ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ
ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ &
ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΚΑΤΕΥΘΥΝΣΗ ΔΙΚΤΥΟΚΕΝΤΡΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Extensible Authentication Protocol

Διπλωματική Εργασία

της

Ζωής Η. Μουτοπούλου

Επιβλέπων : Δρ. Χρήστος Ξενάκης

ΑΘΗΝΑ 2010

Cxe ¢ .z egŰš´ei æeg

Ű gcæi cegŰšeg Ű šeεŰŰšgcqæűŰæšæegŰűgŕ Ű3cæcodŰgəŰəCqdűšŰəı 3 Ł gůŰűqεŸŰý 3 Ű Ű3Ů šůŰűqεœeg Ű Ůšgu Ű gı

Þə Ŭ fɣŮšgəŰə ůgε´Ůȷ Ÿůε ŰŰŰ´eg ´Ůȷ šyGe3ŰŰə ůŰš´Ůȷfȷ3 Ůəúȷ Ÿcegз Ű q ůgɔɔȷŰűiyŰ əŰəŭŰš Ű3Ű´ȷcůɤ´Ůicegз Ů oůdε Ůgdyů Ůgŕ Ű šg Ű Ű3Ů šůŰűqεœeg Ű Ůšgu Ű gı

Αφιερωμένο…

στον παππου Χρήστο, στη γιαγιά Ζωή,

και στους γονεις μου…

Επίσης…

Ευχαριστώ τους φίλους που με παρότρυναν και με «ανέχτηκαν»…

τον Αλέξανδρο,

την Αρετή,

τη Βίκυ,

τη Δανάη,

την Ελενα,

και την Κωνσταντίνα

Περίληψη

Τα ασύρματα τοπικά δίκτυα διαθέτουν μια πληθώρα μεθόδων αυθεντικοποίησης χρησιμοποιώντας το EAP (Extensible Authentication Protocol) πρωτόκολλο. Στα πλαίσια της διπλωματικής αυτής, αφού αναλυθεί το παραπάνω πρωτόκολλο, θα υλοποιηθεί το εξής σενάριο:

Ο χρήστης ενός ασύρματου δικτύου θα αυθεντικοποιείται σε ασύρματο τοπικό δίκτυο, χρησιμοποιώντας το EAP πρωτόκολλο, το οποίο θα βασίζεται στο λογισμικό WIRE1.x. Ο χρήστης θα διαθέτει λειτουργικό σύστημα Windows XP/Vista ή Linux. Ένας RADIUS Server, θα αναλαμβάνει την αυθεντικοποίηση του χρήστη εκ μέρους του ασύρματου τοπικού δικτύου. Ο RADIUS Server θα βασίζεται στο λογισμικό FreeRADIUS και θα έχει εγκατεστημένο το λειτουργικό σύστημα Linux.

ς EAP (Extensible Authentication Protocol), RADIUS, WIRE1x, FreeRADIUS

Abstract

Wireless networks use a variety of authentication methods by using EAP (Extensible Authentication Protocol). In this project EAP and most of its methods will be analyzed. Furthermore the following scenario will be intergraded:

A wireless network user will be authenticated to the wireless network using EAP and WIRE1.x software. User may use either Windows or Linux OS. A RADIUS Server will be responsible for the authentication on behalf of the wireless network. The RADIUS Server will be based on freeRADIUS software running on Linux.

Keywords: EAP (Extensible Authentication Protocol), RADIUS, WIRE1x, Authentication, Wireless Networks Security, FreeRADIUS

Πίνακας περιεχομένων

1 Εισαγωγή

Σε μία εποχή που η ηλεκτρονική ανταλλαγή πληροφορίας διαδραματίζει κυρίαρχο ρόλο στη καθημερινότητα και πλέον χρησιμοποιείται σχεδόν σε κάθε δραστηριότητα του αναπτυγμένου κόσμου, από την εργασία και την εκπαίδευσή ως και την κοινωνική ζωή, κρίνεται απαραίτητο αυτή να είναι ανεξάρτητη από ενσύρματες υποδομές και να πραγματοποιείται με ασφάλεια.

Σε περιβάλλοντα ασύρματης διαδικτύωσης, αναφορικά με την ασφάλεια, είναι απαραίτητη η διαδικασία της αυθεντικοποίησης. Πρόκειται για τη διαδικασία μέσω της οποίας ένα λογικό υποκείμενο, στην περίπτωση ενός ασύρματου δικτύου, ένας ασύρματος σταθμός, παρέχει τις πληροφορίες που απαιτούνται προκειμένου να ελεγχθεί η ταυτότητα του. Ένα από τα πρωτόκολλά που έχουν αναπτυχθεί για την αυθεντικοποίηση είναι το EAP (Extensible Authentication Protocol), το οποίο θα παρουσιαστεί στη παρούσα διπλωματική εργασία.

Στο κεφάλαιο 2 περιγράφονται βασικές έννοιες για τα δίκτυα τηλεπικοινωνιών, όπως τα πρότυπα ISO OSI και TCP/IP. Αναλύονται επίσης αρχές της ασφάλειας υπολογιστών και δικτύων, καθώς και γνωστές επιθέσεις που αυτά μπορεί να δεχτούν.

Στο κεφάλαιο 3 παρουσιάζονται βασικά στοιχεία του Extensible Authentication Protocol, σύμφωνα με το RFC 3748, που θα βοηθήσουν στην κατανόηση της προτεινόμενης λύσης.

Στο κεφάλαιο 4 παρουσιάζεται αντίστοιχα το RADIUS (*Remote Authentication Dial In User Service*) πρωτόκολλο.

Στο κεφάλαιο 5 αναλύεται η λειτουργία των κυριότερων και πιο ευρέως χρησιμοποιούμενων μεθόδων του EAP.

Τέλος, στο κεφάλαιο 6 περιγράφεται η υλοποίηση του ζητούμενου σεναρίου. Παρουσιάζεται η εγκατάσταση και παραμετροποίηση του λογισμικού καθώς και παραδείγματα λειτουργίας με βάση το προδιαγεγραμμένο σενάριο.

2 Επιστημονικό υπόβαθρο

2.1 Δίκτυα Υπολογιστών και Ασύρματα Δίκτυα

Στις επόμενες δύο ενότητες παρουσιάζονται επιγραμματικά, καθώς περαιτέρω μελέτη τους θα ξέφευγε από το πλαίσιο της έρευνας του παρόντος εγγράφου, οι δύο σημαντικότερες αρχιτεκτονικές δικτύων, το θεωρητικό μοντέλο αναφοράς ISO OSI καθώς και το TCP/IP, ώστε να είναι δυνατό στη συνέχεια η αναφορά στα επίπεδα διαστρωμάτωσης τους (layers).

2.1.1 Open Systems Interconnection (OSI)

Το μοντέλο αυτό βασίζεται σε μία πρόταση του Οργανισμού Διεθνών Προτύπων (International Standard Organization, ISO), ως το ένα πρώτο βήμα στην προτυποποίηση των πρωτοκόλλων επικοινωνίας μεταξύ υπολογιστών.

Η διαστρωμάτωση σύμφωνα με το OSI έχει γίνει σε 7 επίπεδα, τα οποία ξεκινώντας από το κατώτερο είναι τα εξής:

Το φυσικό στρώμα (**Physical Layer**) αφορά τη μετάδοση bit μέσω ενός επικοινωνιακού διαύλου. Αναφορικά με τη σχεδίαση, το ζητούμενο είναι όταν από τη μία πλευρά στέλνεται bit 1, αυτό στη άλλη πλευρά να λαμβάνεται ως 1 και όχι ως 0.

Το στρώμα ζεύξης δεδομένων (**Data Link Layer**) έχει ως σκοπό του την μετατροπή του αναξιόπιστου μέσου μετάδοσης σε μία γραμμή ελεύθερη από σφάλματα. Αυτό επιτυγχάνεται με τον τεμαχισμό των δεδομένων στον πομπό σε πλαίσια δεδομένων (data frames) και την αποστολή τους σε σειρά και με την επεξεργασία των πλαισίων επαλήθευσης (acknowledgement frames) που λαμβάνονται από το δέκτη. Έργο του στρώματος αυτού είναι επίσης η επίλυση προβλημάτων που δημιουργούνται από κατεστραμμένα, χαμένα και διπλά πλαίσια. Για την περίπτωση αμφίδρομης μετάδοσης στο δίαυλο ή της κοινής χρήσης αυτού υπάρχει ένα ειδικό υπόστρωμα προσπέλασης στο μέσο μετάδοσης (**Medium Access Control, MAC**).

Στο στρώμα δικτύου (**Network Layer**) σημαντικό ζήτημα είναι αυτό της δρομολόγησης των πακέτων από την αφετηρία ως τον προορισμό. Στο network layer ανήκει επίσης ο έλεγχος συμφόρησης.

Βασική λειτουργία του στρώματος μεταφοράς (**Transport Layer**) είναι ο τεμαχισμός των δεδομένων που δέχεται από το υπερκείμενο στρώμα συνόδου (**Session Layer**) και ο τεμαχισμός του σε μικρότερες μονάδες η μεταβίβαση τους στο στρώμα δικτύου και η

διασφάλιση ότι αυτά φτάνουν στο άλλο άκρο σωστά. Εδώ καθορίζεται επίσης και ο τύπος της υπηρεσία που θα προσφερθεί στο session layer. Στο στρώμα αυτό επίσης δύναται να πολυπλεκτούν πολλές συνδέσεις μεταφοράς μέσα από την ίδια σύνδεση δικτύου. Τέλος υποστηρίζεται μηχανισμός έλεγχου ροής (flow control), ώστε να αποφεύγετε το φαινόμενο κατά το οποίο ένας γρήγορος host, πλημυρίζει έναν αργό με δεδομένα.

Το session layer επιτρέπει στους χρήστες διαφορετικών μηχανών την αποκατάσταση συνόδων μεταξύ τους. Μία σύνοδος επιτρέπει τη συνηθισμένη μεταφορά δεδομένων, όπως και το στρώμα μεταφοράς, αλλά παρέχει επίσης υπηρεσίες όπως η διαχείριση του ελέγχου του διαλόγου, η διαχείριση σκυτάλης (token management), και ο συγχρονισμός (**synchronization**). Ελέγχει τις συνόδους (δηλαδή τις ανταλλαγές δεδομένων) μεταξύ δύο υπολογιστών, του Α και του Β. Ξεκινά, διαχειρίζεται και τερματίζει τη σύνδεση μεταξύ μιας τοπικής και μιας απομακρυσμένης εφαρμογής.

Το επίπεδο παρουσίασης (**Presentation Layer**) ασχολείται με τη σύνταξη και την παρουσίαση της πληροφορίας, σε αντίθεση με τα κατώτερα επίπεδα που ασχολούνται με την ορθή και αξιόπιστη μεταφορά των bit. Επίσης μετασχηματίζει τα δεδομένα σε τυπική μορφή που την αναμένει το επίπεδο εφαρμογών

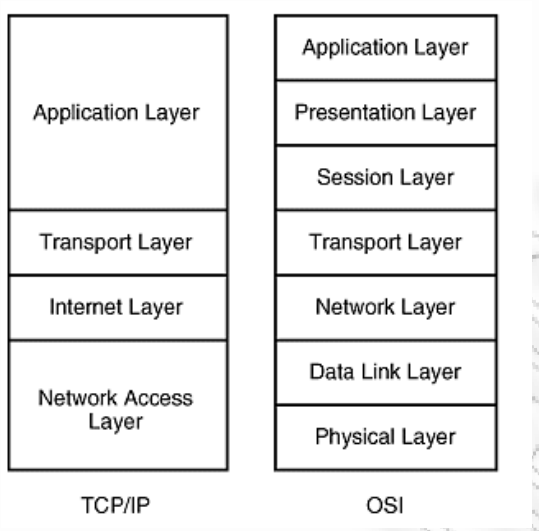
Το ανώτερο επίπεδό στο OSI είναι το επίπεδο εφαρμογής (**Application Layer**). Το επίπεδό αυτό περιλαμβάνει υπηρεσίες όπως η μεταφορά αρχείων, το ηλεκτρονικό ταχυδρομείο κτλ.

2.1.2 TCP / IP

Το TCP/IP, συχνά αναφέρεται και ως Internet Protocol Suite αποτελεί ένα σύνολο τηλεπικοινωνιακών πρωτοκόλλων που χρησιμοποιούνται από το Internet και άλλα παρόμοια δίκτυα.

Το TCP/IP, όπως το OSI, μπορεί να θεωρηθεί ως ένα σύνολο από επίπεδα:

Στο TCP/IP το φυσικό επίπεδο και το επίπεδο ζεύξης δεδομένων, τα οποία αναφέρονται παραπάνω, ομαδοποιούνται και συνιστούν το Επίπεδο πρόσβασης δικτύου (**Network Access Layer**). Εδώ περισσότερο χρησιμοποιούνται πρότυπα του Data Link και του Physical παρά καθορίζονται νέα αποκλειστικά για το TCP/IP.

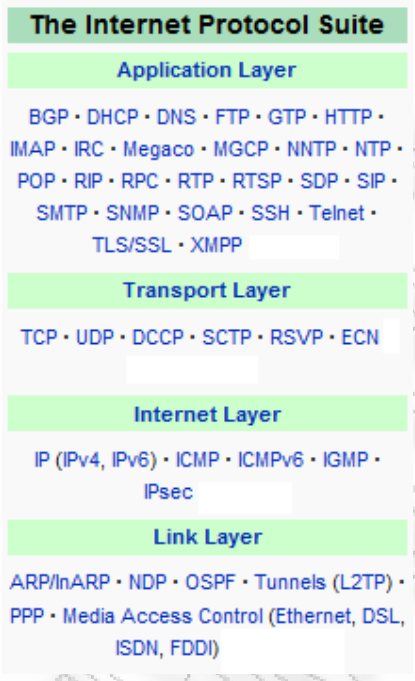


Εικόνα 1 Αντιστοιχίες επιπέδων μεταξύ των μοντέλων OSI και TCP/IP

Στο επίπεδο διαδικτύου (**Internet Layer**) του TCP/IP κυρίαρχο ρόλο διαδραματίζει το IP (Internet Protocol). Υπάρχουν επίσης επιπρόσθετα υποστηρικτικά πρωτόκολλα, όπως το ICMP (Internet Control Message Protocol), που χρησιμοποιείται για τη διευκόλυνση και τη διαχείριση της διαδικασίας δρομολόγησης. Σκοπός του είναι η μετάδοση datagrams, ακόμη και πέρα από τα όρια ενός δικτύου, από ένα αποστολέα σε έναν παραλήπτη που καθορίζεται από μία διεύθυνση δικτύου (IP διεύθυνση).

Ο σκοπός του επιπέδου μεταφοράς (**Transport Layer**) είναι όμοιος με αυτόν του OSI. Συνίσταται στον τεμαχισμό των δεδομένων από τα ανώτερα επίπεδα και η παράδοση τούς στα κατώτερα και αντίστροφα στην ανασύσταση τους. Στο TCP/IP χρησιμοποιούνται δύο πρωτόκολλα στο επίπεδο αυτό, το TCP και το UDP. Η κύρια διαφορά τους έγκειται στο ότι το TCP είναι αξιόπιστο (εξασφαλίζεται η ορθή μετάδοση των δεδομένων στο άλλο άκρο), ενώ αντίθετα το UDP είναι αναξιόπιστο (δε διαθέτει μηχανισμούς επαναποστολής χαμένων πακέτων ή μηχανισμούς τοποθέτησης των πακέτων στη σειρά κτλ.). Και τα δύο χρησιμοποιούνται ευρέως ανάλογα με το είδος της εφαρμογής που χρησιμοποιείται.

Το επίπεδο εφαρμογής (**Application Layer**) στο TCP/IP περιλαμβάνει τις λειτουργίες των επιπέδων εφαρμογής, παρουσίασης και συνόδου του OSI. Για το TCP/IP κάθε λειτουργία πάνω από το επίπεδο μεταφοράς αναφέρεται ως εφαρμογή. Για την περιγραφή της διαδρομής μέσω της οποίας επικοινωνούν δύο εφαρμογές χρησιμοποιούνται τα socket και τα ports. Παραδείγματα πρωτοκόλλων εφαρμογής είναι το HTTP, το FTP, το POP κτλ.



Εικόνα 2 Τα σημαντικότερα πρωτόκολλα που χρησιμοποιούνται ανά επίπεδο του TCP/IP

2.1.3 Ασύρματα Δίκτυα (IEEE 802.11)

Στο Network Access επίπεδο κυρίαρχο πρωτόκολλο, όταν αναφερόμαστε σε ασύρματη δικτύωση, αποτελεί το **IEEE 802.11**. Το πρωτόκολλο αυτό αρχικά υποστήριζε ρυθμούς μετάδοσης 1 και 2 Mbps στο φάσμα συχνοτήτων των 2,4 GHz.

Στο IEEE 802.11 ορίζονται η αρχιτεκτονική των ασύρματων τοπικών δικτύων, υπηρεσίες όπως η συσχέτιση (association), η επανασυσχέτιση (reassociation), η αυθεντικοποίηση (authentication) και η διασφάλιση του ιδιωτικού απορρήτου (privacy). Ορίζονται επίσης η δομή των πλαισίων, οι λειτουργίες διαμόρφωσης καθώς και ο αλγόριθμός WEP για την διασφάλιση του ιδιωτικού απορρήτου. Από το 1997 μέχρι και σήμερα το πρότυπο εξελίχτηκε και παρουσιάστηκαν εκδόσεις που υποστηρίζουν επιπρόσθετες δυνατότητες καθώς και υψηλότερες ταχύτητες μετάδοσης, που επιτυγχάνονται με αλλαγές στην χρησιμοποιούμενη διαμόρφωση. Οι αλλαγές αυτές δεν αποτελούν αντικείμενο μελέτης στο συγκεκριμένο έγγραφο και δε θα αναλυθούν περεταίρω.

Άξιες αναφοράς κατά την εξέλιξη του πρωτοκόλλου είναι οι παρακάτω διαφοροποιήσεις που παρουσιάστηκαν στις νεότερες εκδόσεις:

Με το 802.11b, γνωστό και ως Wi-Fi, προστέθηκαν ταχύτητες μετάδοσης 5 και 11 Mbps, ενώ με το 802.11g ταχύτητες 22 και 54 Mbps. Το πρότυπο 802.11 υποστηρίζει αυθεντικοποίηση των συσκευών και κρυπτογράφηση των δεδομένων, ενώ το 802.11b επιβάλει ένα ελάχιστο επίπεδο ασφαλείας και καθορίζει άλλα ασφαλέστερα επίπεδα. Απαιτείται η υποστήριξη ενός τουλάχιστο κλειδιού κρυπτογράφησης (WEP key) μήκους 40 bits. Τα δεδομένα κρυπτογραφούνται ενώ οι επικεφαλίδες μεταδίδονται χωρίς κρυπτογράφηση. Τόσο το 802.11b, όσο και το 802.11g χρησιμοποιούν τη μπάντα των 2,4 GHz σε αντίθεση με το 802.11a, το οποίο αν και επιτύγχανε υψηλούς ρυθμούς μετάδοσης λειτουργούσε στα 5 GHz, γεγονός που το καθιστούσε μη συμβατό με τις υπόλοιπες εκδόσεις.

Στο IEEE 802.11f περιγράφονται επίσης υπηρεσίες και πρωτόκολλα μεταξύ των σταθμών βάσης. Με την έκδοση αυτή επίσης καθορίζονται οδηγίες για την περιαγωγή των χρηστών σε διαφορετικούς σταθμούς βάσης.

Το IEEE 802.11e εισάγει την ποιότητα υπηρεσιών και την λειτουργικότητα για χρήση πολυφασικών εφαρμογών. Στο IEEE 802.11k προσθέτει λειτουργίες αναφορών. Ενώ στο IEEE 802.11h επεκτείνεται το IEEE 802.11a (5 GHz).

Στο IEEE 802.11i ενισχύεται η ασφάλεια του MAC επιπέδου και προδιαγράφονται πρωτόκολλα όπως το 802.1x, το TKIP και το CCMP.

Τέλος το 802.11n είναι το πιο πρόσφατο που εγκρίθηκε. Θεωρητικά οι συσκευές που υποστηρίζουν ασύρματη δικτύωση 802.11n μπορούν να συνδεθούν στα 300Mbps, 6 φορές μεγαλύτερο από το προηγούμενο πρότυπο, το 802.11g. Σε αυτό βοηθάει η υποστήριξη τεχνολογίας MIMO(multiple-input multiple-output) η οποία κάνει χρήση πολλαπλών κεραιών στο πομπό και το δέκτη, για όσο το δυνατόν μεγαλύτερη ταχύτητα.

Το 802.11n εκπέμπει στα 5GHz και φυσικά κρατάει τη συμβατότητα με δίκτυα 802.11b/g που εκπέμπουν στα 2.4GHz. Η εμβέλεια του σε εσωτερικούς χώρους υπολογίζεται (θεωρητικά) στα 90 μέτρα ενώ σε εξωτερικούς χώρους στα 182 μέτρα.

2.2 Θέματα ασφαλείας στα δίκτυα και επιθέσεις σε ασύρματα δίκτυα

2.2.1 Ασφάλεια Δικτύων – Μηχανισμοί ασφαλείας

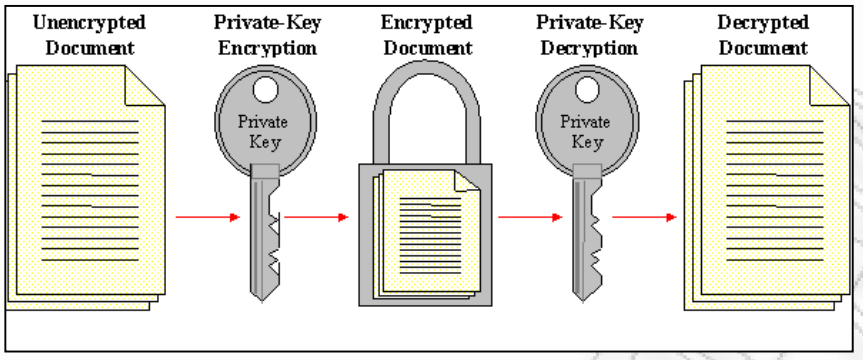
Οι τρεις βασικές ιδιότητες για την ασφάλεια των δικτύων υπολογιστών είναι η **εμπιστευτικότητα**, η **ακεραιότητα** και η **διαθεσιμότητα**. Και οι τρεις απαιτούνται ώστε ένα δίκτυο να μπορεί να θεωρηθεί ασφαλές. Η μη χρήση μηχανισμών προστασίας των παραπάνω καθιστά το δίκτυο τρωτό έναντι επιθέσεων. Οι επιτιθέμενοι συνήθως έχουν ως στόχο τους μία ή περισσότερες από τις παραπάνω ιδιότητες.

Εικόνα 3 Αρχές Ασφάλειας Δικτύων

Στόχος της εμπιστευτικότητας είναι η διασφάλιση του ότι οι ευαίσθητές πληροφορίες θα πρέπει να παραμένουν μυστικές και η πρόσβαση σε αυτές να είναι περιορισμένη σε όσους κρίνεται απαραίτητο και μόνο σε αυτούς. Στην ασφάλεια δικτύων είναι δυνατό να επιτευχτεί μέσω της κρυπτογράφησης των δεδομένων, της μετατροπής τους δηλαδή σε μια μη αναγνώσιμη μορφή.

Η κρυπτογράφηση μπορεί να είναι συμμετρική (ιδιωτικού κλειδιού) ή ασύμμετρη.

Στη **συμμετρική** χρησιμοποιείται το ίδιο κλειδί και ένας αλγόριθμος κρυπτογράφησης και αποκρυπτογράφησης.

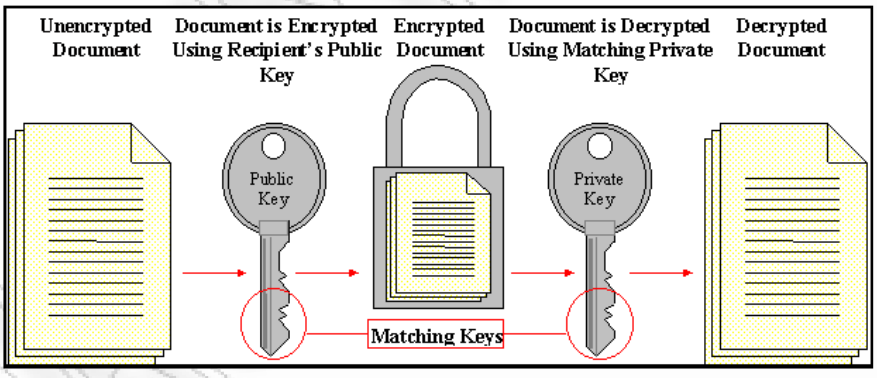


Οι συμμετρικοί αλγόριθμοι κρυπτογράφησης διακρίνονται σε δύο κατηγορίες:

- Κρυπτογράφημα Δέσμης (block cipher) - αλγόριθμοι που λειτουργούν σε μηνύματα συγκεκριμένου μήκους
- Κρυπτογράφημα ροής (stream cipher) – αλγόριθμοι που λειτουργούν σε ένα bit τη φορά.

Στην περίπτωση της συμμετρικής κρυπτογραφίας θα πρέπει να δίνεται ιδιαίτερη προσοχή στην ανταλλαγή των κλειδιών.

Η **ασύμμετρη** κρυπτογραφία χρησιμοποιείται για τη διασφάλιση τόσο της εμπιστευτικότητας, όσο και της ακεραιτότητας, της αυθεντικοποίησης και της μη αποποίησης ευθύνης.



Κάθε οντότητα σε ένα σύστημα ασύμμετρης κρυπτογραφίας διαθέτει ένα ζεύγος κλειδιών. Το δημόσιο κλειδί το οποίο δημοσιεύεται και χρησιμοποιείται για την κρυπτογράφηση των δεδομένων. Το ιδιωτικό κλειδί το οποίο παραμένει μυστικό και χρησιμοποιείται για την αποκρυπτογράφηση των μηνυμάτων.

Η **ακεραιότητα** στην ασφάλεια δικτύων έγκειται στο ότι ένα μήνυμα που μεταδίδεται δεν θα πειραχτεί. Δηλαδή δεν θα αφαιρεθεί κάποιο κομμάτι του ή θα αλλαχτεί. Βασικό μέτρο

για την πρόληψη ή την ανίχνευση επιθέσεων έναντι της ακεραιότητας είναι η παραγωγή ενός κρυπτογραφικού αθροίσματος (checksum) η επαλήθευση του οποίου εγγυάται πως το μήνυμα δεν έχει τροποποιηθεί. Επιπλέον διασφάλιση της ακεραιότητας είναι δυνατό να επιτευχθεί μέσω της χρήσης ψηφιακών υπογραφών.

Τέλος, η **διαθεσιμότητα** σημαίνει πως τα δεδομένα θα είναι διαθέσιμα ανά πάσα στιγμή ζητηθούν από κάποιο εξουσιοδοτημένο χρήστη ή διεργασία. Οι επιθέσεις έναντι στην διαθεσιμότητα αναφέρονται και ως Denial of Service (DoS) επιθέσεις. Τέτοιου είδους επιθέσεις καθιστούν αδύνατη τη πρόσβαση από το χρήστη σε συγκεκριμένη εφαρμογή ή υπηρεσία.

Υψηλή ασφάλεια σε ένα δίκτυο σημαίνει, όχι πως το δίκτυο δεν είναι ευαίσθητο σε επιθέσεις αλλά, πως οι μηχανισμοί που χρησιμοποιούνται παρέχουν υψηλού βαθμού ασφάλεια και δεν έχουν ακόμη «σπάσει» ακόμη. Η ασφάλεια τόσο των υπολογιστικών συστημάτων, όσο και των δικτύων συνεχώς εξελίσσονται. Ακόμη και οι πιο ισχυροί μηχανισμοί ασφαλείας απαιτείται να εξελίσσονται διαρκώς. Επίσης, αν οι παλαιότεροι μηχανισμοί ασφαλείας «σπάσουν», θα πρέπει να αναπτυχθούν νέοι.

2.2.2 Επιθέσεις σε ασύρματα δίκτυα

Τα ασύρματα δίκτυα αποτελούν στόχο επιθέσεων. Οι επιθέσεις δύναται να είναι παθητικές ή/και ενεργητικές. Οι παθητικές περιορίζονται στην συλλογή των σημάτων που αποστέλλονται, ενώ σε ενεργητικές επιθέσεις και ο επιτιθέμενος εκπέμπει σήματα. Στόχοι μίας επίθεσης είναι η αναγνώριση του δικτύου, η πρόκληση άρνησης υπηρεσιών, η απόκτηση δυνατότητας ανάγνωση καθώς και εγγραφής.

2.2.2.1 War Driving

Είναι αρκετά εύκολο, με χρήση μίας ασύρματης κάρτας δικτύου, μίας παραβολικής κεραίας πλέγματος και κατάλληλου λογισμικού (πχ. Netstumpler http://netstumpler.com , kismet http://kismetwireless.net), ο εντοπισμός και η καταγραφή των ασύρματων δικτύων μιας περιοχής. Σε αρκετές περιοχές έχουν παρατηρηθεί οδηγοί αυτοκινήτων που με επιπρόσθετη χρήση ενός δέκτη GPS έχουν χαρτογραφήσει ασύρματα τοπικά δίκτυα.

Εικόνα 4 Εξοπλισμός War Driving

Στην περίπτωση δε που τα ασύρματα δίκτυα δεν διαθέτουν μηχανισμούς ασφαλείας, τότε για την πρόσβαση σε ένα τέτοιο δίκτυο αρκεί μόνο μία ασύρματη κάρτα δικτύου.

