Contents

[IPSec (IP Security) 3](#_Toc535430697)

[VPN – Virtual Private Network – Navidezna zasebna omrežja 3](#_Toc535430698)

[Implementacija IPSec 4](#_Toc535430699)

[Vzpostavitev SA 4](#_Toc535430700)

[2 Načina komunikacije 4](#_Toc535430701)

[IPSec datagram: Tunnel mode in ESP 4](#_Toc535430702)

[Kako izbrati datagrame za IPSec zaščito 5](#_Toc535430703)

[Kakšno zaščito ponuja IPSec 5](#_Toc535430704)

[Protokol IKE 5](#_Toc535430705)

[IKE Ima 2 fazi 6](#_Toc535430706)

[SSL – Secure sockets layer 6](#_Toc535430707)

[Zasnova SSL 6](#_Toc535430708)

[Poenostavljeni SSL 7](#_Toc535430709)

[Poenostavljeni SSL: Rokovanje 7](#_Toc535430710)

[Poenostavljeni SSL: Izpeljava ključa 7](#_Toc535430711)

[Poenostavljeni SSL: Pošiljanje podatkov 7](#_Toc535430712)

[Poenostavljeni SSL: Pošiljanje podatkov 8](#_Toc535430713)

[Pravi SSL: podrobnosti 8](#_Toc535430714)

[Pravi SSL: Rokovanje 8](#_Toc535430715)

[SSL: Pretvorba v zapise 9](#_Toc535430716)

[SSL: Izpeljava ključev 10](#_Toc535430717)

[Varnost v omrežju 10](#_Toc535430718)

[Požarna pregrada 10](#_Toc535430719)

[Požarna pregrada: vrste filtriranj 10](#_Toc535430720)

[Brezstranjsko filtriranje paketov 10](#_Toc535430721)

[Brezstranjsko filtriranje paketov: primeri 11](#_Toc535430722)

[Brezstanjsko filtriranje: Dostopovni seznami 11](#_Toc535430723)

[Stanjsko filtriranje paketov 11](#_Toc535430724)

[Aplikacijski prehodi 12](#_Toc535430725)

[Sistemi za zaznavanje vdorov 12](#_Toc535430726)

[Načini zaznavanja vdorov 12](#_Toc535430727)

[Primer IDS/IPS sistema 13](#_Toc535430728)

[Pogosti napadi na omrežne sisteme 13](#_Toc535430729)

[DoS – Denial-of-service 14](#_Toc535430730)

[Tehnike obrambe 15](#_Toc535430731)

[Fizično varovanje sistema 15](#_Toc535430732)

[Posodabljanje aplikacij 15](#_Toc535430733)

[Uporaba antivirusa / požarne pregrade 15](#_Toc535430734)

[Varovanje uporabniških računov 15](#_Toc535430735)

[Varovanje datotečnega / omrežnega sistema 15](#_Toc535430736)

[Varovanje aplikacij 16](#_Toc535430737)

[AAA 16](#_Toc535430738)

[Avtentikacija 16](#_Toc535430739)

[Avtentikacija z gesli 16](#_Toc535430740)

[Hranjenje gesel 16](#_Toc535430741)

[Napadi na gesla 16](#_Toc535430742)

[Naslov kot geslo 17](#_Toc535430743)

[Zaupanja vredni posredniki 17](#_Toc535430744)

[Avtentikacja ljudi 17](#_Toc535430745)

[PPP in PAP 18](#_Toc535430746)

[CHAP – Challenge-Handshake authentication protocol (MS-CHAP) 18](#_Toc535430747)

[Katera razpršilna funkcija 18](#_Toc535430748)

[CHAP – Oblika paketa 19](#_Toc535430749)

[MS-CHAP 19](#_Toc535430750)

[EAP – Extensible Authentication Protocol 19](#_Toc535430751)

[Avtorizacija 19](#_Toc535430752)

[Avtorizacija – dostopovna matrika 20](#_Toc535430753)

[Beleženje 20](#_Toc535430754)

[IETF 20](#_Toc535430755)

[Programska oprema 20](#_Toc535430756)

[Syslog protokol 21](#_Toc535430757)

[Protokol RADIUS – Remote Authentication Dial In User Service 22](#_Toc535430758)

[Komunikacija nas – RADIUS (..A) 23](#_Toc535430759)

[RADIUS – Beleženje 23](#_Toc535430760)

[Protokol RADIUS 23](#_Toc535430761)

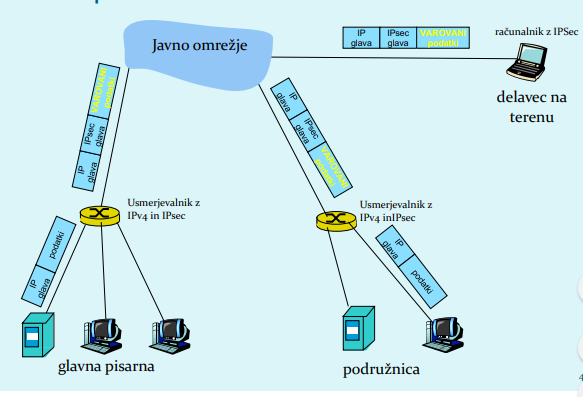
[DIAMETER 26](#_Toc535430762)

# IPSec (IP Security)

* IP Security protocol (Varnost na omr. Plasti)
* Zavarovanje povezav med 2 entitetama. Uporaba za VPN
* Varnost na omr. Plasti:
  + Zakrivanje vseh vrst podatkov (TCP, UDP, ICMP, OSPF...)
  + Zagotavljanje avtentikacije izvora
  + Integriteta podatkov pred spreminjanjem
  + Zaščita pred ponovitvijo komunikacije (Reply attack)

# VPN – Virtual Private Network – Navidezna zasebna omrežja

* Angl. Virtual Private Network
* Podjetja, ki so na različnih geograf. Lokacijah in si želijo visoke varnosti pri kom.
  + Gradnja **ZASEBNEGA** omrežja – ločeno od interneta (Drago)
  + **NAVIDEZNO ZASEBNO** omrežje (VPN) z infrastrukturo Javnega omrežja:
    - podatki znotraj lokalnih (zasebnih) delov omrežja se prenašajo tradicionalno (IP)
    - podatki, ki potujejo preko javnih delov omrežja se prenašajo zaščiteno (IPSEC)

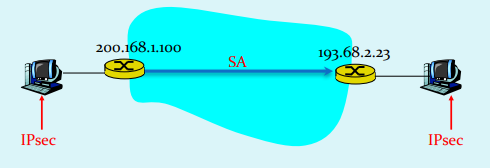


# Implementacija IPSec

* Ponuja dva protokola varovanja
  + **AH** – Authentication Header (Zagotavlja avtentikacijo izvora in integriteto podatkov)
  + **ESP** – Encapsulation Security Payload (Polek zgornjega zagotavlja še zaupnost podat.)
* Za vsako smer IPSec kom. Je potrebno vzpoztaiti **SA** (Security Associtation)

# Vzpostavitev SA

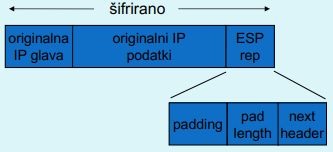
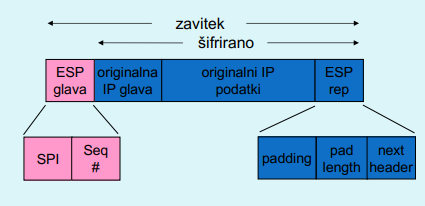
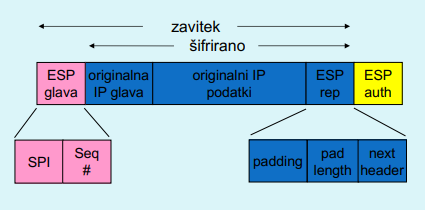
* Usmerjevalnik ima bazo SAD (Security Association Database), kjer hrani podatke o SA:
  + 32 bitni ID SA, Imenovan **SPI** (Security Parameter Index)
  + Izvorni in ponorni IP **SA**
  + Vrsta enkripcije in ključ
  + Vrsta preverjanja integritete (MD5, SHA1..)
  + Ključ za avtentikacijo



