Metody a prostředky spojené s bezpečností dat a ochraně Microsoft SQL Serverů

Methods and Tools Related to Data Security and the Protection of Microsoft SQL Servers

Bc. Lukáš Pálka DiS.

Diplomová práce
2012

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2011/2012

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Lukáš PÁLKA, DiS.**
Osobní číslo:	**A10879**
Studijní program:	**N 3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Téma práce:	**Metody a prostředky spojené s bezpečností dat a ochraně Microsoft SQL Serverů**

Zásady pro vypracování:

1. **Popište serverové součásti a nástoje MS SQL Serveru pro verze Standard, Enterprice a Datacenter. Popište SQL komunikaci server/klient.**
2. **Popište prostředky zabezpečení MS SQL Serverů sloužících k ochraně a zabezpečení dat.**
3. **Popište a v praxi realizujte a porovnejte metody zajišťující autentizaci k výše uvedeným verzím MS SQL servrům. Popište systém autorizace MS SQL Serveru.**
4. **Popište a navrhněte HW a SW prostředky potřebné k zajištění zabezpečení serveru, ochrany dat a zálohování.**
5. **Popište a realizujte metody zálohování pomocí nástrojů MS Serverů a aspoň jednoho dalšího robustního řešení určeného k zálohování dat databází.**
6. **Popište jak všechny výše uvedené nástroje spolu kombinovat tak, aby byla zajištena v maximální míře bezpečnost MS SQL Serverů, ochrana serveru, ochrana dat a robustnost.**
7. **Navrhněte a realizujte v praxi nasazení MS SQL serveru do potencionálně nebezpečné zóny, jak chránit servery před útoky z venčí, jak chránit data a jak zabezpečit chod těchto serverů v demilitarizované zóně společně se servery uvnitř podnikové sítě.**

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **STANEK, William R. Mistrovství v Microsoft Windows Server 2008. První vydání. Computer Press, a.s., 2009. ISBN 978-80-251-2158-0.**
2. **GROFF, James R. a Paul N. WEINBERG. SQL Kompletní průvodce. První vydání. Brno: CP Books, a.s., 2005. ISBN 80-251-0369-2.**
3. **STANEK, William R. Active Directory: Kapesní rádce administrátora. První vydání. Brno: Computer Press, a.s., 2009. ISBN 978-80-251-2555-7.**
4. **VIEIRA, Robert. SQL Server 2000: Programujeme profesionálně. První vydání. Brno: Computer Press Brno, 2001. ISBN 80-7226-506-7.**
5. **STANEK, William R. Microsoft SQL Server 2005: Kapesní rádce administrátora. První vydání. Brno: Computer Press, a.s., 2006. ISBN 80-251-1211-X.**
6. **MICROSOFT CORPORATION. Microsoft SQL Server 2000: Administrace systému MCSE Training Kit. První vydání. Brno: Computer Press, 2001. ISBN 80-7226-526-1.**
7. **IBM CORPORATION. IBM Tivoli Storage Manager Version 6.2 information center [CD]. Armonk, NY 10504-1785, U.S.A.: IBM Director of Licensing, 2009, 2010 [cit. 12.1.2012].**
8. **MICROSOFT CORPORATION. Compare Microsoft SQL Server Editions [online]. 21. listopadu 2011 19:45:42 [cit. 2012-01-12]. Dostupné z: http://www.microsoft.com/sqlserver/en/us/product-info/compare.aspx**

Vedoucí diplomové práce: **Ing. Petr Šilhavý, Ph.D.**
Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce: **24. února 2012**

Termín odevzdání diplomové práce: **21. května 2012**

Ve Zlíně dne 24. února 2012

L.S.

prof. Ing. Vladimír Vašek, CSc.
děkan

doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

Diplomová práce se skládá z několika částí. První z nich seznamuje čtenáře s problematikou bezpečnosti dat databázového serveru společně s popisem řešení. Další část práce se zabývá softwarovými nástroji SQL Serveru pro maximální bezpečnost dat a to instalaci služeb, jako je nasazení vlastní certifikační autoritou PKI do sítě LAN, konfigurací nativní služby MS SQL serveru pro zálohování a instalaci a konfiguraci TSM, pro kompletní zálohu serveru a databází. Rovněž je zde matematicky spočítána bezpečnost síly hesla, je zde realizována bezpečnost dat pomocí konfigurací databázových rolí a provedeno šifrování uživatelských databází.

Poslední část práce se týká porovnání všech uvedených technologií, instalaci a konfiguraci všech služeb a je zde vysvětleno a detailně popsáno nasazení těchto technologií.

Klíčová slova: PKI, Tivoli Storage Manager, bezpečnost dat databázového serveru, metody autentikace, šifrování dat, bezpečnost komunikace.

ABSTRACT

This thesis consists of several parts. The first introduces a reader to an issue of data security, database server, together with a description of a solution. Another part deals with software tools of SQL Server for maximum security and installation services, such as CA's own deployment of PKI in LAN configurations, natively in MS SQL Server backup and TSM installation and configuration, for complete server backup and databases. There is also mathematically calculated password security forces, there is implemented security configuration data using database roles and performed an encryption of user databases.

The last part concerns a comparison of all these technologies, installation and configuration of all services. It is explained in details and described of technologies deployment.

Keywords: PKI, Tivoli Storage Manager server database data security, authentication methods, data encryption, security of communications.

Poděkování

Tato diplomová práce je výsledkem už osmileté praxe ve správě databázového serveru na Magistrátě města Přerova. Proto bych chtěl poděkovat této instituci, co se týká profesionálního růstu v oboru IT.

Rád bych touto cestou poděkoval i mému garantovi práce Ing. Petru Šilhavému, Ph. D.

Chtěl bych rovněž poděkovat mé přítelkyní a rodině, kteří mě při studiu velmi podporovali.

Na závěr pak musím vyjádřit svůj dík všem lidem v mém okolí, kteří mě podporovali a motivovali dál ve studiu a samozřejmě k vypracování této diplomové práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně …….

podpis diplomanta

OBSAH

ÚVOD

Cílem práce je prozkoumat, popsat a vyhodnotit metody zabezpečení Microsoft databázového serveru a operačního systému pomocí nasazení prostředků bezpečné komunikace, implementace bezpečnostních prvků operačního systému a databázového systému. Popsat metody autentikace a vyhodnotit bezpečnostní rizika a navrhnout řešení těchto problémů, zrealizovat a popsat nasazení databázového serveru v demilitarizované zóně a v neposlední řadě popsat a instalovat službu pro zálohování dat databázového serveru.

Úkolem je prozkoumat a popsat celou infrastrukturu zabývající se bezpečností databázové služby a dat samotných v MS SQL serveru.

Práce je rozdělena na oddělené části zabývající se konkrétní problematikou.

První kapitola této práce se zabývá teoretickou částí, kde popíši serverové součásti MS SQL serveru a rozdíly ve verzích a edicích, popíši komunikaci přenosu dat mezi databázovým serverem a klientem. V teoretické části rovněž popíši, jaké jsou prostředky zabezpečení MS SQL serveru k přístupům k datům v databázích, popíši metody autentikace a problematiku zálohování dat a v neposlední řadě i možnosti šifrování přenosu informací a použití certifikátu pro ověření pravosti v komunikaci.

Další kapitolou je praktická část, která se zabývá již samotnou instalací a implementací vlastní certifikační autoritou PKI, konfigurací nativní služby MS SQL serveru pro zálohování a instalaci a konfiguraci TSM, pro kompletní zálohu serveru a databází. V praktické části je matematicky spočítána bezpečnost síly hesla, je zde realizována bezpečnost dat pomocí konfigurací databázových rolí a provedeno šifrování uživatelských databází.

V závěru diplomové práce, navrhneme konkrétní řešení a dostaneme se ke kombinaci metod zajišťující bezpečnost SQL Serveru. V rámci celé infrastruktury je popsáno, jak použít softwarové i hardwarové nástroje, které nám zajistí v maximální míře bezpečnost služeb a dat uložených na serverech. Je zde popsáno, jak nasadit databázový server do demilitarizované zóny, jak nasadit databázový server do vnitřní sítě s možností autentizace přes domain controler. Je zde detailně popsáno oddělení databázového serveru společně s domain controlerem a realizace trustu mezi doménami s konfigurací

jednosměrné důvěry mezi doménami. Vše je detailně popsáno, jsou zde schémata zapojení, přesný popis komunikace a doporučení pro implementaci.

Diplomovou práci lze použít jako návod pro budování infrastruktury zabývající se databázovými servery v organizaci, ale také, jak začlenit službu SQL Serveru do stávajícího prostředí.

Zejména jsou rozebrány nejnovější technologie týkající se bezpečnosti ochrany dat služby MS SQL 2008 R2 a MS SQL 2012 v kombinaci s dalšími službami OS po stránce SW.

Rád bych zde zmínil i souvislost s mou bakalářskou prací týkající se zabezpečení vysoké dostupnosti MS SQL Serveru, kdy spojením bezpečnosti dat a dostupnosti služby získáváme komplexní řešení, jak provozovat databázový server i v největších podnicích, jak po stránce bezpečnosti, tak i dostupnosti.

I. TEORETICKÁ ČÁST

1 SERVEROVÉ SOUČÁSTI A NÁSTROJE MS SQL SERVERU

Než se detailně podíváme na problematiku bezpečnosti Microsoft SQL Serveru, popíši poslední dvě edice a také v jakých verzích byly tyto edice vydány. Popíši rozdíly mezi nimi, jaké obsahují nástroje spojené se službou SQL serveru a další součásti služby SQL.

Popíši zde i rozdíly v edici pro verzi SQL Serveru 2008 R2 a SQL Serveru 2012, to jsou dvě poslední edice SQL serveru od společnosti Microsoft.

Zaměřím se zejména na bezpečnost dat a služby SQL v popisu v čem se jaké edice liší.

Verze SQL serveru jsou následující:

- SQL Server 7,
- SQL Server 2000,
- SQL Server 2005,
- SQL Server 2008,
- SQL Server 2008 R2 (Datacenter, Enterprice, Standard, Web, Workgroup, Express),
- SQL Server 2012 (Enterprice, Business Intelligence , Standard, Express, Compact a Developer).

Edice SQL serveru 2008 R2 jsou následující: Datacenter, Enterprice, Standard, Web, Workgroup, Express.

Názvy edicí SQL serveru 2012 se od verze 2008 R2 liší v levnějších variantách. Známé edice jsou následující: Enterprice, Business Intelligence, Standard, Express, Compact a Developer, (Web).

1.1 SQL Server 2012

Jednotlivých edicí SQL Serveru 2012 je sedm. Web edice lze pořídit již jen v licenčním modelu SPLA (Service Provider License Agreement), tedy jen prostřednictvím poskytovatelů služeb, za což je považován hosting a outsourcing.

SQL Server 2012 bude v 2K/2012 roku dostupný ve třech hlavních edicích:

Enterprise – tato edice je nejvyšší edice od verze SQL 2012 a je primárně určena pro provoz Mission Critical aplikací s vysokou dostupností, vysokou mírou zabezpečení společně s nástroji pro neustálý dohled Business Intelligence. Jelikož verze Datacenter zanikla, přebírá tato edice jako nejvyšší všechny její nástroje, služby a prostředky.

Business Intelligence – je druhou nejvyšší edicí, která je naprosto nová, bohužel vybavená pro bezpečnost serveru velice chudě, protože přišla o zajímavou možnost, co se týká bezpečnosti. Edice kromě chybějících nástrojů dostala i nějaké to omezení. Jedna z těch významnějších služeb, které v edici BI nenajdeme, ohledně bezpečnosti, je rozšířené zabezpečení a rozšířená vysoká dostupnost. V této edici přicházíme o možnost auditování nad fyzickými daty v databázích a šifrování dat v databázích. Co se týká vysoké dostupnosti, nelze clustering provozovat mimo LAN síť a je omezen počet aktivních serverů v cluster na dva, stejně jako standard. Další zajímavou věcí, které v Business verzi nenajdeme, je snímkování dat (snapshot) a online obnovu databáze, která je dostupná, ještě před dokončením restoru.

Standard – Tato verze zůstala od původní verze SQL 2008 R2 nezměněna, obsahuje základní funkcionality, včetně některých služeb BI s omezením. V řadě věcí se však liší oproti vyšší verzi BI a tím je například zpráva upozornění a robustní zasílání zpráv a alertů.

1.1.1 Služby a vlastnosti edicí SQL Serveru 2012

Tabulka je ve velké míře upravena a je zde plno vlastních postřehů a zkušeností.

FUNKCE MS SQL 2012	ENTERPRISE	BUSINESS INTELLIGENCE	STANDARD
MAXIMÁLNÍ POČET JADER	NEOMEZENO	16 JADER A MAX 4 PATICE	16 JADER A MAX 4 PATICE
LICENCOVÁNÍ	NA JÁDRO	NA SERVER NEBO CAL CLIENT ACCESS LICENCE	NA JÁDRO, NA SERVER NEBO CAL CLIENT ACCESS LICENCE
PROGRAMOVATELNOST *(T-SQL, DATOVÉ TYPY, FILETABLE)*	NEOMEZENO	NEOMEZENO	NEOMEZENO
OVLADATELNOST *(SQL SERVER MANAGEMENT STUDIO, POLICY-BASED MANAGEMENT)*	NEOMEZENO	NEOMEZENO	NEOMEZENO

FUNKCE MS SQL 2012	ENTERPRISE	BUSINESS INTELLIGENCE	STANDARD
ZÁKLADNÍ VYSOKÁ DOSTUPNOST 2 UZLY FAILOVER CLUSTERINGU V LAN, POUZE PASIVNÍ MOD PŘEPÍNÁNÍ, VŽDY JE JEN JEDEN AKTIVNÍ A DRUHÝ ČEKÁ. PODPORA ZRCADLENÍ DATABÁZÍ.	NEOMEZENO	NEOMEZENO	NEOMEZENO
ZÁKLADNÍ FUNKCE BI *(REPORTING, ANALYTICS, MD SÉMANTICKÝ MODEL, DOLOVÁNÍ DAT)*	NEOMEZENO	NEOMEZENO	NEOMEZENO
ZÁKLADNÍ DATOVÉ INTEGRAČNÍ NÁSTROJE *(BUILT-IN DATA CONNECTORS, DESIGNER TRANSFORMS)*	NEOMEZENO	NEOMEZENO	NEOMEZENO
SELF-SERVICE BUSINESS INTELLIGENCE *(VAROVÁNÍ, ROZŠÍŘENÉ POHLEDY NA ZPRÁVY SERVERU, POWERPIVOT PRO SHAREPOINT SERVER)*	NEOMEZENO	NEOMEZENO	
DATABASE MIRRORING A REPLIKACE DAT	**ANO**	**ANO**	**ANO**
ROZŠÍŘENÉ FUNKCE BI (*POKROČILÉ ANALYTIKY A REPORTOVACÍ SLUŽBY, VERTIPAQ ™ IN-MEMORY ENGINE, ADVANCED DATA MINING – DOLOVÁNÍ DAT, TABULAR BI SEMANTIC MODEL)*	NEOMEZENO	NEOMEZENO	
ROBUSTNÍ ZPRÁVA DAT POMOCÍ NÁSTROJŮ MANAGEMENT STUDIA *(DATA QUALITY SERVICES, MASTER DATA SERVICES)*	NEOMEZENO	NEOMEZENO	

FUNKCE MS SQL 2012	ENTERPRISE	BUSINESS INTELLIGENCE	STANDARD
POKROČILÁ INTEGRACE DAT *(FUZZY PROPOJENÍ A VYHLEDÁVÁNÍ, ZMĚNA A SBĚR DAT)*	NEOMEZENO		
ROZŠÍŘENÉ ZABEZPEČENÍ *(SQL SERVER AUDIT, TRANSPARENTNÍ ŠIFROVÁNÍ DAT)*	NEOMEZENO		
DATOVÉ SKLADY *(COLUMNSTORE INDEX, KOMPRESE DAT A DATABÁZÍ, ROZDĚLENÍ DATABÁZÍ NA PÁRTIŠNY)*	NEOMEZENO		
ROZŠÍŘENÉ HIGH AVAILABILITY *(NEOMEZEN POČET NODU, AKTIVNÍ CLUSTERING – VÍCE SERVERŮ ZPRACOVÁVÁ DATA, MULTI-SITE, GEOCLUSTERING – ZAPOJENÍ NOODU I MIMO JEDNU LOKALITU (SÍŤ), A HI EVAILEBILITY CLUSTERING PŘES VEŘEJNÉ SITĚ)*	NEOMEZENO		

Tabulka 1 – Rozdíly v edicích SQL Serveru 2012- 1 [4]

Další rozdíly v bezpečnosti dat a zálohovaní ve verzích popíši dle své praxe a poznatku.

FUNKCE MS SQL 2012	ENTERPRISE	BUSINESS INTELLIGENCE	STANDARD
DATABASE SNAPSHOT – SNÍMKOVÁNÍ DAT	ANO		
ROZHRANÍ PRO NAPOJENÍ NA ORACLE	ANO		
MDS – SLUŽBA ZAJIŠŤUJÍCÍ JEDINÝ AUTORITATIVNÍ ZDROJ INFORMACÍ PRO APLIKACE A DATABÁZE	ANO		

FUNKCE MS SQL 2012	ENTERPRISE	BUSINESS INTELLIGENCE	STANDARD
PŘEDPOVÍDÁNÍ A CACHING DAT – WAREHOUSE	ANO		
KEY PERFORMANCE INDICATORS	ANO	ANO	
PODPORA HYPERVIZORU A MIGRACE ONLINE POMOCÍ HYPER-V, PODPORA CLUSTERINGU PŘI VIZUALIZACI SQL	ANO	ANO	ANO
PODPORA PŘIDÁVÁNÍ PAMĚTI A PROCESORŮ ZA CHODU	NEOMEZENO		
PODPORA ZÁSAD PRO HESLA V SYSTÉMU WINDOWS	ANO	ANO	ANO

Tabulka 2 - Rozdíly v edicích SQL Serveru 2012- 2

1.2 SQL Server 2008 R2

Jednotlivých edicí SQL Serveru 2008 R2 je šest. SQL Server 2008 R2 je dostupný ve třech hlavních edicích:

Datacenter a Enterprise – tyto edice jsou nejvyšší edicí ve verzi SQL 2008 a jsou primárně určeny pro provoz těch největších databází s plně otevřenými možnostmi všech modulů. Najít v čem se liší verze datacenter a enterprice je velice obtížné a obecně lze chápat datacenter edici shodnou s enterprice s tím, že jsou rozšířeny smluvní partnerské dohody gold partnerů firmy Microsoft, kde jsme schopni okamžitě dostat k řešení našeho problému certifikovaného odborníka. Jedná se tedy spíše o servisní podporu, než rozdíl v edici z hlediska funkcionality. Verze datacenter lze koupit pouze v rámci smlouvy Esurance s MS.

Standard – standardní verze obsahuje základní funkcionality včetně služeb HI evailebility s omezením na dva nody čistě v režimu pasive nod. V této verzi nelze provozovat řadu funkcionalit těch vyšších verzí jako například Analysis Services, rychlou obnovu záloh a jsou zde omezeny i reportovací chyby pro administrátory. Při obnově záloh ve verzi standard je nutné počkat na kompletní obnovu všech částí databáze včetně obnovy transakčních logů do full backupu.

1.2.1 Služby a vlastnosti edicí SQL Serveru 2008 R2

Tabulka je ve velké míře upravena a je zde plno vlastních postřehů a zkušeností.

FUNKCE MS SQL 2008 R2	DATACENTER	ENTERPRISE	STANDARD
MAXIMÁLNÍ POČET PROCESORŮ	NEOMEZENO	**8**	**4**
LICENCOVÁNÍ	**NA PROCESOR NEBO CAL** **CLIENT ACCESS LICENCE**	**NA PROCESOR NEBO CAL** **CLIENT ACCESS LICENCE**	**NA PROCESOR NEBO CAL** **CLIENT ACCESS LICENCE**
MAXIMÁLNÍ VYUŽITÁ PAMĚŤ	NEOMEZENO	**2 TB**	**64 GB**
ZRCADLENÍ ZÁLOH, RYCHLÁ OBNOVA DATABÁZÍ	NEOMEZENO	NEOMEZENO	
DATABASE SNAPSHOT – SNÍMKOVÁNÍ DAT	NEOMEZENO	NEOMEZENO	
ONLINE INDEXOVÁNÍ, PARALELNÍ INDEXOVÁNÍ	NEOMEZENO	NEOMEZENO	
PODPORA HYPERVIZORU A MIGRACE ONLINE POMOCI HYPER-V, PODPORA CLUSTERINGU PŘI VIZUALIZACI SQL	**ANO**	**ANO**	**ANO**
ZÁKLADNÍ ŠIFROVÁNÍ DAT A SPRÁVA KLÍČŮ	NEOMEZENO	NEOMEZENO	NEOMEZENO
ROZŠÍŘENÉ ZABEZPEČENÍ *(SQL SERVER AUDIT, TRANSPARENTNÍ ŠIFROVÁNÍ DAT)*	NEOMEZENO	NEOMEZENO	
INTEGROVANÉ OVĚŘOVÁNÍ SYSTÉMU WINDOWS (VČETNĚ TECHNOLOGIE KERBEROS)	NEOMEZENO	NEOMEZENO	NEOMEZENO
ROZHRANÍ PRO NAPOJENÍ NA ORACLE	NEOMEZENO	NEOMEZENO	

FUNKCE MS SQL 2008 R2	DATACENTER	ENTERPRISE	STANDARD
PODPORA PŘIDÁVÁNÍ PAMĚTI A PROCESORŮ ZA CHODU	NEOMEZENO	NEOMEZENO	
PODPORA ZÁSAD PRO HESLA V SYSTÉMU WINDOWS	NEOMEZENO	NEOMEZENO	NEOMEZENO
SELF-SERVICE BUSINESS INTELLIGENCE *(VAROVÁNÍ, ROZŠÍŘENÉ POHLEDY NA ZPRÁVY SERVERU, POWERPIVOT PRO SHAREPOINT SERVER)*	NEOMEZENO	NEOMEZENO	
DATOVÉ SKLADY *(*PARALELNÍ ZPRACOVÁNÍ DOTAZŮ U DĚLENÝCH TABULEK A INDEXŮ*, KOMPRESE DAT,* DĚLENÉ KRYCHLE, ZACHYTÁVÁNÍ ZMĚN DAT, PROAKTIVNÍ UKLÁDÁNÍ DO MEZIPAMĚTI*)*	NEOMEZENO	NEOMEZENO	
ANALYSIS SERVICES – ANALYTICKÉ FUNKCE PRO ANALÝZU ÚČTŮ, PERSPEKTIVY, ŠKÁLOVATELNÉ SDÍLENÉ DATABÁZE, DIMENZE ZPĚTNÉHO ZÁPISU A DALŠÍ	NEOMEZENO	NEOMEZENO	
DOLOVÁNÍ DAT – PARALELNÍ ZPRACOVÁNÍ MODELŮ, PROGNÓZY SEKVENCÍ, KONFIGURACE A LADĚNÍ V ALGORITMECH DOLOVÁNÍ DAT	NEOMEZENO	NEOMEZENO	
REPORTOVACÍ NÁSTROJE	NEOMEZENO	NEOMEZENO	NEOMEZENO
MAPY A MAPOVÉ VRSTVY (OD VERZE SQL S. 2008)	NEOMEZENO	NEOMEZENO	NEOMEZENO

FUNKCE MS SQL 2008 R2	DATACENTER	ENTERPRISE	STANDARD
EXPORTY DAT DO FORMÁTŮ (XLS, WORD, PDF, PODPORA STAND. OBRÁZKŮ)	NEOMEZENO	NEOMEZENO	NEOMEZENO
BEZPEČNOST PŘÍSTUPŮ POMOCÍ SPRÁVY ROLÍ	NEOMEZENO	NEOMEZENO	NEOMEZENO
INTEGRACE SE SLUŽBOU SHAREPOINT	NEOMEZENO	NEOMEZENO	NEOMEZENO
SLUŽBA MASTER DATA SERVICES (SLUŽBA PRO ZAJIŠTĚNÍ TRVALE INTEGRITY INFORMACÍ A KONZISTENCI DAT V APLIKACÍCH).	NEOMEZENO	NEOMEZENO	

Tabulka 3 - Rozdíly v edicích SQL Serveru 2008 R2 [8]

1.3 Rozdíl MS SQL 2012 a MS SQL 2008 R2

MS SQL 2012	MS SQL 2008 R2
COLUMNAR INDEX - **INDEX ULOŽENÝ VE SLOUPCÍCH**	**ZABĚHLÝ DATABÁZOVÝ SYSTÉM S SP**
CLOUD PODPORA – MINIMUM SPRÁVY PRO ADMINA, ZARUČEN PROVOZ BEZ VÝPADKU, SNADNÁ MIGRACE VŠECH DAT A DATABÁZÍ, PŘEPRACOVANÁ ŠKÁLOVATELNOST	**OBSÁHLA KNOWLEDGE BÁZE**
CONTAINED DIABASE – **ZPŮSOB AUTENTIZACE BEZ NUTNOSTI MIT LOGIN NA SERVERU. DÍKY CD ZÍSKÁME OBROVSKOU SVOBODU MIGRACÍ DAT A MOŽNOST PŘESOUVAT DATA MEZI JEDNOTLIVÝMI PROSTŘEDÍMI**	**POŘIZOVACÍ NÁKLADY JSOU VÝRAZNĚ NIŽŠÍ – LICENCOVÁNÍ NA PROCESOR A NE NA JÁDRO**
PŘEPRACOVÁN *POWERPIVOT A ZMĚNĚN NA NÁZEV* POWERVIEW BĚŽÍCÍ POD **BUSINESS INTELLIGENCE SEMANTIC MODEL (BISM). NOVÝ ROBUSTNÍ ONLINE REPORTOVACÍ NÁSTROJ PODPORUJÍCÍ WEBOVÉ TECHNOLOGIE** **(PRO REPORTY NUTNÝ SHAREPOINT A DATOVÁ KRYCHLE MUSÍ BÝT V NOVÉM FORMÁTU, TZN. PŘEVODY DAT)**	**OLAP KRYCHLE A BĚH POWERPIVOTU JE TÉMĚŘ NEPOUŽITELNÝ, TĚŽKOPÁDNÝ A POMALÝ**
MODELOVÁNÍ OLAP KRYCHLÍ PŘEPRACOVÁNO NA NOVÝ MODEL. (PROZATÍM ŠPATNÁ PODPORA KONVERZE)	**OLAP MODELOVÁNÍ KRYCHLÍ STÁRNE A JIŽ NENÍ DOKONALÝ PRO VYTVÁŘENÍ ANALYTICKÝCH DATABÁZÍ**

Tabulka 4 – Provonání edicí SQL Serveru 2012 a 2008 R2

2 SQL KOMUNIKACE SERVER / KLIENT

Pro přístup uživatelé k SQL Serveru prostřednictvím klientských aplikací podporuje SQL Serveru dvě metody přístupu k datům uložených na serveru.

Prvním typem jsou aplikace na principu komunikace založené v režimu relační databáze. Tyto aplikace jsou nejběžnějším typem klientských aplikací používaným v dvouvrstvých prostředích klient/server. Tyto klientské aplikace komunikují pomocí jazyka Transact-SQL vůči relačnímu databázovému stroji a výsledky dotazů jsou posílány zpět po-té, co je relační databázový server zpracuje a odešle zpět ve stejném formátu jazyka T-SQL. Výpočet dotazu tedy provádí SQL Server a posílá se zpět pouze odpověď a lze práci chápat, jako získávání informací z množiny dat pomocí dotazu, kdy získáme jen ty údaje, které požadujeme.

Druhým typem internetové aplikace, jsou takové aplikace, které jsou pracují s rozhraním platformy Microsoft.NET. Takové klientské aplikace posílají příkazy vůči databázovému relačnímu stroji v jazyce Transact-SQL a dostávají jako odpovědi dokumenty ve formátu XML.

Obě tyto metody klientských aplikací se připojují k SQL serveru různými způsoby.

Aplikační programová rozhraní relační databáze [2]

Relační databázové aplikace přistupují k SQL serveru prostřednictvím databázového aplikačního programového rozhraní API. Databázové rozhraní API definuje způsob, jakým má být napsána aplikace, aby se mohla připojit k instanci SQL Serveru a předávat příkazy databází SQL Serveru. Služba SQL serveru poskytuje nativní podporu pro obě hlavní třídy databázových aplikačních programových rozhraní, kterými jsou OLE DB a ODBC.

Rozhraní OLE DB je aplikační programové rozhraní, které umožňuje aplikacím typu COM zpracovávat data z datových zdrojů OLE DB. SQL Server obsahuje nativního poskytovatele dat OLE DB. Poskytovatel dat OLE DB je součást typu COM, která přijímá volání rozhraní OLE DB API a dělá vše potřebné pro zpracování tohoto požadavku na zdroji dat. Poskytovatel podporuje aplikace napsané pomocí rozhraní OLE DB nebo pomocí jiných rozhraní API využívajících OLE DB, například rozhraní ADO.

ODBC je rozhraní na úrovní volání Call-Level Interface (CLI), které umožňuje aplikacím napsaným v jazycích C a C++ přístup k datům ze zdrojů dat ODBC. SQL Server obsahuje nativní ovladač ODBC. Ovladač ODBC je knihovna DLL, která přijímá volání funkcí

rozhraní ODBC API a dělá vše potřebné pro zpracování takových požadavků na zdroj dat. Ovladač podporuje aplikace nebo součásti napsané pomocí rozhraní ODBC, nebo pomocí jiných rozhraní API využívajících ODBC, jakými jsou například Data Access Objects (DAO), Remote Data Objects (RDO) a databázové třídy Microsoft Foundation Classes (MFC). Rozhraní DAO a RDO se obecně nahrazují rozhraním ADO.

SQL Server rovněž podporuje knihovnu DB-Library (pouze pro zpětnou kompatibilitu) a rozhraní Embeded SQL.

Knihovny Net-Library [2]

Poskytovatel OLE DB nebo ovladač ODBC používá klientskou knihovnu Net-Library ke komunikaci se serverovou knihovnou Net-Library na instanci SQL Serveru. Komunikace může probíhat vrámci jednoho počítače, nebo komunikace server-client. Knihovny Net-Library obalují požadavky mezi klientskými počítači a servery, aby mohly být přenášeny síťovými protokoly nižších vrstev. Komunikace mezi klientskými a serverovými knihovnami Net-Library může být šifrována pomocí protokolu Secure Sockets Layer (SSL). Klienti a servery SQL mohou být konfigurovány pro použití některé nebo všech knihoven Net-Library.

Shared memory – používá se k připojení SQL serveru na jednom počítači jako je klient. Princip zdílené paměti stroje.

Named Pipes – knihovna pro připojení k SQL serveru pomocí pojmenovaných rour. Roura je mechanismus systému souborů používaný pro komunikaci mezi procesy. Toto je jeden z výchozích protokolů pro SQL server.

TCP/IP Soskets – používá se pro připojení k SQL Serveru pomocí protokolu TCP/IP. Nejpoužívanější metoda spojení v dnešní době.

NWLink IPX/SPX – použití v prostředí novell nad protokolem IPX/SPX.

VIA GigaNET SAN – používá se pro podporu vysokorychlostní technologie SAN v sítích se serverovými farmami GigaNet cLAN.

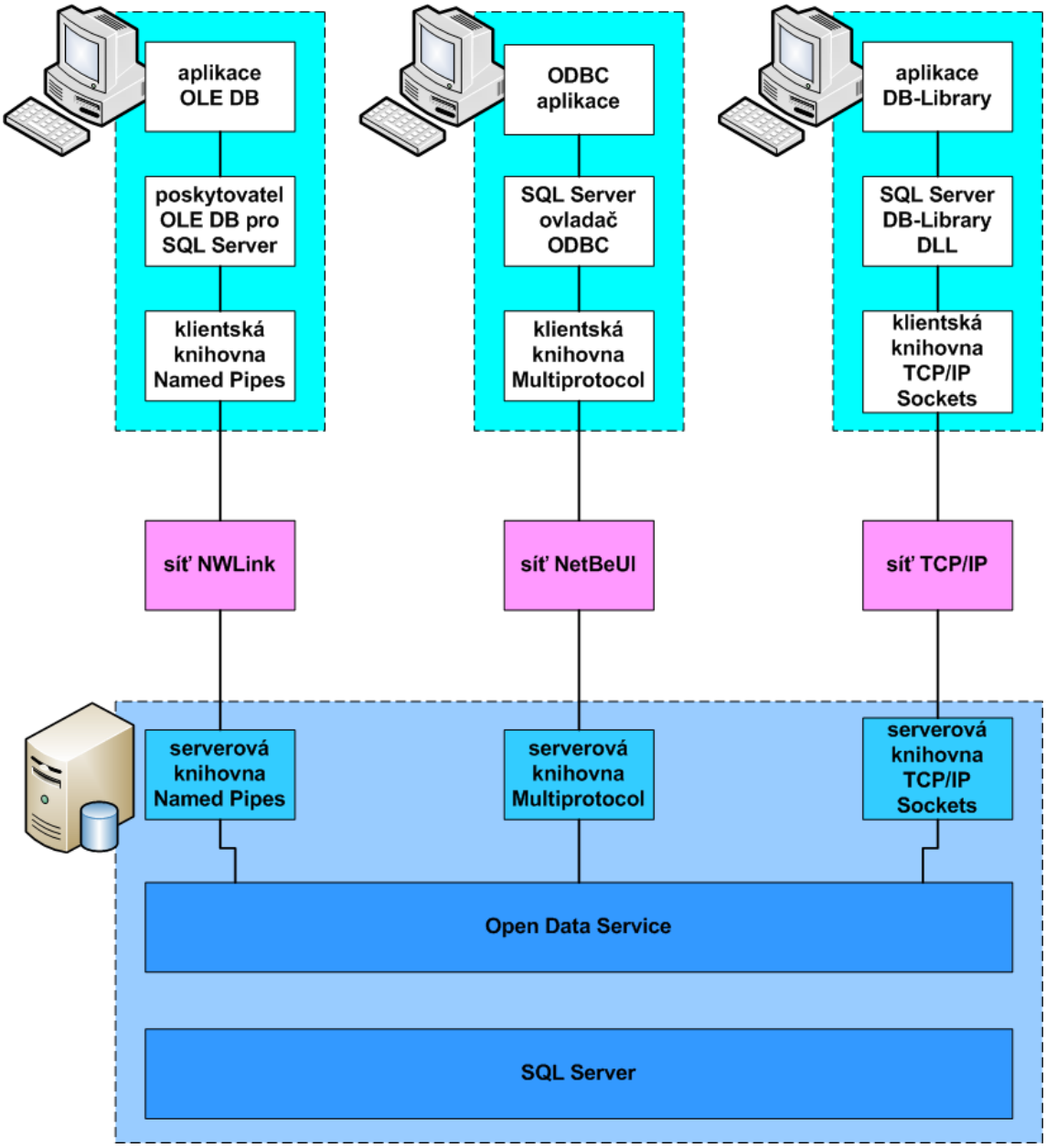
Multiprotokol – podporuje všechny dostupné komunikační metody mezi servery prostřednictvím volání Windows NT RPC po libovolném síťovém protokolu. Ve starších verzích SQL Serveru byla tato knihovna vyžadována, mělo-li být umožněno šifrování a podporováno ověření systému Windows. V současné době, je zde tento protokol pouze pro zpětnou kompatibilitu.

AppkeTalk ADSP – používá se v prostředí Apple a Macintosh, kdy je umožněno Apple zařízením spojit se s MS SQL Serverem.

Banyan VINES – historické rozhraní, dnes již nenajdeme aplikaci, ani dokumentaci k tomuhle rozhraní.

Komunikace klient/server [2]

Na obrázku je znázorněna zjednodušená verze klientských komunikačních součástí v případě, že klientská aplikace a instance SQL Serveru je instalována na různých počítačích.



Obrázek 1 - Komunikace Klient/Server

Open Data Services [2]

Serverové knihovny Net-Library komunikují v relačním databázovém stroji s vrstvou Open Data Services, která tvoří rozhraní mezi relačním databázovým strojem a serverovými knihovnami Net-Library. Rozhraní Open Data Services transformuje pakety přijaté od serverových knihoven Net-Library na události a ty pak předává příslušné části relačního databázového stroje. Relační databázový stroj pak používá rozhraní Open Data Services pro odesílání odpovědí zpět klientům SQL Serveru prostřednictvím serverových knihoven Net-Library.

Přístup Internetové aplikace k SQL Serveru [2]

Internetové aplikace přistupují k SQL Serveru pomocí prostředků kořene virtuálního serveru definovaného ve službě IIS, který odkazuje na instanci SQL Serveru. SQL Server obsahuje knihovnu ISAPI DLL (sqlisapi.dll), která takový přístup umožňuje. Takové aplikace mohou používat Uniform Resource Locator (URL), rozhraní ADO API nebo OLE DB API a provádět dotazy XPath nebo příkazy jazyka Transact-SQL.

Když služba IIS příjme dotaz XPath nebo příkaz jazyka Transact-SQL, zavede server IIS do paměti knihovnu ISAPI DLL. Knihovna ISAPI DLL pro připojení k instanci SQL Serveru uvedené ve virtuálním kořenu používá poskytovatele OLE DB pro SQL Server (SQLOLEDB) a předává SQL Serveru dotazy XPath nebo příkazy jazyka T-SQL.

3 PROSTŘEDKY ZABEZPEČENÍ MS SQL SERVERU SLOUŽÍCÍCH K OCHRANĚ A ZABEZPEČENÍ DAT

Prostředky zajišťující bezpečnost SQL serveru jsou takové prostředky, které jsou integrovány do služby SQL serveru. Tyto mechanizmy a zabezpečení na této úrovni obsahují definici rolí, správu uživatelů a přístupu k daným tabulkám a pohledům, obecně lze říci datům.

Data uložená v SQL serveru publikována vůči stanicím je třeba chránit. Zpřístupnit či nezpřístupnit data klientovi se děje pomocí mechanizmu zabezpečení SQL serveru, kdy je vyhodnoceno přihlašovací jméno, nastavení přístupových oprávnění a přiřazení rolí. Přiřazená oprávnění a role určují, kam smí klient přistoupit, neboli s jakými daty může pracovat. V neposlední řadě je i vyhodnoceno oprávnění nad danou tabulkou, kde jsou vyhodnoceny práva pro čtení, pro zápis, pro změnu dat a další.

V SQL serveru jsou veškeré objekty databáze definovány ve schématech. V každém schématu by měla být přiřazena role, nikoli uživatel, kde jsou specifikovány práva na konkrétní tabulku v dané databázi. Role je skupina uživatelů s definicí práv na tabulku či více tabulek. V případě uživatele lze rovněž přiřadit oprávnění k tabulce, ale jakákoliv změna schématu je následně úzce svázána s objektem uživatele, což v budoucnu přinese řadu komplikací. Proto doporučuji, nikdy nepoužívat oprávnění definované na uživatele, ale vždy na roli databáze.

Bezpečnostní kontext (Security Principal) [1]

Jedná se o entitu, která vnáší požadavek vůči serveru, databázi nebo schématu. Bezpečnostní kontext vždy obsahuje SID – Security identifikátor.

Bezpečnostní kontext dělíme na tři úrovně:

- Windows (přihlašovací jméno domény Windows, lokální přihlášení Windows, Skupiny Windows),
- SQL server (Role SQL serveru, přihlašovací jméno SQL serveru),
- Databáze (uživatel databáze, mapovaná databáze, role aplikace a role databáze).

Úroveň, na niž je bezpečnostní kontext definován, určuje definice práv a přístupů. Některé bezpečnostní kontexty jako databázové role a role aplikací obsahují další bezpečnostní kontexty. Tyto bezpečnostní kontexty se nazývají kolekce.

Každý uživatel databáze je automaticky členem role Public. Pokud uživateli není přiřazena i další role, dědí oprávnění přiřazená této roli Public pro každou databázi.

Autentizační režimy SQL Serveru

Uživatel se ověřuje vůči SQL serveru, nebo Active Directory který autentizuje uživatele SQL serveru, který je jeho členem. Podrobnější informace jsou v následující kapitole Metody autentizace MS SQL Serveru. Výhodou AD autentizace je centrální zpráva účtu a bezpečnost na úrovni jedné služby.

3.1 Defaultní oprávnění SQL serveru

Výchozí oprávnění po instalaci SQL serveru a výchozí režimy autentizace nesou řadu přidělených systémových či jiných práv na data sql serveru, které ne vždy je vhodné zachovat.

Oprávnění doménové skupiny administrátors – tato skupina je zařazena mezi sql skupinu sysadmin, která má plnou kontrolu nad všemi databázemi SQL serveru. Toto oprávnění doporučuji ponechat vzhledem k tomu, že se nejedná o běžnou skupinu, kde jsou zařazeni uživatelé.

Oprávnění lokálního účtu Administrátor, SYSTEM a network service – jedná se o vestavěné lokální účty, které nesou opět oprávnění sysadmina. Účet administrátor doporučuji z bezpečnostních důvodů odebrat z této skupiny. Před tím, než tento účet odebereme, provedeme změnu spouštění služby pod účtem SYSTÉM. Účet administrátor je nejčastěji vyhledávaným účtem, na který je prováděn útok a odebráním z role sysadmina a spouštěním služby pod SYSTÉM účtem OS získáme robustnější zabezpečení. Dalším krokem je nastavit auditing na účet Administrátor a zamykání účtu po několika pokusech o přihlášení. Účet network services ponecháme v sysadminech, je to účet, přes který komunikují reportovací služby serveru a v případě clusteringu jsou na tento účet pověšeny komunikace mezi jedním a druhým SQL serverem.

Oprávnění SA – jedná se o systémový účet SQL serveru, který je obecně znám útočníkům. Z povahy věci, je to opět jeden z účtů, kterému přiřadíme silné heslo. Pokud nad SQL serverem všechny plánované joby běží pod účtem SQL agenta a účet agenta běží pod jiným účtem, než je SA, účet samotný zablokujeme. Pokud by však z historických důvodů byla na našem serveru databáze v režimu databáze level SQL 7.0, musíme tento účet nechat povolen.

Obecně nad sysmadnin účty lze říci, že se jedná o bezpečnostní incident. Všechny tyto účty v ideálním případě zakážeme a nahradíme je účty jinými. Nastavíme nad účty logování a právo sysadmin přidělíme jen jednomu účtu a to správci databázového serveru, případně doménovému účtu. Ani pro plné záloh všech databází, nepotřebujeme použít sysadmin účty, k takovému procesu stačí práva backup operátora, která najdeme pod systémovými účty sql serveru.

