Hostingový systém a zabezpečení serveru studentských projektů

Secure hosting system for student projects server

Bc. Tomáš Zimáček

Diplomová práce
2009

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2008/2009

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Tomáš ZIMÁČEK**
Studijní program:	**N 3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Téma práce:	**Hostingový systém a zabezpečení serveru studentských projektů**

Zásady pro vypracování:

Navrhněte a implementujte řešení pro hosting studentských projektů na serveru studenti.fai.utb.cz (Linux Debian):

1. **-seberegistrace studentů s ověřením proti LDAP serveru UTB,**
2. **-automatické vytvoření účtů pro SFTP, WWW, MySQL, PostgresSQL,**
3. **-automatické nastavení kvót,**
4. **-oddělení uživatelů tak, aby neviděli a nemohli ovlivnit cizí soubory na serveru,**
5. **-monitoring provozu - detekce nadměrného využívání kapacity serveru určitými uživateli,**
6. **-anti-intrusion systém,**
7. **-zálohování svstému.**

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **J. Zeldman: Tvorba webů podle standardů, Computer Press 2004, ISBN: 80-251-0347-1**
2. **M. Kysela a kolektiv: 333 tipů a triků pro Linux, Computer Press 2007, ISBN: 80-722-6866-X**
3. **R. Flickenger: Linux server na maximum, Computer Press 2005, ISBN: 80-251-0586-5**
4. **B. Hatch, J. Lee, G. Kurtz: Hacking bez tajemství, Computer Press 2003, ISBN: 80-7226-869-4**

Vedoucí diplomové práce: **Ing. Tomáš Dulík**
Ústav aplikované informatiky

Datum zadání diplomové práce: **20. února 2009**

Termín odevzdání diplomové práce: **27. května 2009**

Ve Zlíně dne 13. února 2009

prof. Ing. Vladimír Vašek, CSc.
děkan

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

doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Webhostingový trh je již velmi nasycen, ale ne vždy jsou nabízené programy vhodné pro práce studentů a velmi často se jedná o placené služby. Služby zdarma mají většinou hodně velká omezení, a to jak v technologických možnostech, tak i ve stabilitě dané služby.

Projekt **LW Hosting** názorně dokazuje, že provoz vlastního studentského web-hostingu je reálný a především udržitelný. Výhodou pro studenty je především nulová cena a rychlost zřízení. Pro správce systému jsou důležité především prvky zabezpečení a rozšířitelnost celého systému.

Klíčová slova: LW Hosting, wehosting, zabezpečení, modularita, GNU/Linux, Apache, PHP, MySQL, firewall, anti-intrusion systém

ABSTRACT

Web hosting market is saturated lately, but offered programs are not always suitable for student purposes and these services are often charged for. Free of charge services do have prevailingly major user limitations in technological posibilities as well as in constancy of given service.

Project **LW Hosing** clearly demonstrates that functioning of own student web hosting service is feasible and primarily sustainable. The greatest benefit for students is in the zero cost and promptness of establihment. Elements of security and expandability of whole systém are the most important aspects for system administrator.

Keywords: LW Hosting, web hosting, security, modularity, GNU/Linux, Apache, PHP, MySQL, firewall, anti-intrusion system

Děkuji své rodině a přítelkyni za trpělivost a podporu. Svému psovi za to že byl rozumný a pochopil, že při programování si opravdu nemůžu hrát, i když bych velmi chtěl.

Prohlašuji, že

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

Prohlašuji,

že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval.

V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně

…….

Podpis diplomanta

OBSAH

ÚVOD

Projekt **LW Hosting** je příkladem řešení hostingového serveru s webovou administrací pro studenty nejen vysokých škol. Je kladen důraz především na zabezpečení, jednoduchost a rozšiřitelnost. Celý systém musí obstát jak v náročném prostředí internetu plného hrozeb, tak musí být přínosem pro studenty a správce serveru.

Klíčovou vlastností je automatizovaný provoz. Každý student si může sám vytvořit *hostingový účet*, *databázi* a *FTP přístup*. Systém se již postará o přidělení práv, nastavení kvót, vytvoření konfiguračních souborů. Parametry použité při tvorbě nového účtu si přímo pomocí webové administrace nastavuje správce systému, a to globálně pro všechny nové účty. Díky tomu je *registrace uživatele* a *vytvoření webhostingu* otázkou několika málo minut a vše probíhá bez nutnosti jediného zásahu administrátora.

Dále je možné libovolný uživatelský účet upravit, přenastavit kvóty. Tímto je systém připraven i na úspěšné studentské projekty, které mohou mít vyšší nároky na diskový prostor a šířku přenosového pásma.

Proti případným hrozbám z internetu je na serveru nakonfigurován firewall a provoz je kontrolován *anti-intrusion systémem*. Samozřejmě je myšleno i na hrozbu v podobě studenta. Jednotlivé studentské sekce jsou důkladně odděleny, *PHP skripty* jsou spuštěny a provozovány pod vlastními uživatelskými účty a je kladen důraz na bezpečnost použitých hesel.

Mezi jednu z posledních vlastností patří i automatické rozdílové zálohování, které je prováděno každou hodinu a jednou týdně kompletní zálohování.

Celkový koncept hostingového systému je postaven tak, aby byla možná snadná rozšiřitelnost. Jednotlivé sekce jsou rozděleny do modulů, které se navzájem neovlivňují a je tak možné kdykoliv modul zakázat a nebo naopak přidat zcela nový. Mezi další klíčové vlastností patří i rozdělení na část obsahovou a programovou, cachování obsahu a *hezká URL*. Systém je navíc naprogramován s využitím znalostí o objektově orientovaném programování *skriptovacího jazyka PHP 5*, což přispívá k čitelnosti kódu a k následnému rozšiřování celého projektu.

I. TEORETICKÁ ČÁST

1 LW HOSTING

Moderní hostingový systém, který automatizuje běžné kroky nutné při tvorbě nového web-hostingu. Skládá se ze serveru s operačním systémem *Debian GNU/Linux* a webové administrace. Administrace využívá moderních technologií, jako je například *AJAX* a pro její běh je potřeba minimálně *PHP* verze 5.2 a *MySQL server*.

1.1 Open source software

Open source a nebo také open-source software (OSS). Otevřený zdrojový kód zde znamená jak technickou dostupnost kódu, tak legální dostupnost licencí, která umožňuje kód prohlížet a upravovat.

1.1.1 Použitý software

Pro vývoj a provoz projektu **LW Hosting** je potřeba software dostupný pod licencí open-source software. Nespornou výhodou nasazení takového systému jsou nulové zřizovací náklady a díky instalačnímu skriptu i snadnost zprovoznění.

Jedná se o tento software:

- Debian GNU/Linux [7] operační systém
- Apache [4] webový server
- MySQL [9] databázový server
- PHP [12] skriptovací jazyk
- Shorewall [16] firewall
- Fail2ban [8] zabezpečení serveru
- OpenSSH [11] šifrovaný přístup k shellu
- Postfix [13] Mail server
- Pure-FTPd [14] FTP server
- rsync [15] Synchronizace a zálohování

2 VÝVOJ SYSTÉMU

2.1 Volba distribuce

Při hledání nejvhodnější distribuce pro server bylo mé rozhodnutí jednoznačné a velice rychlé. Zvolil jsem *Debian* pro svou stabilitu a jednoduchou údržbu. Hlasy navíc získal i vydáním dlouho očekáváné 5. verze s označením „*Lenny*“. Tato nová verze přináší opět nové verze programů a stejně jako její předchůdci se vyznačuje svou stabilitou. I proto je velmi oblíbený zejména pro serverové instalace, naopak jeho podíl na desktopech v posledních letech poněkud poklesl, zejména po příchodu *distribuce Ubuntu* [18], která je na *Debianu* založena. *Ubuntu* je systém určený speciálně pro desktopové užívání a právě proto obsahuje ten nejnovější software, který ovšem není vždy zcela otestován a není tím zaručena 100% stabilita systému, které se blíží právě *Debian*.

Debian GNU/Linux nevyvíjí komerční subjekt, ale je vyvíjena velkým množstvím dobrovolníků z celého světa. Je známa především svou konzervativností. Přesto je to jedna z nejrozšířenějších linuxových distribucí na světě.

Oproti klasickým „*komerčním distribucím*“, jako jsou *Red Hat*, *Mandriva* nebo *SUSE*, nepoužívá balíčkovací systém postavený na *RPM* (RPM Package Manager, dříve Red Hat Package Manager), ale má vlastní balíčkovací systém tzv. *deb-balíčků*, který je velmi propracovaný a umožňuje velmi jednoduše provádět správu balíčků z různých zdrojů. Nazývá se *Advanced Packaging Tool* (APT).

2.2 Vývoj internetu

2.2.1 World Wide Web

Tato služba známe pod její zkratkou *www* nebo taky *web*. Byla spuštěna v roce 1990. Autorem webu je Tim Berners-Lee, který navrhl jak *jazyk HTML*, *protokol HTTP*, tak i první internetový prohlížeč *WorldWideWeb* a spustil i první webový server na světě.

2.2.2 HyperText Transfer Protocol

HTTP je přenosový protokol, který zajišťuje přenos *HTML stránek* ze serveru do internetového prohlížeče. Dnešním standardem je *HTTP 1.1*, který podporují jak prohlížeče, tak i webové servery. Tento protokol je společně s elektronickou poštou tím nejvíce využívaným a právě díky němu se velmi rychle rozvinul celý internet.

Protokol funguje způsobem dotaz-odpověď. Uživatel (pomocí programu, obvykle internetového prohlížeče) pošle serveru dotaz ve formě čistého textu, obsahujícího označení požadovaného dokumentu, informace o schopnostech prohlížeče apod. Server poté odpoví pomocí několika řádků textu popisujících výsledek dotazu (zda se dokument podařilo najít, jakého typu dokument je, atd.), za kterými následují data samotného požadovaného dokumentu.

HTTP definuje několik metod při práci s dokumentem:

- CONNECT – spojí se s objektem přes uvedený port, používá se při použití proxy.
- DELETE – smaže objekt ze serveru, jsou nutná oprávnění.
- GET – výchozí metoda při dotazu na zobrazení internetových stránek, data přenáší v URL adrese a jejich velikost je limitována 512b.
- HEAD – metadata o požadovaném cíli, jako je například velikost, typ, datum změny, ...
- OPTIONS – dotaz na server, jaké podporuje metody.
- POST – nejčastěji se používá např. při odesílání formuláře, funguje podobně jako GET, ale je možné přenést větší množství dat.
- PUT – nahraje data na server, objektem zde je název nahrávaného souboru. Používá se velmi zřídka, protože pro nahrávání dat na server jsou přímo protokoly FTP nebo SCP/SSH.
- TRACE – odešle kopii požadavky zpět odesilateli. Tímto můžeme zjistit, co na požadavku mění servery, přes které jde.

