Catch me, Yes we can! - Pwning Social Engineers using Natural Language Processing Techniques in Real-Time

Myeongsoo Kim[1], Changheon Song[1], Hyeji Kim[1], Deahyun Park[1], Yeeji Kwon[2], Eun Namkung[1], Ian G. Harris[3], Marcel Carlsson[4]

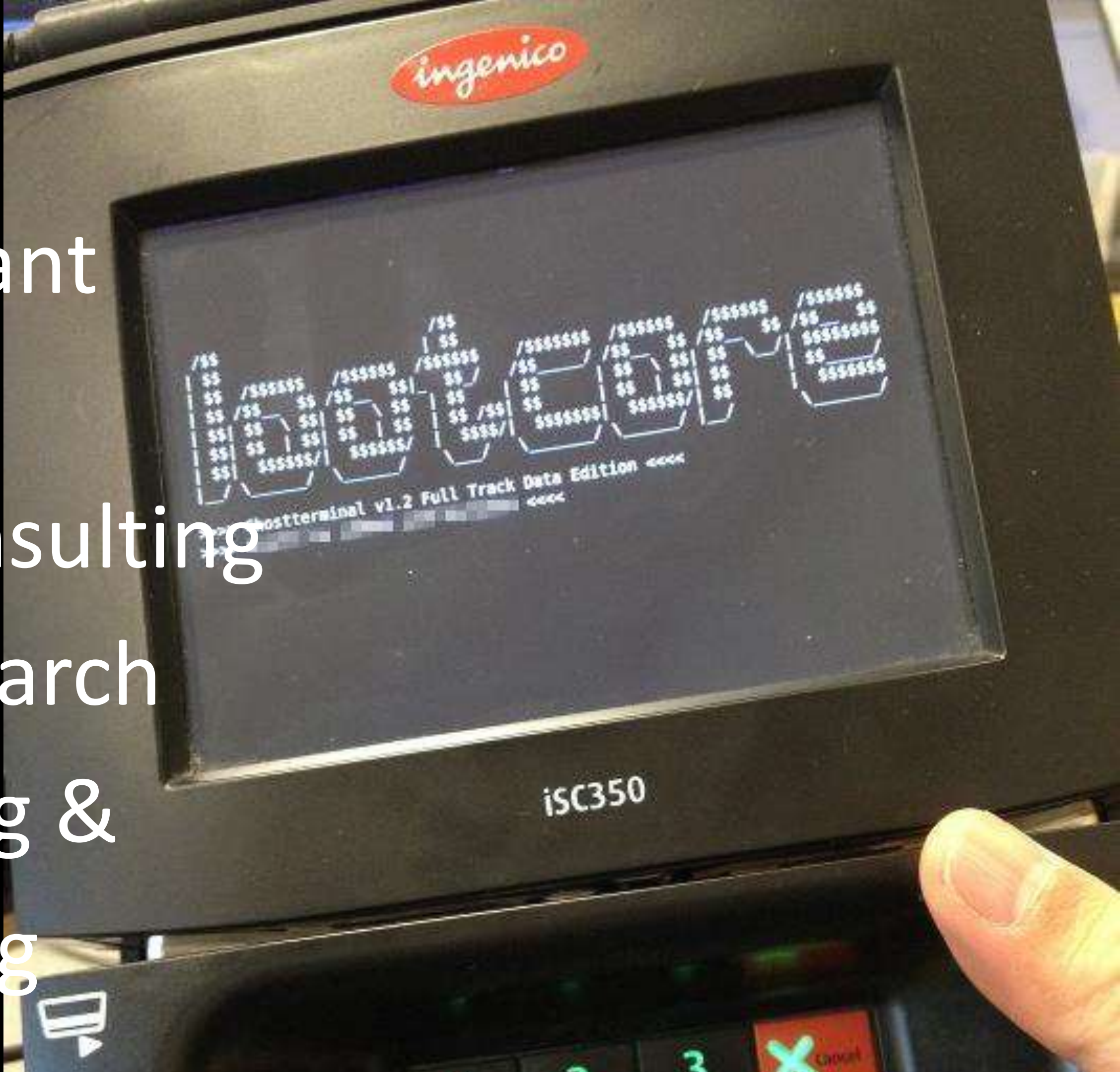
1. Kookmin University, 2. Seoul Women's University, 3. University of California Irvine, 4. Lootcore

Ian G. Harris

- Professor of Computer Science at the University of California Irvine
- Research in HW Verification and Security
- Applies Natural Language Processing techniques

Marcel Carlsson

- Principal consultant Lootcore
- Red teaming, consulting and security research
- Hardware hacking & Social Engineering



Social Engineering (SE) 101

"Any act that influences a person to take an action that may or may not be in their best interest"

