Tvorba bezpečných webových prezentací

The Creation of Safe Website Presentations

Bc. Jiří Růčka

Diplomová práce
2012

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Jiří RŮČKA**
Osobní číslo:	**A10461**
Studijní program:	**N 3902 Inženýrská informatika**
Studijní obor:	**Bezpečnostní technologie, systémy a management**
Téma práce:	**Tvorba bezpečných webových prezentací**

Zásady pro vypracování:

1. **Vypracujte literární rešerši technologií pro tvorbu webových stránek. Výběr omezte na Vámi použité nástroje.**
2. **Popište typy útoků na webové stránky a servery, včetně možnosti ochrany proti zmiňovaným útokům.**
3. **Vytvořte webové stránky pro zadavatele s ohledem na bezpečnost a validitu zdrojového kódu.**
4. **Současně se zaměřte na zabezpečení serveru, na kterém stránky poběží.**
5. **Prakticky otestujte dostupnými nástroji zabezpečení stránek i serveru a výsledky vyhodnoťte.**
6. **Zajistěte nalezení stránek ve vyhledavačích a správné zobrazení v nejpoužívanějších prohlížečích.**

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **HOWARD, Michael; LEBLANC, David. Bezpečný kód : Techniky a strategie tvorby bezpečných webových aplikací. Vyd. 1. Brno : Computer Press, 2008. 895 s. ISBN 978-80-251-2050-7.**
2. **ECCHER, Clint. Profesionální webdesign : Techniky a vzorová řešení pro XHTML a CSS. Vyd. 1. Brno : Computer Press, 2010. 672 s. ISBN 978-80-251-2677-6.**
3. **HUSEBY, Sverre H. Zranitelný kód. Vyd. 1. Brno : Computer Press, 2006. 207 s. ISBN 80-251-1180-6.**
4. **LECKY-THOMPSON, Ed; NOWICKI, Steven D. PHP 6 : Programujeme profesionálně. Vyd. 1. Brno : Computer Press, 2010. 718 s. ISBN 978-80-251-3127-5.**
5. **VRÁNA, Jakub. 1001 tipů a triků pro PHP. Vyd. 1. Brno : Computer Press, 2010. 456 s. ISBN 978-80-251-2940-1.**
6. **KUBÍČEK, Michal; LINHART, Jan. 333 tipů a triků pro SEO : Sbírka nejlepších technik optimalizace webů pro vyhledávače. Vyd. 1. Brno : Computer Press, 2010. 262 s. ISBN 978-80-251-2468-0.**
7. **BERTINO, Elisa, et al. Security for web services and service-oriented architectures. Berlin : Springer, 2010. 226 s. ISBN 978-3-540-87742-4.**

Vedoucí diplomové práce: **Ing. Petr Skočík**
Ústav elektroniky a měření

Datum zadání diplomové práce: **24. února 2012**

Termín odevzdání diplomové práce: **15. května 2012**

Ve Zlíně dne 24. února 2012

L.S.

prof. Ing. Vladimír Vašek, CSc.
děkan

doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

ABSTRAKT

Tato práce se zabývá tvorbou webových stránek s ohledem na jejich maximální bezpečnost. Představeny jsou technologie, které byly použity k napsání webových stránek, na ně navazuje část věnující se bezpečnosti. Jsou rozebrány nejčastější útoky na webové stránky a servery a uvedeny možnosti, jak se proti takovým kybernetickým hrozbám chránit. V rámci praktické části je řešena otázka tvorby stránky tak, aby byla dobře přístupná uživatelům z hlediska zobrazení, nalezitelná ve vyhledávačích a především bezpečná. Napsané stránky jsou otestovány na známé hrozby skenery webových zranitelností. Na základě těchto výsledků jsou stránky upraveny tak, aby byla možná nebezpečí minimalizována. Závěrečná část pojednává o zabezpečení webového serveru, na němž stránky běží.

Klíčová slova: bezpečné webové prezentace, zabezpečení webu, zranitelnost webu, webový server, test zranitelnosti, skener webových zranitelností

ABSTRACT

This work focuses on the creation of web pages, taking into account security. Introduced are technologies for the creation of web pages, after which the focus shifts to security. The most common attacks on Web sites and servers are presented, as well as options on how to protect against such cyber threats. The practical part addresses making the page so as to be readily accessible to users in terms of usability, finding the site in search engines and in particular its safety. Written pages are scanned for known vulnerabilities against web scanners. Based on these results, pages are modified so as to minimize potential hazards. The final section deals with web server security, on which the site is running.

Keywords: safe website presentations, web security, web vulnerability, web server, vulnerability test, web vulnerability scanner

Na tomto místě bych rád poděkoval Ing. Petru Skočíkovi za užitečné rady, náměty a celkové vedení diplomové práce. Za poskytnutí přístupu ke školnímu serveru děkuji Ing. Jiřímu Korbelovi, Ph.D. Děkuji taky svým rodičům, kteří mě celé mé vysokoškolské studium po všech stránkách podporovali. V neposlední řadě bych chtěl poděkovat taky Mateřské škole Pramínek ve Valašské Bystřici, za možnost realizace webové prezentace, která je součástí této práce. Všem, na které v tomto krátkém odstavci nevyšlo místo, a podporovali mě v psaní diplomové práce a celkovém studiu, rovněž děkuji.

Motto:

„Neuspějí pouze ti, kteří se o nic nepokusí.“

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

Otázka bezpečnosti je dnes více než aktuální a to ve všech oblastech každodenního života. Výjimkou nejsou ani webové aplikace a servery, na nichž aplikace běží. V této oblasti se jedná o bezpečnostní problém poměrně velkých rozměrů. Když se řekne pojem webová stránka, tak téměř každý civilizovaný člověk ví, o čem je řeč. Drtivá většina osob na síť Internet přichází za účelem vyhledávání textových pramenů, obrázků a dalších souborů. Tato skupina lidí si chce takto obohatit své poznání. Druhá skupina se chce rovněž obohatit, ale naprosto jinak. Do druhé užší a nebezpečnější skupiny můžeme zařadit jak lidi z masa a kostí, tak naprogramované roboty, kteří si kladou za účel poškodit stránky s pomocí využití bezpečnostních chyb ve zdrojovém kódu či nastavení serveru.

Z tohoto hlediska padá do rukou programátorů aplikací běžících na webových doménách právě otázka bezpečnosti webových prezentací, fór, publikačních systémů, internetových obchodů, aukcí, internetových bankovních systémů a dalších webových aplikací. S ohledem na účel použití webové aplikace je třeba definovat zabezpečení, patřičně bezpečnost otestovat a případné bezpečnostní chyby opravit.

Informace zmíněné v této práci se zabývají nejen problémem bezpečnosti webových prezentací, ale i jejich konkurenceschopnosti za pomocí optimalizace stránek pro vyhledávače. Finálním bodem je vytvoření bezpečné, validní, optimalizované a otestované webové prezentace demonstrující teoreticky zmíněné informace a principy.

I. TEORETICKÁ ČÁST

1 TECHNOLOGIE PRO TVORBU WEBOVÝCH STRÁNEK

1.1 Historie

Téma webových stránek nepatří k těm oblastem, u kterých vývoj sahá daleko do historie. Web vznikl pouze jako vedlejší produkt v 80. letech minulého století. Tehdy si chtěli výzkumní pracovníci Evropské organizace pro jaderný výzkum (CERN) ulehčit svou namáhavou práci při získávání podkladů. [10]

Za otce webu (World Wide Web) je považován Tim Berners-Lee, který sepsal v roce 1991 informace o struktuře webu a jeho základním programovacím jazyku. V dnešní době Tim Berners-Lee sleduje směřování vývoje webu a podílí se na udávání standardů v rámci společenství World Wide Web Consortium (W3C). [10]

Za dvě desetiletí existence se Internet stal nepostradatelnou součástí lidského života. Ulehčuje, stejně jako organizaci CERN, i ostatním komunikaci a propagaci, ale na druhou stranu se stává stále více nebezpečnou sítí. Nebezpečí můžeme vidět, kromě virové nákazy počítačů, rovněž v nebezpečném fungování webových stránek, které mohou poškodit jak osobu stránky provozující, tak osoby stránky používající. Dříve, než bude řeč o těchto nebezpečích, je důležité pochopit schopnosti a fungování webových technologií, které budou v rámci této práce využity.

1.2 HTML a XHTML

Jak již bylo řečeno, základy HTML položil britský fyzik Tim Berners-Lee. První verze HTML, kterou Tim Berners-Lee popsal, obsahovala několik logických úrovní textu. Dokument byl lépe čitelný, byly zvýrazněny důležité části dokumentu a bylo možné vložení obrázků a hypertextových odkazů. [9]

Se zvyšujícími se nároky uživatelů vývoj pokračoval, následovala verze HTML 2.0 definující implementaci formuláře. Další změny následovaly ve specifikaci HTML 3.0 definující tabulky a matematické vzorce. Problémem u verze 3.0 bylo, že tehdejší prohlížeče Netscape a Mosaic nepodporovaly všechny elementy napsané v dokumentech standardu. Proto se vytvořila konsorciem W3C verze 3.2, která nabízela pouze to, co prohlížeče umožňovaly. U této verze bylo, kromě vyjmutí některých částí, zdokonaleno formátování písma. Koncem roku 1997 byl přijat standard HTML 4.0, ten umožnil

používání rámů (rozdělení stránky na několik částí) nebo vkládání skriptů. Dva roky na to byla vydána opravená verze HTML 4.01, neobsahovala tedy nic nového, ale pouze opravila chyby ve standardu předchozím. [9]

Současně s HTML 4.0 vznikal i standard XHTML (Extensible HyperText Markup Language), který z HTML vycházel. Standard řešil výměnu dat a jejich ukládání. Časem se zjistilo, že XHTML 1.0 není správné řešení. Standard nenabízel, až na drobné výjimky, ve srovnání s HTML 4.01 žádnou novinku. Problémy zde nastávaly i s podporou prohlížečů, taky proto XHTML 2.0 označilo společenství W3C za dočasně nevyhovující. [9]

V současné době pracuje skupina WHATWG (Web Hypertext Application Technology Working Group) a W3C na HTML5. To počítá s podporou jak HTML, tak XHTML. Jeho nasazení není plánováno dříve než v roce 2020. [9]

Pojmem, který provází celou problematiku HTML, jsou párové či nepárové značky, nazývané někdy tagy. Z nich se skládá celá stránka, párový tag tvoří dvojice značek, mezi nimi je uzavřen obsah, mimo to může být v první z dvojice značek uvedena další specifikace formátování. Celé to může vypadat takto [40]:

```
<p align="justify">
Tohle je odstavec tvořený párovou značkou P. Tag P může obsahovat
informace o formátování, je zde použito zarovnání do bloku zapsáním
align="justify".
</p>
```

Pro správnou formu celé stránky by tohle nestačilo, aby byla stránka validní (odpovídala standardům) je třeba ctít určité zásady. Pro přehlednost je v příloze P I. příklad konkrétní jednoduché stránky. Informace uvedeny mezi znaky <!-- a --> vysvětlují každou část řádku. Tento zvláštní tag s vykřičníkem a pomlčkami říká prohlížeči, že se jedná pouze o komentář. [40]

Dále je třeba se podrobněji vyjádřit k tagu <!doctype>, ten informuje o tom, jaká verze HTML je skutečně použita. V rámci HTML 4.01 je možné využít tři deklarace [11]:

- Strict: striktní deklarace neobsahující jiné než standardní tagy

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- Transitional: deklarace obsahuje i nestandardní tagy

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

- Frameset: deklarace stejná jako Transitional, ale obsahující i rámy (pro rozdělení stránky na více částí)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

V závislosti na použitých deklaracích HTML se mění i zobrazení v nejrůznějších prohlížečích. Pro praktickou část diplomové práce byla použita verze Transitional.

Pokud je stránka uvedená v příloze P I. zobrazena v prohlížeči, vypadá hodně chudě, ale z hlediska validity HTML 4.01 Transitional je správně. Mohli bychom sice text dále zvýraznit pomocí párového tagu nebo obarvit pomocí , ale výhodnější, z hlediska dalších úprav a ladění stránky, je použití kaskádových stylů. [40]

1.3 CSS (Cascading Style Sheets)

V přírodě je pojmem kaskáda označován vodní tok, který překonává určité změny vlivem skokových nerovností povrchu. Příkladem může být pád několika za sebou jdoucích vodopádů. Ačkoliv to nemusí tak vypadat, je vysvětlení kaskádových stylů pomocí vodního toku ideální. Příkladem může být tabulka napsaná v jazyku HTML. První kaskáda může být tabulka jako celek, druhá řádek tabulky, třetí jednotlivé buňky. Každou část je možné si tak naformátovat dle vlastní potřeby, platí vždy nejnižší styl pro element. Docílí se tak nádherných efektů a v případě, že bude takových tabulek deset, postačí pro jejich úpravu, změnit pouze styl a není třeba jednotlivě měnit jednu tabulku po druhé. To platí i o ostatních prvcích HTML dokumentu jako jsou nadpisy, odstavce, seznamy, obrázky nebo odkazy.[40]

Kaskádové styly vznikly v roce 1997, kde byla snaha stimulovat grafický design stránky. Kaskádové styly jsou rovněž ideální, na rozdíl od rámců, k rozdělení stránky. [5] Možnosti implementace kaskádových stylů jsou tři [40]:

- První možností je psaní stylu přímo do příslušných tagů prvků na stránce. Pro zvýraznění nadpisu zelenou barvou stačí napsat:

```
<h1 style="color: green">NADPIS OBRAVENÝ ZELENOU BARVOU</h1>.
```

Tím, že přidáme značce h1 atribut style="color: green" se docílí zelené barvy nadpisu, ale práci to neulehčí. V případě, že je na stránce zvýrazněných nadpisů více, je při změně barvy nutné přepsat všechny. Ideální je změnit vše najednou. To řeší další dvě možnost využití CSS.

- Druhá možnost je zapsání jednoho stylu pro všechny nadpisy úrovně h1. Do hlavičky, mezi značky <head> </head> dokumentu se v tomto případě vloží tento kód:

```
<style>
h1 {color: green; font-size: 8mm}
</style>
```

Nyní by se všechny nadpisy h1 změnily na barvu zelenou. Nadpisům zde byla přiřazena ještě jedna vlastnost – velikost písma. Mezi složené závorky je možno nadefinovat prakticky všechny vlastnosti, které prvek podporuje, ale vždy musí být odděleny středníkem.

- Třetí a zároveň nejpraktičtější možností je používat CSS v samostatném externím souboru *.css. Do hlavičky tak pouze zapisujeme cestu k danému souboru obsahující styl:

```
<style type="text/css" media="all">
@import url("styl.css");
</style>
```

Zde je uvedená relativní cesta k souboru, může být ale uvedena i absolutně. Oddělení stylů středníkem může být importováno do jedné stránky více *.css souborů.

Z celkového pohledu je poslední možnost nejvýhodnější. Nejen, že je možné využít stejný styl pro jakékoliv HTML či PHP dokumenty, ale současně tato možnost snižuje dobu načítání stránky. [4]

Kromě schopnosti ovládání konkrétních tagů CSS obsahuje i třídy a identifikátory, které je možné aplikovat na jakýkoliv tag. Je tak možné obarvit jedním stylem všechny nadpisy, odstavce a kterékoliv další písmo. Rozdíl mezi identifikátorem a třídou je ten, že

identifikátor je jeden unikátní a třída není, může být tedy použita na více prvků. V praxi to vypadá takto [40]:

- Třída:

```
Zápis stylu: .vlevotucne {text-align: left; font-weight: bold;}
Zavolání stylu: <p class=" vlevotucne ">Textová výplň</p>
```

- Identifikátor:

```
Zápis stylu: #vlevotucne {text-align: left; font-weight: bold;}
Zavolání stylu: <p id=" vlevotucne ">Textová výplň</p>
```

1.4 PHP

Vznik PHP je datován do roku 1996, jedná se o programovací jazyk, který nezpracovává prohlížeč, ale webový server. Zkratka v počátcích znamenala Personal Home Page, dnes se jedná o rekurzivní zkratku PHP: Hypertext Preprocessor. PHP má v rámci webových aplikací široké využití. Je možné třeba vytvořit anketu, počítadlo návštěv, graf nebo funkce PHP využít pro složení stránky z více dílčích částí. Právě tohle stejně jako kaskádové styly může ušetřit mnoho času v budoucnu, kdy se bude upravovat třeba jen patička stránky. [12] Právě onu část stánky je možné vložit do celkové stránky buď funkcí include, nebo require. Obě fungují prakticky stejně, malou změnou funkcí je jejich chování v případě, že odkazovaný soubor není nalezen. U include skript doběhne, u require zahlásí fatální chybu [13]. Celé části stránek je možné vkládat takto:

```
<?include "jedna_cast_webove_stranky.php";?>
```

U include a require musí být uvedena pouze relativní cesta, nikoliv absolutní, jinak by k vložení nedošlo. Všechno co patří do skriptu se zapisuje mezi tagy <? ?> nebo <?php ?>. středník v PHP zakončuje každý příkaz. [12]

V době psaní práce je k dispozici verze PHP 5.4.0. To však neznamená, že skutečný webový server, na kterém stránky běží, má verzi stejnou. Pro zjištění aktuální verze PHP nainstalovanou na webovém serveru je možné použít PHP funkci phpversion(). [38]:

```
<?php echo "verze PHP " . phpversion(); ?>
```

Na server dojde dotaz „Napiš verzi používaného PHP.“, server to zjistí, a pomocí funkce echo vypíše do prohlížeče. Na tomto principu pracuje PHP vždy, jednoduše dělá to, co po něm žádáme.

Kódy napsané v PHP můžou mít koncovku podle verze jazyka jako php4 či php5, koncovku podobnou statickým stránkám phtml nebo nejběžnější koncovku php. Napsané dokumenty je možné testovat doma na počítači, ale je třeba nainstalovat webový server a potřebné knihovny. [12] Nejlepší a nejjednodušší varianta, je testovat napsané soubory přímo na skutečném webovém serveru, na kterém poběží stránky.

