CARNet

HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

CAPTCHA

NCERT-PUBDOC-2010-06-302

Nacionalni CERT

u suradnji s

LSS

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem Nacionalni CERT kontinuirano radi.

Rezultat toga rada je i ovaj dokument, koji je nastao suradnjom Nacionalnog CERT–a i LS&S–a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

Nacionalni CERT, www.cert.hr

Nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.LSS.hr

Laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Većina administratora *web* stranica često se susreće s problemom *spama*. *Spam* može biti neželjena poruka, komentar ili bilo koji drugi oblik poruke razmijenjen preko Interneta. On prikazuje reklamu, oglas ili poveznicu na neku stranicu koja korisniku nije bitna niti ju je on zatražio. Korisnikov elektronički sandučić i komentari na blogu ili nekoj drugoj stranici mogu biti „zakrčeni" takvim porukama koje obično smetaju. U najgorem slučaju, *spam* poruke pokazuju poveznicu na sadržaje koji sadrže zloćudne programe poput virusa ili crva. Korisnik otvaranjem te poveznice i preuzimanjem sadržaja, koji može imati primamljivi izgled, preuzima zloćudni program na svoje računalo. *Spam* poruke na stranice postavljaju tzv. *botovi*, tj. automatizirani računalni programi. Kada napadač želi izvesti *spam* napad, on programira *bot* koji umjesto njega izvodi napad. *Bot* može u kratkom vremenu postaviti veliki broj komentara ili napraviti bilo koju drugu radnju za koju je programiran. U svega nekoliko minuta *web* stranica može imati tisuće *spam* komentara. *Bot* je zapravo računalni program i kada bi se mogao na neki način razlikovati stvarni čovjek (korisnik) i računalni program, moglo bi ga se spriječiti u njegovom radu.

Upravo razlikovanje ljudi i računala zadatak je CAPTCHA programa. Svaku radnju koju obavlja automatizirani računalni program (*bot*) moguće je spriječiti ako ga se traži da riješi CAPTCHA test. Ako ne riješi ispravno test, ne može obaviti daljnju radnju (poput postavljanja komentara). Računala najčešće ne prolaze na testovima jer nemaju tako dobar sustav za raspoznavanje kao ljudi. Čovjek će CAPTCHA test riješiti s lakoćom i nastaviti dalje. Upotrebe CAPTCHA programa nisu ograničene samo na postavljanje komentara i otvaranje *mail* adresa. Ostale primjene bit će navedene u ovom dokumentu, kao i načini implementacije koji su najčešće vrlo jednostavni. Ipak, nije sve savršeno, pa tako ni CAPTCHA programi. Loši CAPTCHA program se može jednostavno zaobići. U dokumentu će se također opisat dobra i ona loša svojstva CAPTCHA programa koja se mogu iskorištavati za njegovo zaobilaženje. Na kraju je navedeno nekoliko smjerova u kojima se kreće razvoj novih vrsta CAPTCHA programa.

2. CAPTCHA

CAPTCHA (eng. *Completely Automated Public Turing test to tell Computers and Humans Apart*) je program koji stvara test kojeg ljudi mogu pročitati, a računala ne mogu. Stvara se jednostavan test za koji CAPTCHA program može provjeriti točnost unesenog rješenja. Budući da računala nisu u stanju proći test, kod točnog unosa pretpostavlja se da je rješenje unio čovjek. Često se za CAPTCHA test kaže da je obrnuti Turingov test. Kod Turingovog testa računalo pokušava oponašati čovjeka pa tako ako ljudski sudac ne može prepoznati je li neku radnju obavio čovjek ili računalo, računalo prolazi na testu. Kod obrnutog Turingovog testa, sudac je računalo, a računa se na nemogućnost računala da obavi neku radnju (rješavanje CAPTCHA testa) isto kao čovjek.

Potreba za CAPTCHA programima se javila zbog ljudi koji pokušavaju iskoristiti *online* sustave poput stranica za stvaranje besplatnih e-mail adresa, *online* anketa, sustava za kupnju karata i mnoge druge koristeći *botove* (računalne programe koji se izvode samostalno). *Botovi*, automatizirano, po nekoliko tisuća puta rade istu radnju, zagušuju sustave i narušavaju kvalitetu usluge zagušujući stranice s raznim oglasima i *spam* porukama. U stanju su u velikom broju otvarati e-mail račune, kupovati ulaznice te postavljati reklame unutar komentara u jako kratkom vremenu. CAPTCHA program onemogućuje takve napade.

Zahtjevi koji se postavljaju CAPTCHA programima su sljedeći:

1. rješenje CAPTCHA testa ne smije biti uvjetovano korisnikovim jezikom i dobi,
2. CAPTCHA test se treba stvarati automatski i učinkovito,
3. provjera rješenja CAPTCHA testa treba biti jednostavna,
4. korisnikova privatnost se ne smije narušavati te
5. svima, osim ljudima, rješavanje testa treba predstavljati problem.

CAPTCHA testovi mogu biti:

- vizualni i
- auditivni

Vizualni CAPTCHA testovi korisniku prikazuju sliku, a od korisnika se traži da obavi određenu radnju na temelju prikazane slike. Najčešći tip vizualnih CAPTCHA je slika sa iskrivljenim znakovima kojima je dodan neki šum (dodatni proizvoljni elementi slike). Od korisnika se traži unos znakova sa slike. Obični OCR (eng. *Optical Character Recognition*) programi nisu u stanju prepoznati znakove sa slike pa računala ne prolaze na takvim testovima. Čovjek nema problema sa čitanjem iskrivljenih znakova, te prolazi na testu bez većih problema. Primjeri testova mogu se vidjeti na slici 1.

Slika 1. Primjer CAPTCHA testova
Izvor: Google

Kod auditivnih CAPTCHA testova potrebno je poslušati kratki audio zapis i potom unijeti nekoliko slova ili riječi kao rješenje. Izgovori znakova se nalaze u bazi podataka, a kod stvaranja testa odaberu se nasumične snimke znakova iz baze koje se na neki način izobličuju i dodaje im se pozadinski šum, kako ih računalni programi ne bi jednostavno prepoznali. Korisnik kao rješenje unosi znakove koje čuje u snimci. Drugi način je pustiti zvučnu sekvencu iz nekog filma ili neke druge snimke. Kod takvih snimki često već postoji nekakav pozadinski šum pa nema potrebe za velikim distorzijama snimke. Često se u CAPTCHA programu koriste oba oblika. Razlog je što korisnici sa slabim vidom mogu imati problema za prepoznavanjem vizualnih CAPTCHA testova te im je potrebno omogućiti alternativu.

Kako umjetna inteligencija napreduje, tako napreduju i CAPTCHA testovi. Kao nove vrste predlažu se CAPTCHA testovi sa slikama na kojima je potrebno označiti neke dijelove ili čak 3D CAPTCHA. Da bi neki CAPTCHA test bio pouzdan, prema istraživanjima stručnjaka K. Chellapilla i P. Simard, ljudi trebaju imati prolaznost od 80%, a računala ne više od 0.01%. U protivnom, zlonamjerni napadači mogu iskoristiti svoje *botove* za izvođenje napada.