– social-engineer.com

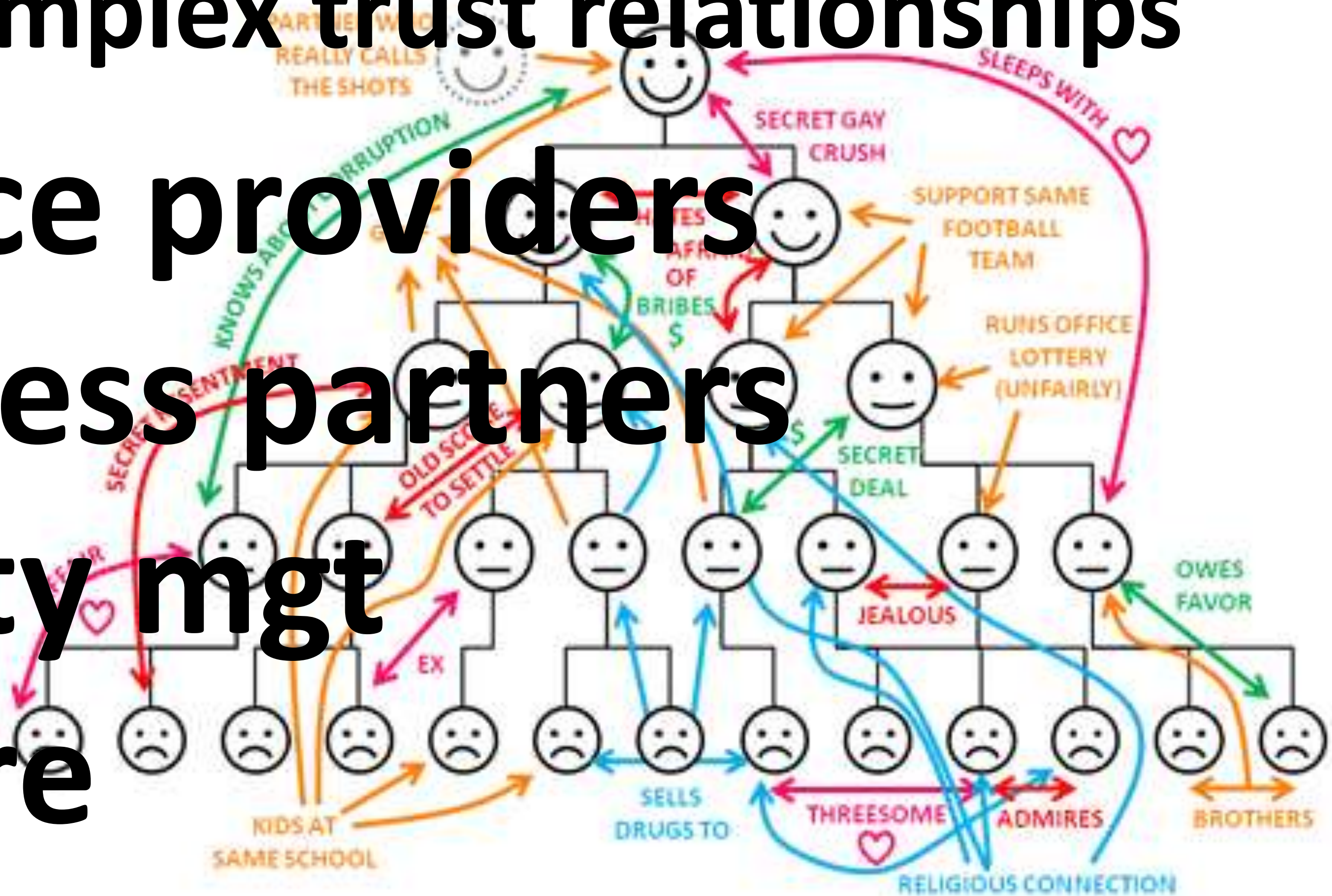
SE == complex concept

SE threat underestimated

SE awareness low

User decision burden

Complex trust relationships



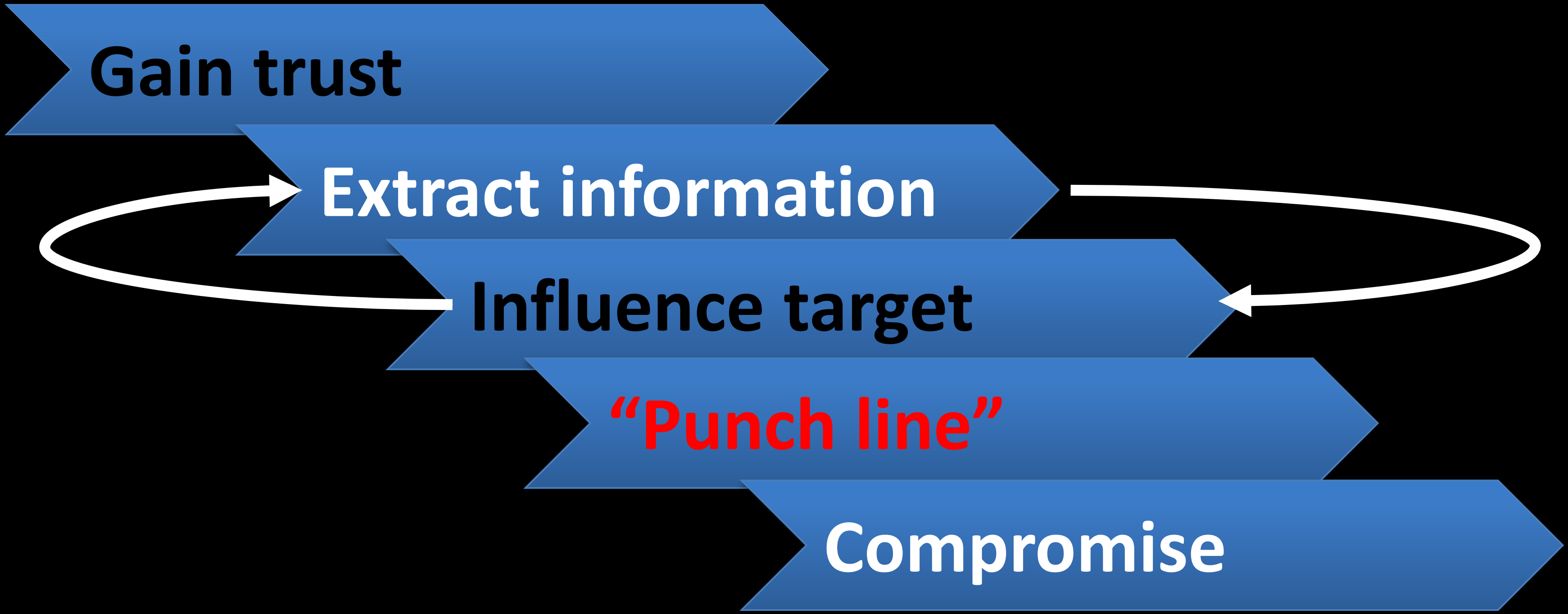
Service providers

Business partners

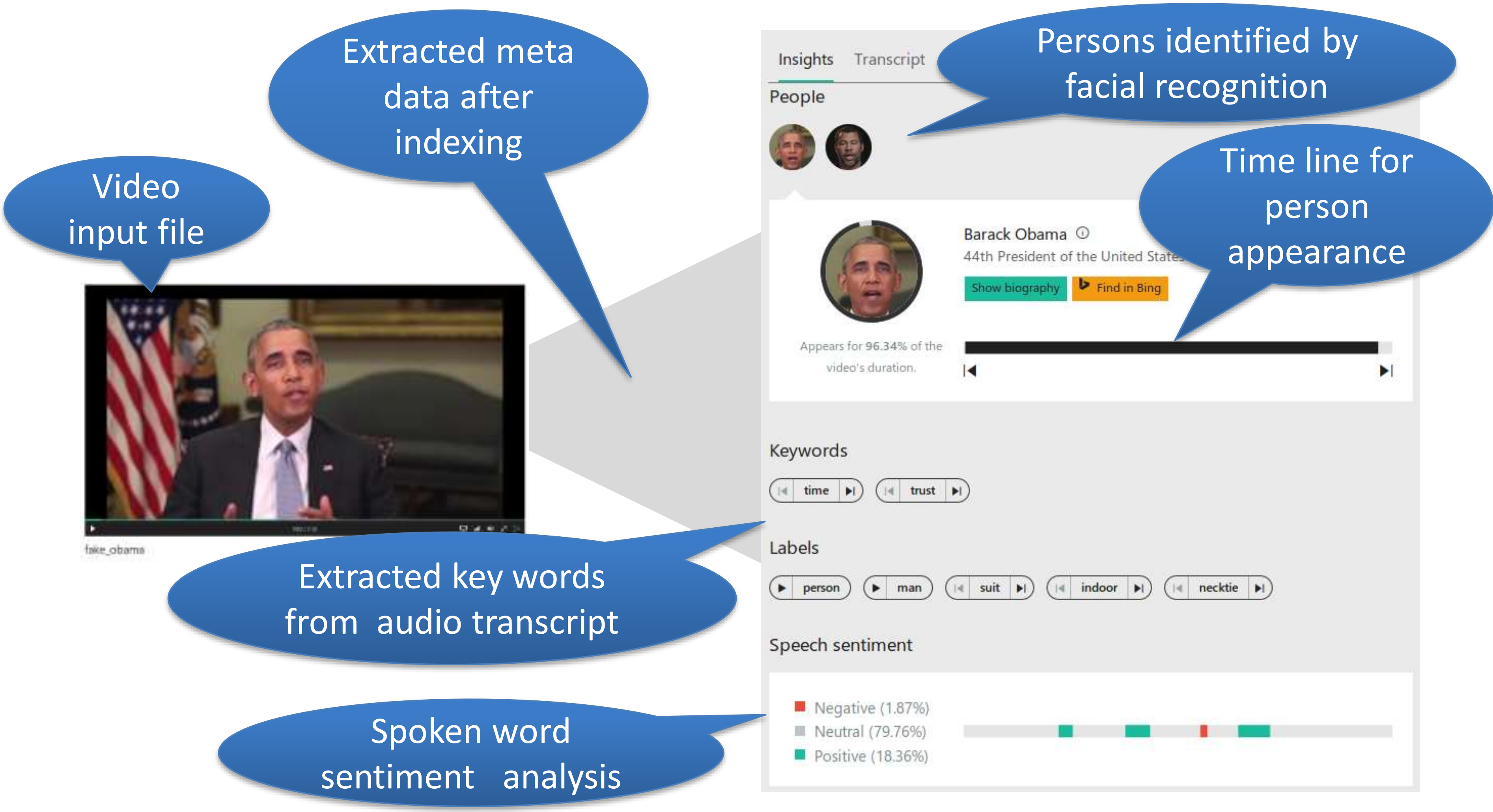
Facility mgt

+ more

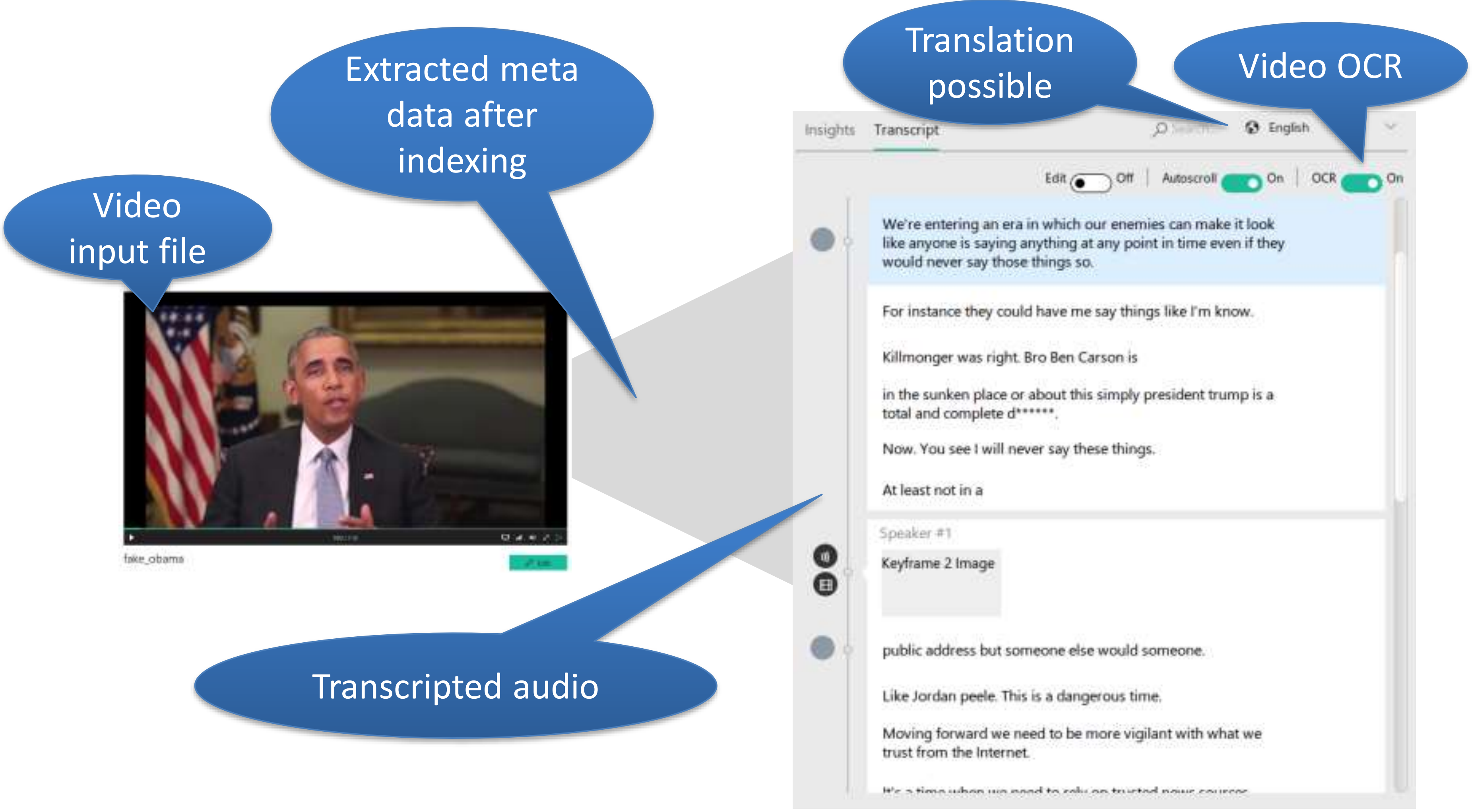
SE methodology



Open Source Intelligence Gathering (OSINT)



Extracted meta data after indexing
Persons identified by facial recognition
Video input file
Time line for person appearance
Insights
Transcript
People
Barack Obama
44th President of the United States
Show biography
Find in Bing
Appears for 96.34% of the video's duration.
fake_obama
Keywords
time
trust
Labels
person
man
suit
indoor
necktie
Extracted key words from audio transcript
Speech sentiment
Negative (1.87%)
Neutral (79.76%)
Positive (18.36%)
Spoken word sentiment analysis


Video input file
Extracted meta data after indexing
Translation possible
Video OCR
Transcripted audio
fake_obama
Insights
Transcript
English
Edit
Off
Autoscroll
On
OCR
On
We're entering an era in which our enemies can make it look like anyone is saying anything at any point in time even if they would never say those things so.
For instance they could have me say things like I'm know.
Killmonger was right. Bro Ben Carson is
in the sunken place or about this simply president trump is a total and complete d******.
Now. You see I will never say these things.
At least not in a
Speaker #1
Keyframe 2 Image
public address but someone else would someone.
Like Jordan peele. This is a dangerous time.
Moving forward we need to be more vigilant with what we trust from the Internet.

