ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΑΣΦΑΛΕΙΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Εγκατάσταση και Παραμετροποίηση Μηχανισμών Ασφάλειας FreeRADIUS/MySQL-FreeRADIUS/LDAP-OpenVPN-WiKID-PhoneFACTOR

Διπλωματική Εργασία

Τσιρτσής Σταύρος

Α/Μ: MTE1069

Επιβλέπων: Επικ. Καθηγητής Ξενάκης Χρήστος

Πειραιάς, Μάρτιος 2012

"*Η σελίδα αυτή παραμένει σκόπιμα κενή.*"

Πρόλογος

Στη σημερινή εποχή των ψηφιακών υπηρεσιών, η ανάγκες ταυτοποίησης και διασφάλισης της ταυτότητας μας, κρίνονται απαραίτητες στις καθημερινές μας ανάγκες. Οι ανάγκες αυτές περιλαμβάνουν ένα μεγάλο εύρος ψηφιακών υπηρεσιών, όπως τραπεζικές συναλλαγές μέσω ηλεκτρονικής τραπεζικής, εμπορικές συναλλαγές σε ηλεκτρονικά καταστήματα με χρήση πιστωτικών καρτών, υπηρεσίες ηλεκτρονικής αλληλογραφίας, υπηρεσίες κοινωνικών δικτύων, συνένωση απομακρυσμένων ιδιωτικών δικτύων, πρόσβαση σε Η/Υ, πρόσβαση σε ασύρματα εταιρικά δίκτυα κλπ. Για την επίτευξη της ταυτοποίησης των χρηστών, χρησιμοποιούνται συνήθως συμβατικές μέθοδοι, όπως διαπιστευτήρια ονόματος χρήστη και κωδικού σε ηλεκτρονικές υπηρεσίες αυθεντικοποίησης.

Δεδομένου ότι οι μέθοδοι αυτές είναι πλέον ευάλωτες και μη απόλυτα ασφαλείς στις υπηρεσίες αυτές, έχουν επινοηθεί κάποιες νέες μέθοδοι ταυτοποίησης που δεν χρησιμοποιούνται ακόμη ευρέως. Οι μέθοδοι αυτές μπορούν να περιλαμβάνουν τη χρήση ψηφιακών πιστοποιητικών (RADIUS EAP-TLS) μέχρι και πολλαπλούς παράγοντες αυθεντικοποίησης, όπως συνδυασμός ταυτοποίησης με κωδικούς μιας χρήσης με χρονική λήξη OTP (One Time Password), άμεση τηλεφωνική κλήση στους τηλεφωνικούς αριθμούς των χρηστών από κέντρα δεδομένων για αυθεντικοποίηση και χρήση βιομετρικών χαρακτηριστικών όπως η αναγνώριση φωνής (PhoneFACTOR).

Σκοπός της διπλωματικής εργασίας αυτής είναι να επιδείξει την ασφάλεια και την ευκολία που παρέχουν οι διάφοροι αυτοί μηχανισμοί ασφάλειας με την εγκατάσταση και παραμετροποίηση τους, χωρίς την ανάγκη κάποιες φορές της χρήσης διαπιστευτηρίων απομνημόνευσης ή επιπρόσθετων συσκευών παραγωγής κωδικών, παρά με τη βοήθεια μόνο ενός κινητού τηλεφώνου που είναι πλέον ευρέως διαδεδομένο. Οι μηχανισμοί ασφάλειας που μελετούνται σε αυτή τη διπλωματική εργασία είναι οι *FreeRADIUS-MySQL, FreeRADIUS-LDAP, OpenVPN, WiKID* και *PhoneFACTOR.* Οι μηχανισμοί μελετούνται μεμονωμένοι, αλλά και με διάφορα σενάρια σε μεταξύ τους λειτουργία μέσω σενεργασίας των μηχανισμών.

Ευχαριστίες

Παρουσιάζοντας την παρούσα διπλωματική εργασία, θα ήθελα να ευχαριστήσω όλους όσους συνετέλεσαν στην ολοκλήρωσή της. Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας μου, Επίκουρο Καθηγητή κ. Χρήστο Ξενάκη και τον Μεταδιδακτορικό Ερευνητή κ. Χριστόφορο Νταντογιάν, για την πολύτιμη βοήθεια και καθοδήγησή τους κατά τη διάρκεια της μελέτης αυτής.

Ευχαριστώ όλους τους καθηγητές και συμφοιτητές του Μεταπτυχιακού Προγράμματος Ασφάλειας Ψηφιακών Συστημάτων που μοιραστήκανε τη γνώση τους μαζί μου.

Τέλος ευχαριστώ τους γονείς μου για την ηθική υποστήριξή τους.

Πίνακας Περιεχομένων

Λίστα Εικόνων

Λίστα Πινάκων

Λίστα Σχημάτων

Περίληψη

Η διπλωματική αυτή εργασία αποτελείται από πέντε ξεχωριστές ενότητες, που η καθεμία περιγράφει την εγκατάσταση και παραμετροποίηση διαφορετικών μηχανισμών ασφαλείας, όπου ως αυτόνομοι ή συνδυασμένοι μεταξύ τους, είναι απαραίτητοι ώστε να παρέχουν τη μέγιστη ασφάλεια σε χρήστες διάφορων υπηρεσιών στις καθημερινές απαιτήσεις αυθεντικοποίησης τους.

Στο πρώτο κεφάλαιο περιγράφεται η εγκατάσταση και παραμετροποίηση του μηχανισμού αυθεντικοποίησης FreeRADIUS, όπου χρησιμοποιεί τη βάση δεδομένων MySQL και για τη διαχείρισή της χρησιμοποείται η δικτυακή κονσόλα phpMyAdmin . Η εγκατάσταση έγινε σε ένα τοπικό δίκτυο υλοποιώντας ένα σενάριο πρόσβασης σε ένα Wi-Fi Access Point, χρησιμοποιώντας αυθεντικοποίηση των χρηστών μέσω των πρωτοκόλλων PAP, CHAP και EAP-TLS. Στην αυθεντικοποίηση μέσω πρωτόκολλου EAP-TLS, χρησιμοποιούνται μόνο ψηφιακά πιστοποιητικά, χρησιμοποιώντας τη βιβλιοθήκη κρυπτογράφησης OpenSSL για τη δημιουργία τους.

Στο δεύτερο κεφάλαιο περιγράφεται η εγκατάσταση και παραμετροποίηση του μηχανισμού αυθεντικοποίησης FreeRADIUS, όπου χρησιμοποιεί το πρωτόκολλο LDAP (Light Directory Access Protocol). Η βάση δεδομένων του OpenLDAP που έχει εγκατασταθεί στον ίδιο εξυπηρετητή, γίνεται ευκολότερα διαχειρίσιμη εγκαθιστώντας τη δικτυακή κονσόλα phpLDAPAdmin. Οι υπηρεσίες καταλόγου βασίζονται σε μία βάση δεδομένων, η οποία οργανώνει τις εγγραφές και παρέχει βελτιστοποιημένες διαδικασίες ανάγνωσης και αναζήτησης δεδομένων.

Στο τρίτο κεφάλαιο αναλύεται ο μηχανισμός δύο παραγόντων αυθεντικοποίησης WiKID, που χρησιμοποιείται ως λογισμικό παραγωγής κωδικών μιας χρήσης OTP (One Time Password). Ο πρώτος παράγοντας αυθεντικοποίησης είναι το PIN (Personal Identification Number) του αντίστοιχου κάθε φορά τομέα WiKID που ανήκει και πρέπει να γνωρίζει ο χρήστης και ο δεύτερος παράγοντας είναι ο εξαφήφιος κωδικός μίας χρήσης OTP που δημιουργείται με χρονική περίοδο λήξης. Γίνεται αναλυτική περιγραφή της εγκατάστασης του WiKID εξυπηρετητή και του WiKID πελάτη και στη συνέχεια γίνεται έλεγχος αυθεντικοποίησης με την εφαρμογή NTRadPing, μέσω πρωτοκόλλου RADIUS που είναι εγκατεστημένο ως υπηρεσία στον WiKID εξυπηρετητή.

Στο τέταρτο κεφάλαιο γίνεται ανάλυση της εγκατάστασης και παραμετροποίησης του λογισμικού OpenVPN, όπου εφαρμόζει τεχνικές εικονικών ιδιωτικών δικτύων VPN (Virtual Private Network) και βασίζεται στην εγκατάσταση ενός SSL (Secure Sockets Layer) τούνελ εικονικού ιδιωτικού δικτύου (VPN tunnel). Χρησιμοποιείται για τη δημιουργία ασφαλών συνδέσεων από σημείο σε σημείο ή από τοποθεσία σε τοποθεσία. Χρησιμοποιεί ένα προσαρμοσμένο πρωτόκολλο ασφάλειας που χρησιμοποιεί πρωτόκολλο SSLv3/TLSv1 για την ανταλλαγή κλειδιών. Για τη δημιουργία κλειδιών και ψηφιακών πιστοποιητικών χρησιμοποιήθηκε εκτενώς η βιβλιοθήκη κρυπτογράφησης OpenSSL.

Στο πέμπτο κεφάλαιο περιγράφεται ο μηχανισμός πολλαπλών παραγόντων αυθεντικοποίησης PhoneFACTOR. Ο μηχανισμός PhoneFACTOR χρησιμοποιεί αυτοματοποιημένες τηλεφωνικές κλήσεις ή SMS σε σταθερά και κινητά τηλέφωνα, για την εξακρίβωση της ταυτότητας των χρηστών και την πρόσβαση τους σε διάφορες εφαρμογές που απαιτούν αυθεντικοποίηση. Χρησιμοποιώντας τον τηλεφωνικό αριθμό του χρήστη (πρώτος παράγοντας), ένα PIN (δεύτερος παράγοντας) που γνωρίζει ο χρήστης ή/και το βιομετρικό χαρακτηριστικό της φωνή του (τρίτος παράγοντας), ο μηχανισμός υποστηρίζει αυθεντικοποίηση δύο ή τριών παραγόντων αντίστοιχα. Γίνεται περιγραφή

της παραμετροποίησης του PhoneFACTOR Agent και της διαχειριστικής κονσόλας του server. Επίσης περιγράφεται ένα σενάριο αυθεντικοποίησης τεσσάρων παραγόντων που υλοποιήθηκε στην πρώτη περίπτωση, χρησιμοποιώντας το συνδυασμό των δύο μηχανισμών αυθεντικοποίησης PhoneFACTOR και WiKID, για την πρόσβαση του χρήστη στο απομακρυσμένο VPN δίκτυο που βρίσκεται ο OpenVPN server.Στη δεύτερη εναλλακτική περίπτωση, χρησιμοποιούνται οι δύο μηχανισμοί PhoneFACTOR και FreeRADIUS με βάση LDAP χρησιμοποιώντας αυθεντικοποίηση τριών παραγόντων, για την πρόσβαση του χρήστη στο απομακρυσμένο VPN δίκτυο που βρίσκεται ο OpenVPN server.

1 Εγκατάσταση & Παραμετροποίηση FreeRADIUS-MySQL

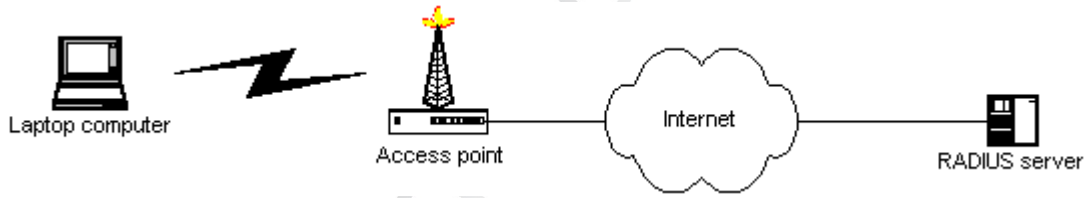
1.1 Εισαγωγή

Σε κάθε ανάγκη ελέγχου της πρόσβασης σε διάφορους υπολογιστικούς πόρους, πρέπει να χρησιμοποιείται ένας μηχανισμός ελέγχου. Ο RADIUS Server είναι ένας μηχανισμός για τη ρύθμιση της πρόσβασης σε ένα δίκτυο υπολογιστών (συνήθως πελάτες εταιρειών), όπου αναλαμβάνει την εξουσιοδότηση τους. Η υπηρεσία του RADIUS αποτελείται από την Αυθεντικοποίηση (*Authentication)* του χρήστη, την εξουσιοδότηση του (*Authorization)* και τον Λογισμό της χρήσης του *(Accounting).*

Authentication είναι η διαδικασία ταυτοποίησης του χρήστη. Ο πιο κοινός τρόπος αυθεντικοποίησης αποτελείται από username και password.

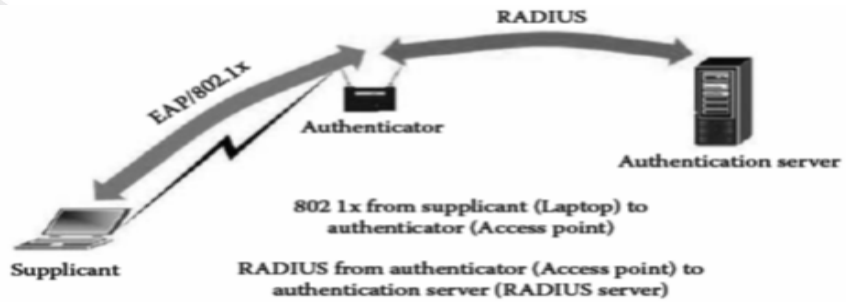
Authorization είναι η διαδικασία καθορισμού ποιων υπηρεσιών επιτρέπονται στο χρήστη να χρησιμοποιήσει και σε ποιο βαθμό. Απαιτείται να έχει προηγηθεί αυθεντικοποίηση του χρήστη από κάποια διαδικασία αυθεντικοποίησης. Το αυθεντικοποιημένο user ID, τότε αυθεντικοποιείται από ένα αρχείο ή ένα πίνακα βάσης δεδομένων ή από υπηρεσίες καταλόγου LDAP.

Accounting είναι η διαδικασία διατήρησης ιστορικού της χρήσης του δικτύου. Καταγράφει την ημερομηνία και ώρα της έναρξης κάθε συνόδου, τη διάρκεια της και τον bytes που μεταφέρονται.



