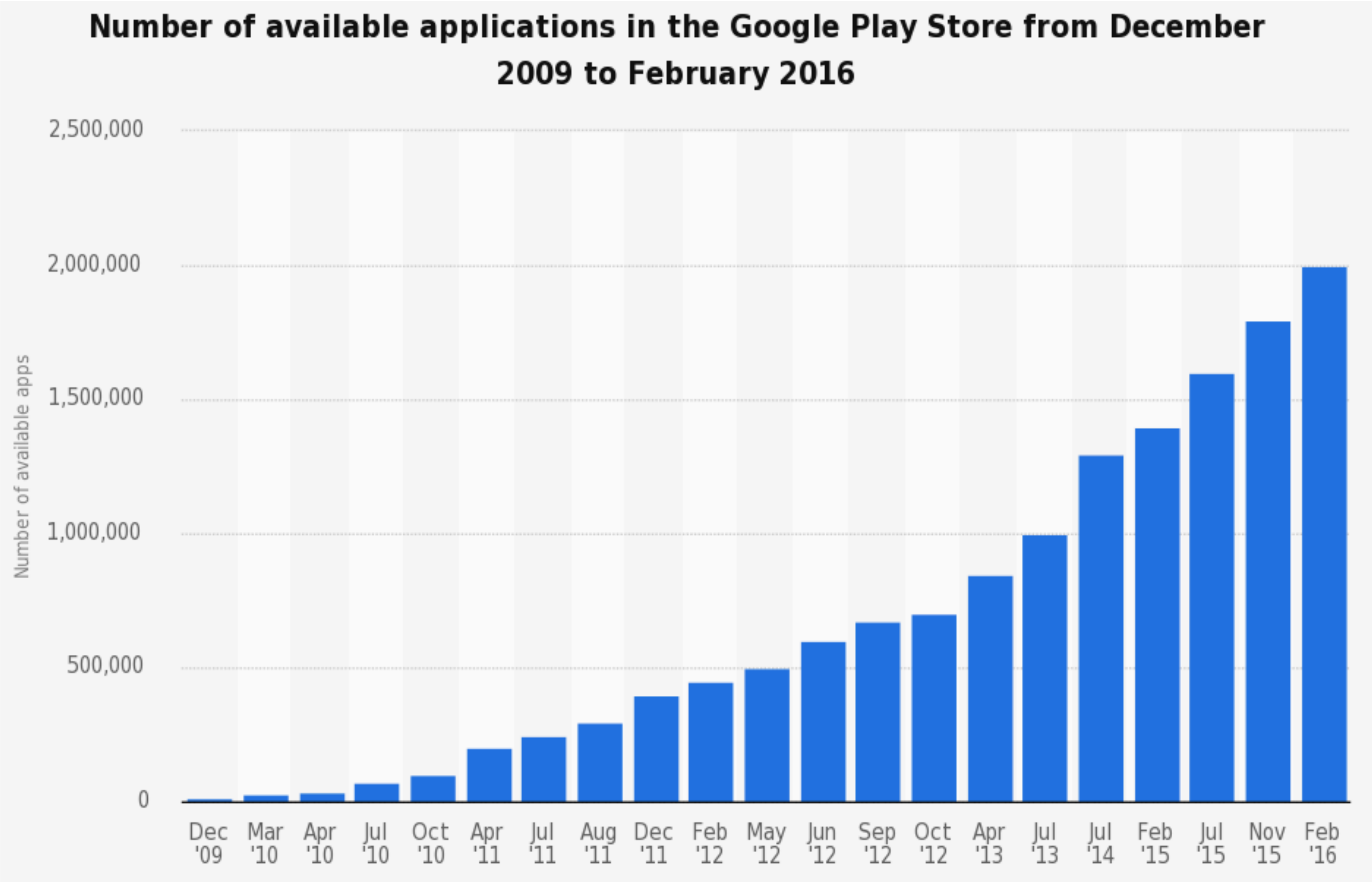
Mobile App Security: Detection and Family Identification of the Malice in Your Pocket

Sam Malek
Associate Professor
Institute for Software Research
University of California, Irvine
malek@uci.edu

