

*Jesenji semestar, 2016/17*

*PREDMET: IT331 Računarske mreže i komunikacija*

*PROJEKTNI ZADATAK:*

Analiza POP3 protokola

Ime i prezime: **Nemanja Kuzmanovic**

Broj indeksa: **2851**

Profesor: **Miroslava Raspopović**

Asistent: **Miljan Marković**

Beograd 2017.

Sadržaj

[1. Cilj projektnog zadatka: 3](#_Toc472391443)

[1.1 Analiza POP3 protokola 3](#_Toc472391444)

[2. Uvod 3](#_Toc472391445)

[2.1 Šta je POP3? 3](#_Toc472391446)

[4. Analiza 6](#_Toc472391447)

[4.1 Način analize 6](#_Toc472391448)

[4.2 Podizanje potrebnih alata 6](#_Toc472391449)

[4.3 Analiza - Wireshark 11](#_Toc472391450)

[4.4 Analiza poruke i rekonstrukcija 17](#_Toc472391451)

[4.5 Simulacija komunikacije CMD 20](#_Toc472391452)

[5. Bezbednost 22](#_Toc472391453)

[5.1 Nedostatci 22](#_Toc472391454)

[5.2 Zaštita i bezbednost 22](#_Toc472391455)

[5. Zaključak 23](#_Toc472391456)

[Reference: 25](#_Toc472391457)

# 1. Cilj projektnog zadatka:

## 1.1 Analiza POP3 protokola

Cilj: Analiza POP3 protokola sa akcentom na bezbednosni segment. Opis osobina POP3 protokola i komande koje POP3 koristi, koje ce se i prakticno prikazati kroz eksperimentalnu analizu. Analiza saobracaja prenesena putem POP3 protokola. Od interesa je bezbednost prenosa njegovih paketa, odnosno detaljna analiza mogucnosti rekonstruisanja poruke iz uhvacenih paketa, uz predlog kako je moguce obezbediti bezbedan prenos.

# 2. Uvod

## 2.1 Šta je POP3?

POP ili Post Office Protocol (verzija 3)je protokol koji služi za prijem pošte sa e-mail servera. Na taj način što se sva pošta sa servera preuzima i čuva na lokalnom uređaju i sve radnje koje izvršimo nad primljenim porukama se vrše na našem uređaju.

Neki e-mail servisi koji podržavaju POP3, ukoliko bi sa jednog računara proverili poštu i taj računar preuzeo poruke sa servera, bi izgubio poruke, i ne bi se više nalazile na serveru, pošto ih jednom preuzmemo one se brišu sa samog servera, pa bi naš pristup porukama bio ograničen za taj konkretni uređaj na kome smo obavili proveru pošte.

Međutim, danas, u većini slučajeva možemo izbeći ovu situaciju tako što bi u podešavanjima samog mail klienta izabrali odgovarajuću opciju, kojom bi se kopije poruka i dalje čuvale na samom serveru, pa bi nam bio omogućen pristup i sa ostalih uređaja samim porukama na serveru. Naravno ovde postoji boljka, a sastoji se u tome da će se poruke preuzete na jednom uređaju u pročitane na istom na ostalim uređajima i dalje prikazivati kao nove, tj. nepročitane.

Uz to, podešavanja jednog mail klienta će se čuvati samo na uređaju na kome je klient podešen, kao što će pravljenje foldera, markera za sortiranje i samih filtera, takođe biti vezano za lokalni uređaj, jer POP protokol svu poštu sa servera smešta u inbox.

Takođe, sva nova pošta sa prilozima, koji mogu biti neki fajlovi ili nešto slično što sadrži veličinu fajla kao prilog koji se šalje, će se ponovo preuzimati na lokalni uređaj pri svakoj proveri pošte, što naravno nikako nije praktično, pogotovo ukoliko imamo ograničenu ili sporu internet konekciju.

POP3 protokol koristi port 110 za komunikaciju sa mail serverom. A TCP vezu.

Sesija počinje konektovanjem POP3 klijenta na port 110 POP3 servera, pomoću „3 way handshake“-a, server na to šalje pozdravnu poruku, pa potom sledi razmena komandi i odgovora. Klijent se mora identifikovati komandama **USER**<korisničko ime> i **PASS**<lozinka>. Nakon ovoga se izvršava par komandi, slično kao i kod SMTP, a posle svake sledi odgovor POP3 servera. Sesija se zatvara slanjem QUIT komande od strane POP3 klijenta.

POP3 komande:

***USER <name>*** - korisničko ime za ovaj mail server

***PASS <password>*** - šifra

***QUIT*** – kraj sesije u kojoj server iz faze transakcije prelazi u fazu

ažuriranja

***STAT*** – broj i ukupna veličina svih poruka

***LIST <message#>*** - spisak sadržaja poštanskog sandučeta, uključujući

dužinu svake poruke u posebnom redu

***RETR message#*** - preuzmi obeležene poruke

***DELE message#*** - brisanje obeleženih poruka

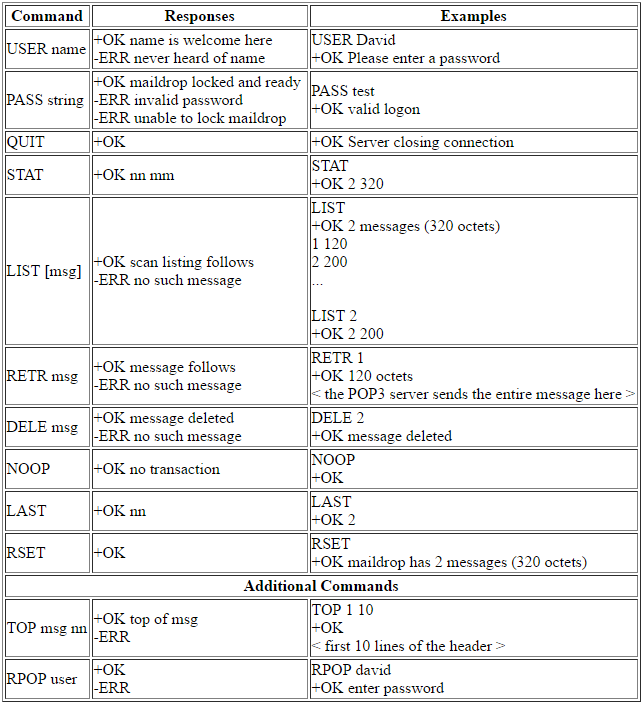
***UIDL*** – preuzimanje numeričke liste svih poruka i njihove jedinstvene

identifikacione brojeve, ili jedinsteveni ID za određenu poruku

***NOOP*** - drži konekciju otvorenom

***RSET*** – resetuje poštansko sanduče i vraća izbrisane poruke.

Server koristi dve komande najčesće a to su **+OK** gde vraća da se zahtevana komanda ili radnja izvršila i da je registrovao kao i **+ERR** gde vraća da nije uspeo da primi ili registruje odgovarajuću akciju.



*Slika 1 – primer pop3 komandi*

Redosled stanja:

Autorizacija – Kada server izrazi spremnost da prihvata komande, klijent treba da se identifikuje da bi mu se omogućio pristup poštanskom sandučetu.

Transakcija – U ovom stanju klijentu je dozvoljeno da izvrši operacije nad sandučetom. Uobičajene operacije su: preuzimanje spiska poruka, preuzimanje samih poruka, označavanje preuzetih poruka spremnih za brisanje.

