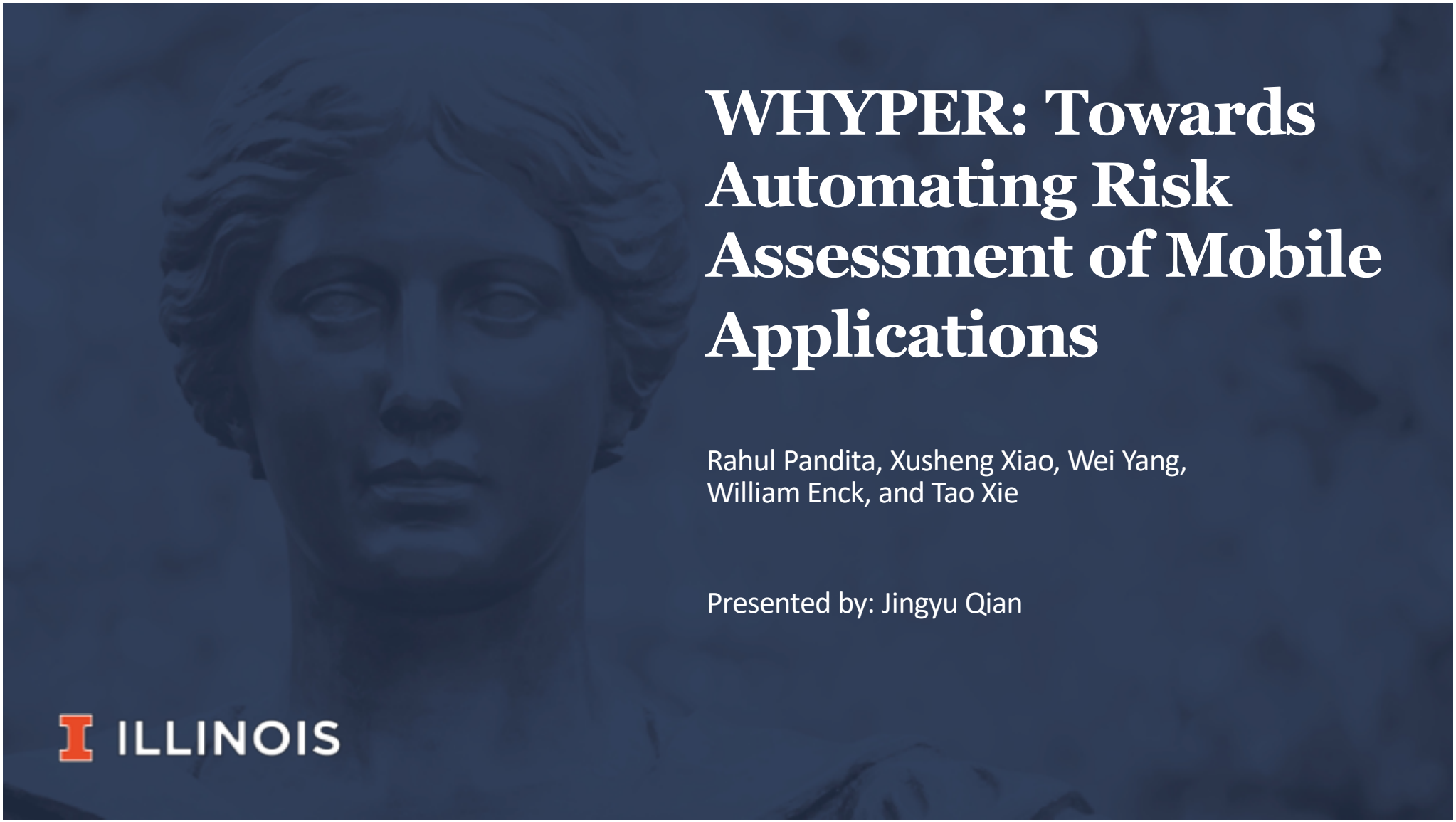


WHYPER: Towards Automating Risk Assessment of Mobile Applications

Rahul Pandita, Xusheng Xiao, Wei Yang,
William Enck, and Tao Xie

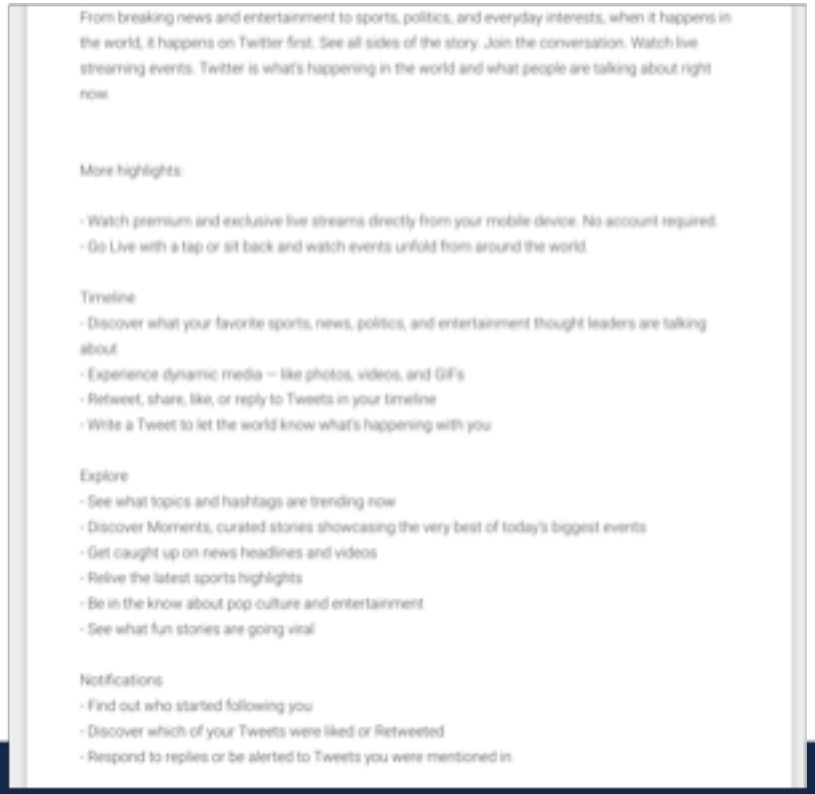
Presented by: Jingyu Qian

ILLINOIS

Apps on Google Play Store

- No manual inspection before publishing an application
- Permission model for security
- How can users know whether the permissions are appropriate?
- What does the user expect?

From Description to Permissions