Σχήμα 1-1 RADIUS Server

802.1X Αυθεντικοποίηση είναι ένας μηχανισμός επικύρωσης βασισμένος στις θύρες, που λειτουργεί κάτω από το Extensible Authentication Protocol (EAP). Το 802.1X χρησιμοποιείται για την επικύρωση στις θύρες επικοινωνίας. Αυτό σημαίνει ότι το πρότυπο παίρνει την αίτηση επικύρωσης και αποφασίζει εάν πρέπει να της επιτραπεί ή όχι πρόσβαση στο δίκτυο. Το 802.1X είναι απλά ένας μηχανισμός που απορρίπτει όλη την κίνηση που έχει πρόσβαση σε ένα δίκτυο εκτός από τα EAP πακέτα. Εάν το EAP πει ότι η συσκευή μπορεί να αποκτήσει πρόσβαση στο ασύρματο δίκτυο, το 802.1X πρωτόκολλο λέει στους διακόπτες ή στα σημεία πρόσβασης να επιτρέψουν την κίνηση που προέρχεται από το χρήστη. Το πρωτοκολλό χρησιμοποιεί 2 πρότυπα. Από τον πελάτη στον επικυρωτή το EAP και από τον επικυρωτή στον διακομιστή επικύρωσης το RADIUS (Σχ. 1-2).



Σχήμα 1-2 Πρότυπο 802.1X

Ο FreeRADIUS server είναι μία αρθρωτή, RADIUS σουίτα ανεπτυγμένη και κατανεμημένη υπό την άδεια του GNU General Public License, v2 και είναι ελεύθερη για φόρτωση και χρήση σε Η/Υ. Περιλαμβάνει τον RADIUS server, μια RADIUS client βιβλιοθήκη άδειας BSD , μία PAM βιβλιοθήκη, μία υπομονάδα Apache, πολλές πρόσθετες εφαρμογές σχετικές με το RADIUS και βιβλιοθήκες ανάπτυξης. Ο FreeRADIUS server είναι ο πιο δημοφιλής RADIUS server ανοικτού κώδικα και ο πιο διαδεδομένος RADIUS server.

Στο κεφάλαιο αυτό περιγράφεται η διαδικασία εγκατάστασης του FreeRADIUS server, έκδοσης 2.1.12 σε Λειτουργικό Σύστημα Ubuntu 11.10 (32-bit). Η εφαρμογή βάσης δεδομένων που χρησιμοποιείται για τον FreeRADIUS server είναι η MySQL. Η διαχείριση της MySQL γίνεται μέσω της εφαρμογής phpMyAdmin.

Το Λειτουργικό Σύστημα έχει εγκατασταθεί σε VM (Virtual Machine) στην εφαρμογή VMWARE Workstation 8 με τα παρακάτω χαρακτηριστικά: μνήμη RAM 768 MB, 1 core CPU και 8 GB χωρητικότητας σκληρού δίσκου. Ο χρήστης που έχει δημιουργηθεί για την εγκατάσταση της εφαρμογής είναι ο “radius” και έχει δικαιώματα διαχειριστή. Στην κάρτα δικτύου έχουν οριστεί οι παρακάτω παράμετροι για την επικοινωνία του VM με το υπόλοιπο VLAN:

IP Address: 192.168.1.12, Netmask: 255.255.255.0, Gateway**:** 192.168.1.1, DNS Server: 192.168.1.1

1.2 Εγκατάσταση του LAMP - server (Linux-Apache-MySQL-PHP server)

Για την εγκατάσταση της MySQL και του PHP Server εγκαθίσταται ολόκληρο το πακέτο Linux-Apache-MySQL-PHP server για να αποφευχθούν αργότερα τυχόν σφάλματα που οφείλονται σε εξαρτήσεις μεταξύ των προγραμμάτων.

Αρχικά εγκαθίσταται η εφαρμογή *Tasksel* όπου είναι μια εφαρμογή Package Manager και χρησιμοποιείται για την εγκατάσταση εφαρμογών. Χρησιμοποιώντας τη γραμμή εντολών εγκαθίσταται η εφαρμογή Tasksel:

```
sudo apt-get install tasksel
```

Στη συνέχεια γίνεται εγκατάσταση του LAMP server χρησιμοποιώντας την παρακάτω εντολή:

```
sudo tasksel install lamp-server
```

Κατά την εγκατάσταση ζητείται να εισαχθεί το password του root χρήστη για τον MySQL Server, όπου αυτό έχει οριστεί ως *“setupRADIUS”.* (Για την επιλογή του LAMP server πρέπει να πιεστεί το πλήκτρο Space)

1.3 Εγκατάσταση του FreeRADIUS server

Για την εγκατάσταση του FreeRADIUS server απαιτείται στη γραμμή εντολών να δοθεί η παρακάτω εντολή :

```
sudo apt-get install freeradius
```

Κατά την εγκατάσταση όπου ζητείται, εισάγεται το password που ορίζεται.

1.4 Εγκατάσταση του FreeRADIUS με αυθεντικοποίηση LDAP

Για την υποστήριξη του freeRADIUS server ώστε να χρησιμοποιεί μηχανισμό αυθεντικοποίησης LDAP απαιτείται στη γραμμή εντολών να δοθεί η παρακάτω εντολή για την εγκατάσταση:

sudo apt-get install freeradius-ldap

1.5 Εγκατάσταση του FreeRADIUS για υποστήριξη MySQL

Για την υποστήριξη του freeRADIUS server ώστε να χρησιμοποιεί MySQL βάση δεδομένων, απαιτείται στη γραμμή εντολών να δοθεί η παρακάτω εντολή για την εγκατάσταση:

sudo apt-get install freeradius-mysql

1.6 Επανεκκίνηση του FreeRADIUS service

Μετά το τέλος των παραπάνω εγκαταστάσεων πρέπει να γίνει επανεκκίνηση του FreeRADIUS service με την παρακάτω εντολή:

sudo /etc/init.d/freeradius restart

1.7 Έλεγχος του Radius Server με Radtest

Στη συνέχεια γίνεται έλεγχος του Radius Server για την ορθή ανταπόκριση του, χρησιμοποιώντας την εντολή radtest εισάγοντας το χρήστη, το password, τον τοπικό server,την προεπιλεγμένη πόρτα την προεπιλεγμένη πόρτα και το προεπιλεγμένο μυστικό password:.

Αρχικά στο αρχείο users.conf πρέπει να εισαχθεί η παρακάτω γραμμή για το χρήστη radius:

radius Cleartext-Password := "setupRADIUS"

Για την εντολή του radtest :

radtest radius setupRADIUS localhost 1812 testing123

Το αποτέλεσμα που πρέπει να επιστραφεί είναι το παρακάτω:

```
Sending Access-Request of id 93 to 127.0.0.1 port 1812
User-Name = "radius"
User-Password = "setupRADIUS"
NAS-IP-Address = 127.0.1.1
NAS-Port = 1812
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=93, length=20
```

Αν κατά την εκτέλεση εμφανιστεί το παρακάτω σφάλμα:

```
Radclient: socket: cannot initialize udpfromto: Function not implemented
```

Θα πρέπει στο αρχείο */etc/hosts* , με root χρήστη να χαρακτηριστεί με μορφή σχολίου η παρακάτω γραμμή: *::1 localhost ip6-localhost ip6-loopback*

1.8 Δημιουργία Βάσης Δεδομένων και Χρήστη για το FreeRADIUS server

- Πρόσβαση στη Βάση Δεδομένων MySQL:

```
radius@RADIUS:~$ mysql -u root -p
Enter password: setupRADIUS
```

- Στη συνέχεια γίνεται δημιουργία του χρήστη ”radius”, της Βάσης Δεδομένων “radius” και η απόδοση δικαιωμάτων στη Βάση Δεδομένων για το χρήστη “radius”:

```
mysql> create user 'radius'@'localhost' identified by 'setupRADIUS';
mysql> create database radius;
mysql> grant all privileges on radius.* to 'radius'@'localhost';
mysql> exit
```

1.9 Εγκατάσταση phpMyAdmin

Για μεγαλύτερη ευκολία στην διαχείριση της βάσης δεδομένων στη MySQL και την ύπαρξη γραφικού περιβάλλοντος, γίνεται εγκατάσταση της εφαρμογής phpMyAdmin.

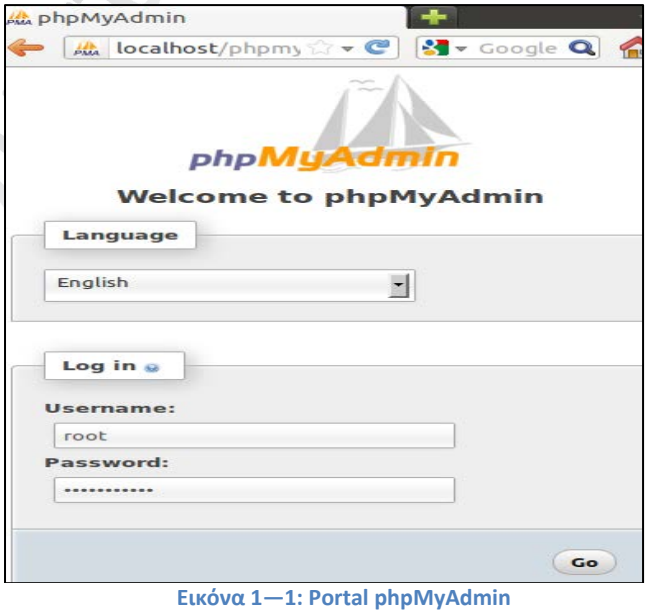
Στη γραμμή εντολών δίνονται οι παρακάτω εντολές:

```
radius@RADIUS:~$ sudo bash
root@RADIUS:~#  apt-get install phpmyadmin
```

1.10 Δημιουργία Πινάκων στη βάση radius

Στη συνέχεια θα πρέπει στη βάση δεδομένων “radius” της MySQL να δημιουργηθούν όλοι οι πίνακες που χρειάζονται για την επικοινωνία με το FreeRADIUS server.

Η δημιουργία των βάσεων θα γίνει μέσω της phpMyAdmin γραφικής εφαρμογής.



Εικόνα 1—1: Portal phpMyAdmin

Σε έναν περιηγητή ιστοσελίδας πληκτρολογείται το url http://localhost/phpmyadmin/ και εμφανίζεται το portal του phpMyAdmin (εικ. 1-1).

Σημείωση: Εάν επιστραφεί " Error 404 " στην ιστοσελίδα , τότε θα πρέπει να διαμορφωθεί το αρχείο apache2.conf , ώστε να μπορέσει να λειτουργήσει με το phpMyAdmin, δίνοντας την παρακάτω εντολή:

```
gksudo gedit /etc/apache2/apache2.conf
```

και συμπεριλαμβάνοντας την παρακάτω γραμμή, στη συνέχεια γίνεται αποθήκευση του αρχείου:

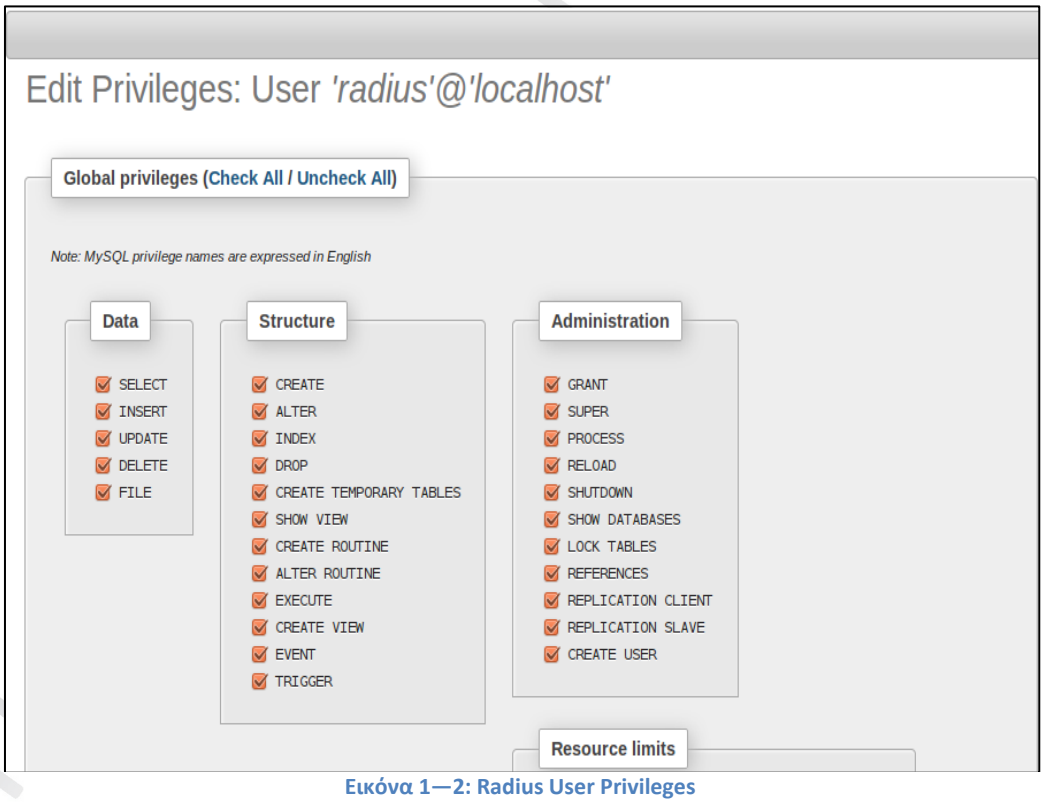
Include /etc/phpmyadmin/apache.conf **[1]**

Τα στοιχεία που πρέπει να εισαχθούν στο Portal είναι αυτά που δημιουργήθηκαν στην παράφραφο 1.7 :

Username : *root*
Password: *setupRADIUS*

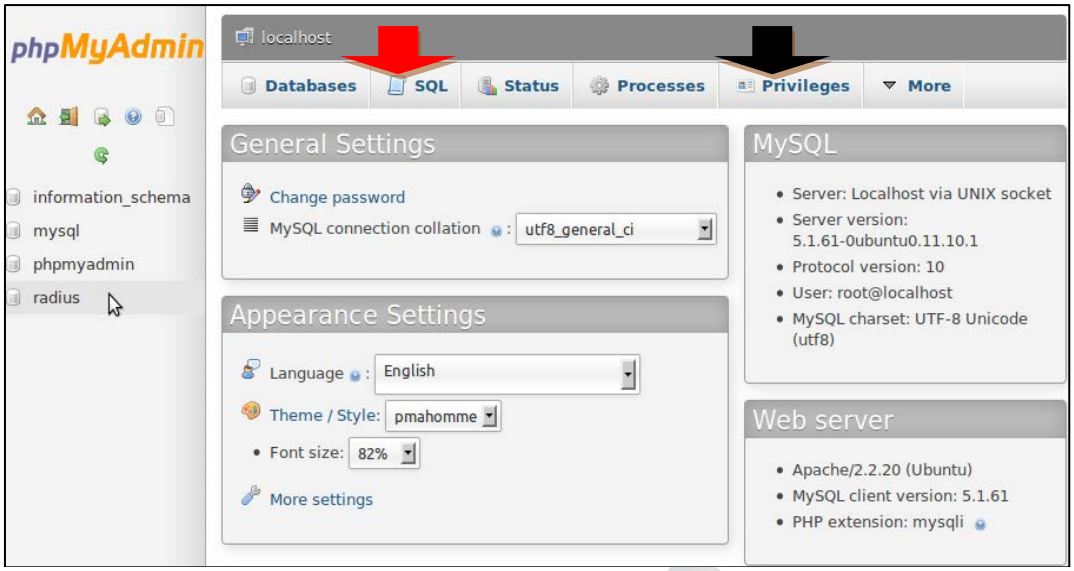
Έπειτα πιέζεται το κομβίο «Go» για να γίνει η εισαγωγή στο portal.

