Kerberos

Vítejte u prvního dílu seriálu, který se věnuje protokolu Kerberos se zaměřením na Single Sign-On (SSO) v prostředí Microsoft Active Directory.

Dnešní díl se přímo Kerberos protokolu nevěnuje, ale popíšeme si základní termíny Active Directory, které potřebujeme znát, a s kterými Kerberos autentizace souvisí (když ji používáme v doménovém prostředí). Nejprve stručně zmíníme AD komponenty, protože struktura souvisí s Kerberos Realm. Popíšeme si, jak klient hledá doménový řadič (což je zároveň Kerberos autentizační server). A docela podrobně se podíváme na přihlašovací jména uživatelských účtů (User Principal Name) a jména instancí služeb (Service Principal Name).

V dalších dílech si popíšeme obecně vlastnosti a funkce Kerberos protokolu. Podíváme se na to, co vlastně znamená SSO a jaké jsou možnosti. Rozebereme si jednotlivé komponenty a prvky, které Kerberos protokol používá. Popíšeme si proces Kerberos autentizace (tedy SSO) od zjednodušeného, přes podrobnější, až k detailnímu popisu. Podíváme se na fungování SSO autentizace mezi různými doménami. A zmíníme možnosti využití Kerberos SSO pro webové aplikace.

Jestli si někdo myslí, že této oblasti dobře rozumí, tak může dnešní díl přeskočit. Myslel jsem si to i já, ale když jsem studoval určité specifické detaily (například okolo UPN), tak jsem našel řadu nových informací.

Microsoft AD DS

Samozřejmě si tu nepopíšeme vše (a s určitou znalostí AD DS počítáme), nějaké další věci jsem popsal již před lety v článcích Adresářové služby a LDAP, Active Directory komponenty - domain, tree, forest, site a DNS (Domain Name System) zaměřeno na Microsoft. Odkaz na více materiálů se nachází na konci článku.

Active Directory Domain Services

Active Directory (AD) je adresářová služba od Microsoftu. **Active Directory Domain Controller** (DC či doménový řadič nebo pouze řadič) je server na kterém běží **Active Directory Domain Services** (AD DS, tento název se používá od Windows Server 2008, předtím se používal pouze Active Directory). AD DS poskytuje distribuovanou databázi, která v hierarchické struktuře obsahuje síťové objekty (jako je uživatel, počítač, skupina). Zajišťuje také autentizaci a autorizaci uživatelů a počítačů v síti. Využívá se LDAP protokol, rozšířený Kerberos verze 5 a DNS (Domain Name System).

Pro správu AD DS od Windows Server 2008 se využívá balíček **Remote Server Administration Tools (RSAT)**, který je součástí serverových OS a na klientské jej můžeme doinstalovat.

Doména (Domain).

Doména (Domain) je logická skupina počítačů, které sdílí společnou AD databázi. Když vytvoříme doménu, tak se automaticky vytvoří strom a les. Nejjednodušší je mít vše pouze jednou. V praxi to ale vždy nelze, takže musím vytvářet složitější struktury. Doména se ve schématech znázorňuje jako trojúhelník, prvky uvnitř jsou členy dané domény.

Jméno Active Directory **domény** je běžně plné **DNS** (Domain Name System) jméno (příklad firma.local). Může obsahovat písmena, číslice, pomlčku a tečku (pouze pro oddělení komponent), maximální délka je 64 znaků. Každá doména má ale také (z důvodu kompatibility) pre-Windows 2000 jméno, které se označuje jako **NetBIOS doménové jméno** (příklad firma). To může být maximálně 15 znaků dlouhé a nesmí obsahovat tečku a řadu dalších znaků.

Zobrazit doménová jména si můžeme například pomocí nástroje **Active Directory Domains and Trusts**. V okně vidíme seznam domén a jejich **DNS jména**, pokud na doménu klikneme pravým tlačítkem a zvolíme Properties, tak uvidíme **NetBIOS jméno** a také stupeň lesa a domény.

Strom (Tree) a hierarchie domén

V rámci **stromu** (Tree) existuje hlavní **kořenová doména** (Tree Root Domain), která má svůj **jmenný prostor** (namespace, třeba firma.local). Pod kořenovou doménou můžeme vytvořit další domény, které se označují **Child Domain**, ty obsahují jmenný prostor nadřazené domény (Parent Domain) a svoje relativní jméno (tedy třeba test.firma.local). Každá doména má vlastní AD databázi, která se nachází na všech doménových řadičích dané domény. V rámci stromu může existovat pouze jedna kořenová doména, pokud chceme druhou (hlavně kvůli novému namespace), tak musíme vytvořit nový strom. Strom se ve schématech znázorňuje jako ovál či kruh.

Les (Forest), schéma a vztahy důvěry

Jeden nebo více stromů se nachází v **lese** (Forest). Les sdílí společné AD schéma (Active Directory Schema). **Schéma** definuje AD databázi, co v ní může být uloženo a jakou to má strukturu. Každá doména má vlastní databázi (obsah), ale stejnou strukturu (schéma). V rámci lesa existuje jedna **kořenová doména**(Forest Root Domain), jde o prvně vytvořenou doménu v daném lese. Ta obsahuje skupiny Enterprise Admins a Schema Admins. Les se ve schématech znázorňuje jako obdélník či kruh.

Když chceme zjistit, jaká doména je **Forest Root**, tak můžeme spustit nástroj**Active Directory Domains and Trusts**, který také použijeme pro správu **vztahů důvěry**. Zvolíme **Action – Change forest** a zde vidíme aktuální kořenovou doménu.

Všechny domény v rámci lesa mají automaticky vytvořenou **důvěru** (Trust). Jde o**Two-way Transitive Trust**, to znamená, že vztah důvěry je obousměrný a není omezen na dvě přímo propojené domény, ale může se rozšířit dále (pokud má druhá doména důvěru s další doménou, tak této třetí doméně důvěřuje i ta první). Vytváří se důvěra mezi kořenovými doménami stromů v rámci lesa, ta se označuje **Tree-Root Trust**. A mezi nadřízenými doménami v rámci stromu, to je**Parent-Child Trust**. Ručně můžeme vytvořit důvěru mezi různými lesy (navazuje se mezi Forest Root Domain), to je **Forest Trust**, a dále Shortcut Trust, Realm Trust a External Trust.

Pro správu schématu slouží nástroj **Active Directory Schema**. Ten se standardně nenachází mezi **Administrative Tools** (ani když jsme nainstalovali RSAT, který potřebujeme), ale je třeba zaregistrovat knihovnu regsvr32 schmmgmt.dll. Potom můžeme použít **mmc konzoli** a přidat Snap-in **Active Directory Schema** (nejlépe následně uložit).

Global Catalog

V rámci lesa je důležitá služba **Global Catalog** (GC). Ta obsahuje index pro všechny objekty v rámci lesa (jde o částečnou kopii adresářových oblastí pouze pro čtení), ke každému má pouze základní informace. V každé doméně musí být alespoň jeden GC server (a ten je zároveň DC). Doménový řadič má údaje pouze o objektech, které patří do jeho domény. Pokud chceme přistoupit k prvku z jiné domény, tak se právě využije Global Catalog, který nám poskytne informace, na jakém doménovém řadiči se tento objekt nachází.

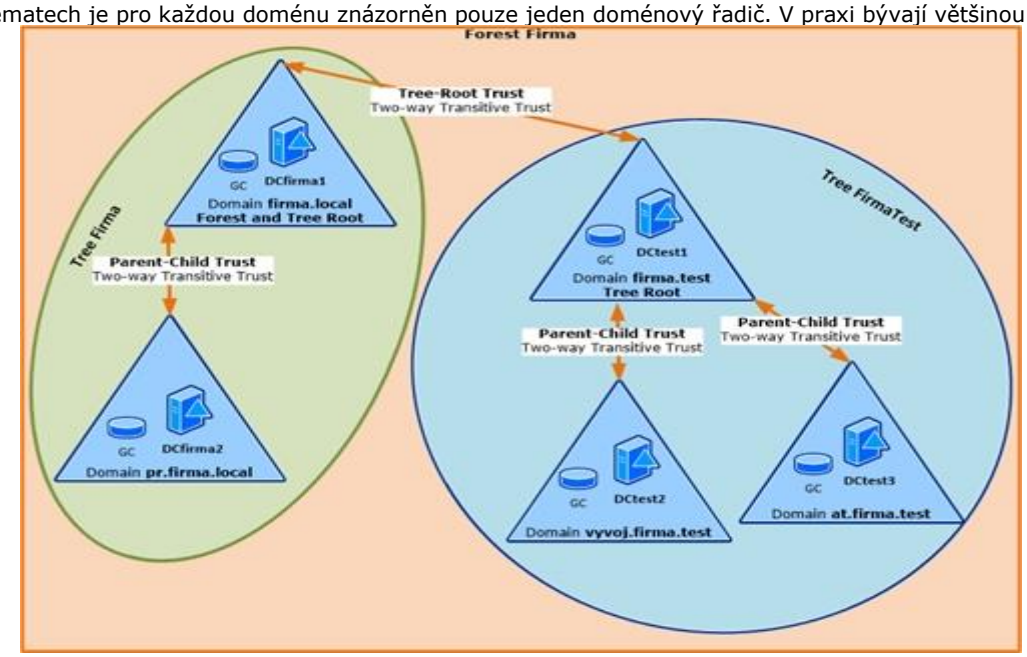
Pro nastavení **globálního katalogu** se používá nástroj **Active Directory Sites and Services**, kde se proklikáme přes odpovídající Site na hledaný DC a pod ním klikneme pravým tlačítkem na **NTDS Settings** a zvolíme Properties.

