

AAA

KOMUNIKACIJSKI PROTOKOLI IN OMREŽNA VARNOST

AAA

- × **Authentication** – avtentikacija: kdo je pravzaprava oseba (računalnik), s katerim se pogovarjamo
- × **Authorization** – avtorizacija: ali ima oseba (računalnik), s katerim se pogovarjam, pravico do vira/uporabe storitve/...
- × **Accounting** – beleženje: kdo je uporabil kdaj kakšen vir/storitev/...

VSEBINA

- ✘ avtentikacija: kaj je to, kako jo lahko izvajamo, protokoli
- ✘ avtorizacija: kako jo lahko izvajamo
- ✘ beleženje: sistemsko beleženje
- ✘ protokoli za AAA

- ✘ Literatura: C. Kaufman, R. Perlman, M. Speciner. Network Security – Private Communication in a Public World. Prentice Hall.

AVTENTIKACIJA

- ✘ dve strani (Ana in Borut) se pogovarjata in morata verjeti, da se v resnici pogovarjata s pravo stranjo
 - + vzpostavitev identitet na začetku
 - + vzdrževanje identitete skozi pogovor
 - + kako lahko verjamem, da je v resnici druga stran tista prava
 - + stran tukaj je lahko oseba ali storitev/program
- ✘ Ana mora vedeti:
 - + nekaj o Borutu, po čemer razpozna Boruta
 - + to, po čemer razpozna Boruta, ne more *imeti* nihče drug

AVTENTIKACIJA Z GESLI

- ✘ Borut Ani pove svoje geslo
- ✘ možni napadi:
 - + prisluškovanje (kraja v prenosu)
 - + vlom v sistem (kraja shranjenih gesel)
 - + ugibanje gesel
- ✘ obrambe:
 - + uporaba varne kriptografske povezave
 - + varovanje sistema / gesel
 - + število poskusov ugibanj omejimo
- ✘ dodatna zaščita
 - + Ana pošlje Borutu izziv, ki ga mora Borut znati rešiti

HRANJENJE GESEL

- ✘ gesla hranimo na vseh mestih, kjer jih potrebujemo
 - + velika ranljivost, problem spreminjanja
- ✘ gesla hranimo na enem mestu in jih vsi uporabljajo
 - + zaščita prenosa kopije do uporabnika
- ✘ imamo posebno vozlišče, ki nudi storitev preverjanja gesla
 - + poseben protokol

HHRANJENJE GESEL

- ✘ hranjena gesla varujemo dodatno s kriptografsko zaščito
- ✘ gesla ne hranimo v izvorni obliki, ampak ščitena z enosmerno razpršilno funkcijo f
 - + avtentikacija:
 1. Borut izračuna $f(\text{geslo}) \rightarrow g$
 2. Borut pošlje g
 3. Ana hrani v bazi g in ne gesla ter samo preveri prisotnost g v bazi

NAPADI NA GESLA

- ✘ z ugibanjem: omejimo število poskusov
 - + kartico avtomat zaseže
 - + geslo je veljavno omejeno število poskusov
- ✘ Omejevanje veljavnosti gesla:
 - + The S/KEY One-Time Password System, RFC1760
 - + A One-Time Password System, RFC2289
 - * **obvezno: poiščite ga na spletu ter ga preberite – literatura!**
 - * **izziv: spišite svoj programn za S/Key ali se izmislite svoj OTP.**

NAPADI NA GESLA

- × kraja gesel
 - + ukradeni čistopisi – menjaj gesla
 - + ukradene preslikave
- × na spletu obstajajo baze/storitve, ki sistematično računajo preslikave gesel
 - + možna obramba – gesla zasolimo
 - × izziv: kako izvesti soljenje?

NASLOV KOT GESLO

- ✘ (IP) naslov predstavlja geslo ali njegov del
 - + zaupanje določenim računalnikom
- ✘ prijava samo iz teh računalnikov
 - + zaupamo tem računalnikom, da so opravili ustrezno avtentikacijo (datoteka hosts.equiv,)
 - + dovolimo avtentikacijo samo tem računalnikom
 - + **obvezno: proučite, kako je z avtentikacijo in naslovom pri ssh?**

ZAUPANJA VREDNI POSREDNIKI

- ✘ posrednik za razpečevanje gesel (*key distribution centre*)
 - + posrednik tvori ključ (geslo) za vsako novo nastalo povezavo
 - + kratkoživi ključi
- ✘ posrednik za avtentikacijo (*certification authority*)
 - + posrednik zagotavlja (avtorizira) geslo
 - + dolgoživa potrdila, zato jih mora biti možno preklicati
- ✘ hierarhija posrednikov

AVTENTIKACIJA LJUDI

- ✘ uporaba gesla
- ✘ avtentikacijski pripomočki
- ✘ uporaba biometričnih značilnosti

- ✘ drugi možnosti zahtevata dodatno strojno opremo (ki ji moramo zaupati)

GESLA

- ✘ geslo ne sme biti preprosto: dolžina, število znakov, kateri znaki, ...
 - + admin/admin, 1234, EMŠO
- ✘ geslo ne sme biti prezapleteno
 - + NaWUwra66nu5UHAd ☹
 - ✘ izziv: poiščite sisteme za tvorjenje varnih gesel.
- ✘ gesla sistematično menjamo
- ✘ kaj, če geslo pozabimo?

AVTENTIKACIJSKI PRIPOMOČKI

- ✘ kartice
 - + samo nosilci informacije (magnetni zapis, optični zapis, ...)
- ✘ pametne kartice
 - + vsebujejo računalnik, ki ščiti informacijo in za dostop do računalnika potrebujemo geslo, ...
 - + uporaba izziva
- ✘ kriptografski računalniki
 - + tvorijo časovno odvisna gesla

BIOMETRIČNE ZNAČILNOSTI

- ✘ nadomestijo geslo
- ✘ neprenosljivost

- ✘ retina, prtni odtis, razpoznavna obraza, zenica, glas, ...