Software Engineering and Analysis Lab
SEAL

ISR Research Forum, May 27, 2016

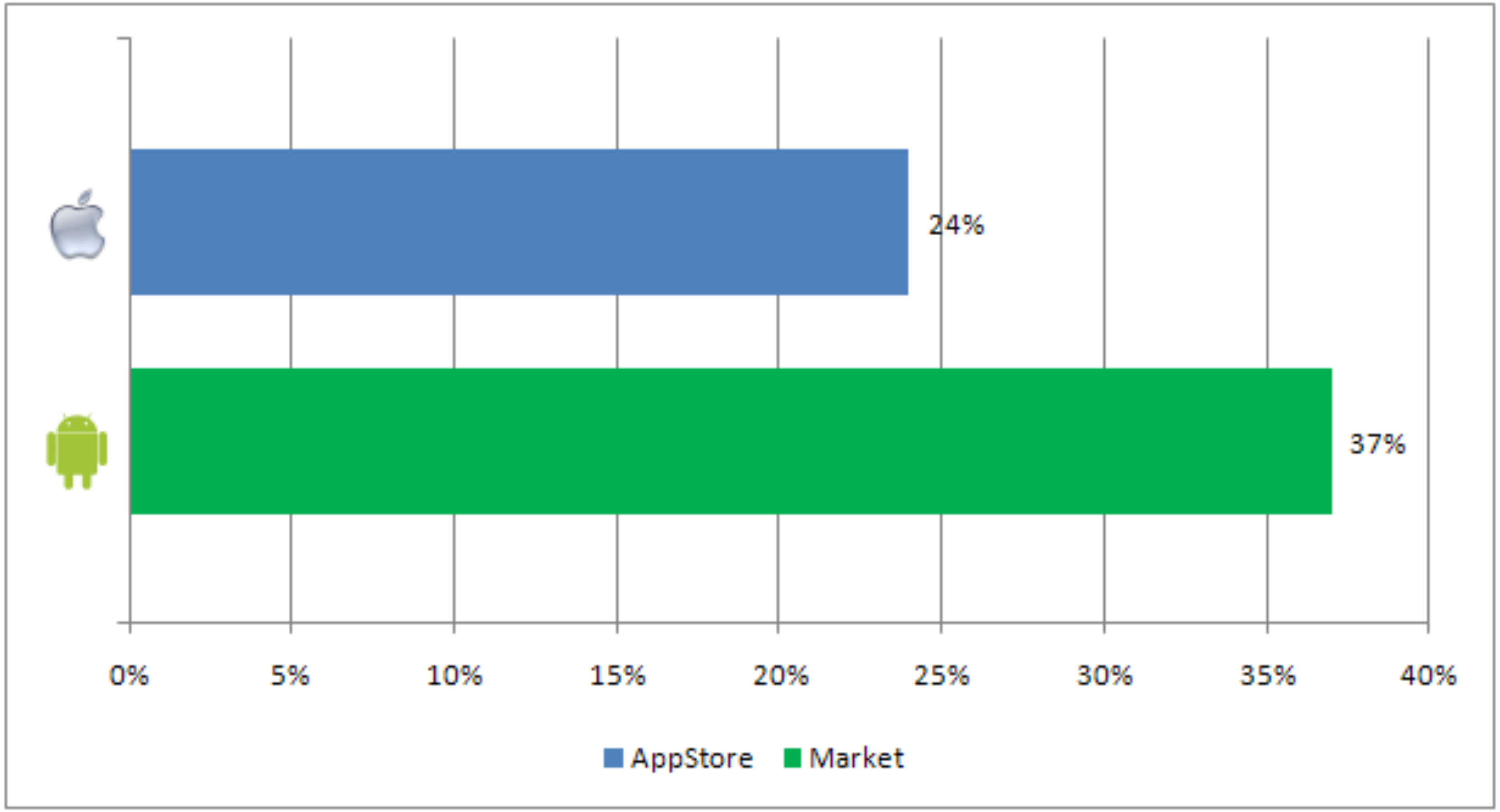
UCIRVINE



Source: Statista 2016

Typical App Developer

Many Low Quality Apps



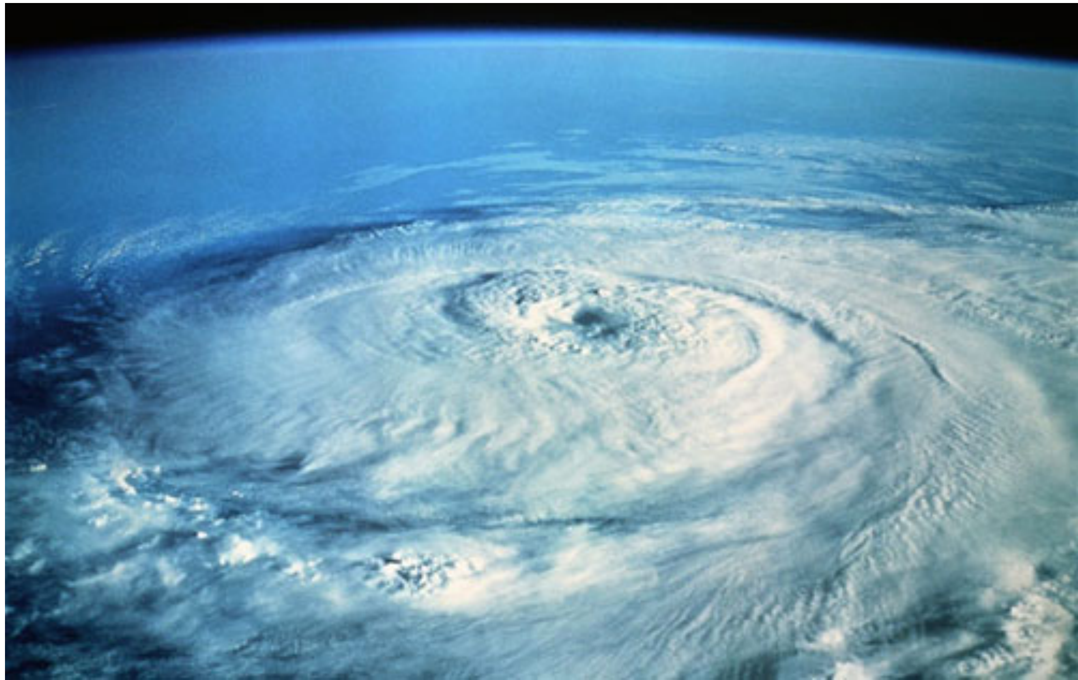
Source: Research2Guidance

Potentially have Access to Lots of Private Data

- Camera
- Microphone
- Accelerometer
- Gravity sensor
- Linear acceleration sensor
- Magnetic field sensor
- Orientation sensor
- Gyroscope
- Light sensor
- Proximity sensor
- Temperature sensor
- Pressure sensor

