CAPTCHA

Rozpoznávanie ľudí a počítačov na webe

BAKALÁRSKA PRÁCA

Rastislav Vaško

UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

KATEDRA INFORMATIKY

Študijný odbor: Informatika 9.2.1

Vedúci bakalárskej práce

RNDr. Richard Ostertág

BRATISLAVA 2008

Čestne prehlasujem, že som túto bakalársku prácu
vypracoval samostatne s použitím citovaných zdrojov.

. . . . . .

Touto cestou sa chcem poďakovať svojmu vedúcemu
bakalárskej práce, RNDr. Richardovi Ostertágovi,
za cenné rady a pripomienky pri písaní.

Abstrakt

Táto práca sa zaoberá problematikou CAPTCHA - automatizovaným testom na rozpoznávanie ľudí a počítačov. Cieľom práce je objasniť dôvody vzniku a význam CAPTCHA, určiť kritéria a vlastnosti nutné pre vhodný návrh a následne aj pomocou nich analyzovať súčasný stav a existujúce implementácie a taktiež navrhnúť nové.

Kľúčové slová: CAPTCHA, world wide web, umelá inteligencia

Predhovor

Internet sa stal v ostatných pár rokoch nesmierne dynamickým miestom, pričom najvýraznejšie sa to prejavilo na World Wide Web. Web už nie je len o prístupe k dokumentom, ale aj o prístupe k službám. Či už ide o webmail, diskusné fóra alebo nakupovanie, sú to služby poskytované pre ľudí, ktoré vo forme, ako ich poznáme dnes, na webe pred niekoľkými rokmi neexistovali.

Prinieslo to však problém - ako rozpoznať, kedy k službe pristupuje človek a kedy automatizovaný skript? Samozrejme je táto otázka zmysluplná aj pri prezeraní dokumentov a statických webových stránok, avšak tu nie je problémom. Napríklad pri diskusnom fóre už to problém je - skript, ktorý pridá tisícky príspevkov za hodinu, nie je cieľová skupina takejto služby. Ako reakcia na tento problém vznikla CAPTCHA - automatizovaný test na rozpoznávanie ľudí od počítačov, pričom na to využíva otvorené problémy umelej inteligencie. Väčšina užívateľov webu sa s nimi už stretla - najčastejším zástupcom CAPTCHA sú obrázky s deformovaným textom, ktorý je treba prepísať do okienka pod obrázkom.

Cieľom práce je objasniť význam CAPTCHA, určiť kritéria a vlastnosti nutné pre vhodný návrh a aj s ich pomocou analyzovať súčasný stav tejto problematiky na webe a taktiež ponúknuť nové návrhy.

Obsah

Zoznam obrázkov

Zoznam použitých skratiek

AI Artificial Intelligence

API Application Programming Interface

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

OCR Optical Character Recognition

URL Uniform Resource Locator

WWW World Wide Web

W3C World Wide Web Consortium

XHTML Extensible Hypertext Markup Language

Kapitola 1

Úvod

Internet sa stal každodennou súčasťou našich životov. Začiatkom roku 2007 ho v priebehu troch mesiacov využilo zhruba 57% populácie EÚ, pričom aspoň raz týždenne ho používa viac ako 80% z týchto ľudí [Eur08][Eur06]. Veľká väčšina ľudí využíva Internet na prezeranie stránok z World Wide Web (ďalej len *web*) alebo na komunikáciu prostredníctvom e-mailov.

Web predstavuje platformu poskytujúcu týmto užívateľom možnosti inak nerealizovateľné. Možnosti, na ktoré sme si v nesmierne krátkom čase zvykli natoľko, že by sme ich neprítomnosť pocítili veľmi rýchlo. Najvýraznejšie sa to azda prejavuje na jazyku - výrazy ako "vygoogli si to", "pozri sa na wikipédiu" alebo "videl si na youtube..." sa dostali do bežnej komunikácie nielen mladých ľudí. Úspech webu (a Internetu celkovo) stojí na fakte, že bol koncipovaný ako všeobecne dostupný systém založený na otvorených štandardoch.

V dnešných dňoch je bežné nakupovať cez web, prispievať na rôzne fóra alebo písať vlastné blogy. Táto dynamickosť však priniesla problém. Povaha webu umožnuje softvéru 'tváriť sa' ako človek. Vlastnosť počítačov vo všeobecnosti zase je, že sú dobré na vykonávanie stereotypných operácií rýchlo a exaktne. Využijúc tieto dve spomenuté vlastnosti môžeme napríklad napísať softvér schopný prispievať na vybrané fórum tisícky príspevkov za hodinu s obsahom nami určeným. Alebo softvér registrujúci sa na free-mailovú službu, z ktorej následne odosiela haldy nevyžiadaných e-mailov. Takýchto príkladov môže byť ľubovoľne veľa, ale myšlienka je rovnaká - neschopnosť webovej aplikácie rozlišovať medzi relevantnými príkazmi (od ľudí) a spamom (od softvéru tváriaceho sa ako človek).

Ako reakcia na tento problém bol navrhnutý test s názvom **CAPTCHA** (**C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part).

Jednoducho povedané, ide o test, ktorý ľahko vyrieši človek, ale súčasné počítače nie.

Cieľom tejto práce je objasniť dôvody vzniku a dôležitosti CAPTCHA, definovať kritériá a vlastnosti nutné pre vhodný návrh, analyzovať súčasný stav a existujúce implementácie a taktiež navrhnúť nové.

To, že ide o dôležitý problém potvrdzuje fakt, že za prvé štyri mesiace roku 2008 sa vyskytlo hneď niekoľko útokov na populárne fungujúce implementácie CAPTCHA [Win08] [Web08a] [Web08c], čo vzbudzuje otázku, či vôbec funkčná a spoľahlivá implementácia existuje.

Obrázok 1.1: CAPTCHA webovej služby del.icio.us

Kapitola 2

Definícia

Problémom automatizovaného rozlišovania počítačov a ľudí sa pravdepodobne ako prvý zaoberal Moni Naor v [Nao96]. Pojem CAPTCHA bol však prvýkrát použitý až v roku 2000 v [vABHL00] a formalizovali ho o pár rokov neskôr v [vABHL03], odkiaľ definíciu preberám.

2.1 CAPTCHA

Nech C je pravdepodobnostná distribučná funkcia. Označíme $[C] = \{x \in R : Pr(\{x\}) > 0\}$. Ak $P(\cdot)$ je pravdepodobnostný program, budeme pod označením $P_r(\cdot)$ rozumieť deterministický program, ktorý sa vykoná, keď P použije ako náhodnú mincu r.

Nech (P, V) je dvojica pravdepodobnostných programov. Výsledok interakcie P a V označíme $\langle P_{r1}, V_{r2}\rangle$ kde $r1$ a $r2$ sú náhodné mince (pokiaľ je program deterministický, index je vynechaný). Hovoríme, že V je *test*, pokiaľ pre všetky P a $r1$, $r2$ interakcia medzi P_{r1} a V_{r2} skončí a $\langle P_{r1}, V_{r2}\rangle \in \{accept, reject\}$. P označujeme ako *dokazovateľ* a V *overovateľ* prípadne *tester*.

Úspešnosť entity A nad testom V označíme $Succ_A^V = Pr_{r,r'}[\langle A_r, V_{r'}\rangle = accept]$. Predpokladáme, že A má všetky potrebné vedomosti o tom, ako V funguje. Jediné čo A nesmie vedieť je r' - náhodu použitú V.

Hovoríme, že test V je $(\alpha;\beta)$-*ľudsky vykonateľný*, ak aspoň α časť populácie má úspešnosť viac ako β nad testom V.

AI problém je trojica $P = (S, D, F)$, kde S je množina inštancií problému, D je pravdepodobnostná distribučná funkcia nad S a $f : S \to \{0,1\}^*$ je funkcia odpovedí na inštancie. Nech $\delta \in (0, 1]$. Požadujeme, aby pre $\alpha > 0$ časť populácie H

$Pr_{x \leftarrow D}[H(x) = f(x)] \geq \delta$.

Hovoríme, že AI problém P je (δ, τ)-*vyriešený* ak existuje program A bežiaci najviac v čase τ na hocijakom vstupe z S tak, že $Pr_{x \leftarrow D,r}[A_r(x) = f(x)] \geq \delta$. Hovoríme, že A je (δ, τ) *riešením* P. Naopak, P je (δ, τ)-*ťažký AI problém* ak v súčasnosti neexistuje (δ, τ) riešenie pre P a AI komunita suhlasí, že je ťažké takéto riešenie nájsť.

(α, β, η)-CAPTCHA je test V, ktorý je (α, β)-ľudsky vykonateľný a má nasledovnú vlastnosť:

> Existuje (δ, τ)-ťažký AI problém P a program A taký, že pokiaľ B má úspešnosť vyššiu ako η nad V, tak A^B je (δ, τ) riešením pre P (A^B berie v úvahu aj čas potrebný na beh B).

Voľne povedané, CAPTCHA je test založený na ťažkom probléme umelej inteligencie (AI), ktorým má s úspešnosťou β prejsť aspoň α časť populácie. Pokiaľ nájdeme program, ktorý s úspešnosťou viac ako η prejde týmto testom, tak tento program môžeme transformovať na riešenie daného AI problému.

Taktiež sa dá na CAPTCHA pozerať ako na špecifický typ jednosmerných funkcií, s tým rozdielom, že musia byť ľudsky riešiteľné.

Treba zdôrazniť, že to, čo Kerckhoffov princíp znamená pre Kryptológiu, je nutné aplikovať aj na CAPTCHA - jediné o čom môžeme predpokladať, že útočník [1] nevie, je náhodnosť použitá pri výbere konkrétnej inštancie problému.

Vlastnosťou každej CAPTCHA je, že pokiaľ sú ľudia stále schopní riešiť ju úspešnejšie ako počítače, dá sa tento rozdiel ľubovoľne (po 1) zväčšiť, nech je sebemenší. Ten spôsob je viacnásobné následné riešenie a je bližšie opísané v [vABHL03].

2.2 Porovnanie s Turingovym testom

Turingov test, navrhnutý A. M. Turingom v [Tur50], je známy návrh na testovanie schopnosti stroja preukázať inteligenciu. V základnej verzii sa ľudský rozhodca rozpráva s jedným strojom a jedným človekom, pričom obaja sa snažia predstierať, že sú ľuďmi. Stroj prešiel testom, pokiaľ rozhodca nie je schopný rozlíšiť stroj od človeka. Všetká komunikácia prebieha cez textový terminál.

Hlavný rozdiel medzi Turingovým testom a CAPTCHA je, ako samotný akronym napovedá, v tom, že CAPTCHA má namiesto ľudského rozhodcu algoritmus. Inak

[1] Útočník v tomto zmysle predstavuje program, ktorý sa snaží vyriešiť danú CAPTCHA

povedané, znamená to, že hladáme problémy, ktoré sú algoritmicky generovateľné a overovateľné, ľudsky riešiteľné a pritom v súčasnosti neriešiteľné algoritmicky.

Kapitola 3

Problematika

Aby sme mohli analyzovať možnosti riešenia problému neformálne popísateľného ako "rozpoznávanie ľudí a počítačov", potrebujeme si sformulovať a opísať nástroje a prostredie, v ktorom budeme pracovať - ich možnosti a limity.

