Virtuální tunel pro elektronické informační zdroje

Virtual tunnel for electronic information sources

Jan Marek

Bakalářská 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 BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan MAREK**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Virtuální tunel pro elektronické informační zdroje**

Zásady pro vypracování:

1. **Vytvořte dns alias pro každý elektronický informační zdroj (EIZ).**
2. **Zprovozněte webový server apache s požadovanými moduly (mod_auth, mod_proxy...).**
3. **Proveďte ověřování uživatelů přes server Apache.**
4. **Zprovozněte a nakonfigurujte proxy server, pro přístup do EIZ.**

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **STREBE, Matthew, PERKINS, Charles. Firewally a proxy-servery : Praktický průvodce. Libor Pácl; Lenka Hendrychová, Jakub Mikuláštík. 1. vyd. Brno : Vydavatelství a nakladatelství Computer Press, 2003. 442 s. ISBN 80-7226-983-6.**
2. **TEIXERA, Steve, PACHECO, Xavier. Borland Delphi : průvodce vývojáře - kniha 5-6. [s.l.] : Knihy iDNES, 2002. 512 s. ISBN 80-86593-10-X.**
3. **KABIR, Mohammed J. Apache Server 2 : kompletní příručka administrátora. Brno : Computer Press, a.s., 2004. 724 s. ISBN 80-251-0319-6.**
4. **DOSTÁLEK, Libor, KABELOVÁ, Alena. Velký průvodce protokoly TCP/IP a systémem DNS. 1. vyd. Praha : Computer Press, c1999. 418 s. , 1 CD-ROM. ISBN 80-7226-193-2.**

Vedoucí bakalářské práce: **RNDr. Ing. Miloš Krčmář**
Ústav aplikované informatiky

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

Termín odevzdání bakalářské práce: **1. června 2009**

Ve Zlíně dne 13. února 2009

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky

L.S.

prof. Ing. Vladimír Vašek, CSc.
děkan

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

ABSTRAKT

Systém umožňuje klientovi připojit se svým účtem z libovolného počítače přes webové rozhranní k poskytovaným elektronickým informačním zdrojům (EIZ). Tyto EIZ jsou registrovány a vázány na pevnou IP adresu poskytovatele této služby. Systém mezi klientem a poskytovatelem vytvoří virtuální tunel, takže komunikuje skrze tuto pevnou IP a tím jsou mu EIZ přístupné.

Klíčová slova: address record (DNS), Apache, mod_proxy, Proxomitron (proxy)

ABSTRACT

The system enables the client to connect to your account from any computer via a web interface provided by the electronic information sources (EIS). These EIS are registered and linked to a fixed IP address of the service provider. System between the client and the provider will create a virtual tunnel, so that communicates through a fixed IP and that are accessible to him EIS.

Keywords: address record (DNS), Apache, mod_proxy, Proxomitron (proxy)

Motto: „Informace bez následné aplikace je bezcenná.“

Chtěl bych poděkovat hlavně své ženě Veronice, za její trpělivost během mého studia. Čas, který jsem strávil studiem ti snad jednou vynahradím. Lásko, miluji tě.

Dále bych chtěl poděkovat svému zaměstnavateli, Krajské knihovně Františka Bartoše, příspěvkové organizaci ve Zlíně, za to že jsem si mohl své teoretické znalosti odzkoušet i v praxi a bakalářskou práci tak realizovat a její výsledek poskytnout široké veřejnosti.

Dík patří i mému dobrému kamarádovi a kolegovi Františku Janošovi za jeho pomoc při tvorbě programu Expin.

A nakonec panu RNDr. Krčmářovi, že mě po celou dobu tvorby bakalářské práce vedl a podával mi potřebné informace.

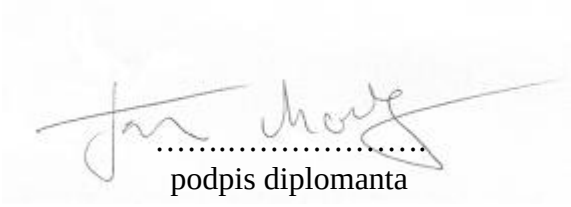
Prohlašuji, že

- beru na vědomí, že odevzdáním 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 bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk 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 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 – 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 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 bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem 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 bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně 20.5.2009



…….
podpis diplomanta

OBSAH

ÚVOD

Práce se zabývá tvorbou přístupu k elektronickým informačním zdrojům (dále jen EIZ). Tyto EIZ jsou v internetu poskytovány komerčně a nepatří zrovna k levným zdrojům informací. Roční platby za tyto EIZ dosahují desítek až stovek tisíc Kč. Většina těchto EIZ je konfigurována a poskytována vůči jediné pevné internetové adrese (dále jen IP). Jedná se o poměrně vhodně zvolený bezpečnostní prvek. Nicméně většina těchto poskytovatelů má ve svých licenčních podmínkách zakotvenou klauzuli, která povoluje nakupovateli tyto informační zdroje dále poskytovat svým uživatelům (zaměstnancům, čtenářům). Přístup je však podmíněn autentifikací. Je tedy tyto EIZ možno poskytnout nejen v organizaci s touto IP, ale i mimo ni. A to skrze systém, který příslušného uživatele osloví (autentizuje), autorizuje a přístup do EIZ mu skrze svou IP umožní.

Cílem této bakalářské práce je zprovoznit DNS aliasy na vybrané EIZ, autorizační systém (Apache a proxy server) a možnost automatické aktualizace logovacích údajů.

I. TEORETICKÁ ČÁST

1 DNS SYSTÉM

DNS (Domain Name System) je hierarchický systém doménových jmen, který je realizován servery DNS a protokolem stejného jména, kterým si vyměňují informace. Jeho hlavním úkolem a příčinou vzniku jsou vzájemné převody doménových jmen a IP adres uzlů sítě.[5]

1.1 Funkce DNS systému

Všechny aplikace, které zajišťují komunikaci mezi počítači používají k identifikaci komunikujících uzlů IP-adresu. Pro člověka jako uživatele jsou však IP-adresy název síťového rozhraní. Pro každou IP-adresu máme zavedeno jméno síťového rozhraní (počítače), přesněji řečeno doménové jméno. Toto doménové jméno můžeme používat ve všech příkazech, kde je možné použít IP adresu. Výjimkou, kdy se musí použít IP-adresa, je identifikace samotného name serveru.

Jedna IP-adresa může mít přiřazeno i několik doménových jmen.[1]

Prostor doménových jmen tvoří strom. Každý uzel tohoto stromu obsahuje informace o části jména (doméně), které je mu přiděleno a odkazy na své podřízené domény. Kořenem stromu je tzv. kořenová doména, která se zapisuje jako samotná tečka. Pod ní se v hierarchii nacházejí tzv. domény nejvyšší úrovně (Top-Level Domain, TLD). Ty jsou buď tematické (**com** pro komerci, **edu** pro vzdělávací instituce atd.) nebo státní (**cz** pro Českou republiku, **sk** pro Slovensko, **jo** pro Jordánsko atd.).

Strom lze administrativně rozdělit do zón, které spravují jednotliví správci (organizace nebo i soukromé osoby), přičemž taková zóna obsahuje autoritativní informace o spravovaných doménách. Tyto informace jsou poskytovány autoritativním DNS serverem.

Výhoda tohoto uspořádání spočívá v možnosti zónu rozdělit a správu její části svěřit někomu dalšímu. Nově vzniklá zóna se tak stane autoritativní pro přidělený jmenný prostor. Právě možnost delegování pravomocí a distribuovaná správa tvoří klíčové vlastnosti DNS a jsou velmi podstatné pro jeho úspěch. Ve vyšších patrech doménové hierarchie platí, že zóna typicky obsahuje jednu doménu. Koncové zóny přidělené organizacím připojeným k Internetu pak někdy obsahují několik domén – například

doména kdesi.cz a její poddomény vyroba.kdesi.cz, marketing.kdesi.cz a obchod.kdesi.cz mohou být obsaženy v jedné zóně a obhospodařovány stejným serverem.

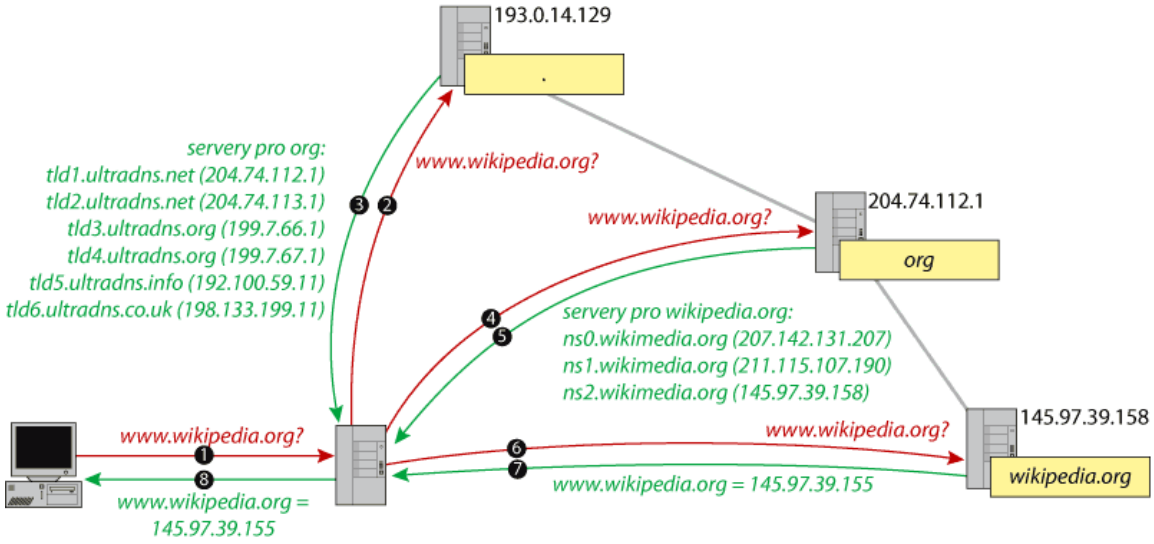
Celé jméno se skládá z několika částí oddělených tečkami. Na jeho konci se nacházejí domény nejobecnější, směrem doleva se postupně konkretizuje.

- část nejvíce vpravo je doména nejvyšší úrovně, např. wikipedia.org má TLD org.
- jednotlivé části (subdomény) mohou mít až 63 znaků a skládat se mohou až do celkové délky doménového jména 255 znaků. Doména může mít až 127 úrovní. Bohužel některé implementace jsou omezeny více.

1.2 Řešení DNS dotazu

Pokud počítač hledá určitou informaci v DNS (např. IP adresu k danému jménu), obrátí se s dotazem na tento lokální server. Každý DNS server má ve své konfiguraci uvedeny IP adresy kořenových serverů (autoritativních serverů pro kořenovou doménu). Obrátí se tedy s dotazem na některý z nich.

Kořenové servery mají autoritativní informace o kořenové doméně. Konkrétně znají všechny existující domény nejvyšší úrovně a jejich autoritativní servery. Dotaz je tedy následně směrován na některý z autoritativních serverů domény nejvyšší úrovně, v níž se nachází cílové jméno. Ten je opět schopen poskytnout informace o své doméně a posunout řešení o jedno patro dolů v doménovém stromě. Tímto způsobem řešení postupuje po jednotlivých patrech doménové hierarchie směrem k cíli, až se dostane k serveru autoritativnímu pro hledané jméno, který pošle definitivní odpověď.



Obr. 1. Postup hledání v DNS př. www.wikipedia.org. [5]

1.3 Adress record

Záznam typu **A** (Address) přiřazují doménovým jménům počítačů IP-adresy. Za IP-adresou nesmí být tečka. [1]

Například když jménu *cosi.kdesi.cz* náleží IP adresa 1.2.3.4, bude zónový soubor pro doménu *kdesi.cz* obsahovat záznam:

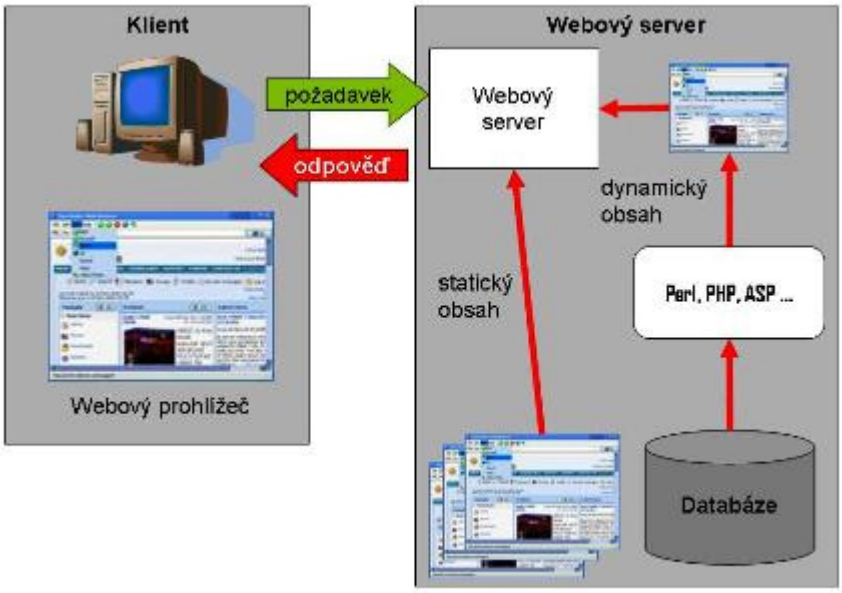
```
cosi  A  1.2.3.4
```
[5]

V protokolu DNS jsou předávány jako 32-bitové číslo.[6]

2 WEBOVÝ SERVER

Webový server může být:

- Počítač, který je odpovědný za vyřizování požadavků HTTP od klientů – programů zvaných webový prohlížeč. Vyřízením požadavků se rozumí odeslání webové stránky. Webové stránky jsou obvykle dokumenty HTML.
- Počítačový program, který provádí činnosti popsané v předchozím bodě (démon).