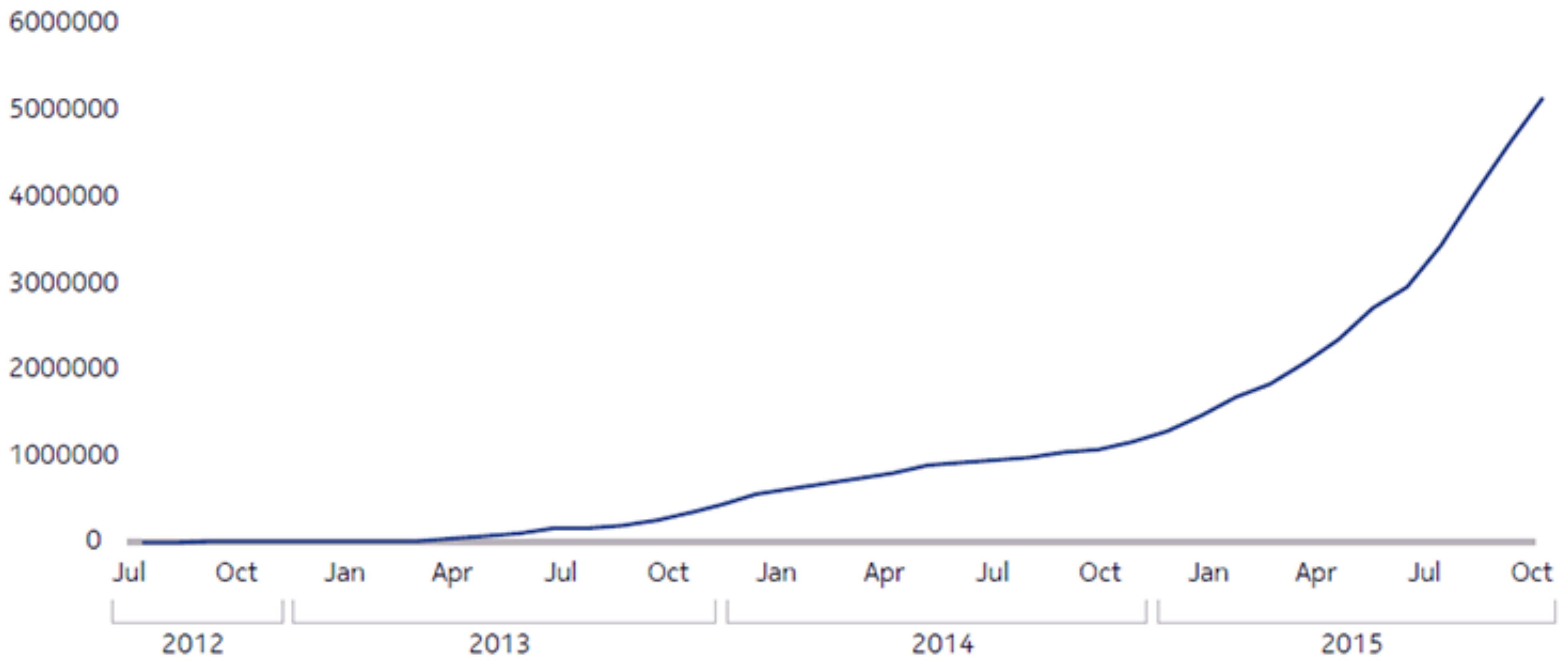
Perfect Storm

- App markets → best tool ever known to attackers for delivering malicious payload
- Market operators are challenged by the limitations of program analysis
  - Halting problem
- Lots of riches to be gained
  - Premium numbers
  - Adware
  - …



Malicious Android Apps

- Immense number of Android malware apps
  - 342% growth in 2015



Source: Calyptix Security

Malware Family

- GingerMaster
  - First Android malware using root exploit
  - Steal sensitive info (IMEI, SIM card number, etc.)
- DroidJack
  - No root access required
  - Remote Access Tool
  - Update itself
  - Record phone calls and audio
  - Steal sensitive info
- …

Countermeasures

- Detection and removal

Countermeasures

- Detection and removal is not enough

Countermeasures

- Detection and removal is not enough---identify **families**

Countermeasures

- Detection and removal is not enough---identify **families**
- Malware likes to **hide**

Countermeasures

- Detection and removal is not enough---identify **families**
- Malware likes to **hide**
- Catch them **fast**

