Don’t Use Computer Vision For Web Security

Florian Tramèr

CV-COPS

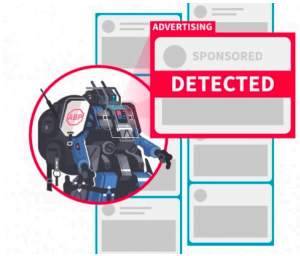
August 28th 2020

Computer Vision For Web Security

(Most) users ingest web content visually

Detection of undesirable content

can (partially) be framed as a computer vision problem



Ad-blocking

"Is this image an ad?"

Anti Phishing

"Does this webpage look similar to Google.com?"

Content takedown

"Is this a video of a terrorist attack"

Act I

Don’t Use Computer Vision For **Client-Side** Web Security

ML model is run on the user’s machine

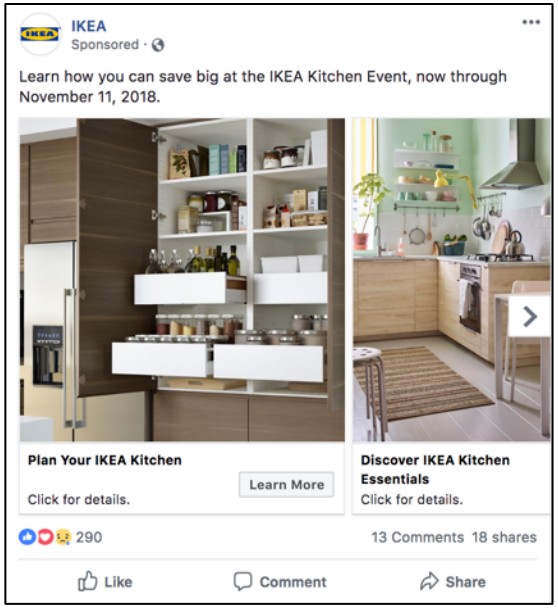
An illustrative example: Ad-Blocking

“AdVersarial: Perceptual Ad Blocking meets Adversarial Machine Learning”

(with Pascal Dupré, Gili Rusak, Giancarlo Pellegrino and Dan Boneh)

ACM CCS 2019, https://arxiv.org/abs/1811.03194

Why use CV for Ad-Blocking?



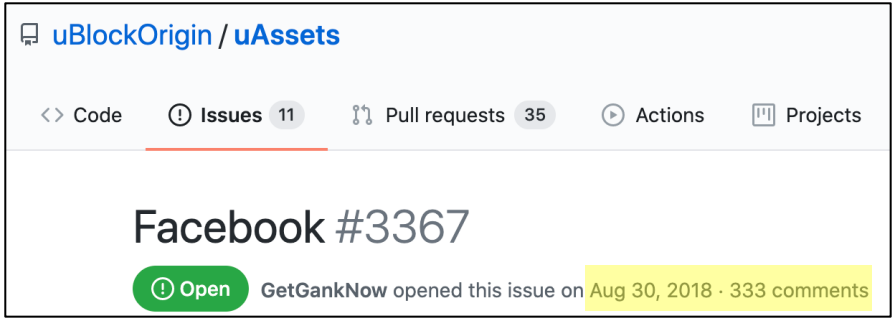
Humans should be able to recognize ads

Why use CV for Ad-Blocking?

Detecting ad-disclosures programmatically is hard!

```
<a><span>
<span class="c1">Sp</span>
<span class="c2">S</span>
<span class="c1">on</span>
<span class="c2">S</span>
<span class="c1">so</span>
<span class="c2">S</span>
<span class="c1">red</span>
<span class="c2">S</span>
</span></a>
```

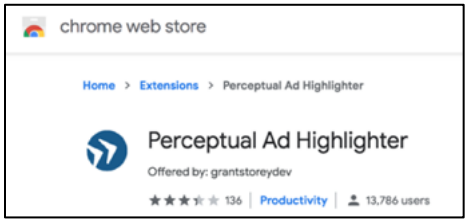
```
.c2 { font-size: 0; }
```

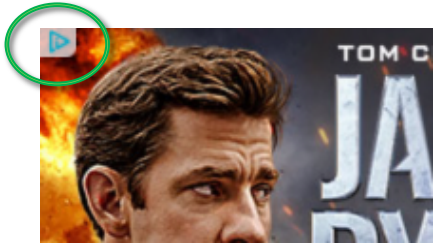


Perceptual Ad-Blocking

Ad Highlighter [Storey et al., 2017]

- *Traditional vision techniques (image hash, OCR)*



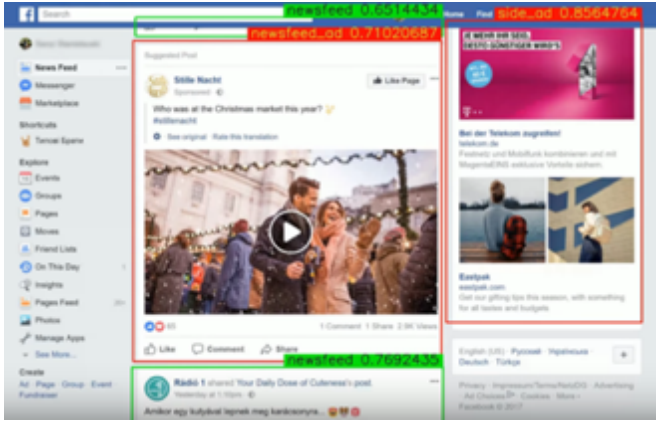


Sentinel by Adblock Plus [Paraska, 2018]