1.5 JavaScript

JavaScript je oproti PHP zpracováván ve webovém prohlížeči. Uživatel počítače tedy vyšle přes prohlížeč požadavek na přečtení dokumentu HTML obsahující JavaScript, prohlížeč zjistí, že se ve čteném souboru vyskytuje syntaxe JavaScript, zpracuje ji a nám zobrazí výsledek. Je zde určité riziko, že se uživateli skript nespustí. Nezáleží jen na tom, zda uživatel má JavaScript zapnutý, ale taky na tom jaký prohlížeč má zrovna k dispozici. Dnešní verze prohlížečů však se čtením těchto skriptů mají problémy jen velmi zřídka. [12]

Vložení skriptu do stránky je možné dvěma způsoby. Prvním způsobem je umístění syntaxe do HTML dokumentu. Druhou možností je pouze odkázat na soubor, v němž se skript nachází, tyto soubory musí mít příponu *.js nebo *.jse. Ve stránce je odkaz na skript nebo skript samotný vložen mezi tagy <body> </body> nebo <head> </head>, samostatný skript je pak uzavřen mezi značkami <script> </script>. Ve skutečnosti značka <script> obsahuje ještě další atributy [12]:

```
<script language="JavaScript" type="text/JavaScript">
<!-- napsaný kód //-->
</script>
```

Existuje více druhů skriptů, atribut language říká prohlížeči, že je napsaný v jazyce JavaScript. Atribut type informuje o typu uvedeného skriptu. Ačkoliv by takovéto zapsání mohlo stačit, je lepší uvažovat i s možností, že bude podpora JavaScriptu v prohlížeči vypnutá a napsaný kód uzavřít mezi značky `<!--` a //-->. Tím pádem kód mezi značkami <script> </script> přímo nevypíše. Pokud by se jednalo o externí skript, tak stačí přidat atribut src, JavaScript by pak vypadal následovně [12]:

```
<script language="JavaScript" type="text/JavaScript"
src="spousteny_skript.js">
</script>
```

Důležité je také vědět, že nelze zaměňovat malá písmena za velká. Tento požadavek je označován jako Case-Sensitive, tedy citlivost na malá a velká písmena. Proto je třeba dodržovat standardy tohoto skriptovacího jazyka. Použití je velmi široké. Může se jednat o blikající text, otevírání obrázků, tvorbu hodin nebo o způsob zamezení spamování přes webové formuláře. [12]

2 NEJČASTĚJŠÍ ÚTOKY NA WEBOVÉ STRÁNKY

Následující podkapitoly řeší zásadní problémy, s nimiž se potýkají tvůrci webových stránek, kteří využívají technologie zmíněné v první kapitole.

2.1 Vstup od uživatele

Zadávání vstupních dat od uživatele je dnes prakticky nepostradatelnou součástí každé webové stránky. Ať už se jedná o internetový obchod, stránky firmy nebo internetové bankovnictví, téměř vždy, lze takovou část kódu najít. Každý vstup od uživatele na stránkách skýtá potenciální nebezpečí a je třeba se mu pečlivě věnovat. [37]

2.1.1 Definice vstupu

Vstupem od uživatele se rozumí prakticky cokoliv, co zadává uživatel do pole formuláře stránky a odesílá prostřednictvím e-mailu nebo nahrává na stránku. Nejen, že zadaná data můžou být zadaná špatně – nesmyslně, ale mohou určitým způsobem napadnout webové stránky. Z tohoto důvodu je potřeba analýza zadávaných textových informací a souborů od uživatele. [37]

2.1.2 Kontrola vstupu

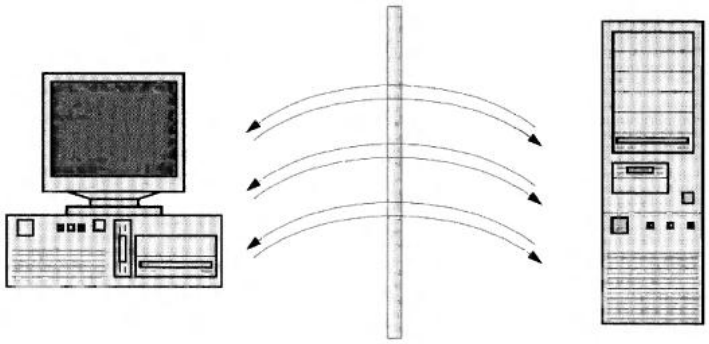
Úkolem kontroly vstupů je definice toho, co nesmí vstupy obsahovat. Jedná se o účinnou cestu, jak zamezit útoku Cross-site Scripting, který je popsán dále. [39] V případě, že je třeba ošetřit emailovou adresu, může napsaný skript vypadat podobně jako v příloze P II.

Pomocí takto napsaného skriptu je možné maximální ošetření vkládané e-mailové adresy a to hned z několika pohledů. Aby uživatel nemohl vkládat do textových polí nesmyslně dlouhé textové řetězce, je délka znaků před i po zavináči nastavena na maximální možnou délku. Pomocí regulárních výrazů PHP je specifikováno, co část za i před znakem zavináč smí obsahovat. Existence domény je testována přes parametr $checkdns v PHP implementovanou funkcí checkdnsrr. [38]

Kontrola emailové adresy by se dala udělat podobně i přes JavaScript, ale bylo by to značně nevhodné. JavaScript je spouštěn na straně uživatele, to znamená, že ani nemusí být spuštěn a do příslušných textových polí uživatel napíše prakticky cokoliv. Z tohoto důvodu

je třeba vždy provádět takovéto kontroly bezpečnosti na straně webového serveru a nikoliv na straně prohlížeče – uživatele. [39]

Jakákoliv data ze serveru, která jsou poskytnuta prohlížeči, je třeba při zpětném odesílání na server znovu testovat, uživatel je totiž schopen je vždy pozměnit. Je třeba si představit počítač oddělený od webového serveru pomyslnou neviditelnou bezpečnostní bariérou, a podle toho testovat vstupy. [39]



Obr. 1. Neviditelná bariéra mezi stránkami a uživatelem [39]

2.1.3 Identifikace správných dat

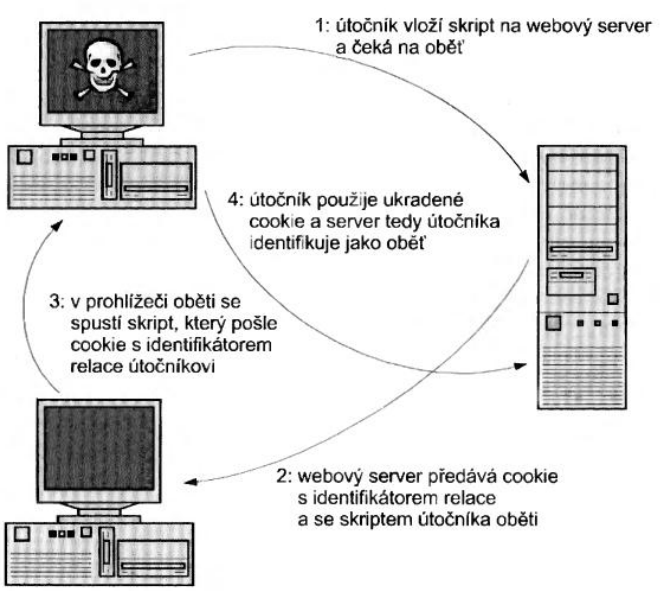
K rozdělení dat na správná a nebezpečná je možné využít dva přístupy. Prvním z nich je vytvoření černé listiny, na kterou se zapíší nebezpečná data. Je zde ale nevýhoda, že se nemusí takto identifikovat všechny nebezpečné vstupy. Lepší přístup je využití bílé listiny, kdy se jednoznačně definují správná data a zbytek se musí odfiltrovat. Při pohledu na HTML jazyk by se za nebezpečná data dala považovat značka <script>, která by na stránce mohla spustit nežádoucí skript. Naopak HTML tag <br>, kromě zalomení řádku, neprovede žádnou katastrofu a mohl by být identifikován jako součást bílé listiny. [39]

2.2 Cross-site Scripting (XSS)

Výraz Cross-site Scripting vznikl po roce 2000, kdy se poprvé objevil tento druh útoku. Pracuje na principu napadení webové stránky, z níž následně odesílá nebezpečný kód uživateli. Z názvu je patrné, že tímto kódem je skript, ten může ukrást relace nebo obsahově změnit webové stránky, velmi populární je taky přesměrování dat z formulářů hackerovi. [39]

2.2.1 Krádež relace

Při krádeži relace hrají hlavní roli soubory Cookies. Pod tímto pojmem je třeba si představit textový soubor, může být dvojího druhu. Jeden typ je uložen v počítači na delší časový úsek a může zaznamenávat různé informace o navštívených stránkách. Druhým typem jsou relační Cookies, ty obsahují informace o relaci při procházení stránek a nejsou dále dlouhodobě ukládány. [39] Postup krádeže relace je zobrazen na obrázku.



Obr. 2. Krádež relace pomocí útoku XSS [39]

Pro krádež relace je třeba nejprve nahrát na stránku JavaScript, který ukrade soubor Cookie. Při spuštění takové stránky uživatelem je pomocí JavaScriptu prohlížeč přesměrován na webový server hackera – ten přijme soubor Cookie. Následně je uživatel přesměrován na původní stránku. [39]

Webový prohlížeč uživatele nahraje webovou stránku ze serveru hackera – tím se zajistí, aby k získané Cookie pasoval i název domény. Zbývá opět pouze přesměrování skriptem na původní stránku. [39]

Nyní už se může hacker připojit ze svého webového prohlížeče jako uživatel na původní webový server, a uživatel (oběť) o tom nemusí mít žádné tušení. Může pouze zpozorovat rychlé probliknutí webového okna. [39]

2.2.2 Obsahová změna stránek

Pod obsahovou změnou stránek je možné si představit cokoliv, co mění původní charakter stránek. Pokud není na stránkách filtrování vkládaného obsahu, je možné značkami jazyka HTML a JavaScriptem značně poškodit stránky. Ukázka nebezpečného kódu může vypadat takto: [39]

```
<script>
      for (q = 0; q<100; q++)
          window.open ("http://www.funnypictures.com/");
</script>
```

Podmínka FOR totiž bude otevírat okna webové stránky www.funnypictures.com tak dlouho, dokud jich nebude přesně sto. Takové praktiky mohou vést ke značnému poškození dobrého jména webu a autor stránek může přijít o mnoho návštěvníků. [39]

2.2.3 Ochrana před XSS

Nejlepší cesta jak odstranit potíže z Cross-site Scripting je zaměřit se na to, aby obsah vkládaný do stránky byl filtrován. V případě, že je uveden znak, který může potenciálně ohrozit stránky, musí být nahrazen tak, aby nebezpečný nebyl. Znak mající funkci formátování či spouštění skriptu na stránce již nebude dostupný. Z hlediska bezpečnosti je třeba kontrolovat výše uvedené znaky a interpretovat je jako čistý text, ne jako metaznačky HTML. Kódy příslušných znaků jsou vidět v tabulce. [39]

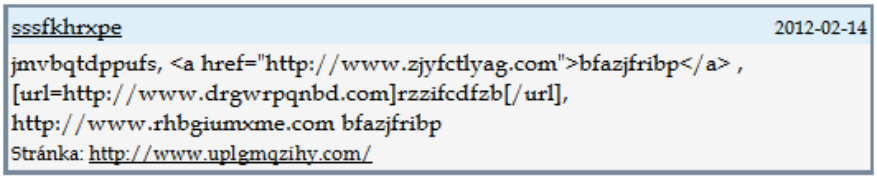
Znak	Znak s formátovací vlastností	Nahrazení kódem
menší než	<	&lt;
větší než	>	&gt;
uvozovky	“	&quot;
apostrof	’	&#39;
ampersand	&	&amp;

Tabulka 1. Kódování znaků HTML

Pro správné pochopení kódování prohlížečem je třeba definovat znakovou sadu v hlavičce dokumentu HTML. V PHP existuje pro nahrazení speciálních HTML znaků přímo funkce htmlspecialchars. [38]

2.3 Komentářový a emailový spam

Pojmem spam byl v počátcích označován pouze obsah šířící se přes elektronickou poštu, za účelem oslovení nejvyššího možného počtu uživatelů e-mailových schránek. Postupem času, kdy se dynamicky tvořené webové stránky rozrůstaly, se zvyšovala i schopnost využití takových stránek k šíření nevyžádaného reklamního sdělení. Diskuse, fóra, a emailové formuláře umístěné na webu se staly rychle kanálem pro tuto nekalou činnost. Problém by se nestal tak masovým, kdyby ony příspěvky na webové stránky vkládali lidé místo spamovacích robotů. Proto bylo potřeba vyvinout takový systém ochrany, který zamezí robotům vydávat se za lidi a odesílat často nesmyslné příspěvky. [14]


sssfkhrxpe 2012-02-14
jmvbqtdppufs, bfazjfribp ,
[url=http://www.drgwrpqnbd.com]rzzifcdfzb[/url],
http://www.rhbgiumxme.com bfazjfribp
Stránka: http://www.uplgmqzihy.com/

Obr. 3. Příklad komentářového spamu [5]

Na obrázku je znázorněn příspěvek od spamovacího robota. Způsobů, jak se vyvarovat takovému napadení stránky je několik, důležité je však myslet jako robot a přemýšlet nad tím, co dokáže a co mu dělá problémy.

2.3.1 CAPTCHA

CAPTCHA je zkratka výrazu Completely Automated Public Turing test to tell Computers and Humans Apart. Jedná se o Turingův test, ten má jasně odlišit, zda za počítačem sedí člověk, nebo spambot (spamovací robot). CAPTCHA ochrana funguje na principu zobrazení generovaného obrázku uživateli, ten na jeho základě musí provést rozhodnutí o tom, co se na obrázku vyskytuje. [14] Nejběžnějším a nejrozšířenějším mechanismem je zobrazení obrázku s čísly či písmeny nebo jejich kombinací. Osoba za počítačem musí rozhodnout, o zobrazených znacích a zapsat do příslušného políčka to, co vidí. Velkým problémem je, že tyto obrázky nemusí být dobře čitelné ani pro lidi. Navíc softwary

založeny na OCR technologii (Optical Character Recognition - optické rozeznávání znaků) rozpoznají prakticky jakýkoliv znak. [15]

Kromě čísel a písmen se historicky později zrodila CAPTCHA založená na poslechu audio souboru, na videosekvenci nebo na otáčení obrázků. Hodně zajímavým projektem je technologie od Google jménem reCAPTCHA. [16]

Obr. 4. reCAPTCHA [16]

Jsou zde vždy dvě slova na opsání, ale jedno je pro tento ochranný prostředek neznámé. Neznámé slovo vzniklo naskenováním knih světové literatury, tak každý kdo opisuje tyto písmena, se stává do jisté míry knihovníkem a přepisuje tištěné knihy do digitální podoby. Systém lze zdarma stáhnout a užívat pro libovolný počet webových domén. Stejně jako ostatní mechanismy i tento jde prolomit. [16]

Nejnovějším konceptem, prozatím ve fázi testování, je 3D CAPTCHA založená na rotaci 3D objektu. [16] Výhodou zde může být nepřeberné množství objektů, nevýhodou naopak nutnost přítomnosti technologie Flash v prohlížeči.

2.3.2 Časový zámek

Za chytré řešení by se dalo označit ověřování identity člověka podle toho, za jaký časový úsek je formulář vyplněn. Roboti vyplňují formuláře v řádu několika vteřin, kdežto lidé i několik minut. Stačí tak sledovat čas od načtení stránky s formulářem a při odeslání porovnat čas s časem minimálním. Pokud bude větší, odešle se, jinak zahlásí chybu. [18]

2.3.3 Textové pole pro robota

Při jakékoliv ochraně formulářů je třeba uvažovat, zda chceme vytvořit něco, co roboti uvidí, nebo co naopak vidět nesmí. V tomto příkladu je tvořeno textové pole jen pro robota. Ten jej vyplní, uživatel pole ani neuvidí, protože bude skryto přes kaskádové styly. PHP kód je zde oproti předchozímu časovému zámku mnohem jednodušší. [18]

```
if($www) {  $status .= " JSI SPAM-BOT!!! BĚŽ PRYČ! <br>";}
```

Tato podmínka kontroluje textové pole formuláře, které se v dokumentu nachází ve tvaru:

```
<input type="text" name="www" id="www" >
```

Skrytí je realizováno jednoduchým kaskádovým stylem, aby nebylo pro uživatele viditelné.

```
#www {display: none;}
```

Jednoduchost, funkčnost a především neobtěžování uživatele, to je to čím je tento druh ochrany proti komentářovému spamu výjimečný.

2.3.4 Logické otázky

Logické otázky jsou založeny na omezené inteligenci robota. Robot neví odpověď, pole vyplní holým nesmyslem, a přesně tady ztroskotá. Je třeba otázky nadefinovat tak, aby uživatel nemusel tápat v encyklopediích a hledat odpověď. [18] Otázka může být formulována třeba takto:

```
Nejčastější barva očí je zelená, hnědá a _____. (piště bez diakritiky)
<input type="text" name="otazka" id="otazka" ><br>
```

Pokud se do tohoto textového pole zapíše „modra“, tak je vše správně. Odpověď kontroluje skript napsaný v PHP.

```
session_start();
      if($_POST["otazka"] == "modra")
            {//echo ("jsi uživatel");}
      else  {//echo ("jsi robot!"); }
```

Pro odeslání formuláře je stejně jako v předchozím příkladu žádoucí splnit podmínku vyžadující správné vyplnění textového pole.

```
if(!$otazka){ $status .=  "JSI ROBOT!!! BĚŽ PRYČ! <br>"; }
```

2.3.5 Kontrola přes JavaScript

Kontrola pomocí JavaScriptu využívá, podobně jako skryté textové pole pro robota to, že robot není dokonalý a neumí všechno. V tomto případě si neporadí s JavaScriptem. [17] Kódy napsané v PHP plní zde stejnou funkci jako v předešlém příkladu logických otázek.

```
session_start();
      if($_POST["kod"] == "ochranapredrobotem")
```

```
            {//echo ("jsi uživatel");}
      else  {//echo ("jsi robot!");}
if(!$kod){ $status .= "JSI ROBOT!!! BĚŽ PRYČ! <br>"; }
```

Výhodou níže uvedeného skriptu je to, že pamatuje i na uživatele, kteří nemají JavaScript na svém prohlížeči zapnutý.

```
<noscript>
Vyplňte "ochranapredrobotem" : <input type="text" name="kod" id="kod" >
</noscript>
<script type="text/JavaScript">
//<![CDATA[document.write('<input type="hidden" name="kod"
value="ochranapredrobotem" />');//]]>
</script>
```

V případě, že JavaScript v prohlížeči funguje, dojde k naplnění skrytého textového pole textem „ochranapredrobotem“. Pokud však JavaScript nebude fungovat, pole se normálně zobrazí s popiskem: Vyplňte "ochranapredrobotem". [17] Párový tag noscript totiž informuje prohlížeč o tom, co má dělat v případě, když JavaScript na stránce nefunguje.

Bylo představeno celkem pět možností jak se co nejlépe chránit před komentářovým spamem. Poslední uvedenou možnost společně se skrytým polem pouze pro robota, považuji za dnes nejlepší způsob ochrany. Nejen že je robotovi maximálně zamezeno, vzhledem k jeho schopnostem, ochranu prolomit, ale je přívětivá i pro uživatele stránky, na niž takový ochranný mechanismus je. Ti prakticky nevědí, že testování robota vůbec existuje.

