CSE 484 / CSE M 584: Computer Security and Privacy

Mobile Platform Security
[continued]

Spring 2015

Franziska (Franzi) Roesner
franzi@cs.washington.edu

Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Roadmap

- Today: Finish mobile platform security
- Remaining topics:
  - 5/29: Usable security
  - 6/1: Anonymity
  - 6/3: Social engineering, physical security
  - 6/5: Side channels, emerging technologies

- Homework #3 – due 5pm on 5/29 (Friday)
- Lab #3 – due 5pm on 6/5 (next Friday)

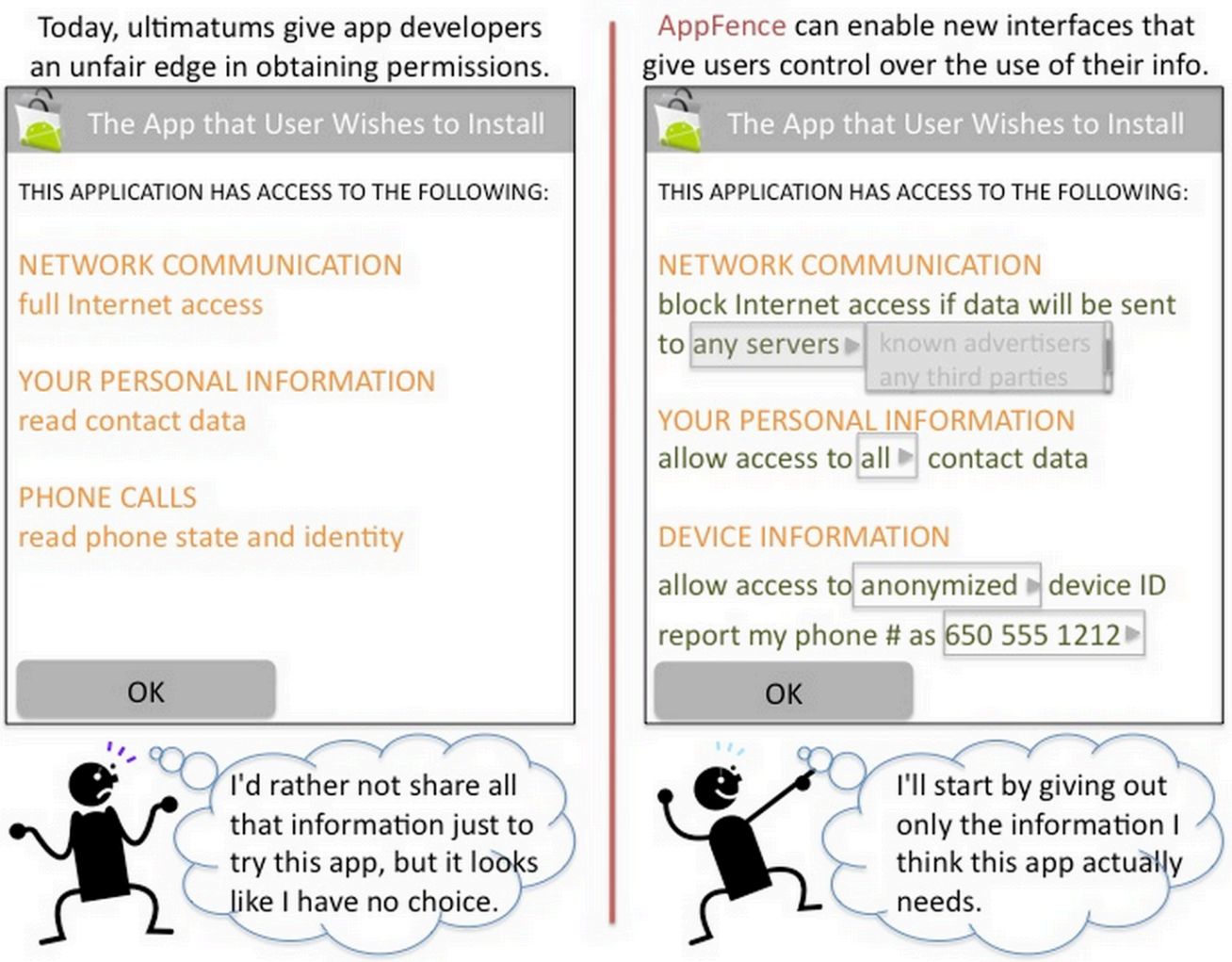
Reminder: Challenges with Isolated Apps

So mobile platforms isolate applications for security, but…

1. Permissions: How can applications access sensitive resources?
2. Communication: How can applications communicate with each other?

