On the Susceptibility to Adversarial Examples Under Real-world Constraints

Lujo Bauer

Professor, Electrical & Computer Engineering + Computer Science

With: Keane Lucas, Clement Fung, Weiran Lin, Mahmood Sharif, Mike Reiter (UNC), …

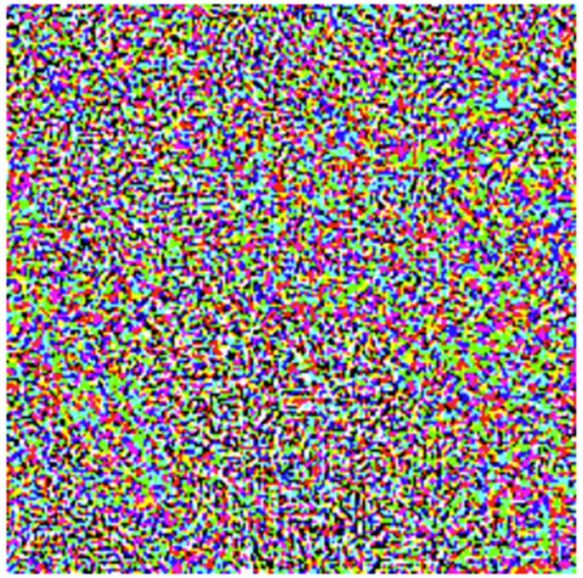
September 2020

Attacks and Defenses for *Practical* Uses of ML

- Face recognition (previous but very cool)
- Malware detection (ongoing; some updates)
- Anomaly detection in industrial control systems (new)

ML Algorithms Are Fragile

+ 0.007x



=

"Panda"

"Gibbon"

[Sharif, Bhagavatula, Bauer, Reiter CCS '16, arXiv '17, TOPS '19]

Can *an Attacker* Fool ML Classifiers?

Fooling face recognition (e.g., for surveillance, access control)

What is the attack scenario?

Does scenario have constraints…

… on how attacker can manipulate input?

… on what the changed input can look like?

Can change physical objects, not pixels

Can't control camera position, lighting

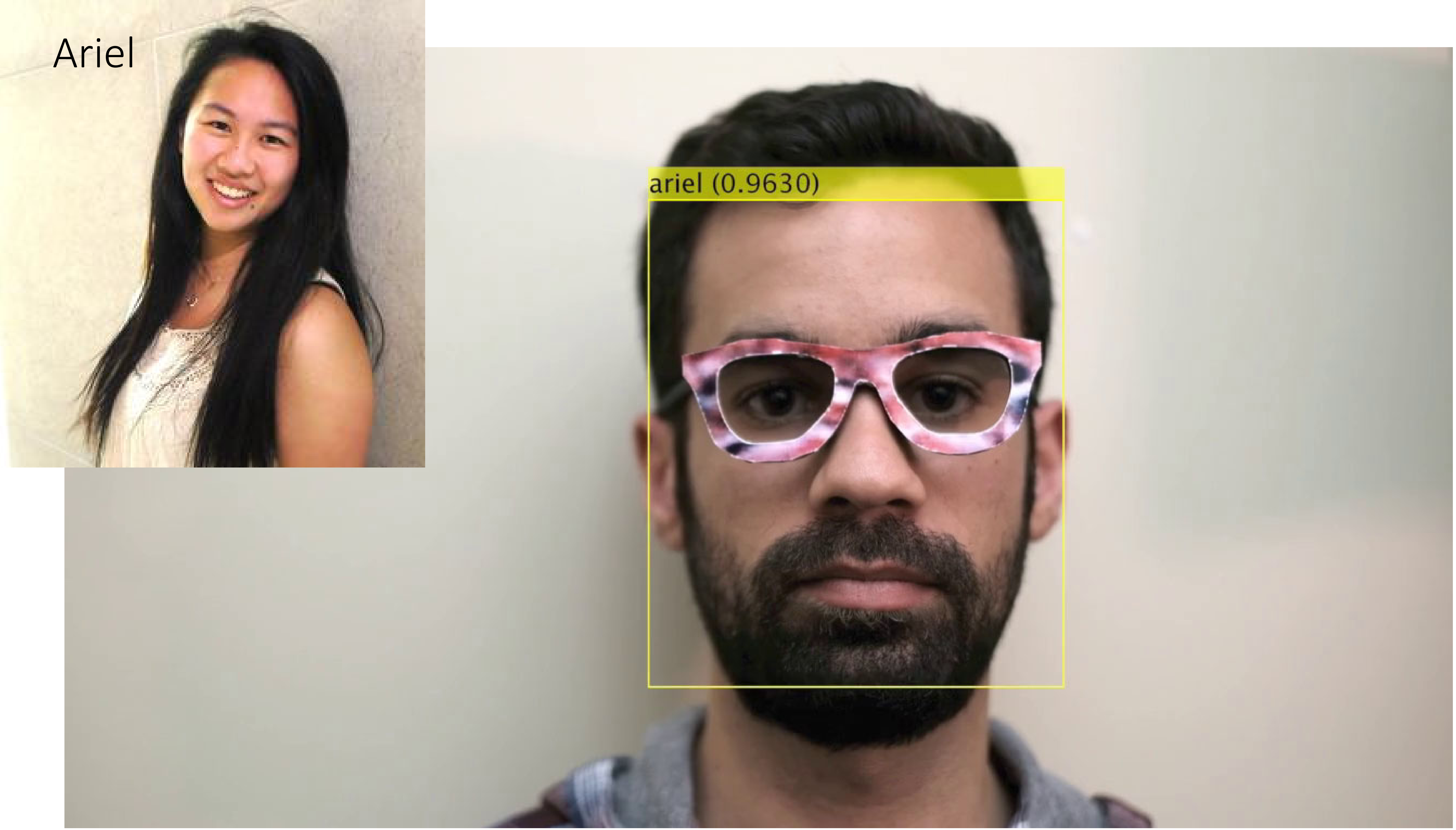
Defender / beholder doesn't notice attack
(as measured by a user study)

