

SAND2015-3420C

# Attacking DBSCAN for Fun and Profit

**Jonathan Crussell, Philip Kegelmeyer**

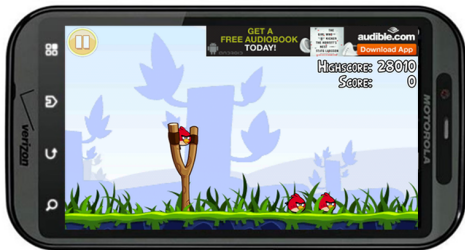
Sandia National Laboratories, California<sup>1</sup>

April 30th, 2015

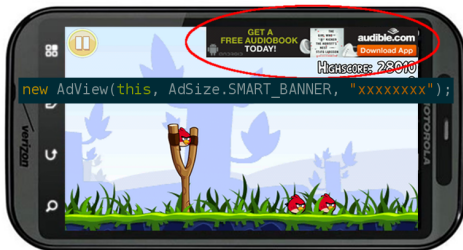
---

<sup>1</sup>Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

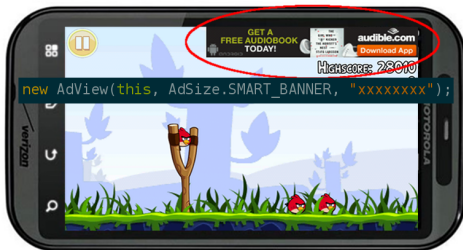
# App Plagiarism



# App Plagiarism



# App Plagiarism



Miscreants copy apps to siphon ad revenue

- Gibler et al. (MobiSys'13) estimate losses of 14%

# AnDarwin

AnDarwin (Crussell et al., ESORICS'14):

- Crawled 265K apps from 17 Android markets

# AnDarwin

AnDarwin (Crussell et al., ESORICS'14):

- Crawled 265K apps from 17 Android markets
- Detected copied apps via clustering based on DBSCAN

# AnDarwin

AnDarwin (Crussell et al., ESORICS'14):

- Crawled 265K apps from 17 Android markets
- Detected copied apps via clustering based on DBSCAN
- One application: plagiarism detection

# AnDarwin

AnDarwin (Crussell et al., ESORICS'14):

- Crawled 265K apps from 17 Android markets
- Detected copied apps via clustering based on DBSCAN
- One application: plagiarism detection
- Designed to be robust to attacks against data representation

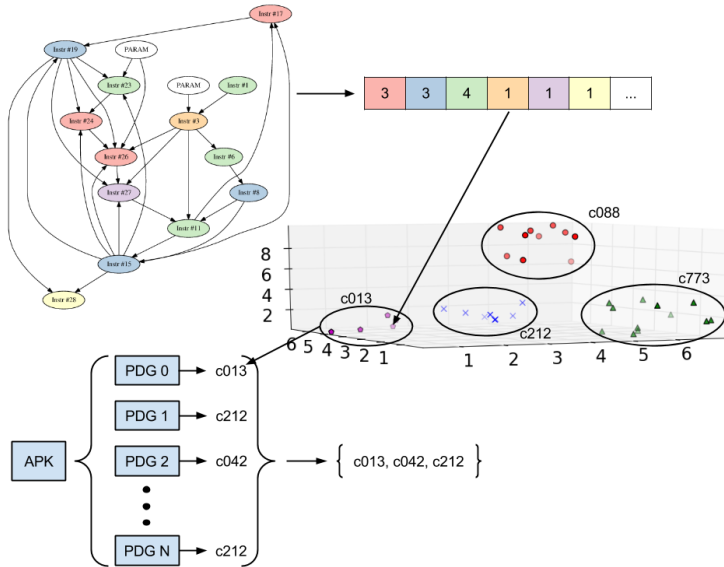


# AnDarwin

AnDarwin (Crussell et al., ESORICS'14):

- Crawled 265K apps from 17 Android markets
- Detected copied apps via clustering based on DBSCAN
- One application: plagiarism detection
- Designed to be robust to attacks against data representation
- \*Not\* designed to be robust to attacks against data analysis

# AnDarwin



# Thinking like an Adversary

What goals might an adversary have?

