

Univerzita Pardubice

Fakulta ekonomicko-správní

Rozlišení člověka a robota ve webových aplikacích

Kateřina Havlíčková

Bakalářská práce

2009

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky
Akademický rok: 2008/2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Kateřina HAVLÍČKOVÁ**
Studijní program: **B6209 Systémové inženýrství a informatika**
Studijní obor: **Informatika ve veřejné správě**

Název tématu: **Rozlišení člověka a robota ve webových aplikacích**

Z á s a d y p r o v y p r a c o v á n í :

Problematika webových aplikací, ochrana proti nežádoucím činnostem.
Možnosti ochrany proti robotům.
Zvolení vhodného řešení obrany proti robotům, jeho otestování a zhodnocení.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

BELDA, Květoslav. Graphical Environment for Testing of the Control of Planar Parallel Robot. 1. vydání. Praha : ÚTIA AV ČR, 2000. Research Report 1996.

NARAMORE, Elizabeth, et al. PHP5, MySQL, Apache : vytváříme webové aplikace. 1. vydání. Brno : Computer Press, 2006. ISBN 80-251-1073-7.

KOSEK, Jiří. PHP : tvorba interaktivních internetových aplikací. 1. vydání. Praha : Grada Publishing, 1999. ISBN 80-7169-373-1.

SIROVICH Jamie - DARIE Christian. SEO v PHP : programujeme profesionálně. 1. vydání. Brno : Computer Press, 2008. ISBN 978-80-251-2083-5.

Vedoucí bakalářské práce:



Ing. Martin Novák

Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce:

6. října 2008

Termín odevzdání bakalářské práce:

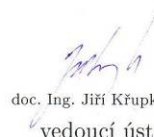
1. května 2009



doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.



doc. Ing. Jiří Křupka, Ph.D.

vedoucí ústavu

V Pardubicích dne 6. října 2008

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením Ing. Martina Nováka. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byla jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 1. 5. 2009

.....
Kateřina Havlíčková

Poděkování

Děkuji vedoucímu mé bakalářské práce panu Ing. Martinu Novákovi za cenné informace a odbornou pomoc při vypracování této práce.

Anotace

Práce je zaměřená na analýzu problematiky identifikace robota a člověka při běžných operacích ve webových aplikacích, jako je například vkládání nového příspěvku do diskusního fóra. Hlavní myšlenkou a tedy i praktickou částí je vytvoření testu pro identifikaci robota a člověka založeného na Turingových testech a následná analýza získaných údajů.

Klíčová slova

Captcha, Turingův test, Asirra, PHP, robot

Title

Distinction between man and robot in web applications

Annotation

This work is focus on distinction between man and robot in web application at current operation in web application, like is for example insertion new report in discussion forum. Main idea and so practical part are create a test for identification man and robot on Turing test established and subsequent data analysis.

Keywords

Captcha, Turing test, Asirra, PHP, robot

Obsah

1	Úvod	10
2	Robot	11
	2.1 Jak pracuje (ro)bot	11
	2.1.1 Roboti vyhledávačů	11
	2.1.2 Spamboti.....	11
	2.1.3 Ostatní roboti	12
3	Člověk nebo stroj	13
	3.1 Turingův test	13
	3.2 Turingův test dnes.....	14
4	Roboti a diskusní fóra	15
	4.1 Komentářový spam	15
	4.2 Současné metody rozpoznání člověka a robota	16
	4.2.1 Triviální matematický příklad / logická otázka.....	16
	4.2.2 Kontrola času	17
	4.2.3 Kontrola javascriptem.....	17
	4.2.4 Skryté pole.....	17
	4.2.5 Hodnocení příspěvků	17
	4.2.6 Bayesovský filtr	18
	4.2.7 Povinná registrace uživatelů	18
	4.2.8 Služba Mollom	18
5	Metoda Captcha	20
	5.1 ReCaptcha	21
	5.2 Imagination Captcha	22
	5.3 3D Captcha	25
6	Metoda Asirra	28
	6.1 ESP-PIX.....	29
	6.2 SQ-PIX.....	30
	6.3 KittenAuth	31
	6.4 Útoky na Asirru	32
	6.4.1 Útoky skrz odkaz na Petfinder.....	32
	6.4.2 Útoky hrubou silou	32
	6.4.3 Útoky prostřednictvím třídění obrázků	32
	6.4.4 Útoky prostřednictvím manuální rekonstrukce databáze	33
	6.4.5 Útoky na obrázky databáze	33

7	Návrh vlastního řešení.....	34
8	Implementace	36
	8.1 První řešení.....	36
	8.2 Druhé řešení	39
9	Výsledky testování.....	42
10	Závěr	47
Přílohy		I
	Příloha č.1: Zdrojový kód vzkazníku č.1.....	I
	Příloha č.2: Zdrojový kód vzkazníku č.2.....	VI

Seznam obrázků

Obrázek 1 - Turingův test	13
Obrázek 2 - Turingův test II.....	14
Obrázek 3 - Služba Molom.....	19
Obrázek 4 - Captcha – ukázky řešení	20
Obrázek 5 - Zdokonalená Captcha	21
Obrázek 6 - Příklad chyby při transformaci obrazu na text technologií OCR	21
Obrázek 7 - Ukázka 1. kroku Imagination Captcha	23
Obrázek 8 - Ukázka 2. kroku Imagination Captcha	23
Obrázek 9 - Statistika prolomení Captchy	25
Obrázek 10 – Ukázka testu s 3D Captchou	26
Obrázek 11 – Praktický příklad využití 3D Captchy	27
Obrázek 12 - Ukázka testu s metodou Asirra I.	28
Obrázek 13 - Ukázka testu s metodou Asirra II.	28
Obrázek 14 - Ukázka testu s metodou ESP-PIX	29
Obrázek 15 - Ukázka testu s metodou SQ-PIX	30
Obrázek 16 - Ukázka řešení testu s metodou SQ-PIX	30
Obrázek 17 - Ukázka testu s metodou KittenAuth.....	31
Obrázek 18 – Databázové tabulky	35
Obrázek 19 - Ukázka vlastního testu na bázi 1. řešení	39
Obrázek 20 - Ukázka vlastního testu na bázi 2. řešení	41

Seznam tabulek

Tabulka 1 - Statistika přístupů podezřelého uživatele I.	42
Tabulka 2 - Statistika přístupů podezřelého uživatele II.	43
Tabulka 3 - Statistika přístupů podezřelého uživatele III.	44
Tabulka 4 - Statistika přístupů podezřelého uživatele IV.	45

Seznam příloh

Příloha č. 1: Zdrojový kód vzkazníku č. 1

Příloha č. 2: Zdrojový kód vzkazníku č. 2

1 Úvod

Tato práce je zaměřena na problematiku rozlišení člověka a robota na webových aplikacích. Toho se dá využít například pro zabránění vkládání spamu do diskusních fór a chatů. Neustále se v různých diskusních fórech častěji a častěji setkáváme s velkým množstvím příspěvků obsahující spam. Se spammem neboli nevyžádanou poštou se v současné době setkal téměř každý a to ať už v diskusních fórech, tak i v e-mailech či u instant messagingu ICQ. Spam rozesílají různé společnosti pro svou reklamu například prostřednictvím sítě botnet. Zde se však konkrétně zabýváme komentářovým spammem, který se stal obrovským problémem všech správců webových stránek provozujících diskusní fóra, návštěvní knihy či chaty. Spamboti se zdají být stále neúnavní a zahlcují především mnohá diskusní fóra svými reklamními příspěvky počínaje finančními produkty a afrodiziaky konče. Komentářový spam se však bohužel vyvíjí a stává se tak mnohem záladnějším. Na diskusních fórech či blozích se v poslední době můžeme setkat s příspěvkem typu, kdy nám neznámý pisatel chválí naše webové stránky a informace v nich obsažené. Bohužel těchto příspěvků na fóru rapidně přebývá a my se už jen můžeme dohadovat, zda opravdu onen pisatel byl člověk či robot rozesílající spam. Takovýto spam se v nás snaží vyvolat dojem hodnověrnosti a spolehlivosti a staví se tak na úroveň rovnocenného nepřítele. Proto se neustále mnozí správci diskusních fór snaží identifikovat, zda uživatel vstupující do diskuse je opravdu člověk či robot. V případě, že se jedná o robota, pak vytvořit taková opatření, které mu znemožní příspěvek do diskuse vůbec vložit.

S myšlenkou jak rozeznat člověka od robota jako první přišel britský matematik Alan Turing. Na základě jeho poznatků vznikla metoda Captcha, která uživatele rozlišuje prostřednictvím opisování kódu z vygenerovaného obrázku. Bohužel roboti si už našli prostředky, jak Captchu prolomit a tak se stále hledá nová metoda, která zabráni robotům v masivním rozesílání spamu. Vyhlídkou na lepší časy se zdá být metoda Asirra, která uživatele rozlišuje prostřednictvím rozeznávání obrázků zvířat. Asirra je však novinkou, a přestože se jeví jako velmi účinná, nelze s jistotou říci, že taková opravdu bude. Pro ověření její účinnosti jsem navrhla dvě řešení rozlišení člověka a robota vkládajícího příspěvek do diskusního fóra. Tyto vlastní navržené testy jsou založeny na její myšlence, kdy před odesláním příspěvku do diskuse je nutné z 8 vygenerovaných obrázků zvolit ten, na který je uživatel dotazován. Pokud tak učiní, je považován za člověka, pokud však zvolí jiný obrázek než na ten, na který je dotazován, je považován za robota či za nepozorného uživatele. Zda se opravdu jedná o robota či o nepozorného uživatele, který se pouze spletl, nám až ukáže pohled do statistiky návštěv či další opatření, která jsem do testu také zahrнула.

2 Robot

Robot (pavouk, trawler či spider) je speciální počítačový program, který přes jednotlivé hypertextové odkazy prochází webové stránky za účelem indexace těchto stránek (roboti vyhledávačů), sběr e-mailových adres (spamboti), vkládání komentářového spamu (spamboti), kontroly funkčnosti odkazů (roboti sloužící k ladění webů), konverzace či získání osobních údajů (chatterboti). [1, 2]

2.1 Jak pracuje (ro)bot

Tato kapitola uvádí různé druhy robotů, se kterými se lze na internetu nejčastěji setkat. Je zde také vysvětleno k čemu se jednotliví roboti využívají a jaké provádějí činnosti při plnění svých úkolů.

2.1.1 Roboti vyhledávačů

Na svém začátku vyhledávací robot (někdy nazývaný též bot) získá k dispozici seznam URL adres webových stránek, které má za úkol projít. Během tohoto procházení si zapisuje odkazy na jednotlivých stránkách do svého seznamu URL adres a vytvoří si kopii této stránky. Díky těmto kopiím si robot tvoří index všech slov, která se na dané stránce nachází. Tento index slov pak následně prohledává, když uživatel zadá dotaz do vyhledávače. Pokud robot ve svém indexu slov najde stránky odpovídající onomu dotazu, zobrazí je uživateli v seznamu sestupně podle jejich relevantnosti. [3]

2.1.2 Spamboti

Spamboti se stali bezmála celosvětovým problémem. Na internetu se lze setkat především se spamboty využívající e-mailové adresy pro rozesílání spamu a se spamboty vkládající komentářový spam. [4]

Spamboti využívající e-mailové adresy pro rozesílání spamu prohledávají internet za účelem získání e-mailových adres. Na webových stránkách se tak snaží najít symbol „zavináče“ (@), podle něhož zjistí, že se jedná o e-mailovou adresu. Bohužel nyní roboti

nehledají pouze symbol @, ale také samotné slovo „zavináč“ či „at“, „from“ a jiné, kterými se dříve dalo ochránit e-mailové adresy proti tomu, aby je spambot našel.

Všechny takto získané e-mailové adresy si robot ukládá do databáze. Tato databáze se pak následně stává velice cennou pro jisté subjekty, které tento seznam adres odkoupí od autora tohoto robota a využijí jej k rozesílání spamu, většinou za účelem propagace svých výrobků. [4]

Druhá verze spambotu vyhledává na internetu především diskusní fóra, návštěvní knihy a chaty, do nichž vkládá komentářový spam. Nejzranitelnější jsou starší verze návštěvních knih a diskusní fóra postavená na technologii CMS Joomla a phpBB. [4]

2.1.3 Ostatní roboti

Na webu se dále lze setkat s roboty pro správu a údržbu webů, kteří například kontrolují dostupnost souborů. Na chatu zase může jít o setkání s IRC botem (Internet Relay Chat bot), který poskytuje uživateli rozhraní pro komunikaci s ostatními uživateli na chatu, může vyhodit uživatele při použití nevhodných výrazů a jiné. Ve službě ICQ je možné obdržet požadavek na přidání do seznamu od ICQ bota a text takového požadavku pak může obsahovat spam nebo odkaz na zavirovanou internetovou stránku. Někteří ICQ boti jsou už tak vyspělí, že dokážou s člověkem komunikovat, flirtovat, vymáhat od něj jistou finanční částku či získat osobní údaje nebo čísla bankovních účtů. [4, 5]

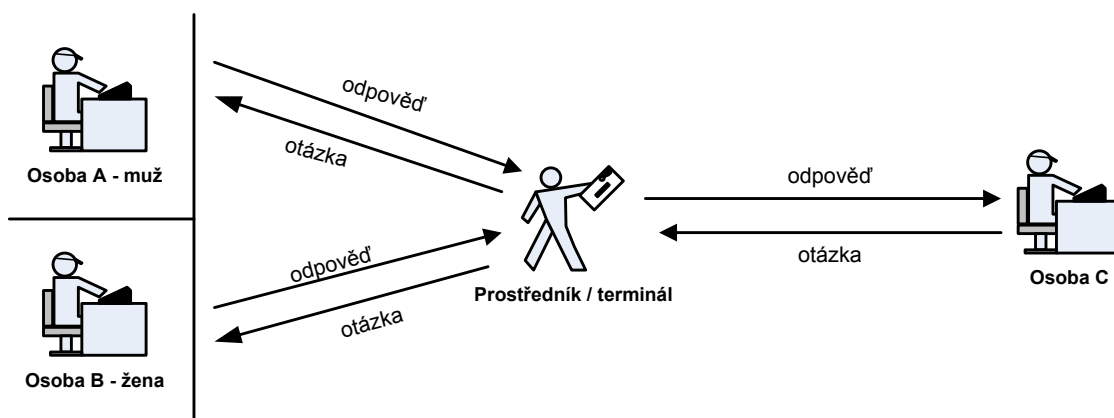
3 Člověk nebo stroj

Jako první se otázkou identifikace člověka a robota (stroje) zabýval Alan Mathison Turing. Alan Turing byl významný logik, matematik, který svůj život zasvětil umělé inteligenci a významně přispěl k rozvoji výpočetní techniky. Na jeho počest se od roku 1966 uděluje cena A. M. Turinga. [6]