Schéma doménových vazeb

Následující obrázek jednoduše zobrazuje výše popsané. Je zde prostředí s**jedním lesem, dvěma stromy** a celkem **pěti doménami**. První byl instalován řadič DCfirma1, čímž vznikla doména firma.local (a strom i les) a stala se kořenovou pro strom i celý les. K této doméně byla přidána podřazená doménapr.firma.local. Byl také vytvořen druhý les, který má kořenovou doménufirma.test. V tomto lese jsou dvě podřazené domény.

Automaticky se vytvořily vztahy důvěry, takže z domény pr.firma.local můžeme komunikovat s objekty ve vyvoj.firma.test (musí to být samozřejmě zařízeno síťově, zde řešíme logickou část). Komunikace postupuje po vztazích důvěry, nejprve na kořenovou doménu stromu firma.local, přes kořenový vztah lesa nafirma.test až na vyvoj.firma.test.

Pozn.: Ve schématech je pro každou doménu znázorněn pouze jeden doménový řadič. V praxi bývají většinou minimálně dva.



Nalezení doménového řadiče potažmo KDC

Když se uživatel přihlašuje v rámci domény, tak musí nalézt vhodný **doménový řadič** (DC). V případě Kerberos autentizace kontaktuje službu **Authentication Service** (AS) na **Key Distribution Center** (KDC), tato služba běží na každém DC.

Řadič se hledá pomocí **DNS SRV záznamu** (případně NetBIOS) pro danou doménu. Obecně jde o záznam _ldap._tcp.DnsDomainName (třeba_ldap._tcp.firma.local), ale Microsoft používá speciální záznamy, aby hledal pouze Windows LDAP servery, _ldap._tcp.dc._msdcs.DnsDomainName (třeba_*ldap._tcp.dc._msdcs.firma.local*) Klient získá seznam všech dostupných řadičů a odešle na ně LDAP dotaz, oni odpoví základními informacemi. Klient použije první DC, který odpověděl a pomocí své IP adresy a subnetu zjistí, do jaké patří Site. Pomocí DNS SRV záznamu_ldap._tcp.SiteName._sites.dc._msdcs.DnsDomainName (třeba_*ldap._tcp.Praha._sites.dc._msdcs.firma.local*) zjistí seznam všech DC v dané Site. Na všechny odešle LDAP dotaz, první který odpoví, použije pro autentizaci. Výsledek je, že se upřednostňují řadiče ve stejné Site jako klient, pokud jich je více, tak se volí pomocí Round Robin.

Pro některé situace je třeba nalézt Global Catalog (GC), to se provádí obdobně pomocí záznamu _ldap._tcp.gc._msdcs.DnsForestName (třeba_*ldap._tcp.gc._msdcs.firma.local*). Existují i další záznamy pro GC. Identicky existuje několik záznamů pro Kerberos KDC, některé obecné a jiné speciálně pro Microsoft servery. Pro MS je hlavní _kerberos._tcp.dc._msdcs.DnsDomainName(třeba _*kerberos._tcp.dc._msdcs.firma.local*).

Pozn.: Všechny tyto SRV záznamy si registruje NetLogon služba na DC při svém startu.

Uživatelské účty

Uživatelský účet je objekt v Active Directory a jedná se o **Security Principa**l, takže umožňuje autentizaci a autorizaci. Pro správu účtů můžeme použít **Active Directory Users and Computers** (ADUC). Kerberos slouží k autentizaci uživatelů, takže je důležité jaké uživatelské jméno z AD používá.

Každý objekt v AD má několik **různých jmen** (Naming Attributes – možností jak na objekt odkazovat či jej identifikovat). Pro provázání se používají ID, takže ostatní jména můžeme měnit, aniž by se ovlivnili různé vazby. Možné formy jména objektu, spolu s těmi, které jsou speciálně pro uživatelský účet:

- **Relative Distinguished Name (RDN)** – relativní LDAP jméno v rámci kontejneru (Organizational Unit), u uživatelů jde o Common Name (CN), atribut cn, příklad bouska
- **Distinguished Name (DN)** – globální unikátní LDAP jméno, obsahuje RDN a umístění objektu v rámci AD hierarchie, atribut distinguishedName, příkladCN=bouska,CN=Users,DC=firma,DC=local
- **Canonical Name** – obdoba DN v jiné notaci, jde o konstruovaný atribut (vytváří se při dotazu) canonicalName, příklad firma.local/Users/bouska
- **Name** – je stejné jako **Common Name** (CN), atribut name
- **Security Identifier (SID)** – unikátní identifikátor uživatele, používá se pro Kerberos a zařazování do skupin, atribut objectSid
- **Globally Unique Identifier (GUID)** – unikátní identifikátor objektu, atributobjectGUID

Active Directory používá v **DN názvu** tři klíčová slova dle LDAP normy, jde o **CN - Common Name**, **OU - Organizational Unit** a **DC - Domain Component**. Oproti tomu **Canonical Name** využívá DNS formát.

Uživatelský účet má dva možné typy přihlašovacího jména (Logon Name), jsou to zároveň také Naming Attributes:

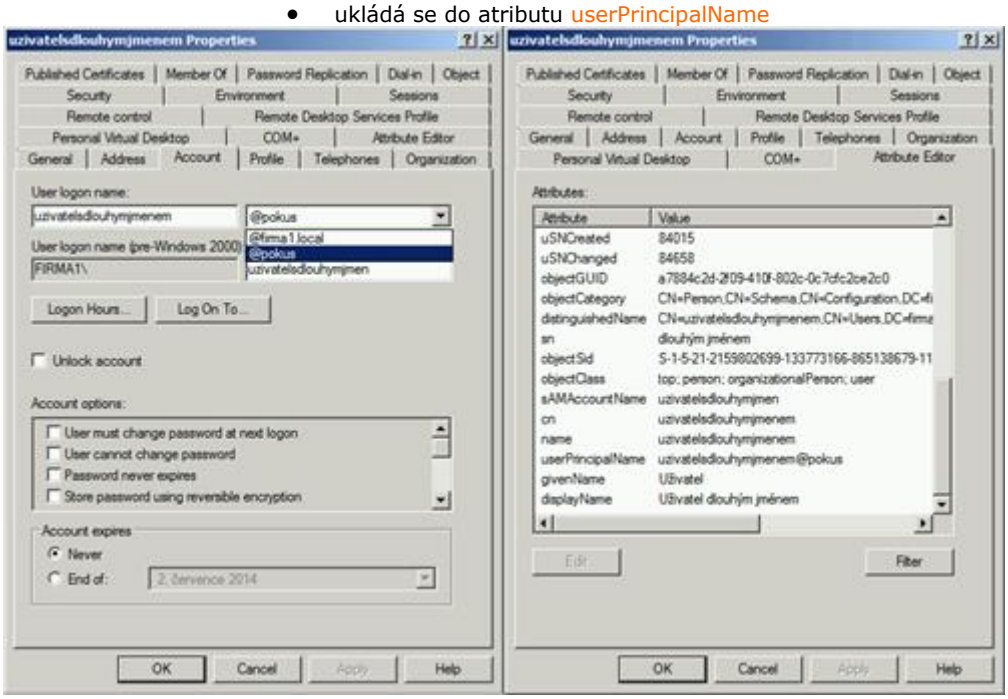
sAMAccountName - Security Accounts Manager (SAM) Account Name

- označuje se také jako **Pre-Windows 2000 Name**
- používá se zápis Domain\sAMAccountName, který se označuje jako**NetBIOS Logon Name**
- maximální délka je 20 znaků
- je unikátní v rámci domény a povinný
- ukládá se do atributu sAMAccountName

User Principal Name (UPN)

- označuje se jako *Internet-style login name* a je založen na RFC 822
- skládá se ze dvou částí, **UPN předpony** (UPN prefix) - uživatelské přihlašovací jméno (User Logon Name) a **UPN přípony** (UPN suffix) – DNS doménové jméno, spojených pomocí znaku @, příkladlogon.name@firma.local
- maximální délka je 64 znaků
- je unikátní v rámci lesa a nepovinný

- ukládá se do atributu userPrincipalName



User Principal Name (UPN)

Protože věci okolo UPN nejsou úplně jednoduché, tak se na ně podíváme podrobněji.