2.2.3 Uniform Resource Locator

URL je řetězec znaků s definovanou strukturou, který slouží k přesné specifikaci umístění zdrojů informací (ve smyslu dokument nebo služba) na Internetu. Některá pole jsou nepovinná – buď nemají význam, nebo se předpokládá předdefinovaná hodnota, závislá např. na schématu (např. pro *protokol HTTP* je implicitní port 80), nebo na aplikaci (pro webový prohlížeč se předpokládá *protokol HTTP*).

Struktura URL (http://www.notif.info:80/news/editovat?id=21):

- Schéma: HTTP – odpovídající protokolu téhož jména,
- server: www.notif.info,

- port: 80 – jelikož pro HTTP je port 80 implicitní, není ho třeba v tomto konkrétním případě uvádět,
- dokument: /news/editovat – je uveden včetně cesty (adresáře) v rámci webserveru,
- parametry: jsou-li specifikovány, jsou uvozeny znakem otazníku. Zde je první parametr se jménem „id“ a hodnotou „21“. Pokud je parametrů více, tak se oddělují ampersandem,

URL definuje doménovou adresu serveru, umístění zdroje na serveru a protokol, kterým je možné zdroj zpřístupnit.

2.2.4 Hypertext Preprocessor

PHP (rekurzivní zkratka PHP: Hypertext Preprocessor, „PHP: Hypertextový preprocesor“) je skriptovací programovací jazyk. Verze *PHP 1.0* byla vydána již v roce 1995. Od té doby ušel tento skriptovací jazyk velký kus cesty. V současné době je ve verzi 5 a již velmi dobře podporuje i *objektové programování*. I proto se velmi často využívá při tvorbě webových aplikací. Méně časté je využití *PHP* k tvorbě konzolových a desktopových aplikací, ale i to je možné.

PHP skripty jsou prováděny na straně serveru, k uživateli je přenášen až výsledek jejich činnosti. Syntaxe jazyka je inspirována několika programovacími jazyky (*Perl*, *C*, *Pascal* a *Java*). *PHP* je nezávislý na platformě, skripty fungují bez větších úprav na mnoha různých operačních systémech. Podporuje celou řadu internetových protokolů, z nichž jsme některé využili i v naší práci. Jsou to například: *HTTP*, *SMTP* a *LDAP*.

PHP se stalo velmi oblíbeným především díky jednoduchosti použití a tomu, že kombinuje vlastnosti více programovacích jazyků a nechává tak vývojáři částečnou svobodu v syntaxi. Nejčastěji jej můžeme vidět v kombinaci s operačním systémem *GNU/Linux*, databázovým systémem *MySQL* a webovým serverem *Apache*. Této kombinaci už se vžilo pojmenování *LAMP* (Linux, Apache, MySQL, PHP).

2.2.5 OOP

Objektově orientované programování je metodika vývoje software. Představuje novější přístup k vytváření programů oproti metodě strukturovaného programování. Metody OOP napodobují vzhled a chování objektu z reálného světa s možností velké abstrakce. Přínosem OOP je také větší strukturovanost a modularita vytvářeného programu.

V určitých programovacích jazycích, jako je například *Java* nebo *Python* se s objekty setkáte doslova na každém kroku. Nevyhneme se jim ani při tvorbě jednoduchých programů.

Objekty v sobě zahrnují nejen data ale i funkce, které nad uvedenými daty pracují. Data i funkce jsou svázány dohromady takovým způsobem, že objekt můžete předat z jedné části programu do druhé a obě části mohou přistupovat nejen k datovým atributům, ale přístupné jsou i operace. [21]

2.3 PHP framework

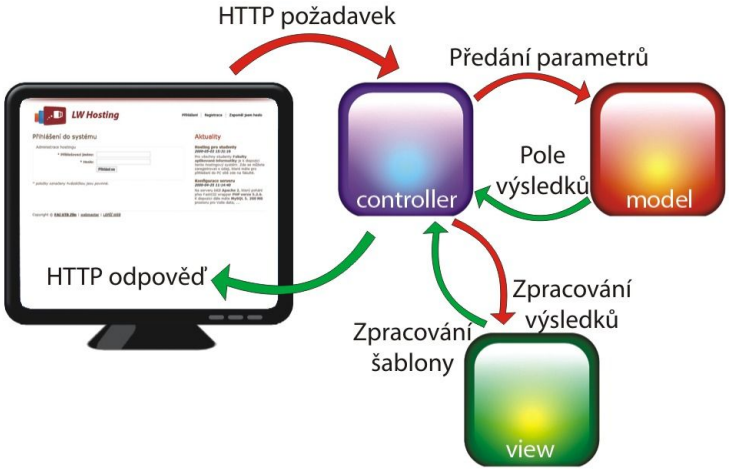
Při návrhu webové aplikace **LW Hosting** je velmi důležité dbát jak na kvalitu kódu, tak i na jeho přehlednost, srozumitelnost a samozřejmě i rozšiřitelnost. Předpokladem úspěšného systému je jeho budoucí vývoj a tomu by neměl bránit špatně položený základní kámen. Proto jednou z prvních podmínek při vývoji bylo použití objektově orientovaného programování a vhodná volba jazyka. Zde jsem logicky zvolil *skriptovací jazyk PHP* verze 5. Jednak jej zná a používá spousta programátorů a navíc splňuje i další požadavek, kterým je kvalitní framework. Aplikace musí být snadno rozšiřitelná, s přehlednou strukturou a jasně definovanými pravidly pro následný vývoj. V tomto nám framework pomůže a navíc přidá sadu tříd, které usnadní řešení častých a stále se opakujících operací.

Určit nejlepší framework je velmi složité. Existují velmi komplexní řešení, jakými je například *Zend Framework* [19] nebo *CakePHP* [5]. Tyto excelentní frameworky jsou sice velmi dobře řešeny, zdokumentovány, ale jsou to již obrovské balíky dat a z mého pohledu nejsou vhodnými kandidáty pro aplikaci **LW Hosting**. Při započítání tohoto kritéria mi zůstali dva zajímavé projekty. *CodeIgniter* [6] a *Nette Framework* [10] od českého autora. I když u prvního jmenovaného je lépe zpracována dokumentace, zvolil jsem právě dílo od českého autora, *Nette Framework*. Tento projekt se zdá být velmi nadějným a velmi rychle roste i počet jeho uživatelů. Díky tomu bude do budoucna snadné náš projekt rozšiřovat. Celkový vývoj tohoto frameworku jde v současné době velmi rychle dopředu, a to je další nesporný klad.

2.3.1 MVC (Model-View-Controller)

MVC je architektura, která rozděluje aplikaci do tří vrstev: datový model, řídící logiku a uživatelské rozhraní. *MVC* je často chápán jako návrhový vzor, nicméně se týká architektury aplikací mnohem více než klasický návrhový vzor. Přidá aplikaci na

přehlednosti a srozumitelnosti. Výhodou je, že modifikací kterékoliv z vrstev ovlivníme ty ostatní jen minimálně.



Obr 1: MVC

Datový model se zde stará pouze o komunikaci s databází, řídící logika přichystá data pro naplnění šablon, které jsou reprezentovány poslední vrstvou, která je uživatelské rozhraní. Díky šablonám můžeme velice snadno měnit vzhled aplikace, a to bez vlivu na funkční vlastnosti. Šablony se skládají převážně z *XHTML*, *CSS* a *JavaScriptu*.

3 BEZPEČNOST, ZRANITELNOST SYSTÉMU

Nejdůležitější na každém projektu je bezesporu jeho bezpečnost. A to jak bezpečnost při užívání, tak i zabezpečení a bezproblémový provoz. Jakákoliv zranitelnost může znamenat například pád celého systému a nebo dokonce ztrátu citlivých informací.

Je velmi důležité povolit tedy pouze ty služby a následně porty, které budou využívány a neotevírat zbytečně zadní vrátka. Mezi základní techniky jak ochránit systém patří aktualizace a dále i správně nakonfigurovaný firewall.

3.1 GNU/Linux

Systémy *GNU/Linux* jsou nejrozšířenější na serverech. Zde vyhrávají především svou výbornou stabilitou a podporou hardware. *GNU/Linux* systémy můžeme rozdělit na část jádra Linuxu, které je velmi dobře odladěno a stabilní. A dále na část aplikační, kde se nejčastěji vyskytují bezpečnostní problémy. Proto je velmi důležitá již zmíněná pravidelná aktualizace systému a zabezpečení klíčových služeb.

3.1.1 Zabezpečení služeb

U všech služeb, které jsou na serveru dostupné ze sítě internet je velmi důležité dbát na bezpečnost. Systém je totiž právě tak bezpečný, jak je bezpečný jeho nejslabší článek. Tohle pravidlo velmi často využívají útočníci a hledají povolené porty a nezabezpečené služby, přes které mohou ovládnout celý systém.

3.2 Apache

Vývoj *Apache* začal v roce 1993 v *NCSA* (National Center for Supercomputing Aplications) na Illinoiské univerzitě. Původní jméno projektu bylo *NCSA HTTPd*. První veřejná verze s označením 0.6.2 byla vydána v dubnu 1995. Následovalo kompletní přepsání kódu (Apache2 už neobsahuje nic z původního NCSA HTTPd) a založení *Apache Group*, která je dnes základem vývojářského týmu.

Webový server Apache patří mezi hojně využívané a stabilní řešení na webhostingovém poli. Základním pravidle při zabezpečování serveru je, že čím méně informací útočník ví, tím složitější je provedení následného útoku. Zde můžeme začít například s omezením chybových hlášení. To provedeme nastavení direktivy „*ServerTokens*“ na hodnotu „*Prod*“. Tím nastavíme identifikaci serveru v HTTP hlavičce na jednoduché „*Apache*“ místo

kompletního vypsání čísla verze a operačního systému a zakompilovaných modulů. Druhou důležitou direktivou při zabezpečování je „*ServerSignature*“, kterou musíme taky vypnout. Tím zamezíme vkládání podpisu na stránky generované serverem. [22]

3.2.1 PHP

Pokud přidáme podporu *skriptovacího jazyka PHP*, tak nastává několik bezpečnostních rizik, které musíme minimalizovat. Díky *PHP* může uživatel psát skripty, které například přistupují do systému souborů a odsud získat důležitá data. Mezi základní bezpečnostní opatření patří omezení direktivou *safe_mode* a nastavení *open_basedir*. Tyto dvě direktivy nám však nezajistí 100% zabezpečený webhosting. Především direktiva *safe_mode* velmi omezuje některé funkce, a to nám ani samotné *PHP* nezaručí, že veškerá rozšíření respektují tuto directivu.