„Turing se konstrukcí svého testu snažil otázku "mohou stroje myslet" převést z oblasti filozofických spekulací na srozumitelnější úroveň. Jinak řečeno, za myslící podle něj prohlásíme počítač tehdy, když jeho chování nebudeme schopni rozeznat od chování člověka.“ [7]

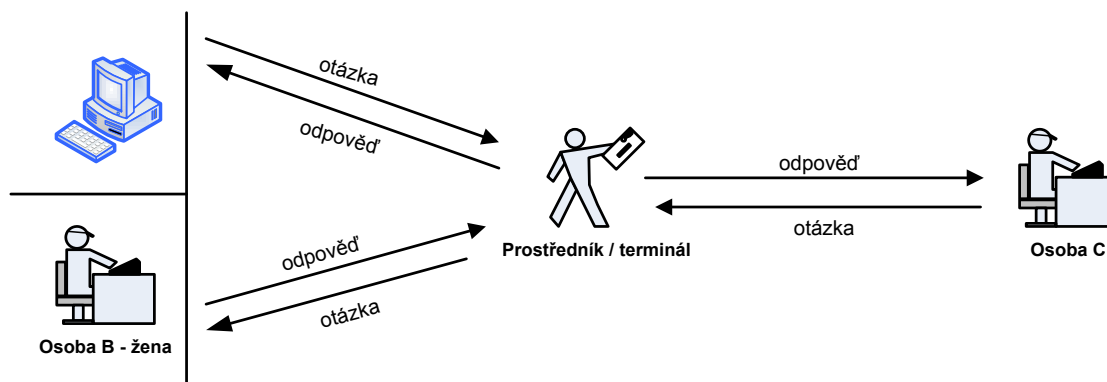
3.1 Turingův test

Turingův test (viz obrázek 1) spočívá v tom, že do jedné místnosti je umístěna osoba A (muž) a osoba B (žena) jako skupina testovaných a do druhé místnosti je umístěna osoba C jako testující, u níž není pohlaví relevantní. Osoba C neví nic o osobě A ani o osobě B a má za úkol zjistit, kdo je osoba A, kdo je osoba B. Osoba C oběma klade otázky a následně získává odpovědi od osob A i B v tištěné podobě například prostřednictvím terminálu či přes nějakého prostředníka. Odpovědi jsou záměrně zasílány v tištěné podobě, aby ze zabarvení hlasu nebylo možné rozeznat, o koho se jedná. Osoba B (žena) se snaží odpovídat co nejpravdivěji a osoba A (muž) se snaží osobu C zmást a hodnotu pravdivosti odpovědi zkresluje.



Obrázek 1 - Turingův test, zdroj: autor

Lze si však položit otázku: Jestliže osobu A nahradíme počítačem, bude osoba C (testující) ve svých zjištěních mýlit či nikoliv?



Obrázek 2 - Turingův test II., zdroj: autor

V Turingově testu tedy počítač předstíral, že je člověk (viz obrázek 2). A pokud osoba C (testující) nedokázal z odpovědí testovaných zjistit, zda se jedná o člověka či o počítač, pak počítač lze považovat za „stroj, který umí myslet“. [7, 8]

3.2 Turingův test dnes

V současnosti se již Turingův test bohužel běžně nevyužívá. Je totiž považován za zastaralý a nepoužitelný, protože nebere v úvahu procesy probíhající uvnitř v systému ani vlastnosti mysli. Dále je mu vytýkáno, že není dostatečně objektivní, jelikož je kladen důraz na osobu testujícího a na jeho fantazii při kladení otázek.

I přes to všechno se dnes využívají modifikace Turingova testu a několik programů z Turingova testu dokonce vychází. Takovým příkladem může být program Eliza či Parry. Eliza je inteligentní program, který se snaží simulovat chování psychoterapeuta, zjišťuje aktuální pocity pacienta (člověka), snaží se mu porozumět a následně pomoci. Parry je zase opakem Elizy, který napodobuje chování pacienta trpícího paranoiou a schizofrenií. Člověk (psychoterapeut) mu klade několik otázek a snaží se zjistit, zda je Parry skutečným člověkem či programem.

Turingův test jako celek se však dodnes dosud žádnému programu nepodařilo překonat, i když se o to lidé konstrukcí stále lepších a lepších programů neustále pokouší. [9]

4 Roboti a diskusní fóra

Na Turingově testu je založena především metoda Captcha a další modifikované metody z ní vycházející, kterými se budeme zabývat v následujících kapitolách. Hojně je tato metoda využívána v identifikaci robota a člověka v různých diskusních fórech, na které se zde zaměřuji. Proto následující zmiňované metody a způsoby ochrany jsou zacíleny na tuto problematiku, na problematiku komentářového spamu.

4.1 Komentářový spam

Pod pojmem komentářový spam si lze představit vkládání nevyžádaných, reklamních, obtěžujících a mnohdy i nesmyslných příspěvků do diskusních fór, návštěvních knih a chatů. Objem tohoto spamu stále narůstá a je tak nutné proti němu najít účinnou ochranu. Pro nalezení optimální ochrany proti komentářovému spamu je nejprve nezbytné zjistit, co všechno roboti, kteří tento spam rozesílají, dovedou a nedovedou. Na základě těchto poznatků pak vytvořit test, který bude po určitou dobu umístěn na webu a testován. Z databáze se poté provede analýza přístupů uživatelů, frekvence jejich přístupů, identifikace robotů, zjištění počtu odepřených přístupů a jiné, což je náplní praktické části této práce.

Co tedy všechno víme o robotech?

- neznají význam políček, která vyplňují
- neumí CSS (kaskádové styly)
- neinterpretují javascript
- nevidí objekty typu Flash
- neslyší zvuk, nerozpoznají barvy ani obsah obrázků. [10-12]

Bohužel nelze se 100% jistotou tvrdit, že vše, co nyní víme o robotech, stále platí či bude platit do budoucna. Roboti se neustále vyvíjí a jsou čím dál více sofistikovanější a tak už je jen otázkou času, kdy se tyto výše zmíněné nedostatky stanou jejich přednostmi.

4.2 Současné metody rozpoznání člověka a robota

Metod, jak identifikovat, zda se jedná o člověka či robota je několik. Zároveň se tyto metody využívají jako ochrany proti komentářovému spamu [10-18]:

- Triviální matematický příklad / logická otázka
- Kontrola času
- Kontrola javascriptem
- Skryté pole
- Hodnocení příspěvků
- Bayesovský filtr
- Povinná registrace uživatelů
- Služba Molom
- Metoda Captcha
- Metoda Asirra
- a jiné.

4.2.1 Triviální matematický příklad / logická otázka

Prozatím velice osvědčenou metodou je kladení triviálních matematických příkladů typu „Kolik je 1+2 ?“ či logických otázek typu „Jakou barvu má červené auto?“. Zdá se, že pro robota je prakticky nemožné na otázky tohoto typu odpovědět. Je však nutné otázky/čísla měnit při každém načtení a aktualizaci stránky, protože v případě stále stejné otázky/čísel, může robot neustálým střídáním různých kombinací nakonec odpověď/výsledek nalézt.

U identifikace výpočtem matematického příkladu je nutné, aby při každém načtení či aktualizaci stránky se náhodně generovaly čísla tvořící daný příklad. S těmito čísly se poté provede požadovaná matematická operace a výsledek výpočtu se porovná s číslem, které napsal uživatel. Jedinou nevýhodou této metody je fakt, že to obtěžuje uživatele. [10]

4.2.2 Kontrola času

Jednoduchá kontrola času nám velice snadno objasní, jestli dotyčný byl člověk nebo robot. Tato metoda využívá toho, že robot je mnohem rychlejší než člověk a zvládne tak během jedné sekundy několik operací. Dokáže například vložit příspěvek do diskuse během 2 sekund, oproti člověku, kterému to podstatně trvá déle.

Metoda kontroly času spočívá v tom, že si stačí uložit čas zobrazení stránky a čas odeslání příspěvku do diskuse a následně tyto časy od sebe odečíst. Robota tak jednoduše poznáme podle toho, že svůj příspěvek odešle příliš rychle. Bohužel už ale existují takoví roboti, kteří vyčkávají určitý čas než svůj příspěvek odešlou. [10]

4.2.3 Kontrola javascriptem

Tato metoda využívala toho, že roboti neumí interpretovat javascript. V dnešní době však její význam klesá, protože velké procento robotů javascript již ovládá.

Postatou této metody je, že do formuláře se vloží pole s výzvou k opsání určitého textu, které se zobrazí, pouze pokud není povolen javascript, tzn. toto pole uvidí i robot. Ten však není schopen pole vyplnit správně podle pokynu. Je-li ovšem javascript povolen, do formuláře se vloží skryté pole, které odešle příslušný text, aniž by vyžadovalo aktivní účast uživatele. [10]

4.2.4 Skryté pole

Přidáním skrytého pole a následným testováním zda bylo vyplněno či nikoliv lze také velice snadno zjistit, zda se jedná o člověka či robota. Robot z principu vyplňuje všechny pole, aniž by znal význam polí, která vyplňuje, proto i toto skryté pole vyplní. Naopak člověk, jelikož pole nevidí, jej nevyplní. Proto lze konstatovat, že pokud je pole vyplněno, jde o robota a pokud není, jde o člověka. [10]

4.2.5 Hodnocení příspěvků

Metoda hodnocení příspěvků je velmi jednoduchá - každý příspěvek ve fóru může být kýmkoliv ohodnocen. Díky tomuto hodnocení je následně příspěvek zvýrazněn a stojí tedy za přečtení či potlačen, obsahuje-li nepřínosné informace (spam). Správce fóra si nakonec nechá zobrazit všechny příspěvky s nízkým hodnocením a ty poté vymaže. [13]

Takto jednoduše může správce identifikovat příspěvky od robota či od uživatelů, za jejichž pomoci se zbaví komentářového spamu ve svém fóru. Nevýhodou této metody je skutečnost, že nízké hodnocení může získat i příspěvek, který je pro část uživatelů nepřínosný, ale pro druhou část uživatelů má zistný význam. [13]

4.2.6 Bayesovský filtr

Bayesovský filtr se hlavně používá pro boj se spamem v e-mailech, dnes je možné se s ním setkat už i v některých diskusních fórech. Funguje na jednoduchém principu, kdy si u každého slova, které má v databázi, pamatuje pravděpodobnost toho, že onen příspěvek, ve kterém je toto slovo obsaženo, je spam či nikoliv. Podle toho, co filtr vyhodnotí jako spam a co ne se pak rozhoduje, zda je příspěvek přidán do diskuse či nikoliv. [14, 15]

4.2.7 Povinná registrace uživatelů

Nemožnost přidání příspěvku do fóra bez předchozí registrace, součástí které je samozřejmě dokončení registrace přes e-mail, se zdá jako účinná metoda. Avšak někteří roboti jsou už tak sofistikovaní, že si uživatelský účet dokážou aktivovat a následně diskusní fórum doslova tapetovat spamem, takovým příkladem mohou být fóra postavená na technologii phpBB. [16]

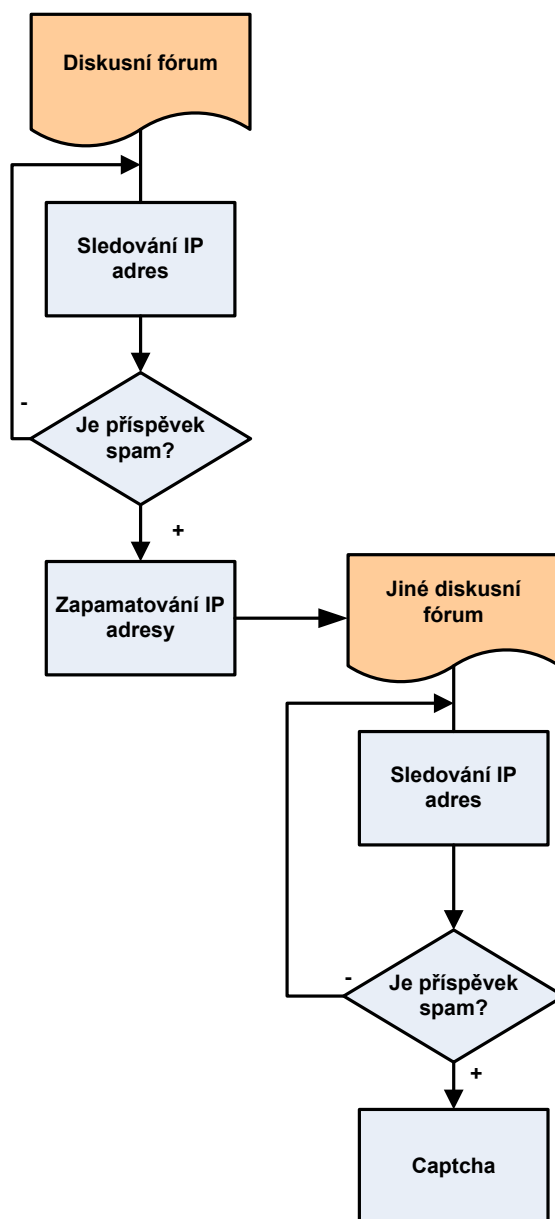
4.2.8 Služba Mollom

Mollom je webová služba pomáhající rozeznat spam a účinně ho odstranit. Služba je k dispozici zdarma, pokud denní počet spamů vložených do fóra nepřesáhne hodnotu 100, v opačném případě je nutné připlatit určitou finanční částku.

