ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra telekomunikační techniky

Autentizační mechanismy v heterogenních sítích

květen 2015

Diplomant: Jiří Tišler

Vedoucí práce: Ing. Tomáš Vaněk, Ph.D.

Čestné prohlášení

Prohlašuji, že jsem zadanou diplomovou práci zpracoval sám s přispěním vedoucího práce a konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé diplomové práce nebo její části se souhlasem katedry.

Datum: 11. 5. 2015

.....

podpis diplomanta

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Tišler Jiří**

Studijní program: Komunikace, multimédia a elektronika
Obor: Sítě elektronických komunikací

Název tématu: **Autentizační mechanismy v heterogenních sítích**

Pokyny pro vypracování:

Seznamte se s autentizačními mechanismy používanými v rozsáhlých počítačových sítích. Popište možnosti předávání autentizačních informací pomocí různých protokolů (LDAP, AD, Kerberos) a jejich vzájemnou spolupráci. Realizujte systém jednotné správy a autentizace počítačů/uživatelů na katedře telekomunikační techniky. Na základě praktické realizace vytvořte vzorovou sadu doporučení pro centralizaci správy aplikovatelnou na jiných pracovištích ČVUT.

Seznam odborné literatury:

[1] RFC4511, http://tools.ietf.org/html/rfc4511
[2] RFC4120, http://www.ietf.org/rfc/rfc4120.txt
[3] Desmond B.: Active Directory: Designing, Deploying, and Running Active Directory, O'Reilly Media; Fifth Edition edition, 2013, ISBN: 1449320023

Vedoucí: Ing. Tomáš Vaněk, Ph.D.

Platnost zadání: do konce letního semestru 2015/2016

L.S.

prof. Ing. Boris Šimák, CSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 13. 11. 2014

Anotace:

Tato diplomová práce zkoumá spolupráci autentizačních mechanismů v rozsáhlých počítačových sítích. Jsou prozkoumány a zdokumentovány i možnosti využití stávající infrastruktury pracovišť ČVUT a vydána sada doporučení pro centralizovaný způsob správy koncových stanic a integraci Microsoft Active Directory s operačními systémy Linuxové rodiny.

Klíčová slova:

Active Directory, Linux, systémové inženýrství, kerberos, ldap, realm, správa stanic

Summary:

This diploma thesis examines the interoperability of authentication mechanisms on large networks. The different ways of usage of current workplace infrastructure at CTU were examined and documented. A set of recommendations was issued on how to manage workstations and integrate Microsoft Active Directory with Linux-family operating systems.

Index Terms:

Active Directory, Linux, systems engineering, kerberos, ldap, realm, workstation management, interoperability

Obsah

1 Seznam zkratek

AD	Active Directory
AS	Autentizační služba
DNS	Domain Name System
KDC	Key Distribution Center
LDAP	Lightweight Directory Access Protocol
SPN	Service Principal Name
SQL	Structured Query Language
TGS	Ticket-granting service
TGT	Ticket-granting ticket
UPN	User Principal Name
WDS	Windows Deployment Services

2 Úvod

Systémové inženýrství, neboli „disciplinovaný přístup pro definici, implementaci, integraci a provoz systému (produktu či služby)“ (NASA, 2014), může být fascinující při účasti na celém životním cyklu systému. Tato diplomová práce se však zabývá převážně integrací a provozem systému správy a autentizace jednotlivých kateder a fakult univerzity ČVUT, který byl definován a implementován před několika lety a od té doby se paralelně vyvíjel různými směry a přizpůsoboval aktuální poptávce. Práce pojednává nad smyslem, možností sjednocení a zefektivnění procesu správy počítačové sítě v rámci univerzity ČVUT.

Praktická realizace vychází z potřeb katedry telekomunikační techniky, kdy bylo zapotřebí zavést sofistikovanější systém správy výpočetní techniky, neboť řadu úkonů s touto činností spojených vykonávali přímo pedagogičtí pracovníci. Realizace spojená vypracováním této práce má proto za cíl nejen zlepšit kvalitu poskytovaných informačních služeb na katedře telekomunikační techniky, ale zároveň odhalit různé možnosti způsobu práce se systémem, které dosud nebyly plně využity. Dílčími cíli bylo popsat autentizační mechanismy, analyzovat možnosti jednotlivých protokolů a na jedné z možných implementací ukázat praktické řešení. Nabyté poznatky pak shrnout do sady doporučení pro centralizaci správy aplikovatelnou i na jiných pracovištích.

3 Předávání autentizačních informací

V této části je pojednáváno o autentizaci, autentizačních protokolech a jsou vymezeny klíčové pojmy, jako například Adresářová Služba.

3.1 Autentizace

Autentizace, neboli proces ověření pravosti či identity (Oxford University Press), je jeden z klíčových mechanismů, na kterém závisí všechny systémy pracující s důvěrnými informacemi. Počítačová síť rozhodně mezi tento typ systému patří nehledě na její rozsáhlost. Z pohledu autentizace budeme za rozsáhlou považovat takovou síť, která obsluhuje více administrativních celků a která kombinuje větší množství autentizačních mechanismů, přístupových bodů (zařízení) a autentizačních protokolů.

3.2 Autentizační protokoly a jejich spolupráce

Pro účely této práce zařadíme mezi „autentizační protokoly“ nejen protokoly přímo zajišťující autentizaci, ale rovněž protokoly, které tuto autentizaci bezprostředně využívají k autorizaci. Autorizace je akt udělení pověření k provedení určité akce (Oxford University Press), po ověření identity se tedy na základě dat z adresářové služby určuje, zda obdrží klient autorizaci k provedení dané akce, např. přístup na sdílený disk, tisk, vstup do webové aplikace.

3.3 Adresářová služba

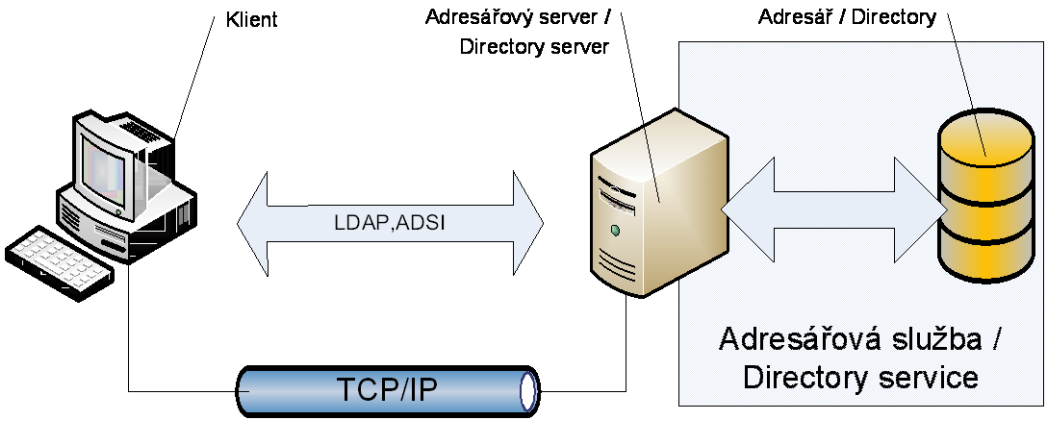
Vhodná organizace dat je pro centralizovanou správu klíčová. Data lze hierarchicky uchovávat ve struktuře zvané adresář (Directory). V běžném životě lze takovou strukturu

přirovnat například k telefonnímu seznamu, ale v kontextu informačních technologií se jedná nejčastěji o specializovanou databázi.

Tato databáze se často liší od běžných víceúčelových relačních databází tím, že je navržena pro ukládání spíše statických a málo proměnných dat dle předem definovaného schématu, tedy pro rychlé a časté čtení, ale delší a pomalejší zápisy. Další rozdíl je v typu přístupu: k práci s daty obvyklých relačních databází je používán strukturovaný dotazovací jazyk (SQL), kdežto k přístupu a práci s adresářovými službami se nejčastěji (ale ne výhradně) používá LDAP (Lightweight Directory Access Protocol), který umožňuje standardizovanou komunikaci mezi LDAP klientem a LDAP serverem.

LDAP server je hostitelem adresářové služby, jejímž úkolem je na základě LDAP či jiných příkazů zajišťovat fyzický zápis a čtení dat spolu s ostatními administrativními úkony prováděnými nad adresářem. Standardizace způsobu přístupu k datům rozdílných adresářových služeb poskytuje velkou výhodu v heterogenním prostředí.

Pro ilustraci jednoduché vazby mezi protokoly můžeme použít následující obrázek:



Obr. 3.3.1 - Adresářová služba a protokoly

Zdroj: Vlastní zpracování

3.3.1 LDAP

LDAP neboli „Lightweight Directory Access Protocol“ zajišťuje standardizovaným způsobem komunikaci s adresářovou službou a tedy zprostředkovaně i s adresářem. Jedná se o „odlehčeného“ nástupce protokolu DAP (Directory Access Protocol), odvozeného ze standardu X.500, kde mezi hlavní rozdíly patří zjednodušení přístupu k adresářovým službám a adaptace pro TCP/IP síťový model, neboť DAP je založen na OSI (Open Systems Interconnection) modelu.

LDAP předpokládá u adresářové služby předem definovaný datový model, tedy data (záznamy) ukládaná ve stromové struktuře dle určitého schématu. Každý záznam se skládá ze dvojic atribut-hodnota (některé atributy mohou mít více hodnot) a má v rámci celého adresáře své unikátní rozlišovací jméno (distinguished name), které zohledňuje i jeho pozici ve stromové struktuře.

3.3.2 Kerberos

Kerberos, jakožto síťový autentizační protokol, vyvinutý v prvotních třech verzích výhradně pro účely projektu Athena (základy distribuované informační technologie) na Massachusettském technologickém institutu (MIT), dostal většinu své nynější podoby ve své páté verzi, která byla standardizována roku 1993 jako RFC1510 (Kohl, a další, 1993). Své jméno obdržel po bájném tříhlavém psovi Kerberovi (Cerberus) z Řecké mytologie, možná právě díky svojí třístranné orientaci mezi uživatelem, Kerberos serverem a aplikačním serverem.

Od roku 1993 se stal Kerberos pro své vlastnosti výchozím autentizačním mechanismem pro řadu systémů, v čele s Microsoft Windows. V roce 2005 byl pak tento protokol upraven normou RFC4120 (Neuman, et al., 2005) a v dnešní době v hojné míře na standardním TCP/UDP portu 88 poskytuje autentizaci mnoha platformám. Nejdříve naznačíme zjednodušeným způsobem funkci protokolu tak, jak je popsána v RFC4120, ovšem s grafickými doplňky a zaměřením na použití s Microsoft Active Directory, neboť je v tomto kontextu Kerberos rozšířen o další služby:

Kerberos poskytuje možnost ověření identity tzv. security principalu (uživatele koncové stanice, či serveru na síti) na otevřené (nezabezpečené) síti. Kerberos při ověřování nezávisí na hostitelském operačním systému, na důvěře hostitelské adrese, na fyzickém zabezpečení všech hostitelských strojů na síti a naopak předpokládá že každý vyslaný paket může být při průchodu sítí volně přečten, upraven, či vložen. Kerberos provádí autentizaci za těchto podmínek za pomoci ověření důvěryhodné třetí strany s použitím tradiční kryptografické metody (shared secret key). Dá se využít i prostoru pro rozšíření pro použití dalších kryptografických metod.

Postup autentizace popsaný v RFC4120 (Neuman, et al., 2005) a článku pana Boušky (Bouška, 2014) doplníme ještě grafickými prvky pro lepší znázornění:

- Obr. 3.3.2 představuje standardního Kerberos klienta, tedy koncovou stanici schopnou komunikovat s Kerberos serverem pomocí standardizovaného protokolu. Budeme předpokládat, že tuto stanici použije uživatel nejprve pro přístup do samotného operačního systému a následně pro přístup k síťové službě na jiném serveru, rovněž využívající Kerberos protokolu.