Στη συνέχεια επιλέγεται η βάση "*radius*" και έπειτα επιλέγεται στην πάνω γραμμή το κομβίο «Privileges» (εικ. 1-3) και στη συνέχεια "Edit Privileges", ώστε να γίνει έλεγχος αν ο χρήστης "radius" έχει πλήρη διαχειριστική πρόσβαση στη βάση δεδομένων. (εικ. 1-2)



Εικόνα 1—2: Radius User Privileges

Στη συνέχεια επιλέγεται ξανά η βάση "*radius*" και έπειτα επιλέγεται στην πάνω γραμμή το κομβίο «SQL» ώστε να δημιουργηθούν μέσω query όλοι οι απαραίτητοι πίνακες όπου είναι οι ***nas, radacct, radcheck, radgroupcheck, radgroupreply, radpostauth, radreply*** και ***radusergroup***.



Εικόνα 1—3: SQL-Privileges

Το περιεχόμενο των πινάκων αυτών περιλαμβάνεται στο αρχείο *schema.sql* του freeRADIUS server. Παρακάτω δίνονται οι εντολές που θα χρησιμοποιηθούν για να γίνει αντιγραφή των περιεχομένων από το αρχείο, ώστε να δημιουργηθεί ο κάθε πίνακας στη βάση:

radius@RADIUS:/$ cd etc/freeradius/sql/mysql/
radius@RADIUS:/etc/freeradius/sql/mysql$ sudo bash
radius@RADIUS:/etc/freeradius/sql/mysql$ gedit schema.sql

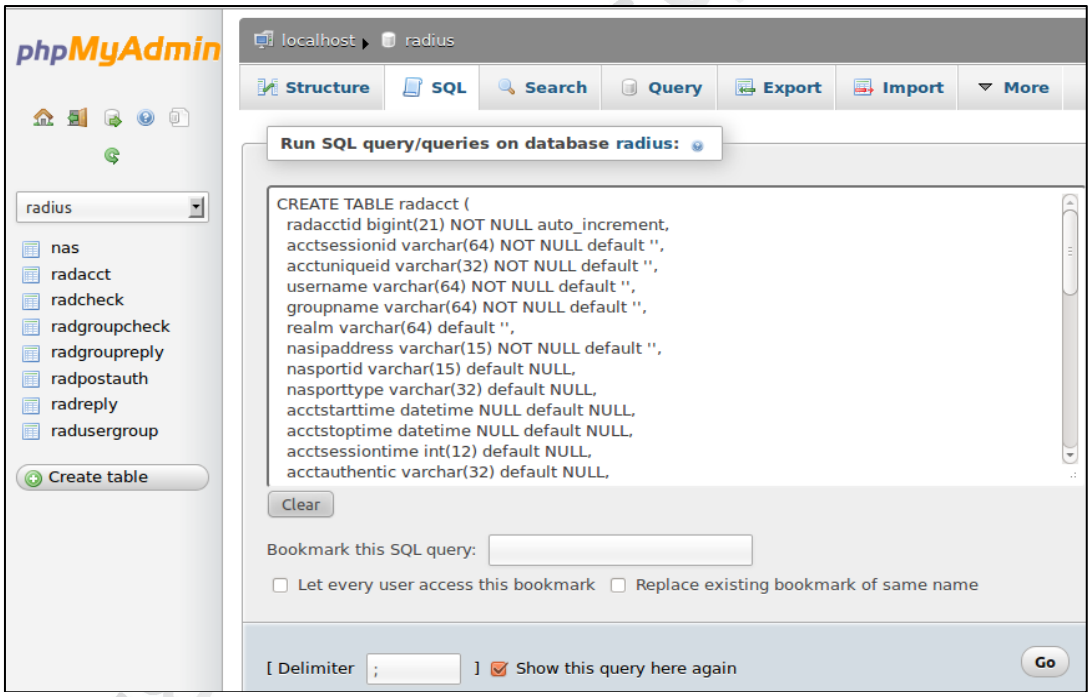
1.10.1 Πίνακας Radacct

Ο πρώτος πίνακας που δημιουργείται είναι ο *Radacct* και το SQL query που δίνεται παρουσιάζεται παρακάτω, όπως φαίνεται και στην Εικ. 1-4:

```
CREATE TABLE radacct (
  radacctid bigint(21) NOT NULL auto_increment,
  acctsessionid varchar(64) NOT NULL default '',
  acctuniqueid varchar(32) NOT NULL default '',
  username varchar(64) NOT NULL default '',
  groupname varchar(64) NOT NULL default '',
  realm varchar(64) default '',
  nasipaddress varchar(15) NOT NULL default '',
  nasportid varchar(15) default NULL,
  nasporttype varchar(32) default NULL,
  acctstarttime datetime NULL default NULL,
  acctstoptime datetime NULL default NULL,
  acctsessiontime int(12) default NULL,
  acctauthentic varchar(32) default NULL,
  connectinfo_start varchar(50) default NULL,
  connectinfo_stop varchar(50) default NULL,
  acctinputoctets bigint(20) default NULL,
  acctoutputoctets bigint(20) default NULL,
  calledstationid varchar(50) NOT NULL default '',
  callingstationid varchar(50) NOT NULL default '',
  acctterminatecause varchar(32) NOT NULL default '',
```

```
servicetype varchar(32) default NULL,
framedprotocol varchar(32) default NULL,
framedipaddress varchar(15) NOT NULL default '',
acctstartdelay int(12) default NULL,
acctstopdelay int(12) default NULL,
xascendsessionsvrkey varchar(10) default NULL,
PRIMARY KEY  (radacctid),
KEY username (username),
KEY framedipaddress (framedipaddress),
KEY acctsessionid (acctsessionid),
KEY acctsessiontime (acctsessiontime),
KEY acctuniqueid (acctuniqueid),
KEY acctstarttime (acctstarttime),
KEY acctstoptime (acctstoptime),
KEY nasipaddress (nasipaddress)
) ;
```

Στο τέλος πρέπει να «πιεστεί» το πλήκτρο Go, ώστε να καταχωρηθεί ο πίνακας στη βάση.



Εικόνα 1—4: SQL Create Table

Αντίστοιχα η ίδια διαδικασία θα πρέπει να γίνει και για όλους τους υπόλοιπους πίνακες.

1.10.2 Πίνακας Radcheck

Στον πίνακα *Radcheck* δίνεται το παρακάτω SQL query:

```
CREATE TABLE radcheck (
  id int(11) unsigned NOT NULL auto_increment,
  username varchar(64) NOT NULL default '',
  attribute varchar(64)  NOT NULL default '',
  op char(2) NOT NULL DEFAULT '==',
  value varchar(253) NOT NULL default '',
```

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

1.10.3 Πίνακας Radgroupcheck

Στον πίνακα *Radgroupcheck* δίνεται το παρακάτω SQL query:

```
CREATE TABLE radgroupcheck (
  id int(11) unsigned NOT NULL auto_increment,
  groupname varchar(64) NOT NULL default '',
  attribute varchar(64)  NOT NULL default '',
  op char(2) NOT NULL DEFAULT '==',
  value varchar(253)  NOT NULL default '',
  PRIMARY KEY  (id),
  KEY groupname (groupname(32))
 ) ;
```

1.10.4 Πίνακας Radgroupreply

Στον πίνακα *Radgroupreply* δίνεται το παρακάτω SQL query:

```
CREATE TABLE radgroupreply (
  id int(11) unsigned NOT NULL auto_increment,
  groupname varchar(64) NOT NULL default '',
  attribute varchar(64)  NOT NULL default '',
  op char(2) NOT NULL DEFAULT '=',
  value varchar(253)  NOT NULL default '',
  PRIMARY KEY  (id),
  KEY groupname (groupname(32))
) ;
```

1.10.5 Πίνακας Radpostauth

Στον πίνακα *Radpostauth* δίνεται το παρακάτω SQL query:

```
CREATE TABLE radpostauth (
  id int(11) NOT NULL auto_increment,
  username varchar(64) NOT NULL default '',
  pass varchar(64) NOT NULL default '',
  reply varchar(32) NOT NULL default '',
  authdate timestamp NOT NULL,
  PRIMARY KEY  (id)
) ;
```

1.10.6 Πίνακας Radreply

Στον πίνακα *Radreply* δίνεται το παρακάτω SQL query:

```
CREATE TABLE radreply (
  id int(11) unsigned NOT NULL auto_increment,
  username varchar(64) NOT NULL default '',
```

```
  attribute varchar(64) NOT NULL default '',
  op char(2) NOT NULL DEFAULT '=',
  value varchar(253) NOT NULL default '',
  PRIMARY KEY  (id),
  KEY username (username(32))
) ;
```

1.10.7 Πίνακας Radusergroup

Στον πίνακα *Radusergroup* δίνεται το παρακάτω SQL query:

```
CREATE TABLE radusergroup (
  username varchar(64) NOT NULL default '',
  groupname varchar(64) NOT NULL default '',
  priority int(11) NOT NULL default '1',
  KEY username (username(32))
) ;
```

1.10.8 Πίνακας Nas

Για τη δημιουργία του πίνακα *Nas πρέπει να αντιγραφεί το περιεχόμενο του αρχείου nas.sql με την παρακάτω εντολή:*

/etc/freeradius/sql/mysql# gedit nas.sql

και στη συνέχεια να δοθεί το παρακάτω SQL query:

```
CREATE TABLE nas (
  id int(10) NOT NULL auto_increment,
  nasname varchar(128) NOT NULL,
  shortname varchar(32),
  type varchar(30) DEFAULT 'other',
  ports int(5),
  secret varchar(60) DEFAULT 'secret' NOT NULL,
  server varchar(64),
  community varchar(50),
  description varchar(200) DEFAULT 'RADIUS Client',
  PRIMARY KEY (id),
  KEY nasname (nasname)
);
```

1.11 Παραμετροποίηση DEFAULT του FreeRADIUS server

Παραμετροποιώντας το αρχείο */etc/freeradius/sites-enabled/default* δίνεται η δυνατότητα στον FreeRADIUS server να μπορεί να ξεκινήσει, να δημιουργήσει ένα default virtual host και να συνδεθεί στη MySQL βάση δεδομένων.

Με την εκτέλεση της παρακάτω εντολής:

sudo gedit /etc/freeradius/sites-enabled/default

ανοίγει το αρχείο default προς επεξεργασία και θα πρέπει να τροποποιηθούν οι αρχικές ρυθμίσεις όπου χρειάζεται, όπως παρουσιάζεται παρακάτω:

Στο σκέλος *authorize{}*

```
#
#  Look in an SQL database.  The schema of the database
#  is meant to mirror the "users" file.
#
#  See "Authorization Queries" in sql.conf
sql
```

(Γίνεται uncomment η ρύθμιση #sql → sql)

Στο σκέλος *accounting{}*

```
#
#  Log traffic to an SQL database.
#
#  See "Accounting queries" in sql.conf
sql
```

(Γίνεται uncomment η ρύθμιση #sql → sql)

Στο σκέλος session{}

```
session {
        radutmp
        #
        #  See "Simultaneous Use Checking Queries" in sql.conf
        sql
         }
```

(Γίνεται uncomment η ρύθμιση #sql → sql)

Στο σκέλος post-auth{}

```
#
#  After authenticating the user, do another SQL query.
#
#  See "Authentication Logging Queries" in sql.conf
sql
```

(Γίνεται uncomment η ρύθμιση #sql → sql)

1.12 Παραμετροποίηση RADIUSD.CONF του FreeRADIUS server

Το αρχείο RADIUS.CONF περιέχει είναι ο κεντρικός κορμός παραμετροποίησης του Radius Server. Περιλαμβάνει παραμέτρους ψευδοεντολών όπως και δείκτες και άλλα δυο αρχεία παραμετροποίησης όπου βρίσκονται στη μηχανή. Υπάρχουν επίσης γενικές επιλογές παραμετροποίησης για μεγάλο αριθμό modules, διαθέσιμες άμεσα και για μελλοντική χρήση για το FreeRADIUS. Τα modules μπορούν να αιτηθούν γενικές επιλογές και το FreeRADIUS θα περάσει αυτές τις ορισμένες επιλογές στο module μέσω του API του.

Το σύμβολο = ορίζει την τιμή ενός αντικειμένου. Το σύμβολο := ορίζει την τιμή ενός αντικειμένου και ξαναγράφει πάνω σε οποιαδήποτε τιμή που είχε οριστεί για ένα αντικείμενο. Το σύμβολο == συγκρίνει μια κατάσταση με μία ορισμένη τιμή. Είναι πολύ κρίσιμο ο χρήστης να καταλάβει πως αυτά τα σύμβολα λειτουργούν για να πετύχει την επιθυμητή παραμετροποίηση.

Ανοίγοντας το αρχείο προς επεξεργασία, θα πρέπει να τροποποιηθούν κάποιες από τι αρχικές ρυθμίσεις του αρχείου όπου χρειάζονται αλλαγές, όπως παρουσιάζεται παρακάτω.

