

ΑΔΚΕ

Εργασία Εξαμήνου

Κόκθη Μπρούνο

Πιπιλίκας Ευάγγελος

Χατζηβουκούδη Μυρτώ

Περιεχόμενα

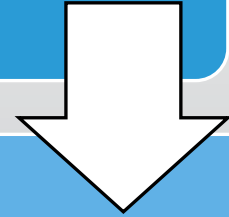
- Θέμα εργασίας
- Στόχοι εργασίας
- Τρόπος υλοποίησης
- Αναπαράσταση εκτέλεσης αλγορίθμου
- Σύγκρισή με εφαρμογή αξιολόγησης καναλιού

Θέμα εργασίας:

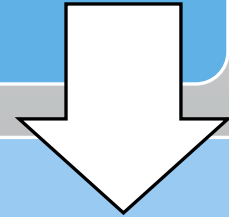
***Επιλογή καναλιών βάσει φόρτου
σε ασύρματα δίκτυα WIFI***

Στόχοι εργασίας

Λήψη πολλών δεδομένων
αυτοματοποιημένα σχετικά
με τον φόρτο καναλιών



Δημιουργία εμπειρικού
κανόνα για την αξιολόγηση
του κάθε καναλιού



Εύρεση καταλληλότερων
καναλιών βάσει του
λιγότερου φόρτου

Αρχικές σκέψεις πάνω στο θέμα

Δύο βασικές εναλλακτικές:

A. Χρήση λογισμικού για την εκτέλεση μετρήσεων

B. Χρήση κατάλληλων διεπαφών λειτουργικών συστημάτων (Linux)

Τι ακολουθήσαμε και γιατί; Χρήση λογισμικού για την εκτέλεση μετρήσεων

Βασικοί αποτρεπτικοί παράγοντες:


- Δυσκολία δημιουργίας αυτοματοποιημένης αποθήκευσης δεδομένων
- Παροχή λογισμικών επί πληρωμή
- Χρήση παραπάνω από δύο λογισμικών για την υλοποίηση του αλγορίθμου μας

Τι ακολουθήσαμε και γιατί;

KALI Linux (virtual machine)

Λόγοι χρήσης έναντι λογισμικού:

- Υψηλή προσβασιμότητα σε αρκετές λειτουργίες του hardware έναντι άλλων λειτουργικών συστημάτων (Windows)
- Ευελιξία χρήσης διάφορων εργαλείων όπως airmmon-ng, airodump κλπ
- Εύκολη σάρωση πακέτων σε χαμηλότερο επίπεδο
- Εύκολη αυτοματοποίηση των μετρήσεων ώστε να δρομολογούνται σε συγκεκριμένα χρονικά διαστήματα



Βασικά βήματα

1^ο Βήμα: Χρήση Wifi interface KALI Linux (via virtual machine)

1

1^ο βασικό πρόβλημα:
Δυσκολία ενεργοποίησης
wifi σε virtual machine

2

2^ο βασικό πρόβλημα:
Δυσκολία ενσωμάτωσης wifi
adapters που υποστηρίζουν
Monitor Mode

3

3^ο βασικό πρόβλημα:
Δυσκολία ενεργοποίησης
του Monitor Mode στους wifi
adapters

2^ο Βήμα: Εύρεση κατάλληλων εργαλείων για λήψη μετρήσεων

Airmon-ng

Προεγκαταστημένο εργαλείο των Linux για αλλαγή του wifi adapter από Managed mode σε Monitor mode

Iwconfig

Προεγκαταστημένο εργαλείο των Linux για λήψη διάφορων πληροφοριών με χρήση μίας ασύρματης διεπαφής

Python3 Scapy library

Βιβλιοθήκη που υποστηρίζει ανίχνευση πακέτων 802.11 πρωτοκόλλου

Airodump-ng

Σκανάρισμα δικτύων και παροχή πληροφορίας σχετικά με την κινητικότητα (data frames)

Crontab

Αυτοματοποίηση σκαναρίσματος σε συγκεκριμένες ώρες

Airmon-ng

Σε τι μας βοήθησε;

Χρήση για αλλαγή της λειτουργίας του wifi adapter από Managed mode σε Monitor mode.

```
~# airmon-ng start wlan0
```

PID Name

718 NetworkManager

PHY Interface	Driver	Chipset
---------------	--------	---------

phy0	wlan0	rt2888usb	Ralink Technology, Corp. RT5370
------	-------	-----------	---------------------------------

(mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)

(mac80211 station mode vif disabled for [phy0]wlan0)

Τι είναι το Monitor mode;

Το Monitor mode πρόκειται για μία λειτουργία ορισμένων wifi adapters που επιτρέπει σε έναν υπολογιστή να σκανάρει οποιαδήποτε κίνηση περνάει από το σημείο.