2.2.2.2 Man- In-The-Middle

Η επίθεση του ενδιάμεσου ατόμου περιλαμβάνει την επίθεση «κρυφακούσματος» (eavesdropping) και την επίθεση «χειραγώγησης» (manipulating).

Κατά το eavesdropping ο επιτιθέμενος συλλέγει και εξετάζει τις πληροφορίες όπου εκπέμπονται. Κάτι τέτοιο είναι αρκετά ευνόητο αν αναλογιστεί κανείς πως τα όρια ενός ασύρματα δεν έχουν φυσική υπόσταση. Μέτρο προστασίας από το eavesdropping αποτελεί η κρυπτογράφηση είτε στο επίπεδο ζεύξης δεδομένων (Data Link Layer, DLL), είτε σε ανώτερα επίπεδα.

Κατά το manipulating ο επιτιθέμενος μεταβάλει τα δεδομένα που έχει υποκλέψει και τα επανεκπέμπει υποδυόμενος τον επιτιθέμενο. Αυτό επιτυγχάνεται με ARP spoofing μεθόδους. Ένας διαχειριστής συστήματος μπορεί να εντοπίσει και να καταπολεμήσει τέτοιου είδους παραβιάσεις παρακολουθώντας τον κατάλογο για παράξενες αλλαγές (π.χ. τήρηση ιστορικού των καταχωρήσεων και σύγκριση τους με τις πρόσφατες εγγραφές. Η διαδικασία μπορεί να γίνεται αυτοματοποιημένα με χρήση ειδικών εργαλείων.

2.2.2.3 Denial of Service

Οι επιθέσεις «Άρνησης Παροχής Υπηρεσιών» (Denial of Service, DoS) αποσκοπούν να πλήξουν την διαθεσιμότητα των πόρων ενός δικτύου από του εξουσιοδοτημένους χρήστες του.

Dos επίθεση μπορούν να πραγματοποιηθεί στο DLL (OSI Layer 2) με σκοπό κάποιο μηχάνημα του δικτύου να μην μπορεί να προσπελάσει το δίκτυο έστω κι αν είναι συνδεδεμένο σε αυτό. Οι επιθέσεις στο Layer 2 δεν πραγματοποιούνται συχνά σε ασύρματα δίκτυα καθώς συνήθως αποτρέπονται από τις συσκευές δικτύου. Στο επίπεδο του δικτύου (Network Layer, OSI Layer 3) οι επιθέσεις υλοποιούνται με την αποστολή περισσοτέρων δεδομένων από τα δεδομένα που το δίκτυο μπορεί να μεταδώσει. Αυτό προκαλεί την απόρριψη των πακέτων από τις συσκευές του δικτύου και τον φόρτο στην μονάδα επεξεργασίας και την κεντρική μνήμη των συσκευών του δικτύου. Στο επίπεδο μεταφοράς (Transport Layer, OSI Layer 4) μία DoS επίθεση συνίσταται στην αποστολή μεγάλου αριθμού αιτήσεων σύνδεσης (πχ SYN flooding) Στην περίπτωση αυτή αν και το δίκτυο είναι λειτουργικό το Λειτουργικό Σύστημα δεν μπορεί να παρακολουθήσει τις ενεργές συνδέσεις με αποτέλεσμα να μην είναι προσβάσιμες οι υπηρεσίες που παρέχονται από το μηχάνημα. Τέλος, επιθέσεις στο επίπεδο εφαρμογής υλοποιούνται με την αποστολή πολλαπλών αιτήσεων σε μία δικτυακή εφαρμογή. Η εφαρμογή υλοποιώντας τις συναλλαγές του επιτιθέμενου και η χρήση από άλλους χρήστες είναι αποτρέπεται.

Οι DoS επιθέσεις σε ασύρματα δίκτυα διαφοροποιούνται συγκριτικά με τα ενσύρματα στα εξής:

Physical Layer: Το φυσικό μέσο ενός ασύρματου δικτύου δεν είναι σαφώς ορισμένο και δεν περιορίζεται από τα φυσικά όρια ενός οργανισμού, έτσι οι επιτιθέμενοι μπορούν να πραγματοποιούν επιθέσεις και εκτός των ορίων του. Το φάσμα των συχνοτήτων που χρησιμοποιείται από τα ασύρματα δίκτυα είναι σαφώς ορισμένο από τα πρότυπα που ακολουθούνται σε αυτά (πχ IEEE 802.11), είναι λοιπόν αρκετά εύκολο κάποιος να γεμίσει το συγκεκριμένο εύρος με θόρυβο, προκαλώντας έτσι μια DoS επίθεση. Επίθεση σε φυσικό επίπεδο μπορεί επίσης να θεωρηθεί και η κλοπή συσκευών δικτύου.

DLL: Στο επίπεδο σύνδεσης δεδομένων λόγω της ελεύθερης πρόσβασης στο φυσικό μέσο υπάρχουν προοπτικές για DoS επιθέσεις. Για παράδειγμα αν χρησιμοποιείται μηχανισμός antenna diversity (χρήση πολλαπλών κεραιών για την αντιμετώπιση του προβλήματος της εξασθένισης του σήματος) και αυτός δεν είναι ορθά υλοποιημένος (οι περιοχές κάλυψης

των κεραιών δεν καλύπτουν ίδιους χώρους), τότε είναι δυνατό κάποιος να βγάλει εκτός δικτύου έναν εξουσιοδοτημένο χρήστη του δικτύου, απλά εκπέμποντας με χρήση της MAC του εξουσιοδοτημένου.

Network Layer: Στο επίπεδο δικτύου είναι δυνατές επιθέσεις, αν το δίκτυο είναι προσπελάσιμο από οποιονδήποτε. Ο επιτιθέμενος μπορεί να κατακλύσει το δίκτυο με κίνηση και να αποκλείσει τους υπόλοιπους από την πρόσβαση σε αυτό. Αν δε η ταχύτητα μεταφοράς είναι μικρή, αυτό μπορεί να συμβεί χωρίς απαραίτητα να υπάρχει κακόβουλη πρόθεση.

Disassociation και deauthentication επιθέσεις: Οι επιθέσεις αυτές έχουν ως σκοπό την παύση της συσχέτισης και τις αυθεντικοποίησης. Τέτοιες επιθέσεις στηρίζονται στην αδυναμία του IEEE 802.11 αναφορικά με τα πλαίσια διαχείρισης, χρησιμοποιώντας μηχανισμούς του ίδιου του πρωτόκολλου. Ο επιτιθέμενος μπορεί να επαναποστέλλει τα πλαίσια λήξης της συσχέτισης και της αυθεντικοποίησης που αποστέλλονται από το σταθμό βάσης με αποτέλεσμα ένας ασύρματος σταθμός να μένει εκτός δικτύου.

Επίθεση στο μηχανισμό αποφυγής συγκρούσεων: ένα πλαίσιο IEEE 802.11 περιέχει το πεδίο transmit duration, στο οποίο αναφέρεται ο χρόνος μετάδοσης του. Είναι ο χρόνος που αναμένει ένας ασύρματος σταθμός πριν προσπαθήσει να εκπέμψει. Αν λοιπόν ο επιτιθέμενος εκπέμπει τέτοια πλαίσιά, τότε οι υπόλοιποι σταθμοί σιγούν.

Επιθέσεις αυθεντικοποίησης

Ο έλεγχος της αυθεντικότητας αφορά δύο περιπτώσεις την ταυτότητα των χρηστών και την ταυτότητα των συστημάτων.

Αν και το IEEE 802.11 ενσωματώνει μεθόδους αυθεντικοποίησης και προστασίας του ιδιωτικού απόρρητου για τα ασύρματα δίκτυα, αυτές δεν έχουν τα επιθυμητά αποτελέσματα. Έτσι έχουν υιοθετηθεί νέοι μηχανισμοί αυθεντικοποίησης που βασίζονται στα 802.1x και EAP.

Επιθέσεις στο κοινό κλειδί: Κατά την αυθεντικοποίηση των δύο συμβαλλόμενων αποστέλλεται και από τους δύο ένα τυχαίο challenge text. Η κατοχή του WEP κλειδιού αποδεικνύεται με την κρυπτογράφηση του challenge text και την αποστολή του στον άλλο συμβαλλόμενο. Ο επιτιθέμενος μπορεί να συλλέξει πληροφορίες μίας τέτοιας επιτυχούς αυθεντικοποίησης και τότε δύναται να απαντά σωστά και να αυθεντικοποιείται. Συγκεκριμένα αυτό μπορεί να επιτευχθεί με γνώση του challenge text και του

κρυπτογραφημένου μηνύματος που αντιστοιχεί σε αυτό, εφαρμόζοντας την πράξη XOR, υπολογίζει το κλειδί ροής που αντιστοιχεί στο διάνυσμα αρχικοποίησης. Ο επιτιθέμενος γνωρίζοντας το διάνυσμα αρχικοποίησης και το κλειδί ροής μπορεί να αυθεντικοποιηθεί επιτυχώς.

ARP poisoning, MAC address Spoofing: Το ARP (Address Resolution Protocol ή Πρωτόκολλο Ανάλυσης Διευθύνσεων) αποτελεί το μηχανισμό που χρησιμοποιείται για την αντιστοίχιση των MAC διευθύνσεων με τις IP. Κάθε φορά που μία συσκευή συνδεδεμένη στο δίκτυο επιθυμεί να επικοινωνήσει με μία άλλη, εκπέμπει μία αίτηση ARP, την οποία λαμβάνουν όλες οι συσκευές που βρίσκονται στο δίκτυο. Κάθε συσκευή που λαμβάνει την αίτηση ελέγχει την IP της αίτησης. Αν η IP της αίτησης δεν είναι ίδια με την IP του υπολογιστή από τον οποίο έλαβε την αίτηση τότε αυτή αγνοείται, αλλιώς απαντά με την MAC διεύθυνση της. Σε κάποια λειτουργικά τηρείται τοπικά ένας MAC πίνακας και εφόσον υπάρχει εγγραφή σε αυτόν δεν αποστέλλεται ARP αίτημα. Σε μία επικοινωνία μεταξύ πελάτη – εξυπηρετητή ένας επιτιθέμενος, αν βρίσκεται στο ίδιο IP δίκτυο, δύναται να αποστείλει μια απόκριση ARP με Source IP την IP του εξυπηρετητή και source MAC τη δική του. Στην περίπτωση αυτή ο πελάτης θα αποστέλλει όλα τα frames, με προορισμό τον εξυπηρετητή, στον επιτιθέμενο καθώς θα ανανεωθεί λανθασμένα ο τοπικός πίνακας MAC.

2.3 Πιστοποιητικά

Ως πιστοποιητικό ορίζεται μία ηλεκτρονική δομή δεδομένων που χρησιμοποιείται για την αναγνώριση μίας οντότητας, πχ. ενός ατόμου , ενός εξυπηρετητή, μιας εταιρείας κτλ. Στις λειτουργίες ενός πιστοποιητικού περιλαμβάνεται επίσης η συσχέτιση της ταυτότητας αυτής με ένα δημόσιο και ένα ιδιωτικό κλειδί.

Τα πιστοποιητικά αντιστοιχίζουν τα δημόσια κλειδιά σε οντότητες, έτσι ώστε οι απομακρυσμένοι χρήστες που κάνουν χρήση του δημόσιου κλειδιού να είναι βέβαιοι ότι το αντίστοιχο ιδιωτικό κλειδί χρησιμοποιείται από το κατάλληλο άτομο ή σύστημα. Τα πιστοποιητικά αποτρέπουν την χρήση πλαστών δημόσιων κλειδιών που πιθανόν να χρησιμοποιηθούν σε περιπτώσεις πλαστοπροσωπίας. Μόνο ένα δημόσιο κλειδί που πιστοποιείται μέσω ενός πιστοποιητικού δύναται να λειτουργήσει με το αντίστοιχο ιδιωτικό κλειδί.

Η πιο διαδεδομένη μορφή πιστοποιητικών καθορίζεται από το διεθνές πρότυπο της ITU X.509 (RFC 3280), το Internet X.509 Public Key Infrastructure Certificate και το Certificate Revocation List (CRL) Profile.

Οι Αρχές πιστοποιητικών (Certificate authorities,CAs) είναι οντότητες που επικυρώνουν της ταυτότητες αυτές και εκδίδουν πιστοποιητικά. Ένας οργανισμός δύναται να έχει τη δική του αρχή για τα πιστοποιητικά ή να τα αγοράζει από μία τρίτη αξιόπιστη αρχή πιστοποιητικών.

Οι μέθοδοι που χρησιμοποιούνται για την επικύρωση της ταυτότητας εξαρτώνται από την πολιτική που εφαρμόζει μία αρχή πιστοποιητικών. Γενικά για την έκδοση ενός πιστοποιητικού μία CA θα πρέπει να διαπιστώσει την ταυτότητα της οντότητας και να υπογράψει ψηφιακά το πιστοποιητικό ώστε να εξασφαλιστεί η μη τροποποίηση του. Έτσι διασφαλίζεται ότι ένα πιστοποιητικό αντιστοιχεί ένα δημόσιο κλειδί σε μία οντότητα που ταυτοποιείται μέσω αυτού. Επιπρόσθετα εκτός από το δημόσιο κλειδί ένα πιστοποιητικό περιέχει το όνομα της οντότητας που αντιστοιχεί, την ημερομηνία λήξης του, το όνομα και το URI της αρχής που το εξέδωσε, ένα σειριακό αριθμό και μία ψηφιακή υπογραφή. Οι απομακρυσμένες οντότητες οφείλουν να επιβεβαιώνουν ότι ένα πιστοποιητικό είναι έγκυρο και σε ισχύ.

Συνήθεις τύποι πιστοποιητικών είναι οι ακόλουθοι:

Τα πιστοποιητικά της αρχής πιστοποιητικών (**Certificate Authority certificates**), τα οποία πιστοποιούν την ικανότητα υπογραφής πιστοποιητικών.

Τα πιστοποιητικά εξυπηρετητών Server certificates (**Server certificates**) τα οποία χρησιμοποιούνται σε ένα server, για να μπορέσει ο client να επιβεβαιώσει την εγκυρότητα μίας σύνδεσης και να δημιουργήσει ένα κρυπτογραφημένο κανάλι επικοινωνίας μεταξύ τους.

Τα πιστοποιητικά των Clients (**Client certificates**), μέσω τον οποίων ένας server επιβεβαιώνει την ταυτότητα του πελάτη και δίνει τη δυνατότητα στο χρήστη του να υπογράφει ψηφιακά ή να κρυπτογραφεί δεδομένα. Τέτοιου είδους πιστοποιητικά είναι ισχυρότερα αποδεικτικά ταυτότητας ενός client από τα στοιχεία σύνδεσης (username και password)

3 Extensible Authentication Protocol (EAP)

3.1 Γενικά

Το Extensible Authentication Protocol (EAP) αποτελεί ένα πλαίσιο αναφοράς για την αυθεντικοποίηση το οποίο υποστηρίζει πολλαπλές μεθόδους.

Τοποθετείται πάνω από το επίπεδο ζεύξης δεδομένων (DLL) όπως το PPP (Point –to – Point Protocol) και το IEEE 802, χωρίς να απαιτείται IP.

Το EAP διαθέτει τους δικούς του μηχανισμούς για απαλοιφή των διπλοτύπων και επαναποστολή. Δεν υποστηρίζεται κατάτμηση (fragmentation) από το ίδιο, αλλά μπορεί να υποστηριχθεί από τις μεθόδους του. Είναι δυνατό να χρησιμοποιεί τόσο σε αποκλειστικής χρήσης ζεύξεις (dedicated links), όσο και σε ζεύξης μεταγωγής κυκλώματος (switched circuits), σε ενσύρματες αλλά και ασύρματες ζεύξεις.

3.2 Ανταλλαγή μηνυμάτων στο EAP

Παρακάτω περιγράφεται η ακολουθία μηνυμάτων, όπως αυτή προδιαγράφεται στο EAP.

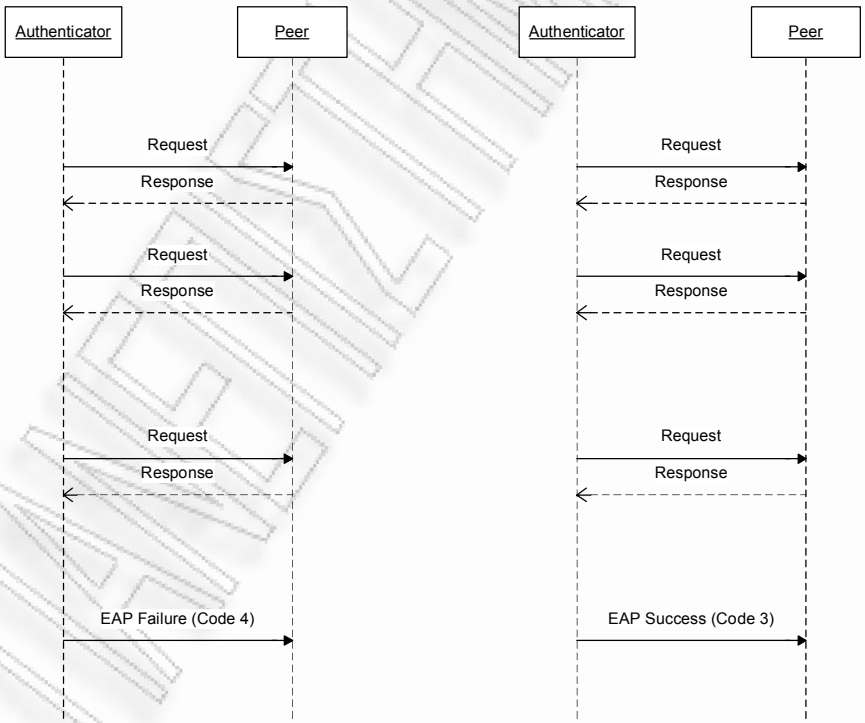
1. Ο authenticator στέλνει ένα Request (Αίτημα) για την αυθεντικοποίηση του peer (κόμβος). Το Request περιλαμβάνει ένα πεδίο Type (Τύπος), όπου αναφέρεται το τι ακριβώς ζητείται. Παραδείγματα τέτοιων Τύπων είναι Identity (Ταυτότητα) , MD5-challenge κτλ. Συνήθως αποστέλλεται ένα αρχικό Identity Request, αν και κάτι τέτοιο δεν είναι υποχρεωτικό και μπορεί να παραληφθεί. Για παράδειγμα δεν απαιτείται όταν καθορίζεται από το port (θύρα) που είναι το peer είναι συνδεδεμένο ή όταν μπορεί να ανακτηθεί από αλλού.

2. Το peer αποστέλλει ένα Response (Απάντηση) πακέτο, σε απάντηση ενός έγκυρου Request. Και το πακέτο αυτό περιέχει ένα πεδίο Type που ανταποκρίνεται στο πεδίο Type του Request.

3. Ο authenticator στέλνει ένα επιπλέον Request και το peer απάντα με ένα Response. Η ακολουθία αυτών των μηνυμάτων μπορεί να συνεχιστεί για όσο χρειαστεί. Το EAP χαρακτηρίζεται ως ‘lock step’ (κλειδωμένου βήματος)πρωτόκολλο, καθώς δεν επιτρέπεται η

αποστολή Requests (πλην του αρχικού) αν πρώτα δεν ληφθεί ένα έγκυρο Response. Είναι όμως δυνατό να γίνει συγκεκριμένος αριθμός επαναποστολών ενός Request, από τον authenticator, μετά από εύλογο χρονικό διάστημα. Ο authenticator δεν στέλνει μηνύματα Success ή Failure κατά τη διάρκεια των επαναποστολών ή όταν δεν λαμβάνει απαντήσεις από το peer.

4. Η ανταλλαγή μηνυμάτων συνεχίζεται μέχρι ο authenticator να μην μπορεί να αυθεντικοποιήσει τον κόμβο (μη αποδεκτές αποκρίσεις σε μία η περισσότερες αιτήσεις), περίπτωση στην οποία ο authenticator πρέπει να αποστείλει μήνυμα EAP Failure – code 4 (αποτυχία). Εναλλακτικά η ανταλλαγή μηνυμάτων συνεχίζεται μέχρι ο authenticator να ορίσει πως έχει γίνει επιτυχής αυθεντικοποίηση, στην περίπτωση αυτή ο authenticator πρέπει να αποστείλει μήνυμα EAP Success – code 3 (επιτυχία).



Εικόνα 5 Ανταλλαγή μηνυμάτων μεταξύ Authenticator και Peer

Πλεονεκτήματα:

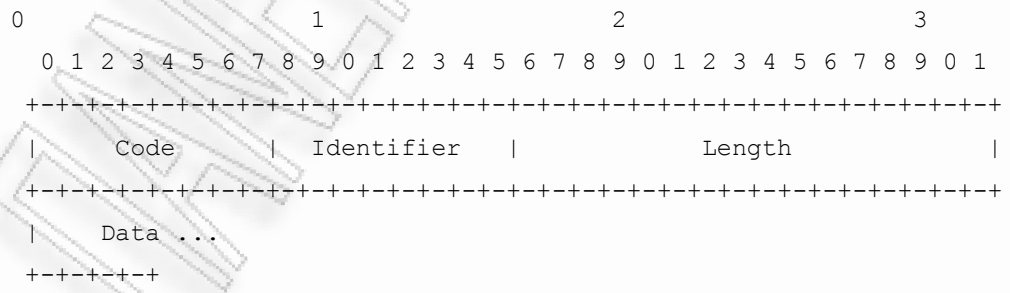
- Υποστήριξη πολλαπλών μηχανισμών αυθεντικοποίησης χωρίς την ανάγκη προκαθορισμού κάποιου
- Συσκευές Network Access Server (εξυπηρετητές πρόσβασης δικτύου, πχ switches ή access points) δεν είναι απαραίτητο να υποστηρίζουν την μέθοδο αυθεντικοποίησης καθώς δύνανται να λειτουργούν pass through agents για κάποιον εξυπηρετητή αυθεντικοποίησης.
- Ο διαχωρισμός του authenticator από τον backend authentication server απλοποιεί τη λήψη αποφάσεων για τη διαχείριση και πολιτική των στοιχείων πρόσβασης.

Μειονεκτήματα:

- Για χρήση στο PPP απαιτείται η προσθήκη ενός νέου τύπου αυθεντικοποίησης στο PPP LCP καθώς και εφαρμογή για την χρήση του. Απομακρύνεται δε από το προηγούμενο μοντέλο αυθεντικοποίησης PPP που διαπραγματεύεται έναν συγκεκριμένο μηχανισμό κατά τη διάρκεια το LCP. Όμοια εφαρμογές θα πρέπει να γίνουν στο switch ή το access point για την υποστήριξη του EAP.
- Σε υλοποιήσεις όπου ο authenticator είναι ξεχωριστός από τον backend authentication server, η ανάλυση ασφαλείας είναι πιο πολύπλοκη αν απαιτείται διανομή κλειδιών.

3.3 Δομή μηνυμάτων στο EAP

Η δομή ενός EAP πακέτου φαίνεται στο παρακάτω σχήμα:



Η μετάδοση πραγματοποιείται από αριστερά προς τα δεξιά.

Το πεδίο **Code** αποτελείται από οκτώ bit και χαρακτηρίζει τον τύπο του μηνύματος. Λαμβάνει τις τιμές 1 αν πρόκειται για Request, 2 για την περίπτωση του Response, 3 για Success και 4 αν πρόκειται για Failure.

Το πεδίο **Identifier** αποτελείται και αυτό από οκτώ bit και χρησιμοποιείται για το συσχετισμό Requests και Responses.

Το πεδίο **Length** έχει μήκος 16 bits και σε αυτό αναφέρεται το μήκος του συνολικού πακέτου EAP (Code, Identifier, Length και Data). Σε περίπτωση που το μήκος του πακέτου που λαμβάνεται ξεπερνά αυτό που ορίζεται στο πεδίο Length, τότε τα επιπλέον bit θεωρούνται ως padding bits που προστέθηκαν από το Data Link Layer και αγνοούνται. Αντίθετα, αν ένα πακέτο έχει μήκος μικρότερο από αυτό που περιγράφεται στο Length, απορρίπτεται.

Το πεδίο **Data** μπορεί να έχει μηδέν ή περισσότερα bytes. Περιεχόμενο καθορίζεται από το πεδίο Code.

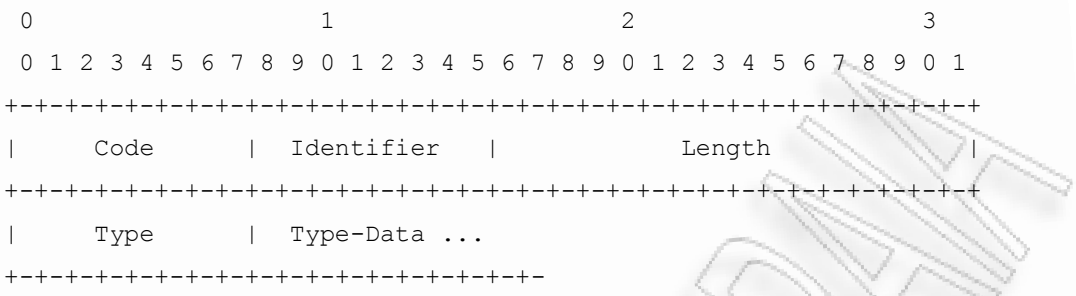
3.3.1 Requests και Responses

Κάθε Request στο τμήμα Data του έχει ένα πεδίο Type που υποδεικνύει τι αιτείται. Από τον authenticator αποστέλλονται πρόσθετα requests έως ότου να ληφθεί ένα έγκυρο response ή να πιθανόν να λήξει κάποιος counter ή να ληφθεί κάποια ένδειξη σφάλματος σε υποκείμενο layer.

Τα πακέτα που επανεκπέμπονται θα πρέπει να έχουν το ίδιο Identifier με το αρχικό, ώστε να μην συγχέονται με καινούρια.

Ο peer στέλνει ένα Response για κάθε έγκυρο Request που λαμβάνει. Τα response μηνύματα δεν επανεκπέμπονται, αν λήξει κάποιος timer. Αν ο peer λάβει έγκυρο αλλά διπλότυπο request τότε αποστέλλει ξανά το αρχικό response χωρίς να επεξεργαστεί το αίτημα εκ νέου. Τα requests επεξεργάζονται από τον peer με τη σειρά που λαμβάνονται και δεν ξεκινά η επεξεργασία του επόμενου αν δεν τελειώσει του προηγούμενου.

Παρακάτω φαίνεται η δομή των πακέτων Request και Response:



Το πεδίο **Code** συμπληρώνεται με 0 ή 1 για Request και Response αντίστοιχα.

Το πεδίο **Identifier** ενός Request παραμένει ίδιο για κατά την επαναμετάδοση ενώ αλλάζει για κάθε νέο Request. Στο Response θα πρέπει να αναφέρεται το identifier του Request στο οποίο αυτό αντιστοιχεί. Responds με άγνωστα για τον authenticator Identifiers θα απορρίπτονται.

Το πεδίο **Length** περιγράφει το μήκος του πακέτου.

Το πεδίο **Type** αποτελείται από οκτώ bits. Θα πρέπει σε κάθε Request ή Response να ανατίθεται ένα Type. Οι τιμές που δύναται να λάβει το Type θα περιγράφουν παρακάτω. Σε ένα Response μήνυμα το πεδίο Type ταιριάζει με αυτό του Request ή αντιστοιχίζεται σε ένα legacy ή σε ένα expanded Nak.

Το πεδίο **Type-Data** ποικίλει ανάλογα με το Type.

3.3.1.1 Αρχικοί EAP Request/Response Types

Παρακάτω φαίνονται τα EAP Types, που αρχικά ορίστηκαν στο RFC 3748 και που χρησιμοποιούνται κατά την ανταλλαγή μηνυμάτων Request και Response:

1	Identity
2	Notification
3	Nak (Response only)
4	MD5-Challenge
5	One Time Password (OTP)
6	Generic Token Card (GTC)
254	Expanded Types
255	Experimental use

Το πεδίο αυτό έχει μήκος ένα bit και καθορίζει τη δομή ενός πακέτου Request ή Response. Οι τρεις πρώτοι τύποι θεωρούνται ειδική περίπτωση. Οι υπόλοιποι ορίζουν ανταλλαγές αυθεντικοποίησης. Το Nak και το Expanded Nak (3 και 254 αντίστοιχα) θεωρούνται έγκυροι μόνο αν περιλαμβάνονται σε Responses. Στις υλοποιήσεις του EAP είναι υποχρεωτική η υποστήριξη των τεσσάρων πρώτων, οι οποίες αναφέρονται επιγραμματικά παρακάτω, και προτείνεται η υποστήριξη του Type 254.

Το Type **Identity** χρησιμοποιείται για το ερώτημα της ταυτότητας του peer. Το Response σε ένα τέτοιο Request έχει επίσης Type 1. Το πεδίο Type-Data είναι δυνατό να περιέχει ένα εμφανιζόμενο μήνυμα στο Request. Προτείνεται το Identity Response να χρησιμοποιείται για σκοπούς δρομολόγησης και για την επιλογή της EAP μεθόδου που θα χρησιμοποιηθεί. Οι μέθοδοι δε συνίσταται να περιλαμβάνουν μηχανισμούς, οριζόμενους από αυτές, για την αναγνώριση της ταυτότητας και να μην βασίζονται στο Identity Response.

