VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

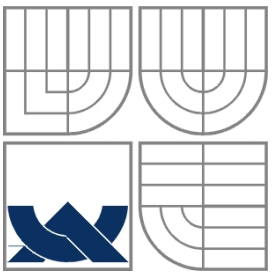
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

MODUL ANTIVIROVÉ KONTROLY PRO FIREFOX

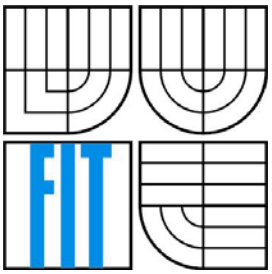
BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE Miroslav Ingr
AUTHOR

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

Modul antivirové kontroly pro firefox

Virus checking module for firefox

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE — Miroslav Ingr
AUTHOR

VEDOUCÍ PRÁCE — Peringer Petr, Dr. Ing.
SUPERVISOR

BRNO 2008

Zadání

1. Seznamte se s metodami pro testování přítomnosti škodlivého kódu v souborech a s implementací modulů pro Firefox.
2. Navrhněte modul antivirové kontroly pro Firefox s využitím programového rozhraní AVG API. Uživatelské rozhraní modulu musí umožňovat nastavení parametrů kontroly, aktivaci/deaktivaci a zobrazení základních informací o stavu AVG.
3. Implementujte navržený modul pro Firefox v prostředí operačního systému Microsoft Windows.
4. Výsledné řešení řádně otestujte a zhodnoťte jeho přínos.

Kategorie: **Uživatelské rozhraní**

Licenční smlouva

Abstrakt

Práce popisuje návrh a implementaci bezpečnostního rozšíření do internetového prohlížeče Firefox. V nástrojové liště se nachází komponenta, která kontroluje stažené soubory a aktivně komunikuje se softwarem AVG.

Klíčová slova

Firefox, anti-virus, anti-malware, AVG, XPCOM, XUL, JavaScript, C++

Abstract

Thesis describes the development of an anti-malware extension for the Firefox web browser. The component, located in a toolbar, checks downloaded files and communicates with the AVG software.

Keywords

Firefox, anti-virus, anti-malware, avg, XPCOM, XUL, JavaScript, C++

Citace

Ingr Miroslav: Modul antivirové kontroly pro Firefox, bakalářská práce, Brno, FIT VUT v Brně, 2008

Modul antivirové kontroly pro Firefox

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Dr. Ing. Petra Peringera. Další informace mi poskytl Ing. Karel Obluk, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Miroslav Ingr
10. května 2008

Poděkování

Hlavně děkuji za cenné informace při konzultacích Ing. Karlovi Oblukovi, Ph.D.
Také chci poděkovat Dr. Ing. Petrovi Peringerovi za vedení formální stránky bakalářské práce.

Obsah

1 Úvod

Bezpečnostní řešení v oboru informačních technologií je čím dál více aktuální téma. Na světovém trhu je mnoho společností zabývajících se touto tematikou. Mezi takové patří i firma AVG Technologies, s.r.o. (dále v textu zkráceně AVG), která se aktivně věnuje vývoji antivirového řešení. Kromě virové problematiky se zabývá internetovou bezpečností a všeobecně zabezpečení různých operačních systému pro PC. Díky komunikaci s Fakultou informačních technologií na Vysokém učení technickém v Brně vzniklo i zadání mé bakalářské spolupráce s AVG.

Práce se zabývá tvorbou bezpečnostního rozšíření pro Firefox. Výsledkem bude aplikace, která kontroluje stahované soubory, umožňuje nastavení parametrů testování a další možnosti. Při návrhu modulu pro Firefox budou používány moderní programovací metody a jazyky. Rozšíření je založeno na strukturované architektuře s popisem uživatelského rozhraní v jazyce XUL. Technologie XPCOM propojuje tuto aplikaci s binární knihovnou, která implementuje vlastní testování.

Úvodní kapitola popisuje přehled informací o bezpečnosti a použitých technologiích. Navazuje kapitola obsahující objektově orientovaný návrh komponenty a uživatelského rozhraní. Závěrečná kapitola je věnována implementaci projektu a jeho testování.

2 Základní informace

2.1 Zabezpečení operačního systému

Ne dlouho po rozšíření osobních počítačů, které měly původně sloužit k ulehčení náročných výpočtů a otravné stereotypní práce, se objevily stinné stránky moderní technologie. Zjištění, že dlouhé sezení za počítačem škodí lidskému zdraví, ani do dnešní doby nebylo zcela pochopeno a přijato lidmi. Nejprve se kazily oči od zářících monitorů, které se snažily omezovat přídavné radiační filtry. Nyní filtry vystřídala lepší technologie tekutých krystalů. Později přichází deformace páteře, způsobená sedavým zaměstnáním. Kromě zmíněných se přidávají psychologické změny - značná návykovost, gamblerství, odloučení od reality, …

Není divu, že se brzy projevily negativní účinky přímo na některých programátorech. Mění se jejich chování i postoj k počítačům, také k celé společnosti. Když Vám někdo ublíží, jak se zachováte? Já bych teda přemýšlel, jak mu to vrátit, pokud možno stejně nebo více. „Zlo plodí další zlo.“ Rychle vzniká první škodlivý kód, který touží upozornit každého uživatele. Ničivý kód označujeme jako virus. Trojský kůň je kód, který umí získat citlivé informace o uživateli. Rootkit umí zpřístupnit celý systém třetí straně. Nevyžádanou reklamu a poštu nazýváme spam.

Přechodem na elektronické uložení dat, ať už armádních či běžných uživatelů, se začínají hledat mezery v dobových technologiích. Informace jsou mocná zbraň. Přesná informace může rozhodnout o výsledku bitvy. Pro nepřátelské státy jsou pak klíčové informace o druhém cennější než zlato. Co řeknete na přístup k Vaší emailové schránce, k bankovnímu účtu? A co na přístup ke všem bankovním účtům jedné banky? Tak vzniká tzv. hacking a komunita lidí, kteří se hackingem zabývají. Původní hackeři byli nadšenci, jimž hledání bezpečnostních děr v systému přinášelo uspokojení. Pak už začínají snadno vydělávat na živobytí. Ti nejlepší pracují pro tajné služby jednotlivých mocností. Ostatní na vlastní pěst. Rozdělit je lze na kladné, neutrální a záporné.

Cenné informace se chránily kódováním již dávno před naším letopočtem. Šifrování dat je aktuální od první velké lidské říše. U počítačů se mluví o komunikaci více stran a sebelepší šifru nebude problém rozluštit, pokud se nežádaný pozorovatel dostane ke klíči, případně oběma klíčům. Nejen to, může ovládnout celý systém na dálku. V současné době poskytují operační systémy mnoho nástrojů a protokolů pro vzdálenou správu. Na odvetu se formují společnosti poskytující zákazníkům ochranu jejich počítačů, systémů a dat. Nejdříve bylo vše oddělené. Poslední dobou mají bezpečnostní programy snahu integrovat komplexní ochranu v jednom balíčku. Bezpečnostní sektor je velice dynamický, a proto je velmi důležitá aktuální databáze programu.

Komerční prostředí a tvrdá konkurence nadnárodních korporací prospívají v šíření strachu mezi lidmi, kteří nerozumí informačním technologiím do hloubky. Je tedy dost možné, že chamtivé subjekty v současné době nepřímo i přímo finančně podporují tvůrce virů a trojských koní. Každopádně existuje komunita, která tvoří škodlivý kód. Hlavně z osobních, finančních, politických a dalších důvodů. Mnoho aktuálních informací o zabezpečení počítačů najdete ve článcích [8].

2.2 Zabezpečení internetové komunikace

Internet je globální informační síť. Každý připojený počítač tvoří zároveň poskytovatele i příjemce informací, protože musí komunikovat se svým okolím. Některé informace jsou bezcenné, ale některé jsou výhodně zneužitelné. Hlavně se jedná o hesla uživatelů pro přístup k datovým i finančním úložištím. Kromě podobných záměrů je zde ještě hrozba napadení jednotlivých serverů. Důvody jsou takovéto: radost z dobytí pevnosti, poškození poskytovatele, informace o klientech, atd. Hackeři používají sofistikované metody a nástroje, které zcela zpřístupní nechráněný systém.

Software či hardware, který chrání otevřené komunikační kanály, se ustálil na pěkném názvu – firewall. Ohnivá zeď je propracovaná brána, ve které si uživatel nebo lépe zkušený správce sítě zvolí, která komunikace je povolená a zakázaná. Správně zkonfigurovaný firewall ještě nezaručuje uživateli dokonalou ochranu před hackery. Nemusí se totiž o díře v systému vůbec vědět. Firewall může také selhat po napadení zevnitř, hlavně po ovládnutí systému trojským koněm. Instalace rootkitu a ovládnutí cílového počítače je potom pro hackera již snadná záležitost. Dalším zajímavým pojmem je určitě nevyžádaná pošta, spam. Různé subjekty ji rozesílají hromadně mnoha cílovým adresátům. Šíří se tak hlavně reklama, potom viry a v neposlední řadě také žádosti hackerů o zadání důležitých hesel.

