ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΜΣ – Κατεύθυνση:

"Ψηφιακές Επικοινωνίες & Δίκτυα"

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΕΛΕΤΗ, ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΕΙΡΑΜΑΤΙΚΗ ΧΡΗΣΗ ΕΝΟΣ ΣΥΣΤΗΜΑΤΟΣ ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗΣ ΚΑΙ ΕΛΕΓΧΟΥ ΠΡΟΣΒΑΣΗΣ ΧΡΗΣΤΩΝ ΣΕ ΕΝΑ ΑΣΥΡΜΑΤΟ ΔΙΚΤΥΟ (WLAN) ΠΟΥ ΒΑΣΙΖΕΤΑΙ ΣΤΟ ΠΡΩΤΟΚΟΛΛΟ RADIUS

Μαρέσκας Ευριπίδης

Α.Μ. ΜΕ/0542

ΑΘΗΝΑ 2008

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΜΣ – Κατεύθυνση:

"Ψηφιακές Επικοινωνίες & Δίκτυα"

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΕΛΕΤΗ, ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΠΕΙΡΑΜΑΤΙΚΗ ΧΡΗΣΗ ΕΝΟΣ ΣΥΣΤΗΜΑΤΟΣ ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗΣ ΚΑΙ ΕΛΕΓΧΟΥ ΠΡΟΣΒΑΣΗΣ ΧΡΗΣΤΩΝ ΣΕ ΕΝΑ ΑΣΥΡΜΑΤΟ ΔΙΚΤΥΟ (WLAN) ΠΟΥ ΒΑΣΙΖΕΤΑΙ ΣΤΟ ΠΡΩΤΟΚΟΛΛΟ RADIUS

Μαρέσκας Ευριπίδης

Α.Μ. ΜΕ/0542

ΕΠΙΒΛΕΨΗ: Λέκτορας Ξενάκης Χρήστος

ΑΘΗΝΑ 2008

Αφιερώνεται στους γονείς μου

Κωνσταντίνο και Μαρία

Πρόλογος

Καθώς τα δίκτυα εξαπλώνονται πέρα από το φυσικό χώρο των επιχειρήσεων η έννοια της ασφάλειας γίνεται πιο σημαντική και σύνθετη. Οι εταιρίες πρέπει να προστατέψουν τα δίκτυά και τους δικτυακούς τους πόρους από απομακρυσμένους χρήστες που μπαίνουν παράνομα στο σύστημα αποκτώντας πρόσβαση με κάποιο τρόπο. Η διακίνηση των δεδομένων μέσω τηλεπικοινωνιακών δικτύων δημιουργεί προβλήματα καθώς αυτά καθίστανται ευπρόσβλητα σε κακόβουλες ενέργειες. Επίσης η σύνδεση των ιδιωτικών δικτύων με το Internet ή η διασύνδεσή τους μέσω αυτού δίνει τη δυνατότητα επιθέσεων προς τους υπολογιστές των ιδιωτικών δικτύων.

Η μελέτη, εγκατάσταση και πειραματική χρήση ενός συστήματος αυθεντικοποίησης και ελέγχου πρόσβασης χρηστών σε ένα ασύρματο δίκτυο (WLAN) που βασίζεται στο πρωτόκολλο RADIUS, αποτελεί το θέμα της παρούσας Διπλωματικής Εργασίας

Στόχος και αντικείμενο αυτής, είναι η παρουσίαση του πρωτοκόλλου RADIUS καθώς και του προγράμματος ανοιχτού κώδικά FreeRADIUS και η λειτουργία αυτού σε ασύρματο περιβάλλον. Επίσης δίνονται μερικά σενάρια λειτουργίας σε μορφή εργαστηριακών ασκήσεων για την καλύτερη κατανόηση, καθώς και μία από τις μεθόδους για την δημιουργία WiFi Hotspot.

Η συγκεκριμένη Διπλωματική Εργασία φιλοδοξεί να συμβάλλει, όσο το δυνατόν καλύτερα, στην κάλυψη των παραπάνω στόχων και να κάνει όσο το δυνατόν πιο κατανοητή την τεχνολογία αυτή.

Κλείνοντας τον πρόλογο αυτό, θα ήθελα να εκφράσω τις ευχαριστίες μου στον καθηγητή μου, κύριο *Χρήστο Ξενάκη*, που με την πολύτιμη συμβολή του βοήθησε ουσιαστικά στην πραγμάτωση της παρούσας Διπλωματικής Εργασίας.

Ιούλιος 2008

Αθήνα

Πίνακας περιεχομένων

Λίστα εικόνων

Αριθμός ***Σελίδα***

Κεφάλαιο 1

Εισαγωγή

Καθώς τα δίκτυα εξαπλώνονται πέρα από το φυσικό χώρο των επιχειρήσεων η έννοια της ασφάλειας γίνεται πιο σημαντική και σύνθετη. Οι εταιρίες πρέπει να προστατέψουν τα δίκτυά και τους δικτυακούς τους πόρους από απομακρυσμένους χρήστες που μπαίνουν παράνομα στο σύστημα αποκτώντας πρόσβαση με κάποιο τρόπο. Τα συστήματα της Cisco χρησιμοποιούν μία στρατηγική που είναι γνωστή σαν Πιστοποίηση, Έγκριση και Παρακολούθηση (authentication, authorization, accounting-AAA) για να εκτελέσει τις λειτουργίες της πιστοποίησης της ταυτότητας του χρήστη, τη παροχή ή όχι πρόσβασης και την παρακολούθηση των κινήσεων των απομακρυσμένων χρηστών αντίστοιχα. Στα σημερινά δίκτυα χρησιμοποιούνται τα πρωτοκολλά TACACS+ (Terminal Access Controller Access Control System plus) και RADIUS (Remote Access Dial-In User Service) για τη παροχή AAA λύσεων. Η υποστήριξη των RADIUS και TACACS+ δίνει τη δυνατότητα στη Cisco να προτείνει μία πολύ ευέλικτη και αποδοτική AAA λύση.

1.1 Ανάλυση των Απαιτήσεων Ασφάλειας (AAA)

Authentication – Πιστοποίηση

Η Πιστοποίηση είναι η διαδικασία με την οποία καθορίζεται ποιος έχει πρόσβαση στο LAN. Απλές μέθοδοι έγκρισης χρησιμοποιούν μια βάση δεδομένων που αποτελείται από usernames και passwords στον server πρόσβασης. Πιο εξελιγμένα συστήματα χρησιμοποιούν μεθόδους όπως το TACACS και το Kerberos.

Ωστόσο, το ότι πιστοποιείται η ταυτότητα κάποιου χρήστη δε σημαίνει ότι αυτός έχει αποκτήσει πρόσβαση σε όλες τις υπηρεσίες του δικτύου είναι πιθανό να του ζητηθεί εκ νέου κάποιος κωδικός από κάποια συγκεκριμένη υπηρεσία UNIX, NetWare ή AppleShare. Ένας καλός NAS server υποστηρίζει μία πλειάδα επιλογών πιστοποίησης.

Authorization - Έγκριση

Η Έγκριση είναι η ικανότητα του περιορισμού των δικτυακών υπηρεσιών σε διαφορετικούς χρήστες βάση μιας δυναμικά εφαρμοζόμενης λίστας πρόσβασης (access list) που μερικές φορές αναφέρεται και ως "προφίλ χρήστη" και που βασίζεται στο δίδυμο username/password. Αυτό το χαρακτηριστικό είναι σημαντικό για δύο λόγους: βοηθάει στη μείωση της έκθεσης του εσωτερικού δικτύου στον έξω κόσμο και απλοποιεί τη μορφή του δικτύου για τον τελικό χρήστη που αγνοεί τις τεχνικές του λεπτομέρειες.

Το χαρακτηριστικό της έγκρισης επιτρέπει στους χρήστες να κινούνται. Κινούμενοι και προσωρινοί χρήστες (χρήστες με φορητά από ξενοδοχεία και τηλεργαζόμενοι με modems και ISDN συνδέσεις από το σπίτι) θέλουν να συνδεθούν στη πιο κοντινή τοπική σύνδεση διατηρώντας ωστόσο όλα τα προνόμιά των LAN τους.

Ο Διαχειριστής του δικτύου (Network Administrator) πρέπει να είναι σε θέση να περιορίζει τη πρόσβαση στο δίκτυο για όλα τα πρωτόκολλα και τις υπηρεσίες (Telnet, IP, IPX και AppleTalk) όσο οι χρήστες συνδέονται κάνοντας κλήση (dial-in) στο ίδιο σύνολο συσκευών modem. Η διαδικασία έγκρισης με τη χρήση λίστας πρόσβασης για κάθε χρήστη δεν περιορίζεται σε συγκεκριμένες διεπαφές **(**interfaces**)** αλλά ανατίθεται δυναμικά στη συγκεκριμένη πόρτα στην οποία συνδέεται ο χρήστης. Για παράδειγμα όταν ο χρήστης Α συνδέεται στη πόρτα 1, μπορεί να δει τα υπο-δίκτυα 1, 2, 3 και τις AppleTalk ζώνες bldg D, bldg E και bldg F. Όταν ο χρήστης 2 συνδέεται στη πόρτα 1, τότε το προφίλ του τον περιορίζει στο υπο-δίκτυο 1 και στη ζώνη bldg D.

Από τη στιγμή που το NAS υποστηρίζει πολύ περισσότερους απομακρυσμένους χρήστες από τις φυσικές γραμμές που έχει στη διάθεσή του κάθε χρήστης ή group, μπορεί να τηλεφωνήσει στο ίδιο περιστροφικό κέντρο και να πάρει πρόσβαση στο δίκτυο. Αυτή η λίστα πρόσβασης βασίζεται στο username και σαν τέτοια κάθε NAS μπορεί να υποστηρίξει χιλιάδες χρήστες στη βάση δεδομένων που έχει για τα usernames και passwords.

Accounting - Παρακολούθηση

Η παρακολούθηση είναι το τρίτο κύριο συστατικό ενός ασφαλούς συστήματος. Οι διαχειριστές του συστήματος μπορεί από το να θέλουν να χρεώσουν τους πελάτες τους για την ώρα που

παρέμειναν συνδεδεμένοι στο δίκτυο μέχρι να παρακολουθήσουν ύποπτες προσπάθειες σύνδεσης στο δίκτυο.

1.2 Το Πρωτόκολλο RADIUS

Το Remote Authentication Dial In User Service (RADIUS) είναι ένα πρωτόκολλο AAA (authentication, authorization and accounting protocol) για εφαρμογές όπως πρόσβαση στο δίκτυο ή φορητότητα IP (IP mobility). Είναι σχεδιασμένο να λειτουργεί και σε περιπτώσεις τοπικού δικτύου και περιαγωγής.

Αναπτύχθηκε από την Livingston Enterprises ως ένας server πρόσβασης, πιστοποίησης και παρακολούθησης. Από τότε έχει υλοποιηθεί από διάφορες άλλες εταιρίες και έχει κερδίσει ευρεία υποστήριξη ανάμεσα ακόμα και στους περιχείς υπηρεσιών (ISPs).

Είναι βασισμένο στο μοντέλο client/server όπου οι servers πρόσβασης (NAS-Network Access Servers) λειτουργούν σαν clients του RADIUS. Ο client είναι υπεύθυνος για την προώθηση της πληροφορίας του χρήστη στον αρμόδιο RADIUS server και την εκτέλεση των εντολών που θα του σταλούν πίσω από το server.

Ο RADIUS server ή daemon παρέχει υπηρεσίες πιστοποίησης και παρακολούθησης σε έναν ή περισσότερους RADIUS clients δηλαδή συσκευές NAS. Οι RADIUS servers είναι υπεύθυνοι για το να λαμβάνουν τις αιτήσεις σύνδεσης των χρηστών, να τους πιστοποιούν και τέλος να επιστρέφουν όλη τη πληροφορία με τις απαιτούμενες ρυθμίσεις για τους clients ώστε να δοθούν οι αιτούμενες υπηρεσίες στους χρήστες. Ο RADIUS server πρόσβασης είναι συνήθως ένας αφιερωμένος σταθμός εργασίας συνδεδεμένος με το δίκτυο.

NAS

Όπως προαναφέρθηκε, ο NAS είναι ο RADIUS client και όχι ο τελικός χρήστης ή η συσκευή η οποία πιστοποιεί στο δίκτυο μέσω του NAS. Κατά τη διάρκεια των διαδικασιών αυθεντικοποίησης, ο RADIUS client είναι υπεύθυνος στο να προωθεί πληροφορίες του χρήστη με τη μορφή αιτημάτων στο RADIUS server και να περιμένει απάντηση από αυτόν. Ανάλογα με την πολιτική, ο NAS μπορεί να

χρειάζεται μονάχα μια επιτυχημένη αυθεντικοποίηση ή περεταίρω οδηγίες αυθεντικοποίησης από το server για να ανοίξει τις θύρες κίνησής του (traffic ports) στη κίνηση του πελάτη (client's traffic). Ο NAS μπορεί αν χρειαστεί να καθιερώσει ασφαλή κανάλια επικοινωνίας με το πελάτη πριν ξεκινήσει οποιαδήποτε επικοινωνία με τον τελικό χρήστη. Επί προσθέτως, όταν απαιτείται παρακολούθηση (accounting), ο NAS είναι επίσης υπεύθυνος για την περισυλλογή resource usage data και την αναφορά πίσω στο server.

Ο RADIUS server, από την άλλη, είναι υπεύθυνος για την επεξεργασία αιτημάτων, πιστοποίηση των χρηστών και να επιστρέφει τις απαραίτητες πληροφορίες για τις ρυθμίσεις του πελάτη ώστε να προσφέρει την υπηρεσία στο χρήστη.

Οι προδιαγραφές του RADIUS αποτελούνται από αρκετά RFCs.Η βασική προδιαγραφή του RADIUS αναθεωρήθηκε αρκετές φορές (RFCs 2058, 2138) και τώρα περιλαμβάνεται στο RFC 2865 (RADIUS2865) και περιγράφει την διαδικασία πιστοποίησης ενός χρήστη σε έναν server χρησιμοποιώντας PAP ή CHAP,αλλά δεν περιγράφει την υποστήριξη για παρακολούθηση, οι οποίες έγιναν στάνταρτ σε ξεχωριστό RFC (RADACC2866). Αργότερα έγινε επέκταση της προδιαγραφής ώστε να υποστηρίζονται και άλλες λειτουργίες όπως πιστοποίηση με χρήση EAP. Πολλές από τις λειτουργίες περιγράφονται σε περεταίρω προδιαγραφές RFC. Ξεκινάμε την περιγραφή του πρωτοκόλλου όπως περιγράφεται στο βασικό RFC και έπειτα προχωράμε σε λεπτομέρειες όπως αναφέρονται στις επεκτάσεις του.

Πολλές υπηρεσίες δικτύου (συμπεριλαμβανομένων εταιρικών δικτύων και δημοσίων ISP με χρήση modem, DSL, ή ασύρματες τεχνολογίες 802.11.X) απαιτούν την παρουσίαση διαπιστευτηρίων (όπως ένα όνομα χρήστη και κωδικό ή πιστοποιητικό ασφαλείας) ώστε να γίνει σύνδεση στο δίκτυο. Πριν η πρόσβαση στο δίκτυο γίνει εφικτή, η πληροφορία αυτή προωθείται σε μία συσκευή Network Access Server (NAS) μέσω του πρωτοκόλλου σύνδεσης δεδομένων (Data Link layer), για παράδειγμα το Point-to-Point πρωτόκολλο (PPP) στην περίπτωση πολλών dialup ή DSL παρόχων), έπειτα σε RADIUS server μέσω του πρωτοκόλλου RADIUS. Ο RADIUS server ελέγχει ότι η πληροφορία είναι σωστή με τη χρήση συστημάτων αυθεντικοποίησης όπως PAP, CHAP ή EAP. Εάν γίνει αποδεκτή, ο server υποδεικνύει στο NAS υπάρχει εξουσιοδότηση για την πρόσβαση στο δίκτυο. Επίσης το RADIUS επιτρέπει στον server πιστοποίησης να προμηθεύει το NAS με επιπλέον παραμέτρους, όπως

- Τη συγκεκριμένη διεύθυνση IP που θα ανατεθεί στο χρήστη
- Το σύνολο διευθύνσεων απ' όπου η IP του χρήστη θα επιλεχθεί
- Τη μέγιστη διάρκεια που μπορεί ο χρήστης να παραμείνει συνδεδεμένος
- Μια λίστα πρόσβασης, ουρά προτεραιότητας ή απαγορεύσεις στις παραμέτρους πρόσβασης L2TP του χρήστη

Το πρωτόκολλο RADIUS δεν μεταδίδει κωδικούς πρόσβασης σε απλό κείμενο μεταξύ του NAS και του RADIUS server (ούτε καν με το πρωτόκολλο PAP), αλλά κωδικοποιημένα, με χρήση μιας πολύπλοκης λειτουργίας, που περιλαμβάνει σύνοψη MD5 **(**Συνάρτηση Κατακερματισμού MD5) και μέθοδο κοινού μυστικού, όπως περιγράφεται στα RFC's. Το πρωτόκολλο RADIUS είναι επίσης γνωστό για χρήση του σε περιπτώσεις παρακολούθησης. Ο NAS μπορεί να χρησιμοποιήσει τα πακέτα παρακολούθησης RADIUS για να γνωστοποιεί στο RADIUS server γεγονότα όπως τα ακόλουθα.

- Την έναρξη συνόδου του χρήστη
- Το τέλος της συνόδου του χρήστη
- Το σύνολο των πακέτων που μεταφέρθηκαν κατά τη διάρκεια της συνόδου
- Ποσότητα δεδομένων που μεταφέρθηκαν κατά τη διάρκεια της συνόδου .
- Αιτία και τερματισμό συνόδου

Πρωτεύων σκοπός των δεδομένων αυτών είναι η χρέωση του χρήστη σύμφωνα με αυτά, καθώς και για στατιστική χρήση για γενική παρακολούθηση του δικτύου. Επίσης το RADIUS χρησιμοποιείται ευρέως από παρόχους υπηρεσιών VoIP, ώστε να περνούν διαπιστευτήρια εισόδου από ένα τερματικό SIP (όπως ένα τηλέφωνο διαδικτύου) σε έναν SIP Registrar με τη χρήση αυθεντικοποίησης σύνοψης (digest authentication**)**, και έπειτα σε έναν RADIUS server με τη χρήση του RADIUS πρωτοκόλλου. Μερικές φορές χρησιμοποιείται για τη συλλογή πληροφοριών καταγραφής κλήσεων (call detail records ή εν συντομία CDRs**)** που αργότερα μπορούν να χρησιμοποιηθούν, για παράδειγμα, για τη χρέωση πελατών για διεθνής κλήσεις .

Το πρωτόκολλο RADIUS αρχικά καθορίστηκε από ένα RFI από το Merit Network το 1991 για να ελέγχει την dial-in πρόσβαση στο NSFnet. Η Livingston Enterprises ανταποκρίθηκε στο RFI με μια περιγραφή ενός RADIUS server. Το Merit Network επιβράβευσε το συμβόλαιο με την Livingston Enterprises που τους παρέδωσε την σειρά PortMaster των Network Access Servers και τον αρχικό RADIUS server στο Merit. Το 1997 το πρωτόκολλο RADIUS δημοσιεύτηκε ως RFC 2058 και RFC 2059 (οι παρούσες εκδόσεις είναι RFC 2865 και RFC 2866). Στην εποχή μας υπάρχουν πολλές υλοποιήσεις RADIUS server, εμπορικές και ανοικτού κώδικα. Οι δυνατότητες αυτών ποικίλουν, αλλά οι περισσότερες μπορούν να αναζητούν τους χρήστες σε αρχεία κειμένου, LDAP servers,διάφορες βάσεις δεδομένων, κλπ. Οι εγγραφές παρακολούθησης μπορούν να γραφούν σε αρχεία κειμένου, βάσεις δεδομένων, να προωθηθούν σε εξωτερικούς servers, κλπ.. Το πρωτόκολλο SNMP χρησιμοποιείται συχνά για απομακρυσμένη διαχείριση. Οι RADIUS proxy servers χρησιμοποιούνται και κεντρική διαχείριση και μπορούν να αναδημιουργήσουν πακέτα RADIUS άμεσα (on the fly) για λόγους ασφαλείας ή για μετατροπή μεταξύ διαφορετικών "διαλέκτων" επικοινωνίας. Επίσης είναι ένα κοινότυπο πρωτόκολλο πιστοποίησης που αξιοποιείται από το στάνταρντ ασφαλείας 802.1X (που συχνά χρησιμοποιείται από τα ασύρματα δίκτυα). Παρ' όλο που δεν προοριζόταν αρχικά να είναι μέθοδος ασύρματης πιστοποίησης ασφαλείας, βελτιώνει τη χρήση της κρυπτογράφησης WEP, με ενσωμάτωση άλλων μεθόδων ασφαλείας όπως τα EAP-PEAP.Οι θύρες που χρησιμοποίει το RADIUS είναι οι UDP 1812 ή 1645 για πιστοποίηση και 1813 ή 1646 για παρακολούθηση. Η επίσημη ανάθεση θύρας από την IETF είναι οι θύρες 1812 και 1813.

Το πρωτόκολλο RADIUS ορίζεται στα ακόλουθα RFC:

RFC 2865 Remote Authentication Dial In User Service (RADIUS)

RFC 2866 RADIUS Accounting

Άλλα σχετικά RFCs είναι:

RFC 2548 Microsoft Vendor-specific RADIUS Attributes

RFC 2607 Proxy Chaining and Policy Implementation in Roaming

RFC 2618 RADIUS Authentication Client MIB

RFC 4668 RADIUS Authentication Client MIB for IPv6 (Obsoletes: RFC 2618)

RFC 2619 RADIUS Authentication Server MIB

RFC 4669 RADIUS Authentication Server MIB for IPv6 (Obsoletes: RFC 2619)

RFC 2620 RADIUS Accounting Client MIB

RFC 4670 RADIUS Accounting Client MIB for IPv6 (Obsoletes: RFC 2620)

RFC 2621 RADIUS Accounting Server MIB

RFC 4671 RADIUS Accounting Server MIB for IPv6 (Obsoletes: RFC 2621)

RFC 2809 Implementation of L2TP Compulsory Tunneling via RADIUS

RFC 2867 RADIUS Accounting Modifications for Tunnel Protocol Support

RFC 2868 RADIUS Attributes for Tunnel Protocol Support

RFC 2869 RADIUS Extensions

RFC 2882 Network Access Servers Requirements: Extended RADIUS Practices

RFC 3162 RADIUS and IPv6

RFC 3575 IANA Considerations for RADIUS

RFC 3576 Dynamic Authorization Extensions to RADIUS

RFC 3579 RADIUS Support for EAP (Updates: RFC 2869)

RFC 3580 IEEE 802.1X RADIUS Usage Guidelines

RFC 4014 RADIUS Attributes Suboption for the DHCP Relay Agent Information Option

RFC 4372 Chargeable User Identity

RFC 4590 RADIUS Extension for Digest Authentication (new revision pending)

RFC 4675 RADIUS Attributes for Virtual LAN and Priority Support

RFC 4679 DSL Forum Vendor-Specific RADIUS Attributes

RFC 4818 RADIUS Delegated-IPv6-Prefix Attribute

RFC 4849 RADIUS Filter Rule Attribute

Κεφάλαιο 2 Αρχιτεκτονική

2.1 Λειτουργία Πρωτοκόλλου

Η επικοινωνία μεταξύ ενός NAS και ενός RADIUS server βασίζεται στο User Datagram Protocol (UDP). Το σχήμα 1 δείχνει τη μορφή ενός πακέτου RADIUS.

Οι δημιουργοί του RADIUS επέλεξαν το UDP ως το πρωτόκολλο μεταφοράς για τεχνικούς λόγους. Γενικά το RADIUS θεωρείται μία υπηρεσία άνευ συνδέσεως(connectionless). Θέματα που σχετίζονται με τη διαθεσιμότητα του server, την επανεκπομπή και τα timeouts διαχειρίζονται από διάφορες συσκευές του RADIUS και όχι από το πρωτόκολλο μεταφοράς.

Το σύνολο των μηνυμάτων RADIUS που απαιτείται είναι αρκετά απλό και αποτελείται από οκτώ μηνύματα, από τα οποία τα πρώτα τέσσερα προσδιορίζονται στη βασική προδιαγραφή του πρωτοκόλλου. Συνοπτικά η αλληλουχία των μηνυμάτων είναι η ακόλουθη:

- **Access Request:** Αυτό το μήνυμα δημιουργείται από το NAS (RADIUS client) προς το server για να προωθήσει το αίτημα από ή εκ μέρους ενός χρήστη. (NAS-->AS).
- **Access Challenge:** Αυτό το μήνυμα αποστέλλεται από το RADIUS server στον RADIUS client (NAS) και γενικά χρησιμοποιείται για να ρωτά το χρήστη ή το NAS κάτι ή να πραγματοποιήσει κάποιο είδος διαμεσολάβησης.
- **Access Accept:** Αυτό το μήνυμα αποστέλλεται από το RADIUS server στο NAS για να υποδείξει ότι το αίτημα ολοκληρώθηκε επιτυχώς (και τυπικά επιτρέπεται η πρόσβαση).
- **Access Reject:** Αυτό το αίτημα αποστέλλεται από το server για να υποδείξει την απόρριψη του αιτήματος.
- **Accounting request:** Αυτό το μήνυμα αποστέλλεται από το πελάτη προς τον server παρακολούθησης (accounting server**)** για να μεταβιβάσει τις

πληροφορίες παρακολούθησης (accounting information) που συσχετίζονται με την υπηρεσία που παρέχεται στο χρήστη.

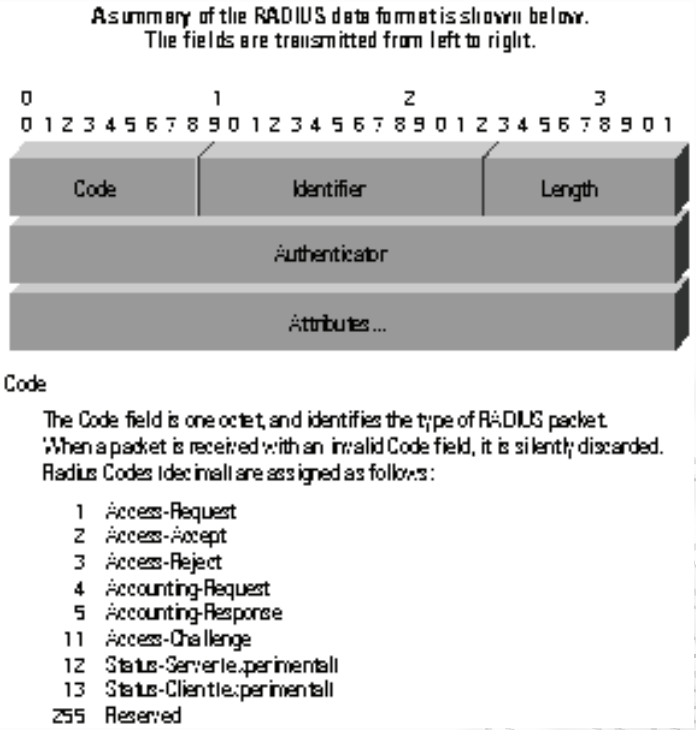
- **Accounting response:** Αυτό το μήνυμα αποστέλλεται από το server στο πελάτη για να γνωστοποιήσει ότι η πληροφορία παρακολούθησης που έχει σταλεί από το πελάτη έχει ληφθεί και υποδεικνύει το αποτέλεσμα από την πραγματοποιηθείσα διεργασία παρακολούθησης (accounting function) από το server.
- **Status-Server και Status-Client:** Αυτά τα δύο μηνύματα έχουν πειραματική χρήση.

Η μορφή των πακέτων RADIUS είναι επίσης πολύ απλή και αποτελείται από επικεφαλίδα, που περιλαμβάνει κωδικό,ID, και ένα πεδίο πιστοποίησης που ακολουθείται από ένα φορτίο, που είναι μια ακολουθία από μηδενικά η περισσότερα χαρακτηριστικά όπως φαίνεται στον ακόλουθο πίνακα.

Μορφή πακέτων RADIUS		
Όνομα πεδίου	**Όνομα υποπεδίου**	**Description**
Επικεφαλίδα (Header)	Κώδικας (Code)	Προσδιορίζει τον τύπο του πακέτου RADIUS (αίτημα πρόσβασης, Απάντηση, κλπ)
	ID	Για ταίριασμα αιτημάτων και απαντήσεων
	Μήκος (Length)	Μήκος 2 οκτάδων αναφέροντας το μήκος ολόκληρου του μηνύματος
	Authenticator (αυθενικοποιητής)	Τιμή 16 οκτάδων που υπολογίστηκε όπως περιγράφηκε προηγουμένως.
Χαρακτηριστικό 1 …		Πρώτο χαρακτηριστικό στο πακέτο
Ν-οστό Χαρακτηριστικό		Ν τελευταίο χαρακτηριστικό στο πακέτο

Σχήμα 1 Μορφή πακέτων RADIUS

Στο παρακάτω σχήμα φαίνονται και γραφικά τα ανωτέρω χαρακτηριστικά.

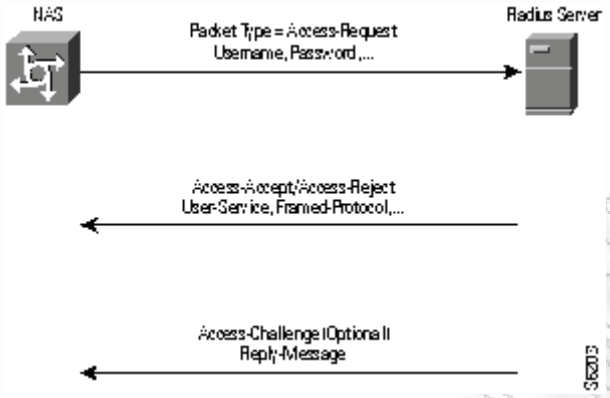


Εικόνα 1:Μορφή πακέτων RADIUS από RFC 2058

Τυπικά μία αίτηση για login αποτελείται από μία αίτηση (Access Request) από το NAS server στον RADIUS server και μια απάντηση, θετική ή αρνητική, του τελευταίου (Access-Accept ή Access-Reject). Το πακέτο αίτησης που στέλνει ο NAS server περιέχει το username, το κρυπτογραφημένο password, την IP διεύθυνση του NAS server και τη πόρτα. Η μορφή της αίτησης παρέχει επιπλέον πληροφορίες για τον τύπο της σύνδεσης την οποία ο χρήστης θέλει να ξεκινήσει. Για παράδειγμα εάν η αίτηση παρουσιάζεται σε τύπο χαρακτήρων τότε το "Service-Type = Exec-User" αλλά εάν παρουσιάζεται σε τύπο PPP πακέτου τότε το "Service-Type = Framed User" και "Framed-Type = PPP".

Όταν ο RADIUS server λαμβάνει μια αίτηση από κάποιον NAS, ψάχνει σε μια βάση δεδομένων για το όνομα χρήστη (username) που υπάρχει στην αίτηση. Εάν το username δεν υπάρχει στη βάση δεδομένων τότε είτε ένα τυπικό προφίλ φορτώνεται και ο RADIUS server αποστέλλει μήνυμα αποδοχής (Access-Accept) είτε αποστέλλει μήνυμα απόρριψης (Access-Reject) το οποίο μπορεί να συνοδεύεται και από κάποιο επεξηγηματικό μήνυμα του λόγου απόρριψης.

