
**FAKULTA INFORMATIKY A MANAGEMENTU UHK**

# Bezpečnost dat v databázích

semestrální projekt předmětu OBDAI


| | |
|---|---|
| Vypracoval: | **Petr Voborník** |
| Obor: | im(5) |
| Forma: | prezenční |
| Ročník: | 4. |
| Datum: | 1.12.2004 |
| E-mail: | vobornik@mikmik.cz |

# Obsah

# 1. Úvod

V této semestrální práci jsem se pokusil shrnout základní zásady zabezpečení dat v databázích. Práce se zabývá jak zajištěním dat před jejich odcizením nepovolanými osobami, tak zabezpečením dat před jejich fyzickou ztrátou.

Čerpal jsem převážně z vlastních vědomostí a zkušeností a samozřejmě jsem některé části přejal z odborných článků na internetu. Ty jsou vždy označeny číslem odkazu bezprostředně za takovýmto textem.

# 2. Zabezpečení před zcizením dat

Jednou z hlavních složek bezpečnosti databázových dat je jejich utajení. Ať se totiž jedná o důvěrné firemní informace či data chráněná zákonem o ochraně osobních informací nebo jen zdánlivě bezvýznamné seznamy a statistiky, většinou není žádoucí, aby se k nim mohl kdokoli kdykoli dostat. Proto je nezbytné se zabývat tím, jak nepovolaným osobám získání těchto dat znesnadnit, či ještě lépe úplně znemožnit. Znemožnění přístupu útočníkům sice asi nikdy nebude stoprocentní, přesto je třeba se snažit, aby ono znesnadnění bylo maximální. Hlavní způsoby jak tohoto docílit jsou popsány dále.

## 2.1. Ochrana šifrováním

Ochrana dat šifrováním sice nebrání útočníkovi v jejich získání, ale zapříčiní, že pokud se k nim dostane, tak budou v takovém tvaru, že z nich nedokáže nic vyčíst (neporozumí jejich obsahu).

### 2.1.1. Šifrování databázového souboru

Šifrování přímo databázového souboru (souboru nebo systému souborů, do něhož databázový systém ukládá databázová data) má smysl pouze v případě, že by se tento jako celek dostal do rukou nepovolaným osobám. To by znamenalo, že jej někdo neoprávněně zkopíroval přímo ze serveru, nebo se dostal k některé ze záloh.

K prvnímu případu (zcizení souboru ze serveru) by při dobré ochranné politice nemělo dojít, nicméně v praxi nic není nemožné a opatrnosti není nikdy dostatek. Stálé uchování souboru v zašifrovaném stavu by však znemožnilo či přinejmenším znatelně zpomalilo jeho používání, což u tohoto typu souboru není žádoucí. Pokud tuto funkci neposkytuje přímo databázový systém, pak nezbývá než použít externích programů, které by soubor zabezpečily. U nich však právě nastává ono zpomalení práce se souborem.

Ideálním řešením by bylo udržovat data souboru v nešifrovaném tvaru v dynamické paměti (alespoň ta aktuálně používaná) - například v paměti RAM nebo CACHE, kde k nim bude nepřetržitý velmi rychlý přístup, bez nutnosti dešifrování. Úpravy v datech by se pak teprve pravidelně ukládaly na pevný disk do šifrovaného souboru. Při vypnutí serveru by však došlo k okamžitému vymazání veškerých nešifrovaných dat v dynamických pamětech a tyto by museli být opět získány ze zašifrovaného databázového souboru – tedy by bylo nezbytné znát příslušné heslo pro daný program šifrující systém souborů.

Tato funkce chrání data především v případě zcizení celého serveru nebo jeho pevného disku. V případě dedikovaného serveru je třeba zajistit, aby se ale k němu nedostal útočník ve chvíli, kdy je spuštěn a přístup k souboru aktivován. Vždy je tedy nezbytné, aby byla v nepřítomnosti správce zapnuta přístupová ochrana, přes kterou

se bez znalosti hesla nelze dostat. Dále by bylo dobré používat šifrovací program, který umožňuje povolit přístup k souboru pouze jednomu procesu, tedy databázovému systému. Ten jediný by s ním pak mohl pracovat jako s „obyčejným“ souborem. Ostatní procesy by pak k souboru neměly přístup (pouze pod heslem) a ten by tak nebylo možné v čitelné podobě zkopírovat.

Ve Windows XP jsou nástroje pro šifrování souborů vestavěny přímo do systému, pro starší Windows může dobře posloužit například program PGP – Pretty Good Privacy (www.pgp.com). [2]

V případě možnosti zcizení (či jen pouhého zkopírování) zálohy databázového souboru, by mělo být samozřejmé, že tento soubor bude zálohován v zašifrovaném tvaru. Zde již není třeba rychlého přístupu k souboru, proto může být chráněn i složitějšími šiframi a elektronickými klíči.

### 2.1.2. Šifrování dat v databázi

Šifrování dat, která jsou ukládána do databáze je vcelku běžná záležitost. Většinou se o to stará přímo databázový systém. Data uložená v databázovém souboru, jsou pak jinými programy než jím nečitelná. To však spíše spadá do předchozí kapitoly.

Zde jde spíš o to, aby data nebyla čitelná ani například přes konzoly databázového systému při znalosti správného hesla k databázi. Tuto funkci má například databázový systém MySQL (funkce password). Data jsou pak uložena přímo do „tabulky“ v zašifrovaném tvaru a nejsou tak bez použití dešifrovací funkce čitelná.

Častěji se ale používá „vlastního“ šifrovacího algoritmu, který vykoná aplikace než data předá / převezme databázovému systému. Tento přístup ale značně komplikuje práci s daty, jejichž přesné znění není pak databázovému systému známo a nemůže s nimi korektně pracovat. Navíc v kombinaci s předchozím a následujícím opatřením (šifrováním databázového souboru a šifrování přenosu dat) nemá tato ochrana dat zásadní význam.

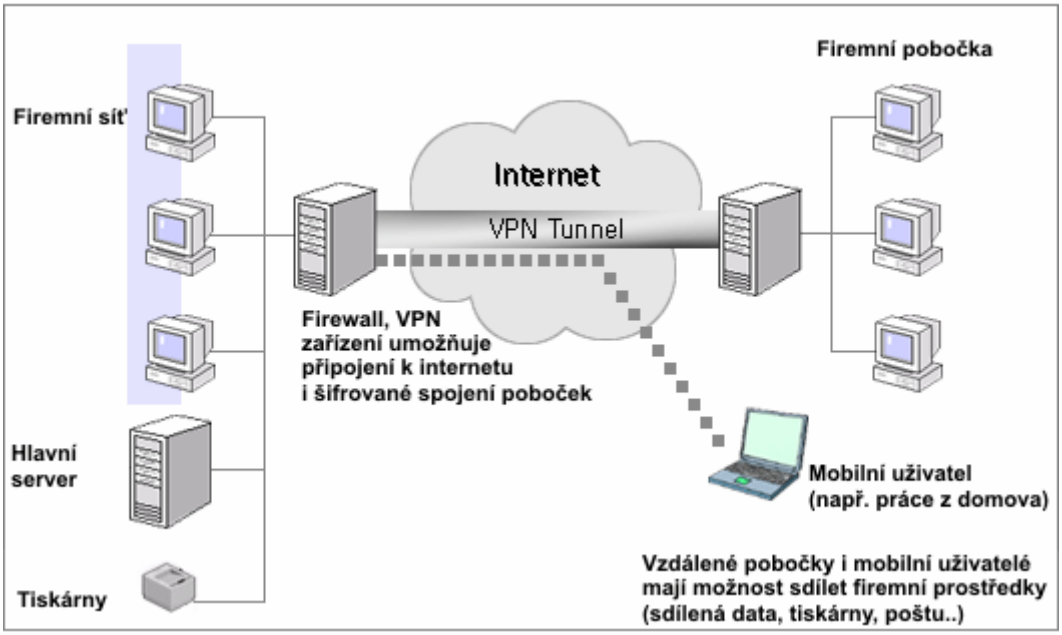
Využití šifrování dat v databázi by tak mělo být omezeno na skutečně zásadní textová data, která by měla zůstat utajena i správci databáze. Jde tedy především o přístupová hesla jednotlivých uživatelů evidovaných v databázové tabulce.

### 2.1.3. Šifrování přenosu dat

Data, která putují mezi jejich uživatelem a databází, by vzhledem k jejich možnému odposlechu měla být přenášena také pouze v zašifrovaném tvaru. V případě internetu by se k nim totiž mohl dostat v podstatě téměř úplně každý a i v případě intranetu není možnost odposlechu nikdy úplně vyloučena.