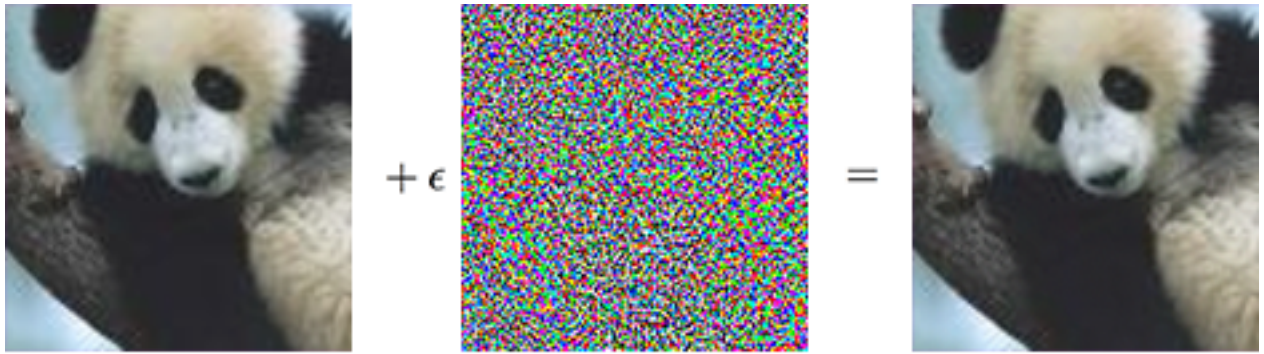
- *Locates ads in screenshots using neural networks*

Percival by Brave [Din et al., 2019]

- *CNN embedded in Chromium's rendering pipeline*

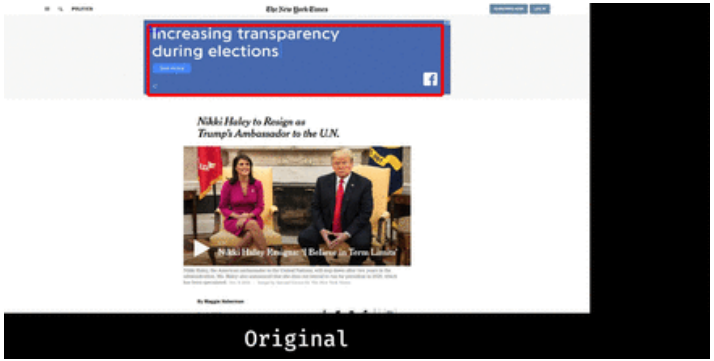


The Problem: Adversarial Examples



Biggio et al. 2014, Szegedy et al. 2014, Goodfellow et al. 2015, ...

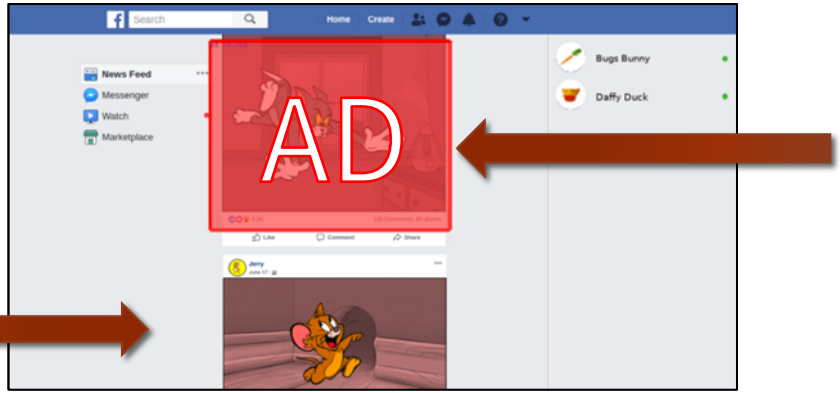
How Secure is Perceptual Ad-Blocking?



AdChoices AdChoices

How (in)-Secure is Perceptual Ad-Blocking?

