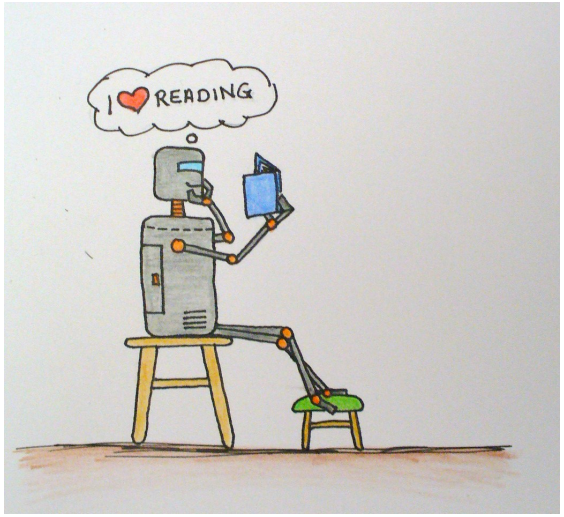


FeatureSmith

Learning to Detect Malware by Mining the Security Literature

Security and Machine Learning

Used for detecting spam, phishing, malware, network attacks, malicious domains, vulnerability exploits in the wild, compromised Web sites, …

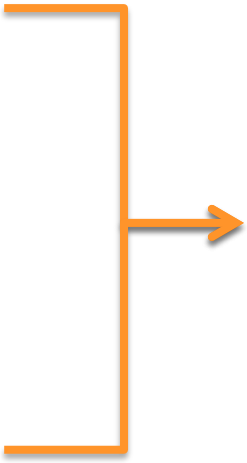
Malicious

Benign

What does it *mean* for two samples to be similar?

Features in Machine Learning Models

- How should we compare samples?
  - Spam: keywords, features from email header, …
  - AI bots: vocabulary, sentence structure, …



Feature engineering

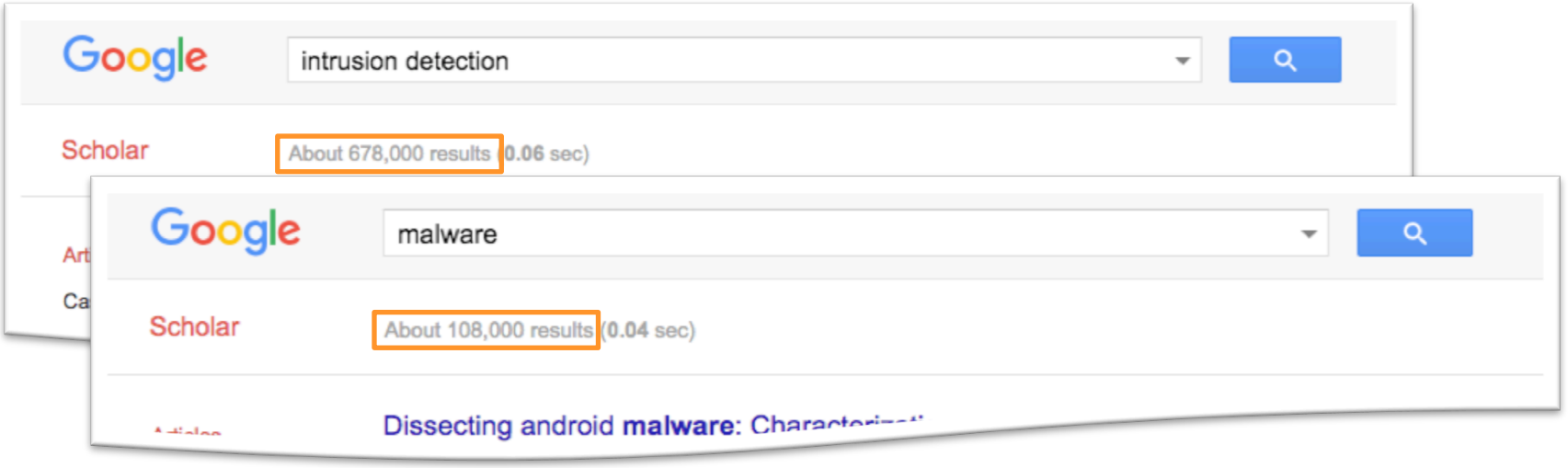
Running Example: Android Malware Detection

- How should we compare samples?
  - Permissions
    - Protect sensitive data and functionality
    - Does not work for privilege escalation
  - API method calls
    - Reveal malware behaviors

- **Feature engineering**
  - Use domain knowledge to identify useful features
  - Must consider **threat semantics**

The Security Body of Knowledge

- Growing volume of papers, industry reports, blogs, …



Difficult to assimilate all relevant knowledge

Dilemma

Growing body of knowledge

VS.

Need for good features

Can we engineer features **automatically**, by mining security papers?

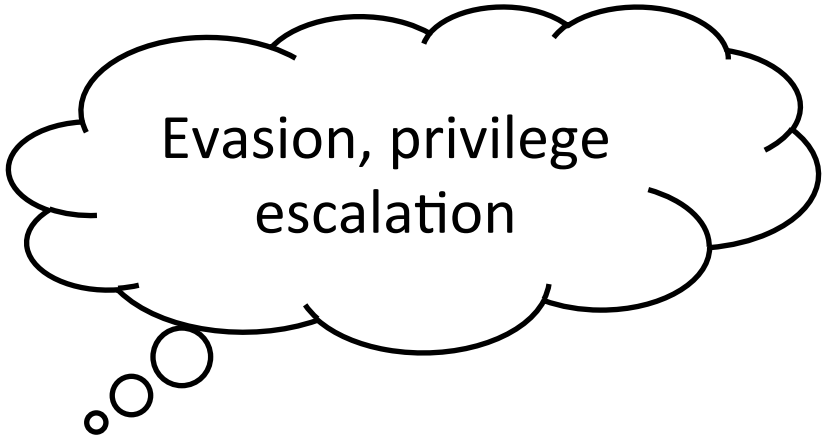
Can we create an artificial intelligence that helps us build other intelligent systems?