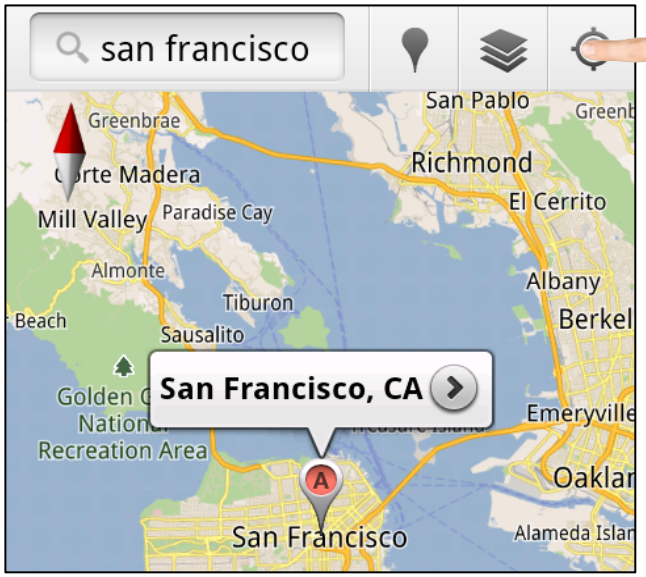
[Hornyack et al.]

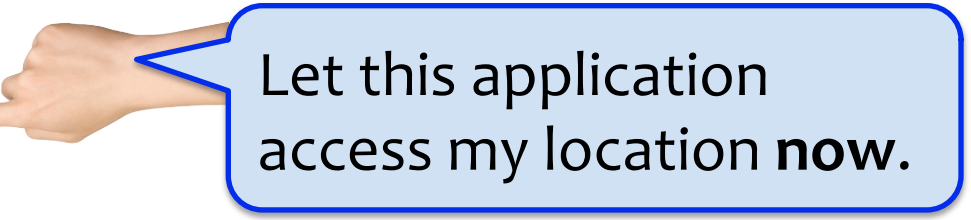
Improving Permissions: AppFence



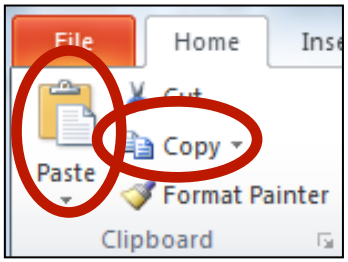
[our work]

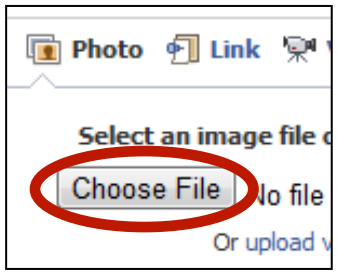
Improving Permissions: User-Driven Access Control



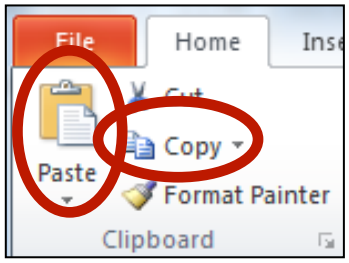


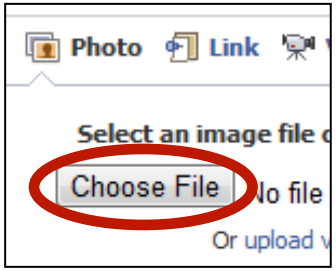
Insight:
A user's natural UI actions within an application implicitly carry permission-granting semantics.





