element in the layout file.
  - The parent/child relationship of nodes in a layout tree is the same as that in the layout file.
  - attributes for each node



Detection Methodologies – Layout Tree

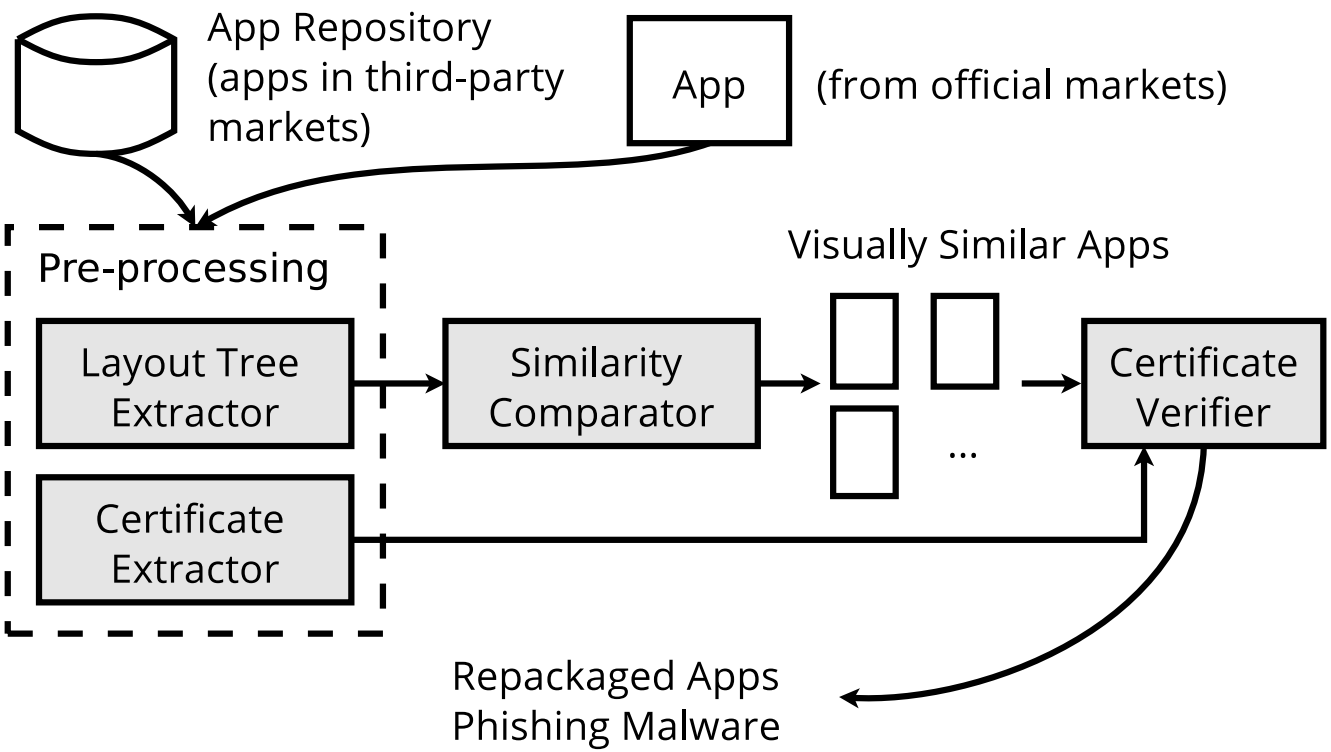
1. layout tree defines the visual structure of an app's user interface
2. repackaged malware and phishing malware rely on same layout tree to deceive users
3. detailed attributes in layout tree accurately describe visual appearance
4. layout tree is easily obtained in android package file
5. modifications on layout files can mess up appearance