Security Threats in Natural Language

- *"The Zsone malware is designed to send SMS messages to certain premium numbers"* *

* Zhou et al. 'Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets,' NDSS 2012.

Security Threats in Natural Language



- "*GingerMaster […] is often bundled with benign applications and tries to gain root access*" *

* Arp et al. 'Drebin: Effective and Explainable Detection of Android Malware in Your Pocket,' NDSS 2014.

Plato's Allegory of the Cave

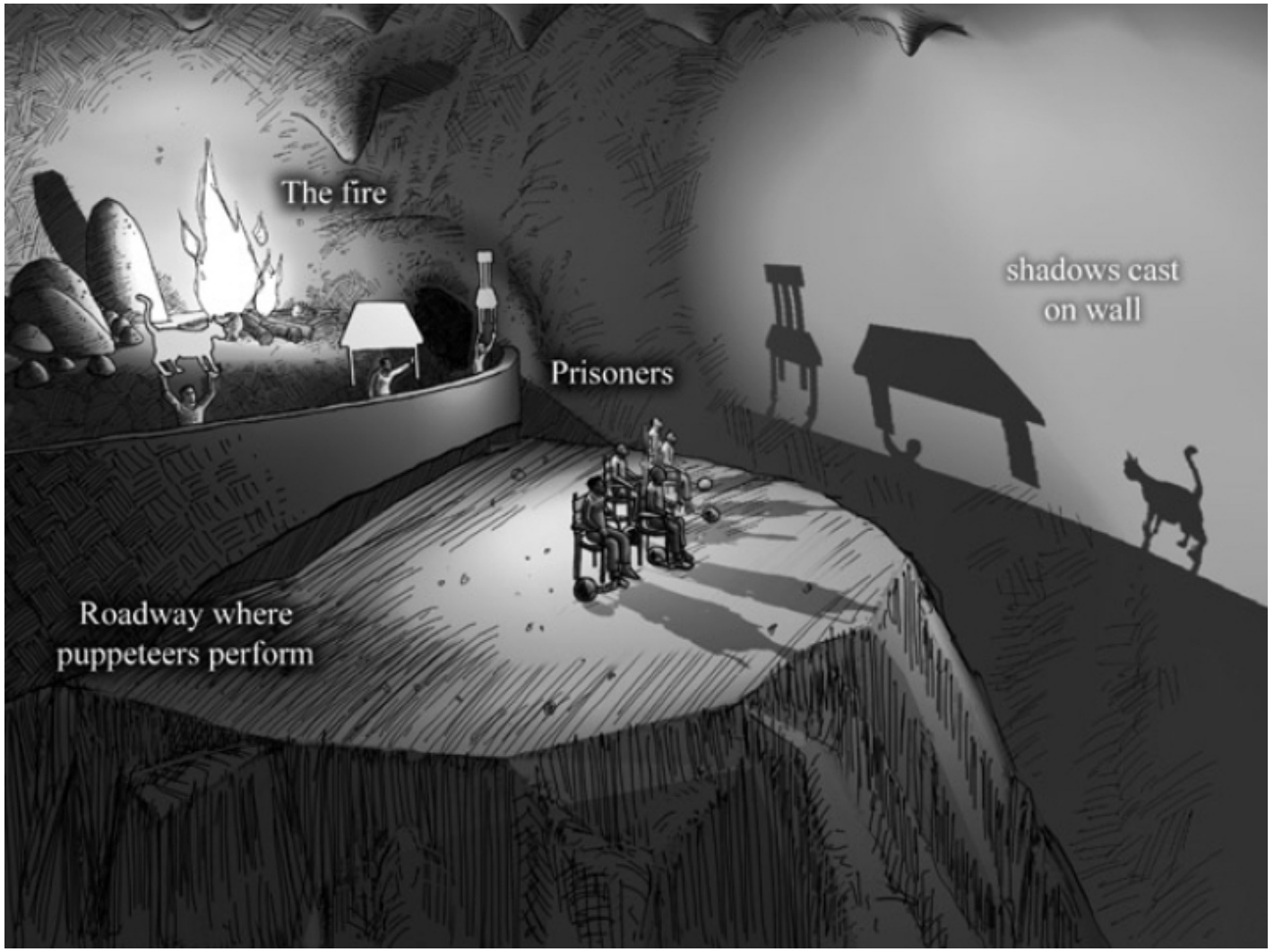
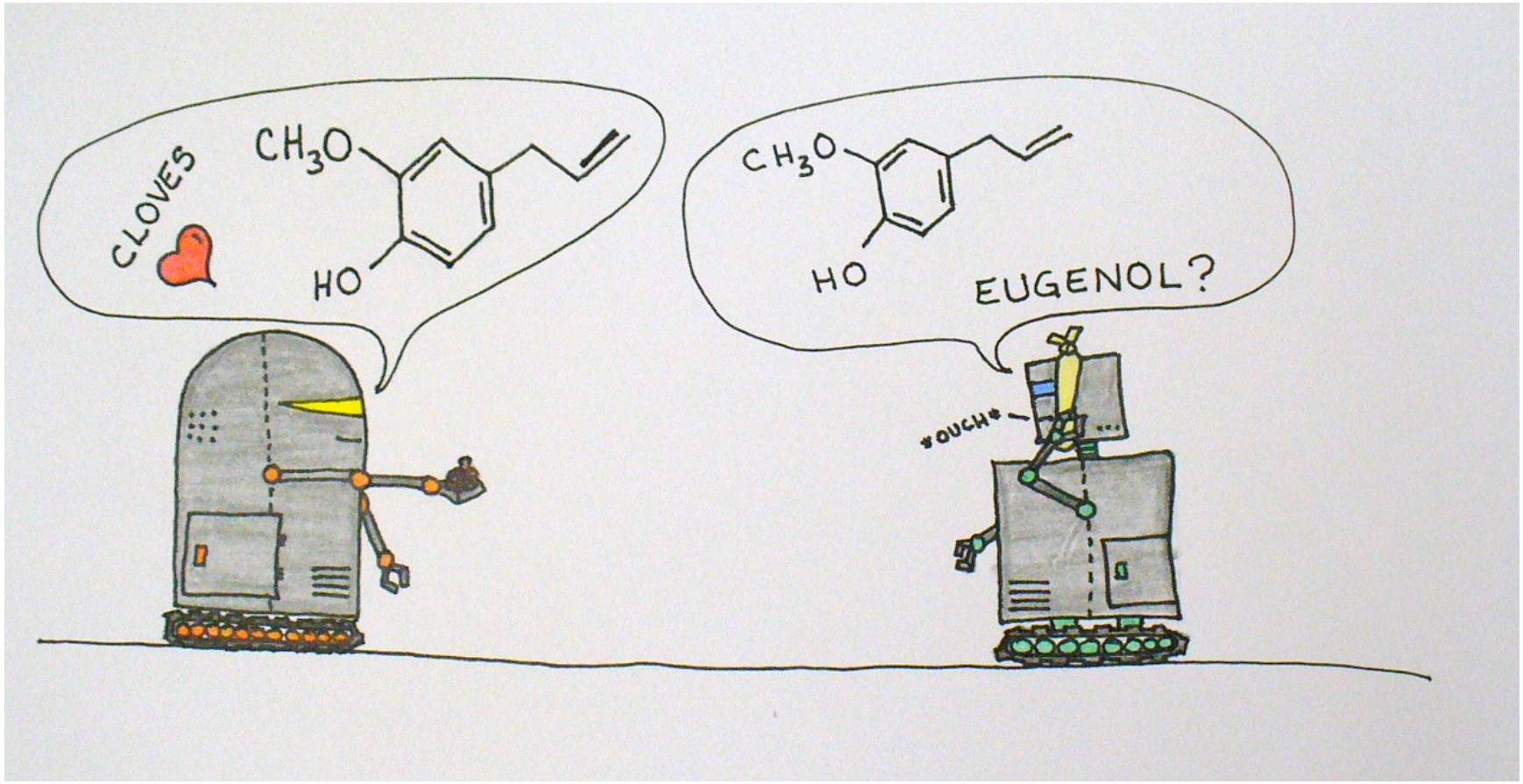


