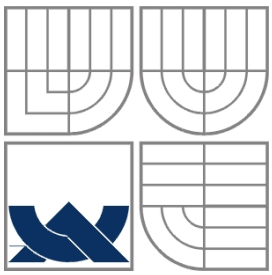
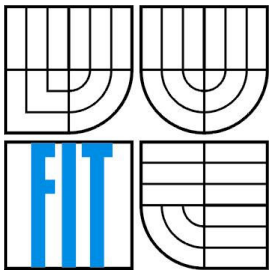


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

# ZPRACOVÁNÍ STATISTIK PŘÍSTUPŮ K WEBOVÝM SYSTÉMŮM

SYSTEM FOR PROCESSING OF STATISTIC INFORMATION OF WEB SYSTEMS ACESSING

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

VLADAN ZAJDA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. PAVEL OČENÁŠEK

BRNO 2007

# Abstrakt

Tato práce se zabývá zpracováním statistik přístupů k internetovým stránkám a systémům, které slouží ke zjištění cenných informací o jejich uživatelích. Statistiky umožňují zjistit z jakých míst uživatelé přicházejí, kolik času na stránkách tráví, jaké informace je zajímají a jaké technické prostředky k prohlížení stránek používají. Naměřené údaje jsou ukládány pro pozdější vyhodnocení majitelem či správcem sledované internetové aplikace. Výsledky měření se zobrazují v grafické podobě. Získané informace lze využít ke zvýšení kvality internetové aplikace či jejího obsahu, aby co nejlépe vyhovovala potřebám uživatelů.

U zabezpečených částí aplikace lze, při využití modulu pro zabezpečení, sledovat pokusy o neoprávněný přístup k aplikaci. Získané bezpečnostní statistiky mohou posloužit ke zvýšení bezpečnosti aplikace, případně k identifikaci útočníků.



# Klíčová slova

statistiky přístupů, monitorování návštěvnosti, vyhodnocení návštěvnosti, přístup k internetové aplikaci, přístup do zabezpečené oblasti, autorizace uživatele, bezpečnost internetové aplikace, ochrana internetové aplikace před útoky



# Abstract

This study deals with processing of web traffic statistics and traffic statistics of systems in general. Statistics allow finding out where the users came from, how much time they spent on the site, in which information were they interested in, and which technical devices were they using to browse the web site. Measured data are stored fore later evaluation by the owner or administrator of the monitored web site. Results are displayed in form of statistics. Gathered information could be used for quality or content improvement in order to the best satisfaction and benefit to the users.

By secured areas of the application the attempts for not granted access could be tracked. Those gathered information statistic can be used to increase security of the application or eventually for identification of the attackers.



# Keywords

web traffic statistics, visit rate monitoring, visit rate evaluation, access to internet application, access to secured area, user authorization, web application security, web site attack prevention



# Citace

Vladan Zajda: Zpracování statistik přístupů k webovým systémům, diplomová práce, Brno, FIT VUT v Brně, 2007

# Zpracování statistik přístupů k webovým systémům

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Pavla Očenáška.
Další informace mi poskytl Ing. David Martinek.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Vladan Zajda
květen 2007

## Poděkování

Děkuji vedoucímu projektu Ing. Pavlu Očenáškovi za odborné vedení, rady a připomínky, které mi při řešení projektu poskytoval.

# Obsah

# 1 Úvod

Žijeme v době, kdy se internet rozšiřuje nebývalou rychlostí. Roste nejen počet aktivních uživatelů internetu, ale zároveň i počet dostupných internetových stránek a internetových aplikací. S tím roste i potřeba znát informace o jejich návštěvnosti, aby bylo možné vyhodnotit úspěšnost aplikací.

V rámci semestrálního projektu jsem vytvořil systém pro sběr a prezentaci statistik přístupů k internetovým aplikacím. Ten umožňuje, pomocí měřícího kódu, zjišťovat jaké informace návštěvníky na internetových stránkách zajímají, odkud návštěvníci přicházejí a jaké prostředky využívají při prohlížení internetových stránek. Získané hodnoty se dlouhodobě ukládají do databáze a na jejich základě se vytvářejí statistiky rozdělené na časová období.

Rovněž jsem provedl analýzu již existujících internetových služeb či programů poskytujících podobné možnosti jako řešený projekt. Rozebírám rozdíly mezi jednotlivými řešeními. Dále poukazuji na vlastnosti řešeného projektu, které se liší od existujících systémů.

Při tvorbě diplomového projektu jsem zaměřil na možnosti rozšíření možností systému především s ohledem na oblast analýzy přístupových protokolů serveru Apache a zjištění nových měřených veličin, které nelze získat pomocí měřícího kódu. Dále se zaměřuji na oblast bezpečnosti sledovaných internetových aplikací s ohledem na detekci známých typů útoků na internetový server pomocí internetového prohlížeče a na prevenci proti dalším útokům. Jedná se zejména o typy útoků, které využívají dříve objevené bezpečnostní nedostatky rozšířených internetových aplikací či internetových serverů a mohou vést ke ztrátě kontroly nad serverem, na kterém internetová aplikace běží. Získaná data poté slouží v reálném čase k prevenci proti uskutečnění dalších pokusů o průnik do systému ze stejné IP adresy, jakou má útočník.

V systému pro monitorování návštěvnosti jsem rovněž vytvořil modul, který slouží k přihlašování registrovaných uživatelů do zabezpečené oblasti aplikace. Modul slouží taktéž pro vytváření bezpečnostních statistik, které ukazují pokusy o neoprávněný přístup do zabezpečené části aplikace.

# 2 Teoretický základ

V této části ukazuji, jakým způsobem lze technicky provádět monitorování přístupů na internetové stránky. Jsou zde rozebrány základní principy identifikace uživatelů, přenos a zpracování měřených dat a jejich uchování.

## 2.1.1 Identifikace unikátního uživatele

Základním problémem při zjišťování návštěvnosti internetových stránek je určit, který uživatel je na internetových stránkách poprvé, tedy je unikátní návštěvník. Za unikátního návštěvníka je považován takový návštěvník, kterého je možné odlišit od jiných pomocí unikátního identifikátoru.
Identifikace návštěvníka je velmi problematická z důvodu, že neexistuje žádný spolehlivý identifikátor uživatele, který by se předával mezi počítačem, resp. internetovým prohlížečem uživatele a internetovým serverem, na který směřuje požadavek. Proto je nutné hledat alternativní způsoby, jak rozlišovat různé uživatele, aby se měřené údaje a výsledné statistiky co nejvíce přiblížily skutečnosti. Z tohoto faktu je zřejmé, že míra kvality statistických dat, které vznikly měřením návštěvnosti internetových stránek je závislá na schopnosti rozlišovat návštěvníky internetových stránek.

Nejjednodušším prostředkem, jak lze rozpoznat návštěvníky, je rozlišení na základě IP adresy, kterou je možné zjistit z požadavku odeslaného na měřící server. Problémem tohoto způsobu je velmi nepřesné rozlišení návštěvníků, neboť jednou IP adresou může disponovat mnoho různých uživatelů. Jedná se nejčastěji o situaci, kdy se velká skupina uživatelů internetu nachází za zařízením zvaným „směrovač“ (angl. router). Ten bývá často používán v různých firmách či institucích, např. ve školách. Uživatelé se pak na internet připojují prostřednictvím tohoto zařízení, které se navenek jeví jako jeden počítač. Někdy mohou mít stejnou IP adresu i uživatelé běžného internetového připojení a to v případě, že jim poskytovatel internetového připojení nepřidělí veřejnou IP adresu.

Chyba měření, která by vznikla při rozlišování uživatelů na základě IP adresy počítače je přímo úměrná množství návštěvníků měřených internetových stránek. U internetových stránek s nízkou návštěvností nemusí být takto způsobená chyba patrná, neboť pravděpodobnost výskytu stejné IP adresy u více návštěvníků současně je velmi malá.

## 2.1.2 Cookies

Mnohem lepším prostředkem pro identifikaci unikátního uživatele jsou tzv. cookies. Cookies umožňují ukládat do prostředí internetového prohlížeče dočasné informace spravované původní internetovou stránkou či serverem. Po eventuelním vypnutí internetového prohlížeče, případně počítače, tyto informace přetrvají uložené na disku do další návštěvy stejných internetových stránek. Pro účely měření návštěvnosti lze tedy toho prostředku využít tak, že se na straně serveru vytvoří unikátní identifikátor, který se odešle v odpovědi na požadavek prohlížeče a tento identifikátor se pomocí cookie do prohlížeče uloží. Při další komunikaci internetového prohlížeče a serveru se již

přenáší tento identifikátor. Takové řešení je výhodné v tom, že neexistuje možnost záměny více uživatelů.

Tento způsob má však i své nevýhody. Jednou z nich je nutnost podpory cookies internetovým prohlížečem a navíc musí být povoleno jejich použití. Dalším problémem je fakt, že mohou být na straně klienta kdykoliv vymazány, např. při provádění údržby počítače. Při jejich odstranění je uživatel chybně interpretován jako unikátní. Některé prohlížeče navíc mohou odmítat přijetí pouze některých cookies. Jedná se nejčastěji o tzv. cookies třetích stran, což je případ, kdy je cookie vytvářena jiným internetových serverem, než na kterém se nacházejí internetové stránky. Z nich zpravidla vede odkaz na měřící server, který je odlišný od toho, na kterém se stránky nacházejí. V případě měření návštěvnosti jsou cookies, které založí server provádějící měření, označeny za cookies třetí strany a proto mohou být některými internetovými prohlížeči ignorovány, aby nedošlo k případnému zneužití osobních informací o uživateli. Pro účel jejich rozlišení či se používá tzv. P3P hlavička požadavku.

### 2.1.3 Hlavička P3P

P3P je zkratka pro projekt nazvaný *Platform for Privacy Preference* [3], který je spravován konsorciem W3C, zabývajícím se internetovými standardy. Tento projekt si klade za cíl umožnit uživateli kontrolu nad tím, jaká osobní data jsou předávána z internetového prohlížeče internetovým serverům a jakým způsobem je s osobními daty nakládáno. V hlavičce P3P jsou uvedeny informace o tom jaká data jsou do cookies ukládána, případně za jakým účelem. Hlavička P3P se odesílá jako součást odpovědi na požadavek klienta pomocí protokolu HTTP. Informace získané z této hlavičky využívá internetový prohlížeč při rozhodování, zdali má umožnit přijetí této cookies při nastavené úrovni soukromí a zabezpečení.

Příklad použití P3P hlavičky odesílané klientům při provádění měření návštěvnosti:

```
P3P: CP="ADM DEV PSD OUR IND COM NAV PRE DSP NON COR"
```

### 2.1.4 Zjištění dat pomocí vloženého měřícího kódu

Aby bylo možné sbírat informace potřebné pro vytváření statistik v reálném čase, je nutné na měřené stránky umístit tzv. měřící kód, který umožňuje zjistit požadované informace na straně klienta a zajišťuje jejich přenos na internetový server, který se stará o jejich zpracování. Tento kód se načítá ze serveru, který provádí měření a analýzu údajů.

Některé důležité informace lze zjistit již z hlavičky HTTP požadavku, který je odeslán z prohlížeče uživatele na cílový server při načítání měřícího kódu. Určité informace je však možné zjistit až po provedení měřícího kódu. Po načtení do internetového prohlížeče je měřící kód automaticky spuštěn. Jeho provedení umožňuje sledování některých technických informací, které lze zjistit pouze na počítači uživatele. Mezi ně patří:

- rozlišení obrazovky
- nastavená barevná hloubka
- velikost okna internetového prohlížeče
- přítomnost podpory cookies v prohlížeči

Informace o rozlišení, barevné hloubce obrazovky a velikosti okna prohlížeče lze využít při optimalizaci vzhledu měřené aplikace pro potřeby návštěvníků, případně při rozhodování o vzhledu a umístění reklamních ploch apod.

Informace, které lze zjistit přímo z odeslaného HTTP požadavku:

- IP adresa počítače (případně směrovače), přes který je uživatel připojen
- použitý internetový prohlížeč a jeho verze
- použitý operační systém a jeho verze
- výchozí jazyk, který je nastaven v prohlížeči
- adresa internetové stránky, ze které uživatel přichází. Z adresy lze zjistit o jaký server či doménu se jedná.

Pomocí těchto informací lze určit, které prohlížeče a operační systémy uživatelé nejčastěji používají. Toho lze využít při výběru technologií, které mohou být v rámci internetové aplikace nasazeny.
Spolu s HTTP požadavkem se odesílá i cookie, kterou spravuje měřící server. V ní jsou uloženy především informace o čase poslední návštěvy uživatele na internetových stránkách. Je zde uložen i unikátní identifikátor, který slouží k rozlišení uživatelů. Díky znalosti unikátního identifikátoru lze určit následující data, která se vztahují k danému časovému období:

- počet unikátních návštěvníků
- počet návštěv
- počet unikátních IP adres
- počet shlédnutých stránek
- případně další

Z naměřených dat se generují statistiky, které umožňují sledovat návštěvnost internetových stránek v různých časových obdobích.

### 2.1.5 Využití přístupových protokolů serveru k měření dat

Internetový server Apache ukládá informace o provozu serveru a o přístupech na internetové stránky, které jsou na něm umístěné do protokolů. Do nich zapisuje prakticky neustále po dobu své činnosti. Záznamy v protokolech jsou cenným zdrojem, který poskytuje detailní informace o aktivitách serveru, které není možné sledovat pomocí měřícího kódu umístěného na stránkách internetové aplikace.

Využít soubory protokolů k měření návštěvnosti je možné pouze v případě, že k nim má umožněn přístup nástroj pro jejich analýzu. Nejčastěji je analyzátor umístěn na stejném serveru, na kterém se nacházejí samotné protokoly o běhu serveru nebo je k protokolům možné zřídit přístup přes vzdálené zabezpečené spojení.

Mezi informace, které je možné zjistit pro účely statistik pouze z protokolů o běhu serveru patří:

- chybové kódy serveru, které vznikly jako reakce na chybný požadavek klienta, případně označují chybu běhu serveru
- velikost odesílaných dat klientovy
- typy odesílaných souborů

Server Apache podporuje tři druhy protokolů:

- protokol o chybách
- protokol o přístupech klientů
- rozšířený protokol o přístupech klientů

### *Protokol o chybách (error.log)*

Obsahuje diagnostické informace a záznamy o chybách, které mohly vzniknout při provozu serveru, případně vlivem přístupů klientů na neexistující adresu dokumentu nebo souboru, který se na serveru již nevyskytuje.

### *Protokol o přístupech klientů (access.log)*

Obsahuje záznamy všech přístupů klientů na internetové stránky případně k dalším souborům umístěným na serveru. Obsah protokolu lze využít k analýze a k následnému vygenerování statistik o přístupech návštěvníků. V základním nastavení ukládá server Apache veškeré záznamy o přístupech na jednotlivé hostované internetové domény do jediného souboru protokolu s názvem *access.log*. Toto chování však není pro účely zpracování vhodné z důvodu, že analyzovaný soubor obsahuje promíchané záznamy pro různé měřené internetové aplikace. Analýza takového souboru by se navíc, z důvodu jeho velikosti, projevila snížením výkonu systému. Pro účely měření dat z protokolů je vhodné nastavit pro každou sledovanou aplikaci samostatný soubor se záznamy o přístupech.