2.1. Povijest

Izraz CAPTCHA su osmislili Luis von Ahn, Manuel Blum, Nicholas Hopper i John Langford sa sveučilišta Carnegie Mellon Univeristy 2000. godine. CAPTCHA je prvi put upotrijebljena kako bi se spriječilo *spam* napade na Yahoo, koji je omogućavao besplatno otvaranje e-mail računa. Veliki broj e-mail računa nisu otvarali stvarni korisnici, već *botovi* koji su ih koristili za *spam* poruke ostalim korisnicima. *Botovi* su postajali problem i za neke druge *online* usluge, poput anketa. 1999. godine postavljena je *online* anketa koje sveučilište ima najbolji studijski program. Studenti sveučilišta Carnegie Mellon i MIT-a napravili su *botove* koji su glasali za njihova sveučilišta. Na kraju se anketa svela na natjecanje tko će napraviti bolji *bot* za glasanje. Rezultat ankete je bio po nekoliko tisuća glasova za Carnegie Mellon i MIT, te svega nekoliko stotina glasova za sva ostala sveučilišta. Ovaj događaj je potegao pitanje vjerodostojnosti *online* anketa. Pomoć pri rješenju ovih problema pruža CAPTCHA jer onemogućava *botovima* izravan pristup ovakvim uslugama. Prvi CAPTCHA testovi su bili u obliku znakova koji su se nasumično odabirali i prikazivali kao slike, bez ikakvog šuma ili distorzije. Takvi CAPTCHA programi brzo su zaobiđeni upotrebom OCR sustava.

2.2. Primjena

CAPTCHA programi se koriste svugdje gdje automatizirani računalni programi (*botovi*) mogu utjecati na kvalitetu sustava. Neki od primjera upotrebe su:

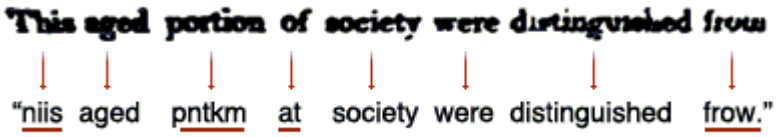
- **E-mail registracije** – mnoge stranice pružaju uslugu besplatnih e-mail adresa. Takve stranice postaju metom spamera jer oni svojim *botovima* svake minute stvaraju tisuće e-mail adresa s kojih šalju *spam* poruke. Rješenje pružaju CAPTCHA testovi koji besplatne e-mail adrese pružaju samo stvarnim korisnicima, a ne računalnim programima kojih ih zloupotrebljavaju. CAPTCHA program se postavlja na obrazac za registraciju. Kako bi registracija bila potpuna, zahtjeva se prolaz na CAPTCHA testu.
- ***Online* ankete** – već je spomenuto kako *botovi* mogu utjecati na *online* ankete. Ankete mogu izgubiti svoju vjerodostojnost jer se ne može sa sigurnošću utvrditi tko je glasao na njima - stvarni ljudi ili računala. Rješenje je da korisnik, prije nego se njegov glas prihvati u anketi, riješi CAPTCHA test i time potvrdi da nije računalni program.
- **Sprječavanje *spam* poruka u komentarima na blogovima** – bez neke vrste zaštite poput CAPTCHA programa, komentari na blogovima, ili drugim sličnim stranicama poput portala s vijestima, mogu postati zakrčeni *spam* porukama koje stvaraju *botovi*. S CAPTCHA programom, komentare mogu ostaviti samo drugi ljudi i komentari su oslobođeni raznih oglasa i *spam* poruka. Slično kao u prethodna dva slučaja, prije postavljanja komentara zahtjeva se ispravno rješavanje CAPTCHA testa. Tek nakon provjere ispravnosti rješenja, komentar će biti objavljen.
- **Sakrivanje e-mail adrese** – vrlo često korisnici imaju potrebu prikazivati svoje e-mail adrese na Internet stranicama kako bi ih drugi korisnici mogli kontaktirati. Tada pristup njihovim adresama imaju *botovi* koji šalju *spam* poruke. Upotrebom CAPTCHA programa zahtjeva se prolaz na testu kako bi se prikazala cijela e-mail adresa. Na taj način, pristup e-mail adresi je omogućen samo stvarnim korisnicima.

- **Kupovina karata preko Interneta** – u ovom slučaju se *botovi* koriste za kupnju velikog broja karata koje se kasnije prodaju po većoj cijeni. Upotrebom CAPTCHA programa smanjuje se broj karata koje se kasnije preprodaju. Zlonamjerni korisnik još uvijek može kupovati karte u cilju preprodaje, ali je broj karata koje je u mogućnosti kupiti puno manji.
- **Sprječavanje provaljivanja lozinki** – ako računalni program pokušava provaliti lozinku isprobavajući razne kombinacije znakova, moguće ga je spriječiti upotrebom CAPTCHA programa. Ideja je nakon nekoliko krivih unosa lozinke zahtijevati prolaz na CAPTCHA testu kako bi se dobio novi pokušaj unosa lozinke. Ako je čovjek unosio krive lozinke, on će proći na CAPTCHA testu i omogućit će mu se daljnji unos lozinki (pretpostavlja se da je korisnik zaboravio svoju lozinku i potrebno mu je omogućiti daljnje pokušaje). Ako je računalni program unosio netočne lozinke, vrlo vjerojatno se radi o pokušaju provaljivanja u račun korisnika i to je potrebno zaustaviti. Ovaj način je bolji od zaključavanja računa nakon nekoliko krivih unosa, jer se onemogućava namjerno zaključavanje nečijeg računa.
- ***Botovi* tražilica** – neke Internet stranice je poželjno ostaviti neindeksirane kako ih se ne bi lagano nalazilo. One u svom kodu imaju html oznaku koji sprječava *botove* tražilica u čitanju stranice. Međutim, neki *botovi* to ne poštuju i jedini način da se *botovima* onemogući čitanje stranica je postavljanje CAPTCHA testova koje ne mogu proći.

2.3. ReCAPTCHA projekt

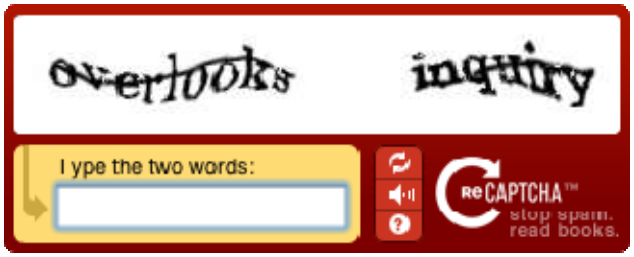
Svaki dan korisnici riješe oko 200 milijuna CAPTCHA testova. Ideja reCAPTCHA programa je nekako iskoristiti raširenost i broj ispravno riješenih CAPTCHA testova. ReCAPTHCA je besplatni CAPTCHA program sa svim svojstvima dobrog CAPTCHA programa koji istodobno radi dva posla: osim što sprječava pristup *botovima*, pomaže u digitalizaciji knjiga.