Ažuriranje – Kada je klijent završio sa operacijama i pošalje QUIT komandu automatski nastupa ovo stanje. Poruke, u prethodnom koraku, označene za brisanje sada se brišu i TCP veza se prekida

# 4. Analiza

## 4.1 Način analize

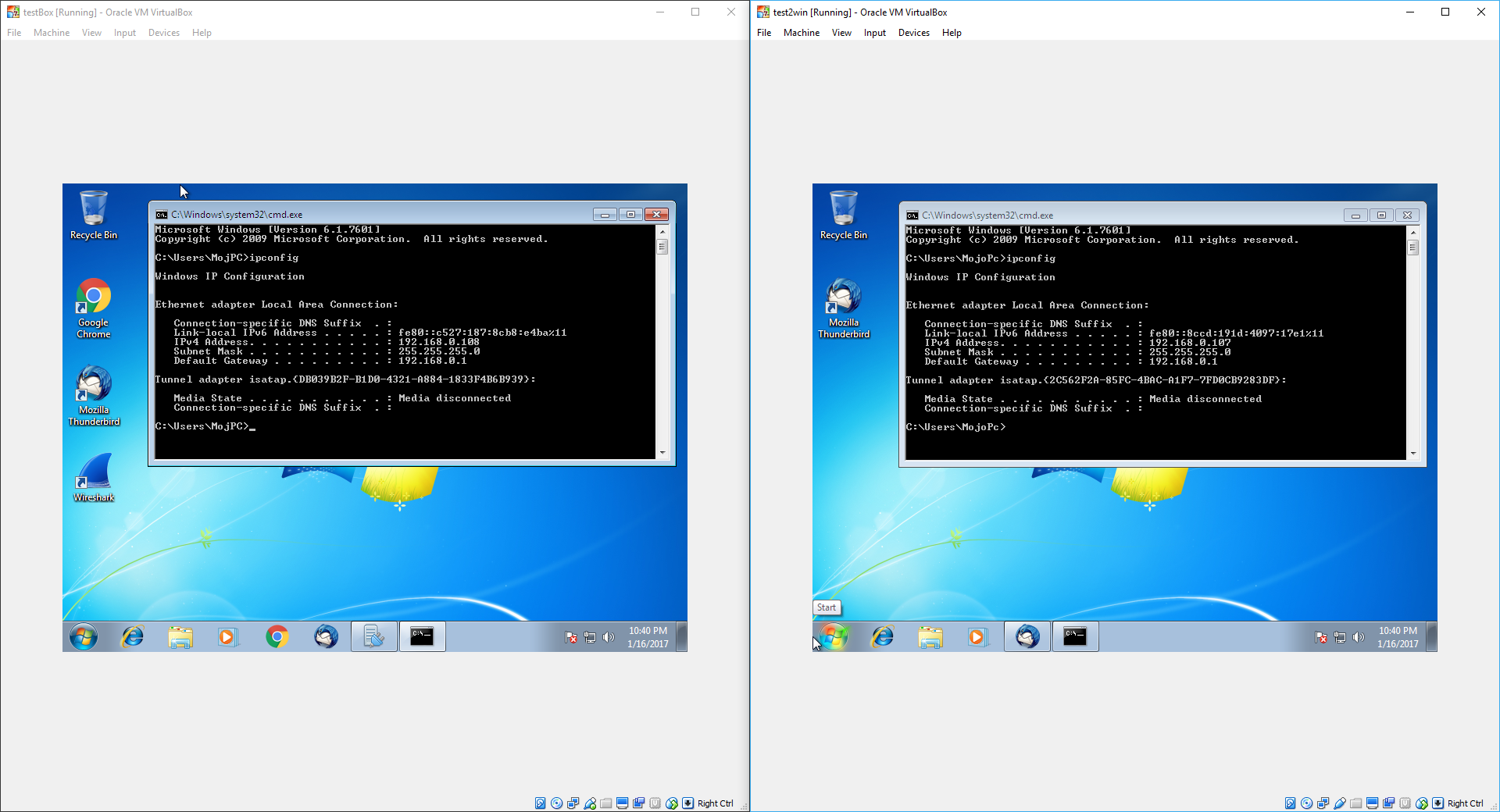
Kako bi se analizirao sam POP3 protokol, kao i bezbednost, i sve što ide uz to, pored osnovnih alata kao što su Wireshark, CMD, Virtual Box, Mozzila Thunderbird, koristiće se i besplatni softver za podizanje mail servera – hMailServer

Ideja je da se na dve virtuelne mašine podigne neko „desktop“ okruženje koje bi simuliralo na jednoj strani korisnika a na drugom sam mail server. Potom bi se na VM (Vrituelna Mašina) na kojoj je hostovan hMailServer pokrenuo alat Wireshark, i izvršila bi se sama analiza paketa kao i mogućnosti rekonstrukcije poruke kao i bezbednosti POP3 protokola.

Kao što je već navedeno pored ovih alata, koristiće se mail klient pod nazivom Thunderbird, i konfigurisaće se tako da se poveže na server koji se nalazi na drugoj VM uz pomoć „Bridged Adapter“ konekcije kako bi se simulirala neka mreža, na kojoj se konfigurisala različita IP i MAC adresa od druge VM kako ne bi došlo do konflikta.

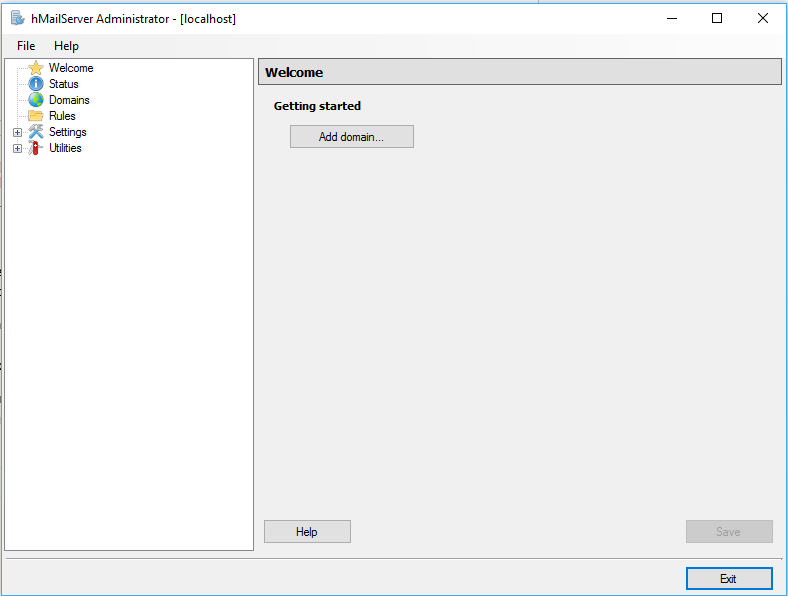
## 4.2 Podizanje potrebnih alata

Prvenstveno će se podići kao što je navedeno dve VM, jedna će simulirati mail hosta a druga klijenta, u kome će klijent radi jednostavnosti i lakše analize sam sa sobom razmeniti mail.