POSTOPEK AVTENTIKACIJE

- × neposredno
 - + prijava na konzolo računalnika
 - + oddaljen dostop: telnet (TELNET Protocol, RFC 139), ssh (ali obstaja RFC za ssh?)
 - × izziv: poiščite ostale RFC dokumente o telnet-u.
- × ad hoc način
- × z uporabo protokola

PROTOKOLI ZA AVTENTIKACIJO

- ✘ PPP in PAP: Password authentication protocol
- ✘ CHAP: Challenge-handshake authentication protocol (MS-CHAP)
- ✘ EAP: Extensible Authentication Protocol

PPP IN PAP

- ✘ The Point-to-Point Protocol (PPP), RFC 1661
 - + *izziv: poiščite in preberite RFC.*
- ✘ nadomešča povezavno plast
- ✘ ob pričetku seje potrebna avtentikacija

PPP

Protocol	Information	Padding
8/16 bits	*	*

- × protocol:
 - × 0001 Padding Protocol
 - × 0003 to 001f reserved (transparency inefficient)
 - × 007d reserved (Control Escape)
 - × 00cf reserved (PPP NLPID)
 - × 00ff reserved (compression inefficient)
 - × 8001 to 801f unused
 - × 807d unused
 - × 80cf unused
 - × 80ff unused
 - × c021 Link Control Protocol
 - × **c023 Password Authentication Protocol**
 - × c025 Link Quality Report
 - × **c223 Challenge Handshake Authentication Protocol**

PAP

- ✘ prenos gesla v čistopisu
- ✘ zadnja možnost, če vse ostalo odpove (in če smo še vedno pripravljeni to početi)

CHAP

- ✘ PPP Challenge Handshake Authentication Protocol (CHAP), RFC 1994
 - ✘ *obvezno: poiščite ga na spletu ter ga preberite – literatura!*
- ✘ pripravljen za potrebe PPP (point to point protocol)
- ✘ zasnovan na osnovi izziva, ki ga pošlje Ana Borutu
- ✘ prenosni protokol načeloma ni definiran (glej zgoraj PPP)

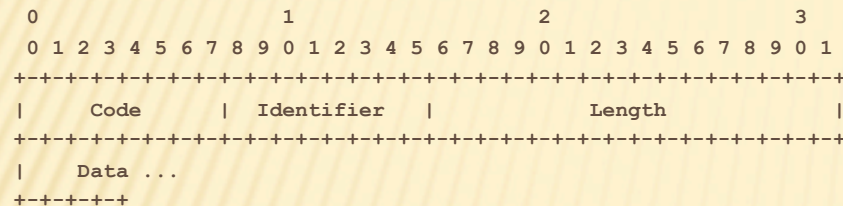
CHAP

- ✘ tri koračni protokol:
 1. Ana pošlje izziv
 2. Borut izziv združi z geslom in ga vrne zakriptiranega z enosmerno razpršilno funkcijo
 3. Ana preveri pravilnost odgovora
- ✘ koraki se pri PPP protokolu lahko poljubnomnogokrat ponovijo
- ✘ izziv se pošlje v berljivi obliki
- ✘ geslo se mora hraniti na obeh straneh
- ✘ ker se izziv menja, težko napasti s ponavljanjem

KATERA RAZPRŠILNA FUNKCIJA

- ✘ ppp protokol ima svoj nadzorni protokol LCP
- ✘ z njim lahko nastavljamo različne lastnosti in tudi vrsto razpršilne funkcije
 - + *izziv: kje in kako to nastavimo*

CHAP – OBLIKA PAKETA



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

MS-CHAP

- ✘ Microsoft PPP CHAP Extensions, Version 2, RFC 2759
 - + *izziv: poiščite ga na spletu ter ga preberite; kako je izvedena zamenjava gesla in na kaj je potrebno pri tem paziti?*
- ✘ obstaja dve inačici
 - + *obvezno: v čem se inačica dve razlikuje od ena?*
- ✘ zasnovan na CHAP protokolu z dvema bistvenima dodatkom:
 - + vzajemna avtentikacija
 - + možnost spreminjanja gesla

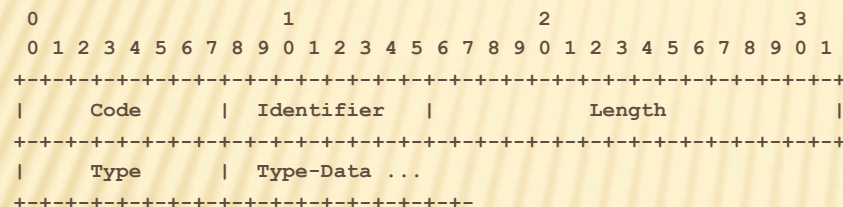
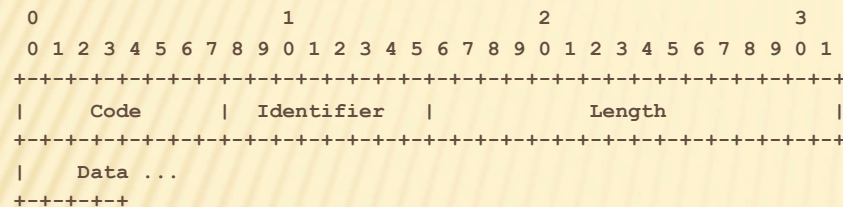
EAP

- ✘ Extensible Authentication Protocol (EAP), RFC 3748 – osnovni protokol in popravki v RFC5247
 - + *izziv: poiščite in preberite RFC*
- ✘ okvir za protokole in ne pravi protokol saj definira zgolj obliko sporočil
- ✘ običajno neposredno nad povezavno plastjo (ppp, IEEE 802 – ethernet) a tudi UDP, TCP
 - + *izziv: v RFC poiščite, kateri protokol uporablja UDP*
- ✘ možnost prepošiljanja – avtentikacijski strežnik