Στην περίπτωση που το όνομα χρήστη βρεθεί και ο κωδικός πρόσβασης είναι σωστό ο RADIUS server επιστρέφει μία απάντηση Access-Accept, η οποία περιλαμβάνει μια λίστα των χαρακτηριστικών των ρυθμίσεων που πρέπει να χρησιμοποιηθούν από τη μεριά του NAS για τη σύνδεση. Τυπικές παράμετροι περιλαμβάνουν το τύπο της υπηρεσίας (shell ή framed), το τύπο του πρωτοκόλλου, την IP διεύθυνση που θα δοθεί στο χρήστη (στατική ή δυναμική), την access list που πρέπει να εφαρμοστεί ή τη στατική διεύθυνση που πρέπει να εγκατασταθεί στον πίνακα δρομολογίων του NAS. Η εικόνα 2 δείχνει τη διαδικασία της σύνδεσης και πιστοποίησης RADIUS.



Εικόνα 2: Διαδικασία σύνδεσης και πιστοποίησης RADIUS

2.2 Χαρακτηριστικά Διαδικασίας Πιστοποίησης και Έγκρισης

Η πιστοποίηση είναι η πιο απαιτητική πλευρά της ασφάλισης απομακρυσμένων χρηστών λόγω της δυσκολίας που σχετίζεται με τη σίγουρη αναγνώριση του χρήστη. Για τη διασφάλιση της ταυτότητας ενός απομακρυσμένου χρήστη το πρωτόκολλο RADIUS υποστηρίζει πολλές μεθόδους πιστοποίησης περιλαμβανομένων των Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP) και token cards. Προς το παρόν όλες οι εκδόσεις του RADIUS απαιτούν να τρέχει ένας server για τα token cards επιπρόσθετα του RADIUS server. Όταν βγει στην αγορά η έκδοση υποστήριξης του RADIUS, CiscoSecure, θα περιέχει OEM υποστήριξη για CryptoCard token κάρτες και έτσι δεν θα είναι απαραίτητος επιπλέον server για τα token cards.

2.3 Ενεργοποίηση της Πιστοποίησης, Έγκρισης και παρακολούθησης RADIUS

Για κάθε τύπο σύνδεσης που χρειάζεται πιστοποίηση και έγκριση, πρέπει να εισαχθεί μια γραμμή εντολών. Αυτή η γραμμή είναι η λίστα που χρησιμοποιείται για σύνδεση μέσω του RADIUS εκτός αν υπάρχει κάποια άλλη λίστα που έχει ρυθμιστεί. Η παρακολούθηση μπορεί να χρησιμοποιηθεί ανεξάρτητα από τις άλλες διαδικασίες και επιτρέπει την αποστολή δεδομένων στην αρχή και στο τέλος των συνδέσεων καταδεικνύοντας τη ποσότητα των πόρων που χρησιμοποιήθηκαν κατά τη σύνδεση. Ένας ISP θα μπορούσε να χρησιμοποιήσει το RADIUS για να καλύψει ειδικές απαιτήσεις ασφάλειας και χρέωσης.

Κεφάλαιο 3 RADIUS και ασφάλεια

Το RADIUS, όπως έχει αναφερθεί και πιο πριν, είναι ένα πρωτόκολλο που από την αρχή έχει σχεδιαστεί να παρέχει ασφάλεια ώστε μονάχα πιστοποιημένοι χρήστες να μπορούν να απολαμβάνουν τους πόρους που προσφέρονται σε ένα μεγάλο γκρουπ ατόμων. Δυστυχώς έχει αρκετά προβλήματα ασφάλειας, που ορισμένα είναι πάρα πολύ σημαντικά.

Η πιο σημαντική αδυναμία ασφαλείας στηρίζεται στην ευρεία χρήση του πρωτοκόλλου. Υποστηρίζεται από έναν αρκετά μεγάλο αριθμό από εξοπλισμό δικτύου και χρησιμοποιείται από σχεδόν όλους τους παρόχους Ίντερνετ και εταιρικών εφαρμογών dial-up.Παρ' όλα αυτά, αυτή η δημοτικότητα είναι δίκοπο μαχαίρι. Οι αδυναμίες ασφαλείας στο πυρήνα του πρωτοκόλλου αφήνουν εκτεθειμένα χιλιάδες συστήματα. Επίσης, μεγάλες αλλαγές στη δομή του πρωτοκόλλου δεν μπορούν να γίνουν, λόγω του ρίσκου ασυμβατότητας σε χιλιάδες συστήματα που τρέχουν RADIUS.

Στη συνέχεια θα γίνει μία αναφορά των ευπαθειών αυτών, και μερικές μέθοδοι προστασίας από τις ατέλειες του πρωτοκόλλου

3.1 Ευπάθειες πρωτοκόλλου

Έχει ανακαλυφθεί ότι το RADIUS έχει κάποια πηγαία ελαττώματα που μπορεί να επιτρέψουν σε έναν επιτιθέμενο να διακυβεύσει την ακεραιότητα μιας συναλλαγής. Αρχικά, ο μηχανισμός ασφαλείας User-Password είναι εγγενώς αρκετά ανασφαλής, μη εφαρμόζοντας σωστά τις κρυπτογραφικές τεχνικές. Γενικά η ιδέα της απάντησης του αυθεντικοποιητή (response authenticator**)** στο πακέτο RADIUS είναι πραγματικά καλή, αλλά η υλοποίηση αυτού στο πρωτόκολλο δεν είναι καλά σχεδιασμένη. Το πακέτο Access-Request δεν πιστοποιείται —τουλάχιστον όπως ορίζεται από το πρωτόκολλο—από κανένα μέρος στη συναλλαγή. Επίσης η τυχαιότητα της γέννησης πελατών από αυθεντικοποιητές αιτήματος (request authenticators) δεν είναι αρκετά τυχαία. Και τέλος, το διαμοιραζόμενο μυστικό είναι μια απλοϊκή μέθοδος για ασφάλεια των συναλλαγών client-server.

3.2 MD5 και το Διαμοιραζόμενο Μυστικό (Shared Secret)

Η μέθοδος διαμοιραζόμενου μυστικού είναι ευάλωτη λόγω της ασθενούς σύνοψης MD5 το οποίο κρύβει την απάντηση του αυθεντικοποιητή. Ένας εισβολέας θα μπορούσε εύκολα αν επιτεθεί στο διαμοιραζόμενο μυστικό με sniffing ενός έγκυρου πακέτου Access-Request και της αντίστοιχης απάντησης του. Μπορεί εύκολα να πάρει το διαμοιραζόμενο μυστικό με "προ-υπολογισμό" του υπολογισμού MD5 από το κώδικα, το ID, το μήκος, το αίτημα του authenticator, και μέρους των χαρακτηριστικών από τα πακέτα και έπειτα επαναϋπολογίζει τη σύνοψη για κάθε εικασία που κάνει.

3.3 Το Πακέτο Access-Request

Εξ ορισμού δεν υπάρχει καμία επιβεβαίωση ούτε πιστοποίηση του πακέτου RADIUS Access-Request, ως μέρος της προδιαγραφής RFC. Ο RADIUS server θα πραγματοποιήσει έναν έλεγχο για να εξασφαλίσει ότι το μήνυμα προέρχεται από μία διεύθυνση IP που είναι στη λίστα ως ένας από τους πελάτες του, όμως στις μέρες μας, πλαστές διευθύνσεις IP (spoofed IP addresses) εύκολα βρίσκονται και χρησιμοποιούνται. Αυτό είναι ένα πολύ σοβαρό μειονέκτημα του σχεδιασμού του πρωτοκόλλου RADIUS.

Έως τώρα, η μόνη εφαρμόσιμη λύση είναι ναι απαιτείται η παρουσία του χαρακτηριστικού Message-Authenticator σε όλα τα μηνύματα Access-Request. Εν συντομία, ο Message-Authenticator είναι μια σύνοψη MD5 ολόκληρου του μηνύματος Access-Request, χρησιμοποιώντας το διαμοιραζόμενο μυστικό του πελάτη ως κλειδί. Όταν ένας RADIUS server είναι ρυθμισμένος να δέχεται μονάχα μηνύματα Access-Request με ένα έγκυρο χαρακτηριστικό Message-Authenticator παρών, πρέπει "ήσυχα" να απορρίψει αυτά τα πακέτα με μη έγκυρα ή χωρίς καθόλου χαρακτηριστικά (attributes). Για περισσότερες πληροφορίες για το χαρακτηριστικό του Message-Authenticator μπορούν να βρεθούν στο RFC 2869.

Εάν για κάποιο λόγο η υλοποίηση αποτρέπει τη χρήση του χαρακτηριστικού Message-Authenticator, θα πρέπει τουλάχιστον να υλοποιηθεί κάποιο είδος αποκλεισμού του λογαριασμού

(account-lockout), που θα απενεργοποιεί πιστοποιήσεις ύστερα από έναν ορισμένο αριθμό προσπαθειών πιστοποίησης μέσα σε ένα καθορισμένο χρονικό διάστημα.

3.4 Το σύστημα αλγορίθμου κρυπτογράφισης ροής User-Password

Υπό πολύ γενική έννοια, ο τρόπος κατά τον οποίο το χαρακτηριστικό User-Password χειρίζεται, είναι γνωστό ως αλγόριθμος κρυπτογράφησης ροής (stream cipher). Το stream cipher είναι μία μέθοδος κρυπτογράφησης που λειτουργεί με συνεχής ροές εισόδου, που είναι συνήθως μία ροή από bits απλού κειμένου και όχι φιξαριστά μπλοκ. Το ακριβώς το αντίθετο του είναι ένα block cipher, η οποία είναι μία μέθοδος κρυπτογράφησης που επεξεργάζεται την είσοδό της σε μπλοκ προκαθορισμένου μεγέθους, που είναι συνήθως 64 ή 128 bits μήκους. Ένα stream cipher δημιουργεί ένα keystream, και αυτό χρησιμοποιείται στην κρυπτογράφηση. Όταν συνδυαστεί αυτό το keystream με το stream εισόδου plain-text με τη χρήση της πράξης XOR, τα περιεχόμενα του stream είναι κρυπτογραφημένα. Η δημιουργία του keystream μπορεί να είναι ανεξάρτητη από το plain text και το ciphertext ή μπορεί να εξαρτάται στα δεδομένα και την κρυπτογράφησή τους, που σε αυτή την περίπτωση το stream cipher καλείται αυτό-συγχρονισμένο.

Στο σύστημα User-Password scheme, τα πρώτα 16 οκταδικά ψηφία δρουν σαν συγχρονισμένο stream cipher, μιας και η είσοδος plain text είναι ανεξάρτητη του keystream. Παρ' όλα αυτά, μετά τα πρώτα 16 οκταδικά ψηφία, το keystream ολοκληρώνει την αμέσως προηγούμενη είσοδο απλού κειμένου και γίνεται αυτοσυγχρονιζόμενο. Ενώ αυτό μπορεί να δείχνει πολύ τεχνικό, ή ασφάλεια αυτού του cipher είναι αμφισβητήσιμη: η προδιαγραφή του πρωτοκόλλου RADIUS δεν κάνει ξεκάθαρες ποιες είναι οι απαιτήσεις για αυτό το cipher. Οι συνόψεις MD5 τείνουν να είναι κρυπτογραφικές, όχι stream ciphers. Μπορεί να υπάρξει πρόβλημα ασφαλείας σε αυτή την πιθανά κακή χρήση.

Δυστυχώς ο μόνος τρόπος (τουλάχιστον με τη χρήση ενός Internet standard) για την περεταίρω ασφάλεια των χαρακτηριστικών και του μηνύματος ενός πακέτου RADIUS είναι με την χρήση του πρωτοκόλλου IPsec με επεκτάσεις ενθυλακωμένου φορτίου ασφάλειας (encapsulated security payload - ESP) και έναν αλγόριθμο όπως ο 3DES. Το RFC 3162 περιγράφει αυτή την διαδικασία με περισσότερη λεπτομέρεια.

3.5 Το Διαμοιραζόμενο Μυστικό User-Password

Από τη στιγμή που το χαρακτηριστικό User-Password προστατεύεται από ένα stream cipher, όπως έχει περιγραφεί νωρίτερα, είναι σίγουρα πιθανή η απόσπαση πληροφοριών για το διαμοιραζόμενο μυστικό από εισβολείς εάν μπορούν να υποκλέψουν κίνηση δικτύου και προσπαθήσουν να πιστοποιηθούν σε ένα RADIUS server. Για παράδειγμα, ένας εισβολέας θα μπορούσε να προσπαθήσει να πιστοποιηθεί χρησιμοποιώντας έναν κωδικό πρόσβασης γνωστό σε αυτόν. Έπειτα μπορεί να λάβει και να καταγράψει ένα πακέτο Access-Request και να χρησιμοποιήσει μια σύνοψη σε ένα συνδυασμό του προστατευμένου μέρους του User-Password και του κωδικού πρόσβασης μου αρχικά χρησιμοποιήθηκε. Μόλις αυτός ο υπολογισμός έχει ολοκληρωθεί, έχει το αποτέλεσμα της πράξης MD5 (διαμοιραζόμενο μυστικό + αυθεντικοποιητή αιτήματος). Γνωρίζει ήδη τον αυθεντικοποιητή αιτήματος από το αρχικό του αίτημα, επομένως με χρήση brute-force attack στο διαμοιραζόμενο μυστικό μπορεί να το καθορίσει εκτός σύνδεσης (offline).

3.6 Το χαρακτηριστικό User-Password και επιθέσεις κωδικού πρόσβασης

Ένας εισβολέας μπορεί να παρακάμψει οποιοδήποτε περιορισμό πιστοποίησης που έχει εγκατασταθεί από το διαχειριστή του RADIUS server λόγω της χρήσης του stream cipher που χρησιμοποιείται για τη προστασία του χαρακτηριστικού User-Password. Η μέθοδος της επίθεσης είναι η ακόλουθη:

1. Ο εισβολέας πρώτα προσπαθεί να πιστοποιηθεί σε έναν RADIUS server με τη χρήση ενός γνωστού και σωστού username και ενός γνωστού, αλλά πολύ πιθανόν μη σωστού, password.

2. Έπειτα παίρνει το πακέτο Access-Request που προκύπτει και μαθαίνει το αποτέλεσμα MD5 του υνδυασμού αιτήματος πιστοποίησης και διαμοιραζόμενο μυστικό, όπως περιγράφηκε προηγουμένως.
3. Έπειτα μπορεί να χρησιμοποιήσει επίθεση password brute-force αλλάζοντας τα passwords στο πακέτο και χρησιμοποιώντας το ίδιο αίτημα πιστοποίησης και διαμοιραζόμενο μυστικό.

Αυτό θα λειτουργήσει μονό ένα το password είναι λιγότερο ή ίσο με 16 χαρακτήρες, μιας και το User-Password cipher γίνεται αυτοσυγχρονιζόμενο στον $17^{ο}$ χαρακτήρα περιλαμβάνοντας προηγούμενο ciphertext στην κρυπτογράφηση.

3.7 Επιθέσεις με τη χρήση του Αυθεντικοποιητή αιτήματος

Υπάρχουν αρκετές πιθανές μέθοδοι επίθεσης με τη χρήση του μέρους του αιτήματος του authenticator ενός πακέτου RADIUS. Στην πραγματικότητα, όλη η ασφάλεια στο RADIUS βασίζεται σε αυτά τα πεδία πιστοποίησης, καθώς χρησιμεύουν ως μοναδικοί και τυχαίοι "ταυτοποιητές" (δεν πρέπει να συγχέεται με το πεδίο ID του πακέτου) για κάθε πακέτο. Παρ' όλα αυτά, η απόλυτη ασφάλεια εξαρτάται στο πόσο τυχαία αυτοί οι authenticators δημιουργούνται. Η μέθοδος ασφαλείας αυτή καταρρέει όταν χρησιμοποιούνται γεννήτριες τυχαίων αριθμών με πολύ μικρό κύκλο επανάληψης ή όταν οι τιμές επαναλαμβάνονται.

3.7.1 Επαναλαμβανόμενοι αυθεντικοποιητές αιτήματος και το χαρακτηριστικό User-Password

Είναι πιθανό να δημιουργηθεί ένα σύνολο από αυθεντικοποιητές αιτήματος και τα ανάλογα χαρακτηριστικά User-Password εάν ο κακόβουλος χρήστης μπορεί να υποκλέψει κίνηση στη γραμμή μεταξύ RADIUS client και RADIUS server κατά τη διάρκεια μιας συναλλαγής. Τότε μπορεί να ελέγξει εάν χρησιμοποιούνται επαναλαμβανόμενες τιμές για τον αυθεντικοποιητή αιτήματος, και εάν υπάρχουν μπορεί να αποσπάσει το διαμοιραζόμενο μυστικό από τις 16 πρώτες οκτάδες 16 του password. Κάνοντας αυτό, παίρνει τις πρώτες 16 οκτάδες από 2 εντελώς απροστάτευτων κωδικών πρόσβασης που έχουν γίνει XOR μαζί.

Το συμπέρασμα είναι ότι ο επιτιθέμενος έχει πάρει τις 16 πρώτες οκτάδες απροστάτευτα. Οι περισσότεροι κωδικοί πρόσβασης που διαλέγουν οι χρήστες δυστυχώς δεν είναι τόσο μεγάλοι, και ακόμα και να ήταν, ο επιτιθέμενος χρήστης έχει την βάση για να κάνει αργότερα επίθεση brute-force. Ο επιτιθέμενος δεν μπορεί να πάρει καμία πληροφορία μονάχα εάν οι χρήστες έχουν τυχαίους κωδικούς πρόσβασης ιδίου μήκους, που είναι μία πολιτική που θα πρέπει να υιοθετείται και να εφαρμόζεται από τον διαχειριστή του συστήματος. Για αυτή την επίθεση, ο επιτιθέμενος χρήστης χρειάζεται 2 διαφορετικά passwords με αξιοσημείωτη διαφορά μήκους. Εφόσον το μικρότερου μήκους password έχει περισσότερους χαρακτήρες συμπλήρωσης (padding), μόλις ολοκληρώσει την XOR θα έχει αποκαλύψει τους μη επικαλυπτόμενους χαρακτήρες του μεγαλυτέρου password με την λιγότερη προσπάθεια από την μεριά του.

Μια άλλη επίθεση μπορεί να επιτευχθεί εάν ένας κακόβουλος χρήστης προσπαθήσει να πιστοποιηθεί πολλαπλές φορές χρησιμοποιώντας γνωστά password και υποκλέψει τα συσχετιζόμενα με αυτά πακέτα Access-Request. Από αυτά τα πακέτα, μπορεί να αποσπάσει τον αυθεντικοποιητή αιτήματος και το χαρακτηριστικό User-Password. Έπειτα με το αποτέλεσμα από το XOR του γνωστού password με το χαρακτηριστικό User-Password που έχει υποκλαπεί, έχουμε ένα σύνολο και από τις τιμές request-authenticatorκαι τις τιμές του MD5 (αυθεντικοποιητής αιτήματος και διαμοιραζόμενο μυστικό). Εάν συνεχιστεί η υποκλοπή και η παρατηρηθεί τιμή ενός αυθεντικοποιητή αιτήματος και ταιριάζει με ένα από την λίστα του, μπορεί να αποκτήσει τις πρώτες 16οκτάδες του User-Password κοιτώντας το άθροισμα MD5 από τη λίστα και εφαρμόζοντας XOR σε αυτό με το χαρακτηριστικό User-Password. Επίσης, Με τη χρήση της λίστα αυτής από αυθεντικοποιητές αιτήματος,, μπορεί επίσης να προσθέσει κατάλληλους identifiers και τις απαντήσεις server responses που αποκόμισε από την υποκλοπή του. Έτσι μπορεί να υποκριθεί ως ο server και να απαντά τις παλιές απαντήσεις όταν έρθει το κατάλληλο πακέτο Access-Request packet.

3.7.2 Διαμοιραζόμενα μυστικά (Shared secrets)

Η χρήση ενός διαμοιραζόμενου μυστικού στο πρωτόκολλο RADIUS είναι μία από τις χειρότερες πιθανές αποφάσεις σχεδιασμού στα πλαίσια ασφάλειας δικτύου. Εξ αρχής, η προδιαγραφή του πρωτοκόλλου RADIUS επιτρέπει το ίδιο διαμοιραζόμενο μυστικό να χρησιμοποιείται ανάμεσα σε οποιοδήποτε αριθμό πελατών. Εξ αιτίας αυτού, από την μεριά του κακόβουλου χρήστη, όλοι οι RADIUS clients που χρησιμοποιούν το ίδιο διαμοιραζόμενο μυστικό μπορούν αποτελεσματικά να θεωρηθούν ο ίδιος πελάτης για τους σκοπούς του. Εάν ένας πελάτης είναι "ελαττωματικός", αυτή η μηχανή μπορεί να εκτεθεί και να χρησιμοποιηθεί για να εκθέσει και άλλες μηχανές που δεν έχουν κληρονομήσει το ελάττωμα compromise, μιας και το διαμοιραζόμενο μυστικό έχει αποκαλυφθεί. Επιπλέον, επιτρέπονται μονάχα χαρακτήρες ASCII για τη δημιουργία του διαμοιραζόμενο μυστικό, και είναι στο σύνολό τους 94. Επίσης, το διαμοιραζόμενο μυστικό συνήθως περιορίζεται σε 16 ή και λιγότερους χαρακτήρες. Αυτό καθιστά πάρα που εύκολο στον επιτιθέμενο να το μαντέψει, μιας και έχει πεπερασμένα όρια και για τους χαρακτήρες και για το μήκος του μυστικού που προσπαθεί αν μαντέψει.

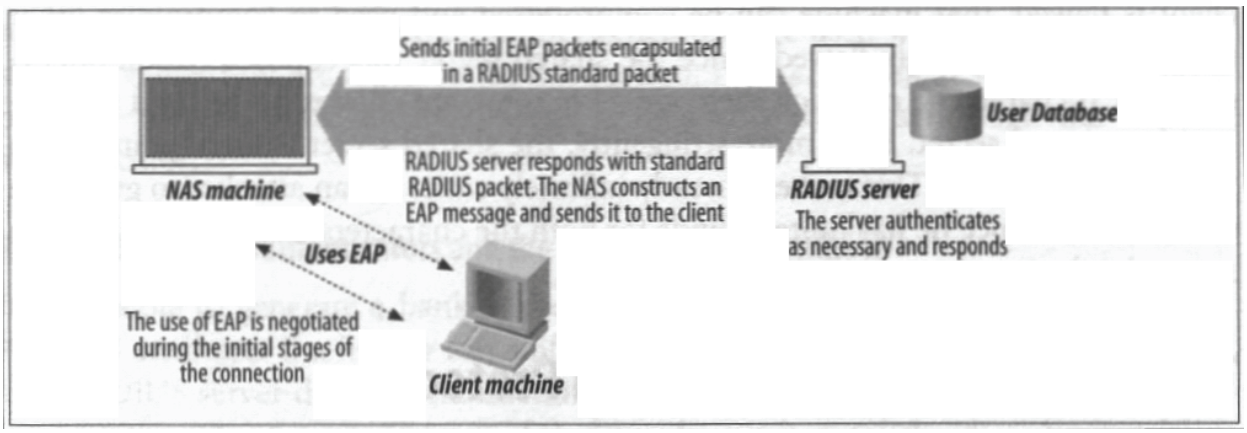
3.8 Το Extensible Authentication Protocol (EAP)

Το EAP είναι μία επέκταση του πρωτοκόλλου PPP που επιτρέπει μια πληθώρα από πρωτόκολλα πιστοποίησης να χρησιμοποιηθούν. Το EAP δεν περιορίζεται στενά από τη μέθοδο ασφαλείας. Προωθεί την εναλλαγή των μηνυμάτων αυθεντικοποίησης, επιτρέποντας στο λογισμικό πιστοποίησης που είναι αποθηκευμένο σε έναν server να αλληλεπιδρά με το αντίστοιχο στο πελάτη. Το EAP δρα σαν ένα είδος πρωτοκόλλου αντικατάστασης, επιτρέποντας την αρχική διαπραγμάτευση ενός πρωτοκόλλου πιστοποίησης (όπως το CHAP και MS-CHAP Version 1 και 2) και έπειτα την αποδοχή της σύνδεσης και στα δύο άκρα υπό μορφή link, το οποίο είναι μια συγκεκριμένη διάταξη EAP-πιστοποίησης. Μόλις και τα δύο μέλη επιβεβαιωθούν, το EAP επιτρέπει μία "ανοιχτή" επικοινωνία μεταξύ του RADIUS server και του πελάτη του.

Το EAP έχει σχεδιαστεί για να λειτουργεί σαν ένα "plug-in" αυθεντικοποίησης, με βιβλιοθήκες και στον πελάτη και στον server σε μια σύνδεση PPP. Κάθε διάταξη πιστοποίησης συσχετίζεται με ένα

συγκεκριμένο φάκελο βιβλιοθήκης, και μόλις μια συγκεκριμένη βιβλιοθήκη υπάρχει και στα δύο άκρα της σύνδεσης αυτή η νέα διάταξη μπορεί να χρησιμοποιηθεί. Κατά συνέπεια, το πρωτόκολλο μπορεί εύκολα να επεκτείνει τη χρηστικότητα του επεκτάσεις οποιαδήποτε στιγμή χωρίς να είναι απαραίτητος ο ανασχεδιασμός ολόκληρου του πρωτοκόλλου. Το EAP αυτή τη στιγμή είναι σε θέση αν υποστηρίξει διατάξεις αυθεντικοποίησης όπως Generic Token Card, OTP, MD5-Challenge, και Transport Level Security (TLS) για χρήση σε εφαρμογές smart-card και την υποστήριξη πιστοποιητικών. Επιπλέον για την υποστήριξη PPP, το EAP υποστηρίζεται και στο επίπεδο σύνδεσης όπως περιγράφεται στο IEEE 802. Το IEEE 802.lx ορίζει τη χρήση του EAP για πιστοποίηση συσκευές 802, όπως WiFi access points και Ethernet switches.

Το EAP συσχετίζεται με το RADIUS εξασφαλίζοντας μεγαλύτερη ασφάλεια για αυτό. Η χρήση RADIUS με EAP δεν είναι επίσημη διάταξη για το EAP, αλλά μπορούμε να το δούμε ως προώθηση μηνυμάτων οποιοδήποτε τύπου EAP από τον εξοπλισμό του πελάτη RADIUSκαι του RADIUS server. Το EAP μέσω RADIUS συνήθως ακολουθεί την έξης νοοτροπία: ο access server είναι ρυθμισμένος να χρησιμοποιεί το EAP αλλά και το RADIUS σαν πάροχο αυθεντικοποίησης. Όταν ένας πελάτης υπηρεσίας επιχειρεί να συνδεθεί, διαπραγματεύεται τη χρήση του EAP με τον εξοπλισμό του RADIUS client. Ο τελικός χρήστης τότε στέλνει ένα μήνυμα EAP στο RADIUS client, και ο RADIUS client εσωκλείει το μήνυμα EAP σαν μήνυμα RADIUS και το αποστέλλει στο RADIUS server. Ο RADIUS server δρα στο ενθυλακωμένο μήνυμα και αποστέλλει ένα τύπου RADIUS μήνυμα πίσω στο RADIUS client. Ο RADIUS client τότε κατασκευάζει ένα μήνυμα EAP από το μήνυμα RADIUS και το στέλνει πίσω στο τελικό χρήστη (ή πελάτη υπηρεσίας). Η παρακάτω εικόνα παρουσιάζει αυτή τη ροή.



Εικόνα 3 Λειτουργία EAP και RADIUS μαζί

3.9 Αντιστάθμιση των ατελειών του πρωτοκόλλου

Τα προβλήματα ασφάλειας που αναφέρθηκαν έχουν τρόπους επίλυσης. Μερικοί τρόποι παρουσιάστηκαν μαζί με την αναφορά του προβλήματος. Σε αυτή την ενότητα παρουσιάζονται μερικά βασικά βήματα για την αντιμετώπιση των βασικότερων προβλημάτων:

1. Χρήση του πρωτοκόλλου IPsec με ESP και αλγορίθμου κρυπτογράφησης όπως ο 3DES. Όταν το IPsec κρυπτογραφεί ολόκληρο το μήνυμα RADIUS, τα πεδία του αυθεντικοποιητή αιτήματος και το User-Password, Tunnel-Password, και MPPE-Key attributes δεν είναι ορατά. Για αποκρυπτογράφηση αυτών τον πεδίων, ο επιτιθέμενος πρέπει πρώτα να "σπάσει" το προστατευμένο μήνυμα ESP. Αυτή η μέθοδος προστατεύει ολόκληρο το μήνυμα RADIUS.
2. Απαίτηση οποιοδήποτε από τα διαμοιραζόμενα μυστικά που χρησιμοποιούνται να είναι είτε 22χαρακτήρες πληκτρολογίου είτε μήκους 32 δεκαεξαδικών ψηφίων. Αυτό το μέτρο προστατεύει ενάντια στις αδυναμίες της απροστάτευτης φύσης της λογικής διαμοιραζόμενου μυστικού που χρησιμοποιείται.

3. Χρήση διαφορετικού διαμοιραζόμενου μυστικού για κάθε ζεύγος RADIUS client και server. Αυτό είναι μονάχα ένα βασικό μέτρο ασφαλείας, όπως είναι η λογική να υπάρχει διαφορετικός κωδικός πρόσβασης για τα διάφορα web sites και υπολογιστικοί πόροι (π.χ. CPU).
4. Χρήση του χαρακτηριστικού Message-Authenticator σε όλα τα μηνύματα Access-Request. Από τη μεριά του πελάτη, πρέπει να επιβεβαιωθεί ότι χρησιμοποιείται ο Message-Authenticator και πρέπει να εξασφαλιστεί ότι μπορεί να ρυθμιστεί κατάλληλα. Από τη μεριά του server, πρέπει να εξασφαλιστεί ότι το χαρακτηριστικό Message-Authenticator είναι παρών και επίσης επιτρέπεται η ρύθμιση του. Αυτό επιτυγχάνει να μην υπάρχουν πιστοποιημένα μηνύματα Access-Request οπουδήποτε κατά μήκος της διαδρομής ανταλλαγής μηνυμάτων.
5. Χρήση μίας γεννήτριας τυχαίων αριθμών με " κρυπτογραφική αξία " για την δημιουργία του αυθεντικοποιητή αιτήματος. Αυτό αντισταθμίζει την κατά τ' άλλα περιορισμένη ποιότητα της υλοποίησης του αυθεντικοποιητή αιτήματος.

Θα πρέπει επίσης να προστατευθούν τα links από το τελικό χρήστη στον εξοπλισμό του RADIUS client. Με τη χρήση του EAP και ενός από τους ισχυρούς τύπους κρυπτογράφησης που διατίθενται για τη χρήση του, όπως για παράδειγμα το EAP-TLS, που είναι μία ισχυρή μέθοδος EAP που απαιτεί ανταλλαγή πιστοποιητικών και από το client και το RADIUS server. Η χρήση μηνυμάτων EAP έχει ως επακόλουθο την απαίτηση έγκυρου πιστοποιητικού Message-Authenticator, το οποίο προστατεύει τα μηνύματα που δεν μπορούν να προστατευθούν με άλλο τρόπο χρησιμοποιώντας το IPsec.

Επίσης, μαζί με το EAP, θα ήταν συνετό να χρησιμοποιηθούν αμοιβαίες μέθοδοι πιστοποίησης. Για παράδειγμα, και τα δύο άκρα της σύνδεσης να πιστοποιούν το peer τους με αμοιβαία αυθεντικοποίηση. Με αυτό τον τρόπο η προσπάθεια πιστοποίησης απορρίπτεται εάν στο άλλο άκρο αποτύχει η πιστοποίηση. Το EAP-TLS είναι μία αμοιβαία μέθοδος αυθεντικοποίησης: ο RADIUS server επικυρώνει το πιστοποιητικό του client, και ο client επικυρώνει το πιστοποιητικό του υπολογιστή του RADIUS server.

