

**ELEKTROTEHNIČKI FAKULTET UNIVERZITETA U BEOGRADU**



**SIMULACIJA POVEZIVANJA HOST UREĐAJA I VEB SERVERA  
KORIŠĆENJEM VIRTUELNIH MAŠINA**

–Diplomski rad–

Kandidat:

Luka Radak 2010/357

Mentor:

doc. dr Zoran Čiča

Beograd, mart 2017.

# SADRŽAJ

SADRŽAJ .....	2
<b>1. UVOD.....</b>	<b>3</b>
<b>2. PREGLED KORIŠĆENIH SOFTVERSKIH ALATA I OPERATIVNOG SISTEMA .....</b>	<b>4</b>
2.1. <i>VIRTUALBOX</i> .....	4
2.1.1. <i>Instalacija alata VirtualBox</i> .....	4
2.2. OPERATIVNI SISTEM <i>LINUX</i> .....	6
2.2.1. <i>Uopšteno o Linux OS</i> .....	6
2.2.2. <i>Načini blokade pristupa serveru</i> .....	6
2.3. <i>APACHE SERVER</i> .....	8
<b>3. NETWORKING MODOVI ALATA <i>VIRTUALBOX</i> .....</b>	<b>9</b>
3.1. <i>NAT (NETWORK ADDRESS TRANSLATION)</i> .....	10
3.2. <i>BRIDGED NETWORKING</i> .....	10
3.3. <i>INTERNAL NETWORKING</i> .....	12
3.4. <i>HOST-ONLY NETWORKING</i> .....	12
<b>4. POSTAVKA SIMULACIJE .....</b>	<b>13</b>
4.1. POSTAVKA TESTIRANE MREŽE .....	13
4.1.1. <i>Topologija mreže i cilj simulacije</i> .....	13
4.1.2. <i>Dodavanje virtuelnih mašina</i> .....	13
4.1.3. <i>Testiranje međusobne povezanosti virtuelnih mašina</i> .....	19
4.2. POKRETANJE SERVERA.....	21
4.3. KONTROLA PRISTUPA SERVERU .....	22
4.4. OSTALA PODEŠAVANJA U <i>LINUX OS</i> .....	23
4.4.1. <i>Kreiranje veb stranice na serveru</i> .....	23
4.4.2. <i>Definisanje naziva domena servera</i> .....	23
<b>5. REZULTATI SIMULACIJE .....</b>	<b>25</b>
5.1. POSTAVKA SNIMANJA SAOBRAĆAJA.....	25
5.2. PRISTUP SERVERU SA VIRTUELNE MAŠINE <i>Host1</i> .....	26
5.3. PRISTUP SERVERU SA VIRTUELNE MAŠINE <i>Host2</i> .....	27
5.4. BLOKADA HOSTA KORIŠĆENJEM UFW.....	28
<b>6. ZAKLJUČAK .....</b>	<b>30</b>
<b>LITERATURA .....</b>	<b>31</b>

# 1. UVOD

Koncept virtuelizacije poslednjih godina sve više dobija na značaju, jer omogućava povećanje efikasnosti i fleksibilnosti u radu, uz smanjenje troškova instalacije i održavanja opreme. Razlog za ovo je to što virtuelizacija predstavlja mogućnost pokretanja više softverskih predstava operativnih sistema (OS), aplikacija i servera na jednom fizičkom uređaju, u obliku virtuelnih mašina, čime se njegovi resursi mogu iskoristiti u većoj meri. Primer prednosti ovakvog pristupa se ogleda i u ovom radu, s obzirom na to da se vrši simulacija ponašanja računarske mreže od tri računara, upotrebom samo jednog fizičkog uređaja. Veb serveri se mogu definisati kao računari koji služe za skladištenje veb sadržaja. Ovakav sadržaj je često potrebno zaštititi od neželjenog pristupa i napada, što predstavlja deo teme ovog rada.

Kao cilj teze postavlja se testiranje povezanosti veb servera sa host uređajima koji žele da mu pristupe, od kojih je jednom to dozvoljeno, a drugom ne. Simulacija ovakvog scenarija se vrši korišćenjem virtuelnih mašina, kreiranih u softverskom alatu *VirtualBox*. Ovaj rad treba da prikaže implementaciju i rezultat nekih od mogućih načina zaštite servera, u ovom slučaju *Apache* servera, kroz jedan prost i skalabilan primer.

Rad je segmentiran tako da čitalac na jednostavan način može da reprodukuje proces simulacije. Sledeća dva poglavlja upoznaju čitaoca sa operativnim sistemom i alatima koji su upotrebljeni, zatim i sa načinima umrežavanja virtuelnih mašina. U četvrtom poglavlju je detaljno opisana postavka simulacije, kako same mreže virtuelnih mašina tako i svih relevantnih podešavanja. Nakon toga sledi prikaz rezultata simulacije koji demonstrira uspešan pristup hosta veb serveru, kao i uspešno zabranjivanje pristupa serveru sa zadatih IP adresa.

## 2. PREGLED KORIŠĆENIH SOFTVERSKIH ALATA I OPERATIVNOG SISTEMA

### 2.1. *VirtualBox*

*VirtualBox* predstavlja besplatni softver kompanije *Oracle* (prvobitno razvijen od strane kompanije *Innotek*) kojim je omogućena virtuelizacija operativnih sistema, kao što su *Windows* i *Linux*. Njegovo korišćenje je omogućeno na bilo kom računaru baziranom na *AMD* ili *Intel* procesorima novijih generacija, pri čemu jedina ograničenja po pitanju mogućnosti i broja pokrenutih virtuelnih mašina predstavljaju količina memorije i slobodnog prostora na disku.

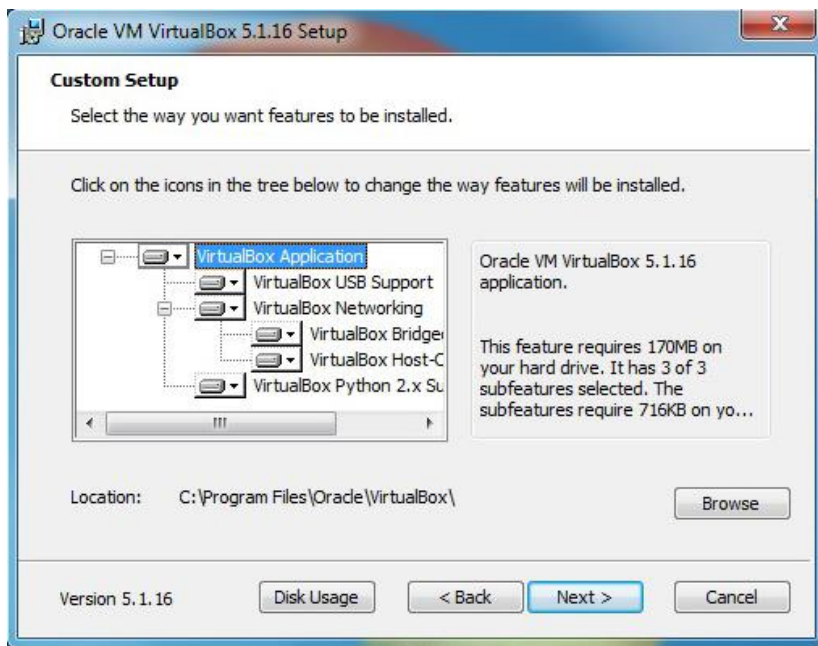
Prilikom virtuelizacije, bitno je razlikovati nekoliko osnovnih pojmova:

- *Host OS* – operativni sistem računara na kom je pokrenut *VirtualBox* (u slučaju ovog rada to je 64-bitni *Windows 7*)
- *Guest OS* – operativni sistem koji služi za željenu simulaciju, pokrenut unutar odgovarajuće virtuelne mašine (*Ubuntu 15.04* u ovom radu)
- Virtuelna mašina – služi kao specijalno okruženje kreirano od strane softvera *VirtualBox* unutar kog se pokreće željeni *guest OS*, a predstavlja vid skupa parametara koji određuju ponašanje operativnog sistema, kao što su hardverske postavke (veličina zauzete memorije, način povezivanja na mrežu itd.) i informacije o trenutnom stanju virtuelne mašine
- *Guest Additions* – razni drajveri i sistemske aplikacije koje su predviđene da budu instalirane unutar *guest OS*, u cilju boljih performansi i lakšeg korišćenja operativnog sistema od strane korisnika.

Osnovna odlika ovog softvera je povećanje mogućnosti host računara u smislu sposobnosti paralelnog pokretanja više različitih operativnih sistema na jednom uređaju.

#### 2.1.1. *Instalacija alata VirtualBox*

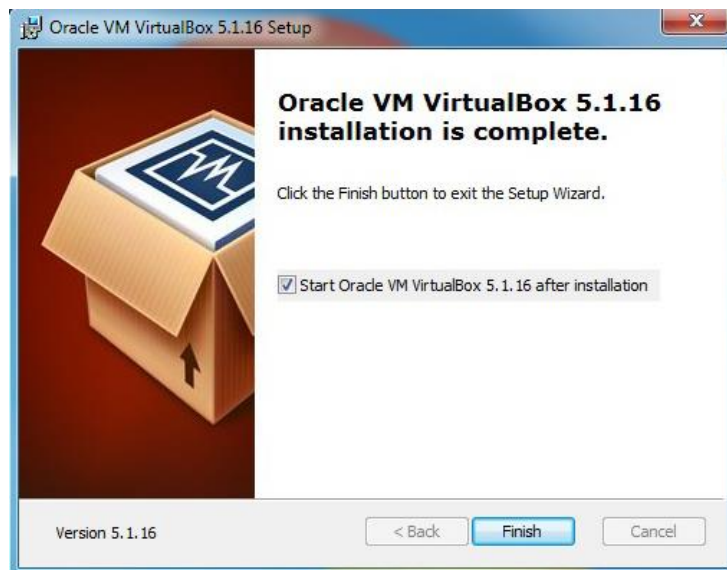
Instalacija ovog softvera je moguća preko sajta <https://www.virtualbox.org/wiki/Downloads>, preuzimanjem datoteke za odgovarajući *host OS* i njenim pokretanjem. Nakon otvaranja inicijalnog prozora i klikom na dugme *Next*, korisnik bira odlike koje softverski alat treba da sadrži, kao i lokaciju u koju će biti smeštene potrebne datoteke i direktorijumi, a u sledećem koraku se biraju dodatne opcije za instalaciju ovih funkcija. U okviru narednog prozora od korisnika se traži potvrda da želi da nastavi sa instalacijom, imajući u vidu da će njegovi mrežni interfejsi biti van funkcije neko vreme. Nakon ove potvrde, može se pokrenuti sam proces instalacije dugmetom *Install*. Po završetku ovog procesa softverski alat *VirtualBox* je spreman za upotrebu.



Slika 2.1.1. Odabir funkcija softvera i lokacije na disku



Slika 2.1.2. Upozorenje o isključenju sa mreže



Slika 2.1.3. Finalni prikaz instalacije

## 2.2. Operativni sistem *Linux*

### 2.2.1. *Uopšteno o Linux OS*

Poreklo ovog operativnog sistema potiče iz 1969. godine, kada su Denis Riči i Ken Tompson stvorili *Unix*, da bi 1980-ih godina to preraslo u GNU projekat za koji je zaslužan Ričard Stalman. Otprilike jednu deceniju kasnije zalaganjem Finca Linusa Torvaldsa dolazi do razvijanja operativnog sistema *Linux* u formi u kojoj postoji i u današnjici. On se najpreciznije može definisati kao *Linux kernel* (jezgro), koje korisnik koristi upotrebom GNU alata (*utilities*), kao što su *Bash* (*Bourne Again Shell*), *GCC*, *Coreutils* (*ls*, *cat*, *chmod*,...) itd.