EAP – OSNOVNO DELOVANJE

- ✘ način avtentikacije se doreče med odjemalcem in strežnikom (avtentikatorjem)
- ✘ koračni protokol:
 1. avtentikator pošlje zahtevo po podatkih; npr. identifikacija, zahteva za avtentikacijo vključno z načinom avtentikacije, ...
 2. odjemalec odgovori ali zavrne način avtikacije
 3. koraka 1. in 2. se ponavljata dokler strežnik ne ugotovi identitet odjenalca

EAP – OBLIKA PAKETA



- identična CHAP
- request/response paket
- type – kaj zahteva avtentikator in kaj odgovarja odjemalec:
 - 1 Identity
 - 2 Notification
 - 3 Nak (Response only)
 - 4 MD5-Challenge
 - 5 One Time Password (OTP)
 - 6 Generic Token Card (GTC)
 - 254 Expanded Types
 - 255 Experimental use

AVTORIZACIJA

- ✘ ko je uporabnik avtenticiran (identificiran), lahko preverimo pravice, ki jih ima
- ✘ na Unix sistemih običajno postane član skupine ali večih skupin, katere imajo določene pravice (*group*)
- ✘ na MS Windows sistemih podobno
 - ✘ *izziv: obstaja RFC 2904, AAA Authorization Framework. O čem govori in definira kakšne zahteve ali kaj drugega?*

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 datoteki/datotekah
 - + podobne težave kot pri hranjenju gesel
- ✘ hrani se na strežniku
 - ✘ *izziv: kako je z varnostjo prenešenih sporočil in njihovim kriptiranjem?*

BELEŽENJE

- ✘ sistem, ki bo beležil vsebino dogodkov ter kje in kdaj so se zgodili
- ✘ običajna oblika beleženja na operacijskih sistemih je syslog (POSIX standard)
- ✘ standardiziran tudi pri IETF kot RFC 5424, *The Syslog Protocol*.
 - + izziv: primerjajte RFC z “man -k syslog” stranmi?
 - + izziv: poiščite še ostale RFCje o syslogu in IETF stran, kjer je delovna skupina za syslog objavljala dokumente.

BELEŽENJE IN SYSLOG

✘ log se hrani v datoteko /var/log ...:

- + Nov 13 17:00:17 svarun0 sshd[92530]: error: PAM: authentication error for root from ip-62-129-164-36.evc.net
- + možne stopnje sporočil: Emergency, Alert, Critical, Error, Warning, Notice, Info or Debug
- + *izziv: Poglejte si datoteke /var/log/...*

PROGRAMSKA OPREMA

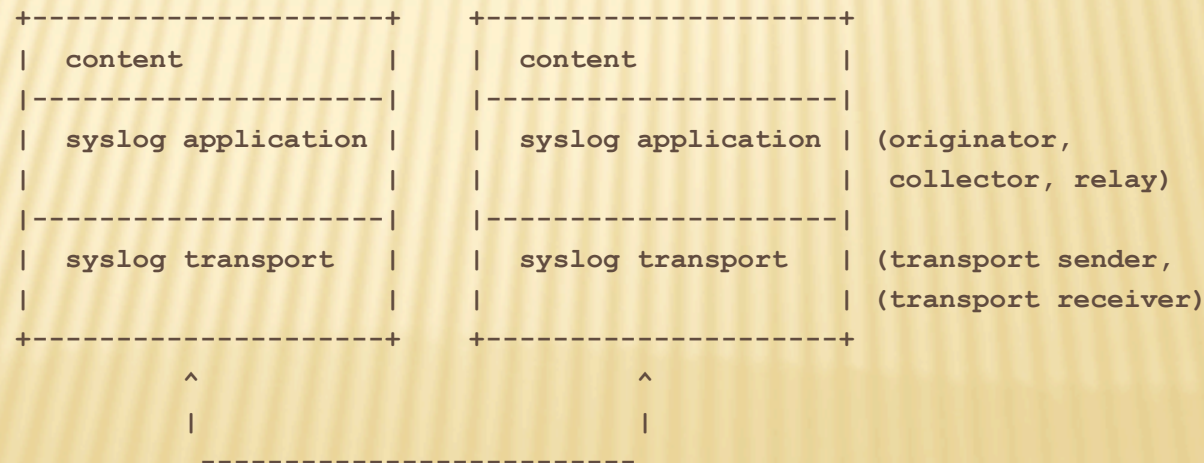
- ✗ na FreeBSD syslogd
- ✗ konfiguracija v /etc/syslog.conf
 - + *izziv: spremenite konfiguracijo tako, da se bodo vsa sporočila zapisovala v /var/log/super-log; kako poslati zabeležko na drug računalnik?; ali lahko isto zabeležko shranimo na več mest?*

```
security.*
auth.info;authpriv.info
mail.info
lpr.info
ftp.info
cron.*
```

```
/var/log/security
/var/log/auth.log
/var/log/maillog
/var/log/lpd-errs
/var/log/xferlog
/var/log/cron
```

SYSLOG PROTOKOL

- ✘ notranja arhitektura razdeljuje:
 - + obliko sporočil ter njihovo vsebino (RFC 5424)
 - + način prenosa sporočil (RFC 5425)
 - ✘ **obvezno: poiščite RFC 5425 in pogledjte o katerih sestavinah govori – literatura!**
 - ✘ **izziv: poiščite še ostale RFCje, ki govorijo o syslog.**



SYSLOG PROTOKOL – OBLIKA SPOROČIL

SYSLOG-MSG = HEADER SP STRUCTURED-DATA [SP MSG]

HEADER = PRI VERSION SP TIMESTAMP SP HOSTNAME
SP APP-NAME SP PROCID SP MSGID

PRI = "<" PRIVAL ">"

