securityandtrust.lu

Challenges in Android Malware Detection

OWASP BeNeLux Days @ Belval
Friday, 18th March 2016

KEVIN ALLIX, kevin.allix@uni.lu

SnT, Université du Luxembourg

UNIVERSITÉ DU LUXEMBOURG

Contents

Android

Android in one minute

- A complete Software Stack
- Linux based Kernel + custom (Non POSIX) Libc
- Dalvik Virtual Machine
- Userland Apps written in Java, and compiled to Dalvik ByteCode
- Self-contained Applications packaged in One file
- Solid User Base (Billion)
- Strong ecosystem (Millions of Apps)
- + *alternative* markets (AppChina, Amazon, Opera, GetJar, etc.)

With great market shares comes great risks

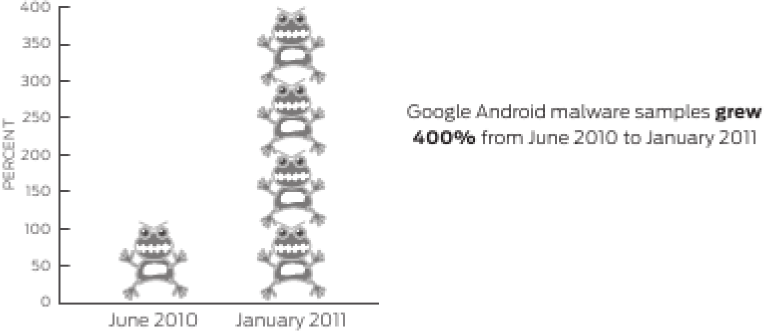
Android is becoming Ubiquitous

SnT
securityandtrust.lu

- A lot of device types (Media Players, STBs, DECT Phones, etc.)
- Huge amounts of personal data on each device
- Connected to both the phone network and the Internet

A target of choice for attackers

The Rise of the Malware



[1]

- Bank Phishing Apps (2010)
- Botnet (2010)
- GPS tracking disguised as a game (2010)
- SMS Trojan, SMS Leakage, Contacts Leakage, etc.
- Without using any Exploit (i.e. without breaking the permission-based security model)

[1] *Malicious Mobile Threats Report 2010/2011*, Juniper Networks, 2011

Research Question

Android Malware Detection

- How can we detect Malware Applications?

Android Malware Detection I

SnT
securityandtrust.lu

The traditional Antivirus method

- Collect supicious samples
- Analyze each sample (Static and/or dynamic analysis)
- Extract a *signature*

What I'm trying to do

- Given a set of known malware
- And given a set of known goodware
- Use Data Mining to detect unknown malware samples

Android Malware Detection II

securityandtrust.lu

Machine-Learning Android Malware : A Recipe

- Extract a Feature Vector from each known Malware sample;
- Extract a Feature Vector from each known Goodware sample;
- Extract a Feature Vector from an unknown Android App;
- Add some Machine Learning Magic.

Machine Learning I

Feature Vector…

- A *Feature* is just a characteristic, a property, a trait
- Example for Human Beings: Age, Gender, Height, Weight, Skin color, Eye color, Hair color, etc.
- Can you spot correlations between those variables?
- Can you spot variables that would allow to guess the variable *Gender*?

  $\rightarrow$ **Machine Learning finds correlations between variables**
- Machine Learning will spot that on average, men are taller than women

securityandtrust.lu

Feature Matrix

Example with 2 Features and One class:

Height	Weight	Gender
185.42	70.3	male
172.72	60.3	female
185.42	70.3	male
157.48	49.9	female
180.34	68.0	male
170.18	68.0	female
172.72	70.3	male
156.845	48.9	female
⋮	⋮	⋮

Back to Android Malware

SnT
securityandtrust.lu

What Features to detect Malware ?

→Put everything you can think of that *may* be statistically different for malware.

Back to Android Malware

securityandtrust.lu

What Features to detect Malware ?

→Put everything you can think of that *may* be statistically different for malware.

!TROLL ALERT! Have no idea at all ?

- You don't know what you're doing ?
- YOLO ?