Na ovom jezgru se zasnivaju *Linux* distribucije, koje su uglavnom otvorenog koda i zapravo predstavljaju operativne sisteme koji sadrže kernel, GNU alate i biblioteke, kao i dodatni softver i dokumentaciju. Neke od popularnijih distribucija su *Ubuntu*, *CentOS*, *Red Hat* i *Debian*, pri čemu je za izradu ovog rada korišćena 15.04 verzija distribucije *Ubuntu*, kako je već i napomenuto. Preveliki broj distribucija se može smatrati i manom, međutim *Linux* je veoma rasprostranjen zbog lakoće prilagođavanja različitim tipovima hardvera, otvorenosti koda, svoje sigurnosti i lakoće otkrivanja i ispravljanja grešaka u radu sa sistemom. Uz to, on je besplatan i predviđen za neprekidan rad, pa se može koristiti na raznim platformama, kao što su serveri, super računari i sl.

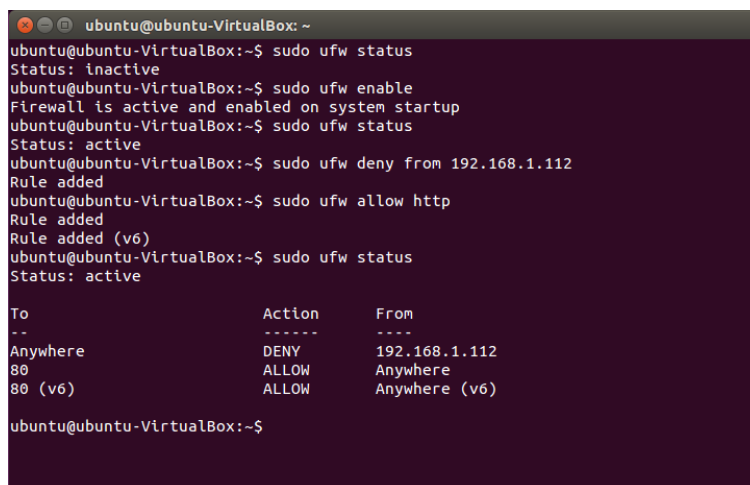
### 2.2.2. *Načini blokade pristupa serveru*

Onemogućavanje pristupa serveru određenim korisnicima predstavlja vrlo bitan pojam u računarskim mrežama, pa tako i u ovoj simulaciji. Razlozi za blokiranje koji potiče od neke IP adrese ili bloka IP adresa mogu biti različiti, kao što su odbrana od potencijalnih napada na server, cenzura za određenu regiju ili sprečavanje korišćenja usluga servera korisnicima koji su pod zabranom. Problem sa ovim pristupom odbrane se ogleda u dinamičkoj alokaciji IP adresa koja otežava blokiranje određenih klijenata bez rizikovanja kolateralne štete i onemogućavanja pristupa drugim računarima koji dele isti adresni prostor koji je blokiran usled pokušaja pokrivanja što većeg potencijalno opasnog opsega. Još jedan način na koji se može zaobići ovakva zabrana je upotreba *proxy* servera, ukoliko server kom se pristupa nema dobru odbranu protiv njih. Postoji više načina da se onemogućeni nedozvoljen pristup, od kojih će neki od uobičajenih biti opisani u nastavku teksta.

### i) Upotreba fajervola

Distribucija *Ubuntu* po pravilu ima automatski instaliran jedan veoma koristan alat za zabranu pristupa serveru korišćenjem fajervola, a to je UFW (*Uncomplicated Firewall*), koji predstavlja interfejs prema alatki *iptables* i omogućava pojednostavljenje procesa konfiguracije fajervola. Ovo je posebno pogodno za početnike, kojima bi *iptables* mogao biti komplikovan za ovakvu realizaciju. Ukoliko je nekim slučajem UFW obrisan sa operativnog sistema, njegova instalacija se može pokrenuti komandom *sudo apt-get ufw*.

Da bi fajervol bio pokrenut, kao i da bi se mogla izvršiti njegova konfiguracija, potrebno je izvršiti komandu *sudo ufw enable*. Ovime se, bez ikakvih izvršenih podešavanja i dodatih pravila, blokira sav saobraćaj koji pristiže na veb server mašine koja pokreće fajervol. Iz tog razloga neophodno je dodati pravila, koja će odrediti kome je dozvoljen pristup serveru. U slučaju ove simulacije, postoji želja za propuštanjem *http* saobraćaja koji potiče od bilo koje IP adrese, sem od one koja odgovara virtuelnoj mašini *Host2*. Ukoliko bi se UFW koristio za ovaj primer, najpre bi bilo dodato pravilo koje brani saobraćaj koji potiče od određene IP adrese, navedene u daljem delu rada, a zatim i omogućen ostali *http* saobraćaj. Treba naglasiti da je redosled ove dve komande vrlo bitan, jer bi pri obrnutom redosledu bio omogućen sav *http* saobraćaj. Spisak pravila se može proveriti sa *sudo ufw status*. Ceo ovaj postupak prikazan je na slici 2.2.2.1., a umesto naziva protokola *http*, može se iskoristiti i odgovarajući broj porta - 80.



```
ubuntu@ubuntu-VirtualBox: ~
ubuntu@ubuntu-VirtualBox:~$ sudo ufw status
Status: inactive
ubuntu@ubuntu-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
ubuntu@ubuntu-VirtualBox:~$ sudo ufw status
Status: active
ubuntu@ubuntu-VirtualBox:~$ sudo ufw deny from 192.168.1.112
Rule added
ubuntu@ubuntu-VirtualBox:~$ sudo ufw allow http
Rule added
Rule added (v6)
ubuntu@ubuntu-VirtualBox:~$ sudo ufw status
Status: active

To Action From
--
Anywhere DENY 192.168.1.112
80 ALLOW Anywhere
80 (v6) ALLOW Anywhere (v6)

ubuntu@ubuntu-VirtualBox:~$
```

Slika 2.2.2.1. Konfigurisanje UFW

Pored ovakve primene UFW, moguće je blokirati ili dozvoliti pristup čitavim opsezima IP adresa, dodavanjem dužine mrežnog dela adrese na njen kraj u komandi. Takođe se može i definisati port ili interfejs uređaja sa fajervolom za koji navedeno pravilo treba da važi, kao i druge slične opcije.

Brisanje određenih pravila fajervola se može izvršiti na dva načina. Prvi način je brisanje korišćenjem rednog broja pravila iz tabele, dobijenog na osnovu rezultata komande *sudo ufw status numbered*, pri pokrenutom UFW. Ovo se realizuje naredbom *sudo ufw delete 23*, za primer brisanja pravila pod rednim brojem 23. Drugi pristup podrazumeva brisanje na osnovu naziva pravila, pa bi tako brisanje prethodno navedenog pravila koji dozvoljava *http* saobraćaj bilo izvršeno komandom *sudo ufw delete allow http*. Gašenje fajervola se izvršava sa *sudo ufw disable*.

Prilikom pokušaja nedozvoljenog pristupa veb stranici zaštićenog servera, korisnik kojem je pristup zabranjen neće moći da pokrene stranicu, a u ovakvoj postavci neće čak dobiti ni povratnu informaciju o tome da mu je zahtev odbijen.

#### ii) *Izmena konfiguracije Apache servera*

Drugi način blokiranja pristupa veb serveru podrazumeva izmenu teksta konfiguracione datoteke *Apache* servera, čija su svojstva navedena u narednom odeljku. Ovaj pristup će biti korišćen i pri simulaciji primera iz ovog rada, iako upotreba UFW daje željene rezultate, najviše zbog vizuelnog prikaza, s obzirom na to da izmena konfiguracije servera omogućava povratnu informaciju korisniku koji nedozvoljeno pokuša da pristupi serveru, ispisujući poruku o zabrani.

Konfiguracija *Apache* servera se može izmeniti nekim od tekstualnih editora, otvaranjem datoteke na putanji `/etc/apache2/apache2.conf`. Nakon toga, potrebno je pronaći deo teksta koji se odnosi na direktorijum `/var/www/`, odnosno između tagova „<Directory /var/www/>” i „</Directory>”. Unutar ovog dela datoteke neophodno je dodati pravila o dozvoljenim i zabranjenim IP adresama, a ključne reči za ovo su *Order*, *Allow* i *Deny*. Prvom od ove tri se određuje redosled izvršavanja druge dve direktive. Tako bi pri liniji “Order allow,deny” najpre bila izvršena provera kasnije navedenih uslova direktive o dozvoljenom IP saobraćaju, a zatim ona koja vrši zabranu. Treba imati na umu, da se korisniku odmah odbija pristup serveru ako ne ispunjava makar jedan uslov pod *Allow* direktivom, a u suprotnom prelazi na uslove *Deny* direktiva, kojih takođe može biti više. Ukoliko bi ovaj redosled bio obrnut, prvo bi bio ispitivan uslov naveden pod direktivom *Deny*, a u slučaju da korisnik ispunjava ovaj uslov pristup serveru mu je onemogućen, osim ukoliko ispunjava i uslov direktive *Allow* koji se sledeći ispituje. Kako bi ove direktive imale efekta, izmenjena je i linija sa direktivom *AllowOverride*, menjajući joj argument “None” u “All”.

Konkretan primer ove izmene konfiguracije i primena u slučaju ove teze, detaljno je opisan u poglavlju posvećenom postavci simulacije.

#### iii) *Fail2Ban*

Ovaj alat reaguje na učestale neuspešne pokušaje autentifikacije nekog korisnika na određeni servis, ili druge aktivnosti potencijalno štetne po server. Do saznanja o neuspešnim autentifikacijama *Fail2Ban* „zatvori“ dolaze traženjem šablona koji bi na to ukazali, unutar logova („dnevnika“) ispisanih od strane tog servisa. *Fail2Ban* može da izda privremenu zabranu IP adresa od kojih potiču takve pretnje, dinamičkom modifikacijom pravila fajervola. Detalji same instalacije ovog alata, kao i relevantna podešavanja za rad sa *Apache* serverom, neće biti opisivani u sklopu ovog rada, ali mogu se pronaći na stranici <https://www.digitalocean.com/community/tutorials/how-to-protect-an-apache-server-with-fail2ban-on-ubuntu-14-04>.

## 2.3. *Apache* server

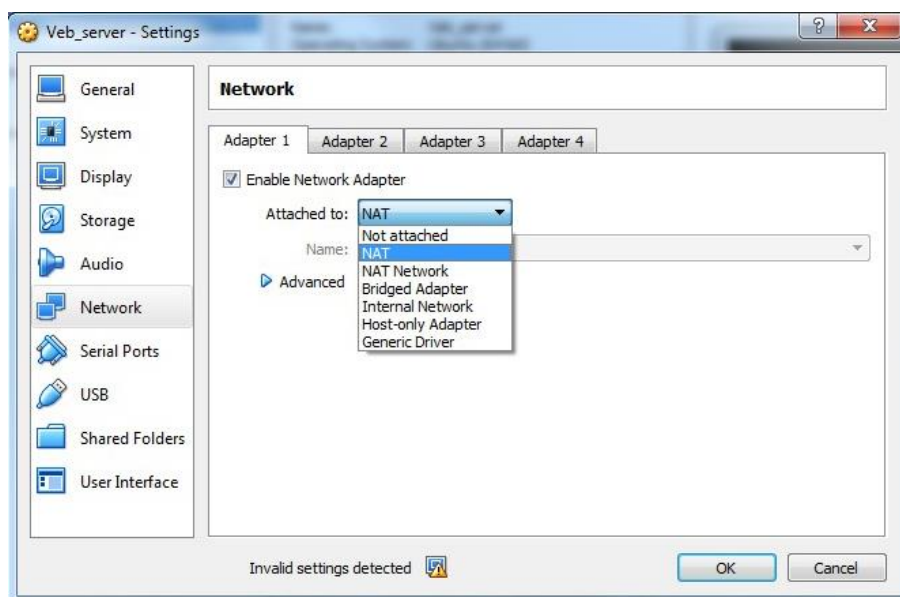
*Apache HTTP Server Project* predstavlja projekat upravljani od strane grupe volontera lociranih širom sveta sa ciljem da se implementira HTTP (veb) server koji pruža puno mogućnosti, otporan je i besplatan, a deo je fondacije *Apache Software Foundation*. Članovi ovog projekta teže ka tome da *Apache* ostane platforma pomoću koje institucije i pojedinci mogu da rade na izgradnji pouzdanih sistema. Shodno tome, *Apache* server je najzastupljeniji softver za veb server na svetu.

