A Low-Cost Attack against the hCaptcha System

Md Imran Hossen and Xiali (Sharon) Hei

University of Louisiana at Lafayette

hCaptcha

- CAPTCHAs protect websites from bots, spam, and other forms of automated abuse

- hCaptcha is a relatively new **Image CAPTCHA** system developed by *Intuition Machines, Inc*

- Designed as a drop-in replacement for Google's reCAPTCHA[1]

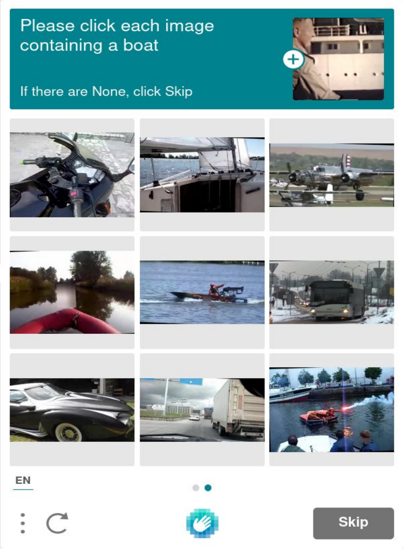

Please click each image containing a boat
If there are None, click Skip
EN
Skip

Fig. 1. hCaptcha challenge

[1] https://www.google.com/recaptcha/about/

Contributions

- We designed and developed a **low-cost**, end-to-end system to break hCaptcha service
- We evaluated the system against 270 live hCaptcha challenges and achieved a **success rate of attack of over 95%** in less than **19 seconds** on average
- We conducted a preliminary security analysis of the hCaptcha system, revealing weaknesses of the CAPTCHA system against automated abuse

Threat Model

- Our threat model involves an attacker with **limited resources**

- We will assume the attacker is limited to
  - One computer with a small-size RAM
  - One IP address

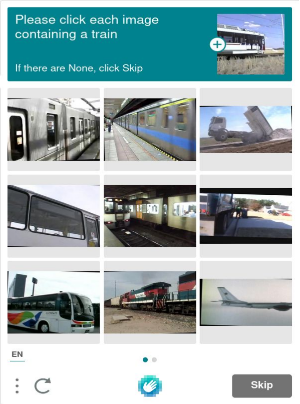
- We will aim for an **accuracy benchmark above 50%**

System Overview

1) Obtaining the challenge

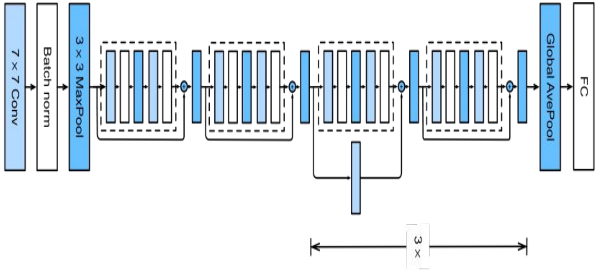
2) Solving the challenge

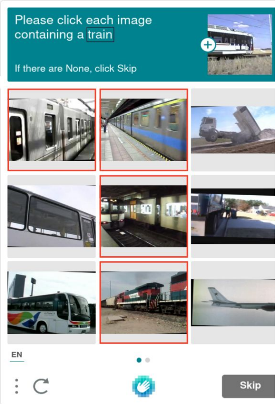
3) Submitting and verifying the solution



Please click each image containing a train

If there are None, click Skip





Please click each image containing a train

If there are None, click Skip

✓

✗

Attack Evaluation

Implementation and Evaluation Platform

- The **puppeteer-firefox** framework with Firefox web browser was used for **browser automation**

- **ResNet-18** as the **image classifier** network built on top of PyTorch
  - Pretrained on ImageNet
  - Fine-tuned further on 45000 images from 9 classes extracted from the OpenImages dataset

- Experiments were **run inside a docker instance**
  - Running Ubuntu 20.04 image configured to use only **2GB Memory** and **3 CPU cores** from the physical (host) machine

- Experimental Setting:
  - A regular (non-academic) IP address
  - Caches and cookies were cleared during each run

Frequently Appeared Image Classes

- **5000** hCaptcha challenges were collected from 3 websites during the period of May 2020 to July 2020

