**AAA – AVTENTIKACIJA**

**AVTENTIKACIJA Z GESLI**

|  |  |
| --- | --- |
| Možni napadi in obrambe | |
| Prisluškovanje (kraja v prenosu) | Uporaba varne šifrirane povezave |
| Vlom v sistem (kraja shranjenih gesel) | Varovanje Sistema / gesel |
| Ugibanje gesel | Število poskusov omejimo |

|  |  |
| --- | --- |
| Načini hranjenja gesel in problemi glede teh načinov | |
| Hranjenje gesla na vseh mestih, kjer potrebujemo | Velika ranljivost, problemi pri spreminjanju gesla |
| Hranimo na enem mestu in jih vsi uporabljajo | Zaščita prenosa kopije do uporabnika |
| Imamo posebno vozlišče, ki nudi storitev preverjanja | Poseben protokol |

**A ONE-TIME PASSWORD SYSTEM (RFC 2289)**

Sistemi enkratnega gesla (OTP) zagotavljajo mehanizem za prijavo v omrežje ali storitev s pomočjo posebnega gesla, ki je veljavno samo enkrat. S tem dosežemo, da če je geslo zaseženo, je lahko uporabljeno samo enkrat. Sodelujeta dve entiteti, strežnik in generator.

Potek:

* Strežnik OTP pošlje generatorju izziv (“challenge”), ki vsebuje seme (“seed”) in zaporedno številko. Seme je sestavljeno iz 1 do 16 alfanumeričnih znakov z malimi črkami. Izziv generatorju pove, kako pravilno izračunati OTP iz skrivne pass-kode.
* Generator združi uporabnikov secret pass-phrase ter seme (concat). Ta niz je nato poslan skozi varno hash funkcijo in zmanjšan na 64 bitov s pomočjo posebnih algoritmov.
* Prvo OTP geslo, ki ga bomo uporabili, je poslano N-krat skozi hash funkcijo. Število N poda uporabnik. Naslednje geslo bo poslano (N-1)-krat skozi hash funkcijo. Če je nekdo prisluškoval, ne bo mogel generirati naslednjega gesla, saj to pomeni, da bo moral obrniti hash funkcijo. Podprte hash funkcije so MD4, MD5 in SHA.
* Geslo, zmanjšano na 64 bitov, se pretvori v 4 besede, ki so izbrane iz slovarja dolgega 2048 besed.

Ker moramo podati število N, to pomeni, da bo enkrat geslo poteklo (pridemo do 0). Kako to narediti, je odvisno od implementacije. RFC ukazuje le, da implementacija MORA zagotoviti reinicializacijo brez, da uporabnik ogrozi svoje skrivno geslo.

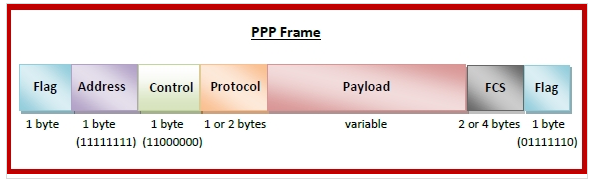
**PPP (POINT-TO-POINT) PROTOCOL (RFC 1661)**

Je komunikacijski protocol na povezavni plasti, ki ga uporabljamo za prenos podatkov med dvema neposredno povezanima napravama. Njegove glavne storitve so:

* Definicija okvirja (frame) podatkov - Definira procedure vzpostavljanja povezave
* Določi metodo enkapsulacije omrežne pl. - Določa pravila za avtentikacijo

Protokol je sestavljen iz treh komponent:

* Encapsulation component: enkapsulira datagram, da ga lahko prenesemo prek nekega prenosnega medija
* Link Control Protocol (LCP): odgovoren za vzpostavljanje, nastavljanje, testiranje, vzdrževanje in prekinitev povezav, izbira avtentikacijskega protokola, izbira hash funkcije pri CHAP...
* Authentication protocols (AP): avtenticirajo sodelujoče. Med te protokole spadajo PPP in CHAP



\*FCS uporablja CRC za preverjanje napak pri prenosu.

**PAP - Password Authentication Protocol (RFC 1334)**

Omogoča enostavno metodo za dokazovanje identitete. Zgodi se samo ob vzpostavitvi povezave. Ob vzpostavitvi vozlišče neprestano pošilja par (id, geslo) drugemu vozlišču, dokler ga to ne potrdi ali se povezava konča.

Gesla so poslana v čistem besedilu in ni zaščite proti ponovitvi komunikacije.

**CHAP – Challenge-Handshake Authentication Protocol (RFC 1994)**

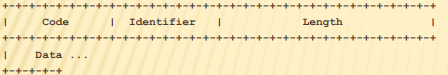
* Po vzopostavitvi povezave, avtentikator pošlje izziv
* Odjemalec odgovori s vrednostjo (geslo × izziv), izračunano s pomočjo enosmerne razpršilne (hash) funkcije
* Avtentikator primerja dobljeno vrednost s svojo izračunano vrednostjo
* Koraki se lahko ponavljajo ob naključnih intervalih

Prednosti:

* Zaščita pred playback napadom (veča se identifier in izziv se spreminja)
* Odvisen od skrivnosti, ki jo poznata samo odjemalec in avtentikator. Skrivnost ni nikoli poslana po mreži
* Lahko isto skrivnost uporabimo za avtentikacijo obeh smeri

Slabosti:

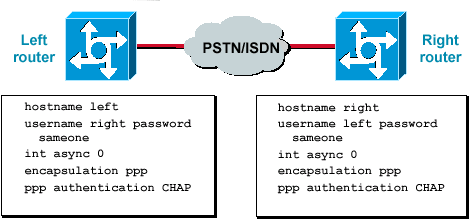
* Geslo mora biti shranjeno v plaintext obliki (kriptirana gesla se ne morejo uporabljati)
* Manj uporaben v velikih omrežjih

**code** Koda sporočila

1 Challenge, 2 Response,

3 Success, 4 Failure

**Identifier** povezovanje med koraki protokola



**MS-CHAP – Microsoft PPP CHAP Extensions (RFC 2759)**

Microsoftova različica CHAP protokola. Uporabljen kot ena od avtentikacij v PPTP (Point-To-Point-Tunneling protocol) – Protokol za implementacijo VPNjev z veliko znanimi napakami.

