

Εργαστήριο Δικτύων Υπολογιστών

Εργαστηριακή Άσκηση 2

Δημήτριος Κόγιος

03119220

Όνομα PC: lekog-HP-Laptop-15s-fq1xxx

Άσκηση 1:

1.1) Done.

1.2) Done.

1.3) Done.

1.4) Done.

1.5) Done.

1.6) Μέσω passwd -l root μας ζητάει να γράψουμε τον νέο κωδικό.

1.7) Done.

1.8) Done.

1.9) Done.

1.10) Done μέσω reboot και ps aux | grep sshd.

1.11) Done.

1.12) Μέσω history -c.

1.13) Done.

1.14) Done.

1.15) Done.

Άσκηση 2:

2.1) ifconfig

2.2) ifconfig emo down για απενεργοποίηση.

Αφού την απενεργοποιήσουμε βλέπουμε ότι τα flags UP και RUNNING δεν εμφανίζονται πλέον για τη συγκεκριμένη κάρτα δικτύου.

Μετά μέσω ifconfig emo up τα flags αυτά επανέρχονται.

2.3) man tcpdump, man pcap, man pcap-filter

2.4) tcpdump -i emo -n

2.5) tcpdump -i emo -x : Για hex

tcpdump -i emo -A : Για ASCII

Και οι δύο δείχνουν τα πλαίσια χωρίς όμως το link layer header .

2.6) Μπορούμε να προσθέσουμε το -e στις παραπάνω εντολές για να δούμε και το link layer header.

2.7) tcpdump -i em0 -s 68

-s Αντί για το default αριθμό των 262144 bytes, τυπώνει μόνο τα πρώτα 68 (ή όποιον άλλο αριθμό του δώσουμε) bytes των data των πακέτων.

2.8) tcpdump host 10.0.0.1 -v

Το host φιλτράρει πακέτα προς και από έναν συγκεκριμένο host. Το -v μας δίνει πληροφορίες για το IPv4 header όπως το time to live, identification, total length και options του IPv4 header.

2.9) tcpdump -i em0 '(host 10.0.0.1 and 10.0.0.2)'

2.10) tcpdump '(net 1.1.0.0/16 and ip)'

2.11) tcpdump -e -x '((not net 192.168.1.0/24) and ip)'

2.12) tcpdump '(ip and ether broadcast)'

2.13) tcpdump '(ip[2:2] > 576)'

Θέλουμε να φιλτράρουμε με βάση με το total length που είναι τα bits [15-31] του IPv4 header ή αλλιώς τα bytes [3-4]. Δηλαδή σκιπάρουμε τα πρώτα 2 bytes και ελέγχουμε εάν τα επόμενα 2 (που είναι το total length) έχουν τιμή πάνω από την επιθυμητή.

2.14) tcpdump '(ip[8:1] < 5)'

Αφού το TTL είναι το byte 8 του IPv4 header (με την αρίθμηση να ξεκινάει από το 0) και έχει μήκος 1 byte.

2.15) Θέλουμε το πρώτο byte του IP header να έχει τιμή πάνω από 69 γιατί: τα πρώτα 4 bits του byte ο θα είναι σίγουρα 0100 (version = 4). Άμα το πακέτο δεν έχει Options, τα επόμενα 4 bits είναι 0101 (=5, δηλαδή $5 * 32 / 8 = 20$ bytes). Άρα θέλουμε τιμή μεγαλύτερη του 0x01000101 = 64 + 4 + 1 = 69dec.

Τελικά:

tcpdump '(ip[0:1] > 69)'

Εναλλακτικά θα μπορούσαμε και με μάσκα για τα τελευταία 4 bits δηλαδή: tcpdump '(ip[0] & 0xf > 5)'

2.16) tcpdump '(src host 10.0.0.1 and icmp)'

2.17) tcpdump '(dst host 10.0.0.2 and tcp)'

2.18) tcpdump '(udp and dst port 53)'

2.19) tcpdump '(tcp and host 10.0.0.10)'

2.20) tcpdump '(tcp and port 23 and host 10.0.0.10)' -w sample_capture

2.21) tcpdump '(tcp[tcpflags] & tcp-syn != 0)'

2.22) tcpdump '((tcp[tcpflags] = tcp-syn) or (tcp[tcpflags] & (tcp-syn|tcp-ack) != 0))'

2.23) tcpdump '(tcp[tcpflags] & tcp-fin != 0)'

2.24) Πάει στο byte #12 (η αρίθμηση ξεκινάει από το 0) του TCP header και shift-άρει δεξιά 2 θέσεις τα πρώτα 4 bits αυτού του byte. Πρακτικά τα διαιρεί κατά 4.