3.1 Základné pojmy

Pre väčšinu ľudí bez informatického vzdelania sú slová *Internet* a *Web* synonymom. V zásade to pre verejnosť veľkým problémom nie je, pre nás je však tento rozdiel podstatný.

Internet je najväčšia počítačová sieť na svete. Tvorí ju množstvo menších autonómnych sietí, ktoré sú navzájom poprepájané. Je založená na rodine štandardov TCP/IP. Slúži ako infraštruktúra pre prenos dát a rôzne služby ako napr. e-mail, web, VoIP, prenos súborov a iné.

World Wide Web je systém navzájom poprepájaných hypertextových dokumentov a webových služieb, ktoré sú prístupné cez Internet pomocou webového prehliadača. Je ťažké odhadnúť, koľko dokumentov sa na webe nachádza, ale najnovšie odhady sa pohybujú medzi 12 až 45 miliardami voľne prístupných stránok a 550 miliardami stránok v tzv. Hlbokom Webe (týmto pojmom sa označuje tá časť webu, ktorá je neprístupná pre crawleri webových vyhľadávačov) [AG05][dK08][Ber01].

Webová stránka je dokument, zvyčajne napísaný v HTML a dostupný cez Internet.

Webové sídlo je množina všetkých webových stránok fungujúcich na rovnakej doméne alebo subdoméne, zvyčajne dostupných cez Internet, ktoré na seba pomocou hyperliniek odkazujú.

Internetový robot (skrátene *bot*) je softvér, ktorého úlohou je vykonávať určitú automatizovateľnú úlohu cez Internet. Vykonáva úlohu, ktorú treba mnohokrát opakovať a netreba pri nej ľudskú intervenciu. Typickým príkladom sú boti na rozosielanie nevyžiadaných e-mailov alebo boti internetových vyhľadávačov, ktorí sa označujú ako crawleri [1].

Crawler je softvér, ktorý prechádza webovými stránkami a analyzuje ich dáta podľa potreby. Štandardnou úlohou crawlera je extrahovať hyperlinky z kódu aktuálnej stránky, čím rozšíri vlastnú databázu liniek o ďalšie URL, ktoré potenciálne plánuje navštíviť. Najbežnejší crawleri sú roboti internetových vyhľadávačov, ktorí archivujú a analyzujú obsah stránky. Avšak existujú aj crawleri, ktorých zámery sú menej ušľachtilé - napríklad takí, ktorí extrahujú nielen linky, ale aj e-mailové adresy zo stránok, aby na tieto následne rozosielali nevyžiadané e-maily.

HTTP je protokol aplikačnej vrstvy pre distribuované kolaboratívne hypermediálne informačné systémy [IET99]. Je to bezstavový protokol. Názov je trochu zavádzajúci v zmysle, že naznačuje, akoby bol protokol výhradne určený na prepravu hypertextových dokumentov. Nie je tomu tak. Protokol je generický čo sa týka formátu textového obsahu a v praxi je využívaný na transport dát vo formátoch (okrem HTML) na báze XML, prípadne JSON a iných.

HTML je najpoužívanejší značkovací jazyk na písanie webových stránok. Pod týmto pojmom budeme pre účely tejto práce rozumieť ako jazyk HTML 4.01 tak XHTML 1.0. Oba sú sémanticky ekvivalentné, ale majú určité rozdiely v syntaxy, pričom ich podrobné opisovanie by nemalo pre účel tejto práce zmysel [2].

3.2 Význam

Možných a častokrát kritických aplikácií CAPTCHA je už v dnešnej dobe mnoho a je pravdepodobné, že ich bude v budúcnosti pribúdať. Cieľom tejto sekcie je načrtnúť tie najbežnejšie a tie, ktoré v dnešnej dobe tak bežné nie sú, ale mali by sa nimi stať.

[1] Najpresnejší slovenský ekvivalent slova ‘crawler’ je pravdepodobne ‘lezúň’, ale keďže tento pojem nie je zaužívaný, rozhodol som sa používať anglický originál.

[2] V zásade nám nič nebráni obsiahnuť aj aktuálnejšie verzie XHTML - XHTML 1.1 a XHTML 2.0. Tieto sa však na Webe vyskytujú v marginálnom množstve a na závery tejto práce by nemali vplyv.

3.2.1 Registrácia užívateľov

S nástupom webových sídiel dynamickejšej podstaty ako sú statické dokumenty sa stalo bežným štandardom, že služba požaduje od užívateľa, aby sa v určitej forme registroval. Je prirodzené, že takéto služby majú záujem, aby sa registrovali len užívatelia, ktorí majú záujem službu využívať vhodným spôsobom. Pre malé stránky častokrát postačuje možnosť, aby sa o registrácie starala poverená osoba s ktorou potenciálni noví užívatelia komunikujú či už e-mailom alebo inou formou, prípadne táto osoba overuje formulár vyplnený pri registrácií. Pre väčšie stránky je takýto systém zle škálovateľný a tým pádom nevhodný. CAPTCHA poskytuje pri registrácii spôsob ochrany pred jedným typom útoku. Nedokáže samozrejme rozpoznať úmysly potenciálneho užívateľa, ale dokáže efektívne zabrániť, aby útočník použil automatizovaný skript na posielanie potenciálne neobmedzeného množstva nových požiadaviek na registráciu.

Klasickým príkladom môžu byť webmailové služby ako napr. Gmail, Hotmail alebo Yahoo Mail. Tieto webmaily umožňujú hocikomu zaregistrovať sa a vytvoriť si týmto spôsobom e-mailovú adresu na príslušných doménach. Spolu viac ako pol miliardy zaregistrovaných užívateľov a denný prírastok rádovo v tisícoch registrácií na každom spomínanom webmaily predstavuje množstvo, ktoré je nepraktické spracovávať inak ako plne automatizovane. Na druhej strane sú kontá na spomínaných populárnych webmailoch pomerne cennou komoditou pre ľudí zaoberajúcich sa rozosielaním nevyžiadaných e-mailov. Práve z dôvodu, že výnosnosť spamu rastie s množstvom rozoslaných e-mailov, vyžaduje každý z vyššie uvedených webmailov už dlhšiu dobu úspešné zvládnutie CAPTCHA pri registrácii. Samozrejme, nezabraňuje to vytváraniu e-mailových adries na rozosielanie spamu, avšak relatívne účinne to zťažuje a brzdí tento proces.

3.2.2 Stop crawlerom

Ako už bolo spomenuté, nie každý crawler je napísaný s dobrými úmyslami. Existuje protokol s názvom *Robots Exclusion Protocol* popísaný v [Kos], ktorý ponúka jednoduchý formát a spôsob pre majiteľa domény na definovanie pravidiel prístupu pre roboty. Problém je, že tieto sú definované na strane domény a musia byť implementované na strane crawlera a tým pádom nie sú vynútiteľné. Inak povedané, iba ‘slušné’ roboty tento protokol rešpektujú. Na druhej strane, CAPTCHA predstavuje veľmi účinný spôsob na ochranu informácií pred crawlermi. Príklad použitia je skrývať e-mailové adresy na stránke a ukázať ich až po úspešnom zvládnutí testu.

3.2.3 Hlasovanie

Voľby, ankety a iné formy hlasovania tak ako ich poznáme z reálneho života nie sú vždy jednoducho napodobiteľné na webe. Pokiaľ máme vlastnú spoľahlivú formu autentifikácie užívateľov, vieme relatívne jednoducho zabezpečiť, aby každý oprávnený užívateľ hlasoval najviac raz. Avšak ak chceme, aby bolo hlasovanie dostupné pre každého 'anonymného' návštevníka stránky, tak to je netriviálny problém. Riešenie, ktoré by zabezpečilo, že návštevník môže hlasovať najviac raz, neexistuje. Je to preto, že jediné, k čomu môžeme hlas priradiť, je počítač (resp. IP adresa, prípadne prehliadač) a nie človek, ako je to pri autentifikovanom užívateľovi. Viacnásobnému hlasovaniu sa síce nedá zabrániť, avšak dá sa sťažiť.

Jednou možnosťou je pamätať si IP adresu, ale to nerieši dynamické IP adresy, NAT, proxy servery, atď. Druhou možnosťou je uložiť do cookie prehliadača návštevníka informáciu, ktorá nám pri overení povie, že daný užívateľ už hlasoval. Cookie sú však ukladané na strane klienta, preto ich môže klient odstrániť a tým pádom sa informácia stratí. Pre hlasovanie s väčším počtom hlasujúcich môžu byť vyššie spomenuté opatrenia dostatočné, aby odradili väčšinu potenciálnych útočníkov a účinok tých, ktorým sa podarí hlasovať viackrát, bude zanedbateľný. Pokiaľ však títo napíšu robota, ktorý bude hlasovať raz za interval, ktorým sa stanú viacnásobné hlasy relevantnými pre výsledky obchádzajúc spomenuté opatrenia, môže to byť problém. Z tohto dôvodu je použitie CAPTCHA pri hlasovaní bez možnosti autentifikácie vhodným bezpečnostným opatrením.

3.2.4 Autentifikácia

Bežný spôsob autentifikácie na webe je vyplnenie formulára s položkami 'meno' a 'heslo', ktorý sa overí na strane servera. Je známe, že užívatelia používajú krátke heslá, častokrát zložené len z písmen, prípadne čísel. Jednoducho také, ktoré sa ľahko pamätajú, keďže sú zaregistrovaní na desiatkách webových sídiel a denne sa na časť z nich prihlasujú. O to viac je takáto forma autentifikácie náchylná na slovníkový útok - postupne automatizovane posielať pre dané prihlasovacie meno potenciálne heslá zo zoznamu slov, až kým nenájdem to správne. Malé ani stredne veľké stránky nezvyknú mať nijakú ochranu pred týmto útokom; napríklad vo forme systému, ktorý sleduje, koľkokrát sa v danom intervale niekto neúspešne snaží prihlásiť ako určitý užívateľ. Takýto systém pritom nie je sám o sebe veľmi dobré riešenie, pretože je buď neefektívny alebo má potenciál zakázať autentifikáciu pre reálnych užívateľov. Zatiaľ takmer vôbec nepoužívaným a pritom veľmi užitoč-

ným bezpečnostným opatrením je pridanie CAPTCHA pri autentifikácií užívateľa cez webovú stránku po prvom nesprávnom zadaní hesla. Jednoducho povedané - pri prvom pokuse o prihlásenie sa vypĺňa len meno a heslo. Pokiaľ sú korektné, prihlásenie je úspešné. Pokiaľ nie, zobrazí sa znova prihlasovací formulár, ale aj s CAPTCHA. Toto zabezpečí, že nie je možné použiť automatizovaný slovníkový útok, ale pre reálneho užívateľa to nepredstavuje problém väčší ako nutnosť prejsť danou CAPTCHA. Zostáva možnosť, že ľudský útočník bude 'ručne' vypĺňať meno a heslo pri každom pokuse a od druhého pokusu aj CAPTCHA. Tento typ útoku je zťažený práve o to, že útočník musí pri každom pokuse vyplniť aj CAPTCHA, čo pri n pokusoch môže predstavovať netriviálny čas v závislosti od jej typu.