Oprávnění QUEST – účet host je jeden z defaultních účtu, který je nadefinován i v model databází. Tzn. jakákoliv uživatelská databáze, bude obsahovat právo přístupu pro tohoto uživatele. Jelikož nemůžeme účet hosta zrušit, protože pod tímto oprávněním přistupuje většina uživatelů do databáze tempdb a master, účet předefinujeme v databázi model. Doporučuji odebrat nad databází model oprávnění přístupu guest a tím oprávnění do uživatelských databází nebude obsahovat tento přístup. Účet host je členem role public a proto dědí veškerá oprávnění.

Účet je vhodné používat tehdy, pokud uživatel nemá k uživatelské databázi nadefinován přístup, ale má vytvořen účet v AD nebo v SQL serveru.

Uživatel DBO [1]

Vlastník databáze neboli dbo je zvláštní typ databázového uživatele a jsou mu přidělena zvláštní privilegia. Uživatel, který vytvořil databázi, je vlastníkem databáze. Uživateli dbo jsou implicitně přidělena všechna oprávnění v databázi a může udělovat oprávnění i dalším uživatelům. Členové role sysadmin jsou automaticky mapováni na tohoto uživatele, z toho vyplívá, že všechny oprávnění sysadmin jsou shodné s oprávněním dbo.

Účet dbo ve skutečnosti zvláštní účet bez přihlašovacího jména. K serveru se nemůžeme přihlásit jako dbo účet, ale můžeme se stát osobou, která vytvořila databázi nebo její objekty.

3.2 Oprávnění a role SQL serveru

Oprávnění určuje, kam daný účet smí přistoupit a jaké operace nad daty smí vykonávat. Oprávnění se uděluje podle přihlašovacího jména, členství ve skupině a ROLI SQL Serveru.

ROLE SQL serveru jsou předdefinované skupiny pro stanovenou činnost nad daty daným uživatelem. Zařazením uživatele do takové role smí pak daný účet vykonávat konkrétní činnost.

Role SQL severu respektive role, které mají vliv na data SQL serveru, jsou dvě skupiny – Serverové role a Databázové role (SQL role).

3.2.1 Serverové role

Přiřazením serverové role přiřadíme oprávnění danému účtu ke správě serveru. V okamžiku přiřazení uživatele do této skupiny, smí uživatel plně provádět ty akce, které jsou příslušné k dané skupině. Serverové role se přiřazují na úrovni serveru a práva dané role se uplatní na celý server jako takový a ne jenom v rámci služby SQL serveru.

Serverová oprávnění jsou následující:

Sysadmin – nejvyšší oprávnění SQL serveru. Pod tímto účtem můžeme provádět všechny operace SQL serveru, měnit a vytvářet všechny komponenty i definovat další oprávnění. Máme rovněž přístup ke všem databázím SQL serveru. Proto tuhle roli doporučuji přiřadit jen jednomu účtu SQL serveru a to doménovému administrátorovi.

Setupadmin – role určena pro správu serverů, které komunikují s SQL serverem. Role slouží ke správě připojených serverů, správě připojení a řízení spouštěcích algoritmů a procesů.

Serveradmin – role pro čistě správu daného serveru. Pro úpravu HW zařízení, restart serverů a správy služeb.

Securityadmin – tato role umožňuje definovat nová, či měnit stávající oprávnění uživatelům sql serveru. Obecně lze říci, že to je administrátor starající se o účty serveru. Rovněž jde s touto rolí číst eventy OS.

Processadmin – role pro správu procesů serveru. S touto rolí lze zastavovat služby a znovu je spouštět.

Diskadmin – role pro správu disků a to jak systémového oddílu, tak i například externích diskových polí přiřazení serveru.

DBreator – role pro úplnou správu uživatelských databází. S touto rolí lze vytvářet, modifikovat, detachovat a obnovovat databáze. Role je úzce svázána s rolí diskadmin,

jelikož s touto rolí má uživatel právo vytvářet databáze na oddílech serveru pro SQL serveru.

Bulkadmin – role s definicí práv pro hromadné vkládání dat. Role používá k tomuto účelu příkaz BULK INSERT.

3.2.2 Role SQL serveru

Tyto role slouží k přidělení oprávnění na úrovni databázového serveru. Známe tři úrovně databázových rolí – standardní role, aplikační role a databázové role.

Standardní role je typická skupina uživatelů, kde dané skupině přiřazujeme určitá oprávnění. Do této skupiny pak zařazujeme uživatele, kteří dědí oprávnění díky této roli.

Aplikační role je z povahy věci chápána jako třívrstvá architektura přístupu k datům. Uživatel přistupuje k databázi přes aplikační můstek a ta mu předává role, potřebné pro dané operace.

3.2.3 Databázové role

Mezi předdefinované skupiny rolí v SQL serveru se řadí několik rolí s danými oprávněními.

Než vyjmenuji všechny role SQL serveru, upřesním, že definice práv se děje vždy nad danou databází. Tzn. pokud má mít uživatel práva na tabulky v databázi, je vždy oprávnění definováno již v dané databázi. Neplatí to ale pro účet samotný, ten je definován v master databázi. Pokud tedy přenášíme databázi z jednoho místa na druhý, stačí provést full backup a restore databáze a nemusíme dále řešit definici práv.

Role Public – výchozí role s defaultním přístupem k dané databázi. Pokud uživateli přidělíme práva k dané databázi, uživatel získá oprávnění public. Tato role nese jen minimum oprávnění, jako například ověření connect stringu při přístupu.

Role DB_accessadmin – role pro správu účtu k uživatelské databázi.

Role DB_backupoperator – role pro zálohování všech i systémových databází. Restore databáze není povoleno.

Role DB_Datareader – role s právy přístupu ke všem databázím s právem pro čtení. Přístup je tedy ke všem tabulkám i pohledům.

Role DB_Datawriter – oprávnění jako datareader s právy DELETE, UPDATE a INSERT

Role DB_Ddladmin – právo skupiny vykonávat příkazy DDL

Role DB_Denydatareader – role, kterou lze chápat jako filtr nad danou tabulkou. Uživatel má nadefinován přístup k databázi, ale chceme omezit přístup pro danou tabulku.

Role DB_Denydatawriter - oprávnění jako Denydatareader s filtrem práv pro příkazy DELETE, UPDATE a INSERT.

Role DB_securityadmin – role je určena uživatelům, kteří potřebují spravovat vlastnictví objektů, zprávu rolí a oprávnění.

Role DB_owner – nejvyšší role co se týká přístupu k SQL serveru. Typickým uživatelem této role je účet SA. Po instalaci SQL serveru musíme předefinovat heslo účtu SA, případně účet SA zrušit. Tyto kroky byly popsány v bezpečnosti dat SQL serveru. Role s těmito právy smí cokoliv na úrovni databázového serveru a člen má přístup ke všem tabulkám, nastavení služby, obecně ke všemu co je v SQL serveru.

3.2.4 Konfigurace rolí databáze a uživatele

Konfiguraci rolí databáze či uživatele provádíme v managament studiu. Jsou dvě cesty, jak oprávnění nastavit. Jedna je z pohledu databáze, druhá z pohledu uživatele. Obě metody jsou podstatné, protože ne vždy víme, kteří uživatele mají oprávnění nad db, případně naopak, kdy potřebujeme konkrétnímu účtu nastavit danou roli.

Oprávnění nad DB – nad vlastnosti dané databáze v záložce Properties máme možnost na úrovni Permissions nastavit tyto oprávnění.

- Na záložce Permissions lze zobrazit aktuální oprávnění k objektům a umožňuje jejich správu.

Oprávnění nad User – nad vlastností daného uživatele jsou dvě záložky pro definici rolí a oprávnění.

- Záložka Server Roles umožňuje přidat nebo odebrat serverové role.
- Databáze Access obsahuje nastavení pro přiřazení databázových rolí uživateli.

4 METODY AUTENTIZACE MS SQL SERVERU

Autentizace uživatelů do systému je jedna ze základních prvků ochrany sítí a systémů a také samotných uživatelských či jiných dat.

Autentizace se nejčastěji provádí dvojicí údajů: **uživatelské jméno** a **heslo**. Mezi dnes již další možnosti autentizace řadíme hardwarové metody identifikace uživatele a to pomocí tokenů, čipovou kartou, případně pomocí biometriky (otisk prstu, scan sítnice).

Autentizace MS SQL Serveru

Princip autentizace vůči Microsoft SQL Serveru je již od prvních verzi SQL serveru velmi podobný, až do nejnovější dostupné verze SQL Serveru 2008 R2, případně SQL 2012 RC0.

Systém autentizace používá dvě vestavěné metody pro ověření uživatele.

Jak zabezpečit server, jaká autentikace k datům je vhodná vzhledem k bezpečnosti dat uložených v databázovém serveru a jaké možnosti nastavení jsou, popíši v následující kapitole.

Z pohledu zabezpečení dat a přihlášení uživatelů chápeme proces přihlášeni ve dvou úrovních.

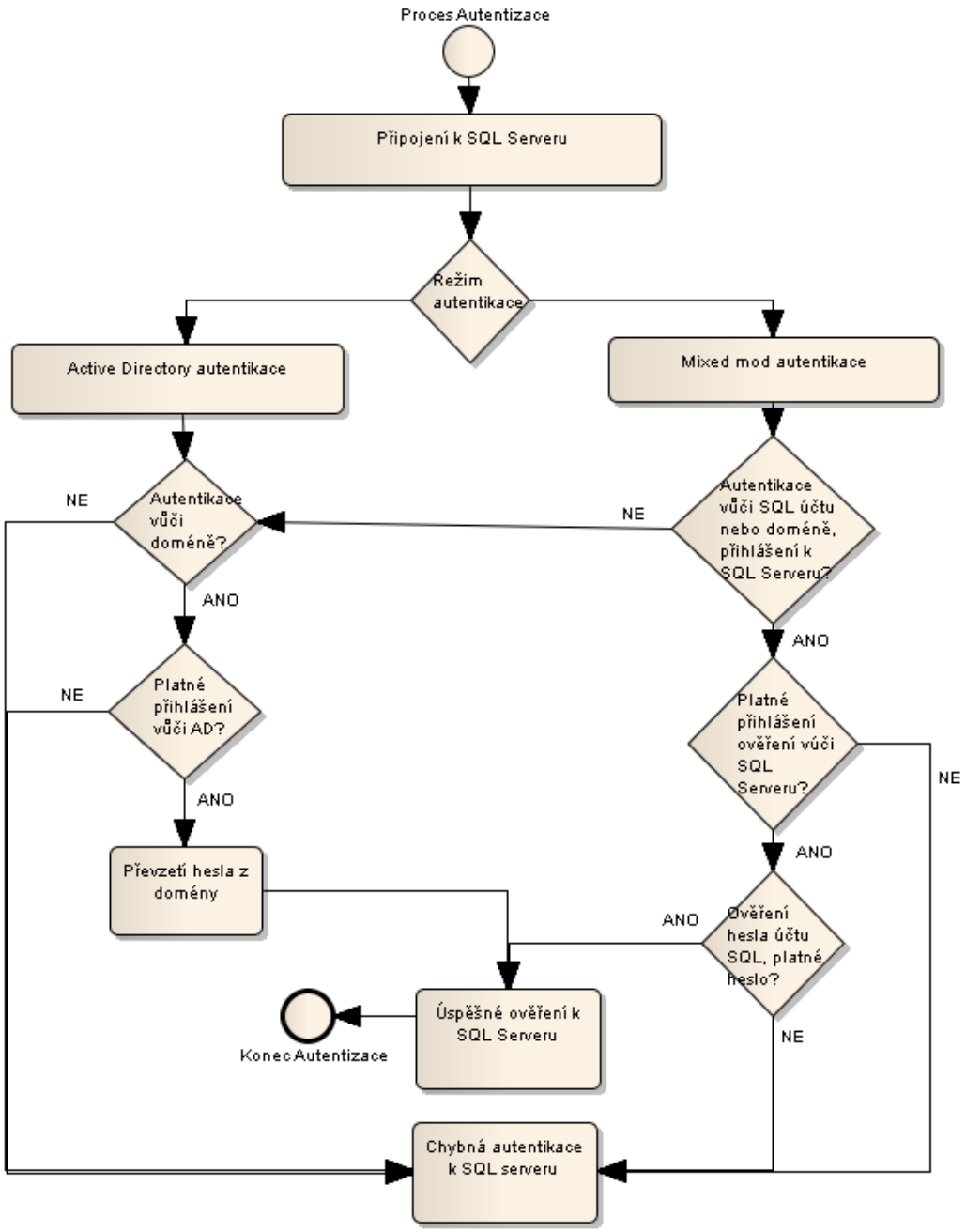
První úrovní chápeme systém autentizace. Druhou úrovní chápeme autorizace do systému.

První úroveň je systém, kdy určujeme, zda uživatel má platný uživatelský účet a heslo a chápeme ho jako důvěryhodnou osobou pro databázový systém.

Druhá úroveň autorizace je proces, kdy konkrétnímu účtu přiřazujeme dané oprávnění k daným tabulkám, funkcím a prostředkům SQL serveru.

4.1 Autentizace

Autentizace, nebo-li také autentikace je ověřovací mód a patří mezi klíčovou problematiku v zabezpečení SQL Serveru. Máme zde dvě možnosti jak se autentizovat vůči SQL Serveru a to buď MS Windows autentizací nebo MS SQL autentikace.



Obrázek 2 - Autentizace klienta

Popis Autentikace

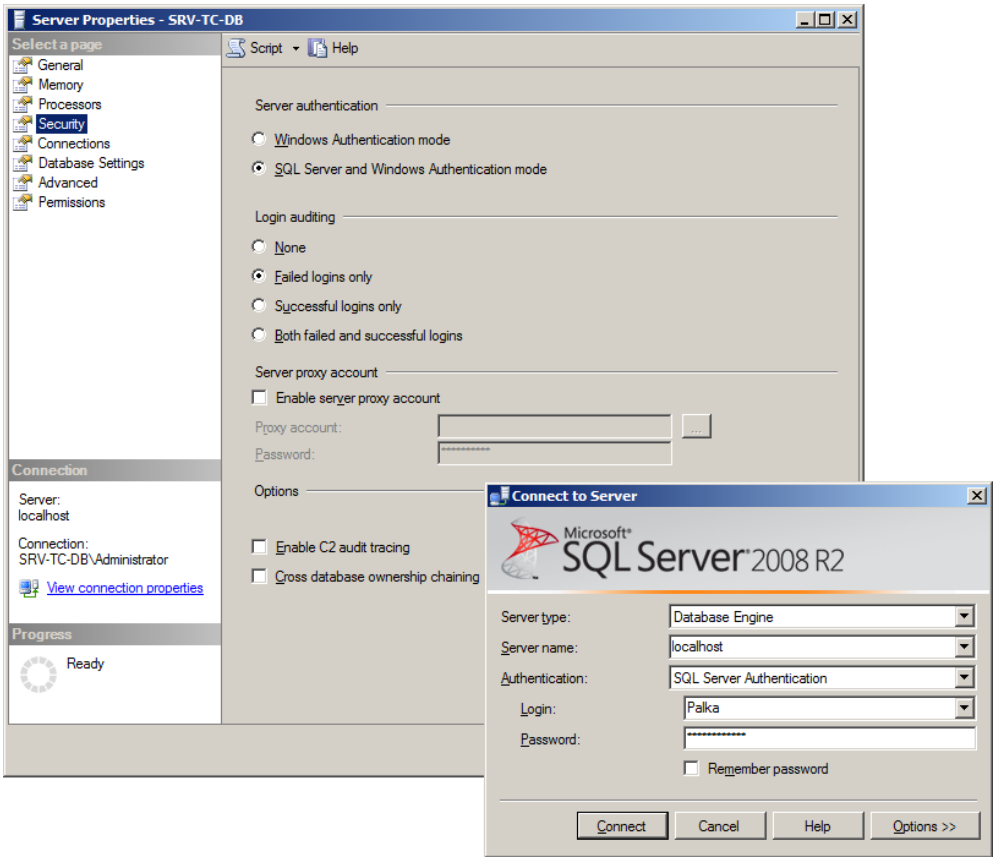
Režim Windows autentikace je chápán jako výhradní režim přístupu, takže pokud nastavíme tento režim, veškeré ověřování probíhá proti doménovému řadiči, který jako jediný server v doméně uchovává přístupové údaje pro klienty. V tomto režimu je potřeba si uvědomit, že SQL Server musí být členem domény a také, že k datům SQL Serveru se dostanou pouze ti klienti, kteří mají účet v dané doméně, neboli v důvěryhodné doméně.

V režimu autentikace SQL Serveru a Windows autentikace je možné přistupovat k datům SQL Serveru jako klient, který je členem domény, nebo pomocí ověřování samotného SQL Serveru. Tato volba se obecně nazývá MIXED Autentikační mód. Je třeba si uvědomit, že výhradní režim Windows autentikace je jediným ověřovatelem přístupu a jedná se o jiný server, který provádí ověřování. Režim autentikace SQL Serveru je režim, kdy přístup k datům ověřuje samotný SQL Server, respektive služba SQL Serveru.

MS SQL Autentikace je proces, kdy přímo SQL Server ověřuje uživatele vůči své vlastní databázi uživatelů. Databáze uživatelů je uložena v master databázi na datovém disku jako databáze master.mdf. Tento název nejde nijak změnit a musí být vždy dostupná pro službu SQL Serveru. Pokud by dostupná tato databáze nebyla, SQL Server by nemohl být spuštěn, protože tato databáze obsahuje i systémové nastavení, seznam prostředků a účtu pro službu SQL Serveru.

4.2 Nastavení Autentikace

Na následujícím obrázku je nastavení security autentikace na straně SQL serveru, kde zvolíme způsob autentizace.

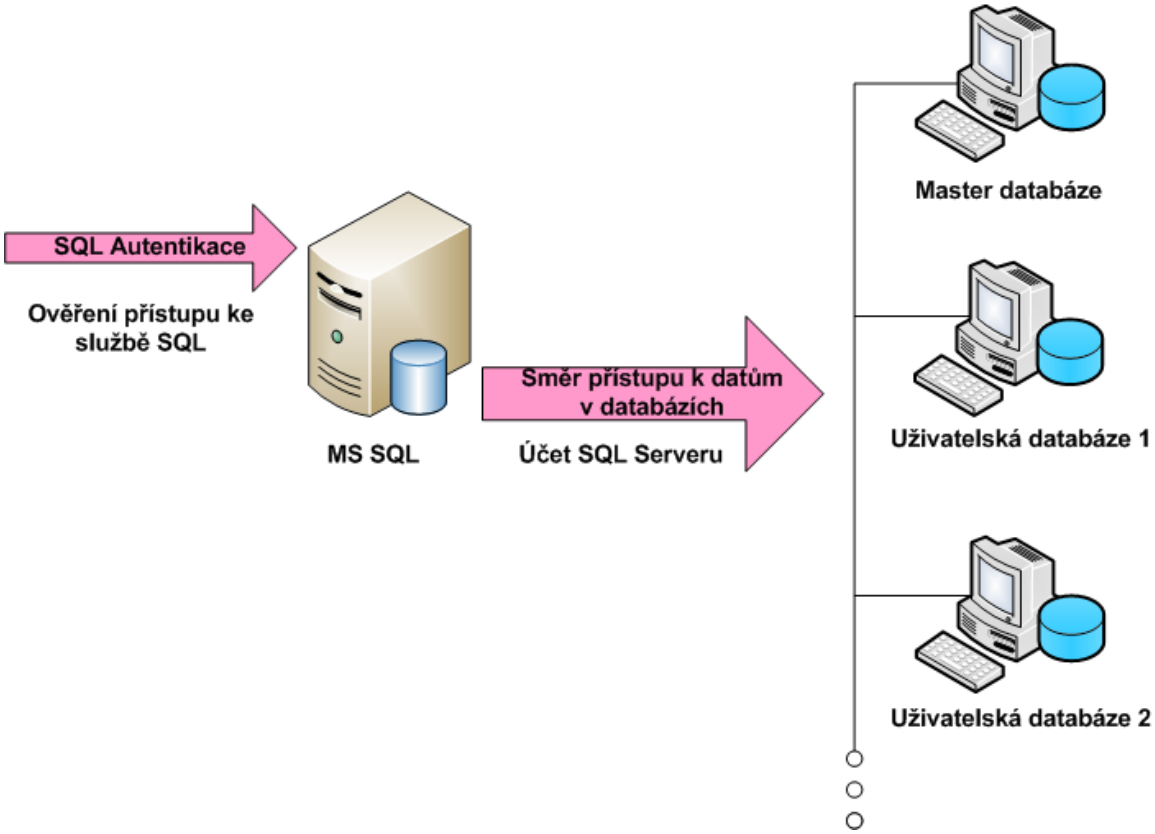


Obrázek 3 - Autentikace na straně klienta a nastavení autentikace na serveru

Nastaveni security autentikace se děje i na straně clienta. Klientem databázového serveru je jakákoliv aplikace, která se připojuji k SQL Serveru. Managament Studio je nástroj, přes který spravujeme SQL Server a lze ji chápat rovněž jako clientskou aplikaci, kde je třeba nadefinovat režim připojení.

4.3 MS SQL nativní Autentikace

MS SQL Autentikace je nativní prostředek autentizace uživatelů vůči datům v databázi, kdy přímo SQL Server ověřuje uživatele vůči své vlastní databázi uživatelů a dle práv přiřazení danému účtu, jsou zpřístupněny konkrétní informace.



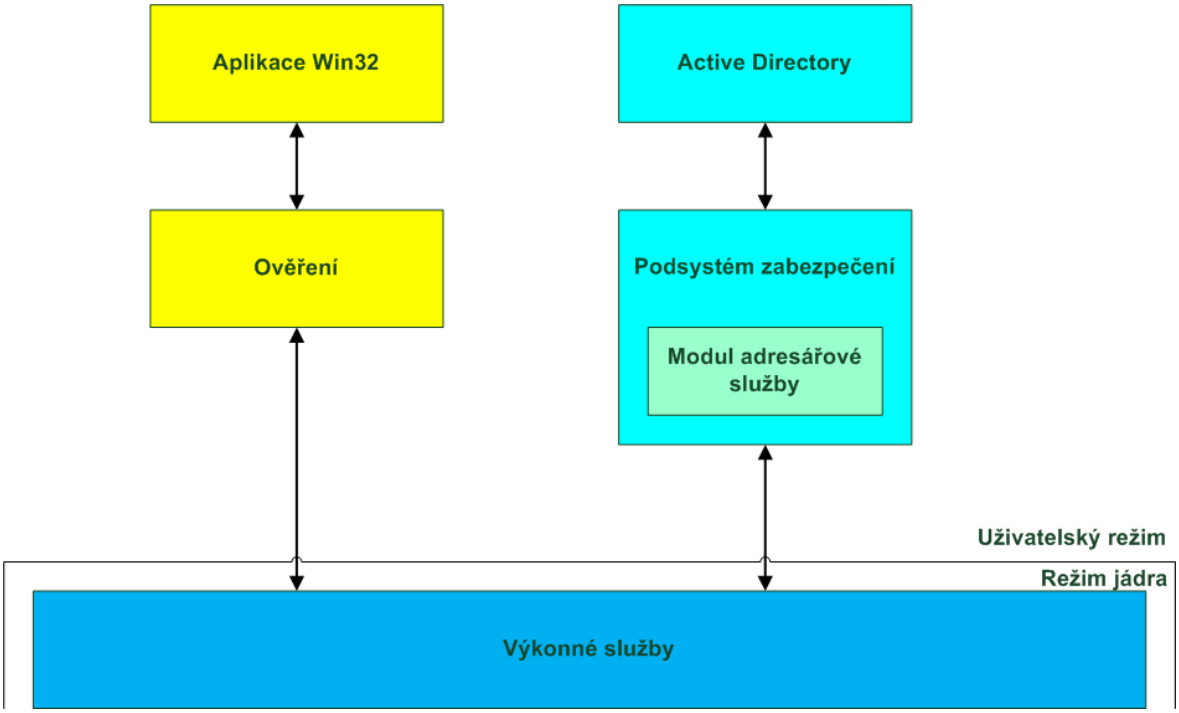
Obrázek 4 - MS SQL Autentikace

4.4 Active Directory autentikace vůči službě MS SQL serveru

Jedná se o výhradní režim autentikace. Nyní zde popíši princip autentizace uživatele a princip zabezpečení.

4.4.1 Architektura služby Active Directory

Z pohledu zabezpečení a autentizace služby Active Directory lze chápat tuto službu, jako podsystém zabezpečení dle následujícího obrázku.



Obrázek 5 - Active Directory Autentikace [3]

Podsystém zabezpečení funguje v uživatelském režimu následovně. Aplikace uživatelského režimu, které si ověřují (autentikují se) do AD, nemají přímý přístup k operačnímu systému ani hardwaru. To znamená, že požadavky aplikací musí projít vrstvou výkonných autentizačních služeb a před svým provedením vyžadují ověření.

Princip Ověření autentizace v Active Directory [3]

Každý prostředek ve službě Active Directory je reprezentován jako objekt. Každému uživateli, který se pokouší získat přístup k objektu, je nutné udělit oprávnění. Seznamy oprávnění, které popisují, kdo nebo co může k objektu přistupovat, se označují jako seznamy řízení přístupu ACL (Access Control List). Všechny objekty v adresáři mají přidružen seznam ACL.

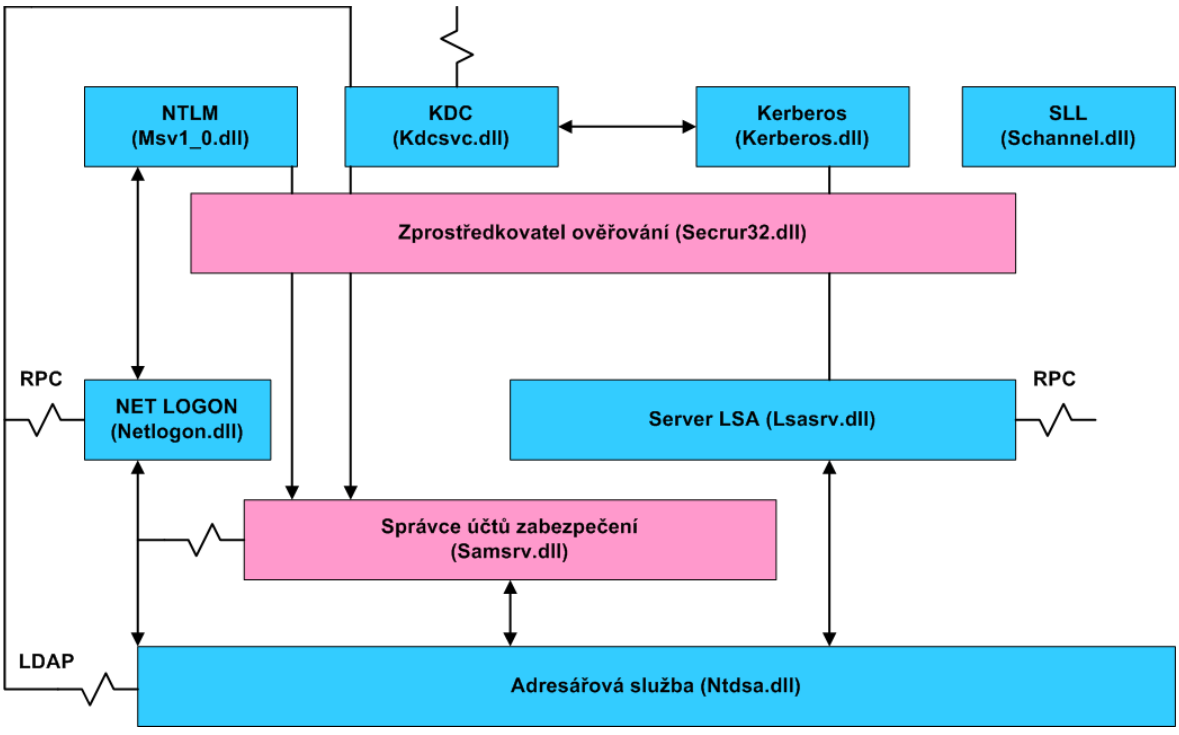
V širším měřítku lze oprávnění omezit pomocí zásad skupin. Infrastruktura zabezpečení služby Active Directory vynucuje pomocí zásad modelu zabezpečení pro několik objektů, které jsou logicky seskupeny. Lze také nastavit vztahy důvěryhodnosti mezi skupinami objektů. Tyto vztahy umožňují dále zvětšit rozsah řízení zabezpečení mezi důvěryhodnými skupinami objektů, které potřebují vzájemně si předávat oprávnění.

Princip ověření podsystému zabezpečení služby Active Directory [3]

Služba AD tvoří v rámci podsystému jednu z části LSA zabezpečení (Local Security Authority). Z následujícího obrázku je patrné, že LSA obsahuje mnoho komponent, které poskytují bezpečnostní funkce autentizačního systému a zajišťují, že řízení přístupu a ověřování funguje ve velmi robustně řešeném bezpečnostním režimu. Systém zabezpečení LSA plní i kromě ověřovacích a autentizačních funkcí i následující funkce:

- generuje identifikátory zabezpečení,
- poskytuje interaktivní procesy pro přihlášení,
- spravuje auditování.

4.4.2 Schéma podsystému zabezpečení založený na službě Active Directory



Obrázek 6 - Schéma podsystému zabezpečení AD[3]

Mechanismy ověřování [3]

- NTLM (Msv1_0.dll) se používá pro ověřování protokolu NTLM (Windows NT LAN Manager).
- Kerberos (Kerberos.dll) a KDC (Key Distribution Center – Kdesvc.dll) se používají pro ověřování Kerberos V5.
- SSL (Schannel.dll) je použito pro ověřování protokolu SSL (Secure Sockets Layer).

- Zprostředkovatel ověřování (Secur32.dll) se používá při správě ověřování.

Mechanismy přihlášení a řízení přístupu [3]

- Příkaz NET LOGON (Netlogon.dll) se používá pro interaktivní přihlášení pomocí protokolu NTLM. U ověřování NTLM předává příkaz NET LOGON přihlašovací pověření modulu adresářové služby a vrací identifikátory zabezpečení objektů klientům, kteří o ně požádali.
- Server LSA (Lsasrv.dll) se používá k vynucování zásad zabezpečení u ověřování Kerberos a SSL. V případě ověřování Kerberos a SSL předává server LSA přihlašovací pověření modulu odresářové služby a vrací identifikátory zabezpečení objektů klientům, kteří o ně požádali.
- Správce účtu zabezpečení (Samsrv.dll) se používá k vynucování zásad zabezpečení u ověřování NTLM.

Komponenta adresářové služby

- Adresářová služba (Ntdsa.dll) se používá k poskytování adresářových služeb v systému Windows Server 2008. Jedná se o modul, který umožňuje ověřovat hledání a načítání informací.

Jak je zřejmé, uživatelé jsou ověřeni předtím, než mohou pracovat s komponentou adresářové služby. Ověřování probíhá předáním bezpečnostních pověření uživatele řadiči domény. Jakmile jsou uživatelé ověřeni v síti, mohou pracovat s prostředky a provádět akce podle oprávnění a práv, která jim byla v adresářové službě udělena.

Komunikační protokol klienta s Active Directory [3]

Primárním protokolem pro přístup ke službě Active Directory je protokol LDAP. LDAP představuje standardní protokol pro přístup k adresářům, který funguje nad protokolem TCP/IP. Služba AD je kompatibilní s protokolem LDAP verzi 2 a 3. Klienti se pomocí protokolu LDAP dotazují na informace adresářové služby. Při této operaci vytvářejí připojení protokolem TCP k řadiči domény, na kterém běží adresářová služba. Výchozí port TCP používaný klienty LDAP má číslo 389 v případě standardní komunikace a 636 pro bezpečnou komunikaci na SSL.

Architektura služby Active DIrectory

Logická vrstva služby AD určuje, jakým způsobem jsou prezentovány informace obsažené v datovém úložišti, a řídí také přístup k těmto informacím. Logická vrstva to zajišťuje definováním oborů názvů a názvových schémat, která slouží pro přístup k prostředkům uloženým v adresáři. Díky tomu je k dispozici konzistentní metoda pro přístup k datům uloženým v adresáři bez ohledu na jejich typ. Díky této metodě je ověřen účet v AD a předány veškeré informace autentizačního procesu a to jak pro uživatele, tak i třeba pro objekt počítače, pokud je autentizace vázána a ověřována i na MAC adresu zadanou v objektu stanice. Objekt uživatele nese sám o sobě řadu ověřovacích atributů a vazeb na další objekty v AD, které lze vzhledem k bezpečnosti autentizačního systému použít.

4.5 Porovnání výhod SQL autentizace a AD autentizace

4.5.1 SQL autentizace

SQL Autentizace je typická pro ověřování různých databázových systémů, složených z uživatelského jména a hesla. SQL Server má svůj vlastní adresář uživatelských účtů publikovaných pomocí SQL autentizace. Při použití autentizace vůči více LAN sítím, kde různí uživatelé mají přístup k různým SQL databázím SQL, použijeme SQL metodu autentizace. SQL autentizace je hlavní metoda ověřování, která ovšem nese řadu problémů, pokud chceme účty integrovat s jinými adresáři. Mnohem pohodlnější je použití AD ověřování systému Windows, kdy získáme silnější zprávu nad účty, větší bezpečnost v silnějších a robustnějších nástrojích.

4.5.2 Ověřování systému Windows

Při použití AD autentikace, není uživatel nucen při přístupu k SQL serveru, použít ověření. Platné přihlášení ke službě Active Directory předá službě SQL Serveru veškeré autentizační informace.

Výhody AD autentikace:

- silnější a tím pádem bezpečnější autentizační protokoly,
- robustnější mechanizmy zabezpečení účtu,
- jednotná zpráva účtu a přístupů,
- jednotná zpráva hesel,
- robustnější nástroje pro definici politiky hesel a jejich zpráva,

- ověření uživatele vůči vice serverům – více doménových kontrolerů,
- odpadá ukládání hesel na aplikačních stranách do dočasných a konfiguračních souborů,
- přístup je definován pouze na standardní požadavek autentikace bez dalšího ukládání hesel = nejvyšší bezpečnostní standard.

4.5.3 Bezpečnost AD vs SQL autentikace

SQL autentizace je méně bezpečná, protože přístupy jsou definovány jako prostý text a ne jako objekty. Tento text je sice šifrován a vyžaduje heslo na straně clienta či aplikace přihlašující se do databáze. V případě windows prostředí LAN je tato autentizace považována za nejlépe integrovanou a výrazně bezpečnější metodu autentizace.

5 HARDWAROVÉ A SOFTWAROVÉ PROSTŘEDKY K ZAJIŠTĚNÍ ZÁLOHOVÁNÍ DAT SQL SERVERU

Zálohování dat, v našem případě zálohování databází, je jedna z nejdůležitějších metod pro ochranu dat a bezpečnosti dat SQL Serveru.

Před samotným zálohováním je třeba si uvědomit, jaká data chceme zálohovat, jaká je povaha dat, jak často je třeba zálohovat data a kolik dat máme a v budoucnu budeme mít pro zálohování.

Z těchto požadavků pro zálohování si určíme:

- jaké je pro nás vhodné zálohovací médium,
- jaké zálohovací zařízení potřebujeme,
- jakou politiku pro zálohování zvolíme.

Volba zálohovacího média

Zda zvolit páskovou mechaniku, nebo úložiště na diskový prostor, vychází z povahy dat, které chceme zálohovat. Jakou metodu zvolit a jaké jsou výhody té či druhé varianty jsou následující:

Pásková mechanika		Uložiště na diskovém prosoru	
Výhody	Nevýhody	Výhody	Nevýhody
Kapacita	Dlouhá obnova dat	Rychlá obnova dat	Kapacita
Nutný další server s DB pro zálohy na páskách			Práce se zálohami a téměř žádný dohled nad daty
Práce se zálohami a komfortní dohled nad daty	Stráta DB = ztráta dat na páskách	Jednoduchost práce se zálohami	

Tabulka 5 – Porovnání média pro zálohování

5.1 Zálohování na páskovou mechaniku

I když lze v dnešní době, provozovat zálohy na pásku snad ve všech možných režimech z povahy, kde se pásková mechanika nachází z pohledu dat, které chceme zálohovat, je vhodné tuto metodu zvolit následovně:

- Pásková mechanika na stejném serveru jako je databázový server.
- Pásková mechanika mimo databázový server, ale ve stejné lokalitě jako serverovna.

- Pásková mechanika mimo serverovnu s vybudovanou vlastní samostatnou infrastrukturou.

Výhody a nevýhody dle lokality

PM na stejném serveru	PM lokalita serverovna	PM mimo servrovnu
+ nejlevnější řešení		- nejdražší řešení
- bezpečnost zálohovaných dat	- bezpečnost zálohovaných dat	+ bezpečnost zálohovaných dat
- nutná obsluha pro výměnu pásek pro zálohování		- nutná dvojí infratruktura a další náklady na pořízení a údržbu
		+ rychlost
		+ přímé propojení páskové mechaniky k uložišti

Tabulka 6 – Porovnání úložiště pro zálohování dle lokality

Pásková mechanika v serveru chápeme takovou páskovou mechaniku, která není opatřena robotickým zařízením pro automatickou výměnu pásek.

Nejlepší a nejrobustnějším řešením je Páskovou mechaniku společně s robotickým zařízením umístit mimo serverovnu, jakožto místo, kde máme centrálně umístěny data. Je to nejdražší řešení, ale získáme tak největší bezpečnost i pro případ totální havárie serverovny, například požáru. Data jsou navíc indexována nad databází a práce s nimi je plně automatická. Stejně jako režim zprávy dat, kdy můžeme pro každou databázi určit, jak dlouho, či jaký režim záloh zvolíme. Nemusí nás dále zajímat, jak jsou data zálohována, protože o vše se nám stará řídící server.

Další obrovskou výhodou je přímé propojení páskové mechaniky k úložišti. Tím odpadá starost nefunkčnosti služby nad OS Serveru. Řízení a zpětná kontrola se děje na úrovni backup serveru.

A neposlední výhodou robotické páskové mechaniky je paralerizmus páskových mechanik a v dnešní době jdou kupředu s kapacitou i páskové mechaniky, kdy například pásková mechanika od IBM LTO5 má kapacitu až 3000GB (3TB), při použití HW komprese.

Naopak u levnějších řešení zálohování nedosáhneme na žádné výhody zálohování vyššími technologiemi.

Na zálohování databází SQL je potřeba se dívat i v širší míře a to jako na režim, zda lze celé zálohování centralizovat do jednoho celku. Kdy pomocí robustního řešení zálohujeme všechny servery, respektive všechna data, která chceme chránit. Například ze souborových,

mailových či jiných serverů. Pokud budeme zálohovat každý server samostatně, není téměř možné cíleně zpravovat zálohovaná data. Další výhodnou centralizovaného zálohování je cena, kdy potřebujeme pouze jeden řídící server.

5.1.1 Parametry LTO5 mechaniky

- Otevřený formát ukládaných dat se zpětnou kompatibilitou.
- Životnost archivace dat - 15let.
- Na trhu od druhého kvartálu roku 2010.
- Kapacita jedné pásky 1500GB/3000GB.
- Rychlost přenosu 140MB/s.

Technologie:

- WORM,
- Encryption,
- Partitioning,
- SAS a FC.

5.2 Zálohování na disky

Záloha na disky je hodně podobná zálohováním na pásky v nejlevnější variantě. Zejména se tato metoda používá pro případy, kdy pracujeme se zálohami poměrně často a potřebujeme mít samostatně konkrétní data v uceleném celku. Zálohování na diskový prostor je rozhodně levnější variantou než zálohováním na pásové mechaniky. Odpadá zde potřeba mechanik a pásek a umožní nám větší prostor, kde záloha bude umístěna. Cena diskového prostoru vůči páskovým mechanikám je v menších kapacitách výhodnější, jako při použití pásek. Obecně lze říci, že záloha na disky nám přinese vyšší výkon než na pásky.

5.3 Strategie a scénáře zálohování SQL serveru

Pokud už víme, na jaké médium provádět zálohy, popíši co je potřeba v databázovém pojetí zálohovat:

- systémové databáze,
- uživatelské databáze,
- systémové prostředky OS.

Systémové databáze: databáze potřebné pro běh SQL Serveru: db master, db model, db temp, db msdb.
Uživatelské databáze: aplikační databáze informačních či jiných systémů.

Systémové prostředky OS: prostředky OS, zálohy systémového oddílu.

Zálohovací metody SQL serveru:
Full backup – kompletní záloha databáze.
Diferential backup – rozdílová záloha databáze.
Trasaction log backup – záloha transakčního protokolu.

5.4 Scénáře zálohování

Pro každou uživatelskou databázi definujeme její scénář zálohování. Scénářem se rozumí, jak časová závislost vůči záloze bude nastavena pro režim obnovy. Jinými slovy, pokud budeme databázi obnovovat, tak definujeme tímto parametrem, zda obnovu databáze chceme i pro konkrétně daný čas, nebo pouze jen k datu provedení (zahájení procesu) zálohy.

Scénáře zálohování se nastavují konkrétně nad databází na záložce option – recovery mod.

Známe 3 druhy režimy obnovy:

- Full recovery mod – obnova dané db ke každému okamžiku.
- Bulk recovery model – obnova k ukončené transakci.
- Simple recovery model – obnova k procesu zahájení procesu zálohy.

5.5 Zálohovací strategie

Zálohovací strategie a postup nasazení zálohování je jedna z klíčových věcí informačního systému. Zálohovací strategii bychom měli chápat z pohledu IS a ne jen konkrétního serveru, který chceme zálohovat. Ve strategii bychom si jasně měli říci pro každý server, jak data chceme zálohovat. My se budeme zabývat pouze databázovým MS SQL Serverem.

Systémová databáze MASTER

Nejdůležitější a nejkritičtější je databáze master. Master databáze kromě uživatelských účtů a jejich práva a role udržuje i systémové nastavení jak služby sql serveru, tak i nastavení a informace všech databází včetně uživatelských. Bez této databáze nelze

provozovat databázi. Záloha databáze master by měla být natolik častá, jak často se mění účty nebo změny nad uživatelskými databázemi. Standardně se doporučuje jednou denně fullbackup. Master databáze umožňuje pouze fullbackup a rozhodně by se měla tato systémová databáze zálohovat do samostatného backupu.

Systémová databáze MODEL

Databáze model, i když je systémová, nenese žádné informace, které by bylo třeba pravidelně zálohovat. Obecně lze říci, že to je šablona pro uživatelské databáze. Podle nastavení této databáze budou vznikat databáze uživatelské.