From breaking news and entertainment to sports, politics, and everyday interests, when it happens in the world, it happens on Twitter first. See all sides of the story. Join the conversation. Watch live streaming events. Twitter is what's happening in the world and what people are talking about right now.

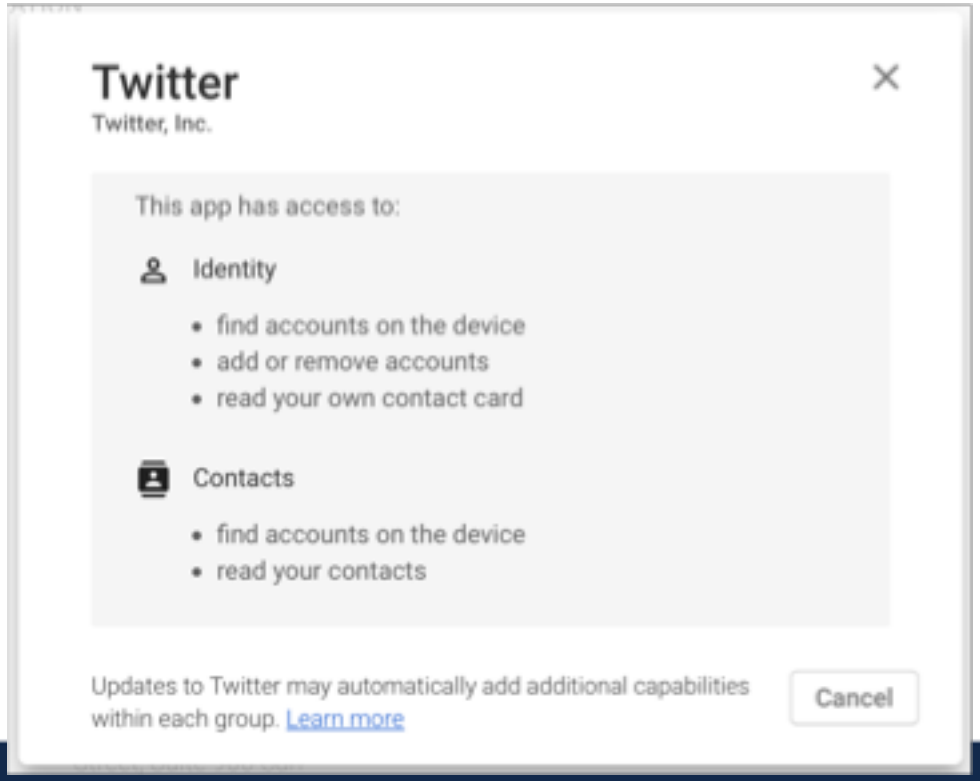
More highlights:

· Watch premium and exclusive live streams directly from your mobile device. No account required.
· Go Live with a tap or sit back and watch events unfold from around the world.

