Bezpečnost v distribuovaných systémech, kryptografické funkce, ověřování uživatelů, ověřovací servery, obranné valy, DoS.

Operační systémy vyžadují ochranu na třech úrovních:

- ochrana zdrojů – ochrana proti neoprávněnému použití prostředků v OS
- bezpečná komunikace – vlastní ochrana přenášené informace
- ověřování uživatelů – zabezpečení, aby zprávy přicházely od ověřeného zdroje a bez modifikace

Napadení systému:

- pasivní
  - odposlech
  - analýza přenosu – odkud, kam, kolik, ...
- aktivní
- modifikace, zadržování nebo podstrkávání zpráv
- modifikace toku dat – změna obsahu, opakování, změna pořadí, rušení, syntéza zpráv, změna adresy, změna dat, atd

Cíl zabezpečení:

- prevence pasivního útoku
- detekce aktivního útoku.

Ohodnocení bezpečnosti

Existuje více způsobů, uvedeme TCB (Trusted Computing Base)

Klasifikace rozdělení do 4 tříd (A – nejlepší, D – bez zabezpečení)

- skupina D – bez zajištění bezpečnosti, minimální ochrana (MS-DOS)
- skupina C – volná ochrana (ponechána na uvážení)
  - třída C1 – volná ochrana – oddělení uživatele od dat, ochrana dat před neautorizovaným přístupem
  - třída C2 – řízená ochrana přístupu – přihlašování přes LOGIN+heslo
- skupina B – nařízená nebo vynucená ochrana
  - třída B1 – značená ochrana bezpečnosti – všechny objekty mají klasifikační značky, přístup subjektů (procesů) k objektům (zdrojům) řízen, přístup pouze k objektům s nižší klasif. značkou než uživatel
  - třída B2 – strukturovaná ochrana – systém od základů psán podle formálního modelu bezpečnosti, musí být identifikován každý kanál, který může ohrozit bezpečnost systému – rozumná úroveň zabezpečení
  - třída B3 – zavádí oblasti bezpečnosti – systém musí obsahovat monitor odkazů, jsou vytvořeny oblasti bezpečnosti, tj. seznamy uživatelů a skupin s jejich přístupovými právy k objektu a dále seznam uživatelů a skupin, pro které není zaručen žádný přístup
- skupina A – verifikovaná ochrana – vyžaduje úplný formální návrh systému, který je orientován na klasifikaci informace
  - třída A1 – systém s verifikovaným návrhem – obdoba B3 a navíc úplný formální návrh

Ochrana zdrojů v OS:

- přístupovou maticí – obsahuje – model informačního toku, objekt, jeho typ a povolené operace, subjekty, které mají právo manipulovat, přístupová prává a oblasti jejich použití
- přístupovým seznamem
- seznamem schopností (capabilities)

Zajištění bezpečné komunikace:

- zaměření na linku – zabezpečení na linkové úrovni
  - musí být transparentní pro uživatele
  - zabezpečení a šifrování kontinuálního toku dat
  - není vhodné pro otevřené systémy – není zaručena bezpečnost v koncových a mezilehlých uzlech (pouze dvoubodové sítě)
- zaměření na koncové uzly – šifrování mezi koncovými uzly.
  - použitelné jak v dvoubodových, tak mnohabodových sítích.
- zabezpečení na úrovni spojení – na relační nebo aplikační úrovni
  - volba úrovně zabezpečení programátorem

Bezpečnost

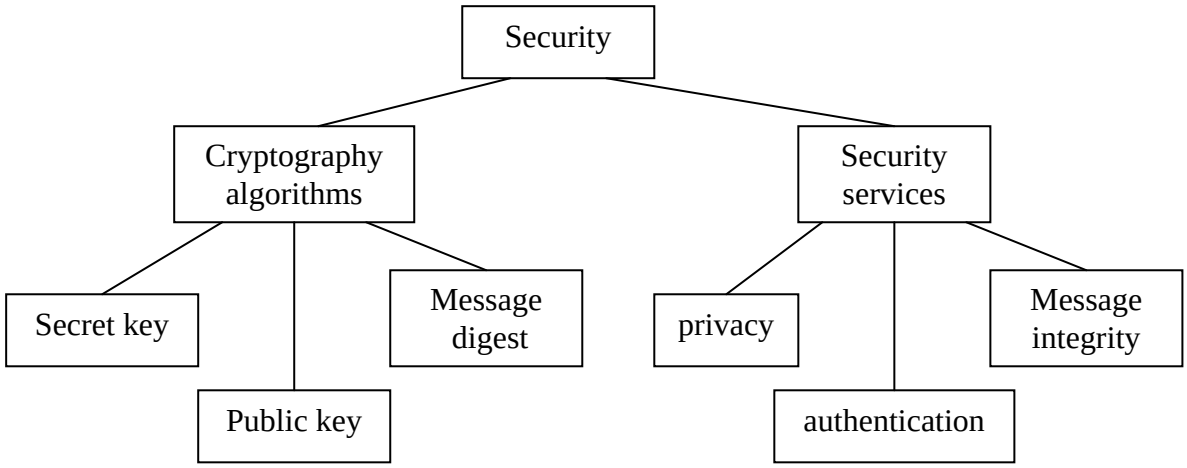
- požadavky na bezpečnost se v poslední době výrazně mění
- tradičně byla zajišťována zamezením přístupu (uzamykáním a administrativně)
- se zavedením výpočetní techniky vznikla potřeba vytvářet automatizované prostředky pro ochranu souborů a dalších informací
- použití počítačových sítí a komunikačních linek vyžaduje zajistit ochranu dat během přenosu

Definice

- počítačová bezpečnost – všeobecný název pro soubor prostředků, navržených k ochraně dat a maření úsilí hackerů
- síťová bezpečnost – opatření k ochraně dat během přenosu
- bezpečnost Internetu – opatření k ochraně dat během přenosu přes soubor propojených sítí
  - spočívá v opatření k odrazení, prevenci, detekci a korekci bezpečnostních hrozeb poškozujících přenos informace

Služby, mechanizmy, útoky

- bezpečnostní služby – zvýšení bezpečnosti přenosu a zpracování dat
- bezpečnostní mechanizmus – navržen k detekci, prevenci a obnově po bezpečnostním útoku, používá šifrovacích technik
- bezpečnostní útok – jakákoliv akce, která naruší bezpečnost informací



Bezpečnostní architektura OSI

- doporučení ITU-T X.800, v Internetu RFC 2828
- X.800 definuje 5 hlavních kategorií
  - authentication – ověření pravosti – ujištění, že entita je to, za co se vydává
  - access control – řízení přístupu – zamezení neautorizovaného využívání zdrojů
  - data confidentiality – důvěrnost dat – ochrana dat před neautorizovaným přístupem
  - data integrity – integrita dat – ujištění, že přijatá data byla odeslána ověřenou entitou
  - non-repudiation – nepopiratelnost – ochrana proti popření jednou z komunikujících entit