Obr. 3.3.2 - Kerberos klient

Zdroj: Vlastní zpracování

- Obr. 3.3.3 představuje Kerberos server, který slouží jako Key Distribution Center (KDC), kde beží autentizační služby (AS), jako např. Ticket-Granting Service

(TGS). V případě zvláštní implementace Kerbera u produktu Microsoft Active Directory je tato služba přímo navázána i na adresářovou službu a pracují společně. Ve Windows implementacích běží tyto služby společně na serveru zvaném Domain Controller (doménový řadič), který provádí autentizaci a autorizaci v jednom.

Obr. 3.3.3 - Kerberos server, KDC, DC

Zdroj: Vlastní zpracování

- Obr. 3.3.4 představuje obecně aplikační server, na kterém běží služba, ke které klient požaduje přístup. Může se jednat o webový server, file system server, či jiný. Tento server může být součástí jednoho realmu, či Active Directory domény, ale také to může být naprosto oddělený server v internetové síti. Pro korektní funkci stačí, aby byla daná služba a dns jméno serveru s heslem zaregistrovány v adresáři KDC a aby měl tento server keytab se svým heslem, které se shoduje s heslem zaregistrovaným pro danou kombinaci jména dns a služby v adresáři.

Obr. 3.3.4 - Aplikační server

Zdroj: Vlastní zpracování

V prvním kroku chce uživatel přistoupit k operačnímu systému koncové stanice. Ta musí být předem nakonfigurována pro použití správného kdc serveru (nebo alespoň DNS služby, která jí tuto adresu poskytne). Zadá doménu (v případě AD),realm (v ostatních případech), kam se chce přihlásit, například CVUT.CZ a své uživatelské jméno. Kerberos vždy spolupracuje s nějakým typem adresáře, který uchovává tzv. User Principal Name, neboli UPN. To je kombinace uživatelského jména a realmu spojená zavináčem, označovaná jako Internet-style login name a založena na RFC 822.

UPN se skládá dvou částí: UPN prefix / předponou, uživatelským jménem (část před zavináčem) a UPN suffix / příponou, DNS doménové jméno (část za zavináčem). (Microsoft, 2015) Standardně by se tedy uživatel se jménem „novak“ měl do domény „CVUT.CZ“ přihlašovat následujícím UPN: novak@CVUT.CZ. V praxi je však možné použít pouze uživatelské jméno, pokud je doména doplňována automaticky, tedy „novak“. V závislosti na použité konfiguraci se ještě (zejména v prostředí Microsoft Windows) používá:

- tzv. NetBIOS jméno domény, což je ta část DNS jména před první tečkou. Zde tedy „CVUT“ s výsledným tvarem přihlášení: CVUT\novak
- DNS jméno domény, tedy CVUT.CZ, a to ve tvaru: CVUT.CZ\novak.

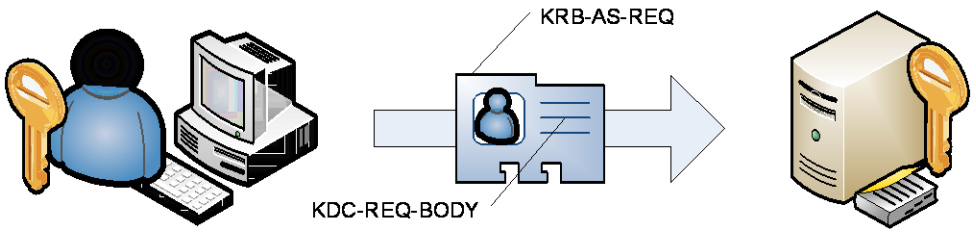
Ve výsledku je tedy možné se například na stanici, která je součástí Active Directory domény CVUT.CZ lze přihlásit těmito způsoby:

- novak@CVUT.CZ (UPN)
- CVUT.CZ\novak (DNS název domény\sAMAccountName)
- CVUT\novak (NetBIOS název domény\sAMAccountName)
- cokoli@neco.cz (UPN přímo definovaný v atributu userPrincipalName v adresáři)

Kde sAMAccountName je v rámci domény unikátní identifikátor každého objektu a UPN je tedy unikátní identifikátor v rámci forestu. (Bouška, 2014)

Kombinace UPN a uživatelského hesla je stanicí zahashována jednosměrnou hashovací funkcí (UPN se přidává, aby neměli dva uživatelé se stejným heslem i stejný výsledný hash) a uložena v paměti stanice pro pozdější použití. Tento hash se nazývá Secret Key.

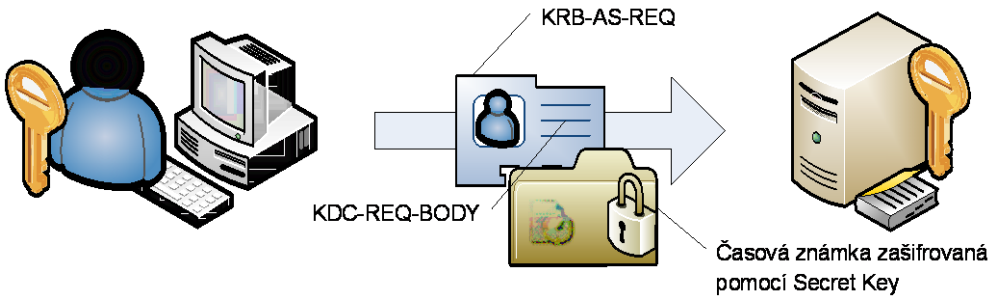
V první fázi však klient odesílá v běžně čitelném textu zprávu (KRB-AS-REQ), která obsahuje pouze KDC-REQ-BODY, tedy informace charakterizující klienta (UPN), realm, službu ke které chce přistoupit a jakým způsobem může provést autentizaci (způsob šifrování, oboustranná autentizace, atd). Zpráva může být odeslána bez šifrování, jelikož neobsahuje ani Secret Key, ani žádnou jinou formu uživatelského hesla. Na obrázku níže je rovněž znázorněn i zatím nepoužitý Secret Key, který mají obě strany.



Obr. 3.3.5 - 1. Krok autentizace

Zdroj: Vlastní zpracování

V běžném scénáři ovšem Autentizační Služba požaduje tzv. pre-autentizaci, která ale v požadavku zatím chybí. Odpoví proto klientovi chybovou zprávou KRB-ERROR s kódem KRB5KDC_ERR_PREAUTH_REQUIRED (25) a ten tedy sestaví nový KRB-AS-REQ požadavek, který tentokrát bude v části pro preautentizaci (padata) obsahovat ještě časovou známku zašifrovanou pomocí SecretKey:

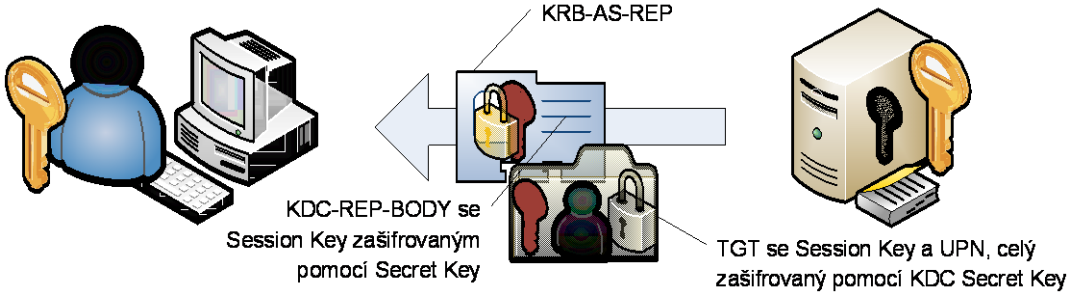


Obr. 3.3.6 - 2. Krok autentizace (preautentizace)

Zdroj: Vlastní zpracování

AS tedy ze zprávy vyčte UPN klienta a zkontroluje, zda existuje v adresáři. Pokud ano, tak pomocí jeho Secret Key (který je AS dostupný) rozšifruje časovou známku obsaženou ve zprávě. Tato operace je možná díky použití symetrického šifrování, kdy je jeden klíč použit jak pro zašifrování, tak pro dešifrování zprávy. Časová známka se dle standardního nastavení nesmí lišit od aktuálního serverového času o více než 5 minut a také to musí být nejnovější dostupná časová známka, která od tohoto klienta přišla.

V případě, že je časová známka v pořádku, vygeneruje AS tzv. Session Key, který bude platný pouze pro danou relaci. Tento klíč navíc zašifruje pomocí Secret Key, aby jej mohl klient později použít při komunikaci s KDC. Dále je do odpovědi vložen Ticket-Granting Ticket, neboli TGT, ve kterém je znovu ukryt Session Key, nicméně TGT je šifrován pomocí tajného klíče, který zná pouze KDC (KDC Secret Key). Z pohledu klienta je tedy TGT jakási „černá skříňka“, jejíž obsah nemůže modifikovat, ale přikládá ji při komunikaci s KDC.

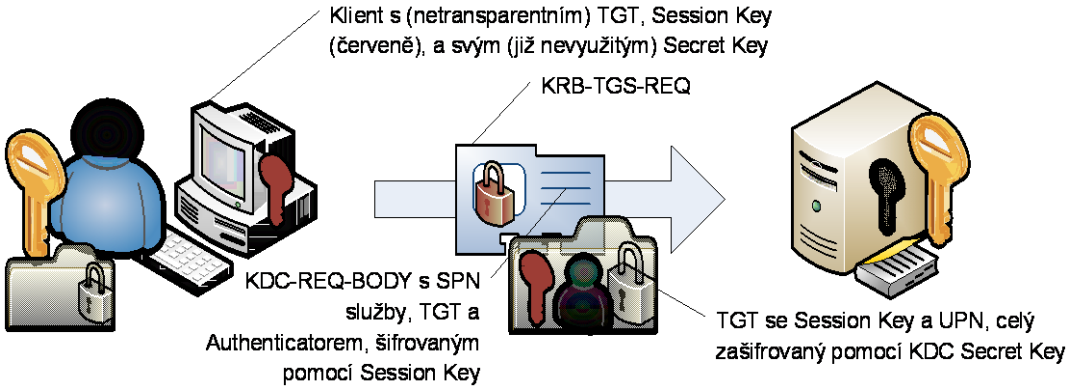


Obr. 3.3.7 - Obdržení TGT

Zdroj: Vlastní zpracování

TGT kromě Session Key obsahuje i jméno klienta, adresu a dobu platnosti. Při komunikaci si klient s KDC pouze předává TGT. Po obdržení TGT jej KDC pomocí svého tajného klíče rozšifruje a pouze pro ověření porovná Session Key uvnitř TGT s tím, kterým byla zašifrována část komunikace. Vyplývá z toho tedy, že KDC nemusí jednotlivé Session Key držet v paměti, ale nachází je uvnitř TGT.

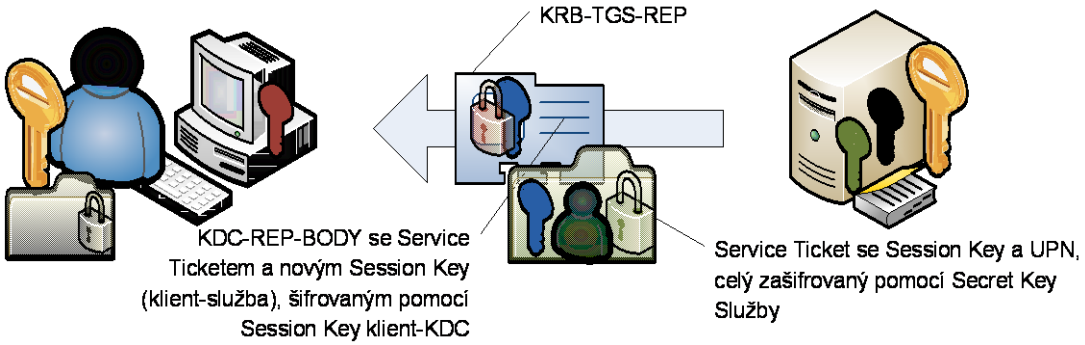
V případě, že chce pak klient přistoupit ke službě na webovém serveru, opět se obrátí na svůj KDC, tentokrát však ne na Authentication Service, ale na Ticket-Granting Service, kam odešle ve zprávě KRB-TGS-REQ spolu s TGT ještě údaje o službě, tedy mj. jméno služby a serveru, kde je služba spuštěna. Tato kombinace se nazývá Service Principal Name, neboli SPN. K tomu je do těla zprávy ještě přidán tzv. Authenticator, což je v podstatě identifikace klienta a časová známka. Authenticator je zašifrovaný pomocí Session Key.