Τέλος, εάν το πρωτόκολλο πιστοποίησης PAP δεν απαιτείται, θα ήταν καλό να μην είναι ενεργοποιημένο και στα δύο άκρα client και server. Το PAP θα πρέπει να χρησιμοποιείται ως ασφαλή σύνδεση όταν χρησιμοποιείται σε συνδυασμό με πιστοποίηση OTP και Token Card όπου λογικά το

password πολύπλοκο και αλλάζει με κάθε χρήση. Παρ' όλα αυτά, ακόμα και σε αυτή την περίπτωση, έχοντας ενεργοποιημένο το PAP επιτρέπει σε "κακορυθμισμένους" τελικούς χρήστες να διαπραγματεύονται με τον εξοπλισμό του RADIUS client, και μπορεί ενδεχομένως να αποστείλουν μη προστατευμένα. Εάν είναι δυνατό και σε αυτή την περίπτωση, συνίσταται η χρήση EAP με τους τύπους πιστοποίησης OTP και Token Card αντί του PAP. Κατά τον ίδιο τρόπο σκέψης, καλό είναι να απενεργοποιηθεί η κωδικοποίηση LAN Manager αν χρησιμοποιείται το MS-CHAP.

Κεφάλαιο 4 FreeRADIUS

4.1 FreeRADIUS

Ο **FreeRADIUS** είναι ένας δωρεάν ανοικτού λογισμικού RADIUS server. Προσφέρει μία εναλλακτική λύση σε σχέση με άλλους επαγγελματικούς RADIUS servers, μιας και είναι ένας από τους πιο αρθρωτούς και πλούσιους σε δυνατότητες RADIUS servers διαθέσιμους σήμερα. και θεωρείται από τους περισσότερο εφαρμόσιμους RADIUS servers παγκοσμίως από άποψη αριθμού υλοποιήσεων, και αριθμό χρηστών που πιστοποιούνται μέσω αυτού κάθε μέρα. Κλιμακώνεται από ενσωματωμένα συστήματα (embedded systems**)** με μικρά ποσά μνήμης, έως συστήματα με πολλά εκατομμύρια χρηστών. Είναι γρήγορος, ευέλικτος,, εύκολος σε ρύθμιση και υποστηρίζει περισσότερα πρωτόκολλα πιστοποίησης από τις περισσότερες εμπορικές υλοποιήσεις. Ο server έχει ένα εργαλείο web διαχείρισης βασισμένο σε PHP, που ονομάζεται dialupadmin και επί του παρόντος χρησιμοποιείται ως η βάση για πολλά εμπορικά προϊόντα RADIUS.

Έχει γραφτεί από μια ομάδα από developers που έχουν πολύ μεγάλη εμπειρία στην εφαρμογή και εγκατάσταση λογισμικού RADIUS, σε software engineering, και διαχείριση πακέτων Unix.

4.1.2 Ιστορία

Ο FreeRADIUS ξεκίνησε τον Αύγουστο του 1999 από τον Alan DeKok και τον Miquel van Smoorenburg. Ο Miquel πιο πριν είχε γράψει το Cistron RADIUS server, που είχε αποκτήσει ευρεία χρήση όταν σταμάτησε η υποστήριξη και ανάπτυξη του Livingston server. Ο FreeRADIUS ξεκίνησε να δημιουργεί έναν νέο RADIUS server, με τη χρήση νέου σχεδιασμού που θα ενθάρρυνε περισσότερο την ενεργή ανάμειξη της κοινότητας, και κλιμάκωση που παλαιότερα οι νεότεροι servers δεν μπορούσαν να αντεπεξέλθουν. Πολύ γρήγορα ο server απέκτησε την υποστήριξη της, με την προσθήκη αρθρωμάτων για ενσωμάτωση των LDAP, SQL, και άλλων βάσεων δεδομένων. Η υποστήριξη του EAP προστέθηκε τέλος του 2001, και του PEAP και του EAP-TTLS στο τέλος του

2003. Τώρα πια ο server υποστηρίζει όλα τα κοινά πρωτόκολλα πιστοποίησης, βάσεις δεδομένων και κατάλογοι κατασκευαστών (vendor dictionaries).

Η έκδοση 2.0.0 δημοσιεύτηκε στις αρχές του 2008, και προστέθηκε πειραματικά η υποστήριξη πολύ περισσότερων τύπων EAP (EAP-FAST, EAP-TNC, κτλ.). Επίσης προστέθηκε και ή υποστήριξη για Virtual hosting, IPv6, και νέας γλώσσας πολιτικής που απλοποιεί κατά πολύ πολλές πολύπλοκες ρυθμίσεις.

Κεφάλαιο 5 Υλοποίηση FreeRADIUS server

5.1 Εγκατάσταση Λειτουργικού

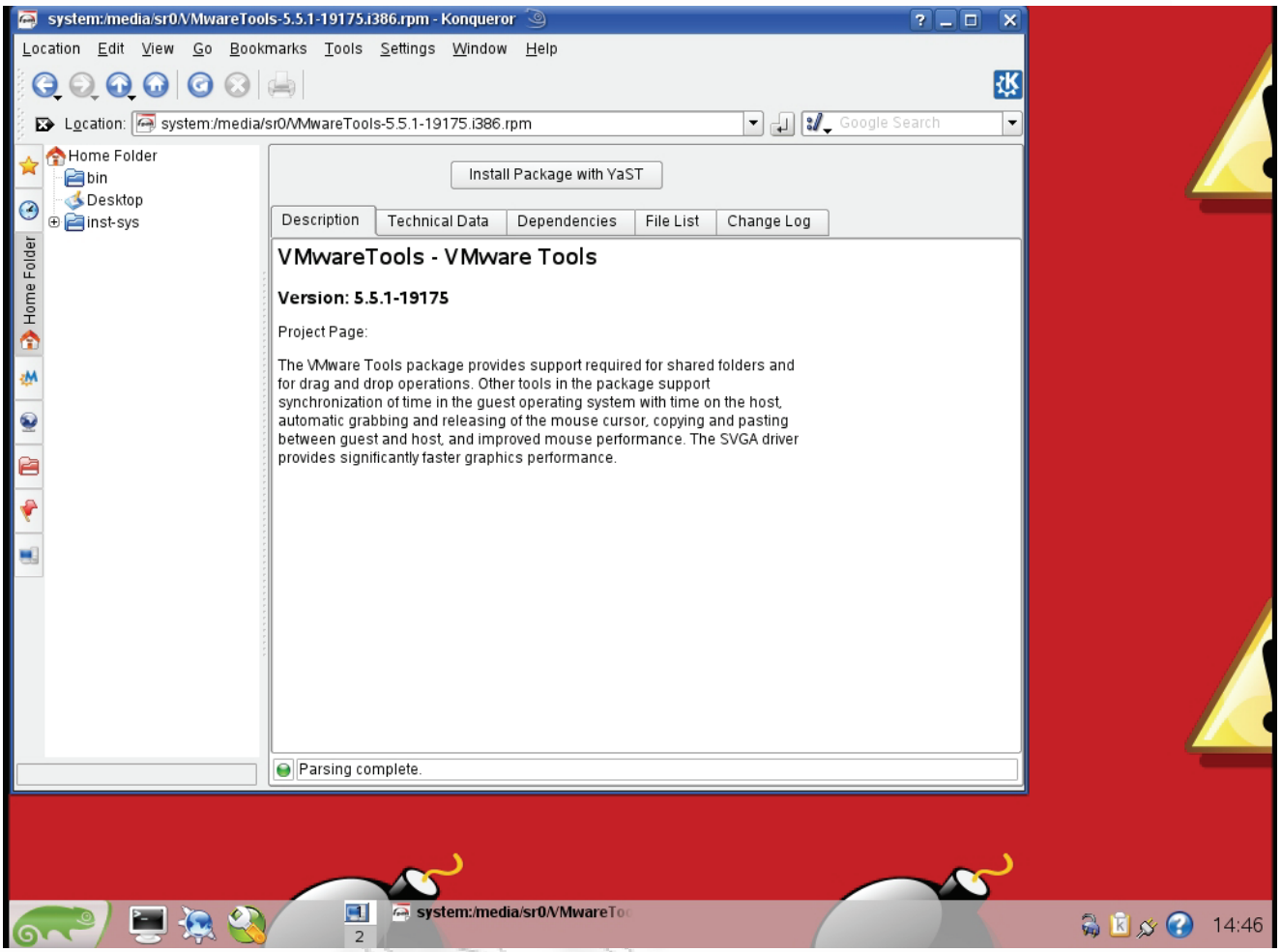
Εγκαθιστούμε το λειτουργικό opensuse 10.3 και βάζουμε για root password **admin**. Έπειτα στην επόμενη οθόνη απενεργοποιούμε το firewall του λειτουργικού.

5.2 Εγκατάσταση VMWare tools

Αφού εκκινήσουμε για πρώτη φορά την εικονική μηχανή (VM) με το λειτουργικό εγκαθιστούμε τα VMware tools που μας διευκολύνουν στην καλύτερη αλληλεπίδραση μεταξύ των 2 λειτουργικών. Όπως δείχνει και η εικόνα 4 επιλέγουμε από την μπάρα εργαλείων του VMware το VM και επιλέγουμε Install VMware tools.

Εικόνα 4 Εγκατασταση VMware tools βημα 1

Έπειτα επιλέγουμε την εκτέλεση του πακέτου VMwaretools.rpm με το Yast. Όπως φαίνεται και στην εικόνα 5



Εικόνα 5 Εγκατάσταση VMware tools βήμα 2

5.3 Εγκατάσταση FreeRADIUS

Θα εγκαταστήσουμε την τελευταία έκδοση του FreeRADIUS που έχει φτάσει αισίως την 2.0.4. Αρχικά κατεβάζουμε το αρχείο από την ιστοσελίδα http://www.freeradius.org. Έπειτα αποσυμπιέζουμε το αρχείο με την ακόλουθη εντολή.

 # tar -zxvf freeradius.tar.gz

Για να ξεκινήσει η εγκατάσταση πάμε στο φάκελο που δημιουργήθηκε από την αποσυμπίεση και εκτελούμε την εντολή

 # ./configure --localstatedir=/var --sysconfdir=/etc

Τα αρχεία θα ετοιμαστούν για να γίνουν compile. Μετά εκτελούμε την εντολή make για να κάνει compile τα binary αρχεία, και τέλος εκτελούμε την εντολή make install για να τοποθετηθούν τα αρχεία στις κατάλληλες θέσεις.

Με τη χρήση του εργαλείου NTRadPing κάνουμε έλεγχο ότι ο Radius Server λειτούργει κανονικά. Το NTRadPing είναι ένα δωρεάν εύκολο εργαλείο ελέγχου που τρέχει σε Microsoft Windows και είναι διαθέσιμο στη διεύθυνση http://www.mastersoft-group.com/download/. Τα βήματα που ακολουθουμε για τον έλεγχο είναι τα ακόλουθα:

1. Εισάγουμε την διεύθυνση IP του FreeRADIUS server στο πεδίο RADIUS Server/port και τη θύρα που λειτουργεί ο server στο επόμενο πεδίο (συνήθως 1812)
2. Γράφουμε το μυστικό κλειδί που έχουμε βάλει στο /etc/raddb/clients για αυτή την κονσόλα Windows.Στο αρχείο clients.conf ορίζουμε το subnet από το οποίο μπορεί να δεχθεί αιτήματα και ορίζουμε κωδικό πρόσβασης για αυτό. Στη περίπτωση μας το πεδίο είναι το ακόλουθο:

```
client 192.168.2.0/24{
  secret        =testing123
  shortname     =private-network-1
  }
```

3. Στο πεδίο username, εισάγουμε root και στο πεδίο password το κωδικό πρόσβασης του root για το σύστημα όπου τρέχει ο FreeRADIUS server.
4. Διαλέγουμε Authentication Request από την λίστα Request Type.
5. Πατάμε Send.

Πρέπει να τονιστεί ότι για να πάρουμε απάντηση Access-Accept με username root και password το κωδικό πρόσβασης του root, πρέπει να τα εκτελούμε στον ίδιο το FreeRADIUS server.

Εάν ο server δουλεύει σωστά και έχει εισαχθεί έγκυρος κωδικός πρόσβασης root, η απάντηση στο πεδίο απάντησης του NTRadPing είναι κάτι σαν την ακόλουθη:

```
Sending authentication request to server 192.168.2.60:1812
Transmitting packet, code=i id=l length=47
Received response from the server in 15 milliseconds
Reply packet code=2 id=l length=20
Response: Access-Accept
----------attribute dump----------
```

Τώρα, εάν αλλαχθεί ο κωδικός πρόσβασης του root στο NTRadPing σε κάτι μη σωστό και αποσταλεί ξανά το αίτημα θα ληφθεί ένα μήνυμα Access-Reject, που μοιάζει με το ακόλουθο:

```
Sending authentication request to server 192.168.2.60:1812
Transmitting packet, code=l id=3 length=47
No response from server (timed out), new attempt (#l)
Received response from the server in 3516 milliseconds
Reply packet code=3 id=3 length=20
Response: Access-Reject
----------attribute dump----------
```

Έπειτα θα πρέπει να δοκιμαστούν τα πακέτα παρακολούθησης (accounting packets). Στο NTRadPing επιλέγουμε *Accounting Start* από τη λίστα Request Type. Ελέγχουμε ότι έχουμε βάλει

πάλι το σωστό κωδικό πρόσβασης root, και αποστέλλουμε το αίτημα. Η απάντηση μοιάζει με το ακόλουθο:

```
Sending authentication request to server 192.168. 2.60:1812
Transmitting packet, code=4 id=5 length=38
Received response from the server in 15 milliseconds
Reply packet code=5 id=5 length=20
Response: Accounting-Response
----------attribute dump----------
```

Τέλος, σταματάμε την διαδικασία παρακολούθησης αλλάζοντας το Request Type σε *Accounting Stop* και ξαναστέλνουμε το αίτημα. Η απάντηση μοιάζει με το ακόλουθο:

```
Sending authentication request to server 192.168. 2.60:1812
Transmitting packet, code=4 id=6 length=38
Received response from the server in 16 milliseconds
Reply packet code=5 id=6 length=20
Response: Accounting-Response
----------attribute dump----------
```

Εάν λάβουμε επιτυχής απαντήσεις και στα 4 ping τεστ, τότε ο FreeRADIUS λειτουργεί σωστά.

Για να λειτουργήσει σε άλλο υπολογιστή θα πρέπει να ενεργοποιήσουμε και ένα όνομα χρήστη με ένα κωδικό στο users.conf. Για ευκολία ενεργοποιούμε το ακόλουθο χρήστη που μας δίνει απάντηση εάν αποδεχθεί το αίτημα πιστοποίησης ο server το μήνυμα Hello. Η ρύθμιση είναι η ακόλουθη που βγαίνει από σχόλια:

```
"John Doe" Cleartext-Password := "hello"
        Reply-Message = "Hello, %{User-Name}"
```

Επομένως εάν δώσουμε username "John Doe" και password "hello" θα πάρουμε το ακόλουθο μήνυμα:

```
Sending authentication request to server 192.168. 2.60:1812
Transmitting packet, code=4 id=5 length=38
Received response from the server in 15 milliseconds
Reply packet code=5 id=5 length=20
Response: Accounting-Response
-------------------------attribute dump-------------------------------
Reply-Message= Hello, John Doe
```

Εάν λάβουμε επιτυχής απαντήσεις και στα 5 ping τεστ, τότε ο FreeRADIUS λειτουργεί σωστά.

5.4 Εγκατάσταση MySQL

Η MySQL είναι προεγκατεστημένη με τη διανομή που χρησιμοποιούμε. Αρχικά πρέπει να οριστεί κωδικός πρόσβασης για τη βάση δεδομένων μιας και η εγκατάσταση της δεν ορίζει. Αυτό επιτυγχάνεται με την ακόλουθη εντολή:

mysqladmin –u root password admin

Με την παραπάνω εντολή ορίσαμε κωδικό πρόσβασης για τον root χρήστη της βάσης δεδομένων τη λέξη “admin”. Επιλέξαμε τη συγκεκριμένη λέξη ώστε να μη γίνει σύγχυση κωδικών πρόσβασης μιας και τον ίδιο κωδικό έχουμε και για την είσοδο στο σύστημα.

Επόμενο βήμα είναι να δημιουργηθεί μια βάση δεδομένων. Αφού εισέλθουμε στην mysql με την εντολή

mysql -u root -p mysql

Εκτελούμε τις ακόλουθες εντολές:

> create database radius;

> grant all privileges on radius. to 'dialupadmin'@localhost identified by 'admin' with grant option;*

Τώρα θα δημιουργήσουμε τις καταχωρήσεις με το script db_mysql.sql η δομή του οποίου υπάρχει σε ένθετο στο τέλος. Για να το επιτύχουμε αυτό θα εκτελέσουμε την ακόλουθη εντολή:

mysql -uroot –padmin radius < db_mysql.sql

Για να επαληθεύουμε ότι δημιουργήθηκαν οι καταχωρήσεις εκτελούμε τα παρακάτω:

 # mysql –u root –p mysql

mysql> use radius;

Database changed

mysql> show tables;

+------------------+mysql

| Tables_in_radius |

+------------------+

| nas |

| radacct |

| radcheck |

| radgroupcheck |

| radgroupreply |

| radpostauth |

| *radreply* |

| *usergroup* |

+-----------------+

Έπειτα πρέπει να ρυθμιστεί ο FreeRADIUS ώστε να χρησιμοποιεί τη βάση δεδομένων MySQL για όλες τις RADIUS λειτουργίες του. Μέχρι πριν την έκδοση 2.0.0 η ρύθμιση αυτή γινόταν στο αρχείο radiusd.conf που βρίσκεται στο φάκελο /etc/raddb/, όμως για λόγους ευκολίας και απλοποίησης του κώδικα ρυθμίσεων του FreeRADIUS από την έκδοση 2.0.0 αυτό γίνεται στο αρχείο default που βρίσκεται στο φάκελο /etc/raddb/sites-enabled/. Η αλλαγή αυτή καθιστά δυνατό να έχουμε διαφορετικές ρυθμίσεις ανά εικονική συσκευή χρήστη. Το σύστημα εάν δεν έχει πολλούς virtual hosts χρησιμοποιεί τις ρυθμίσεις που έγιναν στο αρχείο default. Για την ρύθμιση χρήσης της sql αφαιρούμε τα σχόλια (τα σχόλια προσδιορίζονται με την δίεση "#") από τη λέξη sql στα τμήματα authorize και accounting και βάζουμε σε σχόλιο τις λέξεις unix και file στα αντίστοιχα τμήματα.

Κεφάλαιο 6 Σενάρια ασκήσεων

Άσκηση 1 Έλεγχος σωστής λειτουργίας RADIUS server

Με τη χρήση του εργαλείου NTRadPing να γίνει έλεγχος ότι ο Radius Server λειτούργει κανονικά.

Λύση

Διαδικασία Επίλυσης:

1. Εισάγουμε την διεύθυνση IP του FreeRADIUS server στο πεδίο RADIUS Server/port και τη θύρα που λειτουργεί ο server στο επόμενο πεδίο (συνήθως 1812)
2. Γράφουμε το μυστικό κλειδί που έχουμε βάλει στο /etc/raddb/clients για αυτή την κονσόλα Windows
3. Στο πεδίο username, εισάγουμε root και στο πεδίο password το κωδικό πρόσβασης του root για το σύστημα όπου τρέχει ο FreeRADIUS server
4. Διαλέγουμε Authentication Request από την λίστα Request Type.
5. Πατάμε Send.

Εάν ο server δουλεύει σωστά και έχει εισαχθεί έγκυρος κωδικός πρόσβασης root, η απάντηση στο πεδίο απάντησης του NTRadPing είναι κάτι σαν την ακόλουθη:

Sending authentication request to server 192.168.2.60:1812

Transmitting packet, code=i id=l length=47

Received response from the server in 15 milliseconds

Reply packet code=2 id=l length=20

Response: Access-Accept

----------attribute dump----------

Τώρα, εάν αλλαχθεί ο κωδικός πρόσβασης του root στο NTRadPing σε κάτι μη σωστό και αποσταλεί ξανά το αίτημα θα ληφθεί ένα μήνυμα Access-Reject, που μοιάζει με το ακόλουθο:

Sending authentication request to server 192.168.2.60:1812

Transmitting packet, code=l id=3 length=47

No response from server (timed out), new attempt (#l)

Received response from the server in 3516 milliseconds

Reply packet code=3 id=3 length=20

Response: Access-Reject

----------attribute dump----------

Έπειτα θα πρέπει να δοκιμαστούν τα πακέτα παρακολούθησης (accounting packets). Στο NTRadPing επιλέγουμε *Accounting Start* από τη λίστα Request Type. Ελέγχουμε ότι έχουμε βάλει πάλι το σωστό κωδικό πρόσβασης root, και αποστέλλουμε το αίτημα. Η απάντηση μοιάζει με το ακόλουθο:

Sending authentication request to server 192.168. 2.60:1812

Transmitting packet, code=4 id=5 length=38

Received response from the server in 15 milliseconds

Reply packet code=5 id=5 length=20

Response: Accounting-Response

----------attribute dump----------

Τέλος, σταματάμε την διαδικασία παρακολούθησης αλλάζοντας το Request Type σε *Accounting Stop* και ξαναστέλνουμε το αίτημα. Η απάντηση μοιάζει με το ακόλουθο:

```
Sending authentication request to server 192.168. 2.60:1812
Transmitting packet, code=4 id=6 length=38
Received response from the server in 16 milliseconds
Reply packet code=5 id=6 length=20
Response: Accounting-Response
-------------------------attribute dump-------------------------------
```

Εάν λάβουμε επιτυχής απαντήσεις και στα 4 ping τεστ, τότε ο FreeRADIUS λειτουργεί σωστά.

Άσκηση 2 Δημιουργία Χρηστών

2.1 Αναζητήστε το αρχείο καταγραφής χρηστών και παρατηρήστε τους χρήστες και τα group που είναι δηλωμένα στο σύστημα. Ο υπολογιστής σας ανήκει σε κάποιο από αυτά;

2.2 Να προστεθεί ένας νέος χρήστης στον Radius Server και να γίνει είσοδος στο σύστημα με τα εμπιστευτικά στοιχειά που δόθηκαν. Να καταγραφεί το log του NTRadPing

Άσκηση 3 Εγκατάσταση και χρήση του passgen

3.1 Εγκατάσταση του προγράμματος passgen

Η εφαρμογή 'passgen' είναι διαθέσιμη από το ακόλουθο site http://www.linuxbuilt.com/software/passgen/, είναι απλή στη χρήση την εγκατάσταση και παράγει κωδικούς πρόσβασης πολύ γρήγορα.

Τα βήματα της εγκατάστασης του προγράμματος είναι τα ακόλουθα:

 # tar -jxvf passgen-0.4.1.tar.bz2

 # cd pass passgen-0.4.1

 # ./configure

 # make

 # make install

και εγκατάσταση ολοκληρώθηκε. Για μία δοκιμή όπου θα δημιουργηθούν 4 passwords μήκους 6 χαρακτήρων τρέχουμε την ακόλουθη εντολή.

 # passgen -g 1000 -l 6

3.2 Με τη χρήση προγράμματος passgen και του script passgenscr.sh να δημιουργηθούν 1000 usernames με τα αντίστοιχα passwords. Έπειτα να τροποποιηθεί το script ώστε εκτελώντας το να δημιουργούνται μόνο 20 usernames και passwords. Τέλος να ενσωματωθούν, να επαληθευθούν και να δοκιμαστούν μερικά από τα 20 usernames στο Radius server.

Λύση

Πριν εκτελέσουμε το script πρέπει να δημιουργήσουμε 2 βοηθητικούς φακέλους τον "/tmp/usergennum" και τον "/tmp/usergenname"

Για να εκτελεστεί το script εκτελούμε την εντολή:

./passgenscr.sh

Με την εκτέλεση του δημιουργούνται στο φάκελο /tmp/ 2 αρχεία το usergen.csv, που περιέχει τα usernames με τα αντίστοιχα passwords που δημιουργήθηκαν σε εκτυπώσιμη μορφή, και το usergen.sql που όταν εκτελεστεί ενσωματώνει τα usernames που δημιουργήσαμε με τους κωδικούς τους στη βάση δεδομένων του RADIUS server.

3.3 Να τροποποιηθεί το script ώστε εκτελώντας το να δημιουργούνται μόνο 20 usernames και passwords. Τέλος να ενσωματωθούν, να επαληθευθούν και να δοκιμαστούν μερικά από τα 20 usernames στο Radius server.

Λύση

Για να δημιουργήσουμε μονάχα 20 usernames και passwords πάμε στη γραμμή MAXCOUNT=1000 και γράφουμε 20.

Για να ενσωματώσουμε τα usernames στη βάση δεδομένων μας εκτελούμε την παρακάτω εντολή:

mysql –uroot –padmin radius < usergen.sql

Έπειτα εκτελώντας τις ακόλουθε εντολές στην mysql επιβεβαιώνουμε την προσθήκη των χρηστών.

mysql> use radius; (για χρήση της db radius)

mysql> show columns from usergroup;

*mysql> select * from usergroup;*

*mysql> select * from radreply;*

*mysql> select * from radcheck;*

Τέλος για να δοκιμάσουμε αν λειτουργεί σωστά και δέχεται τους χρήστες που δημιουργήσαμε εκτελούμε το εργαλείο NTRadPing με όνομα χρήστη και κωδικό πρόσβασης έναν από τη λίστα και παρατηρούμε το log προγράμματος.

Άσκηση 4 Χρήση μεθόδου πιστοποίησης EAP-TLS για αυθεντικοποίηση χρηστών σε ασύρματο δίκτυο με χρήση προστασίας WPA & WPA2 Enterprise

Στη παρούσα άσκηση θα χρησιμοποιήσουμε το FreeRADIUS Server για την πιστοποίηση χρηστών με τη χρήση της μεθόδου πιστοποίησης EAP-TLS σε ασύρματο περιβάλλον με ασφάλεια WPA ή WPA2 Enterprise.

Για την πραγματοποίηση της άσκησης θα χρειαστούμε επιπλέον ένα Access Point με δυνατότητα ασφαλείας WPA ή κατά προτίμηση WPA2 Enterprise που στην προκειμένη περίπτωση είναι το Linksys WRT54GL με firmware DD-WRT v24 RC-7, και εγκατεστημένο στο Linux περιβάλλον το OpenSSL 0.9.8.g ή μεταγενέστερο για τη δημιουργία των πιστοποιητικών.

Η άσκηση είναι χωρισμένη σε 2 μέρη. Στο πρώτο μέρος γίνεται παραμετροποίηση και εγκατάσταση των πιστοποιητικών και κλειδιών καθώς και οι ρυθμίσεις στο FreeRADIUS server. Στο δεύτερο μέρος γίνεται η παραμετροποίηση του εξοπλισμού του NAS (Linksys router) και των πελατών

Α΄ Μέρος Παραμετροποίηση OpenSSL και FreeRADIUS

4.1 Ρύθμιση OpenSSL

Αρχικά θα δημιουργήσουμε μερικούς φακέλους για να οργανώσουμε τα κλειδιά που θα δημιουργηθούν. Στο root φάκελο δημιουργούμε το φάκελο "CA" με τους υποφακέλους "signed_certs" και "private". Αυτό επιτυγχάνεται τρέχοντας διαδοχικά τις ακόλουθες εντολές

 # cd /root/

 # mkdir CA

 # mkdir CA/signed_certs

 # mkdir CA/private

 # chmod 700 CA/private

Στον υποφάκελο "signed_certs" σώζονται αντίγραφα από όλα τα πιστοποιητικά που υπογράφονται από την δική μας Certificate Authority (CA). Με αυτό τον τρόπο, εάν χρειαστεί να ανακαλεστεί ένα πιστοποιητικό, υπάρχει ένα αντίγραφο τοπικά.. Ο υποφάκελος "private" κρατά το προσωπικό κλειδί της CA (CA's private key). Είναι πάρα πολύ σημαντικό να μείνει το κλειδί της CA μυστικό, διότι εάν υποκλαπεί, θα μπορεί να υπογράφει μη εμπιστευτικά πιστοποιητικά που μπορεί να χρησιμοποιηθούν ώστε να ξεγελάσουν τους πελάτες ώστε να ανταλλάσουν ευαίσθητες πληροφορίες με μία μη έμπιστη μηχανή. Γι' αυτό το λόγο έγινε αλλαγή άδεια ανάγνωσης εγγραφής και εκτέλεσης του φακέλου ώστε μόνο ο root χρήστης να μπορεί να εκτελέσει.

Ένα επόμενο βήμα είναι να αντιγράψουμε το αρχικό αρχείο openssl.cnf και να το τροποποιήσουμε ώστε να μας διευκολύνει στην δημιουργία των πιστοποιητικών. Το αρχείο αυτό εκτελείται με την επιλογή "-config" κατά την εκτέλεση της εντολής του OpenSSL. Η αντιγραφή του από την αρχική του θέση γίνεται με την εντολή:

 # cp /etc/ssl/openssl.cnf /root/CA/

Ανοίγουμε το αρχείο openssl.cnf με το κειμενογράφο και αλλάζουμε τα σημεία όπως φαίνεται παρακάτω:

```
35 [ CA_default ]
36
37 dir       = /root/CA/                 # Where everything is kept
38 certs      = $dir/          # Where the issued certs are kept
39 crl_dir     = $dir/crl       # Where the issued crl are kept
40 database    = $dir/index.txt    # database index file.
41 #unique_subject = no          # Set to 'no' to allow creation of
42                     #several crtificates with same subject.
43 new_certs_dir  = $dir/signed_certs   # default place for new certs.
```

Εάν σκοπεύουμε να χρησιμοποιήσουμε τα Windows για τη διαχείριση του ασύρματου δικτύου στους πελάτες, πρέπει να προστεθούν και αυτές οι γραμμές στο τέλος του αρχείου "openssl.cnf":

```
316 # Windows XP TLS Extensions
317 [ xpclient_ext ]
318 extendedKeyUsage=1.3.6.1.5.5.7.3.2
319 [ xpserver_ext ]
320 extendedKeyUsage=1.3.6.1.5.5.7.3.1
```

Έπειτα από τη γραμμή 123 αλλάζουμε τις αρχικές τιμές στο τμήμα "distinguished name" ώστε να προσαρμοστεί η εφαρμογή στις ανάγκες μας. Το τμήμα "distinguished name" περιέχει μερικές χρήσιμες πληροφορίες που χαρακτηρίζουν τα δημόσια κλειδιά.. Μπορεί να γίνει αλλαγή στην αρχική τιμή για τις παραμέτρους προσθέτοντας "_default" στο τέλος του ονόματος της μεταβλητής. Για παράδειγμα όπως βλέπουμε παρακάτω, η μεταβλητή "countryName_default" είναι η αρχική τιμή για το "countryName".

```
123 [ req_distinguished_name ]
124 countryName           = Country Name (2 letter code)
125 countryName_default       = GR
126 countryName_min         = 2
127 countryName_max         = 2
...
```

Τέλος, εκτελούμε την εντολή touch "index.txt", που είναι μία απλή βάση δεδομένων σε μορφή κειμένου και χρησιμοποιείται ώστε να εντοπίζονται τα υπογεγραμμένα πιστοποιητικά

```
~/CA # touch index.txt
```

4.1.1 Δημιουργία CA και κλειδιών

Με τα προηγούμενα βήματα ρυθμίστηκε το περιβάλλον λειτουργίας, σε αυτό το κομμάτι θα δημιουργηθεί η CA και μερικά κλειδιά. Για τη δημιουργία ενός ζεύγους κλειδιών, αρχικά πρέπει να δημιουργηθεί ένα "αίτημα πιστοποιητικού" (υποεντολή "req"). Έπειτα το αίτημα πιστοποίησης αποστέλλεται για να υπογραφεί από την CA και γίνεται ένα αξιόπιστο δημόσιο κλειδί. Η δημιουργία του ζεύγους κλειδιού CA αρχικά έχει την ίδια διαδικασία με ένα κανονικό ζεύγος κλειδιού εκτελώντας την παρακάτω εντολή.

