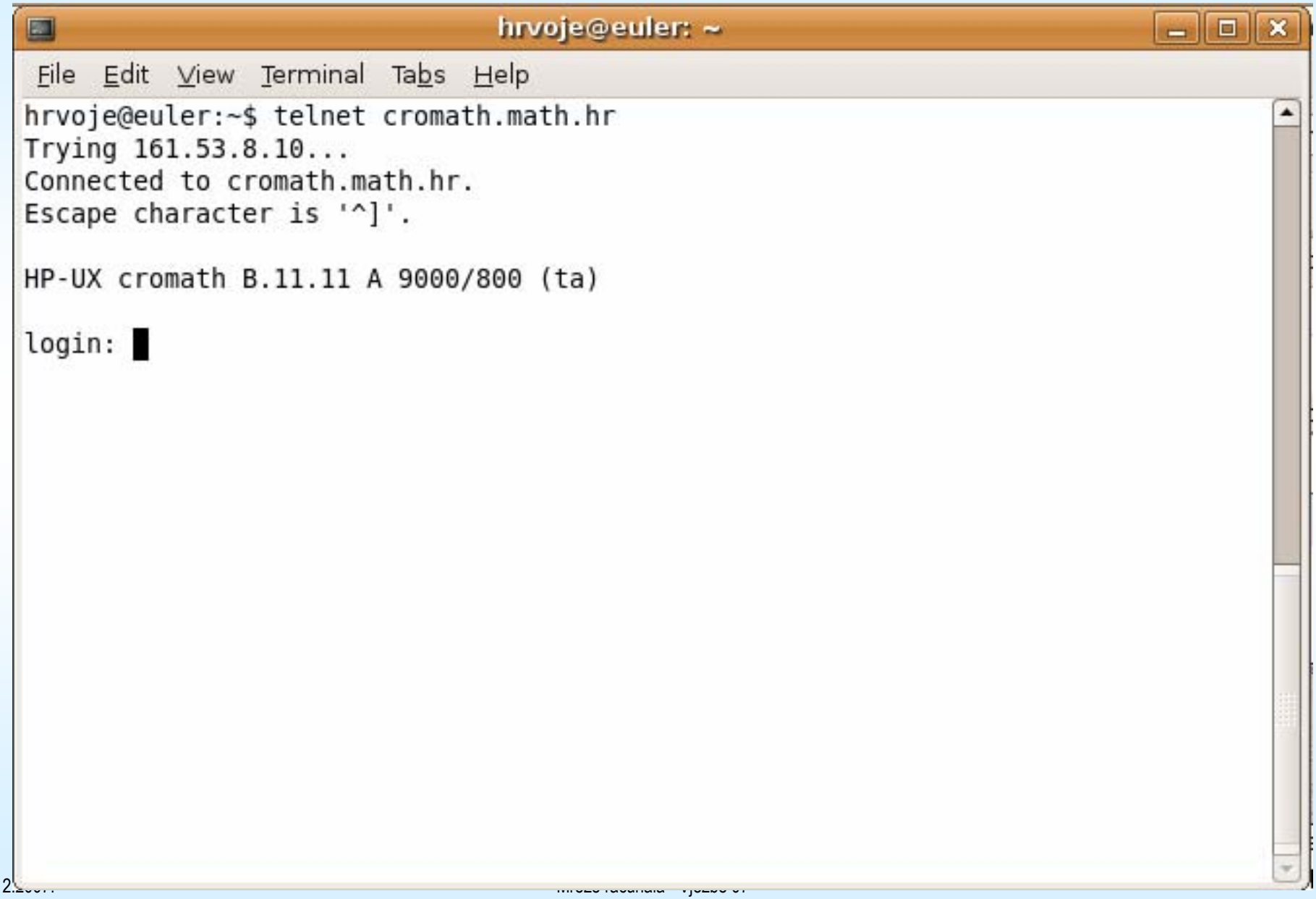
Prije početka...