Το Managed mode πρόκειται για την κανονική λειτουργία των wifi adapters που επιτρέπει σε έναν υπολογιστή να σκανάρει την κίνηση στο δίκτυο του μόνο εφόσον συνδεθεί σε αυτό μέσω του Access point.

Iwconfig

Σε τι μας βοηθάει;

- Παροχή ασύρματης διεπαφής για λήψη διαφόρων πληροφοριών
- Πληθώρα παραμέτρων ανάλογα με τις ανάγκες μας

Στην συγκεκριμένη περίπτωση, θέλουμε για ένα συγκεκριμένο χρονικό διάστημα να σκανάρει ένα συγκεκριμένο κανάλι.

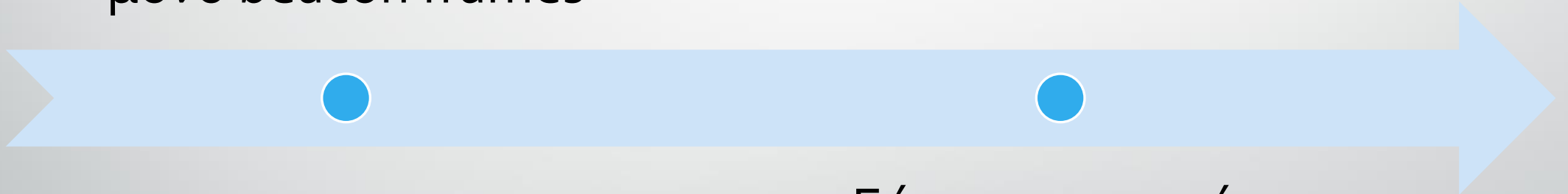
```
iwconfig {interface} channel {ch}
```

Python3 Scapy library

Σε τι μας βοήθησε;



Χρήση για ανίχνευση
μόνο beacon frames



Εύρεση κοντινών
δικτύων

Python3 Scapy library

Πως το χρησιμοποιήσαμε;



Χρήση pandas για διευκόλυνση αποθήκευσης δεδομένων.

Η πληροφορία που αποθηκεύουμε για το κάθε δίκτυο είναι η εξής:

```
networks = pandas.DataFrame(columns=["BSSID", "SSID",  
"dBm_Signal", "Channel", "Crypto"])
```

1. Σκανάρουμε το κάθε κανάλι για μισό δευτερόλεπτο με την βοήθεια του iwconfig
2. Την ίδια στιγμή καταγράφουμε όλες τις πληροφορίες για τα πακέτα που έχουμε ανιχνεύσει και είναι beacon frames.
3. Αφού σκαναριστούν και τα 14 κανάλια, γράφουμε όλα τα στοιχεία σε ένα .csv αρχείο.

Python3 Scapy library

Μορφή αρχείου networks.csv



~/Desktop/adke/networks.csv [Read Only] - Mousepad

File Edit Search View Document Help

1	BSSID	SSID	dBm_Signal	Channel	Crypto	
2	b8:be:f4:		devolo-094	-71	1	{'WPA2/PSK'}
3	b0:4e:26:		VASAGGROU_EXT	-39	1	{'WPA2/PSK', 'WPA/PSK'}
4	e4:26:86:		VASAGGROU	-73	1	{'WPA2/PSK'}
5	e4:26:86:		COSMOTE WiFi Fon	-75	1	{'OPN'}
6	e0:0e:e4:		COSMOTE WiFi Fon	-71	3	{'OPN'}
7	3c:98:72:		COSMOTE WiFi Fon	-75	8	{'OPN'}
8	e4:26:86:		Merm wifi	-67	9	{'WPA2/PSK'}
9	e4:26:86:		COSMOTE WiFi Fon	-69	9	{'OPN'}
10	e0:0e:e4:		COSMOTE	-69	3	{'WPA2/PSK'}
11	cc:32:e5:		COSMOTE-791959_EXT	-75	9	{'WPA2/PSK', 'WPA/PSK'}
12						

Airodump-ng

Σε τι μας βοήθησε;

Χρήση πληροφορίας σχετικά με τα κοντινά δίκτυα που λάβαμε από το Scapy

Για καθένα από τα δίκτυα, αναλύει την κίνηση του για 30 δευτερόλεπτα και γράφει την πληροφορία σε ένα αρχείο.

```
airodump-ng --bssid {BSSID} --essid {SSID} -c {Channel} -w  
network --output-format csv wlan0mon
```

Για την χρήση εντολών περιβάλλοντος Linux μέσα σε script Python, γίνεται η χρήση της βιβλιοθήκης subprocess και μετέπειτα χρήση της συνάρτησης Popen.