Το Type **Notification** χρησιμοποιείται για την μεταβίβαση ενός μηνύματος από τον authenticator στον peer, που θα εμφανιστεί στην οθόνη του. Ο peer θα πρέπει να απαντήσει με ένα Notification Response, εκτός αν κάτι τέτοιο δεν επιτρέπεται από την EAP μέθοδο που χρησιμοποιείται. Ο peer θα να εμφανίζει στο χρήστη το μήνυμα ή να το αποθηκεύει στο log του. Το μέγιστό μέγεθος για το μήνυμα είναι 1015 bytes. . Το πεδίο Type-Data περιέχει το μήνυμα.

Το Type **Nak** (Type 3 για legacy Nak), χρησιμοποιείται μόνο σε μηνύματα Response και αποστέλλεται σε απάντηση ενός Request όπου το authentication Type δεν είναι αποδεκτό. Το Response περιέχει και το επιθυμητό authentication Type. Στο πεδίο Type-Data περιέχονται οι επιθυμητές από τον peer μέθοδοι.

Αντίστοιχα ένα **Expanded Nak** αποστέλλεται σε απόκριση ενός Request με Type 254. Περιλαμβάνει δε τους επιθυμητούς από τον peer τύπους αυθεντικοποίησης.

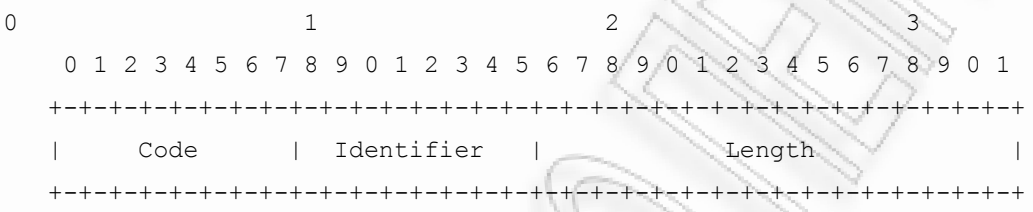
Το Type **MD5-Challenge** είναι ανάλογο με το PPP CHAP πρωτόκολλο που χρησιμοποιεί MD5 αλγόριθμο. Το Request περιέχει ένα μήνυμα πρόκληση (challenge) για το peer. Το Response θα πρέπει να είναι Type 3 (Nak), 254 ή 4. Για το Type-Data στην περίπτωση αυτή θα πρέπει κανείς να ανατρέξει στο PPP Challenge Handshake Authentication Protocol (RFC1994)

3.3.2 Success και Failure

Τα πακέτα Success και Failure δεν περιέχουν Data. Το Length συμπληρώνεται με 4.

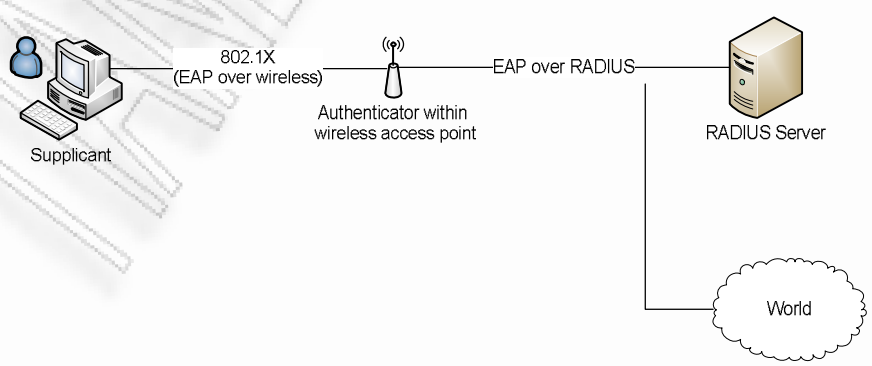
Το πεδίο **Code** συμπληρώνεται με 3 ή 4 για επιτυχή ή ανεπιτυχή αυθεντικοποίηση αντίστοιχα.

Το πεδίο **Identifier** χρησιμοποιείται για την αντιστοίχηση των αποκρίσεων στα Responses.



3.4 IEEE 802.1X

Το IEEE 802.1 είναι ένα πρότυπο για την μεταφορά του EAP σε ένα ενσύρματο ή ενσύρματο LAN. Μέσω του προτύπου αυτού τα EAP μηνύματα ενσωματώνονται σε Ethernet Frames. Σε ασύρματα περιβάλλοντα το 802.1X περιγράφει επίσης ένα τρόπο ώστε ο χρήστης και το access point να ανταλλάσσουν και να αλλάζουν τα κλειδιά κρυπτογράφησης. Μέσω των μηνυμάτων αυτών παρέχεται μία ασφαλής λύση στην αδυναμία που παρουσιάζεται στο 802,11, με την διαχείριση των WEP κλειδιών. Οι τρεις οντότητες που περιλαμβάνονται στο 802.1X είναι ο **supplicant** (ο χρήστης ή ο client που επιθυμεί να αυθεντικοποιηθεί), ο **authentication server** (ο εξυπηρετητής που αναλαμβάνει την αυθεντικοποίηση, συνήθως είναι ένας RADIUS server) και ο **authenticator** (η συσκευή ανάμεσα στα δύο παραπάνω, συνήθως ένα ασύρματο access point).



Στο 802.1X η πολυπλοκότητα του authenticator είναι μικρή και οι λειτουργίες να διαμοιράζονται σε supplicant και authentication server, γεγονός που είναι ιδανικό για τα ασύρματα access points που συνήθως έχουν μικρή μνήμη και υπολογιστική ισχύ.

Η συνήθης λειτουργία του 802.11X περιληπτικά συνοψίζεται στα εξής βήματα:

1. Ο Authenticator αποστέλλει ένα EAP-Request/Identity πακέτο στον Supplicant μόλις ανιχνεύσει μία ενεργή σύνδεση.
2. Ο Supplicant απαντά με ένα EAP-Response/Identity πακέτο με την ταυτότητα του, το οποίο περνά στον Authentication Server (RADIUS) ενθυλακωμένο σε RADIUS πρωτόκολλο.
3. Ο Authentication Server αποστέλλει μία πρόκληση (challenge) στον Authenticator, ο οποίος αποστέλλει μέσω EAPOL την πρόκληση στον Supplicant. Ο αριθμός των μηνυμάτων που θα ανταλλαχθούν σε αυτό το βήμα εξαρτάται από την μέθοδο αυθεντικοποίησης που χρησιμοποιείται. Οι μέθοδοι που μπορούν να χρησιμοποιηθούν μπορεί να απαιτούν αυθεντικοποίηση του πελάτη μόνο ή και των δύο. Για ασύρματα δίκτυα θεωρείται κατάλληλη μόνο η αυθεντικοποίηση και των δύο.
4. Ο Supplicant απαντά στον Authenticator την απόκριση του στην πρόκληση και αυτός με τη σειρά του την αποστέλλει στον Authentication Server.
5. Αν τα στοιχεία που στάλθηκαν είναι σωστά, τότε ο Authentication Server αποστέλλει ένα μήνυμα success, το οποίο περνά στον Supplicant, και ο Authenticator επιτρέπει στον Supplicant την πρόσβαση στο δίκτυο.

4 Remote Authentication Dial In User Service (RADIUS)

4.1 Γενικά

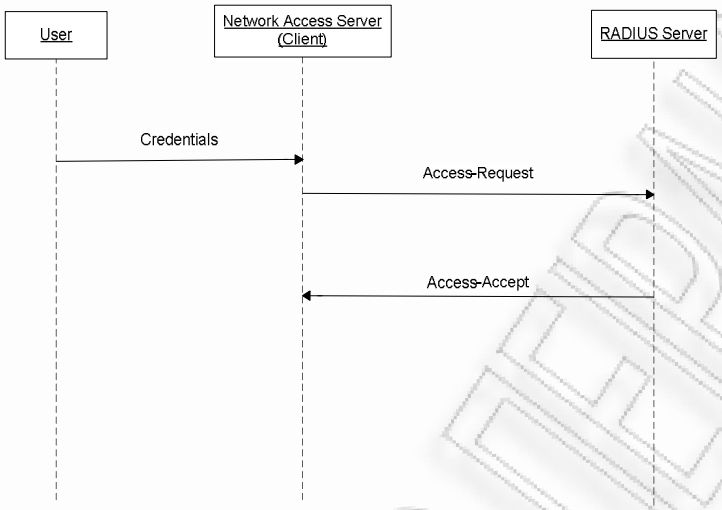
Το **Remote Authentication Dial In User Service** (**RADIUS**) αποτελεί ένα πρωτόκολλο για τη μεταφορά πληροφορίας σχετικής με την αυθεντικοποίηση, την εξουσιοδότηση και την ρύθμιση μεταξύ ενός Network Access Server που επιθυμεί την αυθεντικοποίηση των συνδέσεων του και ενός εξυπηρετητή αυθεντικοποίησης. Βασίζεται σε client – server αρχιτεκτονική, όπου ο NAS λειτουργεί ως client στον RADIUS Server. Ένας RADIUS Server είναι υπεύθυνος για τη λήψη και αιτημάτων σύνδεσης χρηστών, την αυθεντικοποίηση των χρηστών και την αποστολή των κατάλληλων ρυθμίσεων στον client ώστε τελικά ο χρήστης να λάβει κάποια υπηρεσία.

Χρησιμοποιείται το UDP ως πρωτόκολλό μεταφοράς και οι θύρες 1645 και 1646 ή της θύρες 1812 και 1813 για τα μηνύματα αυθεντικοποίησης και καταγραφής.

4.2 Ανταλλαγή μηνυμάτων RADIUS

Όταν ένας client είναι ορισμένος να χρησιμοποιεί RADIUS, τότε ο χρήστης του παρέχει την πληροφορία αυθεντικοποίησης στο client, συνήθως με μία φόρμα σύνδεσης, όπου θα πρέπει να συμπληρωθεί το όνομα χρήσης και ο κωδικός του χρήστη. Ο client μόλις λάβει την πληροφορία δημιουργεί και αποστέλλει ένα Access-Request (αίτημα σύνδεσης), το οποίο περιέχει πληροφορίες όπως το όνομα χρήστη και ο κωδικός, το ID του Client καθώς και το ID της πόρτας μέσω της οποίας έχει πρόσβαση ο χρήστης.

Το μήνυμα «Access Request» υποβάλλεται στον RADIUS server μέσω του δικτύου. Αν δε ληφθεί απάντηση σε εύλογο χρονικό διάστημα πραγματοποιείται επαναποστολή. Δύναται επίσης η αποστολή του μηνύματος σε εναλλακτικούς εξυπηρετητές.



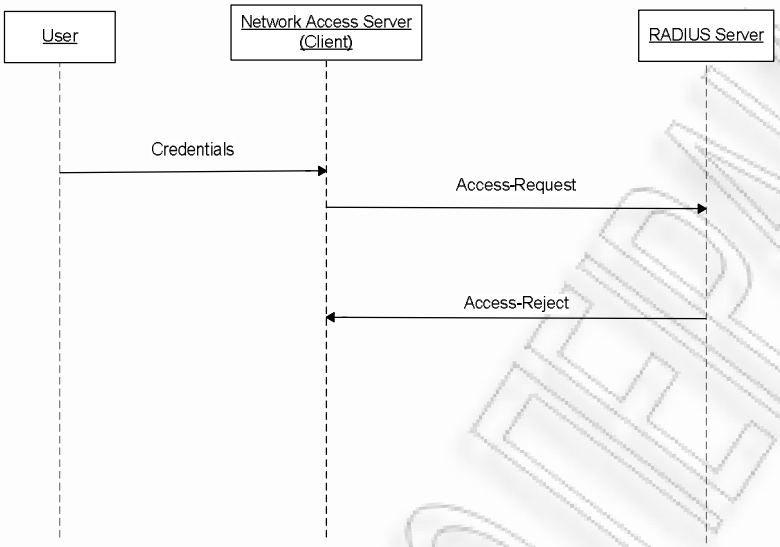
Εικόνα 6 Επιτυχής αυθεντικοποίηση RADIUS

Όταν ο RADIUS λάβει ένα αίτημα τότε επικυρώνει τον client. Ένα αίτημα από έναν Client για τον οποίο ο Εξυπηρετητής RADIUS δεν μοιραστεί ένα shared secret, θα πρέπει να απορρίπτονται. Αν πρόκειται για έγκυρο client τότε ο εξυπηρετητής ανατρέχει σε μία βάση δεδομένων με τους χρήστες για να βρει το χρήστη με το όνομα που ταιριάζει στο αίτημα. Η εγγραφή του χρήστη σε αυτήν την λίστα περιέχει τις απαιτήσεις που θα πρέπει να ικανοποιούνται ώστε να δοθεί πρόσβαση στο χρήστη. Θα πρέπει να γίνεται επιβεβαίωση του Password, αλλά υπάρχει δυνατότητα και για καθορισμό των clients ή των θυρών μέσω των οποίων θα επιτρέπεται η πρόσβαση στο χρήστη.

Ο RADIUS εξυπηρετητής δύναται να αποστέλλει αιτήματα και σε άλλους εξυπηρετητές. Σε αυτήν την περίπτωση λειτουργεί ως client.

Αν στο Access Request υπάρχουν χαρακτηριστικά Proxy-State, τότε αυτά θα πρέπει να αντιγράφονται αμετάβλητα και στο μήνυμα απόκρισης. Είναι δυνατό να προστεθούν και άλλα πριν ή μετά.

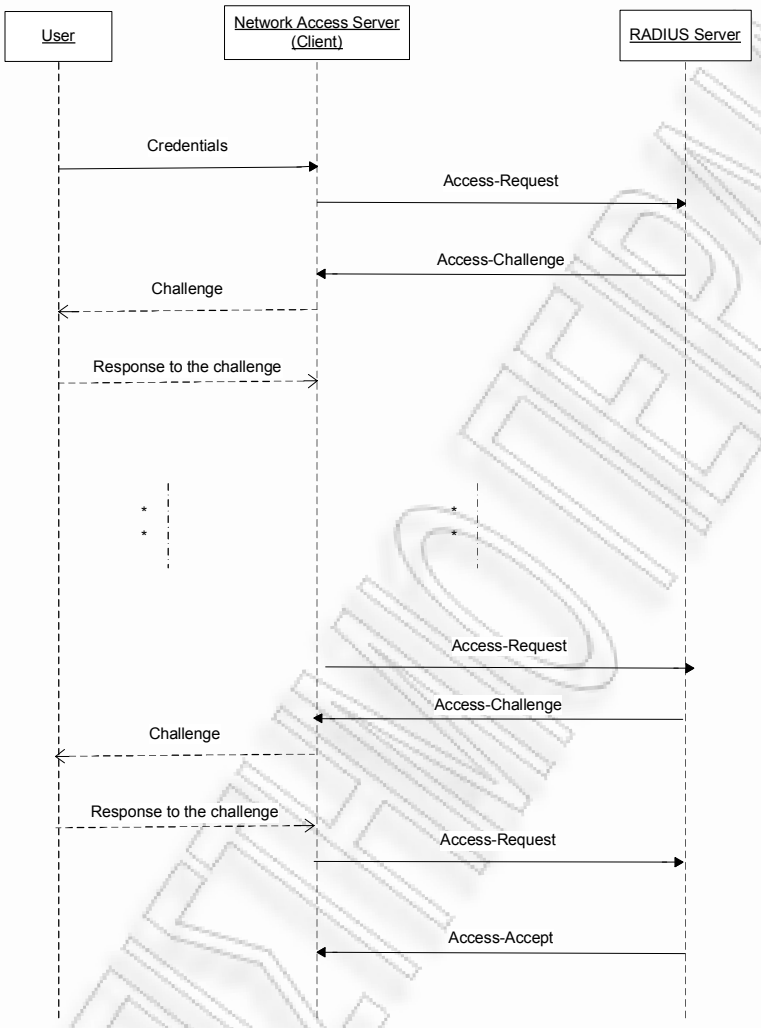
Στην περίπτωση που οι παραπάνω συνθήκες δεν ικανοποιούνται, δηλαδή ο χρήστης ή το συνθηματικό δεν είναι έγκυρα ή ο client ή θύρα προσπέλασης δεν είναι έγκυρες για το χρήστη, τότε αποστέλλετε μήνυμα Access- Reject.



Εικόνα 7 Ανεπιτυχής αυθεντικοποίηση RADIUS

Υπάρχει δυνατότητα αποστολής και μηνύματος προς το χρήστη. Στο Access-Reject μήνυμα δεν περιλαμβάνονται άλλα ορίσματα.

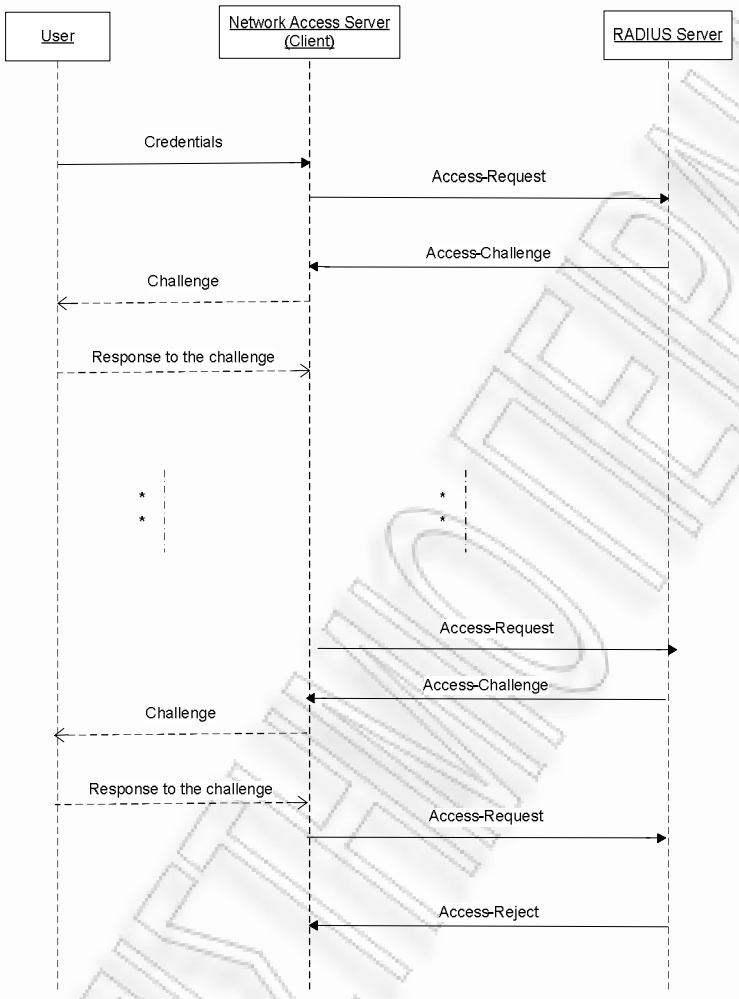
Σε περίπτωση που οι παραπάνω συνθήκες ικανοποιούνται και ο RADIUS επιθυμεί μία "πρόκληση" (challenge), στην οποία ο χρήστης θα πρέπει να αποκριθεί τότε αποστέλλεται ένα μήνυμα απόκρισης Access-Challenge. Πιθανόν να περιέχει ένα μήνυμα προτροπής προς το χρήστη για απόκριση στην πρόκληση, ίσως επίσης περιέχει και ένα όρισμα κατάστασης (State attribute).



Εικόνα 8 Επιτυχής RADIUS αυθεντικοποίηση με Access Challenge

Στην περίπτωση που ο client λάβει ένα μήνυμα Access challenge τότε εμφανίζει το μήνυμα που περιέχεται και ειδοποιεί το χρήστη να απαντήσει. Ο client επανυποβάλλει το Access Request με νέο ID και αντικαθιστώντας το User-Password όρισμα με την απόκριση κρυπτογραφημένη, και συμπεριλαμβάνοντας το όρισμα κατάστασης της πρόκλησης.

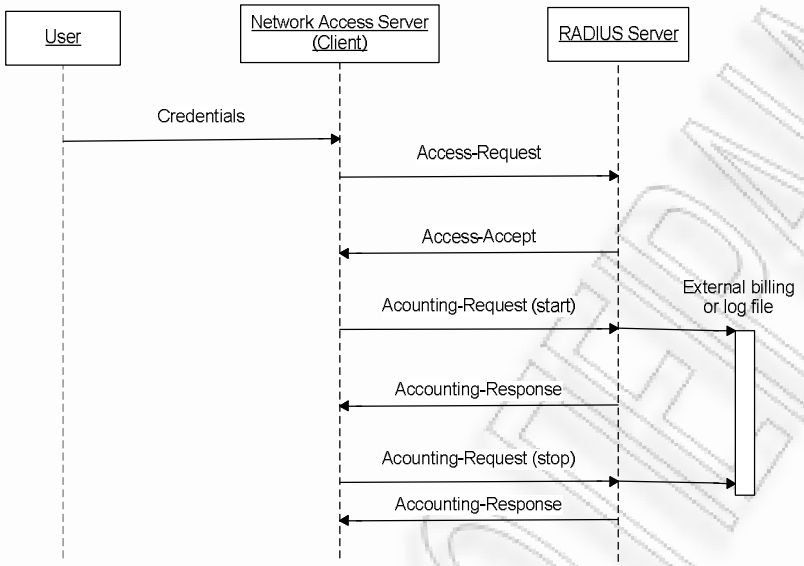
Ο εξυπηρετητής μπορεί να αποκριθεί στο νέο αυτό Access-Request με Access-Accept, Access-Reject, ή άλλο ένα Access-Challenge.



Εικόνα 9 Ανεπιτυχής αυθεντικοποίηση RADIUS με Access-Challenge

Αν ικανοποιούνται όλες οι απαιτήσεις τότε ο εξυπηρετητής αποστέλλει τη λίστα με τις «ρυθμίσεις» του χρήστη στο Access-Accept μήνυμα απόκρισης. Οι τιμές αυτές καθορίζουν τον τύπο της υπηρεσίας και τις απαραίτητες τιμές για την παράδοση της (πχ για service SLIP ή PPP περιέχει τιμές όπως η IP, η subnet mask, MTU κτλ.

Κατά την παραλαβή ενός Access-Accept πακέτου ο NAS πιθανόν να επιθυμεί να ενημερώσει τον RADIUS να εκκινήσει την τιμολόγηση του συγκεκριμένου χρήστη. Αυτό υλοποιείται μέσω ενός Accounting Request πακέτου προς τον εξυπηρετητή. Ο RADIUS Server ξεκινά την τιμολόγηση είτε επικοινωνώντας με ένα εξωτερικό σύστημα τιμολόγησης είτε καταγράφοντας την πληροφορία σε ένα δικό του αρχείο καταγραφής (log file).



Εικόνα 10Έναρξη και λήξη τιμολόγησης μέσω RADIUS

Αποστέλλεται επίσης ένα πακέτο Accounting Response προς τον πελάτη. Κατά το κλείσιμο μίας σύνδεσης από το χρήστη ή κατά την αποσύνδεση του για οποιοδήποτε λόγω, αποστέλλεται ένα μήνυμα Accounting Request, ώστε να διακοπεί η τιμολόγηση και ο RADIUS server αποκρίνεται με τη σειρά του με ένα Accounting Response.

4.3 Δομή μηνυμάτων στο RADIUS

Η δομή ενός RADIUS πακέτου φαίνεται παρακάτω:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Authenticator                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Στο πεδίο **Code** περιγράφεται ο τύπος του RADIUS πακέτου. Οι έγκυροι κωδικοί περιγράφονται παρακάτω. Αν ένα πακέτο δεν περιέχει έγκυρο Code, τότε αυτό απορρίπτεται.

1 Access-Request

2 Access-Accept

3 Access-Reject

4 Accounting-Request

5 Accounting-Response

11 Access-Challenge

12 Status-Server (experimental)

13 Status-Client (experimental)

255 Reserved

Τα Codes 4 και 5 περιγράφονται στο RFC 2139, που αναφέρεται στο RADIUS Accounting, ενώ τα 12 και 13 έχουν δεσμευτεί για άλλες πιθανές χρήσεις.

Το πεδίο **Identifier** χρησιμοποιείται για την αντιστοίχηση αιτημάτων και αποκρίσεων. Ένας RADIUS εξυπηρετητής δύναται να ανιχνεύσει ένα διπλό αίτημα που λαμβάνει, αν περιέχει την ίδια IP διεύθυνση, θύρα και Identifier και ληφθεί σε ένα μικρό σχετικά χρονικό διάστημα.

Στο πεδίο **Length** αναφέρεται το μήκος του συνολικού πακέτου. Αν ένα πακέτο είναι μεγαλύτερο από την τιμή που υπάρχεις στο length, τότε τα επιπλέον δεδομένα θεωρούνται padding και αγνοούνται. Αν είναι μικρότερο από το length, αγνοείται από το δέκτη.

Το πεδίο **Authenticator** αποτελείται από 16 octets. Η πιο σημαντική μεταδίδεται πρώτη. Αυτή η τιμή χρησιμοποιείται για την αυθεντικοποίηση της απάντησης από τον RADIUS εξυπηρετητή και χρησιμοποιείται επίσης και για τον αλγόριθμό απόκρυψης του κωδικού πρόσβασης.

Request Authenticator

Στα Access-Request πακέτα η τιμή του Authenticator είναι ένας τυχαίος 16Bit αριθμός που ονομάζεται Request Authenticator. Δεν θα πρέπει να είναι εύκολα προβλέψιμος και θα πρέπει να είναι μοναδικός για κάθε secret (ο κωδικός που μοιράζονται ο Client και ο εξυπηρετητής RADIUS).

Ο NAS και ο RADIUS Server μοιράζονται ένα secret. Αυτό ακολουθούμενο από το Request Authenticator μετατρέπεται με έναν μονόδρομο MD5 hash αλγόριθμο ακολουθία 16 byte η οποία, γίνεται XOR με το password του χρήστη και τοποθετείται στο πεδίο User-Password ενός Access Request πακέτου.

Response Authenticator

Η τιμή του πεδίου Authenticator στα Access-Accept, Access-Reject και Access-Challenge πακέτα αναφέρεται ως Response Authenticator. Περιέχει ένα MD5 hash που υπολογίζεται ως εξής: MD5(Code+ID+Length+RequestAuth+Attributes+Secret).

5 Οι πιο διαδεδομένες μέθοδοι στο EAP

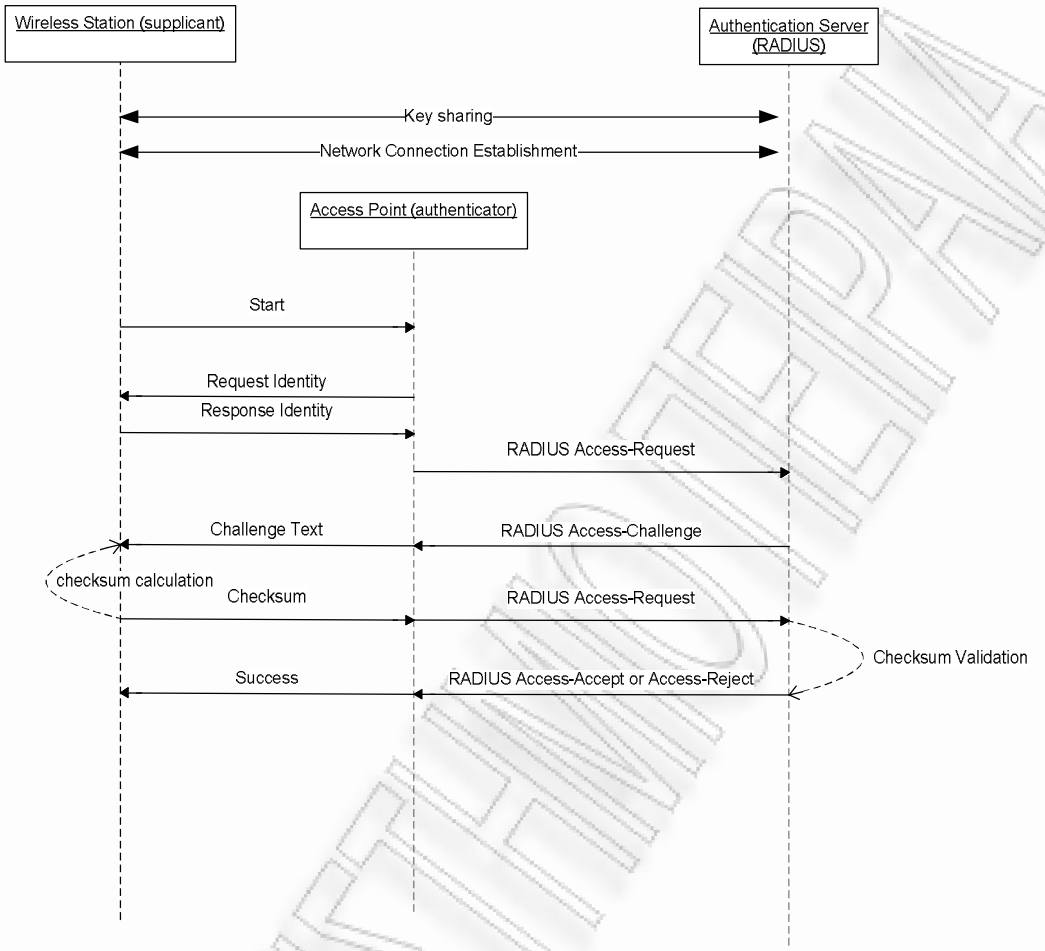
Οι μέθοδοι αυθεντικοποίησης του RADIUS κρίνονται περιορισμένες, μέσω όμως του Extensible Authentication Protocol μπορούμε να προσθέσουμε νέους μηχανισμούς αυθεντικοποίησης.

