Zkouškové otázky z BIS

Info
Každý si prosím vyberte jedno téma a to poté zpracujte. Zkuste si informace nějak ověřovat, vykopírované informace z borce ze starých Palovského přednášek bych sem vážně nedával, podle zběžného nahlédnutí to opravdu nejsou moc dobrý materiály. A u obsahu si napište do závorky jméno, kdo tu otázku zpracovává… ať pak v tom nemáme bordel a neděláme něco dvakrát. Díky :-)

Obsah

1. Bezpečnostní analýza a analýza rizik.

vic stránka http://www.cleverandsmart.cz/analyza-rizik-jemny-uvod-do-analyzy-rizik/

!!dle Pavlíčka nejtěžší otázka!!

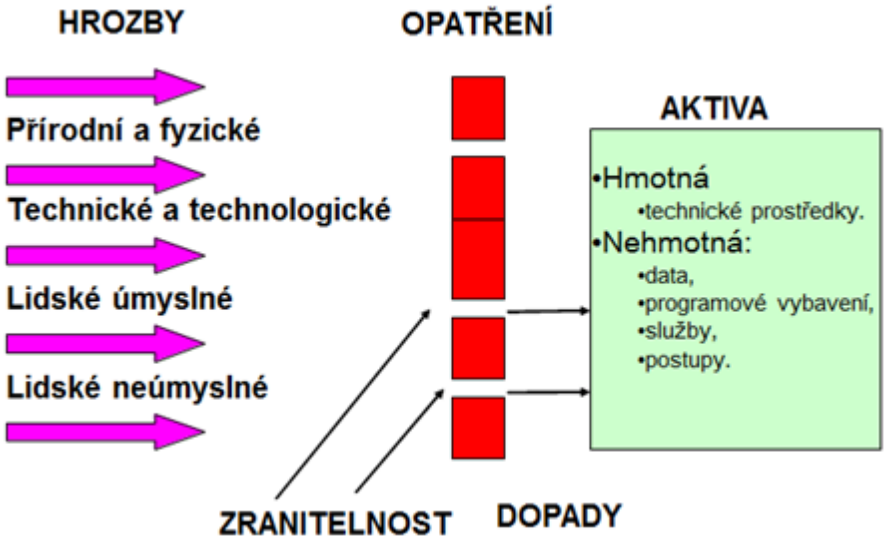
Vymezení bezpečnosti

Aktiva

- **cokoli co má** pro organizaci **nějakou hodnotu** (problém je s učením hodnoty při poškození)
- hmotná
    - technické prostředky
- nehmotná
    - data
    - programy
    - služby,…

Hrozby

- přírodní a fyzické
- technické a technologické (selhání zařízení)
- lidské úmyslné a neúmyslné



Vymezení bezpečnosti

- důvěrnost
  - **k datům má přístup pouze oprávněný uživatel**
- dostupnost
  - **ten, kdo má právo k přístupu, se k nim dostane**
- integrita
  - **vlastnost dat, že nebudou změněna nebo upravena neautorizovaným uživatelem**
- odpovědnost a prokazatelnost
  - **akce budou spojené s určitou osobou, která dané operace provedla**

Analýza rizik

- Je součástí plánování
- Důležité klasifikovat aktiva a spojit je do skupin (PC v kanclu + kdo za to odpovídá)
- Vymezím oblast co chci chránit
- Určím aktiva co tam patří
  - Určím odpovědnou osobu
  - Ocením aktiva
- Určím hrozby pro aktiva
  - Vycházím z katalogů
  - Omezení (zákony, smlouvy,…)

Hodnocení aktiv

- Znehodnocení majetku
- Ztráty dat v daných aktivech (pc+data v něm)

Hrozby

- Katalogy hrozeb (ISO 27005)
    - Pro různé lokality
    - Povodně, letadla, doprava
        - Pro různá patra, sklep,…
        - Rozvody vody, okna,…

Riziko

- Vyhnutí se riziku
    - Změna lokality
- Akceptace
    - Prostě to tak je a nic s tím neudělám a as i nemůžu
- Přenesení
    - Např. pojištění, outsourcing
- Řešeníà navrhovaná opatření

Projekty

- Sdružování rizik do skupin
- Přenést navrhnuté projekty vedeníà vedení bere na vědomí, že tu jsou daná rizika a jak jsou ošetřena a co se má udělat
- Vedení musí schválit dokument a dá souhlas pro aplikaci (jde-nejde)

Prohlášení o aplikovatelnosti

- **Seznam opatření, které budou aplikovány**
- Pokud se nekryje s navrhovanými řešení, zmiňuje se tato skutečnost do prohlášení (nedostatek peněz, nedostatek lidí)

Aktualizace hrozeb

- Nové vedení; mobilní telefony; jíné nové technologie, které musím nově zapracovat do analýzi (mobilní bankovnictví atd.
- Aktualizovat 1 x za 12 měsíců

2. Cíle kryptografie, kryptografické elementy, kryptografické protokoly.

Kryptologie = Kryptografie + kryptoanalýza

Kryptografie

– věda o tajných kódech zaručující důvěrnost komunikace přes nezabezpečený komunikační kanál
– jak zprávu zašifrovat tak, aby ji nepovolaná osoba nerozuměla

Kryptoanalýza

– věda o analýze slabin šifrovacích systémů
– jak dešifrovat zašifrovanou zprávu bez znalosti klíče

Historie je neustálý boj kryptografie a kryptoanalýzy, v současnosti vyhrává kryptografie

Cíle kryptografie:

<u>důvěrnost (confidentiality)</u> - též bezpečnost - jedná se o udržení obsahu zprávy v tajnosti.

celistvost dat (data integrity) - též integrita - jedná se o zamezení neoprávněné modifikace dat. Tato modifikace může být smazání části dat, vložení nových dat, nebo substituce části stávajících dat jinými daty. Se zamezením neoprávněné modifikace souvisí i schopnost tuto modifikaci detekovat.
autentizace (authentication) - též identifikace, neboli ztotožnění znamená prokazování totožnosti, tj. ověření, že ten, s kým komunikujeme, je skutečně ten, se kterým si myslíme, že komunikujeme. Autentizace může probíhat na základě znalosti (heslo), vlastnictví (klíče od bytu, kreditní karta) nebo charakteristických vlastností (biometrické informace - např. otisky prstů).
nepopiratelnost (non-repudiation) - souvisí s autorizací - jedná se o jistotu, že autor dat nemůže své autorství popřít (např. bankovní transakci).

Kryptografické elementy

1. Šifrování
2. Hashování
3. Digitální Podpisy

Kryptografické protokoly
Kryptografický protokol (bezpečnostní protokol nebo také šifrovací protokol) je protokol, který zajišťuje bezpečný přenos informací na základě předem daných ustanovení. Protokol popisuje, jak by měl být daný algoritmus použit.
Charakteristiky:
– Každý kdo používá protokol, musí znát protokol a všechny jeho kroky v předstihu.
– Každý kdo používá protokol, musí souhlasit s posloupnosti kroků
– Protokol musí být bez možnosti chyb.
– Protokol musí být úplný.
Typy protokolu: Na tohle se Pavlíček vůbec neptal, když jsem mu řekl, že to tam mam, tak se jen pousmál a zeptal se jestli vím konkrétní příklady těchto protokolů
Potvrzované protokoly – protokoly kde komunikaci potvrzuje nezávislá důvěryhodná osoba.
Posuzované protokoly – protokoly, u nich je možné v případě sporu stran najit viníka.
Samoopravné protokoly – protokoly kde komunikují pouze zúčastněné strany a součástí protokolu je identifikace viníka v případě porušení protokolu.

Příklad protokolu: Podepisování mailu SMIME (Chtěl konkrétně popsat krok za krokem co se dělá při podepisování mailu). Ale dal mi na výběr popsat libovolný protokol (podpis souboru, …)
Open PgP

3. Šifrování (symetrické a asymetrické šifry, délka klíče), módy, blokové a proudové šifry

Šifrování (Kryptografie) - nezabezpečená elektronická data se převádí za pomoci kryptografie na data šifrovaná, čitelná pouze pro majitele dešifrovacího klíče

Rozdělení šifer
- symetrické
  - blokové
  - proudové
- asymetrické

Symetrická šifra – používá k šifrování i dešifrování stejný klíč. Výhodou je jejich nízká výpočetní náročnost. Nevýhodou je nutnost sdílení tajného klíče, takže se odesílatel a příjemce tajné zprávy musí předem domluvit na tajném klíči. Často se používají v kombinaci s asymetrickými šiframi, kdy se otevřený text zašifruje symetrickou šifrou s náhodně vygenerovaným klíčem. Tento symetrický klíč se zašifruje veřejným klíčem asymetrické šifry, takže dešifrovat data může pouze majitel tajného klíče dané asymetrické šifry.