# 2 Načina komunikacije

* **Transport mode –** Impmementiran na vmesnikih računalnikov, ščiti zgornje plasti protokola, kriptira samo podatke v paketu
* **Tunnel mode –** Usmerjevalnik – Usmerjevalnik ali Usmerjevalnik – Uporabnik. Šifrira podatke in glavo paketa. Torej IPSec lahko izvede že usmerjevalnik in pošlje uporabniku dešifrirano
* Torej vse možnosti:
  + Transport mode z AH
  + Transport mode z ESP
  + Tunnel mode z AH
  + **Tunnel mode z ESP –** Najpogosteje

# IPSec datagram: Tunnel mode in ESP

* Najbolj pogosto uporabljen način
* Originalni podatki: [ Originalna IP Glava | Originalni IP Podatki ]
* Na konec datagrama se doda ESP glava (Padding je potreben za bločno šifriranje, next header je protokol, vsebovan v podatkih)
* Rezultat se šifrira (Algoritem in ključ določi SA)
* Doda se ESP glava: rezultat je »enchilada« (**zavitek**)
  + **SPI** – Indeks SA, ki se uporabi za določanje nastavitev
  + **Seq#** - Zaščita proty repl. Att.
* Doda se polje **ESP Auth**, ki je izračunana zgoščena vrednost celega zavitka. Algoritem in ključ določa Sađ
* Izdela se nova IP Glava, ki se doda pred podatke
* Oblikuje se nov IP paket, ki se klasično pošlje skozi omrežje.
* V novi glavi paketa
  + **Protokol = 50** (Pomeni, da so podatki ESP)
  + IP Pošiljatelja in prejemnika sta vozlišči, med katerima poteka IPSec (Usmerjevalnika)
* Prejemnik:
  + Iz **SPI** v glavi poišče podatke o SA, preveri MAC zavihka,
  + Preveri Seq#
  + Odšifrira zavihek
  + Odstrani zapolnitev
  + Izloči podatke
  + Posreduje cilnjemu računalniku

# Kako izbrati datagrame za IPSec zaščito

* To določa **SPD (Security Policy Database)** 
  + Določa, kateri SA naj se uporabi
* SPD določa **KAJ** narediti z datagramom
* SAD določa **KAKO** to narediti

# Kakšno zaščito ponuja IPSec

* Denimo, da je Cefizelj naš man-in-the-middle med R1 in R2. Ne pozna ključev. Kaj lahko naredi?
  + Ali lahko vidi vsebino datagrama, izvor, ponor, protokol, port? **Ne, vse je zašifrirano**
  + Ali lahko spremeni bite v paketu? **Da, vendar z tem uniči šifriranje, paket se smatra kot invalid**
  + Ali lahko pošilja v imenu R1? **Da**
  + Ali lahko ponovi komunikacijo? **Lahko, a zaradi Seq# nima efekta**

# Protokol IKE

* Internet Key Exchange, protokol za iuzmenjavo ključev preko interneta.
* **Pri IPSec je potrebno vzpostaviti SA med odjemalci npr:** 
  + Primer vzpoztavljenega SA ->
* Ročno določanje SA je nepraktičnp in zamudno: potrebno ga je določiti za vsako smer komunikacije in vsak par odjemalcev!
* **Rešitev**: Uporabimo protokol **IPsec IKE**

# IKE Ima 2 fazi

* Uporablja **PKI** ali **PSK (pre-shared key)** za avtentikacijo odjemalcev med seboj. Ima dve fazi:
  + Vzpoztavi dvosmeren **IKE SA (INIT in AUTH)**
    - IKE SA je ločen SA od IPSec SA, ki se uporablja samo za izmenjavo ključev (Imenuje se tudi **ISAKMP SA**)
    - V IKE SA se vzpoztavi ključ za zavarovanje nadaljnje komunikacije glede izmenjave ključev (Avtentikacija se izvede z PSK, PKI ali podpisom)
    - Dva načina: **Aggressive mode (Krajši, vendar razkrije identiteto odjemalcev)** in **Main mode (daljši, skrije identiteto)**
  + IKE generira ključe za druge storitve, kot je npr IPSec. Vzpozravi se torej IPSec SA (**CREATE\_CHILD in INFO**)
    - Edini način: Quick Mode

# SSL – Secure sockets layer

* Široko uporabljen varnostni protokol
  + Podprt v skoraj vseh brskalnikih in na vsek strežnikih (**HTTPS**)
  + Z uporabo SSL se opravi za 10 milijard dolarjev nakupov letno
* Razvil ga je Netscape leta 1993
* Več vrst
  + **TLS:** Transport layer security
* Zagotavlja: **Zaupnost, integritetom avtentikacijo**
* Cilji pri razvoju:
  + Uporaba pri spletnih transakcijah
  + Zakrivanje podatkov (Še posebej številk kred. Kartic)
  + Avtentikacija spletnih strežnikov
  + Možnost avtentikacije odjemalca
  + Čim manjši napor pri opravljanju nakupa pri drugem prodajalcu
* Dostopen je vsem TCP aplikacijam preko aplikacijskega vmesnika SSL
  + [Application | TCP | IP]
  + [Application | SSL | TCP | IP]

# Zasnova SSL

* Lahko bi ga zasnovali na osnovi kriptografije **PKI (Šifriranje z javnim ključem prejemnika, zasebnim ključom pošiljatelja, upraba zgočevalnik funkcij),** vendar..
  + Želimo pošiljati **TOKOVE BYTOV** in interaktivne podatke, ne sporočila – **Povezani način prenosa**
  + Za eno povezavo želimo imeti **MNOŽICO** ključev, ki se spreminjajo
  + Kljub temu želimo uporabljati certifikate – avtentikacija
    - Ideja: **uporabimo jih pri rokovanju**

# Poenostavljeni SSL

* Vsebuje 4 faze:

1. **Rokovanje:** Ana in Brane uporabita certifikate, da se avtenticirata eden drugemu in izmenjata glavni ključ
2. **Izpeljava ključa:** Ana in brane uporabita izmenjani glavni ključ, da izpeljeta množico ključev
3. **Prenos podatkov:** Podatki, ki se prenašajo so združeni v **ZAPISE**
4. **Zaključek povezave:** Za varen zaključek povezave se uporabijo posebna sporočila

# Poenostavljeni SSL: Rokovanje

* **MS** = Glavni ključ (**master secret**)
* **EMS** = šifrirani glavni ključ (**encrypted master secret)**
* **KB+** = Branetov javni ključ

# Poenostavljeni SSL: Izpeljava ključa

* Slaba praksa je **uporabljati isti ključ za več šifrirnih opracij**.
* Uporabimo poseben ključ za zakrivanje in posebnega za preverjanje integritete (**MAC**)
* Uporabimo torej 4 ključe:
  + Kc – Ključ za **zakrivanje** podatkov, poslanih od **odjemalca** strežniku
  + Mc – Ključ za **zgoščevanje** podatkov, poslanih od **odjemalca** strežniku
  + Ks – Ključ za **zakrivanje** podatkov, poslanih od **strežnika** odjemalcu
  + Ms – Ključ za **zgoščevanje** podatkov, poslanih od **strežnika** odjemalcuž
* Ključi se izpeljejo z uporabo posebne funkcije. Ta uporablja **glavni ključ (Master Secret)** in **dodatne (naključne)** podatke za generiranje naslednjih ključev

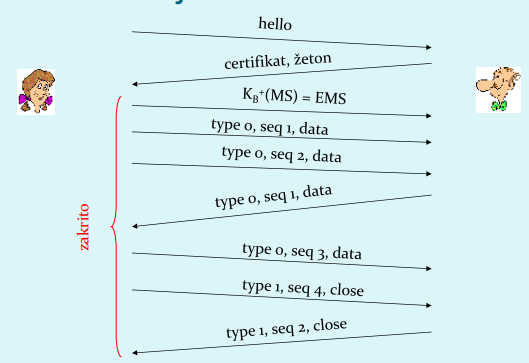
# Poenostavljeni SSL: Pošiljanje podatkov

