Tutorial: Text Analytics for Security

Tao Xie
University of Illinois at Urbana-Champaign
http://taoxie.cs.illinois.edu/
taoxie@illinois.edu

William Enck
North Carolina State University
http://www.enck.org
enck@cs.ncsu.edu
What is Computer Security?

“A computer is secure if you can depend on it and its software to behave as you expect.”
User Expectations

• User expectations are a form of context.

• Other forms of context for security decisions
  – Temporal context (e.g., time of day)
  – Environmental context (e.g., location)
  – Execution context
    • OS level (e.g., UID, arguments)
    • Program analysis level (e.g., control flow, data flow)
Defining User Expectations

• User expectations are difficult to formally (and even informally) define.
  – Based on an individual’s perception the results from past experiences and education
  – ... so, we can’t be perfect

• Starting place: look at the user interface
Why Text Analytics?

• User interface consists of graphics and text
  – End users: includes finding, installing, and running the software (e.g., first run vs. subsequent)
  – Developers: includes API documentation, comments in code, and requirements documents

• Goal: process natural language textual sources to aid security decisions
Outline

• Introduction
• Background on text analytics
• Case Study 1: App Markets
• Case Study 2: ACP Rules
• Wrap-up
Challenges in Analyzing NL Data

• Unstructured
  – Hard to parse, sometimes wrong grammar

• Ambiguous: often has no defined or precise semantics (as opposed to source code)
  – Hard to understand

• Many ways to represent similar concepts
  – Hard to extract information from

/* We need to acquire the write IRQ lock before calling ep_unlink(). */
/* Lock must be acquired on entry to this function. */
/* Caller must hold instance lock! */
Why Analyzing NL Data is Easy(?)

• Redundant data
• Easy to get “good” results for simple tasks
  – Simple algorithms without much tuning effort
• Evolution/version history readily available
• Many techniques to borrow from text analytics: NLP, Machine Learning (ML), Information Retrieval (IR), etc.
Why Analyzing NL Data is Hard(?)

• Domain specific words/phrases, and meanings
  – “Call a function” vs. call a friend
  – “Computer memory” vs. human memory
  – “This method also returns false if path is null”

• Poor quality of text
  – Inconsistent
  – grammar mistakes
    • “true if path is an absolute path; otherwise false” for the File class in .NET framework
  – Incomplete information
Some Major NLP/Text Analytics Tools

Text Miner

Stanford Parser
http://nlp.stanford.edu/software/lex-parser.shtml

Text Analytics for Surveys

http://nlp.stanford.edu/links/statnlp.html
http://www.kdnuggets.com/software/text.html
Dimensions in Text Analytics

• Three major dimensions of text analytics:
  – Representations
    • ...from words to partial/full parsing
  – Techniques
    • ...from manual work to learning
  – Tasks
    • ...from search, over (un-)supervised learning, summarization, ...
Major Text Representations

- Words (stop words, stemming)
- Part-of-speech tags
- Chunk parsing (chunking)
- Semantic role labeling
- Vector space model
Words’ Properties

- Relations among word surface forms and their senses:
  - **Homonymy**: same form, but different meaning (e.g. bank: river bank, financial institution)
  - **Polysemy**: same form, related meaning (e.g. bank: blood bank, financial institution)
  - **Synonymy**: different form, same meaning (e.g. singer, vocalist)
  - **Hyponymy**: one word denotes a subclass of another (e.g. breakfast, meal)

- General thesaurus: WordNet, existing in many other languages (e.g. EuroWordNet)
  - [http://wordnet.princeton.edu/](http://wordnet.princeton.edu/)
Stop Words

• Stop words are words that from non-linguistic view do not carry information
  — ...they have mainly functional role
  — ...usually we remove them to help mining techniques to perform better

• Stop words are language dependent – examples:
  — **English**: A, ABOUT, ABOVE, ACROSS, AFTER, AGAIN, AGAINST, ALL, ALMOST, ALONE, ALONG, ALREADY, ...
Stemming

• Different forms of the same word are usually problematic for text analysis, because they have different spelling and similar meaning (e.g. learns, learned, learning,...)

• Stemming is a process of transforming a word into its stem (normalized form)
  —...stemming provides an inexpensive mechanism to merge
Stemming cont.