Asymetrická šifra (kryptografie s veřejným klíčem) – K šifrování a dešifrování se používají odlišné klíče – veřejný a soukromý klíč. Klíč určený k šifrování je veřejný, majitel klíče ho volně uveřejní, a kdokoli jím může šifrovat jemu určené zprávy. Dešifrovací klíč je privátní (soukromý), majitel jej drží v tajnosti a pomocí něj může tyto zprávy dešifrovat.
Asymetrická kryptografie je založena na tzv. jednocestných funkcích, což jsou operace, které lze snadno provést pouze v jednom směru: ze vstupu lze snadno spočítat výstup, z výstupu však je velmi obtížné nalézt vstup. Například násobení: je velmi snadné vynásobit dvě i velmi velká čísla, avšak rozklad součinu na činitele (tzv.faktorizace) je velmi obtížný.
Asymetrické šifrování je podstatně náročnější než symetrické. Používá se také pro elektronický podpis.

- konkrétní příklad – RSA, Diffie-Hellman, Digital Signature Algorithm...

Délka klíče – označuje v kryptografii délku šifrovacího klíče příslušného šifrovacího algoritmu, která je uváděna v bitech. Délka klíče algoritmu je odlišná od jeho šifrovací bezpečnosti, která je logaritmickou mírou nejrychlejšího známého výpočetního útoku algoritmu. Bezpečnost algoritmu nemůže překročit jeho délku klíče (protože každý algoritmus může být prolomen hrubou silou), ale může být menší.

Módy – Operační módy blokových šifer jsou způsoby použití blokových šifer. Pomocí nich je možné získat nové vlastnosti a využití blokových šifer. Jedná se o ECB, CFB, OFB, CBC, CTR a XTS.

ECB (Electronic Codebook Mode)- základní šifrování každého bloku odpovídajícím algoritmem. Identické bloky budou vypadat stejně i po zašifrování, stejné bloky otevřeného textu mají vždy stejný šifrový obraz. Ve zprávě šifrované módem ECB může útočník bloky šifrového textu libovolně vkládat, zaměňovat nebo odstraňovat. Mód ECB nezajišťuje integritu otevřeného textu.
CFB (Cipher feedback) **-** Bloková šifra se chová jako proudová šifra s vlastní synchronizací**.** Pro CBC šifrování začíná, když jsou v zásobě data pro celý blok. V CFB jsou data šifrována v jednotkách menších než je velikost bloku. Šifrování: Blok je naplněn náhodným číslem, je nabrán první byte prostého textu do registru B. Blok je zašifrován a jeho nejlevější byte je použit pro XOR s bytem v registru B, získáme zašifrovaný byte S. Byte S je odeslán do komunikační linky, celý šifrovací blok se posune o jeden byte vlevo a na místo nejpravějšího bytu v bloku se umístí S. Dešifrování: Reverzní proces.
CBC (Cipher block chaining) - zřetězení šifrovaných bloků, každý znak otevřeného textu ovlivňuje pouze jeden znak šifrového textu, každý další blok je zašifrován pomocí předcházejícího bloku. Identické nezašifrované bloky nejsou stejné po zašifrování, nejprve se první blok promění náhodným inicializačním vektorem, který se pošle přijímací straně otevřeně před šifrovým textem, až poté se zašifruje.

Bloková šifra – Typ symetrické šifry, která pracuje s bloky pevně stanovené délky, nejčastěji 64 bitů (AES 128 bitů). Delší zprávu rozdělí do více bloků, ten poslední doplní o tzv. výplň.

- Šifrování – každý blok je zašifrován pomocí šifrovacího algoritmu řízeného utajeným šifrovacím klíčem.

- Dešifrování – Zašifrované bloky stejné délky jsou postupně rozšifrovány stejným šifrovacím algoritmem pomocí stejného šifrovacího klíče.
  Slabým místem je opakované použití stejného klíče na všechny bloky. Řešením je použití inicializačního vektoru – tj. blok náhodných dat, který zajistí, aby výsledkem šifrování dvou identických zpráv byly dvě různé šifrované zprávy.
- konkrétní příklad – DES, AES, IDEA, Triple DES, RC2, RC5...
  Proudová šifra – Tato šifra zpracovává text po jednotlivých bitech. Vstupní datový tok je kombinován (typicky pomocí funkce XOR) s pseudonáhodným proudem bitů vytvořeným z šifrovacího klíče a šifrovacího algoritmu. Výsledkem je zašifrovaný datový tok (proud), který je šifrován neustále se měnící transformací (na rozdíl od blokové šifry, kde je transformace konstantní). Proudové šifry jsou typicky rychlejší než blokové šifry a pro implementaci potřebují jednodušší hardware. Naopak jsou na rozdíl od blokových šifer náchylnější ke kryptoanalytickým útokům, pokud jsou nevhodně implementovány (počáteční stav nesmí být použit dvakrát).
- konkrétní příklad – RC4, FISH

4. Hashové funkce, digitální podpisy.

Hashové funkce

Hašové funkce jsou takové funkce, které jednosměrně převedou zprávu na bitový otisk. Výsledný otisk nelze zpětně převést na její výchozí hodnotu. Dalším specifikem hashovacích funkcí je jednotná délka výstupu (Která se liší dle

jednotlivých hashovacích funkcí). V neposlední řadě 2 takřka identické vstupy lišící se např. jen v jednom znaku vyústí v zcela jiný výsledek.
Příklad výstupů hashovací FCE SHA1 (http://www.sha1.cz/):

bis01	5db4fc2eb3e7a707854b20ec9a2a9e211e8484fc
bis02	f56fa23bc5274f27bb5bd6a580f83ec44bf89a6f
dlouhyTextMaStejnouVyslednouDelku	1cfa97278fc518f2c1e26379f46fad0060d969e4

Proč je použití hashovacích funkcí bezpečné?
1) Jednosměrnost hashování - z "bis01" snadno získám "5db4fc2eb3e7a707854b20ec9a2a9e211e8484fc" ale z "5db4fc2eb3e7a707854b20ec9a2a9e211e8484fc" nelze jednoduše získat "bis01"
2) Je pochopitelné, že vzhledem k omezené délce výstupu otisku musejí existovat takové 2 vstupy, které budou mít stejný výstup. K nalezení dalšího vstupu pro stejný výstup však neexistuje snadný systematický postup. (Který by zvládl nalézt předlohu v realném čase).
3) Obtížnost nalezení kolize (Kolize je stav kdy pro 2 různé vstupy nalezneme 1 výstup)

Prolomení hashovacích funkcí
Kolize - Hashovací fce MDA5 byla roku 2004 prolomena, neboť byl nalezen efektivní způsob jak získat předlohu pro otisk v řádu sekund. Prolomentí hashovacích funkcí je obvykle pouze záležitostí času. (S rostoucí výpočetní kapacitou je nalezení kolize, zvláště pro hashovací funkce malé délky výstupu, realnější a realnější). V případě, že dojde k prolomení konkrétní hashovací funkce, se doporučuje danou hashovací funkci nahradit jinou, silnější. Teoreticky již byla prolomena i funkce SHA1 a doporučuje se používat SHA2.
Bruteforce - Princip spočívá v generování výstupů pro inkrementálním vstupy (popřípadě slovníkové vstupy). Za pomoci GPU není nerealné louskat hesla i rycholostí 100M/sec. Útok bruteforce je úspěšný, jakmile nalezený otisk souhlasí s hledaným otiskem (Může dojít ke kolizi a útočník může nalézt jiného heslo než to vaše, ale pro přihlášení do systému, který ukládá hashovaná hesla mu daná kolize bude stačit pro získání přístupu.
Rainbow tables - jsou velká databáze již vygenerovaných hashů (které byly získány generováním metodou bruteforce). Generování hashe je nároční na výpočet (CPU/GPU) řádově více, než prohledávání dat na disku. Chce-li si účtočník usnadnit práci, může investovat do velkokapacitních HDD a nasledně si může stáhnout velké databáze existujících hashů.
Platí, že lze obětovat CPU/GPU za místo na HDD a naopak.