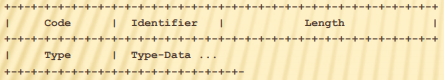
Zasnovan je na CHAP protokolu z dvema bistvenima dodatkoma: - Vzajemna avtentikacija - Možnost spreminjanja gesla

Razlika med MS-CHAPv1 in MS-CHAPv2 je, da različica v2 ponuja obojestransko avtentikacijo tako, da v paketu Response ene strani nosi tudi izziv za drugo stran, enako tudi v Succes paketu nosi Authenticator response.

**EAP – Extensible Authentication Protocol (RFC 5247)**

Koračni protocol:

* Avtentikator pošlje zahtevo po podatkih (npr. Identifikacija, zahteva za avtentikacijo vključno z načinom)
* Odjemalec odgovori ali zavrne način avtentikacije
* Ta dva koraka se ponavljata dokler strežnik ne ugotovi identitete odjemalca

**Type** kaj zahteva avtentikator in kaj odgovarja odjemalec

1 Identity 2 Notification 3 NAK 4 MD5-Challenge

5 OTP 6 Generic token card

**AAA – AVTORIZACIJA**

**DOSTOPOVNA MATRIKA**

Dostopovna matrika (Access Matrix) določa, katere pravice ima posamezna skupina uporabnikov

* Seznam zmožnosti (capability list) - Seznam pravic dostopa (Access Control List)

Hrani se lahko lokalno v datotekah, ali pa na strežniku.

**AAA – BELEŽENJE**

Uporabljamo za beleženje dogodkov (kje in kdaj so se zgodili).

Obstajajo različne stopnje sporočil: Emergency, Alert, Critical, Error, Warning, Notice, Info in Debug.

**SYSLOG PROTOCOL (RFC 5424 & 5425)**

Syslog je standard, uporabljen za shranjevanje stanja nekega sporočila od pošiljatelja do prejemnika. Vsako sporočilo dobi posebno številko, ki nam pove, katera aplikacija je ustvarila sporočilo.

Nevarnosti Syslog sporočila:

* Syslog sporočila navadno ne potujejo neposredno od pošiljatelja do prejemnika, ampak preko posrednikov, torej jih ne moremo nadzorovati
* Neavtoriziran pošiljatelj lahko poizkuša zavesti avtoriziranega prejemnika, da mu odgovori
* Napadalec poskuša odpreti sporočilo
* Napadalec zbriše sporočilo ali zamenja vrstni red

TLS uporablja certificate za avtentikacijo ter kriptira sporočila.

Varnost:

* Običajno gre vsako sporočilo preko varne TSL povezave, kjer sta tako pošiljatelj kot prejemnik zaupanja vredna, ker sta prestala avtentikacijo
* Možno je tudi, da vsebina sporočila ni pomembna in prejemnik ne zaheva avtentikacije pošiljatelja
* Prav tako se dogaja, da pošiljatelj ne zahteva avtentikacije prejemnika; pomeni da Syslog sporočilo, lahko dobi vsakV okoljih, kjer varnost sploh ni vprašanje npr. zaprta omrežja, pa tako pošiljatelj kot prejemnik ne gresta skozi avtentikacijski procesVsi zgoraj našteti primeri pošiljanja(razen prvega) se odsvetujejo, če obstaja kakršnakoli varnostna grožnja

Pomanjkljivosti: Če je povezava prekinjena zaradi neznanega razloga ali pa jo prekine prejemnik, pošiljatelj ne ve katera sporočila so prispela uspešno in katera ne.

**RADIUS – Remote Authentication Dial-In User Service (RFC 2865 & 2866)**

**OSNOVE**

* RADIUS je široko razširjen protokol. Omogoča centralizirano upravljanje avtentikacije, avtorizacije in upravljanje uporabnikov pri priklopu na omrežje in uporabi omrežnih storitev
* Za avtentikacijo lahko uporablja tudi druge protokole
* Na prenosni plasti uporablja UDP transportni protokol

**UPORABNIK <-> ODJEMALEC <-> STREŽNIK**

* RADIUS odjemalec je Network Access Server (NAS). Odjemalec je odgovoren za posredovanje uporabnikovih informacij do RADIUS strežnikov ter za pravilno ravnanje glede na strežnikov odgovor
* Primeri NAS so Remote Access Server, VPN Strežnik, omrežno stikalo
* Komunikacija med uporabnikom in NAS odjemalcem najpogosteje poteka nepsoredno na povezavni plasti (ppp / ethernet). Avtentikacija lahko poteka prek PAP, CHAP, MS-CHAP, EAP
* RADIUS strežnik je odgovoren za sprejem zahtev za povezavo, avtentikacijo uporabnika in vračanje vseh konfiguracijskih podatkov, ki so potrebni, da odjemalec uporabniku omogoči uporabo storitve, avtorizacijo odjemalcev ter spremljanje dejavnosti uporabnikov

**VARNOST**

* Transakcije med odjemalcem in strežnikom uporabljajo za avtentikacijo skupno skrivnost, katera se ne pošilja preko omrežja
* Uporabniška gesla, ki se prenašajo med odjemalcem in strežnikom so kriptirana
* Za avtentikacijo lahko podpira različne metode (PAP, CHAP...)
* Ni zaščite pred prisluškovanjem
* Je (delna) zaščita pri verodostojnosti poslanih paketov
* Ni zaščite pred zanikanjem poslane vsebine