Dříve nebyl internet obecně rozšířený a viry se šířily infikováním spustitelných souborů. Soubor si pak půjčil kamarád a virus se replikoval na jeho systém. Kvalitnější exempláře takových virů měly dopad na světovou ekonomiku a rozšířily se do celého světa. Nyní se k této hrozbě přidává internet se svým rozsahem po celém světě. A tak si lze nový virus stáhnout ze serveru, z elektronické pošty či jinak. Zjednodušeně dnes již nestačí samotný antivirový program, bude se hodit firewall, antispyware a spamový filtr. Dohromady tyto jednotlivé programy tvoří balíček označovaný jako antimalware software.

2.3 Architektura prohlížeče Mozilla

Operační systém Windows je nejrozšířenějším komerčním produktem po celém světě. Má velmi kvalitní podporu ve vývoji. Má dvě nevýhody. Není kompatibilní s ostatními operačními systémy a uživatele stojí spoustu peněz. Naproti tomu dnes již kultovní internetový prohlížeč Mozilla Firefox, který je velice dobře přenositelný na většinu známých operačních systémů. Přitom si zachovává svůj

vzhled, funkčnost i strukturu. Užitečné informace jsou k nalezení v diplomové práci pana Koska [2], dále potom v sérii článků [1].

Projekt Mozilla je možné popsat jako velmi zajímavou snahu o otevřenou tvorbu počítačového software. Projekt se zaměřuje na prohlížení internetu. V minulosti byl známý univerzální a komerční internetový prohlížeč Netscape Navigator. V současné době se projekt diferencoval na konkrétní zaměření. Mezi nejznámější patří prohlížeč Mozilla Firefox, emailový klient Mozilla Thunderbird, plánovač Sunbird a další. Hlavní silou Mozilly je velká kompatibilita s různými systémy a snadná rozšiřitelnost. Toho dosahuje promyšlenou a kvalitně navrhnutou architekturou. Úsilí programátorů přineslo brzy své úspěchy. Mozilla se jako blesk rozšířila mezi uživateli operačního systému Linux a MacOS. Po příchodu Firefoxu a Thunderbirdu se prosazuje i v Microsoft Windows.

Uživatelské rozhraní je malá specialitka, protože je popsáno v samostatných souborech za pomoci jazyka XUL. Tento jazyk umožňuje uživateli poměrně snadno měnit vzhled jak samotného prohlížeče, tak všech jeho doplňků. Uživatel tak získává nástroj pro tvorbu vlastních programů založených na společném projektu Mozilla. Ostatní jazyky používají při vytváření programů grafické knihovny. Tyto knihovny poskytují programátorovi prvky uživatelského rozhraní, například textové pole, obrázek, tlačítko, okno dialogu a jiné. Pro Linux je to knihovna Gtk, ve Windows se používá MFC. Hned na první pohled si zkušený vývojář všimne nevýhody takového řešení. Je jím velmi omezená přenositelnost. Knihovnu MFC podporuje pouze Microsoft Windows, balíky Gtk či Qt pak jen distribuce Linux a částečně také MacOS a Windows. Jmenovat lze multiplatformní wxWidgets. Tvůrce má tak možnost s mírnými problémy obsadit několik málo operačních systémů.

Mozilla však zahrnuje ve svém řešení tyto platformy: Windows XX, Windows XP, Windows Vista, různé distribuce Linux, MacOS, MacOS X, FreeBSD, NetBSD, BSD/OS, OS/2, Solaris, AIX, BeOS, OpenVMS, HPUX, Tru64 Unix. Programátor běžných vývojových jazyků bohužel nemá k dispozici tak univerzální prostředek, aby pokryl všechny jmenované systémy. S velkou trpělivostí by si musel takovou knihovnu vytvořit sám. A tak postupovali i tvůrci Mozilly. Problém kompatibility dal vzniknout velmi flexibilnímu jádru Gecko, na kterém si celý projekt Mozilla zakládá. Jádro interpretuje kód HTML a XHTML do vizuální formy navíc s kaskádovými styly CSS. Nesmí chybět funkční kód, proto jádro interpretuje bloky kódu popsané v jazyce JavaScript. Dnes se bez takového základu neobejde žádný internetový prohlížeč.

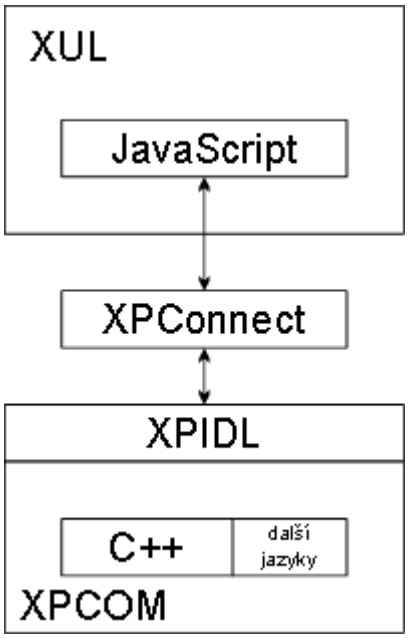
Mozilla se odlišuje od konkurence právě další vlastnost jádra Gecko. Jedná se o interpret jazyka XUL, ve kterém se popisuje celé uživatelské rozhraní od oken po tlačítka a další prvky. Jazyk umožňuje kromě popisu základního vzhledu také jeho dědičnost do dalších rozhraní programu. Jazyk XUL přímo nemusí specifikovat vzhled rozhraní. Rozložení, velikosti, barvy a jiné lze popsat pomocí kaskádových stylů v samostatném CSS souboru. Výsledný vzhled se nazývá skin, který lze použít v celé aplikaci nebo alespoň částečně v konkrétní komponentě. Celá aplikace a moduly pak mají (ale nemusí mít) ucelený a jednotný vzhled.

Jazyk XUL je odvozený z jazyka XML. Jednotlivé prvky uživatelského rozhraní jsou zapsány stejně jako XML elementy. Jejich vzhled popisují kaskádové styly nebo vlastnosti elementů. Stejně jako jejich vlastnosti, které jsou odvozeny od XML atributů. Uživatelské rozhraní může být definováno i vícenásobně, uživatel si pak snadno zvolí optimální nastavení. Jazyk XUL sám o sobě neposkytuje žádnou funkčnost naší aplikaci, pouze definuje vzhled. A na co je uživateli vzhled, když neplní jeho požadavky?! Proto jádro Gecko interpretuje funkční jazyk JavaScript, který snadno zvládne obsloužit jednodušší události. Například při stisku tlačítka je možno definovat vlastní obslužnou rutinu MyOnClick(), která vyvolá patřičnou reakci.

Nutno zmínit, že na kombinaci XUL-JavaScript je postavena celá Mozilla. Není proto příliš rychlá a výkonná, ale za to přenositelná a rozšiřitelná. Naneštěstí není JavaScript vhodný při tvorbě složitějších a komplexních aplikací, které vyžadují výpočetní výkon, či interakci s konkrétním operačním systémem. Prostě se v některých případech nelze vyhnout přednostem jazyků C++, C#, Delphi a jejich knihovnám. Nebo obecně se nelze vyhnout dynamicky linkovaným knihovnám, modulům, které poskytují svůj kód, často závislý na konkrétní platformě. Na první pohled je zřejmé, že použití obecného nativního jazyka je složité, protože je kompilován do binárního souboru závislého na konkrétní platformě, tedy není příliš univerzální.

Dynamické knihovny využijeme hlavně při tvorbě vlastních rozšíření. Vývojáři Mozilly s tím počítali a ta poskytuje pokročilé technologie propojující JavaScript s nativním kódem. Jedná se o tzv. XP-technologie. Základní model přenositelného binárního kódu se jmenuje XPCOM. Kódu psanému v nativním jazyce se přidá univerzální XPIDL obálka, která poskytuje nativně nezávislé rozhraní. XPCOM komponenta je pak schopna poskytovat své služby přes XPIDL rozhraní pomocí spojení XPConnect všem dalším komponentám, které poskytují rozhraní XPIDL. Systém Windows používá podobné technologie se jménem COM. Kód nativního jazyka je sice vždy nutné přeložit pro konkrétní systém, ale volání je stále stejné a univerzální.

XPCOM (cross-platform component object model) je funkční komponenta psaná v nativním jazyce (C++, C#, Delphi, …) využívající rozhraní XPIDL. XPIDL (cross-platform interface definition language) popisuje rozhraní pro XP komponenty. Umožňuje generování speciálních hlavičkových a deskriptivních souborů. Vychází z obecnějšího jazyka IDL (interface definition language), který se používá v obdobných případech. XPConnect oboustranně propojuje interpretovaný kód JavaScriptu s komponentami XPCOM. Přehledně je XP technologie zachycena na Obrázku 1.



Obrázek 1 Architektura XP technologie

Každý produkt vzešlý z projektu Mozilla přebírá tyto základní vlastnosti. Uživatel si může zvolit svůj vzhled aplikace pomocí skinů, nebo si ho sám naprogramovat. Program lze rozšířit externími zásuvnými moduly (plugins) a rozšířeními (extensions). Mezi klasické pluginy patří podpora multimédií Flash, QuickTime, RealTime a aplikací Java. Obecně lze říci, že plugin přináší do Mozilly podporu různých formátů.