Obr. 3.3.8 - Žádost o autentizaci u služby

Zdroj: Vlastní zpracování

Pokud je vše v pořádku a KDC najde podle SPN konkrétní službu ve svém adreáři (a tím pádem i její Secret Key), vydá KDC takzvaný Service ticket, který je šifrovaný právě pomocí Secret Key služby a obsahuje především UPN žádajícího uživatele, SPN a nově vygenerovaný Session Key, který mezi sebou posléze bude používat klient a služba. Službě je tento klíč předán v Service ticketu a klientovi v odpovědi na žádost, v poli šifrovaném pomocí Session Key, který mezi sebou používá ke komunikaci klient a KDC.

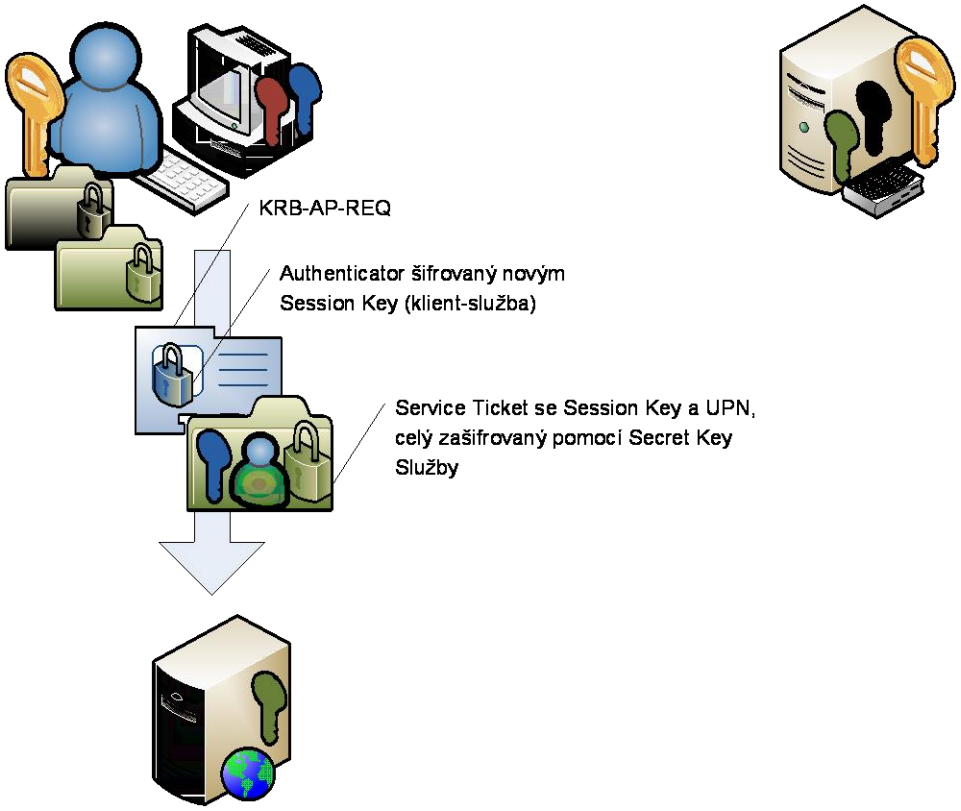


Obr. 3.3.9 - Udělení Service Ticketu a nového Session Key

Zdroj: Vlastní zpracování

Klient po přijetí zprávy rozšifruje nový Session Key, který použije pro pozdější komunikaci se službou. Klient nyní vlastní i Service ticket, který je však pro něj netransparentní (je určen pro službu).

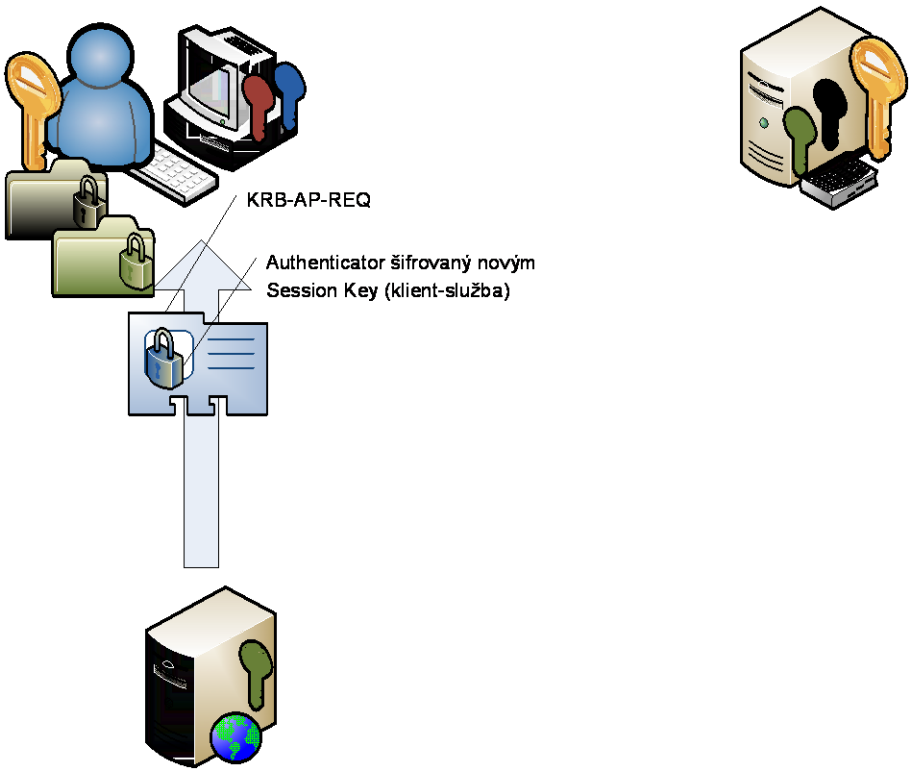
Pro autentizaci u služby ji klient odešle (po předchozím vyjednání autentizace pomocí Kerberos protokolu) Service ticket a nový Authenticator, šifrovaný pomocí Session Key klient-služba:



Obr. 3.3.10 - Autentizace u služby pomocí Service ticketu

Zdroj: Vlastní zpracování

Služba rozšifruje Service ticket pomocí předem stanoveného Secret Key služby, který zná jen služba a KDC. Ze Service ticketu obdrží UPN a Session Key, který bude používat při komunikaci s klientem a klient je tedy úspěšně autentizován. Pokud je vyžadována oboustranná autentizace, odešle ještě služba klientovi zpět Authenticator, šifrovaný vzájemným Session Key:



Obr. 3.3.11 - Dokončení vzájemné autentizace

Zdroj: Vlastní zpracování

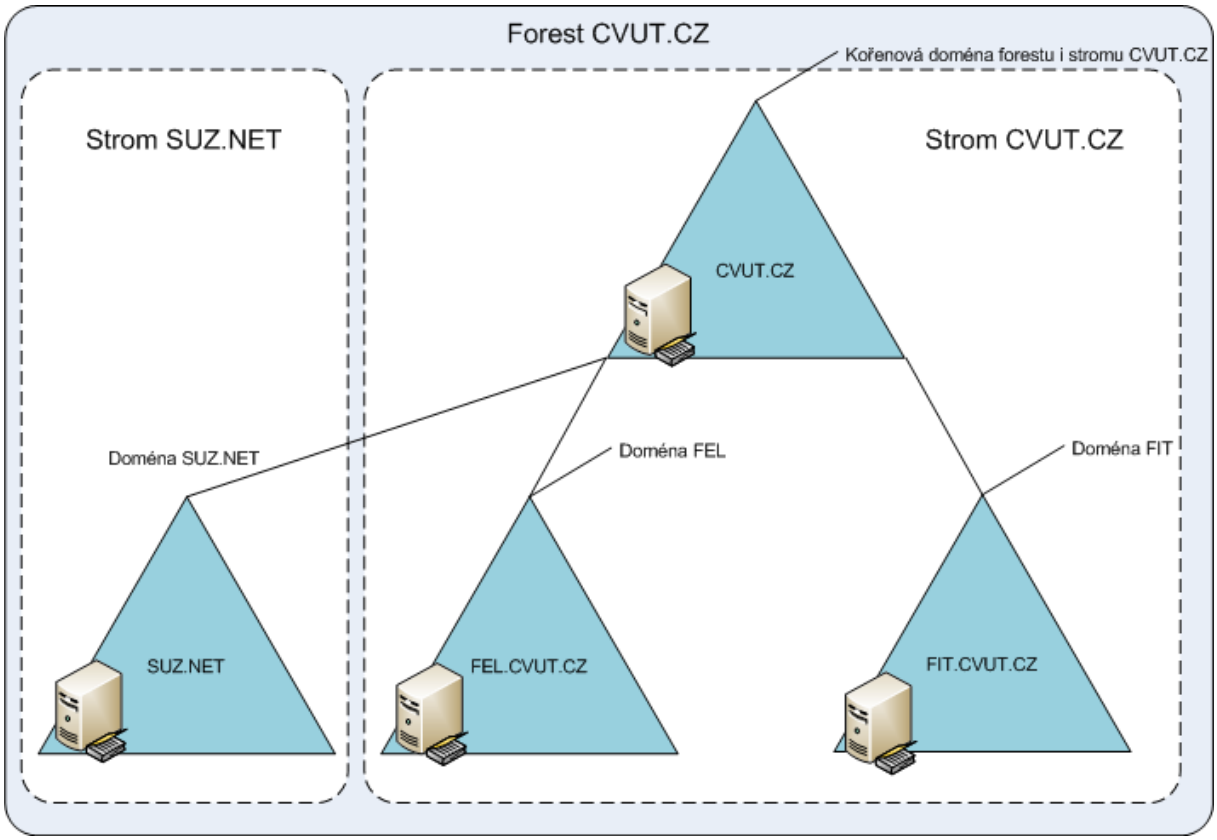
Klient tedy mohl provést vzájemnou autentizaci s KDC serverem i aplikačním serverem.

3.3.3 Active Directory

Active Directory (AD) je název proprietární technologie společnosti Microsoft, která zahrnuje mimo jiné i protokoly LDAP a Kerberos a spojuje je v tzv. „aktivní“ adresářovou službu. Tyto protokoly spolupracují s řadou rozšíření: například u protokolu Kerberos je využito volných polí (Extension fields) pro předávání informací spojených s autorizací. Kerberos Service ticket v doméně Active Directory tedy navíc mj. obsahuje informaci o tom, jakých skupin je uživatelský účet členem. Active Directory Domain Services je sada služeb, které pracují na serveru, který se formálně nazývá Domain Controller (DC). Tento server je bezpodmínečně hostitelem adresářových služeb (dostupných pomocí protokolu LDAP a ADSI – Active Directory Service Interface), Kerberos autentizačních služeb (AS, KDC, TGS) a služeb DNS, přestože v jiných prostředích mohou tyto služby fungovat odděleně (každá na jiném serveru).

Při založení první domény se zároveň vytváří tzv. AD forest. To je kořenová doména, struktura, která obsahuje subdomény, tzv. domain trees, které ještě mohou mít své dceřiné domény, tzv. child domains. Forest ohraničuje domény a kontejnery, které mají stejná schémata a sdílejí stejnou logickou strukutru. Standardně jsou v rámci jednoho forestu sdíleny informace mezi logickými kontejnery (doménami) pomocí vzájemné důvěry (two-way trust).