PRIVAL = 1*3DIGIT ; range 0 .. 191

VERSION = NONZERO-DIGIT 0*2DIGIT

HOSTNAME = NILVALUE / 1*255PRINTUSASCII

APP-NAME = NILVALUE / 1*48PRINTUSASCII

PROCID = NILVALUE / 1*128PRINTUSASCII

MSGID = NILVALUE / 1*32PRINTUSASCII

TIMESTAMP = NILVALUE / FULL-DATE "T" FULL-TIME

FULL-DATE = DATE-FULLYEAR "-" DATE-MONTH "-" DATE-MDAY

DATE-FULLYEAR = 4DIGIT

DATE-MONTH = 2DIGIT ; 01-12

DATE-MDAY = 2DIGIT ; 01-28, 01-29, 01-30, 01-31 based on
; month/year

FULL-TIME = PARTIAL-TIME TIME-OFFSET

PARTIAL-TIME = TIME-HOUR ":" TIME-MINUTE ":" TIME-SECOND
[TIME-SECFRAC]

TIME-HOUR = 2DIGIT ; 00-23

TIME-MINUTE = 2DIGIT ; 00-59

TIME-SECOND = 2DIGIT ; 00-59

TIME-SECFRAC = "." 1*6DIGIT

TIME-OFFSET = "Z" / TIME-NUMOFFSET

TIME-NUMOFFSET = ("+" / "-") TIME-HOUR ":" TIME-MINUTE

STRUCTURED-DATA = NILVALUE / 1*SD-ELEMENT

SD-ELEMENT = "[" SD-ID *(SP SD-PARAM) "]"

SD-PARAM = PARAM-NAME "=" %d34 PARAM-VALUE %d34

SD-ID = SD-NAME

PARAM-NAME = SD-NAME

PARAM-VALUE = UTF-8-STRING ; characters '"', '\', and
; ']' MUST be escaped.

SD-NAME = 1*32PRINTUSASCII
; except '=', SP, ']', %d34 (")

MSG = MSG-ANY / MSG-UTF8

MSG-ANY = *OCTET ; not starting with BOM

MSG-UTF8 = BOM UTF-8-STRING

BOM = %xEF.BB.BF

UTF-8-STRING = *OCTET ; UTF-8 string as specified
; in RFC 3629

OCTET = %d00-255

SP = %d32

PRINTUSASCII = %d33-126

NONZERO-DIGIT = %d49-57

DIGIT = %d48 / NONZERO-DIGIT

NILVALUE = "."

PROTOKOL RADIUS

- ✘ definiran v RFC 2865, *Remote Authentication Dial In User Service (RADIUS)* in RFC 2866, *RADIUS Accounting*
 - ✘ *obvezno: poiščite ga na spletu ter ga preberite – literatura!*
 - ✘ *izziv: poiščite še ostale RFC dokumente, ki se ukvarjajo s tftp ter preverite, kaj piše v njih.*
- ✘ osnovne funkcionalnosti:
 - + avtentikacija, avtorizacija, beleženje
 - + za avtentikacijo lahko uporablja druge protokole
 - + glej tudi RFC 4962, *Guidance for Authentication, Authorization, and Accounting (AAA) Key Management*

RADIUS – OSNOVNA ARHITEKTURA

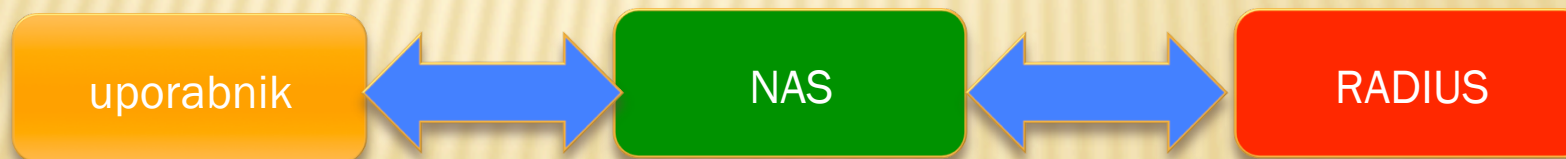
× tri udeležene stranke:

+ **uporabnik** neke storitve

+ **ponudnik storitve** – ponudnik storitve: NAS, *Network access server*, ki je hkrati **RADIUS odjemalec**

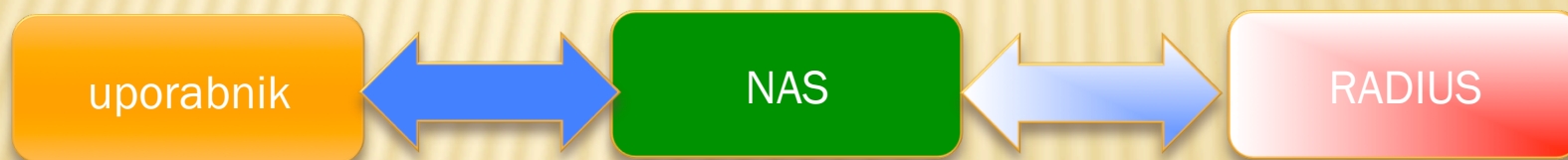
+ **RADIUS strežnik**

+ RADIUS strežnik je lahko samo vmesni člen pri dostopu do drugega RADIUS strežnika



