Aalto University

http://asokan.org/asokan/

@nasokan on twitter

Security, Privacy and Machine Learning

N. Asokan
Secure Systems Group, Aalto University

What we will learn today

Why worry about security and privacy of machine learning (ML) applications?

What is an example of applying ML to a security/privacy problem?

[From a security/privacy perspective, what to watch out for when applying ML?]

How do you evaluate ML-based systems?

Effectiveness of inference

- accuracy/score measures on held-out test set?

Performance

- inference speed and memory consumption?

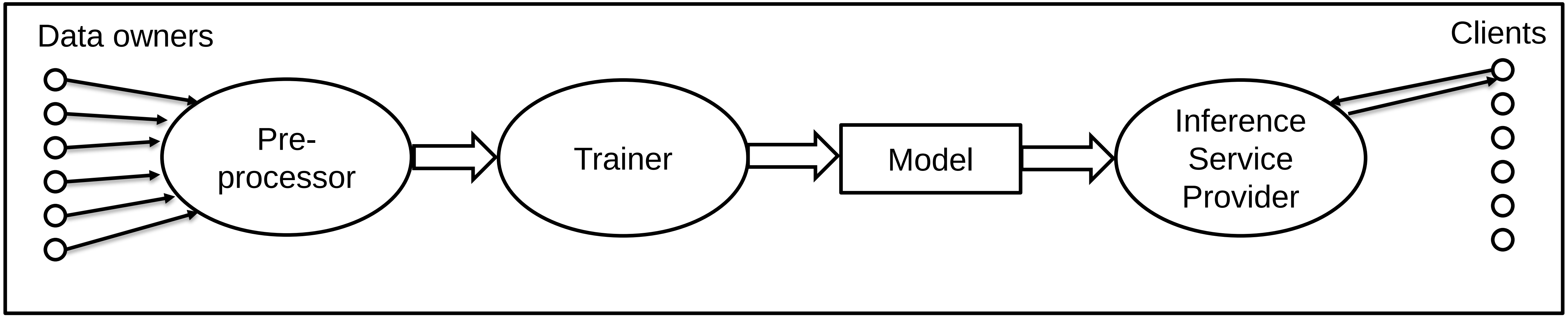
Hardware/software requirements

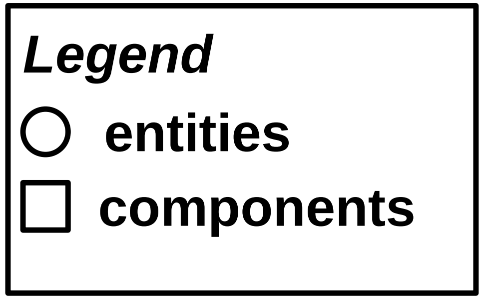
- e.g. memory/processor limitations, or specific software library?

Security & Privacy?

Meeting requirements in the presence of an adversary

Machine learning pipeline



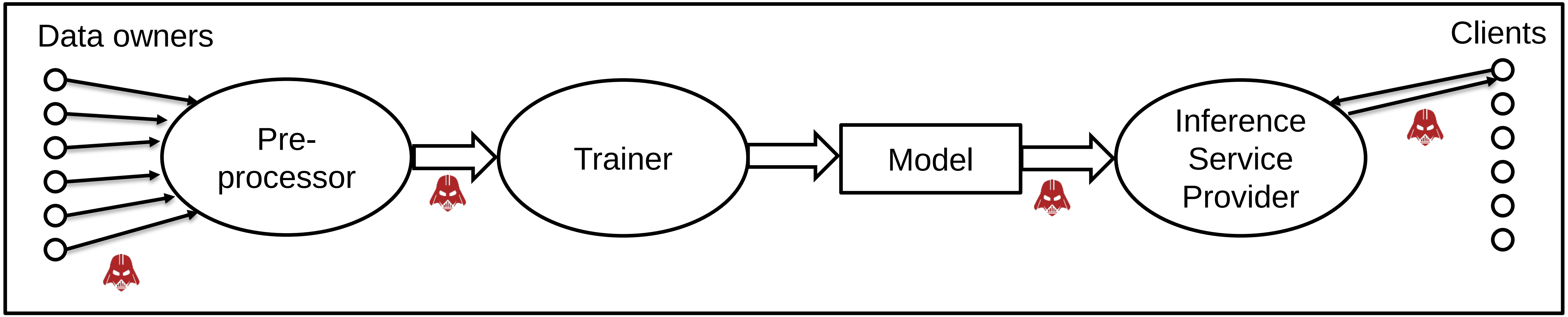


Adversarial behaviour

Different concerns arise depending on

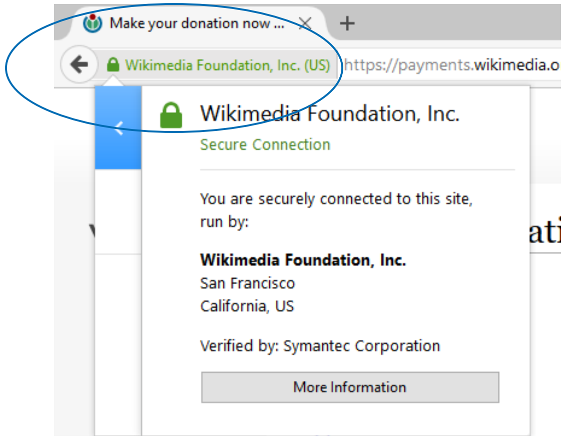
- **Who is the adversary?**
  - resources, capabilities, goals
- **What is its target?**
  - model, training data, input/output for predictions
- **What property does it wants to compromise?**
  - e.g., confidentiality, integrity