Με την παρακάτω εντολή ανοίγει το αρχείο:

```
sudo gedit /etc/freeradius/radiusd.conf
```

Το “Authentication” του Radius server χρησιμοποιεί την port 1812

```
#  Port on which to listen.
#  Allowed values are:
#        integer port number (1812)
#        0 means "use /etc/services for the proper port"
port = 1812  ##Αλλαγή σε port 1812
```

Το “Accounting” του Radius server χρησιμοποιεί την port 1813

```
#  This second "listen" section is for listening on the accounting
#  port, too.
#
listen {
        ipaddr = *
#       ipv6addr = ::
        port = 1813  ##Αλλαγή σε port 1813
        type = acct
#       interface = eth0
#       clients = per_socket_clients
}
```

Στη συνέχεια γίνονται ρυθμίσεις για το logging , username, password κλπ

```
#  Log the full User-Name attribute, as it was found in the request.
#
# allowed values: {no, yes}
#
stripped_names = yes
#  Log authentication requests to the log file.
#
#  allowed values: {no, yes}
#
auth = yes
#  Log passwords with the authentication requests.
#  auth_badpass  - logs password if it's rejected
#  auth_goodpass - logs password if it's correct
#
#  allowed values: {no, yes}
#
auth_badpass = yes      ## Αλλαγή από no σε yes
```

```
auth_goodpass = no
```

Στο σκέλος `modules{}` πρέπει το αρχείο sql.conf να ενεργοποιηθεί για να μπορεί να διαβάζει από τη βάση δεδομένων

```
# Include another file that has the SQL-related configuration.
# This is another file only because it tends to be big.
#
$INCLUDE sql.conf  (Γίνεται uncomment η ρύθμιση)
```

1.13 Απενεργοποίηση ανάγνωσης από το CLIENTS.CONF του FreeRADIUS server

Επόμενο βήμα είναι η απενεργοποίηση της ανάγνωσης από το αρχείο clients.conf , διότι βάσει των αρχικών ρυθμίσεων όταν ο radius server ξεκινήσει, θα διαβάσει τους Radius Clients από το αρχείο clients.conf και λόγω των ανωτέρω αλλαγών, θα διαβάσει και από τη βάση δεδομένων. Για να αποφευχθεί η επιπλοκή αυτή μεταξύ των δύο, θα χρησιμοποιηθεί μόνο η ανάγνωση από τη βάση δεδομένων MySQL. Αυτό γίνεται με δύο τρόπους:

I. Με τη διαγραφή του αρχείου clients.conf
II. Με την προσθήκη comment στο αρχείο radius.conf στο *# $INCLUDE clients.conf*

Προτιμότερος είναι ο δεύτερος τρόπος και είναι αυτός που έχει χρησιμοποιηθεί.

1.14 Παραμετροποίηση αρχείου SQL.CONF του FreeRADIUS server

Το αρχείο SQL.CONF , περιέχει πληροφορίες σχετικά με τον τρόπο σύνδεσης στη βάση δεδομένων, τους πίνακες της βάσης δεδομένων που χρησιμοποιούνται στο FreeRADIUS server κλπ.

Το αρχείο παραμετροποιείται με την παρακάτω εντολή:

```
sudo gedit /etc/freeradius/sql.conf
```

Θα πρέπει να τροποποιηθούν οι αρχικές ρυθμίσεις όπου χρειάζεται, όπως παρουσιάζεται παρακάτω:

```
sql {
        #
        #  Set the database to one of:
        #
        #       mysql, mssql, oracle, postgresql
        #
        database = "mysql"
        #
        #  Which FreeRADIUS driver to use.
        #
        driver = "rlm_sql_${database}"

        # Connection info:
        server = "localhost"
        #port = 3306
```

```
login = "radius"          ## Αλλαγή του username σε radius
password = "setupRADIUS"  ## Αλλαγή password σε setupRADIUS
```

Στη συνέχεια γίνεται έλεγχος στις ρυθμίσεις του FreeRADIUS που επιτρέπουν την ανάγνωση, από τη βάση δεδομένων.

```
# Set to 'yes' to read radius clients from the database ('nas' table)
# Clients will ONLY be read on server startup.  For performance
# and security reasons, finding clients via SQL queries CANNOT
# be done "live" while the server is running.
#
readclients = yes    ##Γίνεται uncomment η default ρύθμιση
```

1.15 Παραμετροποίηση αρχείου INNER-TUNNEL.CONF του FreeRADIUS server

Στο αρχείο inner-tunnel.conf παραμετροποιείται ένας virtual server όπου διαχειρίζεται μόνο τις απαιτήσεις εσωτερικής σήραγγας για τους τύπους EAP-TTLS και PEAP. Η τροποποίηση θα γίνει μόνο στο σημείο που χρειάζεται να βλέπει την MySQL βάση για τη διαχείριση των χρηστών, καταργώντας έτσι το αρχείο users.conf.

Με την εκτέλεση της παρακάτω εντολής:

```
sudo gedit /etc/freeradius/sites-enabled/inner-tunnel
```

ανοίγει το αρχείο *inner-tunnel* προς επεξεργασία και θα πρέπει να τροποποιηθούν οι αρχικές ρυθμίσεις όπου χρειάζεται, όπως παρουσιάζεται παρακάτω:

```
#  Look in an SQL database.  The schema of the database
#  is meant to mirror the "users" file.
#  See "Authorization Queries" in sql.conf
sql     ## Γίνεται uncomment η ρύθμιση  #sql → sql
```

1.16 Παραμετροποίηση αρχείου EAP.CONF του FreeRADIUS server

Το αρχείο αυτό περιέχει όλους τους τύπους αυθεντικοποίησης, όπου μπορεί να υποστηρίξει το Extended Authentication Protocol. Αυτοί οι τύποι είναι MD5, LEAP, GTC (Generic Token Card), EAP-TLS (Transport Layer Security), TTLS, PEAP. Ο FreeRadius Server στα default σχόλια του, ενημερώνει το χρήστη ότι οι τύποι αυθεντικοποίησης MD5, LEAP, και GTC δεν προτείνονται για ασφαλή ασύρματη αυθεντικοποίηση, λόγω ευπάθειας των αλγορίθμων αυτών. Ο τύπος αυθεντικοποίησης όπου έχει οριστεί ως στάνταρτ είναι ο EAP-TLS όπου δεν χρησιμοποιεί καθόλου credentials και λειτουργεί μόνο με υποδομή ψηφιακών πιστοποιητικών PKI, όπου πάνω σε αυτόν βασίζονται οι περισσότερες ασφαλείς εγκαταστάσεις. Στο αρχείο eap.conf οι αρχικές ρυθμίσεις από την εγκατάσταση έχουν σημειωθεί ως σχόλια με # και η παραμετροποίηση γίνεται μόνο για την υπάρχουσα υποδομή.
Αρχικά γίνεται έλεγχος αν είναι ενεργοποιημένη η λειτουργία ανάγνωσης από το αρχείο EAP.CONF

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

```
#
$INCLUDE eap.conf
```

Η παραμετροποίηση του αρχείου eap.conf όπου χρησιμοποιείται με EAP-TLS στο server παρουσιάζεται παρακάτω. Σημειώνεται ότι οι τύποι αυθεντικοποίησης EAP-MD5,LEAP και GTC έχουν απενεργοποιηθεί. Παρακάτω παρουσιάζεται το τμήμα μόνο EAP-TLS του αρχείου που χρειάζεται να παραμετροποιηθεί.

```
eap {
# Καθορίζεται ο προκαθορισμένος τύπος EAP που θα χρησιμοποιηθεί
default_eap_type = tls

# Λίστα EAP-Response – EAP-Request πακέτων. Μετά το χρόνο 60secs η λίστα διαγράφεται
timer_expire     = 60

# Εάν ο server λάβει αίτηση για ένα EAP τύπο όπου δεν υποστηρίζει τότε απορρίπτει την
αίτηση.
ignore_unknown_eap_types = no

# Διορθώνει bug του AP Cisco AP1230B firmware
cisco_accounting_username_bug = yes
max_sessions = 4096

# Υποστηριζόμενοι EAP-τύποι αυθεντικοποίησης
tls      {
        certdir = ${confdir}/certs
        cadir = ${confdir}/certs

        # Το password του ιδιωτικού κλειδιού
        private_key_password = password

        # Path ιδιωτικού κλειδιού
        private_key_file = ${certdir}/server.key

        #  If Private key & Certificate are located in
        #  the same file, then private_key_file &
        #  certificate_file must contain the same file
        #  name.

        #Path του Dig. certificate
        certificate_file = ${certdir}/server.pem

        # Trusted Root CA list
        # Path του Dig. Certificate Root CA
        CA_file = ${cadir}/ca.pem

        # Path  Diffie-Helman αρχείου 1024 bits
        dh_file = ${certsdir}/dh

        # Path του random αρχείου
        random_file = /dev/urandom
```

```
        # Μέγεθος σε bits του πακέτου αποστολής
        fragment_size = 1024

        # Το παρακάτω flag ενεργοποιεί την αποστολή όλου του μήκους 1024 #bits σε κάθε
        αποστολή πακέτου, ενώ με "no" μόνο στο αρχικό πακέτο αποστολής.
        include_length = yes

        # Αν δηλωθεί τότε ελέγχει τη Certificate Revocation List όπου πρέπει να
        δημιουργηθεί με OpenSSL (Δεν έχει δημιουργηθεί, διότι ο έλεγχος πρόσβασης των
        χρηστών γίνεται μόνο από τη βάση δεδομένων)
        #  check_crl = yes
        CA_path = ${cadir}

        # Η παρακάτω παράμετρος ελέγχει  το Chain of Trust μέσω του πιστοποιητικού
        χρήστη σε σχέση με το  DN του εκδότη.Σε αντίθετη περίπτωση απορρίπτει το χρήστη.
        check_cert_issuer =
        "/C=GR/ST=ATTIKI/L=PIRAEUS/O=UNIPI/OU=TEMSEC/CN=ROOTCA"

        # Έλεγχος του πεδίου cn του ψηφιακού πιστοποιητικού ώστε να #ταυτίζεται με το
        user name της λίστας πρόσβασης. Αν δεν #πληρείται τότε η πρόσβαση αποκλείεται
        στο χρήστη.
        check_cert_cn = %{User-Name}

        # Καθορισμός των επιτρεπόμενων TLS  cipher suites
        cipher_list = "DEFAULT"

        # Η παρακάτω παράμετρος πρέπει να διαγραφεί όταν ο server εκτελεστεί σε
        κανονική λειτουργία (χρησιμοποιείται μόνο για αρχικές ρυθμίσεις)
        make_cert_command = "${certdir}/bootstrap"

        # Ενεργοποιώντας το caching, διατηρείται  στην cache για όσες ώρες οριστεί στο
        lifetime, το session Id της αρχικής αυθεντικοποίησης  μέσω SSL
        cache {
        enable = no
        lifetime = 24 # hours
        max_entries = 255
        }
    }
}
```

1.17 Παραμετροποίηση πινάκων της Βάσης Δεδομένων

Στο βήμα αυτό γίνονται όλες οι απαραίτητες παραμετροποιήσεις στους πίνακες της βάσης δεδομένων "radius" που δημιουργηθήκαν στην παρ. 1.10, ώστε να μπορεί ο FreeRADIUS server να λειτουργήσει βάσει των απαιτήσεων της εκάστοτε ανάγκης.

1.17.1 Πίνακας nas

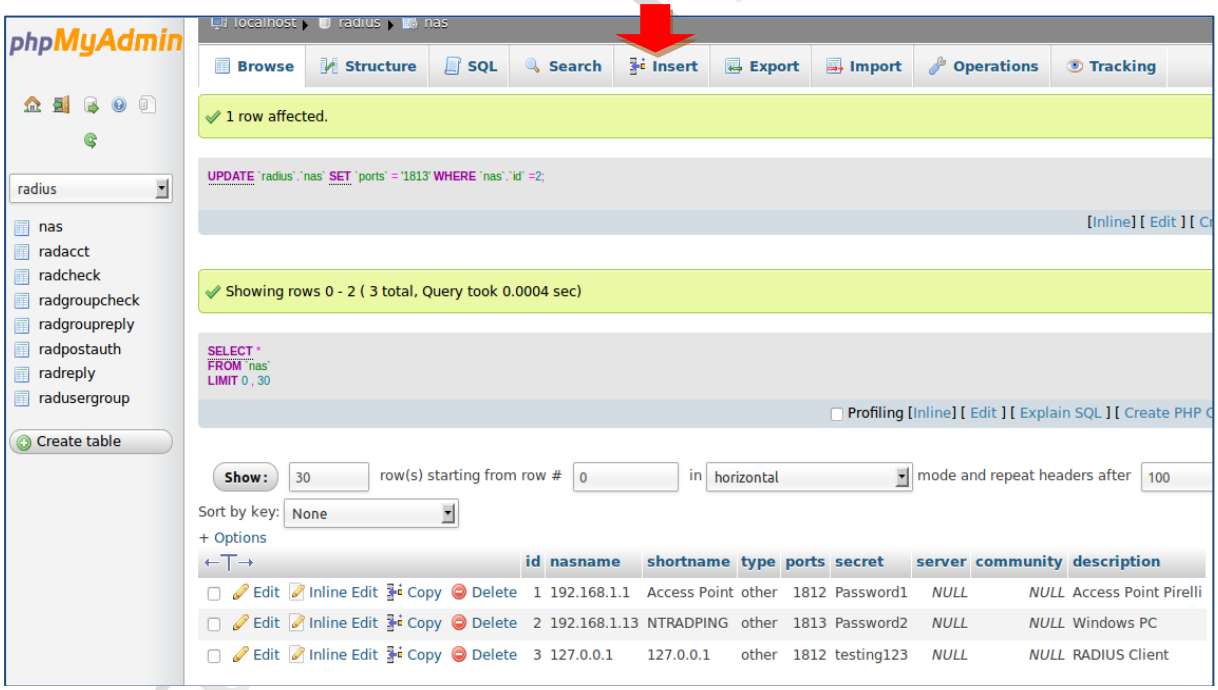
Στον πίνακα nas εισάγονται οι παράμετροι που χρειάζονται για να μπορεί να επικοινωνήσει ο Network Access Server του FreeRADIUS Server με το Access Point ή οποιαδήποτε άλλο τερματικό χρειάζεται να λειτουργήσει ως Authenticator μέσω πρωτοκόλλου 802.1X, όπου η μεταξύ τους επικοινωνία γίνεται σε φυσικό επίπεδο μέσω του πρωτοκόλλου 802.3.