Táto forma ochrany autentifikácie ešte nie je takmer vôbec zaužívaná, čo som si všimol pri uskutočňovaní malého prieskumu - využívajú ju Google, Yahoo a Microsoft pri prihlasovaní a AOL pri obnove hesla. Pravidelne využívam služby prvých troch menovaných spoločností a toto som si všimol až teraz, keď som sa schválne zle autentifikoval, čo naznačuje, že tento systém nie je užívateľsky nepríjemný, ale poskytuje ochranu pred určitým typom útokov.

3.3 Útoky

Od začiatku roka 2008 sa objavilo hneď niekoľko správ o prelomení CAPTCHA veľkých spoločností ako Google alebo Microsoft; napríklad [Win08] [Web08c] [Web08b]. Väčšinou ide o správy, ktoré nie je možné overiť, pretože zverejnené informácie sú nedostatočné, aby nebolo možné útok so zlým úmyslom reprodukovať, alebo sú to skôr zreprodukované spôsoby zachytenia správ útočníkov.

> Úlohou CAPTCHA nie je možnému útoku na službu zabrániť, ale obmedziť jeho škálovateľnosť.

Poďme sa teraz pozrieť na základné typy útokov známe z kryptoanalýzy a následne na niektoré špecifické pre CAPTCHA ako tému a generické ako množinu realizácií:

3.3.1 Útok hrubou silou

Na CAPTCHA nie je vo všeobecnosti reálne možné vykonať útok hrubou silou, tak ako je tomu zvykom napríklad pri kryptoanalýze symetrických šifier. Dôvod je ten, že pri načítaní stránky je vygenerovaná nová inštancia CAPTCHA ktorej riešenie

nesúvisí s predchádzajúcou inštanciou. To znamená, že pokiaľ nenájdem riešenie danej inštancie na prvý pokus, dostanem novú a som na začiatku. Samozrejme, pokiaľ sa bavíme o konkrétnej inštancií CAPTCHA s odôvodnením, že tak ako máme pri symetrickom šifrovaní jeden klúč, ktorý sa snažíme uhádnuť, rovnako máme fixovanú jednu "náhodnú mincu", ktorá vyberie inštanciu CAPTCHA. V tom prípade je možné útok hrubou silou vykonať, ale toto je v praxi irelevantné z dôvodu, že po odoslaní chybného riešenia by mala byť náhodne vygenerovaná nová inštancia.

Šifrovaný text častokrát nie je potrebné rozlúštiť 'okamžite' a časové intervaly, v ktorých má jeho dešifrovanie ešte hodnotu, sú vo všeobecnosti oveľa dlhšie ako je to u CAPTCHA. Taktiež šifrovaný text je použiteľný samostatne, v zmysle, že pre útočníka predstavuje všetko čo potrebuje, aby sa pokúsil o útok hrubou silou, pokiaľ má určité poznatky o tom, ako má dešifrovaná správa vyzerať (či už ide o jazyk, formu alebo samotnú správu). Naopak, CAPTCHA je interaktívny systém, pretože inštancia sama o sebe hodnotu nemá, ale stráži niečo ďalšie, k čomu útočník prístup nemá a ten na overenie správnosti svojho riešenia musí kontaktovať druhú stranu.

3.3.2 Útok opakovaním

Tento typ útoku prebieha vo všeobecnosti tak, že sa validné dáta použijú viackrát, prípadne neskoršie. V prostredí CAPTCHA je tento útok možný len chybou implementácie. Ako bolo spomenuté, protokol HTTP je bezstavový a preto je nutnosťou každej implementácie CAPTCHA, aby si vytvorila spôsob zaznamenávania relácií. Nutné je to preto, že CAPTCHA celkom logicky prebieha vždy minimálne v troch krokoch - najprv sa pošle problém, následne sa odošle riešenie a nakoniec sa pošle odpoveď. Nezáleží ani tak na tom, ako sú relácie implementované [3]. Pokiaľ však po úspešnom prejdení CAPTCHA systém danú reláciu neukončí, je možné vykonať útok opakovaním. Nemusíme pritom ani prelomiť danú CAPTCHA - človek ju raz vyrieši a následne robot opakovane použije danú odpoveď, pretože relácia (inak povedané konkrétna inštancia CAPTCHA) je stále platná. Aj keď to nie je to najlepšie, čo si môže útočník želať, pretože tento útok vyžaduje ľudskú intervenciu, pokiaľ je čas do vypršania relácie dostatočne dlhý, stáva sa tento typ útoku pomerne dobre škálovateľným a v praxi použiteľným.

[3] pokiaľ to je v rámci možností urobené spoľahlivo a bezpečne

3.3.3 Útok presmerovaním

V prostredí CAPTCHA je tento typ útoku známejší pod názvom "Pornografický útok". Volať ho útokom však nie je celkom presné, i keď to má v názve a častokrát je zaň v rôznych článkoch považovaný. Postup demonštrujeme na príklade registrácie na službu Gmail, pričom ju budeme označovať ako G a konkrétna stránka má adresu `https://mail.google.com/mail/signup` a obsahuje CAPTCHA na báze rozpoznávania textu (viac o tomto type v sekcií 5.1). Server, ktorý 'útok' vykoná označíme ako E. Predpokladom je, že užívateľ A má záujem o zobrazenie stránky od E:

1. A pošle cez HTTP požiadavku o načítanie web stránky od E
2. E načíta web stránku G a dostane HTML kód danej stránky. Kedže jeho štruktúra mu je dopredu známa, vie ho sparsovať tak, aby dostal URL adresu na ktorej je vygenerovaný obrázok konkrétnej inštancie CAPTCHA pre danú reláciu (ktorú začal E keď požiadal o načítanie stránky G)
3. E pošle späť A HTML kód s formulárom, ktorý obsahuje `<img>` element ukazujúci na danú vygenerovanú CAPTCHA a textové políčko kde očakáva odpoveď s odôvodnením, že pokiaľ chce A vidieť reálny obsah stránky od E, tak musí najprv prepísať text, ktorý vidí na danom obrázku
4. A pošle E odpoveď
5. E prepošle odpoveď od A spolu s ďalšími údajmi potrebnými na registráciu (meno, heslo, atď), ktoré zvolil E sám, serveru G
6. G pošle odpoveď E
7. E vie, že môže dostať dva typy odpovede - buď, že registrácia bola úspešna, ak A vyriešil danú CAPTCHA správne, alebo bola registrácia zamietnutá. V prvom prípade pošle E stránku, o ktorú A v prvom kroku požiadal. V druhom prípade pošle chybovú hlášku alebo sa pokúsi zopakovať postup od druhého kroku

Ako vidno, samotná CAPTCHA prelomená nie je - je riešená človekom. Celá myšlienka je v automatizácií všetkého ostatného čo ide. Variáciou na tento systém je, že motivácia užívateľa A nespočíva v záujme o obsah stránky od E, ale sú ňou peniaze. Je mu zobrazovaná jedna CAPTCHA za druhou a po správnom riešení si

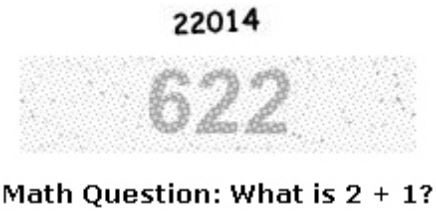
vždy na svoj účet pripíše určitú sumu. Takýto systém nie je prelomením CAPTCHA a jeho jediné využitie je biznis. To je súčasne aj jeho slabina - útočník musí byť schopný vyťažiť zo získanej vyriešenej CAPTCHA viac, ako sú jeho náklady na zabezpečenie motivácie človeka, ktorý ju vyrieši.

3.3.4 Bezpečnosť pomocou utajenia

Táto (z anglického *Security through obscurity*) kontroverzná technika z oblasti informačnej bezpečnosti sa snaží dosiahnuť bezpečnosť pomocou utajenia (systému, dizajnu, implementácie, atď). Je v kontraste s Kerckhoffovým princípom. V praxi sa uplatňuje tak, že stránka, ktorá má pocit, že je pre ľudských útočníkov dostatočne nezaujímavá, využíva techniky s efektom blížiacim sa CAPTCHA, avšak len na boty, ktoré sú dostatočne generické a nie sú dopredu napísane s tým, aby danú techniku prekonali - aj keď to (väčšinou pomerne ľahko) ide.

Najbežnejšie sa tak deje pri chránení diskusných fór. Nie je nič netradičné, keď aj malé stránky s diskusným fórom dostanú denne desiatky umelo vygenerovaných HTTP príkazov, ktorých účelom je vytvorenie príspevku obsahujúceho spam. Bez náležitej ochrany sa tak môže veľmi rýchlo stať, že sa diskusné fórum premení na nelogickú ale búrlivú debatu botov. Aby sa tak nestalo a v spojení s predstavou, že neexistuje útočník, ktorý by si dal námahu, aby napísal bota šitého na mieru danej stránky, použije jej správca techniku, ktorá funguje, ale iba keď je predpoklad o neexistujúcom útočníkovi splnený. Medzi tieto techniky patria *honeypots*, všetko čo sa dnes tvári ako *textová* CAPTCHA a množstvo toho, čo sa tvári ako CAPTCHA *na báze rozpoznávania textu* (o posledných dvoch menovaných viac v kapitole 5).

Pri útoku na "textové CAPTCHA" ide o vhodné sparsovanie textu, keďže sú to vždy úlohy typu "Koľko je $X + Y$?" alebo "Napíš písmeno Z do okienka vedľa" čo za CAPTCHA považované byť nemôže. U tretej skupiny častokrát stačí použiť dostupný OCR softvér (pretože text nie je dostatočne deformovaný) a tým pádom to už CAPTCHA nie je. Príklady spomínaných dvoch skupín sú:



Obrázok 3.1: Príklady príliš jednoduchých testov

Pre úplnosť si pár riadkami opíšme, ako honeypots fungujú. HTML formuláre s rovnakým využitím majú väčšinou podobné `name` atribúty elementov. Napríklad, typický formulár na pridanie komentáru k článku má položky 'name', 'email', 'url' a 'text' kde je už z tohto atribútu jasné, čo bude obsahom. Toto využije zákerný crawler, ktorý keď uvidí formulár s menovanými položkami, ľahko si hneď overí, či je stránka náchylná na automatizované útoky - pošle serveru vhodne vygenerovaný HTTP request, kde text je typu 'Skvelá stránka. Páči sa mi. Čau!' čo vyzerá nenápadne. Následne sa pozrie na stránku znova či sa tam jeho príspevok zobrazil. Honeypots fungujú tak, že pridám do formulára položky s nezmyselnými názvami a aspoň jeden s rozumným názvom (ako tie štyri vyššie uvedené). Ten posledný menovaný dám vizuálne skryť. Na strane servera si následne pri overovaní formulára pozriem, či bol vyplnený aj ten skrytý. Ak áno, predpokladám, že ho vyplnil bot, pretože človek ho nevidel. Opačne keď vyplnený nie je. Jednou z nevýhod tejto techniky je, že môže byť pomerne neprístupná, pokiaľ užívatelia používajú iné ako vizuálne prehliadače a nezaručuje nijakú bezpečnosť pred sofistikovanejšími botmi.