Η τελευταία εκτελεί αυτούσιο τον κώδικα σαν να τον τρέχαμε από terminal.

```
sub.Popen(['airodump-ng', '--bssid', row["BSSID"], '--essid',  
row["SSID"], '-c', str(row["Channel"]), '-w', "network", '--  
output-format', 'csv', 'wlan0mon'])
```

Airodump-ng

Πως το χρησιμοποιήσαμε;

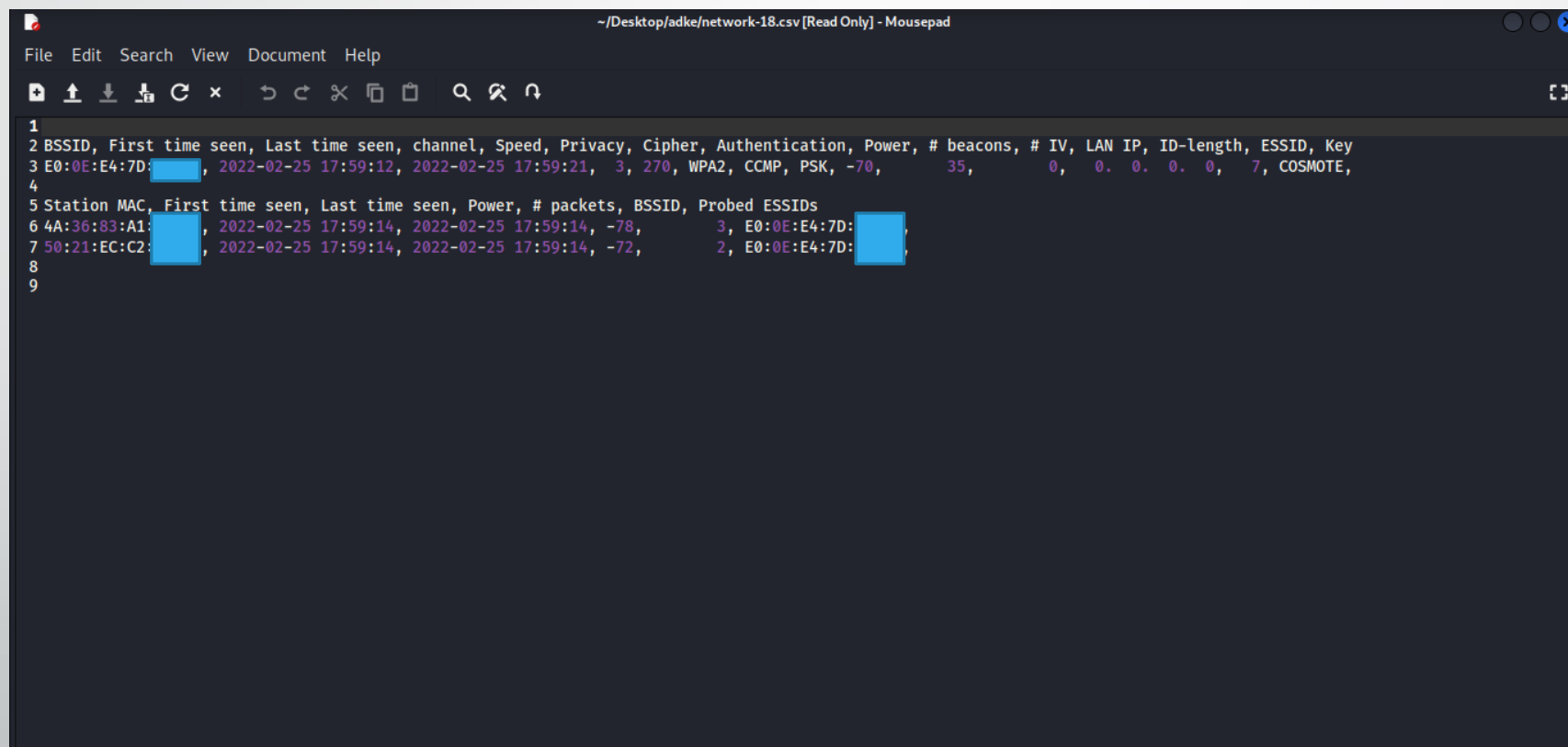
Για καθένα από τα δίκτυα που έχουμε αποθηκευμένα στο αρχείο `networks.csv`, κάνε το εξής:

1. Χρησιμοποιώντας την μορφή της εντολής που δείξαμε πριν, σκάνανε το κάθε δίκτυο για 30 δευτερόλεπτα
2. Γράψε όλη την πληροφορία σε `.csv` αρχείο

Η δημιουργία ενός subprocess παραπέμπει σε thread, το οποίο με χρήση της μεθόδου `time.sleep(30)` το «κοιμίζουμε» και με το πέρας αυτού του διαστήματος το τερματίζουμε (kill).