External adversaries

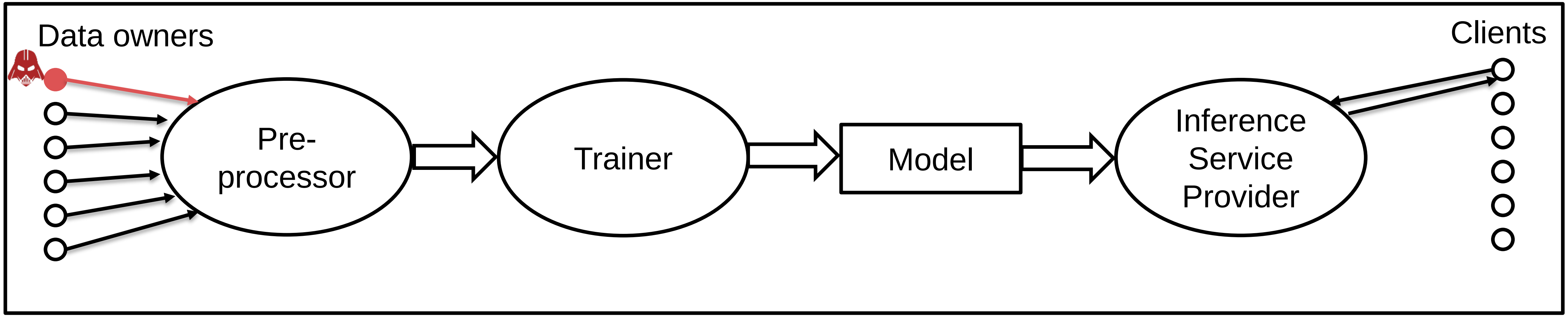


Standard security mechanisms can protect against external adversaries

- **Authentication, integrity, confidentiality**



1. Malicious data owners

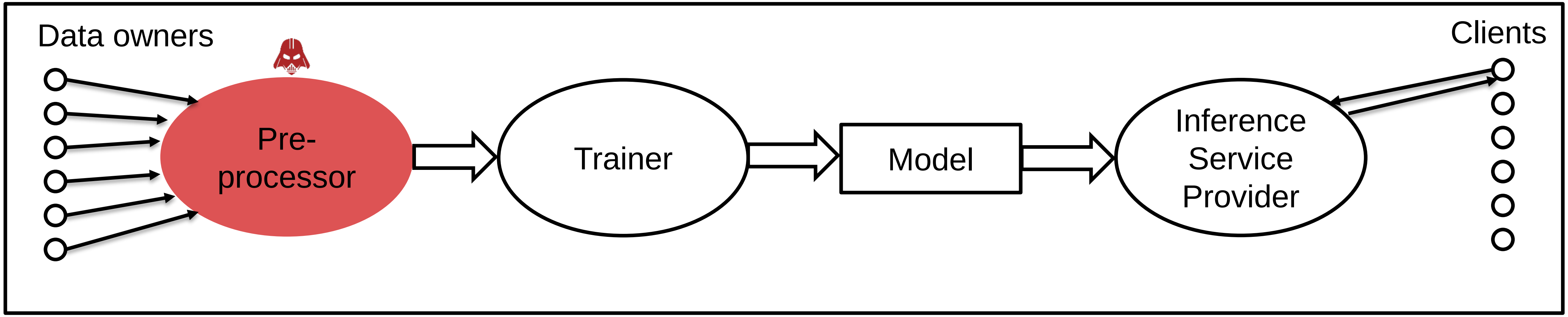


Attack target	Risk	Remedies
Model (integrity)	Data poisoning [1, 2]	Access control Robust estimators Active learning (human-in-the-loop learning) Outlier removal / normality models

[1] https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist

[2] https://wikipedia.org/wiki/Naive_Bayes_spam_filtering#Disadvantages

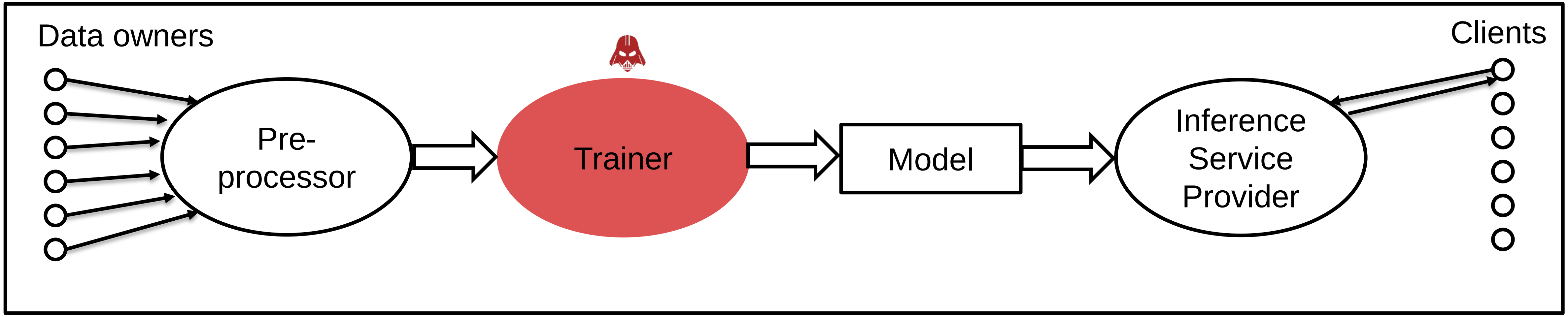
2. Malicious pre-processor



Attack target	Risk	Remedies
Model (integrity)	Data poisoning	Access control Robust estimators Active learning (human-in-the-loop learning) Outlier removal / normality models
Training data (confidentiality)	Unauthorized data use (e.g. profiling)	Adding noise (e.g. differential privacy) [1] Oblivious aggregation (e.g., homomorphic encryption)

[1] Heikkila et al. "Differentially Private Bayesian Learning on Distributed Data", NIPS'17