Bezpečnostní mechanizmy

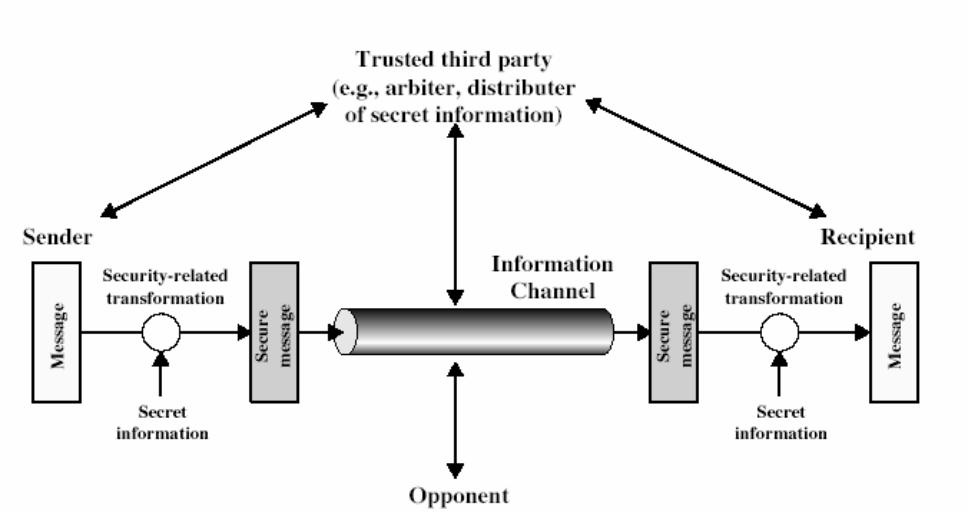
- šifrování
- digitální podpisy
- řízení přístupu
- integrita dat
- ověřování výměny dat
- vyplňování přenosu
- řízené směrování
- ověřování třetí stranou

Klasifikace útoků

- pasivní útoky
  - odezírání, monitorování (získání obsahu, monitorování toků)
- aktivní útoky
  - přerušení přenosu
  - maskování za jinou entitu
  - opakování předchozích zpráv

- modifikace zpráv během přenosu
- odepření služby

Model síťové bezpečnosti



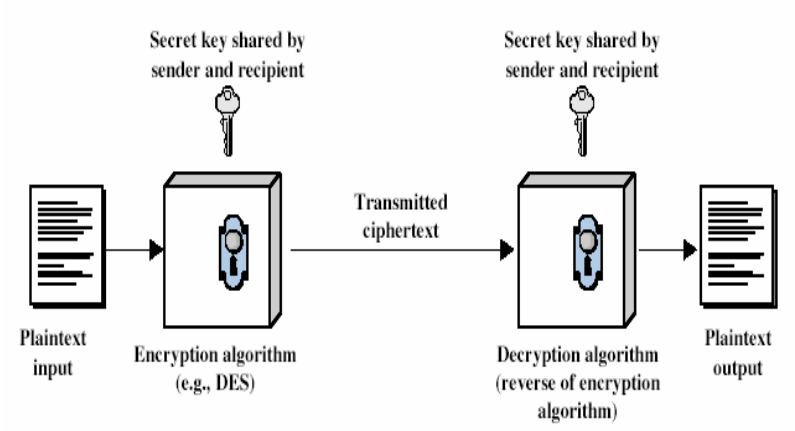
Šifrování

- symetrické (konvenční, tajný klíč, jeden klíč)
- asymetrické (tajný a veřejný klíč)

Terminologie

- otevřený text (plaintext)
- šifrovaný text (ciphertext)
- šifra – algoritmus pro transformaci otevřeného textu na šifrovaný
- klíč – parametr šifrování
- šifrování – převod otevřeného textu na šifrovaný
- dešifrování – převod šifrovaného textu na otevřený
- kryptografie – studium šifrovacích principů a metod
- kryptoanalýza – studium principů a metod pro dešifrování bez znalosti klíče
- kryptologie – kryptografie a kryptoanalýza

Symetrický model šifrování



Požadavky

- silný šifrovací mechanizmus
- šifrovací klíč zná pouze odesílatel a příjemce
- známý šifrovací (a dešifrovací) algoritmus
- bezpečný kanál pro distribuci klíče

$Y = E_K(X)$
$X = D_K(Y)$

Šifrovací operace

- substituce
- transpozice

Šifra

- bloková
- proudová

Útok hrubou silou

- absolutní bezpečnost – bez znalosti klíče nelze odhalit otevřený text
- výpočetní bezpečnost – šifra nemůže být zlomena pro nedostatečnou výpočetní výkonnost

Key Size (bits)	Number of Alternative Keys	Time required at 1 encryption/μs	Time required at 10^6 encryptions/μs
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} years	6.4×10^{6} years

Monoalfabetické šifry

- náhodné přiřazení písmen (klíč 26 písmen dlouhý – 26! = $4x10^{26}$)

Caesarova šifra (substituční)

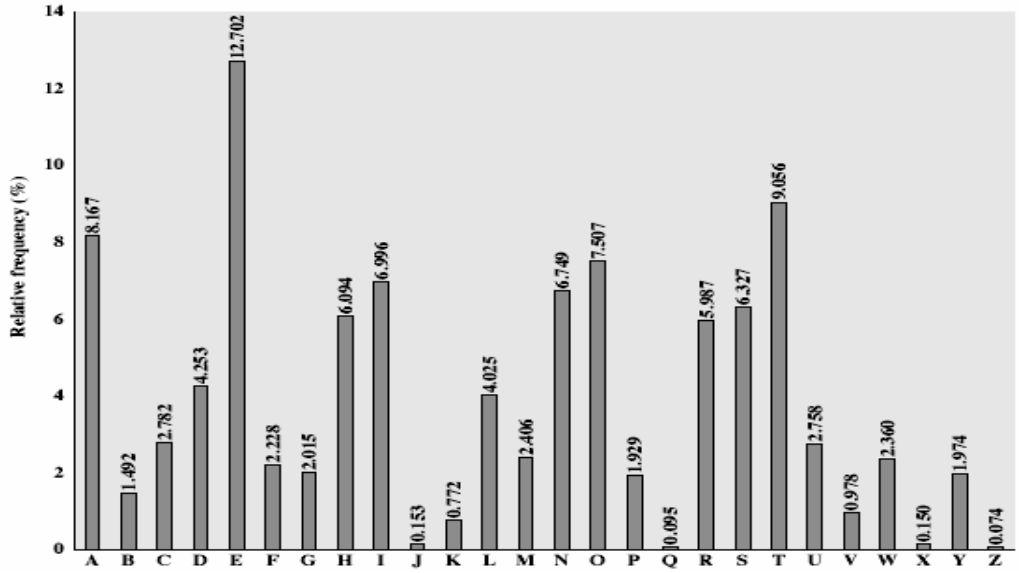
- pouze 26 možností
- útok hrubou silou

Plain: abcdefghijklmnopqrstuvwxyz
Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN
Plaintext: ifwewishtoreplaceletters
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA

Polyalfabetické šifry

- kombinace transpozice a substituce
- šifrování na dané pozici závisí na klíči, šifrování pozic se opakuje s periodou délka klíče
- řešení je nalézt délku klíče a pak jde a o několik monoalfabetických šifer

Frekvenční analýza



Šifrování tajným klíčem, symetrické metody šifrování.

Substituční šifry

- Každé písmeno nebo skupina písmen je nahrazena jiným písmenem nebo skupinou písmen
- Např. Caesarova šifra – použita Caesarovými vojsky
- Jednoduše prolomitelné

Transpoziční šifry

- Přeuspořádání písmen, ale ne překódování
- Sloupcové šifrování – otevřený text je šifrován po sloupcích různými klíčovými slovy
- Ne tak jednoduché prolomení jako u substitučních šifer.