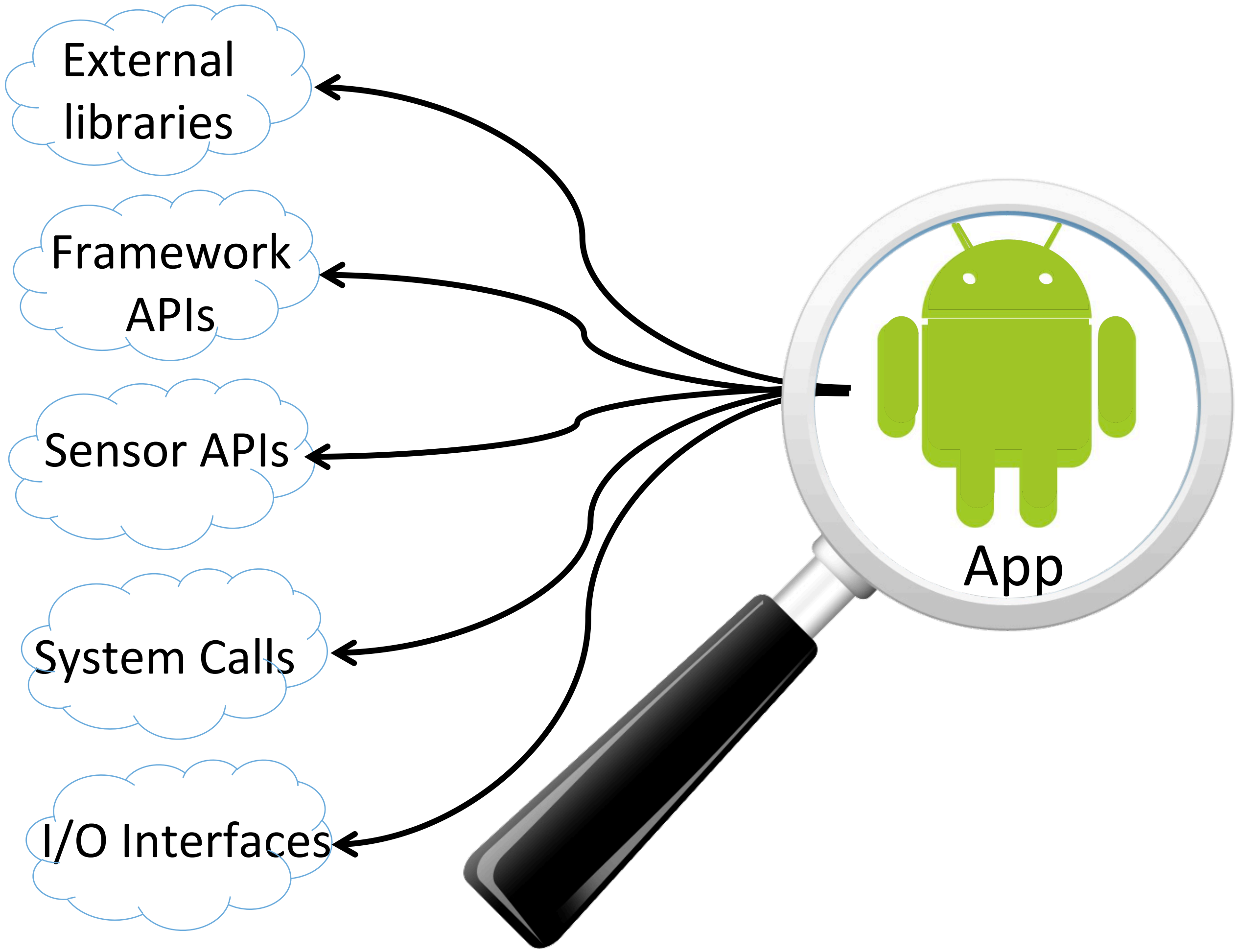
Our Research

1. Is it possible to learn what makes an app malicious?

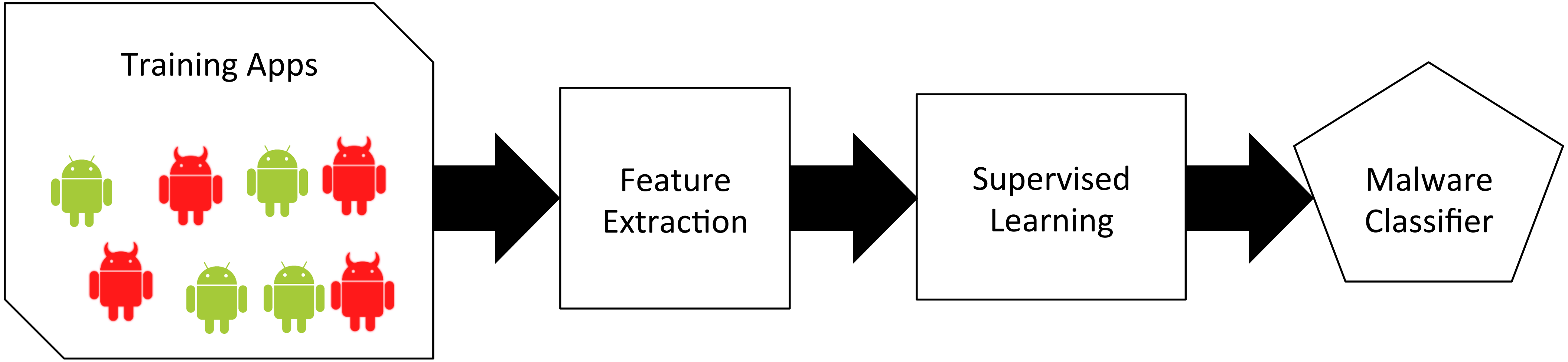
2. If so, is it possible to automatically learn the family of malicious apps?

RevealDroid

