Zabezpečení a optimalizace webových stránek v internetu

Security and optimalization web sites on the Internet

Bc. Martin Jurák

Diplomová práce
2010

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2009/2010

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Martin JURÁK**
Studijní program:	**N 3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Téma práce:	**Zabezpečení a optimalizace webových stránek v internetu.**

Zásady pro vypracování:

1. **V teoretické části se věnujte problematice týkající se bezpečnosti dat, přístupnosti webových stránek a jejich optimalizace pro vyhledávače.**
2. **V praktické části prezentujte popsané technologie na jednoduchém portálu pro zveřejňování inzerce.**
3. **Popište vytvořené části aplikace a konkrétní způsob zabezpečení a aplikaci SEO na těchto stránkách.**
4. **V optimalizaci pro vyhledávače sledujte průběžné výsledky po dobu alespoň jednoho měsíce pro nejvýznamnější vyhledávače, výsledky vyhodnoťte a navrhněte zlepšení.**

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **Castro E. HTML, XHTML a CSS Názorný průvodce tvorbou WWW stránek. Computer Press 2007, ISBN: 978-80-251-1531-2**
2. **Naramore E., Gerner J., Stolz J., Glass M. K. Vytváříme webové aplikace v PHP5, MySQL a Apache. Computer Press 2006, ISBN: 80-251-1073-7**
3. **Huseby S. Zranitelný kód, Computer Press 2006, ISBN: 80-251-1180-6**
4. **Kubíček M. Velký průvodce SEO, Computer Press 2008, ISBN: 987-80-251-2195-5**
5. **Lavin P. PHP-objektově orientované, Grada 2009, ISBN: 978-80-247-2137-8**
6. **Resig J. JavaScript a Ajax, Computer Press 2007, ISBN: 978-80-251-1824-5**
7. **Sirovich J. SEO v PHP, Computer Press 2008, ISBN: 978-80-251-2083-5**

Vedoucí diplomové práce: **doc. Ing. Martin Sysel, Ph.D.**
Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce: **19. února 2010**

Termín odevzdání diplomové práce: **8. června 2010**

Ve Zlíně dne 19. února 2010

prof. Ing. Vladimír Vašek, CSc.
děkan

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Tato práce pojednává o problémech, týkajících se bezpečnosti internetových aplikací a jejich optimalizace pro vyhledávače. Obsahuje teoretickou část, která popisuje v současnosti převažující bezpečnostní hrozby a dále vystvětluje pojmy týkající se přístupnosti a optimalizace internetových stránek. Praktická část se věnuje popisu částí vytvořeného portálu pro zveřejňování inzerce a aplikace konkrétních kroků pro zabezpečení a optimalizaci stránek. Tato část také zahrnuje výsledky měření statistik návštěvnosti, jejich následné vyhodnocení a návrh řešení dalšího postupu při optimalizaci.

Klíčová slova:

www, html, php, mysql, cross-site scripting, xss, sql injection, directory traversal, cross-site request forgery, csrf, seo

ABSTRACT

This thesis disserts on the problems concerning security of Internet applications and their search engine optimization. It contains theoretical part describing present prevalent security issues and explains the concepts of accessibility and web site optimization. Practical part of the thesis deals with the description of the parts created by the portal for publishing advertisement and application of concrete steps to safeguard and optimize the Site. This part also includes the results of the measurement statistics, their subsequent evaluation and design solutions to further progress in optimizing.

Keywords:

www, html, php, mysql, cross-site scripting, xss, sql injection, directory traversal, cross-site request forgery, csrf, seo

Na tomto místě bych rád poděkoval vedoucímu diplomové práce, kterým je doc. Ing. Martin Sysel, Ph.D., za jeho podnětné připomínky a konzultace při psaní této práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně …….

podpis diplomanta

OBSAH

ÚVOD

Internet zaznamenal v posledních letech významný rozmach i technologický pokrok. Změnami prochází internet jako celek, ale největší růst zaznamenala především služba World Wide Web, což není nic jiného než síť vzájemně propojených webových stránek.

Webové stránky jsou významným zdrojem informací, ale také prostředkem hojně využívaným k propagaci firem a obchodu. Tyto činnosti často vyžadují zpětnou odezvu uživatele, a tím ruku v ruce určitý typ dynamičnosti stránek. S tím narůstá náchylnost těchto aplikací k různým útokům a pokusům o jejich narušení. Cíle útoků mohou být různé, od méně nebezpečných zásahů do vzhledu stránek pomocí úprav jejich CSS stylů, přes krádeže osobních dat až po útoky zaměřené na destrukci webu.

Mezi nejznámnější útoky patří Cross-site scripting (XSS), SQL Injection, Cross-Site Request Forgery (CSRF) a Directory traversal (dot dot slash). Většina z nich využívá nedostatečného zabezpečení vstupů do aplikace, Cross-Site Request Forgery navíc využívá technik tzv. sociálního inženýrství.

Se zabezpečením úzce souvisí základní zásady používání hesel, problematika tvorby, šifrování a uchovávání hesel ať už z hlediska vývojáře tak i uživatele samotného.

Mnohem významější může být z hlediska „poslání“ webových stránek optimalizace pro vyhledávače neboli SEO. Tento obor zahrnuje všechny kroky vedoucí k nejvyšším pozicím ve vyhledávačích což by mělo být prioritou pro majitele a vývojáře. Od přípravy samotných stránek, fází tzv. on-page optimalizace, přes pokročilejší optimalizaci off-page, neboli optimalizací okolí a okolních vlivů. Pro vývojáře, kteří se tímto zabývají, je k dispozici nepřeberné množství on-line nástrojů. Mezi hojně používané patří služba Google Analytics určená na hloubkovou analýzu stránek z hlediska návštěvnosti, struktury návštěv, zdrojů návštěvnosti a dalších. Další užitečný nástroj je online služba Seoservis.cz. Tato slouží hlavně k okamžité analýze z hlediska on-page faktorů (validita a sémantika kódu, analýza klíčových slov, výpis pozic ve vyhledávačích pro daná klíčová slova a celková oblíbenost stránek) i off-page faktorů.

I. TEORETICKÁ ČÁST

1 POUŽITÉ TECHNOLOGIE A SOFTWARE

1.1 HTML

Všechny webové stránky jsou napsány pomocí nějaké formy jazyka HTML. HTML umožňuje formátovat text, přidávat grafiku, zvuk a video, a to vše uložit do textového souboru, který dokáže přečíst jakýkoli počítač.

HTML stránky jsou po internetu přenášeny pomocí protokol HTTP (HyperText TransferProtocol). Vývoj jazyka HTML řídí konsorcium W3C (World Wide Web Consortium).

1.2 CSS

Kaskádové styly neboli CSS (z anglického Cascading Style Sheets) jsou novým systémem formátování HTML a nahradily již zmíněné zavržené formátovací elementy.

Původní specifikace pro kaskádové styly byla omezená pouze ke tvorbě efektů jazyka HTML. Druhá verze CSS, vydaná v roce 1998, a v průběhu zavádění lehce aktualizovaná do podoby verze 2.1, přinesla nové možnosti, například možnost mnohem preciznějšího umístění elementů na webové stránce. CSS tak dokázaly, že pomocí nich lze nejen přetvořit formátováni jazyka HTML, ale lze s nimi též vytvořit profesionálně vypadající rozvržení (layout) stránek.

1.3 PHP

PHP (*PHP: Hypertext Preprocessor*, původně *Personal Home Page*) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML či WML, což je velmi výhodné pro tvorbu webových aplikací. PHP skripty jsou prováděny na straně serveru, k uživateli je přenášen až výsledek jejich činnosti. PHP obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory, přístup k většině databázových serverů (mj.MySQL, ODBC…), podporu celé řady internetových protokolů (HTTP, SMTP, FTP, POP3, …). Od verze PHP 5 již podporuje objektově orientované programování (OOP). Aktuální verze je PHP 5.3.2.

1.4 MySQL

MySQL je databázový systém, jenž umožňuje technologii PHP spolupracovat na zpřístupnění a zobrazení dat ve formátu čitelném v internetových prohlížečích. Je to server zpracovávající dotazy ve strukturovaném dotazovacím jazyce (Structured Query Language - SQL) navržený pro zpracování velkého množství velmi složitých dotazů. Jde o relační databázový systém, MySQL tedy umožňuje spojování mnoha různých tabulek. Díky tomu nabízí maximální efektivitu a rychlost.

1.5 Javascript

JavaScript je klientský skript. To znamená, že se program odesílá se stránkou na klienta (do prohlížeče) a teprve tam je vykonáván. (Protikladem klientských skriptů jsou skripty serverové, které jsou vykonávány na serveru a na klienta jdou už jen výsledky.)

JavaScript je interpretovaný, multiplatformní programovací jazyk se základními objektově orientovanými schopnostmi. Univerzální jádro jazyka bylo vloženo do webových prohlížečů a rozšířeno přidáním objektů reprezentující okno prohlížeče a jeho obsah. Tato *klientská* verze JavaScriptu umožňuje vložit do webových stránek proveditelný obsah. Stránky se tak stávají dynamické – mohou obsahovat nejrůznější programy, které komunikují s uživatelem, řídí prohlížeč, či dynamicky vytváří obsah HTML. Při práci skriptu není třeba kontaktovat server, veškerou práci skriptu zajišťuje sám prohlížeč.

Jádro jazyka syntakticky připomíná C++ a Javu. Avšak syntaxí podobnost končí. JavaScript je jazyk bez typové kontroly, což znamená, že proměnné nemusí mít specifikovaný typ. A navíc JavaScript je čistě interpretovaný jazyk, na rozdíl například od kompilovaných C a C++ a na rozdíl od Javy, která je před interpretací kompilována do bajtového kódu.

Co se týče bezpečnosti v JavaScriptu. JavaScript na straně klienta neumožňuje čtení a zapisování souborů, z důvodů které jsou zřejmé. Nebyl by pak problém jednoduchým prográmkem naprosto znehodnotit obsah pevného disku. Rovněž nepodporuje práci se sítí, až na jednu důležitou vyjímku: může donutit webovský prohlížeč k načtení libovolného URL.

1.6 WampServer 2.0

WampServer je webové vývojové prostředí pro Windows. Umožňuje uživateli vytvářet webové aplikace s pomocí Apache, PHP a MySQL databázemi. Obsahuje také phpMyAdmin a SQLiteManager pro snadnou správu uživatelských databází.

WampServer se instaluje automaticky a jeho použití je velmi jednoduché a intuitivní. Uživatel je schopen nastavovat server bez nutnosti přímo editovat systémové soubory a soubory nastavení. Tento balík řešení umožňuje úpravu nebo obnovu serveru, bez nutnosti jeho opětovné instalace. V případě, že je jednou nainstalován, není problém přidávat či aktualizovat na nově vydané Apache, MySQL a PHP verze.

Po úspěšném zavedení do systému je k dispozici ikona WampServeru, pro snadnou navigaci, nastavení serveru a systémová nastavení umístěná v hlavním panelu.

Obr. 1 – Menu WampServeru

Při kliknutí na uvedenou ikonu se rozbalí menu s následujícími položkami: localhost – úvodní strana WampServeru, phpMyAdmin a SQLiteManager pro správu uživatelských databází, dále je zde odkaz přímo do hlavního adresáře WampServeru. Další položky slouží k nastavení jednotlivých částí a služeb serveru.

1.7 Internetové prohlížeče

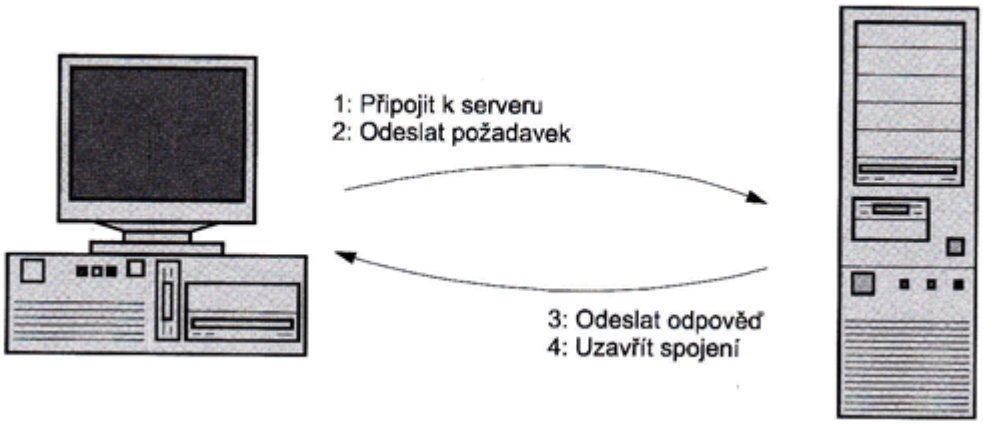
Pro vývoj a testování byly použity tyto internetové prohlížeče a jejich doplňky:

- Mozilla Firefox 3.6.3 (vývoj)
    - Web Developer Toolbar 1.1.8
    - Firebug 1.5.3
- Internet Explorer 8
- Google Chrome 4.1.249.1059
- Apple Safari 4.0.5
- Opera 10.51
- IETester v0.4.3 (testování starších verzí prohlížeče IE)

2 BEZPEČNOST DAT NA WEBOVÝCH STRÁNKÁCH

2.1 Aplikace typu Klient - Server

Když chce webový prohlížeč zobrazit webovou stránku, připojí se k serveru uvedenému v URL, od kterého se pokouší získat obsah stránky. Jakmile se ustaví spojení TCP, zašle prohlížeč požadavek HTTP, ve kterém požádá webový server o hledaný dokument. Webový server zašle odpověď, která obsahuje obsah stránky a uzavře spojení. Iniciátorem spojení je vždycky prohlížeč, server nikdy nezavolá zpátky. To znamená, že HTTP je protokol typu klient/server. Klientem je většinou webový prohlížeč, ale nemusí to tak být vždy. Postačí jakýkoliv program který umí protokolem HTTP odesílat požadavky webovému serveru.



Obr. 2 – Aplikace typu Klient-Server (převzato[3])

Pokud se použije tzv. *persistentní spojení,* může spojení zůstat nějaký čas (většinou krátký) otevřené, což při více požadavcích umožní snížit režii TCP. Persistentní spojení urychlí například zobrazení stránek, které obsahují větší množství obrázků. Pokud dokument obsahuje hypertext odkazující na vložený obsah, jako jsou například obrázky nebo Java-aplety, musí prohlížeč kvůli zobrazení dokumentu odeslat více požadavků.

2.2 Cíle útoků

Otázka zabezpečení webových aplikací je bezesporu nezanedbatelnou položkou, které musí být při vývoji zejména programové a kontrolní části webu věnován dostatečný prostor. Stále totiž existují principielně jednoduché metody, jak narušit aplikační bezpečnost, a ve většině případů k tomu stačí přístup na úrovni běžného uživatele webu.

Problém je však často (zejména začínajícími vývojáři) ignorován, což útočníkům značně zjednodušuje situaci.

2.2.1 Krádež ID relace

Cílem útoků vedených za účelem získání citlivých dat je ve většině případů session ID. Systém sessions (relací) je jednou z cest, jak kontrolovat pohyb uživatele na webu (neboť protokol http je standardně bezstavový). Princip sessions je jednoduchý – každý přistoupivší uživatel obdrží jednoznačný identifikátor, tzv. **session ID (SID).** Každá session může na serveru adresovat téměř neomezený prostor vlastních proměnných, je tedy možné jednoduše ukládat na serveru stavové informace. Jediným přístupovým údajem k těmto informacím je SID. Manipulace se SID je tedy pochopitelně základem většiny útoků.

Session ID se totiž musí uložit nejen na serveru, pamatovat si ji musí i prohlížeč (klient). Ukládání tohoto parametru se provádí buď pomocí systému cookies, nebo (nejsou-li cookies k dispozici) pomocí URL – což je ale z bezpečnostního hlediska velmi nešťastné řešení, neboť jde o profilovaný a vysoce používaný parametr, který se může ukládat na více místech (historie prohlížeče, cache, statistiky, HTTP hlavička apod.). Útočníkovi pak stačí dostat se k takové informaci - k tomu slouží také např. metody spojené s přesvědčováním neznalého uživatele (klamavé reklamy apod.). Jednou z procedur, která může bránit získání informací pomocí SID, je její životnost, která se definuje podobně jako u cookies a po uplynutí dané doby ukončuje platnost session. Tím se dá do jisté míry zabránit zpracování získaných informací vypršením určité lhůty (pokud tedy není automatizované - potom doba zpracování není klíčový aspekt).

2.2.2 Poškození aplikace

Velmi specifickým cílem je poškozování systému. V první řadě je třeba rozlišit, zda byl útok cílený nebo vznikl nevědomostí (omylem) uživatele a chybou aplikace. Cílené útoky většinou spočívají ve vkládání nebezpečných klientských skriptů, které spustí nějakou operaci. Ta může být relativně nedestruktivního charakteru (úpravy vzhledu, poškození za účelem chybné validace kódu), nebo vyloženě destruktivní (smazání dat, nevratné poškození aplikace).

2.3 Typy útoků

Nejrozšířenější útoky by se daly rozdělit na dvě základní skupiny. A to podle toho, co je předmětem jejich útoku. V prvním případě využívají přesunu dat od uživatele do subsystému (databáze). Druhý případ využívá nedostatečně ošetřeného výstupu na web. V následujících podkapitolách jsou tyto případy rozebrány.

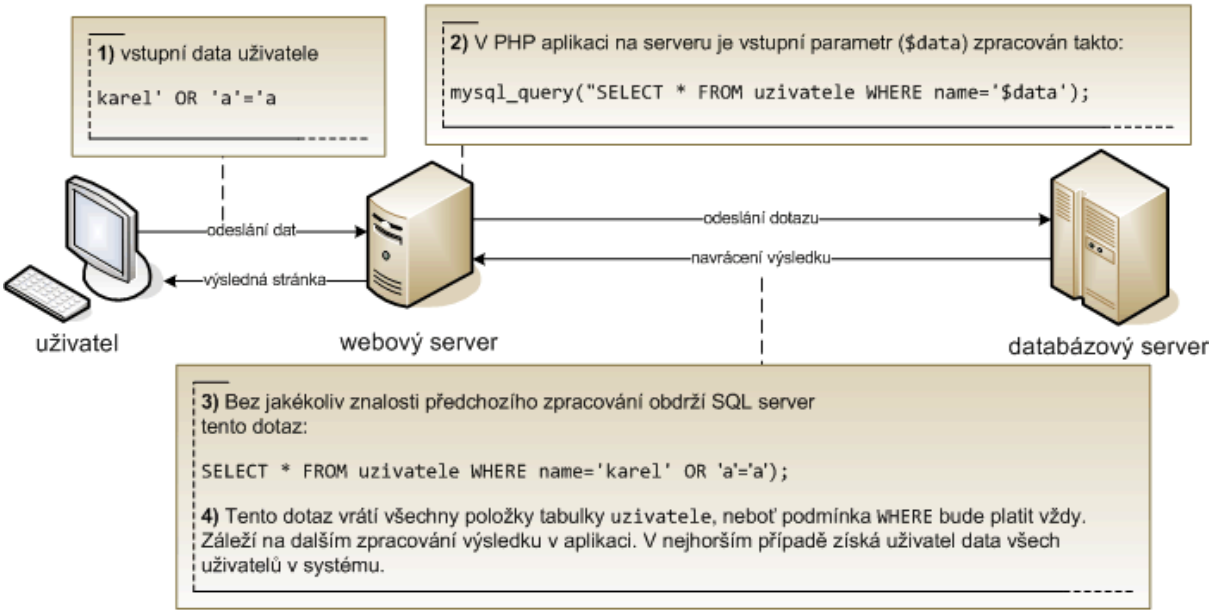
2.3.1 SQL Injection

Moderní web je založen na modelu relačních databází a přístupovém systému klient-server. Protože pro manipulaci s daty existuje jazyk (ve většině případů na bázi SQL), může se projevit snaha o změnu dotazu, který klademe databázi. Tímto způsobem pak můžeme docílit získání citlivých dat, v případě nejhoršího scénáře i přístup k databázovému serveru a jeho ovládnutí.

Pomocí SQL Injection je útočník schopen měnit nebo zadávat dotazy, které se odesílají do databáze prostřednictvím vstupů do webové aplikace. Samotný útok začíná v okamžiku, kdy program vytváří dotazy založené na řetězcích pocházejících od klienta a když je následně posílá na databázový server, aniž by ošetřil znaky, které server považuje za speciální.

2.3.1.1 Příklad útoku

Pokud klademe databázi dotaz, který je ovlivněn vstupními parametry, nejčastěji tyto parametry reflektujeme jako hodnoty (ať už při ukládání, nebo čtení). Hodnoty se v SQL jazyku ohraničují pomocí znaku `'`. Pokud se v tomto bodě nefiltruje vkládaná hodnota na obsah tohoto znaku a neošetří se pomocí patřičné escape sekvence – v tomto případě `\'`, SQL jazyk pochopí výskyt tohoto řetězce jako konec hodnoty a dál pracuje se zbývajícími daty jako s pokračováním dotazu. Tím se tedy z úrovně hodnoty dostáváme na úroveň vyšší, ve které můžeme ovlivnit tvar celého příkazu. Typický příklad zneužití tohoto faktu je znázorněn na (Obr.3).