### 2.1.3.1. Komunikace přes veřejnou informační síť

Při komunikaci s využitím veřejné informační sítě internet, může být dobrým pomocníkem například použití VPN – šifrovaného tunelu, do jehož proniknutí je díky síle použité šifry jen velmi obtížné se dostat. Tuto služba je dostupná například přímo přes operační systém Windows XP, kde umožňuje připojení vzdáleného počítače přímo do podnikové sítě právě přes internet prostřednictvím VPN. Uživatel pak je jakoby přímo součástí této sítě (může do sdílených složek, má přístup k tiskárně...), přičemž vlastní pouze připojení k internetu. Přenášená data jsou však dostatečně zabezpečena. VPN ovšem samozřejmě způsobuje zpomalení oproti přímé nezašifrované komunikaci, neboť potřebuje značné procento přenesených dat navíc pro vlastní režii.

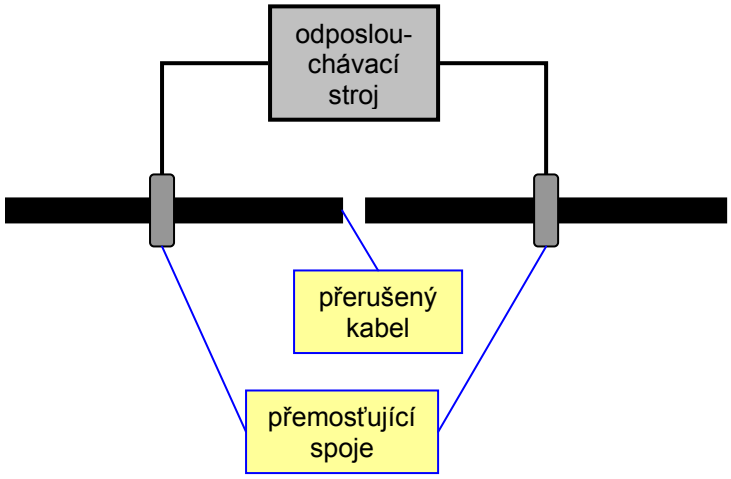



Obr. 1 [6]

### 2.1.3.2. Komunikace v intranetu

U intranetu není dobré spoléhat na to, že jelikož je celá síť od té veřejné oddělena (například serverem s firewallem) pak nemá cenu data v ní přenášená šifrovat. Existuje totiž mnoho způsobů, jak právě tuto komunikaci odposlouchávat, či do ní i zasahovat. Mezi ty nejjednodušší patří přímé připojení vlastního počítače (či obsazení nějakého k síti již připojeného). Pak je přístup k datům, ať už aktuálně přenášeným nebo i současně sdíleným, již zcela snadný. Těmto případům by však měla zabránit především kvalitní zabezpečení přístupu do „budovy“.

O něco složitějším způsobem, jak odposlouchávat přenos dat v intranetu je pak jejich přímé získávání z přenosového média – kabelu. Zde je opět řada možností, jak zjistit, co po něm právě putuje za data. Jednou z nich je přemostění kabelu přes jednotku zachycující tato data. Následující schéma zachycuje princip této činnosti.




Obr. 2

Jak je patrné z obrázku, je nezbytné, aby došlo k přerušení původního kabelu. Toto přerušení vyvolá na okamžik nedostupnost sítě, což by mohlo být signálem, že k něčemu takovému došlo. Připojit přemostění, totiž nelze současně s nepřerušeným kabelem, protože by docházelo ke zdvojení přenosu a ten by vykazoval chyby (čili signál pokusu o napojení). Prevencí by tedy mělo být dostatečné mechanické zabezpečení přenosových kabelů na každém jeho úseku, sledování výpadků sítě a upozorňování na podezřelé výpadky a především již zmíněné šifrování přenášených dat.

Kromě přemostění kabelu lze u nedostatečně odstíněných typů také sledovat elektromagnetické vlny v bezprostřední blízkosti kabelu a tak data odposlouchávat téměř bez možnosti odhalení. Tato možnost však odpadá například u optických kabelů.

Další velkou skupinou jsou dnes stále více se rozmáhající bezdrátové sítě. Bezdrátové sítě mají z bezpečnostního hlediska jednu nepříjemnou vlastnost – jejich dosah nelze jednoduše omezit. Jejich signál lze tak zachytit i v blízkém okolí mimo objekt, ve kterém je umístěna základnová stanice. Navíc s vhodnou směrovou anténou lze signál zaměřit i na daleko větší vzdálenost. Další nepříjemnou skutečností je nemožnost detekce pasivního odposlechu. Pro potencionálního útočníka jsou tak přístupové body vhodným místem pro pokus o prolomení ochrany a nabourání se do vnitřní sítě. Metod, jak toho docílit je mnoho, například již zmíněný pasivní odposlech, pasivní statistická derivace klíče, odcizení relací, induktivní derivace klíče, útok na kontrolu integrity zpráv (ICV) záměnou bitů, konfigurační útok, útok na sdílenou autentizaci, zranitelnost v EAP-MD5 a LEAP, podvržení MAC adresy, podvržení přístupového bodu, zrcadlový útok (Man-in-the-middle attack) nebo DoS útoky. [4]

Způsobů, jak bezdrátovou síť zabezpečit již existuje celá řada a jsou přímo součástí aktivních prvků těchto sítí. Mezi základní kroky, které by měli následovat bezprostředně po instalaci sítě patří změna standardního SSID (továrně zadaný

název bezdrátově obsluhované oblasti na nesnadno uhádnutelnou hodnotu), změna přístupových hesel do webové administrace zařízení, zakázat broadcast SSID (síť s deaktivovanými signálními rámci se útočníkovi skryje a on tak nezná její SSID identifikátor, čímž k ní nemůže získat přístup nebo se dokonce nedozví o její existenci), filtrace MAC adres a především opět šifrování komunikace.

#### 2.1.3.3. Nejběžnější používané šifrovací technologie

U bezdrátových sítí používajících starší šifrovací technologií v sítích 802.11 je WEP, který stejně jako novější WPA zajišťuje šifrování přenášených rámců v bezdrátové síti na druhé síťové vrstvě (MAC Layer). Šifrují se tedy veškeré datové rámce přenášené mezi přístupovým bodem a bezdrátovým zařízením bez ohledu na to, zda-li se jedná o síťovou signalizaci nebo o vlastní data. Šifrovací technologie WEP je však bohužel založena na kryptografickém algoritmu RC4 a existují nástroje pomocí kterých i amatér zvládne zachycené vysílání dekódovat a přečíst.

Lepší možností je zabezpečení pomocí WPA, které sice používá rovněž RC4, ovšem implementuje také další změny, které zvyšují celkovou bezpečnost. Určitou nevýhodou novějšího zabezpečení WPA může být skutečnost, že by jej měly podporovat všechny prvky připojené k bezdrátové síti. Současný provoz zařízení používající různá šifrování, tzv. mixed-mode, může být totiž velmi problematický.

Častým argumentem proti využití šifrovacích technologií v bezdrátových sítích je obava ze zpomalení komunikace. Tato obava je ve většině případů neoprávněná. Podle tesu několika bezdrátových zařízení (v říjnovém čísle časopisu Connect!), bylo zaznamenáno pouze 5,4% zpomalení při použití WPA-TKIP šifrování oproti naměřeným rychlostem bez šifrování. Daleko větší vliv na přenosové rychlosti má kvalita signálu. [4]

#### 2.1.3.4. Šifrování vstupních dotazů a výstupních dat

Při komunikaci s databází, abychom získali požadovaná data, je nezbytné poslat databázovému systému korektní příkaz (dotaz) v jazyku SQL. Ten pak databázový systém zpracuje, upraví podle něj data (v případě INSERT, UPDATE nebo DELETE), či odešle požadovaná data zpět uživateli (SELECT). Tyto vstupní dotazy je samozřejmě nezbytné šifrovat, neboť obsahují zásadní informace o struktuře databáze, ID požadovaných záznamů, klíčová slova a samozřejmě také při přihlašování login a heslo.

Data, která databázový systém vrací již obsahují konkrétní data z databáze a proto je přímo nezbytné aby byla při přenosu taktéž zajištěna šifrou.

### 2.1.4. Typy šifer

V této části nastíním princip dvou základních typů šifer: symetrických a asymetrických.

#### 2.1.4.1. Symetrické

Mějme dvě komunikující strany A a B. Jestliže bude chtít A zaslat B zašifrovaná data (ať už email nebo výpis z databáze), vymyslí si šifrovací klíč (posloupnost bitů – např. náhodných 128 bitů). S tímto klíčem zašifruje daný dokument a pošle jej B.

