



# Adding ReCAPTCHA to Self-Service Admissions



Admissions Fraud Prevention

BUGMI

June 22, 2018

## Introductions

- Anita Lees, Application Systems Analyst
- ITS-ES Banner Support team

## About LCC

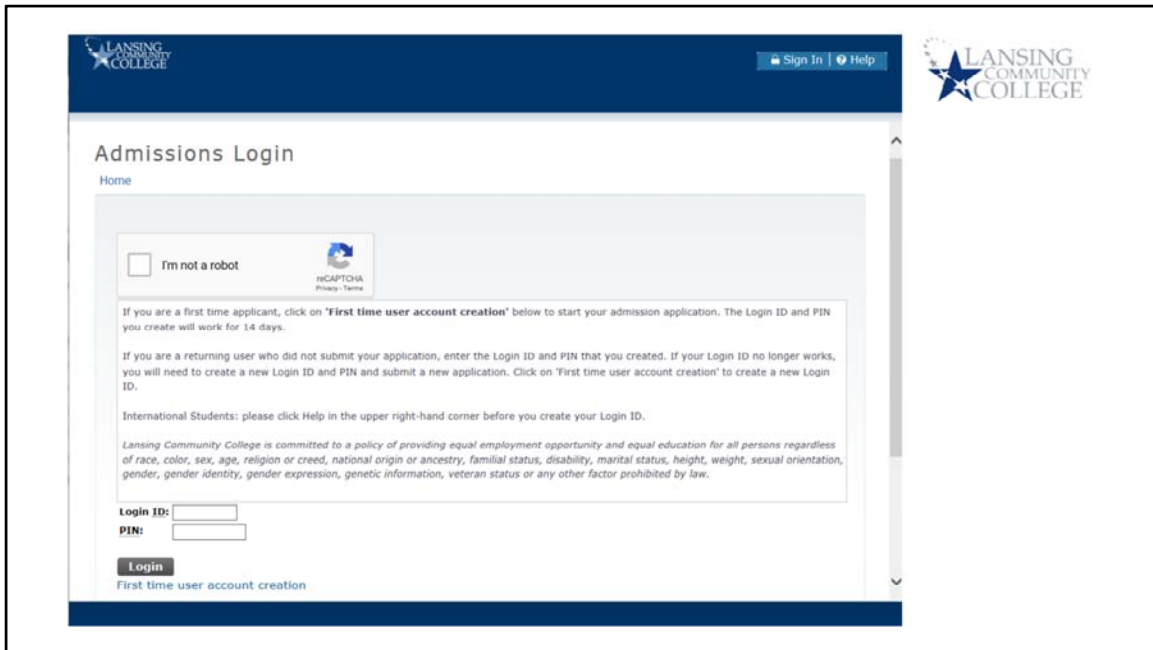
---



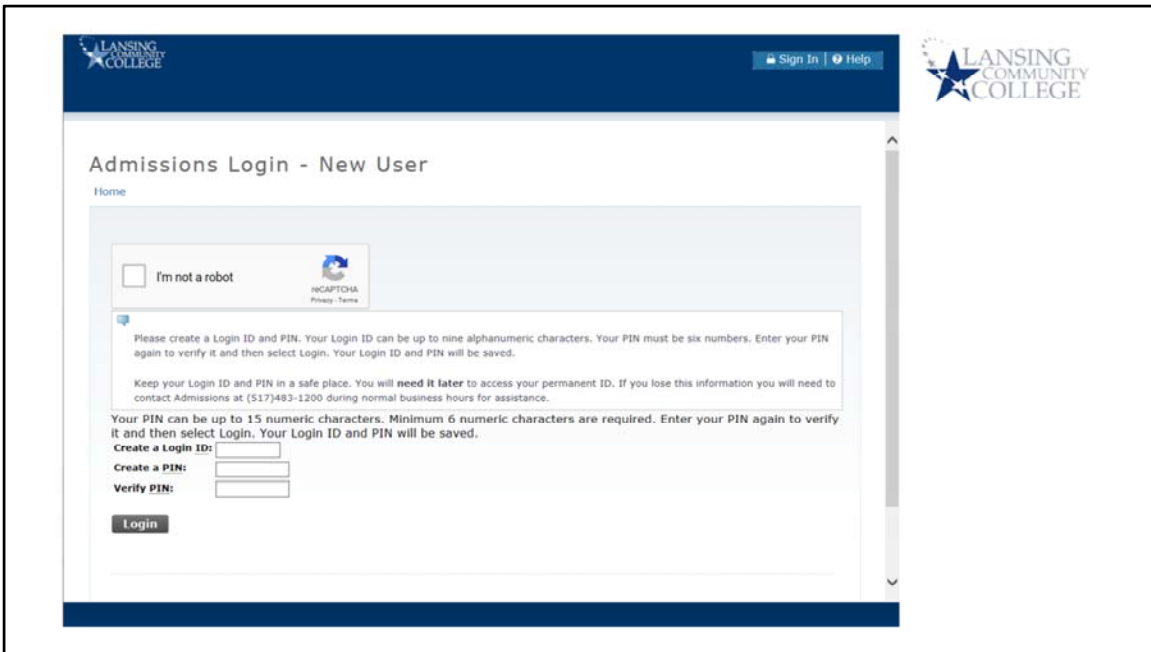
- 4<sup>th</sup> Largest Community College in Michigan
  - Close behind Macomb
  - According to Community College Review
- Quick Admit
  - We quick admit anyone over 18
  - Not reviewed unless put into Suspense
- Free email account (Google)