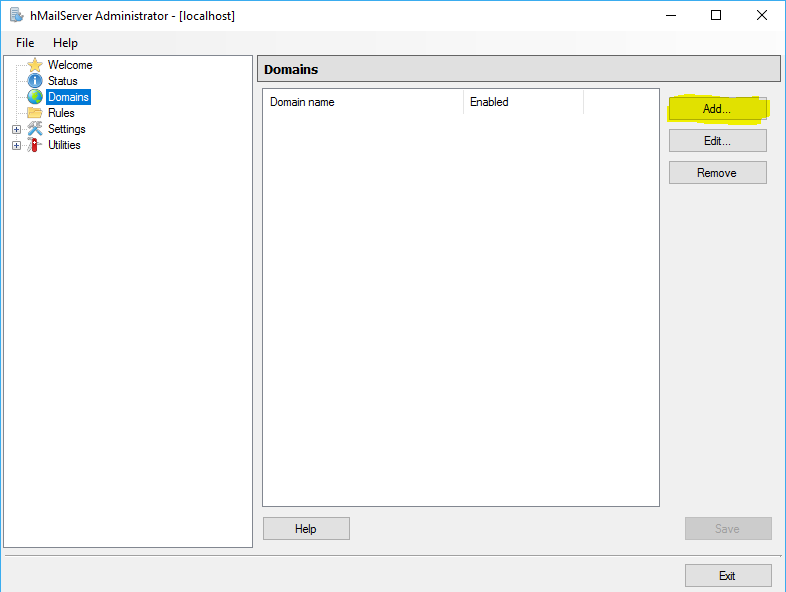


*Slika 2 – Podizanje Virtuelnih mašina sa alatima*

Kada su se podigle mašine, pokrećemo hMailServer alat kako bi jednostavno i bez troškova napravili simulaciju domena u lokalnoj mreži. U narednih nekoliko slika biće detaljno prikazano pokretanje svih alata potrebnih za ovu analizu.

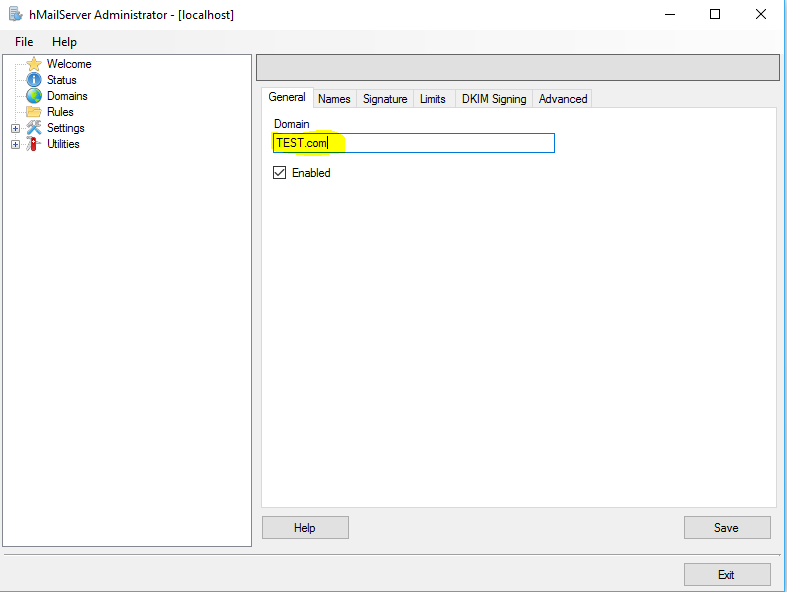


*Slika 3 – hMailServer alat*



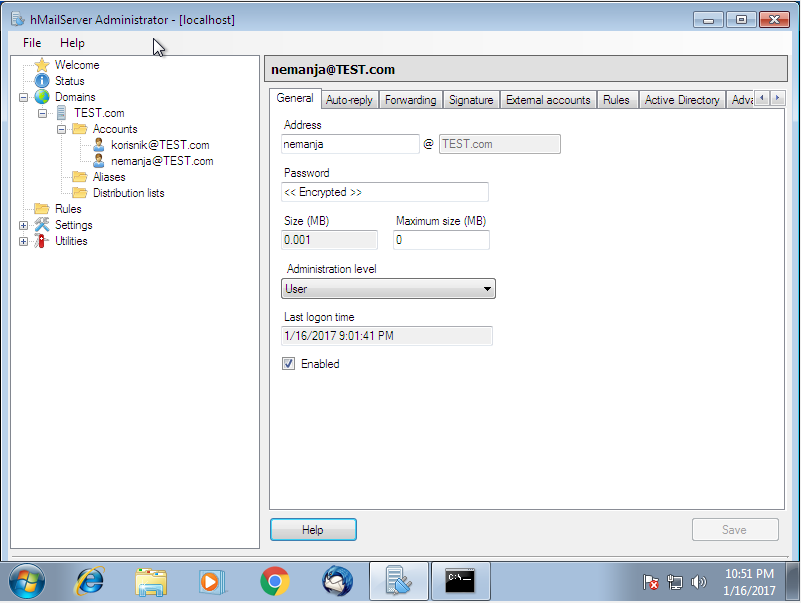
*Slika 4 – Dodavanje domena*

Dakle, kao što je prikazano na slikama, korisnički interfejs je veoma jednostavan i prijateljski ka korisnicima, i ne bi trebalo doći do većih problema kod same konfiguracije domena kao i naloga korisnika.



*Slika 5 – Domen TEST.com*

U ovom koraku pristupamo pravljenju samog domena, obzirom da se ne radi analiza samog softvera neće se dublje ulaziti u softver, ali kao što je prikazano na samoj slici, softver sadrži popriličan broj dodatnih opcija, i veoma je fleksibilan u pogledu konfiguracije samog domena.

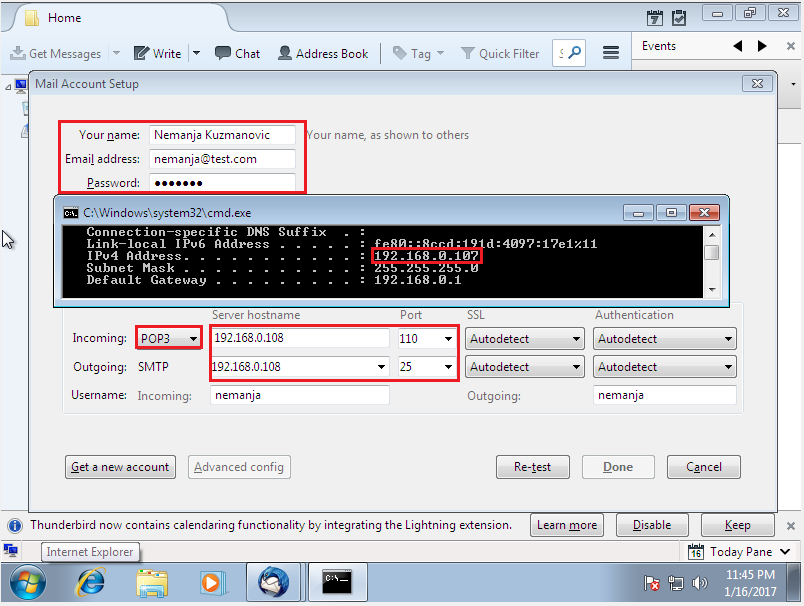


*Slika 6 – dodavanje naloga*

Dalje pristupamo dodavanju korisničkih naloga koji će biti upotrebljeni u testnoj analizi.   
Prilikom samog kreiranja naloga došlo je do upozorenja od strane klienta da lozinka koja je „enkriptovana“ nije jaka, i izašao je prozor koji nas je pored upozorenja pitao da li želimo da nastavimo. Podatci koji su korišćeni su:

[nemanja@TEST.com](mailto:nemanja@TEST.com) sa šifrom: nemanja

Ovi podatci će dalje koristiti kada se budu analizirali paketi u komunikaciji klijenta i servera.