Airodump-ng

Μορφή αρχείων network-x.csv

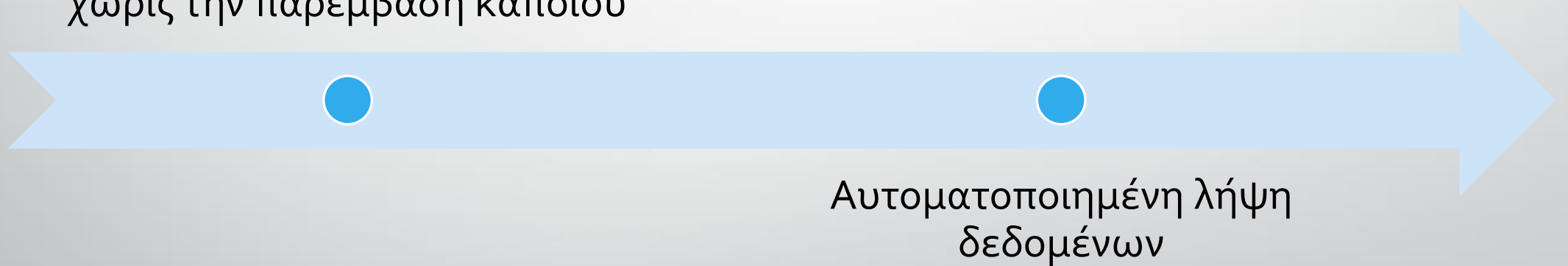


```
~/Desktop/adke/network-18.csv [Read Only] - Mousepad
File Edit Search View Document Help
📄 ⬆️ ⬇️ 📄 ↺️ ⏪ ⏩ ✂️ 📄 🗑️ 🔍 🔍 🔍
1
2 BSSID, First time seen, Last time seen, channel, Speed, Privacy, Cipher, Authentication, Power, # beacons, # IV, LAN IP, ID-length, ESSID, Key
3 E0:0E:E4:7D: [REDACTED], 2022-02-25 17:59:12, 2022-02-25 17:59:21, 3, 270, WPA2, CCMP, PSK, -70, 35, 0, 0. 0. 0. 0, 7, COSMOTE,
4
5 Station MAC, First time seen, Last time seen, Power, # packets, BSSID, Probed ESSIDs
6 4A:36:83:A1: [REDACTED], 2022-02-25 17:59:14, 2022-02-25 17:59:14, -78, 3, E0:0E:E4:7D: [REDACTED],
7 50:21:EC:C2: [REDACTED], 2022-02-25 17:59:14, 2022-02-25 17:59:14, -72, 2, E0:0E:E4:7D: [REDACTED],
8
9
```

Crontab

Σε τι μας βοήθησε;

Λήψη δεδομένων σε
συγκεκριμένες ώρες της ημέρας
χωρίς την παρέμβαση κάποιου



Crontab

Πως το χρησιμοποιήσαμε;

Στο αρχείο crontab περιέχονται οι εντολές με την παρακάτω μορφή:

```
# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name  command to be execute
```

Για να τρέξει το script μας σε διαφορετικές ώρες, αρκεί να προσθέσουμε τους κατάλληλους κανόνες στο αρχείο.

3^ο Βήμα: Συλλογή διάφορων μετρήσεων σε αρχεία

Έχουμε αποθηκεύσει σε αρχεία csv τα παρακάτω:

- Δίκτυα που λάβαμε τα beacon frames τους
- Πληροφορία σχετικά με την κίνηση του κάθε δικτύου

4^ο Βήμα: Επεξεργασία δεδομένων

Από τα αρχεία με την κίνηση κάθε δικτύου θέλουμε να εξάγουμε τα εξής:

- Κινητικότητα πακέτων σε κάθε κανάλι
- Ενεργά δίκτυα σε κάθε κανάλι
- Ενεργοί σταθμοί σε κάθε κανάλι

Η παραπάνω πληροφορία αποθηκεύεται σε αντικείμενα κατάλληλων κλάσεων (class Channel και class Network)

5^ο Βήμα: Εύρεση αλγορίθμου

Χρήση δύο αλγορίθμων:

- Αλγόριθμος εκτίμησης καναλιού
- Αλγόριθμος εκτίμησης γειτονικών καναλιών

Αλγόριθμος εκτίμησης καναλιού

Για καθένα από τα κανάλια που λάβαμε κίνηση, υπολογίζουμε:

- Το ποσοστό πακέτων σε σχέση με τα υπόλοιπα (packets)
- Το ποσοστό ενεργών δικτύων σε σχέση με τα υπόλοιπα (networks)
- Το ποσοστό ενεργών σταθμών σε σχέση με τα υπόλοιπα (stations)

Έπειτα, η φόρμουλα που ακολουθούμε είναι η παρακάτω:

$$channel_{evaluator} = 0.45 * packets + 0.35 * networks + 0.2 * stations$$

Αλγόριθμος εκτίμησης γειτονικών καναλιών

Έχοντας υπολογίσει όλες τις εκτιμήσεις για όλα τα κανάλια, προβαίνουμε στον έλεγχο επίδρασης τους από τυχών επιβαρυμένα γειτονικά κανάλια.