Systémová databáze MSDB

Obsahem této databáze jsou data pro naplánované úlohy a procesy. Veškeré plány, jako například plány zálohování, jsou uloženy v této databázi. V této databázi ale není uloženo schéma procesu záloh, ale pouze job, který inicializuje tento proces a zajišťuje spuštění, reporty a kontroly nad naplánovanými úkoly. Rovněž jako v případě databáze MODEL, není třeba tuto databázi pravidelně zálohovat.

Systémová databáze TEMP

V režimu zálohování nejde o databázi, kterou je třeba vůbec zálohovat. Tato databáze se vždy vytváří znovu, při inicializaci služby SQL Serveru. Co se týká obsahu dat, jsou to pouze dočasné data, která nemají povahu informační, ale pouze systémovou. Informační data, o které bychom mohly přijít, se ukládají do transakčních logů dané databáze. Tuto databázi proto není třeba vůbec zálohovat.

Transakční soubory databází

Jsou to záznamy pro danou konkrétní databází, kde podle modelu režimu obnovy (full revovery model, bulk recovery model a simple recovery model) se určuje, jak se zaznamenávají změny v databázi. Při úplném záznamu a logování se zaznamenávají všechny transakce, které se spustí nad danou databází pro konkrétní čas. Transakční soubor chápeme jako přírůstkový soubor, kde každá transakce je zaznamenána v podobě transakcí nad databází. Tím, že máme tyto data k dispozici, můžeme se v případě obnovy vrátit k libovolnému času obnovy pro danou databázi. Jinými slovy jsme schopni obnovit databázi pro jakýkoliv čas a datum. Proto, abychom takovou obnovu byli schopni provést, je třeba vždy mít poslední full backup dané databáze, kde po skončení full backupu

navážeme na obnovu transakčních logu. Z hlediska zálohování, je transakční soubory třeba zálohovat co nejčastěji. Čím častěji tyto soubory zálohujeme, tím vice bezpečná data máme. Pokud se nám mění extrémě velmi často hodně záznamů a tabulek v databázi a transakční log nám extrémě díky tomu narůstá, je třeba provést full backup dané databáze a shring databáze. Tím se vyprázdní transakční log, který nesl informace o změnách a data se zaznamenala do full backupu.

6 DEMILATIRIZOVANÁ ZÓNA V PODNIKOVÉ SÍTI

Demilitarizovaná zóna je chápána, jako potencionálně nebezpečná zóna pro podnikovou síť. Je to zóna pro servery a prvky v síti, kde umísťujeme servery pro komunikaci s vnějšími servery a klienty. Služby serverů, které komunikují s vnějším prostředím, můžeme považovat natolik nebezpečné, že veškerou komunikaci separujeme do DMZty. Veškerou komunikace, ať už LAN-DMZ, nebo DMZ-Internet úzce specifikujeme jen na nejnutnější komunikaci s úzce specifikovaným seznamem adres. V praxi takovou komunikaci úzce popíšeme pomocí portů, slotů a adres. Veškerá komunikace je tím bezpečně oddělena pomocí firewallu na směrování paketů do konkrétních zón.

Nasazení služeb databázových systému do Internetu

Máme dvě možnosti, jak zabezpečit služby SQL serveru, jak je budeme poskytovat mimo LAN prostředí:

- nasazení SQL Serveru do DMZ,
- nasazení Aplikačního serveru do DMZ po komunikaci s DB serverem v LAN.

Výhody/Nevýhody řešení

SQL v DMZ	Aplikační Server v DMZ pro SQL v LAN
- méně bezpečné	+ bezpečnější
- nastavení AD autentikace do LAN složitá	+ autentikace do AD nebo SQL jednoduchá
- v případě úspěšného útoku jsou data v nebezpečí	+ data jsou LAN a potencionálně ve větším bezpečí
+ jednoduchost řešení	- server navíc v DMZ oproti řešení SQL v DMZ
- každá služba, co běží na serveru znamená další otevřené porty	+ minimum otevřených portů a komunikace ven

Tabulka 7 – Porovnání bezpečnosti dle umístění SQL Serveru

Nasazení SQL Severu do DMZ

Služba SQL Serveru v DMZ je úzce svázaná s ověřením autentizace. Možnosti autentizace:

- SQL Autentizace,
- AD Autentizace.

7 HARDWAROVÉ A SOFTWAROVÉ PROSTŘEDKY ZABEZPEČENÍ SERVERU A KOMUNIKACE ZAJIŠTUJÍCÍ BEZPEČNOST SERVERU A OCHRANU DAT

Mezi prostředky, které zajišťují zabezpečení serveru, patří zabezpečená komunikace. Metodou k této ochraně je zajištění a použití certifikátů pro důvěryhodnost komunikace mezi servery a clienty.

7.1 Certifikáty, bezpečná komunikace a důvěryhodnost

Zabezpečená komunikace mezi servery a klienty je jedna z dalších způsobů zabezpečení dat na serverech.

Veřejný klíč je přesná informace o objektu, je to jako občanka prokazující, s kým dochází ke komunikaci. Je to přesný popis identity, komu veřejný klíč patří.

Při lepší ověření autentikace klientů je vhodné použít certifikáty vázáné s daným účtem, nebo použít ověřovací karty.

Serverové certifikáty, HASH, popis komunikace

Digitální serverové certifikáty:

- jsou určeny výhradně pro servery,
- certifikát zajišťuje bezpečnou komunikaci mezi serverem a klientem,
- zajišťuje ověření pravosti serveru, s kterým stanice komunikuje,
- je vždy jasně dán vlastník serveru,
- časová platnost certifikátu a důvěryhodnost kořenového certifikátu je zřejmá z každého vydaného certifikátu, stejně tak informace, kde ověřím platnost certifikátu.

Použití certifikátu

Ověřuje se vždy důvěryhodnost zařízení, s kterým komunikujeme. Pokud má certifikát pouze server, tak si stanice ověřuje důvěryhodnost serveru. Pokud má stanice taky certifikát – dojde k ověření i serverem.

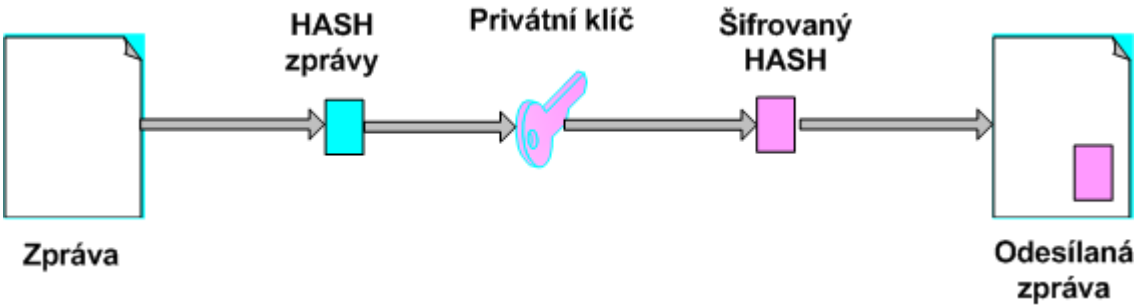
HASH

Hash je jedinečný klíč, který slouží k ověřené komunikaci mezi servery a klienty (stroji). Stanice (klienti) se servery nekomunikují pomocí jména a hesla, tímto se pouze autentizují do systému, ale komunikují po úspěšném ověření pomocí vygenerovaného platného HASHe. Proto HASH slouží pro komunikaci a autentizaci zajišťuje jméno a heslo, případně ověření jinými prostředky jako karty, či biometrické systémy.

HASH algoritmus je matematický algoritmus výpočtu nad daty do symbolického čísla, kde algoritmus je postaven tak, že čím menší změna ve zdrojovém kódu, tím více se liší HASH číslo. HASH je proto považován za otisk, signaturu či miniaturu v podobě čísla pomocí hašovací funkce, kde se vstupní posloupnost bitů převádí na jedinečně dané číslo pevné délky n bitů. Z hash hodnot nejde v žádném případě dostat původní zprávu.

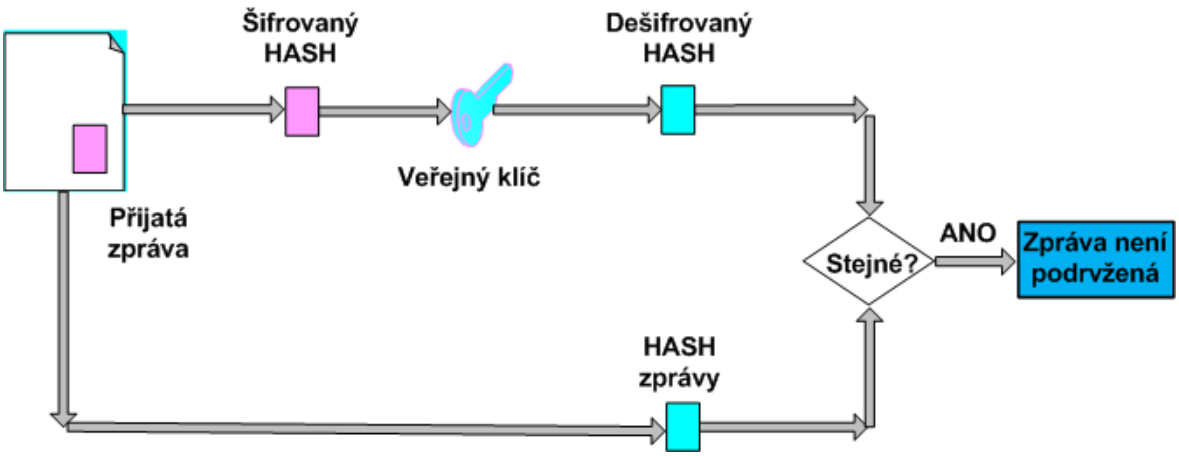
HAShovací funkce je tedy funkce jednosměrná.

Sestavení zprávy pro přenos dat mezi clientem a serverem



Obrázek 7 – Sestavení ověřené zprávy pomocí HASHe

Kontrola zprávy serverem pomocí kontroly HASH hodnoty



Obrázek 8 – Kontrola pravosti zprávy pomocí HASH hodnoty

7.2 Bezpečnostní politika hesel

Nejčastějšími druhy útoků do Informačního systému můžeme považovat:

- Hrubá síla prolomení hesla nad jedním či více uživateli.
- Získání HASH uživatele.
- Sociální útok, kdy útočník využívá znalosti místních zvyklostí a snaží se nějakým způsobem získat platné heslo, nebo jej uhodnout.
- Odcizení hesel (papírky na monitoru, špatná volba hesla. Hesla vychází z osobních údajů osob daného účtu – jména dětí, SPZ apod..).

Útok hrubou silou

Útok hrubou silou je nejběžnější a nejprimitivnější, ale taky někdy nejefektivnější metodou útoku do systému. Útočník zkouší všechny možné kombinace a to buď nad celou znakovou sadou, nebo pomocí slovníkových útoků.

V dnešní době se považuje síla útoku hrubou silou na 1Mil kombinací/s, což představuje 2,56 dne při 8 znakovém hesle.

V praktické části je výpočet bezpečnosti hesla a doporučení, jak silné heslo je třeba si zvolit.

7.3 Autentizační protokol Kerberos

Jedná se o autentizační sítový protokol, který je založen na ověření vzájemné identity. Pro komunikaci je použit ověřovací ticket (lístek). Ticket je jednorázové oprávnění s přesně stanovenou časovou platností, kterou je zaručeno prokázání identity protistraně s kterou komunikujeme. Při vzájemné komunikaci server – client je ověřená autorizace všech stran.

Kerberos používá symetrické šifrování, a proto v síti musí existovat další server (centrální server služby kerberos) se seznamem serverů a uživatelů.

Kerberos (Kerberos Network Authentication Service), je nyní ve verzi 5 (V5) pro OS2008 R2 popsán v RFC 1510. Jedná se o nejpoužívanější metodu autentikace pro Active Directory. Definuje autentizační proces, který poskytuje metody pro ověření identity, například pro pracovní stanici či uživatele (clienty). Pro autentizaci klienti používají Kerberos tickets (lístky), které reprezentují síťové credentials (pověření) pro klienta. Klient získá ticket od KDC (Kerberos Key Distribution Center) a tento lístek ukazuje serveru,

když se vytváří síťové spojení. Kerberos představuje identitu klienta pomocí uživatelského jména, hesla a jména domény.

7.3.1 Popis AD komunikace server – client zabezpečené službou kerberos

Vlastní autentizační proces popíši nejdříve zjednodušeně. Klient se autentizuje vůči Autentizačnímu Serveru a obdrží tiket. Poté kontaktuje řídící Ticket Granting Server (dále TGS) a pomocí tiketu prokáže svou identitu a požádá o přístup ke službě. Pokud na něj má nárok, pošle TGS klientovi další tiket, který je použit při komunikaci se servisním střediskem (SS) a tímto ticketem je potvrzen nárok na službu.

Podrobnější princip komunikace vypadá následovně: klient se jednorázově autentizuje pomocí vzájemného tajemství (totožného hesla) a dostane tiket opravňující uživatele ke komunikaci s řídícím serverem (TGT). Později, když chce klient kontaktovat nějaké servisní středisko, použije se (i opakovaně) tohoto TGT k ověření u TGS a tím k získání tiketu pro komunikaci se SS bez toho, aby bylo znovu využíváno původního známého tajemství.

Následuje popis jednotlivých fází [5]

Login uživatele

1. Uživatel vloží své jméno a heslo na klientském počítači.
2. Klient provede jednostrannou funkci (obvykle Hash) na vloženém hesle a to se stane tajným klíčem pro klienta/uživatele (klíč 1).

Autentizace klienta [5]

1. Klient odešle nešifrovaně zprávu na Autentizační Server (AS) obsahující uživatelovo ID a jeho žádost o přístup ke službě. (všimneme si, že jak heslo tak ani uživatelův tajný klíč nejsou odeslány) AS vygeneruje tajný klíč Hashováním hesla uživatele, které vyhledá v databázi.
2. Autentizační server zkontroluje, zda se uživatel nachází v jeho databázi. Pokud ano, odešle klientu nazpět 2 zprávy:
    - zpráva A: Klient/TGS klíč (klíč 2) zašifrovaný klíčem klient/uživatel (klíč 1).
    - zpráva B: Tiket oprávňující uživatele ke komunikaci s řídícím serverem (Ticket Granting Ticket)(TGT) obsahující ID klienta, jeho síťovou adresu,

životnost tohoto tiketu a klient/TGS klíč (klíč 2), to vše zašifrované tajným klíčem TGS (klíč 3).

3. Jakmile klient obdrží zprávy A a B, pokusí se dešifrovat zprávu A klíčem vygenerovaným z hesla uživatele (klíč 1). Pokud uživatel vložil nesprávné heslo tj. rozdílné než to, co má v databázi autentizací server, klíč 1 nebude odpovídat a tudíž s ním nepůjde dešifrovat zpráva A. Se správným heslem a tudíž i správným klíčem rozšifruje klient zprávu A, čímž získá klient/TGS klíč (klíč2). Tento je využíván pro další komunikaci s řídícím serverem (Ticket-granting server)(TGS). Klient nedokáže dešifrovat zprávu B, jelikož je šifrována s použitím tajného klíče TGS (klíč 3). V tuto chvíli má klient dostatek informací k autentizaci vůči TGS.

Autorizace přístupu ke službě [5]

1. Žádost o přístup ke službě se skládá ze dvou zpráv, jež klient odešle na TGS:
   - zpráva C: ta obsahuje zprávu B a ID služby, ke které klient žádá přístup.
   - zpráva D: autentikátor (složený z ID klienta a časové značky) šifrovaný klient/TGS klíčem (klíč 2)
2. Po obdržení zpráv C a D rozkóduje TGS zprávu C z níž získá zprávu B, kterou dále dešifruje svým tajným klíčem (klíč 3). Tím získá klient/TGS klíč (klíč 2). Pomocí tohoto klíče dále dešifruje zprávu D (autentikátor) a odešle klientu následující dvě zprávy:
   - zpráva E: klient/server tiket, obsahující ID klienta, jeho síťovou adresu, dobu platnosti a klient/server klíč (klíč 4), to vše zašifrované tajným klíčem služby (klíč 5).
   - zpráva F: klient/server klíč (klíč 4) šifrovaný klient/TGS klíčem (klíč 2).

Vyřízení žádosti o přístup ke službě [5]

1. Jakmile klient obdrží zprávy E a F od TGS, má dostatek informací k autentizaci vůči SS. Klient se k němu připojí a odešle mu tyto dvě zprávy:
   - zpráva E: z předchozího kroku (klient/server tiket zašifrovaný tajným klíčem služby (klíč 5).
   - zpráva G: nový autentikátor, obsahující ID klienta a časovou značku, zašifrovaný klient/server klíčem (klíč 4).

2. Servisní středisko (SS) dešifruje zprávu E pomocí svého tajného klíče (klíč 5) a získá tak klient/server klíč (klíč 4). Tímto pak dešifruje zprávu G, z níž zjistí autentikátor a poté odešle klientu zpět následující zprávu, aby potvrdil svou identitu a ochotu posloužit klientu:
   - zpráva H: časová značka z klientova autentikátoru navýšená o 1, zašifrovaná klient/server klíčem (klíč 4).
3. Klient dešifruje zprávu H klient/server klíčem (klíč 4) a zkontroluje, zda je časová značka správně navýšena. Pokud ano, tak může klient serveru důvěřovat a také může začít posílat žádosti o služby na server.
4. Server poskytuje klientu služby, o které žádá.

7.3.2 Výhody a nevýhody kerberos protokolu

1. Stačí zabezpečit pouze 1 server.
2. Nutný neustálý běh autentizačního serveru a řídícího serveru.
3. Nutný NTP server pro časovou synchronizaci času na stanici a serveru. Pokud se čas odchýlí více jak o 4:59 min., autentizace selže.
4. Pokud by došlo k ovládnutí řídícího serveru, lze podvrhnout identitu kteréhokoliv uživatele.
5. Podpora Hi Evailebility služby v cluster modu na více serverech s jedním adresářem účtů.

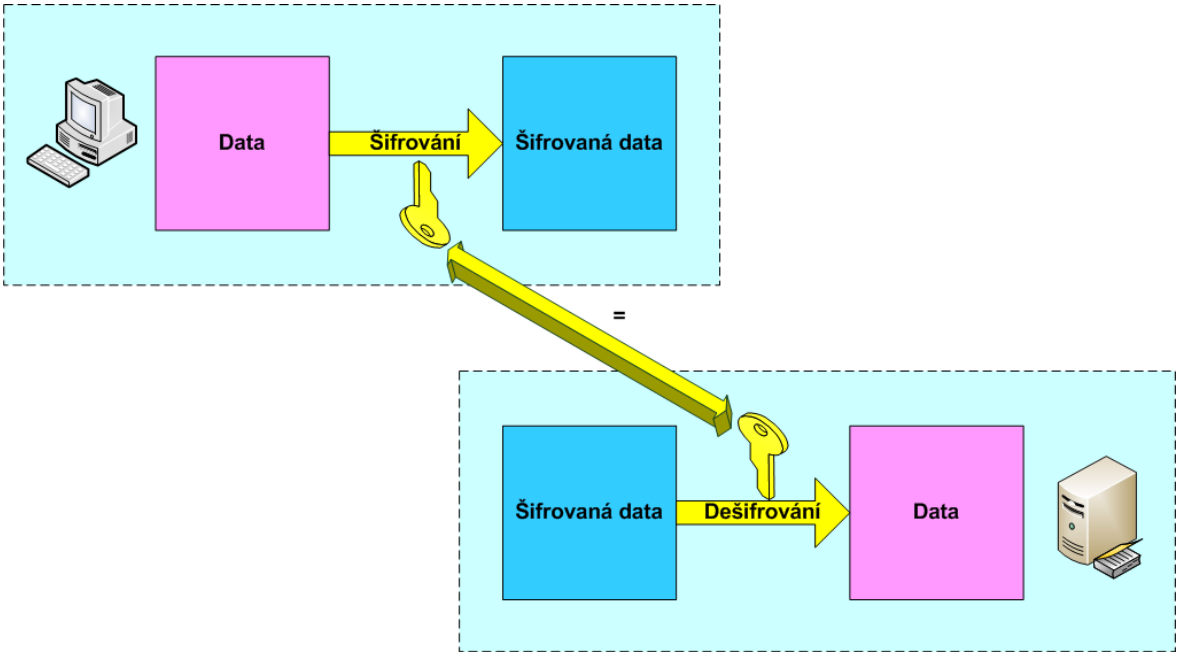
7.4 Šifrování

Šifrování je proces kódování dat za účelem jejich ochrany (kryptografie).

- Symetrické
- Asymetrické

7.4.1 Symetrické šifrování

Pro symetrické šifrování se používá stejný klíč k zakódování i rozkódování zprávy. Symetrické šifry můžeme rozdělit na blokové a proudové šifry.



Obrázek 9 – Symetrické šifrování

7.4.2 Blokové šifry

Zpracovávají zprávu a kódují data po ucelených celcích (blocích) stejné délky.

- AES, Blowfish, DES, GOST, IDEA, RC2, RC5, Triple DES, Twofish, Skipjack.
- Nejčastěji se používá blok o velikosti 64b., EAS používá bloky o velikosti 128bitů.

EAS je v dnešní době považován za bezpečný díky 128bitovému klíči.

7.4.3 Proudové šifry

Zpracovávají zprávu a kódují data po bitech.

- FISH, RC4

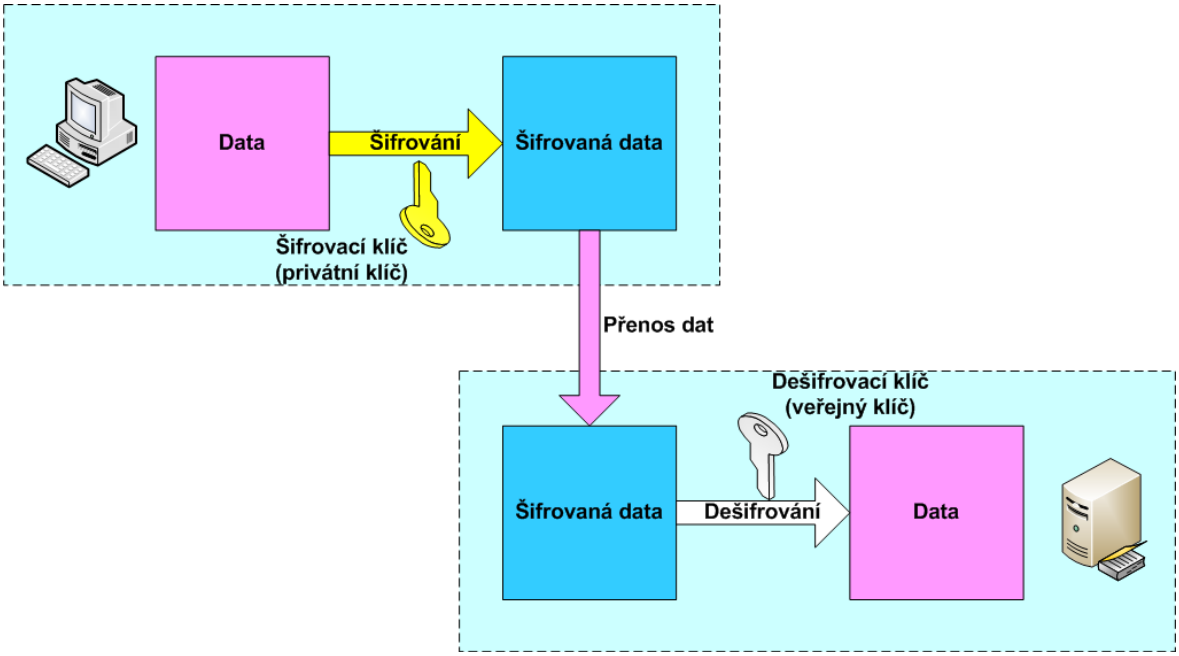
7.4.4 Asymetrické šifrování

Pro asymetrickou kryptografii je použita metoda klíče, kdy odesilatel a příjemce používají odlišný klíč, přičemž není třeba přenášet soukromý klíč odesílatele. Šifrování dat provede odesílatel zprávy veřejným klíčem a dešifrování se provede privátním klíčem adresáta.

Při šifrování je tedy veřejný klíč pro šifrování zprávy jiný, než soukromý klíč, kterým následně zprávu dešifrujeme. V této metodě je použita řada matematických netriviálních funkcí RSA a DSA algoritmů.

Princip asymetrické kryptografie je ten, že použijeme jednosměrnou matematickou funkci pro získání výsledku, kde z výsledku nejde sestavit vstupní informaci, ani ji nijak nalézt. RSA algoritmus je založen na triviálním matematickém výpočtu násobení, kdy z množiny daných hodnot násobíme velká čísla. Z takové matematické operace nejsme schopni udělat rozklad součinu na činitele.

Bezpečnostně čím déle by trvalo výpočetním výkonem rozluštit šifru, tím více je šifra bezpečná. Pokud je tedy algoritmus pro výpočet časově větší jak tisíc let, je považována za nerozluštitelnou, dokud se neobjeví lepší metoda pro faktorizaci dat.



Obrázek 10 – Asymetrické šifrování

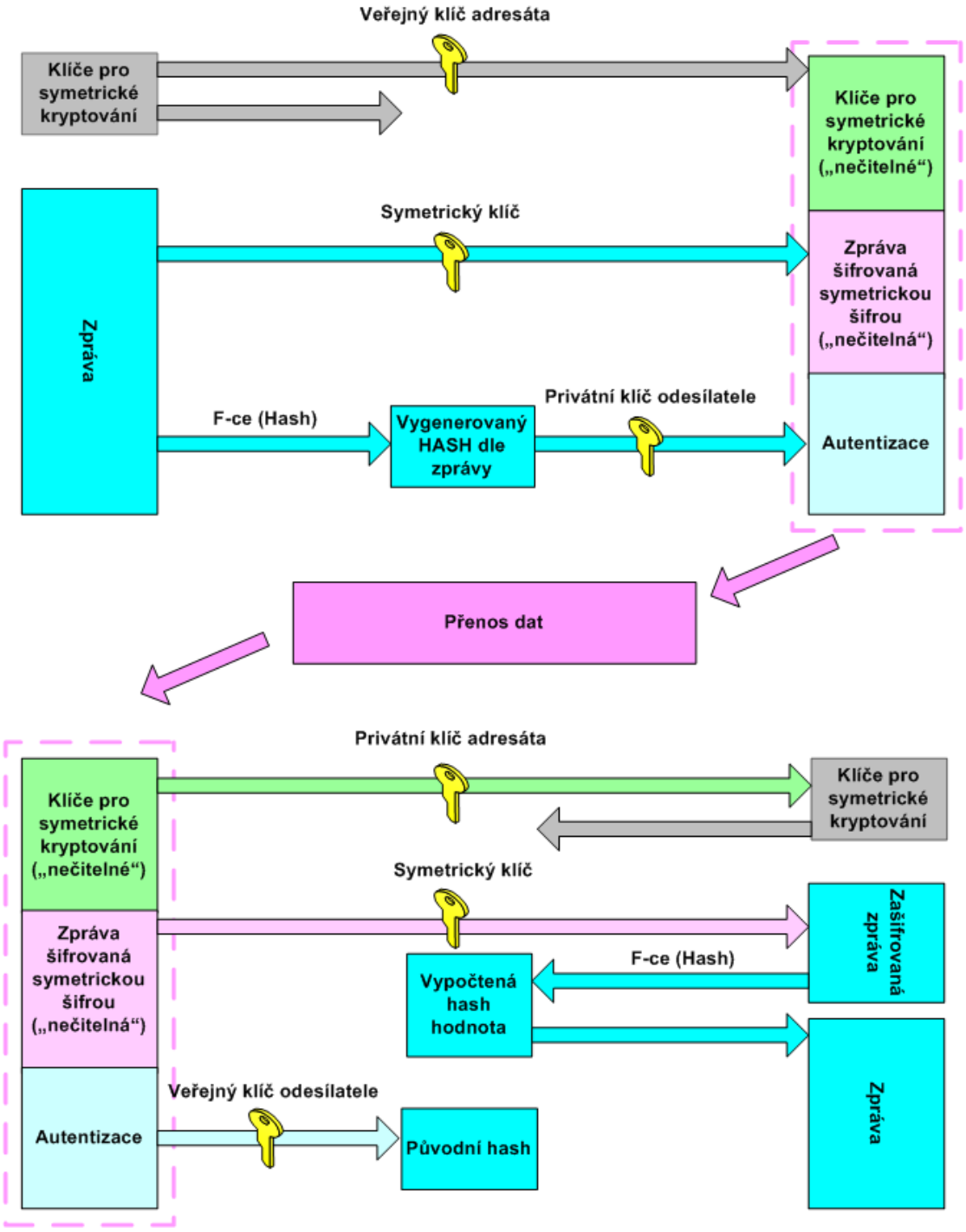
Nejpoužívanější asymetrické šifrování je RSA šifrování s veřejným klíčem, kdy délka klíče 2048b. je považována za bezpečnou.

Princip výpočtu

Bezpečnost RSA je postavena na faktu, že rozklad velkého čísla na součin prvočísel (faktorizace) je velmi obtížná úloha. Z čísla n = a*b je tedy nemožné zjistit hodnotu čísla (a) a (b), neboť nelze najít, ani jinak vypočítat z výsledné hodnoty, jaká čísla byla pro násobení použita. Naopak násobení dvou čísel je triviální výpočet postavený na minimálních výpočetních nárocích.

7.4.5 Použití šifrování symetrického s asymetrickým

Symetrické šifry mají veliké uplatnění v kombinaci s asymetrickými šiframi. Data zašifrujeme pomocí symetrického kódování a vygenerujeme k němu klíč. Pouze tento klíč je pak schopen odkrýt původní zprávu. Tento klíč vezmeme a zašifrujeme asymetrickou šifrou veřejného klíče a tím docílíme ochrany dat na trase a máme jistotu, že dešifrovaná data může přečíst pouze majitel soukromého klíče dané asymetrické šifry.



Obrázek 11 – Přenos dat zašifrované zprávy

8 POUŽITÍ CERTIFIKÁTU PRO OVĚŘENÍ PRAVOSTI V KOMUNIKACI

Před každou úspěšně ověřenou komunikací přes certifikát se ověřuje následující, jsou to údaje, které jsou obsaženy v každém certifikátu:

- Časová platnost certifikátu.
- Předmět certifikátu. V každém certifikátu je specifikováno, k jakému účelu byl vystaven. Ve vlastnostech certifikátu je to položka subject-details nebo subject alternativ name.
- Důvěryhodnost – certifikát vždy vydá certifikační autorita a duvěryhodná certifikační autorita musí být v seznamu path.

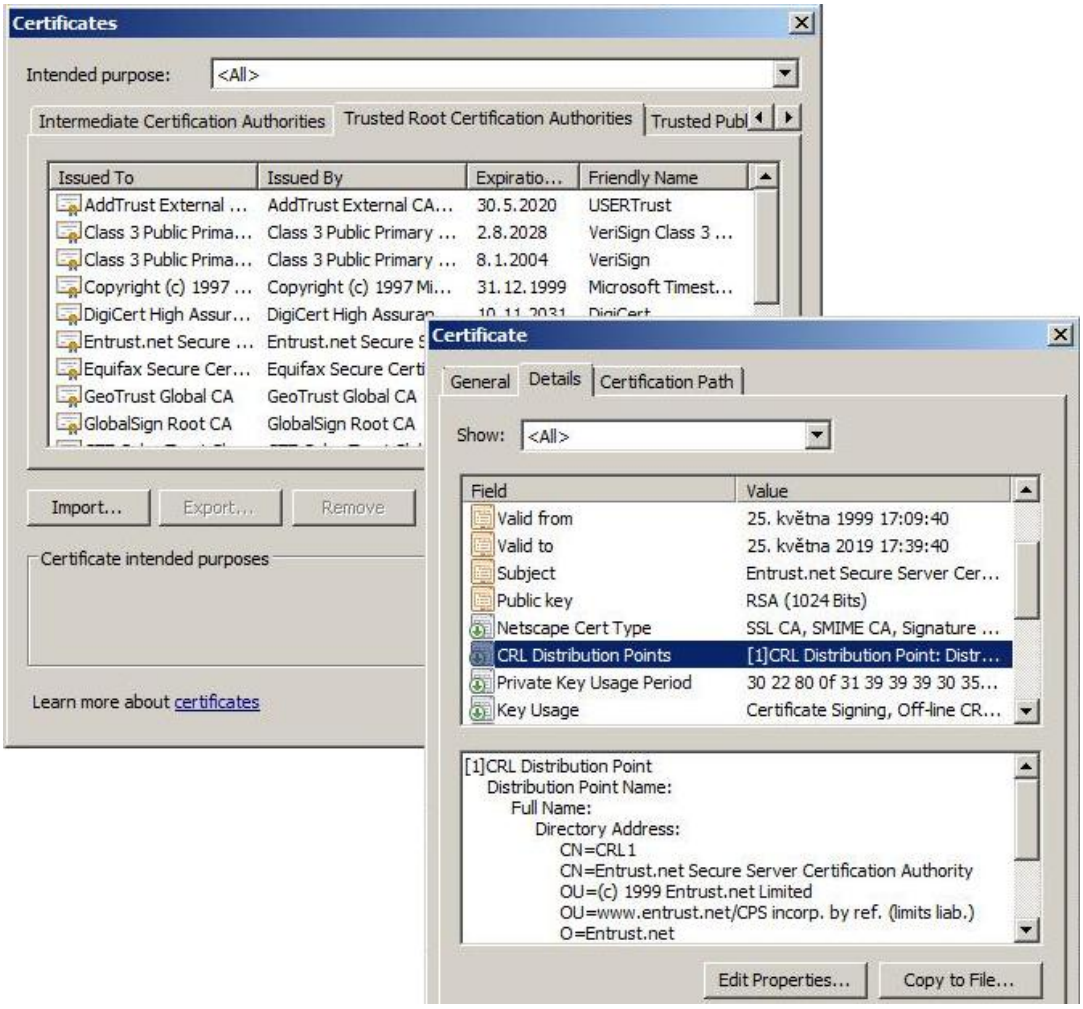
Pokud jsou splněny tyto podmínky, je vše v pořádku a je vše splněno pro ověření.

Velice doporučuji si předem důkladně rozmyslet, jaký certifikát instalujeme na servery a klienty. Pokud přidáme serveru kořenovou certifikační autoritu neznámého nedůvěryhodného serveru, vystavujeme se potencionálnímu nebezpečí. Díky přidané cizí kořenové certifikační autoritě se může stát, že budu důvěřovat i serverům, kterým vůbec důvěřovat nechci, a je zle. Tato cizí zlá certifikační autorita pak vystaví podvrh banky a náš server takové bance bude důvěřovat a vesele takovému serveru bude předávat veškeré údaje, například o kreditních kartách, účtech a podobně.

Neznamená to však, že všechny stavající certifikační autority, které již jsou na serveru nainstalovány odstraníme a budeme důvěřovat pouze našim. Všechny certifikáty, které po instalaci OS najdeme, jsou důvěryhodné a slouží například pro vztah důvěry pro aktualizační security balíčky.

Zpět k důvěryhodným certifikačním autoritám. Velká většina těchto autorit je stahována dynamicky od Microsoft již důvěryhodných serverů a tím vzniká strom důvěry. V takovémto stromu je pak garance pravých autorizačních autorit a vše se navzájem hlídá. Já jako správce si nemůžu dovolit přidat si kořenovou certifikační autoritu, jelikož neznám žadatele.

Při ověření si server nebo stanice stáhne z MS serveru aktualizaci a ověří si, zda autorita, s kterou chci komunikovat je ověřena, nebo je to nějaká garážová nedůvěryhodná autorita. Pokud systém ověří autoritu a je vše v pořádku, přidá si tuto autoritu do seznamu certification.



Obrázek 12 – Vlastnosti Certifikátu

Použití certifikátu:

- stanice si důvěřují,
- servery si důvěřují,
- stanice-servery si důvěřují.

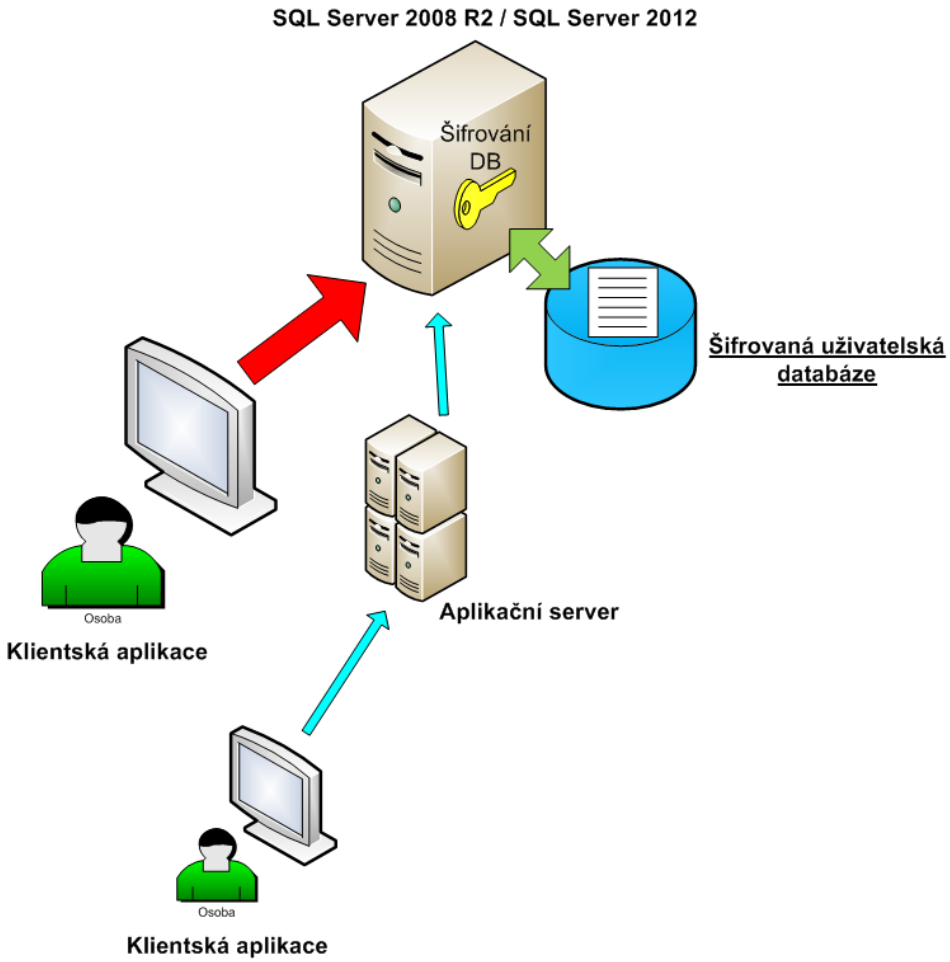
V každém certifikátu je odkaz, kde si ověřit certifikát, na jaké adrese: autority certificate adress. Na crl veřejně dostupné adrese je uvedeno, kde stáhnout seznam neplatných certifikátů (blacklist).

V praktické části realizuji instalaci PKI - Active Directory Certificate Services, službu zajištující vydávání a bezpečné ověřování stanic a serveru v síti.

9 ŠIFROVÁNÍ DAT NA SQL SERVERU

Zabezpečení databází na SQL Serveru lze od verze SQL serveru 2008 Enterprice Edition pomocí Transparent Data Encryption (TDE).

Při použití TDE šifrování je celá databáze šifrována naprosto automaticky bez nutnosti zásahu uživatele či administrátora a hlavně bez změny programového kódu užívaného pro databázové aplikace. Díky této obrovské výhodě, lze šifrovat jakoukoliv databázi bez jakýchkoliv úprav. Další obrovskou výhodou je šifrování datových nosičů a záloh uživatelských i systémových databází. To znamená, že i samotné zálohy databází jsou stále pod ochranou TDE a tedy není možné vzít si full backup databáze nebo fyzicky soubor databáze a obnovit jej na jiný server.



Obrázek 13 – Šifrování databází

Princip šifrování SQL Transparent Data Encryption [6]

Proces šifrování a dešifrování probíhá na úrovni načítání datových stránek do paměťového bufferu. V paměti se poté již nachází data v rozšifrované podobě. Cílem TDE je tedy chránit data ve fyzických souborech určených pro ukládání dat (mdf), transakčním logu (ldf) a záloze (bak). K šifrování dat dochází při jejich uložení na disk z paměťového bufferu zcela automaticky bez jakéhokoli zásahu uživatele. Celý proces vyžaduje určitý výpočetní výkon, obě operace – šifrování i dešifrování – potřebují pro svou práci systémové prostředky serveru. Tyto nároky jsou však mnohem nižší než u klasického šifrování sloupců, které bylo dostupné na dřívějších verzích SQL Serveru. Microsoft udává, že dochází ke zpomalení v řádu cca 3 – 5 %, což může být vzhledem k vyšší bezpečnosti dat přijatelná hodnota. Velkou výhodou TDE je také zachování možností indexace dat a bezproblémového využití cizích klíčů.