Illustration by John D'Alembert

Domain Knowledge



Challenge #1

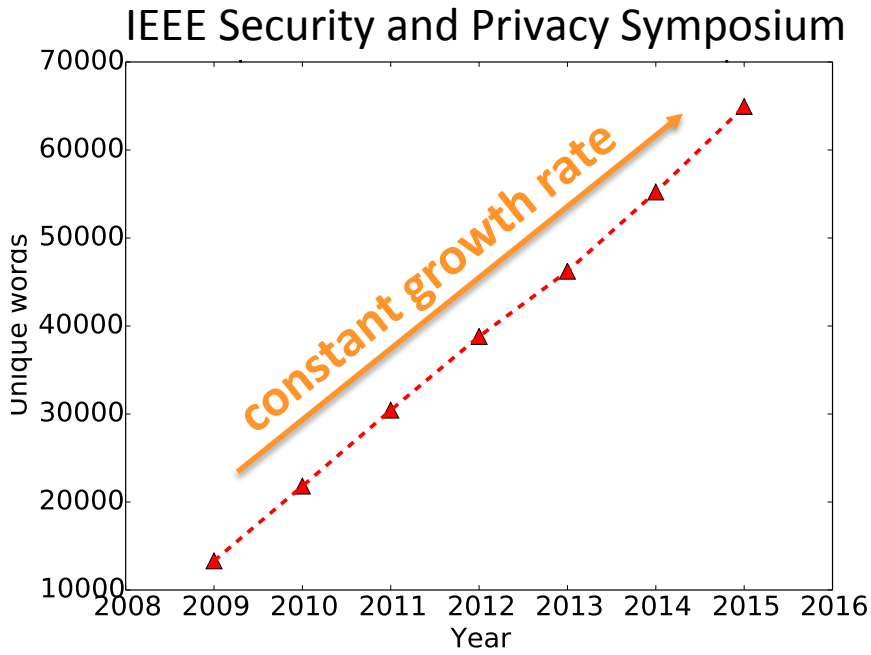
Understanding the **semantic meaning**

- Based on common sense, knowledge of security domain

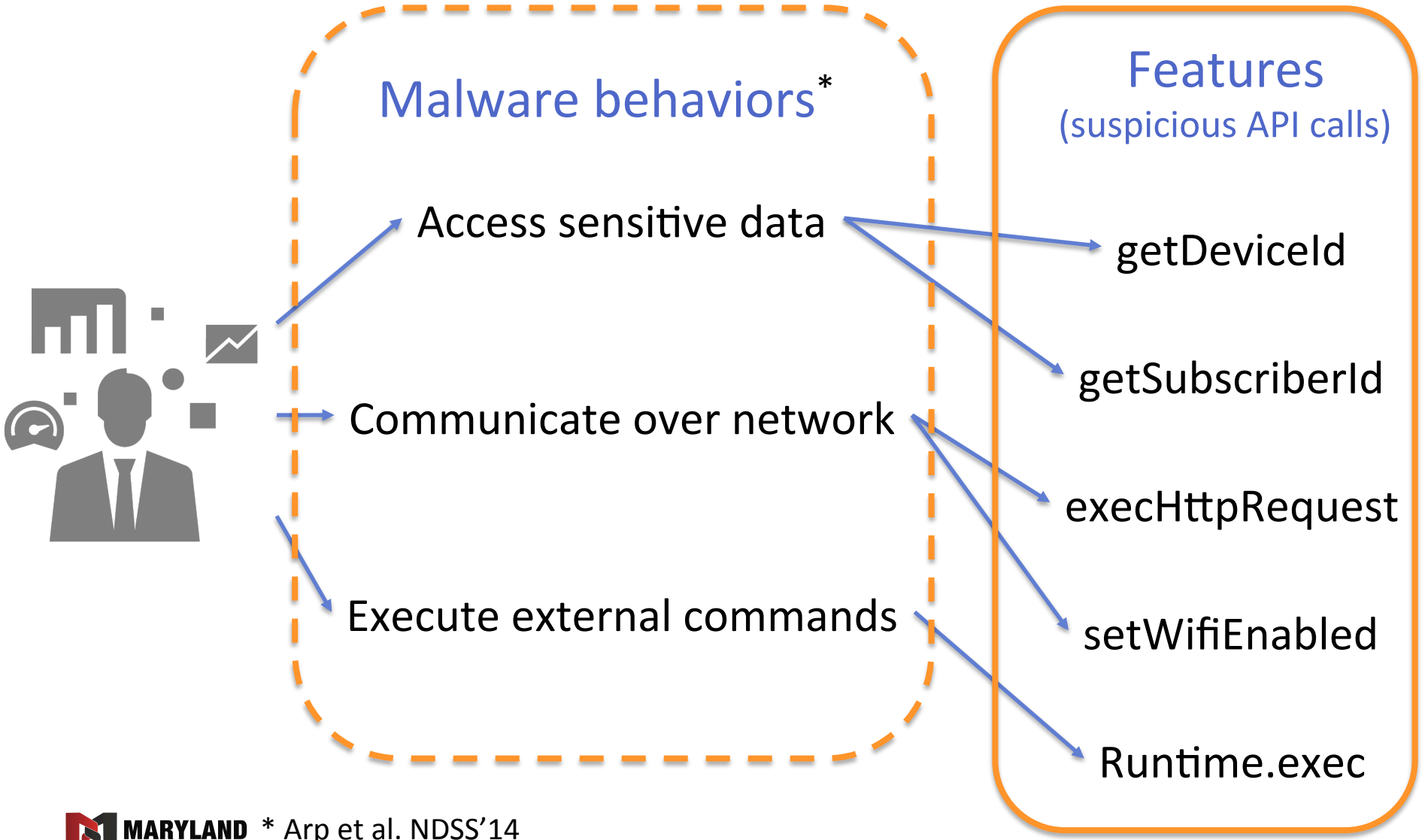
Challenge #2

Attacker behaviors
keep evolving

- Security arms race
- Must discover
  open-ended behaviors



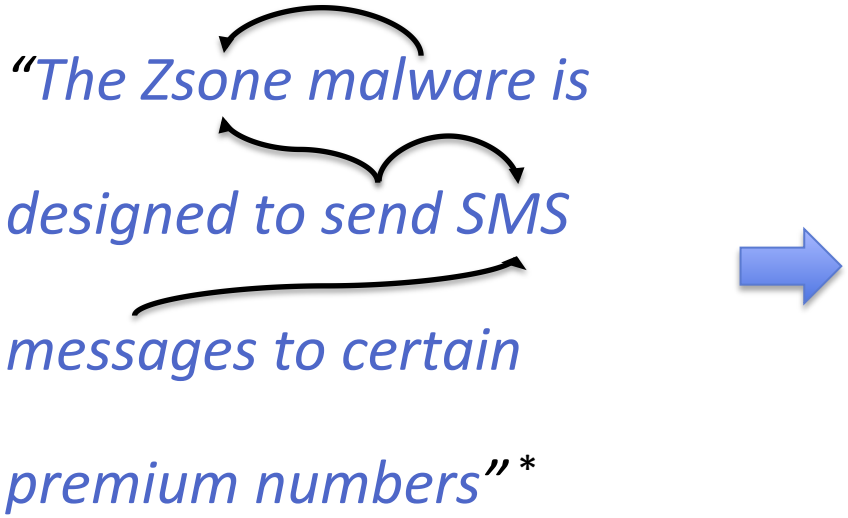