Access Control Gadgets (ACGs)





- Special UI elements that carry permission-granting semantics: When user clicks, grant access.
- ACGs are owned by system and embedded by apps: need to secure them!
  - No clickjacking, no programmatic clicking, etc.

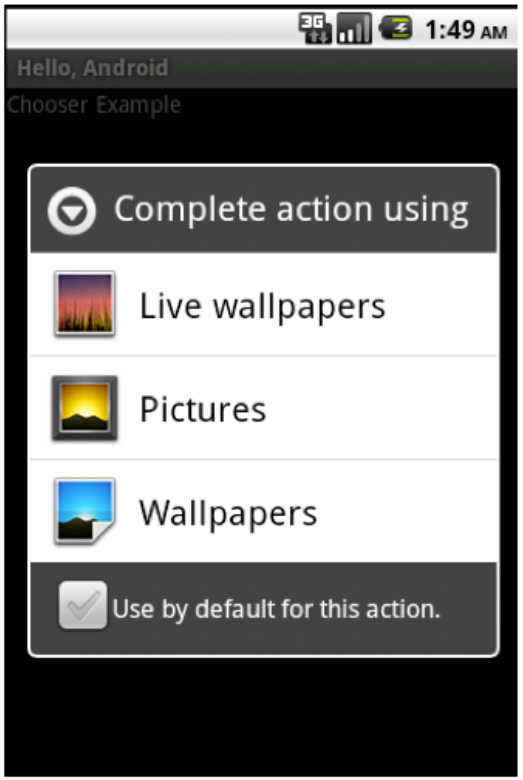
(2) Inter-Process Communication

- Primary mechanism in Android: Intents
  - Sent between application components
    - e.g., with `startActivity(intent)`
  - Explicit: specify component name
    - e.g., com.example.testApp.MainActivity
  - Implicit: specify action (e.g., ACTION_VIEW) and/or data (URI and MIME type)
    - Apps specify Intent Filters for their components.

[Chin et al.]

Unauthorized Intent Receipt

- Attack #1: Eavesdropping / Broadcast Thefts
  - Implicit intents make intra-app messages public.
- Attack #2: Activity Hijacking
  - May not always work:
- Attack #3: Service Hijacking
  - Android picks one at random upon conflict!



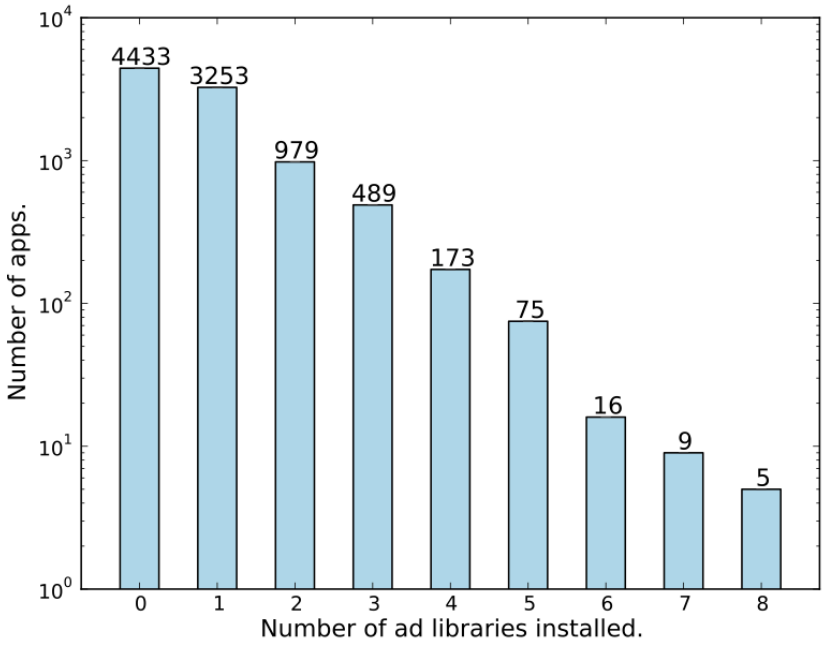
[Chin et al.]