3.2.2 mod_fcgid

Je jedno z řešení na zabezpečení webhostingu. Díky *fcgid* můžeme provozovat každý virtuální hosting pod vlastním uživatelským jménem, skupinou a vkládat individuální konfigurační soubor *php.ini*. Ideální je tedy spojení s *open_basedir* a *disable_function*. Zakážeme uživateli potencionálně nebezpečné funkce skriptovacího jazyka *PHP*, jako jsou například: *exec*, *shell_exec*, *passthru*, *system*, …

V této konfiguraci můžeme nechat direktivu *safe_mode = Off*. Neomezujeme tím uživatele ve využití funkcí, nejsou takové problémy při nahrávání souborů a především uživatelé nemůžou přistupovat a číst informace ze souborů, kterých nejsou vlastníky.

3.3 Fail2ban

Jedná se o velmi šikovný nástroj na zabezpečení serveru a ochranu proti nepovolenému přístupu. Tento program je napsán v Pythonu. Průběžně kontroluje logy v systému a vyhodnocuje možná rizika. Výhodou je, že se dá nakonfigurovat na téměř každou službu, kterou provozujeme. Pokud pro ni neexistují konfigurační soubory, můžeme si je vytvořit sami. Je to velmi snadné, protože stačí zadat regulární výraz, který poté Fail2ban bude vyhodnocovat.

Mezi výhody patří především jednoduchá konfigurace, rozšiřitelnost a nenáročný provoz. Program umí v případě nebezpečí např. zaslat informační mail, přidat pravidla do firewallu nebo i vypnout danou službu.

3.4 MySQL

MySQL je *databázový systém*, vytvořený švédskou firmou MySQL AB. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark.

MySQL je multiplatformní a můžeme jej tedy používat jak na různých operačních systémech, tak i platformách. Komunikace s ní probíhá – jak už název napovídá – pomocí *jazyka SQL*. Podobně jako u ostatních *SQL databází* se jedná o dialekt tohoto jazyka s některými rozšířeními.

Ihned po instalaci databázového serveru je velmi důležité jeho zabezpečení. Často se stane, že je server nainstalován a zprovozněn bez hesla uživatele *root*. Ale systém Debian nám nabízí u *MySQL serveru* i skripty, z nichž je pro nás důležitý *mysql_secure_installation*.

3.4.1 Tímto skriptem provedeme následující bezpečnostní opatření:

- Změna hesla uživatele „*root*“,
- vymazání uživatele „*guest*“,
- zakázání vzdáleného přihlášení uživatele „*root*“,
- vymazání databáze „*test*“, pokud existuje
- a provede se aktualizace oprávnění.

3.5 Shorewall

Linuxové systémy mají již ve svém jádře firewall, který je však potřeba správně nakonfigurovat. K tomu nám poslouží *Shorewall*, který obsahuje sadu skriptů usnadňující jeho konfiguraci. Předností je jeho přehlednost a rozdělení konfigurace do předem určených souborů. Využijeme jej pouze především na povolení portů, které budou využívat služby našeho serveru. Patří tam *SSH*, *FTP*, *HTTP*. Ostatní porty jsou zakázány. Protože konkrétně na port *SSH* bude mít přístup pouze administrátor systému, tak můžeme tento port ještě dále filtrovat a povolit přístup pouze z vybraných IP adres.

3.6 SSH

OpenSSH (*OpenBSD Secure Shell*) je soubor počítačových programů zajišťující šifrovaný přístup k shellu operačního systému přes počítačovou síť. Má dvě části: *SSH klienta* a *SSH server*. *SSH klient* kromě zpřístupnění *Shellu* umí také vytvářet *SSH tunely*.

Je velice důležité mít bezpečné heslo a nebo se přihlašovat pomocí certifikátu. Služba *SSH* je nejčastěji napadána roboty, kteří přes ni zkoušejí proniknout do systému.

3.6.1 Zabezpečení SSH

- Přihlašování pomocí certifikátu,
- zákaz vzdáleného přihlášení uživatele „*root*“,
- zákaz přihlašování se pomocí hesla.

Mezi další často využívané techniky patří změna portu, na kterém poslouchá SSH server. Tohle nám však v zabezpečení serveru moc nepomůže. Ideálním řešením je omezení přístupu na port SSH pouze z určitých IP adres a k tomu nám velmi dobře poslouží Shorewall.

3.7 Webová aplikace

Terčem útoků není samozřejmě jen operační systém a nainstalovaný software, ale i naše webová aplikace. Při vývoji takového systému, který bude nasazen ve specifickém prostředí vysoké školy je velmi důležité nepodcenit žádné bezpečnostní riziko a důkladně se věnovat samotnému návrhu aplikace.

3.7.1 Oddělení skriptů

Proto je systém rozdělen na dvě oddělené části. První jsou běžné *CSS styly*, obrázky a *index.php*, které jsou dostupné přímo z prohlížeče. Skripty, moduly a knihovny jsou načítány z chráněného adresáře pomocí již zmiňovaného souboru *index.php*. Díky tomu není možné přistupovat z webového prohlížeče přímo na jednotlivé soubory jádra hostingového systému. Tímto se eliminuje možnost podstrčení škodlivých dat.

3.8 Zálohování

I když se to na první pohled nemusí zdát, tak i zálohování velmi úzce souvisí se zabezpečením serveru. V předchozích odstavcích jsme si řekli, jak můžeme zabezpečit jednotlivé služby proti nástrahám z internetu, nekalým praktikám studentů a jejich chytře napsaným skriptům, ale stále tady zůstává velmi významné a podceňované riziko, kterým je selhání hardware. Proto je velmi důležité pravidelně zálohovat nejen uživatelské adresáře, ale i konfiguraci celého systému a databáze. V tomto nám výborně poslouží program *rsync*.

U linuxových systémů je k dispozici nepřeberné množství dalších programů s jejichž pomocí si můžeme vytvořit nástroje k zálohování. Volba však padla na *rsync*, který zajistí jak zabezpečený přenos dat pomocí protokolu *SSH*, ale i rozdílové zálohování. Rozdílového zálohování můžeme využívat velmi často, protože jsou přenášeny pouze pozměněná a nová data. Díky zkrácení intervalu záloh minimalizujeme riziko ztráty důležitých dat a samozřejmě i nároky na šířku přenosového pásma.

II. PRAKTICKÁ ČÁST

4 DOKUMENTACE SYSTÉMU

O projektu **LW Hosting** můžeme říci, že je multiplatformní. Pro svou bezproblémovou funkčnost potřebuje *Apache*, *PHP* verze 5 a databázový server *MySQL*. Ve svém návrhu počítám s použitím na platformě *GNU/Linux*, ale vzhledem k použitému software je možné aplikaci provozovat jak na systémech *MS Windows*, tak i *Max OS X*. Na všech platformách je totiž možné provozovat jak webový server *Apache*, tak i databázi *MySQL* a skriptovací jazyk *PHP*.

4.1 Základní informace

4.1.1 Serverová část

Serverová část obsahuje kromě samotného operačního systému Debian GNU/Linux, software pro zabezpečení, webový a databázový server, ale i instalační skripty a další podpůrné skripty potřebné pro 100% funkčnost celého systému jako celku. Nechybí samozřejmě ani výchozí konfigurační soubory pro jednotlivé aplikace. Tyto konfigurační soubory jsou již upraveny tak, aby splňovaly specifické požadavky webhostingového serveru.

Ve službě *cron* je nakonfigurováno automatické spouštění těchto skriptů kontrolující a provádějící instalace a mazání virtuálních serverů jednotlivých uživatelů, zálohování systému a databází. Kontrola nových virtuálních serverů je prováděna periodicky každých 10 minut. Rozdílové zálohování probíhá každou hodinu. Tyto časy je samozřejmě možné upravit, ale dle požadavků správce systému.

Pro umístění skriptů je zvolen adresář */opt/lwhosting/*. Tento adresář není opět striktně vyžadován a je možné zvolit vlastní. Při jakékoliv změně stačí upravit konfigurační soubor, který obsahuje nastavení proměnných a ty se následně použijí i v dalších skriptech.

4.1.2 Webová aplikace

Pro umístění webové aplikace se předpokládá adresář */var/www/lwhosting/* a uživatelské adresáře jsou umístěny v */var/www/virtual/*. Názvy jednotlivých podadresářů se volí dle registrace v systému a výběru subdomény. Například uživatel „*t_zimacek*“ si může volit z následujících subdomén – t-zimacek, tzimacek, zimacek, tomas-zimacek.

Zvolme si tedy subdoménu *t-zimacek*. Naše adresa bude *http://t-zimacek.host.utb.cz* a adresář pro umístění prezentace se bude nacházet zde: */var/www/virtual/t-zimacek/www/*.

Zde vidíme, že v našem domovském adresáři je ještě podadresář www, který již bude obsahovat naši internetovou prezentaci, kterou zde chceme provozovat. Ale v našem domovském adresáři nalezneme i další podadresáře:

- */backup* – zálohy naší prezentace
- */logs* – logy generované webovým serverem
- */tmp* – místo pro dočasné soubory
- */www* – adresář dostupný z internetového prohlížeče

Volba této struktury má výhodu v tom, že skripty uložené mimo adresář */www* nejsou přímo dostupné z internetového prohlížeče, ale pomocí internetové aplikace k nim můžeme přistupovat a spouštět je. Navíc zde můžeme uživateli přidat další adresáře, které budou obsahovat další důležité informace. Této bezpečnostní vlastnosti je využito i v naší aplikaci.

Struktura **LW Hosting**:

- */app* – hlavni zdrojové kódy aplikace
  - */models*
  - */presenters*
  - */templates*
- */backups* – záloha systému
- */libs* – framework a další knihovny
  - */dibi*
  - */nette*
  - */swift*
- */logs*
- */tmp*
- */www* – adresář dostupný z internetového prohlížeče

4.1.3 Připojení k LDAP serveru

K adresářovému serveru se realizuje připojení přímo pomocí funkcí obsažených ve skriptovacím jazyku PHP. Zde jsou uchovávány informace o uživatelích a toho využijeme

pro naši aplikaci. Každý uživatel zde uložený, který má platné přístupové údaje si tak může vytvořit účet a hostingový prostor na **LW Hosting** serveru.