Τα πρώτα 4 bits του 13ου byte του TCP header είναι το data offset (το header length του TCP header) δηλαδή αυτή η εντολή διαιρεί το header length κατά 4. Δηλαδή βρίσκει τον αριθμό των bytes του TCP header.

2.25) tcpdump '(((tcp[12:1] & 0xfo) >> 4) > 5)'

Δηλαδή έλεγχος λέξεων. Ή μπορούμε και με έλεγχο bytes:

tcpdump '(((tcp[12:1] & 0xfo) >> 2) > 20)'

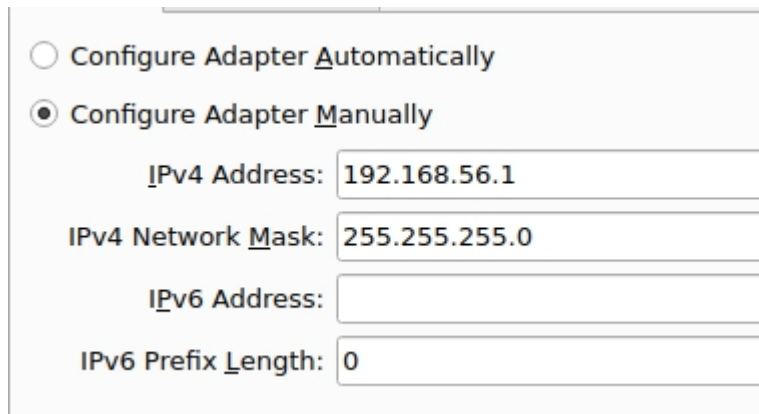
2.26) tcpdump '(port 80)' -A

2.27) tcpdump '(port 23 and dst edu-dy.cn.ntua.gr)'

2.28) tcpdump '(ip6)'

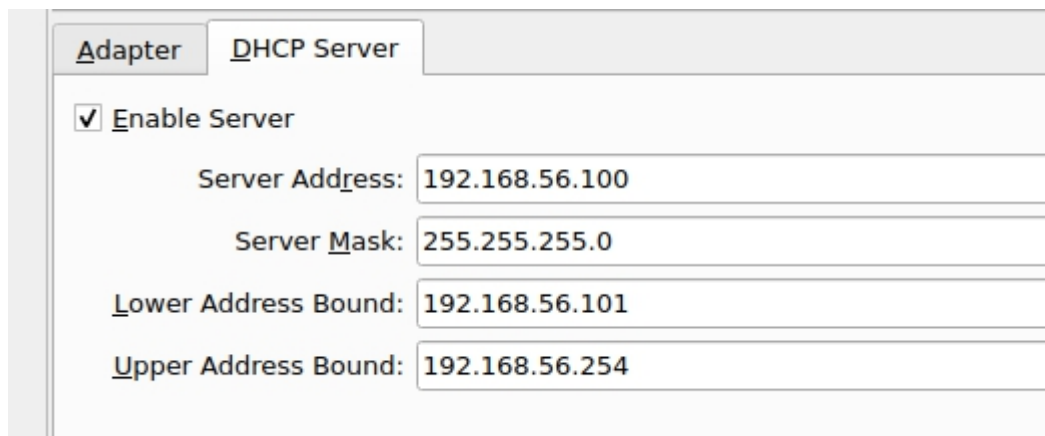
Άσκηση 3:

3.1) 192.168.56.1



The screenshot shows a network configuration window with two radio buttons at the top: "Configure Adapter Automatically" (unselected) and "Configure Adapter Manually" (selected). Below the radio buttons are four input fields: "IPv4 Address:" with the value "192.168.56.1", "IPv4 Network Mask:" with the value "255.255.255.0", "IPv6 Address:" which is empty, and "IPv6 Prefix Length:" with the value "0".

3.2)



The screenshot shows a DHCP Server configuration window with two tabs: "Adapter" and "DHCP Server". The "DHCP Server" tab is selected. Below the tabs is a checkbox labeled "Enable Server" which is checked. Below the checkbox are four input fields: "Server Address:" with the value "192.168.56.100", "Server Mask:" with the value "255.255.255.0", "Lower Address Bound:" with the value "192.168.56.101", and "Upper Address Bound:" with the value "192.168.56.254".

