Institut
Mines-Telecom

Detecting Mobile Malware with Classification Techniques

Ludovic Apvrille
ludovic.apvrille@telecom-paristech.fr

Axelle Apvrille
aapvrille@fortinet.com

Labex UCN@Sophia, Security Day, Dec. 18th, 2013

Outline

Context

- So many Android malware!
- SherlockDroid

Alligator

- Main principles
- Learning stage
- Guessing stage

Results

Outline

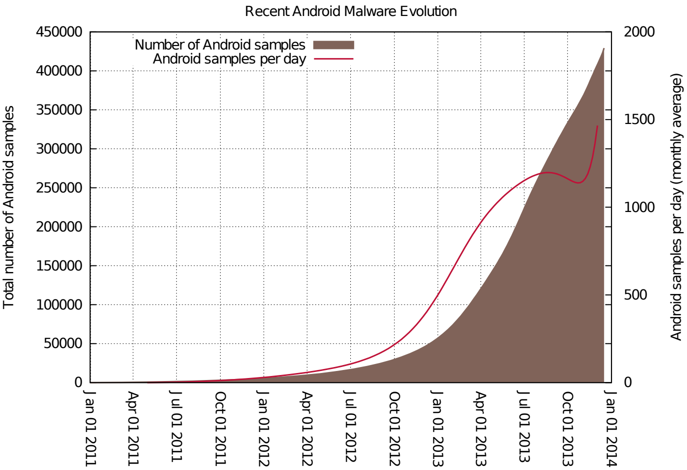
Context

- So many Android malware!
- SherlockDroid

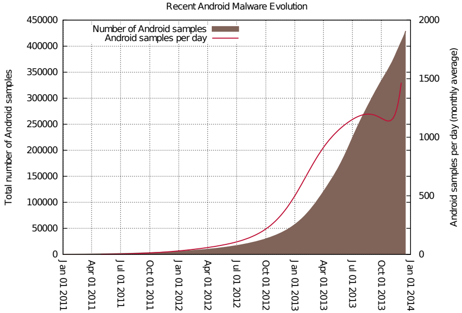
Alligator

Results

The Big Picture on Android Malware



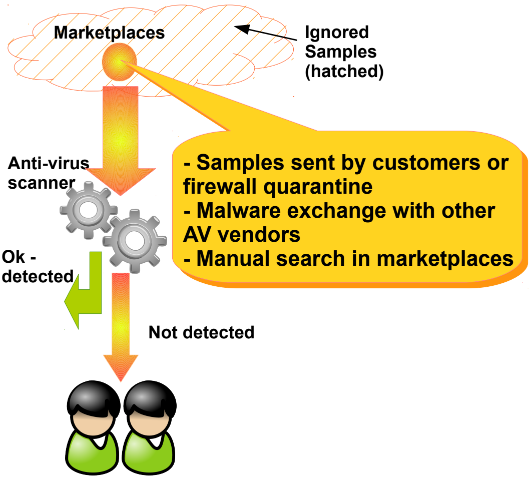
The Big Picture on Android Malware



Also, many malware remain undetected for a long time!

(Maybe you are currently using one on your mobile phone instead of listening to me?)

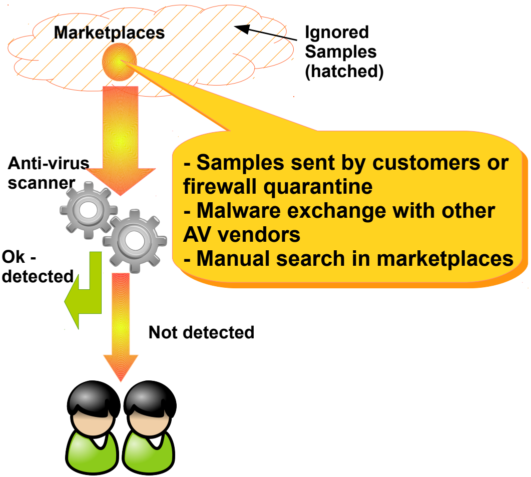
Are AV Analysts Lazy? No, Too Much Work!



- Samples sent by customers or firewall quarantine
- Malware exchange with other AV vendors
- Manual search in marketplaces

Manual inspection for advanced analysis by AV analysts and researchers

Are AV Analysts Lazy? No, Too Much Work!