Narozeninový paradox - poukazuje na pravděpodobnost, že ve skupině 23 lidí (a více) je pravděpodobnost, že 2 lidé budou mít narozeniny ve stejný den větší než 50%. Obdobně snižuje pravděpodobnost nalezení kolize na 2^(délka hashe/2) z původního 2^(délka hashe)

Použití:
Uložení hesel - Ukládání hesel je jedním z nejčastějších použití hashovacích funkcí. Namísto uložení uživatelského hesla je uložen pouze jeho otisk. Při autentizaci uživatele se vypočítá otisk ze zadaného hesla a porovnávají se pouze otisky. Tímto způsobem je zaručeno, že pokud se podaří útočníkovi získat databázi uživatelů, nemůže jednoduše získat jejich hesla v textové podobě. K zvýšení bezpečnosti uložených hesel se používá **sůl** - solí je myšleno přidání dalších znaků k uživatelově heslu. Pokud by uživatel používal snadné heslo "heslo", jeho prolomení na základě slovníkového útoku, resp. použití rainbow tables, by bylo otázkou pár vteřin. Přidáme-li ale náhodný textový řetězec k heslu, např. "heslof56fa23bc5274f27bb5bd6a580f83ec44bf89a6f" odhalení za pomocí rainbow tables se snižuje. Sůl by měla být unikátní pro každý hash a měla by být uložena spolu s otiskem hesla (otiskem hesla se solí). Předpokládá se, že útočník sůl ví, ale ztíží se mu získávání textových hesel.
Integrita souboru - otisky bitových souborů dokáží ověřit "shrnout" informaci o bitovém souboru do několika mála znaků. Při změně jakéhokoliv bitu v několika GB souboru bude výsledný otisk zcela jiný.

Digitální podpis

Digitální podpis se používá pro ověření pravosti digitálního obsahu (např. PDF, Word či jiná textová/binární data). Validní podpis dává příjemci jistotu, že přijímaný soubor nebyl od jeho odeslání změněn a že byl podepsán pouze osobou, která vlastní privátní klíč k certifikátu.

Proces vzniku podpisu:

Vezme se konkrétní soubor (jeden). Ze souboru se vypočítá hash pomocí hashovací funkce. Tento hash (např. převedeno do hexadecimální podoby: da39a3ee5e6b4b0d3255bfef95601890afd80709)
je dále zašifrován pomocí privátního klíče podepisujícího. Výsledek šifrování spolu s certifikátem tvoří tzv. "podpis". Podpis může být připojen přímo k souboru (formát a cílová strana to musí podporovat) nebo může vzniknout jako samostatný soubor, který je předáván spolu s originálním souborem.

Ověření podpisu:

Aby proces podepisování měl smysl, musí být pouze jedna strana schopná podpis vytvořit a kdokoliv jiný podpis ověřit.

Samotné ověření podpisu se skládá ze dvou částí:

1) Ze souboru se vypočítá hash pomocí hashovací funkce (Původní soubor resp. soubor s odstraněným podpisem!)
2) Podpis souboru se dešifruje za pomoci veřejného klíče.

Následně dojde k porovnání výše zmíněných 2 částí. Rovnají-li se je podpis platný. V opačném případě byl soubor pravděpodobně neoprávněně změněn.

Bezpečnost:

Je důležité podotknout, že podpis nešifruje předávaný obsah. Ten je i nadále čitelný případnému útočníkovi. Dále také není žádný problém podpis odstranit (buďto smazáním samostatného souboru, nebo odstraněním podpisu přímo ze souboru). Útočník tedy může soubor vzít, podpis odstranit a upravit předávaný obsah. Pro příjemce souboru by takto upravený soubor vypadal jakože nikdy nebyl podepsán a je na jeho úsudku, zda-li bude souboru důvěřovat. (Pochopitelně pokud jsou obě strany domluveny, že všechny zprávy/soubory budou podepisovány. Je to podezřelé).

Použití podpisů:

1) Autentizace

Validně podepsaný e-mail se správným certifikátem může nahradit např. ověřování proti občanskému průkazu. Např. pokud si zařídíte kvalifikovaný certifikát na pobočce pošty, můžete s obecními úřady komunikovat pomocí e-mailu a tato konverzace je pro úřad závazná. Bohužel tento způsob komunikace stále není moc rozšířený.

2) Zajištění integrity předávaných dat

Platný podpis se vztahuje ke konkrétnímu obsahu souboru. Pokud bych například posílal příkaz k převodu do banky na částku 10 000 a validně bych jej podepsal, pak bych se rozhodl, že chci raději převést 20 000, soubor bych upravil ale neaktualizoval bych podpis, pro banku tento příkaz nebude platný, neboť podpis nebude validní. Tento scénář brání útočníkovi, aby např. změnil číslo účtu (na své).

3) Neodvolatelnost

Jakmile je soubor podepsán (a vlastní jej ještě někdo jiný) nelze tento podpis odvolat. Nelze tedy prohlásit, že jste s tímto souborem (a tímto konkrétním obsahem) nesouhlasili. Často je k podpisu přidáváno i časové razítko, které stvrzuje čas podpisu. Není tedy možné např. po pojistné události vytvořit žádost o sjednání havarijního pojištění, podepsat jej a zaslat pojišťovně s tím, že jste žádost vyplnili již před rokem

5. Certifikáty, certifikační autorita.

Ve světě kryptografie certifikát slouží ke spojení identity s veřejným klíčem (public key). Obsahuje informace o vlastníkovi. Obvykle jméno, e-mail, adresu, atd… Certifikáty jsou vydávány třetí stranou - certifikační autoritou (CA). Certifikační autorita je obvykle firma, která má na trhu důvěryhodné postavení a pro funkčnost systému musí panovat důvěra k certifikátům, které daná CA vystavila. Příklady certifikačních autorit:

Položky certifikátu X.509:

- **Pořadové číslo cerifikátu** (SN) - Unikátní číslo certifikátu
- **Algoritmus podpisu** - pro výpočet hashe (např. SHA1), asymetrickýalg pro zašifrování hashe (např. RSA)
- **Platnost** (NotBefore-NotAfter) - Po vypršení platnosti se nesmí klíče certifikátu použít k podepisování, šifrování či autentizaci. **Ale nesmí se vyhodit!** Je potřeba pro dešifrování dříve přijatých zpráv
- **Vydavatel** (Issuer)
- **Předmět** (subject) - Specifikace držitele ve formátu jedinečného jména. CA by měla pro jeden předmět vydat pouze jeden certifikát (výjímkou je prodloužení či revokované certifikáty)
- **Jedinečné jméno** (Distringuished Name) - Složené z mnoha částí (C=US, ST=UT, L=Salt Lake City, O=The USERTRUST Network, OU=http://www.usertrust.com, CN=UTN…)
- **Veřejný klíč** (Subject public key) - vlastní veřejný klíč, algoritmus, dodatečné info o algoritmu
- **Rozšíření certifikátu** (x509v3 extensions) - Další informace, programy nemusí rozumnět všem informacím, ale pokud položka obsahuje příznak **critical**, musí certifikát odmítnout.
- **Digitální podpis** - hash vlastností certifikátu podepsaný vydavající CA

Certifikát může mít různé funkce. :

- **Digitální podpis** (Digital Signature) - certifikát je určen k elektronickému podpisu dat, např. pro autentizaci uživatelů či k ověření integrity dat. Nelze samostatně použít k ověření pravosti (nepopiratelné odpovědnosti).
- **Neodvolatelnost** (Non Repudation) - certifikát je určen k ověření pravosti, tj. je nutný k ověření digitálního podpisu dokumentu. Existuje diskuse, zda pro to musí být nastaven i bit Digital Signature (pokud nelze podepsat, tak nelze ani ověřit pravost podpisu - viz série článků Kvalifikovaný certifikát na dvě věci,Kvalifikovaný certifikát na dvě věci II a Kvalifikovaný certifikát na dvě věci III). V poslední verzi standardu X509 se tato položka jmenuje contentCommitment(odpovědnost za obsah).
- **Zakódování klíče** (Key Encipherment) - certifikát je určen k šifrování soukromých či tajných klíčů (tajný klíč se používá v symetrických šifrách). Klasickým příkladem je poslání zašifrované zprávy elektronickou poštou. Aplikace nejdříve vygeneruje náhodný tajný klíč pro symetrickou šifru, následně pomocí tohoto tajného klíče zašifruje vlastní data. Následně zašifruje tajný klíč pomocí veřejného klíče příjemce (získá z jeho certifikátu) a výsledek přidá ke zprávě.
- **Zakódování dat** (Data Encipherment) - veřejný klíč lze použít pro šifrování dat (jiných než klíče). Jeho užití je velmi neobvyklé.
- **Key Agreement** - certifikát pro výměnu klíčů. Používá se v souvislosti s algoritmem Diffie-Hellman či při výměně klíčů s využitím eliptických křivek. Nemá smysl u algoritmu RSA.
- **Podepisování certifikátu** (Key Certificate Sign) - veřejný klíč je určen pro verifikaci certifikátů, tj. certifikát certifikační autority.
- **Podepisování CRL** (CRL Sign) - veřejný klíč k privátnímu klíči pro podepisování CRL, pouze u certifikátů certifikační autority.
- **Encipher Only** - ve spojení s Key Agreement, dohodnutý klíč lze použít pouze pro šifrování.
- **Decipher Only** - ve spojení s Key Agreement, dohodnutý klíč lze použít pouze pro dešifrování.