- Upoznat ćemo se sa klijentima i serverima najkorištenijih mrežnih aplikacija kao što su telnet, ssh, ftp, web, smtp, pop3, imap, dns, dhcp...
- Objasnit ćemo pozadinu i principe funkcioniranja protokola koje te aplikacije koriste

telnet

- telnet (TELecommunication NETwork) je mrežni protokol koji se koristi za rad na udaljenom računalu.
- Razvijen 1969. godine, ne koristi enkripciju podataka, iz sigurnosnih razloga ga sve više zamjenjuje SSH.
- zbog jednostavnosti implementacije i zanemarivih zahtjeva za sistemskim resursima, kao metodu konfiguracije i podešavanja podržava ga mnoštvo malih mrežnih uređaja poput kućnih ADSL routera, VOIP telefona, bežičnih pristupnih točaka (AP) i sl.
- telnet poslužitelj standardizirano sluša na TCP portu br. 23
- on se brine samo za ostvarivanje komunikacije, nakon spajanja na port 23, na Unix sustavima on poziva program *login* koji zatim, nakon ispravnog unošenja korisničkog imena i lozinke poziva korisničku ljusku, odn. *shell*

primjer korištenja telneta



telnet

- uočimo liniju " Escape character is '^]' “ - escape character je kombinacija tipki na tipkovnici (ctrl+]) koja nam omogućuje da kontrolu privremeno prebacimo s udaljenog na lokalno računalo
- Na primjer, ukoliko smo prijavljeni na *cromath.math.hr*, a želimo se u tom prozoru privremeno vratiti radu na lokalnom računalu bez da zatvorimo vezu sa *cromathom*, potrebno je *istovremeno* stisnuti kombinaciju tipki CTRL i] te bi se pokazao ovakav prompt: *telnet>*
- na udaljeni sustav se vraćate ako ne unesete nijednu naredbu već samo stisnete tipku enter

telnet

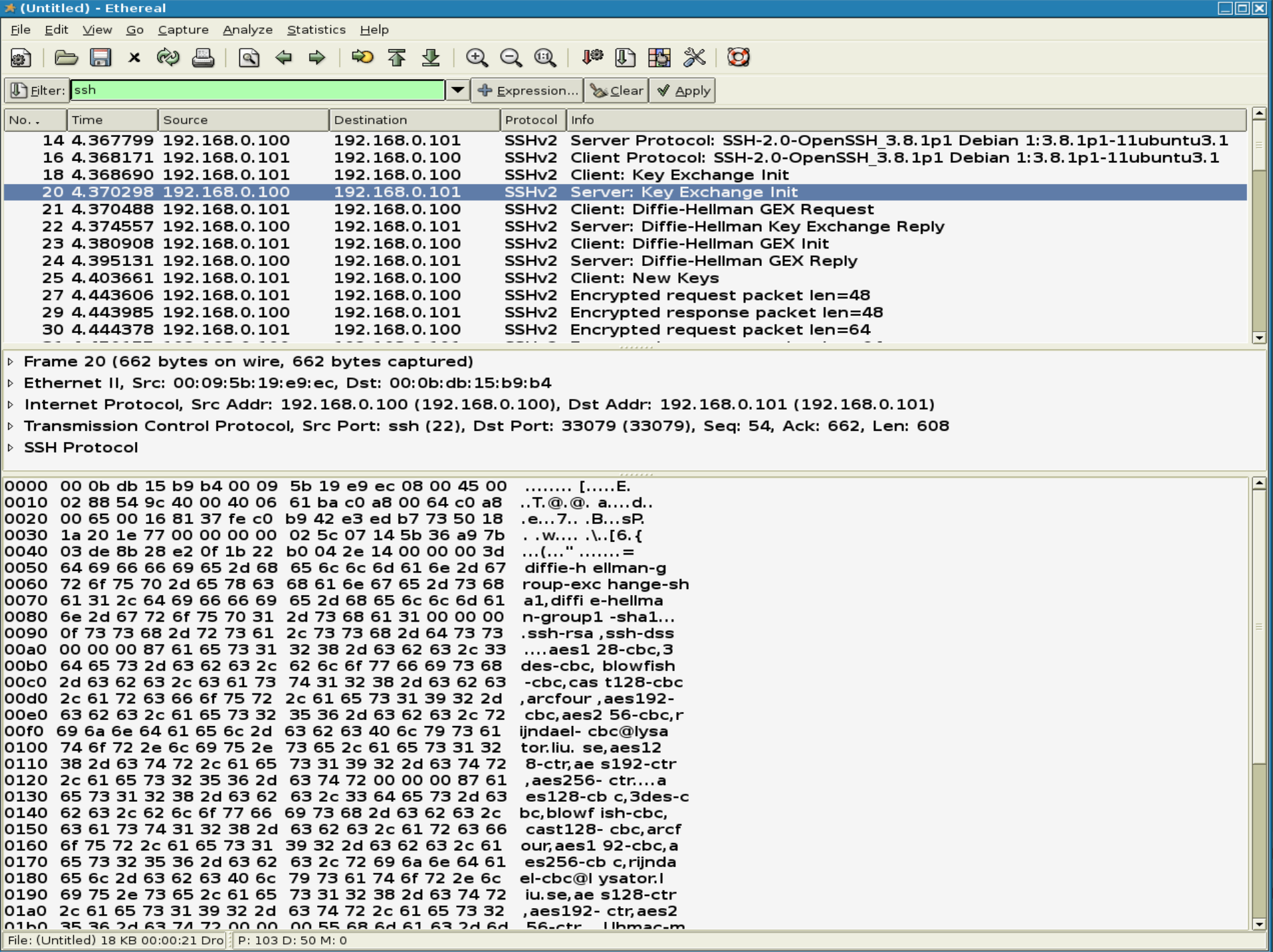
- sa aspekta sigurnosti korištenje telneta za prijavu i rad na udaljenom računalu ne preporučuje se
- postoje programi, tzv. *snifferi* koji su u stanju pratiti i na praćenom mediju iz niza TCP paketa koji pripadaju jednoj telnet seansi rekonstruirati istu u cjelosti, saznati lozinku korisnika koji koristi telnet, kao i sve ostale lozinke koje je korisnik možda upotrijebio tijekom telnet veze
- dobra zaštita je korištenje enkripcije – protokola **ssh**