Intuition for Automatic Feature Engineering



* Arp et al. NDSS'14

Behavior Extraction

- Behavior
  - Description of malware activity
  - Short phrase
    - `<subject?, verb, object?>`
    - Parse grammatical structure of sentences



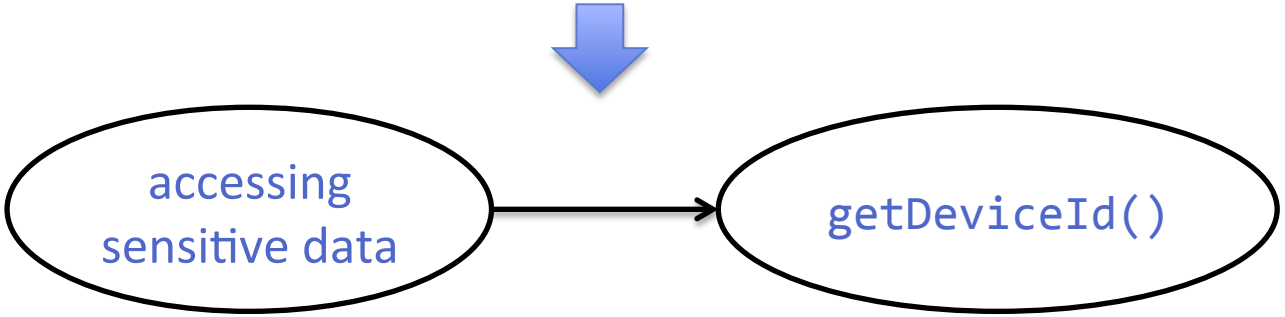
- *Zsone malware send SMS messages*
- *designed Zsone malware*
- *Zsone malware send to certain premium numbers*