**POTEK PROTOKOLA**

1. Uporabnik odjemalcu predstavi svoje avtentikacijske podatke
   1. Npr. preko forme uporabniško ime in geslo, ali uporabi protokol, kot je PPP (v avtentikacijskih paketih to prenaša)
2. Odjemalec dobi uporabniške podatke in se lahko odloči za RADIUS avtentikacijo
   1. Preko omrežja pošlje strežniku paket **Access-Request** (uporabniško ime, geslo (MD5), ID uporabnika, ID vrat)
   2. Če strežnik ne odgovori, lahko pošlje ponovno, ali pa pošlje sekundarnim strežnikom v primeru nedostopnosti)
3. Strežnik prejme zahtevo
   1. V primeru, da nima s tem odjemalcem vzpostavljene skupne skrivnosti, paket tiho zavrže
   2. **Če je skupna skrivnost napačna, odgovora ne bo**
   3. Strežnik opravi validacijo (primerja podatke iz zahteve s podatki v podatkovni bazi, kjer so shranjene zahteve, ki morajo biti izpolnjene za odobritev dostopa – geslo, vrata..)
   4. Lahko odgovori s **Access-Reject**, **Access-Challenge**, **Access-Accept**
4. Če so vsi pogoji zadovoljeni, strežnik pošlje **Access-Accept**

* V primeru **Access-Challenge**
  + Izziv je lahko npr. dodatno geslo, PIN koda ali kaj drugega
  + Odjemalec uporabniku posreduje izziv ter zahteva odgovor
  + Po prejemu odziva pošlje nov **Access-Request** paket, kjer geslo zamenja z odzivom uporabnika in poda “**State**” atribut iz **Access-Challenge paketa**
  + Odziv je kriptiran
  + Izzovov je lahko več
* V primeru **Access-Reject**
  + Napačno geslo / uporabniško ime...
  + Neustrezne pravice
  + Dodatno pojasnilo lahko doda strežnik v sporočilo
* V primeru **Access-Accept**
  + Sporočilo lahko prinaša dodatne podatke, ki jih NAS potrebuje za vzpostavitev storitve (IP naslov, L2TP tunel itd.)

**ZAKAJ UDP?**

* Če zahteva na primarni avtentikacijski strežnik spodleti, mora biti posredovana naslednjemu sekundarnemu strežniku
* Časovne zahteve so zahtevnejše, kot jih zagotavlja TCP
* Brezstanjski način tega protokola poenostavlja uporabo UDP
* UDP poenostavlja strežniško implementacijo
* Poenostavljeno delovanje medstrežnikov, ker nimajo odprtih povezav
* Pri vsem tem UDP potrebuje eno stvar, ki je tipična za TCP. Gre namreč za časomere ponovnih pošiljanj istemu strežniku, ki pa ne potrebujejo takšne pozornosti, kot jo je potrebno nameniti časomerom TCPja

**MEDSTREŽNIK IN PODROČJA**

* Uporabniki so razdeljeni na sfere/področja. Vsako področje ima svoj RADIUS strežnik
* Ponudnik storitve lahko preko RADIUS strežnika dovoli gostovanje uporabnikov iz drugih domen v svojem področju

**BELEŽENJE**

* Poznamo paketa **Accounting-Request** in **Accounting-Response**
* Beležimo lahko tri vrste dogodkov (začetek rabe storitve, nadaljnjo rabo ali popravljene podatke ter zaključek rabe)

**PODPISOVANJE AA. PAKETOV**

* Odjemalec: 128 bitno naključno število – izziv
* AA. odgovor: MD5(Code × ID × Length × RequestAuth × Attributes × Secret)

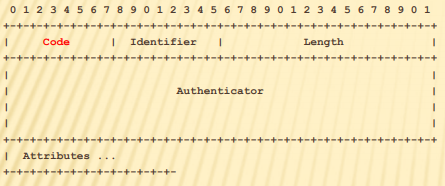
A picture containing text

Description automatically generated

**PODPISOVANJE ..A PAKETOV**

* Odjemalec: MD5(Code × ID × Length × 0016 × Attributes × Secret)
* Strežnik odgovori: MD5(Code × ID × Length × RequestAuth × Attributes × Secret)
* Diagram

  Description automatically generatedPodpis ščiti odjemalčevo zahtevo

**CODE** Koda ukaza

1 – Access Request, 2 – Access Accept ...

**IDENTIF.** Ker je RADIUS koračni protokol, mora odjemalec vedeti na katero zahtevo prejema odgovor

**LENGTH** Dolžina celotnega paketa vključno z glavo v zlogih,

dolžina je med 20 in 4096 B (daljši se skrajša)

**PRILASTKI**

* 256 različnih možnosti, zahteva je da uporabnik mora imeti možnost dodati svoje prilastke
* TLV notacija, zato lahko poljubno razširimo protokol brez spremembe implemetacije samega protokola
* User-Name, User-Password, CHAP-Password
  + Geslo se šifrira z uporabo izziva v avtentikatorju (RA) in skupne skrivnosti (S)
  + Geslo razdelimo v 128-bitne dele p[1...n]
    - b[1] = MD5(S×RA); c[1] = p[1] XOR b[1]
    - b[i] = MD5(S×c[i-1]; c[i] = p[i] XOR b[i]

**DIAMETER (RFC 3588)**

Je naslednik protokola RADIUS. Osnovna ideja DIAMETER-a je zagotoviti osnoven protokol, ki je razširljiv in zagotavlja storitve AAA (Avtentikacija, Avtorizacija, Beleženje) tehnologijam , ki omogočajo dostop do omrežja.

**OSNOVA**

* TCP na transportni plasti
* Definira format sporočil, transport, poročanje o napakah in varnostne storitve
* Vsako vozlišče lahko sproži zahtevo - je P2P protokol
* **AVP** Vsebujejo podatke za avtentikacijo, avtorizacijo, beleženje, varnost in usmerjanje

**RAZLIKA RADIUS / DIAMETER**

* Varnejši prenosni protokol (TCP, STCP)
* Vgrajena omrežna varnost (SSL, IPsec)
* Možnih več prilastkov (32-bitni)

**VARNOSTNI ELEMENTI**

**IP Security Protocol – IPSEC (RFC 2411)**

**VARNOST NA OMREŽNI PLASTI**

* Zakrivanje vseh vrst podatkov (TCP segment, UDP segment, ICMP sporočilo, OSPF itd.)
* Zagotavljanje avtentikacije izvora
* Integriteta podatkov
* Zaščita pred ponovitvijo komunikacije