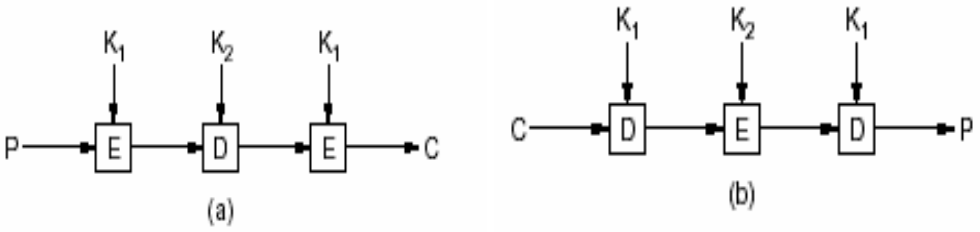
Jednorázová hesla

- Šifrovaný text je vytvářen konverzí otevřeného textu na bitový řetězec a XOR-ován s náhodným bitovým řetězcem. Délka přenášených dat je omezena délkou řetězce (klíče)
- Neprolomitelná šifra
- Klíč je obtížné si pamatovat – odesílatel i příjemce musí přenášet i kopii klíče
- Vyžaduje striktní synchronizaci mezi odesílatelem a příjemcem. Jeden chybějící bit může pomotat cokoliv

Příklady – šifrování tajným klíčem

1. DES – Data Encryption Systém
   - Každá iterace i používá jiný klíč K_i. Složitost závisí na komolící funkci *f*.
   - Klíč K_i je odvozován od počátečního 56 bitového klíče.
   - Šifrovací algoritmus vyvinut v r. 1970 National Bureau of Standards and Technology a IBM.
   - Používá délku klíče 56 bitů a 19 různých stavů
   - Velmi silný, ale prolomitelný

2. Triple DES – řeší problém příliš krátkého klíče DES jeho rozšířením na 112 bitů
   - Pro šifrování postupně používá algoritmus šifrování klíčem K_1, dešifrování klíčem K_2 a šifrování klíčen K_1.
   - Pro dešifrování postupně používá algoritmus dešifrování klíčem K_1, šifrování klíčem K_2 a dešifrování klíčen K_1.



3. AES/Rijndael
   - DES je nyní příliš slabý
   - Nyní nahrazován vítězem konkurzu o šifrovací standard (AES – Advanced Encription Standard) – Rijndael.
   - délka klíče 128, 196 nebo 256 bitů

4. IDEA – International Data Encription Standard
   - Publikován v r. 1990
   - Používá klíč délky 128 bitů
   - Velmi silné šifrování, nebyly publikovány žádné praktické útoky, útok hrubou silou není praktický
   - Pokrytý různými mezinárodními patenty

5. Skipjack
   - Tajný algoritmus vyvinutý NSA
   - Je použit v šifrovacím čipu Clipper

- Využívá klíč délky 80 bitů

Metody šifrování veřejným klíčem

RSA – vytvořeno pány Rivest, Shamir a Adlemin v r. 1978

- Velmi silná šifra
- Podporuje proměnnou délku klíčů
- Délka klíče 1024 bitů, 2048 bitů
- Delší klíče zajišťují větší bezpečnost
- Algoritmus založen na počítání s velkými prvočísly

p, q … velká prvočísla N = p.q
P.S = 1 [mod $\phi(N)$] $\phi(N)$ = nsn(p-1, q-1)
zašifr. $C = M^P \pmod N$
dešifr. $M = C^S \pmod N$
P … public key, S … secret key
– předává se P, N a utají S
– výpočet P, S, N musí být jednoduchý

Další systémy s veřejným klíčem

- Elgamal (Taher Gamal)
- Diffie-Hellman
- DSA (Digital Signature Algorithm)

ElGamal

- zahrnuje tři komponenty
  - generátor klíče
  - šifrovací algoritmus
  - dešifrovací algoritmus
- generátor klíče
  - zvolí se velké náhodné číslo p takové že p-1 = kq, kde k je malé a q je prvočíslo
  - nalezne se takové g, že $g^q = 1 \bmod p$ a $g \neq 1$
- Alice zvolí náhodné $x \in \{0, ..., q-1\}$ a vypočte $h=g^x$
- Alice publikuje popis množiny, ze které generovala g a {h, g, q} (veřejný klíč)
- Bob volí náhodné $y \in \{0, ..., q-1\}$ a vypočte $c_1=g^y$ a $c_2=m.h^y$, kde m je upravená zpráva
- Bob posílá Alici šifrovaný text $\{c_1, c_2\}$
- Alice vypočte $c_2(c_1^x)^{-1} = \frac{m \cdot h^y}{g^{xy}} = \frac{m \cdot g^{xy}}{g^{xy}} = m$

Je-li zpráva větší než G, musí být rozdělěna na několik částí

Výměna klíčů Diffie-Hellman (1976)

- Problém: Dvě entity (Alice a Bob) chtějí vzájemně bezpečně komunikovat přes otevřenou síť, aniž by sdílely nějaké tajemství

- Základní myšlenka spočívá ve faktu, že vypočítat g^a je rychlé, zatímco vypočítat a z g^a je mnohem těžší
- Alice najde veliké prvočíslo n, generátor g a náhodné číslo x, kde
  - prvočíslo n je 512 bitů dlouhé nebo delší
  - generátor g << n je také prvočíslo
  - pro každé i < n existuje a takové, že platí $g^a = i \bmod n$
- Alice vygeneruje náhodné číslo x, vypočte $X = g^x \bmod n$
- pošle X, g, n Bobovi
- Bob vygeneruje náhodné číslo y, vypočte $Y = g^y \bmod n$
- Bob pošle Y Alici
- Alice vypočte $K_{AB} = Y^x \bmod n$
- Bob vypočte $K_{AB} = X^y \bmod n$
- Obě hodnoty jsou stejné
  - Bez znalosti x nebo y není možné jednoduše vypočítat K_{AB}
  - Diskrétní logaritmus mod n je příliš složitý
- Vypočtenou hodnotu K_{AB} použijí jako tajný klíč
- Nevýhoda metody spočívá v tom, že Alice i Bob musí spolupracovat v reálném čase. Nehodí se např. pro dohodu na klíčí pro elektronickou poštu.

Použití šifrování s veřejným klíčem

- šifrování zpráv (časově náročné)
- šifrování relačního klíče (relační klíč se použije k šifrování (symetrické) vlastní zprávy, výměna klíčů
- ověření integrity dat (ke zprávě se pomocí hashovací funkce vygeneruje otisk, který se zašifruje tajným klíčem odesílatele – je schopen provézt pouze majitel tajného klíče – plus ověření pravosti)
- nepopiratelnost (informace zašifrovaná tajným klíčem)

Hashovaní funkce

$h = f(m)$ h … výsledek, m …zpráva

$h_1 \neq h_2 \Rightarrow m_1 \neq m_2$ a $m_1 = m_2 \Rightarrow h_1 = h_2$

– metody: SHA

MD5 – kromě zprávy m pošlu $\{h\}_{KS}$,

kde $h = f(m)$

– neexistuje inverzní funkce k f

– zpráva se znovu zakóduje a porovnají obrazy

$\{\{h\}_{KS}\}_{KP} \rightarrow h$

$m \rightarrow h_1 = f(m) \rightarrow h_1 = h$

Digitální podpisy

- Garantují autentičnost digitálně podepsané zprávy
- Digitální podpis je sám o sobě šifrován tajným klíčem, aby se dala potvrdit
  Autenticita
  Integrita
  Pravost podpisu - nedal se popřít

- Podpisy tajným klíčem
  - Jako úložiště všech digitálních podpisů je použita centrální autorita
  - Centrální autoritě musí všichni věřit
- Podpisy veřejným klíčem
  - Zpráva je šifrována veřejným klíčem odesílatele a dešifrována tajným klíčem odesílatele

Autentikace

- Autentikace je technika, pomocí které se ověřuje, že komunikující partner je ten, za kterého se vydává a ne podvodník.