Ο nas server έχει την έννοια του firewall όπου επιτρέπει από συγκεκριμένες IPs και ports να υπάρχει πρόσβαση στο radius server. Επιπρόσθετα, ορίζεται για κάθε διαφορετικό κανόνα που δημιουργείται ένα "Shared Secret" για την κρυπτογράφηση της επικοινωνίας μεταξύ του καναλιού, για την επίτευξη μιας ασφαλούς επικοινωνίας. Όταν δεν χρησιμοποιείται η επικοινωνία του SQL Server με τον Radius Server, οι ρυθμίσεις αυτές γίνονται στο αρχείο clients.conf.

Οι ρυθμίσεις που χρησιμοποιούνται στο nas server είναι οι παρακάτω:

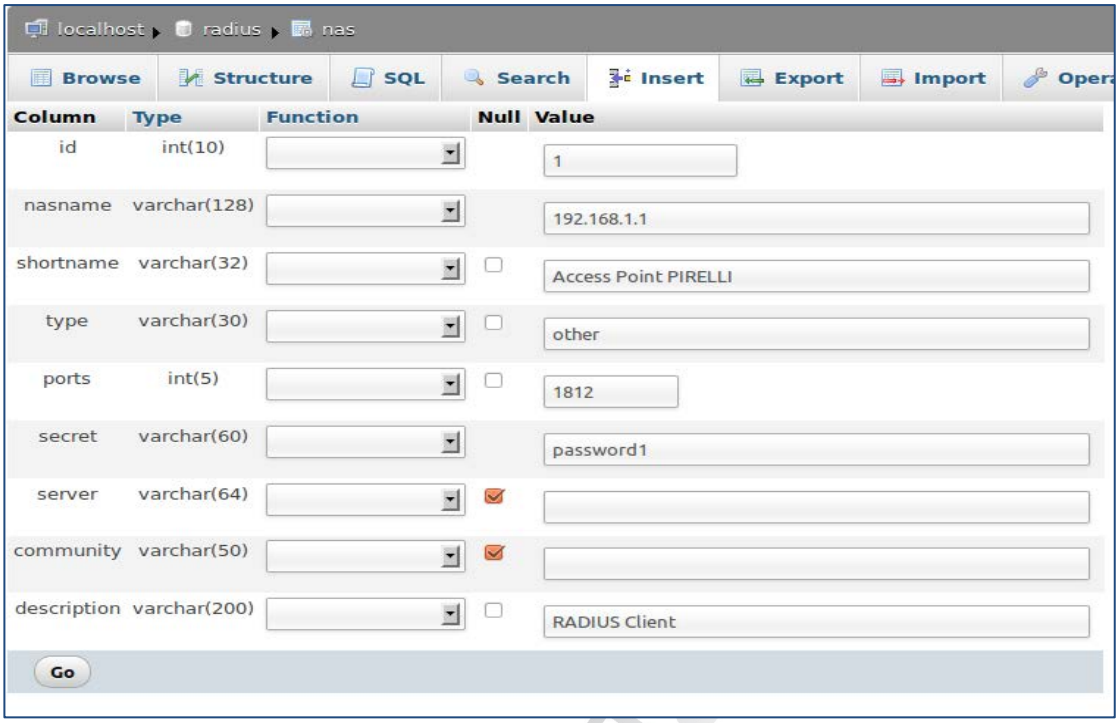
id : Άυξων Αριθμός
nasname: IP διεύθυνση ή έυρος IP διευθύνσεων (π.χ. 192.168.1.0/24)
shortname : Σύντομη ονομασία (για το logging)
type: Τύπος nas (cisco,Livingston,other,κλπ)
ports : Θύρα που χρησιμοποιείται (π.χ. 1812 -Authentication , 1813 -Accounting)
secret: Μυστικός κωδικός για την κρυπτογράφηση Radius-Authenticator
description: Περιγραφή

Για να γίνουν οι ρυθμίσεις αυτές πρέπει να επιλεχθεί ο πίνακας nas και στη συνέχεια να πιεστεί το κομβίο "Insert" (μέσω phpMyAdmin) όπως φαίνεται στην εικ. 1-5.



Εικόνα 1—5: Πίνακας nas

Τα στοιχεία που εισάγονται για το πρώτο id του Access Point παρουσιάζονται παρακάτω , όπως φαίνεται στην εικ. 1-6 και στη συνέχεια πιέζεται το κομβίο "Go" για καταχώρηση.



Εικόνα 1—6: Nas Insert

Η ίδια διαδικασία ακολουθείται και για τα επόμενα Ids που θα καταχωρηθούν. Στο πίν. 1-1 παρουσιάζονται οι ρυθμίσεις που έχουν οριστεί για το Access Point και για το id: 2, όπου χρησιμοποιείται για την επικοινωνία με την εφαρμογή NTRadPing πάνω σε Windows λειτουργικό σύστημα, για την υποστήριξη της λειτουργίας Accounting του FreeRadius server.

id	nasname	shortname	type	ports	secret	description
1	192.168.1.1	Access Point PIRELLI	other	1812	Password1	Access Point
2	192.168.1.13	NTRADPING	other	1813	Password2	Windows PC

Πίνακας 1—1: Παράμετροι Nas

1.17.2 Πίνακας radcheck

Ο πίνακας radcheck της βάσης radius περιέχει τους χρήστες που επιτρέπεται να αυθεντικοποιηθούν στον Radius Server με το αντίστοιχο password. Αντίστοιχα με την ίδια διαδικασία συμπληρώνονται τα παρακάτω δεδομένα (πιν. 1-2) στον πίνακα radcheck όπως έγινε με τον πίνακα nas.

id	username	attribute	op	value
1	user2	Cleartext-password	:=	
10	DEFAULT	Cleartext-password	:=	
3	pingpap	Cleartext-password	:=	1234
2	user1	Cleartext-password	:=	
4	pingchap	Cleartext-password	:=	1234

Πίνακας 1—2: Παράμετροι radcheck

1.17.3 Πίνακας radgroupcheck

Ο πίνακας radgroupcheck περιέχει τις ομάδες που ελέγχονται κατά την αυθεντικοποίηση στον Radius Server. Η ομάδα radping αντιστοιχεί σε αποδοχή πρόσβασης μέσω πρωτοκόλλου CHAP, η ομάδα radping2 αντιστοιχεί σε αποδοχή πρόσβασης μέσω πρωτοκόλλου PAP, η ομάδα users αντιστοιχεί σε αποδοχή πρόσβασης μέσω της τιμής EAP , μόνο εάν πληρούνται οι απαιτήσεις του αρχείου EAP.CONF και στην ομάδα DEFAULT δεν επιτρέπεται η πρόσβαση. Η ομάδα DEFAULT επιλέγεται όταν κανένας χρήστης δεν ανήκει σε καμία ομάδα.

id	groupname	attribute	op	value
2	radping	Auth-Type	:=	CHAP
1	users	Auth-Type	:=	EAP
4	DEFAULT	Auth-Type	:=	Reject
3	Radping2	Auth-Type	:=	PAP

Πίνακας 1—3: Παράμετροι radgroupcheck

1.17.4 Πίνακας radgroupreply

Στον πίνακα radgroupreply ορίζεται η απάντηση που επιστρέφει στον client η κάθε ομάδα, σε κάθε προσπάθειας αυθεντικοποίησης του χρήστη που περιέχει στον authenticator.

id	groupname	attribute	op	value
1	users	Framed-Protocol	:=	PPP
2	users	Service-Type	:=	Framed-User
3	users	Framed-Compression	:=	Van-Jacobsen-TCP-IP
4	radping	Framed-Protocol	:=	PPP
5	radping	Service-Type	:=	Framed-User
6	radping	Framed-Compression	:=	Van-Jacobsen-TCP-IP
7	radping2	Framed-Protocol	:=	PPP
8	radping2	Service-Type	:=	Framed-User
9	radping2	Framed-Compression	:=	Van-Jacobsen-TCP-IP

Πίνακας 1—4: Παράμετροι radgroupreply

1.17.5 Πίνακας radreply

Στον πίνακα radreply ορίζεται η απάντηση που επιστρέφει στον client ο κάθε χρήστης, σε κάθε προσπάθεια προς αυθεντικοποίηση του.

id	username	attribute	op	value
1	DEFAULT	Reply-Message	=	u are authorised
2	user1	Fall-Through	=	yes
3	User2	Fall-Through	=	yes
4	pingchap	Fall-Through	=	yes
5	pingpap	Fall-Through	=	yes

Πίνακας 1—5: Παράμετροι radreply

1.17.6 Πίνακας radusergroup

Στον πίνακα radusergroup ορίζεται η ομάδα που ανήκει ο κάθε χρήστης.

username	groupname	priority
DEFAULT	DEFAULT	1
user1	users	1
user2	users	1
pingchap	radping	1
pingpap	Radping2	1

Πίνακας 1—6: Παράμετροι radusergroup

1.17.7 Πίνακας radacct

Στον πίνακα radacct καταγράφονται αυτόματα πληροφορίες κάθε φορά που ξεκινάει ένα καινούριο session μέσω της πόρτας 1813, όπως username, η IP διεύθυνση του nas, χρόνος εκκίνησης και τερματισμού του session. Ο πίνακας αυτός χρησιμοποιείται μόνο για το accounting. (Δεν χρειάζεται παραμετροποίηση ο πίνακας)

radacctid	acctsessionid	acctuniqueid	Username	nasipaddress	acctstarttime	acctstoptime
24	6628	3df5dd5eb22acdc9	pingchap	192.168.1.13	2012-07-21 00:26:20	2012-07-21 01:57:14
38	6628	8039d7ada9eaaa50	pingap	192.168.1.13	2012-07-21 02:02:29	2012-07-21 02:02:32
36	6628	fd5b27fafa6a4d16	user2	192.168.1.13	2012-07-21 02:00:03	2012-07-21 02:00:10
37	6628	f2a36242d5680a5f	user1	192.168.1.13	2012-07-21 02:00:16	2012-07-21 02:00:20

Πίνακας 1—7: Παράμετροι radacct

1.17.8 Πίνακας radpostsuth

Στον πίνακα radpostauth καταγράφεται κάθε κίνηση προσπάθειας αυθεντικοποίησης. Κατά την παραμετροποίηση έχει οριστεί να καταγράφονται μόνο οι επιτυχείς αυθεντικοποιήσεις, για λόγους χωρητικότητας του logging. (Δεν χρειάζεται παραμετροποίηση ο πίνακας)

id	username	pass	reply	authdate
296	pingpap	1234	Access-Accept	2012-07-21 00:38:19
218	user1		Access-Accept	2012-07-08 22:34:57
219	User2		Access-Accept	2012-07-11 00:39:45
298	pingchap	0xe34f15d5227e19455e4a68f00cc9730a0f	Access-Accept	2012-07-21 01:32:36

Πίνακας 1—8: Παράμετροι radpostauth

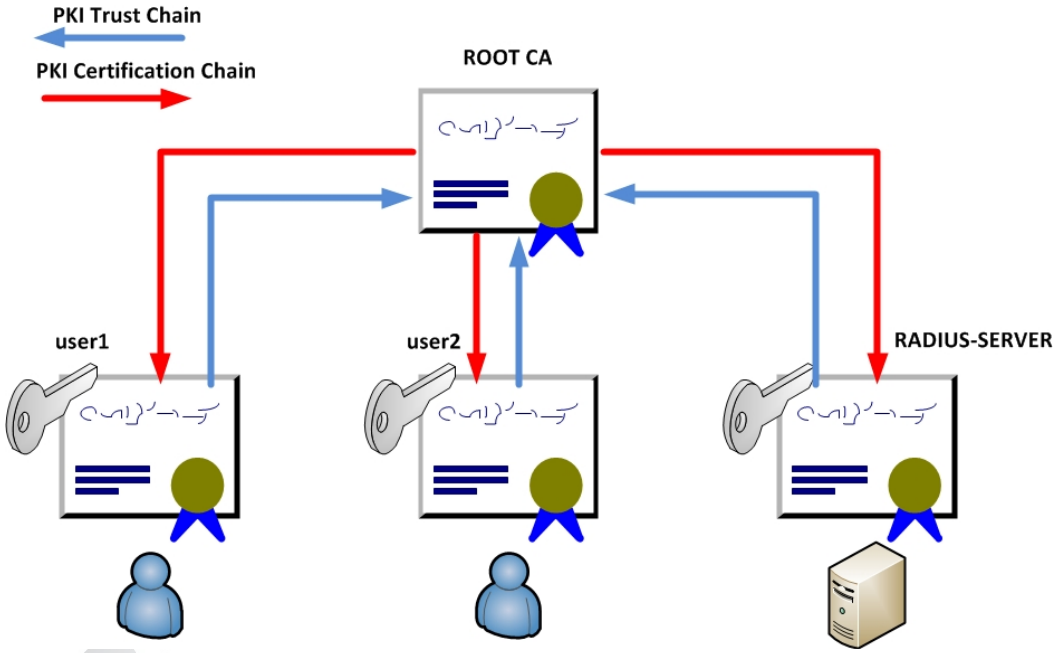
1.18 Δημιουργία Ψηφιακών Πιστοποιητικών

Χρησιμοποιώντας την εφαρμογή OPENSSL (έχει γίνει εγκατάσταση της εφαρμογής σε MS Windows περιβάλλον), δημιουργούνται τα ψηφιακά πιστοποιητικά όπου απαιτούνται για την αυθεντικοποίηση στον Radius Server μέσω του 802.1X EAP-TLS.

Αν τα ψηφιακά πιστοποιητικά των χρηστών πρόκειται να εγκατασταθούν και σε MS Windows περιβάλλον ,τότε πρέπει να δημιουργηθεί *το* αρχείο *xpextensions (*χωρίς κατάληξη αρχείο) στο path *C:\OpenSSL-Win32\bin* της εφαρμογής Openssl, για διόρθωση σφάλματος της εφαρμογής, εισάγοντας τα παρακάτω δεδομένα (μέσω notepad) :

```
[ xpclient_ext ]
extendedKeyUsage = 1.3.6.1.5.5.7.3.2
[ xpserver_ext ]
extendedKeyUsage = 1.3.6.1.5.5.7.3.1)
```

Στο Σχήμα 1-3 παρουσιάζεται η υλοποίηση PKI ,όπου εφαρμόζεται για την εμπιστοσύνη και πιστοποίηση χρηστών και Radius Server μέσω του ριζικού πιστοποιητικού ROOT CA για την επίτευξη του EAP-TLS.