Test koji stvara reCAPTCHA je slika s dvije riječi, a za prolaz je potrebno ispravno upisati riječi na slici. Posebnost reCAPTCHA programa je ta što su riječi koje se koriste dobivene iz skenirane knjige koja se želi digitalizirati, a koje OCR program nije mogao prepoznati. Činjenica da postojeći OCR sustav nije mogao prepoznati riječ osigurava da će računalni programi imati problema s rješavanjem reCAPTCHA testa. Kvaliteta skenirane slike riječi ovisi o kvaliteti tiska, papira, starosti i istrošenosti knjige. U gornjem redu slike 2 je primjerak teksta dobivena skeniranjem. Zbog starosti papira i načina tiska, kvaliteta skeniranog teksta nije velika. U drugom redu je rezultat primijenjenog OCR programa na taj tekst. Program je prepoznao dio riječi, ali dio nije (podcrtane su crvenom crtom). Neprepoznate riječi koristi reCAPTCHA program u svojim testovima.



Slika 2. Rezultat OCR programa
Izvor: reCAPTCHA

Riječi se dodatno iskrivljuju i dodaje im se neki oblik smetnje poput valovite crte kako bi računalnim programima bilo još teže prepoznati o kojoj se riječi radi. Zbog toga je reCAPTCHA program vrlo djelotvoran u filtriranju *botova*. Postavlja se pitanje: kako se provjerava ispravnost rješenja ako se upotrebljavaju riječi koje računalo nije prepoznalo u postupku OCR-a? Odgovor je sljedeći: u svakom testu, korisniku su prikazane dvije riječi: za jednu sustav zna odgovor, a za drugu ne zna. Korisnik upisuje obje riječi. Ako je korisnik dobro upisao riječ za koju sustav zna odgovor, on prolazi na testu. Tada sustav pretpostavlja da je i druga riječ dobro upisana. Nakon što se nepoznata riječ pojavila na testovima nekoliko puta, sustav sa velikom sigurnošću može reći da zna koja je nepoznata riječ. Riječ koju OCR sustav prilikom digitalizacije nije mogao prepoznati, sada je poznata. Na taj način, sve nepoznate riječi se mogu prepoznati i cijela knjiga se digitalizira.



Slika 3. Izgled reCAPTCHA testa
Izvor: reCAPTCHA

Uz vizualni test, korisniku može odabrati alternativni audio test. Korisnik odsluša kratku rečenicu iz filma ili neku drugu snimku kojoj je dodan pozadinski šum, a kao rješenje unosi nekoliko riječi koje je prepoznao. Pri tome nije potrebno unijeti sve riječi iz rečenice, nego samo nekoliko njih.

Implementacija reCAPTCHA programa je besplatna i vrlo jednostavna. Potrebno je dodati nekoliko gotovih linija koda u postojeći kod aplikacije. Implementacija će biti objašnjenja u sljedećem poglavlju.

3. Praktične implementacije

Brojni CAPTCHA programi su besplatni, a njihova implementacija nije zahtjevna. Međutim, loša implementacija može omogućiti jednostavno zaobilaženje CAPTCHA testa o čemu će biti više objašnjeno u sljedećem poglavlju. Zato je prilikom odabira CAPTCHA programa potrebno obratiti pažnju na sigurnost implementacije.

Kod odabira CAPTCHA programa koji će se koristiti, dobro je proučiti način na koji se stvaraju testovi. CAPTCHA programe sa konačnim brojem testova treba izbjegavati. Napadač može nakon kratkog vremena popisati sve testove i s lakoćom ih rješavati. CAPTCHA programi koji imaju više različitih algoritama za stvaranje CAPTCHA testa su bolje rješenje. Da bi napad uspio, napadač treba biti u stanju zaobići svaki od algoritama, a za svaki algoritam treba imati poseban program koji ga zaobilazi. To komplicira napad i napadači često odustanu kod složenijih problema. Naravno, iskusnog napadača, koji ciljano napada stranicu, ovo neće zaustaviti.

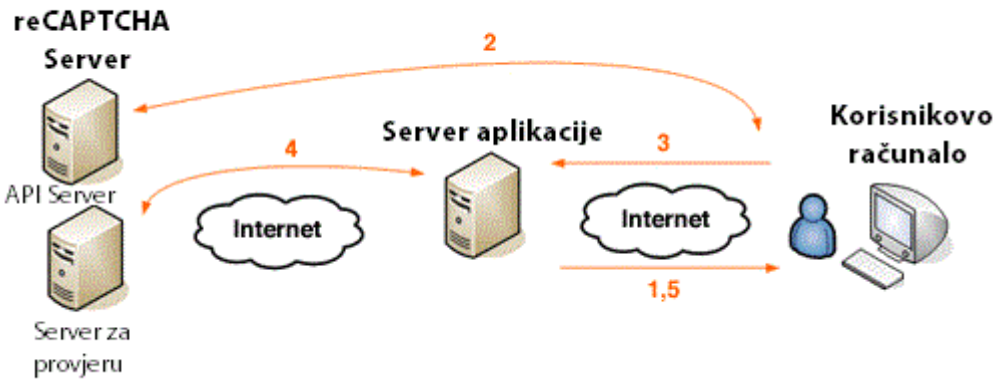
Druga stvar na koju je potrebno obratiti pažnju je način implementacije. Stvaranje CAPTCHA testa oduzima procesorsko vrijeme zbog složenih postupaka obrade slike. Što je distorzija složenija, više se procesorskog vremena troši. Zato je korisno implementirati takve CAPTCHA programe koji test ne stvaraju na poslužitelju stranice na kojoj se prikazuje CAPTCHA test, već na nekom drugom. Test se dohvaća komunikacijom između dva poslužitelja i potom prikazuje korisniku. Provjera rješenja se obavlja ili na poslužitelju stranice ili na nekom drugom poslužitelju, ovisno o implementaciji.

Korisno je implementirati CAPTCHA program koji, osim vizualnog testa, korisniku stvori audio test. Korisnici sa slabijim vidom mogu imati problema s vizualnim testom, pa je audio test korisna alternativa. CAPTCHA testovi postaju sve složeniji i potrebna je sve veća distorzija slike kako ju računalni programi ne bi prepoznali. Zbog toga čak i korisnici sa odličnim vidom imaju problema sa rješavanjem nekih CAPTCHA testova. Omogućavanje dohvata novog CAPTCHA testa je nužna mogućnost. Ako korisnik ne može riješiti prikazani test, jednostavno zatraži novi s nadom da će sljedeći moći riješiti.

3.1. ReCAPTCHA

ReCAPTCHA program jedan je od najpouzdanijih CAPTCHA programa današnjice, a njegova implementacija je krajnje jednostavna. Svodi se na registriranje na reCAPTCHA stranici, dohvat reCAPTCHA biblioteke i dodavanje nekoliko linija koda u postojeći kod. Gotove biblioteke postoje za programske jezike PHP, ASP.NET, Python, Ruby i mnoge druge. Na stranicama su navedene upute kako dodati reCAPTCHA program u aplikacije kao što su WordPress, phpBB, MediaWiki i mnoge druge. Detaljne upute se mogu pronaći na:

http://recaptcha.net/resources.html



Slika 4. Komunikacija u reCAPTCHA programu
Izvor: reCAPTCHA

Svim programskim implementacijama je zajednički reCAPTCHA API (eng. *Application Programming Interface*) preko kojeg se dohvaćaju svi testovi. Zbog toga je *web* aplikacija oslobođena stvaranja reCAPTCHA testa koje inače zahtjeva dosta procesorskog vremena. Točnost rješenja ne provjerava se na poslužitelju *web* stranice na kojoj je reCAPTCHA program, već na posebnim reCAPTCHA poslužiteljima.