Obr. 3 – Příklad SQL Injection útoku(převzato [10])

Situace se může znatelně zkomplikovat, pokud útočník použije při formulaci dotazu klíčové slovo `UNION`, které spojuje více dotazů typu SELECT. Nejvážnější problém ale tento útok představuje, pokud je veden proti databázovým systémům jako SQLite nebo postgreSQL, které umožňují odeslat více SQL příkazů v jednom dotazu na databázi. Pak je možné pomocí několika příkazů zcela získat kontrolu nad databázovým systémem.

Chybná manipulace s datovými typy představuje velmi podobné riziko. Využívá faktu, že číselné hodnoty v SQL nemusíme ohraničovat pomocí znaků `'`. Potom můžeme místo číselné hodnoty podstrčit řetězec a situace je úplně stejná, jako v předchozím případě. Navíc zde nemusíme ošetřovat správné ukončení escape sekvencí a tím zajistit korektní syntaxi dotazu.

2.3.2 Cross-site Scripting

Pro útoky, které jsou založeny na vložení nebezpečného skriptu do obsahu webové stránky se používá termín *Cross Site Scripting (XSS)*. Spočívají v záměrném vložení nebezpečného kódu (který je napsán v klientském skriptovacím jazyku – např. Javascript, JScript, VBScript) do „bezpečného“ obsahu webové stránky. Tyto techniky často spoléhají i na sociální inženýrství (protože k úspěšně provedenému útoku je třeba vykonat skript v

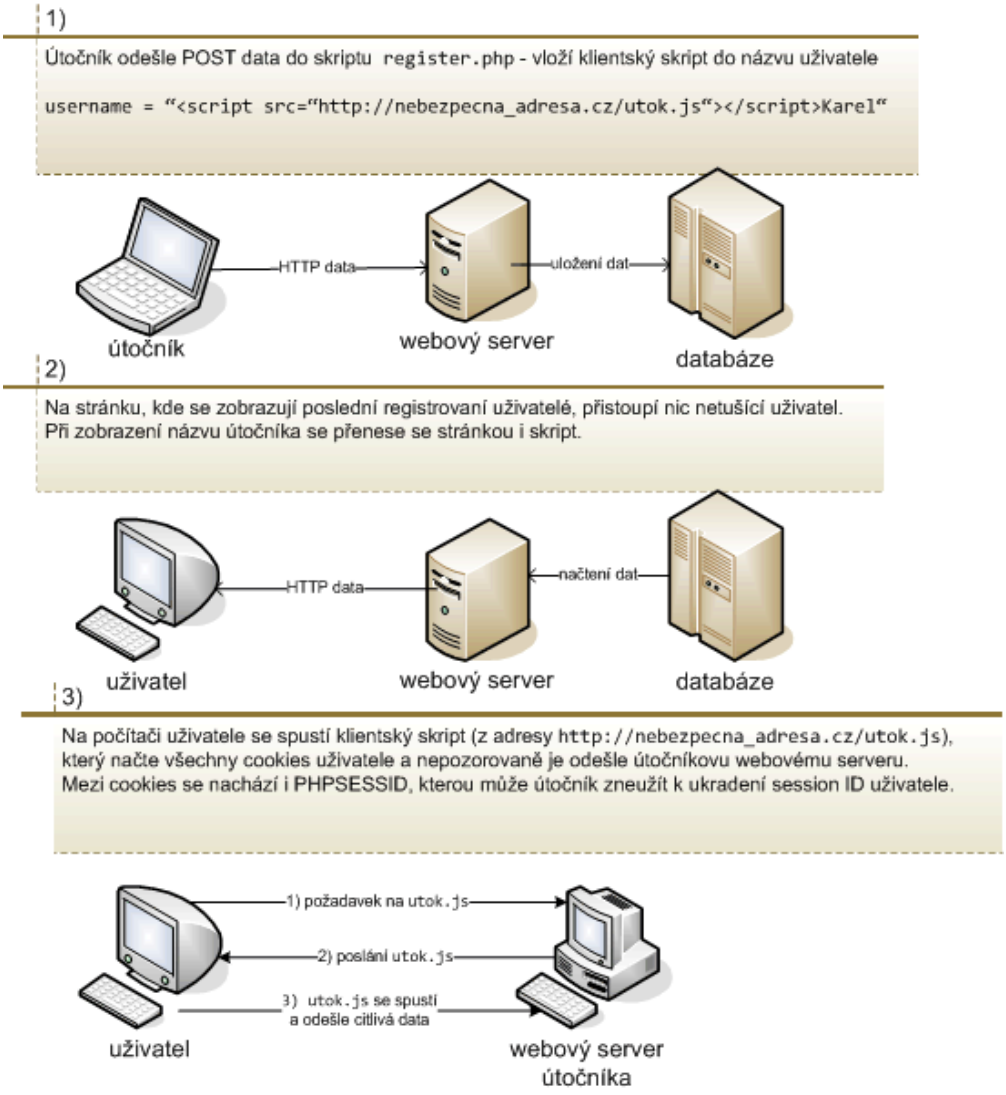
prohlížeči napadeného uživatele, a někdy je proto nutné donutit jej ke spolupráci). Rozeznáváme v základě tři typy těchto útoků: okamžitý, perzistentní a lokální.

2.3.2.1 Okamžitý útok

Okamžitý útok (v originále non-persistent nebo reflected) je založen na okamžitém zobrazení (vykonání) nebezpečného skriptu. V praxi jde o webové aplikace, které zobrazí výsledek okamžitě na základě požadavku, tento však není dále uložen. Na první pohled nejde o podstatný problém, neboť nebezpečný kód zpracuje ve většině případů jen webový prohlížeč útočníka. Pokud však webová aplikace využívá parametry z URL (což s největší pravděpodobností dělá), může útočník přesvědčit existujícího uživatele, aby kliknul na odkaz s vloženou nebezpečnou skriptovací konstrukcí, která následně může ukrást údaje, uložené v prohlížeči uživatele. Interní logika prohlížeče toto nebere jako prohřešek, protože skript, který byl vykonán, náleží webové aplikaci a tato má právo k přístupu k těmto informacím.

2.3.2.2 Persistentní útok

Daleko nebezpečnější je tzv. perzistentní útok (popisován termíny persistent, stored). Ten využívá možnosti běžného uživatele vkládat vlastní obsah na web (diskusní fóra, komentáře atp.). Útok potom může těžit z faktu, že špatně zabezpečená aplikace vkládaná data nijak nekontroluje. Potom stačí vložit běžný HTML kód se skriptem, který provádí nebezpečné operace. Ten bude opět posouzen prohlížečem jako bezpečný (neboť náleží webové aplikaci). Rozsah poškození může být daleko větší než u předchozího typu útoku, protože se může týkat všech uživatelů, kteří si danou informaci zobrazí ve svém prohlížeči. Jde o velmi nebezpečný typ útoku, proti kterému by měla vždy existovat alespoň základní ochrana. Typický příklad perzistentního útoku je popsán na (*Obr.4).*



Obr. 4 – Příklad perzistentního útoku (převzato [10])

2.3.2.3 Lokální útok

Poslední typ útoku byl popsán relativně nedávno. Bývá v originále často označován jako DOM-based nebo local. V podstatě je velmi podobný charakteristikám okamžitého útoku, ale ke zpracování nebezpečného kódu se zneužije existujícího klientského skriptu. Hlavní nebezpečí (ze kterého také plyne název útoku) spočívá v lokálních webových aplikacích, protože nejrozšířenější z prohlížečů, Internet Explorer, považuje javascripty spouštěné v lokální zóně za bezpečné a uděluje jim podle toho určitá pravidla pískoviště (*sandbox* - označení prostoru proměnných, funkcí a možností, která má v dané situaci klientský skript k dispozici). Lokální skripty mají např. možnost přistupovat k souborům

na lokálních discích. Z toho plynou evidentní nebezpečí (např. odesílání soukromých dat třetí straně).

Tento typ útoku byl úspěšně použit např. v softwaru Bugzilla, což je typický případ webové aplikace, běžící na lokálním prostředí. V této aplikaci byl javascript použit k vypsání současné URL, pomocí DOM objektu document.location, bez jakékoliv kontroly či filtrování. Toho bylo brzy zneužito ke vložení nebezpečného kódu přímo do webové stránky, a zejména v Internet Exploreru tato chyba mohla mít dalekosáhlé následky [3].

2.3.2.4 Prevence útoku

Nejdůležitějším zabezpečením je nepropouštět do veřejných částí webu vůbec žádné skriptovací konstrukce. Protože jazyk HTML je schopný zpracovávat kód klientských skriptů i jako reakci na události jednotlivých značek (např. onchange, onclick) nebo může měnit vzhled prvku z hlediska kaskádových stylů (parametr style), jedno z nejlepších řešení je překódovat HTML ohraničovací značky < a > jako entity. Tímto způsobem docílíme zobrazení zdrojového kódu přímo v textu, aniž by ho prohlížeč chápal jako značky.

Řešení má pochopitelně svoje nevýhody. Někdy se můžeme dostat do situace, kdy je pro nás žádoucí povolit uživatelům vkládat alespoň nějaké HTML značky (např. obrázky). Pak je nutné toto ošetřit např. pomocí regulárních výrazů a úplně odstranit některé parametry značek. Úplně nejlepším řešením je použít vlastní systém značek, které převedeme v poslední etapě zobrazení na patřičné HTML tagy. Takovým systémem je např. BBCodes (Bulletin Board Code), velmi známé zejména z diskuzních fór – implementuje je např. systém phpBB [8] a mnohé další.

V PHP využijeme při ošetřování funkci `htmlspecialchars()`, která převede všechny HTML značky a speciální znaky na entity. Mírnou nevýhodou této funkce je, že navýší objem výsledného přenášeného textu. Výhodou je téměř absolutní bezpečnost – veškerý kód se zobrazí jako součást textu. Druhou možností je použít ořezávací funkci `strip_tags()` – ta odstraní veškeré HTML značky. Nepřevádí však entity, což např. u znaku & (ampérsand), vyskytuje-li se volně v textu, způsobí znevalidnění celé stránky – z čehož mohou plynout problémy (např. při použití striktního formátu XHTML).

`htmlspecialchars()`se doporučuje používat všude tam, kde je očekáván neformátovaný text (příp. text formátovaný jinak, než pomocí HTML) [10].

2.3.3 Cross-Site Request Forgery

Cross-site Request Forgery (CSRF) je jedna z metod útoku do internetových aplikací (typicky implementovaných skriptovacími jazyky nebo cgi) pracující na bázi neočekávaného resp. nezamýšleného požadavku pro vykonání určité akce v této aplikaci, který ovšem pochází z nelegitimního zdroje. Většinou se nejedná o útok směřující k získání přístupu do aplikace (i když i pro to může být zneužit); spíše využívá (zneužívá) akce uživatelů, kteří jsou k ní již v okamžiku útoku přihlášeni [6].

2.3.3.1 Příklad útoku

Existuje nějaká netriviální internetová aplikace (typu wiki, blog, diskuzní fórum, e-shop, redakční systém, …), která má svou administrační část, přístupnou pouze pro administrátory, a u které útočník zná (nebo dokáže odhadnout) URL adresy (popřípadě i posílané proměnné) pro spuštění akcí určených na změnu (editaci obsahu, smazání, …) jejích objektů (příspěvků na blogu či fóru, článků v redakčním systému či wiki, apod…).

Útočník současně zná (je v kontaktu, nebo dokáže oslovit a přesvědčit) jiného uživatele, který se do této aplikace již přihlásil a operuje v ní s administrátorskými právy. Útočník poté (většinou s využitím tzv. sociálního inženýrství) přiměje tohoto administrátora, aby zobrazil (jím předem připravenou) maligní internetovou stránku, která provede samotný CSRF útok.

Ten spočívá v tom, že součástí této maligní stránky je vyslání požadavku na adresu (popř. kombinaci adresy a proměnných) do zmíněné internetové aplikace, který způsobí změnu určitých záznamů či objektů, které spravuje. Tento požadavek (HTTP Request) může být realizován (pro HTTP metodu GET) přímo v HTML, pomocí značky, u které se specifikuje zdroj (obrázek, rám stránky, …, navíc často pomocí stylů nebo atributů skryté nebo minimalizované, aby si jich původce požadavku nevšimnul), nebo (pro metodu POST) sestavením požadavku ve skriptovacím jazyce při zpracování stránky.

Útok je úspěšný, pokud v okamžiku požadavku na tuto stránku je uživatel, který maligní stránku spustil, do aplikace platně přihlášen a tato aplikace není proti tomuto typu

útoku zabezpečena. Skrytý požadavek na editaci nebo smazání objektů v inkriminované aplikaci se tak vykoná, protože aplikace není schopna odlišit, z jakého podnětu požadavek přišel (zda-li z její vlastní administrační stránky nebo právě z CSRF útoku) – tento útok tedy patří do skupiny tzv. „problému zmateného zástupce“ (en:Confused deputy problem), která je charakteristická tím, že strůjcem maligní akce je nikoli útočník, ale legitimně přihlášený uživatel. Změny se projeví, aniž by to tušil a mnohdy zůstanou dlouho nezjištěny.

Útočník nemusí znát, který záznam chce takto nechat nic netušícím administrátorem smazat nebo změnit, přesto v nechráněné aplikaci je schopen způsobit často neopravitelné škody. Méně často se může pokusit nechat spustit požadavek, který žádný objekt nemění, a místo toho zobrazí pro útočníka zajímavé nebo potenciálně zneužitelné informace. Přístupové údaje k jiným účtům mezi ně ale většinou nepatří a konkrétně hesla většina aplikací z pochopitelných důvodů zobrazuje nepředvyplněná a při možnosti jej změnit navíc s podmínkou zadání starého hesla.

2.3.3.2 Prevence útoku

V administrační části internetových aplikací, pro akce, které mažou určité záznamy nebo je jiným způsobem mění, se doporučuje zásadně používat HTTP metodu POST. (To útok CSRF znesnadňuje, ale ještě zcela nevylučuje.)

Používat **autorizační token** – tedy náhodně vygenerovaný řetězec pro tuto akci, platící jen pro aktuálního užfivatele, ideálně pokaždé (tj. pro každý vygenerovaný formulář) jiné. Typicky skript, starající se o administrační část aplikace, si před zobrazením formuláře vygeneruje tento autorizační token, který si jednak zapamatuje (uloží do session, databáze, …) a současně do onoho formuláře vloží (jako skryté vstupní pole). Při zpracování odeslaných dat pak tuto proměnnou porovnává s předtím uloženou hodnotou. V případě shody může požadavek zpracovat, v případě neshody se zřejmě jedná o pokus o Cross-Site Request Forgery. Autorizační token by neměl být od ničeho odvozený, zcela stačí v podobě (dostatečně velkého) náhodného čísla. Zatímco administrátor jej má v každém nabídnutém formuláři aplikace automaticky předvyplněný, útočník (nezávisle na počtu zaslaných podvrhnutých požadavků) autorizační token není schopen uhodnout.

Implementace autorizačním tokenem je sama o sobě považována za dostatečné opatření proti CSRF útokům. Nicméně, je více než vhodné používat ji v rámci ostatních bezpečnostních opatřeních, s kterými je možno ji kombinovat (zabezpečení webového serveru, nastavení limitů a přístupových práv, použití SSL/HTTPS nebo HTTP autentizace, zásady pro ukládání citlivých údajů a hesel (např. tzv. *password salting*, vyžadování starého hesla při jeho změně), ošetřování vstupů od uživatele, stratifikace uživatelských práv, atd.) [6].

2.3.4 Directory traversal

Directory traversal využívá nedostatečné zabezpečení jména vstupního souboru. Cílem tohoto útoku je získat přístup k souboru, který není přístupný. Tento útok využívá nedostatečné zabezpečení (software jedná přesně tak, jak má), jak protichůdný k využívání chybu v kódu.
Directory Traversal je také známý jako `.. /` (dot dot slash) útok, directory climbing, a ustupování [7].

2.3.4.1 Příklad útoku

```
<?php
$template = 'red.php';
if ( isset( $_COOKIE['TEMPLATE'] ) )
   $template = $_COOKIE['TEMPLATE'];
include ( "/home/users/inzertum.cz/templates/" . $template );
?>
```

Útok proti tomuto systému by mohly být pro odeslání těchto HTTP požadavku:

```
GET /vulnerable.php HTTP/1.0
Cookie: TEMPLATE=../../../../../../../../../../etc/passwd
```

Opakování `../` za adresou `/home/users/inzertum.cz/templates/` způsobilo přejítí do kořenového adresáře, a pak zahrnutí UNIX souboru pro hesla `/etc/passwd`.

2.3.4.2 Prevence útoku

Některé webové aplikace hledají nebezpečné znaky jako:

- `..`
- `../`
- `..\`

aby předešly tomuto útoku. Avšak řetězec dotazu je před použitím dekódován a tak mohou být tyto aplikace náchylné na přepisy typu:

- `%2e%2e%2f` (ekvivalent pro `../`)
- `%2e%2e/` (ekvivalent pro `../`)
- `..%2f` (ekvivalent pro `../`)
- `%2e%2e%5c` (ekvivalent pro `..\`)

Při ochraně před Directory Traversal je nutné splnit následující:

- Při požadavku na soubor nebo adresář je nutné postavit kompletní cestu a normalizovat všechny znaky (např. % 20 na mezery).
- Vždy se ujistit, jestli nevede požadovaná cesta mimo „Document Root“ (přístup k souborům mimo je odepřen).
- Ujistit se, že prvních N znaků v úplné cestě k souboru je stejných jako „Document Root“.

2.4 Hesla

Aplikace příslušným způsobem uchovává, eviduje a spravuje přihlašovací, osobní i jiné údaje registrovaných uživatelů. Zejména registrační údaje, jako jsou uživatelská jména a hesla, by měla být na serveru *uložena na bezpečném místě,* typicky v databázi s omezeným přístupem. Hesla navíc není vhodné ukládat v čisté podobě, ale *upravená některou jednocestnou hashovací funkcí*, například SHA nebo MD5. Nebudou tak čitelná pro administrátora, ale ani pro případného narušitele, který by se nějakým způsobem k databázi hesel dostal. Jeho jedinou možností, jak původní podobu hesla odhalit, pak zůstává útok hrubou silou, tzn. zkoušet stejnou jednocestnou funkcí šifrovat jedno potenciální heslo za druhým a porovnávat je s uloženými záznamy. Proto by měla být zvolená hashovací funkce dostatečně silná, aby byl takový postup v přiměřeném čase výpočetně nedosažitelný.

Hesla volená uživateli musí být *nesnadno odhadnutelná*. Rozhodně by neměla být tvořena normálními slovy běžně používanými v některém jazyce, a už vůbec ne výrazy úzce spojenými s daným uživatelem, jako jméno manželky, rodné číslo, poštovní adresa či snad dokonce heslem shodným s přihlašovacím jménem. Tradičně se doporučuje vymyslet

alespoň osmiznakové heslo, v němž budou zkombinovány číslice, verzálky, mínusky a nealfanumerické znaky. Zdánlivě protichůdnou podmínkou je, aby se samotnému uživateli dobře pamatovalo. Toho lze ale snadno docílit třeba určením hesla jako akronymu z nějaké pro uživatele snadno zapamatovatelné věty. Pokročilejší systémy implementují algoritmy, které požadovanou složitost a neodhadnutelnost nastavovaného hesla kontrolují. Myslí při tom třeba i na takové věci, jako symetrická kombinace kláves na klávesnici apod.

2.4.1 Hashovací funkce

Hashovací funkce řadíme mezi jednocestné funkce, které na vstupu prijímají text a na jeho základě vytvoří výstupní „haš“. Výstup hašovací funkce, haš, je malý, přitom však umožňuje identifikaci zprávy.

Vlastnosti hašovací funkce:

- vrací haš o pevné délce, nejčastěji 16 až 20 bajtů (nezáleží přitom na tom, kolik MB je na vstupu, výstup bude stejný)
- každému vstupu odpovídá nějaký výstup
- je prakticky nemožné vytvořit takový vstup, který by vyprodukoval stejný výstup
- po změně i jediného bitu je výstup naprosto odlišný

2.4.1.1 MD5

Message-Digest algorithm je rozšířená rodina hašovacích funkcí, která vytváří ze vstupních dat výstup (otisk) fixní délky. Otisk je též označován jako miniatura, kontrolní součet (v zásadě nesprávné označení), fingerprint, hash (česky někdy psán i jako haš). Jeho hlavní vlastností je, že malá změna na vstupu vede k velké změně na výstupu, tj. k vytvoření zásadně odlišného otisku.

Algoritmus MD5 se prosadil do mnoha aplikací (např. pro kontrolu integrity souborů nebo ukládání hesel). MD5 je popsán v internetovém standardu RFC 1321 a vytváří otisk o velikosti 128 bitů. Byl vytvořen v roce 1991 Ronaldem Rivestem, aby nahradil dřívější hašovací funkci MD4.

2.4.1.2 SHA-1

SHA (**Secure Hash Algorithm**) navrhla organizace NSA (Národní bezpečnostní agentura v USA). SHA je rozšířená hašovací funkce, která vytváří ze vstupních dat výstup (otisk) fixní délky. Otisk je též označován jako miniatura, kontrolní součet (v zásadě

nesprávné označení), fingerprint, hash (česky někdy psán i jako haš). Jeho hlavní vlastností je, že malá změna na vstupu vede k velké změně na výstupu, tj. k vytvoření zásadně odlišného otisku.