Σχήμα 1-3: PKI υλοποίηση μεταξύ ROOT CA FreeRadius Server & χρηστών

1.18.1 Δημιουργία ROOT CA Certificate

1. Δημιουργία ιδιωτικού κλειδιού και δημιουργία request ψηφιακού πιστοποιητικού για το ROOT CA διάρκειας 3 ετών

   openssl req –new –x509 –keyout privatecakey.pem –out cacert.pem –days 1095 –config openssl.cfg

2. Εισαγωγή του request certificate υπογράφοντας το με το ιδιωτικό κλειδί και εξαγωγή του ιδιωτικού ψηφιακού πιστοποιητικού ROOT CA

openssl pkcs12 –export –in cacert.pem –inkey privatecakey.pem –out caroot.p12 –cacerts –descert

3. Μετατροπή του αρχείου από .p.12 σε .pem

openssl pkcs12 –in caroot.p12 –out caroot.pem

4. Εισαγωγή του request certificate και εξαγωγή του δημόσιου ROOT CA

openssl x509 –in cacert.pem –inform PEM –out cacert.der –outform DER

Μετατονομάζουμε το αρχείο cacert.pem σε ca.pem και το τοποθετούμε στο path */etc/freeradius/certs/*.

1.18.2 Δημιουργία του Radius server certificate

1. Δημιουργία ιδιωτικού κλειδιού RSA 1024 bit 1 έτους και request ιδιωτικού ψηφιακού πιστοποιητικού του Radius Server

openssl req –nodes –new –x509 -keyout radiusreq.pem –out radiusreq.pem –days 365 –config openssl.cfg

2. Δημιουργία προσωρινού ψηφ. πιστοποιητικού σε μορφή request ακόμη για αποστολή στον ROOT CA

openssl x509 –x509toreq –in radiusreq.pem –signkey radiusreq.pem –out radiustmp.pem

Επικύρωση του ιδιωτικού ψηφ. πιστοποιητικού του Radius Server με το ROOT CA

openssl ca –config openssl.cfg –policy policy_anything –out radiuscert.pem -extensions xpserver_ext –extfile xpxtensions –infiles radiustmp.pem

Μετατονομάζουμε το αρχείο radiusreq.pem σε server.key και το τοποθετούμε στο */etc/freeradius/certs/.* Στο ίδιο path τοποθετείται το radiuscert.pem που έχει δημιουργηθεί αφού μετονομασθεί σε server.pem.

1.18.3 Δημιουργία client certificate

Τα παρακάτω βήματα επαναλαμβάνονται για κάθε χρήστη που χρειάζεται να προστεθεί.

1. Δημιουργία ιδιωτικού κλειδιού RSA 1024 bit 1 έτους και αίτηση ιδιωτικού ψηφιακού πιστοποιητικού του client, για ταυτοποίηση του στον Radius Server

openssl req –nodes –new –x509 –keyout user1req.pem –out user1req.pem –days 365 –config openssl.cfg

2. Δημιουργία προσωρινού ψηφ. πιστοποιητικού σε μορφή request ακόμη για αποστολή στον ROOT CA

openssl x509 –x509toreq –in user1req.pem –signkey user1req.pem –out user1tmp.pem

3. Πιστοποίηση του requested ψηφ. πιστοποιητικού του client με το ROOT CA

   openssl ca –config openssl.cfg –policy policy_anything –out user1cert.pem -extensions xpclient_ext –extfile xpxtensions –infiles user1tmp.pem

4. Εξαγωγή του ιδιωτικού ψηφ. πιστοποιητικού του χρήστη για εισαγωγή του στο τερματικό του χρήστη

   openssl pkcs12 –export –in user1cert.pem –out user1cert.p12 –inkey user1req.pem –descert

5. Εξαγωγή του ιδιωτικού κλειδιού του χρήστη
   openssl pkcs12 –nodes –in user1cert.p12 –out user1key.pem

1.18.4 Δημιουργία Diffie-Helman 1024 bit παραμέτρων

Για τη δημιουργία του αρχείου dh.pem μήκους 1024-bit , πρέπει να δοθεί η παρακάτω εντολή:

openssl dhparam –out dh.pem 1024

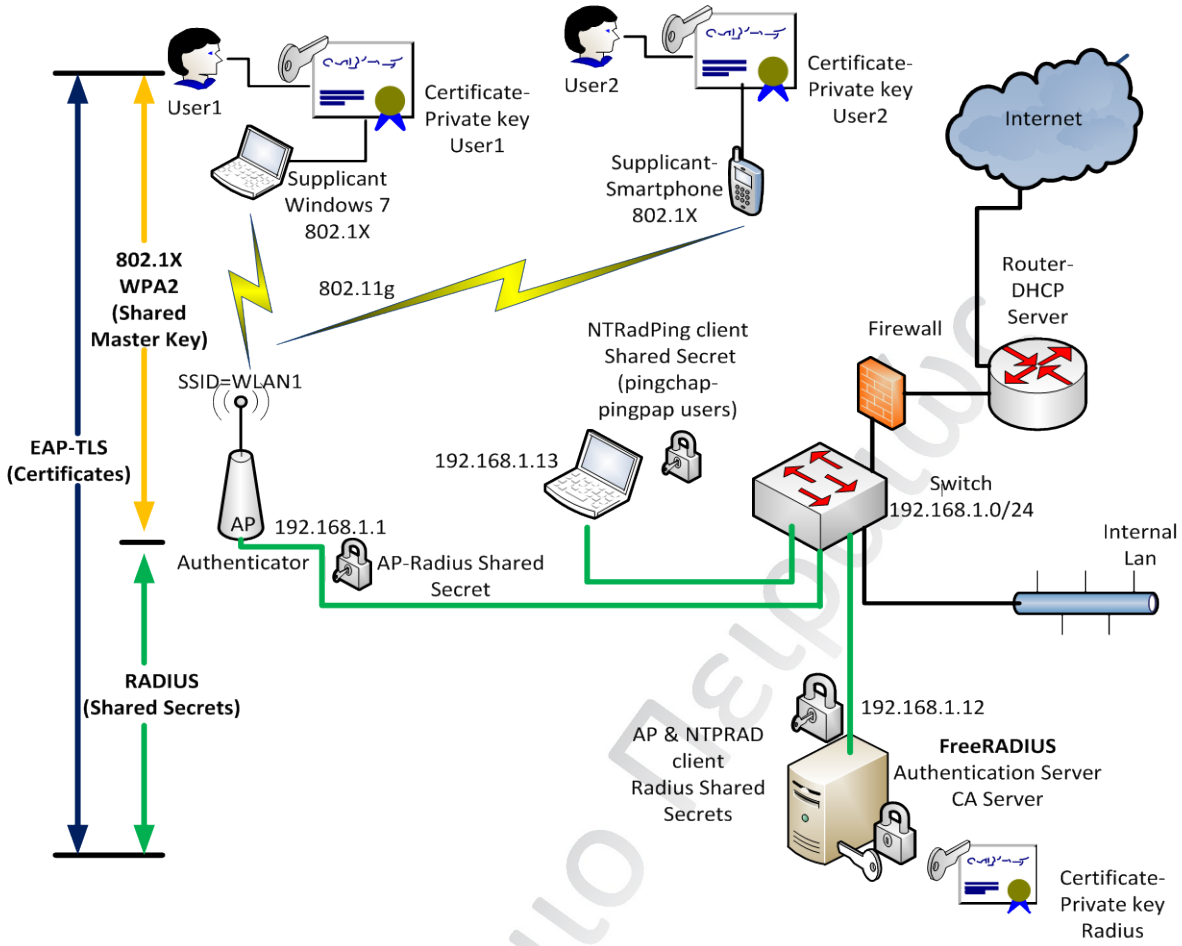
Το δημιουργούμενο αρχείο θα πρέπει να τοποθετηθεί στη διαδρομή /etc/freeradius/certs.

1.19 Εκκίνηση του Radius Server

Όταν όλες οι ρυθμίσεις έχουν ολοκληρωθεί, μπορεί πλέον ο FreeRADIUS server να ξεκινήσει και σε κανονική λειτουργία με την εντολή *sudo /etc/init.d/freeradius start και να σταματήσει με την εντολή sudo /etc/init.d/freeradius stop.* Είναι προτιμότερο όμως να εκτελείται σε debugging mode (με radiusd –X) για συνεχή δυνατότητα ανάγνωσης του log.

1.20 Υλοποίηση FreeRADIUS Server – Access Point με 802.1X EAP-TLS

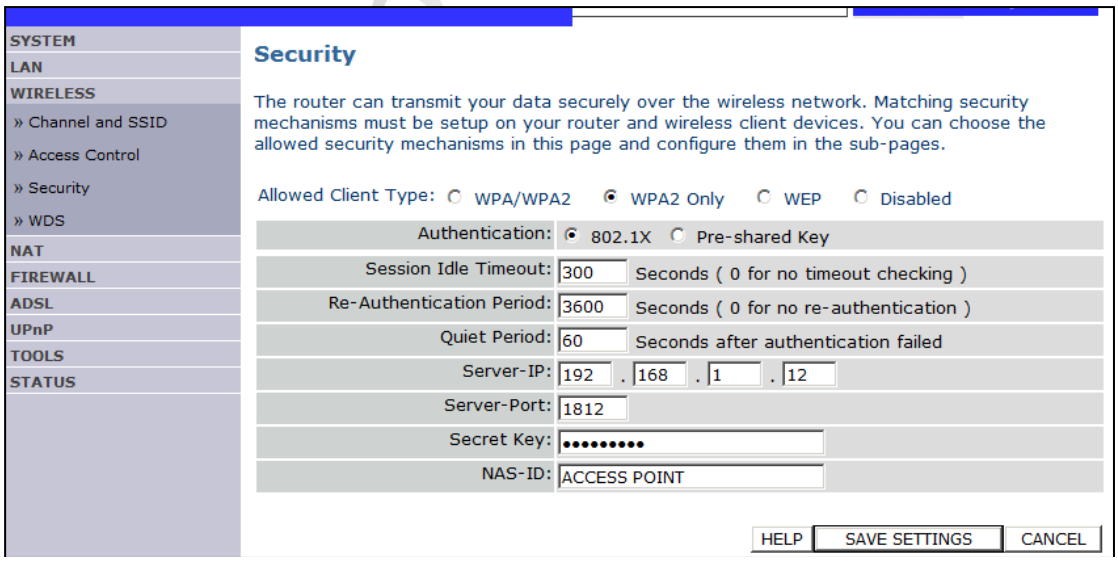
Ο FreeRadius Server επικοινωνεί με ένα Access Point που υποστηρίζει το πρωτόκολλο 802.1X μέσω LAN σύνδεσης και έχει γίνει η παρακάτω παραμετροποίηση στο AP, όπου χρησιμοποιείται Authentication μέσω 1812 port. Το Accounting είναι διαθέσιμο μέσω της 1813 port από τον server, αλλά δεν υποστηρίζεται από το Access Point, οπότε και χρησιμοποιείται από την εφαρμογή NTRADPING μέσω client Η/Υ. Η υλοποίηση της εφαρμογής έχει γίνει με προεπιλεγμένο το πρωτόκολλο επικοινωνίας 802.1X EAP-TLS στον Radius Server, για την ασφαλέστερη επικοινωνία μεταξύ Access Point και χρήστη. Στο σχήμα 1-4 παρουσιάζεται η σχηματική υλοποίηση της εφαρμογής.



Σχήμα 1-4: FreeRadius Server – Access Point με 802.1X EAP-TLS

1.20.1 Παραμετροποίηση Access Point

Η παραμετροποίηση των ρυθμίσεων του Access Point παρουσιάζεται στην εικ. 1-7.



Εικόνα 1—7: Παραμετροποίηση Access Point WPA2 802.1X Authentication

Security Mode: WPA2
Authentication: 802.1X
Server-IP: 192.168.1.12
Server Authentication Port: 1812
Secret Key: Password1
NasID: other
Session Idle Timeout : 300 sec
Re-Authentication Period: 3600 sec

1.21 Παραμετροποίηση Client με αυθεντικοποίηση EAP-TLS

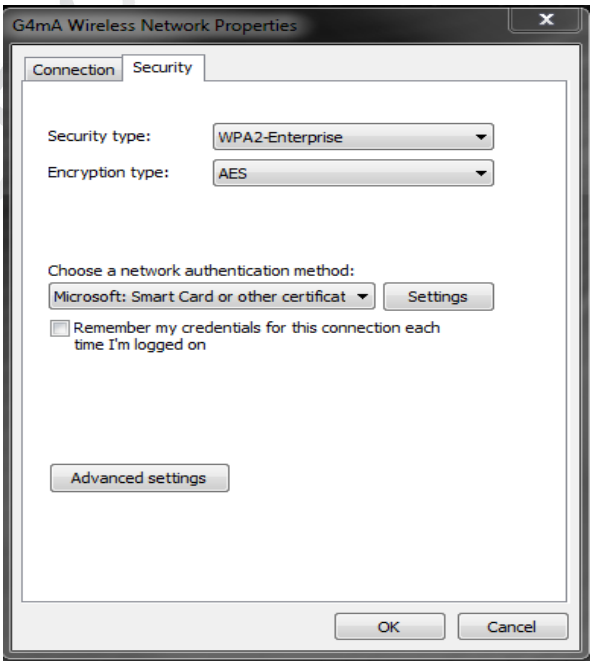
Η παραμετροποίηση του client μπορεί να γίνει σε οποιοδήποτε ΛΣ που υποστηρίζει ασύρματο δίκτυο με αυθεντικοποίηση EAP-TLS (μέσω ψηφιακών πιστοποιητικών). Η παραμετροποίηση έχει γίνει σε 2 Λειτουργικά Συστήματα, σε Microsoft Windows 7 και σε Microsoft Windows Mobile 6.5.

1.21.1 Παραμετροποίηση LAN interface σε MS Windows 7 client

Αρχικά ο κάθε client όπου πρέπει να κάνει log-in στο ασύρματο δίκτυο, πρέπει να εισάγει το ROOT CA *selfcacert.der* (μετονομασία σε *selfcacert.crt*) ως Trusted Root CA και επίσης να εισάγει το ιδιωτικό ψηφιακό πιστοποιητικό του κάθε χρήστη *User1cert.pfx ή User2cert.pfx* (Κατά την εισαγωγή του για επιλογή στη λίστα του Lan interface, δεν πρέπει να επιλεχθεί το "Enable strong private key protection").