* Kako preveriti integriteto podatkov
  + Če bi pošiljali po zlogih (byte-ih), kam bi pripeli **MAC (Zgoščena vrednost sporočila)**
  + Tudi če **MAC** pošljemo po zaključku celega prenosa (vseh zlogov), nimamo vmesnega preverjanja integritete!
* **Rešitev:** Tok podatkov razbijemo v **ZAPISE**
  + Vsakemu zapisu pripnemo x
  + Prejemnik lahko reagira na (ne)veljavnost integritete posameznega zapisa
* **Problem 1:** Številka paketa se nahaja v nešifrirana v glavi TCP, Kaj lahko naredi napadalec?
  + Zajame in ponovi komunikacijo
  + Preštevilči vrsti red paketov
  + Prestreže in odstrani paket
* **Rešitev:** Pri računanju MAC upoštevaj številko paketa
  + MAC = MAC(**Ključ Mx, zaporedna\_št, || podatki)**
  + Nimamo ločene št. Paketa
  + Zaščita proti replay attack: **Uporabi enkratni žeton.**

# Poenostavljeni SSL: Pošiljanje podatkov

* **Problem 2:** Napadalec predčasno zaključi sejo
  + Ena ali obe strani dobita vtis, da je podatkov manj, kot jih je
* **Rešitev:** Uvedimo poseben »tip zapisa«, ki nosi posebno vrednost, če gre za **zaključni paket**
  + Npr: 0 pomeni podatke, 1 pomeni zaključek
  + Uporabimo vrednost pri izračunu MAC
    - MAC = MAC(Ključ Mx, zaporedna\_št || tip || podatki)





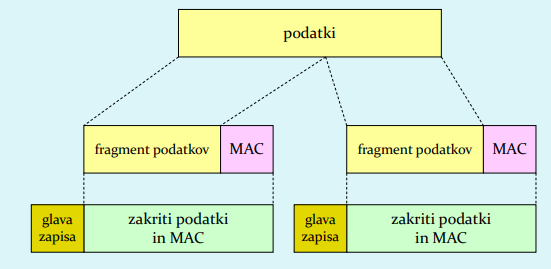
# Pravi SSL: podrobnosti

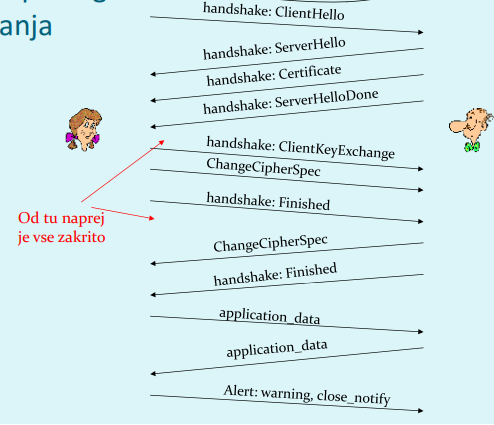
* Dogovor o uporabi protokola:
  + Odjemalec strežniku **ponudi** šifrirne algoritme, strežnik pa **izbere**
* Najpogostejši simetrični algoritmi
  + **DES – Data encryption standard: block**
  + **3DES – Triple strength: block**
  + **RC2 – Rivest Cipher 2: block**
  + **RC4 – Rivest Chipter 4: stream**
* Najpogostejši algoritem za **PKI** šifriranje
  + **RSA**

# Pravi SSL: Rokovanje

* **Poenostavljeni SSL:** Hello ->, <- certifikat, šifriran MS ->
* **Pravi SSL dejansko izvaja:**  avtentikacijo strežnika, izbiro algoritmov, določanje ključev, avtentikacijo odjemalca (opcijsko)
* **Postopek:**
  1. Odjemalec pošlje seunam podprtih algoritmov + **žeton**
  2. Strežnik **izbere algoritem** s seznama, vrne izbiro, **certifikat** (podpisan javni ključ) in svoj **žeton**
  3. Odjemalec preveri certifikat, generira **PMS (Master Secret), z javnim ključem** strežnika ga šifrira in pošlje strežniku
  4. Odjemalec in strežnik neodvisno **izračunata šifrirne in MAC ključe**  iz PMS in žetonov
  5. Odjemalec pošlje MAC od vseh sporočil v rokovanju
  6. Strežnik pošlje MAC od vseh sporočil v rokovanju
* **Zakaj izmenjava MAC v korakih 5 in 6?**
  + Odjemalec običajno ponudi več algoritmov, nekateri so šibki, drugi močnejši. Napadalec bi lahko iz ponudbe izbrisal močnejše algoritme.
  + Zadnji dve sporočili zagotavljata integriteto vseh prenešenih sporočil in preprečita tak napad
* **Zakaj uporaba žetonov?**
  + Denimo, da cefizelj posluša sporočila med Ano in Branetom ter jih shrani. Naslednji dan pošlje Cefizelj Branetu popolnoma enaka sporočila, kot jih je prešjni dan poslala Ana:
    - Če ima Brane trgovino, bo mislil da Ana ponovno naroča artikle
    - Brane za vsako komunikacijo uporabi drug žeton, tako Cefizelj me bo mogel ponoviti iste kom.

# SSL: Pretvorba v zapise



* **GLAVA ZAPISA:** vrsta vsebina (1B); SSL verzija (2B), dolžina (3B)
* **MAC:** zaporedna\_št, MAC kljuc Mx
* **FRAGMENT:** vsak je dolg 214 bytes ~ (16Kbytes)

# SSL: Izpeljava ključev

* Žetona odjemalca in strežnika ter PMS se uporabijo v funkciji, ki izračunava psevdo-naključna števila. Dobimo **MS (Master secret)**
* MS in **novi žetoni** se vstavijo v drugi naključni generator, dobimo **BLOK.** BLOK se razreže na 6 delov, da se dobi:
  + MAC Ključ odjemalca
  + MAC ključ strežnika
  + Enkripcijski ključ odjemalca
  + Enkripcijski ključ strežnika
  + **Inicializacijski vektor (IV) odjemalca \***
  + **Inicializacijski vektor (IV) strežnika \***
  + **\* -** Potrebna sta samo, kadar uporabljamo simetrični algoritem u bločno kriptografijo (3DES ali AES), ki potrebuje inicializacijo!

# Varnost v omrežju

* Administrator omrežja lahko uporabnike deli na:
  + **Dobri** (good guys): uporabniki, ki legitimno uporabljajo vire omrežja, pripadajo organizaciji
  + **Slabi** (bad guys): vsi ostali, njihove dostope moramo skrbno nadzorovati
* Omrežje ima običajno eno samo točko vstopa, na njej nadzorujemo dostope
  + Požarna pregrada (**firewall**)
  + Sistem za zaznavanje vdorov **(IDS, intrusion detection system)**
  + Sistem za preprečevanje vdorov **(IPS, intrusion prevention system)**

# Požarna pregrada

* Osami notranje omrežje od velikega javnega omrežka, določenim paketom dovoli prehod, druge zaustavi. Ima 3 naloge:
  + Filtrira VES promet
  + Prepušča samo promet, ki je **DOPUSTEN** glede na politiko
  + Je **IMUN** na napade

# Požarna pregrada: vrste filtriranj

* Brezstanjsko filtriranje paketov (angl. **Stateless traditional**) – Filtriranje na omrežni plasti
* Stanjsko filtriranje paketov (**Stateful filter**) – Filtriranje na prenosni plasti
* Aplikacijski prehodi (**Application gateways**) – Filtriranje na apk. Plasti