- Only **9 image categories/classes** were observed

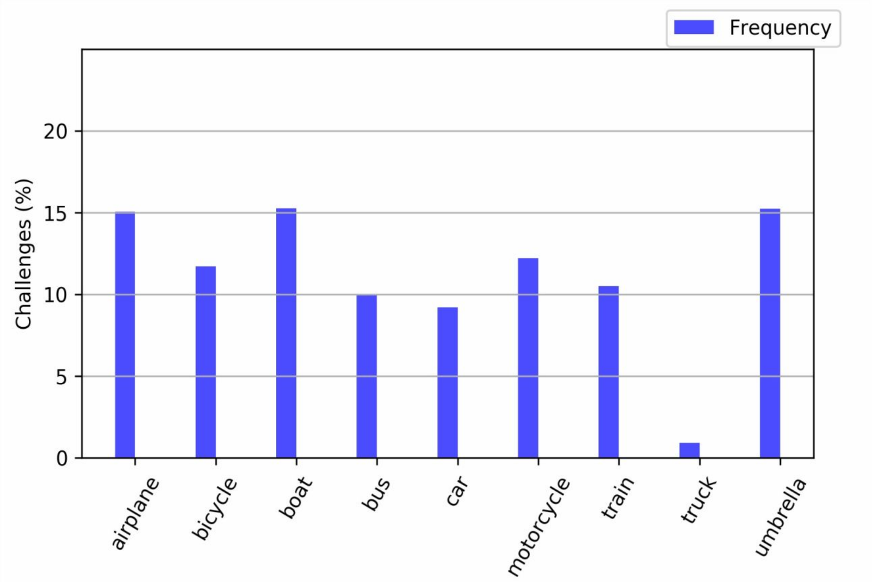


Fig. 2. The frequency of each image category appears in collected challenges.

Accuracy and Speed of Attack

- The number of challenges **attempted:** 270
- The number of challenges successfully **solved**: 259
- **Attack accuracy:** 95.93%
- **Avg. speed of attack:** 18.76 seconds

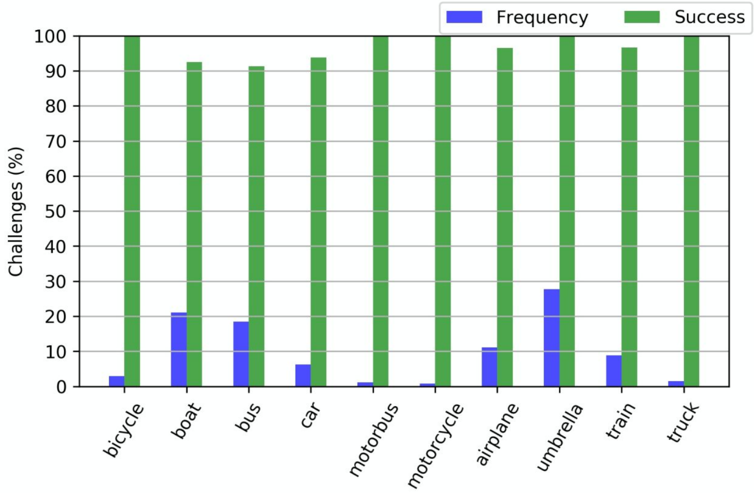


Fig. 3. The accuracy and frequency of each image category in the solved challenges.

Accuracy and Speed of Attack (cont'd)

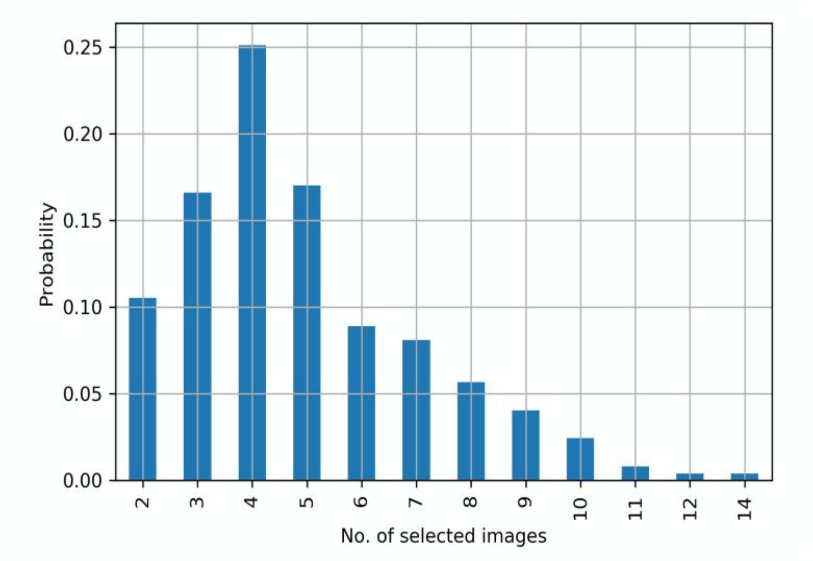


Fig. 4. The probability distribution of no. of images selected per challenge.