Microsoft začal **UPN** používat pro doménové uživatele v době **Windows 2000**. Mělo se jednat o **primární přihlašovací jméno**, které by měl uživatel používat. Jeho forma je kratší než Distinguished Name a MS představa byla, že bude stejné jako emailová adresa (což v praxi může zjednodušit práci útočníkovi, který nemusí hádat jméno, ale pouze heslo), takže bude pro uživatele jednoduché pro pamatování. Oproti DN také není závislé na přesunu do jiného kontejneru ani přejmenování domény (přípona nemusí odpovídat doméně). Přesto bych řekl, že i v kombinaci Windows Server 2012 a Windows 8 se hojně využívá **NetBIOS Logon Name**.

Jako **UPN přípona** se defaultně nabízí DNS jméno aktuální a kořenové domény, ale můžeme přidat alternativní doménové jméno (v podstatě libovolný řetězec v DNS formátu), které vytvoří administrátor v rámci lesa. Správa se provádí pomocí **Active Directory Domains and Trusts**, kde klikneme pravým tlačítkem na stejně pojmenovanou položku a zvolíme **Properties**.

V souvislosti se **sAMAccountName** Microsoft doporučuje používat začátek UPN shodný. Ale protože je zde omezení na 20 znaků, tak v řadě případů se třeba v UPN používá jméno a příjmení, ale v sAMAccountName první písmeno jména a příjmení.

Microsoft v některých materiálech uvádí, že uživatelský účet má vždy**sAMAccountName** i **User Principal Name**. Jiné potvrzují to, co můžeme ověřit v praxi, že povinný je pouze atribut sAMAccountName a nikoliv userPrincipalName. Přesto je tu zajímavá informace, kterou jsem nalezl pouze v neoficiálních materiálech. A to, že **UPN existuje vždy**, i když není atributuserPrincipalName vyplněn. Někde mluví o defaultním UPN, někde o**implicitním UPN**, ale ve výsledku je to stejné (a možná ani nejde o UPN, ale prostě o jiný zápis sAMAccountName). Ať máme nebo nemáme vyplněné UPN, tak můžeme použít username@DnsDomainName, kde **username** je*sAMAccountName* a ***DnsDomainName*** je *DNS jméno domény*, kde se účet nachází. Pokud atribut userPrincipalName vyplníme, tak tím definujeme další **explicitní UPN**, kde můžeme použít libovolné uživatelské jméno a příponu (tu musíme zaregistrovat v rámci lesa).

Co je **uživatelské jméno** je hodně důležité pro Kerberos autentizaci. Kerberos užívá termín **Client Name** (Principal Name) a **Realm**, často narazíme i na použití*User Principal Name*. V AD doméně je *Realm* rovno *DNS doméně* a praktickými pokusy jsem ověřil, že *Client Name* je rovno *sAMAccountName*. Ve výsledku to odpovídá implicitnímu UPN.

Abych výše uvedené ověřil v praxi, tak jsem provedl následující testy. V **doméně s DNS jménem** firma1.local a **NetBIOS jménem** firma1 jsem vytvořil UPN příponu pokus a uživatele s následujícími jmény:

distinguishedName CN=uzivatelsdlouhymjmenem,CN=Users,DC=firma1,DC=local
sAMAccountName uzivatelsdlouhymjmen
cn uzivatelsdlouhymjmenem
name uzivatelsdlouhymjmenem
userPrincipalName uzivatelsdlouhymjmenem@pokus

Pak jsem se zkoušel přihlásit. Následující jména prošla:

- FIRMA1\uzivatelsdlouhymjmen - NetBIOS logon name
- uzivatelsdlouhymjmenem@pokus – zadané UPN
- uzivatelsdlouhymjmen@firma1.local – implicitní UPN
- firma1.local\uzivatelsdlouhymjmen – použití DNS jména domény v NetBIOS přihlášení

Přihlášení se nepodařilo:

- FIRMA1\uzivatelsdlouhymjmenem
- uzivatelsdlouhymjmenem@firma1.local
- firma1.local\uzivatelsdlouhymjmenem

V popisu u MS jsem se o přihlašování dočetl následující. Když použijeme pro**přihlášení sAMAccountName**, tak zároveň určujeme doménu, v které se účet nachází (případně se použije aktuální). Při **použití UPN** není jasná doména (zadáváme pouze příponu, která nemusí odpovídat doméně), takže se nejprve hledá v **aktuální doméně**, pokud se UPN nenalezne, tak se použije **globální katalog** a hledá se doména, kde se účet nachází.

Při pokusech s různým přihlášením jsem také **zachytával komunikaci**. Aby bylo vše zajímavější, tak jsem se přihlašoval z jiné důvěryhodné domény (šlo o*firma2.local*). Když se klient snaží přihlásit, tedy **získat TGT**, tak v žádosti posílá přihlašovací údaje,

které jsme zadali (viz. výše uvedené příklady). Pokud úspěšně získáme tiket, tak je v něm vždy **DNS doména** a uživatelské jméno**sAMAccountName**.

Pokud zadáme tvar doména\jméno, tak se hledá **jméno** (porovnává se s atributem *sAMAccountName*) v rámci domény **doména** (jedno jestli jde o NetBIOS nebo DNS). Tedy v **KRG-AS-REQ** paketu se použije **Client Name = jméno, Realm = doména**.

Pokud zadáme jméno@doména (tedy UPN), tak se nejprve hledá v **lokální doméně**, použije se **Client Name = jméno@doména** (porovnává se s atributem*userPrincipalName*), **Realm = DNS jméno aktuální domény**. Pokud se nenalezne, tak DC hledá pomocí **globálního katalogu**, jestli UPN existuje v jiné doméně nebo přípona odpovídá jiné doméně. Pokud ano, tak vrátí odkaz na doménu. Klient provede nový dotaz s jiným **Realm** a zaslaný na DC z odkazu.

Informace o uživateli na stanici

Na stanici můžeme použít určité řádkové příkazy, abychom získali informace o uživateli. První příkaz je dotaz na libovolného **uživatele v doméně**:

```
C:\>net user bouska /domain
The request will be processed at a domain controller for domain firma.local.
User name                    bouska
Full Name                    Bouška Petr
Comment                      Ing. Petr Bouška
User's comment
Country/region code          (null)
Account active               Yes
Account expires              Never
Password last set            1.3.2014 14:54:14
Password expires             Never
Password changeable          1.3.2014 14:54:14
Password required            Yes
User may change password     Yes
Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   19.5.2014 16:24:41
Logon hours allowed          All
Local Group Memberships
Global Group memberships     *G Jabber IM *Domain Users
The command completed successfully.
```

Druhá možnost je řada různých parametrů známého příkazu whoami, který zobrazí aktuálně přihlášeného uživatele.

```
C:\>whoami /upn
bouska@firma.local
C:\>whoami /fqdn
CN=Bouška Petr,OU=IS,OU=Firma,DC=firma,DC=local
C:\>whoami /logonid
S-1-5-5-0-1835268
C:\>whoami /user /fo list
USER INFORMATION
----------------
User Name: ok\bouska
SID: S-1-5-21-2200232112-3866456066-1594429224-1223
```

Hodně informací se můžeme dozvědět i z jednoduchého výpisu **systémových proměnných**. Je zde vidět **jméno počítače**, **autentizačního severu** (DC),**domény** (NetBIOS i DNS) či **uživatele**. Ukázkový kousek výpisu:

```
C:\>set
COMPUTERNAME=BOUSKAP
LOGONSERVER=\\DC
USERDNSDOMAIN=FIRMA.LOCAL
USERDOMAIN=FIRMA
USERNAME=bouska
USERPROFILE=C:\Users\bouska
windir=C:\WINDOWS
```

Service Principal Name (SPN)

Service Principal Name (SPN) se využívají v Active Directory, kde jsou uloženy v atributu servicePrincipalName u účtu. Jde o jeden ze základních prvků fungování Kerberos autentizace, spolu se službou DNS. **SPN je unikátní identifikátor služby běžící na serveru**. Jinak řečeno, je to jméno, pomocí kterého klient jednoznačně identifikuje instanci služby v rámci lesa. Každá služba, kde chceme využít Kerberos autentizaci, musí mít SPN registrované na účtu, který používá k přihlášení. Jedna služba může mít více rozdílných SPN, stejně tak k jednomu účtu může být přiřazeno více SPN.

SPN syntaxe je obecně ServiceClass/Host:Port/ServiceName, ale nejčastěji jde pouze o ServiceClass/Host. První část je název, který **identifikuje obecně třídu služby**, příkladem je *ldap, GC, HOST, DNS, HTTP, TERMSRV, MSSQLSvc*. Pro webové servery, ať jde o protokol HTTP nebo HTTPS, se používá HTTP. Druhá část je**jméno počítače**, kde služba běží. Většinou se používá **FQDN** (Fully Qualified Domain Name), ale pokud ke službě přistupujeme přes **NetBIOS jméno**, tak musíme použít to. Stejně tak v některých případech potřebujeme přidat DNS alias. Naštěstí SPN můžeme registrovat více. Příkladem SPN pro webový server je HTTP/www.firma.local.