Treba spomenúť, že každá z troch spomínaných techník má argument, prečo ju používať - v prvom prípade je to kritérium rýchlosti, v druhom kritérium prístupnosti a v tretom kritérium presnosti (kritériam sa podrobnejšie venuje nasledujúca kapitola). Tieto kritéria sa však do určitej miery dajú (aj súčasne) dosiahnúť použitím vhodnej CAPTCHA.

Kapitola 4

Kritéria

V kapitole číslo 2 sme zaviedli formálnu definíciu toho, čo považujeme za CAPTCHA. V praxi si však nie sú všetky CAPTCHA úplne rovné. Tým aký problém ako využívame zároveň určujeme jej vlastnosti a obmedzenia. Zvolil som päť kritérií, ktoré určujú, ako dobrý je v rôznych ohľadoch daný typ CAPTCHA. Táto pätica určite nie je vyčerpávajúca a neobsiahne všetky špecifiká vyplývajúce z relatívne veľkej slobody pri vytváraní CAPTCHA, dokáže však poskytnúť určitú informáciu o tom, aké sú výhody a slabiny jednotlivých typov CAPTCHA.

4.1 Presnosť

Ako už bolo niekoľkokrát spomínané, účelom CAPTCHA je rozlíšiť ľudí od počítačov. Je preto zrejmé, že to s akou presnosťou túto funkciu plní je dôležitým kritériou na určovanie jej kvality.

Pozerať sa na presnosť len ako na percento úspešnosti plnenia úlohy sa síce dá, vypovedalo by to však len obmedzene o tejto vlastnosti. Skúsme sa preto na moment pozerať na úlohu CAPTCHA očami štatistiky. V tomto svete by sa úloha dala preformulovať na overovanie hypotézy H "CAPTCHA vyriešil človek". Na definovanie kritéria presnosti použijeme dva známe pojmy - chyba 1. druhu a chyba 2. druhu.

4.1.1 Chyba 1. druhu

Označovaná aj ako *false positive* je pravdepodobnosť, že hypotéza H bude zamietnutá, keď je pravdivá. Inak povedané, že človek cez CAPTCHA neprejde, teda ju

nevyrieši alebo vyrieši nesprávne. Táto vlastnosť súvisí s parametrom β v definícii CAPTCHA, ktorý hovorí, s akou pravdepodobnosťou ňou má α časť populácie prejsť. Jemný rozdiel je v tom, že zatiaľ čo β určuje tú najnižšiu hladinu, ktorá musí platiť pre celé dané percento populácie, môžeme sa na túto vlastnosť pozerať jemnozrnnejšie, pretože málokedy[1] si je celá relevantná populácia rovná v tom, ako často sa pri riešení CAPTCHA mýli.

Obrázok 4.1: Príklady neprečítateľných CAPTCHA

4.1.2 Chyba 2. druhu

Druhý dôležitý druh chýb, označovaný aj *false negative*, je pravdepodobnosť, že hypotéza H nebude zamietnutá, keď je nepravdivá. Inými slovami, počítač prejde testom. Je to hladina, určená parametrom η z definície CAPTCHA, ktorú môže počítač dosiahnúť bez toho, aby sme daný test považovali za prelomený. Napríklad, pokiaľ má inštancia práve n možných odpovedí (čo je v praxi veľmi pravdepodobné) je triviálna pravdepodobnosť, že počítač testom prejde rovná $1/n$.

Obrázok 4.2: Príklady prelomiteľných CAPTCHA

4.2 Prístupnosť

Webová prístupnosť (z anglického *web accessibility*) je pojem týkajúci sa praktiky tvorby webových stránok použiteľných ľudmi všetkých schopností aj hendikepov. Človek bez hendikepov si ani neuvedomuje, ako rozdielne sa na web pozerajú ľudia so znevýhodnením, či už zrakovým, sluchovým, pohybovým, kognitívnym alebo iným. Webové stránky sa však dajú robiť tak, aby boli do takmer úplne rovnako

[1] v skutočnosti, v žiadnej známej existujúcej implementácií CAPTCHA tomu tak nie je

použiteľné pre ľudí s aj bez hendikepov. Treba však priznať, že písať prístupné stránky je minimálne rovnako náročné ako písať *validné* HTML stránky, čo je činnosť výrazne zložitejšia ako písanie *fungujúcich* HTML stránok.

Táto vlastnosť súvisí s CAPTCHA hneď dvakrát. Jednak je pre autora prístupného webové sídla prirodzene dôležité, aby bola prístupná aj použitá CAPTCHA. Naopak, ľudský užívateľ by nemal byť prečo neschopný testom prejsť len z dôvodu, že má znevýhodnenie. Bohužiaľ to v praxi celkom tak nie je a žiadna známa CAPTCHA nie je univerzálne prístupná. Pomerne dobrú prístupnosť je možné v súčasnosti dosiahnúť systémom, ktorý dáva dokazovateľovi na výber, či chce riešiť CAPTCHA založenú na zvuku, alebo takú, na ktorú treba zrak.

4.3 Rýchlosť

Užívatelia webu sú vo všeobecnosti veľmi nároční čo sa týka rýchlosti prístupu k požadovanej informácií. Málokto bezcieľne prechádza od stránky ku stránke. Väčšina užívateľov ide za konkrétnou informáciou, stránkou, údajom a čím rýchlejšie ju dosiahne, tým je spokojnejší. Je známe, že užívatelia neradi čakajú viac ako štyri sekundy na načítanie stránky, väčšinou len prebehnú očami po stránke (a tento pohyb je skôr tvaru písmena F než úplny pohyb zľava-doprava a zhora-dole) namiesto čítania paragrafov textu a priemerné množstvo textu prečítaného na danej stránke je zhruba 20% [AJ06][Nie08]. Z tohto vyplýva, že ak v ostatných ohľadoch aj kvalitná CAPTCHA trvá príliš dlho na vyriešenie, stáva sa nepraktickou. ‘Príliš dlho’ je relatívny pojem, avšak štandardne predstavuje nanajvýš niekoľko sekúnd.

Rýchlosť pri CAPTCHA je teda čas potrebný pre ľudského užívateľa na prejdenie testom - od momentu kedy sa užívateľ rozhodne test vyriešiť až po moment ukončenia zadávania odpovede. Tento časový úsek sa vo všeobecnosti dá rozdeliť na tri fázy: pochopenie úlohy, riešenie úlohy a zadanie odpovede. Prvá fáza je veľmi špecifická pre jednotlivé typy CAPTCHA. V súčasnosti sú však takmer všetky bežne používané typy založené na úlohe prepísať informáciu z grafickej alebo zvukovej podoby do podoby textovej, kde je čas potrebný na túto fázu veľmi krátky. Naviac, užívateľ ktorý už v minulosti daný typ úlohy vyriešil, vie takmer okamžite, čo sa od neho očakáva, keď uvidí inú inštanciu rovnakej CAPTCHA znova, čo nemusí byť pravda vždy. Dĺžka druhej fázy, viac ako ostatné dve fázy, závisí ako od konkrétnej CAPTCHA tak aj od konkrétnej inštancie. Optimálne by mal užívateľ vedieť odpoveď okamžite, prípadne *takmer* okamžite. O tretej fáze má zmysel diskutovať až pri implementácií CAPTCHA na webovej stránke, pričom je ovplyvnená

prehliadačom a potrebnými vstupnými zariadeniami, avšak častokrát tvorí veľkú časť celkového času potrebného na prejdenie.

4.4 Náročnosť

Náročnosť je merateľná minimálne v dvoch ohľadoch: náročnosť na implementáciu CAPTCHA a hardvérová náročnosť vygenerovania jednej inštancie. Druhá menovaná nepredstavuje v dnešných dňoch na aktuálnych počítačoch pre v praxi používané CAPTCHA veľký problém, v závislosti od miesta nasadenia ním však môže byť. Náročnosťou na implementáciu sa myslia náklady (či už vo forme času alebo peňazí za hardvér alebo potrebné dátové zdroje) spojené so zavedením danej CAPTCHA do praxe - napísanie kódu, testovanie, obstaranie potrebných dát, atď. Napríklad CAPTCHA, ktorá potrebuje na fungovanie pol milióna špecifických obrázkov má predpoklad byť náročnejšia na implementáciu ako CAPTCHA založená na rozpoznávaní textu, ktorá si dokáže obrázky sama generovať.

4.5 Prostredie

Aj keď je trh s webovými prehliadačmi čím ďalej tým viac rozkúskovaný, ochota väčšiny ich tvorcov spolupracovať na rovnakých štandardoch sa postupne zvyšuje. Súčasná generácia prehliadačov má pomerne dobré výsledky v nasledovaní štandardov ako HTML 4.01, CSS 2.1, DOM Level 2 a implementácií jazyka ECMAScript. Problémom je, že staršie avšak stále používané prehliadače takúto podporu nezabezpečujú. Preto môže byť pri konkrétnej implementácií CAPTCHA dôležité, o aké jazyky sa opiera.

Väčšinou si CAPTCHA vystačí s HTML, prípadne s CSS na úpravu prezentácie. Ak implementácia využíva skriptovací jazyk, mala by tak robiť výlučne v rámci princípu *Progresívneho vylepšovania* (z anglického *Progressive enhancement*), ktorý hovorí, že webové stránky by mali byť vytvárané pre tie najmenej zdatné zariadenia s tým, že kvalitnejším zariadeniam ponúkne lepšie verzie danej stránky. Inak povedané, každý by mal mať prístup aspoň k základnému obsahu a funkcionalite stránky a ak zariadenie na strane užívateľa zvládne viac (či už ide o podporované jazyky prehliadača, šírku pásma pripojenia alebo iné kritérium), ponúkneme mu to. Samozrejme to neznamená, že budeme každú stránku robiť v n verziách. Treba zvoliť technológie tak, že pokiaľ nie sú dostupné, nestane sa stránka úplne neprí-

stupnou. ECMAScript je dobrým príkladom, pretože zhruba 5% užívateľov ho má v prehliadači buď vypnutý alebo ho nepodporuje a tým pádom sa stane CAPTCHA nepoužiteľnou pre túto časť populácie len z dôvodu zvolenej technológie. Obdobne sa dá pozerať na proprietárne technológie ako napríklad Adobe Flash alebo Microsoft Silverlight.

Prostredie pre CAPTCHA znamená, aké technológie sú potrebné, aby implementácia na webe fungovala.

Kapitola 5

Typy

Predchádzajúce kapitoly opisovali CAPTCHA s určitým nadhľadom a genericky, prípadne to, čo platí pre tie v súčasnosti používané. Táto kapitola má za cieľ pozrieť sa detailnejšie na jednotlivé použiteľné typy CAPTCHA, rozdelené podľa toho, aký problém umelej inteligencie využívajú. Taktiež sa pozrieme, aké výsledky dosahujú pre jednotlivé kritéria opísané v predchádzajúcej kapitole.