ssh (*Secure SHell*)

- sigurna zamjena za telnet, koristi enkripciju podataka
- Uveden 1995
- ssh poslužitelj na udaljenom računalu standardizirano koristi port 22
- koristi se kriptografija javnog ključa za sigurnu identifikaciju udaljenog računala na koje se prijavljujemo, kako bi ne bi mogao desiti slučaj preusmjeravanja prometa na lažno računalo koje glumi ono na koje se prijavljujemo
- prilikom svake prijave koristi se novi kriptografski ključ, a razmjenjuje se putem Diffie-Helman algoritma za javnu razmjenu ključeva
- koriste se MAC – *message authentication codes,* kriptografsko potpisivanje paketa kako bi se onemogućilo njihovo “snimanje” i naknadno reproduciranje

ssh

- Pojednostavljeni postupak prijave izgleda ovako:
  - Klijent se pomoću ssh programa spaja na port 22 na odredišnom računalu gdje sluša ssh poslužitelj
  - Generira se ključ koji će biti korišten za šifriranje te ssh veze, te se pomoću Diffie-Helman algoritma sigurno razmjenjuje između klijenta i poslužitelja, usput sigurno identificirajući poslužitelj
  - Nakon što je dogovoren ključ, sva daljnja komunikacija odvija se šifrirano jednim od raspoloživih algoritama za enkripciju, najčešće Rijndael-128 (AES)
- Sintaksa korištenja: znamo od ranije
- evo kako zapravo izgleda razmjena ključeva, uhvaćeno *snifferom*