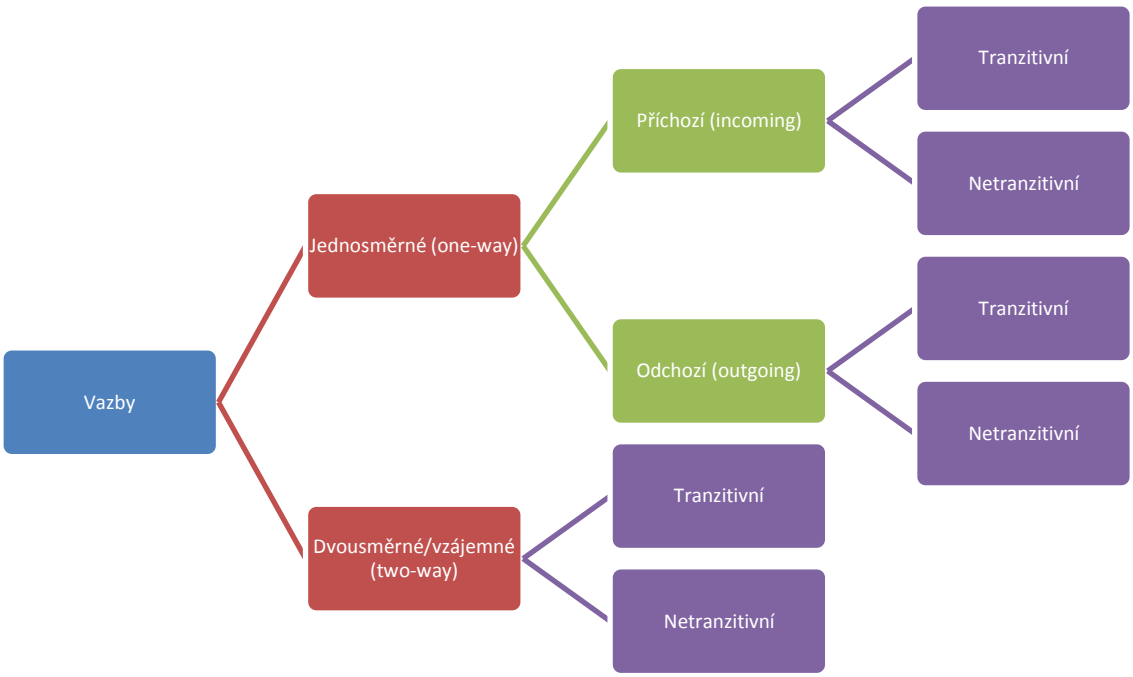
Příklad takového forestu je naznačen na Obr. 3.3.12:



Obr. 3.3.12 - Struktura AD forestu

Zdroj: Vlastní zpracování

Vazby důvěry mezi doménami se dělí dle Grafu 1:



Graf 3.3.1 - Dělení vazeb mezi doménami

Zdroj: Vlastní zpracování

Tranzitivita vazeb určuje, zda daná vazba platí pouze pro dvě domény, či domény na ně navázané. Dvousměrná vazba mezi doménami A a B značí, že uživatelé z domény A se mohou autentizovat i v doméně B a naopak. Pokud je vazba pouze jednosměrná, záleží na jejím směru:

Pro (jednosměrnou) odchozí vazbu z domény A do domény B (z pohledu domény B je tato vazba příchozí) platí, že doména A „důvěřuje“ doméně B a tudíž mohou uživatelé z domény B přistupovat ke zdrojům v doméně A. Symbolicky se tento typ vazby značí šipkou od A směrem k B, kde šipka značí „směr důvěry“, tedy A ->B. Odchozí vazba z domény B do domény A pak umožňuje autentizaci analogicky opačným způsobem, tedy klienti z domény A mohou využívat zdroje v doméně B.

Při vytvoření dceřiné domény se zároveň automaticky vytváří dvousměrná tranzitivní vazba mezi mateřskou a dceřinou doménou. To platí i pro další dceřinou doménu atd. Každá doména však musí bezpodmínečně mít alespoň jeden doménový řadič. V praxi se však kvůli redundanci a eliminaci tzv. „single point of failure“, tedy jediného slabého místa, používá minimálně dvojice doménových řadičů na každou doménu, z nichž jeden by měl být separátní fyzický stroj určený pouze k této funkci. Pokud existuje více doménových řadičů, tak mezi sebou doménové informace replikují dle svých rolí. Jedna z nejvýznamnějších rolí, kterou může doménový řadič zastávat je tzv. Global Catalog. Standardně udržuje každý doménový řadič informace o doméně, ve které se vyskytuje spolu se schématem a konfigurací forestu. Global Catalog udržuje ve své databázi všechny objekty ze všech domén ve forestu. Kromě plné, zapisovatelné databáze své vlastní domény (pro kterou je autoritativním doménovým řadičem) udržuje ještě kopii některých atributů všech objektů ze všech domén ve forestu, která je pouze pro čtení, ale velice usnadňuje vyhledávání napříč doménami. Tato kopie je udržována v aktuálním stavu pomocí replikačních služeb Active Directory. Synchronizují se pouze ty parametry, dle kterých je nejčastěji vyhledáváno. Sadu těchto atributů lze manuálně změnit editací schématu. (Microsoft, 2014)

Další role spadají do kategorie tzv. „Operations Masters“ (Microsoft, 2014) a existují kvůli zachování integrity dat při replikaci a prevenci duplikátů. V praxi to znamená, že nekteré operace jsou vždy prováděny pouze na serveru, který je pro tento typ operace zvolen.

Jsou to:

- Schema master (jediný pro celý forest)
- Domain naming master (jediný pro celý forest)
- Primary Domain Controller (PDC) emulator (jediný pro každou doménu)
- Infrastructure master (jediný pro každou doménu)
- Relative ID (RID) master (jediný pro každou doménu)

Kromě těchto rolí existuje ještě dělení dle způsobu funkce na klasický DC a Read-Only Domain Controller, zkráceně RODC. RODC neumožňuje zápis do adresáře, ale pomáhá rozložit pracovní zátěž pro ostatní doménové řadiče tím, že poskytuje veškeré služby jako klasický DC, kromě služeb vyžadujících zápis do adresáře. RODC navíc poskytuje další redundanci a typicky se umisťuje do lokalit vzdálenějších od centrálních doménových řadičů, kde je horší konektivita a není zaručena fyzická bezpečnost stroje (RODC nemusí držet databázi hashů uživatelských hesel).

Přívlastek „aktivní“ si adresářová služba Active Directory vysloužila ještě díky nástroji pro správu skupinových zásad (Group Policies). Pomocí tohoto nástroje, který je nativní součástí Active Directory Domain Services, lze jednoduše spravovat nastavení a chování politik domény, uživatelů, tiskáren, sdílených úložišť, softwaru, serverů i koncových stanic s operačním systémem Microsoft Windows.

Skupinové zásady se definují v takzvaných objektech skupinových zásad (Group Policy Objects), které se dělí na uživatelské a počítačové. V tomto kontextu může být počítačem myšlena jak koncová stanice, tak server. Tyto objekty lze pak přiřadit k jednotlivým organizačním jednotkám (hierarchické oddíly adresáře - menší administrativní celky) a tím se aplikují na všechny objekty daného typu v této organizační jednotce a standardně i na objekty ve všech podřízených organizačních jednotkách. Dopad politik lze ještě dále filtrovat dle uživatelských či počítačových skupin, předem definovaných v Active Directory. Zásady jsou aplikovány v určitém pořadí od hierarchicky nejvyšší organizační jednotky po tu nejnižší, shodné parametry s rozdílnými hodnotami jsou postupně přepsány pravidly hierarchicky nižší organizační jednotky.

3.4 Problémy při spolupráci protokolů

Při spolupráci protokolů přirozeně nastávají potíže tam, kde se rozcházejí se standardy a protokoly. Jako příklad můžeme uvést Kerberos autentizační mechanismus, který je v doméně Active Directory používán zároveň i k autorizaci. Klienti a služby spojené s Active Directory tudíž tuto autorizační informaci ve zprávě očekávají a její absence může vyvolat nestandardní chování, či selhání korektní funkce řady služeb.

Zároveň rovněž není standardizován protokol pro přístup a správu KDC (Key Distribution Center), takže je obtížné hromadně spravovat různé implementace KDC v jedné síti.

4 Jednotná správa

4.1 Problém nejednotné a decentralizované správy

Administrativním celkem se rozumí část sítě spravovaná jedním administrativním orgánem. Tento celek má za cíl poskytovat služby uživatelům a zařízením, které jsou do něj zařazeny. V případě rozsáhlejší sítě, která zahrnuje více administrativních celků, je tedy nutné duplikovat administrativní aparát na každý z těchto celků. Vznikají tedy další náklady nejen na obslužné systémy (síťová a výpočetní technika), ale především na lidské zdroje. Decentralizovaný způsob správy takové sítě v praxi znamená, že každý administrativní celek se co nejvíce přizpůsobuje svému prostředí na úkor jednoty výsledného řešení. Vzájemné propojení, tedy přístup z adminstrativního celku A ke zdrojům v administrativním celku B, je tím značně ztížený, často neefektivní z hlediska času a výkonu a někdy dokonce zcela nemožný.

4.2 Výhody jednotné správy

Sjednocení správy, tedy snížení počtu administrativních celků na jeden (který však může být dle potřeb hierarchicky rozdělený) má za cíl celkové zvýšení stability a kvality poskytovaných služeb pro jejich uživatele. V univerzitním prostředí jsou uživateli služeb převážně studenti, pedagogové a administrativní pracovníci.

Studenti by v případě sjednocení správy mohli pocítit hlavní změnu v počtu přihlašovacích údajů, které musí uchovávat. Pokud by mohli profesionální správci zajistit integritu a dodržování bezpečnostních standardů napříč celou sítí, mohli by studenti bez újmy na bezpečnosti používat jedno přihlašovací jméno a heslo do všech spravovaných informačních systémů (například studijní informační server, Komponenta Studium, školní emailový účet či datové úložiště přístupné přes protokoly FTP či WebDAV).

Pedagogičtí pracovníci by kromě jednotných přístupových údajů mohli využívat ještě společných nástrojů jak pro výuku, tak pro řízení projektů. Kromě toho by řadě z nich odpadla povinnost pečovat o počítačovou a síťovou strukturu v učebnách a mohli by plně zaměřit své síly tam kde jsou skutečně potřeba – na výuku či vědeckou, projektovou a výzkumnou činnost. Řada pedagogů, výzkumných pracovníků, i externistů je navíc součástí více kateder, či fakult a toto řešení by zvýšilo jejich mobilitu a efektivitu inter-katederní, resp. inter-fakultní spolupráce.

Administrativní síly jsou rovněž z velké části odkázány na funkční síťové služby a jednotné systémy mohou vést k vyšší efektivitě při předávání úkonů jak paralelně mezi pracovišti, tak hierarchicky nadřazeným či podřazeným. Systém jednotné správy usnadňuje přístup k datům napříč celým systémem.

4.3 Systém jednotné správy

Pro účely této práce budeme systémem jednotné správy chápat jednotnou sadu technologií, protokolů a doporučení, která má za cíl unifikovaným způsobem za pomoci automatizace

zajišťovat správnou funkci operačního systému a softwarové sady koncových stanic a serverů, případně dalších pomocných zařízení.

4.4 Volba systému (OS)

Způsob práce se systémem jednotné správy do velké míry určuje hlavní operační systém, který je pro správu zvolen, neboť každý operační systém do jisté míry určuje i sadu nástrojů a protokolů se kterými je nativně kompatibilní.

4.4.1 Kandidáti

Vhodné operační systémy budeme vybírat na základě:

- Předchozí zkušenosti
- Kompatibility s ostatními OS a službami
- Složitosti konfigurace
- Dlouhodobější perspektivy

Předchozí zkušenost hraje roli spíše jako ekonomický faktor, který určuje jak dlouho (a kolik prostředků) je zapotřebí, aby stávající administrátoři i uživatelé začali efektivně používat nový operační systém. V prostředí ČVUT koexistuje spousta operačních systémů, mezi hlavními však OS z rodiny Linux/Unix (včetně specializovaných odvětví pro Novell či terminálová zařízení) a Microsoft Windows®.