- For English is mostly used Porter stemmer at http://www.tartarus.org/~martin/PorterStemmer/
- Example cascade rules used in English Porter stemmer
  - ATIONAL -> ATE  relational -> relate
  - TIONAL -> TION  conditional -> condition
  - ENCI -> ENCE  valenci -> valence
  - ANCI -> ANCE  hesitanci -> hesitance
  - IZER -> IZE  digitizer -> digitize
  - ABLI -> ABLE  conformabli -> conformable
  - ALLI -> AL  radicalli -> radical
  - ENTLI -> ENT  differentli -> different
  - ELI -> E  vileli -> vile
  - OUSLI -> OUS  analogousli -> analogous
Part-of-Speech Tags

• Part-of-speech tags specify word types enabling to differentiate words functions
  – For text analysis, part-of-speech tag is used mainly for “information extraction” where we are interested in e.g., named entities (“noun phrases”)
  – Another possible use is reduction of the vocabulary (features)
    • ...it is known that nouns carry most of the information in text documents

• Part-of-Speech taggers are usually learned on manually tagged data
# Part-of-Speech Table

<table>
<thead>
<tr>
<th>part of speech</th>
<th>function or &quot;job&quot;</th>
<th>example words</th>
<th>example sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Verb</strong></td>
<td>action or state</td>
<td>(to) be, have, do, like, work, sing, can, must</td>
<td>EnglishClub.com is a web site. I like EnglishClub.com.</td>
</tr>
<tr>
<td><strong>Noun</strong></td>
<td>thing or person</td>
<td>pen, dog, work, music, town, London, teacher, John</td>
<td>This is my dog. He lives in my house. We live in London.</td>
</tr>
<tr>
<td><strong>Adjective</strong></td>
<td>describes a noun</td>
<td>a/an, the, 69, some, good, big, red, well, interesting</td>
<td>My dog is big. I like big dogs.</td>
</tr>
<tr>
<td><strong>Adverb</strong></td>
<td>describes a verb, adjective or adverb</td>
<td>quickly, silently, well, badly, very, really</td>
<td>My dog eats quickly. When he is very hungry, he eats really quickly.</td>
</tr>
<tr>
<td><strong>Pronoun</strong></td>
<td>replaces a noun</td>
<td>I, you, he, she, some</td>
<td>Tara is Indian. She is beautiful.</td>
</tr>
<tr>
<td><strong>Preposition</strong></td>
<td>links a noun to another word</td>
<td>to, at, after, on, but</td>
<td>We went to school on Monday.</td>
</tr>
<tr>
<td><strong>Conjunction</strong></td>
<td>joins clauses or sentences or words</td>
<td>and, but, when</td>
<td>I like dogs and I like cats. I like cats and dogs. I like dogs but I don't like cats.</td>
</tr>
<tr>
<td><strong>Interjection</strong></td>
<td>short exclamation, sometimes inserted into a sentence</td>
<td>oh!, ouch!, hi!, well</td>
<td>Ouch! That hurts! Hi! How are you? Well, I don't know.</td>
</tr>
</tbody>
</table>

Part-of-Speech Examples

<table>
<thead>
<tr>
<th>verb</th>
<th>noun</th>
<th>verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop!</td>
<td>John</td>
<td>works.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>noun</th>
<th>verb</th>
<th>verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>is</td>
<td>working.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>pronoun</th>
<th>verb</th>
<th>noun</th>
</tr>
</thead>
<tbody>
<tr>
<td>She</td>
<td>loves</td>
<td>animals.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>noun</th>
<th>verb</th>
<th>adjective</th>
<th>noun</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animals</td>
<td>like</td>
<td>kind</td>
<td>people.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>noun</th>
<th>verb</th>
<th>noun</th>
<th>adverb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tara</td>
<td>speaks</td>
<td>English</td>
<td>well.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>noun</th>
<th>verb</th>
<th>adjective</th>
<th>noun</th>
<th>adverb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tara</td>
<td>speaks</td>
<td>good</td>
<td>English.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>pronoun</th>
<th>verb</th>
<th>preposition</th>
<th>adjective</th>
<th>noun</th>
<th>adverb</th>
</tr>
</thead>
<tbody>
<tr>
<td>She</td>
<td>ran</td>
<td>to</td>
<td>the</td>
<td>station</td>
<td>quickly.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>pron.</th>
<th>verb</th>
<th>adj.</th>
<th>noun</th>
<th>conjunction</th>
<th>pron.</th>
<th>verb</th>
<th>pron.</th>
</tr>
</thead>
<tbody>
<tr>
<td>She</td>
<td>likes</td>
<td>big</td>
<td>snakes</td>
<td>but</td>
<td>I</td>
<td>hate</td>
<td>them.</td>
</tr>
</tbody>
</table>

Here is a sentence that contains every part of speech:

<table>
<thead>
<tr>
<th>interjection</th>
<th>pron.</th>
<th>conj.</th>
<th>adj.</th>
<th>noun</th>
<th>verb</th>
<th>prep.</th>
<th>noun</th>
<th>adverb</th>
</tr>
</thead>
<tbody>
<tr>
<td>Well,</td>
<td>she</td>
<td>and</td>
<td>young</td>
<td>John</td>
<td>walk</td>
<td>to</td>
<td>school</td>
<td>slowly.</td>
</tr>
</tbody>
</table>
# Part of Speech Tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Example</th>
<th>Tag</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>Coordin. Conjunction</td>
<td>and, but, or</td>
<td>SYM</td>
<td>Symbol</td>
<td>+, %, &amp;</td>
</tr>
<tr>
<td>CD</td>
<td>Cardinal number</td>
<td>one, two, three</td>
<td>TO</td>
<td>“to”</td>
<td>to</td>
</tr>
<tr>
<td>DT</td>
<td>Determiner</td>
<td>a, the</td>
<td>UH</td>
<td>Interjection</td>
<td>ah, oops</td>
</tr>
<tr>
<td>EX</td>
<td>Existential ‘there’</td>
<td>there</td>
<td>VB</td>
<td>Verb, base form</td>
<td>eat</td>
</tr>
<tr>
<td>FW</td>
<td>Foreign word</td>
<td>mea culpa</td>
<td>VBD</td>
<td>Verb, past tense</td>
<td>ate</td>
</tr>
<tr>
<td>IN</td>
<td>Preposition/sub-conj</td>
<td>of, in, by</td>
<td>VBG</td>
<td>Verb, gerund</td>
<td>eating</td>
</tr>
<tr>
<td>JJ</td>
<td>Adjective</td>
<td>yellow</td>
<td>VBN</td>
<td>Verb, past participle</td>
<td>eaten</td>
</tr>
<tr>
<td>JJR</td>
<td>Adj., comparative</td>
<td>bigger</td>
<td>VBP</td>
<td>Verb, non-3sg pres</td>
<td>eat</td>
</tr>
<tr>
<td>JJR</td>
<td>Adj., comparative</td>
<td>biggest</td>
<td>VBZ</td>
<td>Verb, 3sg pres</td>
<td>eats</td>
</tr>
<tr>
<td>JJS</td>
<td>Adj., superlative</td>
<td>wildest</td>
<td>WDT</td>
<td>Wh-determiner</td>
<td>which, that</td>
</tr>
<tr>
<td>LS</td>
<td>List item marker</td>
<td>1, 2, One</td>
<td>WP</td>
<td>Wh-pronoun</td>
<td>what, who</td>
</tr>
<tr>
<td>MD</td>
<td>Modal</td>
<td>can, should</td>
<td>WPS</td>
<td>Possessive wh-</td>
<td>whose</td>
</tr>
<tr>
<td>NN</td>
<td>Noun, sing. or mass</td>
<td>llama</td>
<td>WRB</td>
<td>Wh-adverb</td>
<td>how, where</td>
</tr>
<tr>
<td>NNS</td>
<td>Noun, plural</td>
<td>llamas</td>
<td>$</td>
<td>Dollar sign</td>
<td>$</td>
</tr>
<tr>
<td>NNP</td>
<td>Proper noun, singular</td>
<td>IBM</td>
<td>#</td>
<td>Pound sign</td>
<td>#</td>
</tr>
<tr>
<td>NNPS</td>
<td>Proper noun, plural</td>
<td>Carolinas</td>
<td>“</td>
<td>Left quote</td>
<td>( ‘ or “)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>”</td>
<td>Right quote</td>
<td>( ‘ or ”)</td>
</tr>
<tr>
<td>PDT</td>
<td>Predeterminer</td>
<td>all, both</td>
<td>(</td>
<td>Left parenthesis</td>
<td>( [, (, {, &lt;)</td>
</tr>
<tr>
<td>POS</td>
<td>Possessive ending</td>
<td>’s</td>
<td>)</td>
<td>Right parenthesis</td>
<td>) ), ) }, &gt;)</td>
</tr>
<tr>
<td>PP</td>
<td>Personal pronoun</td>
<td>I, you, he</td>
<td>,</td>
<td>Comma</td>
<td>,</td>
</tr>
<tr>
<td>PP$</td>
<td>Possessive pronoun</td>
<td>your, one’s</td>
<td>;</td>
<td>Sentence-final punc</td>
<td>. ! ?</td>
</tr>
<tr>
<td>RB</td>
<td>Adverb</td>
<td>quickly, never</td>
<td>:</td>
<td>Mid-sentence punc</td>
<td>( ; ; ... --)</td>
</tr>
<tr>
<td>RBR</td>
<td>Adverb, comparative</td>
<td>faster</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RBS</td>
<td>Adverb, superlative</td>
<td>fastest</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RP</td>
<td>Particle</td>
<td>up, off</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

http://www2.sis.pitt.edu/~is2420/class-notes/2.pdf
Full Parsing

• Parsing provides maximum structural information per sentence

• Input: a sentence ➔ output: a parse tree

• For most text analysis techniques, the information in parse trees is too complex

• Problems with full parsing:
  – Low accuracy
  – Slow
  – Domain Specific

©M. Grobelnik, D. Mladenic
Chunk Parsing

- Break text up into non-overlapping contiguous subsets of tokens.
  - aka. partial/shallow parsing, light parsing.

