Fakultet elektrotehnike i računarstva
Preddiplomski studij Računarstvo

Komunikacijske mreže
Akademska godina 2020./2021.

2. domaća zadaća

Zadatak 1

Znamenke a, b i c u tekstu zadatka odnose se na posljednje tri znamenke Vašeg JMBAG-a (npr. 0036512abc). Ako je bilo koja od znamenki Vašeg JMBAG-a 0 (nula), zamijenite ju sa znamenkom 5.

Klijent K uspostavlja TCP-vezu s poslužiteljem P, a odmah po uspostavi veze poslužitelj šalje podatke duljine a000 (a tisuća) okteta. Za vrijeme trajanja veze ne dolazi do gubitaka podataka, potvrde se šalju bez odgađanja, a parametri prozora ne mijenjaju. Nakon što primi svih a000 okteta, klijent inicira raskid TCP-veze. Pretpostavite da je inicijalni apsolutni slijedni broj (prije uspostave veze) na poslužitelju P_{init} = b000, da je inicijalni apsolutni slijedni broj na klijentu K_{init} = c000, da je maksimalna veličina segmenta na poslužitelju i na klijentu *MSS* = 2000 okteta, da je veličina prozora primatelja na poslužitelju i na klijentu $rwnd_P = rwnd_K$ = 1500 okteta te da prosječno obilazno vrijeme RTT iznosi 2 vremenske jedinice.

(a) Koja je vrijednost polja *Broj u nizu* u prvom segmentu s podacima (koji šalje poslužitelj nakon uspostave veze)?
(b) Koja je vrijednost polja *Broj potvrde* u prvoj potvrdi poslanoj od strane klijenta (nakon uspostave veze)?
(c) Koliko je podataka poslano od strane poslužitelja u zadnjem segmentu (prije raskida veze)?

Očekivano rješenje zadatka je tekstualna datoteka, pohranjena kao **<JMBAG>.txt**, koja sadrži tri retka u kojima su zapisani isključivo brojevi (odgovori na pitanja pod (a), (b) i (c)).

Zadatak 2

Znamenke a, b, c, d, e i f u tekstu zadatka odnose se na posljednjih šest znamenki Vašeg JMBAG-a (npr. 0036abcdef). Ako je bilo koja od znamenki Vašeg JMBAG-a 0 (nula), zamijenite ju sa znamenkom 5.

Preuzmite topologiju *dz2_kommre.imn* s poveznice http://public.tel.fer.hr/km/dz2/dz2_kommre.imn i otvorite ju u programu IMUNES. Potrebno je stvoriti 3 podmreže (A, B i C) dodavanjem računala (čvorova tipa *PC*) i komutatora (čvorova tipa *LAN switch*) na usmjeritelje routerA, routerB i routerC. Podmreže A, B i C moraju se sastojati od po jednog čvora *LAN switch* i po dva čvora *PC*.

Adrese podmreža A, B i C su zadane kao:
A: 2*(1+ab) . 2*cd . 2*ef . 0 / 24
B: 2*(1+cd) . 2*ef . 2*ab . 0 / 26
C: 2*(1+ef) . 2*ab . 2*cd . 0 / 28

1.

a) b000+1=6001
b) b000+1+1500=7501
c) a000 % 1500= 1500
+ iznimno, ako je a000 % 1500 == 0, onda 1500.

https://www.youtube.com/watch?v=xMtP5ZB3wSk

365, a=3, b=6, c=5

2*(1+52) . 2*43 . 2*65 . 0 / 24 == 106.86.130.0/24 A

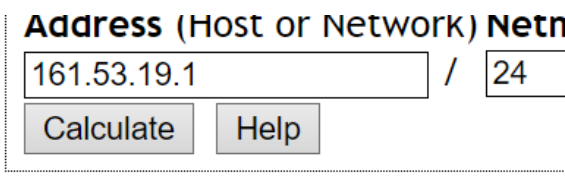
a=5,b=2,c=4,d=3,e=6,f=5

```
Address:   106.86.130.0
Netmask:   255.255.255.0 = 24
Wildcard:  0.0.0.255
=>
Network:   106.86.130.0/24
Broadcast: 106.86.130.255
HostMin:   106.86.130.1
HostMax:   106.86.130.254
Hosts/Net: 254
```