Co se týká **kompatibility** s ostatními operačními systémy, jedná se především o operační systémy podružných serverů (například webových, či aplikačních) a stanic koncových uživatelů. Když vezmeme v potaz dvě hlavní rodiny operačních systémů z předchozího bodu, můžeme narazit na některé nedostatky, a to hlavně s operačním systémem a produkty společnosti Microsoft, které nebývají open-source a tudíž v mnoha aspektech neexistuje nativní kompatibilita mezi produkty Microsoft a jinými. O přemostění tohoto problému se snaží hned několik, většinou open-source projektů, jako například Samba, které jsou však limitovány časem svých developerů a financováním projektu. Navíc je téměř pro každou sadu funkcí, které nativně fungují mezi Microsoft Windows klientským a serverovým operačním systémem, potřeba jiný nástro pro kompatibilizaci, což stěžuje jejich správu. Obecně lze říct, že kompatibilita je lépe zajišťována mezi systémy od jednoho prodejce, než mezi systémy různých prodejců/vydavatelů. Pokud je žádoucí, aby studenti, či pedagogové využívali produky jako například Microsoft Office (či Microsoft Office 365), emailový server Microsoft Exchange, či administrativní pomůcky typu Sharepoint (hromadná editace online tabulek či informací), je z hlediska úspory času při implementaci i zajištění (dopředné i zpětné) kompatibility a stability systému do budoucnosti výhodnější využít pro správu těchto produktů serverové operační systémy z rodiny Microsoft Windows.

Složitost konfigurace je pouze relativní pojem a dá se jen obtížně kvantifikovat. Nicméně je třeba podotknout, že ne všechny operační systémy jsou stejnou mírou připraveny na rychlou konfiguraci požadovaných rolí serveru dle standardního nastavení. Linuxové OS většinou nabízejí širší možnosti konfigurace, vykoupené časem, který je k tomu potřeba.

Oproti tomu Windows Server OS poskytují celou škálu základních rolí, které požadují jen nezbytné minimum informací od uživatele (administrátora) a z velké části se snaží automatizovat konfiguraci. Například instalace role doménového řadiče pro Active Directory automaticky přidá stroji i roli DNS serveru, KDC, TGS a AS bez zásahu uživatele, což může být v mnohých ohledech praktické, ale o to víc je zapotřebí mít detailní přehled o dopadech a možných následcích každé provedené akce.

Z hlediska **dlouhodobější perspektivy** je potřeba zvážit náklady potřebné na údržbu a aktualizaci daného operačního systému a zároveň dopřednou a zpětnou kompatibilitu mezi různými verzemi.

Po zhodnocení výše uvedených kritérií se budeme nadále v této práci zabývat pouze koexistencí OS z rodiny Linux a Microsoft Windows.

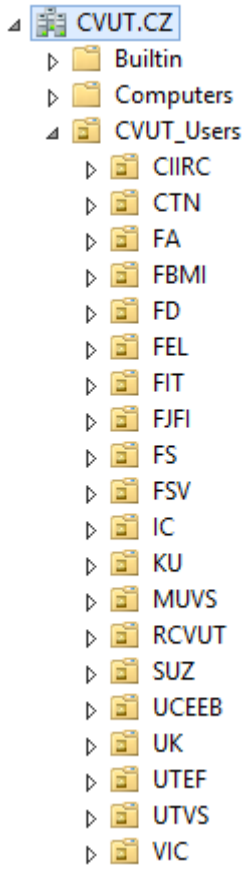
5 Míra unifikace

Správa velké organizace, jako například univerzity, vyžaduje pro vyšší efektivitu jistou míru jednotnosti. Na druhou stranu ale bohužel není vždy možné nastolit jednotné podmínky pro celou takovou organizaci, neboť by se mohly negativně odrazit na poskytovaných službách. Výjimka potvrzuje pravidlo a při plánování je třeba počítat i s neobvyklými případy. Nicméně obecně lze říci, že jednodušší návrhy často fungují nejlépe a před tím, než přejdeme k popisu správy heterogenního prostředí bychom měli zvážit zda neexistuje jednodušší řešení.

Pro adresářové služby je jedna ze zásadních otázek, zda může celá organizace spadat do jednoho forestu, což vlastně vede na otázku zda je možné udržet jednotné schéma pro celou organizaci. To je ovlivněno zejména aplikacemi a službami vyžadujícími zásah do schématu. Patří mezi ně zpravidla emailové služby (např. Microsoft Exchange mailserver). V takových případech se doporučuje vytvořit zvláštní doménu pouze pro tyto účely, která bude spravována odděleně od hlavní stromové struktury.

5.1 Jedna doména

Pokud bychom uvažovali nad variantou, ve které by byla celá organizace pouze jednou doménou, nabízí se okamžitě výhoda v podobě synchronizace a integrace. Interní mechanismy adresářových služeb v rámci jedné domény automaticky zajišťují, že tentýž objekt nebude veden duplicitně a všechny objekty mohou být vedeny centralizovaně, v jedné databázi. Síť a výpočetní prostředky by nebyly zbytečně vytěžovány mezi-doménovými nebo mezi-forestovými dotazy na objekty v jiných doménách či forestech. Rovněž by byla snažší správa systému DNS jako jednoho jmenného prostoru. V případě, že by kořenovou doménou bylo CVUT.CZ, mohly by jednotlivé fakulty být reprezentovány organizačními jednotkami, do kterých by byli rozděleni uživatelé dle jejich primárního zařazení:



Obr. 5.1.1 - Organizační jednotky jediné domény

Stejným způsobem by byly rozděleny i koncové stanice a servery jednotlivých fakult/kateder. Administrátoři domény CVUT.CZ by pak mohli delegovat určité úkony na osoby pověřené administrací fakult. Například administrátoři fakulty elektrotechnické by měli právo plné kontroly nad organizační jednotkou FEL, ale oprávnění pouze pro čtení na objekty v organizačních jednotkách jiných fakult. Adresářové struktury navíc standardně nabízejí možnosti granulárně nastavit přístupová práva (čtení/zápis) na každý jednotlivý atribut. Kombinací těchto dvou faktů lze dosáhnout situace, kdy by měli administrátoři fakult přístup pouze k některým atributům uživatelů omezených členstvím ve fakultní organizační jednotce. U uživatelů v ostatních organizačních jednotkách by mohli zobrazit pouze základní parametry a byla by tak zabezpečeno zachování důvěrných údajů (které by mohly být přístupné pouze servisním účtům dedikovaným pro konkrétní účely).

Stejným způsobem se dají v případě Active Directory delegovat i práva na vytváření a správu skupinových zásad. Zásady společné pro všechny uživatele i počítače by mohly být navázány přímo na kořenovou doménu a méně důležité, častěji měnící se zásady by mohly být spravovány přímo na úrovni organizačních jednotek jednotlivých fakult (případně kateder). Skupinové zásady jsou vázané na organizační jednotky, na jejichž objekty jsou aplikovány. Rozsah aplikace můžeme ještě zúžit výběrem pouze některých skupin, uživatelů, či počítačů v dané OU. Zásady aplikované na objekty v jedné organizační jednotce nemohou ovlivnit objekty v paralelních nebo vyšších organizačních jednotkách.

Například pokud aplikujeme určitou zásadu pro uživatele v OU FEL, nebude se tato zásada dotýkat uživatelů v OU jiných fakult, i když se přihlásí ke koncové stanici spadající pod FEL, protože uživatelské a počítačové zásady jsou striktně odděleny. Pokud ale aplikujeme zásadu na vrcholu struktury, například pro celou doménu CVUT.CZ, bude aplikována na uživatele ve všech fakultních organizačních jednotkách. Na úrovni fakultní OU však lze tuto zásadu modifikovat, neboť se vždy jako poslední aplikuje zásada, která je hierarchicky blíže koncovému objektu a přitom přepíše všechna předchozí nastavení.

Můžeme ještě zmínit službu DNS, která je ve Windows prostředí na doménovém řadiči integrována s Active Directory. Standardně jsou mezi DNS záznamy názvy a adresy hlavních služeb, které doména poskytuje ldap, kerberos, gc (Global Catalog),kpasswd (protokol pro změnu hesla), atd. Navíc jsou ale DNS jména udržovány i jako atributy jednotlivých objektů v adresáři Active Directory. Windows DNS server navíc může kromě standardního DNS souboru použít přímo Active Directory jako úložiště svých záznamů, které jsou pak dle pravidel replikace sdíleny mezi různými doménovými řadiči.

Přestože se logicky jedná o jedinou doménu, může samozřejmě fungovat na skupině fyzických či virtuálních doménových řadičů, pro dostatečnou distribuci pracovní zátěže a redundanci v případě katastrofy. Tato skupina doménových řadičů je navíc velice snadno škálovatelná oběma směry, přidání či odebrání doménového řadiče do existující domény je pro zkušeného administrátora otázka několika minut. Předmětem replikace jsou nejen objekty, atributy a schéma adresáře, ale u AD se navíc pomocí služby DFS (Distributed File System) replikují objekty skupinových zásad spolu s asociovanými soubory (například skripty, které mají proběhnout při přihlášení uživatele), které jsou uloženy standardně ve složkách SYSVOL a NETLOGON na doménových řadičích.

Otázkou jsou ovšem hodnoty atributů, které nejsou napříč organizací jednotné. Jako příklad můžeme uvést emailovou adresu. Předpokládá se, že student „novak“ na fakultě elektrotechnické bude mít emailovou adresu novak@fel.cvut.cz, pokud bude souběžně studovat i na fakultě stavební, potom tam bude mít adresu novak@fsv.cvut.cz. Pokud se do atributu email vepíše pouze jedna hodnota, bude s největší pravděpodobností student dostávat pouze polovinu emailů. Pokud se vepíšou obě hodnoty, může ještě nastat problém při jejich zpracování poštovní aplikací, kde záleží zda jsou hodnoty odděleny čárkou nebo středníkem. Jako řešení tohoto konkrétního problému se nabízí buď jednotný poštovní systém, nebo rozšíření schématu o atributy pro jednotlivé fakulty, například místo jednoho atributu „mail“ by měl uživatelský objekt atributy „mail-fel“ s hodnotou novak@fel.cvut.cz a „mail-fsv“ s hodnotou novak@fsv.cvut.cz. Zde ale můžeme opět narazit na problémy s kompatibilitou u poštovních aplikací a každé rozšíření schématu je nevratná operace, která jistou mírou stěžuje vyhledávání a práci s adresářem, tudíž je vhodné schéma rozšiřovat co nejméně.

Pokud z těchto důvodů není možné přistoupit na variantu jediné domény, jako další krok se nabízí jednotný forest.

5.2 Jeden forest

Jediný forest je strukturně složitější koncept nežli jediná doména, přestože funkčnost je podobná. Jedná se o striktnější oddělení jednotlivých organizačních celků (například fakult) nejprve do domén, které se pak dále dělí na organizační jednotky jako v bodě 5.1. Pokud by domény jednotlivých fakult byly dceřiné domény k doméně CVUT.CZ (FEL.CVUT.CZ, FIT.CVUT.CZ, atd) a rozdělovaly jmenný prostor CVUT.CZ na menší celky, tak by tím zároveň rozdělily i správu jednotlivých jmenných prostorů. Navíc by i ty nejmenší domény musely udržovat minimálně jeden, ale spíše dva a více doménových řadičů.

Oproti jediné doméně však tato konfigurace nabízí zásadní rozdíl, kterým je možnost existence uživatelů se stejným přihlašovacím jménem v různých doménách. Pokud jsou fakultní domény dceřinými doménami ke kořenové, funguje mezi nimi implicitní obousměrná vazba, která umožňuje uživatelům jakékoli domény přihlášení v jakékoli jiné doméně. Při přihlašování bývá přihlašovací realm nastaven na stanici implicitně a uživatel tedy většinou zadává pouze své přihlašovací jméno. Pokud se uživatel „novak“ bude přihlašovat ke koncové stanici patřící do domény FEL.CVUT.CZ, zadá pouze „novak“ a UPN je implicitně doplněno na novak@FEL.CVUT.CZ, analogicky pak na jiných fakultách. Ze systémového hlediska jsou FEL\novak a FIT\novak naprosto rozdílní uživatelé. Z uživatelského hlediska je však přívětivé používat pouze jedny přihlašovací údaje napříč celou organizací. Aby se tak však mohlo dít, je potřeba přičinění centrální správy, která bude mít za cíl sjednotit společné prvky nastavení jednotlivých domén. V této konfiguraci již je například možné, aby uživatel „novak“ používal automaticky emailovou adresu fakulty, kde je přihlášen ke koncové stanici. Pokud by ale chtěl používat tzv. domovský adresář, museli by se administrátoři jednotlivých domén domluvit na formátu sdíleného úložiště, propagovat tyto parametry správně ve všech dotčených doménách a správně nastavit přístupová práva na dané úložiště: aby si mohl uživatel „novak“ prohlédnout své dokumenty jak z fakulty elektrotechnické, tak z fakulty informačních technologií, musely by ke složce mít přístup účty FEL\novak i FIT\novak.