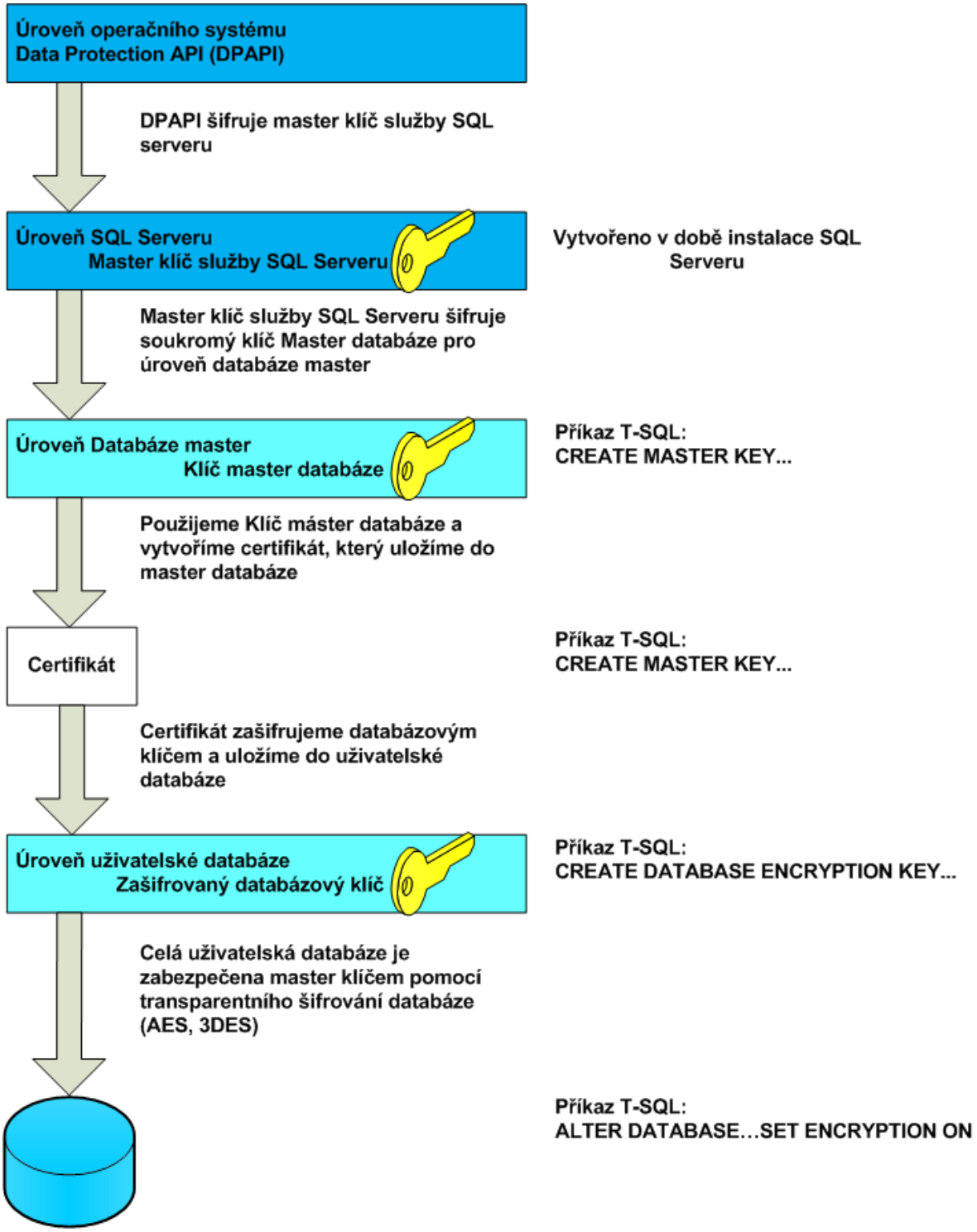
9.1 Implementace Transparent Data Encryption

Samotná implementace TDE spočívá ve 4 krocích, které je nutné provést:

- vytvoření master klíče,
- vytvoření nebo instalace certifikátu, chráněného master klíčem,
- vytvoření databázového šifrovacího klíče a nastavení ochrany certifikátem,
- konfigurace TDE v databázi.

Transparent Data Encryption postup šífrování databází

Na následujícím schématu principu šifrování databází je použito vlastních postřehů a nejedná se o přesnou citaci z anglického originálu.



Obrázek 14 – Princip šifrování databází [6]

10 VPN TUNEL, RADIUS PROTOKOL, RADIUS AUTENTIKACE FIREWALLU VŮČI ACTIVE DIRECTORY

Co je VPN tunel a jeho možnosti

VPN tunel jako takový je zpřístupnění části jiné sítě v zabezpečeném pásmu. Při takovém přístupu lze definovat, zda VPN klient na stanici odřízne stávající spojení a bude platný jen VPN klient, nebo budou k dispozici obě sítě. Bezpečnější pro komunikaci je restriktivní komunikace čistě na VPN tunelu, ale ne vždy je toto možné. Pokud klientská stanice nepřistupuje do naší sítě za účelem centrální a plně pokrytého systému služeb a využívá například jen SQL server, tak v takovém případě by klientské stanici fungovala pouze jedna jedniná a to VPN aplikace. Proto se tento režim používá jen pro centrální spojení VPN vůči infrastruktuře podniku.

Radius Autentikace Firewallu vůči Active Directory [7]

Mezi nejdůležitější vlastnosti patří jeho vysoká síťová bezpečnost, neboť transakce mezi klientem a RADIUS serverem je autentizována pomocí sdíleného tajemství, které není nikdy posíláno přes síť. Všechna uživatelská jména jsou přes síť zasílána šifrovaně. Uživatelské heslo je ukryto metodou založenou na RSA Message Digest algoritmu MD5.

Postup autentizace [7]

Postup autentizace je následující: uživatel vydá klientovi Požadavek na autentizaci, klient vytvoří Požadavek na přístup (Access Request) obsahující uživatelské jméno, heslo a ID portu, přes který je uživatel připojen. Požadavek na přístup je odeslán RADIUS serveru a čeká se na odpověď. Pokud nepřijde do určeného času (timeoutu) žádná odezva, Požadavek na přístup se opakuje, zpravidla 3 až 5 krát.

Pokud není splněna některá z podmínek, RADIUS server odešle Zamítnutí přístupu (Access Reject). Do datové oblasti paketu je dovoleno umístit maximálně textovou zprávu, která smí být zobrazena pomocí klienta uživateli. Žádné další atributy nejsou v odpovědi Access Reject povoleny.

Jestliže jsou všechny podmínky splněny, RADIUS server odešle Povolení přístupu (Access Accept), kde v datové oblasti paketu jsou uloženy všechny potřebné konfigurační informace (IP adresa, maska sítě, login uživatele a vše, co je potřeba předat požadované službě).

10.1 Použití protokolu Radius v DP

Pro použití RADIUS autentikace je vždy potřeba další služba, která zajišťuje předávání komunikace na aplikační vrstvě TCP/IP protokolu. RADIUS server lze chápat jen jako prostředek pro navázání bezpečného spojení při zadání, IP adresy cílového ověřovatele (NAS), jména a hesla. Informace je tedy poslána do takzvaného Network Access Server (NAS) zařízení přes Point-to-Point Protocol (PPP). Následně je předána službě RADIUS serveru informace pomocí RADIUS protokolu. RADIUS server ověří pravost informace použitím autentizačních prostředků systému. Po zadání správného jména a hesla server autorizuje přístup ke clientovi a ze svého nastavení přidělí IP adresu (např. DHCP) a další parametry spojení. Na clienta jsou zaslány i doplňující údaje, jako L2TP přihlašovací údaje, čas vypršení spojení, rychlost připojení a další parametry spojení. RADIUS protokol posílá hesla mezi NAS a RADIUS serverem v zašifrované podobě MD5.

V našem případě je chápána jako služba klienta služba zajišťující firewallové funkcionality, protože se ověřuje vůči radius serveru. Zní to zvláštně, ale opravdu v tomto případě, kdy firewall čerpá autentizaci z radius serveru, musíme jej považovat za klienta.

RADIUS server bude také upozorněn na spuštění nebo ukončení sezení, takže uživatel může platit přesně podle těchto RADIUS informací nebo mohou být tyto použity pro statistické účely. Tyto údaje mohou sloužit (při použití SIP přihlašovacích údajů z koncového VoIP zařízení) k účtování hovorů.

Formát RADIUS paketu [7]

RADIUS paket je zabalen do datové části UDP segmentu s hodnotou cílového portu 1812 - oficiálně přidělené číslo portu pro RADIUS je 1812. Skládá se z HEAD (8 bitů) identifikuje typ RADIUS paketu, identifikátor (8 b), délka (16 b) určuje velikost RADIUS paketu od HEAD po atributy, minimálně 20 bytů, maximálně 4096 bytů, datová oblast - Authenticator (128 bitů, je náhodně vygenerované číslo, které je použito na ověřování správné autentizace - viz níže) a oblast atributy, které nesou specifické autentizační, konfigurační nebo autorizační detaily pro požadavky či odpovědi. Konec seznamu atributů je určen délkou RADIUS paketu.

Ověření správnosti Autentizace: na sdílené tajemství (šifru) a Request Authenticator (128 bitů) je aplikována jednocestná hashovací funkce MD5, pomocí které je vytvořena 128 bitů velká hodnota, která je dále xorována s uživatelským heslem.

II. PRAKTICKÁ ČÁST

11 BEZPEČNOSTNÍ POLITIKA HESEL, VÝPOČET SÍLY HESLA

V praktické části bezpečnosti hesla, vypočítáme, jak důležité je mít nastaveno silné heslo na účtech, které jsou použity k autentikaci se servery.

Příklad prolomení 8 znakového hesla

62 znaků (26Velkých+26malých+10číslic) umocníme osmi (8 znakové heslo) = x / 1Mil kombinací za sekundu = 218340 / 60 = 3639 minut / 60 = 60 hodin / 24 = 2,52 dní. Příliš slabé heslo, pokud není použit speciální znak, nebo není použita metoda hlídání a zamykání účtů.

Kombinace znaků	Počet znaků v hesle	Výpočet	Výsledek
62	8	/1 Mil-s /60 / 60 / 24	2,52 dní
62	10	/1 Mil-s /60 / 60 / 24	9714 dní
72 (+10 spec. znaků)	8	/1 Mil-s /60 / 60 / 24	8,3 dní
62	5	/1 Mil-s	1 vteřin
72	14	/1 Mil-s /60 / 60 / 24 /365	3.190.423.555 let

Tabulka 8 – Výpočet síly hesla

Další ochranou, kromě nastavení dostatečně robustního hesla je blokace účtu pomocí politiky hesel. Po několika špatných pokusech o přihlášení, je účet na určitou dobu zablokován. Tím prodloužíme časovou odezvu a maximalizuje se bezpečnost a doba prolomení hesla.

Bezpečnostní politiky hesel

Mezi bezpečnostní politiky hesel patří řada způsobů, jak postupovat dostatečně bezpečně:

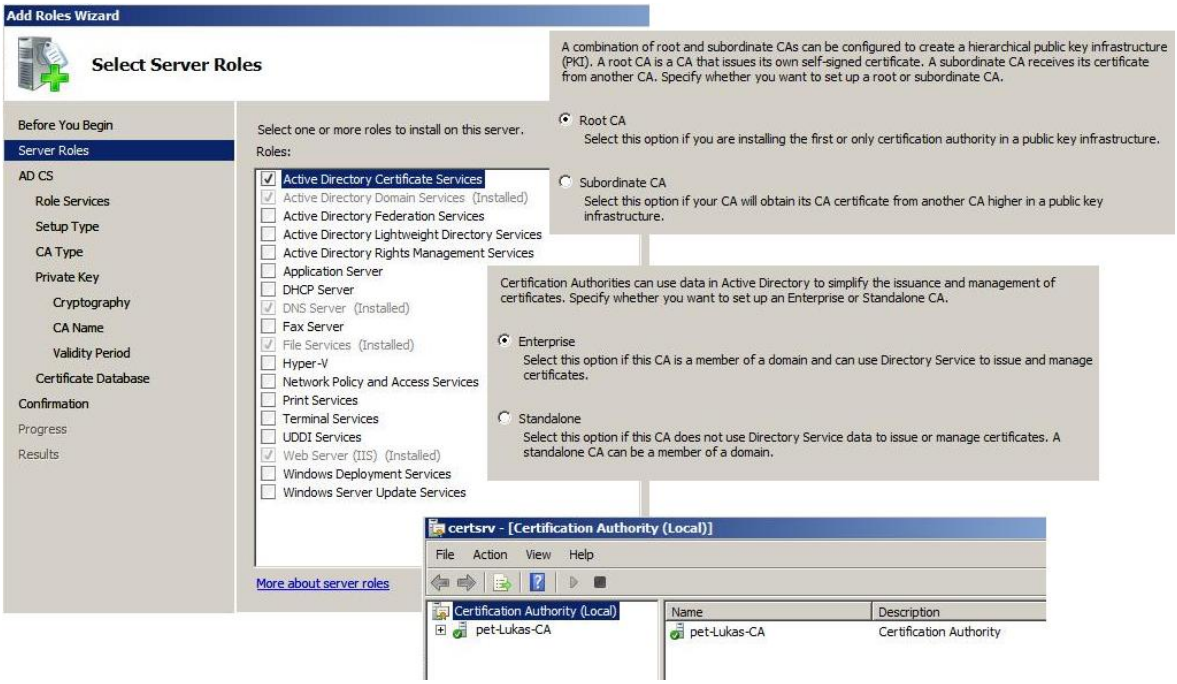
- periodická změna hesla,
- dostatečně silné heslo,
- mazání (blokování) neaktivních nebo zbytečných účtů.

12 INSTALACE VLASTNÍ CERTIFIKAČNÍ AUTORITY PKI – ACTIVE DIRECTORY CERTIFICATE SERVICES PRO OVĚŘENÍ PRAVOSTI V KOMUNIKACI POMOCÍ CERTIFIKÁTU

Popis ïnstalace vlastní certifikační autority PKI, pro vnitřní použití ověřené komunikace client-server. Vlastní certifikační autoritou zajistíme ověření klientských stanic a jejich pravost v síti. Instalaci provedeme v krocích následovně:

1. Přidáme roli serveru - active directory certificate services a průvodce instalace nás provede postupnou konfigurací služby.
2. Instalace role PKI - certification autority ano, web autentizaci ano, pokud clienty máme třeba na webu, volba enrollment je pro routery, kde není nabídka pro otevření žádné web stránky pro instalaci certifikátu.
3. Volba enterprise nebo standalone? – máme na výběr, jakým způsobem se budou vydávat certifikáty. Enterprice volba má výhodu, že clienti můžou dostat certifikát automaticky přes group policy. Volbou standalone budeme muset vše provést ručně.

   Role jsou v Active Directory a nejdou dát do trezoru (enterprice), do trezoru jde dát certifikáty verze standalone.
4. Cert. Rootovská, volbu zvolíme pouze pro kořenovou CA, který si podepisuje sama sobě.
5. Délka klíče – výchozí key character lenght = 2048b., definici certifikační karty necháme výchozí.
6. Název PKI autority - důležitá věc (domácí CA a podobně) název PKI autority vystupuje v LDAPu. A jako poslední důležitý parametr je doba rootovské autority tzn. její platnost.



Obrázek 15 – Instalace služby PKI

Poznatky z praxe

Vlastní rootovská certifikační autorita ANO, ale není vhodné ji umísťovat na síť a to ani lokální LAN. Root Autoritu musíme chránit, aby nám ji někdo nenapadl a nezmocnil se jí. Tuto autoritu nastavíme do režimu secure LAN (trezoru) a až další dostupné servery jsou v síti (ideálně Root Autoritu zvirtualizujeme).

Pokud máme rootovskou certifikační autoritu, tak ji zapíšeme do Active Directory a CA si zapíše svůj certifikát v interním ADcku a stává se tak výhradní autoritou, které se okamžitě a vždy důvěřuje.

Rootovskou CA vždy doporučuji vytvářet jako standalone a enteprice zvolíme CA, která automaticky pomocí group policy vydává certifikáty v režimu podřízené rootovské. Enteprice CA bude tedy zařazena do domény, Rootovská do domény zařazena nebude.

12.1 Konfigurace služby Active Directory Certificate Services

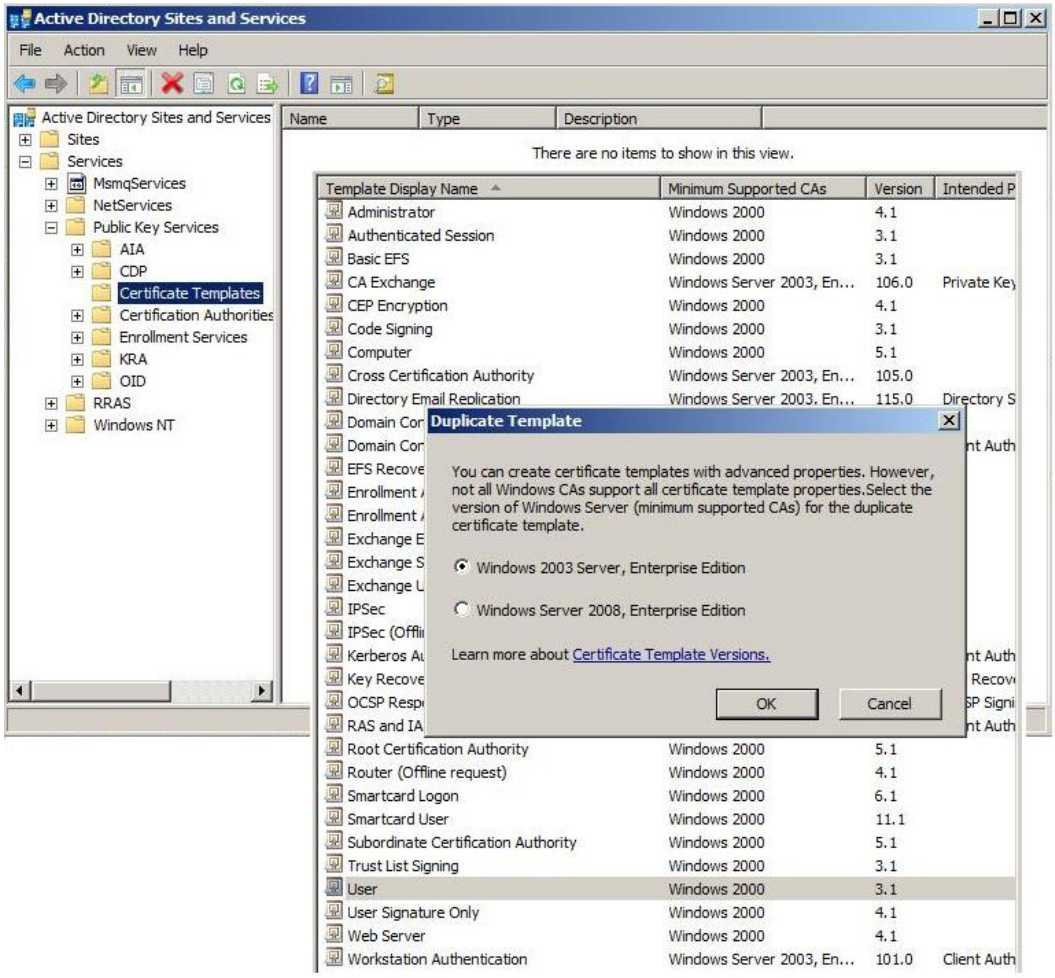
Provedeme instalaci Enterprice CA s automatickým vydáváním certifikátu stanicím.

V konzoli Certificate Templates Console můžeme vidět verze šablon a to ve verzi 1,2,3. Verze se liší také podle OS:

- v1 - přidělení certifikátu je potřeba provést ručně,
- v2 - automaticky dostane klient, když se přihlásí do domény po úspěšném ověření,

- v3 - Jsou podporovány algoritmy pro OS Vista, Windows 7 a vyšší.

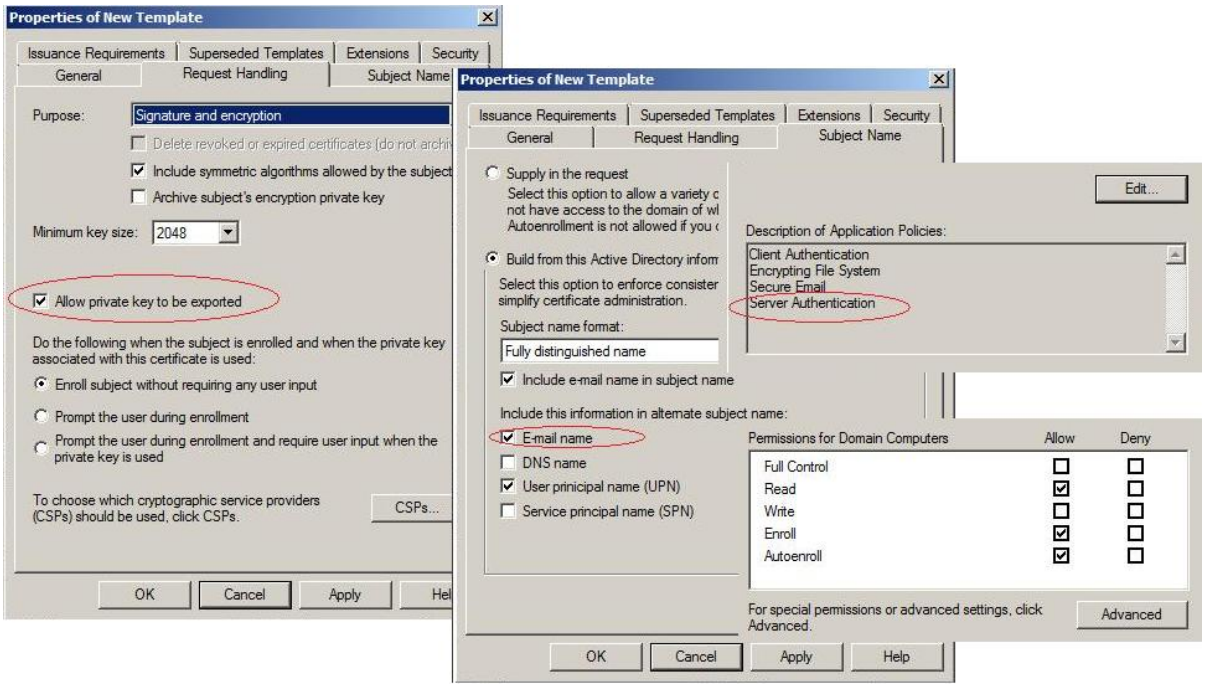
1. Vytvoříme si šablonu (template) – použijeme funkci duplikate template na users šabloně a zvolíme verzi OS.



Obrázek 16 – Konfigurace šablony PKI 1

2. V parametrech najdeme desítky voleb, proto uvedu ty, které je dobré změnit, či se nad nimi nějak pozastavit.
   - Na záložce Request handing je volba možnosti exportovat privátní klíč (Allow priváte key to be exported).
   - Pro režim instalace Enterprice CA s automatickým vydáváním certifikátů klientům musí být vyplněn v AD u účtu uživatele položka mail.
   - Přidáme roli certifikátu na záložce Extensions – k čemu je certifikát určen v našem případě Server Autentication (tím se ověřuje autentikace uživatelů vůči serveru).

- Dále na záložce Security potvrdíme práva pro doménový počítač Enroll a AutoEnroll.



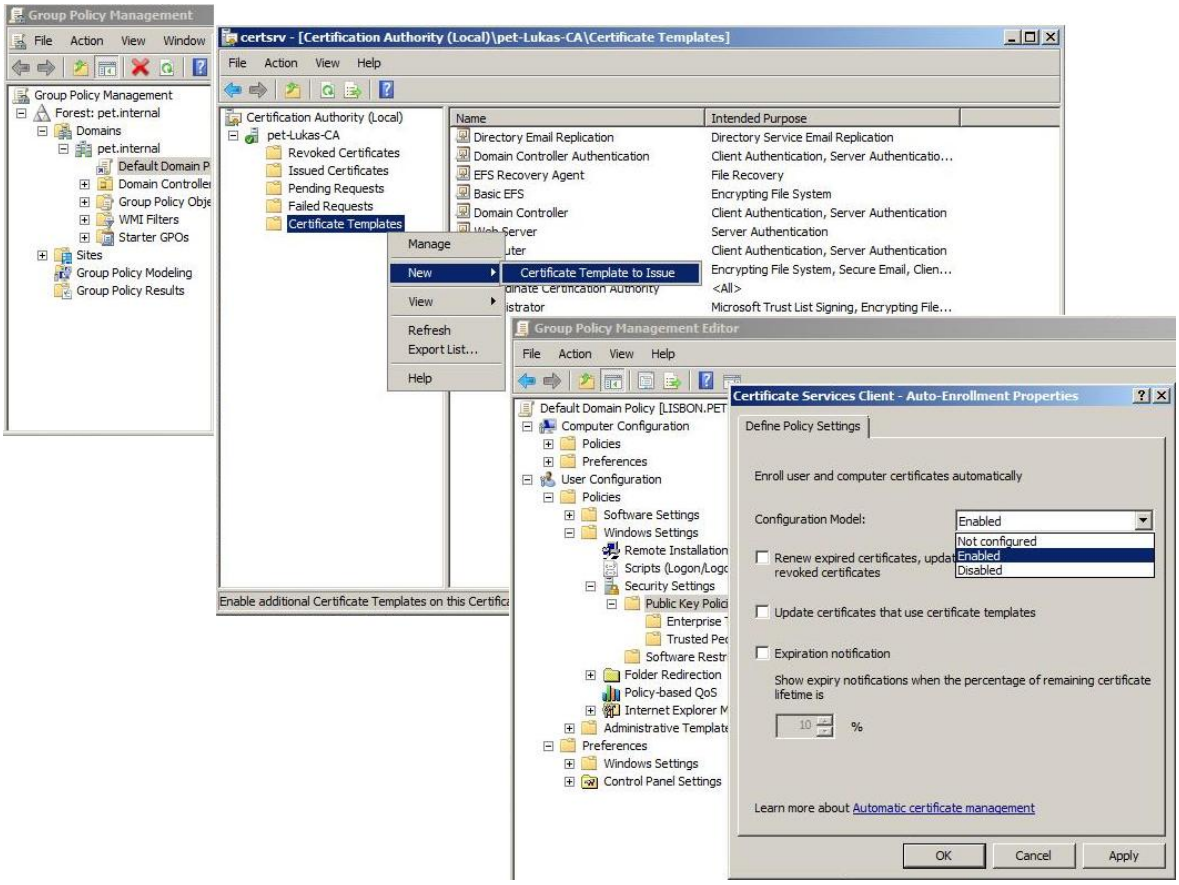
Obrázek 17 – Konfigurace šablony PKI 2

- Tím je certifikát pro Enterprice CA vytvořen a můžeme nakonfigurovat doručování stanicím.

3. Nastavení Group Policy (doručení certifikátů klientům). Spustíme consoli AD a nastavíme autoenrollment, certifikační autorita musí být enterprice - CAX (2ková - automatická), klient musí dostat oprávnění security: read, enroll, autoenroll - pokud je vše splněno klienti dostanou automaticky certifikát vydanou naší certifikační podnikovou autoritou.

   Na šabloně domain user nastavíme autoenroll - pojmenuji si šablonu a na záložce security - autoenroll - uložím nastavení a v modulu certsrv provedeme import.

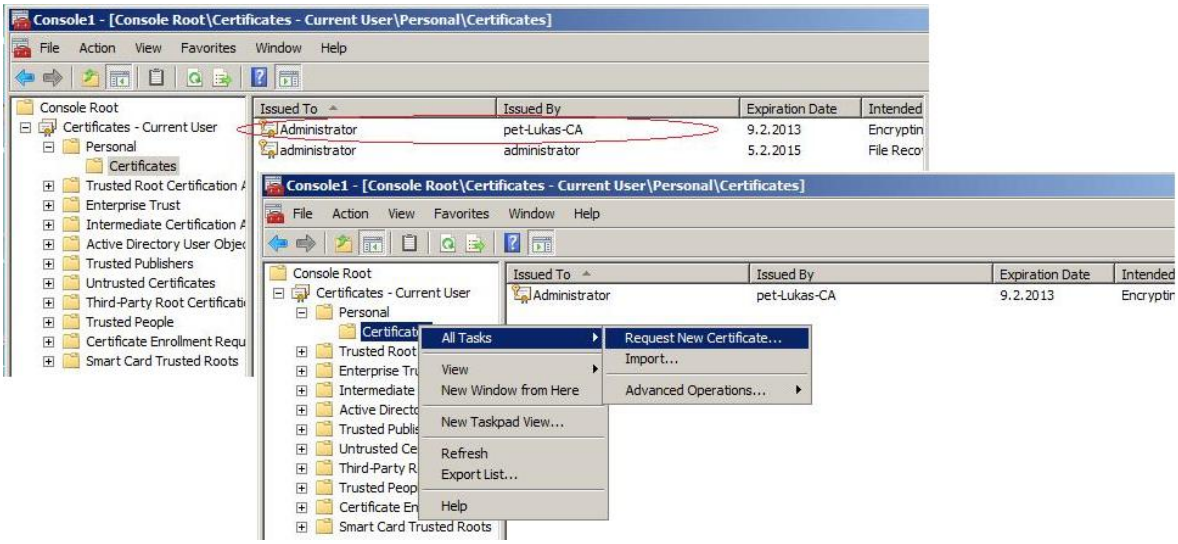
4. Následně přidáme šablonu certifikátu:
   - Spustíme příkaz MMC, zvolíme Console – Add/Remove Snap-in.
   - Přidáme konfigurační rozhraní Add Standalone Snap-In.
   - Vybereme Certificates ze seznamu snap-ins. A potvrdíme Add The Certificates Snap-in.
   - Vybereme jednu z možností My user account nebo Computer account, podle toho, zda chceme zabezpečit komunikaci na klienta z pohledu účtů nebo stroje.



Obrázek 18 – Konfigurace žádosti o certifikát na službě PKI

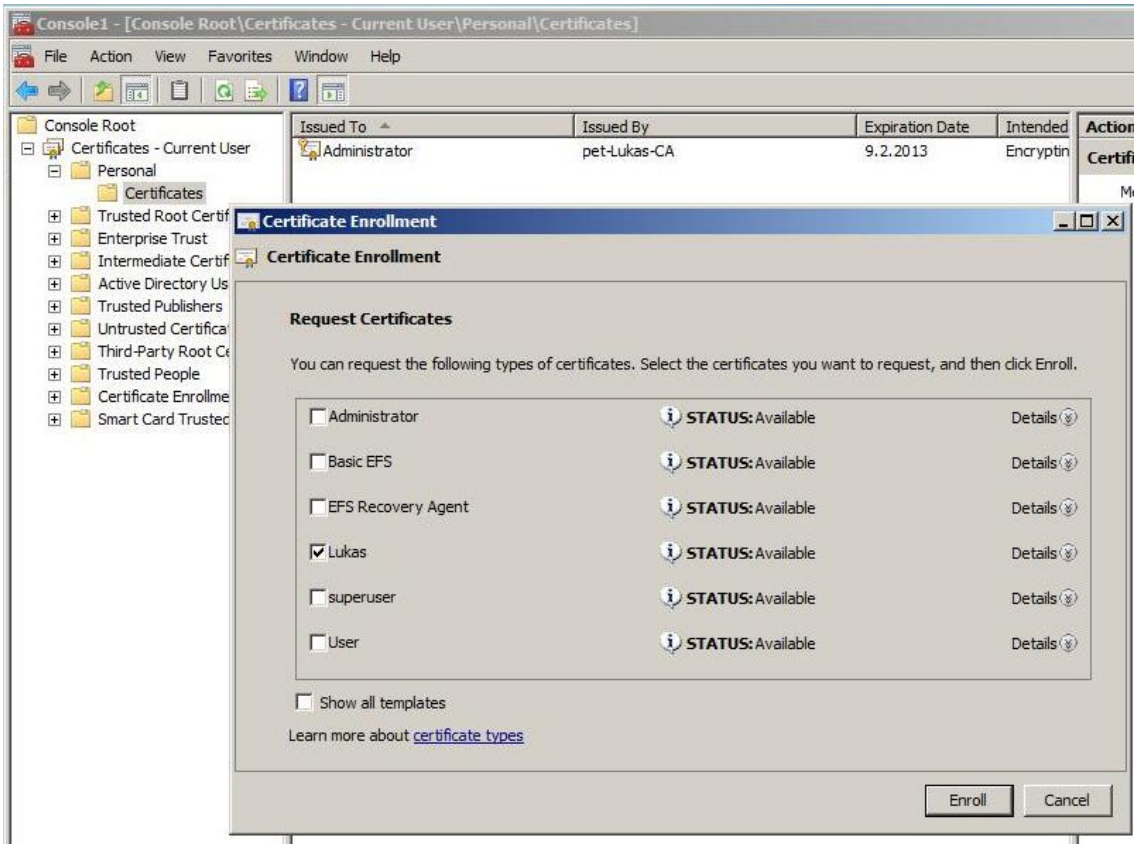
Na obrázku 18 je konfigurace automatické žádosti o certifikát na doméně – na konzoli group policy management - nastavíme politiku pro uživatele a musíme vyplnit na uživatelském účtu v AD položku email v atributu email adress.

5. V menu Přidání šablony - console1 {console root\certificates – Currest User\personal, certificates] - task – zvolíme add certificate environment a přidáme certifikát do šablony.



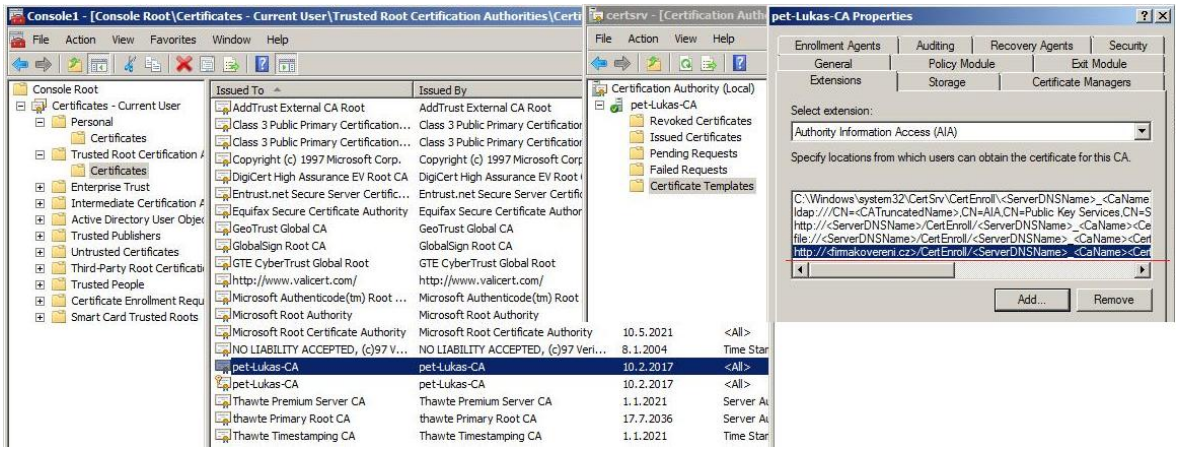
Obrázek 19 – Požadavek o certifikát Klientem

Na následujícím obrázku je znázorněna konfigurace žádosti o certifikát, kterou potvrdíme vybráním certifikátu, který jsme před chvílí vytvořili a jeho odsouhlasení pomocí tlačítka Enroll.



Obrázek 20 – Požadavek o certifikát

6. Jako poslední krok je nastavení důvěryhodnosti certifikátu. Zde zadáme, vůči jakému serveru si bude ověřovat důvěryhodnost. Musí sedět server se seznamem odvolaných certifikátů.



Obrázek 21 – Nastavení důvěryhodnosti serveru pro certifikáty

12.2 Instalace certifikátu na klientovi (stanici) a možnosti bezpečné komunikace

Uživatel nyní může data vůči serveru šifrovat, podepisovat či mít ověřenou autentikaci na server.

Instalace díky CA ve verzi Enterprise proběhne plně automaticky a skrytě.

Po autentikaci klienta do AD se provede spuštění balíčku na základě konfiguračního balíčku na group policy. Na stanici je zobrazena bublina na liště s informací, že byl doručen certifikát na stanici automaticky!

Od této chvíle stanice komunikuje bezpečně opravdu s tím severem, s kterým potřebuje.

Žádost pod Internet Information Services (IIS) do IPsecurity a požadavek certifikační autority je zobrazen na následujícím obrázku.

Obrázek 22 – Požadavek ISS o certifikát

12.3 Poznatky z instalace PKI

- Ověření certifikátem lze doručit i na zařízení, jakou jsou switche a wifi zařízení, přes které klient komunikuje se serverem.
- Pokud použijeme smart kartu zajistíme o úroveň vyšší bezpečnost, díky generátoru klíčů.
- Manažer pro certifikaty , použijeme konzoli certmgr.msc.
- Je třeba dát pozor na archivaci privátních klíčů - primární klíč by měl mít jen recovery agent pro správu těchto klíčů.
- Autentikace na stanici pomocí čipové karty + pinu.

Certifikát vystaven pro použití pro domain autentication v alternamе: přihlášení do domény, doménový řadič musí byt schopen zkontrolovat vůči VRLku, že nebyl zneplatněn.

driver + crypto api = logon stanice pomocí tokenu nebo čipové karty.

12.4 Princip bezpečné komunikace serveru s klientem

Teď, když už jsme popsali certifikáty, popíši, jak dojde k ověření serveru s klientem.

Server pošle certifikát pro bezpečnou komunikaci v bezpečném kanálu a tím se naváže spojení, kde dojde k šifrování a dešifrování dat pro přenos informací.

Dalšími aspekty PKI bezpečné komunikace je šifrování, problematika sestaveni HASHe a další. Tyto věci jsou detailně popsány v teoretické části.

12.4.1 Jak funguje HASH v praxi

Pomocí příkazu: *certutil.exe -hash název souboru*, vygenerujeme HASH, který se následně využívá při ověření pravosti zprávy. Nejjednodušeji to lze chápat na souboru, kdy vygenerujeme hash a následně pošleme hash s dokumentem příjemci a neověřeně. Cokoliv se po trase změní, nebude nám sedět hash s originálem. Na základně tohoto hashe se ověří na druhé straně pravost dokumentu a dokument je prohlášen za důvěryhodný či nikoliv.

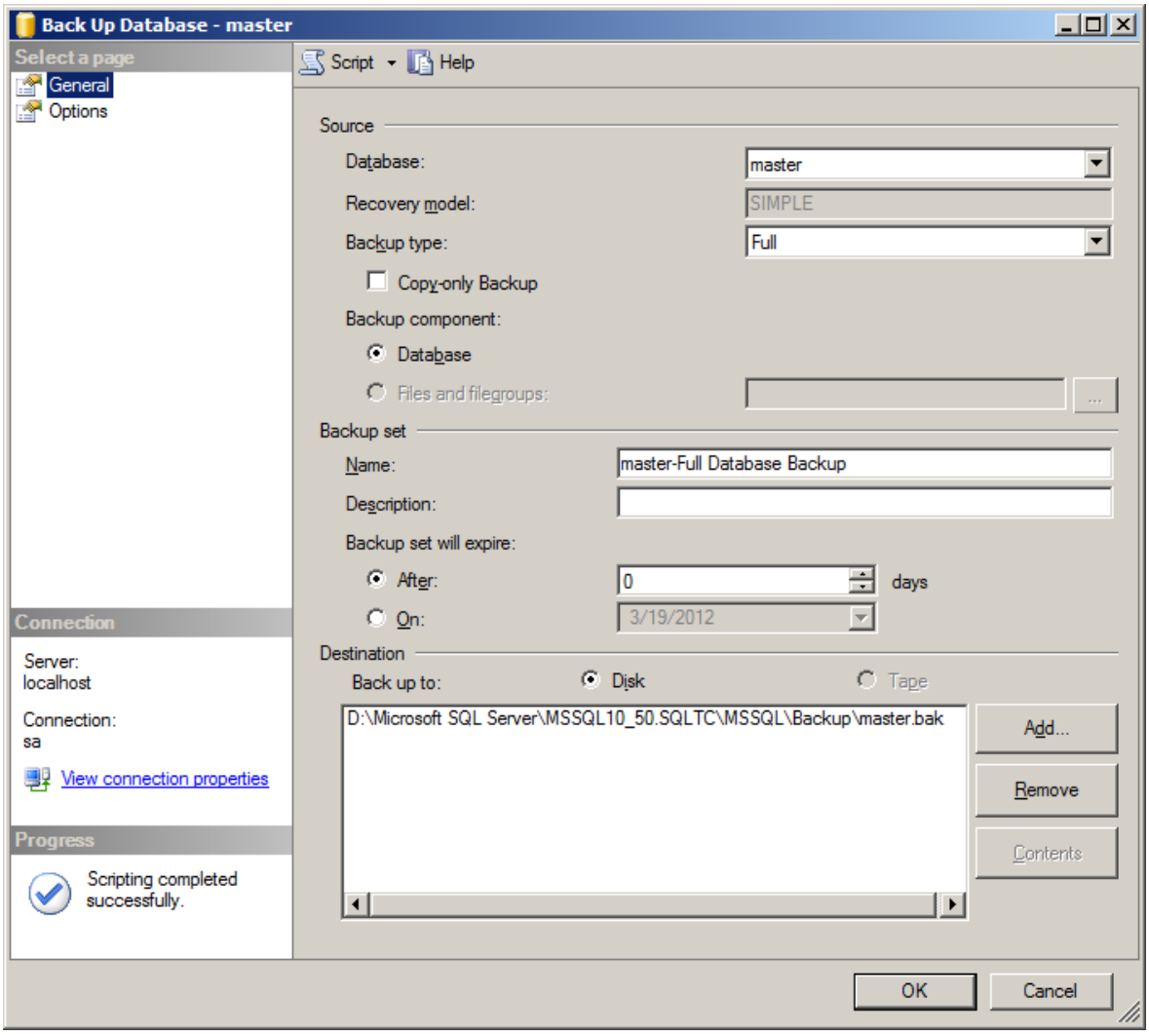
Hash se skládá ze 164bitu = 2na164tou možných kombinací.

13 ZÁLOHOVÁNÍ DAT SQL SERVERU POMOCÍ NÁSTROJŮ MS SQL SERVERU

Zálohování dat v databázích SQL serveru a to jak uživatelských, tak i systémových v prostředí MS SQL Serveru se provádí v konzoli Management Studia.

Zálohování pomocí nástrojů MS SQL serveru Management Studia

SQL Server Management studio pro verzi SQL Server 2008 R2 nabízí robustní nástroj v podobě grafického rozhraní, který obsahuje mimo jiné i zálohovací nástroje. Nástroj pro zálohování je zde obsažen, jak ve formě zálohovacího průvodce, kde velmi jednoduše provedeme plán záloh, tak i ke každé databázi máme možnost velice jednoduše provést zálohu.



Obrázek 23 – Plná záloha Master databáze

Vyscriptovaný příkaz plné zálohy master databáze pomocí transact SQL s verifikací zapsaných dat do souboru:

```
BACKUP DATABASE [master] TO  DISK = N'D:\Microsoft SQL
Server\MSSQL10_50.SQLDMZ\MSSQL\Backup\master.bak' WITH NOFORMAT, NOINIT,
NAME = N'master-Full Database Backup', SKIP, NOREWIND, NOUNLOAD,  STATS =
10
GO
declare @backupSetId as int
select @backupSetId = position from msdb..backupset where
database_name=N'master' and backup_set_id=(select max(backup_set_id) from
msdb..backupset where database_name=N'master' )
if @backupSetId is null begin raiserror(N'Verify failed. Backup
information for database ''master'' not found.', 16, 1) end
RESTORE VERIFYONLY FROM  DISK = N'D:\Microsoft SQL
Server\MSSQL10_50.SQLDMZ\MSSQL\Backup\master.bak' WITH  FILE =
@backupSetId,  NOUNLOAD,  NOREWIND
GO
```

Důležitou věcí, které SQL Server nabízí je vyscriptování řady příkazů a nastavení. Tím můžeme například hravě přenést řadu nastavení zálohování nebo samotných plánů záloh na jiný server, nebo pro potřeby zazálohovat si vlastní nastavení. Vyscriptované procesy, jsou v podobě textu a snadno editovatelné s jasně danou strukturou a kompatibilitou na všechny verze a edice MS SQL Serverů.