SHA-1 (stejně jako SHA-0) vytváří 160 bitový obraz zprávy s maximální délkou 2^{64} - 1 bitů. Je založený na principech, které používal Ronald L. Rivest z Massachusetts Institute of Technology (MIT) v návrhu MD4 a MD5 algoritmů. SHA se používá u několika různých protokolů a aplikací, včetně TLS a SSL, PGP, SSH, S/MIME a IPsec, ale i pro kontrolu integrity souborů nebo ukládání hesel.

3 PŘÍSTUPNOST WEBOVÝCH STRÁNEK

3.1 Validní web

Každý jazyk má svá pravidla a nejinak je tomu u HTML. Validita by se dala přirovnat ke gramatickým pravidlům. Analýzou validity webu se rozumí kontrola zdrojového kódu webu ve formátu HTML a XHTML a kaskádových stylů CSS. Validní kód zaručuje správnou čitelnost webu pro vyhledávače a indexující roboty. Kód musí odpovídat obecně přijatým standardům společnosti W3C (World Wide Web Consortium) - http://validator.w3.org/. Jestliže web není validní, je nutné prohřešky proti standardům W3C odstranit, protože právě ony mohou být příčinou toho, proč indexující roboti některé webové stránky či jejich obsah přeskočí.
Nedostatky, které se nacházejí pouze ve formátu HTML, lze většinou snadno opravit. Pokud však chyby generují skriptové aplikace, pak může být jejich odstranění zdlouhavé a náročné.

3.2 Sémantický web

Jestliže má HTML gramatická pravidla v podobě validity, má také stylistická pravidla, což není nic jiného než sémantika.

Ta sleduje jestli je dodržováno správné vyznačování jednotlivých úseků zdrojového kódu každé webové stránky. HTML zná mnoho značek (či tagů) a většina z nich má také svůj význam, který bychom měli dodržovat. Jestliže tedy kupříkladu máme značku pro nadpis a chceme vložit do stránek nadpis, měli bychom použít značku nadpis [11].

Špatný zápis:

```
<div id="hlavni-nadpis">Nesémantický nadpis</div>
```

Správný zápis:

```
<h1>Sémantický nadpis</h1>
```

3.2.1 Výhody dodržování sémantiky

- Sémantický web se s vypnutými styly zobrazí korektně a uživatel by jej měl bez jakýchkoliv větších problémů umět použít. Tím se web samozřejmě stává mnohem přístupnějším. Sémantika je jedním z klíčových kamenů přístupnosti**.**

- Pokud jsou stránky nesémantické a neubdou řádně vyznačeny alespoň základní části stránky, web nebude nikdy SEO-friendly a jen těžko se bude prokousávat na přední místa ve vyhledávačích. Sémantický web je přátelský nejen vůči postiženým, ale také vůči robotům**.**

- Pojetí moderního webdesignu zní, oddělit obsah od vzhledu. Neboli kaskádové styly mají určovat *vzhled* dokumentu a HTML značky zase jeho *smysl*.

3.2.2 Sémantické značky

3.2.2.1 Kořenový element `<html>`

V XHTML musí být (narozdíl od HTML) kořenovým elementem značka `<html>`. Sémantický dokument by měl touto značkou začínat vždy, čímž v podstatě definujeme, že všechno mezi těmito značkami má být součástí webové stránky.

Poté by měla následovat značka `<head>`, ve které je definována hlavička dokumentu. Právě zde by se měly nacházet informace, které nenesou pro uživatele prohlížející si stránky prakticky žádnou informační hodnotu. To znamená meta-tagy, odkazy na externí styly či skripty apod. Jedinou výjimku tvoří značka `<title>`, která obsahuje stručný popisek dané webové stránky a kterou uživatel obvykle vidí v horním pruhu prohlížeče.

Další důležitá značka je `<body>`, která je pravý opak k předchozímu tagu. Zde by se měl nalézat samotný obsah dokumentu, kvůli kterému na stránky uživatel zavítal. To znamená texty, obrázky, soubory ke stáhnutí (respektive odkazy na ně), flashové animace apod.

3.2.2.2 Nadpisy

Nadpisy jsou snad nejdůležitější značkou pro logické rozdělení stránky a proto by se určitě měli používat. HTML zná celkem šest úrovní nadpisů a sice `<h1>` až `<h6>`, přičemž `<h1>` je nejdůležitější a `<h6>` nejméně důležitý. Některé zdroje tvrdí, že nadpis `<h1>` by se měl na stránce vyskytnout pouze jednou, což opravdu není špatné dodržovat. Stránce to dodá lepší řád, protože ve chvíli, kdy bude na stránce více `<h1>` nadpisů, jediný prvek, který bude celou stránku logicky sdružovat, bude titulek `<title>`, který však není tak úplně součástí stránky.

Další důležitá věc je návaznost nadpisů, to znamená, že by měly jít postupně podle důležitosti. Neboli po hlavním nadpisu `<h1>` by měl následovat nadpis `<h2>` a potom `<h3>`. Nebylo by zrovna logické, kdyby po `<h1>` následoval nadpis `<h3>`. Pochopitelně tím není myšleno, že by nadpisy měly neustále klesat.

Nadpisem může být i obrázek. Má to tu výhodu, že například prohlížeče, které nepracují s obrázky, budou jako hlavní nadpis brát alternativní text (který pochopitelně musí být vyplněn).

```
<h1>
  <img src="nadpis.jpg" alt="Sémantika - pravý význam HTML značek">
</h1>
```

3.2.2.3 Odstavce

Text nemůže volně plavat v dokumentu, každý text totiž musí být vložen do nějaké *logické* značky. V případě XHTML validního kódu by stačilo text vložit například do nějakého blokového elementu jako je `<div>`. Validita by byla splněna, ale sémantice nikoliv, protože `<div>` není sémantická značka. Tudíž v tomto případě je nejlepší řešení vložit text do odstavce.

```
<p>Text v odstavci.</p>
```

3.2.2.4 Obrázky

Moderní webdesign praví, že dekorační obrázky se mají vkládat přes kaskádové styly (jako pozadí nějakého elementu), zatímco obrázky, které mají nějakou informační hodnotu, přes značku `<img>`. Je to logické, protože při vypnutých stylech jsou narušeny pouze obrázky bez informační hodnoty. Každý takový obrázek vkládaný pomocí značky `<img>` by také měl mít atribut `alt`, neboli alternativní text.

3.2.2.5 Zvýraznění textu

Značka `<b>` slouží k vymezení textu, který má být tučný, kdežto značka `<strong>` vymezuje důležitý či silně zvýrazněný text (nenechte se zmást, na tomto webu mám značku `<strong>` přestylovanou na zeleno, bez stylů se zobrazí stejně).

Kromě značky `<strong>` existuje ještě element `<em>`, který slouží ke stejnému účelu. Rozdíl mezi značkou `<strong>` a `<em>` je ten, že `<strong>` je důležitější, dává na slovo větší důraz. Žádný jiný sémantický rozdíl mezi nimi není. V prohlížeči se pak `<em>` obvykle vykreslí kurzívou. Rozdíl mezi `<em>` a `<i>` je v podstatě stejný jako rozdíl, který jsem vysvětlil v předchozí kapitole. Značka `<i>` nemá sémantický význam. Pouze vykreslí obsah kurzívou, nic víc neurčuje.

HTML má sice dvě značky na zesílení důrazu, ale ani jeden na zeslabení důrazu, což je trochu škoda. Jediná značka, která má podobný účel, je `<small>`. Jenže u ní je ten problém, že vlastně nemá sémantický význam, je to podobný problém jako třeba s `<b>`. Dá se například použít pro patičku.

3.2.2.6 Číslované seznamy

Číslované seznamy se uvozují párovou značkou `<ol>` a jednotlivé položky se pak tvoří nepárovou značkou `<li>`. Značka `<ol>` má ještě dva zásadní atributy a to `type` a `start`. První z nich určuje typ číslování.

Atribut `type` má svůj jasný sémantický význam. Změnu číslování můžete totiž provádět ze dvou důvodů: jeden je grafický a druhý logický. A v tomto případě je sémanticky správné použít atribut `type`.

3.2.2.7 Nečíslované seznamy

Pro nečíslovaný seznam (nezáleží zde na pořadí položek), je vhodné použít značku `<ul>`. I tento tag má zavržený atribut `type`. Zde totiž atribut `type` mění pouze vzhled odrážek, takže o sémantice nemůže být řeč. Pro změnu vzhledu odrážek u nečíslovaného seznamu, je dobré použít kaskádové styly.

3.2.2.8 Menu

S menu bývá obvykle problém, protože hodně stránek řeší vkládání menu různými způsoby. Některé jsou sémantické více, některé méně, ale snad žádné nejsou sémantické úplně. V zásadě jediná sémantická možnost, jak do stránky vložit menu, je prostřednictvím značky `<menu>` (nečekané, že ano). Jednotlivé položky menu se pak definují stejně jako položky jiného seznamu, tudíž značkou `<li>`. Ve striktní verzi je pouze obyčejný nečíslovaný seznam `<ul>`.

Sémanticky správný zápis menu**:**

```
<menu>
  <li><a href="#">Návody</a></li>
  <li><a href="#">Praxe</a></li>
  <li><a href="#">Odkazy</a></li>
</menu>
```

3.2.2.9 Citace

K zapisování citací slouží celkem tři elementy a sice `<q>`, `<blockquote>` a `<cite>`. Každý z nich má pochopitelně jinou funkci. První dvě značky jsou si nejvíce podobné, protože do obou z nich vkládáte citovaný text, ale s tím rozdílem, že `<q>` je řádkový element a `<blockquote>` blokový. Takže pokud máte krátkou citaci, kterou vložíte přímo do odstavce, použijete značku `<q>`. Jestliže ale chcete zapsat delší citaci, která se má zobrazit jako samostatný odstavec (nebo odstavce - `<blockquote>` jakožto blokový element může obsahovat další odstavce), použijte právě `<blockquote>`.

Obě tyto značky mají ještě nepovinný atribut `cite` (není značka `<cite>`), který obsahuje URL, odkud je text citován.

Poslední značka `<cite>` obsahuje další zdroje k citaci nebo například jméno osoby, která je zrovna citována.

3.2.2.10 Zkratky

I zkratky mají v HTML svou značku, respektive hned dvě. První z nich, `<abbr>`, slouží k vyznačováních zkratek a druhá značka `<acronym>` k označení zkratového slova. Rozdíl mezi zkratkou a zkratovým slovem je asi takový, že zkratové slovo se vyslovuje jako jedno slovo, kdežto zkratka se hláskuje po písmenech. Takže například NATO je zkratové slovo a CSS je zkratka. Používání značky `<abbr>` má ale jeden takový háček a to ten, že MSIE tuto značku nezná.

Pouze vyznačit nějakou zkratku moc velký význam nemá, správně by totiž ještě měla obsahovat vysvětlení. Obvykle se vkládá do atributu `title`, takže například zkratka CSS má ve zdrojovém kódu takovýto zápis:

```
<abbr title="Cascading Style Sheets - Kaskádové styly">CSS</abbr>
```

3.2.2.11 Zdrojový kód

Zdrojový by měl být vložen do značky `<code>`. Prohlížeč ho pak bude chápat jako zdrojový kód něčeho. Žádné další významné atributy tato značka nemá. Často se kombinuje se značkou `<pre>`, která zaručí, že budou ponechány a prohlížečem interpretovány bílé mezery. Toho samého efektu lze docílit pomocí kaskádových stylů, ale nejbezpečnější a nejsémantičtější možnost je za pomocí této značky. Takže sémantický zápis zdrojového kódu by vypadal asi takto:

```
<pre>
  <code>
    &lt;em&gt;William Shakespeare&lt;/em&gt;
  </code>
</pre>
```

3.2.2.12 Tabulky

Tabulku by měla začínat značkou `<table>`. Po této značce by měla následovat značka `<caption>`, což je vlastně takový nadpis tabulky - slouží k rychlému zorientování, k čemu ta tabulka slouží nebo co obsahuje. Atribut `summary` by měl obsahovat detailnější obsah tabulky. Atribut se vkládá do značky `<table>`.

Dále existují tři podobné značky na celkové rozdělení tabulky: `<thead>`, `<tbody>` a `<tfoot>`. První značka, `<thead>`, ohraničuje záhlaví tabulky, to znamená obvykle první řádek tabulky, kde se nachází popisky sloupců a kde vlastně ještě žádná tabulková data nejsou, nacházejí se zde pouze ty popisky. Další značka, `<tbody>`, potom obsahuje tělo tabulky, neboli samotný obsah. Tento tag může být použít vícekrát, v závislosti na tom, kolik různých částí tabulka má. Poslední značka, `<tfoot>`, tvoří zápatí a obvykle obsahuje nějaké součty nebo celkový souhrn tabulky. Tyto tři značky však nejsou povinné a mnohdy se ani použít nedají. V případě možnosti je však doporučeno použít je.

Samotná tabulka se pak tvoří další trojicí značek a sice: `<tr>`, `<th>` a `<td>`. Jednotlivé řádky tabulky tvoříme pomocí značky `<tr>`. Druhá značka, `<th>`, by měla určovat hlavičkovou informaci, měla by tedy býti v elementu `<thead>`. Pro datové informace slouží značka `<td>`, která, stejně jako předchozí značka, tvoří jednotlivé sloupce tabulky.

```
<table summary="ceník služeb">
  <caption>
    Ceník našich služeb
  </caption>
  <thead>
    <tr>
      <th>služba</th>
      <th>cena bez dph</th>
      <th>cena s dph</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Úklid</td>
      <td>3000,-</td>
      <td>3570,-</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>cena celkem</td>
      <td>6000,-</td>
      <td>7140,-</td>
```

```
    </tr>
  </tfoot>
</table>
```

3.2.2.13 Formuláře

Formulář začíná značkou `<form>`, ve které jsou potom další a další značky, tvořící samotný formulář. Ze sémantického hlediska má smysl se zmínit o třech značkách: `<fieldset>`, `<legend>` a `<label>`.

Delší formulář, který je možné rozdělit do nějakých logických podskupin, se rozděluje za pomocí značky `<fieldset>`. Je vhodné ho používat, protože prohlížeč vykreslí kolem části formuláře ohraničení, čímž se celý formulář stává mnohem přehlednějším.

Předchozí značka měla za úkol spojit dohromady tématicky blízké části formuláře. Tag `<legend>` potom tyto jednotlivé části pojmenovává - je to v podstatě nadpis. Tento text se potom objeví v rámečku okolo fieldsetu. Vypadá to velmi efektivně a každý pochopí o co jde, takže je velmi výhodné tuto značku používat. Značka - `<legend>` musí následovat hned za `<fieldset>`.

Každé pole obvykle obsahuje nějaký doprovodný text, který vysvětluje, co se do daného políčka má zapsat, stejně tak je u každého přepínače uveden nějaký název, který vybíráte. Tento text by měl obsažen ve značce `<label>`. A aby bylo jednoznačně určené, k jakému poli se popisek vztahuje, přidává se zde obvykle ještě atribut `for`, kde je obsažen identifikátor onoho pole. Značka `<label>` se určitě vyplatí používat, protože formuláře jsou pak použitelnější, neboť například u radiobuttonu stačí kliknout na popisek a přepínač se aktivuje [11].

```
<form>
  <fieldset>
    <legend>Matka</legend>
    <p>
      <label for="jmeno-m">Jméno matky:</label>
      <input type="text" id="jmeno-m">
    </p>
    <p>
      <label for="prijmeni-m">Příjmení matky:</label>
      <input type="text" id="prijmeni-m">
    </p>
  </fieldset>
  <fieldset>
    <legend>Otec</legend>
    <p>
```

```
      <label for="jmeno-o">Jméno otce:</label>
      <input type="text" id="jmeno-o">
    </p>
    <p>
      <label for="prijmeni-o">Příjmení otce:</label>
      <input type="text" id="prijmeni-o">
    </p>
  </fieldset>
</form>
```

3.2.3 Nesémantické značky

Přehled všech nesémantických značek, které by se tudíž neměli používat příliš často, ideálně vůbec:

- `<u>`
- `<tt>`
- `<s>`
- `<strike>`
- `<small>`
- `<big>`
- `<blink>`
- `<marquee>`

3.3 Přístupný web

Pravidla tvorby přístupného webu pro účely novely Zákona č. 365/2000 Sb., o informačních systémech veřejné správy.

3.3.1 Pravidla tvorby přístupného webu

3.3.1.1 Dotupný a čitelný obsah webu

- Netextové prvky nesoucí významové sdělení mají svou textovou alternativu.
- Informace sdělované prostřednictvím skriptů, kaskádových stylů, obrázků a jiných doplňků na straně uživatele jsou dostupné i bez kteréhokoli z těchto doplňků.
- Informace sdělované barvou jsou dostupné i bez barevného rozlišení.
- Barvy popředí a pozadí jsou dostatečně kontrastní. Na pozadí není vzorek, který snižuje čitelnost.
- Předpisy určující velikost písma nepoužívají absolutní jednotky.
- Předpisy určující typ písma obsahují obecnou rodinu písem.

3.3.1.2 Práci s webovou stránkou řídí uživatel

- Obsah WWW stránky se mění, jen když uživatel aktivuje nějaký prvek.
- Webová stránka bez přímého příkazu uživatele nemanipuluje uživatelským prostředím.
- Nová okna se otevírají jen v odůvodněných případech a uživatel je na to předem upozorněn.
- Na webové stránce nic nebliká rychleji než jednou za sekundu.
- Webová stránka nebrání uživateli posouvat obsahem rámů.

3.3.1.3 Srozumitelnost a přehlednos t informací

- Webové stránky sdělují informace jednoduchým jazykem a srozumitelnou formou.
- Úvodní webová stránka jasně popisuje smysl a účel webu. Název webu či jeho provozovatele je zřetelný.
- Webová stránka i jednotlivé prvky textového obsahu uvádějí své hlavní sdělení na svém začátku.
- Rozsáhlé obsahové bloky jsou rozděleny do menších, výstižně nadepsaných celků.
- Informace zveřejňované na základě zákona jsou dostupné jako textový obsah webové stránky.
- Na samostatné webové stránce je uveden kontakt na technického správce a prohlášení jasně vymezující míru přístupnosti webu a jeho částí. Na tuto webovou stránku odkazuje každá stránka webu.

3.3.1.4 Jasné a pochopitelné ovládání webu

- Každá webová stránka má smysluplný název, vystihující její obsah.
- Navigační a obsahové informace jsou na webové stránce zřetelně odděleny.
- Navigace je srozumitelná a je konzistentní na všech webových stránkách.
- Každá webová stránka (kromě úvodní webové stránky) obsahuje odkaz na vyšší úroveň v hierarchii webu a odkaz na úvodní WWW stránku.
- Všechny webové stránky rozsáhlejšího webu obsahují odkaz na přehlednou mapu webu.
- Obsah ani kód webové stránky nepředpokládá, že uživatel již navštívil jinou stránku.
- Každý formulářový prvek má přiřazen výstižný nadpis.

- Každý rám má vhodné jméno či popis vyjadřující jeho smysl a funkčnost.

3.3.1.5 Zřetelné a návodné odkazy

- Označení každého odkazu výstižně popisuje jeho cíl i bez okolního kontextu.
- Stejně označené odkazy mají stejný cíl.
- Odkazy jsou odlišeny od ostatního textu, a to nikoli pouze barvou.
- Obrázková mapa na straně serveru je použita jen v případě, že nebylo možné pomocí dostupného geometrického tvaru definovat oblasti v obrázkové mapě. V ostatních případech je použita obrázková mapa na straně uživatele. Obrázková mapa na straně serveru je vždy doprovázena alternativními textovými odkazy.
- Uživatel je předem jasně upozorněn, když odkaz vede na obsah jiného typu, než je webová stránka. Takový odkaz je doplněn sdělením o typu a velikosti cílového souboru.

3.3.1.6 Technicky způsobilý a strukturovaný kód

- Kód webových stránek odpovídá nějaké zveřejněné finální specifikaci jazyka HTML či XHTML. Neobsahuje syntaktické chyby, které je správce webových stránek schopen odstranit.
- V metaznačkách je uvedena použitá znaková sada dokumentu.
- Prvky tvořící nadpisy a seznamy jsou korektně vyznačeny ve zdrojovém kódu. Prvky, které netvoří nadpisy či seznamy, naopak ve zdrojovém kódu takto vyznačeny nejsou.
- Pro popis vzhledu webové stránky jsou upřednostněny stylové předpisy.
- Je-li tabulka použita pro rozvržení obsahu webové stránky, neobsahuje záhlaví řádků ani sloupců. Všechny tabulky zobrazující tabulková data naopak záhlaví řádků a/nebo sloupců obsahují.
- Všechny tabulky dávají smysl čtené po řádcích zleva doprava.

4 SEM A SEO

4.1 SEM – Search Engine Marketing

SEM neboli marketing ve vyhledávačích vychází z předpokladu, že nejdůležitější je být nalezen vyhledávačem a na dané klíčové slovo související s obsahem stránky se zobrazit hledajícímu uživateli.

SEM využívá opačného principu, než je obvyklý v reklamě. Většina reklamních technik používá lineární strategii tlaku, tzv. push marketing, tlačí produkty k zákazníkm, kteří o ně často ani nestojí. Reklamní agentury na zákazníka útočí z billboardů, televizních obrazovek, rozhlasu, z regálů vykukují vlaječky, v metru a v autobusu jsou pestrobarevné letáky, schránku zaplňují tuny nabídkových katalogů.

Naopak SEM není tak agresivní, je totiž postaven na nelineárním strategii tahu (pull marketing). Dává zákazníkům přesně to co chtějí.