- Avoid being clustered with similar apps
- Favorably alter clustering structure
- ...

# Thinking like an Adversary

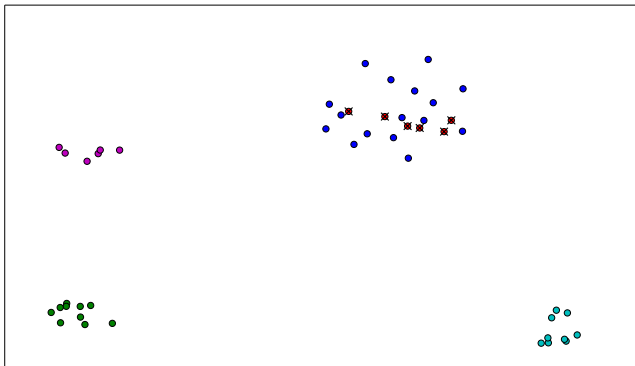
What goals might an adversary have?

- Avoid being clustered with similar apps
- Favorably alter clustering structure
- ...

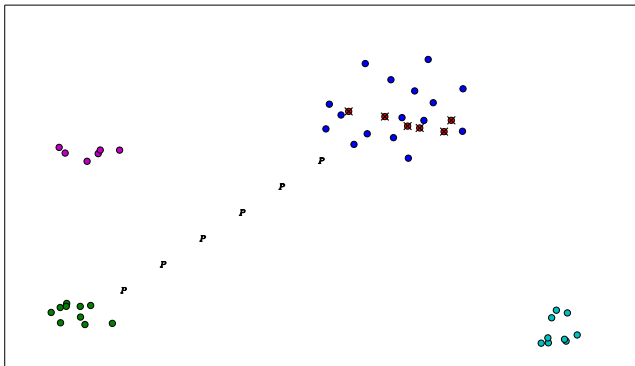
*Confidence Attack*

- Inject new points into dataset to poison the clustering

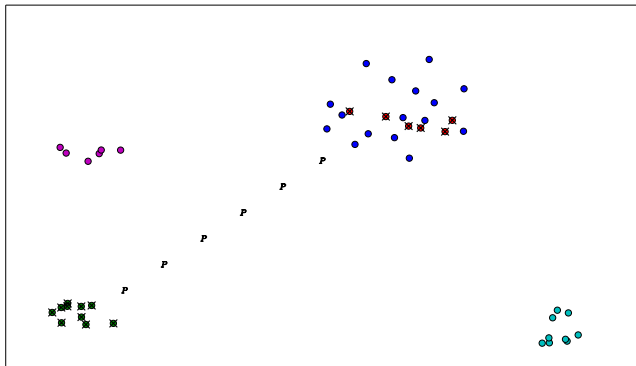
# Confidence Attack



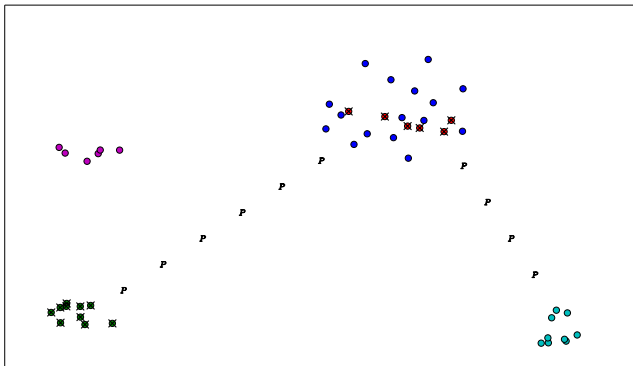
# Confidence Attack



# Confidence Attack

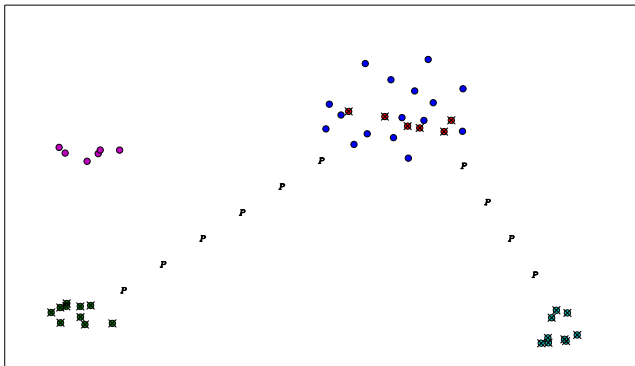


# Confidence Attack

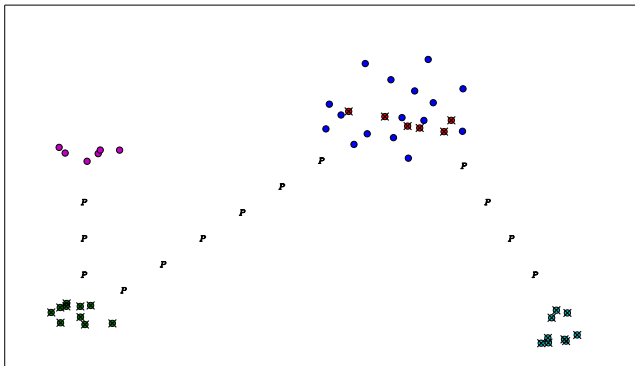




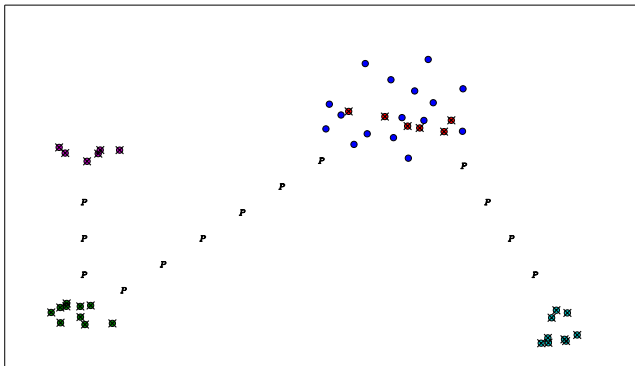
# Confidence Attack



# Confidence Attack

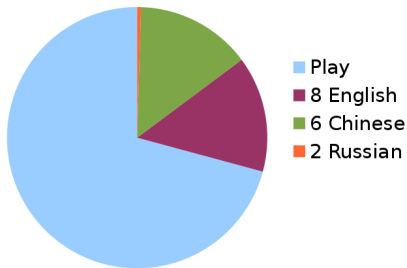


# Confidence Attack



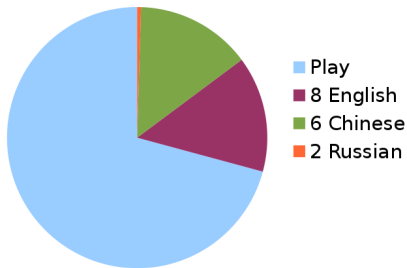
## Is this Feasible?

In most cases, we analyze “found data:”



## Is this Feasible?

In most cases, we analyze “found data:”



Semantic Gap (Jana and Shmatikov, IEEE S&P'12)

- Program analysis vs program execution

# Attack Methodology

1. Pick two clusters to merge

# Attack Methodology

1. Pick two clusters to merge
2. Generate series of optimal data mines between two clusters

# Attack Methodology

1. Pick two clusters to merge
2. Generate series of optimal data mines between two clusters
3. Goto 1 until all desired merges completed



## Generating Data Mines

AnDarwin represents apps as sets

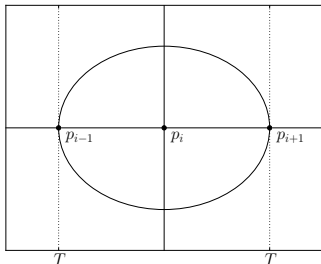
- Minimum Jaccard similarity threshold  $T$

## Generating Data Mines

AnDarwin represents apps as sets

- Minimum Jaccard similarity threshold  $T$

Generate points exactly  $T$ -width apart:



## Generating Data Mines

DBSCAN (Ester et al., KDD'96):