Prilikom simulacije teze, *Apache* server će biti pokrenut na virtuelnoj mašini, a on će „ugostiti“ jednu veb stranicu koja će se tu nalaziti. Sa uređaja (ili u ovom slučaju virtuelne mašine) na kom je pokrenut server, direktorijumu servera na hard disku u kom se nalazi pomenuta veb stranica može se pristupiti ukucavanjem `localhost/` u veb pretraživač, a *default* pretraživač u *Linux OS* je *Mozilla Firefox*. Na ovaj URL je moguće nadovezati naziv stranice koju treba pokrenuti.



### 3. NETWORKING MODOVI ALATA VIRTUALBOX

Softverski alat *VirtualBox* obezbeđuje i do 8 virtuelnih *PCI Ethernet* kartica svakoj virtuelnoj mašini koja je dodata, pri čemu je za svaku od tih kartica moguće odabrati koji će hardver biti virtuelizovan i mod virtuelizacije u zavisnosti od željene povezanosti sa fizičkom mrežom. Svih 8 mrežnih kartica je moguće konfigurisati preko komandne linije sa *VboxManage modifyvm*. Korišćenjem grafičkog interfejsa alata *VirtualBox*, podešavanjima za 4 adaptera je moguće pristupiti odabirom željene virtuelne mašine i opcije *Settings*, otvarajući karticu *Network* unutar novootvorenog prozora. Nakon ovoga, neophodno je štiklirati opciju *Enable Network Adapter* čime se aktivira željeni adapter, zatim odabrati mod virtuelizacije iz padajućeg menija *Attached to* i naziv adaptera iz padajućeg menija *Name*.



Slika 3.1. *Network* podešavanja

Postoji sedam ponuđenih modova:

- 1) *Not attached* – gostujuća mašina ima postojeću mrežnu karticu, ali ne i konekciju, kao da *Ethernet* nije priključen na karticu
- 2) *NAT (Network Address Translation)* – pogodan ukoliko *guest OS* treba da služi samo za pretraživanje Interneta, preuzimanje datoteka ili za *e-mail*
- 3) *NAT Network* – mreža koja koristi NAT servis poput kućnog rutera
- 4) *Bridged networking* – za naprednije potrebe umrežavanja poput pokretanja servera unutar *guest OS* i simulacije mreže
- 5) *Internal networking* – međusobno umrežavanje virtuelnih mašina, koje nije vidljivo od strane *host OS* ili njegovih programa

- 6) *Host-only networking* – povezivanje host uređaja sa virtuelnim mašinama bez upotrebe njegovog fizičkog mrežnog interfejsa, već kreiranjem virtuelnog interfejsa nalik *loopback* interfejsu, na *host OS*
- 7) *Generic networking* – korisniku omogućava odabir drajvera koji može biti distribuiran unutar *extension pack* ili uključen u *VirtualBox*. U ovoj situaciji retko upotrebljavani modovi *UDP Tunnel* i *VDE (Virtual Distributed Ethernet) networking* koriste isti opšti interfejs.

U nastavku će biti detaljnije opisani neki od najčešće korišćenih modova.

### 3.1. NAT (*Network Address Translation*)

NAT predstavlja *default* način rada mrežne kartice u softverskom alatu *VirtualBox*, s obzirom na to da najčešće nije potrebno ni izvršiti konfiguraciju na host i *guest* sistemima. Ovaj mod je samim tim najjednostavniji način na koji virtuelna mašina može pristupiti nekoj eksternoj mreži, nalik pravom kompjuteru koji se putem rutera povezuje na internet. Funkciju rutera u ovom slučaju vrši *VirtualBox networking engine* koji na transparentan način mapira dolazni i odlazni saobraćaj za virtuelnu mašinu i postavlja se između svake od pokrenutih mašina i host uređaja.

DHCP server integrisan u *VirtualBox* dodeljuje mrežnu adresu i određuje konfiguraciju virtuelne mašine na privatnoj mreži, pa zato ova adresa uglavnom pripada potpuno različitoj mreži od one kojoj pripada host. Prva mrežna kartica koja koristi NAT je povezana na mrežu 10.0.2.0/24, druga na 10.0.3.0/24 itd., pri čemu je moguće promeniti ova automatska podešavanja ukoliko postoji potreba za tako nešto.

Nedostatak NAT moda se ogleda u tome što je virtuelna mašina nedostupna od strane drugih mreža, poput privatne mreže povezane na ruter, pa stoga nije moguće pokrenuti server na ovaj način ukoliko ne bi bio postavljen *port forwarding*. Takođe, postoje četiri ograničenja ovog moda:

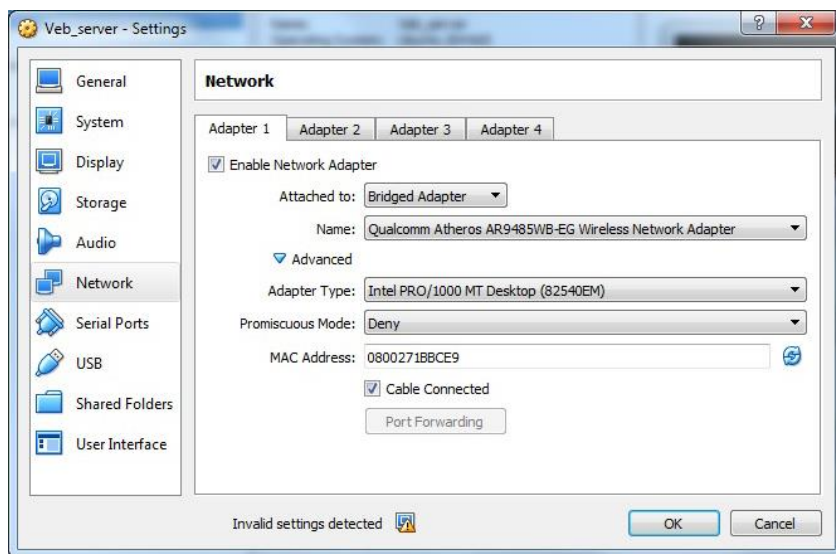
- 1) ICMP protokol – postoje određeni problemi sa mrežnim *debugging* koje koriste ovaj protokol pri slanju i primanju poruka, uprkos tome što je poboljšana podrška za ICMP od verzije 2.1, kada je problem sa alatkom *ping* otklonjen
- 2) Prijem UDP *broadcast* poruka je nepouzdan
- 3) Nedostatak podrške za protokole koji nisu TCP ili UDP
- 4) Nemogućnost korišćenja portova čiji je broj manji od 1024 za *port forwarding*.

### 3.2. *Bridged networking*

U ovom odeljku će biti više reči o modu umrežavanja koji je korišćen i pri simulaciji ove teze. Kada je uključen *Bridged adapter*, *VirtualBox* se povezuje sa jednom od instaliranih mrežnih kartica i vrši direktnu razmenu mrežnih paketa, zaobilazeći mrežni stek host operativnog sistema. Mrežni stek predstavlja slojevit skup protokola koji zajedničkim radom obezbeđuje neku mrežnu funkciju. *VirtualBox* koristi drajver sa host sistema („net filter“ drajver) kojim se filtriraju podaci sa fizičkog mrežnog adaptera i omogućava ovom programu presretanje podataka sa fizičke mreže i ubacivanje podataka u nju. Ovime se praktično kreira novi mrežni interfejs unutar softvera, pa deluje kao da je *guest OS* povezan na interfejs preko kabla. To znači da host može da razmenjuje podatke sa *guest* sistemom koristeći ovaj interfejs, čime je omogućeno rutiranje između gostujućeg sistema i ostatka mreže.

Postavka *bridged* moda je, kao i kod NAT, vrlo jednostavna, jer je neophodno samo odabrati željeni host interfejs koji će služiti za umrežavanje, u sklopu *Network* dela dijaloga *Settings* za

određenu virtuelnu mašinu (slika 3.2.1.). Pritom, povezivanje sa bežičnim interfejsom je nešto drugačije od običnog povezivanja, jer mnogi bežični adapteri nemaju podršku za *promiscuous* mod rada. S obzirom na to da je za sav saobraćaj neophodna upotreba MAC adrese bežičnog adaptera host uređaja, *VirtualBox* vrši zamenu izvorišne MAC adrese u sklopu *Ethernet* zaglavlja odlazećeg paketa, kako bi odgovor sigurno mogao da bude poslat na host interfejs. Ukoliko dolazeći paket za destinacionu adresu ima IP adresu adaptera jedne od virtuelnih mašina, *VirtualBox* vrši ubacivanje MAC adrese adaptera odgovarajuće virtuelne mašine na mesto odredišne MAC adrese *Ethernet* zaglavlja paketa i prosleđuje ga. Ispitivanjem ARP (*Address Resolution Protocol*) i DHCP paketa, *VirtualBox* dolazi do informacija o IP adresama interfejsa pokrenutih virtuelnih mašina.



Slika 3.2.1. Podešavanja *bridged* adaptera

U zavisnosti od *host OS*, postoje određena ograničenja koja se moraju uzeti u obzir pri konfiguraciji mreže. Tako je u slučaju *Macintosh* operativnog sistema i upotrebe *AirPort* bežičnog interfejsa za ovaj način rada moguć samo rad sa IPv4 i IPv6 protokolom. Ovo važi i za bežične adaptore operativnog sistema *Linux*, kao i moguć gubitak paketa usled podešavanja vrednosti MTU (*Maximum Transmission Unit*) na manje od 1500 bajtova na određenim interfejsima koji se povezuju kablom. Neki adapteri ovog operativnog sistema brišu VLAN (*Virtual Local Area Network*) tagove na hardveru, što onemogućava *VLAN trunking* između virtuelne mašine i eksterne mreže sa nekim starijim verzijama *Linux* jezgra. Ukoliko se virtuelna mašina pokreće na *Solaris OS*, ne postoji podrška za bežične interfejse, kao ni potpuna podrška filtriranja saobraćaja gostujućeg operativnog sistema.

S obzirom na to da je za realizaciju simulacije ove teze potrebno pokrenuti server na jednoj od virtuelnih mašina, kao i međusobno povezati kreirane mašine, pri čemu ne postoji potreba za sigurnošću i „odsecanjem“ ove mreže od ostatka sveta, odabran je *Bridged networking* mod za umrežavanje *guest* sistema jer ispunjava ove zahteve. Pri tome, simulacija je pokrenuta na *Windows OS* kao host sistemu, pa nisu očekivani neki od problema navedenih u prethodnom pasusu.

### 3.3. *Internal networking*

Potencijalni nedostatak *Bridged networking* moda u nekim slučajevima može biti nedostatak sigurnosti unutar mreže. Ovaj problem se može rešiti odabirom *Internal* moda umrežavanja, koji je sličan prethodno opisanom, s tim što uređaji koji nisu na istom host računaru ne mogu pristupiti ovoj mreži, već se komunikacija vrši samo interno (i direktno) među virtuelnim mašinama pokrenutih na istom host uređaju. Takođe, sam host ne može da prati komunikaciju između dve virtuelne mašine koja je privatna. Razlog za ovo je to što se, za razliku od *Bridged networking* slučaja, saobraćaj ne odvija preko fizičkog interfejsa host sistema.

### 3.4. *Host-only networking*

*Host-only* način umrežavanja se može smatrati kao mešavina *Bridged* i *Internal networking* modova. Komunikacija između host sistema i virtuelnih mašina se odvija preko fizičkog *Ethernet* sviča, nalik *Bridged* modu, dok ne postoji mogućnost komuniciranja sa „spoljnim svetom“, kao i kod internog umrežavanja, zbog nedostatka povezanosti sa fizičkim mrežnim interfejsom host uređaja. Zato *VirtualBox* prilikom korišćenja ovog moda kreira novi softverski interfejs na host računaru, nalik *loopback* interfejsu, čiji saobraćaj može biti presretnut i analiziran, što nije moguće pri korišćenju *Internal networking* moda, dok se međusobna komunikacija virtuelnih mašina ne može videti. Primer upotrebe *Host-only* načina umrežavanja je umrežavanje virtuelne mašine na kojoj se nalazi veb server i virtuelne mašine na kojoj se nalazi baza podataka. Veb server za ostvarivanje svoje funkcionalnosti koristi podatke iz baze podataka (na primer, za dinamički ispis sadržaja), ali spoljašnje okruženje nema potrebe da pristupa bazi podataka, pa je za povezivanje dve navedene mašine *Host-only* načina umrežavanja adekvatan. Pošto, veb serveru treba omogućiti da se pristupa i iz spoljašnjeg sveta, onda će virtuelna mašina na kojoj se nalazi veb server pomoću *Bridged* moda da se poveže na spoljašnje okruženje, ali bez obzira na to virtuelna mašina sa bazom podataka i dalje ostaje nevidljiva spoljašnjem svetu.