2*(1+43) . 2*65 . 2*52 . 0 / 26 == 88.130.104.0/26 B

```
Address:   88.130.104.0
Netmask:   255.255.255.192 = 26
Wildcard:  0.0.0.63
=>
Network:   88.130.104.0/26
Broadcast: 88.130.104.63
HostMin:   88.130.104.1
HostMax:   88.130.104.62
Hosts/Net: 62
```

HOST




```
Address:   161.53.19.1
Netmask:   255.255.255.0 = 24
Wildcard:  0.0.0.255
=>
Network:   161.53.19.0/24
Broadcast: 161.53.19.255
HostMin:   161.53.19.1
HostMax:   161.53.19.254
Hosts/Net: 254
```

2*(1+65) . 2*65 . 2*52 . 0 / 28 == 132.130.104.0/28 C

```
Address:   132.130.104.0
Netmask:   255.255.255.240 = 28
Wildcard:  0.0.0.15
=>
Network:   132.130.104.0/28
Broadcast: 132.130.104.15
HostMin:   132.130.104.1
HostMax:   132.130.104.14
Hosts/Net: 14
```

anaa Vjerojatno su krivo postavljena(ili uopce nisu) sucelja na ruterima. Ne mogu ovako zakljuciti bez da vidim konfiguracije rutera. Neka opca formula kako bi trebale izgledati konfiguracije:

```
adresa druge mreze | sucelje preko kojeg se dolazi do te mreze
```

npr. za routerA(adresa podmreze A je npr. 152.102.80.0/24)
recimo da su adrese ostalih podmreza 161.52.102.0/24(podmreza B), 152.54.102.0/26(podmreza C) i 161.53.19.0/24(podmreza HOST)

U konfiguraciju routerA se upisuju:

```
161.52.102.0/24 10.0.2.1
152.54.102.0/26 10.0.2.1
161.53.19.0/24 10.0.2.1
```

U konfiguracije racunala podmreze A se upisuje defaultni usmjeritelj(u ovom slucaju 152.102.80.1(router A)):

```
0.0.0.0/0 152.102.80.1
```

Taj postupak se treba napraviti sa sve ostale usmjeritelje u mreži (routerB, routerC, routerX) i sva ostala racunala i onda ce se moci pingati svako racunalo. Nisam 100% siguran jel to ispravan nacin, ali radi ispravno.

NAPOMENA: Ako je a=5, b=6 (ilustrativni primjer), onda je ab=56, a **NE** ab=30.

U tako definiranim podmrežama, konfigurirajte IP adrese i podrazumijevane usmjeritelje dodanim čvorovima, te statičke rute na **svakom** od usmjeritelja u topologiji, kako bi svi Vaši čvorovi bili dostupni sa svih ostalih čvorova u mreži (naredbom *ping*). IP adresa usmjeritelja mora biti prva valjana adresa u podmreži, a IP adrese svakog od dodanih čvorova moraju biti zadnje valjane adrese u podmreži (time isključujući adresu razašiljanja, *broadcast*).

Očekivano rješenje zadatka je topologija, .imn datoteka, pohranjena kao **<JMBAG>.imn**.

Zadatak 3

Korištenjem kriptografije javnog ključa potrebno je digitalno potpisati datoteku <JMBAG>.imn iz prethodnog zadatka. Datoteka može biti u izvornom ili promijenjenom obliku (tj. nije nužno riješiti Zadatak 2 za rješavanje Zadatka 3).