13.1 Konfigurace průvodce zálohováním Maintenance Plan Wizard

Maintenance plány jsou úlohy SQL serveru, které probíhají v určitý čas a spouští jasně dané transact sql skripty, kde můžeme jednoduše řetězit řadu procesů nad danou databází či více databázemi. Díky těmto plánům můžeme provádět následující operace:

- integrita databáze,
- shrink (odmazání) transakčního logu po full backupu databáze,
- reorganizaci indexů,
- rebuid (obnova) poškozených indexů,
- update statistics,
- odmazávání a hlídání stavu logu sql serveru,
- zálohy databází.

Provedeme tedy v maximální míře všechny procesy údržbou a zálohovaní databáze

Proces údržby a zálohování databáze je jednou z těch důležitých operací pro bezpečnost dat. Je třeba na bezpečnost dat pohlížet i tak, že nemusíme data ztratit jen útokem cizí osobou, nebo živelnou katastrofou, ale o data můžeme přijít i pokud se o ně nebudeme

starat a jednoduše se něco v tabulkách dostane do nepopsaného stavu a my nemáme žádný nástroj, kterým bychom data byli schopni znovu sestavit nebo opravit.

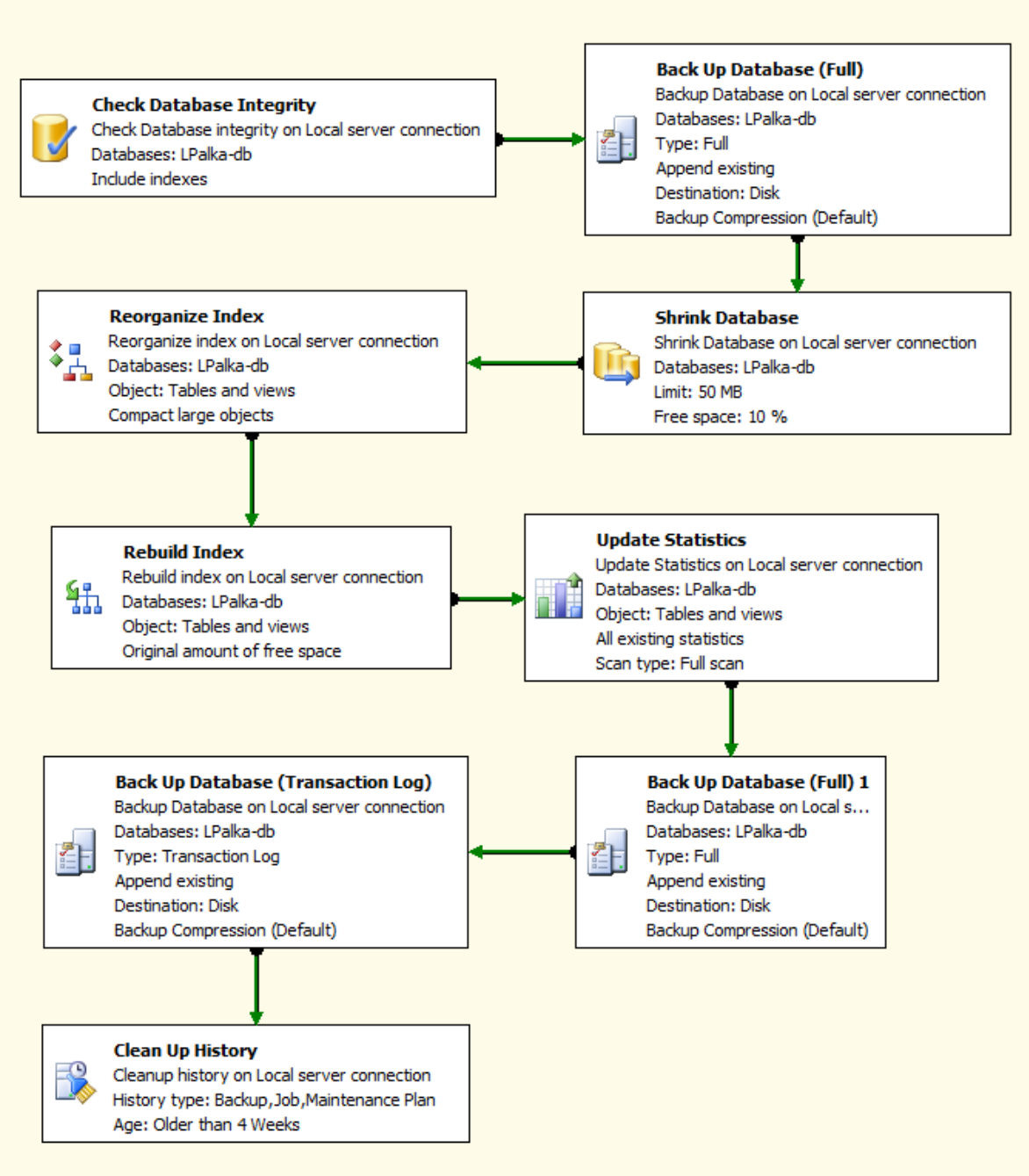
Pomocí průvodce nastavíme, jaké operace budeme nad databází provádět. Jako nejdůležitější věc považuji, v jakém pořadí operace budeme provádět. Pořadí procesu je klíčovou problematikou pro bezpečnost databáze, její údržby a bezpečnosti jejich dat.

1. Provedeme test na konzistenci dat, a zda je databáze v pořádku. To bychom měli provádět vždy na začátku všech údržeb a záloh.
2. Než se pustíme do zásahů nad databází, vždy provedeme plnou zálohu databáze. Je to jistota, pokud by se něco nepovedlo, nebo by náš návrh reindexace byl špatný a došlo tak ke zpomalení dolování dat, můžeme se vždy vrátit k původnímu stavu databáze.
3. Po plné záloze provedeme zmenšení transakčního logu pomocí shrink. V parametrech lze vybrat, na jakou velikost transakční log ořízneme a kolik z původní velikosti necháme alokovaný právě transakčnímu logu. Enormnímu růstu transakčního logu dochází například při upgradech tabulek, indexu a podobně.
4. V dalším kroku provedeme reorganizaci Indexů. Při této operaci se překlápějí indexy tak, aby čtení těchto indexů bylo co nejrychlejší a odezva na dotazy nad daty byla v co nejkratším čase. Obecně bychom tuto operaci mohli popsat jako defragmentaci záznamů nad rejstříky v knize. Reorganizování indexu způsobí přeskupení dat uvnitř vnějších datových stránek a také komprimaci indexu. Tato operace není výkonově příliš náročná, proto lze provést za plného provozu databáze online.
5. Asi nejdéle trvající operace, pokud máme robustní databázi s tisíci tabulkami a indexy, je sestaveni nových indexů zvané v MS SQL jako rebuild indexů. Zde se provádí operace, kdy zrušíme dané indexy a vytváříme je znovu. Tyto operace nedoporučuji provádět online, protože dostupnost dat bez indexů může způsobit výpadky a velmi pomalé odezvy při složitých operacích.

   Ve verzi MS SQL Serveru 2008 R2 lze provést online rebuild indexu. Před samotným spuštěním vytvoření nových indexů online, proběhne řada kroků před jeho spuštěním. Původní index je zachován pouze pro čtení a modifikaci dat zajišťuje funkce row versioning a tím je zaručena konzistence a rychlý přístup k datům. Nový index se vytvoří tak, že se tváří jako jeho předešlá verze. Všechny změny dat se v době rebuildu indexu zapisují jak do původního indexu tak

i do nově vytvářeného, který se v tuto dobu nepoužívá pro čtení, ale výhradně pro zápis. Protože oba indexy existují v jednu dobu, jsou operace nad diskovým prostorem dvojnásobné. V okamžiku, kdy se online rebuild indexu dokončí, všechny operace směřují do nového indexu a původní je po zpětné kontrole odstraněn.

6. Update statistics je velmi šikovný nástroj, který dokáže srovnat informace o tabulkách a indexech a díky těmto statistikám SQL server zvolí nejlepší plán pro vykonávání svých dotazů. Statistiky o datech se vedou na úrovni tabulek a indexů. Statistiky jako takové se vždy přetvářejí, když je vytvořen nový index. Jako velkou výhodu ve statistikách a jejího správného chodu je vytváření informací o sloupcích, tím získáme vyšší rychlost k přístupu k datům. V našem případě po rebuildu indexů zvolíme FULL SCAN statistics.
7. Tím, že jsme sestavili indexy, sestavili statistiky a všechny procesy nám dopadly v pořádku bez chyb, může proběhnout plná záloha databáze. Jelikož vše proběhlo v pořádku, víme, že data jsou v pořádku a můžeme provést zálohu db. Pokud bychom tuto operaci neprovedli a db nám zhavarovala, měli bychom stav db před provedenou údržbou dat a celou operaci, která může trvat několik hodin, bychom prováděli znovu.
8. V dalším kroku provedeme backup transakčních logů. Je to z důvodu, že veškeré operace, které jsme prováděli online a v průběhu údržby, které nám clienti vytvořili, se nám uložily do full backupu, ale v průběhu plné zálohy proběhly další změny a ty jednoduše a velice rychle uložíme. Operace oproti fullbackupu trvá minimálně dlouho a my jsme tímto posledním krokem získali kompletní zálohu pro daný čas společně s údržbou db.
9. V posledním kroku už jen smažeme log sql serveru starší 4 týdnů. Mažeme touto operací všechny logy SQL serveru týkající se chodu a autentikace vůči SQL, nejen záznamy o provedených zálohách. Pokud tedy máme dlouhodobý problém na SQL serveru, tuto operaci nebudeme provádět.



Obrázek 24 – MS SQL Maintenance plán Wizard

Jedná se tedy o robustní nástroj pro správu a bezpečnost databází.

Jelikož procesy, které nadefinujeme, se musí nějakým způsobem spustit, je potřeba mít vždy na straně sql serveru spuštěn SQL Server Agent. Tato utilita spouští plány a zapisuje reporty do logu sql serveru.

13.2 Obnova uživatelské (aplikační) databáze

Nejsnazším a nejvhodnějším nástrojem pro obnovu nesystémových databází pomocí nástrojů MS SQL serveru je použití management studia. Je to robustní nástroj jak pro kontrolu stavu databází, tak i obnovu dat. Při obnově databáze, nebo více uživatelských databází, postupujeme následovně.

Pokud se z nějakých důvodů stane, že zhavaruje nějaká databáze, nebo se databáze nějak poškodí a musíme použít nástroje k obnově dat, měli bychom ještě před obnovou získat co nejvíce informací, proč k havárii došlo.

Nemá například smysl obnovovat databázi na diskové úložiště, které chybuje. Proto vždy zkontrolujeme stav všech zařízení, kterých se obnova dotkne. Zkontrolujeme úložiště, switche a kabely, přes které dochází k přenosu informací.

Zkontrolujeme systémové logy z OS a logy ze zařízení, případně pokud sever má RSA kartu, tak i logy v ní. RSA karta poskytuje asi nejdokonalejší způsob, jak monitorovat konkrétní server a jeho stav.

Kontrola stavu databáze

1. Jako první, než spustíme obnovu databáze, zkontrolujeme stav msdb databáze. Tato systémová databáze obsahuje informace o všech zálohách a velice nám usnadní práci při obnově. Díky ní zjistíme, ke kterému datu máme poslední zálohu, pomůže nám s obnovou i při sestavení původních dat z transakčních logů a také systémově zpřístupní data z pásky nebo disku. Lze říci, že díky msdb databázi získáme informace o zařízeních a souborech, na nichž je uložena záloha.
2. Pokud je msdb databáze poškozena, obnovíme ji ještě před obnovou uživatelské databáze.

13.2.1 Příprava obnovy uživatelské databáze

Obnovit databázi do požadovaného stavu nese řadu kroků ještě před obnovou, které bychom si měli předem ujasnit, než se do obnovy databáze pustíme.

Pokud zjistíme, že je databáze poškozena, nikdy nepoužijeme rovnou obnovu databáze.

K takové databázi se chováme následovně:

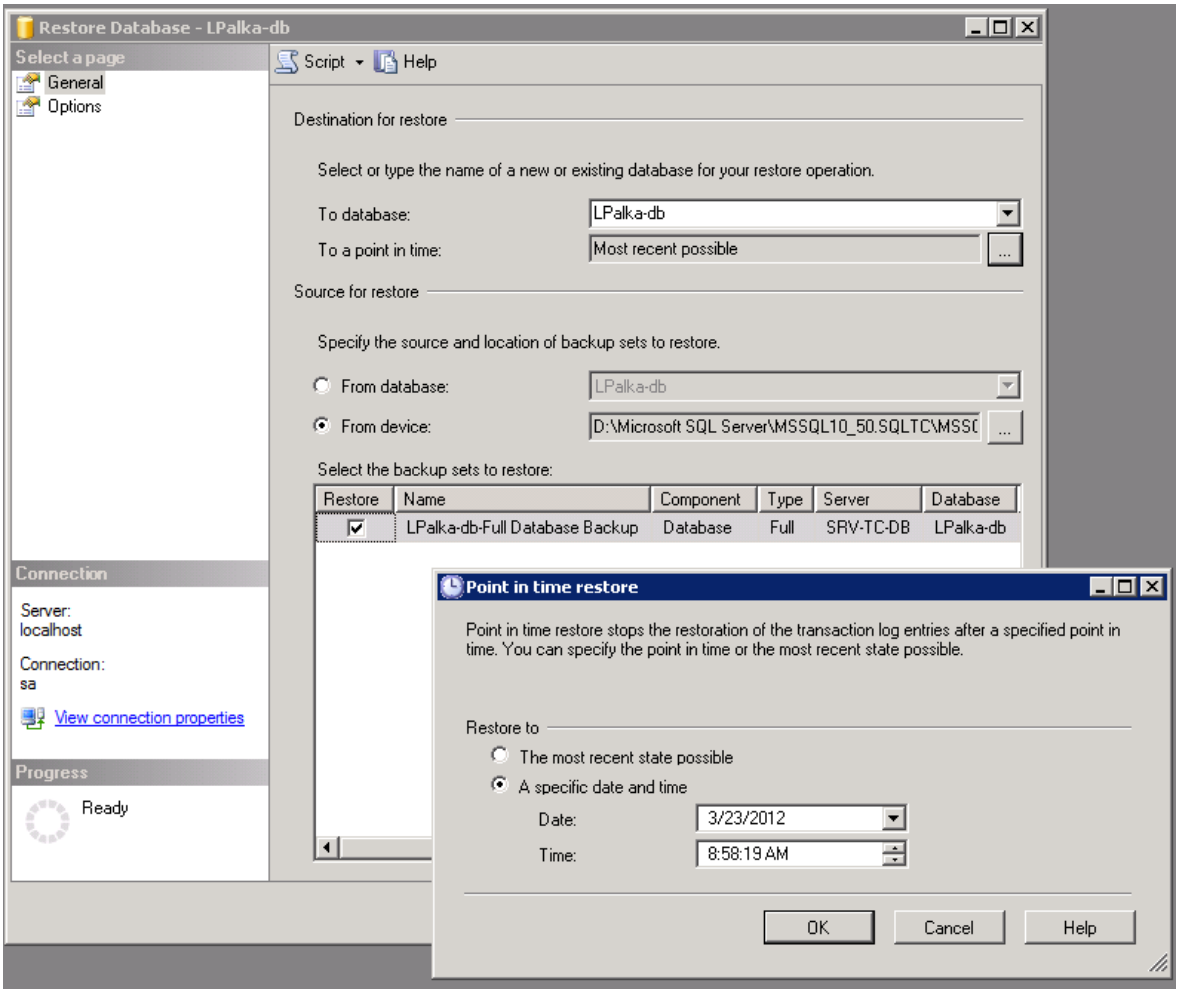
1. Zkontrolujeme, zda proběhl full backup dané databáze a zapíšeme si, o jakou zálohu šlo a její čas.
2. Spustíme backup transakčních logů dané databáze. Tím zajistíme návrat ke konzistentním datům těsně před havárií databáze (Je potřeba, aby vždy byl na všech databázích nastaven full recovery mod ve vlastnostech databáze).
3. Zajistíme, aby uživatelé nemohli přistupovat k naší obnovované databázi. Například toho docílíme tím, že dočasně znepřístupníme aplikační část služby, která přistupuje do databáze.
4. Pokud jsme zjistili příčinu chybu databáze, tak po zazálohování transakčního logu databáze můžeme tuto chybu opravit. Pokud jde například o externí úložiště, vždy opravujeme chybu po kompletní záloze všech databází. Proto databáze, které jsou v chybovém stavu, zazálohujeme pomocí transakčních záloh a databáze, které jsou v pořádku, zazálohujeme full backupem. Provedeme plnou zálohu i systémových databází. Nejdůležitější je pak databáze master. Tyto zálohy jsou velice potřebné pro další práci a nedostaneme se tak při dalších krocích do slepé uličky, pokud bychom zjistili nepředvídatelnou chybu.
5. Poté, co máme zálohy databází a následně po zálohách jsme opravili zjištěnou chybu, například díky chybě hardwaru, se můžeme pustit do obnovy dat.

Tím máme provedeny všechny důležité kroky před obnovou databáze.

13.2.2 Obnova uživatelské databáze

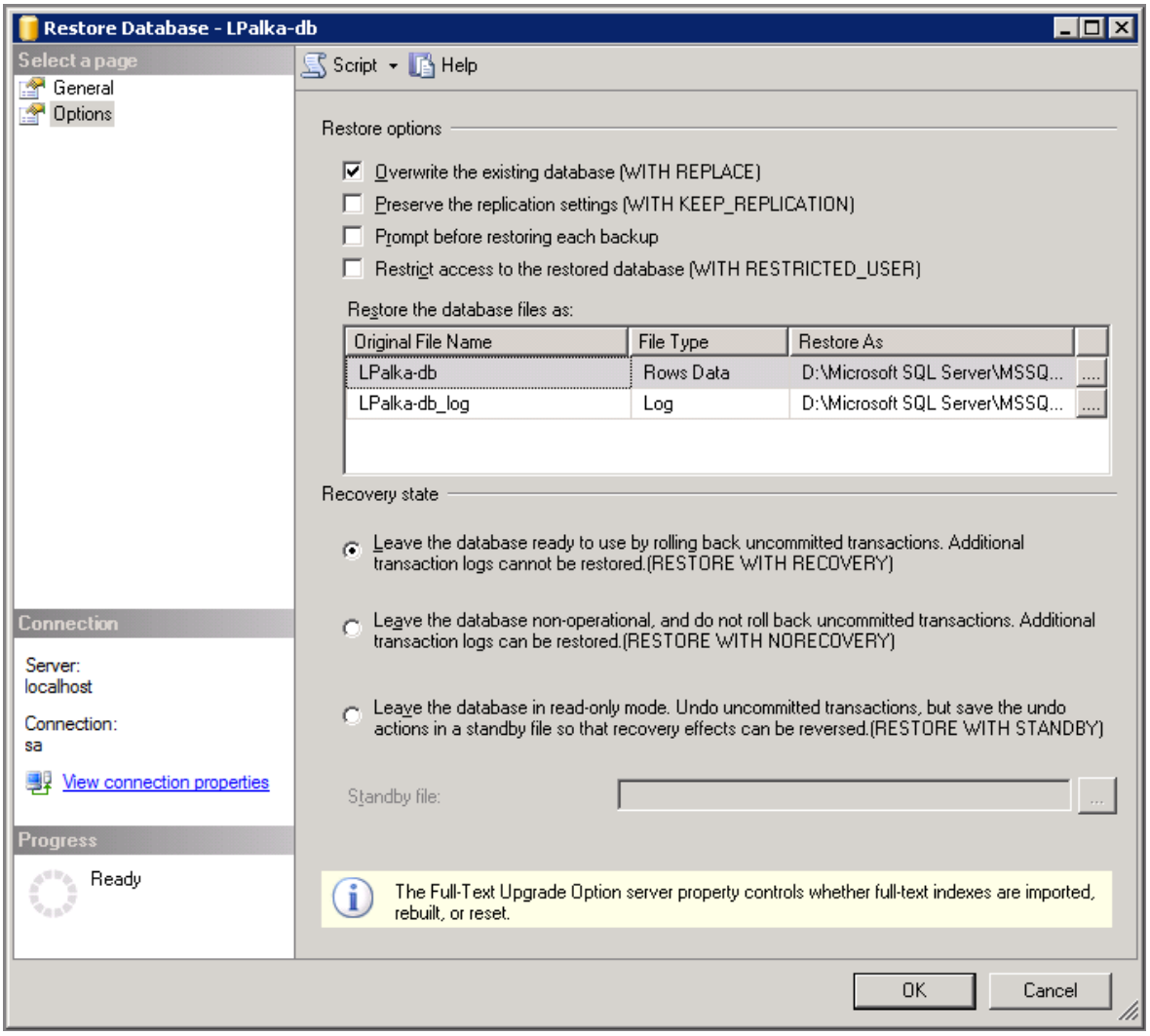
Obnova databáze se provede v následujících krocích. Provedeme obnovu uživatelské databáze. Systémové databáze se obnovují jinak, tu provedeme až nakonec naší kapitoly.

1. Pro databázi, kterou budeme obnovovat ze zálohy, zvolíme volbu Tasks – Restore – Database...
2. Nastavení pro obnovu databáze se nabízí ve dvou záložkách (General a Options).
3. Záložka General obsahuje název databáze, kterou chceme obnovit, nastavení Point in the time. Zde se můžeme plynule v čase, při režimu databáze FULL Recovery, posunout na konkrétní vteřinu a datum do minulosti ke stavu dat databáze. Dále zde můžeme vybrat umístění zálohy (soubor), ze kterého budeme obnovovat databázi.



Obrázek 25 – Obnova uživatelské databáze 1

4. Záložka Options obsahuje dva důležité údaje – cestu k fyzickým souborům databáze a to jak pro soubor databáze, tak k transakčnímu logu. V položce Restore options volíme, jak proběhne obnova dat do databáze, ve které jsou již data. V našem případě obnovujeme db plnou zálohou a nezajímají nás stávající data, proto zvolíme volbu Overvrite.



Obrázek 26 – Obnova uživatelské databáze 2

Obnova uživatelské databáze pomocí Transact SQL

Do Query zadáme následující přikaz:

```
RESTORE DATABASE [LPalka-db] FROM  DISK = N'D:\Microsoft SQL
Server\MSSQL10_50.SQLDMZ\MSSQL\Backup\lpalka-db.bak' WITH  FILE = 1,
NOUNLOAD,  REPLACE,  STATS = 10

GO
```

Obnova uživatelské databáze do jiné uživatelské databáze

Při obnově uživatelské databáze do jiné… k čemu nám to je dobré? V praxi se s tím můžeme setkat celkem často. Pokud chceme vyzkoušet, zda zásadní aktualizace databázové aplikace proběhne v pořádku a potřebujeme vše provést cvičně na zkušebních datech, v takovém případě obnovíme stávající ostrou db do jiné, kterou jinak pojmenujeme.

Důležitou vlastností MS SQL Serveru je to, že databáze si nese sebou oprávnění a práva daných uživatelů k datům. To znamená, pokud obnovujeme databázi, po obnově máme práva na data již nastavena tak jako v originálu a není třeba definovat přístupy do nové db znovu.

Obnova je velmi podobná té, při které obnovujeme data do stejné db.

1. Restore provádíme na db, kterou chceme mít jako kopii, tzn. pokud takovou db nemáme, vytvoříme si prázdnou.
2. Zadáme restore databáze a na záložce general vybereme volbu from device pro upřesnění zálohy. Potvrdíme restore db a přepíšeme název db na název cvičné databáze. Poté na záložce Options provedeme zadání cest k souborům na cvičnou databázi. Proto je vhodné vždy cvičnou db nazvat test, aby se nám nestalo, že přepíšeme např. transakční log ostré databáze. Pokud bychom se i toho obávali, je to velice jednoduché. Nejprve provedeme plnou zálohu ostré databáze a následně sestavujeme tu cvičnou. Tím máme zazálohován stav ostré db, a i když cokoliv pokazíme, vždy se máme kam vrátit.

Obnova uživatelské databáze do jiné instance SQL Serveru

Tato operace je velmi podobná obnově uživatelské databáze do jiné uživatelské databáze. Zde zachováme název db a provedeme kroky obdobně, tak jak jsou uvedeny výše.

13.3 Obnova master databáze

Jedná se o systémovou databázi, které nese veškeré informace o všech db, nastavení SQL serveru, účtech, heslech, přístupech a podobně. Pokud by nám zhavarovala tato databáze a my neměli zálohu, byli bychom ve velkých problémech, i když se nejedná o uživatelsky cenná nenahraditelná data. Proto zde popíši, jak takovou databázi obnovovat. Zálohu databáze master jsme provedli na začátku této kapitoly.

Než popíši obnovu master databáze, je potřeba si uvědomit, že operace rozhodně nemůže běžet online za chodu clientu.

1. Vypneme SQL Server a zapneme jej příkazem *SQLservr –m*. Tím nastartujeme SQL Server v secure modu.
2. Tento příkaz spustí SQL Server v command line jako aplikaci s výpisem chybových hlášek. Doporučuji zkontrolovat, zda mezi chybami není chyba řadiče disku nebo

diskového oddílu. Pokud ano, provedeme kontrolu těchto zařízení, abychom měli jistotu, že obnovujeme master db na spolehlivé úložiště.

3. Spustíme Query Analyzer a obnovíme následujícím příkazem master databázi. Musíme pamatovat na dvě věci. Master databázi vždy zálohujeme jako fullbackup a danou zálohu držíme samostatně mimo uživatelské databáze tak, abychom vždy měli snadnější přístup k této záloze a nebyla například pod režimem robotické knihovny, která je závislá na dalších systémech, které díky SQL serveru nemusí při havárii master databáze fungovat. Tím bychom se k master záloze nedostali a nejsme schopni problém opravit.
4. Příkaz pro restore master databáze:

   ```
   RESTORE DATABASE master FROM  MASTERFULLBACKUP
   ```

   V našem případě:

   ```
   RESTORE DATABASE master FROM  DISK = 'D:\Microsoft SQL
   Server\MSSQL10_50.SQLDMZ\MSSQL\Backup\MASTER.bak'
   ```

5. Po obnově master databáze, která poběží opravdu jen krátce, se služba ukončí s výpisem, že master db byla obnovena.
6. Nyní můžeme spustit SQL server běžným způsobem a provedeme kontrolu všech db následovně:
   - Zkontrolujeme účet SA, zda jsou přiřazena DBO oprávnění na uživatelské databáze.
   - Vybereme si náhodně uživatelskou databázi a zkontrolujeme přiřazení práv a rolí uživatelům. Nesmíme zapomenout, že i když jsou účty uloženy v uživatelských databázích, tak o hesla a loginy se stará master databáze.
   - Provedeme checkdb intergrity na všech databázích. Tím se ověří vazba uživatelských databází s master db.
   - Zkontrolujeme nastavení SQL serveru a zda jsou dostupné SQL eventy.
   - Zkontrolujeme k jaké havárii došlo ve všech dostupných log souborech, a to jak SQL serveru, tak i eventy OS. Pokud nedošlo k HW závadě, ale pouze k SW chybě provedeme kontrolu všech db pomocí Maintenance Planu.

     Pokud došlo k HW závadě, okamžitě vypneme službu SQL serveru a opravíme havárii.

14 ZÁLOHOVÁNÍ DAT SERVERU A DATABÁZÍ SQL SERVERU POMOCÍ NÁSTROJŮ TIVOLI STORAGE MANAGERU

Instalace služby TIVOLI Storage Manager (TSM)

Nejprve než začneme s instalací, je potřeba se podívat, zda budeme provádět novou instalaci upgrade nebo rekonfiguraci storage managera. Pokud provádíme upgrade nebo rekonfiguraci knihovny či serveru provádíme jen ty kroky, které jsou nezbytné pro změnu.

Výhody a přínosy použití TSM zálohování jsou popsány v tabulce 9 – Porovnání výhod TSM a SQL zálohování databází. Obecně lze říci, že TSM jako nástroj pro zálohování je považován za jeden z nejrobustnějších řešení s celou další podpůrnou infrastrukturou pro monitoring záloh, přímou podporu driverů pro páskové mechaniky, servery i IBM utility.

14.1 Návrh instalace TIVOLI Storage Manager

Před tím než instalaci spustíme, je vhodné si říci, co TSM potřebuje ke svému běhu, kde je vhodné jej nainstalovat, jaké HW prostředky budeme potřebovat a také, kde je vhodné zálohovací server umístit v informačním systému, který budeme zálohovat jako celek.

Backup server jednoznačně doporučuji instalovat jako samostatnou službu serveru. Součástí TSM, bez které služba nemůže běžet, je databázový server. Databázi rovněž doporučuji instalovat mimo backup server, pokud již máme v prostředí podporované databázové prostředky (například Microsoft SQL Server). Díky databázovému serveru zabezpečíme bezpečně data TSM serveru. Musíme si uvědomit, že bez databáze nejsme schopni uprovozovat Tivoli zálohování a pokud databázi ztratíme, nebo ji poškodíme, nejsme schopni obnovit žádná data z pásek, pokud není sestaven fullbackup úložiště s exportem db na pásky.

TSM server slouží mimo jiné i jako cache pro data a velikost této cache je vhodné dostatečně naddimenzovat.

Poslední služba, která je vhodná pro TSM Server, je začlenit server do domény. Je to vzhledem k údržbě a centrální správě serveru.

Jako vhodné řešení pro TSM server vychází samostatný server, nikoli virtualizovaný server. Je to z důvodu, že vmware či virtualizaci MS infrastrukturu lze rovněž zálohovat, a tím bychom si mohli tuto možnost velice zkomplikovat. Je to vzhledem ke startům

virtualizačního prostředí a velice špatně se pak řeší havárie prostředků, které zajišťují samotný běh zvirtualizovaných serverů.

Všeobecná IBM příručka instalace Tivoli Storage manageru [9]

V dokumentaci k TSM službě, je řada odpovědí jak instalovat obecně službu TSM, ovšem z praxe se domnívám, že lze tato příručka použít pouze jako obecný postup s propracovanými kroky postupu týkající se řešení problému a ne samotné instalace.

14.2 Instalace a konfigurace OS Windows pro službu TSM

Systémová Lokalizace v jazyce *EN*

Vypnutí *UAC* (Pokud budeme zálohovat VMWARE \ kvůli mountum z pole)

spustíme diskpart a vypneme automount disable

založíme účty: *TSM /heslo*, ve skupinách *DB2ADMNS, DB2USERS, ADMINISTRATORS*

Přihlásíme se pod TSM účtem a provedeme z instalačního CD

INSTALCD: SERVER, DEVDRV,

Doinstalujeme Licence z base package

14.3 Instalace TSM

Instalaci lze provést pomoci grafického průvodce, nebo z příkazové řádky, nebo spuštěním tiché instalace.

Grafickou instalaci spustíme z CD, v našem případě Storage Manager for Windows 6.2

install.exe -i console -r C:\response.rsp (parametr c:\response.rsp uloží do souboru nastavení a log z průběhu instalace).

Průvodce instalace nás vyzve k zadání parametrů

- Výběr jazyka.
- Akceptace licenčního ujednání (En).
- Následně je seznam komponent k instalaci (server, languages, licenses, device driver, storage agent).
- Pokud vybereme některou ze serverových komponent je automaticky nainstalováno API rozhraní, DB2 verze 9.7 a GSKit 7.

- Pokud již na serveru není nainstalována předchozí verze TSM, jsme vyzvání k zadání umístění instalace.
- Jako poslední obrazovka průvodce instalací je souhrn nastavení. Poté se zahájí kopírování souborů, zápis konfigurace do registru a nastavení aplikace jako služba windows.

Tichá instalace TSM

Tichá instalace má význam, pokud chceme konfigurovat více zálohovacích serverů na vzdálených lokalitách.

Je dobré si uvědomit, že TSM je vhodné pro zálohování i ze sekundárních serverových míst a páskovou mechaniku je vhodné separovat do samostatné lokality.

Tichá instalace se spouští parametrem:

install.exe -i silent -DLICENSE_ACCEPTED=true -DUSER_INSTALL_DIR=install_dir
-DINSTALL_SERVER=1 -DINSTALL_SERVER_LANGUAGES=1
-DINSTALL_LICENSE=1 -DINSTALL_DEVICES=1
-DINSTALL_STAGENT=1

Tichá instalace TSM:

install.exe -i silent -DLICENSE_ACCEPTED=true -DUSER_INSTALL_DIR=C:\TSM
-DINSTALL_SERVER=1 -DINSTALL_SERVER_LANGUAGES=1
-DINSTALL_ENGLISH=1 -DINSTALL_LICENSE=1

Tichá instalace pomocí konfiguračního souboru:

install.exe -i silent -DLICENSE_ACCEPTED=true -f konfiguraceTSM.rsp

14.4 Konfigurace TSM Serveru

Jako první, než zkonfigurujeme TSM klienta, předpřipravíme si úložiště dat pomocí použití příkazové řádky TSM-DS2:

db2cmd (spustíme tento shell):
vyrobíme adresáře:

- *mkdir D:\TSM*
- *mkdir C:\TSM*
- *mkdir C:\TSM\DB*
- *mkdir C:\TSM\LOG*
- *mkdir D:\TSM\LOG*
- *mkdir C:\TSM\ARCLOG*

cd /D C:\tsm

14.4.1 Vytvoření User ID pro službu TSM

Vytvoříme účet, pod kterým bude služba TSM nabíhat. Účet musí být členem administrators s právy zápisu do všech adresářů, které jsme v předchozím kroku vytvářeli. Účtu zadáme dostatečně silné heslo a zrušíme vypršení hesla. Tím by nám po určité době přestalo zálohování fungovat.

Vytvoření účtu pomocí příkazové řádky: *net user user_ID * /add* (user_ID = jméno účtu)

Přidáme uživatele do skupin OS (groups):

net localgroup Administrators user_ID /add

net localgroup DB2ADMNS user_ID /add

net localgroup DB2USERS user_ID /add

14.4.2 Konfigurace TSM pomocí průvodce

Předtím, než se dáme do konfigurace je třeba mít dokončeno následující. Proces kontroly ověří, zda jsou provedeny následující kroky:

- musí být nainstalována služba TSM,
- vytvořená DB2 databáze,
- předchystány adresáře pro TSM,
- uživatel a jeho ID pro instanci serveru TSM.

1. Kontrola komunikace TSM

- Je zapotřebí, aby byla povolena SSH komunikace na firewallu serveru, na kterém instalujeme TSM.
- Povolit nástroj řízení uživatelských účtů.
- Povolit SMB protokol. Protokol pro používání sdílení souborů a tiskáren (port 445 povoleno out a in komunikace).

Pokud máme vše povoleno, vyzkoušíme přihlášení do systému pod uživatelem, pod kterým poběží služba TSM. Jedná se o účet pro službu TSM

2. Dokončení konfigurace TSM pomocí management konzole

Spustíme z adresáře c:\Program Files\Tivoli\TSM soubor dsmicfgx.exe, tím vyvoláme dokončení konfigurace. Tato operace si vyžádá restart služby TSM. Po spuštění služby nám naběhne aplikace, kde provedeme další důležité nastavení pro zálohování.

Spustíme aplikaci Start – Programs – Tivoli Storage Manager – Management Console.

Zvolíme položku Start, kde se nám spustí po inicializaci a ověření služby průvodce, kde provedeme dokončení konfigurace zálohování. Každá část průvodce si nese řadu nastavení, a proto je popíšeme v samostatných kapitolách.

14.5 Kapitoly zálohování po-instalační TSM konfiguraci

Zvolíme výchozí (default settings) a zadáme uživatele s právy pro službu TSM. Pokud by jsme tohoto uživatele nezadali, služba TSM by nemohla běžet, totiž se vůbec spustit.

A) Povolíme (Enable the database manager and configure the TSM).

B) Následně spustíme Storage Manager Tivoli jako instalaci serveru.

C) Na další obrazovce nahrajeme licence k TSM do správy licencí.

D) Nastavíme systém pro zálohování své databáze, kde se uchovává všechna nastavení a indexy na pásku k zálohovaným datům.

E) Nastavíme monitoring zálohování.

14.6 A) Zálohování a konfigurace TSM Databáze Manager

Než provedeme nastavení záloh databáze TSM, popíši: „Proč právě tahle data je důležité zálohovat?“.

Databáze TSM obsahuje dvě klíčové informace:

- První z nich je nastavení TSM serveru, její konfigurace a také seznamy procesy (joby), neboli co se kdy a jak má pravidelně vykonávat a co má služba TSM provést.
- Druhou informaci, který je obsažena v databázi TSM, je seznam indexu, atributu a odkazů na zálohovaná data, které ukládáme na pásky. Data, která jsou ukládána na pásku, přes páskovou mechaniku, kterou řídí právě TSM a její databáze jsou indexovány v TSM databázi. Pouze databáze nese informaci, kde konkrétní soubor (obecně data zálohování) jsou na pásce a kolik verzí těchto dat uchováváme. Obecně můžeme říci, že databáze TSM je v podstatě alokační tabulka pro data na pásce. Pokud chceme cokoliv dělat s páskou, vždy se vše řídí pomocí TSM Managera a ten veškeré informace ukládá, čte a mění v databázi TSM.

  Právě pro tyto důvody je velmi důležité tato data chránit. Bez těchto informací nejsme téměř schopni odzálohovat data serverů, které centrálně zálohujeme. Tím je myšleno, že jsme schopni obnovit soubory z pásky, nejsme schopni sestavit strukturu dat,

nevíme jaký soubor jsme obnovili a z jakého je data, a pokud zálohujeme více verzí stejných dat, nevíme, které jsou ty novější, případně ty, co opravdu chceme.

A v neposlední řadě, TSM umí i šifrovat své zálohy na páskách a bez databáze jsou tyto pásky i se znalosti šifrovaného hesla, kterým byly pásky zašifrovány, naprosto nečitelné.

14.6.1 Vytvoření a instalace databáze pro Tivoli Storage Manager (TSM)

- Vytvoříme soubor *tsmdbmgr.env* v adresáři *C:\TSM* a do něj zapíšeme následující konfiguraci:

  DSMI_CONFIG=C:\tsm\tsmdbmgr.opt

  DSMI_LOG=C:\tsm

- Spustíme příkaz cmd a následně *db2cmd* a provedeme jeho inicializaci:
  Db2set –i SERVER1 DB2_VENDOR_INI = c:\TSM\ tsmdbmgr.env

  (pokud nenalezneme příkaz db2cmd z cmd, cesta k němu je:

  C:\Program Files\Tivoli\TSM\db2\bin

- V dalším kroku vytvoříme soubor *tsmdbmgr.opt* v adresáři *C:\TSM* a do nej vypíšeme:

  nodename $$_TSMDBMGR_$$

  commmethod tcpip

  tcpserveraddr localhost

  tcpport 1500

  passwordaccess generate

  errorlogname c:\tsm\TSMDBMGR.log

- Restart databáze:

  Spustíme opět *db2cmd*

  Inicializujeme instanci databáze TSM: *set db2instance=SERVER1*

  Zadáme postupně příkazy: *db2stop* a *db2start* a zavřeme okno db2cmd

- Spustíme z *cmd* obrazovky příkaz:

  "c:\program files\tivoli\tsm\server\dsmsutil.exe"

  UPDATEPW /NODE:$$_TSMDBMGR_$$ /PASSWORD:TSMDBMGR /VALIDATE:NO /OPTFILE:

  " C:\TSM\tsmdbmgr.opt"

  Tím je proces vytvoření databáze pro TSM dokončen. Název instance je tedy SERVER1 a databáze pro TSM je nasazena.

  Provedeme ještě inicializaci env db2 TSM databáze:

  db2set -i server1 DB2_VENDOR_INI=C:\tsm\tsmdbmgr.env

Založení služby TSM Server1:

cd /d C:\tsm

"c:\program files\tivoli\tsm\console\install" "TSM Server1"

"C:\ProgramFiles\Tivoli\TSM\server\dsmsvc.exe" .\tsm IBM.1234

14.7 A2) Zálohování a konfigurace TSM Databáze Manageru - Ruční konfigurace

V následující části provedeme konfiguraci databáze služby TSM, vytvoříme její konfigurační soubor, provedeme následně import databáze a její inicializaci do systému.