Zadefinujme si najprv niekoľko pojmov, ktoré nám môžu byť pri formálnom popisovaní jednotlivých typov užitočné. Tieto definície sú buď zhodné alebo vychádzajú z [vABHL03]. Nech *obrázok* je matica veľkosti $h \times w$ (výška $\times$ širka) pre nejaké $h, w \in \mathbb{N}$, kde každý záznam je trojica (R, G, B) pre $0 \leq R, G, B \leq M$, $M \in \mathbb{N}$. Nech *transformácia obrázka* je funkcia $t : o_1 \rightarrow o_2$, kde o_1 a o_2 sú obrázky veľkostí h_1, w_1 a h_2, w_2, pričom veľkosti oboch obrázkov nemusia byť rovnaké. Pod transformáciou obrázka sa chápe napríklad rotácia, zmenšenie, atď.

5.1 Rozpoznávanie textu

Optické rozpoznávanie znakov (ang. *Optical character recognition*, ďalej len OCR) je technológia transformácie obrázku obsahujúceho text do elektronickej strojovospracovateľnej textovej podoby. Najčastejšie je onen obrázok získaný naskenovaním príslušného textového dokumentu. Snáď každý, kto nejaký naskenovaný dokument nechal prejsť cez OCR softvér si všimol, že nie vždy sú všetky znaky v elektronickej podobe zhodné s tými na papieri. Úloha prepísať text z papiera do počítača, normálne tak triviálna pre človeka, je pre počítač problém. Pre neho je obrázok matica pixelov bez sémantickej hodnoty a aby v ňom ‘uvidel’ písmená, potrebuje sofistikované algoritmy. Práve preto sa táto úloha stáva dobrým základom

pre CAPTCHA.

Toto bol pravdepodobne vôbec prvý problém použitý na vytvorenie CAPTCHA, vytvorený v roku 1997 A. Broderom pracujúcim v tom čase pre vyhľadávač AltaVista. Dôvodom bolo znemožnenie pridávania URL do databázy vyhľadávača botmi. V súčasnosti tvorí tento typ drvivú väčšinu všetkých primárne používaných CAPTCHA. Jediný iný používaný typ je zvuková CAPTCHA, ktorá je však zvyčajne používaná ako alternatíva, kvôli problémom prístupnosti CAPTCHA založených na rozpoznávaní textu. Dovolím si to tvrdiť bez presných čísiel z toho dôvodu, že si nepamätám, že by som sa nejakej webovej stránke stretol s CAPTCHA iného typu, pokiaľ priamo účelom danej stránky nebola demonštrácia experimentálnej implementácie CAPTCHA iného druhu.

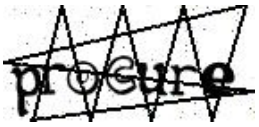
Nech I je množina všetkých možných obrázkov, L je podmnožina jazyka Σ^* nad abecedou Σ, T je množina transformácií obrázka, $g : T \times L \rightarrow I$ je funkcia generujúca obrázky. Nech $S = \{i : i \in I \wedge \exists\, t \in T, l \in l : i = g(t, l)\}$ je množina všetkých možných inštancií a $f : S \rightarrow L$ kde $\forall l \in L, t \in T : f(g(t, l)) = l$ je funkcia odpovedí. Potom tažký AI problém na ktorom stojí tento typ CAPTCHA je $P = (S, D, F)$, pričom D je distribučná funkcia ovplyvňujúca výber $l \in L$ a $t \in T$.

Povedané žargónom programovania, funkcia náhodne vygeneruje textový reťazec zvolenej dĺžky, k nemu príslušný obrázok obsahujúci daný reťazec a náhodne zvolí transformácie (prípadne ich parametre), ktoré sa následne na obrázok aplikujú.

Abeceda Σ je väčšinou tvorená znakmi a-z, prípadne aj číslami. Málokedy je vhodné kombinovať malé aj veľké písmená, pretože vznikajú problémy s čitateľnosťou - napríklad *I*, *l*, *1* (veľké i, malé l, jedna) alebo *0*, *O* (nula, veľké o) nie je v každom druhu písma jednoduché rozoznať, hlavne pokiaľ je vygenerovaný obrázok zámerne deformovaný. Taktiež nie je dobré zvoliť L len ako všetky reťazce stanovenej maximálnej resp. minimálnej dĺžky, pretože napríklad dvojica písmen *r* a *n* môže byť po transformácií nerozoznateľná od *m*. Vhodné transformácie sú napríklad deformované čiary idúce cez písmená, vlniace sa písmená, pootočené písmená, atď. Ďalej je vhodné pri rôznych inštanciách používať rôzne písma a rôzne veľkosti písma. Farba je síce často používanou formou deformácie, avšak samotná zmena farby písma alebo pozadia zmysel nemá, jedine ak estetický. Na druhej strane, vhodne generovaný 'šum' v pozadí zmysel má, i keď výrazne sťažuje čitateľnosť.

Snáď najväčším problémom pri tomto type CAPTCHA je práve nájdenie správnej rovnováhy medzi čitateľnosťou pre ľudí a dostatočnou deformáciou na zame-

dzenie čitateľnosti pre stroje. Taktiež práve popularita tejto techniky ju robí zaujímavejším cieľom pre útočníkov aj akademickú sféru než ostatné typy, čo dokazujú publikácie ako [MM03][CS05] a stránky [Hoc04][Che05][Yee05][CS05].



Obrázok 5.1: CAPTCHA pri registrácii na Slashdot

Kvalitná implementácia s dostatočne širokým záberom transformácií vedie k minimálnym hodnotám pravdepodobnosti chyby druhého druhu, avšak môže viesť k zvýšeniu pravdepodobnosti chyby prvého druhu. Práve zlepšenie technológie na prelomenie takýchto CAPTCHA viedlo k tomu, že sa začali používať transformácie výraznejšie deformujúce text ako doteraz, čo vedie k zvýšeniu chýb prvého druhu. Prístupnosť tohto typu je prirodzene obmedzená tým, že je vizuálna. HTML poskytuje možnosti ako nahradiť obrázok textom (napríklad vo forme `alt` atribútu pre element `img`), avšak poskytnutím textovej alternatívy by test stratil význam. Tento nedostatok sa dá čiastočne napraviť poskytnutím možnosti voľby medzi touto a zvukovou CAPTCHA, avšak ani to nerieši problém s prístupnosťou kompletne. Táto CAPTCHA je častokrát riešiteľná veľmi rýchlo, pretože človek text jednoducho vidí. Pokiaľ teda nie je príliš deformovaný. Pre zmenu určitou výhodou je, že široké nasadenie zabezpečilo, že keď užívateľ uvidí obrázok s deformovaným textom, už vie čo sa očakáva. Druhá vec je, že bežní užívatelia nevedia, prečo to vôbec vypĺňať musia. Náročnosť nie je veľká - jediným zádrhelom môže byť kvalitná knižnica na prácu s obrázkami. Prostredie je optimálne, pretože na fungovanie nepotrebuje tento typ žiaden iný jazyk ani technológiu ako podporu HTML, čo je samozrejme predpoklad pre každú CAPTCHA.

5.2 Vizuálne

Tento typ subjektívne predstavuje potenciálne oveľa zaujímavejšiu skupinu problémov ako tomu bolo pri CAPTCHA založených na rozpoznávaní textu. Vizuálne sú preto, že využívajú z potenciálu obrázkov viac ako len text. Človeku sa pri pohľade na obrázok okamžite vytvoria určité asociácie a pocity, ktoré môžu byť zdrojom zaujímavých návrhov pre CAPTCHA. Ako príklad tohoto typu môže slúžiť systém KittenAuth (`http://www.kittenauth.com/`), i keď formálne to CAPTCHA

nie je z dôvodu opísaného o dva paragrafy nižšie. Ide o žart a súčasne o zaujímavý nápad. Užívateľovi je zobrazených deväť obrázkov, z ktorých práve tri sú mačky a ostatné sú iné zvieratá. Úloha spočíva v označení práve spomínaných obrázkov mačiek. Jednoduché a do určitej miery zábavné. Triviálna pravdepodobnosť prelomenia 1:84 však nie je zďaleka ideálna, aj keď do určitej miery riešiteľná je, ako si povieme na konci tejto sekcie.

Obrázok 5.2: Vizuálna CAPTCHA Asirra
(`http://research.microsoft.com/asirra/`)

Do tohoto typu CAPTCHA spadajú problémy, ktoré sú založené na pochopení sémantiky použitých obrázkov. Formálnejšie povedané:

Nech I je podmnožina množiny všetkých obrázkov, T je množina transformácií obrázka, $g : I^m \rightarrow \{0,1\}^n$ pre $n \in \mathbb{N}$ sú informácie priradené jednotlivým obrázkom, $S = \{(t_1(i_1), ..., t_m(i_m)) : i_x \in I, t_x \in T \text{ pre } x = \overline{1,n}, m \in \mathbb{N}\}$ je množina inštancií, $f : S \rightarrow \{0,1\}^n$ je funkcia odpovedí, kde f je definované tak, aby platilo $f(t_1(i_1), ..., t_m(i_m)) = g(i_1, ..., i_m)$ a D je distribučná funkcia pre I a T. Potom $P = (S, D, F)$.

V podstate sme takto definovali úlohy, kde máme množinu obrázkov a pre každý obrázok uloženú n-bitovú informáciu. Dokazovateľovi ukážeme m obrázkov súčasne, pričom každý obrázok má 2^n možných odpovedí. Pre KittenAuth je $n = 1$ a je to jednoducho odpoveď na otázku "Je na obrázku mačka?" a $m = 9$ pretože jedna inštancia má deväť obrázkov. Iný typ úlohy môže byť napríklad "V akom ročnom období bola fotka odfotená?" pričom uložená informácia by bola dvojbitová.

Generovať kvalitné a pritom rôznorodé obrázky automaticky a dostatočne rýchlo softvérovo v dnešnej dobe nie je možné, preto sa vizuálne CAPTCHA musia momentálne opierať o databázu obrázkov. Na tomto mieste treba pripomenúť, že bezpečnosť CAPTCHA nesmie závisieť na vedomosti alebo nevedomosti použitého al-

goritmu a dát. Preto treba pri implementácií rátať s tým, že útočník má k dispozícií rovnakú databázu obrázkov ako my. Pokiaľ to do úvahy neberieme, môže to byť bezpečnostný problém vedúci k prelomeniu CAPTCHA bez vyriešenia AI problému, na ktorom je postavená. Princíp je triviálny a ukážeme si ho na príklade KittenAuth: útočník požiada o inštanciu, následne si vyhľadá deväť obrázkov danej inštancie v databáze a pozrie si k nim priradenú informáciu. Tie tri, pri ktorých je *true* sú tie, ktoré pošle späť ako odpoveď. Pre urýchlenie vyhľadávanie si môže dopredu vypočítať obľúbenou hašovacou funkciou odtlačky jednotlivých obrázkov v databáze a následne pri útoku vypočítať odtlačky aktuálnych obrázkov inštancie a vyhľadávať podľa nich. Tento útok je v podstate len preformulovaný problém toho, že takto navrhnutá CAPTCHA nie je jednosmerná funkcia. Preto je dôležité pri takomto používaní obrázkov z databázy vždy použiť náhodné transformácie ako zmenšenie, sploštenie, vyrezanie, pridanie šumu, zmena farebnosti a podobne.