Celý proces ověřování je velice jednoduchý. **LW Hosting** se připojí k *LDAP serveru*, který je dostupný na adrese *ldap://ldap.utb.cz*, zašle údaje zadané uživatelem a čeká, zda se vrátí kladná odpověď. V případě platných údajů požádá aplikace o bližší informace. Ty jsou následně použity při registraci. Nemusíme tedy vyplňovat své jméno a příjmení, to nám poskytne adresářový server.

4.2 Instalace a konfigurace

Celá aplikace se skripty je obsažena v jednom archivu. Po nainstalování nového Debian 5 serveru a rozbalení archivu s aplikací již stačí jen spustit instalační skript.

4.2.1 Příprava souborů, instalační skript

Archiv rozbalte do adresáře */opt/lwhosting/* a spusťte instalační skript: *sh setup.sh*:

- Vyplňte základní informace, jakými jsou jméno, příjmení, e-mail, heslo pro administrátorský přístup do aplikace a heslo pro MySQL databázi.
- Po zadání údajů následuje instalace potřebných balíčků,
- přidání nového uživatele do systému,
- tvorba adresářové struktury,
- přidělení práv pro adresáře,
- přidání modulů do webového serveru Apache,
- tvorba uživatele a databáze hostingového programu **LW Hosting**,
- import tabulek a naplnění základními daty,
- zaslání informačního e-mailu administrátorovi serveru.

Nyní máme nastaveny oprávnění, adresářovou strukturu a přístupy do aplikace. Systém je nainstalován a obsahuje všechen potřebný software. Proto můžeme nakopírovat konfigurační soubory jednotlivých služeb do adresáře */etc/* a zdrojové kódy aplikace do */var/www/lwhosting/*. Služby jsou zkonfigurovány, aplikace nainstalována a nyní je doporučen restart jednotlivých služeb a nebo v ideálním případě celého serveru.

4.2.2 Zabezpečení MySQL

Po restartu je doporučeno zabezpečení databázového serveru MySQL. Obsahuje totiž po instalaci i testovací tabulky a může mít přednastaveno prázdné heslo pro uživatele root.

V konzoli spusťme následující příkaz: *mysql_secure_installation*. Tímto spustíme průvodce, který nás provede nastavením a zabezpečením MySQL serveru. O zamezení vzdáleného přístupu k databázím se nám již postará firewall, který je zkonfigurován a nastaven pro povolení portů pouze u služeb *HTTP*, *FTP*, *SSH*.

4.2.3 Konfigurační soubor config.php

Aplikace LW Hosting obsahuje hlavní konfigurační soubor, ve kterém jsou definovány základní proměnné dostupné v celé aplikaci. Patří zde např. nastavení informací o databázi, adresa serveru, hash pro cryptování hesel u uživatelských databází a další důležité proměnné.

4.2.4 Nastavení automatického spouštění skriptů

K zautomatizování některých úkonů se v systémech GNU/Linux využívá služby *cron*. Zde přidáme skripty, které chceme pravidelně spouštět a poté se již nemusíme starat o jejich běh. Patří zde například kontrola nových virtuálních webhostingů a pravidelné zálohování.

Do souboru */etc/crontab* přidáme skripty které chceme pravidelně spouštět:

- 0 * * * * root sh /opt/lwhosting/backup.sh
- 0 1 * * 1 root sh /opt/lwhosting/backup_week.sh
- */4 * * * root sh/opt/lwhosting/hosting.sh

Přidáním těchto řádků řekneme službě *cron*, aby pravidelně spouštěla naše skripty. První bude spuštěn každou hodinu a jedná se o rozdílové zálohování konfiguračních souborů a obsahu uživatelských adresářů. Druhý provádí kompletní zálohu vždy jednou týdně. Výhodou tohoto řešení je uchování i smazaných dat až do jedné hodiny ranní v pondělí, kdy bude kompletně synchronizován obsah všech adresářů. Při hodinovém zálohování nedochází k odstraňování smazaných souborů a jsou stále uchovány na záložním serveru. Proto můžeme v případě nechtěně smazaného souboru najít ještě jeho kopii na záložním serveru, a to až do konce týdne, kdy budou adresáře synchronizovány. Týdenní zálohování můžeme například rozšířit i na měsíční. Poupravením skriptu dosáhneme toho, že se týdenní záloha

přesune do archivu. Tyto archivy můžeme uchovávat dle potřeby, ale je velmi důležité pamatovat na objemy uložených dat.

4.3 Změny v nastavení

I když jsou všechny použité programy již nastaveny a jejich konfigurační soubory přibaleny ke zdrojovým kódům aplikace **LW Hosting**, tak může nastat situace, kdy potřebujeme změnit chování některého z programů. Mezi nejčastěji prováděné úpravy patří především konfigurace firewallu a nebo změny v nastavení webového serveru.

Konfigurační soubory k jednotlivým programům naleznete v adresářo */etc/název-programu*.

4.3.1 Apache

Konfigurační soubory se nacházejí v adresáři */etc/apache2/*.

Hlavním konfiguračním souborem je zde *apache2.conf*. Zde je možné nastavit chování webového serveru, chybové stránky, znakové sady, … V nejnovější verzi jsou jsou z tohoto konfiguračního souboru přesunuty bezpečnostní direktivy, který se přestěhovaly do souboru *conf.d/security*.

Dále zde naleznete adresáře *mods-enabled* a *sites-enabled*. První obsahuje všechny dostupné moduly pro webový server a další definici jednotlivých hostingových účtů registrovaných uživatelů.

4.3.2 Fail2ban

Konfigurační soubory naleznete v adresáři: */etc/fail2ban/*.

Základní nastavení programu provedete v souboru *fail2ban.conf*. Ten obsahuje pouze základní nastavení logování a stylu výpisu. Pro nás nejdůležitější část software se konfiguruje v souboru *jail.conf*. Zde se spouští sledování jednotlivých služeb, zadávají se kritéria, dle kterých se vyhodnocuje útok na server, chování programu při útoku.

Fail2ban je velmi univerzální a můžeme si zde přidat i sledování vlastních služeb. Stačí vytvořit filter a zaregistrovat jej v souboru *jail.conf*. Filtry nalezneme v adresáři *filter.d/*. Základní jsou přidány již při instalaci programu a libovolné další si můžeme vytvořit.

4.3.3 PHP5

Vzhledem k tomu, že zde běží *PHP5* pod modulem *fcgid* webového serveru *Apache*, tak každý webhosting má svůj vlastní konfigurační soubor *php.ini*. Při běžném použití naleznete konfigurační soubor zde: */etc/php5/cgi/php.ini*. V našem případě jsou všechny konfigurační soubory rozděleny hezky podle adresářů s názvem hostingu a naleznete je ve složce: */var/www/fcgi/*. Výchozí soubor, který se kopíruje do každého virtuálního serveru je umístěn zde: */var/www/fcgi/master/php.ini*. Jeho změnou ovlivníte všechny nově registrované uživatele.

Mezi nejzajímavější direktivy patří:

- *disable_function* – omezíme funkce, které uživatel nemůže využívat ve svých skriptech. Doporučuje se zakázat především exec, passthru, system.
- *memory_limit* – nastavíme maximum paměti, které může využít PHP skript. V našem případě bude dostatečných 16MB.
- *post_max_size* – omezuje velikost všech dat poslaných z jednoho formuláře.
- *register_globals* – přijatá data od uživatele jsou automaticky převedena na globální proměnné. Při špatné implementaci představuje bezpečnostní riziko, proto je doporučeno ponechat funkci vypnutou.
- *safe_mode* – direktiva upravuje chování *PHP* především při práci se soubory. Ověřuje vlastníka souboru a na základě ověření povolí a nebo zakáže přístup. V našem případě může zůstat direktiva ve stavu „*Off*“ - vypnutá, protože díky modulu *fcgid* a *suexec* běží každý PHP skript pod vlastním uživatelským jménem, což převyšuje i funkci direktivy *safe_mode*.
- *upload_max_filesize* – maximální velikost souboru, který můžeme poslat pomocí formuláře.

4.3.4 Pure-FTPd

Konfigurace FTP serveru je rozdělena na menší části, a to přidává i na přehlednosti. Základní nastavení naleznete v: */etc/pure-ftpd/*. Zde jsou další podadresáře:

- *auth/* - nastavení způsobu přihlašování
- *conf/* - běžné nastavení FTP serveru
- *db/* - nastavení přístupu k MySQL databázi

Definice FTP účtů jednotlivých uživatelů je uložena v *MySQL* databázi. To přináší obrovskou výhodu a snadnou správu přímo z aplikace *LW Hosting*. V databázi jsou definovány i velikosti adresářů, souborů a omezení rychlosti připojení k FTP serveru.

4.3.5 Shorewall

Konfigurace je opět rozdělena do více souborů. Pro nás nejdůležitější bude rules. Zde nastavujeme jednotlivá pravidla firewallu. Adresář se soubory nalezneme v */etc/shorewall/*.

4.4 Monitoring provozu

K základnímu sledování přenesených dat nám poslouží nástroje pro konfiguraci firewallu. Přímo pomocí *iptables* nastavíme sledování jednotlivých portů a budeme měřit průtok dat v obou směrech.

4.4.1 Nastavení iptables

Do startovacího skriptu aplikace Shorewall přidáme pravidla, která chceme aplikovat vždy po spuštění serveru a nebo restartu firewallu.

- iptables -N LW_INPUT
- iptables -N LW_OUTPUP
- iptables -I INPUT -j LW_INPUT
- iptables -I OUTPUT -j LW_OUTPUT
- iptables -I LW_INPUT -p tcp --dport 80
- iptables -I LW_OUTPUT -p tcp --dport 80
- iptables -A LW_INPUT -j RETURN
- iptables -A LW_OUTPUT -j RETURN

Tímto nastavíme sledování provozu na HTTP portu 80.

4.4.2 Omezení počtu požadavků na uživatelský účet

Aby bylo možné kontrolovat a omezovat velmi vytížené uživatelské webhostingy, využijeme funkce, kterou nám nabízí modul fcgid, který řeší nejen zabezpečení jednotlivých účtů, ale i omezení počtu požadavků. Díky tomu můžeme nastavit hranici, která bude zaručovat kvalitní a rychlé odezvy pro všechny hostingové účty.