Konfigurace databáze TSM:

db2icrt -u tsm SERVER1

set db2instance=server1

db2 update dbm cfg using dftdbpath C:

db2set -i server1 DB2CODEPAGE=819

Vytvoříme c:\tsm\dsmserv.opt:

DEVCONFIG C:\tsm\devconfig.out

VOLUMEHISTORY C:\tsm\volhist.out

DEVCONFIG D:\tsm\devconfig.out

VOLUMEHISTORY D:\tsm\volhist.out

DEVCONFIG N:\tsm\devconfig.out

VOLUMEHISTORY N:\tsm\volhist.out

CD /D c:\tsm

"c:\program files\tivoli\tsm\server\dsmserv" -k SERVER1 LOADformat dbdir=C:\TSM\DB activelogdir=D:\tsm\LOG archlogdir=C:\tsm\ARCLOG mirrorlogdir=C:\tsm\LOG activelogsize=2048

Naimportovat databázi *(export příkaz je v kapitole D) DB-TSM)*

"c:\Program Files\Tivoli\TSM\server\dsmserv" insertdb manifest=d:\manifest.txt 1>>D:\insert.out 2>&1

Vyrobit c:\tsm\tsmdbmgr.env:

DSMI_CONFIG=C:\tsm\tsmdbmgr.opt

DSMI_LOG=C:\tsm

Vyrobit D:\tsm\tsmdbmgr.opt:

nodename $$_TSMDBMGR_$$
commmethod tcpip
tcpserveraddr localhost
tcpport 1500
passwordaccess generate
errorlogname c:\tsm\TSMDBMGR.log

Inicializace env db2:

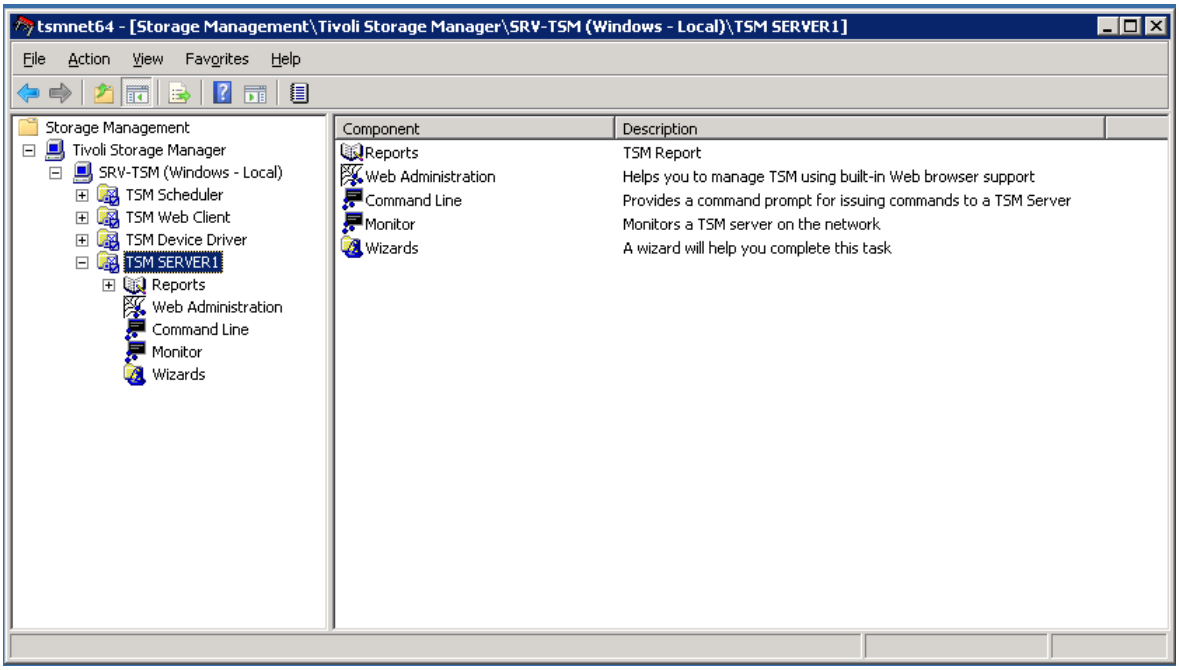
db2set -i server1 DB2_VENDOR_INI=C:\tsm\tsmdbmgr.env

14.8 B) Spuštění instance serveru TSM

Spuštění služby lze provést z adresáře *C:\Program Files\Tivoli\TSM*. Spustíme příkaz *cmd* a příkazem spustíme TSM *dsmserv -k SERVER1.*

Pokud se instance nespustí je potřeba zkontrolovat:

- Oprávnění OS na spuštění dsmserv.exe, zda máme práva full control.
- Zkontrolujeme, že jsme ve všech konfiguračních souborech nastavili server_instance = SERVER1. Jedná se o výchozí název Tivoli Storage Manager serveru.
- Zkontrolujeme verzi OS a instalovaný TSM, zda sedí 32 a 64b. verze. Pokud bychom omylem nainstalovali špatnou verzi, objeví se chybová hláška s kódem 216 error.



Obrázek 27 – TSM Server Storage Manager

Spuštění TSM jako služba OS pomocí TSM management Console

Na obrázku běžící konzole služby TSM pokud zvolíme vlastnosti TSM SERVER1, lze nastavit, že aplikace TSM poběží na serveru jako služba. Povinné parametry pro nadefinování služby TSM:

- Zadat pod jakým účtem služba poběží.
- Dalším důležitým parametrem je, zda služba poběží automaticky nebo bude spouštěna ručně,
- zadání výstupního log souboru *console.log.*

Spuštění TSM jako služba OS pomocí OS Windows Services

V sekci services na záložce **nástroje pro správu** vybereme službu ze seznamu a zadáme následující parametry.

Ve vlastnosti služby uvedeme účet, pod kterým služba poběží, heslo k danému účtu (platnost hesla takového účtu by měla být bez časové expirace). Dále vybereme spuštění služby automatické nebo manuální.

Tím je služba nastavena a záznamy o stavu služby a její události najdeme v Aplikačním Event Windows logu.

14.8.1 Zastavení serveru TSM

Pokud chceme zastavit službu TSM, provedeme to příkazem Halt z Tivoli Storage Manageru.

Důležitou vlastnosti TSM je to, že databáze běží nezávisle na Managementu, takže po zastavení serveru běží stále její databáze.

14.9 C) Licencování

Po zakoupení licencí Tivoli Storage Managementu dostaneme licenční ujednání a soubor s ověřenou pravostí a jedinečným klíčem k aktivaci. Aktivaci provedeme spuštěním příkazu:
REGISTER LICENSE v managementu TSM.

14.10 D) Konfigurace zálohování TSM databáze

Jedná se o zálohy databáze, která je čistě pro potřeby TSM Serveru. Co je obsahem této databáze jsem již napsal v kapitole A) Zálohování a konfigurace TSM Databáze Manager.

Pokud chceme připravit systém pro automatické a manuální zálohy TSM databáze, musíme určit DEVICE CLASS, kterou budeme používat. DEVICE CLASS je třída zařízení pro zálohování.

Před zahájením konfigurace je třeba určit páskovou knihovnu (pásku) nebo jiné zařízení pro zálohy (disk). Tuhle operaci provedeme příkazem *SET DBRECOVERY* s parametrem, kde určíme zařízení pro zálohování databáze TSM.

Instalační postup

Spustíme aplikaci Tivoli Storage Manager a v ní příkazový řádek.

Obecný tvar příkazu set dbrecovery:

set dbrecovery device_class_name

V naši konfiguraci spustíme příkaz:

SET DBRECOVERY DBTODISK

Tím jsme úspěšně nastavili zálohy dbTSM. Nyní se služba TSM automaticky postará o její zálohy a my můžeme pokračovat v konfiguraci monitorování služby TSM.

Pokud chceme provést export DB na pásku z db2cmd provedeme to příkazem:

```
cd /d "C:\Program Files\Tivoli\TSM\server1"
..\upgrade\dsmupgrd preparedb devc=lib1 >d:\preparedb.out 2>&1
..\upgrade\dsmupgrd extractdb devc=lib1 manifest=d:\manifest.txt >d:\extract.out 2>&1
```

Import databáze se pak provádí příkazem:

```
"c:\Program Files\Tivoli\TSM\server\dsmserv" insertdb manifest=d:\manifest.txt 1>>D:\insert.out 2>&1
```

Tento příkaz můžeme použít, i pokud budeme povyšovat TSM na vyšší verzi, kdy potřebujeme vyexportovat (zazálohovat) databázi na samostatnou pásku.

14.11 E) Monitoring zálohovacího serveru Tivoli

Služba zálohování má své režijní a jiné provozní potřeby a to jak na HW tak i SW. Bez důkladného sledování v jakém stavu se služba nachází, zda nedochází k nějakým kolizím, musíme službu TSM sledovat. Jedině tak máme jistotu, že služba zálohuje. Informace, které budeme sledovat.

Aktive log také obsahuje informace týkající se úložiště a jeho volného prostoru. Informace, které získáme z aktive logu:

- Velikost úložiště je správně nastavena vzhledem k plánům záloh. Pokud se nám z dlouhodobého hlediska zmenší volný prostor pod úroveň 20%, je nutné problém řešit.
- Informace a charakteristiky průběhu záloh.

Pokud TSM zálohuje databáze, je to jeden obrovský soubor, který se skrývá pod jednou transakcí a potvrzení dokončení transakce může trvat i několik desítek minut, nebo hodin. Pokud naopak zálohujeme soubory, zálohujeme stovky tisíc malých souborů, kde každý soubor je jedna transakce. Proto zde máme několik překrývajících se transakcí a každá z nich si vyžaduje potvrzení a tím velikost aktive logu extrémně roste.

Mail server může mít povahu dat jako soubory, nebo jako jedna databáze. I nové technologie mail serveru, jako Kerio Mail Server, obsahují data ve formě souborů, naopak Microsoft Exchange Server je zálohována jako databáze.

- Informace o rychlostech přenosu dat mezi serverem, který zálohujeme, respektive jeho data a serverem TSM.

Aktive log jako takový je jedna z kritických věcí, která může ochromit zálohovací server, pokud bychom o něj nějakým způsobem přišli, a nebo by serveru došel datový prostor k jeho uložení.

Aktivity log se zaplňuje velice nerovnoměrně a proto je potřeba hned na samotném začátku požadavek na úložiště naddimenzovat.

Jak udržet aktivity log TSM v rozumné míře je nastavením režimu záloh hned od počátku spuštění služby TSM. Plnou zálohou databáze TSM docílíme i odmazání aktivity logu. Tím jsme schopni udržet log aktivit v rozumné velikosti a s maximální mírou zabezpečení.

Zálohu TSM databáze je proto potřeba zálohovat minimálně jednou denně. Pokud proces záloh neběží například ve večerních hodinách, kdy v podniku je systémový klid, zálohu TSM databáze provádějme aspoň 2x denně. Vždy by doba záloh měla reflektovat stav, kdy v podniku není maximální systémový provoz, ale spíše rutinní provoz. Pokud špatně načasujeme proces zálohování, můžeme se dostat do situace, že zálohy dat sice máme, ale jsou vzhledem k podnikovým návykům ne zrovna použitelné. Takové zálohy můžeme nazvat slepé, protože nezohledňuji například právě probíhající aktualizace uživatelských dat, systémové aktualizace aplikačních databází a podobně.

Tím jsme nastavili zálohy TSM databáze a zbývá zkonfigurovat páskovou mechaniku a obslužné služby TSM, vše provedeme z db2cmd příkazové řádky.

14.12 Konfigurace páskové mechaniky a obslužné služby TSM

Spustíme server TSM v interaktivním módu:
"c:\program files\tivoli\tsm\server\dsmserv" a provedeme registraci licencí a nastavíme účet pro Tivoli monitoring (ITM):

```
setopt MAXSESSIONS 1000
setopt NumOpenVolsAllowed 256
reg lic file=tsmee.lic
regi admin ITM ITM
```

Ověříme, zda TSM klient komunikuje s DBTSM

Provedeme přihlášení pomocí backup TSM db klienta:

```
dsmc q ses -optfile=C:\tsm\tsmdbmgr.opt
prvni radek odklepnout
zadat heslo tsmdbmgr
```

Zrestartujeme TSM jako službu:

1. Shodit běžící TSM server příkazem *HALT.*

2. Dále nahodíme službu TSM Server1 (Net Start "TSM Server1"), lze i přes správce služeb OS.

Nyní můžeme konečně nastavit ty servery do TSM, které chceme zálohovat. Přidáme je proto jako nové noody pro zálohování. (db2cmd):

```
upda node * maxnumm=7
regi node SRV-TSM      heslo backdel=YES maxnumm=7 DOMAIN=STANDARD
regi node SRV-AD1      heslo backdel=YES maxnumm=7 DOMAIN=STANDARD
regi node SRV-AD2      heslo backdel=YES maxnumm=7 DOMAIN=STANDARD
regi node SRV-CA       heslo backdel=YES maxnumm=7 DOMAIN=STANDARD
regi node SRV-MIS      heslo backdel=YES maxnumm=7 DOMAIN=STANDARD
regi node SRV-SQL1     heslo backdel=YES maxnumm=7 DOMAIN=STANDARD
regi node SRV-SQL2     heslo backdel=YES maxnumm=7 DOMAIN=STANDARD
```

Nadefinujeme páskovou knihovnu, na kterou budeme ukládat zálohy

V našem případě zálohovací knihovna s dvěma páskovými hlavami a s automatickým robotem na výměnu pásek. Pásky LTO3 – kapacita 250GB.

```
DEFI LIBRA LIB1 LIBTYPE=SCSI autolabel=yes
DEFI path tsm1 lib1 srct=server destt=libra device=lb0.1.0.3
DEFI DRI LIB1 DRV1.1 CLEANFREQ=NONE
DEFI DRI LIB1 DRV1.2 CLEANFREQ=NONE
DEFI DRI LIB1 DRV1.3 CLEANFREQ=NONE
DEFI DRI LIB1 DRV1.4 CLEANFREQ=NONE
DEFI PATH tsm1 drv1.1 srct=server destt=drive libra=lib1 device=mt0.0.0.3
DEFI PATH tsm1 drv1.2 srct=server destt=drive libra=lib1 device=mt1.0.0.4
```

Načteme inventory:

```
checkin libvol lib1 search=yes checklabel=barcode status=SCRATCH
checkin libvol lib1 search=yes checklabel=barcode status=PRIVATE
```

Nyní páskovou mechaniku máme na TSM serveru nadefinovanou. Komunikace páskové mechaniky a TSM serveru se děje pomocí komunikaci na úrovní sítě LAN. Přenášená data pak lze přenášet dvěma způsoby:

- Data přenášíme přes LAN prvky v síti.
- Data přenášíme pomocí optických kabelů, kde pásková mechanika je připojena přímo do optického switche, stejně jako úložiště (pole) s daty serverů.

Nyní nadefinujeme DEVCLASSy

DEVCLASSy jsou místa, kam se budou ukládat data z TSM a to jak databáze TSM, tak i zálohovaná data z ostatních serverů

Vše provedeme opět v db2cmd konzoli TSM:

mkdir N:\TSM\ARC
mkdir N:\TSM\TSMDBB
mkdir N:\TSM\DR
mkdir N:\TSM\HSM
mkdir N:\TSM\MX
mkdir N:\TSM\PC
mkdir N:\TSM\BAC
mkdir D:\TSM\SQL

Přenastavímě úložiště pro TSM DB:

UPD DEVC DBTODISK dir=N:\TSM\TSMDBB
SET DBRECOVERY DBTODISK

Provedeme test zálohy DB:

backup db devc=dbtodisk t=f

Provedeme přidání definic pro uložení zálohovaných dat:

def devc sqlfile dir=D:\TSM\SQL mountl=200
def devc COPYNAS1 dir=N:\TSM\DR mountl=200
def DEVC HSMFILE dir=N:\TSM\HSM mountl=200
def DEVC MXFILE dir=N:\TSM\MX mountl=200
def DEVC PCFILE dir=N:\TSM\PC mountl=200
def DEVC arcFILE dir=N:\TSM\ARC mountl=200 devt=file maxcapa=50G
def devc bacfile dir=N:\TSM\BAC mountl=200 devt=file maxcapa=50G

A konfiguraci kvót pro ukládané zálohy:

defi stg mxfile maxscr=100 coll=no dedup=yes high=99 lo=99 next=''
defi stg pcfile maxscr=100 coll=no dedup=yes
defi stg sqlfile maxscr=100 high=99 lo=99 coll=no dedup=yes MAXSIZE=20G
defi stg bactape lib1 maxscr=200 high=80 low=70 coll=no
defi stg bacfile bacfile maxscr=200 high=99 low=99 coll=no dedup=yes next=''
dele stg arcfile
defi stg arcfile arcfile maxscr=100 high=99 low=99 coll=no dedup=yes crcd=yes
defi stg arccopy lib1 coll=no poolt=copy crcd=yes maxscr=100

Tím máme nadefinovány procesy pro ukládání dat a vytvořeny kvóty pro zálohy ze zálohovaných serverů.

Nastavení politik archivace

Nyní nastavíme politiky, kdy a jak se která data budou zálohovat. Z povahy dat lze nastavit, která data se budou zálohovat častěji, která se nebudou zálohovat vůbec, která budou mít více verzí a jak historicky budeme udržovat zálohy. Nastavení politik archivace je klíčovou rolí pro zálohovací server a podle této politiky pořizujeme zálohovací zařízení tak, aby bylo vyhovující, a data jsme opravdu bezpečně uchovávali. Politika hesel jasně udává, která data zálohujeme a definici exspiračních pravidel.

Expirace zálohovaných dat v TSM

- Expirace dat není závislá na úložném mediu. To je důležité zejména u páskových medií, většina jiných systémů exspiruje data z pásek přepisem fullbackup zálohy.
- TSM dokáže exspirovat i jednotlivé soubory na páskách bez nutnosti opakování fullbackupu.

Nastavili jsme politiky pro BACKUP s exspirací zálohovaných dat:
14 dni, 1 měsíc, 1 rok, 5 let, navěky a politika "Standard"

Politiku "standard" nastavíme na defaultní hodnotu exspirace 31 dní. Tuto politiku použijeme globálně pro všechny zálohy, pokud nebude pro specifická schválena jiná politika.

Archivní politiky nastavíme také na: 14 dni, 1 měsíc, 1 rok, 5 let, navěky. Výchozí politika "Standard" ale bude přednastavena s neomezenou trvanlivostí, tedy navěky.

upda copy standard standard standard dest=bacfile
upda copy standard standard 14d dest=bacfile
upda copy standard standard 14d-dyn dest=bacfile
upda copy standard standard 1y dest=bacfile
upda copy standard standard 5y dest=bacfile
upda copy standard standard dyn dest=bacfile
upda copy standard standard forever dest=bacfile
upda copy standard standard MX dest=MXfile

upda copy standard standard standard dest=arcfile t=a
upda copy standard standard 14d dest=arcfile t=a

upda copy standard standard 14d-dyn dest=arcfile t=a
upda copy standard standard 1y dest=arcfile t=a
upda copy standard standard 5y dest=arcfile t=a
upda copy standard standard dyn dest=arcfile t=a
upda copy standard standard forever dest=arcfile t=a

acti pol standard standard

upda copy PC standard standard dest=PCfile
upda copy PC standard 14d dest=pcfile
upda copy PC standard 14d-dyn dest=pcfile
upda copy PC standard 1y dest=pcfile
upda copy PC standard 5y dest=pcfile
upda copy PC standard dyn dest=pcfile
upda copy PC standard forever dest=pcfile

upda copy PC standard standard dest=arcfile t=a
upda copy PC standard 14d dest=arcfile t=a
upda copy PC standard 14d-dyn dest=arcfile t=a
upda copy PC standard 1y dest=arcfile t=a
upda copy PC standard 5y dest=arcfile t=a
upda copy PC standard dyn dest=arcfile t=a
upda copy PC standard forever dest=arcfile t=a

acti pol pc standard

Pro přehled, toto nastavení vygenerujeme příkazem:

run mglist > c:\expirace.txt

DOMAIN_NAME	CLASS_NAME	DESTINATION	VEREXISTS	VERDELETED	RETEXTRA	RETONLY
PC	14D	PCFILE	NOLIMIT	NOLIMIT	14	14
PC	14D-DYN	PCFILE	NOLIMIT	NOLIMIT	14	14
PC	1Y	PCFILE	NOLIMIT	NOLIMIT	366	366
PC	5Y	PCFILE	NOLIMIT	NOLIMIT	1830	1830
PC	DYN	PCFILE	NOLIMIT	NOLIMIT	31	31
PC	FOREVER	PCFILE	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
PC	STANDARD	PCFILE	NOLIMIT	NOLIMIT	31	31
SQL	FILE	SQLFILE	NOLIMIT	NOLIMIT	31	31
SQL	STANDARD	SQLFILE	NOLIMIT	NOLIMIT	31	31
SQL	TAPE	SQLTAPE	NOLIMIT	NOLIMIT	31	31
STANDARD	14D	BACFILE	NOLIMIT	NOLIMIT	14	14
STANDARD	14D-DYN	BACFILE	NOLIMIT	NOLIMIT	14	14
STANDARD	1M	BACFILE	NOLIMIT	NOLIMIT	31	31
STANDARD	1Y	BACFILE	NOLIMIT	NOLIMIT	366	366
STANDARD	5Y	BACFILE	NOLIMIT	NOLIMIT	1830	1830
STANDARD	DYN	BACFILE	NOLIMIT	NOLIMIT	31	31
STANDARD	FOREVER	BACFILE	NOLIMIT	NOLIMIT	NOLIMIT	NOLIMIT
STANDARD	IMAGE	IMAGE	NOLIMIT	NOLIMIT	31	31
STANDARD	STANDARD	BACFILE	NOLIMIT	NOLIMIT	31	31
STANDARD	VCB	VCB	NOLIMIT	NOLIMIT	31	31

Nakonec nadefinujeme plán spuštění procesů (úloh)

V procesu úloh říkáme, kdy se proces zálohy spustí a jaká bude perioda, například každý den, nebo jednou týdně.

upda sched standard tsmdr object='C:\TSM\TSMDR.cmd >C:\TSM\TSMDR.log 2>&1'
upda sched standard SQL_FULL1 object='H:\TSM\SQL_FULL.cmd >H:\TSM\SQL_FULL.log 2>&1'
upda sched standard SQL_FULL2 object='H:\TSM\SQL_FULL.cmd >H:\TSM\SQL_FULL.log 2>&1'
defi sched standard SQL_logs action=command object='H:\TSM\SQL_logs.cmd >H:\TSM\SQL_logs.log 2>&1' period=4 peru=h startt=00:50:00 startd=01/01/2011 dura=10 duru=m

defi assoc STANDARD DAILY_INCR clusterSQL
defi assoc STANDARD SQL_FULL1 clusterSQL
defi assoc STANDARD SQL_FULL2 clusterSQL
defi assoc STANDARD SQL_LOGS clusterSQL

Přehled vygenerujeme zpětně příkazem: *run SCHEDLIST*

```
SCHEDULE_NAME     ACTION        OBJECTS                                          DAYOFWEEK
STARTTIME
---------------   ------------  ------------------------------------------------ ------------------
--
SQL_LOGS          COMMAND       H:\TSM\SQL_logs.cmd >H:\TSM\SQL_logs.log 2>&1    Any
00:50:00
SQL_FULL2         COMMAND       H:\TSM\SQL_FULL.cmd >H:\TSM\SQL_FULL.log 2>&1    Thursday
01:10:00
SQL_FULL1         COMMAND       H:\TSM\SQL_FULL.cmd >H:\TSM\SQL_FULL.log 2>&1    Tuesday
01:10:00
WINARCH           COMMAND       c:\Arch.cmd>c:\Arch.log 2>&1                      Any
05:00:00
TSMDR             COMMAND       C:\TSM\TSMDR.cmd >C:\TSM\TSMDR.log 2>&1          Any
14:10:00
DAILY_INCR        INCREMENTAL                                                    Any
21:00:00
PC_BACKUP         INCREMENTAL                                                    Weekday
08:00:00
```

Jak jsou úlohy přiděleny zálohovaným serverům (klientům) získáme příkazem: *Q ASSOC*

```
Policy Domain   Schedule Name      Associated Nodes
--------------  ---------------    ---------------------------------------------------------------

STANDARD        DAILY_INCR         CLUSTERSQL SRV-SQL1 SRV-SQL2 SRV-DMZ SRV-LAN SRV-TSM

STANDARD        SQL_FULL1          CLUSTERSQL SQLDMZ

STANDARD        SQL_FULL2          CLUSTERSQL SQLDMZ

STANDARD        SQL_LOGS           CLUSTERSQL SQLDMZ

STANDARD        TSMDR              SRV-TSM

STANDARD        WINARCH            -

PC              PC_BACKUP          -
```

Tím máme instalaci serveru TSM hotovou a zkonfigurovanou. Proto doporučuji ještě jednou provést kompletní zálohu TSM databáze a stgpoolu.

14.13 Instalace TSM na klienty

Klienti jsou ty servery, ty stanice, ty zařízení, které chceme zálohovat. Instalace bude mnohem snazší, než byla instalace TSM a provádí se buď pomocí metody zálohování souborů, nebo zálohování databází (pokud jde o MS SQL Server).

Chceme typicky zálohovat nějaký server ***souborově,*** denní zálohou, která bude probíhat 1x za den.

Na straně TSM Serveru provedeme registraci klienta:

reg node ARACHNE heslo dom=STANDARD backdel=yes maxnumm=7

*def assoc STANDARD daily_incr **

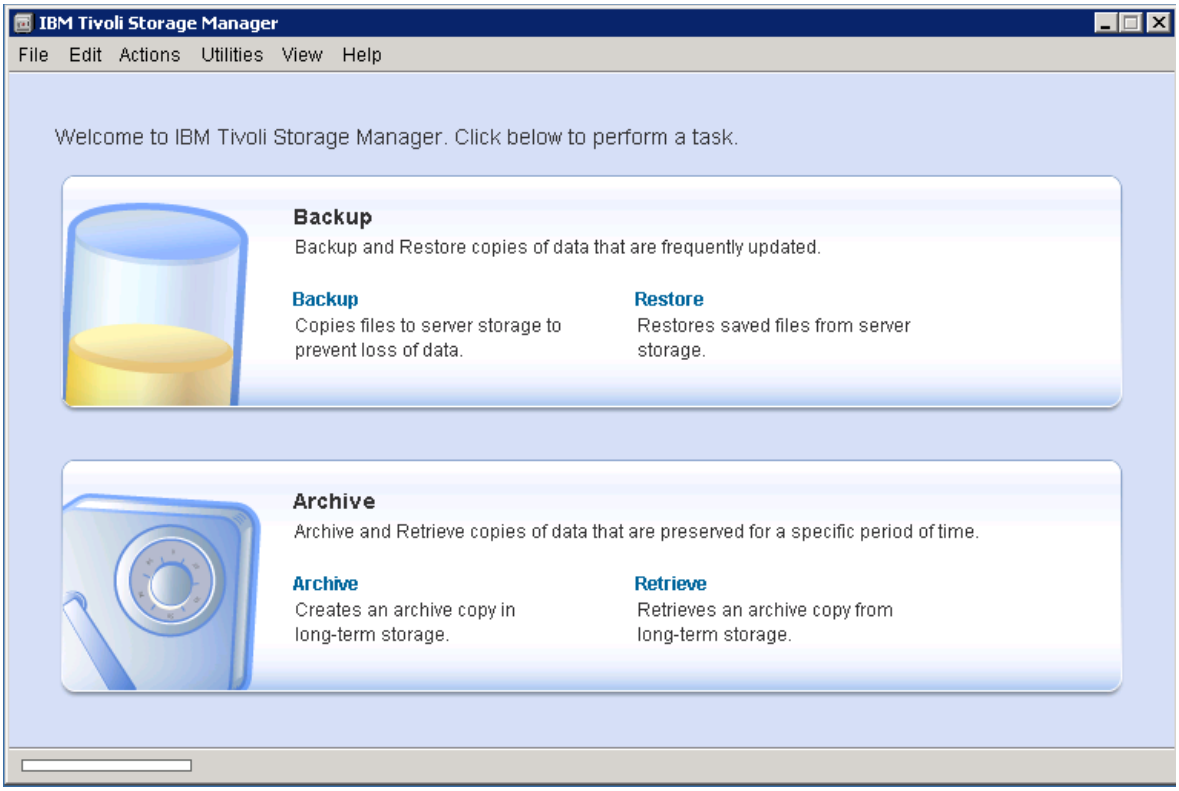
Na straně zálohovaného serveru (clienta)

- Spustíme instalaci BA Clienta pro 32bitovou verzi: *6.3.0.0-TIV-TSMBAC-WinX32.exe*.

  Pro 64bit verzi OS: *6.3.0.0-TIV-TSMBAC-WinX64.exe*.

  Průvodce instalace provede kontrolu verze OS, kontrolu chybějících komponent a po úspěšném ověření nainstaluje vše potřebné. Po Instalaci BA Client vypadá následovně.

  Aplikaci je možné spustit z nabídky start nebo spuštěním aplikace:
  C:\Program Files\Tivoli\TSM\baclient\dsm.exe



Obrázek 28 – Konzole Aplikace TSM

Jelikož máme anglickou verzi OS clienta, instalovali jsme EN verzi BA clienta

- Do proměnných prostředí (Environment) přidáme systémové proměnné:

  DSM_DIR C:\Program Files\Tivoli\TSM\baclient

  DSM_LOG C:\Program Files\Tivoli\TSM\baclient

  DSM_CONFIG C:\Program Files\Tivoli\TSM\baclient\dsm.opt

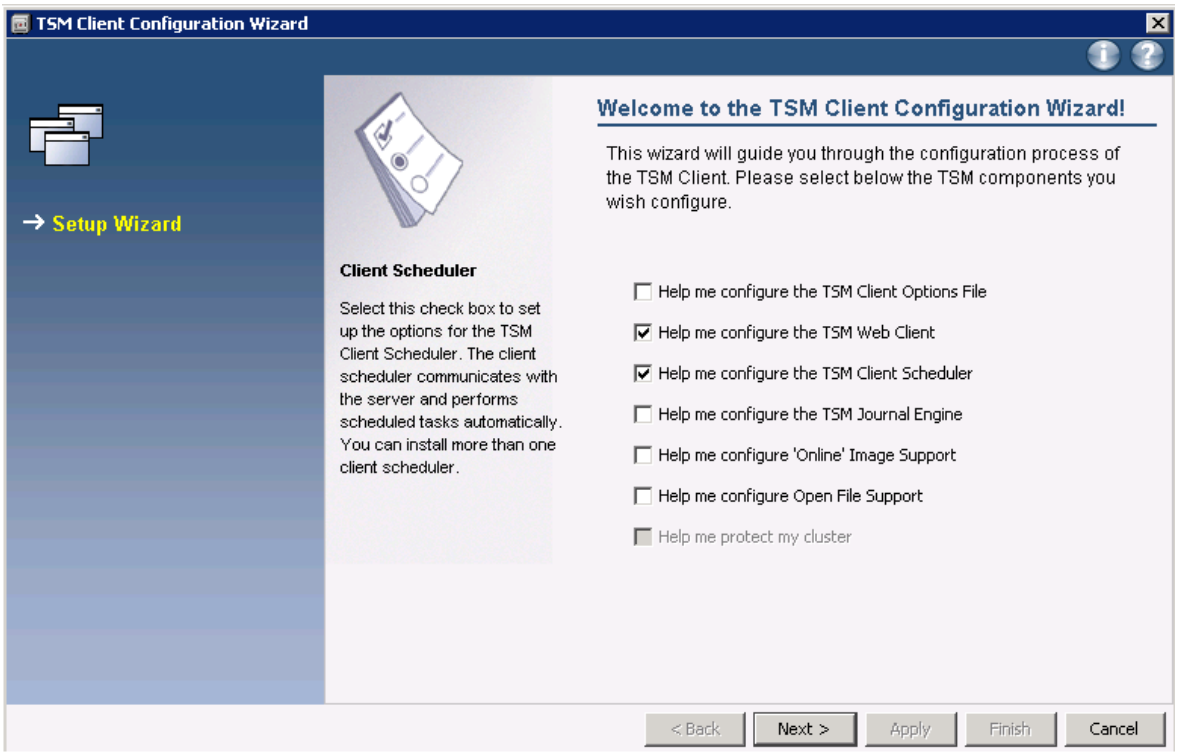
  pridat do path C:\Program Files\Tivoli\TSM\baclient

- Nakopírujeme dsm.opt do složky: *C:\Program Files\Tivoli\TSM\baclient\ dsm.opt*.

Obsahem tohoto souboru je Connect string na TSM zálohovací server a vyjímky ze záloh. Soubor a jeho konfigurace je obsažen v příloze.

- Spustíme z *příkazové řádky* příkaz pro zinicializování hesla: *dsmc q ses* a přihlášením připojíme klienta. Potvrdíme zadání hesla a všechny registrace se provedou automaticky.
- Nakonec instalace spustíme *dsm.exe* (Backup Archove Gui) a ve volbě *Utilities - vybereme Setup Wizard.*

  Tím se spustí instalace a po potvrzení instalace konfigurace TSM Web Client a TSM Client Sheduler zahájíme instalaci.



Obrázek 29 – Instalace TSM Klienta pomocí průvodce 1

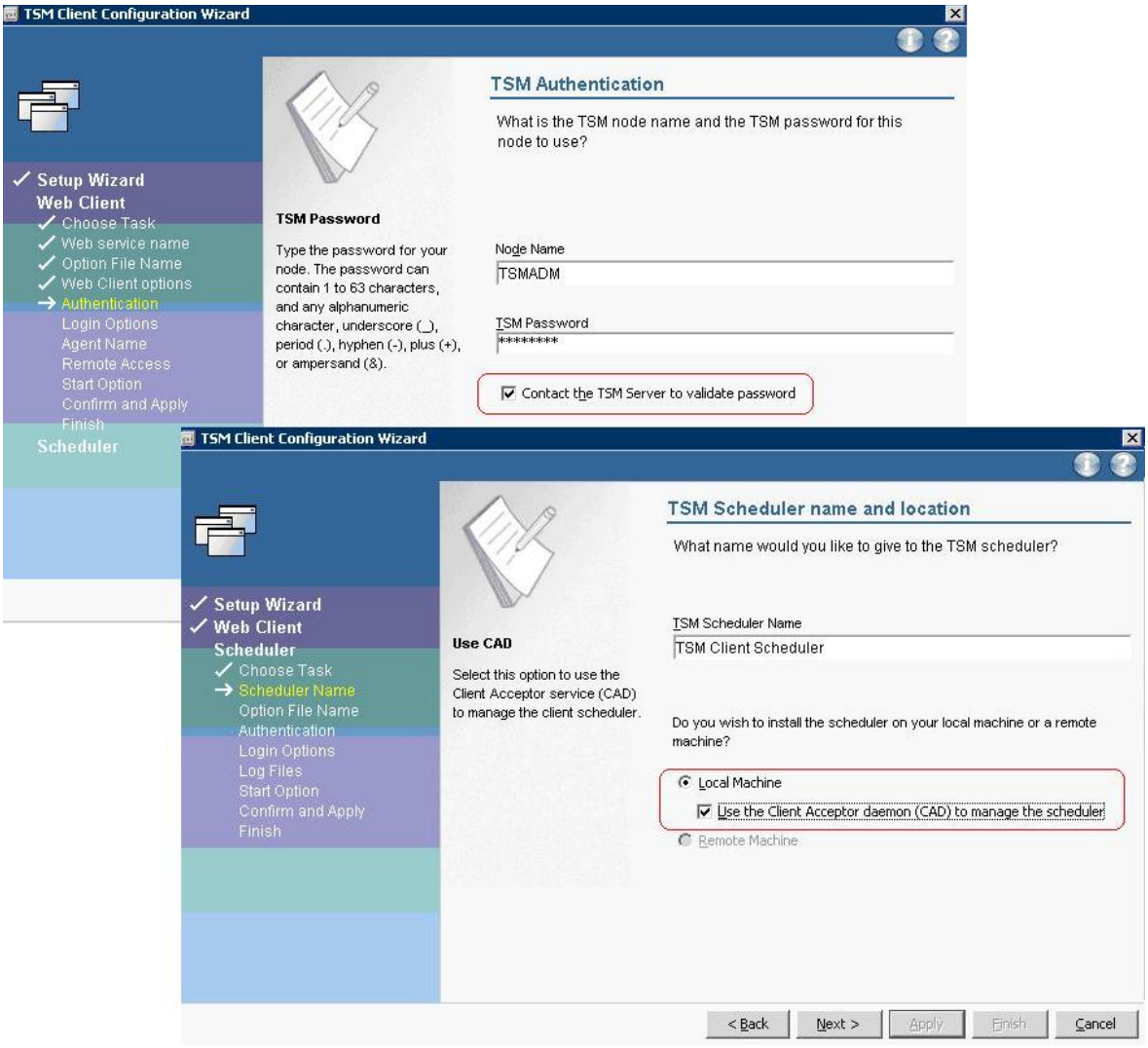
V průvodci instalace budeme muset zadat následující parametry:

- Po rozbalení instalačních souborů vybereme Client Acceptor.
- V průvodci Web clienta potvrdíme výchozí port pro komunikaci 1581.
- Zadáme umístění, kde má být TSM Client nainstalován.
- Další obrazovka je hodně důležitá, zadáme Node Name a TSM Password. Node name je název serveru, pod kterým budeme provádět zálohy. Tento název se pak prováže

s konfiguračními soubory a pouze pod tímto názvem budeme dohledávat informace z Tivoli Monitoringu. TSM Password je heslo k TSM účtu, ne k účtu serveru.

Důležitým parametrem je i contact the TSM Server to validace password. Tato volba nám ověří, zda heslo bylo zadáno správně, volbu zaškrtneme.

- Další důležitou obrazovkou je právě ta následující, zadání pod jakým účtem poběží klient TSM. Doporučuji buď použít systémový účet, nebo účet samostatný pro daný server, kde zrušíme vypršení platnosti hesla. Službu BA Klient TSM zvolíme startovat automaticky po spuštění OS.

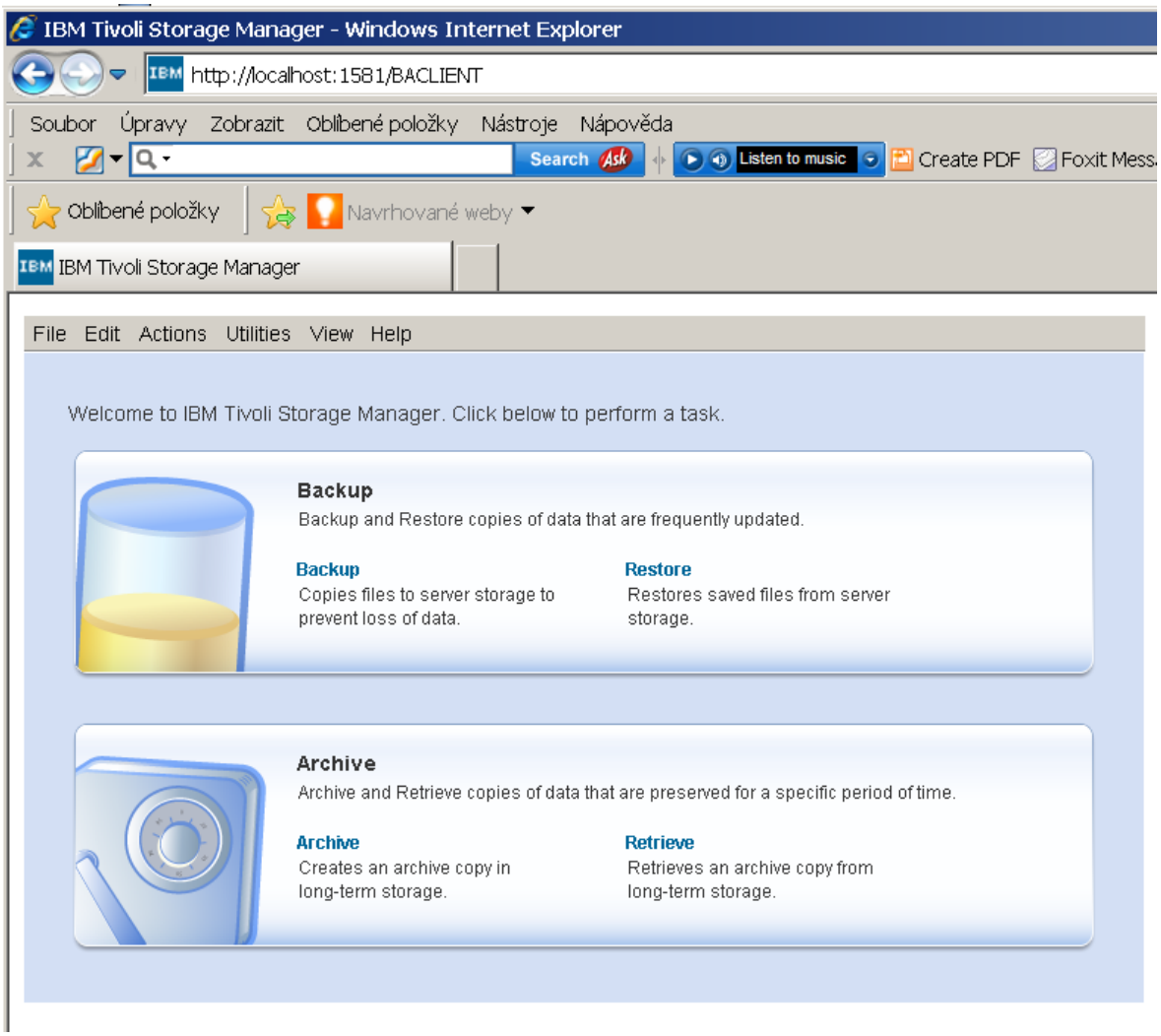


Obrázek 30 - Instalace TSM Klienta pomocí průvodce 2

- Na obrazovce týkající se Schedules zatrhneme položku CAD, tím se podřídíme CAD plánům z TSM serveru.
- Nakonec potvrdíme výchozí cesty k TSM shedule a error logum.

Na závěr doporučuji provést test spojení TSM serveru s TSM klientem.

Z clienta zadáme cestu do URL prohlížeče ve tvaru: *http://servername:1581*, provedeme přihlášení pod TSM účtem a pokud se nám zobrazí obrazovka, jako je na obrázku 31, je vše v pořádku. Před spuštěním jsme upozorněni, na instalaci apletu JAVA a pokud máme tento plugin nainstalován, zobrazí se stejná konzole ve Web rozhraní.



Obrázek 31 – Konzole Aplikace TSM přes IE

14.14 TSM SQL Klient

Instalaci SQL klienta pro zálohování se provádí pouze pro servery, kde je služba MS SQL ve verzi 2005 a vyšší, libovolně na všechny edice.

Před instalací SQL klienta je potřeba si uvědomit, že na zálohování databází SQL serveru lze použít pouze jeden typ záloh. Pokud tedy používáme zálohy pomocí nativními prostředky MS SQL Serveru, je potřeba tyto zálohy zrušit. Důvodem je práce

nad transakčním logem, kdy pokud provedeme full backup, smažeme transakční log. Pokud by tedy k takové záloze přistupovalo více zálohovacích technologií, nedostali bychom se k plným zálohám dat s žádným zálohovacím systémem.

Pokud srovnáme MS SQL zálohování s TSM zálohováním, jednoznačně širší možnosti máme právě s TSM zálohami.

Výhody daných řešení zálohování

TSM SQL zálohování	MS SQL zálohování
+ Robustní zálohování dat	+ Jednoduchost v nastavení zálohování
+ Robustní politika a verzování dat	- Není možné verzování záloh
+ Zpětná kontrola a monitoring záloh	- Není možnost automatického odmazávání starých záloh
+ Rychlost zálohování	
- Složité na údržbu a úpravy záloh	

Tabulka 9 – Porovnání výhod TSM a SQL zálohování databází

SQL TSM Client provádí zálohy ve formě velkých souborů, kdy každá konkrétní databáze je samostatný soubor.

14.15 Instalace TSM SQL Klienta

Instalaci provedeme obdobně jako TSM Clienta pro souborové zálohy, nejprve spuštěním konfigurace a příkazů na TSM Serveru.

Zaregistrujeme MS SQL Server do TSM

MS SQL je v cluster modu, jedna služba nad dvěma servery, 2 nody (sql1, sql2). Spustíme z příkazové řádky db2cmd:

regi node CLUSTERSQL heslo backdel=yes maxnumm=7 domain=standard
grant proxy target=clustersql agent=srv-sql1
grant proxy target=clustersql agent=srv-sql2

regi node SQL-clustersql heslo backdel=yes maxnumm=7 domain=SQL
grant proxy target=SQL-clustersql agent=srv-sql1
grant proxy target=SQL-clustersql agent=srv-sql2
grant proxy target=SQL-clustersql agent=CLUSTERSQL

Na MS SQL Serveru (klientovi)

Vytvoříme konfigurační soubory pro zálohy na MS SQL serveru. Disk D je clusterový disk, externí migrující disk mezi dvěma nody v závislosti na službě SQL. Disk D je vždy vynucen službou SQL serveru.