Aby B dokázal rozšifrovat takto chráněná data potřebuje k tomu šifrovací klíč, který A použil k zašifrování dokumentu. *U symetrické šifry se pro „zašifrování“ a „dešifrování“ dat používá jeden a ten samý šifrovací klíč.* (Mohou se použít dva různé klíče, ale tyto jsou na sobě jednoznačně závislé tak, že z jednoho se dá vypočítat druhý.) Symetrické šifry mají tyto *nevýhody*:

- *Obě strany musí mít daný klíč k dispozici.* Obvykle jedna strana klíč "vymyslí" a potom je potřeba tento klíč "bezpečně předat" na "druhou stranu". K tomu se dnes používá především asymetrických šifer, oba typy šifer - symetrické i asymetrické - se tak používají ve většině systémů společně.
- Druhým problém je to, že pokud máme N komunikujících stran a každá si chce bezpečně vyměňovat zašifrovaná data se všemi ostatními (tak aby je přečetl pouze adresát a nikdo jiný ze skupiny N), potom je zapotřebí celkem *N(N-1)/2 klíčů*. Zatímco u asymetrických šifer má každý účastník pouze svůj veřejný a privátní klíč, tj. je zapotřebí celkem pouze 2N klíčů.
- Pomocí symetrických šifer velmi těžko zajistíme "*nepopiratelnost*" (obě strany mají šifrovací klíč, tedy není jasné která z nich šifrovaný dokument vytvořila).

Naopak výhodou je, že symetrické šifry jsou mnohem *rychlejší, než šifry asymetrické*.

Každá šifra vyžaduje jinou délku šifrovacího klíče, nicméně u většiny symetrických šifrovacích algoritmů je minimální požadavek dneška (s ohledem na vývoj výpočetní techniky pro následujících 20 let) alespoň 80 bitů. S tím, že pro vyšší bezpečnost a praktičnost (pro programování jsou výhodné délky mocniny dvou: 64,128,256,...) se dnes doporučují délky *128 bitů*. Např. nový Advanced Encryption Standard = *AES*, kterým se stal algoritmus *Rijndael*, používá 128, 192 a 256 bitové klíče.

Příkladem symetrických šifer jsou DES, 3DES, AES, IDEA, RC4... [1]

#### 2.1.4.2. Asymetrické

Zde je základem pár klíčů. *Veřejný klíč (public key) a privátní klíč (private key).* Každý uživatel (A i B) mají jeden pár takovýchto klíčů. Tedy při použití příkladu, kdy A posílá B email budou „ve hře“ celkem 4 klíče.

Privátní klíč charakterizuje jeho vlastníka a nesmí být nikomu sdělován, posílán, atd. Naopak veřejný klíč musí druhá komunikující strana znát a v podstatě věci jej může znát kdokoli (i případný útočník).

Pár klíčů privátní-veřejný se generuje najednou – tyto klíče jsou na sobě matematicky závislé. Závislé jsou tak, že je výpočetně „velmi náročné“ při znalosti veřejného klíče, „vypočítat“ klíč privátní (pro vyšší počet bitů klíčů – např. *1048 bitů* šifry *RSA* je při současné výpočetní síle počítačů považován takový výpočet za "nemožný"). 2048 bitů šifry RSA se používá pro jistotu tam, kde je důležitost zabezpečení opravdu vysoká. A i B si tedy vygenerují každý svůj pár těchto klíčů a svůj veřejný klíč zašlou „bezpečným“ způsobem protistraně. Jestliže jsou data zašifrována privátním klíčem, lze je dešifrovat veřejným klíčem a naopak. [1]

Nevýhodou je zde fakt, že pokud osobu se kterou takovýmto způsobem po internetu komunikujeme neznáme osobně, nemůžeme si být nikdy jisti, je-li skutečně tím, za koho se vydává (již od okamžiku výměny klíčů – prvního kontaktu).

Kromě šifry RSA se lze ještě poměrně často setkat s algoritmy:

- DSA (podle standardu neumožňuje data šifrovat, ale jen digitálně podepisovat)
- DH – viz. PKCS3, ECDSA (EC = Eliptické křivky) [1]

## 2.2. Zabezpečení přístupu k serveru

Aby bylo možné databázi vůbec ochránit, je třeba zabezpečit, aby se k serveru, na kterém se nachází, nikdo nepovolaný nedostal. Ať už osobně nebo prostřednictvím sítě. Tímto se zabývají následující podkapitoly.

### 2.2.1. Mechanické zabezpečení

Mechanické zabezpečení přístupu k serveru s databázovým systémem by mělo maximálně znesnadňovat možnost fyzického přístupu k serveru. Nejlepší by asi bylo umístit server do bankovního trezoru, ovšem tato možnost většinou nepřipadá v úvahu. V každém případě by server měl být v samostatné místnosti, jejíž všechny přístupové cesty budou dostatečně zabezpečeny. To znamená, silné zdi, v oknech mříže (či ještě lépe okna žádná), vyšší patro než přízemí, bytelné dveře (nejlépe dvojité nebo s mřížemi) a malá či zamřížovaná větrací šachta. Vstup do místnosti by měl být chráněn minimálně dvojím způsobem (dva zámky, nebo lépe kód a zámek).

V místnosti by také mělo být poplašné zařízení s detekcí pohybu, které by v případě narušení spustilo poplach. Nemělo by ani chybět monitorovací zařízení (kamery) s přenosem na kontrolní monitor hlídače objektu. To by ovšem nemělo ani vzdáleně zachycovat záběrem klávesnici, kde je zadáváno přístupové heslo! Celá budova by měla být taktéž dostatečně zabezpečena již ve svém nejbližším okolí (plot, kamery, hlídací psi) a samozřejmě i zajištěny všechny vchody do ní (hlavní vchod, vedlejší vstupy, požární schodiště, okna v přízemí...).

Server by pak mohl být zajištěn ještě samostatnou ochranou v rámci místnosti, například trezorem, mřížemi, nebo jen uzamknut k podlaze. Jeho ovládací prvky (klávesnice, myš, monitor), by také měly bát dostatečně zajištěny, včetně jejich připojení k serveru.

### 2.2.2. Personální zabezpečení

Zatímco mechanické zabezpečení chrání server spíše v noci, kdy na blízku nejsou většinou žádní svědkové, personální opatření má chránit server spíše během pracovní doby. Tato opatření by měla omezovat přístup serveru pouze na nezbytné případy a pouze povolaným lidem.

#### 2.2.2.1. Povolený přístup

Přístup do místnosti se serverem, by měl být omezen na minimální počet osob, přičemž by všechny měly být dostatečně prověřeny o jejich důvěryhodnosti a poučeny o důležitosti svého privilegia. Seznam osob, majících přístup by měl zahrnovat pouze administrátora, ředitele firmy nebo pobočky, případně jeho zástupce. Na druhou stranu však není dobré, aby měl do místnosti přístup pouze jediný člověk, neboť při jeho indispozici by byl přístup k serveru znemožněn i při závažných důvodech. Rezervní klíč a kód ke dveřím je dobré pro nejhorší případy uchovávat v záloze na bezpečném místě (třeba v bance).

Kód ke dveřím, jsou-li takto zabezpečeny, by měl být pro každou osobu mající přístup jiný, aby se mohlo zaznamenávat, kdo a kdy k serveru přistupoval (podobně jako je tomu u logu).

Další ochranné prvky, jako je otisk prstu, hlasová kontrola, skenování oční sítnice či kontrola DNA jsou jistě také vítané, ale na České poměry dosti finančně nákladné. A i tak nejsou nepřekonatelné (útočník se snadněji zmocní palce pro otisk, než kódu uloženého v hlavě).

Co se týče údržby serveru a úklidu místnosti, tak pokud se tomu nehodlá propůjčit přímo administrátor, musí každé takové akci alespoň osobně přihlížet a být po celou její dobu osobně přítomen. Po skončení by před opětovným uzamčením místnosti měl vše zkontrolovat, nedošlo-li neodbornou manipulací k narušení bezpečnostních prvků.

#### 2.2.2.2. Hlídači

Kromě nejrůznějšího mechanického a monitorovacího zabezpečení je také žádoucí zajistit i tzv. „inteligentní ochranu“ v podobě lidského faktoru. Osoby střežící přístup k serveru ve dne v noci by byly ideální, ale jistě finančně velmi nákladní. Ve většině případů je s funkcí hlídače serveru spojeno i střežení celé budovy. Hlídač samotný by pak přístup přímo do místnosti se serverem neměl mít umožněn. Ovšem monitor zobrazující dění uvnitř by mít přístupný měl.

Hlídač by také neměl být jeden jediný, ale minimálně dva. Hlídají se tak na vzájem a v případě potřeby jsou operativnější a obranyschopnější. Při výběru osob do funkce hlídačů je opět třeba postupovat s nejvyšší opatrností a uchazeče si dostatečně prověřit.