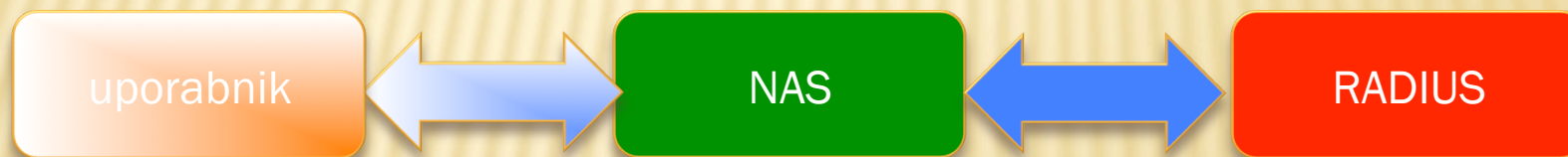
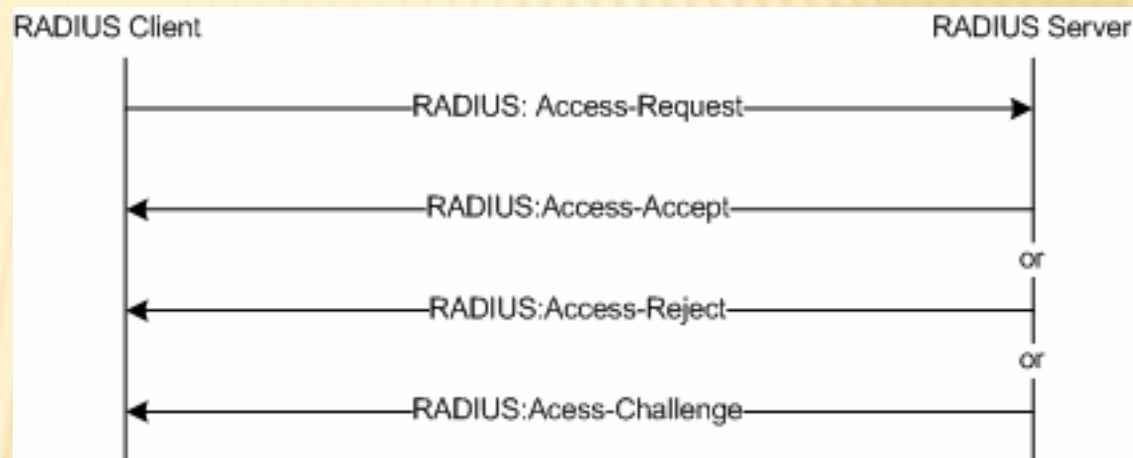
KOMUNIKACIJA UPORABNIK – NAS

- ✘ običajno neposredno na povezavni (!) plasti
 - + ppp
 - + ethernet
- ✘ včasih višje plasti kot na primer https
- ✘ varnost!



KOMUNIKACIJA NAS – RADIUS (AA.)

- ✗ RADIUS protokol
 - + NAS pošlje: *Access Request*
 - + RADIUS odgovori: *Access Reject, Access Challenge, 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
 - + izziv: preglej, kako je podprt MS-CHAP; RFC 2548, *Microsoft Vendor-specific RADIUS Attributes*.
 - + izziv: kako je s podporo za EAP?

RADIUS – ODKLONITEV

- ✘ sporočilo *Access Reject*
- ✘ različni razlogi:
 - + napačno geslo / uporabniško ime, ...
 - + neustrezne pravice
 - + dodatno pojasnilo lahko v sporočilo

RADIUS – IZZIV

- ✘ sporočilo *Access Challenge*
- ✘ dodatno geslo ali sporočilo v različnih primerih:
 - + drugo geslo,
 - + PIN koda
 - + vzpostavljen tunel med uporabnikom in avtentikatorjem, ...
 - + nekaj tretjega ...

RADIUS – POTRJEN

- ✘ sporočilo *Access Accept*
- ✘ RADIUS meni, da je dostop potrjen / dovoljen
 - + tako geslo/uporabniško ime kot avtorizacija
 - + sporočilo prinaša lahko dodatne podatke, ki jih NAS potrebuje za vzpostavitev storitve (IP naslov, kako vzpostaviti L2TP tunel, ...); 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
 - × andrej.brodnik@fri.uni-lj.si
- × vsako območje ima svoj RADIUS strežnik

RADIUS – MEDSTREŽNIK IN GOSTOVANJA

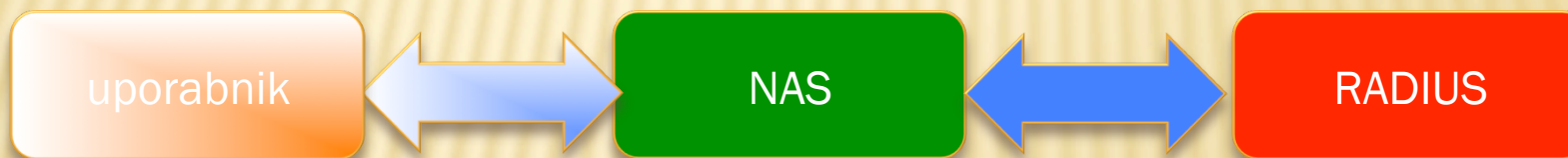
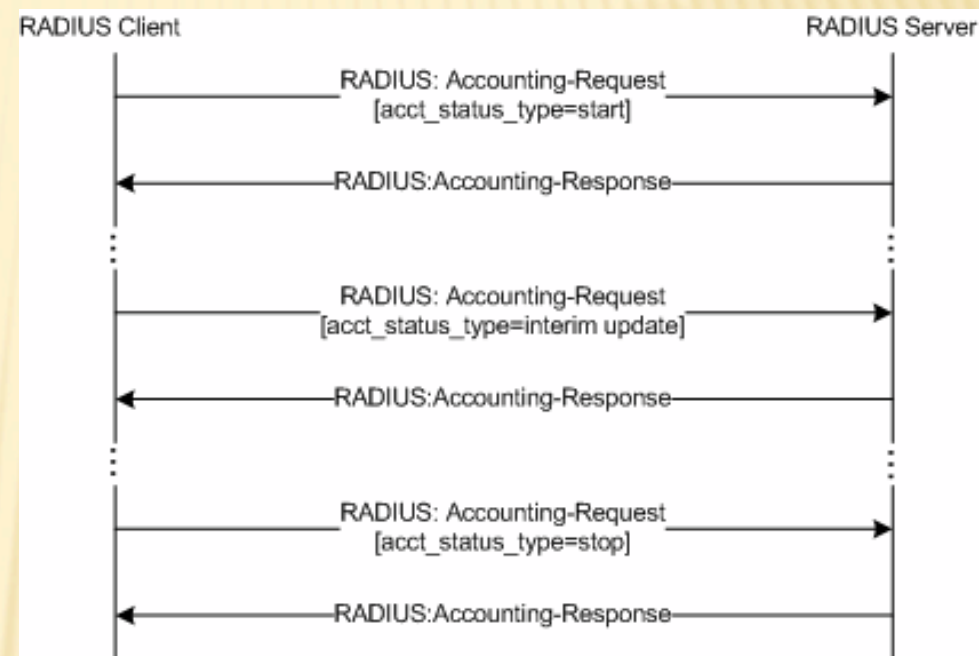
- ✘ *roaming*
- ✘ ponudnik storitve lahko preko RADIUS strežnika dovoli gostovanje uporabnikov iz drugih domen v svojem področju
- ✘ uporabniku iz drugega področja lahko dodeli pravico do uporabe storitev (avtorizacija)
 - + vzpostavitev sodelovanja med področji
 - + avtentikacija v drugo področje