4.4.3 Maximální rychlost stahování z www stránek

Tento modul webového serveru se nám postará o omezení rychlosti při stahování větších souborů. Server je nastaven tak, aby při běžném prohlížení stránek uživatele nijak nelimitoval. Pokud si však budete chtít stáhnout soubor o velikosti větší než 20MB, bude jej možné stahovat maximálně rychlostí 50 kB/s. Tímto zamezíme využívání webhostingu jako download serveru a zároveň zajistíme pro všechny uživatele dostatečnou konektivitu.

Nastavení limitů je provedeno přímo při registraci uživatele. Tato konfigurace se ukládá do adresáře: */etc/apache2/sites-available/*.

Ukázka nastavní:

- *BandwidthModule On* - inicializujeme modul
- *ForceBandWidthModule On* - zapneme omezení
- *Bandwidth all 1024000* - maximální rychlost komunikace je omezena na 10MB/s
- *MinBandwidth all 50000* - minimální rychlost komunikace je 50kB/s
- *LargeFileLimit * 20000 50000* – soubory o velikosti větší jak 20MB jsou omezeny při stahování na rychlost 50kB/s.

4.4.4 Stav serveru

Dále je připraven skript na zasílání pravidelných reportů o stavu serveru do e-mailu jeho správci. V současné verzi jsou v zaslané zprávě informace o discích, velikost volného prostoru, vytížení serveru, odezva od předdefinovaného serveru a počet běžících úloh. Tyto reporty je doporučeno zasílat jednou týdně, abychom měli zběžný přehled o funkčnosti.

4.5 Webové rozhraní

Webové rozhraní označuji jako „*Click and go*“. Tento název vychází ze způsobu užívání, práce se systémem a jednoduchosti ovládání. Tomuto heslu se podřizuje i grafická podoba a styl prezentace.

4.5.1 Použité technologie při tvorbě, možnosti rozšíření

LW Hosting je postaven na *Nette Frameworku*, který je od českého autora Davida Grudla. Dále systém využívá *knihoven dibi* pro komunikaci s databází a *Swift Mailer* [17] pro zasílání zpráv.

Šablony a celkový koncept stránek se řídí striktní normou jazyka XHTML a dodržují doporučení pro přístupnost a použitelnost webu. Tímto se stává aplikace přístupná všem uživatelům a nezáleží na tom, zda přistupují na stránky pomocí prohlížeče a nebo mobilního telefonu, zda mají nějaký handicap a nebo zakázané některé technologie ve svém prohlížeči.

Dodržováním pravidel přístupného webu neklademe uživatelům zbytečné překážky v užívání. Stránky pracují korektně jak na novém systému s nejnovějším prohlížečem internetových stránek, tak i na mobilním telefonu bez podpory *JavaScriptu* a s malým displejem.

Systémem generovaný kód a CSS styly jsou *validní*. Je dbáno na velikost výsledného dokumentu a na rychlost práce se systémem. I proto je zde implementována technologie *AJAX*, díky které se práce stává pružnější a příjemnější.

Mezi určitý technologický pokrok u tohoto typu systému patří *generování čistých URL adres*. U běžných internetových stránek je tato funkce dnes již docela běžná. Podporuje ji každý modernější *redakční systém*. Ale u specifických aplikací není vývoj tak výrazný a je dbáno především na technicky dokonalé řešení. I proto je tato funkce implementována především díky konceptu systému a na podporu uživatelů. Přeci jen jen příjemnější zobrazit uživateli v adrese např.: *http://host.utb.cz/hosting/vytvorit-databazi/*. Z adresy velice snadno poznáme, že se jedná o modul „*Hosting*“ a prováděná funkce je taky pěkně čitelná.

4.5.2 Konfigurační soubor webové aplikace

Aplikace obsahuje hlavní konfigurační soubor, s jehož pomocí nastavíme a nakonfigurujeme celou aplikaci. Soubor naleznete v */app/config.php* a obsahuje 6 částí:

- Konfigurace MySQL databáze,
- nastavení připojení k LDAP serveru,
- konfigurace mail serveru,
- nastavení znakové sady,
- informace o serveru, adresa a adresářová struktura,
- název, copyright a verze systému.

4.5.3 Administrátorský přístup

Administrátor má k dispozici samozřejmě všechny sekce a moduly systému. Může vytvářet, editovat a mazat uživatele, nastavovat kvóty hostingového programu, spravovat aktuality. Systém může mít i více administrátorských účtů. Potom každý s tímto oprávněním může vykonávat stejné operace jako hlavní správce, který byl vytvořen při instalaci systému.

4.5.4 Moduly

Předností **LW Hostingu** je modularita. Jakákoliv chybějící funkce tak může být velice snadno vytvořena a implementována do systému. Tomu napomáhá i objektové programování, které zpřehlední celý kód.

4.5.5 Zasílání e-mailů

Tento informační kanál je využíván v aplikaci velice často. Uživatel je informován o registraci a je mu zasláno v informačním e-mailu i heslo. Při tvorbě databáze a nebo FTP účtu se taky automaticky zašle informace s vygenerovaným jménem a heslem. Výhodou tohoto řešení je uchování důležitých informací. Již nikdy při práci se systémem nemusíte hledat papír a poznamenávat si nové heslo k databázi a nebo přístup na FTP účet. Dnešní schránky mají kapacitu v řádech gigabajtů, a proto není problém mít zprávu uchovánu dlouhé roky.

4.6 Klíčové funkce systému

Základní třídou naší aplikace je *BasePresenter*. Tato třída je rodičem všech modulů systému a její funkce je především prezentační. To znamená, že nám chystá veškerá data, validuje vstupní informace a koordinuje práci šablon. Taky se s její pomocí přistupujeme k databázi a dále nám nabízí i velmi užitečné funkce.

4.6.1 beforeRender()

Nastavení filtrů pro práci se šablonami a aktualitami. Tato funkce je volána při každém generování šablony.

4.6.2 showNews()

Inicializace „*Aktualit*“ a vytvoření šablony pro snadné importování do template webu. Kontrola zda je uživatel přihlášen a podle toho i příprava dat k výběru z databáze.

4.6.3 createEmailTemplate()

Funkce slouží k tvorbě e-mailové šablony. Je využívána téměř v každém modulu našeho systému. Slouží jak k zasílání informačních e-mailů po registraci, tak i informace o databázi a FTP účtu. Předností funkce je celková práce s e-maily. Vytvoříme si HTML šablony e-mailů a poté jen předáme jednotlivé proměnné a zprávu můžeme zaslat. Tímto se velice usnadní tvorba hezkých HTML e-mailů, které ukazují, že se jedná o profesionální systém a je myšleno i na pohodlí a přístup k uživatelům.

4.6.4 mailSender(array $arrTo, $strSubject, $strMessage)

S pomocí předchozí funkce jsme vytvořili hezký e-mail a nyní nám zbývá jej poslat. K zasílání zpráv využíváme knihovny *swiftmailer* a tato funkce nám celý proces velice zjednoduší. Inicializuje knihovnu, předá jí důležité informace o SMTP serveru, kódování a typu zprávy a samozřejmě i již sestavenou zprávu ve formátu HTML.

4.6.5 genPasswd($strLen = 6)

Generování náhodných hesel. Abychom předešli případným problémů, řešíme tady i posloupnosti znaků, povolené znaky a samozřejmě i bezpečnost generovaného hesla. Díky tomu můžeme tuto funkci využít jak při nové registraci uživatele, tak i u databází a FTP účtů.

Eliminujeme zde i případné překlepy. Proto se ve vygenerovaném řetězci nevyskytují například znaky *0* (nula), *O* (velké písmeno „o“), *1* (jedna), *l* (velké „el“) a další, které můžou znepříjemnit přihlašování.

4.6.6 strongPasswd($strPasswd)

Kontrola, zda uživatel zadal dostatečně bezpečné heslo. Návratový kód může nabývat hodnot „*true*“ nebo „*false*“.

Bezpečné heslo musí splnit následující kritéria:

- Velikost 6 – 20 znaků,
- obsahuje minimálně jedno malé a jedno velké písmeno,
- obsahuje minimálně jednu číslici.

4.6.7 encryptText($text)

Zašifrování zadaného textu. Aplikace jej využívá k šifrování hesel u uživatelských databází.

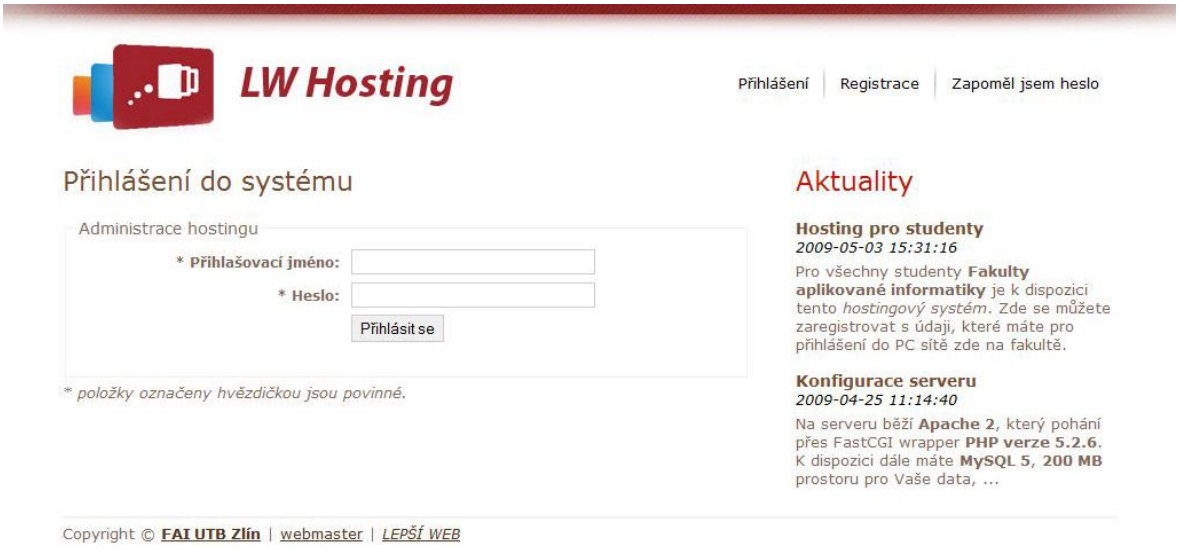
4.6.8 decryptText($text)

Dešifrování hesla. K šifrování i dešifrování slouží hash, který si můžete nastavit v souboru *config.php*. Bez tohoto kódu je rozluštění hesla téměř nemožné.