Rozšíření se pak snaží doplnit prohlížeči chybějící funkcionalitu, vylepšit jeho vzhled a přizpůsobit produkt konkrétnímu uživateli. Toto posunuje Mozillu o třídu výše od ostatních prohlížečů, neskutečná podpora platforem pak ještě o další třídu.

Protože jádro Mozilly interpretuje několik jazyků zároveň, samo zabírá v paměti mnoho prostoru a zpomaluje chod aplikace. Uživatel s moderním počítačem však rozdíl ani nepocítí. Uživatel ocení přizpůsobitelnost programu vlastnímu pohodlí díky zásuvným modulům a rozšířením. Na oficiálních stránkách Mozilly jsou k dispozici stovky různých doplňků, které si uživatel může lehce instalovat přímo do programu. Jistě se můžou hodit filtry reklamy, předpověď počasí či aktuální zprávy. Stejným způsobem lze přizpůsobit formální vzhled Mozilly pomocí skinů, které jsou rovněž ke stažení na oficiálním webu.

2.4 Jazyk XUL

XUL (XML user-interface language) vychází z jazyka XML, stejně jako ten používá entity, elementy (prvky), atributy (vlastnosti prvků) a další. Na rozdíl od obecného XML je jazyk XUL zaměřen na popis uživatelských rozhraní. XUL byl stvořen, aby usnadnil a zrychlil vývoj prohlížeče Mozilla. Z jazyka XML dědí všechny vlastnosti a konkretizuje další specifikaci. Většina aplikací je vyvíjena na cílové platformě. Převést takový software do multiplatformní podoby pak stojí spoustu času a peněz. V minulosti bylo vyvinuto několik multiplatformních řešení. Například Java, která zakládala na přenositelnosti. XUL právě tak specificky navržený pro tvorbu přenositelných uživatelských rozhraní. Důvod je jednoduchý. Tvorba takové aplikace zabírá mnoho času. Je neskutečně složité takovou aplikaci ladit, aby fungovala univerzálně na všech platformách. Díky jazyku XUL je implementace mnohem snazší a rychlejší.

XUL má výhody jiných XML jazyků. Například XHTML nebo jiné jako MathML či SVG mohou být použity v kódu. Všechny texty jsou snadno lokalizovatelné, což znamená, že mohou být přeloženy do několika světových jazyků nezávisle na prvcích a vzhledu uživatelského rozhraní. Jazyk umožňuje vytvoření většiny prvků moderních grafických rozhraní. Mezi tyto prvky patří:

- Vstupní ovladače jako textová pole či zaškrtávací tlačítka
- Nástrojové lišty s tlačítky a dalším obsahem
- Uživatelská a vyskakovací menu
- Záložky
- Hierarchické stromy a tabulky
- Klávesové zkratky

Zobrazovaný obsah je popsán v souboru nebo v jiném zdroji dat. Pro Mozillu to může být třeba poštovní schránka uživatele, výsledky hledání nebo oblíbené odkazy. Obsah menu, stromových struktur a jiných elementů může být spojen s těmito daty, nebo s vlastními informacemi uloženýma ve speciálním RDF souboru. Programátor má možnost několika typů XUL aplikací, viz Tabulka 1.

Rozšíření Mozilla	Rozšíření funkcionality prohlížeče Firefox
Samostatná XULRunner aplikace	XULRunner je zabalená verze Mozilly, která dovoluje spouštění nezávislých aplikací. Přitom není vyžadován prohlížeč, protože takovéto programy jsou samostatně spuštěné.
XUL balíček	Směsice předchozích dvou možností. Jedná se o samostatné aplikační okno spuštěné přímo v prohlížeči.
Vzdálená XUL aplikace	Samozřejmě můžete umístit svůj kód na webový server a používat ho v prohlížeči. Bohužel je tato metoda dost omezená, protože by jinak umožňovala bezpečností porušení jako otevírání nových oken.

Tabulka 1 Typy XUL aplikací

První tři možnosti jsou instalovány jako lokálním počítači. Tyto typy aplikací nemají žádná bezpečnostní omezení. Takže mohou třeba přistupovat k souborům, zapisovat do nich a číst. V rozšířeních jsou většinou XUL soubory zabaleny společně s přiloženými skripty a dalšími soubory. Tento balíček si potom může uživatel stáhnout a nainstalovat. Mozilla poskytuje správce rozšíření, který tvůrčí práci značně usnadňuje. Není třeba psát rozsáhlé instalační skripty, stačí jen opravdu základní údaje o aplikaci.

2.5 AVG API

Program AVG nabízí komunikační rozhraní pro interakci s okolními nástroji (pomocí technologie COM). Poskytuje tak prostředky pro tvorbu vnějších aplikací, které mohou využívat služeb AVG a jeho technologií. AVG obsahuje tyto prostředky:

- vytvoření komunikačního kanálu
- antivirové testování souborů
- metody reakce na nalezenou infekci
- nastavení některých parametrů programu
- vnitřní AVG slovník
- informace o programu

3 Návrh bezpečnostního rozšíření

Při výběru jména projektu jsem použil pracovní jméno AVFoxy. První dvě písmena „AV“ jsou zkratkou slova anti-virus a „Foxy“ vzniklo změkčením slova „Fox“. Slovo „Firefox“ znamená v českém jazyce „panda červená“. Svým vzhledem připomíná lišku, která je ve znaku prohlížeče Mozilla Firefox. Jméno AVFoxy jsem přisoudil XPCOM komponentě, ale pro nástrojovou lištu byl použit jednotný komerční název AVG SecurityBar. Proto se uživatel s pojmem AVFoxy ani nesetká.

3.1 Základní funkce

Základní funkcí mého rozšíření je kontrola stažených souborů pomocí internetového prohlížeče Mozilla Firefox. Za pomoci profesionálního software AVG 7.5 je možné tuto kontrolu zajistit. Přeci jenom se viry šíří nejvíce po internetu. Když uživatel zatouží stáhnout infikovaný soubor, projde tento nejdříve kontrolou a v případě pozitivního nálezu musí být upozorněn. Zároveň musí mít uživatel jednoduchou možnost jednat.

Vizuální částí AVFoxy je nástrojová lišta, ikonka a menu v hlavní nabídce programu Firefox. Uživatel si může zvolit, co chce používat. V případě, že ho nástrojová lišta ruší, může používat malou ikonku v rohu. Nebo úplně vizualizaci vypnout a používat jen hlavní menu. Důležité bylo si předem dobře rozmyslet, které části bude obsahovat binární XPCOM komponenta, a který kód lze jednoduše zprostředkovat pomocí jazyků XUL a JavaScript. Rozdělení projektu do cílových jazyků je zobrazeno v Tabulce 2.

Inicializace komunikace s AVG	XPCOM
Inicializace XPConnect rozhraní	JavaScript
Ukončení a uvolnění paměťových prostředků	XPCOM
Komunikace s AVG	XPCOM
Naslouchání událostí v programu Mozilla Firefox	JavaScript
Antivirová kontrola souborů	XPCOM
V případě nalezené infekce varování uživatele	JavaScript
Provedení příslušné reakce na infekci	XPCOM
Zjištění informací z AVG	XPCOM
Uživatelské rozhraní	XUL
Chování UI	JavaScript

Tabulka 2 Rozdělení projektu do cílových jazyků, resp. technologií

3.2 XPCOM komponenta

Komponenta je dynamicky linkovaná knihovna, která poskytuje své služby. Pro vývoj takové knihovny je vhodný jazyk C++, který je zároveň kompatibilní s vývojovými prostředky v programu AVG. Komunikační roura AVG API zajišťuje dostatečné prostředky, které komponenta využije ke kontrolování souborů a také dalším funkcím. V těchto případech je nevyhnutelné psát kód v nativním jazyce, protože jazyk JavaScript neposkytuje žádné jiné možnosti, jak volat rozhraní AVG API.

Pokud chceme kompatibilitu s programem Firefox, bude třeba implementovat rozhraní XPIDL. Po kompilaci získá programátor binární knihovnu, kterou přiloží ke svému rozšíření. Cílová architektura XPCOM komponenty je Microsoft Windows, ale vzhledem k jejímu charakteru je možné ji přeložit i pro Linux a další. U Linuxu však mohou nastat problémy s komunikací se softwarem AVG, který je přednostně určen pro Windows.