Služba Mollom sleduje IP adresy všech uživatelů, kteří přes něj projdou. Když je jejich příspěvek ve fóru označen za spam, Mollom si jejich IP adresy zapamatuje. Pokud tito uživatelé budou chtít svůj příspěvek vložit na jiném fóru, na kterém služba Mollom také funguje, budou muset nejdříve překonat Captchu, kterou sem Mollom zaslal. Díky tomu jsou roboti identifikováni a znevýhodněni u jiných diskusních fór provozujících stejnou službu. Mollom také nabízí podrobnou statistiku v podobě grafů zachycujících informace o zadrženém spamu.

Podle současných statistik Mollom chrání 6813 webových stránek. A jeho průměrná účinnost je 99,96%. [17, 18]

Schéma činností k rozeznání spamu u služby Molom je k vidění na obrázku níže.



Obrázek 3 - Služba Molom, zdroj: autor

5 Metoda Captcha

Metoda Captcha (*anglicky Completely Automated Public Turing test to tell Computers and Humans Apart* = zcela automatický veřejný Turingův test na rozlišení člověka a počítače) je dnes nejrozšířenější metodou identifikace robota a člověka a zároveň účinným způsobem v boji proti spamu na mnoha diskusních fórech.

Tato identifikace je založená na opsání kódu z vygenerovaného obrázku, přičemž tento kód se skládá z alfanumerických znaků a zároveň je různě deformován. Na obrázku 4 jsou některé příklady Captchy. [19, 20]



Obrázek 4 - Captcha – ukázky řešení, zdroj: [21]

Nejvíce se metoda Captcha používá u potvrzování formulářů / příspěvků / komentářů. Předtím než se například samotný příspěvek do diskuse vloží je potřeba pro jeho potvrzení opsat kód z vygenerovaného obrázku a je-li tento kód opsán správně, příspěvek je vložen do diskuse, v opačném případě je vygenerovaný jiný kód pro opsání. [19]

Bohužel v současné době začínají být roboti vybaveni stále lepšími a lepšími technikami pro rozpoznání textu a díky tomu se Captcha snaží stále více zdokonalovat, avšak na úkor své čitelnosti. Stává se tak mnohem složitější nejen pro roboty, ale dokonce už i pro běžné

uživatelé, nemluvě o zrakově postižených či slabozrakých uživateli, kteří jsou díky tomu ještě více znevýhodněni. Na obrázku 5 jsou ukázky zdokonalené (nečitelné) Captchy. [20]

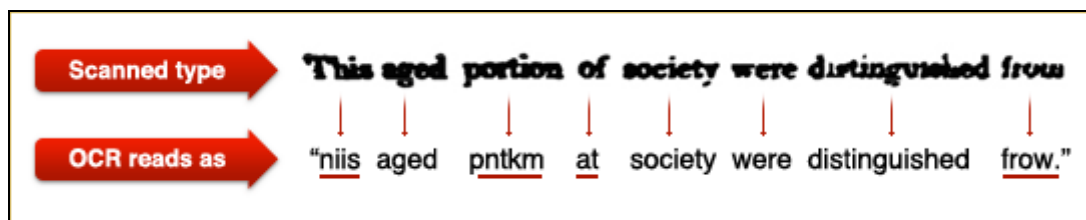


Obrázek 5 - Zdokonalená Captcha, zdroj: [22, 23]

5.1 ReCaptcha

„ReCaptcha je bezplatná služba, která pomáhá při digitalizaci knih, novin a rozhlasových pořadů z dřívějších let.“ [24]

V procesu digitalizace jsou jednotlivé stránky knihy oskenovány a následně transformovány do textu při použití OCR programů, které opticky rozeznávají jednotlivé znaky textu. Bohužel ne všechny znaky jsou správně rozeznány (viz obrázek 6) a tak slovo, které program OCR nedokáže správně rozeznat je uloženo jako obrázek, aby mohlo být využito metodou Captcha jako ochrana webových formulářů, komentářů, e-mailů a příspěvků diskusních fór. [24]



Obrázek 6 - Příklad chyby při transformaci obrazu na text technologií OCR, zdroj: [24]

ReCaptcha tedy výrazně usnadňuje digitalizaci knih a navíc slouží jako zdroj účinných prostředků pro metodu Captcha, jelikož programy oné slovo nedokážou rozpoznat, avšak uživatel ano. Díky tomu jsou uživatelé ušetřeni neustálého luštění textu z obrázků, texty jsou čitelnější a zároveň lze takto automaticky opravovat nedostatky vzniklé při převodu obrazu na text v procesu digitalizace, protože tyto nedostatky opravují sami uživatelé. Zároveň má ReCaptcha i audio test, který umožňuje procházet webové stránky také nevidomým lidem.

Aby bylo zamezeno tomu, že text z obrázku by byl zadán nesprávně (například by mohl pole náhodně vyplnit robot, i přesto, že není schopen text rozeznat – roboti využívají také OCR technologie), uživatelům jsou dány k vyřešení dvě slova v jenom obrázku mající stejnou úroveň obtížnosti. První slovo systém zná, druhé však nikoliv. Pokud tedy uživatel správně rozpozná první slovo, je automaticky i druhé slovo považované za správně rozpoznané. Do digitalizace je zařazeno až poté, kdy je odpověď na toto druhé slovo porovnáno s odpověďmi ostatních uživatelů a zmíněná odpověď se zde vyskytuje v převažujícím počtu. Princip je tedy takový: co nedokáže rozeznat technologie OCR, rozezná člověk a jedno slovo musí být ověřeno více lidmi. [24]

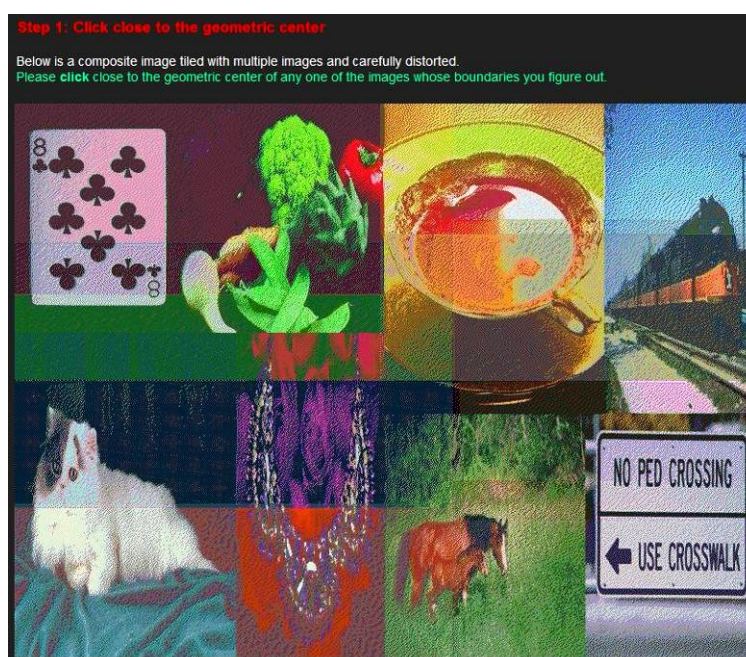
5.2 Imagination Captcha

Test založený na imagination (obrazotvorné) Captcha spočívá ve dvou krocích nutných k rozlišení člověka a robota. V prvním kroku (viz obrázek 7) se uživateli zobrazí kompozitní obraz tvořený několika různými obrázky a uživatel je vyzván, aby jeden obrázek vybral a kurzorem myši kliknul na místo, kde si myslí, že se nachází střed onoho obrázku. Obrázek si však nemůže být vybrán náhodně, musí být vybrán právě ten obrázek, který má přibližně geometrický tvar, např. hlava – koule, pyramida – trojúhelník atd. A následně u tohoto obrázku musí uživatel najít jeho střed. Pro člověka - jednoduchý úkol, avšak pro robota – velký problém. V druhém kroku (viz obrázek 8) pak uživatele čeká popsání obrázku, kdy z nabídky musí vybrat, co obrázek znázorňuje. Pakliže uživatel projde oběma kroky, test splnil a uživatelem je tedy člověk, v opačném případě robot. Uvažujeme-li, že jistou shodou okolností se robotu podaří projít prvním krokem, je zde pravděpodobnost 1:15, že zvolí správné označení obrázku a projde testem. Pokud však zvolí nesprávnou variantu, musí opakovat znovu krok č. 1. Po několika nesprávných pokusech je uživateli test zablokován z důvodu ochrany proti případnému organizovanému útoku robotů. [25]

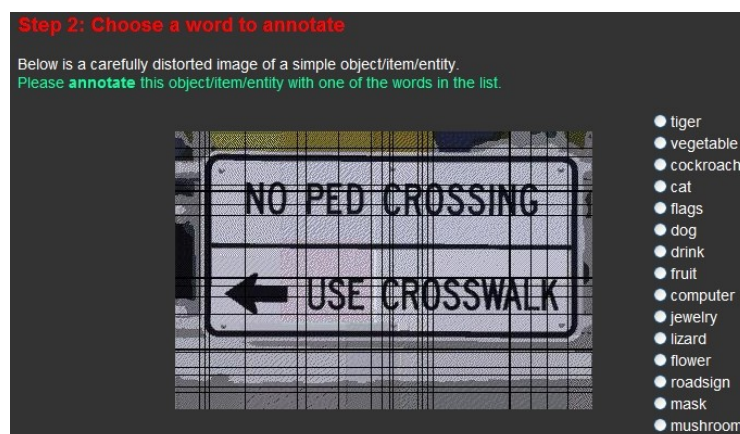
Tento test, přestože to tak může vypadat, jednoduchý příliš není. Z vlastní zkušenosti mohu potvrdit, že v prvním kroku je někdy velice těžké najít onen obrázek mající geometrický tvar. Někdy se zde nachází dva nebo tři obrázky mající geometrický tvar a je těžké určit,

který je ten správný. Díky tomu může být po několika pokusech uživateli test zablokován, protože systém je přesvědčen, že onen uživatel je robot, přestože tomu tak není.

„Obrázkové CAPTCHA testy, pokud si najdou své fanoušky, mohou jistý čas odolávat náporu robotů, z dlouhodobého hlediska ale nemají šanci, už dnes se totiž pracuje na inteligentních OCR technikách, které si poradí nejen s rozpoznáváním textu v obrázku ale i s rozpoznáváním složitějších geometrických struktur. Za zmínku stojí chytré kompaktní fotoaparáty, které při ostření najdou ve scéně lidskou tvář nebo vyhledávače Google, Live Search a také český MUFIN. Zůstává tedy otázkou, kdy podobné technologie zdokonalí i autoři botnetů a OCR robotů.“ [25]



Obrázek 7 - Ukázka 1. kroku Imagination Captcha, zdroj: [26]



Obrázek 8 - Ukázka 2. kroku Imagination Captcha, zdroj: [26]

V roce 2008 informovala společnost Websense na svých webových stránkách o prolomení Captchy, která se využívá na ověření uživatelů při registracích. Roboti nyní dokážou projít ochranou, která slibovala odlišit robota a člověka a robotům tak zamezit v přístupu. Díky této skutečnosti již pro robota není nic těžkého založit si stovky, ba dokonce tisíce e-mailových adres a z nich pak rozesílat spam. [29]

„Útok současného bota vede k úspěšnému založení účtu sice jen v 10-15 % případů, to je však z hlediska útočníka vzhledem k rychlosti útoku nepodstatné, respektive zcela dostačující.“ [30]

Útočníci pro své útoky využívají především sítě botnet. A dokázali tak už prolomit Captchu ve službách Gmail, Yahoo a Live Hotmail (u e-mailu Live Hotmail na to stačí útočníkovi pouhých 6 sekund). [29, 31]

„To má lavinový efekt a podvodníci se zaměřují i na další servery, respektive i na inovované verze ochrany. Poslední případ umožňuje útočníkům využít po prolomení obrany Captcha výhod bezplatného hostingu uživatelů generovaných dokumentů. Adresy v doménách jako blogspot lze kromě přidávání komentářového spamu používat i pro vytváření vlastních dokumentů a odkazy z e-mailového spamu. Protože tyto domény jsou obecně považovány za důvěryhodné, antispamové technologie je tak intenzivně nefiltrují.“ [31]*




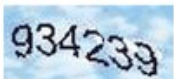


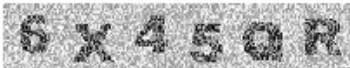

Dokonce se také na různých diskusních fórech objevují příspěvky, že byla prolomena už i nová verze technologie phpBB 3.0.4. [32, 33]

Společnost Websense zatím nezná způsob, jak se útočníkům podařilo prolomit Captchu, ale sestavila 5 možných scénářů, jak k této události mohlo dojít [29]:

- útočník získal k oprávnění k databázi obrázků, kde se nachází i jejich překlad na text,
- pro rozpoznání textu útočník využívá velmi pokročilý OCR skener,
- útočník využívá matematický software pro výpočet hashe, který je přiřazen k textu,
- útočník využívá hlasový skener, díky němuž rozluští hlasovou Captchu,
- útočník využívá několik lidí, kteří ručně přepisují obrázky Captcha.

* Blogspot známý také jako Blogger je velmi oblíbený free hosting.