### 2.2.3. Síťové

Kromě nebezpečí fyzického přístupu k serveru je zde také možnost vzdáleného přístupu k serveru po síti. Útočník se tak může dostat nejen k datům databáze, ale případně i ovládat celý server, jako kdyby byl fyzicky přímo u něho. Při nedostatečném zabezpečení této přístupové cesty nám pak neprolomitelné mechanické zabezpečení moc nepomůže, proto je nezbytné věnovat mu přinejmenším stejnou pozornost.

Zabezpečení přenosu dat šifrováním bylo podrobně probráno v předchozí kapitole, proto se nyní budeme věnovat spíše zabezpečení samotného přístupu.

#### 2.2.3.1. Operační systém

Všechny dnešní serverové operační systémy (OS) mají celou řadu možností, jak je zabezpečit proti útočníkům po síti. Mezi nejčastěji používané OS na serverech v současnosti patří Windows (v poslední verzi 2003 server), Unix a Linux. Mezi základní bezpečnostní vlastnosti těchto OS patří fakt, že veškeré možnosti komunikace se serverem jsou ve výchozím stavu zablokovány a záleží pouze na administrátorovi, které z nich povolí, a které tím pádem bude také muset zabezpečit. Tato vlastnost v minulosti, zejména u Windows, byla spíše na opak – povoleno bylo kde co a administrátor musel blokovat co povolit nechtěl, což bylo příčinou řady bezpečnostních děr.

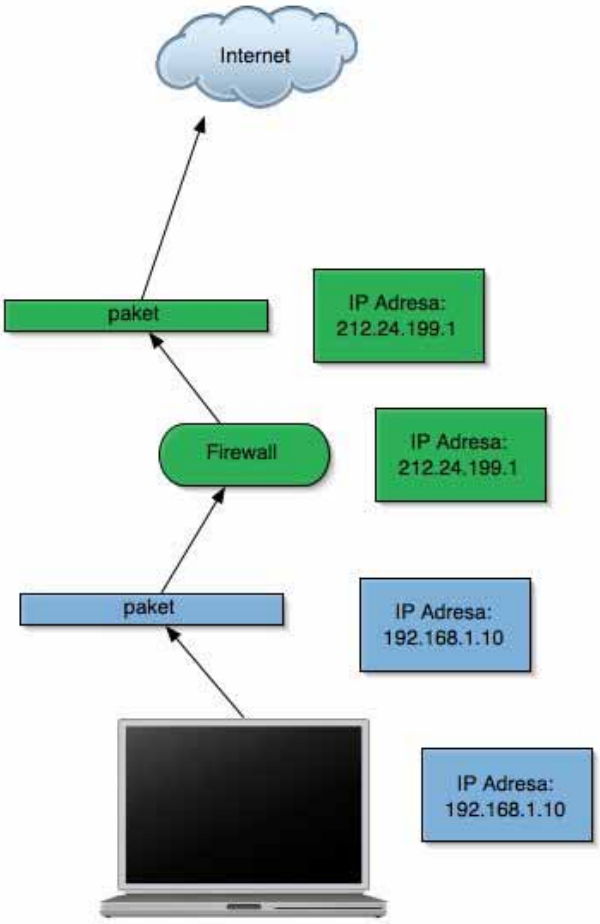
Hlavní obranou bývá správné nastavení firewallu či zprostředkovaného přístupu, což bude detailněji probráno v následujících podkapitolách. Důležité je také co nejčastěji updatovat OS a instalovat tak všechny vyšlé záplaty na postupně objevované bezpečnostní díry. Tyto záplaty však většinou vycházejí až po nějakém čase od objevení a zneužití nějaké chyby systému, proto by měl správný administrátor sledovat fóra a konference týkající se jím spravovaného OS a v případě objevení se informace o nějaké nově nalezené bezpečnostní díře, ji okamžitě zabezpečit dostupnými prostředky, do doby než na ni vyjde oficiální záplata.

#### 2.2.3.2. Firewall

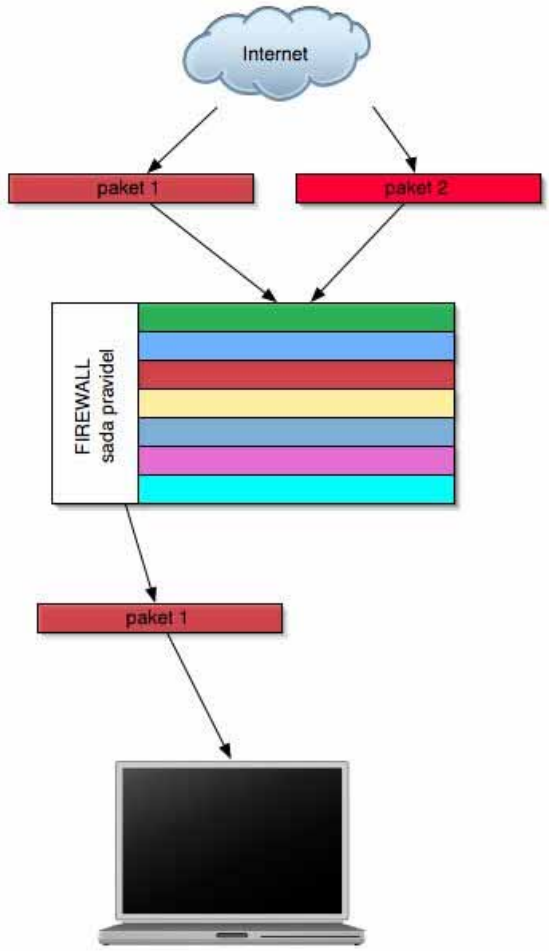
Firewall je hardwarové zařízení nebo program (případně kombinace obojího), který "stojí" mezi daným počítačem a Internetem. [7] Jeho hlavní vlastností je, že do vnitřní sítě (potažmo ani z ní) nevpustí žádný paket, který nevyhovuje sadě pravidel v něm uložených. Projít pak mohou pouze pakety vyžádané zevnitř, nebo pakety ze spojení navázaného z povolené IP adresy a se správným heslem. Takto je server a i celá jeho případná vnitřní síť chráněna před útoky z vnější sítě (internetu).

Obecně jsou dva principy firewallů. Jeden pracuje na základě *paketového filtru* a druhý je *proxy server*. Paketový filtr má tedy nastavenu sadu pravidel, a pokud přijde nějaký paket, tak ho otestuje oproti této sadě. Pokud paket vyhovuje, projde,

pokud ne, je zahozen. Proxy server funguje tím způsobem, že schovává počítače na vnitřní síti, takže veškeré požadavky, které vznikají mají identifikaci proxy serveru. [7]




Obr. 3 [7]




Obr. 4 [7]

### 2.2.3.3. Zprostředkovaný přístup

Zprostředkovaného přístupu se využívá především přímo u databází. Používá se zde jednoho počítače jako mezistupně komunikace mezi serverem a veřejnou sítí, podobně jako u firewallu. Tento počítač však má narozdíl od firewallu vlastní OS s vlastním databázovým systémem, který dokonce ani nemusí být totožný s databázovým systémem na hlavním serveru. Veškerá komunikace pak jde přes tento mezistupeň. (Tento počítač by však měl být umístěn až za firewallem, aby byl také chráněn před neoprávněnými přístupy zvenčí.) Požadovaná data, projdou-li autentizační kontrolou jsou, jsou vlastním SQL dotazem tohoto systému získána ze serveru (přes opětné ověření pravomocí k nim), uložena do databáze na tomto počítači a pak teprve předána uživateli. Zásah do úpravy dat probíhá obdobným způsobem.

Výhoda tohoto postupu je zřejmá – uživatel v podstatě pracuje s kopií dat na počítači mezi serverem a ním, takže je opět minimalizováno nebezpečí jejich zneužití. Přitom však pracuje vždy pouze s daty aktuálními. Nevýhodou je pak

zpomalení komunikace a komplikace zprávy serveru po síti, v případě, že se jedná o k tomuto oprávněného uživatele. To je zároveň bezpečnostní výhoda i nevýhoda. Je zajištěno, že této možnosti nezneužije nikdo cizí, ovšem v případě nutné potřeby, třeba odstavit napadený server na dálku (administrátor je třeba momentálně od serveru několik hodin cesty) není toto možné.

Existuje-li totiž cesta, nedá se nikdy na 100% zaručit, že po ní nepůjde i nikdo jiný. Jedinou možností je neexistence této cesty. Pak po ní ovšem nemůžeme jít ani my.