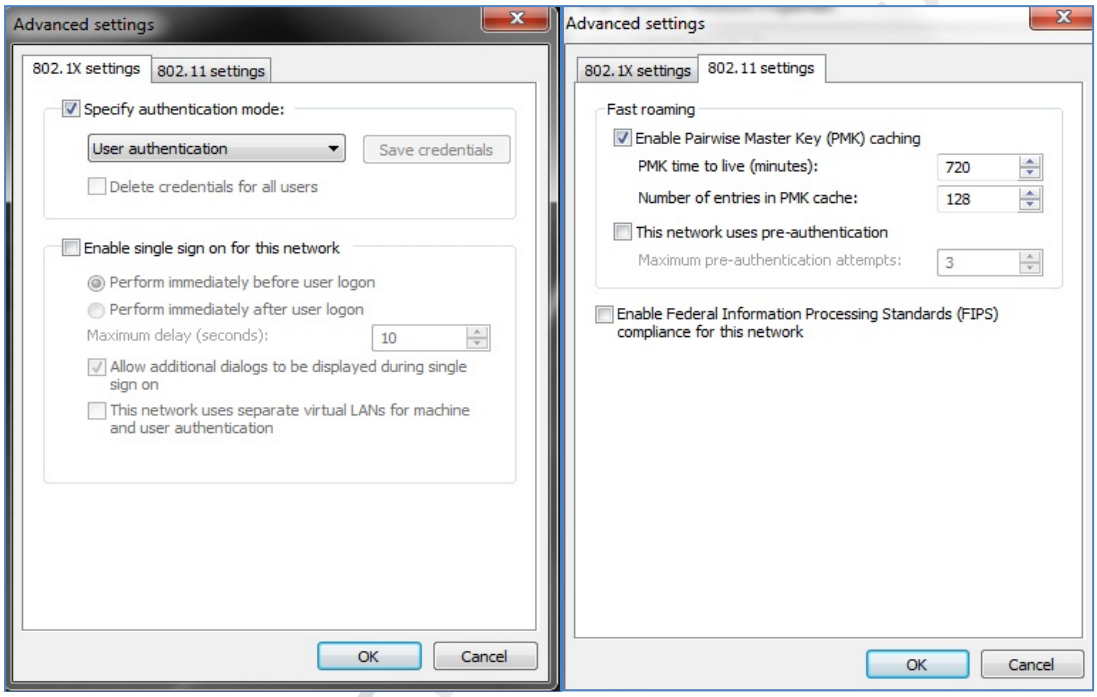
Το LAN interface σε Windows 7 περιβάλλον, πρέπει αρχικά να παραμετροποιηθεί όπως παρουσιάζεται στην εικ. 1-8 επιλέγοντας:

Security type: WPA2- Enterprise
Encryption type: AES
Authentication method: Smart Card or other certificate



Εικόνα 1—8: Wireless Network properties (Security)

Στη συνέχεια επιλέγοντας “Advanced Settings” στο πλαίσιο 802.1X settings επιλέγεται “specify authentication mode” και από τη λίστα επιλέγεται “User Authentication” (εικ. 1-9). Λόγω χρησιμοποίησης 802.11i (WPA2) δίνεται η δυνατότητα να χρησιμοποιηθεί ένα επιπλέον πλαίσιο το 802.11 settings , όπου ενεργοποιώντας το Pairwise master Key (PMK) caching,ορίζεται ο χρόνος TTL 720 λεπτά, όπου μπορεί να “ζήσει” το κλειδί σε ένα session.Επίσης ορίζεται ο αριθμός καταχωρήσεων των εισόδων όπου θα αποθηκεύει στην cache μνήμη του PMK, όπου είναι ίσος με 128. Βέβαια ο default χρόνος TTL όπου έχει οριστεί για το PMK δεν ανταποκρίνεται λόγω του έχει οριστεί στο AP Reauthentication period=3600 secs.

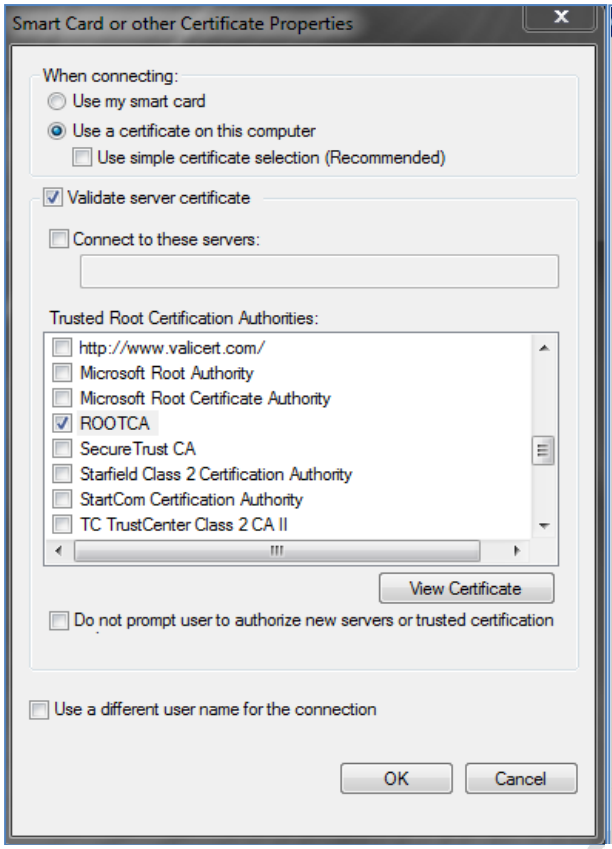


Εικόνα 1—9: Advanced 802.1X - 802.11 Settings

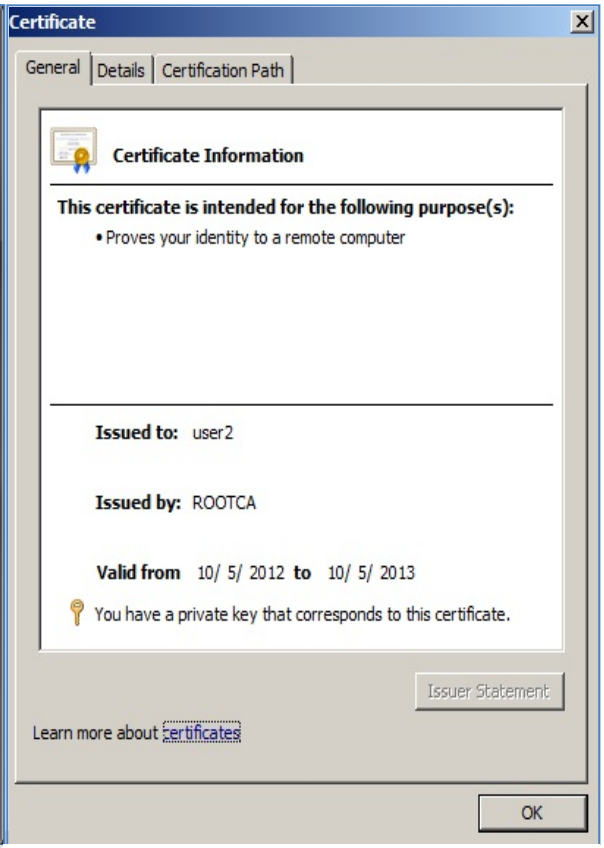
Στη συνέχεια όπως φαίνεται στην εικ. 1-10 επιλέγοντας settings αν υπάρχει smart card με ψηφιακό πιστοποιητικό εγκατεστημένο επιλέγεται αντίστοιχα, αλλιώς αφού έχουν εισαχθεί τα ψηφιακά πιστοποιητικά, επιλέγεται το “Use a certificate on this computer” και επίσης επιλέγεται “Validate server certificate” με το ROOTCA authority όπου εισήχθηκε πιο πριν.

Έχοντας επιλέξει σύνδεση με το ασύρματο δίκτυο, παρουσιάζεται στην εικ. 1-12 η επιλογή ψηφ. πιστοποιητικού του χρήστη, για αυθεντικοποίηση στον Radius Server. Επιλέγοντας “View Certificate” (εικ. 1-11) παρουσιάζονται οι ιδιότητες του ψηφιακού πιστοποιητικού του κάθε χρήστη.

Σε περίπτωση χρησιμοποίησης του client σε περιβάλλον linux, ο χρήστης θα χρειαστεί να εισάγει τα παρακάτω αρχεία: selfcacert.pem, user1cert.pem, user1req.pem.

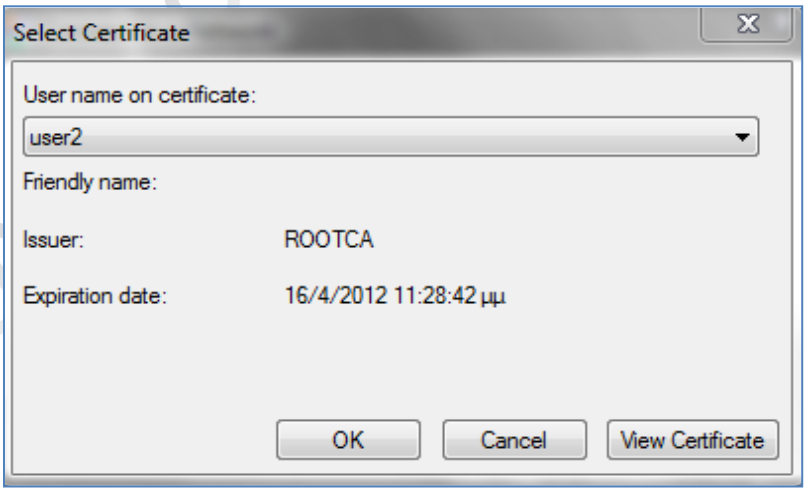


Εικόνα 1—11: Certificate Properties



Εικόνα 1—10: Ιδιωτικό Ψηφιακό Πιστοποιητικό του user2

Επιλέγοντας το κατάλληλο ψηφιακό πιστοποιητικό χρήστη μετά από επιβεβαίωση και αφού έχει ολοκληρωθεί η επιτυχής διαδικασία handshaking του ψηφιακού πιστοποιητικού στον radius server,ο χρήστης user2 εισέρχεται επιτυχώς στο δίκτυο, όπως φαίνεται στην εικόνα 1-13 και εικόνα 1-14.



Εικόνα 1—12: Επιλογή certificate για αυθεντικοποίηση στον Radius Server

Εικόνα 1—13: Επιτυχής σύνδεση του client στο Access Point

```
rlm_sql (sql): Released sql socket id: 4
++[sql] returns ok
++[exec] returns noop
Sending Access-Accept of id 6 to 192.168.1.1 port 32792
        Framed-Protocol := PPP
        Service-Type := Framed-User
        Framed-Compression := Van-Jacobson-TCP-IP
        MS-MPPE-Recv-Key = 0x771b9adc97fe63d670f5849006b262ab4ff523b51b5494292c256fe29c98fbc2
        MS-MPPE-Send-Key = 0x9736b3593b8e6e82825aaea37c1851f20ead259091694acace653ae5b6606eb6
        EAP-Message = 0x03060004
        Message-Authenticator = 0x00000000000000000000000000000000
        User-Name = "user2"
Finished request 169.
Going to the next request
Waking up in 4.8 seconds.
```

Εικόνα 1—14: Απάντηση αποδοχής πρόσβασης του RADIUS server στο Access Point για το χρήστη

1.21.2 Radius Server Logging

Στο σημείο αυτό παρουσιάζεται το σημαντικότερο μέρος της συνολικής καταγραφής ενός κύκλου αυθεντικοποίησης ενός χρήστη στον RADIUS server με τη λειτουργία καταγραφής του FreeRADIUS server (Freeradius –X).

```
rad_recv: Access-Request packet from host 192.168.1.1 port 32783, id=2, length=144
        User-Name = "user2"
        NAS-IP-Address = 0.0.0.0
        Called-Station-Id = "00-1C-A2-AB-68-61:G4mA"
        Calling-Station-Id = "00-23-76-C6-E1-BA"
        NAS-Identifier = "ACCESS POINT"
        NAS-Port = 29
        Service-Type = Framed-User
        Framed-MTU = 1400
        NAS-Port-Type = Wireless-802.11
        EAP-Message = 0x0202000a017573657232
        Message-Authenticator = 0x37d7d005eadb65691d6d42be5553d317
.
.
 [eap] EAP packet type response id 2 length 10
```

```
.
[sql]    expand: %{User-Name} -> user2
[sql] sql_set_user escaped user --> 'user2'
.
 [sql]    expand: SELECT id, username, attribute, value, op        FROM radcheck        WHERE username =
'%{SQL-User-Name}'        ORDER BY id -> SELECT id, username, attribute, value, op        FROM radcheck
WHERE username = 'user2'        ORDER BY id
[sql] User found in radcheck table

[sql]    expand: SELECT id, username, attribute, value, op        FROM radreply        WHERE username =
'%{SQL-User-Name}'        ORDER BY id -> SELECT id, username, attribute, value, op        FROM radreply
WHERE username = 'user2'        ORDER BY id

[sql]    expand: SELECT groupname        FROM radusergroup        WHERE username = '%{SQL-User-Name}'
ORDER BY priority -> SELECT groupname        FROM radusergroup        WHERE username = 'user2'
ORDER BY priority

[sql]    expand: SELECT id, groupname, attribute,        Value, op        FROM radgroupcheck        WHERE
groupname = '%{Sql-Group}'        ORDER BY id -> SELECT id, groupname, attribute,        Value, op        FROM
radgroupcheck        WHERE groupname = 'users'        ORDER BY id
[sql] User found in group users
.
.
Found Auth-Type = EAP
 [eap] EAP Identity
[eap] processing type tls
[tls] Requiring client certificate
[tls] Initiate
[tls] Start returned 1
++[eap] returns handled
rad_recv: Access-Request packet from host 192.168.1.1 port 32784, id=3, length=230

        User-Name = "user2"
        NAS-IP-Address = 0.0.0.0
        Called-Station-Id = "00-1C-A2-AB-68-61:G4mA"
        Calling-Station-Id = "00-23-76-C6-E1-BA"
        NAS-Identifier = "ACCESS POINT"
        NAS-Port = 29
        Service-Type = Framed-User
        Framed-MTU = 1400
        NAS-Port-Type = Wireless-802.11
        State = 0xe627eb9be624e699f3e11eec6b39c9f0
        EAP-Message =
0x0203004e0d800000004416030100 3f0100003b03017d5e6312692c663f036c82a7a7738263fa815eed2c7fc98c
2a73b784a9161d81000014002f003500040005000a000900640062000300060100
        Message-Authenticator = 0xe4525cc85c0b455a4b049dbc623ca6a6

Sending Access-Challenge of id 4 to 192.168.1.1 port 32785
        Framed-Protocol := PPP
        Service-Type := Framed-User
        Framed-Compression := Van-Jacobson-TCP-IP
        EAP-Message =
0x010501c10d80000005ad42dd3405e7e69dccefdce415fbc23028178c47810f2738f07d1b3eb1bc56292e2acf877
b2696d1858924e4b6e26c59bdc93701abff08d4e1dfda9c594846b6f5b2e01fa6e314392893a3f6e81457c4fdafd2
5af90203010001a350304e301d0603551d0e0416041411ec5d729e59ccca7ca23911936541c80ae38621301f060
3551d2304183016801411ec5d729e59ccca7ca23911936541c80ae38621300c0603551d13040530030101ff300d
```

06092a864886f70d010105050003818100960456 22ddc036339bfdf9a33f1db7506f41980423734a68254761908 ddb6a2ce85ef590c567f1eb5f970f3be4a8c6ed4eca9aa2cd

EAP-Message = 0xc616911d073549201dc28e3df121355ec907036206cfab171b81708ed3d5a8100c0332f920215ef542bfa8a82a b56622a7f68946c806f516d7f4130afdf34f901f55c62d733bee0d044f9316030100740d00006c030102400066006 43062310b30090603550406 13024752310f300d06035504080c06415454494b493110300e06035504070c07504 95241455553310e300c060355040a0c05554e495049310f300d060355040b0c0654454d534543310f300d06035 504030c06524f4f5443410e000000

Message-Authenticator = 0x00000000000000000000000000000000

State = 0xe627eb9be422e699f3e11eec6b39c9f0

Finished request 61.