RepoEagle: Repository Analysis

Repository Analysis

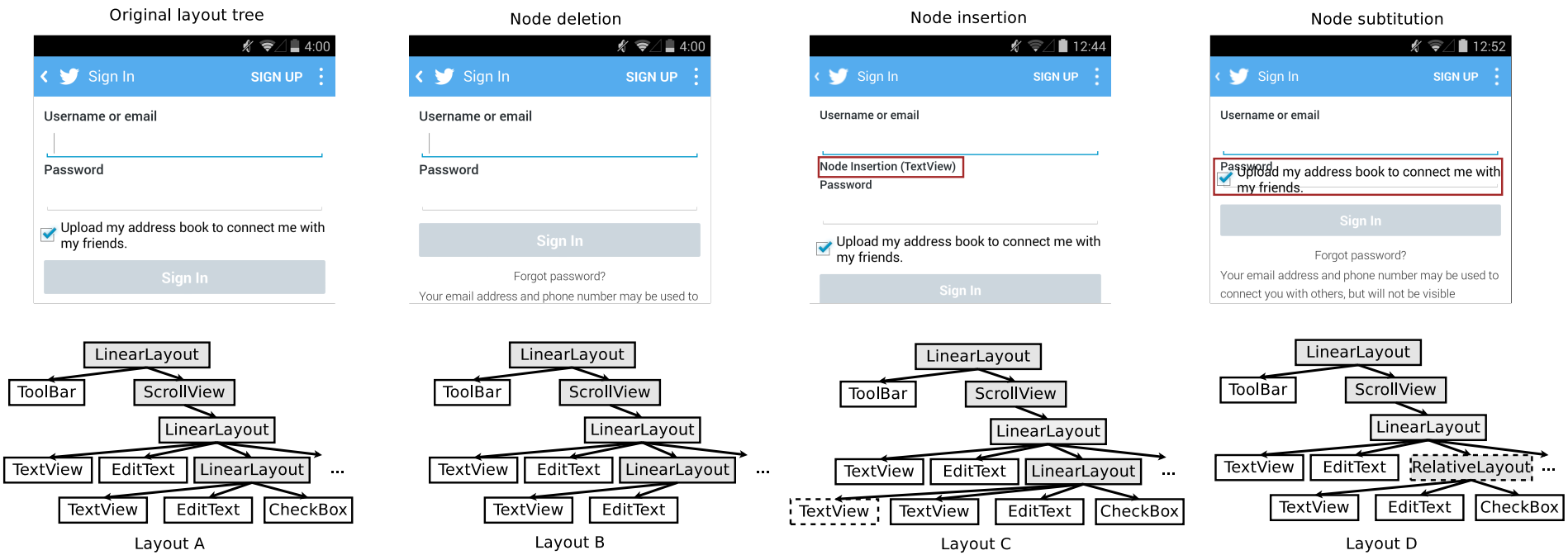
- RepoEagle can analyze all apps in an app repository to discover all visually similar apps
  - layout tree extractor, certificate extractor, similarity comparator, certificate verifier



RepoEagle: Repository Analysis

Similarity Comparison: layout edit distance (LED)

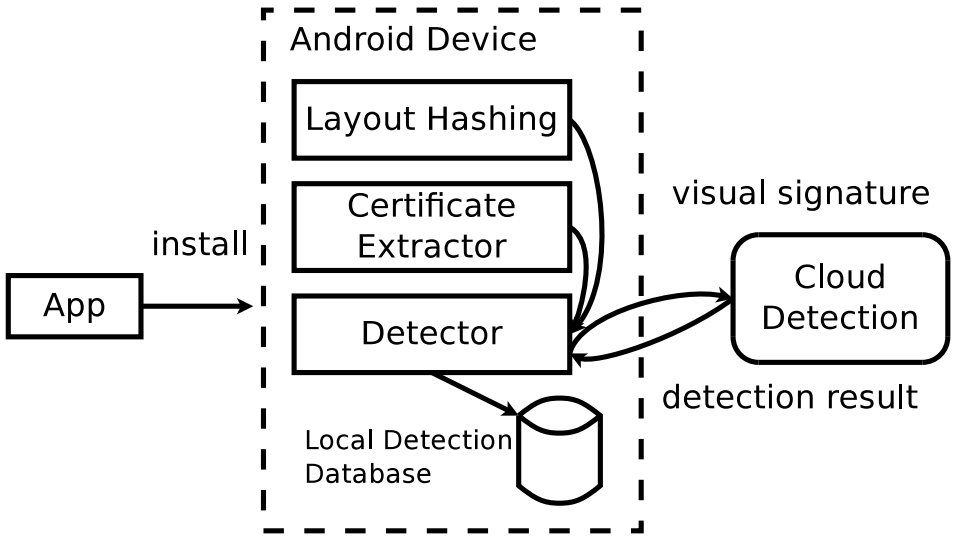
- measure the similarity between two layout trees
- LED is the *minimum number of operations* required to transform from one layout tree to another tree