**MEHANIZEM IPSEC PONUJA DVA PROTOKOLA VAROVANJA**

* AH (Authentication Header): zagotavlja avtentikacijo izvora in integriteto podatkov
* ESP (Encryption Security Payload): zagotavlja avtentikacijo izvora, integriteto podatkov in zaupnost podatkov

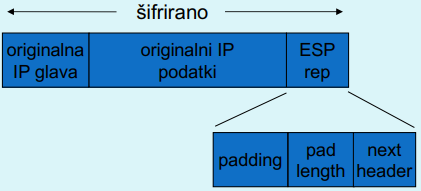
**SA**

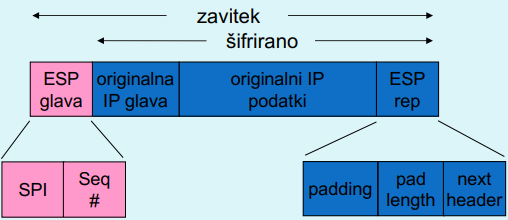
* Za vsako smer IPSec komunikacije je vzpostavljen SA ( Security Association )
* Usmerjevalnik ima bazo SAD ( SA Database), ki vsebuje vnose za SA:
  + 32 bitni ID SA => SPI ( Security Parameter Index)
  + Izvorni in ponorni IP SA
  + Vrsta enkripcije (npr. 3DES) in ključ
  + Vrsta preverjanja integritete (hash funkcije – MD4, MD5, SHA...)
  + Ključ za avtentikacijo

**DVA NAČINA KOMUNIKACIJE**

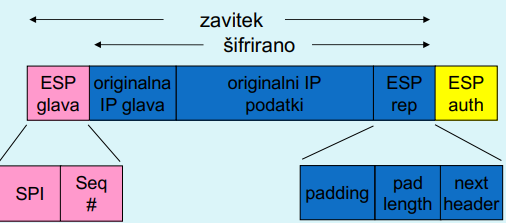
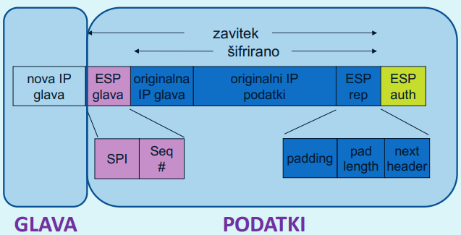
* Transport mode
  + Implementiran med končnimi odjemalci (vmesniki računalnikov), vmesnikom transparentno
  + Ščiti samo zgornje plasti – kriptirani so samo podatki v paketu
* Tunnel mode
  + Transparentno končnim odjemalcem (vmesnikom računalnikov)
  + Usmerjevalnik – Usmerjevalnik ali Usmerjevalnik – Uporabnik
  + Šifrira podatke in glavo paketa
  + Za odjemalce ni nujno, da izvajajo IPSec

**NAJBOLJ POGOSTO UPORABLJEN NAČIN**

* Najbolj pogosto uporabljen način komunikacije IPSec je TUNNEL MODE + ESP



* SPI je indeks SA, ki se ga uporabi za določanje nastavitev
* Seq# je zaščita proti ponovitvi komunikacije
* Rezultat je zavitek (enchilada)
* Next header je protokol, vsebovan v podatkih
* Zapolnitev je potrebna za bločno šifriranje
* Rezultat se šifrira ( algoritem določa SA )



Protocol = 50 (ESP)

IP pošiljatelja in prejemnika sta vozlišči, med katerima poteka IPsec

**PREJEMNIK**

Iz glave poišče podatke o SA, preveri MAC zavitka, preveri Seq#, odstrani zapolnitev, izloči podatke, posreduje računalniku

Izdela se nov IP paket

Dodano polje ESP Auth, ki je izračunana zgoščena vrednost celotnega zavitka. Algoritem in ključ določa SA.

**IKE – Internet Key Exchange**

* Je protokol za izmenjavo ključev preko interneta
* Uporablja PKI ali PSK za avtentikacijo odjemalcev med seboj
* **Izmenjava poteka s pomočjo Diffie-Hellman**
* Potek:
  + Faza 1: Vzpostavi dvosmeren IKE SA (init in auth)
    - IKE SA je poseben SA, ki se uporablja **samo za izmenjavo ključev**
    - Vzpostavi se ključ za varovanje nadaljne komunikacije
      * Avtentikacija se izvede s PSK, PKI ali podpisom
    - Dva načina – aggressive mode (krajši, vendar razkrije identiteto odjemalcev) in main mode (daljši, skrije)
  + Faza 2: IKE generira ključe za druge storitve, kot je npr. IPSec

**SSL – Secure Socket Layer**

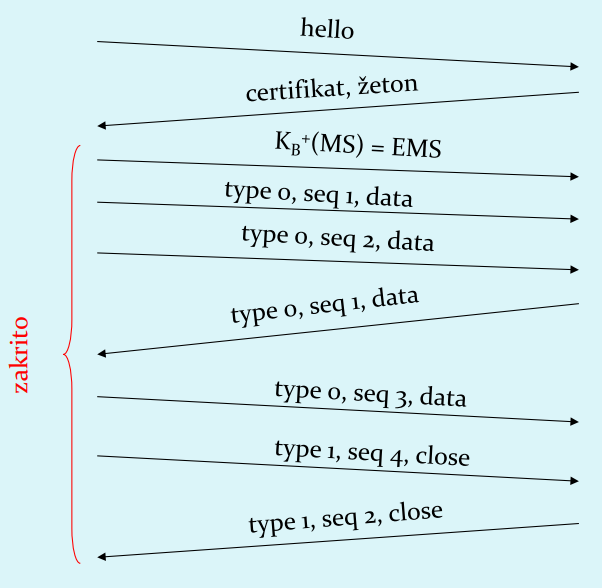
Je široko uporabljen varnostni protokol, ki zagotavlja zaupnost, integriteto in avtentikacijo.