Παρακάτω αναλύονται οι πιο διαδεδομένες EAP μέθοδοι και ο τρόπος με τον οποίο επεκτείνουν το RADIUS. Για την πλήρη λίστα με τις EAP μεθόδους μπορεί κανείς να ανατρέξει στη διεύθυνση http://www.iana.org/assignments/eap-numbers .

5.1 EAP-MD5

Κατά τη μέθοδο αυτή ένας RADIUS εξυπηρετητής αυθεντικοποιεί τους σταθμούς επιβεβαιώνοντας το κρυπτογραφημένο κατά MD5 κωδικό του χρήστη.

Για τη λειτουργία του EAP-MD5 θα πρέπει ο εξυπηρετητής αυθεντικοποίησης και ο supplicant να διαμοιράζονται ένα κοινό κλειδί. Η διαδικασία διανομής του κοινού κλειδιού γίνεται χειροκίνητα και εκτός σύνδεσης στο πρώτο βήμα της μεθόδου. Στη συνέχεια εγκαθίσταται η σύνδεση. Στο επόμενο βήμα γίνεται ανταλλαγή μηνυμάτων που περιέχουν την ταυτότητα του supplicant. Ο εξυπηρετητής αυθεντικοποίησης αποστέλλει ένα συνθηματικό διέλευσης. Ο supplicant υπολογίζει την σύνοψη του συνθηματικού διέλευσης με χρήση του κοινού κλειδιού και το απαντά. Όταν ο εξυπηρετητής λάβει τη σύνοψη την επαληθεύει υπολογίζοντας την και ενημερώνει για την επιτυχία ή την αποτυχία της αυθεντικοποίησης.



Εικόνα 11 Διαδικασία αυθεντικοποίησης EAP-MD5

Αποτελεί μία απλή λύση για ενσύρματα τοπικά δίκτυα όπου η πιθανότητα επιθέσεων sniffing ή ενεργών επιθέσεων είναι μικρή. Σε περιπτώσεις ασύρματων δικτύων καθώς και δημόσιων ενσύρματων, θα πρέπει να αποφεύγεται καθώς κρίνεται επισφαλής για τις περιπτώσεις αυτές.

5.2 LEAP

Το LEAP (Lightweight EAP) αναπτύχθηκε από τη Cisco και χρησιμοποιεί mutual authentication (απαιτείται δηλαδή αυθεντικοποίηση τόσο του access point όσο και του χρήστη πριν επιτραπεί η πρόσβαση στο δίκτυο). Τα κλειδιά αλλάζουν δυναμικά, σε κάθε

σύνοδο ώστε να αποτρέπονται οι επιτιθέμενοι να συλλέγουν πληροφορίες που μπορεί να χρησιμοποιηθούν για την αποκωδικοποίηση των μηνυμάτων. Χρησιμοποιούν δε ένα κοινό μυστικό (shared secret), συνήθως τον κωδικό του χρήστη, για την αυθεντικοποίηση της επικοινωνίας μεταξύ του χρήστη και του access point. Απαιτείται από το access point υποστήριξη Cisco VSA, καθώς χρησιμοποιούνται για την διανομή κλειδιών WEP, καθώς και υποστήριξη του αλγόριθμου LEAP για την παραγωγή τους. Ο τρόπος λειτουργίας του περιγράφεται συνοπτικά παρακάτω:

Αρχικά ο client αποστέλλει ένα αίτημα σύνδεσης προς το access point. Το access point αποκρίνεται με ένα EAP request και ζητά την ταυτότητα του client. Ο client αποκρίνεται στέλνοντας ένα EAP response, που περιέχει την ταυτότητα του, στο access point. Στη συνέχεια η ταυτότητα προωθείται στο RADIUS εξυπηρετητή μέσω ενός μηνύματος Access-Request. Ο RADIUS Server απαντά με ένα Access-Challenge Που προωθείται στον client μέσω ενός μηνύματος EAP Request. Ο client απαντά με ένα hash του password και άλλων πιθανών στοιχείων σύνδεση στο access point και αυτό προωθεί με τη σειρά του τα στοιχεία μέσω ενός Access Request στο RADIUS. Ο RADIUS υπολογίζει το hash βάση των στοιχείων που έχει στη βάση του και το συγκρίνει με το πεδίο με το challenge value. Αν είναι ίδια ενημερώνεται το access point, το οποίο με τη σειρά του ενημερώνει τον client.

Στη συνέχεια ο client αποστέλλει ένα challenge μήνυμα στο access point με σκοπό την αυθεντικοποίηση του δικτύου. Το access point επιστρέφει ένα hash με τα στοιχεία σύνδεσης και την challenge value του client. Αν η αυθεντικοποίηση του δικτύου είναι επιτυχής ο client ενημερώνει το RADIUS και αυτός αποστέλλει ένα Access-Accept στο access point. Το access point ανοίγει τη πρόσβαση στον client.

Η ταυτότητα όμως του σταθμού και το ο κωδικός πρόσβασης είναι εκτεθειμένα σε επιθέσεις, κατά τις οποίες οι επιτιθέμενοι χρησιμοποιούν sniffers και εργαλεία dictionary επιθέσεων. Η μεγάλη του αδυναμία έγκειται στην χρήση MSCHAP v1, χωρίς ασφάλεια κρυπτογράφησης για την αυθεντικοποίηση. Ο επιτιθέμενος θα πρέπει να κρυφακούσει το challenge text κατά την απάντηση της LEAP αυθεντικοποίησης. Στη συνέχεια προσπελαύνει όλες τις καταχωρίσεις ενός ευρετηρίου για να εντοπίσει αυτή που ταιριάζει. Ο επιτιθέμενος μπορεί πλέον να υποδυθεί τον ασύρματο σταθμό. Μέτρο για την προστασία κατά τέτοιων επιθέσεων αποτελεί η χρήση δύσκολων συνθηματικών που δεν βγάζουν νόημα και η συχνή αλλαγή τους.

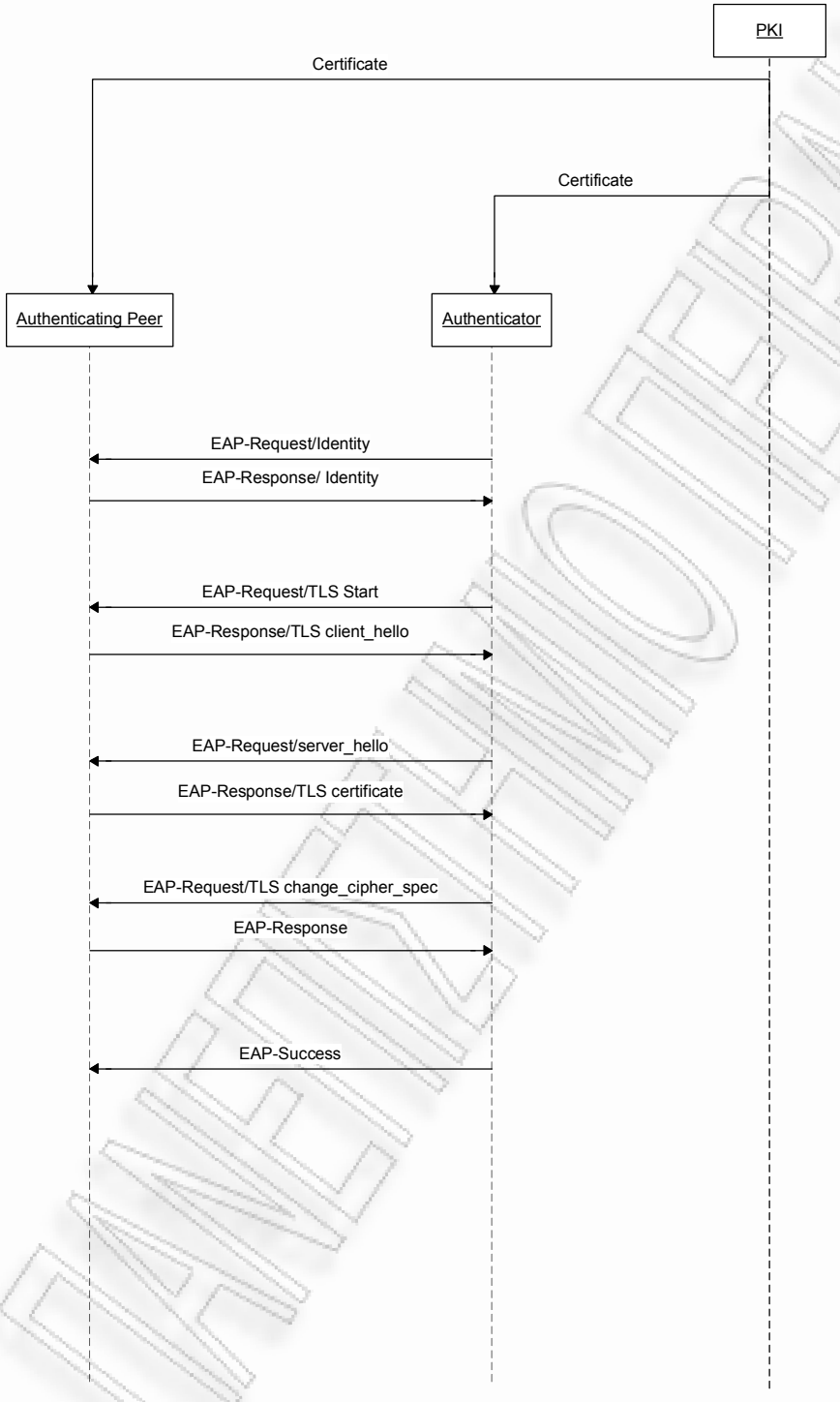
5.3 EAP- TLS

Η μέθοδος EAP-TLS (EAP with Transport Layer Security) είναι από τις ασφαλέστερες επιλογές στα ασύρματα LANs. Τόσο ο σταθμός όσο και ο RADIUS εξυπηρετητής πρέπει να αποδείξουν την ταυτότητα τους μέσω κρυπτογραφίας δημόσιου κλειδιού(ψηφιακά πιστοποιητικά ή έξυπνες κάρτες). Η ανταλλαγή αυτή εξασφαλίζεται μέσω ενός κρυπτογραφημένου TLS tunnel, γεγονός που καθιστά τη μέθοδο ιδιαίτερα ανθεκτική σε επιθέσεις Man In the Middle και dictionary. Χρησιμοποιείται κυρίως από μεγάλους οργανισμού που χρησιμοποιούν Windows λειτουργικά συστήματα και έχουν υιοθετήσει πιστοποιητικά. Το TLS ορίζεται από δύο επίπεδα, το TLS πρωτόκολλό εγγραφής (TLS Record Protocol) και το TLS πρωτόκολλο χειραψίας (TLS Handshake Protocol).

Η λειτουργία του περιγράφεται από τα εξής βήματα:

Μοιράζονται τα απαραίτητα πιστοποιητικά, μέσω μια υποδομής δημόσιου κλειδιού σε client και Server, με σκοπό την αυθεντικοποίηση των δύο μερών. Στη συνέχεια εγκαθιδρύεται μια TCP σύνδεση. Ο Server αποστέλλει ένα EAP Request ρωτώντας την ταυτότητα του client και αυτός αποκρίνεται με τη σειρά του αποστέλλοντας ένα EAP Response με την ταυτότητα του. Ο EAP Server μόλις ενημερωθεί για την ταυτότητα του peer θα πρέπει να αποκριθεί με ένα EAP- TLS /Start Packet, ένα EAP Request δηλαδή με EAP-Type ορισμένο ως EAP-TLS, το Start (S) bit ενεργό και χωρίς Data.

Ο peer θα απαντήσει με ένα EAP Response με EAP-Type=EAP-TLS. Το πεδίο Data περιέχει ένα TLS client_hello μήνυμα. Οι προδιαγραφές του cipher θα είναι TLS_NULL_WITH_NULL_NULL και null συμπίεση. Αυτό το τρέχων cipher θα παραμείνει ίδιο μέχρι να σταλεί μήνυμα με cipher change. Το client hello περιέχει επίσης το TLS version number του peer,ένα sessionId, ένα τυχαίο αριθμό και τους κρυπταλγόριθμούς που υποστηρίζει. Ο EAP server θα αποκριθεί με ένα EAP-Request με EAP-Type=EAP-TLS. Το Data πεδίο του ενθυλακώνει μια ή περισσότερες TLS εγγραφές, που περιέχουν ένα TLS server_hello handshake μήνυμα, πιθανόν ακολουθούμενο από ένα TLS certificate, ένα server_key_exchange, ένα certificate_request, ένα server_hello_done και/ή finished handshake messages, και/ή ένα TLS μήνυμα με τις προδιαγραφές του cipher. Το server_hello handshake μήνυμα περιέχει την έκδοση του TLS, έναν τυχαίο αριθμό, ένα sessionId, και τους κρυπταλγόριθμούς που υποστηρίζει.



Εικόνα 12 Επιτυχής EAP-TLS mutual authentication

Ο client θα πρέπει να αποκριθεί με ένα EAP – Response, το οποίο για την περίπτωση που το Request δεν αναφέρεται στη συνέχεια κάποιας προηγούμενης συνόδου, θα πρέπει να

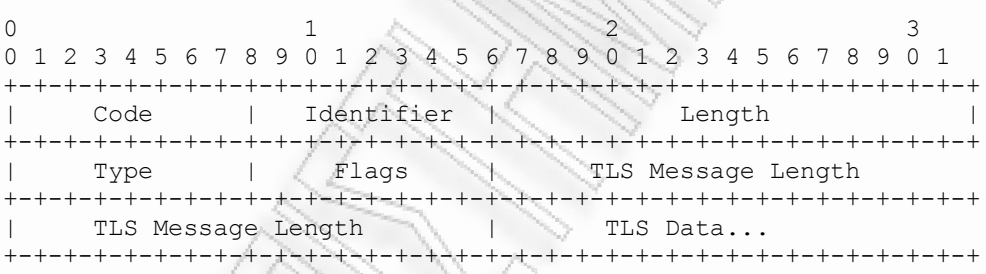
ενθυλακώνει μία η περισσότερες TLS εγγραφές που περιλαμβάνουν το TLS client_key_exchange, το change_cipher_spec, και τα μηνύματα finished. Σε κάποιες περιπτώσει ο peer πρέπει επιπρόσθετα να αποστείλει το certificate και το μήνυμα certificate_verify. Περιλαμβάνεται ένα πιστοποιητικό για το δημόσιο κλειδί της υπογραφής του peer και υπογράφεται από τον peer.

Κατά τη λήψη ενός τέτοιου πακέτου ο EAP Server επιβεβαιώνει το πιστοποιητικό και την ψηφιακή υπογραφή, αν απαιτηθεί.

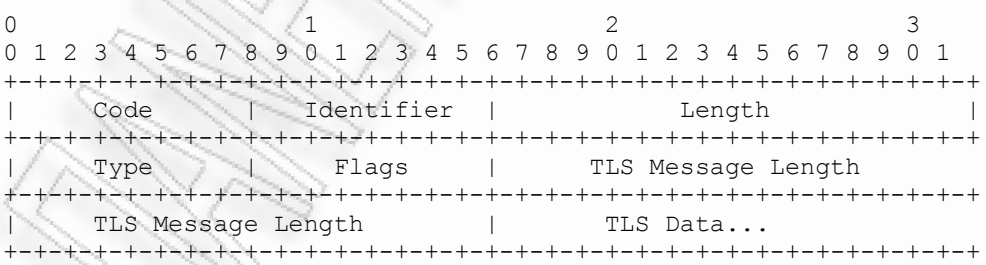
Στην περίπτωση που το server_hello περιέχει μήνυμα που αναφέρεται στην συνέχιση προηγούμενης συνόδου, τότε ο peer αποστέλλει μόνο change_cipher_spec και finished handshake μηνύματα.

Παρακάτω φαίνεται η δομή των πακέτων EAP-TLS Request και EAP-TLS Response:

EAP-TLS Response Packet:



EAP-TLS Response Packet:



Αδυναμία της παραπάνω μεθόδου αποτελει το γεγονός πως είναι δυνατό να αποκαλυφτεί μέσω sniffing η ταυτότητα του σταθμού.

5.4 EAP-TTLS

Το EAP-TTLS (EAP- Tunneled TLS) αποτελεί επέκταση του EAP-TLS. Χρησιμοποιεί πιστοποιητικά για την αυθεντικοποίηση του διακομιστή στον πελάτη και ένα πιο απλό μηχανισμό αυθεντικοποίησης για την αυθεντικοποίηση του πελάτη στο διακομιστή. Δε απαιτείται έτσι δημιουργία και διαχείριση πιστοποιητικών για τους πελάτες, καθώς αυτοί δεν τα χρειάζονται. Η αυθεντικοποίηση του πελάτη λαμβάνει χώρα μέσω άλλου μηχανισμού αυθεντικοποίησης στο ασφαλές κανάλι. Διακρίνονται δύο εκδόσεις το EAP-TTLS (γνωστό και ως EAP-TTLSv0) και το EAP-TTLSv1.

Η EAP-TTLS επικοινωνία συνίσταται σε δύο φάσεις, το TLS handshake και το TLS tunnel. Κατά τη διάρκεια της πρώτης χρησιμοποιείται TLS για την αυθεντικοποίηση του TTLS server στον client. Αποτέλεσμα της φάσης αυτής είναι η ενεργοποίηση ενός συνόλου cipher που επιτρέπει στη δεύτερη φάση να εκτελεστεί ασφαλώς χρησιμοποιώντας το επίπεδο TLS record. Ο βαθμός ασφαλείας της δεύτερης φάσης εξαρτάται από το σύνολο που θα οριστεί.

Στη δεύτερη φάση χρησιμοποιείται το TLS record επίπεδο για το tunneling της πληροφορίας που ανταλλάσσεται μεταξύ client και TTLS server για τις όποιες λειτουργίες αυτοί μπορεί να εκτελούν. Τέτοιες λειτουργίες είναι οι εξής : user authentication, client integrity validation, negotiation of data communication security capabilities, key distribution, communication of accounting information κτλ. Η πληροφορία μεταξύ client και TTLS server ανταλλάσσεται μέσω attribute-value pairs (AVPs) συμβατά με το RADIUS και το Diameter.

5.5 PEAP

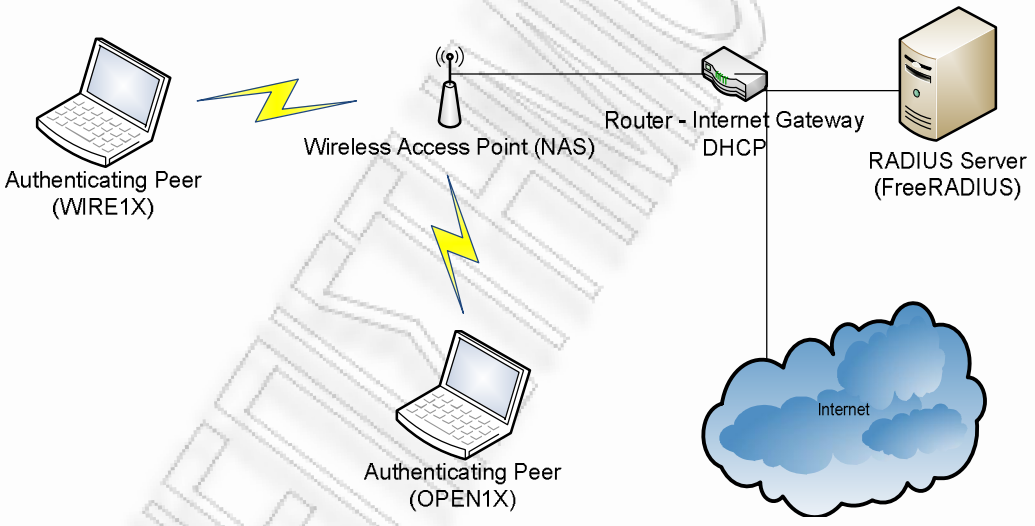
Το PEAP (Protected EAP) αναπτύχθηκε από τις Microsoft και Cisco με σκοπό να αποφύγουν την υλοποίηση και διαχείριση μιας υποδομής δημόσιο κλειδιού. Η δομή του είναι όμοια με το EAP-TTLS και αποτελεί μηχανισμό δύο φάσεων. Στην πρώτη ο διακομιστής αυθεντικοποιείται στον ασύρματο σταθμό, με τη βοήθεια ενός πιστοποιητικού, ενώ στη συνέχεια ο ασύρματος σταθμός αυθεντικοποιείται στο διακομιστή με κάποιο τρόπο. Η διαφορά με το EAP-TTLS έγκειται στις μεθόδους αυθεντικοποίησης κατά τη δεύτερη φάση. Το PEAP δύναται να χρησιμοποιήσει όλες τις μεθόδους.

6 Υλοποίηση σεναρίου

6.1 Περιληπτικά

Ο χρήστης ενός ασύρματου δικτύου θα αυθεντικοποιείται σε ασύρματο τοπικό δίκτυο, χρησιμοποιώντας το EAP πρωτόκολλο, το οποίο θα βασίζεται στο λογισμικό WIRE1X. Ο χρήστης θα διαθέτει λειτουργικό σύστημα Windows XP/Vista ή Linux (για την λειτουργία σε Linux λειτουργικά θα χρησιμοποιηθεί το OPEN1X). Ένας RADIUS Server, θα αναλαμβάνει την αυθεντικοποίηση του χρήστη εκ μέρους του ασύρματου τοπικού δικτύου. Ο RADIUS Server θα βασίζεται στο λογισμικό FreeRADIUS και θα έχει εγκατεστημένο το λειτουργικό σύστημα Linux.

Οι EAP μέθοδοι που θα χρησιμοποιηθούν είναι οι EAP-TLS, EAP-TTLS και PEAP.



Εικόνα 13 Τοπολογία Δικτύου

6.2 Εργαλεία

6.2.1 FreeRADIUS

Ο εξυπηρετητής FreeRADIUS είναι ένας «δαίμονας», για λειτουργικά τύπου linux ή unix, που επιτρέπει την εγκατάσταση ενός Server πρωτοκόλλου RADIUS, ο οποίος θα χρησιμοποιείται για τη αυθεντικοποίηση και την τιμολόγησης διαφόρων τύπων δικτυακής

πρόσβασης. Για τη χρήση του απαιτείται ένας client, σωστά ρυθμισμένος, για την επικοινωνία μαζί του. Τέτοιοι client είναι Ethernet Switches, Ασύρματα σημεία πρόσβασης ή ένας προσωπικός υπολογιστής με κατάλληλο λογισμικό προσομοίωσης του client.

Έχει αναπτυχτεί από μία ομάδα ανθρώπων που αποκαλούνται the FreeRADIUS Project και αποτελείται από ένα RADIUS Server, μία βιβλιοθήκη client με BSD άδεια, μία PAM βιβλιοθήκη και ένα module για τον Apache. Αποτελεί δε τον πιο ευρέως διαδεδομένο RADIUS server. Χρησιμοποιείται επίσης σε μεγάλο βαθμό και από την ακαδημαϊκή κοινότητα.

Ο FreeRADIUS υπήρξε ο πρώτος Open Source RADIUS server που υποστήριξε EAP. Στη δεύτερη του έκδοση υποστηρίζει περισσότερες EAP μεθόδους από οποιοδήποτε RADIUS server.

Συγκεκριμένα σύμφωνα με το FreeRADIUS project υποστηρίζονται οι παρακάτω EAP μέθοδοι, για ενσύρματη ή ασύρματη αυθεντικοποίηση:

EAP-AKA

EAP-FAST

EAP-GPSK

EAP-IKEv2 (experimental)

Cisco LEAP

EAP-PAX

EAP-PEAPv0

- EAP-MSCHAPv2
- EAP-GPSK
- EAP-GTC
- EAP-MD5-Challenge
- EAP-PAX
- EAP-PSK
- EAP-SAKE
- EAP-TLS

EAP-PEAPv1

- EAP-MSCHAPv2

- EAP-GPSK
- EAP-GTC
- EAP-MD5-Challenge
- EAP-PAX
- EAP-PSK
- EAP-SAKE

EAP-PSK

EAP-SAKE

EAP-SIM

EAP-TLS

EAP-TTLS

- PAP
- CHAP
- MS-CHAP
- MS-CHAPv2
- EAP-MSCHAPv2
- EAP-MD5
- EAP-GPSK
- EAP-GTC
- EAP-PAX
- EAP-PSK
- EAP-SAKE
- EAP-TLS

EAP-TNC (experimental)

6.2.2 WIRE1X

Το WIRE1X είναι μία υλοποίηση πελάτη IEEE802.1x. Πρόκειται για εφαρμογή ανοιχτού κώδικα που αναπτύχθηκε από το Wireless Internet Research & Engineering (WIRE) Lab.

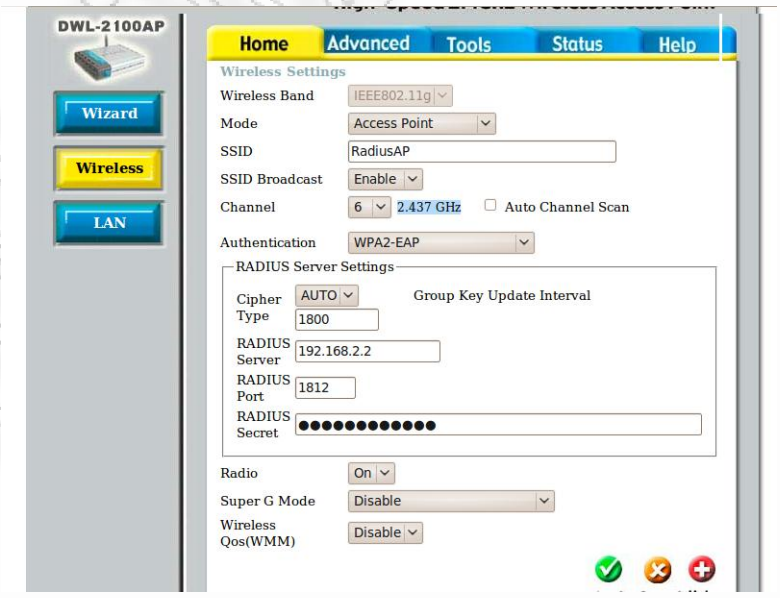
Βασίζεται στο OPEN1X, το οποίο υποστηρίζεται από Linux OS και αναπτύχτηκε για την κάλυψη των αναγκών των φοιτητών του National Tsing Hua University που χρησιμοποιούν Windows OS και απαιτούν ένα 802.1x client για την πρόσβαση στο ασύρματο δίκτυο του πανεπιστημίου.

Το WIRE1X υποστηρίζει αρκετές EAP μεθόδους αυθεντικοποίησης όπως MD5, TLS, TTLS, PEAP, MSCHAPv2, SIM, and AKA. Για την κρυπτογράφηση WEP, WPA με TKIP, WPA2 (802.11i) με CCMP. Λειτουργεί επίσης με freeRADIUS και υποστηρίζει ικανοποιητικό αριθμό ασύρματων καρτών. Τόσο ο πηγαίος κώδικας όσο και τα εκτελέσιμα αρχεία του είναι διαθέσιμα στο http://wire.cs.nthu.edu.tw/wire1x/.

6.3 Εγκατάσταση και παραμετροποίηση

6.3.1 Access Point

Το AP που χρησιμοποιήθηκε είναι το D-LINK DWL-2100AP. Στην εικόνα φαίνεται η παραμετροποίηση που απαιτείται για την χρήση RADIUS κατά την αυθεντικοποίηση των χρηστών.



Εικόνα 14 Ρυθμίσεις AP

Ορίζεται η μέθοδος αυθεντικοποίησης, η IP διεύθυνση και η θύρα στην οποία «ακούει» ο RADIUS Server , καθώς και το RADIUS Secret που μοιράζονται ο NAS (Access Point) και ο RADIUS Server. Το RADIUS Secret θα οριστεί στη συνέχει και κατά τις ρυθμίσεις του FreeRADIUS.

6.3.2 FreeRADIUS

Κρίνεται σκόπιμο η εγκατάσταση του OpenSSL να προηγηθεί της εγκατάστασης του FreeRADIUS. Η διαδικασία έγινε από το χρήστη με δικαιώματα root.

Η εγκατάσταση πραγματοποιήθηκε με την εξής διαδικασία:

```
$ wget "ftp://ftp.openssl.org/snapshot/openssl-0.9.7-stable-
SNAP-20100414.tar.gz"

$ mkdir -p /usr/src/802/openssl
$ cd /usr/src/802/openssl
$ cp /home/zoi/openssl-0.9.7-stable-SNAP-20100414.tar.gz
$ openssl-0.9.7-stable-SNAP-20100414.tar.gz

$ gunzip openssl-0.9.7-stable-SNAP-20100414.tar.gz
$ tar xvf openssl-0.9.7-stable-SNAP-20100414.tar
$ cd openssl-0.9.7-stable-SNAP-20100414

$./config shared --prefix=/usr/local/openssl
$ make
$ make install
```

Εγκατάσταση freeRADIUS:

Για την εγκατάσταση του freeRADIUS χρησιμοποιούνται οι παρακάτω εντολές:

```
$ wget ftp://ftp.freeradius.org/pub/radius/freeradius-server-
2.1.8.tar.gz

$ mkdir -p /usr/src/802/radius
$ cd /usr/src/802/radius
$ cp /home/zoi/ freeradius-server-2.1.8.tar.gz\
freeradius-server-2.1.8.tar.gz

$ gunzip freeradius-server-2.1.8.tar.gz
$ tar xvf freeradius-server-2.1.8.tar
$ cd freeradius-server-2.1.8

$./configure --with-openssl-
includes=/usr/local/openssl/include \
```

```
--with-openssl-libraries=/usr/local/openssl/lib \
--prefix=/usr/local/radius
$ make
$ make install
```

Πριν τον έλεγχο της επιτυχούς εγκατάστασης του freeRADIUS θα πρέπει να ελεγχθούν τα αρχεία users και clients.conf.