# Brezstranjsko filtriranje paketov

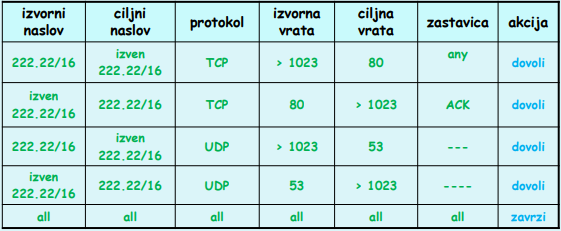
* Filtriranje običajno izvaja že **»usmerjevalnik«, ki meji na javno omrežje.** Na podlagi vsebine paketov se odloča, ali bo posredoval **posamezen paket**, odločitev na podlagi:
  + IP izvornega/ponornega naslova
  + Številke IP protokola: TCP, UDP, ICMP, OSPF ...
  + TCP/UDP izvori in cilji vrat
  + Tip sporočila ICMP
  + **TCP SYN** (Vzpoztavitev povezave) in **ACK bits** (ACK=1 velja za prvi segment pri povezovanju)

# Brezstranjsko filtriranje paketov: primeri



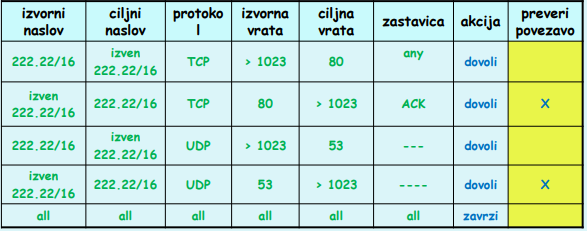
# Brezstanjsko filtriranje: Dostopovni seznami

* Dostopovni seznam (**ACL, access control list**)
* Tabela pravil, upošteva se jo od zgoraj navzdol
* Zapisi so par: (**pogoj, akcija**)
* **Primer**: onemogoči ves promet razen WWW navzven in DNS v obe smeri



# Stanjsko filtriranje paketov

* **Stateful filter,** upošteva povezavo in njeno trenutno stanje
  + Izoliranje filtriranje lahko dovoli vstop nesmiselnim paketom (npr. Vrata = 80, ACK = 1; čeprav noranji odjemalec ni vzpoztavil povezave)
* **Izboljšava: stanjsko filtriranje paketov** spremlja in vodi evidenco o stanju vsake vzopoztavljene TCP povezave
  + Zabeleži vzpostavitev povezave (SYN) in njen konec (FIN): na tej podlagi odloči, ali so paketi smiselni
  + Po preteku določenega časa obravnavaj povezavo kot neveljavno (**timeout**)
  + Uporabljaj podoben dostopovni seznam, ki določa, kdaj je potrebno kontrolirati veljavnost povezave (**check connection**)



# Aplikacijski prehodi

* Omogočajo **dodatno filtriranje** glede na izbiro uporabnikov, ki lahko uporabljajo določeno storitev
* Omogočajo filtriranje na podlagi podatkov na **aplikacijskem nivoju** poleg polj IP/TCP/UDP
* Tudi aplikacijski prehodi imajo omejitve:
  + Če uporabnik **potrebuje več aplikacij** (telnet, HTTP, FTP itd.), potrebuje **vsaka aplikacija svoj apk. Prehod**
  + Odjemalce je potrebno nastaviti, da se znajo **povezati s prehodnom** (npr. IP naslov medstrežnika v brskalniku)

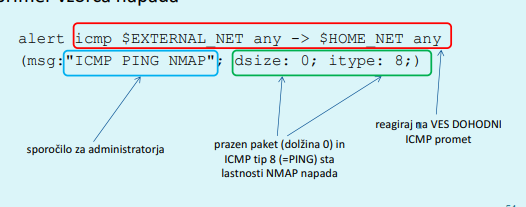
# Sistemi za zaznavanje vdorov

* Požarna pregrada kot filter paketov filtrira samo na podlagi glave **IP, TCP, UDP in ICMP**, to ne omogoča zazavanja vseh napadov
  + DoS, črvi, virusi, napadi na OS, aplikacije..
* Dodatna naprava – **IDS,** ki izvaja **pogljobeno analizo paketov.** Na podlagi vstopa sumljivih paketov jim lahko naprava vstop prepreči ali pa razpošlje obvestila
  + Sistem za zaznavanje vdorov (**IDS**) pošlje sporočilo o potencialno škodljivem prometu
  + Sistem za preprečevanje vdorov (**IPS**) filtrira sumljiv promet
  + Cisco, CheckPoint, Snort IDS
* V omrežju imamo lahko **več IDS/IPS naprav** (koristno zaradi **zahtevnega primerjanja vsebin paketov s shranjenimi vzorci**)

# Načini zaznavanja vdorov

* Orimerjava s shranjenimi vzorci napadov (**signatures)**
  + Vzorci napadov lahko hranijo izvorni IP, ponorni IP, protokol, zaporednje bitov v podatkih paketa, lahko so vezani na serijo paketov
  + Varnost je torej odvisna od **baze znanih vzorcev**; IDS/IPS slabo zaznava še nevidene napade
  + Možni **lažni alarmi**
  + Zahtevno procesiranje (lahko spregleda napad)
* Opazovanje netipičnega prometa (**anomaly-based**)
  + Sistem opazuje običajen promet in izračuna statistike, vezane nanj
  + Sistem reagira na statistično neobičajen promet (npr. Nenadno velik delež ICMP paketov (DOS))
  + Možno zaznavanje še nevidenih napadov
  + Težko ločevanje med normalnim in nenavadnim prometom

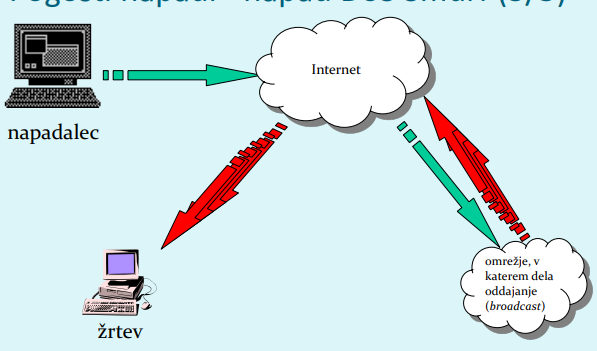
# Primer IDS/IPS sistema

1. Snort IDS
   * Public-domain, odprtokodni IDS za Linux, UNIX, Windows
   * Primer vzorca napada

# Pogosti napadi na omrežne sisteme

1. Ogrožanje zaupnosti, integritete in razpoložljivosti omrežnih sistemov
2. Napad s spreminjanjem informacij (**moification attack**)
3. Zanikanje komunikacije (**repudation attack**)
4. Odpoved delovanja sistema (**denial-of-service attack**)
5. Nepooblaščen dostop (**access attack**)
6. **Pregledovanje sistema** (reconnaissance): napadalec z različnimi tehnikami poskuša odkriti arhitekturo sistema, storitve v njem itd.
   * pomaga propraviti napad na sistem
   * primer (**war-dialing**) – napadalec s klicanjem na naključne telefonske št. Poskuša odkriti klicno številko modema za dostop do omrežja
7. **Prisluškovanje** (eavesdropping): prestrezanje omrežnega prometa, prisotno zlasti pri brezžičnih omrežjih (napadalec pridobi gesla, številke kreditnih kartic, ...)
   * **Pasivni in aktivni napadalec**
8. **Ugibanje gesel** (Groba sila (brute force), napad s slovarjem)
9. **Virusi, črvi, trojanci**
10. **Izkoriščanje šibkosti v programski opremi**
11. **Socialni inžiniring**
12. **Pregled vrat** (port scan): napadalec testira, kateri strežniki so delujoči (npr. Ping) in katere storitve ponujajo. Napadalec lahko pridobiva podatke o sistemu (DNS, storitve, op. Sistemi)
13. **Brskanje po smeteh** (dumpster diving): način, s katerim lahko napadalci pridejo do informacij o sistemu (navodila za uporabo, seznam gesel, telefonskih št., organizacija dela)
14. **Matematični napadi** na kriptografske algoritme in ključe
15. **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**.
16. **Zadnja vrata** (back door). Napadalec zaobide varnodstne kontrole in dostopi do sistema preko druge poti.
17. **Ponarejanje IP nasovov** (IP Spoofing): napadalec prepriča ciljni sistem, da je nekdo drug (poznan) s spreminjanjem paketov
18. **Prestrezanje komunikacije** (man-in-the-middle): napadalec prestreže komunikacijo in se obnaša, kot da je ciljni sistem (pri tem uporabi certifikatov lahko žrtev uporablja tudi javni ključ napadalca)
19. **Ponovitev komunikacije** (replay): napadalec prestreže in shrani stara sporočila ter jih ponovno pošlje kasneje, predstavljajoč se kot eden izmed udeležencev
20. **Ugravitev TCP sej** (TCP hijacking): napadalec preikne komunikacijo med uporabnikoma in se vrine v mesto enega od niju; drugi verjame, da še vedno komunicira z prvim.
21. **Napadi s fragmentacijo** (Fragmentation attack): z razbijanjem paketa na fragmente razdelimo glavo paketa med fragmente tako, da jih požarna pregrada nemore filtrirati
    * **Tiny fragment attack:** deli glavo prvega paketa
    * **Overlapping frag.** **att.:** napačen offset prepiše prejšne pakete