Třídy CA:

1. Třída - CA ručí pouze za jednoznačnost certifikátu. Neprobíhá žádná kontrola pravdivosti zadaných údajů, etc… CA 1. třídy je možné zredukovat na program, který bude automaticky certifikáty vydávat

2. Třída - CA ručí za jednoznačnost certifikátu a za pravdivost v něm uvedených informací. CA tedy musí mít vybudouvanou infrastrukturu pro kontrolu těchto údajů. (Tedy pobočky v různých městech, kde je možné potvrdit svoji identitu). Např. VŠE v Praze má tuto pobočku ve výpočetním centru
3. Třída - CA ručí za jednoznačnost certifikátu a za pravdivost v něm uvedených informací, navíc ale certifikáty vydané touto CA jsou určeny pouze pro konkrétní použití (např. pro interní přístup do aplikace firmy). Výhodou takovýchto speciálních certifikátů je, že nemusejí dodržovat standartizované formy (X.509) ale mohou si (dle kompatibility aplikace) formát upravit.

Ověření platnosti certifikátu
Pro ověření platnosti certifikátu je potřeba provést následující kroky:
1) ověřit, že certifikát není použit mimo vymezenou dobu platnosti
2) CA podepisující certifikát pro mne musí být důvěryhodná
3) CA není na seznamu revokovaných certifikátů (CRL) - některé prohlížeče tento krok přeskakují (viz níže)
4) Ověření integrity certifikátu (ověření hashe)

Důvěryhodnost CA
CA je důvěryhodná tehdy, když jí mám uloženou v seznamu důvěryhodných CA. Seznam důvěryhodných CA může mít každý program jiný (firefox bude mít jiné úložiště než windows, ale windows a internet explorer jej budou pravděpodobně sdílet, …). Není -li CA v seznamu důvěryhodných CA, získá se veřejný klíč certifikační autority, která vystavila certifikát naší CA. Tento proces se opakuje dokud není nalezena CA, které věříme. Pokud dojdeme až na úroveň kořenového certifikátu (Certifikát, který má v položce "Issuer" uveden sám sebe a podpis vydávající CA je shodný s daným certifikátem) a nenalezli jsme důvěryhodnou CA, certifikát je odmítnut. Zároveň je vhodné podotknout, že by se měl vždy strom CA kontrolovat až na kořenovou úroveň, neboť je potřeba ověřit, zda-li certifikát vydávající naší důvěryhodnou CA nebyl revokován - tedy kompromitován. V takovém případě se nedá věřit ani naší "důvěryhodné" CA.

Revokované certifikáty
Pokud pojmeme podezření, že byl certifikát kompromitován (někdo jiný se dostal k privátnímu klíči), nebo jsme sami privátní klíč ztratili, můžeme požádat CA, aby certifikát tzv. "revokovala". Tedy zneplatnila - přidala na seznam zneplatněných certifikátů. Je ovšem na jednotlivých programech, které uživatelé používají, jestli v rámci validace certifikátu zkontrolují zda-li je certifikát revokovaný. Většina prohlížečů má bohužel tuto funkci defaultně vypnutou.

http://zam.opf.slu.cz/botlik/Cd-II/CD-ca/ca5.htm

6. Identifikace, autentizace (co vím, co mám, co jsem), autorizace,

Identifikace - určení/sdělení, kým jsem, za koho se vydávám
lze řešit například zadáním uživatelského jména
Autentizace - prokázání, že jsem skutečně tím, za koho se vydávám
v nejjednodušším případě: pomocí hesla (prokazuji jeho znalost),
lépe pomocí jednorázového hesla (OTP), pomocí autentizačního certifikátu, ……
Autorizace - získání souhlasu / oprávnění k určitému úkonu, kroku, přístupu apod.
někdo souhlas/oprávnění uděluje, podle určitých pravidel, …

Příklad: Někdo zaťuká na dveře s tím, že se jmenuje Martin (Identifikace). Požádám ho, aby se postavil před kukátko, abych mohl ověřit, že skutečně vypadá jako Martin (Autentizace) ("autentický" - pravý). Poté co jsme si jistí, že jde o Martina se rozhodujeme, jestli ho pustíme dál (Autorizace). Jsme-li na něj naštvaní, zůstává stát za dveřmi.
Prostředky:

- Informace, kterou vím
    - Dobrá hesla (Vím jen já)
    - Špatná hesla (Jakékoliv heslo, které by mohl používat kdokoliv jiný, je špatné.)
    - Kvalitní hesla

        - Používají velká i malá písmena
        - Číslice a interpunkční znaky
        - Dlouhá nejméně 6 znaků
        - Rychle se píší ?
    - Hesla - Podpůrné prostředky
        - Kontroly proti slovníkům
        - Kontroly znakové složitosti
        - Povinný interval výměny
        - Kontroly proti opakování
- Předmět, který mám
    - Čipové karty
    - Generátory jednorázových hesel.
    - USB tokeny
- Něco, co jsem
    - Otisky prstů a rukou
    - Parametry oka
    - Rozpoznávání obličeje

7. Logování a zálohování.

Logování

Logování - Vytváření záznamů o událostech. Výsledkem jsou logovací soubory

Logovací soubory můžeme použít k

- Odhalení chyby v programu
- Zdroje napadení
- Rozsahu provedených škod
- Provádět průzkumy

Standardní systémový logovací mechanismus syslog

Logovací soubory

- Logovací adresáře:
    - /usr/adm Dřívější log adresář
    - /var/adm Současný log adresář.
    - /var/log Některé verze Linuxu.
- acct nebo pacct příkazy spouštěné všemi uživateli
- lastlog čas posledního úspěšného přihlášení uživatele loginlog chybná přihlášení wtmp všechna přihlášení trvale
- messages zprávy z konzole a z syslog
- sulog zprávy o použití su příkazu.

Zálohy

Důvodem je bezpečí uložených dat, v případě havárie disku, útoku apod.

Havárie počítače

- Před několika lety bylo denní zálohování počítačů běžnou praxi, protože počítače byly nespolehlivé a hardware počítačů bez zjevných příčin často selhal. V té době byly zálohy jedinou ochranou proti ztrátě dat.
- I dnes přestavuje selhání hardware dobrý důvod pro zálohování. Kromě poruchy hardware jsou možné ještě mnohé jiné důvody pro ztrátu dat. Některé z nich jsou v následujícím seznamu.

Co zálohovat

Jsou dvě základní strategie zálohování:

- Zálohovat všechno co je na vašem systému jedinečné. Tj. uživatelské adresáře, systémové databáze (jako `/etc/passwd`) důležité systémové adresáře (`/bin` nebo `/usr/bin`) které jsou kritické nebo možná zmodifikované.
- Zálohovat úplně všechno. Obnova kompletního systému je rychlejší ze zálohy než z instalace plus obnova ze zálohy.

Typy záloh.

Zálohy dělíme do tří hlavních typů:

- Výchozí záloha
- Úplná záloha
- Inkrementální záloha

Výchozí záloha

- Představuje kopii původního systému. Provádí se po nainstalování systému a provedení případných záplat (patch) před zahájením používání.
- Důležitá při nabourání do systému.

Úplná záloha

- Obsahuje kopii všech souborů na počítači.
- Podobá se výchozí záloze, na rozdíl od ní se provádí pravidelně

Inkrementální záloha

- Zálohují se pouze ty části souborového systému, které se změnily od určité události nebo od určitého data.
- Je možné mít víceúrovňové inkrementální zálohy

Zálohovací strategie

- Popisuje průběh a provádění záloh. Zvláště provádění úplných a inkrementálních záloh. Existuje několik možných zálohovacích strategií.
    - 1. Každého prvního v měsíci se provádí úplná záloha.
    - 2. Každého dne večer se provádí záloha všeho, co se změnilo od poslední úplné zálohy.
- Druhá strategie:
    - Každého prvního v měsíci se provádí úplná záloha.
    - Každou v neděli v noci se provádí záloha všeho, co se změnilo od poslední úplné zálohy (inkrementální záloha 1. úrovně).
    - Každého dne večer se provádí záloha všeho, co se změnilo od poslední inkrementální zálohy 1 úrovně.

Ochrana před selháním média

- Tandemové zálohy jsou zálohy, které paralelně provádějí dva zálohovací cykly.
    - První cyklus dělá úplnou zálohu, každého 1. V měsíci a inkrementální zálohu, každý lichý den.