Další překážkou vycházející z faktu, že v každé doméně je uživatel veden zvlášť, je správa hesel. Přestože přihlašovací jméno může být stejné, hesla jsou v obou doménách vedena zvlášť a pokud vyprší, musí je uživatel znovu nastavovat ve všech doménách. Tento aspekt lze vyřešit správou hesel mimo domény, nebo použitím reverzibilního šifrování, které následně umožňuje replikovat hesla mezi doménami, děje se tak však na úkor bezpečnosti (šifrování musí být reverzibilní, jelikož při šifrování je spolu s heslem šifrován i název domény: uživatel se stejným přihlašovacím jménem i heslem bude mít v různých doménách různá šifrová slova).

Zřejmě největším úskalím takového řešení je zakládání, synchronizace a mazání uživatelů a skupin. Pokud uživatel „novak“ ukončí studium, nebo změní sadu zapsaných předmětů, je zapotřebí propagovat tuto změnu do všech domén a to buď paralelně (například z nástroje pro správu uživatelů), nebo postupně nejdříve v kořenové doméně a pak níže, aby byly informace konzistentní.

Z tohoto pohledu lze ve všech doménách použít účet z kořenové domény: CVUT\novak kvůli konzistenci a správě hesel, ale to vede na stejné výhody i nevýhody, které již byly popsány v kap. 5.1.

Vzhledem k tomu, že schéma v celém forestu musí být jednotné, ani tato konfigurace neumožňuje různá schémata pro různé fakulty. Výhodou však může být oddělená správa Active Directory, DNS a Group Policy pro danou doménu: Miskonfigurace, nebo havárie jedné celé domény nebude mít zásadní vliv na zbytek domén ve forestu (pokud se nejedná o kořenovou doménu).

5.3 Více forestů

Pokud nelze implementovat předchozí dvě varianty, tato varianta nabízí největší možnosti přizpůsobení na úkor jednoty a centralizace.

Každý forest může existovat bez návaznosti na jiné foresty, pokud však mezi nimi vytvoříme vazby, mohou vzniknout zajímavé kombinace. Uveďme jako příklad forest CVUT.CZ a forest FEL.CZ. V praxi je možné vytvořit foresty CVUT.CZ a FEL.CVUT.CZ, aniž by byl FEL.CVUT.CZ dceřinou doménou CVUT.CZ a následně je provázat jednosměrnou či obousměrnou vazbou, nicméně pro názornost použijeme jména z rozdílných jmenných prostorů.

Hlavní výhodou několika forestů je možnost individuálního přizpůsobení schématu v každém forestu. Jelikož je každý forest samostatná jednotka, musí jeho správce zabezpečit chod DNS, adresářových služeb i dalších návazností.

Předpokládejme ve struktuře AD jednosměrnou vazbu FEL.CZ -> CVUT.CZ, která zajišťuje, že uživatelé CVUT.CZ mohou přistupovat ke zdrojům z FEL.CZ. Pokud se uživatel CVUT.CZ přihlásí na stanici patřící do FEL.CZ, budou se na něj aplikovat uživatelské zásady z jeho mateřské domény (CVUT.CZ) a na stanici naopak počítačové zásady z její mateřské domény, tedy FEL.CZ. (Microsoft, 2005)

Navíc lze uživatele z domény CVUT.CZ přiřadit do skupin vytvořených ve FEL.CZ. Tyto skupiny musí mít platnost pouze v lokální doméně (v kontrastu s globálními či univerzálními skupinami). A standardně povolit či omezit práva na základě členství ve skupině již není těžký úkol. Lze tímto způsobem přiřadit práva stejně, jako by tito uživatelé existovali v lokální doméně. Při zařazení do takové skupiny je v doméně FEL.CZ automaticky vytvořen v organizační jednotce „ForeignSecurityPrincipals“ zástupný objekt, který nese jméno shodné se SecurityID (SID) atributem původního objektu. SecurityID původního a zástupného objektu jsou rovněž shodné. Zástupný objekt je tímto způsobem vytvořen nejen pro uživatele, ale i pro skupiny.

Tyto zástupné objekty ovšem dodržují schéma domény (forestu), ve které se nacházejí. Představují tedy určitý způsob jak využít dvě rozdílná schémata. Oficiálně jsou však tyto objekty pouze součástí systému a nemělo by s nimi být manipulováno. Některé specifické aplikace, nebo skripty by však této možnosti mohly využít pro usnadnění správy, případně

přidání nezbytných atributů bez zbytečného rozšiřování schématu domény, ve které jsou uživatelské objekty standardně uloženy. Na následujícím obrázku je pouze reprezentace zástupného objektu v nástroji pro správu Active Directory domény:

Obr. 5.3.1 - Foreign Security Principal

Zdroj: Vlastní zpracování

Pro tento typ spolupráce mezi foresty je však nezbytná správná organizace DNS. Obě strany by měly zaregistrovat do svých záznamů DNS zónu druhého forestu a delegovat všechny požadavky směřující do této zóny na DNS server druhého forestu. Službu DNS lze v případě potřeby částečně delegovat i na jiný server (než doménové řadiče obou forestů).

Delegovat lze i jiné role, pokud je to zapotřebí. Můžeme delegovat například i roli autentizace na jiný server podporující Kerberos autentizaci.

Můžeme uvažovat heterogenní prostředí, kde CVUT.CZ je Linuxový server na kterém běží autentizační služba Kerberos a adresářové služby v implementaci zvané OpenLDAP a FEL.CVUT.CZ je samostatný Active Directory forest. Můžeme vytvořit odchozí vazbu z FEL.CVUT.CZ na CVUT.CZ, aby uživatelé z domény CVUT.CZ mohli přistupovat ke zdrojům v doméně FEL.CVUT.CZ. Tento typ vazby se nazývá „cross-realm trust“ a v zásadě umožňuje centrální správu uživatelských hesel. Tímto způsobem může být centrální server propojen s forestem každé fakulty a uživatel může napříč celou organizací používat (a měnit) pouze jedno heslo.

Při tomto typu řešení je ale nutné, aby všechny uživatelské objekty byly v AD předem vytvořeny a aby měly namapovány UPN, který je doménovým řadičem přeposlán na centrální KDC server. Přestože tato kombinace dokáže vyřešit problém nejednotného hesla, nijak neřeší problém synchronizace uživatelů a skupin v lokálních doménách. Toto řešení lze implementovat poměrně snadno, je škálovatelné a poskytuje velkou flexibilitu (schéma, skupinové zásady, kompletní kontrola nad forestem), avšak na druhou stranu vede k decentralizaci, heterogenitě a nižší efektivitě správy, pokud je implementována na příliš nízké úrovni (například katedra). Další nevýhodou jsou pak drobné rozdíly v podobě a použití Kerberos ticketů vydávaných Kerberem v kontextu AD a v jiné implementaci. Základní služby, jako přihlášení do stanice, tisk a přístup k souborům lze zajistit tímto způsobem, ale některé specifické (Microsoft) aplikace nemusejí nyní, či v budoucnu toto řešení podporovat.

Tento typ řešení je více do detailu popsán v sekci realizace.

6 Míra automatizace

Míru automatizace můžeme zařadit mezi kritéria pro volbu správného systému pro správu. Dostatečná automatizace je důležitá pro lepší využití času administrátorů, ale přílišná automatizace může způsobit nižší míru pružnosti reakce na změny či výjimky a dlouhou reakční dobu v takových situacích.

V bodech, kde byly hodnoceny klady a zápory jednotlivých typů řešení centralizované správy se často objevuje jako negativní stránka nutnost synchronizace skupin, uživatelů a jejich členství ve skupinách. Tudíž automatizace právě tohoto úkonu může odstranit tyto negativa a vyzdvihnout jiné přednosti daných metod, které by byly jinak jen obtížně použitelné.

Synchronizace v heterogenním prostředí bývá ještě ztížena pokud jsou na obou stranách rozdílné technologie, schémata a atributy jako například při synchronizaci mezi adresářovými službami typu OpenLDAP se schématem univerzity/fakulty a službami Active Directory se schématem a specifickými potřebami jedné katedry. Právě pro tento účel jsem v rámci realizace vazby mezi OpenLDAP serverem s Linuxovým Kerberos protokolem a Active Directory forestem vyvinul skript pro synchronizaci uživatelů, skupin a členství uživatelů ve skupinách.

7 Realizace

7.1 Požadavky

Základním požadavkem je realizace systému jednotné správy a autentizace počítačů/uživatelů na katedře telekomunikační techniky (FEL). Dodatečně byla ještě specifikována priorita praktické realizace tak, aby systém mohl sloužit prioritně pedagogům a administrativním pracovníkům, předtím než se rozšíří i na synchronizaci studentských účtů. Nicméně veškeré kroky byly koncipovány pro základní požadavek synchronizace všech účtů a proto stačí provést pro splnění tohoto účelu drobné úpravy, které uvedu spolu s řešením.

Zároveň s autentizačními metodami a synchronizací je pro úplnost uveden i postup konfigurace Active Directory, který může sloužit jako základní sada doporučení pro pracoviště, která chtějí adopotovat podobné řešení.

Mezi požadavky na správu figuruje:

- Nahradit stanice s operačním systémem Windows XP novějšími
- Sjednotit laboratoře katedry vůči studentům: jednotné přihlašovací jméno i heslo napříč laboratořemi, výhledově univerzitní přihlašovací jméno a heslo
- Výhledově mapovat studentům domovské adresáře
- Možnost češtiny i angličtiny při přihlášení do systému
- Maximální homogenita softwarové sady koncových stanic s cílem usnadnit náhlé přesuny výuky
- Maximální automatizace správy výpočetní techniky (přiřazení do domény, obnovení výchozího stavu počítačů po zapnutí, dálková správa aktualizací, instalace softwaru, možnost online konfigurace firewallu při testech v učebnách, menší aktualizace během výukové části semestru, větší aktualizace v době prázdnin/zkouškového období)

7.2 Topologie – heterogenní síť

Součástí topologie je Kerberos server (kerberos.cvut.cz), OpenLDAP server pod správou fakulty elektrotechnické (ldap.feld.cvut.cz), který již obsahuje veškeré uživatele i skupiny ze, které byly vyhodnoceny jako relevantní pro tuto fakultu. Způsob aktualizace a správy dat v tomto adresáři je mimo rámec této realizace. Dále je součástí realizace dvojice doménových řadičů Microsoft Windows 2012 R2 a koncové stanice v několika učebnách a administrativní části. Pro ilustraci bude realizace popsána pouze na sadě 20 koncových stanic jedné učebny.

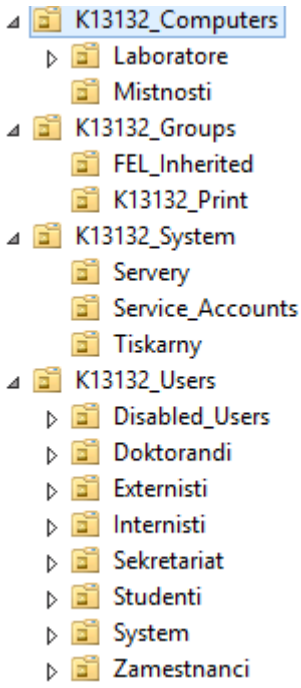
7.3 Konfigurace serveru – doménový řadič

Jméno primárního doménového řadiče bylo zvoleno na KTTDC1, sekundární pak analogicky KTTDC2. Jako jméno forestu (a zároveň jediné domény) bylo zvoleno K13132.FEL.CVUT.CZ, přestože nebylo zamýšleno propojení s nadřazenou Active Directory doménou či forestem.

7.3.1 Active Directory

Jako první byla přidána a nakonfigurována role Active Directory Domain Services spolu s DNS jako prerekvizitou. Při konfiguraci je třeba zvolit master domain reset password, které slouží v extrémních případech k obnovení přístupu k doméně.