Obr. 2. Schéma komunikace klient – web server.

Jednotlivé webové servery se mohou v různých jednotlivostech značně lišit. Přesto mají několik společných vlastností.

Každý webový server je připojen k počítačové síti a přijímá požadavky ve tvaru HTTP. Tyto požadavky vyřizuje a počítači, který požadavek vznesl, vrací odpověď. Odpověď obvykle představuje nějaký HTML dokument. Může to být ale i dokument v jiném formátu – text, obrázek apod.

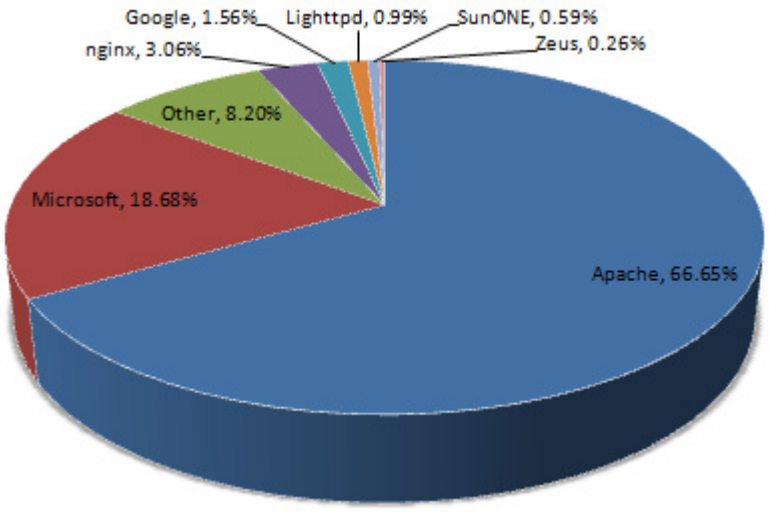
Obvykle server nějakým způsobem protokoluje přijímané požadavky. To pomáhá správci webového serveru vytvářet statistiky a podle typu a množství požadavků optimalizovat obsah, způsob uložení i způsob prezentace požadovaných dat.

Webový server má v zásadě dvě možnosti, jak získávat informace, které vrací klientům:

- jsou to buď předem připravené datové soubory (HTML stránky), které webový server bez změny poskytne klientovi (tzv. statický obsah)
- teprve na základě požadavku klienta jsou data shromážděna (přečtena ze souboru, databáze, nebo nějakého koncového zařízení), zformátována a připravena k prezentaci ve formátu HTML a poskytnuta webovému prohlížeči (tzv. dynamický obsah)

K dynamickému vytváření obsahu se používá celá řada různých technologií (Perl, PHP, ASP, ASP.NET, JSP apod.). Statický obsah je schopen server poskytnout významně rychleji než dynamický. Na druhé straně pomocí dynamického obsahu lze poskytovat mnohem větší obsah informací a lze reagovat i na různé „ad hoc“ dotazy klientů. Proto se v praxi v mnoha případech oba způsoby poskytování obsahu kombinují.[7]

Mezi nejčastěji používané webové servery patří bezesporu Apache web server a jako druhý nejčastější je webový server firmy Microsoft Windows Internet Information Services (IIS).



Obr. 3. Využití webových serverů v březnu 2009.[8]

2.1 Apache

Apache HTTP Server je softwarový webový server s otevřeným kódem pro Linux, BSD, Microsoft Windows a další platformy. V současné době dodává prohlížečům na celém světě většinu internetových stránek.[9]

Podle údajů poskytovaných webovou stránkou http://news.netcraft.com je dnes na webovém serveru Apache provozováno neuvěřitelných 104 796 820 (březen 2009) serverů.

Název vznikl z úcty a obdivu k domorodému kmenu nativních Američanů – Apačů anebo anglického slovního spojení „A patchy server“ (patchovaný server, kdysi byl Apache pouze sada patchů pro jiný web server). Jako indiánský symbol je ve znaku ptačí pero.[9]

Obr. 4. Značka webového serveru Apache.

Apache je volně šiřitelný otevřený software. To že se dá software získat zdarma, je sice důležité, ale daleko důležitější je skutečnost, že se jedná o otevřený software.[2]

2.2 Moduly

Apache je vysoce konfigurovatelný webový server s modulárním návrhem. Je velmi snadné jeho schopnosti rozšířit. Kdokoliv s patřičnou znalostí programování v jazycích C nebo Perl může napsat modul, který provádí zvláštní funkce. To znamená, že jsou k dispozici stovky modulů, které se dají využít.[2]

Moduly Apache serveru nelze vyjmenovat, neboť je jich opravdu mnoho, můžeme ale vyjmenovat skupiny, do kterých tyto moduly patří:

Týkající se systémového prostředí: Tyto direktivy umožňují nastavovat a měnit systémové proměnné.

Autentifikace a řízení přístupu: Tyto direktivy zajišťují autentifikaci a autorizaci uživatelského přístupu do chráněných částí webového serveru.

Dynamické generování obsahu: Direktivy umožňují spouštění externích programů, jako jsou skripty CGI, nebo příkazů SSI, pomocí nichž lze dynamicky vytvářet obsah serveru.

Konfigurování typu obsahu: Tyto direktivy umožňují řídit typy MIME u souborů.

Adresářové výpisy: Direktivy nám umožňují řídit formátování adresářových výpisů.

Záhlaví odpovědi: Pomocí těchto direktiv lze řídit hlavičky odpovědí protokolu HTTP.

Informace o serveru a protokolování: Direktivy umožňují řídit protokolování serveru a jeho stavové informace.

Mapování URL: Direktivy nám umožňují mapovat a přepisovat lokátory URL a vytvářet pro ně náhradní jména.

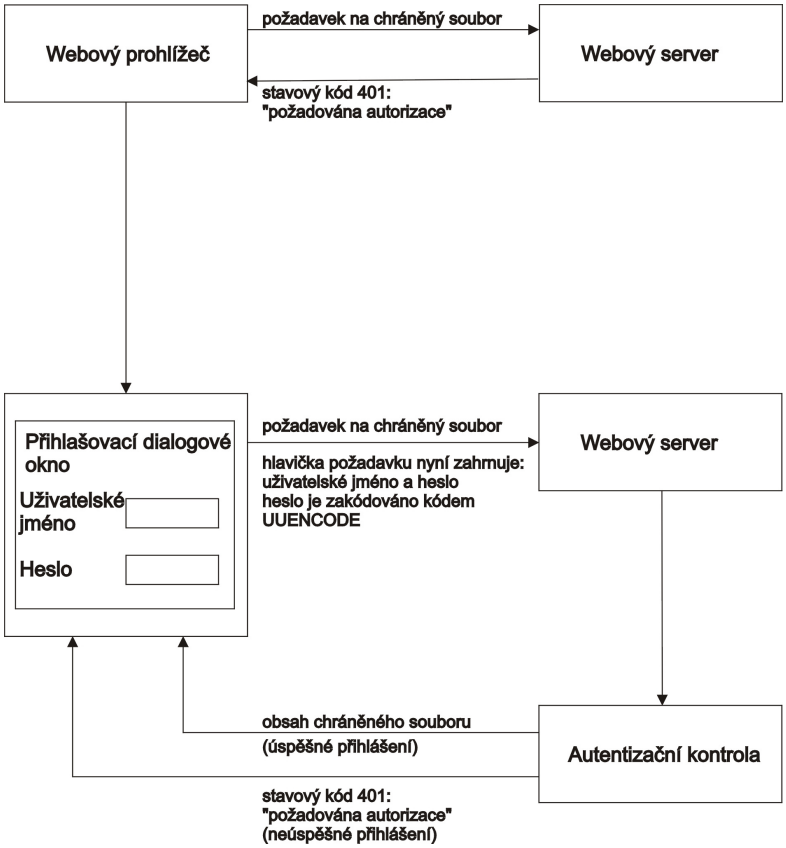
Různé moduly: Tyto direktivy umožňují řízení rozličných aspektů serveru, jakými jsou třeba služby proxy, modul WEBDAV apod.

2.3 Autentizace a autorizace

Server Apache již nějaký ten pátek disponuje podporou základní autentizace protokolu HTTP. Pro Apache bylo napsáno značné množství modulů, které tento typ autentizace zajišťují.[2]

Existují mnohé způsoby autentizací ať už použitím souboru hesel, databází, různými technickými prostředky jako jsou USB klíče, tokeny, ale i také biometrické otisk prstu, sítnice či duhovka oka.

Základem autentizace protokolu http je poměrně jednoduchá. K autentizaci uživatelů se používá mechanismu výzvy a odpovědi.



Obr. 5. Proces základní autentizace protokolu HTTP.

Autentizace versus autorizace

Mnoho lidí zaměňuje pojem autentizace a autorizace a někteří si dokonce myslí, že se jedná o stejnou věc, což samozřejmě není pravda.

V počítačové terminologii *autentizace* obvykle zahrnuje poskytnutí uživatelského jména a hesla. Úspěšné přijetí a kontrola jména a hesla znamená, že jsme ti, za které se vydáváme. Jinými slovy – *autentizovali* jsme sami sebe.[2]

Pokud byl proces *autentizace* proveden korektně, je nám většinou udělen, pokud tomu nebrání další systémové zabezpečovací prvky, *autorizovaný* přístup. Ve zkratce řečeno po *autentizaci* následuje *autorizace*.

3 APLIKACE EXPIN

Tato aplikace byla vytvořena pro účely exportu logovacích údajů (jmen a hesel) z SQL databáze knihovního systému. Data se zapisují do batch souboru v textové a nešifrované podobě. Celá operace se provádí v systémové oblasti serveru, mimo webový obsah. Soubor je dále zpracován a výsledkem je soubor .htpasswd s šifrovanými hesly v MD5 šifře. Ten je již použit pro účel autentizace chráněného obsahu webového serveru Apache. Aplikace vznikla v konstrukčním (programovacím) prostředí Delphi (Borland), za pomoci mého kolegy pana Ing. Františka Janoše.

3.1 Databáze SQL

Databáze (neboli **Datová základna**) je určitá uspořádaná množina informací (dat) uložená na paměťovém médiu. V širším smyslu jsou součástí databáze i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento software se v české odborné literatuře nazývá systém řízení báze dat (SŘBD). Běžně se označením *databáze* – v závislosti na kontextu – myslí jak uložená data, tak i software (SŘBD).[10]

SQL je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. SQL je zkratka anglických slov **Structured Query Language** (strukturovaný dotazovací jazyk).[11]

3.2 Delphi

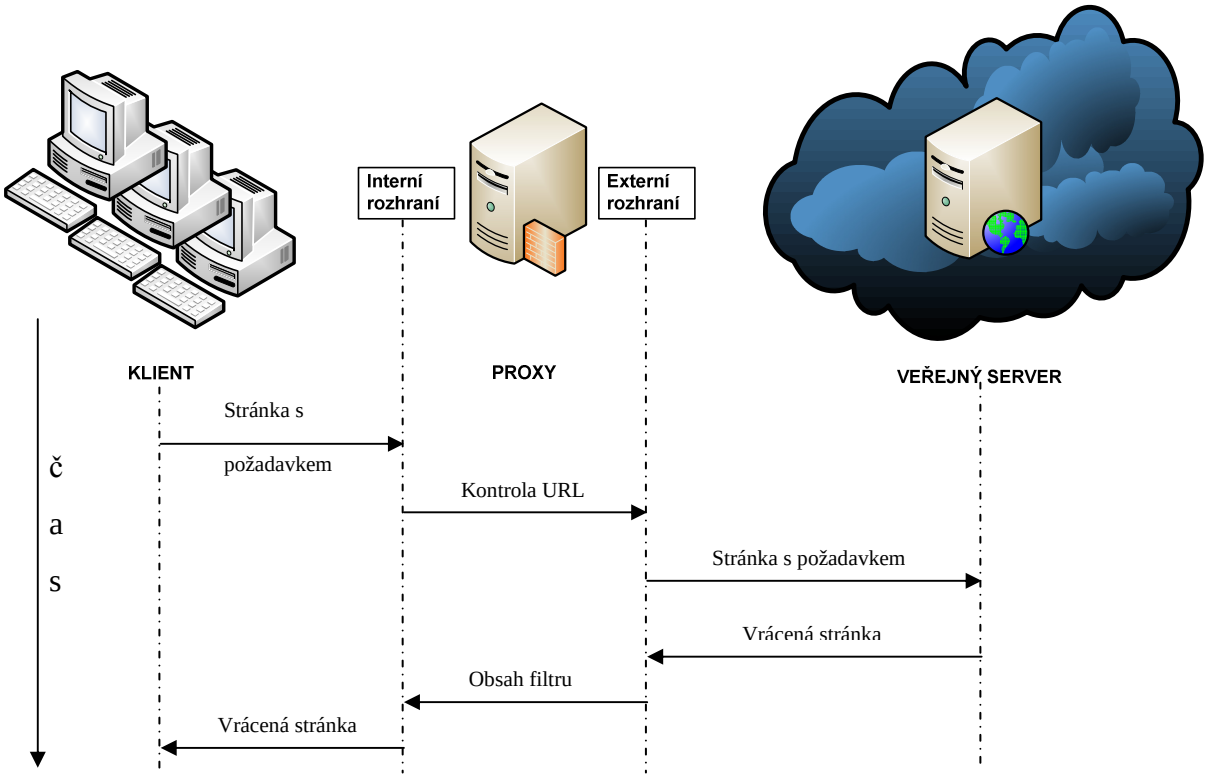
Delphi je integrované grafické vývojové prostředí firmy Borland určené pro tvorbu aplikací na platformě MS Windows v jazyce Object Pascal (objektové nástavbě Pascal). Obsahuje systém RAD (Rapid Application Development), který umožňuje vizuální návrh grafického uživatelského rozhraní, na jehož základě je automaticky vytvářena kostra zdrojového kódu, což výrazně urychluje vývojový cyklus.