Search engine marketing se děli na dvě základní části. Základní rozdělení je, že s vyhledávačí buď spolupracujeme formou umisťování placených odkazů, anebo se snažíme přizpůsobit obsah stránek tak, aby je vyhledávač sám při řazení výsledků po zadání dotazu návštěvníkem umístil na vyhledávanou stránku na co nejlepší pozici.

První část se tedy zaobírá především přidáváním odkazů na samotné vyhledávače a práci s kontextovými odkazy a PPC systémy. Další z technik, která je podrobněji rozebrána v následující kapitole je SEO.

4.2 SEO – Search Engine Optimization

SEO je optimalizace stránek pro vyhledávače. Zjednodušeně se jedná o konkrétní techniky zabývající se způsobem, jak umístit stránky na co nejlepší pozici ve vyhledávání.

4.3 Katalogy

Katalog je ve své podstatě web rozdělený podle kategorií, který obsahuje odkazy na jiné weby. Vyhledávací funkce je dnes v katalozích samozřejmostí. Vyhledávač v katalogu umí najít výsledek pokud se hledané slovo shoduje s:

- slovem v titulku
- slovem v obecném popisu
- názvem kategorie, do které je web zařazen

- klíčovým slovem zadaným při registraci do katalogu

4.3.1 Nejznámější katalogy

Mezi nejznámější katalogy ve světě patří Yahoo!, v Česku je katalog stránek na Seznamu, Atlasu i Centru. Odkazy jsou zde řazeny do tématických oblastí. Záznam provádí sám provozovatel stránky.

Bez zmínky nesmí zůstat ani projekt ODP – Open Directory Project – http://dmoz.org, který upravují dobrovolní editoři z celého světa. Jen v České republice nyní funguje zhruba stovka dobrovolných editorů. Registrace stránek v tomto katalogu je pro SEO velmi důležitá. Právě vzhledem ke své nestrannosti a systému, jakým jsou odkazy do katalogu přidávány (o zařazení rozhodují prověření a nezávislí editoři), jsou většinou stránky tohoto katalogu vyhledávači dobře hodnoceny a považovány za důvěryhodnou autoritu. Odkazy směrující z podobných stránek mají pro optimalizované stránky velkou hodnotu.

4.4 Vyhledávače

Fulltextový vyhledávač aktivně vyhledává a prohledává weby a jejich stránky, vytváří vlastní index (databázi) a na základě dotazu návštěvníků nabízí výsledky vyhledávání ze svého – fulltextového indexu. Např. Google vytváří svou databázi tak, že neustále vysílá po síti několik speciálních programů často nazývané robot, spider (pavouk), crawler, fish, worm. Jejich úkolem je stahovat navštívené stránky do hlavní databáze vyhledávače. V největším provozu umí pavouci Google současně stahovat 100 stránek za jedinou sekundu a průběžně tak nepřetržitě procházejí miliardy stránek, které dnes na webu existují. Tento proces nazývající se „procházení“ (anglicky „crawling“) se řídí přesným algoritmem: počítačové programy určují, které stránky se budou procházet a kolik stránek z každé jednotlivé webové prezentace se projde.

4.4.1 Index vyhledávače

Index vyhledávače je jednoduše řečeno databáze všech slov vyskytujících se na prohledávaných stránkách, která má za úkol co nejvíce urychlit prohledáváni.

V současné době se používají dva typy indexů:

- Invertované

- Příponové stromy

Invertovaný index je abecedně seřazený seznam všech termínů, které se na stránce vyskytují. Jedná se o nejpoužívanější typ indexu v současných vyhledávačích. Invertované indexy vytvoří slovník sestávající ze všech slov v dokumentu nebo souboru dokumentů (stránek) a ke každému slovu přiřadí dokumenty, ve kterých se nachází, a také jeho pozici v rámci dokumentu.

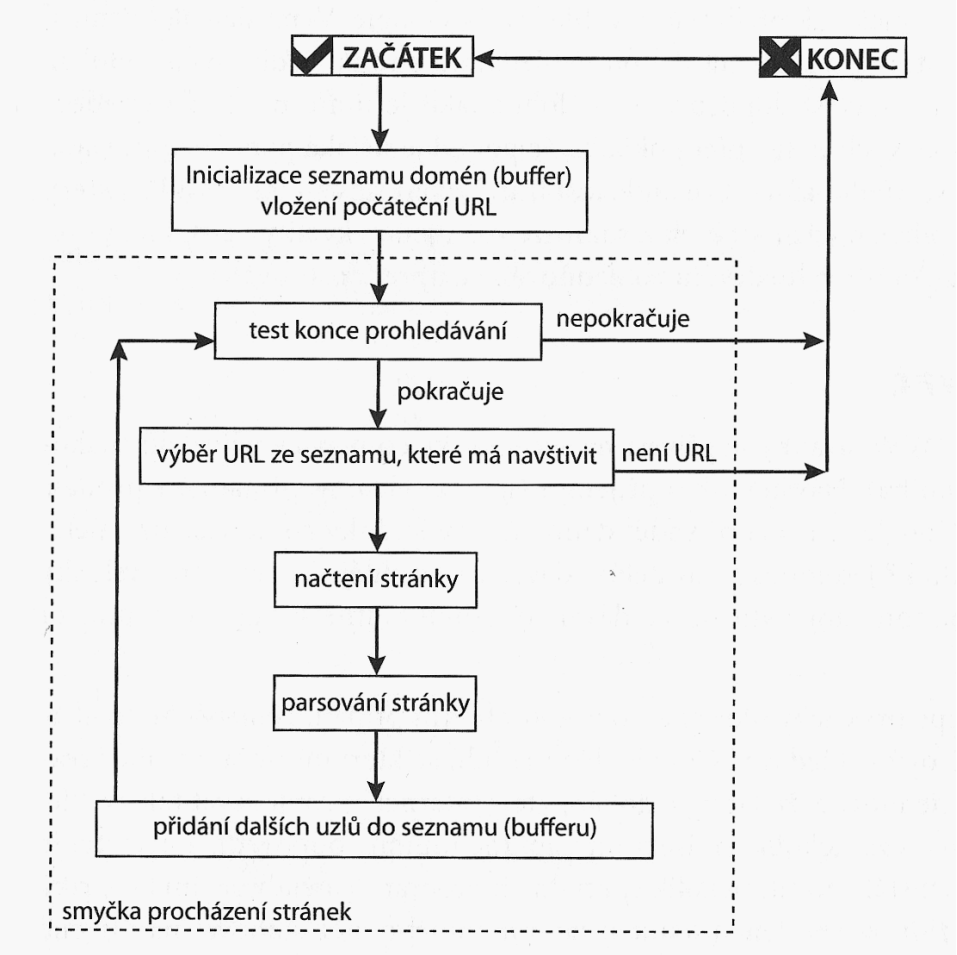
Tab. 1 – Ukázka výpisu z indexu vyhledávače

pozice	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
výrazy	t	e	n	t	o		t	e	x	t		b	u	d	e		i	n	d	e	x	o	v	á	n

Příponové stromy využívají stromovou strukturu na ukládání řetězců znaků nad danou abecedou, vhodnější pro práci s v frázemi, těžko se však udržují a vytváří, následně se však s nimi pracuje rychleji.

4.4.2 Pavouci (roboti)

Pavoukova cesta začíná ve vlastní databázi. Zde si vybere ve své databázi odkaz stránky, kterou následně navštíví. Přijde na hlavní stránku, tu si pročte a její obsah si stáhne do skladiště na vlastním serveru. Ve zdrojovém kódu hledá odkazy ukryté v párových značkách `<a>` a `</a>`. Jakmile na tento odkaz přijde, novou stránku načte a to několikrát opakuje.



Obr. 5 – Cesta pavouka, který načítá adresy a odkazy (převzato[5])

4.4.2.1 Typy pavouků

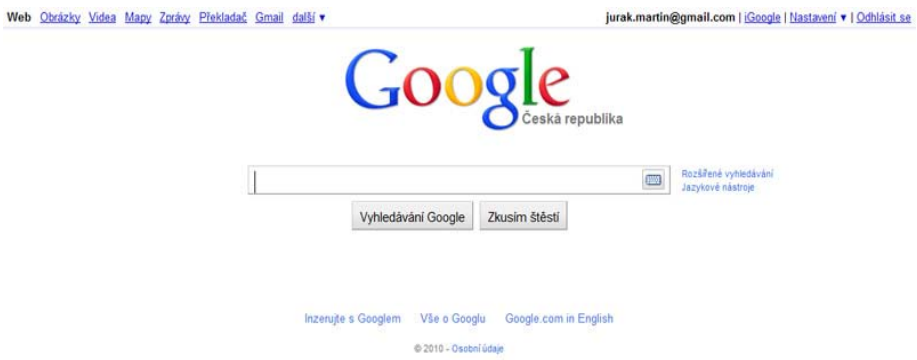
Různé typy pavouků mají za úkol jiné úkoly a činnosti:

- Stahovači – vytvářejí výtahy z dokumentů, které ukládají do databáze. Vytvářejí indexy, které stanoví vztah slov a frází nalezených v dokumentech HTML k adrese URL.
- Prohlížeči – prohlížejí okolí a názvy obrázků, které potom stahují opět v náhledech do databáze společně s odkazem na stránku, kde jsou umístěny. Existují podobné typy těchto pavouků pro různé typy dokumentů, jako jsou DOC, XLS, PDF a další.
- Kontroloři – hledají odkazy na již neexistující stránky. Informace vrací do databáze a při opakovaném nenalezení navrhne vyřazení z hlavní databáze.
- Statistici – shromažďují údaje o počtech odkazů a využívání jednotlivých stránek, podle čehož následně určují oblíbenost stránek.
- Počtáři – počítají množství stránek, čímž zjišťují jejich nárůst.

Jakmile se informace o stránce dostane do hlavní databáze, na místo vyrazí robot, který má na starosti načítání stránek, tzv. fetching. Denně tak robot zpracuje až dva miliony stránek. Robot odešle HTTP dotaz a přečte a zpracuje odpověď. Přitom ho zajíma:

- Hlavička dokumentu, kde zpracuje informace v metaznačkách
- Redirekce, přesměrování
- Stavové kódy
- Informace o poslední změně na stránce
- Čas, za který stránku zpracuje (na příliš velké, nedostupné či poškozené stránky není třeba odkazovat)

4.4.3 Google



Obr. 6 – Úvodní strana fulltextového vyhledávače Google

4.4.3.1 Přehled technologie

Společnost Google se jako jediná zaměřuje na vývoj „dokonalého vyhledávače“ definovaného spoluzakladatelem společnosti Larrym Pagem jako něco, co „přesně rozumí zadání a přesně poskytuje požadované výsledky“. K tomuto cíli chce společnost Google dospět vytrvalým úsilím o inovace i tím, že odmítá přijímat omezení stávajících modelů. Výsledkem je, že společnost Google vyvinula vlastní obslužnou infrastrukturu a průlomovou technologii PageRank™, která změnila způsob vyhledávání.

Vývojáři společnosti Google si byli od začátku vědomi toho, že poskytování nejrychlejších a nejpřesnějších výsledků hledání vyžaduje nový způsob uspořádání serverů. Zatímco většina vyhledávačů se uchýlila k několika velkým serverům, u kterých při plném zatížení často docházelo ke zpomalení, společnost Google použila propojené osobní počítače, s jejichž pomocí bylo možné rychle nalézt odpověď na každý dotaz. Tato inovace se vyplatila rychlejší dobou odezvy, větší škálovatelností a nižšími náklady. Tento nápad postupně převzali ostatní, ale společnost Google pokračovala ve zlepšování koncové technologie, aby byla ještě účinnější.

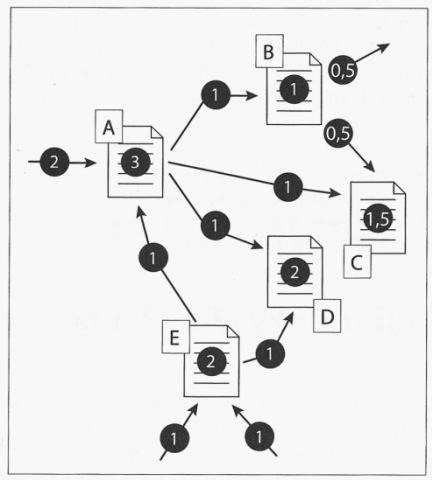
Software, který stojí za vyhledávací technologií společnosti Google, provádí sérii simultánních výpočtů vyžadujících jen zlomek vteřiny. Tradiční vyhledávače jsou ve značné míře závislé na četnosti výskytu slova na webové stránce. Společnost Google využívá technologii PageRank™, která umožňuje prozkoumat celkovou strukturu odkazů na webu a určit nejdůležitější stránky. Dané informace jsou potom využity k provedení analýzy podle hypertextu, z níž vyplyne, jaké stránky jsou vzhledem k prováděnému vyhledávání relevantní. Díky kombinaci celkové důležitosti a relevantnosti vzhledem k danému dotazu mohou být při vyhledávání na Google nejdůležitější a nejdůvěryhodnější výsledky uvedeny jako první.

- Technologie PageRank: Technologie PageRank umožňuje objektivní měření důležitosti webových stránek řešením rovnice s více než 500 miliony proměnných a 2 miliardami termínů. Místo počítání přímých odkazů interpretuje technologie PageRank odkaz ze stránky A na stránku B jako hlas pro stránku B od stránky A. Technologie PageRank poté vyhodnotí důležitost stránky podle počtu získaných hlasů.

  Technologie PageRank zohledňuje rovněž důležitost každé stránky, která udělila hlas. Hlasy od některých stránek mají větší hodnotu, a odkazovaná stránka tak získá vyšší ohodnocení. Důležité stránky obdrží vyšší ohodnocení PageRank a zobrazí se na začátku výsledků vyhledávání. Technologie společnosti Google používá k určení důležitosti stránky souhrnné informace webu. Protože nedochází k lidskému zásahu ani k manipulaci s výsledky, uživatelé důvěřují vyhledávači Google jako zdroji objektivních informací, který není ovlivněn placenou inzercí.

Míra předávání této hodnověrnosti klesá s množstvím odkazů na stránce uvedených.

Jak je vidět na (Obr.6), na stránku A směrují pouze dva odkazy, ale protože jeden má hodnocení 2 a druhý 1, celkové hodnocení stránky A je 3. Na stránku C směrují také dva odkazy, ale protože jeden má hodnotu 1 a druhý odkaz ze stránky B jen 0,5, celkové hodnocení stránek je poloviční oproti hodnocení stránky A.



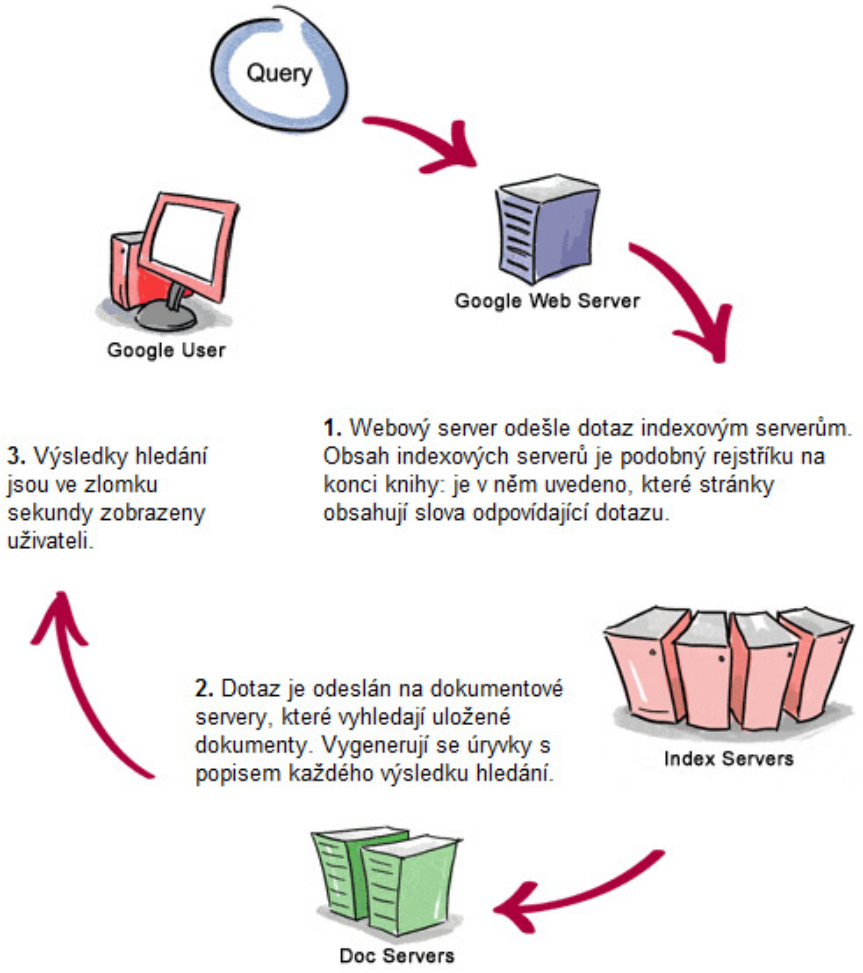
Obr. 7 – Jednoduché schéma předávání PageRanku mezi stránkami(převzato [5])

- Analýza podle hypertextu: Vyhledávač Google analyzuje také obsah stránky. Avšak místo prostého procházení textu na stránkách (který může být provozovateli webu manipulován prostřednictvím metaznaček) analyzuje technologie společnosti Google celý obsah stránky a zohledňuje faktory, jako jsou písma, odstavce a přesné umístění každého slova. Google také analyzuje obsah sousedních webových stránek a zajišťuje tak zobrazení výsledků, které nejlépe odpovídají dotazu uživatele.

PageRank je důležitý faktor, kterým se určuje umístění ve výsledcích vyhledávání (SERP – search engine result page). Není však nejdůležitější. Relevanci stránky po zadání vyhledávaného slova určuje Google přes 200 různých faktorů a PageRank je jen jedním z nich.

4.4.3.2 Životnost dotazu Google

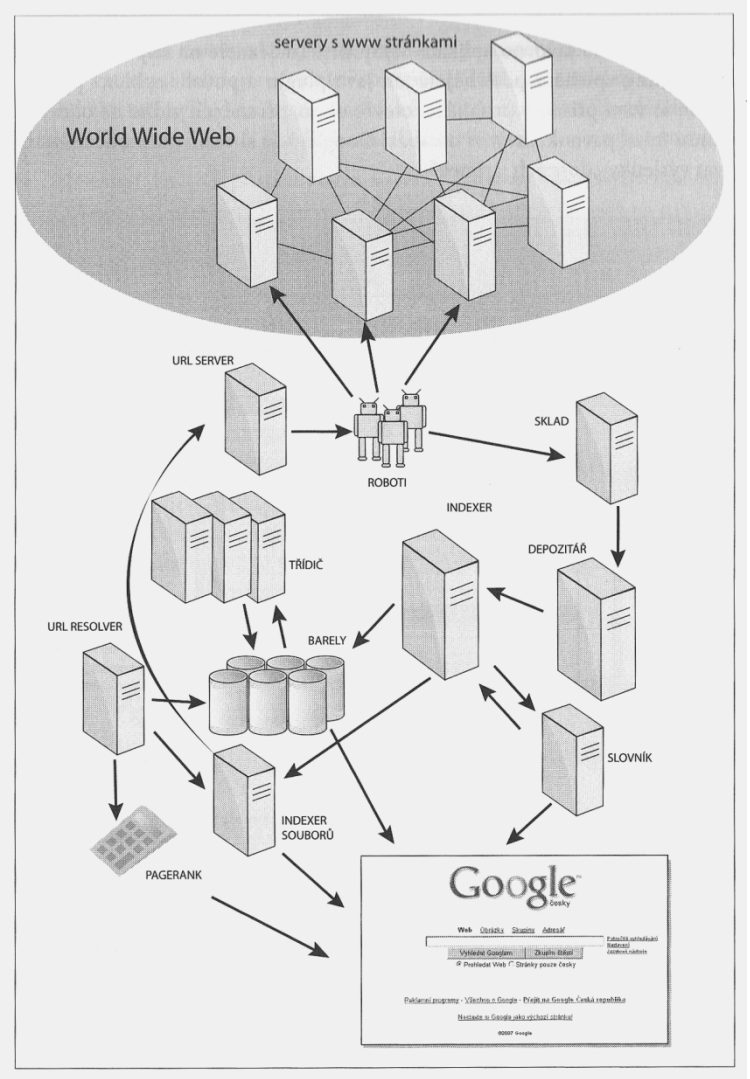
Životnost dotazu přes Google je obvykle méně než půl sekundy, ale je během ní nutné provést řadu různých kroků. Tyto kroky je třeba vykonat předtím, než lze výsledky doručit osobě vyhledávající informace.



Obr. 8 – Životnost dotazu Google

4.4.3.3 Procesy na pozadí Google

Pozadí Google se skládá ze *serveru obsahujícího seznam URL* adres, které pravidelně zasílá témto pavoukům. Stažené stránky jsou posílány do *skladového serveru* (store-reserver). Ve skladišti se stránky komprimují a uloží dále do depozitáře. Každá stránka dostane unikátní identifikační číslo, kterému se u Google říká *docID*. O zařazení do rejstříku (indexu) neboli indexování se stará tzv. indexer a sorter (třídič). Dalo by se říct, že indexer je srdcem vyhledávače.