Následující obrázek ukazuje nejčastěji používané metody Captcha u jednotlivých technologií, procento úspěšnosti prolomení Captchy a požadovaný čas nutný k tomuto úkonu. [34]

Editor	Captcha	Procento úspěšnosti	Potřebný čas
Google		20% 80%	6s 1min
Microsoft		15% 70%	6s 1min
Yahoo		30% 80%	6s 40s
Ebay		80%	4s
PayPal		100%	2s
Clubic		100%	1s
phpBB (forum)		97%	3s
IPB (forum)		97%	7s

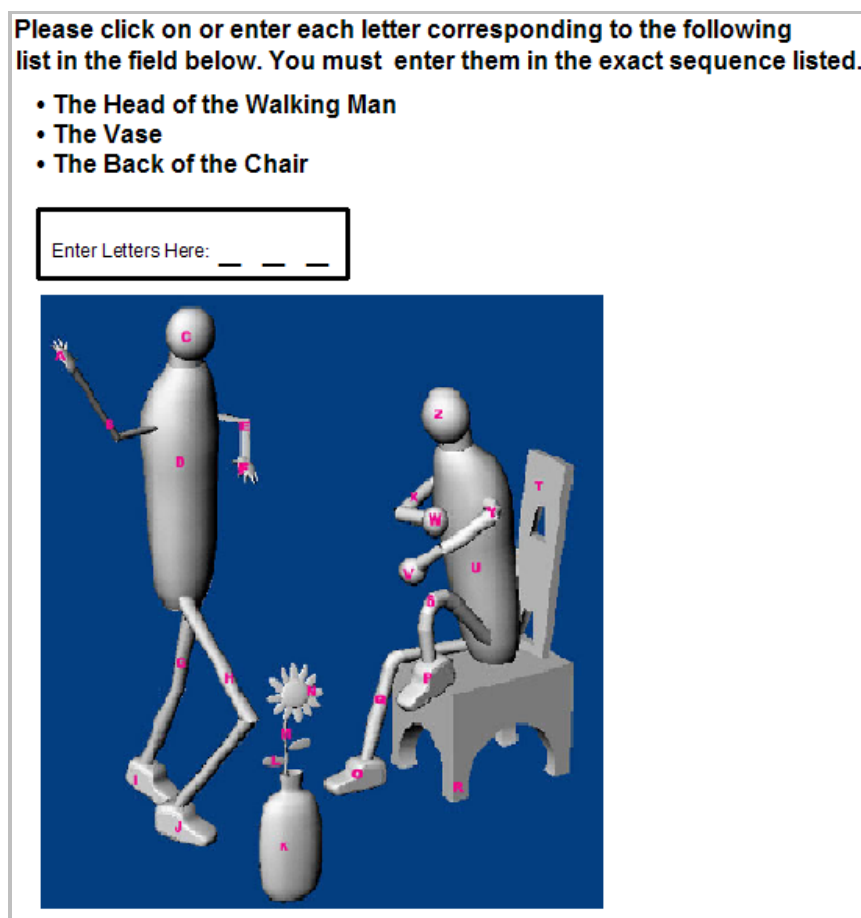
Obrázek 9 - Statistika prolomení Captchy, zdroj: autor – upraveno na základě zdroje [34]

5.3 3D Captcha

Náhradou 2D Captchy by se měla stát 3D Captcha využívající toho, že lidský mozek je schopen rozeznávat objekty v prostoru z různých úhlů pohledu. Toho se dá využít jako účinná ochrana proti robotům, kteří tuto schopnost nemají - jejich technologie OCR nejsou na tuto novinku stavěné. 3D Captcha tak může nahradit dnes velice zranitelnou 2D Captchu, se kterou všichni se běžně setkáváme.

Metoda 3D Captchy se skládá ze 4 kroků [27]:

- 1) v prvním kroku se vytvoří knihovna různých 3D objektů,
- 2) v druhém kroku jsou objekty popsány svým atributy,
- 3) ve třetím kroku počítač vybere z knihovny několik objektů, z nichž sestavuje různé scény,
- 4) v posledním kroku jsou pak jednotlivé atributy objektů popsány písmeny a zakódovány. Test s 3D Captchou pak vypadá následovně, viz obrázek 10.



Obrázek 10 – Ukázka testu s 3D Captchou, zdroj: [27]

Pro vyřešení testu je tedy nutné zapsat písmena odpovídající výrazům v odrážkách:

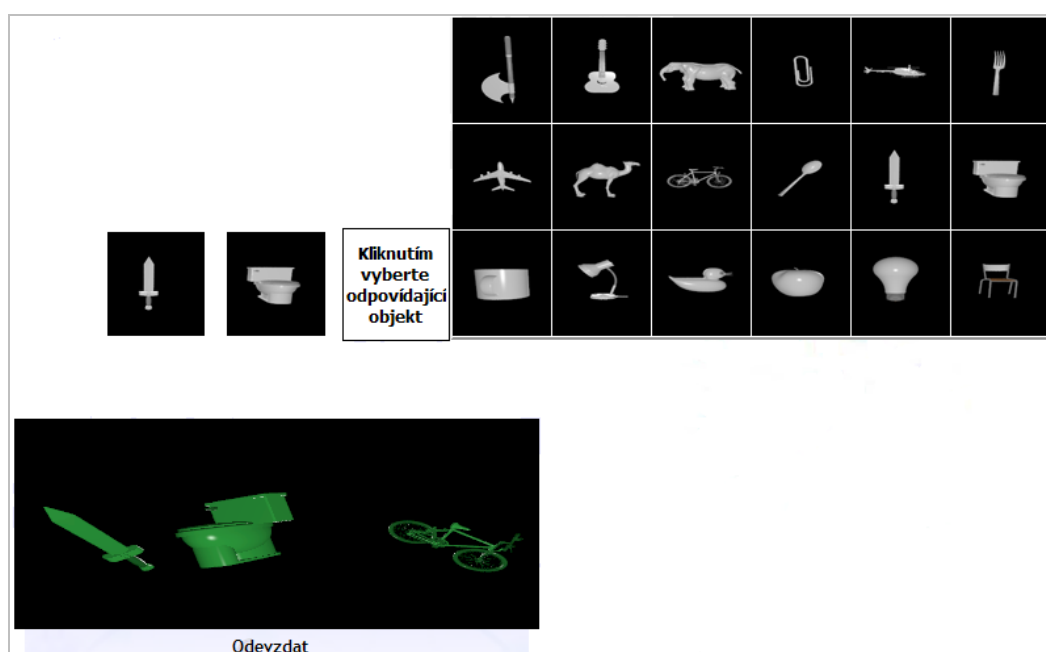
- hlava chodícího muže → C
- váza → K
- opěradlo → T

Správným řešením testu je kód CKT. [27]

Pokusí-li se robot hrubou silou o prolomení testu s 3D Captchou dotazující se na 3 atributy objektu, má pravděpodobnost na úspěch 1:15600. V případě zvýšení atributů na 5, se jeho pravděpodobnost úspěchu snižuje na 1:7893600, ale efektivnost testu se tak zvýší. Je-li např. od uživatele požadováno vyplnění 4 identifikujících atributů při toleranci jedné chyby, pak šance na úspěch je 1:13800.

Ve 30 % případů dokážou někteří roboti rozeznat jeden objekt ze tří, například květinu. Díky tomu se snaží identifikovat atributy i ostatních dvou objektů. Vygenerují tak několik možných kombinací, které zkusí jednu po druhé. Vždy tak správně identifikují ten samý objekt (květinu), ale ostatní nikoli. Tím dochází k příjmu mnoha nesprávných odpovědí, kdy je vždy identifikován jen jeden stále stejný objekt. V důsledku toho se automaticky na místo identifikovaného objektu (květiny) vygeneruje objekt nový. Tento proces nastává automaticky vždy při příjmu většího množství odpovědí, což zabraňuje robotům pokračovat v útoku. [27]

Toto je další z příkladů využití 3D Captchy, který na svých stránkách využívá společnost YUNiTi.com při odesílání formulářů. Nejprve se vygeneruje obrázek se třemi objekty v různých úhlech pohledu, viz obrázek 11. Uživatel je pak požádán, aby z nabídky 18 objektů vybral ten, který se shoduje s objektem na vygenerovaném obrázku. Pokud všechny tři objekty identifikuje správně, pak je považován za člověka.



Obrázek 11 – Praktický příklad využití 3D Captchy, zdroj: [28]

6 Metoda Asirra

Metoda Asirra (Animal Species Image Recognition for Restricting Access – rozpoznávání obrázků druhů zvířat pro omezení přístupu) od firmy Microsoft funguje na principu rozeznávání koček a psů. Fotografie potřebné k tomuto testu jsou získány ve spolupráci s Petfinder.com, kde je jich k dispozici více než 3 miliony. [35]



Obrázek 12 - Ukázka testu s metodou Asirra I., zdroj: [35]

Uživatel je tedy vyzván, aby z 12 obrázků označil ty, na kterém se nachází kočka. Pokud všechny obrázky označí správně, je považován za člověka. Pokud však označí obrázek psa či neoznačí všechny obrázky koček, je považován za robota, jak lze vidět na obrázku 12. Zároveň je možné se během testu přesměrovat na web service Asirry, který poskytuje požadované přesměrování na webové stránky útulků nabízejících tyto domácí miláčky, protože při najetí kurzorem myši na obrázek se u něho zobrazí odkaz „Adopt me“ a člověk tak může i přispět dobré věci a podpořit organizaci Petfinder, jak lze vidět na obrázku 13. [35]



Obrázek 13 - Ukázka testu s metodou Asirra II., zdroj: [35]

Modifikací metody Asirra vzniklo několik nových metod, které ale mají jednu zásadní nevýhodu - využívají poměrně malou databázi obrázků, čímž zvyšují procento svého prolomení roboty [35]:

- ESP-PIX
- SQ-PIX
- KittenAuth.

Na základě uživatelských studií bylo zjištěno, že 99,6% uživatelů zvládlo test s metodou Asirra vyřešit pod 30 sekund. Dále se očekává, že s pravděpodobností 1:54000 nebude mít robot šanci Asirru prolomit. Asirra na rozdíl od Captchy bohužel úplně znevýhodňuje nevidomé. U Captchy existuje možnost si poslechnout kód z obrázku, ale u Asirry nic podobného není. Nadále je však Asirra ve fázi testování. [35]

6.1 ESP-PIX

Metoda ESP-PIX je velice podobná metodě Asirra. Pro splnění testu s metodou ESP - PIX je zapotřebí vybrat z nabídky slovo, které vystihuje všechny čtyři obrázky. Například na obrázku 14 je spojujícím prvkem všech obrázků testu slovo most. [36]



Obrázek 14 - Ukázka testu s metodou ESP-PIX, zdroj: [37]

6.2 SQ-PIX

Stejně jako metoda ESP-PIX je metoda SQ-PIX téměř totožná s metodou Asirra. Test založený na této metodě tvoří tři obrázky a uživatel je vyzván, aby označil, co se na obrázku nachází. Na konkrétním případě (obrázek 15) je uživatel dotázán, aby označil robota na zobrazených obrázcích. [25]



Obrázek 15 - Ukázka testu s metodou SQ-PIX, zdroj: [38]

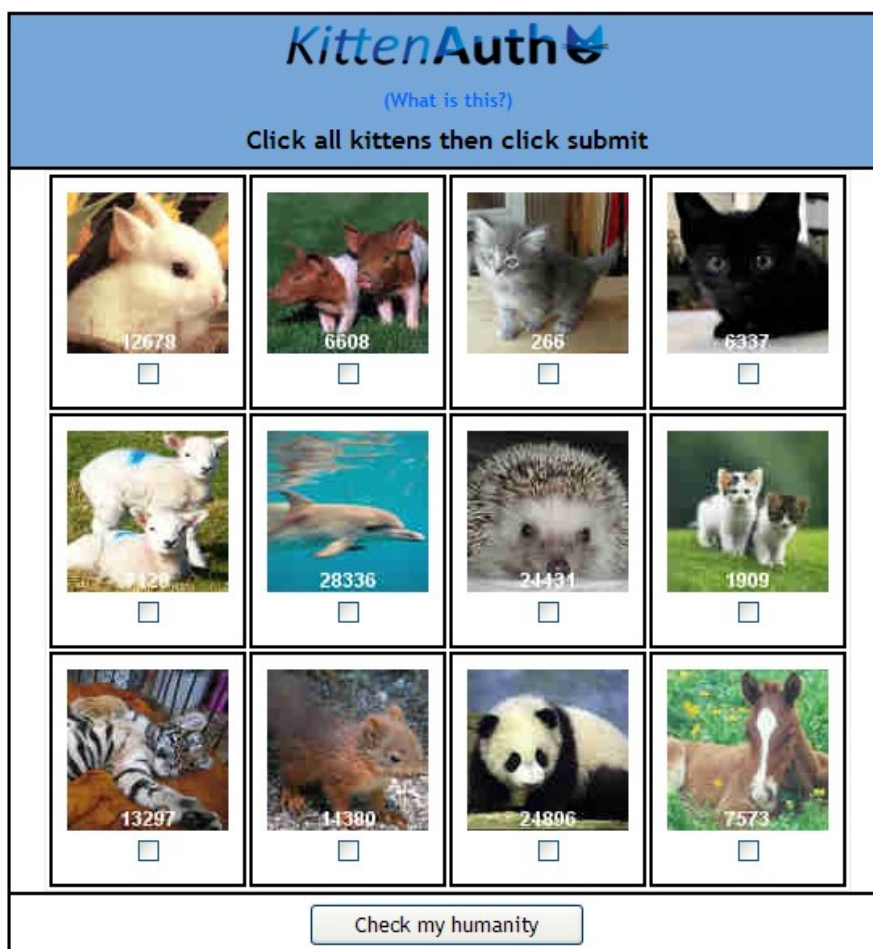
Nestačí však pouze kliknout na onen obrázek, ale je nutné dotazovaný objekt označit celý (viz obrázek 16), což může robotu, řešícím tento test velice ztížit práci. Test je sice dosti jednoduchý, ale ne pouze robot nemusí projít tímto testem. Stačí malá nepozornost a uživatel (člověk) test nesplní a je tedy neoprávněně považován za robota. [25]



Obrázek 16 - Ukázka řešení testu s metodou SQ-PIX, zdroj: [38]

6.3 KittenAuth

Projekt KittenAuth stejně jako již zmiňované metody ESP-PIX a SQ-PIX vychází z metody Asirra. Test je založený na principu rozeznání obrázků zvířat. Ukázka testu je zobrazena na obrázku 17, zde systém vyžaduje zaškrtnutí všech obrázků koťat z 12 zobrazovaných. Pokud je toto kritérium splněno, uživatel prošel testem. Test se v počátcích zaměřoval spíše jen na identifikaci obrázků s koťaty, od toho název KittenAuth. Později však byly do testu zahrnuty i identifikace jiných zvířat. [36]



Obrázek 17 - Ukázka testu s metodou KittenAuth, zdroj: [39]

Pokud budou do testu zahrnuty i obrázky exotických zvířat, ne příliš známých mnoha lidem, může se stát, že uživatel nebude schopen test vyřešit. Ne každý totiž ví, jak například vypadají surikaty či kapybary.