Rozdělení protokolů se provádí pomocí virtuálních serverů v nastavení konfiguračního souboru *httpd-vhosts.conf* serveru Apache. Každý vytvořený virtuální server odpovídá jedné sledované aplikaci a jeho vlastní soubor protokolů se nastavuje pomocí direktivy *CustomLog* jako v následujícím příkladě:

```
<VirtualHost 127.0.0.1>
...
    CustomLog /umisteni-aplikace/logs/access.log combined
...
</VirtualHost>
```

Jediný soubor se záznamy přístupů však stále není zcela efektivní řešení v případě, kdy se provádí jeho analýza automatizovaným procesem, neboť v důsledku růstu jeho velikosti by se neustále prodlužovala doba potřebná k jeho analýze. To by nevyhnutelně vedlo k nutnosti provádět opakované zálohování a snižovat jeho velikost, aby se docílilo zrychlení analýzy.

Z toho důvodu je vhodné využít externí modul pro Apache server s názvem *mod_log_rotate* [16], který umožňuje rozdělit soubor se záznamy na samostatné logické celky, které odpovídají nastavenému časovému intervalu. Příslušnost jednotlivých souborů časovému intervalu lze urči díky tomu, že v názvu souboru je uloženo časové razítko, které umožňuje zjistit počátek intervalu.

Záznamy protokolu mají přednastavený formát toho, jaké informace se o přístupech ukládají. Formát je vhodné pro účely měření upravit, aby jednotlivé záznamy obsahovaly více detailních informací vhodných pro zpracování konkrétních statistik. Pro jednoduchost je vhodné dodržet formát protokolových souborů pro všechny měřené aplikace, aby nedošlo k chybné interpretaci dat při jejich analýze. Pro účely analýzy je rovněž vhodné využít rozšířený formát záznamů, který obsahuje navíc internetovou adresu, ze které pochází požadavek klienta a identifikační řetězec internetového prohlížeče.

Příklad definice rozšířeného formátu protokolu:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
```

Příklad záznamu o přístupu klienta na základě definovaného formátu:

```
127.0.0.1 - - [14/Apr/2007:15:33:00 +0200] "GET /index.php HTTP/1.1" 200
3142 "http://www.nazevdomeny.cz/" "Mozilla/5.0 (Windows; U; Windows NT
5.1; cs; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3"
```

Ze záznamu lze postupně vyčíst následující informace, které odpovídají položkám formátu (v závorce):

- 127.0.0.1 (%h) – IP adresa klienta, který provedl požadavek na server.
- - (%l) – identita klienta. Pomlčka označuje nezjištěnou identitu.
- - (%u) – uživatelské jméno při HTTP autorizaci. Pomlčka označuje, že se autorizace neprovádí.
- [14/Apr/2007:15:33:00 +0200] (%t) – čas přijetí požadavku. Za mezerou následuje posun času vzhledem k GMT.

- "GET /index.php HTTP/1.1" (\"%r\") – obsahuje metodu požadavku klienta, požadovaný zdroj a číslo verze protokolu.
- 200 (%>s) – kód stavu serveru odeslaný klientovi po zpracování požadavku.
- 3142 (%b) – velikost zdroje odeslaného klientovi v bytech.
- "http://www.nazevdomeny.cz/" (\"%{Referer}i\") – adresa, ze které byl klient odkázán na požadovaný zdroj.
- "Mozilla/5.0 (Windows; U; Windows NT 5.1; cs; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3" (\"%{User-agent}i\") – identifikátor klientského prohlížeče.

### *Chybový dokument*

Internetový server Apache umožňuje nastavit dokument pro zobrazení uživatelské zprávy v případě výskytu některého chybového kódu v průběhu zpracování požadavku pomocí direktivy konfiguračního souboru *ErrorDocument*. Obvykle se chybový dokument používá k tvorbě uživatelsky přívětivých chybových stránek informujících uživatele o chybě, která nastala v době zpracování požadavku a o případných možnostech pokračování v návštěvě internetových stránek a nalezení požadovaných informací.

### *Konfigurační soubor .htaccess*

Kromě hlavního konfiguračního souboru *http.conf* internetového serveru Apache, jež je umístěn v adresáři samotného serveru, je možné využívat k řízení chování serveru také speciální konfigurační soubor s názvem *.htaccess*. Ten umožňuje změnit nastavení serveru pro každý veřejně přístupný adresář ve kterém se soubor nachází. Díky podpoře hierarchické struktury je nastavení platné i pro případné podadresáře všech úrovní. Na rozdíl od hlavního konfiguračního souboru, který se načítá pouze jednou při spuštění serveru Apache, se samostatné konfigurační soubory načítají při každém požadavku v rámci dané hierarchie adresářů kam požadavek směřuje. To umožňuje dynamickou konfiguraci serveru za jeho běhu.

Syntaxe souboru *.htaccess* je stejná jako u hlavního konfiguračního souboru serveru Apache. Podpora pro separátní konfigurační soubory na bázi adresářů musí být explicitně povolena v hlavním konfiguračním souboru serveru pomocí direktivy *AllowOverride* [8].

Tento způsob konfigurace serveru je na většině služeb poskytujících hostování internetových aplikací buďto výrazně omezen nebo zcela zakázán. Proto je možné tento konfigurační soubor používat pouze na serveru, na kterém je umožněn přístup k hlavnímu konfiguračnímu souboru.

Konfigurovat lze mnoho aspektů přístupu do adresáře – autorizace uživatelů, zamezení přístupu k některým typům souborů, povolení či zakázání přístupu ze seznamu IP adres.

Příklad struktury samostatného konfiguračního souboru *.htaccess* se seznamem IP adres majících omezený přístup na server:

```
Order  allow, deny
Allow from all
Deny from 192.168.1.2
Deny from 192.168.1.3
Deny from …
```

Jednotlivé direktivy konfiguračního souboru a jejich funkce:

Order – výchozí pořadí, ve kterém se vyhodnocují direktivy allow a deny.
Allow – umožňuje přístup vyjmenovaným klientům do oblasti na serveru
Deny – zamezuje přístup vyjmenovaným klientům do oblasti na serveru

### *Zjištění pokusů o narušení bezpečnosti serveru*

Z protokolu o přístupech, který vytváří internetový server Apache je možné zjistit informace o pokusech převzít kontrolu nad serverem pomocí známých bezpečnostních chyb v některé z předešlých verzí internetového serveru Apache, případně chyby aplikace, která může být na serveru veřejně přístupná, avšak její oprávněné použití podléhá autorizaci pomocí přihlašovacích údajů. Takové aplikace mohou sloužit například ke vzdálené administraci databázového serveru či obsahu internetových stránek.

Pokusy o útok na server je možné odhalit analýzou řetězce adresy, kterou útočník odesílá jako požadavek na internetový server. Ze záznamu o přístupu klienta lze zjistit z jaké IP adresy k pokusu o útok došlo, v jakém čase a o jaký typ pokusu se jedná, respektive jakou aplikaci se snaží útočník napadnout. Získané informace lze ihned využít například k prevenci proti uskutečnění dalších pokusů o útok ze stejné IP adresy pomocí spolupráce s internetovým serverem Apache. Tak je možné dočasně omezit přístup útočníka na server, aby neměl možnost provádět další útoky, které by mohly mít za následek, v případě úspěšného uskutečnění, ztrátu kontroly nad serverem.

## 2.1.6 Uchovávání naměřených dat

Naměřené údaje, které jsou na serveru zpracovány, je vhodné uchovávat v databázovém serveru. Ten může být umístěna na stejném serveru, který provádí měření, nebo může být umístěna na zvláštním fyzickém serveru. Samostatný server bývá nejčastěji dedikovaný (používá veškeré prostředky výhradně k zajištění běhu databáze). Řešení, u něhož se databáze nachází na samostatném serveru se používá v případech, kdy je požadována maximální bezpečnost dat, případně pokud se měření provádí na velkém množství internetových stránek a výkon samotného serveru by nebyl dostatečný pro zpracování měřených dat a současně jejich ukládání. Větší bezpečnosti je zpravidla dosaženo tím, že databázový server není připojen na internet, ale přímo k hlavnímu serveru pomocí lokální sítě. Pravděpodobnost zneužití citlivých informací uložených v databázi se tak minimalizuje. Výkon a datová kapacita databázového serveru musí být úměrná počtu měřených internetových stránek.

# 3 Analýza požadavků

V této části se zaměřuji na analýzu typů metod měření návštěvnosti internetové aplikace a porovnání existujících řešení. Dále popisuji požadavky na navrhovaný systém pro měření a tvorbu statistik. Ukazuji podobné rysy existujících aplikací s navrhovaným systémem, ale také nové vlastnosti, které se v existujících řešeních nevyskytují.

## 3.1 Analýza způsobů měření a existujících řešení

Existující internetové služby a programy lze rozdělit na dva typy, podle způsobu, jakým provádějí měření dat o přístupech uživatelů na internetové stránky.

Prvním a zároveň nejčastějším typem jsou internetové služby, které poskytují možnost monitorovat návštěvnost vložením měřícího kódu do analyzovaných stránek. Po jeho provedení na straně klienta se odešlou a zaznamenají informace o jeho přístupu na specializovaném serveru, který provádí sběr měřených dat. Příkladem takové služby jsou např. Google analytics [9] nebo český portál Navrcholu.cz [10].

Druhým typem jsou obvykle samostatné programy či skripty, které fungují jako analyzátory protokolů serveru o jednotlivých přístupech klientů a na základě toho zjišťují měřená data, která slouží jako zdroj pro generování statistik návštěvnosti. Tyto programy bývají nejčastěji provozovány na stejném serveru, na kterém se nacházejí analyzované internetové stránky a kde se ukládají serverové protokoly o přístupech. Mezi nejznámější programy tohoto typu patří AWStats [11].

Oba způsoby měření dat poskytují podobné typy informací o přístupech, avšak ve zjištění některých informací se zásadně odlišují. Analýza přístupových protokolů totiž umožňuje zjistit přístupy k binárním souborům na serveru, množství přenesených dat, chybové kódy serveru, které nastaly při zpracování požadavku klienta, případně další informace, které poskytuje server za běhu a není možné je zjistit pomocí měřícího kódu umístěném v měřené aplikaci.

Nasazení některého z těchto typů měření je podmíněno technickými možnostmi vlastníka internetové aplikace. Analýza přístupů pomocí měřícího kódu vyžaduje modifikaci internetové prezentace a vložení kódu na každou stránku, kde se má provádět měření. Naopak provádění analýzy přístupových protokolů vyžaduje přístup k protokolům serveru, což v mnoha případech z bezpečnostních důvodů není umožněno. Tato situace nejčastěji nastává například při hostování internetových stránek na cizím serveru. Na vlastním serveru je možné pomocí nastavení zajistit analyzátoru přístup k protokolům.

V navrhovaném systému hodlám spojit oba způsoby měření, aby bylo možné sledovat komplexní statistiky běhu sledované aplikace.

## 3.2 Bezpečnostní problematika přístupů

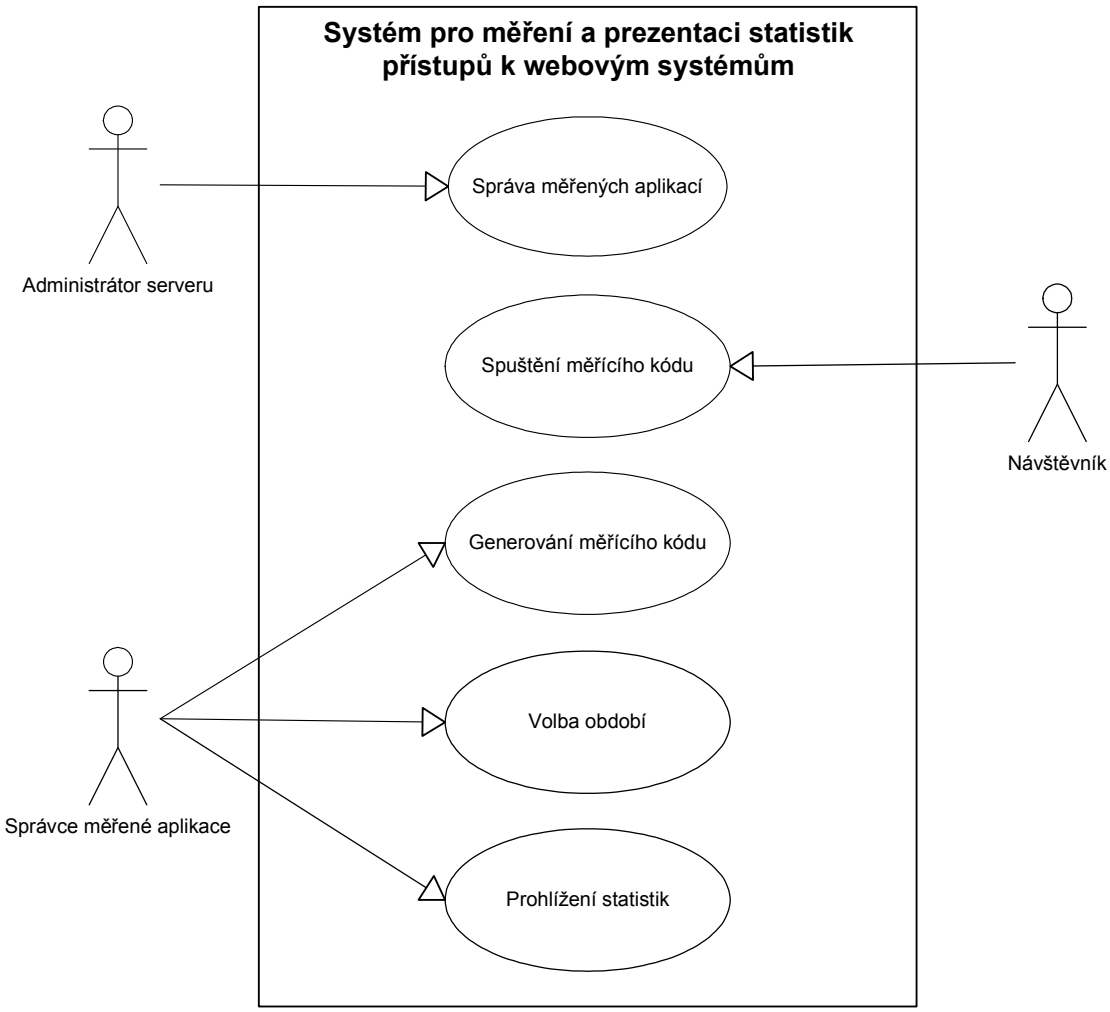
U sledovaných systému se nevyskytuje část zaměřená na analýzu bezpečnosti sledované aplikace, která by umožnila buď pasivně sledovat pokusy o narušení bezpečnosti nebo by aktivně zasáhla při detekci útoku.

Protože se jedná o oblast, která se v této době dostává do popředí zájmu, rozhodl jsem se tuto problematiku v navrhovaném systému zohlednit. Jedná se zejména o detekci známých typů útoků na internetové servery a následné zamezení v provádění dalších útoků.

Dále se zaměřuji na možnost sledování neúspěšných pokusů o přístup do zabezpečené části aplikace.

# 4 Návrh aplikace

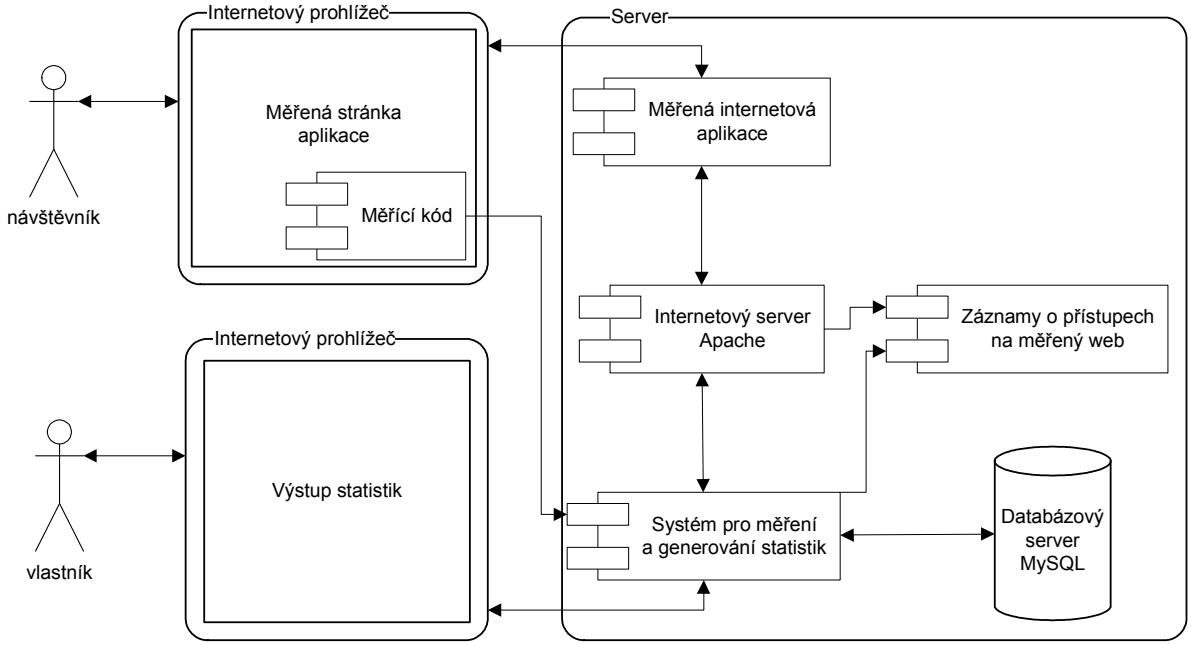
Aby bylo možné vytvářet statistiky, je nejprve nutné umožnit sběr informací o uživatelích internetových stránek. Sběr požadovaných informací se provádí umístěním měřícího kódu na všechny sledované stránky internetové aplikace. Při návštěvě uživatele na některé ze sledovaných stránek se spustí měřící kód v internetovém prohlížeči a ten odešle zjištěné informace o čase návštěvy, technickém vybavení počítače uživatele případně některé další údaje, které mohou sloužit ke generování statistik. Informace se odesílají na internetový server, který provádí sběr a archivaci dat pro pozdější využití.




*Obrázek 1: Případ použití systému pro tvorbu statistik*

Kromě měřícího kódu lze také provádět analýzu přístupových protokolů na straně internetového serveru. Tato metoda poskytuje především možnost analyzovat provoz serveru jako takového vzhledem k měřené aplikaci. Umožňuje zjistit např. množství přenesených dat či vzniklé chybové kódy při běhu aplikace. Jedná se o informace, které není možné zjistit pomocí měřícího kódu.

Naměřená data jsou přístupná pro provozovatele analyzovaných internetových stránek v informačním systému umístěném na internetu. Data jsou prezentována ve formě statistik, které se generují v reálném čase a jsou dostupné pro různá časová období v nichž se měření provádělo.




*Obrázek 2: Diagram rozmístění*

V systému budou rovněž zahrnuty dva typy bezpečnostních statistik. První typ bezpečnostních statistik má zachycovat známé útoky pomocí internetového prohlížeče na internetový server, které byly učiněné v průběhu dne. Budou zaznamenány IP, ze kterých útoky pocházejí. Typy známých útoků jsou uloženy v databázi a při každém přístupu uživatele je kontrolován požadavek na internetový server, zdali neodpovídá některému ze známých typů útoků. V případě, že je požadavek vyhodnocen jako pokus o útok na internetový server, zaznamenají se informace o požadavku a zjištěnou IP adresu počítače útočníka lze využít k další analýze či okamžitému zabezpečení serveru tak, že server nebude po jistou dobu přijímat z dané adresy žádné další požadavky.

Druhým typem jsou bezpečnostní statistiky, které jsou tvořeny z dat naměřených pomocí modulu pro autorizaci registrovaných uživatelů. Modul zaznamenává neúspěšné pokusy o přihlášení do zabezpečené oblasti použitím nesprávných přihlašovacích údajů. V naměřených datech je pak možné kontrolovat, jaké chybné pokusy o přístup byly v průběhu dne učiněny.

# 5 Implementace

V první části této kapitoly popisuji nástroje, pomocí kterých jsem vytvořil celý systém pro sběr a prezentaci statistik.

Dále je rozebrána implementace části systému sloužící ke zjištění měřených dat na straně klienta a jejich přenosu pomocí měřícího kódu vloženého do sledovaných stránek internetové aplikace. Také je tu rozebrán způsob měření dat, který spočívá ve zpracování záznamů o přístupech klientů na straně internetového serveru Apache.

Následuje popis implementace systému pro generování statistik z naměřených dat uložených v databázi. Jsou zde prezentovány všechny typy statistik, které jsem v rámci projektu implementoval. Součástí jsou ukázky výstupů statistik zhotoveného systému.

Ve čtvrté části se zaměřuji na bezpečnostní analýzu měřených dat. Ukazuji jakým způsobem systém detekuje útočníky provádějící útoky na sledovanou aplikaci a jak útočníkovi zamezit v provádění dalších útoků.

V poslední částí jsou popsány externí knihovny, které jsem využil při tvorbě systému a nově vytvořené knihovny, které vznikly v rámci jeho tvorby. Dále následuje seznam hlavních souborů systému spolu s jejich popisem.

## 5.1 Programovací nástroje

Systém pro měření dat, jejich zpracování a následné vyhodnocení statistik je naprogramován v jazyce PHP 5 [5]. Jedná se o vyšší programovací jazyk určený pro tvorbu dynamických internetových aplikací a stránek. Jazyk je interpretovaný na straně serveru a jeho výstupem je obvykle HTML kód, který se odesílá do internetového prohlížeče. Jazyk PHP jsem zvolil z důvodu, že se jedná o platformě nezávislý prostředek pro generování dynamických internetových stránek. Zároveň je nejrozšířenější a nejvíce používaný prostředek tohoto typu, neboť je volně dostupný stejně tak jako jeho zdrojový kód. Díky tomu je možné jej nasadit na téměř libovolném internetovém serveru.

Měřící kód ke zjišťování dat na straně klienta je naprogramován v jazyce JavaScript [6]. Jedná se o skriptovací jazyk, který se interpretuje v internetového prohlížeče uživatele.

Pro uložení naměřených dat se využívá relační databázový server MySQL 5 [7] Data se do databáze ukládají ve formátu, který je vhodný pro přímé a rychlé generování statistik. Díky tomu je možné generovat a zobrazovat statistiky v reálném čase, tedy i v době, kdy se provádí měření pro dané časové období. Relační databázový server MySQL 5 jsem vybral z důvodu, že poskytuje vysokou rychlost a spolehlivost a je na rozdíl od mnoha jiných databázových serverů volně dostupný pro použití. Jak název napovídá, jedná se o databázi využívající syntaxi jazyka SQL. MySQL 5 je v dnešní době již velice vyspělý databázový nástroj, který umožňuje používat i pokročilé techniky jako databázové procedury či vnořené SQL dotazy (narozdíl od jeho předešlé verze MySQL 4). MySQL tedy nabízí prostředky, které plně postačují i pro tvorbu vysoce výkonných databázových aplikací.

Jako internetový server je použit HTTP server Apache 2.2 [8]. Opět se jedná o platformě nezávislý prostředek sloužící ke zpřístupnění obsahu na internetu. Je vyvíjen jako open-source projekt se zaměřením na bezpečnost, efektivnost a rozšiřitelnost. Apache je dosud nejvíce používaný internetový server na světě již od roku 1996.

## 5.2 Techniky měření

V rámci projektu kombinuji metodu měření návštěvnosti pomocí vloženého kódu do analyzovaných internetových stránek s metodou analýzy protokolů o běhu internetového serveru Apache [8]. Výhoda tohoto řešení spočívá v možnosti zjišťovat efektivně velké množství informací o jednotlivých návštěvách a reakcích serveru na požadavky klientů. Díky tomu je možné zjišťovat informace, které by za použití pouze jediného z obou řešení nebylo možné.

V případě vložení kódu do internetové stránky je možné zjistit informace o klientském počítači, tedy například rozlišení a barevnou hloubku zobrazovacího zařízení, podporu technologií v internetovém prohlížeči klienta případně jiných informací, které vyžadují použití skriptovacího jazyka na straně klienta. Naopak použitím analýzy protokolů serveru je umožněno zjistit informace o výkonu serveru či další praktická data spojená s provozem serveru (viz 2.1.5). V projektu tedy kombinuji dva možné přístupy, které se obvykle v jiných řešeních vyskytují samostatně a tudíž neposkytují tak komplexní pohled na provoz internetové aplikace.

Dále se zaměřuji na využití informací o přístupech na internetový server ke zvýšení bezpečnosti sledované aplikace tím, že jsem vytvořil databázi známých typů útoků pomocí internetového prohlížeče na rozšířené internetové aplikace a na starší verze internetových serverů a díky tomu je možné odhalit případné útočníky a včas jim zabránit v provádění dalších útoků.

Ke zvýšení bezpečnosti sledované aplikace slouží také modul pro přihlašování autorizovaných uživatelů. Pomocí něj je možné sledovat pokusy o neoprávněné vniknutí do zabezpečené oblasti aplikace vícenásobným zadáváním chybných přihlašovacích informací.

### 5.2.1 Zjišťování dat pomocí měřícího kódu

Měřící kód napsaný v JavaScriptu. Slouží výhradně ke zjištění technických informací o internetovém prohlížeči uživatele a jeho podpoře technologie cookies. Skript se načítá současně s obsahem stránky a prohlížeč se sám postará o provedení jeho obsahu. Po provedení skriptu jsou zjištěné informace odeslány na server.

JavaScript není jediná technologie pro spouštění skriptů v internetovém prohlížeči, tedy na straně klienta. Kromě JavaScriptu je možné použít například VBScript, což je skriptovací jazyk odvozený od programovacího jazyka Visual Basic pro platformu Microsoft® Windows. Tato technologie však není, narozdíl od JavaScriptu, platformě nezávislá, proto se její použití pro obecné účely nehodí. Lze ji však použít na zmíněném operačním systému ke zjištění dodatečných informací o klientovi. Internetový prohlížeč však daný skriptovací jazyk musí podporovat a jeho použití musí být povoleno.

### *Ukázka kódu*

Kód jsem úmyslně zapsal stylem, který není obecně příliš vhodný pro programování, ale v tomto případě má použitý styl svůj důvod. Kód se nahrává ke každému uživateli spolu s obsahem internetové stránky, proto musí být jeho velikost co nejmenší, aby nezpomaloval načítání stránky. Druhý důvod je ekonomický. S velikostí kódu roste množství přenesených dat směrem k uživateli, což zvyšuje náklady na provoz serveru. Tato situace se projevuje zvláště v případě, kdy se provádí sledování několika desítek či stovek internetových aplikací současně nebo pokud jsou sledované internetové stránky navštěvovány velkým množstvím uživatel.

```
var d=document;
var s=screen;
var t=new Date();
t.setTime(t.getTime()-2592000);
var n="cookie_support=";
d.cookie=n+"1";
var c=(d.cookie.indexOf(n) != -1) ? 1:0;
d.cookie=n+"1; expires="+t.toGMTString();
cd=(s && s.colorDepth) ? s.colorDepth:s.pixelDepth;
ww=(d.all) ? document.body.clientWidth:innerWidth;
d.write("<img src=\"http://dwstat.no-ip.com/hit.php?site=<?php echo
$_GET['site']; ?>&amp;cs=" + c + "&amp;ref=" + escape(document.referrer)
+ "&amp;res=" + s.width + "x" + s.height + "&amp;cd=" + cd + "&amp;ww="
+ ww + "&amp;rnd=" + Math.random() + "\" width=\"1\" height=\"1\"
alt=\"\" />");
```

### *Načtení měřícího kódu*

Měřící kód se do internetových stránek vkládá pomocí HTML kódu, který zajišťuje načtení daného skriptu z měřícího serveru.