## 2.3. Bezpečnostní politika uživatelů

Abychom vůbec mohli využívat komunikace s databázovým serverem, musí možnost této komunikace existovat. Existuje-li, nelze ovšem nikdy zcela vyloučit její zneužití. I přes její maximální zabezpečení je však vždy nezbytné, aby ji alespoň někdo využívat mohl. A právě tyto osoby nesmí brát toto privilegium na lehkou váhu a musí mít možnosti a znalosti, jak zabezpečit, aby svůj přístup mohli využívat pouze oni a nezneužil jej nikdo cizí. To je zabezpečeno většinou omezením přístupových bodů a především individuálními hesly. O základních pravidlech jejich užívání pojednává tato kapitola.

### 2.3.1. Školení o práci s hesly

Každá osoba, která má k databázi přístup, ať již jako uživatel databáze nebo správce serveru, na kterém je databáze umístěna, musí mít tento přístup zabezpečen heslem. Protože ne každý uživatel má povědomí o bezpečném zacházení s hesly, měly by jim tyto zásady být osvětleny dříve než jim heslo bude svěřeno.

Pravidla zacházení s heslem by měla být součástí interního nařízení firmy, aby k nim měl každý uživatel stále přístup a nemohl se případně vymlouvat na jejich neznalost. Krátké doprovodné školení by pak mělo zajistit, aby skutečně každý uživatel věděl co je po něm požadováno (pro případ, že by psaná pravidla přešel s tím, že je to jasné). Hlavními body školení práce s hesly by měly být:

- správná volba hesla (složení, délka)
- častá obměna hesla
- uchovávání hesla (pouze ve vlastní paměti)
- nesdělovat heslo žádným dalším osobám
- bezpečné zadávání hesla (kde, kdy, jak)
- postihy za porušení těchto pravidel

### 2.3.2. Preventivní nucená změna hesel

I když uživatel se svým heslem zachází nanejvýš opatrně, možnost jeho vyzrazení narůstá s dobou jeho používání. Je proto žádoucí, aby každý uživatel své heslo v pravidelných intervalech měnil. Uživatelé se však k těmto změnám většinou staví spíše negativně, neboť se jen neradi nazpaměť učí řekněme každý měsíc novou posloupnost nic nevyjadřujících znaků. Proto není dobré spoléhat na jejich uvědomělost a s využitím prostředků integrovaných v systému je k periodické změně hesla „donutit“.

Všechny dnešní operační systémy již nabízejí prostředky zavedení takovéto nucené změny hesla. Po přihlášení uživatele, který si již danou dobu heslo nezměnil, ho pak nejprve „donutí“ (aby vůbec mohl pokračovat dál) změnit si heslo. Zároveň dokáží ohlídat, aby uživatel jako nové heslo nezadal to staré (i několik změn dozadu) a aby jej zvolil alespoň relativně bezpečně (musí obsahovat malá i velká písmena, číslice a speciální znaky, případně na něj zkusí i aplikovat slovníkový útok).

### 2.3.3. Postihy za únik hesla

Jedním ze způsobů, jak motivovat zaměstnance, aby se svými přístupovými hesly zacházeli s nejvyšší opatrností, je hrozba postihu za případné zneužití systému jejich heslem. Tedy pokud někdo ze zaměstnanců nedostatečně zabezpečí své heslo a to se tak dostane do rukou útočníka (škůdce), který pak v databázi napáchá škody (či jen získá tajná data), bude osoba, jejíž heslo bylo k tomuto zneužito nějak trestně postihnuta (pokutou, změnou místa, ztrátou zaměstnání, soudní žalobou apod.).

Toto opatření, ač je uplatněno až po vzniku škody, je spíše preventivní. Proto by varování o případném postihu mělo být zaměstnancům sděleno již při nástupu do zaměstnání a pak co nejčastěji připomínáno spolu s patřičným školením o práci s hesly. To by mělo zabránit laxnímu přístupu k heslům (volit hesla snadno odhadnutelná, psaní si jej na papírek...), vyzrazení jej jakékoli další osobě (včetně zdánlivě důvěryhodným) a při sebemenším podezření o úniku hesla jeho majitele donutit k jeho okamžité změně.

### 2.3.4. Sledování logů

Samozřejmostí jak u databázového serveru tak u databázového systému je zaznamenávání všech přihlášení všech uživatelů do tzv. logu – souboru s tabulkou obsahující datum, čas a délku trvání připojení každého uživatele, spolu s jeho uživatelským jménem a IP a MAC adresou, z níž bylo připojení uskutečněno. Dále jsou zde zaznamenány objekty (soubory, tabulky, záznamy...), ke kterým uživatel přistupoval (četl je nebo měnil). Z tohoto souboru je pak snadno zjistitelné, kdo, kdy kde a co na serveru a v databázi prováděl.

Tento soubor je samozřejmě nutné dostatečně zabezpečit a často jej automaticky zálohovat, neboť případný útočník by se mohl pokusit záznam o svém přístupu na server z logu odstranit, nebo jej dokonce smazat celý.

V případě útoku na data je log tím prvním (a většinou i jediným) vodítkem k dopadení pachatele. V případě několikanásobného přístupu přes různé servery pak stačí získat informace z logů i těchto serverů a tak se i zpětně dopátrat výchozí stanice útočníka.

Nevýhodou logů bývá množství dat v něm uložených a z toho plynoucí nepřehlednost. Pro hledání konkrétních informací však existují nástroje, které umožňují získání požadovaných skutečností. Například je pak možné bleskově získat odpovědi na otázky typu „kdo všechno se přihlásil k databázi dnes mezi 12 a 13 hodinou“, „ukaž graf všech přihlášení uživatele XY za poslední půlrok“, „o jaká data se zajímal uživatel XY minulý čtvrtek“, „kdo smazal záznamy o zaměstnanci XY“, „kdo zvýšil všem dělníkům platy o 500%“ apod.

## 2.3.5. Definice práv při práci s databází

Zabezpečení dat přístupovými právy přímo na úrovni databázového systému s využitím v něm implementované podpory. Rozlišujeme dva základní druhy tohoto zabezpečení přístupových práv a to Discretionary control a Mandatory control.

### 2.3.5.1. Discretionary control

Volitelné řízení přístupu. Každému uživateli zvlášť je nastaveno k čemu v databázi má jaký přístup. Všechny tabulky v databázi totiž mají čtyři metody práce s nimi. Jde o:

- čtení
- vkládání nových záznamů
- úpravu dat existujících záznamů
- mazání záznamů z tabulky

Každou z těchto metod lze individuálně každému uživateli povolit či zakázat pro každou tabulku. Většina databázových systémů má takovouto ochrannou politiku přímo implementovánu v sobě a proto je také dobré ji využívat.

Nejobvyklejší chybou je přistupování k databázi přes aplikaci pod jednotným uživatelským jménem a heslem, přičemž o kontrolu oprávněnosti manipulace s daty se stará pouze aplikace. Tu však lze samozřejmě snadno obejít například použitím konzole příslušného databázového systému a s heslem zjištěným z aplikace pak takovýto útočník získává neomezený přístup k veškerým datům. Nehledě na to, že z logu databáze se pak dá jen těžko vyčíst, který uživatel se kdy k databázi připojil.

V případě zajištění individuálního přístupu a oprávnění každému uživateli databáze zvlášť pak z aplikace přístupové heslo k databázi zjistit nelze, neboť v ní

není nikde uvedeno a za pomoci konzole se pak útočník se svým uživatelským jménem dostane vždy pouze k těm datům, ke kterým měl přístup i přes aplikaci.

Přiřazování práv uživatelům na jednotlivé tabulky se provádí většinou tímto SQL příkazem:

```
CREATE SECURITY RULE rule
     GRANT privilege-commalist
     ON expresion
     TO user-commalist
     ON ATTEMPTED VIOLATION action

DESTROY SECURITY RULE rule [8]
```

#### 2.3.5.2. Mandatory control

Povinné řízení přístupu. Databázové objekty jsou členěny do předem daných bezpečnostních úrovní a jednotliví uživatelé jsou zařazeni do skupin s příslušnými přístupovými právy. Jedná se o využití principu víceúrovňové bezpečnosti (multilevel security) v databázovém prostředí. Vhodnější se z tohoto pohledu jeví relační modely. V objektově orientovaných databázích nemá zatím koncepce vlastnictví objektů jednoznačnou interpretaci.

Datové objekty mají přiřazen tzv. *classification level* a uživatelé mají *clearance level*, přičemž jednotlivé úrovně jsou striktně uspořádány. Uživatel pak smí vidět objekt pouze tehdy, pokud je jeho *clearance level* $\geq$ *classification level* databázového objektu. Modifikovat jej však může pouze v případě, že se obě úrovně rovnají (*clearance level* = *classification level*). [8]

# 3. Zabezpečení před ztrátou dat

Jsou-li databázová data zabezpečena před jejich zcizením neoprávněnými osobami, musíme je také zabezpečit před tím, abychom o tato data nepřišli. O způsobech, jak tohoto docílit, pojednává tato kapitola.