*Slika 7 – Konfiguracija klienta*

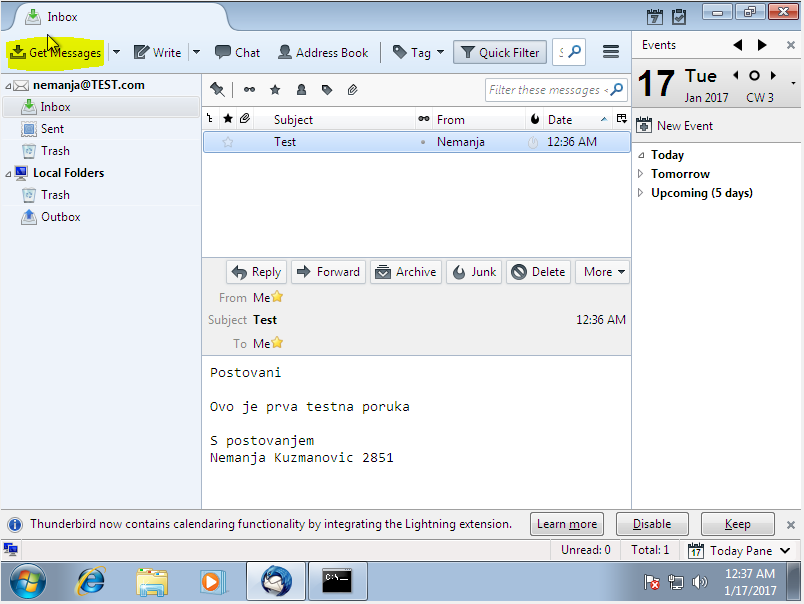
Pri konfiguraciji klienta (Thunderbird) koji se koristi u analizi, pored unosa osnovnih podataka o korisniku kao sto su „username“, „email adress“, i „password“, mora se ručno podesiti konekcija iz razloga što se analiza vrši na lokalnoj mreži a ne registrovanom domenu, kojeg bi takođe, svakako, trebalo ručno konfigurisati.

U ovom slučaju unose se adrese VM na kojoj je podignut mail server. Ta adresa se dobija u Command Prompt-u unosom komande ipconfig. Gleda se IPv4 adresa koja je u ovom slučaju: 192.168.0.108

Pored unešene adrese, mora se izabrati POP3 protokol kao primajući protokol, kao i portovi na kome primajući i odlazni protokoli rade, a to su POP3 – 110 i SMTP – 25

Naredni korak je da se pokrene „Wireshark“ i da se pošalje e-mail sa jednog korisnika. A potom pošto se radi o POP protokolu da se preuzme ta poruka sa servera.

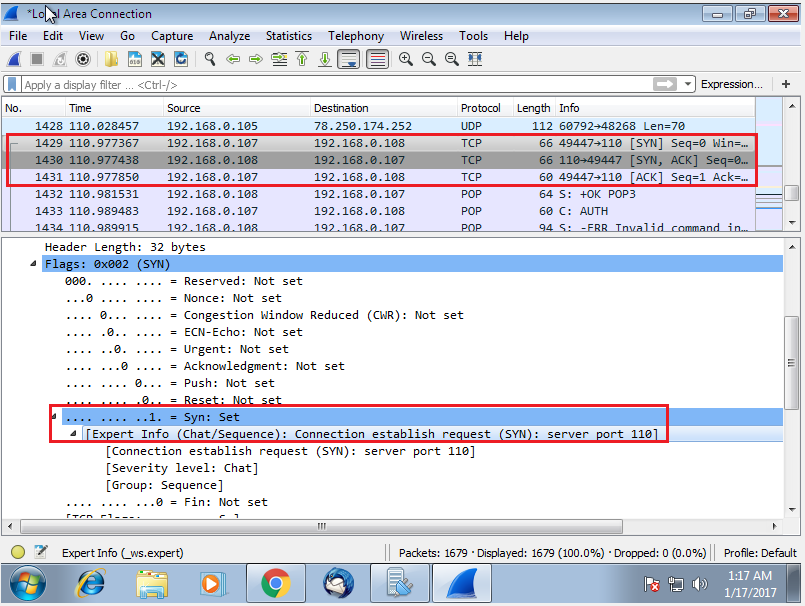
*Slika 8 – Slanje eMaila*



Kada se mail pošalje, on dolazi do servera na kome se čuva sve do trenutka dok određeni korisnik ne preuzme poruke, u ovom slučaju po pritisku na taster „Get Messages“ dostavljena je baš ona poruka koja je poslata.

## 4.3 Analiza - Wireshark

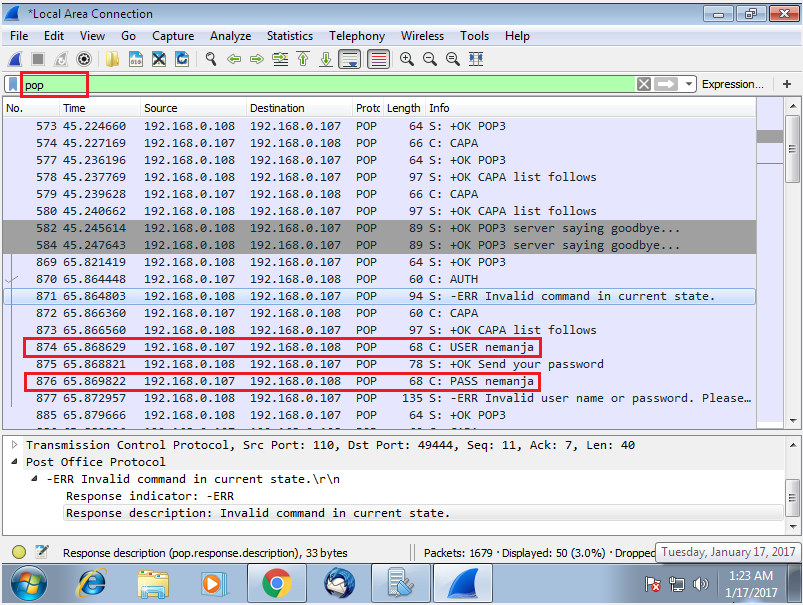
Sada će se pristupiti detaljnoj analizi paketa pokupljenih u samom Wireshark-u.



*Slika 9 – Analiza Wireshark paketa*

Na samom početku, pre nego što u sam filter ubacimo „POP“, možemo primetiti standardnu „3 way handshake“ konekciju. Gde klijent šalje zahtev za prepoznavanje, a server mu odgovara potukom da je primljeno i vraća mu ACK (Acknowledgement) paket.

Pored ovoga se može primetiti kao što je već navedeno da POP radi sa portom 110.



*Slika 10 – Analiza*

Kada se klijent poveže sa serverom sa kojim želi da komunicira u vidu komandi, tačnije, kada želi da dobije određene informacije oni direktno preko istih tih komandi i komuniciraju.  
Na samom početku posle prve „3 way handshake“ konekcije, server vraca +OK poruku kao znak da je prepoznao određeni oređaj koji zahteva konekciju sa njim.

Posle uvodne konekcije javlja se komanda CAPA koja pita da li server ima (capability) sposobnost da porčita UIDL komandu,za preuzimanje numeričke liste svih poruka i njihove jedinstvene identifikacione brojeve, ili jedinsteveni ID za određenu poruku. Što u sledećem koraku server i potvrđuje sa +OK POP3 komandom, u suprotnom najverovatnije bi vratio –ERR kao naznaku na grešku. UIDL je bitan za samog korisnika kako bi sigurnije, efikasnije i bolje identifikovao pojedinačne poruke.

Zatim, klijent ponovo šalje serveru još nekoliko zahteva za proveru komandi, kako bi bio znao koje parametre kasnije prosleđuje.

