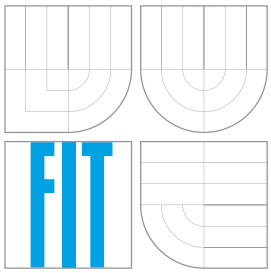
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

INSTALACE A KONFIGURACE SYSTÉMU AVG POMOCÍ POLITIKY ACTIVE DIRECTORY

INSTALLATION AND CONFIGURATION OF AVG SYSTEM USING ACTIVE DIRECTORY POLICIES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MICHAL ŠPAČEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. RADEK KOČÍ, Ph.D.

BRNO 2008

Zadání bakalářské práce

1. Seznamte se s problematikou správy sítí, zejména s možnostmi instalace, konfigurace a sledování různých programových produktů v sítích Microsoft Windows.

2. Nastudujte problematiku politik (group policies) v prostředí Microsoft Active Directory, klientských rozšíření a doplňků editorů skupinových politik, WMI a MMC.

3. Navrhněte a v jazycích C# a C++ implementujte klientské rozšíření (Group Policy Client-side Extension) s podporou Resultant Set of Policy a doplněk editoru skupinových politik (Group Policy Object Editor snap-in) pro správu antivirového programu AVG.

4. Proveďte hodnocení výsledků práce a navrhněte možnosti dalších rozšíření.

Licenční smlouva

Licenční smlouva je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.

Abstrakt

Tato práce se zabývá funkcí systémů Windows nazývanou Zásady Skupiny, která slouží k centralizované správě a konfiguraci uživatelských účtů a počítačů v sítích Microsoft Windows. Tato práce popisuje architekturu tohoto modulu a možnosti rozšíření jeho funkčnosti. Tyto teoretické poznatky jsou pak uplatněny při návrhu a implementaci rozšíření, které slouží ke konfiguraci systému AVG

Klíčová slova

Zásady skupiny, Active Directory, Microsoft Windows

Abstract

This thesis is about Group Policy, which is a feature of Microsoft Windows systems that provides centralized management and configuration of users and computers in Microsoft Windows networks. This thesis describes Group Policy architecture and possible ways of extending its features. This theoretical knowledge is later used for designing and implementing extension, which is used to configure AVG system

Keywords

Group Policy, Active Directory, Microsoft Windows

Citace

Michal Špaček: Instalace a konfigurace systému AVG pomocí politiky Active Directory, bakalářská práce, Brno, FIT VUT v Brně, 2008

Instalace a konfigurace systému AVG pomocí politiky Active Directory

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Radka Kočího, Ph.D. Další informace mi poskytli zaměstnanci firmy AVG Technologies CZ, s.r.o. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Michal Špaček
12. května 2008

Obsah

Kapitola 1

Úvod

V dnešní době se ve velké míře rozšířilo používání privátních počítačových sítí, ať už se jedná o komerční podnikové sítě nebo nekomerční sítě například ve školách. S rostoucím počtem uživatelů a počítačů v těchto sítích roste i problém, jak účinně, efektivně a s minimálními náklady spravovat klientské stanice, používaný software a také samotné uživatelské účty. Mezi nástroje, které umožňují centralizovanou správu v prostředí sítí Microsoft Windows, patří *Zásady Skupiny* (anglicky *Group Policy*), které jsou přímo integrovány do posledních verzí operačních systémů řady Windows NT.

Modul *Zásady Skupiny* je postaven na *Active Directory*, což není nic jiného než implementace LDAP adresáře pro použití v prostředí Microsoft Windows. Způsob, jakým se *Active Directory* používá pro uložení dat *Zásad Skupin*, a způsob, jakým je zajištěno provázání těchto dat s uživatelskými účty a účty počítačů, jsou popsány v kapitole 2.1.

Samotnou funkčnost Zásad Skupin provádějí tzv. rozšíření. Microsoft dodává se svými operačními systémy několik takových předinstalovaných rozšíření, které umožňují správu těchto systémů a programů, které jsou s těmito systémy dodávány (jako je např. Internet Explorer). V kapitole 3 jsou popsány základní principy, které je nutné použít pro implementaci vlastního rozšíření. V kapitole 4 je pak popsán návrh a implementace rozšíření, které bude sloužit ke správě systému AVG.

Kapitola 2

Group Policy - Zásady skupiny

2.1 Architektura

Group Policy je postaveno na architektuře klient-server. Na serveru se používá *Active Directory* a *SYSVOL*[1] pro uložení zásad a pak nástroje na správu a editaci zásad skupiny: *Group Policy Management Console* (GPMC) , *Group Policy Object Editor* (GPOE) a jeho rozšíření. Klientskou část tvoří *Group Policy Engine* implementovaný v knihovně *userenv.dll*, který běží v rámci služby *Winlogon* (implementovaný v *winlogon.exe*). Samostatnou aplikaci nastavení pak provádějí klientská rozšíření, které jsou implementovány v samostatných dll knihovnách.

Základní jednotkou nesoucí data Group Policy je virtuální objekt nazývaný *Group Policy Object* (GPO). V rámci Group Policy pak existuje logická i fyzická reprezentace každého Group Policy Objectu. Logickou částí je *Group Policy container*, který je uložen v Active Directory, fyzická část se pak nazývá *Group Policy template* (GPT), který je uložen v souborovém systému na řadiči domény. GPO může být přilinkováno ke kontejnerovým objektům v AD (doméně, sídlu a organizační jednotce) a tím je určeno, na které objekty se má konkrétní nastavení aplikovat. Aplikace GPO může být ještě omezena nastavením oprávnění u GPO, případně WMI filtrem.

2.1.1 Group Policy container

GPC je v Active Directory reprezentován jako kontejnerový objekt, konkrétně objekt třídy *groupPolicyContainer*. GPC je v Active Directory uložen v kontejnerech *Policies* a *System* pro příslušnou doménu. Jako jméno (cn - Common-Name) je zvoleno GUID, které je zároveň i jednoznačným identifikátorem GPO. Celá LDAP cesta ke GPC v Active Directory (pro doménu testing.local) pak vypadá takto: `LDAP://CN={2B71864E-3BE4-4243-A4C6-50305E049F87},CN=Policies,CN=System,DC=testing,DC=local`.

Použitím GUID jako jména GPO je zaručeno, že všechny GPO mají jedinečné jméno v rámci Active Directory, i když je při vytváření nového GPO zvolen stejný název pro více GPO.

[1] SYSVOL je veřejně přístupný sdílený adresář na serveru, který obsahuje kopie všech veřejných souborů pro doménu. Adresář SYSVOL je replikován na všechny řadiče domény.

Název atributu	**Popis**
createTimeStamp	Obsahuje datum a čas kdy byl GPC vytvořen
displayName	Obsahuje jméno, které bylo zadáno při vytvoření GPO
Flags	Obsahuje status GPO: *Flags* = 0 - GPO je povolený (Enabled) *Flags* = 1 - část konfigurace uživatele je zakázaná *Flags* = 2 - část konfigurace počítače je zakázaná *Flags* = 3 - GPO je zakázaný (Disabled)
gPCFileSysPath	Obsahuje cestu k příslušnému GPT
gPCMachineExtensionNames	Obsahuje seznam GUID klientských rozšíření, které mají zpracovat část konfigurace počítače GPO a také i GUID rozšíření, které uložilo data do GPO
gPCUserExtensionNames	Obsahuje seznam GUID klientských rozšíření, které mají zpracovat část konfigurace uživatele GPO a také i GUID rozšíření, které uložilo data do GPO
versionNumber	Určuje aktuální číslo verze GPO, důležité k určování změn v GPO

Tabulka 2.1: Významné atributy GPC

Atributy GPC

Samotný GPC pak obsahuje základní informace o GPO jako jsou *displayName* (pod jakým názvem se GPO zobrazuje), číslo verze GPO, cestu ke GPT a *access control list* (ACL), který obsahuje seznam oprávnění určující, kdo má přístup ke změně nebo zpracování GPO. GPC pak také obsahuje i seznam klientských rozšíření, které je nutné zavolat na klientské stanici pro zpracování příslušného GPO. Detailnější popis k nejdůležitějším atributům je v tabulce 2.1.

Uložení vlastních dat v GPC

Kromě těchto atributů může *Group Policy container* obsahovat i konkrétní data, které pak zpracovávají klientské rozšíření. Například rozšíření *Wireless Network Policy* ukládá své nastavení do objektu třídy *msieee80211-Policy*, který je v rámci GPC uložen v kontejnerech *CN=Wireless, CN=Windows,CN=Microsoft* . Je třeba si uvědomit, že pokud se rozhodneme při vytváření vlastního klientského rozšíření ukládat data do GPC, bude pravděpodobně nutné modifikovat *Active Direcotry Schema* v lesu[2], kde se bude naše rozšíření používat. Při této modifikaci pak bude třeba nadefinovat vlastní třídu objektů, který bude pak obsahovat data nového rozšíření. Pro modifikaci lze použít *Active Directory Schema Manager snap-in*, případně jiné konzolové nástroje. Je ale třeba si uvědomit, že nově přidané atributy a třídy nelze ze schématu vymazat, lze je pouze deaktivovat.

Zobrazení GPC

Pro zobrazení GPC je možné použít konzoli *Active Directory Users and Computers*, ale je nutné zaškrtnout položku *Advanced Features* v menu *View*. Pak lze GPC najít ve stromu *JmenoDomeny→System→Policies*. Toto zobrazení ovšem nedovoluje zobrazit jednotlivé

[2]les (forest) je spojená skupina doménových stromů, které používají stejné AD schéma

atributy. Pro detailní zobrazení je nutné použít utilitu *ADSI Edit*. Tato konzola je ale dostupná až po instalaci *Windows Support Tools*. *ADSI Edit* pak umožňuje zobrazit a editovat všechny atributy, které jsou uložené v Active Directory.

2.1.2 Group Policy template

Group Policy template tvoří fyzickou část GPO. Je reprezentován soubory uloženými v souborovém systému. Každý GPO má přidělen hlavní adresář, který je uložen v adresáři *%SystemRoot%\SYSVOL\domain\Policies*. Po vytvoření GPO Active Directory vytvoří i příslušný GPT pro daný GPO. Jako jméno adresáře GPT je zvoleno GUID - je identické s GUID příslušného GPC. Cesta k adresáři GPT je také uložena v GPC v atributu *GPC-FileSysPath*. V rámci hlavního adresáře GPT je pak vytvořeno několik dalších adresářů a souborů, které obsahují aktuální nastavení GPO. Ze sítě je pak GPT dostupný v rámci sdíleného adresáře SYSVOL. Pokaždé jsou tedy vytvořeny dvě kopie GPT: První v adresáři %SystemRoot%\SYSVOL\domain\Policies\GPOGUID a druhá v ve sdíleném adresáři SYSVOL - SYSVOL\DomainName\Policies\GPOGUID (kde DomainName je FQDN domény). Každý GPT je také replikován na řadiče domény v doméně za pomocí služby *File Replication Service*.

Struktura GPT

V hlavním adresáři GPT jsou vždy vytvořeny dva adresáře: *Machine* a *User*. *Machine* obsahuje nastavení patřící do části *Konfigurace počítače*, *User* pak obsahuje nastavení patřící do části *Konfigurace uživatele*. V hlavním adresáři GPT je také uložen soubor *Gpt.ini*, který obsahuje číslo verze GPO (ekvivalentní atributu *versionNumber* v GPC) a zobrazované jméno GPO (display name). Číslo verze GPO je desítková interpretace čtyřbytového šestnáctkového čísla, přičemž horní dva byty označují verzi části Konfigurace uživatele GPO a dolní dva byty označují verzi části Konfigurace počítače GPO. Čísla verzí v GPT a GPC se mohou lišit (může to nastat například v případě, pokud se u upraveného GPO replikovala pouze část GPC a GPT teprve čeká na replikaci). GPT také může obsahovat další adresáře a soubory, podle toho, které rozšíření uložilo do GPO své nastavení.