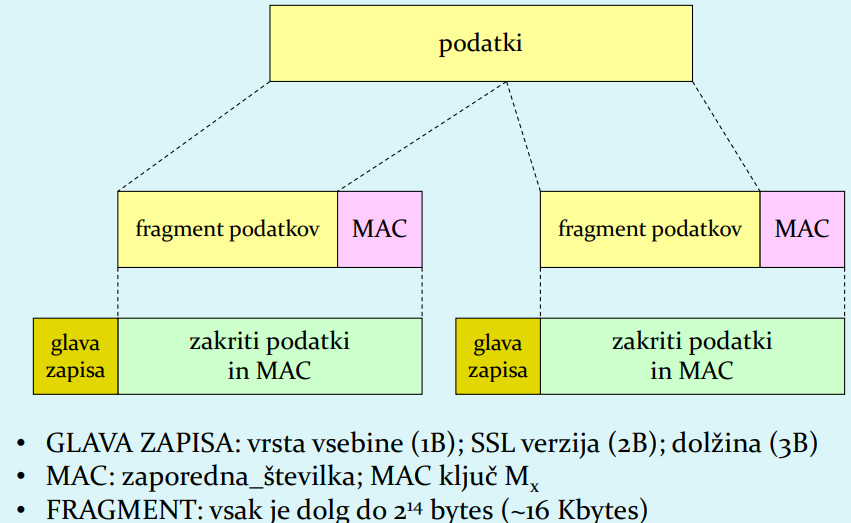
**GLAVNI CILJI PRI RAZVOJU**

* Uporaba pri spletnih transakcijah
* Zakrivanje podatkov
* Avtentikacija spletnih strežnikov
* Možna avtentikacija odjemalcev
* Čim manjši napor pri nakupu pri drugem prodajalcu

**INTEGRITETA PODATKOV, ŠTEVILKA PAKETA V TCP GLAVI**

* Tok podatkov razbijemo v ZAPISE
  + Vsakemu zapisu pripnnemo x
  + **PROBLEM 1:** Ker je številka paketa TCP nešifrirana v glavi TCP, to povzroča ranljivost v primeru MITM napada
    - Številko paketa upoštevamo pri računanju MAC
      * MAC = MAC(Ključ Mx, zaporednaStevilka || podatki)
  + **PROBLEM 2:** Napadalec predčasno zaključi sejo
    - V vsak zapis uvedemo poseben “tip zapisa”, ki pove, ali gre za podatke/zaključek
      * To dodamo v MAC
        + MAC = MAC(Ključ Mx, zaporednaStevilka || tip || podatki)

**POENOSTAVLJENI SSL**

****

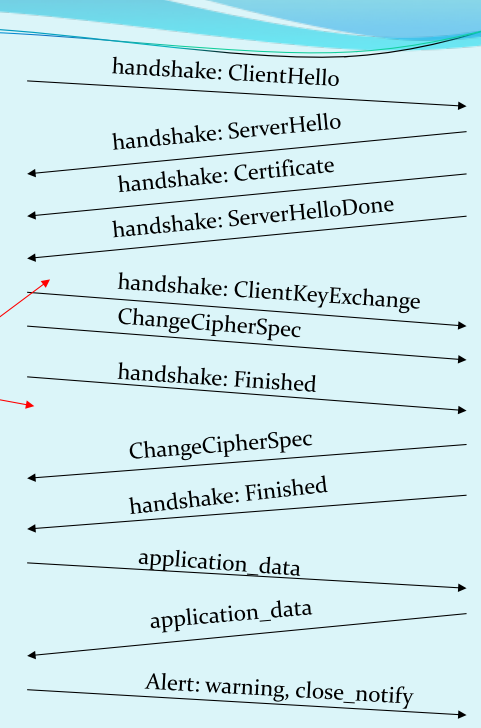
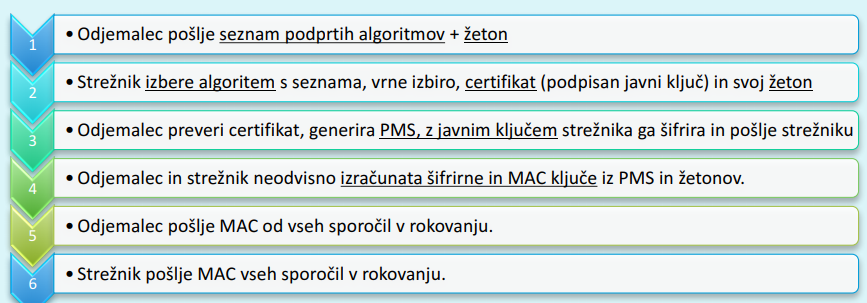
**PRAVI SSL ROKOVANJE**

Zakaj uporaba žetonov?

* Za vsako komunikacijo je uporabljen drug žeton, da onemogočimo ponavljanje komunikacije (replay attack)

Zakaj izmenjava MAC v korakih 5 in 6?

* Odjemalec ponudi več algoritmov za šifriranje, nekateri so šibki, nekateri pa močnejši
* Napadalec bi lahko izbrisal močnejše algoritme
* Zadnji dve sporočili zagotavljata integriteto vseh prenešenih sporočil

****

**OPERATIVNA VARNOST: Požarne pregrade & sistemi za zaznavanje vdorov**

**FIREWALL (Požarna pregrada)**

**VRSTE FILTRIRANJ**

* Osami notranje omrežje od velikega javnega omrežja

**BREZSTANJSKO FILTRIRANJE PAKETOV**

* Filtriranje na omrežni plasti
* Filtriranje običanjo izvaja že usmerjevalnik
* Na podlagi vsebine paketov se odloča, ali bo posredoval posamezen paket (IP, protokol v IP, vrata, sporočila ICMP)
* ACL (Access Control List): seznam vnosov (pogoj, akcija) = pravila za požarno pregrado
* Določenim paketom dovoli prehod, ostale zaustavi
* Ima 3 naloge:
  + Filtrira VES promet
  + Prepušča samo promet, ki je dopusten glede na politiko
  + Je imun na napade