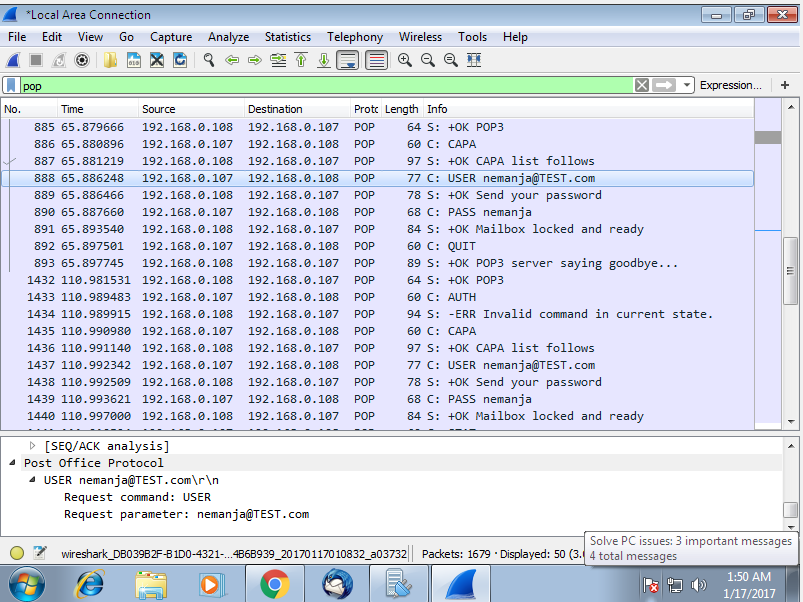
Server potom odgovara, i potvrđuje da ima sve sposobnosti koje klijent zahteva, i šalje odjavnu pozdravnu poruku „Goodbye“.

Sledeći korak, kada se korisnik preko klienta uloguje, je da se proveri njegova autektikacija, da li se njegove informacije nalaze u bazi servera. Međutim u ovom primeru se desava nešto zanimljivo, u najmanju ruku, naime, server nema mogućnost autentikacije, iz razloga što nije konfigurisan za takvu radnju, postavljena je osnova radi analize, koja će se u nastavku više obezbediti, i uporediti razlike i mane ovako laički konfigurisanog servera.

Još jednom korisnik proverava sposobnosti servera za određene komande posle odbačaja autentikacije.  
Napomena:  
Server jeste registrovao korisnika i posle provere u bazi mu omogućio da se autentikuje kao autentični korisnik koji ima mogućnost da komunicira sa serverom, međutim ova autentikacija je ili slabo konfigurisana od samog mail servera, ili od samog klienta alata preko koga koristimo usluge servera. Trenutno nisu postavljeni nikakvi vidovi ni autentikacije ni enkripcije, što je vrlo verovatno da je klient očekivao.

Nakon ovoga, klient šalje korisnikov „USER“ na sta server odgovara da je našao odgovarajuće poklapanje, usled dolazi do slanja „PASS“ takođe od strane klienta koji pokušava da pristupi serveru. Prvenstveno dolazi do –ERR poruke, koja je u ovom slučaju vraćena iz razloga što se u postavkama samog klienta nije navelo puno korisničko ime nego samo deo, što ne predstavlja grešku nego server upozorava da se u bazi nalaze još neki podatci koji nisu dopunjeni, ali da su oni primarni +OK.

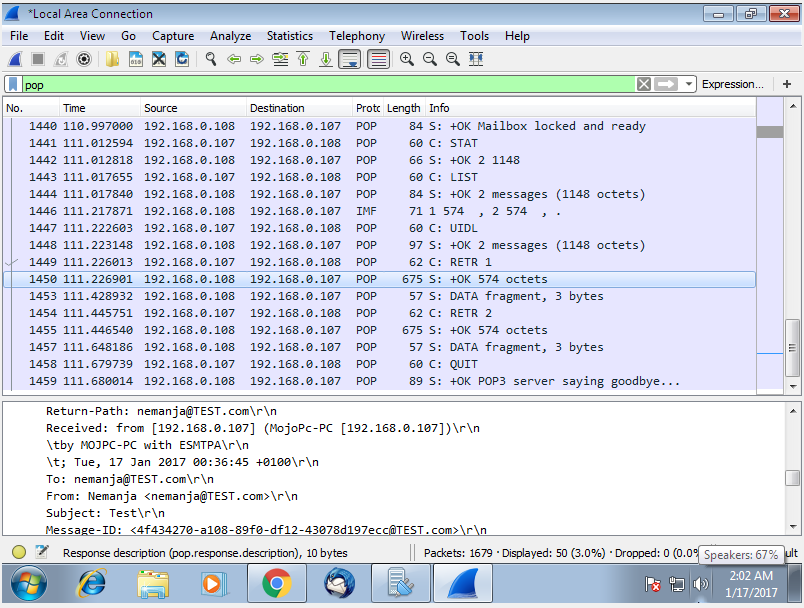
Sada nailazimo na veliki problem. Naime, USER od strane klijenta je poslat i ukoliko se desi da se na putu klienta i servera nađe maliciozan korisnik ili korisnik koji osluškuje pakete, on će mogu da pristupi samom sadržaju USER-a i to više nego očigledno. Još veći problem je što klient odmah posle šalje i otvorenu šifru bez ikakve enkripcije ili pokušaja da se sakrije, nego kao običan tekst, koji je opet, više nego očigledan. Ovako loše konfigurisan klient koji ne šalje enkriptovanu šifru kao i server koji ne zahteva enkriptovanu šifru, moze lako biti upotrebljen u maliciozne radnje, do kojih najčešće dolazi dobro poznatom metodom pribavljanja ovakvih informacija nazvanom „Man in the middle“. Gde se uređaj na putu od klienta do servera „ni kriv ni dužan“ nalazi i dobija jako senzitivne informacije.



*Slika 11 – Analiza paketa*

Komunikacija se nastavlja posle poklapanja rezultata o korisniku i njegove šifre, sada korisnik šalje punu email adresu kako bi prešao na sledeći korak, u ovom slučaju [nemanja@TEST.com](mailto:nemanja@TEST.com), nakon čega server ponovo traži od korisnika da mu klient pošalje sifru. Što se još jednom i dešava klient šalje otvorenu šifru bez ikakve zaštite: nemanja. Na prvi pogled ovo izgleda katastrofalno jer i klient i server daju „na tacni“ podatke o korisniku i njegovu šifru, i kroz ne bezbednu mrežu šalju dalje, gde bilo gde i bilo koji postavljeni osluškivač može iskoristiti ovu nemarnost.

Konačno se korisnik loguje u klient i šalje se komanda QUIT za prekid sesije transakcije, i sada server prelazi u stanje ažuriranja.



*Slika 12 – Analiza*

Kada se osnovni koraci autentikacije, konekcije, i provere korisnika završe, oni se ponavljaju na nekom vremenskom intervalu, kako bi server znao da je klient još uvek tu i da treba da zadrži trenutnu sesiju, jer ukoliko se ne bi proveravalo i sesija se držala non stop, definitivno bi jako brzo došlo do preopterećenja samog servera pa mu niko ne bi mogao pristupiti dok se ne bi očistila radna memorija, što nikako nema smisla nikada raditi.

Klient šalje serveru komandu STAT u kojoj traži status svoga sandučeta, da li ima i koliki broj poruka na serveru.