4.6.9 convertUrl($strUrl)

Při tvorbě subdomén a adresářů uživatelů musíme stále kontrolovat správnost zadaných vstupních informací. Tato funkce transformuje zakázané znaky na povolené, odstraní diakritiku a netisknutelné znaky. Výsledný řetězec je vhodný jak pro použití v URL adrese, tak i jako název adresáře, databáze nebo FTP účtu.

5 UŽIVATELSKÝ MANUÁL



Obr 2: Přihlašovací formulář

Úvodní obrazovka systému obsahuje přihlašovací formulář a sloupec s aktualitami. Dále je k dispozici menu, ve kterém jsou odkazy na registraci a zaslání nového hesla. Ovládání systému musí být jednoduché a intuitivní. Těmito pravidly se řídí i design a uspořádání ovládacích prvků.

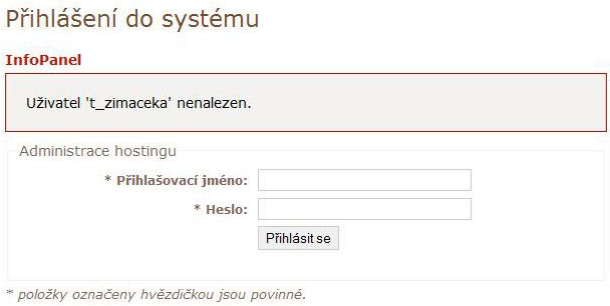
LW Hosting je rozdělen na část „*veřejnou*“ a „*administrační*“. V první části se nachází pouze přihlašovací formulář, registrace, zaslání zapomenutého hesla a veřejná kategorie aktualit. Další informace a nastavení jsou dostupné pouze registrovaným uživatelům.

5.1 Přednosti systému

Při návrhu struktury a designu jsem se řídil pravidlem, „*v jednoduchosti je krása*“. To samozřejmě přineslo i další výhody, mezi které patří snadná obsluha a rychlost celé aplikace. K rychlosti přispěla i technologie *AJAX*, která zajišťuje načítání částí stránek při přechodu mezi jednotlivými odkazy v sekci.

5.1.1 InfoPanel

Pro zvýšení uživatelského komfortu vznikla nová funkce s názvem „*InfoPanel*“. Protože každý chce být informován o změnách které provedl a ne jen při upozornění na chybu, tak Vás „*InfoPanel*“ provází celou aplikací. Informuje o změnách, aktualizacích a samozřejmě i o chybách.



Obr 3: InfoPanel

Vždy když se objeví upozornění, tak informace je podána nejen formou textovou, ale i grafickou. Běžné zprávy se zobrazují s černým rámečkem a varování nebo chyby mají rámeček červený.

5.2 Přihlášení, registrace, hesla

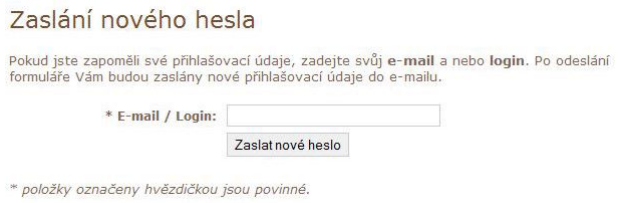
Obr 4: Ověření přístupu k síti UTB

Pro registraci nového uživatele je potřeba zadat přístupové údaje, které máte do počítačové sítě na univerzitě. Toto ověření provádí *LDAP server* a na základě jeho odpovědi je následně uživateli povolena nebo zamítnuta registrace. Tímto je zajištěn přístup pouze studentů nebo zaměstnanců *Univerzity Tomáše Bati ve Zlíně* a každý si může vytvořit maximálně 1 účet.

Obr 5: Registrace

Po ověření, že se jedná o člověka s přístupem do sítě univerzity je zpřístupněna registrace. Systém si automaticky načte jméno, příjmení a uživatel jen zadá svůj telefon a e-mail, který je vyžadován a na ten přijde uživateli vygenerované heslo. Díky tomuto systému je zajištěna správnost zadaných údajů. Pokud uživatel vyplní neplatný e-mail, nedostane se do systému a není mu umožněna ani nová registrace.

Ověřování oproti *LDAP serveru* je využito i pro přidělení přihlašovacího jména. To je shodné s přihlašovacím jménem do PC sítě UTB. Z bezpečnostních důvodů Vám však systém vygeneruje nové přihlašovací jméno, a to Vám bude ihned po registraci zasláno do zadaného e-mailu.



Obr 6: Zaslání nového hesla

V případě zapomenutého hesla si může každý zažádat o nové. To provedete z úvodní strany kliknutím na odkaz „*Zapoměl jsem heslo*“. Zadejte platné přihlašovací jméno nebo e-mail, systém vygeneruje nové heslo a zašle ho na e-mail zadaný při registraci.

5.3 Uživatelský účet

Obr 7: Můj účet

Po přihlášení je uživateli zobrazena základní obrazovka, která jej informuje o jeho zadaných údajích při registraci, je zde možnost změny přihlašovacího hesla do systému a samozřejmě i rychlý odkaz na tvorbu hostingu.

5.3.1 Změna hesla

Obr 8: Změna hesla

Každý uživatel si může v administraci nastavit své vlastní heslo. Jelikož celý systém klade důraz na bezpečnost, tak i nově vytvořené heslo musí splnit základní bezpečnostní kritéria, kterými jsou:

- Délka hesla minimálně 6 a maximálně 20 znaků,
- heslo musí obsahovat alespoň jednu číslici a kombinaci malých a velkých písmen.

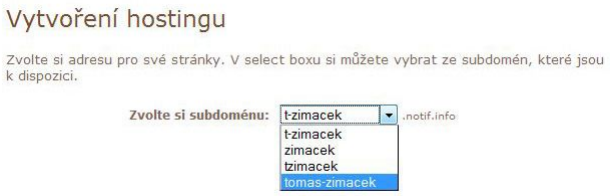
5.4 Konfigurace hostingu

Obr 9: Konfigurace Vašeho hostingu

Modul „*Hosting*“ patří mezi nejdůležitější části uživatelského účtu. Zde se zobrazují všechny důležité informace webhostingu. Vidíte zde URL adresu Vaší prezentace, limity, které máte nastaveny, databáze a informace o FTP účtu.

Celý systém ovládání se dá nazvat „*Click and go*“. Jediným kliknutím si zde velice rychle a snadno vytvoříte databázi, FTP přístup a dokonce i kompletní hosting.

5.4.1 Volba subdomény



Při výběru subdomény si můžete vybrat z několika variant, které pro Vás LW Hosting přichystal. Všechny varianty vycházejí z Vašeho jména a loginu do systému. Maximálně si můžete volit z 5 subdomén, pokud jsou ještě volné.

Jaké subdomény mi nabídne:

- Vaše přihlašovací jméno,
- první znak ze jména – příjmení,
- první znak ze jména a příjmení bez pomlčky,
- celé jméno – příjmení,
- jen samotné příjmení.

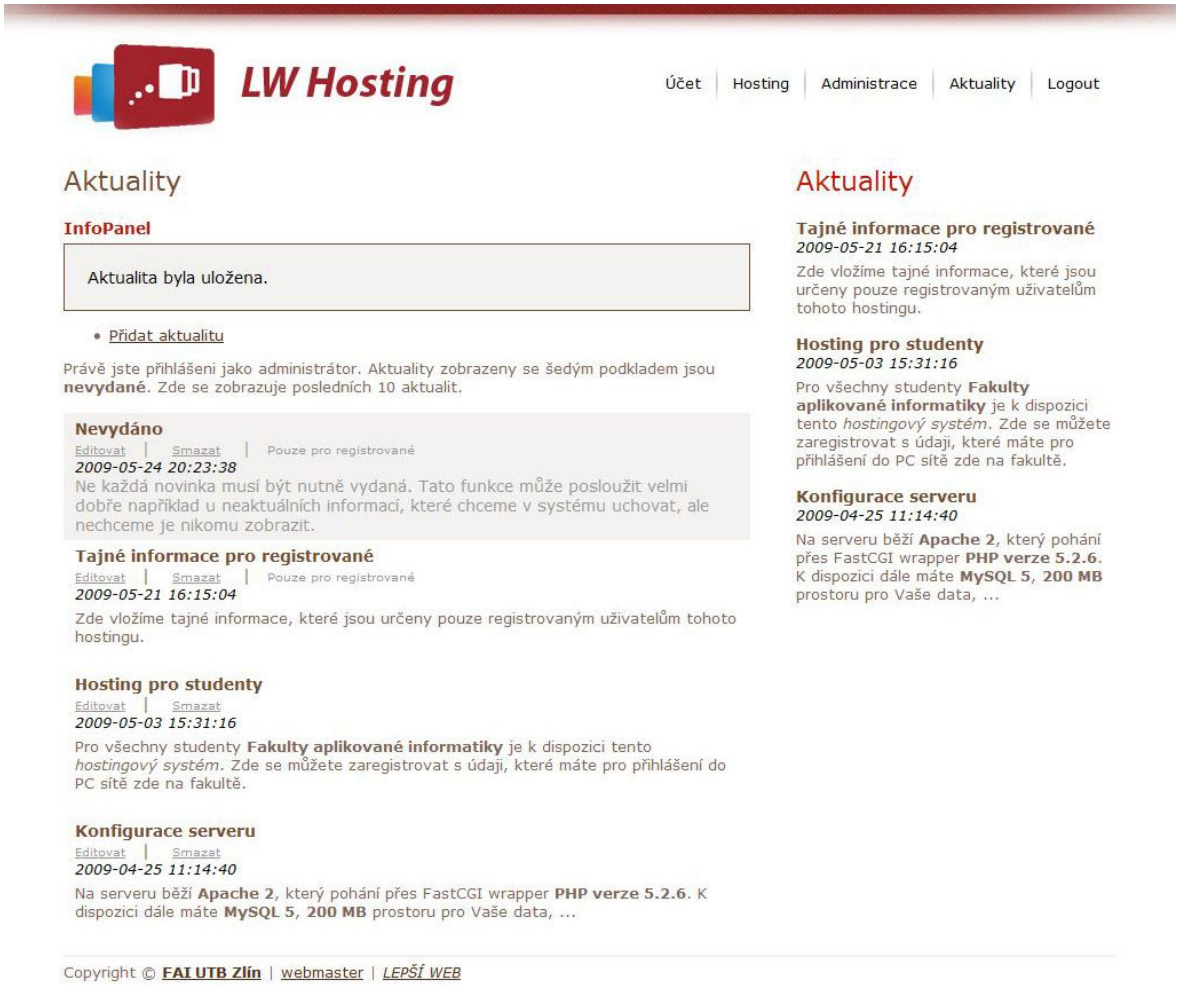
Tento systém tvorby subdomén vznikl především pro pohodlí studentů. Prvotním předpokladem totiž bylo, že se použije pouze přihlašovací jméno a nebudou žádné další možnosti. To by určitě neuvítali studenti, kteří mají v loginu i nějaká čísla a tím by se stala jejich adresa hůře zapamatovatelnou.