**SISTEMI ZA ZAZNAVANJE VDOROV (IDS)**

* Je dodatna naprava, ki izvaja poglobljeno analizo paketov
* Na podlagi vstopa sumljivih paketov v omrežje lahko naprava

prepreči njihov vstop ali razpošlje obvestila

**STANJSKO FILTRIRANJE PAKETOV**

* Upošteva povezavo in njeno trenutno stanje
* Spremlja in vodi evidence o stanju vsake vzpostavljene TCP povezave
* Zabeleži vzpostavitev povezave (SYN) in konec (FIN), na podlagi tega zavrže nesmiselne pakete
* Preverja veljavnost povezave, ki lahko poteče
  + IPS filtrira sumljiv promet, IDS pa pošlje sporočilo

**KAKO DELUJE IPS/IDS ?**

* **Primerjava s shranjenimi vzorci napadov (signatures)**

**×** Vzorci napadov lahko hranijo izvorni IP, ponorni IP, protokol,

zaporedje bitov v podatkih paketa

**×** Varnost je torej odvisna od baze znanih vzorcev, še nevidene napade

**APLIKACIJSKI PREHODI**

* Omogočajo dodatno filtriranje glede na izbiro uporabnikov, ki lahko uporabljajo določeno storitev
* Omogočajo filtriranje na podlagi podatkov na aplikacijskem nivoju poleg IP/TCP/UDP
* **SLABOSTI**
  + V primeru, da imamo več aplikacij, potrebuje vsaka svoj aplikacijski prehod
  + Odjemalce je potrebno nastaviti, da se znajo povezati s prehodom

slabo zaznava

**×** Zahtevno procesiranje

* **Zaznavanje netipičnega prometa**

**×** Sistem opazuje običajen promet in izračuna statistike, vezane nanj

**×** Reagira na statistično neobičajen promet (npr. ogromno ICMP pak.)

**×** Možno zaznavanje še nevidenih podatkov

**×** Težko ločevanje med normalnim in nenavadnim prometom

**NAPADI IN GROŽNJE**

Najpogostejše oblike napadov:

* Napadi s spreminjanjem informacij (modification attack) - Zanikanje komunikacije (repudiation attack)
* Odpoved delovanja Sistema (denial of service attack) - Nepooblaščen dostop (access attack)

**POGOSTI NAPADI**

|  |  |
| --- | --- |
| **Pregledovanje Sistema (reconnaissance)** | Napadalec s različnimi tehnikami poskuša odkriti arhitekturo Sistema, storitve v njem itd. To pomaga pripraviti napad na sistem |
| **Prisluškovanje (eavesdropping)** | Prestrezanje omrežnega prometa, prisotno zlasti pri brezžičnih omrežjih |
| **Ugibanje gesel** | Groba sila, napad s slovarjem |
| **Virusi, črvi, trojanci** |  |
| **Izkoriščanje šibkosti v programski opremi** |  |
| **Socialni inžinering** | Poskus pridobivanja zaupanja žrtve |
| **Pregled vrat (port scan)** | Napadalec testira, Kateri strežniki so delujoči (npr. ping) in katere storitve ponujajo. Napadalec lahko pridobiva podatke o sistemu: DNS, storitve, OS) |
| **Brskanje po smeteh (dumpster diving)** | Način, s katerim napadalci lahko pridejo do informacij o sistemu (navodila za uporabo, seznami gesel, telefonskih številk, organizacija dela) |
| **Matematični napadi** | Na kriptografske algoritme in ključe |
| **Rojstnodnevni napad (birthday attack)** | Je napad na zgoščevalne funkcije, za katere zahtevamo, da nobeni dve sporočili ne generirata iste zgoščene vrednosti. Pri slabših funkcijah napadalec išče sporočilo, ki bo dalo isto zgoščeno vrednost |
| **Zadnja vrata (back door)** | Napadalec zaobide varnostne kontrole in dostopi do Sistema preko druge poti |
| **Ponarejanje IP naslovov (IP Spoofing)** | Napadalec prepriča ciljni sistem, da je nekdo drug (Poznan) s spreminjanjem paketov |
| **Prestrezanje komunikacije (man in the middle)** | Napadalec prestreže komunikacijo in se obnaša, kot da je ciljni sistem (pri uporabi certifikatov lahko žrtev uporablja tudi javni ključ od napadalca) |
| **Ponovitev komunikacije (replay attack)** | Napadalec prestreže in shrani stara sporočila ter jih ponovno pošlje kasneje, predstavljajoč se kot eden izmed udeležencev |
| **Ugrabitev TCP sej** | Napadalec prekine komunikacijo med uporabnikoma in se vrine v mesto enega od njiju, drugi verjame, da še vedno komunicira s pravim |
| **Napad s fragmentacijo (fragmentation attack)** | Z razbijanjem paketa na fragmente razdelimo glavo paketa med fragmente tako, da jih požarna pregrada ne more filtrirati (tiny fragment – deli glavo prvega paketa, overlapping – napačen offset prepiše prejšnje pakete) |

**POGOSTI NAPADI – DoS**

**DoS (Denial of Service)**

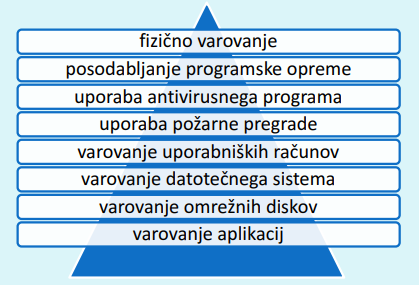
Cilj napadalca je obremeniti omrežne vire tako, da se nehajo odzivati zahtevam regularnih uporabnikov (npr. vzpostavitev velikega števila povezav, zasedanje diskovnih kapacitet...)