Existují tři způsoby autentikace

- Řekni něco co víš (heslo)
- Ukaž něco co máš (identifikační karta)
- Nech systému něco tvého změřit (otisk prstu)

Ověřovací schémata

- musí obsahovat aspoň jedno tajemství
- musí být schopna rozpoznat jeho správné použití

Ověřovací metody

- jednoduché (založeny na heslech)
- přísné (založeny na šifrovacích metodách)

Jednoduché ověřování

- identifikace jménem a heslem,
- přenos otevřeného textu, použití ověřovacího serveru

Přísné metody

- elementární metody – použití symetr. a nesymetr. kódů
- metody založené na ověřovacích serverech
- metody založené na protokolech s minimální znalostí
  - uživatel dokazuje svoji identitu odpovídáním na šifrované otázky serveru
    M1: {R, ID}
    M2: $\{C\}_K$
    M3: $\{f(C)\}_K$
    R … požadavek, K … tajný klíč, C … náhodné číslo, f(C) … domluvená funkce

Ověřovací servery

- slouží k ověření „pravosti“ uživatele
- lepší utajení klíčů

- používá se KDC (Key Distribution Center) – databáze klíčů (je tajná a indexována podle jmen uživatelů)

Certifikační autority

- Používají se k administraci a ověřování veřejného klíče.
- Musí být důvěryhodnou stranou.
- Umožňují ověřování uživatele v rozsáhlém systému – decentralizované ověřování.
- Princip – veřejný klíč je předáván ve formě, jejíž pravost lze ověřit pomocí ověřeného veřejného klíče certifikační autority.
- Certifikát je blok dat (soubor), obsahující:
  - Verze (V3)
  - Sériové číslo (02 1c 6a)
  - Algoritmus podpisu (md5RSA)
  - Vystavitel (CN = CA GE Capital Bank, OU = Direct Banking, O = GE Capital Bank, a.s., C = CZ)
  - Platnost od (28. dubna 2003 12:31:30)
  - Platnost do (27. dubna 2005 12:31:30)
  - Předmět (E = ledvina@kiv.zcu.cz, CN = uid: 120295, CN = Ing. Jiri Ledvina, ... adresa)
  - Veřejný klíč (30 81 87 02 81 81 00 bf 4a ...)
  - Distribuční místo (URL=http://www.gecb.cz/ca_ge.crl)
  - Použití klíče (Digitální podpis, Zakódování klíče)
  - Algoritmus miniatury (sha1)
  - Miniatura (72 19 13 5c 6a 9b 4e ab 30 cf 6b 6f 49 df 15 c0 62 94 79 09)
  - Popisný název (Ing. Jiri Ledvina)
- Certifikát musí být nezpochybnitelný – zneplatnění certifikátu
- Existují různé formáty certifikátů
  - Personal Information Exchange (PEX), PKCS #12 (P12) (Public Key Cryptography Standard)
  - Cryptographic Message Syntax Standard PKCS#7 (P7B)
  - PGP certifikáty

Ověřování certifikátů

- přímé ověřování (nejjednodušší model)
  - Ověřování certifikátů mezi důvěryhodnými subjekty
  - V prohlížečích jsou certifikáty uznávaných autorit instalovány – můžeme (musíme) jim věřit. Existují ale i další certifikační autority, které nejsou uznávané – prohlížeč se na důvěryhodnost ptá.
- hierarchické ověřování – zřetězení certifikátů
  - Certifikačních autorit je hodně – získání certifikátu může být otázkou osobní návštěvy (důvěryhodné získání certifikátu).
  - Certifikační autority mohou vytvářet hierarchický strom – důvěryhodnost CA nižší úrovně je potvrzována CA vyšší úrovně. CA nejvyšší úrovně potvrzuje důvěryhodnost sebe sama.
  - Zřetězení CA je součástí certifikátu.
- kumulativní model – zahrnuje předchozí (přímé, zřetězené)

Protokoly pro bezpečnou komunikaci

Kerberos – ověřování v systému Orion na ZČU

- Používá symetrické šifrování
- Vychází z centralizované databáze uživatelů (každý uživatel musí být registrován)
- Základní část je ověřovací server (Kerberos)
- Po přihlášení (ověření) dostane uživatel lístek, obsahující práva přístupu k požadovanému serveru.
- K dalšímu ověřování uživatele se používaní pověřovací listiny (credentials), obsahující jméno uživatele a adresu jeho počítače.

SSL – Secure Socket Layer

- Vyvinuto fy. Netscape, používá se zejména pro bezpečné přenosy mezi prohlížečem a webovým serverem.
- K ověřování serveru se používají certifikáty serveru. Uživatel není ověřován.
- Po ověření se veřejný klíč použije pro vygenerování relačního klíče, sloužícího k šifrování komunikace.
- Schéma bezpečného HTTP se označuje HTTPS
- SSL se používá i u dalších protokolů (POP, IMAP)
- Je možné je využít univerzálně – vytváří mezivrstvu mezi protokolem TCP a aplikací – před použitím je třeba aplikaci (program) modifikovat.
- Obdobou SSL je TLS (Transport Level Security)

SSH – Secure Shell

- Používá se pro vytvoření šifrovaného kanálu mezi aplikacemi (aplikační úroveň).
- Pro šifrování používá opět relační klíč, vytvořený na základě výměny informací (Diffie - Hellman algoritmus pro výměnu klíčů) nebo na základě asymetrické kryptografie – RSA.

Využívá se pro

- bezpečný vzdálený přístup – náhrada Telnetu (ssh – secure shell),
- bezpečný přenos souborů – náhrada ftp (scp – secure copy),
- vytvoření bezpečného kanálu mezi libovolnými aplikacemi.