    - Druhů cyklus dělá úplnou 14. den v měsíci a inkrementální v každý sudý den.
- Tandemové zálohy nám umožní se vyrovnat s poruchou média. Zálohujeme totiž 2 krát.

Vždy vyzkoušet celý proces.

Nejméně jednou ročně je vhodné zkontrolovat celý proces.

- Záloha,
- Uložení,
- Vyjmutí,
- Přečtení a restaurace dat.

Vhodné je také vyzkoušet obnovení zálohy na jiný, cizí, nový počítač.

8. Modely přístupu, mandatory access control (MAC) (Bell-LaPadula, Biba), discretionary access control, ACL

Řízení přístupu

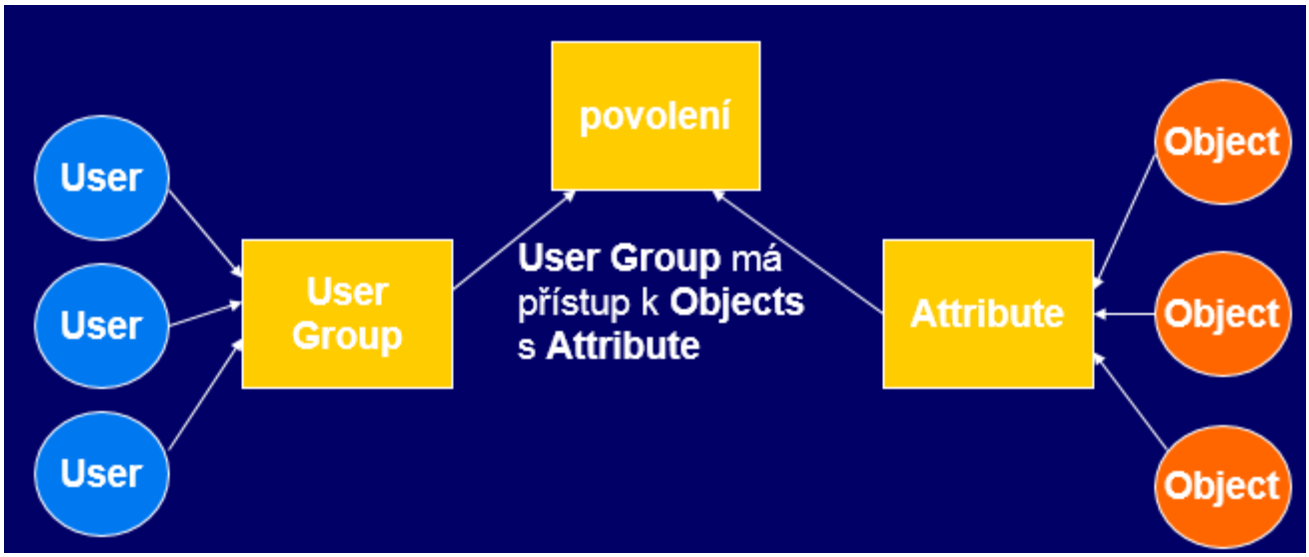
Při práci s IS je potřeba rozlišovat, kdo má k čemu přístup. Jiná práva má anonymní uživatel a jiná administrátor. Druhů uživatelů může být mnoho a je potřeba zajistit, aby se každý z nich dostal jen k datům, jemu určeným.

- Určuje zda **subjekt** může provést požadovanou **operac**i na **objektu**
- Subjekt: uživatel, proces, atd
- Operace: read, write, atd.
- Objekt: soubor, seznam, periferie atd.
- Přístupová matice. 70 léta.

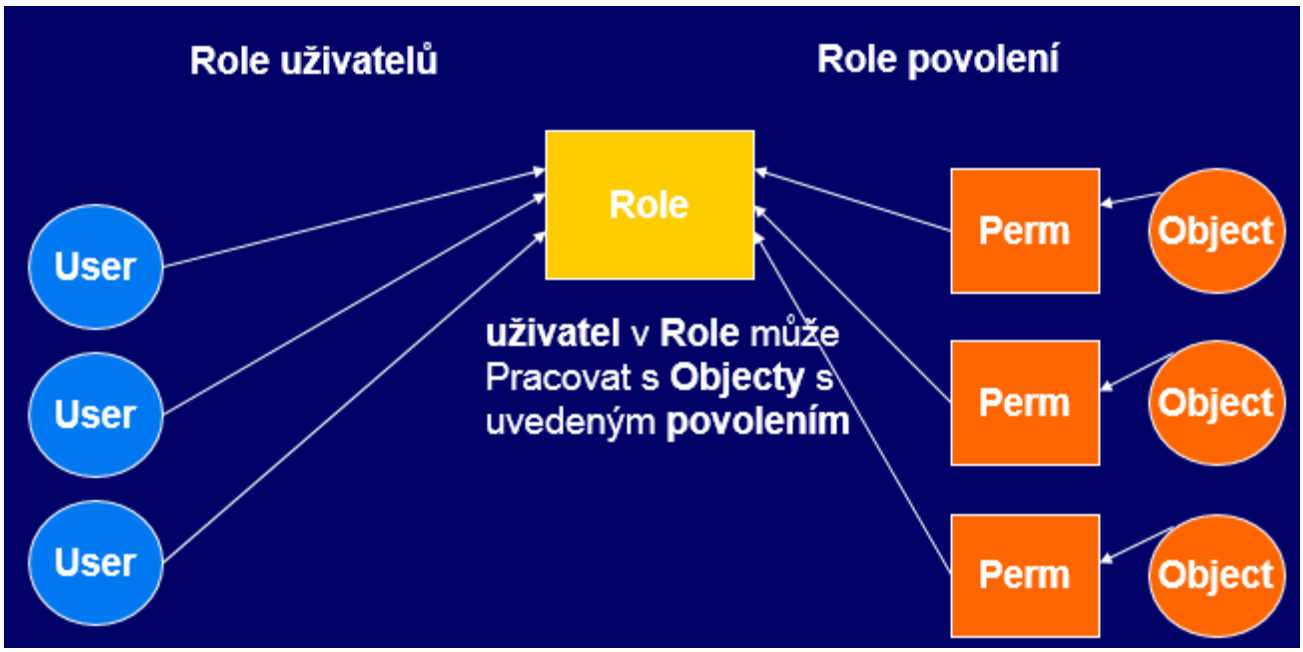
Modely řízení přístupu

- Základní přístupová matice
    - UNIX, ACL
- Agregovaná přístupová matice
    - TE, RBAC, groups and attributes
- Plus Domain Transitions
    - DTE, SELinux, Java
- Formální přístupové modely
    - Bell-LaPadula, Biba, Denning
- Predikátové modely
    - ASL, OASIS, domain-specific models

Skupiny a atributy



Role-based Access Control (RBAC)



DAC a MAC

Discretionary access control (DAC)

- Běžný uživatel může změnit stavy přístupu pokud na to má dostatečná práva
- implementace ve standardních OS's, Unix, Linux, Windows

Mandatory access control (MAC)

- Vynucené celosystémovými pravidly.

- Uživatel schéma nemuže změnit, je předem definováno v systému
- “Silně“ bezpečné systémy vyžadují MAC
- Běžným uživatelům se nedá věřit.

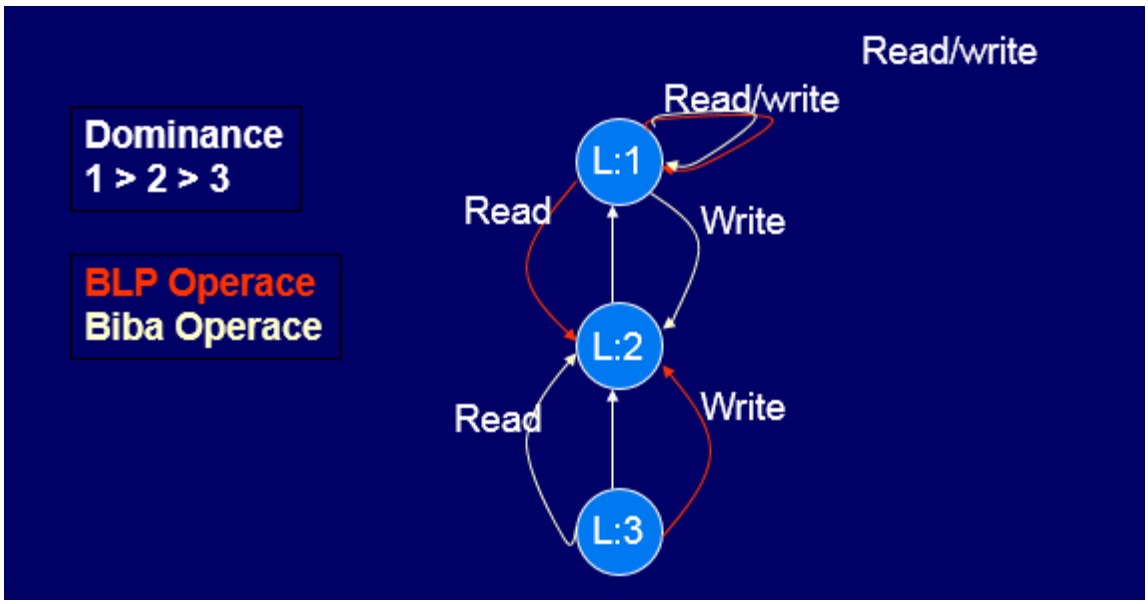
Bell-LaPadula model

- Definuje subjekty, objekty a přístupové operace mezi nimi
- Dvě hlavní bezpečnostní pravidla, jedno pro čtení dat a jedno pro psaní dat.
    - Subjekt nemůže číst data vyššího utajení, než jeho oprávnění
    - Subjekt nemůže zapsat data nižšího oprávnění, než je jeho relace.