```
<script language="JavaScript" type="text/javascript"
src="http://dwstat.no-ip.com/code.js?site=X">
</script>
<noscript>
  <div><img src="http://dwstat.no-ip.com/hit.php?site=X" width="1"
height="1" alt="" />
  </div>
</noscript>
```

Hodnota „X“ parametru *site* se nahradí za identifikátor měřeného serveru.

Pokud internetový prohlížeč nepodporuje JavaScript, skript se ze serveru nenahraje. Místo toho se odesílá požadavek na měřící server, aby se přímo provedlo započítání uživatele. Odeslání požadavku je provedeno pomocí obrázku vloženého do HTML stránky, kde je jeho adresa nastavena

na soubor s koncovkou *.php*, který započítání návštěvy uživatele zajišťuje. Obrázek má velikost 1x1 obrazový pixel a je umístěn do blokového elementu. Díky tomu není nenarušen design internetové stránky.

### *Zpracování dat zjištěných pomocí měřícího kódu*

O zpracování měřených dat se stará soubor *hit.php*, který se pro internetový prohlížeč jeví jako obrázek o rozměrech 1x1 obrazový pixel uložený ve formátu *GIF*. Tento formát je klasický formát obrázku, který dokáže zpracovat každý internetový prohlížeč podporující zobrazení grafických dat.

V hlavičce odesílané spolu s odpovědí ve formě obrázku jsou uloženy pokyny pro zařízení typu Proxy a Cache. Tyto pokyny říkají zařízením, aby neukládaly daný obsah do vyrovnávací paměti, ale místo toho vždy znovu načetli obsah ze serveru. Pokud by se provedlo načtení dat z vyrovnávací paměti místo z měřícího serveru, nebyl by daný přístup započítán. To by vedlo k chybě, která by znemožňovala zpracovávat důvěryhodné statistiky.

### *Příklad hlaviček protokolu http*

```
header('P3P: CP="ADM DEV PSD OUR IND COM NAV PRE DSP NON COR"');
header('Pragma: no-cache');
header('Cache-Control: no-cache');
header('Content-type: image/gif');
header('Expires: ' . gmdate('D, d M Y H:i:s') . ' GMT');
```

První hlavička obsahuje informaci pro prohlížeč, jaký druh dat se zpracovává a k jakému účelu získaná data slouží. Další dvě hlavičky zakazují použití vyrovnávací paměti. Čtvrtá nastavuje typ obsahu jako obrázek ve formátu GIF a poslední udává vypršení platnosti obsahu, které nastává v čase načtení hlavičky. Těmito technikami je zajištěno opětovné vykonání souboru *hit.php* při jeho volání a zamezí se jeho uložení do vyrovnávací paměti internetového prohlížeče.

## 5.2.2 Měření pomocí analýzy přístupových souborů

Proces analýzy přístupových souborů je rozdělen na dvě části – analýzu přístupových souborů a generování výsledných statistik. O každou část se starají samostatné třídy.

### *Třída ApacheLogParser*

Slouží k analýze přístupových souborů serveru. Provádí načítání záznamů o přístupech na server (viz 2.1.5) a jejich uložení do vhodné struktury, která je poté použita pro tvorbu statistik.

Načtení hodnot do struktury se děje pomocí regulárního výrazu, který odpovídá nastavenému formátu záznamů o přístupech v konfiguračním souboru Apache:

```
/^(\S+) (\S+) (\S+) \[([^\/]+)\/([^\/]+)\/([^:]+):(\d+):(\d+):(\d+)
([^\]]+)\] \"(\S+) (.*?) (\S+)\" (\S+) (\S+) \"(.*?)\" \"(.*?)\"/
```

### *Třída ApacheLogStats*

Třída pro tvorbu statistik z poskytnutých dat. Provádí také uložení zjištěných hodnot do databáze, čímž je zajištěna rychlejší analýza a tvorba statistik, neboť přímá analýza souborů je pro delší časové období (např. měsíc) velmi náročná na prostředky serveru. Statistiky, které jsou generovány touto třídou viz 2.1.5.

## 5.2.3 Uložení naměřených hodnot do databáze

Naměřená data se ukládají na databázový server MySQL 5.0 pro pozdější využití při generování statistik přístupů. Hodnoty se ukládají do dvou databází.

První databáze je společná pro všechny měřené servery, druhá je samostatná pro každou měřenou internetovou aplikaci.

### *Společná databáze pro všechny měřené servery*

Obsahuje tabulky společných záznamů, které jsou využívány u všech měřených serverů, které se na ně odkazují pomocí cizích klíčů. Dále obsahuje tabulku záznamů o útocích na samotný internetový server Apache.

Databáze obsahuje následující tabulky s popsanou funkcí:

- *admins* – seznam uživatelů s administrátorským přístupem do systému pro měření statistik
- *browsers_collector* – zjištěné internetové prohlížeče, rodiny a verze podle identifikačního řetězce
- *os_collector* – typy operačních systémů, rodin a verzí
- *screens_collector* – používané kombinace rozlišení a barevné hloubky obrazovky
- *server_url_attacks* – útoky na internetový server Apache bez rozlišení hostovaných stránek. Obsahuje i seznam útoků na všechny hostované servery s výjimkou těch serverů, u kterých se provádí samostatné měření.
- *url_attacks_collector* – šablony jednotlivých typů známých útoků na internetový server a na světově rozšířené internetové aplikace.
- *websites* – seznam serverů u nichž se provádí sběr informací a tvorba statistik

### *Samostatná databáze pro každý měřený server*

Zde se ukládají do tabulek naměřená data pro generování statistik. Část tabulek slouží pro uložení hodnot pocházejících z měřícího kódu a část pro data pocházející z analyzovaných přístupových souborů serveru Apache. Dále databáze obsahuje tabulku útoků zaznamenaných u konkrétní internetové aplikace.

Databáze obsahuje následující tabulky s popsanou funkcí:

- *apache_files* – obsahuje informace o velikostech přenesených souborů
- *apache_files_collector* – kolekce všech souborů přenášených v rámci internetové aplikace

- *apache_status_codes* – adresy, které způsobily vygenerování chybového kódu serverem
- *apache_traffic* – zjištěné množství přenesených dat
- *auth_attempts* – neúspěšné pokusy o přihlášení do oblasti zabezpečené autorizací uživatel
- *auth_users* – uživatelské účty pro autorizaci přístupu do zabezpečené oblasti
- *browsers* – zjištěné internetové prohlížeče uživatelů, jejich rodiny a verze
- *languages* – zjištěné nastavení primárního jazyka v internetovém prohlížeči
- *os* – operační systémy využívané návštěvníky
- *pages* – přístupy na jednotlivé stránky měřené internetové aplikace
- *pages_collector* – seznam stránek, na nichž je umístěn měřící kód. Tabulka slouží jako kolekce, kterou využívají další tabulky a odkazují se na záznamy pomocí cizího klíče
- *pages_duration* – délka trvání návštěv na jednotlivých stranách
- *pages_visited* – množství zobrazených stránek za jednotku času
- *referrers* – seznam odkazujících stránek, ze kterých přicházejí návštěvníci z jiných serverů
- *screens* – nastavení rozlišení a barevné hloubky obrazovek návštěvníků
- *unique_ip* – množství unikátních IP adres, z nichž byl učiněn přístup na stránky za jednotku času
- *unique_ip_collector* – seznam IP adres návštěvníků a jim náležících doménových jmen. Tabulka slouží jako cache při zjišťování názvu domény pro konkrétní IP adresu, neboť kontaktování DNS serveru je časově náročné.
- *url_attacks* – zaznamenané útoky na měřené stránky, přesný čas, typ útoku a IP adresa útočníka
- *visitors* – počet návštěvníků za jednotku času
- *visits* - počet návštěv za jednotku času
- *window_sizes* – nastavená velikost okna internetového prohlížeče návštěvníků

## 5.3 Systém pro vyhodnocení a zobrazení statistik

Generované statistiky přístupů je možné prohlížet v informačním systému, který je rovněž přístupný na internetu. Do systému má přístup správce internetové aplikace, jejíž měření se provádí, volitelně pak další uživatelé. Přístup do systému je vázán na autorizaci uživatele pomocí vložení přístupových informací do přihlašovacího dialogu.

Statistiky se vytvářejí pro různá časová období a v systému lze zobrazení jednotlivých období vybrat pomocí kalendáře. Ten umožňuje uživateli zvolit generování statistik pro jednotlivé hodiny dne, dny v týdnu a pro všechny dny v měsíci. Lze se rovněž přesouvat po jednotlivých měsících a zobrazit tak hodnoty pro starší měřená časová období.

Systém je schopen s uživatelem komunikovat a prezentovat statistiky ve dvou jazycích - v češtině a angličtině. Pro tvorbu jazykových mutací jsem využil modul jazyka PHP s názvem *Gettext* [18]. Modul využívá pro každou jazykovou verzi specifický soubor *messages.po* s překlady vět do druhého jazyka. Z důvodu zvýšení výkonu se provádí převod textového souboru do kompilované podoby souboru s názvem *messages.mo*, který zůstává načten v paměti po celou dobou provozu internetového serveru. V případě požadavku na překlad do jiného jazyka pak stačí pouze vytvořit

nový soubor s překlady pro daný jazyk a drobná změna v programu, která umožní jazyk v systému vybrat pro zobrazení.

K tvorbě výstupu informačního systému jsem využil nástroje pro tvorbu HTML šablon s názvem Smarty [12]. Pomocí něj je možné vytvořit obecnou šablonu HTML stránky (případně jiného strukturovaného či textového dokumentu), která obsahuje především grafický design a strukturu informací, ale neobsahuje žádná data. Díky tomu je možné oddělit prezentační vrstvu od informační vrstvy. Data jsou do šablony vložena pomocí prostředků, které tento nástroj nabízí. Mezi ně patří použití proměnných a příkazů pro řízení toku dat. Šablonový systém zároveň obsahuje možnost dočasného uložení kompilovaných šablon do vyrovnávací paměti, čímž se zajistí menší dopad na výkon internetového serveru.

### *Příklad šablony textového souboru v systému Smarty*

Tato šablona se používá pro odeslání přihlašovacích informací registrovanému uživateli:

```
Account information
------------------------------------------
User name: {$user.name} {$user.surname}
E-mail:    {$user.email}
Login:     {$user.login}
Password:  {$user.password}
```

Ve složených závorkách jsou zapsány proměnné, které jsou v čase vyhodnocení šablony nástrojem Smarty nahrazeny obsahem.

### *Příklad hlavní šablony HTML výstupu pro prezentaci statistik*

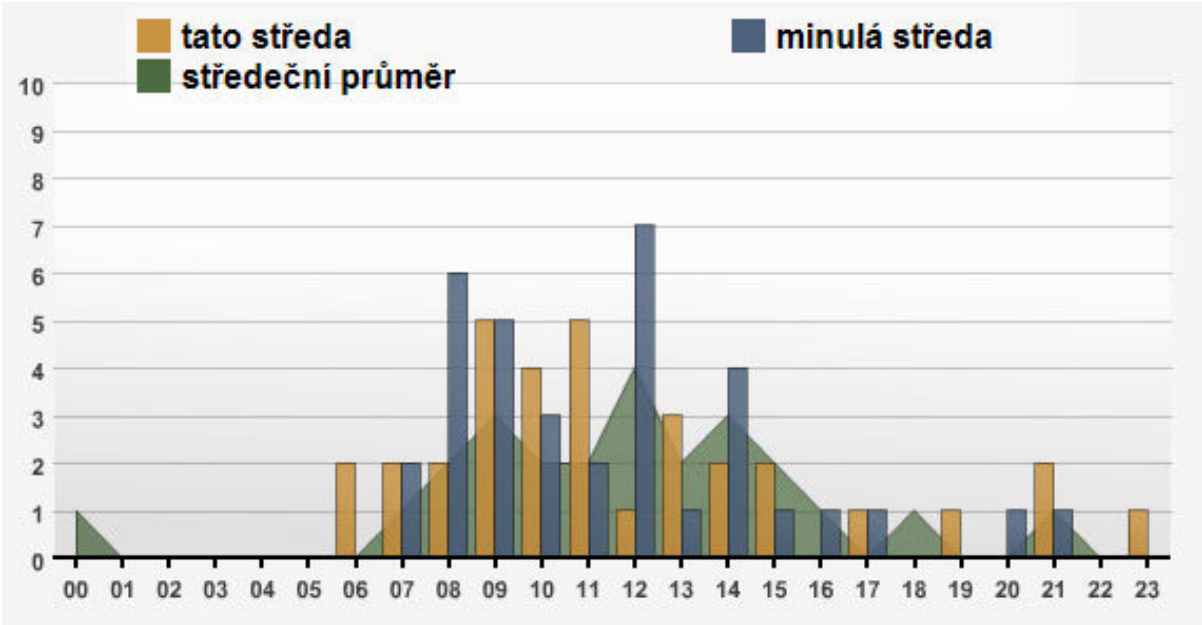
V šabloně se nachází možnost generování více statistik pod sebe. Každá z nich může obsahovat grafické zobrazení a zároveň tabulku naměřených hodnot, případně navigační menu, které umožňuje vybrat typ zobrazených informací, viz B.

### *Grafické zobrazení statistik*

Pro tvorbu grafů ke zobrazeným statistikám jsem využil volně šiřitelé aplikace s názvem PHP/SWF Charts [13], pomocí níž lze prezentovat grafická data pomocí dvourozměrných či trojrozměrných grafů. Grafy jsou do HTML stránky umístěny ve formě objektu ve formátu Flash. Aby bylo možné objekt v internetovém prohlížeči zobrazit, musí v něm být nainstalována podpora pro tento formát. Podpora je dostupná zdarma a vzhledem k tomu, že je formát velmi často používán nejen k prezentaci grafických dat, ale rovněž k tvorbě celých internetových stránek, bývá velmi často v internetovém prohlížeči již nainstalován. V případě chybějící podpory tohoto formátu jsou data ve statistikách dostupná ve formě tabulky.

Aplikace podporuje i možnost tvorby složených grafů, kdy jsou v jednom grafu zobrazeny dva nebo více průběhů. Toho jsem využil pro zobrazení dlouhodobých průměrů hodnot pro dílčí časová období. Aktuální hodnoty a hodnoty z předchozího období jsou zobrazeny ve sloupcových

grafech a dlouhodobý průměr z naměřených hodnot je zobrazen na pozadí formou plošného grafu odlišné barvy.




*Obrázek 3: Složený sloupcový a plošný graf*

## 5.4 Implementované statistiky

Díky tomu, že jsou veškeré naměřené hodnoty dlouhodobě uloženy v databázi, je možné je využít ke zobrazení statistik pro různá časová období. Kromě denních statistik, které zobrazují přístupy v jednotlivých hodinách v průběhu dne je možné generovat statistiky denní, týdenní a měsíční, zobrazující statistiky pro dané časové období. Součástí zobrazení statistiky je i graf, který udává dlouhodobé průměrné hodnoty v daných časových úsecích. Díky tom je možné porovnat, jakým způsobem se v jednotlivých časových obdobích mění návštěvnost internetové aplikace. Součástí zobrazení časových statistik je také plošný graf, který prezentuje zjištěné dlouhodobé průměrné hodnoty v daném časovém intervalu.

Statistiky jsou rozděleny podle typu informací, které poskytují. Statistiky návštěvnosti poskytují informace o množství návštěvníků, množství zobrazených stránek apod. Druhým typem jsou statistiky popisující technické aspekty, jako typ a verze internetového prohlížeče, rozlišení obrazovky apod. Hlavní přínos pro majitele stránek mají především statistiky o návštěvnosti, neboť z nich je možné zjistit, jak jsou internetové stránky úspěšné v různých časových úsecích, které stránky jsou nejvíce vyhledávané, případně jak často se uživatelé na stránky vracejí. Tyto informace mohou posloužit ke zlepšení kvality obsahu a tím i ke zvýšení návštěvnosti. Naměřená technická data mohou být využita pro návrh vzhledu stránek, aby byly lépe zobrazitelné na počítačích či jiných zařízeních, ze kterých uživatelé přistupují ke stránkám při použití daného rozlišení obrazovky a počtu barev ve zjištěných internetových prohlížečích.

### 5.4.1 Bezpečnostní statistika známých pokusů o útok na server

Statistika slouží ke zobrazení útoků na známé bezpečnostní chyby starších verzí internetových serverů Apache a IIS, případně pokusů o útok na některé rozšířené internetové aplikace veřejně přístupné na serveru. Ve statistice jsou zobrazeny nejčastější typy známých útoků, jež byly provedeny za dané období a jejich četnosti. Součástí je i zobrazení IP adres útočníků. Útoky na samotný server jsou mnohem častější než útoky na konkrétní měřený web z důvodu, že útočníci obvykle provádějí útok na konkrétní IP adresu serveru nebo na větší rozsah IP adres.

### 5.4.2 Bezpečnostní statistika známých pokusů o útok na web

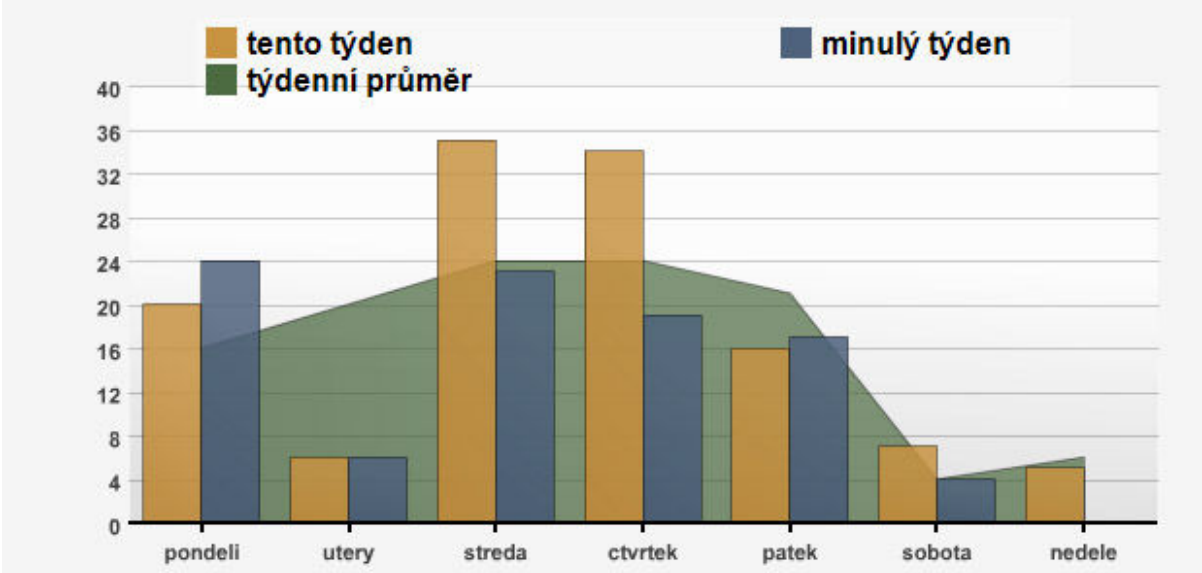
Podobně jako předchozí statistika zobrazuje pokusy o útok na chyby internetových serveru či na rozšířené serverové aplikace, ale s rozlišením na konkrétní měřený web. Statistika tedy neobsahuje útoky, které byly vedeny na samotný internetový server.

### 5.4.3 Bezpečnostní statistika chybných přihlášení

Zobrazuje chybně zadaná přihlašovací jména a hesla, která byla použita při pokusu o přístup do zabezpečené oblasti internetové aplikace. K jednotlivým pokusům se zaznamenává čas provedení pokusu a IP adresa počítače, z něhož byl pokus o přístup proveden. Tato statistika se zobrazuje pouze v případě, že je využit modul pro přihlašování registrovaných uživatelů, viz 5.5.2.

### 5.4.4 Unikátní návštěvníci

Následující ukázka statistiky zobrazuje počet unikátních návštěvníků na internetových stránkách v průběhu jednoho týdne. Křivka na pozadí prezentuje dlouhodobou statistiku průměrných hodnot pro dané časové období, v tomto případě pro každý den týdne.




| Unikátní návštěvníci | | 19. týden, 2007 |
|---|---|---|
| den | počet | % |
| pondělí | 20 | 16,26 % |
| úterý | 6 | 4,88 % |
| středa | 35 | 28,46 % |
| čtvrtek | 34 | 27,64 % |
| pátek | 16 | 13,01 % |
| sobota | 7 | 5,69 % |
| neděle | 5 | 4,07 % |
| celkem | 123 | 100% |

*Obrázek 4: Unikátní návštěvníci*

### 5.4.5 Návštěvy

Zobrazuje návštěvy uživatelů. Za návštěvu se považuje množina požadavků na všechny měřené internetové stránky odeslaných jedním návštěvníkem prostřednictvím jeho internetového prohlížeče. Návštěva se považuje za ukončenou, jakmile návštěvník opustí měřené internetové stránky nebo pokud během doby 30-ti minut neodešle žádný další požadavek na zobrazení některé z měřených stránek.

### 5.4.6 Unikátní IP adresy

Celkové množství unikátních IP adres, ze kterých byl přijat alespoň jeden požadavek v daném časovém období. Obecně statistika unikátních IP adres nepodává věrohodné informaci o návštěvnosti, neboť za jednou IP adresou se může skrývat větší množství fyzických návštěvníků, kteří obvykle pocházejí z jedné organizace. Vyskytuje se zde tedy problém možného překladu adres vnitřní sítě organizace na jednu či více (obecně menší počet) veřejných IP adres a naopak. Jedná se tedy spíše o informativní statistiku, kterou je vhodné použít např. pro porovnání se statistikou návštěvníků, která obvykle nabízí přesnější hodnoty.

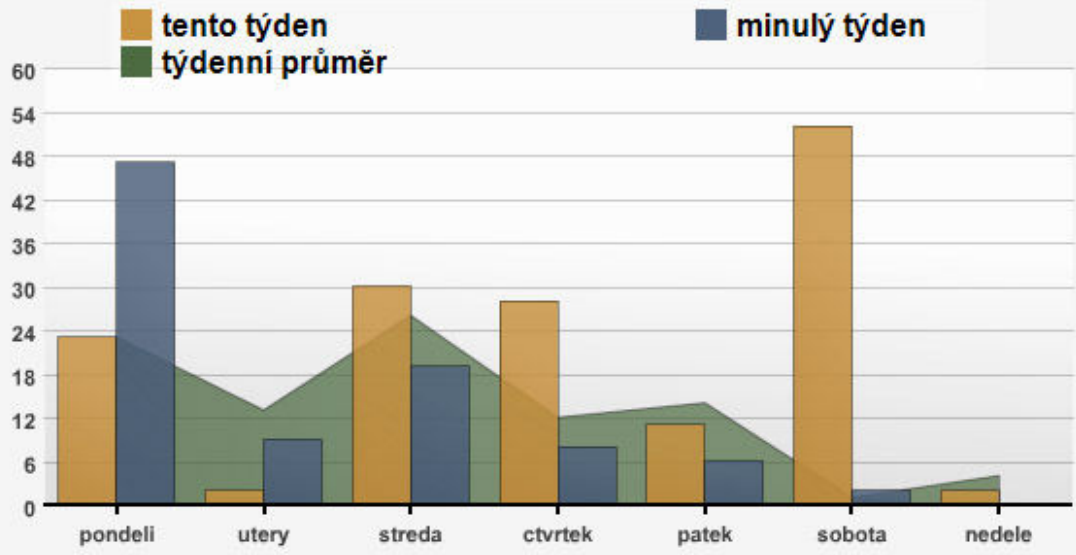
### 5.4.7 Počet zobrazených stránek

Zobrazuje množství zobrazených internetových stránek v daném časovém období. Ve statistice se opět vyskytují pouze ty stránky, na nichž je správně umístěn měřící kód. Ze zobrazených dat je možné si udělat představu o tom, v jakém čase jsou nejvíce požadovány nabízené informace na stránkách. Díky tomu je možné se zaměřit na optimalizaci výkonu serveru pro dané časové období, aby nedošlo k zamezení dostupnosti informací v případě nedostatečných zdrojů.

### 5.4.8 Množství přenesených dat

Statistika zobrazuje skutečné množství přenesených dat v daném časovém období. Ke zjištění množství přenesených dat se již nevyužívá měřícího kódu, jako tomu bylo u statistiky zobrazených stránek, neboť ten by nebyl schopen poskytnout požadované informace. Na místo toho se využívá analýzy přístupových protokolů serveru, které umožňují zaznamenat i přenos binárních souborů do nichž nelze umístit měřící kód.

Naměřené hodnoty mohou posloužit ke sledování objemu přenesených dat a využití šířky přiděleného datového pásma za měřené časové období.

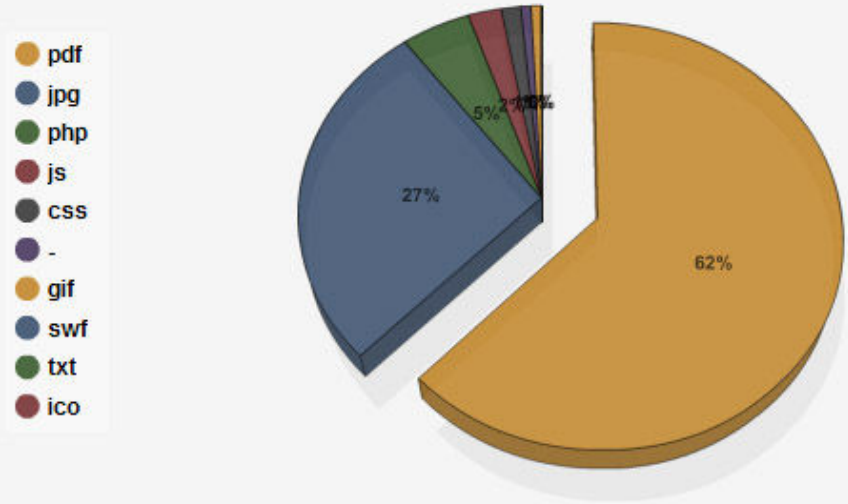



| Přenesená data (MB) | | 19. týden, 2007 |
|---|---|---|
| den | počet | % |
| pondělí | 23.8 MB | 15,81 % |
| úterý | 2.5 MB | 1,63 % |
| středa | 30.6 MB | 20,28 % |
| čtvrtek | 28.2 MB | 18,67 % |
| pátek | 11.0 MB | 7,29 % |
| sobota | 52.7 MB | 34,93 % |
| neděle | 2.1 MB | 1,39 % |
| **celkem** | **150.8 MB** | **100%** |

*Obrázek 5: Množství přenesených dat*

### 5.4.9 Typy souborů a množství jejich stažení

Pro dané časové období obsahuje seznam nejvíce stahovaných typů souborů a množství odeslaných dat v důsledku jejich přenosu. Tato statistika rovněž využívá analýzy protokolů serveru Apache o přístupech klientů na server. Znalost těchto informací umožňuje snížit zatížení serveru například přesunutím velkých a nejvíce žádaných souborů na specializované souborové servery, případně volbou jiného úspornějšího formátu uložených dat či jejich kompresí.




| Typy souborů | | 15. týden, 2007 |
|---|---|---|
| **typ souboru** | **počet** | **%** |
| pdf | 63.5 MB | 62,42 % |
| jpg | 27.8 MB | 27,28 % |
| php | 5.1 MB | 5,01 % |
| js | 2.4 MB | 2,39 % |
| css | 1.4 MB | 1,42 % |
| - | 771.0 KB | 0,74 % |
| gif | 735.7 KB | 0,71 % |
| swf | 37.8 KB | 0,04 % |
| txt | 3.5 KB | 0 % |
| ico | 0 B | 0 % |
| **celkem** | **101.8 MB** | **100%** |

*Obrázek 6: Typy stažených souborů*

### 5.4.10 Chybové kódy

Chybové kódy jsou podmnožinou stavových kódů [15] generovaných internetovým serverem. Poskytují informaci o typu chyby, která nastala v průběhu zpracování klientského požadavku a jsou zasílány v hlavičce odpovědi serveru na požadavek klienta. Statistika zobrazuje jaké požadavky uživatelů na informace či soubory umístěné na serveru nebyly uspokojeny v jednotlivých časových obdobích a k jakým typům chyb nejčastěji docházelo. Díky znalosti chybových kódů je možné

zachytit tzv. „mrtvé odkazy“, což často bývají pozůstatky dřívějších verzí internetové aplikace, jejichž cíl se již na serveru nevyskytuje. Rozbor také může odhalit chybu v aplikaci.

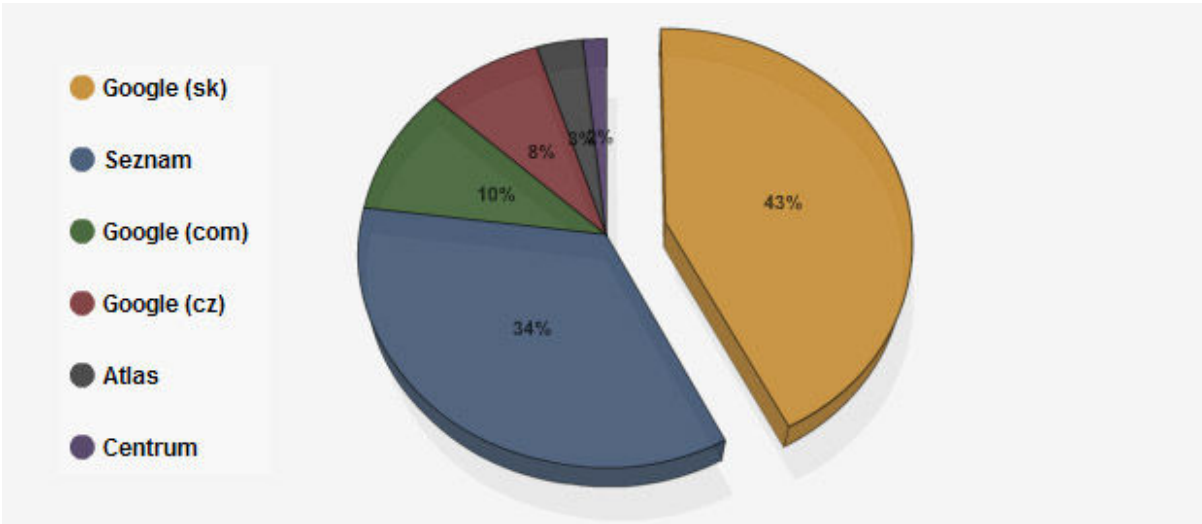
Podle počáteční číslice se rozlišují následující typy chybových kódů serveru:

- 4XX – chyba, jejíž příčina je na straně klienta, např. požadavek na neexistující adresu či soubor na serveru
- 5XX – chyba na straně serveru

### 5.4.11 Odkazující vyhledávače

Pro účely zjištění, ze kterých vyhledávacích portálů přicházejí návštěvníci nejčastěji, slouží statistika odkazujících vyhledávačů. Naměřená data umožní zjistit z jakých vyhledávacích služeb na internetu přicházejí návštěvníci nejčastěji. Získané informace poté mohou sloužit ke zvýšení návštěvnosti internetové aplikace zaměřením pozornosti na konkrétní internetové vyhledávače.

Každý návštěvník, který přistoupí na měřené internetové stránky, odesílá v hlavičce požadavku s názvem *referrer* informaci z jaké internetové adresy přistupuje. Porovnáním adresy s databází adres sledovaných vyhledávačů lze zjistit, zda-li návštěvník přichází z vyhledávací služby.




| Vyhledávače | | 19. týden, 2007 |
|---|---|---|
| **vyhledávač** | **počet** | **%** |
| Google (sk) | 26 | 42,62 % |
| Seznam | 21 | 34,43 % |
| Google (com) | 6 | 9,84 % |
| Google (cz) | 5 | 8,2 % |
| Atlas | 2 | 3,28 % |
| Centrum | 1 | 1,64 % |
| **celkem** | **61** | **100%** |

*Obrázek 7: Odkazující vyhledávače*

Seznam sledovaných vyhledávačů viz [A].

### 5.4.12 Vyhledávaná slovní spojení a jednotlivá slova

Zobrazuje seznam nejvíce vyhledávaných slovních spojení a jednotlivých slov zadaných uživateli do internetového vyhledávače před navštívením vyhledané stránky. Hlavička *referrer* obsahující URL vyhledávače často také obsahuje hledanou frázi zadanou uživatelem. Ta je umístěna v parametru se specifickým názvem pro každý vyhledávač. Díky tomu, že se adresa, ze které pochází požadavek odesílá v hlavičce požadavku, je možné při znalosti formátu adresy zjistit vyhledávané slovní spojení či výraz.

Statistika vyhledávaných slovních spojení obsahuje informaci o četnosti vyhledávaných frází o dvou a více slovech. Statistika vyhledávaných výrazů pak zobrazuje četnosti jednotlivých hledaných slov.

Znalost vyhledávaných dat je klíčová pro vlastníka internetové aplikace, neboť mu poskytuje informaci o tom, jaký obsah uživatele na stránkách skutečně zajímá a vede je k jejich návštěvě.

Příklad řetězce odesílaného v hlavičce *referrer* při vyhledání slovního spojení „statistiky přístupů“ ve vyhledávači Google:

```
http://www.google.cz/search?hl=cs&q=statistiky+p%C5%99%C3%ADstup%C5%AF
```

Jak je patrné, v parametru *q* odesílá prohlížeč vyhledávané slovní spojení (v kódované podobě), které lze zpracovat pro účely statistiky. Seznam parametrů pro jednotlivé sledované vyhledávače viz [A].

| Vyhledávaná slovní spojení | středa, 25. duben 2007 | |
|---|---|---|
| **hodnota** | **počet** | **%** |
| solenoid valve | 1 | 10 % |
| regulačné ventily | 1 | 10 % |
| burkert | 1 | 10 % |
| thermo regulačné ventily priemyselné | 1 | 10 % |
| site:www.burkert.cz proporcionální ventil | 1 | 10 % |
| bürkert | 1 | 10 % |
| piezoventil | 1 | 10 % |
| šikmé ventily | 1 | 10 % |
| solenoid | 1 | 10 % |
| redox-potenciál | 1 | 10 % |
| **celkem** | **10** | **100%** |

| Vyhledávaná slova | středa, 25. duben 2007 | |
|---|---|---|
| **hodnota** | **počet** | **%** |
| ventily | 3 | 16,67 % |
| regulačné | 2 | 11,11 % |
| solenoid | 2 | 11,11 % |
| thermo | 1 | 5,56 % |
| priemyselné | 1 | 5,56 % |
| valve | 1 | 5,56 % |
| burkert | 1 | 5,56 % |
| site:www.burkert.cz | 1 | 5,56 % |
| šikmé | 1 | 5,56 % |
| piezoventil | 1 | 5,56 % |
| redox-potenciál | 1 | 5,56 % |
| bürkert | 1 | 5,56 % |
| proporcionální | 1 | 5,56 % |
| ventil | 1 | 5,56 % |
| **celkem** | **18** | **100%** |

*Obrázek 8: Vyhledávaná slovní spojení a jednotlivá slova*

### 5.4.13 Počítače návštěvníků

Obsahuje seznam počítačů návštěvníků rozlišených podle IP adres a počet zobrazených stránek připadajících na jednotlivé IP adresy. U počítačů, které mají přidělen doménový název a podařilo se jej zjistit, je rovněž zobrazeno jméno počítače. Díky tomu je možné identifikovat například organizace, z nichž návštěvníci pocházejí.

### 5.4.14 Odkazující internetové stránky, servery, domény

Statistika ukazuje, ze kterých externích internetových stránek, serverů a domén nejčastěji přicházejí uživatelé na analyzované stránky. Zobrazená data neobsahují informace o odkazujících vyhledávacích službách, ty jsou ze zobrazení vyňaty, neboť jsou obsaženy v samostatné statistice vyhledávačů.

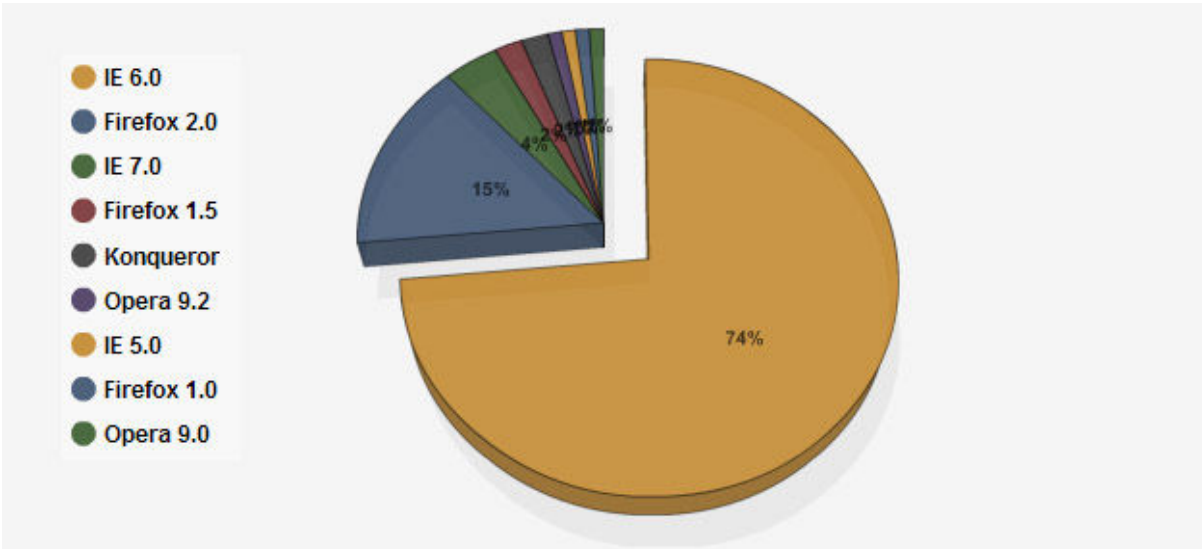
Ze seznamu odkazujících stránek je možné zjistit, kde se na internetu nacházejí zpětné odkazy na měřené stránky a odkud návštěvníci přicházejí.

### 5.4.15 Operační systémy

Jedna ze statistik, která se zaměřuje na technické aspekty návštěv. Zobrazuje informace o operačních systémech, které návštěvníci používají při prohlížení měřených stránek. Jsou zobrazeny rodiny a jednotlivé verze použitých operačních systémů. Statistika zobrazuje naměřená data v koláčovém grafu a také v tabulce. Ze statistiky je možné si utvořit představu o rozšířenosti toho kterého operačního systému.

### 5.4.16 Internetové prohlížeče

Statistika internetových prohlížečů použitých k prohlížení měřených internetových stránek. Opět jsou zobrazeny jak rodiny, tak verze internetových prohlížečů. K identifikaci prohlížeče je využito volně dostupné databáze internetových prohlížečů pro PHP, která obsahuje téměř všechny známé typy a verze internetových prohlížečů [17]. Pokud však zjištěný prohlížeč není v databázi uveden, je položka ve statistice zobrazena jako neznámý typ prohlížeče.

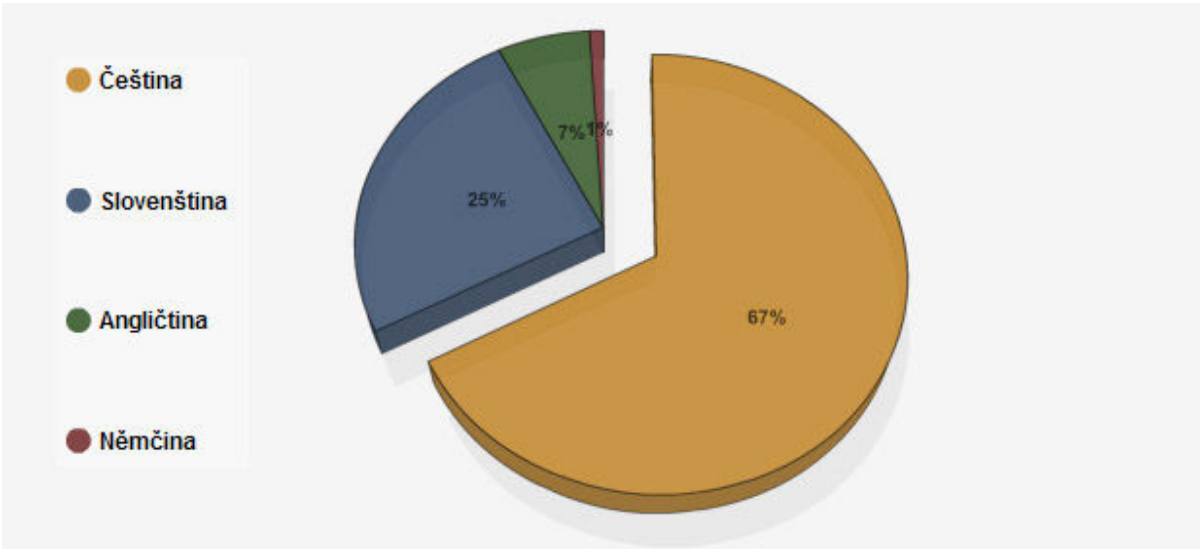



| Verze | | 19. týden, 2007 |
|---|---|---|
| **hodnota** | **počet** | **%** |
| IE 6.0 | 75 | 73,53 % |
| Firefox 2.0 | 15 | 14,71 % |
| IE 7.0 | 4 | 3,92 % |
| Firefox 1.5 | 2 | 1,96 % |
| Konqueror | 2 | 1,96 % |
| Opera 9.2 | 1 | 0,98 % |
| IE 5.0 | 1 | 0,98 % |
| Firefox 1.0 | 1 | 0,98 % |
| Opera 9.0 | 1 | 0,98 % |
| **celkem** | **102** | **100%** |

*Obrázek 9: Internetové prohlížeče*

### 5.4.17 Jazyky

Statistika jazyků, které jsou nastaveny v prohlížečích návštěvníků jako výchozí. Tato statistika však neukazuje, jakým jazykem uživatel komunikuje. Velmi často totiž nastává situace, kdy je v internetovém prohlížeči jako výchozí řeč nastavena angličtina, i když uživatel komunikuje v jiném jazyce. Naopak, pokud je zjištěn jiný jazyk než angličtina, je velmi pravděpodobné, že uživatel v daném jazyce skutečně komunikuje.




| Jazyky | | 19. týden, 2007 |
|---|---|---|
| **jazyk** | **počet** | **%** |
| Čeština | 72 | 67,29 % |
| Slovenština | 27 | 25,23 % |
| Angličtina | 7 | 6,54 % |
| Němčina | 1 | 0,93 % |
| **celkem** | **107** | **100%** |

*Obrázek 10: Výchozí jazyky nastavené v prohlížečích návštěvníků*

### 5.4.18 Obrazovky

Posledním typem statistik, která vypovídá o technickém vybavení návštěvníka, je statistika nastaveného rozlišení a barevnou hloubky obrazovky monitoru.

Díky znalosti technického vybavení návštěvníků je možné optimalizovat aplikaci tak, aby vyhovovala potřebám pro efektivní rozložení a zobrazení informací návštěvníkům.

Více informací o typech internetových statistik viz [4].

## 5.5 Bezpečnostní analýza měřených dat

V této části se zabývám analýzou měřených dat za účelem zjištění pokusů o narušení bezpečnosti internetové aplikace. Zaměřuji se na detekci známých typů pokusů o útok na internetové servery obecně a pokusy o útoky na server prostřednictvím bezpečnostních chyb starších verzí široce rozšířených programů. Jejich bezpečnostní chyby jsou obvykle veřejně publikovány a často využívány jako prostředky k útokům. Zjištěným útočníkům se poté snažím zamezit v dalších pokusech o útok pomocí konfiguračních nástrojů internetového serveru Apache.

Ukazuji také, jak se pomocí systému analyzují pokusy o neoprávněný přístup do zabezpečené části internetové aplikace. Je zde popsán prostředek pro autorizaci registrovaných uživatelů a tvorba bezpečnostní statistiky z chybných pokusů o přihlášení do zabezpečené oblasti.

### 5.5.1 Detekce pokusů o útok na server a prevence proti uskutečnění dalších útoků

Díky dlouhodobé analýze záznamů o přístupech na server je možné vypozorovat často se opakující pokusy o přístup k souborům či adresářům, které se na serveru nenacházejí. Velice často se jedná o adresáře s názvy rozšířených programů, které jsou nainstalovány na mnoha serverech na světě. Jedná se nejčastěji o webové správce databázového serveru, internetová fóra, elektronické obchody a další programy, které jsou buďto volně šířené nebo jsou velmi oblíbené.

Tyto programy mají ve svých dřívějších verzích často nějaké bezpečnostní chyby, které například umožňují útočníkovi proniknout do programu a využít jeho funkce, jejichž použití je jinak omezeno pouze na skupinu oprávněných uživatelů, čímž dojde k ovlivnění činnosti serveru, např. využití jeho prostředků k neoprávněnému účelu či k průniku do samotného operačního systému serveru.

Kromě útoků na programy v jejich obvyklém umístění na serveru se také vyskytují pokusy o útok s využitím známých kritických chyb starších verzí webových serverů, které například umožňovaly, v případě úspěšného spuštění, přístup k operačnímu systému prostřednictvím příkazové řádky, čímž útočník mohl získat úplnou kontrolu nad serverem.

#### *Příklad přístupu ke známé aplikaci s kritickou chybou.*

Mezi nejznámější open-source webové aplikace patří např. phpMyAdmin, což je aplikace sloužící k řízení provozu databázového serveru a ke správě obsahu databáze pomocí webového rozhraní.

Jako příklad je použita verze 2.7.0, u které byla nalezena kritická bezpečnostní chyba, která umožnila útočníkovi obejít bezpečnostní opatření programu a přepsat klíčové systémové soubory, což mělo za následek možnost využívat vzdálené soubory a provádět útoky typu XSS (Cross-site scripting) [14].

Aplikace je distribuována v archivech s názvy ve formátu phpmyadmin-x.y.z-english, phpmyadmin-x.y.z-all-languages a phpmyadmin-x.y.z-all-languages-utf-8-only, kde x.y.x označuje číslo verze. Útočník pak při znalosti kritické chyby u té které verze a znalosti nejčastějších názvů

umístění aplikace na serveru provede například následující útoky z IP adresy 195.101.185.242, které mohou být zaznamenány v protokolu o přístupech následně (pro přehlednost byl odstraněn časový údaj):