Για καθένα κανάλι, λαμβάνουμε το ποσοστό εκτίμησης αυτού ($channel_i$), του αμέσως προηγούμενου ($channel_{i-1}$) και του αμέσως επόμενου του ($channel_{i+1}$).

Η φόρμουλα που ακολουθούμε είναι η εξής:

$$channel_{neighbor} = 0.7 * channel_i + 0.3 * (0.5 * channel_{i-1} + 0.5 * channel_{i+1})$$

5^ο Βήμα: Διατήρηση ιστορικού

Η αυτοματοποίηση συμβαίνει τρεις φορές την ημέρα, όπου για κάθε μία από αυτές επιθυμούμε να κρατήσουμε ένα ιστορικό.

Ώρα	Αρχείο ιστορικού
12:00	morning_measurements.txt
16:00	noon_measurements.txt
20:00	night_measurements.txt

Κάθε φορά που φτάνει η ώρα λήψης μετρήσεων ακολουθούνται τα εξής βήματα:

1. Λάβε τα καινούρια δεδομένα από το σκανάρισμα ($evaluations_{new}$)
2. Φόρτωσε το ιστορικό από το κατάλληλο αρχείο ($evaluations_{old}$)
3. Γράψε το νέο ιστορικό στο κατάλληλο αρχείο ακολουθώντας την παρακάτω φόρμουλα:

$$history_{evaluations} = 0.8 * evaluations_{new} + 0.2 * evaluations_{old}$$

6° Βήμα: Δημιουργία script

script.sh

Συνδυασμός όλων των προηγούμενων scripts προκειμένου να εκτελούνται αυτόματα και με την σειρά την οποία πρέπει.

```
#!/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/home/apipilikas/Desktop/adke

python3 wifi_scanner.py wlan0mon
python3 wifi_data_sniffer.py
python3 evaluator.py
rm *.csv
```

Η τελευταία εντολή εκτελείται προκειμένου να διαγράφει όλα τα αρχεία .csv που δημιουργούνται κάθε φορά που τρέχουμε τον script.

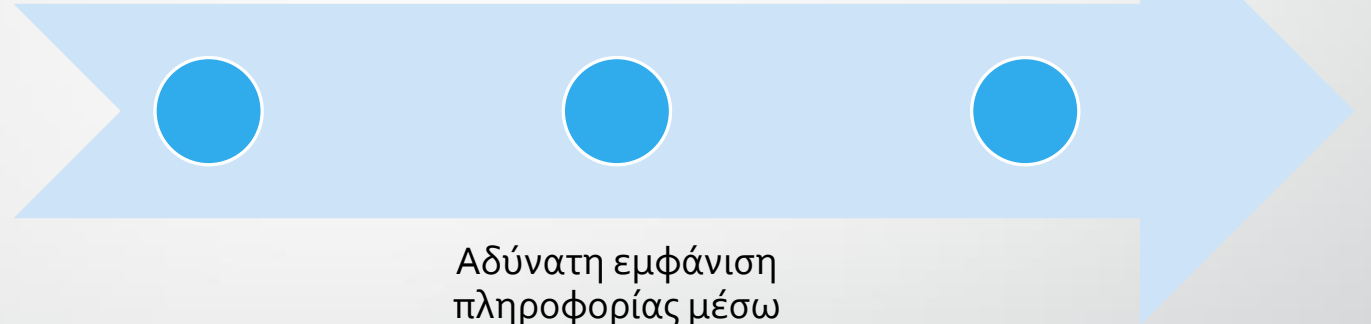
Τέλος, προσθέτοντας τις παρακάτω εντολές στο αρχείο crontab, επιτυγχάνεται η αυτοματοποιημένη λήψη δεδομένων.