## 4. POSTAVKA SIMULACIJE

U ovom poglavlju biće detaljno opisan postupak postavke primera, odnosno podešavanja testirane mreže i *Linux* operativnog sistema odgovarajućih virtuelnih mašina. U sklopu opisa *Linux* podešavanja biće reči o načinu pokretanja servera, blokiranja pristupa, kreiranju veb stranice, kao i instalaciji potrebnih alata. Treba takođe imati na umu da termin *host* sada ima drugačije značenje od prethodno korišćenog i odnosi se na uređaj koji je povezan na server. Neke komande koje se koriste u *Linux OS* zahtevaju administratorske dozvole koje poseduje korisnik *root*. Stoga takve komande započinju sa *sudo*, što omogućava da se komanda izvrši kao da ju je pokrenuo *root* korisnik. Prilikom prve ovakve komande, terminal traži ukucavanje lozinke *root* korisnika, koja u ovom slučaju glasi „reverse“, iz razloga koji je naveden u daljem tekstu poglavlja. OS pamti uspešno ukucanu lozinku neko određeno vreme definisano u sistemu, pa u tom periodu korisniku neće biti ponovo zatraženo njeno ukucavanje. Zato je ovaj korak verifikacije izostavljen u daljem delu teksta.

### 4.1. Postavka testirane mreže

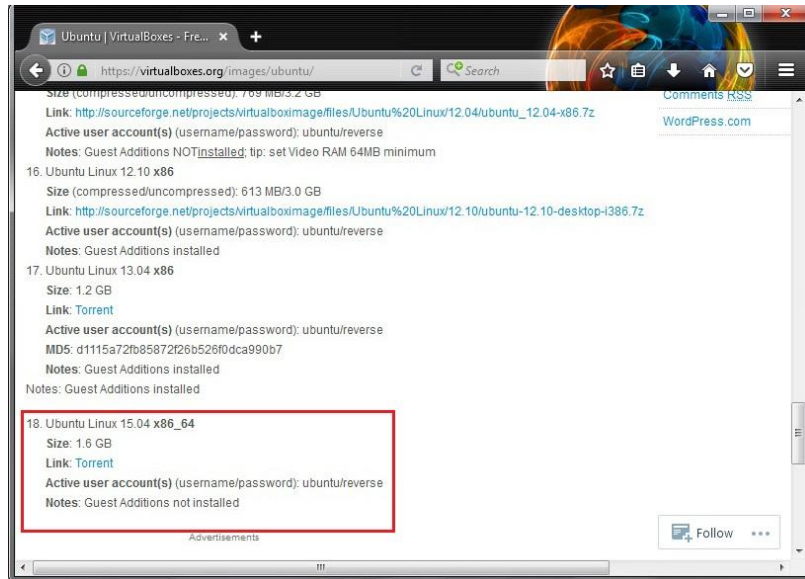
#### 4.1.1. Topologija mreže i cilj simulacije

S obzirom na to da se teza bavi korišćenjem virtuelnih mašina za simulaciju mreže i blokiranjem pristupa serveru, za testiranje je dovoljna mreža koju čine tri virtuelne mašine, koje predstavljaju server i dva host uređaja. Ove mašine su međusobno povezane preko *bridged* adaptera, čije su osobine navedene u prethodnom poglavlju, pri čemu svaka ima po jedan takav interfejs. Prilikom ove postavke, od interesa je povezanost mašine na kojoj će biti pokrenut veb server sa host mašinama, dok međusobna veza dva host uređaja nije od velikog značaja za realizaciju ovog primera, iako će ona svejedno biti uspostavljena.

Cilj simulacije ove mreže virtuelnih mašina je testiranje kontrole pristupa veb serveru, takve da je pristup dozvoljen mašini *Host1*, ali ne i mašini *Host2*. Način na koji će ta blokada biti ostvarena je opisana u daljem delu poglavlja. Ovakvom simulacijom može se prikazati jedna realna situacija kada je iz nekog razloga potrebno blokirati određene korisnike na internetu, ukoliko postoji mogućnost napada na server ili nekog drugog vida neautorizovanog pristupa.

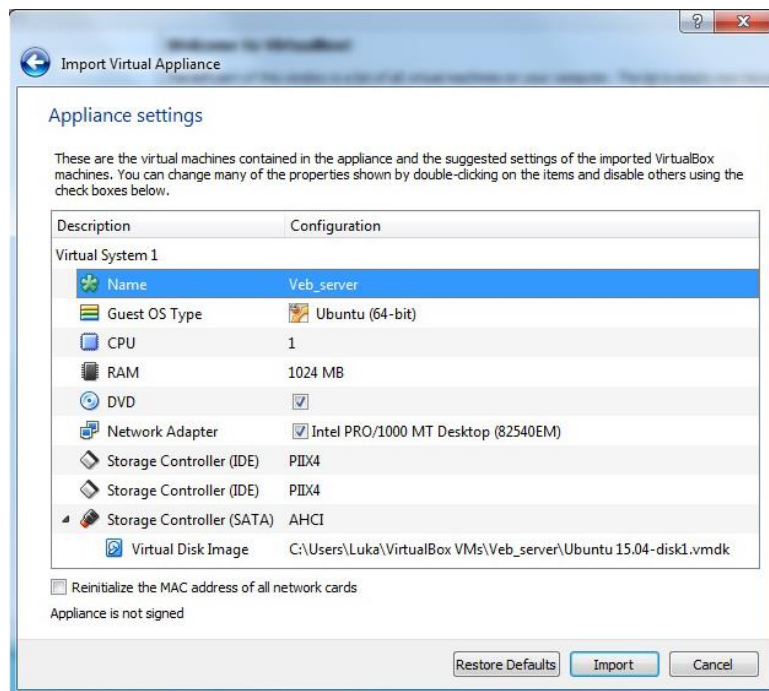
#### 4.1.2. Dodavanje virtuelnih mašina

Postoji više načina na koje se mogu dodati virtuelne mašine u softverskom alatu *VirtualBox* – dodavanjem već pripremljenih virtuelnih mašina korišćenjem *.ova* ili *.vdi* datoteka, ili kreiranjem novih virtuelnih mašina. Prvi pristup se svodi na ubacivanje već kreiranog virtuelnog hard diska sa određenim podešavanjima i podignutim operativnim sitemom, dok se u drugom slučaju kreira novi virtuelni hard disk na kom nije ništa instalirano. Gotove virtuelne mašine, odnosno njihove „slike“ (*VM image*), mogu se preuzeti sa stranice <https://virtualboxes.org/images/>, odabirom daljeg linka koji vodi ka preuzimanju datoteke za željeni operativni sistem i distribuciju. Za izradu ove simulacije odabrana je *Ubuntu 15.04* distribucija operativnog sistema *Linux*, pa se sa stranice <https://virtualboxes.org/images/ubuntu/#ubuntu1504> pokreće odgovarajuća *.torrent* datoteka koja služi za preuzimanje „slike“ virtuelne mašine, koja je u ovom slučaju u *.ova* formatu.



**Slika 4.1.2.1. Preuzimanje “slike” za Ubuntu 15.04**

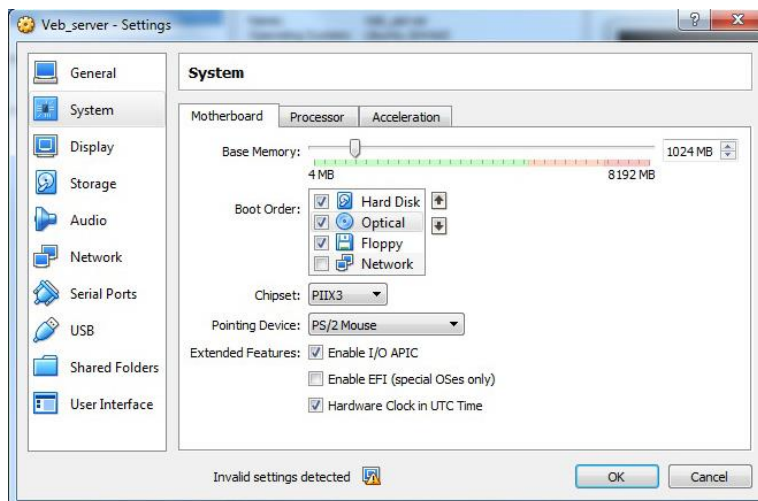
Prva virtuelna mašina koja se ubacuje u *VirtualBox* je ona na kojoj će biti pokrenut veb server. Navedeni postupak je vrlo jednostavan kada je u pitanju rad sa datotekom u .ova formatu. Klikom miša na dugme *Import* u glavnom prozoru alata *VirtualBox* otvara se novi prozor koji nudi opciju biranja datoteke koja će poslužiti kao „slika“ virtuelne mašine. Nakon izvršenog odabira, prelazi se na definisanje nekih parametara mašine koja se dodaje, kao što je prikazano na slici 4.1.2.2.



**Slika 4.1.2.2. Parametri pri dodavanju virtuelne mašine**

Posle promene naziva ove virtuelne mašine u „Veb\_server“, ostavljajući ostale parametre na već postavljene vrednosti, biranjem opcije *Import* se pokreće proces dodavanja željene mašine u *VirtualBox*. Sada je moguće promeniti neka podešavanja virtuelne mašine *Veb\_server* biranjem

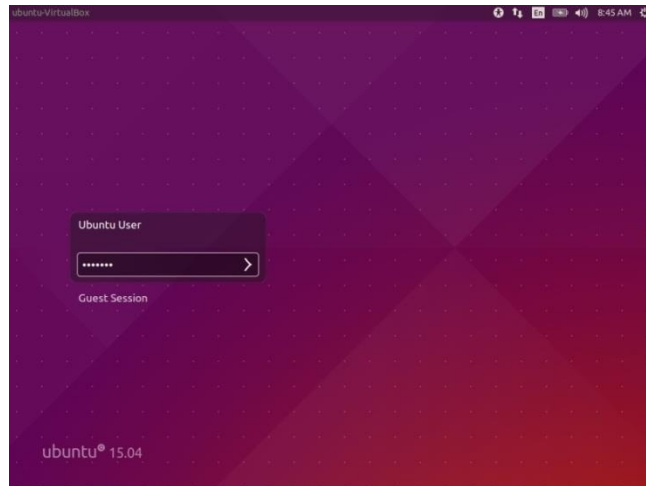
njenog prikaza u listi dodatih mašina i klikom miša na dugme *Settings*. Isto je moguće postići i desnim klikom na ime mašine, birajući opciju *Settings*. Po otvaranju odgovarajućeg prozora, odabirom kartice *System* dolazi se do biranja prioriteta uređaja sa kojih *VirtualBox* pokušava podizanje operativnog sistema pri pokretanju virtuelne mašine, u sklopu dela prozora pod nazivom *Boot order* (slika 4.1.2.3.). Takođe je moguće podesiti i druge parametre, kao što je količina RAM memorije koju mašina zauzima.