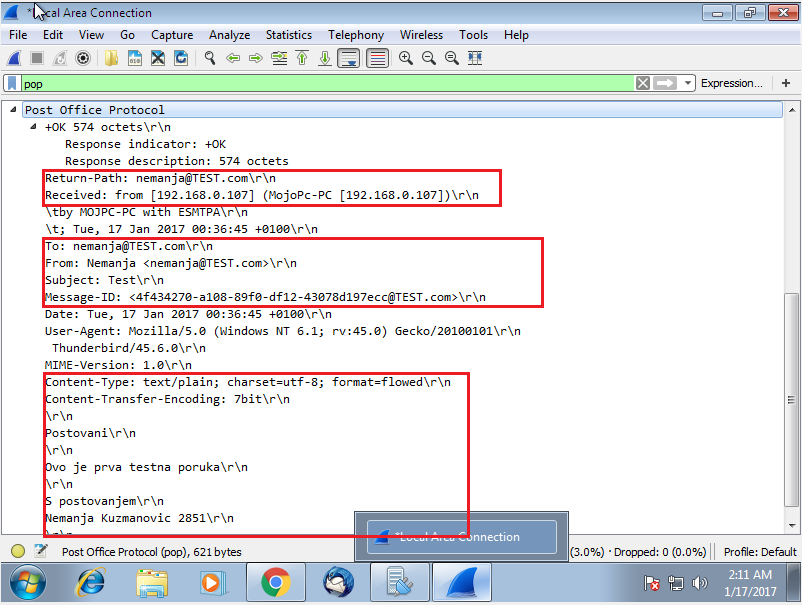
Server dalje prihvata ovu komandu odgovara sa +OK i prosleđuje broj poruka koje čuva za određenog klijenta kao i veličinu tih poruka.

U ovom slučaju nalaze se dve testne poruke, svaka veličine po 574 okteta.

Dakle, kada korisnik pritisne određeno „Get Messages“ dugme, klient šalje komandu stat u kome server vraća broj poruka, nakon čega klient zahteva UIDL, kako bi dobio jedinstvene indentifikatore svake poruke posebno, da bi posle komande klienta RETR server klientu poslao čitavo telo poruke, kao i informacije koje sadrži o njoj.

Na primeru se vide 2 RETR komande, iz razloga koji je naveden a to je da su dve testne poruke poslate. Kada dođe do završetka primanja ponovo se šalje QUIT za prekid sesije i server se odjavljuje.

## 4.4 Analiza poruke i rekonstrukcija

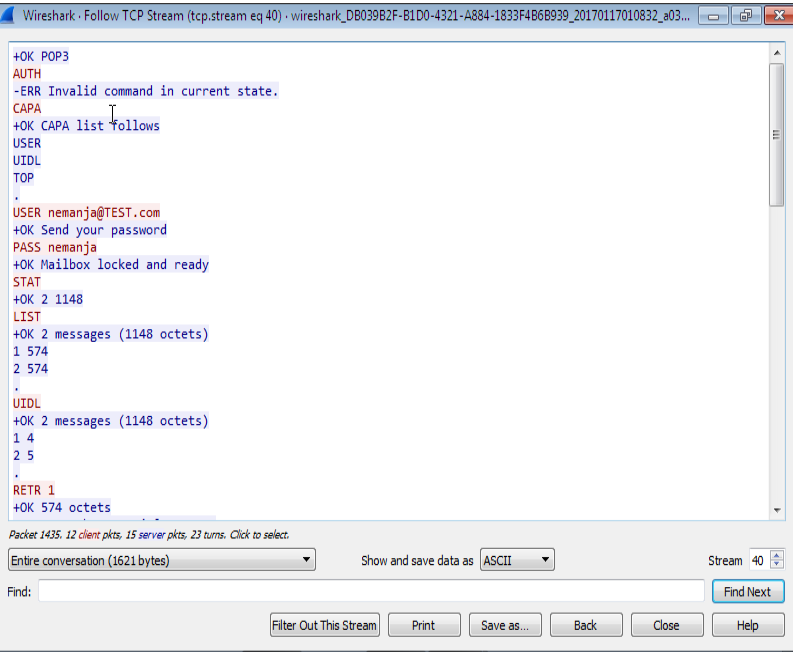


*Slika 13 – poruka*

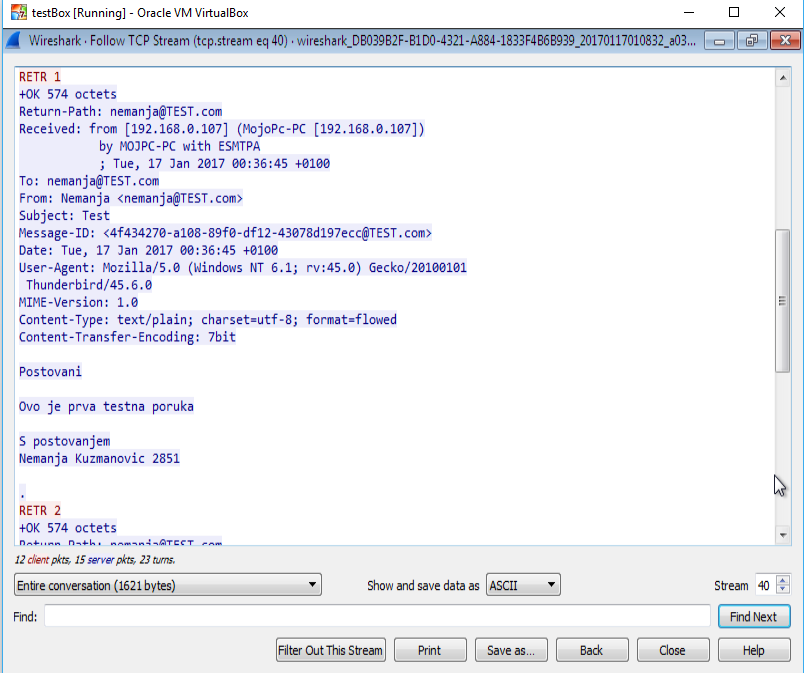
Ukoliko bi detaljnije otvorili fragment primljene poruke kroz našu mrežnu karticu, možemo videti detaljne podatke.  
Na samom početku nalazi se kome je poruka upućena od strane servera pod nazivom „Return-Path“. Potom imamo IP adresu sa koje dolazi ista ta poruka, kao i naziv mašine sa koje je poruka poslata (MojoPc-PC). Naravno, vreme, a pored toga i samo telo poruke.  
Prvenstveno imamo kome je nemenjena poruka, od koga se šalje, naslov poruke, jedinstveni identifikator (Message ID), kao i informacije koji klient je korišćen kako bi se poruka prosledila – Thunderbird.

Konačno, imamo i samo telo poruke, koje jasno možemo pročitati.

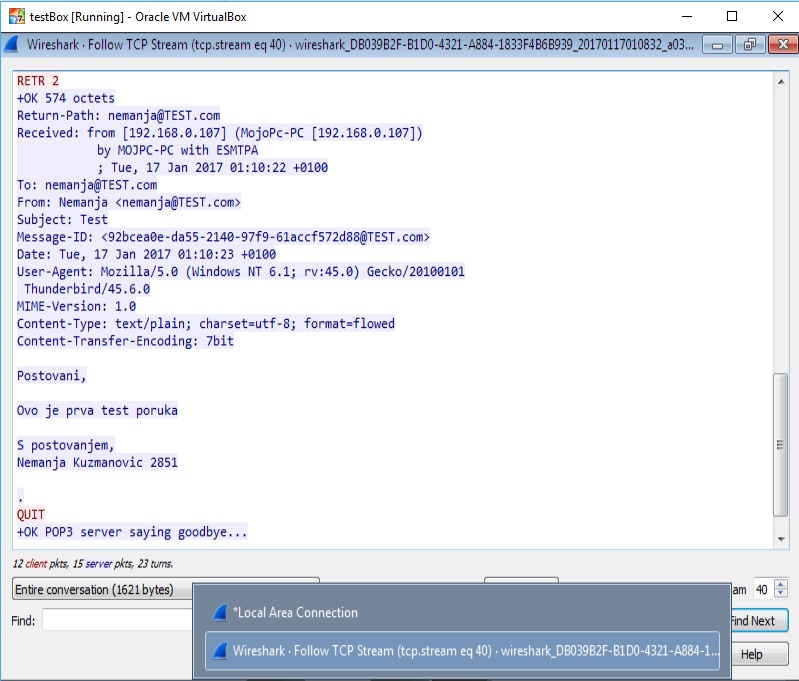
Sva komunikacija između klienta i servera se može još vizuelno bolje ispratiti ukoliko zapratimo TCP stream, tako što se pritisne desni klik na bilo koji od POP paketa pa Follow a zatim TCP Stream, što bi izgledalo:



*Slika 14 – TCP stream 1*



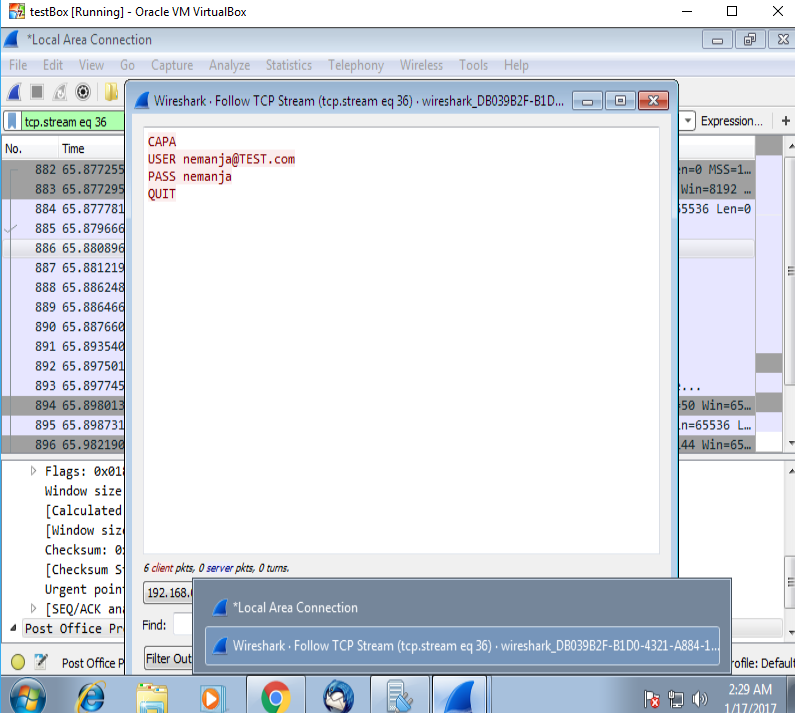
*Slika 15 – TCP stream 2*



*Slika 16 – TCP stream 3*

U prethodnim slikama se očigledno i veoma jednostavno može uočiti pored same komunikacije komandama i same poruke koju razmenjuju klient i server, što nikako ne bi smelo da se dozvoli!

Napomena:   
Paketi se naravno šalju SMTP protokolom, od servera ka klientu, one su enkriptovane.

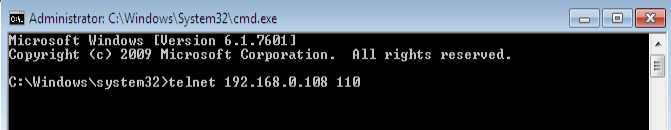


*Slika 17 – poruke koje šalje klient*

Naravno moguće je sortirati samo poruke koje klient šalje, pa njih posle simulirati direktno preko Command Prompta.

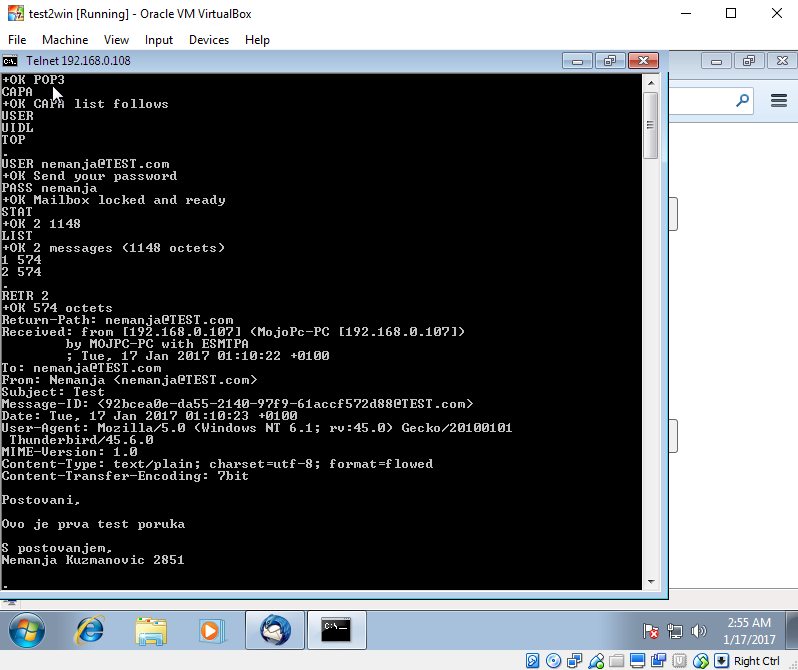
## 4.5 Simulacija komunikacije CMD

Za ovaj eksperiment i analizu koristiće se CMD (Command Prompt) kao i dodatak Windows-a telnet. Simulirace se komunikacija između klienta i servera direktnim unošenjem komandi POP-a u CMD.



*Slika 18 – CMD komunikacija*

Prvenstveno se unosi komanda telnet zatim adresa samog servera, koja je u ovom slučaju u formatu IP-a na lokalnoj mreži, ukoliko postoji pravi server sa zakupljenim pravim domenom, moguće je ukucati domen adresu i pristupiti. Koristi se protokol 110 za pristup.



*Slika 19 – CMD komunikacija*

Prvenstveno je kucana komanda CAPA koja bez problema vaća +OK, kao i sposobnosti koje podržava, a to su USER UIDL i TOP.

Kada se pošalje komanda USER [nemanja@TEST.com](mailto:nemanja@TEST.com) ponovo se vraća poruka +OK što znači da je sve u redu i da nije došlo nidokakve greške. Potom se uz PASS šalje lozinka: nemanja ,što server prihvata, i vraća ponovo +OK poruku uz dodatnu poruku da je Mailbox zaključan i spreman za dalje manipulisanje.

Komandom STAT ponovo dobijamo +OK poruku da je sve prošlo bez problema a server nam vraća status o porukama na samom serveru kao i veličinu, komandom LIST izlistavaju se poruke,a komandom RETR 2 se poruka broj dva prikazuje u komandnom prozoru, potpuno istih osobina koje su se nalazile i u Wireshark-u.