3.3) Μέσω dhclient em0

3.4) Το PC1 πήρε την 192.168.56.102

Το PC2 πήρε την 192.168.56.103

3.5) Μέσω ping. Κάνουμε ping από το ένα στο άλλο και βλέπουμε ότι απαντάει.

3.6) Μέσω ping 192.168.56.1 από τα εικονικά και ping <IP> από το φιλξενούν βλέπουμε ότι απαντάνε.

3.7) netstat -r : display contents of routing tables

3.8) Όχι για είμαστε σε host only. Έχουμε δημιουργήσει ένα τοπικό δίκτυο.

Από τη θεωρία της άσκησης: Εξωτερικά συστήματα δεν μπορούν να επικοινωνήσουν με τα εσωτερικά, εξ ου και η ονομασία, ούτε ορίζεται προκαθορισμένη πύλη.

3.9) Όχι. Δεν ανήκει στο τοπικό δίκτυο και στο host only δεν υπάρχει επικοινωνία με εξωτερικά του δικτύου μηχανήματα. Κάνοντας ping παίρνουμε μήνυμα “no route to host”.

3.10) Μέσω hostname -> PC.ntua.lab

3.11) hostname PC1 και hostname PC2.

3.12) Αλλάζει το prompt σε root@PC1,2::~ #

3.13) Όχι, συνεχίζει να λέει hostname="PC.ntua.lab" άρα σε επανεκκίνηση του PC1 θα πάρει ξανά το όνομα PC.ntua.lab.

3.14) Αλλάζουμε το hostname.

3.15) Όπως μας πληροφορούν τα σχόλια που υπάρχουν στο αρχείο /etc/hosts, πρέπει να προσθέσουμε τις IPv4 διευθύνσεις των PC1 και PC2 στο αρχείο αυτό μαζί με τα aliases.

Δηλαδή, στο αρχείο του PC1 βάζουμε τη γραμμή 192.168.56.103
PC2 και στο αρχείο του PC2 βάζουμε τη γραμμή 192.168.56.102
PC1.

3.16) Αφού συμπληρώσουμε τις γραμμές σε αυτά τα αρχεία,
κάνουμε ping PC1 / PC2 από τα PC2 / PC1

```
root@PC2:~ # ping PC1
PING PC1 (192.168.56.102): 56 data bytes
64 bytes from 192.168.56.102: icmp_seq=0 ttl=64 time=2.034 ms
64 bytes from 192.168.56.102: icmp_seq=1 ttl=64 time=1.557 ms
64 bytes from 192.168.56.102: icmp_seq=2 ttl=64 time=1.361 ms
64 bytes from 192.168.56.102: icmp_seq=3 ttl=64 time=1.083 ms
^C
--- PC1 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.083/1.509/2.034/0.347 ms
root@PC2:~ #
```

3.17) tcpdump -x '(host PC1)' -l | tee test

tcpdump -x '(host PC1)' -l > test & tail -f test

3.18) Μήκος δεδομένων = 56 bytes (άρα μήκος μηνυμάτων ICMP
= 56 + 8 = 64 bytes) και ttl = 64.

3.19) Πάλι ttl = 64.

```
root@PC1:~ # ping -c 4 192.168.56.1
PING 192.168.56.1 (192.168.56.1): 56 data bytes
64 bytes from 192.168.56.1: icmp_seq=0 ttl=64 time=0.538 ms
64 bytes from 192.168.56.1: icmp_seq=1 ttl=64 time=0.820 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=64 time=0.689 ms
64 bytes from 192.168.56.1: icmp_seq=3 ttl=64 time=0.833 ms
--- 192.168.56.1 ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.538/0.720/0.833/0.119 ms
root@PC1:~ #
```


3.20) `tcpdump -vvv '(icmp) -l | tee test`

3.21) Δεν υπάρχει διαφορά (ίσως γιατί το laptop μου έχει linux).

3.22) `ttl = 64` . Δηλαδή το ίδιο με πριν.

3.23) Δεν παρατήρησα καταγραφή.

3.24) Πλέον, αφού βάλαμε την κάρτα δικτύου του PC1 σε promiscuous mode, μπορεί να δει την κίνηση που ακούει στο τοπικό δίκτυο ακόμα και αν δεν αναφέρεται στο PC1.