## 3.1. Zálohování

Nejzákladnější a také nejúčinnější ochranou dat, před jejich ztrátou, je jejich zálohování. Základem úspěchu této metody je dodržování následujících pravidel:

- Zálohovat by se mělo pravidelně
- Záloha by měla být na jiném typu média, ideálně tak, aby nešla smazat
- Zálohovaná data by měl být umístěna v jiné místnosti, než zálohovaný počítač
- Zálohovací média by měla být udržována v čistotě, nejlépe za stálé teploty. To platí především u elektro-magnetických typů záznamů, třeba u pásek. [7]

V případě databáze lze zálohovat celý databázový soubor, případně extrahovat data přímo z databáze za pomoci databázového systému. Záloha celého databázového souboru však většinou znamená nutnost pozastavení nebo dokonce vypnutí databázového systému, který se souborem neustále pracuje (má jej otevřený a zapisuje do něho). To by však znamenalo nežádoucí výpadek (ač třeba jen velmi krátký) celého systému, který nad databází pracuje, což bývá většinou nežádoucí.

Druhý způsob (extrakce dat) je proto nesporně výhodnější, neboť provoz systému tak maximálně dočasně zpomalí. Další výhodou je, že ne u všech databázových systémů lze korektně databázi z „násilím“ do ni nahraného souboru obnovit. Taktéž je možné zálohu dat získat i z libovolné klientské stanice, nikoli pouze ze serveru. V případě této možnosti je však nezbytné dbát zásad z předchozí kapitoly a předejít tak získání dat nepovolanou osobou. Databázové systémy většinou možnost exportu dat (včetně extrakce metadat) nabízejí jak pro jednotlivé tabulky, tak pro databázi jako celek, se všemi jejími prvky (tabulky, triggery, generátory, uložené procedury, viewy, nastavení uživatelských práv...) do jediného souboru.

### 3.1.1. Zabezpečení záloh před zcizením

Provedené zálohy je samozřejmě nezbytné v duchu předchozí kapitoly dostatečně zabezpečit, před jejich zcizením nepovolanými osobami. Samozřejmostí by mělo být zašifrování zálohovaného souboru, jako prvního aktu, před jeho jakoukoli další manipulací. Zde již není třeba brát ohledy na rychlost dešifrování a proto není důvodu nepoužít delšího šifrovacího klíče (2048 bitů a víc).

Zde je třeba poznamenat, že použijeme-li vlastní šifrovací algoritmus (vlastní program), který není obecně rozšířen, ztížíme tím sice potencionálnímu útočníkovi práci, nicméně musíme zabezpečit, abychom se v případě globálnější ztráty dat (např. při úplném zničení serveru i všech firemních počítačů), k tomuto šifrovacímu programu měli jak dostat. Přikládat jej přímo k záloze dat není jistě nejlepší, ale musíme jeho záloze věnovat přinejmenším stejnou péči jako samotným datům.

Dále je nezbytné zálohu dat dostatečně zajistit, jak před jejím zcizením, tak před jejím zničením. Určitě by se neměla nacházet na stejném místě jako samotný server. Ideální by bylo její umístění v bankovním sejfu, nebo alespoň v sejfu jiné pobočky firmy.

## 3.2. Duplikace

Duplikace dat je v podstatě jednou z forem jejich zálohování. Probíhá ovšem ve stejném čase jako zápis do samotné databáze. Jde v podstatě o zdvojení pevných disků, které oba najednou vykonávají naprosto stejné operace a data jsou tak nepřetržitě chráněna proti možnosti selhání jednoho z nich. V případě poruchy pak systém automaticky data na tomto disku buď opraví nebo začne využívat pouze disku druhého, přičemž upozorní administrátora na vzniklou poruchu a ten pak neprodleně opraví nebo vymění poškozený harddisk.

## 3.3. Replikace

Pojem replikace označuje kopírování dat z databáze do více míst. Kopie v jednotlivých místech se nazývají repliky.

V případě replikace musí být zajištěno, že pokud dojde ke změně jednoho řádku relace v jedné replice, musí se odpovídajícím způsobem změnit tento řádek i ve všech ostatních replikách.

Replikace poskytuje uživatelům lokální aktualizované kopie dat. Replikace má řadu výhod. Je efektivnější mít potřebná data replikovaná v místě zpracování dotazu než provádět spojení tabulek rozmístěných v několika místech sítě. Jiným vhodným způsobem využití je zálohování v reálném čase, kdy je na jiném místě udržována kopie databáze (nebo její části).

Replikovat je možné jednotlivé tabulky nebo část databáze, která vznikne jako výsledek dotazu. Výsledek dotazu je zpravidla kopie read-only, resp. momentka (snapshot). Momentky jsou výsledky dotazu, do kterých se periodicky promítají aktualizace řízené z jednoho místa. Jednotlivá místa mohou momentku číst, nemají ji ale právo aktualizovat. Nejčastěji se používají momentky pouze nad jednou tabulkou.

Replikace umožňuje například:

- zvýšení dostupnosti dat - kopie dat jsou přímo uloženy v různých místech sítě, případný výpadek jednotlivých míst nebo komunikační sítě nebrání v činnosti zbytku systému
- zvýšení rychlosti a propustnosti aplikace - použití lokálních dat snižuje síťový provoz
- omezenější provoz sítě
- zajištění kompletní nezávislost - každé z míst může spravovat svá vlastní data
- levnější přístup k distribuovanému zpracování dat

Replikační server podporuje vytváření a automatickou synchronizaci kopií dat. Tyto servery jsou navrženy pro práci v prostředí globálních sítí (WAN) a umožňují replikovat data mezi datovými zdroji různých SŘBD (heterogenní prostředí).

Úkoly replikačního serveru:

- popis replikovaných dat a procedur
- doprava replikovaných data na místo, kde jsou potřebná a když jsou potřeba
- inicializace dat v replikovaných tabulkách
- přenos změn primárních tabulek do replikovaných kopií
- ochrana integrity dat
- synchronizace všech kopií po chybě
- záznam nedokončených transakcí [8]

## 3.4. Databázová ochrana

Databázové systémy již sami o sobě disponují řadou nástrojů, které dokáží data ochránit před některými nejčastějšími problémy.

### 3.4.1. Zotavení se z chyb

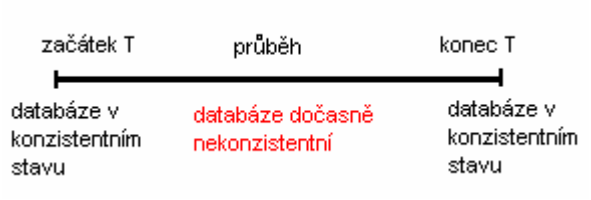
V každém systému a tedy při práci databázové aplikace může dojít k poruchám a chybovým stavům. Chyby mohou být buď hardwarové (např. zhroucení disku, porucha na komunikační lince) nebo programové (chyba v aplikaci nebo v operačním systému). Dojde-li v průběhu zpracování programu k chybě, může být databáze v chybném stavu. Příčinou chybového stavu je například skutečnost, že nedošlo k dokončení prováděných operací. Proto musí dojít k zotavení databáze, tzn. navrácení do korektního stavu.

Základním principem, na němž je zotavení založeno, je redundance (nadbytečnost dat). V tomto případě se jedná o redundanci na fyzické úrovni, která není viditelná na logické úrovni (na logické úrovni je při návrhu schématu databáze spíše snaha redundanci eliminovat). Filosofie redundance na fyzické úrovni je postavená na myšlence: *Má-li být databáze obnovitelná, musí být každý kousek informace rekonstruovatelný z jiných informací uložených jinde v systému.* [8]

### 3.4.2. Transakce

Pro zotavení z chyb se používá transakce. Transakce je programová jednotka. Jedná se o posloupnost akcí (čtení, zápis, výpočet s daty uloženými v databázi), se kterou se zachází jako s jedním celkem. Každá transakce se může skládat z několika UPDATE operací. V transakčním zpracování je tedy transakce *logická jednotka práce*.

Hlavním smyslem transakce je zachovávat konzistenci databáze. Každá transakce transformuje databázi z jednoho konzistentního stavu do jiného konzistentního stavu. V jejím průběhu může být konzistence databáze dočasně porušena.




Obr. 5 [8]

Transakce musí skončit v konečném čase. Transakce se buď provede celá, anebo vůbec.