# 5. Bezbednost

## 5.1 Nedostatci

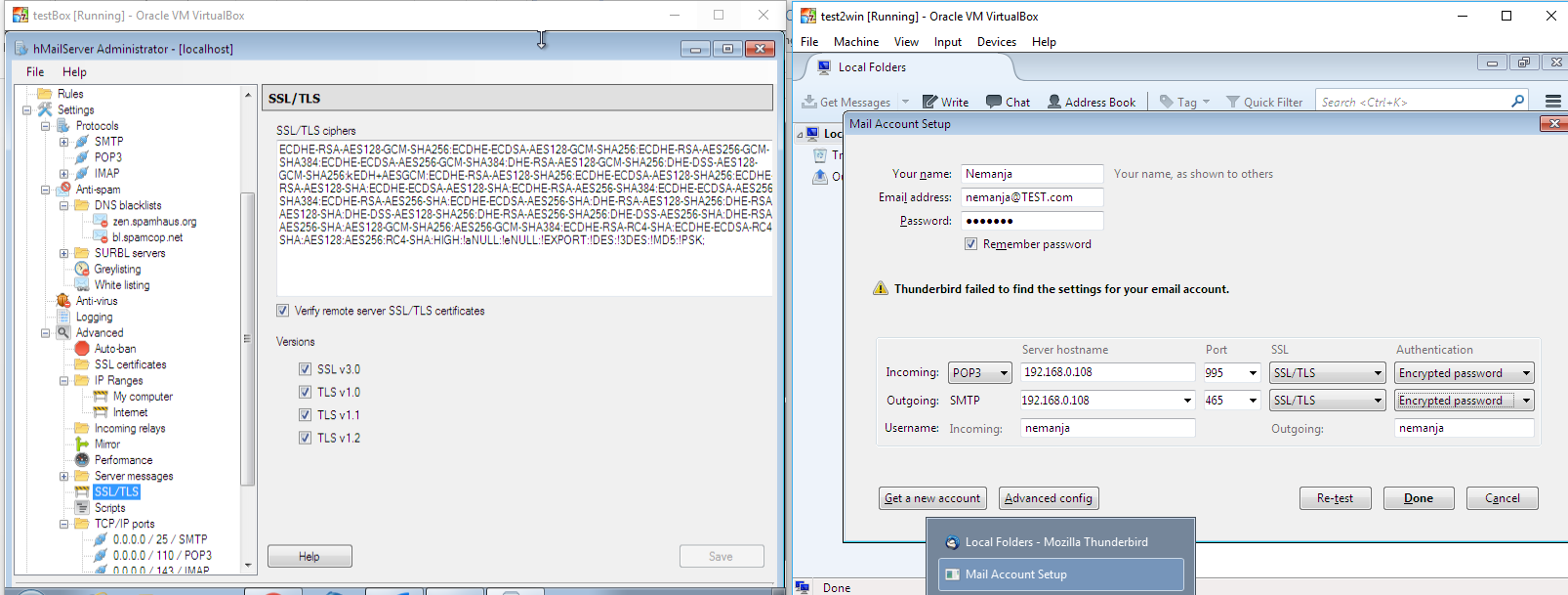
Kao što smo videli u navedenim eksperimentalnim primerima, ukoliko u lokalnoj mreži postavimo mail server i klient bez dodatnog obezbeđenja, doći će do ogromnih posledica. Svi naši korisnici će biti izloženi napadima od ostalih korisnika na mreži, a to je milionski broj. Sam pop3 nema bezbednosne mehanizme jer je protokol za komunikaciju između klienta i samog servera tačnije nam služi sa povlačenje poruka sa servera, pa kao što je već navođeno lako se može desiti da neko ko osluškuje pakete negde između servera i klienta (man in the middle) može doći do naših senzitivnih informacija, što nije dobro. Pored ovih nedostataka ne iskustvo korisnika koji konfiguriše mail server može takođe dovesti do ogromnih nedostataka u sistemu, što je nedopustivo.

Poruke poslate iz klienta i preuzete preko pop3 protokola se kao što je već prikazano mogu lako rekonstruisati, ukoliko nije implementirana zaštita i oslanja se na zaštitu samog protokola(?).

## 5.2 Zaštita i bezbednost

Dakle, nije dovoljno samo konfigurisati klient i mail server i osloniti se na bezbednost protokola, jer sam protokol ne sadrži nikakve bezbednosne mehanizme.

Prva i osnovna i najveća zaštita bi uvek trebalo da bude sertifikat (SSL). On će nam omogućiti da se bezbedno povežemo sa serverom kroz nebezbednu mrežu bez mogućnosti da nam bilo ko oslušne šta se nalazi u poruci. Tako na primer najpouzdaniji i najupotrebljivaniji SSL sertifikat štiti i nas i korisnika, enkriptovajući sav saobraćaj između nas i servera. Naravno, moguće je koristiti i TLS takođe jako dobar sertifikat. Međutim, postoji i ona najjeftinija a najnebezbednija varijanta, a to je, ručno pisani. Za razliku od zvaničnih sertifikata koji su možemo reći sa sigurnošću sigurni, za ručno pisane se ne može znati postoje li propusti, pa takođe svaki put kada se koristi ručno pisani sertifikat, pretraživač ili klient će izbaciti upozoravajuću poruku da se takav sertifikat koristi i da se pažljivo šalju senzitivne informacije, baš zbog toga što zvanično nije utvrđena validnost istog. Sertifikati se prvenstveno podešavaju na samom mail serveru, a potom i u klientu koji koristimo za pristup serveru, tako što mu naglasimo koji bezbednosni port se koristi kao i koja vrsta sertifikata/enkripcije.



*Slika 20 – bezbednosni sertifikati*

Iako bi sertifikat bio najpametnija i najbolja vrsta zaštite postoji i zaštita enkriptovanja samih informacija kao što su email ili lozinka, ili oba. Međutim ovo i nije baš najidealniji način zaštite, jer iako mi enkriptujemo naše podatke, oni neće biti enkriptovani u deljenju između samih servera, što može dovesti do curenja informacija.

Dakle, sama razmena informacija između servera je takođe jedan od bezbednosnih rizika, koja se naravno može srediti sertifikatima, gde ćemo obezbediti ceo tok kuda bi išli podatci od klienta do našeg servera i nazad. Na serverskoj strani.

# 5. Zaključak

U ovom projektu pored analize samog pop3 protokola i njegove funkcionalnosti, u laboratorijskih uslovima (improvizovanim) je odrađena sama analiza paketa kao kao i simulacija slanja kroz mrežu.

Uviđeni su propusti vezani za bezbednost kod samog slanja i preuzimanja poruka POP3 protokola, kao i ne bezbednost same ramene.

Dati su nedostatci kao i predlog zaštite protiv „cirenja“ što informacija putem mail-a tako i senzitivnih informacija kao što su sada user i password, a sutra mogu biti čak i neki kreditni detalji što niko ne zeli.

Pored konfiguracije mail servera, naučeno je kako da se detaljnije konfiguriše i sama bezbednost u vidu enkripcije i SSL sertifikata, a pokušaj improvizovanja i pravljenja sopstvenog kako bi se upotrebio u analizi produbila su znanja o istom, i zagrebana je veoma zanimljiva površina koja će svakako biti korisna u daljem razvou IT karijere.

Sam Wireshark i rekonstrukcija paketa su bili veoma jednostavni i nije bilo potrebe da se detaljnije proučava ni jedan deo paketa protokola jer je sve lepo i detaljno već bilo prikazano u samim lekcijama koje su obrađivane na časovima.

Nikako drugačije do sopstvenim zalaganjem se ne može stići do produbljivanja znanja što je ovaj projekat svakako meni doneo što u pogledu nedostatka samog protokola tako i u konfigurisanju klijenta a pogotovo servera neophodnog za nastavak karijere. Takođe se dobar deo teorije obrađene na predavanjima posle ovako praktičnog rada mnogo bolje razume i radi za zeljom za napredkom.

Kao sam zaključak za kraj bih sumirao šta je naučeno kroz projekat, i koliko je bilo korisno.

Umrežavanje vrituelnih mašina je definitivno jedna od stvari koja je mnogo razumnija, kao i povezivanje među njima. Pravljenje mail servera sada deluje mnogo jednostavnije kao i konfiguracija servera posle eksperimentisanja sa raznim opcijama besplatnog alata hMailServer. Wireshark znanja su obnovljena i ponovo je sa zadovoljstvom odrađen manji „forenzičarski“ projekat. Takođe su zahvaljujući komandnom promptu same komade pop3 protokola mnogo jasnije posle manuelne upotrebe preko istog.

# Reference:

[RFC821] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC

821, USC/Information Sciences Institute, August 1982.

[RFC822] Crocker, D., "Standard for the Format of ARPA-Internet Text

Messages", STD 11, RFC 822, University of Delaware, August 1982.