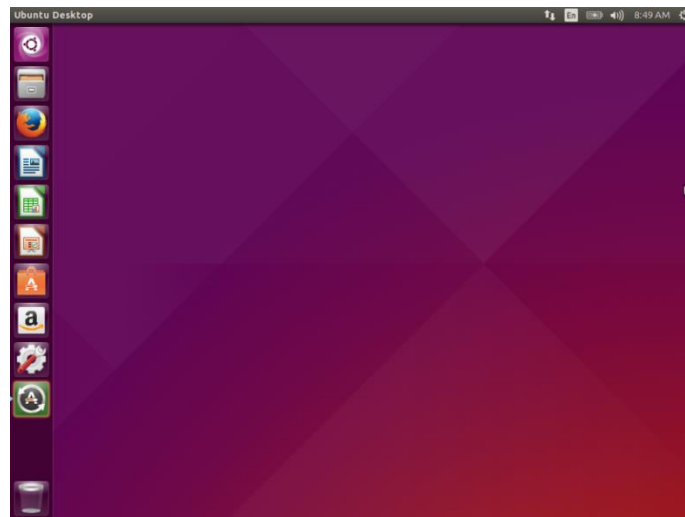
**Slika 4.1.2.3. System podešavanja virtuelne mašine**

Za potrebe ove simulacije, od većeg značaja su podešavanja u sklopu kartice *Network*, pomenute u delu teksta posvećenog *networking* modovima. Upravo u tom odeljku je i navedeno da će biti korišćen *bridged* adapter, pa se nakon osposobljavanja mrežnog adaptera bira opcija *Bridged adapter*, kao na slici 3.1., a zatim i njegov naziv iz liste ponuđenih, gde se nakon klika na opciju *Advanced* mogu videti parametri ovog interfejsa, koje nije potrebno menjati. Ove postavke za virtuelnu mašinu *Veb\_server* mogu se videti na slici 3.2.1. Klikom mišem na dugme *OK* se potvrđuju sve odrađene izmene i vrši se povratak u glavni prozor.

Kako bi alat *Wireshark*, koji će pri pokretanju simulacije služiti za analizu saobraćaja u mreži, bio prisutan na sve tri virtuelne mašine, a da bi bilo izbegnuto pokretanje instalacije za svaku posebno jer će host mašine biti kreirane kloniranjem prve, pre nastavka postavljanja mreže bilo bi efikasno pokrenuti *Veb\_server* i instalirati željeni alat. Pokretanje virtuelne mašine se može izvršiti duplim klikom miša na njen naziv u sklopu liste mašina. Nakon nekog vremena pojaviće se prikaz za logovanje na podignutu *Ubuntu* distribuciju (slika 4.1.2.4.), korišćenjem korisničkog imena i lozinke koji su prikazani na veb stranici sa koje je preuzeta datoteka „slike“ virtuelne mašine. U ovom slučaju to su korisničko ime „ubuntu“ i lozinka „reverse“. Nakon unosa ovih podataka i pritiskom na taster *Enter*, dolazi se do grafičkog prikaza podignutog operativnog sistema.



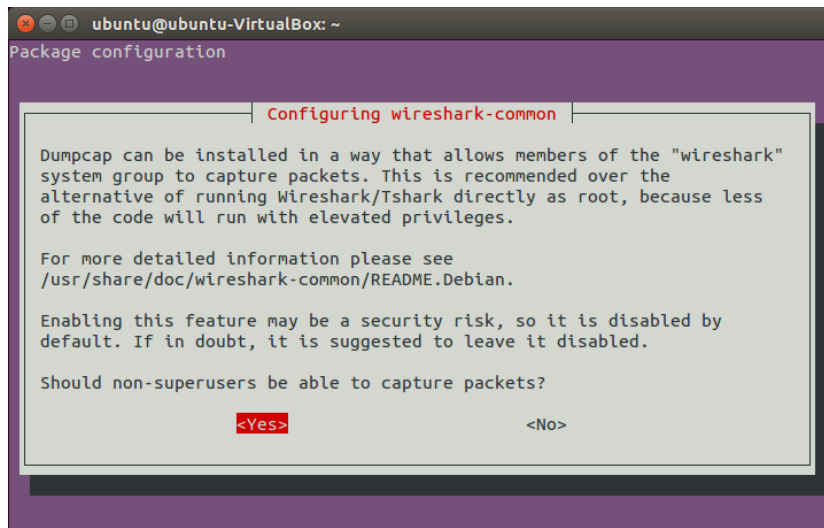
Slika 4.1.2.4. Unos šifre korisnika



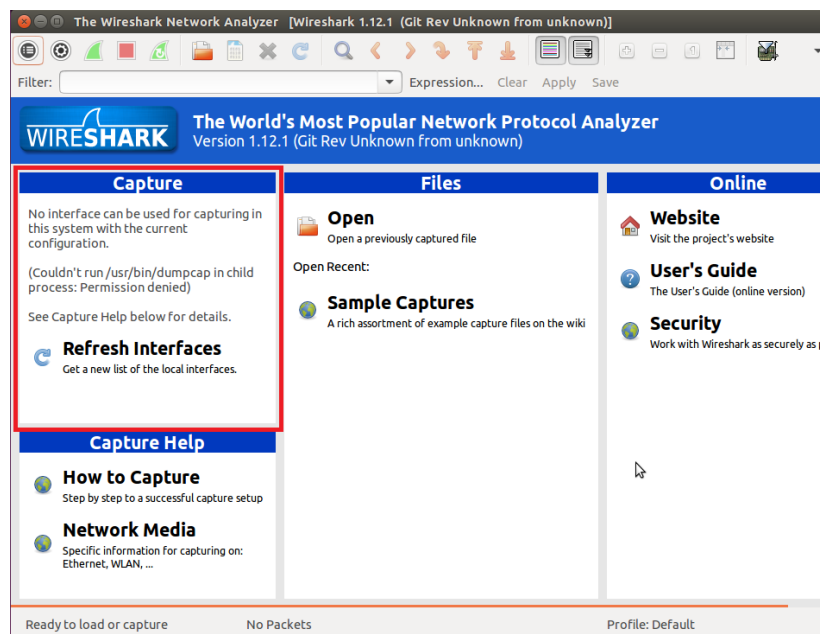
Slika 4.1.2.5. Desktop distribucije *Ubuntu*

Instalacija alata *Wireshark* se pokreće kucanjem komande `sudo apt-get install -y wireshark` u terminalu, koji je moguće otvoriti kombinacijom tastera Ctrl+Alt+T iz desktopa. Opcija `-y` ove komande vrši automatski pristanak na količinu memorije koja će biti zauzeta. Kada proces instalacije dođe do određene tačke, korisniku se postavlja pitanje o tome da li želi da snimanje paketa bude omogućeno i korisnicima sistemske grupe „*Wireshark*“, a ne samo *root* korisniku kao u *default* postavci. Bira se odgovor „*Yes*“ (slika 4.1.2.6.), jer za potrebe ove simulacije povećana bezbednost nije od velike važnosti, a tasterom *Enter* se nastavlja instalacija. Ovo nije dovoljno da bi korisnik bez administrativnih privilegija mogao da snima saobraćaj (slika 4.1.2.7.), već je neophodno dodati ga u pomenutu grupu „*Wireshark*“, komandom `sudo usermod -a -G wireshark ubuntu`, za korisnika „*ubuntu*“ u ovom slučaju. Umesto konkretnog naziva korisnika može se upotrebiti izraz `$(whoami)`. Potrebno je još i izlogovati ovog korisnika sa `gnome-session-quit --logout --no-prompt`, pa će mu sledećim logovanjem biti omogućeno korišćenje alata *Wireshark* u potpunosti. Nakon ovog procesa, gasi se virtuelna mašina prostim zatvaranjem prozora u kojoj je otvorena i biranjem opcije *Power off the machine*, kako bi se prešlo na naredne korake postavke mreže.



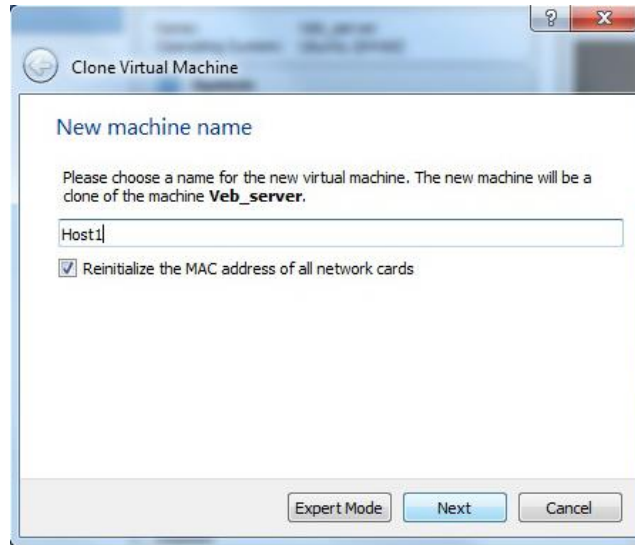


Slika 4.1.2.6. Biranje privilegija za snimanje paketa



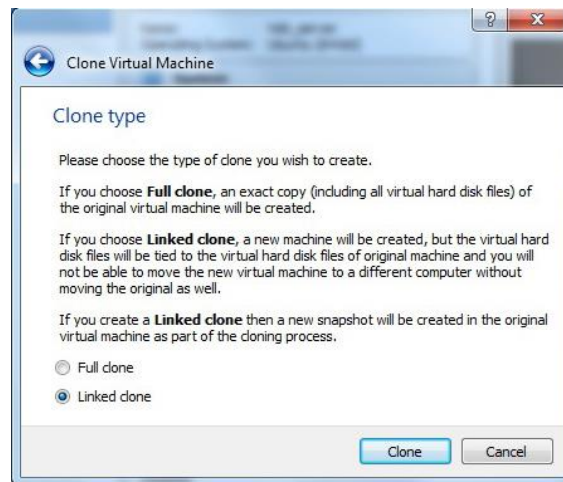
Slika 4.1.2.7. Nemogućnost biranja interfejsa nakon instalacije

Nakon što je dodata prva virtuelna mašina, koja predstavlja veb server, može se pristupiti dodavanju druge dve, koje odgovaraju host uređajima. Ovo je najlakše realizovati kloniranjem prve mašine, umesto ponavljanja procesa koji je prethodno opisan, čime je i obezbeđeno da one odmah imaju instaliran alat *Wireshark*. Postupak kloniranja prve mašine počinje desnim klikom miša na nju i biranjem opcije *Clone*. U prozoru koji je tom prilikom otvoren upisuje se naziv nove virtuelne mašine („Host1“). Pored ovoga, bitno je izabrati opciju reinicijalizacije MAC adrese za sve mrežne kartice, jer bi u suprotnom nova mašina imala istu MAC adresu interfejsa kao i originalna, pa njihova međusobna komunikacija ne bi bila moguća.