Ve Windows (od Windows 7 / Server 2008) máme k dispozici nástroj pro příkazový řádek setspn, který slouží k **prohlížení, nastavování a mazání SPN** v Active Directory. Pro určité operace potřebujeme patřičné oprávnění. Můžeme tak ověřit existenci určitého SPN, či vyhledat všechna SPN pro danou službu či server. Několik příkladů hledání pomocí setspn:

```
setspn -Q HTTP/server.firma.local
setspn -Q HTTP/*
setspn -Q */server.firma.local
setspn -Q */*
```

Pokud chceme hledat v rámci celého lesa, tak musíme přidat přepínač F.

setspn –F -Q HTTP/server.firma.local

Další příklad ukazuje registraci nového SPN pro webový server k účtu webserver.

setspn –S HTTP/www.firma.local firma\

Kerberos, část 2 – popis metody SSO a protokolu Kerberos

Vítejte u druhé části seriálu, který se věnuje protokolu Kerberos se zaměřením na Single Sign-On (SSO) v prostředí Microsoft Active Directory.

V minulém díle jsme si popsali Microsoft prostředí a termíny, které budeme využívat při popisu protokolu Kerberos a jeho navázání na Active Directory Domain Services. Dnes začneme tím, že se pokusíme objasnit, co znamená termín Single Sign-On. Dále si obecně popíšeme Kerberos a jeho vlastnosti a budeme se věnovat všem hlavním prvkům a termínům, z kterých se Kerberos protokol skládá.

Koho zajímá pouze hrubý princip Kerberos autentizace (SSO), tak může počkat na příští díl, kde si jej popíšeme zjednodušeně i detailně. Ale pokud chceme hlubší porozumění, tak je potřeba rozumět jednotlivým detailům, kterým se věnujeme dnes.

Co je to SSO?

Běžně užívaná zkratka **SSO** pochází z názvu **Single Sign-On**, česky se používá označení **jednotné přihlašování**, ale já preferuji anglický termín. Jde o metodu, kdy zadáme svoje přihlašovací údaje pouze jednou, a autentizace k dalším službám proběhne bez našeho zásahu (a zadání přihlašovacích dat). Správa našich údajů se provádí na centrálním místě (serveru) a na něm se autentizujeme. Často je SSO spojováno s **centrální správou uživatelských údajů** (Identity Management), což nám zajistí společné uživatelské jméno, heslo (či alternativní autentizaci jako je certifikát) a další údaje (třeba email a adresu) pro různé služby. Pokud dojde ke změně nějakého údaje, tak je měníme na jednom místě a projeví se to pro všechny služby, které využíváme.

Nejen, že jde o **pohodlné řešení**, ale také **zvyšuje bezpečnost**. Přihlašovací údaje se vůbec nedostanou na službu, ověřujeme se na centrálním místě a službě pouze předáme důvěryhodnou informaci. Takže také pro aplikaci není důležité, jakou metodou došlo k autentizaci uživatele. Nemusí docházet k přímé komunikaci mezi službou a autentizačním serverem, s tím komunikuje klient. Samozřejmě to znamená i nebezpečí, pokud by bylo napadeno centrální místo, tak získá útočník přístup k různým systémům.

SSO se dá prakticky realizovat různým způsobem. Každá metoda přináší určité výhody i nevýhody.

Kerberos SSO

V tomto seriálu se budeme věnovat rozšířené metodě SSO a to využití protokolu Kerberos. Naše zaměření bude speciálně na **Kerberos v podání firmy Microsoft**, což znamená **doménové prostředí**. V tomto případě provádí autentizaci služba KDC, která běží na všech **doménových řadičích** (DC). Klient tedy při využití SSO musí mít dostupný DC. Tuto službu asi nebudeme publikovat do internetu, takže je SSO omezeno na využití v interní síti. Na druhou stranu je možno nastavit důvěryhodnost s jinými doménami (pokud máme zajištěnu síťovou komunikaci, třeba pomocí VPN) a pak SSO funguje i mezi doménami.

Služba, ke které se chceme připojit, může běžet na Windows serveru/stanici, který je i není zařazen do domény. Stejně tak může být na Linuxu. Není ani nutné, aby se služba nacházela v lokální síti, může být v rámci internetu či jiné sítě.

Pokud jsme členem **domény**, tak si možná ani neuvědomujeme, že k **SSO dochází velice často**. Na začátku se přihlásíme do domény (při autentizaci do počítače) a při přístupu k dalším službám (které mají SSO nastavené) dojde automaticky k přihlášení. To nastává pokaždé, když použijeme síťovou tiskárnu, sdílený disk, připojíme se Outlookem k Exchange serveru či řadě dalších síťových služeb. Aniž bychom něco zaznamenali, tak na pozadí dojde k naší autentizace (a následně autorizaci) pomocí SSO. Většinou pokud neprojde autorizace (nemáme práva k dané službě), tak se zobrazí přihlašovací dialog a můžeme použít jiný účet.

Kerberos autentizace využívá vždy přihlášení k nějaké službě a v takovém případě se vždy využívá princip Single Sign-On.

SSO pro web

V internetu se SSO nejvíce využívá pro webové aplikace. V rámci interní sítě nám pro webové servery funguje Kerberos SSO, ale klient se musí nacházet uvnitř interní sítě. To se nehodí pro domácí uživatele, kteří nejsou součástí domény. A je to často problém i pro firemní použití.

Proto existují dva způsoby řešení. Ve firemním prostředí se využije speciální server **Identity Provider**, který nabízí služby prokázání identity do internetu. Interně provádí autentizaci uživatele vůči adresářové službě (třeba Active Directory). Znamená to tedy, že můžeme doménové účty využít k autentizaci ke službám v internetu (používá se třeba u hybridních cloudů). Často se využívá otevřený standard **Security Assertion Markup Language** (SAML a novější SAML2).

Druhá možnost je **služba v internetu**, která spravuje uživatelské účty a provádí autentizaci. Můžeme se tak do různých webů přihlásit jedním účtem (a v ideálním případě pomocí SSO). Něco takového využíváme například u Google, Facebooku či Microsoftu. Některé systémy z této oblasti jsou **OpenAM** (dříve OpenSSO) či **WebAuth**. Jednotná správa identit je OpenID nebo český projekt mojeID.

Praktické použití je takové, že při prvním přístupu k webové službě, která využívá jednotný účet, zadáme přihlašovací údaje. Získáme šifrovanou identitu a při přístupu k další službě již není třeba údaje zadat. Na počátku se ale musíme přihlásit k počítači, kde se využívá jiný účet.

Když jsme zmínili Cloud, tak pro firemní prostředí se využívá ještě jedna možnost. Účty z interního Active Directory se **synchronizují** (třeba MS DirSync) do Cloudového adresáře nebo se provádí **federace** (třeba MS Active Directory Federation Services). Uživatelské účty pak fungují v rámci daných Cloudových služeb. Příkladem je třeba Microsoft Office 365.

Protokol Kerberos

Kerberos je síťový autentizační protokol, který pracuje na základě **tiketů**(Tickets), aby umožnil komunikujícím stranám v nezabezpečené síti bezpečně prokázat svoji identitu. Na začátku zadáme svoje přihlašovací údaje a veškeré další ověřování funguje na principu SSO.

Protokol Kerberos vznikl již v osmdesátých letech dvacátého století na univerzitě Massachusetts Institute of Technology (MIT). Dodnes používaná verze 5 je zde od roku 1993, kdy byla definována v RFC 1510 - The Kerberos Network Authentication Service (V5). Tato norma byla v roce 2005 upravena a specifikována v RFC 4120 - The Kerberos Network Authentication Service (V5). V Microsoftím doménovém prostředí je **Kerberos v5** výchozí autentizační protokol. Jako alternativa, či ve starších Windows (NT 4.0), se používá protokol**NTLM** (NT LAN Manager). Věci související s Kerberos protokolem jsou definovány v řadě dalších RFC norem.

Standardní Kerberos se využívá pro **autentizaci**, Microsoft jej rozšířil o**autorizační údaje** (posílá seznam skupin, kterých je uživatel členem, a případně další informace) pomocí Kerberos Protocol Extensions [MS-KILE]. Tyto údaje jsou uloženy v **Privilege Attribute Certificate (PAC) Data Structure** [MS-PAC].