HostEagle: Host-based Detection

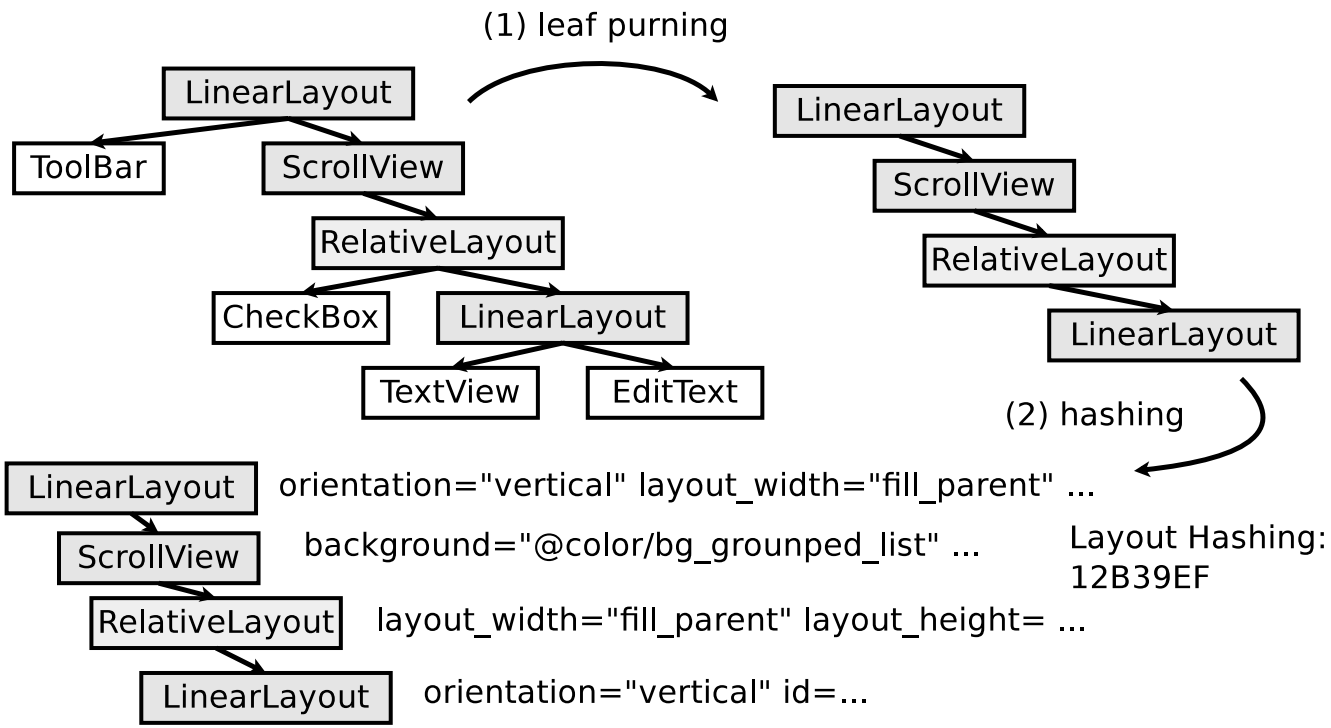
Host-based Detection

- safety and authenticity of apps from unknown sources
- host-based detection system for Android
- HostEagle can detect repackaged malware and phishing malware in-device
  - layout hashing, certificate extractor, detector, local detection & cloud detection



HostEagle: Host-based Detection – Layout Hashing

- leaf pruning: all leaves in the layout tree
  - `View` objects in leaves
  - *layout skeleton* of the user interface
- tree hashing



Evaluation – Repository Statistics

We crawled and collected 100,126 apps from

- Google official market
- third party markets
- public cloud storage

Category	Name	URL	# of Apps	Size
Official	Google Play	play.google.com	500	7.0 GB
Third-party	appchina	appchina.com	34 989	238 GB
	appfun	appfun.cn	12 427	154 GB
	hiapk	apk.hiapk.com	5287	87 GB
	android.d.cn	android.d.cn	4064	163 GB
	jimi168	jimi168.com	23 723	76 GB
	anzhi	anzhi.com	18 736	118 GB
Cloud Storage	Baidu	pan.baidu.com	200	3.5 GB
	Huawei	dbank.com	200	3.1 GB
Total			100 126	849.6 GB

Evaluation – Results of Repository Analysis

RepoEagle statically analyze apps in the repository.

- most of repackaged apps are cracked games with unlocked in-app paid markets
- malware samples are found in public cloud storage
  - hide identity
  - spread in forum and social media

Market	**# of Visually Similar Apps (Percentage)**	**# of Malware**
Third-party Market	1159 (1.6 %)	10
Cloud Storage (Baidu)	50 (10.0 %)	0
Cloud Storage (Huawei)	89 (17.8 %)	15

Experiment – Repackaged Apps

Case study of repository for Andry Bird app

- 8 apps which are visually similar with the official app “Angry Bird” from the Google Play
- different certificate issuers
- detection results: DroidKungFu malware can contact remote server and download malware, gain the root privilege and prevent uninstalling.