2.1.3 Linkování GPO

Linkování GPO určuje, na které počítače a uživatele se má daný GPO aplikovat. GPO lze přilinkovat ke třem druhům kontejnerových objektů v Active Directory: k doméně, sídlu a organizačním jednotkám. Samotný *link* je reprezentován atributem *gPLink* u těchto kontejnerových objektů a obsahuje celou LDAP cestu ke GPC části GPO a také příznak *status flag*, který je od LDAP cesty oddělen středníkem. Hodnota atributu *gPLink* pak může vypadat takto: LDAP://cn={E6AD4E44-5D5D-42E1-A49FFF50F03249E9},cn=policies,cn=system,DC=testing,DC=com;0 . Pokud je k danému kontejnerovému objektu přilinkováno více GPO, jsou jednotlivé hodnoty pro daný GPO v atributu *gPLink* uzavřeny do hranatých závorek ([]). Číslo na konci tohoto atributu (*status flag*) pak udává, zda je link povolen a zda je link vynucený (*enforced*): hodnota 0 znamená, že link je povolen a není vynucen, 1 není povolen a není vynucen, 2 je povolen a vynucen, 3 není povolen a je vynucen.

U kontejnerových objektů je pak ještě jeden atribut *gPOptions*, který je nastaven, pokud je povoleno blokování dědičnosti Group Policy. Standardně všechny následnické objekty dědí GPO objekty svého předchůdce - nastavením tohoto příznaku lze toto chování zakázat.

2.1.4 Lokální GPO

Lokální Group Policy Object je uložen na jednotlivých počítačích, přičemž každý počítač má právě jeden lokální GPO. Tento objekt obsahuje vždy jen část GPT, konkrétně v adresáři *%systemroot%\System32\GroupPolicy*. Struktura souborů v tomto adresáři je stejná jako struktura normálního GPO v Active Directory, jen v souboru *GPT.ini* je navíc uložena informace o parametrech *gPCMachineExtensionNames* a *gPCMachineUserNames*, které jsou normálně uloženy v GPC, ale protože GPC část v lokálním GPO není, je třeba je mít v tomto souboru.

Lokální GPO má nejmenší vliv na zpracování Group Policy, protože je vždy zpracováván jako první a také proto, že nastavení v Active Directory mají větší prioritu.

2.1.5 Zpracování Group Policy

Zpracování Group Policy má na starosti *Group Policy Engine*, který seřazuje GPO do seznamů, které předává jednotlivým klientským rozšířením, jenž následně provádějí samostatné zpracování GPO. Klientská rozšíření Group Policy jsou implementovány v samostatných dll knihovnách. Výjimku tvoří rozšíření *Šablony pro správu* (Administrative Templates), které je implementováno přímo v knihovně *userenv.dll* a je vždy zpracováno jako první.

Existují dva typy zpracování zásad: na popředí (*foreground processing*, taky nazývané prvotní zpracování) a na pozadí (*background processing*).

Zpracování na popředí

Zpracování na popředí nastává při startu počítače a při přihlášení uživatele. Zpracování na popředí je typické v tom, že je dokončeno předtím, než je uživateli k dispozici plocha a než je mu umožněno s ní interaktivně pracovat. Je tedy vhodné pro zpracování takových druhů nastavení, kde je třeba prostředí bez uživatele.

Zpracování na pozadí

Zpracování na pozadí nastává periodicky a asynchronně na ostatních procesech. Je vhodné pro takové nastavení, které je nutné aplikovat periodicky. Zpracování na pozadí nastává na klientských stanicích a na serverech, které nejsou řadiči domény, každých 90 minut. K těmto devadesáti minutám je přičten náhodný čas až 30 minut. Zpracování na pozadí pro řadiče domény je prováděno každých 5 minut. Tyto parametry lze změnit pomocí příslušných nastavení.

Synchronní a asynchronní zpracování

Ve Windows 2000 se zpracování na popředí vždy dělo synchronně, to znamená, že zásady pro počítač byly zpracovány, než se uživateli zobrazila přihlašovací obrazovka a zásady pro uživatele byly zpracovány dřív, než byla uživateli k dispozici plocha. Ve Windows XP přibyla možnost asynchronního zpracování na popředí, které je podporováno mechanismem *fast logon optimization*. Pokud je zapnuta (což je výchozí nastavení), tak Windows nečeká na kompletní inicializaci sítě a umožňuje existujícím uživatelům přihlásit se pomocí uložených osobních údajů (uživatel se už dříve přihlašoval z dané pracovní stanice). Tato optimalizace pak umožňuje uživateli začít pracovat, zatímco jsou skupinové zásady aplikovány na pozadí - to je až v okamžiku, kdy je dostupná síť. Je tedy velice podobná procesu zpracování group policy na pozadí. Tato optimalizace je vždy vypnuta za následujících podmínek: Uživatel se

poprvé přihlašuje k počítači, uživatel má cestovní uživatelský profil nebo domovský adresář, a když uživatel má synchronní přihlašovací skripty. Optimalizaci lze vypnout pomoci nastavení příslušné politiky, pak se stanice s Windows XP bude chovat stejně jako stanice s Windows 2000.

Postup při aplikaci zásad

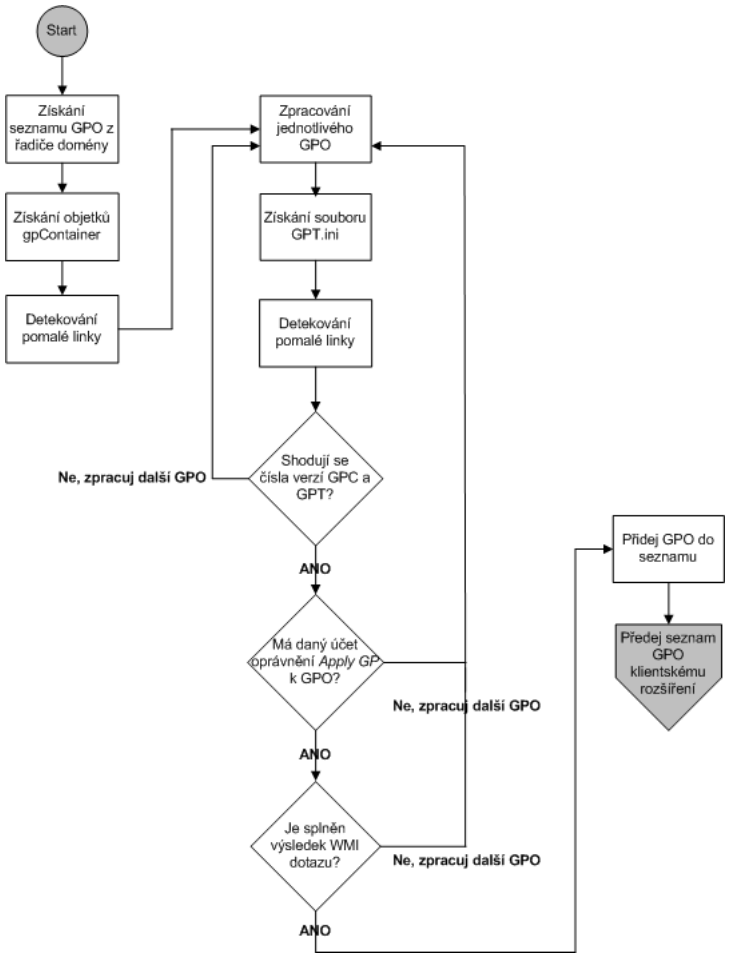
Samotná aplikace zásad skupiny pak probíhá v několika krocích. Prvotní zpracování zásad pro účet počítače probíhá takto:

1. Počítač nalezne řadič domény (využívá přitom DNS) a přihlásí se do domény.
2. Počítač použije ICMP ping na řadič domény aby zjistil, zda je připojen přes pomalou linku k řadiči domény.
3. Pomocí LDAP dotazu se získá z Active Directory všechny GPO, které jsou přilinkovány k organizačním jednotkám, doméně a sídlu, ke kterým patří účet počítače. Z výsledku dotazu sestaví seznam všech DN (distinguished name) GPO, které se mají aplikovat pro daný počítač.
4. Za pomoci výsledků získaných v předchozím kroku, počítač sestaví LDAP dotazy, které získají z Active Directory řadu atributů GPO, jako je např. cesta ke GPT a seznam GUID klientských rozšíření, které je třeba zavolat pro zpracování politiky.
5. Windows se pak pomocí SMB protokolu připojí ke sdílenému adresáři SYSVOL a přečte obsah souboru *Gpt.ini* pro každý GPO nalezený v kroku 3. Ze všech získaných atributů pak sestaví seznamy GPO, které předává klientským rozšířením. Při sestavování seznamu GPO se testuje, zda má počítač oprávnění *Apply Group Policy*. Pokud ne, není GPO do seznamu zařazeno. Dále se také vyhodnocují dotazy WMI oproti klientskému WMI repositáři a určuje se, zda počítač splňuje požadavky tohoto dotazu. Pokud ne, GPO opět není do seznamu zařazeno. Group policy engine pak volá jednotlivé klientské rozšíření a předává jim seznam GPO ke zpracování.
6. Klientské rozšíření pak porovnává číslo verze GPO s číslem verze GPO, které je uloženo v registrech v klíčích *Group Policy history* (bude vysvětleno dále).
7. Pokud se čísla verzí shodují (nezměnilo se od posledního zpracování), je GPO přeskočeno. Pokud od posledního zpracování došlo ke smazání některého GPO, klientské rozšíření odstraní toto nastavení.
8. Klientské rozšíření pak čte obsah GPT (případně také GPC) a aplikuje nastavení
9. Po dokončení zpracování všech GPO, klientské rozšíření zapisuje RSoP data do WMI v CIMOM databázi.