Dále byla zvolena struktura organizačních jednotek v nové doméně, částečně podle předchozí organizační struktury:



Obr. 7.3.1 - Organizační struktura domény

Zdroj: Vlastní zpracování

Následně byly manuálně vytvořeni první uživatelé a skupiny určené primárně pro administraci domény a serveru.

7.3.2 NTP

Následovalo nastavení správné časové zóny a synchronizace s externím zdrojem času pomocí protokolu NTP. Jako zdroje byly použity následující NTP servery: tik.cesnet.cz / 195.113.144.201, tak.cesnet.cz / 195.113.144.238 pomocí příkazů:

w32tm /config /manualpeerlist:195.113.144.201,195.113.144.238 /syncfromflags:MANUAL

w32tm /config /update

7.3.3 WDS

Jako další byla nastavena role „Windows deployment services“ (WDS) pouze pro testovací účely. V praxi není vhodné umisťovat tuto službu na doménový řadič. Služba WDS slouží

pro automatizovanou síťovou instalaci operačních systémů pomocí předem vytvořeného obrazu disku s potřebnou softwarovou výbavou a nastaveními.

7.3.4 Radius

Pro účely ověřování VPN brány a jiných zařízení používající protokol RADIUS byla nastavena služba „Network Access Policy“ (NAP) a přidáni potřební RADIUS klienti.

7.3.5 Group policies/skupinové zásady

Základní skupinové zásady byly definovány na základě doporučení článku Oxford IT Services (IT Services, 2015) a týkají se modifikace šifrovacích sad koncových stanic i doménových řadičů pro kompatibilitu se šifrovacími sadami využívanými Kerberos serverem. Dále přidána zásada pro upřesnění kdc serveru pro koncové stanice i doménový řadič a další zásady pro vyšší komfort uživatelů (jazyková nastavení, specifikace Key Management Serveru pro aktivaci operačního systému i sady Microsoft Office, konfigurace firewallu na stanicích, úprava frekvence automatických aktualizací, mapování tiskáren pro administrativní část, publikování některých ikon a nastavení registrů pro studentské účty, nastavení sady Microsoft Office pro přeskakování dialogu při prvním spuštění, definování oprávnění pro studentské účty).

7.3.6 Vazba

Vytvoření jednosměrné odchozí vazby s realmem CVUT.CZ. Ruční namapování UPN několika uživatelům.

7.3.7 Tisk

Pro účely administračních oddělení byla nainstalována i tisková role, opět pouze pro testovací účely.

7.3.8 Správa antivirového programu

Byla nainstalována rovněž role pro hromadnou správu softwarových klientů antivirového programu, nainstalovaných na koncových stanicích.

7.4 Synchronizace s LDAP serverem

Pro účely synchronizace s LDAP serverem byl vytvořen skript v jazyce Windows Powershell, který je určen k běhu na jednom z doménových řadičů a jehož funkci si zde ukážeme:

Nejdříve v hlavičce skriptu definujeme proměnné, aby nebylo třeba zasahovat do těla skriptu při změně prostředí. Skript je psán i komentován v angličtině kvůli pozdější čitelnosti širším u širšího spektra publika, důraz je kladen na přehlednost a transparentnost:

```
################################
#       Variables              #
################################
#File name of csv export of foreign domain
$foreignCSV = "Foreign.csv"
$foreignGroupsCSV = "ForeignGroups.csv"
#File name of csv export of local domain
$localCSV = "Local.csv"
```

Můžeme si povšimnout, že jsou deklarována jména .csv souborů. Skript totiž nejprve v první fázi sbírá data z vlastního i cizího doménového řadiče do formátu CSV, místo standardního formátu LDIF (LDAP Data Interchange Format) a to jednak kvůli snažšímu mapování atributů, tak možnosti editace dat v tabulkovém editoru před importem. Tímto způsobem lze použít skript i pro vytváření uživatelských účtů zadaných v souboru CSV.

Dále definujeme parametry pro připojení k fakultnímu LDAP serveru:

```
$LDAPServer = "ldap.feld.cvut.cz"
#Base OU of remote LDAP server
$RemBaseOU = "ou=People,o=feld.cvut.cz"
```

Zde definujeme jméno lokální domény pro import a organizační jednotku pro import uživatelů a skupin: Byla zvolena dedikovaná organizační jednotka, která bude synchronizována. Účty manuálně vytvořené mimo tuto organizační jednotku (administrátorské, systémové, servisní) budou ignorovány. Specifikujeme rovněž LDAP filtr pro obdržení pouze objektů relevantních pro katedru 13132 (členové skupiny 13132-all). Zároveň specifikujeme servisní účty cizí domény, které nám umožní přečtení všech potřebných atributů, zde nahrazeno písmeny XXX.

```
$OurDomain = "K13132.FEL.CVUT.CZ"
$OurUserOU =
"OU=Internisti,OU=K13132_Users,DC=K13132,DC=FEL,DC=CVUT,DC=CZ"
$OurUserFilter = "(&(objectCategory=person)(objectClass=user))"
$OurGroupOU =
"OU=FEL_Inherited,OU=K13132_Groups,DC=K13132,DC=FEL,DC=CVUT,DC=CZ"
$RemUserFilter = "(memberOf=""cn=13132-
all,ou=13132,ou=Departments,ou=Groups,o=feld.cvut.cz"")"
$RemUserAuth = '"XXXXXXXX,ou=Special Users,o=feld.cvut.cz" "XXXXX"'
$RemGroupOU = "ou=13132,ou=Departments,ou=Groups,o=feld.cvut.cz"
$RemGroupFilter = "(objectClass=groupofuniquenames)"
$RemGroupAuth = '"XXXXXXXX,ou=Special Users,o=feld.cvut.cz" "XXXXXX"'
```

Níže je definována cesta a jméno logu, kam se zapisují všechny informace o průběhu skriptu.

```
################################
#       Introduction           #
################################
cd $env:USERPROFILE\Desktop
#Creating log file
$LogName = "LDAPSyncLog.log"
$LogPath = "$env:USERPROFILE\Desktop"
$LogFilePath = $LogPath + "\" + $LogName
 #Check if file exists and delete if it does
If((Test-Path -Path $LogFilePath)){
Remove-Item -Path $LogFilePath -Force
}
```

Funkce LogWrite, spolu s její inicializací a způsobem použití byla převzata z (Sturlese, 2013) a následně upravena pro použití v tomto skriptu.

```
#Create file and start logging
New-Item -Path $LogFilePath -ItemType File | Out-Null #We are just
supressing the annoying output about log file being created

Function LogWrite #Define function which we will be calling throughout
the script to write to log
{
   Param ([string]$logstring)
Process{
   Add-content $LogFilePath -value "$([DateTime]::Now): $logstring"
   }
}
```

Uživatelé a skupiny jsou exportováni do CSV souboru pomocí nástroje CSVDE:

```
################################
#       Data import and USERS  #
################################

LogWrite "Starting the script"
#First we need to make sure that our data is up to date -> exporting
reference and local ad users to csv files
LogWrite "Importing data"
LogWrite "Importing foreign LDAP users"
#We're using csvde to get a csv output of foreign domain principals,
filtered by the parameter "-r" and authenticated by username and
password provided in "-a"
csvde -u -f $foreignCSV -s $LDAPServer -d $RemBaseOU -r $RemUserFilter
-a $RemUserAuth
LogWrite "Importing local LDAP users"
#The same for local users, we don't need to authenticate, the script
will be executed in the same context as user (assuming domain admin)
csvde -u -f $localCSV -d $OurUserOU -r $OurUserFilter
LogWrite "Importing foreign LDAP groups"
csvde -u -f $foreignGroupsCSV -s $LDAPServer -d $RemGroupOU -r
$RemGroupFilter -a $RemGroupAuth
```

Do funkce compare-object vložíme jako parametry importy CSV souborů a necháme je porovnat. Funkce vrátí seznam položek, které nebyly nalezeny v druhém souboru, nebo se

liší ve specifikovaném parametru „uid“ (později uvidíme, že v rámci zachování kompatibility jsou v AD vyplňovány i atributy jako „uid“). U každé položky pak figuruje atribut „Side Indicator“, který ukazuje jakým směrem by se měla položka přesunout.

```
#Now let's compare the objects' uids, we're using the passThru
parameter to keep whole differing user objects in case we need to
create the users later
$diffUsers = compare-object -referenceobject $(import-csv $foreignCSV)
-differenceobject $(import-csv $localCSV) -Property uid -passThru #it
seems logically easier to take our local domain as reference to
determine which users go "in" (<=) and "out" (=>)
```

Na základě seznamu rozdílů a SideIndicatoru určíme seznam uživatelů k vytvoření a vypíšeme jej na obrazovku i do logu.

```
#We have our difference Users, let's determine which ones should be
created and create them:
$toBeCreated = $diffUsers | Where-Object {$_.SideIndicator -eq "<=" }
LogWrite "Creating users: $($toBeCreated.uid)"
Write-Host "Creating users"
$toBeCreated.uid
```

V následující části skriptu procházíme seznam uživatelů k vytvoření a jednoho po druhém vytváříme. Bohužel někteří uživatelé nemají některé požadované atributy vyplněné (jako například číslo místnosti), proto přidáváme podmínku ošetřující tuto vlastnost. Nově vytvořeným uživatelům rovněž přiřadíme heslo. To by nemělo být důležité, jelikož v praxi bude použito univerzitní heslo.

```
#Display name is OK, we create users based on the existence of their
"room number" attribute
foreach($Person in $toBeCreated){
$DisplayName=$Person."givenName;lang-cs"+" "+$Person."sn;lang-cs"
if($Person.roomNumber -ne ""){
New-ADUser -SamAccountName $Person.uid -DisplayName $DisplayName -Name
$DisplayName -GivenName $Person."givenName;lang-cs" -Surname
$Person."sn;lang-cs" -EmailAddress $Person.mail -Description
$Person."employeeType;lang-en"-EmployeeNumber $Person.employeeNumber -
Path $OurUserOU -AccountPassword (convertto-securestring -asplaintext
"XXXXXXXX" -Force) -Enabled $true -OtherAttributes
@{'departmentNumber'=$Person.departmentNumber;'employeeType'=$Person."
employeeType;lang-
en";'uid'=$Person.uid;'userPrincipalName'=$Person.uid+'@'+$OurDomain;'
physicalDeliveryOfficeName'=$Person.roomNumber}
}
else{
New-ADUser -SamAccountName $Person.uid -DisplayName $DisplayName -Name
$DisplayName -GivenName $Person."givenName;lang-cs" -Surname
$Person."sn;lang-cs" -EmailAddress $Person.mail -Description
$Person."employeeType;lang-en"-EmployeeNumber $Person.employeeNumber -
Path $OurUserOU -AccountPassword (convertto-securestring -asplaintext
"XXXXXXX" -Force) -Enabled $true -OtherAttributes
@{'departmentNumber'=$Person.departmentNumber;'employeeType'=$Person."
employeeType;lang-
en";'uid'=$Person.uid;'userPrincipalName'=$Person.uid+'@'+$OurDomain;}
}
}
```

Zde naopak budeme odmazávat uživatele, kteří ve fakultním adresáři nebyli nalezeni.

```
#Now let's sort out the extra ones and delete them:
$toBeDeleted = $diffUsers | Where-Object {$_.SideIndicator -eq "=>" }
#We should prompt for some confirmation before the actual deletion
Write-Host "Are you sure you want to delete following users?"
$toBeDeleted.sAMAccountName #Not using uid, because manually created
accounts might not have it set, but every AD account has to have
sAMAccountName
LogWrite "Following users are marked for deletion:
$($toBeDeleted.sAMAccountName)"
foreach($Person in $toBeDeleted){
Remove-ADUser $Person.sAMAccountName
}
```

Nyní přichází import a synchronizace skupin. Nejprve zkusíme vytvořit každou skupinu ze seznamu. Nové skupiny se vytvoří a pro existující skupiny nastane chyba, která je odchycena v bloku „catch“.

```
################################
#       GROUPS                 #
################################

$GroupList=import-csv $foreignGroupsCSV

foreach($Group in $GroupList){

try{ #Let's try to directly create the group, if it already exists, we
will just move on
New-Adgroup -Name $Group.cn -GroupScope Global -Path $OurGroupOU
Write-Host "Creating the group: " -nonewline
Write-Host $($Group.cn) -foregroundcolor yellow
LogWrite "Creating the group: $($Group.cn)"
}
```