App ID	LED	LH	Certificate Issuer	Cert	Rpkg	Market	Detection Result
22d3	0	1cd5	Rovio Mobile Ltd.	5557		Google Play	
233c	0	1cd5	Rovio Mobile Ltd.	5557		appfun.com	
5803	0	1cd5	Rovio Mobile Ltd.	5557		appfun.com	
666c	0	1cd5	Virtuous Ten Studio	A925	✓	appfun.com	Android.Adware. Jumptap.a
7a43	0	1cd5	Rovio Mobile Ltd.	5557		appfun.com	
9ee4	0	1cd5	databin	FC00	✓	appfun.com	
22d3	0	1cd5	Rovio Mobile Ltd.	5557		android.d.cn	
0f6f	0	1cd5	android-debug	264B	✓	android.d.cn	Android.Adware. Dowgin
3b52	0	1cd5	keystore3	990B	✓	dbank.com	Android.Trojan. DroidKungFu
ed0e	2	1cd5	Rovio Mobile Ltd.	5557		jimi168.com	

Experiment – Phishing Malware

Experimental result of layout hashing for phishing malware (fake apps).

- FakeAV masquerades as the “Avast” anti-virus software
- FakeMart masquerades as the Google official market “Google Play”
- Agent: network agent utility

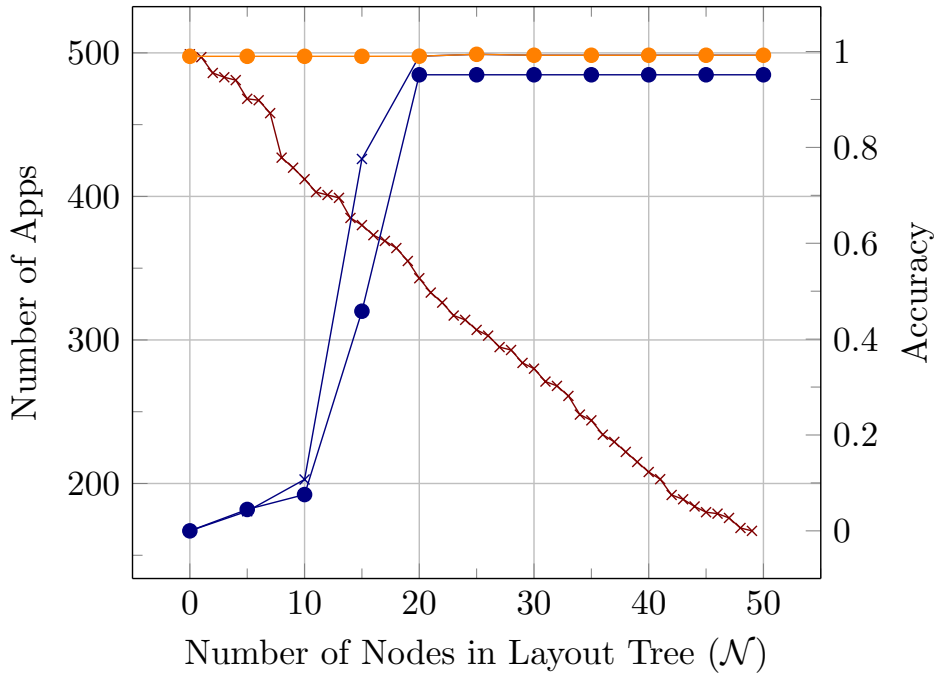
Name	# of Samples	Layout File	LH	Time
FakeAV	8	activity_scanning.xml	5a7e	0.466
FakeMart	3	main.xml	d41d	0.355
Agent	5	activity_main.xml	9344	0.501

* Generation time of LH in second on Nexus 5.

Evaluation – Detection Accuracy

The impact of nodes number in layout files in the repository analysis system.