RADIUS – MEDSTREŽNIK IN PREPOSREDOVANJE

- ✗ *proxy*
- ✗ povezave med strežniki so lahko varne (VPN)
- ✗ medstrežnik prejeto zahtevo lahko preoblikuje in jo posreduje pravem strežniku (skoraj, glej RFC 2865):
 - + medstrežnik zakriptira sporočilo in ga pošlje matičnemu strežniku
 - + matični strežnik vrne zakriptiran odgovor
 - ✗ izziv: kaj lahko in kako spreminja medstrežnik?

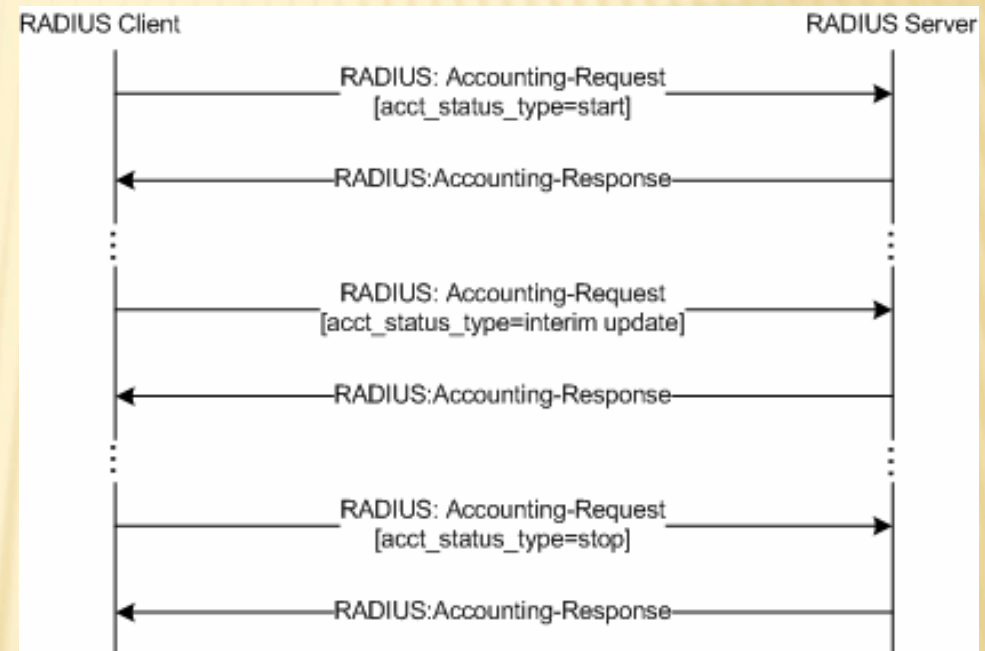
KOMUNIKACIJA NAS – RADIUS (..A)

- ✗ RADIUS protokol
 - + NAS pošlje: *Accounting Request*
 - + RADIUS odgovori: *Accounting Response*
 - + če ni odgovora v določenem času, se zahteva ponovno pošlje
- ✗ RADIUS lahko pošlje zahtevo naprej – *proxy*



RADIUS – BELEŽENJE

- ✘ beležimo lahko tri vrste dogodkov:
 - + začetek rabe storitve
 - + nadaljnjo rabo ali popravljene podatke
 - + zaključek rabe
- ✘ razlika je v vsebini paketa, medtem ko je za vse en sam par ukazov



PROTOKOL RADIUS

- ✘ definirani ukazi (prim. *RPC, RMI*):
 - + *Access Request*
 - + *Access Reject, Access Challenge, Access Accept*
 - + *Accounting Request*
 - + *Accounting Response*
- ✘ vsak od ukazov ima lahko različne dodatne lastnosti / parametre (*attributes*)

PROTOKOL RADIUS

- ✘ RFC predvideva UDP prenosni protokol
 - + RADIUS je transakcijski protokol – podobno kot http
 - + komunikacija je koračna
 - + poenostavljeno delovanje medstrežnikov, ker nimajo odprtih povezav
- ✘ UDP ni varen protokol
 - + prehod na TCP/SSL
 - + varnost na nižjih plasteh: uporaba VPN (IPSec)

PROTOKOL RADIUS – PODPISOVANJE

- ✘ podpisu rečemo *autheticator* in je edini vir zagotavljanja verodostojnosti poslanega paketa
- ✘ NAS in RADIUS strežnik imata skupni ključ *secret (shared secret)*

PROTOKOL RADIUS – PODPISOVANJE

- ✘ podpisovanje AA. paketov:
 - + odjemalec: 128 bitno naključno število – sol
 - + strežnik (odgovor): 128 bitno število izračunano iz *secret*, vsebine paketa in soli odjemalca
 - + podpis je uporabljen kot avtentikacija odgovora in ne ščiti zahteve odjemalca
 - + sol v odjemalčevem podpisu se uporabi tudi kot sol za zaščito poslanega gesla

PROTOKOL RADIUS – PODPISOVANJE

- ✘ podpisovanje ..A paketov:
 - + odjemalec: 128 bitno število 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 (poskušaja)

PROTOKOL RADIUS – VARNOST

× Zaščita:

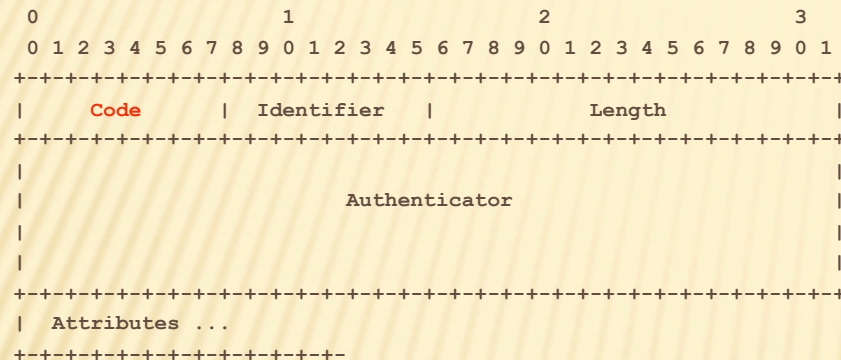
- + ni zaščite pred prisluškovanjem (zakrivanje)
- + je (delna) zaščita verodostojnosti poslanih paketov
- + ni zaščite pred zanikanjem poslane vsebine
 - × izziv: poiščite poglobljenejšo analizo varnosti RADIUS protokola?