**DDoS (Distributed Denial of Service)**

DoS napad, ki ga povzroči napadalec z več omrežnih sistemov naenkrat.

|  |  |
| --- | --- |
| **Prekoračitev medpomnilnika (buffer overflow)** | Procesu pošljemo več podatkov, kot jih lahko sprejme (ping of death – icmp s več kot 65K podatkov je povzročil sesutje Sistema) |
| **SYN napad** | Napadalec pošlje veliko število zahtev za vzpostavitev povezave, na odgovor Sistema ne odzove, pride do preobremenitve vrste zahtev v sistemu (rešitev je timeout in omejitev števila vrat) |
| **Napad Teardrop** | Napadalec spremeni podatke o številu in dolžini fragmentov v IP paketu, kar zmede prejemnika |
| **Napad Smurf** | Uporaba posrednega oddajanja za preobremenitev sistema |

|  |  |
| --- | --- |
| **Varovanje aplikacij** | * Pravilna namestitev aplikacij * Odstranitev odvečnih aplikacij * Onemogočanje priponk v e-pošti * Onemogočanje izvajanja nevarnih tipov datotek * Nameščanje v nestandardna vrata in mape |

**OBRAMBA PRED NAPADI**

|  |  |
| --- | --- |
| **Fizično varovanje** | Omejimo fizičen dostop do strežnikov in računalnikov (zaklepanje računalnikov, nastavitev gesel za zagon, gesel za dostop do BIOS, onemogočimo zagon s zunanjih medijev) |
| **Posodabljanje apl.** | Proizvajalec izdaja popravke za varnostne luknje |
| **Uporaba AV / požarne pregrade** | Namestitev na odjemalcu / strežniku, priporočljivo je na odjemalcu, ker škodljiva oprema začenja delovati tam (npr. aplikacijski prehodi) |
| **Varovanje uporabniških računov** | Napadalci iščejo neuporabljane, neaktivne, nezaščitene račune za dostop do Sistema. Rešitev:   * Preimenovanje uporabniških imen administratorjev * Omejitev števila računov z visokimi pravicami * Onemogočanje uporabe starih računov * Uporaba zahtevnih gesel |
| **Varovanje datotečnega / omrežnega Sistema** | * Za dostop do datotečnega Sistema dodeli uporabnikom najmanjše potrebne pravice * Odstranitev nepotrebnih aplikacij * Zaščita zagonskega področja |

**PODATKI ZA DELOVANJE OMREŽJA**

**Imeniška storitev** je omrežna storitev, ki identificira vse vire na omrežju in jih naredi dostopne uporabnikom in aplikacijam. Ti viri so lahko e-poštni naslovi, računalniki in druge naprave, npr. tiskalniki. Imeniška storitev naj bi naredila topologijo omrežja ter protokole transparentne, tako da uporabnik na omrežju lahko dostopa vsak vir, brez, da bi vedel kje in kako je shranjen.

**PRILASTKI, SHEMA**

V mapi so združeni posamezni **prilastki**. **Mape** vsebujejo prilastke različnih tipov (imeniška struktura je hierarhična). **Prilastki** za študenta so npr. vpisna številka, ime, priimek, geslo, uporabniško ime. Nekateri prilastki so obvezni, nekateri dovoljeni.

Struktura imenikov in prilastkov v njih definira **shemo**.

**PREDMETNI & IMENSKI PROSTOR**

**Shema** združuje različne predmete in prilastke.

**Imenski prostor** je shema v katerem se nahajajo objekti (predmeti oz. entries)

* Predmeti so dejanske vrednosti, ki jih hrani imeniška struktura glede na definirano shemo.
* Predmeti, ki so vstavljeni v imenik, so v vsebovalniku (container)
* vsi predmeti v vsebovalniku so v istem imenskem prostoru (namespace, ki je podobna struktura kot slovar)
* predmeti v imenskem prostoru so ponovno hierarhično oblikovani
* predmete moramo med seboj razlikovati
* **za razlikovanja uporabimo pravila, ki določajo enolično in nedvoumno ime (ključ)**
* ključ je lahko sestavljen
* Predmeti živijo v imenskem prostoru in ne v vsebovalniku

**Razločevalno ime** je ime, po katerem razlikujemo predmete. Lahko je absolutno ali relativno – glede na hierarhijo imenikov. Običajno ni shranjeno v imeniški strukturi, ampak je definirano s pravili.

**UPRAVLJANJE S VSEBINO**

Vsebino imenskega prostora lahko:

* Porazdelimo med različne strežnike (distribution) – porazdeljena imeniška storitev
* Prepišemo še na drug strežnik (replication) – z vsebino imenskega prostora še vedno upravlja izvorni strežnik

**PODATKOVNE BAZE & IMENIŠKE STORITVE**

Običajna relacijska podatkovna baza je organizirana v tabelah – v imeniški strukturi pa imamo tudi prilastke, ki so:

* Obvezni (podobno kot db)
* Neobvezni (na nek način null vrednosti v db)
* Se lahko ponovijo
* Prilastki in njihova struktura so standardizirani (IANA)
* Predmeti so razvrščeni v imenske prostore, pri čemer posamezen predmet podeduje vse lastnosti starša

**Standard X.500**

**Je standard za imeniške storitve v omrežju.**

Sestoji iz 4 protokolov:

* Protokol za dostop do imeniške strukture – operacije nad strukturo: Bind, Read, List, Search, Compare, Modify, Add, Delete in ModifyRDN.

Standard definira imenski prostor in v njem se nahajajo predmeti

Vsak predmet je določen s svojim razločevalnim imenom

Predmet ima lahko eno ali več (tudi ponavljajujočih) vrednosti prilastkov

Imeniška struktura sestoji iz enega samega imenika (posamezne dele imenika poslužujejo različni strežniki).

**LDAP – Lightweight Directory Access Protocol (RFC 4510 – 4519)**

Je protokol, ki omogoča vsem da najdejo podatke o organizacijah, posameznikih in ostalih virih, kot so datoteke in naprave na omrežju. Je predvsem protokol za komunikacijo, ki upošteva metashemo shranjenih podatkov.

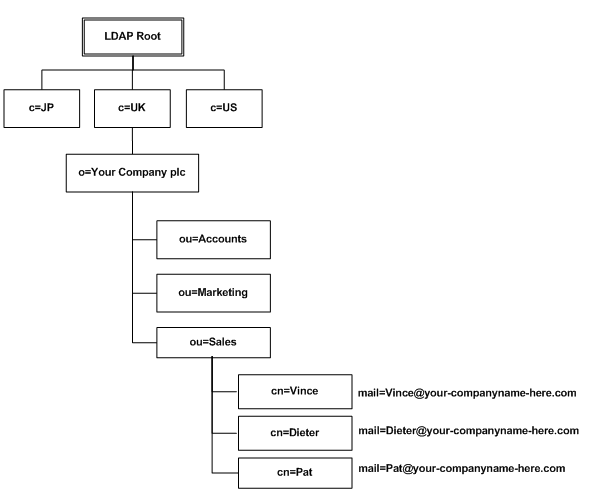
Najpogostejša raba LDAP je centralizirana avtentikacija – shranjuje uporabniška imena in gesla. LDAP je tako lahko uporabljen v različnih aplikacijah in storitvah, da preveri uporabnika.

