CAPTCHAs (also known as Reverse Turing Tests) and Password Security

Michael Peck

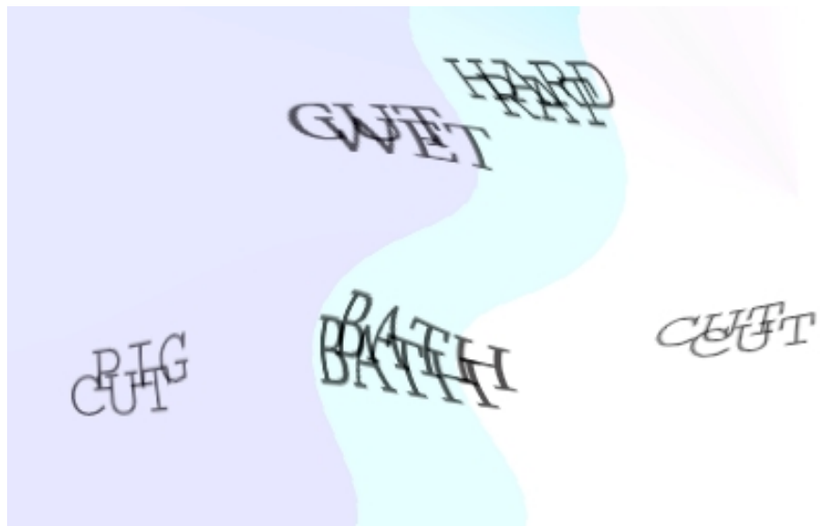
January 27, 2004

Captcha uses

- "completely automated public Turing test to tell computers and humans apart"
  - *public*: Details of algorithm used to generate the tests are publicly available.
- Mostly used for spam-fighting related purposes
  - Prevent automated email account creation: Hotmail, Yahoo! Mail.
  - Make sender of email confirm he or she is a human before the message is delivered to its recipient.
- Many other uses, including:
  - prevent automated attacks on online polls
  - prevent ticket hoarding at Ticketmaster

Examples

from Yahoo! Mail new account signup: uses EZ-Gimpy



Gimpy (www.captcha.net):
Type in any 3 of the words above.
Uses an 850 word dictionary.

Gimpy has been broken:
Greg Mori and Jitendra Malik (UC Berkeley Computer Vision Group): ***Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA,*** CVPR 2003.
92% success against EZ-Gimpy, 33% against Gimpy.

Captcha Disadvantages

- Accessibility issues
  - Visually impaired people
    - Audio captchas now exist, but are they being used in practice?
  - Text-only users
    - Lynx users
    - Cell phone WAP browser users
- Usability/inconvenience issues
  - Forcing customers to answer a captcha may drive some of them to a competitor

Disadvantages

- Computers exist to make things easier for people - captchas seem to contradict this - seems to be a tradeoff.

- Simson Garfinkel: (MIT Tech Review, June 2003)

These tests are the devil. If widely deployed, they will waste our time and confound us—without solving their intended problems.

  - Determined attackers could outsource captcha answering to low-wage workers in third world countries.

  - Or, set up a porn site, and make visitors answer a captcha every few minutes to continue receiving access.

Password Security

- UNIX, in the past:
  - /etc/passwd
    - World readable, but passwords encrypted (one-way) with 56-bit DES (UNIX crypt())
    - Only two ways known to attack crypt: brute force attack and dictionary attack. Dictionary attacks can be quite effective.

Improvements

- /etc/shadow
  - Only readable by root
- md5 hash instead of DES

Offline attacks a thing of the past?

- Can do a dictionary attack at your leisure, offline, with a copy of the hashed passwords.
- Otherwise, need to do attack *online*
  - login attempts more easily detected

Defenses against online attack

- Delayed response
  - When username/password is entered, make system pause briefly before continuing.
    - Attack: Just perform login attempts in parallel.
      - If machines are sharing the same password file, this is even easier.
      - Or, traverse through all of the usernames, only using each username once. Easy to get a list of users.
- Account locking
  - After x unsuccessful login attempts, lock the account.
    - This approach asks for trouble: Yahoo! auctions example from Pinkas and Sander.

CAPTCHAs for Password Security

- Defend against automated, on-line dictionary attacks.
- Basic protocol:
  - User must correctly answer a CAPTCHA before being allowed to enter a username & password.
- Usability and scalability problems

Improved protocol

- Store a cookie on user's computer the first time the test is answered correctly. Let the user bypass the test if the cookie is present.

- 1. User enters name & password.
- 2. System verifies name/password.
- 3. If correct,
  - a. If user has correct cookie, grant access.
  - b. Otherwise, server gives user a test, grants access if answered correctly.
- 4. If incorrect,
  - With probability p, give user a test.
  - Otherwise, immediately deny access.
  - Choice is deterministic to the username/password pair.
  - Can just make p = 1.

Usability & Scalability

- Can make the generated test deterministic of the username and password pair provided by the user.
- Valid user will receive the same exact test every time (and will still only have to answer the test rarely).

Security analysis

- Attack against the CAPTCHA:
  - Defense: increase p value in protocol
    - or, increase p only for the accounts that are being targeted.
- Cookie theft:
  - Stop accepting stolen cookie once it's detected.

Password choice

- Most systems today highly regulate the passwords that users can choose.

- From UVa ITC: http://www.itc.virginia.edu/helpdesk/accounts/passwords.html

  - To protect your files, most UVa password systems only accept new passwords that conform to the following rules:

    - Must be at least 6 characters long.
    - Must not consist of all lowercase, or all uppercase characters, all digits, or all punctuation characters.
    - Must not be part of the local computer's name.
    - Must not match anything in your UNIX account information, such as your login name or an item from your "finger" data entry (full name, login shell, home directory).
    - Must not be in the system's spelling dictionary - unless it has some uppercase letters other than the first character. For example, "Explain" would be rejected but "exPlain" would be accepted.
    - Must not have more than 2 characters repeated in a row - thus "ABCaaa" would be rejected.

  - These rules will probably be expanded to be more stringent in the future.

- Should such restrictions still be around today?

References

- The CAPTCHA Project (at Carnegie Mellon University). www.captcha.net
- Simson Garfinkel. ***Excuse Me, Are You Human?*** MIT Technology Review, June 2003.
- Greg Mori and Jitendra Malik. ***Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA,*** CVPR 2003.
- M. Naor, ***Verification of a human in the loop, or Identification via the Turing test***. 1996.
- Benny Pinkas and Tomas Sander. ***Securing Passwords Against Dictionary Attacks.*** ACM CCS, November 2002.