Kerberos je hojně využíván (nejen v MS světě) díky svým vlastnostem:

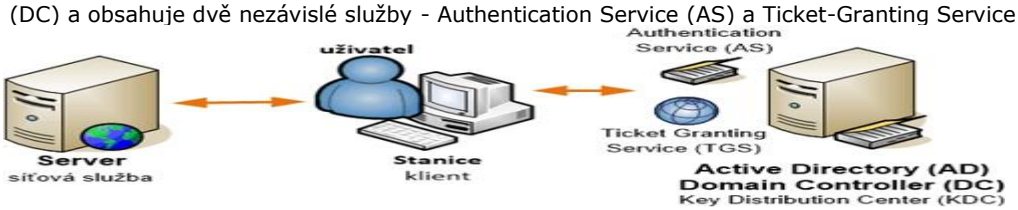
- uživatelské heslo ani jeho hash se nikdy **nepřenáší v síti** – vyměňují se šifrované tikety

- využívá se **symetrická kryptografie** – jsou podporování různé šifry (jako DES, RC4, AES)
- architektura je **rozšiřitelná** – umožňuje použít další či alternativní bezpečnostní metody (šifry, hashe)
- podporuje **vzájemnou autentizaci** (mutual authentication) – volitelně se nemusí pouze autentizovat klient u služby, ale i služba klientovi
- chrání proti **odposlechu** (eavesdropping) či **opakovanému použití stejných rámců** (replay attacks, ochrana pomocí časových známek)
- umožňuje uživatelům přistupovat k různým službám bez zadání hesla =**Single Sign-On**
- zjednodušuje **administraci** – účty jsou spravovány centrálně pro různé aplikace
- funguje v modelu klient server a vyžaduje **důvěryhodnou třetí stranu**(KDC) – to je zároveň slabina, KDC musí stále běžet a musí být dobře zabezpečené
- klient se **neověřuje na serveru**, ten vůbec nedostane uživatelovy credentials, ověření provádí KDC a odesílá šifrovaně uživatelovo jméno (to může rozšifrovat jen server)
- podporuje **impersonation** – uživatel přímo nepřistupuje ke zdrojům, místo něj přistupuje systém
- podporuje **delegation of authentication** (pomocí proxy ticket, forwarded ticket, protocol transition a constrained delegation) – umožňuje službě běžící pod nějakým účtem převzít uživatelovu identitu a přihlásit se k jiné službě (například uživatel se přihlásí k aplikačnímu serveru, ten se musí přihlásit k DB, ale potřebuje to provést pod oprávněním uživatele)

Kerberos protokol je považován za **bezpečný**, chrání proti řadě různých útoků či zneužití. Ze svého principu (jako je používání dočasných tiketů a šifrování různými klíči) a účelu (umožnit bezpečné ověření v nezabezpečené síti) jde o ochranu při **komunikaci v síti**. Zůstávají tedy místa, kde může dojít k zneužití. Jedním bodem je **uživatelské heslo**, pokud je zvoleno slabé heslo a útočník jej může odhadnout/získat, tak bezpečnost vlastního protokolu nezabrání zneužití účtu. Pro zvýšení bezpečnosti je možné použít čipovou kartu a autentizaci certifikátem. Druhé nebezpečí je **napadení stanice**, z které se uživatel přihlašuje. V paměti je dočasně uložen Secret Key, což je vlastně heslo uživatele.

Jméno protokolu vzniklo podle řecké mytologie, kde Kerberos je tříhlavý pes chránící vchod do podsvětí. To proto, že komunikace se účastní tři strany:

- **uživatel** – klient, který chce přistoupit k nějaké síťové službě
- **server** – přesněji řečeno služba, na kterou chce uživatel přistoupit
- **Key Distribution Center (KDC)** – důvěryhodná třetí strana, která provádí ověřování, je součástí doménového řadiče (DC) a obsahuje dvě nezávislé služby - Authentication Service (AS) a Ticket-Granting Service (TGS).



Kerberos V5 standardně využívá **TCP**(/UDP) **port 88**. Je plně integrován v Active Directory, serverovém i klientském OS od Windows 2000, a řadě síťových protokolů, jako je CIFS/SMB, HTTP, RPC, tiskové služby, DFS, IPsec, apod.**Kerberos** je závislý na řadě služeb, jde o **Active Directory Domain** (nefunguje pro lokální účty), T**CP/IP síť, DNS** (Domain Name System), **časové synchronizaci** (Time Service) a **SPN** (Service Principal Names).

Hlavní termíny Kerberos protokolu

Nejprve si vysvětlíme důležité pojmy a používané principy, které hrají roli u Kerberos autentizace. Pomocí nich si následně popíšeme princip ověřování.

Cryptography (kryptografie - šifrování)

Kerberos protokol je postaven na šifrování, proto si na začátku velmi stručně zmíníme hlavní termíny. **Kryptografie** se používá pro dosažení různých cílů. Jde o **Confidentiality** (důvěrnost, ochrana proti čtení dat), **Data integrity** (ochrana dat před změnou), **Authentication** (ověření, že data pochází od dané strany),**Non-repudiation** (neodmítnutelnost).

- **Encryption (šifrování)** – transformace dat (plaintext) za účelem jejich zabezpečení (ciphertext), takže pouze ten, kdo zná klíč, je může rozšifrovat, existuje šifrování se **symetrickým klíčem**, kde je stejný klíč pro šifrování i dešifrování, a šifrování s **veřejným klíčem**, kde šifrovací klíč je veřejně známý, takže každý může data zašifrovat, ale dešifrovat je může pouze majitel privátního klíče, příklad šifer je SSL, PGP, AES, DES, 3DES
- **Hashing (hašování)** – funkce, která z různě dlouhých vstupních dat vytváří kratší výstup pevné délky (hash), stejný vstup má vždy stejný hash, různé vstupy mají různý výstup, z výstupu nelze získat vstupní data, pokud dojde ke změně vstupu, tak se výrazně změní hash (využívá se pro zajištění integrity dat, kdy se k datům vytvoří kontrolní součet), příklad Hash funkcí MD5, SHA, SHA2
- **Encoding (kódování)** – transformace dat do jiného formátu (aby se mohli lépe zpracovávat), proces je veřejně známý a může být obrácen, příklad kódování je Base64

Key Distribution Center (KDC)

KDC je server, který autentizuje klienta a vystavuje tikety. Jde o centralizovanou důvěryhodnou třetí stranu v Kerberos komunikaci. U MS je součástí každého doménového řadiče (DC) a obsahuje dvě nezávislé služby (které se obě nachází na DC):

- **Authentication Service (AS)** – provádí autentizaci uživatele a odesílá mu TGT
- **Ticket-Granting Service (TGS)** – na základě TGT ověří uživatele a odesílá Service Ticket pro požadovanou službu

KDC přistupuje k Active Directory informacím o účtech, včetně hesel (hashů). Pro ověření se uživatelské heslo využívá pouze k získání TGT, a ani v této fázi se neposílá po síti, dále se již využívá TGT.

Secret Key – tajný klíč

Tajný klíč slouží k ověření uživatele pomocí šifrování/dešifrování komunikace, sdílí jej mezi sebou Principal (klient či služba) a KDC. Používá se několik označení, asi nejčastější je **Secret Key**, další je **Shared Secret Key** (sdílený tajný klíč) či obecně **Encryption Key** (šifrovací klíč). Secret Key vznikne z *textového hesla*, ke kterému se přidá *sůl* (Salt), což je *Principal Name* včetně *Realmu* (to zajistí, aby ani uživatelé se stejným heslem, neměli stejný klíč). Tento řetězec zpracuje funkce**string-to-key** (string2Key), která provede jednosměrnou **hashovací funkci**(nelze zpět získat heslo), a výsledkem je klíč.

Princip ověření je takový, že tajemství je známo pouze dvěma stranám (uživatel a KDC). Každá strana může ověřit tu druhou tak, že druhá strana zná stejné tajemství. Aby se ověřilo, že druhá strana zná tajemství, tak Kerberos využívá**Secret Key**

Cryptography (šifrování pomocí klíče). Jedna strana zašifruje zprávu pomocí klíče, pokud ji druhá dokáže rozšifrovat, tak musí znát klíč (tajemství). Využívá se symetrické šifrování (stejný klíč dokáže data zašifrovat i rozšifrovat).
Termín **Secret Key** nejčastěji používáme pro klíč uživatele, ale šifrovacích klíčů se používá více:

- **dlouhodobé symetrické klíče** (Long-Term Symmetric Keys) – vytváří se z hesla, textové heslo se pomocí šifrovací funkce (například DES-CBC-MD5) transformuje na klíč. Patří sem **User key** (uživatel, heslo je zadáno v AD a na stanici se zadá při přihlášení), **System key** (když se počítač zařadí do domény, tak dostane heslo, z něj se vytvoří klíč), **Service key** (služby získají klíč z hesla účtu, který používají pro přihlášení), **Inter-realm keys** (pro autentizaci mezi doménami - cross-realm authentication, kde je vytvořen vztah důvěry)
- **dlouhodobé asymetrické klíče** (Long-Term Asymmetric Keys) – veřejný klíč pro autentizaci certifikátem ze Smartcard
- **krátkodobé symetrické klíče** (Short-Term Symmetric Keys) - **Session Key**pro šifrování komunikace mezi KDC a klientem nebo službou a klientem, KDC si jej neukládá, protože klient jej vždy pošle v šifrovaném tiketu

Tiket (ticket)