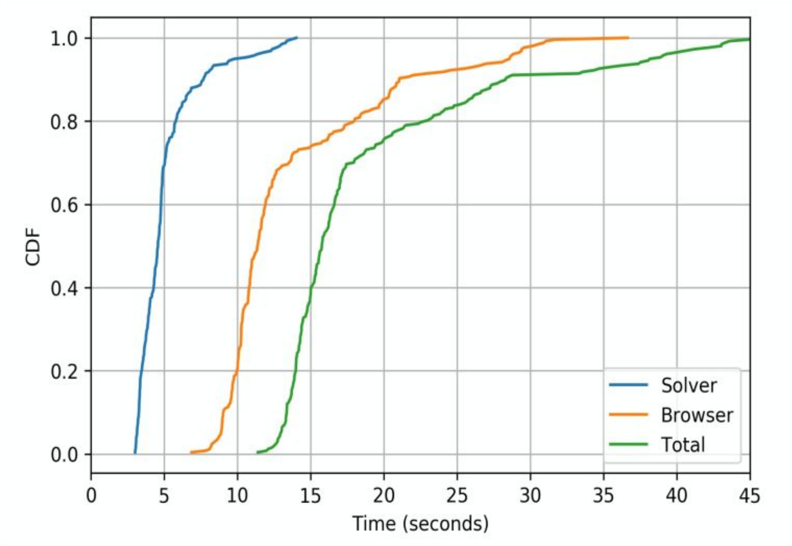


Fig. 5. Cumulative distribution of time required by each module.

Influence of IP Addresses

- **An academic, a VPN**, and **a Tor network** IP address were used for testing
- 200 attempts to solve hCaptcha challenges from each IP address were sent separately with 20-second gaps between each attempt
- Similar attack success rates (**over 95%**) were achieved

Adaptability

- Both the **Selenium** and **Puppeteer** browser automation framework used
- Different experimental settings (e.g., setting the browser in **headless** mode, using various **window.navigator** properties) tested
- **No discrepancies observed**
- **No blocking encountered**
- Achieved **over 90% accuracy of attack** across all settings

Blocking

- hCaptcha allows website owners to adjust the **difficulty levels** for the served CAPTCHAs on their websites
- It supports 4 difficulty levels: *easy*, *moderate* (default), *difficult*, and *always on*
- The blocking was tested on *moderate* and *difficult* difficulty levels by attempting to solve 400 challenges for each of them
- All the requests to our web application were sent in a row with only a 1-second delay between two subsequent requests
- Only **17** of our attempts (out of the total 800 combined) were blocked with the message — **“Rate limited or network error. Please retry”**
- **Accuracy of attacks:** 92.25% and 88.5%

Blocking (cont'd)

- We also attempted to **trigger blocking deliberately** by sending too many requests simultaneously
- **50 instances** of our bot program were run concurrently against our hCaptcha-enabled web page for **10 times** with a 2-second delay between two subsequent iterations
- This time, the hCaptcha system blocked many of our requests with the warning message — **"Your computer or network has sent too many requests"**
- The number of blockages for the 10 iterations are **24,** 40, **48**, 29, 28, 26, 26, 29, 30, and 28.

Image Repetition

- **48330** images were used for analysis
- Both the ***MD5*** and ***perceptual (pHash)*** hashing algorithms were used
- **Both algorithms yielded the same results**
  - **9854** redundant images belonging to **1985** sets of identical images were found

Online Attack

- We performed an online attack using 3 **vision APIs** for image recognition.
- **Google Cloud Vision**, **Microsoft Computer Vision**, and **Amazon Rekognition**.