Programování v Delphi je z velké části založeno na použití komponent. Komponenta je malý program (balíček funkcí), který vykonává určitou činnost (například zobrazuje text nebo obrázky, přehrává multimédia, komunikuje s databází, zprostředkovává FTP přenos, atd…).

Velkou předností Delphi proti některým konkurenčním produktům jsou knihovny komponent, které jsou jejich součástí (např. VCL, CLX, Indy …). Dodávané komponenty významně usnadňují tvorbu aplikací. Další komponenty lze stáhnout z internetu (některé jsou zadarmo, některé se musí koupit). V Delphi lze vytvářet vlastní komponenty.[4] [12]

4 PROXY SERVER

Proxy fungují tak, že naslouchají požadavkům o služby od interních klientů a pak je předávají na externí síť, jako kdyby byl klientem – původcem samotný server proxy. Jakmile obdrží proxy od veřejného serveru odpověď, vrátí tuto odpověď původnímu internímu klientskému počítači, jako kdyby byl sám původním veřejným serverem.



Obr. 6. Funkce proxy serveru.[3]

Proxy server můžeme chápat jak softwarový produkt (program), tak i jako hardwarové zařízení.

4.1 Možnosti proxy serveru

Ochrana soukromí

Pro cílový server je klientem proxy server a nikoliv původní klient. To má za následek, že cílovému serveru není známa IP adresa původního klienta. Zejména u webových proxy toto opatření není stoprocentní, protože některé z nich adresu klienta přidávají do upraveného požadavku. Úpravou požadavku lze ale zvýšit soukromí ještě víc, a sice odstraňováním cookies nebo jiných informací (např. referrer – informace o poslední navštívené stránce).

Zvýšení výkonu komunikace

Pokud se některé požadavky klienta opakují (např. požadavek na stažení loga Wikipedie, dotazy na DNS, atd.), může si proxy server uložit odpověď do vyrovnávací paměti a odpověď odeslat klientovi přímo, aniž by předal komunikaci k cílovému serveru.

Bezpečnost

Aplikační proxy server může analyzovat komunikaci a zjišťovat přítomnost např. virů. Dále může procházející požadavky šifrovat a dešifrovat.

Připojení více klientů k internetu

Klienti nemusí mít přiřazeny veřejné IP adresy, a přesto mohou mít přes proxy server přístup ke službám na internetu (toho je také možno docílit překladem IP adres, tzv. NAT, který je často zkombinován s firewallem).[13]

Možnosti proxy serverů se rozšiřují mnohdy podle jejich potřeby. Existuje mnoho open-source proxy serverů, které mají otevřený kód a možnost psát pro ně mnoho modulů v jazycích C, Perl a dalších. Mezi takové proxy servery patří i proxy server Proxomitron, který bude dále v praktické části ukázán a použit.

II. PRAKTICKÁ ČÁST

5 KONFIGURACE DNS

Pro praktické použití celého systému je potřeba navrhnout a nakonfigurovat nové DNS aliasy (A recordy) pro různé EIZ. Všechny tyto aliasy budou směřovány na stejný webový server, ten již podle konfigurace virtuálních serverů bude sám směřovat na příslušné originální EIZ.

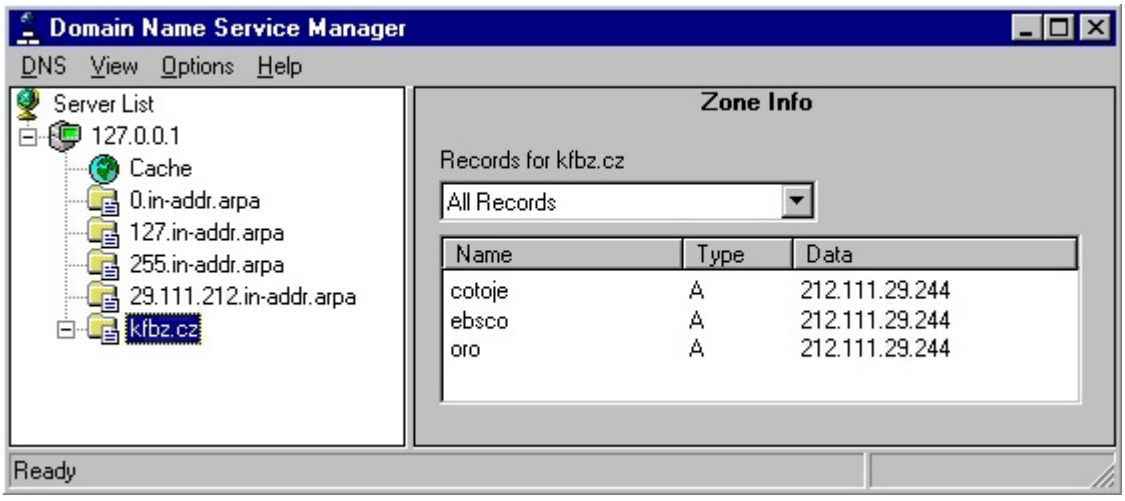
Knihovna je vlastníkem primárního a sekundárního DNS serveru, který je připojen do internetu 24 hodin denně a poskytuje DNS služby primárně knihovně. Servery se automaticky pravidelně synchronizují s DNS servery poskytovatele internetového připojení a ty dále s dalšími DNS v síti internet. DNS servery je možno kontaktovat na IP 212.111.29.242 (primar) a 212.111.29.244 (secundar).

5.1 Tvorba A recordu pro EIZ

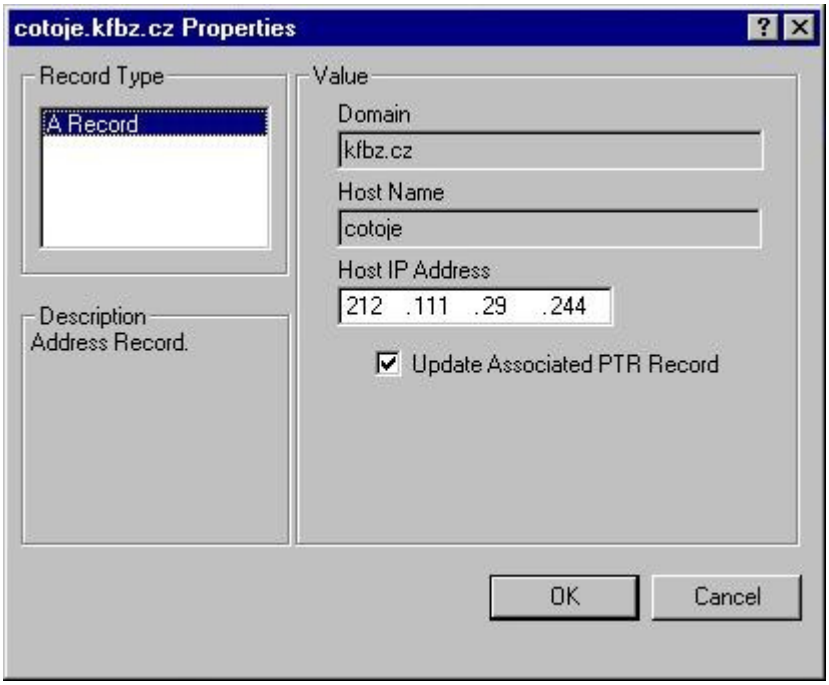
A recordy jsou vytvořeny pro tyto EIZ:

Originální web adresa EIZ	***alias adresa knihovny na EIZ***
`oxfordreference.com`	`oro.kfbz.cz`
`cotoje.cz`	`cotoje.kfbz.cz`
`ebscohost.com`	`ebsco.kfbz.cz`

všechny A recordy jsou směřovány na webový server kfbz.cz s WAN IP 212.111.29.244.



Obr. 7. DNS záznamů (upravený seznam).



Obr. 8. Vlastnosti A recordu cotoje.kfbz.cz.

Nastavení DNS serveru v knihovně je s refreshem 60 minut, tzn. každou hodinu si DNS vyměňují aktuální nastavení.

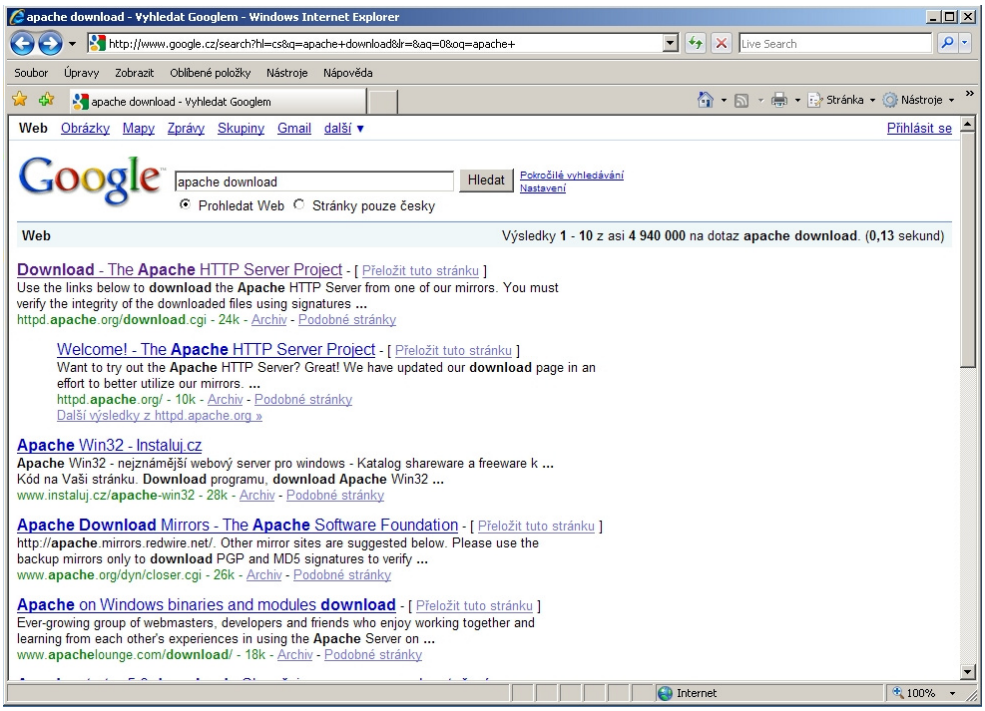
6 KONFIGURACE APACHE

Nejdůležitějším bodem celého systému je webový server Apache, který provádí řadu důležitých funkcí. Mezi skalní funkce tohoto projektu je autentizace a autorizace, pomocí které se provádí speciální proxy operace.

Z bezpečnostních důvodů nebudou uváděny přímé výpisy z konfiguračního souboru webového serveru Apache knihovny, bude demonstrována na instalaci Apache v2.2. Rozdíly jsou minimální.

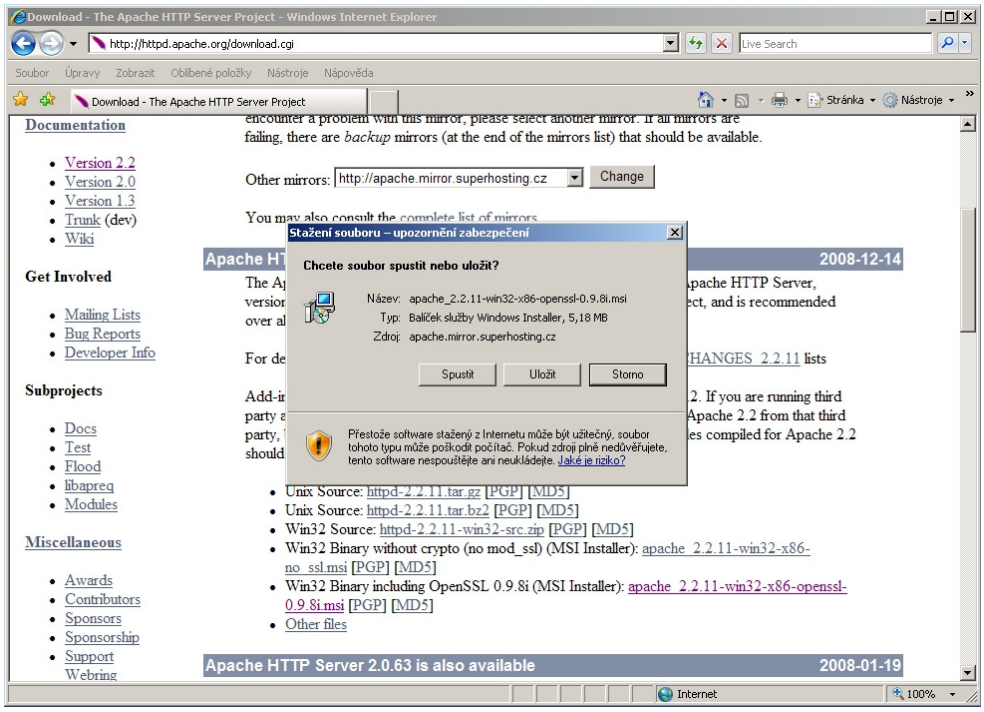
6.1 Instalace Apache serveru

K instalaci Apache byla použita nová verze **apache_2.2.11-win32-x86-no_ssl.msi** stažená přímo z oficiálních webových stránek http://httpd.apache.org/.