Obr. 9 – Cesta informace databázemi Google

(převzato[5])

Indexer mimo jiné:

- Čte informace z depozitáře
- Dekomprimuje komprimované stránky
- Analyzuje rozložené stránky
- Komunikuje se zásobárnou stránek
- Spolupracuje se slovníkem
- Extrahuje odkazy a adresy URL

Každý dokument je převáděn do souboru slovních spojení zvaných hity. Hity obsahují záznam o slovu, pozici v dokumentu, přibližnou velikost fontu (písma) a jeho řez (kapitálky, verzálky, kurziva...). Následně indexer odesílá tyto hity do tzv. barelů, kde jsou připravené pro odesílání k uživatelům. Současně je každé slovo převedeno na další unikátní identifikační číslo (wordId). Přitom indexer spolupracuje se sdíleným slovníkem. Jakmile se slovo převede na wordID, jeho výskyt se zapíše do seznamu v barelech.

Indexer dále zajišťuje jinou důležitou funkci. Extrahuje z indexovaných stránek veškeré odkazy a zaznamenává nejenom samotnou adresu URL v odkazu, ale také tzv. anchor text (text v odkazu). Nyní vstupuje do procesu další program – URL resolver, který čte text v odkazu a převádí relativní adresy URL na absolutní a řadí je podle docID. Následně tyto odkazy páruje se stránkami v již zpracovaném rejstříku. Vytváří databázi odkazů a z této databáze pak čerpá další program data pro výpočet tzv. PageRanku všech dokumentů.

Google uchovává index všech svých internetových dat na řadě oddělených serverů, které jsou různě rozmístěny v tzv.Google data centrech. Dotazy pro vyhledávání jsou pak distribuovány do jednotlivých data center přes hlavní Google server. Aktualizace indexu se objevuje postupně tak, jak se jeden server po druhém průběžně aktualizuje.

4.4.4 Seznam

Nejpoužívanější český vyhledávač Seznam.cz má své vlastní ratingové hodnocení. Je nazýváno *S-rank.* Podobně jako u Google je S-rank stránky veličina, které by měla vyjadřovat důležitost každé stránky na českém webu. Počítá se zejména z odkazové sítě algoritmem, který zohledňuje jednak odkazy, které na stránku míří, ale i to, kam ze stránky odkazy vedou. Přesný výpočet S-ranku není veřejný. Dalo by se říct, že způsob hodnoceni stránek je největším know-how a nejvíce střeženým tajemstvím vyhledávačů.

Ale i tak je známé, že se S-rank vypočítává váženou nelineární kombinací různých veličin, v nichž výrazně převažují off-page faktory. Výpočet hlavního zdroje ranku se podobá algoritmu Hubs&Authorities Jona Kleinberga, ale je upraven tak, aby dával smysl i pro netematické množiny stránek.

Princip *Hubs&Authorities*, kterým se Seznam inspiruje, popisuje dva základní typy stránek – rozcestníky (hubs) a autority (authorities). Zatímco rozcestník je stránka, která odkazuje na mnoho autorit, autorita je stránka, na kterou odkazuje mnoho rozcestníků. Algoritmus,

který se v této souvislosti používá, se nazývá HITS. Ten sleduje, nakolik je stránka autoritou a nakolik je egocentrická. Obě hodnoty se vzájemně podporují.

4.5 Faktory ovlivňující umístění ve vyhledávání

Vyhledávače určují umístění stránky na SERP podle několika faktorů. Tyto faktory by se daly rozdělit na on-page faktory (týkající se stránek samotných, jejich obsahu a kvality kódu) a dále na off-page faktory, které sledují pozici stránek v rámci jejich vztahu k okolí a vyhledávačům.

4.5.1 On-page faktory

- Validita webu
- Správná struktura a sémantičnost webu
- Umístění fráze nebo hledaného slova v titulku stránky, metaznačce Description, nadpisech
- Vzájemná poloha nalezených slov (u víceslovných frází)
- Vhodně použitá klíčových slov v nadpisech a zvýraznění pomocí tagů, titulku, metaznačce Description a nadpisech
- Váha stránky v očích vyhledávače (rank)
- Počet odkazů na stránky a kvalita těchto odkazů

4.5.2 Off-page faktory

- Text odkazů, které odkazují na web (anchor text)
- Stáří domény (starší jsou považovány za relevantnější)
- Obsahová relevance příchozích odkazů = odkazy z obsahově příbuzných stránek jsou hodnotnější
- Celková popularita stránky vyjádřená vysokým počtem odkazujících stránek (partnerské weby, katalogy...)

4.6 Google Analytics

Google Analytics je řešení webové analýzy pro podniky, které poskytuje dokonalý přehled o provozu na webových stránkách a o efektivitě marketingu. Díky výkonným, přizpůsobivým a snadno použitelným funkcím můžete nyní zobrazovat a analyzovat údaje o provozu zcela novým způsobem. Google Analytics vám umožní vytvářet lépe cílené

reklamy, posilovat marketingové aktivity a vytvářet webové stránky s vyššími mírami konverze [8]. Tato služba nabízí uživateli spoustu přehledů se statistikami. Tyto přehledy jsou použity v praktické části. Příloha [P I] obsahuje náhled a popis úvodního panelu a dalších přehledů. Základní implementace služby spočívá ve vložení kódu do hlavičky stránky. Tento kód je vygenerován při registraci webu do této služby.

4.7 Seo-servis.cz

Seo-servis.cz je webová služba poskytující kompletní analýzu on-page i off-page faktorů optimalizovaných stránek.

4.7.1 Služby

Mezi hlavní poskytované služby patří analýza:

- **Zdrojového kódu** - kontroluje validitu HTML, sémantičnost a obsah
- **Síly webu** - kontrola zdrojového kódu, kalkulace zpětných odkazů, analýza návštěvnosti a pozic ve vyhledávačích
- **Vyhledávačů** – informace o **umístění webu ve vyhledávačích na zadanou frázi**, počet zpětných odkazů, počet zaindexovaných stránek, aktuální Google Pagerank, S-rank, Alexarank, trendy ve vyhledávání
- **Klíčových slov** – analýza obsahu stránky, „optimalizovatelnost“ daného slova, návrhy na zlepšení, četnost a hustota frází

4.7.2 Podporované vyhledávače

- Google
- Bing
- Seznam
- Jyxo
- Yahoo!

4.7.3 Technické informace

- Skriptovací jazyk: PHP5
- Databáze: MySQL 5
- Webhosting: Váš-hosting.cz
- Autor: Karel Dytrych

II. PRAKTICKÁ ČÁST

5 PORTÁL PRO ZVEŘEJŇOVÁNÍ INZERCE

Internetové stránky www.inzertum.cz mají primárně sloužit:

- osobám hledajícím vozidlo za účelem jeho nákupu,
- osobám (inzerentům) prodávajícím vozidlo.

Vedlejším, ale nikoliv nezanedbatelným účelem těchto stránek je ovšem příjem z reklamy zde zobrazované a prodeji produktů z nabídky internetového autobazaru (zvýhodňování inzerce).

Proto je důležité, aby byla každá stránka webu plně přístupná uživateli a zároveň optimalizovaná pro fulltextové vyhledávače.

5.1 Hlavní stránky webu

Tato podkapitola se věnuje důležitým prvkům vybraných stránek a popisem jejich významu funkcí.

5.1.1 Úvodní strana portálu

Společně s partnerskou je úvodní strana hlavní vstupní stránkou webu tudíž ji vidí nejvíce potenciálních zákazníků.

Hlavními prvky úvodní strany webu jsou:

- Hlavička s odkazy a menu.
- Panel pro rychlý výběr a podrobnější vyhledávání.
- Panel s výpisem zvýhodněné inzerce.
- Panel s výpisem poslední inzerce.
- Boxy s informacemi pro uživatele.
- Panel pro bannerovou reklamu.
- Patička stránky s menu a výpisem partnerů webu.

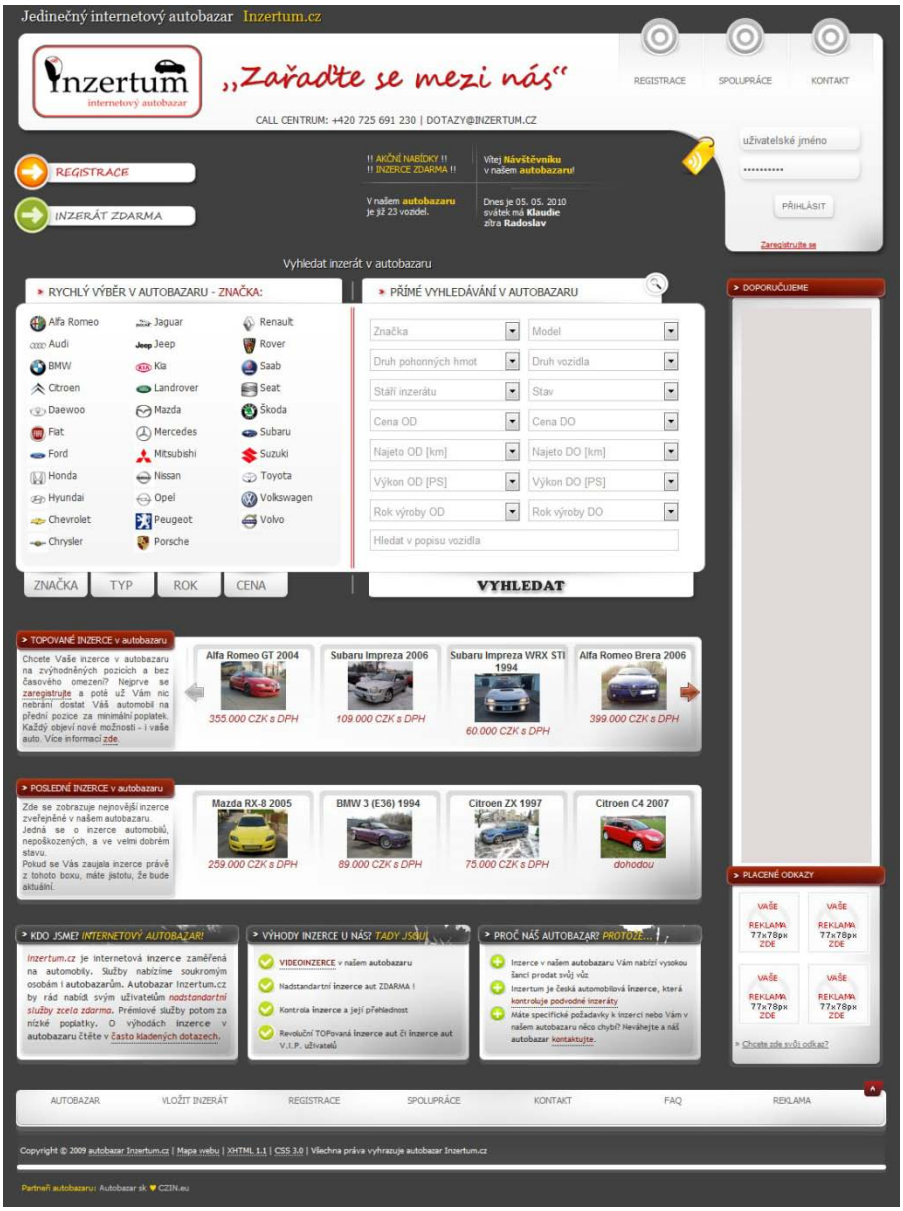
Pro uživatele i SEO optimalizaci je tato stránka nejdůležitější na celém webu. Navíc obsahuje odkazy na všechny podstránky webu. Nejdůležitější stránky na které odkazuje jsou:

- Stránka pro vkládání inzerátu
- Stránka pro registraci
- Partnerská stránka

- Stránka často kladených dotazů
- Stránka s kontakty

Na tyto stránky je odkazováno pomocí tlačítek v menu a textovými odkazy přímo v textu.

Vhodně napsaný obsah, ve kterém jsou obsažena a sémanticky správně zvýrazněna klíčová slova, je přínosem pro uživatele i vyhledávače.

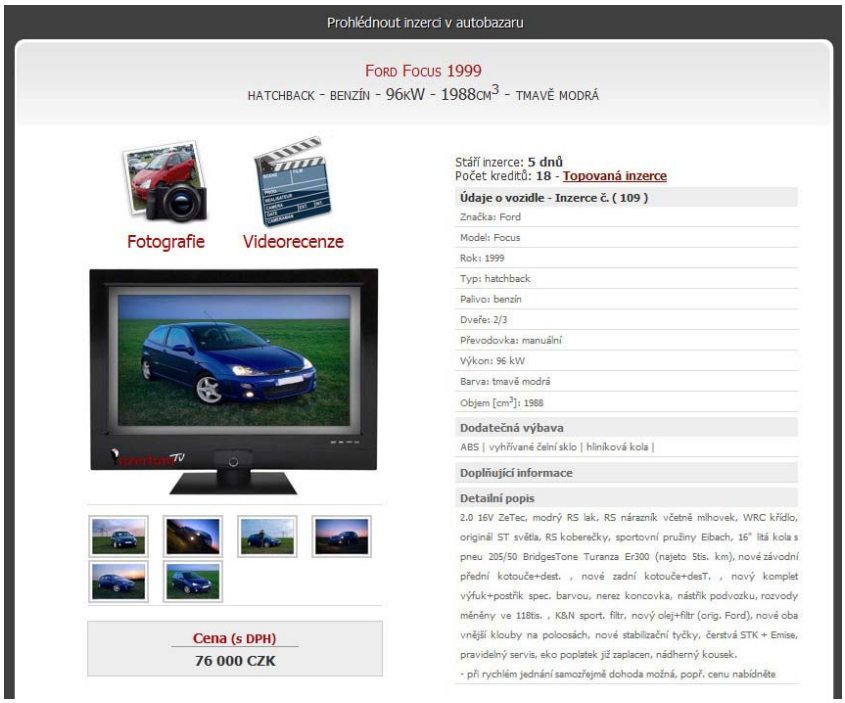


Obr. 10 – Úvodní strana webu

5.1.2 Detail inzerátu

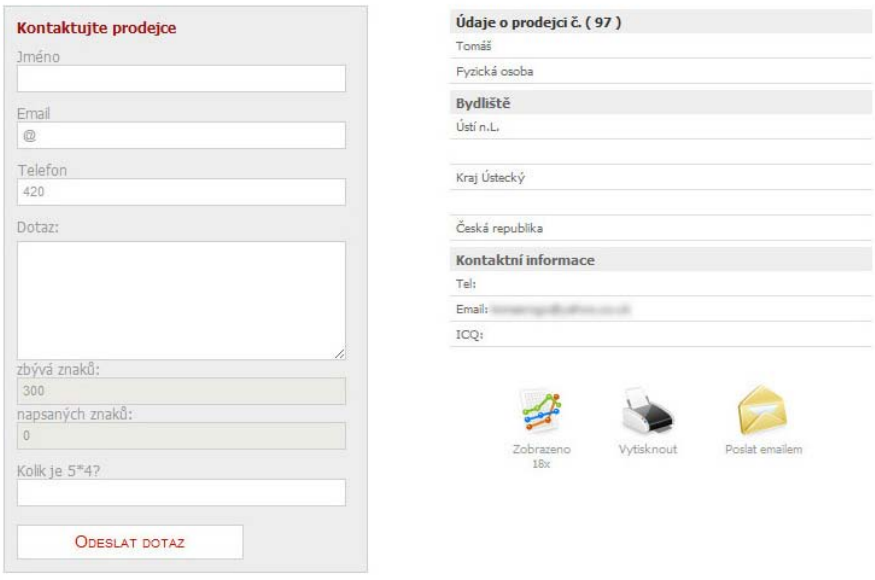
Stránka s detailem inzerátu je rozdělena do dvou částí.

První část (Obr.11) obsahuje informace o stáří inzerátu, informace o vozidle a poskytuje možnost prohlížet fotky nebo video přiložené k inzerátu.



Obr. 11 – Detail inzerátu - vozidlo

Druhá část (Obr.12) zobrazuje informace o prodejci vozidla a formulář pro rychlý dotaz. Jsou zde také ikony pro tisk, odeslání inzerátu emailem a počet zobrazení od zveřejnění.



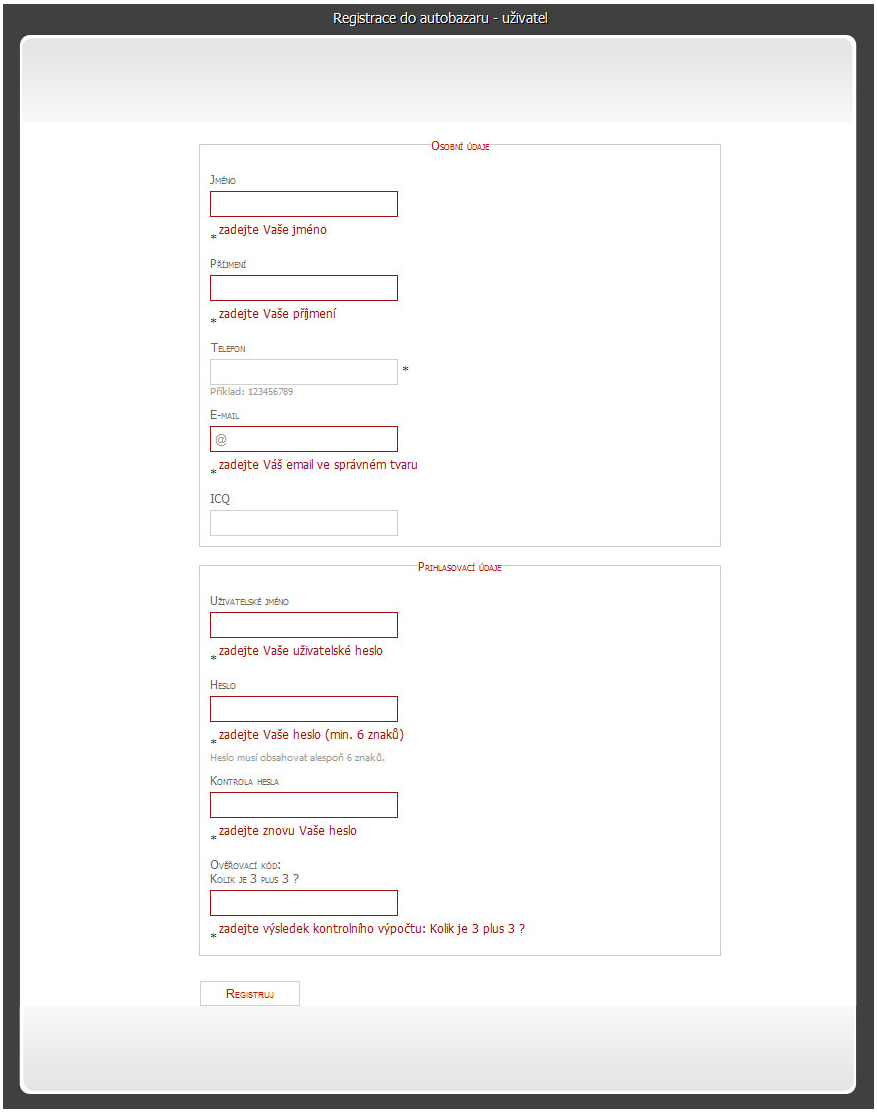
Obr. 12 – Detail inzerátu - prodejce

5.1.3 Registrace uživatele

Zde je registrační formulář, který je ošetřen proti vkládání jiného než povoleného typu vstupu. Typy vstupů:

- Text
- Email
- Celé číslo
- Potvrzení hesla jeho opětovným napsáním

Náhled stránky pro registraci s formulářem i s ukázkou jeho validace je na následujícím obrázku (Obr.13). Po dokončení registrace je na zadanou emailovou adresu odeslán potvrzovací email.



Obr. 13 – Registrační stránka s ukázkou validace vstupů

5.1.4 Vkládání inzerátů

Vložení a potvrzení inzerátu předchází pět kroků:

1. Vyplnění údajů o prodejci vozidla.
2. Vyplnění technických údajů o vozidle a jeho stavu.
3. Vložení fotek a videa.
4. Rekapitulace inzerátu.
5. Potvrzení inzerátu přes email.

5.1.5 Stránka často kladených dotazů

Tato stránka je velmi důležitá, protože má funkci tzv. manuálu webu. Navíc vysvětluje všechny pojmy použité v rámci webu a funguje i jako rozcestník.