# DoS – Denial-of-service

* **Preprečitev delovanja sistema**
* Cilj napada: obremeni omrežne vire tako, da se nehajo odzivati zahtevam regularnih uporabnikov.
* DdoS (distributed): DoS napad, ki ga povzročji napadalec z **več omrežnih sistemov** na enkrat
* Uporabniki porazdeljenih omrežnih sistemov lahko da ne vedo, da je napadalna oprema nameščena pri njih.
* **Primeri**
  + **Prekoračitev medpomnilnika** (buffer overflow): procesu pošljemo več podatkov, kot lahko sprejme (**Ping of death**: ICMP z več kot 65K podatkov)
  + **SYN napad:** napadalec pošlje veliko število zahtev za vzpoztavitev povezave in se na odgovor sistema ne odzove; pride do preobremenitve vrste zahtev v sistemu
    - **Rešitev:** omejitev odprtih povezav, timeout
  + **Napad Teardrop:** napadalec spremeni podatke o št. In dolžini fragmentov v IP paketu, kar zmede prejemnika
  + **Napad Smurf:** uporaba posrednega dodajanja za preobremenitev sistema
* Uporaba botov (**web roBOT**) za organizacijo napadov na ciljni sistem
  + Boti so lahko računalniki, okuženi z trojanskimi konji
  + Njihovi uporabniki običajno ne vedo, da sodelujejo v napadu
* Udeleženci: **napadalec,** osrednji računalnik za **krmilnjenje botov** (herder), **boti** (zombie), cilj

# Tehnike obrambe

* V omrežju zadošča le **en šibek člen** – najšibkejši uporabnik, ki ogrozi omrežje.
* Admin **mora** preprečiti prenos škodljivih programov na delovne postaje uporabnikov in zapreti varnostne luknje v infrastrukture (konfiguracija)

# Fizično varovanje sistema

* Omejimo fizičen dostop do strežnikov in računalnikov
  + Zaklepanje
  + Nastavi geslo za zagon(CMOS/BIOS)
  + Geslo do BIOS
  + Onemogoči boot z zunanjih medijev

# Posodabljanje aplikacij

* Posodabljamo programsko opremo (**patching**), s čimer proizvajalec omogoči porpavljanje varnostnih lukenj
  + Admin potrebuje načrt testiranja, uvajanja in namestitve popravkov

# Uporaba antivirusa / požarne pregrade

* Uporaba antivirusnih programov
  + Več možnosti: namestitev na **odjemalcu/strežniku,** avtomatsko posodabljanje, zaščita v **realnem času,** priporočeno namestiti na odjelamcu, saj virusi delujejo tam.
  + Posodablanje (**posamezno, centralizirano**)
* Uporaba požarne pregrade v omrežju / osebne požarne pregrade

# Varovanje uporabniških računov

* Napadalci iščeju neuporabljene, neaktivne, nezaščitene račune za dostop do sistema:
  + Preimenovanje uporabniških imen administratorja
  + Omejitev št. Računov z visokimi pravicami (Ločeni admin računi, pogosta menjava gesel)
  + Onemogočenje uporabe starih računov
  + Uporaba zahtevnih gesel

# Varovanje datotečnega / omrežnega sistema

* Zaščita datotečnega sistema
  + Za dostop do dat. Sistema dodeli uporabnikom najmanjše potrebne pravice
  + Odstranitev nepotrebnih aplikacij
  + Zaščita zagona področja

# Varovanje aplikacij

* Pravilna namestitev aplikacij (privzete vrednosti niso vedno najbolj varne!)
* Odstranitev ovečnih apk.
* Onemogočanje priponk v e-pošti
* Onemogočanje izvjanja nevarnih tipov datotek
* Nameščanje aplikacij na nestandardna vrata in v nestandardne mape

# AAA

* **Authentication – Avtentikacija:** s kom se pravzaprav pogovarjamo
* **Authorization – Avtorizacija:** Ali ima klient pravico do vira/storitve..
* **Accounting – Beleženje:** kdo je uporabil kdaj kakšen vir/storitev

# Avtentikacija

* Dve strani morata vedeti, da se pogovarjata s pravo stranjo
  + Vzpoztavizev identitet na začetku **(identifikacija)**
  + Vzdrževanje čez pogovor
  + Stran je lahko oseba ali storitev/program
* Ena stran mora o drugi vedeti nekaj, česar **ne more imeti** nihče drug

# Avtentikacija z gesli

* Borut Ani pove svoje geslo
* Možni napadi:
  + **Prisluškovanje –** kraja v prenosu
  + **Vlom v sistem** – kraja shranjenjih gesel
  + **Ugibanje gesel**
* **Obrambe:**
  + Uporaba varne šifrirane povezave
  + Varovanje sistema / gesel
  + Število poskusov ugibanj omejimo
* Dodatna zaščita:
  + Ana pošlje Borutu nekaj, kar zna mora rešiti

# Hranjenje gesel

* Hranimo na vseh mestih, kjer jih potebujemo – **ranljivo, problem spreminjanja**
* Na enem mestu in jih vsi uporabljamo – **zaščita prenosa kopije do uporabnika**
* Imamo posebno vozlišče, ki nudi storitev preverjanja gesel – **poseben protokol**
* Hranjena gesla varujemo z dodatno **kriptografsko zaščito (enosmerna razpršilna funkcija)**
  + Borut izračuna f(geslo) = g
  + Borut pošlje g
  + Ana hrani v bazi g in ne gesla, ter samo preveri prisotnost g v bazi

# Napadi na gesla

* Z ugibanjem
* Omejevanje veljavnosti gesla
  + The S/KEY One-Time Password System
  + A One-Time Password System
    - otp-<algorithm identifier> <sequence integer> <seed>
    - **Race attack –** Napadalec posluša geslo in poskusi zadnji del uganiti sam ter prehiteti uporabnika, pred temu se lahko branimo tako, da dovolimo **samo eno sejo**, a zato rabimo timeout, da preprečimo **DOS** napade
    - **Postopek:**
      1. Izberemo seed (starting values) **s**
      2. Hash funkcija **f(s)** je pognana X-krat... f(f(f(f....f(s)....))).
      3. Uporabnikova prva prijava se izvede tako, da se **funkcija f izvede (X-1)** – krat, tako lahko vemo da f(p) = fx(s), to pa je naša shranjena vrednost. Vrednost se zamenja z p in uporabnik je prijavljen
      4. Naslednjo prijavo se funkcija izvede X-2 krat, vrednost v bazi je f(p), če se ujema se zamenja, uporabnik je prijavljen
      5. Ko pride X do 0, se izbere nov s
* Kraja gesel
  + Ukradeni čistopisi – **menjaj gesla**
  + Ukradene preslikave
* Na spletu obstajajo baze/storitve, ki sistematično računajo preslikave gesel
  + **Obramba:** gesla zasolimo

# Naslov kot geslo

* (IP) naslov predstavlja geslo ali njegov del – zaupanje določenim računalnikom
* Prijava samo iz teh računalnikov
  + Zaupamo, da so opravili usrezno avtentikacijo (**hosts.equiv**)
  + Dovolimo avtentikacijo samo tem računalnikom