- A machine learning-based approach for malware detection and family identification
  - Accurate
  - Highly efficient
  - Obfuscation-resilient



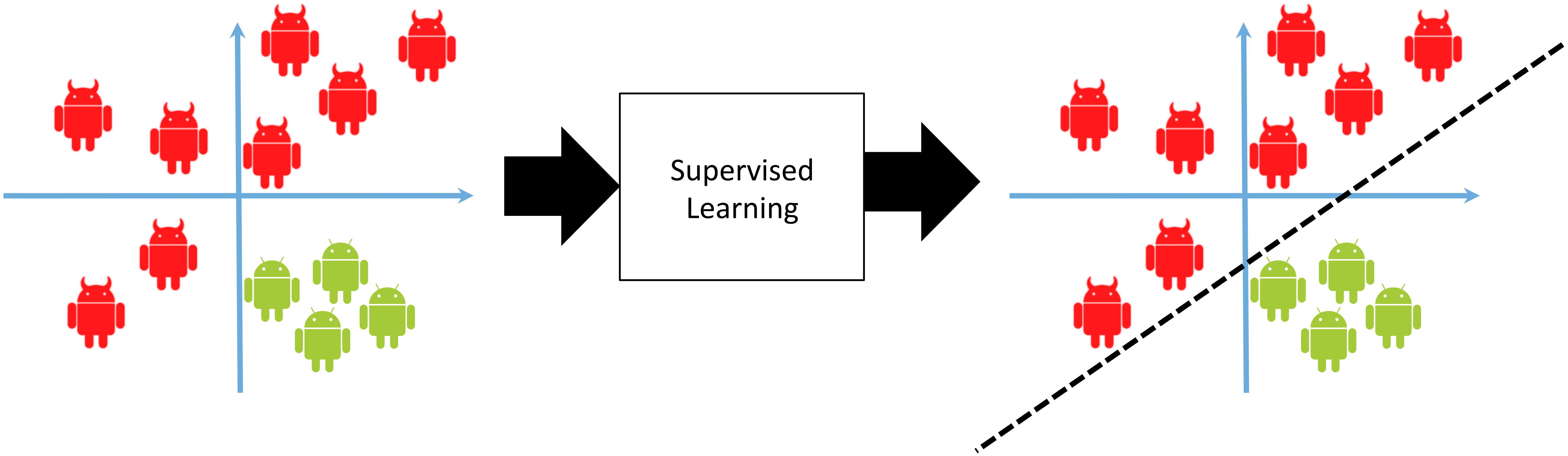
Classifier Construction for Malware Detection