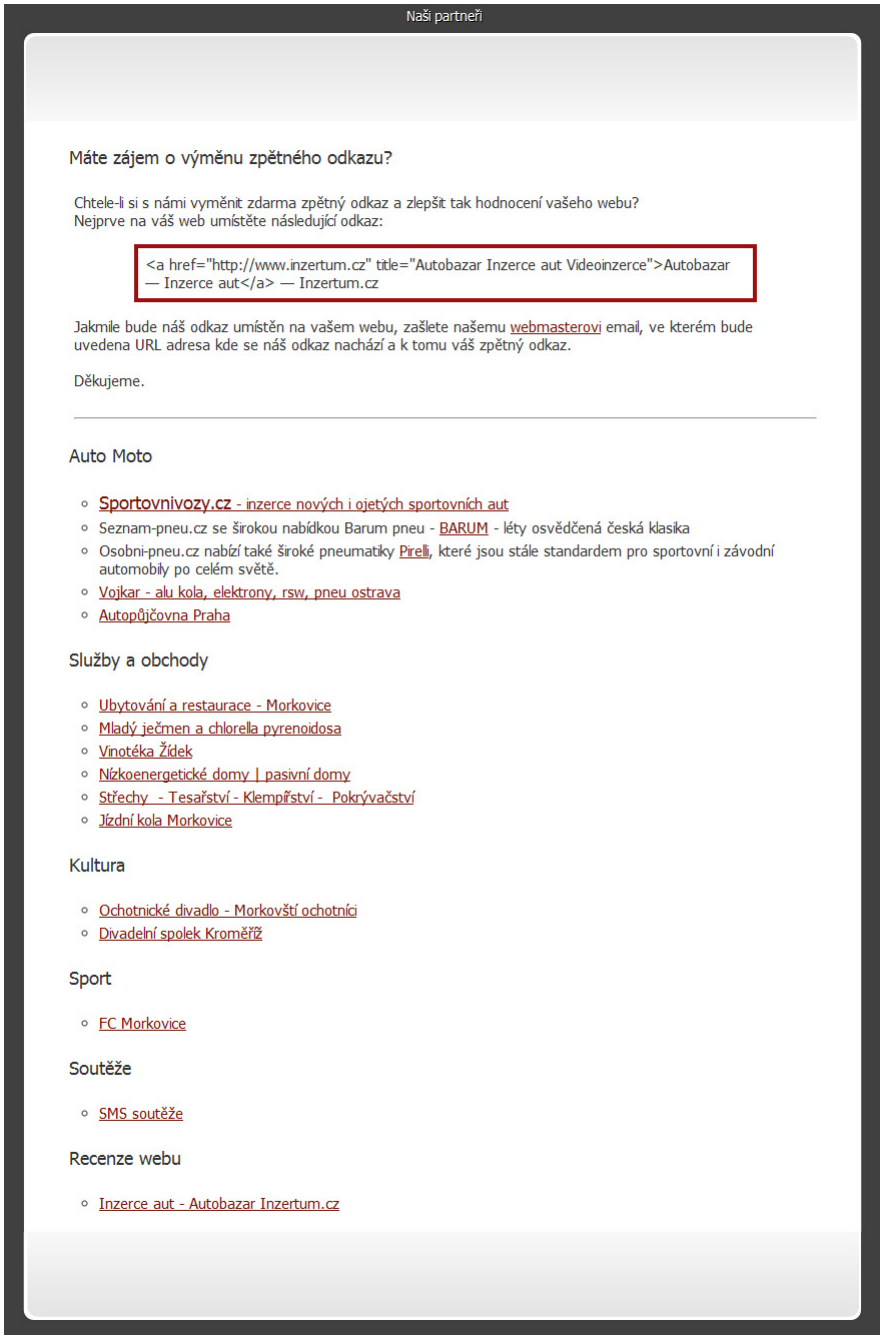
Stránka, která je uživatelsky přívětivá, má velký význam i pro optimalizaci pro vyhledávače. Je to ideální místo, kde je možné vhodně použít klíčová slova vybraná pro web a to ve velké hustotě.

5.1.6 Kontaktní stránka

Obsahuje dodatečné informace pro uživatele a odkaz na často kladené dotazy. Když ani v tomto případě uživatel nenajde informace které hledá, má možnost napsat osobně na vybraný email podle tématu dotazu , nebo může napsat dotaz přímo přes kontaktní formulář pro rychlý dotaz.

5.1.7 Partneři webu

Stránka partnerů webu (Obr.14) má z hlediska optimalizace pro vyhledávače velký význam. Jsou zde informace pro potenciální partnery o tvaru našeho odkazu a postupu při výměně zpětných odkazů. Dále už je zde výpis partnerů sdružených do příslušných kategorií.



Obr. 14 – Stránka pro partnery webu

5.2 Zabezpečení aplikace

Zabezpečení aplikace v podstatě spočívá v:

- Ošetření formulářových vstupů
- Správném čtení URL adres
- Potvrzování akcí přes emailovou adresu uživatele

5.2.1 Ošetření formulářových vstupů

O zabezpečení formulářů z větší části obstarává funkce `validaceVstupu()`, která vrací hodnotu 0 nebo 1, podle toho jestli vstupní hodnota odpovídá zvolenému typu. Využívá php funkce `ereg()`, která porovnává řetězec s regulárním výrazem. V textu jsou navíc povoleny znaky nezbytné pro správnou gramatiku. U čísel je nepovinné znaménko + a -.

```
function validaceVstupu($typ="text",$hodnota,$povinne=0){
 switch($typ){
   case "text":
    if($povinne==0){
return ereg("^[a-zA-Z0-9 ěščřžýáíéóúůďťňĚŠČŘŽÝÁÍÉÓÚŮĎŤŇ\?\,\.\:\*\-
\+]*$",$hodnota);
    } else {
return ereg("^[a-zA-Z0-9 ěščřžýáíéóúůďťňĚŠČŘŽÝÁÍÉÓÚŮĎŤŇ\?\,\.\:\*\-
\+]+$",$hodnota);
    }
    break;


  case "email":
     return ereg("^.+@.+\..+$",$hodnota);
    break;


  case "cislo":
   if($povinne==0){
     return ereg("^[\+\-]?[1234567890]*$",$hodnota);
   } else {
     return ereg("^[\+\-]?[1234567890]+$",$hodnota);
  }
   break;
 }
}
```

Text v popisu prochází před výpisem funkcí `htmlspecialchars()`, která převede poteciálně nebezpečné symboly na odpovídající entity. Je tím zaručeno, že se nebezpečný skript neprovede a odkazy jsou také převedeny do textové formy. Uživatel tím není nijak omezen při výběru znaků.

5.2.2 Ošetření URL adres

Dynamicky tvořená URL adresa pro inzerát má tvar například:

- `116-bmw-3-e36-1993-benzin-bila-2500cm3`

Takový řetězec zpracuje funkce `vratIdzUrl($url)` pomocí php funkce `preg_match_all()`. Následně je vráceno celé číslo, kterým řetězec začínal a zakončeno bylo znakem mínus.

```
function vratIdzUrl($url){
            $reg_vyraz = '#([0-9]+)-.*#';
            $r = preg_match_all($reg_vyraz,$url,$shody);
            return $shody[1][0];
           }
```

V tomto případě funkce vrací číslo id inzerátu 116. Vše ostatní za pomlčkou má význam pouze pro SEO optimalizaci a lepší orientaci uživatele.

5.2.3 Unikátní ID relace

Hodnota unikátního klíče určujícího uživatelskou relaci je generována jako pseudonáhodný řetězec pomocí `uniqid()` a `mt_rand()`. Pro případ, že by ve stejný okamžik přišlo více různých dotazů, je navíc zkombinován s IP adresou klienta. Z výsledného řetězce je následně vytvořen hash pomocí funkce `md5()`:

```
$hodnotaID = md5(uniqid(mt_rand()) . $_SERVER['REMOTE_ADDR']);
```

5.2.4 Potvrzování uživatelských akcí přes email

Na žádost zadavatele je součástí každé akce vedoucí k cíly některé z akcí potvrzení přes kontaktní email. Po dokončení registrace uživatele nebo inzerátu je odeslán potvrzovací email. Tento email obsahuje kompletní shrnutí informací o vozidle i o prodejci. Hlavním důvodem, proč je email uživateli posílán, je ověření zda uživatel zadal opravdu svoji emailovou adresu na kterou má přístup. Emailová adresa je v podstatě jediná povinná položka, která ověřuje identitu uživatele. V každém potvrzovacím emailu je odkaz, který obsahuje jedinečný identifikátor generovaný pro každou relaci. Po potvrzení přes tento odkaz je teprve příslušná akce nebo obsah uživateli přístupný.

5.3 SEO optimalizace portálu

SEO optimalizace zahrnuje analýzu zdrojového kódu, výběr klíčových slov, budování zpětných odkazů, dynamicky generované prvky stránek a průběžné výsledky optimalizace.

5.3.1 Analýza zdrojového kódu

Tab. 2 – Popisné informace stránky

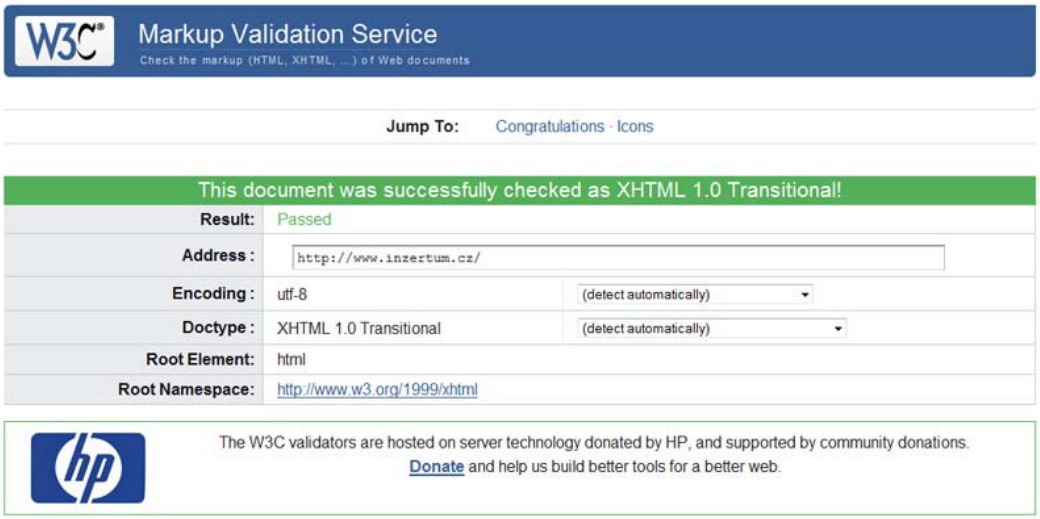
Titulek	Inzerce aut — Autobazar Inzertum.cz
Popis	Autobazar Inzertum.cz je zaměřen na inzerce aut a nabízí uživatelům nadstandartní služby jako jsou videoinzerce, korekce chyb a kontrola inzerátů.
Klíčová slova	autobazar, inzerce aut, videoinzerce
Info pro roboty	index,follow
Autor	Copyright (c) 2010 Martin Jurák
robots.txt	Existuje
Sitemap	www.inzertum.cz/sitemap.xml

Hlavička dokumentu:

- Deklarovaná definice typu dokumentu (DTD) **XHTML 1.0 Transitional**
- Deklarace znakové sady **utf-8**
- Titulek stránky je správně vyplněn
- Popisek stránky je správně vyplněn

Zdrojový kód:

- Zdrojový kód lehce překračuje optimální velikost. Příliš velká stránka zatěžuje vyhledávač stahováním přebytečných dat, a vyhledávač pak také těžko určuje, která část obsahuje je relevantní
- **Celková velikost HTML kódu:** 62 kB
- **Stránka** je **validní** podle deklarovaného XHTML 1.0 Transitional (viz. Obr.10)
- Stránka obsahuje inline vložené CSS styly, které by měly být ve zvláštním souboru. Velikost v CSS navíc: **0.09 kB**
- Stránka je **validní** podle **CSS3** (viz. Obr. 11)



Obr. 15 – Výsledek validace XHTML

Obr. 16 – Výsledek validace kaskádových stylů CSS verze 3

Sémantika a přístupnost:

- Stránka neobsahuje vnořené tabulky.
- Netextové elementy mají alternativní obsah.
- Na stránce je použito pouze správné sémantické zvýrazňování textu.
- Text je kvalitně strukturovaný do odstavců.

Obsahová část:

- Stránka obsahuje **právě jeden nadpis h1**
- Nadpisy na stránce jsou správně strukturované

Přehled nadpisů stránky.

- <h1> Jedinečný internetový autobazar
  - <h2>Vyhledat inzerát v autobazaru
    - <h3>RYCHLÝ VÝBĚR V AUTOBAZARU - ZNAČKA:
    - <h3>RYCHLÝ VÝBĚR V AUTOBAZARU - TYP:
    - <h3>RYCHLÝ VÝBĚR V AUTOBAZARU - CENA:
    - <h3>RYCHLÝ VÝBĚR V AUTOBAZARU - ROK:
    - <h3>PŘÍMÉ VYHLEDÁVÁNÍ V AUTOBAZARU
  - <h2>TOPOVANÉ INZERCE v autobazaru
  - <h2>POSLEDNÍ INZERCE v autobazaru
    - <h3>KDO JSME? INTERNETOVÝ AUTOBAZAR!
    - <h3>VÝHODY INZERCE U NÁS? TADY JSOU!
    - <h3>PROČ NÁŠ AUTOBAZAR? PROTOŽE...
    - <h3>DOPORUČUJEME
    - <h3>CHCETE ZDE SVŮJ ODKAZ?

Stránka obsahuje dostatek textu.

Počet odkazů na stránce: **193**

Počet odkazů na externí zdroje: **2**

5.3.2 Klíčová slova

Vzhledem k zaměření webu byly navrženy jako hlavní klíčová slova:

- autobazar
- inzerce

Tato vybraná slova potom byla vhodně použita a správně semanticky zvýrazněna v titulcích, nadpisech i textu každé stránky. Vyhledávače také sledují jejich sousední slova v textu a tímto se nabízí možnost vnoření slova mezi hledané výrazy. Tabulka (Tab.3) obsahuje údaje o četnosti výskytu těchto slov.

Tab. 3 – Výskyt a procentní zastoupení klíčových slov

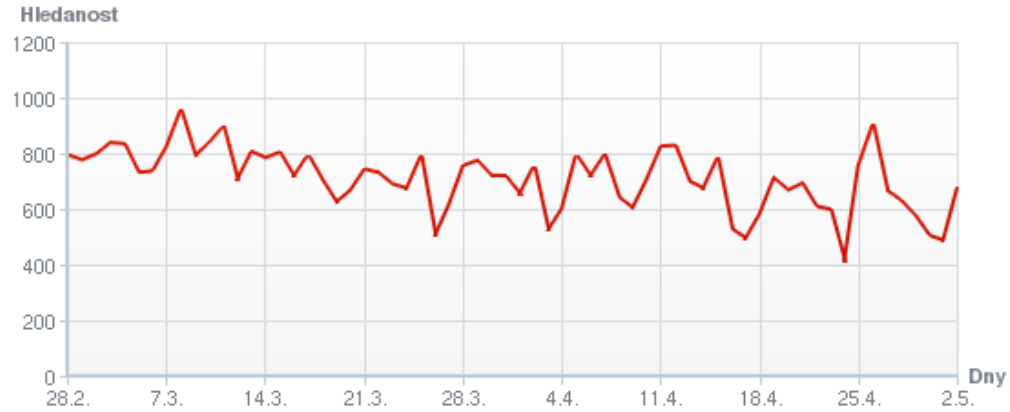
Slovo	Výskytů	Procentní zastoupení	Rank
autobazar	27	2.21 %	65
inzerce	19	1.55 %	43

Orientační počet slov na stránce je 1220.

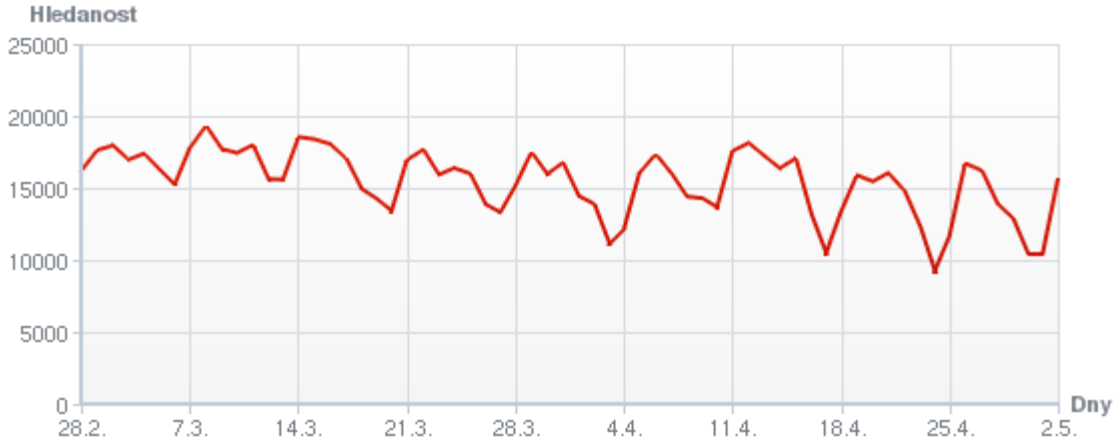
5.3.2.1 Statistiky hledanosti - Seznam

Vyhledávač Seznam.cz umožňuje zjistit objem hledanosti daného výrazu v určitém časovém období. Přitom tento výsledek nabízí pro rozšířenou shodu, kdy je slovo obsaženo ve hledaném řetězci a shodu přesnou (při hledání pouze tohoto výrazu). Tyto přehledy jsou dostupné pro objem návštěv vyšší než 100 denně. Následující grafy zobrazují objemy vyhledávání výrazů autobazar a inzerce.

Obr. 17 - Statistika hledanosti výrazu „autobazar“ (rozšířená shoda)



Obr. 18 - Statistika hledanosti výrazu „autobazar“ (přesná shoda)

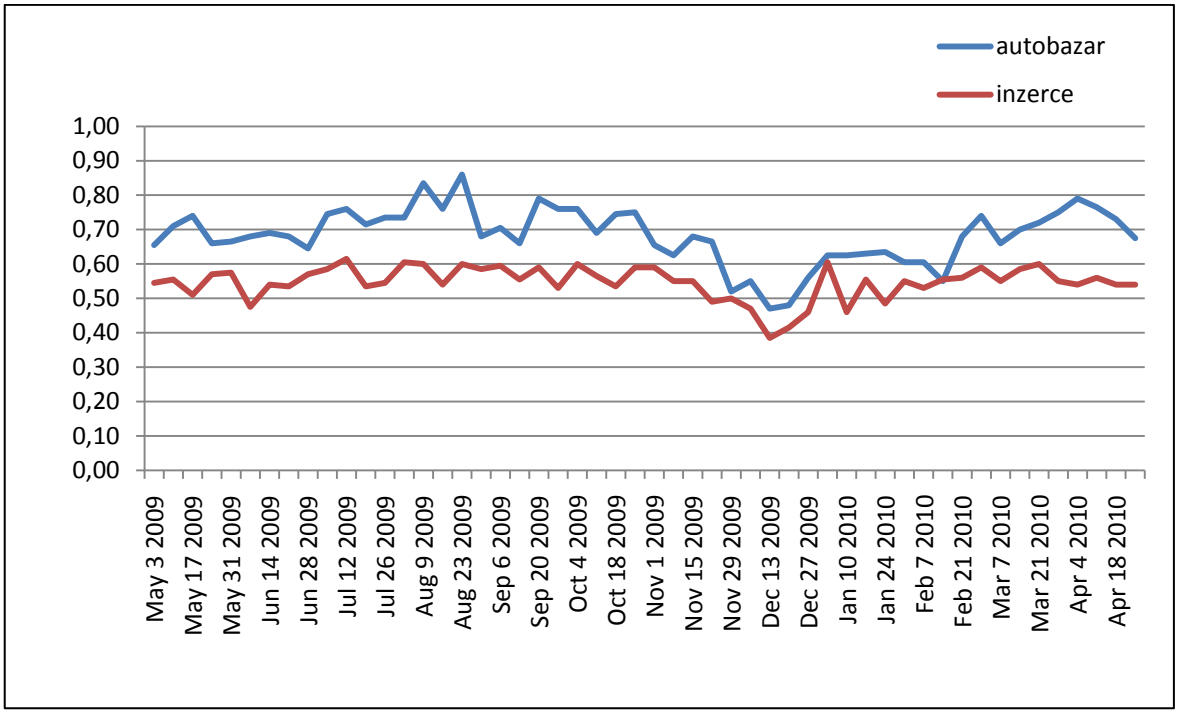


Obr. 19 – Statistika hledanosti výrazu „inzerce“ (rozšířená shoda)

Obr. 20 - Statistika hledanosti výrazu „inzerce“ (přesná shoda)

5.3.2.2 Trend hledanosti – Google

Google nabízí obdobnou službu jako Seznam, až na ten rozdíl, že zde se jedná o trend hledanosti. Ten je určen jako poměr k nejvyšší hodnotě (má hodnotu 1,00) ve sledovaném období.



Obr. 21 - Trend hledanosti výrazů „autobazar“ a „inzerce“

Z grafů (Obr. 17 - 21) lze vyčíst, že o vybraná klíčová slova je u uživatelů stálý zájem ve vyhledávání v Seznamu i v Google. Má tudíž význam stránky na tyto slova optimalizovat.

5.3.3 Robots.txt

Tento soubor upravuje přístup robotů vybraných vyhledávačů a je nutné jej nahrát do rootu domény. V tomto případě všem (*) robotům zakazuje procházet adresáře `/nazev/` a stránky začínající na `/nazev`. V neposlední řadě informuje roboty o umístění mapy webu generované do xml formátu.

```
User-agent: *
Disallow: /css/
Disallow: /design/
Disallow: /layout/
Disallow: /scripts/
Disallow: /vysledky-hledani
Disallow: /znacka
Disallow: /rok
Disallow: /typ
Disallow: /cena
Sitemap: http://www.inzertum.cz/sitemap.xml
```

5.3.4 Dynamické titulky a URL adresy

5.3.4.1 Modul mod_rewrite

Modul mod_rewrite serveru Apache je nástroj pro vytváření složitých dynamických adres, které jsou navíc vstřícné k vyhledávačům. Umožňuje vytvořit sadu pravidel, která Apache za běhu používá k přemapování URL, o kterou požádal návštěvník stránky, na dynamický dotaz s dotazovanou částí pro různé skripty PHP. Co se ovšem týká indexovacích robotů vyhledávačů, URL jsou statické.