2.4 Nahrávání souboru na webový server pomocí PHP

Až dosud práce řešila vstup od uživatele v textovém formátu vkládaný přes textové pole. Text byl následně odeslán v textové podobě prostřednictvím elektronické pošty či vložen na stránky. V případě, že bude nutné na stránky nahrávat soubory, tak je třeba se k nim postavit z hlediska bezpečnosti velmi rázně a jednoznačně definovat, co může a nemůže být na server nahráno. Nezřídka vede neopatrnost programátorů těchto aplikací k obrovským nebezpečím a problémům. [19]

Kvůli tomu je třeba pojmout bezpečnost opět jako součást procesu tvorby aplikace a nejen jako dodatečnou bezpečnostní záplatu. Bezpečnostní část se skládá z několika nutných požadavků. Nejdříve je třeba správně nastavit typ formuláře (enctype) tak, aby přes něj šly bez problému posílat data: [19]

```
<form action="upload.php" method="post" enctype="multipart/form-data">
```

V případě uvedeném výše nese soubor PHP název obsahující slovo upload. Teoreticky se soubor PHP může jmenovat jakkoliv, ale názvům jako upload, file, share a podobným informujících o tom, že stránka obsahuje nahrávání, je dobré se vyhnout. Může se stát, že hacker hledá právě takové soubory, přes které může zaútočit. Bezpečnější by mohly být třeba české mutace slov, ale nejlepší je stránku nazvat naprosto odlišně, tak aby neprozrazovala to, co se na ní skutečně nachází. [20]

Po definování typu formuláře a názvu dokumentu je třeba se zaměřit na velikost nahrávaného souboru. Pro manipulaci se soubory se využívá pole $_FILES. Jazyk PHP obsahuje pro definování maximální velikosti souboru direktivu upload_max_filesize. Ta se nastavuje přímo na webovém serveru, bývá zpravidla nastavena na hodnotu 2MB. Její zvýšení či snížení je možné přes soubor .htaccess, umístěný v kořenovém adresáři webu. [20] Vytvořený řádek v souboru .htaccess může vypadat takto:

```
php_value upload_max_filesize 10M
```

V případě využití některých webhostingů je možné nastavení přímo v jejich uživatelských rozhraních.

Dalším nezbytným faktorem je ošetření formuláře dat tak, aby mohly být nahrány pouze povolené soubory. V případě, že by se na webový server měl nahrávat obrázek typu JPEG, tak by kontrola probíhala v této části kódu. [19]

```
$_FILES["soubor"]["type"]) == "image/jpeg"
```

Konstrukce říká, že se vybraný soubor kontroluje podle koncovky, zda je typu JPEG. Je třeba vždy počítat s trochou nejistoty, uživatel může být totiž tak zákeřný, že se pokusí do nastavené složky na serveru nahrát soubor s povolenou koncovkou, ale ve skutečnosti se bude jednat o jiný typ souboru. Nejjednodušším způsobem jak změnit koncovku je přepsat soubor obrazek.js na soubor obrazek.jpg. Problém je ale vyřešen tím, že spuštění JavaScriptu je degradováno právě koncovkou JPG. Aby bylo nebezpečí ještě

minimalizováno, je možné zkontrolovat mimo koncovky taky obsah souboru direktivou mime_content_type. [19]

```
<? php echo mime_content_type('soubor.jpg') ; ?>
```

Výstup z takového kódu by mohl být například text/plain, pokud se bude jednat o textový soubor namísto deklarované fotografie. Kompletní kód nahrávání fotografie může vypadat podobně, jako je uvedeno v příloze P III.

2.5 Skrytí e-mailové adresy před roboty

Roboti mají e-mailové adresy velmi rádi, protože procházením webu jich získají opravdu spoustu, a posílají pak na ně nevyžádanou poštu. Ani roboti ale nejsou dokonalí a při procházení Internetu hledají především řetězce, v nichž se nachází znak @ (zavináč). Roboti hledají znak v takové podobě jak je napsaný zde. Pokud by se pro napsání znaku použil přímo HTML kód znaku, který je zde `&#064;`, tak vyhledávací robot email nenajde. Často využívaný tvar je i jmeno[at]email.cz, ale znak [at] se roboti už naučili hledat a označovat jej jako součást e-mailové adresy, proto by bylo lepší v českých podmínkách využít tvar konstrukce jmeno[zavináč]email.cz. [13]

2.6 Soubor robots.txt

Jeden moc nepraktický způsob jak „ochránit“ část stránek je využití souboru robots.txt umístěný v kořenovém adresáři webu. Tímto textovým souborem lze jednoduše nadefinovat přístupová práva vyhledávacím robotům, tedy kam můžou a kam ne. [5]

```
User-agent: *
Disallow: /administrace/
```

Výše uvedený zápis zakazuje všem robotům (*) přístup do adresáře administrace. Jiné je to s lidmi, těm totiž tento soubor napoví, kde se skrývá něco, k čemu by se neměli dostat. Stačí spustit soubor robots.txt a zjistit co mají roboti zakázáno. Tímto způsobem je možné se dostat i k administrátorskému rozhraní nepříliš opatrného administrátora a to někdy bez znalosti hesla. Vzhledem k tomu, že je taková složka administrace tvořena pro administrátora, tak nemusí počítat s ochranou vstupů a tím pádem se stává velmi atraktivní pro potenciálního hackera. [5]

3 NEJČASTĚJŠÍ ÚTOKY NA WEBOVÉ SERVERY

Kromě útoku na samotný kód webové aplikace je třeba se zaměřit i na nebezpečí kompletního vyřazení webového serveru z činnosti, či omezení jeho funkce. Těmito útoky bývají ohroženy především velké společnosti a organizace. Nebezpečí takového kyberterorismu může mít někdy velmi fatální následky. Proto musí organizace a subjekty co nejlépe chránit své serverové prostory jak fyzicky, tak na úrovni nastavení serveru. Podkapitoly rozebírají ochranu před vyřazení serveru z činnosti, udržení kontroly nad správou serveru a náležitosti z hlediska nastavení, které nelze zanedbat.

3.1 Webový server

Pod pojem webový server je možné si představit počítač zabezpečující provoz webových stránek. Na serveru je nainstalován program, který je schopen plnit HTTP požadavky zadané uživateli (webovými prohlížeči). Nejčastěji je jako serverový program používán Apache, ten využívá přibližně 60 % všech serverů, mezi další patří například ISS od Microsoft nebo iPlanet od Netscape. [9]

Pro nenáročné aplikace je možné využít běžný počítač s připojením do sítě Internet. Pro náročné aplikace se využívají speciální počítače, které obsahují vícejádrové procesory, velkou paměť RAM a prostorné pevné disky. Ty můžou obsahovat i vícenásobné diskové pole nezávislých disků (RAID) pro případ selhání jednoho z pevných disků. Často je využíván například RAID 1, kdy se všechno ukládá na dva pevné disky, při výpadku jednoho je ihned k dispozici druhý. Tyto speciální počítače jsou většinou prodávány v základním osazení, do něhož je možné přidat další komponenty. Na obrázku je vidět serverový počítač od firmy IBM. [9]

Obr. 5. Serverový počítač IBM x3250M3 [9]

3.2 Denial of Service (DoS) / Distributed DoS (DDoS)

Útok Denial of Service (DoS) se dá volně přeložit jako odmítnutí služby. Na server je posílán proud žádostí (paketů) tak dlouho, dokud není vyčerpána paměť serveru a kapacita procesoru. Stránky běžící na serveru tak přestanou naprosto fungovat nebo se maximálně zpomalí. DoS útok je zpravidla prováděn z jednoho počítače. [31]

Distributed DoS (DDoS) je nadstavba DoS útoku, kdy je k útoku využíváno velké množství počítačů. Uživatelé používající tyto počítače většinou ani nevědí, že se stali součástí útoku DDoS. Každá osoba, která nedostatečně chrání svůj počítač, a má v něm infikovaný program Trojan, se může stát tzv. Zombie. Tedy jedním ze strojů, z něhož je prováděn útok. [31]

Existuje obrovské množství známých i neznámých modifikací těchto DDoS útoků a každý den jich několik přibyde. Díky tomu jsou možnosti jak se chránit značně omezeny a je třeba v závislosti na nových hrozbách upravovat i ochranné mechanismy. [31]

3.2.1 Ochrana IPS (Intrusion Prevention System)

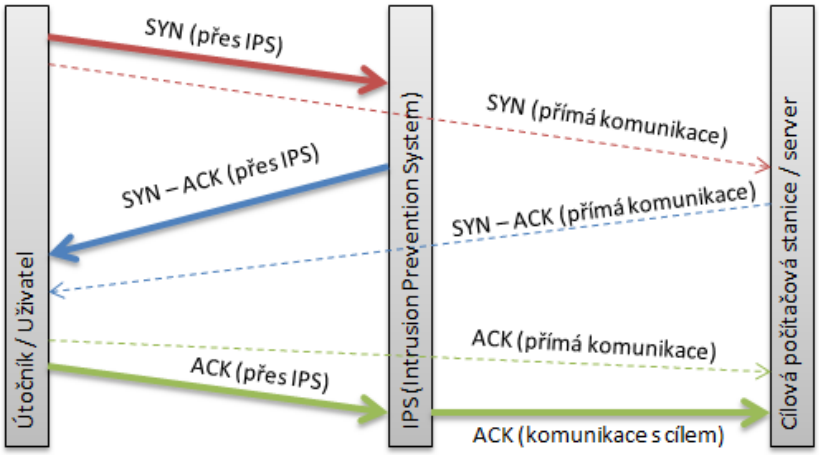
Jednou z možností, jak se chránit před útoky typu DoS a DDoS je využití zařízení mapující průnik z vnějšího světa do vnitřní sítě. Tento systém nefunguje pouze jako detekce nebezpečného provozu, ale je schopen užitečné komunikační kanály ponechat a blokovat ty škodlivé. Zařízení kontroluje pakety až do aplikační vrstvy modelu OSI. Umožňuje tak detekovat nebezpečný provoz ve všech protokolech této vrstvy. [32]

Nejvíce využívaným systémem je HP TippingPoint. Je nejlépe hodnocen díky nízké latenci, která se pohybuje pod hranicí 84 mikrosekund, vysokým výkonem a taky pravidelně aktualizovanými filtry. [32]

Obr. 6. Intrusion Prevention System - HP TippingPoint [32]

Stejně jako u běžné komunikace přes protokol TCP, i zde funguje vše přes třícestný handshake (potřesení rukou). IPS se však stává prostředníkem komunikace a odpovídá místo cíle. Komunikace probíhá v těchto krocích [31]:

1. Uživatel pošle SYN paket cílovému počítači.
2. TippingPoint určí, zda je třeba cíl chránit (v případě že ano, pokračuje se bodem 3.).
3. TippingPoint odpoví pomocí SYN-ACK paketu (normálně by odpovídal cíl)
4. Uživatel odpoví ACK paketem
5. TippingPoint na základě složitých algoritmů rozezná, zda se jedná o odpověď na SYN-ACK a naváže komunikaci s cílem.



Obr. 7. Schéma komunikace bez a s IPS zařízením

Útoky DoS probíhají tak, že je poslán pouze požadavek SYN, ale žádná dopověď na SYN-ACK. V případě, že je takový útok detekován, je spuštěna akce dle nastavení IPS. Jedná se například o blokování IP nebo hlášení útoku příslušným osobám. [31]

3.2.2 Ochrana v rámci HTTP protokolu

Dalším možným způsobem ochrany před DoS útoky je nastavení maximálních možných požadavků z jedné IP za čas nebo útok odfiltrovat podle příznaků, které ho jednoznačně identifikují. Příkladem může být rozšiřující modul serveru Apache známý jako mod_evasive. [33]

Je zde možné nastavit množství dotazů směřující na URL (DOSPageCount) i na server (DOSSiteCount) a časové intervaly v sekundách (DOSPageInterval, DOSSiteInterval).

Pokud jsou tyto intervaly překročeny, je stránka pro dané IP na čas (DOSBlockingPeriod) nedostupná. Součástí modulu je i nastavení emailu, kde mají chodit informace o zaznamenaných útocích (DOSEmailNotify), adresáře logů (DOSLogDir) nebo seznam povolených IP adres (DOSWhitelist). Nastavení modulu může vypadat takto: [33]

```
DOSPageCount                    10
DOSSiteCount                    100
DOSPageInterval                 2
DOSSiteInterval                 2
DOSBlockingPeriod               10
DOSEmailNotify email@domena.cz
DOSLogDir /var/log/apache2/evasive
DOSWhitelist                    10.60.0.7
```

Obr. 8. Ochrana http protokolu - mod_evasive [33]

Ani tato ochrana není stoprocentně účinná, ale výrazně zvyšuje odolnost webového serveru vůči DoS útokům. Z toho důvodu je dobré, pokud to platforma serveru umožňuje, tento modul využívat. [33]

3.3 Přihlašovací údaje

V případě úniku přihlašovacích údajů k administraci webových stránek může být na stránku nahrán nebezpečný kód, stránky pozměněny nebo dokonce kompletně zmazány. Pojem bezpečnosti webové prezentace je třeba chápat i z pohledu ochrany přihlašovacích údajů. Přihlašovací údaj je mechanismus, který jednoznačně identifikuje člověka či skupinu lidí. Ti pak mají na základě znalosti přihlašovacího jména a hesla přístup k dalším, zpravidla citlivým údajům. [34]

Každé odhalení přihlašovacích údajů je nebezpečné, proto je třeba je pečlivě chránit. Nejlépe je třeba tvořit taková hesla, která budou silná a odolná na útok hrubou silou. Je třeba tvořit dlouhá hesla a to jako kombinace čísel, písmen a dalších znaků bez diakritiky. Písmena by neměla tvořit slova. Silná hesla je možné si vygenerovat online generátory, třeba na stránce strongpasswordgenerator.com. [34]

Více informací je na konci praktické části zabývající se zabezpečením serveru.

3.3.1 Odhalení přihlašovacích údajů k administraci webhostingu

V případě, že je heslo už zvoleno, nastává jeho ochrana. Nejdůležitější radou je hesla nikde neukládat, ale pamatovat si je a nesdělovat je nepověřeným osobám. [37]

Obr. 9. Nedůvěryhodný bezpečnostní certifikát

Při zadávání přihlašovacího jména a hesla, k administraci webhostingu, je třeba se ujistit, zda je platný certifikát serveru a komunikace šifrována (adresa obsahuje HTTPS). Informace o nedůvěryhodnosti certifikátu vypisuje prohlížeč, ukázka je viditelná na obrázku. V případě, že server není pronajímán, ale provozován vlastní, je třeba jej udržovat aktualizovaný nejen z hlediska operačního systému, ale i firewallu, antiviru a spywaru. [34]

3.3.2 Odhalení přihlašovacích údajů k FTP

Nejlépe je mít údaje uložené pouze ve své hlavě a nikde jinde, ale z hlediska pohodlí je tento přístup prakticky nepoužíván, přihlašovací hesla a jména jsou kopírována a ukládána. Uložištěm se stali především FTP klienti. Nahrávání souborů přes ně je oproti klasickým administrátorským webovým rozhraním opravdu jednoduché. Hrozba úniku citlivých informací u nich ale stoupá. [21]

```
<iframe src="http://klaomtaF2DDA5" width=1 height=1 style="visibility:hidden;
position:absolute"></iframe>

echo "<iframe src="http://goooogIsene.biz/?click=5E286E" width=1 height=1
style="visibility:hidden;position:absolute"></iframe>"

<script>function c858d4c43w49f79158da084(w49f79158da856){ var
w49f79158db028=16; ..... ;document.write(w49f79158dbfc8(w49f79158df680));
</script>
```

Obr. 10. Nebezpečný kód ve webové stránce [21]

Na obrázku je ukázka kódu vloženého na stránky pomocí odcizení přihlašovacích údajů z FTP. Ten zapříčiní hlášku o infikování webové stránky. Získání přihlašovacích údajů

k FTP je lineárně závislé na míře infikování počítačem viry. Vir se stává nosičem oné cenné informace a někdy dokonce stačí, aby byly údaje pouze zadány, ne uloženy. Nejlepší obranou při takovémto napadení bývá rychlý protiútok. Prvním krokem by mělo být odstranění infikovaného souboru index.php nebo index.html ze stránek a změna přihlašovacích údajů. [21]

Nejlepší je FTP pro aktualizaci webu vůbec nevyužívat, to je ale hodně omezující, naštěstí existují ještě další rady, které je nejlépe dodržovat naprosto všechny bez výjimky. Neukládat hesla k FTP na svém počítači, nepřipojovat se z cizích počítačů, mít aktuální antivirový program, nastavit administraci pouze z konkrétních IP adres nebo využít šifrování SSL nebo TLS. [21]

Obr. 11. Varování před zavirovanou stránkou

Hláška na webové stránce, která je na obrázku nemusí znamenat jen práci navíc, ale rovněž ztrátu návštěvníků, a zařízení na seznam podezřelých webových stránek. Pro zjištění, zda je či není konkrétní stránka podezřelá, je nejlepší variantou použít rozhraní od Google dostupné na webové adrese [21]:

`http://www.google.com/safebrowsing/diagnostic?site=http://www.stranka.cz`

V tomto diagnostickém rozhraní lze nalézt kromě informace škodlivosti stránky, taky to, kdy naposledy byla nakažená, jakými viry nebo z jakých pravděpodobných zdrojů nákazy pocházely.