- What is it useful for?
  - Entity recognition
    - people, locations, organizations
  - Studying linguistic patterns
    - gave NP
    - gave up NP in NP
    - gave NP NP
    - gave NP to NP
  - Can ignore complex structure when not relevant
Chunk Parsing

Goal: divide a sentence into a sequence of chunks.

• Chunks are non-overlapping regions of a text
  
  [I] saw [a tall man] in [the park]

• Chunks are non-recursive
  – A chunk cannot contain other chunks

• Chunks are non-exhaustive
  – Not all words are included in the chunks

©S. Bird
Chunk Parsing Techniques

• Chunk parsers usually ignore lexical content
• Only need to look at part-of-speech tags

• Techniques for implementing chunk parsing
  – E.g., Regular expression matching
Regular Expression Matching

• Define a regular expression that matches the sequences of tags in a chunk
  – A simple noun phrase chunk regrexp:
    `<DT> ? <JJ> * <NN.?>`

• Chunk all matching subsequences:
  The /DT little /JJ cat /NN sat /VBD on /IN the /DT mat /NN
  [The /DT little /JJ cat /NN] sat /VBD on /IN [the /DT mat /NN]

• If matching subsequences overlap, the first one gets priority

DT: Determinner   JJ: Adjective   NN: Noun, sing, or mass
VBD: Verb, past tense   IN: Prepostion/sub-conj   Verb
Semantic Role Labeling

Giving Semantic Labels to Phrases

• \([\text{AGENT} \text{John}] \text{ broke} \ [\text{THEME} \text{ the window}]\)

• \([\text{THEME} \text{ The window}] \text{ broke}\)

• \([\text{AGENT} \text{Sotheby’s}] \text{ .. offered} \ [\text{RECIPIENT} \text{ the Dorrance heirs}] \ [\text{THEME} \text{ a money-back guarantee}]\)

• \([\text{AGENT} \text{Sotheby’s}] \text{ offered} \ [\text{THEME} \text{ a money-back guarantee}] \text{ to} \ [\text{RECIPIENT} \text{ the Dorrance heirs}]\)

• \([\text{THEME} \text{ a money-back guarantee}] \text{ offered by} \ [\text{AGENT} \text{Sotheby’s}]\)

• \([\text{RECIPIENT} \text{ the Dorrance heirs}] \text{ will} [\text{ARM-NEG not}] \text{ be offered} \ [\text{THEME} \text{ a money-back guarantee}]\)
Semantic Role Labeling Good for Question Answering

Q: What was the name of the first computer system that defeated Kasparov?
A: [PATIENT Kasparov] was defeated by [AGENT Deep Blue] [TIME in 1997].

Q: When was Napoleon defeated?
Look for: [PATIENT Napoleon] [PRED defeat-synset] [ARGM-TMP *ANS*]

More generally:

Kristina hit Scott with a baseball yesterday

- Who hit Scott with a baseball?
- Whom did Kristina hit with a baseball?
- What did Kristina hit Scott with?
- When did Kristina hit Scott with a baseball?
## Typical Semantic Roles

<table>
<thead>
<tr>
<th>Thematic Role</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT</td>
<td>The volitional causer of an event</td>
</tr>
<tr>
<td>EXPERIENCER</td>
<td>The experiencer of an event</td>
</tr>
<tr>
<td>FORCE</td>
<td>The non-volitional causer of the event</td>
</tr>
<tr>
<td>THEME</td>
<td>The participant most directly affected by an event</td>
</tr>
<tr>
<td>RESULT</td>
<td>The end product of an event</td>
</tr>
<tr>
<td>CONTENT</td>
<td>The proposition or content of a propositional event</td>
</tr>
<tr>
<td>INSTRUMENT</td>
<td>An instrument used in an event</td>
</tr>
<tr>
<td>BENEFICIARY</td>
<td>The beneficiary of an event</td>
</tr>
<tr>
<td>SOURCE</td>
<td>The origin of the object of a transfer event</td>
</tr>
<tr>
<td>GOAL</td>
<td>The destination of an object of a transfer event</td>
</tr>
</tbody>
</table>