Postupak dohvata i provjere točnosti reCAPTCHA testa prikazan je dijagramom na slici 4. Postupak je sljedeći:

1. Korisnik dohvaća *web* stranicu koja sadrži reCAPTCHA program.
2. Korisnikov preglednik šalje zahtjev za reCAPTCHA testom. API poslužitelj korisnikovom pregledniku predaje reCAPTCHA test i token koji ga identificira .
3. Korisnik rješava reCAPTCHA test i šalje rezultat poslužitelju aplikacije koju koristi, zajedno s tokenom testa. Taj poslužitelj rješenje i token prosljeđuje reCAPTCHA poslužitelju za provjeru.
4. ReCAPTCHA poslužitelj provjerava točnost rješenja i šalje rezultat (točno/netočno rješenje).
5. Ako je rješenje ispravno, aplikacija dozvoljava korisniku daljnji rad, a ako nije korisniku se može dopustiti ponovni pokušaj s novim testom.

Prvi korak u postavljanju reCAPTCHA programa na *web* stranicu je registracija na reCAPTCHA portalu:

https://www.google.com/recaptcha/admin/create

Registracijom se dobivaju dva ključa: javni i privatni. Korištenjem ovih dodatnih ključeva sprječava se „krađa“ rješenja reCAPTCHA testova s nečije tuđe stranice. Prilikom registracije potrebno je navesti ime domene na kojoj će se reCAPTCHA program koristiti. Ključevi vrijede samo za tu domenu i sve njene poddomene. Na primjer, ako se registrira „domena.hr“, ključevi vrijede i za poddomenu „test.domena.hr“. Međutim, moguće je napraviti takve ključeve koji će vrijediti za više domena, a savjet koji prolazi iz prakse kaže da se napravi po jedan ključ za svaku domenu (dodatna sigurnost).

Zatim je potrebno napisati kod koji dohvaća reCAPTCHA test na *web* stranici i prikazuje ju korisniku. Dohvat reCAPTCHA testa može se napraviti na tri načina:

1. koristeći API– najjednostavniji i najčešće korišteni način.
2. koristeći API bez JavaScripta – ako korisnikov preglednik nema podršku za JavaScript.
3. koristeći AJAX API – koristan za dinamičko dodavanje reCAPTCHA programa.

Prva dva načina se najčešće implementiraju zajedno. Podrazumijevano se koristi prvi način, a u slučaju da je JavaScript kod korisnika onemogućen, koristi se drugi način. Za implementaciju ova dva načina potrebno je dodati kod prikazan u nastavku poglavlja (prvi dio). On samo dohvaća reCAPTCHA test i prikazuje ga korisniku. Za provjeru ispravnosti rješenja dodaje se drugi kod.

```
<script type="text/javascript"
   src="http://api.recaptcha.net/challenge?k=<your_public_key>">
</script>

<noscript>
   <iframe src="http://api.recaptcha.net/noscript?k=<your_public_key>"
       height="300" width="500" frameborder="0"></iframe><br>
   <textarea name="recaptcha_challenge_field" rows="3" cols="40">
   </textarea>
   <input type="hidden" name="recaptcha_response_field"
       value="manual_challenge">
</noscript>
```

Navedeni kod se umeće u <*form*> oznaku. U <*script*> oznaci je prvi način dohvata reCAPTCHA testa, a u <*noscript*> drugi način (za slučaj kada nije omogućen *JavaScript*).

Polje *„recaptcha_challenge_field"* je skriveno polje koje opisuje postavljeni reCAPTCHA test. To je token testa koji je poslužitelju potreban za provjeru rješenja.

Polje *„recaptcha_response_field"* je tekstualno polje u kojem se nalazi korisnikov odgovor. Umjesto *„your_public_key"* u kod se upisuje javni ključ dobiven registracijom.

Ovaj dio koda dohvaća reCAPTCHA test na *web* stranicu, prikazuje ga korisniku i omogućuje mu unos rješenja. Da bi se provjerila ispravnost korisnikova rješenja potrebno je povezati poslužitelj *web* stranice s poslužiteljem za provjeru reCAPTCHA testa (inače će se reCAPTCHA test prikazati, ali se rješenje neće moći provjeriti). Provjeru je moguće obavljati na dva načina:

- koristeći već gotove programske *dodatke* ili
- napraviti vlastite funkcije koje obavljaju taj posao.

U nastavku će biti objašnjenje gotove reCAPTCHA implementacije za PHP i ASP.NET.

3.1.1. PHP implementacija reCAPTCHA programa

PHP je skriptni jezik namijenjen stvaranju dinamičkih *web* stranica. PHP kod se umeće unutar HTML koda stranice, te je zbog toga dodavanje novog PHP koda u postojeći sadržaj jednostavno. Implementacija reCAPTCHA programa u PHP-u je vrlo jednostavna. Za početak je potrebno dohvatiti biblioteku sa stranice:

http://code.google.com/p/recaptcha/downloads/list?q=label:phplib-Latest

Arhivu je potrebno otpakirati i kopirati datoteku *recaptchalib.php* u direktorij gdje se nalazi kod stranice na koju se želi postaviti reCAPTCHA program. U njoj se nalaze sve potrebne funkcije za komunikaciju s reCAPTCHA poslužiteljem koji provjerava rješenje testa. Zatim je potrebno obaviti registraciju na reCAPTCHA stranici (poveznica na stranicu za registraciju nalazi se u poglavlju 3.1) kako bi se dodijelili javni i privatni ključevi koji se kasnije koriste. Potom se pristupa izmjenama postojećeg koda *web* stranice koja se želi zaštiti reCAPTCHA programom. Prvi kod dohvaća i prikazuje reCAPTCHA test:

```
require_once('recaptchalib.php');
$publickey = "..."; // javni ključ dobiven pri registraciji
echo recaptcha_get_html($publickey);
```

Sljedeći kod služi za provjeru korisnikovog rješenja. Dodaje se u dio koda koji obrađuje korisnikov zahtjev na *web* stranici (dodavanje komentara, glasanje na anketi i sl.) zaštićen reCAPTCHA programom.

```
require_once('recaptchalib.php');
$privatekey = "...";  // privatni ključ dobiven pri registraciji
$resp = recaptcha_check_answer ($privatekey,
                                $_SERVER["REMOTE_ADDR"],
                                $_POST["recaptcha_challenge_field"],
                                $_POST["recaptcha_response_field"]);
if (!$resp->is_valid) {
  die ("reCAPTCHA nije ispravno unesena. Vratite se i probajte
           ponovo." ."(reCAPTCHA said: " . $resp->error . ")"); }
```

U navedeni kod dodaju se ključevi dobiveni prilikom registracije na označenim mjestima. Koriste se dvije funkcije: *„recaptcha_get_html"* i *„recaptcha_check_answer"*. Prva dohvaća reCAPTCHA test i prikazuje ga u HTML obliku korisniku. Argumenti funkcije prikazani su u sljedećoj tablici.