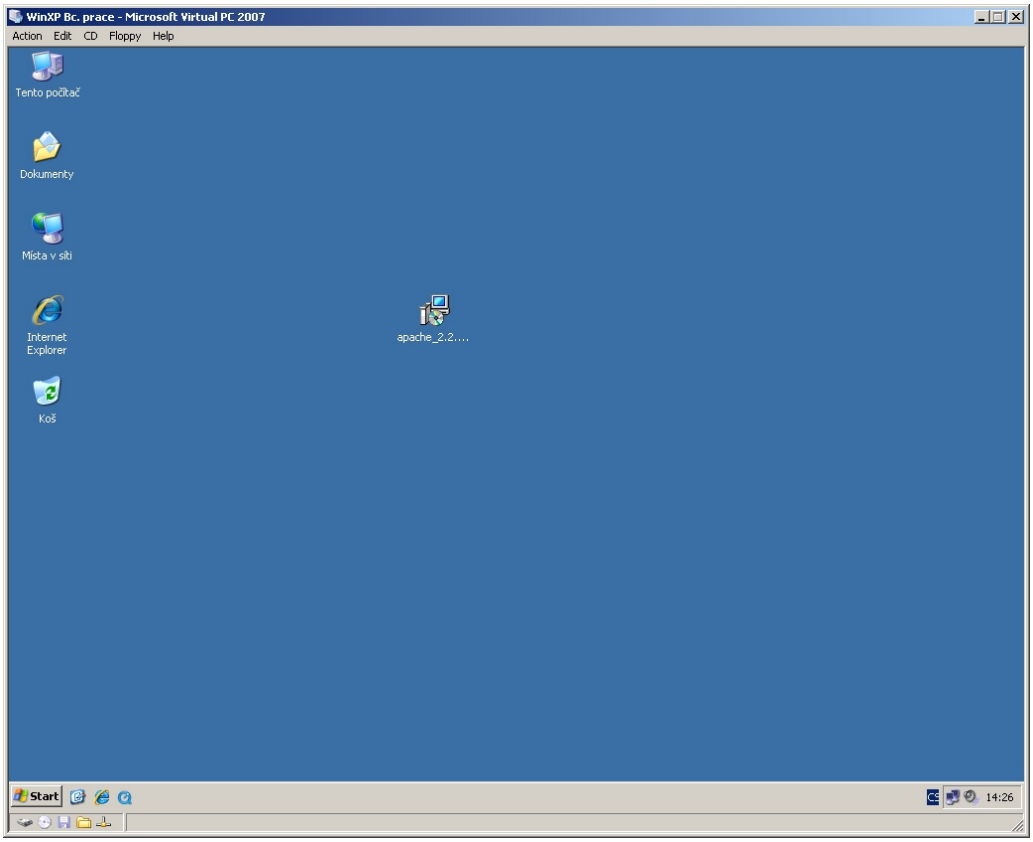


Obr. 9. Formulace dotazu pro stažení Apache přes google.com.

Osobně doporučuji vždy stahovat serverové produkty z oficiálních zdrojů. Vzhledem k účelům těchto aplikací je velice riskantní užívat neoficiální zdroje a vzhledem k volně otevřenému zdrojovému kódu, je možno si tam nějaký ten backdoor dopsat. V té chvíli můžeme mluvit o vážném bezpečnostním problému.

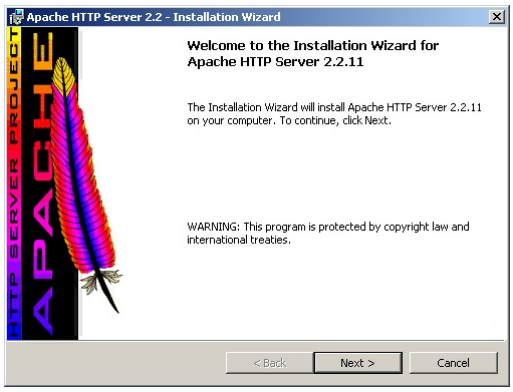


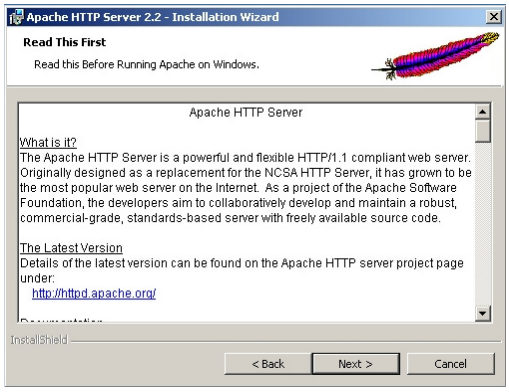
Obr. 10. Samotný instalační soubor serveru Apache má kolem 5 MB.



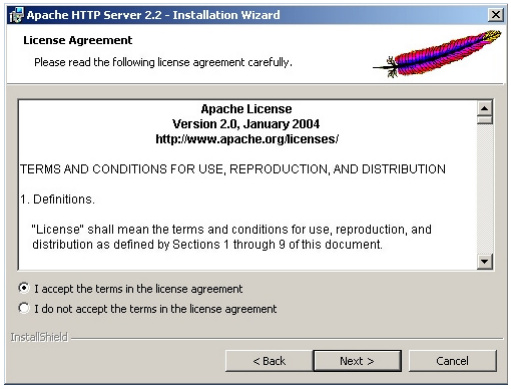
Obr. 11. Uložený instalační soubor na ploše Windows.

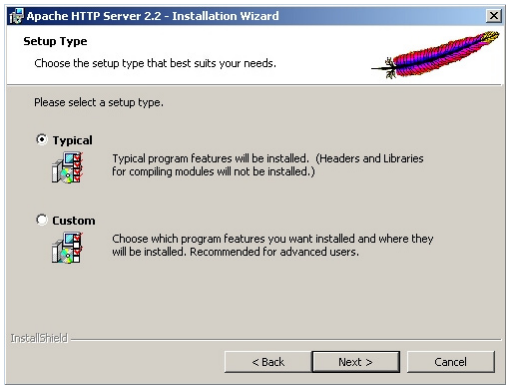
Instalátor Apache web serveru je dodáván pro mnoho platforem mezi ty základní patří Windows, Linux. Je distribuován buď s podporou SSL, nebo bez ní. V tomto projekt není SSL použito, proto použijeme instalaci bez něj.



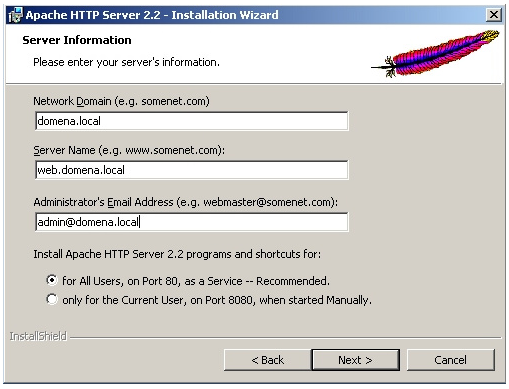


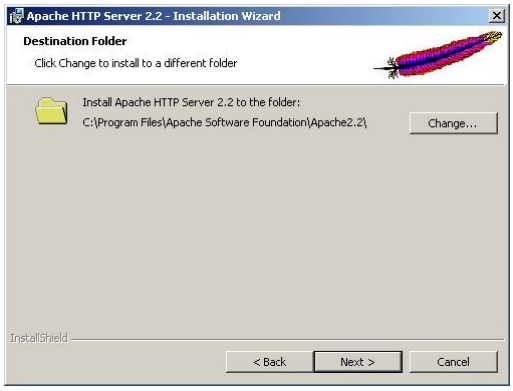
Obr. 12. a 13. Instalace Apache web serveru a informace o Apache projektu.





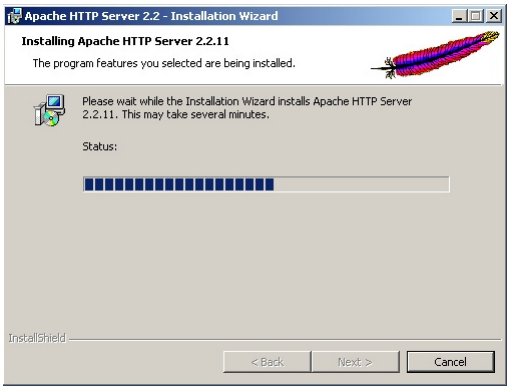
Obr. 14. a 15. Licenční ujednání a volba instalace.

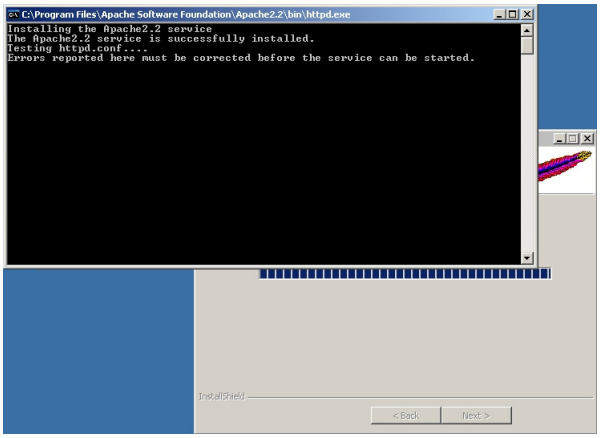




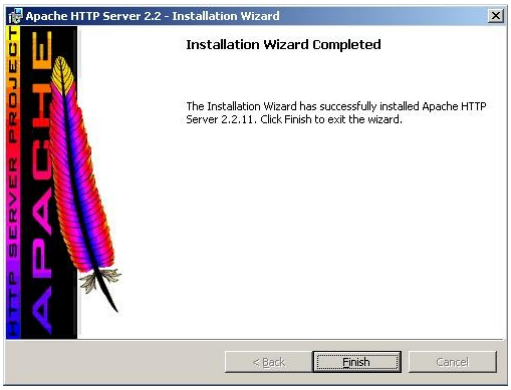
Obr. 16. a 17. Nastavení informací o serveru a výběr portu

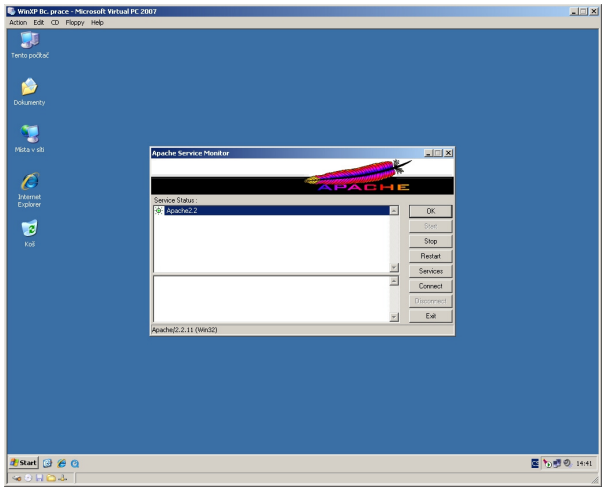
a zvolení instalační cesty (path).



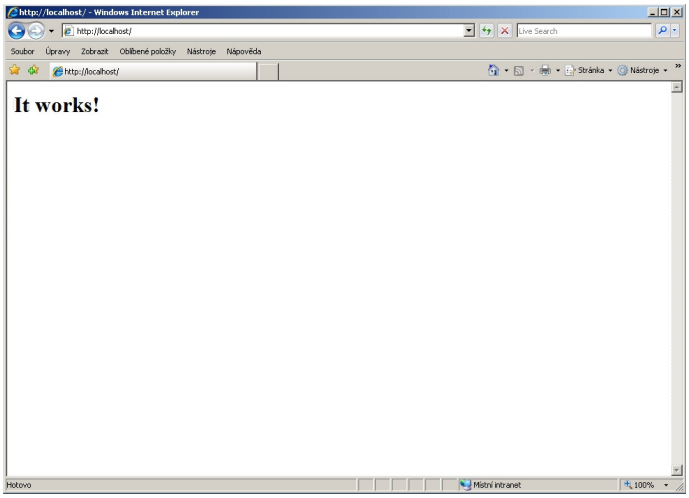


Obr. 18. a 19. Instalační procedura a registrace služby (service).





Obr. 20. a 21. Úspěšné dokončení instalace a ukázka Apache monitoru (systray).



Obr. 22. Odzkoušení Apache (http://localhost).

6.2 Konfigurace Apache serveru

Velká výhoda webového serveru Apache je určitě i v jeho snadné konfigurovatelnosti, princip konfigurace Apache tkví totiž v úpravě jednoduchých textových souborů.

Nejdůležitější konfigurační soubor **httpd.conf** najdeme v adresáři **C:\Program Files\Apache Software Foundation\Apache2.2\conf**.

Ukázka konfiguračního kódu - zde je konfigurace adresáře odkud se načítají webové stránky:

```
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot
"C:/Program Files/Apache Software Foundation/Apache2.2/htdocs"
#
# Each directory to which Apache has access can be configured with
respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
```

Vzhledem k tomu, že bude webový server použit pro jiné účely, než zobrazování webových stránek nemusí být konfigurovat běžné parametry jako je př. DocumentRoot – umístění obsahu webu, DirectoryIndex – které soubory se mají chovat jako spouštěcí (indexy) + asi dnes životně důležitá instalace a podpora PHP.