```
195.101.185.242 - - […] "GET /phpmyadmin/main.php HTTP/1.1" 404 217
195.101.185.242 - - […] "GET /mysql/main.php HTTP/1.1" 404 212
195.101.185.242 - - […] "GET /admin/main.php HTTP/1.1" 404 212
195.101.185.242 - - […] "GET /db/main.php HTTP/1.1" 404 209
195.101.185.242 - - […] "GET /dbadmin/main.php HTTP/1.1" 404 214
195.101.185.242 - - […] "GET /web/phpMyAdmin/main.php HTTP/1.1" 404 221
195.101.185.242 - - […] "GET /admin/phpmyadmin/main.php HTTP/1.1" 404 223
195.101.185.242 - - […] "GET /admin/pma/main.php HTTP/1.1" 404 216
195.101.185.242 - - […] "GET /phpMyAdmin-2.7.0/main.php HTTP/1.1" 404 223
195.101.185.242 - - […] "GET /phpMyAdmin-2.5.6/main.php HTTP/1.1" 404 223
195.101.185.242 - - […] "GET /phpMyAdmin-2.5.4/main.php HTTP/1.1" 404 223
195.101.185.242 - - […] "GET /phpMyAdmin-2.5.1/main.php HTTP/1.1" 404 223
195.101.185.242 - - […] "GET /phpMyAdmin-2.2.3/main.php HTTP/1.1" 404 223
195.101.185.242 - - […] "GET /phpMyAdmin-2.2.6/main.php HTTP/1.1" 404 223
```

V ukázce jsou zobrazeny typické útoky na aplikaci pro její různé obvyklé umístění na serveru. Ve spodní části jsou pak patrné pokusy o útok na jednotlivé verze aplikace s bezpečnostními chybami.

### *Příklad detekce zneužití kritické chyby v ISS serveru u systému Microsoft Windows.*

V přístupových souborech serveru se mohou objevit pokusy, které se mají za cíl zneužít chyby v operačním systémů Microsoft Windows, známé jako nadbytečné dekódování. Tato bezpečnostní chyba je způsobena opakovaným dekódováním URL požadavku, který obsahuje dvojnásobně zakódované znaky podobně jako následující příklad:

```
http://doména/Scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+\
```

Tento příklad umožní, na postižené verzi IIS serveru, přístup ke kořenovému adresáři systému a způsobí zobrazení obsahu kořenového adresáře, případně je možné jej zneužít k dalším nebezpečnějším útokům.

Díky znalosti těchto typů útoků je možné útoky detekovat a v případě včasného zásahu zamezit útočníkovi v provedení dalších útoku na server, které by jej mohli potenciálně ohrozit.

### *Databáze známých typů pokusů o útok*

Obsahuje známé typy útoků, které jsou nejčastěji prováděny. Databáze obsahuje místo konkrétního řetězce útoku, šablonu daného útoku, která může sloužit pro odhalení více podobných útoků, které jsou založeny na stejném principu.

Obsah databáze je spravován ručně. Do databáze se mohou vkládat obecně známé útoky prezentované na internetu nebo pokusy, které vznikly na základě pozorování přístupů útočníků na konkrétní server. Součástí databáze by však neměly být pouze útoky na konkrétní typ serveru, který je provozován a kterého se sledování týká (tedy Apache, IIS, případně nějaký další), ale kombinace všech známých typů útoku. Útočníci často, v případě vhodně nastaveného serveru, nemají přístup k informacím o typu a verzi serveru. Proto útočníci provádějí různé typy pokusů na různé typy serverů současně pomocí automatizovaného nástroje.

### *Využití chybového dokumentu k detekci pokusů o útok*

Díky tomu, že uživatelsky definovaná chybová stránka může obsahovat i dynamický skript zpracovávaný serverem, je možné do tohoto dokumentu umístit kód provádějící detekci známých pokusů o útok na server porovnáním požadavku uživatele s databází známých typů útoků.

Pokusy o útok jsou prováděny zadáváním adres ke zdrojům či dokumentům, které se na serveru nevyskytují, ale směřují na známé chyby starších verzí internetového serveru, případně na starší verze volně šířených aplikací. Starší verze těchto aplikací často mívají bezpečnostní chyby, které jsou obvykle uveřejňovány po vydání nové verze. Tyto chyby se poté snaží někteří útočníci zneužít k získání kontroly nad internetovým serverem, případně ke vniknutí přímo do operačního systému na serveru.

V případě, že je dojde k vygenerování chybového kódu v důsledku neexistujícího obsahu na serveru (kód chyby 404), porovná se požadavek klienta s tabulkou známých typů útoků. V případě nalezení shody s některým typem útoku se zaznamená pokus o útok a zjistí se jeho závažnost. Je-li typ útoku označen jako kritický nebo již došlo k více nekritickým útokům v krátké době, dojde k časově omezenému zablokování přístupu z IP adresy útočníka.

Kromě detekce útoků chybový dokument zajišťuje také samotnou obranu před dalšími pokusy o útok generováním seznamu zakázaných IP adres, ze kterých byl učiněn pokus o útok. Vygenerovaný seznam IP adres se využívá v konfiguračním souboru *.htaccess*.

### *Zabránění útočníkům v dalších pokusech o útok*

Při detekci neúspěšného požadavku na server se porovná požadavek klienta s databází známých útoků. V případě, že je nalezena v databázi shoda, přístup se označí za pokus o útok. Podle závažnosti útoku se poté určí další postup.

V případě mimořádně závažného typu útoku, který je označen stupněm kritický, dojde automaticky k zabránění dalšího přístupu z dané IP adresy na server na omezenou dobu. Doba zamezení přístupu je omezena z důvodu, že IP adresa útočníka může být přidělována dynamicky a po jistém čase tedy může být přidělena jinému klientovi, kterému by bylo nevhodně bráněno v přístupu k obsahu serveru. Po tuto dobu server nezpracovává požadavky útočníka z dané IP adresy. Je-li po uplynutí doby zaznamenán další pokus o útok ze stejné IP adresy, je doba, po kterou je přístup z dané IP adresy na server omezen, prodloužena.

Pokud je požadavek označen za útok nekritický, považuje se první typ takového útoku za omyl (chybné zadání adresy) a může být proveden i vícenásobně, aniž by se na něj reagovalo zamezením přístupu. V případě, že bude klientem proveden následující útok jiného typu se stejnou nebo vyšší

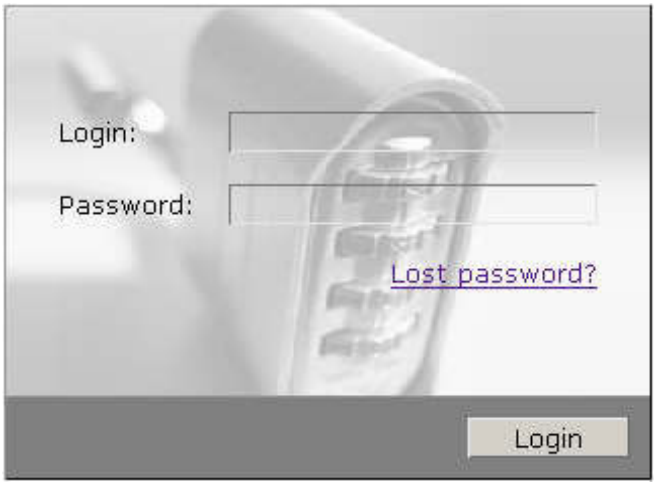
kvalifikací nebezpečnosti, dochází opět k postupnému omezování přístupu na server podobně, jako v předchozím případě. U počtu dvou a více typů pokusů ze stejné IP adresy se k útokům přistupuje, jako by se jednalo o jeden kritický útok.

### *Využití konfiguračního souborů .htaccess k zamezení přístupu ze seznamu zakázaných IP adres*

Do souboru *.htaccess* (viz 2.1.5) je automaticky generováno omezení přístupu z IP adres, ze kterých byly v průběhu posuzovaného času vedeny pokusy o útok. Seznam zakázaných adres je pro jednoduchost automaticky aktualizován při každém dalším pokusu o útok pomocí uživatelsky definovaného souboru pro zobrazení informací o chybném přístupu *ErrorDocument* (viz 2.1.5). Generování seznamu probíhá na základě pravidel o počtu útoků z jedné IP adresy, dále označení stupně kritičnosti útoků a na základě časového rozmezí mezi jednotlivými pokusy o útok.

## 5.5.2 Modul pro autorizaci registrovaných uživatelů

Tento modul je volitelná součást měřených internetových stránek a slouží k zabezpečení oblastí, které nejsou přístupné běžným návštěvníkům internetových stránek, ale pouze registrovaným uživatelům. Modul se načítá automaticky před zobrazením obsahu samotné internetové stránky, kde je umístěn. Je ho výstupní část je tvořena formulářem, do kterého registrovaný uživatel vyplňuje přihlašovací jméno a heslo.




*Obrázek 11: Přihlašovací formulář*

Seznam všech registrovaných uživatelů je umístěn v databázi a záznam o registrovaném uživateli obsahuje následující informace:

- unikátní identifikátor uživatele
- jméno
- příjmení
- adresa elektronické pošty

- přihlašovací jméno
- přihlašovací heslo
- příznak aktivního uživatele

Přístup do zabezpečené oblasti je vázán na zadání platného uživatelského jména a jemu příslušného hesla. Souhlasí-li zadané informace, uživateli je umožněn vstup do zabezpečené oblasti. V případě, že uživatel zapomene přihlašovací informace, je v modulu možnost jejich odeslání na adresu elektronické pošty, která je uvedena v záznamu registrovaného uživatele.

Pokud přihlašovací informace nesouhlasí, modul zapíše do databáze záznam neúspěšného pokusu o přihlášení. Tento záznam se používá při tvorbě bezpečnostních statistik.

Rozlišují se dva případy neúspěšného přihlášení:

- zadání neexistujícího přihlašovacího jména, případně přihlašovacího jména neaktivního uživatele
- zadání chybného hesla ke správnému přihlašovacímu jménu aktivního uživatele

Součástí záznamu o neúspěšném pokusu přihlášení je čas jeho uskutečnění a IP adresa počítače, ze které byl pokus učiněn.

### 5.5.3 Statistiky chybných přihlášení

V informačním systému lze zobrazit statistiky týkající se bezpečnosti, které zobrazují neúspěšné pokusy o přihlášení pomocí chybného přihlašovacího jména či hesla. Tyto varianty jsou zobrazeny odděleně. U každého chybného přihlášení je zároveň patrné, kolik neúspěšných pokusů bylo učiněno z jedné IP adresy. Z těchto informací lze usoudit, zdali se jedná pouze o chybné zadání dat uživatelem, nebo o cílený útok za účelem získat neoprávněný přístup do zabezpečené oblasti. Pokusy, jejichž počet přesáhne stanovenou hranici, jsou zvýrazněny, aby bylo možné identifikovat případné útoky. Majitel sledované aplikace pak může získané informace použít ke zvýšení bezpečnosti aplikace, případně učinit kroky, které povedou k identifikaci útočníka.

## 5.6 Výběr souborů aplikace s popisem obsahu

### 5.6.1 Knihovny

#### *Seznam volně dostupných knihoven, které jsem využil při tvorbě aplikace*

**Smarty**

Nástroj pro práci se šablonami souborů. Umožňuje oddělit strukturu dokumentu od obsahu. Obsahuje kompilátor šablon a správu vyrovnávací paměti pro zrychlení generování. [12]

**PHP/SWF Charts**

Používá se pro zobrazení grafů statistik. Umožňuje vykreslovat množství různých typů grafů. Vzhled grafu je možné libovolně přizpůsobit potřebám pomocí konfigurace v jazyce PHP. Knihovnu je možné používat zdarma, avšak má jediné omezení, které spočívá v tom, že po kliknutí myší na zobrazený graf se otevře nové okno prohlížeče obsahující domovskou stránku knihovny. [13]

**ezSQL**

Databázová třída obsahující funkce pro práci s databází MySQL. Nabízí větší míru abstrakce při používání databáze, než vestavěné funkce jazyka PHP. [19]

**PHPMailer**

Třída pro odesílání elektronické pošty prostřednictvím PHP. Pomocí objektového přístupu umožňuje konfigurovat širokou škálu vlastností odesílané emailové zprávy jako odeslání emailu seznamu příjemců, textový i HTML formát zprávy a další. [20]

## *Seznam knihoven vytvořených v rámci projektu*

**class.ApacheLog.php**

Obsahuje třídy pro analýzu přístupových protokolů serveru Apache. Třída ApacheLogParser se stará načítání záznamů protokolů a jejich transformaci na vhodnou strukturu, kterou využívá třída ApacheLogStats ke generování vlastních statistik a také k uložení zjištěných hodnot do databáze za účelem zrychlení analýzy.

**class.auth.php**

Třída pro autorizaci uživatelů. Umožňuje přihlašování a odhlašování registrovaných uživatelů, kteří mají aktivní záznam v tabulce uživatel. Využívá třídu pro uložení aktuální relace uživatele. Zároveň ukládá záznamy neúspěšných pokusů o vstup do zabezpečených částí aplikace.

**class.htaccess.php**

Třída pro generování adresářově zaměřených konfiguračních souborů *.htaccess*. Využívá se v bezpečnostní části projektu k tvorbě seznamu zakázaných IP adres, kterým má být omezen přístup na server z důvodu vícenásobných pokusů o útok.

**class.log.php**

Třída pro ukládání záznamů o práci aplikace do souboru. Analýzou záznamů v souboru lze zjistit jaké požadavky na zobrazení internetových stránek byly akceptovány a zpracovány, což umožňuje odhalit případné chyby v aplikaci.

**class.Sessions.php**

Třída pro práci s hodnotami, které se vztahují k aktuální relaci. Relací je myšlen vztah návštěvníka a měřených internetových stránek. Nejčastěji se používá pro uložení dočasných informací o uživateli na straně serveru, například přihlašovacích informací.

**functions.php**

Obsahuje obecné funkce, které jsou používány různými částmi aplikace.

### 5.6.2 Soubory s nastavením aplikace

**config.php**

Nastavení, které ovlivňuje běh aplikace. Obsahuje přihlašovací informace k relační databázi. Jedná se o jediný soubor, který je nutné změnit v případě požadavku nasazení aplikace na jiný internetový server. Záměrně jsem se snažil omezit nutnost editace více souborů, aby aplikaci mohl případně nainstalovat i laik, který neovládá jazyk PHP.

**tables.php**

Obsahuje výčet databázových tabulek, používaných v rámci aplikace.

### 5.6.3 Hlavní soubory aplikace

**hit.php**

Třída pro sběr a zpracování měřených dat. Zpracovává veškeré měřené veličiny, které jsou získány v rámci měření návštěvnosti.

**errordoc.php**

Uživatelský chybový dokument serveru Apache, který se stará o detekci známých typů útoků a zjištění útočníků. Na základě zjištěného počtu a nebezpečnosti útoků generuje seznam zakázaných IP adres. Ten se využívá k zabezpečení serveru před dalšími pokusy o útok ve spolupráci s knihovnou *class.htaccess.php*.

**common_stats.php**

Zpracování a zobrazení obecných statistik, které nejsou rozděleny na časové úseky. Používá se při zobrazení statistik operačních systémů, internetových prohlížečů a obrazovek. Výběr statistiky se provádí pomocí parametru souboru.

**referrers.php**

Zobrazuje odkazující internetové stránky, servery a domény.

**security_stats.php**

Bezpečností statistika. Obsahuje statistiky chybně zadaných uživatelských jmen a hesel.

**time_stats.php**

Zobrazuje statistiky tvořené pro časová období. Obsahuje statistiky počtu unikátních návštěvníků, návštěv, zobrazených stránek a unikátních IP adres.

**various_stats.php**

Obsahuje další nezařazené statistiky. Používá se pro zobrazení statistik výchozích jazyků v internetovém prohlížeči.

### 5.6.4 Dokumentace zdrojových kódů

Detailní popis programové části je dostupný v dokumentaci zdrojových kódů na přiloženém CD.

# 6 Závěr

V této práci jsem se zaměřil na dva odlišné způsoby měření návštěvnosti internetových aplikací a tvorbu statistik přístupů se získaných dat. Rozebral jsem možnosti zjišťování informací o návštěvách dvěma zcela rozdílnými způsoby, pomocí měřícího kódu vloženého do měřené aplikace a pomocí analýzy přístupových protokolů internetového serveru, zaznamenaných za běhu aplikace. Ukázal jsem, že oba způsoby měření mají jen málo společných rysů a největší hodnotu poskytuje provozování obou způsobů měření současně.

Pro účel zobrazení naměřených dat a tvorbu statistik přístupů jsem zhotovil informační systém, který poskytuje možnost provádět monitorování více internetových aplikací současně a zároveň umožňuje tvorbu statistik přístupů z naměřených dat v reálném čase. Systém umožňuje zobrazit informace o množství návštěvníků, počtu shlédnutých stránek, množství přenesených dat, odkazující vyhledávače a mnohá další zjištěná data pro různá časová období včetně zobrazení dlouhodobých průměrných hodnot.

V další části jsem se zaměřil na oblast bezpečnosti sledovaných aplikací a internetového serveru pomocí detekce známých typů útoků a prevence proti uskutečnění dalších pokusů o útok od stejného útočníka. To je zajištěno spoluprací zhotoveného systému s internetovým serverem Apache, který poskytuje nástroje pro řízení přístupu internetových klientů k provozované aplikaci. Díky tomu je možné generovat seznam zakázaných IP adres klientů, u nichž byl zaznamenán pokus o útok či zneužití bezpečnostní chyby některých starších verzí hojně využívaných serverových aplikací. Zjištěné pokusy o útok se rovněž zaznamenávají pro účely zobrazení v systému.

Součástí bezpečnostní části systému je také modul, který umožňuje provádět autorizaci uživatelů a řídit přístup do zabezpečené oblasti internetové aplikace. Provádí rovněž sledování pokusů, kdy útočník zadává kombinace přihlašovacích jmen a hesel za účelem neautorizovaného přístupu.

Díky spojení dvou přístupů měření vznikl systém, který poskytuje mnoho užitečných informací umožňujících sledovat nejen vývoj návštěvnosti internetové aplikace, ale také dopad jejího provozu na vytížení internetového serveru v čase a umožňuje zvýšit bezpečnost provozovaných internetových aplikací.

# 7 Literatura


[ 1] Kolektiv autorů. *PHP Programujeme profesionálně*. Brno: Computer Press, 2001. ISBN 80-7226-310-2.

[ 2] SCHLOSSNAGLE, George. *Pokročilé programování v PHP5.* Brno: Zoner Press, 2004. ISBN 80-86815-14-5.

[ 3] *Platform for Privacy Preferences (P3P) Project* [online]. URL: <http://www.w3.org/P3P/> [cit. 2005-05-03]

[ 4] *International Federation of Audit Bureaux of Circulations* [online]. URL: <http://www.ifabc.org/> [cit. 2005-05-03]

[ 5] *PHP: Hypertext Preprocesor* [počítačový program, online]. Ver. 5.2.1. URL: <http://www.php.net/>

[ 6] *JavaScript* [online]. Ver. 1.5. URL: <http://en.wikipedia.org/wiki/JavaScript>

[ 7] *MySQL: The world's most popular open source database* [počítačový program, online]. Ver. 5.0. URL: <http://www.mysql.com/>

[ 8] *The Apache HTTP Server Project* [počítačový program, online]. Ver. 2.2. URL: <http://httpd.apache.org/>

[ 9] *Google analytics* [online]. URL: <http://www.google.com/analytics/> [cit. 2006-12-26]

[10] *Navrcholu.cz* [online]. URL: <http://navrcholu.cz/> [cit. 2006-12-26]

[11] *AWStats* [počítačový program, online]. Ver. 6.5. URL: <http://awstats.sourceforge.net/>

[12] *Smarty: Template Engine* [PHP knihovna, online]. Ver. 2.6.9. URL: <http://smarty.php.net/>

[13] *PHP/SWF Charts* [PHP knihovna, online]. Ver. 4.4. URL: <http://www.maani.us/charts/>

[14] *Cross-site scripting* [online]. URL: <http://en.wikipedia.org/wiki/Cross-site_scripting> [cit. 2007-04-11]

[15] *Status Code Definitions* [online]. URL: <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html> [cit. 2007-04-11]

[16] *mod_log_rotate* [Apache modul, online]. Ver. 08_2006. URL: <http://www.sitebuddy.com/mod_log_rotate>

[17] *browscap.ini* [PHP modul, online]. Ver. 3945. URL: <http://www.garykeith.com/browsers/downloads.asp>

[18] *Gettext* [PHP modul, online]. Ver. N/A. URL: <http://www.php.net/manual/en/ref.gettext.php>

[19] *ezSQL* [PHP knihovna, online]. Ver. 2.03. URL: <http://www.woyano.com/jv/ezsql>

[20] *PHPMailer* [PHP knihovna, online]. Ver. 1.73. URL: <http://phpmailer.sourceforge.net/>

# 8 Přílohy

## A. Sledované vyhledávací služby a parametry obsahující hledané výrazy

| **vyhledávací služba** | **adresa** | **parametr** |
|---|---|---|
| AOL | aol.com | query |
| Atlas | search.atlas.cz | w |
| Centrum | search.centrum.cz | q |
| del.icio.us | del.icio.us | p |
| Google (cz) | google.cz | q |
| Google (sk) | google.sk | q |
| Google (com) | google.com | q |
| Google (de) | google.de | q |
| jyxo | jyxo.cz | s |
| Morfeo | morfeo.centrum.cz | q |
| Live Search (MSN) | search.msn.com | q |
| Live Search | search.live.com | q |
| Seznam | search.seznam.cz | w |
| TOPlist | toplist.cz | search |
| Volný | volny.cz | search |
| Yahoo! | search.yahoo.com | p |
| Zoohoo | zoohoo.cz | q |
| Zoznam | zoznam.sk | s |

# B. Ukázka šablony pro generování statistik