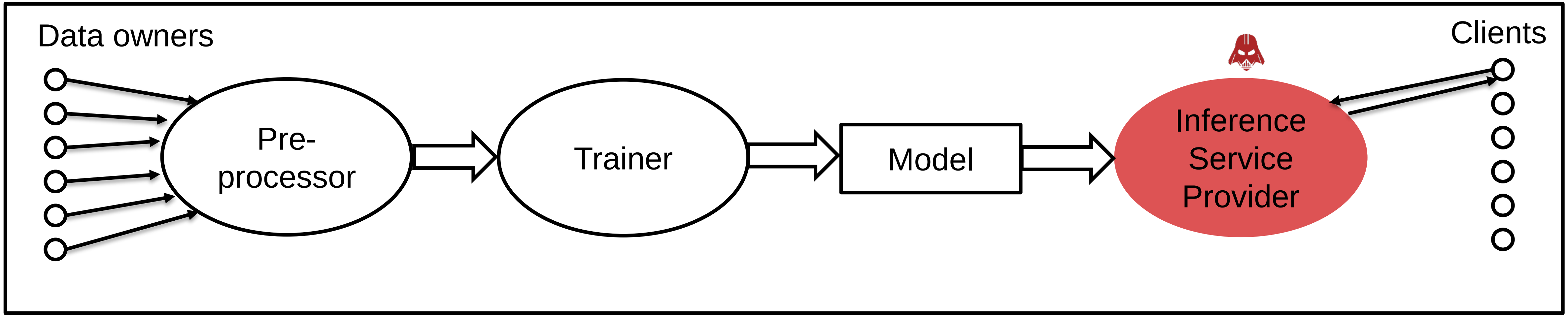
3. Malicious model trainer



Attack target	Risk	Remedies
Training data (confidentiality)	Unauthorized data use (e.g. profiling)	Oblivious training (learning with encrypted data) [1]

[1] Graepel et al. “ML Confidential”, ICISC’12

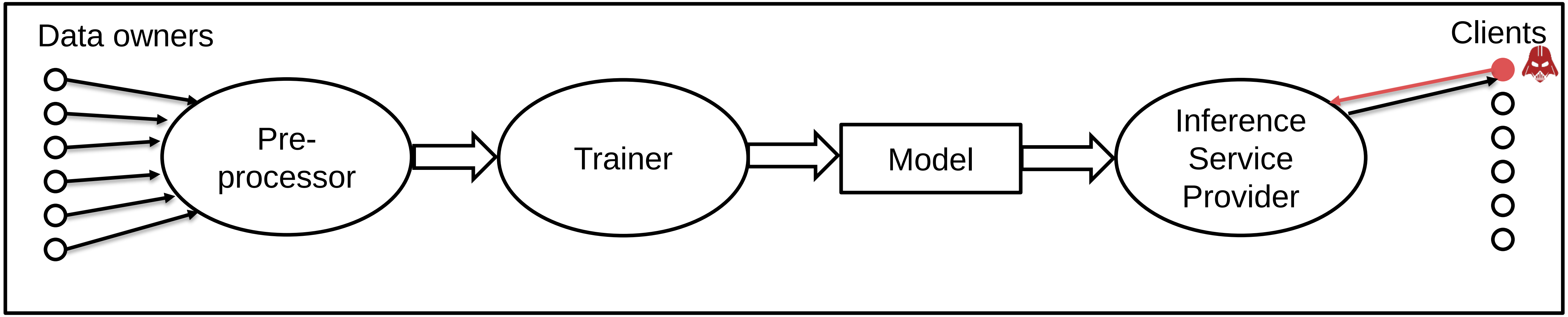
4. Malicious inference service provider



Attack target	Risk	Remedies
Inference queries/results (confidentiality)	Unauthorized data use (e.g. profiling)	Oblivious inference [1,2,3]

[1] Gilad-Bachrach et al. "CryptoNets", ICML'16
[2] Mohassel et al. "SecureML", IEEE S&P'17
[3] Liu et al. "MiniONN", ACM CCS'17

5. Malicious client



Attack target	Risk	Remedies
Training data (confidentiality)	Membership inference Model inversion	Minimize information leakage in responses Differential privacy
Model (confidentiality)	Model theft [1]	Minimize information leakage in responses Normality model for client queries Adaptive responses to client requests
Model (integrity)	Model evasion [2]	

[1] Tramer et al, “Stealing ML models via prediction APIs”, UsenixSEC’16
[2] Dang et al, “Evading Classifiers by Morphing in the Dark”, CCS’17