Metoda KittenAuth se nyní testuje pro použití v diskusních fórech založených na technologii phpBB, kde by mohla nahradit Captchu. [36]

6.4 Útoky na Asirru

V této kapitole jsou popsány možné útoky na Asirru s přihlédnutím na možnosti existence bezpečnostních děr.

6.4.1 Útoky skrz odkaz na Petfinder

Mnozí se domnívají, že odkaz „Adopt me“ může skrývat bezpečnostní díru v systému hlavně díky tomu, že na stránkách Petfinder je u každého obrázku napsáno, zda se jedná o psa či kočku. To však není možné, jelikož odkaz „Adopt me“ neodkazuje přímo na webové stránky Petfinder, ale na web service Asirry a ten následně poskytuje přímé přesměrování na Petfinder. Pokud se však uživatel pokusí vyřešit test poté, co se přes odkaz „Adopt me“ a následně přes web service Asirry prohlédl obrázky těchto domácích mazlíčků, je považován za robota. Je také možné omezit počet přesměrování na Petfinder na jednu IP adresu, aby bylo zabráněno zveřejnění větší části databáze. [35]

6.4.2 Útoky hrubou silou

Útoky hrubou silou jsou jednou z neúčinnějších činností, jak prolomit Asirru. Tyto útoky spočívají v hádání řešení testu, jde tedy o zkoušení náhodných kombinací pro jeho vyřešení. Pravděpodobnost prolomení Asirry za předpokladu, že útočník nemá žádné informace o testu (50% úspěšnost u každého obrázku), je 1:4096 pro 12 obrázků testu. Díky tomuto zjištění lze útok včas rozpoznat a zamezit jemu například blokováním IP adresy. A to v případě, pokud z této IP adresy bylo např. v jediném dni provedeno 500 pokusů o prolomení testu. Útočníci však dnes dokážou ovládat i 10 IP adres denně. A s existencí sítě botnet až tisíce IP adres, proto se musí vzít v potaz i myšlenka prolomení Asirry tímto způsobem. [35]

6.4.3 Útoky prostřednictvím třídění obrázků

Tyto druhy útoků již nespočívají v náhodném hádání řešení, ale v různých formách rozpoznávání obrázků, které útočnickovi dovolí, aby jeho odhady řešení byly lepší než odhady náhodné. Takováto technologie, která je schopná alespoň s 60% úspěšností oklasifikovat všechny obrázky (zda je na obrázku kočka nebo pes) je technologicky velmi náročná.

Disponuje-li však útočník touto technologií (což není příliš pravděpodobné), zvyšuje se úspěšnost prolomení testu z 1:4096 na 1:459. Přesto je nadále oklasifikování obrázků velmi obtížné, protože každý obrázek/fotografie je pořízena v jiném prostředí, v jiném úhlu záběru, při jiném osvětlení atd. [35]

6.4.4 Útoky prostřednictvím manuální rekonstrukce databáze

Dalším z možných útoků je manuální rekonstrukce databáze obrázků. Tento útok je však velice časově náročný, protože databáze Petfinder obsahuje přes 3 miliony obrázků a denně tato databáze naroste o dalších 10 000. Díky tomu by rekonstrukce této databáze trvala několik měsíců. Útočník však může využít i technologii, která ke každému obrázku vytvoří jeho histogram barev. Podle histogramu pak daný obrázek oklasifikuje (zda se jedná o kočku či psa). Během testu pak postupně z každého z 12 obrázků vytvoří také histogram, který porovná se všemi histogramy vytvořenými ke všem obrázkům z databáze. Pokud najde shodu s některým z oklasifikovaných histogramů, které má k dispozici, má řešení. [35]

6.4.5 Útoky na obrázky databáze

Z databáze Petfinder jsou pro test vybírány pouze aktuální obrázky, tedy ty obrázky zvířat, které tato organizace právě nabízí. Tyto obrázky tvoří pouze 10% z celé databáze a pouze těchto 10% je přístupné veřejnosti. Pro automatickou rekonstrukci databáze tak útočníkovi nezbývá nic jiného než napsat skript, který bude opakovaně stahovat aktuální obrázky z Petfinder, které následně oklasifikuje. [35]

7 Návrh vlastního řešení

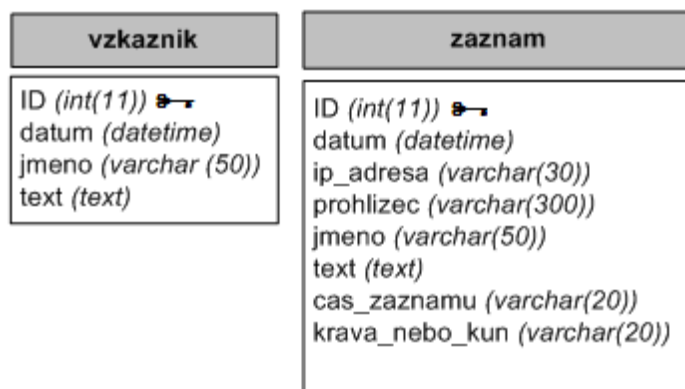
Protože metoda Asirra je mezi roboty dosud ne příliš známá a zatím jimi nepřekonaná, mým záměrem bylo tuto novinku otestovat v praxi. Za tímto účelem jsem vytvořila vlastní test vycházející z této metody, který jednak zabezpečuje diskusní fórum proti robotům a jednak rozlišuje člověka a robota vstupujícího do tohoto fóra či vkládajícího příspěvek. Zároveň jsem navrhla dvě možná řešení provedení testu.

Test v každém z obou řešení se skládá z 8 náhodně vygenerovaných obrázků, kdy je vždy 7 obrázků koní a 1 obrázek krávy. Pro tento test jsem záměrně vybrala obrázky koní a krav, protože tato zvířata si jsou vzhledově velmi podobná – částečně stavbou těla, více pak zbarvením. Tato podobnost pak následně hraje významnou roli při rozpoznávání těchto zvířat v testu – uživatelé s rozpoznáním koně a krávy by neměli mít problém, roboti však ano. Kdybychom například obrázky těchto zvířat zaměnili za geometrické obrazce, takový test by byl naprosto neúčinný, protože roboti jsou již tak vyspělí, že od sebe umí rozeznat různé geometrické tvary a testem by tedy prošli. Proto nejlepším řešením je využití takových objektů, které jsou si podobné svým tvarem a současně se odlišují svým zbarvením. Je tak možné navrhnout různá řešení rozpoznávání typů automobilů, druhů opic či ptáků a jiné. Stále je ale nutné brát v potaz znalosti testovaných uživatelů, protože ne každý dokáže odlišit kočkodana od makaka ze zástupců druhů opic či mercedes od audi ze zástupců druhů automobilů. Proto jsem test navrhla co nejjednodušší, koně a krávy zná jistě každý.

V databázi jsem vytvořila 2 tabulky: tabulku vzkaznik a tabulku zaznam, jejichž primárním klíčem je ID (viz obrázek 18). Hodnota primárního klíče je pro každý záznam unikátní [40].

Jednotlivé sloupce tabulky jsou vymezeny svými datovými typy, například sloupec datum se bude vypisovat ve formátu RRRR-MM-DD HH:MM:SS. Do tabulky vzkaznik se zapisuje pouze to, co se zobrazuje v diskusi a do tabulky zaznam se zapisuje vše, jak odeslané příspěvky, tak i pouhé návštěvy fóra bez přidání příspěvku.

Na obrázku 18 jsou zachyceny databázové tabulky pro první řešení testu, pro druhé řešení by byly totožné, pouze v tabulce zaznam by místo atributu krava_nebo_kun byl atribut pozice_zaznam stejného datového typu.



Obrázek 18 – Databázové tabulky, zdroj: autor

Při přístupu do fóra se automaticky loguje do tabulky zaznam jedinečný identifikátor ID a datum, které představuje aktuální datum a čas načtení stránky. Poté se příkazem **REMOTE_ADDR** zjišťuje IP adresa uživatele. Dále se loguje typ a verze webového prohlížeče, který se generuje z http požadavku, a také verze operačního systému používaného tímto uživatelem. Následně je zaznamenáno jméno a text, které zadal uživatel, přenášené z formuláře metodou POST. Čas záznamu, který představuje dobu trvání psaní příspěvku, se zjišťuje tak, že od aktuálního času (času odeslání příspěvku, který vrací PHP funkce Time()) je odečten čas načtení stránky (ten je uložen v proměnné čas, která je předávána metodou POST). Nakonec v případě prvního řešení testu je do sloupce krava_nebo_kun zapsáno číslo obrázku, na které kliknul uživatel, a název objektu, který se na daném objektu nacházel – tedy kráva nebo kůň. V případě druhého řešení je do sloupce pozice_zaznam zapsána pozice obrázku, kterou zadal uživatel a název objektu, který se na daném objektu nacházel.

8 Implementace

V této kapitole jsou vysvětleny některé důležité části zdrojového kódu obou navržených řešení s ukázkami grafického rozhraní. Diskusní fórum v obou řešeních bylo po dobu více jak 2 měsíců umístěno u hostingové služby webzdarma.cz.

8.1 První řešení

První řešení (dále jen vzkazník č. 1) je založeno na myšlence metody Asirra. Pro odeslání příspěvku do diskuse je nutné kliknout na obrázek krávy a příspěvek je následně odeslán.

Interpretace vybraných částí zdrojového kódu:

```
$krava = false;  
if (in_array($cislo_obrazku, $kravy)) $krava=true;
```

Tento příkaz zjišťuje, zda uživatel kliknul na obrázek krávy. Pro jednodušší práci jsou všechny obrázky krav a koní pojmenovány čísly. Funkce `in_array()` porovnává číslo obrázku, na které kliknul uživatel s číslem obrázku z pole `$kravy`. Pokud je nalezena shoda, nastaví se proměnná `$krava` na `true`.

```
if ($text){  
    if ($krava)  
        {$krava_nebo_kun = $cislo_obrazku." krava";}   
    else {$krava_nebo_kun = $cislo_obrazku." kun";}   
    if ($cas+5 > time()) {$krava_nebo_kun="robot";}   
else {$krava_nebo_kun = "";}   
  
$cas_zaznam = time()-$cas;  
if ($cas=="") {$cas_zaznam=""};
```

Tato část kódu pouze vytváří řetězec obsahující informaci o tom, na který obrázek uživatel kliknul pro odeslání příspěvku. Pokud to byl obrázek krávy, do proměnné \$krava_nebo_kun se uloží číslo tohoto obrázku a v databázi se objeví informace např. 13 krava (obrázek č. 13, na němž je kráva). V opačném případě, kdy uživatel zvolil obrázek koně, se provede to samé s informací, že šlo o obrázek koně.

Dále může nastat případ, kdy je příspěvek vložen do 5 sekund od načtení stránky. Stane-li se tak, je velice pravděpodobné, že tento příspěvek napsal robot a do proměnné \$krava_nebo_kun je uložen řetězec „robot“. Roboti, na rozdíl od uživatelů, pracují velmi rychle a jsou schopni během sekundy vložit i několik příspěvků najednou.

V dalším kroku je do proměnné \$cas_zaznamu uložen čas, znázorňující, jak dlouho uživateli trvalo napsání příspěvku od načtení stránky po jeho odeslání.

Zaznamenávány jsou i pouhé návštěvy tohoto diskusního fóra bez vložení příspěvku. Pokud uživatel toto fórum navštíví a neodešle příspěvek, proměnná \$text bude prázdná, díky tomu i proměnná \$krava_nebo_kun bude prázdná. A současně bude prázdná i proměnná \$cas (čas odeslání příspěvku), a protože v proměnné \$cas_zaznamu bude v tomto případě uložen pouze aktuální čas (což není žádoucí), nastaví se tato proměnná na prázdný řetězec.

```
if ($text){
    $ochrana = mysql_query ("select (text) from vzkaznik order by
        datum desc limit 1");
    $ochrana_z = mysql_fetch_array($ochrana);
    if ($text!= $ochrana_z["text"]){
        if ($cas+5 < time()){
            if ($krava){
                mysql_query ("insert into vzkaznik values
                    ('$datum', '$jmeno', '$text')"); }
            else kun();}
        else rychlik();}}
```

Z důvodu zamezení opakovaného uložení příspěvku do databáze při obnovení stránky se zde porovnává text naposledy uloženého příspěvku s právě ukládaným, a pokud se shodují, nový příspěvek se neuloží. Toto opatření samozřejmě selhává v případě, že je mezi uložení příspěvku a obnovením stránky vložen příspěvek odlišný.

Nejdříve je vybrán sloupec „text“ z prvního řádku tabulky vzkaznik a jeho obsah je následně porovnán s právě odeslaným příspěvkem.