Back to Android Malware

What Features to detect Malware ?

→Put everything you can think of that *may* be statistically different for malware.

!TROLL ALERT! Have no idea at all ?

- You don't know what you're doing ?
- YOLO ?
- "Deep Learning" is made for you !

Two Families of features

Static Analysis

- +Can be fast
- +Can be relatively simple
- −Blind to many things

Dynamic Analysis

- +Can see more things (like downloaded code)
- −Can see more things (so much data)
- −Cannot be fast
- −Exercising apps ? Fuzzing a GUI is highly inefficient, and not necessarily effective

Given the cost in time and CPU of dynamic Analysis, most researchers go the static Analysis way

Evaluation

securityandtrust.lu

But features are just the first step

- Now you need to evaluate the performance of your malware detecor…
- That's incredibly hard to do properly

A few examples of issues…

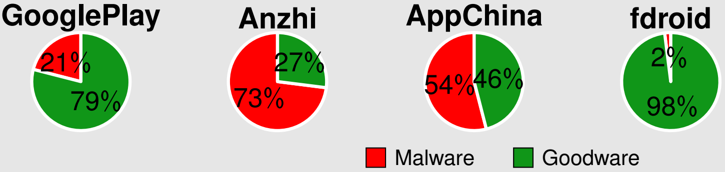
What is a Malware?

Remember the scary Juniper graph?

What is a Malware?

Remember the scary Juniper graph?

I can do scary graphs as well

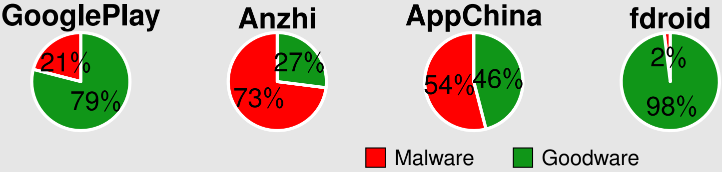


(Malware == detected by at least 1 Antivirus)

What is a Malware?

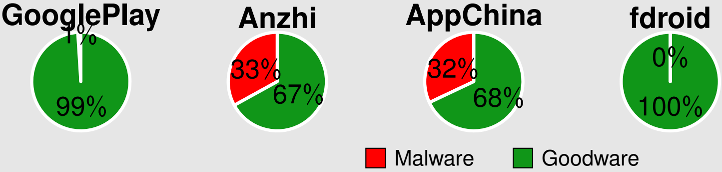
Remember the scary Juniper graph?

I can do scary graphs as well



(Malware == detected by at least 1 Antivirus)

That one is slightly less scarry



(Malware == detected by at least 10 Antivirus)

Ground-truth

To do Machine Learning, we need:

- A set of known Malware
- A set of known Goodware

There are a few (small) sets of known Malware.
Interestingly, there is no set of known Goodware.

Using AntiVirus Products

- Not every AV agree

Ground-truth

To do Machine Learning, we need:

- A set of known Malware
- A set of known Goodware

There are a few (small) sets of known Malware.
Interestingly, there is no set of known Goodware.

Using AntiVirus Products

- Not every AV agree
- Well. . . All AVs Disagree

Ground-truth

To do Machine Learning, we need:

- A set of known Malware
- A set of known Goodware

There are a few (small) sets of known Malware.
Interestingly, there is no set of known Goodware.

Using AntiVirus Products

- Not every AV agree
- Well. . . All AVs Disagree
- Some flag Adware
- Some Don't
- Some Do Sometimes

→ **AVs do NOT share a common definition of what is a Malware**

AVs

securityandtrust.lu

!TROLL ALERT! Increase your performance

AVs

!TROLL ALERT! Increase your performance

By choosing the definition that makes your detector look good.

In-the-Lab vs in-the-Wild

Size does matter

- Malware Detectors are often tested on very small datasets
- Their performance may be over-estimated

In-the-Lab vs in-the-Wild

Size does matter

- Malware Detectors are often tested on very small datasets
- Their performance may be over-estimated
- By a Whole lot

Data Leakage: The Time issue

securityandtrust.lu

One slight methodology problem. . .