Fooling Face Recognition Classifiers x2

1. Traditional gradient descent, augmented to account for:
   - Changing pixels only on eyeglasses
   - Smooth pixel transitions
   - Restricting changes to printable colors
   - Classification over multiple images of attacker

OR

2. Train adversarial eyeglass *generator*
   1. Train eyeglass generator
   2. Additionally train to generate adversarial eyeglasses


Ariel
ariel (0.9630)

[Sharif, Lucas, Bauer, Reiter, Shintre arXiv '19]

Can *an Attacker* Fool ML Classifiers?

Face recognition

Attacker goal: evade surveillance, fool access-control mechanism

Input: image of face

Constraints:

- Can't precisely control camera angle, lighting, pose, …
- Attack must be *inconspicuous*

Malware detection

Attacker goal: bypass malware detection system

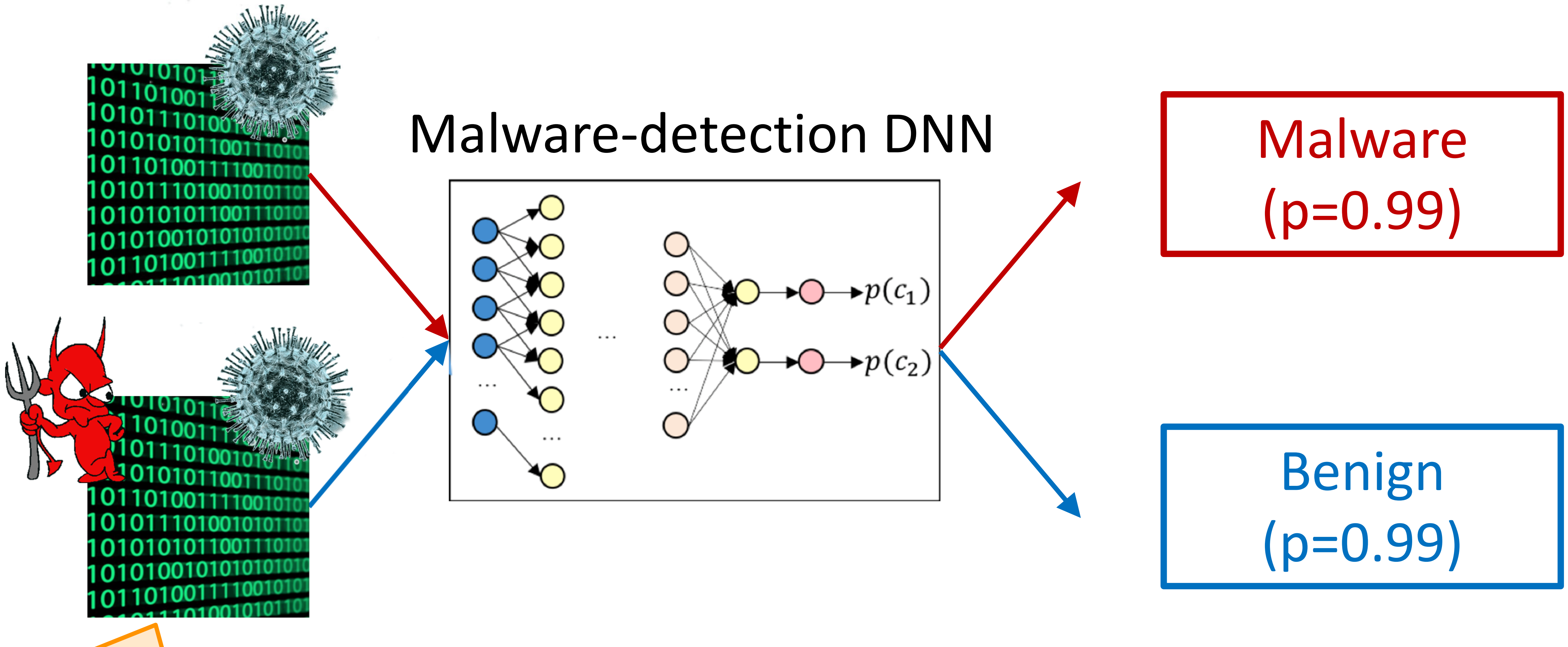
Input: executable in binary format

Constraints:

- Must be functional malware
- Changes to executable must not be easy to remove

Very different constraints! ⇒ Attack method does not carry

Hypothetical Attack on Malware Detection



1. Must be functional malware