Tiket představuje Kerberos **credentials**, jde o záznam (uložený v paměti či v souboru na disku), který může být použit pro ověření identity. Tickety jsou základem autentizace v Kerberosu, kdy se po síti neposílají hesla, ale pouze šifrované tickety. Obsahuje identitu klienta, Session Key, časovou známku a další informace. Šifruje se pomocí serverového Secret Key.
Pro žádost a doručení tiketů se používají **Kerberos zprávy** (pakety). V žádosti (request) se posílá požadovaná (podporovaná) šifrovací metoda a požadované vlastnosti (flags). Na klientovi se tikety ukládají do vyrovnávací paměti (volatile memory space, ne na disk), takže se mohou použít opakovaně. Mají určitou dobu platnosti. Používáme dva typy tiketů:

- **Ticket-Granting Ticket (TGT)** – jde o speciální Service Ticket pro službu KDC (krbtgt), zařizuje bezpečné předání uživatelových credentials z AS na TGS, můžeme říci, že slouží k autentizaci uživatele, je šifrovaný pomocí **KDC Secret Key** (klient nemůže rozšifrovat), obsahuje údaje o klientovi, příznaky (flags), časové razítko, případné další údaje a Session Key (při použití TGT nemusí KDC hledat Secret Key klienta, ale rozšifruje si Session Key a ten použije), defaultní platnost je 10 hodin, může se provádět obnova (renew) bez potřeby zadat heslo, Microsoft rozšířil standardní tiket o autorizační data, takže je zde také SID uživatele a SIDy bezpečnostních skupin, kterých je členem
- **Service Ticket** – zařídí, aby se uživatelovy credentials bezpečně dostaly z TGS ke službě, slouží tedy k autentizaci uživatele u služby, obsahuje nešifrovaně SPN služby a šifrovaně pomocí **Secret Key služby** (takže klient nemůže rozšifrovat) údaje o klientovi (může zde být i adresa), dobu platnosti a Session Key pro komunikaci klienta a služby, Microsoft rozšířil standardní tiket o autorizační data, takže je zde také SID uživatele a SIDy bezpečnostních skupin, kterých je členem

Níže je příklad Service Ticketu, jak si jej můžeme vypsat na klientovi.

C:\>klist
Current LogonId is 0:0x4c2b5
Cached Tickets: (10)
#2> Client: bouska @ FIRMA.LOCAL
Server: HTTP/www.firma.local @ FIRMA.LOCAL
KrbTicket Encryption Type: RSADSI RC4-HMAC(NT)
Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
Start Time: 4/25/2014 12:53:44 (local)
End Time: 4/25/2014 22:18:52 (local)
Renew Time: 5/2/2014 12:18:52 (local)
Session Key Type: RSADSI RC4-HMAC(NT)
Cache Flags: 0
Kdc Called: dc.firma.local

Authenticator

Spolu s tiketem klient posílá i čerstvě generovaný **Authenticator** šifrovaný pomocí **Session Key** (ten je obsažen v tiketu). Služba může věřit, že tiket není podvržený, protože je šifrovaný jejím klíčem (který zná pouze KDC a ona). Nemá ale ověřeného **odesílatele** (protože vlastní tiket může být ukraden), proto se používá **Authenticator**.
Authenticator je šifrovaný klíčem (Session Key), který zná pouze správný klient a server. Obsahuje **časovou známku**, která se ověřuje, jestli není rovna nebo menší, než naposled přijatá časová známka od klienta (tím se zajistí, že authenticator není možno použít opakovaně). Také se kontroluje časová známka s časem na serveru a nesmí se lišit o víc než 5 minut (klient a server musí mít synchronizovaný čas). Dále obsahuje jméno a Realm klienta, kontrolní součet aplikačních dat a může zde být šifrovací klíč (který by se použil místo Session Key).

Pre-authentication

Použití **před-autentizace** zvyšuje bezpečnost přihlašovacího procesu. Při normálním průběhu klient žádá o TGT a v žádosti (KRG-AS-REQ) odesílá pouze svoje jméno a doménu. KDC nalezne uživatele a pomocí jeho **Secret Key**zašifruje **Session Key**, který mu pošle (spolu s TGT). Když klient nezná heslo (Secret Key), tak si nemůže rozšifrovat Session Key a tudíž dále komunikovat s KDC. Útočník by ale mohl poslat požadavek za jiného uživatele. A pak se pokusit provádět nějaké útoky **hrubou silou** na šifrovaný Session Key a tím by mohl zjistit heslo uživatele.
Proto se při Pre-authentication na požadavek KRG-AS-REQ odešle KRB-ERROR s hodnotou KRB5KDC_ERR_PREAUTH_REQUIRED. Na to klient odpoví novým KRG-AS-REQ paketem, kde do oblasti **PADATA** přidá **časovou známku** zašifrovanou pomocí uživatelova **Secret Key**. KDC rozšifruje časovou známku (tím ověřil uživatele) a ještě zkontroluje, jestli je čas v pořádku (stejně jako u Authenticator). Dále pokračuje běžným způsobem.
Pozn.: V Kerberos v5 je Pre-authentication volitelná, ale v Active Directory je defaultně vyžadována.

Realm - doména

Každá organizace, která chce provozovat Kerberos, si definuje svůj vlastní**Realm**. Jméno Realmu, v které je uživatel registrován, je součástí uživatelského jména. Může se navazovat vztah mezi Realmy, mohou být hierarchicky organizovány a nejčastěji se používá DNS zápis. V **Microsoft** prostředí je**Kerberos Realm** ekvivalentem pro **Active Directory doménu**. Kerberos Realm je závislý na velikosti znaků a standardně se používají velká písmena. KDC může vydávat tikety pouze pro svůj Realm. Příkladem Realmu je FIRMA.LOCAL.

Principal Name

Principal je unikátní **identita** (pojmenovaná entita klienta nebo serveru), které může Kerberos přiřadit tiket. Kerberos slouží k prokázání, že komunikující objekt byl u KDC registrován pod identitou, za kterou se vydává. **Principal** může mít řadu komponent, které se oddělují lomítkem /, ale nejčastější tvar je primary@REALM nebo primary/instance@REALM. Souvisí to s dříve popsanými UPN a SPN v Active Directory. Příkladem uživatelského Principal je bouska@FIRMA.LOCAL a Principal služby je HTTP/www.firma.local@FIRMA.LOCAL.

Pozn.: Principal se většinou používá včetně uvedení Realmu, ale není to podmínka.

Secure Channel

Je používán ve Windows, když se přenáší **Secret Key** (třeba když vytvoříme uživatele a zadáme jeho heslo), tak je přenos zabezpečen jiným **Secret Key**.

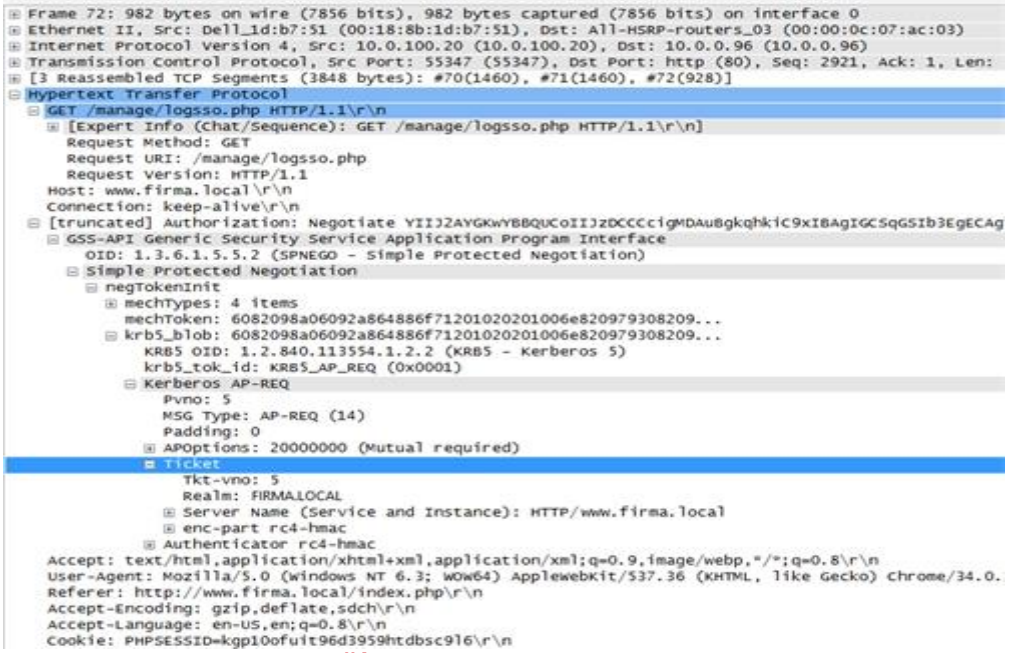
Kerberos zprávy (packety)

Kerberos protokol standardně komunikuje na **TCP portu 88**. Používá několik**typů zpráv** (Message Type), některé mohou být zabaleny v jiném protokolu. Podrobné informace o zprávách, tiketech, jejich obsahu a datových typech nalezneme ve specifikaci RFC 4120 - 5.Message Specifications.