- We don't know the future.

Data Leakage: The Time issue

securityandtrust.lu

One slight methodology problem. . .

- We don't know the future.
- Yes, I learned that during my PhD

Data Leakage: The Time issue

securityandtrust.lu

One slight methodology problem…

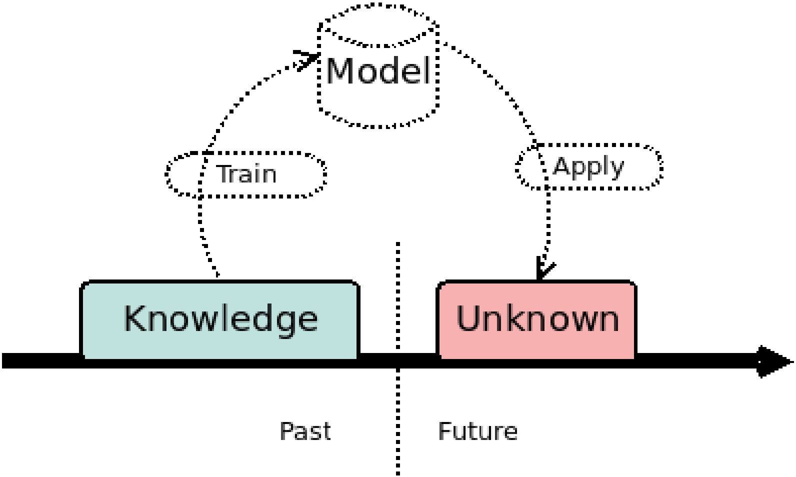
- We don't know the future.
- Yes, I learned that during my PhD
- "Science is a slow process"

The Time Issue: [*Back To The Future* Style]

- Testing an approach in an historically Incoherent way tells us:
  a) How this approach would perform Now on Malware from the Past
  or
  b) How this approach would have performed in the Past with (then-)Present Malware if it has had access to the (then-)Future
- However, it does Not tell us how it would perform Now on Present Malware

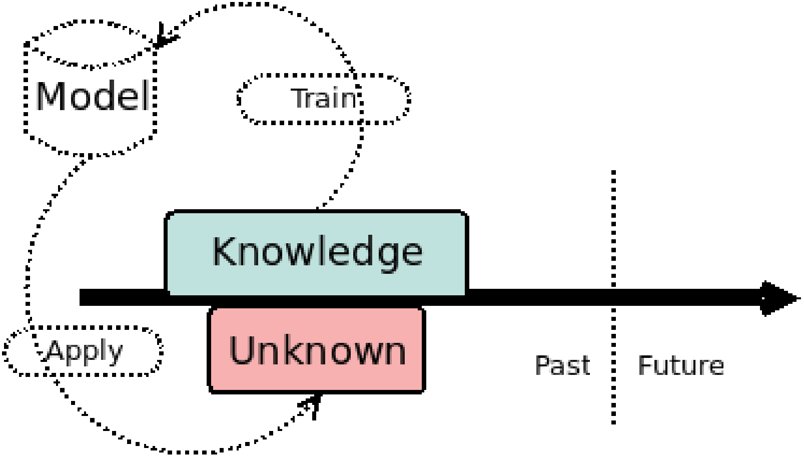
Does your brain hurt ?

Use-Cases I



Filtering / Finding brand new Malware

Use-Cases II



Cleaning Markets

State of the Art ?

- Nearly everyone does the time in-coherent way
- Knowing the Future helps a lot!
- I guess those two things are unrelated...

History Matters!

- History should be taken into account when evaluating a Malware detector;
- Approaches whose evaluation ignores History may actually perform badly where we need them most;

Conclusion

Automatic Malware Detection? We're not quite there...

What is needed ?

- Dependability, Dependability, Dependability
  Increase trust in Machine Learning-based malware detectors ?
  → Predicting performance where it cannot be assessed yet
  → Explanation
- Practicality
  How to tune an approach to match its user needs ?

Thank You!

securityandtrust.lu

Questions?