3.4 Skenování portů

Skenování portu samo o sobě neznamená žádné nebezpečí, to je zde až v případě, kdy útočník využije tyto informace k provedení útoku. Oskenovat se dá prakticky jakákoliv IP adresa, i adresa webového serveru. Nástrojem pro skenování portu může být program

Nmap, nyní pod jménem Zenmap. Skenování portů však stejně jako útočníci využívají přímo správci serverů, aby zmapovali současný stav bezpečnosti sítě.[35]

Ochranou před skenováním portů je zrušení nepotřebných služeb běžících na serveru a ponechání pouze těch, které jsou bezprostředně nutné. Existuje taky program logcheck, který filtruje logy serveru a nachází ty, kde docházelo ke skenování portů. Pomocníkem může být i program portsentry, který dokáže nejen rozpoznat skenování portu, ale rychle na něj reagovat (například zabráněním dalšího připojení útočníka). [35]

3.5 Využití známé chyby

Prakticky žádná aplikace není dokonalá, proto je stále zdokonalována a jsou vydávány nové verze softwaru, které kromě opravených chyb obsahují většinou i nové uživatelské vlastnosti. Při napadení operačního systému, na němž běží server je důležitým faktorem čas. Příkladem může být například Linux, který má volně dostupné zdrojové kódy, tím pádem je možné chybu najít, zveřejnit i zneužít prakticky každým kdo má potřebné znalosti. Program, který využívá zjištěnou chybu, se nazývá expoloit, jeho účelem je ve většině případů ovládnutí napadeného počítače nebo instalace škodlivého softwaru (malware). [35]

Obr. 12. Časový horizont využití exploitu [41]

Jak je vidět na obrázku, doba kterou může útočník využít k útoku je dána časovým horizontem mezi zveřejnění či objevení chyby a mezi instalací aktualizací, která vybranou hrozbu opraví. Je tedy nejlepší mít vždy na serveru aktuální systém. [35]

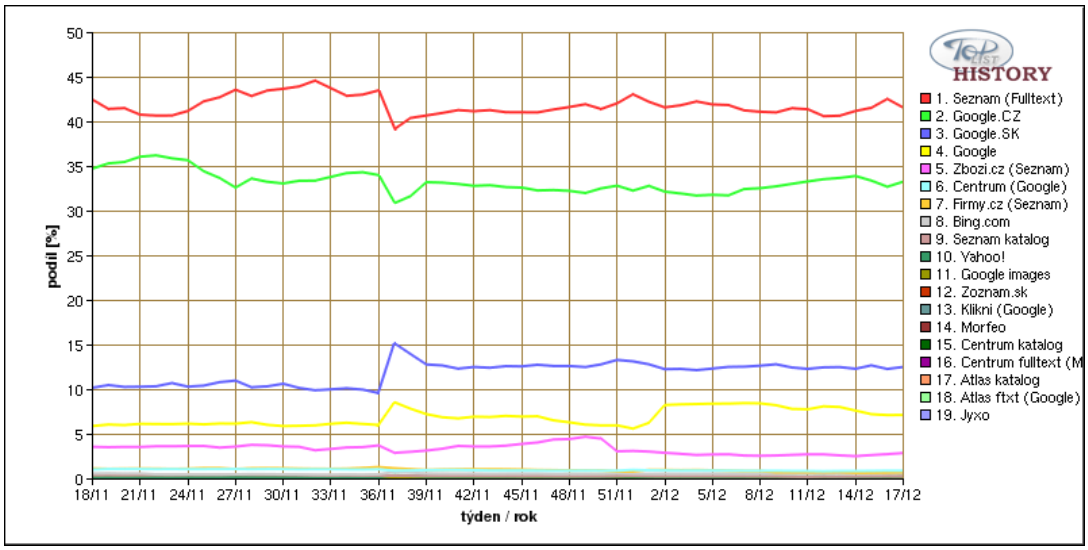
4 OPTIMALIZACE STRÁNEK PRO VYHLEDÁVAČE

Pojem Search Engine Optimization (SEO) je používán v souvislosti s optimalizací webových stránek pro internetové vyhledávače. Jedná se o složitý a časově náročný proces, při němž je třeba respektovat určité zásady, aby byla námi optimalizovaná stránka zobrazována na prvních místech ve výsledcích vyhledávání. [3]

V potaz při optimalizaci je třeba brát velký ohled na konkurenci. Pokud zadáme do vyhledávače Google nebo Seznam slovní spojení „Ubytování Beskydy“, najdeme spoustu relevantních (odpovídajících) výsledků. Máme-li ale opačný případ, že se jedná o slovo nebo slovní spojení, které se na různých stránkách nevyskytuje tak často, tak bývá optimalizace mnohem jednodušší. Příkladem může být vyhledávání různých hudebních interpretů nebo herců. Podobným příkladem mohou být i stránky vysokých škol. [3]

4.1 Vyhledávače v tuzemsku

Dle TOPlist.cz je podíl Seznamu na vyhledávání v České Republice mezi 40 – 45%. Google je na tom o trochu hůře s počtem 30 – 35%. Proto je třeba se zaměřit při optimalizaci domácích webů především na tyto vyhledávače. Na serveru http://toplist.cz/global.html je možné si zobrazit detailní statistiky podle jednotlivých týdnů. Statistika za poslední rok je uvedena na obrázku. [1]



Obr. 13. Statistika nejpoužívanějších vyhledávačů v ČR [1]

4.2 Internetové katalogy stránek

Katalogy stránek někdy bývají mylně označovány za vyhledávače. Rozdíl mezi nimi je jednoduchý, ale na první pohled uživatelem málo znatelný. Vyhledávač zahrne automaticky stránku do výsledků vyhledávání, tedy pro nalezení nových stránek není třeba stránku do vyhledávače nijak registrovat. Přesto je možné, pro rychlejší nalezení stránek, dát vyhledávači o nové stránce vědět. Přidávací formulář stránky do vyhledávače Seznam je na obrázku, stačí vybrat URL adresu stránky a opsat CAPTCHA ochranu proti spamu. [3]

Obr. 14. Vzorově vyplněný přidávací formulář URL na Seznamu [2]

Internetové katalogy stránek jsou tedy databáze stránek, do kterých je třeba stránku ručně přidat. Jen v tuzemsku jich existují stovky. Katalogy můžou být placené nebo neplacené, přidané odkazy můžou být administrátorem schváleny a následně zobrazeny v katalogu za týden, měsíc nebo vůbec. Internetové katalogy slouží k získávání zpětných odkazů. Registrace odkazů do katalogů se někdy nazývá jako indexace. Vyhledávač má svoji databázi neboli index, ve kterém jsou zapsány informace o stránkách. Indexace je prvním krokem k úspěšné optimalizaci stránky, nikoliv však jediným. [3]

Registraci do katalogů je možné si taky objednat nebo zakoupit licence na poloautomatickou registraci softwarovými nástroji. Existuje spousta seznamů katalogů jako například seo-starter.cz nebo www.seznamkatalogu.cz. Při registraci je nutné pečlivě zvolit titulek a popis stránky s využitím klíčových slov. Nejlepší je mít několik popisů a titulků, aby byla stránka vyhledatelná na různá slova a slovní spojení. [3]

4.3 Klíčová slova

Slova, která jsou důležitá k nalezení stránky, jsou nazývána klíčovými slovy. Klíčová slova je třeba na stránkách volit tak, abychom zasáhli co největší spektrum uživatelů internetu.

Jak moc je slovo či slovní spojení hledáno je možno u Seznamu ověřit běžným zadáním hledaného slova a po nalezení výsledků vybrat v patičce stránky „statistika dotazu“. Zadání dalšího klíčového slova, pro porovnání s předchozím, je možno zadat až na stránce statistik. [3]

Konkurenční Google Insights for Search umožňuje sledovat četnost hledaného výrazu ve vybraném časovém horizontu nebo lokality, ze které byl požadavek na vyhledávání zadán. Výhodná může být také inspirace od úspěšné konkurence. [3]

4.4 Optimalizace zdrojového kódu

Prakticky můžeme optimalizaci rozdělit na dva druhy na „on-page“ a „off-page“. Co se týká „off-page“ optimalizace, tak se jedná především o budování zpětných odkazů. Optimalizace „on-page“ probíhá na stránce, optimalizuje se zdrojový kód stránky. [3]

4.4.1 Text hlavní stránky

Mít pečlivě vymyšlený text na stránce obsahující klíčová slova je zásadní. Text nesmí být duplicitní, to znamená, že je třeba originality, ne kopírování z jiných stránek. Důležitým faktorem je i četnost a pozice klíčového slova v textu. Důležitých je zejména prvních 500 znaků textu. [6]

4.4.2 Název stránky

Název stránky je třeba mít uvedený jasně a stručně, každá stránka by měla mít jiný unikátní název. Někdy je součástí názvu stránky i webová adresa, ideální je ji umístit nakonec. Není vhodné dávat titulek delší než 60 znaků, protože pak se stejně celý ve výsledcích vyhledávání nezobrazí. [3]

4.4.3 Nadpisy

Nadpis úrovně h1 má nejvyšší prioritu a bývá využíván v co nejvyšší míře k názvu webu. Z hlediska SEO je ideální používat pouze jeden nadpis h1. Ačkoliv existují úrovně nadpisů h1 – h6, tak roboti pracují s nadpisy nejvíce do úrovně h3. Platí, že s vyšším číslem priorita důležitosti klesá. [6]

4.4.4 Zvýraznění textu

Zvýrazňování důležitých klíčových slov pomocí párových značek je výrazně respektováno vyhledávacími roboty. Značky zdánlivě plní stejnou funkci, ale text jen zvýrazňují, nepřikládají mu žádnou váhu, proto je tato varianta pro klíčová slova nevhodná. [6]

4.4.5 Název domény

Doménou je nazývána adresa webu. Domény s pomlčkami jsou pro SEO výhodnější. Nejideálnější je, když jsou pomlčkou odděleny jednotlivé slova domény. Dříve platilo, že kdo si koupil doménové jméno obsahující klíčové slovo, tak získal ve výsledcích vyhledávání vyšší pozici. Dnes záleží už na mnoha faktorech. [3]

4.4.6 Obrázky

Obrázky na stránkách netvoří jen design, ale slouží i jako zdroje návštěvnosti. Při vkládání obrázků do stránek je třeba dbát na jméno obrázku, alternativní popis, titulek obrázku, ale i na text vyskytující se v jeho okolí, tak aby s obrázkem souvisel. Stejně tak by se nemělo zapomínat ani na rozlišení obrázků. Vzorový příklad by mohl vypadat takto [3]:

```
<img src="levne-ubytovani-chalupa-beskydy.jpg" title="Levné ubytování -
Chalupa v Beskydech" alt="Levné ubytování - Chalupa v Beskydech"
width="190" height="142">
```

4.4.7 Textové odkazy na stránkách

Každý odkaz by měl mít cílený anchor text, tedy text, který uživatel vidí jako odkaz. Častou chybou bývá, že odkaz je třeba slovo „zde“ nebo „stahuj“ místo klíčového slova stránky, na niž odkaz odkazuje. Důležité je také pojmenování odkazu. [3]

```
Zde najdete zmiňovaná <a href="http://www.stranka.cz" title="Horská
kola">horská kola</a> a další cyklistické potřeby.
```

4.4.8 Velikost stránky

Čím je velikostně stránka menší, tím se s ní robotům lépe pracuje a rychleji se načítá. Stránka samotná, bez obrázků a připojených kaskádových stylů, by neměla přesahovat 100kB zdrojového kódu. [3]

4.4.9 Použité kódování na stránce

V českém prostředí, u Seznamu a podobných konkurenčních vyhledávačů je nutno splňovat jedno ze tří kódování. Jedná se o sadu UTF-8, WINDOWS-1250 a nebo ISO-8859-2. V praxi je v hlavičce HTML dokumentu jeden z následujících řádků:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<meta http-equiv="content-type"content="text/html; charset=windows-1250">
<meta http-equiv="content-type" content="text/html; charset= iso-8859-2">
```

Pokud použijeme v hlavičce dokumentu jiné kódování, tak nám ho Seznam nedokáže indexovat. Ideální je alternativa UTF-8, tedy kódování, které lze používat kdekoliv na světě.

4.4.10 Metaznačky v hlavičce

Metaznačkami jsou nazývány informace uváděny v hlavičce stránky, ihned za názvem stránky TITLE. Atributy metaznaček jsou vždy dva. Druhým je atribut „content“, který obsahuje informace vyplývající z atributu prvního označeného jako „name“. Předešlá definice kódování je taky metaznačka, ale definující systémový atribut pomocí „http-equiv“. [3]

Mezi nejdůležitější metaznačku s atributem „name“ v hlavičce stránky patří popis stránky v následujícím formátu:

```
<meta name="description" content="Popis stránky s klíčovými slovy,
přibližně 250 znaků">
```

Ten obsahuje popis stránky (čím se stránka zabývá, co na stránce lze najít). Popis je využíván především jako alternativa v případě, že na stránce není nalezen žádný relevantní text.

Mimo zmíněné lze zadávat do hlavičky spoustu dalších metatagů, např. týkající se zakázání indexace roboty, autorství nebo jako systém ověřující identitu majitele stránek.

4.4.11 Mapa webu

Vyhledávače pracují cyklicky na principu tří posloupností: sběru dat, uložení do databáze a umožnění vyhledávání. Data sbírají roboti, aby sběr dat byl rychlejší a návštěvnost stránek vyšší, tak je nezbytné vytvořit mapu webu (sitemap). Mapu je možné vygenerovat

například pomocí XML-Sitemaps.com. Výsledný soubor sitemap.xml, sitemap.html, nebo sitemap.php pak stačí jen zkopírovat do kořenového adresáře webu. [3]

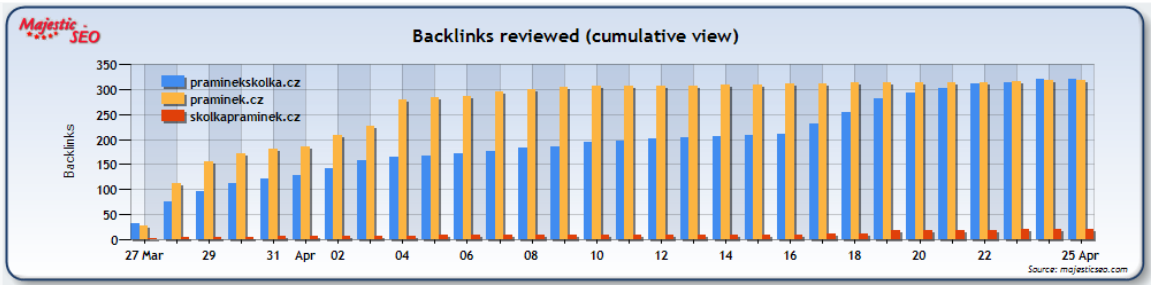
4.5 Optimalizace probíhající mimo stránku

Po napsání kvalitního kódu (on-page optimalizaci) je třeba se zaměřit na optimalizaci probíhající mimo stránku (off-page optimalizaci). Ať už se jedná o jakékoliv níže zmíněné dílčí části, tak se vždy bude jednat a získání co nejvíce zpětných odkazů na stránky. [6]

4.5.1 Zpětný odkaz

Zpětný odkaz je textový odkaz směřující z jedné stránky na druhou. Zpětné odkazy můžeme rozdělit na interní a externí. Interní odkazují na stránku na stejné doméně, jsou podstatné především z hlediska anchor textu, který posílí klíčová slova. Externí odkazy odkazují na stránku mimo doménu. Spoustu zpětných odkazů je možné si zajistit registrací do internetových katalogů nebo výměnou. [3]

Pro mapování nárustu zpětných odkazů je výborné využít analýzy stránek www.majesticseo.com. Nástroj je zdarma, ale prozatím neumí počítat odkazy na subdoménách, takže reálný počet může být i vyšší. [8]



Obr. 15. Zvyšování zpětných odkazů u domény praminekskolka.cz a konkurence [8]

4.5.2 PageRank (PR) / S-rank

Podle počtu zpětných odkazů, jejich umístění na stránce a obsahové podobnosti stránky je počítán rank stránky – její ohodnocení. Neplatí, že čím vyšší rank, tak tím lepší, ale je třeba vycházet z uskutečněných kontraktů. Google používá PageRank, Seznam S-rank. Oba dva hodnotí stránku na stupnici 0 až 10. Nejvyšší hodnocení 10 má jen několik stránek. Algoritmy výpočtu nejsou známé, takže hodnoty nelze brát jako stoprocentní. Stejně tak pozice ve vyhledávačích s hodnocením stránky nejsou pevně spjaty. Je to však jedno

z hodnotících hledisek vyhledávače. Ranky od Google i Seznam je možno si zkontrolovat na mnoha stránkách, například na http://pagerank.jklir.net/. Na zmíněných stránkách je možno změřit si i další ranky jako je například Alexa rank nebo Jyxo rank. [3]

4.5.3 Sociální sítě

Z hlediska SEO je výhodné aktivovat si stránku na sociální síti Facebook. Stránka je na rozdíl od běžných profilů uživatelů, zobrazitelná vyhledávači a získává hodnocení jako každá jiná stránka. Užitečné může být i prezentování fotografií na sociální síti sdílející fotografie Flickr. Výhodou zde je viditelný odkaz na stránku, z níž byla fotografie vložena. Další místo, kde je možné se pomocí krátkých příspěvků prezentovat je Twitter. Twitter a Flick je možné propojit s Facebookem, tak to co se přidá na tyto dvě sociální sítě, se zobrazí i na Facebooku. [3]

4.5.4 Linkovací služby

Dalším způsobem jak propagovat stránky je použití linkovacích služeb (stránek). Linkovací služby fungují na principu vkládání krátkých a výstižných úryvků. U vloženého úryvku je nadpis, který zpravidla obsahuje odkaz vedoucí na propagovanou stránku. Jednotlivé linky (odkazy) obsahují i hodnocení od ostatních uživatelů služeb a jejich komentáře. Mezi linkovací služby bychom mohli zařadit www.linkuj.cz nebo www.pridej.cz. [3]

4.5.5 PR (Public Relations) články

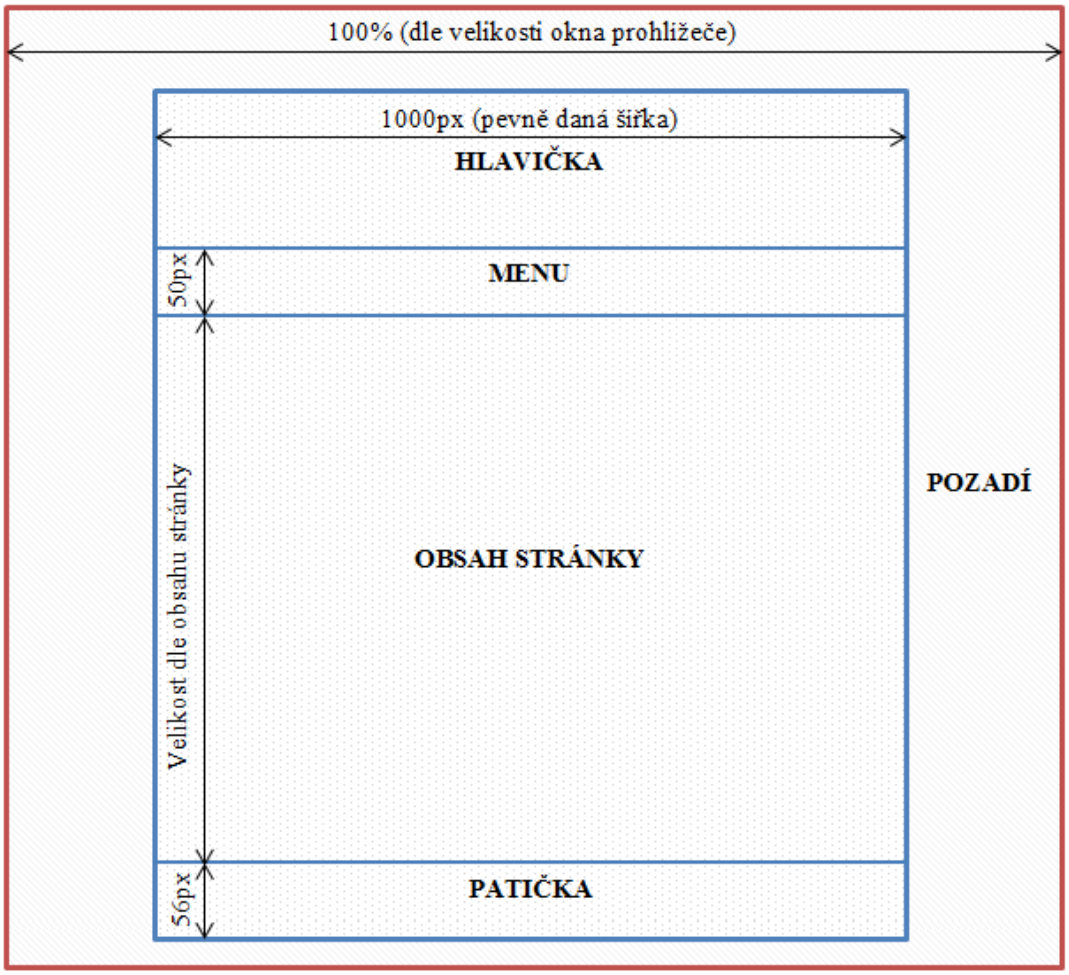
Jedná se o širší kontakt se společností. Články mají za úkol umět prodat produkt nebo službu. PR článek je třeba psát tak, aby neobsahoval zbytečné informace, byl smysluplný a nepříliš dlouhý (30 řádků). Existují placené i neplacené systémy, kde je možné nahrávat tyto články a pomocí klíčových slov a cíleného anchor textu posilovat stránky. Pozor je třeba dávat na duplicitu, ideálně nekopírovat na všechny PR systémy stejné texty. [3]

Uvedené techniky SEO jsou nejpoužívanější. Mezi další můžeme zařadit soutěže, nebo zaregistrování do vyhledávačů zboží. [3]

II. PRAKTICKÁ ČÁST

5 KONCEPCE WEBOVÝCH STRÁNEK

Stránka je rozdělena pomocí CSS na několik bloků tak, aby byla co nejjednodušší její správa a rovněž aby bylo zobrazení ve všech prohlížečích totožné. K vkládání jednotlivých bloků do stránky je využita PHP funkce include. Celá stránka se skládá celkem ze 4 bloků (hlavička, menu, obsah stránky a patička). Jako pozadí stránky je nastaven statický, cyklicky se opakující, obrázek. Šířka stránky je určena pevně na 1000px, což zcela odpovídá rozlišení dnešních monitorů a notebooků. Velikost obsahové části stránky není dána pevně, ale mění se v závislosti na velikosti vkládaného obsahu.



Obr. 16. Rozdělení webové stránky na bloky

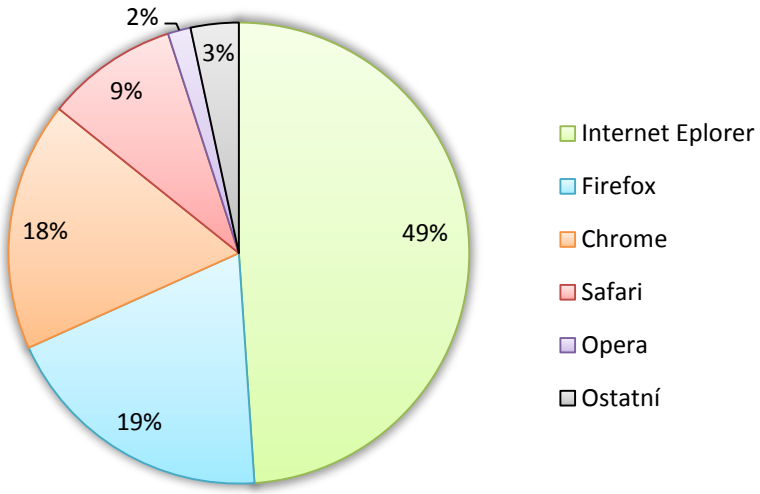
Veškeré informace a fotografie uváděné na adrese www.praminekskolka.cz jsou poskytnuty přímo od vedení MŠ Pramínek, Valašská Bystřice. Po dohodě s touto organizací jsou zdrojové kódy stránek uvedeny na přiloženém CD v adresáři s názvem webove_stranky.

6 TESTOVÁNÍ NAPSANÝCH WEBOVÝCH STRÁNEK A SERVERU

Důležitým bodem napsaných stránek je jejich testování a následná úprava z hlediska zranitelnosti, zobrazení v prohlížečích a optimalizace pro vyhledávače.

6.1 Test zobrazení v prohlížečích

Správné zobrazení v prohlížečích je důležitým faktorem každé úspěšné stránky. Proto je třeba při optimalizaci brát v potaz všechny nejpoužívanější prohlížeče webových stránek. Častou chybou bývá, že stránky jsou laděny pouze na jedné platformě prohlížeče a chyba bývá o to větší, pokud programátor píšící web vůbec nepoužívá Internet Explorer. I přesto, že je tento prohlížeč považován, z hlediska bezpečnosti a správného zobrazení obsahu, za nejnebezpečnější, jej stále používá celosvětově přibližně polovina osob procházejících Internet.



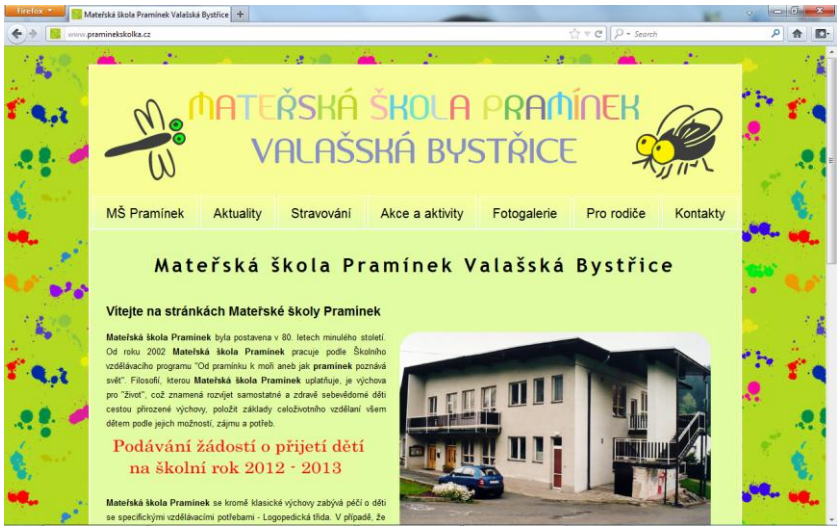
Obr. 17. Podíl používaných prohlížečů celosvětově (únor 2012) [22]

Prohlížeče Firefox a Chrome si dlouhodobě drží druhé pozice, na třetí příčce je prohlížeč Safari, norská Opera tvoří pouhé dvě procenta, zbylé 3 procenta tvoří ostatní webové prohlížeče. Grafické znázornění je zobrazeno na obrázku výše. [22] V České Republice je situace podobná, stejně jako ve světě, popularita Internet Exploreru klesá a zvyšuje se podíl Firefox a Chrome. V roce 2008 byl Internet Explorer využíván 63% českých uživatelů, v roce 2011 to bylo už necelých 45%. [23]

Při psaní webových stránek nastal největší problém u majoritního Internet Exploreru verze 9. Proto, aby bylo možné zobrazit stránku stejně jako u zbývajících čtyř prohlížečů, bylo

nutné porušit určité standardy. Za zmínku stojí například vepsání stylu přímo do tagu způsobem <img style="border: none;">. Ideální řešení je umístit pouze odkaz na název stylu ve tvaru <img style="nemamradexplorer">, ale zde to nefungovalo.

Obr. 18. Zobrazení v prohlížeči Internet Explorer (verze 9.0)



Obr. 19. Zobrazení v prohlížeči Firefox (verze 10.0.2)

U všech testovaných prohlížečů se podařilo dosáhnout stejného zobrazení stránky. [24] Na obrázcích je zobrazení stránek v prohlížečích Internet Exploreru a Firefox. Zobrazení v prohlížečích Google Chrome, Opera a Safari je doloženo v příloze P IV.

6.2 Test nalezení stránek ve vyhledávači

Pro zjištění pozice výsledků vyhledávání byl použit nástroj Collabim dostupný online na adrese www.collabim.cz. Jedná se o jednoduché rozhraní, do něhož se zadají klíčová slova,

na něž je stránka optimalizována. Pro pravidelné kontroly pozic je třeba si nainstalovat modul do prohlížeče. Díky němu je ve verzi free schopen Collabim denně kontrolovat pozice až pěti klíčových slov.

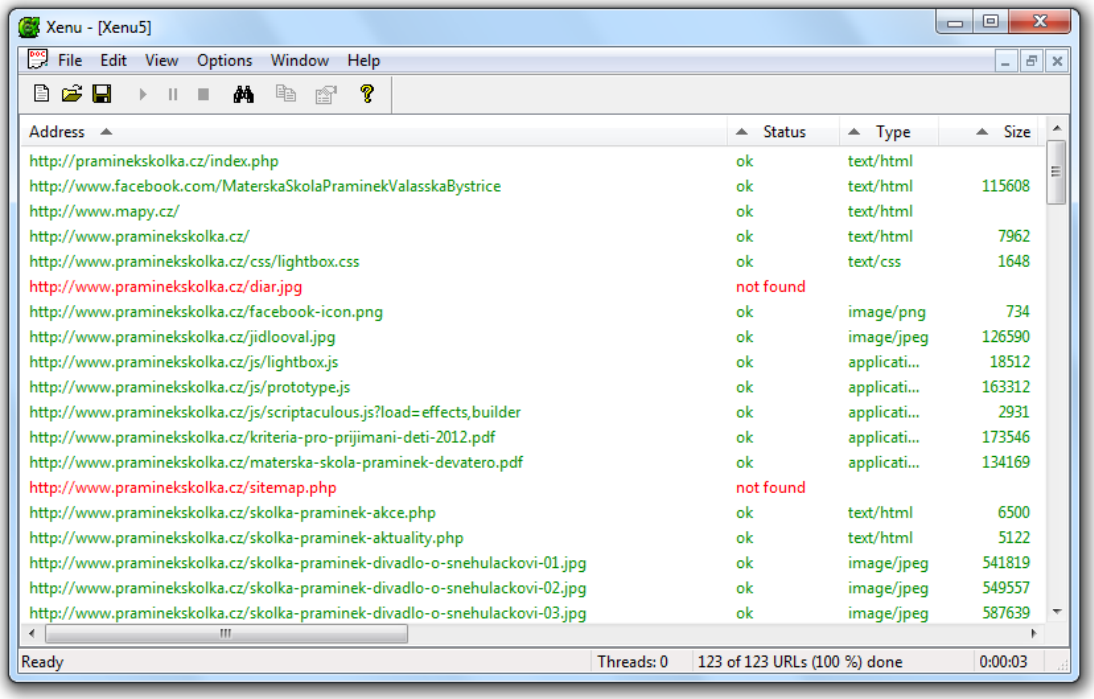
Tabulka na obrázku znázorňuje pozice klíčových slov domény www.praminekskolka.cz. V tabulce pozic je dosaženo výsledků nižších než číslo 10, to znamená, že na všechny tyto vyhledávaná slova je stránka dostupná na první stránce v Seznamu i Google. Pro potenciální návštěvníky tak nebude problém stránku nalézt.

Klíčové slovo	Hledanost Google (svět)	Pozice Google CZ	Pozice Seznam.cz
mateřská škola valašská bystřice	0	2 (+0)	1 (+0)
pramínek valašská bystřice	0	2 (+0)	1 (+0)
mš valašská bystřice	0	2 (+0)	4 (-1)
školka valašská bystřice	0	3 (+0)	1 (+0)
škola valašská bystřice	0	4 (+0)	1 (+0)
mateřská škola pramínek	0	5 (+0)	2 (+0)
školka pramínek	0	6 (+0)	2 (+0)
mš pramínek	22	7 (+0)	2 (+1)
pramínek	58	7 (+0)	7 (+0)

Obr. 20. Zjištění pozic ve vyhledávání pomocí nástroje Collabim

6.3 Test funkčnosti odkazů na webové stránce

Procházení jedné stránky za druhou a zkoušení, zda v ní napsaný hypertextový odkaz funguje, může být i u menších webů zdlouhavé. Rovněž se ručně nemusí otestovat ani všechny zapsané odkazy. Pro funkci oskenování veškerých odkazů na stránce je perfektní volně šiřitelný program jménem Xenu. Program nekontroluje jen odkazy na html či php dokumenty stránky, ale umožňuje taky kontrolu, obrázků, kaskádových stylů, souborů JavaScriptu, PDF a jakýchkoliv dalších souborů, na něž vede hypertextový odkaz.

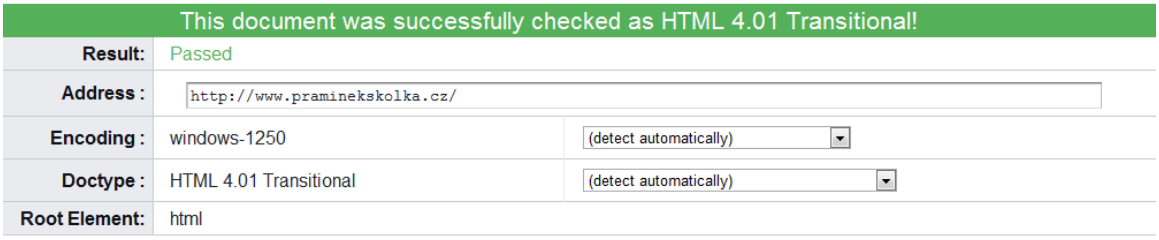


Obr. 21. Kontrola funkčnosti všech odkazů na stránce

V rámci testování tvořené prezentace a funkčnosti programu Xenu Link Sleuth byly ze stránek www.praminekskolka.cz vymazány dva soubory, na něž vedl odkaz. Jednalo se o obrázek „diar.jpg“ a mapu stránek „sitemap.php“. Oba nenalezené soubory program zvýraznil červenou barvou (výsledek je viditelný na obrázku).

6.4 Kontrola validity webové stránky

Pod pojmem validita je třeba si představit webové stránky napsané v souladu s technickými pravidly HTML jazyka. Validita je na webu velice důležitý faktor. Kontrolu takovýchto technických předpokladů je možno kontrolovat online na serveru validator.w3.org. Na obrázku je zobrazen test napsané stránky, pro kontrolu stačí zadat URL adresu stránky a kontrolu potvrdit.



Obr. 22. Kontrola validity stránek na validator.w3.org

Mít validní stránky neznamená mít pouze pocit z dobře odvedené práce. Jedná se o vylepšení klíčových faktorů jako je správné a rychlejší zobrazení stránky, snadná přístupnost pro roboty, zaručená kompatibilita s budoucími verzemi prohlížečů, správná funkčnost i po vypnutí kaskádových stylů nebo přehlednost zdrojového kódu pro budoucí úpravy. [17]

Všechny podstránky webu www.praminekskolka.cz byly otestovány z hlediska validity. Validátorem nahlášeny chyby byly opraveny a nyní jsou stránky naprosto validní podle HTML verze 4.01 Transitional.

6.5 Test zranitelnosti webových stránek a serveru – komerční scannery

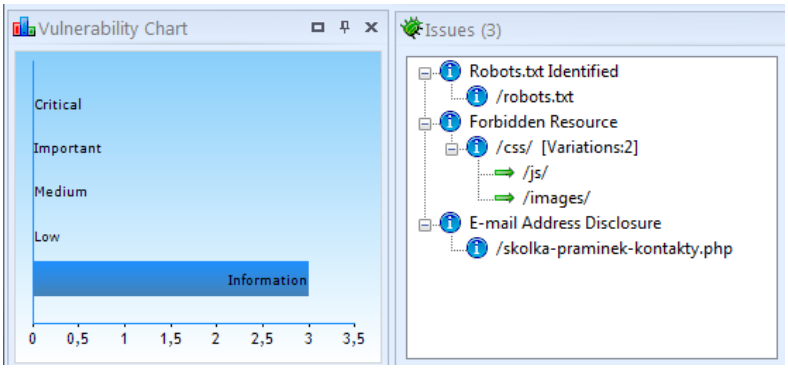
Existuje velké množství aplikací, jimiž lze otestovat webové stránky. Jedná se jak o nástroje komerční, tak o nástroje volně šiřitelné. V komerční i nekomerční oblasti byly pro penetrační testování zvoleny celkem tři web scannery. Dohromady tedy bylo provedeno šest kontrol stránek. Jejich výběr byl realizován na základě hodnocení jednotlivých web scannerů na stránce http://sectooladdict.blogspot.com [29]. Byl kladen důraz na to, aby byl otestován jak scanner s nízkým hodnocením, tak s vysokým.

Pro otestování stránek byl vybrán z velkého množství komerčních produktů Netsparker, Acunetix Web Vulnerability Scanner a WebInspect. Netsparker obsahuje, ze zmíněných scannerů, nejméně funkcí, Nessus nejvíce. Všechny tři testované programy je možné vyzkoušet bezplatně buď v licenci Demo, nebo Trial. Tyto verze jsou, vzhledem k použitým technologiím na stránce, dostačující.

6.5.1 Netsparker

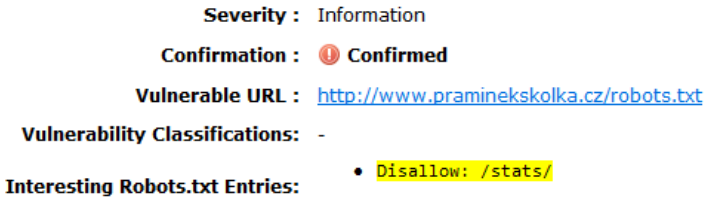
Netsparker od společnosti Mavituna Security je užitečný nástroj pro testování zabezpečení webu. Jeho podpora je pouze pod operačním systémem Windows. Dle specifikace je nastaven tak, aby nalezl minimální počet falešných zranitelností a zaměřil se pouze na to, co je opravdu nebezpečné. [26] Netsparker, podobně jako téměř všechny webové scannery, rozlišuje různé druhy zranitelností podle jejich nebezpečí. U tohoto scanneru je celkem pět úrovní. Nejnižší hrozba je vždy informativního charakteru, je třeba ji vzít na vědomí a uvážit jak moc je nebezpečná a jestli je třeba ji vůbec řešit. Výsledky testu stránek jsou zobrazeny na obrázku.

Netsparker označil informativně, kromě emailové adresy, taky soubor robots.txt a adresáře na webu, k nimž nemá uživatel přístup.



Obr. 23. Nalezené zranitelnosti v programu Netsparker

V soboru robots.txt se tomuto webovému scanneru nezdálo zakázání přístupu do adresáře statistik webové stránky. Scanner tak učinil, protože adresáře uložené v souboru robots.txt by mohl označit hacker za svůj cíl.



Obr. 24. Nalezená zranitelnost „Robots.txt Identified"

Pro opravu stačí umazat v souboru robots.txt řádek, který je na obrázku výše žlutě. Přístup do adresáře /stats/ tím nebude nijak narušen, je totiž chráněn už defaultně nastavením webového serveru.

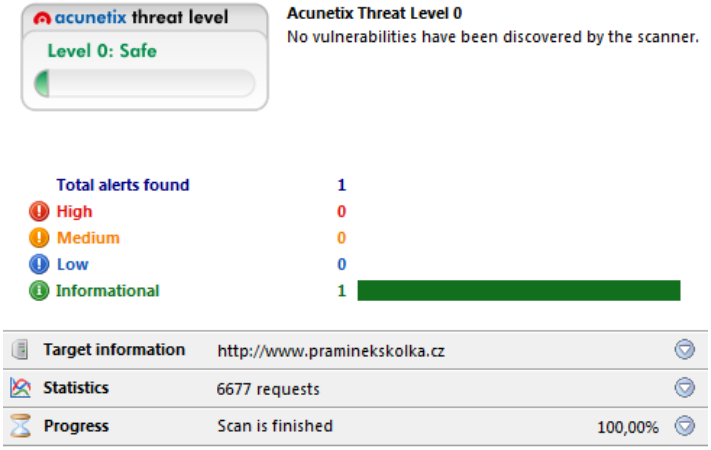
Označení nepřístupných adresářů nepovažuje, dle uvedených informací o nebezpečí, ani samotný scanner za závažný problém, jedná se pouze o dodatečnou informaci. V adresářích se nenachází nic, co by mělo být uživateli přístupné, z tohoto důvodu není řešení tohoto problému nutné. Pro zobrazení kompletních informací o každé hrozbě, ji stačí vybrat a tak o ní zobrazit kompletní informace v programu obsaženy.

Hrozba „E-mail Address Disclosure" informuje o napsání emailové adresy včetně zavináče na webové stránce. To může být zdroj nebezpečí v případě, že roboti procházejí stránky a shromažďují informace o řetězcích obsahující znak @. Program obsahuje přímo popis nebezpečí, zranitelnou stránku, dopad nebezpečí a konkrétní vzorek nalezený na stránce.

Problém emailové adresy byl vyřešen zapsáním znaku zavináč jako HTML kódu ve formě `&#064;` je možné využít i další techniky zmíněné v teoretické části v podkapitole Skrytí emailové adresy před roboty.

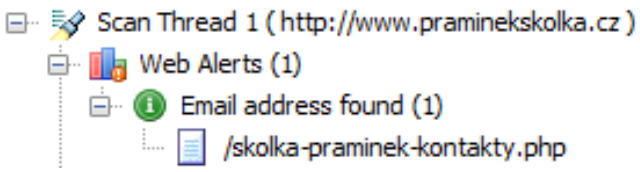
6.5.2 Acunetix Web Vulnerability Scanner

Tento scanner fungující na Windows má, stejně jako Netsparker, přívětivé uživatelské rozhraní a dokáže oskenovat web na velké množství zranitelností. Informuje o nalezených zranitelnostech a obsahuje návod jak je odstranit. Software nevyužívá jen testování na bázi černé skřínky (black box), ale skenuje i zdrojový kód aplikace. Díky této kombinaci Acunetix Web Vulnerability Scanner nahlásí minimum falešných potenciálních chyb. [25]



Obr. 25. Výsledky testu v programu Acunetix

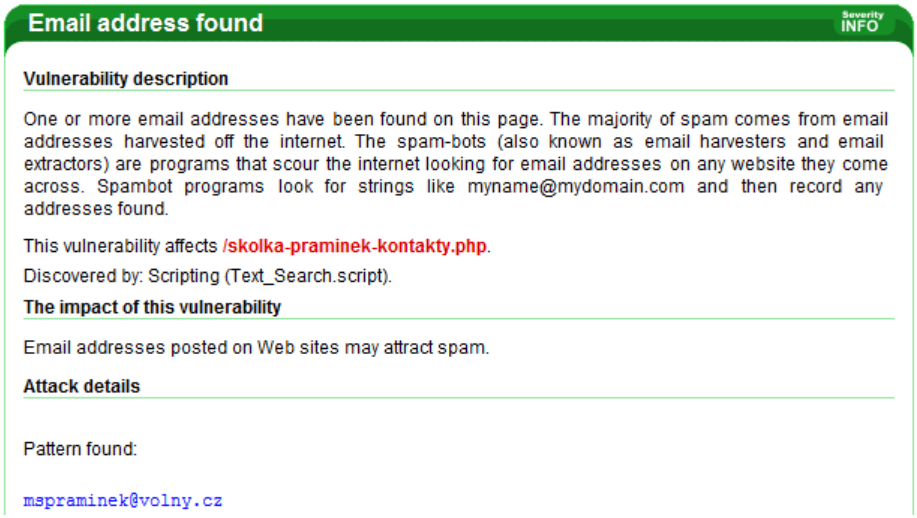
Napsaná stránka byla taky otestována pomocí programu Acunetix Web Vulnerability Scanner. Jak je vidět na obrázku, byl nalezen pouze jeden informativní bezpečnostní problém a web byl označen za bezpečný (Level 0: Safe).



Obr. 26. Nalezená zranitelnost v programu Acunetix

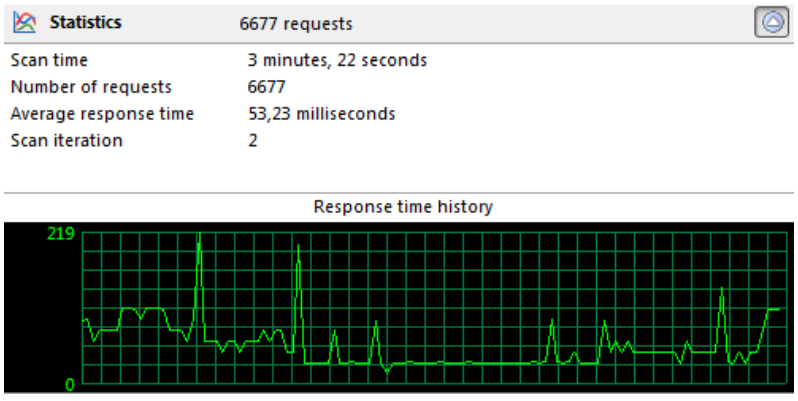
Pro zjištění podrobných informací o nalezených zranitelnostech je třeba se přesunout do pravé části rozhraní programu. Tam je k nalezení seznam všech nalezených zranitelností. Jak je vidět na obrázku, byla nalezena zranitelnost „Email address found". Nejjednodušším vyřešením této zranitelnosti je, jako v předchozím příkladu, zapsání emailové adresy