Více o SSH

- SSH je realizace bezpečného vzdáleného virtuálního terminálu
  - Zajišťuje šifrovanou komunikaci mezi nedůvěryhodnými počítači přes nedostatečně chráněnou síť
    - Předpokládá, že je možné odposlechnout všechnu komunikaci mezi hosty
    - Poskytuje různé metody ověřování
    - Šifruje data vyměňovaná mezi hostitelskými systémy
  - Bylo určeno jako náhrada nedostatečně chráněných programů, jako je rlogin, rsh atd.
  - Zahrnuje i schopnost pro bezpečný přenos souborů
    - Secure copy (scp)
  - Zahrnuje schopnost bezpečně fowardovat spojení X11 i TCP porty
- Je velmi populární a často používaný
  - Není nezranitelný

Ověřování v SSH1

- Prostředky pro ověřování podporované v SSH
  - Jednoduché ověřování pomocí rhosts
    - Uživatelské/systémové jméno v ~/.rhosts a ~/.shosts
    - Snadno zranitelný IP/DNS spoofingem
    - Pro tento režim činnosti vyžaduje zvláštní komplilaci
  - Ověřování založené na ověřování hostitelských systémů
    - Použití RSA k ověření klíče hostitelského systému
    - Používá soubor ~/.rhosts pro ověření uživatele
  - Ověřování založené na uživatelích a hostitelských systémech
    - Ověřování RSA klíče hostitelského systému
    - Ověřování RSA klíče uživatele
- Pokud skončí ověřování chybou, je klient vyzván k zadání hesla
  - Všechna komunikace je šifrována

Protokol pro výměnu klíčů v SSH1

- Server má pár veřejný/tajný klíč
  - Klient zná předem veřejný klíč serveru
    - Musí být poslán předem bezpečným kanálem
- Server pošle klientovi veřejný klíč a náhodný klíč serveru
  - Klient ověří veřejný klíč
- Klient pošle náhodný relační klíč zašifrovaný hostitelským a serverovým klíčem
  - Zbytek relace je šifrován relačním klíčem

Protokol pro výměnu klíčů v SSH2

- Je použit Diffie-Hellman algoritmus pro výměnu klíčů
  - Algoritmus založený na principu přenosu veřejných klíčů
  - Dva uživatelé si mohou vyměnit tajný klíč nedůvěryhodnou linkou bez předchozího sdílení jakéhokoliv tajemství
- Digitální podpis ověřuje identitu serveru vzhledem ke klientovi
- Výsledkem výměny klíčů je sdílený tajný klíč
  - Používá se pro šifrování do konce relace
- Datová integrita je kontrolována pomocí MD5
- Podporuje několik šifrovacích mechanizmů
  - IDEA, Blowfish. DES, Triple DES, …

SSH v praxi

- Veřejný a tajný klíč hosta se generuje při instalaci SSH
  - Veřejný klíč musí být v ~/.ssh/known_hosts na vzdálených systémech
- Ke generování uživatelských veřejných a tajných klíčů je použit příkaz ssh-keygen

    - Vyžaduje aby uživatel vložil heslo
    - Veřejný klíč je kopírován do ~/.ssh-authorized_keys na vzdálených systémech
- ssh-agent a ssh-add eliminují potřebu pro opakované psaní hesla
- ověřování heslem je zranitelné použije-li se útok hádáním hesla
- X11 a forwardování portu vytváří šifrovanou rouru skrz Internet
    - Může být použita pro zajištění zabezpečeného přístupu nezabezpečeným aplikacím jako je SMTP.
    - Může být použito k obejití obranných valů
- Dostupné jako Open Source software (OpenSSH)

IPsec – IP security

Co je to IPsec

- Soubor protokolů pro zajištění bezpečnosti na síťové úrovni
    - Ověřování původu
    - Integrita dat
    - Utajení dat
- Vzhledem k transportním protokolům a aplikacím je transparentní – nevidí ho
- Vzhledem k linkovému protokolu neprůhledný – nerozumí přenášeným datům
- Přizpůsobivý

Režimy činnosti

- Transparentní – mezi koncovými uživateli
- Tunelovaní – mezi dvěma síťovými prvky (směrovači, obrannými valy, …)
- Kombinace předcích – mezi koncovým uživatelem a síťovým prvkem

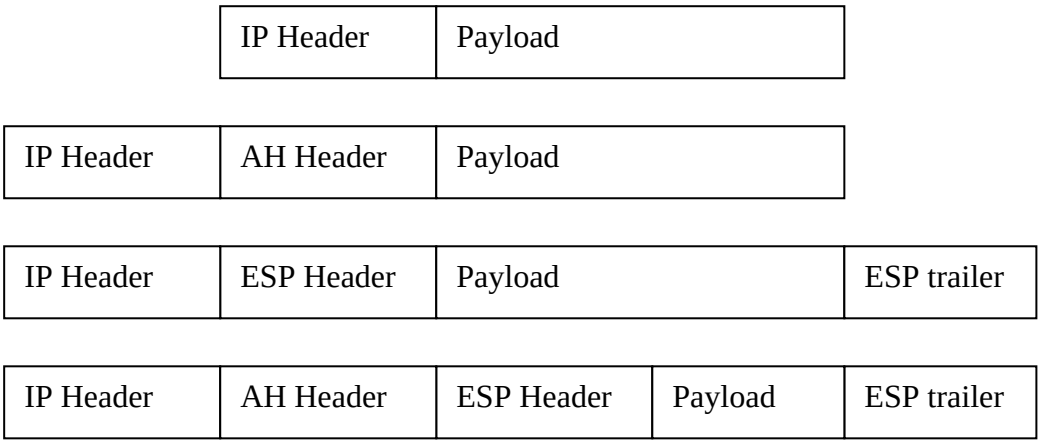
IPsec – vzdálený pohled

- Přenosové protokoly
    - Encapsulation Security Payload (ESP) – šifrování zpráv
    - Authentication Header (AH) – ověřování
- Ochrana
    - Přístupová práva
    - Konfigurace
- Správa klíčů
    - Manuální
    - Automatická (Photuris, IKE)

Přenosové protokoly

- Zapouzdření uživatelských dat s jejich šifrováním (Encapsulation Security Payload

(ESP))
    - Důvěrnost (confidentality)
    - Integritu dat
    - Ověřování původu (Origin authentication) - implicitní
    - Ochrana proti „přehrávání“ (Replay protection)
- Ověřovací záhlaví (Authentication Header (AH))
    - Integrita dat
    - Ověřování původu (Origin authentication) - explicitní
    - Ochrana proti „přehrávání“ (Replay protection)
- Zapouzdření záhlaví