Marketplaces

Ignored Samples (hatched)

Anti-virus scanner

Ok - detected

Not detected

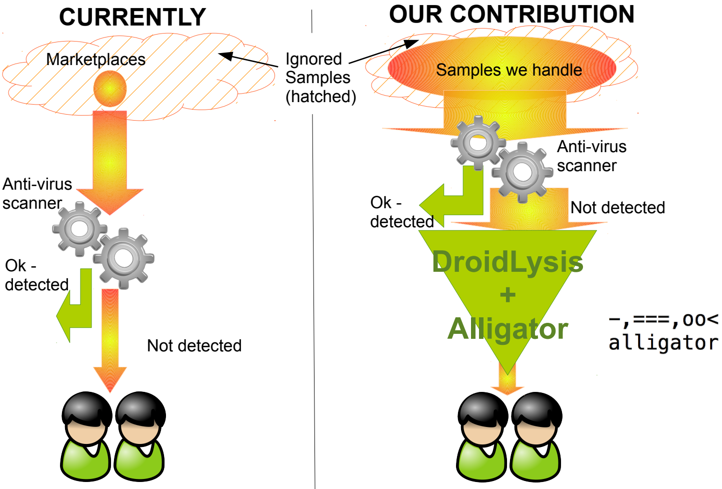
- Samples sent by customers or firewall quarantine
- Malware exchange with other AV vendors
- Manual search in marketplaces

Manual inspection for advanced analysis by AV analysts and researchers

Conclusion:

Smart filtering is necessary!

Prefiltering: Overview



CURRENTLY

Marketplaces

Ignored Samples (hatched)

Anti-virus scanner

Ok - detected

Not detected

Manual inspection for advanced analysis by AV analysts and researchers

OUR CONTRIBUTION

Samples we handle

Anti-virus scanner

Ok - detected

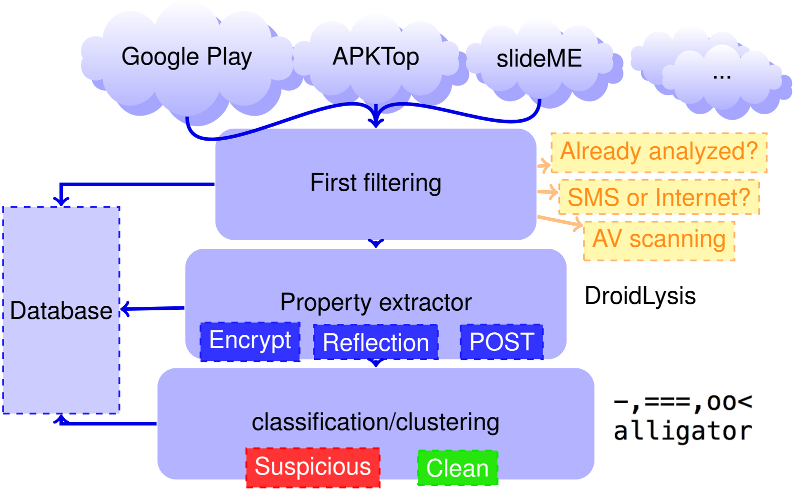
Not detected

DroidLysis + Alligator

-,===,oo<
alligator

Manual inspection for advanced analysis by AV analysts and researchers

SherlockDroid Architecture



Outline

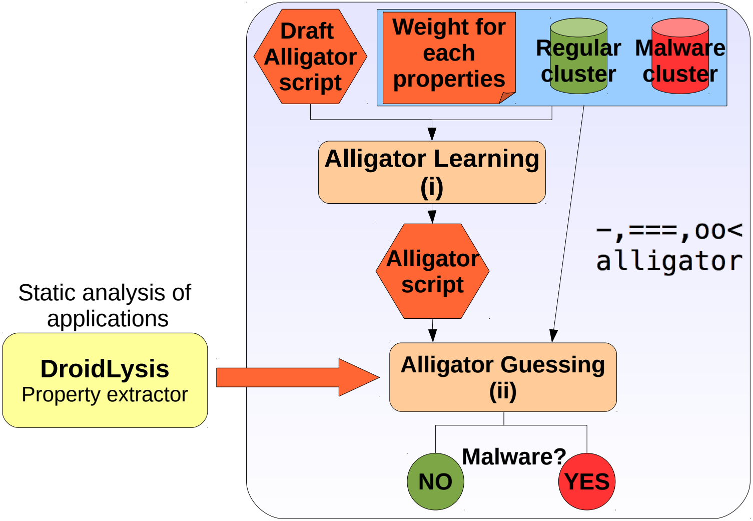
Context

Alligator

- Main principles
- Learning stage
- Guessing stage

Results

Fundamentals of Alligator



Yet Another Clustering Toolkit?

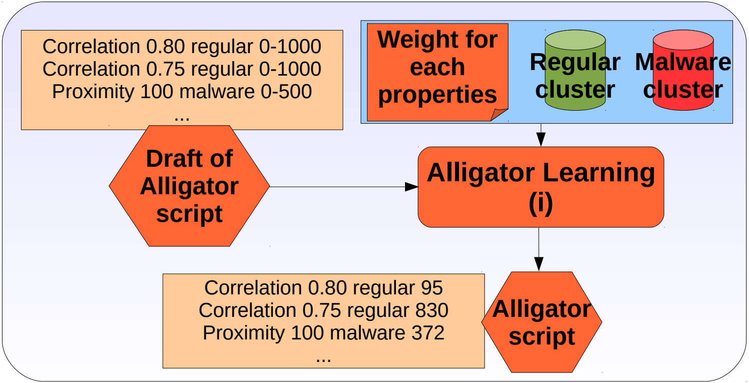
No! Alligator is much better!!!