(Untitled) - Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter: ssh
Expression... Clear Apply
No. Time Source Destination Protocol Info
14 4.367799 192.168.0.100 192.168.0.101 SSHv2 Server Protocol: SSH-2.0-OpenSSH_3.8.1p1 Debian 1:3.8.1p1-11ubuntu3.1
16 4.368171 192.168.0.101 192.168.0.100 SSHv2 Client Protocol: SSH-2.0-OpenSSH_3.8.1p1 Debian 1:3.8.1p1-11ubuntu3.1
18 4.368690 192.168.0.101 192.168.0.100 SSHv2 Client: Key Exchange Init
20 4.370298 192.168.0.100 192.168.0.101 SSHv2 Server: Key Exchange Init
21 4.370488 192.168.0.101 192.168.0.100 SSHv2 Client: Diffie-Hellman GEX Request
22 4.374557 192.168.0.100 192.168.0.101 SSHv2 Server: Diffie-Hellman Key Exchange Reply
23 4.380908 192.168.0.101 192.168.0.100 SSHv2 Client: Diffie-Hellman GEX Init
24 4.395131 192.168.0.100 192.168.0.101 SSHv2 Server: Diffie-Hellman GEX Reply
25 4.403661 192.168.0.101 192.168.0.100 SSHv2 Client: New Keys
27 4.443606 192.168.0.101 192.168.0.100 SSHv2 Encrypted request packet len=48
29 4.443985 192.168.0.100 192.168.0.101 SSHv2 Encrypted response packet len=48
30 4.444378 192.168.0.101 192.168.0.100 SSHv2 Encrypted request packet len=64
Frame 20 (662 bytes on wire, 662 bytes captured)
Ethernet II, Src: 00:09:5b:19:e9:ec, Dst: 00:0b:db:15:b9:b4
Internet Protocol, Src Addr: 192.168.0.100 (192.168.0.100), Dst Addr: 192.168.0.101 (192.168.0.101)
Transmission Control Protocol, Src Port: ssh (22), Dst Port: 33079 (33079), Seq: 54, Ack: 662, Len: 608
SSH Protocol
0000 00 0b db 15 b9 b4 00 09 5b 19 e9 ec 08 00 45 00 .... [.....E.
0010 02 88 54 9c 40 00 40 06 61 ba c0 a8 00 64 c0 a8 ..T.@.@. a....d..
0020 00 65 00 16 81 37 fe c0 b9 42 e3 ed b7 73 50 18 .e...7.. .B...sP.
0030 1a 20 1e 77 00 00 00 00 02 5c 07 14 5b 36 a9 7b . .w.... .\..[6.{
0040 03 de 8b 28 e2 0f 1b 22 b0 04 2e 14 00 00 00 3d ...(..." ....=
0050 64 69 66 66 69 65 2d 68 65 6c 6c 6d 61 6e 2d 67 diffie-h ellman-g
0060 72 6f 75 70 2d 65 78 63 68 61 6e 67 65 2d 73 68 roup-exc hange-sh
0070 61 31 2c 64 69 66 66 69 65 2d 68 65 6c 6c 6d 61 a1,diffi e-hellma
0080 6e 2d 67 72 6f 75 70 31 2d 73 68 61 31 00 00 00 n-group1 -sha1...
0090 0f 73 73 68 2d 72 73 61 2c 73 73 68 2d 64 73 73 .ssh-rsa ,ssh-dss
00a0 00 00 00 87 61 65 73 31 32 38 2d 63 62 63 2c 33 ....aes1 28-cbc,3
00b0 64 65 73 2d 63 62 63 2c 62 6c 6f 77 66 69 73 68 des-cbc, blowfish
00c0 2d 63 62 63 2c 63 61 73 74 31 32 38 2d 63 62 63 -cbc,cas t128-cbc
00d0 2c 61 72 63 66 6f 75 72 2c 61 65 73 31 39 32 2d ,arcfour ,aes192-
00e0 63 62 63 2c 61 65 73 32 35 36 2d 63 62 63 2c 72 cbc,aes2 56-cbc,r
00f0 69 6a 6e 64 61 65 6c 2d 63 62 63 40 6c 79 73 61 ijndael- cbc@lysa
0100 74 6f 72 2e 6c 69 75 2e 73 65 2c 61 65 73 31 32 tor.liu. se,aes12
0110 38 2d 63 74 72 2c 61 65 73 31 39 32 2d 63 74 72 8-ctr,ae s192-ctr
0120 2c 61 65 73 32 35 36 2d 63 74 72 00 00 00 87 61 ,aes256- ctr....a
0130 65 73 31 32 38 2d 63 62 63 2c 33 64 65 73 2d 63 es128-cb c,3des-c
0140 62 63 2c 62 6c 6f 77 66 69 73 68 2d 63 62 63 2c bc,blowf ish-cbc,
0150 63 61 73 74 31 32 38 2d 63 62 63 2c 61 72 63 66 cast128- cbc,arcf
0160 6f 75 72 2c 61 65 73 31 39 32 2d 63 62 63 2c 61 our,aes1 92-cbc,a
0170 65 73 32 35 36 2d 63 62 63 2c 72 69 6a 6e 64 61 es256-cb c,rijnda
0180 65 6c 2d 63 62 63 40 6c 79 73 61 74 6f 72 2e 6c el-cbc@l ysator.l
0190 69 75 2e 73 65 2c 61 65 73 31 32 38 2d 63 74 72 iu.se,ae s128-ctr
01a0 2c 61 65 73 31 39 32 2d 63 74 72 2c 61 65 73 32 ,aes192- ctr,aes2
01b0 35 36 2d 63 74 72 00 00 00 55 68 6d 61 63 2d 6d 56-ctr... Uhmac-m
File: (Untitled) 18 KB 00:00:21 Dro
P: 103 D: 50 M: 0