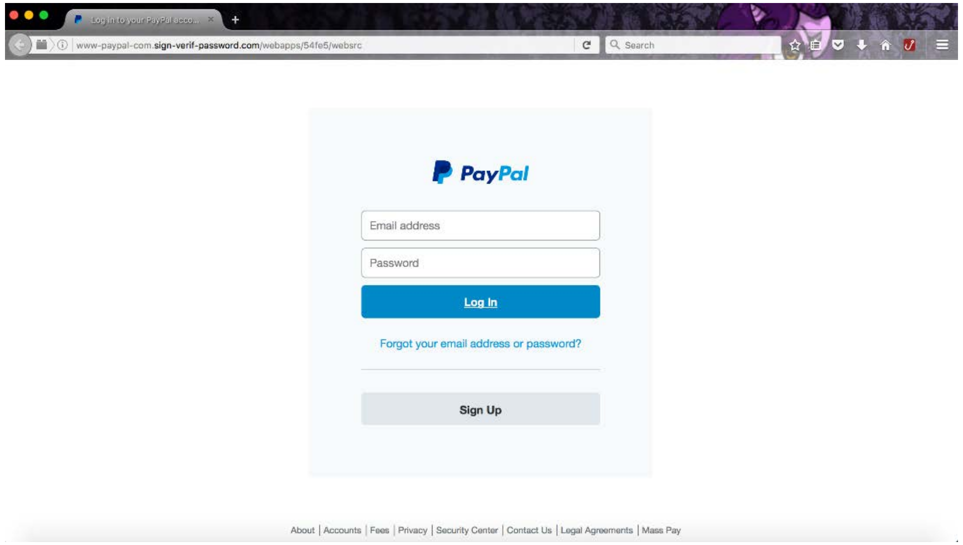
Fast client-side phishing detection

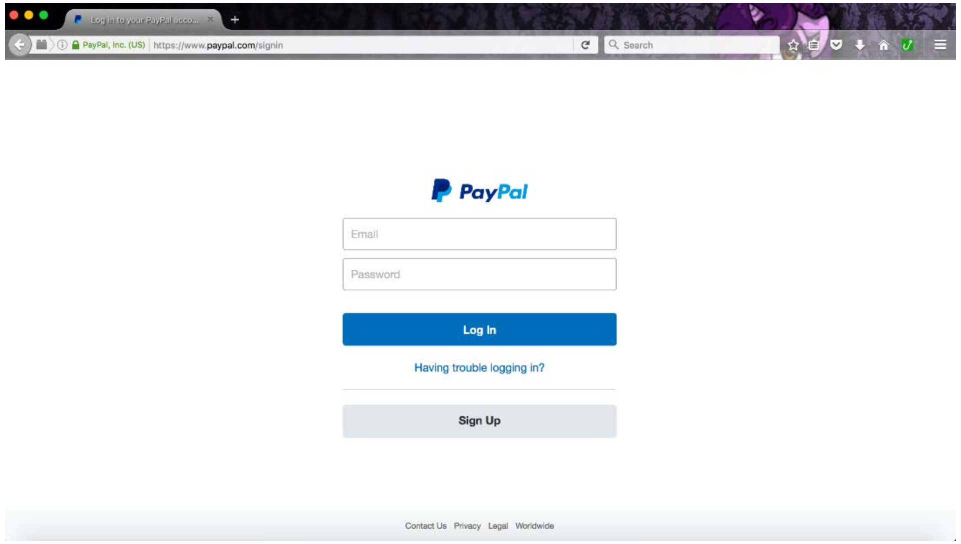
A case-study in applying machine learning to solve security/privacy problems

N. Asokan

(joint work with Samuel Marchal, Giovanni Armano, Kalle Saari, Tommi Gröndahl, Nidhi Singh)

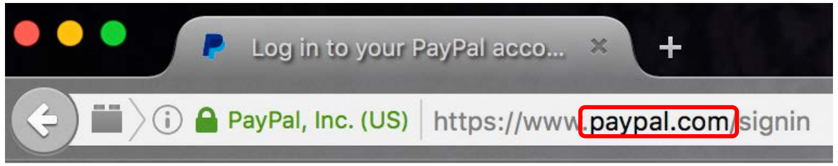
Phishing webpages





Phishing webpage (phish)

Legitimate webpage



State of the art in phishing detection

Centralized black lists

- vulnerability to “dynamic phishing”: content depends on client
- Update time lag
- threat to user privacy

Google Safe Browsing McAfee SiteAdvisor WOT …

Application of machine learning

- may not have “temporal resilience”: accuracy degrading with time

Using ML to identify phishing websites

Data points:

- **Webpage contents**

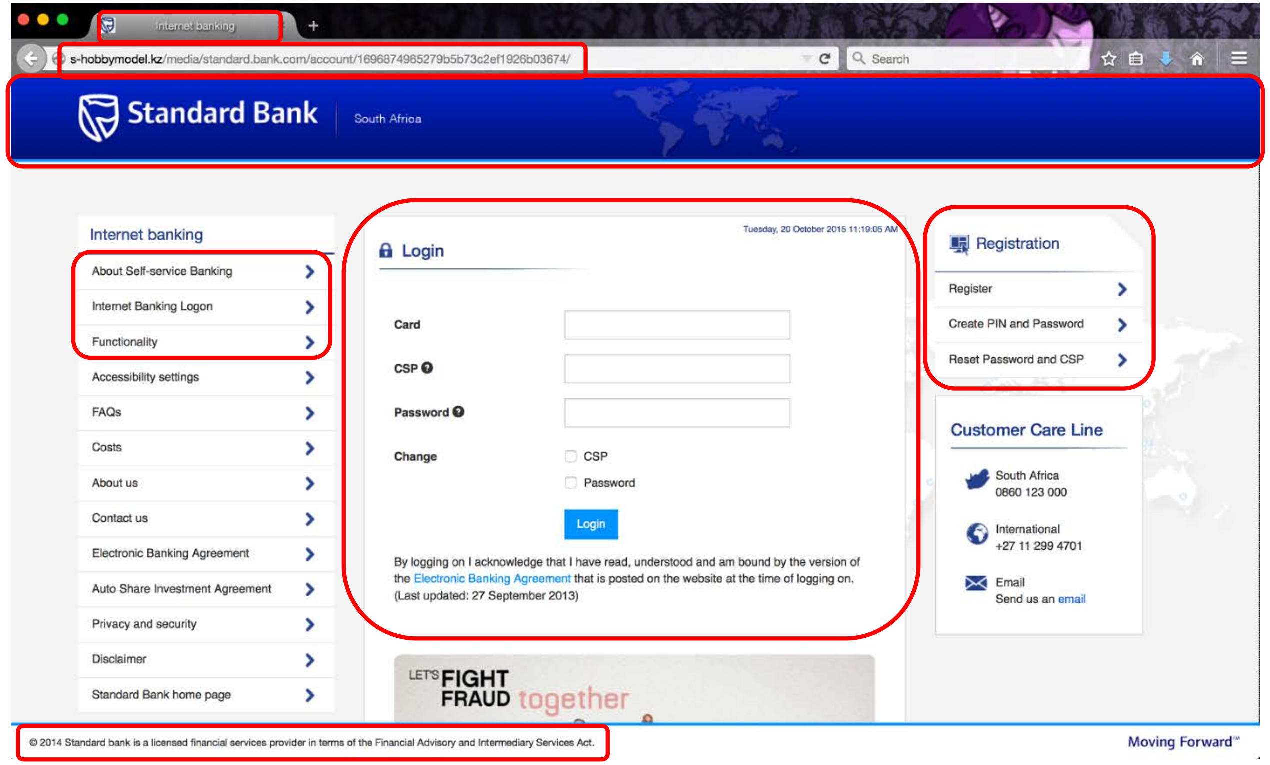
Labels:

- **“phish”, “not phish”**

Features:

(think about the adversary)

Data sources on a webpage



Starting URL
Landing URL
Redirection chain
internal

Logged links
external

HTML source code:

- Text
- Title
- HREF links
- Copyright

Phisher’s control & constraints

Data sources differ in terms of the levels of

- control the phisher has over a source
- constraints placed on the phisher in manipulating that source

URL Structure

$$\textit{protocol:}\overbrace{\textit{//[subdomains.]}}^{\text{FreeURL}}\overbrace{\textit{mld.ps}}^{\text{Registered Domain Name}}\overbrace{\textit{[/path][?query]}}^{\text{FreeURL}}$$

https://www.amazon.co.uk/ap/signin?_encoding=UTF8

- Protocol = *https*
- Registered domain name (RDN) = *amazon.co.uk*
- Main level domain (*mld)* = *amazon*
- FreeURL = *{www, /ap/signin?_encoding=UTF8}*

Phisher’s control & constraints

Control:

- **External** loaded content (logged links) and **external** HREF links are *usually* **not controlled** by page owner.