5.4.2 Správa databází a FTP přístup

Jak již bylo napsáno, tak databáze a FTP účty si můžete vytvořit jediným kliknutím. Systém automaticky vygeneruje dostatečně bezpečné heslo, které Vám zobrazí na obrazovce a zároveň i zašle do e-mailu.

Počet databází není dán pevně, ale je možné jej měnit. Toto nastavení má k dispozici pouze administrátor, který Vám může limity kdykoliv navýšit.

5.5 Aktuality

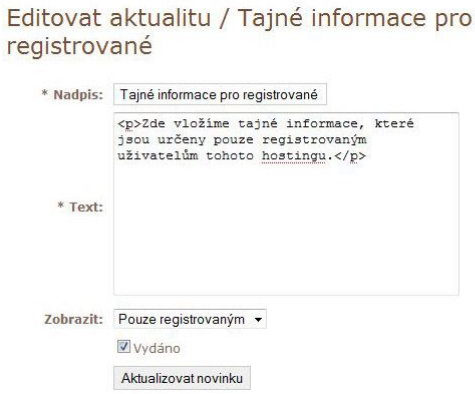


Obr 10: Aktuality

Aktuality jsou zde spíše jednoduchým informačním kanálem. Kratičké novinky využijeme především pro rychlé a snadné informování uživatelů systému. V pravém sloupci se zobrazí maximálně 3 aktuality. Ty jsou rozděleny na 2 hlavní sekce. První je zobrazena všem uživatelům a druhá slouží pouze k informování registrovaných členů.

Registrovaní uživatelé mají navíc přístup i do archivu aktualit. Zde si můžou zobrazit všechny vydané, které jsou zobrazeny po deseti na jedné straně.

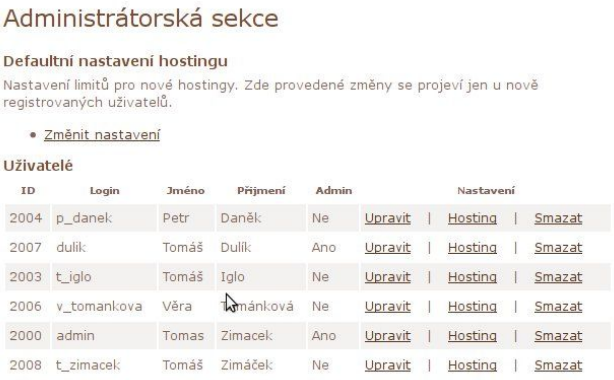
5.5.1 Administrace aktualit



Obr 11: Editace aktualit

Správce systému má samozřejmě v sekci aktualit mnohem více pravomocí. Může je vytvářet, mazat, editovat, zařadit do skupiny. Specialitou je zde skrytí aktuality. To využijeme především k zobrazení dočasných zpráv, které se často opakují, ale není důvod je zobrazovat všem uživatelům v archivu a nechceme je mazat a znovu vytvářet. Skrytí je pod položkou označenou „*Vydáno*“. Tímto aktualitu zařadíme mezi nevydané, běžný uživatel ji nevidí a administrátorovi se zobrazí podbarvena šedou barvou.

5.6 Administrátorská sekce



Obr 12: Administrátorská sekce

Sekce určená pouze správcům systému. Zde provedete nastavení uživatelských účtů a webhostingu. U každého uživatele jsou uvedeny tlačítka pro editaci jeho osobních údajů („*Upravit“*), změnu nastavení webhostingu („*Hosting*“) a smazání uživatelského a webhostingového účtu.

Jak již bylo několikrát napsáno, každý uživatel může mít specifické nastavení. Tato funkce je určena např. pro velké studentské projekty, které převyšují svou potřebou běžné limity.

5.6.1 Globální nastavení

Obr 13: Konfigurace hostingu

Tato sekce ovlivní pouze nově vytvořené hostingové účty. Není totiž žádoucí, aby nastavení ovlivnilo starší a nebo již změněné parametry u ostatních uživatelů. Administrátor může upravit nastavení každého účtu zvlášť.

6 TESTOVÁNÍ VÝKONU WEBOVÉHO SERVERU

Protože je sice velmi hezké mít zabezpečený webhosting, ale potřebujeme znát i jeho výkon. Proto jsem provedl testy serveru Apache při zapnutých ochranných opatřeních a bez nich. Testy byly provedeny ve virtuálním serveru. Virtualizaci zde zajišťuje nástroj Xen.

6.1 Konfigurace serveru

Pro testy jsem využil testovacího serveru. Jeho konfigurace je následující:

- Intel Xeon E3110, DualCore, 3.00GHz,
- 6 GB RAM,
- 2x 500 GB WD Raid Edition II

Virtuální server byl v následující konfiguraci:

- Plná virtualizace,
- 512 MB RAM,
- 20 GB HDD Raid 1

6.2 Výkonnostní testy

Pro vyhodnocení výkonu webového serveru je k dispozici prográmek přímo od tvůrců Apache. Nazývá se „ab“.

6.2.1 Jak probíhalo testování

Pro testování nám velmi dobře posloužil tento příkaz: „*ab -n 1000 -c10 http://notif.info/*“.

Tímto provedeme simulaci kdy na server jde celkem 1000 požadavků na internetovou stránku, a to až 10 v jeden okamžik. Program vyhodnocuje přenesená data, rychlost odezvy a samozřejmě i průměrný počet vyřízených požadavků za vteřinu.

6.2.2 Nastavena direktiva suexec

Tato direktiva nám slouží k nastavení uživatele a skupiny pro PHP skripty a jedná se o doporučené nastavení.

Počet vyřízených požadavků: **21,75 / s**

6.2.3 Vypnuta direktiva suexec

Při vypnuté direktivě *suexec* je nutné zabezpečit webový server zapnutím direktivy *safe_mode*, která nám alespoň částečně vynahradí doporučené řešení.

Počet vyřízených požadavků: **128,79 / s**

6.2.4 Výsledky testů

Tento jednoduchý test propustnosti nám jasně ukazuje, že řešení zabezpečení pomocí *suexec* je více jak 5x pomalejší než řešení pomocí direktivy *safe_mode*. Zde je velmi důležité si položit otázku, zda vyšší výkon stojí za ty omezení, které by nastaly při vypnutí *suexec*. Při dalším testování takto zabezpečeného serveru jsem zjistil, že není problém provozovat podobný virtuální server, kde jsou desítky webhostingových účtů. Na těch běží moderní redakční systémy a celkové počty požadavků za jeden den se blíží ke sta tisícům. Takto vytížený server s podobnou konfigurací je stále dostačující, jen je potřeba více paměti RAM. Rozložení požadavků samozřejmě není symetrické po celý den, a proto je tato konfigurace vhodná do 500 000 požadavků během 12 hodin.

Z toho soudím, že je tato konfigurace vhodná i pro školní hostingový server se studentskými projekty.

ZÁVĚR

Cílem diplomové práce bylo navržení hostingového systému, který by vyhovoval specifickým požadavkům vysoké školy a studentů. Speciální důraz je kladen především na zabezpečení a jednoduchý provoz s minimem zásahů administrátora.

Po prozkoumání stavu trhu jsem se rozhodl pro řešení postaveném na operačním systému *GNU/Linux*, skriptovacím jazyku *PHP* a databázovém serveru *MySQL*. Tato kombinace sebou přináší spoustu kladů, kterými jsou především nulové zřizovací náklady a díky instalačnímu skriptu **LW Hosting** i snadnou instalaci a konfiguraci základního systému. Díky přibaleným konfiguračním souborům jednotlivých služeb zvládne instalaci i uživatel se základními znalostmi práce v prostředí příkazové řádky linuxu.

Současná verze je tedy připravena k plnohodnotnému nasazení, ale zároveň se i předpokládá budoucí rozšiřování a zdokonalování systému dle potřeb jeho uživatelů. Mezi funkce, které již nyní můžete využít patří především propracovaný *registrační systém*, který ověřuje identitu u LDAP serveru. Dále modul pro práci s účty uživatelů, tvorbou hostingových účtů, databází a FTP přístupů. Na pozadí celého systému pracují skripty na zálohování a k úpravám konfiguračních souborů, jako jsou například subdomény webového serveru *Apache*. Součástí je i bezpečnostní systém, který vyhodnocuje hrozby přicházející z internetu a na jejich základě upravuje pravidla firewallu a tím chrání celý systém.

Je tedy položen základní kámen doufejme kvalitního hostingového systému pro vysoké školy. Nyní je velmi důležitá zpětná vazba od uživatelů systému a jeho následný rozvoj.

CONCLUSION

The purpose of this thesis was proposition of hosting system, which would correspond to specific requirements of University and its students.

Particular emphasis is placed on security and unsophisticated maintanance with minimum administrator intervention. After extensive market research I heve decided for solution build on operating systém *GNU/Linux*, scripting language *PHP* and database server *MySQL*. This combination brings plenty of positives whose are zero establihment costs and easy instalation and configuration of primary systém due to the instalation script **LW Hosting**.

Due to the enclosed configuration files of individual services, instalation could be accomplished by user with basic knowleadge of linux command line systém environment.

Current version is prepared for full-featured start-up. This version is supposed to perform future enlargement and systém development on users requests.

Among many functions, which could be already fully used belongs registration systém that verifies identuty of LDAP server. Furthermore, modul for accomplishment of tasks with user accounts, generation of hosting accounts, databasis and FTP acceses.

On the background of whole systém operate scripts for backup and for adjustments of configuration files, such as sub domains of web server Apache. Part of the backgorund is the security systém that evaluates threats from the internet. Based on these threats, system modifies firewall configuration what efends whole system.

This hosting system could be perceived as milestone of superior systém for universities. There is important feedback of systém users for now for subsequent development of the whole project.