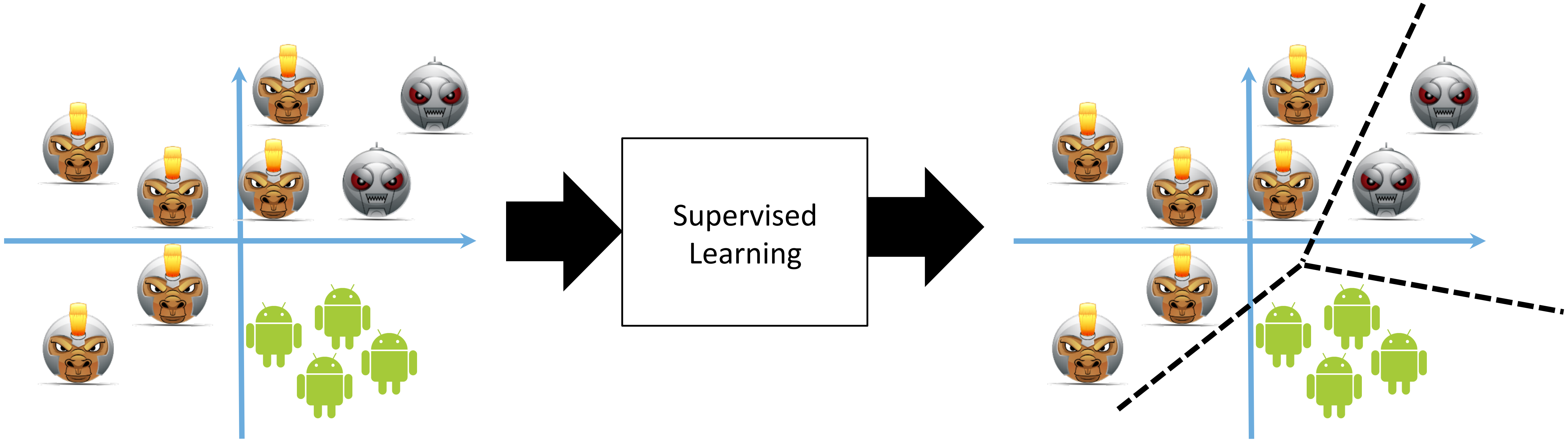
App Representation for Supervised Learning

App	Feature1	Feature2	Feature3	Feature4	Label
	1	0	0	0	Malicious
	0	1	0	0	Malicious
	0	0	1	1	Benign
	0	0	1	0	Benign

Supervised Learning for Malware Detection



Supervised Learning for Family Identification



Feature Selection

	Perm	Comp	IFilters	Flows	UAPI	PAPI	SAPI	IActions	Reflection	Native
Accuracy	✗	✗	✗	✔	✔	✔	✔	✔	✔	✔
Efficiency	✔	✔	✔	✗	✗	✔	✔	✔	✔	✔
Obfuscation	✔	✗	✗	✗	✔	✔	✔	✗	✔	✔

Feature Examples: Package API (PAPI)

- Numbers of Android API methods invoked by app per package
  - android.telephony
    - TelephonyManager.getCellLocation()
    - CellIdentityLte.getCi()

	telephony	location	sqlite	Fam
mal1	8	0	2	jSMSHider
mal2	0	12	0	Geinimi
mal3	2	0	7	BaseBridge