Argument	**Tip**	**Opis**
$pubkey	*string*	Javni ključ dobiven registracijom.
$error	*string*	Opcionalan uvjet. Ako je uključen, prikazat će se broj greške.
$use_ssl	*boolean*	Ako se reCAPTCHA test prikazuje na stranici preko SSL-a potrebno je ovo polje postaviti na *'true'* kako se ne bi pojavila poruka o grešci u korisnikovom pregledniku. Inače je parametar postavljen na *'false'*.

Tablica 1. Parametri funkcije „recatpcha_get_html"

Funkcija vraća tip *string* koji sadrži HTML kod koji se postavlja na stranicu kako bi se reCAPTCHA test prikazao. Druga funkcija (*„recaptcha_check_answer"*.) provjerava točnost unesenog rješenja. Argumenti su prikazani u tablici u nastavku.

Argument	**Tip**	**Opis**
$privkey	*string*	Privatni ključ dobiven registracijom.
$remoteip	string	Korisnikova IP adresa.
$challenge	*string*	Token reCAPTCHA testa. Može se naći u polju *„recaptcha_challenge_field"*.
$response	*string*	Uneseno rješenje. Nalazi se u polju *„recaptcha_response field"*.

Tablica 2. Parametri funkcije „recaptcha_check_answer"

Funkcija vraća tip *boolean* koji daje informaciju je li uneseno rješenje točno ili netočno i string *error* koji daje broj pogreške ako je došlo do nje.

Vizualni izgled testa moguće je mijenjati, a sve informacije vezane uz PHP implementaciju nalaze se na adresi:

http://recaptcha.net/plugins/php/

3.1.2. ASP.NET implementacija reCAPTCHA programa

ASP.NET implementacija je jako slična PHP implementaciji. reCAPTCHA biblioteka za ASP.NET može se dohvatiti sa stranice:

http://code.google.com/p/recaptcha/downloads/list?q=label:aspnetlib-Latest

Nakon preuzimanja .NET bibilioteke potrebno je dodati referencu *„library/bin/Release/Recaptcha.dll"* na *web* stranicu koja prikazuje reCAPTCHA test. Kako bi se dobili javni i privatni ključevi potrebna je registracija (poveznica na stranicu za registraciju nalazi se u poglavlju 3.1). Potom se može pristupiti izmjeni postojećeg koda. Na vrh *aspx* stranice na koju se dodaje reCAPTCHA program dodaje se sljedeća linija koda:

```
<%@ Register TagPrefix="recaptcha" Namespace="Recaptcha"
Assembly="Recaptcha" %>
```

U oznaku <*form runat="server"*> dodaje se sljedeći kod:

```
<recaptcha:RecaptchaControl
  ID="recaptcha"
  runat="server"
  PublicKey="..."      // javni ključ dobiven registracijom
  PrivateKey="..."     // privatni ključ dobiven registracijom
/>
```

Sve ostalo obavljaju funkcije u biblioteci. Dodatne informacije o implementaciji reCAPTCHA programa na ASP.NET stranicama mogu se naći na adresi:

http://recaptcha.net/plugins/aspnet/

3.2. Ostale implementacije

Na Internetu se mogu naći brojne gotove implementacije. Dok su neke besplatne, neke se plaćaju ali i koriste složenije CAPTCHA testove koje je teže zaobići računalnim programom. Kod nekih implementacija testovi se stvaraju na poslužitelju web stranice, dok se kod nekih gotovi testovi samo dohvaćaju (sa udaljenih poslužitelja). U nastavku je popis poznatijih implementacija.

Programsko okruženje	Plaćanje	Poveznica
PHP, AJAX	ne	http://www.webcheatsheet.com/php/create_captcha_protection.php
PHP	da	http://www.sevenscript.net/?p=48
ASP.NET	da	http://captcha.biz/
PHP	ne	http://captchas.net/sample/php/
ASP	ne	http://captchas.net/sample/asp/
Python	ne	http://captchas.net/sample/python/
Perl	ne	http://captchas.net/sample/perl/
JSP	ne	http://captchas.net/sample/jsp/

Tablica 3. Pregled nekih od CAPTCHA implementacija

Upute za izradu sustava za stvaranje i vrednovanje CAPTCHA testova mogu se naći na brojnim stranicama. Neke od njih su:

http://www.icemelon.com/tutorials/23/Create_CAPTCHA_images.htm

http://www.webcheatsheet.com/php/create_captcha_protection.php

4. Zaobilaženje CAPTCHA sustava

Zaobilaženje CAPTCHA programa može se izvesti na nekoliko načina. Najčešće se iskorištavaju nedostaci u dizajnu CAPTCHA programa poput ugradnje u *web* stranicu ili korištenje jednostavnih CAPTCHA testova koji se mogu proći s naprednijim sustavom za raspoznavanje znakova. Drugi način je da stvarni korisnici rješavaju testove, a da nisu svjesni da na taj način pomažu *botovima* u napadu ili su plaćeni da ih rješavaju.

Budući da *botovi* mogu u nekoliko minuta stvoriti tisuće zahtjeva za CAPTCHA testom, prolaznost od tek 10% na testu je dovoljno za *spam* napad. Dakle, ako je *bot* u stanju proći na testu jednom od svakih 10 pokušaja, CAPTCHA test nije zadovoljavajući i neće spriječiti napad. Zbog toga se stalno razvijaju novi testovi.

4.1. Nedostaci u implementaciji

Jedan od glavnih zahtjeva za dobar CAPTCHA program je mogućnost uzastopnog stvaranja drugačijih testova. Ako program ima konačan broj testova, napadač će nakon nekog vremena imati popis svih testova i njihovih odgovora. Za prolazak na testu biti će dovoljno pogledati u popis i prepisati rješenje s njega. Ovakvi CAPTCHA programi neće zaustaviti *botove* u njihovim napadima.