Blended SE attacks

Remote

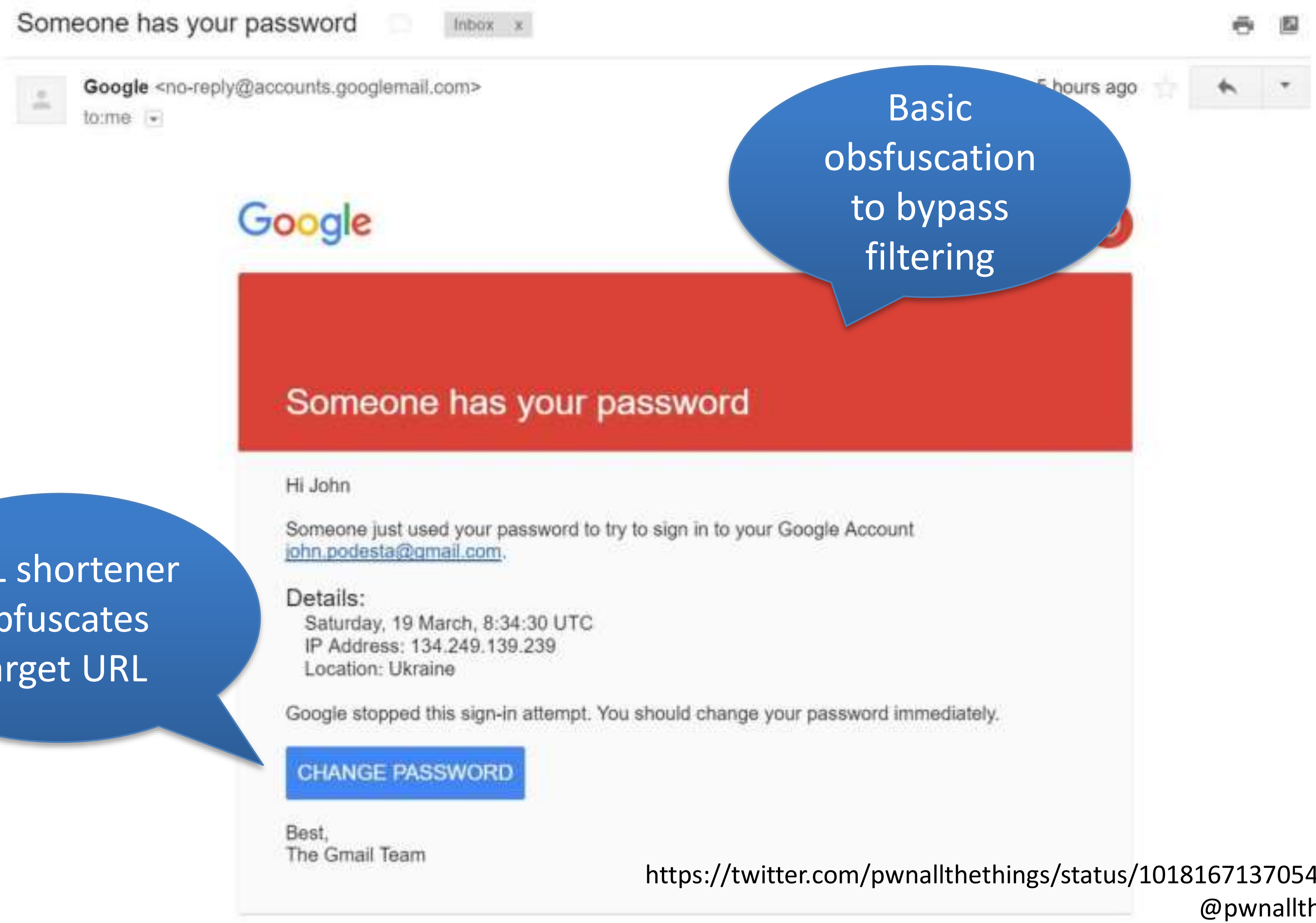
Email
Messaging
SMS
Voice
etc

Local

F**k
0-days

Path of least resistance

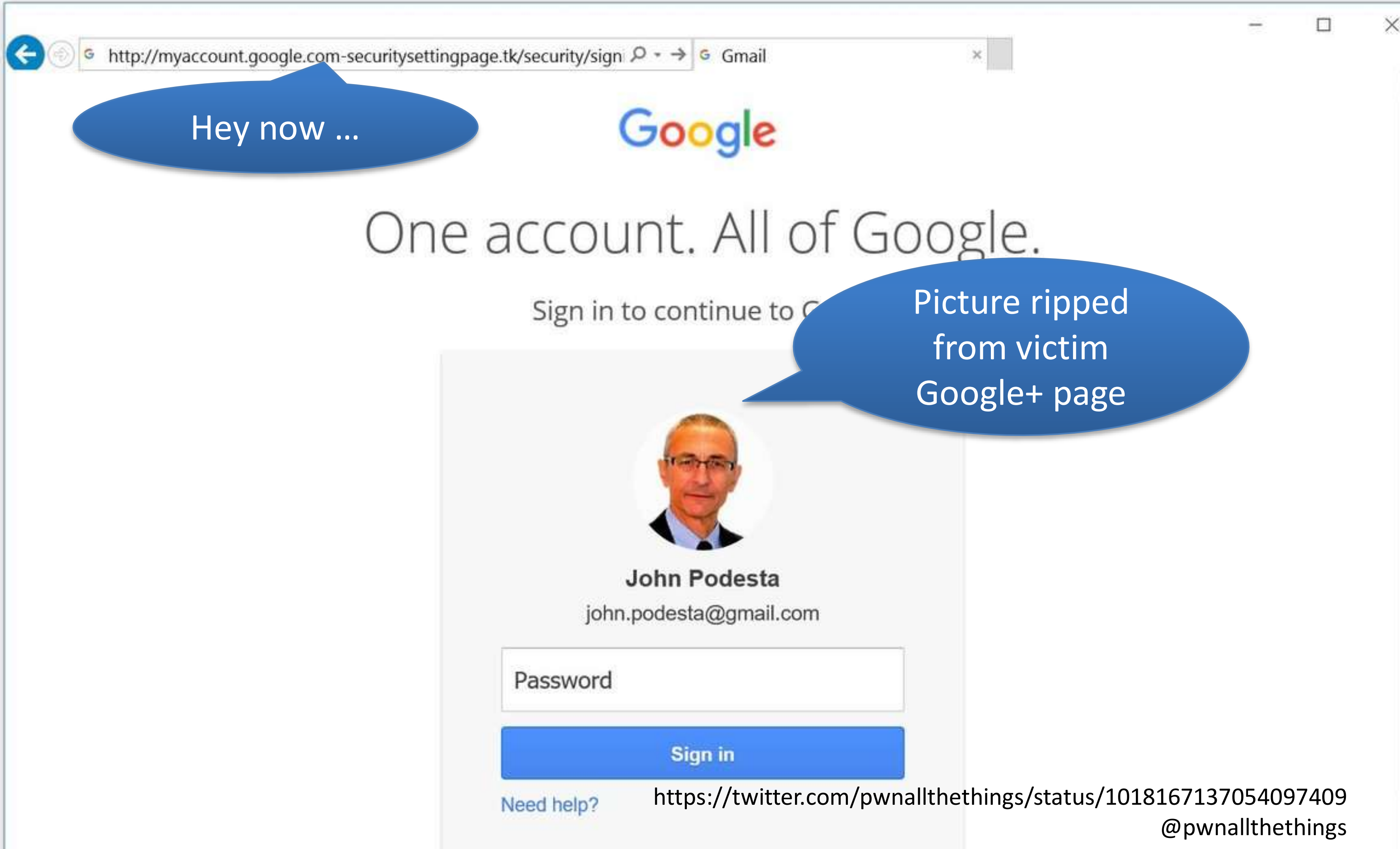
Basic stuff works



Basic obsfuscation to bypass filtering

URL shortener obfuscates target URL

https://twitter.com/pwnallthethings/status/1018167137054097409
@pwnallthethings



Hey now …

Picture ripped from victim Google+ page

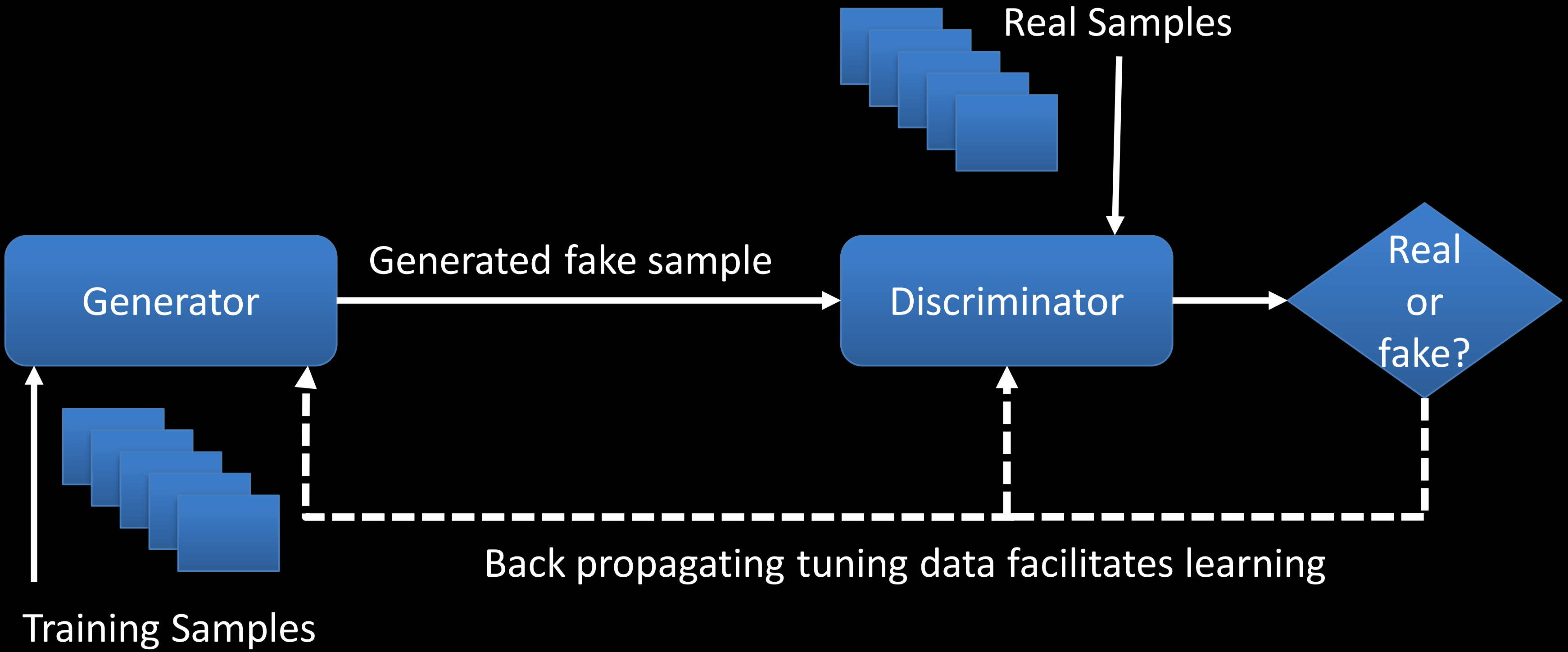
https://twitter.com/pwnallthethings/status/1018167137054097409
@pwnallthethings


COMING SOON

New improved Deepfakes

P0rn drives innovation once again

Generative Adversarial Network (GAN)



https://github.com/goodfeli/adversarial
"Generative Adversarial Networks." Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. ArXiv 2014.

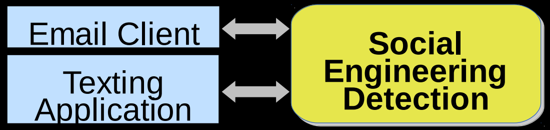
Got SE Defense?

Photo: Marvel Entertainment; John Byrne and Glynis Wein

Current SE defense

- Technology focus (headers etc.)
  - ➢ Emails, mainly
- Keyword filters
  - ➢ Without context