Στο αρχείο clients.conf περιγράφονται οι επιτρεπτοί clients του RADIUS με ορίσματα όπως το RADIUS Secret (θα πρέπει να είναι το ίδιο με αυτό που ορίστηκε στο Access Point) ενώ στο αρχείο users οι εξουσιοδοτημένοι χρήστες με λεπτομέρειες σχετικά με τα δικαιώματα τους. Οι πληροφορίες των χρηστών είναι δυνατό να ελέγχονται και με χρήση βάσης δεδομένων.

Για τον έλεγχο της σωστής εγκατάστασης ακολουθούνται τα παρακάτω βήματα.

- Εκκίνηση σε freeRADIUS debug mode

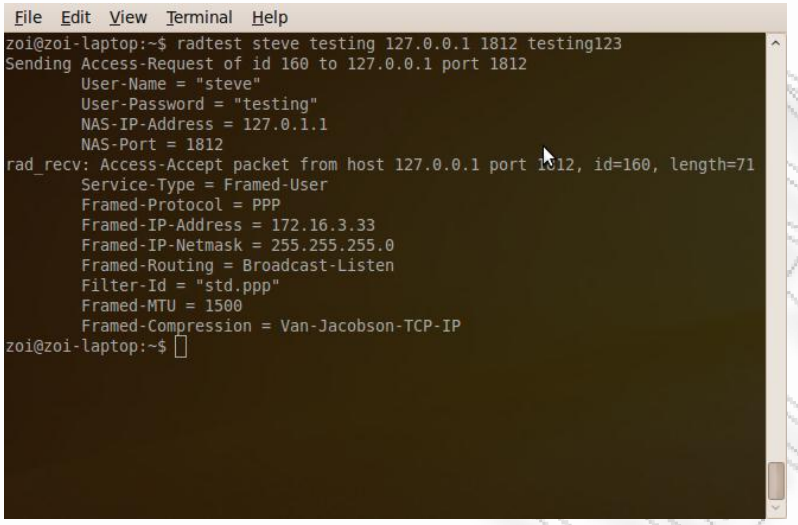
  $ radiusd -X

```
File  Edit  View  Terminal  Help
Module: Linked to module rlm_acct_unique
Module: Instantiating acct_unique
 acct_unique {
        key = "User-Name, Acct-Session-Id, NAS-IP-Address, Client-IP-Address, NAS-Port"
 }
Module: Checking accounting {...} for more modules to load
Module: Linked to module rlm_detail
Module: Instantiating detail
 detail {
        detailfile = "/var/log/freeradius/radacct/%{Client-IP-Address}/detail-%Y%m%d"
        header = "%t"
        detailperm = 384
        dirperm = 493
        locking = no
        log_packet_header = no
 }
Module: Instantiating attr_filter.accounting_response
 attr_filter attr_filter.accounting_response {
        attrsfile = "/etc/freeradius/attrs.accounting_response"
        key = "%{User-Name}"
 }
Module: Checking session {...} for more modules to load
Module: Checking post-proxy {...} for more modules to load
Module: Checking post-auth {...} for more modules to load
}
radiusd: #### Opening IP addresses and Ports ####
listen {
        type = "auth"
        ipaddr = *
        port = 0
}
listen {
        type = "acct"
        ipaddr = *
        port = 0
}
Listening on authentication address * port 1812
Listening on accounting address * port 1813
Listening on proxy address * port 1814
Ready to process requests.
```

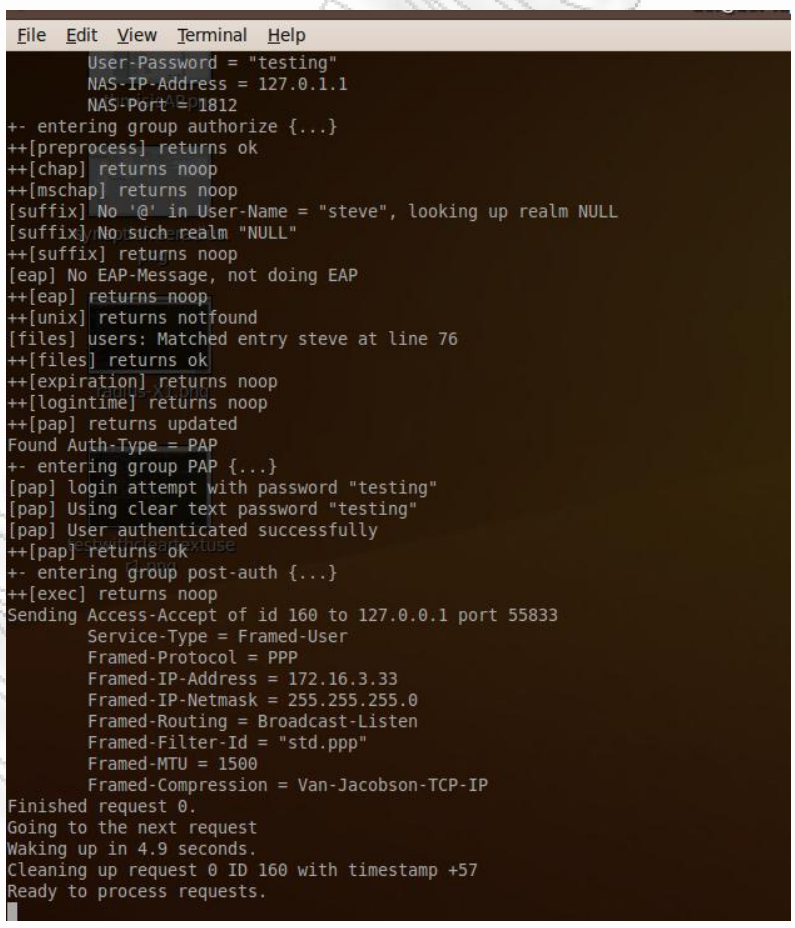
Εικόνα 15 Κονσόλα εκτέλεσης freeRADIUS σε debug mode

- Εκτέλεση radtest
  Χρησιμοποιείται με τα ακόλουθα ορίσματα:
  radtest user passwd radius-server[:port] nas-port-number secret [ppphint] [nasname]
  πχ.
  $ radtest steve testing 192.168.2.3 1812 radiussecret

  Ο χρήστης steve αναφέρεται στο αρχείο users με τον κωδικό 123



Εικόνα 16 χρήση radtest

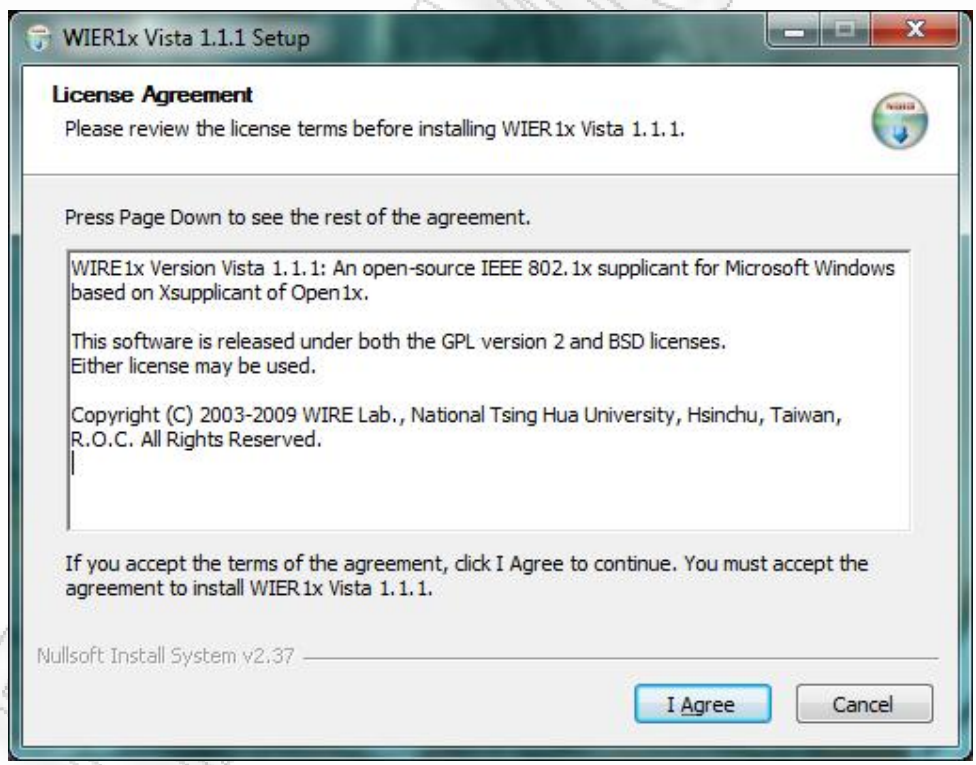


Εικόνα 17 Κονσόλα εκτέλεσης FreeRADIUS κατά το radtest

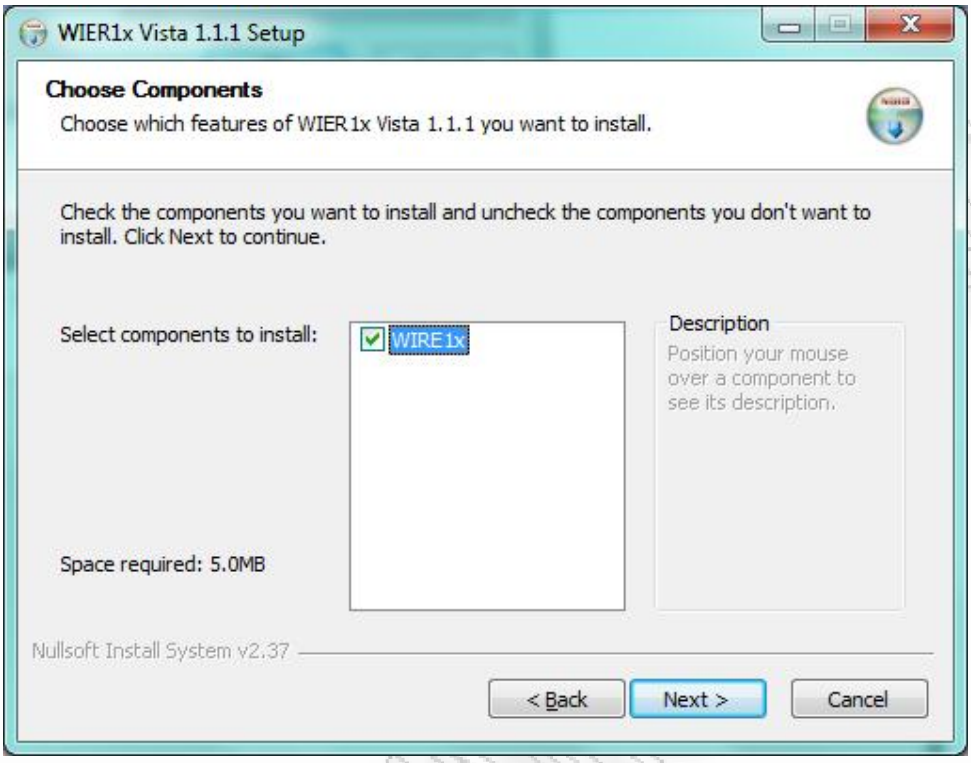
6.3.3 WIRE1X (OPEN1X)

Η εγκατάσταση του WIRE1X σε περιβάλλον WINDOWS, πραγματοποιείται με βοήθεια οδηγού εγκατάστασης. Παρακάτω φαίνεται αναλυτικά η διαδικασία:

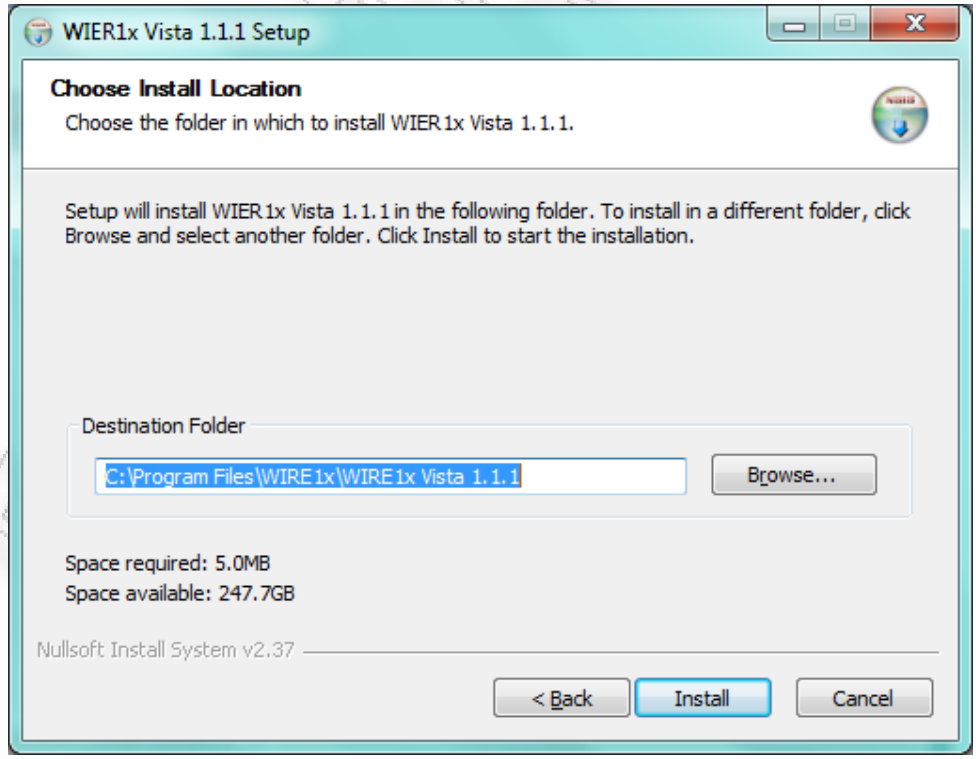
Εικόνα 18 Επιλογή γλώσσας εγκατάστασης



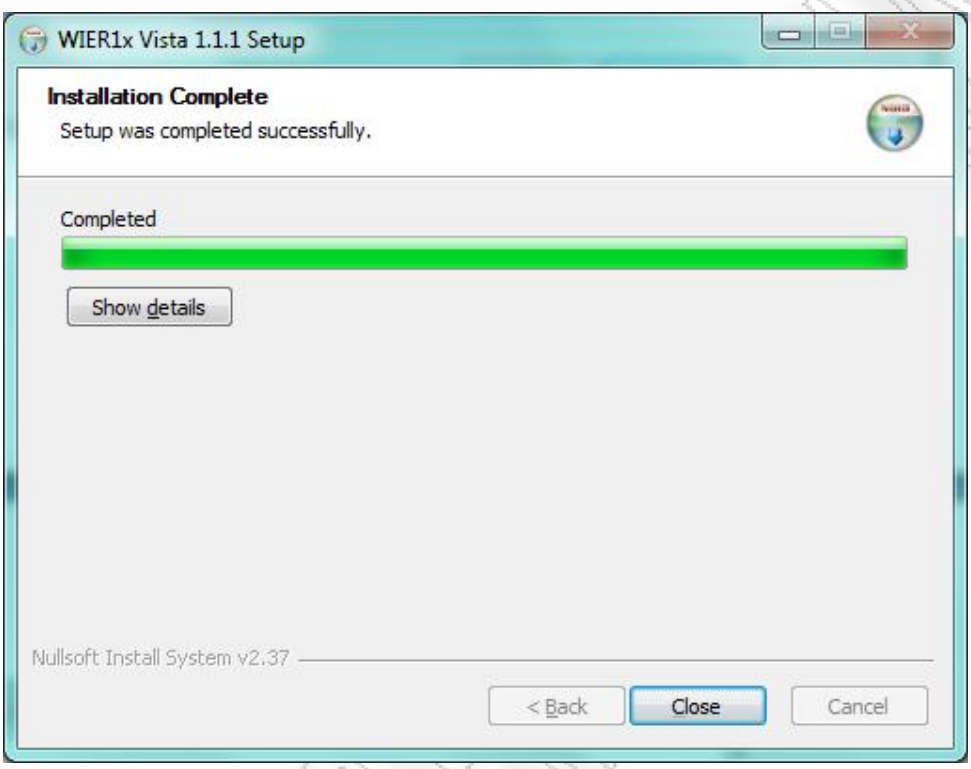
Εικόνα 19 Αποδοχή των όρων της άδειας χρήσης



Εικόνα 20 Επιλογή των components που θα εγκατασταθούν



Εικόνα 21 Επιλογή Φακέλου εγκατάστασης



Εικόνα 22 Ολοκλήρωση εγκατάστασης

6.4 Έκδοση και εγκατάσταση πιστοποιητικών

Τα απαιτούμενα πιστοποιητικά, που θα χρησιμοποιηθούν, θα υπογραφούν από τοπική αρχή πιστοποίησης με χρήση του openssl.

6.4.1 Έκδοση πιστοποιητικών

Η διαδικασία έκδοσης γίνεται μέσω εργαλείων του freeRADIUS περιγράφεται παρακάτω:

Στον κατάλογο /usr/local/etc/raddb/certs

Στον ίδιο κατάλογο υπάρχουν τα αρχεία ca.cnf client.cnf και server.cnf. Τα αρχεία αυτά περιέχουν τις πληροφορίες που θα χρησιμοποιηθούν κατά τη δημιουργία των πιστοποιητικών. Το περιεχόμενο τους είναι διαθέσιμο στο παράρτημα του παρόντος.

Εκτελούμε τις παρακάτω εντολές:

Διαγραφή υποδειγματικών πιστοποιητικών:

```
$ rm -f *.pem *.der *.csr *.crt *.key *.p12 serial* index.txt*
```

Δημιουργία πιστοποιητικών CA:

```
$ make ca.pem

$ make ca.der
```

Δημιουργία πιστοποιητικών Server:

```
$ make server.pem

$ make server.csr
```

Δημιουργία πιστοποιητικών πελάτη:

```
$ make client.pem
```

Τα αρχεία που περιέχονται πλέον στο φάκελο certs είναι τα εξής:

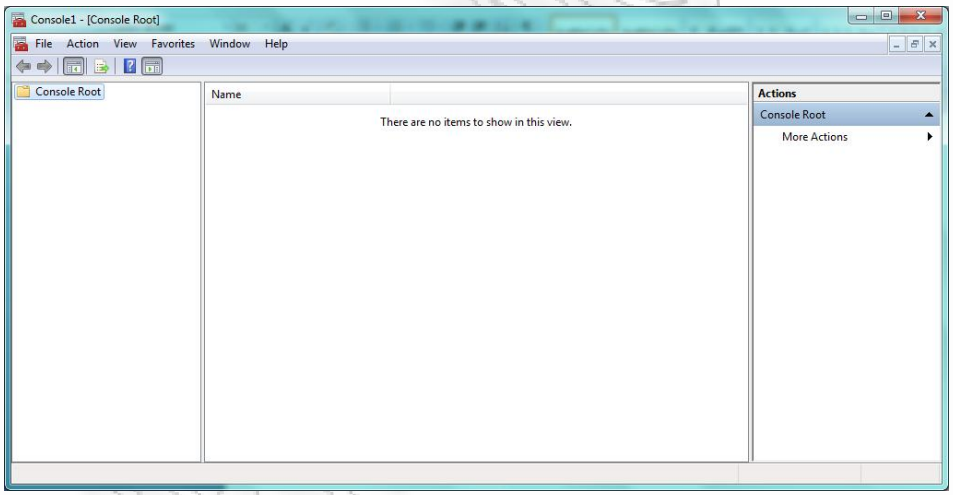
```
# ls -la
total 160
drwxr-x--- 2 root root 4096 2010-04-16 18:17 .
drwxr-xr-x 7 root root 4096 2010-04-15 22:43 ..
-rw-r--r-- 1 root root 4140 2010-04-16 18:15 01.pem
-rw-r--r-- 1 root root 4106 2010-04-16 18:17 02.pem
-rwxr-x--- 1 root root 2072 2010-04-14 00:13 bootstrap
-rw-r----- 1 root root 1269 2010-04-15 22:39 ca.cnf
-rw-r--r-- 1 root root 1147 2010-04-16 18:15 ca.der
-rw-r--r-- 1 root root 1751 2010-04-16 18:15 ca.key
-rw-r--r-- 1 root root 1610 2010-04-16 18:15 ca.pem
-rw-r----- 1 root root 1086 2010-04-15 22:40 client.cnf
-rw-r--r-- 1 root root 4106 2010-04-16 18:17 client.crt
-rw-r--r-- 1 root root 1017 2010-04-16 18:17 client.csr
-rw-r--r-- 1 root root 1743 2010-04-16 18:17 client.key
-rw-r--r-- 1 root root 2493 2010-04-16 18:17 client.p12
-rw-r--r-- 1 root root 3391 2010-04-16 18:17 client.pem
-rw-r----- 1 root root  245 2010-04-14 14:50 dh
-rw-r--r-- 1 root root  200 2010-04-16 18:17 index.txt
-rw-r--r-- 1 root root   20 2010-04-16 18:17 index.txt.attr
-rw-r--r-- 1 root root   21 2010-04-16 18:15
index.txt.attr.old
-rw-r--r-- 1 root root  107 2010-04-16 18:15 index.txt.old
-rw-r----- 1 root root 4279 2010-04-14 00:13 Makefile
-rw-r----- 1 root root 5120 2010-04-14 14:50 random
-rw-r----- 1 root root 7819 2010-04-14 00:13 README
```

```
-rw-r--r-- 1 root root    3 2010-04-16 18:17 serial
-rw-r--r-- 1 root root    3 2010-04-16 18:15 serial.old
-rw-r----- 1 root root 1103 2010-04-15 22:40 server.cnf
-rw-r--r-- 1 root root 4140 2010-04-16 18:15 server.crt
-rw-r--r-- 1 root root 1037 2010-04-16 18:15 server.csr
-rw-r--r-- 1 root root 1751 2010-04-16 18:15 server.key
-rw-r--r-- 1 root root 2509 2010-04-16 18:15 server.p12
-rw-r--r-- 1 root root 3433 2010-04-16 18:15 server.pem
-rw-r----- 1 root root  578 2010-04-14 00:13 xpextensions
-rw-r--r-- 1 root root 3391 2010-04-16 18:17
zmouto@gmail.com.pem
```

6.4.2 Εγκατάσταση πιστοποιητικών

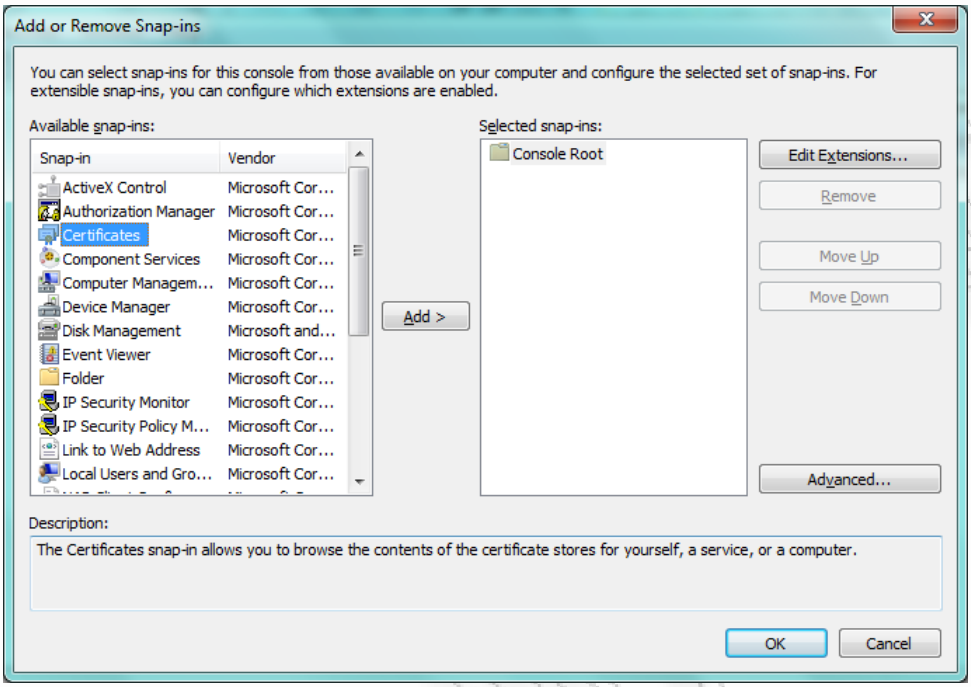
Η εγκατάσταση πιστοποιητικών σε λειτουργικό Windows πραγματοποιείται μέσω του εργαλείου Microsoft Management Console. Τα αρχεία που θα πρέπει να μεταφερθούν στον πελάτη είναι τα `client.p12` (για τη λειτουργία της EAP-TLS) και `ca.der`

Για την εκκίνηση της εφαρμογής επιλέγουμε Start -> Run και πληκτρολογούμε mmc.



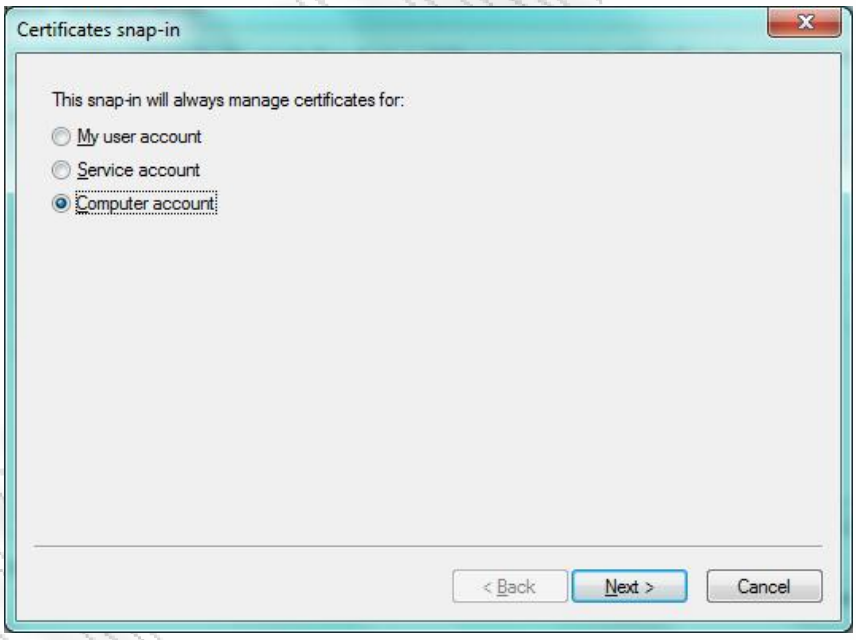
Εικόνα 23 Microsoft Management Console

Επιλέγουμε File -> Add/Remove Snap in …

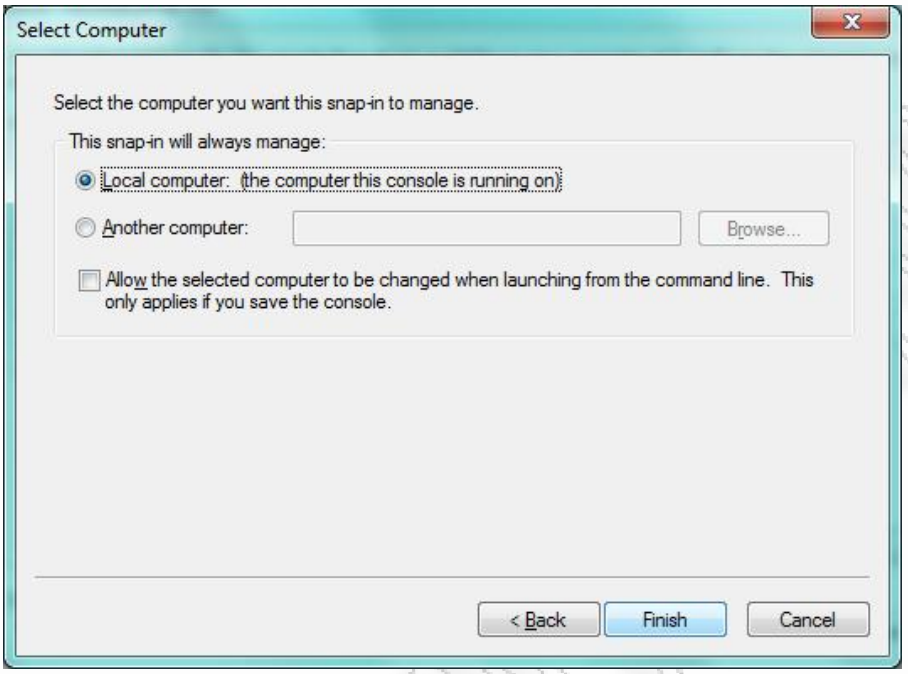


Εικόνα 24 Add or Remove Snap-ins

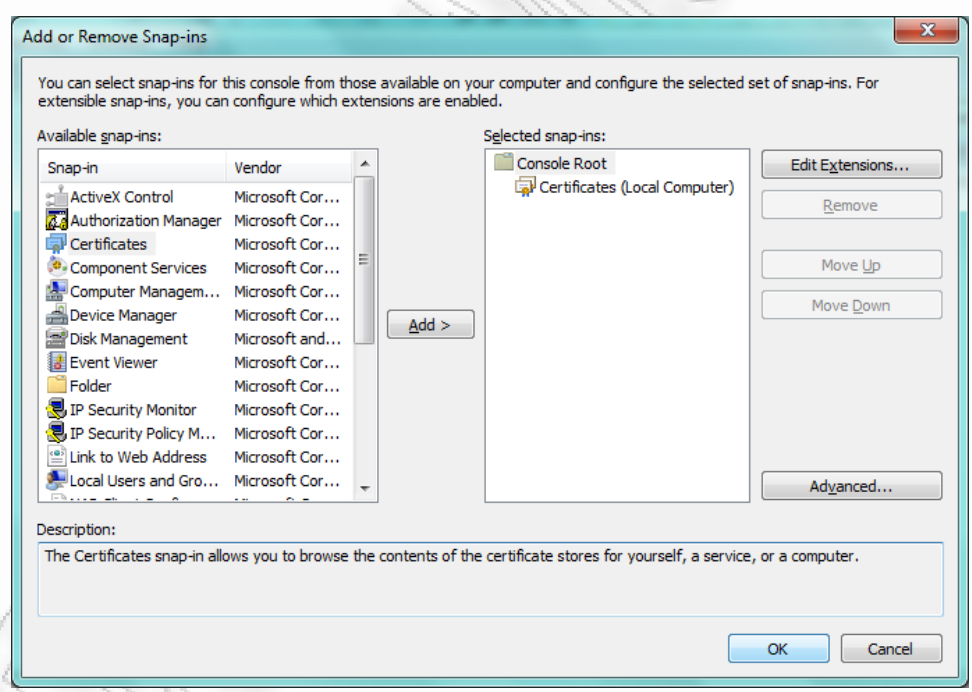
Στη συνέχεια επιλέγουμε Certificates , Add > και κάνουμε τις ακόλουθες επιλογές:



Εικόνα 25 Account Selection Screen



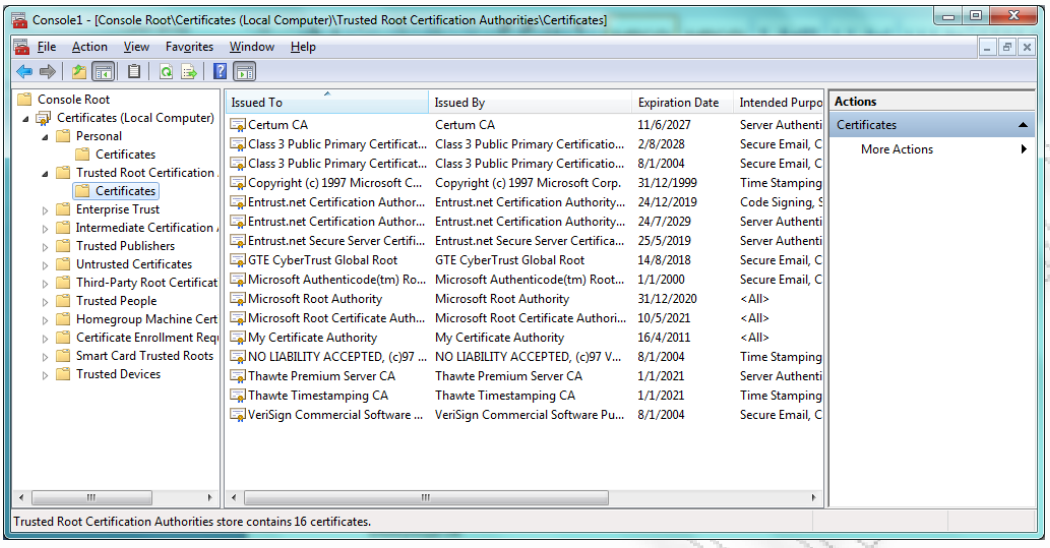
Εικόνα 26 Computer Selection



Εικόνα 27 Add or Remove Snap-ins

Επιλέγουμε OK

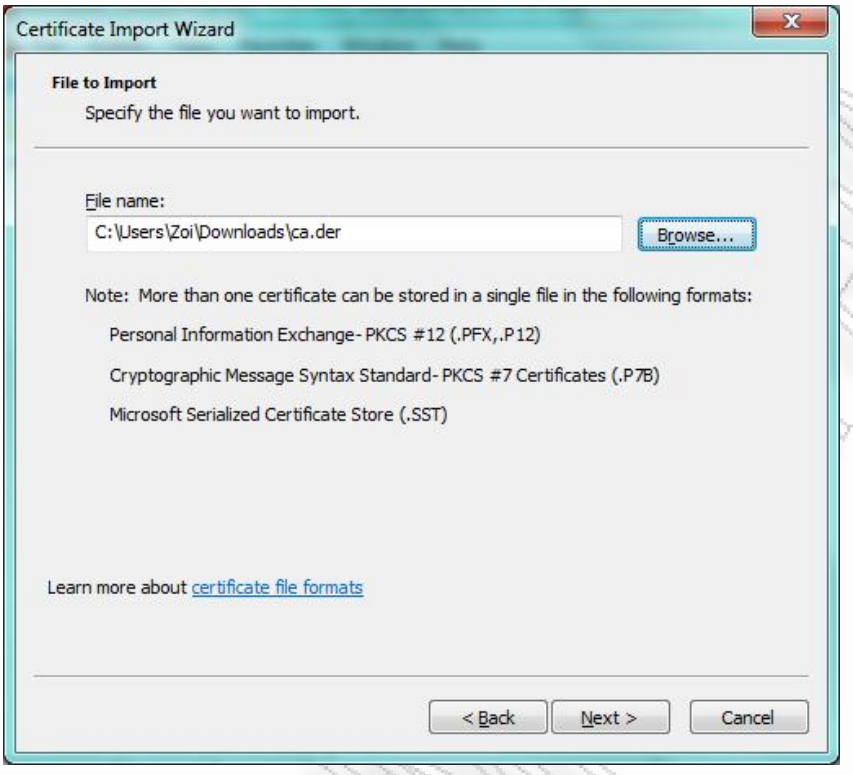
Από τη κεντρική οθόνη του Microsoft Management Console επεκτείνουμε το δένδρο αριστερά ως εξής:



Εικόνα 28 Microsoft Management Console Screen

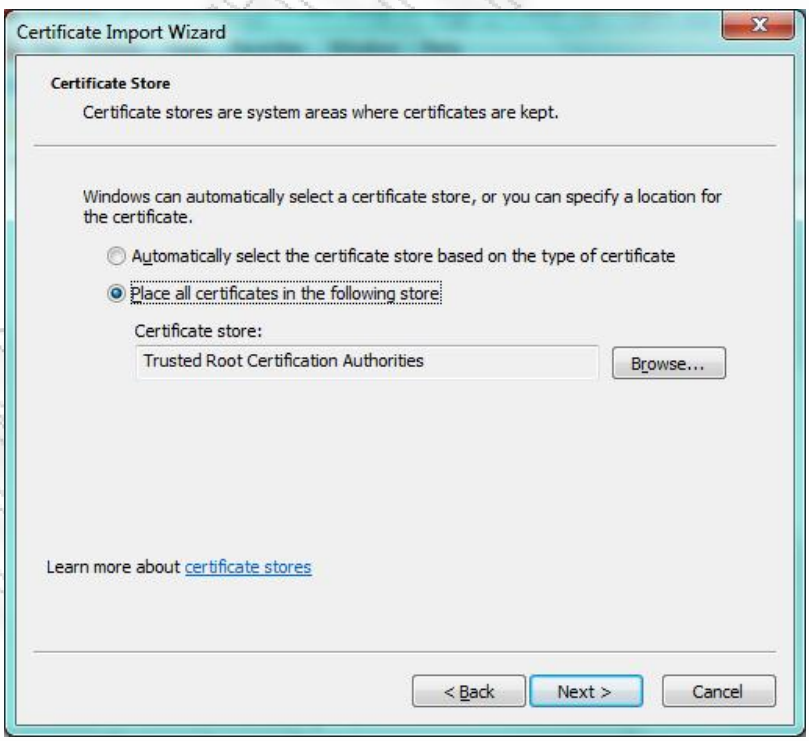
Για την εγκατάσταση του πιστοποιητικού του CA κάνουμε δεξί κλικ στο Certicates κάτω από το Trusted Root Certification και επιλέγουμε All Tasks -> Import

Εικόνα 29 Certification Import Wizard Welcome Screen

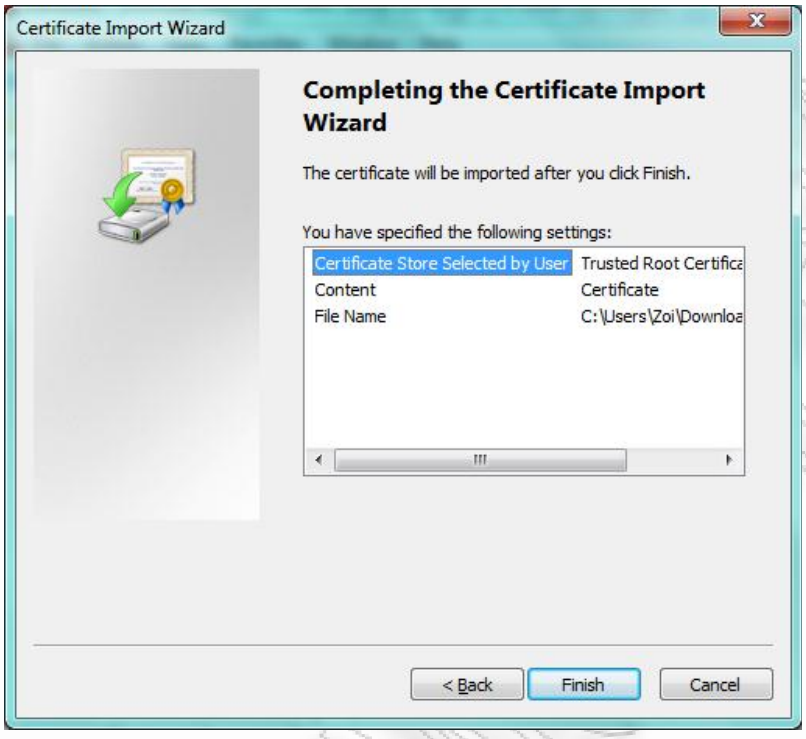


Εικόνα 30 Certification Import Wizard File Selection

Επιλέγουμε Browse και τη διαδρομή που έχουμε αποθηκεύσει το αρχείο ca.der



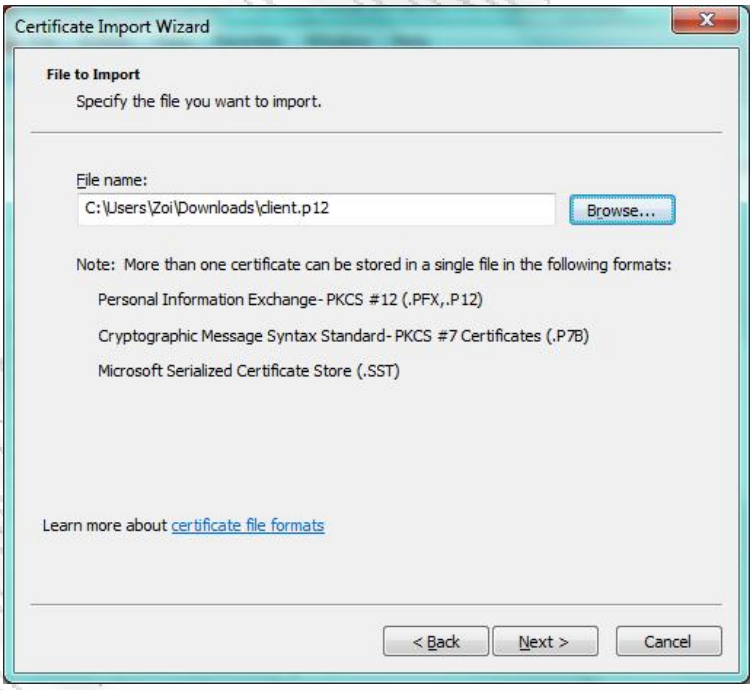
Εικόνα 31 Certification Import Wizard Storage Selection



Εικόνα 32 Certification Import Wizard Completed

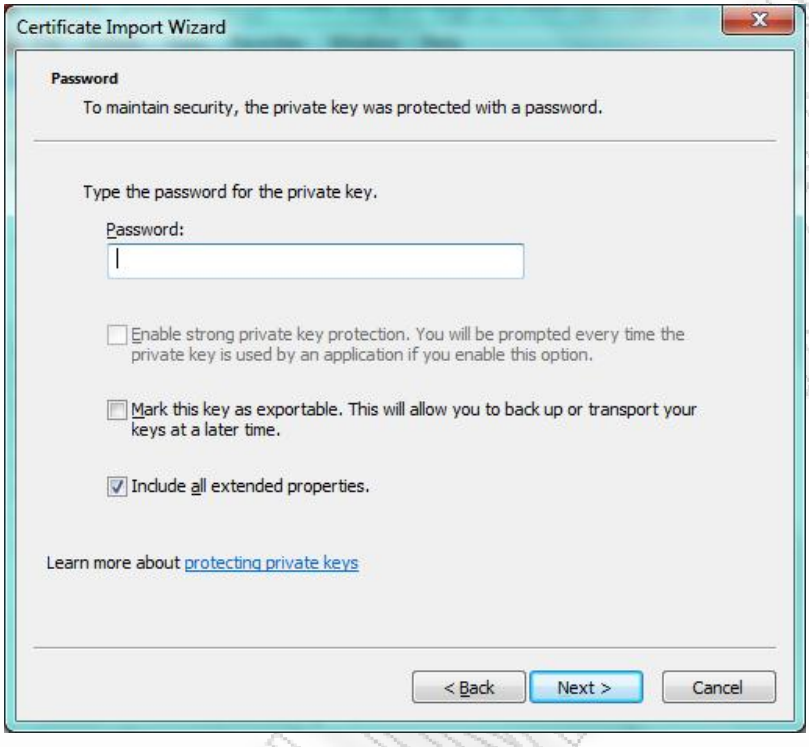
Όμοια για την εγκατάσταση του πιστοποιητικού του πελάτη κάνουμε δεξί κλικ στο Certificates κάτω από το Personal και επιλέγουμε All Tasks -> Import.

Εικόνα 33 Certification Import Wizard Welcome Screen

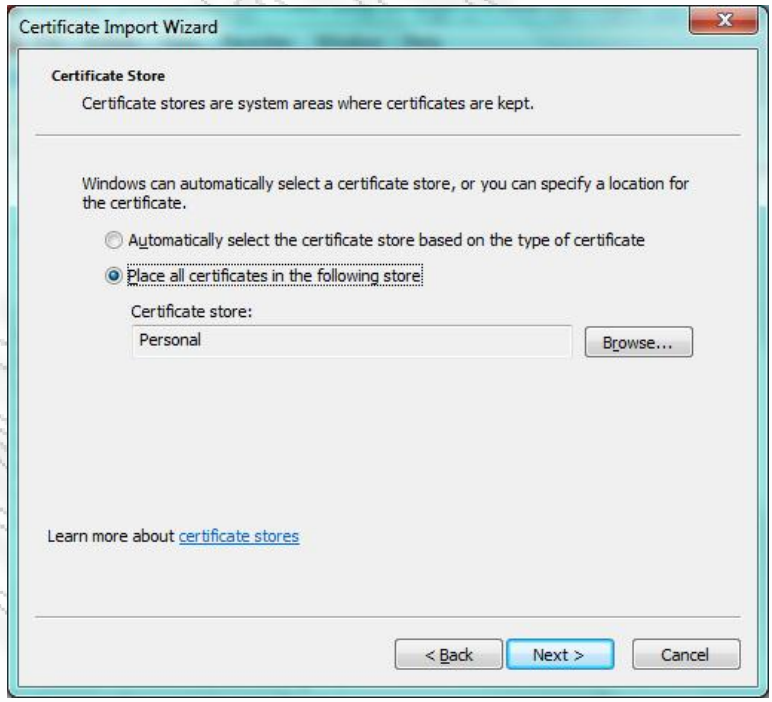


Εικόνα 34Certification Import Wizard File Selection

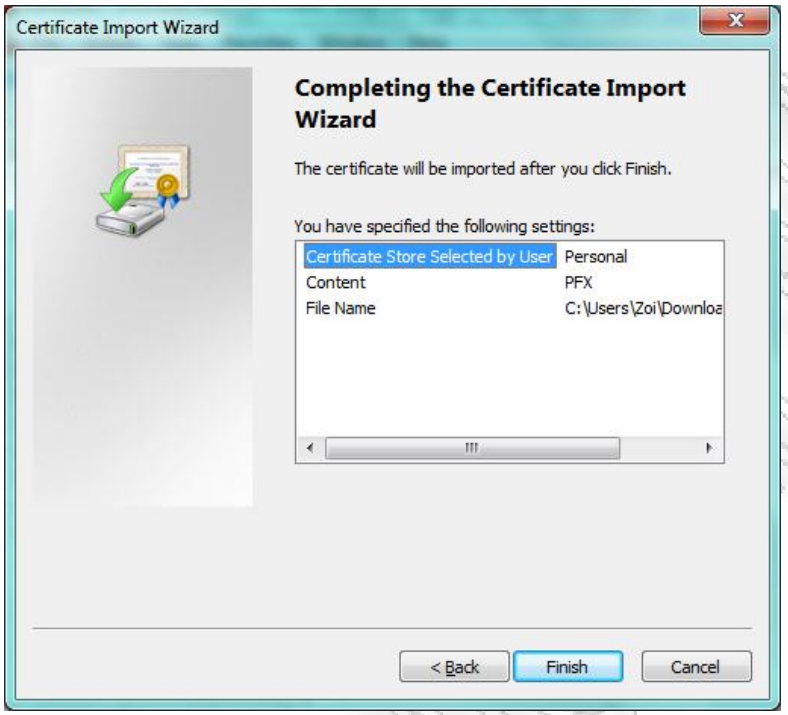
Πληκτρολογούμε το password που επιλέχτηκε για τη δημιουργία του πιστοποιητικού.



Εικόνα 35 Certification Import Wizard Password



Εικόνα 36Certification Import Wizard Storage Selection



Εικόνα 37 Certification Import Wizard Completed

6.5 Παραδείγματα λειτουργίας

Παρακάτω παρουσιάζονται οι ρυθμίσεις που απαιτούνται στον client για χρήση των μεθόδων EAP-TLS, PEAP και EAP-TTLS.

6.5.1 EAP-TLS

Στο EAP-TLS απαιτείται υποδομή δημόσιου κλειδιού. Στον client τα κλειδιά εγκαθίστανται με τη διαδικασία που περιγράφεται στην παράγραφο 6.4.2.

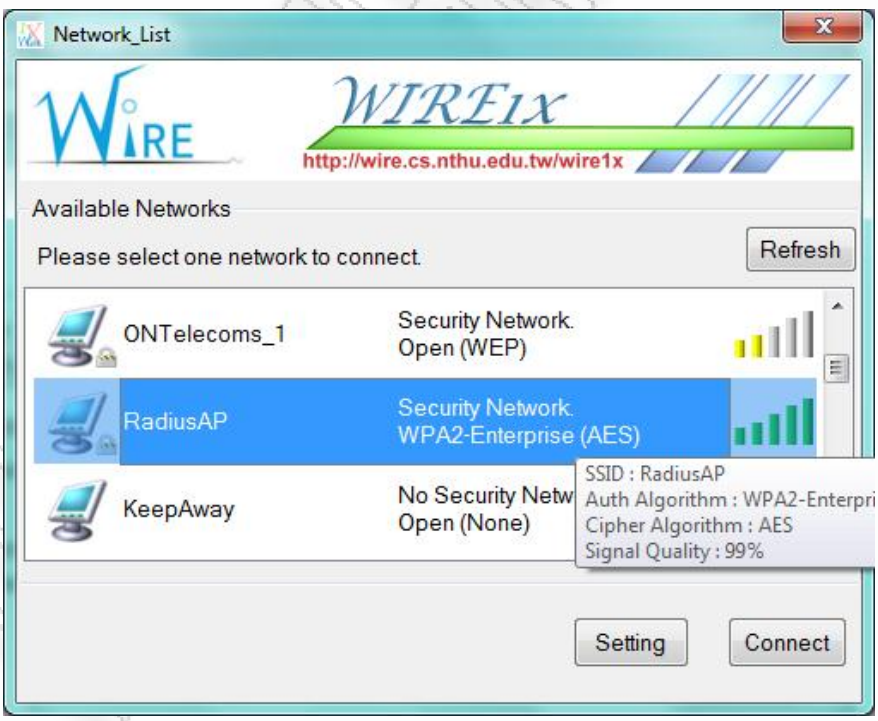
6.5.1.1 WIRE1x

Μετά την εγκατάσταση των πιστοποιητικών θα πρέπει να γίνει ρύθμιση του WIRE1x.

Από την αρχική οθόνη του WIRE1x επιλέγουμε Available Network List

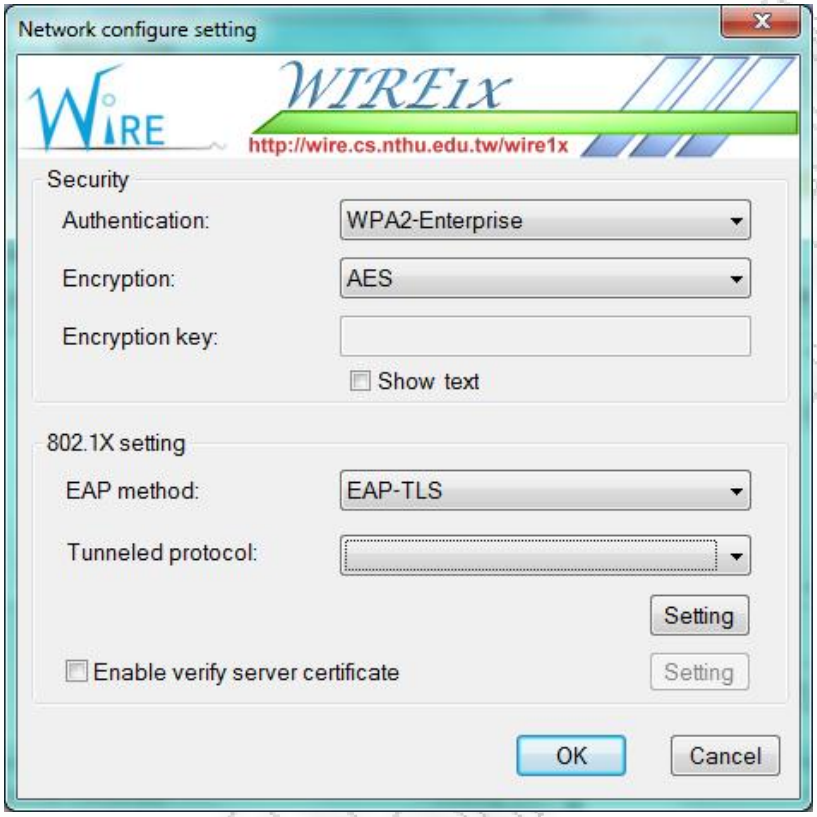
Εικόνα 38 Αρχική οθόνη WIRE1x

Επιλέγουμε το δίκτυο και στη συνέχεια Settings



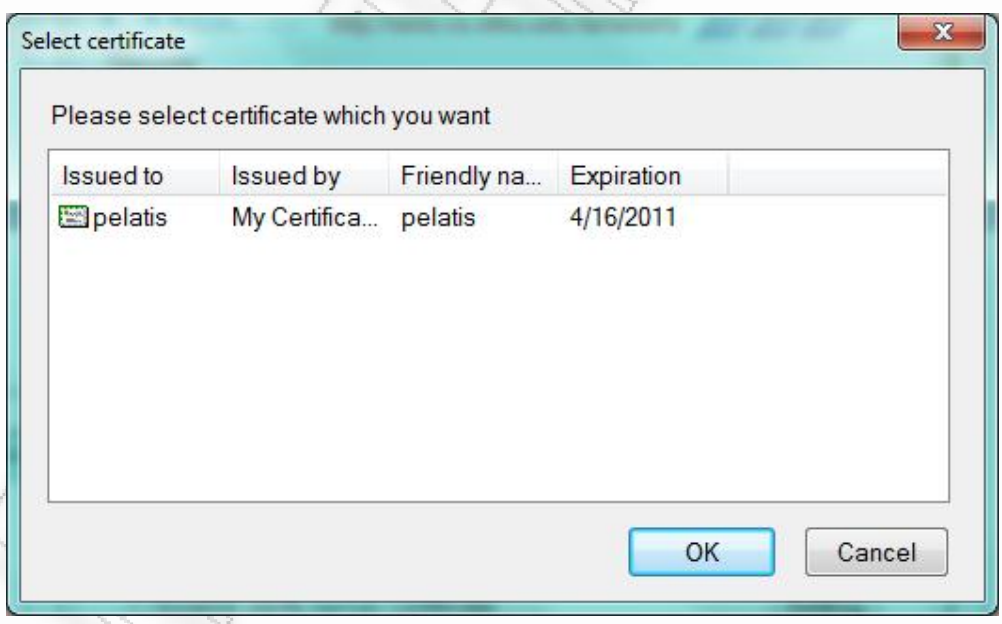
Εικόνα 39 Network List

Οι ρυθμίσεις φαίνονται για το EAP-TLS φαίνονται στις παρακάτω εικόνες:

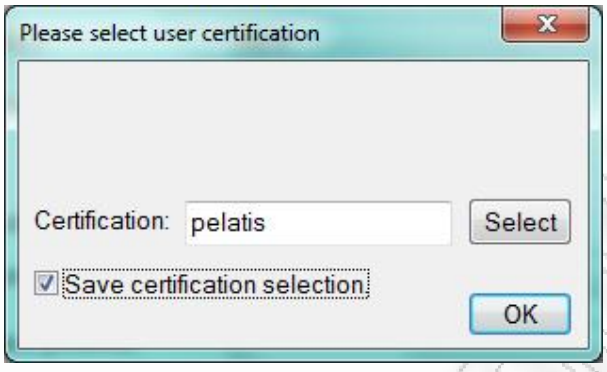


Εικόνα 40 Network configure Settings

Εικόνα 41 Επιλογή User Certification I

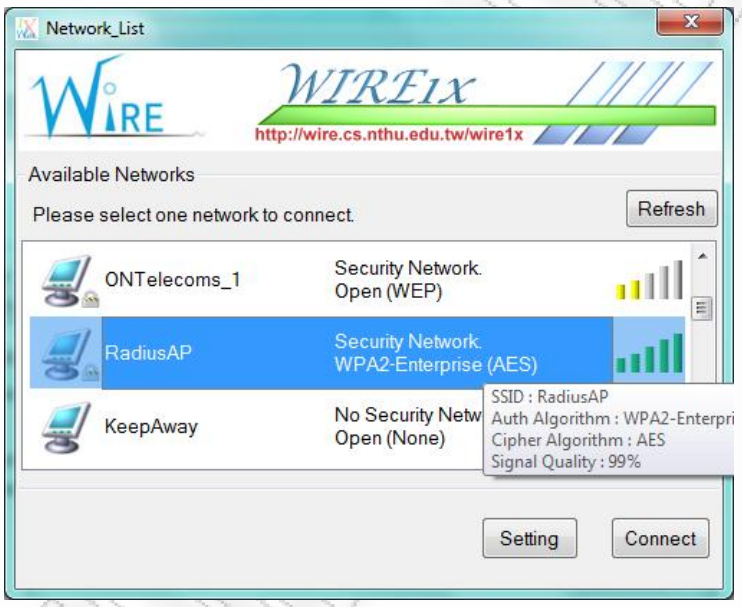


Εικόνα 42 Επιλογή user certification II



Εικόνα 43 Επιβεβαίωση User Certification

Από τη λίστα των διαθέσιμων δικτύων επιλέγουμε Connect στο επιθυμητό δίκτυο.



Εικόνα 44 Network List

6.5.1.2 Επισκόπηση του EAP-TLS με χρήση Wireshark

Παρακάτω φαίνεται η ακολουθία των μηνυμάτων που ανταλλάσσονται μεταξύ AP και RADIUS Server. Το capturing πραγματοποιήθηκε στο RADIUS Server με χρήση του εργαλείου Wireshark.