Constraints:

- **Registered domain name** part of URL cannot be freely defined: **constrained** by DNS registration policies.

Conjectures

Improve phish detection by modeling control/constraints

- generalizable, language independent, hard to circumvent

Identity target of phish by analyzing terms in data sources

- guide users where they really intended to go

Data sources: control & constraints

	Unconstrained	Constrained
Controlled	Text Title Copyright Internal *FreeURL (2)*	Internal *RDNs* (2)
Uncontrolled	External *FreeURL (2)*	External *RDNs* (2)

Feature selection

A small set (212) of features computed from data sources:

- URL features (106): e.g., # of dots in *FreeURL*
- Consistency features (101)
- Webpage content (5): e.g., # of characters in *Text*

Features not data-driven: e.g., no bag-of-words features

- Conjecture: can lead to language-independence, temporal resilience

Consistency features

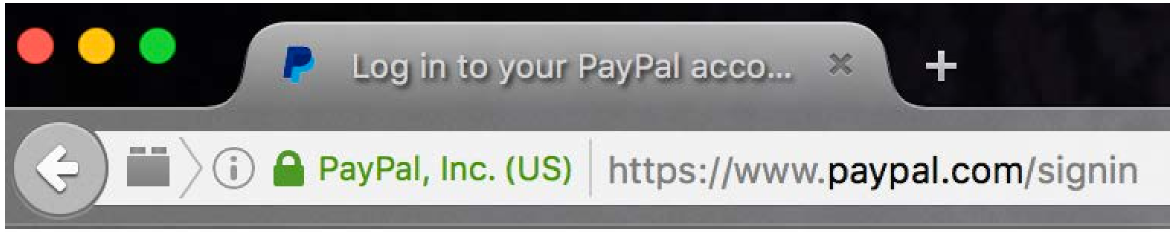
Term usage (66)

- strings of 3 or more characters, separated by standard delimiters

Usage of “Main level domain” *(mld)* from starting/landing URLs *(*32)

“Registered domain name” usage *(RDN)* (13)

Term usage consistency



Title: ***"Log in to your PayPal account"***

RDN*: *paypal.com

$$D_{title} = \{(\text{log},0.25);(\text{your},0.25);(\text{paypal},0.25);(\text{account},0.25)\}$$

$$D_{startrdn} = \{(\text{paypal},1)\}$$

Hellinger distance

$$\text{f} = H(D_{title}, D_{startrdn}) = \frac{\sqrt{0.25 + 0.25 + (\sqrt{0.25} - \sqrt{1})^2 + 0.25}}{\sqrt{2}} = 0.71$$

Classification

Decision trees:

- Easier understanding of the decision process (intelligibility)
- Ability to learn from little training data
- Good performance with a small feature set
- No need for data normalization

Gradient Boosting (ensemble learning):

- Resilient to adversarial inference of model parameters
- Likelihood to belong to a class (score from individual learners) // no hard decision (good for tuning the decision)

→ **Fast decision**

Target identification

Identify terms representing the service/brand: keyterms

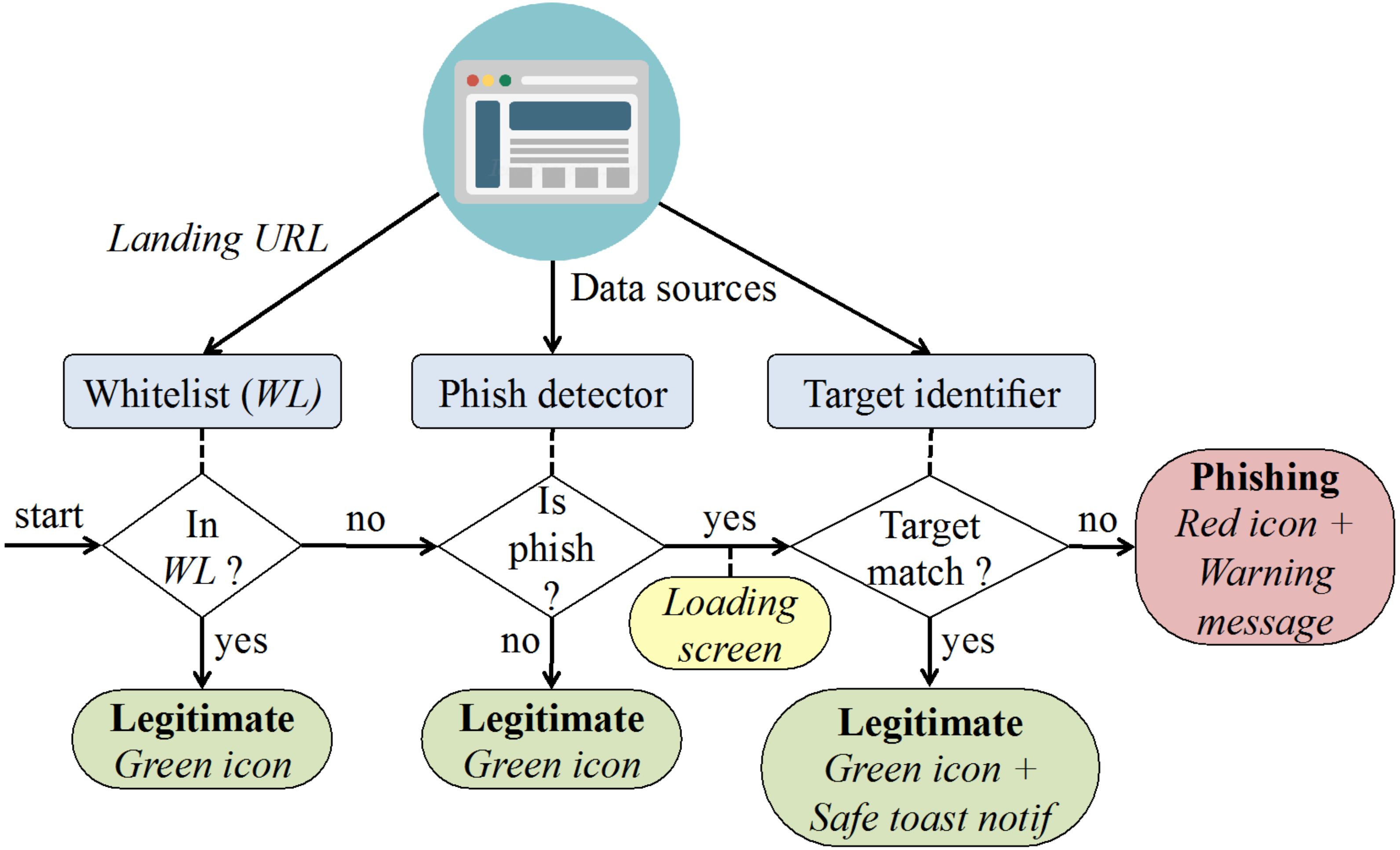
Assumption: keyterms appear in several data sources