Presnosť tohoto typu CAPTCHA závisí od zvoleného problému, avšak väčšina tých, s ktorými som sa doteraz stretol, mala príliš vysokú pravdepodobnosť chyby druhého druhu, čo môže byť zásadný problém. Je tomu tak z toho dôvodu, že väčšina týchto CAPTCHA, vrátane KittenAuth, má malú množinu možných odpovedí na konkrétnu inštanciu. Vždy je to riešiteľné minimálne jedným spôsobom, niekedy dvoma. Oba spôsoby však nie sú ideálne. Prvý spôsob je triviálny pre každú CAPTCHA, ako bolo spomínané v kapitole 2. Druhý, aplikovateľný napríklad na KittenAuth, je rozšírenie inštancie. Konkrétne by to vyzeralo tak, že by bolo viac obrázkov a prípadne väčšie percento mačiek v každej inštancii. Oba spôsoby majú samozrejme negatívny dopad na kritérium rýchlosti. Prístupnosť tohoto typu je obdobná ako u predchádzajúceho typu CAPTCHA. Rýchlosť je relatívna, avšak dovolím si tvrdiť, že pri porovnateľnom čase by užívateľ mohol mať pri riešení vizuálnych CAPTCHA príjemnejší pocit a väčšiu zábavu, pretože dovoľujú väčšiu fantáziu pri navrhovaní. Samozrejme to nie je primárna úloha CAPTCHA, avšak v praxi to význam nepochybne má. Náročnosť je zložitejšia ako u predchádzajúceho typu, pokiaľ návrh vyžaduje databázu špecifických obrázkov. Ukazuje sa, že to nemusí byť tak veľký problém, keďže na webe sú služby, ktoré toto poskytujú. Návrhy v ďalšej kapitole sú príkladom využitia takýchto služieb. Zatiaľ som sa nestretol s CAPTCHA tohoto typu, ktorá by nebola realizovateľná iba cez HTML, avšak v niektorých prípadoch má zmysel využiť skriptovací jazyk na uľahčenie ovládania, prípadne zadávania odpovede.

Trom návrhom vizuálnych CAPTCHA sa venuje [CT04]. V ďalšej kapitole sú opísané dva návrhy vizuálnych CAPTCHA.

5.3 Zvukové

Ako už bolo spomenuté, tento typ CAPTCHA je zväčša využívaný ako alternatíva predchádzajúcich dvoch typov. Používaná implementácia je formálne opísateľná rovnako ako pri CAPTCHA založených na rozpoznávaní textu, avšak použitá abeceda sú čísla, namiesto obrázkov sú generované zvuky o dĺžke niekoľkých sekúnd a množina transformácií je redukovaná na pridanie šumu do pozadia. Výsledok je obdobný, ako keď vám niekto diktuje telefónne číslo v preplnenej reštaurácií.

Zvuk musí byť na webových stránkach používaný opatrne. Častokrát totiž buď nemusí byť užívateľ v situácii, kedy je vhodné, aby jeho zariadenie vydávalo zvuk (napr. v kancelárii zdieľanej s viacerými ľuďmi, v podniku, atď.) alebo to sám nechce (napr. z dôvodu, že počúva hudbu). V prípade zvukových CAPTCHA by mal preto test vždy čakať na povel od užívateľa na spustenie a taktiež v ľubovoľnom momente mal užívateľ možnosť ho zastaviť.

Tento typ nie je predmetom útokov tak často ako ten postavený na OCR, avšak v marci 2008 sa v [Win08] objavil zaujímavý a dobre popísaný útok na zvukovú CAPTCHA využívanú Google.

Presnosť u tohoto typu je dobrá, o to viac, že sa nepoužívajú transformácie, ktoré príliš zvuk deformujú. Dá sa však očakávať, že prípadný zvýšený záujem o tento typ vyústi v podobnú situáciu ako pri CAPTCHA založených na OCR. Horšie je to s prístupnosťou, pretože tento typ je nepoužiteľný pre hluchých ľudí a ľudí so slabším sluchom. Rýchlosť je tiež v existujúcich implementáciách horšia ako pri prvých dvoch spomínaných typoch, kvôli dĺžke zvukovej ukážky. Náročnosť takejto implementácie je o niečo vyššia, čo súvisí s nutnosťou skladania, generovania a transformovania zvukov, čo nie je činnosť tak bežná ako práca s grafikou. Tento typ CAPTCHA je taktiež náročnejší na prostredie, keďže HTML neposkytuje natívnu podporu prehrávania zvukov. Tým pádom musia byť použité nástroje a technológie ako Flash, QuickTime, Silverlight a pod.

5.4 Video

Pokiaľ môžeme využiť obrázky aj zvuk na CAPTCHA, vyvstáva otázka, prečo by to nebolo možné aj s videom. Doteraz som sa však nestretol ani s jednou implementáciou či návrhom využívajúcim video. Jeden návrh založený na videu a zvuku je opísaný v nasledujúcej kapitole.

To, čo je vo všeobecnosti možné povedať o CAPTCHA využívajúcich video je,

že ‘zdedia’ všetky problémy spojené so zvukovou CAPTCHA. Ich rýchlosť je obmedzená dĺžkou videa. Prístupnosť je prinajlepšom rovnaká ako pri vizuálnych typoch. Implementačne je náročnejšie pracovať s videom ako s obrázkami. A prostredie je podobné, keďže väčšina nástrojov, ktoré sú v praxi používané na prehrávanie zvuku na webe, umožňuje aj prehrávanie videa.

5.5 Textové

Jedným z hlavných dôvodov kritiky CAPTCHA je fakt, že neexistuje jej podoba, ktorá by bola univerzálne prístupná. Toto je vlastnosť textovej CAPTCHA. Vyvstáva však otázka, či je vôbec možné, aby takáto existovala.

Abstraktne povedané, riešenie každej CAPTCHA má len dve fázy na strane dokazovateľa:

- spracovanie vstupných informácií
- nájdenie odpovede

Podstatný rozdiel je však v tom, ktorá fáza je pre ktorého dokazovateľa problémom. Pozerajme sa na vec z pohľadu človeka. Pre nás je problematická druhá fáza, teda hľadanie odpovede, a z toho dôvodu sú úlohy CAPTCHA stavané tak, aby toto nebolo náročné. Nerobí nám problém čítať z papiera (OCR CAPTCHA), nerobí nám problém rozoznať mačky od psov (vizuálna CAPTCHA), nerobí nám problém zapisovať si číslo či text, ktorý nám niekto diktuje cez telefón (zvuková CAPTCHA). Počítač má zatiaľ problém pri prvej fáze, teda extrahovať informácie z médií či pomocou vnemou, kde to človeku príde prirodzené, či už ide o obrázky, zvuky alebo video. Preto sú CAPTCHA stavané na týchto spôsoboch distribuovania vstupnej informácie. Keď ich počítač má vo vhodnej forme, tak hľadanie odpovede preňho problematické nie je.

Predstavme si, že máme k dispozícii slovník (respektíve zoznam slov) jedného jazyka, napríklad slovenčiny. Taktiež máme databázu obsahujúcu ustálené slovné spojenia či idiomy v inom (napríklad anglickom) jazyku, pričom ku každému budeme mať slovenský ekvivalent prípadne význam idiomu. Test bude spočívať v tom, že užívateľovi ukážeme anglický idiom a k nemu buď slovenskú časť záznamu pre daný idiom alebo náhodné vygenerovaný kúsok textu pomocou slovníka. Človek ovládajúci oba jazyky je schopný rozoznať, či vidí preklad alebo len náhodný zhluk slov. V súčasnosti strojový preklad nie je ešte na úrovni, aby dokázal takéto

‘zákutia’ jazyka rozumne prekladať. Takto definovaný problém nie je CAPTCHA preto, že útočník s prístupom k databáze (a toto podľa definície musíme predpokladať), by úlohu bez problémov vyriešil. Na rozdiel od obrázkov, zvuku či videa pri texte nemáme široké možnosti transformácii, prípadne tieto nezabraňujú útoku úplným preberaním. Hlavne ak má text ostať pre človeka zrozumiteľným.

Kapitola 6

Návrhy

Hovoriť o teórii, kritériach kvality či celých rodinách problémov pre CAPTCHA je dôležité, ale rovnako dôležité je mať reálne implementácie. Preto je prvá časť tejto kapitoly venovaná práve význačným existujúcim implementáciám. Druhá časť sa venuje trom vlastným návrhom, ktoré sa snažia ponúknuť nové nápady pre CAPTCHA.

6.1 Existujúce implementácie

Štúdie, ktoré by hovorili o počte webových stránok využívajúcich CAPTCHA v súčasnosti neexistujú. Avšak drvivá väčšina webových sídiel, ktoré umožňujú pridávanie komentárov, článkov, registráciu užívateľov alebo úpravu vlastného obsahu v určitom bode potrebuje ochranu proti spamu. Ten bod väčšinou príde veľmi skoro - častokrát stačí, keď je sídlo indexované vyhľadávačmi; nemusí byť ani výraznejšie populárne. CAPTCHA je jednou z foriem ochrany. Z rovnakých dôvodov ako napríklad pri kryptografických protokoloch je pre bežného autora web stránok lepšie použiť existujúcu a fungujúcu implementáciu CAPTCHA, ako sa pokúšať vytvoriť vlastnú.

Jednou z takýchto je *reCaptcha* (`http://www.recaptcha.net/`). Je to webová služba od ľudí, ktorí vymysleli názov CAPTCHA a sformalizovali ho v [vABHL03]. Založená je na rozpoznávaní textu, avšak s rozdielom, že obrázky nie sú generované z náhodného reťazca, ale sú získané digitalizáciou kníh. Dokazovateľ dostane jeden obrázok obsahujúci dve slová - pri prvom systém vie jeho textovú podobu, pri druhom nie. Text v obrázku je samozrejme umelo deformovaný, aby sa predišlo útoku použitím špecializovanejšej OCR. Na prejdenie testom

stačí prepísať len prvé slovo, avšak napísaním aj druhého užívateľ pomáha opraviť nepresnosti vzniknuté pri digitalizácii. Takže táto CAPTCHA pomáha web stránkam brániť sa pred spamom a súčasne digitalizovať knihy.

Obrázok 6.1: Webová služba reCaptcha

Zaujímavý je aj prístup služby *Mollom* (`http://www.mollom.com/`). Tá slúži na chránenie textových formulárov na stránke pred spamom. Deje sa tak analyzovaním textu, pričom výsledkom analýzy môže byť správa, že text je v poriadku, je spam alebo si je systém neistý. Pokiaľ si je neistý, tak sa užívateľovi zobrazí CAPTCHA a až po jej úspešnom zvládnutí je jeho text akceptovaný. Týmto sa majú minimalizovať chyby systému, pokiaľ si nie je istý, či je text spamom. Toto je tiež pekné využitie CAPTCHA, keďže užívateľ ju nemusí vypĺňať vždy, ale len v určitých prípadoch. Popritom sa systém nemusí obávať, že za spam označí dobré príspevky - keď si nie je istý, tak vráti CAPTCHA.