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=14, l=198)
2	0.009674	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=14, l=80)
3	0.015270	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=15, l=210)
4	0.026510	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=15, l=64)
5	1.211217	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=16, l=302)
6	1.261299	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=16, l=1090)
7	1.267795	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=17, l=210)
8	1.268392	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=17, l=1090)
9	1.278395	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=18, l=210)
10	1.280131	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=18, l=850)
11	1.331129	192.168.2.11	192.168.2.2	IP	Fragmented IP protocol (proto=UDP 0x11, off=14
12	1.331762	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=19, l=64)
13	1.340604	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=20, l=1341)
14	1.387496	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=20, l=127)
15	1.394402	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=21, l=210)
16	1.396312	192.168.2.2	192.168.2.11	RADIUS	Access-Accept(2) (id=21, l=170)

Εικόνα 45 EAP-TLS επισκόπιση μέσω Wireshark

Το αρχείο είναι διαθέσιμο στην διεύθυνση :
http://www.fileden.com/files/2010/4/17/2831314//tlsTrace

Όπως αναφέρθηκε παραπάνω τα EAP πλαίσια ενθυλακώνονται μέσα σε RADIUS πακέτα. Ενδεικτικά το RADIUS Access-Request περιέχει τα ακολουθά:

```
Frame 1 (240 bytes on wire, 240 bytes captured)

    Arrival Time: Apr 18, 2010 19:38:21.582686000

    [Time delta from previous captured frame: 0.000000000 seconds]

    [Time delta from previous displayed frame: 0.000000000 seconds]

    [Time since reference or first frame: 0.000000000 seconds]

    Frame Number: 1

    Frame Length: 240 bytes

    Capture Length: 240 bytes

    [Frame is marked: False]

    [Protocols in frame: eth:ip:udp:radius:eap]

    [Coloring Rule Name: UDP]

    [Coloring Rule String: udp]

Ethernet II, Src: D-Link_5a:5e:90 (00:22:b0:5a:5e:90), Dst:
Azurewav_07:d0:37 (00:22:43:07:d0:37)

    Destination: Azurewav_07:d0:37 (00:22:43:07:d0:37)

        Address: Azurewav_07:d0:37 (00:22:43:07:d0:37)

        .... ...0 .... .... .... .... = IG bit: Individual address
(unicast)
```

```
        .... ..0. .... .... .... .... = LG bit: Globally unique
address (factory default)

    Source: D-Link_5a:5e:90 (00:22:b0:5a:5e:90)

        Address: D-Link_5a:5e:90 (00:22:b0:5a:5e:90)

        .... ...0 .... .... .... .... = IG bit: Individual address
(unicast)

        .... ..0. .... .... .... .... = LG bit: Globally unique
address (factory default)

    Type: IP (0x0800)

Internet Protocol, Src: 192.168.2.11 (192.168.2.11), Dst: 192.168.2.2
(192.168.2.2)

    Version: 4

    Header length: 20 bytes

    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN:
0x00)

        0000 00.. = Differentiated Services Codepoint: Default (0x00)

        .... ..0. = ECN-Capable Transport (ECT): 0

        .... ...0 = ECN-CE: 0

    Total Length: 226

    Identification: 0x0115 (277)

    Flags: 0x00

        0.. = Reserved bit: Not Set

        .0. = Don't fragment: Not Set

        ..0 = More fragments: Not Set

    Fragment offset: 0

    Time to live: 64

    Protocol: UDP (0x11)

    Header checksum: 0xf398 [correct]

        [Good: True]

        [Bad : False]

    Source: 192.168.2.11 (192.168.2.11)

    Destination: 192.168.2.2 (192.168.2.2)
```

```
User Datagram Protocol, Src Port: 1028 (1028), Dst Port: radius
(1812)

    Source port: 1028 (1028)

    Destination port: radius (1812)

    Length: 206

    Checksum: 0x5825 [validation disabled]

        [Good Checksum: False]

        [Bad Checksum: False]
```

Radius Protocol

```
    Code: Access-Request (1)

    Packet identifier: 0xe (14)

    Length: 198

    Authenticator: 55457E990E1B398D6FFB7DA5266648C5

    Attribute Value Pairs

        AVP: l=18  t=Message-Authenticator(80):
8C67690DBB401E38B4E20D116ED71E25

            Message-Authenticator: 8C67690DBB401E38B4E20D116ED71E25

        AVP: l=6  t=Service-Type(6): Framed-User(2)

            Service-Type: Framed-User (2)

        AVP: l=10  t=User-Name(1): pelatis\000

            User-Name: pelatis

        AVP: l=6  t=Framed-MTU(12): 1488

            Framed-MTU: 1488

        AVP: l=28  t=Called-Station-Id(30): 00-22-B0-5A-5E-
90:RadiusAP

            Called-Station-Id: 00-22-B0-5A-5E-90:RadiusAP

        AVP: l=19  t=Calling-Station-Id(31): 00-26-C7-04-EE-3E

            Calling-Station-Id: 00-26-C7-04-EE-3E

        AVP: l=21  t=NAS-Identifier(32): D-Link Access Point

            NAS-Identifier: D-Link Access Point

        AVP: l=6  t=NAS-Port-Type(61): Wireless-802.11(19)
```

```
    NAS-Port-Type: Wireless-802.11 (19)
AVP: l=24  t=Connect-Info(77): CONNECT 54Mbps 802.11g
    Connect-Info: CONNECT 54Mbps 802.11g
AVP: l=14  t=EAP-Message(79) Last Segment[1]
    EAP fragment
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 14
        Length: 12
        Type: Identity [RFC3748] (1)
        Identity (7 bytes): pelatis
AVP: l=6  t=NAS-IP-Address(4): 192.168.2.11
    NAS-IP-Address: 192.168.2.11 (192.168.2.11)
AVP: l=6  t=NAS-Port(5): 1
    NAS-Port: 1
AVP: l=14  t=NAS-Port-Id(87): STA port # 1
    NAS-Port-Id: STA port # 1
```

Αναλυτικά το πλήρες περιεχόμενο των πακέτων που συλλέχτηκαν κατά την EAP-TLS αυθεντικοποίηση είναι διαθέσιμο στη διεύθυνση:

http://www.fileden.com/files/2010/4/17/2831314//tls.txt

6.5.2 PEAP

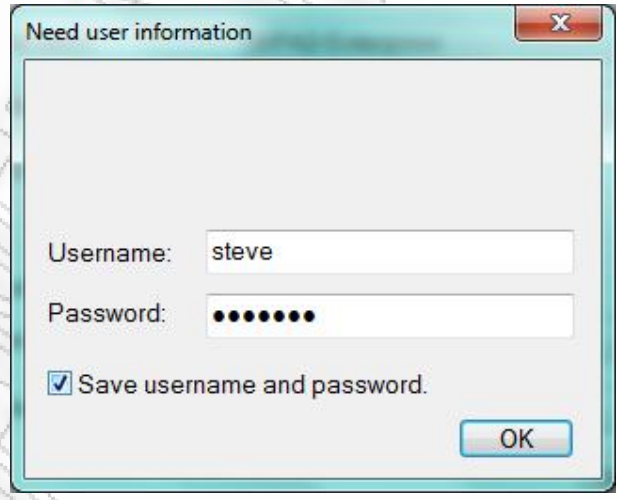
Στη μέθοδο αυτή δεν απαιτούνται πιστοποιητικά για αυθεντικοποίηση του πελάτη στο διακομιστή. Ο χρήστη αυθεντικοποιείται με χρήση ονόματος χρήστη και κωδικού πρόσβασης.

6.5.2.1 Ρυθμίσεις WIRE1x

Στις ρυθμίσεις του δικτύου που επιθυμούμε να συνδεθούμε επιλέγουμε τα παρακάτω:

Εικόνα 46 PEAP Network Configuration

Επιλέγουμε Settings για τον ορισμό του ονόματος χρήστη και του κωδικού πρόσβασης.



Εικόνα 47 User Information

Στη συνέχεια επιλέγουμε Connect.

6.5.2.2 Επισκόπηση του PEAP με χρήση Wireshark

Παρακάτω φαίνεται η ακολουθία των μηνυμάτων που ανταλλάσσονται μεταξύ AP και RADIUS Server. Το Wireshark αρχείο είναι διαθέσιμο στη διεύθυνση http://www.fileden.com/files/2010/4/17/2831314//peapTrace και η ανάλυση των πακέτων στη http://www.fileden.com/files/2010/4/17/2831314//peap.txt

No. .	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=40, l=194)
2	0.000952	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=40, l=80)
3	0.006581	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=41, l=208)
4	0.007556	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=41, l=64)
5	0.019733	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=42, l=300)
6	0.052683	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=42, l=1090)
7	0.063175	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=43, l=208)
8	0.063683	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=43, l=1086)
9	0.070204	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=44, l=208)
10	0.070737	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=44, l=696)
11	0.106861	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=45, l=410)
12	0.115005	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=45, l=123)
13	0.122986	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=46, l=208)
14	0.127569	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=46, l=101)
15	0.137891	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=47, l=282)
16	0.139238	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=47, l=117)
17	0.144782	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=48, l=346)
18	0.151460	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=48, l=149)
19	0.159935	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=49, l=282)
20	0.160979	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=49, l=101)
21	0.168919	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=50, l=282)
22	0.169544	192.168.2.2	192.168.2.11	RADIUS	Access-Accept(2) (id=50, l=168)

Εικόνα 48 PEAP - Πακέτα στο RADIUS Server

Ενδεικτικά παρουσιάζεται παρακάτω τα RADIUS Access-Request και RADIUS Access-Success:

RADIUS Access-Request (Τα EAP δεδομένα σημειώνονται με πλάγια γραφή)

```
Radius Protocol

    Code: Access-Request (1)

    Packet identifier: 0x1d (29)

    Length: 194

    Authenticator: 6DA8116B6D28764F3B5D61EF7EBD2ECB

    Attribute Value Pairs

        AVP: l=18  t=Message-Authenticator(80):
FD6FE5D4BC475A73FBE621A222425B66

            Message-Authenticator:
FD6FE5D4BC475A73FBE621A222425B66

        AVP: l=6  t=Service-Type(6): Framed-User(2)

            Service-Type: Framed-User (2)

        AVP: l=8  t=User-Name(1): steve\000
```

```
            User-Name: steve
        AVP: l=6  t=Framed-MTU(12): 1488
            Framed-MTU: 1488
        AVP: l=28  t=Called-Station-Id(30): 00-22-B0-5A-5E-
90:RadiusAP
            Called-Station-Id: 00-22-B0-5A-5E-90:RadiusAP
        AVP: l=19  t=Calling-Station-Id(31): 00-26-C7-04-EE-3E
            Calling-Station-Id: 00-26-C7-04-EE-3E
        AVP: l=21  t=NAS-Identifier(32): D-Link Access Point
            NAS-Identifier: D-Link Access Point
        AVP: l=6  t=NAS-Port-Type(61): Wireless-802.11(19)
            NAS-Port-Type: Wireless-802.11 (19)
        AVP: l=24  t=Connect-Info(77): CONNECT 54Mbps 802.11g
            Connect-Info: CONNECT 54Mbps 802.11g
        AVP: l=12  t=EAP-Message(79) Last Segment[1]
            EAP fragment
            Extensible Authentication Protocol
                Code: Response (2)
                Id: 29
                Length: 10
                Type: Identity [RFC3748] (1)
                Identity (5 bytes): steve
        AVP: l=6  t=NAS-IP-Address(4): 192.168.2.11
            NAS-IP-Address: 192.168.2.11 (192.168.2.11)
        AVP: l=6  t=NAS-Port(5): 1
            NAS-Port: 1
        AVP: l=14  t=NAS-Port-Id(87): STA port # 1
            NAS-Port-Id: STA port # 1
```

Τα EAP-TLS δεδομένα ενθυλακώνονται στα Access-Challenge και Access-Request πακέτα που ακολουθούν.

Αναλυτικά είναι διαθέσιμα στη διεύθυνση
http://www.fileden.com/files/2010/4/17/2831314//peap-eapPart.txt

Το τελευταίο EAP πακέτο (Code 3) ενθυλακώνεται σε Access-Accept RADIUS πακέτο.

```
Radius Protocol

    Code: Access-Accept (2)

    Packet identifier: 0x27 (39)

    Length: 168

    Authenticator: 8CE1B9A358A71B0D0A4BFA7B30B24899

    [This is a response to a request in frame 36]

    [Time from request: 0.001083000 seconds]

    Attribute Value Pairs

        AVP: l=58  t=Vendor-Specific(26) v=Microsoft(311)

            VSA: l=52 t=MS-MPPE-Recv-Key(17):
C0F0DB4673F93F320DC43F033013CFC94ED3E04AC34FCFCB...

                MS-MPPE-Recv-Key:
C0F0DB4673F93F320DC43F033013CFC94ED3E04AC34FCFCB...

        AVP: l=58  t=Vendor-Specific(26) v=Microsoft(311)

            VSA: l=52 t=MS-MPPE-Send-Key(16):
CF86C05660C6D1981633AE9FA3A03E595F545895AAF3427F...

                MS-MPPE-Send-Key:
CF86C05660C6D1981633AE9FA3A03E595F545895AAF3427F...

        AVP: l=6  t=EAP-Message(79) Last Segment[1]

            EAP fragment

            Extensible Authentication Protocol

                Code: Success (3)

                Id: 39

                Length: 4

        AVP: l=18  t=Message-Authenticator(80):
AF8DA8AA9BB7D0D0FF851953120C5B90

            Message-Authenticator:
AF8DA8AA9BB7D0D0FF851953120C5B90

        AVP: l=8  t=User-Name(1): steve\000
```

```
User-Name: steve
```

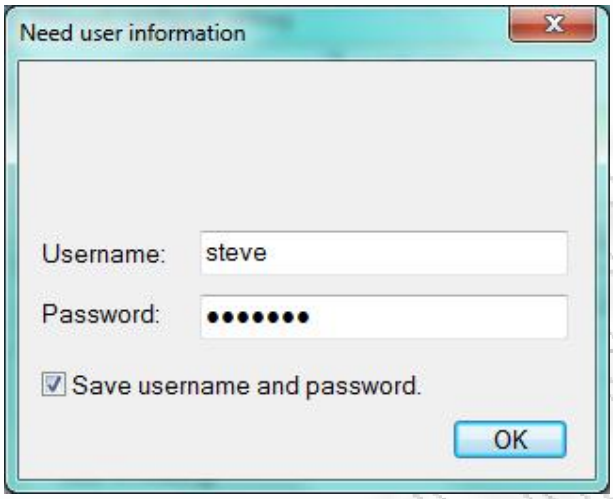
6.5.3 EAP-TTLS

Όπως αναφέρθηκε και παραπάνω οι διαφορές μεταξύ PEAP και EAP-TTLS αφορά μεθόδους αυθεντικοποίησης κατά τη δεύτερη φάση. Παρακάτω παρουσιάζεται ένα παράδειγμα λειτουργίας του EAP-TTLS.

6.5.3.1 WIRE1.X

Οι ρυθμίσεις που απαιτούνται στο WIRE1X για χρήση του EAP-TTLS φαίνονται παρακάτω:

Εικόνα 49 EAP-TTLS Network Configuration



Εικόνα 50 EAP-TTLS User Information

6.5.3.2 *Επισκόπηση του EAP-TTLS με χρήση Wireshark*

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=7, l=194)
2	0.000915	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=7, l=80)
3	0.006499	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=8, l=208)
4	0.007436	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=8, l=64)
5	0.769663	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=9, l=300)
6	0.826985	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=9, l=1090)
7	0.833758	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=10, l=208)
8	0.834238	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=10, l=1090)
9	0.843537	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=11, l=208)
10	0.849825	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=11, l=700)
11	0.882505	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=12, l=410)
12	0.892054	192.168.2.2	192.168.2.11	RADIUS	Access-challenge(11) (id=12, l=127)
13	0.899867	192.168.2.11	192.168.2.2	RADIUS	Access-Request(1) (id=13, l=314)
14	0.900903	192.168.2.2	192.168.2.11	RADIUS	Access-Accept(2) (id=13, l=168)

Εικόνα 51 EAP-TTLS Wireshark II

Το πλήρες Wireshark αρχείο βρίσκεται στη διεύθυνση:
http://www.fileden.com/files/2010/4/17/2831314//ttlsTrace

Τα περιεχόμενα των πακέτων που ανταλλάσσονται κατά την αυθεντικοποίηση μέσω EAP-TTLS είναι διαθέσιμα στη διεύθυνση:

http://www.fileden.com/files/2010/4/17/2831314//ttls.txt

Το EAP τμήμα τους, που ενθυλακώνεται στο RADIUS πακέτα είναι διαθέσιμα στη διεύθυνση:

http://www.fileden.com/files/2010/4/17/2831314//ttls-eapPart.txt

6.6 Το OPEN1X

Λογω απουσίας έκδοσης του WIRE1X, για την υλοποιήση του Supplicant σε λειτουργίκο Linux, θα χρησιμοποιηθεί το λογισμικό OPEN1X - XSupplicant, πάνω στον κώδικα του οποίου βασίζεται το WIRE1X.

Το OPEN1X είναι διαθέσιμο στη διεύθυνση http://open1x.sourceforge.net/

Η λειτουργία του είναι όμοια με αυτή του WIRE1x. Παρακάτω παρουσιάζεται η διαδικασία εγκατάστασης του καθώς και ρύθμισης του.

6.6.1 Εγκατάσταση OPEN1X

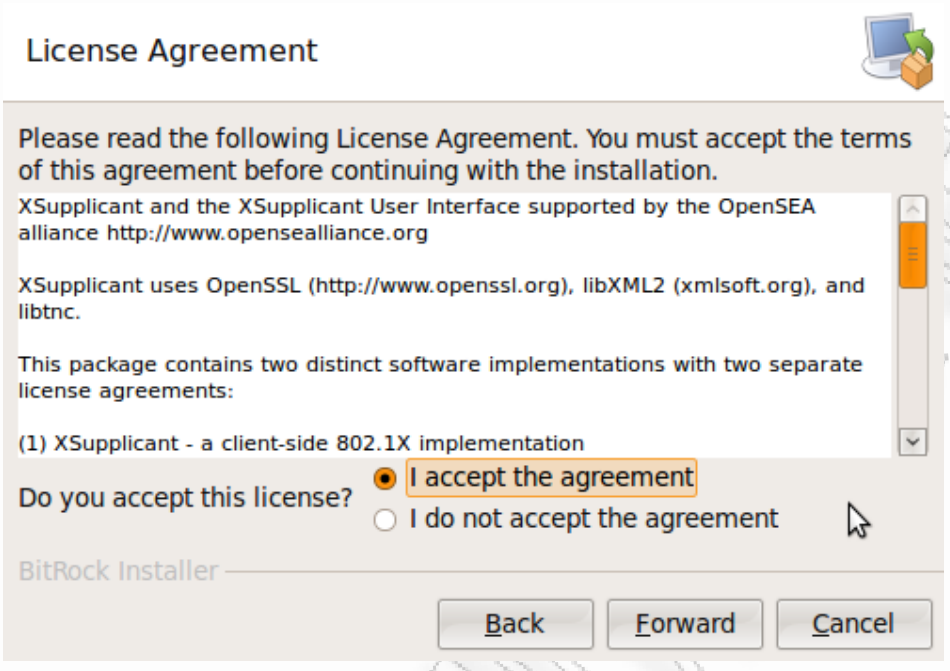
Επιλέγουμε για λήψη το αρχείο xsupplicant-2.2.0.710-linux-installer.bin

Όταν ολοκληρωθεί η λήψη, εκτελούμε τις παρακάτω εντολές και η εγκατάσταση μέσω γραφικού περιβάλλοντος. Η διαδικασία φαίνεται παρακάτω:

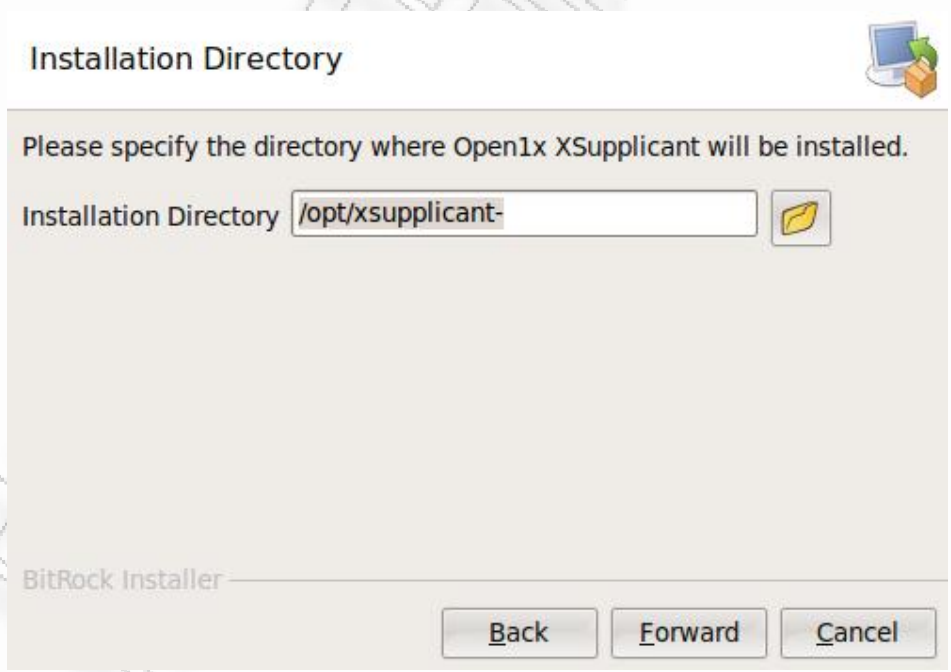
$ chmod 755 xsupplicant-2.2.0.710-linux-installer.bin

$ ./xsupplicant-2.2.0.710-linux-installer.bin

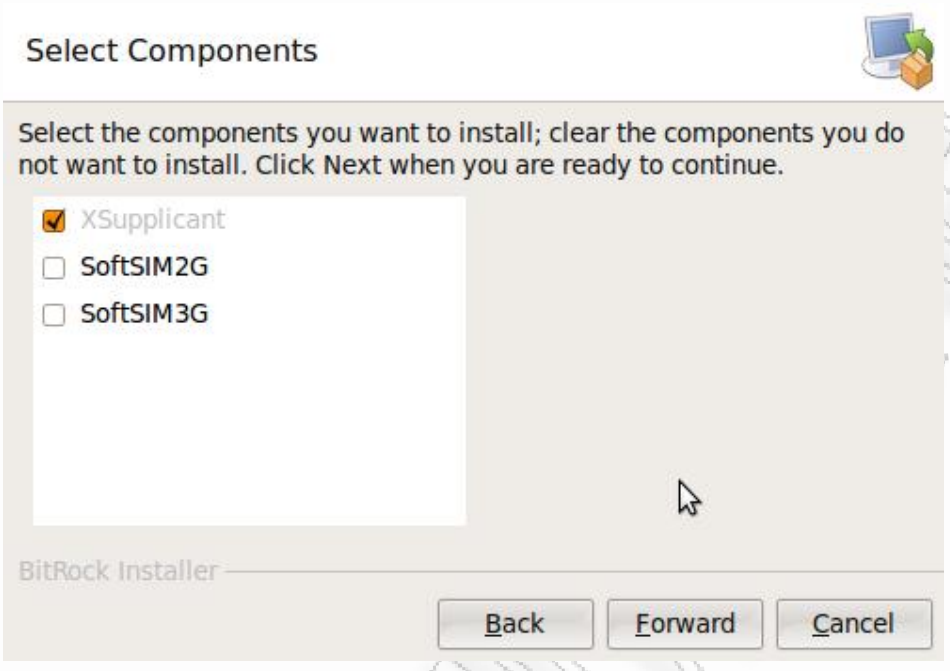
Εικόνα 52 Εκκίνηση εγκατάστασης



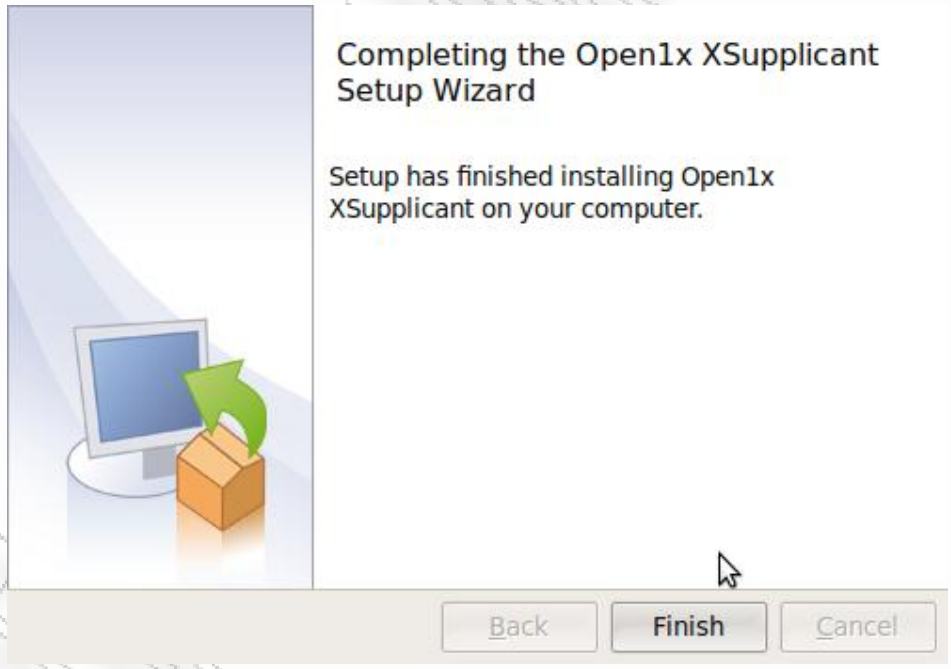
Εικόνα 53 Αποδοχή όρων άδειας χρήσης



Εικόνα 54 Επιλογή φακέλου προορισμού εγκατάστασης



Εικόνα 55 Επιλογή στοιχείων εγκατάστασης



Εικόνα 56 Ολοκλήρωση εγκατάστασης

6.6.2 Παραμετροποίηση OPEN1X

6.6.2.1 Εγκατάσταση πιστοποιητικών

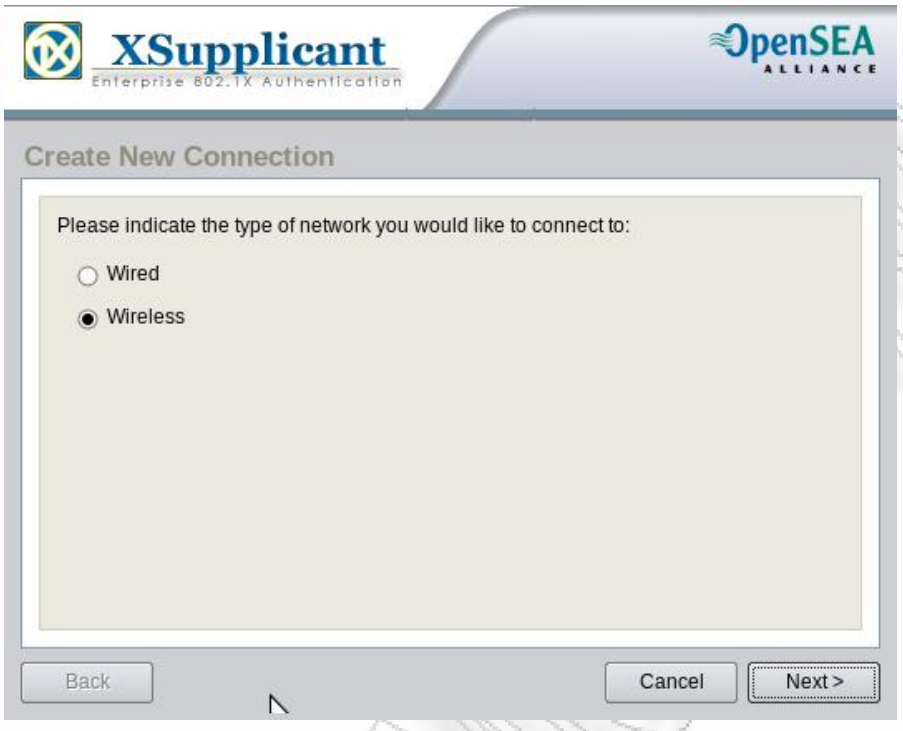
Η εγκατάσταση πιστοποιητικών στην έκδοση είναι 2.2.0 είναι ιδιαίτερα απλή και πραγματοποιήται με την αντιγραφή των .pem αρχείων στους καταλόγους /root/xsupplicant/user_certs, για το πιστοποιητικό του client , και/root/xsupplicant/certs για το πιστοποιητικό CA.

6.6.2.2 Ρυθμίσεις XSupplicant

Στην αρχική οθόνη του XSupplicant, στην καρτέλα Wireless επιλέγουμε στο πεδίο Connection, New Connection και ακολοθούμε τον οδηγό ρυθμίσεων.

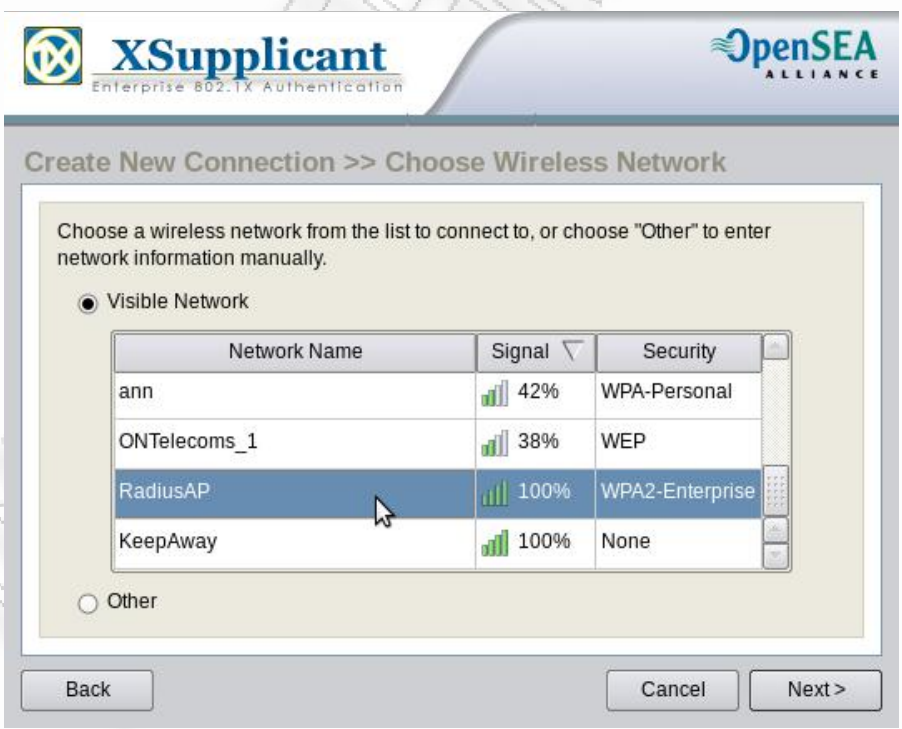
Εικόνα 57 OPEN1x Xsupplicant

Στη συνέχεια επιλέγουμε αν η συνδεση θα είναι ασύρματη ή ενσύρματη. Στη προκειμένη περίπτωση επιλέγουμε Wireless.