→ Intersect sets of terms extracted from different **visible** data sources (title, text, starting/landing URL, Copyright, HREF links)

Query search engine with top keyterms:

- Website appears in top search results → legitimate
- Else, phish; top search results ~ potential targets of phishing

Off-the-Hook anti-phishing system



Off-the-Hook browser add-on

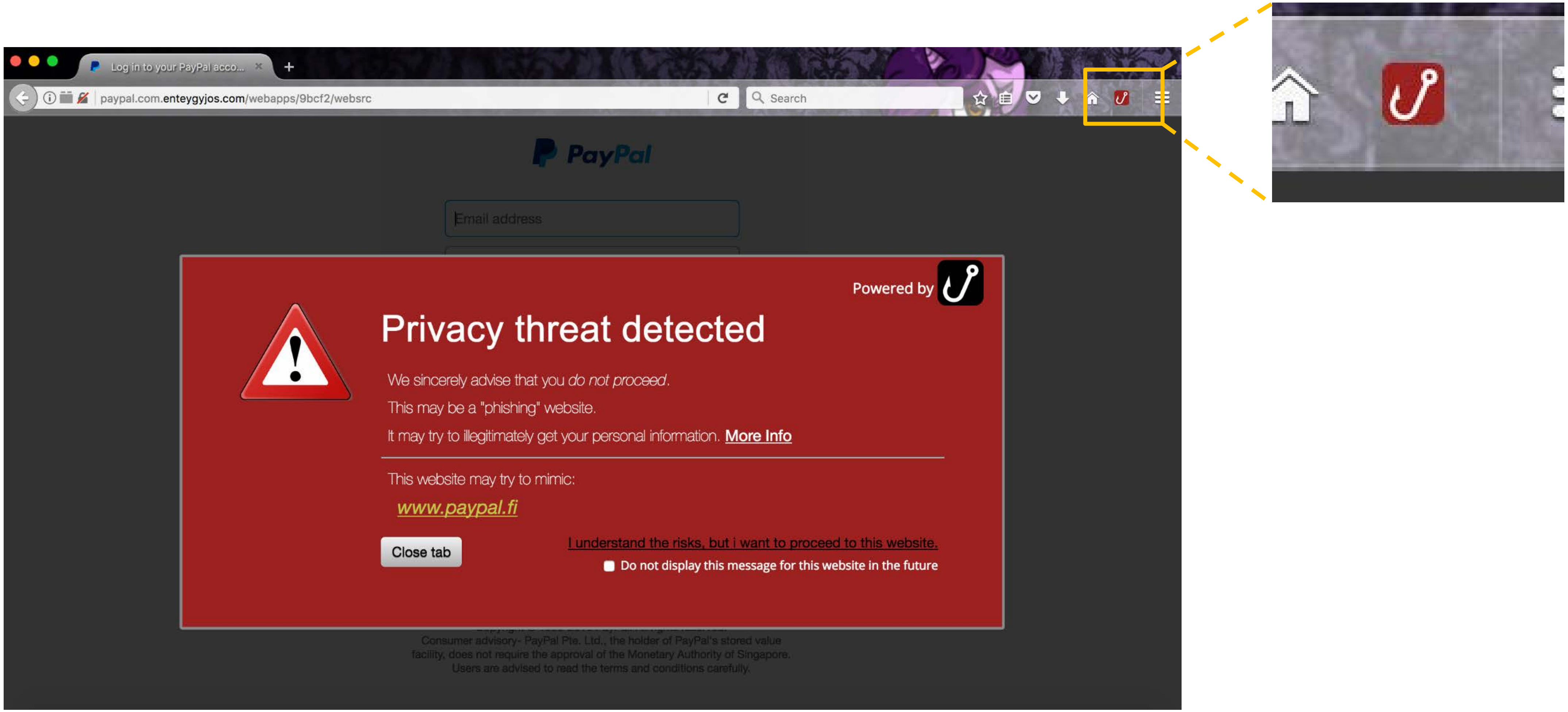
Client-side implementation

- Preserves user privacy
- Resists dynamic phishing

Multi-browser / Cross platform

- Chrome*, Firefox
- Windows (>= 8), Mac OSX (>= 10.8), Ubuntu (>= 12.04)

Off-the-Hook warning



[Skip to Off-the-Hook summary]

Evaluation

Classifier Training:

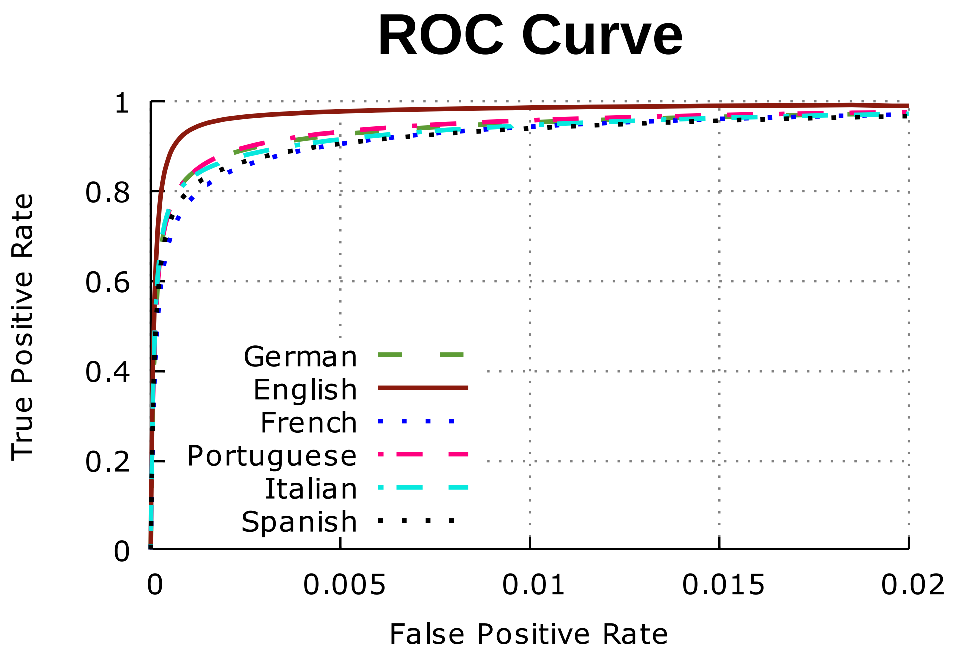
- 8,500 legitimate webpages (English)
- 1,500 phishing webpages (taken from PhishTank & manually verified)

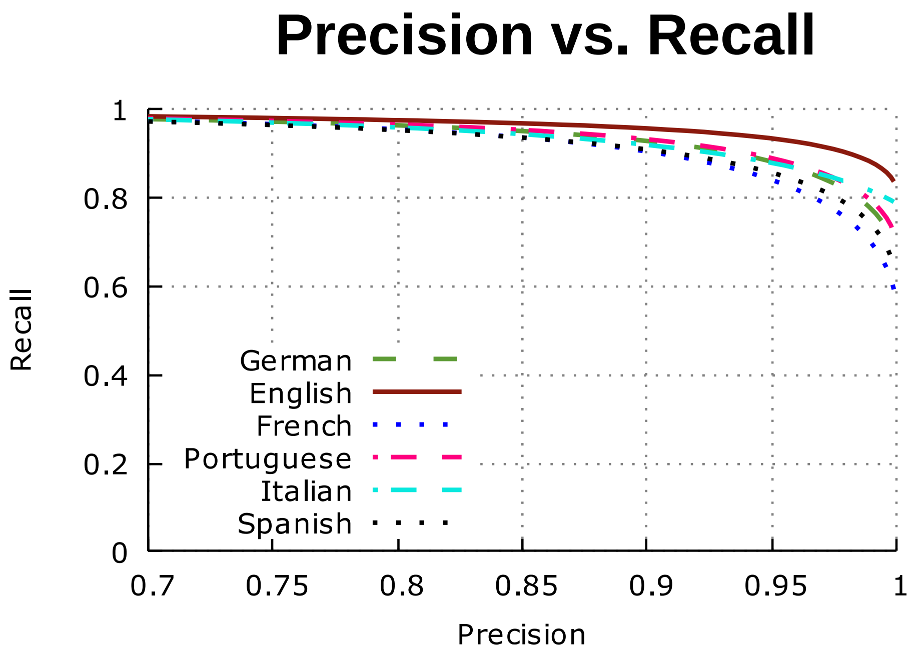
Evaluation:

- Legitimate webpages:
  - 100,000 English
  - 20,000 each in French, German, Italian, Portuguese and Spanish
- 2,000 phishing webpages (PhishTank; manually verified)