Konfigurace podpory virtuálních serverů

Pro správný chod virtuálních serverů musí být v konfiguračním souboru upraven řádek:

```
# Virtual hosts
# Include conf/extra/httpd-vhosts.conf
```

- je značka komentáře (REM příkaz), tzn. vše skryté za touto značkou nebere Apache v potaz, jako by to pro něj neexistovalo.

Řádek se tedy upraví tak, že bude tato značka odstraněna. Tím je povolen rozšiřující konfigurační soubor.

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

Konfigurace podpory modulů proxy

Přestože je již většina nejčastěji používaných modulů v základní konfiguraci povolena, je nutno pro správných chod povolit ještě dva moduly speciálně určené pro proxy operace a to proxy_module modules/mod_proxy.so a proxy_http_module modules/mod_proxy_http.so.

Oba tyto moduly jsou také jen zakomentované, stačí tedy odstranit komentář.

```
LoadModule proxy_module modules/mod_proxy.so
# LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
# LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
# LoadModule proxy_connect_module modules/mod_proxy_connect.so
# LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

Modul proxy a proxy_http implementuje do Apache bránu pro komunikační protokoly jako je FTP, SLL, HTTP a další.

Tímto krokem je úprava souboru httpd.conf ukončena.

6.3 Konfigurace virtuálních serverů

Dále musí být provedena úprava konfiguračního souboru **httpd-vhosts.conf**. V tomto souboru se v nově nainstalované verzi Apache 2.2 konfigurují podrobnosti pro virtuální weby, je možno tuto část kódu zakomponovat do httpd.conf, ale nemá to žádný praktický význam. Direktiva všech modulů a přepínačů je pro obě verze stejná. Soubor se nachází v adresáři **C:\Program Files\Apache Software Foundation\Apache2.2\conf\extra**.

Ukázka příkladu virtuálního serveru v souboru httpd-vhosts.conf

```
<VirtualHost *:80>
ServerAdmin webmaster@kfbz.cz
DocumentRoot
"C:/Program Files/Apache Software Foundation/Apache2.2/htdocs"
ServerName cotoje.kfbz.cz
ProxyPass / http://www.cotoje.cz/
ProxyPassReverse / http://www.cotoje.cz/
ProxyRemote * http://172.16.2.199:3128
ErrorLog
"C:/Program Files/Apache Software
Foundation/Apache2.2/logs/cotoje.kfbz.cz-error.log"
CustomLog
"C:/Program Files/Apache Software
Foundation/Apache2.2/logs/cotoje.kfbz.cz.log" combined
<Location "/">
Order allow,deny
Allow from all
Options Indexes
AuthType Basic
AuthName
"Pro vstup na cotoje.kfbz.cz zadejte jméno a heslo."
AuthUserFile
"C:/Program Files/Apache Software Foundation/Apache2.2/bin/.htpasswd"
Require valid-user
</Location>
</VirtualHost>
```

Globální nastavení webu je přebíráno z nadřazeného httpd.confu, každý virtuální web má zde však své speciální nastavení. To může být různé od zvoleného portu až po umístění souborů s logy atd…

Možnosti Apache jsou zde opravdu skoro nekonečné, to je důvod oblíbenosti tohoto produktu. Blíže si rozebereme výše uvedenou konfiguraci pro EIZ cotoje.cz.

<VirtualHost *:80> </VirtualHost>

Začáteční a koncový kontejner udávající konfiguraci virtuálního serveru na portu 80.

***** může být přímo nahrazena doménovým jménem nebo lépe IP adresou, slouží k přesné specifikaci virtuálního hostitele, může být uvedeno i více adres a portů (př.: 172.16.5.104:80 localhost:8080 218.141.15.162:10000)

:80 udává port, na kterém virtuální server naslouchá (Listen), může být taky nahrazen jiným nebo i *, v takovém případě přebírá nastavení z nadřazeného httpd.confu

ServerAdmin

Tato direktiva ukazuje kontaktní adresu většinou ve spojení s chybovými hláškami, není li tato direktiva zadána, opět se přebírá z nadřazené konfigurace serveru.

DocumentRoot

Direktiva ukazuje na adresář s uloženými dokumenty pro webové stránky. Vzhledem k tomu že webový server nebudeme používat pro zobrazování webového obsahu, nemusí být tato direktiva vypsána, popř. může vést na prázdný adresář.

ServerName

Tato direktiva nastavuje tzv. hostitelské jméno serveru. V našem případě je to **cotoje.kfbz.cz**.

ProxyPass

Direktiva má možnost pracovat jako takové „zrcadlo“, to znamená, že požadavek bude poslán na cílovou adresu. V našem případě **/** značí cokoliv v pracovním adresáři (pro nás to znamená jakoukoliv činnost (stránku), přepošli na reálnou stránku **cotoje.cz**). Vypadá to pak jako by byl obsah stránek lokálně na webovém serveru a ne někde v internetu.

Pro příklad: napsáním **cotoje.kfbz.cz/clanek.htm** předá tato direktiva dotaz ve formátu **cotoje.cz/clanek.htm**, zobrazeno však zůstane **cotoje.kfbz.cz/clanek.htm**.

ProxyPassReverse

Tato direktiva umí nahradit v hlavičce *Location* HTTP dotazu, vzdáleného žadatele za svou a tím pak od poskytovatele informace zpětně tyto data přijmout a vrátit je původnímu žadateli. Každý požadavek je takto předáván na vzdálený server. Navíc umí využívat vyrovnávací paměť, tzn. snižuje zatížení při velkém počtu dotazů.

ProxyPassRemote

Tato direktiva předává dotazy jinému proxy serveru. Podporuje pouze HTTP protokol.

***** zde zajišťuje parametr direktivy, že se jedná o všechny požadavky

http://172.16.2.199:3128 je IP proxy serveru, který tyto požadavky dále zpracovává, pro nás to bude dále zmiňovaný proxy server Proxomitron

ErrorLog a CustomLog

Logovací soubory ukládající error položky a volitelný formát položek.

<Location "/"> </Location>

Začáteční a koncový kontejner udávající řízení přístupu. Přístup je zpracováván podle pořadí vepsaných direktiv.

Order allow,deny

Allow from all

Order direktiva říká v jakém pořadí budou vyhodnoceny direktivy **allow** (přístup povolen) a **deny** (přístup zakázán). Takto napsaná direktiva nám říká, že povoluje všechny žadatele na seznamu **allow** a zakazje všechny na seznamu **deny**. Vzhledem k následující direktivě **Allow from all** (seznam povolených), jsou všichni žadatelé povoleni.

Options Indexes

Tato direktiva říká, že se v adresáři nenachází index soubor, tzn. pokud by nebyla použita možnost proxy, provedl by se pouze výpis adresáře.

AuthType Basic

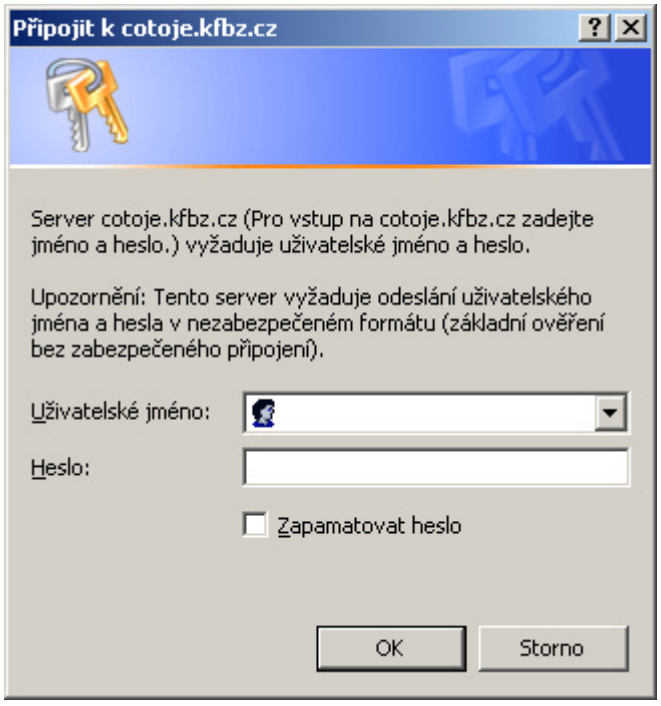
Touto direktivou definujeme způsob autentizace pro námi zvolený virtuální server. Tato direktiva může mít parametr **basic** nebo **digest**. **Basic** metoda požaduje jméno a heslo v textové podobě (tak je i předáváno, nízká bezpečnost). Oproti tomu metoda **digest** je bezpečnější, ale nefunguje ve všech webových prohlížečích.

AuthName

Tato direktiva nastavuje pojmenování chráněné oblasti. Prohlížeč vyžaduje jméno a heslo dialogem, pro takto chráněnou oblast.

Direktivu můžeme použít pro vypsání námi zvoleného textu v logovacím okně.

```
AuthName "Pro vstup na cotoje.kfbz.cz zadejte jméno a heslo."
```



Obr. 23. Ukázka logovacího okna s použitím

textu z AuthName.

AuthUserFile

Direktiva udává cestu a název souboru s logovacími jmény a hesly. Strukturou je to textový soubor. Heslo je při vložení šifrováno v MD5. Pro vkládání a šifrování údajů lze použít program **htpasswd.exe** z adresáře C:\Program Files\Apache Software Foundation\Apache2.2\bin.

Samotný soubor se z bezpečnostních důvodů doporučuje ponechat někde na serveru mimo webovou část, aby se zabránilo jeho stažení nebo jinému úniku.

Dále musí tento soubor existovat před restartem služby Apache, pokud by jej konfigurace obsahovala, ale soubor fyzicky neexistoval, Apache se nespustí. Vypíše chybu v logu.

Soubor stačí vytvořit jen jako prázdný textový soubor bez přípony.

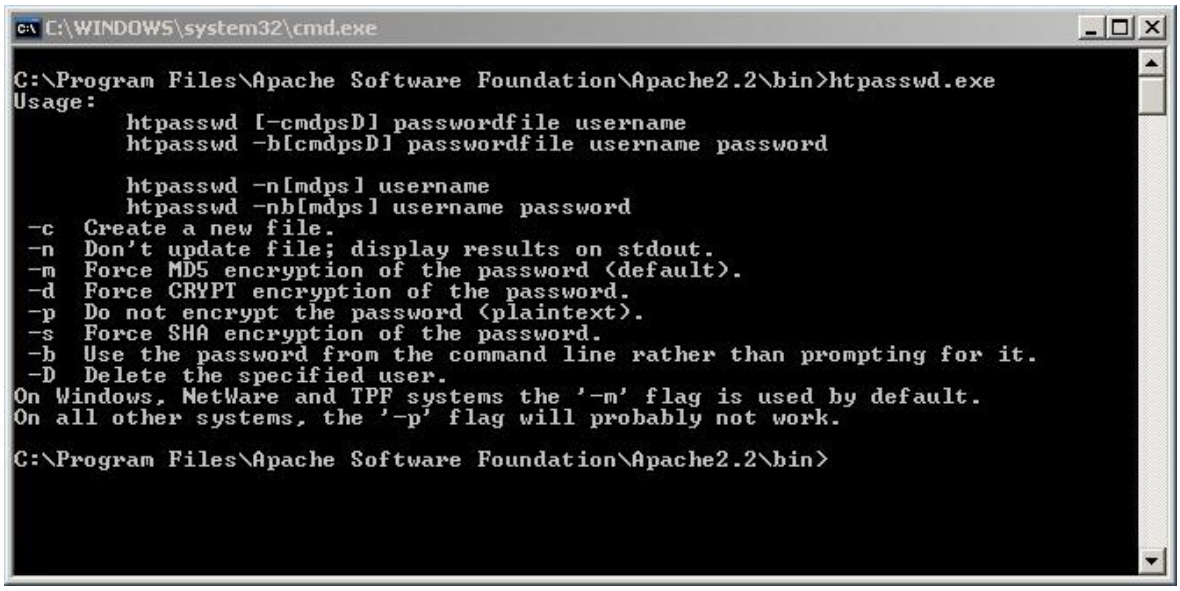
Require valid-user

Touto direktivou a parametrem se nastavuje, kteří uživatelé smí přistoupit k chráněnému adresáři. Parametr **valid-user** udává, že smí přistoupit všichni platní (ověření) uživatelé.

Konfigurace dalších virtuálních serverů, v našem případě EIZ pro domény oro.kfbz.cz a ebsco.kfbz.cz jsou obdobné. Většina parametrů je stejná. Rozdílné jsou direktivy ProxyPass.

7 AUTENTIZACE

Ta je pro naši konfiguraci Apache zvolena skrze soubor .htpasswd umístěný na cestě „C:/Program Files/Apache Software Foundation/Apache2.2/bin/.htpasswd“. Tento soubor můžeme vytvořit nejlépe integrovaným nástrojem **htpasswd.exe** ve stejném adresáři. Jedná se o Dosovskou aplikaci, její ovládání je možno přes Windows **cmd.exe**.



Obr. 24. Program htpasswd.exe.

Soubor s hesly založíme příkazem **htpasswd.exe –cmb *username userpassword***, dále pak už jen **httpasswd.exe –mb *username userpassword***. Takto tvoříme pro Apache soubor s hesly. Tvořit takto list o počtu deset záznamu je ještě reálné, jinak je to pokud je takových záznamů tisíce. Proto pro účely knihovny vznikla aplikace **Expin.exe**.

7.1 .htpasswd s použitím aplikace Expin

Vzhledem k velkému počtu aktivních uživatelů, kterým by měla být povolena možnost aktivně využívat EIZ knihovny se nedá použít výše popsaný postup tvorby souboru s hesly, je tedy potřeba tento proces automatizovat. Seznam uživatelů a jejich hesel již existuje v databázi, kterou využívá ke svému chodu knihovní systém. Uživatelé se pomocí těchto údajů přihlašují do svých kont (výpůjčky, rezervace, prolongace atd…). V tomto kontu je možno si změnit přihlašovací údaje pro toto konto, společné i pro přístup do EIZ. Přihlašují se skrze speciální PHP skripty na webových stránkách knihovny. Pro účely Apache je tento

způsob přihlašování však nepoužitelný, protože PHP v základní instalaci neumí a navíc PHP má mnohé chyby a omezení.

Proto jsme ve spolupráci s kolegou Františkem Janošem napsali jednoduchý jednoúčelový prográmek **expin.exe**. Ten provede export jmen a hesel ve speciální syntaxi vhodné pro **htpasswd.exe**, celá tato operace se provede do batch souboru. Ten pak svým spuštěním naplní soubor s hesly. Celá operace je však časově náročná (hodiny).

Ukázka části kódu programu expin.