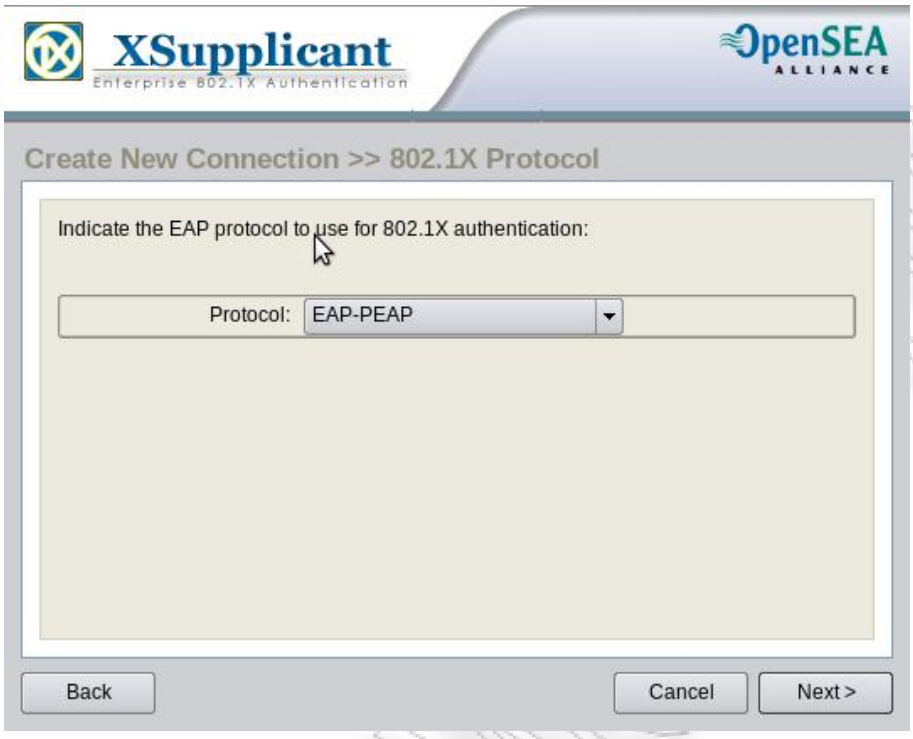
Εικόνα 58 Επιλογή τύπου δικτύου

Επιλέγουμε από τα ορατά δίκτυα, που εκπέμπουν, το επιθυμητό δίκτυο.



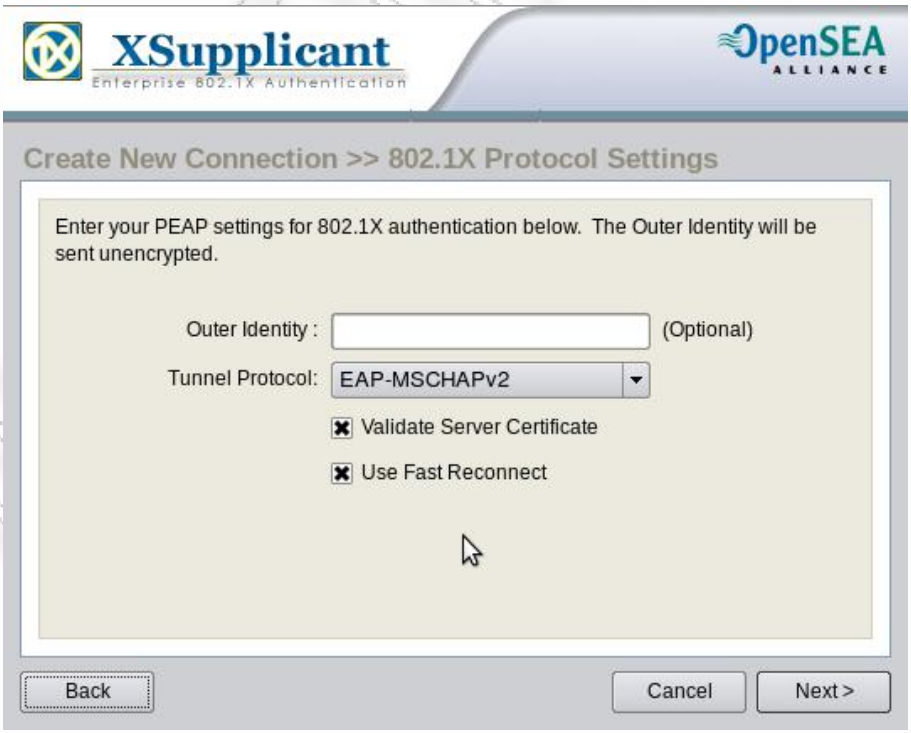
Εικόνα 59 Επιλογή ασύρματου δικτύου

Στη συνέχεια επιλέγουμε τη μέθοδο αυθεντικοποίησης που θα χρησιμοποιηθεί.

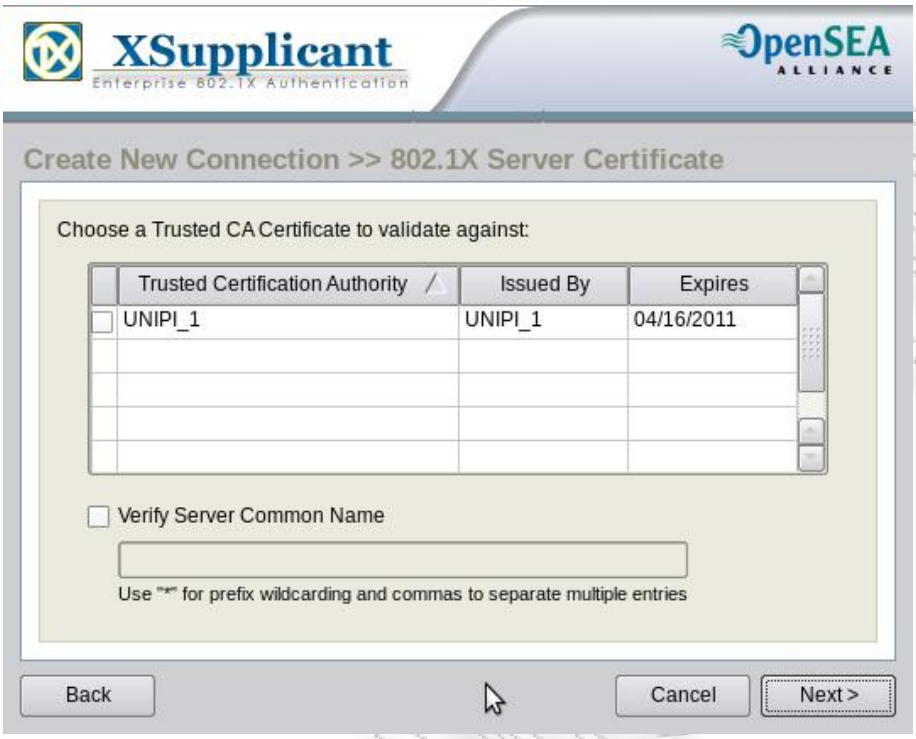


Εικόνα 60 Επιλογή EAP μεθόδου

Επιλέγουμε το Tunnel Protocol που θα χρησιμοποιηθεί καθώς και το πιστοποιητικό CA και πελάτη (αν απαιτείται).

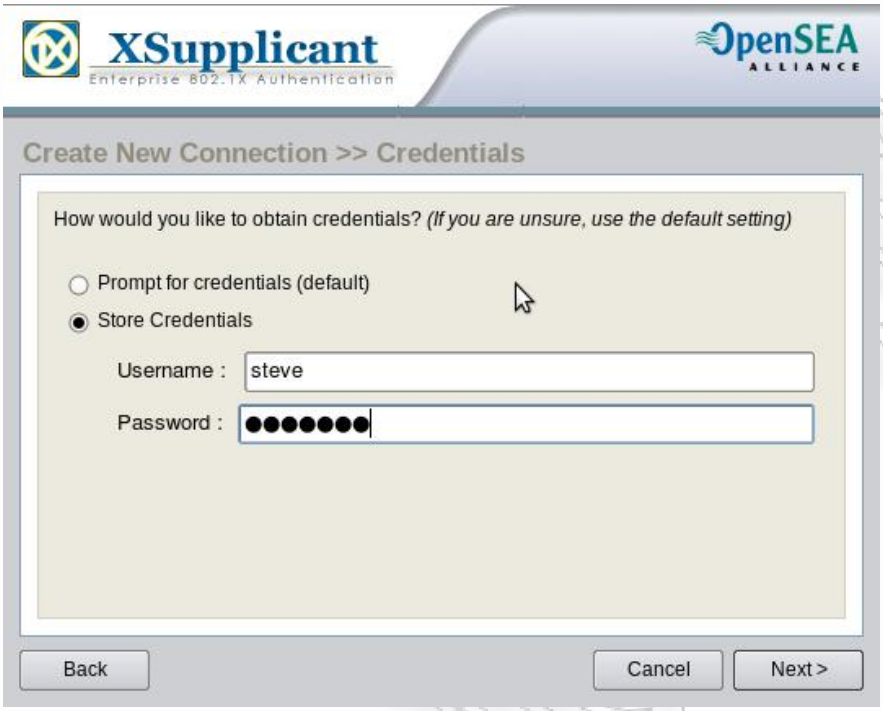


Εικόνα 61 Ρυθμίσεις μεθόδου



Εικόνα 62 Επιλογή πιστοποιητικού CA

Εικόνα 63 Επιλογές σύνδεσης



Εικόνα 64 Καταχώρηση ονόματος χρήστη και συνθηματικού πρόσβασης

Το επόμενο βήμα περιλαμβάνει τη επιλογή ονόματος για την αποθήκευση των ρυθμίσεων.

Τέλος, από την αρχική οθόνη του XSupplicant επιλέγουμε τις ρυθμίσεις που μόλις δημιουργήσαμε και στη συνέχεια Connect.

7 Επίλογος

Το Extensible Authentication Protocol είναι μία ιδιαίτερα ευέλικτη δομή για την ασφάλή αυθεντικοποίηση κατά την πρόσβαση σε ασύρματα δίκτυα. Είναι δυνατή η επιλογή από μία πληθώρα μεθόδων, κάποιες εκ των οποίων εμφανίζουν σημαντικές αδυναμίες. Κάποιες μέθοδοι είναι τρωτές σε dictionary επιθέσεις, και man-in-the-middle επιθέσεις.

Κατά την επιλογή μίας λύσης βασισμένης στο EAP θα πρέπει να λαμβάνονται υπόψην, οι αδυναμίες των μεθόδων, ο βαθμός δυσκολίας της υλοποίησης, καθώς και το κόστος υλοποίησης, σε σχέση με το κόστος που θα προκείψει από την πιθανή παραβίαση ενός τέτοιου μηχανισμού αυθεντικοποίησης. Στις προτεινόμενες υλοποιήσεις είναι το EAP-TLS και αν η ανάπτυξη υποδομής δημόσιου κλειδιου θεωρηθεί περιττό κόστος το TTLS και το PEAP.

ΠΑΡΑΡΤΗΜΑ Ι – Configuration

radiusd.conf

```
# -*- text -*-

##

## radiusd.conf -- FreeRADIUS server configuration file.


prefix = /usr/local

exec_prefix = ${prefix}

sysconfdir = ${prefix}/etc

localstatedir = ${prefix}/var

sbindir = ${exec_prefix}/sbin

logdir = ${localstatedir}/log/radius

raddbdir = ${sysconfdir}/raddb

radacctdir = ${logdir}/radacct


name = radiusd


confdir = ${raddbdir}

run_dir = ${localstatedir}/run/${name}


db_dir = ${raddbdir}


libdir = ${exec_prefix}/lib


pidfile = ${run_dir}/${name}.pid


max_request_time = 30
```

```
cleanup_delay = 5

max_requests = 1024

listen {
        type = auth

        ipaddr = *
        port = 0

}

listen {
        ipaddr = *
        port = 0
        type = acct
}

hostname_lookups = no

allow_core_dumps = no

regular_expressions     = yes
extended_expressions    = yes

log {
```

```
        destination = files
        file = ${logdir}/radius.log

        syslog_facility = daemon
        stripped_names = no

        auth = no

        auth_badpass = no
        auth_goodpass = no
}
checkrad = ${sbindir}/checkrad
security {
        max_attributes = 200
        reject_delay = 1
        status_server = yes
}
proxy_requests  = yes
$INCLUDE proxy.conf

$INCLUDE clients.conf

thread pool {
        start_servers = 5
        max_servers = 32

        min_spare_servers = 3
```

```
        max_spare_servers = 10

        max_requests_per_server = 0
}
modules {

        $INCLUDE ${confdir}/modules/

        #  Extensible Authentication Protocol
        #
        #  For all EAP related authentications.
        #  Now in another file, because it is very large.
        #
        $INCLUDE eap.conf

}

instantiate {

        exec
        expr
        expiration
        logintime
}
$INCLUDE policy.conf
$INCLUDE sites-enabled/
```

users

```
#
#       Please read the documentation file ../doc/processing_users_file,
#       or 'man 5 users' (after installing the server) for more information.


steve   Cleartext-Password := "testing"
        MS-CHAP-Use-NTLM-Auth := No
"user@example.com"    Service-Type == Framed-User
pelatis  Service-Type == Framed-User
DEFAULT         Framed-Protocol == PPP
        Framed-Protocol = PPP,
        Framed-Compression = Van-Jacobson-TCP-IP
DEFAULT         Hint == "CSLIP"
        Framed-Protocol = SLIP,
        Framed-Compression = Van-Jacobson-TCP-IP
```

clients.conf

```
# -*- text -*-
## clients.conf -- client configuration directives
client localhost {


        ipaddr = 127.0.0.1
        secret          = testing123
        require_message_authenticator = no
        nastype     = other         # localhost isn't usually a NAS...
}


client RadiusAP {
      ipaddr = 192.168.2.11
```

```
        secret          = radiussecret
            require_message_authenticator = no
            nastype     = other

}
```

eap.conf

```
##  eap.conf -- Configuration for EAP types (PEAP, TTLS, etc.)
            eap {

                        default_eap_type = md5
                        timer_expire     = 60
                        ignore_unknown_eap_types = no

                        cisco_accounting_username_bug = no

                        max_sessions = 4096
                        md5 {
                        }

                        leap {
                        }
                        gtc {

                                    auth_type = PAP
                        }
                        tls {
```

```
        certdir = ${confdir}/certs
        cadir = ${confdir}/certs

        private_key_password = mypass
        private_key_file = ${certdir}/server.pem

        certificate_file = ${certdir}/server.pem
        CA_file = ${cadir}/ca.pem
        dh_file = ${certdir}/dh
        random_file = ${certdir}/random
        cipher_list = "DEFAULT"
        make_cert_command = "${certdir}/bootstrap"
        cache {
            #
            enable = no
            lifetime = 24 # hours
            max_entries = 255
        }
    }

    ttls {
        default_eap_type = md5
        copy_request_to_tunnel = no
        use_tunneled_reply = no
        virtual_server = "inner-tunnel"
```

```
                }
                peap {
                        default_eap_type = mschapv2
                        copy_request_to_tunnel = no
                        use_tunneled_reply = no
                        virtual_server = "inner-tunnel"
                }
                mschapv2 {
                }
        }
```

ca.cnf

```
[ ca ]
default_ca              = CA_default


[ CA_default ]
dir                     = ./
certs                   = $dir
crl_dir                 = $dir/crl
database                = $dir/index.txt
new_certs_dir           = $dir
certificate             = $dir/ca.pem
serial                  = $dir/serial
crl                     = $dir/crl.pem
private_key             = $dir/ca.key
RANDFILE                = $dir/.rand
```

```
name_opt                = ca_default

cert_opt                = ca_default

default_days            = 365

default_crl_days        = 365

default_md              = md5

preserve                = no

policy                  = policy_match


[ policy_match ]

countryName             = match

stateOrProvinceName     = match

organizationName        = match

organizationalUnitName          = optional

commonName              = supplied

emailAddress            = optional


[ policy_anything ]

countryName             = optional

stateOrProvinceName     = optional

localityName            = optional

organizationName        = optional

organizationalUnitName          = optional

commonName              = supplied
```

```
emailAddress            = optional


[ req ]

prompt                  = no

distinguished_name      = certificate_authority

default_bits            = 2048

input_password          = mypass

output_password                 = mypass

x509_extensions                 = v3_ca


[certificate_authority]

countryName             = GR

stateOrProvinceName     = Attica

localityName            = Athens

organizationName        = UNIPI

emailAddress            = zmouto@gmail.com

commonName              = "My Certificate Authority"


[v3_ca]

subjectKeyIdentifier    = hash

authorityKeyIdentifier  = keyid:always,issuer:always

basicConstraints        = CA:true
```

client.cnf

```
[ ca ]

default_ca              = CA_default


[ CA_default ]

dir                     = ./

certs                   = $dir

crl_dir                 = $dir/crl

database                = $dir/index.txt

new_certs_dir           = $dir

certificate             = $dir/server.pem

serial                  = $dir/serial

crl                     = $dir/crl.pem

private_key             = $dir/server.key

RANDFILE                = $dir/.rand

name_opt                = ca_default

cert_opt                = ca_default

default_days            = 365

default_crl_days        = 30

default_md              = md5

preserve                = no

policy                  = policy_match


[ policy_match ]

countryName             = match

stateOrProvinceName     = match

organizationName        = match
```

```
organizationalUnitName          = optional
commonName              = supplied
emailAddress            = optional

[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName          = optional
commonName              = supplied
emailAddress            = optional

[ req ]
prompt                  = no
distinguished_name      = client
default_bits            = 2048
input_password          = mypass
output_password                 = mypass

[client]
countryName             = GR
stateOrProvinceName     = Attica
localityName            = Athens
organizationName        = UNIPI
emailAddress            = zmouto@gmail.com
commonName              = pelatis
```

server.cnf

```
[ ca ]

default_ca              = CA_default



[ CA_default ]

dir                     = ./

certs                   = $dir

crl_dir                 = $dir/crl

database                = $dir/index.txt

new_certs_dir           = $dir

certificate             = $dir/server.pem

serial                  = $dir/serial

crl                     = $dir/crl.pem

private_key             = $dir/server.key

RANDFILE                = $dir/.rand

name_opt                = ca_default

cert_opt                = ca_default

default_days            = 365

default_crl_days        = 30

default_md              = md5

preserve                = no

policy                  = policy_match
```

```
[ policy_match ]

countryName             = match

stateOrProvinceName     = match

organizationName        = match

organizationalUnitName          = optional

commonName              = supplied

emailAddress            = optional


[ policy_anything ]

countryName             = optional

stateOrProvinceName     = optional

localityName            = optional

organizationName        = optional

organizationalUnitName          = optional

commonName              = supplied

emailAddress            = optional


[ req ]

prompt                  = no

distinguished_name      = server

default_bits            = 2048

input_password          = mypass

output_password                 = mypass
```

```
[server]

countryName             = GR

stateOrProvinceName     = Attica

localityName            = Athens

organizationName        = UNIPI

emailAddress            = zmouto@gmail.com

commonName              = "My Server Certificate"
```

ΠΑΡΑΡΤΗΜΑ ΙΙ – Ευρετήριο όρων και ακρωνυμίων

acknowledgement frames	πλαίσια επιβεβαίωσης, αποστέλλονται ως αποδεικτικά λήψης ενός πλαισίου
antenna diversity	διάταξη που χρησιμοποιεί δύο η περισσότερες κεραίες με σκοπό τη βελτίωση της ποιότητας και της αξιοπιστίας μιας ασύρματης ζεύξης.
Apache	λογισμικό εξυπηρετητή παγκοσμίου ιστού, Web Server
Apache module	πακέτο πρόσθετης λειτουργίας για τον Apache
Application Layer	επίπεδο εφαρμογής
ARP poisoning ή ARP spoofing	τύπος παραβίασης σε δίκτυο υπολογιστών το οποίο βασίζεται στο πρωτόκολλο ARP. Ο κακόβουλος χρήστης μπορεί, μεταδίδοντας λανθασμένα πακέτα ARP, να μπερδέψει άλλους host ώστε να στείλουν τα πλαίσια δεδομένων (data frames) τους σε άλλον υπολογιστή χωρίς να το αντιληφθούν
MAC address Spoofing	Τεχνική για την αλλαγή της MAC διεύθυνσης μιας δικτυακής συσκευής με μία άλλη. Με τη μέθοδο αυτή δύναται να παραβιαστούν λίστες ελέγχου πρόσβασης σε εξυπηρετητές ή δρομολογητές.
association	Συσχέτιση
authentication	Αυθεντικοποίηση
authentication server	εξυπηρετητής αυθεντικοποίησης
Bit	δυαδικό ψηφίο
block cipher	Κρυπτογράφημα Δέσμης
BSD licence	Χαρακτηρισμός άδειας χρήσης που προβλέπει ελάχιστους περιορισμούς.
Certificate	Πιστοποιητικό
Certificate authorities,CAs	Αρχή πιστοποίησης
Certificate Authority certificates	πιστοποιητικό αρχής πιστοποίησης

Challenge	Πρόκληση
challenge text	κείμενο πρόκλησης
Client certificates	πιστοποιητικό πελάτη
data frames	πλαίσια δεδομένων
Data Link Layer, DLL	επίπεδο ζεύξης δεδομένων
Datagrams	Δεδομενογραφήματα
deauthentication	διαδικασία παύσης αυθεντικοποίησης
Denial of Service	Άρνηση υπηρεσίας
Diameter	Πρωτόκολλό για την αυθεντικοποίηση, rfc 3588
Disassociation	Αποσυσχέτιση
EAP	Extensible Authentication Protocol
EAP-MD5	Μέθοδος του EAP
EAPOL	EAP over LANs, το IEEE 802.1x
EAP-TLS (EAP with Transport Layer Security)	μέθοδος του EAP
EAP-TTLS (EAP- Tunneled TLS)	μέθοδος του EAP
eavesdropping	"κρυφάκουσμα", διαδικασία κατά την οποία ένας επιτιθέμενος συλλέγει την πληροφορία που ανταλλάσσεται μεταξύ δύο υπολογιστών υπηρεσίας
flow control	έλεγχος ροής
fragmentation	κατάτμηση πακέτων
FreeRADIUS	Υλοποίηση ανοιχτού κώδικα RADIUS εξυπηρετητή
FreeRADIUS Project	Η ομάδα ανάπτυξης του FreeRADIUS
FTP	File Transfer Protocol, Πρωτόκολλό μεταφοράς αρχείων
hash	Κρυπτογραφική συνάρτηση
host	υπολογιστής υπηρεσίας
HTTP	Hypertext Transfer Protocol, πρωτόκολλο μεταφοράς υπερκειμένου
ICMP	Internet Control Message Protocol
Internet Layer	επίπεδο διαδικτύου
IP (Internet Protocol).	πρωτόκολλο διαδικτύου
Public Key Infrastracture	Υποδομή δημόσιου κλειδιού

kismet	Λογισμικό για την ανάλυση της κίνησης σε ασύρματα δίκτυα
LEAP	Lightweight EAP, μέθοδος
Man- In-The-Middle attack	Επίθεση ενδιάμεσου ατόμου
manipulating	χειραγώγηση, χρησιμοποιείται για τροποποίηση δεδομένων
Medium Access Control, MAC	έλεγχος πρόσβασης στο μέσο
netstumbler	Εργαλείο ανίχνευσης ασύρματων δικτύων
Network Access Layer	επίπεδο πρόσβασης δικτύου
Network Access Server	NAS
Network Layer	επίπεδο δικτύου
OPEN1X	Λογισμικό για Linux που υλοποιεί τον 801.1X supplicant
OSI	Open Systems Interconnection, πρότυπο για την επικοινωνία μεταξύ υπολογιστών
password	κωδικός χρήστη
PEAP (Protected EAP)	μέθοδος του EAP
peer	Κόμβος
Physical Layer	φυσικό επίπεδο
POP	Post Office Protocol, πρωτοκολλο που χρησιμοποιείται στην ηλεκτρονική αλληλογραφία
PPP (Point –to – Point Protocol)	Πρωτοκολλο ζεύξης σημείο προς σημείο
Presentation Layer	επίπεδο παρουσίασης
privacy	εξασφάλιση ιδιωτικού απορρήτου
RADIUS	Remote Authentication Dial In User Service
RADIUS Server	Εξυπηρετητής αυθεντικοποίησης
radtest	εφαρμογή για τον έλεγχο λειτουργίας του freeRADIUS
reassociation	Επανασυσχέτιση
Request	Αίτηση
Server certificates	πιστοποιητικά εξυπηρετητή
Session Layer	Επίπεδο συνόδου
shared secret	Κοινό μυστικό
sniffing	Συλλογή πακέτων που ανταλλάσσονται μεταξύ

	υπολογιστών
State attribute	όρισμα κατάστασης
stream cipher	Κρυπτογράφημα ροής
subnet mask	Μάσκα υποδικτύου
SYN flooding	επίθεση που πραγματοποιείται με αποστολή μεγάλου αριθμού αιτήσεων σύνδεσης
TCP	Transmission Control Protocol,
TCP/IP	πρότυπο για την επικοινωνία μεταξύ υπολογιστών
TKIP και το CCMP	
Transport Layer	επίπεδο μεταφοράς
UDP	User Datagrams Protocol,
URI	Uniform Resource Identifier
WEP	Wired Equivalent Privacy, πρωτόκολλο ασφαλείας για ασύρματη επικοινωνία
Wi-Fi	Το IEEE 802.11
WIRE1.x	Εφαρμογή πελάτη που υλοποιεί το EAP
Wireshark	Εργαλείο ανάλυσης πρωτόκολλων επικοινωνίας
log file	αρχείο καταγραφής
IEEE 802.11	Πρότυπο για ασύρματα τοπικά δίκτυα
attribute-value pairs (AVPs)	Ζευγάρια ορίσματος τιμής
openssl	Λογισμικό που υλοποιεί το ssl, χρησιμοποιείται για την έκδοση και διαχείριση πιστοποιητικών

Βιβλιογραφία

Βιβλία

1. Andrew S. Tanabeum (2000) Δίκτυα *Υπολογιστών,* Εκδόσεις Παπασωτηρίου, Αθήνα
2. Jonathan,Hassell (2002) , *RADIUS,* O'Reilly Media, Inc, USA
3. M.,Nakhjiri (2005), *AAA and Network Security for Mobile Access,* John Wiley and Sons Ltd

Internet sites

1. "Μοντέλο αναφοράς OSI" *Wikipedia.org* [online] http://el.wikipedia.org/wiki/%CE%9C%CE%BF%CE%BD%CF%84%CE%AD%CE%BB%CE%BF_%CE%B1%CE%BD%CE%B1%CF%86%CE%BF%CF%81%CE%AC%CF%82_OSI
2. "Wireless Hacking Tools" Author: Michael Roche [online] http://www.cs.wustl.edu/~jain/cse571-07/ftp/wireless_hacking/index.html
3. "Extensible Authentication Protocol" [online] http://tools.ietf.org/html/rfc3748
4. "Extensible Authentication Protocol Key Management Framework" [online] http://tools.ietf.org/html/rfc5247
5. "Extensible Authentication Protocol" [online] http://en.wikipedia.org/wiki/Extensible_Authentication_Protocol
6. "Extensible Authentication Protocol Method requirements for Wireless LANs" [online] http://tools.ietf.org/html/rfc4017
7. "Lightweight Extensible Authentication Protocol" [online] http://en.wikipedia.org/wiki/Lightweight_Extensible_Authentication_Protocol
8. "The EAP-TLS Authentication Protocol" [online] http://tools.ietf.org/html/rfc5216
9. "Remote Authentication Dial In User Service (RADIUS)" [online] http://tools.ietf.org/html/rfc2865
10. "RADIUS" [online] http://en.wikipedia.org/wiki/RADIUS
11. RADIUS Extensions [online] http://tools.ietf.org/html/rfc2869
12. "How to: EAP /TLS" *Author: Ken Roser* [online] http://freeradius.org/doc/EAPTLS.pdf
13. "Understanding EAP-MD5 Authentication with RADIUS" Author: Sudipto Chakraborty [online] http://images.google.gr/url?sa=t&source=web&ct=res&cd=1&ved=0CAYQFjAA&url=http%3A%2F%2Ftim069.free.fr%2Fwireless%2FUnderstanding%2520EAP-

MD5%2520Authentication%2520with%2520RADIUS-1.doc&ei=KleGS9e-L4-b_Qaa1sSgDw&usg=AFQjCNGjbxtqMwTT9jfI0vuquXXVY_EL2w&sig2=m6Krb0G--dUyqj1_S4BukA

14. "802.1X Port-Based Authentication HOWTO", Author: Lars Strand [online] http://tldp.org/HOWTO/html_single/8021X-HOWTO/#auth
15. "FreeRADIUS Version 2 Documentation" [online] http://freeradius.org/doc/
16. WIRE1x how to [online] http://wire.cs.nthu.edu.tw/wire1x/step_vista_1_1_1_en.htm
17. "Design and Implementation of WIRE1x" Authors :Yu-Ping Wang, Yi-Wen Liu, Jyh-Cheng Chen [online] https://wire.cs.nthu.edu.tw/wire1x/TANET03.pdf