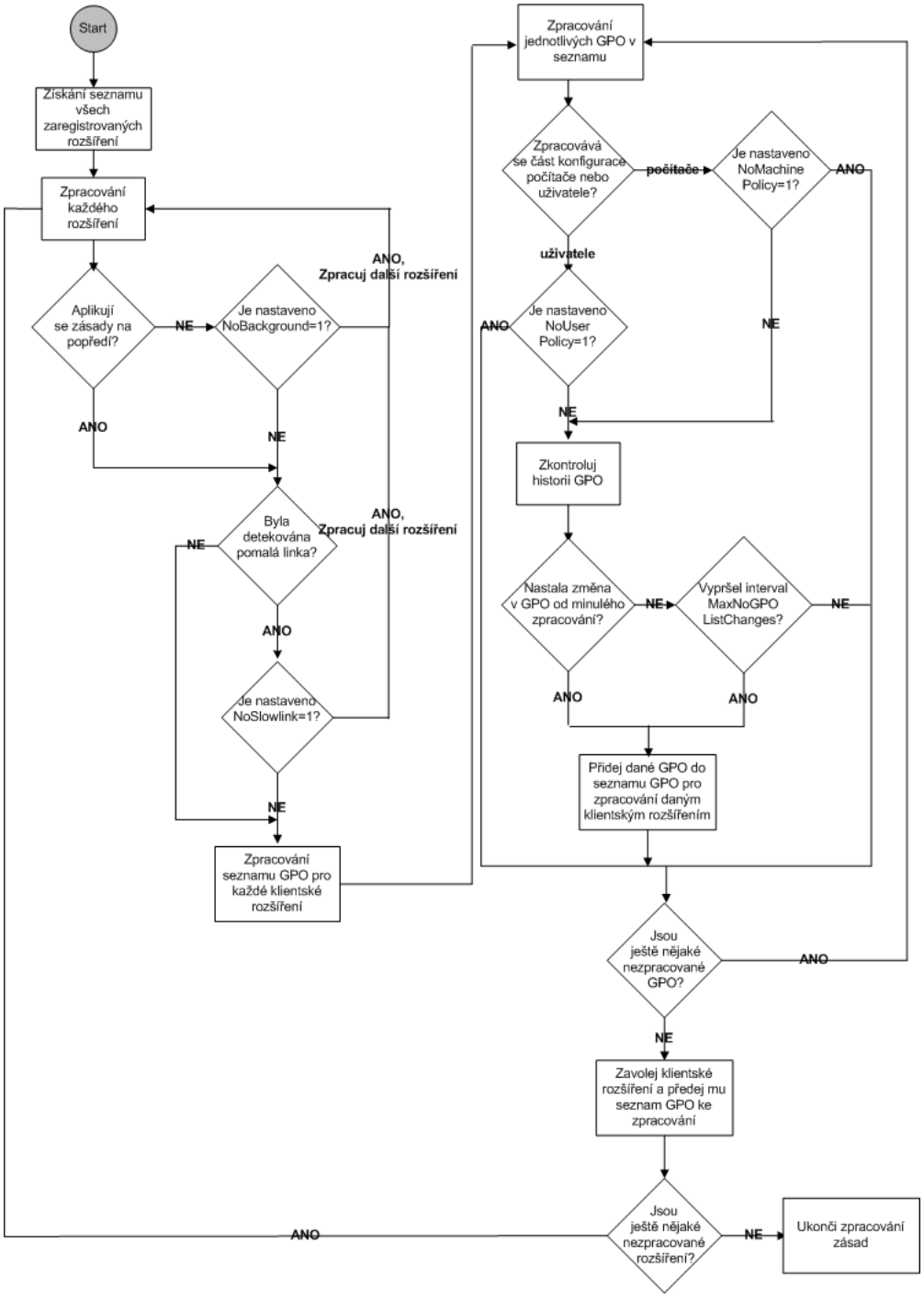
Po přihlášení uživatele se opakuje podobný proces, jen je místo účtu počítače použit účet uživatele. Zpracování na pozadí probíhá také podobně.

Schéma 2.1 shrnuje proces sestavení seznamu GPO a schéma 2.2 shrnuje proces volání klientských rozšíření a zpracování tohoto seznamu.

Zpracování na pozadí lze také vyvolat manuálně pomocí utility *gpupdate.exe*. Aplikace můžou žádat o aktualizaci politiky pomocí funkce `RefreshPolicy`, případně `RefreshPolicyEx`, která na rozdíl od předchozí umožňuje zadat druhý atribut `RP_FORCE` - má to stejný účinek jako při použití *gpupdate.exe /force* - ignoruje se veškerá optimalizace zpracování a znovu se použijí všechna nastavení, i když nebyly detekovány žádné změny v GPO.



Obrázek 2.1: Schéma vytvoření seznamu GPO



Obrázek 2.2: Schéma volání klientských rozšíření a zpracování seznamu GPO

2.1.6 Group Policy History Data

Na každém počítači, který je spravován pomocí Group Policy se ukládají data o historii všech zpracovávaných GPO. Group Policy engine ukládá tyto data do registrů do *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\History*. Pod tímto klíčem jsou klíče reprezentující jednotlivé klientské rozšíření, které zpracovávají GPO a jsou pojmenovány pomocí GUID příslušných klientských rozšíření. Pod tímto klíčem jsou klíče číslovány od 0 a tyto klíče reprezentují GPO, které se zpracovávaly v posledním zpracováním Group Policy. Jedním z hlavních důvodů ukládání těchto dat je, aby klientské rozšíření bylo schopné rozlišit, zda se daný GPO objekt změnil od minulého zpracování (změnilo se číslo verze) a optimalizovat tak zpracování Group Policy. Kromě čísla verze se do registrů ukládají prakticky všechny data, které jsou dostupná klientskému rozšíření o konkrétním GPO ze struktury `GROUP_POLICY_OBJECT` (bude vysvětleno dále).

2.2 Výsledná sada zásad - RSoP

Funkce *Výsledná sada zásad* (Resultant Set of Policy) slouží k simulaci aplikace zásad a také k řešení potíží při aplikaci zásad. RSoP má dva režimy: protokolovací a plánovací. Režim protokolování RSoP (Group Policy Results) slouží k zobrazení konkrétních požitých nastaveních zásad pro daný počítač nebo uživatele. Plánovací režim (Group Policy Modeling) simuluje nastavení zásad, které budou použity pro počítače nebo uživatele. K zobrazení dat RSoP pak lze použít konzoli *Výsledná sada zásad* (rsop.msc), konzolový nástroj *gpresult* nebo také přímo *Group Policy Management console*.

2.2.1 Protokolovací režim RSoP

Režim protokolování RSoP slouží správcům ke kontrole existujících nastavení zásad použité pro počítače nebo uživatele. Používá se v situacích, kdy je třeba zjistit, které konkrétní nastavení zásad jsou použity pro daný počítač nebo uživatele, nebo když je nutné vyhledat přepsané zabezpečení neúspěšných zásad případně zjistit, jakým způsobem ovlivňují skupiny zabezpečení nastavení zásad. Například pokud jsou zásady uplatňovány na více úrovních (například síť, doména, řadič domény a organizační jednotka), může dojít ke konfliktům ve výsledcích. Výsledná sada zásad pomáhá určit sadu použitých zásad a pořadí těchto zásad, ve kterém jsou uplatňovány.

Pro uložení RSoP dat se používá *CIMOM* (Common Information Model Object Management) databáze na lokálním počítači. Tato databáze je pak přístupná pomocí služby WMI [3].

2.2.2 Plánovací režim RSoP

Plánovací režim funkce RSoP pomáhá správcům plánovat rozšiřování a reorganizaci sítě. Pomocí plánovacího režimu funkce RSoP lze od existujících GPO získat veškerá použitelná nastavení zásad. Plánovací režim se používá v situacích, kdy je třeba simulovat účinek konkrétního nastavení pro počítač nebo uživatele, nebo je třeba otestovat prioritu GPO, když je účet uživatele nebo počítače přesouván do nového umístění v rámci Active Directory, případně je přesouván nebo přidáván do nové skupiny zabezpečení.

[3] [6] Služba Windows Management Instrumentation (WMI) je implementace správy WBEM (Web-Based Enterprise Management) společnosti Microsoft. Správa WBEM je iniciativa k ustanovení standardů pro přístup a sdílení informací správy v rozlehlé síti.

V plánovacím režimu RSoP simuluje aplikaci zásad pomocí *Group Policy Directory Access Service* (GPDAS) na řadiči domény. GPDAS simuluje aplikaci GPO tím, že předává GPO virtuálním klientským rozšířením, které pak zapisují RSoP data do lokální databáze *CIMOM* na řadiči domény, odkud jsou pak vyčteny příslušným nástrojem pro zobrazení RSoP dat.

Kapitola 3

Rozšíření Group Policy

3.1 Klientské rozšíření

Jak již bylo zmíněno v předchozím textu, klientské rozšíření se implementují jako dll knihovny. Tato knihovna pak musí implementovat a exportovat callback funkci `ProcessGroupPolicy` nebo `ProcessGroupPolicyEx`. `ProcessGroupPolicyEx` je nutné použít v tom případě, že chceme podporovat režim protokolování RSoP. `ProcessGroupPolicyEx` je podporována až od Windows XP, protože RSoP bylo uvedeno až v této verzi Windows. Pokud je nutná funkčnost i pod verzí Windows 2000, je nutné implementovat i funkci `ProcessGroupPolicy`.

Knihovna klientského rozšíření by měla také implementovat a exportovat funkce `DllRegisterServer` a `DllUnregisterServer`, které registrují (a odregistrují) klientské rozšíření z registrů zapsáním (smazáním) příslušných klíčů a hodnot. K registraci pak slouží utilita *RegSvr32.exe*, které se předá cesta k dll knihovně. Odregistrace se provádí zadáním parametru /u.

3.1.1 Registrace Klientského rozšíření

Každé klientské rozšíření musí být registrováno na klientské stanici. V klíči *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\ GPExtensions* je pro rozšíření nutné zapsat klíč, jehož jméno je GUID klientského rozšíření. V tomto klíči by pak měly být hodnoty popsané v tabulce 3.1.

3.1.2 Callback funkce ProcessGroupPolicy (ProcessGroupPolicyEx)

Tato funkce pokrývá hlavní funkčnost klientského rozšíření, tedy zpracování všech GPO. Při implementaci této funkce je nutné vzít na vědomí, že funkce je volána službou *winlogon* (winlogon.exe / userenv.dll) pod lokálním systémovým účtem, jakákoliv chyba tedy může způsobit nestabilitu Windows - Například chybná práce s pamětí může vést až k ukončení procesu *winlogon.exe*, na což reaguje Windows zobrazením modré obrazovky s chybovou hláškou, že proces *winlogon.exe* byl neočekávaně ukončen.

Vlastní seznam GPO, které se mají aplikovat, je funkci předáván jako obousměrně vázaný seznam (vysvětleno u popisu parametrů). Funkce má vracet hodnotu typu *Win32 error code*. Hodnota `ERROR_SUCCESS` znamená, že aplikace politiky byla úspěšná. Pokud v seznamu GPO objektů nebyla detekována žádná změna a je nutné zajistit znovu volání klientského rozšíření, i pokud je v registrech nastaveno *NoGPOListChanges*, funkce může

DllName	hodnota typu *REG_EXPAND_SZ* obsahující cestu k dll knihovně klientského rozšíření, která obsahuje callback funkci
EnableAsynchronousProcessing	pokud je nastaveno na 1, callback funkce dokončuje své zpracování asynchronně a po dokončení zpracování by měla zavolat funkci *ProcessGroupPolicyCompleted* nebo *ProcessGroupPolicyCompletedEx* (podle toho zda je callback funkce *ProcessGroupPolicy* nebo *ProcessGroupPolicyEx*)
ExtensionEventSource	hodnota typu *REG_MULTI_SZ*, která specifikuje *event source name* a *event log name* pro klientské rozšíření (není podporováno pod Windows 2000)
GenerateGroupPolicy	hodnota typu *REG_SZ*, která obsahuje jméno *GenerateGroupPolicy* callback funkce, která slouží ke generaci dat RSoP v plánovacím módu RSoP
MaxNoGPOListChangesInterval	specifikuje maximální počet minut, kdy se může klientské rozšíření přeskočit, pokud nebyla detekována žádná změna v seznamu GPO. Pokud je limit překročen, je rozšíření voláno standardním způsobem a není nastaven příznak *GPO_INFO_FLAG_NOCHANGES* (viz dále u popisu parametrů funkce *ProcessGroupPolicy*)
NoBackgroundPolicy	pokud je nastaveno na 1, klientské rozšíření není voláno, pokud se má aplikovat policy na pozadí
NoGPOListChanges	pokud je nastaveno na 1, klientské rozšíření není voláno, pokud není detekována žádná změna v seznamu GPO
NoMachinePolicy	pokud je nastaveno na 1, klientské rozšíření není voláno, pokud se má zpracovávat část konfigurace počítače
NoSlowLink	pokud je nastaveno na 1, klientské rozšíření není voláno, pokud je detekována pomalá linka síťového připojení
NotifyLinkTransition	zavolá callback funkci klientského rozšíření, aby bylo informováno, že došlo ke změně rychlosti linky mezi aplikacemi policy
NoUserPolicy	pokud je nastaveno na 1, klientské rozšíření není voláno, pokud se má zpracovávat část konfigurace uživatele
PerUserLocalSettings	Pokud je nastaveno na 1, nastavení policy se má ukládat v cache podle uživatele a podle počítače
ProcessGroupPolicy	hodnota typu REG_SZ, která specifikuje jméno callback funkce ProcessGroupPolicy v klientském rozšíření, která se má volat pro zpracování policy. Pokud je na Windows XP a novějších verzích Windows přítomna i hodnota *ProcessGroupPolicyEx*, je použito jméno funkce uložená v hodnotě *ProcessGroupPolicyEx*.
ProcessGroupPolicyEx	specifikuje callback funci ProcessGroupPolicyEx, není podporováno na Windows 2000
RequiresSuccessfulRegistry	pokud je nastaveno na 1, klientské rozšíření je voláno jen tehdy, pokud bylo úspěšně zpracováno klientské rozšíření *registry*

Tabulka 3.1: Hodnoty definované v registrech pro klientské rozšíření

vrátit hodnotu ERROR_OVERRIDE_NOCHANGES. Všechny ostatní hodnoty znamenají, že zpracování selhalo.