©S.W. Yih & K. Toutanova
<table>
<thead>
<tr>
<th>Thematic Role</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT</td>
<td><em>The waiter</em> spilled the soup.</td>
</tr>
<tr>
<td>EXPERIENCER</td>
<td><em>John</em> has a headache.</td>
</tr>
<tr>
<td>FORCE</td>
<td><em>The wind</em> blows debris from the mall into our yards.</td>
</tr>
<tr>
<td>THEME</td>
<td>Only after Benjamin Franklin broke <em>the ice</em>...</td>
</tr>
<tr>
<td>RESULT</td>
<td>The French government has built a <em>regulation-size baseball diamond</em>...</td>
</tr>
<tr>
<td>CONTENT</td>
<td>Mona asked “<em>You met Mary Ann at a supermarket?</em>”</td>
</tr>
<tr>
<td>INSTRUMENT</td>
<td>He turned to poaching catfish, stunning them with a <em>shocking device</em>...</td>
</tr>
<tr>
<td>BENEFICIARY</td>
<td>Whenever Ann Callahan makes hotel reservations for <em>her boss</em>...</td>
</tr>
<tr>
<td>SOURCE</td>
<td>I flew in <em>from Boston</em>.</td>
</tr>
<tr>
<td>GOAL</td>
<td>I drove to <em>Portland</em>.</td>
</tr>
</tbody>
</table>
Outline

• Introduction
• Background on text analytics
• Case Study 1: App Markets
• Case Study 2: ACP Rules
• Wrap-up
Case Study: App Markets

• App Markets have played an important role in the popularity of mobile devices
• Provide users with a textual description of each application’s functionality

Apple App Store  Google Play  Microsoft Windows Phone
Current Practice

• Apple: market’s responsibility
  – Apple performs manual inspection

• Google: user’s responsibility
  – Users approve permissions for security/privacy
  – Bouncer (static/dynamic malware analysis)

• Windows Phone: hybrid
  – Permissions / manual inspection
Is Program Analysis Sufficient?

• Previous approaches look at permissions, code, and runtime behaviors
• Caveat: *what does the user expect?*
  – GPS Tracker: record and send location
  – Phone-call Recorder: record audio during call
  – One-Click Root: exploit vulnerability
  – Others are more subtle
Vision

• Goal: *bridge gap between user expectation and app behavior*

• WHYPER is a first step in this direction

• Focus on permission and app descriptions
  – Limited to permissions that protect “user understandable” resources
WHYPER Overview

DEVELOPERS

Application Market

WHYPER

USERS
Use Cases

• Enhance user experience while installing apps
• Functionality disclosure to during application submission to market
• Complementing program analysis to ensure more appropriate justifications
Straw man: Keyword Search

• Confounding effects:
  – Certain keywords such as “contact” have a confounding meaning, e.g.,

  “... displays user contacts, ...” vs “... contact me at abc@xyz.com”

• Semantic Interference:
  – Sentences often describe a sensitive operation such as reading contacts without actually referring to the keyword “contact”, e.g.,

  “share yoga exercises with your friends via email, sms”
Preprocessor

• Period Handling
  – Decimals, ellipsis, shorthand notations (Mr., Dr.)

• Sentence Boundaries
  – Tabs, bullet points, delimiters (:)
  – Symbols (*,-) and enumeration sentence

• Named Entity Handling
  – E.g., “Pandora internet radio”

• Abbreviation Handling
  – E.g., “Instant Message (IM)”
WHYPER Framework
Also you can share the yoga exercise to your friends via Email and SMS.
WHYPER Framework

APP Description

APP Permission

API Docs

WHYPER

Preprocessor

Intermediate Representation Generator

FOL Representation

Semantic Graphs

Semantic Graph Generator

Semantic Engine

Annotated Description

NLP Parser
“Also you can share the yoga exercise to your friends via Email and SMS.”
Semantic-Graph Generator

`ContactsContract.Contacts`

extends `Object`

implements `BaseColumns`, `ContactsContract.ContactNameColumns`, `ContactsContract.ContactOptionsColumns`, `ContactsContract.ContactStatusColumns`, `ContactsContract.ContactsColumns`

java.lang.Object

android.provider.ContactsContract.Contacts

**Class Overview**

Constants for the contacts table, which contains a record per aggregate of raw contacts representing the same person.

**Operations**

**Insert**

A Contact cannot be created explicitly. When a raw contact is inserted, the provider will first try to find a Contact representing the same person. If one is found, the raw contact's `CONTACT_ID` column gets the _ID of the aggregate Contact. If no match is found, the provider automatically inserts a new Contact and puts its _ID into the `CONTACT_ID` column of the newly inserted raw contact.