7^ο Βήμα: Λήψη συμπεράσματος

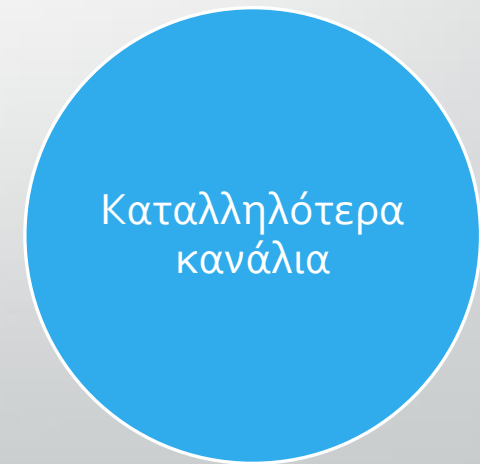
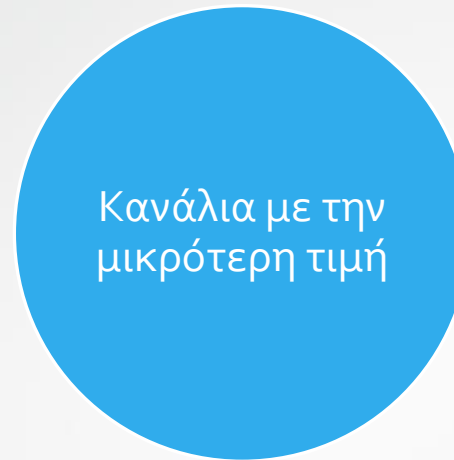
Αυτοματοποιημένη
διαδικασία

Δημιουργία script
για εμφάνιση
καταλληλότερων
καναλιών βάσει
φόρτου

Αδύνατη εμφάνιση
πληροφορίας μέσω
terminal

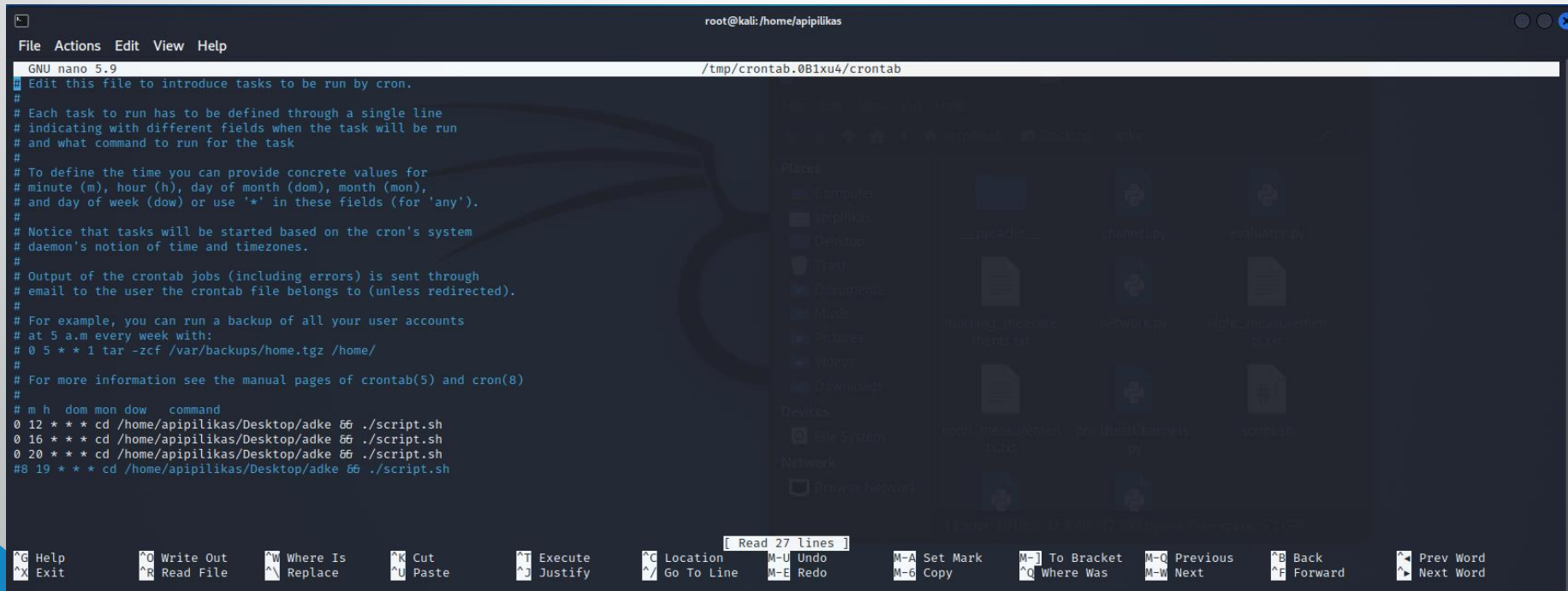


8^ο Βήμα: Λήψη συμπεράσματος



Αναπαράσταση εκτέλεσης αλγορίθμου

Ο αλγόριθμος εκτελείται όταν η ώρα είναι 12:00, 16:00 ή 20:00. Αυτό επιτυγχάνεται με τους παρακάτω κανόνες που είναι γραμμένοι στο αρχείο crontab:



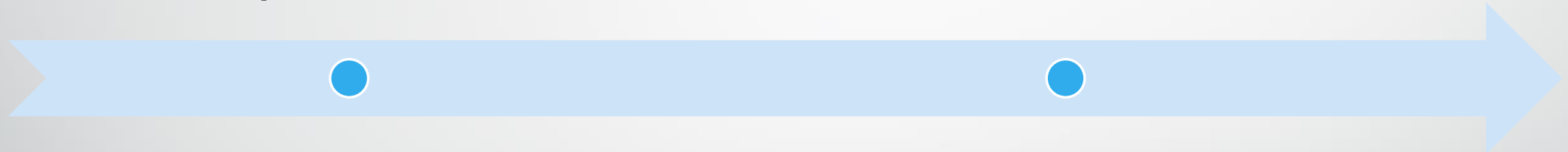
The screenshot shows a terminal window with the nano text editor open. The title bar indicates the user is root at kali, in the directory /home/apipilikas. The editor is editing the file /tmp/crontab.0B1xu4/crontab. The content of the file is as follows:

```
GNU nano 5.9 /tmp/crontab.0B1xu4/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 12 * * * cd /home/apipilikas/Desktop/adke && ./script.sh
0 16 * * * cd /home/apipilikas/Desktop/adke && ./script.sh
0 20 * * * cd /home/apipilikas/Desktop/adke && ./script.sh
#8 19 * * * cd /home/apipilikas/Desktop/adke && ./script.sh
```

The bottom of the window shows a status bar with various keyboard shortcuts for nano, such as ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^M-U Undo, ^M-A Set Mark, ^M-] To Bracket, ^M-Q Previous, ^B Back, and ^X Exit.

Αναπαράσταση εκτέλεσης αλγορίθμου

Ώρα 20:00



Πυροδότηση script.sh

```
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 12 * * * cd /home/apipilikas/Desktop/adke && ./script.sh
0 16 * * * cd /home/apipilikas/Desktop/adke && ./script.sh
0 20 * * * cd /home/apipilikas/Desktop/adke && ./script.sh
#8 19 * * * cd /home/apipilikas/Desktop/adke && ./script.sh
```


Αναπαράσταση εκτέλεσης αλγορίθμου

```
root@kali: /home/apipiliak/Desktop/adke
```

BSSID	SSID	dBm_Signal	Channel	Crypto
b8:be:f4:9c:a0:1b	devolo-094	-73	1	{WPA2/PSK}
78:96:82:3e:fc:84	COSMOTE-3EFC84	-75	1	{WPA2/PSK}
e0:0e:e4:7d:8c:66	COSMOTE	-73	3	{WPA2/PSK}
e0:0e:e4:7d:8c:69	COSMOTE WiFi Fon	-79	3	{OPN}
e4:26:86:41:08:c9	COSMOTE WiFi Fon	-67	6	{OPN}
3c:98:72:9f:65:d9	COSMOTE WiFi Fon	-73	8	{OPN}
3c:98:72:9f:65:d6	COSMOTE-176782	-77	8	{WPA2/PSK}
e4:26:86:12:3b:19	COSMOTE WiFi Fon	-63	9	{OPN}
3c:98:72:c1:58:d6	COSMOTE-166711	-75	5	{WPA2/PSK}
e4:26:86:12:3b:16	Merm wifi	-63	9	{WPA2/PSK}
3c:98:72:c1:58:d9	COSMOTE WiFi Fon	-75	11	{OPN}
b0:4e:26:b3:82:bb	VASAGGROU_EXT	-41	6	{WPA2/PSK, WPA/PSK}

Writing to file ...

Αναπαράσταση εκτέλεσης αλγορίθμου

```
root@kali: /home/apipilikas/Desktop/adke
```

```
File Actions Edit View Help
```

```
CH 6 ][ Elapsed: 6 s ][ 2022-02-27 16:44
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
E4:26:86:41:08:C6	-70	43	28	7 0	6	270	WPA2	CCMP	PSK	VASAGGROU

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
E4:26:86:41:08:C6	B2:4E:26:B3:82:BA	-36	0 - 1e	0	1		
E4:26:86:41:08:C6	B2:4E:26:8A:E4:BB	-40	0 - 1e	0	1		
E4:26:86:41:08:C6	B2:4E:26:07:C5:9B	-40	0 - 1e	0	1		
E4:26:86:41:08:C6	B2:4E:26:B3:82:BB	-40	0 - 1e	4	4		
E4:26:86:41:08:C6	18:01:F1:54:AD:71	-64	0 - 1	1948	9		
E4:26:86:41:08:C6	E0:CC:F8:78:C3:FC	-66	0 - 1e	0	2		

Αναπαράσταση εκτέλεσης αλγορίθμου