5.3.4.2 Soubor .htaccess

Konfiguraci mod_rewrite pomocí souboru .htaccess je vhodné použít v případě že:

- nemám přístup ke globálnímu konfiguračnímu souboru httpd.conf,
- chci konfigurovat nastavení pro konkrétní adresář,
- chci mít možnost měnit konfigurace bez nutnosti restartování Apache.

```
RewriteEngine On
RewriteBase /

RewriteCond %{HTTP_HOST} !www\.inzertum\.cz
RewriteRule ^(.*)$ http://www.inzertum.cz/$1 [R=301]

RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^([^/]+)/([^/]+)?/?$ ?sekce=$1&podsekce=$2 [L]

RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^([^/]+)/?$ ?sekce=$1 [L]
```

Nejdříve je nutno aktivovat přepisování, poté je vhodné stanovit výchozí adresu pro relativní URL, protože by mohlo docházet k chybám při načítání CSS stylů nebo obrázků s relativní adresou zdroje.

První pravidlo zápisu obstarává automatické přesměrování adresy `domena.cz` na `http://www.domena.cz/`. Nedochází tak k duplicitě, která by mohla vést k penalizacím od vyhledávačů.

Následující podmínky a přepisovací pravidla umožňují používat URL adresu ve tvaru:

- `http://www.inzertum.cz/detail/116-bmw-3-e36-1993-benzin-bila-2500cm3`

Kde `/detail` je název sekce a `/116-bmw-3-e36-1993-benzin-bila-2500cm3` je název podsekce (v tomto případě konkrétní inzerát).

Bez zapnutého modulu mod_rewrite by funkční URL adresa musela vypadat takto:

- `http://www.inzertum.cz/?sekce=detail&podsekce=116-bmw-3-e36-1993-benzin-bila-2500cm3`

Takový tvar adresy obsahující znaky ? a & méně vyhovuje indexovacím robotům vyhledávačů.

5.3.4.3 Dynamické prvky stránky

URL adresa je složena z ID inzerátu, za kterým následují klíčová slova tvořena z údajů vložených inzerentem. Jsou to údaje dostupné v základní dokumentaci k vozidlu, kterou by měl prodejce mít. Výsledná adresa je nejen srozumitelnější pro uživatele, ale stejně jako titulek stránky a nadpisy má značnou váhu pro vyhledávače. Následující tabulka (Tab.2) obsahuje příklad dynamicky generovaných prvků konkrétního inzerátu.

Tab. 4 – Ukázka dynamických prvků stránky inzerátu

URL adresa inzerátu	ID inzerátu	Titulek stránky	Nadpis, odkaz, popisky
116-bmw-3-e36-1993-benzin-bila-2500cm3	116	BMW 3 (E36) 1993 - Inzerce aut - Autobazar Inzertum.cz	BMW 3 (E36) 1993

5.3.5 Linkbuilding

Linkbuilding neboli budování zpětných odkazů přichází na řadu ve chvíli kdy jsou stránky vytvořené, naplněné hodnotným obsahem a splňují všechny pravidla přístupnosti a on-page faktorů. Linkbuilding je v některých případech podstatnější část SEO optimalizace a to konkrétně off-page optimalizace.

Nabízí se dva způsoby jak zpětné odkazy získat:

- Nákupem
- Výměnou

Další způsoby jak odkazy získávat:

- Přirozeným způsobem
- Registrací do katalogů
- Pomocí PR – tiskové zprávy články, publikování na cizích webech
- Účastí ve fórech a diskuzích
- Publikováním na blogu
- Publikováním v záložkovacích serverech
- Na inzertních serverech
- Černými praktikami – spam, komentářový spam, diskuzní spam, viry atd.

5.3.5.1 Katalogy

Stránky byly zaregistrovány do nejvýznamějších katalogů.

Mezi tyto katalogy patří:

- Firmy.cz
- Najisto.cz
- Dmoz.org
- Yahoo!

5.3.6 Vyhledávače

Po každé podstatné změně, kterou může být změna obsahu nebo výměna odkazů, byla příslušná stránka nebo hlavní stránka webu odeslána robotům.

Adresy pro přidání stránek do jednotlivých vyhledávačů jsou následující:

- Google.com - http://www.google.com/addurl/
- Seznam.cz - http://fulltext.seznam.cz/

- Centrum.cz - http://kat-reg.centrum.cz/reg.php
- Jyxo.cz - http://jyxo.cz/d/submit
- MSN.com (Windows Live Search) - http://search.msn.com/docs/submit.aspx

5.3.7 Průběžné výsledky optimalizace

Hodnoty v tabulce (Tab.5) byly měřeny maximálně do šedesáté pozice ve vyhledávání. (znaménko – mínus znamená, že výraz nebyl nalezen)

Datum měření: 11.5.2010.

Tab. 5 – Tabulka umístění stránek pro konkrétní výrazy

	inzerce aut		inzerce automobilů		autobazar inzerce		autobazar inzerce aut	
Vyhledávač	Pozice	Strana	Pozice	Strana	Pozice	Strana	Pozice	Strana
Seznam	12	2	40	4	48	5	6	1
Google	31	4	-	-	20	2	6	1
Jyxo	26	3	-	-	-	-	-	-
Bing	1	1	24	3	2	1	2	1

Výše uvedené hodnoty jsou velkou měrou ovlivněny hodnotami ranků a to především PageRanku společnosti Google a S-ranku od Seznamu. Alexa rank se odvíjí od návštěvnosti webu (vyšší návštěvnost = nižší hodnota). Hodnoty těchto ranků jsou zaznamenány v následující tabulce (Tab.6).

Tab. 6 – Hodnoty jednotlivých ranků vyhledávačů

Rank	Hodnota
S-rank	60/100
PageRank	0/10[1]
Alexarank	23.880.008

[1] Hodnota PageRank dosud nebyla přidělena.

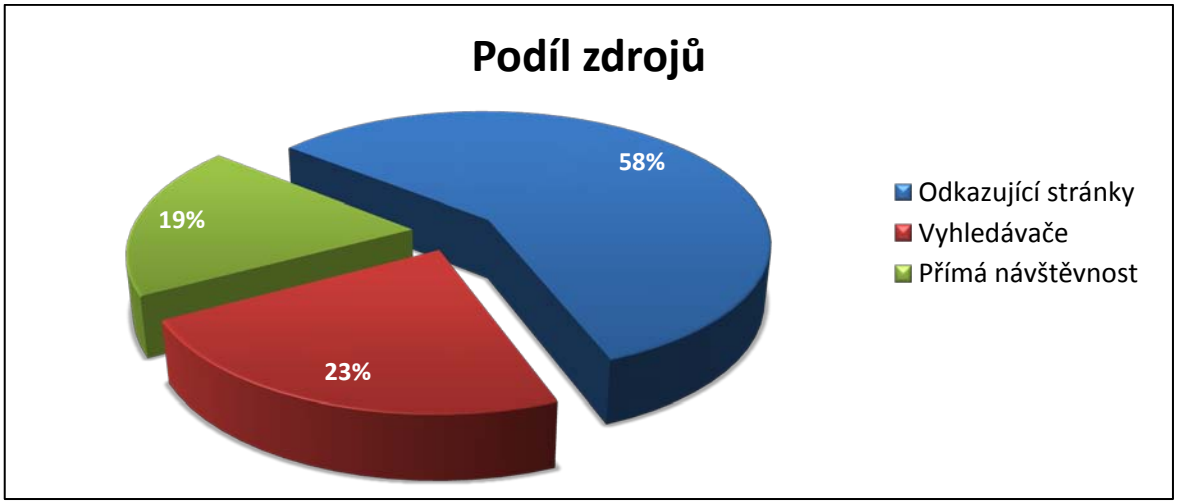
K prezentaci výsledků optimalizace byly dále využity hlavní přehledy služby Google Analytics:

- Počet návštěv ve sledovaném období
- Podíl zdrojů návštěvnosti podle typu
- Nejvýznamnější zdroje návštěvnosti
- Návštěvy přes hledaná klíčová slova

První graf (Obr. 22) z přehledu Google Analytics znázorňuje počet návštěv každý den od spuštění stránek tj. v období od 5.4.2010 do 17.5.2010.

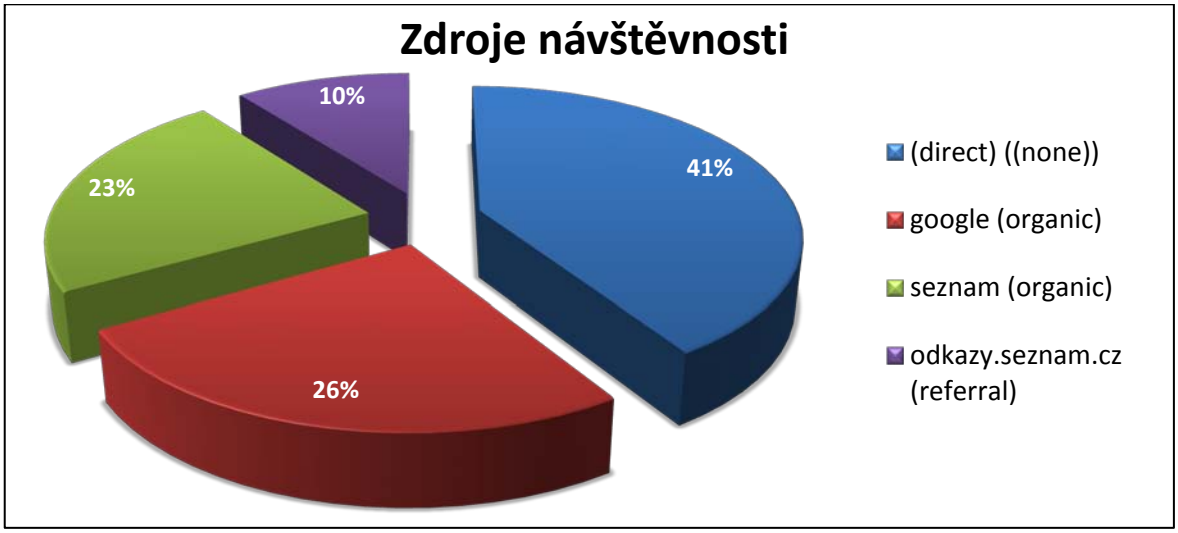
Obr. 22 – Graf návštěv webu za vybrané období

Návštěvník může přijít na stránku z několika základních zdrojů. Buď zadá přímo URL adresu do adresního řádku internetového prohlížeče (přímá návštěvnost), nebo klikne na odkaz na stránce, kde je umístěn odkaz na naše stránky (partnerské stránky, katalogy). Poslední způsob, kterým se může uživatel ocitnout na našich stránkách je přes vyhledávač při zadání hledaného výrazu (vyhledávače). Podíl zastoupení těchto tří kategorií je znázorněn v grafu podílu zdrojů (Obr.23).



Obr. 23 – Graf podílu hlavních zdrojů návštěvnosti

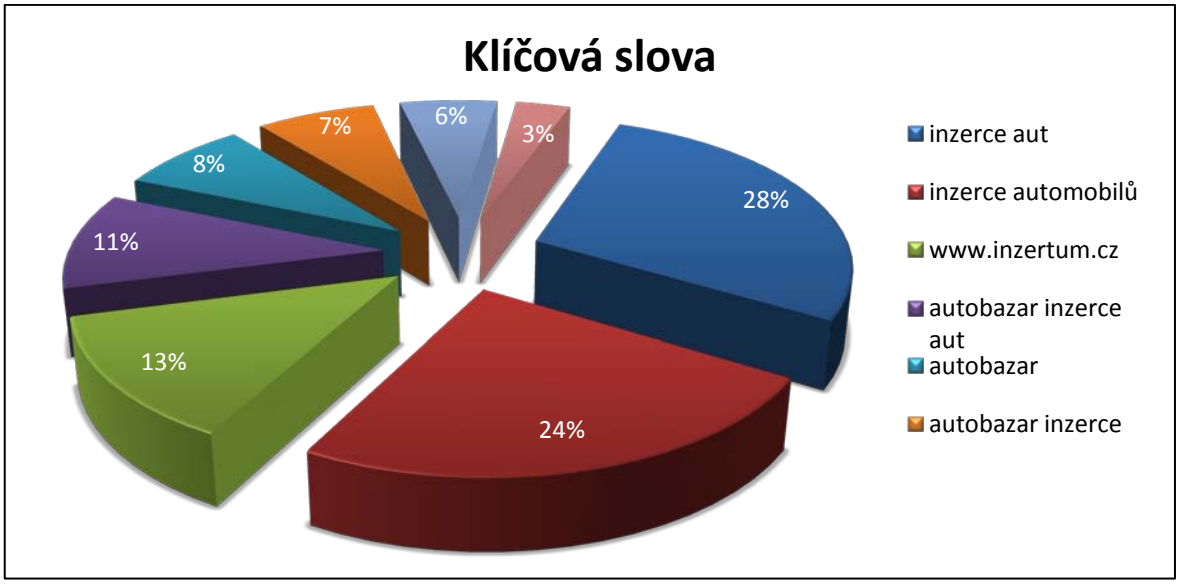
Podrobnější pohled na zdroje návštěvnosti nabízí graf nejvýznamnějších zdrojů návštěvnosti (Obr.24).



Obr. 24 – Graf podílu konkrétní zdroje návštěvnosti

Z grafu je patrné, že nejvýznamnější podíl na návštěvnosti mají vyhledávače (Seznam, Google) a přímá návštěvnost. Následuje katalog na Seznam.cz, do kterého byly stránky také vloženy do příslušné kategorie Auto-Moto - Bazary – Inzertní servery.

V případě, že přijde návštěvník přes odkaz na výsledkové stránce vyhledávače je sledováno klíčové slovo nebo fráze, kterou zadal jako vyhledávací výraz.



Obr. 25 – Rozdělení návštěv přes hlavní klíčová slova (hledané výrazy)

Graf klíčových slov (Obr.25) vychází z výsledků umístění stránek (Tab.5). Pro výrazy „inzerce aut“ a „autobazar inzerce aut“ je stránka umístěna na lepších pozicích než u ostatních (zhruba do druhé stránky výsledků).

Následující graf (Obr.26) je podrobnějším výpisem návštěv optimalizovaných stránek přes vyhledávaná klíčová slova. Jsou zde vypsány všechny vyhledávané výrazy přes které přišel návštěvník na naše stránky. Mimo návštěvy přes vybraná klíčová slova je v grafu patrné, že je pro optimalizaci důležitá univerzálnost stránek, která vychází z dynamických prvků stránky. Skutečnost, že se v titulcích, nadpisech a URL adresách objevují základní informace o vozidle typu Značka-Model-Rok se projevila na návštěvách při vyhledávání výrazů jako například:

- fabia rs 2006
- prevodovka fiat uno
- xenony fabia rs

Výpis hledaných výrazů vedoucích k návštěvě obsahuje následující graf (Obr.26).

Obr. 26 – Kompletní výpis návštěv přes hledané výrazy

5.3.8 Cílové konverze

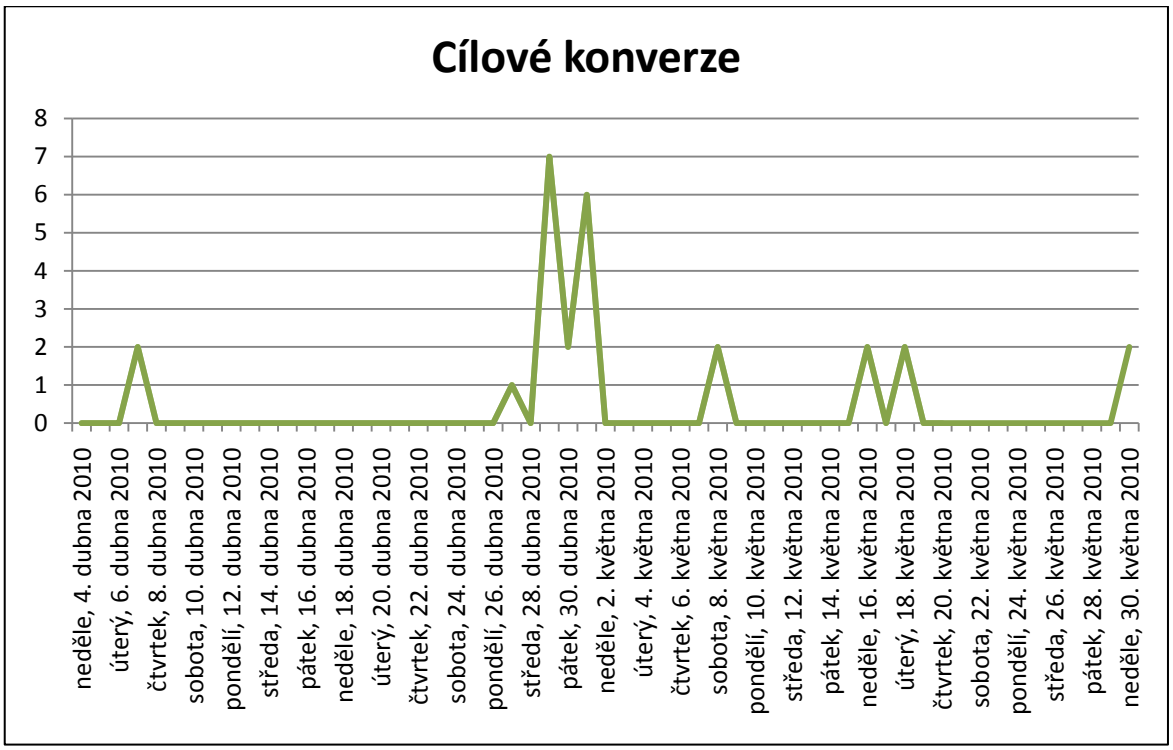
Neméně podstatným ukazatelem, kromě objemu návštěv, je míra cílové konverze. Měření počtu cílových konverzí totiž představuje reálný způsob jakým lze zjistit, zda je webová stránka poutavá pro návštěvníky.

Sledování specifických URL adres jako například:

- http://www.inzertum.cz/inzerat-vlozen
- http://www.inzertum.cz/registrace-dokoncena

spouští zaznamenání cíle. Tyto stránky zpravidla obsahují dodatečné informace pro uživatele, který již prošel konverzí. V tomto kontextu je cíl synonymem ke konverzi.

Jeden z cílů je například úspěšné vložení inzerátu. V případě, že uživatel vstoupí na náš web a vloží pět inzerátů, cílová konverze bude jedna. Je to logické, protože návštěvník může konvertovat pouze jednou.



Obr. 27 – Počet cílových konverzí „Vložení inzerátu“.

V případě vkládání inzerátu vede cesta k úspěšnému cíli přes dva kroky:

1) dokončení inzerátu - stránka Rekapitulace inzerátu (následuje odeslání potvrzovacího emailu)
2) potvrzení inzerátu přes potvrzovací email – inzerát následně čeká na zveřejnění administrátorem webu

Výhoda spočívá v tom, že stačí sledovat kdy je konverze liché číslo. Je to ukazatel neúplného dosažení cíle. Nejčastější příčina je nepotvrzení akce přes email zadaný při registraci nebo vkládaní uživatelských informací. Někteří uživatelé zadávají fiktivní email a na tom ztroskotají. Viz následující graf (Obr.27), kde u 27.4.2010 a 29.4.2010 jsou liché hodnoty, což znamená nedokončení cesty.