Για τις περισσότερες απαντήσεις απλά πατάμε Enter και αποδεχόμαστε τις προκαθορισμένες ρυθμίσεις από το αρχείο ρυθμίσεων που τροποποιήθηκε προηγουμένως. Θα πρέπει να τονιστεί ότι η επιλογή ισχυρού κωδικού πρόσβασης είναι ο μόνος τρόπος προστασίας από κάποιο κακόβουλο χρήστη εάν καταφέρει να υποκλέψει το κλειδί.

```
~/CA # openssl req -new -keyout private/cakey.pem -out careq.pem \
-config ./openssl.cnf
Generating a 2048 bit RSA private key
.........................................+++
...+++
writing new private key to 'private/cakey.pem'
Enter PEM pass phrase: admin
Verifying - Enter PEM pass phrase: admin
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GR]:
State or Province Name (full name) [Athens]:
Locality Name (eg, city) [Pireus]:
Organization Name (eg, company) [UNIPI]:
Organizational Unit Name (eg, section) [TED]:
Common Name (eg, YOUR name) []:CA
Email Address []:you@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:admin
An optional company name []:
```

Έπειτα "αυτό-υπογράφουμε" το πιστοποιητικό για να το μετατρέψουμε σε CA. Ένα δείγμα του τι θα πάρουμε σαν έξοδο είναι το ακόλουθο:

```
   ~/CA # openssl ca -create_serial -out cacert.pem -keyfile private/cakey.pem -
selfsign -extensions v3_ca -config ./openssl.cnf -in careq.pem
   Using configuration from ./openssl.cnf
   Enter pass phrase for private/cakey.pem: admin
   Check that the request matches the signature
   Signature ok
   Certificate Details:
       Serial Number:
         f2:c8:4a:d0:f5:09:28:b7
       Validity
         Not Before: Oct 24 03:17:49 2007 GMT
         Not After : Oct 23 03:17:49 2008 GMT
       Subject:
         countryName        = GR
         stateOrProvinceName    = Athens
         organizationName     = UNIPI
         organizationalUnitName  = TED
         commonName        = CA
         emailAddress       = you@example.com
       X509v3 extensions:
         X509v3 Subject Key Identifier:
           D0:1E:BF:7B:A8:26:B9:98:B0:81:98:2E:E7:96:CA:57:3D:76:F3:02
         X509v3 Authority Key Identifier:
           keyid:D0:1E:BF:7B:A8:26:B9:98:B0:81:98:2E:E7:96:CA:57:3D:76:F3:02
           DirName:/C=US/ST=The Great State You Live In/O ...
           serial:F2:C8:4A:D0:F5:09:28:B7

         X509v3 Basic Constraints:
           CA:TRUE
   Certificate is to be certified until Oct 23 03:17:49 2008 GMT (365 days)
   Sign the certificate? [y/n]:y
   1 out of 1 certificate requests certified, commit? [y/n]y
   Write out database with 1 new entries
   Data Base Updated
```

Με την παραπάνω εντολή, "-create_serial" δημιουργούμε ένα δεκαεξαδικό σειριακό αριθμό γι αυτό το κλειδί. Η εντολή "-extensions" προσδιορίζει το τμήμα στο αρχείο openssl.cnf ώστε να κοιτάξει για συγκεκριμένες επεκτάσεις που θα επισυναφθούν στο νέο πιστοποιητικό (Δημόσιο Κλειδί). Σε αυτή την περίπτωση χρησιμοποιούμε το τμήμα v3_ca,που μεταξύ άλλων περιέχει τη ρύθμιση στη γραμμή 234 που επιτρέπει στο κλειδί να χρησιμοποιείται ώστε να υπογράφει άλλα κλειδιά, δρώντας ως CA:

```
basicConstraints = CA:true
```

Ένα τελευταίο βήμα είναι η δημιουργία αντιγράφου του πιστοποιητικού CA κωδικοποιημένο στη μορφή DER, διότι τα Windows λειτουργούν μονάχα με δυαδικά κωδικοποιημένα πιστοποιητικά. Η εντολή είναι η ακόλουθη:

```
~/CA $ openssl x509 -inform PEM -outform DER -in cacert.pem -out cacert.der
```

4.1.2 Δημιουργία των κλειδιών Client και Server

Τώρα που η CA είναι ρυθμισμένη, θα πρέπει να κατασκευαστούν ζεύγη κλειδιών για το server και για όλους τους πελάτες clients. Αρχικά θα δημιουργηθεί ένα νέο ζεύγος κλειδιών για το server:

```
~/CA # openssl req -new -config ./openssl.cnf -keyout server_key.pem \
-out server_req.pem
Generating a 2048 bit RSA private key
.......+++
.................................+++
writing new private key to 'server_key.pem'
Enter PEM pass phrase: admin
Verifying - Enter PEM pass phrase: admin
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

```
-----
Country Name (2 letter code) [GR]:
State or Province Name (full name) [Athens]:
Locality Name (eg, city) [Pireus]:
Organization Name (eg, company) [UNIPI]:
Organizational Unit Name (eg, section) [TED]:
Common Name (eg, YOUR name) []: server
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:admin
An optional company name []:
```

Έπειτα θα υπογράψουμε το κλειδί με την CA που δημιουργήθηκε προηγουμένως και το αποτέλεσμα θα είναι της μορφής:

```
~/CA # openssl ca -config ./openssl.cnf -in server_req.pem -out server_cert.pem
Using configuration from ./openssl.cnf
Enter pass phrase for /home/brandon/CA/private/cakey.pem: admin
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number:
      f2:c8:4a:d0:f5:09:28:b8
    Validity
      Not Before: Nov 1 02:32:07 2007 GMT
      Not After : Oct 31 02:32:07 2008 GMT
    Subject:
      countryName        = GR
      stateOrProvinceName    = Athens
      organizationName     = TED
      organizationalUnitName  = UNIPI
      commonName        = server
    X509v3 extensions:
      X509v3 Basic Constraints:
        CA:FALSE
```

```
        Netscape Comment:
          OpenSSL Generated Certificate
        X509v3 Subject Key Identifier:
          71:A0:FB:1C:35:B7:B8:1D:1C:A4:C6:DF:A5:BA:80:6E:89:09:B7:CE
        X509v3 Authority Key Identifier:
          keyid:D0:1E:BF:7B:A8:26:B9:98:B0:81:98:2E:E7:96:CA:57:3D:76:F3:02

Certificate is to be certified until Oct 31 02:32:07 2008 GMT (365 days)
Sign the certificate? [y/n]: y


1 out of 1 certificate requests certified, commit? [y/n] y
Write out database with 1 new entries
Data Base Updated
```

Υποσημείωση: Εάν χρησιμοποιηθεί η εφαρμογή των Windows για τη διαχείριση της ασύρματης σύνδεσης στους πελάτες θα πρέπει να χρησιμοποιηθούν οι επεκτάσεις X509v3 που προστέθηκαν προηγουμένως και επομένως η προηγούμενη εντολή θα έχει την ακόλουθη μορφή:

```
~/CA # openssl ca -config ./openssl.cnf -extensions xpserver_ext \
 -in server_req.pem -out server_cert.pem
```

Έπειτα θα δημιουργηθούν τα ζεύγη κλειδιών για τους πελάτες χρησιμοποιώντας την ακόλουθη εντολή και απλώς αλλάζουμε το όνομα του αρχείου του κλειδιού σε μια τιμή που να ταιριάζει στην εφαρμογή μας:

```
~/CA $ openssl req -new -config ./openssl.cnf -keyout linux_laptop_key.pem \
-out linux_laptop_req.pem
...
Common Name (eg, YOUR name) []: linux_laptop
```

και:

```
~/CA $ openssl req -new -config ./openssl.cnf -keyout winxp_laptop_key.pem \
-out winxp_laptop_req.pem
...
Common Name (eg, YOUR name) []: winxp_laptop
```

Η υπογραφή των κλειδιών θα γίνει με τον ίδιο τρόπο που έγινε και η υπογραφή του πιστοποιητικού του server. Σε περίπτωση που ο πελάτης τρέχει σε λειτουργικό περιβάλλον Linux η εντολή είναι η ακόλουθη:

```
~/CA $ openssl ca -config ./openssl.cnf -in linux_laptop_req.pem \
 -out linux_laptop_cert.pem
```

Για περιβάλλον Windows πρέπει αν χρησιμοποιηθούν οι επεκτάσεις X509v3 αν το λειτουργικό διαχειρίζεται τους ασύρματους χρήστες:

```
~/CA $ openssl ca -config ./openssl.cnf -extensions xpclient_ext \
 -in winxp_laptop_req.pem -out winxp_laptop_cert.pem
```

Το λειτουργικό Windows έχει άλλη μία ιδιαιτερότητα, για να επεξεργαστεί τα κλειδιά πρέπει να "πακεταριστεί" το πιστοποιητικό του πελάτη ανάλογα με το ιδιωτικό κλειδί σε ένα αρχείο της μορφής PKCS#12. Αυτό επιτυγχάνεται με τη ακόλουθη εντολή:

```
~/CA $ openssl pkcs12 -export -clcerts -in winxp_laptop_cert.pem \
-inkey winxp_laptop_key.pem -out winxp_laptop.p12
Enter pass phrase for winxp_laptop_key.pem:
Enter Export Password: admin
Verifying - Enter Export Password: admin
```

Η παραπάνω εντολή χρησιμοποιεί την εφαρμογή pkcs12 του OpenSSL's για να εξάγει ("-export") ένα νέο αρχείο PKCS#12. Η εντολή "-clcerts" προσδιορίζει στο OpenSSL να εξάγει μόνο το πιστοποιητικό του πελάτη και το ιδιωτικό κλειδί. Σε άλλες περιπτώσεις είναι δυνατόν να γίνει εξαγωγή πολλαπλών πιστοποιητικών και κλειδιών σε ένα μόνο αρχείο PKCS#12.

Η δημιουργία καλώ κλειδιών βασίζεται στην ύπαρξη καλού σετ από "τυχαία" δεδομένα για να βοηθήσει στη δημιουργία των κλειδιών. Παρ' όλο που δε συσχετίζεται άμεσα με τη δημιουργία των κλειδιών PKI, θα χρειαστούν αυτά τα δεδομένα αργότερα για το FreeRADIUS.M ε την χρήση του OpenSSL θα δημιουργήσουμε τις παραμέτρους Diffie-Hellman για τη συμμετρική δημιουργία κλειδιών.

Αρχικά θα πρέπει να αποκτήσουμε δικαιώματα υπερχρήστη (superuser) και αν δημιουργήσουμε το φάκελο όπου θα τοποθετηθούν το πιστοποιητικό της CA, τα δημόσια και ιδιωτικά κλειδιά του

server, aένα αρχείο dh για τις παραμέτρους Diffie-Hellman και ένα αρχείο τυχαίας ημερομηνίας. Στη περίπτωσή μας τοποθετούνται στο φάκελο /etc/wireless, αν και μπορούν να τοποθετηθούν οπουδήποτε μπορεί να διαβάσει αρχεία το πρόγραμμα του FreeRADIUS.

```
~/CA $ su
Password: admin
/root/CA # mkdir /etc/wireless
```

Η αντιγραφή του δημόσιου και ιδιωτικού κλειδιού του server και το πιστοποιητικό της CA στο φάκελο /etc/wireless γίνεται με την ακόλουθη εντολή:

```
/root/CA # cp cacert.pem server_cert.pem server_key.pem /etc/wireless/
```

Έπειτα δημιουργούμε τις παραμέτρους 1024-bit Diffie-Hellman με την ακόλουθη εντολή:

```
/etc/wireless # openssl dhparam -out dh 1024
```

Και τέλος δημιουργούμε ένα τυχαίο αρχείο που θα βοηθήσει στη δημιουργία των κλειδιών:

```
/etc/wireless # dd if=/dev/urandom of=random count=2
```

4.2 Παραμετροποίηση του FreeRADIUS

Ο FreeRADIUS όπως έχουμε παρατηρήσει και σε προηγούμενες ασκήσεις αποτελείται από εκτενή αρχεία ρυθμίσεων. Η ρύθμιση αυθεντικοποίησης WPA2 είναι μονάχα ένα πολύ μικρό δείγμα των δυνατοτήτων του. Επειδή οι αρχικές ρυθμίσεις είναι παρόμοιες με τις ρυθμίσεις που θα έχουμε, οι αλλαγές θα γίνουν σε αυτές είναι λίγες .

Αρχικά ανοίγουμε το αρχείο ρυθμίσεων radiusd.conf σε έναν επεξεργαστή κειμένου και ρυθμίζουμε τους φακέλους που θα βλέπει ο server μας.

```
23 prefix = /usr/local
24 exec_prefix = ${prefix}
25 sysconfdir = ${prefix}/etc
26 localstatedir = ${prefix}/var
27 sbindir = ${exec_prefix}/sbin
28 logdir = ${localstatedir}/log/radius
29 raddbdir = ${sysconfdir}/raddb
```

```
30 radacctdir = ${logdir}/radacct
31
32 # Location of config and logfiles.
33 confdir = ${raddbdir}
34 run_dir = ${localstatedir}/run/radiusd
35
36 #
37 # The logging messages for the server are appended to the
38 # tail of this file.
39 #
40 log_file = ${logdir}/radius.log
```

Επίσης στο τέλος του αρχείου έχει τις ακόλουθες ρυθμίσεις

```
1757 instantiate {
1758 exec
1759 expr
1760 }
1761 authorize {
1762 eap
1763 files
1764 }
1765 authenticate {
1766 unix
1767 eap
1768 }
1769 preacct {
1770 acct_unique
1771 files
1772 }
1773 accounting {
1774 detail
1775 unix
1776 radutmp
1777 }
1778 session {
1779 radutmp
```

```
1780 }
1781 post-auth {
1782 }
1783 pre-proxy {
1784 }
1785 post-proxy {
1786 eap
1787 }
```

Έπειτα θα ανοίξουμε το αρχείο ρυθμίσεων, clients.conf και θα προσθέσουμε τον router. Ο router είναι ο μόνος πραγματικός πελάτης του RADIUS server. Θα χρησιμοποιήσουμε την διεύθυνση IP του router και ένα ισχυρό μυστικό το οποίο θα είναι και ο "κωδικός" που θα χρησιμοποιεί ο router για τη συνομιλία του με το RADIUS server.

Η μεταβλητή "shortname" χρησιμοποιείται για καταγραφή στο αρχείο log, και επομένως μπορεί να έχει οποιαδήποτε τιμή. Εάν ο τύπος NAS (Network Access Server) δεν υπάρχει στη λίστα του αρχείου clients.conf, χρησιμοποιούμε το "other" για τύπο NAS. Στη περίπτωσή μας το Linksys WRT54GL δεν ανήκει.

```
client 192.168.2.2 {
#    # secret and password are mapped through the "secrets" file.
     secret   = testing123
     shortname  = wirless_ap
#     # the following three fields are optional, but may be used by
#     # checkrad.pl for simultaneous usage checks
     nastype   = other
}
```

Έπειτα, αλλάζουμε το αρχείο users. Προσθέτουμε τις ακόλουθες γραμμές για κάθε κλειδί πελάτη που δημιουργήσαμε χρησιμοποιώντας ως κοινό όνομα αυτό που παρέχεται με το user name του κλειδιού. Για μήνυμα απόρριψης μπορούμε να προσθέσουμε ότι θέλουμε.

```
# users file for FreeRADIUS

winxp_laptop  Auth-type := EAP
```

```
linux_laptop  Auth-type := EAP

DEFAULT     Auth-type := Reject
        Reply-Message := "Your Computer Isn't Welcome Here!"
```

Το επόμενο αρχείο ρυθμίσεων που χρειάζεται αλλαγή είναι το eap.conf. Αλλάζουμε το default_eap_type σε TLS στη γραμμή 23:

```
default_eap_type = tls
```

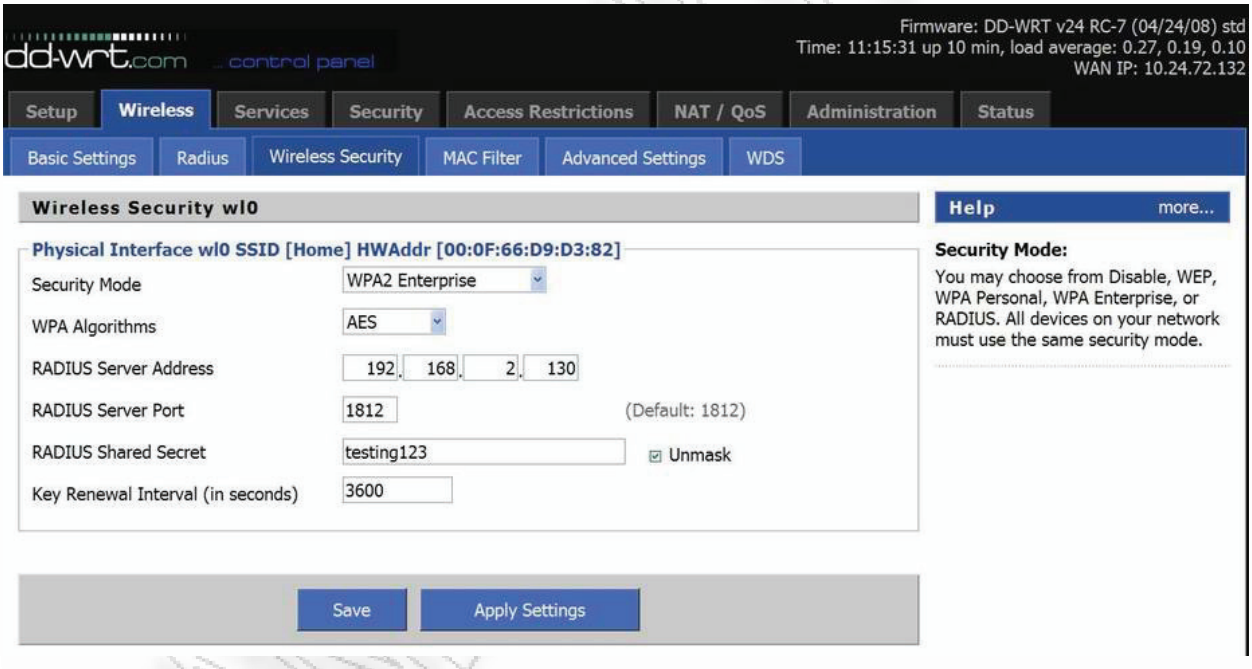
Έπειτα στη συνέχεια προσαρμόζουμε τις ρυθμίσεις TLS ώστε να ανταποκρίνονται με αυτές που κάναμε στα προηγούμενα βήματα:

```
123  tls {
124    private_key_password = admin
125    private_key_file = /etc/wireless/server_key.pem
126
127    # If Private key & Certificate are located in
128    # the same file, then private_key_file &
129    # certificate_file must contain the same file
130    # name.
131    certificate_file = /etc/wireless/server_cert.pem
132
133    # Trusted Root CA list
134    CA_file = /etc/wireless/cacert.pem
135
136
137    #
138    # For DH cipher suites to work, you have to
139    # run OpenSSL to create the DH file first:
140    #
141    #    openssl dhparam -out certs/dh 1024
142    #
143    dh_file = /etc/wireless/dh
144    random_file = /etc/wireless/random
```

Β΄ Μέρος Παραμετροποίηση εξοπλισμού NAS και χρηστών

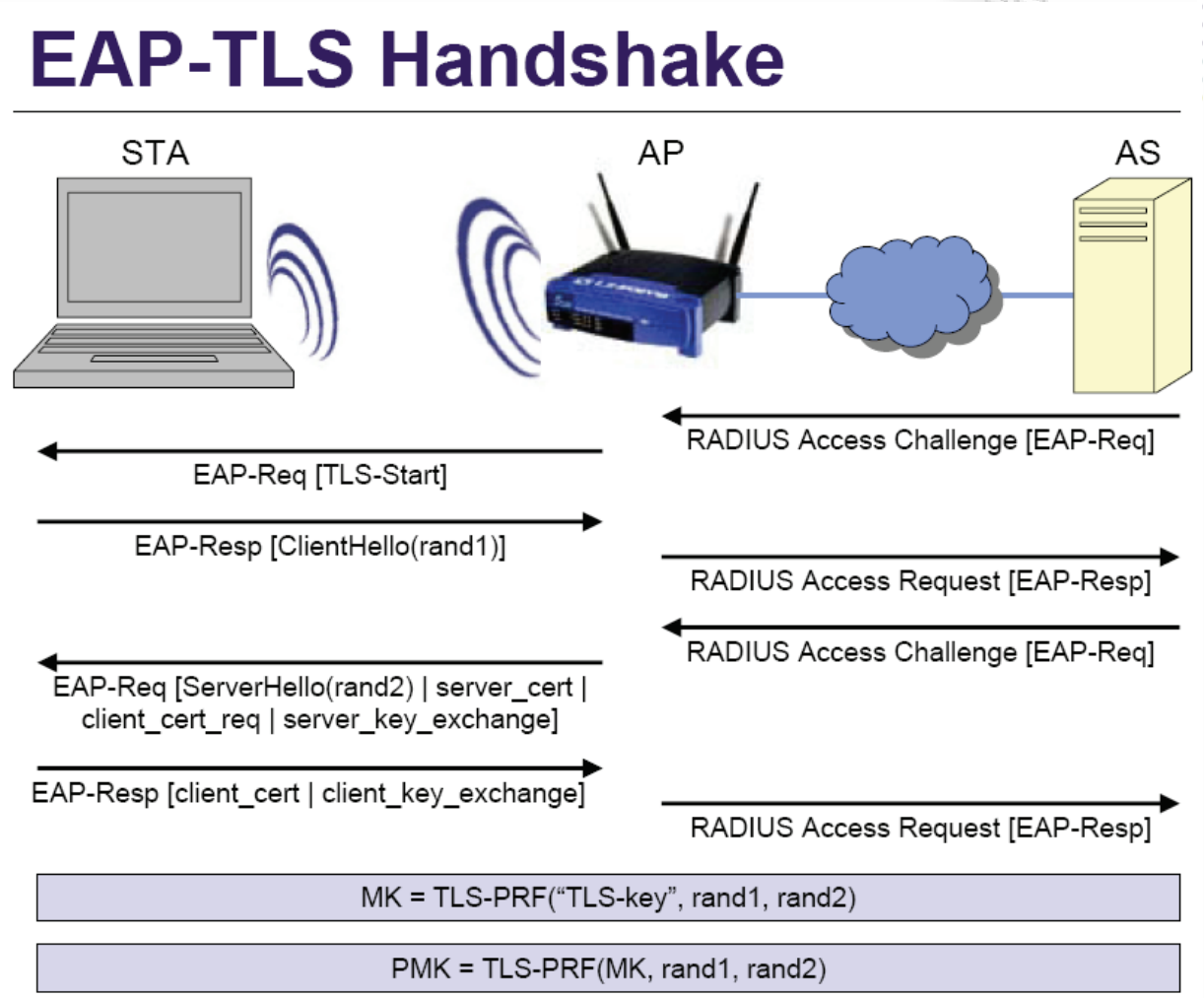
4.3 Παραμετροποίηση του router

Για τη παρούσα άσκηση χρησιμοποιήσαμε το Linksys WRT54GL με εναλλακτικό firmware DD-WRT v24 RC7 και γι αυτό οι σελίδες ρυθμίσεων θα διαφέρουν. Αρχικά αλλάζουμε στις ρυθμίσεις ασφαλείας του ασύρματου δικτύου σε WPA-Enterprise ή WPA2-Enterprise, και προσθέτουμε τη διεύθυνση IP του RADIUS serverκαι το διαμοιραζόμενο μυστικό όπως δείχνουν οι παρακάτω εικόνες.



Εικόνα 6 Επιλογή WPA2 Enterprise Mode

Μια είκόνα οπου φάινεται αναλυτικά η διαδικασία ανταλλαγής των κλειδιών είναι η ακολουθη:



Εικόνα 7 Διαδικασία EAP- TLS

4.4 Παραμετροποίηση του πελάτη Linux

Για τη σύνδεση ενός πελάτη με λειτουργικό Linux με τη χρήση της μεθόδου ασφαλείας WPA ή WPA2 απαιτείται να είναι εγκατεστημένο το πρόγραμμα wpa_supplicant. Θα πρέπει να ρυθμιστεί με τις ακόλουθες επιλογές στο αρχείο ".config" για τους οδηγούς και τις διεπαφές που θα χρησιμοποιηθούν:

```
CONFIG_IEEE8021X_EAPOL=y
CONFIG_EAP_TLS=y
CONFIG_PKCS12=y
```

```
#Make sure to include any other options you need as well
```

Θα πρέπει να ανασυντάξουμε και να επανεγκαταστήσουμε το wpa_supplicant. Έπειτα θα πρέπει να δημιουργηθεί ένας φάκελος στο πελάτη Linux όπου θα αποθηκεύσουμε τα δημόσια και ιδιωτικά κλειδιά (αρχειο PKCS#12) και το πιστοποιητικό CA. Στη περίπτωσή μας, αυτά τα αποθηκεύουμε στο αρχείο /etc/wireless.

Έπειτα διορθώνουμε το αρχείο "wpa_supplicant.conf" και προσθέτουμε ένα τμήμα παρόμοιο με το ακόλουθο, το οποίο προσδιορίζει τις νέες ρυθμίσεις WPA2-Enterprise.