Group Policy engine předává callback funkci tyto parametry:

1. DWORD dwFlags - příznaky, můžou nabývat těchto hodnot:
   (a) GPO_INFO_FLAG_MACHINE - značí, že by se měla aplikovat část konfigurace počítače místo části konfigurace uživatele.
   (b) GPO_INFO_FLAG_BACKGROUND - značí, že klientské rozšíření je voláno za účelem aplikování group policy na pozadí (co to je aplikace na pozadí je vysvětleno v předchozím textu)
   (c) GPO_INFO_FLAG_ASYNC_FOREGROUND - má se vykonat asynchronní aplikace na popředí (také vysvětleno v předchozím textu)
   (d) GPO_INFO_FLAG_SLOWLINK - byla detekována pomalá linka
   (e) GPO_INFO_FLAG_VERBOSE - rozšíření by mělo zapsat do event logu podrobné informace
   (f) GPO_INFO_FLAG_NOCHANGES - nebyly detekovány žádné změny v seznamu GPO
   (g) GPO_INFO_FLAG_LINKTRANSITION - byla detekována změna rychlosti linky mezi aplikacemi Group Policy
   (h) GPO_INFO_FLAG_LOGRSOP_TRANSITION - byla detekována změna v logování dat RSoP mezi aplikací předchozí politiky a aktuální politiky
   (i) GPO_INFO_FLAG_FORCED_REFRESH - jedná se o vynucené aplikování policy - měly by se aplikovat všechny GPO nezávisle na verzi GPO (po volání *gpupdate /force*)
   (j) GPO_INFO_FLAG_SAFEMODE_BOOT - informuje o tom, že systém Windows byl spuštěn v nouzovém režimu
2. HANDLE hToken - token pro účet uživatele nebo účet počítače. Protože klientské rozšíření běží pod lokálním systémovým účtem, je nutné použít tento token k přístupu k síťovým zdrojům.
3. HKEY hKeyRoot - handle k registrovému klíči HKEY_LOCAL_MACHINE nebo HKEY_CURRENT_USER (podle toho, zda se aplikuje GPO pro počítač nebo uživatele)
4. PGROUP_POLICY_OBJECT pDeletedGPOList - ukazatel, přes který se předává seznam smazaných GPO, které jsou implementovány jako struktura GROUP_POLICY_OBJECT. Seznam je implementován jako obousměrně vázaný, to znamená, že ve struktuře jsou ukazatele na předchozí i následující GPO, lze ho tak snadno procházet. GPO jsou uspořádány v pořadí ve kterém se mají zpracovávat (to znamená, že ty co mají nastavené u linku Enforced jsou zařazeny na konec apod)
5. PGROUP_POLICY_OBJECT pChangedGPOList - přes tento ukazatel jsou předávány ukazatele GPO, které se mají aplikovat
6. ASYNCCOMPLETIONHANDLE pHandle - handle, který se předává při volání funkcí ProcessGroupPolicyCompleted a ProcessGroupPolicyCompletedEx, které informují systém, že klientské rozšíření dokončilo asynchronní aplikaci politiky. Pokud rozšíření aplikaci nepodporuje, je hodnota tohoto atributu 0

7. `BOOL* pbAbort` - ukazatel na `BOOL` hodnotu, která značí, jestli se má pokračovat ve zpracování GPO. Pokud nabude hodnoty `TRUE`, klientské rozšíření má přerušit zpracování GPO. V opačném případě se pokračuje ve zpracování. (nastává například pokud se uživatel odhlásí nebo se vypíná počítač)

8. `PFNSTATUSMESSAGECALLBACK pStatusCallback` - ukazatel na `StatusMessageCallback` funkci, která zobrazuje informaci v uživatelském rozhraní o aplikaci politiky, může být `null`.

Následující parametry jsou definovány navíc v ProcessGroupPolicyEx pro podporu logování RSoP dat:

9. `IWbemServices* pWbemServices` - definuje jmenný prostor RSOP kam má klientské rozšíření zapsat data. Pokud je NULL, rozšíření nemá logovat RSOP

10. `HRESULT* pRsopStatus` - přes tento ukazatel rozšíření vrací návratový kód, který značí, zda bylo logování RSoP dat úspěšné

3.1.3 Struktura GROUP_POLICY_OBJECT

Tato struktura je deklarována v souboru `userenv.h` a poskytuje informace o jednotlivých GPO v seznamu GPO. Struktura obsahuje dva ukazatele `pNext` a `pPrev`, které umožňují spojení více GPO do obousměrného seznamu, který se pak předává funkci `ProcesGroupPolicy`.

Struktura má tyto členy:

1. `DWORD dwOptions` - specifikuje příznaky linku. Může nabývat těchto hodnot:
   (a) `GPO_FLAG_DISABLE` - GPO je zakázaný
   (b) `GPO_FLAG_FORCE` - značí, že nastavení uložené v tomto GPO se mají aplikovat nezávisle na tom, zda se GPO změnil od poslední aplikace

2. `DWORD dwVersion` - číslo verze GPO, lze porovnávat s minulou verzí GPO (která je uložená v registrech) a tak detekovat, zda se objekt změnil, a případně optimalizovat tím, že se nezměněné GPO přeskočí

3. `LPCTSTR lpDSPath` - cesta k GPC části (v Active Directory)

4. `LPCTSTR lpFileSysPath` - cesta k GPT (v souborovém systému - SYSVOL)

5. `LPCTSTR lpDisplayName` - řetězec, který obsahuje zobrazované jméno GPO (které jsme zadali při vytváření GPO)

6. `TCHAR szGPOName[50]` - obsahuje unikátní jméno, které jednoznačně identifikuje GPO (GUID)

7. `GPO_LINK GPOLink` - Specifikuje informaci, k čemu je tento GPO přilinkován:
   (a) `GPLinkUnknown` - není dostupná žádná informace o linku
   (b) `GPLinkMachine` - GPO je přilinkován k lokálnímu nebo vzdálenému počítači
   (c) `GPLinkSite` - GPO je přilinkován k sídlu

(d) `GPLinkDomain` - GPO je přilinkován k doméně

(e) `GPLinkOrganizationalUnit` - GPO je přilinkován k organizační jednotce

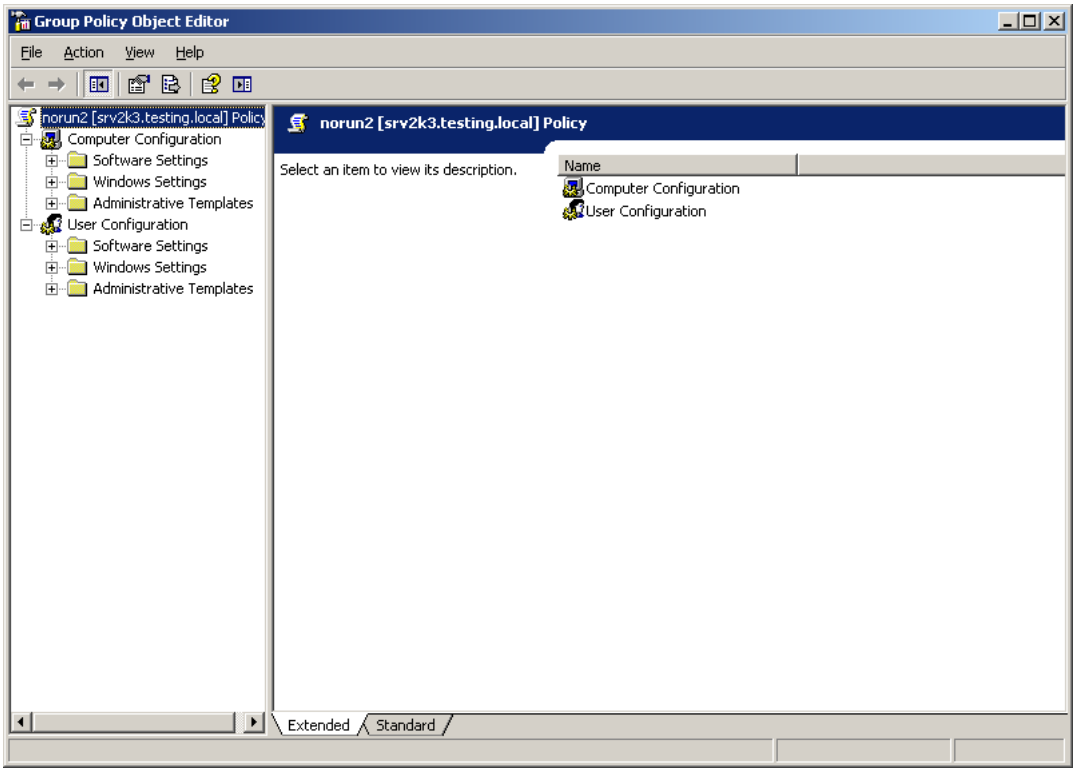
8. `LPARAM lParam` - User-supplied data.
9. `struct _GROUP_POLICY_OBJECT* pNext` - ukazatel na následující GPO v seznamu
10. `struct _GROUP_POLICY_OBJECT* pPrev` - ukazatel na předchozí GPO v seznamu
11. `LPTSTR lpExtensions` - Obsahuje GUID klientských rozšíření a GUID rozšíření Group-Policy Object Editoru, které uložily data do GPO.
12. `LPARAM lParam2` - User-supplied data.
13. `LPTSTR lpLink` - Cesta k sídlu, doméně nebo organizační jednotce, ke které je GPO přilinkován. Pokud se jedná o lokální GPO hodnota je "local"

Poznámka k sestavování dll knihovny klientského rozšíření:

Struktura `GROUP_POLICY_OBJECT` má jak ansi variantu `GROUP_POLICY_OBJECTA` tak i unicode variantu `GROUP_POLICY_OBJECTW`. Která z nich se použije při překladu se rozhodne podle toho, se kterým kódováním se překládá (v závislosti na tom, jestli je definováno makro `UNICODE`). Bohužel ANSI varianta mi nefungovala. Při překladu s nastaveným kódováním ANSI mají pak atributy typu `LPCTSTR` obsahovat řetězce s ANSI kódováním, ale ve skutečnosti obsahovaly unicode řetězce. Při pokusu o logování těchto řetězců se to projeví tím, že se zapíše jen první znak, jelikož za ním následuje hned nulový znak, protože unicode znaky zabírají dva byty. Tento problém by se dal lehce vyřešit, bohužel se ovšem projevuje i další chyba, která se projevovala u ukazatele `pNext`. Tento ukazatel by měl být `NULL`, pokud je v seznamu pouze jeden GPO. Bohužel ale nebyl a tak zpracování zhavarovalo na chybě přístupu do paměti, když se procházel seznam GPO pomocí cyklu. Proto doporučuji překládat raději jen s nastaveným kódováním Unicode.