Jestliže čas načtení stránky po přičtení 5 sekund je větší než aktuální čas (aktuální čas vrací PHP funkce `time()` ve formátu počtu sekund uplynulých od 1. 1. 1970) a jestliže uživatel kliknul na obrázek krávy, pak jsou do tabulky vzkaznik vloženy hodnoty datum, jméno a text. Pokud však čas načtení stránky + 5 sekund je menší než aktuální čas, ale uživatel kliknul na obrázek koně, pak se spustí funkce `kun()`. Není-li splněna ani jedna podmínka, pak se spustí funkce `rychlik()`.

```
$a= array ();
$obr= array ();
$c=0;
$d= array ();
do {
    $d=array_merge($obr,$a);
    $obr = array_unique($d);
    $c=count($obr);
    $a = array_fill(0, 8-$c, $kone[rand(0,21)]); }
while ($c!=7);

$obr[7]=$kravy[rand(0,11)];
shuffle($obr);
```

Tento příkaz se provede vždy, jde o poskládání obrázků v testu. Nejdříve funkce `array_fill()` naplní prvních osm pozic pole `$a` stejným náhodným číslem z pole `$kone`. Pole `$obr` se sloučí s polem `$a` do pole `$d`, které se vyčistí od duplicit a výsledek se uloží do pole `$obr`. Dále se spočítá počet prvků v poli `$obr` a tento údaj se využije k zajištění správného počtu cyklů. V dalším cyklu se opět pole `$a` naplní stejným náhodným číslem z pole `$kone`, avšak počet naplněných pozic bude 8 minus počet prvků v poli `$obr`. Pole `$a` se opět sloučí s polem `$obr` do pole `$d`, jsou odstraněny duplicity a takto cyklus pokračuje až do naplnění pole `$obr` sedmi různými náhodnými čísly z pole `$kone`.

Obrázku na poslední pozici, tedy na pozici 7 (pozice od 0 do 7) se přiřadí náhodné číslo z pole `$kravy`. Nakonec je pole `$obr` promícháno.

Níže si lze prohlédnout ukázkou grafického rozhraní pro test na bázi 1. řešení.

Jméno:

Vzkaz:

pro odeslání klikněte na **KRÁVU**



Obrázek 19 - Ukázka vlastního testu na bázi 1. řešení, zdroj: autor

8.2 Druhé řešení

Druhé řešení (dále jen vzkazník č. 2) je také založeno na myšlence metody Asirra. Na rozdíl od vzkazníku č. 1 jsou obrázky zobrazeny do matice a pro odeslání příspěvku do diskuse je nutné napsat pozici, na které se nachází obrázek krávy v této matici, poté je příspěvek odeslán.

V interpretaci vybraných částí kódu jsou vysvětleny pouze změny od vzkazníku č. 1, ostatní části kódu jsou téměř totožné.

Interpretace vybraných částí zdrojového kódu:

```
$krava=false;
$pozice_zaznam="";
if ($misto)
  {$pozice_zaznam=$pozice." kun";
  if ($misto==$pozice)
    {$krava=true;
    $pozice_zaznam=$pozice." krava"; }}

```

Tento algoritmus porovnává pozici, kterou zadal uživatel s aktuální pozicí obrázku krávy. Pokud se tyto údaje shodují, proměnná \$pozice_zaznam nabývá hodnoty např. "a2 krava", v opačném případě např. "a2 kun".

```
shuffle($obr);
$pozice=0;
for ($t=0; $t<=7; $t++) {
    for($x=0; $x<=11; $x++) {
        if ($kravy[$x]==$obr[$t]) {$pozice=$t; }}
}

$souradnice= array
("a1"=>0, "b1"=>1, "c1"=>2, "d1"=>3, "a2"=>4, "b2"=>5, "c2"=>6, "d2"=>7);
$prirad=array_search ($pozice, $souradnice);
```

Tento výňatek zdrojového kódu zjišťuje, na které pozici je obrázek krávy, protože po promíchání obrázků příkazem shuffle(\$obr) tuto pozici neznáme. Zde každý z 8 obrázků porovnáme s celým polem \$kravy (pole \$kravy obsahuje čísla obrázků, na kterých jsou krávy). Pokud najdeme obrázek, který se shoduje s některým z obrázků z pole \$kravy, jeho pozice se uloží do proměnné \$pozice.

Funkce array_search() přiřadí číslu pozice její souřadnice, které jsou definovány v poli \$souradnice. Je-li například v proměnné \$pozice uložena hodnota 4 (na 4 pozici je obrázek krávy), pak se do proměnné \$prirad uloží a2.

```
<input type="hidden" name="prirad" value="<?php echo $prirad; ?>">
```

V HTML kódu je definované skryté pole prirad, které odešle pozici obrázku krávy na následující stránku. Tuto záležitost lze řešit i pomocí cookies, sessions, kde je proměnná dostupná na všech stránkách, či předáváním parametru v url adrese. V tomto případě však nebylo potřeba, aby byla proměnná dostupná na všech stránkách, proto jsem využila skryté pole a předávání proměnné pomocí atributu value.

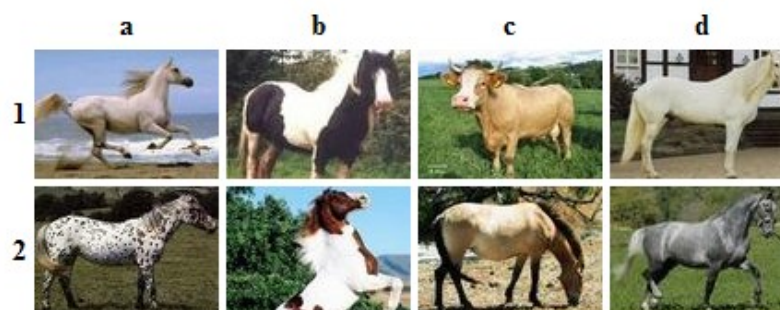
Níže si lze prohlédnout ukázkou grafického rozhraní pro test na bázi 2. řešení.

Jméno:

Vzkaz:

Pozice:

Pro odeslání příspěvku napište pozici, na které se nachází **KRÁVA!**
Vždy začínejte nejprve písmenem a následně číslicí.
Například: a1, b2, c3...



Obrázek 20 – Ukázkou vlastního testu na bázi 2. řešení, zdroj: autor

Pro přilákání robotů na své webové stránky jsem využila několik možností:

- do meta tagu keywords jsem zapsala slova, která roboti nejčastěji vyhledávají, např. phpBB, WordPress, Cyberhome Forum a jiné. Jde o různá diskusní fóra a blogy, která jsou známá svými bezpečnostními dírami a tak jsou neustále roboty vyhledávána. Navíc tato slova zvyšují váhu slov v textu, [41]
- požádala jsem o indexaci své stránky několik vyhledávačů jako například: Google.com, Seznam.cz, Gigablast.com, Zoohoo.cz, Web-friend.com, SearchEngine.com, WhatUseek.com a jiné,
- provedla jsem výměnu odkazů mezi několika správci webových stránek, protože čím více webových stránek na mé stránky odkazuje, tím lepší hodnocení získají mé stránky u mnoha vyhledávačů, čímž se zajistí i vyšší návštěvnost webu,
- vytvořila jsem fórum založené na technologii phpBB, na kterém jsem umístila odkazy na své webové stránky a umožnila zde robotům plný přístup do tohoto fóra.

9 Výsledky testování

Vzkazník č. 1:

Na webových stránkách se vzkazníkem č. 1 bylo zaznamenáno celkem 563 přístupů a přidáno bylo 218 příspěvků, z nichž 22 bylo označeno za špatné – uživatel pro odeslání příspěvku zvolil obrázek koně a příspěvek byl tedy zaznamenán, ale nebyl přidán mezi zobrazované příspěvky. Také bylo zaznamenáno 5 přístupů robotů od vyhledávačů a 6 uživatelů bylo označeno za robota. Za robota je uživatel označen v případě, že svůj příspěvek odešle do diskuse do 5 sekund od načtení stránky.

Zároveň bylo analyzováno několik podezřelých uživatelů. Uživatel s IP adresou 85.70.52.225, který využívá webový prohlížeč vystupující jako Firefox 3.0.8 vložil do diskuse celkem 12 příspěvků (viz tabulka 1). To by nebylo podezřelé, kdyby zmiňované příspěvky nebyly vloženy téměř vždy 2 sekundy po sobě a u každého příspěvku by nebyl zvolen stále ten samý obrázek krávy, vyjímaje poslední příspěvek vložený tímto uživatelem. Při každém odeslání příspěvku je totiž pokaždé stránka obnovena a vygeneruje se nový obrázek krávy. Zároveň je pro člověka téměř nemožné během 2 sekund do fóra vložit vždy jiný příspěvek. Fórum je totiž ošetřeno proti odesílání příspěvků se stejným obsahem.

Vysvětlivky k tabulce: čas záznamu představuje počet sekund uplynulých od načtení stránky po odeslání příspěvku a datum znázorňuje aktuální datum a čas v momentě načtení stránky.

Tabulka 1 - Statistika přístupů podezřelého uživatele I., zdroj: autor

ID	Datum	IP adresa	Prohlížeč	Čas záznamu [s]	Obrázek
1	4.4.2009 16:11:20	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8		
2	4.4.2009 16:12:02	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	42	32 krava
3	4.4.2009 16:23:23	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	723	32 krava
4	4.4.2009 16:23:28	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	728	32 krava
5	4.4.2009 16:23:30	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	730	32 krava
6	4.4.2009 16:23:32	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	732	32 krava
7	4.4.2009 16:23:33	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	733	32 krava
8	4.4.2009 16:23:35	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	735	32 krava

ID	Datum	IP adresa	Prohlížeč	Čas záznamu [s]	Obrázek
9	4.4.2009 16:23:37	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	737	32 krava
10	4.4.2009 16:23:38	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	738	32 krava
11	4.4.2009 16:23:40	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	740	32 krava
12	4.4.2009 16:23:42	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	742	32 krava
13	4.4.2009 16:24:25	85.70.52.225	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	43	24 krava

Téměř totožný případ se objevil i u uživatele s IP adresou 213.29.120.72 (viz tabulka 2), využívající stejný webový prohlížeč jako výše zmiňovaný uživatel. Zde uživatel vložil do diskuse celkem 20 příspěvků a také u mnoha příspěvků se objevuje ten samý obrázek krávy, což z již výše zmiňovaného důvodu není možné. Zde je možné shlédnout statistiku přístupů tohoto uživatele.

Tabulka 2 - Statistika přístupů podezřelého uživatele II., zdroj: autor

ID	Datum	IP adresa	Prohlížeč	Čas záznamu [s]	Obrázek
1	4.4.2009 21:22:51	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8		
2	4.4.2009 21:23:50	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	59	10 krava
3	4.4.2009 21:24:13	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8		
4	4.4.2009 21:25:07	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	54	32 krava
5	4.4.2009 21:25:10	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	57	32 krava
6	4.4.2009 21:25:13	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	60	32 krava
7	4.4.2009 21:25:14	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	61	32 krava
8	4.4.2009 21:25:15	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	62	32 krava
9	4.4.2009 21:25:16	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	63	32 krava
10	4.4.2009 21:25:16	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	63	32 krava
11	4.4.2009 21:25:16	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	63	32 krava
12	4.4.2009 21:25:17	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	64	32 krava
13	4.4.2009 21:25:17	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	64	32 krava
14	4.4.2009 21:25:17	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	64	32 krava

ID	Datum	IP adresa	Prohlížeč	Čas záznamu [s]	Obrázek
15	4.4.2009 21:25:18	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	65	32 krava
16	4.4.2009 21:25:51	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8		
17	4.4.2009 21:26:20	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	28	4 krava
18	5.4.2009 0:23:12	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8		
19	5.4.2009 0:24:12	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	57	6 krava
20	5.4.2009 0:24:19	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	64	6 krava
21	5.4.2009 0:24:22	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	67	6 krava
22	5.4.2009 0:24:29	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	74	6 krava
23	5.4.2009 0:25:21	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	51	20 krava
24	5.4.2009 0:27:47	213.29.120.72	Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	144	25 krava

Jedním z dalších nestandardních případů je i uživatel s IP adresou 85.135.22.198, který využívá webový prohlížeč vystupující jako Firefox 3.0.8 a uživatel s IP adresou 84.242.77.81 využívající tentýž webový prohlížeč. Počet zaznamenaných přístupů do fóra těmito uživateli dosahuje až 5 přístupů během jedné sekundy, což je neobvyklé a pro člověka těžce zvládnutelné. Zároveň oba tito uživatelé byli v jednom ze tří případů označeni za robota, což je podle mého názoru správné označení.

Neposledním z nestandardních uživatelů je uživatel s IP adresou 213.226.232.66 (viz tabulka 3), který využívá webový prohlížeč vystupující také jako Firefox 3.0.8 u již zmíněných uživatelů. Tento uživatel se do diskuse pokoušel vložit celkem 3 příspěvky. První příspěvek do diskuse nebyl vložen, protože uživatel zvolil pro odeslání příspěvku obrázek koně. Ostatní dva příspěvky také nebyly vloženy do diskuse, protože uživatel oba příspěvky odeslal během 4 sekund od načtení stránky a byl tedy označen za robota.

Tabulka 3 - Statistika přístupů podezřelého uživatele III., zdroj: autor

ID	Datum	IP adresa	Prohlížeč	Čas záznamu [s]	Obrázek
1	26.4.2009 16:44:52	213.226.232.66	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5		
2	26.4.2009 16:45:11	213.226.232.66	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5		

ID	Datum	IP adresa	Prohlížeč	Čas záznamu [s]	Obrázek
3	26.4.2009 16:45:24	213.226.232.66	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5	13	2 kun
4	26.4.2009 16:45:28	213.226.232.66	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5	4	robot
5	26.4.2009 16:45:31	213.226.232.66	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5		
6	26.4.2009 16:45:35	213.226.232.66	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5	4	robot
7	26.4.2009 16:46:06	213.226.232.66	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5		

Vzkazník č. 2:

Na webových stránkách se vzkazníkem č. 2 bylo zaznamenáno celkem 201 přístupů a přidáno bylo 36 příspěvků, z nichž 12 bylo označeno za špatné. Zaznamenalo se i 13 přístupů robotů od vyhledávačů a jeden uživatel byl označen za robota, a to z důvodu odeslání svého příspěvku do 5 sekund od načtení stránky.