```
{foreach item=stat from=$stats}
   {if $stat.chart}
   <div class="chart">
     <br /><br />
     {$stat.chart}
   </div>
   {/if}

   {if $stat.menu}
   <div class="stat_menu">
      <ul>
        {foreach item=count key=code from=$stat.menu}
          <li><a href="?error_code={$code}" title="{$code}">{$code}
({$count})</a></li>
        {/foreach}
      </ul>
   </div>
   {/if}

   <table class="stat_hours" cellspacing="0">
      <tr>
          <th align="left">{$stat.title}</th>
          <th colspan="2" align="right"
nowrap="nowrap">{$date_string}</th>
      </tr>
      <tr>
          <td class="title">{$stat.value_title}</td>
          <td class="title" width="100" align="right">{t}count{/t}</td>
          <td class="title" width="100" align="right">%</td>
      </tr>
      {foreach item=row from=$stat.content}
      <tr class="{cycle values="odd,even"}">
          <td>{$row.value}</td>
          <td align="right">{$row.count}</td>
          <td align="right">{$row.percent} %</td>
      </tr>
      {/foreach}
      <tr>
          <td class="title">{t}total{/t}</td>
          <td class="title" align="right">{$stat.sum}</td>
          <td class="title" align="right">100%</td>
      </tr>
   </table>
   <br />
{/foreach}
```

# C. Seznam adresářů a souborů aplikace

**Adresáře:**
***configs/***
***doc/***
***images/***
***libs/***
***locale/***
***logs/***
***sql/***
***styles/***
***templates/***
***templates_c/***

**Knihovny:**
*libs/PHP_SWF_Charts/*
*libs/class.ApacheLog.php*
*libs/class.auth.php*
*libs/class.log.php*
*libs/class.mysql_backup.php*
*libs/class.Sessions.php*
*libs/functions.php*
*libs/tsmarty2c.php*

**Soubory aplikace:**
*.htaccess*
*sitecode*
*code.js*
*auth.php*
*common_stats.php*
*errordoc.php*
*header.php*
*hit.php*
*hosts.php*
*index.php*
*lead_in.php*
*lead_out.php*
*navigation.php*
*referrers.php*
*security_stats.php*
*servers.php*
*sitecode.php*
*time_stats.php*
*various_stats.php*

**Nastavení aplikace:**
*configs/config.php*
*configs/definitions.php*
*configs/tables.php*

**Kaskádové styly:**
*styles/auth.css*
*styles/dwstats.css*

**Výstupní šablony:**
*templates/auth.tpl*
*templates/auth_mail.tpl*
*templates/calendar.tpl*
*templates/common_stats.tpl*
*templates/components.tpl*
*templates/errordoc.tpl*
*templates/footer.tpl*
*templates/header.tpl*
*templates/hosts.tpl*
*templates/informations.tpl*
*templates/main.tpl*
*templates/menu_left.tpl*
*templates/security_stats.tpl*
*templates/servers.tpl*
*templates/sitecode.tpl*
*templates/url_attacks.tpl*

# D. Uživatelská příručka

Zde nalezete popis instalace a základní nastavení aplikace vytvořené v rámci diplomového projektu. Je zde rovněž uvedeno důležité nastavení internetového serveru, databázového serveru a jazyka PHP, které je podmínkou správné funkce aplikace.

Samotná aplikace je platformě nezávislá. Pro její funkčnost je však nutná podpora základních nástrojů (ze strany OS), které byly použity k tvorbě aplikace.

## Instalace aplikace

Před samotnou instalací a konfigurací aplikace je nutné mít v operačním systému nainstalovaný a nakonfigurovaný internetový server Apache a v něm nastavenu podporu jazyka PHP formou modulu. Dále musí být nainstalovaný databázový server MySQL. Instalace těchto požadovaných částí a jejich správná konfigurace není triviální a vyžaduje znalost příslušné dokumentace k jednotlivým nástrojům. Seznam verzí použitých nástrojů viz 5.1.

Instalace je především zaměřena na webové správce, kteří rozumí dané problematice a rozhodně není určena pro obyčejného uživatele, který nemá zkušenosti s nastavením serveru Apache.

### *Nastavení konfiguračních souborů serveru Apache*

**httpd.conf**

```
LoadModule log_rotate_module modules/mod_log_rotate.so
RotateLogs On
RotateLogsLocalTime On

<Directory "/www">
...
    AllowOverride All
</Directory>

<FilesMatch "^\.ht">
    Order allow,deny
    Deny from all
</FilesMatch>

<Files ~ "\.(log|log\..*)$">
   Order deny,allow
   Deny from all
</Files>

ErrorDocument 404 „cesta k souboru errordoc.php“
```

**httpd-vhosts.conf**

Pro každou sledovanou aplikaci nastavit:

```
# virtuální hostitel "příklad"
<VirtualHost 127.0.0.1>
...
    CustomLog /www/"příklad"/logs/access.log combined
    ErrorDocument 404 /errordoc.php?server="databáze sledované aplikace"
</VirtualHost>
```

## *Nastavení konfiguračního souboru PHP5*

**php.ini**

```
safe_mode = Off
max_execution_time = 180 ; případně více
memory_limit = 128M ; případně více
error_reporting = E_ALL & ~E_NOTICE
register_globals = Off
magic_quotes_gpc = Off
extension=php_gettext.dll
extension=php_mbstring.dll
extension=php_mysql.dll
browscap = "cesta k souboru browscap.ini" ; viz [17]
date.timezone="příslušná časová zóna" ; např. Europe/Prague
```

## *Nastavení konfiguračního souboru aplikace*

**configs/config.php**

Nastavení, které ovlivňuje běh aplikace. Obsahuje přihlašovací informace k relační databázi MySQL. Jedná se o jediný soubor aplikace, který je nutné modifikovat z důvodu provedení nové instalace.

Některé důležité parametry, které je nutné vyplnit před spuštěním instalačního skriptu:

```
# URL provozovaného systému
  define ("SERVER_URL", "http://dwstats/");
# SMTP pro odesílání emailů (mmj. v modulu pro autorizaci uživatel)
  define ("SMTP", "smtp.mail.cz");

# umístění hostitelského serveru mysql
  define ("DBhost", "localhost");
# přihlašovací jméno k mysql
  define ("DBuser", "root");
# přihlašovací heslo k mysql
  define ("DBpass", "root");
# název hlavní databáze s informacemi o měřených serverech
  define ("DBadmin", "dwstats");
```

### *Zahájení instalace*

Pro započetí instalace zadejte v okně internetového prohlížeče cestu k instalačnímu skriptu:

```
"URL adresa systému"/install.php
```

Instalační skript vytvoří základní strukturu sdílené databáze pro všechny měřené servery a umožňuje založit databázi pro tvorbu statistik jednoho měřeného serveru. Další měřené servery je případně možné vkládat a upravovat v administrační části aplikace pomocí XML souboru, viz [D]. Skript rovněž vyžaduje zadaní přihlašovacích informací pro vytvoření administrátorského účtu, který zajistí vstup do systému.

Po dokončení instalace je vhodné z bezpečnostních důvodů odstranit instalační skript *install.php* z adresáře aplikace.

## Spuštění aplikace a generování měřícího kódu

Po instalaci je možné se přihlásit do systému pomocí vytvořených přístupových údajů zadaných v instalačním skriptu.

V administrační části systému se nachází funkce pro vygenerování měřícího kódu, který je nutné následně vložit do měřených stránek internetové aplikace. Dále je možné pomocí XML souboru upravovat a vkládat další měřené aplikace. XML soubor je nutné editovat ručně v textovém či tabulkovém editoru a následně jej nahrát do systému, který zajistí potřebné úpravy. Výhoda použití XML souboru namísto formuláře spočívá v možnosti hromadného vložení a editace webů. Obsah souboru musí odpovídat struktuře v následujícím příkladě:

```
<dwstats>
  <servers>
    <server>
      <id>"id editované aplikace 1"</id>
      <dbname>"název databáze 1"</dbname>
      <name>"název aplikace 1"</name>
      <uri>"fyzické umístění na serveru 1"</uri>
      <url>"URL adresa aplikace 1"</url>
      <description>"popis aplikace 1"</description>
    </server>
    <server>
      <id>"prázdné – vkládání nové aplikace 2"</id>
      <dbname>"název databáze 2"</dbname>
      <name>"název aplikace 2"</name>
      ...
    </server>
    ...
  </servers>
</dwstats>
```

Hodnoty se vkládají do značek bez uvozovek. Hodnoty značek <*uri*> a <*description*> jsou nepovinné.

V případě odstranění měřené aplikace ze systému se před vymazáním dat provede jejich automatická záloha do souboru *název_databáze.sql* do adresáře *sql/* v systému.

# Ovládání systému

V horní části systému se nachází menu pro výběr jazyka, které umožňuje kdykoli v průběhu práce v systému zobrazit odlišnou jazykovou verzi. K výběru zobrazení jednotlivých statistik slouží hlavní menu v levé části systému. Obsahuje položky, jejichž názvy odpovídají některému typu sledovaných statistik.