PROTOKOL RADIUS – VARNOST

× Napadi:

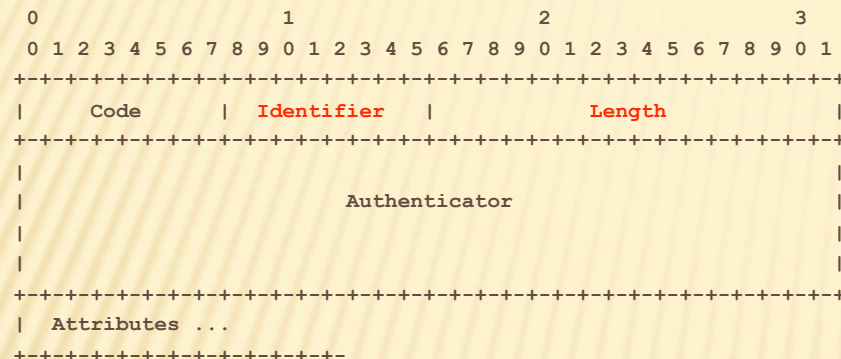
- + napad s ponavljanjem
- + napad srednjega napadalca
- + razlika ali gre za AA. del ali za ..A del
- + kako je z razpečevanjem *secret* in kako je deljen med strežnikom ter odjemalci
 - × izziv: pogledjte, kako se rokuje s *secret*?

RADIUS – OBLIKA PAKETA



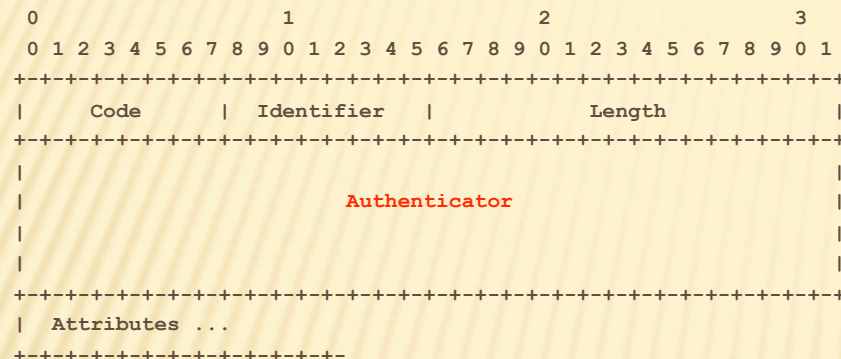
- Code – koda ukaza:
 - (1) Access-Request
 - (2) Access-Accept
 - (3) Access-Reject
 - (4) Accounting-Request
 - (5) Accounting-Response
 - (11) Access-Challenge
 - (12) Status-Server (poskusno)
 - (13) Status-Client (poskusno)
 - (255) Reserved

RADIUS – OBLIKA PAKETA



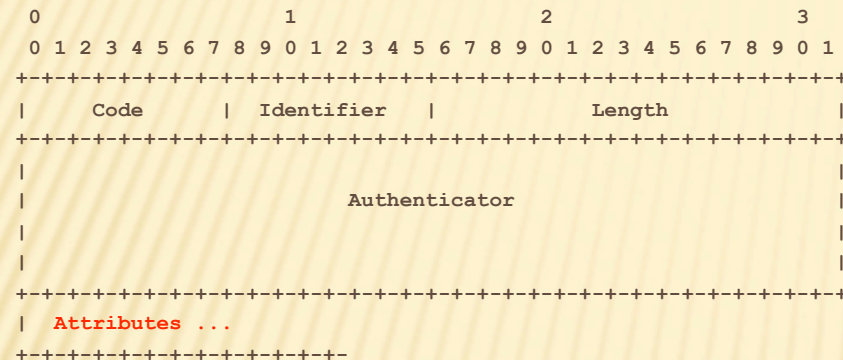
- Identifier – RADIUS protokol je koračni protokol in mora odjemalec vedeti odgovor na katero zahtevo prejema
- Length – dolžina celotnega paketa vključno z glavo v zlogih
 - najmanjša dolžina je 20 in največja 4096
 - če je paket daljši se ga skrajša na dolžino in če je krajši, se ga zavrže

RADIUS – OBLIKA PAKETA



- Authenticator – „podpis” paketa dolžine 16 zlogov:
 - AA. zahteva: 128 bitno naključno število
 - AA. odgovor: $MD5(\text{Code} \bullet ID \bullet \text{Length} \bullet \text{RequestAuth} \bullet \text{Attributes} \bullet \text{Secret})$
 - ..A zahteva: $MD5(\text{Code} \bullet ID \bullet \text{Length} \bullet 00^{16} \bullet \text{Attributes} \bullet \text{Secret})$
 - ..A odgovor: $MD5(\text{Code} \bullet ID \bullet \text{Length} \bullet \text{RequestAuth} \bullet \text{Attributes} \bullet \text{Secret})$
- operacija • je stik (konkatenacija)

RADIUS – OBLIKA PAKETA

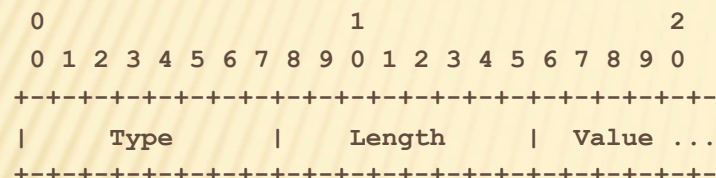


- Attributes – dodatni parametri poslanega ukaza

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, ...