U druhého řešení testu bylo taktéž analyzováno několik podezřelých uživatelů. Jde především o uživatele s IP adresou 212.90.248.182 používající webový prohlížeč tvářící se jako Firefox 3.0.8. Tento uživatel se pokoušel v krátkém časovém intervalu vložit do diskuse celkem 4 příspěvky (viz tabulka 4), z nichž 3 do diskuse přidány nebyly z těchto důvodů:

- uživatel dvakrát zadal nesprávnou souřadnici obrázku krávy
- jedenkrát odeslal příspěvek do 5 sekund od načtení stránky, tudíž byl označen za robota a jeho příspěvek byl pouze zaznamenán, ale nebyl vložen do diskuse.

Tabulka 4 - Statistika přístupů podezřelého uživatele IV., zdroj: autor

ID	Datum	IP adresa	Prohlížeč	Čas záznamu [s]	Pozice
1	18.4.2009 22:05:43	212.90.248.182	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8		
2	18.4.2009 22:06:12	212.90.248.182	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	29	d1 kun
3	18.4.2009 22:06:17	212.90.248.182	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	5	robot

ID	Datum	IP adresa	Prohlížeč	Čas záznamu [s]	Pozice
4	18.4.2009 22:06:24	212.90.248.182	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	7	a1 kun
5	18.4.2009 22:06:39	212.90.248.182	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8	15	a2 krava
6	18.4.2009 22:07:05	212.90.248.182	Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.9.0.8) Gecko/2009032609 Firefox/3.0.8		

Dalším nestandardním případem je uživatel s IP adresou 78.128.153.189 využívající prohlížeč Crazy Browser, který se pokoušel do diskuse vložit celkem 3 příspěvky, ale neuspěl. Vždy pro odeslání příspěvku zadal nesprávnou pozici, tedy pozici obrázku koně, a proto příspěvek nebyl vložen do diskuse. Přestože příspěvky nebyly vloženy v příliš krátkém časovém okamžiku (v rozmezí jedné minuty), nelze vyloučit, že tímto uživatelem by mohl být robot. Lze však také spekulovat, zda se nejedná pouze o nepozorného uživatele.

S neustálým rozvojem informačních technologií se vyvíjí i schopnosti robotů. Ještě nedávno bylo možné robota jednoduše identifikovat podle času odeslání příspěvku, což je u některých robotů stále možné, ale u některých sofistikovanějších nikoliv. Tito roboti, kteří dříve odesílali formuláře hromadně ihned po načtení stránky, dnes s odesláním formulářů vyčkávají, aby vzbudili zdání, že jsou běžnými uživateli. Někteří roboti již dokázali obejít kontroly doby psaní příspěvku díky rozpoznání pole s informací o čase vstupu na stránku. A tak místo toho, aby vyčkávali, při odeslání příspěvku od tohoto času automaticky odečtou určitý počet sekund a příspěvek je tak bez pochybností odeslán do diskuse. Účinným řešením je pak šifrování pole s informací o čase vstupu na stránku.

Dalším možným způsobem identifikace by mohla být analýza webového prohlížeče, který uživatel používá. Vyhledávací roboti mají v tomto řetězci slova typu bot, robot či spider, avšak spamboti se stále více schovávají za běžně používané prohlížeče, takže je prakticky nemožné je tímto způsobem odhalit.

Účinná identifikace uživatelů by podle mého názoru:

- neměla obtěžovat uživatele, např. opisováním dlouhého a nečitelného kódu,
- měla by v sobě zahrnovat kombinace různých metod, např. kontrola vyplnění skrytého pole + kontrola doby psaní příspěvku + šifrování času vstupu na stránku + kontrola klíčových slov v obsahu příspěvku + rozpoznání objektů na obrázcích a jiné,
- měla by být nenáročná z hlediska systémových prostředků.

10 Závěr

Cílem této práce bylo navržení testu rozlišujícího člověka a robota na webových aplikacích, jeho otestování a zhodnocení. Pro splnění těchto úkolů jsem vytvořila na základě metody Asirra prostřednictvím jazyka PHP dva testy identifikující robota a člověka. Oba fungují na jednoduchém principu rozpoznávání obrázků zvířat. První test (vzkazník č. 1) je podle mého názoru uživatelsky přívětivější než test druhý (vzkazník č. 2), kdy je nutné uživatele obtěžovat, aby napsal pozici dotazovaného obrázku a následně kliknul na tlačítko odeslat. Druhý test však více působí jako „lákadlo“ na roboty, kteří vyplňují všechna pole formuláře, a tak je-li v poli pozice vepsáno cokoliv jiného než pozice některého obrázku, lze uvažovat, že tento příspěvek vložil robot. Samozřejmě pro správnou identifikaci, zda se opravdu jednalo o robota, je nutné využít zároveň s tímto skrytým polem ještě nějakou metodu identifikace, např. kontrolu údajů jako čas odeslání příspěvku a samotný obsah příspěvku. Zároveň jsem provedla kroky potřebné ke zviditelnění svých webových stránek popsané v kapitole 9.2. U vzkazníku č. 1 bylo identifikováno pět a u vzkazníku č. 2 dva podezřelí uživatelé.

Slabinu tohoto testu vidím v malém počtu obrázků, pouhých 34, na rozdíl od testu společnosti Microsoft s metodou Asirra (viz obrázek 13), využívající databázi až 3 milionů obrázků, která se každým dnem navyšuje. S použitím databáze této velikosti by byl test mnohem efektivnější.

Pro spuštění mého řešení testu je zapotřebí pouze webový server s PHP verzí alespoň 4.x a databázový systém MySQL.

Převážná většina uživatelů tohoto fóra oba testy hodnotila velice kladně a uvedla, že by je uvítala více než testy s hojně využívanou metodou Captcha. Uživatelé však více preferují test na bázi prvního řešení, který uživatele příliš neobtěžuje opisováním souřadnice obrázku jako test na bázi druhého řešení.

Použité zdroje

- [1] *The Web Robots Pages* [online]. c2007 [cit. 2009-04-09]. Dostupný z WWW: <<http://www.robotstxt.org/>>
- [2] *Chatbot Popularity* [online]. c2008-2009 [cit. 2009-02-21]. Dostupný z WWW: <http://www.jeeney.com/default.aspx?page=chatbot_popularity>
- [3] KORANDA, Petr. *SEO - jak pracují vyhledávací roboti* [online]. c2007-2008, 14.4.2008 [cit. 2009-02-23]. Dostupný z WWW: <<http://www.peakpointnet.cz/cz/piseme/clanky/seo-jak-pracuji-vyhledavaci-roboti>>
- [4] ŘEHORŮ, Michal. Hrozby : Internetový (ro)bot. *Svět hostingu : Magazín pro všechny tvůrce a vlastníky webů* [online]. 2008 [cit. 2008-02-18]. Dostupný z WWW: <<http://svethostingu.ignum.cz/2008-08.pdf>>. ISSN 1801-3392
- [5] GOLBECK, Jennifer, MUTTON, Paul. *Semantic Web Interaction on Internet Relay Chat* [online]. [2004] [cit. 2009-02-21]. Dostupný z WWW: <<http://www.mindswap.org/papers/irc.pdf>>
- [6] HODGES, Andrew . *The Alan Turing Home Page* [online]. c1995 [cit. 2009-04-10]. Dostupný z WWW: <<http://www.turing.org.uk/turing/>>
- [7] MICHÁLEK, Bedřich. *Turingův test - může počítač myslet?* [online]. 2003 [cit. 2009-03-04]. Dostupný z WWW: <<http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xmichal4.htm>>
- [8] PETERKA, Jirí. *Turing test* [online]. 1994 [cit. 2009-03-04]. Dostupný z WWW: <<http://www.earchiv.cz/a94/a434c120.php3>>
- [9] BLAŽEK, Jakub. *Porovnání procesů probíhajících v lidském mozku a počítači*. [s.l.], 2006. 73 s. MASARYKOVA UNIVERZITA FILOZOFICKÁ FAKULTA. Vedoucí bakalářskou diplomovou PhDr. Michal Lorenz. Dostupný z WWW: <http://is.muni.cz/th/99702/ff_b/bakalarka.txt>

- [10] CAJTHAML, Martin. Ochrana formulářů proti spamu. *Symbio : internetová agentura* [online]. 2006 [cit. 2009-03-08]. Dostupný z WWW: <<http://www.symbio.cz/clanky/ochrana-formularu-proti-spamu.html>>
- [11] SEO [online]. [2009] [cit. 2009-04-10]. Dostupný z WWW: <<http://www.kao.cz/prostredky/seo.php>>
- [12] BALABÁNOVÁ, Tereza . *Problematika přístupnosti webů knihoven*. [s.l.], 2008. 145 s. Masarykova univerzita, Filozofická fakulta, Ústav české literatury a knihovnictví. Vedoucí diplomové práce Ing. Zdeněk Kadlec, Dr. Dostupný z WWW: <http://is.muni.cz/th/109381/ff_m/DP.txt>
- [13] *Hodnocení komentářů - ochrana proti komentářovému spamu* [online]. c2005 [cit. 2009-03-11]. Dostupný z WWW: <http://www.hellish.cz/ochrana_proti_komentarovemu_spamu/>
- [14] KÁRA, Michal. Jak funguje bayesovský antispamový filtr? (1.). *Lupa.cz : server o českém internetu* [online]. 2005 [cit. 2009-03-11]. Dostupný z WWW: <<http://www.lupa.cz/clanky/jak-funguje-bayesovsky-antispamovy-filtr-1/>>
- [15] HUMPÁL, Matěj. *(Můj) boj s komentářovým spamem* [online]. c2006 [cit. 2009-03-11]. Dostupný z WWW: <<http://weblog.finwe.info/item/muj-boj-s-komentarovym-spamem>>
- [16] MACICH, Jiří. Spam v diskuzních fórech: marný boj?. *Lupa.cz : server o českém internetu* [online]. 2007 [cit. 2009-03-11]. Dostupný z WWW: <<http://www.lupa.cz/clanky/spam-v-diskuznich-forech-marny-boj/>>
- [17] POLZER, Jan. Mollom: komentářový antispam a zkušenosti z praxe. *Maxiorel* [online]. 2008 [cit. 2009-03-23]. Dostupný z WWW: <<http://www.maxiorel.cz/mollom-komentarovy-antispam-zkusenosti-z-praxe>>
- [18] *Mollom* [online]. 2008 [cit. 2009-03-23]. Dostupný z WWW: <<http://mollom.com/>>
- [19] ČÍŽEK, Jakub. CAPTCHA: Jak se stát otrokem podivného obrázku. *Živě.cz* [online]. 2008 [cit. 2008-03-15]. Dostupný z WWW: <<http://www.zive.cz/Clanky/CAPTCHA-Jak-se-stat-otrokem-podivneho-obrazku/sc-3-a-144482/default.aspx>>

- [20] ŠPINAR, David. CAPTCHA, aneb lze odlišit člověka a robota přístupným způsobem?. *Přístupnost* [online]. 2008 [cit. 2009-03-16]. Dostupný z WWW: <<http://pristupnost.nawebu.cz/weblog/blogpost.php?post=81>>
- [21] *Captcha Creator PHP Script 1.2* [online]. c2004-2009 [cit. 2009-04-15]. Dostupný z WWW: <http://www.sharewareplaza.com/Captcha-Creator-PHP-Script-download_50739.html>
- [22] *Hellshare.com* [online]. c2009 [cit. 2009-03-16]. Dostupný z WWW: <<http://download.cz.hellshare.com/>>
- [23] ZITTRAIN, Jonathan. *Jonathan Zittrain : Professor of Internet Governance and Regulation, Oxford Internet Institute* [online]. [2000] [cit. 2009-03-16]. Dostupný z WWW: <<http://people.oii.ox.ac.uk/z/page/2/>>
- [24] *Digitizing Books One Word at a Time* [online]. Carnegie Mellon University, c2009 [cit. 2009-03-20]. Dostupný z WWW: <<http://recaptcha.net/learnmore.html>>
- [25] ČÍŽEK, Jakub. Blíží se konec klasického CAPTCHA testu?. *Živě.cz* [online]. 2009 [cit. 2009-03-22]. Dostupný z WWW: <<http://www.zive.cz/Clanky/Blizi-se-konec-klasickeho-CAPTCHA-testu/sc-3-a-145402/default.aspx>>
- [26] *IMAGINATION : Image-based Authentication* [online]. c2006-2008 [cit. 2009-04-10]. Dostupný z WWW: <<http://goldbach.cse.psu.edu/s/captcha/>>
- [27] KAPLAN, Michael. *The 3-D CAPTCHA* [online]. [2009] [cit. 2009-04-03]. Dostupný z WWW: <<http://spamfizzle.com/CAPTCHA.aspx>>
- [28] *Yuniti : go something good* [online]. c2009 [cit. 2009-04-03]. Dostupný z WWW: <<http://www.yuniti.com/register.php> >
- [29] Google captcha úspěšně prelomená. *Feed Subscription* [online]. 2008 [cit. 2009-03-20]. Dostupný z WWW: <<http://blog.synopsi.com/2008-02-26/google-captcha-uspesne-prelomena>>