The most attractive things about LCC to fraudsters are the quick admit and the free email account.

- Financial Aid fraud
- Selling email addresses



This is our Self-Service Admissions page showing the ReCAPTCHA. Note that we have this on the Login page, and also on the new user page, at the link. [https://starnetb.lcc.edu/LCCB/bwskalogs.P\\_DisLoginNon](https://starnetb.lcc.edu/LCCB/bwskalogs.P_DisLoginNon)  
Some confusion – many people click on both pages, even if “First time user”.



This is the main (arguably, the only) place the human verification needs to be.

## What is ReCAPTCHA?

---

- History
  - CAPTCHA
    - = Completely Automated Public Turing test to tell Computers and Humans Apart
  - Distorted Text/Audio
  - Accessibility Issues
- ReCAPTCHA
  - Checkbox
  - Works with Screen Readers

A sort of Turing test – but “Reverse Turing” because it’s administered by the computer

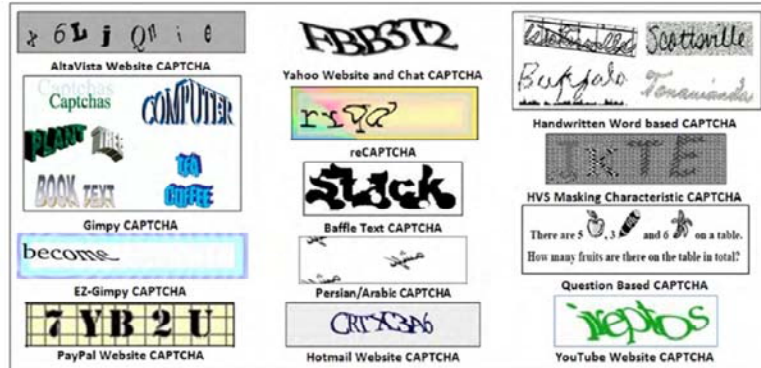
Slogan “Easy for Humans, Hard on Bots” but CAPTCHA was difficult for humans – avg.