```
# WPA2-EAP/AES using EAP-TLS
network={
    ssid="Home_wpa"
    key_mgmt=WPA-EAP
    eap=TLS
    identity="linux_laptop"
    ca_cert="/etc/wireless/cacert.pem"
    private_key="/etc/wireless/linux_laptop.p12"
    private_key_passwd="admin"
}
```

Το πεδίο "identity" πρέπει να ταιριάζει με το κοινό όνομα στο πιστοποιητικό του πελάτη και το χρήστη που προσδιορίσαμε στο αρχείο FreeRADIUS users. Τέλος επανεκκινούμε το wpa_supplicant και συνδεόμαστε στο δίκτυο.

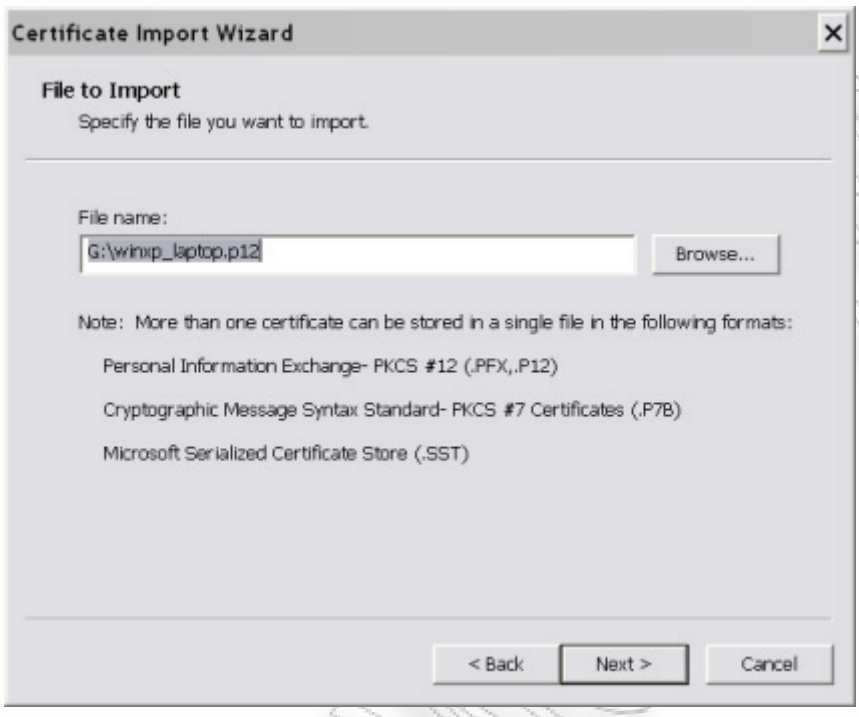
4.5.1 Ρύθμιση πελάτη Windows XP (εγκατάσταση πιστοποιητικών)

Για τους χρήστες Windows απαιτείται να είναι εγκατεστημένο το WPA2 patch, το οποίο μπορεί να βρεθεί από το δικτυακό τόπο της Microsoft. Μετά την εγκατάσταση του patch, μεταφέρουμε το πιστοποιητικό CA και το αρχείο p12 από το server που το δημιουργήσαμε.

Αρχικά, εγκαθιστούμε το πιστοποιητικό CA ως έμπιστη οντότητα κάνοντας διπλό κλικ πάνω του.

Εικόνα 8 Εγκατάσταση της CA

Κάντε κλικ στην επιλογή "Install Certificate" και ολοκληρώστε τον οδηγό. Έπειτα, κάντε διπλό κλικ στο αρχείο p12 που περιέχει το πιστοποιητικό του πελάτη και το κλειδί του και εγκαταστήστε το.

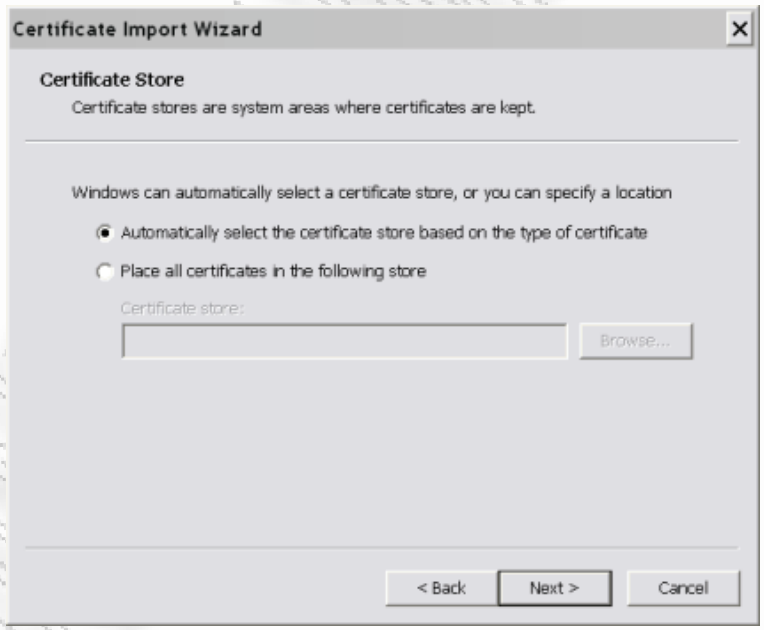


Εικόνα 9 Εγκατάσταση των κλειδιών πελατών

Εισάγετε το κωδικό πρόσβασης για το προσωπικό κλειδί του πελάτη. Εδώ παρέχεται η επιλογή ο κωδικός πρόσβασης να εισάγεται κάθε φορά που χρησιμοποιείται το κλειδί, αλλά στην περίπτωσή μας δεν είναι αναγκαίο βήμα.

Εικόνα 10 Κωδικός Πρόσβασης Πελάτη

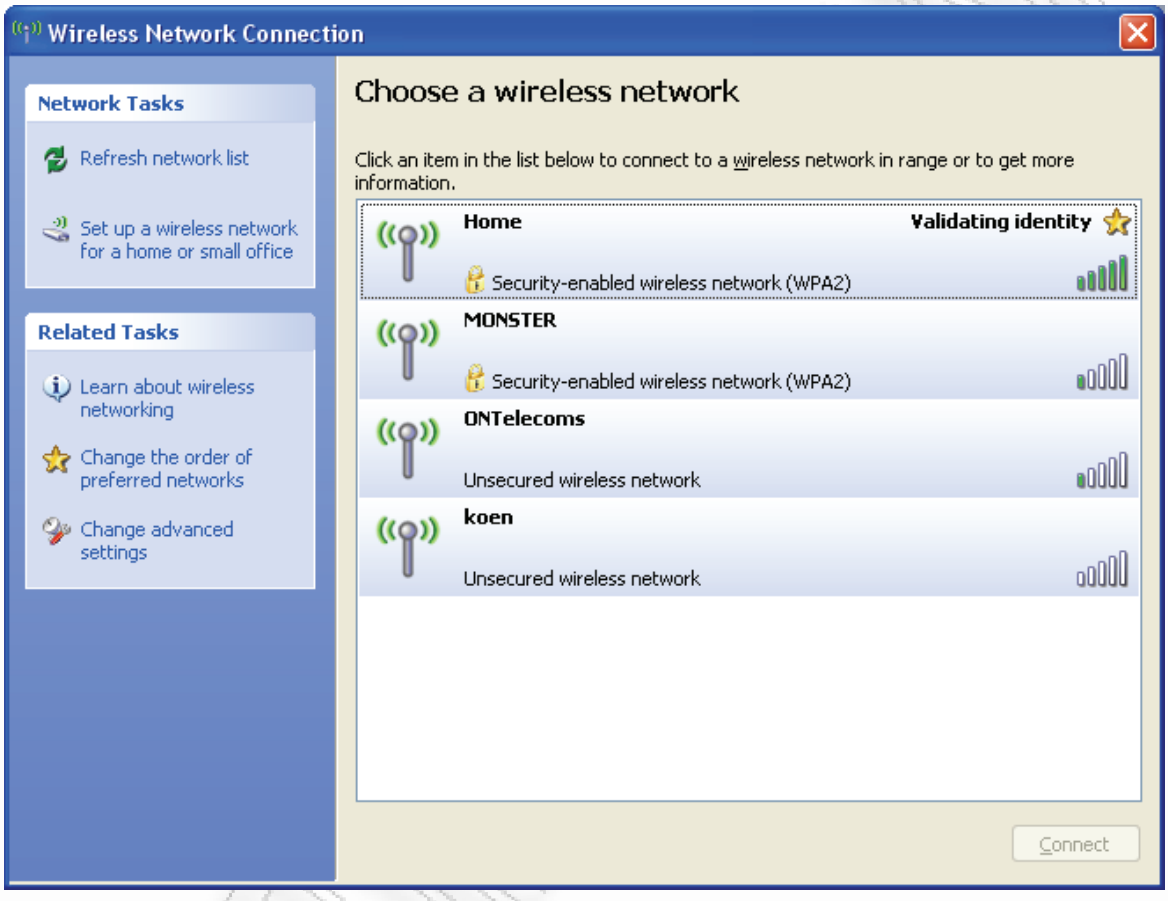
Έπειτα αφήνουμε το λειτουργικό να αποφασίσει πού θα αποθηκεύσει το πιστοποιητικό.



Εικόνα 11 Αποθήκευση των κλειδιών πελάτη

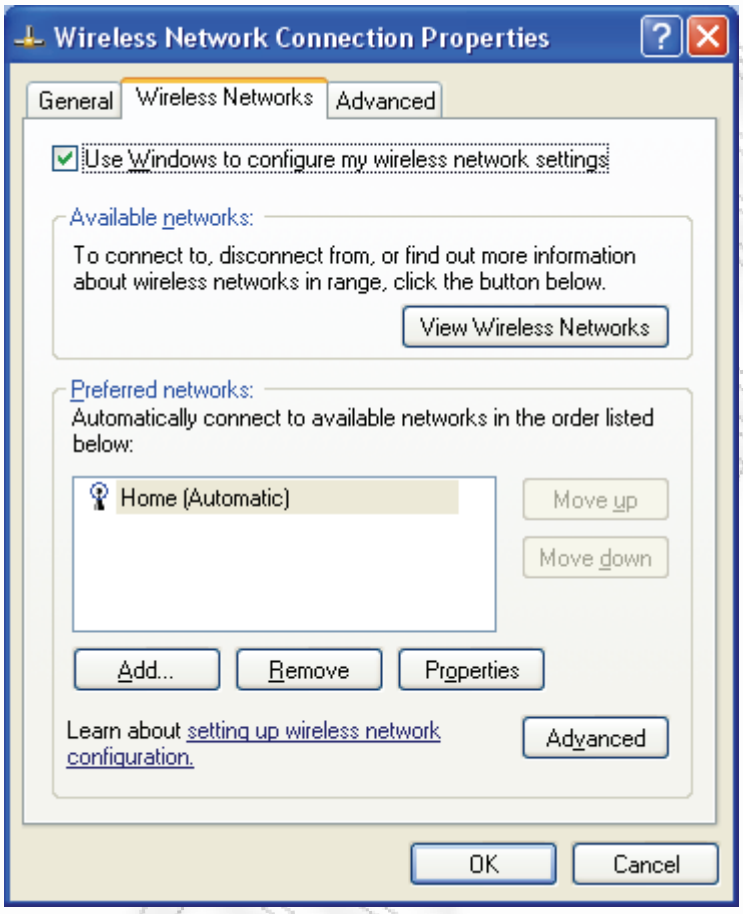
4.5.2 Ρύθμιση πελάτη Windows XP (ρύθμιση σύνδεσης)

Αφού εγκαταστήσαμε τα πιστοποιητικά, ανοίγουμε την αναζήτηση των ασύρματων δικτύων και επιλέγουμε το "Change Advanced Settings".



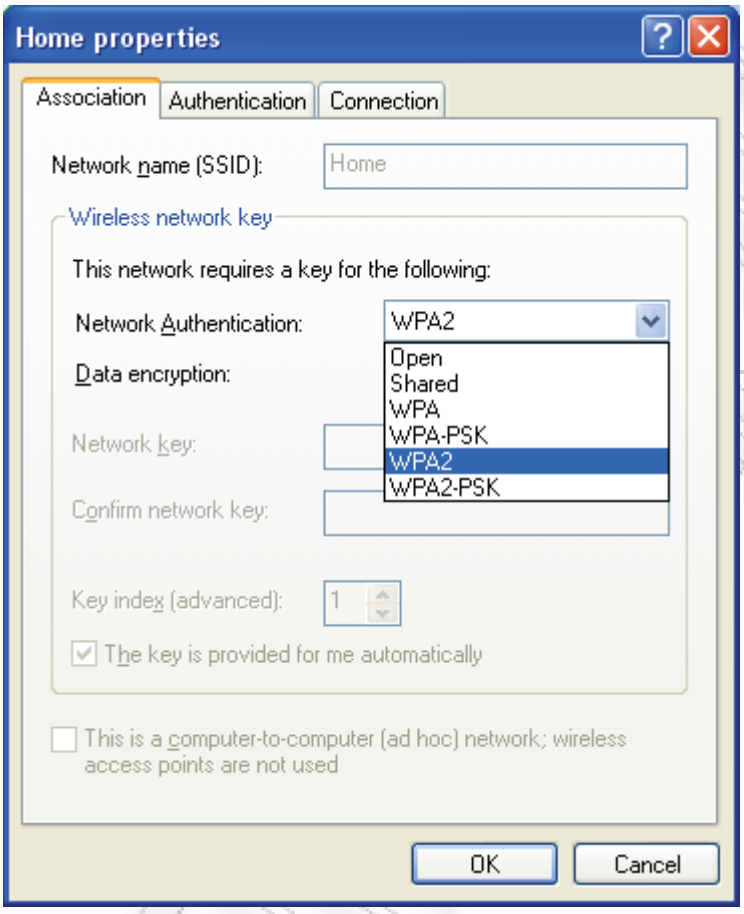
Εικόνα 12 Διαθέσιμα ασύρματα δίκτυα

Στην επιλογή "Wireless Networks", πατάμε "Add" στο Preferred Networks.



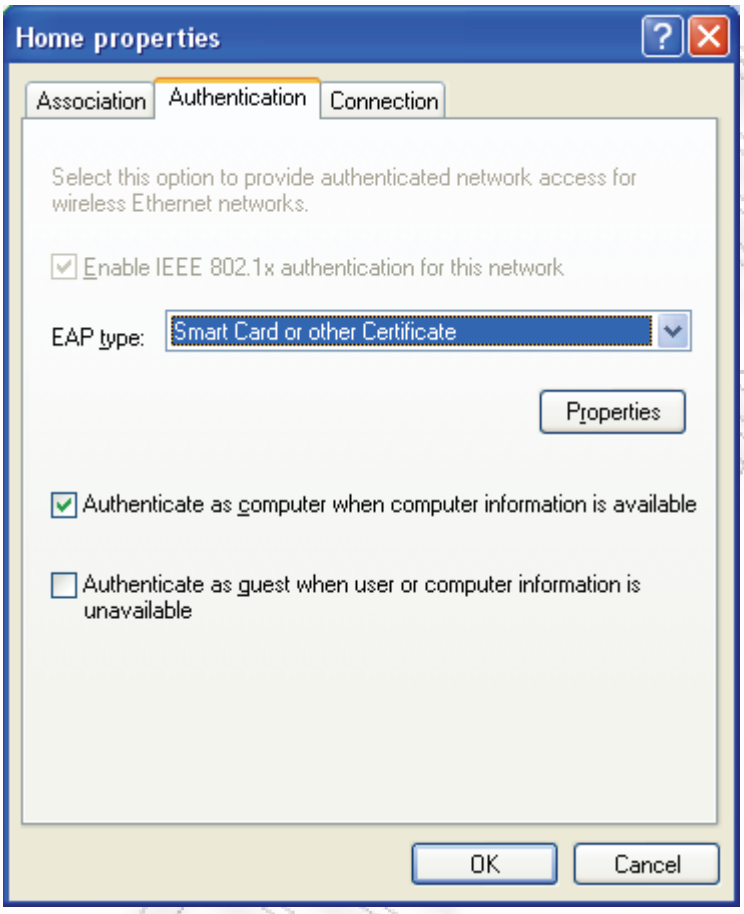
Εικόνα 13 Προχωρημένες Ρυθμίσεις

Εισάγουμε την SSID του router και αλλάζουμε την αυθεντικοποίηση του δικτύου σε WPA2.



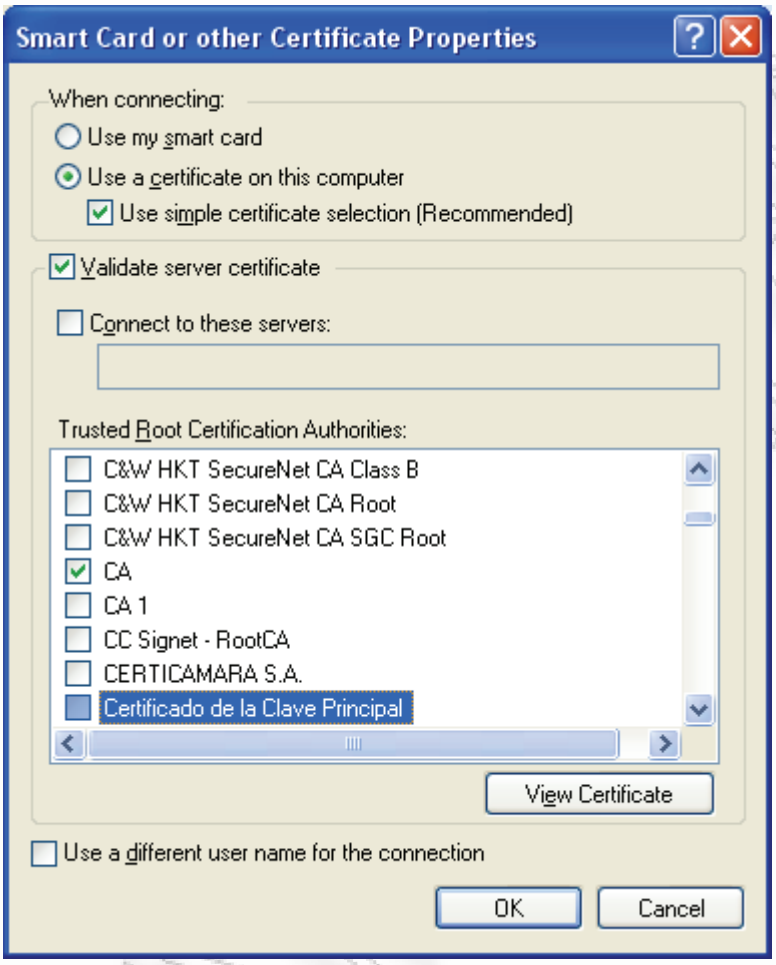
Εικόνα 14 Ρύθμιση WPA2

Στην επιλογή "Authentication", πατάμε Properties κάτω από το EAP Type.



Εικόνα 15 Ρύθμιση EAP

Επιλέγουμε τη CA από τη λίστα, και επιλέγουμε "Use a different username for this connection".



Εικόνα 16 Επιλογή πιστοποιητικού

Τέλος Πατάμε OK. Ανοίγουμε πάλι την επιλογή wireless networks και συνδεόμαστε στο νέο ασφαλές δίκτυο εάν όλες οι ρυθμίσεις έχουν γίνει σωστά.

4.6 Δημιουργία σύνδεσης και παρατήρηση του log του FreeRADIUS server

Τρέξτε σε debug mode το Radius server και πραγματοποιήστε μια νέα σύνδεση με τη μέθοδο που περιγράφηκε προηγουμένως και καταγράψτε το log του. Παρατηρήστε τα βήματα σύνδεσής και ανταλλαγής μηνυμάτων.

Έπειτα εγκαταστήστε ένα μη έγκυρο πιστοποιητικό και επαναλάβετε τη διαδικασία. Να γίνει καταγραφή του log και έπειτα να γίνει μια σύγκριση με το αρχικό. Σχολιάστε τις διαφορές στα δύο log

Κεφάλαιο 7 Case Study Hotspot

Στο πάρων κεφάλαιο, θα μελετήσουμε τη δημιουργία ενός Hotspot με χρήση του FreeRADIUS server που έχουμε ρυθμίσει στο κεφάλαιο 5 και του λογισμικού Microtik RouterOS.

7.1 Εισαγωγή

7.1.1 Τι είναι το Hotspot

Το Hotspot είναι ένα ασύρματο σημείο πρόσβασης στο Διαδίκτυο. Στην πραγματικότητα, δεν είναι απλώς ένα σημείο, αλλά μία περιοχή η οποία καλύπτεται από συσκευές που επιτρέπουν και διαχειρίζονται την ασύρματη πρόσβαση των χρηστών στο διαδίκτυο.

Ένα Hotspot μπορεί να έχει εμβέλεια από μερικά μέτρα και να φτάσει ακόμη και αρκετά χιλιόμετρα κάλυψης, αν αυτό είναι επιθυμητό.

7.1.2 Πρόσβαση στο Hotspot και χρήσεις του

Η πρόσβαση στο ασύρματο δίκτυο είναι δυνατή από ένα σύνολο συσκευών συμβατών με τα κατάλληλα πρωτόκολλα επικοινωνίας, όπως φορητοί υπολογιστές (laptops), έξυπνες συσκευές χειρός (handheld PDAs, τηλέφωνα κλπ), ασύρματες κάμερες και οθόνες τηλε-προβολής κ.α.

Ένας χρήστης, εκμεταλλευόμενος τις δυνατότητες που του παρέχει η ασύρματη σύνδεσή του με το hotspot, είναι σε θέση να πραγματοποιήσει στον υπολογιστή του οποιαδήποτε εργασία έχει σχέση με το Internet σαν να ήταν στο σπίτι του ή στο γραφείο του.

Αυτό σημαίνει ότι ο χρήστης του hotspot μπορεί να το χρησιμοποιήσει για τις ακόλουθες εργασίες:

- Πλοήγηση στο διαδίκτυο (Web surfing)

- Ανταλλαγή αρχείων και online επικοινωνία μεταξύ των χρηστών
- Πρόσβαση σε εφαρμογές πολυφασικού περιεχομένου (multimedia), για τη λήψη εικόνων, διαδραστικού βίντεο και μουσικής
- Λήψη ενημερωτικού ή εκπαιδευτικού περιεχομένου
- Η επιχείρηση που προσφέρει το ασύρματο δίκτυο στους επισκέπτες και τους πελάτες της, μπορεί να το χρησιμοποιήσει για να αναπτύξει μία σειρά υπηρεσιών. Οι υπηρεσίες αυτές μπορεί ενδεικτικά να περιλαμβάνουν:
- Προβολή μηνυμάτων διαφημιστικού περιεχομένου
- Υπηρεσίες ψυχαγωγίας
- Υπηρεσίες που βασίζονται στη γεωγραφική τοποθεσία του χρήστη
- Αυτόματη λήψη ή παροχή εξειδικευμένων δεδομένων και πληροφοριών, που σχετίζονται με τον καλυπτόμενο από το ασύρματο δίκτυο χώρο και μπορεί να αξιοποιηθούν τόσο από επισκέπτες, όσο και από τον πάροχο του ασύρματου δικτύου

7.2 Δημιουργία Hotspot

7.2.1 Παραμετροποίηση FreeRADIUS Server

Για την λειτουργία του Hotspot απαραίτητη προϋπόθεση είναι η προσθήκη των χρηστών στον RADIUS server καθώς και η δήλωση του δρομολογητή που θα λειτουργεί ως NAS και ως Access Point.

Αρχικά προσθέτουμε τη διεύθυνση του AP ώστε να έχει πρόσβαση στον RADIUS server. Για να το επιτύχουμε αυτό προσθέτουμε τις ακόλουθες γραμμές στο αρχείο clients.conf

```
client 192.168.2.129 {
     secret   = testing123
     shortname  = hotspot
```

```
        nastype    = other
    }
```

Επίσης για να υπάρχει δυνατότητα διαχείρισης του χρόνου πρόσβασης των πελατών πρέπει να ρυθμιστεί κατάλληλα και το αρχείο radiusd.conf ώστε να είναι δυνατή η χρήση του εργαλείου Rlm_sqlcounter που είναι προεγκατεστημένο. Αρχικά δημιουργούμε ένα αρχείο κειμένου με την ονομασία sqlcounter.conf στο ίδιο φάκελο πού είναι εγκατεστημένο και το αρχείο radiusd.conf, το οποιο έχει το ακόλουθο περιεχόμενο για τη Mysql:

```
    sqlcounter noresetcounter {
    counter-name = Max-All-Session-Time
    check-name = Max-All-Session
    sqlmod-inst = sql
    key = User-Name
    reset = never
    query = "SELECT SUM(AcctSessionTime) FROM radacct WHERE UserName='%{%k}'"
    }
    sqlcounter dailycounter {
    driver = "rlm_sqlcounter"
    counter-name = Daily-Session-Time
    check-name = Max-Daily-Session
    sqlmod-inst = sqlcca3
    key = User-Name
    reset = daily
    query      =      "SELECT      SUM(AcctSessionTime      -      GREATEST((%b      -
UNIX_TIMESTAMP(AcctStartTime)),   0))   FROM   radacct   WHERE   UserName='%{%k}'   AND
UNIX_TIMESTAMP(AcctStartTime) + AcctSessionTime > '%b'"
    }
    sqlcounter monthlycounter {
    counter-name = Monthly-Session-Time
    check-name = Max-Monthly-Session
    sqlmod-inst = sqlcca3
    key = User-Name
    reset = monthly
```

```
    query      =      "SELECT      SUM(AcctSessionTime      -      GREATEST((%b      -
UNIX_TIMESTAMP(AcctStartTime)),   0))   FROM   radacct   WHERE   UserName='%{%k}'   AND
UNIX_TIMESTAMP(AcctStartTime) + AcctSessionTime > '%b'"
    }
```

Έπειτα συμπεριλαμβάνουμε τον παραπάνω αρχείο στο radiusd.conf προσθέτοντας μία γραμμή στο τμήμα modules{ }