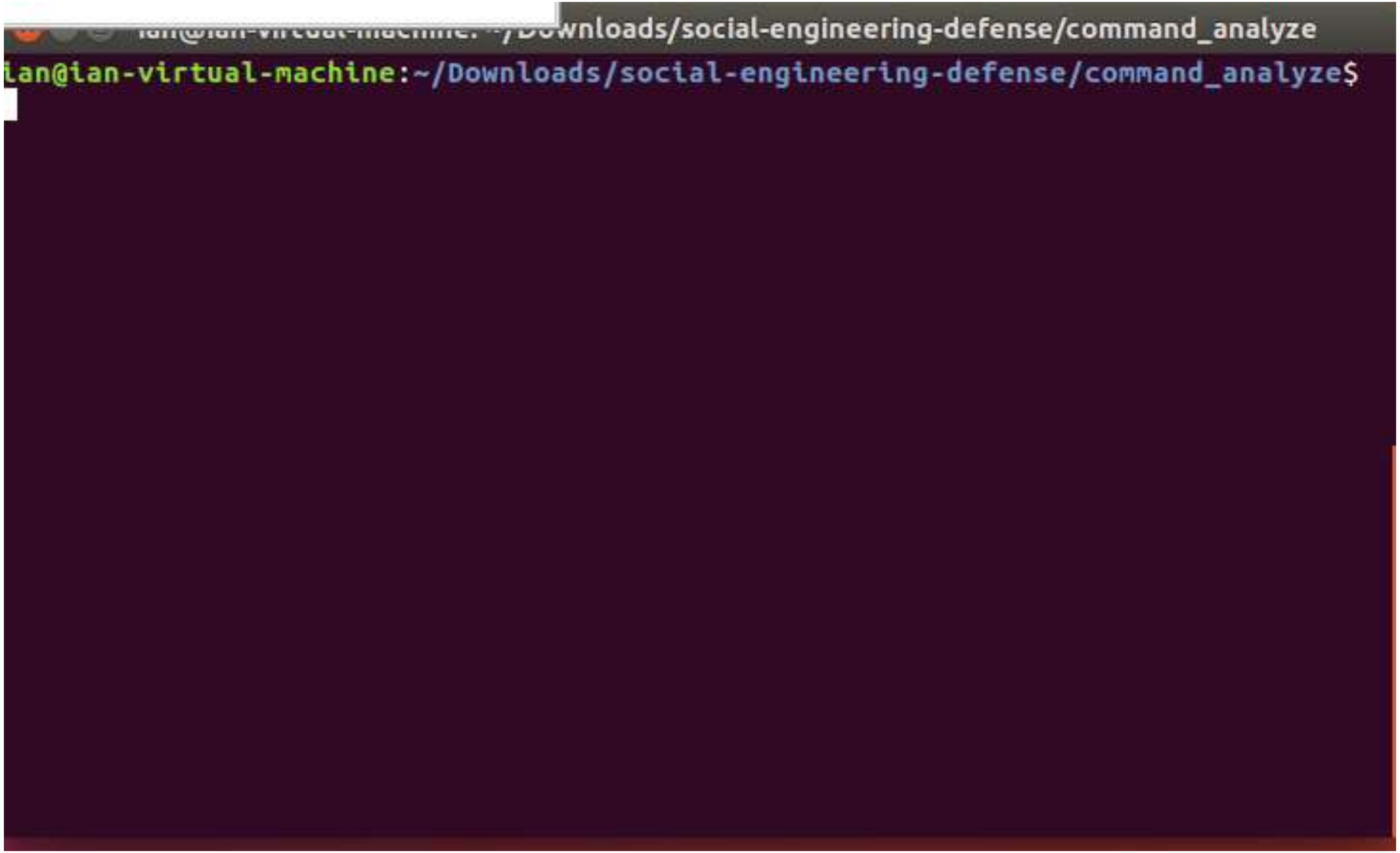
Use Cases for Attack Detection



- Difficult because evidence is only in the text of the dialog
- Cannot rely on vector-specific cues
  - images on a phishing website
  - links in a phishing email
- Need to perform some **semantic analysis**
  - consider the meaning of the dialog

Common Features of SE Attacks

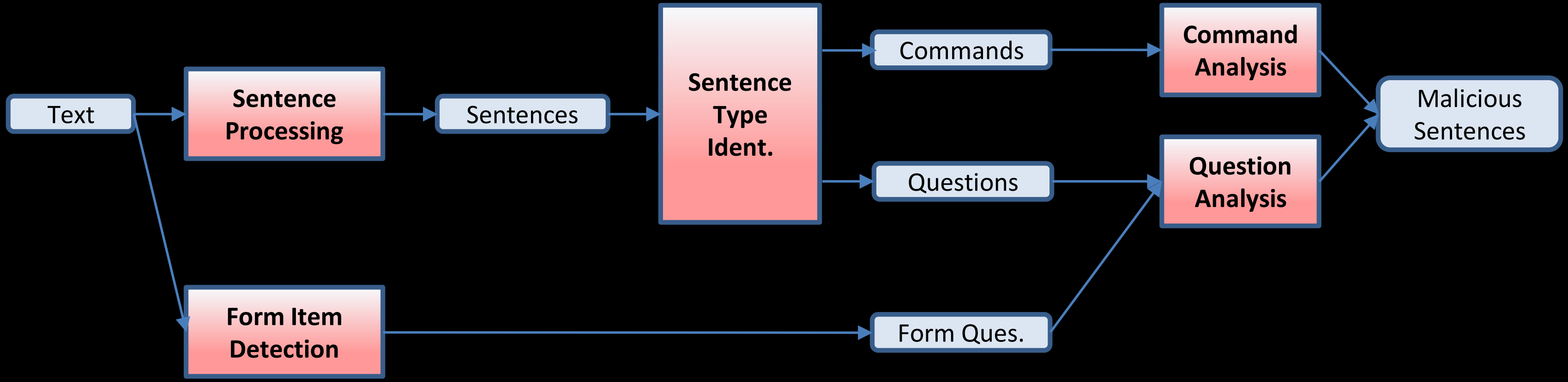
- In a social engineering dialog, the attacker must perform one of the following dialog acts:
  1. **Ask an inappropriate question**
     - “What is your social security number?”
  2. **Issue an inappropriate command**
     - “Please click on this link.”


ian@ian-virtual-machine:~/Downloads/social-engineering-defense/command_analyze$

Different approach needed

- Not just technical headers
  - Not just emails
- No filtering without context
  - Goodbye“spam filter”