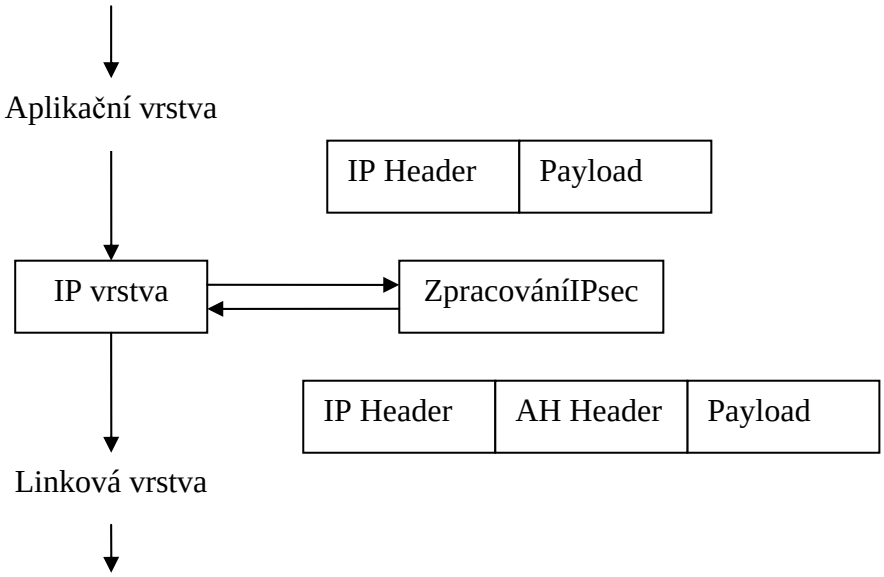


Formát rámců

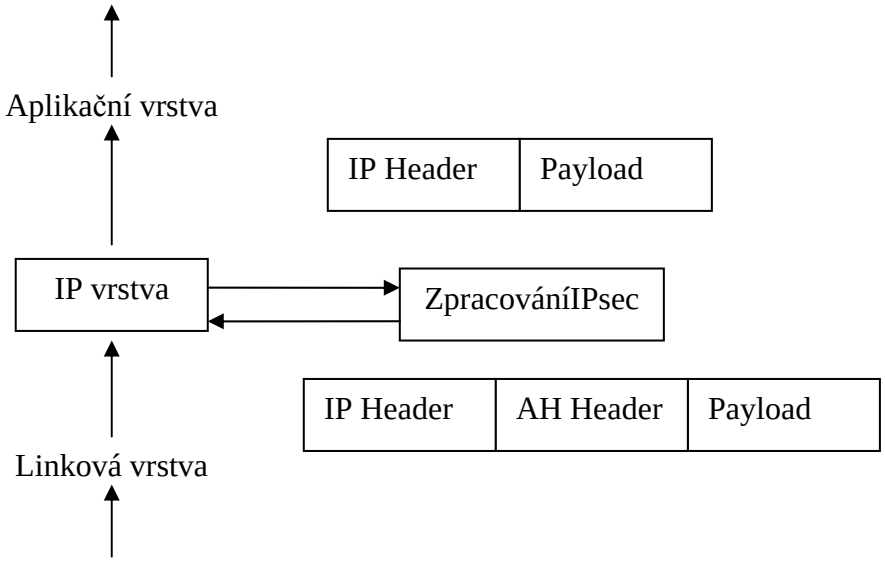
SA Security Association – bezpečné propojení

- SA – vztah mezi dvěma nebo více entitami, který popisuje, jak budou entity využívat bezpečnostní služby k bezpečné komunikaci
- SA je identifikován dvojicí bezpečnostní protokol, SPI
- SPI je Security Parameter Index – může být implementačně závislý na systému, protokolu, atd.
- Informace týkající se toku dat v IPsec
    - Kryptografické algoritmy
    - Klíče
- Security Policy Database (SPD)
    - Příchozí/odchozí politika
    - Co akceptovat, co odmítnout, zpracování IPsec

Odchozí zpracování



Příchozí zpracování



Správa klíčů

- Manuální/statická správa klíčů
  - Obtížná
  - Vyžaduje významný zásah člověka
  - Náchylný k chybné konfiguraci
  - Typicky slabé klíče
  - Špatně rozšiřitelný (škálovatelnost)
  - Otravný

- Správu klíčů je třeba automatizovat

Požadavky na správu klíčů

- Vyjednávání parametrů
- Silná bezpečnost
  - Ověřování, klíče
- Dynamická změna klíče
- Minimální konfigurace
- Nezávislost na algoritmu
- Ochrana identity
- Potřeba bezpečného forwardování
- Výkonnost
- Rozšiřitelnost

Dostupné prostředky

- Všeobecně dostupné šifrovací protokoly a systémy
  - Diffie-Hellman výměna klíčů,
  - RSA/DSA,
  - 3DES/AES,
  - MDA5, SHA1

- Struktura pro psaní bezpečnostních protokolů
  - Standardizovaná struktura uživatelských dat
  - Zavedeny typy výměny
  - Pravidla pro zpracování uživatelských dat
  - Přizpůsobivost
- Domain of interpretation concept

ISAKMP

Přesně definuje procedury a formáty paketů pro vytvoření, dohadování, modifikaci a zrušení SA (Security Association). SA obsahuje všichny informace požadované pro vykonávání různých síťových bezpečnostních služeb, jako jsou služby IP vrstvy (ověřování záhlaví AH a zapouzdření dat ESP), služby transportní a aplikační vrstvy, nebo vlastní ochrana přenosu při dohadování. ISAKMP definuje data pro výměnu klíčů a ověřování dat. Tyto formáty poskytnou shodný základ pro přesun klíče a ověřování dat která jsou nezávislá na technice generován klíče, algoritmu šifrování a mechanizmu ověřování.
ISAKMP je odlišný od protokolů výměny klíčů protože jasně odděluje detaily ovládání SA (a správu klíčů) od detailů výměny klíčů. Může existovat mnoho různých protokolů pro výměnu klíčů, každý s různými bezpečnostními vlastnostmi. Avšak společný rámec je potřeba pro dohadování na formátu SA atributů, pro vyjednávání, modifikaci a rušení SA. ISAKMP slouží jako běžný základ.
ISAKMP může být implementovaný nad jakýmkoliv transportním protokolem.
Všechny implementace musí zahrnovat posílání a příjem kapabilit (schopností) pro ISAKMP prostřednictvím UDP/500.

Internet key exchange (IKE)

- Kombinace ISAKMP a Oakley
- Používá UDP/500
- Dvoufázový protokol
  - Vytvoření bezpečného kanálu
  - Ověřování účastníků
  - Výměna aplikačních parametrů
- Existují různé ověřovací mechanizmy
- Existují různé mechanizmy pro výměnu klíčů
  - Diffie-Hellman
  - Kerberos

Zabezpečení elektronické pošty

PEM (Privacy Enhancement for Internet Electronic Mail)

- Dnes historický protokol pro vytváření a zpracování bezpečných zpráv.
- Vznikl v druhé polovině 80. let.
- Původní specifikace RFC989, poslední specifikace RFC1421 až RFC1424 (1993).
- V praxi nedošlo k jeho masovému využití nejširší veřejností - nebyl totiž běžně dostupný software, který by jej podporoval.
- Na přelomu 80. a 90. let nebyla ještě masová poptávka po software tohoto druhu.
- Stal základem pro novější protokoly (S/MIME)

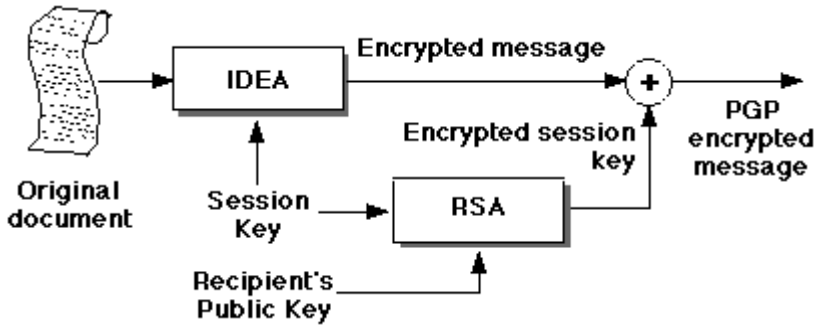
S/MIME

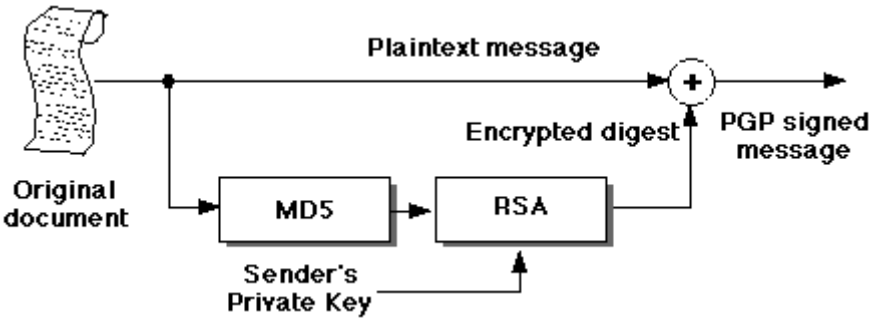
- Podobné PEM
- Kontrolní součet (otisk) SHA-1 a MD5
- Asymetrické šifrování (šifrování symetrických šifrovacích klíčů a elektronický podpis): RSA s délkou klíče minimálně 512 bitů.
- Symetrické šifrování - šifrování textu zprávy (DES-CBC, triple DES).
- Norma PKCS-7 pro tvorbu bezpečných zpráv - elektronický podpis, šifrování, obojí.
- Definuje MIME hlavičku Content Type: Application/pkcs7-mime