3.2.1 Možnosti AVG API 7.5

Technologie XPCOM se používá v programu, protože umožňuje propojení s běžnými potřebami programátora. V případě AVFoxy bude hlavní výhodou spojení s rozhraním AVG API, které poskytuje potřebné prostředky pro anti-virovou kontrolu. Následující informace jsem čerpal z dokumentace [7]. Díky API může program AVG komunikovat se svým okolím a naopak. Celé prostředí je postaveno na technologii COM. Verze 7.5 poskytuje následující služby programu AVG:

- Inicializace rozhraní COM – AVG API
- Uvolnění rozhraní COM – AVG API
- Kontrola stavu licence AVG
- Získání verze programu AVG
- Kontrola souboru
- Kontrola datového proudu
- Získání a nastavení základních nastavení
- Zjištění používaného jazyka
- Překlad textů dialogu
- Od/Registrace vlákna
- Nastavení rezidentního štítu

Manipulace s infikovaným souborem:

- Přesun do virového trezoru
- Čištění virové infekce, pokud je to možné
- Ignorování infekce

3.2.2 Antivirová kontrola

Uživatel začíná stahovat soubor. Je třeba zachytit událost v momentu dokončeného stahování. Rozhraní Mozilla toto umožňuje použitím tzv. observeru, kterým je sledované rozhraní přiřazeno i s konkrétní událostí. Každému observeru lze přiřadit i více událostí zaráz. Každá událost s sebou nese důležité informace. Díky nim předáme cestu k souboru AVFoxy komponentě, která zajistí potřebné propojení se softwarem AVG. Výsledek kontroly se pak vrací zpět z komponenty do naší aplikace, běžící v prostředí Mozilla Firefox. Výsledek může mít následující hodnoty:

- Žádná známá infekce, soubor je v pořádku
- Nalezená infekce
- Poškozený soubor
- Nepřístupný soubor
- Málo paměti
- Software AVG není nainstalován, nelze zkontrolovat

V případě nalezené infekce musí být uživatel vhodně obeznámen. Malé dialogové okno je správné řešení. Upozorní na infikovaný soubor, ohlásí textovou formou všechny známé informace o viru a souboru. Pomocí několika tlačítek si může uživatel vybrat vhodnou reakci:

- Přesunout soubor do virového trezoru
- Pokusit se vyčistit infekci ze souboru, tato možnost není vždy reálná, proto se objevuje pouze v případě, kdy je možná.
- Vymazat soubor
- Ignorovat infekci

Musíme počítat s variantou více stahovaných souborů. Nelze použít více vláken, protože by to bylo výpočetně i paměťově příliš náročné řešení. Druhým řešením je použití fronty, zásobníku či obecně lineárního seznamu. Seznam dostatečné velikosti uloží aktuální roury kontrolovaných souborů. Jakmile se uživatel rozhodne pro akci, například přesun souboru do virového trezoru, je soubor přesunut a odkaz v seznamu uvolněn z paměti. Seznam je tak možné udržovat stále volný pro další infekce. Přetečení seznamu způsobuje implicitní reakci systému na vzniklou událost.

3.2.3 Informace o AVG

V dialogovém okně O programu se uživatel může dozvědět zajímavé informace o programu AVG a také o stavu jeho licence. Také se může stát, že licence vyprší a uživatel by měl být o tomto stavu obeznámen a funkčnost aplikace omezena. Rozhraní AVG API nabízí metody, jak zjistit aktuální verzi programu, virové databáze a datum vydání databáze. Dostaneme typ uživatelské licence:

- Neplatná
- Volná
- Zkušební
- Zlevněná
- Plná

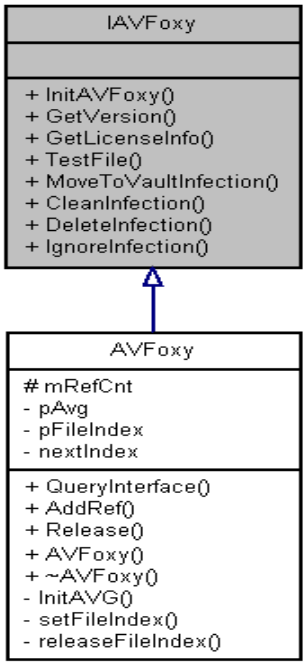
Poslední metoda zjišťuje stav uživatelské licence:

- Neplatná
- Neplatné datum
- Platná
- Varování o vypršení platnosti
- Platnost vypršela
- Umrtvená licence

Všechny tyto informace lze získat díky AVG API a poskytnout v AVFoxy rozšíření, kde je má uživatel snadno dostupné přímo ve svém internetovém prohlížeči.

3.2.4 Diagram tříd

Z výše zmíněných podkapitol lze navrhnout třídu AVFoxy, která bude odvozena z rozhraní IAVFoxy. V kapitole 4.2 se blíže popisuje implementace takové třídy. Rozhraní IAVFoxy musí být definováno v jazyce IDL, ze kterého je později možné generovat hlavičkové soubory a typovou knihovnu. Tu bude potřebovat aplikace Firefox ke spojení s dynamickou knihovnou DLL. Třída musí poskytovat všechny potřebné metody pro inicializaci a destrukci objektu. Dále metodu pro inicializaci komunikace s programem AVG, testování souborů a činnost při zjištěné infekci. Nakonec nesmí chybět metody odvozené z rozhraní Mozilly postavené na technologii XP, které zaobalují celou třídu přenositelností kódu.



Obrázek 2 Digram tříd

3.3 Uživatelské rozhraní

Tvorba uživatelského rozhraní podléhá několika fázím vývoje. Nejdříve se pracuje s určitým konceptem na papíře. Mění se podle názoru vývojáře o výsledném vzhledu produktu. Při implementaci se pak mění detaily, pouze výjimečně celý vzhled. Rozšíření pro Firefox popisuje svůj vzhled jazykem XUL, který dává designérovi do rukou mocný nástroj. Jednotlivé prvky jsou do sebe postupně vnořovány jako při klasické tvorbě webové prezentace v HTML. Nejdříve se popisuje vzhled celého prohlížeče a jeho součástí, následně třeba nástrojový panel, do kterého vložíme lištu s tlačítky a obrázky.

Hlavní menu

Hlavní menu by nijak nemělo rušit uživatele. Mělo by však být dostupné.

Hlavní menu -> Nástroje -> *jméno komponenty* -> *menu komponenty*

Uživatel si volí v menu základní volby myší i klávesnicí. Mezi základní volby patří:

- Nastavení
- Vypnout/Zapnout komponentu
- Statistiky
- Informace o programu

Ikona

Jak upozornit uživatele, že má nainstalován tento produkt, poskytnout mu celou funkcionalitu, ale přitom ho neobtěžovat rozměrnou lištou? Použil jsem jednoduchou stavovou ikonku loga AVG, která v sobě ukrývá prvky výše zmíněného uživatelské menu. Ikonka mění svůj vzhled podle stavu komponenty, jako v Tabulce 3.

Zapnutá	barevné logo programu AVG
Vypnutá	šedé logo AVG
Pokažená	šedé a červeně přeškrtnuté logo AVG

Tabulka 3 Stav aplikace AVFoxy

Uživatel na první pohled vidí, v jakém stavu je tato lišta a také program AVG.

Nástrojová lišta

Nástrojová lišta je akorát rozšířená ikonická lišta s objemnějším vzhledem. Snaha se stále stáčí směrem o co nejmenší rušivý efekt pro uživatele. Obsahuje:

- Ikonu stavu aplikace
- Název *AVG SecurityBar*
- Tlačítko *Pokročilé*

Ikona neobsahuje menu, ale informuje uživatele o stavu aplikace stejně jako ikonická lišta. Ploché tlačítko s názvem lišty poskytuje odkaz na lokální webové stránky společnosti AVG Technologies. Nakonec tlačítko *Pokročilé* skrývá zmíněné menu.

Virová výstraha

Po stažení infikovaného souboru musí být uživatel včas varován. K dispozici jsou nutné informace o infekci, aby se mohl člověk vhodně rozhodnout, jak reagovat:

- Přesunout soubor do virového trezoru
- Vyčistit infekci
- Vymazat soubor
- Ignorovat infekci

Možnost vyčištění infekce, nebo lépe vyléčení infekce není vždy k dispozici, protože je tato funkce v mnoha případech nedostupná a lepší je infikovaný soubor přesunout do trezoru či smazat. Programátor si předem zjistí, zda je reálné soubor léčit, a pak už jen volbu nabídne nebo vůbec nezobrazí, aby nemátla uživatele.

Uživatel se přehledně dozví všechny užitečné informace. Zobrazuje se jméno souboru, jméno infekce, popis infekce a nabízená reakce. Virová výstraha není běžné upozornění, ale dialogové okno se speciálními parametry:

- Centrované
- Vždy nahoře
- Nelze odklepnout bez interakce uživatele
- Dědí vzhled aplikace AVFoxy pro Firefox, nepoužívá tedy knihoven Windows

Nastavení

Běžná nastavení jsou k dispozici v každém vyspělejším programu. Lze tak snadno přizpůsobovat parametry aplikace přáním uživatele. První volba je zapnutí a vypnutí komponenty:

- Zapnuta
- Vypnuta

Například implicitní akce pro infikovaný soubor může usnadnit uživateli rozhodování a nemusí ho rušit varovné okno. Uživatel se dozví nenásilnou formou o provedené akci. Typ implicitní akce s infekcí si můžeme zvolit právě v okně nastavení:

- Přesunout do trezoru
- Léčit
- Smazat
- Vždy se ptát (Virová výstraha)

Třetí volba je vzhled aplikace, uživatel má možnost si vybrat tak, aby ho neobtěžovala, nebo naopak chce vidět celou lištu. Volby jsou takovéto:

- Jednoduché menu
- Ikona
- Nástrojová lišta

Nesmí chybět tlačítko *Potvrdit* a *Storno*. Potvrzením se volby uloží, program patřičně změní své chování a okno nastavení je uzavřeno. Stornováním se okno nastavení také uzavírá, ale nic se nemění oproti předchozímu nastavení.