Za potpisivanje datoteke koristite naredbu *openssl*. Naredba *openssl* dostupna je na bilo kojem operacijskom sustavu, ali preporučuje se korištenje na virtualnom stroju IMUNES. Samostalno istražite naredbu *openssl* te pomoću nje generirajte RSA par ključeva potrebnih za potpisivanje datoteke. Stvorite digitalni potpis datoteke <JMBAG>.imn, te pritom zapišite sve Vaše korake u tekstualni izvještaj (<JMBAG>_log.txt).

Provjera vašeg rješenja izvodit će se na način da se pokrene naredba:

```
$ openssl rsautl -verify -inkey <kljuc> -pubin -keyform PEM -in <digitalni_potpis>
```

Naredba kao izlaz mora dati liniju nalik ovoj:

```
SHA256(0036443921.imn)=78dc88c7f2f93f7e4f0687e3afbf6e8646c38642bb87e1849b85291143b7a138
```

Ta vrijednost mora se poklapati sa sažetkom datoteke, dobivenim naredbom:

```
$ sha256 0036542199.imn
SHA256(0036443921.imn)=78dc88c7f2f93f7e4f0687e3afbf6e8646c38642bb87e1849b85291143b7a138
```

Očekivano rješenje zadatka su datoteke s kojima će se moći provjeriti digitalni potpis (**<JMBAG>.imn, <JMBAG>.pem** i **<JMBAG>.sig**) te kratki opis koraka dobivanja istih (**<JMBAG>_log.txt**).

Dosta sam se mučila za 3. zadatak jer mi nije radilo čak ni na @cosko10 način, pa da podijelim svoj način sad kad mi konačno radi, ako nekome još treba.
Instaliraj OpenSSL na windows, jebeš immunes
generiraj privatni ključ:
OpenSSL>genrsa -out private.pem 2048
generiraj javni ključ:
OpenSSL>rsa -in private.pem -outform PEM -pubout -out <JMBAG>.pem
generiraj potpis
OpenSSL>dgst -sha256 -sign private.pem -out <JMBAG>.sig <JMBAG>.imn
provjera
dgst -sha256 -verify <JMBAG>.pem -signature <JMBAG>.sig <JMBAG>.imn
Ako valja, console će izbaciti “Verified OK!”
UPOZORENJE: način provjere iz uputa za dz ne radi uz ovaj način jer potpis nije u bazi 64 pa ćete dobiti kinesko pismo, a baza 64 više ne stane u ključ. Ipak, ako želite dobiti potpis u obliku baze 64 za asistente, iskoristite ovu naredbu:
OpenSSL> enc -base64 -in <JMBAG>.sig -out <JMBAG>base64.sig

Ako nekom treba za 3. zad, meni ovako radi
1.openssl dgst -sha256 jmbag.imn > hash
2.generirate public i private key (public key je ovdje u jmbag.pem):
    openssl genrsa -out private.pem 2048
    openssl rsa -in private.pem -outform PEM -pubout -out jbmag.pem
3.signate hash pomoci private keya
    openssl rsautl -sign -inkey privatekey.pem -keyform PEM -in hash > jmbag.sig
4. openssl rsautl -verify -inkey jmbag.pem -pubin -keyform PEM -in jmbag.sig
    ovo zadnje je za provjeru i to je t

Ako nekom treba za 3. zad, meni ovako radi

1. openssl dgst -sha256 jmbag.imn > hash
2. generirate public i private key (public key je ovdje u jmbag.pem):
   openssl genrsa -out private.pem 2048
   openssl rsa -in private.pem -outform PEM -pubout -out jbmag.pem
3. signate hash pomoci private keya
   openssl rsautl -sign -inkey privatekey.pem -keyform PEM -in hash > jmbag.sig
4. openssl rsautl -verify -inkey jmbag.pem -pubin -keyform PEM -in jmbag.sig
   ovo zadnje je za provjeru i to je to