Timeline
· Discover what your favorite sports, news, politics, and entertainment thought leaders are talking about
· Experience dynamic media — like photos, videos, and GIFs
· Retweet, share, like, or reply to Tweets in your timeline
· Write a Tweet to let the world know what's happening with you

Explore
· See what topics and hashtags are trending now
· Discover Moments, curated stories showcasing the very best of today's biggest events
· Get caught up on news headlines and videos
· Relive the latest sports highlights
· Be in the know about pop culture and entertainment
· See what fun stories are going viral

Notifications
· Find out who started following you
· Discover which of your Tweets were liked or Retweeted
· Respond to replies or be alerted to Tweets you were mentioned in



Motivation

- Semantic gap between what the user expects the app to do and what it actually does

- Most users do not understand what permissions mean.

Overview of WHYPER

- Goal: why an application requires a permission
- Infer permission usage from app descriptions
- Natural Language processing (vs. keyword-based searching)
  - Confounding effects
  - Semantic inference

NLP Preliminaries

- Part-of-speech (POS) tagging

```
Share/VB  updates/NNS  and/CC  photos/NNS  ./.
```

- Phrase and clause parsing

```
(ROOT
  (S
    (VP (VB Share)
      (NP (NNS updates)
        (CC and)
        (NNS photos)))
    (. .)))
```

- Typed dependencies

```
root(ROOT-0, Share-1)
dobj(Share-1, updates-2)
cc(updates-2, and-3)
conj(updates-2, photos-4)
```

NLP Preliminaries

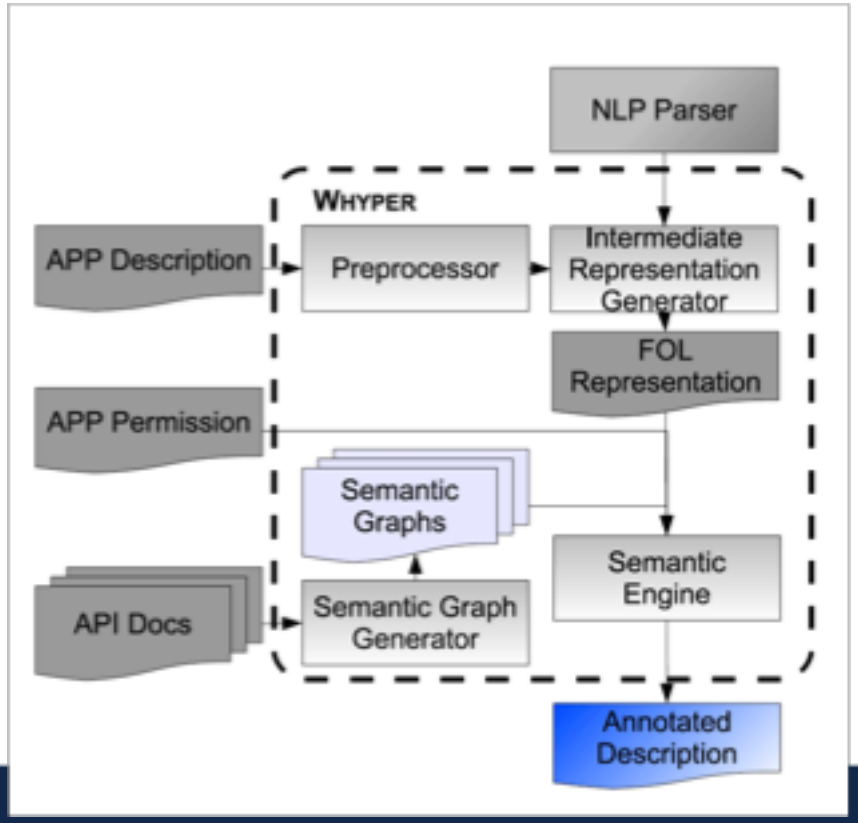
- Named entity recognition (NER)

Navigate your world faster and easier with <ORGANIZATION>Google Maps</ORGANIZATION>.