**Update**

Only certain columns of Contact are modifiable: `TIMES_CONTACTED`, `LAST_TIME_CONTACTED`, `STARRED`, `CUSTOM_RINGTONE`, `SEND_TO_VOICEMAIL`. Changing any of these columns on the Contact also changes them on all constituent raw contacts.

**Delete**

Be careful with deleting Contacts! Deleting an aggregate contact deletes all constituent raw contacts. The corresponding sync adapters will notice the deletions of their respective raw contacts and remove them from their back end storage.

**Query**

- If you need to read an individual contact, consider using `CONTENT_LOOKUP_URI` instead of `CONTENT_URI`.
- If you need to look up a contact by the phone number, use `PhoneLookup.CONTENT_FILTER_URI`, which is optimized for this purpose.
- If you need to look up a contact by partial name, e.g. to produce filter-as-you-type suggestions, use the `CONTENT_FILTER_URI` URI.
Semantic-Graph Generator

• Systematic approach to infer graphs
  – Find related API documents using Pscout [CCS’12]
  – Identify resource associated with permissions from the API class name
    • ContactsContract.Contacts
  – Inspect the member variables and member methods to identify actions and subordinate resources
    • ContactsContract.CommonDataKinds.Email
Evaluation

• Subjects
  – Permissions: READ_CONTACTS, READ_CALENDAR, RECORD_AUDIO
  – 581/600* application descriptions (English only)
  – 9,953 sentences

• Research Questions
  – RQ1: What are the precision, recall, and F-Score of WHYPER in identifying permission sentences?
  – RQ2: How effective is WHYPER in identifying permission sentences, compared to keyword-based searching
<table>
<thead>
<tr>
<th>Permissions</th>
<th>#N</th>
<th>#S</th>
<th>(S_p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ_CONTACTS</td>
<td>190</td>
<td>3,379</td>
<td>235</td>
</tr>
<tr>
<td>READCALENDAR</td>
<td>191</td>
<td>2,752</td>
<td>283</td>
</tr>
<tr>
<td>RECORD_AUDIO</td>
<td>200</td>
<td>3,822</td>
<td>245</td>
</tr>
<tr>
<td>TOTAL</td>
<td>581</td>
<td>9,953</td>
<td>763</td>
</tr>
</tbody>
</table>
Classification

- TP: WHYPER(s) && Manual(s)
- FP: WHYPER(s) && not( Manual(s) )
- TN: not( WHYPER(s) ) && not( Manual(s) )
- FN: not( WHYPER(s) ) && Manual(s)
### RQ1 Results: Effectiveness

- Out of 9,061 sentences, only 129 flagged as FPs
- Among 581 apps, 109 apps (18.8%) contain at least one FP
- Among 581 apps, 86 apps (14.8%) contain at least one FN

<table>
<thead>
<tr>
<th>Permission</th>
<th>$S_1$</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>TN</th>
<th>Prec.</th>
<th>Recall</th>
<th>F-Score</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ_CONTACT</td>
<td>204</td>
<td>186</td>
<td>18</td>
<td>49</td>
<td>2,930</td>
<td>91.2</td>
<td>79.2</td>
<td>84.8</td>
<td>97.9</td>
</tr>
<tr>
<td>READ_CALENDAR</td>
<td>288</td>
<td>241</td>
<td>47</td>
<td>42</td>
<td>2,422</td>
<td>83.7</td>
<td>85.2</td>
<td>84.5</td>
<td>96.8</td>
</tr>
<tr>
<td>RECORD_AUDIO</td>
<td>259</td>
<td>195</td>
<td>64</td>
<td>50</td>
<td>3,470</td>
<td>75.3</td>
<td>79.6</td>
<td>77.4</td>
<td>97.0</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>751</strong></td>
<td><strong>622</strong></td>
<td><strong>129</strong></td>
<td><strong>141</strong></td>
<td><strong>9,061</strong></td>
<td><strong>82.8</strong></td>
<td><strong>81.5</strong></td>
<td><strong>82.2</strong></td>
<td><strong>97.3</strong></td>
</tr>
</tbody>
</table>
## R2 Results: Comparison to Keyword-based search

<table>
<thead>
<tr>
<th>Permission</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ_CONTACTS</td>
<td>contact, data, number, name, email</td>
</tr>
<tr>
<td>READCALENDAR</td>
<td>calendar, event, date, month, day, year</td>
</tr>
<tr>
<td>RECORD_AUDIO</td>
<td>record, audio, voice, capture, microphone</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Permission</th>
<th>Delta Precision</th>
<th>Delta Recall</th>
<th>Delta F-score</th>
<th>Delta Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ_CONTACTS</td>
<td>50.4</td>
<td>1.3</td>
<td>31.2</td>
<td>7.3</td>
</tr>
<tr>
<td>READCALENDAR</td>
<td>39.3</td>
<td>1.5</td>
<td>26.4</td>
<td>9.2</td>
</tr>
<tr>
<td>RECORD_AUDIO</td>
<td>36.9</td>
<td>-6.6</td>
<td>24.3</td>
<td>6.8</td>
</tr>
<tr>
<td>WHYPER Improvement</td>
<td>41.6</td>
<td>-1.2</td>
<td>27.2</td>
<td>7.7</td>
</tr>
</tbody>
</table>
Results Analysis: False Positives