```
while (not IbrQuery.Eof) do
begin
Usr:= Trim(IbrQuery['username']);
Psw:= Trim(IbrQuery['passwd']);
PomS:= PswPrgName + ' -mb ' + PswFileName + '_new "' + usr + '" "' + psw
+ '"' + Chr(13) + Chr(10);
Writeln(F, PomS);
Inc(Rec);
IbrQuery.Next;
Caption:= 'Export PIN ' + IntToStr(Rec);
Repaint;
end;
```

Při programování bylo využíváno fragmentů ukázkových kódů.

Soubor pak tvoří batch s takovou strukturou.

```
htpasswd.exe -mb .htpasswd_new "yyyyyyy" "xxxxxxxxxxx"
htpasswd.exe -mb .htpasswd_new "yyyyyyy" "xxxx"
htpasswd.exe -mb .htpasswd_new "yyyyyyy" "xxxx"
htpasswd.exe -mb .htpasswd_new "yyyyyyyyy" "xxxx"
htpasswd.exe -mb .htpasswd_new "yyyyyy" "xxxxxxxxxxxxx"
htpasswd.exe -mb .htpasswd_new "yyyyyy" "xxxxxx"
```

yyyyyyy – logovací jména uživatelů

xxxxxxx - logovací hesla uživatelů

Soubor batch je pak spuštěn a jeho výsledkem je pak již reálný soubor .htpasswd, který se používá k autentizaci uživatelů.

Ukázka takto tvořeného souboru.