Άσκηση 4:

4.1) `ifconfig em0 192.168.56.102/24` για το PC1
`ifconfig em0 192.168.56.102/24` για το PC1

4.2) Δεν μου έβγαλε κάποιο μήνυμα λάθους.

4.3) `tcpdump -vvv -l | tee test`

4.4) Όχι, παίρνουμε μήνυμα Destination Host Unreachable

4.5) Βλέπω ARP requests για την IPv4 του PC2.

4.6) Όχι, δεν παίρνουμε απάντηση.

4.7) Όχι.

4.8) Ναι αλλά πρώτα έπρεπε να ξαναορίσω την IPv4 διεύθυνση του PC1 με `ifconfig`.

4.9) Όχι, παίρνουμε μήνυμα Destination Host Unreachable.
Αυτό είναι αναμενόμενο αφού το φιλοξενούν δεν συμμετέχει στο εσωτερικό δίκτυο του Internal Networking.

4.10) tcpdump -n

4.11) arp -d -a

Το PC2 στέλνει ARP requests για να μάθει ποιος έχει την διεύθυνση 192.168.56.1

4.12) Αφού το φιλοξενούν δεν συμμετέχει στο τοπικό δίκτυο, το PC2 δεν παίρνει απάντηση στα ARP requests που στέλνει.

4.13) ifconfig em0 10.11.12.61/26 στο PC1
ifconfig em0 10.11.12.62/26 στο PC2

4.14) Ναι αλλά προφανώς πρέπει να γράψουμε ολογράφως την επιθυμητή διεύθυνση που θέλουμε να ping-άρουμε. Οι συντομογραφίες ping PC1/2 δεν δουλεύουν πλέον αφού έχουν αλλάξει οι διευθύνσεις των μηχανημάτων.

Άσκηση 5:

5.1) dhclient emo

5.2) Πήραν την 10.0.2.15 που τους αποδόθηκε από 10.2.2.2 .

5.3) Μέσω netstat -r βλέπουμε ότι η default gateway έχει IPv4 διεύθυνση 10.0.2.2 .

5.4) Μέσω cat /etc/resolv.conf διαβάζουμε nameserver 10.0.2.3 που είναι ο DNS server που προσφέρει το NAT.

5.5) Στο /var/db/dhclients.leases.emo

5.6) Ναι μέσω ping 10.0.2.2 .

5.7) Ναι μέσω του default gateway. Για παράδειγμα κάνοντας ping www.google.com βλέπουμε ότι παίρνουμε απάντηση.

5.8) 10.0.2.1 δεν απαντάει

10.0.2.2 απαντάει. Είναι το φιλέξενούν το οποίο δρα ως default gateway αλλά και ως DHCP server.

10.0.2.3 απαντάει. Είναι ο proxy DNS εξυπηρετητής.

10.0.2.4 απαντάει. Είναι ο εξυπηρετητής tftp για εκκίνηση του φιλεξιενούμενου μηχανήματος από το δίκτυο.

5.9) Όχι, κάθε εικονικό μηχάνημα είναι σαν να βρίσκεται σε δικό του ξεχωριστό δίκτυο.

5.10) -I: use ICMP ECHO instead of UDP datagrams. Όπως θυμόμαστε από το προηγούμενο εξάμηνο , η traceroute στα Linux (και προφανώς και σε άλλα παρόμοια ΛΣ όπως το

FreeBSD) αντί για ICMP μηνύματα στέλνει UDP datagrams
οπότε με αυτό το flag δηλώνουμε ότι θέλουμε αντί για UDP
datagrams να χρησιμοποιήσουμε ICMP μηνύματα

-n: Print hop addresses numerically

-q <num>: Αντί για το default των 3 probes per hop στείλε num.
(δηλαδή ο αριθμός των requests που θα σταλούν πριν αυξηθεί το
TTL)

5.11) Είναι ICMP echo request με πηγή το 10.0.2.15.

5.12) Είναι ICMP echo request με πηγή το 10.3.22.33 (αυτή είναι
η διεύθυνση IPv4 της φυσικής μου κάρτας καθώς βρίσκομαι στις
NΦΕΕΜΠ και όχι στο δίκτυο του σπιτιού μου).