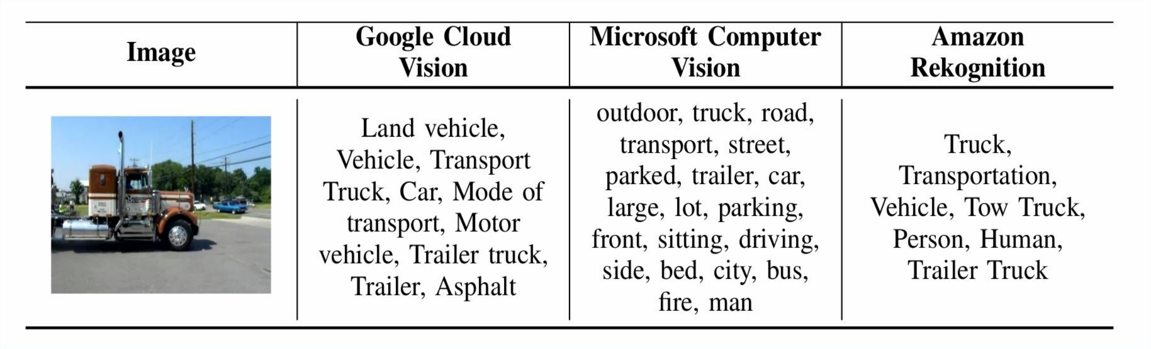
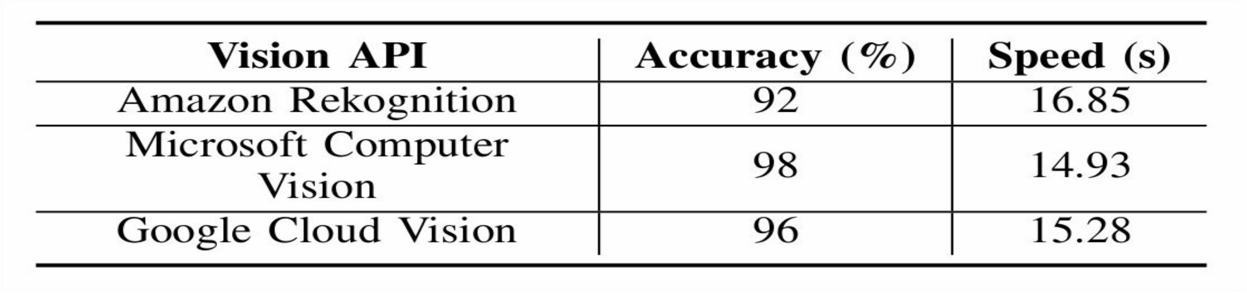


Image	Google Cloud Vision	Microsoft Computer Vision	Amazon Rekognition
	Land vehicle, Vehicle, Transport Truck, Car, Mode of transport, Motor vehicle, Trailer truck, Trailer, Asphalt	outdoor, truck, road, transport, street, parked, trailer, car, large, lot, parking, front, sitting, driving, side, bed, city, bus, fire, man	Truck, Transportation, Vehicle, Tow Truck, Person, Human, Trailer Truck

Fig. 6. List of labels returned by three image recognition APIs for a sample image from hCaptcha challenge

Online Attack (cont’d)



Vision API	Accuracy (%)	Speed (s)
Amazon Rekognition	92	16.85
Microsoft Computer Vision	98	14.93
Google Cloud Vision	96	15.28

Table 1. **Attack performance of off-the-shelf vision APIs.**

Countermeasures

- **Use broader image categories**
  - Expanding the image categories will make the data collection process relatively challenging
- **Adversarial examples**
  - Can lower the attack accuracy by misleading deep neural networks
- **Resist web automation software**
  - Resisting requests originating from widely used web automation frameworks will likely lower attackers’ success rates
- **Commonsense knowledge**
  - Machines usually perform poorly involving a task that requires higher-order reasoning

Conclusion and Future Work

- hCaptcha challenges could be solved automatically with high accuracy using deep learning-based methods
- Even a low-resource adversary can mount a powerful attack using our method
- The CAPTCHA system lacks other stringent security requirements making it highly vulnerable to automated abuse
- In the future, we want to test our methodology on other similar Image CAPTCHA systems

Thanks for listening!

Questions?

Md Imran Hossen, md-imran.hossen1@louisiana.edu

Xiali (Sharon) Hei, xiali.hei@louisiana.edu