**RAZLIKE MED LDAP v2 IN LDAP v3**

* LDAPv3 dovoljuje vrednosti prilastkov zakodirane z UTF-8
* LDAPv3 podpira porazdeljeno hranjenje podatkov domene tako, da strežnik lahko prenaslovi odjemalca k drugemu strežniku
* LDAPv3 uporablja močno avtentikacijo in varnovanje podatkov z uporabo SASL

**UKAZI**

|  |  |
| --- | --- |
| **Bind** | Želja po avtentikaciji ter ostalih možnih parametrih komunikacije. Seja je lahko tudi neavtenticirana |
| **Unbind** | Zaključek komunikacije (seje) |
| **Search** | Iskanje posameznih predmetov v bazi. Rezultat je lahko odvisen od tega, ali je odjemalec avtenticiran ali ne. |
| **Compare** | Možnost primerjave vrednosti predmeta. Ni potrebno razkriti prave vrednosti predmeta, samo preverjamo enakost |
| **Add** | Dodajanje predmeta v bazo |
| **Delete** | Pobrišemo predmet iz baze |
| **Modify** | Spremenimo vrednosti prilastkov predmeta |
| **Modify DN** | Spremenimo ime predmeta (rename) |
| **Abandon** | Prekinemo izvajanje zahteve, ki smo jo poslali (lahko prekinemo iskanje in primerjanje ter popravke baze) |
| **Extended** | Generična možnost poljubnega dodatnega ukaza |



Vsako vozlišče je lahko unikatno identificirano s svojim **DN (Distinguished name)**

**Shema** združuje različne predmete in prilastke.

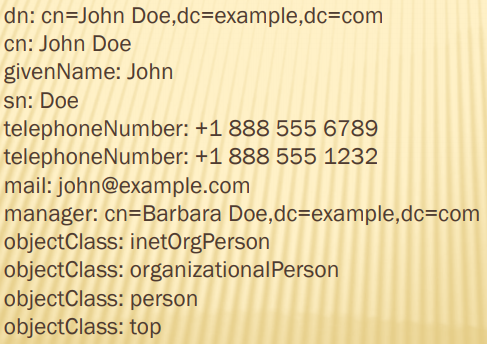
**Razredi** (ObjectClass) združujejo prilastke:

* Opisani s zapisom ASN.1
* So del hierarhije in dedujejo lastnosti starša
* Določajo obvezne in neobvezne prilastke

**Prilastki** opisujejo lastnosti:

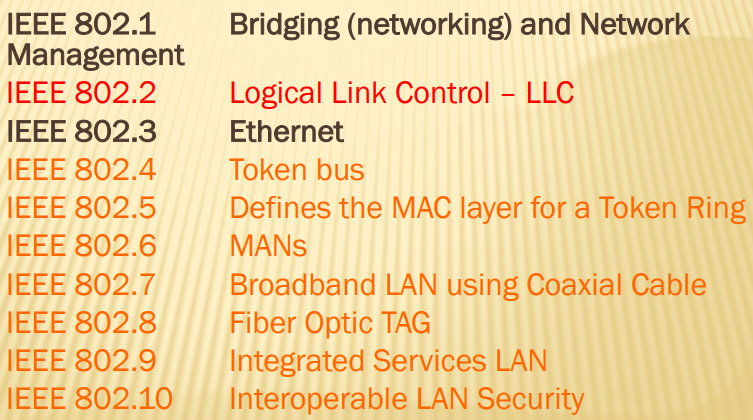
* Na nek način definicija tipa
* Njihovo udejanjenje (instanciacija) bo dejansko hranila vrednosti
* Opisujejo tudi sintakso, način primerjave ipd.

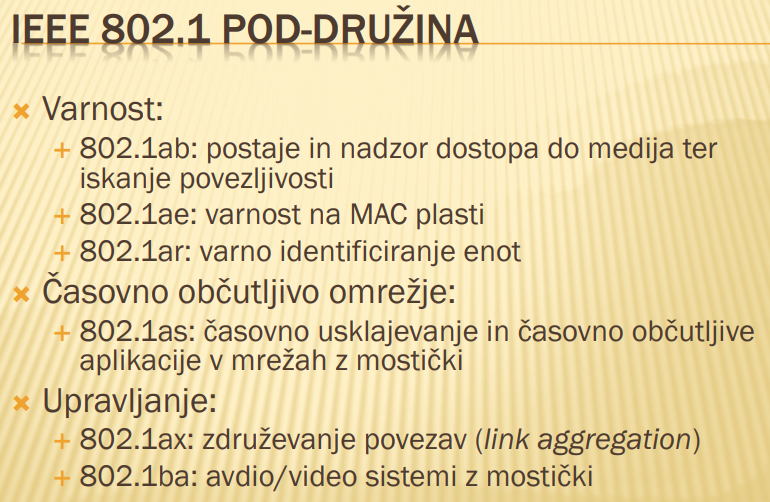
**LDIF** je format, ki ga uporabljamo pri prenosu podatkov med LDAP strežniki.



**IEEE 802**

Je družina standardov, ki opisujejo delovanje lokalnih (LAN) in mestnih (MAN) omrežij.



**** **IEEE 802.1 – PREMOŠČANJE & UPRAVLAJNJE OMREŽIJ**

* Bridging (networking) and Network Management
* Povezovanje med pod-mrežami
* Upravljanje omrežij
* Varnost v mrežah
* Deluje na vrhu LLC

**NADZOR PRIKLJUČITVE V MREŽO (IEEE 802.1X)**