Jerry uploads malicious content …



… so that Tom's post gets blocked

Attacking Perceptual Ad-Blocking

How? Adversarial Examples (aka gradient descent)

- *Nothing too special here*

Why? Ad-blocking is the perfect threat model for adversarial examples

- *This is the cool part!*

The Adversarial Examples Threat Model

1. (There's an adversary)

2. Adv. **cannot change the distribution** of inputs
   - *Otherwise, Adv could just use a "test-set attack" (Gilmer et al. 2018)*

3. Adv. can only use **"small" perturbations**
   - *Otherwise, Adv could just change the class semantics*

4. Adv. has access to model weights or query API

The Adversarial Examples Threat Model

1. There's an adversary
2. Adv. **cannot change the distribution** of inputs
3. Adv. can only use **"small" perturbations**
4. Adv. has **access to model** weights or query API

Challenge: find a setting where this threat model is realistic

The Ad-Blocking Threat Model

1. There's an adversary
   - *Web publishers, ad-networks have financial incentive to evade ad-blocking*
2. Adv. **cannot change the distribution** of inputs
   - *Ad campaigns are meticulously designed to maximize user engagement*
3. Adv. can only use **"small" perturbations**
   - *Website users should be unaffected and still click on ads!*
4. Adv. has **access to model** weights or query API
   - *Ad-blocker is run client-side so the model weights are public*

New challenge: find a setting **other than ad-blocking** where this threat model is realistic

Client-Side Web-Security is Hard

Near-impossible to resist *dynamic/adaptive* attacks

True beyond ad-blocking:

- *Don't do client-side visual anti-phishing!*

True beyond computer vision:

- *Don't use client-side ML models to detect spam or malware*

So What Can We Do?

1. Client-side black-lists:
   - *Signatures of known malware*
   - *List of known phishing domains (e.g., Google safe browsing)*
   - *Ad-blocking filter lists*

2. Server-side ML:
   - *Real-time spam & malware detection*
   - *Content takedown*
   - *What about computer-vision?*

+ Efficiency
+ More features
+ “Security by obscurity”

Act II

Computer Vision In **Server-Side** Web Security:
A Privacy Nightmare

The Problem

Server-side ML

=

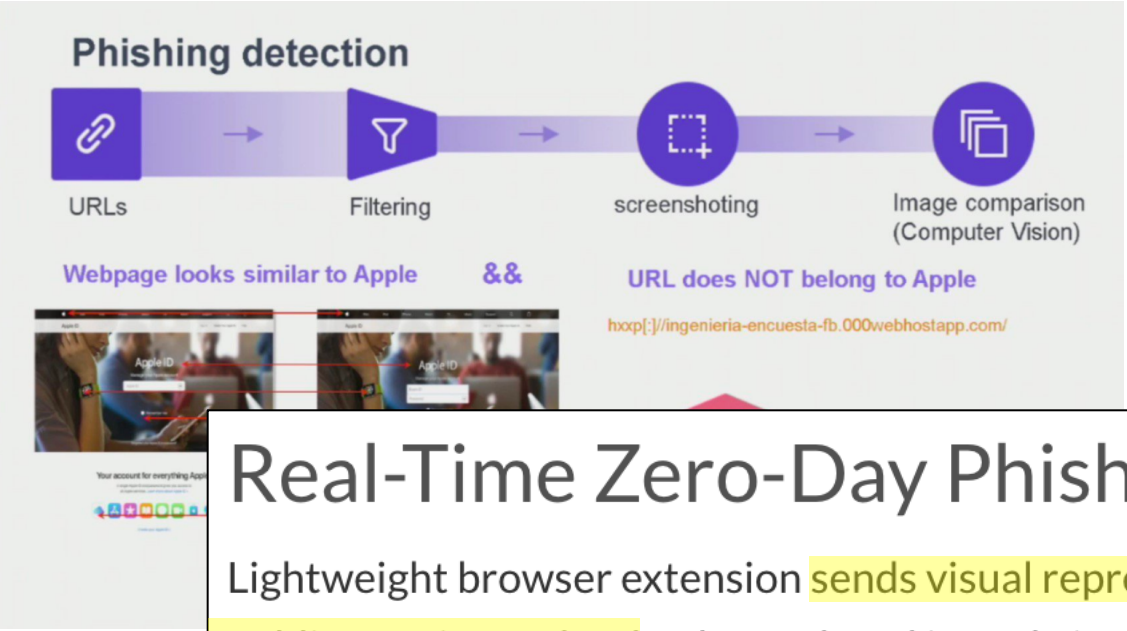
Server-side Data

Privacy vs Security: Choose One

Does content-security warrant sharing our...

- Emails?
  - > *It seems so*
- Downloaded apps?
  - > *Google / Apple / ... already know this anyway*
- Website screenshots for ad-blocking or anti-phishing?
  - > *That seems excessive...*

Screenshot Sharing For Security is a Thing!



Real-Time Zero-Day Phishing Prevention

Lightweight browser extension sends visual representation to PhishProtect™ public or private cloud to be analyzed in real-time and block phishing sites immediately.

source: https://www.phish.ai/

Some Research Questions

Is visual anti-phishing secure?

- *Can computer vision achieve low-enough false positives?*
- *Do phishing websites have to look similar to legitimate websites?*
- *Automated black-box attacks?*

Is it private?

- *Can browser extensions be tricked into screenshotting sensitive data?*
- *Can this data be extracted from trained neural nets?*

Conclusion

1. Don't Use ~~Computer Vision~~
   Machine Learning
   For Client-Side Web Security

"In fact, it's better if you don't use ML at all"

2. Don't collect screenshots from my browser!

⇒ **Don't Use Computer Vision For Web Security**

Questions? tramer@cs.stanford.edu