v jiném tvaru neobsahující znak zavináč. Detailní informace o zranitelnosti se uživatel dozví, když ji označí. Na obrázku je popis zranitelnosti „Email address found".



Obr. 27. Nalezená zranitelnost „Email address found"

Při provádění těchto penetračních testů jsou vynakládány velké požadavky na webový server. Při rozbalení položky Statistics ve výsledcích testu se lze dočíst, že skenování stránek trvalo celkem 3 minuty a 22 sekund, během této doby bylo položeno serveru celkem 6677 požadavků. Z tohoto důvodu nejsou provozovatelé webhostingových služeb zrovna rádi, že někdo na jejich serverech takové testy provádí a „zbytečně" je zatěžuje. Grafický průběh je viditelný na obrázku zobrazeném pod tímto textem.

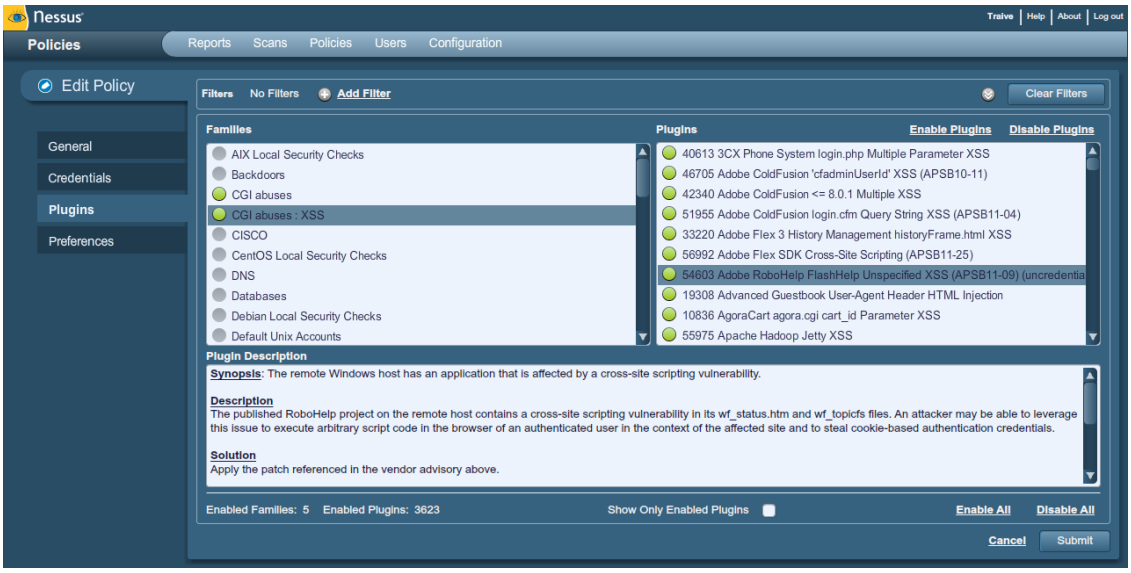


Obr. 28. Časová statistika testu v Acunetix WVS

6.5.3 Nessus – test webového serveru

Nessus se liší od ostatních testovaných web scannerů online rozhraním v prohlížeči. Program je možno plně využívat po obdržení aktivačního klíče zdarma celých 15 dní.

Nessus se pohybuje na prvních místech v žebříčku testovacích rozhraní pro webové aplikace, zobrazen je na obrázku níže. [30] V současnosti nabízí databázi téměř 48 000 zranitelností ve 42 kategoriích. Lze si detailně nastavit jaké knihovny a zranitelnosti v nich testovat. Nastavené rozhraní je třeba uložit jako politiku testování, na jejímž základě je proveden test.



Obr. 29. Nastavení politiky testování v rozhraní Nessus

Pro provedení testu byly vybrány knihovny CGI, řešící vstupy od uživatele (CGI abuses, CGI abuses: XSS). Základní informace o webovém serveru a službám na něm běžících byly zjišťovány pomocí knihovny General. Zde je prováděn test zjištění operačního systému, informace o SSL, výrobce serveru, nebo zda stránky běží pouze na virtuálním serveru. Dalším podstatným bodem pro to, aby byl proveden kompletně test webového serveru, je zatržení kategorie Settings a Web Servers. Informace obsažené v Settings mohou upozornit na místa, které nemohly být oskenovány. Web Servers testuje, zda je server náchylný na útoky XSS, SQL Injection a jiné. Jestli lze získat informaci o verzi vzdáleného serveru, o narušení certifikátu serveru, či jiné narušení bezpečnosti.

Webový server i stránky byly otestovány, nalezené problémy viditelné na obrázku, se však týkaly výhradě serveru, nikoliv stránek samotných. Stránky a server byl testován celkem na 3623 zranitelností. Po dokončení testu byly ve výsledcích zobrazeny celkem 2 informativní hlášky.

Plugin ID ▲	Count ▼	Severity ▼	Name	Family
10287	1	Info	Traceroute Information	General
12053	1	Info	Host Fully Qualified Domain Name (FQDN) Resolution	General

Obr. 30. Nalezené zranitelnosti webovým rozhraním Nessus

První se týkala traceroute, tedy cesty od zdroje požadavku k zadanému cíli. Aby hackeři nemohli kompletně tímto požadavkem mapovat celou cestu, konec sledování trasy většinou končí na rozhraní veřejné a lokální sítě.

Plugin ID: 10287 **Port / Service:** general/udp **Severity:** Info
Plugin Name: Traceroute Information

Synopsis: It was possible to obtain traceroute information.

Description
Makes a traceroute to the remote host.

Solution
n/a

Risk Factor: None

Plugin Output
For your information, here is the traceroute from 10.0.0.2 to 176.9.114.0 :
10.0.0.2
10.0.0.138
88.103.200.71
88.103.203.137
80.188.33.245
91.210.16.12
81.209.172.21
213.239.242.214
213.239.240.157
?

Plugin Publication Date: 1999/11/27

Obr. 31. Mapování cesty k IP adrese – traceroute

Na obrázku je vidět, že není zobrazena celá cesta až ke koncové IP. Routery lokální sítě jsou tedy nastaveny tak, aby na takové požadavky neodpovídaly. Pokud by takové nastavení nebylo provedeno, byla by hackerovi prozrazena kompletní cesta k dané IP adrese a mohl by vymezit nejzranitelnější místo a napadnout jej. Takto, ale nebezpečí nehrozí.

Další nalezená záležitost říká, že bylo možné přeložit název vzdáleného hostitele. IP adresa 176.9.114.0 byla úspěšně přeložena na www.praminekskolka.cz. Jedná se pouze o informativní záležitost a překlad IP adresy na doménový název zde nepředstavuje žádné riziko.

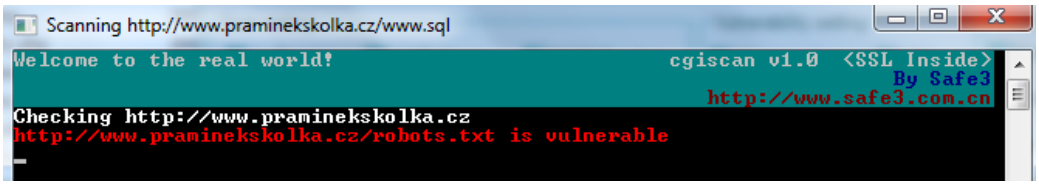
Obr. 32. Překlad IP adresy na doménu

Celkově byl web a server označen za bezpečný a testem nebyly prokázány žádná slabá místa. Nebyly nelezeny žádné zranitelnosti ve zdrojovém kódu. Veškerá upozornění se zde týkala pouze serveru.

6.6 Test zranitelnosti webových stránek – nekomerční scannery

6.6.1 Safe3 Web Vulnerability Scanner

Prvním testovaným programem byl Safe3 Web Vulnerability Scanner, který je řazen dle průzkumu http://sectooladdict.blogspot.com [29] mezi scannery s nejhoršími funkcemi. Navíc funguje pouze pod OS Windows. Přesto aplikace detekovala na stránce zranitelnost. Nalezená zranitelnost nesla název robots.txt. Ale grafický výstup byl zarážející.

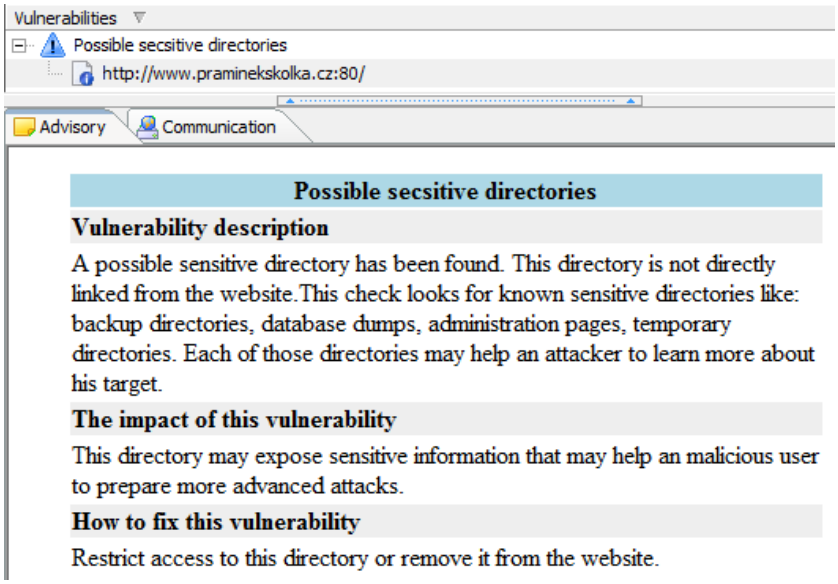


Obr. 33. Nalezená zranitelnost programem Safe3 Web Vulnerability Scanner

Výsledky byly zobrazeny pouze probliknutím okna, které je vidět na obrázku. Nevyskytuje se zde ani žádná dodatečná informace o nalezených zranitelnostech. Nejjednodušší způsob jak vyřešit tuto zranitelnost je soubor robots.txt kompletně odstranit nebo neuvádět cesty k vyhledávači nedostupným adresářům.

6.6.2 JSKY

Druhým testovaným a volně použitelným programem je JSKY. Obsahuje několik testovacích módů, kde je možné si vybrat zranitelnosti, na které je třeba stránky otestovat. Umí provádět například test na Cross-Site Scripting (XSS), detekovat možné citlivé adresáře nebo nezabezpečené objekty. Scanner funguje na Linuxu, OS X i na operačním systému Windows. [28]



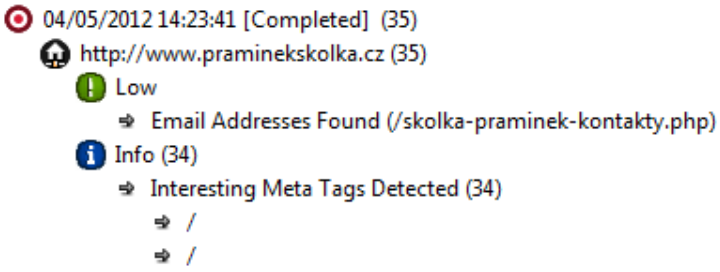
Obr. 34. Nalezená zranitelnost v programu JSKY

Na obrázku je nalezená zranitelnost spadající do kategorie nízkého nebezpečí. Ačkoliv se nejednalo o pouhý informativní charakter zranitelnosti, do této kategorie bych ji zařadil. Jedná se totiž pouze o informaci, která říká: „pozor, tento adresář může být citlivý“. Program nalezl za adresou lomítko, a proto takto jednal. Nenacházejí se zde ale žádné dočasné adresáře nebo citlivé databáze klientů, po zadání adresy http://www.praminekskolka.cz/ se zobrazí pouze hlavní stránka.

6.6.3 Vega

Vega je free software umožňující rychlé testování webové aplikace. Program běží na Linuxu, OS X i Windows. Stejně jako u téměř všech předchozích komerčních nástrojů, tak i zde je rozřazení nalezených zranitelností do čtyř skupin: info, low, medium a high. Vega dokáže odhalit například, Cross-Site Scripting (XSS), nedopatřením zveřejněné citlivé informace a další zranitelnosti. [27]

Na testovaných stránkách našel tento webový scanner celkem 35 zranitelností, z nich 34 informativního charakteru. Zranitelnost nízké úrovně se týkala možnosti napadení emailové adresy, zbylé informativní upozorňovaly na to, že za lomítkem, které je součástí adres by mohl útočník nalézt citlivý obsah. Žádný citlivý obsah nenachází, tím pádem není třeba se těmito záležitostmi dále zaobírat. Souhrn zranitelností je zobrazen na obrázku.



Obr. 35. Nalezené zranitelnosti programem Vega

6.7 Test zranitelnosti webových stránek – Souhrnné výsledky

Bylo odzkoušeno dohromady 6 scannerů webových aplikací a to jak s volnou licencí, tak s omezenou. Celkem dva webové scannery označily jako nebezpečí soubor robots.txt, který obsahoval zakázaný přístup pro vyhledávací roboty do adresáře /stats/. Zde se jednalo pouze o statistiky návštěvnosti webových stránek, které jsou defaultně webhostingem chráněny. Častou chybou bývá, že je jediným řešením zakázání přístupu adresáře či stránky pouhé vyřazení z výsledku vyhledávání. Pokud by si útočník stáhl soubor robots.txt, mohl by jednoduše zjistit, kde může najít potenciálně citlivý obsah. Soubor robots.txt byl z webu vymazán.

Tři scannery označily určité adresáře na webu za zdroj citlivých informací, dokonce i kořenovou doménu stránky, citlivé informace však neobsahovaly a tudíž nebyly nijak dodatečně chráněny. Znak @ na stránce označila polovina scanneru za nebezpečí šíření nevyžádané pošty.

Souhrnné výsledky jsou uvedeny taky v tabulce 2. Zelenou barvou je znázorněno nalezení zranitelnosti, červenou barvou nenalezení a modře je zobrazeno pouze políčko programu Nessus, který našel zranitelnosti na serveru, nikoliv na stránkách.

Název scanneru	Nalezeno / řešeno hrozeb	Robots.txt	Citlivé adresáře	E-mailová adresa
Netsparker	3 / 2	Opraveno	Neobsahují je	Opraveno
Acunetix WVS	1 / 1	Nenalezeno	Nenalezeno	Opraveno
Nessus	2 / 0 (server)	Nenalezeno	Nenalezeno	Nenalezeno
Safe3 Web	1 / 1	Opraveno	Nenalezeno	Nenalezeno
JSKY	1 / 0	Nenalezeno	Neobsahují je	Nenalezeno
Vega	2 / 1	Nenalezeno	Neobsahují je	Opraveno

Tabulka 2. Souhrnné výsledky testování zranitelnosti webových stránek

V rámci testování webové stránky www.praminekskolka.cz uspěl nejvíce Netsparker, který označil největší počet potencionálních nebezpečí. V rámci testování bezpečnosti webového serveru se ukázal jako nejvýhodnější program Nessus. Provedením testů bylo ověřeno, že je třeba se zaměřit vždy raději na více testujících mechanismů od různých tvůrců. Není možné se spoléhat na jeden konkrétní produkt.

7 ZABEZPEČENÍ SERVERU

7.1 Rozdělení zabezpečení serveru

Co se zabezpečení serveru týče, je možné ho pojmout z více úhlů pohledů, jež jsou znázorněny na obrázku. Bezpečnost můžeme chápat z pohledu fyzického zabezpečení prostor a serveru samotného, definování portů, které mohou být používány, nebo z hlediska nainstalovaných programů na serveru (včetně operačního systému).

Obr. 36. Znázornění zabezpečení serveru

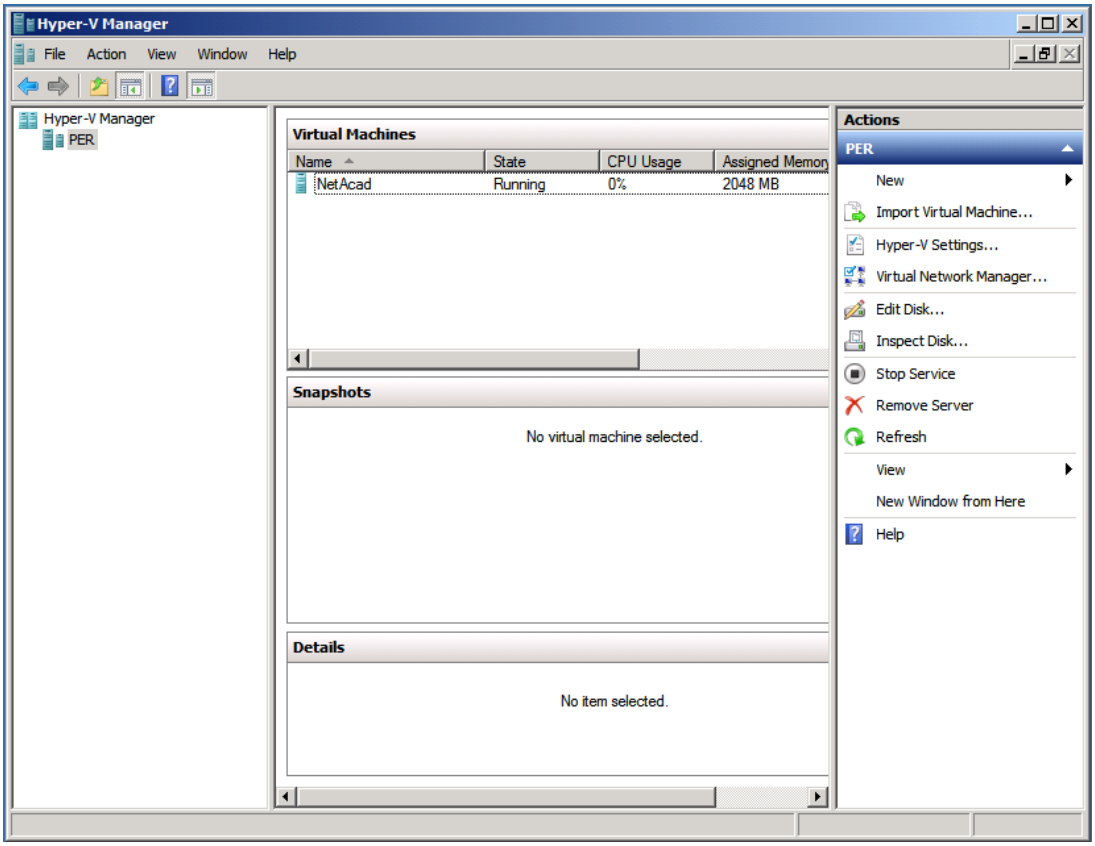
7.2 Platforma serveru a hardware

Z důvodu běhu stránek na serveru Axfone (poskytovatele webhostingových služeb), nebylo možné zasahovat do nastavení serveru. Proto byly stránky, po dobu testování, dočasně přesunuty na školní server s IP adresou 195.178.89.58.

Obr. 37. Windows Server 2008 R2 Standard

Byla využita 64 bitová verze operačního systémemu Windows Server 2008 R2 Standard, a jeho služba Hyper-V, díky níž je možný provoz více virtuálních serverů na jednom

fyzickém stroji. Byly využity služby z balíku XAMPP (server Apache, PHP a FileZilla FTP Server).



Obr. 38. Rozhraní Hyper-V v OS Windows Server 2008 R2 Standard

Hardware na němž byl OS nainstalován byl osazen dvoujádrovým procesorem Intel Xenon 5140 o frekvenci 2,33 GHz a paměťí RAM o celkovém prostoru 4 GB. Jednalo se o běžný počítač PC.

7.3 Fyzická bezpečnost

Server samotný i přístup do prostor, v němž se server vyskytuje, je třeba chránit před poškozením či jakoukoliv manipulací. V případě, že se jedná o server s nutností funkce při výpadku proudu nebo o rozsáhlé serverové sály, je nutné je zabezpečit náhradním zdrojem energie a mít SHZ (stabilní hasicí zařízení) před možným vznikem požáru. Hašení probíhá za pomocí plynů, většinou se jedná o Inergen (směs dusíku, argonu a oxidu uhličitého). [42]

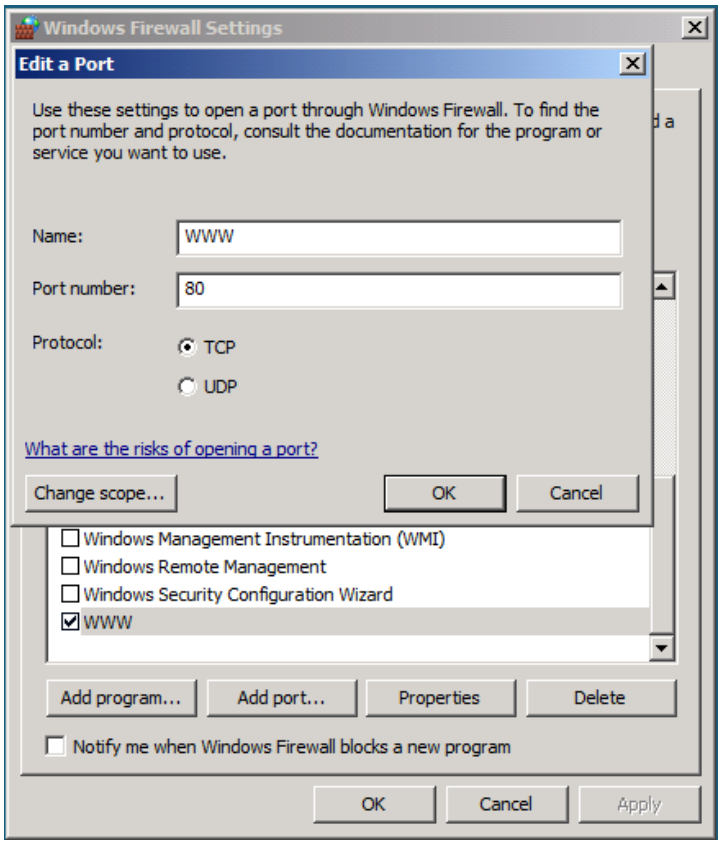
Testovaný server byl umístěn na Univerzitě Tomáše Bati ve Zlíně v budově U5 v místnosti spolu s dalšími servery. Tato místnost byla zamčena klíčem a vstup kontrolován

přístupovým systémem. Ten identifikoval osobu na základě přiložené karty a následně vyžadoval zadání kombinace čísel.

7.4 Bezpečnost na perimetru sítě

7.4.1 Firewall

Tuto bezpečnost je třeba chápat jako definici jasných pravidel, které zabezpečí přístup k datům umístěných v počítači. Jedná se zde o průnik do systému přes otevřený port. V nastavení Firewallu je možné si vytvořit seznam povolených čísel portů. V zásadě by mohl vystačit pouze jeden s číslem 80, který je určen pro HTTP protokol, ale nefungovala by komunikace přes FTP, proto je třeba nastavit jako povolený i port 21.



Obr. 39. Nastavení Windows Firewall – úprava portu

Pokud by tyto dva porty nebyly dostačující, stačí vybrat další port z nabídky, nebo přidat vlastní přes tlačítko Add port. V nastavení Firewallu byly vybrány celkem 3 otevřené porty, zmíněné 80 a 21 a taky 3389 použitý k přístupu přes vzdálenou plochu.

Nejlepší jak si ověřit otevřené porty je provést mapování serveru (například pomocí nástroje Nmap, nyní Zenmap). Pomocí něj byla otestována i IP adresa 195.178.89.58 s výsledky zobrazenými na obrázku:

Port	Protocol	State	Service	Version
21	tcp	open	ftp	
80	tcp	open	http	
3389	tcp	open	ms-term-serv	

Obr. 40. Test otevřených portů programem Zenmap

7.4.2 Silná hesla

Přístup přes port vzdálené plochy může být nebezpečný v případě, že jsou volena slabá hesla, tedy hesla krátká, složená ze slov, bez čísel, bez rozlišování velkých a malých písmen a speciálních znaků. Obecně platí pravidlo, že čím, je heslo delší a využívá více znaků, tak tím je odolnější. Heslo Np3,ks7nn. je možné si zapamatovat podle citátu „Neuspějí pouze ti, kteří se o nic nepokusí.“. Jedná se o počáteční písmena slov, kdy písmeno o je nahrazeno číslem 7 a písmeno t číslem 3 (dle pořadí znaků). Kromě složitosti hesla je výhodné změnit defaultní jméno uživatele a hesla průběžně měnit.

Obr. 41. Změna hesla u uživatele Administrator

Na obrázku uvedený uživatel Administrator je značně nebezpečný. Nelze opomenut totiž fakt, že Administrator je běžně používaný název uživatele a v případě slabého hesla, by

tento defaultní název mohl představovat vysokou zranitelnost (například při realizování připojení přes vzdálenou plochu).

7.5 Softwarová bezpečnost

Nejideálnějším řešením webového serveru je, aby na něm byly instalovány nejnovější verze a aktualizace operačních systémů, antivirových a antispyware programů. Je třeba rychle reagovat a nejlépe tyto aktualizace automaticky instalovat, aby časová prodleva mezi vydáním a instalací záplaty byla co nejnižší.

Na server je ideální pohlížet jako na zařízení, které pracuje pouze a výhradně za jedním účelem, jako server. Ačkoliv je možné na něj instalovat další programy, výhradně pro osobu na počítači (serveru) pracující, není to z hlediska bezpečnosti vhodné. Pokud je to však nutné a server je využíván i jako běžný počítač pro denní potřeby uživatele, je nutné k němu přistupovat o to obezřetněji.

ZÁVĚR

Cílem této diplomové práce bylo poukázat na problematiku webových stránek z pohledu jejich celkové bezpečnosti. Zkoumány byly webové technologie HTML, CSS, PHP a JavaScript. Pod pojmem bezpečnosti webových prezentací je třeba si představit nejen bezchybně napsaný zdrojový kód odolný vůči známým zranitelnostem, ale i bezpečný provoz stránek na webovém serveru a správné zobrazení v nejrůznějších webových prohlížečích.

V teoretické části díla jsou popsány zranitelnosti, které hrozí jak webovým stránkám, tak webovým serverům. Současně jsou řešeny postupy, jak je možné se proti těmto nebezpečím kyberterorismu chránit. Nechybí ani přímé ukázky zdrojových kódů a nastavení, které proti vybraným útokům webové stránky a servery chrání či útoky výrazně eliminují. Kromě bezpečnosti není opomenuta ani optimalizace pro vyhledávače, která má podstatný vliv na nalezení webové stránky přes internetový vyhledávač. Popsány jsou zásadní optimalizační techniky vedoucí k umístění stránek na horní pozice ve výsledcích vyhledávání.

Praktická část práce je tvořena celkem třemi bloky. Prvním z nich je vytvoření webové prezentace pro příspěvkovou organizaci (Mateřská Škola Pramínek, Valašská Bystřice) s respektováním zásad uvedených v teoretické části díla. V další části je stránka testována na správné zobrazení v nejpoužívanějších prohlížečích, na zobrazení ve výsledcích vyhledávání, validitu kódu a především na zranitelnosti. Webové stránky i server jsou softwarovými scannery otestovány a důležité nalezené zranitelnosti jsou opraveny. Díky výběru více produktů od různých výrobců je možné rozpoznat větší množství potenciálních nebezpečí.