2. Changes to binary must not be easy to remove

Attack Building Block: Binary Diversification

- Originally proposed to mitigate return-oriented programming [3,4]
- Uses transformations that preserve functionality:
  1. Substitution of equivalent instruction
  2. Reordering instructions
  3. Register-preserving (push and pop) randomization
  4. Reassignment of registers
  5. Displace code to a new section
  6. Add semantic nops

In-place randomization (IPR): items 1–4

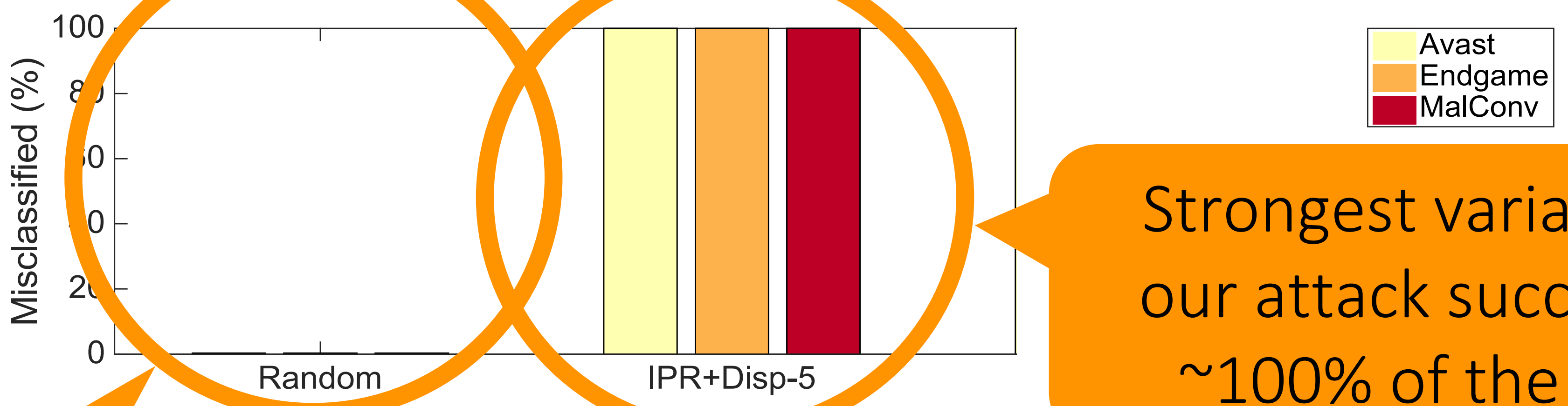
Displacement (Disp): items 5–6

[3] Koo and Polychronakis, “Juggling the Gadgets.” AsiaCCS ’16
[4] Pappas et al., “Smashing the Gadgets.” IEEE S&P ’12

Transforming Malware to Evade Detection

Experiment: 100 malicious binaries, 3 malware detectors (80-92% TPR)

Success rate (success = malicious binary classified as benign):



Strongest variant of our attack succeeds ~100% of the time

Tran applied at random don’t work

Success rate for 68 commercial anti viruses (black-box):

Up to ~50% of AVs classify transformed malicious binary as benign

Can *an Attacker* Fool ML Classifiers?