Statistiky

Během provozu AVFoxy lze jednoduše zaznamenávat podstatné události a ukládat tak výslednou statistiku. Pro uživatele mají význam pouze tyto dvě hodnoty:

- Celkový počet stáhnutých souborů
- Počet nalezených infekcí

Zaznamenat lze však i další vlastnosti programu, například počet hodin strávených na internetu, doba provozu AVFoxy, kvantum stažených dat a další. Z funkčního hlediska však nemají pro uživatele význam. Potvrzovacím tlačítkem se okno statistiky uzavře. Resetovacím tlačítkem se vynulují hodnoty do počátečního stavu.

Informace o programu

Základní informace je dobré přehledně roztřídit do tabulky. Kromě informací je užitečné i logo společnosti AVG a majoritní verze programu, který AVFoxy podporuje. První nás zajímají informace o programu AVG:

- Verze AVG
- Verze virové databáze
- Datum vydání virové databáze

Z komerčních důvodů je nevyhnutelné podat údaje o aktuální licenci pro software AVG:

- Stav licence
- Typ licence
- Počet počítačů, které licence zahrnuje
- Konec platnosti licence

Když je licence těsně před vypršení své platnosti, může být takto uživatel obeznámen o tomto stavu. Dále se dozví, zdali je vůbec jeho licence platná. Pokud není licence platná, aplikace AVFoxy samozřejmě nefunguje, tak se dozví uživatel o chybě první. V informacích si pak může dohledat rozšiřující zajímavosti a podívat se na dobu platnosti své licence. Nemělo by chybět licenční ujednání z právního hlediska.

4 Implementace

V této kapitole se dozvíme alespoň částečný popis tvorby rozšíření pro Mozilla Firefox. Díky homogenní architektuře celého projektu Mozilla je tvorba rozšíření pro Firefox, Netscape Navigator Thunderbird i Sunbird velmi podobné. Na začátek uvedu základní náležitosti při vývoji takové aplikace. Uvedené informace předpokládají znalosti objektově orientovaného programování v C++ a JavaScript. Komponentu AVFoxy jsem vyvíjel v prostředí Microsoft Visual Studio, při ladění skriptů jsem užíval debugger FireBug, chybovou konzolu Firefoxu a intuitivní přístup.

4.1 Adresářová struktura

Rozšíření do Firefoxu tvoří jakýsi balíček s očekávanou strukturou. Balíček se skládá z adresářového stromu a souborů. Rozšíření obvykle mění uživatelské rozhraní Firefoxu nazývané *chrome*. Dále mění jeho vzhled, nejčastěji tzv. rámečkem, kterým může být dialogové okno, lišta, tlačítko, menu a mnoho dalších. Tyto rámečky jsou vlastně obsahem balíčku – *content*. Většina rozšíření má několik vzhledů *skin* a vícejazyčnou podporu – lokalizace *locale*. Evidentně je důležitá organizace projektu a hlavně smysluplné názvy klíčových souborů. Některé soubory jsou speciální, hlavně s příponou *rdf*, protože se v nich popisují data a jejich struktura. Vysvětlení poslání každého souboru leží mimo rámec této práce.

Mozilla používá adresářovou architekturu v celém projektu. Má totiž své jednoznačné přednosti. Všechny soubory a adresáře lze snadno adresovat pomocí URL. Relativní adresování ve jmenném prostoru *chrome://* tak nevyžaduje znalost přesného umístění souboru, viz Kód 1.

```
chrome://extension/skin/picture.png
```

Kód 1 Příklad adresování obrázku

Projekt AVFoxy používá adresářovou strukturu z Tabulky 4. Název ukončený lomítkem označuje adresář, název ukončený znakem „*“ je binární soubor, znak „!“ označuje archivovaný soubor a bez lomítka jsou textové soubory.

avfoxy/	hlavní adresář
chrome.manifest	organizace souborů
install.rdf	instalační informace
components/	XPCOM komponenty
AVFoxy.dll*	dynamicky linkovaná knihovna
IAVFoxy.xpt*	typová knihovna komponenty
defaults/	základní nastavení
preferences/	adresář nastavení
avfoxy.js	popis proměnných nastavení
chrome/	Firefox základní složka
avfoxy.jar!	JAR strukturovaný balíček
content/	obsah aplikace, zdrojové texty
avfoxy.js	hlavní funkce AVFoxy
avfoxy_about.js	skript informací
avfoxy_about.xul	okno informací
avfoxy_alert.js	skript výstrahy
avfoxy_alert.xul	okno výstrahy
avfoxy_observers.js	základní observery
avfoxy_overlay.xul	vzhled aplikace AVFoxy
avfoxy_settings.js	skript nastavení
avfoxy_settings.xul	okno nastavení
avfoxy_stats.js	skript statistiky
avfoxy_stats.xul	okno statistiky
commons.js	definice základních struktur
contents.rdf	popis obsahu složky
locale/	lokalizace rozšíření
en-US/	anglický jazyk
content.rdf	popis obsahu složky
dynamics.properties	dynamická lokalizace
statics.dtd	statická lokalizace
skin/	vzhledy rozšíření
classic/	klasický vzhled
avfoxy_statusbar.css	kaskádový styl statusbaru
avfoxy_toolbar.css	kaskádový styl nástrojové lišty
avg_broken.png*	obrázek pokažené aplikace
avg_disabled.png*	obrázek vypnuté aplikace
avg_enabled.png*	obrázek zapnuté aplikace
avg_logo.png*	logo AVG
avg_logo_medium.png*	větší logo AVG
contents.rdf	popis obsahu složky

Tabulka 4 Adresářová struktura

4.2 AVFoxy komponenta (XPCOM)

Doporučuji si nainstalovat prohlížeč Firefox s ladícím režimem. Potom si vytvoříme běžný uživatelský profil s běžnými doplňky a nastavením. K běžnému profilu přidáme druhý vývojový, ve kterém celou aplikaci vytvoříme a odladíme. Na internetu si stáhneme užitečné vývojové rozšíření:

- Console^2 – vylepšená chybová konzole
- FireBug – ladící nástroj pro JavaScript
- Mozilla's extension developer – multifunkční nástroj pro tvorbu rozšíření

XPCOM komponentu AVFoxy vyvíjím jako projekt Microsoft Visual Studio 2005. Samozřejmě lze použít jakýkoliv kompilátor jazyka C++ s přiloženými knihovnami AVG, Firefox a Windows. Všechny potřebné knihovny a utility Mozilla jsou dostupné ve vývojovém prostředí Gecko-SDK.

4.2.1 Úvod do problematiky rozšíření Firefox a XPCOM

Výsledkem našeho snažení bude instalační balíček XPI, který je již plně samostatný a instaluje se přímo do prohlížeče Mozilla Firefox. Jedná se o komprimovaný soubor ZIP s příponou *xpi*. V kapitole 0 se probírá konkrétní adresářový strom projektu AVFoxy. Vývoji vzhledu a chování se věnuje kapitola 4.3. Na internetu je ke stažení komplexní příručka [6], která popisuje detailně postup tvorby ukázkové aplikace.

XPCOM komponenta je dynamicky linkovaná knihovna DLL, tedy binární soubor. Aby mohla plnit svou úlohu v našem rozšíření, musí implementovat rozhraní technologie XPIDL a exportovat popis svých služeb ve speciálním souboru typové knihovny XPT. Na začátku si definujeme IDL soubor, kde popíšeme služby komponenty jazykem IDL (interface definition language), viz Kód 2.

```
#include "nsISupports.idl"
[scriptable, uuid(…)]

interface IAVFoxy : nsISupports  {
  PRBool InitAVFoxy();
  …
};
```

Kód 2 IDL rozhraní

Takto popsané rozhraní počítá s rozhraním Firefox (nsISupports.idl), nutně musí být unikátní díky vygenerovanému uuid. Pak zapisujeme vlastní rozhraní IAVFoxy a jeho metody dle specifikace jazyka IDL. Nakonec ze souboru snadno vygenerujeme hlavičkový soubor *IAVFoxy.h* a typovou knihovnu *IAVFoxy.xpt* pomocí překladače *XPIDL.exe* v sadě Gecko-SDK. Typová knihovna se přikládá k binární knihovně DLL, jinak nebude zajištěno spojení XPCONNECT s Firefoxem.

Hlavičkový soubor *IAVFoxy.h* obsahuje již vygenerované veřejné metody a informace, jak popsat třídu AVFoxy, která implementuje rozhraní IAVFoxy. Uděláme to ve vlastním hlavičkovém souboru *AVFoxy.h:*

```
#define AVFOXY_CID { … }       // originální UUID třídy AVFoxy, jiné než rozhraní IAVFoxy
/* Hlavičkový soubor */
class AVFoxy : public IAVFoxy    {
public:
    NS_DECL_ISUPPORTS
    NS_DECL_IAVFOXY
    AVFoxy();                // Konstruktor
    virtual ~AVFoxy();       // Destruktor
};
```

Kód 3 Definice hlavičky

Všimněte si kódu NS_DECL_ISUPPORTS a NS_DECL_IAVFOXY. Jedná se o předdefinovaná makra rozhraní nsISupports, která programátorovi ulehčí práci s psaním opakovaného kódu. Toto rozhraní poskytuje mnoho dalších užitečných maker, která můžeme využít ve svých programech, velmi si tak usnadnit práci a zorientovat kód na podstatné věci. Abych vynechal zdlouhavý text zdrojových souborů, uvádím jen příklady dalších užitečných maker v Tabulce 5.