Zadatak 1

- pomoću naredbe “man ssh” proučite neke opcije ssh klijenta te ih isprobajte
- Ustanovite što rade opcije -v, -vv, -q, -p, -o, -c, -C (neke uz odgovarajuće parametre)
- pokrenite ssh naredbu za spajanje na *student.math.hr* uz uključenu kompresiju podataka, forsirani ssh2 protokol, aes128-cbc način šifriranja i opširno ispisivanje dijagnostičkih poruka prilikom spajanja

ssh – napredne opcije

- Spomenut ćemo još neke napredne opcije koje nam omogućuje SSH:
  - “tuneliranje” X windows aplikacija kroz ssh vezu, tj. lokalni prikaz X windows aplikacija koje se zapravo izvršavaju na računalu na koje smo se prijavili ssh vezom
  - “tuneliranje” portova, putem enkriptirane ssh veze moguće je napraviti siguran “tunel” npr između udaljenog porta 23 (telnet) računala na koje smo se prijavili putem ssh protokola i lokalnog porta npr. 10023, te ćemo telnetom na lokalni port 10023 zapravo dobiti port 23 na udaljenom računalu, kroz sigurnu SSH vezu
  - sftp podsustav kao sigurna alternativa ftp-u

ftp (*File Transfer Protocol*)

- protokol iz aplikacijskog sloja, koristi se za prijenos datoteka između računala baziranih na TCP/IP mrežama
- FTP poslužitelj sluša na portu 21, koristi TCP protokol
- također nesiguran protokol koji ne koristi enkripciju
- zamjena je sftp (*SSH File Transfer Protocol*)

Način rada FTP-a

- FTP poslužitelj sluša na portu 21
- klijent uspostavlja konekciju i prijavljuje se korisničkim imenom i lozinkom
- uspostavlja se kontrolni kanal, putem kojeg klijent može izlistavati direktorije i datoteke, zatražiti prijenos neke datoteke ili postaviti neku datoteku na FTP poslužitelj
- Za prijenos podataka otvara se novi, podatkovni kanal kojim se prenose podaci
- postoji nekoliko vrsta rada s obzirom na otvaranje podatkovnog kanala i sam prijenos podataka:
    - **Aktivni mod** – FTP klijent otvara slučajno odabrani port veći od 1023, šalje poslužitelju broj tog porta i čeka TCP vezu od strane FTP poslužitelja sa polaznog porta broj 21 na taj odabrani port, te nakon toga počinje sam prijenos podataka

Način rada FTP-a

- **Pasivni mod** – kad FTP klijent nije u mogućnosti primiti dolaznu TCP konekciju na slučajno odabran visoki port npr zbog *firewalla*
- tada FTP server otvara visoki port (>1023) i dojavljuje svoju IP adresu i broj tog porta na koji se klijent treba spojiti nakon čega počinje prijenos podataka
- Pasivni mod se uključuje naredbom **PASV** nakon čega server vraća nešto poput “227 Entering Passive Mode (192,84,105,1,4,1) gdje su prva 4 broja IP adresa, a zadnja dva port, i to na način da se prvi broj od ta dva množi sa 256 i pribraja drugom (pa je ovdje port 4*256+1 = 1025)
- **Napredni pasivni mod –** isto kao pasivni mod, ali server šalje samo broj porta a pretpostavlja se da je IP isti