* Zhou et al. NDSS'12

Behavior Understanding

- Link behaviors to concrete features

*"API calls for accessing sensitive data, such as `getDeviceId()`"**

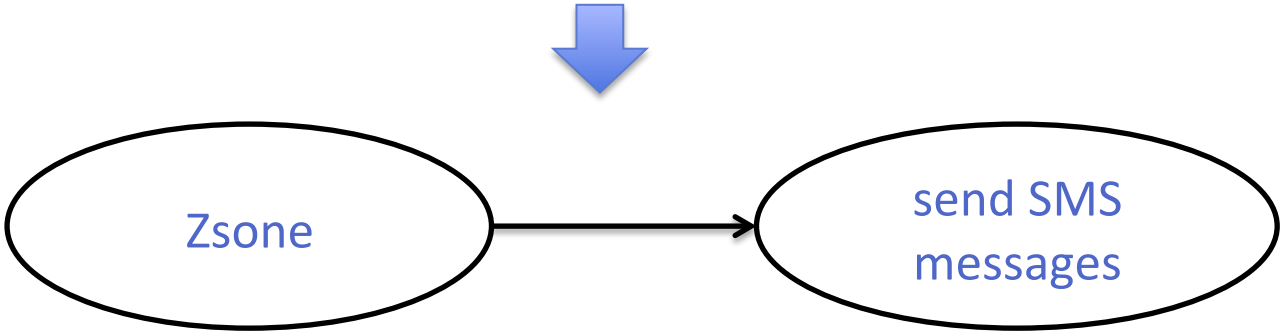


* Arp et al. NDSS'14

Behavior Understanding

- Link behaviors to malware

Zsone malware is designed to send SMS messages to premium numbers



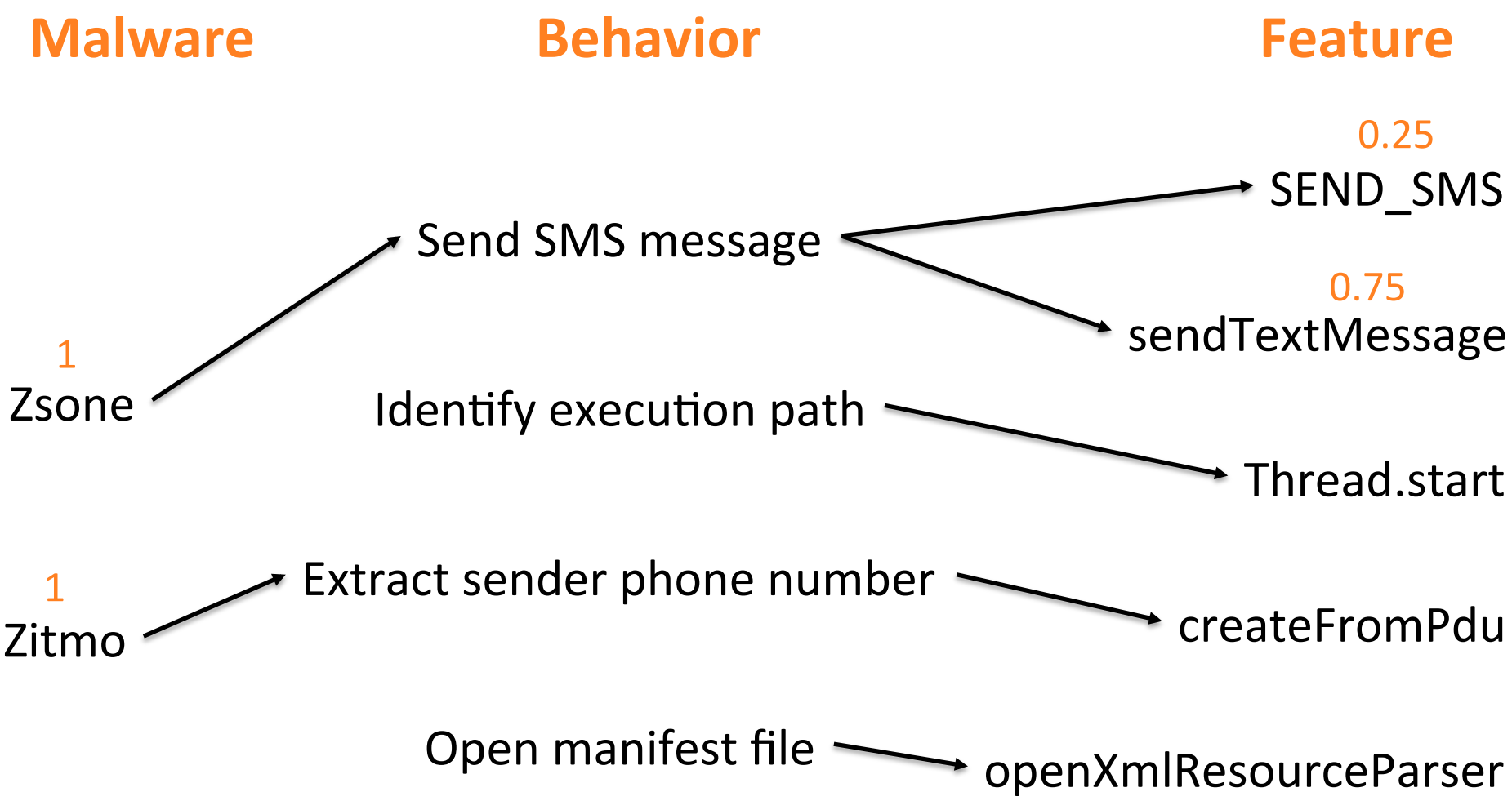
* Zhou et al. NDSS'12

Semantic Network

- Nodes: security concepts
  - Malware families: named entities
  - Concrete features: named entities
  - Behaviors: open ended

- Edges: semantically related concepts
  - Weights based on distance and co-occurrence

Semantic Network Example



How Well Does This Work?