Biba model

- Zaměřuje se na integritu ne na důvěrnost
- no read down, no write up
    - Subjekt nemůže číst data o nižší důležitosti
    - Subjekt nemůže psát data pro vyšší úroveň

Ukázka obou dvou modelů



Access control list (ACL)

Seznam oprávnění připojený k nějakému objektu (např. souboru). Seznam určuje, kdo nebo co má povolení přistupovat k objektu a jaké operace s ním může provádět. V typickém ACL specifikuje každý záznam v seznamu uživatele a operaci. Například: Záznam v ACL [*Pepa, smazat*] pro soubor *XYZ* dává uživateli *Pepa* právo *smazat* soubor *XYZ*.
U bezpečnostního modelu používajícího ACL tak systém před provedením každé operace prohledá ACL a nalezne v něm odpovídající záznam, podle kterého se rozhodne, zda operace smí být provedena.

Bezpečnostní modely založené na ACL

Jednou ze základních otázek při tvorbě bezpečnostního modelu založeného na ACL je otázka, jak bude možno editovat samotné ACL. Systémy, které používají ACL, se dají klasifikovat do dvou kategorií: **volitelné** (*discretionary*) a **povinné** (*mandatory*).
Systémy s volitelným řízením přístupu umožňují tvůrci či vlastníkovi objektu plně řídit přístup k objektu, včetně například úpravy ACL tak, aby přístup získal kdokoli jiný. U povinného řízení přístupu (též „nevolitelné řízení přístupu“) jsou všechny operace podmíněny i omezeními stanovenými operačním systémem ještě nad rámec omezení definovaných v ACL.
Tradiční systémy ACL stanovují oprávnění uživatelům jednotlivě, takže systém s velikým počtem uživatelů se poněkud obtížně spravuje. Modernějším přístupem pak jsou bezpečnostní modely založené na tzv. *rolích*, kdy jsou oprávnění přidělována rolím (např. „správce“, „sekretářka“) a uživatelům jsou přidělovány jednotlivé role.

Systém souborů založený na ACL
List je datová struktura, obvykle tabulka obsahující položky specifikující práva uživatele nebo skupiny k určitým objektům, jako jsou programy, procesy, nebo soubory. Tyto položky známé jako položky pro řízení přístupu (ACE) ve Windows, OpenVMS a Mac OS X operačních systémech. Každý dostupný objekt obsahuje identifikator v ACL. Specifická přístupová práva rozhodují o povolení nebo oprávnění, jako který uživatel ma právo čtení, zápisu, nebo provedení objektu. V některých provedeních může ACL řídit zdali může uživatel, nebo skupina uživatelů měnit ACL na nějaký objekt.

9. Centralizovaná autentizace (LDAP, Radius), Kerberos

LDAP

Je štandardizovanýaplikačný protokol navrhnutý na dotazovanie a modifikáciu adresárových služieb, fungujúci cez TCP/IP.

Je to protokol na ukladanie a prístup k dátam na adresárovom serveri. Podľa tohto protokolu sú jednotlivé položky na serveri ukladané pomocou záznamov a usporiadané do stromovej štruktúry. Je vhodný na udržiavanie adresárov a prácu s informáciami o používateľoch, napríklad na vyhľadávanie používateľov v príslušných adresároch. Protokol LDAP je založený na odporúčaní X.500, ktoré bolo vyvinuté vo svete ISO/OSI, ale do praxe sa nie celkom presadilo, hlavne kvôli svojej veľkosti a následnej ťažkopádnosti.

Protokol LDAP vo svojom názve zdôrazňuje, že je odľahčenou verziou odvodenou z X.500.

V praxi to například znamená seznam lidí firmy, jejich přihlašovací jména, domovské adresáře, osobní informace, jména jejich e-mailů nebo čísla telefonů. LDAP může stejně tak dobře uchovávat nastavení uživatelských programů. Data se nemusejí nutně vztahovat na osoby, protokol může pomoct vyhledat různé přístroje ve velké firemní síti (např. fax nebo tiskárnu) a zobrazit jejich umístění či obsahovat různé číselníky (budov, pracovišť atd.).

Výhodou tohoto řešení je možnost velmi efektivně nastavovat přístupová práva (dokonce i na jednotlivé atributy objektů). LDAP servery nabízejí také autentifikaci a dokonce i přes SSL, takže se o svá data nemusíte bát. Systém LDAP umožňuje repliky, takže můžete zátěž rozprostřít na více serverů.

Nejvýznamnější je **autentifikační funkce**. Server bude znát jména i hesla uživatelů a bude autentifikovat veškeré účty v síti (poštovní, ftp…). Výhodou je, že uživatelé budou mít na všechno jedno jméno a heslo. Dají se tak pohodlně „synchronizovat“ uživatelské účty na různých systémech (např. NOVELL nebo NT). Větší firmy mohou LDAP

využít k vyhledávání informací o objektech v síti (osoby, emaily, tiskárny, faxy…). K prohledávání emailů stačí správně nakonfigurovat poštovní program – většina LDAP podporuje (Pegasus, Outlook, Netscape, Mozilla…).

!pýta sa na slabiny protokolu, tj mezdi aplikaciou na ktorej je auth. a LDAP serverom je nešifrované spojenie !!!

Radius

Shrnut: Radius se používal(á) pro vytáčení připojení. Tedy když obvykle platíte za dobu, kdy jste na netu. Radius spolupracuje s NAS. NAS průběžně posílá informace o tom, zda-li Vaše připojení stále trvá a po ukončení pošle informace o datech - kolik jste toho stáhli, uploadli, atd… na základě těchto informací Vám pak přijde účet.

Radius (*Remote Authentication Dial In User Service*, česky *Uživatelská vytáčená služba pro vzdálenou autentizaci*) je AAA protokol (*authentication, authorization and accounting*, česky *autentizace, autorizace a účtování*) používaný pro přístup k síti nebo pro IP mobilitu. Může pracovat jak lokálně tak i v roamingu.

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.

Postup autentizace

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).

Formát RADIUS paketu

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.

Použití RADIUSU

Při připojení k poskytovateli Internetu pomocí vytáčeného připojení, DSL, nebo Wi-Fi je u některých poskytovatelů vyžadováno přihlašovací uživatelské jméno a heslo. Tato informace je poslána do takzvaného Network Access Server (NAS) zařízení přes Point-to-Point Protocol (PPP). Poté je předána RADIUS serveru přes RADIUS protokol. RADIUS server ověří pravost informace použitím autentizačních schémat jako PAP, CHAP nebo EAP. Pokud je uživatelské jméno a heslo přijato, server autorizuje přístup k poskytovateli internetu a vybere IP adresu (popřípadě rozsah adres) a další parametry spojení, což mohou být např. L2TP přihlašovací údaje, doba, po kterou může být uživatel připojen, rychlost připojení, kterou může uživatel používat, nebo jiná omezení. RADIUS protokol neposílá hesla mezi NAS a RADIUS serverem v čistém textu (ani při použití s PAP protokolem), používá se MD5 hašování.

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.

!pýta sa aj na bezpečnostné slabiny a ako komunikacia prebieha teda ako sa vytvorí tunel medzi klientom a radius serverom.

Kerberos

je síťový autentizační protokol umožňující komukoli komunikujícímu v nezabezpečené síti prokázat bezpečně svoji identitu někomu dalšímu. Kerberos zabraňuje odposlechnutí nebo zopakování takovéto komunikace a zaručuje integritu dat. Byl vytvořen primárně pro model klient-server a poskytuje vzájemnou autentizaci – klient i server si ověří identitu své protistrany.