5.13) Το πρώτο: 10.3.22.1

Το δεύτερο: 62.217.77.8

Το τρίτο: 176.126.38.5

Το τέταρτο request φτάνει πριν μηδενιστεί το TTL του.

5.14) Είναι 10.3.22.33 που όπως είπαμε είναι η διεύθυνση IPv4 της
φυσικής κάρτας του υπολογιστή μου.

5.15) Το πρώτο: 10.0.2.2

Το δεύτερο: 10.3.22.1

Το τρίτο: 62.217.77.8

Το τέταρτο: 176.126.38.5

Το πέμπτο φτάνει πριν μηδενιστεί το TTL του.

5.16) Είναι 10.0.2.15 δηλαδή η διεύθυνση IPv4 του εικονικού
μηχανήματος PC3.

5.17) Ναι εκτός του πρώτου. Οι διευθύνσεις προορισμού είναι διαφορετικές αφού στο Wireshark έχουν ως προορισμό τη φυσική κάρτα του υπολογιστή μου ενώ στο tcpdump έχουν ως προορισμό το PC3. Αυτό γίνεται λόγω του NAT.

5.18) (Κάνω traceroute αντί για tracert αφού έχω Linux)
Είναι 4 hops. Ο λόγος είναι ότι ανάμεσα στο εικονικό μηχάνημα και το 1.1.1.1 παρεμβάλλεται και ένας ακόμα δρομολογητής, το default gateway 10.0.2.2 που είναι το φιλοξενούν μηχάνημα. Οπότε είναι λογικό που από το φιλοξενούν χρειαζόμαστε ένα hop λιγότερο.

Άσκηση 6:

6.1) Είναι 10.0.2.0/24 .

6.2) ifconfig em0 delete και rm.

6.3) dhclient em0

6.4) Το PC1 πήρε 10.0.2.15 (ίδια με πριν) ενώ το PC2 πήρε 10.0.2.4 (διαφορετικό αφού πριν είχε 10.0.2.15).

6.5) Είναι 10.0.2.3 αφού αυτό μας κάνει DHCP offer.

6.6) Μέσω cat /etc/resolv.conf διαβάζουμε nameserver 10.0.2.1 .

6.7) Μέσω netstat -r βλέπουμε ότι default gateway: 10.0.2.1.

6.8) Ναι μέσω ping 10.0.2.1 λαμβάνουμε απάντηση.

6.9) Ναι μέσω ping 10.0.2.3 λαμβάνουμε απάντηση.

6.10) Ναι μέσω ping 10.0.2.2. Απαντά το ίδιο μηχάνημα με την IPv4 διεύθυνση 10.0.2.1 (βλέπουμε ότι η .2 και η .1 έχουν ίδια MAC διεύθυνση). Όπως γνωρίζουμε, στο NAT Network και οι 2 αυτές IPv4 διευθύνσεις ανήκουν στο φιλοξενούν.

6.11) Ναι πχ μέσω ping www.google.com λαμβάνουμε απάντηση. Αυτό είναι αναμενόμενο αφού στο NAT Network (όπως και στο σκέτο NAT της άσκησης 5), το φιλοξενούν δρα ως default gateway του τοπικού δικτύου που δημιουργείται.

6.12) Ναι μπορούμε από το ένα να ping-άρουμε το άλλο.

6.13) Όχι αφού NAT != NAT Network.

6.14) Από το PC3 κάνοντας ping το 10.0.2.15 αλλά και το 10.0.2.4 παίρνουμε απάντηση η οποία όμως δεν προέρχεται από τα PC1 και PC2 (αλλά από τον εαυτό του στην περίπτωση του .15 και τον tftp server στην περίπτωση .4) .

Καταλαβαίνουμε ότι δεν προέρχονται από τα PC1 και PC2 αφού τρέχοντας tcpdump στα PC1 και PC2 δεν βλέπουμε κίνηση κάνοντας ping από το PC3 (επίσης έχοντας ανοιχτά τα παράθυρα που τρέχουν τα PC1 και PC2 δεν αναβοσβήνει τα λαμπάκι του 4ου εικονιδίου (Network)).

Αντιθέτως, άμα το PC1/2 κάνει ping το PC2/1 τότε έχοντας το tcpdump να τρέχει βλέπουμε icmp echo requests και replies (επίσης αναβοσβήνουν τα λαμπάκια δείχνοντάς μας ότι υπάρχει επικοινωνία μεταξύ τους).