Drugi način zaobilaženja je iskorištavanje identifikatora sjednice (eng. *session ID*). Veliki broj CATPCHA programa ne uništava identifikator sjednice kada se unese ispravno rješenje. Korištenjem identifikatora sjednice za već jednom ispravno riješeni CAPTCHA test, mogu se automatizirati zahtjevi na stranici zaštićenoj CAPTCHA programom. Kao primjer se može uzeti kupnja karata preko Interneta. Zlonamjerni korisnik želi kupiti nekoliko karata koristeći *bot* koji automatizirano kupuje karte. Prije kupnje jedne karte, sustav traži ispravno rješenje CAPTCHA testa. Korisnik riješi test te zabilježi identifikator sjednice i rješenje testa. Nakon što sustav provjeri rješenje i utvrdi da je ispravno, korisniku se dopušta kupnja karte. Nakon ovog postupka *bot* može automatizirano dalje kupovati karte. Jednostavno pošalje zabilježeni identifikator sjednice i rješenje s drugim korisničkim podacima te kupi novu kartu. Ako je CAPTCHA program loše implementiran i dopušta ovako opisano zaobilaženje, *botu* će biti dovoljno riješiti jedan CAPTCHA test i zapamtiti identifikator sjednice, a on dalje može automatizirano slati zahtjeve sve dok ne istekne ta sjednica. Problem se može jednostavno riješiti. Pseudokod rješenja koje je ranjivo na ovakav napad je sljedeći:

```
if form_submitted and captcha_stored!=""
   and captcha_sent=captcha_stored then
process_form();
endif:
```

Pseudokod ispravljenog koda koji više nema ovakvih problema se od prošloga razlikuje u samo jednom redu. U tom redu se u varijablu zapisuje posljednje korisnikovo rješenje. Ako je trenutačno korisnikovo rješenje jednako posljednje zapisanom rješenju u *captcha_stored*, korisnikov zahtjev se ne obrađuje.

```
if form_submitted and captcha_stored!="" and
   captcha_sent=captcha_stored then
captcha_stored="";
process_form();
endif:
```

Dobri CAPTCHA programi ne dozvoljavaju više pokušaja rješenja za isti CAPTCHA test. Ako uneseno rješenje nije ispravno, korisniku se prikazuje novi CAPTCHA test. To sprječava isprobavanje rješenja dok se ne nađe ispravno. Većina CAPTCHA programa ima implementiranu ovakvu zaštitu.

Neki sustavi, za provjeru ispravnosti rješenja, koriste kriptografski sažetak (eng. *hash*) ispravnog rješenja testa. Sažetak se prosljeđuje poslužitelju koji provjerava korisnikovo rješenje. Prilikom provjere, korisnikovo rješenje se uspoređuje s ispravnim rješenjem koje poslužitelj za provjeru određuje iz sažetka. Često je sažetak dovoljno malen da ga se može dešifrirati i preko njega saznati točno rješenje pa se savjetuje korištenje „dugačkih sažetaka" (npr. duljine 2048 ili 4096 bitova).

4.2. Napredni sustavi za raspoznavanje znakova

Standardni OCR (eng. *Optical Character Recognition*) program nije u stanju prepoznati vizualni CAPTCHA test. Međutim, s nešto naprednijim sustavom za raspoznavanje moguće je CAPTCHA testove rješavati s velikom prolaznošću. Raspoznavanje znakova CAPTCHA testa se može podijeliti na tri faze:

1. predprocesiranje,
2. segmentacija i
3. klasifikacija.

Prva i treća faza nisu problem za računala. U predprocesiranju se uklanja pozadina i što je moguće više šuma, dok se u klasifikaciji prepoznaju segmentirani znakovi. Ono što predstavlja najveći problem računalima i zašto ne prolaze većinu testova jest segmentacija. U toj fazi svaki znak je potrebno odijeliti (segmentirati) od ostalih kako bi ga klasifikator mogao prepoznati. Čovjek nema problema sa segmentacijom znakova. Čak i kada se znakovi djelomično preklapaju ili su znakovi isprekidani, ljudski mozak je u stanju prepoznati znak. Računala s ovim situacijama imaju problema.

Klasičan izgled vizualnog CAPTCHA testa je slika sa znakovima koje treba unijeti. Ti znakovi su iskrivljeni, rotirani ili se preklapaju. Njima je dodan neki oblik šuma poput valovitih linija, kružnica, dijelova pozadine u drugim bojama ili nešto slično kako bi segmentacija bila što složenija. Takva distorzija slike često dovodi do situacija kada niti čovjek nije u mogućnosti proći test.

Slika 5. Primjer neprepoznatljivog CAPTCHA testa
Izvor: top 10 worst captchas

Ipak, napredni sustavi za raspoznavanje, iskorištavajući razna svojstva slike koja se daje na test, mogu imati dovoljnu prolaznost na testovima da bi bili iskoristivi. Osobine koje se iskorištavaju u ovakvim sustavima su:

- **Znakovi u slici tvore riječi nekog jezika** – ako se koriste riječi, sustav može isprobavati sve riječi iz rječnika jezika dok ne nađe točnu riječ. Rješenje je u stvaranju nasumično odabranih znakova, ne nužno slova.
- **Fiksna dužina riječi u slici** – ako je broj znakova koje se treba unijeti kao rješenje fiksan, broj mogućih kombinacija je puno manji nego kada se koriste riječi različitih duljina. Varijabilnom dužinom riječi povećava se broj riječi koje se mogu koristiti i teže je pronaći pravu kombinaciju znakova.
- **Pozadina i šum su svjetliji od znakova** – često su znakovi najtamniji dijelovi slike. Odvajanje znakova od pozadine i šuma je moguće izvesti jednostavnim postupkom koji se naziva *tresholding*. Kod ovog postupa, odredi se granica (*treshold*) i sve ispod te granice se odbacuje. Ako su znakovi najtamniji dijelovi slike, granica se postavi na neku tamnu boju. Sve svjetlije od te boje se odbacuje i kao rezultat ostaju samo znakovi, bez šuma. Rješenje je stvoriti sliku u kojoj je šum iste svjetline kao znakovi. Često se u pozadinu postavlja mreža (*gridline*) koja je iste boje kao slova. Time se otežava daljnja segmentacija.
- **Znakovi iste boje** – ako su svi znakovi iste boje i ta boja je drugačija od šuma, odvajanje šuma od znakova ne predstavlja problem. Šum više neće otežavati segmentaciju znakova.
- **Znakovi istog stila (*fonta*)** – ako slovo A izgleda isto u svakoj slici, segmentacija slova, a pogotovo njegova klasifikacija nije nikakav problem. Rješenje je upotrebljavati različite stilove

za prikaz znakova. Što su stilovi različitiji, klasifikacija znakova će biti teža jer je potrebna veća baza znakova s kojima se uspoređuje znak prilikom klasifikacije.

- **Znakovi su orijentirani u istom smjeru** – ovo svojstvo se koristi kod klasifikacije. Ako svi znakovi stoje uspravno, jednom segmentirani, nisu problem za klasifikator. Ako su svi orijentirani u istom smjeru, ponovo, klasifikator će moći prepoznati slova. Ali ako je svaki znak usmjeren u svom smjeru, on će predstavljati problem za klasifikator i bit će ga teže segmentirati. Znakovi rotirani za različite kutove su dobro svojstvo.
- **Znakovi se ne preklapaju** – preklapanje je korisno svojstvo za pouzdanije CAPTCHA testove. Ljudsko oko u pravilu nema problema sa znakovima koji se dijelom preklapaju. Računalo najviše problema sa segmentacijom ima upravo ako se znakovi preklapaju. Ono ne može odrediti je li to jedno veliko slovo ili više slova koja su jako blizu. Ako slova budu pogrešno segmentirana, klasifikator ih neće moći prepoznati i računalo neće proći na testu.
- **Znakovi nisu isprekidani** – ljudsko oko može vidjeti stvari koje u stvarnosti ne postoje, ali mozak zna da bi trebale postojati. Isprekidanu crtu mi doživljavamo jednako kao punu crtu, tako da čovjek neće imati problema s prepoznavanjem znaka kojem su crte isprekidane. Računalo takav znak nije u stanju prepoznati.