Automatic feature engineering

- **FeatureSmith**
  - Analyzed 1,068 security papers
  - **Automatically** engineered **195** features relevant to Android malware
    - Out of 383 found in the papers

Manual feature engineering

- **Drebin***
  - State-of-the-art Android malware detector
  - Uses **545,334** features
    - Including **315** suspicious API calls, **manually** curated

* Arp et al. NDSS'14

Auto vs. Manual: Experiment

Automatic

- Features engineered by FeatureSmith

Manual

- Features used in Drebin

- Same classification algorithm
- Same corpus of benign and malicious apps
- Same feature types
- Experiment: **Compare the two feature sets**

Auto vs. Manual: Features

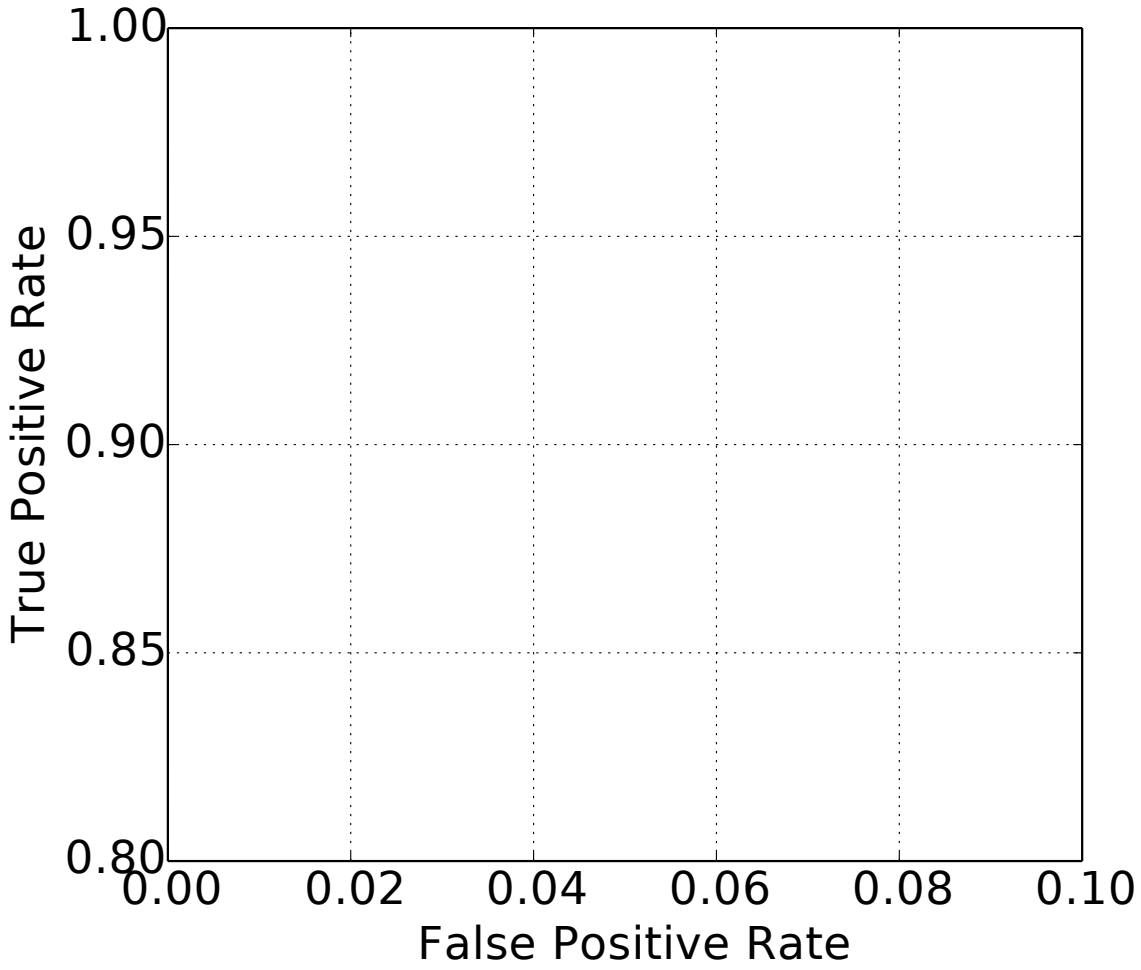
- FeatureSmith discovered new features
  - `getSimOperatorName`
  - `getNetworkOperatorName`
  - `getCountry`