3.2 Rozšíření Group Policy Object Editoru

Group Policy Object Editor 3.1 (GPOE) sám o sobě žádnou funkčnost neobsahuje. Tu vykonávají až jeho rozšíření, které ukládají data do GPO. GPOE je implementován jako snap-in pro *Microsoft Management Console* (MMC) a jeho rozšíření jsou také moduly snap-in pro tuto konzoli, které rozšiřují jmenný prostor GPOE. Ten je organizován následujícím způsobem: Na kořenový uzel GPOE navazují uzly *Konfigurace počítače* (Computer Configuration) a *Konfigurace uživatele* (User Configuration). Jasně je tak rozděleno nastavení, které se aplikuje na počítače a které na uživatele (tak jak bylo vysvětleno v předchozím textu). V obou uzlech se nachází další tři uzly: *Nastavení softwaru* (Software settings), *Nastavení systému Windows* (Windows settings) a *Šablony pro správu* (Administrative Templates). Uzly *Nastavení softwaru* a *Nastavení systému Windows* mohou obsahovat rozšíření jak pro část *Konfigurace počítače* tak i pro část *Konfigurace uživatele*. Uzel *Šablony pro správu* obsahuje nastavení, které je založeno na registrech, a může být rozšířen pomocí *Administrativních šablon*, což jsou speciální .adm soubory. GUID uzlů, které mohou být rozšířeny, jsou definovány v souboru `gpedit.h`.



Obrázek 3.1: Group Policy Object Editor

Rozšíření GPOE

Samotné rozšíření GPOE je tedy snap-in pro MMC konzoli, který rozšiřuje jeho jmenný prostor. Pro vývoj takového snap-in existují prakticky dva způsoby implementace. První způsob je jen pro nejnovější verzi konzole MMC verze 3.0., druhý způsob je implementace standardní dll knihovny v c++, která jde použít ve všech verzích MMC konzole.

Implementace GPOE snap-inu v jazyce C#

Pro verzi 3.0 MMC konzole se implementují snap-iny v jazyce C# pro platformu .NET. Pro implementaci základní funkčnosti rozšíření jmenného prostoru stačí implementovat jen třídu zděděnou ze třídy `NamespaceExtension` a nastavit atribut `ExtendsNodeTypeAttribute`, který označuje GUID primárního uzlu, pod který se má připojit náš uzel. V této třídě pak pro nejjednodušší funkčnost stačí implementovat konstruktor, ve kterém vytvoříme nový objekt třídy `ScopeNode` a pomocí zděděné vlastnosti `PrimaryNode` ho přidat do jmenného prostoru:

```
ScopeNode extensionRootNode = new ScopeNode();
PrimaryNode.Children.Add(extensionRootNode);
```

Toto nám zajistí velice základní funkčnost - zobrazení našeho rozšíření ve stromu jmenného prostoru.

Implementace GPOE snap-inu v jazyce C++

Druhý způsob psaní snap-inů lze použít ve všech verzích MMC konzole. Implementace takového snap-in je vlastně implementace COM serveru. Nejzákladnější snap-in musí implementovat MMC rozhraní `IComponentData` a `IComponent`, a navíc ještě dvě standardní COM rozhraní `IDataObject` a `IClassFactory`. Pro každé další rozšíření funkčnosti (přidání tlačítek na panel, přidání položek do kontextového menu apod.) je nutné implementovat další příslušné rozhraní. Snap-in, který rozšiřuje jmenný prostor musí být zaregistrován v registrech na příslušných místech a musí implementovat metodu `IComponentData::Notify`. MMC konzole volá tuto metodu, když uživatel vybere uzel, který rozšiřujeme, a zasílá zprávu `MMCN_EXPAND`. Snap-in v reakci na tuto zprávu implementuje přidání svých uzlů pod rozšiřovaný uzel. Je tedy jasně vidět, že tento způsob implementace je mnohonásobně složitější než předchozí.

Komunikace rozšíření s GPOE

Kromě toho že snap-in rozšiřuje jmenný prostor GPOE (zobrazuje se v něm) musí rozšíření i nějakým způsobem komunikovat s GPOE aby zjistilo informace o editovaném GPO, cestu ke GPT (kam ukládat data) a hlavně také informovat GPOE, že jsme do GPO uložili nějaké data, aby GPOE mohl zapsat příslušné data do GPC (GUID klientského rozšíření). Tyto funkce jsou implementovány v metodách rozhraní `IGPEInformation`. K tomuto rozhraní se nedá bohužel dostat, pokud se implementuje snap-in prvním způsobem, tedy v C#. Pokud implementujeme rozšíření GPOE druhým způsobem, lze rozhraní získat následovně: Musíme implementovat metodu `IComponentData::Notify` a reagovat na zprávu `MMCN_EXPAND`, což děláme, pokud se naše rozšíření už zobrazuje ve jmenném prostoru GPOE. Pokud tedy přijde zpráva `MMCN_EXPAND` a parametr `arg` je nastaven na `TRUE`, můžeme rozhraní získat například do proměnné `m_pGPTInformation` pomocí volání (`lpDataObject` je parametr metody `IComponentData::Notify`):

```
lpDataObject->QueryInterface(IID_IGPEInformation,
                             (LPVOID *)&m_pGPTInformation);
```

Ze tohoto způsobu získání rozhraní je jasné, že nelze použít `COM Interop`[1] a zpřístupnit tak rozhraní `IGPEInformation` v řízeném kódu a implementovat snap-in v C#. Pokud tedy chceme implementovat rozšíření GPOE, musíme použít druhý značně složitější způsob implementace v C/C++.

3.3 Implementace podpory RSoP

3.3.1 Definování RSoP schématu

Každé klientské rozšíření musí definovat schéma pro RSoP data, která bude ukládat do databáze CIMOM. Třída objektů, které se do této databáze ukládají, se vytvoří jako třída zděděná od třídy `RSOP_PolicySetting`, která již obsahuje základní atributy pro RSoP (creationTime, GPOID, id, name, precedence, SOMID). V této zděděné třídě je nutné definovat atributy, které jsou relevantní k datům klientského rozšíření, a také znovu definovat atributy `id` a `precedence` jako klíče(`key`). Definice této třídy se zapisuje do souborů .MOF (Managed Object Format), pomocí speciálního jazyka podobného c++, který vychází z IDL

[1] COM Interop umožnuje použití stávajících COM objektů v řízeném kódu aplikací .NET

(Interface Definition Language). Po vytvoření .MOF souboru je pak nutné tento soubor zkompilovat při instalaci klientského rozšíření a tím jej zahrnout do jmenného prostoru WMI. Ke kompilaci .MOF souborů slouží program *mofcomp.exe*. Příklad takové zděděné třídy:

```
// RSOP_SamplePolicySetting
// Comment: Sample class for GroupPolicy extension
//
[
Description("Sample class descritption")
]
class RSOP_SamplePolicySetting : RSOP_PolicySetting
{
[key, Description("Inherited from RSOP_PolicySetting"), Read,
       DisplayName("ID")]
string id;

[key, Description("Inherited from RSOP_PolicySetting"), Read,
       DisplayName("Precedence")]
uint32 precedence;

[Description("Sample string value"), Read, DisplayName("string value")]
string stringValue = "";

[Description("Sample integer value"), Read, DisplayName("integer value")]
uint32 integerValue = 0;

[Description("Sample array of bytes"), Read, DisplayName("byte array")]
uint8  byteArray[];

[Description("Sample boolean value"), Read, DisplayName("boolean value")]
boolean booleanValue = false;
};
```

3.3.2 Podpora protokolovacího módu RSoP v klientském rozšíření

Jak již bylo zmíněno v kapitole 3.1.2, pro podporu protokolovacího módu je nutné implementovat funkci `ProcessGroupPolicyEx`. Pro zapsání RSoP dat je nejprve nutné získat přístup k naší třídě `RSOP_SamplePolicySetting` a to pomocí volání metody `GetObject` objektu `pWbemServices`, který byl jako parametr předán funkci `ProcessGroupPolicyEx`:

```
IWbemClassObject*   pClass;
HRESULT hr = pWbemServices->GetObject(
    SysAllocString(L"RSOP_SamplePolicySetting"),
    0L,
    NULL,
    &pClass,
    NULL );
```

Dále je pak nutné vytvořit novou instanci naší třídy, což učiníme pomocí volání metody `SpawnInstance`:

```
IWbemClassObject *pInstance = NULL;
HRESULT hr = pClass->SpawnInstance( 0, &pInstance );
```

Pokud již máme vytvořený nový objekt, můžeme do něj zapisovat RSoP data pomocí volání metody `Put`. Všechny typy dat se předávají této metodě jako typ `VARIANT`. Například hodnotu vlastnosti `booleanValue` můžeme zapsat takto:

```
VARIANT var;
var.vt = VT_BOOL;
var.boolVal = VARIANT_TRUE;
hr = pInstance->Put( L"booleanValue", 0, &var, 0 );
```

Pokud již jsou uloženy všechny atributy, můžeme objekt uložit do databáze pomocí volání metody `PutInstance`:

```
hr = pWbemServices->PutInstance(
       pInstance, WBEM_FLAG_CREATE_OR_UPDATE, NULL, NULL );
```

3.3.3 Podpora plánovacího módu RSoP v klientském rozšíření

Pro podporu plánovacího módu RSoP v klientském rozšíření je nutné implementovat a exportovat funkci `GenerateGroupPolicy`. Tato funkce je volána službou Group Policy Data Access Service, která simuluje aplikaci GPO. Funkce `GenerateGroupPolicy` má tyto atributy:

1. `DWORD dwFlags` - příznaky, můžou nabývat těchto hodnot:
   (a) `GPO_INFO_FLAG_SLOWLINK` - pokud je nastaven tento příznak, má se simulovat připojení přes pomalou linku
   (b) `GPO_INFO_FLAG_VERBOSE` - rozšíření by mělo zapsat do event logu podrobné informace
2. `BOOL* pbAbort` - pokud je tento parametr nastaven na TRUE, rozšíření má okamžitě ukončit zpracování
3. `WCHAR* pwszSite` - ukazatel na řetězec, který obsahuje jméno sítě
4. `PRSOP_TARGET pComputerTarget` - ukazatel na strukturu `RSOP_TARGET`. Tento parametr může být NULL, ale pokud je NULL, je požadována nastavená atribut `pUserTarget`
5. `PRSOP_TARGET pUserTarget` - ukazatel na strukturu `RSOP_TARGET`. Tento parametr může být NULL, ale pokud je NULL, je požadována nastavená atribut `pComputerTarget`

Vlastní seznam GPO objektů, které jsou nutné pro simulaci aplikace GPO, je předaván v rámci stuktury `RSOP_TARGET`, která obsahuje tyto atributy:

1. `WCHAR* pwszAccountName` - ukazatel na řetězec, který obsahuje jméno účtu počítače nebo uživatele

2. `WCHAR* pwszNewSOM` - ukazatel na řetězec, který obsahuje novou doménu nebo novou organizační jednotku, které je členem účet v atributu `pwszAccountName`

3. `SAFEARRAY* psaSecurityGroups` - ukazatel na pole, které obsahuje skupiny zabezpečení

4. `PRSOPTOKEN pRsopToken` - ukazatel na `RSOPTOKEN`, který je nutné použít při volání funkcí `RSoPAccessCheckByType` a `RSoPFileAccessCheck`

5. `PGROUP_POLICY_OBJECT pGPOList` - ukazatel na seznam GPO objektů, tento seznam se prochází stejně jako ve funkci `ProcessGroupPolicy`

6. `IWbemServices* pWbemServices` - definuje jmenný prostor RSOP kam má klientské rozšíření zapsat RSoP data

RSoP data se zapisují stejným způsobem, jako v případě protokolovacího módu. Opět je nutné využít atribut `pWbemServices`. Pokud je navíc použito i stejné schéma, je postup naprosto stejný jako v předchozí kapitole 3.3.2

3.3.4 Rozšíření snap-inu Výsledná sada zásad

MMC snap-in *Výsledná sada zásad* slouží k zobrazení RSoP dat. Tento snap-in má prakticky stejnou architekturu jako *Group Policy Object Editor*, takže je výhodné použít již existující snap-in pro GPOE a jen do něj dopsat podporu pro zobrazení dat RSoP, které se zobrazí v již stávajícím uživatelském rozhraní, které samozřejmě musí být přístupné pouze v režimu pro čtení. Postup pro implementaci tohoto snap-inu je tedy podobný implementaci rozšíření pro *Group Policy Object Editor*, ale pokud přidáváme podporu pro RSoP do již stávajícího rozšíření pro GPOE, je vhodné pro snap-in zaregistrovat pod druhým *Class ID* (CLSID), aby se dalo v kódu rozeznat (konkrétně ve funkci `DllGetClassObject`), ve kterém módu snap-in běží a tím patřičně upravit uživatelské rozhraní.