Ideální případ, kdy se provedou všechny operace transakce, nelze zajistit (např. systém spadne mezi dvěma operacemi v důsledku výpadku elektrického proudu). Transakční zpracování ale zaručuje, že pokud nebyla celá transakce úspěšně dokončena (zhroutila se z důvodu nějaké chyby) budou všechny změny, které provedla, vráceny zpět. Proto lze na transakci pohlížet jako na atomickou operaci. [8]

## 3.5. Záložní zdroj energie

Server je také nezbytné ochránit před zrovna ne málo pravděpodobnou možností vyhoření v důsledku přepětí v síti nebo náhlému výpadku elektrické energie. K tomu slouží zařízení zvaná UPS.

Anglická zkratka UPS, která se pro záložní zdroje u počítačů poměrně často používá, pochází z anglického Uninterruptable Power Supply neboli nepřerušitelný zdroj energie. UPS je poměrně zásadním a navíc nepříliš drahým pomocníkem a to nejen tam, kde jsou časté výpadky elektrické energie.

UPS může mít hned několik funkcí. Tu základní, neboli zdroj elektrické energie v době výpadku napájení z elektrorozvodné sítě, najdeme u všech UPS. Další, poměrně běžnou schopností UPS, je vyrovnávání přepětí a podpětí v elektrorozvodné síti – to se může řešit buď pomocí elektronického zvýšení či snížení výstupního napětí bez asistence baterií (u line-interactive) a nebo přepnutím na napájení z baterií (u off-line a line-interactive). Některé UPS si pak umějí také poradit s výraznějším rázovým přepětím (např. s bleskem). UPS dovedou ochránit

před přepětím také datovou síť či odfiltrovat přepětí, které by se mohlo do serveru dostat přes datový kabel či veřejnou telefonní síť, velká část z nich je pak také vybavena slotem pro přídavné moduly, které nabízejí například možnost vzdálené správy přes SNMP protokol, dodatečné monitorovací funkce nebo třeba přímé ovládání pomocí modemu bez asistence počítače. [5]

## 3.6. Antivirová ochrana

Téma ochrany dat před viry by vydalo na samostatné dílo, proto jej zde zmíním pouze velmi stručně a obecně.

Samozřejmostí každého serveru by měla být dostatečná antivirová ochrana. Tu by měl zabezpečovat jeden pečlivě vybraný antivirový systém (více současně nedělá dobrotu), který je na tolik komplexní, že pokryje veškeré funkce serveru (nejen ty databázové). Jedná se například o kontrolu průchozích souborů, SMTP a POP3 server, firewall apod.

Z hlediska databázového je předně třeba zabránit průniku viru na samotný server. Ten by mohl oslabit jeho ochranu, či případně nějak přímo způsobit škody na databázovém souboru a případnou ztrátu dat.

Vir může do databáze proniknout i „neškodně zapouzdřený“ jako databázová data typu BLOB nebo se skrýt do filesystému (v případě databázové aplikace, která dovoluje uživatelům manipulovat se soubory). Ač by takovýto „vyčkávající“ virus neměl bez své aktivace způsobit žádné škody, neměl by kvalitní antivir jeho proniknutí na server vůbec dovolit, ať již v jakékoli formě.

# 4. Zabezpečení relevantnosti dat

Další oblastí, kde je třeba databázová data zabezpečit, je jejich relevantnost (smysluplnost, informační hodnotu). Tedy aby data samotná dávala smysl a bylo možné z nich získat nějaké informace. Tato zabezpečení zajišťuje systém pravidel integrovaný do databázového systému, přičemž pravidla samotná jsou součástí přímo konkrétní databáze. O jejich správné nastavení se stará tvůrce databáze, potažmo tvůrce celého systému (aplikačního), který nad databází pracuje.

## 4.1. Integrita dat

Integrita dat zabezpečuje přesnost a korektnost dat v databázi. Na rozdíl od ochrany dat není integrita uživatelsky závislá. Pravidla, která ji tvoří platí pro všechny kategorie uživatelů naprosto stejně.

Integrita je definována množinou pravidel, která se nazývají integritní omezení. Integritní omezení (IO) jsou tvrzení, která mají platit o datech v databázi. IO by měla být v úzkém vztahu k tvrzením na konceptuální úrovní, která platí o objektech reálného světa. IO zabezpečují, že data uložená v databázi jsou odrazem reality. Někdy se mohou nazývat také "bussines rules".

Aby tedy mohla být zajištěna integrita dat, musí být známa pravidla (IO), která uživatel nesmí narušit. Tato pravidla musí být specifikována. Definici IO provádí zpravidla návrhář databáze (data administrator) při návrhu schématu databáze.

Protože IO se ověřují neustále, musí být v databázi zaznamenána. Jsou v zapsána v jazyku databázového systému a uložena v systémovém katalogu. Databázový systém pak sleduje veškeré uživatelovi operace a kontroluje, zda při nich nedochází k narušení IO. K tomuto účelu musí mít databázový systém integritní subsystém, který sleduje uživatelovy UPDATE operace (INSERT, UPDATE a DELETE) a zjišťuje, zda nebyla některá IO narušena. [8]

### 4.1.1. Doménová pravidla

Doménová pravidla definují množinu hodnot přípustných pro danou doménu. [8] Vymezují v podstatě typ hodnoty, která je pro danou doménu přípustná (např. pouze celé číslo, desetinné číslo, znak, text, datum atd.).

### 4.1.2. Atributová pravidla

Atributová pravidla specifikují doménu pro daný atribut. Nepoužívá se pro ně žádný explicitní příkaz CREATE ATTRIBUTE INTEGRITY RULE, ale jsou specifikovány přímo při specifikaci atributů.

Jméno pravidla je stejné jako jméno přiřazované domény. Při každé UPDATE operaci se provádí bezprostřední kontrola hodnoty atributu, pokud je atributové

pravidlo porušeno (pokoušíme se do sloupce v tabulce vložit jinou než povolenou hodnotu), je operace zamítnuta.

Zrušení pravidla se provádí automaticky při zrušení příslušného atributu. [8]

### 4.1.3. Relační pravidla

Relační pravidla jsou pravidla, která se týkají atributů jedné relace. [8] To znamená, že kontrolují vzájemné vazby mezi záznamy různých tabulek a v případě změny některého z provázaných záznamů zajistí, aby i ty ostatní na něm závislé zůstaly konzistentní.

Kontrola pravidla se provádí bezprostředně při každé UPDATE operaci. [8]

### 4.1.4. Databázová pravidla

Databázová pravidla umožňují definovat omezení, která spojují dvě nebo více různých relací (tabulek) v databázi. V rámci definice omezujících podmínek musí být definována i podmínka spojení tabulek.

Tato IO se nekontrolují bezprostředně, jejich kontrola se odkládá na skončení transakce (COMMIT) a výchozí akcí při porušení pravidla není REJECT(odmítnutí operace) ale ROLLBACK (zrušení celé transakce). [8]

# 5. Závěr

Každou ze zde uvedených metod zabezpečení dat je pro dosažení požadovaného úspěchu nutno používat ve spojení s ostatními, neboť řetěz je pevný stejně jako jeho nejslabší článek. Z tohoto důvodu je nezbytné každému z nich věnovat dostatečnou pozornost a úroveň bezpečnosti stanovit patřičně vysoko.

Zvláštní důraz bych kladl především na dostatečně časté a komplexní zálohování dat, neboť ač tuto neustále opakovanou a připomínanou zásadu všichni dobře znají, ne vždy ji dodržují. Dále je třeba dostatečně zabezpečit přístupy jednotlivých uživatelů, neboť útok po síti, kdy přes jedno heslo lze všechno, je tím nejsnadnějším a nejnebezpečnějším co může nastat.

Lepší databázové systémy většinou poskytují celou řadu přímo do nich integrovaných zabezpečovacích prvků, načež není důvod jich v plné míře nevyužít. Taktéž je důležité správně nastavit zabezpečení operačního systému a teprve potom přidávat další zabezpečovací prvky (antivir, firewall, šifrovací zařízení...).

Doporučení zmíněná v této práci jsou samozřejmě přímo závislá na důležitosti a druhu zabezpečovaných informací a v neposlední řadě i na finančních možnostech firmy, spravující databázi. Ne vždy může být server umístěn v nedobytné místnosti se dveřmi jako bankovní trezor a ozbrojenou stráží před nimi. Každopádně je ale důležité věnovat zabezpečení serveru v rámci možností maximální možnou péči a v případě potřeby se nebát tato opatření konzultovat s odborníky.