Anonimni FTP

- Još uvijek vrlo korišten način distribucije raznih datoteka i softvera je korištenje anonimnog ftp pristupa, gdje se korisnik logira korisničkim imenom *anonymous,* a kao lozinku koristi svoju e-mail adresu.
- **Zadatak 2**: prijavite se ftp klijentom na poslužitelj ftp.funet.fi kao anonimni korisnik, pronađite datoteku README i preuzmite ju na svoje računalo, te u njoj pronađite koju je konfiguraciju poslužitelj ftp.funet.fi imao 1990-e godine
- (znači: prvo se logirati na *studenta.* Zatim ukucati *ftp* ftp.funet.fi, za username upisati *anonymous*, za password bilo što, *ascii*, *get README*, *bye*)

SMTP

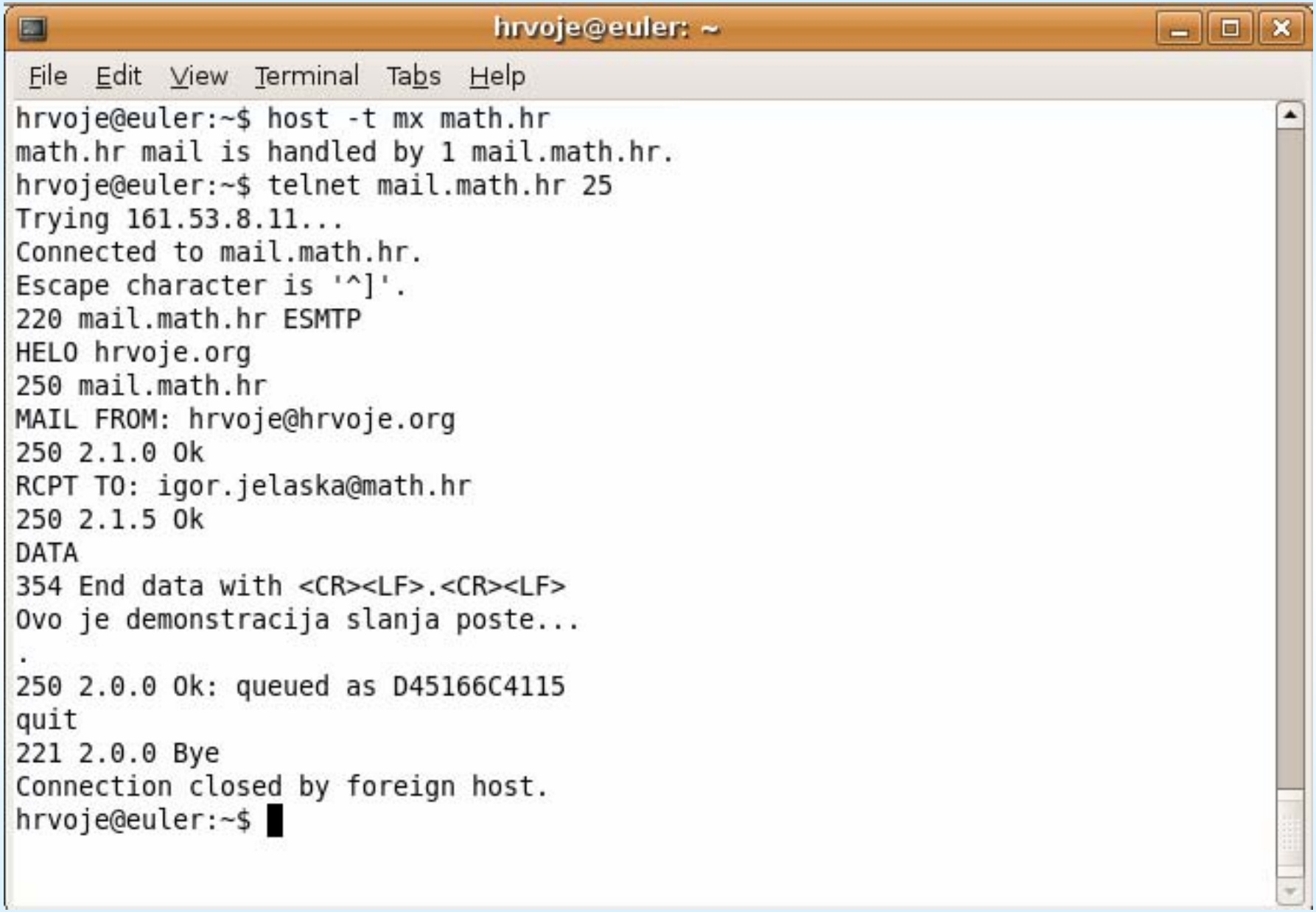
- SMTP – *Simple Mail Transfer Protocol* je standard za prijenos i isporuku elektroničke pošte među računalima na Internetu. Kad god pošaljete poruku elektroničke pošte, bilo preko nekog *webmaila* ili npr MS Outlooka, razmjena pošte između polaznog i odredišnog poslužitelja odvija se putem SMTP protokola
- S obzirom da je protokol napisan isključivo kao tekstualni za razmjenu ASCII datoteka, otkad se pojavila potreba za razmjenom programa i ostalih binarnih datoteka koristi se enkodiranje 8-bitnog sadržaja u 7-bitni ASCII oblik (MIME)
- SMTP poslužitelj standardno sluša na portu 25

