FeatureSmith: Automatically Engineering Features for Malware Detection by Mining the Security Literature

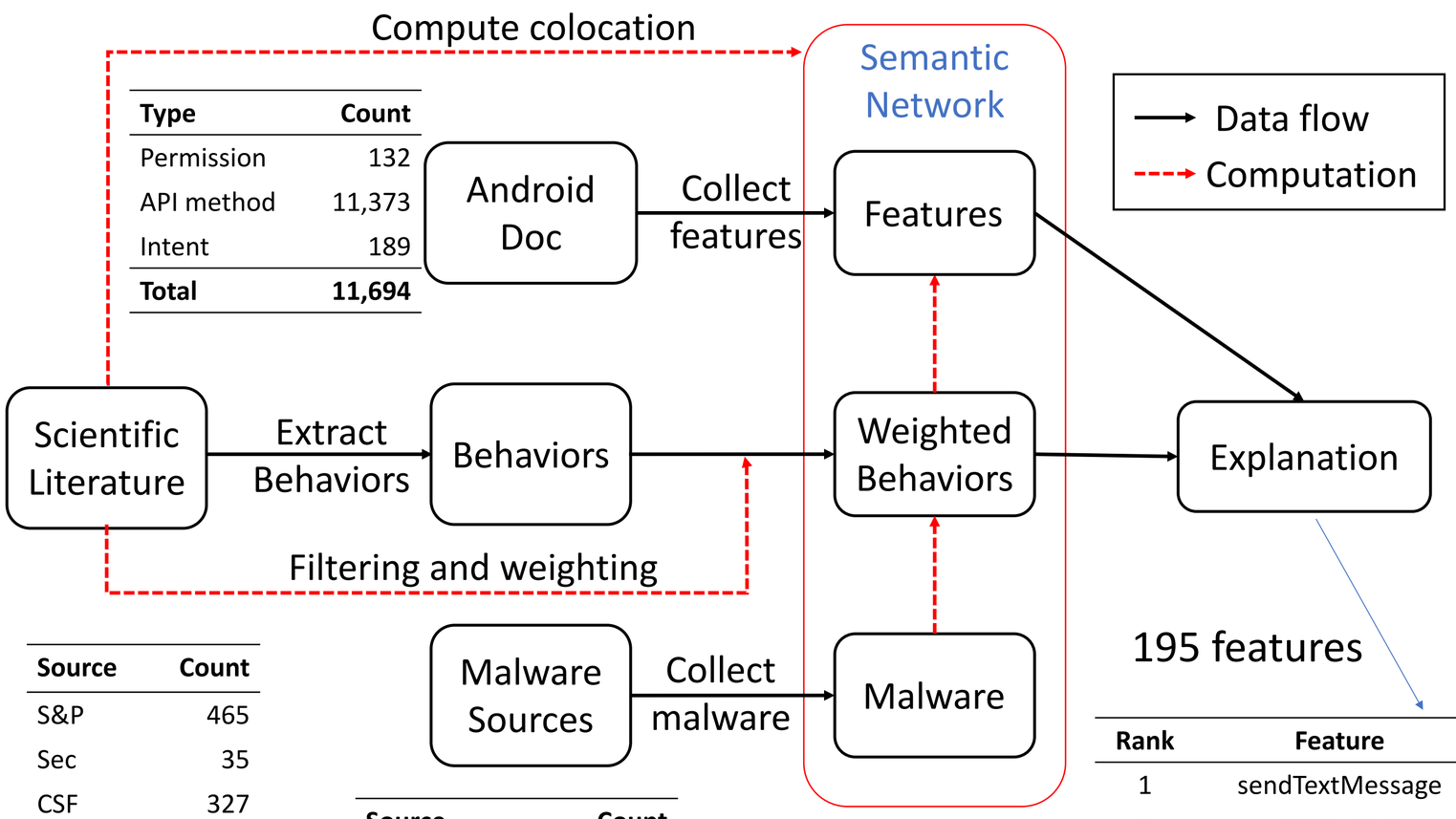
Ziyun Zhu, Tudor Dumitraș
University of Maryland, College Park
{zhuziyun, tdumitra}@umiacs.umd.edu

Malware keeps evolving
- Increasing vulnerabilities
- Different strategy

→ Unstructured data sources
- Security literature
- Malware report

→ Can we engineer features to detect the malware by mining the security literature?

Approach



Type	Count
Permission	132
API method	11,373
Intent	189
Total	**11,694**

Source	Count
S&P	465
Sec	35
CSF	327
Google	241
Total	**1,068**

Source	Count
Drebin	180
Mobile-Sandbox	210
Total	**280**

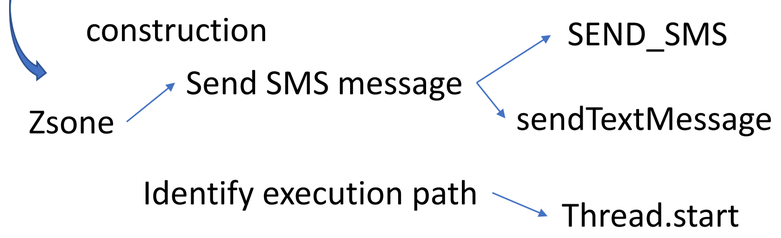
Example

"Zsone malware is designed to send SMS messages to certain premium numbers."