Kerberos je postavený na symetrické kryptografii a potřebuje proto důvěryhodnou třetí stranu. Volitelně může využívat asymetrického šifrování v určitých částech autentizačního procesu. Standardně používá port 88.

Kerberos je založen na Needham-Schroeder Symmetric Key Protocol. Používá důvěryhodné třetí strany nazývané též Key Distribution Center (KDC) sestávající ze dvou logicky oddělených částí: Autentizačního serveru (AS) a Ticket-Granting Serveru (TGS) . Kerberos pracuje na principu tiketů sloužících k ověření identity uživatelů. KDC si udržuje databázi tajných klíčů; každá entita v sítí, ať už klient nebo server, vlastní svůj tajný klíč známý pouze jí a KDC. Znalost tohoto klíče slouží k prokázání identity dané entity. Pro komunikaci mezi entitami KDC vygeneruje „session key“, kterým obě protistrany zabezpečí vzájemnou komunikaci. Bezpečnost tohoto protokolu významně závisí na vzájemné synchronizaci času protistran a krátké životnosti tiketů.

Autentizační proces

Klient se jednorázově autentizuje AS pomocí navzájem známého tajemství (např. heslo) a dostane TGT(Ticket Granting Ticket - tiket opravňující uživatele ke komunikaci s řídícím serverem). Později, když chce klient kontaktovat nějaké SS (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. Detailny postup zde http://cs.wikipedia.org/wiki/Kerberos_(protokol)

Největší riziko spočívá v nutnosti nepřetržitého běhu centrálního serveru. Pokud tento server neběží, nikdo se nemůže přihlásit. Toto riziko lze zmírnit použitím více Kerberos serverů a záložních autentizačních mechanismů.

10. Firewally a proxy

Firewall je síťové zařízení, které slouží k řízení a zabezpečování síťového provozu mezi sítěmi s různou úrovní důvěryhodnosti a zabezpečení. Zjednodušeně se dá říct, že slouží jako kontrolní bod, který definuje pravidla pro komunikaci mezi sítěmi, které od sebe odděluje. Tato pravidla historicky vždy zahrnovala identifikaci zdroje a cíle dat (zdrojovou a cílovou IP adresu) a zdrojový a cílový port, což je však pro dnešní firewally už poměrně nedostatečné – modernější firewally se opírají přinejmenším o informace o stavu spojení, znalost kontrolovaných protokolů a případně prvky IDS. Firewally se během svého vývoje řadily zhruba do následujících kategorií:

- **Paketové filtry** - Nejjednodušší a nejstarší forma firewallování, která spočívá v tom, že pravidla přesně uvádějí, z jaké adresy a portu na jakou adresu a port může být doručen procházející paket. Výhodou je rychlost kontroly, nevýhdou je pak nízká kontrola spojení.

- **Aplikační brány** - Oddělují sítě, mezi které jsou postaveny. Veškerá komunikace přes aplikační bránu probíhá formou dvou spojení – klient (iniciátor spojení) se připojí na aplikační bránu (proxy), ta příchozí spojení zpracuje a na základě požadavku klienta otevře nové spojení k serveru, kde klientem je aplikační brána. Data, která aplikační brána dostane od serveru, pak zase v původním spojení předá klientovi. Výhodou je vysoká míra zabezpečení, nevýhodou pak vysoké nároky na hardware.

- **Stavové paketové filtry** - Stavové paketové filtry provádějí kontrolu stejně jako jednoduché paketové filtry, navíc si však ukládají informace o povolených spojeních, které pak mohou využít při rozhodování, zda procházející pakety patří do již povoleného spojení a mohou být propuštěny, nebo zda musí znovu projít rozhodovacím procesem. To má dvě výhody – jednak se tak urychluje zpracování paketů již povolených spojení, jednak lze v pravidlech pro firewall uvádět jen směr navázání spojení a firewall bude samostatně schopen povolit i odpovědní pakety a u známých protokolů i další spojení, která daný protokol používá. Například pro FTP tedy stačí nastavit pravidlo, ve kterém povolíte klientu připojení na server pomocí FTP a protože se jedná o známý protokol, firewall sám povolí navázání řídícího spojení z klienta na port 21 serveru, odpovědi z portu 21 serveru na klientem použitý zdrojový port a po příkazu, který vyžaduje přenos dat, povolí navázání datového spojení z portu 20 serveru na klienta na port, který si klient se serverem dohodli v rámci řídícího spojení a pochopitelně i odpovědní pakety z klienta zpět na port 20 serveru. Zásadním vylepšením je i možnost vytváření tzv. virtuálního stavu spojení pro bezstavové protokoly, jako např. UDP a ICMP.

K největším výhodám stavových paketových filtrů patří jejich vysoká rychlost, poměrně slušná úroveň zabezpečení a ve srovnání s výše zmíněnými aplikačními branami a jednoduchými paketovými filtry řádově mnohonásobně snazší konfigurace – a díky zjednodušení konfigurace i nižší pravděpodobnost chybného nastavení pravidel obsluhou. Nevýhodou je obecně nižší bezpečnost, než poskytují aplikační brány.

- **Stavové paketové filtry s kontrolou známých protokolů a popř. kombinované s IDS** - Moderní

stavové paketové filtry kromě informací o stavu spojení a schopnosti dynamicky otevírat porty pro různá řídící a datová spojení složitějších známých protokolů implementují něco, co se v marketingové terminologii různých společností nazývá nejčastěji *Deep Inspection* nebo *Application Intelligence*. Znamená to, že firewally jsou schopny kontrolovat procházející spojení až na úroveň korektnosti procházejících dat známých protokolů i aplikací. Mohou tak například zakázat průchod http spojení, v němž objeví indikátory, že se nejedná o požadavek na WWW server, ale tunelování jiného protokolu, což často využívají klienti P2P sítí (ICQ, gnutella, napster, apod.), nebo když data v hlavičce e-mailu nesplňují požadavky RFC apod.

Nejnověji se do firewallů integrují tzv. *in-line IDS* (*Intrusion Detection Systems* – systémy pro detekci útoků). Tyto systémy pracují podobně jako antiviry a pomocí databáze signatur a heuristické analýzy jsou schopny odhalit vzorce útoků i ve zdánlivě nesouvisejících pokusech o spojení, např. skenování adresního rozsahu, rozsahu portů, známé signatury útoků uvnitř povolených spojení apod.

Výhodou těchto systémů je vysoká úroveň bezpečnosti kontroly procházejících protokolů při zachování relativně snadné konfigurace, poměrně vysoká rychlost kontroly ve srovnání s aplikačními branami, nicméně je znát významné zpomalení (zhruba o třetinu až polovinu) proti stavovým paketovým filtrům.

Nevýhodou je zejména to, že z hlediska bezpečnosti designu je základním pravidlem bezpečnosti udržovat bezpečnostní systémy co nejjednodušší a nejmenší. Tyto typy firewallů integrují obrovské množství funkcionality a zvyšují tak pravděpodobnost, že v některé části jejich kódu bude zneužitelná chyba, která povede ke kompromitování celého systému.

Proxy

Proxy server funguje jako prostředník mezi klientem a cílovým počítačem (serverem), překládá klientské požadavky a vůči cílovému počítači vystupuje sám jako klient. Přijatou odpověď následně odesílá zpět na klienta. Může se jednat jak o specializovaný hardware, tak o software provozovaný na běžném počítači. Proxy server odděluje lokální počítačovou síť (intranet) od Internetu.

Aplikační proxy je server speciálně určený pro určitý protokol nebo aplikaci. Proxy server může analyzovat obsah komunikace, případně ji pozměňovat (např. odstraňování reklam z http požadavků, blokování webových stránek podle obsahu a podobně) nebo ukládat požadavky do vyrovnávací paměti (cache), ze které mohou být při opakovaném požadavku odpovědi poskytnuty rychleji.

Typy proxy serverů

- Proxy server, který prochází žádosti a odpovědi nijak neupravuje, se obvykle nazývá gateway nebo tunneling proxy.
- Forward proxy je internetově orientovaná proxy používaná k načítání z celé řady zdrojů (ve většině případů kdekoliv na Internetu).
- Reverzní proxy je obvykle internetově orientovaný proxy server, který se používá k řízení ochrany přístupu k serveru v privátní síti, plnění úkolů jako je vyrovnávání zatížení, autentizace, dešifrování nebo ukládání do mezipaměti.

Obecně tedy vypadá navenek síť za proxy serverem jako jeden počítač. To ztěžuje útoky na jednotlivé počítače v síti, protože útočník musí nejdříve prlomit zabezpečení proxy serveru, pak zjistit strukturu sítě za ním a pak prolomit ochranu jednotlivých serverů, což je mnohem náročnější.