NS_IMPL_ISUPPORTS1(AVFoxy, IAVFoxy)	*implementace rozhraní*
NS_IMETHODIMP AVFoxy::InitAVFoxy(PRBool *_retval)	*implementace metody*
NS_GENERIC_FACTORY_CONSTRUCTOR(AVFoxy)	*předání konstruktoru třídy*
NS_IMPL_NSGETMODULE("AVFoxyModule", components)	*registrace modulu*

Tabulka 5 Užitečná makra

Více informací o konkrétních záležitostech tvorby XPCOM se dočtete na vývojovém webu MDC. Všechny uvedené makra pouze zrychlují formální stránky vývoje XPCOM komponenty, sami žádnou užitečnou práci neplní, pouze zajistí komunikaci komponenty s aplikací Firefox. Pro správnou identifikaci komponenty je nevyhnutelné správně definovat jedinečné identifikační 128bit číslo UUID a jméno kontraktu, například:

```
#define CONTRACTID "@avg.com/avg/avfoxy;1"
```

Díky tomuto řetězci jsme schopni snadno identifikovat naši komponentu, jak brzy uvidíme. Inicializace rozhraní se provádí v klientské části (JavaScript), kde Firefox poskytuje potřebné metody. Inicializace rozhraní IAVFoxy může vypadat jako Kód 4.

```
// Vytvoří instanci AVFoxy DLL komponenty
myFoxy = Components.classes["@avg.com/avg/avfoxy;1"].createInstance();
if(myFoxy) {
    // Instance vytvořena, upřesníme požadované rozhraní
    myFoxy = myFoxy.QueryInterface(Components.interfaces.IAVFoxy);

    // Inicializace komponenty, volá se již metoda InitAVFoxy() třídy AVFoxy
    HandleInit(myFoxy.InitAVFoxy());
}
Else {     // Chyba: Komponenta není načtená
    …
}
```

Kód 4 Inicializace XPCOM

Po sepsání funkčního kódu a překladu do dynamické knihovny se musí binární soubor uložit do instalačního balíčku. Konkrétně se jedná o adresář *Component*, ve kterém jsou očekávána binární data a jejich popis pomocí typové knihovny XPT.

Následující kapitoly se věnují konkrétním náležitostem, jak zajistit antivirovou kontrolu souborů, jak získávat informace a data z AVG a jak data předávat do prostředí JavaScript. Musím upozornit, že kód je parciálně rozdělen mezi rozšíření Firefox (JavaScript) a komponentu XPCOM v jazyce C++.

4.2.2 Sledování událostí v prostředí Firefox

Firefox poskytuje dvě navzájem odvozená rozhraní nsIObserverService a nsIObserver. První zajišťuje službu šíření vnitřních událostí v aplikaci, tedy se jedná o server událostí. Klient se nazývá *observer* a užívá rozhraní nsIObserver. Při své inicializaci se přihlásí ke službě typu nsIObserverService a řekne, o kterých konkrétních událostech chce být oznamován. Zdrojový kód této kapitoly se řadí do klientské (JavaScript) části AVFoxy. Inicializace observeru vypadá například takto:

```
// Přihlášení ke službě sledování událostí
var observerService = Components.classes["@mozilla.org/observer-service;1"]
                                    .getService(Components.interfaces.nsIObserverService);

// Přidání konkrétní události observeru (objekt jménem DownloadObserver)
observerService.addObserver(DownloadObserver, "dl-done", false);
```

Kód 5 Vytvoření observeru

Pro aplikaci AVFoxy nás zajímají následující události:

Událost	Kód
Začátek stahování souboru	dl-start
Přerušení stahování souboru	dl-cancel
Stahování selhalo	dl-failed
Úspěšné stažení souboru	dl-done

Tabulka 6 Pozorované události

Při destrukci aplikace je důležité správně odregistrovat všechny observery, aby nedocházelo k paměťovým únikům. Kromě naslouchání plní observer také funkci vyvolávání událostí. Přitom si můžeme zaregistrovat své vlastní události, kterým potom můžou naslouchat i další aplikace. Tato možnost je využita pro změnu nastavení celé aplikace a při resetování statistik. Po uložení se tak snadno celá aplikace dozví, že si musí znovu načíst základní nastavení a s tím i své chování. Nedochází tak k problémům při komunikaci jinak téměř samostatných částí programu. AVFoxy uživatelské události jsou:

Událost	Kód
Proběhla změna nastavení	avfoxy-settings-saved
Reset statistiky	avfoxy-stats-reset

Tabulka 7 Definované události

Oznámení události resetu statistiky proběhne po provedení příkazu v Kódu 6:

```
observerService.notifyObservers(null, "avfoxy-stats-reset", null);
```

Kód 6 Reset statistiky

4.2.3 Antivirová kontrola

Pro kontrolu souborů využívá AVFoxy komponentu XPCOM (C++). Komponenta pracuje se softwarem AVG díky AVG API, které nabízí metody antivirové kontroly. Dynamická knihovna tak může zprostředkovat výhody profesionálního nástroje přes vlastní metodu TestFile() se vstupním parametrem *umístění souboru*. Metoda má však bohatší výsledek než jeden parametr. Výstupní parametry jsou v Tabulkách 8 a 9:

Parametr	Typ
Index souboru v seznamu	integer
Možnost léčení infekce	boolean
Pouze oznámení	boolean
Textový popis infekce	string

Tabulka 8 Výstupní parametry metody TestFile

	Výsledek
I	Žádná známá infekce, soubor je v pořádku
II	Nalezená infekce
III	Poškozený soubor
IV	Nepřístupný soubor
V	Málo paměti
VI	Software AVG není nainstalován, nelze zkontrolovat

Tabulka 9 Celkový výsledek operace

Metoda TestFile může být volána z klientské části AVFoxy JavaScript díky technologii XPConnect, která propojí XPCOM komponentu s aplikací Firefox. U negativního nálezu není uživatel nijak obeznámen, ale po nalezené infekci čeká komponenta na reakci klienta. Objekt s kontrolovaným souborem je v této chvíli uložen ve speciálním seznamu pod číselným indexem. Hned po provedení reakce je paměť bezpečně uvolněna.

4.2.4 Reakce na infikovaný soubor

Uživatel má možnost volby vhodné reakce na nalezenou infekci. Klientská část je psána v JavaScriptu a ovládá prvky varovného dialogu. Také, pokud si uživatel zvolil implicitní akci, pak volá klient některou z metod XPCOM komponenty, dle Tabulky 10.

Akce	Metoda
Přesunutí do virového trezoru	MoveToVaultInfection(index)
Léčení	CleanInfection(index)
Mazání	DeleteInfection(index)
Ignorování	IgnoreInfection(index)

Tabulka 10 Metody reakce na infekci

Podle indexu pozná komponenta, který soubor bude například smazán. Aktuální indexovaný seznam má svůj důvod. Počítá s možností, že je stahováno více souborů potenciálních infekcí zaráz a uživatel třeba odskočí na oběd. Varovná zpráva nemusí být tedy vyhodnocena, přesto je nezbytné infikovaný soubor blokovat. V nejhorším případě, tedy při nulové interakci uživatele a ukončování systému, jsou všechny zbývající infekce přesunuty do virového trezoru a paměť uvolněna bezpečně. Ignorováním infekce se neprovede žádná akce se souborem, ale odblokuje se přístup a uvolní zabraná paměť.
Jmenované metody vracejí *TRUE* v případě úspěšné akce, jinak se vrací *FALSE* a klientská část musí zkusit jiné řešení pro danou infekci. Uvedený případ nastane, když selže třeba léčení souboru, pak stále zbývají reálnější možnosti jako přesun či smazání.

4.2.5 Informace o AVG

Uživatel má k dispozici základní informace o aktuálních verzích programu AVG i aplikace AVFoxy. K získání potřebných dat se používá AVG API s metodami GetVersion a CheckLicense. Komponenta data posouvá pro klientskou část metodami GetVersion a GetLicenseInfo již ve správném formátu. Metoda GetVersion vrací *TRUE* v případě úspěchu. Selhává, pokud není dostupný kompatibilní AVG software. Má následující výstupní parametry:

- Majoritní verzi AVG
- Minoritní verzi AVG
- Verzi jádra AVG
- Verzi virové databáze
- Datum vydání virové databáze

Druhá metoda GetLicenseInfo vrací *TRUE* v případě úspěchu. Selže, pouze pokud není dostupný kompatibilní AVG software. Má následující výstupní parametry:

- Stav licence
- Typ licence
- Počet počítačů, pro které licence platí
- Datum vypršení platnosti

4.3 Uživatelské rozhraní

Všechny užitečné znalosti pro tvorbu uživatelského rozhraní v jazyce XUL jsou k dispozici na webových stránkách [4], kde se soustřeďuje aktuální specifikace tohoto jazyka. Doporučuji použít nástroj XUL Editor. Do běžného textového pole zadáte svůj kód a výsledek je okamžitě zobrazen v náhledu. Programátor si tak ušetří hodiny neefektivní práce.