Intent Spoofing

- Attack #1: General intent spoofing
  - Receiving implicit intents makes component public.
  - Allows data injection.
- Attack #2: System intent spoofing
  - Can't directly spoof, but victim apps often don't check specific "action" in intent.

Aside: Incomplete Isolation

Embedded UIs and libraries always run with the host application's permissions! (No same-origin policy here…)



[Shekhar et al.]

Ad from ad library

Map from Google library

Social button from Facebook library

More on Android…

Security-Enhanced Linux (SELinux)

- Added in Android 4.3 to strengthen app isolation
- Mandatory access control (MAC): central system authority makes all access control decisions
  - In addition to standard Linux discretionary access control (DAC), in which objects have owners that make access control decisions
  - Result: Even processes running as root can be limited by explicit policy (example: only system server should modify system files)

More details: https://source.android.com/devices/tech/security/selinux/

Android Application Signing

- Apps are signed
  - Often with self-signed certificates
  - Signed application certificate defines which user ID is associated with which applications
  - Different apps run under different UIDs
- Shared UID feature
  - Shared Application Sandbox possible, where two or more apps signed with same developer key can declare a shared UID in their manifest

Shared UIDs

- App 1: Requests GPS / camera access
- App 2: Requests Network capabilities

- Generally:
  - First app can't exfiltrate information
  - Second app can't exfiltrate anything interesting
- With Shared UIDs (signed with same private key)
  - Permissions are a superset of permissions for each app
  - App 1 can now exfiltrate; App 2 can now access GPS / camera

File Permissions

- Files written by one application cannot be read/ written by other applications
  - Not true for files stored on the SD card
  - SD card changes in Android 4.4: limited write ability

- Full file system encryption
  - Encryption key is protected with AES128 using key derived from user password (salted/hashed)
  - Root access not sufficient to break – need password
  - Enabled by default in Android 5.0

Memory Management

- Address Space Layout Randomization to randomize addresses on stack
- Hardware-based No eXecute (NX) to prevent code execution on stack/heap
- Stack guard derivative
- Some defenses against double free bugs (based on OpenBSD's dmalloc() function)
- etc.

[See http://source.android.com/tech/security/index.html]

Android Fragmentation

- Many different variants of Android (unlike iOS)
  - Motorola, HTC, Samsung, …
- Less secure ecosystem
  - Inconsistent or incorrect implementations
  - Slow to propagate kernel updates and new versions
    - "At Google I/O 2011, many of the largest OHA partners committed to providing updates to devices for 18 months after initial shipment."

CSE 484 / CSE M 584: Computer Security and Privacy

[And now for something completely different...]

CAPTCHAs

Spring 2015

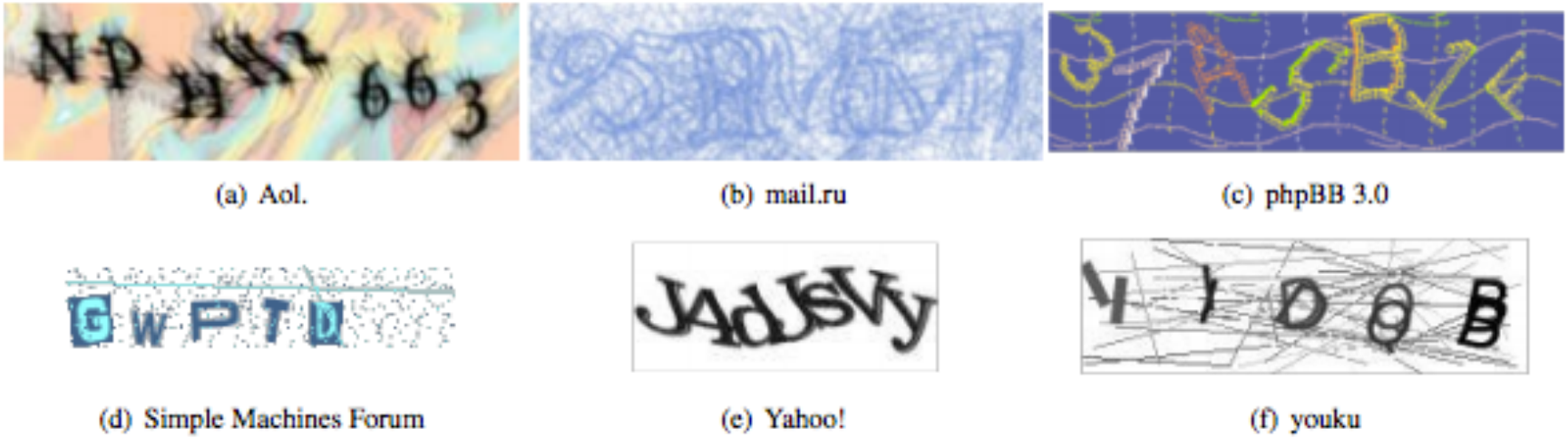
Franziska (Franzi) Roesner
franzi@cs.washington.edu

Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Human Verification

- Problem:
  - Want to make it hard for spammers to automatically create many free email accounts
  - Want to make it difficult for computers to automatically crawl some data repository
- Need a method for servers to distinguish between human users and machine users
- Approach: CAPTCHA
  - Completely Automated Public Turing Test to Tell Computers and Humans Apart

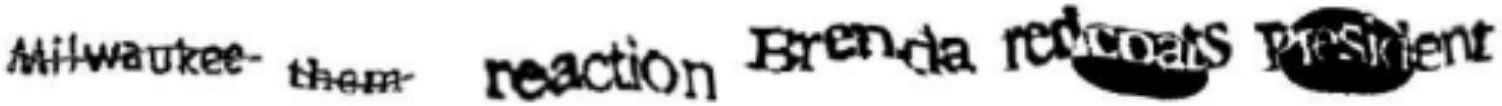
CAPTCHAs



(a) Aol. (b) mail.ru (c) phpBB 3.0

(d) Simple Machines Forum (e) Yahoo! (f) youku

Figure 1: Examples of CAPTCHAs from various Internet properties.



(a) Early 2008 (b) December 16th 2009 (c) January 24th 2010

Figure 2: Examples of CAPTCHAs downloaded directly from reCaptcha at different time periods.

Image from http://static.usenix.org/event/sec10/tech/full_papers/Motoyama.pdf

Questions

Q1: What do you like/dislike about CAPTCHAs?

Q2: What properties of CAPTCHAs are valuable?

Q3: What properties of CAPTCHAs are "problematic"?

Q4: Should web sites use CAPTCHAs?

Q5: Suppose you are a spammer and want to create free accounts on Webmail Provider X, and Webmail Provider X uses CAPTCHAs during enrollment. How would you go about breaking those CAPTCHAs?

CAPTCHA Solving

Re: CAPTCHAs – Understanding CAPTCHA-Solving Services in an Economic Context

Marti Motoyama, Kirill Levchenko, Chris Kanich, Damon McCoy,
Geoffrey M. Voelker and Stefan Savage
University of California, San Diego
{*mmotoyam, klevchen, ckanich, dlmccoy, voelker, savage*}*@cs.ucsd.edu*

Abstract

Reverse Turing tests, or CAPTCHAs, have become an ubiquitous defense used to protect open Web resources from being exploited at scale. An effective CAPTCHA resists existing mechanistic software solving, yet can be solved with high probability by a human being. In response, a robust solving ecosystem has emerged, reselling both automated solving technology and real-time human labor to bypass these protections. Thus, CAPTCHAs can increasingly be understood and evaluated in purely economic terms; the market price of a solution *vs* the monetizable value of the asset being protected. We examine the market-side of this question in depth, analyzing the behavior and dynamics of CAPTCHA-solving service providers, their price performance, and the underlying labor markets driving this economy.

alphanumeric characters that are distorted in such a way that available computer vision algorithms have difficulty segmenting and recognizing the text. At the same time, humans, with some effort, have the ability to decipher the text and thus respond to the challenge correctly. Today, CAPTCHAs of various kinds are ubiquitously deployed for guarding account registration, comment posting, and so on.