SEZNAM POUŽITÉ LITERATURY

[1] B. Hatch, J.Lee, G. Kurtz: *Hacking bez tajemství*
Computer Press 2003, ISBN: 80-7226-869-4

[2] J. Zeldman: *Tvorba webů podle standardů*
Computer Press 2004, ISBN: 80-251-0347-1

[3] M. Kysela a kolektiv: *333 tipů a triků pro Linux*
Computer Press 2007, ISBN: 80-722-6866-X

[4] The Apache Software Foundation [online]. Dostupný z www:
http://www.apache.org/

[5] CakePHP [online]. Dostupný z www:
http://cakephp.org/

[6] CodeIgniter [online]. Dostupný z www:
http://www.codeigniter.com/

[7] Debian GNU/Linux [online]. Dostupný z www:
http://www.debian.org/

[8] Fail2ban [online]. Dostupný z www:
http://www.fail2ban.org/wiki/index.php/Main_Page

[9] MySQL [online]. Dostupný z www:
http://www.mysql.com/

[10] Nette Framework [online]. Dostupný z www:
http://nettephp.com/cs/

[11] OpenSSH [online]. Dostupný z www:
http://www.openssh.com/

[12] PHP [online]. Dostupný z www:
http://php.net/

[13] Postfix [online]. Dostupný z www:
http://www.postfix.org/

[14] Pure-FTPd [online]. Dostupný z www:
http://www.pureftpd.org/project/pure-ftpd

[15] rsync [online]. Dostupný z www:
http://www.samba.org/rsync/

[16] Shorewall [online]. Dostupný z www:
http://shorewall.net/

[17] Swift Mailer [online]. Dostupný z www:
http://swiftmailer.org/

[18] Ubuntu [online]. Dostupný z www:
http://www.ubuntu.cz/

[19] Zend Frameword [online]. Dostupný z www:
http://framework.zend.com/

[20] LDAP [online]. Dostupný z www:
http://cs.wikipedia.org/wiki/LDAP

[21] OOP [online]. Dostupný z www:
http://cs.wikipedia.org/wiki/Objektov%C4%9B_orientovan%C3%A9_programov%C3%A1n%C3%AD

[22] Zabezpečení serveru Apache a PHP [online]. Dostupný z www:
http://www.security-portal.cz/clanky/zabezpe%C4%8Den%C3%AD-serveru-apache-php

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ab	Apache Benchmark, utilita na testování výkonu webového serveru
AJAX	Asynchronous JavaScript and XML, technologie vývoje interaktivních webových aplikací
Apache	Multiplatformní webový server
CSS	Cascading Style Sheets, kaskádové styly
CMS	Systém pro správu obsahu
FTP	File Transfer Protocol, protokol pro přenos souborů
GNU/Linux	Svobodný operační systém
HTTP	Hypertext Transfer Protocol, internetový protokol původně určený pro výmenu HTML dokumentů
HTTPS	Šifrovaná verze HTTP protokolu
Java	Objektově orientovaný programovací jazyk
LDAP	Lightweight Directory Access Protocol, protokol pro přístup k adresářovému serveru
OOP	Objektově orientované programování
PC	Personal computer, osobní počítač
PHP	Hypertext Preprocessor, skriptovací jazyk
Python	Objektově orientovaný programovací jazyk
SMTP	Simple Mail Transfer Protocol, protokol pro přenos elektronické pošty
SSH	Secure Shell, zabezpečený komunikační protokol
UNIX	Operační systémy
URL	Uniform Resource Locator, schéma internetové adresy

SEZNAM OBRÁZKŮ

SEZNAM PŘÍLOH

PŘÍLOHA P I: AUTENTIZACE UŽIVATELE

```
class Authorization implements IAuthenticator {

	protected $db;

/* Nastaveni komunikace s DB */

function __construct() {

	$this->db = new DibiConnection(array(
	'driver' => MYSQL_DRIVER,
	'host' => MYSQL_HOST,
	'user' => MYSQL_USER,
	'password' => MYSQL_PASSWORD,
	'database' => MYSQL_DATABASE,
	'charset' => MYSQL_CHARSET,
	'profiler' => TRUE));

}

function authenticate(array $credentials) {

	// jméno, heslo i role mohou být získány třeba z databáze
	$username = strtolower($credentials[self::USERNAME]);
	$password = md5($credentials[self::PASSWORD]);


	$row = $this->db->select('*')->from('users')->where('login=%s', $username)->fetch();
	$roles = NULL;

	if (!$row) {
		throw new AuthenticationException("Uživatel '$username' nenalezen.",
		self::IDENTITY_NOT_FOUND);
	}

	if ($row->pass !== $password) {
		throw new AuthenticationException("Zadali jste chybné heslo.",
		self::INVALID_CREDENTIAL);
	}

	if ($row->admin) $roles = array('admin');
	unset($row->pass);
	return new Identity($row->login, $roles, $row);

}

}
```

PŘÍLOHA P II: KONFIGURAČNÍ SOUBOR

```
// MySQL

define(MYSQL_DRIVER, 'mysqli');

define(MYSQL_HOST, 'localhost');

define(MYSQL_USER, 'lwhosting');

define(MYSQL_PASSWORD, 'sjdhsaY323Huda');

define(MYSQL_DATABASE, 'lwhosting');

define(MYSQL_CHARSET, 'utf8');


// LDAP connect

define(LDAP_SERVER, 'ldap://ldap.utb.cz/');

define(LDAP_O, 'utb');


define(SERVER_CHARSET, 'utf-8');

define(SERVER_URL, 'host.utb.cz');

define(HOSTING_DIR, '/var/www/virtual');


define(LW_NAME, 'LW Hosting');

define(LW_VERSION, '0.1');

define(LW_COPYRIGHT, 'Univerzita Tomáše Bati ve Zlíně (info@utb.cz)');

define(LW_KEY, 'jWfsd2aqu2311i3ygd0');


// Mail info

define(MAIL_FROM, 'info@host.utb.cz');

define(MAIL_HOST, '127.0.0.1');

define(MAIL_PORT, '25');
```

PŘÍLOHA P III: TABULKY DATABÁZE – STRUKTURY

ftpd

Sloupec	Typ	Nulový	Výchozí
User	varchar(16)	Ne	
status	enum(„0“, “1“)	Ne	0
Password	varchar(64)	Ne	
Uid	varchar(11)	Ne	-1
Gid	varchar(11)	Ne	-1
Dir	varchar(128)	Ne	
ULBandwidth	smallint(5)	Ne	0
DLBandwidth	smallint(5)	Ne	0
comment	tinytext	Ne	
ipaccess	varchar(15)	Ne	*
QuotaSize	varchar(5)	Ne	0
QuotaFiles	int(11)	Ne	0

hosting

Sloupec	Typ	Nulový	Výchozí
id	smallint(4)	Ne	
uid	smallint(4)	Ne	
dir	varchar(32)	Ne	
mysql	smallint(3)	Ano	
quotasize	int(11)	Ano	
quotafiles	int(11)	Ano	
ulbandwidth	int(11)	Ano	
dlbandwidth	int(11)	Ano	

kontext

Sloupec	Typ	Nulový	Výchozí
id	smallint(4)	Ne	
name	varchar(20)	Ne	

mysql

Sloupec	Typ	Nulový	Výchozí
id	smallint(4)	Ne	
uid	smallint(4)	Ne	
dbname	varchar(16)	Ne	
pass	varchar(60)	Ne	

news

Sloupec	Typ	Nulový	Výchozí
id	smallint(4)	Ne	
uid	smallint(4)	Ne	
date	timastamp	Ne	CURRENT_TIMESTAMP
title	varchar(100)	Ne	
text	text	Ne	
status	enum(„0“, „1“)	Ne	1
register	enum(„0“, „1“)	Ne	0

programs

Sloupec	Typ	Nulový	Výchozí
id	smallint(4)	Ne	

mysql	smallint(3)	Ano	
quotasize	int(11)	Ano	
quotafiles	int(11)	Ano	
ulbandwidth	int(11)	Ano	
dlbandwidth	int(11)	Ano	

users

Sloupec	Typ	Nulový	Výchozí
uid	smallint(4)	Ne	
login	varchar(30)	Ne	
pass	varchar(32)	Ne	
kontext	smallint(4)	Ano	NULL
firstname	varchar(30)	Ne	
lastname	varchar(30)	Ne	
tel	varchar(20)	Ano	NULL
email	varchar(50)	Ne	
admin	tinyint(1)	Ne	0

PŘÍLOHA P IV: UKÁZKA INSTALAČNÍHO SKRIPTU

```
#!/bin/sh

USER=lwhosting
echo -n "Vase jmeno: "
read FNAME
echo -n "Vase prijmeni: "
read LNAME
echo -n "Vas e-mail: "
read EMAIL
echo -n "Zadejte heslo uzivatele admin: "
read ADMIN
echo -n "Zadejte heslo pro MySQL uzivatele lwhosting: "
read PASS
echo ""

# instalace nezbytných balíčků

aptitude install $(cat debian_lenny.install)

# vytvoření hlavní skupiny a uživatele

groupadd -g 2000 lw2000
useradd -u 2000 -s /bin/false -d /var/www/lwhosting -g lw2000 lw2000

# pridání modulů do Apache

a2enmod suexec
a2enmod rewrite
a2enmod include
a2enmod fcgid
a2enmod bw
/etc/init.d/apache2 restart

# přidání nového uživatele do DB

echo "GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP ON $USER.* TO '$USER'@'localhost'
IDENTIFIED BY '$PASS';" > db_user.mysql
echo "GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP ON $USER.* TO
'$USER'@'localhost.localdomain' IDENTIFIED BY '$PASS';" >> db_user.mysql
echo "FLUSH PRIVILEGES;" >> db_user.mysql
echo "USE lwhosting;" >> db_user.mysql
echo "INSERT INTO users ( uid , login , pass , firstname , lastname , tel , email , admin ,
domain ) VALUES ( '2000', 'admin', MD5( '$ADMIN' ) , '$FNAME', '$LNAME', NULL , '$EMAIL',
'1', '' );" >> db_user.mysql
echo "INSERT INTO kontext ( name ) VALUES ( 'fai-st' );" >> db_user.mysql
echo "INSERT INTO kontext ( name ) VALUES ( 'fai' );" >> db_user.mysql
echo "INSERT INTO kontext ( name ) VALUES ( 'fame-st' );" >> db_user.mysql
echo "INSERT INTO kontext ( name ) VALUES ( 'fame' );" >> db_user.mysql
echo "INSERT INTO programs ( ulbandwidth , dlbandwidth , quotasize , quotafiles , mysql )
VALUES ( '300' , '200' , '200' , '50' , '1' );" >> db_user.mysql
mysql -u root -p < db_user.mysql

# Kompletní instalační skript naleznete v adresáři „lwhosting“ po rozbalení archivu.
```