```
root@kali: /home/apipilikas/Desktop/adke
File Actions Edit View Help
night_measurements.txt
Writing to night_measurements.txt
Channel 1 0.06999999999999999
Channel 2 0
Channel 3 0.06999999999999999
Channel 4 0
Channel 5 0
Channel 6 0.5963636363636363
Channel 7 0
Channel 8 0.10318181818181818
Channel 9 0.16045454545454546
Channel 10 0
Channel 11 0
Channel 12 0
Channel 13 0
Channel 14 0
-----
Channel 1 0.0392
Channel 2 0.0168
Channel 3 0.0392
Channel 4 0.0084
Channel 5 0.07156363636363636
Channel 6 0.33396363636363635
Channel 7 0.08394545454545455
Channel 8 0.07703636363636364
Channel 9 0.10223636363636364
Channel 10 0.019254545454545455
Channel 11 0.0
Channel 12 0.0
Channel 13 0.0
Channel 14 0.0
[11, 12, 13, 14]
```

Αποτελέσματα αλγορίθμου αξιολόγησης καναλιού

Αποτελέσματα αλγορίθμου αξιολόγησης γειτονικών καναλιών και ιστορικού

Καταλληλότερα κανάλια

```
(root@kali) - [ /home/apipilikas/Desktop/adke ]
```

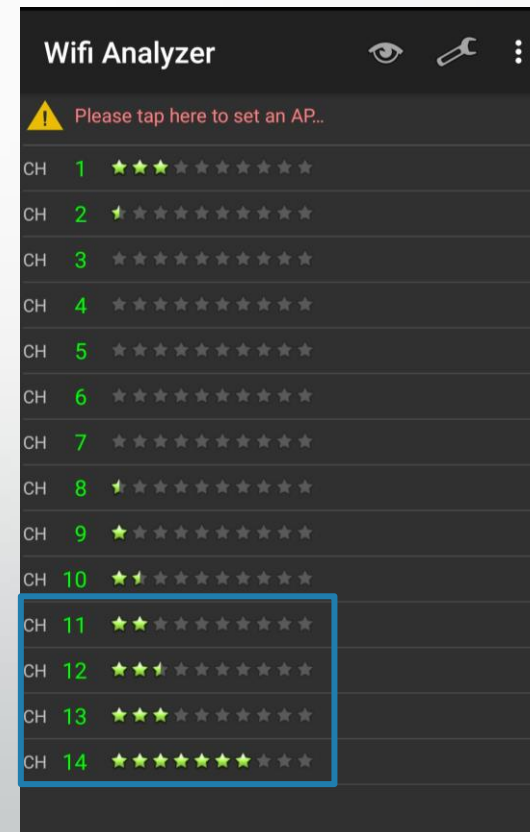
Σύγκριση με εφαρμογή Android αξιολόγησης καναλιών

Ο αλγόριθμος μας

```
File Actions Edit View Help
night_measurements.txt
Writing to night_measurements.txt
Channel 1 0.06999999999999999
Channel 2 0
Channel 3 0.06999999999999999
Channel 4 0
Channel 5 0
Channel 6 0.5963636363636363
Channel 7 0
Channel 8 0.10318181818181818
Channel 9 0.16045454545454546
Channel 10 0
Channel 11 0
Channel 12 0
Channel 13 0
Channel 14 0
-----
Channel 1 0.0392
Channel 2 0.0168
Channel 3 0.0392
Channel 4 0.0084
Channel 5 0.07156363636363636
Channel 6 0.33396363636363635
Channel 7 0.08394545454545455
Channel 8 0.07703636363636364
Channel 9 0.10223636363636364
Channel 10 0.019254545454545455
Channel 11 0.0
Channel 12 0.0
Channel 13 0.0
Channel 14 0.0
[11, 12, 13, 14]

(root@kali)-[/home/apipilikas/Desktop/adke]
```

Wifi Analyzer (Android)



Τι μάθαμε στην εργασία;

- Μελετήσαμε τα βασικά πεδία ενός beacon frame πακέτου
- Μάθαμε να χρησιμοποιούμε τις διεπαφές του Linux
- Μάθαμε να κρίνουμε τα κανάλια βάσει φόρτου
- Δοκιμάσαμε διάφορους αλγορίθμους και διάφορες αναλογίες



ΤΕΛΟΣ

Ευχαριστούμε για την προσοχή σας!