RADIUS – PRILASTKI



- *TLV* zapis
- *Type* – za kateri prilastek gre
- *Length* – število zlogov za zapis vrednosti prilastka
- *Value* – vrednost prilastka
 - besedilo: UTF-8 kodirano dolžine večje od 0 in dolžine največ 256 zlogov
 - niz: poljuben niz dolžine večje od 0 in dolžine največ 256 zlogov
 - naslov: 32 bitni zapis
 - celo število: 32 bitni zapis
 - čas: 32 bitna vrednost od 00:00:00 1.1.1970 UTC (standardni prilastki ne uporabljajo)

PROTOKOL RADIUS – PRILASTKI

- ✘ sprehod skozi prilastke:
 - + (1) User-Name
 - + (2) User-Password
 - + (3) CHAP-Password

PROTOKOL RADIUS – PRILASTKI: GESLO

- ✘ geslo se zakriptira z uporabo soli v avtentikatorju (RA) in skupne skrivnosti (S):
 - + geslo razdelimo v 128-bitne dele $p[1..n]$
 - + $b[1] = MD5(S \cdot RA)$; $c[1] = p[1] \text{ XOR } b[1]$
 - + ...
 - + $b[i] = MD5(S \cdot c[i-1])$; $c[i] = p[i] \text{ XOR } b[i]$

PROTOKOL RADIUS – PRILASTKI

× sprehod skozi prilastke:

- × (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

PROTOKOL RADIUS – PRILASTKI

× sprehod skozi prilastke:

- × (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

PROTOKOL RADIUS – PRILASTKI

× sprehod skozi prilastke – beleženje:

- × (40) **Acct-Status-Type**
- × (41) Acct-Delay-Time
- × (42) Acct-Input-Octets
- × (43) Acct-Output-Octets
- × (44) **Acct-Session-Id**
- × (45) Acct-Authentic
- × (46) Acct-Session-Time
- × (47) Acct-Input-Packets
- × (48) Acct-Output-Packets
- × (49) Acct-Terminate-Cause
- × (50) Acct-Multi-Session-Id
- × (51) Acct-Link-Count

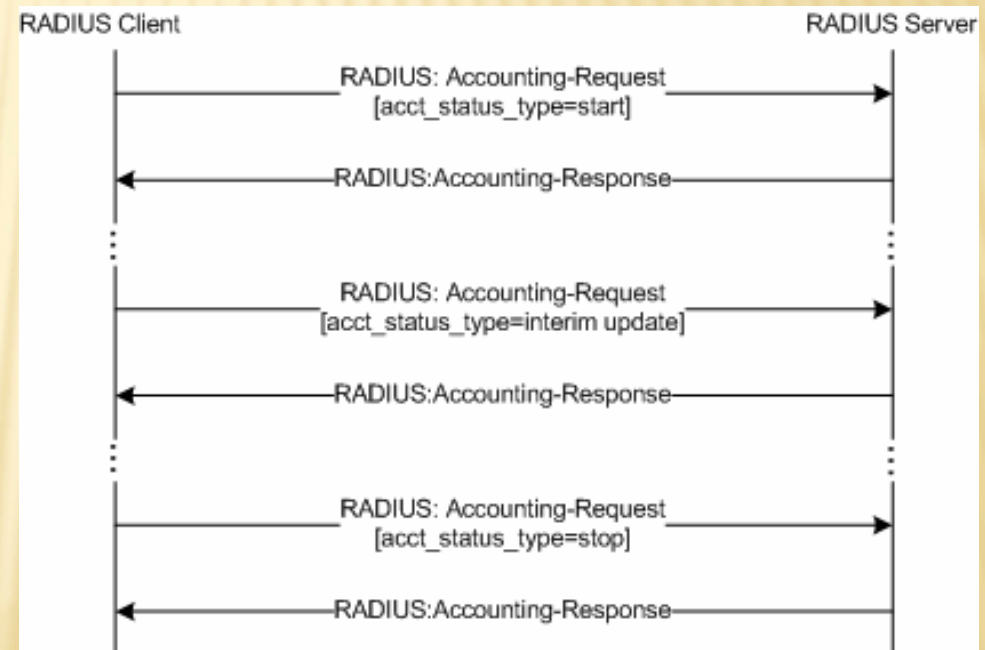
- × *izziv: kaj je s prilastki 52-59 in 64-255?*
- × *izziv: kaj je s prilastkoma 17 in 21?*

PROTOKOL RADIUS – 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



PROGRAMSKA OPREMA

- ✘ Na FreeBSD (Linux): freeradius
- ✘ konfiguracija v `/usr/local/etc/radiusd.conf`
 - + izziv: *poiščite priročnik ter samo nastavite datoteko ter poženite strežnik.*
 - + izziv: *kje je shranjena skupna skrivnost in kako je deljena med strežnikom in odjemalci?*
 - + izziv: *kje se hrani zabeleške?*
 - + izziv: *kako lahko RADIUS uporabi druge storitve za avtentikacijo?*

DIAMETER

- ✘ definiran v RFC 3588, *Diameter Base Protocol* in RFC 5719, 5729
 - * **obvezno: poiščite ga na spletu ter ga preberite – literatura!**
 - * **izziv: poiščite še ostale RFC dokumente, ki se ukvarjajo s tftp ter preverite, kaj piše v njih.**
- ✘ predvsem varnostni odgovor na RADIUS
- ✘ ni povsem skladen z RADIUS

DIAMETER

- ✘ razlike med RADIUS in DIAMETER:
 - + varnejši prenosni protokoli (TCP, ...)
 - + vgrajena omrežna varnost (SSL, IPsec)
 - + možnih več prilastkov (32-bitni)
- ✘ programska oprema: freeDiameter