D:\tsm\dsmcl.opt .... konfigurační soubor pro clusternode B/A klienta
D:\tsm\dsmsql.opt .... konfigurační soubor pro clusternode SQL klienta
D:\tsm\tdpsql.conf .... konfigurační soubor SQL agenta
D:\tsm\SQL_Full.cmd .... konfigurační soubor pro plnou zálohu databází
D:\tsm\SQL_Lgos.cmd .... konfigurační soubor pro plnou zálohu transakčních logů

Obsahy těchto souborů jsou v příloze DP

V dalším kroku provedeme zinicializování hesla klienta:

dsmc q ses

- potvrdíme volbu a zadáme jméno a heslo k ověření clienta vůči TSM Serveru, následně spustíme:

dsmc q ses -node=CLUSTERSQL
dsmc q ses -node=SQL-CLUSTERSQL

Následně spustíme instalační balík *5.5.5.1-TIV-TSMSQL-WinX64.exe*. Součástí instalace je kontrola na chybějící softwarové komponenty instalované v OS, a pokud je vše pořádku instalujeme EN verzi TSM SQL, potvrdíme licenční ujednání a výchozí cestu k instalaci clienta na každý nood SQL serveru: (*c:\program files\Tivoli\TSM*), na všech serverech doporučuji cestu ke klientovi stejnou vzhledem ke konfiguračním souborům.

Nakopírujeme licenční soubor do adresáře: *"c:\program files\Tivoli\TSM\TDPSql"*.

Nyní stopneme službu TSM na klientovi, jedná se o TSM Acceptor, provedeme přeinstalaci BA clienta a následně služby TSM opět spustíme. Zastavení a spuštění služby ručně jsme provedli proto, abychom nemuseli restartovat celý SQL Server vzhledem k tomu, že jsme chtěli ověřit, zda jde SQL TSM Client nainstalovat bez nutnosti výpadku SQL Služeb.

Spustíme soubor dle verze bitOS a zvolíme opravu clienta:

D\install\6.3.0.0-TIV-TSMBAC-WinX32.exe
D\install\6.3.0.0-TIV-TSMBAC-WinX64.exe

Do příkazové řádky zadáme příkaz pro přiřazení konfiguračního souboru klientovi TSM SQL.

Tento soubor je pak konfiguračním souborem pro zálohování dat SQL Serveru.

"c:\program files\Tivoli\TSM\TDPSql\tdpsql" /config=h:\tsm\tdpsql.conf /tsmoptf=h:\tsm\dsmsql.opt

V souboru je definováno připojení k TSM, jaké databáze nechceme, aby se zálohovaly a výstupy k log souborům pro Shedule a chybové zprávy clienta. Soubor je součástí příloh DP.

Dalším krokem upravíme soubor *D:\tsm\dsmsql.opt* a v něm položky INCLUDE a EXCLUDE.

V Include položce definujeme seznam záloh rozšířený o ty, které nejsou v archivemodu. Jsou to například metadata databází.

Naopak v Exclude položkách definujeme, které data nechceme zálohovat. Jsou to například testovací databáze a podobně.

Soubor je v příloze DP

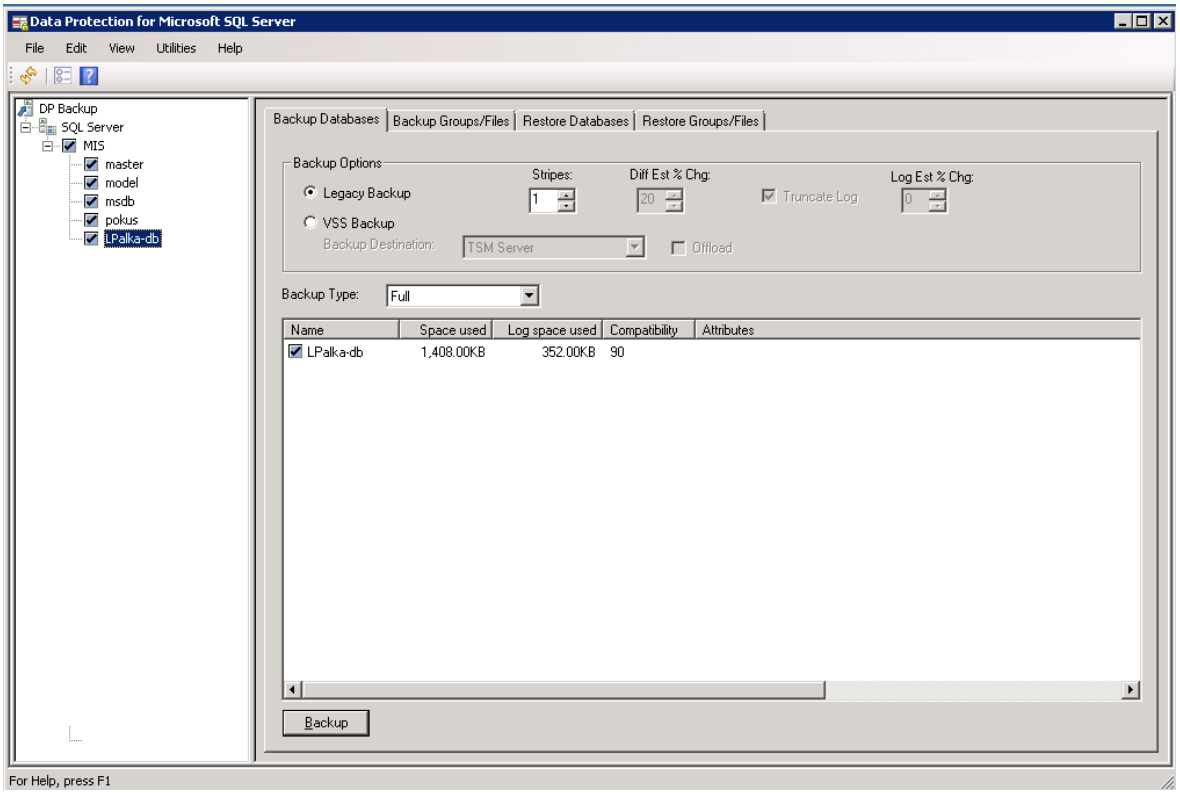
Tím je zálohování SQL databází pomocí Tivoli Storage Managementu nastaveno a je potřeba provést testy.

14.16 Testy a konfigurace SQL TSM Clienta

Po instalaci balíčku *TSM 5.5.5.1-TIV-TSMSQL-WinX64.exe* jsme nainstalovali SQL Clienta na Server. Vyzkoušíme jako první plnou zálohu master databáze. Přes start nabídku vybereme aplikaci TSM ClusterSQL Gui a spustíme ji. Ta nám vyvolá následující příkaz:

"C:\Program Files\Tivoli\TSM\TDPSql\tdpsql.exe" /config=h:\tsm\tdpsql.conf /tsmoptf=h:\tsm\dsmsql.opt

V nabídce vybereme plnou zálohu master databáze a spustíme proces. Pokud vše proběhne v pořádku, spustíme zálohu všech databází. Pokud i pak proběhne vše jak má, spustíme fullbackup přes konfigurační soubor *D:\tsm\sql_full.cmd* a následně *D:\tsm\sql_logs.cmd*.



Obrázek 32 – Obnova uživatelské databáze pomocí TSM Klienta

Před dalším krokem je potřeba si odzkoušet, že opravdu proběhly následující testy v pořádku:

- Test plné zálohy master databáze.
- Test plné zálohy všech databází pomocí skriptu sql_full.cmd.
- Test záloh databází transakčních logů pomocí skriptu sql_logs.cmd.

Pokud instalujeme TSM SQL Klienta na stand alone server provedeme následující konfiguraci sheduleru a CADu.

Pokud provozujeme SQL Server v Hi Evailebility modu, v našem případě dva nody SQL serveru, provedeme instalaci na všech nodech clusteru konfiguraci sheduleru a CADu.

Konfiguraci provedeme vždy na aktivním nodu. Není nutné restartovat OS, bohužel ale musíme jednou přemigrovat službu SQL serveru na druhý nod.

Popis a vyvětlení služby Sheduleru a CADu je popsáno v kapitole TSM zálohování filesystému.

Pro každý server provedeme následující konfiguraci. Spustíme db2smd a spustíme příkazy.

Připojení klienta:

SET NODE=CLUSTERSQL

SET PASS=heslo

SET OPT=D:\tsm\dsmcl.opt

Instalace sheduleru:

dsmcutil install SCHEDuler /name:"TSM scheduler: %NODE%" /clientdir:"c:\ProgramFiles\tivoli\tsm\baclient" /optfile:%OPT% /node:%NODE% /password:%PASS% /validate:yes /autostart:no /startnow:no /clusternode:yes /clustername:clustersql

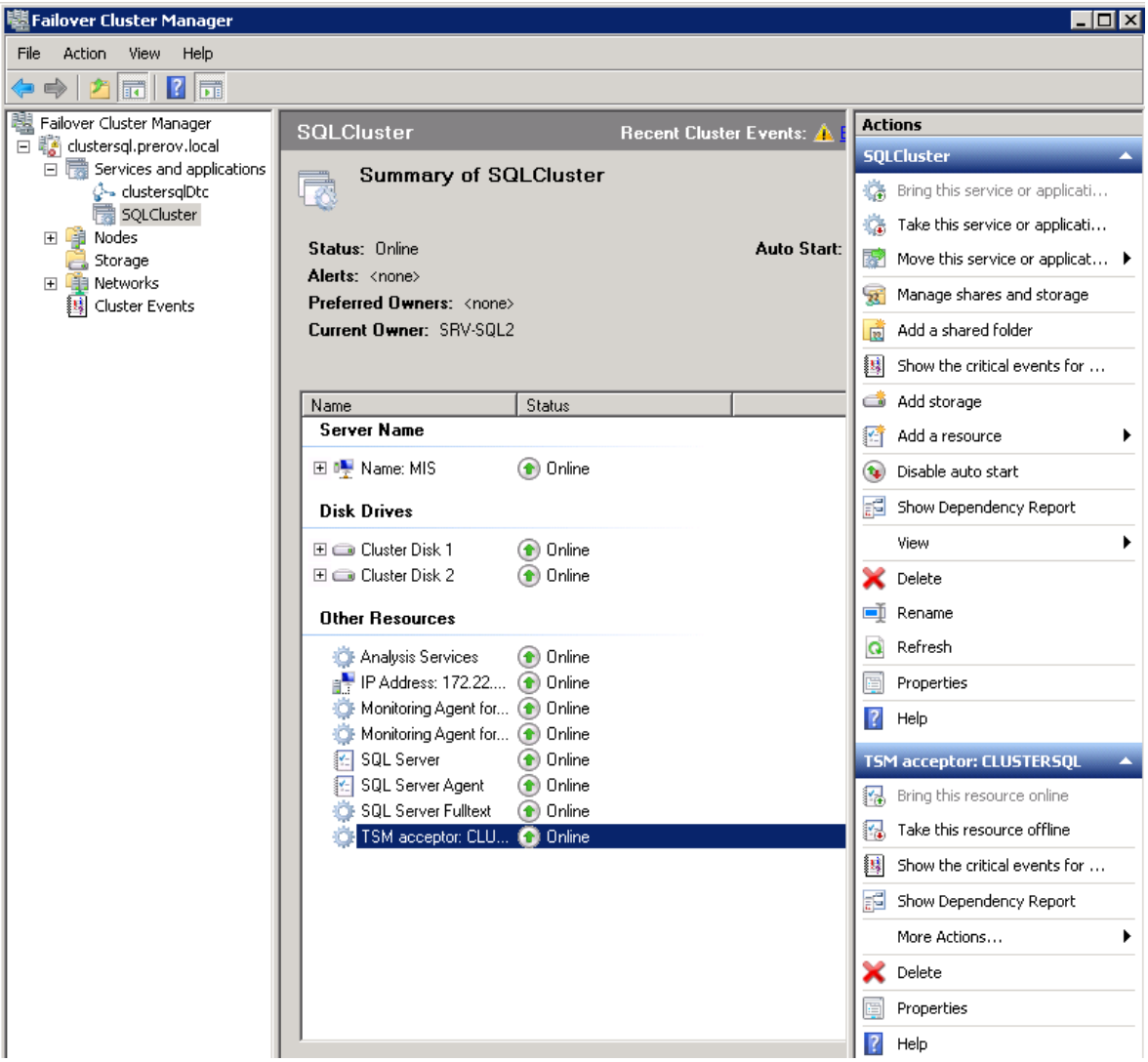
Instalace CADu (client acceptor daemon)

dsmcutil install cad /name:"TSM acceptor: %NODE%" /clientdir:"c:\Program Files\tivoli\tsm\baclient" /optfile:%OPT% /node:%NODE% /password:%PASS% /validate:yes /autostart:no /startnow:no /cadschedname:"TSM scheduler: %NODE%"

Instalace Remote agenta

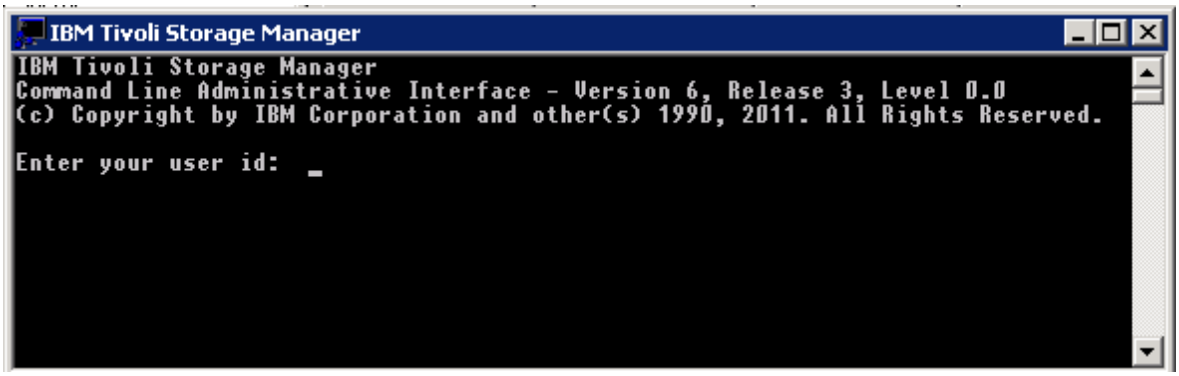
dsmcutil install remoteagent /name:"TSM remoteagent: %NODE%" /clientdir:"c:\Program Files\tivoli\tsm\baclient" /optfile:%OPT% /node:%NODE% /password:%PASS% /validate:yes /startnow:no /partnername:"TSM acceptor: %NODE%"

Jako poslední věc, přidáme do cluster services TSM akceptor (CAD). TSM akceptor musíme nastavit jako clusterovou službu, přes cluster administrator. Je to z důvodu, aby nám služba migrovala v závislosti na službě SQL Serveru.



Obrázek 33 – TSM akceptor jako clusterová služba

Administrative Console TSM – Command Line:



Obrázek 34 – Příkazová konzole TSM

15 KONFIGURACE SERVEROVÝCH A DATABÁZOVÝCH ROLÍ

V praktické části jen technicky realizuji nastavení práv uživateli SQL serveru. V teoretické části jsou popsána jak serverová role, tak i databázové role. Rovněž v teoretické části jsou doporučení, jakou zvolit koncepci přiřazení systémových práv společně s konkrétními bezpečnostními detaily.

Pro přiřazení systémových nebo databázových rolí použijeme Management Studio SQL serveru.

Nad Vlastností účtu v Login Properties můžeme nastavit obě tyto role.

Serverové role se nastaví přes záložku Server Roles. Nastavení vypadá následovně. T-SQL příkaz pro přiřazení uživatele do role sysadmin.

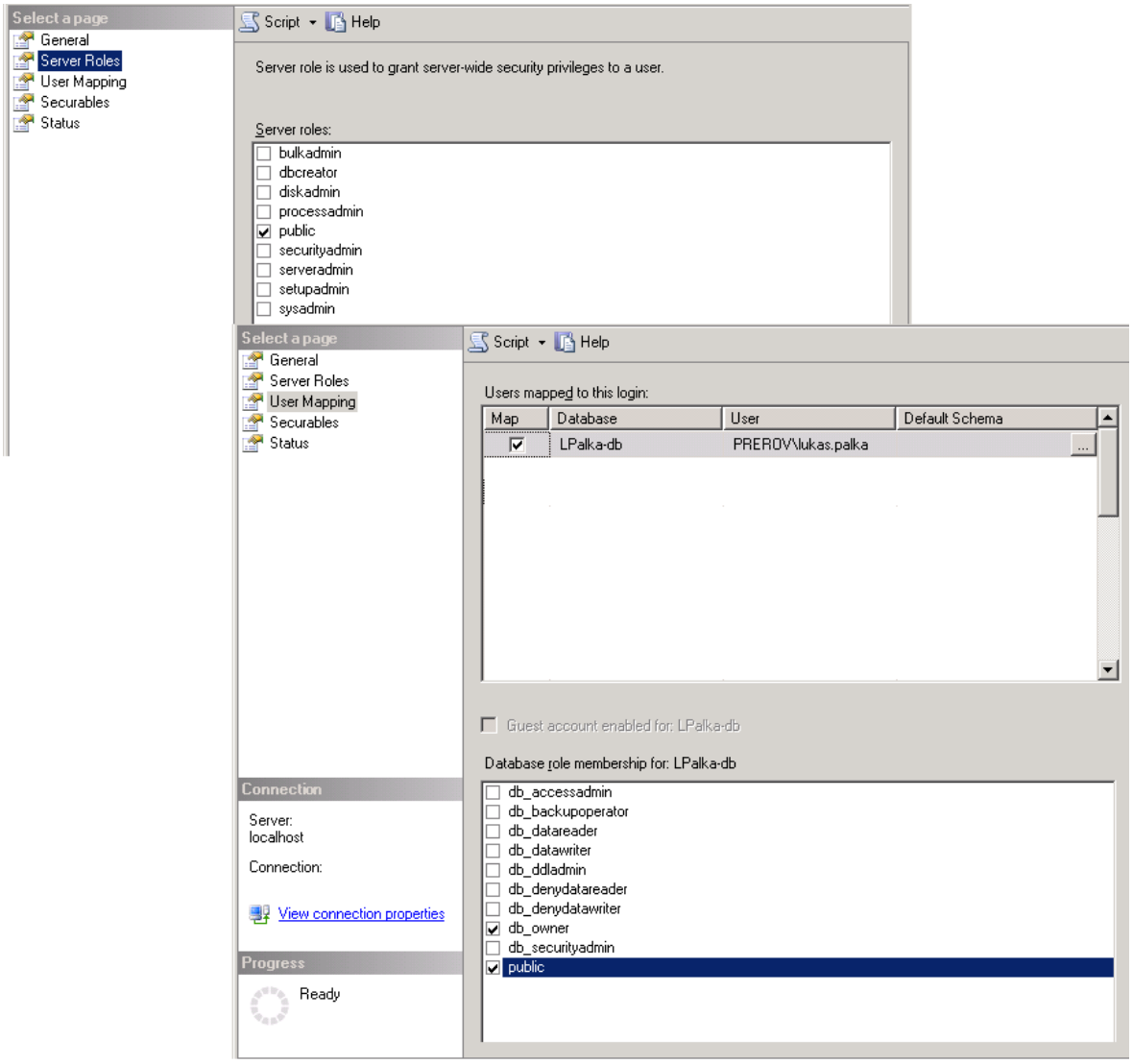
Přiřazení uživatele do role sysadmin:

```
EXEC master..sp_addsrvrolemember @loginame = N'PREROV\lukas.palka',
@rolename = N'sysadmin'
GO
```

Pro přiřazení databázové role použijeme záložku User Mapping. Nastavení se vždy vztahuje ke konkrétní databázi, proto vybere danou databázi a v role membership nastavíme oprávnění. T-SQL příkaz pro namapování práv na databázi LPalka-db s rolí db_owner.

```
USE [LPalka-db]
GO
CREATE USER [PREROV\lukas.palka] FOR LOGIN [PREROV\lukas.palka]
GO
USE [LPalka-db]
GO
EXEC sp_addrolemember N'db_owner', N'PREROV\lukas.palka'
GO
```

Grafické znázornění přiřazení práv uživateli přes management studio. Na první obrazovce přiřazení serverových rolí a na následující obrazovce přiřazení databázové role.



Obrázek 35 – Přiřazení práv uživateli pro danou uživatelskou databázi

Co SQL serveru z bezpečnostního hlediska vzhledem k použití rolí umožňuje? Je toho velká řada, od vytvoření vlastních rolí, přes hromadné operace s účty a rolemi, vazbu na centrální zprávu s doménovými účty a AD autentikací a mnoho dalších.

Rovněž musíme nastavení rolí a jejich použití chápat jako možnosti T-SQL. Což je obecný jazyk uzpůsoben k čistě databázovému použití. To co nám tedy neumožňuje grafické prostředí, lze leckdy mnohem rychleji a snadněji realizovat pomocí příkazů, které spustíme pomocí query analyzéra v Management Studiu, či v jiné konzoli po navázání spojení a autentikaci s platnými právy pro spuštění.

16 ŠIFROVÁNÍ DAT NA SQL SERVERU

V teoretické části jsem popsal princip a kroky šifrováním databáze, nyní si ukážeme, jak prakticky provedu zašifrování libovolné uživatelské databáze a popíši úskalí a vlastnosti této metody ochrany dat SQL serveru.

16.1 Zašifrování uživatelské databáze

Jako první krok provedeme vytvoření master klíče. Tento klíč je uložen v systémové databázi master. Při jeho vytváření se využívá ochrana heslem, které je předáno jako parametr dotazu pro vytvoření master klíče. Zajímavou novinkou SQL Serveru 2012 je šifrování tohoto klíče pomocí AES algoritmu namísto 3DES, který byl použit ve verzi SQL Serveru 2008 R2.

```
Use master
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'DiplomovaPracePalka4'
```

Je zde podmínka použití silného hesla (malé, VELKÉ písmeno, číslo a minimum 8 znaků).

Dále v databázi master vytvoříme certifikát, který bude použit pro ochranu šifrovacího databázového klíče. Následujícím příkazem vytvoříme certifikát.

```
CREATE CERTIFICATE TdeCert WITH SUBJECT = 'TDE Protection Certificate'
```

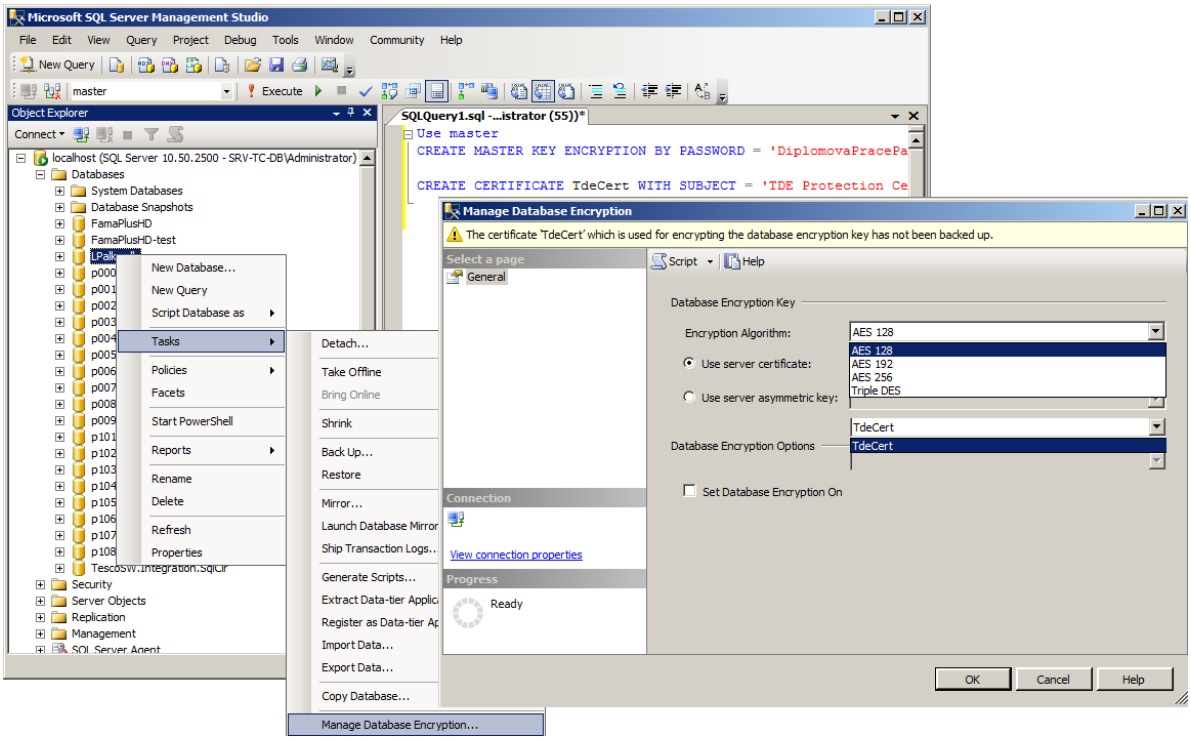
V dalším kroku v databázi vytvoříme samotný šifrovací klíč pro ochranu dat. Tuto operaci provedeme pomocí T-SQL příkazu:

```
USE [LPalka-db]

GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE [TdeCert]


GO
```

Případně můžeme nastavení SQL Serveru provést v konzoli Management Studia.



Obrázek 36 – Nastavení šifrování databáze

Nastavení šifrovacího algoritmu [6]

V rámci tohoto dialogu je potřeba vybrat šifrovací algoritmus, kterým bude celá databáze chráněna. Máme zde na výběr 4 možnosti – AES algoritmus s různou délkou klíče a algoritmus 3DES. Dále vybíráme již vytvořený certifikát. Je zde možnost použít i asymetrický klíč. Této volby se dá velmi dobře využít v kombinaci s tzv. Hardware Security Module, což je externí zařízení určené pro kryptografické operace a ukládání klíčů. Tímto způsobem můžeme oddělit klíče od administrátorů databázového serveru, a dosáhnout tak mnohem vyšší bezpečnosti uložení dat. Dále již stačí vybrat volbu Set Encryption On pro zapnutí šifrování. Pokud se podíváme následně na výstup dalšího dotazu, zjistíme, že databáze je v přechodném stavu šifrována. Zároveň však v tomto seznamu vidíme ještě jednu databázi a to databázi tempdb. Tato databáze je šifrována automaticky s první databází, na které zapneme TDE. Jsou tedy chráněna i data v databázi tempdb, jakmile jsou uložena na disk. Jedná se zejména o dočasné tabulky, data pro triggery (tabulky inserted, deleted) atd.

```
select dbName = [LPalka-db], encryption_state
FROM sys.dm_database_encryption_keys
```

Záloha databází chráněných pomocí TDE [6]

Zálohování databáze, která je šifrována pomocí TDE probíhá zcela standardně, např. pomocí System Center Data Protection Manager 2010. Velmi důležitou součástí zálohy je však záloha certifikátu, který je použit pro ochranu databázového šifrovacího klíče. Pokud přijdeme o tento certifikát, nebude možná obnova samotné databáze a data jsou nenávratně ztracena. Samotné dialogové okno pro šifrování i T-SQL nás varují při využití tohoto certifikátu, že ještě nebyla provedena jeho záloha. Samotná záloha klíče a certifikátu je poměrně jednoduchá. Pro obnovu certifikátu se jen musí využít možná trochu nezvyklý příkaz CREATE:

```
BACKUP CERTIFICATE TdeCert TO FILE = 'D:\Microsoft SQL
Server\MSSQL10_50.SQLDMZ\MSSQL\Backup'
WITH PRIVATE KEY
{FILE = D:\Microsoft SQL
Server\MSSQL10_50.SQLDMZ\MSSQL\Backup\TDECert.pfx', ENCRYPTION BY
PASSWORD = 'PalkaLukas123'}
```

Omezení TDE [6]

Transparentní šifrování dat není tak granulární jako předchozí mechanismy. Je nutné si uvědomit, že je šifrována opravdu celá databáze, což může být pro úvodní zašifrování v závislosti na velikosti dat časově náročná úloha. Je také nutné si uvědomit, že TDE nerozlišuje mezi uživateli, je tedy nutné stále využívat zabezpečení dat pomocí oprávnění, TDE jen doplňuje bezpečnost uložených dat.

Velkou nevýhodou je absence podpory pro funkci FILESTREAM, která dovoluje ukládat velké binární objekty mimo hlavní databázový soubor do NTFS filesystému. Další důležitou nevýhodou je nemožnost kombinovat TDE a novou kompresi zálohování, která byla uvedena stejně jako TDE v SQL Serveru 2008. Díky nové kompresi záloh je možné vytvářet zálohy mnohem rychleji a tyto zabírají méně diskového prostoru. Záleží vždy na povaze dat jaké zrychlení a úspora místa budou dosaženy. Nejsou nereálné případy 2x rychlejšího provedení zálohy, která je 3 – 5x menší než záloha nekomprimovaná. Stejně tak je nutné dbát možných dopadů na výkon, které mohou být způsobeny u vysoce zatížených systémů tím, že při šifrování databáze je vždy šifrována i databáze tempdb.

16.2 Zabezpečení a přínos šifrování SQL databází

Je důležité si uvědomit, že databáze je šifrována na úrovní SQL serveru, kde sysadmin účet v SQL serveru má samozřejmě plný přístup i do všech zašifrovaných databází. Šifrování jako takové má obrovský význam na úrovni file systému, kdy pokud nám někdo ukradne soubory databází nebo zálohy databází, tak jsou tato data bezcenná bez soukromého klíče master databáze.

Pokud chceme zabezpečit server i na úrovni file systému, použijeme metodu ACL z technologie NTFS formátu oddílu. Následně pomocí práv na uživatele, pouze účtu, pod kterým běží SQL služba, přiřadíme práva full control na celou složku, kde jsou uloženy datové soubory SQL serveru.

17 TRUST MEZI DOMÉNAMI V LAN A V DEMITALIRIZOVANÉ ZÓNĚ SÍTĚ.

Trust mezi Doménami v Microsoft Active Directory

Vztah důvěry mezi doménami, vzhledem k zařazení AD do DMZ, vzhledem k nastavení důvěryhodnosti zvolíme jednosměrný trust mezi doménami. To znamená, že doména v síti LAN důvěřuje jen sama sobě, ale doména v DMZ plně důvěřuje AD v LAN.

1. Před nastavením trustu mezi doménami, musíme nastavit routy tak, aby obě domény na sebe viděly. Na straně firewallu proto nastavíme komunikaci jednoho doménového řadiče k druhému a naopak.
2. Následně pokud nám fungují pingy mezi servery, musíme nastavit službu DNS serveru nad oběma doménami.
   - Vytvořili jsme STUB zóny v DNS serverech obou domén (Na serveru v DMZ – Stub zónu „jméno domény v LAN“ a na serveru v LAN – Stub zónu „jméno domény v DMZ“).

     (Na SRV-DMZ – Stub zónu „prerov.local“ a na SRV-LAN – Stub zónu „DMZprerov.local“)
   - Otestovali jsme komunikaci pomocí PING a NSLOOKUP, což fungovalo oběma směry na FQDN serverů (SRV-DMZ.local, SRV-LAN.local).

Nyní provedeme konfiguraci trustu mezi doménami s jednosměrnou důvěrou ve směru: *Doména v DMZ důvěřuje doméně v LAN.*

17.1 Na straně příchozí důvěry Active Directory v LAN (doména prerov.local)

Nastavíme příchozí důvěru následovně:

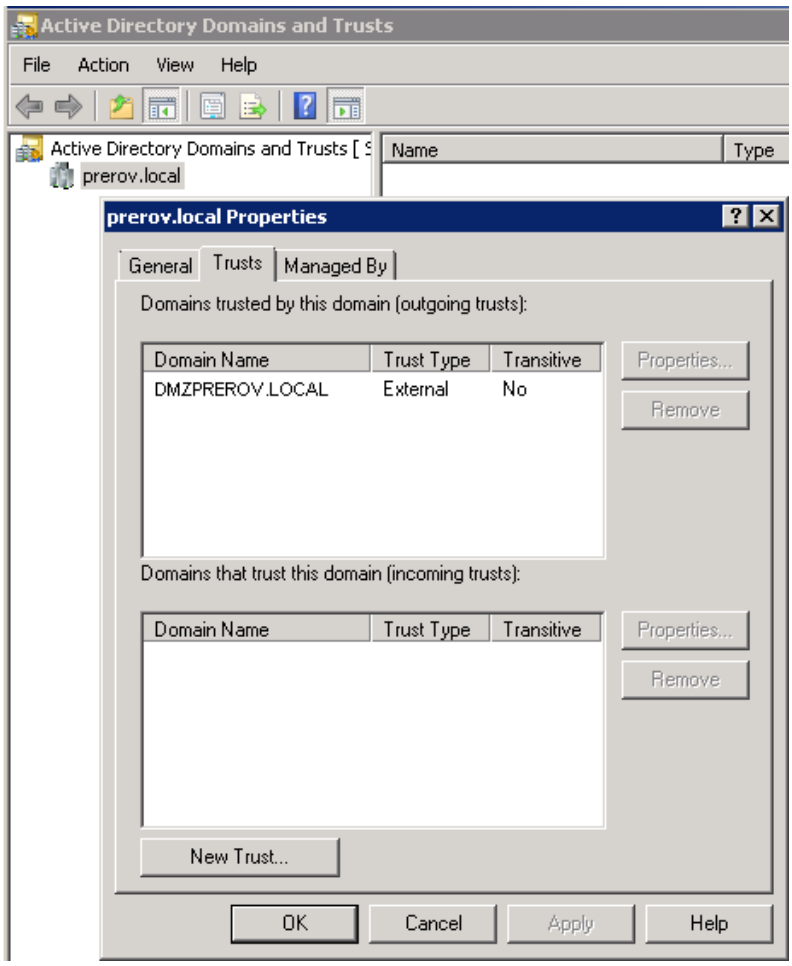
1. Spustíme konzoli Active Directory Domains and Trusts.
2. Ve vlastnosti domény jsem přidal nový trust a v definici trustu zvolíme následující parametry.
3. Název cizí existující domény, která si bude ověřovat účty v LAN doméně. Následně se provede ověření komunikace, zda na sebe doménové řadiče vidí. V případě

komunikace LAN a DMZ je potřeba nastavit na firewallu a serverech komunikaci a vydefinovat routy do těchto sítí.

4. Zvolíme Typ zabezpečení – externí důvěra (External Trust).
5. Na další obrazovce zadáme jednosměrný trust a to volbou One-way: příchozí. Tato volba říká, důvěřuje daná doména mojí doméně a ověří si právě u mě účet, který nenajde ve své doméně v DMZ.
6. Zadáme už jen heslo vztahu důvěry mezi doménami a tím je trust pro vnitřní AD nastaveno. Zkontrolujeme souhrnné informace a dáme dokončit.
7. Na záložce Trust Creation Complete ještě zvolíme příchozí zabezpečení (Confirm Incoming Trust). V našem případě NE, nechceme důvěřovat doméně v DMZ.

   Pokud nechceme potvrdit tuto důvěru, zvolíme příchozí důvěru NE (No, do not confirm the incoming trust).

   Pokud chcete potvrdit tuto důvěru, klepněte na tlačítko Ano (Yes, confirm the incoming trust).



Obrázek 37 – Konfigurace Trustu dvou domén 1

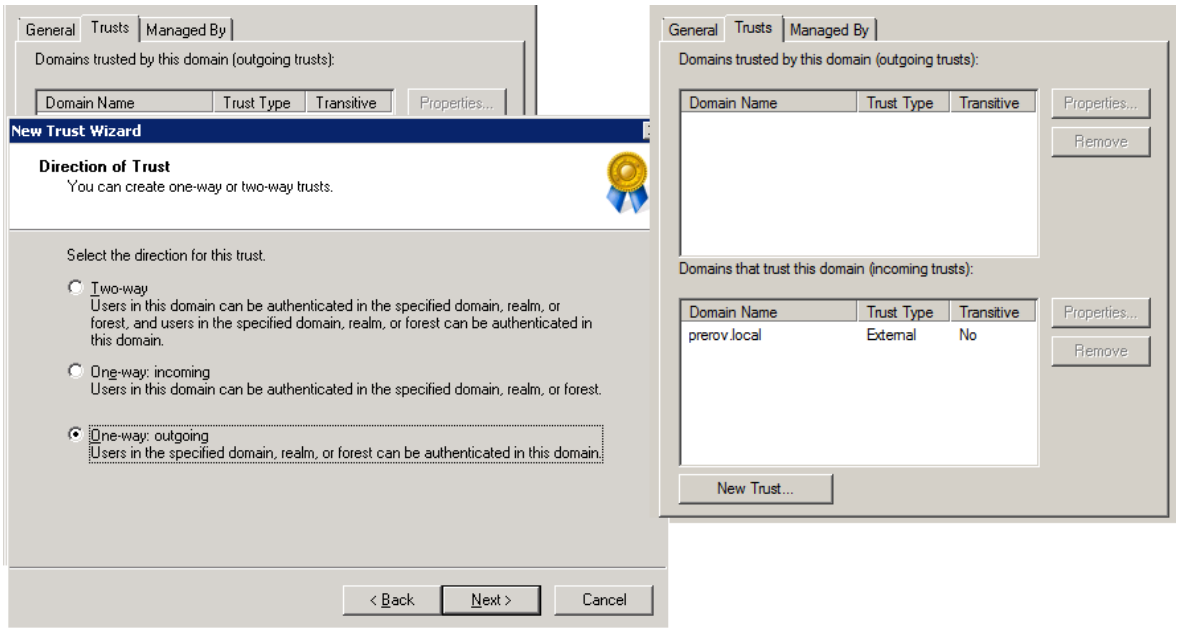
17.2 Na straně odchozí důvěry Active Directory v DMZ (doména DMZprerov.local)

Nastavíme odchozí důvěru následovně:

1. Spustíme konzoli Active Directory Domains and Trusts a na záložce vztahy důvěryhodnosti vytvoříme novou důvěru (trust).
2. Opět zvolíme známé jméno domény, kde přes DNS server se ověří, zda na sebe domény vidí a zvolíme opět externí důvěru trustu.
3. Směr zabezpečení nyní zvolíme One-way: odchozí.
4. Na záložce pro definici odchozí důvěryhodnosti (Outgoing Trust Authentication Level) zvolíme volbu pro ověření domény (Domain-wide authentication) a vybereme volbu definice typu ověření (Selective authentication).
5. Zadáme heslo pro trust domény a zkontrolujeme souhrnné informace.
6. Na záložce Confirm Outgoing Trust zvolíme směr důvery mezi doménami. Máme na výběr opět následující vztahy:

   No, do not confirm the outgoing trust a Yes, confirm the outgoing trust.

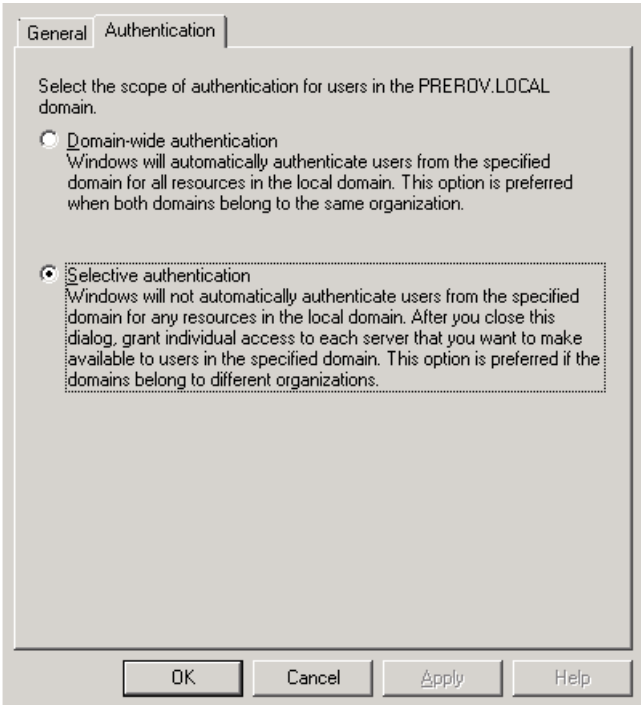
   V našem případě důvěřujeme doméně v LAN, zvolíme tedy volbu YES, ...



Obrázek 38 - Konfigurace Trustu dvou domén 2

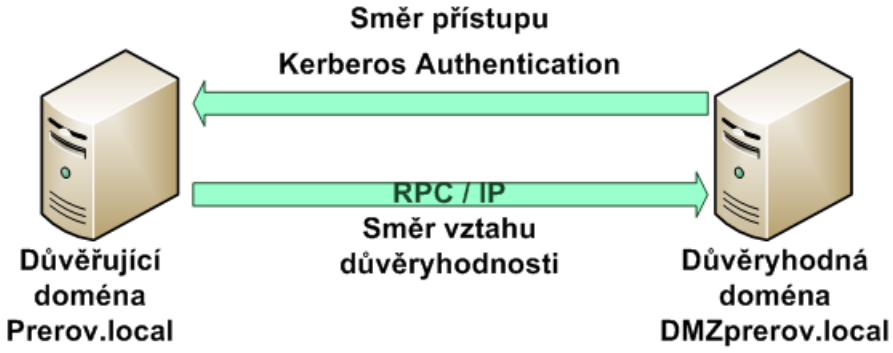
17.3 Dokonfigurace Trustu

Protože byl vybrán typ zabezpečení „selective“, tak se musí na každém serveru, kam mají přistupovat uživatelé z druhé domény, nastavit oprávnění.



Obrázek 39 - Konfigurace Trustu dvou domén 3

1. VAD users and computers - v Zobrazení zaškrtnout Advanced view.
2. Otevřít kontejner se servery nebo PC, které chceme povolit.
3. Na daný server klikneme pravým tlačítkem myši a zvolit Security - přidat uživatele nebo skupiny z druhé domény, kteří budou moci přistupovat k tomuto objektu. Při potvrzení se zadá účet z druhé domény (DMZprerov\administrator).