```
    modules {
    ...
    $INCLUDE ${confdir}/sqlcounter.conf
    ...
    }
```

Επίσης πρέπει να προστεθούν οι ονομασίες που περιέχονται στο sqlcounter στο τμήμα authorize όπως φαίνεται παρακάτω:

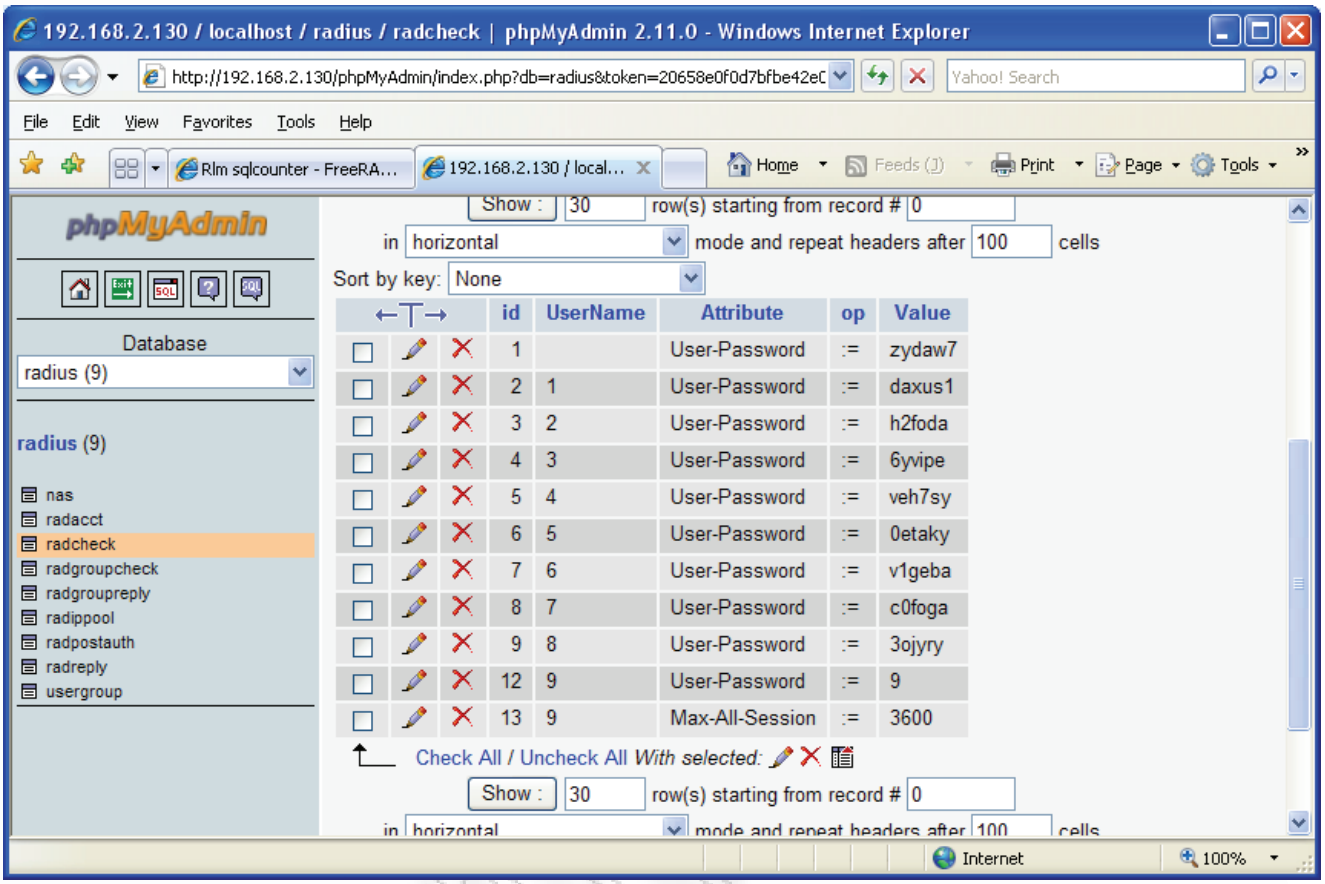
```
    authorize {
    ...
    noresetcounter
    dailycounter
    monthlycounter
    }
```

Μια απλή επεξήγηση των παραπάνω εντολών είναι η ακόλουθη:

- **noresetcounter :** Ο μετρητής δεν επανεκκινείται ποτέ, μπορεί να χρησιμοποιηθεί για συνόδους με πεπερασμένη διάρκεια (π.χ. για χρήση καρτών προπληρωμένου χρόνου)
- **dailycounter :** Ο μετρητής επανεκκινείται καθημερινά, μπορεί να χρησιμοποιηθεί για καθημερινό έλεγχο πρόσβασης (πχ. 3 ώρες ημερησίως)
- **monthlycounter:** Ο μετρητής επανεκκινείται μηνιαίως**,** μπορεί να χρησιμοποιηθεί για περιορισμό της μηνιαίας πρόσβασης (π.χ. 50 ώρες ανά μήνα)

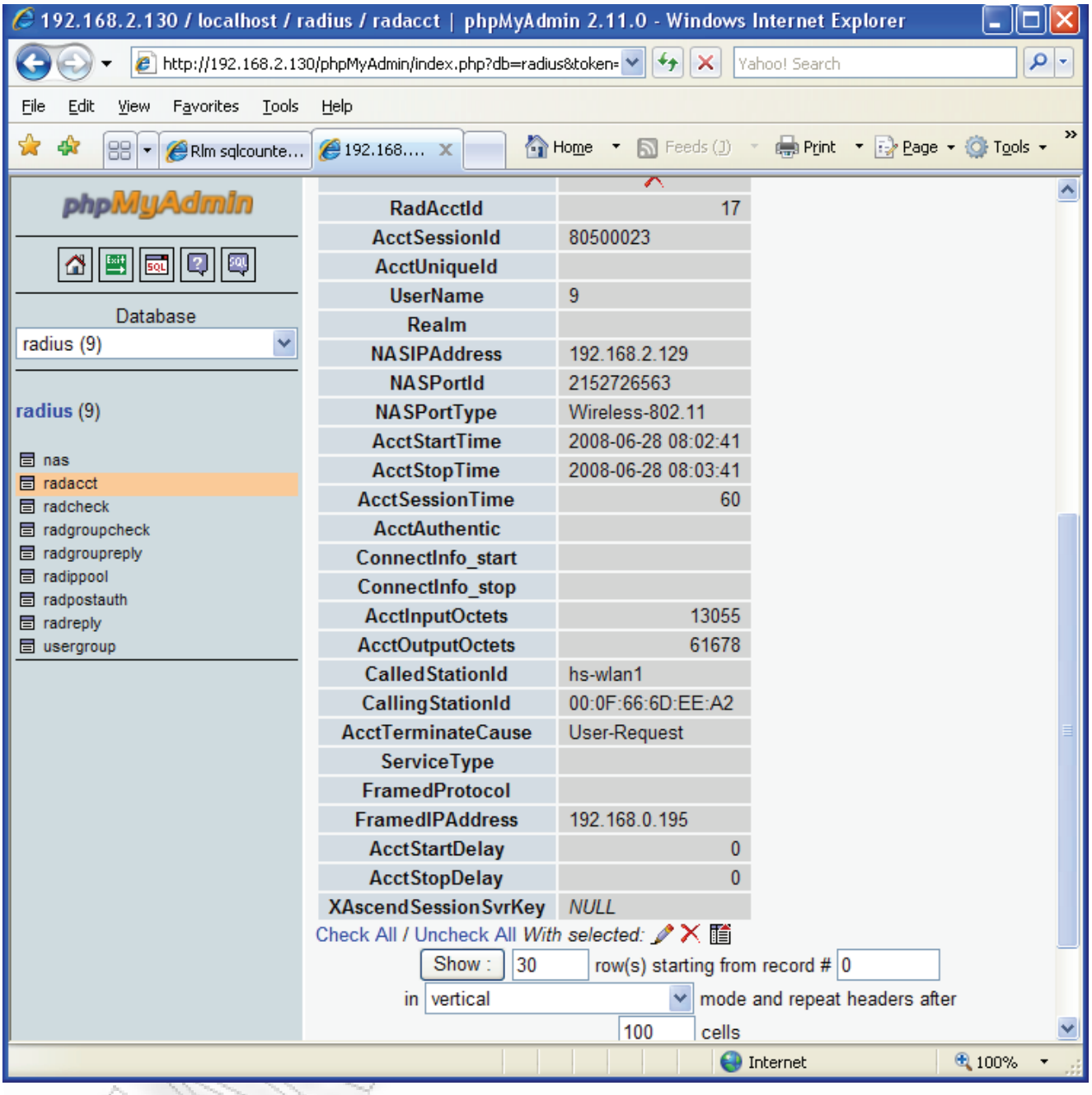
Θεωρούμε ότι ο RADIUS server έχει ρυθμιστεί κατάλληλα για να χρησιμοποιεί τη Mysql ως βάση δεδομένων των χρηστών του όπως έχει περιγραφεί εκτενώς στο κεφάλαιο 5. Για την προσθήκη χρηστών στη βάση δεδομένων θα χρησιμοποιηθεί το εργαλείο phpMyAdmin το οποίο παρέχει διαχείρισης της Mysql μέσω γραφικού περιβάλλοντος από το φυλλομετρητή (browser) του Linux server. Ένα παράδειγμα δημιουργίας χρήστη είναι το ακόλουθο:

Αρχικά προσθέτουμε έναν χρήστη με όνομα "9" και κωδικό πρόσβασης "9" που του παρέχεται πρόσβαση για μία ώρα και μετά το τέλος του χρόνου αυτού θα αποσυνδεθεί από το σύστημα. Για την υλοποίηση αυτού η προσθήκη του χρήστη θα γίνει στη καταχώρηση radcheck και θα εισαχθεί επιπλέον το πεδίο "Max-All-Session" που όπως προαναφέρθηκε προσφέρει σύνοδο με πεπερασμένη διάρκεια.



Εικόνα 17 Εισαγωγή χρήστη στη βάση δεδομένων

Επίσης ένα άλλο χρήσιμο πεδίο στη βάση δεδομένων είναι το radacct όπου καταγράφονται οι σύνοδοι που έχουν πραγματοποιηθεί και πληροφορίες γι αυτές όπως για παράδειγμα το NAS που πραγματοποίησε τη σύνδεση, διάρκεια συνόδου, χρόνος έναρξης και λήξης και πληροφορίες για το πελάτη που τη πραγματοποίησε όπως όνομα χρήστη και διεύθυνση MAC. Ένα δείγμα των παραπάνω είναι η ακόλουθη εικόνα:

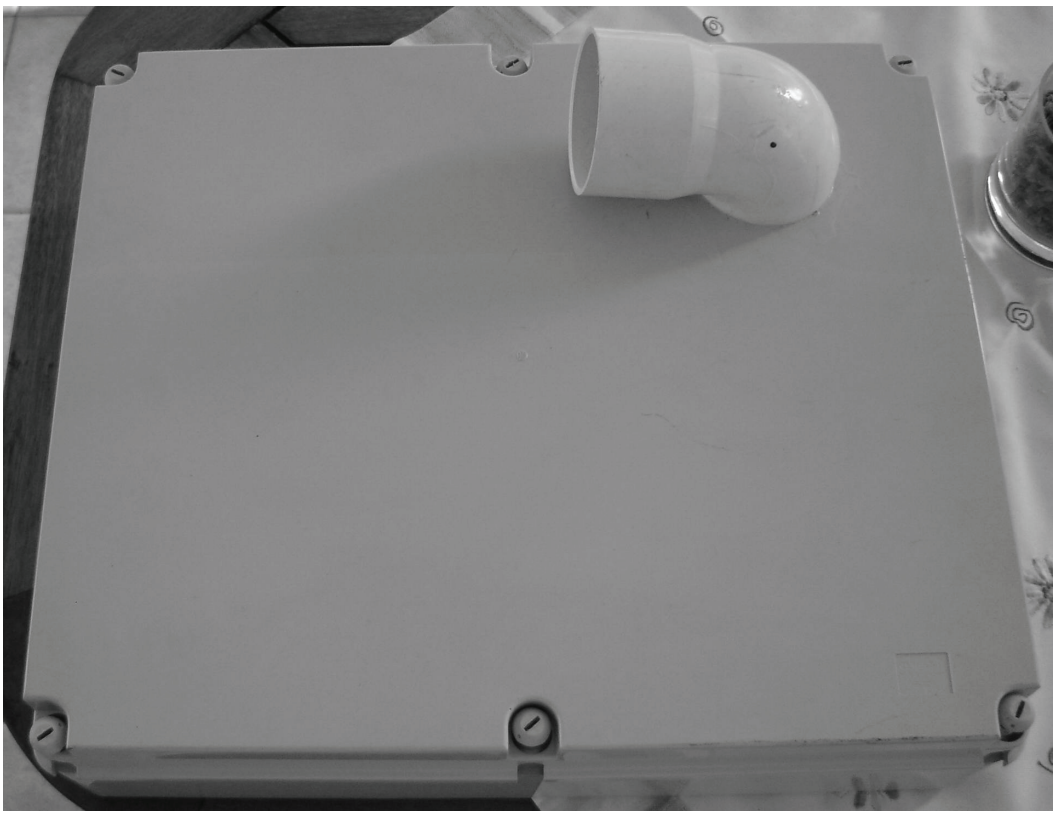


Εικόνα 18 Δείγμα στοιχείων καταχώρησης radacct

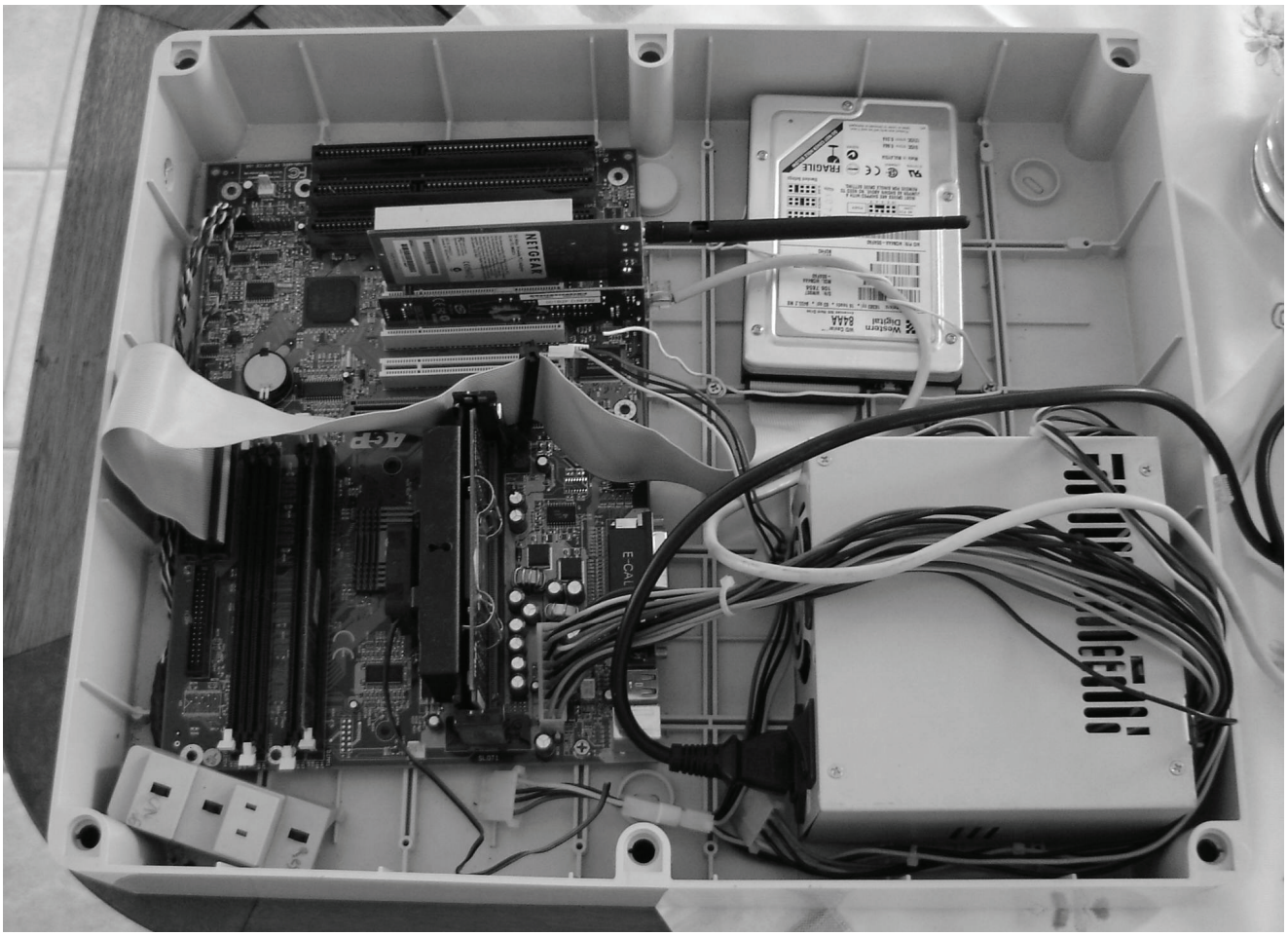
7.2.2 Παραμετροποίηση Εξοπλισμού Hotspot.

Το λειτουργικό Microtik RouterOS είναι μια εμπορική έκδοση λειτουργικού UNIX που κύριο στόχο έχει τη μετατροπή ενός υπολογιστή σε δρομολογητή με επιπλέον δυνατότητες όπως λειτουργία τείχους προστασίας (firewall), διαχείρισης εύρους ζώνης, λειτουργιά ασύρματου σημείου πρόσβασης (wireless access point), πύλη hotspot, VPN server και πολλές άλλες. Η εγκατάσταση του λειτουργικού μπορεί να γίνει είτε σε ειδικό εξοπλισμό (Routerboard) είτε σε υπολογιστή τεχνολογίας x86.

Στη περίπτωσή μας χρησιμοποιήθηκε ένας υπολογιστής τεχνολογίας Pentium II Celeron 333MHz με 64 MB RAM, σκληρό δίσκο 8 GB, κάρτα δικτύου Ethernet 10/100 Mbps Level One και ασύρματη κάρτα δικτύου 802.11b/g Netgear WG311 v1 που διαθέτει υποστηριζόμενο Atheros Chipset. Ακολουθούν μερικές φωτογραφίες της ιδιοκατασκευής που κατασκευάστηκε για το λόγο αυτό.



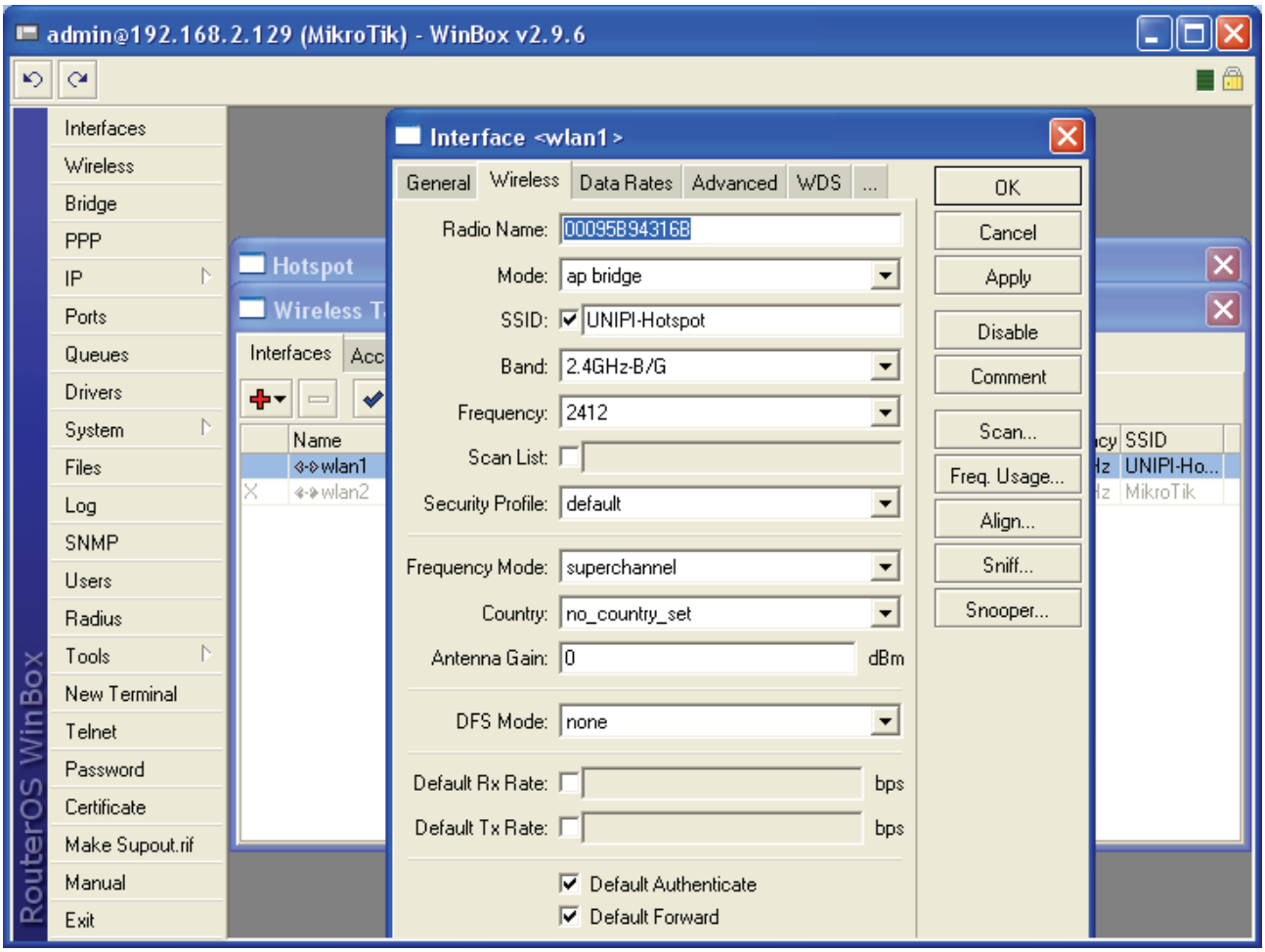
Εικόνα 19 Ιδιοκατασκευή AP Hotspot



Εικόνα 20 Εσωτερικό Ιδιοκατασκευής

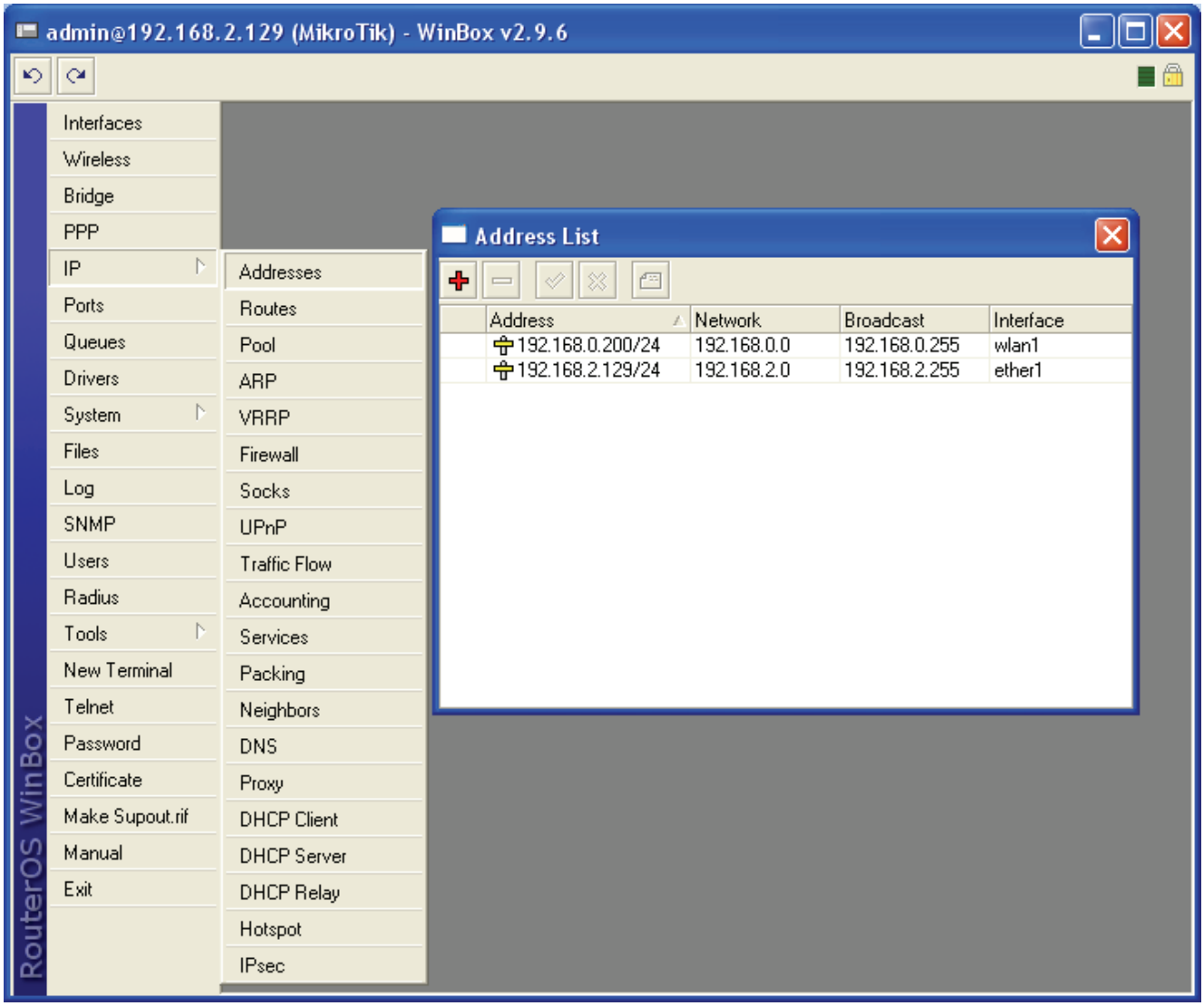
7.2.2.1 Αρχική παραμετροποίηση Microtik RouterOS

Αρχικά ρυθμίζουμε το AP Interface. Δίνουμε το επιθυμητό SSID και διαλέγουμε το κανάλι που μας βολεύει. Όπως φαίνεται και στην παρακάτω εικόνα σαν όνομα δικτύου (SSID) έχουμε δώσει το UNIPI-HOTSPOT, καθώς και ο τρόπος λειτουργίας της ασύρματης κάρτας (mode) είναι το AP bridge. Επίσης ορίζουμε και το κανάλι λειτουργίας του Hotspot μας που στη προκειμένη περίπτωση είναι το κανάλι 1 (2412 MHz)



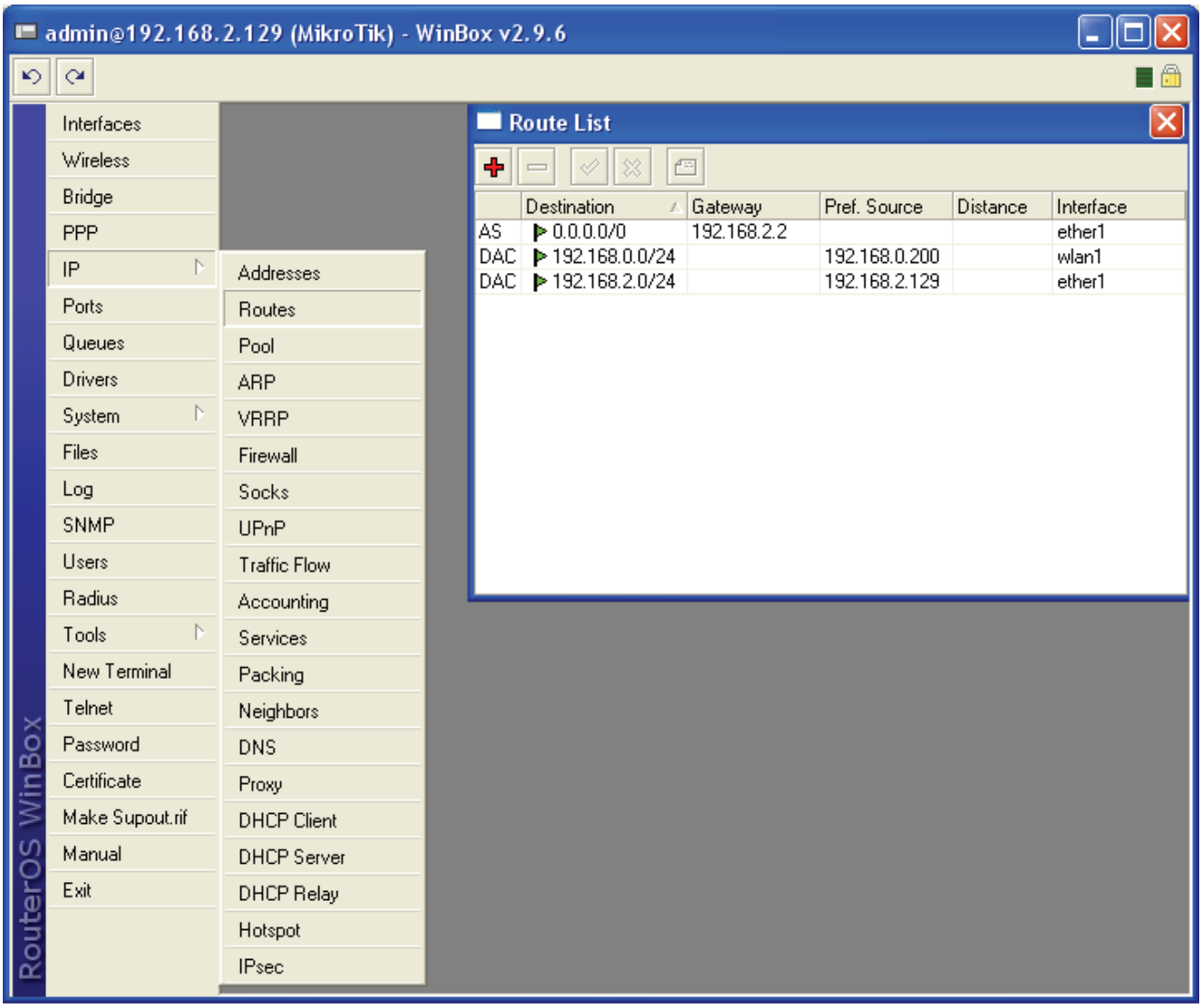
Εικόνα 21 Ρύθμιση AP Interface

Έπειτα πρέπει να ορίσουμε τις διευθύνσεις IP για την ενσύρματη και ασύρματη διεπαφή ether1και wlan1 αντίστοιχα. Αυτό ορίζεται όπως φαίνεται και στην παρακάτω εικόνα στη καρτέλα IP -> Addresses. Στη περίπτωσή μας η ενσύρματη διεπαφή έχει διεύθυνση 192.168.2.129/24 και η ασύρματη 192.168.0.200/24.Υπο άλλες συνθήκες μπορεί να καθοριστεί ένα μικρότερο subnet για το σύνολο των ασύρματων χρηστών.



Εικόνα 22 Ορισμός διευθύνσεων IP για την ενσύρματη και ασύρματη διεπαφή

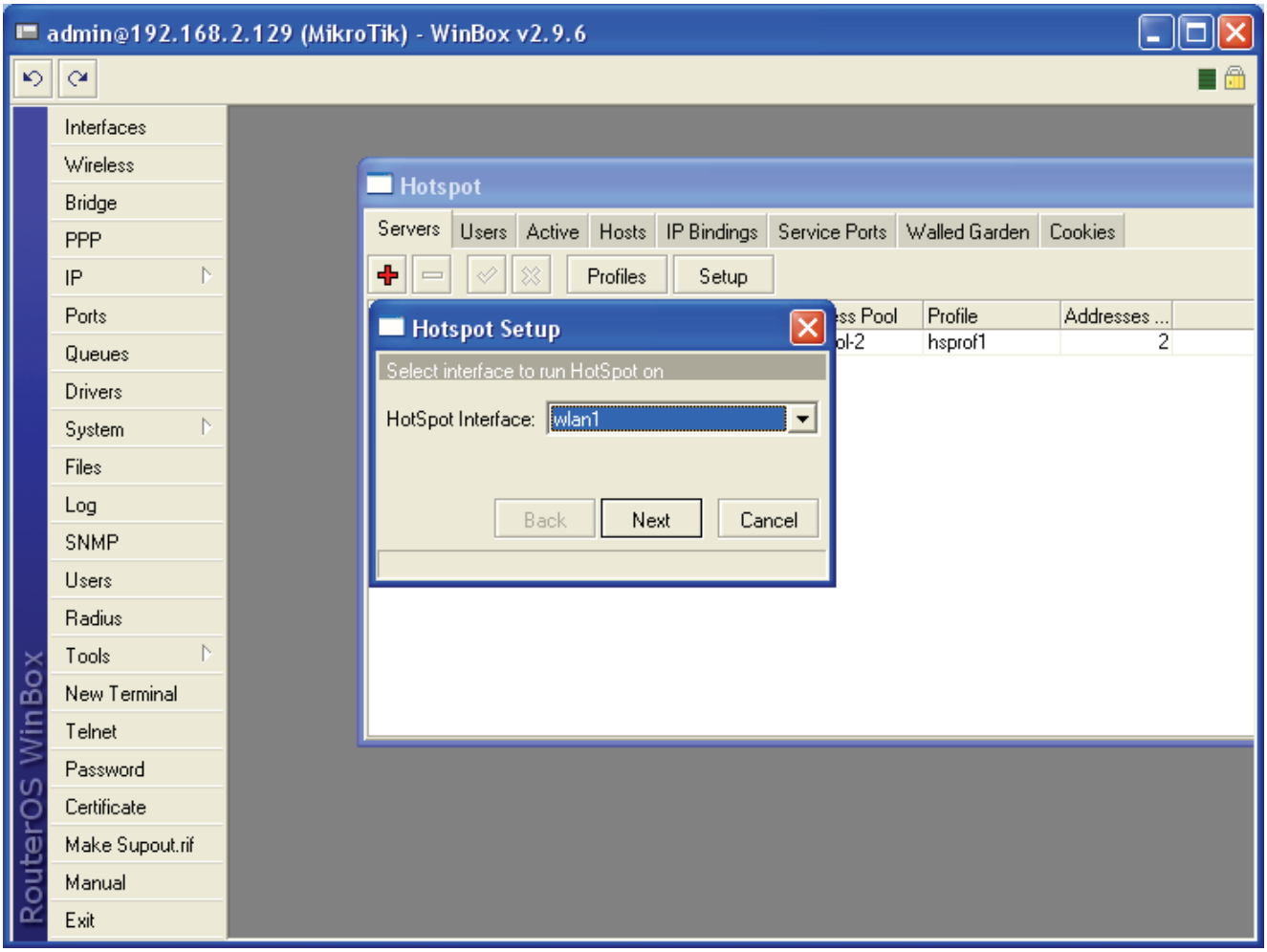
Επόμενο βήμα είναι η ρύθμιση του πίνακα δρομολόγησης ώστε να μπορεί το Hotspot να προσφέρει πρόσβαση στο διαδίκτυο. Αυτό ορίζεται όπως φαίνεται και στην παρακάτω εικόνα στη καρτέλα IP -> Routes. Εκεί ορίζουμε μη στατική δρομολόγηση 0.0.0.0/0 με πύλη 192.168.2.2 η οποία είναι και η πύλη που μας παρέχει πρόσβαση στο διαδίκτυο.



Εικόνα 23 Ρύθμιση του πίνακα δρομολόγησης

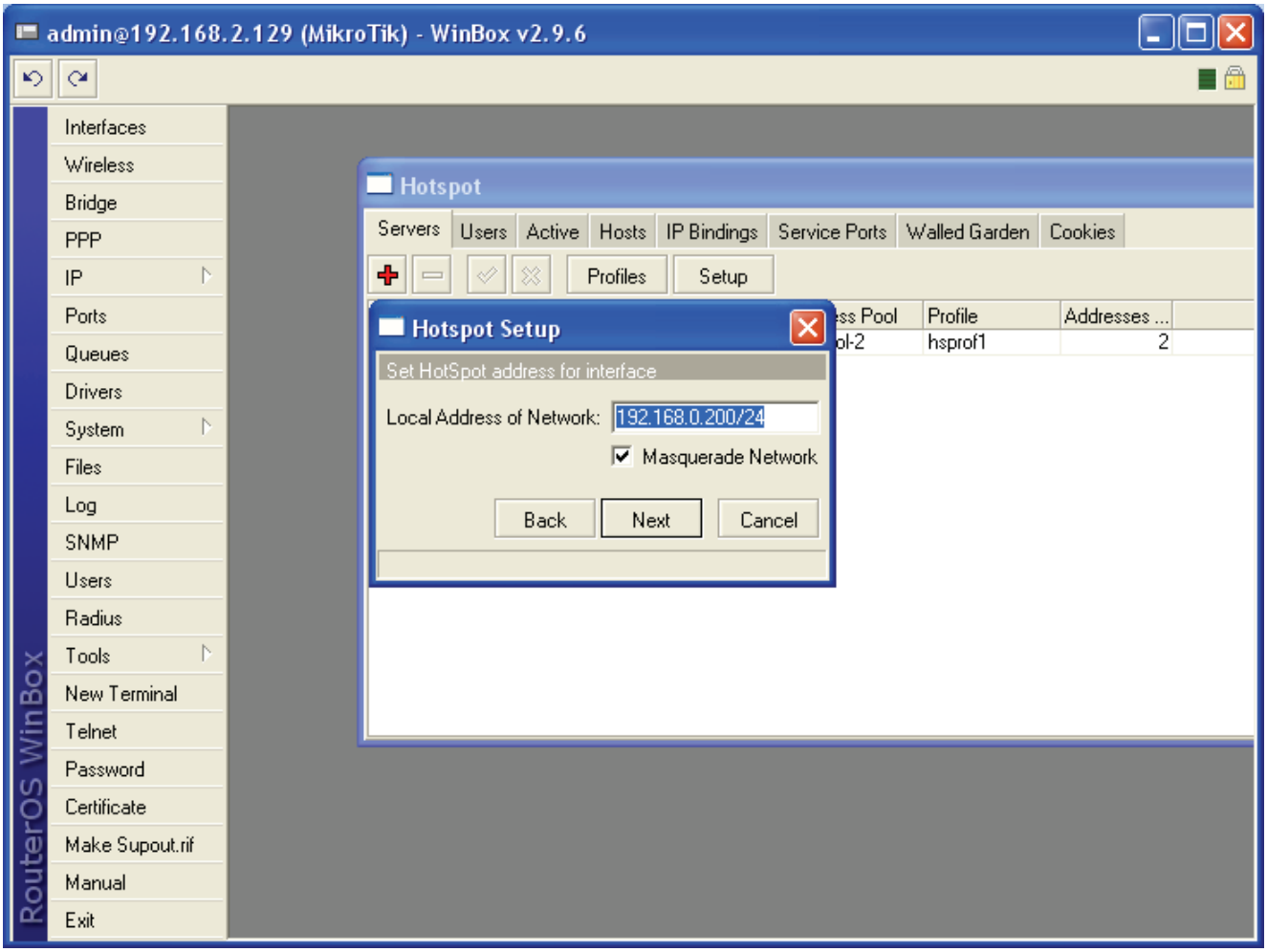
7.2.2.2 Παραμετροποίηση Hotspot

Αφού ολοκληρώσαμε τις βασικές ρυθμίσεις για τη σωστή λειτουργία του AP, σειρά έχει η ρύθμιση της λειτουργίας Hotspot. Οι ρυθμίσεις βρίσκονται στην καρτέλα IP-> Hotspot Στο παράθυρο του Hotspot Service και στο tab servers πατάμε setup και έπειτα επιλεγούμε το interface του AP.



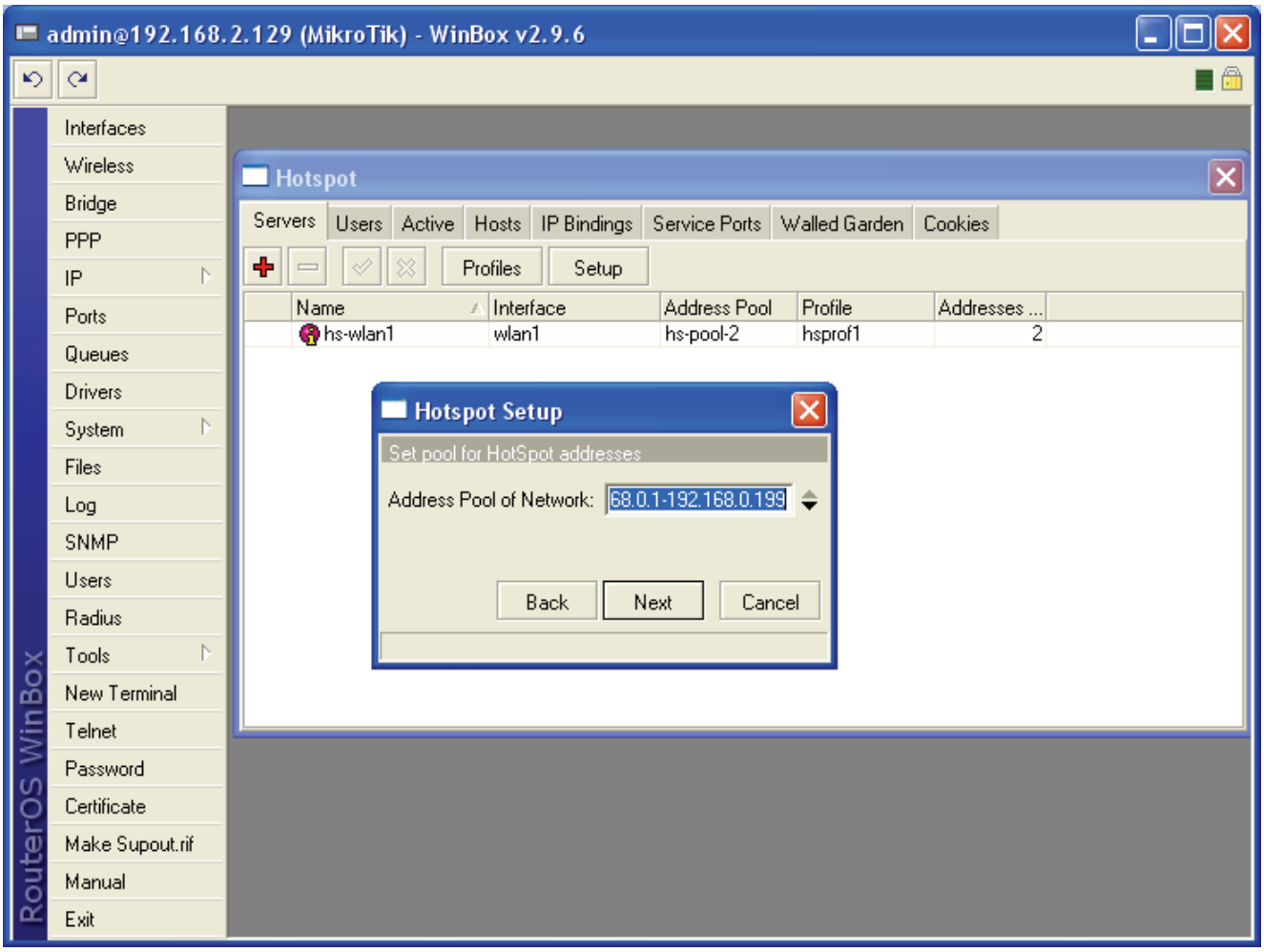
Εικόνα 24 Επιλογή διεπαφής για λειτουργία Hotspot.

Έπειτα στην επόμενη επιλογή εμφανίζεται η προεπιλεγμένη διεύθυνση IP που έχουμε δώσει στην ασύρματη διεπαφή του AP, οπότε δε χρειάζεται καμία αλλαγή. Πατάμε next.



Εικόνα 25 Η προεπιλεγμένη διεύθυνση IP που έχουμε δώσει στην ασύρματη διεπαφή του AP

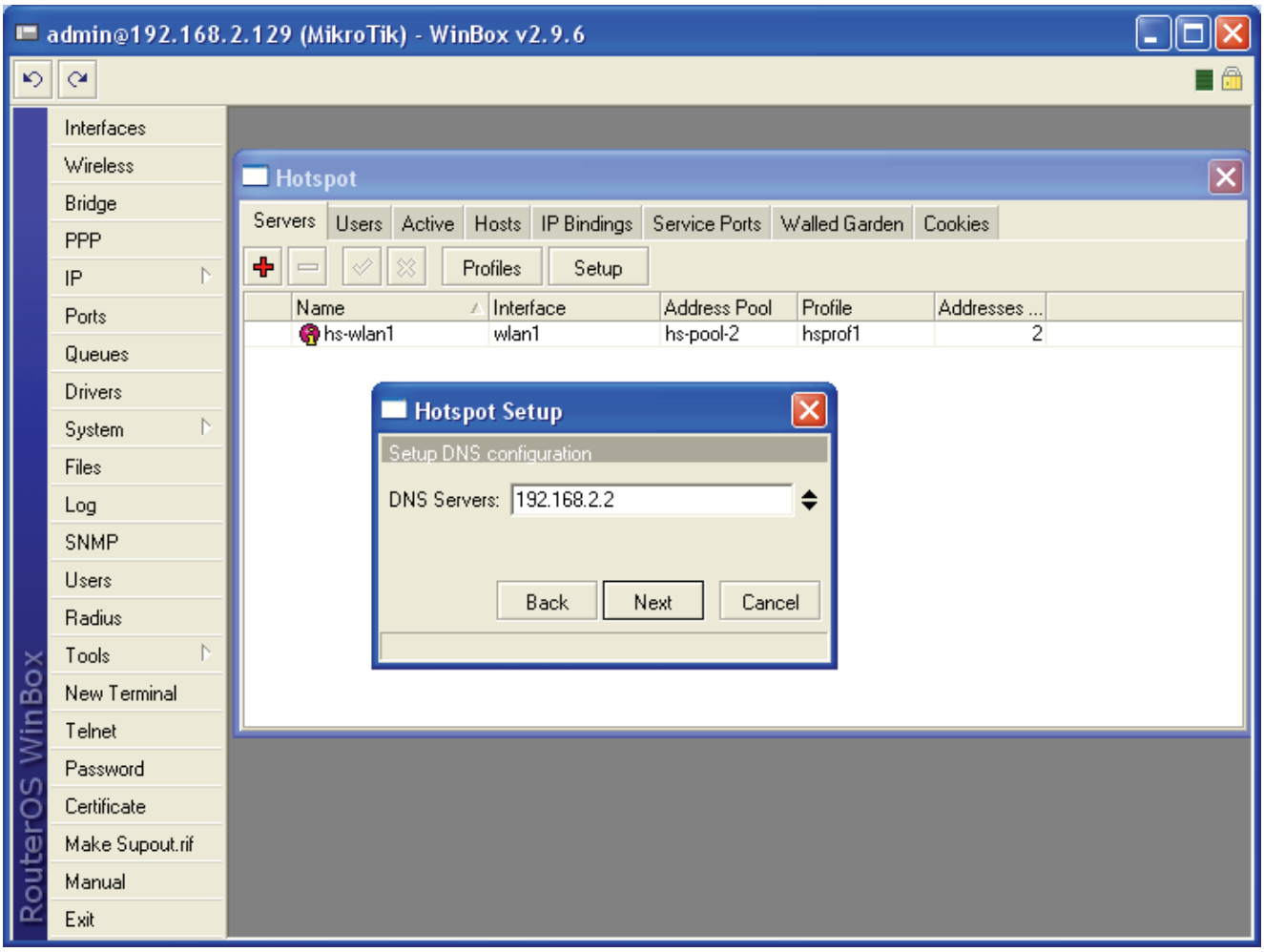
Η αμέσως επόμενη επιλογή είναι ο ορισμός εύρους διευθύνσεων που μπορούν να πάρουν οι ασύρματοι χρήστες του Hotspot. Στη περίπτωσή μας το εύρος αυτό είναι 192.168.0.1- 192.168.0.199.



Εικόνα 26 Εύρος διευθύνσεων IP ασύρματων χρηστών

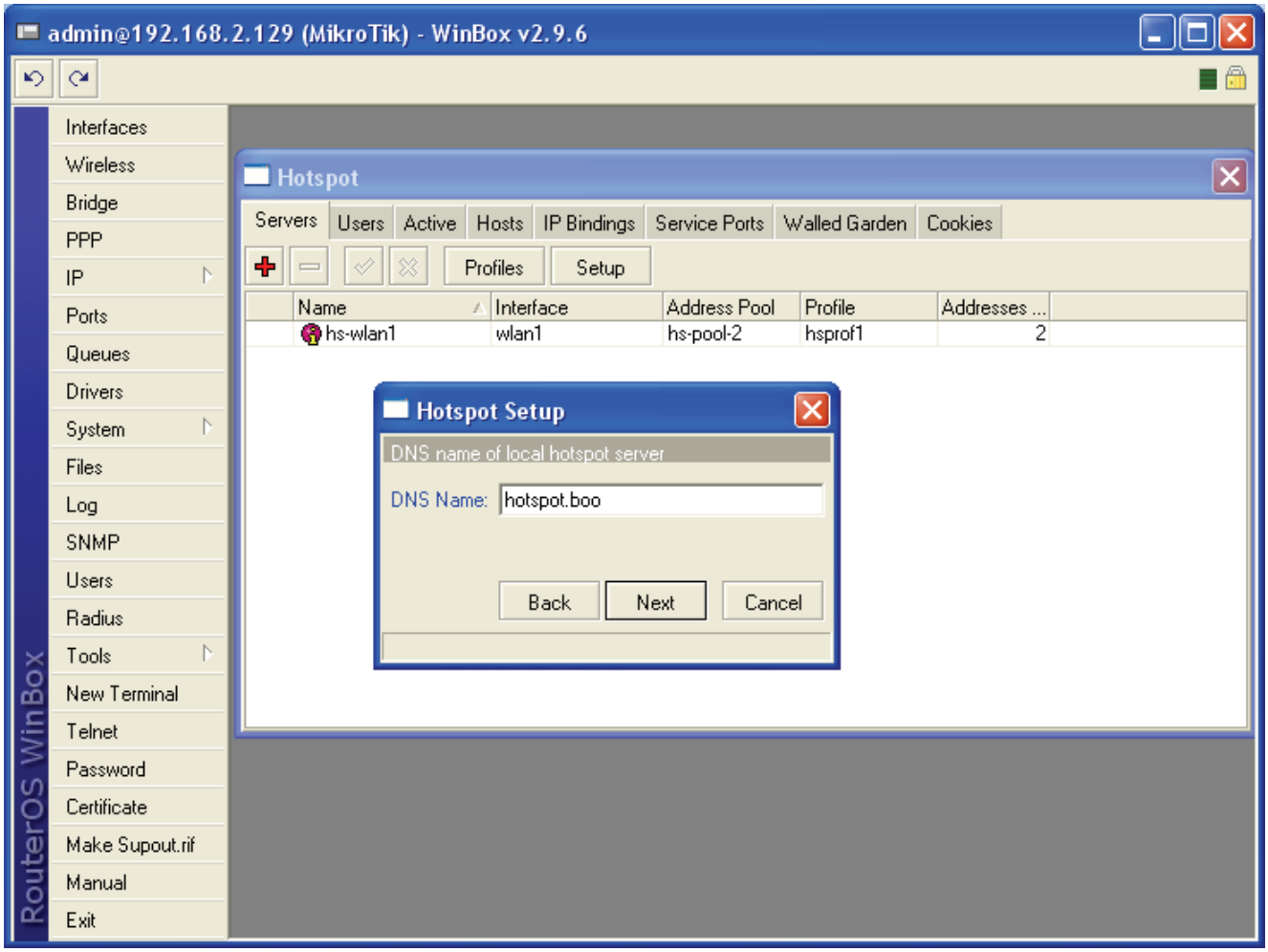
Στις επόμενες δύο επιλογές ορίζουμε αν θα χρησιμοποιήσουμε πιστοποιητικά για την ταυτοποίηση των χρηστών και τον ορισμό SMTP server για την λήψη των μηνυμάτων e-mail που μπορεί ο διαχειριστής να λάβει από τους πελάτες του Hotspot. Στη συγκεκριμένη περίπτωση δεν χρησιμοποιούμε πιστοποιητικά, οπότε επιλέγουμε “none” και δεν έχουμε SMTP server οπότε αφήνουμε την επιλογή αυτή ως έχει.

Στην επόμενη επιλογή ορίζουμε την διεύθυνση IP του DNS server που θα εξυπηρετεί τους πελάτες του Hotspot. Η διεύθυνση αυτή τους γνωστοποιείται κατά τη σύνδεση τους με το Hotspot.



Εικόνα 27 Ορισμός των DNS servers

Έπειτα ορίζουμε το τοπικό DNS όνομα της υπηρεσίας του Hotspot. Αυτή χρησιμοποιείται μόνο τοπικά οπότε μπορεί να έχει οποιοδήποτε όνομα, έστω ότι έχει όνομα “hotspot.boo”



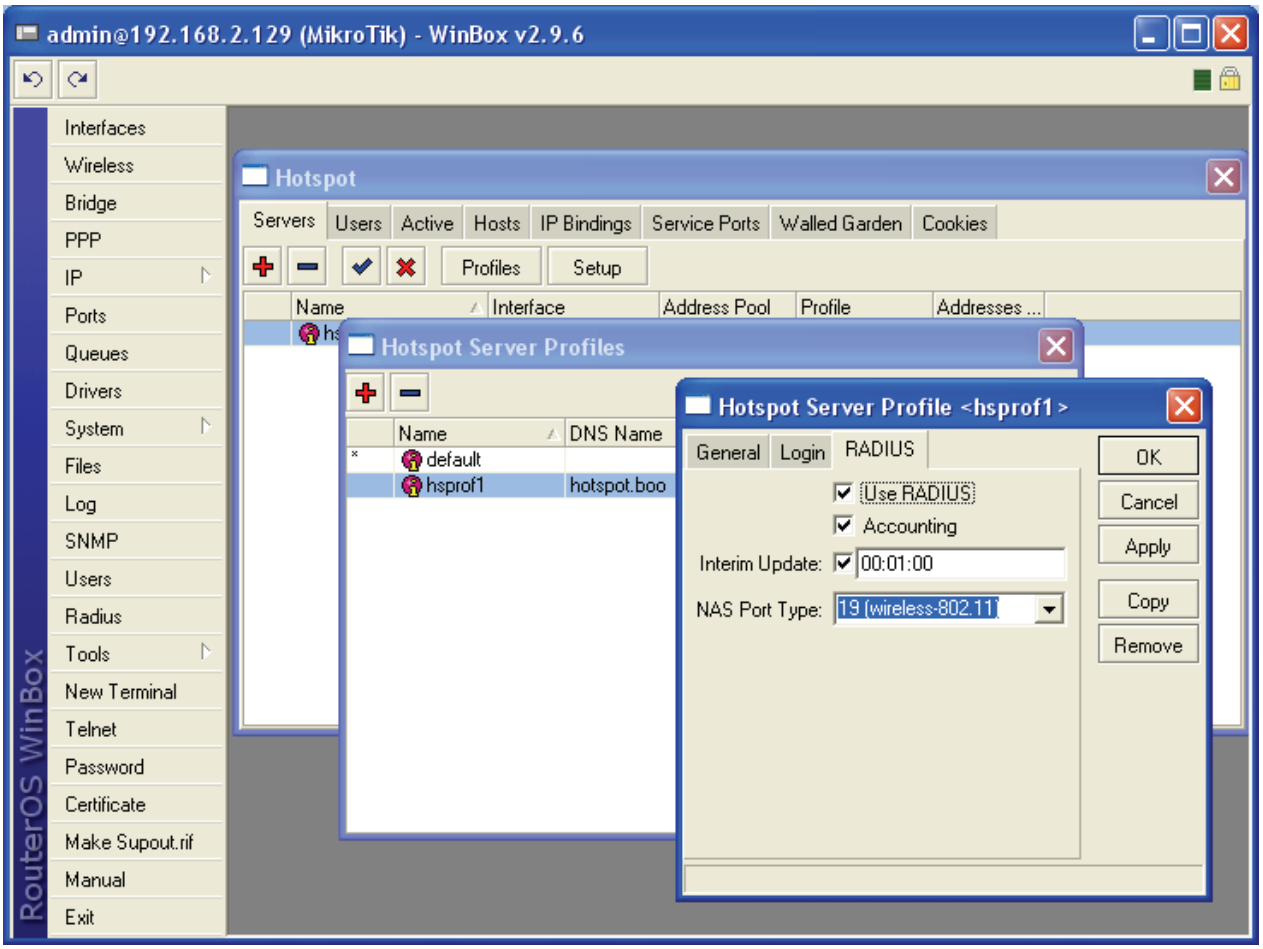
Εικόνα 28 Ορισμός DNS ονόματος υπηρεσίας Hotspot

Τέλος ορίζουμε έναν τοπικό χρήστη για την είσοδο στην υπηρεσία του Hotspot για λόγους δοκιμής.. Έστω ο χρήστης είναι ο admin με κωδικό πρόσβασης 123

Εικόνα 29 Ορισμός τοπικού χρήστη

7.2.2.3 Παραμετροποίηση Hotspot – Ρύθμιση επικοινωνίας με FreeRADIUS

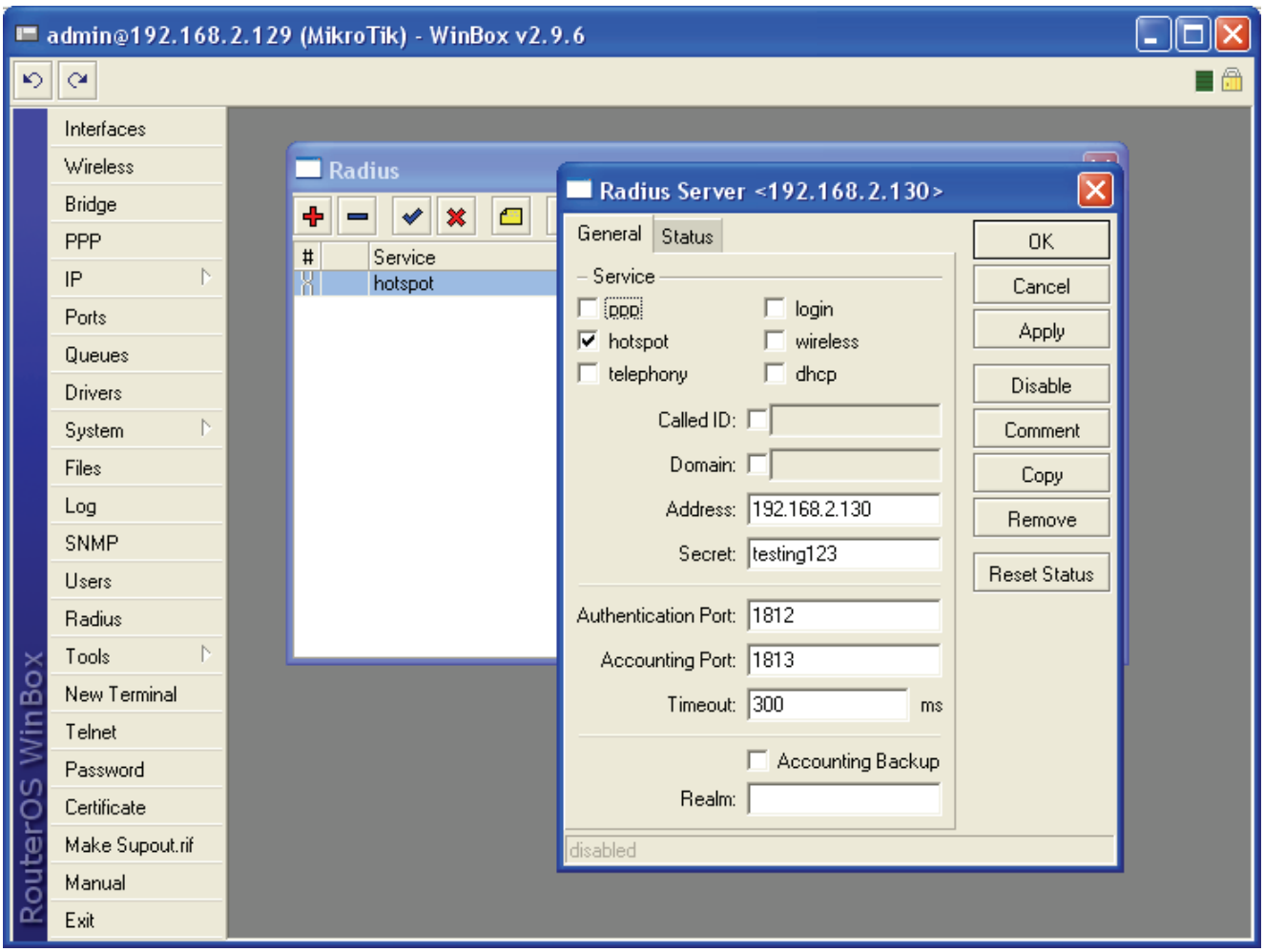
Για τη ρύθμιση λειτουργίας με το FreeRADIUS, αρχικά επιλέγουμε από το μενού του Hotspot την επιλογή Profiles και διαλέγουμε το προφίλ που δημιουργήσαμε στα προηγούμενα βήματα. Έπειτα στο προφίλ διαλέγουμε το RADIUS και ενεργοποιούμε τις επιλογές "Use Radius" και "Accounting" και ως NAS-Type επιλέγουμε "19(wireless 802.11)"



Εικόνα 30 Ρύθμιση επιλογής Radius για αυθεντικοποίηση

Η τελευταία ρύθμιση είναι ο ορισμός του RADIUS server στο AP. Αυτό γίνεται στην επιλογή "Radius". Ορίζουμε ότι η υπηρεσία θα χρησιμοποιηθεί μόνο σε Hotspot και ορίζουμε τη διεύθυνση του

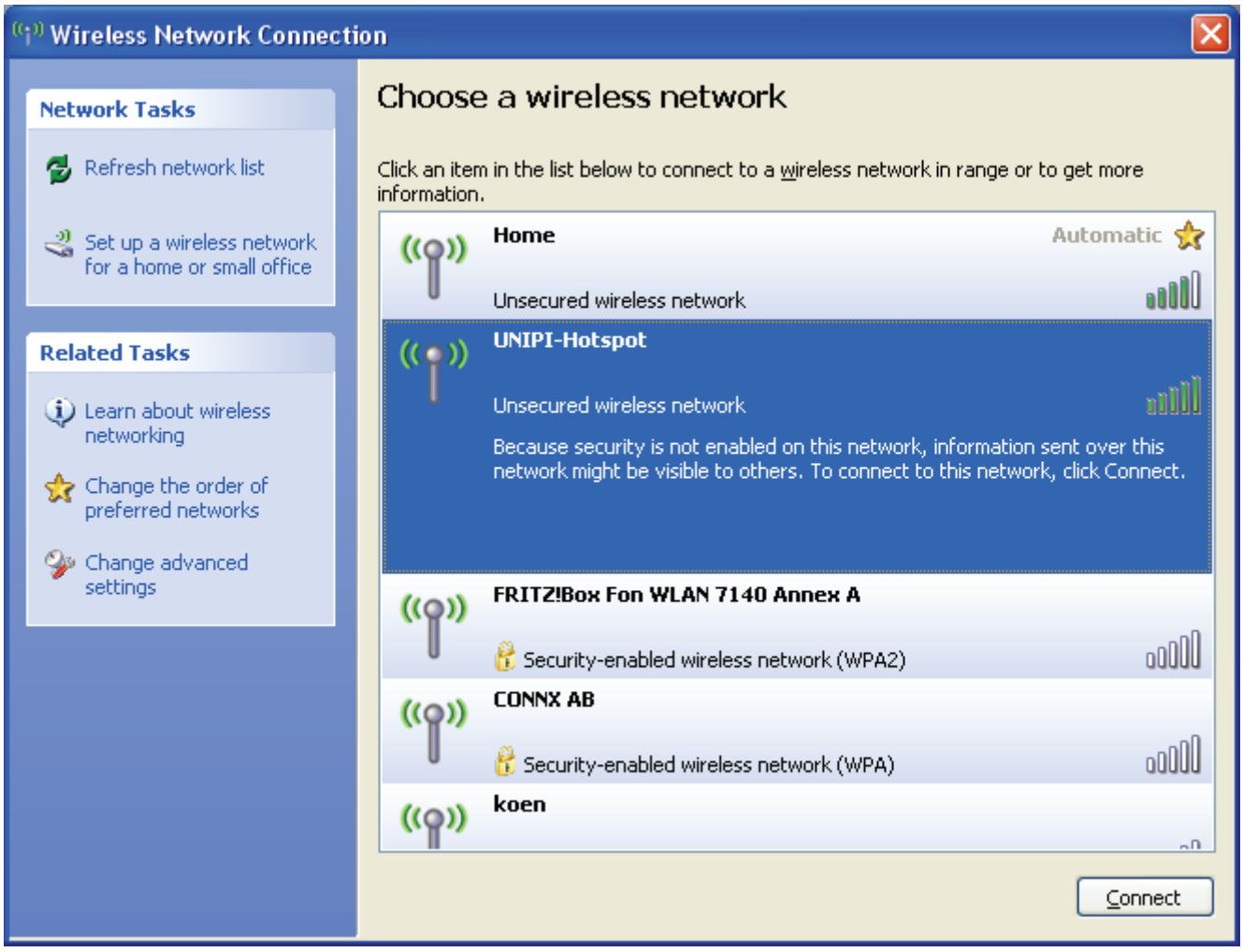
FreeRADIUS server και το διαμοιραζόμενο μυστικό, που στη περίπτωσή μας είναι 192.168.2.130 και testing123 αντίστοιχα.



Εικόνα 31 Ρύθμιση RADIUS server

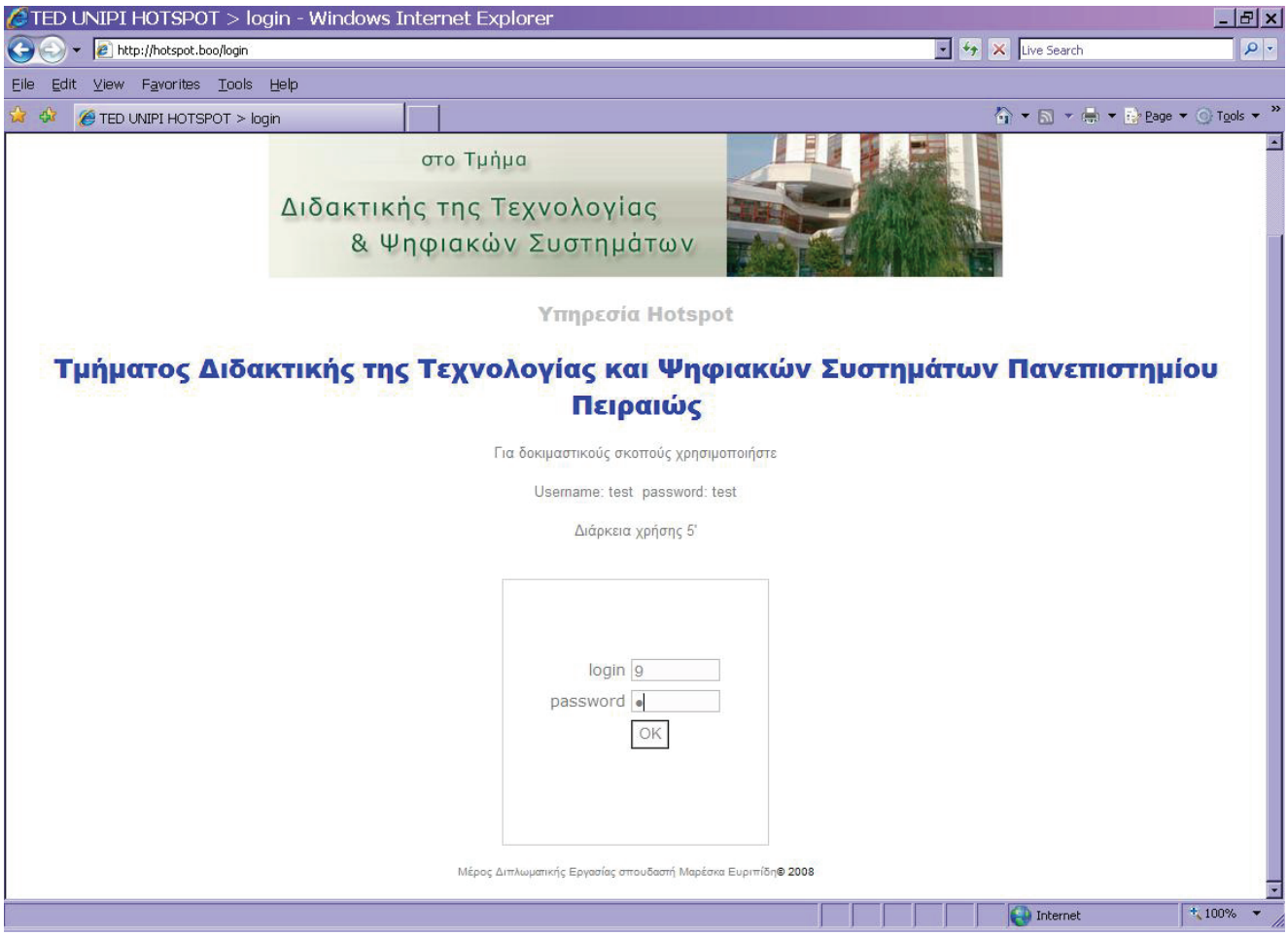
7.3 Παράδειγμα Λειτουργίας Hotspot

Αρχικά αναζητούμε το AP με την SSID του Hotspot μας. Στη περίπτωσή μας το Hotspot εχει SSID "UNIPI-Hotspot"



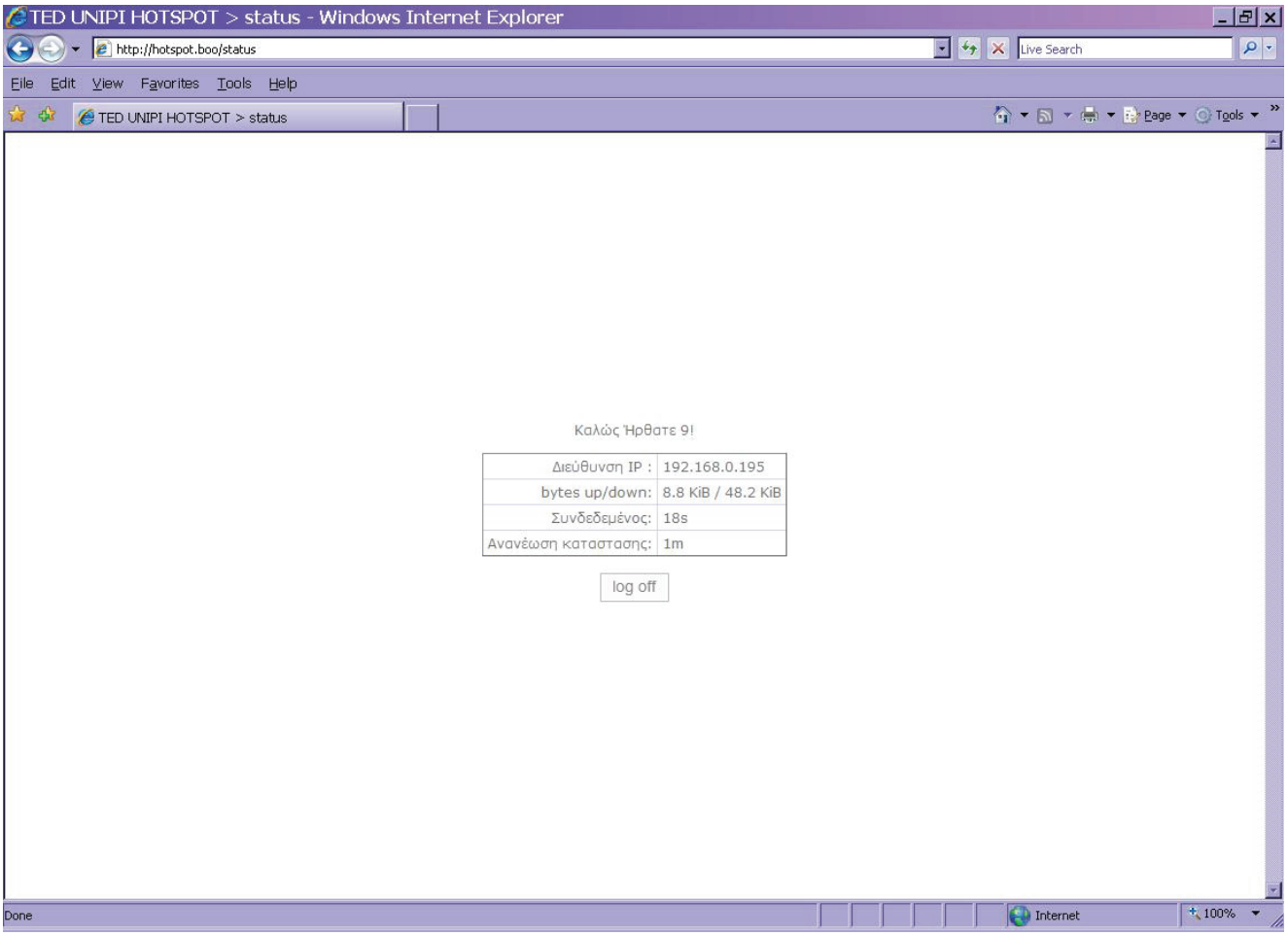
Εικόνα 32 Αναζήτηση Hotspot

Αφού γίνει η σύνδεση ανοίγουμε τον Internet Explorer και θα μας εμφανίσει την αρχική σελίδα όπου θα όνομα χρήστη και κωδικό πρόσβασης για να συνδεθούμε. Έστω ότι συνδεόμαστε με όνομα χρήστη "9" και κωδικό πρόσβασης "9" όπως ορίσαμε προηγουμένως με δυνατότητα πρόσβασης για 60 λεπτά. Η μορφή της σελίδας είναι η ακόλουθη:



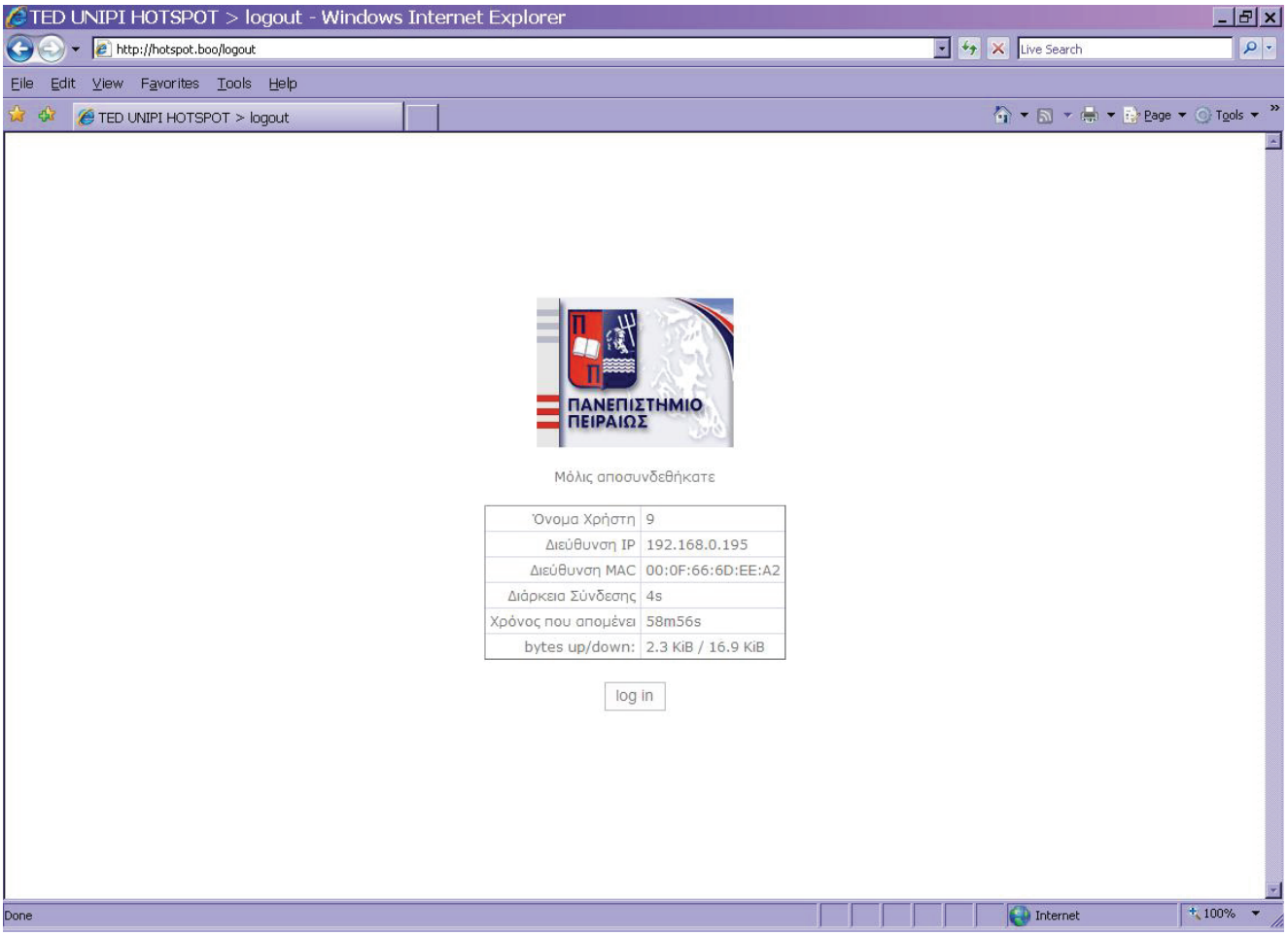
Εικόνα 33 Σελίδα Login του Hotspot

Αφού γίνει επιτυχής σύνδεση μας εμφανίζεται η σελίδα κατάστασης όπου αναφέρεται η διάρκεια σύνδεσης, η διεύθυνση IP που αποδόθηκε καθώς και το σύνολο των bytes που έχουν διακινηθεί



Εικόνα 34 Σελίδα κατάστασης του Hotspot

Η διακοπή της σύνδεσης μπορεί να επιτευχθεί πατώντας στην σελίδα κατάστασης “log off”. Έπειτα εμφανίζεται μία νέα σελίδα όπου αναγράφονται η διεύθυνση IP του πελάτη, η διεύθυνση MAC της κάρτας δικτύου του, και πληροφορίες για το χρόνο που παρέμεινε συνδεδεμένος, το χρόνο που του απομένει εάν επανασυνδεθεί και το σύνολο των bytes που διακινήθηκαν.



Εικόνα 35 Σελίδα Αποσύνδεσης από το Hotspot

Τέλος αξιοσημείωτο είναι ότι το σύστημα είναι ρυθμισμένο να παρέχει πρόσβαση στους χρήστες που δεν έχουν εγγραφεί στο σύστημα μόνο στις ιστοσελίδες του πανεπιστημίου www.ted.unipi.gr και www.unipi.gr.

Αναφορές

[1] Jonathan Hassel, Radius: Securing public access to private resources. O'Reilly 2002

[2] Madjid Nakhjiri, Mahsa Nakhjiri, AAA and Network Security for Mobile Access. Wiley 2005

[3] http://www.islab.demokritos.gr/gr/html/ptixiakes/kostas-aris_ptyxiakh/Phtml/radius.htm

[4] http://coova.org/wiki/index.php/JRadius

[5] http://freeradius.org/

[6] http://www.dd-wrt.com/dd-wrtv3/index.php

[7] http://www.winet.gr/sol_hotspots.html

[8] http://www.modwest.com/help/kb6-33.html

[9] http://infodotnet.blogspot.com/2007/11/part-1-mikrotik-hotspot-freeradius.html

[10]http://www.smallnetbuilder.com/index2.php?option=com_content&task=view&id=30210&pop=1&page=0&Itemid=98

[11]http://www.smallnetbuilder.com/index2.php?option=com_content&task=view&id=30213&pop=1&page=0&Itemid=98

[12] http://tldp.org/HOWTO/html_single/8021X-HOWTO/

Παράρτημα Α Γλωσσάρι Όρων

PAP (Password Authentication Protocol)

Είναι ένα απλό πρωτόκολλο αυθεντικοποίησης που χρησιμοποιείται για την αυθεντικοποίηση ενός χρήστη στο NAS που μπορεί να χρησιμοποιείται για παράδειγμα. από παρόχους υπηρεσιών ίντερνετ. Το PAP χρησιμοποιείται από το πρωτόκολλο PPP. Το PAP μεταδίδει μη κρυπτογραφημένους ASCII κωδικούς μέσω δικτύου και γι αυτό θεωρείται μη ασφαλές.. Χρησιμοποιείται ως έσχατη λύση όταν ο απομακρυσμένος server δεν υποστηρίζει πιο δυνατό πρωτόκολλο πιστοποίησης, όπως το CHAP ή EAP. Το πακέτο PAP ενθυλακώνεται σε ένα PPP frame. Το πεδίο του πρωτοκόλλου έχει τιμή C023 (hex).

Κύκλος εργασιών πρωτοκόλλου:

1. Ο πελάτης αποστέλλει όνομα χρήστη και κωδικό πρόσβασης
2. Ο Server αποστέλλει εάν τα διαπιστευτήρια είναι αποδεκτά μήνυμα authentication-ack ή εναλλακτικά εάν δεν είναι αποδεκτά authentication-nak

Περιγραφή	1 byte	1 byte	2 bytes	1 byte	Μεταβλητή	1 byte	Μεταβλητή
Authentication-request	Code= 1	ID	Length	Username length	Username	Password length	Password
Authentication-ack	Code= 2	ID	Length	Message length	Username		
Authentication-nak	Code= 3	ID	Length	Message length	Username		

Πακέτα PAP

CHAP (Challenge-Handshake Authentication Protocol)

Το Challenge-Handshake Authentication Protocol (CHAP) πιστοποιεί ένα χρήστη ή host δικτύου σε μια οντότητα πιστοποίησης που μπορεί να είναι πχ. ένας πάροχος πρόσβασης στο ίντερνετ. Το πρωτόκολλο προσδιορίζεται από το RFC 1994: PPP Challenge Handshake Authentication Protocol (CHAP).

Το CHAP είναι ένα scheme αυθεντικοποίησης που χρησιμοποιείται από Point to Point Protocol (PPP) servers για να πιστοποιήσει την ταυτότητα τον απομακρυσμένων χρηστών. Το CHAP περιοδικά επιβεβαιώνει την ταυτότητα του πελάτη με τη χρήση τριών δρόμων χειραψία (three-way handshake). Αυτό πραγματοποιείται κατά τη διάρκεια εδραίωσης της αρχικής σύνδεσης, και μπορεί πραγματοποιηθεί πάλι οποιαδήποτε άλλη στιγμή μετέπειτα. Η επιβεβαίωση βασίζεται σε ένα κοινό μυστικό (όπως για παράδειγμα ο κωδικός πρόσβασης του πελάτη).

1. Έπειτα από την ολοκλήρωση της φάσης εγκαθίδρυσης σύνδεσης, ο authenticator αποστέλλει ένα "challenge" μήνυμα στον αποδέκτη (άλλο άκρο) peer.
2. Ο peer απαντά με μία τιμή που υπολογίζεται από μία μονόδρομη συνάρτηση σύνοψης (one-way hash function), όπως η σύνοψη MD5 checksum.
3. Ο authenticator ελέγχει την απάντηση μέσω του δικού του υπολογισμού της αναμενόμενης τιμής σύνοψης. Αν οι τιμές ταιριάζουν, ο authenticator αναγνωρίζει την πιστοποίηση, αλλιώς τερματίζει τη σύνδεση.
4. Κατά τυχαία χρονικά διαστήματα ο authenticator αποστέλλει νέο challenge στο peer και επαναλαμβάνει τα βήματα 1 έως 3.

Το πρωτόκολλο CHAP παρέχει προστασία ενάντια σε επιθέσεις playback attack από το peer μέσω της χρήσης ενός αυξητικά αλλαγμένου identifier και μιας μεταβλητής challenge-value. Επίσης απαιτείται και ο πελάτης και ο server να γνωρίζουν το κείμενο του μυστικού παρ' όλο που δεν αποστέλλεται ποτέ μέσω δικτύου.

Η Microsoft έχει δημιουργήσει μια παραλλαγή του CHAP, που την ονομάζει MS-CHAP, και δεν απαιτεί κανένας από τα peer να γνωρίζει το κείμενο του μυστικού.

Κύκλος εργασιών πρωτοκόλλου

1. Challenge Packet (Σύστημα στο Χρήστη)
2. Response Packet (Χρήστης στο Σύστημα)
3. Success or failure packet (Σύστημα στο Χρήστη)

Πακέτα CHAP

Περιγραφή	1 byte	1 byte	2 bytes	1 byte	Μεταβλητή	Μεταβλητή
Challenge	Code = 1	ID	Length	Challenge length	Challenge value	Name
Response	Code = 2	ID	Length	Response Length	Response value	Name
Success	Code = 3	ID	Length		Message	
Failure	Code = 4	ID	Length		Message	

Το πακέτο CHAP ενθυλακώνεται σε ένα PPP frame. Το πεδίο του πρωτοκόλλου έχει τιμή C223(hex)

EAP (Extensible Authentication Protocol)

Το EAP έχει παρουσιαστεί τελευταία ως το νεότερο πρωτόκολλο πιστοποίησης PPP και υιοθετεί δυνατότητες του MS CHAP V2. Κατά τη διάρκεια της φάσης πιστοποίησης το EAP δεν λειτουργεί ακόμα. Αυτή είναι και η μεγαλύτερη διαφορά του από τις άλλες μεθόδους. Το EAP δεν πραγματοποιεί κανενός είδους πιστοποίηση, αλλά μεσολαβεί μόνο μεταξύ του πραγματικού EAP τύπου και η πιστοποίηση του χρήστη γίνεται από τον Domain controller που κρατά την βάση χρηστών, ή ένα RADIUS που σκοπός του είναι σαν ατζέντης (agent) να πάρει τα διαπιστευτήρια του χρήστη που επιβεβαιώνονται από έναν Domain Controller.

Μέχρι το MS-CHAP V2, αυτή η πιστοποίηση γινόταν μόνο στο NAS server με την βάση δεδομένων του χρήστη αλλά με το EAP, αυτό δεν γίνεται μονάχα από μία κεντρική βάση δεδομένων χρηστών ή έναν Domain Controller.

Το EAP είναι ένα νέο πρωτόκολλο πιστοποίησης PPP που επιτρέπει μια αυθαίρετη μέθοδο πιστοποίησης. Από τη στιγμή που ο χρήστης συνδεθεί μέσω PPP,ο NAS server αμέσως συλλέγει τα διαπιστευτήρια του χρήστη και τα αποστέλλει σε ένα RADIUS η Domain Controller για επιβεβαίωση.

Οι κατηγορίες πιστοποίησης EAP είναι οι ακόλουθες:

- **EAP-MD5:** Το MD5-Challenge απαιτεί όνομα χρήστη και κωδικό πρόσβασης, και είναι ισάξιο με το πρωτόκολλο PPP CHAP [RFC1994]. Αυτή η μέθοδος δε παρέχει αντοχή σε επιθέσεις λεξικού dictionary attack, αμοιβαία αυθεντικοποίηση, ή παραγωγή κλειδιού, και γι' αυτό το λόγο έχει πολύ μικρή χρήση σε περιβάλλοντα ασύρματης πιστοποίησης
- **Lightweight EAP (LEAP):** Ένας συνδυασμός όνομα χρήστη και κωδικού πρόσβασης αποστέλλεται σε έναν server Πιστοποίησης (RADIUS) για πιστοποίηση. Το LEAP αναπτύχθηκε από τη Cisco, και δε θεωρείται ασφαλές.
- **EAP-TLS:** Δημιουργεί μια σύνοδο TLS session με το EAP, μεταξύ του πιστοποίητη και του server Πιστοποίησης. Και ο server και οι πελάτες χρειάζονται ένα έγκυρο πιστοποιητικό (x509), και επομένως ένα PKI. Αυτή η μέθοδος παρέχει αμοιβαία πιστοποίηση. Το EAP-TLS περιγράφεται στο RFC2716.
- **EAP-TTLS:** Δημιουργεί ένα κρυπτογραφημένο TLS-tunnel για ασφαλή μεταφορά των δεδομένων πιστοποίησης. Μέσα στο TLS tunnel, μπορούν να χρησιμοποιηθούν οποιεσδήποτε άλλες μέθοδοι αυθεντικοποίησης. Αναπτύχθηκε από την Funk Software και Meetinghouse, και είναι προς το παρών προσχέδιο της IETF.
- **Protected EAP (PEAP):** Χρησιμοποιεί, όπως το EAP-TTLS, ένα κρυπτογραφημένο TLS-tunnel. Αναπτύχθηκε από τη Microsoft, Cisco, και RSA Security, και είναι προσχέδιο της IETF.
- **EAP-MSCHAPv2:** Απαιτεί όνομα χρήστη και κωδικό πρόσβασης, και έιανι βασικά μία EAP ενθυλάκωση του MS-CHAP-v2 [RFC2759]. Συνήθως χρησιμοποιείται μέσα στε ένα κρυπτογραφημένο PEAP tunnel. Αναπτύχτηκε από την Microsoft, είναι προς το παρών προσχέδιο της IETF.

L2TP (Layer 2 Tunneling Protocol)

Το L2TP δρα σαν ένα πρωτόκολλο επιπέδου ζεύξης δεδομένων (data link layer) (2ου επιπέδου του μοντέλου OSI) για δίοδο (tunnelling) της κίνησης του δικτύου μεταξύ 2 peers πάνω από ένα

υπάρχον δίκτυο (συνήθως το ίντερνετ). Στη πραγματικότητα το L2TP είναι πρωτόκολλο 5ου επιπέδου (επιπέδου συνόδου), και χρησιμοποιεί την εγγεγραμμένη θύρα UDP 1701. Ολόκληρο το πακέτο L2TP, συμπεριλαμβανομένου του payload και της επικεφαλίδας L2TP, αποστέλλεται μέσα στο UDP datagram. Είναι κοινότυπο να έχουμε συνόδους Point-to-Point Protocol (PPP) μέσα σε ένα L2TP tunnel.

Το L2TP δεν παρέχει από μόνο του εμπιστευτικότητα ή ισχυρή πιστοποίηση IPsec συχνά χρησιμοποιείται για να ασφαλίσει τα πακέτα L2TP με την παροχή εμπιστευτικότητας, πιστοποίησης και ακεραιότητας. Ο συνδυασμός αυτών των δύο πρωτοκόλλων είναι ευρέως γνωστό ως L2TP/IPsec.

Τα δύο άκρα ενός L2TP tunnel ονομάζονται LAC (L2TP Access Concentrator) και LNS (L2TP Network Server). Ο LAC ξεκινά το tunnel ενώ ο LNS είναι ο server, που περιμένει για νέα tunnels. Όταν ένα tunnel εγκατασταθεί, Η κίνηση του δικτύου μεταξύ των peers είναι αμφίδρομη. Για να είναι χρήσιμο για δικτύωση, μεγαλύτερου επιπέδου πρωτόκολλα τρέχουν μέσω του L2TP tunnel. Για διευκολυνθεί αυτό αυτή η σύνοδος L2TP πραγματοποιείται μέσα στο tunnel σε κάθε υψηλότερου επιπέδου πρωτοκόλλου όπως το PPP. Η έναρξη της συνόδου μπορεί να γίνει είτε από το LAC είτε το LNS. Η κίνηση για κάθε σύνοδο απομονώνεται από το L2TP, έτσι είναι δυνατό να πραγματοποιήσουμε πολλαπλά εικονικά δίκτυα μέσα από ένα και μόνο tunnel.Το MTU πρέπει επίσης να λαμβάνεται υπ όψιν εφαρμόζεται το L2TP.

Τα πακέτα που ανταλλάσσονται με ένα L2TP tunnel κατηγοριοποιούνται είτε ως πακέτα έλεγχου είτε ως πακέτα δεδομένων. Το L2TP παρέχει δυνατότητες αξιοπιστίας για τα πακέτα ελέγχου, αλλά καμία αξιοπιστία για τα πακέτα δεδομένων. Εάν απαιτείται αξιοπιστία και σε αυτά πρέπει να παρέχεται από τα ενθυλακωμένα πρωτόκολλα που τρέχουν μέσα σε κάθε σύνοδο του L2TP tunnel.

IEEE 802.1X

Το IEEE 802.1X είναι ένα IEEE standard για έλεγχο πρόσβασης δικτύου βασισμένο σε θύρα (port-based Network Access Control) και είναι μέρος του συνόλου πρωτοκόλλων IEEE 802 (802.1). Παρέχει πιστοποίηση σε συσκευές συνδεμένες σε μια θύρα LAN, παρέχοντας σύνδεση point-to-point ή αποτρέποντας πρόσβαση από αυτή τη θύρα αν η πιστοποίηση αποτύχει. Χρησιμοποιείται από ορισμένα κλειστά ασύρματα access points, και βασίζεται στο EAP.

Το 802.1X είναι διαθέσιμο από ορισμένα switches δικτύου, και μπορεί να ρυθμιστεί να πιστοποιεί hosts που είναι εξοπλισμένοι από το συμβατό λογισμικό, απορρίπτοντας μη εξουσιοδοτημένη πρόσβαση στο επίπεδο ζεύξης δεδομένων.

Μερικοί κατασκευαστές εφαρμόζουν 802.1X για ασύρματα access points, για να χρησιμοποιούνται σε συγκεκριμένες περιστάσεις όπου ένα access point χρειάζεται να λειτουργεί σαν "κλειστό" access point, αντιμετωπίζοντας τις αδυναμίες ασφαλείας του WEP. Η πιστοποίηση γίνεται συνήθως από μια τρίτη οντότητα, όπως για παράδειγμα ένας RADIUS server. Αυτό παρέχει πιστοποίηση μόνο του πελάτη, ή πιο σωστά, ισχυρή αμοιβαία πιστοποίηση με τη χρήση πρωτοκόλλων όπως το EAP-TLS.

Σε πολλές περιπτώσεις, το κοινό προσκαλείται στις προϋποθέσεις αλλά όχι για να συνδεθεί στο δίκτυο. Στη περίπτωση ενσύρματου δικτύου, είναι δυνατό να γίνεται έλεγχος πρόσβασης μέσω φυσικής ασφάλειας (physical security) σε όλες τις θύρες δικτύου. Επειδή αυτό δεν είναι δυνατό σε ένα IEEE 802.11 ασύρματο σήμα, οι διαχειριστές των κλειστών access points μπορούν εναλλακτικά να χρησιμοποιήσουν 802.1X ή άλλους ελέγχους πρόσβασης στο επίπεδο ζεύξης δεδομένων. Αυτή η συσχέτιση μεταξύ ασύρματης διασύνδεσης και χρήσης της 802.1X πιστοποίησης έχει εσφαλμένα οδηγήσει μερικούς να ονομάζουν το στάνταρτ "802.11x" όταν χρησιμοποιείται hen σε ένα ασύρματο δίκτυο.

Έπειτα από την ανίχνευση του νέου πελάτη (supplicant), η θύρα στο switch (authenticator) θα ενεργοποιηθεί και θα μπει στη κατάσταση μη εξουσιοδοτημένου ("unauthorized" state). Σε αυτή την κατάσταση, μόνο 802.1X κίνηση θα επιτρέπεται και οποιαδήποτε άλλη κίνηση, όπως DHCP και HTTP, θα μπλοκάρεται στο επίπεδο ζεύξης δεδομένων. Ο authenticator θα στείλει το αίτημα ταυτότητας EAP (EAP-Request identity) στο supplicant, έπειτα ο supplicant θα στείλει το πακέτο απάντησης EAP (EAP-response packet**)** που ο authenticator θα προωθήσει στο server πιστοποίησης. Ο server πιστοποίησης μπορεί να δεχθεί ή να απορρίψει το αίτημα EAP. Εάν το αποδεχθεί, ο authenticator θα θέσει τη θύρα σε κατάσταση εξουσιοδοτημένου ("authorized" mode) και κανονική κίνηση θα επιτρέπεται. Όταν ο supplicant αποσυνδέεται θα στείλει ένα EAP-logoff μήνυμα στον authenticator η θύρα θα επανέλθει στην κατάσταση "unauthorized", μπλοκάροντας πάλι όλη την μη EAP κίνηση.

Παράρτημα Β Scripts

1. Το ακόλουθο script χρησιμοποιείται για την δημιουργία των εγγραφών του FreeRADIUS που χρειάζονται στη βάση δεδομένων της MySQL όπως είδαμε στο Κεφάλαιο 5.4