Behavior extraction

Behaviors:
- Zsone malware send SMS message
- Zsone malware send to certain premium number

Semantic network construction



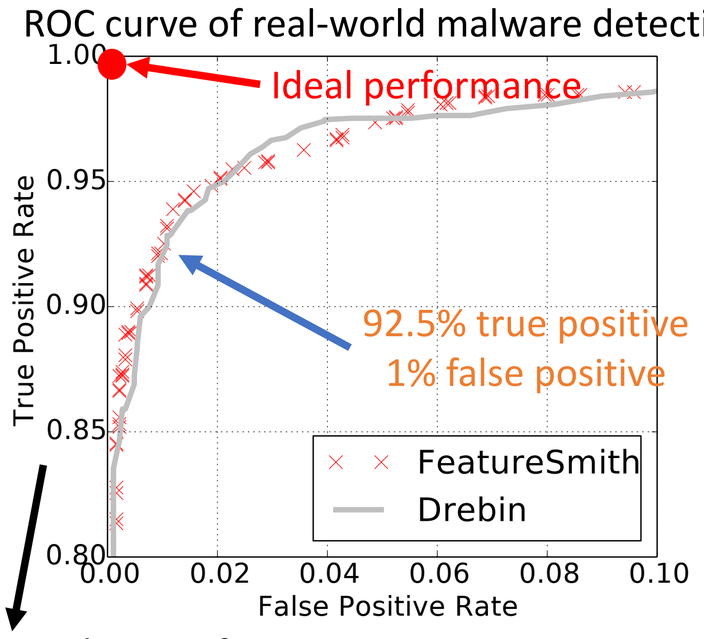
195 features

Rank	Feature	Type	Behaviors
1	sendTextMessage	API method	Subscribe premium-rate service
2	SEND_SMS	Permission	Send SMS message
3	BOOT_COMPLETED	Intent	Register for related system-wide event
4	RECEIVE_SMS	Permission	Block incoming SMS message
5	onStart	API method	Call sendTextMessage method
			.....

Evaluation

➢ Are the features useful?

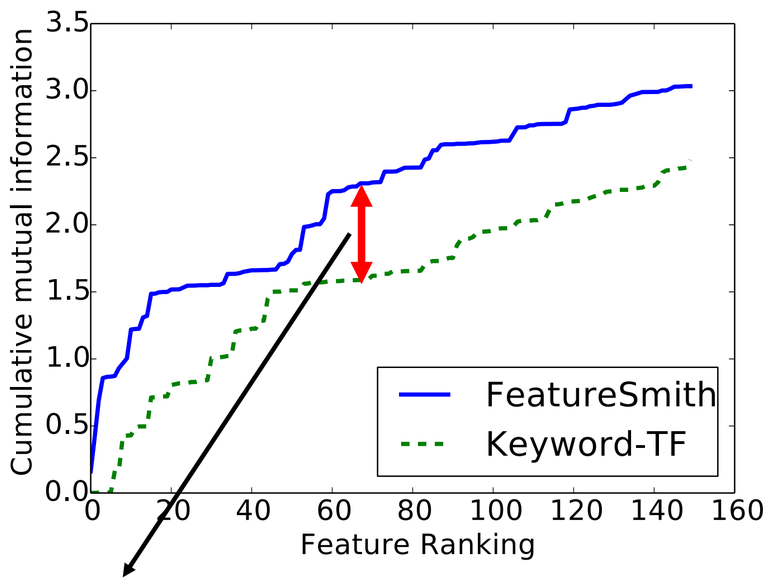
- ROC curve of real-world malware detection



Equivalent performance as the state-of-the-art technique (Drebin)

- 18 "false positives":
  - Mislabeling apps (8)
  - Security apps (2)
    - Intercept phone calls
    - filter messages
  - Parental supervision app (1)
    - Track child's location
  - banking app (1)
  - unknown (6)

➢ How good is the ranking?



FeatureSmith is able to assign high rank to the features with high mutual information.

Feature	MI	Ranking	
		FeatureSmith	Keyword-TF
BOOT_COMPLETED	0.27	3	151
SEND_SMS	0.26	2	9
READ_PHONE_STATE	0.22	11	16
startService	0.18	60	37
RECEIVE_BOOT_COMPLETED	0.17	54	351

Features with highest mutual information also on the top of the list by FeatureSmith

➢ What is the benefit?

✓ Fill the semantic gap

Feature	*getNetworkOperatorName*
Behavior	Send to malicious server Read network operator name
Reference	*"As an example , SUSI identifies as source the unprotected getNetworkOperatorName( ) method in the TelephonyManager class , which returns the name of the network operator or carrier. Our study reveals malware samples that use this method for reading out the network operator name and sending it to a malicious server."* [Rasthofer et al. NDSS 14']

✓ Overlooked features from manual feature engineering

Excluded from Drebin feature set:
- *getSimOperatorName*
- *getNetworkOperatorName*
- *getCountry*

False negatives:
- *Gapussin* (downloader)

Feature Smith Is able to Identify them

✓ Overlooked signatures from data driven selection
- Uncommon patterns
  - createFromPdu, getOriginatingAddress
- Alternative implementations
  - onLocationChanged, onNmeaReceived
- Potential threat
  - isMusicActive

More information: featuresmith.org