6.2 Nové návrhy

Ako už bolo spomenuté, drvivá väčšina všetkých používaných CAPTCHA sú založené na rozpoznávaní textu. Odhliadnuc od prístupnosti, vidím ako problém tohoto typu fakt, že je nezaujímavý a bežný užívateľ nerozumie, prečo vôbec musí text prepísať. Keď som nedávno vykonával malú anketu medzi priateľmi[1], pýtal som sa ich 'Vieš čo to je CAPTCHA?' Nikto nevedel. Následne som sa spýtal 'Videl si už niekedy pri písaní nejakého príspevku alebo registrácií taký obrazok s nečitateľným textom, ktorý si mal prepísať?' a štandardná odpoveď znela 'No jasne!' Taktiež typická bola však aj následná otázka 'A načo to vlastne je dobré?' čo nebolo vždy ľahké vysvetliť. Preto som začal uvažovať, či sa nedá spraviť CAPTCHA, ktorá by pôsobila priateľskejšie a pútavejšie, čo by následne aj zmiernilo problém nepochopenia.

[1] väčšinou bez informatického vzdelania, ale pravidelní používatelia Internetu

Návrhy popísané v tejto sekcii sú čiastočne alebo úplne implementované na webovom sídle *captcha.sk* (`http://www.captcha.sk/`), kde sú okrem nich aj odkazy na ďalšie materiály a implementácie, ktoré sa z rozsahových dôvodov nezmestili do tejto práce.

6.2.1 Obrázková

Do tejto kategórie spadajú dva návrhy. Prvým je CAPTCHA založená na tom, že dokazovateľovi je predostrený obrázok, ktorý bol s 50% pravdepodobnosťou otočený o 90 stupňov a otázka je 'Je obrázok otočený?' Očividne, triviálna pravdepodobnosť útočníka prelomiť túto CAPTCHA v priemere každý druhý pokus ju robí nepraktickou. Systém preto funguje ako postupnosť testov, až kým pravdepodobnosť dosiahne hranicu, ktorú považuje za dostatočnú. Takáto CAPTCHA je jednoducho implementovateľná, pokiaľ máme k dispozícii dostatočnú databázu obrázkov. Toto je riešené využitím populárnej webovej služby Flickr (`http://www.flickr.com/`), na ktorú bolo doteraz nahratých viac ako 2,5 miliardy obrázkov. Poskytuje totiž API, cez ktorú je možné robiť so službou mnoho vecí, okrem iného aj vyhľadávať obrázky podľa kľúčových slov (tzv. *tagy*). Generovanie jednej inštancie vyzerá tak, že zo zoznamu pomerne bežných slov sa jeden tag náhodne vyberie a podľa neho pomocou Flickr API systém požiada o náhodný obrázok, ktorý je daným tagom označený. Následne si systém hodí mincou, či sa obrázok má otočiť alebo nie, a táto informácia sa uloží. Potom sa obrázok pomocou náhodne zvolených transformácií mierne deformuje a nakoniec sa prezentuje dokazovateľovi.

Princíp druhého návrhu je jednoduchý - užívateľ uvidí obrázok, pod ktorým je otázka 'Aké slovo najlepšie opisuje tento obrázok?' Súčasný softvér nedokáže veľmi úspešne rozpoznať sémantický význam a objekty v obrázku. Existuje služba Alipr (`http://www.alipr.com/`), špecializujúca sa na automatické tagovanie obrázkov, ktorá občas dáva solídne výsledky, avšak obrázky musia byť relatívne veľké. Flickr je znova použitý na sprostredkovanie databázy obrázkov, pričom je možné pristupovať aj k tagom, ktorými označil autor svoj obrázok.

Aby bol takýto systém úspešný a robustný, musí byť schopný učiť sa z predchádzajúcich odpovedí. Jeden obrázok môže mať súčasne niekoľko správnych odpovedí. Ako bolo spomenuté, užívatelia Flickru môžu svoje obrázky tagovať, avšak má to dva nedostatky - tagy nemusia byť zďaleka všetky relevantné pre daný obrázok a tagy od autora obrázku môžu byť neintuitívne pre človeka, ktorý daný obrázok

nikdy predtým nevidel. Preto systém v pozadí CAPTCHA musí mať algoritmus na dynamické spravovanie tagov jednotlivých obrázkov. To najjednoduchšie pravidlo je, že pokiaľ n z m užívateľov označilo obrázok ako t, tak môžeme predpokladať, že t je pre daný obrázok relevantné. Pre malú databázu obrázkov by mohlo dôjsť k tomu, že útočníci budú vytrvalí v označovaní obrázkov dopredu dohodnutým slovom (či už by išlo o všetky obrázky zaradom, alebo by čakali na určitú podmnožinu) a následne spustia útok botmi. Avšak pre rozsiahlu databázu je veľmi nepravdepodobné, že by často dostávali rovnaké obrázky. Taktiež musia byť obrázky pred zobrazením mierne deformované, aby sa zabránilo útoku prehľadávaním databázy, sťažilo rozpoznávanie pomocou softvéru a zamedzilo útoku spomenutému o pár riadkov vyššie.

Tento systém má niekoľko nedostatkov - CAPTCHA bude (minimálne zo začiatku) označovať ľudí odpovedajúcich správne ako počítače, pretože nebude dané tagy poznať. Táto zvýšená pravdepodobnosť chyby prvého druhu by sa však dala znížiť počiatočným obmedzením náhodného výberu obrázku, respektíve veľkosti databázy, aby mal užívateľ dobrú pravdepodobnosť, že dostane obrázok, pre ktorý vôbec vieme nejaké tagy. Ďalším problémom je, že najprv vyberáme náhodné obrázky z Flickru pomocou zoznamu bežných tagov a následne chceme od užívateľov, aby zvolili vhodný tag pre obrázok. Na útok sa to zneužiť nedá, pokiaľ je interný zoznam dostatočne veľký. Taktiež je vhodné mať zoznam najbežnejších tagov, ktoré je možné aplikovať na príliš veľkú skupinu obrázkov a tieto zakázať.

6.2.2 Komiksy

Spraviť kvalitnú CAPTCHA založenú na OCR nie je ľahké. Najdôležitejšia je množina transformácií - treba mať dostatočnú diverzitu písiem, treba text a jednotlivé písmena rotovať a deformovať, pridať šum alebo pozadie, atď. Dá sa to však aj inak - spomínaná reCaptcha negeneruje obrázky z textu, ale má databázu naskenovaného textu zo starých kníh, ktoré obsahujú rôzne písma a čiastočne deformácia vznikla aj vekom. Následne sú tieto obrázky ešte mierne deformované pridaním náhodných čiar a pokryvení. Môj návrh je sčasti podobný - využiť databázu titulných strán komiksov. Dôvodov je niekoľko - komiksov je veľa druhov a ešte som nevidel dva, ktoré by mali identické písma, názvy a štýl. Ešte lepšie je, že to neplatí len pre rôzne typy komiksov, ale čiastočne aj pre rôzne výtlačky komiksov jednej značky - aj keď je väčšinou použité rovnaké písmo a podobný štýl nadpisu, je častokrát umiestnený niekde inde, na rôznych miestach niečim prekrytý, v iných farbách,

atď. Webové sídlo CoverBrowser (`http://www.coverbrowser.com/`) má databázu obsahujúcu viac ako 77000 takýchto obrázkov. Keď už mám obrázok, zvyšok testu je obdobný ako pri reCaptcha.

6.2.3 Hudba

Jednou z najviditeľnejších zmien na webe v ostatných troch rokoch je dramatický nárast popularity pozerania videa cez stránky. YouTube je v súčasnosti už jedným z najnavštevovanejších sídiel na webe, aj keď presné čísla sa líšia v závislosti od použitej metodiky. Každú minútu je naň nahratých zhruba desať hodín videa. Zaujímavý je pre nás však YouTube z iného dôvodu - stáva sa terčom súdnych sporov veľkých mediálnych a hudobných spoločností, pretože vraj porušuje autorské práva tým, že sa na jeho stránkach vyskytujú hudobné videá a seriály, ktorých práva vlastnia dotknuté spoločnosti. V súčasnosti však neexistuje automatizovaný spôsob, ktorý by efektívne vyhodnotil, či je dané video alebo hudba autorsky chránená. Z toho vznikla myšlienka, či by sa to nedalo využiť ako základ pre CAPTCHA. Prirodzene, nie autorské práva, ale hudba (resp. hudobné videoklipy).

Samotný test by bol jednoduchý a prebiehal nasledovne - zo zoznamu by sa vybrala skladba, ktorá by po miernej transformácii bola následne prezentovaná užívateľovi. Ten by mal z ukážky danej skladby byť schopný povedať, o akú hudobnú skupinu alebo skladbu ide. Transformácia by mala byť hlavne na znemožnenie vytvorenia jednoduchého odtlačku z pesničky - možnou transformáciou je úprava vybraných tónov, šum v pozadí, pridanie efektov (aby napríklad zvuk raz znel ako v koncertnej sále, inokedy ako v malej miestnosti) a vystrihnutie len určitej časti pesníčky, dlhej napríklad 30 sekúnd. Vhodné by bolo, aby sa užívateľovi pri písaní zobrazila relevantná nápoveda skupín a skladieb podľa textu, ktorý už zadal. Pomohlo by to v prípade, že si užívateľ nepamätá celý názov. Týmto by sa odstránili aj chyby vzniknuté tým, že užívateľ zadal nie úplne presný názov - napríklad by zadal ‘beatles’ a systém očakával ‘The Beatles’. Prirodzene by musel byť zoznam nápovied dostatočne dlhý, aby bola pravdepodobnosť uhádnutia zanedbateľná.

Takto fungujúca CAPTCHA by bola zaujímavá z iného pohľadu ako len zabraňovanie botom pri prístupe k službe a informáciám. Jej vyriešenie by trvalo dlhšie ako vyriešenie bežnej OCR CAPTCHA, taktiež by bola náročnejšia na implementáciu, hardvér a prenesené dáta a na fungovanie by potrebovala technológie spomínané pri zvukovej CAPTCHA. Bola by však pre užívateľa zaujímavejšia. Zároveň by mohla byť využitá napríklad pre reklamu - pod videom by bol odkaz na stránku,

kde by si užívateľ mohol zakúpiť danú skladbu alebo celý album. Samozrejme by táto linka bola funkčná až po vyriešení CAPTCHA, pretože inak by mohla byť potenciálne využitá na útok, keďže by obsahovala informácie ako názov skladby, autora, atď.

Kapitola 7

Budúcnosť

Web je v prvom rade obrovský zdroj informácií a v druhom rade obrovský biznis. Našťastie sa tieto dva ciele úplne navzájom nevylučujú. S istotou však môžeme predpokladať, že dôvody, kvôli ktorým sa v dnešnej dobe CAPTCHA používa, ešte dlho pretrvajú.

W3C, najdôležitejšia medzinárodná štandardizačná organizácia pre web, sa čiastočne vyjadrila ku CAPTCHA v [May05][1]. V dokumente sú hlavnými bodmi kritiky neprístupnosť a 'chybný pocit bezpečia'. Druhý nie je relevantný, pretože popísané chyby buď nie sú chybami CAPTCHA alebo sú chybami jednotlivých implementácií, pričom všetky sú spomínané na rôznych miestach aj v tejto práci. Prvý bod *je* dôležitý. Prístupnosť musí byť jednou z priorít všetkých technológii spojených s webom. Problém je, že CAPTCHA je to najlepšie, čo je momentálne k dispozícii a spomínaný dokument neprináša lepšie riešenia. Uvedené návrhy sú:

- logické úlohy - ťažko realizovateľný návrh. Systém by musel mať veľkú databázu takýchto úloh, alebo by musel byť schopný takéto úlohy generovať. Je otázne, koľko logických úloh, ktoré je bežný človek schopný ľahko vyriešiť, je možné zostrojiť, pokiaľ sa nemajú líšiť len v drobnostiach (od ktorých sa dá abstrahovať rozumným parsovaním). Pri generovaní úloh je zase nutné spraviť tento systém odolným proti útoku popísanom v 5.5.