Feature Examples: Reflective Calls

- Apps may dynamically load libraries/classes through reflection
  - Used frequently to obfuscate malicious behavior

```
ClassLoader cl = MyClass.getClassLoader();
try { Class c = cl.loadClass("MyActivity");
      ...
      Method m = c.getMethod("onPause",...);
      ...
      m.invoke(...); }
catch { ... }
```

Feature Examples: Native Calls

- Apps can make system calls and calls to native binaries
  - Analysis of native binaries requires disassembly of ELF files

```
99ec: e59d0010 ldr r0, [sp, #16]
99f0: e59f13c0 ldr r1, [pc, #960]
99f4: ebfffc3e bl  8af4 <chmod@plt>
```

Code segment where *chmod* is invoked in *GingerBreak* malware

Labeling and Classifier Selection

- Classifier for detection
  - 2-way classifier with labels “benign” or “malicious”
  - Support Vector Machine (SVM)
- Classifier for family identification
  - *n*-way classifier where *n = the number of families*
  - Classification and Regression Trees (CART)

Experiments

Experimental Setup

- Prototype built using open-source software
  - Java-based
- Over 23,300 benign and 28,100 malicious apps
  - Collected from Malware Genome, Drebin, and Virus Share repositories
- 68 different malware families

Detection accuracy on non-obfuscated apps

	Precision	Recall	F1
Benign	95%	85%	90%
Malicious	89%	96%	92%
Average	92%	91%	91%

Greater than 90% precision and recall

Family identification accuracy on non-obfuscated apps

	No. Apps	No. Families	Correct Classification Rate
Malware Genome	1,250	49	92%
Virus Share	18,065	68	87%

A random classifier would obtain only 1.5% correct classification rate

Detection accuracy on obfuscated apps

- Testing apps were obfuscated using DroidChameleon
  - Shown to evade all commercial antivirus products
  - String/Array encryption, class renaming, call indirection, etc.