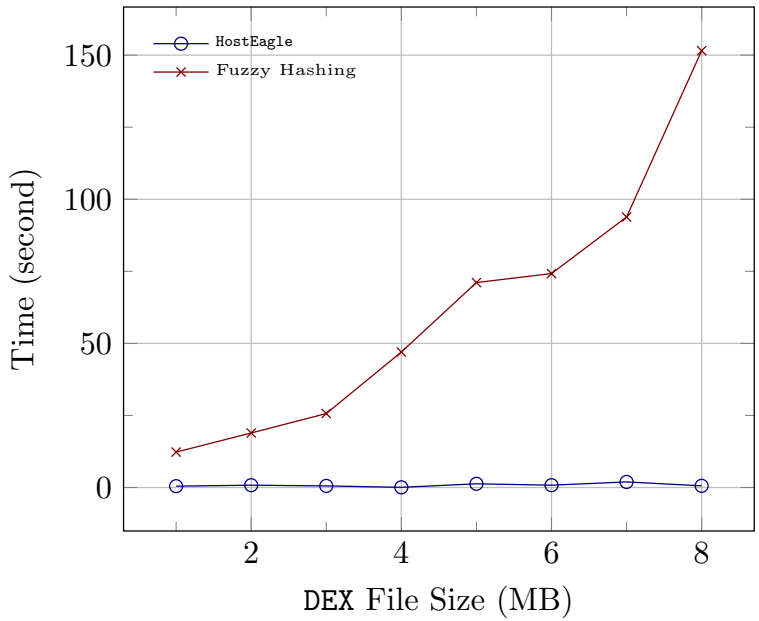
- number of qualified apps
- detection accuracy of repackaged “Fruit Ninja” and “PPS” respectively



Evaluation – Detection Efficiency

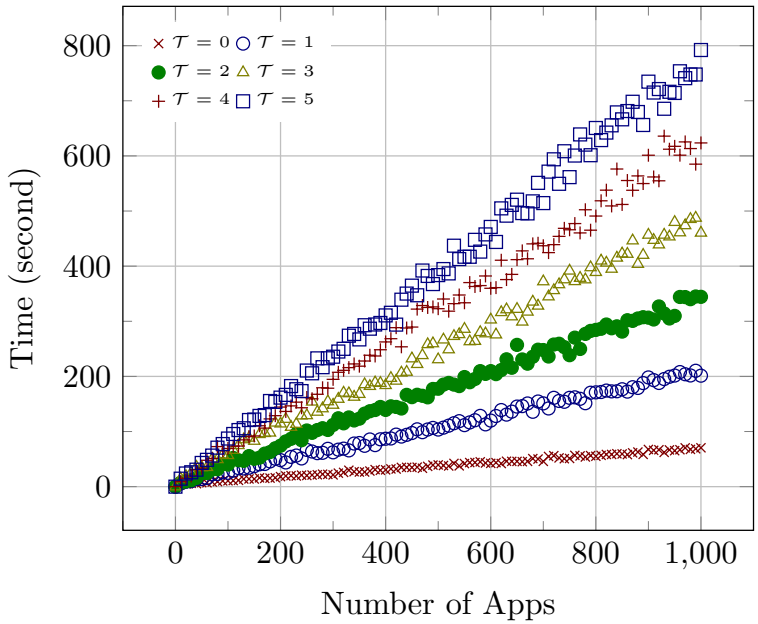
Time of hash value generation on Android device for different sizes of DEX file.

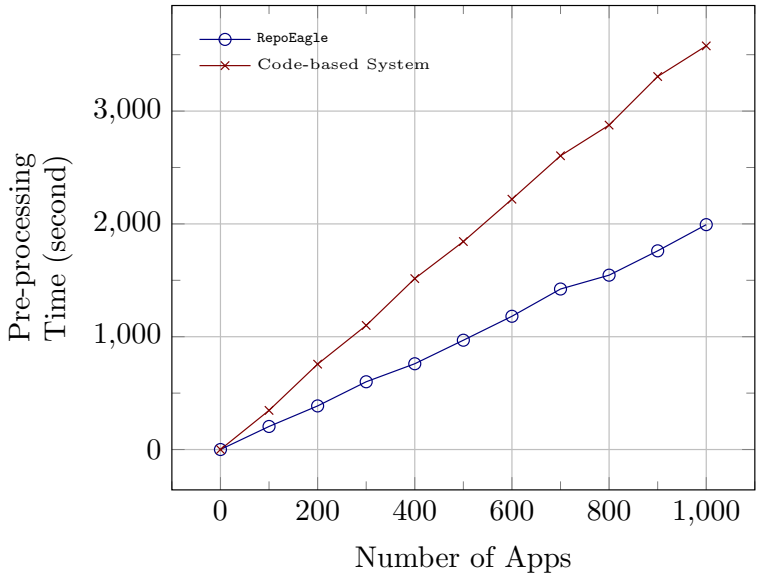
- traditional fuzzy hashing method should disassemble the source code, which will be affected by dex file size
- HostEeagle will not be affected



Evaluation – Others

- relationship between threshold and the analysis time
- pre-processing time for different numbers of apps





Conclusion

- repackaged malware and phishing malware
- detection based on visual resource: layout file
- 3 hours, 1298 visually similar apps with 25 malware

Questions?