Poslední blok praktické části je zaměřen na zabezpečení serveru, na němž jsou stránky umístěny. Obsah vytvořených stránek (www.pramineckskolka.cz) je však umístěn na serveru poskytovatele webhostingových služeb, proto nebylo možné zajištění ani zjištění jeho zabezpečení. Poskytovatelé totiž podávají, z důvodu bezpečnosti, co nejmenší množství informací. Protože bylo třeba splnit bod zadání, byly stránky dočasně, se svolením zadavatele, zkopírovány na školní server a na něm provedeny potřebné testy a nastavení. Zdrojové kódy webových stránek jsou poskytnuty na CD v adresáři s názvem

webove_stranky. Obsah zde byl umístěn až po souhlasu ze strany zadavatele projektu webových stránek.

Vždy je obtížné formulovat budoucí vývoj a v síti Internet tomu není jinak. Webové stránky nefungují ani tři desetiletí, takže je zde vše poměrně nové a bezpečnostní prognózy složité. Myslím si, že díky stále rozrůstajícímu se počtu doménových jmen je taky více potencionálních konkurentů a tím pádem více možných útoků na webová sídla. Motivací vždy nemusí být jen konkurenční boj, ale i jiné faktory jako například seberealizace. Dle mého názoru, je budoucnost v narůstajícím počtu dokonalejších robotů, kteří si budou klást za úkol ničení nedostatečně chráněných webových stránek. Proto bude třeba vymýšlet nové sofistikovanější ochranné metody. Z hlediska webových serverů by bylo možné předpokládat stále větší přesouvání provozu webových služeb do virtuálního rozhraní, a doplňování jich o další hardwarové zařízení bránící jejich napadení. To co se bude dít není nikdy úplně jasné, jedna zásada bude ale platit vždy, tedy počítat se vším co se může stát a raději chránit web a webový server co nejlépe a všemi dostupnými prostředky.

ZÁVĚR V ANGLIČTINĚ

The aim of this thesis was to highlight issues of web sites in terms of their overall security. Among investigated web technologies were HTML, CSS, PHP and JavaScript. The term 'web security' does not necessarily indicate perfectly written source code resistant to known vulnerabilities, but also safe operation of the site on the web server as well as correctly displaying in various web browsers.

The theoretical part of the work describes vulnerabilities that threaten both web sites and the web servers themselves. At the same time, procedures that deal with this potential 'cyber-terrorism' are also investigated. Direct source code samples and settings that protect against or eliminate potential threats are also explained and listed. In addition to security, search engine optimization, which plays a significant part in finding a web page via the internet, is also thoroughly investigated. Described are fundamental optimization techniques that help place pages in top search result positions.

The practical part consists of three blocks. The first is to create websites for contributory organizations (Kindergarten Pramínek, Valašská Bystřice) with respect to the principles set out in the theoretical part of the work. In the next part, the page is tested for correct display on the most popular browsers, the validity of the code and, most importantly, the vulnerability. Web pages and their associated server is software-scan tested and any discovered vulnerabilities are corrected. Due to the choice of various products from different manufacturers, more potential threats can be uncovered.

The last section of the practical part focuses on the security of the server where the site is located. The content of created pages (www.praminekskolka.cz) is located on an external web hosting service provider's server, thus it was not possible to determine or ensure its security. The provider allows access to, for safety reasons, the least amount of information possible. Because it was necessary to meet the project requirements, the site was temporarily, with the consent of the contracting authority, copied to the school server where the necessary tests could be performed. The source code of the afore-mentioned web pages are provided on a CD in the directory titled webove_stranky. The content was placed here with approval from the sponsor of the project site.

It is always difficult to formulate future development and the Internet is no different. Web pages barely exist for three decades, so everything is relatively new and security predictions

complex. I think that thanks to continued expansion in the number of domain names, more potential competitors come forth and thus more potential attacks on web sites. The motivation may not always be competition, but also factors such as self-realization. The vision of the future is, in my opinion, an increasing number of sophisticated robots put to the task of destroying inadequately protected websites. It will therefore be necessary to invent new, more sophisticated protection methods. In terms of web servers, it may be possible to assume an increasing shift in web service traffic to virtual interfaces, with additional hardware components assisting in their protection. What will happen is never entirely clear, but one principle will always apply and that is to expect anything and preferably protect the web server and its content with all means available.

SEZNAM POUŽITÉ LITERATURY

[1] *TOPlist - audit návštěvnosti webových stránek* [online]. © 1997-2012 [cit. 2012-03-08]. Dostupné z: http://toplist.cz/

[2] *Seznam – Najdu tam, co hledám* [online]. © 1996-2012 [cit. 2012-03-08]. Dostupné z: http://www.seznam.cz/

[3] KUBÍČEK, Michal a Jan LINHART. *333 tipů a triků pro SEO*: *Sbírka nejlepších technik optimalizace webů pro vyhledávače*. Brno: Computer Press, 2010. ISBN ISBN 978-80-251-2468-0.

[4] *SEO optimalizace a aktivní SEO servis pro vyhledávače* [online]. © 2008-2011 [cit. 2012-03-08]. Dostupné z: http://www.seo-pruvodce.cz/

[5] JANOVSKÝ, Dušan. *Jak psát web* [online]. [2012] [cit. 2012-03-08]. Dostupné z: http://www.jakpsatweb.cz

[6] *Otestujte si kvalitu SEO on-line | SEO analyzátor.cz* [online]. © 2011-2012 [cit. 2012-03-08]. Dostupné z: http://www.seo-analyzator.cz/

[7] *Builder - Informacni server o programovani* [online]. © 1997-2003 [cit. 2012-03-08]. Dostupné z: http://www.builder.cz/

[8] *Majestic SEO : Site Explorer* [online]. [2012] [cit. 2012-03-08]. Dostupné z: https://www.majesticseo.com/

[9] KOSEK, Jiří. *HTML5*: *Tvorba dokonalých webových stránek* [online]. © 1997–2010 [cit. 2012-04-24]. Dostupné z: http://htmlguru.cz

[10] Web slaví 20 let, zrodil se jako vedlejší produkt atomového výzkumu. KASÍK, Pavel. *Technet.cz*: *Technika kolem nás* [online]. 6.8.2011 [cit. 2012-03-13]. Dostupné z: http://technet.idnes.cz/web-slavi-20-let-zrodil-se-jako-vedlejsi-produkt-atomoveho-vyzkumu-1ca-/sw_internet.aspx?c=A110805_162149_sw_internet_pka

[11] W3C. *World Wide Web Consortium (W3C)* [online]. © 2012 [cit. 2012-03-13]. Dostupné z: http://www.w3.org/

[12] *Tvorba WWW*: *o webdesignu, grafice a reklamě* [online]. © 2003-2008 [cit. 2012-03-13]. Dostupné z: http://www.tvorba-webu.cz/

[13] VRÁNA, Jakub. *1001 tipů a triků pro PHP*. Brno: Computer Press, 2010. ISBN ISBN 978-80-251-2940-1.

[14] CAPTCHA – past na roboty, ale také lidi. *Cnews.cz* [online]. 4.5.2011 [cit. 2012-03-22]. Dostupné z: http://www.cnews.cz/captcha-past-na-roboty-ale-take-lidi

[15] Potřebujete obejít CAPTCHA? Zaplaťte si armádu Indů. *Root.cz* [online]. 2.8.2010 [cit. 2012-03-22]. Dostupné z: http://www.root.cz/clanky/potrebujete-obejit-captcha-zaplatte-si-armadu-indu/

[16] CAPTCHA jak ji možná neznáte. *CleverAndSmart - ICT management* [online]. 15.08.2011 [cit. 2012-03-22]. Dostupné z: http://www.cleverandsmart.cz/captcha-jak-ji-mozna-neznate/

[17] *SYMBIO - Creative Digital Agency* [online]. © 1999–2012 [cit. 2012-04-02]. Dostupné z: http://www.symbio.cz/

[18] Pokročilá antispamová ochrana formulářů. *NET-VORův blok* [online]. 9.5.2010 [cit. 2012-03-23]. Dostupné z: http://blok.net-vor.cz/pokrocila-antispamova-ochrana-formularu/

[19] VRÁNA, Jakub. Ukládání souborů od uživatele. *Jakub Vrána* [online]. 28.12.2005 [cit. 2012-03-26]. Dostupné z: http://php.vrana.cz/ukladani-souboru-od-uzivatele.php

[20] .htaccess - změna upload_max_filesize. *Tipy pro tvorbu www stránek* [online]. 2012 [cit. 2012-03-26]. Dostupné z: http://tvorbastranek.okamzite.eu/htaccess-zmena-upload_max_filesize.html

[21] Škodlivý kód na www stránkách. *SvetHostingu.cz*: *webhosting, místo pro vaše www stránky* [online]. © 1998-2012 [cit. 2012-03-26]. Dostupné z: http://www.svethostingu.cz/podpora/?ind=208

[22] Únorové statistiky: všechny hlavní prohlížeče navýšily své tržní podíly. *ExtraWindows*: *Vše, co potřebujete vědět o Windows* [online]. 2.3.2012 [cit. 2012-03-27]. Dostupné z: http://extrawindows.cnews.cz/comment/18999

[23] Internetové prohlížeče: Jaký si vybrat?. *Tvorba webových stránek* [online]. 28.04.2011 [cit. 2012-03-27]. Dostupné z: http://www.web-tvorba.com/clanky/internetove-prohlizece-jaky-si-vybrat/

[24] PERNICKÝ, Štěpán. Podíly webových prohlížečů za měsíc říjen. *Online časopis o počítačích - PC Magazín* [online]. 10.11.2011 [cit. 2012-03-27]. Dostupné z: http://www.pcmagazin.cz/internet/podily-webovych-prohlizecu-za-mesic-rijen

[25] *Website Security - Acunetix Web Security Scanner* [online]. © 2012 [cit. 2012-04-03]. Dostupné z: http://www.acunetix.com/

[26] *Netsparker, False Positive Free Web Application Security Scanner - Mavituna Security* [online]. © 2012 [cit. 2012-04-03]. Dostupné z: https://www.mavitunasecurity.com/

[27] Vega – Open Source Cross Platform Web-Application Security Assessment Platform. *Darknet - The Darkside*: *Ethical Hacking, Penetration Testing & Computer Security* [online]. 5.6.2011 [cit. 2012-04-05]. Dostupné z: http://www.darknet.org.uk/2011/07/vega-open-source-cross-platform-web-application-security-assessment-platform/

[28] JSKY - Free Vulnerability Scanner. *Pro Hack* [online]. 2011 [cit. 2012-04-05]. Dostupné z: http://www.theprohack.com/2010/03/jsky-free-vulnerability-scanner.html

[29] The Scanning Legion: Web Application Scanners Accuracy Assessment & Feature Comparison. *Security Tools Benchmarking* [online]. 1.8.2011 [cit. 2012-04-05]. Dostupné z: http://sectooladdict.blogspot.com/

[30] Nessus 5.0 is Here. *Tenable Network Security* [online]. © 2002-2012 [cit. 2012-04-10]. Dostupné z: http://www.tenable.com/products/nessus

[31] Denial of Service. *CompuNet*: *Network desing, supervison and security* [online]. © 2012 [cit. 2012-04-16]. Dostupné z: http://www.compunet.cz/denial-of-service/

[32] BEZPEČNOST NA PERIMETRU SÍTĚ: BEZPEČNOST NEJEN NA PERIMETRU SÍTĚ S VYUŽITÍM IPS. *AG COM* [online]. © 2012 [cit. 2012-04-16]. Dostupné z: http://www.agcom.cz/cs/reseni-pro-ict/zvysovani-bezpecnosti-ict/bezpecnost-na-perimetru-site.shtml

[33] Moduly pro Apache I – mod_evasive. *GLUX.org* [online]. © 2009-2011 [cit. 2012-04-16]. Dostupné z: http://glux.org/2009/11/moduly-pro-apache-i-mod_evasive/

[34] Vytvoření silného hesla a jeho vlastnosti. *Bezpečný internet* [online]. © 2012 [cit. 2012-04-16]. Dostupné z: http://bezpecnyinternet.cz/zacatecnik/hesla/vytvoreni-silneho-hesla.aspx

[35] Bezpečnosť servera v sieti. *Deja-vix* [online]. 21.3.2007 [cit. 2012-04-16]. Dostupné z: http://deja-vix.sk/sysadmin/security.html

[36] BERTINO, Elisa. *Security for web services and service-oriented architectures*. Berlin: Springer, 2010. ISBN 978-3-540-87742-4.

[37] HOWARD, Michael a David LEBLANC. *Bezpečný kód*: *Techniky a strategie tvorby bezpečných webových aplikací*. Brno: Computer Press, 2008. ISBN 978-80-251-2050-7.

[38] LECKY-THOMPSON, Ed a Steven D. NOWICKI. *PHP 6*: *Programujeme profesionálně*. Brno: Computer Press, 2010. ISBN 978-80-251-3127-5.

[39] HUSEBY, Sverre H. *Zranitelný kód*. Brno: Computer Press, 2006. ISBN 80-251-1180-6.

[40] ECCHER, Clint. *Profesionální webdesign*: *techniky a vzorová řešení pro XHTML a CSS*. Brno: Computer Press, 2010. ISBN 978-80-251-2677-6.

[41] PADRTA, Aleš. Vulnerabilities - Zranitelnosti. In: *Seznámení, technické základy a typy zranitelností* [online]. 2009 [cit. 2012-04-17]. Dostupné z: http://www.cesnet.cz/akce/2009/zazemi-pro-cert-csirt/p/vulnerabilities.pdf

[42] Inergen. *Klika BP* [online]. 2012 [cit. 2012-04-27]. Dostupné z: http://www.klika.cz/aktivni-pozarni-systemy/plynove-hasici-systemy/stabilni-hasici-zarizeni-s-inertnimi-hasivy/inergen/

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

3D	Trojrozměrný
CAPTCHA	Completely automated public Turing test to tell computers and humans apart
CD	Compact Disc
CERN	European Organization for Nuclear Research
CGI	Common Gateway Interface
CSS	Cascading Style Sheets
DDoS	Distributed Denial of Service
DoS	Denial of Service
DTD	Document Type Definition
FTP	File Transfer Protocol
GB	Gigabajt
GHz	Gigahertz
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IPS	Intrusion Prevention System
ISO	International Organization for Standardization
ISS	Internet Information Services
JPEG	Joint Photographic Experts Group
kB	Kilobajt
MB	Megabajt
OCR	Optical Character Recognition
OS	Operační systém

OSI	Open Systems Interconnection
PDF	Portable Document Format
PHP	PHP: Hypertext Preprocessor
PNG	Portable Network Graphics
PR	Public Relations / Page Rank
px	Pixel
RAID	Redundant Array of Inexpensive/Independent Disks
RAM	Random-access memory
RSS	Really Simple Syndication
SEO	Search Engine Optimization
SHZ	Stabilní hasicí zařízení
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UCS	Universal Character Set
URL	Uniform Resource Locator
UTF-8	UCS Transformation Format
W3C	World Wide Web Consortium
WHATWG	Web Hypertext Application Technology Working Group
WWW	World Wide Web
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
XSS	Cross-site scripting

POJMY POUŽÍVANÉ V SOUVISLOSTI S BEZPEČNOSTI V IT

Bezpečnost

Bezpečnost je stav, při němž je možné naplňovat bez problému činnost informačního systému. Bezpečnost je nutno brát v dnešní době jako součást informačních systémů, nejen jako dodatečnou nadstavbu. [36]

Hrozba

Hrozba je v informační bezpečnosti definována jako jakákoliv okolnost či událost, jejíž potenciál může negativně ovlivnit správnou činnost informačního systému i jeho reputaci. Ovlivnění je prováděno prostřednictvím neoprávněného přístupu, zničení, zpřístupnění, změnou informací nebo vyřazení služby z provozu. [36]

Zranitelnost

Zranitelnost definuje slabé místo služby či její části, prostřednictvím něj může být naplněna hrozba. Jedná se o vyjádření, jak je citlivý informační systém na působení hrozby. Zranitelnosti se dělí podle jejich závažnosti do několika úrovní. Zranitelnost zkoumá, jak daná hrozba může poškodit chráněná aktiva. [36]

Útok

Útokem je označován úmyslný postup vedoucí k narušení, napadení či využití informačního systému. Je realizován úmyslným využitím zranitelností. Jeho cílem je poškodit určitým způsobem subjekt, proti němuž je útok veden. Osoba, která provádí útok, je nazývána útočníkem. [36]

Citlivá data

Za citlivá jsou považována taková data, jejichž únik nebo modifikace může způsobit konflikt zájmů osob a organizací, které s takovými daty pracují, nebo informace v nich uvedeny vypovídají o nich samotných. Proto je třeba tyto data v informačních systémech ukládat, zpracovávat i přenášet bezpečně. [36]

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: VZOROVÝ HTML DOKUMENT