Komunikace rozšíření se snap-inem Výsledná sada zásad

Kromě přidání našeho rozhraní do jmenného prostoru snap-inu *Výsledná sada zásad* je také nutné s tímto snap-inem komunikovat. K tomu slouží rozhraní `IRSOPInformation`. Přístup k tomuto rozhraní lze získat stejným způsobem, jako rozhraní `IGPEInformation` v případě rozšíření GPOE, tedy pomocí volání metody `QueryInterface` objektu `lpDataObject`, který jsme získali, když MMC konzole volala naši implementaci metody `IComponentData::Notify`, jejíž prostřednictvím nám byla zaslána zpráva `MMCN_EXPAND`. Z rozhraní `IRSOPInformation` budeme potřebovat zejména metodu `GetNamespace`, pomocí níž získáme řetězec obsahující jmenný prostor, kde jsou uloženy RSoP data pro daný počítač nebo uživatele. K tomuto jmennému prostoru se pak lze připojit pomocí metody `IWbemLocator::ConnectServer` (objekt `WbemLocator` získáme standardním způsobem pomocí volání funkce `CoCreateInstance`) a tím získáme přístup k rozhraní `IWbemServices`, jehož metody využijeme při získávání konkrétních RSoP objektů. Pro tento účel nejprve zavoláme metodu `IWbemServices::ExecQuery`, pomocí které můžeme vyhledat objekty určité třídy. Samotné hledání se provádí pomocí jazyka *WQL* (WMI Query Language), což je podmnožina jazyka ANSI SQL (SQL 92). Tento jazyk nemá podporu pro modifikace, to znamená že nepodporuje příkazy `INSERT` a `UPDATE`. Všechny objekty naší třídy pak lze získat pomocí dotazu `SELECT * FROM RSOP_SamplePolicySetting`. Metoda `IWbemServices::ExecQuery` ovšem

nevrací přímo hledané objekty, ale vrací nám ukazatel na rozhraní `IEnumWbemClassObject`, které použijeme pro procházení objektů odpovídající příslušnému WQL dotazu. Konkrétně použijeme metodu `IEnumWbemClassObject::Next`, která vrací jednu nebo více instancí z výsledku (objekty se ukládají do pole, proto je možné zavolat tuto metodu pro získání více instancí). Vlastní objekt je pak reprezentován stejně jako v případě klientského rozšíření, které do něj data zapisovalo, pomocí rozhraní `IWbemClassObject`. Pro získání konkrétní hodnoty nějakého atributu pak lze zavolat metodu `IWbemClassObject::Get`, přičemž hodnoty se opět vracejí jako typ `VARIANT`. Po skončení zpracování RSoP dat nesmíme zapomenout uvolnit enumerator výsledku pomocí metody `IEnumWbemClassObject::Release`.

Kapitola 4

Návrh a implementace rozšíření pro správu systému AVG

4.1 Stručný popis systému konfigurace AVG

Konfigurace AVG je uložena v konfiguračních souborech `.cfg`, které se standardně nacházejí v adresáři `C:\Documents and Settings\All Users\Data aplikací\avg8\Cfg`. K těmto souborům a celé konfiguraci je možné přistupovat pomocí několika metod definovaných v rozhraních `IAvgConfigManager` a `IAvgConfig`. Metody těchto rozhraní jsou implementovány v knihovně `avgcfgx.dll`, která je součástí instalace AVG.

Struktura konfigurace AVG

Všechny konfigurační hodnoty jsou uloženy v několika stromových strukturách. Každý tento strom ma svůj kořen (`root`). Rozdělení konfigurace do více stromů odděluje různé konfigurace částí systému AVG. Například kořen `krnl` obsahuje konfiguraci jádra AVG (jako je například konfigurace rezidentního šítu). Existuje mnoho dalších rootů pro další části systému AVG.

Každý root obsahuje obecně více instancí konfigurací. Většina kořenů včetně `krnl` obsahuje pouze jednu instanci (u toho kořenu více instancí nemá smysl), ale například pro konfigurace scanů existuje více instancí.

Typy konfiguračních hodnot

Každá konfigurační položka je určitého typu. Těchto základních typů je pouze šest:

1. `BOOL` - boolovský typ, může nabývat pouze hodnoty TRUE a FALSE
2. `DWORD` - celočíselný typ
3. `STRING` - řetězec znaků
4. `DATE` - typ, jehož hodnoty určují datum a čas
5. `MULTISTRING` - pole řetězců
6. `BYTEARRAY` - pole bytů, používá se pro uložení ostatních dat v binární podobě

4.2 Uložení konfiguračních dat v GPO

Soubory s konfigurací v GPT

Pro uložení dat v GPO byla zvolena část GPT, tedy uložení dat v souborovém systému. Tato možnost byla zvolena hlavně s ohledem na jednodušší implementaci, kdy k přístupům k datům stačí využít standardní funkce pro přístup k souborům. V rámci GPT tak bude přidán adresář AVG, který bude obsahovat jednotlivé soubory s konfiguracemi. Každý tento soubor bude představovat jednu konkrétní instanci konfigurace pro daný kořen. Jako jméno souboru byl zvolen název kořenu, pak následuje tečka a za ní číslo konfigurační instance. Jako přípona byla zvolena přípona `.dat`.

Formát .dat souborů

V každém souboru je nutné mít pro jednotlivé konfigurační hodnoty název hodnoty, typ hodnoty a samotnou hodnotu. Vzhledem k těmto faktům byl navrhnut co nejjednodušší formát uložení dat. Ty se tedy ukládají v textové formě, kde každé jedné hodnotě přísluší jeden řádek textového souboru. Jako název k hodnoty se použije celá cesta v konfiguračním stromu a to zejména z toho důvodu, že tento celý název lze použít v metodách rozhraní `IAvgConfig`. Za celým názvem hodnoty pak následuje typ hodnoty uzavřený v hranatých závorkách ([ ]). Za pravou hranatou závorkou pak následuje znak =, za kterým je uvedena konkrétní textová reprezentace hodnoty. Způsob reprezentace hodnoty závisí na typu hodnoty. Pro typ `BOOL` je povolena pouze hodnota `T` nebo `F`, pro `DWORD` je to sekvence číslic 0 až 9, pro `STRING` jsou přípustné ASCII znaky. Typ `DATE` musí být ve formátu YYYY-MM-DD/HH-MM-SS (rok, měsíc, den, hodina, minuta, sekunda) nebo může obsahovat hodnotu `INVALIDTIME`, která označuje neplatné datum. Typ `MULTISTRING` je reprezentován jedním řetězecem, který obsahuje všechny podřetězce. Ty jsou odděleny znakem `|`. Pokud je nutné mít znak `|` v řetězci, použije se zpětné lomítko `\|`. Dva po sobě následující znaky `|` pak znamenají prázdný podřetězec. Jelikož položky typu `BYTEARRAY` můžou obsahovat jakékoliv hodnoty, tedy i hodnoty jako je konec řádku, byl pro tento typ zvolen způsob uložení dat zakódováním do formátu Base64.

Příklad hodnoty konfigurace

Příklad jedné hodnoty konfigurace (povolení scanování komunikace v síti ICQ):

```
krnl.networkscanner.imscanner.icq.enabled[BOOL]=T
```

4.3 Návrh a implementace klientského rozšíření

Klientské rozšíření implementuje základní funkčnost rozšíření Group Policy. Jeho hlavní činností je čtení konfiguračních dat ze seznamu GPO objektů a aplikování konfigurace na systém AVG. Další činností je spojení dat z více GPO, kde jsou v zásadě použity základní principy Group Policy, kdy GPO umístěné ke konci seznamu GPO přepisují nastavení předchozích GPO.

Jádro klientského rozšíření

Jak již bylo několikrát zmíněno v předchozím textu, klientské rozšíření je implementováno jako dll knihovna. Všechny hlavní exportované funkce této knihovny jsou implementovány

v souboru cse.cpp - je zde tedy implementována základní funkce `ProcessGroupPolicy`, funkce `DllMain` - vstupní bod do dll knihovny a registrační a odregistrační funkce `DllRegisterServer` a `DllUnregisterServer`.

Funkce `DllMain` je volána při inicializaci dll knihovny. Je v ní implementována inicializace loggerů, což jsou objekty zapouzdřující práci s logy. K samotnému logování je využito loggerů systému AVG, které jsou implementovány v dll knihovně, která je součástí instalace AVG. Pokud inicializace logů selže, vrací funkce `DllMain` hodnotu `FALSE`, čímž dává najevo, že se inicializace nezdařila. Knihovna `userenv.dll` pak zprávu o neúspěšné inicializaci zapíše do Windows Event Logu, čímž je administrátor upozorněn, že je něco špatně s instalací AVG. Pokud se inicializace logu zdařila, jsou veškeré chybové, varovné i informační hlášky, které vznikají při zpracování zásad, zapisovány pomocí loggeru do souboru `avggp.log`, který se nachází ve stejném adresáři jako ostatní logy systému AVG, tedy standardně v adresáři `C:\Documents and Settings\All Users\Data aplikací\avg8\log`.

Implementace zpracování GPO

Funkce `ProcessGroupPolicy` pak podle nastavených příznaků vyhodnotí, zda je nutné aplikovat konfiguraci (pokud není detekována žádná změna od předchozího zpracování zásad, nastavení se znovu neaplikuje). Funkce pomocí cyklu prochází seznam GPO a sestavuje seznam cest k jednotlivým GPT. Tento seznam předává `GPOhandleru`, což je třída zapouzdřující práci s GPO. Seznam GPT je předán pomocí metody `GPOhandler::ReadAndMergeGPO`, která pak přečte nastavení ze všech GPO a všechna nastavení do jednoho seznamu hodnot. Jelikož může být v seznamu více GPO objektů, vznikají konflikty. Jako přednostní hodnota je vybrána ta, která je zpracovávána jako poslední. Samotné čtení souborů je implementováno ve třídě `AvgGpoParser`. V metodě `GPOhandler::ReadAndMergeGPO` se pak tento parser volá na všechny .dat soubory adresáři AVG v rámci všech GPO. Samotné parsování je implementováno v metodě `AvgGpoParser::ParseFile`, kde pro každou konfigurační položku vytváří objekt třídy `AvgGpoParserItem`. Objekt této třídy ma konfigurační hodnotu uloženou jako řetězec. Pro přístup ke konkrétním typům jsou implementovány metody `GetBoolValue`, `GetDWordValue`, `GetStringValue`, `GetDateValue`, `GetMultiStringValue` a `GetByteArrayValue`. Tyto metody převádějí interní typ na konkrétní příslušný typ, který je použitelný pro metody rozhraní `IAvgConfig`. Každý objekt třídy `AvgGpoParserItem` je pak uložen v asociativním poli, kde je pak uložen řetězec obsahující název hodnoty konfigurace a pak samotný ukazatel na objekt `AvgGpoParserItem`.