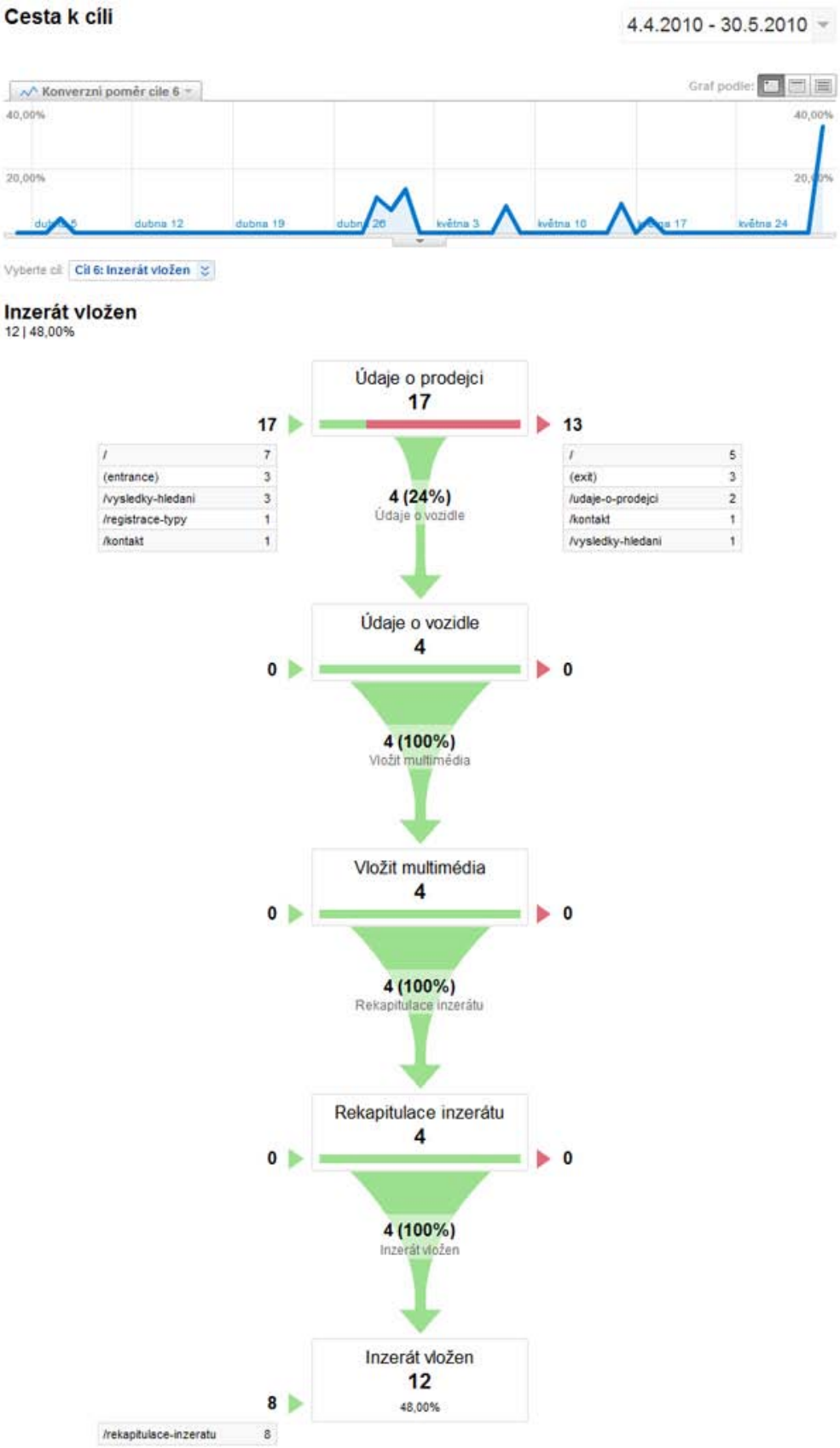
5.3.9 Vizualizace cesty k cíli

Pro důkladnou analýzu se nabízí vizualizace cesty k cíli jako trychtýř. Využívá se u cílů, které mají jasně definované cesty, po kterých se návštěvníci ubírají k cíli. Vizualizace cesty umožňuje zhodnotit, jak si stránky vedou při přesvědčování, neboli jak se jim daří přimět návštěvníky pokračovat k dalšímu kroku definované cesty, čímž se dostávají blíže ke konverzi.

Zjevným příkladem je proces vložení inzerátu. V tomto případě jsou stránky, přes které musí návštěvník projít jsou:

- údaje o prodejci
- údaje o vozidle
- vložit multimédia
- rekapitulace inzerátu
- inzerát vložen,

kde vstupní stránka je „Údaje o prodejci“ a cíl je stránka „Inzerát vložen“. V ideálním případě nemaji stránky cesty k cíli žádné výstupní stránky (napravo od jednotlivých sekcí vedle červené šipky). V případě „hladké cesty“, kdy uživatel projde právě přes dané stránky, nemají ani vstupní stránky. Toto se ovšem netýká vstupní stránky „Údaje o prodejci“. Následující obrázek (Obr.28) představuje grafickou vizualizaci cesty k cíli „Inzerát vložen“.



Obr. 28 – Vizualizace cesty k cíli

Z obrázku (Obr.28) je patrné, že nejvíce závisí na rozhodnutí uživatele přímo na vstupní stránce (Údaje o prodejci). Pokud se dostane k následujícímu kroku (Údaje o vozidle), tak už dokončí celý inzerát a dojde cíli (konverzi). Hodnota 8 u vstupu ke stránce „Inzerát vložen“ je způsobena tím, že tato funkce (Vizualizace cesty jako trychtýř) byla spuštěna s časovým zpožděním. Je to nepodstatný rozdíl, protože procento konverze mezi stránkami zůstane 100. Naopak by to kladně ovlivnilo procento konverze vstupní stránky (Údaje o prodejci).

5.3.10 Vyhodnocení výsledků optimalizace a návrh řešení

V době psaní této práce jsou stránky v provozu sedm týdnů, což se výrazně projevilo na výsledcích. Vzhledem k tomu, že od spuštění webu dosud neproběhl přepočet ranků u vyhledávače Google, tak zatím nebyl přidělen PageRank. S-rank vyhledávače Seznam.cz vzhledem ke kratším intervalům mezi přepočty ranků přidělen byl. Hodnoty kolísaly vždy po přidání do katalogu nebo na partnerskou stránku (hodnota se zvyšovala ve skocích 0, 10, 30, 60), až do té doby než se projevily změny na našich stránkách (patička s partnery, partnerská sekce webu). Hodnota S-rank je aktuálně 60/100.

V oblasti on-page faktorů už není tolik prostoru pro optimalizaci. Zdrojový kód je platný pro standard XHTML 1.0 Transitional a kaskádové styly verze 3 (CSS3). Sémantika webu je také v pořádku. Dynamicky generované prvky každé stránky jakou jsou titulky, nadpisy i URL adresy konkretních inzerátů jsou přínosem pro přístupnost stránek i optimalizaci pro fulltextové vyhledávače. Důkazem toho jsou data naměřená pomocí služby Google Analytics a Seoservis.cz.

Zásadní pro budoucnost umístění optimalizovaných stránek bude určitě budování zpětných odkazů a to zejména výměna odkazů s lépe hodnocenými partnerskými weby (umístění textového odkazu v patičce webu). Samozřejmě byl měl být výběr cílený na tématicky příbuzné weby. V našem případě v oblasti Auto-Moto (servisy, půjčovny vozidel, autobazary...). Po každé významné změně je nutné vložit odkaz na danou stránku každému z vyhledávačů. V tuto chvíli probíhá jednání s Centrum.cz. Nabídka se týká garantovaného počtu návštěvníků za určité časové období.

ZÁVĚR

V souladu se zadáním práce byla nastudována problematika bezpečnosti webových stránek a jejich optimalizace pro vyhledávače a vytvořen dokument tuto problematiku mapující. Při studiu současné situace bylo zjištěno, že bezpečnost internetových aplikací je v dnešní době podceňovanou hrozbou mezi uživateli i vývojáři. Spousta lidí si myslí, že k zabezpečení webového místa stačí dobrý firewall, šifrovaná komunikace a bezpečnostní záplaty. Naneštěstí tomu tak není, protože většina webových stránek obsahuje kód, který z něj činí dynamický web. A celkem často tento kód píší programátoři, kteří si myslí, že zabezpečení je úkolem správců systému. Důsledkem toho je velké množství bezpečnostních trhlin, které činí webovou aplikaci zranitelnou. A těchto trhlin využívají hackeři, kteří mohou snadno získat citlivá osobní data uživatelů a poté je například prodávat na černém trhu. Část práce je tedy zaměřena jednotlivě na nejrozšířenější útoky. Těmito útoky jsou Cross-site scripting, SQL Injection, Cross-site request forgery a Directory traversal. Dále jsou zde popsány základní zásady práce a používání hesel ať už z hlediska vývojáře, tak i uživatele samotného. Věnuje se také nejpoužívanějším šifrovacím metodám používaných ve webových aplikacích.

Součástí práce je i webová aplikace pro zveřejňování inzerce. Vytvořená aplikace je detailně popsána v praktické části. Na této webové aplikaci jsou popsány konkrétní způsoby zabezpečení proti výše uvedeným útokům.

Dále tato aplikace slouží pro názornou demonstraci optimalizace webových stránek pro vyhledávače. Všechny faktory, kterými se SEO neboli optimalizace pro vyhledávače zabývá jsou, popsány v teoretické části práce. Konkrétní způsoby aplikace SEO jsou popsány v části praktické. Průběžné výsledky optimalizace byly zaznamenávany službou Google Analytics a Seoservis.cz. Sledované statistiky se převážně týkají struktury a zdrojů návštěvnosti pro Google Analytics, v případě Seoservis.cz v podobě analýzy zdrojového kódu a hodnot ranků ovlivňijcí výrazně umístění. Z analýzy naměřených výsledků vyplívá, že optimalizovaná aplikace má dobrou perspektivu pro dobré výsledky ve vyhledávačích. Nutností je ovšem výměna zpětných odkazů s tématicky podobnými weby s vysokými hodnoceními, protože v oblasti on-page faktorů již není tolik prostoru pro optimalizaci.

CONCLUSIONS

In accordance with the specifications of the work was staged safety issues to their website and search engine optimization and a documentary charting the issue. In studying the current situation revealed that the safety of internet applications today is underestimated threat among users and developers. Many people think that the security of the site just a good firewall, encrypted communication and security patches. Unfortunately this is not so, because most websites contain code that makes it a dynamic website. And quite often the code written by programmers who think that security is the responsibility of system administrators. As a result a large number of security fissures, which makes Web application vulnerable. And these fissures using hackers who could easily obtain sensitive personal data of users and then as sold on the black market. Part of the work is focused individually on most attacks. These attacks are cross-site scripting, SQL Injection, Cross-site request forgery and Directory Traversal. It further describes the basic principles of work and the use of passwords, both in terms of developers and users alone. He is also the most widely used encryption methods used in Web applications.

The work also includes a web application for the publication of advertisements. Created application is described in detail in the practical part. On this web application describes specific methods for securing against the above attacks.

Furthermore, this application is used for visual demonstration of optimizing websites for search engines. All factors of SEO or search engine optimization addresses are described in the theoretical part. Specific options for SEO are described in the practical part. Continuous optimization results were recorded and Google Analytics Seoservis.cz. Observed statistics are mainly concerned with the structure and resource sites for Google Analytics, where Seoservis.cz in the form of source code analysis and value ranks influencing significantly location. An analysis of the measured results vyplívá that optimized applications has good prospects for good results in search engines. But necessity is the exchange back links with sites similar theme with a high rating because of on-page factors are no longer much room for optimization.

SEZNAM POUŽITÉ LITERATURY

[1] CASTRO E. HTML, XHTML a CSS Názorný průvodce tvorbou WWW stránek. Computer Press 2007, ISBN: 978-80-251-1531-2.

[2] NARAMORE E. Vytváříme webové aplikace v PHP5, MySQL a Apache. Computer Press 2006, ISBN: 80-251-1073-7.

[3] HUSEBY S. Zranitelný kód. Computer Press 2006, ISBN: 80-251-1180-6.

[4] LAVIN P. PHP - objektově orientované. Computer Press 2009, ISBN: 978-80-247-2137-8.

[5] KUBÍČEK M. Velký průvodce SEO. Computer Press 2008, ISBN: 978-80-251-2195-5.

[6] Cross-site request forgery. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 12.5.2010, last modified on 12.5.2010 [cit. 2010-05-20]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Cross-site_request_forgery>.

[7] Directory traversal. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 28.7.2005, last modified on 24.4.2010 [cit. 2010-05-23]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Directory_traversal>.

[8] *Google Analytics - Official Website* [online]. 2010 [cit. 2010-05-28]. Http://www.google.com/intl/cs/analytics/. Dostupné z WWW: <http://www.google.com/intl/cs/analytics/>.

[9] Sirovich J. SEO v PHP. Computer Press 2008, ISBN: 978-80-251-2083-5.

[10] České vysoké učení technické v Praze. Katedra telekomunikační techniky. *Access Server* [online]. 15. 08. 2007 [cit. 2010-05-28]. Zabezpečení webových aplikací I. - klientské skriptovací jazyky. Dostupné z WWW: <http://access.feld.cvut.cz/view.php?cisloclanku=2007090001>. ISSN 1214-9675.

[11] HAVRLANT, Lukáš. *Sémantika - pravý význam html značek* [online]. 2007 [cit. 2010-05-28]. Dostupné z WWW: <http://semantika.name/>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

WWW	World Wide Web
HTML	HyperText Markup Language
XHTML	Extensible HyperText Markup Language
HTTP	HyperText Transfer Protocol
URL	Uniform Ressource Locator
PHP	Hypertext Preprocesor
CSS	Cascading Style Sheets
SQL	Structured Query Language
W3C	The World Wide Web Consortium
XSS	Cross-site scripting
SID	Session ID
MD5	Message-Digest algorithm
SHA	Secure Hash Algorithm
SERP	Search Engine Result Page
SEM	Search Engine Marketing
SEO	Search Engine Optimization
XSS	Cross-site scripting
CSRF	Cross-site Request Forgery

SEZNAM OBRÁZKŮ

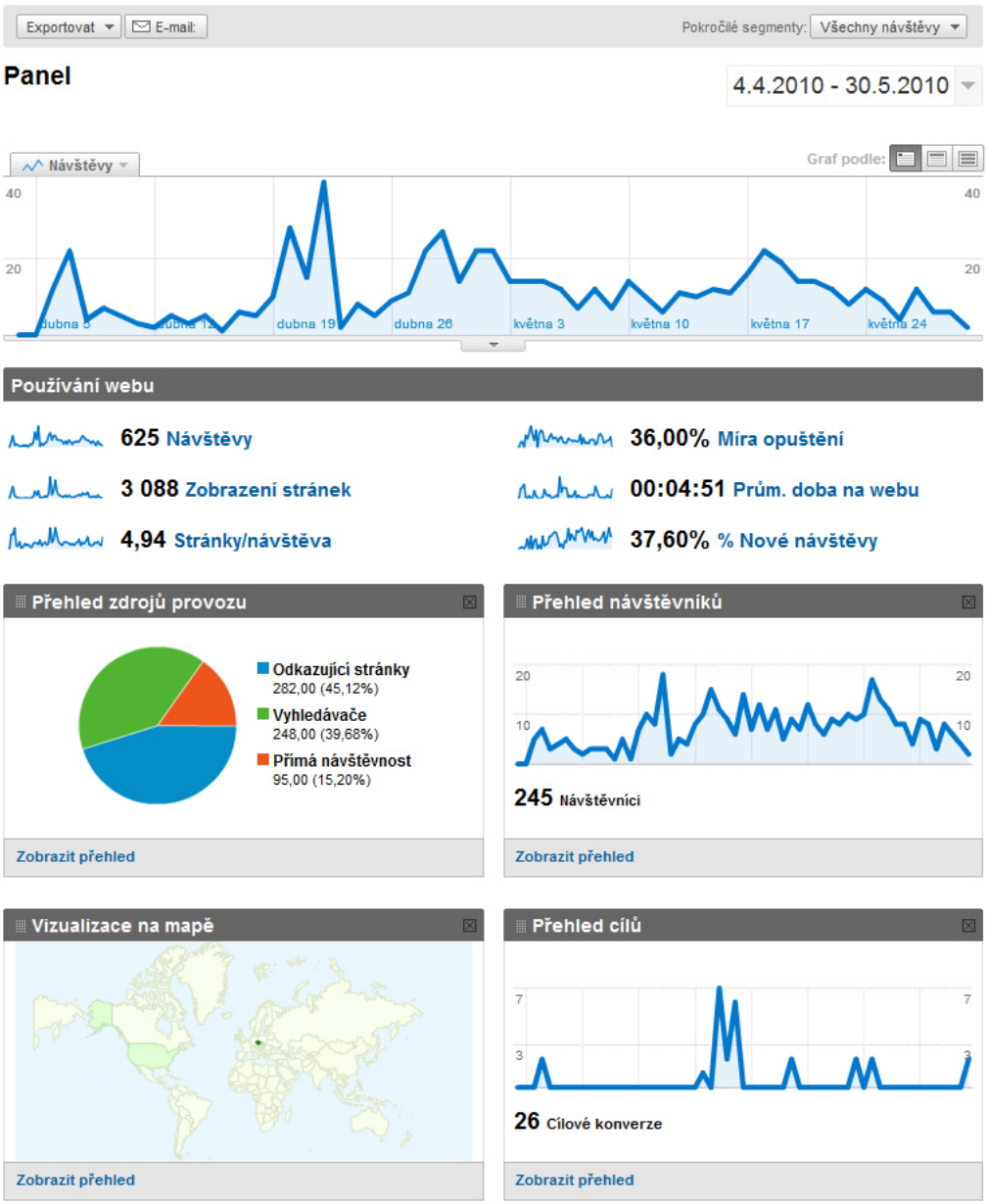
SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: PANELY SLUŽBY GOOGLE ANALYTICS

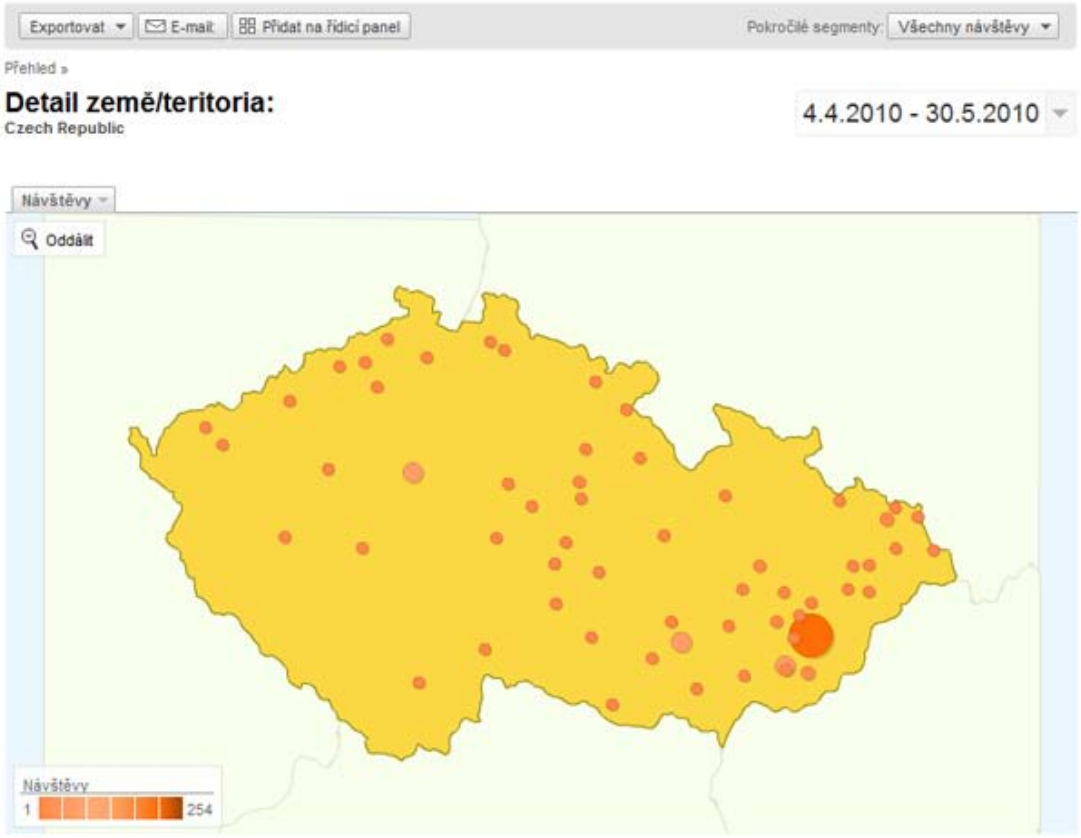
Služba Google Analytics nabízí uživateli monitoring stránek přes různé přehledy v podobě panelů. Na následujícím obrázku je úvodní panel s náhledem nejdůležitějších přehledů:

- návštěvnost
- zdroje provozu
- přehled cílů
- vizualizace návštěv na mapě



Je jen na uživateli jaké přehledy si vybere do úvodního panelu.

Následující přehled je vizualizace návštěv na mapě, v tomto případě zaměřena na Českou republiku, protože pokrývá největší podíl návštěv.



59560města

Stupeň detailu: Město Dimenze: Žádný

Používání webu | Nastavení cíle 1 | Nastavení cíle 2 | Zobrazení:

Návštěvy	Stránky/návštěva	Prům. doba na webu	% Nové návštěvy	Míra opuštění
595 % z celkového počtu stránek: 95,20%	5,06 Průměr stránek 4,94 (2,32%)	00:05:01 Průměr stránek 00:04:51 (3,35%)	36,30% Průměr stránek 37,44% (-3,04%)	35,13% Průměr stránek 36,00% (-2,43%)

	Stupeň detailu: Město	Návštěvy ↓	Stránky/návštěva	Prům. doba na webu	% Nové návštěvy	Míra opuštění
1.	Zlin	254	6,75	00:07:48	8,27%	37,01%
2.	Prague	68	3,46	00:02:16	86,76%	35,29%
3.	Brno	68	4,03	00:05:41	30,88%	32,35%
4.	Uherske Hradiste	67	3,52	00:02:27	4,48%	52,24%
5.	Uhersky Brod	18	3,11	00:03:20	33,33%	27,78%
6.	Ostrava	15	2,60	00:01:42	66,67%	46,67%
7.	Kunovice	9	5,00	00:03:18	55,56%	0,00%
8.	Olomouc	8	4,12	00:03:09	100,00%	0,00%
9.	Kromeriz	7	4,86	00:01:29	85,71%	0,00%
10.	Usti Nad Labem	6	4,33	00:05:57	100,00%	16,67%

Filtr Město: obsahující | Přejít | Pokročilý filtr

Přejít do služby: 1 Zobrazit řádky: 10 110 z 60

Z hlediska přístupnosti webových stránek je užitečný přehled „Prohlížeče a OS“, ze kterého lze vyčíst jaký prohlížeč a operační systém návštěvníci používají a podle toho lze směřovat optimalizaci.

Všechny tyto přehledy lze také exportovat do různých formátů. (CVS pro Excel, PDF...).