```
<!doctype html public "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!-- definice verze html pro prohlížeč -->

<html>
<!-- začátek formátování html -->

    <head>
    <!-- začátek hlavičky dokumentu -->
        <title>Ukázková webová stránka</title>
        <!-- název dokumentu -->
        <meta http-equiv="content-type" content="text/html; charset=windows-1250">
        <!-- použitá znaková sada -->
        <meta name="description" content="Stránka na ukázku.">
        <!-- popis stránky -->
        <!-- další meta-tagy slouží například pro copyright, klíčové slova, autorství,
        informace pro vyhledávače,... -->
    </head>
    <!-- konec hlavičky dokumentu -->

    <body>
    <!-- tělo dokumentu dokumentu -->
        <h1 title="Popisek prvního nadpisu">Nadpis 1</h1>
        <!-- Nadpis první úrovně - po najetí na něj se zobrazí "Popisek prvního nadpisu" -->
        <h2 align="center">Nadpis X</h2>
        <!-- Nadpis druhé úrovně zarovnaný na střed -->
        <p>Text<p>
        <!-- Odstavec je rovněž párové značky-->
        <img src="http://www.domena.cz/obrazek.jpg" alt="Alternativní popis">
        <!-- nepárová značka je třeba obrázek, src značí jeho adresu,
        alternativní popis je standardem vyžadován, jinak by stránka obsahovala chybu -->
    </body>
    <!-- konec těla dokumentu -->

</html>
<!-- konec formátování html -->
```

Cesta k souboru na CD: ukazky/ukazkova.html

PŘÍLOHA P II: KONTROLA EMAILOVÉ ADRESY V PHP

```php
<?php

function isValidEMailAdress ($email , $checkdns = TRUE) {
    #nastavení maximální délky zadávané adresy
    if (strlen($email) > 50)
        return FALSE;
    #rozdelení na dvě části (před a za znakem @)
    if (!preg_match("/^([^@]+)@(.*)$/", $email, $casti))
        return FALSE;
    $uzivatel = $casti[1];
    $domena   = $casti[2];
    #kontrola správnosti uživatelského jména
    if (!preg_match ("/[^a - zA - Z0 - 9_.+-]/" , $uzivatel))
        return FALSE;
    #kontrola správnosti domény
    if (!isValidDomain($domena, $checkdns))
        return FALSE;
#v případě, že všechno je správně
return TRUE; }


#---------------druhá část kódu testující doménové jméno---------------


function isValidDomain ($domena , $checkdns = TRUE) {
    #nastavení maximální délky zadávané domény
    if (strlen($domena) > 50)
        return FALSE;
    #kontrola správnosti názvu domény
    if (!preg_match ("/[^a - zA - Z0 - 9_.+-]/" , $domena))
        return FALSE;
    #doména musí obsahovat alespoň jednu tečku
    if (!preg_match ("/\\./" , $domena))
        return FALSE;
    if ($checkdns && !checkdnsrr($domena, "ANY"))
        return FALSE;
#v případě, že všechno je správně
return TRUE;
}

?>
```

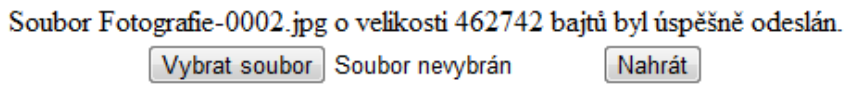
Cesta k souboru na CD: ukazky/email.php

PŘÍLOHA P III: NAHRÁVÁNÍ SOUBORU NA WEBOVÉ STRÁNKY

Zdrojový kód:

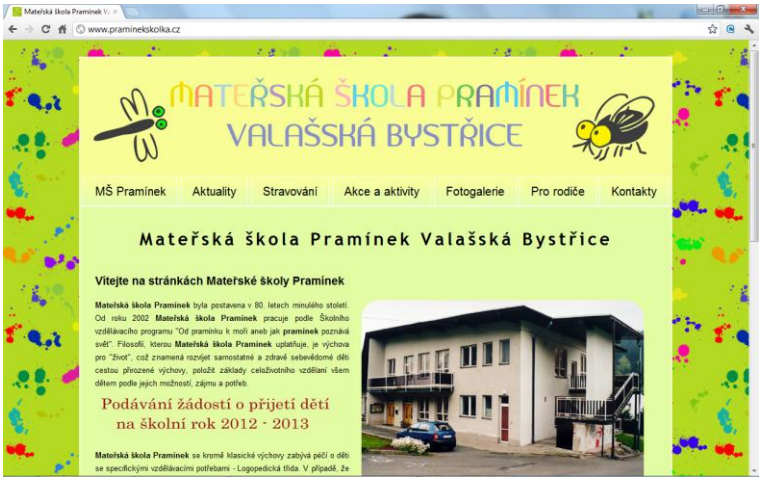
```php
<?php
if (!$_FILES || $_FILES["soubor"]["error"] == UPLOAD_ERR_INI_SIZE) {
    echo "Vyberte soubor, maximální velikost je " . ini_get('upload_max_filesize') . "B.\n";}
if (is_uploaded_file($_FILES["soubor"]["tmp_name"])):
//kontrola zda je vybrán soubor
if (($_FILES["soubor"]["type"]) == "image/jpeg"):
//kontrola typu souboru
  $cesta="foto/";
  //adresář, v němž bude soubor uložen
    if (move_uploaded_file ($_FILES["soubor"]["tmp_name"], $cesta.$_FILES["soubor"]["name"])):
    // přeseun do nastaveného adresáře
        echo "Soubor ".$_FILES["soubor"]["name"]." o velikosti ";
        echo $_FILES["soubor"]["size"]." bajtů byl úspěšně odeslán.";
    else:
        echo "Chyba při nahrávání, zkuste to prosím znovu.";
    endif;
  else:
    echo "Soubor není typu JPEG/JPG! Nelze nahrát.";
  endif;
else:
  echo ""; //další nepředvídaná chyba
endif;
?>

<form action="testup.php" method="post" enctype="multipart/form-data">
    <input name="soubor" type="file">
    <input value="Nahrát" type="submit">
</form>
```

Zobrazení v prohlížeči:

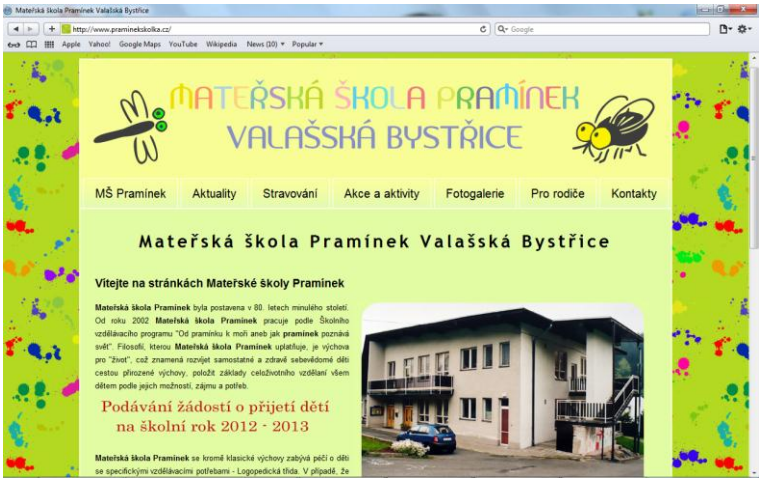


Cesta k souboru na CD: ukazky/foto.php

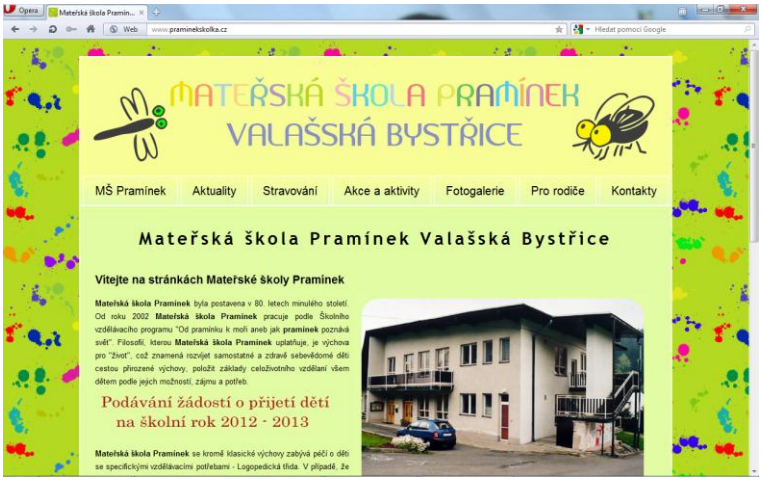
PŘÍLOHA P IV: TEST ZOBRAZENÍ V PROHLÍŽEČÍCH



Zobrazení v prohlížeči Chrome (verze 17.0)



Zobrazení v prohlížeči Safari (verze 5.1.2)



Zobrazení v prohlížeči Opera (verze 11.62)