Skip to Off-the-Hook summary

Classification accuracy



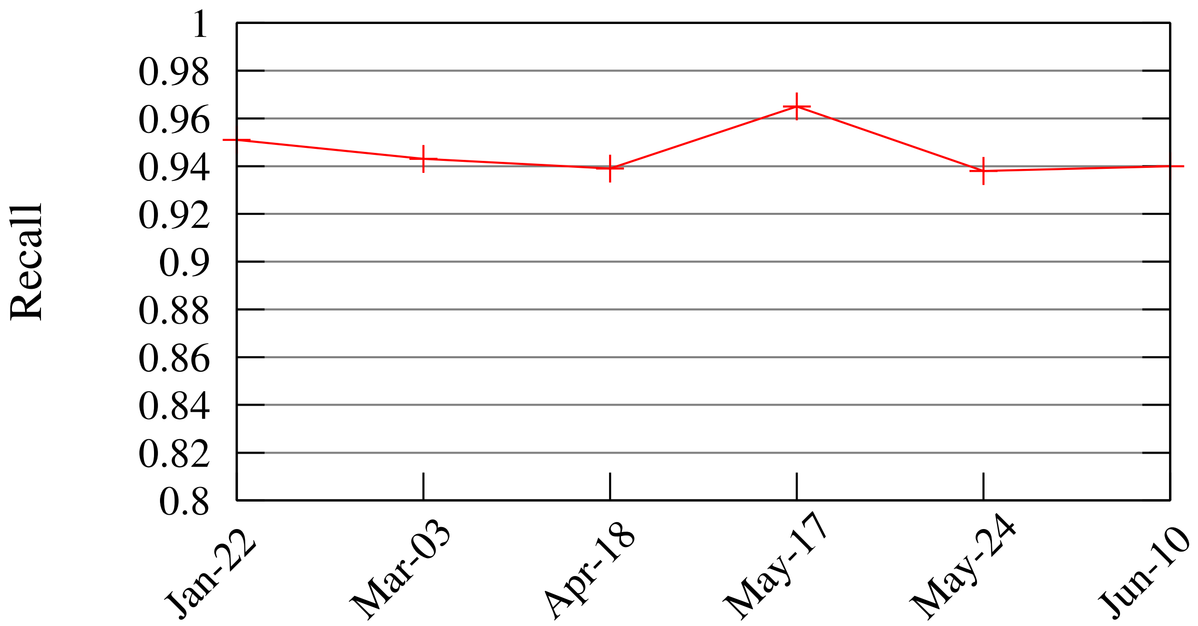


200,000 multi-lingual legit
/ 2,000 phishs
(≈ **real world distribution**)

Precision	***Recall***	***FP Rate***	***AUC***	***Accuracy***
0.975	0.951	**0.0008**	0.999	**0.999**

Skip to Off-the-Hook summary

Classification accuracy over time



Model trained:

- September 2015

Applied on phishs:

- January – June 2016
- ~2500 fresh, verified phishtank entries

Skip to Off-the-Hook summary

Comparison: effectiveness

	FPR	Precision	Recall	Accuracy
Cantina (CMU)	**0.03**	**0.212**	0.89	0.969
Cantina+ (CMU)	**0.013**	0.964	0.955	0.97
Ma et al. (UCSD)	0.001	**0.998**	0.924	0.955
Whittaker et al. (Google)	**0.0001**	0.989	0.915	**0.999**
Monarch (UCB)	0.003	0.961	**0.734**	**0.866**
Off-the-Hook	**0.0008**	0.975	**0.951**	**0.999**

Skip to Off-the-Hook summary

Comparison: dataset sizes

	Training	Testing
Cantina (CMU)	-	2,119
Cantina+ (CMU)	2062	884
Ma et al. (UCSD)	17,750	17,750
Whittaker et al. (Google)	9,388,395	1,516,076
Monarch (UCB)	750,000	250,000
Off-the-Hook	10,000	202,000

Off-the-Hook summary

https://ssg.aalto.fi/projects/phishing/

Off-the-Hook phishing website detection system:

- Exhibits language independence
- Resists dynamic phishing
- Fast: < 0.5 second per webpage (average for all webpages)
- Accurate: > 99.9% accuracy with < 0.1% false positives

Target identification system:

- Fast: < 2 seconds per webpage
- Success rate: > 90% (1 target); 97.3% (set of three potential targets)

Skip to conclusions

[MSSA16] Know Your Phish: Novel Techniques for Detecting Phishing Sites and their Targets, ICDCS 2016
[AMA16] Real-Time Client-Side Phishing Prevention Add-On, ICDCS 2016
[MAGSSA17] Off-the-Hook: An Efficient and Usable Client-Side Phishing Prevention Application, IEEE Trans. Comput., 2017

Pitfalls in using ML (for security)

Adversaries will circumvent detection

The ML model is intended to detect/counter attacks

Adversary *will* attempt to circumvent detection:

- poison learning process
- infer detection model
- mislead classifier

In Off-the-Hook:

- Modeling constraints and controls while training
- Adversary can control External RDNs!

→ **Resistance to adversaries**

Privacy concerns are multilateral

Data used for ML may be sensitive

- Sensitive information about users in
  - training data → model inversion, membership inference
  - prediction process → user profiling, e.g., in a cloud setting (ML-as-a-service)

By Source, Fair use, https://en.wikipedia.org/w/index.php?curid=54119040

In Off-the-Hook:

- Client-side classifier to avoid disclosure of URLs
- But model stealing may be a concern
- Better alternatives like "MiniONN"
  - Allows converting *any* neural network to an "oblivious" variant

→ **Multilateral privacy guarantees**

https://eprint.iacr.org/2017/452

Skip to conclusions

Classification landscapes are dynamic

Attacks evolve fast

Prediction instances likely differ from training instances

- E.g., Android malware evolves due to for changes in API

In Off-the-Hook:

- Avoidance of data-driven features
- Models that allow inexpensive retraining

➡ **Temporal resilience**

Maintaining labels is expensive

More training data is good; but unbalanced classes typical

Data about malicious behavior difficult to obtain

- Labeling is cumbersome, requires expertise, may be inaccurate or may evolve (e.g. phishing URLs)

In Off-the-Hook:

- Manage with small training sets
- Minimize ratio of training set size to test size

→ **Minimal training data**

Predictions need to be intelligible

Ability of humans to understand why a prediction occurs

- Detection as malicious → forensic analysis
- Explain predictions to users, e.g. why access is prevented
- “Explainability” obligations under privacy regulations like GDPR

In Off-the-Hook:

- Small set of “meaningful” features
- Use of (ensemble of) shallow decision trees

→ **Transparent decision process**

ML failures can harm user experience

Security is usually a secondary goal

Use of ML must not negatively impact usability

- Decision process should be efficient
- Wrong predictions may have a significant usability cost

In Off-the-Hook:

- Prediction effectiveness and speed
- In phishing detection, one false positive may be one too much!

➡ **Lightweight and accurate**

Skip to conclusions

Security/privacy applications: desiderata

Circumvention resistance

- Resistance to adversaries

Temporal resilience

- Resilience in dynamic environments

Minimality

- Use of minimal training data

Privacy

- Model privacy, training set privacy, and input/output privacy

Intelligibility

- Transparent decision process

Effectiveness

- Lightweight, accurate models

Did you learn:

Why worry about security and privacy of machine learning (ML) applications?

What is an example of applying ML to a security/privacy problem?

[From a security/privacy perspective, what to watch out for when applying ML?]

http://asokan.org/asokan/

@nasokan on twitter

Acknowledgements: Mika Juuti and Samuel Marchal contributed to making this presentation.