Základní prostředí aplikace AVFoxy tvoří nástrojová lišta, funkční ikonka a menu. Jsou popsány v souboru *avfoxy_overlay.xul*. Další soubory s příponou *xul* popisují dialogová okna pro nastavení, informace, statistiky a varování.

Hlavní menu

```
<menupopup id="menu_ToolsPopup" >
    <menu id="avfoxy-menu" label="&avfoxy.txt.menu;" insertbefore="javascriptConsole">
        <menupopup>
            …
            <menuitem id="avfoxy-menu-about" label="&avfoxy.txt.menu.about;" oncommand="ShowAbout();" />
        </menupopup>
    </menu>
</menupopup>
```

Kód 7 Ilustrace zdrojového kódu

Menu se stane součástí Nástrojového menu s identifikací *menu_ToolsPopup*. Bude mít lokalizovaný štítek a jeho pozice je hned před konzolou JavaScriptu. Štítek obsahuje vyskakovací menu s jednotlivými ovládacími prvky. Atributem *oncommand* přiřadíme prvku spustitelný kód, který se nachází ve skriptovém souboru.

Ikona

```
<statusbar id="status-bar">
    <statusbarpanel id="avfoxy-iconic" class="statusbarpanel-menu-iconic" >
        <menupopup>
            …
        </menupopup>
    </statusbarpanel>
</statusbar>
```

Kód 8 Ilustrace zdrojového kódu

Jednoduchá ikonka skrývající menu. Zobrazuje se na pozici *status-bar*, což je dolní stavová řádka v prohlížeči Firefox. Změnu obrázku dosáhneme dvěma příkazy:

```
        var iconic = document.getElementById('avfoxy-iconic');
        iconic.src = "chrome://avfoxy/skin/avg_enabled.png";
```

Kód 9 Změna ikony

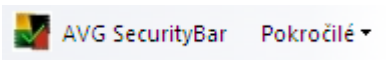
Obrázek 3 Ukázka ikony

Nástrojová lišta

```
<statusbar id="status-bar">
    <statusbarpanel id="avfoxy-toolbar" class="chromeclass-toolbar-additional" hidden="false" pack="start" >
        <toolbarbutton id="avfoxy-button-status" label="&avfoxy.txt.toolbar;" oncommand="OpenURL();"/>
        <toolbarbutton label="&avfoxy.txt.button.advanced;" type="menu" align="center" >
            <menupopup>
                ...
            </menupopup>
        </toolbarbutton>
    </statusbarpanel>
</statusbar>
```

Kód 10 Ilustrace zdrojového kódu

Rozložitější nástrojový panel zaujímá pozici na dolní stavové liště prohlížeče Firefox. Skládá se ze dvou tlačítek. První tlačítko zobrazuje ikonku a text, po kliknutí načte lokalizovanou webovou prezentaci produktu AVG. Druhé tlačítko se jmenuje *Pokročilé* a po kliknutí zobrazí rolovací uživatelské menu.



Obrázek 4 Ukázka lišty

Virová výstraha

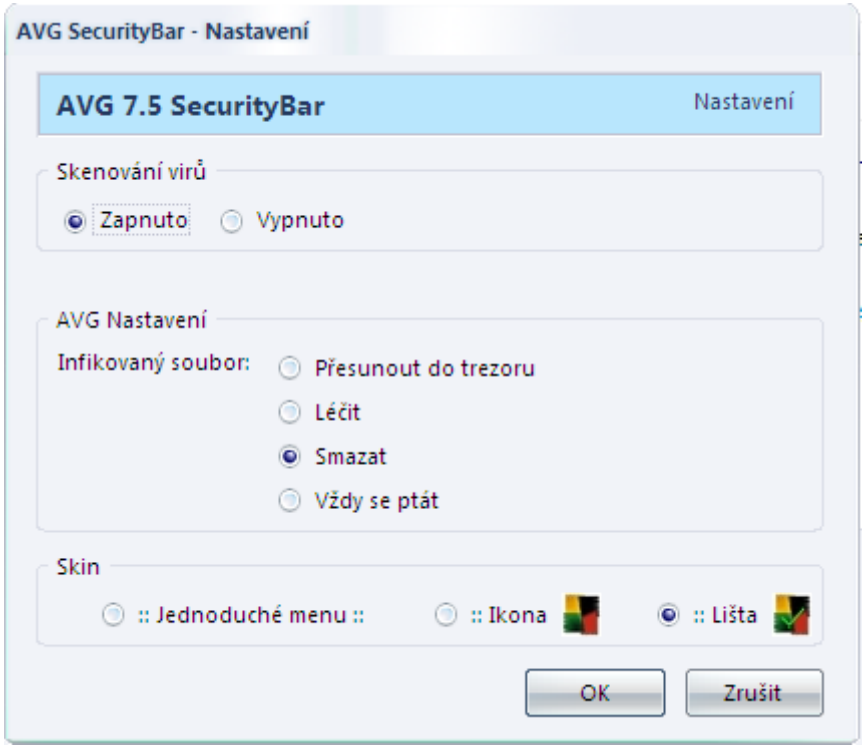
Dialogové okno, u kterého už není vhodné exemplárně předvádět celý kód. Ten je k dispozici v přílohách mezi všemi ostatními zdrojovými texty. Okno se při spuštění inicializuje, získá potřebná data z komponenty AVFoxy a zobrazí uživateli varovnou zprávu s tlačítky volby patřičné reakce na nalezenou infekci. Skládá se z několika textových štítků a pěti tlačítek:

- Virový trezor
- Léčit
- Smazat
- Ignorovat
- Potvrdit

Při čemž ne vždy jsou všechna tlačítka k dispozici. Závisí to na hodnotách získaných dat z XPCOM komponenty a programu AVG. Pokud není možné infekci léčit, tlačítko léčení zůstane skryté. Pokud AVG usoudí, že soubor se má pouze ohlásit, ale nemusí být infikován, zobrazí se jediné tlačítko *Potvrdit*.Při nalezené infekci je záměrně u popisů použita červená barva, aby byla upoutána pozornost ke zprávě. Okno nelze obejít ani měnit jeho rozměry a objevuje se na popředí obrazovky.

Nastavení

Okno nastavení se skládá ze tří tematických rámců a jejich možností.

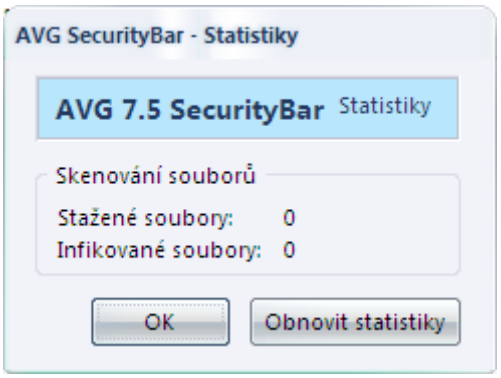


Obrázek 5 Nastavení

První dva rámce jsou povolené pouze, když AVFoxy spolupracuje s kompatibilním software AVG. Jinak je totiž virová kontrola nedostupná a možnosti volby jsou pro uživatele zbytečné. Vzhled aplikace si však může zvolit vždy.

Statistiky

Aplikace zaznamenává celkový počet stažených souborů a detekovaných infekcí.



Obrázek 6 Statistiky

Informace o programu

Po zobrazení lze vidět dva tematické rámce a jejich položky.

Obrázek 7 Informace o programu

Kaskádové styly CSS

Kaskádové styly definují obecný i konkrétní vzhled webové prezentace HTML. Vývojový tým Mozilly se rozhodl přidat tuto možnost i pro tvorbu uživatelského prostředí. Klasické CSS bohužel není pro takovou tvorbu vhodné, protože nebylo dostatečně dimenzované ani funkční. Řešením bylo vytvoření rozšířeného CSS se standardním jádrem.

Mozilla používá CSS k definici všech vlastností jeho prvků. Navíc lze hotové ovládací prvky ještě dále rozšiřovat díky jazyku XBL. Odpadá definice rozložení, která přešla na výhodnější správce rozložení. Stále je možné polohovat prvky i pomocí CSS. V projektu AVFoxy jsou CSS užity velmi okrajově na definici základní ikony loga pro různé rozlišení obrazovky. Pokusy o dynamické CSS, které je teoreticky možné, skončily negativně, a tak jsou problémy dynamických obrázků řešeny v JavaScriptu.

4.4 Lokalizace textů

Přenositelnost Mozilly se projevila také ve snadné lokalizaci všech textů, které se v programu objevují. V adresáři *locale* se nacházejí složky pro jednotlivé národnosti, například *en-US*“. Do složky se nahrají soubory definující texty. Jejich forma je vždy stejná, pouze jazyk textu se mění.