Využití pro proxy server je mnoho. Např. bezpečnost, filtr dat, logování komunikace, caheování komunikace, atd.

11. Šifrování komunikace, ssh, SSL/TLS, VPN, IPsec

ssh

Secure Shell - Náhrada za telnet, která umožňuje komunikaci šifrovaným kanálem.

Umožňuje bezpečnou komunikaci mezi dvěma počítači, která se využívá pro zprostředkování přístupu k příkazovému řádku, kopírování souborů a též jakýkoliv obecný přenos dat (s využitím síťového tunelování). Zabezpečuje autentizaci obou účastníků komunikace, transparentní šifrování přenášených dat, zajištění jejich integrity a volitelnou bezeztrátovou kompresi. Server standardně naslouchá na portu TCP/22.

Má několik vrstev:

- Transportní - Zajišťuje vytvoření spojení, integritu a jestli je na druhé straně stále ten, s kým sme se spojovali
- Autentizační - Ověřuje identitu uživatele. Upožňuje jak autentizaci heslem, tak veřejným klíčem.
- Spojení - definuje koncept kanálů, požadavků kanálů a globálních požadavků skrze které jsou poskytovány SSH služby. Jedno SSH spojení může hostovat více kanálů zároveň, kdy každý může přenášet data v obou směrech. Požadavky kanálů jsou použity pro přenos mimopásmových dat, jako jsou např. změna velikosti terminálového okna nebo návratový kód procesu na straně serveru. SSH klient si může pomocí globálního požadavku vyžádat forwardování (tunelování) portu na straně serveru. Standardní typy kanálů zahrnují:

- shell pro terminálové shelly, SFTP a požadavky exec (zahrnující SCP přenosy
- direct-tcpip pro forwardovaná klient-server spojení
- forwarded-tcpip pro forwardovaná server-klient spojení

SSL/TSL

Secure Sockets Layer/Transport Layer Security je protokol, resp. vrstva vložená mezi vrstvu transportní (např. TCP/IP) a aplikační (např. HTTP), která poskytuje zabezpečení komunikace šifrováním a autentizaci komunikujících stran. Nástupcem SSL je TSL

Nejčastěji se používá k bezpečné komunikaci s webserverem (HTTPS protokol).

Ustavení SSL/TSL spojení funguje na principu asymetrické šifry, kdy každá z komunikujících stran má dvojici šifrovacích klíčů – veřejný a soukromý. Veřejný klíč je možné zveřejnit a pokud tímto klíčem kdokoliv zašifruje nějakou zprávu, je zajištěno, že ji bude moci rozšifrovat jen majitel použitého veřejného klíče svým soukromým klíčem.
Ustavení SSL spojení (*SSL handshake*, tedy „potřásání rukou“) pak probíhá následovně:

1. Klient pošle serveru požadavek na SSL spojení, spolu s různými doplňujícími informacemi (verze SSL, nastavení šifrování atd.).
2. Server pošle klientovi odpověď na jeho požadavek, která obsahuje stejný typ informací a hlavně certifikát serveru.
3. Podle přijatého certifikátu si klient ověří autentičnost serveru. Certifikát také obsahuje veřejný klíč serveru.
4. Na základě dosud obdržených informací vygeneruje klient základ šifrovacího klíče, kterým se bude šifrovat následná komunikace. Ten zašifruje veřejným klíčem serveru a pošle mu ho.
5. Server použije svůj soukromý klíč k rozšifrování základu šifrovacího klíče. Z tohoto základu vygenerují jak server, tak klient hlavní šifrovací klíč.
6. Klient a server si navzájem potvrdí, že od teď bude jejich komunikace šifrovaná tímto klíčem. Fáze handshake tímto končí.
7. Je ustaveno zabezpečené spojení šifrované vygenerovaným šifrovacím klíčem.
8. Aplikace od teď dál komunikují přes šifrované spojení. Například POST požadavek na server se do této doby neodešle.

Během první fáze ustanovení bezpečného spojení si klient a server dohodnou kryptografické algoritmy, které budou použity. V dnešní implementaci jsou následující volby:

- pro výměnu klíčů: RSA, Diffie-Hellman a DSA
- pro symetrickou šifru: RC2, RC4, IDEA, DES, 3DES nebo AES
- pro jednocestné hašovací funkce: MD5 nebo SHA

VPN

Virtual Private Network simuluje komunikaci počítačů v privátní síti přes veřejnou síť. V rámci VPN má každý počítač svou virtuální MAC a IP adresu, takže komunikují, jako by byli na privátní síti. Veškerá komunikace je šifrována.

Používá se k bezpečnému přístupu zvenčí do uzavřené privátní sítě (např. firemní) kde je žádoucí, aby dovnitř měli přístup jen důvěryhodní uživatelé.

IPsec

IP security je bezpečnostní rozšíření IP protokolu založené na autentizaci a šifrování každého IP datagramu. V architektuře OSI se jedná o zabezpečení již na síťové vrstvě, poskytuje proto transparentně bezpečnost jakémukoliv přenosu (kterékoliv síťové aplikaci). Bezpečnostní mechanismy vyšších vrstev (nad protokoly TCP/UDP, kde pracují TLS/SSL, SSH apod.) vyžadují podporu aplikací. IPsec je definován v několika desítkách RFC vydaných IETF, základními jsou 2401 a 2411.

Princip - Vytváří logické kanály – *Security Associations* (SA), které jsou vždy jednosměrné, pro duplex se používají dvě SA.
Bezpečnostní rozšíření vypadá následovně:

- Ověřování – při přijetí paketu může dojít k ověření, zda vyslaný paket odpovídá odesilateli či zda vůbec existuje.
- Šifrování – obě strany se předem dohodnou na formě šifrování paketu. Poté dojde k zašifrování celého paketu krom IP hlavičky, případně celého paketu a bude přidána nová IP hlavička.

Základní protokoly (jsou často používány zároveň, protože se vzájemně doplňují):

- Authentication Header (AH) – zajišťuje autentizace odesílatele a příjemce, integritu dat v hlavičce, ale vlastní data nejsou šifrována.
- Encapsulating Security Payload (ESP) – přidává šifrování paketů, přičemž vnější hlavička není nijak chráněna a není zaručena její integrita.

Na rozdíl od VPN, která vytvoří svůj zašifrovaný packet (celý), pak jej předá IP protokolu, který jej celý zabalí do svého packetu a odešle, IPsec nechá hlavičku a zašifruje jen obsah packetu.

12. Fyzická bezpečnost, přístupové systémy, kamery, elektřina, voda.

Fyzická bezpečnost je je důležitá jako každá jiná. I když fyzický útok na IS je mnohem obtížnější, je také velice nebezpečný.

Zabezpečení Serverů

Je důležité zajistit, aby se k fyzickým strojům nikdo nedostal. K tomu nám poslouží:
- Bezpečnostní dveře a zámky
- Bezpečnostní kamery
- Ochranka

a podle důležitosti dat uložených na serverech, případně vyšší stupeň zabezpečení (ID karty, otisky prstů, snímky sítnice, atd..)

Alarmy, pohybová čidla, senzory rozbitého okna
Kamery se umisťují nad vchod tak, aby zabírali přístupovou cestu a případně pok budovy (kdyby šel někdo ze strany). Pokud je na pozemku alespoň třímetrový stožár, můžeme zabírat i vchod a vnitřek, s tím však mají někteří lidé problém. Pokud snímáme ulici, potřebujeme povolení od úřadů.

Stejně jako by se nikdo neměl dostat k serverům, neměl by mít ani přístup ke kabeláži. I z ní lze dostat citlivá data.

Ochrana proti nečekaným událostem

Pozamykat místnosti a boxy se servery ovšem nestačí. Je potřeba počítat se všemi eventualitami

1. Chlazení - Kvalitní systém klimatizace a cirkulace vzduchu v místnosti, aby se stroje nepřehřívali.
2. Požární ochrana - Snímače a automatické hasící systémy, hasící přístroje vhodné k hašení elektroniky.
3. Výpadek elektrického proudu - Dieselové agregáty na výrobu elektřiny. Pojistné zásuvky, které krátkodobě vyplní výpadek mezi přerušením dodávky elektřiny a nahozením generátorů.
4. Živelné pohromy (povodeň, zemětřesení, atd.) - Duplicitní servery v jiné lokalitě. Duplicitní server musí běžet zároveň s hlavním. Musí být zaručena jeho aktuálnost pro hladký přechod v případě poruchy. Na to je software (zjistit)

řikal,že Vám nakreslí obrazek