PGP (Pretty Good Privacy)

- Uživatelsky jednoduchý program dostupný nejširší veřejnosti.
- PGP je nejrozšířenější prostředek pro zpracování bezpečných zpráv (RFC1991).
- Vytvořil Američan P.R.Zimmerman (1991).
- Bezpečný přenos zpráv pomocí SMTP, POP, IMAP (nepotřebuje nový protokol, nadstavba nad stávajícími).
- Asymetrické šifrování - RSA (šifrování symetrického relačního klíče pro šifrování vlastní zprávy).
- Symetrické šifrování algoritmus - IDEA.
- Komprese dat před šifrováním - PKZIP.
- Výpočet kontrolního součtu (otisku) - MD5.

- Převod binárních dat na ASCII - Radix-64.





Steganography

(a) Three zebras and a tree. (b) Three zebras, a tree, and the complete text of five plays by William Shakespeare.

Obranné valy (Firewalls)

- Provádí ochranu sítě před napadením (ochrana počítačů nestačí)
- Odděluje uživatele (prvek nespolehlivosti) od prvků ochrany

Vlastnosti

- Filtrování paketů a vlastnost odstínění
- Různé úrovně ověřování
- Přihlašování (registrace) a účtování
- Transparentnost a přizpůsobení uživatelům
- Ovladatelnost (management)
- Rozlišení požadavků dle klientů nebo sítí

Realizace obranných valů

- Komerční produkty
- Vlastní realizace

Nechráněná síť

- Jednotlivé hostitelské systémy mohou být dosažitelné z libovolných systémů Internetu
  - Je třeba chránit interní hostitelské systémy
  - Tato situace je nezvládnutelná prostředky operačního systému a není bezpečná

Chráněná síť

- Spojení mezi vnitřními a vnějšími hostitelskými systémy je filtrováno a chráněno odděleným systémem – obranným valem.
  - Jeden silný bod ochrany
  - Daleko ovladatelnější a bezpečnější

Základní typy obranných valů

- Kombinace technických a programových prostředků
  - Technické prostředky – směrovač a/nebo počítač (UNIX)
  - Programové prostředky – přístupový seznam ve směrovači, specializovaný oddělovací program

Základní rozdělení podle úrovně filtrování

- Filtrování na síťové úrovni (IP filtrování)
- Filtrování na transportní úrovni (úroveň spojení)
- Filtrování na aplikační úrovni (aplikační filtry)

V praxi kombinace výše uvedených

- Založeno na podpoře klientů v hostitelských počítačích
- Založeno na službách podporovaných obranným valem pro „své“ klienty
- Založeno na podmínkách závislých na bezpečnosti, pružnosti a transparentnosti

Typy filtrů

- Filtry pro konkrétní služby
- Filtry nezávislé na službách

Filtry pro služby

- Specifikace pro konkrétní port
- Filtrování standardních služeb (Telnet, SMTP, FTP, http, Gopher, DNS)

- Filtrování podle směru navazovaného spojení

Filtry nezávislé na službách

- Poskytují ochranu proti útokům založeným na vlastnostech TCP
  - Source IP spoofing attack
  - Pakety s IP volitelnými parametry (source routing …)
  - Tunelování IP over IP
  - Pokusy o degradaci služeb pomocí ICMP zpráv

Další komplikace filtrování na IP úrovni

- Interní adresy jsou viditelné na Internetu
- Interní klienti jsou schopni předávat externí resoluce jméno/adresa (DNS běžící na interní síti může kooperovat s vnějším DNS.

Principy filtrování na IP úrovni

- Jednotlivé pakety jsou analyzovány a filtrovány (blokovány nebo propouštěny)
- Filtrovací kritéria
  - IP adresa (zdrojová, cílová, obě)
  - Port (zdrojový, cílový, oba)
  - Typ paketu (IP, jiný)
- Nejsou filtrována žádná aplikační data
- Obecně bezestavové filtrování
  - Nejsou k dispozici žádné znalosti o spojení klient/server
  - Nezávisle filtruje přicházející a odcházející pakety
- Výhody
  - Transparentní vzhledem k účastníkům, jednoduše přizpůsobitelné
    - Podporuje libovolný protokol klient/server
    - Není třeba modifikovat ani server, ani klienta
  - Jednoduché levné odstínění (směrovač)
  - Vysoká propustnost
- Nevýhody
  - Méně bezpečné
    - Bezestavový charakter
    - Založeno na omezeném filtrování
    - Využívá implicitní předpoklady
    - Slabé ověřování (IP adresa)
    - Nebrání „prosakování“ IP paketů
  - Není filtrován vlastní protokol server/klient
  - Neumožňuje (nebo jen omezeně) logování a účtování
  - Pravidla filtrování mohou být složitá a náchylná k chybám
  - Může se stát ne-managementovatelný
- Závěr
  - Jednoduché a laciné filtrování na IP úrovni je bezpečné pouze pro
    - Blokování všech paketů (deny)
    - Propouštění všech paketů (allow)

Principy filtrování na aplikační úrovni

- Hlavní princip – vnitřní pakety nesmí přecházet přímo do vnější sítě a naopak
- Klient se spojuje s obranným valem, ne přímo se serverem
- Na obranném valu je umístěn proxy (zástupce), který přijímá pakety, kontroluje je a rozhoduje o tom, má-li být paket propuštěn nebo zachycen
- Lze provádět ověřování klienta v širokém rozsahu
  - Od IP zdrojové adresy
  - K přísným ověřovacím technikám typu výzva/odpověď
- Může být filtrován i protokol server/klient

Výhody

- Zvýšená bezpečnost
  - Předcházení problémům s bezpečností na IP úrovni
  - Má informaci o stavu spojení (filtrování na aplikační úrovni může být stavové)
  - Použití ověřovacích technik
  - Filtrace protokolu server/klient
- Umožňuje rozšířené logování a účtování
- V zásadě méně složitá pravidla filtrování, méně náchylná k chybám, jednodušší ovládání
- Interní DNS nemusí spolupracovat s externím DNS
- Je možné rozšířit proxy o cache – zachycování často požadovaných dat

Nevýhody

- Méně transparentní a přizpůsobivé
  - Klient si může proxy uvědomit
  - Podpora jednoduchých klientů nebo proxy klientů
  - Omezený počet proxy, pro každý protokol musí existovat proxy
  - Vyžaduje vyhrazený počítač

Principy filtrování na úrovni spojení

- V zásadě vypadá jako filtrování na aplikační úrovni
  - Generické proxy na úrovni spojení pracuje na obranném valu
  - Klienti se spojují s proxy na úrovni spojových služeb (TCP)
  - Proxy ověřuje klienty na úrovni tohoto spojení
  - Spojení mezi proxy a serverem je pak transparentní
- Od filtrování na aplikační úrovni se liší
  - Slabším ověřováním
  - Nezajišťuje filtrování protokolu na úrovni aplikace klient/server
- Realizace
  - Realizace vyžaduje zásah do programového vybavení klienta
  - Systémové volání na nižší úrovni v klientu nahrazeno spojkami (connect)
  - Spojka spojuje klienta a spojované proxy
  - S použitím speciálního protokolu posílá adresu a port cílového počítače.