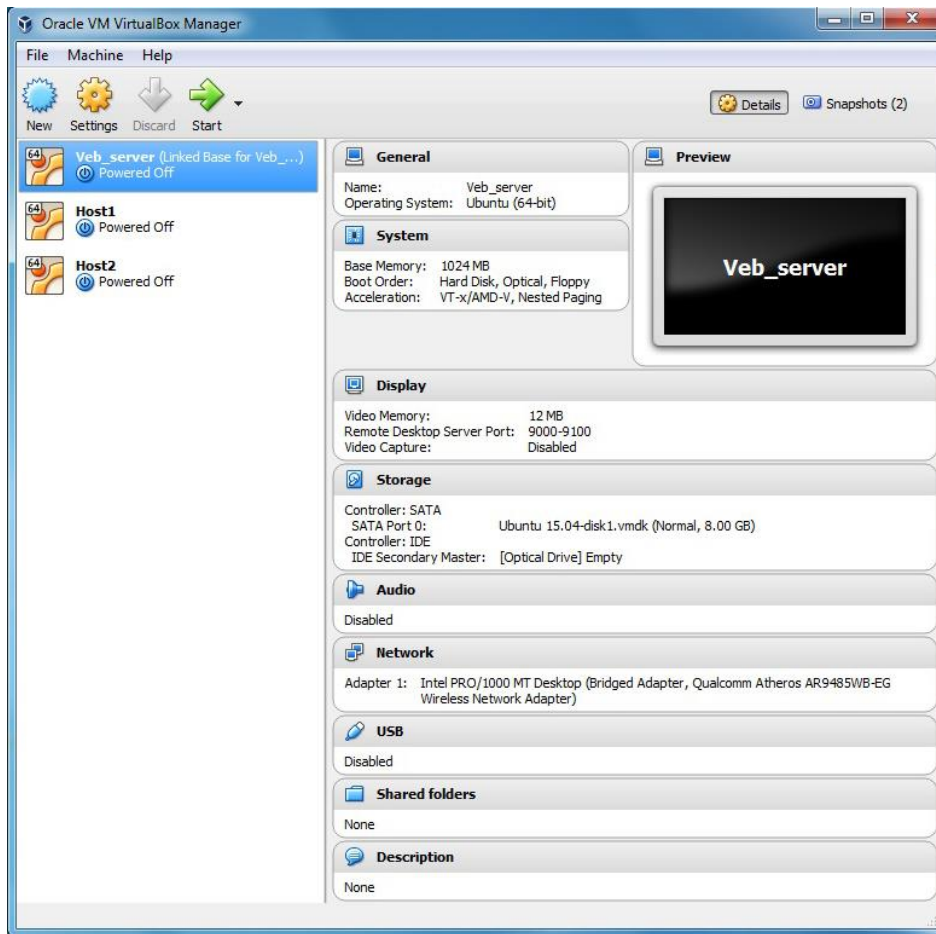


Slika 4.1.2.8. Prvi korak kloniranja

Klikom na dugme *Next* dolazi se do izbora tipa klona. U ovom slučaju se može izabrati opcija *Linked clone*, jer ona podrazumeva povezivanje virtuelnog hard diska novokreirane virtuelne mašine na virtuelni hard disk originalne mašine, umesto njegovog kopiranja kao u slučaju *Full* klona, čime se vrši ušteda prostora na računaru. Ovime je zato onemogućeno prebacivanje nove virtuelne mašine na drugi računar bez prebacivanja i originala (slika 4.1.2.9.). Klikom na dugme *Clone* pokreće se završni proces kloniranja virtuelne mašine i vrši se povratak u glavni prozor, gde se može primetiti da se u listi virtuelnih mašina sada nalazi i *Host1*. Postupak se ponavlja i pri kreiranju mašine *Host2*, sa razlikom samo u pisanju naziva mašine, nakon čega je završeno postavljanje mreže koja se testira. Obeležavanjem jedne od virtuelnih mašina, sa desne strane prozora se dobija prikaz njenih parametara (slika 4.1.2.10.).



Slika 4.1.2.9. Odabir tipa klona

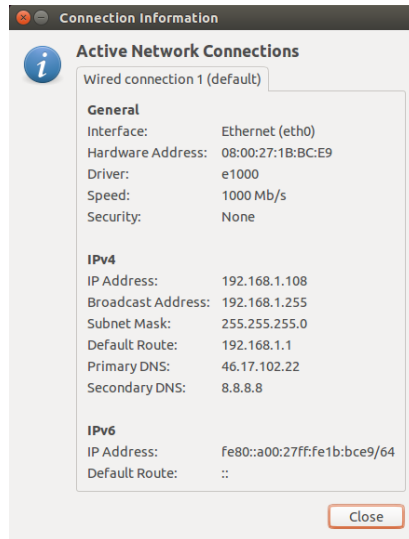


Slika 4.1.2.10. Spisak virtuelnih mašina i parametri

#### 4.1.3. Testiranje međusobne povezanosti virtuelnih mašina

Nakon dodavanja sve tri virtuelne mašine, trebalo bi izvršiti proveru njihove povezanosti međusobnom razmenom *ping* paketa. Da bi ovo bilo uspešno, obe virtuelne mašine čija se veza testira moraju biti pokrenute u isto vreme.

Najpre je neophodno dobiti informacije o IP adresama interfejsa virtuelnih mašina koje su deo ove mreže. To se može realizovati odabirom ikone koja predstavlja konekciju na mrežu u gornjem desnom uglu grafičkog interfejsa *Ubuntu* distribucije, birajući potom iz padajućeg menija opciju *Connection Information* (slika 4.1.3.1.). Drugi način dobijanja navedenih informacija je pokretanjem terminala i unošenjem komande *ip addr* ili *ifconfig*, nakon čega dolazi do ispisa podataka o svim pokrenutim interfejsima na mreži, kao što je prikazano na slici 4.1.3.2.



Slika 4.1.3.1. Prozor *Connection Information*

```

ubuntu@ubuntu-VirtualBox: ~
ubuntu@ubuntu-VirtualBox:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:1b:bc:e9 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.108/24 brd 192.168.1.255 scope global dynamic eth0
        valid_lft 6801sec preferred_lft 6801sec
    inet6 fe80::a00:27ff:fe1b:bce9/64 scope link
        valid_lft forever preferred_lft forever
ubuntu@ubuntu-VirtualBox:~$

```

Slika 4.1.3.2. Informacije o adresama interfejsa

Ovim postupkom dobijene su sledeće IP adrese:

- *Veb\_server*: 192.168.1.108/24
- *Host1*: 192.168.1.106/24
- *Host2*: 192.168.1.112/24

Sada je moguće testirati povezanost virtuelnih mašina *ping* komandom. Prilikom ispitivanja veze između *Host1* i *Veb\_server* mašine, ona bi bila pokrenuta na bilo kojoj od njih i imala sledeći oblik ukoliko bi bila pokrenuta sa mašine *Veb\_server*: *ping -c 4 192.168.1.106*. Navedenom komandom vrši se slanje 4 *ping* paketa. Analogno tome se vrši provera veze između druge host mašine i mašine sa serverom, kao i između dve host mašine, iako to nije od velikog interesa za ovu tezu, ali pokazuje karakteristike *bridged networking* moda. Sa slika 4.1.3.3. i 4.1.3.4. je moguće zaključiti da sve testirane konekcije funkcionišu i da je mreža za simulaciju uspešno postavljena.

```
ubuntu@ubuntu-VirtualBox: ~
ubuntu@ubuntu-VirtualBox:~$ ping -c4 192.168.1.106
PING 192.168.1.106 (192.168.1.106) 56(84) bytes of data.
64 bytes from 192.168.1.106: icmp_seq=1 ttl=64 time=0.672 ms
64 bytes from 192.168.1.106: icmp_seq=2 ttl=64 time=0.752 ms
64 bytes from 192.168.1.106: icmp_seq=3 ttl=64 time=0.644 ms
64 bytes from 192.168.1.106: icmp_seq=4 ttl=64 time=0.414 ms

--- 192.168.1.106 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.414/0.620/0.752/0.128 ms
ubuntu@ubuntu-VirtualBox:~$ ping -c4 192.168.1.112
PING 192.168.1.112 (192.168.1.112) 56(84) bytes of data.
64 bytes from 192.168.1.112: icmp_seq=1 ttl=64 time=0.407 ms
64 bytes from 192.168.1.112: icmp_seq=2 ttl=64 time=0.637 ms
64 bytes from 192.168.1.112: icmp_seq=3 ttl=64 time=0.706 ms
64 bytes from 192.168.1.112: icmp_seq=4 ttl=64 time=0.640 ms

--- 192.168.1.112 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.407/0.597/0.706/0.115 ms
ubuntu@ubuntu-VirtualBox:~$
```

Slika 4.1.3.3. Ping sa virtuelne mašine *Veb\_server*

```
ubuntu@ubuntu-VirtualBox: ~
ubuntu@ubuntu-VirtualBox:~$ ping -c4 192.168.1.112
PING 192.168.1.112 (192.168.1.112) 56(84) bytes of data.
64 bytes from 192.168.1.112: icmp_seq=1 ttl=64 time=0.435 ms
64 bytes from 192.168.1.112: icmp_seq=2 ttl=64 time=0.632 ms
64 bytes from 192.168.1.112: icmp_seq=3 ttl=64 time=0.734 ms
64 bytes from 192.168.1.112: icmp_seq=4 ttl=64 time=0.389 ms

--- 192.168.1.112 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.389/0.547/0.734/0.143 ms
ubuntu@ubuntu-VirtualBox:~$
```

Slika 4.1.3.4. Ping između *Host1* i *Host2*

## 4.2. Pokretanje servera

Po završetku postavljanja mreže virtuelnih mašina, može se preći na dalja podešavanja koja se tiču samog *Linux OS* koji je pokrenut. Prvo od njih je instalacija *Apache* servera na mašini *Veb\_server*, njenim pokretanjem i kucanjem komande *sudo apt-get install apache2*, a zatim i unosom slova *y* kao potvrdu da je korisnik siguran da želi da utroši navedeni deo memorije, nakon čega sledi proces instalacije servera. Postoji i opcija pokretanja komande koja automatski odgovara na pomenuto pitanje: *sudo apt-get install -y apache2*, kako je već pomenuto prilikom instalacije alata *Wireshark*.

Postupak pokretanja servera je nakon instalacije praktično gotov, međutim potrebno je modifikovati korisničke dozvole za pristup direktorijumu */var/www/html/*, u koji će biti smeštena veb stranica korišćena za demonstraciju uspešnog ili neuspešnog pristupa serveru. Ovo je moguće postići komandom *sudo chmod o+w /var/www/html*, koja vrši dodavanje *write* dozvole korisnicima koji nisu kreirali direktorijum niti pripadaju istoj grupi u kojoj se nalazi kreator direktorijuma. U nastavku se na slici 4.2.1. mogu videti rezultati detaljnog listinga direktorijuma

`/var/www/`, prikazujući aktivne dozvole za poddirektorijum `html` pre i posle njihove modifikacije. Tako je omogućeno kreiranje datoteka unutar pomenutog direktorijuma, kao i pristupanje njima, od strane korisnika koji nisu `root` korisnik. Postupak pisanja koda za veb stranicu dat je u daljem delu teksta.

```
ubuntu@ubuntu-VirtualBox: ~
ubuntu@ubuntu-VirtualBox:~$ ls -l /var/www/
total 4
drwxr-xr-x 2 root root 4096 Mar 10 20:07 html
ubuntu@ubuntu-VirtualBox:~$ sudo chmod o+w /var/www/html
ubuntu@ubuntu-VirtualBox:~$ ls -l /var/www/
total 4
drwxr-xrwx 2 root root 4096 Mar 10 20:07 html
ubuntu@ubuntu-VirtualBox:~$
```

Slika 4.2.1. Korisničke dozvole za direktorijum `/var/www/`

### 4.3. Kontrola pristupa serveru

Blokiranje IP adrese virtuelne mašine `Host2` predstavlja ključni deo ove simulacije. Kao što je navedeno u odeljku 2.2.2. ovog rada, u tu svrhu će biti modifikovana konfiguraciona datoteka `Apache` servera. Pošto je na ovoj mašini instaliran tekstualni editor `Vim`, o čemu govori naredni odeljak teksta, može se koristiti u ovom slučaju, pa se odgovarajuća datoteka otvara komandom `sudo vim /etc/apache2/apache2.conf`. S obzirom na to da je potrebno blokirati jednu IP adresu, a dozvoliti pristup svim ostalim, redosled direktiva treba da bude takav da se najpre ispita uslov čijim ispunjavanjem bi se omogućio pristup svim adresama, a zatim uslov direktive `Deny`, kojim se filtrira saobraćaj, odnosno brani pristup mašini `Host2` u ovom primeru. U slučaju obrnutog redosleda obrade uslova, pri pokušaju pristupa virtuelne mašine `Host2`, njena adresa bi ispunila uslov za odbijanje pristupa, ali i naredni uslov koji se ispituje, koji dozvoljava saobraćaj poreklom od bilo koje IP adrese, čime bi zabrana bila „poništena“, zbog načina funkcionisanja ovih direktiva. Izmenjeni deo teksta datoteke sa konfiguracijom je obeležen na slici 4.3.1., uz napomenu da je ove promene potrebno sačuvati i nakon toga restartovati server, komandom `sudo service apache2 restart`.

```
ubuntu@ubuntu-VirtualBox: ~
Require all granted
</Directory>
<Directory /var/www/>
Options Indexes FollowSymLinks
AllowOverride All
Require all granted
Order allow,deny
Allow from all
Deny from 192.168.1.112
</Directory>
#<Directory /srv/>
# Options Indexes FollowSymLinks
# AllowOverride None
# Require all granted
#</Directory>
# AccessFileName: The name of the file to look for in each directory
171,12 79%
```

Slika 4.3.1. Izmene konfiguracije `Apache` servera







## 5. REZULTATI SIMULACIJE

Posle izvršenih postavki simulacije i konfigurisanja mreže, prelazi se na njenu realizaciju i analiziranje rezultata dobijenih korišćenjem alata *Wireshark*.