} Missing from manually engineered set

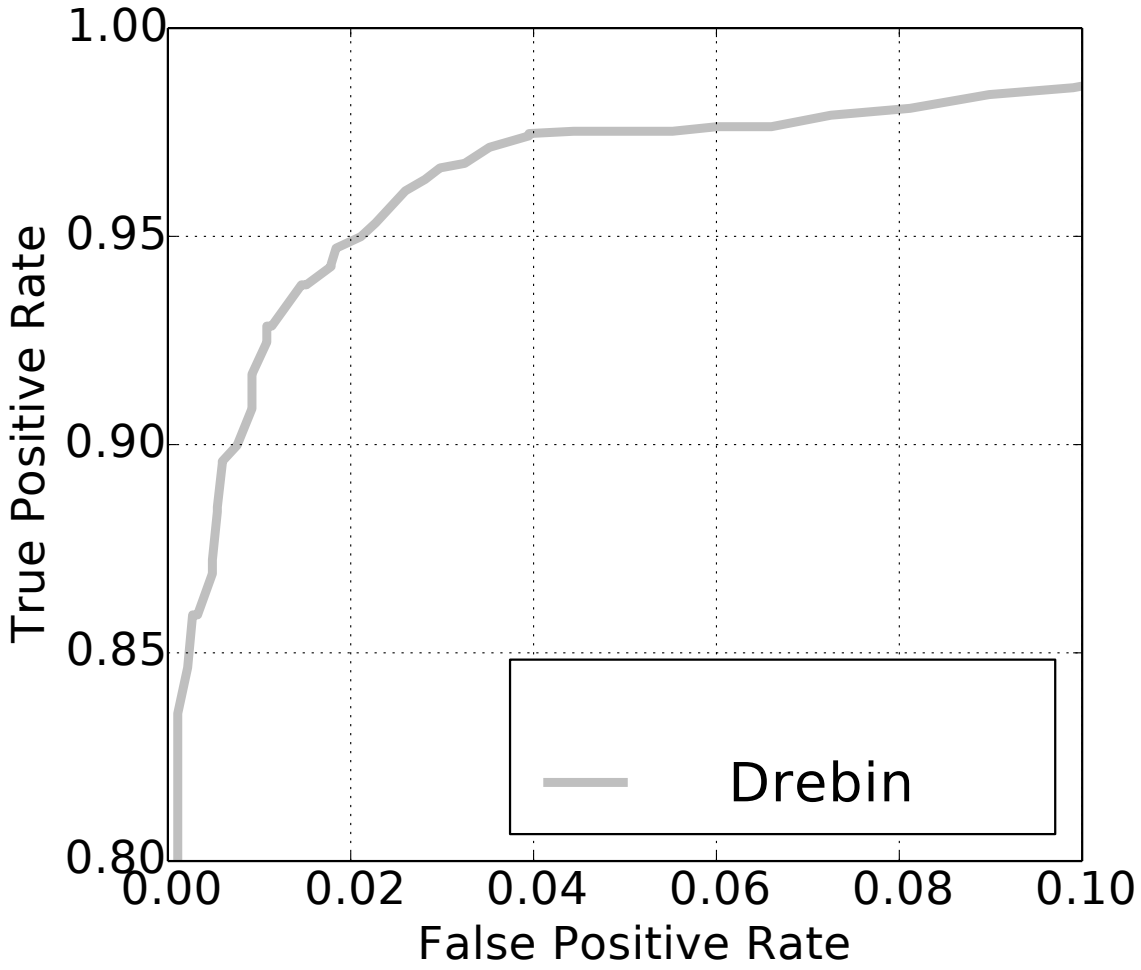
- Often used by malware
  - Help detect Gappusin family (not detected by Drebin)