• Incorrect Parsing
  – “MyLink Advanced provides full synchronization of all Microsoft Outlook emails (inbox, sent, outbox and drafts), contacts, calendar, tasks and notes with all Android phones via USB”

• Synonym Analysis
  – “You can now turn recordings into ringtones.”
Results Analysis: False Negatives

• Incorrect parsing
  – Incorrect identification of sentence boundaries and limitations of underlying NLP infrastructure

• Limitations of Semantic Graphs
  – Manual Augmentation
    • Microphone (blow into) and call (record)
    • Significant improvement of delta recalls: -6.6% to 0.6%
  – Future: automatic mining from user comments and forums
Broader Applicability

• Generalization to other permissions
  – User-understandable permissions: calls, SMS
  – Problem areas
    • Location and phone identifiers (widely abused)
    • Internet (nearly every app requires)
Dataset and Paper

• Our code and datasets are available at https://sites.google.com/site/site/whypermission/

Outline

• Introduction
• Background on text analytics
• Case Study 1: App Markets
• Case Study 2: ACP Rules
• Wrap-up
Access Control Policies (ACP)

- Access control is often governed by security policies called Access Control Policies (ACP)
  - Includes rules to control which **principals** have **access** to which **resources**

**ex.**

“The Health Care Personnel (HCP) does not have the ability to edit the patient's account.”

- A policy rule includes four elements
  - Subject – **HCP**
  - Action – **edit**
  - Resource - **patient's account**
  - Effect - **deny**
Access Control Vulnerabilities

Improper access control causes problems (e.g., information exposures)
- Incorrect specification
- Incorrect enforcement

5. Classic buffer overflow
4. Cross-site request forgery
5. Improper access control (Authorization)
6. ...
Problems of ACP Practice

• In practice, ACPs
  – Buried in requirement documents
  – Written in NL and not checkable

• NL documents could be large in size
  – Manual extraction is labor-intensive and tedious
Overview of Text2Policy

Linguistic Analysis

A HCP should not change patient’s account.

An [subject: HCP] should not [action: change] [resource: patient’s account].

Model-Instance Construction

Transformation

```
<Policy PolicyId="ACP0" RuleCombAlgId="deny-overrides">
  <Target/>
  <Rule Effect="Deny" RuleId="rule-1">
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="string-equal">
            <AttrValue>HCP</AttrValue>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch MatchId="string-equal">
            <AttrValue>patient.account</AttrValue>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="string-equal">
            <AttrValue>DataType="string">UPDATE</AttrValue>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
</Policy>
```
Linguistic Analysis

• Incorporate syntactic and semantic analysis
  – **syntactic** structure -> noun group, verb group, etc.
  – **semantic** meaning -> subject, action, resource, negative meaning, etc.

• Provide New techniques for model extraction
  – Identify ACP and AS sentences
  – Infer semantic meaning
Common Techniques

- Shallow parsing
- Domain dictionary
- Anaphora resolution

An HCP can view patient’s account.
He is disallowed to change the patient’s account.
Technical Challenges (TC) in ACP Extraction

- **TC1: Semantic Structure Variance**
  - different ways to specify the same rule
  
  ACP 1: An HCP cannot change patient’s account.
  ACP 2: An HCP is disallowed to change patient’s account.

- **TC2: Negative Meaning Implicitness**
  - verb could have negative meaning

Semantic-Pattern Matching

• Address TC1 Semantic Structure Variance

• Compose pattern based on grammatical function

ex. An HCP is disallowed to change the patient’s account.

passive voice followed by to-infinitive phrase
Negative-Expression Identification

• Address TC2 Negative Meaning Implicitness

• Negative expression
  – “not” in subject:
    ex. No HCP can edit patient’s account.
  – “not” in verb group:
    ex. HCP can not edit patient’s account.
    HCP can never edit patient’s account.

• Negative meaning words in main verb group
  ex. An HCP is disallowed to change the patient’s account.
AS: Syntactic-Pattern Matching

• Syntactic elements
  – Subject, Main verb, Object

• Subject and Object Checking
  – subject is a not a user or object is not a resource

ex. The prescription list should include medication, the name of the doctor. . .