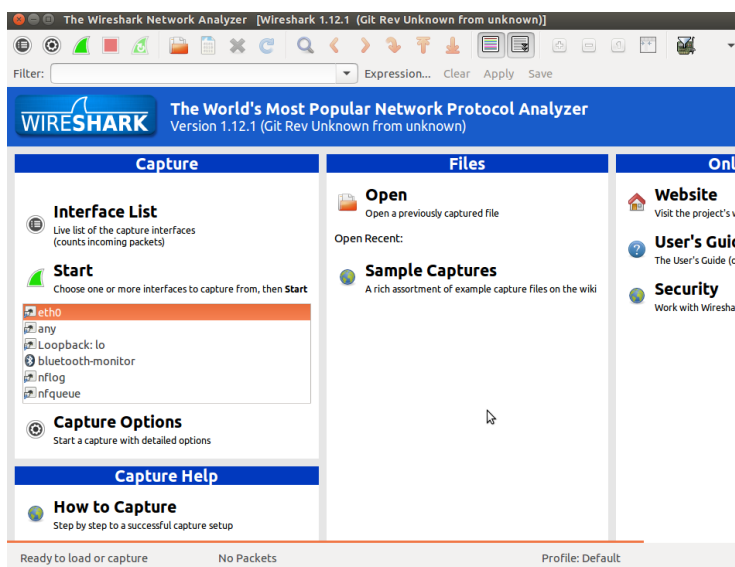
### 5.1. Postavka snimanja saobraćaja

Pre pokretanja simulacije, potrebno je postaviti i početi snimanje *http* saobraćaja alatom *Wireshark* na željenom interfejsu, u cilju sagledavanja poruka koje se razmenjuju pri pokušaju komunikacije host mašina sa veb serverom. Stoga će ovakva snimanja biti puštena na interfejsima virtuelnih mašina *Host1* *Host2* pri pokušaju njihovog pristupa veb serveru, dok će na interfejsu mašine *Veb\_server* ono biti pokrenuto samo pri testiranju pristupa od strane *Host2*. Razlog za ovo je to što se pri takvom testu očekuje da *Host2* može da šalje pakete virtuelnoj mašini *Veb\_server*, ali da neće dobiti odgovor, dok se u slučaju pristupa mašine *Host1* očekuje isti izgled paketa koji je snimljen na host strani veze.

Sve tri virtuelne mašine postavljaju snimanje *http* saobraćaja na isti način. Najpre se *Wireshark* može pokrenuti nakon pretraživanja ovog pojma, kao na slici 5.1.1., čime se otvara grafički interfejs (slika 5.1.2.). Kao što se može videti, spisak interfejsa je sada dostupan korisniku, o čemu je bilo reči u odeljku 4.1.2.

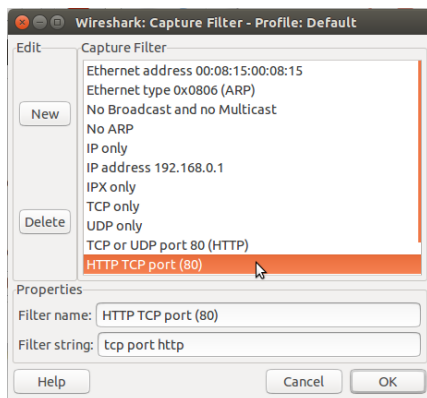


Slika 5.1.1. Pretraga aplikacije *Wireshark*

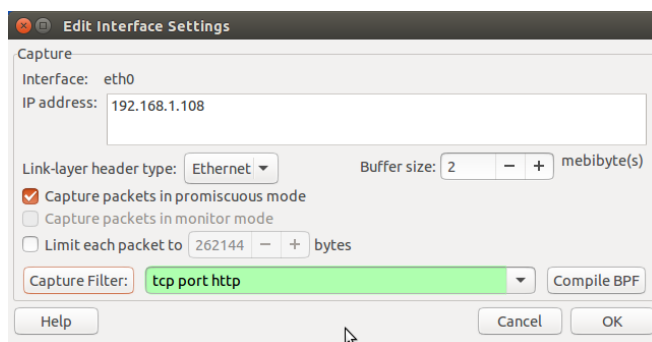


Slika 5.1.2. Grafički interfejs programa *Wireshark*

Sa ovog spiska neophodno je duplim klikom odabrati interfejs na kom je potrebno snimanje paketa, nakon čega se otvara prikaz za podešavanje parametara snimanja. U okviru toga za potrebe ove simulacije dovoljno je izabrati snimanje *http* tipa paketa, biranjem ove opcije nakon klika na dugme *Capture Filter* (slika 5.1.3.), a ostale vrednosti se ne moraju menjati. Potvrdom ovih parametara snimanje je pripremljeno za praćenje simulacije.



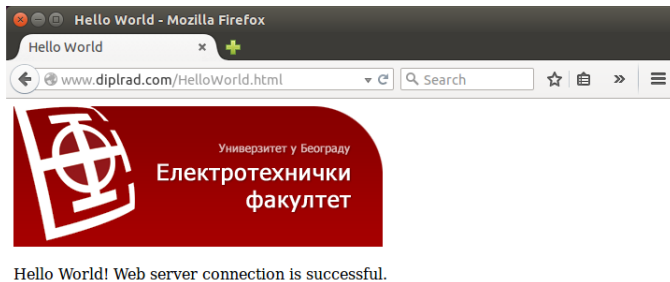
Slika 5.1.3. Biranje filtera paketa



Slika 5.1.4. Postavke za snimanje paketa na interfejsu

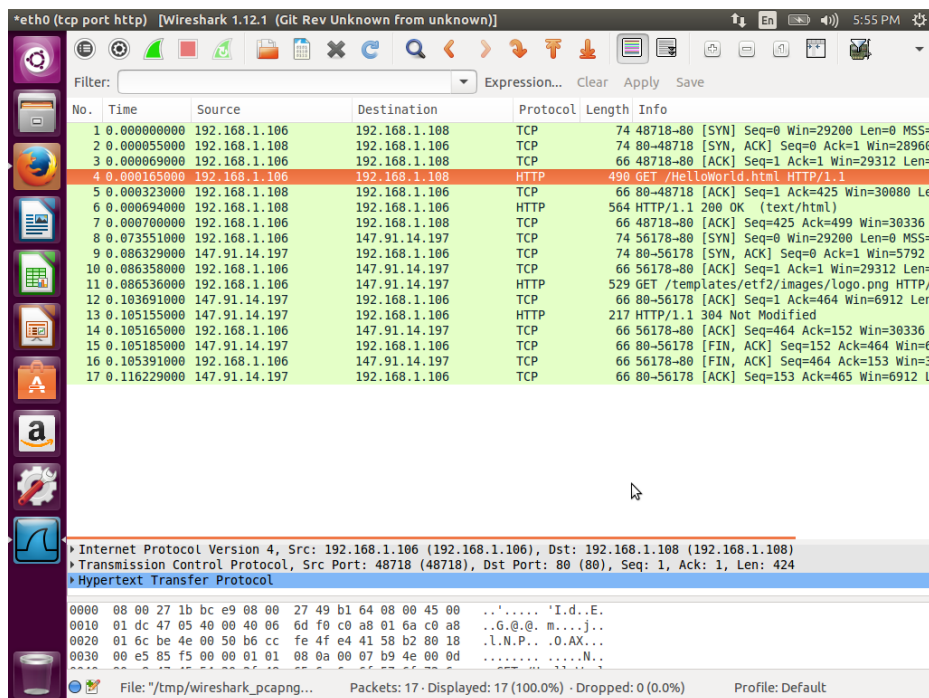
## 5.2. Pristup serveru sa virtuelne mašine *Host1*

Nakon postavke opisane u prethodnom odeljku, sada treba pokrenuti snimanje saobraćaja klikom na dugme *Start* u okviru prethodno opisanog grafičkog prikaza alata *Wireshark* i izvršiti testiranje pristupa veb stranici *HelloWorld.html* na veb serveru, od strane virtuelne mašine *Host1*. Otvaranjem internet pretraživača i upisivanjem adrese *www.diplrad.com/HelloWorld.html*, dolazi do uspešnog pristupa veb stranici kreiranoj na veb serveru pokrenutog na virtuelnoj mašini *Veb\_server*, prikazanog na slici 5.2.1.



Slika 5.2.1. Uspešan pristup veb stranici

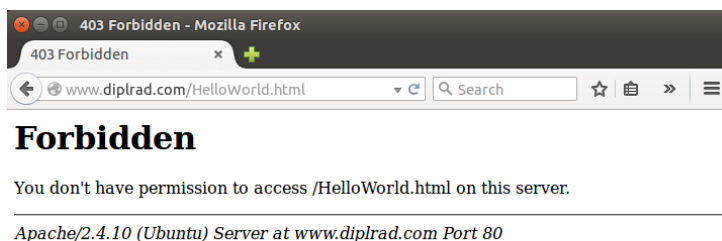
Snimanje saobraćaja se može zaustaviti dugmetom *Stop*, predstavljenim crvenim kvadratom u gornjem delu grafičkog interfejsa alata *Wireshark*, a na narednoj slici se mogu videti dobijeni rezultati. Na osnovu njih se ustanovljuje da je razmena *http* paketa između ove dve virtuelne mašine protekla neometano.



Slika 5.2.2. Paketi snimljeni pri uspešnom pristupu serveru

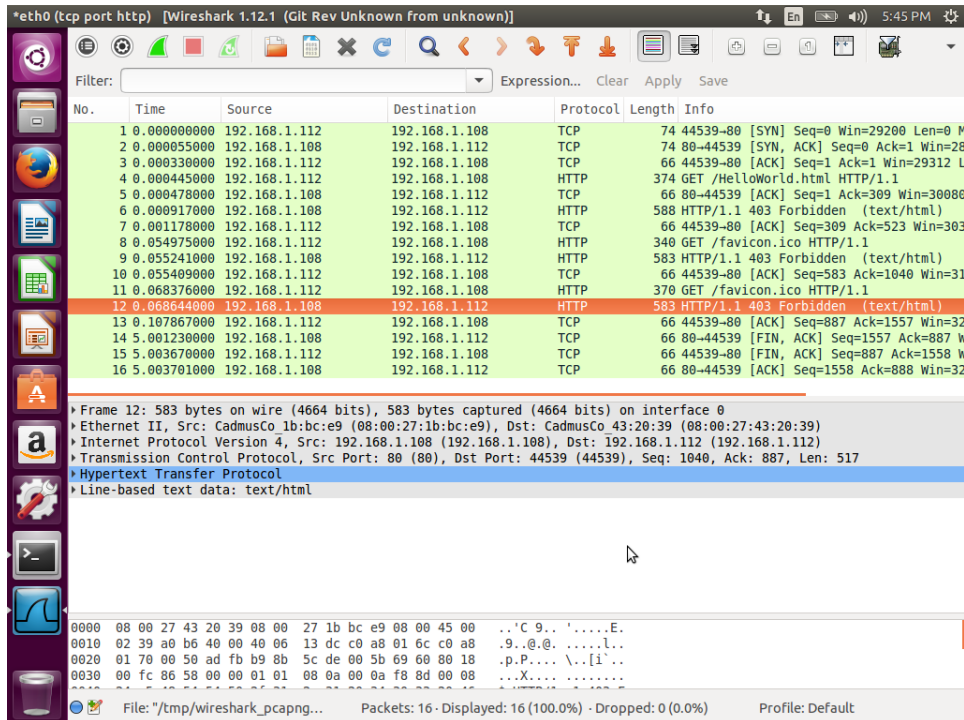
### 5.3. Pristup serveru sa virtuelne mašine *Host2*

Pred početak testiranja ovog slučaja na virtuelnim mašinama *Veb\_server* i *Host2* se pokreće snimanje *http* saobraćaja, uz postavku opisanu u odeljku 5.1. Kao i u prethodnom delu simulacije, u internet pretraživaču se pokreće adresa veb stranice *HelloWorld.html* locirane na veb serveru pokrenutog na mašini *Veb\_server*. Nakon nekog vremena, pretraživač obavestava korisnika o zabrani pristupa veb serveru, izbacujući grešku „403 Forbidden“, što je bilo i očekivano (slika 5.3.1.).