# Zaupanja vredni posredniki

* Posrednik za rapečevanje gesel (**Key distribution centre**)
  + Posrednjik tvori ključ za vsako novo nastalo povezavo
  + Kratkoživi ključi
* Posrednik za avtentikacijo (**certification authority**)
  + Posrednik zagotavlja (**avtorizira**) geslo
  + Dolgoživa potrdila, zato jih moramo biti možno preklicati
* Hirerhija posrednikov

# Avtentikacja ljudi

* Uporaba gesla
* Avtentikacijski pripomočki
* Uporaba biometričnih značilnosti
* Drugi možnosti zahteva dodatno strojno opremo (ki ji moramo zaupati)

# PPP in PAP

* The Point-to-Point Protocol
* Nadomešča povezavno plast
* Ob pričetku seje potrebna avtentikacija
* **Protocol**:
  + 0001 Padding Protocol
  + 0003 to 001f reserved (transparency inefficient)
  + 007d reserved (Control Escape)
  + 00cf reserved (PPP NLPID)
  + 00ff reserved (compression nefficient)
  + 8001 to 801f unused
  + 807d unused
  + 80cf unused
  + 80ff unused
  + c021 Link Control Protocol
  + c023 Password Authentication Protocol – **PAP**
  + c025 Link Quality Report
  + c223 Challenge Handshake Authentication Protocol – **CHAP**
* **PAP – Password Authentication protocol**
  + Prenos gesla v čistopisu
  + Zadnja možnost, če vse ostalo odpove (in če smo še vedno pripravljeni to početi)

# CHAP – Challenge-Handshake authentication protocol (MS-CHAP)

* PPP Challange Handshake Authentication Protocol (CHAP)
* Pripravljen za potrebe PPP
* žZasnovan na osnovi izziva, ki ga pošlje Ana Borutu
* Prenosni protokol načeloma ni definiran
* **Tri koračni protokol:**
  1. Ana pošlje izziv
  2. Borut izziv združi z geslom in ga vrne šifriranega z enosmerno razpršilno f.
  3. Ana preveri pravilnost odgovora
* Koraki se pri PPP protok. Lahko poljubnomnogokrat ponovijo
* Izziv se pošlje v berljivi obliki
* Geslo se mora hraniti **na obeh straneh**
* Ker se izziv menja, je replay napad težak
* **Incrementally** se spreminja identifier in variable challenge value
* Temelji na **skrivnosti**, ki jo poznata **samo autentikator in odjemalec**, skrivnost ni poslana prek povezave
* Ni uporaben za za velike scale, zato ker je treba vzdrževati vse skrivnosti na **obeh (vseh) straneh**
* Lahko uporabimo isto skrivnost za avtenticiranje v obe smeri

# Katera razpršilna funkcija

* PPP protokol ima svoj nadzorni protokol **LCP**
* Z njim lahko nastavljamo različne lastnosti in tudi vrsto razpšilne funcije
  + |Code (Izbrana operacija) , ID, Length, Data (Parametri za operacijo) |

# CHAP – Oblika paketa

* **Code** – koda sporočila: 1, Challenge, 2 Response, 3 Success, 4 Failure
* **Identifier** – Povezovanje med koraki protokola

# MS-CHAP

* Microsoft PPP CHAP Extenstions, Version 2
* Microsoft Verzija **CHAP**
* Dve verziji (v1, v2)
  + V2 Omogoča menjavo gesla, če je slednje poteklo (**Change-Password Paket**)
  + Avtentikacija deluje v **obe smeri**
  + Uporabljena so daljša, bolj močna **gesla**
* Zasnovan na CHAP protokolu z dvema bisvenima dodatkoma
  + **Vzajemna avtentikacija**
  + **Možnost spreminjanja gesla**

# EAP – Extensible Authentication Protocol

* Okvir za protokole in **ne pravi protokol saj definira zgolj obliko sporočil**
* Običajno neposredno nad povezavno plastjo (PPP, IEEE 802 – Ethern.) A tudi UDP, TCP
* **Možnost prepošiljanja** – avtentikacijski strežnik
* Uporaba v omrežjih, kjer **povezovanje prek IP** **ni na voljo**.
* **Proces avtentikacije:**
  + Avtentikator pošlje klientu, ki se želi pridružiti omrežju, zahtevo za avtentikacijo
  + Lokalna naprava se odzove na zahtevo z **RESPOND**  paketom
  + KO avtentikator prejme **RESPOND** paket, pošlje novo zahtevo za vtentikacijo in klient se še enkrat odzove
  + Koraki od 1-3 se ponavljajo dokler avtentikator ne pošlje **SUCCESS** paketa, ki označuje, da so se vsi koraki actentikacije končali, oziroma dokler avtentikator ne more več pošiljati zahteve za avtentikacijo (če prejme nezadovoljne odgovore na zahtevo)
* **Okvir je isti kor pri CHAP** 
  + |Code (Izbrana operacija) , ID, Length, Data (Parametri za operacijo) |
  + **Request / response paket**
  + Type – kaj zahteva avtentikator in kaj odgovarja odjemalec
    - **1 Identitiy**
    - **2 Notification**
    - **3 Nak (Response only)**
    - **4 MD5-Challenge**
    - **5 One Time Password (OTP)**
    - **6 Generic Token Card (GTC)**
    - **254 Expanded Types**
    - **255 Experiental use**

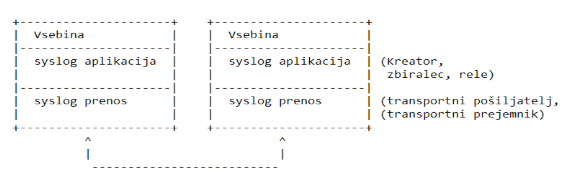
# Avtorizacija

* Ko je uporabnik avtenticiran, lahko preverimo **pravice**, ki jih ima
* Ponavadi je to določeno z uporabniškimi skupinami (Unix, MS Windows)

# 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 **lokalno** v datotekah (podobne težave kot pri hranjenju gesel)
* Lahko se hrani tudi na **strežniku**

# Beleženje

* Sistem, ki bo beležil vesbino dogodkov ter kje je in kdaj so se zgodili
* Običajna oblika beleženja na operacijskih sistemih je syslog (POSIX stadnard)
* Standardiziran tudi pri **IETF, The Sylog Protocol**
* **Možne stopnje sporočil:**
  + Emergency
  + Alert
  + Critical
  + Error
  + Warning
  + Notice
  + Info
  + Debug

# IETF

* **Uporablja tri plasti:**
  + Vsebina Syslog – info. O uporavljanju vsebovana v syslog sporocilu
  + Syslog aplikacijska plast – Upravlja z generacijo, interpreatcijo usmerjanjem in shranjevanjem syslog sporocil
  + Syslog transportna plast – daje sporočila na žlico in jih jemlje z nje
* Nekatere vrste funkcij se iz vaja na posameznem konceptualnem nivoju
  + **Kreator** generira syslog vsebino ki se nosi v sporocilu
  + **Zbiralec** zbira syslog vsebino za nadaljno analizo
  + **Rele** posreduje sporocila, sprejema sporocila od kreaterjov ali drugih relejev, in jih posilja zbirateljem ali drugim relejem
  + **Transportni pošiljatelj** preda yslog sporocila specificnemu transportnemu protokolu
  + **Transportni sprejemnik** prevzame syslog sporocila s specifičnega transportnega protokola
* Osnovni principi
  + Konceptualno je to le enostaven komunikacijski protokol
  + Kreatorje in releje lahko nastavimo tako, da pošljejo isto sporocilo na vec zbiralcev in relejev
  + Kreator, rele zbiralec so funkiconalno lahko na istem sistemu