Threat Model

- Privacy infringements in relatively benign applications
- Identify malware which adds additional permissions
- Do not detect lies in app's description

WHYPER framework



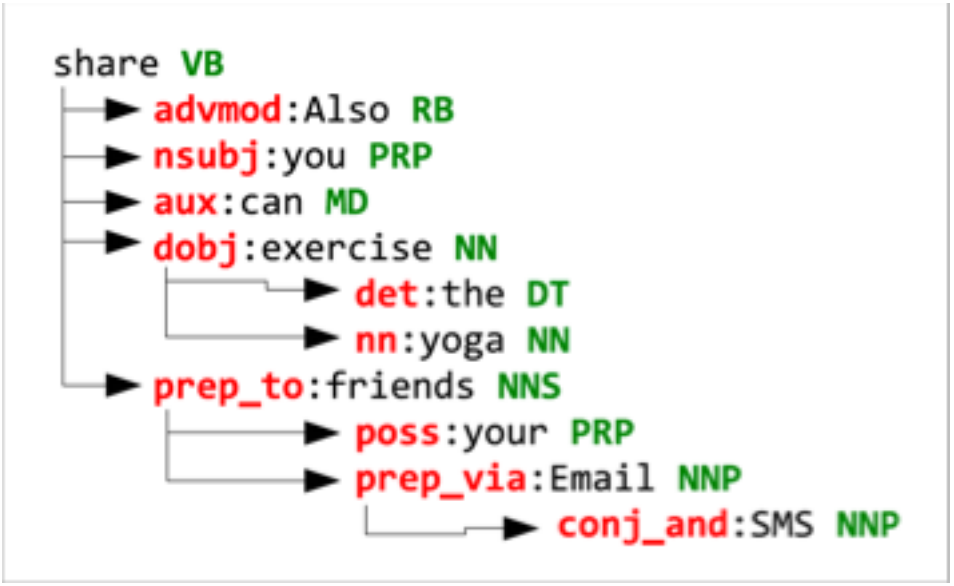
Preprocessing

- Annotate sentence boundaries
  - Period handling (“1.2”, “…”, “Dr. ”)
  - Enumeration list

- Reduce number of lexical tokens
  - Named entity handling (“Google map”)
  - Abbreviation handling (“Instant message (IM)”)

NLP Parser

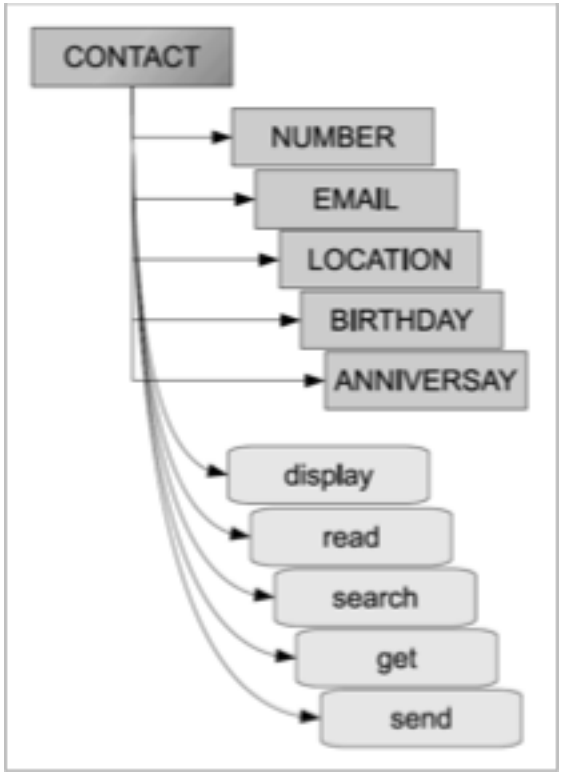
- E.G. “Also you can share the yoga exercise to your friends via Email and SMS. ”