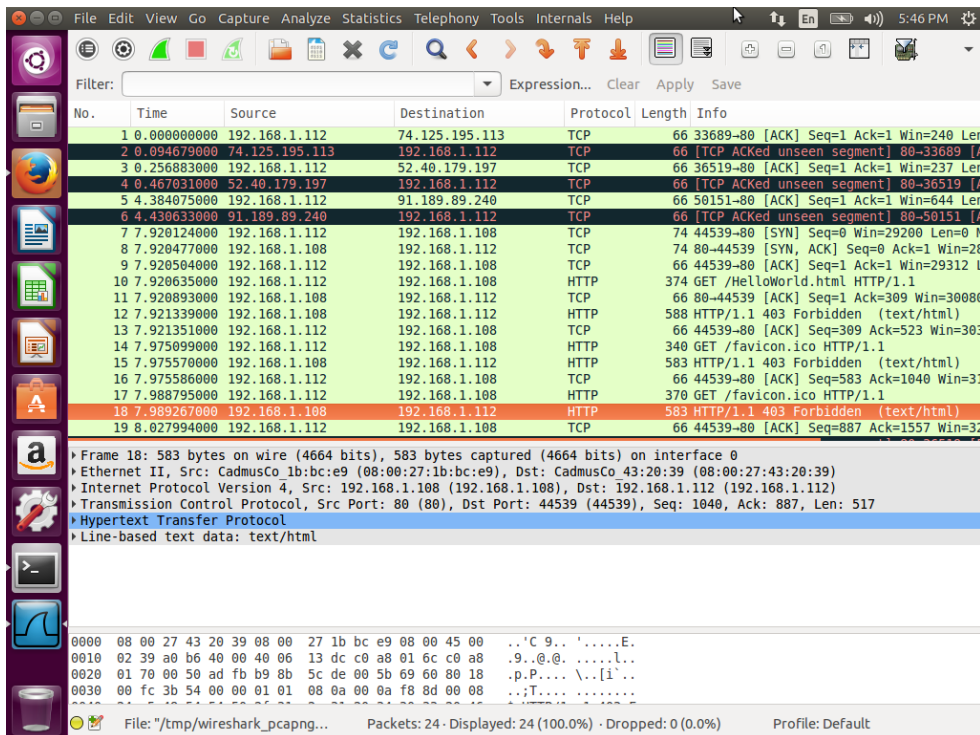


Slika 5.3.1. Zabranjen pristup veb stranici

Snimanje saobraćaja na interfejsima se zaustavlja i dobijeni rezultati su prikazani na narednim slikama. Iz njih se dolazi do zaključka da paketi od *Host2* dolaze do virtuelne mašine *Veb\_server*, ali se host mašini šalju paketi obavestjenja o nedozvoljenom pristupu, zbog uspešne blokade IP adrese.



Slika 5.3.2. Paketi snimljeni sa *Veb\_server* strane

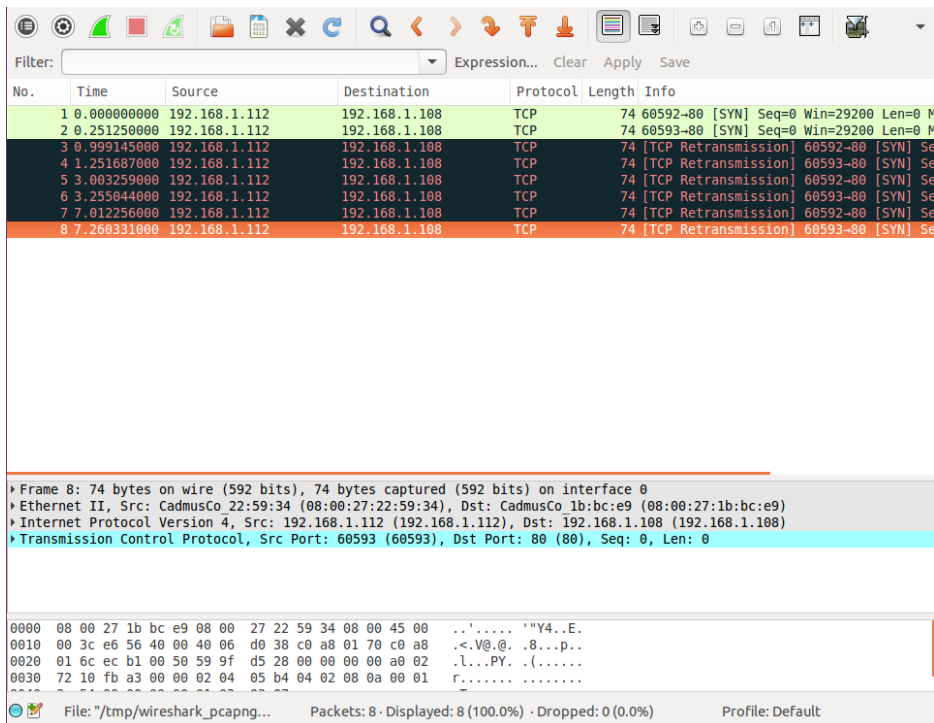


Slika 5.3.3. Paketi snimljeni sa *Host2* strane

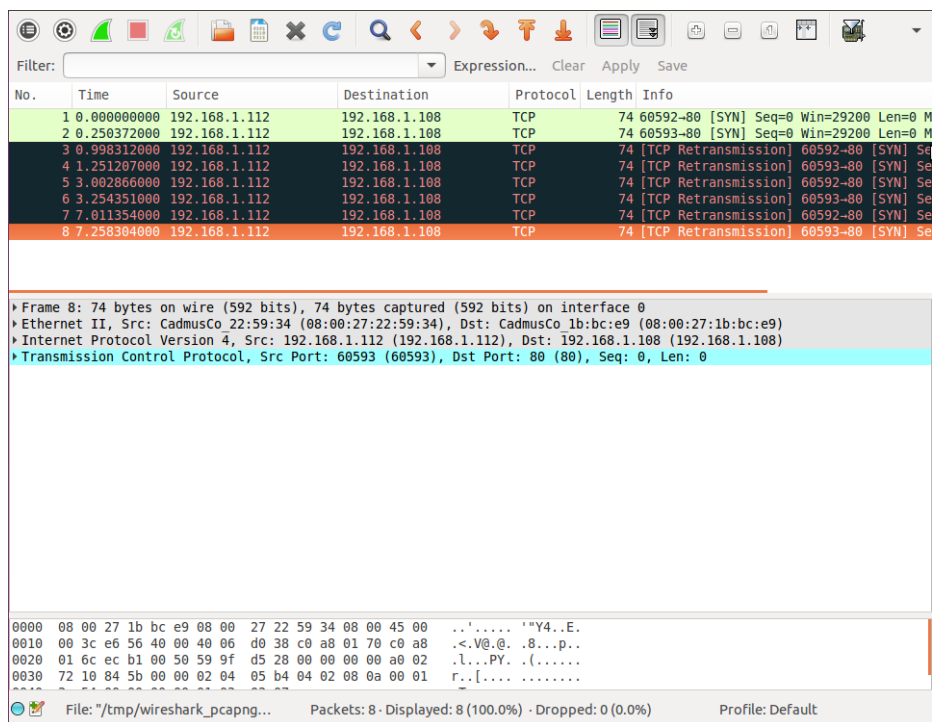
## 5.4. Blokada hosta korišćenjem UFW

U ovom delu teksta ukratko će biti prikazan rezultat simulacije pokušaja pristupa serveru od strane *Host2* u slučaju korišćenja fajervola, na osnovu podešavanja prikazanih na slici 2.2.2.1. Proces simulacije i snimanja paketa isti je kao i u prethodnom odeljku.

Za razliku od prethodno testirane situacije u kojoj je za blokadu pristupa odrađena modifikacija konfiguracione datoteke *Apache* servera, prilikom analize paketa u ovom slučaju može se ustanoviti da paketi poslani sa *Host2* sada stižu do veb servera, ali da sada nema povratnog odgovora na njih. Prikaz snimljenih paketa se može videti na slikama 5.4.1. i 5.4.2.



Slika 5.4.1. Paketi snimljeni sa *Veb\_server* strane



Slika 5.4.2. Paketi snimljeni sa *Host2* strane

## 6. ZAKLJUČAK

Na osnovu prethodno prikazanih rezultata simulacije, može se zaključiti da je cilj teze ispunjen, s obzirom na to da je uspešno izvršena blokada IP adrese jednog host uređaja na dva načina, promenom konfiguracije *Apache* servera i upotrebom fajervola. Ovime je na efikasan način, bez velike upotrebe resursa, testirana jedna svakodnevna situacija, pa bi ovakvo rešenje moglo biti primenjeno i na realne fizičke uređaje.

Treba imati na umu da postoje i drugi načini na koje bi ovakvo ponašanje servera bilo realizovano, kao što je i navedeno u tekstu rada. Takođe, odrađeni primer je skalabilan, pa postoji mogućnost testiranja daleko veće i pravilno organizovane mreže, kao i blokiranja čitavog opsega IP adresa, što mnogim organizacijama može biti od velike koristi. Problem u takvom slučaju jedino se ogleda u tome što fizički uređaj koji implementira takav test može biti usporen nakon određenog broja istovremeno pokrenutih virtuelnih mašina. Takav efekat već je bio ispoljen i pri pokretanju ove simulacije, s obzirom na količinu RAM memorije računara koja je bila upotrebljena. Mogućnosti virtuelizacije su brojne, pa se tako i podešavanja korišćenih mašina mogu menjati u skladu sa potrebama određene simulacije, kao što su količina korišćene RAM memorije, izbor distribucije ili čitavog operativnog sistema, ili način umrežavanja. Imajući to na umu, primer odrađen u okviru ovog rada predstavlja odličnu osnovu za kompleksnije realizacije i dalju edukaciju u oblasti virtuelizacije.

## LITERATURA

- [1] doc. dr Zoran Čiča, Mrežna administracija i programiranje, predavanja, 2016
- [2] <http://www.vmware.com/solutions/virtualization.html>
- [3] [https://www.tutorialspoint.com/internet\\_technologies/web\\_servers.htm](https://www.tutorialspoint.com/internet_technologies/web_servers.htm)
- [4] <https://www.virtualbox.org/manual/UserManual.html>
- [5] <https://en.opensuse.org/VirtualBox>
- [6] [https://en.wikipedia.org/wiki/IP\\_address\\_blocking](https://en.wikipedia.org/wiki/IP_address_blocking)
- [7] <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-14-04>
- [8] [http://httpd.apache.org/docs/1.3/mod/mod\\_access.html#order](http://httpd.apache.org/docs/1.3/mod/mod_access.html#order)
- [9] <https://www.digitalocean.com/community/tutorials/how-to-protect-an-apache-server-with-fail2ban-on-ubuntu-14-04>
- [10] [https://httpd.apache.org/ABOUT\\_APACHE.html](https://httpd.apache.org/ABOUT_APACHE.html)
- [11] <https://www.virtualbox.org/manual/ch06.html>
- [12] <http://encyclopedia2.thefreedictionary.com/Network+stack>
- [13] <http://www.addictivetips.com/ubuntu-linux-tips/15-ubuntu-text-editors-grab-your-favorite/>
- [14] <http://www.wpbeginner.com/glossary/apache/>
- [15] <http://askubuntu.com/questions/52147/how-can-i-access-apache-on-virtualbox-guest-from-host>
- [16] Link za preuzimanje softverskog alata *VirtualBox*:  
<https://www.virtualbox.org/wiki/Downloads>
- [17] Link za preuzimanje „slike“ *Ubuntu* distribucije:  
<https://virtualboxes.org/images/ubuntu/#ubuntu1504>