Nejčastější zprávy spočívají ve výměně **požadavku** (Request) a **odpovědi** (Reply). První komunikace je mezi klientem a Kerberos serverem (tedy KDC). Pro získání**TGT** se komunikuje s **Authentication Service** (AS) a posílá se KRB-AS-REQ (10) a KRB-AS-REP (11). Pro získání **Service Ticket** se komunikuje s **Ticket-Granting Service** (TGS) a posílá se KRB-TGS-REQ (12) a KRB-TGS-REP (13).

Když se podíváme na obsah zpráv, tak **požadavky** mají dvě hlavní části. padata, což je zkratka **pre-authentication data**, zde mohou být informace potřebné pro zpracování nebo dešifrování. Třeba u KRB-TGS-REQ je zde **Authentication Header** (Ticket-Granting Ticket a Authenticator). A KDC-REQ-BODY, kde jsou vlastní data požadavku, třeba *Client Name, Realm, Server Name* (SPN). **Odpověď**může mít více částí, je zde třeba *Client Name, Client Realm* a Ticket, což je vlastní vystavený **Ticket**, v něm vidíme Server Name (SPN), Realm a šifrovanou část.

Pro **autentizaci uživatele** k aplikačnímu serveru komunikuje klient a server. Posílá se KRB-AP-REQ (14), který obsahuje hlavně tiket (Service Ticket) a authenticator. V *ap-options* může být příznak *mutual-required* a pak musí server odpovědět zprávou KRB-AP-REP (15) čímž autentizuje sebe u klienta. Objevit se může také zpráva KRB-ERROR (30) s nějakou **chybovou informací**. Existuje ještě několik dalších typů zpráv, ale ty se používají pro speciální případy.



Kerberos, část 3 – princip Kerberos autentizace

Vítejte u třetí části seriálu, který se věnuje protokolu Kerberos se zaměřením na Single Sign-On (SSO) v prostředí Microsoft Active Directory.

Po té, co jsme si vysvětlili potřebné základy a termíny patřící k *Active Directory Domain Services*, *Single Sign-On* a *Kerberos protokolu* (první díl seriálu, druhý díl seriálu), se dnes pustíme do hlavní části. Popis principu, jak ***funguje Kerberos autentizace***, což zároveň znamená také ***Single Sign-On***. Princip není úplně jednoduchý, používají se různé šifrovací klíče, časové známky, tikety, atd. Někomu stačí znát logický princip funkce a nepotřebuje vědět, jaké pakety se posílají a co obsahují. Proto je popis uveden třikrát, kdy každý jde více do hloubky.

Ač je vlastní princip ověřování stejný, tak se v různých situacích (třeba autentizace mezi doménami - Cross Realm Kerberos) přidávají části do celkového fungování. Řadu z nich si postupně popíšeme později.

Velmi jednoduchý popis

Princip Kerberosu se dá rozdělit do tří kroků. Předpokladem je, že jsme se na začátku přihlásili do Windows, kdy jsme zadali jméno a heslo (případně použili certifikát a PIN). Následně chceme přistoupit k nějaké síťové službě, ta podporuje Kerberos a tuto informaci nám předá spolu s požadavkem na autentizaci.

- ***získání TGT***– ověříme, jestli máme TGT, pokud ne, tak se kontaktuje KDC služba AS a získáme TGT, který se uloží do vyrovnávací paměti (většinou se získává jen na začátku při přihlášení do počítače):
- ***získání Service Ticket*** – kontaktujeme KDC službu TGS a požadujeme *Service Ticket* pro službu, kde se chceme přihlásit, ověřujeme se pomocí TGT
- ***autentizace u služby*** – *Service Ticket* odešleme aplikačnímu serveru a ten z něj získá důvěryhodně naše uživatelské jméno

Trochu podrobněji

Autentizace uživatele– získání TGT

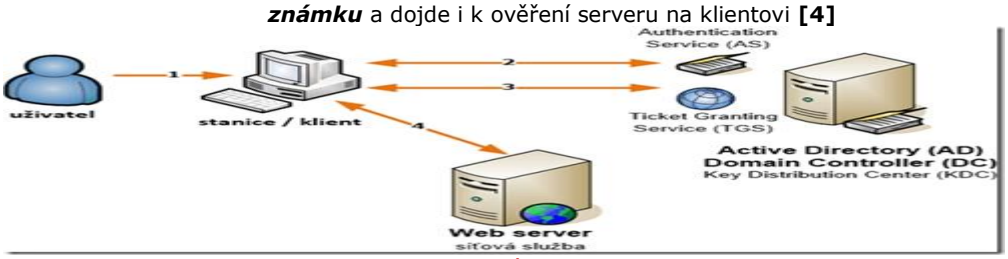
- při přihlašování do počítače zadá uživatel svoje přihlašovací údaje (credentials) na stanici **[1]**
- stanice si vytvoří uživatelův **Secret Key**
- nalezne se DC pro autentizaci a komunikuje se s **KDC AS** (která má přístup k AD databázi uživatelů)
- za pomoci šifrování se ***Secret Key*** se získá ***Ticket-Granting Ticket*** (TGT) **[2]**

Získání Service Ticket – když chce **klient přistoupit k nějaké síťové službě**(třeba aplikaci na webovém serveru), která podporuje Kerberos a vyžaduje autentizaci, tak se použije princip SSO

- klient se obrátí na svoje DC na ***KDC službu TGS***
- požádá o Service Ticket, *v* žádosti se posílá (mimo jiné) *typ služby* (třeba HTTP, KRBTGT, LDAP, CIFS) a *adresa serveru* (třeba www.firma.local), což dohromady tvoří ***Service Principal Name*** (SPN), tedy unikátní identifikátor služby běžící na serveru
- pokud **TGS** nalezne **SPN** v Active Directory (a jsou splněny další autentizační podmínky), tak odesílá v odpovědi ***Service Ticket*** **(ten je šifrovaný pomocí*Secret Key* služby) [3]**

Autentizace u služby – ***ověření klienta na síťové službě***(serveru)

- pomocí protokolu, kterým komunikuje aplikační server, odešle klient data na server a ta (mimo jiné) obsahují ***Service Ticket***
- server rozšifruje tiket pomocí svého klíče a z něj se dozví ***údaje o uživateli***(jeho jméno)
- ve vlastním procesu autentizace server nekomunikuje s KDC (tedy DC), ale pouze s klientem
- pokud je zapnuta ***obousměrná autentizace*** (Mutual Authentication), tak server odesílá klientovi šifrovanou ***časovou známku*** a dojde i k ověření serveru na klientovi **[4]**



Podrobný popis

Uvažujeme běžnou situaci v doménovém prostředí, kdy je vyžadována ***Pre-authentication*** a ***Mutual Authentication***.

Autentizace uživatele a ***získání Ticket-Granting Ticket***

- při přihlášení do počítače zadá **uživatel** své přihlašovací údaje (budeme uvažovat jméno a heslo) **[1]**
- **klient** k ***heslu*** přidá ***uživatelské jméno s doménou*** a provede jednosměrnou***hashovací funkci***, výsledkem je ***Secret Key***
- **klient** sestaví žádost KRB-AS-REQ a odešle ji na ***Authentication Server*** (AS) což je součást ***Key Distribution Center*** (KDC), v žádosti je část KDC_REQ_BODY, kde je uživatelské jméno a Realm (doména), SPN služby (krbtgt), adresa klienta, podporované šifry, vše v holém textu (nikdy zde není heslo ani Secret Key) **[2]**
- **AS** vyžaduje *před-autentizaci*, která v požadavku nebyla, takže odpoví chybovou zprávou KRB-ERROR s kódem KRB5KDC_ERR_PREAUTH_REQUIRED (25) **[3]**
- **klient** sestaví nový požadavek KRB-AS-REQ, který navíc v části padata obsahuje ***časovou známku*** (timesamp) zašifrovanou pomocí ***Secret Ke***y klienta **[4]**
- **AS** nalezne uživatele v AD a vytvoří si jeho ***Secret Key***, pomocí něj rozšifruje časovou známku (tím ověřil uživatele), ověří její platnost (jestli není proti času serveru posunuta o více než 5 minut a jestli již od klienta nepřišla stejná nebo novější)
- pokud bylo vše v pořádku, tak **AS** vygeneruje ***Session Key*** (pro šifrování další komunikace s klientem) a **Ticket-Granting Ticket** (TGT) pro klienta, připraví odpověď pro klienta KRB-AS-REP, do které vloží jméno uživatele, ***Session Key***zašifrovaný pomocí ***Secret Key klienta*** (část enc-part) a ***Ticket-Granting Ticket*** (TGT), který zašifruje svým klíčem (KDC Secret Key) **[5]**
- **klient** rozšifruje ***Session Key*** (tím se potvrdilo, že je to opravdu on) a uloží si jej pro další komunikaci s KDC, TGT rozšifrovat nedokáže, takže jej ani nemůže modifikovat
- ***TGT*** má omezenou životnost (default 10 hodin), ale může probíhat automatická ***reautentizace***, při každém přihlášení se vytváří nový TGT
- ***TGT*** se používá pro další komunikaci s KDC, aby se nemuselo používat ***Secret Key*** (a tedy heslo), obsahuje jméno klienta, adresu, dobu platnosti, Session Key
- při každém požadavku na KDC posílá klient ***TGT***, rozšifrovat jej může pouze KDC, takže důvěřuje obsaženým informacím, je tam i ***Session Key***, takže si KDC nemusí udržovat žádné informace v paměti (z toho plyne vysoká škálovatelnost)