Human data scientists cannot assimilate all relevant knowledge

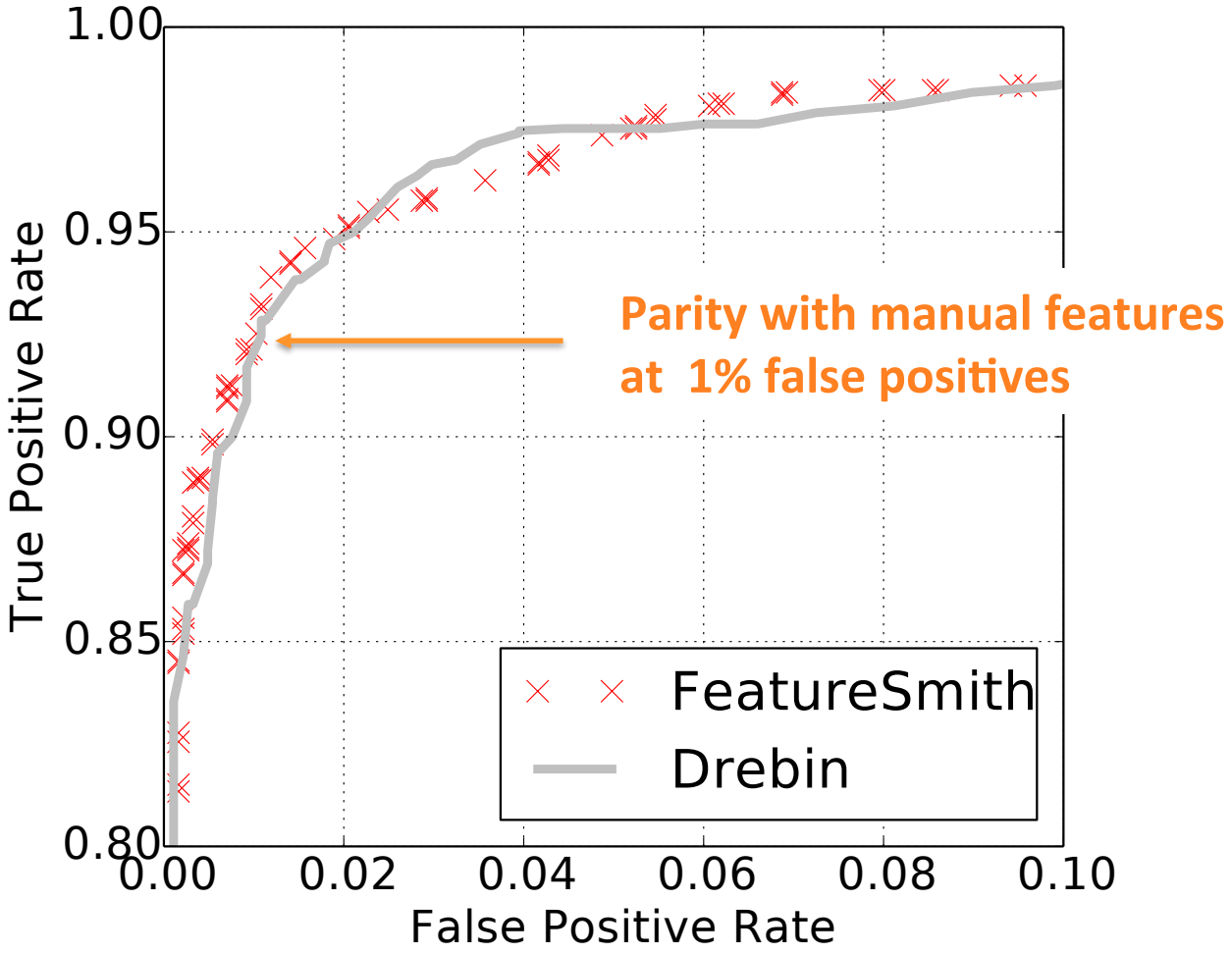
Auto vs. Manual: Detection Performance



Auto vs. Manual: Detection Performance

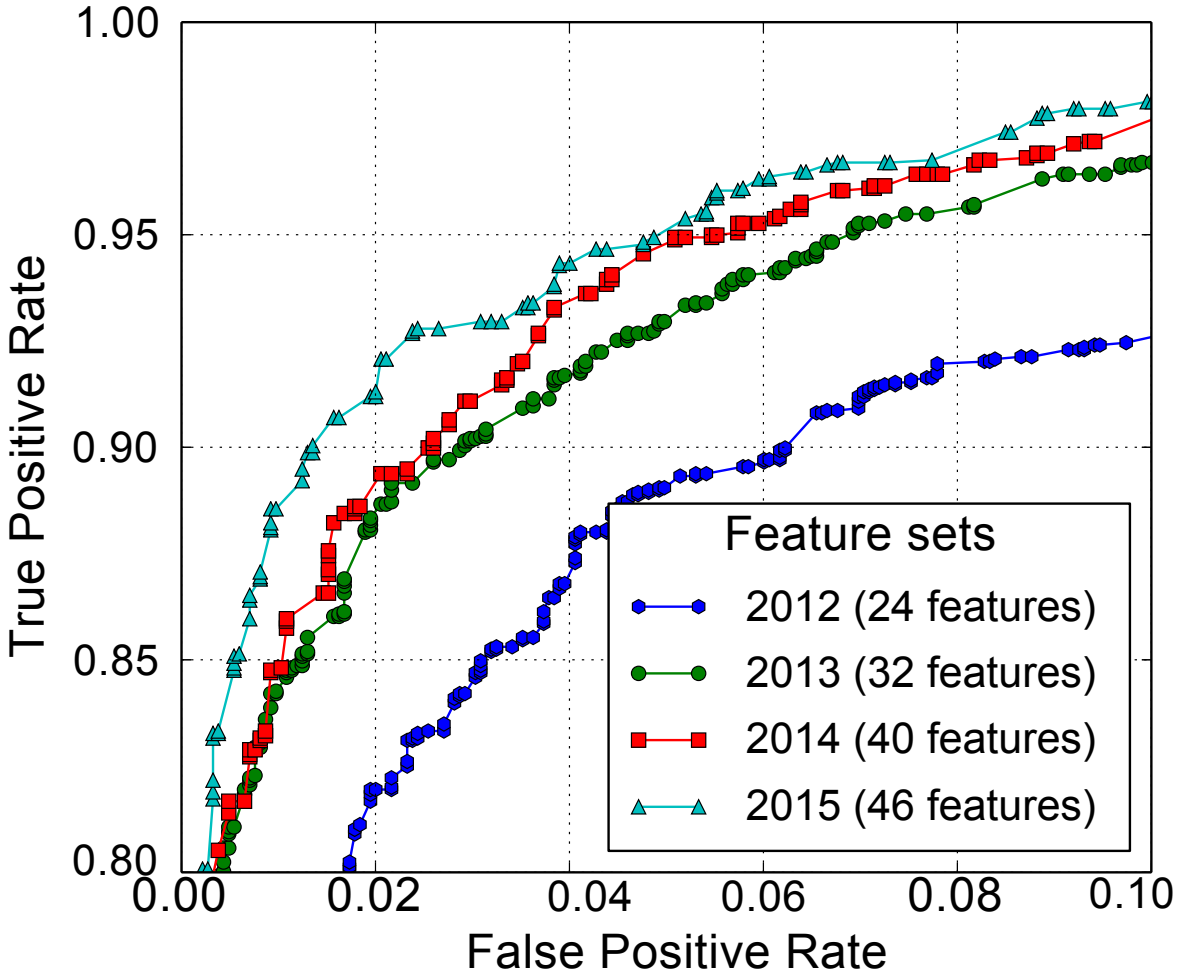


Auto vs. Manual: Detection Performance



Knowledge Evolution

Effectiveness of features discovered in different years



Alternatives

- Feature selection
  - Must enumerate all possible features in advance (e.g. all Android permissions)
- Representation learning
  - Discovers useful features (representations) from raw data (e.g. using a neural network)
- Disadvantages
  - Data-driven: may reflect **biases in the ground truth**
  - No automatic discovery of **threat semantics**

In A Nutshell

- Automatic feature engineering
  - Discover **semantically meaningful features**
    - Some missing from manually curated set
  - Performance **on par** with state-of-the-art malware detector
  - Many potential applications
    - Security: AI bots, threat intelligence, intrusion detection, …
    - Other fields: biomedical research, IBM's Watson Q&A system
- Complements human-driven feature engineering
  - **Human** data scientists have **intuition**
  - **FeatureSmith** can reason over **entire body of knowledge**
- Paper and data: http://featuresmith.org

Automated systems can understand the semantics of security concepts

This is a powerful tool for creating attacks and defenses

Thank you!

Tudor Dumitraș

@tudor_dumitras

http://featuresmith.org

Acknowledgments:

- Work with Ziyun Zhu
- Robot cartoons by Katy Tresedder