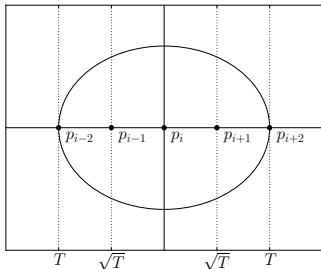
- Core point has  $\geq MinPts$  neighbors in  $T$ -neighborhood
- Clusters form around a core point:
  - Other core points that are at least  $T$  similar to a core point already in the cluster
  - Points in the  $T$ -neighborhood of a core point

## Generating Data Mines

DBSCAN (Ester et al., KDD'96):

- Core point has  $\geq MinPts$  neighbors in  $T$ -neighborhood
- Clusters form around a core point:
  - Other core points that are at least  $T$  similar to a core point already in the cluster
  - Points in the  $T$ -neighborhood of a core point

Generate points to match  $MinPts$ :



## Which Clusters to Merge?

Depends on adversary goals (and, perhaps, budget)

## Which Clusters to Merge?

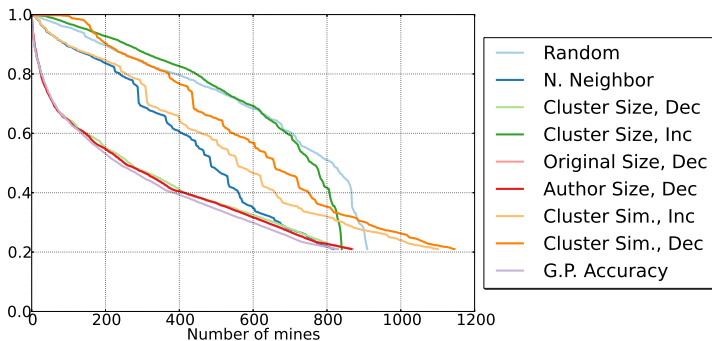
Depends on adversary goals (and, perhaps, budget)

- Maximally degrade plagiarism detection accuracy

## Which Clusters to Merge?

Depends on adversary goals (and, perhaps, budget)

- Maximally degrade plagiarism detection accuracy



Dataset: 273 randomly selected clusters (1,394 apps total)

## Defenses?

Increasing  $T$  and  $MinPts$  may cause us to miss plagiarizing apps



## Defenses?

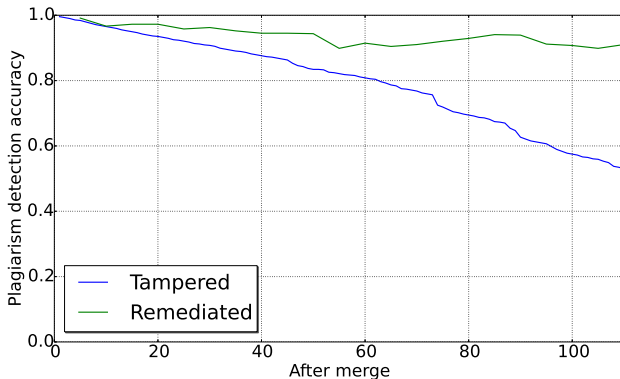
Increasing  $T$  and  $MinPts$  may cause us to miss plagiarizing apps

Instead, can we detect and remove data mines?

## Defenses?

Increasing  $T$  and  $MinPts$  may cause us to miss plagiarizing apps

Instead, can we detect and remove data mines?



# Conclusion

## Contributions:

- Methodology for selecting and then merging arbitrary clusters
- Evaluate effectiveness in a real-world scenario
- Show DBSCAN's vulnerability to the chaining phenomenon
- Propose and evaluate outlier-based remediation

Questions/Comments?

Presenter: Jonathan Crussell  
jcrusse@sandia.gov

This work was supported by the CADA LDRD program at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

## How Many Data Mines?

As a function of the  $T$ :

$$UBAC(T) = \frac{1+T}{1-T} - 1$$

## How Many Data Mines?

As a function of the  $T$ :

$$UBAC(T) = \frac{1 + T}{1 - T} - 1$$

As a function of  $T$  and  $MinPts$ :

$$UBAC(T, MinPts) = \frac{1 + \frac{MinPts-1}{2}\sqrt{T}}{1 - \frac{MinPts-1}{2}\sqrt{T}} - 1$$