- [30] Ochrana CAPTCHA Hotmailu lze prolomit za 6 sekund. *SecurityWorld* [online]. 2008 [cit. 2009-03-20]. Dostupný z WWW: <<http://securityworld.cz/securityworld/ochranu-captcha-hotmailu-lze-prolomit-za-6-sekund-670>>
- [31] Ochrana Captcha Googlu byla zřejmě prolomena. *SecurityWorld* [online]. 2008 [cit. 2009-03-20]. Dostupný z WWW: <<http://securityworld.cz/securityworld/ochrana-captcha-googlu-byla-zrejme-prolomena-208>>
- [32] Jak bojovat proti spamu v phpBB3. *PhpBB : vytváříme komunitu* [online]. 2009 [cit. 2009-03-25]. Dostupný z WWW: <<http://www.phpbb.cz/viewtopic.php?t=15950>>
- [33] *Captcha na phpBB3 prolomena*. [online]. c2000-2009 [cit. 2009-03-25]. Dostupný z WWW: <<http://webtrh.cz/29417-captcha-phpbb3-prolomena>>
- [34] PAGET, Francois . Cracking CAPTCHA: Another Russian Business. *McAfee* [online]. 2008 [cit. 2009-03-20]. Dostupný z WWW: <<http://www.avertlabs.com/research/blog/index.php/2008/10/10/cracking-captcha-as-a-business/>>
- [35] *Asirra* [online]. Microsoft Corporation, 2009 [cit. 2009-03-22]. Dostupný z WWW: <<http://research.microsoft.com/en-us/projects/asirra/default.aspx>>
- [36] *Building a better spam-blocking CAPTCHA* [online]. [2009] [cit. 2009-03-25]. Dostupný z WWW: <<http://wang.ist.psu.edu/docs/related/reports/2009/ComputerWorld/computerworld.htm>>
- [37] *The Captcha Project* [online]. Carnegie Mellon University, c2004 [cit. 2009-03-24]. Dostupný z WWW: <<http://www.captcha.net/cgi-bin/esp-pix>>
- [38] *SQ-PIX* [online]. [2009] [cit. 2009-03-24]. Dostupný z WWW: <<http://www.toallwhosekit.net/cgi-bin/sq-pix>>
- [39] *KittenAuth* [online]. c2001-2007 [cit. 2009-03-24]. Dostupný z WWW: <<http://www.thepcspy.com/kittenauth>>

[40] KOSEK, Jiří. *PHP : tvorba interaktivních internetových aplikací*. [s.l.] : Grada Publishing, 1999. 492 s. ISBN 80-7169-373-1

[41] SIROVICH, Jamie, DARIE, Christian. *SEO v PHP : Programujeme profesionálně*. Ondřej Žížka. [s.l.] : Computer Press, 2008. 384 s. ISBN 978-80-251-2083-5

Přílohy

Příloha č.1: Zdrojový kód vzkazníku č.1

```
<?php
Header("Expires:".GMMDate("D, d M Y H:i:s")." GMT");
$datum = Date("Y-m-j H:i:s");
$jmeno = @$_POST["jmeno"];
$text = nl2br(htmlspecialchars(@$_POST["text"]));
$cislo_obrazku = @$_POST["cislo_obrazku"];
$cas = @$_POST["cas"];
$ip_adresa=$_SERVER['REMOTE_ADDR'];
$prohlizec=$_SERVER['HTTP_USER_AGENT'];

$kravy = array (3,4,6,10,13,19,20,21,24,25,28,32);
$kone = array
(1,2,5,7,8,9,11,12,14,15,16,17,18,22,23,26,27,29,30,31,33,34);

$krava = false;
if (in_array($cislo_obrazku, $kravy)) $krava=true;

$spojeni = MYSQL_Connect("hostname", "username", "password");
mysql_select_db("databasename");
mysql_query("SET NAMES 'cp1250'") or die('Could not set names');

if ($text){
    if ($krava)
        {$krava_nebo_kun=$cislo_obrazku." krava";}
    else {$krava_nebo_kun=$cislo_obrazku." kun";}
    if ($cas+5 >= time()) $krava_nebo_kun="robot";
}
else {$krava_nebo_kun="";}

$cas_zaznam = time()-$cas;
if ($cas=="") {$cas_zaznam="";}

```

```

mysql_query ("insert into zaznam values
    ('$datum', '$ip_adresa', '$prohlizec', '$jmeno', '$text',
    '$cas_zaznam', '$krava_nebo_kun'"));

if ($text){
    $ochrana = mysql_query ("select text from vzkaznik order by datum
        desc");
    $ochrana_z = mysql_fetch_array($ochrana);
    if ($text != $ochrana_z["text"]){
        if ($cas+5 <= time()){
            if ($krava){
                mysql_query ("insert into vzkaznik values
                    ('$datum', '$jmeno', '$text'"));
            }
            else {kun();}
        }
        else {rychlik();}
    }
}

$a = array ();
$obr = array ();
$c = 0;
$d = array ();
do {
    $d = array_merge($obr, $a);
    $obr = array_unique($d);
    $c = count($obr);
    $a = array_fill(0, 8-$c, $kone[rand(0,21)]);
}
while ($c!=7);

$obr[7]=$kravy[rand(0,11)];
shuffle($obr);
?>

```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
      charset=Windows-1250" />

    <style type="text/css">
      .zarovnani{position:relative; left:100px}
      body {background-color: dodgerblue}
      .font {text-align:center; font-weight:bold; color:white;
        font-size:40px}
    </style>

    <script language="javascript">
      function obrazek(cisloobr)
        {document.form.cislo_obrazku.value=cisloobr;
          document.form.submit();}
    </script>

  </head>
  <body>

    <?php function kun()
    {echo ("Kliknul/a jste na nesprávný obrázek!");}
    function rychlik()
    {echo ("Příspěvek nebyl vložen do diskuse z důvodu aktivace
    protispamové ochrany! Odešlete příspěvek min. 5 sekund po načtení
    stránky.");}
    ?>

    <div class="font">Zde můžete zanechat Vaše vzkazy a
    připomínky</div><br>
    <center><table border="0" cellpadding="10" >
      <tr>
        <td>

```

```

<form action="http://www.vzkaznik2.wz.cz/vzkaznik.php" name="form"
method="post">
<b>Jméno:</b><input type="text" name="jmeno" value=""
style="position:relative; left:50px"><br><br>
<b>Vzkaz:</b><textarea name="text" cols="50" rows="5"
style="position:relative; left:54px"></textarea
wrap="soft"><br><br>
<div class="zarovnani">Pro odeslání klikněte na <b>KRÁVU</b>
</div><br>
<input type="hidden" name="cislo_obrazku" value="">
<input type="hidden" name="cas" value="<?php echo Time(); ?>">
</form>

```

```

<?php
echo"<div class='zarovnani'>
<img src='".\$obr[1]".jpg' onClick='obrazek(".\$obr[1].") '>
<img src='".\$obr[2]".jpg' onClick='obrazek(".\$obr[2].") '>
<img src='".\$obr[3]".jpg' onClick='obrazek(".\$obr[3].") '>
<img src='".\$obr[4]".jpg' onClick='obrazek(".\$obr[4].") '>
</div>
<div class='zarovnani'>
<img src='".\$obr[5]".jpg' onClick='obrazek(".\$obr[5].") '>
<img src='".\$obr[6]".jpg' onClick='obrazek(".\$obr[6].") '>
<img src='".\$obr[7]".jpg' onClick='obrazek(".\$obr[7].") '>
<img src='".\$obr[0]".jpg' onClick='obrazek(".\$obr[0].") '>
</div>"
?>

```

```

<br><table align="left" border="0" width="550">

```

```

<?php
\$vsechno=mysql_query ("select jmeno, datum, text from vzkaznik order
by datum desc");

```



```

while ($zaznam=mysql_fetch_array($vsechno))
{echo "<tr>
      <td><b>". $zaznam["jmeno"] . "</b></td>
      <td align='right'><small>". $zaznam["datum"] . "</small>
      </td>
</tr>
<tr>
      <td colspan='2' width='800' style='word-wrap:
      break-word'>". $zaznam["text"] . "<hr width='100%'></td>
</tr>";
}
mysql_Close($spojeni);
?>

</table>
      </td></tr>
</table>
</center>
</body>
</html>

```

Příloha č.2: Zdrojový kód vzkazníku č.2

```
<?php
Header("Expires:".GMDate("D, d M Y H:i:s")." GMT");
$datum = Date("Y-m-j H:i:s");
$jmeno = @$_POST["jmeno"];
$text = nl2br(htmlspecialchars(@$_POST["text"]));
$cislo_obrazku = @$_POST["cislo_obrazku"];
$cas = @$_POST["cas"];
$misto = @$_POST ["prirad"];
$pozice = @$_POST ["pozice"];
$ip_adresa=$_SERVER['REMOTE_ADDR'];
$prohlizec=$_SERVER['HTTP_USER_AGENT'];

$kravy = array (3,4,6,10,13,19,20,21,24,25,28,32);
$kone = array
(1,2,5,7,8,9,11,12,14,15,16,17,18,22,23,26,27,29,30,31,33,34);

$krava = false;
$pozice_zaznam = "";
if ($misto){
    $pozice_zaznam = $pozice." kun";
    if ($misto==$pozice){
        $krava = true;
        $pozice_zaznam = $pozice." krava";
    }
}

if ($cas+5 >= time()) $pozice_zaznam = "robot";
$cas_zaznam = time()- $cas;
if ($cas=="") {$cas_zaznam="";}

$spojeni = MYSQL_Connect("hostname", "username", "password");
mysql_select_db("databasename");
mysql_query("SET NAMES 'cp1250'") or die('Could not set names');
```

```

mysql_query ("insert into zaznam values
    ('$datum', '$ip_adresa', '$prohlizec', '$jmeno', '$text',
    '$cas_zaznam', '$pozice_zaznam')");

if ($text){
    $ochrana = mysql_query ("select (text) from vzkaznik order by
        datum desc limit 1");
    $ochrana_z = mysql_fetch_array($ochrana);
    if ($text != $ochrana_z["text"]){
        if ($cas+5 <= time()){
            if ($krava){
                mysql_query ("insert into vzkaznik values
                    ('$datum', '$jmeno', '$text')");
            }
            else {kun();}
        }
        else {rychlik();}
    }
}

$a = array ();
$obr = array ();
$c = 0;
$d = array ();
do {
    $d = array_merge($obr, $a);
    $obr = array_unique($d);
    $c = count($obr);
    $a = array_fill(0, 8-$c, $kone[rand(0,21)]);
}
while ($c!=7);

$obr[7]=$kravy[rand(0,11)];
shuffle($obr);

```

```

$pozice=0;
for ($t=0; $t<=7; $t++) {
    for ($x=0; $x<=11; $x++) {
        if ($kravy[$x]==$obr[$t]) {$pozice=$t;}
    }
}

$souradnice = array
("a1"=>0,"b1"=>1,"c1"=>2,"d1"=>3,"a2"=>4,"b2"=>5,"c2"=>6,"d2"=>7);
$prirad = array_search ($pozice, $souradnice);
?>

```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 TRANSITIONAL//EN">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;
      charset=Windows-1250" />

    <style type="text/css">
      .zarovnani{position:relative; left:100px}
      .zarovnani2 {position:relative; left:102px}
      body {background-color: dodgerblue}
      .stred {text-align:center}
      .font {text-align:center; font-weight:bold; color:white;
        font-size:40px}
    </style>

    <script language="javascript">
      function obrazek(cisloobr)
      {document.form.cislo_obrazku.value = cisloobr;
        document.form.submit();}
    </script>

  </head>
  <body>

```

```

<?php function kun()
{echo ("Kliknul/a jste na nesprávný obrázek!");}
function rychlik()
{echo ("Příspěvek nebyl vložen do diskuse z důvodu aktivace
protispamové ochrany! Odešlete příspěvek min. 5 sekund po načtení
stránky.");}
?>

```

```

<div class="font">Zde můžete zanechat Vaše vzkazy a
připomínky</div><br>

```

```

<center><table border="0" cellpadding="10" >
  <tr>
    <td>

```

```

<form action="http://www.vzkaznikkh.wz.cz/vzkaznik.php" name="form"
method="post">

```

```

<b>Jméno:</b><input type="text" name="jmeno" value=""
  style="position:relative; left:50px"><br><br>

```

```

<b>Vzkaz:</b><textarea name="text" cols="50" rows="5"
  style="position:relative; left:54px"></textarea
  wrap="soft"> <br><br>

```

```

<b>Pozice:</b><input type="text" name="pozice" value=""
  style="position:relative; left:52px">

```

```

<input type="submit" name="odeslat" value="Odeslat"
  style="position:relative; left:60px"><br><br>

```

```

<div class="zarovnani2">Pro odeslání příspěvku napište pozici,
na které se nachází <b>KRÁVA!</b></div>

```

```

<div class="zarovnani2"><b>Vždy začínajte nejprve písmenem a
následně číslicí.</b></div>

```

```

<div class="zarovnani2">Například: a1, b2, c3...</div>

```

```

<input type="hidden" name="prirad" value="<?php echo $prirad;?>">

```

```

<input type="hidden" name="cas" value="<?php echo Time(); ?>">

```

```

</form>

```

```

<?php
    echo"<div class='zarovnani'><table border=0>
    <tr>
        <td></td>
        <td class='stred'><b>a</b></td>
        <td class='stred'><b>b</b></td>
        <td class='stred'><b>c</b></td>
        <td class='stred'><b>d</b></td>
    </tr>
    <tr>
        <td><b>1</b></td>
        <td><img src='\".$obr[0]\".jpg'></td>
        <td><img src='\".$obr[1]\".jpg'></td>
        <td><img src='\".$obr[2]\".jpg'></td>
        <td><img src='\".$obr[3]\".jpg'></td>
    </tr>
    <tr>
        <td><b>2</b></td>
        <td><img src='\".$obr[4]\".jpg'></td>
        <td><img src='\".$obr[5]\".jpg'></td>
        <td><img src='\".$obr[6]\".jpg'></td>
        <td><img src='\".$obr[7]\".jpg'></td>
    </tr>
</table></div>"
?>

```

```

<br><table align="left" border="0" width="550">

```

```

<?php
    $vsechno=mysql_query ("select jmeno, datum, text from vzkaznik order
    by datum desc");

```

```

while ($zaznam=mysql_fetch_array($vsechno))
    {echo "<tr>
        <td><b>". $zaznam["jmeno"] . "</b></td>
        <td align='right'><small>". $zaznam["datum"] . "
            </small></td>
    </tr>
    <tr>
        <td colspan='2' width='800' style='word-wrap: break-
            word'>". $zaznam["text"] . "<hr width='100%'></td>
    </tr>";
    }
mysql_Close($spojeni);
?>

</table>
    </td></tr>
</table>
</center>
</body>
</html>

```