	Precision	Recall	F1
Benign	96%	70%	81%
Malicious	82%	98%	89%
Average	89%	84%	85%

Family identification accuracy on obfuscated apps

	No. Apps	No. Families	Correct Classification Rate
Malware Genome	1,188	49	94%

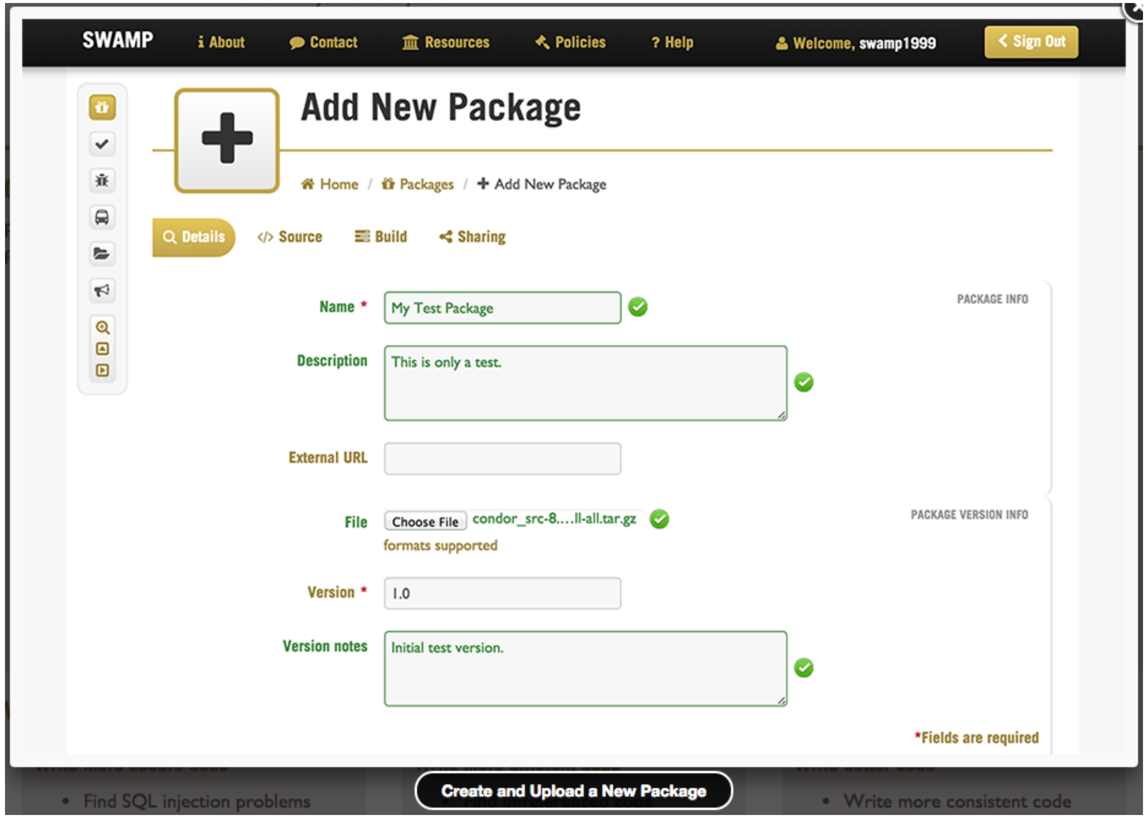
Performance

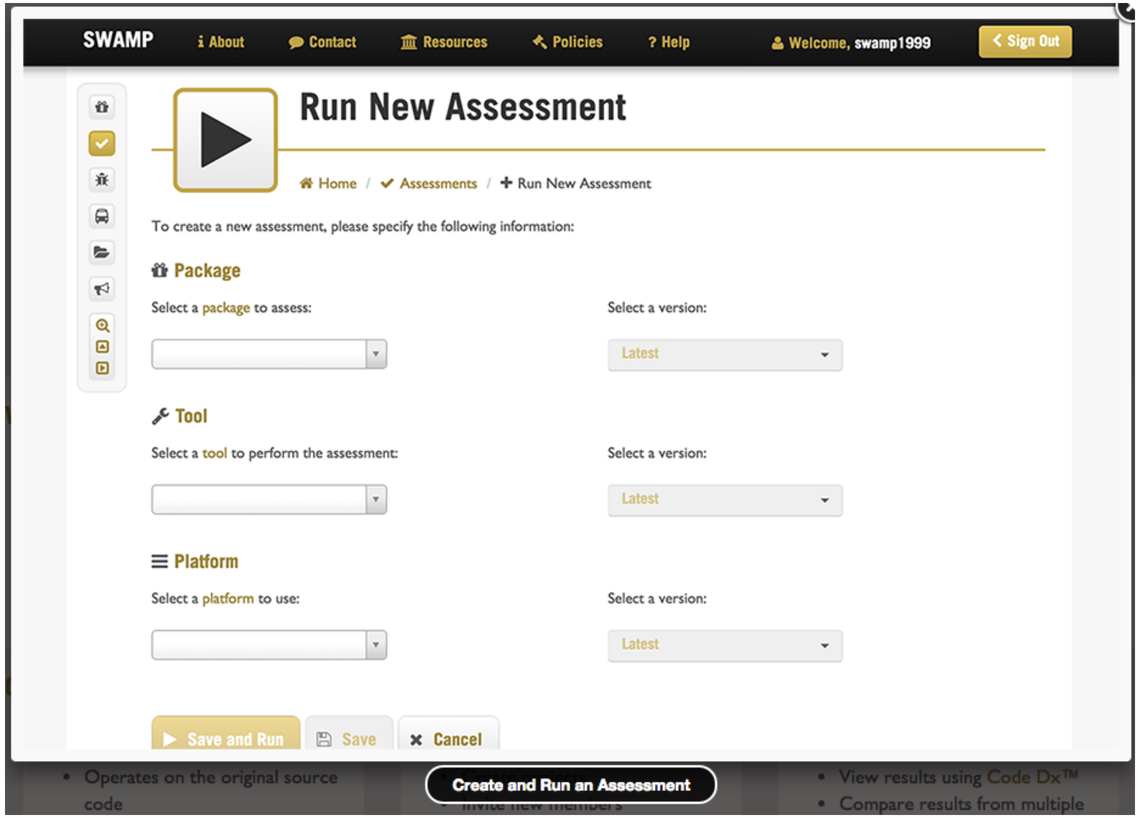
No. of apps	Feature Extraction			Classification (s)
	Native (s)	Reflection (s)	PAPI (s)	
100 randomly selected	18	31	24	2

It takes around 30 seconds to run RevealDroid on an app

Department of Homeland Security

- Available for use through the SWAMP portal
  - https://continuousassurance.org/





Conclusion

- **RevealDroid**
  - A machine-learning based approach for malware detection and family identification
  - Highly accurate, obfuscation resilient, and fast

- **Acknowledgement**
  - Joshua Garcia
  - Mahmoud Hammad
  - Kari Nies

Backup

Mobile Software Ecosystems

- Successful software platforms open themselves to third party developers, resulting in massive product lines
  - E.g., Android app ecosystem

1,200 different Android devices

46 different versions of OS