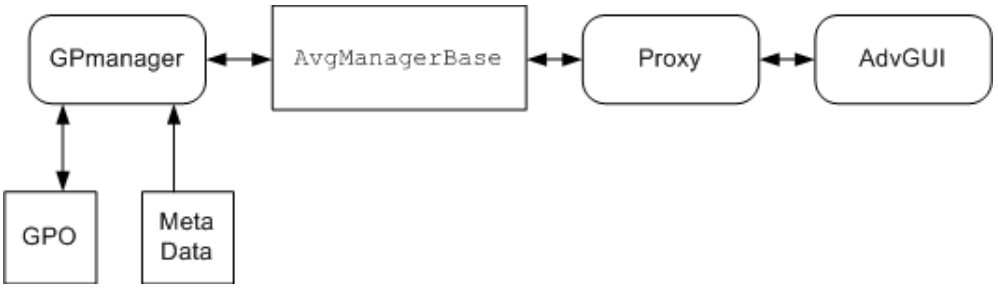
Aplikace konfigurace

Aplikace zásad je implementována v metodě `GPOhandler::ApplyGP`. V této metodě je také implementován přístup ke konfiguraci AVG. Ten je impelementován pomocí interního mechanismu AVG, který je principiálně podobný technologii COM. Tohoto mechanismu je využito k přístupu k rozhraní `IAvgConfigManager`, které zapouzdřuje celou konfiguraci AVG. Pomocí jeho metod lze získat seznam všech kořenů konfigurace, id těchto kořenů a samozřejmě také seznam konfiguračních instancí pro každý kořen. Na základě id rootu a id konfigurační instance lze získat přístup k rozhraní `IAvgConfig`, které zapouzdřuje práci s jednou konfigurační instancí. Toto rozhraní obsahuje metody pro zápis i čtení konfiguračních hodnot pro všechny možné typy použitelné pro konfiguraci. Navíc jsou zde také metody pro nastavení různých příznaků, které znemožní uživateli editaci těchto hodnot skrze uživatelksé rozhraní AVG. Metoda ApplyGP tedy prochází asociativní pole hodnot konfigurací, na základě jména rootu a jména hodnoty získá id hodnoty a pak podle

příslušného typu hodnoty volá konkrétní metodu rozhraní `IAvgConfig` pro uložení konkrétní hodnoty (například pro typ bool volá metodu `putBoolValue`). Pokud dojde při ukládání k chybě, je chybová hláška zapsána do logu.

4.4 Návrh a implementace rozšíření GPOE

Při návrhu rozšíření Group Policy Object Editoru byla snaha o maximální využití již existujících komponent, zejména pak bylo použito komponenty tzv. "Advaced GUI", což je dll knihovna (guiadv.dll), ve které je implementováno uživatelské rozhraní pokročilého nastavení AVG (viz obrázek 4.3). Funkčnost snap-inu je tedy pouze ta, že pomocí rozhraní `IGPEInformation` získá požadované údaje od GPOE (cestu ke GPT), přečte údaje z GPO (pokud se nejedná o nově vytvořený GPO) a předá tyto údaje knihovně guiadv.dll. Samotné předání konkrétní konfigurace se děje pomocí tzv. `Proxy`, což je třída, která zprostředkovává komunikaci s Advanced GUI. Samotné načítání a ukládání dat konfigurace má na starosti tzv. Manager, který vznikne implementací třídy zděděné z abstraktní třídy `CAvgManagerBase`. Navíc manager poskytuje proxy metadata - popis všech konfiguračních hodnot, jejich typy a defaultní hodnoty. Jako proxy vrstva byla při implementaci využita proxy vrstva `CAdminProxyUI` implementovaná v rámci projektu AVG Admin [1](ten pak využívá Protocol Manager, což je také implementace `CAvgManagerBase`, která ovšem čte data ze sítě). Tato architektura je pak stručně zachycena na schématu 4.1.



Obrázek 4.1: Tok dat mezi GPO a AdvGUI

Celý návrh a implementace jsou rozděleny do dvou větších částí: Implementace samotného snap-inu pro Group Policy Object Editor a implementace GPmanageru, který má na starosti načtení a ukládání dat do GPO.

4.4.1 Implementace snap-inu pro GPOE

Při implementaci snap-inu jsem vycházel z příkladů implementací snap-inů pro MMC konzoli, které jsou dostupné v Platform SDK [2]. Zdrojové kódy těchto příkladů jsem pak využil k implementaci základní funkčnosti snap-inu pro GPOE.

[1]AVG Admin je nástroj pro vzdálenou správu AVG, přičemž k nastavení se používá databázový server, ke kterému se připojují klienti AVG přes TCP/IP

[2]Tyto příklady jsou dostupné pouze ve verzi Windows® Server 2003 SP1 Platform SDK a starších, ve Microsoft ® Windows Server® 2003 R2 Platform SDK jsou již dostupné pouze příklady pro konzoli MMC 3.0 napsané v jazyce C#

Implementace snap-inu pro GPOE jako implementace COM serveru

Jak již bylo řečeno, snap-in pro MMC je COM server, který poskytuje přístup MMC konzoli k implementaci několika rozhraní.

Vstupní bod do snap-inu je v souboru BaseSnap.cpp, ve kterém je implementováno rozhraní `IClassFactory` jako třída `CClassFactory`. Voláním metody `CClassFactory::-CreateInstance` je pak vytvořena instance třídy `CComponentData`, což je implementace rozhraní `IComponentData`, která se nachází v souboru CompData.cpp. V této třídě pak je implementována řada metod, které jsou volány konzolí MMC pro práci s tzv. Stromem konzoly, což je panel v levé části konzoly obsahující strom všech položek (anglicky se tato část GUI MMC nazývá Scope Pane).

Pro práci s hlavní části uživatelského rozhraní MMC (anglicky nazývanou Result Pane) slouží rozhraní `IComponent`. které je implementováno jako třída CComponent v souboru Comp.cpp. Instance této třídy je vytvořena v metodě `CComponentData::CreateComponent`, kterou se také pak vrací reference na nově vytvořený objekt MMC konzoli. Mezi nejdůležitější činností implementovaných ve třídě `CComponentData`, je získání přístupu k rozhraní `IGPEInformation`. To je implementováno přesně tak, jak je popsáno v kapitole 3.2, a to zachycením zprávy `MMCN_EXPAND`. Při zpracování této zprávy je také implementováno samotné přidání uzlu AVG do stromu konzoly. Toto přidání je implementováno v metodě `OnExpand`, ve které je vytvořena nová instance třídy `CStaticNode` (implementace je v souboru StaticNode.cpp), která představuje náš uzel AVG. Při inicializaci tohoto uzlu se také vytvářejí položky, které se po rozbalení uzlu AVG zobrazí v Result Pane. V našem případě je to pouze jedna položka s textem "AVG configuration", ale je možné v budoucnu přidat libovolné množství těchto položek. Tato položka je implementována ve třídě CParamNode. Dvojklikem na tuto položku pak dochází k otevření okna konfigurace AVG. Samotný dvojklik je zachycen jako zpráva `MMCN_DBLCLICK` od MMC konzole, která je předána pomocí metody `CComponent::Notify`.

Dále je k položce "AVG configuration" přiřazeno kontextové menu. Toho je dosaženo implementací rozhraní `IExtendContextMenu` v třídě `CComponent`. Příslušné volání metod tohoto rozhraní jsou pak delegovány k objektu třídy `CParamNode`, kde je implementováno přidání položek do menu a reakce na jejich stisknutí. V tomto menu se konkrétně nacházejí dvě položky: "Open AVG configuration", která otevře okno konfigurace stejně jako dvojklik, a položka "Delete all AVG settings from this GPO", pomocí které se vymaže veškeré nastavení AVG z GPO (tedy vymaže veškeré soubory v adresáři AVG v GPT a informuje se GPOE, že došlo ke smazání konfigurace).

Výsledný snapin je na obrázku 4.2.

4.4.2 Implementace GPmanageru

GPmanager je implementace třídy zděděné z třídy `CAvgManagerBase`. V této třídě jsou implementovány všechny čtyři metody, které potřebuje Proxy k úspěšnému načítání a ukládání dat. Jelikož je třeba využít některých funkcí, které byly implementovány pro klientské rozšíření (jako je např načítání dat z GPO), byla při implementaci použita třída GPOhandler, do které byly doplněny věci nutné pro správnou funkčnost s GPmanagerem.

Načtení metadat

První metodou, kterou bylo nutno implementovat v GPmanageru, je metoda `GetMeta-Data`. Tato metoda vrací metadata, tedy popis struktury konfigurace (všechny kořenové

uzly, datové typy hodnot a defaultní hodnoty). Metadata jsou uložena ve dvou souborech .dct: GpMetaFields.dct obsahuje metadata všech konfiguračních hodnot (jejich názvy, id, id příslušného rootu a mnoho dalších údajů) a GpMetaTables.dct obsahuje seznam všech rootů, jejich jména a id. Metoda `GetMetaData` vrací tyto metadata ve formě objektu třídy `CAvgMetaDatabase`. Načtení souborů .dct je implementováno ve statické metodě `CAvgReportBaseFileSerializer::Deserialize`, která načte obsah souborů do objektu třídy `CAvgReport`, který pak lze využít k načtení metadat pomocí metody `CAvgMetaDatabase::DeserializeTablesFieldsFromReports`.

Načtení konfigurace z .dat souborů

Načítání dat z GPO mají na starosti metody `LoadMergedConfig` a `LoadRawConfig`. Tyto metody byly původně vymyšleny pro použití v programu AVGAdmin, kde je nutné zobrazovat i nastavení z více zdrojů, a ty pak spojit do jedno konkrétní konfigurace (existuje globální konfigurace, konfigurace pro skupinu a konfiguraci pro konkrétní stanici), ale jelikož v GPO je uložena vždy je jedna konfigurace, vrací obě metody stejné data. Konfigurační data se vracejí jako pole objektů `CAvgReport`. Tato třída je zjednodušeně řečeno implementací tabulky, kde každý sloupec odpovídá jedné konfigurační hodnotě a každý řádek pak odpovídá jedné konfigurační instanci. Načtení dat do těchto reportů je implementováno v metodě `GPOhandler::ReadConfigToReport`, která opět využívá `AvgGpoParser` k načítání .dat souborů.