Intermediate-Representation Generator

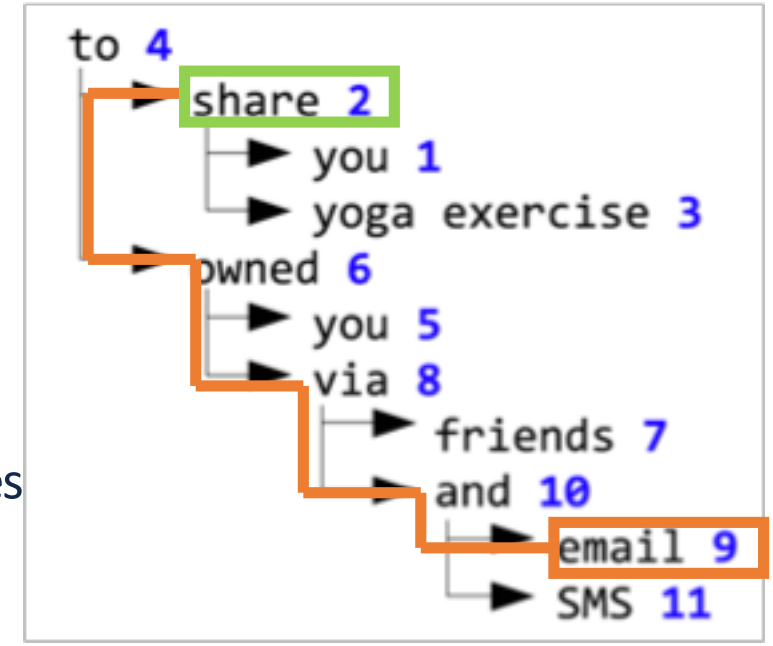
Semantic-Graph Generator

- Infer semantic graph from API documents
- Resource: API name
  - “CONTACTS”, “ADDRESS BOOK”
- Subordinate resources: member variables
  - “EMAIL”, “NUMBER”, …
- Actions: member methods



Semantic Engine

- Locate resource name in the leaf node
- Traverse tree from leaf node to the root
  - Match predicate with actions associated with the resource in the semantic graph
- Recursively search all the subordinate resources
- Deal with synonyms to find matches



Evaluation

- RQ1: What are the precision, recall and F-score of WHYPER
- RQ2: How effective is WHYPER, compared to keyword-based searching?

Table 2: Statistics of Subject permissions

Permission	#N	#S	S_P
READ_CONTACTS	190	3379	235
READ_CALENDAR	191	2752	283
RECORD_AUDIO	200	3822	245
TOTAL	581	9953	763

#N: Number of applications that requests the permission; #S: Total number of sentences in the application descriptions; S_P: Number of sentences manually identified as permission sentences.

Results

Table 3: Evaluation results

Permission	S_I	TP	FP	FN	TN	P (%)	R (%)	F_S (%)	*Acc* (%)
READ_CONTACTS	204	186	18	49	2930	91.2	79.1	84.7	97.9
READ_CALENDAR	288	241	47	42	2422	83.7	85.1	84.4	96.8
RECORD_AUDIO	259	195	64	50	3470	75.9	79.7	77.4	97.0
TOTAL	751	622	129	141	9061	82.8*	81.5*	82.2*	97.3*

* Column average; S_I: Number of sentences identified by WHYPER as permission sentences; TP: Total number of True Positives; FP: Total number of False Positives; FN: Total number of False Negatives; TN: Total number of True Negatives; P: Precision; R: Recall; F_S: F-Score; and *Acc*: Accuracy

Results

- How WHYPER incorrectly identifies a sentence as a permission sentence?
  - Incorrect matching of semantic actions against a resource
  - Incorrect parsing of sentences
- How WHYPER fails to identify a valid permission sentence?
  - Missing matching due to semantic graphs created from API documents (e.g. “blow into” -> “mic”)
  - Incorrect parsing of sentences

WHYPER vs. Keyword Search

- Keyword-based search
  - High FP due to confounding effects, named entities, and lack of semantic context around a keyword
  - Can synonyms help?
- WHYPER
  - Decline in recall

Table 5: Comparison with keyword-based search

Permission	ΔP%	ΔR%	ΔF_S%	ΔAcc%
READ_CONTACTS	50.4	1.3	31.2	7.3
READ_CALENDAR	39.3	1.5	26.4	9.2
RECORD_AUDIO	36.9	-6.6	24.3	6.8
Average	41.6	-1.2	27.2	7.7

Discussion

- Other permissions
- Semantic graphs from API document
- Automatic semantic graph generation

Questions?