# 6. Přílohy

## 6.1. Glosář

| Výraz | Vysvětlení |
|---|---|
| Atribut | Jedna hodnota záznamu – sloupec v databázové tabulce. |
| BLOB | Nedefinovaný typ (doména) atributu libovolné velikosti (vhodné pro dlouhé texty, binární soubory apod.). |
| Broadcast | Souhrn informací vysílaných z jednoho zdroje většímu množství adresátů obyčejně prostřednictvím počítačové sítě. [3] |
| Cache | (vyrovnávací paměť) Rychlá paměť mezi hlavní pamětí a procesorem počítače. Hlavní paměť může být základní systémová paměť RAM nebo také vnější paměťové médium (např. pevný disk). Do paměti cache se průběžně ukládají data, čtená z hlavní paměti. Při požadavku na přečtení dalších dat se nejdříve prohledá rychlejší paměť cache. Pokud cache data obsahuje, načtou se mnohem rychleji než z hlavní paměti. Nejsou-li požadovaná data ve vyrovnávací paměti, musí se načíst standardním postupem z pomalejší hlavní paměti nebo média. [3] |
| Commit | Potvrzení transakce. Všechny dosud „pouze virtuálně“ provedené změny v databázi se do ní najednou již nastálo uloží. |
| Databázový soubor | Fyzicky reprezentuje danou databázi. Obsahuje veškerá data obsažená v databázových tabulkách a dalších objektech. |
| Databázový systém | Systém (program), který je spuštěn na serveru a zprostředkovává přístup k datům uloženým v databázovém souboru. Po autentizaci uživatele (uživatelské aplikace) od něho převezme dotaz (SQL) a zpátky vrátí v něm požadovaná data, má-li k nim uživatel povolený přístup. |
| FireWall | Hardware či software (ev. obojí), které slouží k oddělení jedné sítě od druhé z důvodů bezpečnosti. Nejčastěji se dnes používá k oddělení Internetu od lokální počítačové sítě (tj. aby nikdo bez příslušných práv přístupu nemohl získat přístup k počítačům v LAN). Hardwarový firewall se též někdy označuje jako firewall machine, programový firewall pak jako firewall code. [3] |

| Výraz | Vysvětlení |
|---|---|
| IP | Síťový unixový protokol, vycházející z modelu OSI (vrstva 3). Původně vytvořen pro komunikaci mezi sítěmi (intenetworking) se stal protokolem Internetu zejména díky tomu, že směrovače jsou schopny pružně reagovat na zahlcení sítě zpomalováním a při uvolnění pak zrychlováním vysílání paketů. Tím je docíleno nezbytné pružnosti a nedochází k totálnímu ucpání informační dálnice či dokonce ke ztrátám dat. [3] |
| IP Addres | Adresa počítače v síti používající protokol (viz) IP. Sestává ze čtyř osmibitových čísel oddělených tečkami, tj. např. může být 191.254.12.255. IP adresa identifikuje počítač jedinečně v celosvětové síti Internet. IP adresy se dělí na třídy A až E (Class A až Class E). [3] |
| Kryptografie | Obor zabývající se kódováním dat (zakódováním i rozkódováním). |
| MAC | Media Access Control. Obecný termín pro způsob, kterým stanice získává přístup k přenosovému médiu. [3] |
| Metadata | SQL skripty, kterými je definována databáze. Jejich spuštěním v databázovém systému by se tato databáze měla od počátku vytvořit až do své konečné fáze (všechny tabulky, viewy, uložené procedury, integritní omezení…). Součástí metadat však nejsou samotná data. |
| MySQL | Volně šiřitelný databázový systém. |
| PGP | Vysoce kvalitní šifrovací aplikace používající veřejný klíč systému (viz) RSA; lze jej provozovat na systémech PC, Unix a dalších. Vytvořil jej Philip R. Zimmermann a je šířen zcela volně; systém využívá algoritmu patentovaného jinou americkou společností. PGP umožňuje vyměňovat data a zprávy s vysokým stupněm bezpečnosti - zprávy jsou zašifrovány tzv. veřejnými klíči, které nemusí být mezi uživateli (příjemce i vysílající používají stejný klíč, kterým zprávu zakódují a rozkódují) posílány zabezpečenými kanály. [3] |
| RAM | (paměť s přímým přístupem) Druh paměti používané v počítačích, umožňující zápis i čtení. K paměti RAM existuje přímý přístup. Paměť RAM je energeticky závislá, tj. její obsah musí být neustále obnovován přívodem energie a při jejím výpadku se ztrácí. Paměť typu RAM je základní pracovní pamětí, ve které jsou uloženy aktuálně spuštěné programy a zpracovávaná data. [3] |

| Výraz | Vysvětlení |
|---|---|
| Rollback | Zrušení transakce. Všechny operace v databázi v rámci takto zakončené transakce se zruší a databáze se bude nacházet ve stejném stavu, jako před zahájením transakce. |
| RSA | Šifrování s využitím veřejného klíče umožňující jak šifrování, tak autentizaci. Vynalezeno v roce 1977 pány Rivestem, Shamirem a Adlemanem, odtud jméno. Veřejný klíč vychází ze dvou velkých prvočísel, které vyústí ve dvě čísla, určující veřejný a privátní klíč. Prvočísla jsou tajná, jejich odhalení (a tudíž odhalení i privátního klíče a celé šifry) z veřejného klíče vyžaduje enormní výpočetní úsilí. [3] |
| SQL | Databázový dotazovací jazyk, který se dnes stává standardem, ke kterému se hlásí veškeré vedoucí databázové systémy současnosti. Podstatou SQL je používání interaktivních dotazů při práci s databází; jazyk rovněž obsahuje příkazy pro další obvyklé činnosti s databází. Nejpoužívanější dotazovací jazyk u aplikací client-server. [3] |
| SŘBD | (systém řízení báze dat) Obecně větší databázový systém zaměřený na systematické zpracování a údržbu rozsáhlého množství informací databázového charakteru. [3] |
| Transakce | Transakce je programová jednotka. Jedná se o posloupnost akcí (čtení, zápis, výpočet s daty uloženými v databázi), se kterou se zachází jako s jedním celkem. Každá transakce se může skládat z několika UPDATE operací. [8] |
| Trigger | Procedura v SQL jazyku, která je součástí databáze a spouští se před nebo po uskutečnění určité operace (INSERT, UPDATE nebo DELETE) s každým záznamem dané tabulky. |
| Uložená procedura | Procedura v SQL jazyku, která je součástí databáze. Může vracet data formátovaná do tabulky, nebo na základě vstupních parametrů vykonávat v databázi nějaké operace. Uložená procedura se spouští samostatným příkazem (buď v rámci SELECT dotazu nebo se na ni odvolávají jiné procedury či triggery). |

| Výraz | Vysvětlení |
|---|---|
| View | Pohled na data databáze. V podstatě se „tváří“ jako databázová tabulka, ovšem View sám o sobě žádná data neobsahuje, je pouze SQL dotazem, který jej definuje, a data přebírá z jiných databázových tabulek. Při definic triggerů pro view se tento stává editovatelným. O aktualizaci dat se však musí postarat ony triggery. |
| VPN | Šifrovaný tunel mezi dvěma sítěmi skrz veřejnou informační síť. |
| Wi-Fi | Wireless Fidelity – bezdrátová technologie v bezlicenčním pásmu 2,4 GHz. Wi-Fi je prakticky jen komerční označení (vytvořené organizací WECA - Wireless Ethernet Compatibility Alliance) síťového protokolu 802.11b. [3] |
| Záznam | Data tabulky, která spolu souvisí a mají mezi sebou vztah, přehledně zformátovaná do jednoho řádku databázové tabulky. |

## 6.2. Přehledy

### 6.2.1. Použitá literatura

#### 6.2.1.1. Internet


[1] http://www.askon.cz/ikey/teorie.html

[2] http://notebook.cz/ /,clanky,ostatni,2004,notebook-a-zlodeji,index.html

[3] http://www.zive.cz/slovnik/Default.asp

[4] http://www.zive.cz/h/Uzivatel/AR.asp?ARI=119812

[5] http://www.zive.cz/h/Uzivatel/AR.asp?ARI=106772

[6] http://firewalls.securenet.cz

[7] http://www.mujmac.cz/art/sw/bezpecnost3.html

[8] http://lide.uhk.cz/home/fim/ucitel/fupoulp1/www/DBS2/pr6t.htm


### 6.2.2. Související zdroje

#### 6.2.2.1. Literatura


- Josef Zelenka, Jan Čapek, Jiří Francek, Hana Janáková; *Ochrana dat – Kryptologie;* Vydání první, Hradec Králové: Gaudeamus, 2003; Počet stran: 198; ISBN: 80-7041-737-4


#### 6.2.2.2. Internet


- http://www.krypta.cz
- http://sweb.cz/enforcer/admin05.htm
- http://www.systemonline.cz/site/bezpecnost/sustr2.htm
- http://interval.cz/clanek.asp?article=1282
- http://www.earchiv.cz/b01/b0100024.php3
- http://www.viry.cz
- http://gama.fsv.cvut.cz/~soukup/dis/kap1.html
- http://www.techbox.cz/clanek.asp?id=2476