```
sokola:$apr1$0w4.....$y4RGtiTz5UnL4aYbM5Dtd1
3190001:$apr1$0w4.....$xES6r9hmOmy.HdvWgV.yn/
3300316:$apr1$0w4.....$x9kHHTTleewIfnymhf0Ag1
3300644:$apr1$0w4.....$JQQ5uGI/arJ9nyTZY1IEE.
Jitka:$apr1$3w4.....$EbamRq6INEDiSYk9rq9231
3301963:$apr1$3w4.....$ZqZDnqWBi5cadn0VHFVKL/
```

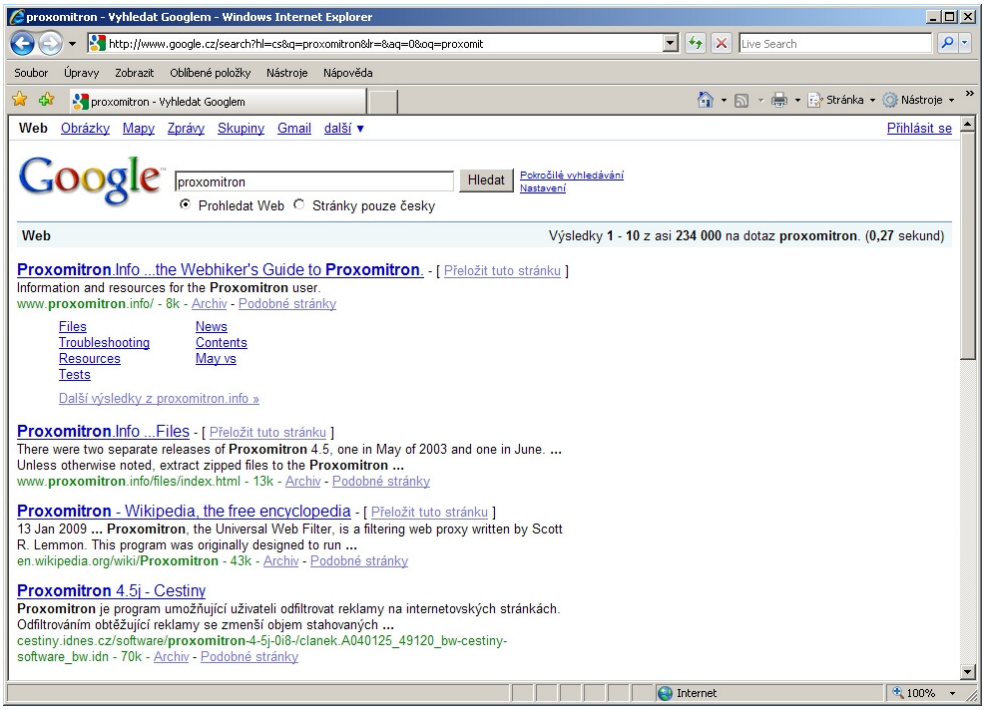
Logovací jména jsou viditelná, ale heslo je šifrováno v MD5. Vzhledem k tomu že Apache nepoužívá klasickou MD5, ale mutaci této šifry je dost složité heslo zpětně dešifrovat.

Export jmen (loginů) a hesel se provádí automaticky v nočních hodinách. Změna loginu a hesla se tedy promítne vždy až následující den.

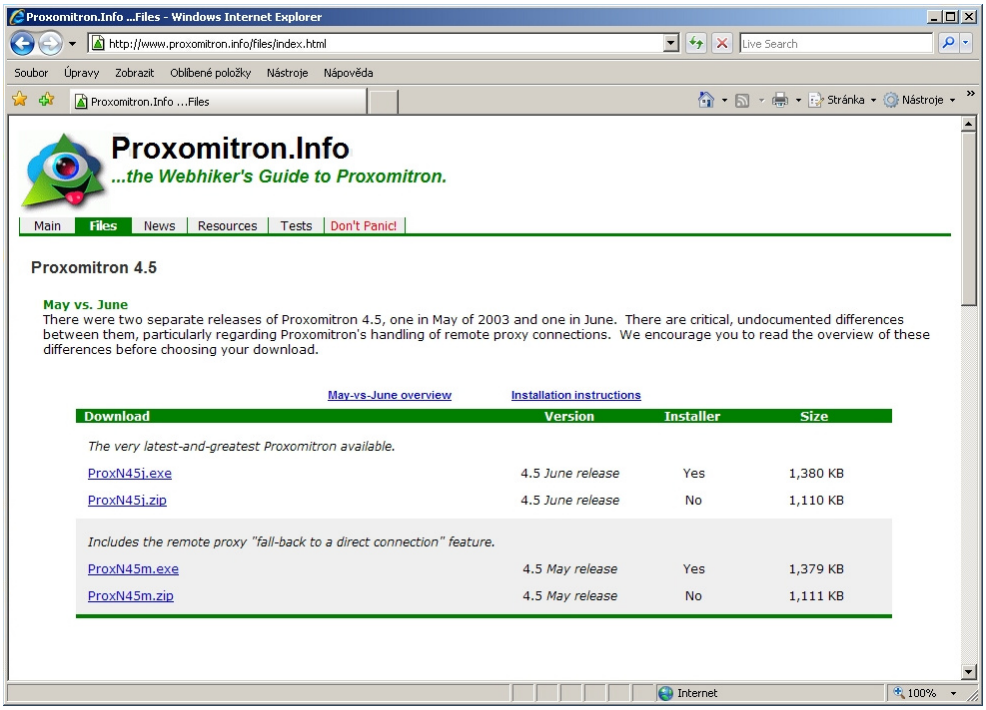
Nyní se aktivně pracuje na náhradě programu expin za přímé ověřování buď z databáze MySQL, nebo přímo databáze knihovního systému Interbase.

8 KONFIGURACE PROXY

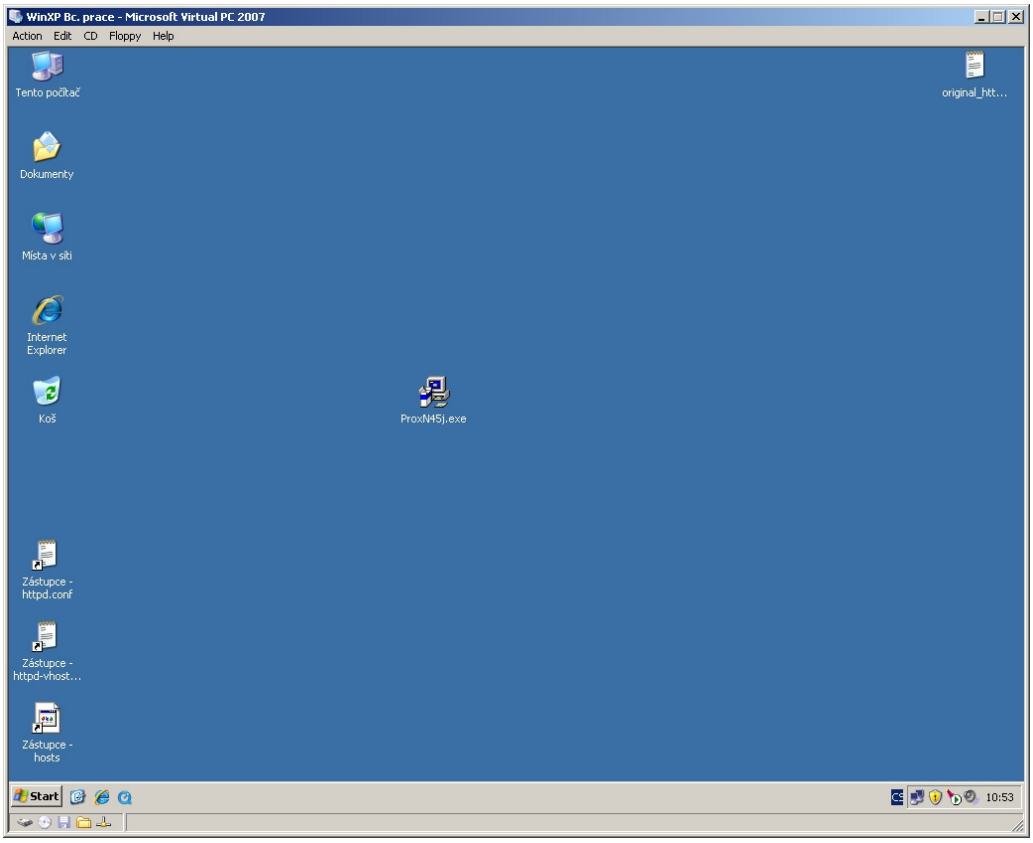
Použitý program Proxomitron je freeware. Stáhnout jej můžeme ze stránek **proxomitron.info**. Program můžeme i vyhledat skrze vyhledávače, třeba **google.com**.



Obr. 25. Formulace dotazu pro stažení Proxomitronu přes google.com.



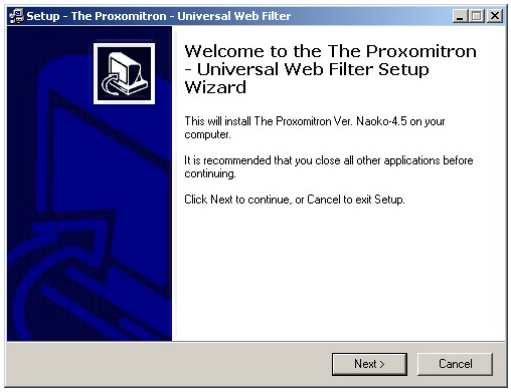
Obr. 26. Domácí stránky Proxomitronu, stránka pro download programu.

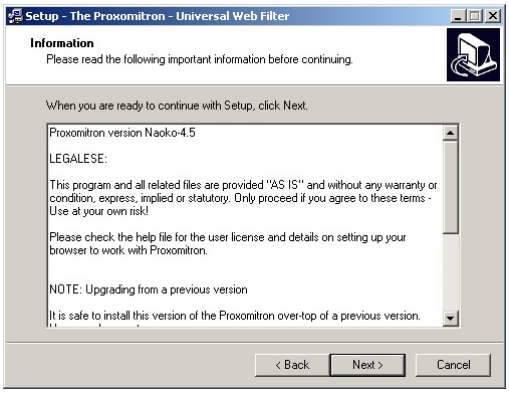


Obr. 27. Uložený instalační soubor na ploše Windows.

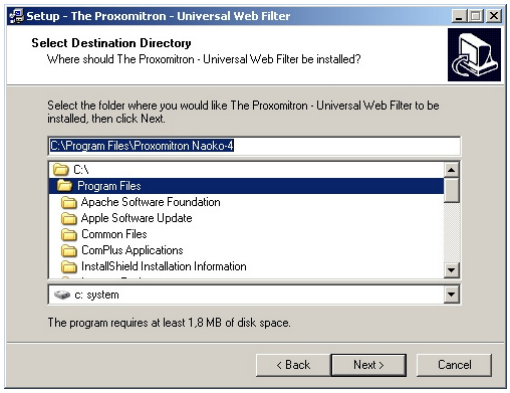
8.1 Instalace Proxomitronu

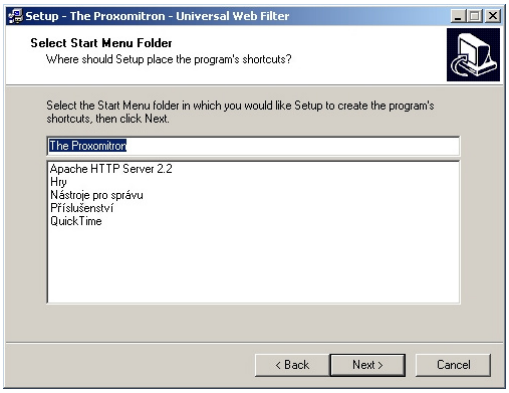
Instalace proxomitronu je jednoduchá a není potřeba ji nějak speciálně popisovat.



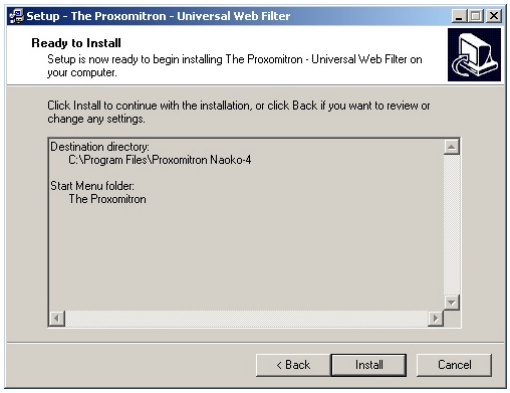


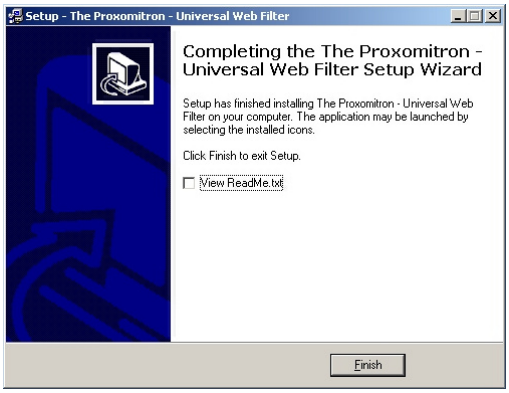
Obr. 28. a 29. Spuštění instalace a informace o programu a licenci.



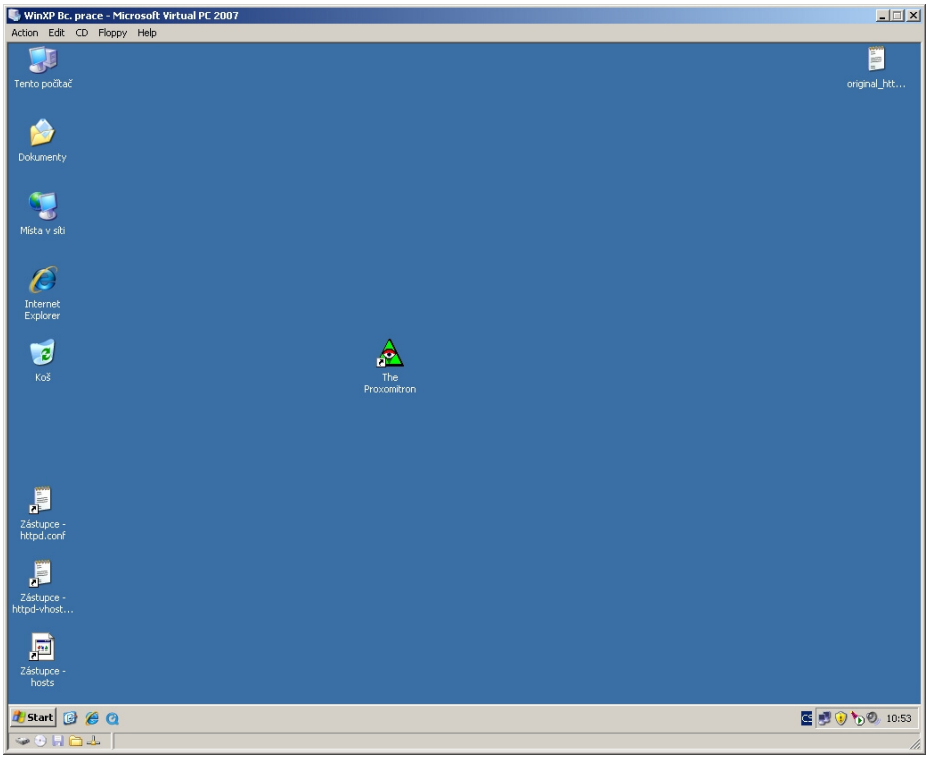


Obr. 30. a 31. Zadání instalační cesty a pojmenování ve složce Programy (Start).





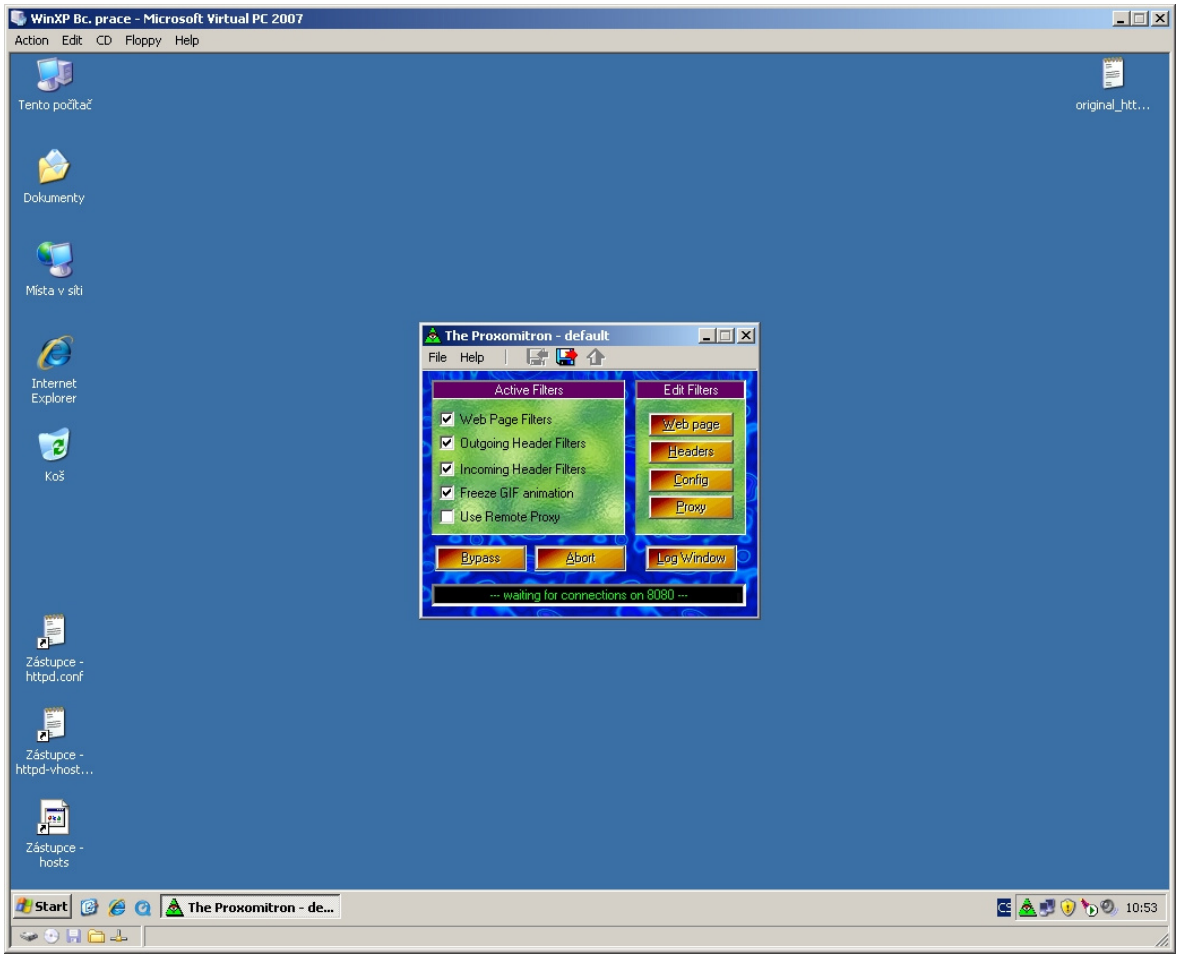
Obr. 32. a 33. Spuštění samotné instalace a úspěšné dokončení instalace.



Obr. 34. Zástupce programu Proxomitron na ploše.

8.2 Konfigurace Proxomitronu

Proxomitron je jednoduše konfigurovatelný a spravovatelný proxy server, celá jeho konfigurace vychází z povolování a zakazování jednotlivých filtrů nebo jejich částí. Opět je možno celý systém konfigurovat buď v textovém režimu (úpravou konfiguračních souborů) nebo využitím grafického režimu.



Obr. 35. Spuštěný Proxomitron, ikona programu v systray, ovládací okno na popředí.

Proxomitron se skládá ze dvou nejdůležitějších panelů oken a to **Active Filters** a **Edit Filters**.

Konfigurace okna Active Filters

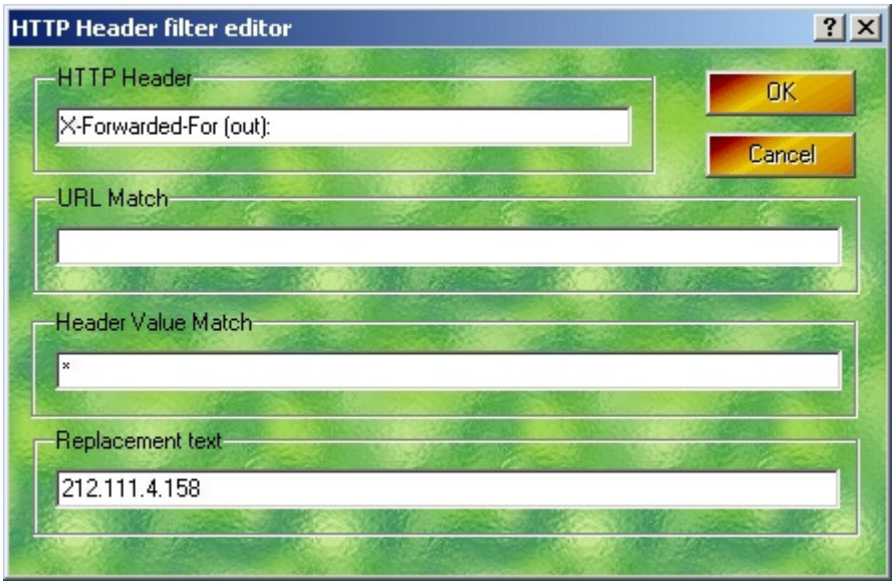
Pro funkčnost celého systému EIZ se z proxy serveru používá jen položka **Outgoing Header Filters**, tuto jedinou necháme zapnutou, zbytek vypneme.

Konfigurace okna Edit Filters

Editujeme jedinou položku a to **Headers** v té vyhledáme položku **X-Forwarded-For (out)** a zapneme ji na **out** rozhraní. Ve filtru zůstane pouze tato položka jako aktivní.

Obr. 36. Povolení položky X-Forvarded-For.

Položku dále upravíme tlačítkem Edit. Touto úpravou si zabezpečíme, že všechny požadavky od uživatele (libovolná IP adresa) v sobě budou mít přepsanou adresu na adresu knihovny a proto pro EIZ i povolenou adresu. Požadavek pak bude vrácen proxy a ten jej zase vrátí příslušnému žadateli.



Obr. 37. Úprava položky X-Forwarded-For.

To je v konfiguraci Headers Filtru vše.

Dále upravíme konfiguraci proxy serveru a to hlavně na stránkách Visuals, Access a HTTP.

Obr. 38. Konfigurace položky Config.

Úprava karty Visuals

Zde doporučuji vypnout používání textur a to zaškrtnutím políčka **Don't use textures**.

Úprava karty Access

Prohodíme checkbox z **Limit access to only this PC** na položku pod ní a to **Allow access to the following IP adress range** - tímto povolíme námi zvolený rozsah IP adres, pro nás je žádoucí každá IP adresa, takže nastavíme 0.0.0.0 – 255.255.255.255.

Úprava karty HTTP

Zde už jen změníme hodnotu portu z **8080** na **3128**. Tato operace je vhodná z důvodů že tento port 8080 využívají dost často webové servery. Dále se musí adresa stroje s proxy serverem a port shodovat s konfigurací Apache a to položkou **ProxyPassRemote**.

Konfiguraci dáme potvrdit **Ok**, potom **uložit** a program restartujeme.

Ukázka výpisu z logu při použití přístupu do EIZ