- zvukový výstup - ako bolo spomenuté, CAPTCHA je možné vytvárať aj zvukové, avšak tie sú tiež neprístupné pre skupinu ľudí.

- používateľské účty s obmedzenými právami - nie vždy použiteľný prístup. Pokiaľ má anonymný užívateľ mať možnosť písať do môjho diskusného

[1] pričom je v dokumente zdôraznené, že je to len koncept a hlavne *work in progress*

fóra, tak to buď môže alebo nie.

- neinteraktívne overovanie - toto je častokrát veľmi dobrý spôsob, avšak ako súčasť CAPTCHA.

- federatívne identifikačné systémy - táto téma mnohokrát presahuje rozsahom aj zložitosťou problémy opisované v tejto práci. Takéto systémy však musia taktiež riešiť problémy, ktorým zamedzuje CAPTCHA. Pokiaľ ju nevyužijú, musia mať iný spôsob ochrany - či už pomocou biometrických alebo osobných údajov, čo prináša častokrát mnoho iných problémov (prinajmenšom správa a ochrana týchto údajov) a nemusí vždy zabezpečiť úplnu ochranu (odcudzenie pasu). Osobne si však myslím, že dôvod prečo takéto systémy nie sú náhradou CAPTCHA je ten, že každá webová služba by mala o mne mať len toľko údajov, koľko potrebuje. Metaforicky povedané, v školskej jedálni sme sa nemuseli pri vydávaní obedov preukazovať občianskym preukazom a ani nám neboli odňaté odtlačky prsta.

Takže sme v situácii, kedy si musíme vybrať medzi používaním funkčného riešenia, i keď suboptimálneho z hľadiska prístupnosti a používaním riešenia, ktoré je buď nefunkčné alebo prináša iné problémy. Zároveň, ako bolo spomenuté, dá sa neprístupnosť pomerne dobre znížiť použitím systému, ktorý dá užívateľovi na výber medzi dvoma CAPTCHA na ktoré treba rozdielne zmysly.

7.1 Neistá

Neprekvapujúcim faktom je, že počítače sa rýchle zdokonaľujú v každom aspekte. To, čo bolo prednedávnom víziou budúcnosti, bude čoskoro zastaralé. Občas však vývoj jedného vedie k zániku iného. O čo pokročí vývoj v rozpoznávaní textu, o to musí CAPTCHA založená na OCR viac deformovať generované obrázky. Problém je, že to nemôže robiť donekonečna, pretože každá ďaľšia deformácia znižuje presnosť testu. V určitom bode už bude musieť byť text tak deformovaný, aby ho softvér nerozoznal, že ho nerozoznajú ani ľudia. Toto sa nestane až keď bude OCR softvér lepší v rozpoznávaní textu ako všetci ľudia, tak ako to *Deep Blue* dosiahol v šachu, ale keď bude lepší ako *priemerný* človek. Vtedy sa stane CAPTCHA na báze OCR nepoužiteľná.

Je to trochu paradoxné, ale CAPTCHA, využívajúc problémy súčasnej AI, napomáha jej zdokonaľovaniu. Minimálne tým, že ak útočník prelomí skutočnú CAPT-

CHA, tak musel nájsť spôsob ako prekonať súčasný softvér v danej oblasti. Totiž, CAPTCHA sme definovali tak, že súčasný softvér nie je schopný dostatočne dobre riešiť problém, na ktorom test stojí.

Otázkou je, či budú nakoniec vyriešené všetky problémy umelej inteligencie. Ak áno, tak neexistuje CAPTCHA, ktorá niekedy nebude prelomená. Kedy to bude si nedovolím hádať. Avšak, ktovie, možno v tom čase problémy, ktoré podnietili vznik CAPTCHA, už nebudú aktuálne.

Kapitola 8

Záver

Internet sa stáva viac než len každodennou súčasťou našich životov. Umožňuje celej Zemi komunikovať spolu, pričom dokáže odbúravať všetky predsudky. Má potenciál byť tým najdemokratickejším systémom, aký vôbec existuje. Potenciál je jedna vec, skutočnosť iná. Jednak na Internet neustále útočia koncepty z reálneho života, ako napríklad vlády, presadzujúce opatrenia proti voľnosti Internetu v záujme 'bezpečnosti obyvateľstva', čo je dôležitá, ale odlišná téma.

Iná forma útokov sú tie v rámci Internetu. Bez ohľadu na to, či využívajú chyby v technológiách alebo neinformovanosť jeho užívateľov, mali by byť pre nás motiváciou. Technológie sú produktom ľudí a ľudia ich vedia zdokonaľovať. CAPTCHA vznikla ako reakcia na existujúci problém, ktorý bolo treba riešiť. Je vystavovaná kritike, oprávnenej aj nie, avšak faktom je, že je častokrát to najpraktickejšie riešenie. Má chyby, ale hľadajme riešenia. Je to ešte veľmi mladá téma a o to zaujímavejší problém predstavuje.

Ako bolo uvedené v úvode, cieľom tejto práce bolo objasniť pojem, históriu a dôležitosť CAPTCHA, pomenovať kritériá kvality a následne aj pomocou nich analyzovať súčasný stav a existujúce implementácie a navrhnúť nové. Tieto ciele boli splnené, avšak neobsiahli sme zďaleka všetko, čo je potenciálne pre ďalší vývoj v tejto oblasti dôležité. Ďalšie štúdium by bolo potrebné napríklad v týchto oblastiach:

- Existencia textovej CAPTCHA, prípadne univerzálne prístupnej CAPTCHA.
- Štúdie použiteľnosti a užívateľskej príjemnosti rôznych implementácií. Kombináciou zvukovej a vizuálnej CAPTCHA dosiahneme pomerne dobrý stupeň prístupnosti. Avšak funkčnosť a príjemnosť sú dva rozdielne pojmy.

Poďme hľadať spôsoby, ako spraviť CAPTCHA pre užívateľov čo najpríjemnejšou.

- Návrhy a analýza kombinovaných systémov, založených na neinteraktívnych testoch v súčinnosti s CAPTCHA.

Literatúra

[AG05] A. Signorini A. Gulli. The indexable web is more than 11.5 billion pages. Máj 2005. http://www.cs.uiowa.edu/~asignori/web-size/size-indexable-web.pdf.

[AJ06] Akamai and JupiterResearch. Retail web site performance. 2006. http://www.akamai.com/html/about/press/releases/2006/press_110606.html.

[Ber01] Michael K. Bergman. The deep web: Surfacing hidden value. 2001. http://www.brightplanet.com/images/stories/pdf/deepwebwhitepaper.pdf.

[Che05] Casey Chesnut. Using ai to beat captcha and post comment spam. 2005. http://www.brains-n-brawn.com/default.aspx?vDir=aicaptcha.

[CS05] Kumar Chellapilla and Patrice Y. Simard. Using machine learning to break visual human interaction proofs (hips). 2005. http://research.microsoft.com/~kumarc/pubs/chellapilla_nips04.pdf.

[CT04] Monica Chew and J. D. Tygar. Image recognition captchas. Technical Report UCB/CSD-04-1333, EECS Department, University of California, Berkeley, Jún 2004. http://www.eecs.berkeley.edu/Pubs/TechRpts/2004/5256.html.

[dK08] Maurice de Kunder. The size of the world wide web. 2008. http://www.worldwidewebsize.com/.

[DLW05] Ritendra Datta, Jia Li, and James Z. Wang. Imagination: A robust image-based captcha generation system. 2005.
http://infolab.stanford.edu/~wangz/project/imsearch/IMAGINATION/ACM05/datta.pdf.

[Eur06] Eurostat. Nearly half of individuals in the eu25 used the internet at least once a week in 2006. 2006.
http://epp.eurostat.ec.europa.eu/pls/portal/docs/PAGE/PGP_PRD_CAT_PREREL/PGE_CAT_PREREL_YEAR_2006/PGE_CAT_PREREL_YEAR_2006_MONTH_11/4-10112006-EN-AP.PDF.

[Eur08] Eurostat. One person in eight in the eu27 avoids e-shopping because of security concerns. 2008.
http://epp.eurostat.ec.europa.eu/pls/portal/docs/PAGE/PGP_PRD_CAT_PREREL/PGE_CAT_PREREL_YEAR_2008/PGE_CAT_PREREL_YEAR_2008_MONTH_02/4-08022008-EN-AP.PDF.

[Hoc04] Sam Hocevar. Pwntcha - captcha decoder. 2004.
http://libcaca.zoy.org/wiki/PWNtcha.

[IET99] IETF. *Hypertext Transfer Protocol – HTTP/1.1*, 1999.
http://tools.ietf.org/html/rfc2616.

[Kos] Martijn Koster. *A Standard for Robot Exclusion*.
http://www.robotstxt.org/orig.html.

[May05] Matt May. Inaccessibility of captcha. November 2005.
http://www.w3.org/TR/turingtest/.

[MM03] Greg Mori and Jitendra Malik. Recognizing objects in adversarial clutter: Breaking a visual captcha. 2003.
http://www.cs.sfu.ca/~mori/research/papers/mori_cvpr03.pdf.

[Nao96] Moni Naor. Verification of a human in the loop or identification via the turing test. September 1996.
http://www.wisdom.weizmann.ac.l/~naor/PAPERS/human.ps.

[Nie08] Jakob Nielsen. How little do users read? 2008. http://www.useit.com/alertbox/percent-text-read.html.

[Tur50] Alan Mathison Turing. Computing machinery and intelligence. 1950. http://loebner.net/Prizef/TuringArticle.html.

[vA06] Luis von Ahn. Human computation. Júl 2006. http://video.google.com/videoplay?docid=-8246463980976635143.

[vABHL00] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Captcha: Web page. 2000. http://www.captcha.net.

[vABHL03] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Captcha: Using hard ai problems for security. 2003. http://www.captcha.net/captcha_crypt.pdf.

[Web08a] Websense. Google's 'blogger' under attack by streamlined anti-captcha operations for spam. Apríl 2008. http://securitylabs.websense.com/content/Blogs/3073.aspx.

[Web08b] Websense. Google's captcha busted in recent spammer tactics. Február 2008. http://securitylabs.websense.com/content/Blogs/2919.aspx.

[Web08c] Websense. Microsoft live hotmail under attack by streamlined anti-captcha and mass-mailing operations. Apríl 2008. http://securitylabs.websense.com/content/Blogs/3063.aspx.

[Win08] Wintercore. Breaking gmail's audio captcha. Marec 2008. http://blog.wintercore.com/?p=11.

[Yee05] Howard Yeend. Breaking captchas without using ocr. 2005. http://www.puremango.co.uk/cm_breaking_captcha_115.php.