Iz ovih osobina moguće je zaključiti koliko je neki vizualni CAPTCHA test pouzdan. Međutim, vizualni i audio CAPTCHA testovi često idu zajedno. Zbog toga je pažnju potrebno posvetiti i audio testu. Vizualni test može biti najbolji na svijetu, ali to ništa ne znači ako je audio test koji se može paralelno koristiti jednostavan za zaobilaženje. U audio CAPTCHA programima je potrebno unijeti riječi iz audio zapisa. Njih je također moguće zaobići koristeći naprednih programa za raspoznavanje. Ako se u audio zapisu nalazi izgovor zasebnih znakova, zaobilaženje testa je olakšano. Znakove je jednostavno segmentirati jer su međusobno odvojeni s kratkom pauzom. Jednom segmentirane znakove nije više teško prepoznati. Najčešće se prvo primjeni FFT (eng. *Fast Fourier Transformation*) transformacija na snimku kako bi se ona prikazala u frekvencijskoj domeni, a zatim se prikaže njen spektrogram. Preko spektrograma se mogu relativno jednostavno odrediti znakovi, jer izgovor svakog slova ima drugačiji spektrogram. Teži za zaobilaženje su testovi koji kao audio snimku preuzimaju dio stvarne snimke poput audio sekvence iz filma ili neke pjesme. One već sadrže prirodni šum, a slova nisu jasno odvojena što otežava segmentaciju.

4.3. Ručno rješavanje CAPTCHA testova

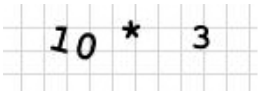
Neki *botovi*, za prolazak CAPTCHA testova, koriste ljude koji su plaćeni za svaki riješeni CAPTCHA test. Postoje tvrtke u Kini i Indiji čiji zaposlenici rješavaju po 1000 CAPTCHA testova za jedan dolar. Drugi spameri stvaraju stranice s velikom posjećenosti za čiji pregled je potrebno riješiti CAPTCHA test. Rješenje *bot* koristi za svoj napad. U ovom slučaju, korisnici najčešće nisu svjesni da pomažu u napadu jednog *bota*. Kao stranice s velikom posjećenosti najčešće se koriste *webproxy* poslužitelji. Kada je korisnicima zabranjena posjeta neke stranice s računala na kojem rade, oni koriste *webproxy* kako bi došli do zabranjene stranice. Jedan *webproxy* može imati veliki broj posjeta i ako se za njegovo korištenje zahtjeva rješenje jednog jednostavnog CAPTCHA testa od stvarnog korisnika, vlasnik *webproxya* može to iskoristiti da njegovi *botovi* zaobiđu CAPTCHA program na stranici koju napada.

Kod ovakvog načina zaobilaženja CAPTCHA testa nema djelotvorne zaštite. Na sreću, ovaj način nije toliko popularan kao prva dva jer zahtjeva znatno više napora. Potrebno je osmisliti i održavati stranicu koja će imati veliki promet ili je potrebno plaćanje nekog iznosa za radnike koji rješavaju CAPTCHA testove. Taj iznos nije velik, ali je ipak iznos koji treba platiti.

5. Budućnost

Kako umjetna inteligencija računala napreduje i CAPTCHA programi se moraju unaprjeđivati. Potrebno je pratiti koliko je CAPTCHA program djelotvoran. Razvojem sustava za prepoznavanje, CAPTCHA koja je donedavno bila jedna od najpouzdanijih, može postati ranjiva.

Distorzija znakova u vizualnim CAPTCHA testovima dolazi do granica ljudske mogućnosti raspoznavanja. Ako čovjek više ne može proći na testu, takva CAPTCHA je jednako loša kao kada bi ju svako računalo moglo proći. Zato se istražuju novi oblici CAPTCHA programa poput matematičkih i logičkih problema, prepoznavanja dijelova slike i sl. Kod matematičkih i logičkih problema postoji problem da ljudi sa slabijim mentalnim sposobnostima neće biti u stanju proći takav test. Takav CAPTCHA program ne bi bio pogodan za korištenje.



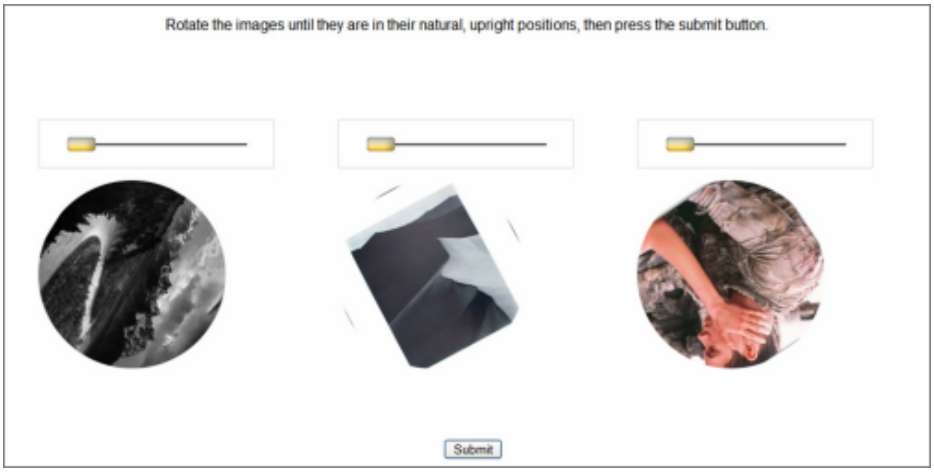
Slika 6. Matematički problem u CAPTCHA testu
Izvor: Google

Predlažu se CAPTCHA programi koji bi stvarali testove s nekoliko stvarnih slika i pitanjem. Pitanje je takvog oblika da je na slikama potrebno pronaći objekt i označiti ga. Cilja se na lošu sposobnost računala da segmentira dijelove slike i prepozna što je na njima. Problem je što su pitanja najčešće na engleskom jeziku što ograničava područje primjene takvih testova.

Slika 7. Nova vrsta CAPTCHA programa
Izvor: captcha

Još jedan oblik CAPTCHA programa koji koristi stvarne slike zahtjeva od korisnika da odredi geometrijsku sredinu nekog od objekata u slici. Čovjek jednostavno razdvaja pojedine objekte od okoline i intuitivno određuje središte tog objekta. Već je rečeno koliko problema računalo ima sa segmentacijom slike, tako da bi ovakav oblik CAPTCHA testa računalu predstavljao veliki problem.