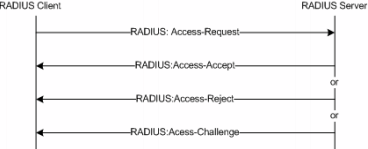
# Programska oprema

* Na freeBSD syslogd
* Konfiguracija v /etc/syslog.conf

# Syslog protokol

* Notranja **arhitektura razdeljuje**
  + **Obliko** sporočil ter njihovo **vsebino**
* Način **prenosa** sporocil**Pojmi:**
  + **Originator** – generira syslog vsebino, ki se prenese v sporocilo
  + **Collector** – zbere syslog vsebine za nadalnje analize
  + **Relay** – posreduje sporočila, sprejema sporočila od originatorjev ali drugih relayev in jih pošilja collectorjev ali drugim relayem
  + **Transportni pošiljatelj –** poda syslog soporocilo določenemu transportnemu protokolu
  + **Transportni prejenik –** prejme syslog sporocilo od dolocenega transportnega protokola
  + **TLS client -**  aplikacija, ki začne TLS povezavo s poslanimi **Client Hello** serverju
  + **TLS server –** aplikacija, ki prejme Client Hello od clienta in odgovori z **Server Hello**
* **Nevarnosti syslog groženj** (Lahko pride do nevarnosti, ker ponavadi sporočilo ne potoje direktno or originatorja do collectorja)
  + **Masquerade** – Nepravi pošiljatelj lahko pošlje pravemu prejemniku ali napačen prejemnik zavede pravega pošiljatelja
  + **Modification –** Napadalec lahko syslog sporočilo spremeni in zmede Transp. Prej.
  + **Disclosure – »**Man in the middle«, lahko dobi sensitivne informacije
  + **Message stream modification**  - Lahko spremeni zaporedje sporočil ali pa jih podvaja, kar lahko zmede Collectorja.
* **Začita TLS za zašito SYSLOG**
  + **TLS** varuje samo syslog transport v hop-by-hop načinu in ga ne skrbi vssebina syslog sporočil
  + **Port –** Syslog transportni pošiljatelj je vedno TLS client in prejemnik TLS server. TCP port 6513 je privzeti port za syslog preko TLS.
  + Transportn pošiljatelj vzpoztavi TLS Client Hello, da začne TLS rokovanje, po tem lahko pošiljatelj prvi pošlje syslog sporočilo, **za pristnost soležnikov uporablja certifikate**
  + Oba syslog transport pošiljatelj in prejemnik morata imeti **certificate-based** overitev. Imeti morata enak privatni ključ.
  + Uporablja se **SHA-1** zgoscevalna funkcija
* **Nevarnosti**
  + Neavtoriziran transportni pošiljatelj – **Masquerade attack**
  + Neavtoriziran transportni prejemnik - **Masquerade attack**
  + Neavtorizitana oba – **Venurabiltiy pred VSEMI napadi**

# Protokol RADIUS – Remote Authentication Dial In User Service

* Osnovne funkcionalnosti
  + Avtentikacija, avtorizacija, beleženje
  + Za avtentikacijo lahko uporablja druge protokole
* **Tri udeležene stranke:**
  + **Uporabnik** neke storitve
  + **Ponudnik** neke storitve – ponudnik storitve: **NAS, Network Access Server,** ki je hkrati **RADIUS odjemalec**
  + **Radius strežnik –** Lahko je tudi samo vmesni člen pri dostopu do drugega RADIUS strežnika
  + **UPORABNIK <-> NAS <-> RADIUS**
* **Komunikacija UPORABNIK – NAS**
  + Običajno neposredno na povezavni plasti
    - **PPP**
    - **Ethernet**
  + Včasih višje plasti kot na primer https
* **Komunikacija NAS- RADIUS (AA.)**
  + - Radius protokol**NAS** pošlje: **Access request**
    - RADIUS odgovori: Access Reject, Access Challange, Access Accept
    - Če ni odgovora v določenem času, se zahteva ponovno pošlje
  + RADIUS lahko pošlje zahtevo **naprej – proxy**
* **RADIUS – Zahteva za dostop**
  + Sporočilo Access Request
  + Različni protokoli – **PAP, CHAP, MS-CHAP, EAP**
* **RADIUS – Odklonitev**
  + Sporočilo **Access Reject**
  + Različni razlogi:
    - Napačno geslo / ime...
    - Neustrezne pravice
    - Dodatni pojasnilo lahko damo v sporočilo
* **RADIUS – Izziv**
  + Sporočilo Access Challenge
  + Dodatno geslo ali sporočilo v različnih primerih:
    - Drugo geslo, PIN koda, tunel, nekaj tretjega
* **RADIUS – Potrjen**
  + Sporočilo **Access Accept**
  + **RADIUS** meni, da je dostop potrjen
    - Geslo / up ime kot avtorizacija
    - Sporočilo prinaša lahko dodatne podatke, ki jih NAS potrebuje za vzpostavizev storitve (IP naslov, kako vzpoztaviti L2TP tunell); odvisno od storitve
    - NAS lahko pridobi še dodatne podatke iz drugih storitev – datoteke, LDAP
* **RADIUS – Medstrežnik in področja**
  + Proxy
  + Razdelitev uporabnikov na področja (sfere), (realm)
  + Področje je definirano s poljubnim nizom črk, ki je običajno podoben imenu domene
    - [Peter.zmeda@butale.isp](mailto:Peter.zmeda@butale.isp)
    - [peter.zmeda@gov.bu](mailto:peter.zmeda@gov.bu)
      * CN=Peter Zmeda,C=Butale,STREET=Glavna ulica \#5,O=Ob. Butale,DC=gov,DC=bu
  + Vsako območje ima svoj RADIUS strežnik
* **RADIUS – Medstrežnik in gostovanja**
  + Roaming
  + Ponudnik storitve lahko preko RADIUS strežnika dovoli gostovanje uporabnikov iz drugih domen v svojem področju
  + Uporabniku iz drugega področja lahko dodeli pravico do uporabe storitve (avtorizacija)
  + Uporabiku iz drugega področja lahko dodeli pravico do uporabe storitev (avtorizacija)
    - Vzpostavitev sodelovanja med poročji
    - Avtentikacija v drugo področje
* **RADIUS – medstrežnik in posredovanje**
  + Proxy
  + Povezave med strežniki so lahko varne (VPN)
  + Medstrežnik prejeto zahtevo lahko preoblikuje in jo posreduje pravemu strežniku
    - Medstrežnik šifrira sporočilo in ga pošlje matičnemu strežniku
    - Matični strežnik vrne šifriran odgovor

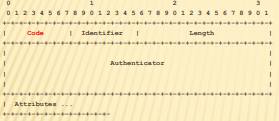
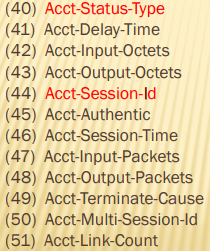
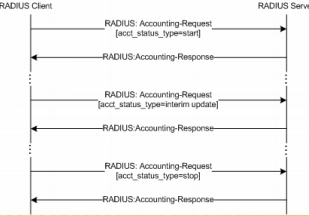
# Komunikacija nas – RADIUS (..A)

* RADIUS protokol
  + NAS pošlje: **Accounting Request**
  + Radius odgovori: **Accounting Response**
  + Če ni odgovora v določenem časi, se zahteva pošlje še enkrat
* RADIUS lahko zahtevo pošlje najprej - proxy

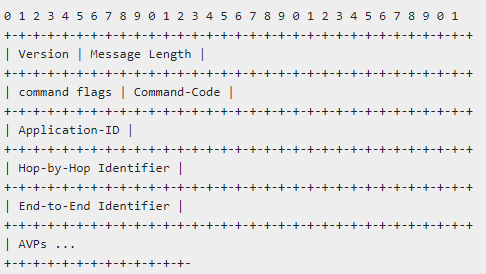
# RADIUS – Beleženje

* Beležimo lahko tri vrste dogodkov:
  + **Začetek rabe storitve**
  + **Nadaljno rabo ali popravljene podatke**
  + **Zaključek rabe**
* Razlika je v vsebini paketa, medtem ko je vse en sam par ukazov