This innovation has, in turn, attached value to the problem of solving CAPTCHAs and created an industrial market. Such commercial CAPTCHA solving comes in two varieties: automated solving and human labor. The first approach defines a technical arms race between those developing solving algorithms and those who develop ever more obfuscated CAPTCHA challenges in response. However, unlike similar arms races that revolve around spam or malware, we will argue that the underly-

CAPTCHA-Solving Economies

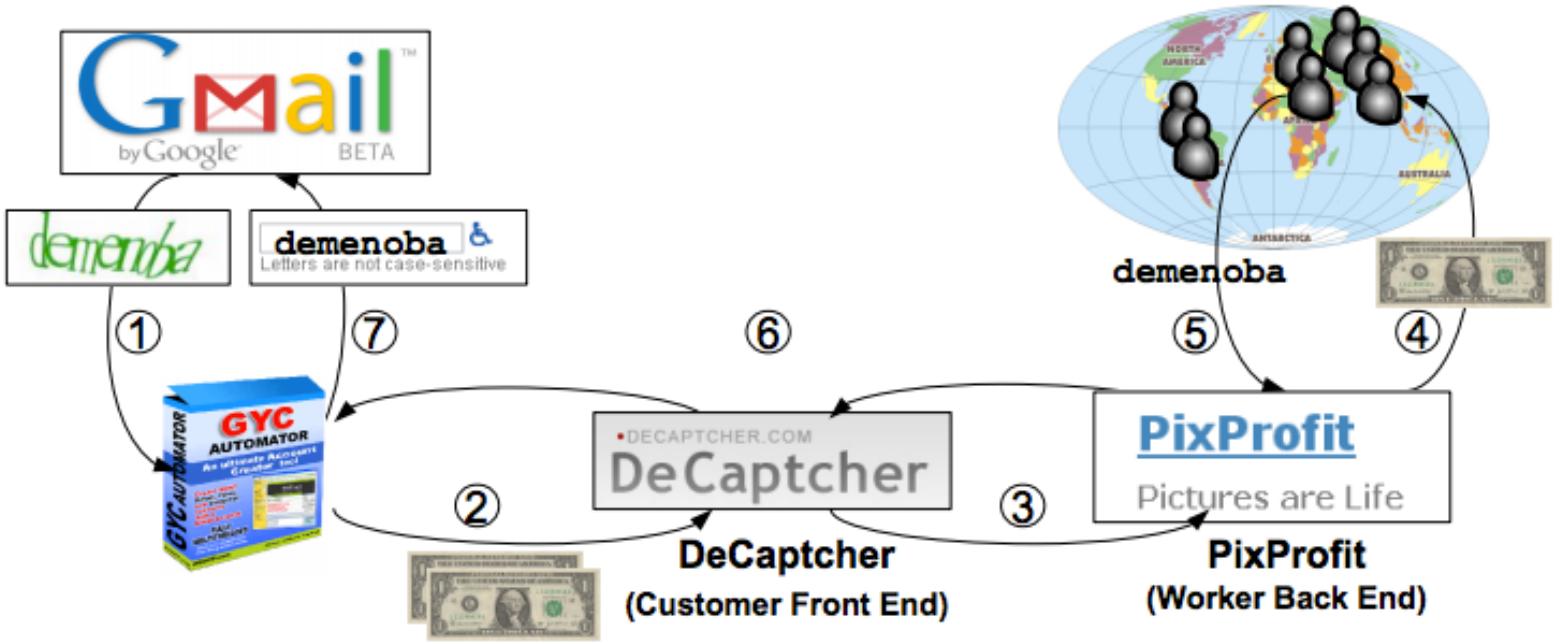


Figure 3: CAPTCHA-solving market workflow: ① GYC Automator attempts to register a Gmail account and is challenged with a Google CAPTCHA. ② GYC uses the DeCaptcher plug-in to solve the CAPTCHA at $2/1,000. ③ DeCaptcher queues the CAPTCHA for a worker on the affiliated PixProfit back end. ④ PixProfit selects a worker and pays at $1/1,000. ⑤ Worker enters a solution to PixProfit, which ⑥ returns it to the plug-in. ⑦ GYC then enters the solution for the CAPTCHA to Gmail to register the account.