Vlastnosti

- Bezpečnost mezi IP úrovní a aplikační úrovní

- Transparentnost mezi IP úrovní a aplikační úrovní
- Zahrnuje logování a účtování
- Programové vybavení klienta musí být přizpůsobeno

Výhody

- Doplnění programu o spojku je jednodušší než zavedení proxy
- Je však nutné mít zdrojový kód, knihovny, …

Socks

- Představuje programové vybavení spojky pro realizaci proxy na úrovni spojení
- Navrženo pro aplikace typu klient/server
- Klient naváže spojení se socks, přenese adresu cíle, port cíle, typ spojení a identitu uživatele
- Socks vytvoří vlastní komunikační kanál, kterým posílá data klienta do serveru
- Během vytváření spojení lze provádět doplňkové funkce (ověřování, vyjednávání o bezpečnosti, …)

Model socks

- Zahrnuje 3 základní operace
  - Požadavek na spojení
  - Nastavení proxy spojení
  - Přepínání aplikačních dat
  - Ověřování

Typy obranných valů

Filtrující směrovač (Screening Router)

- Provádí filtraci paketů podle směru přenosu, IP adresy a čísla portu

Opevněný počítač (Bastion Host)

- Používá se při realizaci důležitých serverů, které mají být navíc velmi bezpečné. Např. SMTP, FTP, DNS, HTTP, atd.

Brána se dvěma vstupy (Dual Homed Gateway)

- Úplně odděluje vnitřní a vnější síť. Služby musí být umístěny na této bráně a jsou přístupné jak z vnitřní sítě, tak i z vnější sítě.

Screened Host Gateway

- Vnitřní síť je chráněna filtrujícím směrovačem, který propouští pouze pakety určené pro vybraný počítač (Bastion Host). Pakty mohou být filtrovány nejen podle IP adresy, ale i podle portu (přístup k určitým službám).

Screened Subnet

- Pomocí dvou filtrujících směrovačů se vytvoří oblast mezi vnitřní a vnější sítí, nazývaná demilitarizovaná zóna. Do této subsítě se připojí Bastion Hosts, nesoucí služby, které mají být přístupné jak z vnější, tak i z vnitřní sítě. Filtrujícími směrovači

lze dosáhnout toho, že pakety s vnějšími adresami nejsou přenášeny do vnitřní sítě a naopak pakety s adresami vnitřní sítě nejsou přenášeny do sítě vnější.

Brána aplikační úrovně

- Pomocí filtrujícího směrovače jsou propouštěny pouze pakety určené aplikační bráně. Zde jsou instalovány aplikační proxy, které umožní komunikaci a klienty ve vnitřní síti.

Útoky typu „Denial of Services“

Útoky Denial of Service

- Jeden z mnoha základních forem útoků na vnitřní sítě
  - Založen na přetížení systému
  - Výsledkem je omezení výkonnosti serveru nebo úplný výpadek cílového systému
  - Útok může být zaměřen na síťové komponenty nebo na hostitelské systémy
- Další z mnoha základních forem útoku jsou
  - Skenování portů
  - Přetečení bufferů
  - Prolomení hesla

Přetížení systému

- Cílem DoS je ztlumit skutečný provoz „odpadním“ provozem
  - Dochází k vytěsňování reálných přenosů
    - Klienti na základě detekce zahlcení zpomalují vysílání
    - Směrovače musí přebytečné pakety odstraňovat
- Zahozené pakety vedou k exponenciálnímu nárůstu času opakování
- Směrovače jsou přetíženy
- Servery se mohou přetížit zvyšováním počtu požadavků na vytvoření spojení
  - Vytváření TCP spojení vyžaduje zapamatování stavu a odezvu serveru
  - Server požaduje odpověď na SYN od klientů
  - Klienti ale neodpovídají na výzvu serveru

IP spoofing (navádění k nepravostem)

- Navádění systému, aby vložil odlišné zdrojové IP adresy do IP záhlaví
  - DoS útočníci navádějí ze dvou důvodů
    - Nechtějí být odhaleni
    - Spoofing může přidat další zatížení
- Jestliže navádíte s cizími ale legitimními IP adresami
  - Může být spuštěn Reset buď z napadeného počítače, nebo z počítače s použitou IP adresou
    - Okamžité uvolnění zdrojů serveru
  - Pečlivý výběr sekvenčních čísel na straně serveru může ukončit pokus o navázání spojení
- Jestliže navádíte s náhodně vybraným IP
  - Odpověď serveru na klientské SYN se ztratí
  - Server uvolní zdroje ale typicky až za 75 sekund

Klíčové prvky DoS útoku

Převedení těžiště činnosti na jiné uzly

Princip – co je jednoduché pro mě musí být složité pro tebe

Př.: IP spoofing

Já: generuji SYN pakety jak rychle to jen jde (mikrosekundy)

Ty: timeout odstraňuje SYN každých 75 sekund

Charakteristiky DoS útoku

- Musí být rozšířen na mnoho systémů
  - Typickým cílem je útočit z mnoha míst najednou
    - Umožňuje lepší využití síťových zdrojů
    - Pomáhá čelit preventivnímu měření
    - Pomáhá zatajit útočníka
- DoS software je lehce dostupný a lze jej jednoduše napsat
  - Mnohý lze najít v IRC
- DoS útoky jsou často předcházeny prolomením systému a instalací DoS programů
  - Dává o mnoho více anonymity útočníkovi

Usnadnění DoS útoků

- V počítačové síti běží mnoho systémů
- Počítačová síť je velmi rozlehlá
- Mnozí uživatelé jsou naivní – dávají šanci uchvátit vzdálený systém
- Protokoly internetu jsou známé, to vytváří podmínky pro využití jejich slabin
- Mnoho volného software, ve kterém mohou být zahrnuty utajené funkce
- Nedostatečná ochranná politika používání a managementu
- Velmi rozsáhlý software s mnoha známými děrami
- Nedostatek prostředků pro zastavení útoků

Chování se při DoS útoku

- Neuchovávej stav dokud nedostaneš od klienta ACK
  - DoS útočníci, používající spoofing neposílají ACK
    - Jinak by si museli zapamatovat stav
  - Užívej kryptografii abys předcházel ukládání stavu
    - Pošli klíč pro jedno použití jako odpověď serveru na SYN
    - Odpověď s ACK musí vrátit klíč
- Prostředky detekce proniknutí
  - Zachyť útok na firewallu, pokud jej rozpoznáš
  - Snort
- IP metody zpětného trasování

Snort (Open Source Intrusion Detection System)

- Systém pro detekci útoků (Intrusion Detection Systém)
- Je schopen provádět analýzu toku dat v reálném čase a logování paketů v IP sítích
- Může provádět analýzu protokolů, vyhledávání údajů
- Je schopen detekovat různé útoky a sondování
- Používá jazyk pro popis toku dat
- Obsahuje automat pro detekci podle tohoto popisu
- Umožňuje informovat o útoku v reálném čase (syslog, soubor, sockety, …)