# Protokol RADIUS

* Definirani ukazi (prin. RPC, RMI)
  + Access Request
  + Access Reject, Access Challenge, Access Accept
  + Accounting Request
  + Accounting Response
* Vsak od ukazov ima lahko različne dodatne lastnosti / parametre (attributes)
* RFC predvideva **UDP** prenosni protokol
  + RADIUS je **transakcijski protokol –** podobno kot http
  + Kounikacija je **koračna**
  + Poenostavljeno delovanje medstrežnikov, ker nimajo odprti povezav
* UDP ni varen protokol
  + Prehod na **TCP/SSL**
  + Varnost na nižjih plasteh, uporaba VPN (IPSec)
* **Podpisovanje**
  + Podpisu rečemo **authenticator** in je edini vir zagotavljanja verodostojnosti poslanega paketa
  + **NAS** in **RADIUS** strežnik imata skupni ključ **secret (shared secret)**
  + Podpisovanje AA. Paketov:
    - Odjemalec: 128 bitno naključno število – »**izziv**«
    - Strežnik (odgovor): 128 bitno število izračunano iz **secret, vsebine paketa in izziva odjemalca**
    - Podpis je uporabljen kot avtentikacija odgovora in ne ščiti zahteve odjemalca
    - Izziv v odjemalčevem podpisu se uporabi tudi kot izziv za zaščito poslanega gesla
  + Podpisovanje ..A paketov:
    - Odjemalec: 128 Bitno št. Izračunano iz **secret in vsebine paketa**
    - Strežnik (odgovor): 128 bitno število izračunano iz **secret, podpisa odjemalčevega paketa in vsebine paketa**
    - Podpis ščiti odjemalčevo zahtevo za beleženje (poskuša)
* **Podpisovanje**
  + Ni zaščite pred prisluškovanjem (zakrivanje)
  + Je **delna** zaščita verodostojnosti poslanih paketov
  + Ni zaščite pred zanikanjem poslane vsebine
* **Napadi**
  + Napad s ponavljanjem
  + Napad srednjega napadalca
  + Razlika ali gre za AA. Del ali za ..A del
  + Kako je z razpečevanjem **secret** in kako je deljen med strežnikom ter odjemalci
* **Oblika paketa**
  + **Code:**
    - 1 – Access Request
    - 2 – Access Accept
    - 3 – Acces Reject
    - 4 – Accounting Request
    - 5 – Accounting Response
    - 11 – Access Challenge
    - 12 – Status-server
    - 13 – Status-Client
    - 255 – Reservedl
  + **Identifier –** Radius protokol je koračni in mora odjemalec vedeti odgovor na katero zahtevo prejema
  + **Lenght –** dolžina celotnega paketa vključno z glavo v zlogih
    - Najmanjša dolžina je 20 in največja 4096
    - Če je paket daljši se ga skrajša na dolžino in če je krajši, ga zavrže
  + **Authenticator**  - Podpis paketa dolžine 16 zlogov:
    - **AA**. Zahteva: 128 bitno naključno število
    - **AA**. Odgovor: MD5(Code | ID | RequestAuth | Attributes | Secret)
    - **.AA** zahteva: MD5(Code |ID | Length | 00^16 | Attributes | Secret
    - **.AA** odgovor: MD5(Code |ID | Lenght | RequestAuth | Attributes | Secret)
    - **Operacija | je stik (KONKATENACIJA)**
  + **Attributes –** dodatni parametri (prilastki) poslanega ukaza
* **Protokol RADIUS – Prilastki**
  + Število možnih prilastkov je 256
  + Zahteva: uporabnik mora imeti možnost dodajanja svojih prilastkov
  + Vrednosti prilastkov naj bodo poljubne: število, datum, čas, niz
  + Prilastek: [ Type | Length | Value ]
    - **Type** – Za kateri prilastek gre
    - **Length –** število zlogov za zapis vrednosti prilastka
    - **Value –** vrednosti prilastka
      * **Besedilo:** UTF-8 kodirano dolžine večje od 0 in dolžine največ 256 zlogov
      * **Niz**: poljuben niz dolžine večje od 9 in dolžine največ 256 zlogov
      * **Naslov**: 32 bitni zapis
      * **Celo število:** 32bizni zapis
      * **Čas:** 32 bitna vrednost od 00:00:00 1.1.1979 UTC
* **Prilastki – GESLO**
  + 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[1-i]); c[i] = p[i] XOR b[i]
* **Vsi prilastki**
  + 1 – User-Name
  + 2 – User-Password
  + 3 – CHAP-Password
  + 4 – NAS-IP-Address
  + 5 – NAS-Port
  + 6 – Service-Type
  + 7 – Framed-Protocol
  + 8 – Framed-IP-Address
  + 9 – Framed-IP-Netmask
  + 10 – Framed-Routing
  + 11 – Filter-Id
  + 12 – Framed-MTU
  + 13 – Framed-Compression
  + 14 – Login-IP-Host
  + 15 – Login-Service
  + 16 – Login-TCP-Port
  + 17 – (unassigned)
  + 18 – Reply-Message
  + 19 – Callback-Number
  + 20 – Callback-Id
  + 21 – (unassigned)
  + 22 – Framed-Route
  + 23 – Framed-IPX-Network
  + 24 – State
  + 25 – Class
  + 26 – Vendor-Specific
  + 27 – Session-Timeout 28 – Idle-Timeout
  + 29 – Termination-Action
  + 30 – Called-Station-Id
  + 31 – Calling-Station-Id
  + 32 – NAS-Identifier
  + 33 – Proxy-State
  + 34 – Login-LAT-Service
  + 35 – Login-LAT-Node
  + 36 – Login-LAT-Group
  + 37 – Framed-AppleTalk-Link
  + 38 – Framed-AppleTalk- Network
  + 39 – Framed-AppleTalk-Zone
  + 40-59 – beleženje
  + 60 – CHAP-Challenge
  + 61 – NAS-Port-Type
  + 62 – Port-Limit
  + 63 – Login-LAT-Port
* **Beleženje**
  + Acct-Status-Type in Acct-Session-Id služita za podporo beleženju v okviru ene seje na storitvi, ki jo nudi NAS
  + **Status:**
    - 1 – Start
    - 2 – Stop
    - 3 – Interim Update
    - 7 – Accounting – On
    - 8 – Accounting Off
    - 9-14 – Reserved for tunnel accounting
    - 15 – Reserved for Failed

# DIAMETER

* Diameter base protocol
* Predvsem varnostni dogovor na RADIUS
* Ni povsem skladen z RADIUS
* Razlike med **RADIUS** in **DIAMETER**
  + Varnejši prenostni protokoli (TCP, ...)
  + Vgrajena omrežna varnost (SSL, Ipsec)
  + Možnih več prilastkov (**32 bitni**)
* Programska oprema: **freeDiameter**
* Aplikacijska plast
* Storitve kot so **3G,** **4G in IMS** za aplikacije dostopa do omrežja in mobilnosti
* Temelj sistema EPS (Evolved Packet System), ki omogoča tehnologijo LTE
* **Prednosti pred RADIUS:**
  + TCP ali STCP namesto UDP
  + Tajnost na nivoju paketa zagotovljena **IPSec in TLS**
  + Nudi potrditve na aplikacijski plasti
  + Podpira **sporočila o napakah**
  + Možna implementacija zraven **RADIUS**
  + Možnost definiranja novih ukazov
  + Zagotovljeni pogoki za skaliranje (32-bitni naslovni prostor)
  + Preko **DNS-ja** omogoča dinamično odkrivanje vrstnikov, odpravi varnostne liknje ki lahko nastanejo če »len« admin uporabi iste skrivnosti
* Uporablja vrata **3868 za TCP/STCP in 5868 za TLS/DTLS**
* **Command flags:** 
  + R – Request (1/0)
  + P – Proxiable (1/0)
  + E – Error (1/0)
  + T – (Potentially re-transmitted message)
  + r – Rezerviran za rabo v prihodnosti (1/0)
* **Hop-by-Hop ID:** identifikator sporočila med dvema vozliščema
* **End-to-End ID:** namenjen zaznavanju podvojenih spročil
* **Length:** dolžina glave + podatkov
* **AVP (Attribute-Value Pair)**
  + Enota, ki vsebuje podatke za aplikacijo
  + 1-255 rezervirana za RADIUS
  + 256 naprej uporabna za DIAMETER
* **AVP Flag:** Zastavice **V(Vendor Specific), M (Mandatory), P(Protected)**
* **Vendor ID:** Identifikator ponudnika (če je postavljena zastavica V)