```
GET /img/tit_listovani_malacs_sbalit.gif HTTP/1.1
Host: www.cotoje.cz
Accept: */*
Referer: http://cotoje.kfbz.cz/default.aspx
Accept-Language: cs
UA-CPU: x86
Accept-Encoding: gzip, deflate
If-Modified-Since: Mon, 09 Jun 2003 12:19:00 GMT
If-None-Match: "0928c4e812ec31:2c74"
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR
2.0.50727; .NET CLR 3.0.04506.30)
Cookie: __utma=239221649.1740450417.1237902344.1237902344.1237902344.1;
__utmz=239221649.1237902344.1.1.utmccn=(direct)|utmcsr=(direct)|utmcmd=(n
one); ASP.NET_SessionId=4xg5u2451hdtqh45hf1kkc55;
.COTOJE=0A2954D3235D5428A6CEFF98F26017F275ECF1FC59210B782BF71AC5253BDAE5C
67FDCD9F562B0DB5540D7A19D5B2726F9044FA4E6EFC513799ADFC96FF2C4C2585ADDBE96
63EB32F651BE5166CDAB6CC0E71A6FF3A055CFB066195C74446CB70FF458210905527EAE4
069843567F69D8C820B7A85F98387AC6B4C0F8529C331
Authorization: Basic dXNlcjp0ZXN0
X-Forwarded-For: 212.111.4.158
X-Forwarded-Host: cotoje.kfbz.cz
X-Forwarded-Server: cotoje.kfbz.cz
Connection: keep-alive
```

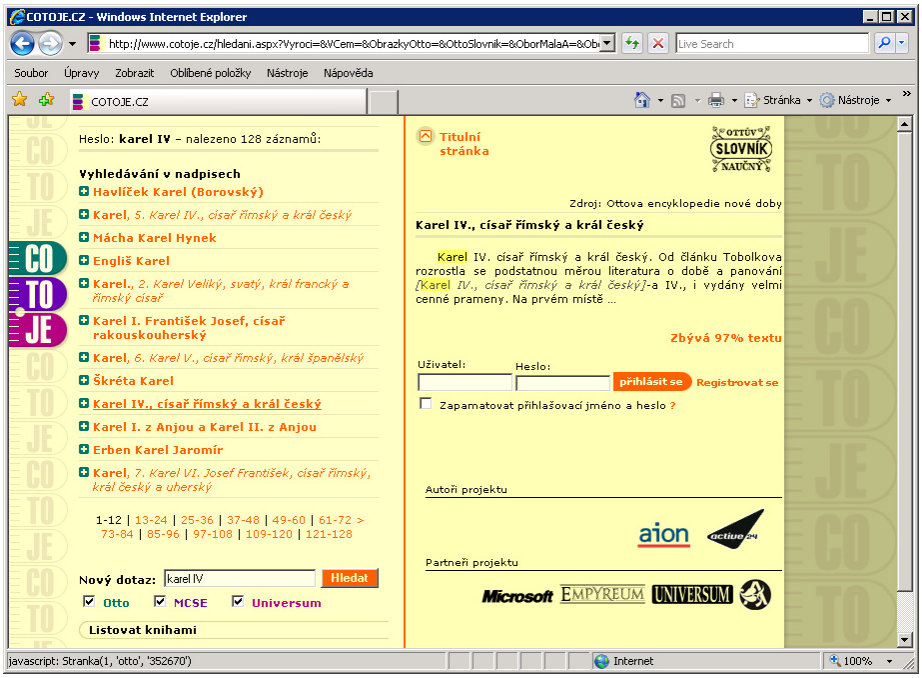
ZÁVĚR

Cíle bakalářské práce byly splněny. Byly vytvořeny DNS aliasy, byl nakonfigurován a zprovozněn webový a proxy server. Celý systém je v této době již v aktivním požívání čtenářů knihovny KKFB.

V této práci jsem použil software v licencích open source a freeware, tzn. vstupní náklady na projekt byly nulové.

Konfigurace i další údržba systému je dle mého názoru jednoduchá a efektivní.

Funkčnost znázorňují následující obrázky.



Obr. 39. Ukázka přístupu do databáze cotoje.cz volně z internetu, databáze poskytne jen část obsahu informací (zbývá 97% textu).

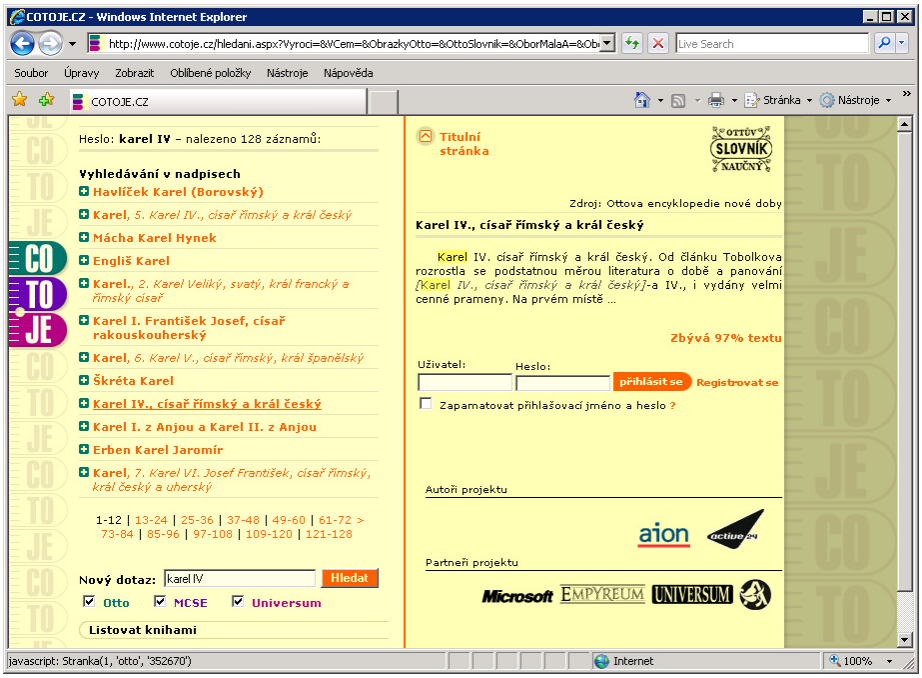
Obr. 40. Ukázka přístupu do databáze cotoje.kfbz.cz přes systém knihovny pro EIZ, databáze poskytuje plný text (100%) – uživatele identifikuje podle IP jako Krajskou knihovnu Františka Bartoše (viz. pravý horní roh).

ZÁVĚR V ANGLIČTINĚ

Objectives of work have been met. Have been established DNS aliases, has been configured and launched a web and proxy server. The whole system is at this time already in the active enjoyment of readers KKFB library.

In this work I used software licenses open source and freeware, ie. input costs of the project have been zero.

Configuration and other maintenance system is in my opinion simple and effective. The functionality of showing the pictures.



Pic. 41. Example access to the database cotoje.cz free from the Internet, the database will provide only part of the contents of information (remaining 97% of text).

Pic. 42. Example access to databases through the library cotoje.kfbz.cz for EIZ, the database provides full text (100%) - identifies the user by IP as a Regional Library František Bartoš (see right upper corner).

SEZNAM POUŽITÉ LITERATURY

Monografie:

[1] DOSTÁLEK, Libor, KABELOVÁ, Alena. *Velký průvodce protokoly TCP/IP a systémem DNS.* 1. vyd. Praha : Computer Press, c1999. 418 s. , 1 CD-ROM. ISBN 80-7226-193-2.

[2] KABIR, Mohammed J. *Apache Server 2 : kompletní příručka administrátora.* Brno : Computer Press, a.s., 2004. 724 s. ISBN 80-251-0319-6.

[3] STREBE, Matthew, PERKINS, Charles. *Firewally a proxy-servery : Praktický průvodce.* Libor Pácl; Lenka Hendrychová, Jakub Mikuláštík. 1. vyd. Brno : Vydavatelství a nakladatelství Computer Press, 2003. 442 s. ISBN 80-7226-983-6.

[4] TEIXERA, Steve, PACHECO, Xavier. *Borland Delphi : průvodce vývojáře - kniha 5-6.* [s.l.] : Knihy iDNES, 2002. 512 s. ISBN 80-86593-10-X.

Internetové zdroje:

[5] *WIKIPEDIE Otevřená encyklopedie* [online]. [cit. 2009-03-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Domain_Name_System>.

[6] *Pweb.cz Web o internetu a programování* [online]. [cit. 2009-03-10]. Dostupný z WWW: <http://www.pweb.cz/dns/zaznam-a.html>.

[7] *WIKIPEDIE Otevřená encyklopedie* [online]. [cit. 2009-03-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Webový_server>.

[8] *NETCRAFT Secure Server Survey* [online]. [cit. 2009-03-10]. Dostupný z WWW: <http://news.netcraft.com/archives/web_server_survey.html>.

[9] *WIKIPEDIE Otevřená encyklopedie* [online]. [cit. 2009-03-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Apache_HTTP_Server>.

[10] *WIKIPEDIE Otevřená encyklopedie* [online]. [cit. 2009-03-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Databáze>.

[11] *WIKIPEDIE Otevřená encyklopedie* [online]. [cit. 2009-03-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/SQL>.

[12] *WIKIPEDIE Otevřená encyklopedie* [online]. [cit. 2009-03-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Delphi>.

[13] *WIKIPEDIE Otevřená encyklopedie* [online]. [cit. 2009-03-10]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Proxy_server>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

.NET	DotNET (tečka NET; NETWORK = síť)
A	Adrress
Apache	A patchy server (Patchovaný server)
ASP	Active Server Page
BSD	Berkeley Software Distribution (Berkeley Unix)
CGI	Common Gateway Interface
CLX	Component Library for Cross Platform
DNS	Domain Name Server (Doménový názvový server)
EIS	Electronic(s) information source(s)
EIZ	Elektronický(é) informační zdroj(e)
FTP	File Transfer Protokol (Souborový přenosový protokol)
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IIS	Internet Information Services (Internetová informační služba)
IP	Internet protocol
JSP	Java Server Page
LAN	Local Area Network (Lokální síť)
Linux	Linus Unix
MD5	Message-Digest algorithm 5
MIME	Multipurpose Internet Mail Extensions
MS	Microsoft System
NAT	Network address translation (Překlad síťových adres)
PHP	Hypertext Preprocessor (dříve Personal Home Page)
RAD	Rapid Application Development

SQL	Structured Query Language (Strukturovaný dotazovací jazyk)
SŘBD	Systém řízení báze dat
SSI	Server Side Includes
SSL	Secure Sockets Layer (Vrsta bezpečných soketů)
Systray	System Tray (Systémový zásobník – prostor vedle hodin ve Windows)
TLD	Top Domain Level (Doména nejvyšší úrovně)
URL	Uniform Resource Locator
USB	Universal Serial Bus (Univerzální sériová sběrnice)
VCL	Visual Component Library
WAN	Wide Area Network (Rozšířená síť)
WEBDAV	Web-based Distributed Authoring and Versioning

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: ZDROJOVÝ KÓD PROGRAMU EXPIN.EXE

```
 aunit ExpPinRun;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, KpwSqlProcs, DB, IBCustomDataSet, IBQuery, IBDatabase,
  ShellApi, TLHelp32;

type
  TForm1 = class(TForm)
    IBDatab: TIBDatabase;
    IBRTransaction: TIBTransaction;
    IBRQuery: TIBQuery;
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

const
 PswFileName = '.htpasswd';
 PswPrgName ='htpasswd.exe';
var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.FormShow(Sender: TObject);
function IsProcessRun(ProcessName: string): boolean;

var
   Proc   : TprocessEntry32;
   Snap   : THandle;
   TempName: string;
   IsModule: boolean;
```

```
begin
   Result:= false;
   Snap := CreateToolHelp32SnapShot(TH32CS_SNAPprocess,0);
   Proc.dwSize := SizeOf(TprocessEntry32);
   process32First(Snap,Proc);
   repeat
//    IsModule:= GetProcessModule(Proc.th32ProcessID,
//                Proc.th32ModuleID, ^Proc, sizeof(MODULEENTRY32));
    if length(string(Proc.szExeFile)) > 0
     then tempname := proc.szExeFile
     else tempname := '< Unknown >';
    if tempname = ProcessName
     then Result:= true;
   until (not process32Next(Snap,Proc));
end;

var
 i, j, Rec, RecNo : integer;
 Usr, Psw, PomS: string;
 Fp: file;
 F: textfile;

begin
 IbServer:= ParamStr(1);
 IbName:= ParamStr(2);

 if OpenKpwSQLDb(IbDatab) then
  begin

   AssignFile(Fp, PswFileName + '_new');
   REWrite(Fp,1);
   CloseFile(Fp);

   AssignFile(f, PswFileName + '.bat');
   Rewrite(F);

   SetSQL(IbRQuery, 'select * from PIN where XCILEG <> 0 and USERNAME is
not NULL and PASSWD is not NULL');

   Rec:= 0; RecNo:= IbrQuery.RecordCount;
   while (not IbrQuery.Eof) do
```

```
    begin
     Usr:= Trim(IbrQuery['username']);
     Psw:= Trim(IbrQuery['passwd']);

     PomS:= PswPrgName + ' -mb ' + PswFileName + '_new "' + usr + '" "' +
psw + '"' + Chr(13) + Chr(10);
     Writeln(F, PomS);

     Inc(Rec);
     IbrQuery.Next;
     Caption:= 'Export PIN ' + IntToStr(Rec);
     Repaint;
    end;

   PomS:= 'del ' + PswFileName + '_old';
   Writeln(f,PomS);
   PomS:= 'copy ' + PswFileName + ' ' + PswFileName + '_old';
   Writeln(f,PomS);
   PomS:= 'del ' + PswFileName;
   Writeln(f,PomS);
   PomS:= 'copy ' + PswFileName + '_new ' + PswFileName;
   Writeln(f, PomS);
   CloseFile(f);

   IbDatab.Close;

//   WinExec(PChar(PswFileName + '.bat'), sw_normal);

   Close;
  end;
end;

end.
```