Získání Service Ticket

- když se chce **klient** přihlásit k nějaké službě v síti (třeba webová stránka či souborové sdílení) pomocí Kerberos protokolu (a tedy SSO), tak sestaví žádost KRB-TGS-REQ a odešle ji na **Ticket Granting Service** (TGS), další součást **KDC [6]**
- ***žádost*** obsahuje část KDC_REQ_BODY, kde jsou údaje o žádosti, hlavně jméno služby a Realm (což dohromady tvoří SPN), podporované šifry, žádanou dobu platnosti tiketu, náhodně vygenerované číslo
- druhá část padata obsahuje autentizační údaje, což je hlavně náš ***TGT*** tiket (stále šifrovaný KDC klíčem) a ***Authenticator*** (šifrovaný pomocí Session Key, obsahem je hlavně identifikace klienta a časová známka), tyto údaje tvoří dočasné credentials uživatele
- **TGS** si z požadavku vezme ***TGT*** a ten rozšifruje svým klíčem, z něj se dozví***Session Key*** a pomocí něj rozšifruje ***Authenticator*** a ověří jeho data
- pokud jsou údaje v požadavku v pořádku, tak **TGS** připraví odpověď KRB-TGS-REP a odešle ji klientovi **[7]**
- do odpovědi vloží jméno uživatele, ***Service Ticket*** pro požadovanou službu, který obsahuje svoje SPN a v části šifrované pomocí ***Secret Key služby***uvedeno jméno uživatele a ***Session Key*** pro komunikaci mezi klientem a službou, a v části ***enc-part*** je ***Session Key*** pro komunikaci se službou (klient - služba) zašifrovaný pomocí ***Session Key*** z TGT (klient - KDC), pro šifrování by se mohl použít i klíč z ***Authenticator*** (pokud by jej sem klient umístil)
- ***klient*** získá šifrovaný ***Service Ticket*** a rozšifruje si ***Session Key*** pro komunikaci se službou, oboje uloží do cache

Ověření u služby (serveru)

- **klient** má nyní všechna potřebná data, aby se ***pomocí Kerberosu autentizoval na serveru***, když začal se službou komunikovat, tak se pomocí jejího ***protokolu*** (třeba HTTP) dozvěděl, že vyžaduje autentizaci a podporuje Kerberos
- **klient** nyní připraví odpověď v rámci stejného protokolu (třeba HTTP) do kterého zabalí další vyjednávací protokol (třeba SPNEGO GSS-API) a dovnitř již vloží Kerberos zprávu KRB-AP-REQ a odešle na server **[8]**
- do ***autentizačního požadavku*** vloží hlavně ***Service Ticket*** pro službu a***Authenticator*** (šifrovaný pomocí Session Key pro komunikaci se službou, obsahuje údaje o klientovi kontrolní součet, časovou známku)
- **server** rozšifruje ***Service Ticket***, z něj se dozví ***Session Key*** a rozšifruje***Authenticator***, ověří přijatá data
- pokud je vše v pořádku (a je vyžadována oboustranná autentizace), tak**server** připraví odpověď KRB-AP-REP, kterou odešle klientovi opět zabalenou pomocí aplikačních protokolů **[9]**
- odpověď je šifrována pomocí ***Session Key*** (klient-služba) a obsahuje ***časovou známku***, kterou poslal klient jako součást ***Authenticator***

Navázání session

- **klient** rozšifruje potvrzení a porovná časovou známku, pokud je vše v pořádku, došlo k úspěšné vzájemné autentizaci

Pozn.:Pokud chceme nalézt ještě detailnější popis, tak doporučuji prostudovat specifikaci RFC 4120.

Doplňující informace

Výše jsme si docela podrobně popsali proces autentizace, přesto může vzniknout řada otázek na určité praktické detaily, na které nám tento postup neodpoví. Většinu věcí můžeme odvodit z obecných informací, které jsme si uvedli dříve. Přesto si tu ukážeme praktický případ, na kterém si popíšeme speciální problémy.

Budeme uvažovat ***připojení klienta*** pomocí webového prohlížeče k aplikaci na***web serveru***, kde dochází k Single Sign-On přihlášení pomocí Kerberosu. Použití webové aplikace (protokolu http) slouží pouze k tomu, abychom mluvili konkrétně, obecně je situace obdobná u většiny ostatních protokolů.

Aby byla situace složitější, a mohli jsme si popsat speciální situace, tak budeme uvažovat následující scénář. ***Webový server*** je Apache na Linuxu a fyzicky běží v jiné síti než uživatelské stanice (nemůže komunikovat s DC). Adresa je www.firma.cz a je síťově dostupná klientovi. V rámci lokální domény firma.local jsme vytvořili účet, registrovali SPN HTTP/www.firma.cz@FIRMA.LOCAL (to může obsahovat libovolnou adresu z libovolné domény, za zavináčem je naše doména, kde jsme záznam registrovali) a vytvořili ***keytab soubor***, který jsme nahráli a nastavili na Apache.

Uživatel ***Petr Bouška*** má v doméně účet s UPN petr.bouska@jina-firma.local a***sAMAccountName*** firma\bouska. Je přihlášen na stanici, která je součástí domény a korektně autentizován pomocí Kerberosu.

- otevřeme stránku www.firma.cz
- dostaneme odpověď, že stránka vyžaduje autentizaci a podporuje vyjednáni metodou ***Negotiate***, klient volí ***Kerberos***
- klient potřebuje získat ***Service Ticket***, nijak neřeší, že adresa serveru je v jiné doméně než on, prostě se obrátí na svoje ***KDC*** (které nalezl při přihlášení dle postupu, který jsme popsali dříve)
- připraví žádost KRB-TGS-REQ, kam vloží autentizační údaje a data požadavku, tedy ***Realm*** (doménu KDC severu) FIRMA.LOCAL a ***Server Name***(Principal Name služby) HTTP/www.firma.cz
- doménový řadič hledá v rámci své domény SPN HTTP/www.firma.cz, tak nalezne účet, který jej má přiřazené (SPN by se mohlo nacházet i v rámci jiné důvěryhodné domény, to popíšeme později)
- tím má DC všechny údaje, aby sestavil ***Service Ticket*** (do něj vloží jméno uživatele), odešle jej klientovi
- klient odešle ***Service Ticket*** (jako součást HTTP hlavičky) na webový server, ten využije ***keytab soubor*** a rozšifruje zaslaná data
- tak server získá uživatelské jméno, což je ***Client Name:*** bouska a ***Client Realm:*** FIRMA.LOCAL

Z výše uvedeného plyne několik závěrů, které jsou důležité pro **praktické nasazování Kerberos SSO**.

- ***server nemusí komunikovat s DC***, stačí, když bude mít keytab soubor
- ***klient musí komunikovat s DC*** v průběhu autentizace (není to úplně pravda, komunikovat musí poprvé, pak si tikety i klíče na určitou dobu uloží do cache a může je opakovaně použít)

- ***DNS jméno služby může být z jiné domény*** než klient a buď je SPN registrováno v klientově doméně, nebo se může uplatnit Cross Realm autentizace (ověření v rámci jiné důvěryhodné domény)
- jako ***jméno uživatele se vždy bere sAMAccountName*** a DNS jméno domény, může se zapisovat ve tvaru bouska@FIRMA.LOCAL

Kerberos SSO mezi doménami

Výše jsme si popsali průběh Kerberos autentizace (SSO) v rámci jedné domény (Realmu). Výhoda protokolu Kerberos je, že umí navazovat ***vztahy mezi Realmy*** a provádět autentizaci Cross-Realm. V případě Microsoft domén to znamená, že když máme navázaný vztah důvěry mezi nějakými doménami, tak nám***automaticky funguje autentizace***. Takže uživatelský účet se může nacházet v jedné doméně a server (služba) v jiné, uživatel se přesto přihlásí pomocí SSO.

I když vše funguje automaticky, tak je dobré vědět, jakým způsobem autentizace probíhá. To jsem popsal v článku Cross Domain Kerberos Single Sign-On.

Troubleshooting Kerberos SSO

Pokud používáme Kerberos autentizaci, obzvláště ve speciálních (webových) aplikacích, tak určitě narazíme na různé problémy. Určitý způsob detekce a řešení některých problémů jsem popsal v článku Troubleshooting Kerberos Single Sign-On.Více na: http://www.zive.cz/clanky/kerberos-cast-3--princip-kerberos-autentizace/sc-3-a-175765/default.aspx