Uložení dat do .dat souborů

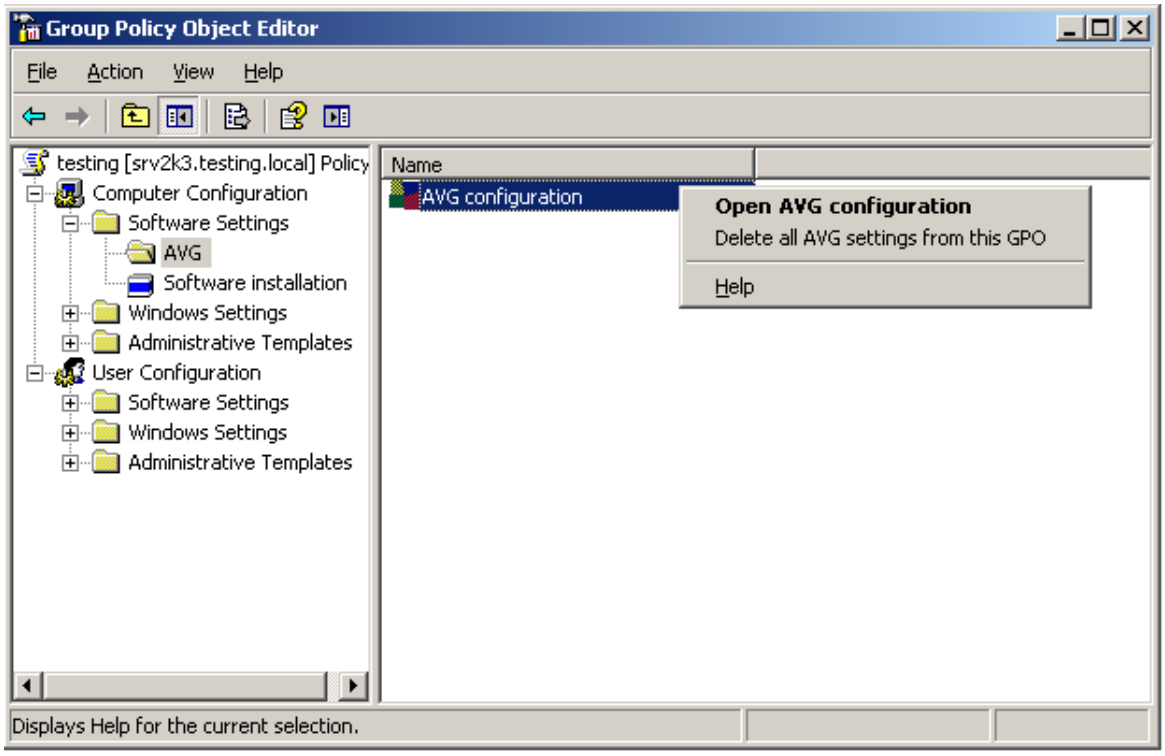
Ukládání dat je řešeno pomocí metody `StoreMergedConfig`, kde jsou data předávaná GPmanageru opět pomocí objektů třídy `CAvgReport`. Samotné ukládání dat je implementováno ve třídě `GPOhandler`. Pomocí metody `SaveValue` se vkládá jedna konfigurační hodnota do mapy konfiguračních hodnot. Metoda `SaveValues` pak uloží konkrétní instanci do .dat souboru.

4.5 Využití Group Policy k instalaci AVG

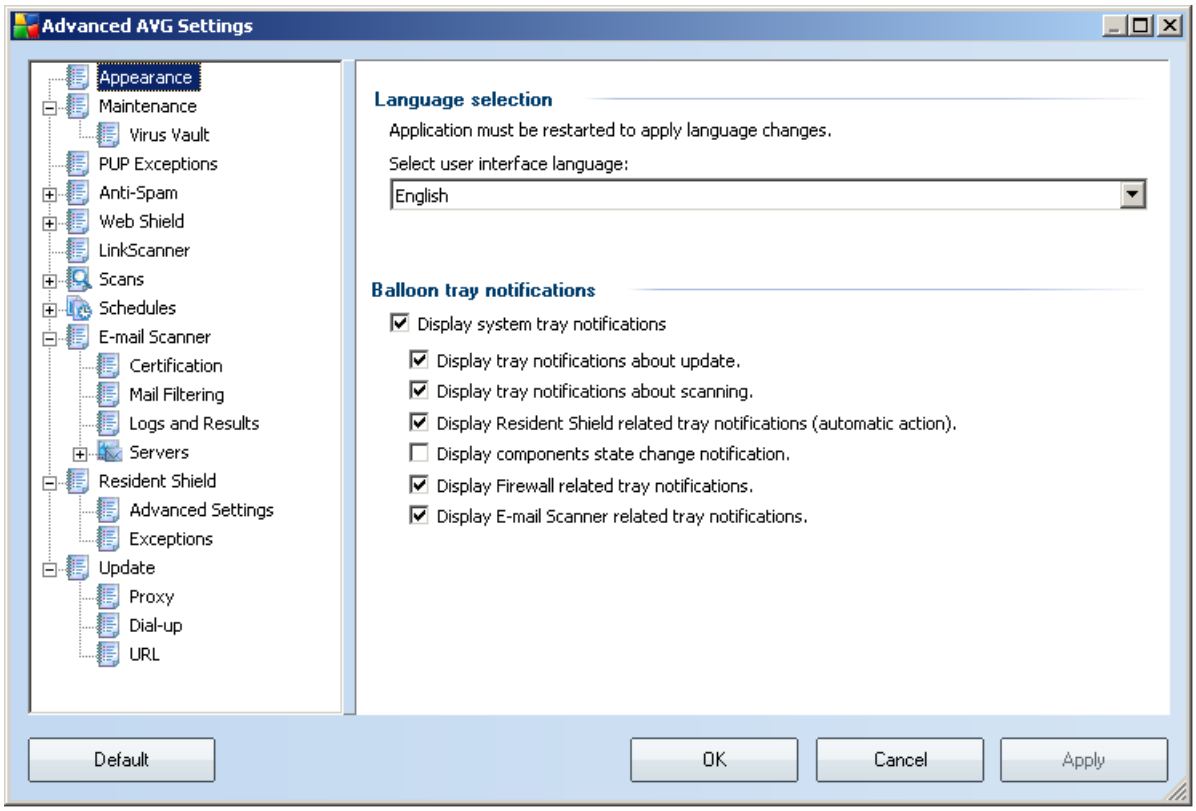
Operační systémy Windows obsahují přeinstalované rozšíření Software Installation, které umožňuje vzdálenou instalaci a odinstalaci softwaru. Programy můžou být pomocí tohoto rozšíření instalovány ve dvou módech: Assigned - přiřazuje aplikaci uživatelům nebo počítačům, software je nainstalován automaticky; Published - lze nastavit pouze pro uživatele, software není nainstalován, ale uživatel si jej může nainstalovat sám pomocí pomocí ovládacího panelu Přidat nebo Odebrat programy.

Tento způsob instalace software ma ovšem jednu "nevýhodu". Protože používá službu Windows Installer, je nutné pro každý program, který chceme instalovat, mít k dispozici MSI balíček. Lze sice využít .zap soubory, což jsou textové soubory, které obsahují cestu k instalaci programu ve forme setup.exe, ale funkčnost instalace je omezena - lze použít pouze možnost published, takže software nelze instalovat automaticky.

Proto zatím nelze použít tuto službu k instalaci AVG, ale v budoucnu je podpora MSI balíčků plánovaná, takže to bude možné.



Obrázek 4.2: AVG snap-in do Group Policy Object Editoru



Obrázek 4.3: Okno pokročilého nastavení AVG

Kapitola 5

Závěr

Cílem této práce bylo nastudovat problematiku GroupPolicy a poté implementovat rozšíření GroupPolicy sloužící ke konfiguraci systému AVG. Tento cíl se podařilo z velké části splnit.

V rámci této práce byla detailně nastudována a popsána architektura GroupPolicy. Byly také nastudovány postupy, jak implementovat klientské rozšíření i rozšíření Group Policy Object Editoru. V kapitole 3.2 bylo popsáno, proč bylo nutné implementovat GPOE snap-in v jazyce C++, ačkoliv se původně v zadání počítalo s implementací v jazyce C#. Tento poznatek vedl k nutnému dalšímu studiu, bylo třeba nastudovat základy technologie COM, jejíž znalost je nutná pro implementaci snap-inu pro konzoli MMC 2.0. Bylo také nutné nastudovat samotnou architekturu konzole MMC 2.0. To také vedlo k částečnému prodloužení doby implementace, na základě čehož se nestihla implementovat podpora RSoP pro obě rozšíření.

Samotná konfigurace antivirového programu AVG 8.0 pomocí GroupPolicy je plně funkční. Rozšíření byla testována v prostředí testovací domény ve virtuální síti vytvořené v programu VMware Workstation, kde řadič domény tvořil Windows Server 2003 R2 a klientskou stanici Windows XP.

Práce na rozšíření budou nadále pokračovat, počítá se hlavně s dokončením implementace (přidání podpory RSoP). V budoucnu bude také přidána podpora pro konfiguraci dalších částí systému AVG, zejména firewallu. Poté bude nutné podrobit tyto rozšíření intenzivním testům, hlavně v prostředí různých verzí systémů Windows, jako je například nově uvedený Windows Server 2008. Bude také dobré prostudovat novinky spojené s GroupPolicy, které Microsoft uvedl s touto verzí operačního systému, a případně jich i využít k vylepšení stávající implementace. Po testech přijde i na řadu testování v reálné počítačové síti a snad také by mělo dojít k uvolnění rozšíření na komerční trh kde budou zařazeny do jednoho z produktů AVG.

V rámci této bakalářské práce jsem se nejen seznámil s jednou z nejpoužívanějších technologií využívaných ke správě v sítích založených na systémech Microsoft Windows, ale také jsem měl šanci proniknout do vývoje komerčního produktu, jakým je AVG Internet Security, což pro je mě jako studenta neocenitelná zkušenost.

Literatura

[1] Robbie Allen and Alistair G. Lowe-Norris. *Active Directory: implementace a správa Microsoft Active Directory*. Grada Publishing, 2005. ISBN 8024709732.

[2] Michael Dunn. Introduction to COM. `http://www.codeproject.com/KB/COM/comintro.aspx`.

[3] Jeff Glatt. COM in plain C. `http://www.codeproject.com/KB/COM/com_in_c1.aspx` ; `http://www.codeproject.com/KB/COM/com_in_c2.aspx`.

[4] Darren Mar-Elia, Derek Melber, William Stanek, and MS Group Policy Team. *Microsoft WINDOWS Group Policy Guide*. Microsoft Press, 2005. ISBN 978-0735622173.

[5] Microsoft. MSDN Library. `http://msdn2.microsoft.com/en-us/library/default.aspx`.

[6] Microsoft. Windows Server TechCenter. `http://technet2.microsoft.com/`.

[7] Simon Robinson. *Professional ADSI Programming*. Wrox Press, 1999. ISBN 1-961002-26-2.

[8] WWW stránky. GPOGUY.COM - The Group Policy Hub. `http://www.gpoguy.com/`.

[9] Craig Tunstall and Gwyn Cole. *Developing WMI Solutions: A Guide to Windows Management Instrumentation*. Addison-Wesley Professional, 2002. ISBN 0-201-61613-0.

Příloha A

Paměťové médium

K bakalářské práci je přiloženo paměťové médium (DVD) obsahující elektronickou verzi technické zprávy a zdrojové kódy klientského rozšíření i rozšíření Group Policy Object Editoru.

Příloha B

Návod k použití

Pro správu Group Policy doporučuji použití konzole Group Policy Management Console. Pokud GPMC není nainstalována, lze ji získat zde: http://www.microsoft.com/downloads/details.aspx?FamilyId=0A6D4C24-8CBD-4B35-9272-DD3CBFC81887.

Pro použití konfigurace AVG pomocí GroupPolicy se předpokládá, že obě rozšíření i antivirus AVG jsou správně nainstalovány. Postup pro správnou instalaci je uveden v souboru readme.txt na přiloženém paměťovém médiu.

Pro vytvoření nové konfigurace nebo editaci stávající konfigurace tedy spustíme GPMC (např. pomocí Start → Run → gpmc.msc). Ve stromu v levé části konzole vybereme příslušný kontejnerový objekt, který obsahuje účty počítačů, pro které chceme vytvořit novou konfiguraci AVG. Kliknutím pravým tlačítkem na tento objekt vyvolá kontextové menu, zde zvolíme položku *Create and Link a GPO here...* To zaručí vytvoření nového GPO objektu, který je zároveň automaticky přilinkován k vybranému objektu. Kliknutím pravým tlačítkem na nově vytvořený GPO nebo nově vytvořený link vyvolá další kontextové menu, kde zvolíme položku Edit. Tímto se otevře Group Policy Object Editor. V levé rozbalíme položku *Computer configuration* pak rozbalíme položku *Software settings* a následně i položku AVG. V pravé části konzole se pak dvojklikem na položku *Open AVG configuration* otevře konfigurační okno pokročilého nastavení AVG. V tomto okně pak nastavíme požadované hodnoty konfigurace. Stisknutím tlačítka OK se konfigurace uloží do GPO. Tímto je konfigurace vytvořena. Konfigurace bude uplatněna při dalším aplikaci GroupPolicy na klientských stanicích.