Face recognition

Attacker goal: evade surveillance, fool access-control mechanism

Input: image of face

Constraints:

- Can’t precisely control camera angle, lighting, pose, …
- Attack must be *inconspicuous*

Malware detection

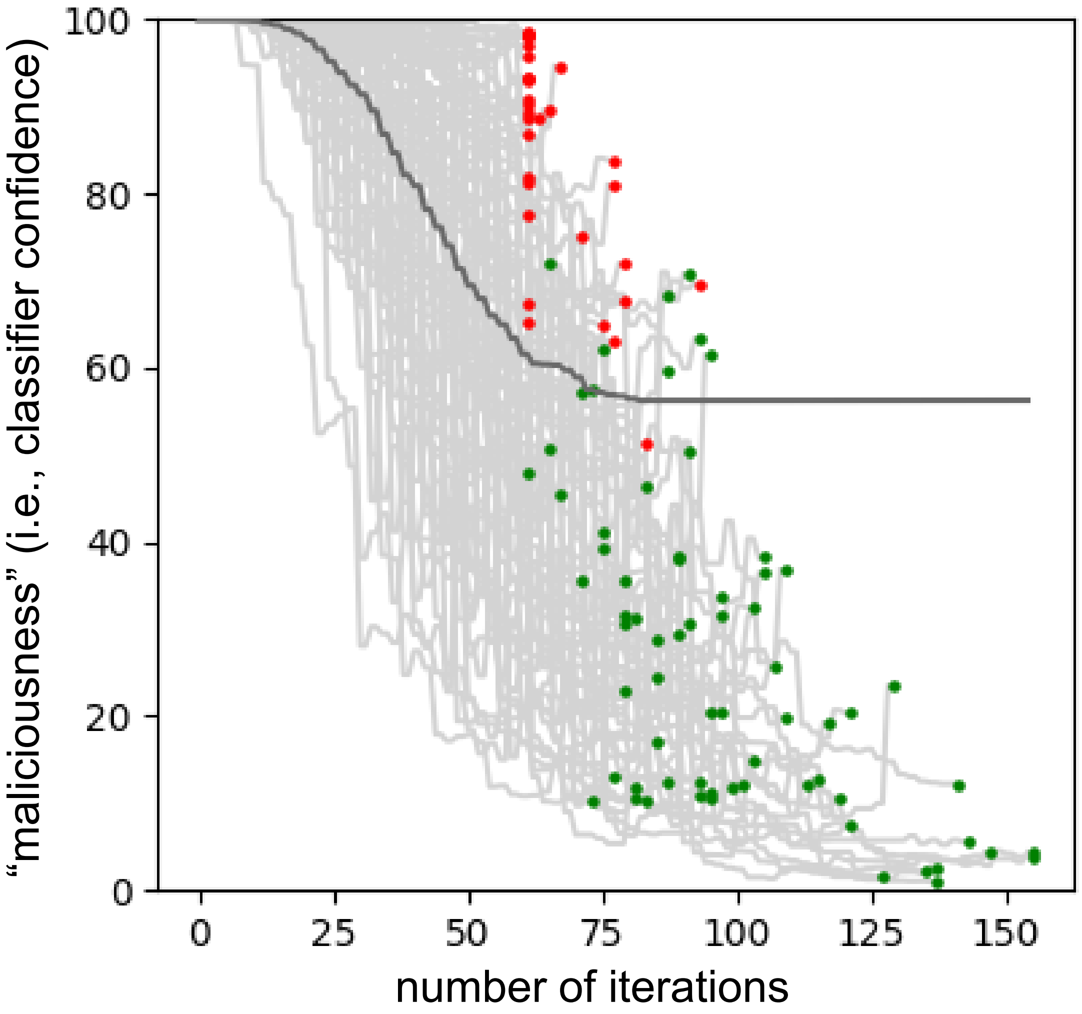
Attacker goal: bypass malware detection system

Input: malware binary

Constraints:

- Must be functional malware
- Changes to binary must not be easy to remove

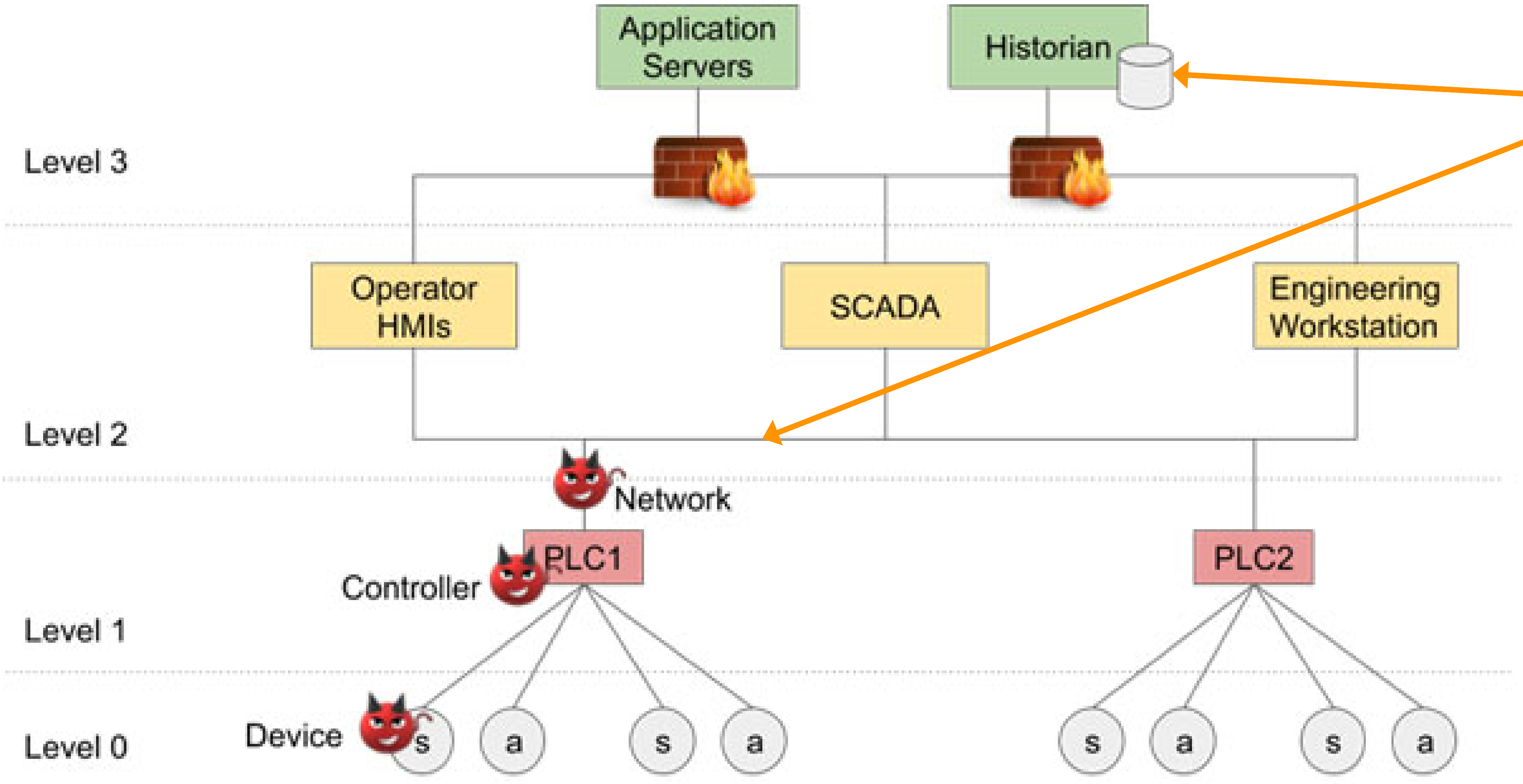
Can an Attacker Do Even Better?



Unfortunately, yes!

Natural question: can we *learn* what distinguishes more successful attack attempts from less successful ones?

ML-based Anomaly Detection in Industrial Control Systems



Train ML to detect anomalies

But… in ICS the Cost of Errors Is Very High

- Shutdown because of detected anomaly can take hours or days to reverse
- Hence: explanations are critical!
  - For both the benign case and the adversarial case
  - Operator needs explanation before reacting to detected anomaly
- On-going work:
  adapt approaches to explaining AI decisions to non-image, time-series data

On the Susceptibility to Adversarial Examples Under Real-world Constraints

- Practical applications of machine learning may be susceptible to attack
- Defenses are on the way

Lujo Bauer
lbauer@cmu.edu