Transformací textového řetězce z výstupního formátu na pouhé atributy UID oddělené čárkami, který připravíme pro porovnání se stávající skupinou v doméně.

```
catch{
 LogWrite "The group: $($Group.cn) already exists, moving on"
}
#Now we will get the foreign domain group members and modify the
output to an array of given format = probably the most heavy data
transformation, leaves room for optimisation
$members = $Group.uniqueMember -replace
("^uid=|,ou=People,o=feld\.cvut\.cz;?","") -replace ("uid=",",") -
split (",") #erase starting "uid=" and all trailing
",ou=People,o=feld.cvut.cz;" (with or without semi-colon at the end)
and replace the rest of "uid=" by commas

#Now we get the corresponding group in our local domain. It should
always exist, because it is created in the try-catch step above.
$ADmembers = Get-ADGroupMember $Group.cn | select sAMAccountName #This
command outputs plenty of members' properties. We are only interested
in the sAMAccountName/uid
```

Pokud ekvivalentní skupina v AD není prázdná, srovnáme její členy se členy stejnojmenné skupiny ve fakultním adresáři a opět získáváme seznam uživatelů, kteří by měli být do skupiny zařazeni a seznam uživatelů, kteří by naopak měli být ze skupiny odstraněni:

```
if($ADmembers) #comparison is worth while only if our group is NOT
empty
    {
        $diffMembers = compare-object -referenceobject $members -
differenceobject $ADmembers.sAMAccountName -passThru #Comparing
members of foreign and local group
        #Getting the lists of users
        $toBeCreated = $diffMembers | Where-Object {$_.SideIndicator -
eq "<=" } #the $_ represents currently handled object
        $toBeDeleted = $diffMembers | Where-Object {$_.SideIndicator -
eq "=>" }
```

Po zapsání hlavních informací do logu a na obrazovku zkusíme funkci „Add-ADGroupMember“ předat jedním parametrem celé pole uživatelů k vytvoření, neboť je to

rychlejší než procházet polem uživatelů a pro každého znovu volat funkci „Add-ADGroupMember“.

```
        #Taking the appropriate action:

        if($toBeCreated){ #proceed with adding members only if there
is something to add ($toBeCreated NOT null)
        Write-Host "Adding following users to the group " -nonewline
#Of course we keep the user informed and nicely output the values in
yellow
        Write-Host $($Group.cn) -foregroundcolor yellow
        $toBeCreated
        LogWrite "Adding following users to the group $($Group.cn):
$toBeCreated" #Everything that happens needs to be logged
        # Add-ADGroupMember -Identity $Group.cn -Members $toBeCreated
# this would be just a simple command without error-detection,
prevention or logging
        try{
        Add-ADGroupMember -Identity $Group.cn -Members $toBeCreated
```

Nejčastěji se setkáváme s chybou, že se snažíme do skupiny přidat neexistujícího uživatele (nejčastěji studenta), nicméně chybu zalogujeme a vypíšeme na obrazovku:

```
catch{
        $ErrorMessage = $_.Exception.Message
        Write-Host "Following exception occurred: "
        Write-Host $ErrorMessage -foregroundcolor red
        Write-Host "We will now try to add users to the group one by
one. You will find the list of errors in the log."
        LogWrite "Following exception occurred: $ErrorMessage"
        LogWrite "We will now try to add users to the group one by
one."
```

Nyní již procházíme seznam uživatelů k přidání do skupiny a v případě, že přidání některého z nich selže, přímo vypíšeme konkrétního uživatele, který tuto chybu působí:

```
foreach($member in $toBeCreated){
                    try{
                    Add-ADGroupMember -Identity $Group.cn -Members
$member
                    }
                    catch{
                            LogWrite "$member is causing sync error
(user does not exist in local domain)"
                            }
                }

        }
        }
```

Zde ještě kontrolujeme seznam uživatelů pro vyjmutí ze skupin, jelikož v doméně již předtím byli, není zcela nezbytné je odebírat po jednom a můžeme najednou:

```
        if($toBeDeleted){ #proceed with deleting members only if there
is something to delete ($toBeDeleted NOT null)
        Write-Host "Deleting following users from the group
$($Group.cn)" -foregroundcolor red
        $toBeDeleted
        LogWrite "Deleting following users from the group
$($Group.cn): $toBeDeleted"
        Remove-ADGroupMember -Identity $Group.cn -Members $toBeDeleted
        }
    }
```

Ještě kontrolujeme, zda seznam členů není prázdný a opět zkusíme hromadně přidat členy skupiny najednou. Poslední množinová závorka ukazuje konec kódu, který se provádí pro každou skupinu v seznamu skupin.

```
elseif($members -ne '') #if our AD group is empty we will just add new
members from foreign domain, if there are any
    {
        Write-Host "Adding following users to the group " -nonewline
        Write-Host $($Group.cn) -foregroundcolor yellow
        $members
        LogWrite "Adding following users to the group $($Group.cn):
$members"
        try{
        Add-ADGroupMember -Identity $Group.cn -Members $members
        }
        catch{
        $ErrorMessage = $_.Exception.Message
        Write-Host "Following exception occurred: "
        Write-Host $ErrorMessage -foregroundcolor red
        Write-Host "We will now try to add users to the group one by
one. You will find the list of errors in the log."
        LogWrite "Following exception occurred: $ErrorMessage"
        LogWrite "We will now try to add users to the group one by
one."
        foreach($member in $members){
                    try{
                    Add-ADGroupMember -Identity $Group.cn -Members
$member
                    }
                    catch{
                            LogWrite "$member is causing sync error
(user does not exist in local domain)"
                            }
                }

        }
    }

}
LogWrite "Synchronisation finished"
```

Jednostranná synchronizace tedy skončila a stav uživatelů a skupin by měl být totožný se stavem ve fakultním LDAPu.

8 Možnosti vylepšení

Uvedený skript neprovádí kontrolu jednotlivých atributů uživatelů, nicméně pro tuto kontrolu lze použít stejný algoritmus jako pro kontrolu seznamu uživatelů skupiny, která je již ve skriptu implementována.

Vzhledem k vyššímu počtu uživatelů a atributů by však bylo dobré nejprve porovnat datum poslední změny uživatelů ve fakultním adresáři a vybrat k synchronizaci pouze ty, jenž se změnili v době od poslední synchronizace.

Dále je tento skript uzpůsoben pro běh v konzolovém prostředí, ale grafické rozhraní by zajisté bylo uživatelsky přívětivější.

Na závěr ještě chybí zásahy pro úplnou automatizaci běhu skriptu. Stávající verze je vhodná pro začátek, ladění chyb a běh pod dozorem administrátora. Hlavním cílem skriptu by však mělo být běžet v pozadí v synchronizačních intervalech (například několikrát denně) a v případě potřeby manuálního zásahu například odeslat email administrátorovi, který by zásah provedl.

9 Závěr

V práci je zrekapitulován způsob funkce autentizačních protokolů a jejich spolupráce, zejména pak provázanost těchto protokolů a služeb v implementaci Active Directory Domain Services.

Zároveň jsou popsány způsoby využití Active Directory ať už v homogenní síti, nebo v kombinaci se službami jiných dodavatelů. V kapitole č. 5, u přehledu možností využití a implementace adresářových služeb pro správu počítačové sítě a uživatelských účtů, jsou uvedena rovněž doporučení v jakých případech použít které možnosti a podle jakých kritérií se lze rozhodovat. Z těchto doporučení je zřejmě nejzásadnější „postupovat při návrhu vylučovací metodou od nejjednodušších návrhů až po ty nejsložitější“, které je však špatně aplikovatelné v již fungující struktuře.

Na praktické realizaci byly demonstrovány základní principy, výhody, i úskalí jedné z možností implementace s následným návrhem řešení vážnějších potíží.

Pro katedru telekomunikační techniky byl realizován systém správy koncových stanic, stanice s operačním systémem Windows XP byly nahrazeny novějšími při zavádění automatizovaných metod instalace operačního systému. Celkově bylo vyhověno všem požadavkům katedry telekomunikační techniky, kromě těch, které mají plánované datum dokončení po ukončení tohoto semestru.

10 Zdroje

2015. Active Directory Users, Computers, and Groups. *Microsoft Technet.* [Online] Microsoft, 2015. [Citace: 22. March 2015.] https://technet.microsoft.com/en-us/library/bb727067.aspx.

Bouška, Petr. 2014. Kerberos část 1 – Microsoft Active Directory. *Živě.cz.* [Online] Mladá fronta, a.s., 6. June 2014. [Citace: 14. February 2015.] http://www.zive.cz/clanky/kerberos-cast-1--microsoft-active-directory/sc-3-a-173970/default.aspx. 12128554.

—. 2014. Kerberos část 3 – Princip Kerberos Autentizace. *Živě.cz.* [Online] Mladá Fronta, a.s., 11. November 2014. [Citace: 16. February 2015.] http://www.zive.cz/clanky/kerberos-cast-3--princip-kerberos-autentizace/sc-3-a-175765/default.aspx. 12128554.

IBM Redbooks. 2004. *Understanding Ldap - Design And Implementation.* s.l. : IBM.com/redbooks, 2004. 073849786X.

International Telecommunication Union. 1993. X.500 : Information technology - Open systems Interconnection - The directory: Overview of concepts, models and services. *International Telecommunication Union.* [Online] November 1993. [Cited: March 22, 2015.] http://www.itu.int/rec/T-REC-X.500-199311-S.

IT Services. 2015. Oxford SSO Integration of AD via a Cross-Realm Trust | IT Services Help Site. *IT Services Help Site*. [Online] Oxford IT Services, March 17, 2015. [Cited: March 22, 2015.] http://help.it.ox.ac.uk/services/iam/kerberos/ad-xrt-howto.

Kohl, John a Neuman, Clifford. 1993. The Kerberos Network Authentication Service (V5). *RFC1510.* [Online] IETF, September 1993. [Citace: 14. February 2015.] http://tools.ietf.org/html/rfc1510.

Microsoft. 2014. Domain Controller Roles: Active Directory:. *Technet.* [Online] Microsoft, November 19, 2014. [Cited: February 28, 2015.] https://technet.microsoft.com/en-us/library/cc786438%28WS.10%29.aspx.

—. 2005. Group Policy features that are supported across forests. *Technet.* [Online] Microsoft, January 21, 2005. [Cited: Feburary 22, 2015.] https://technet.microsoft.com/en-us/library/cc737072%28v=ws.10%29.aspx.

—. 2015. User Naming Attributes. *Microsoft Developer Network.* [Online] Microsoft, January 1, 2015. [Cited: February 16, 2015.] https://msdn.microsoft.com/en-us/library/ms677605%28v=vs.85%29.aspx.

NASA. 2014. Research and Engineering: Systems Engineering and Integration Branch. *NASA.* [Online] NASA, February 19, 2014. [Cited: May 1, 2015.] http://www.nasa.gov/centers/armstrong/capabilities/CodeZ/flight/systems_engineering.html#.VVAKSZNVjmg.

Neuman, Clifford, et al. 2005. The Kerberos Network Authentication Service (V5). *RFC4120.* [Online] IETF, July 2005. [Cited: February 14, 2015.] http://ietf.org/rfc/rfc4120.txt.

Oxford University Press. Definition of authenticate in English. *Authenticate.* [Online] Oxford University Press. [Cited: April 2, 2015.] http://www.oxforddictionaries.com/definition/english/authenticate.

—. Definition of authorization in Oxford Learner's Dictionary. *Oxford Dictionaries.* [Online] Oxford University Press. [Cited: April 10, 2015.] http://www.oxforddictionaries.com/us/definition/learner/authorization.

2015. Samba - opening windows to a wider world. *Home of Samba, the SMB file server.* [Online] Samba.org, 01. 01 2015. [Citace: 30. March 2015.] https://www.samba.org/samba/.

Sturlese, Luca. 2013. PowerShell: How to easily create log files for your scripts | 9to5IT. *9to5IT.* [Online] 9to5IT, 19. February 2013. [Citace: 12. September 2014.] http://9to5it.com/powershell-logging-function-library/.

11 Seznam obrázků

12 Seznam grafů