• Filtering negative-meaning sentences
  – Negative sentences tend not to describe ASs
Overview of Text2Policy

A HCP should not change patient’s account.

An [subject: HCP] should not [action: change] [resource: patient’s account].

Linguistic Analysis

Model-Instance Construction

Transformation
ACP Model-Instance Construction

ex. An HCP is *disallowed* to change the patient’s account.

- Identify subject, action, and resource:
  - Subject: HCP
  - Action: change
  - Resource: patient’s account

- Infer effect:
  - Negative Expression: none
  - Negative Verb: disallow
  - Inferred Effect: deny

- Access Control Rule Extraction (ACRE) approach [ACSAC’14] discovers more patterns
  - Able to handle existing, unconstrained NL texts
Evaluation – RQs

• RQ1: How effectively does Text2Policy identify ACP sentences in NL documents?

• RQ2: How effectively does Text2Policy extract ACP rules from ACP sentences?
Evaluation – Subject

• iTrust open source project
  – 448 use-case sentences (37 use cases)
  – preprocessed use cases

• Collected ACP sentences
  – 100 ACP sentences
  – From 17 sources (published papers and websites)

• A module of an IBMApp (financial domain)
  – 25 use cases
RQ1 ACP Sentence Identification

- Apply Text2Policy to identify ACP sentences in iTrust use cases and IBMApp use cases

<table>
<thead>
<tr>
<th>Subjects</th>
<th># Sent.</th>
<th># ACP Sent.</th>
<th># Ident.</th>
<th>FP</th>
<th>FN</th>
<th>Prec</th>
<th>Rec</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>iTrust</td>
<td>448</td>
<td>117</td>
<td>119</td>
<td>16</td>
<td>14</td>
<td>86.6%</td>
<td>88.0%</td>
<td>87.3%</td>
</tr>
<tr>
<td>IBMApp</td>
<td>479</td>
<td>24</td>
<td>23</td>
<td>0</td>
<td>1</td>
<td>100.0%</td>
<td>95.8%</td>
<td>97.9%</td>
</tr>
<tr>
<td>Total</td>
<td>927</td>
<td>141</td>
<td>142</td>
<td>16</td>
<td>15</td>
<td>88.7%</td>
<td>89.4%</td>
<td>89.1%</td>
</tr>
</tbody>
</table>

- Text2Policy effectively identifies ACP sentences with precision and recall more than 88%

- Precision on IBMApp use cases is better
  - proprietary use cases are often of higher quality compared to open-source use cases
Evaluation –
RQ2 Accuracy of Policy Extraction

- Apply Text2Policy to extract ACP rules from ACP sentences

<table>
<thead>
<tr>
<th>Subjects</th>
<th># ACP Sent.</th>
<th># Extracted</th>
<th>Accu.</th>
</tr>
</thead>
<tbody>
<tr>
<td>iTrust</td>
<td>217</td>
<td>187</td>
<td>86.2%</td>
</tr>
<tr>
<td>IBMAApp</td>
<td>24</td>
<td>21</td>
<td>87.5%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>241</strong></td>
<td><strong>208</strong></td>
<td><strong>86.3%</strong></td>
</tr>
</tbody>
</table>

- Text2Policy effectively extracts ACP model instances with accuracy above 86%
Dataset and Paper

• Our datasets are available at https://sites.google.com/site/asergrp/projects/text2policy


Outline

- Introduction
- Background on text analytics
- Case Study 1: App Markets
- Case Study 2: ACP rules
- Wrap-up
Take-away

• Computing systems contain textual data that partially represents expectation context.

• Text analytics and natural language processing offers an opportunity to automatically extract that semantic context
  – Need to be careful in the security domain (e.g., social engineering)
  – But potential for improved security decisions
Future Directions

• Only beginning to study text analytics for security
  – Many sources of natural language text
  – Many unexplored domains
  – Use text analytics in software engineering as inspiration
    • https://sites.google.com/site/text4se/

• Hard problem: to what extent can we formalize “expectation context”?

• Creation of open datasets (annotation is time intensive)

• Apply to real-world problems
Thank you!

Questions?

Tao Xie
University of Illinois at Urbana-Champaign
http://taoxie.cs.illinois.edu/
taoxie@illinois.edu

William Enck
North Carolina State University
http://www.enck.org
enck@cs.ncsu.edu

Acknowledgment: We thank authors of the original slides that some slides from this tutorial were adapted from. The work is supported in part by NSA Science of Security Lablet grants, a Google Faculty Research Award, NSF grants CNS-1513939/1513690, CCF-1434596, CNS-1434582, CNS-1222680, and CNS-1253346.