10 seconds for human – easy to hack in various ways

1997 Anti-OCR

ReCAPTCHA v1 was used to digitize difficult to scan books

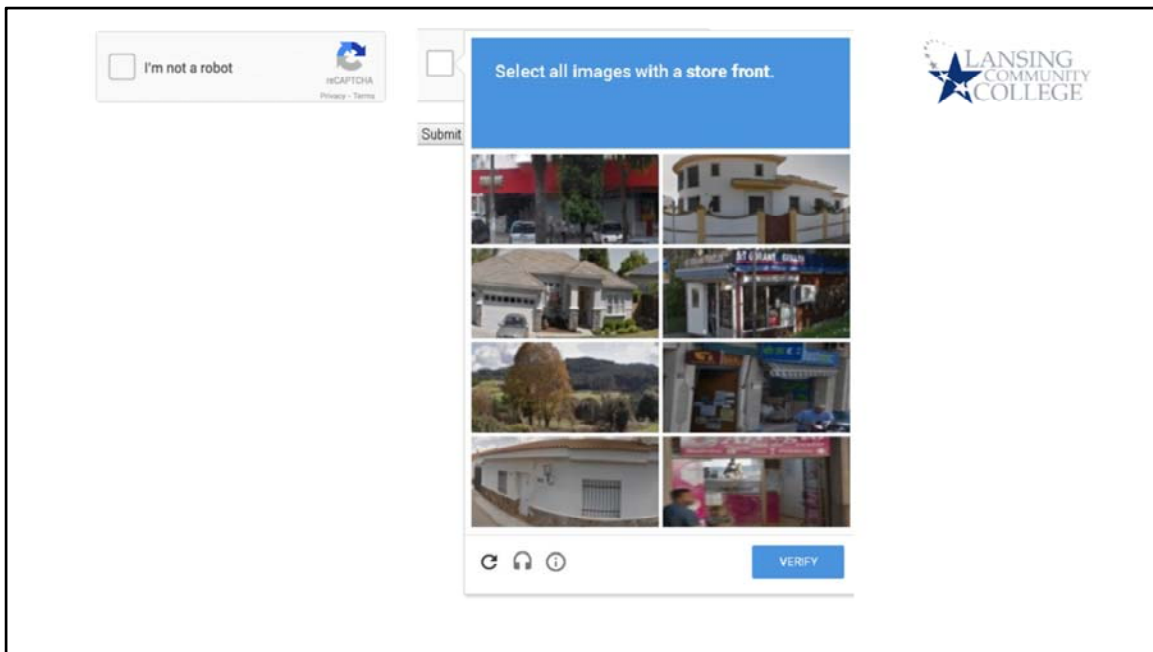
ReCAPTCHA v2 Aria Messages



Some historic CAPTCHA images – before ReCAPTCHA, before Google’s use to digitize texts



reCAPTCHA which became a de facto standard  
Here we see it's much more clear than some of the previous incarnations  
Hackable by session ID, mechanical Turk, image processing



The UI:

1. The user is presented with a checkbox
2. The user "checks the box"
  - Clicks the box
  - Google server makes a decision, and displays checkbox is successful
3. The user may need to check images based on some stated criterion.

Note the Refresh, Audio, and information.



## The Problem

---

- Fake applications were appearing on the Residency Report
  - First noticed prior to start of Fall 2017
- Concern about Financial Aid fraud
  - Caught early – inactivated students

We have had a history of Financial Aid fraud attempts, so that's what we suspected. It may be the original intent, but we found some indications that the lists of email addresses were being sold; perhaps when the users were not able to register.

## The Solution

---

- First ran script to inactivate students
- Reg Holds on Out of State Students – require ID
- Blocked Foreign IPs
- Add ReCAPTCHA to the Admissions Application
  - Anonymous and Authenticated Application pages

First we did queries to identify the students and ran scripts to inactivate them, so they couldn't register.

It became clear that someone was running scripts, but adapting to what we were doing.

If JavaScript is disabled, it can't work so API returns failure result, but a success is required in the submitted page.

## Implementation

---

- PL/SQL Function
  - Call Google API
- Web Tailor – interim, didn't work
- Baseline Mods
  - BANINST1. bwskalog.P\_DisLoginNon
  - BANINST1. bwskalog.P\_DisLoginNew

First I tried putting the JavaScript with callbacks into Web Tailor but it wasn't working. Also, there were some concerns that scripting seemed to bypass the JavaScript this way.

## Testing and Trouble Shooting

---

- Testing
  - PL/SQL Anonymous Block
    - Call function with old result, should get expired result message
    - Check “I’m not a robot” box on page while inspecting elements
  - Use of jMeter to verify that script does not bypass the verification.
- Trouble Shooting
  - Similar to testing
  - May need to look over logs on the server for the actual errors

Need to work closely with the server team, if you don’t have access to logs. While testing, you can log into a known application or attempt to create an application with an already used web ID. We did have an issue in which one particular Google server was always rejecting the result; the fix was to set etc/hosts for google.com to a server which was cooperating. Later we had to update a certificate which had expired. This may happen again at some point.

## Future

---

- Invisible ReCAPTCHA
  - Tied to a button e.g. Login
  - User is unaware of the verification
- Impact of AI?
  - Easier to crack security including ReCAPTCHA
- Meanwhile...

It is currently possible to crack most reCAPTCHAs using tools, but so far difficult for scripts. AI is expected to make it easier to get past reCAPTCHA, so what will happen in the future? Meanwhile, we keep doing what's working.