Kao mogući izgled novih CAPTCHA programa Google predlaže test na kojem je objekt rotiran za neki veći kut. Od korisnika se traži da objekt vrati u pravilni položaj. Na primjer, na slici je drvo koje je rotirano tako da je njegova krošnja na dnu. Čovjek bez problema vraća drvo u pravilan položaj s krošnjom prema gore. Računalo nema informaciju da je to drvo, pa niti ne zna u koju stranu ta slika treba biti orijentirana. Moguće ga je 'naučiti' kako drvo izgleda i kako treba biti orijentirano, ali ako na sljedećem CAPTCHA testu ne bude prikazano drvo, računalo ponovo neće znati dobro riješiti test. Prednost ovog CAPTCHA programa je što ne ovisi o jeziku korisnika, ne zahtjeva veliku mentalnu sposobnost i moguće je brzo stvoriti neograničeni broj testova . Nedostatak ovog programa se može javiti ako se koriste slike ljudi. Već postoje računalni programi koji su sposobni prepoznati ljude na slikama. Nakon toga, vraćanje u pravilan položaj nije problem, jer se ljudska glava na većini slika nalazi 'gore'. Sličan problem se javlja ako je na slici nebo. Segmentacija neba, zbog njegove boje, računalima u pravilu ne predstavlja problem. Kako je nebo uvijek na vrhu slike računalo će bez problema orijentirati sliku u pravom smjeru. Zato se potencijalne slike za testove prvo ispituju posebnim programima kako bi se provjerila sposobnost računala da ga prođu. Slike koji takav program uspije riješiti ne prikazuju se u CAPTCHA testovima ovog tipa.



Slika 8. CAPTCHA test sa rotiranjem
Izvor: Google

Microsoft je predložio svoju inačicu CAPTCHA programa koji se također temelji na realnim slikama. Program se naziva Asirra (eng. *Animal Species Image Recognition for Restricting Access*). Korisniku je prikazan niz nekoliko slika mačaka i pasa. Kao rješenje je potrebno označiti sve slike mačaka. Da bi ovaj CAPTCHA program funkcionirao potreban je veliki broj slika mačaka i pasa. Ako je baza slika malena, sustav se zaobilazi bez problema jednom kada se sve slike iz baze označe kao slike psa ili mačke. Kako bi se to izbjeglo, Asirra projekt je u partnerstvu sa Petfinder.com, stranicom koja pomaže udomljavanju mačaka i pasa. Petfinder.com opskrbljuje Asirra program sa slikama životinja, a Asirra program prikazuje slike životinja kojima je potreban dom tisućama korisnika. Ispod svake slike se nalazi poveznica preko kojeg korisnik može udomiti životinju sa slike. Na slici 9 prikazan je jedan Asirra test. Prelaskom miša preko sličice životinje otvara se uvećana slika i poveznica preko kojeg se može tu životinju udomiti. Klikom miša na sličicu moguće ju je označiti. Za prolazak na testu potrebno je označiti ili sve slike pasa ili sve slike mačaka. Trenutačno, Asirra raspolaže sa preko 3 milijuna slika mačaka i pasa. To je dovoljan broj slika da spriječi većinu napada, ali ne sve. Napravljeni su programi koji na temelju boje i teksture krzna životinja sa 10.3% točnošću prolazi Asirra test. To nije puno, ali je dovoljno za *spam* napad. Također, napadač može nekome platiti recimo 1 cent za svaku sliku koju klasificira. Za 30000 dolara cijela se baza može klasificirati i dalje s lakoćom zaobilaziti.

Slika 9. Asirra test
Izvor: Microsoft research

Umjesto CAPTCHA testova sa slikama predlažu se CAPTCHA testovi s videom. Korisniku se prikazuje video na kojem se slova pomiču. Kao rješenje korisnik treba unijeti slova. Nažalost, video CAPTCHA test se može svesti na problem slikovnog CAPTCHA testa. Video je zapravo niz sličica. Rastavljanjem na sličice dobivamo više problema slikovnog CAPTCHA testa koja računala sve lakše rješavaju.

Još jedna obećavajuća ideja je 3D CAPTCHA test. U testu se prikazuje nekoliko 3D modela životinja koji se rotiraju, te jedna slika modela životinje koju treba naći među ostalima koji se rotiraju. Budući da ovako modelirane životinje nemaju dodane teksture i boje krzna preko kojih bi ih računalni program mogao prepoznati, računalo ne zna koja je životinja na kojoj slici. Kako se životinje i rotiraju, nije moguće

uspoređivati sličicu po sličicu sa životinjom koju treba pronaći. Algoritmi koji bi mogli zaobići ovakav CAPTCHA test su vrlo složeni i nisu do kraja razrađeni. Ovaj tip testa se još uvijek istražuje.

Na kraju, potrebno je spomenuti da su CAPTCHA programi i umjetna inteligencija međusobno povezani. Svaki CAPTCHA program koji je postao loš jer ga je računalni program zaobišao pridonio je razvoju umjetne inteligencije jer se računalni sustav za raspoznavanje malo približio ljudskom sustavu raspoznavanja. Jednog dana kada računalna sposobnost za raspoznavanje bude jednaka ljudskoj, CAPTCHA testovi će izgubiti svoju svrhu, a to je razlikovanje čovjeka od računala. Do tada, istraživat će se novi oblici CAPTCHA programa koji će s više ili manje uspjeha filtrirati *spam* napade.

6. Zaključak

Iz svih podataka iznesenih u ovom dokumentu može se zaključiti da CAPTCHA programi ne pružaju potpunu zaštitu od *spam* napada, ali mogu uspješno filtrirati veliki broj napada. Razlog je što napadači najčešće traže sustave sa slabijom zaštitom od napada. Ako je zaštita složena i potrebno je uložiti puno vremena u njeno zaobilaženje, većina napadača će odustati i preći na ranjivije sustave. CAPTCHA programima se ne filtriraju iskusni napadači koji su čvrsto odlučili napasti stranicu. Oni će naći način da i najsigurniji CAPTCHA program zaobiđu. Zaobilaženje se može izvesti na nekoliko načina kao što je opisano u dokumentu. Na kraju, napadaču uvijek ostaje mogućnost plaćanja za usluge rješavanja CAPTCHA zadataka. Na sreću, ovakvi napadači su rijetki jer im se jednostavno ne isplati plaćati kako bi, na primjer, postavili oglase u komentare na nečijem blogu.

Zbog svega toga, CAPTCHA program se isplati postaviti na stranicu, pogotovo zato što su implementacije najčešće besplatne i jednostavne. Dodavanje par linija koda može spriječiti veliki broj *spam* napada. Međutim, potrebno je redovito provjeravanje sigurnosti CAPTCHA programa i njegove zamjene ukoliko se utvrdi da postoji potreba za tim.

7. Reference

[1] Wikipedia: CAPTCHA, http://en.wikipedia.org/wiki/CAPTCHA

[2] ReCATPCHA projekt: http://recaptcha.net/

[3] Jonathan Strickland: How CAPTCHA Works, http://www.howstuffworks.com/captcha.htm/printable

[4] The History of CAPTCHA and the Future of CAPTCHA, http://baditaflorin.com/2009/06/the-history-of-captcha-and-the-future-of-captcha/, lipanj 2009.

[5] How to break captchas, http://www.blackhat-seo.com/2008/how-to-break-captchas/, ožujak 2008.

[6] Breaking CAPTCHA without OCR, http://www.puremango.co.uk/2005/11/breaking_captcha_115/, 2009.

[7] Using AI to beat CAPTCHA and post comment spam, http://www.mperfect.net/aiCaptcha/, siječanj 2005.