Relaying

- Kad korisnik s osobnog računala želi poslati e-mail poruku, koristi se tzv. postupak proslijeđivanja *(relaying)* – korisnik se spaja na port 25 gdje sluša **SMTP poslužitelj**, te ako je korisnikova IP adresa u dozvoljenom rasponu adresa za koje se dopušta proslijeđivanje pošte poruka (izbacuju se one s kojih dolazi spam) poruka se preuzima od strane poslužitelja koji ju dalje šalje na odredište
- **Kako se zna koje računalo prima e-mail za koju domenu?**
  - Svaka domena ima u svom **DNS sustavu** (iduće vježbe) tzv. MX zapis (mail exchanger) koji pokazuje na računalo koje prima mail za tu domenu, dakle kad želimo poslati e-mail na igor.jelaska@math.hr, SMTP poslužitelj prvo pita DNS poslužitelj koje računalo prima mail za math.hr domenu, dobija odgovor da je to mail.math.hr, spaja se na njega i isporučuje mu poruku na slijedeći način:

Isporuka pošte

- želimo na adresu e-pošte igor.jelaska@math.hr **ručno** (da naučimo kako to server radi) isporučiti neku poruku. To radimo na slijedeći način:
- Naredbom telnet spojimo se na port 25 poslužitelja mail.math.hr
- Server odgovara “220 mail.math.hr ESMTP”
- Šaljemo naredbu “HELO *hostname*” gdje je *hostname* ime našeg računala, kako bi se predstavili odredišnom poslužitelju
- Server odgovara “250 mail.math.hr”
- Šaljemo naredbu “MAIL FROM: naša@email.adresa” (pošiljatelj)
- Server odgovara “250 2.1.0 Ok”
- Šaljemo naredbu “RCPT TO: igor.jelaska@math.hr” (primatelj)
- Server odgovara “250 2.1.5 Ok”
- Šaljemo naredbu “DATA”
- Server odgovara kako završiti poruku
- Pišemo tijelo poruke, kad smo gotovi odemo u novi red, napišemo točku i stisnemo enter
- Pogledajmo kako to izgleda na primjeru:


hrvoje@euler: ~
File Edit View Terminal Tabs Help
hrvoje@euler:~$ host -t mx math.hr
math.hr mail is handled by 1 mail.math.hr.
hrvoje@euler:~$ telnet mail.math.hr 25
Trying 161.53.8.11...
Connected to mail.math.hr.
Escape character is '^]'.
220 mail.math.hr ESMTP
HELO hrvoje.org
250 mail.math.hr
MAIL FROM: hrvoje@hrvoje.org
250 2.1.0 Ok
RCPT TO: igor.jelaska@math.hr
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Ovo je demonstracija slanja poste...
.
250 2.0.0 Ok: queued as D45166C4115
quit
221 2.0.0 Bye
Connection closed by foreign host.
hrvoje@euler:~$

Zadatak 4

- Koristeći isključivo naredbu telnet i gornji primjer kao referencu, direktno se spajajući na odredišni e-mail poslužitelj pošaljite na e-mail asistenta neku poruku (napomene: prvo se spojite na *studenta*. Kada šaljete s fakulteta, onda umijesto *mail.math.hr* i *student.math.hr* treba pisati *mail* i *student* u koracima *telnet* i *HELO*)
- Saznajte koje računalo prima e-mail za domene iskon.hr, predsjednik.hr, gmail.com, yahoo.com (vidi naredbu *host* na prethodnom slideu)
- Što mislite zbog čega gmail.com ima više različitih računala koje primaju e-mail?