O přítomnosti lokalizace je třeba říci Firefoxu v instalačním a organizačním souboru. Také musí každá národnostní složka zahrnovat popisný soubor RDF. Aplikace vždy rozlišuje mezi statickými a dynamickými texty. Statické texty se načítají při překladu souborů XUL. Používá se vlastností jazyka XML. Každý statický text představuje jednu entitu s unikátním jménem. Dynamické texty se načtou do zvláštních proměnných a používají se v JavaScriptu. Mezi statickou a dynamickou technologií je propastný rozdíl, a proto je nelze sloučit. V dynamickém případě je důležité použít kódování Unicode – UTF8.

Soubor statických textů *statics.dtd:*

```
<!ENTITY avfoxy.txt.menu.settings "Nastavení" >
```

Použití entity při definici vzhledu *overlay.xul:*

```
<!DOCTYPE dialog SYSTEM "chrome://avfoxy/locale/statics.dtd">
…
<menuitem id="avfoxy-iconic-settings" label="&avfoxy.txt.menu.settings;" />
```

Soubor dynamických textů *dynamics.properties:*

```
avfoxy.txt.error.unknown = Chyba: N\u011Bco je \u0161patn\u011B.
```

Použití proměnné při varování uživatele o neznámé chybě:

```
// Inicializace dynamické lokalizace
var localized = document.getElementById("avfoxy.locale.dynamic");
// Zpráva o chybě
alertlocalized.getString('avfoxy.txt.error.unknown');
```

V souboru, kde se definuje rozhraní používající skripty JavaScript je ještě nezbytné uvést inicializační formuli pro dynamické texty, například soubor *overlay.xul:*

```
<script type="application/x-javascript" src="chrome://avfoxy/content/avfoxy.js" />
…
<stringbundleset id="stringbundleset">
    <stringbundle id="avfoxy.locale.dynamic" src="chrome://avfoxy/locale/dynamics.properties" />
</stringbundleset>
```

4.5 Instalace

Vytvořená adresářová struktura je téměř připravená k instalaci do libovolného prohlížeče Mozilla. V instalačním souboru *install.rdf* uvedeme všechny důležité informace o programu a jeho autorovi. V organizačním souboru *chrome.manifest* je popsána struktura naší aplikace. Potřebné znalosti jsou přesně zapsány na vývojovém webu Mozilly [3] a ve článku [5].

Jakmile máme připraveny všechny náležitosti. Zabalíme všechny soubory a adresáře kořenové složky formátem ZIP a výslednému souboru dáme příponu *xpi*. XPI je instalační balíček Mozilly, můžeme ho vystavit na serveru ke stažení či ho do prohlížeče instalovat jednoduchým přetáhnutím myší. Nainstalované doplňky jsou sledovány Správcem rozšíření, kde je možné aktualizovat novější verze aplikací nebo nechtěný doplněk odinstalovat.

4.6 Testování

Aplikace je otestována na cílovém systému Windows XP a Vista v 32bitové i 64bitovém vydání. Do testů jsem samozřejmě zahrnul názory kamarádů a zkušených uživatelů internetových prohlížečů. Virové testy proběhly za velké pomoci projektu Eicar [9]. Program Eicar je považován za virus, ale není tvořen škodlivým kódem. Je však považován za virus, na který musí antivirový software správně reagovat.

V testovacím období byly odhaleny slabosti prvních řešení, což autora vedlo k vylepšení chování programu v různých chybových stavech. Cílem testování bylo dosažení spolehlivé aplikace schopné se vyrovnat neočekávanými událostmi. Kromě samotných testů bylo nutné zachovat lokalizaci celého programu v různých jazycích, aniž by si koncový uživatel všimnul, že rozdíl mezi rozšířením Firefox a binárně kompilovanou komponentou je propastný. Z testů pak vyplynulo, že 95% informací bude podáno klientskou částí aplikace, kde bude zajištěn jednotný vzhled zprávy i jazyk.

5 Závěr

Vývoj rozšíření do Firefoxu se může na první pohled zdát triviální, ale jak se projekt Mozilla rychle vyvíjí, stává se, že mnoho dostupných informačních zdrojů je neaktuálních a to zpomaluje práci a komplikuje studium problematiky. Programátor má jen minimální šanci svůj projekt interaktivně ladit, protože jsou problémy s ladícími nástroji pro Firefox. Součástí práce na projektu bylo pochopení a využití řady moderních technologických přístupů, od architektury AVG a Mozilly, jazyků založených na XML, přes XPCOM technologii a dynamicky linkované knihovny.

Většina těchto technologií byla do nedávna autorovi neznámá, po důkladném studiu se podařilo navrhnout výsledný systém a jeho implementace proběhla bez větších problémů. Při testování bylo odhaleno a opraveno mnoho bezpečnostních chyb v kódu. Současná verze aplikace v0.4 již stabilně funguje na cílovém systému Microsoft Windows XP a Vista. S přihlédnutím k současným možnostem AVG API aplikace AVFoxy splňuje zadání a rozšiřuje toto ještě o další možnosti (statistiky a informace o AVG). Aplikace bude zdarma k dispozici všem zákazníkům společnosti AVG. Hlavním požadavkem byla výsledná stabilita aplikace, kterou v současné době AVFoxy plní. Rozšíření nijak zbytečně neruší uživatele při prohlížení internetu, a přesto na pozadí pracuje důmyslná kontrola, která odhalí potenciální bezpečnostní průniky.

V budoucnosti bude zřejmě aplikace rozšířena o další poskytované služby. Ve spolupráci s AVG může zajišťovat pokročilou kontrolu internetové komunikace. Dále se uvažuje o možnostech blokačních filtrů, které by hlídaly nechtěnou reklamu a nevhodné stránky. Také bude vhodné zajistit kompatibilitu i pro jiné systémy než Windows, reálně lze přemýšlet například o operačním systému Linux a jeho distribucích.

Literatura

[1] GAGYI, Matěj. Rozšírte si Firefox a Thunderbird - sami!. *ABC Linuxu* [online]. 2005 [cit. 2008-04-19]. Dostupný z WWW: <http://www.abclinuxu.cz/clanky/programovani/rozsirte-si-firefox-a-thunderbird-sami>.

[2] KOSEK, Jiří. *Inteligentní podpora navigace na WWW s využitím XML*. [s.l.], 2002. 64 s. Diplomová práce. Dostupný z WWW: <http://www.kosek.cz/diplomka/>.

[3] Mozilla.org. *Mozilla Developer Documentation* [online]. c1998-2008 [cit. 2008-04-18]. Dostupný z WWW: <http://www.mozilla.org/docs/>.

[4] ANDERSEN, Aaron, DEAKIN, Neil. *XulPlanet* [online]. c1999-2006 [cit. 2008-04-20]. Dostupný z WWW: <http://www.xulplanet.com/>.

[5] *Extension development* [online]. c1998-2008 [cit. 2008-04-19]. Dostupný z WWW: <http://kb.mozillazine.org/Extension_development>.

[6] TURNER, Doug, OESCHGER, Ian. Creating XPCOM Components. *Mozilla.org* [online]. 2003 [cit. 2008-04-22]. Dostupný z WWW: <http://www.mozilla.org/projects/xpcom/book/cxc/>.

[7] GRISOFT, s.r.o. *AVG API Interface 7.5 Documentation*. 2007.

[8] *Lupa.cz : server o českém Internetu* [online]. c1998-2008 [cit. 2008-05-03]. Dostupný z WWW: <http://www.lupa.cz/>.

[9] *Eicar Homepage* [online]. c1998-2008 [cit. 2008-05-05]. Dostupný z WWW: <http://www.eicar.org/>.

Seznam příloh

Příloha 1

Nápověda k ovládání

Rozšíření AVG SecurityBar si nainstalujeme stažením ze serveru do cílového prohlížeče nebo přetáhnutím instalačního souboru do jeho okna. Uživatel dosáhne všech možností počítačovou myší. Samozřejmě lze vstoupit do základního menu i pomocí klávesnice, ale nijak to práci nezrychluje. Nejdříve zvolíme optimální nastavení aplikace. K tomu slouží položka menu *Nastavení.* Názvy možností v hlavním menu přesně vypovídají o jejich funkci:

- *Nastavení*
- *Vypnutí / Zapnutí virové kontroly*
- *Statistiky*
- *Informace*

Při nalezené virové infekci je nutné rozhodnutí uživatele. Na obrazovce se zobrazí dialogové okno s přehledným popisem infekce a několika tlačítky. Uživatel by měl obratem reagovat na vzniklou situaci, na základě svobodného rozhodnutí, jednou z nabízených voleb:

- *Přesun do virového trezoru*
- *Léčení infekce*
- *Smazání souboru*
- *Ignorování infekce*

Příloha 2

Technické požadavky

Aplikace je optimalizována pro minimální výkon počítače.

Doporučená **minimální sestava**:

Procesor	Intel Celeron / AMD Athlon
Grafická karta	SVGA
Operační paměť	256 MB
Harddisk	10 GB
Připojení k internetu	56 Kbps modem / ISDN
Myš	
Klávesnice	

Potřebný software:

Microsoft Windows XP / Vista
Mozilla Firefox 1.6 a vyšší
AVG 6.0 – 7.5