- Dedicated to **work with two pre-known clusters**
- **Handles several up-to-date clustering algorithms at the same time**
  - Automatically determines how to combine them in an optimal way
- Option to settle a preference in **reducing false positive or negative**
- Very efficient - because we are very good programmers ;-)
- Free software
  - "Free": As in "free beer" AND as in "freedom" ;-)

Principle of Learning

Purpose

- Determining the importance to give to each couple (**clustering algorithm**, **parameter**)


Correlation 0.80 regular 0-1000
Correlation 0.75 regular 0-1000
Proximity 100 malware 0-500
...
Draft of Alligator script
Weight for each properties
Regular cluster
Malware cluster
Alligator Learning (i)
Correlation 0.80 regular 95
Correlation 0.75 regular 830
Proximity 100 malware 372
...
Alligator script

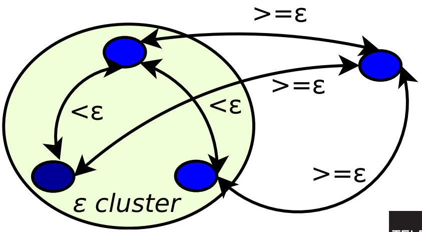
Clustering Algorithms

Cluster-center oriented algorithms

1. Standard deviation
2. Correlation
3. Probability difference
4. Probability factor

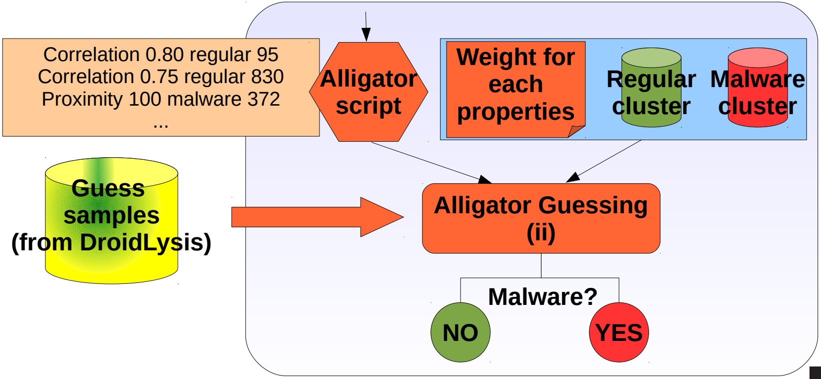
Neighbourhood oriented algorithms

5. Proximity (a.k.a. k-NN)
6. Proximity with limited properties
7. Epsilon clusters



Guessing Stage

Determining the cluster (regular, malware) of **unknown samples**



Correlation 0.80 regular 95
Correlation 0.75 regular 830
Proximity 100 malware 372
...

Outline

Context

Alligator

Results

Test Bench

Type of cluster	Malware samples	Regular samples	Period
Learning clusters	82,985	8,299	Before June 14
Guess clusters	19,171	1,103	From June 15 to June 24
Total of samples tested	102,156	9,402	

Number of samples in our test clusters

Test Bench (Learning Stage)

- All clustering algorithms considered with an average of 5 parameters for each
- Example:
  - Correlations: 0.80, 0.75, 0.70, 0.60
  - Epsilon clusters: ϵ-path of 10^{-5} to $10^{-}1$
- Computation time: around 10 hours on a non dedicated host

Results of Guessing Stage

Alligator was tested over those new sets of malware and clean files (20k new samples)

		Regular	Malware
Guessing	Number of failed / recognized	2 / 1,101	375 / 18,796
	Failure / success rates in %	0.18% 99.81%	1.96% 98.04%

Conclusions

SherlockDroid is efficient!

- SherlockDroid = efficient combination of market crawler + property extractor + clustering
- Large sets of clusters tested
- Objective reached: → 99.8% of clean applications are filtered out.
  - AV analysts can now be lazy ;-)
- Unknown malware discovered thanks to Alligator[a]
  - A new one discovered yesterday! *Android/MisoSMS.A!tr.spy*

[a] see e.g., http://blog.fortinet.com/Alligator-detects-GPS-leaking-adware/.

Conclusions (Cont.)

Limitations and Future work

- Clean cluster much smaller than malware cluster!
- More clustering algorithms
- Alligator could be used for many other purposes

```
-,===,oo<
alligator
```

Do Try Alligator!

```
-,===,oo<
alligator
```

perso.telecom-paristech.fr/~apvrille/alligator.html

(Are you sure your qr-code reader application is not a malware???)