```
##########################################################################
# db_mysql.sql      rlm_sql - FreeRADIUS SQL Module  #
#                   #
#  Database schema for MySQL rlm_sql module       #
#                   #
#  To load:               #
#   mysql -uroot -prootpass radius < db_mysql.sql     #
#                   #
#         Mike Machado <mike@innercite.com>  #
##########################################################################
#
# Table structure for table 'radacct'
#


CREATE TABLE radacct (
 RadAcctId bigint(21) NOT NULL auto_increment,
 AcctSessionId varchar(32) NOT NULL default '',
 AcctUniqueId varchar(32) NOT NULL default '',
 UserName varchar(64) NOT NULL default '',
 Realm varchar(64) default '',
 NASIPAddress varchar(15) NOT NULL default '',
```

```
    NASPortId varchar(15) default NULL,
    NASPortType varchar(32) default NULL,
    AcctStartTime datetime NOT NULL default '0000-00-00 00:00:00',
    AcctStopTime datetime NOT NULL default '0000-00-00 00:00:00',
    AcctSessionTime int(12) default NULL,
    AcctAuthentic varchar(32) default NULL,
    ConnectInfo_start varchar(50) default NULL,
    ConnectInfo_stop varchar(50) default NULL,
    AcctInputOctets bigint(20) default NULL,
    AcctOutputOctets bigint(20) default NULL,
    CalledStationId varchar(50) NOT NULL default '',
    CallingStationId varchar(50) NOT NULL default '',
    AcctTerminateCause varchar(32) NOT NULL default '',
    ServiceType varchar(32) default NULL,
    FramedProtocol varchar(32) default NULL,
    FramedIPAddress varchar(15) NOT NULL default '',
    AcctStartDelay int(12) default NULL,
    AcctStopDelay int(12) default NULL,
    XAscendSessionSvrKey varchar(10) default NULL,
    PRIMARY KEY (RadAcctId),
    KEY UserName (UserName),
    KEY FramedIPAddress (FramedIPAddress),
    KEY AcctSessionId (AcctSessionId),
    KEY AcctUniqueId (AcctUniqueId),
    KEY AcctStartTime (AcctStartTime),
    KEY AcctStopTime (AcctStopTime),
    KEY NASIPAddress (NASIPAddress)
```

```
) ;

#
# Table structure for table 'radcheck'
#

CREATE TABLE radcheck (
 id int(11) unsigned NOT NULL auto_increment,
 UserName varchar(64) NOT NULL default '',
 Attribute varchar(32) NOT NULL default '',
 op char(2) NOT NULL DEFAULT '==',
 Value varchar(253) NOT NULL default '',
 PRIMARY KEY (id),
 KEY UserName (UserName(32))
) ;

#
# Table structure for table 'radgroupcheck'
#

CREATE TABLE radgroupcheck (
 id int(11) unsigned NOT NULL auto_increment,
 GroupName varchar(64) NOT NULL default '',
 Attribute varchar(32) NOT NULL default '',
 op char(2) NOT NULL DEFAULT '==',
 Value varchar(253) NOT NULL default '',
 PRIMARY KEY (id),
```

```
 KEY GroupName (GroupName(32))
) ;

#
# Table structure for table 'radgroupreply'
#

CREATE TABLE radgroupreply (
 id int(11) unsigned NOT NULL auto_increment,
 GroupName varchar(64) NOT NULL default '',
 Attribute varchar(32) NOT NULL default '',
 op char(2) NOT NULL DEFAULT '=',
 Value varchar(253) NOT NULL default '',
 PRIMARY KEY (id),
 KEY GroupName (GroupName(32))
) ;

#
# Table structure for table 'radreply'
#

CREATE TABLE radreply (
 id int(11) unsigned NOT NULL auto_increment,
 UserName varchar(64) NOT NULL default '',
 Attribute varchar(32) NOT NULL default '',
 op char(2) NOT NULL DEFAULT '=',
 Value varchar(253) NOT NULL default '',
```

```
 PRIMARY KEY (id),
 KEY UserName (UserName(32))
) ;



#
# Table structure for table 'usergroup'
#

CREATE TABLE usergroup (
 UserName varchar(64) NOT NULL default '',
 GroupName varchar(64) NOT NULL default '',
 priority int(11) NOT NULL default '1',
 KEY UserName (UserName(32))
) ;

#
# Table structure for table 'radpostauth'
#

CREATE TABLE radpostauth (
 id int(11) NOT NULL auto_increment,
 user varchar(64) NOT NULL default '',
 pass varchar(64) NOT NULL default '',
 reply varchar(32) NOT NULL default '',
 date timestamp(14) NOT NULL,
 PRIMARY KEY (id)
```

```
) ;

##############################################################################
#
# The next table is commented out because it is not
# currently used in the server.
#

#
# Table structure for table 'dictionary'
#
#CREATE TABLE dictionary (
# id int(10) DEFAULT '0' NOT NULL auto_increment,
# Type varchar(30),
# Attribute varchar(64),
# Value varchar(64),
# Format varchar(20),
# Vendor varchar(32),
# PRIMARY KEY (id)
#);

#
# Table structure for table 'nas'
#
CREATE TABLE nas (
 id int(10) NOT NULL auto_increment,
 nasname varchar(128) NOT NULL,
```

```
 shortname varchar(32),
 type varchar(30) DEFAULT 'other',
 ports int(5),
 secret varchar(60) DEFAULT 'secret' NOT NULL,
 community varchar(50),
 description varchar(200) DEFAULT 'RADIUS Client',
 PRIMARY KEY (id),
 KEY nasname (nasname)
);

#
# Table structure for table 'radippool'
#
CREATE TABLE radippool (
 id     int(11) unsigned NOT NULL auto_increment,
 pool_name    varchar(30) NOT NULL,
 FramedIPAddress  varchar(15) NOT NULL default '',
 NASIPAddress   varchar(15) NOT NULL default '',
 CalledStationId  VARCHAR(30) NOT NULL,
 CallingStationID  VARCHAR(30) NOT NULL,
 expiry_time   DATETIME NOT NULL default '0000-00-00 00:00:00',
 username    varchar(64) NOT NULL default '',
 pool_key    varchar(30) NOT NULL,
 PRIMARY KEY (id)
);
```

2. Το ακόλουθο script χρησιμοποιείται για την δημιουργία των τυχαίων εγγραφών Radius χρηστών στη βάση δεδομένων της MySQL με τη χρήση του προγράμματος passgen όπως είδαμε στην άσκηση 3 στο Κεφάλαιο 6

```
#! /bin/bash
# Generate a batch of 1000 usernames and passwords
# for a Pre-Paid Card system

# This script is rather crude.
# The idea is that it will generate a new batch of
# username data each time you run it and that it will
# remember the last run, so usernames will be consecutive
# and won't repeat.

# Files:
# /tmp/usergenname: The username prefix, eg. 'S'
# /tmp/usergennum: The first username suffix, eg. '1000'
# /tmp/usergen.csv: Comma separated value file
# File format: username,password,sessiontime
# /tmp/usergen.sql: SQL insert command file.

# The session time defaults to 1h, change it below
# The number of passcodes to generate is set to 1000
# which should be enough for a batch of cards.

# Initialize the presets with:
# echo -n "1000">/tmp/usergennum
# echo -n "S">/tmp/usergenname
```

```
# Cleanup
rm -f /tmp/usergen.csv
rm -f /tmp/usergen.sql

# Session Time
TIME="1h"
echo "Session Time=$TIME"

# Read the last username
USERNAME="$(cat /tmp/usergenname)"
#echo Username=$USERNAME
USERNUM="$(cat /tmp/usergennum)"
echo Usernum=$USERNUM
echo First Username=$USERNAME$USERNUM

# Number of Passcodes to Generate
MAXCOUNT=1000
COUNT=1

echo "Generate $MAXCOUNT Usernames:"

while [ "$COUNT" -le $MAXCOUNT ]
do
  # Random generators usually have trouble in a script
  # Passgen also needs some help to reduce obvious repeats
  # Run it a random number of times and use the last value returned
```

```
  MAXSEED=$RANDOM
  let "MAXSEED %= 10"
  let "MAXSEED += 10"
  SEED=1
  while [ "$SEED" -le $MAXSEED ]
  do
    PASS=$(/usr/local/bin/passgen -g 1 -l 6)
    #echo Password=$PASS

    let "SEED += 1"
  done

  # Output results to the screen and to a CSV file
  echo $USERNAME$USERNUM,$PASS,$TIME
  echo $USERNAME$USERNUM,$PASS,$TIME >> /tmp/usergen.csv

  # Output results to a SQL command file
  echo "insert into usergroup (UserName,GroupName)
values (\"$USERNAME$USERNUM\",\"users\");" >> /tmp/usergen.sql
  echo "insert into radreply (UserName,Attribute,op,Value)
values         (\"$USERNAME$USERNUM\",\"Reply-Message\",\"=\",\"Hello        user
$USERNAME$USERNUM\");" >> /tmp/usergen.sql
  echo "insert into radreply (UserName,Attribute,op,Value)
values      (\"$USERNAME$USERNUM\",\"Session-Timeout\",\"=\",\"$TIME\");"      >>
/tmp/usergen.sql
  echo "insert into radcheck (UserName,Attribute,op,Value)
values       (\"$USERNAME$USERNUM\",\"User-Password\",\":=\",\"$PASS\");"      >>
/tmp/usergen.sql
```

```
  # Increment
  let "COUNT += 1"
  let "USERNUM += 1"
done

# Save last username in /tmp directory
echo $USERNAME > /tmp/usergenname
echo $USERNUM > /tmp/usergennum

# Comments
echo "Comments:"
echo "Quality Assurance: Give the output files a look over
before using them"
echo "Send file /tmp/usergen.csv to a ticket printer"
echo "Feed file /tmp/usergen.sql into the radius database"
echo "Example: # mysql -uroot -ppass -Dradius < /tmp/usergen.sql"

echo Done!
```

Παράρτημα Γ Ευρετήριο