System Structure



- **Question Analysis** and **Command Analysis** are the main steps

Detecting Questions/Commands

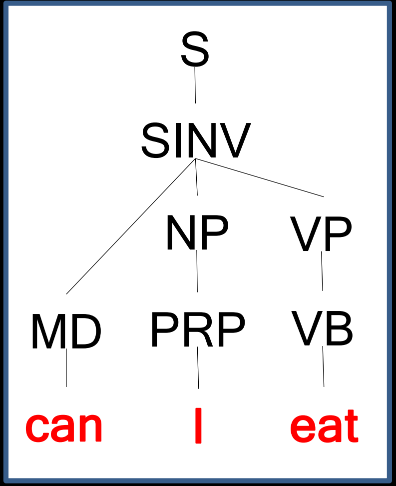
- Parse each sentence using a **syntactic parser**
  - **Stanford Parser**, https://nlp.stanford.edu/software/lex-parser.shtml
- Resulting **parse tree** reveals syntactic structure
  - Parts of speech, phrase decomposition
- Syntactic features are used to identify questions/commands

Question Detection

- Yes/No questions include subject/auxiliary inversion
- The auxiliary verb appears before the subject
  - Auxiliary verbs are “helper” verbs which add meaning
  - “will”, “may”, “can”, etc.
- “I can eat.” vs. “Can I eat?”

Recognition of Yes/No Questions

- SQ or SINV tag



Question Analysis

- Our goal is to determine if the answer to a question is private or not
- Sound an alarm if the answer is private data

1. “Where is the bathroom?”, answer is not private
2. “What is your social security number?”, private, alarm

Question Answer Systems

- User enters a question in natural language
- System provides an answer to the question

  “What is the tallest building in South Korea?”

  Lotte World Tower
- Search a structured database
  - DBPedia – structured data from wikipedia

Paralex QA System

"Paraphrase-Driven Learning for Open Question Answering", Anthony Fader and Luke Zettlemoyer and Oren Etzioni, ACL, 2013

rel	arg1	arg2
be_official_language.r	Cantonese	Hong Kong
be_plural_for.r	Bacterium	Bacteria
be_highest_mount.r	Ararat	Turkey

- Searches SQLite database
- Each entry is a triple, (relation, arg1, arg2)

Paralex QA Queries

Natural language:

“What is the nickname of Kansas?”

Query:

SELECT arg2 FROM tuples WHERE rel= “be-nickname.r”
AND arg1= “kansas.e”

Answer:

sunflower-state.e, Private = No

Multiple Queries

- Many SQL queries are generated from each question
- Top ranked SQL query is chosen

"What year was apple founded?"

1. SELECT arg1 FROM tuples WHERE rel= "found.r" AND arg2= "apple.e"
    - Answer is **steve-jobs.e**
2. SELECT arg2 FROM tuples WHERE rel= "be_found_on.r" AND arg1= "apple-computer.e"
    - Answer is **april-1-1976.e**

Modification to Database

rel	arg1	arg2
social_security_num.r	<user>	--------------
password.r	<user>	--------------
location.r	router	--------------

- Only keep private triples which describe your assets
- If triple is found in the database, the data is private
- Do not keep actual private data

Privacy from Queries

- Assume that the correct answer is somewhere among the top 15 answers
- A question is private if any of the top 15 answers private

- Increases the rate of true positives
- May create false positives

Command Analysis

- Determine if the answer to a command is forbidden or not
- Sound an alarm if the command is a forbidden action

1. "Take a left at the next corner.", command is OK
2. "Please tell me your social security number." forbidden, alarm

Command Summarization

- Represent command with **verb-direct object**

1. “Take a left at the next corner”

   (“take”, “left”)

2. “Please give me your password.”

   (“give”, “password”)

topic blacklist

Verb and Direct Object

- Use **Stanford Typed Dependency Parser** to find the verb and its direct object
- Determines semantic relationships between words

“Please give me your password”

`dobj(give-2, password-5)`

- `dobj` relates verb to its direct object

Topic Blacklist

Verb	Direct Object
give	password
send	money

- Pairs can be compiled to protect your assets
- We found most relevant pairs in phishing emails
- Used **term-frequency inverse document frequency (TF-IDF)** metric
    - TF-IDF ranking is high if pair is in phishing emails but not in non-phishing emails
    - 100,000 phishing emails and non-phishing emails examined

Custom Blacklist

Experimental Datasets

Database	URL	Size
Scamdex	http://www.scamdex.com	56555
Scamwarners	http://www.scamwarners.com	43241
Scamalot	http://scamalot.com	18149
Antifraudintl	http://antifraudintl.com	69209
Total		187154

- Evaluated phishing emails
  - Non-email attacks not available
- Trained with 100,000
  - private answers
  - verb-object blacklist
- Non-phishing emails taken from the Enron Email Dataset
  - https://www.cs.cmu.edu/~enron/

Experiment Results

	Phishing	Enron
Detected	56616 (True Positive)	14168 (False Positive)
Not-Detected	30432 (False Negative)	72880 (True Negative)

- Precision (TP/(TP+FP)) = 0.80
- Recall (TP/(TP+FN)) = 0.65

- Why so many False Negatives and False Positives?

False Negatives

- 35% of phishing emails were not detected

- Our approach only detects the **punchline** of the attack
  - Malicious question/command
- We cannot detect pretexting or elicitation
- Phishing attacks often involve a **sequence of emails**
- Only the final email may contain the punchline

Analysis of False Negatives

- Manually checked 100 False Negative emails
- 79% were early in the sequence, before the punchline

```
MY NAME IS MR TERRY ARUMAH FROM GHANA WEST AFRICA . I
AM A MARKETING MANGER …
IF YOU ARE INTERESTED PLEASE YOU CAN CALL US HERE
+2335403977 OR REPLY US HERE OKAY.
```

- All pretext, invitation to continue the conversation
- Punchline would occur in a later email

Our approach

- Focus on human communication
- Any text-based communication
  - Or speech converted to text
- Language and context analyzed

Thank You