Image from http://static.usenix.org/event/sec10/tech/full_papers/Motoyama.pdf

CAPTCHA-Solving Economies

Service	$/1K Bulk	Dates (2009–2010)	Requests	Responses
Antigate (AG)	$1.00	Oct 06 – Feb 01 (118 days)	28,210	27,726 (98.28%)
BeatCaptchas (BC)	$6.00	Sep 21 – Feb 01 (133 days)	28,303	25,708 (90.83%)
BypassCaptcha (BY)	$6.50	Sep 23 – Feb 01 (131 days)	28,117	27,729 (98.62%)
CaptchaBot (CB)	$1.00	Oct 06 – Feb 01 (118 days)	28,187	22,677 (80.45%)
CaptchaBypass (CP)	$5.00	Sep 23 – Dec 23 (91 days)	17,739	15,869 (89.46%)
CaptchaGateway (CG)	$6.60	Oct 21 – Nov 03 (13 days)	1,803	1,715 (95.12%)
DeCaptcher (DC)	$2.00	Sep 21 – Feb 01 (133 days)	28,284	24,411 (86.31%)
ImageToText (IT)	$20.00	Oct 06 – Feb 01 (118 days)	14,321	13,246 (92.49%)

Table 1: Summary of the customer workload to the CAPTCHA-solving services.

Image from http://static.usenix.org/event/sec10/tech/full_papers/Motoyama.pdf

Language	Example	AG	BC	BY	CB	DC	IT	All
English	one two three	51.1	37.6	4.76	40.6	39.0	62.0	39.2
Chinese (Simp.)	一 二 三	48.4	31.0	0.00	68.9	26.9	35.8	35.2
Chinese (Trad.)	一 二 三	52.9	24.4	0.00	63.8	30.2	33.0	34.1
Spanish	uno dos tres	1.81	13.8	0.00	2.90	7.78	56.8	13.9
Italian	uno due tre	3.65	8.45	0.00	4.65	5.44	57.1	13.2
Tagalog	isá dalawá tatló	0.00	5.79	0.00	0.00	7.84	57.2	11.8
Portuguese	um dois três	3.15	10.1	0.00	1.48	3.98	48.9	11.3
Russian	один два три	24.1	0.00	0.00	11.4	0.55	16.5	8.76
Tamil	ஒன்று இரண்டு மூன்று	2.26	21.1	3.26	0.74	12.1	5.36	7.47
Dutch	een twee drie	4.09	1.36	0.00	0.00	1.22	31.1	6.30
Hindi	एक दो तीन	10.5	5.38	2.47	1.52	6.30	9.49	5.94
German	eins zwei drei	3.62	0.72	0.00	1.46	0.58	29.1	5.91
Malay	satu dua tiga	0.00	1.42	0.00	0.00	0.55	29.4	5.23
Vietnamese	một hai ba	0.46	2.07	0.00	0.00	1.74	18.1	3.72
Korean	일 이 삼	0.00	0.00	0.00	0.00	0.00	20.2	3.37
Greek	ένα δύο τρία	0.45	0.00	0.00	0.00	0.00	15.5	2.65
Arabic	واحد اثنين ثلاثة	0.00	0.00	0.00	0.00	0.00	15.3	2.56
Bengali	এক দুই তিন	0.45	0.00	9.89	0.00	0.00	0.00	1.72
Kannada	ಒಂದು ಎರಡು ಮೂರು	0.91	0.00	0.00	0.00	0.55	6.14	1.26
Klingon	☐ < ∈	0.00	0.00	0.00	0.00	0.00	1.12	0.19
Farsi	یک دو سه	0.45	0.00	0.00	0.00	0.00	0.00	0.08

Table 2: Percentage of responses from the services with correct answers for the language CAPTCHAs.

Image from http://static.usenix.org/event/sec10/tech/full_papers/Motoyama.pdf