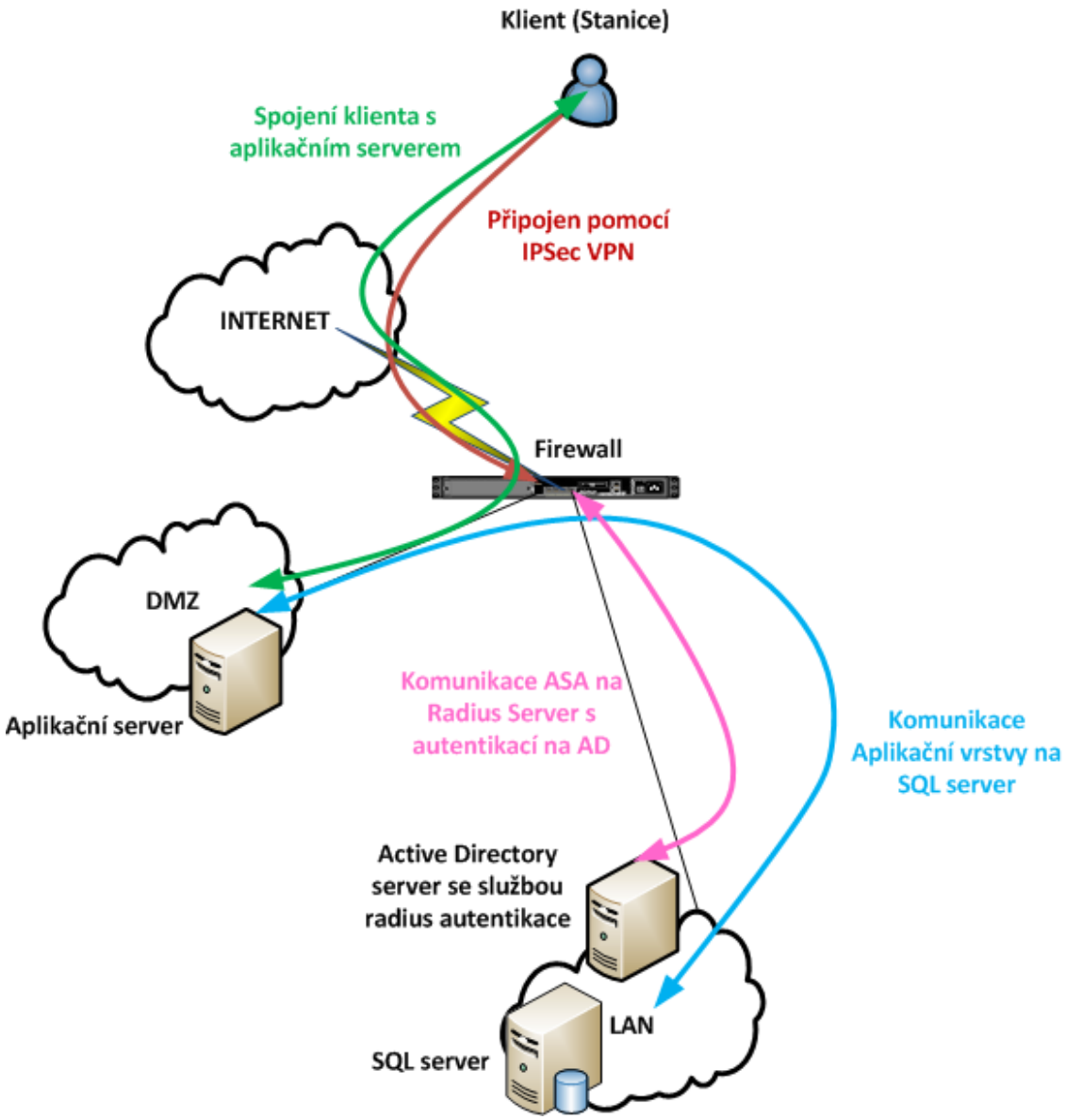


Obrázek 40 – Vztah důvěry mezi doménami

18 SQL SERVER V POTENCIONÁLNĚ NEBEZPEČNÉ ZÓNĚ DMZ. OCHRANA SERVERU, ZABEZPEČENÍ CHODU SERVERU A BEZPEČNÁ AUTENTIKACE

Popíši dvě varianty nasazení SQL serveru v síti, kdy SQL Server musí komunikovat s vnější síti přes Internet s klientem. V první variantě bude SQL Server nasazen do vnitřní sítě LAN a v druhé variantě je DMZ. Popíši rizika spojení, návrhy zabezpečení, jak spojit autentikaci s aplikační vrstvou serverů, jak bude fungovat autentikace do DMZ a LAN a také separaci dat dvou sítí s vazbou na trust domén a jejich vztah důvěryhodnosti.

18.1 SQL server v LAN



Obrázek 41 – Komunikace s SQL Serverem v LAN

18.1.1 Popis komunikace na databázový server

Popíši zde doporučení návrhu bezpečného ověřování vůči SQL serveru.

Klientská stanice, která přistupuje k datům SQL serveru prostřednictvím 2 či 3 vrstvé architektury se ze svého PC z konkrétní IP veřejné adresy v Internetu, připojí pomocí IPSec VPN klienta na firewall. Tato komunikace probíhá na TCP protokolu 4500. IPSecVpn je zabezpečena minimálně pomocí přihlašovacího jména (VPN group) a příslušného hesla. Před přihlášením je ověřena IP adresa, s kterou chceme navázat spojení. Pokud veřejná IP adresa nesouhlasí s IP adresou na seznamu přístupu, firewall, který zajišťuje službu VPN, nebude vůbec odpovídat na požadavek. Pokud bychom chtěli ještě důkladněji ověřit pravost klienta, lze svázat IP adresu s MAC adresou zařízení. Pokud je vše v pořádku, jsou akceptovány příchozí žádosti o navázání VPN tunelu a může dojít k procesu autentikace clienta VPN. Každý uživatel má unikátní přihlašovací údaje. Ověření probíhá přes radius server v doméně TC na TCP portu 1812. Každý účet smí navázat pouze jedno VPN spojení, to znamená, že více VPN tunelů není povoleno. Je to opět bezpečnostní ochrana, aby za routerem na klientovi nemohlo proběhnout více spojení a vždy byla zaručena ověřená autentizace s jednou konkrétní stanicí.

Potom může uživatel navázat šifrované spojení na daný aplikační server, který má vydefinován komunikační port na protokolu TCP a UDP.

- *Požadavky na klientskou stanici: veřejná IP adresa a VPN client (cisco VPN, kerio VPN – vždy je třeba, aby client byl podporován s ověřovacím zařízením. Např. cisco VPN Client a Cisco Firewall apod.).*
- *Požadavek na firewall: Podpora VPN, 3 zóny + 1 pro management s konfigurací každé zóny s vlastním síťovým rozhraním RJ45.*
- *Požadavek na DMZ: Vlastní zóna s jediným přístupem přes firewall ve vlastní oddělené síti.*
- *Požadavek na Aplikační server v DMZ: Server se síťovým rozhraním a OS firewallem.*

Spojení Aplikačního serveru z databázovým serverem v LAN

Aplikační Server, který přistupuje na SQL server, standardně komunikuje na TCP port 1433.

Spojení je vždy navázáno požadavkem ze strany aplikačního serveru. Databázový server komunikuje na známém portu pro aplikační server. Tento port lze změnit a bezpečnostně lze toto považovat za zvýšenou ochranu databáze. Výchozí port je TCP 1433. Databázový server nese řadu zabezpečení (autentikace, role a práva účtu, šifrování databází). Další možností jak zabezpečit data na SQL serveru, je data rozdělit a to do více instancí, kdy každá instance komunikuje na jiném TCP portu. Jedna z metod je FAKE SQL server, kdy na standardním portu 1433 nám běží instance s prázdnou databází a defaultním účtem SA s ultra bezpečným heslem spojený s editovacím nástrojem na monitoring autentikace. Autentikace ověřená a správná je považována následně za útok.

Požadavek na SQL server: Více instancí SQL serveru, běh služby na nestandardním portu.

Pokud stanice z internetu přistupuje přímo na SQL server – Tahle možnost, zde samozřejmě je ta nejhorší varianta vzhledem k bezpečnosti. Firewall má otevřené porty přímo pro SQL Server na VPN tunelu. Což sice může znít bezpečně, ale porty jsou otevřeny hned v okamžiku, kdy klient naváže VPN spojení. Proto je přístup k SQL serveru možný pro všechny aplikace a prográmky na stanici a ne vyhrazeně pro aplikační server v DMZ. Nejsme schopni zde pohlídat, že na klientovi nebude nainstalován závadný software, který se nám tímto pokusí o útok na SQL server, nebo se nám zmocní dat.

Autentizace SQL serveru vůči Aktive Directory

V rámci zabezpečení dat a serverů doporučuji mít účty pro SQL server v Aktive Directory. V takovém případě si před samotným požadavkem aplikačního serveru z SQL serveru převezme autentikační údaje z AD a následně je předá aplikačnímu serveru. Pokud autentikace proběhla v pořádku, klient obdrží data, o které požadoval.

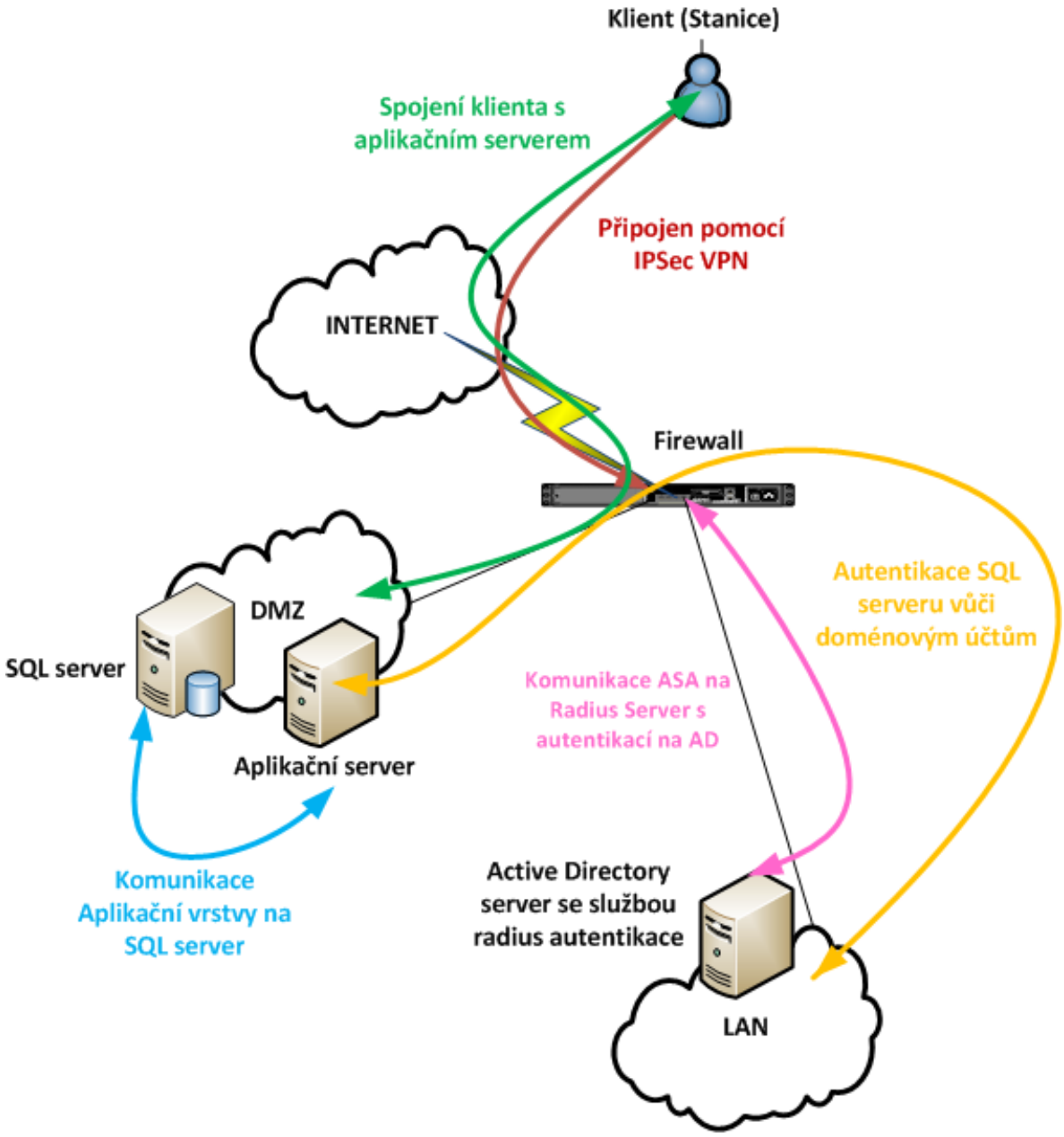
Autentizace SQL serveru vůči SQL službě

Pokud použijeme mixed mod v rámci autentikace SQL serveru, neprovádí se nám autentizace vůči centrálnímu autentizačnímu systému, který v Microsoft prostředí představuje Aktive Direktory. Z hlediska bezpečnosti, jsou účty lépe chráněny, pokud jsou centrálně řízeny v AD, jelikož server služba autentizace přistupuje k jinému serveru. Obecně použití mixed mod vzhledem k bezpečnosti, není žádným velkým problémem. Systém ověření a zabezpečení účtu je robustní a veškeré účty jsou definovány v dané uživatelské databázi a master databázi. V master databázi je řečeno, do jaké databáze má

účet umožněn přístup, login a heslo a restrikce. Nad uživatelskou databázi je to pak definice rolí a práv k daným tabulkám, sloupcům, pohledům a k indexům.

Pokud by útočník v nejhorším případě ovládl SQL server, již se dostane k databázím jako takovým, které jsou uloženy fyzicky na file systému. Zde nás neochrání ani SQL autentizace, ani Active Direktory autentizace. V takovém případě je jediná možnost ochrany a to šifrování databází. Tato technologie je rovněž obsahem této diplomové práce - Služba Radius Firewall – Active Direktory.

18.2 SQL server v DMZ



Obrázek 42 - Komunikace s SQL Serverem v DMZ

SQL server v DMZ je bezpečnostně mnohem horší varianta vzhledem k bezpečnosti, než varianta 3 vrstvé architektury, kdy SQL Server je v LAN a Aplikační část je v DMZ. Zde klient, i když přistupuje k datům pomocí aplikačního serveru, přistupuje k datům nebezpečně, protože po navázání VPN jsou již otevřeny porty mezi SQL serverem a stanicí. Popíši zde, jak dochází ke komunikaci. Ke každé části komunikace, popíši, jaká jsou rizika tohoto řešení.

Klientská stanice přistupuje do DMZ pomocí IPSec VPN klienta, kde naváže spojení vůči firewallu. Komunikace probíhá na TCP protokolu 4500. IPSecVpn je zabezpečena minimálně pomocí přihlašovacího jména (VPN group) a příslušného hesla. Opět dojde k ověření IP adresy, s kterou chceme navázat spojení, případně i MAC + IP adresy. Pokud je vše v pořádku, jsou akceptovány příchozí žádosti o navázání VPN tunelu a může dojít k procesu autentikace clienta VPN. Každý uživatel má unikátní přihlašovací údaje. Ověření probíhá přes radius server v doméně TC na TCP portu 1812. Každý účet smí navázat pouze jedno VPN spojení.

Potom může uživatel navázat šifrované spojení vůči serverům v DMZ, pomocí vydefinování komunikačního portu na protokolu TCP a UDP.

Toto řešení je nejčastěji použito pro přímý přístup clientské aplikace přímo k databázovému serveru v rámci spojení VPN. Autentizace probíhá vůči SQL serveru, který v mixed módu použije vlastní ověření. Pokud chceme zvýšit bezpečnost tohoto řešení, provedeme následující opatření:

1. SQL serveru provedeme změnu portu pro komunikaci (jiný port než TCP 1433).
2. Použijeme 2 instance SQL serveru, kdy defaultní instance použije port 1433, ovšem v této instanci nebudou žádné databáze. V instanci nastavíme plný auditing na login nad účtem SA a nastavíme maximálně silné heslo.
3. Operační systém SQL serveru zařadíme jako stroj do domény a k druhé (pravé) instanci s daty použijeme čistě autentikaci vůči Active Directory v LAN.
4. Mezi SQL serverem a klientem v Internetu přidáme aplikační můstek. Tato aplikační vrstva bude vyhrazeným strojem, který bude komunikovat s SQL serverem na daném portu. To zajistíme konfigurací firewallu OS SQL Serveru s vymezenou IP adresou.
5. Pro DMZ vytvoříme samostatný subnet s co nejmenším počtem IP adress. Pokud použijeme v DMZ pouze 2 servery. Aplikační a SQL použijeme masku 255.255.255.252/30 (2 IP adresy), nebo 255.255.255.248/29 (6 IP adres). Další možností je rozdělit subnet v DMZ pro servery a clienty.
6. Nastavení VPN spojení. VPN tunel na vyhrazenou IP a MAC adresu s clientem.
7. Automatické security aktualizace OS, robustní heslo pro administrátora OS kde je MS SQL nebo Aplikační Server.
8. Použijeme šifrování databází, lze od verze SQL serveru 2008 Standard Ed.

9. Použijeme robustní firewall např. od Cisca a autentizace VPN clienta pomocí Radius služby na Active Directory v LAN síti. Přenos radius serveru na protokolu TCP port 1812.

SQL Server komunikující na standardním portu 1433 s mixed autentikací je velmi snadným cílem zjištění infrastruktury scan aplikací. Po scanu celého rozsahu daného subnetu, si útočník velice snadno může udělat obrázek o tom, kam má smysl vést útok. Pokud zjistí takový SQL Server a následně použije běžné útoky posledních kritických záplat nad OS nebo SQL služby a uspěje, není už cesty zpět. Poslední možností je šifrování databází pokud pronikne na server díky security chybě OS. Takto se sice dostane k úložišti dat, ale nedostane se k datům SQL serveru.

Autentikace na Active Directory v LAN síti přinese řadu bezpečnostních i jiných výhod. Účty, které jsou v AD v LAN jsou samozřejmě více bezpečné než server v DMZ, vůči kterému se autentikuje aplikační server, nebo přímo klient se spojením VPN. Druhou obrovskou výhodou je centralizace zprávy účtu. Tato centralizace může zahrnout i zprávu tunelu a přístupu na firewallu, kdy v dnešní době většina zařízení podporuje Radius. Pomocí této rádius služby můžeme spravovat přístupy pod jedním centrálním adresářem, v našem případě Active Directory. Přenos mezi firewallem přes port Radius je šifrován a podporuje všechny bezpečnostní standardy.

Aplikační můstek s přesně definovanou komunikací je vždy obrovským přínosem v bezpečnosti. Pokud SQL server je v jedné síti jako aplikační server, což v DMZ je splněno, pro vyšší bezpečnost si pomůžeme nadefinováním přesné komunikace na úrovni OS firewallu nad oběma servery. SQL server nebude komunikovat na svém vyhrazeném komunikačním portu SQL služby (defaultně 1433 – my ale zvolili jiný) s jiným serverem, než je aplikační server. Tím docílíme nulového útoku z jiného serveru, než je aplikační server. SQL server je tak ohrožen jen v případě, že nejprve se útočník dostane přes tento aplikační server. My samozřejmě zabezpečíme aplikační server rovněž o přesně vydefinovanou komunikaci vůči firewallu a následně VPN, takže riziko útoku výrazně snižujeme. Aby bezpečnost byla ještě dokonalejší, můžeme SQL server i Aplikační server začlenit do domény v LAN a požadovat autentikaci administratorských účtů jen pomocí doménových účtů. Tím získáme opět velice silnou ochranu před útokem. V tomto režimu nejsme navíc omezeni, ani SQL autentikací aplikačního serveru či klientské autentikace vůči službě SQL a můžeme účty pro tyto služby definovat v již zmíněném mixed módu.

Definice Subnetů s omezením na masku v případě dvou serverů v DMZ je ideální kombinace zabezpečení v režimu síťové hardwarové ochrany. Musíme ale počítat s tím, že clienti, kteří navazují VPN spojení, dostávají IP adresu z rozsahu DMZ. Tím pádem musíme předem dostatečně naddymenzovat rozsah adres, nebo v lepším případě udělat následující. Pro servery použijeme maximálně možnou restrikci na počet zařízení v DNZ a pro clienty vytvoříme samostatný subnet s routem do subnetu pro SQL a Aplikační server. Oddělit subnet serveru od clientu je už dnes standardem v sítích, není to ale zvykem v DMZ, přitom zrovna zde to považuji za velké plus pro bezpečnost.

VPN spojení na clientech je vždy výhodné svázat s IP, a pokud to lze i MAC adresou. Ve většině případů ale vazba na MAC adresu není možná, protože musíme pamatovat na to, že MAC adresa není adresa routeru, se kterou stanice komunikuje mimo síť, kterou spravujeme, ale je to konkrétní MAC adresa dané stanice. Pokud takových stanic máme stovky, nejspíše bychom denně měnili přístupy do naší DMZ a to není úplně uspokojivé. Vazba MAC adresy s IP nese sice vylepšení zabezpečení, ale na Internetu dnes najdete řadu mini aplikací na změnu MAC adresy, které by mohly posloužit případnému neoprávněnému clientovi (útočníkovi).

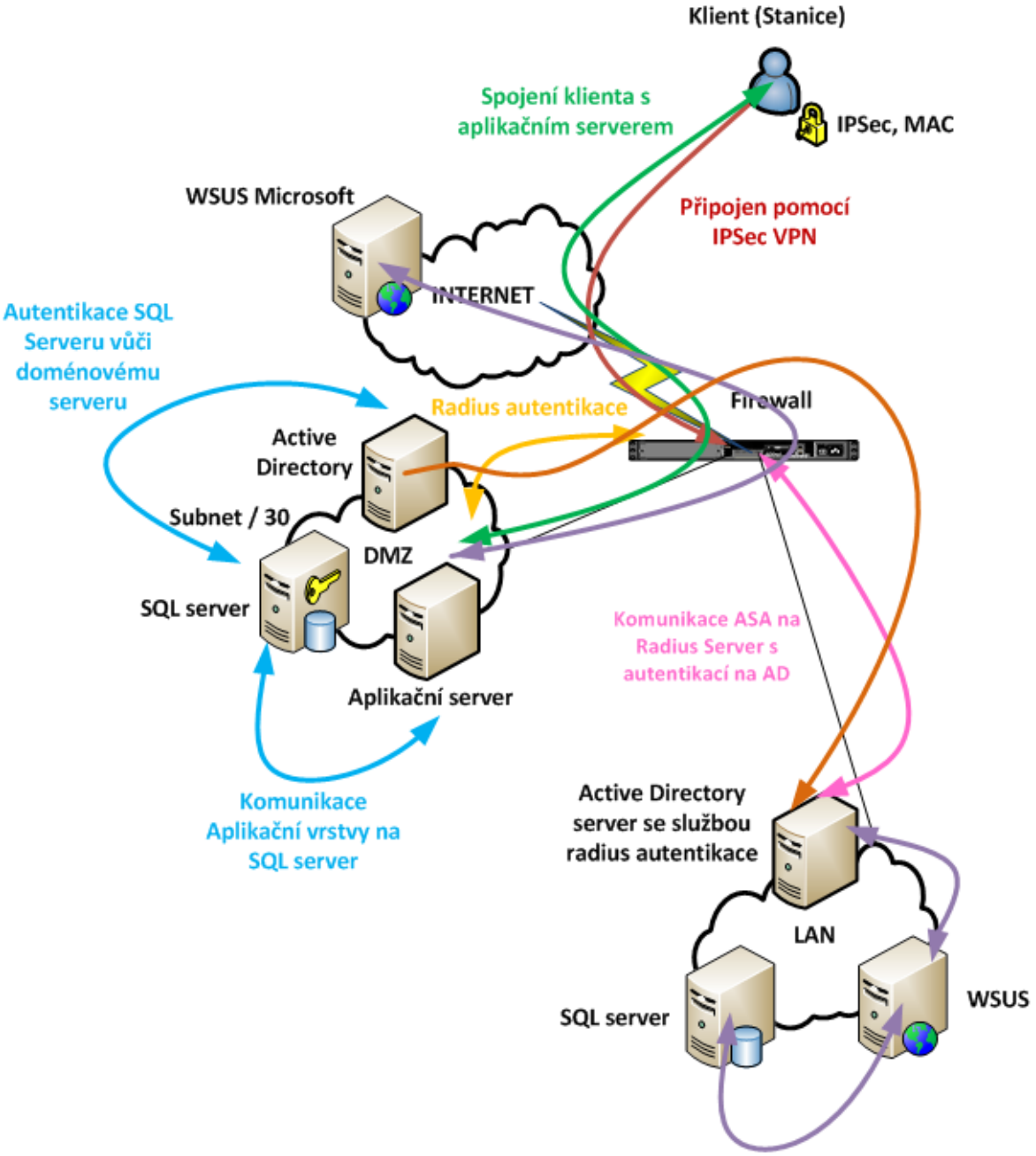
Aktualizace serveru o security balíčky by měly být nedílnou součástí kontroly administrátorů odpovědných za bezpečnost. Není nic horšího, jako obecně známá kritická chyba OS, kterou dokáže zneužít, díky publicitě i hacker amatér. Pokud v síti LAN použijeme např. Microsoft WSUS, máme tyto servery pod centrální kontrolou. Servery v DMZ lze rovněž napojit na WSUS a to přímo na spojení nad IP komunikací, i když servery nebudou v doméně. Pokud WSUS nemáme, nebo se nám tohle řešení nelíbí, použijeme výchozí režim aktualizace v OS, kdy máme nadefinován bezpečný aktualizační server přímo v security záložkách prohlížeče. Rovněž jsou zde nahrány certifikáty pro ověření pravosti a nemusíme se bát, že by došlo k podvrhu aktualizačního serveru.

Neměli bychom zapomenout, ani na záplaty OS na stanicích. Dalším bezpečnostním prvkem je antivir, který hlídá aplikace keyloger a podobně. Pokud chceme tuto funkcionalitu mít řízenou, lze to vše plně zautomatizovat. Tyto prostředky se jmenují karanténní zóna, kdy stanice musí splnit řadu pravidel, než je navázáno spojení. V této karanténě lze řídit kontroly na nejnovější aktualizace OS, nastavení OS, práva ve stanici (user) apod.. kontrola účtu, kontrola aplikací a nejnovější aktualizace antiviru. Karanténní zóna tak zajišťuje ověření stanice na její nezávadnost.

Šifrováním databází jako další možností ochrany dat serveru se zabývá samostatná kapitola této diplomové práce.

18.3 Variantní kombinace a další doporučení na bezpečnost SQL Serveru

Autentikace druhé AD v DMZ a trust mezi doménami (jednosměrný trust)



Obrázek 43 - Komunikace s výhradním SQL Serverem v DMZ a navázání vztahu s SQL Server v LAN

Pokud chceme oddělit interní uživatele a uživatele externí a získat a neumožnit ani externím uživatelům přístup do LAN, můžeme zvolit i tuto metodu zabezpečení.

V takovém případě budeme mít dva autentizační adresáře s dvěma režimy autentikace. Opět to je rozumné řešení a také v dnešní době již časté, kdy musíme zpravovat účty a data cizích počítačů a chceme být velice opatrní a sítě co nejvíce oddělit, se zachováním principů autentizace a zprávy serverů. Důvodem mohou být cizí organizace, kterým musíme publikovat část svých dat jakožto aplikaci, kterou nasadíme do vyhrazené sítě DMZ. Tím, že oddělíme adresářové servery, získáme dva režimy autentikace pro různé klienty, kdy cizí budou přistupovat pouze k aplikačnímu serveru a SQL serveru v DMZ, kdežto vnitřní uživatele, kteří rovněž budou přistupovat z Internetu, budou autentizováni vůči internímu AD a budou moci použít rovněž SQL a Aplikační server v DMZ. Celé toto řešení nese opět zvýšení bezpečnosti, kdy ne každý client musí přistupovat do vnitřní sítě, respektive služby, která zajistí funkčnost, kterou client požaduje.

Řešením problému je nasazení druhé Active Directory do DMZ. Tato druhá AD ponese účty jen svých externích clientů. Ověření interních účtů proběhne rovněž vůči tomuto serveru, ale interní client nebude autentizován, jelikož se zde nenajde jeho účet, který je na druhém AD v LAN. V takovém případě postupujeme tak, že použijeme Trust mezi doménami. Použijeme jednosměrný trust, kdy AD v DMZ důvěřuje AD v LAN, nikoliv obráceně. Tím nám proběhne autentizace cizích clientů proti AD v DMZ a ověření vlastních clientů se uskuteční díky předání požadavku AD v DMZ na AD v LAN. Ten pak požadavek vyhodnotí, přepošle do AD v DMZ a autentizace proběhla, jak měla.

Výhody:

- pokud nechci pustit cizí subjekty (klienty) do lan lépe SQL v DMZ,
- oddělený provoz dvou sítí s jednotnou správou účtů,
- v maximální míře bezpečnost řešení.

18.4 Volba operačního systému pro SQL Server a Aplikační Server vzhledem k bezpečnosti

SQL server doporučuji provozovat vždy na OS Windows Server 2008R2 – jsou zde defaultně vypnuté všechny služby.

Instalace Aplikačního, SQL serveru a dalších serverů, které chceme umístit do DMZ, doporučuji instalovat na verzi OS Windows Server 2008 R2.

Doporučuji to právě z těchto důvodů:

- Po instalaci OS jsou funkční jen ty služby, které jsou nutné pro běh OS. Vše co není třeba, je defaultně vypnuto a to je vzhledem k bezpečnosti to nejlepší. Každá služba, která nemusí běžet a která komunikuje přes TCPIP, je potencionálním rizikem pro ostatní služby.
- Na OS je defaultně zapnut firewall s automatickým povolováním těch portů, které jsou v závislosti na běžících službách. Opět od minulých verzí velký pokrok směrem k bezpečnosti. Všechna komunikace, která není potřeba, je vypnuta a funkční komunikace je pouze pro funkční služby běžící pod OS.
- Knowladge base, security balíčky a jiné záplaty.
- Koncepčně správně navržený OS pro bezpečnost služeb a dat.

ZÁVĚR

Hlavním cílem diplomové práce bylo seznámit čtenáře se všemi možnostmi a metodami, sloužících k zajištění bezpečnosti služby Microsoft SQL Serveru a samotných databází. Diplomová práce obsahuje komplexně řešenou bezpečnost dat databázového serveru. Diplomová práce detailně popisuje metody a prostředky služeb zajišťující bezpečnost serveru a dat, nasazení instalovaných bezpečnostních komponent a popis konfiguračních nastavení.

Softwarové řešení popisuje technologie jako je bezpečnost dat databázového serveru, metody autentikace, šifrování dat, bezpečnost komunikace, služba PKI, instalace Tivoli Storage Manageru a zálohování nativními prostředky MS SQL Serveru. Do hardwarového řešení patří návrh zapojení sítě, návrh umístění serveru v LAN a DMZ a parametry zálohovacích zařízení.

V diplomové práci mám detalně vysvětleny všechny zmíněné technologie, jak po stránce teoretické, tak po stránce praktické. Po stránce teoretické jsou popsány možnosti dané technologie, po stránce praktické je to pak instalace a konfigurace.

V závěru diplomové práce se dozvíme, za jakých okolností a jak je vhodné dané řešení použít a také jejich vzájemnou provázanost. Z diplomové práce jednoznačně nevyplývá použití jedné metody, každá metoda má své výhody, i nevýhody. Diplomová práce obsahuje popis hrozeb každého řešení a podrobné srovnání všech zmíněných nástrojů k zajištění bezpečnosti dat Microsoft SQL Serveru.

Jako poslední bod diplomové práce uvádím ucelený náhled na celkovou realizaci všech služeb bezpečnosti.

Na základě vypracování diplomové práce jsem našel řešení, jakým způsobem zabezpečit data SQL serveru, jaké jsou možnosti a prostředky, které lze v současné době použít k účelu ochrany dat SQL serveru. Diplomová práce toto řešení detailně popisuje po všech možných stránkách s popisem úskalí a postupných kroků, včetně důsledků a vazby na ostatní systémy.

Pokud shrneme do jedné věty cíl diplomové práce, je to studie k zajištění bezpečnosti SQL serveru za účelem realizace řešení postavené na použití bezpečnostních prostředků k ochraně služby Microsoft SQL Serveru a následně databází.

ZÁVĚR V ANGLIČTINĚ

The main objective of this thesis was to acquaint the reader with all the possibilities and methods used to ensure the security of Microsoft SQL Server and databases themselves. The thesis contains a comprehensive data security database server. This thesis describes in detail the methods and means of ensuring the security of the server service and data security deployment of installed components and description of the configuration settings.

Software solution describes technologies such as data security database server, authentication methods, data encryption, security, communications, PKI service, installation of Tivoli Storage Manager backup and native MS SQL Server resources. The hardware solution is the design of the network connection, the proposed location of the server in the DMZ and LAN backup devices and parameters.

In this thesis I explained all this technology in detail, both the theoretical and practical side. From the theoretical possibility of the described technology in terms of practical, it is the installation and configuration.

At the end of the thesis, we learn under what circumstances and how the solution should be used and their interdependence. The thesis clearly does not use one method, each method has its advantages and disadvantages. This thesis describes the threats to each solution and a detailed comparison of all these instruments to ensure data security of Microsoft SQL Server.

As a last point of the thesis I present a comprehensive insight into the overall implementation of all security services.

Based on the elaboration of the thesis, I found a solution, how to secure SQL Server data, what are the opportunities and resources that can currently be used for the purpose of protecting SQL data server. This thesis describes in detail the solution for all kinds of pages describing the pitfalls and successive steps, including the implications and links to other systems.

If we summarize in one sentence the objective of the thesis is a study to ensure the security of SQL Server in order to implement a solution based on the use of safety devices to protect Microsoft SQL Server database and then database.

SEZNAM POUŽITÉ LITERATURY

Kapitoly v knize

[1] STANEK, William R. Microsoft SQL Server 2005: Kapesní rádce administrátora. První vydání. Brno: Computer Press, a.s., 2006. ISBN 80-251-1211-X.

[2] MICROSOFT CORPORATION. Microsoft SQL Server 2000: Administrace systému MCSE Training Kit. První vydání. Brno: Computer Press, 2001. ISBN 80-7226-526-1.

[3] STANEK, William R. Mistrovství v Microsoft Windows Server 2008. První vydání. Computer Press, a.s., 2009. ISBN 978-80-251-2158-0.

Elektronické zdroje (WWW)

[4] MICROSOFT CORPORATION. SQL SERVER 2012 EDITIONS [online]. 17. dubna 2012 1:18:09 [cit. 2012-23-04]. Dostupné z: http://www.microsoft.com/sqlserver/en/us/sql-2012-editions.aspx

[5] Kerberos (protokol). In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-23]. Dostupné z: http://cs.wikipedia.org/wiki/Kerberos_(protokol)

[6] Šifrování dat na SQL Serveru. Šifrování dat na SQL Serveru - TechNet Blog CZ/SK - Site Home - TechNet Blogs [online]. 23. dubna 2012 12:17:12 [cit. 2012-04-23]. Dostupné z:http://blogs.technet.com/b/technetczsk/archive/2011/10/11/sifrovani-dat-na-sql-serveru.aspx

[7] RADIUS. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-23]. Dostupné z: http://cs.wikipedia.org/wiki/RADIUS

[8] MICROSOFT CORPORATION. Compare Microsoft SQL Server Editions [online]. 21. listopadu 2011 19:45:42 [cit. 2012-01-12]. Dostupné z: http://www.microsoft.com/sqlserver/en/us/product-info/compare.aspx

Digitální dokumentace

[9] IBM CORPORATION. IBM Tivoli Storage Manager Version 6.2 information center [CD]. Armonk, NY 10504-1785, U.S.A.: IBM Director of Licensing, 2009, 2010 [cit. 12.1.2012].

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AD	Active Directory
DB	Databáze
T-SQL	Transact SQL
OS	Operační Systém
KDC	Key Distribution Center
DMZ	Demilitarized Zone
CA	Certificate authority
MS	Microsoft
TSM	Tivoli Storage Manager

Standardizované zkratky:

HDD, CPU, SQL, RAM, LAN, NOD, NETBIOS, MB, GB, TCP/IP, DHCP, SID, IIS, I/O, HW, SW, HI, ODBC, API, CLI, LDAP, SSL, NTLM, NET, SAS, FC, HASH, TGS, RSA, TDE, RADIUS, PKI, VPN, IP

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: KONFIGURAČNÍ SOUBORY TSM

TSM CLIENT: SOUBOR DSM.OPT

tcpport 1500
TCPSERVERADDRESS srv-tsm.prerov.local

Exclude ":\microsoft uam volume\...*"*
Exclude ":\microsoft uam volume\...*.*"*
Exclude ":\...\EA DATA. SF"*
*Exclude *:\...\pagefile.sys*
*Exclude *:\IBMBIO.COM*
*Exclude *:\IBMDOS.COM*
*Exclude *:\MSDOS.SYS*
*Exclude *:\IO.SYS*
Exclude *:\...\system32\config\...
**Exclude *:\...\system32\Perflib*.dat*
Exclude *:\...\system32\dhcp\...
*Include *:\...\system32\dhcp\backup\...**
Exclude *:\...\system32\dns\...
*Include *:\...\system32\dns\backup\...**
Exclude.dir ":\System Volume Information"*
Exclude.dir ":\...\Temporary Internet Files"*
*Exclude.dir *:\Recycled*
*Exclude.dir *:\Recycler*
*Exclude.dir *:\$Recycle.bin*

passwordaccess generate

**
*exclude.dir *:\..._nobackup**

SCHEDLOGNAME "C:\Program Files\Tivoli\TSM\baclient\dsmsched.log"
ERRORLOGNAME "C:\Program Files\Tivoli\TSM\baclient\dsmerror.log"
SCHEDLOGRET 3 D
ERRORLOGRET 30 D

SNAPSHOTPROVIDERFS VSS
SNAPSHOTPROVIDERIMAGE VSS

WEBPORTS 1601 1602
MANAGEDSERVICES WEBCLIENT SCHEDULE

TSM CLIENT SQL: SOUBOR DSMCL.OPT

CLUSTERNODE YES
PASSWORDAccess Generate

COMMMethod TCPip
TCPServeraddress SRV-TSM
TCPPort 1500

SCHEDLOGNAME H:\TSM\dsmclsched.log
SCHEDLOGRET 3 D
ERRORLOGNAME H:\TSM\dsmclerror.log
ERRORLOGRET 30 D

exclude.dir g:\cluster
exclude.dir g:\MSCS
*exclude *:\...\MSSQL*\...**
exclude i:\....ldf*

SNAPSHOTPROVIDERFS VSS
SNAPSHOTPROVIDERIMAGE VSS

MANAGEDSERVICES WEBCLIENT SCHEDULE

WEBPORTS 1603 1604
HTTPPORT 1582

TSM CLIENT SQL: SOUBOR SQL_FULL.CMD

```
@ECHO OFF
echo Script %0 started: %DATE% %TIME%
set RC=0

rem "c:\program files\Tivoli\TSM\TDPSql\tdpsqlc" backup "*" full /config=h:\tsm\tdpsql.conf
/tsmoptf=h:\tsm\dsmsql.opt
"c:\program files\Tivoli\TSM\TDPSql\tdpsqlc" backup "SDOF*" full /config=h:\tsm\tdpsql.conf
/tsmoptf=h:\tsm\dsmsql.opt
IF %ERRORLEVEL% NEQ 0 set RC=1

echo Script %0 finished with RC=%RC%
echo.
exit /B %RC%
```

TSM CLIENT SQL: SOUBOR DSMSQL.OPT

```
ASNODE            SQL-clustersql
PASSWORDAccess    Generate

*====================================================================*
* TCP/IP Communication Options                                       *
*====================================================================*
COMMMethod        TCPip
TCPServeraddress  SRV-TSM
TCPPort           1500
TCPWindowsize     63
TCPBuffSize       32

*====================================================================*
* The following include statements assign all meta objects to       *
* management class SqlDbMetaMgmtClass and all data objects to       *
* SqlDbDataMgmtClass                                                *
*====================================================================*
*INCLUDE "\...\meta\...\*"     SqlDbMetaMgmtClass
*INCLUDE "\...\data\...\*"     SqlDbDataMgmtClass

*====================================================================*
* The following include statements assign all log meta objects to   *
* management class SqlLogMetaMgmtClass and all log data objects to   *
* SqlLogDataMgmtClass                                               *
*====================================================================*
*INCLUDE "\...\meta\...\log*"  SqlLogMetaMgmtClass
*INCLUDE "\...\data\...\log*"  SqlLogDataMgmtClass

*====================================================================*
* The following exclude statements exclude all log backups for      *
* databases master and msdb                                         *
*====================================================================*

EXCLUDE "\...\master\...\log*"
EXCLUDE "\...\msdb\...\log*"
EXCLUDE "\...\sdo\...\log*"
EXCLUDE "\...\SDOFT\...\log*"
EXCLUDE "\...\SDOlog\...\log*"

EXCLUDE "\...\Gintest\...\*"
EXCLUDE "\...\dc2test\...\*"
exclude "\...\sdo\...\*"
exclude "\...\SDOFT\...\*"

SCHEDLOGNAME H:\TSM\dsmsqlsched.log
SCHEDLOGRET 3 D
ERRORLOGNAME H:\TSM\dsmsqlerror.log
ERRORLOGRET 30 D
```

TSM CLIENT SQL: SOUBOR SQL_LOGS.CMD

```
@ECHO OFF
echo Script %0 started: %DATE% %TIME%
set RC=0

"c:\program files\Tivoli\TSM\TDPSql\tdpsqlc" BACKUP "*" LOG /config=h:\tsm\tdpsql.conf /tsmoptf=h:\tsm\dsmsql.opt
IF %ERRORLEVEL% NEQ 0 set RC=1

echo Script %0 finished with RC=%RC%
echo.
exit /B %RC%
```

TSM CLIENT SQL: SOUBOR TDPSQL.CONF

LASTPRUNEDate 04/22/2012 13:27:12
SQLAUTHentication INTegrated
MOUNTWaitfordata Yes
BACKUPMethod LEGACY
DIFFESTimate 20
BUFFers 3
BUFFERSIze 1024
STRIPes 1
SQLBUFFers 0
SQLBUFFERSIze 1024
LOGPrune 60
DATEformat 1
NUMBERformat 1
TIMEformat 1
LANGuage ENU
QUERYOPTIMIZATION 0
SQLSERVer MIS
FROMSQLserver MIS

TSM SERVER: SOUBOR TSMDBMGR.OPT

nodename $$_TSMDBMGR_$$
commmethod tcpip
tcpserveraddr localhost
tcpport 1500
passwordaccess generate
errorlogname c:\tsm\TSMDBMGR.log

TSM SERVER: SOUBOR TSMDBMGR.ENV

DSMI_CONFIG=C:\tsm\tsmdbmgr.opt
DSMI_LOG=C:\tsm