Going to the next request

Waking up in 4.9 seconds.

rad_recv: Access-Request packet from host 192.168.1.1 port 32786, id=5, length=1026

User-Name = "user2"

NAS-IP-Address = 0.0.0.0

State = 0xe627eb9be422e699f3e11eec6b39c9f0

EAP-Message = 0x020503b50d80000003ab160301036b0b00025b0002580002553082025 1308201baa003020102020201203 00d06092a864886f70d0101050003062310b30090603550406 13024752310f300d06035504080c06415454494b4 93110300e06035504070c075049524 1455553310e300c060355040a0c05554e495049310f300d060355040b0c0 654454d534543310f300d06035504030c06524f4f544341301e170d313230353130313 03131303535335a170d31333 035313031313035353 35a3061310b30090603550406 13024752310f300d06035504080c06415454494b4931103 00e06035504070c075049524 1455553310e300c060355040a0c05554e49

EAP-Message = 0x5049310f300d060355040b0c0654454d534543310e300c06035504030c05757365723230819f300d06092a86 4886f70d010101050003818d003081902818100bcc03f025f4c8306153e4645673d5359e543bf686313fe82d8df 7a1cb5c01cc193d6ff10f17afb1f344f384e6890bebb5d38ebe2c0b046fd45fc48f4d502d73f9c9af514c00d90a8b48 0ae226f7e17caaee07ede0ae936d20fe015e35740cf5d1f83dc3ef2f5d0d847c3fd57da7ee4e44b37f2dc953310015 aecb5db297a73b70203010001a3173015301306035 51d25040c300a06082b060105050703023 00d06092a8648 86f70d010105050003818100080159277 1baa3aaeed43d2f

EAP-Message = 0x1ad8cb3ec32848935dafb71b38bfc5af29ddeeb1d43ba5549e276d3a95d087ff35f84eae6e56a47b706fced8eb7 12416f5ad9037467ea007f181c0f332d87efd7e5b1df7ed928e7142b7cc68a930960266512daea25fc2b37e38232 8e029981537b5426e3df58adb790a7c1bca022c1206491e1110000082008 0b8a4ff32d98e4fe6c9dd77de76f6266 f1427de64a3217ba5466ebe0cf99f2b26e0a98dac63c6743642c9476087a5dfb384c5a3b4f995cbcd30af04773d32 077caad3c0ad89ee66af4e4df60214cd6e8dbe06dcf6890efb8c59dc50afcd146ab0a4cea7836412ebf5740e2e675 357273562240aab95fb58eb32b6061c870db20e0f0000

EAP-Message = 0x8200804f39341659c5512aea413b530d4e9721cee2ff6be40a8783fa4a94a0ce435ec2a72eecae0c12fb10f0bcf7 18b93a5fdbd00b0326bcab6c31a90bf861f69a06232784774319c8195968f4ef9146a75fcb0c0b5f93b73f99cc0ca 147e975daa0684048678212d23192c9b2089689f1479657be69f186ef2e71755a3d54f6888f3414030100010116 03010030056a4f8c1d7e4ab5f34a3181b5fb4fb0e6b07835f0c52c5140e0169e2be7ea1bb2ab49c6bcddd30e9f85 03f12805dfac

Message-Authenticator = 0x87bd5a67b6fe32c9ff5e85cb45a29ead

.

.

Found Auth-Type = EAP

 # Executing group from file /etc/freeradius/sites-enabled/default

+- entering group authenticate {...}

[eap] Request found, released from the list

[eap] EAP/tls

[eap] processing type tls

[tls] Authenticate

[tls] processing EAP-TLS

TLS Length 939

[tls] Length Included

```
[tls] eaptls_verify returned 11
[tls] <<< TLS 1.0 Handshake [length 025f], Certificate
[tls] chain-depth=1,
[tls] error=0
[tls] --> User-Name = user2
[tls] --> BUF-Name = ROOTCA
[tls] --> subject = /C=GR/ST=ATTIKI/L=PIRAEUS/O=UNIPI/OU=TEMSEC/CN=ROOTCA
[tls] --> issuer  = /C=GR/ST=ATTIKI/L=PIRAEUS/O=UNIPI/OU=TEMSEC/CN=ROOTCA
[tls] --> verify return:1
[tls]     expand: %{User-Name} -> user2
[tls] checking certificate CN (user2) with xlat'ed value (user2)
[tls] chain-depth=0,
[tls] error=0
[tls] --> User-Name = user2
[tls] --> BUF-Name = user2
[tls] --> subject = /C=GR/ST=ATTIKI/L=PIRAEUS/O=UNIPI/OU=TEMSEC/CN=user2
[tls] --> issuer  = /C=GR/ST=ATTIKI/L=PIRAEUS/O=UNIPI/OU=TEMSEC/CN=ROOTCA
[tls] --> verify return:1
[tls]   TLS_accept: SSLv3 read client certificate A
[tls] <<< TLS 1.0 Handshake [length 0086], ClientKeyExchange
[tls]   TLS_accept: SSLv3 read client key exchange A
[tls] <<< TLS 1.0 Handshake [length 0086], CertificateVerify
[tls]   TLS_accept: SSLv3 read certificate verify A
[tls] <<< TLS 1.0 ChangeCipherSpec [length 0001]
[tls] <<< TLS 1.0 Handshake [length 0010], Finished
[tls]   TLS_accept: SSLv3 read finished A
[tls] >>> TLS 1.0 ChangeCipherSpec [length 0001]
[tls]   TLS_accept: SSLv3 write change cipher spec A
[tls] >>> TLS 1.0 Handshake [length 0010], Finished
[tls]   TLS_accept: SSLv3 write finished A
[tls]   TLS_accept: SSLv3 flush data
[tls]   (other): SSL negotiation finished successfully
SSL Connection Established
[tls] eaptls_process returned 13
++[eap] returns handled
Sending Access-Challenge of id 5 to 192.168.1.1 port 32786
        Framed-Protocol := PPP
        Service-Type := Framed-User
        Framed-Compression := Van-Jacobson-TCP-IP
        EAP-Message =
0x010600450d800000003b1403010001011603010030eac8b129a6b3de7245ca9a83cd4c78cceedbd0e3724b69
7bb786d33f29515bed90bceaa59b8f8a81885f46d29fcc03f5
        Message-Authenticator = 0x00000000000000000000000000000000
        State = 0xe627eb9be521e699f3e11eec6b39c9f0
Finished request 62.
Going to the next request
Waking up in 4.8 seconds.

rad_recv: Access-Request packet from host 192.168.1.1 port 32787, id=6, length=158
        User-Name = "user2"
        NAS-IP-Address = 0.0.0.0
        Called-Station-Id = "00-1C-A2-AB-68-61:G4mA"
        Calling-Station-Id = "00-23-76-C6-E1-BA"
        NAS-Identifier = "ACCESS POINT"
        NAS-Port = 29
        Service-Type = Framed-User
        Framed-MTU = 1400
```

```
        NAS-Port-Type = Wireless-802.11
        State = 0xe627eb9be521e699f3e11eec6b39c9f0
        EAP-Message = 0x020600060d00
        Message-Authenticator = 0x2f04fe674233b67e1f8e1dba0fec296c
.
.
Found Auth-Type = EAP
# Executing group from file /etc/freeradius/sites-enabled/default
+- entering group authenticate {...}
[eap] Request found, released from the list
[eap] EAP/tls
[eap] processing type tls
[tls] Authenticate
[tls] processing EAP-TLS
[tls] Received TLS ACK
[tls] ACK handshake is finished
[tls] eaptls_verify returned 3
[tls] eaptls_process returned 3
[tls] Adding user data to cached session
[eap] Freeing handler
++[eap] returns ok
Login OK: [user2] (from client Access Point port 29 cli 00-23-76-C6-E1-BA)
.
.
Sending Access-Accept of id 6 to 192.168.1.1 port 32787
        Framed-Protocol := PPP
        Service-Type := Framed-User
        Framed-Compression := Van-Jacobson-TCP-IP
        MS-MPPE-Recv-Key = 0x0ac5040f79708aa4265ccd011c833a8470ffd1eeab85df437808062d86ad695e
        MS-MPPE-Send-Key = 0x78172ee33714b1b659b56b26771d521f3c0c78c0e6baa29e1ed6e18744ec0a8a
        EAP-Message = 0x03060004
        Message-Authenticator = 0x00000000000000000000000000000000
        User-Name = "user2"
Finished request 63.
Going to the next request
Waking up in 4.8 seconds.
Cleaning up request 59 ID 2 with timestamp +614591
Ready to process requests.
```

Στην παραπάνω καταγραφή παρατηρείται ότι γίνεται έλεγχος του χρήστη και του group που ανήκει ο χρήστης στη MySQL βάση δεδομένων, όπως επίσης και του Access Point. Γίνεται αυθεντικοποίηση μέσω πρωτοκόλλου EAP TLS και γίνεται έλεγχος του πεδίου cn(common name) του ψηφιακού πιστοποιητικού ώστε να ταυτίζεται με το user name της λίστας πρόσβασης, όπως επίσης και έλεγχος του Chain of Trust μέσω του πιστοποιητικού χρήστη σε σχέση με το DN(Domain Name) του εκδότη.

1.22 Παραμετροποίηση NTRadPing Test Utility

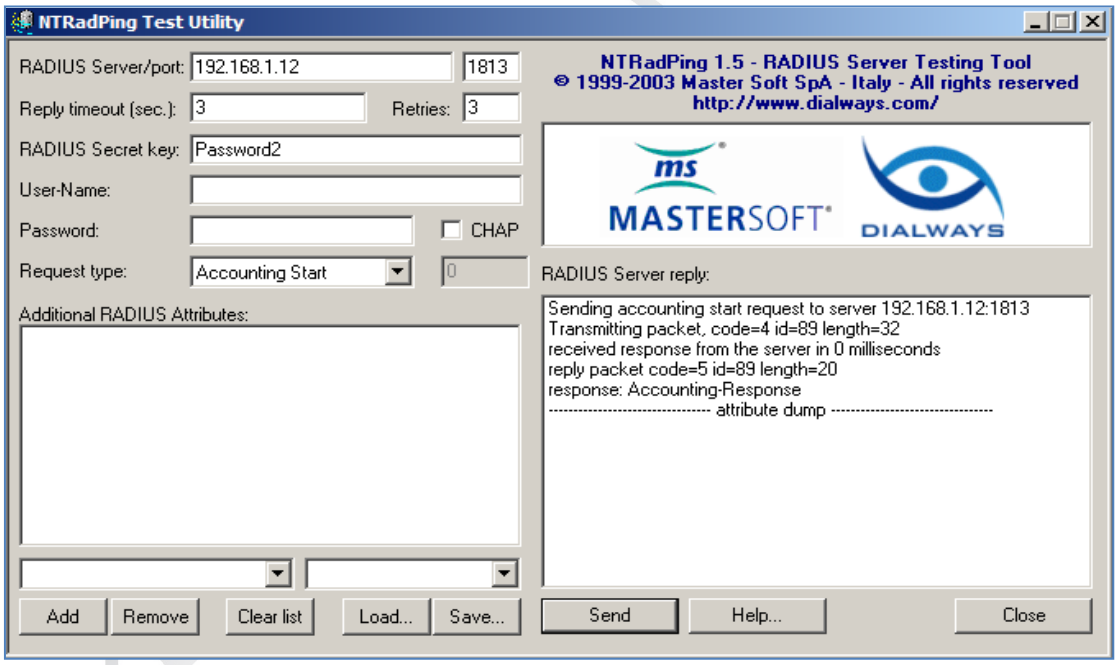
Το NTRadPing Test Utility είναι μια εφαρμογή για Windows Λ.Σ. όπου μπορεί με κατάλληλη παραμετροποίηση να δοκιμάσει τη λειτουργία του FreeRadius server για Authentication, Authorazition & Accounting.

1.22.1 Παραμετροποίηση NTRadPing Test για Accounting

Στην εφαρμογή όπου περιγράφηκε παραπάνω, λόγω μη διαθεσιμότητας του Access Point να χρησιμοποιήσει Accounting στον Freeradius Server, η εφαρμογή γίνεται συνδέοντας ένα client με εγκατεστημένη την εφαρμογή NTRadPing στο δίκτυο, ώστε να χρησιμοποιεί την 1813 port και να γίνεται η καταγραφή στον Server βάσει της παραμετροποίησης όπου φαίνονται στην εικ. 1-15. Αν τα στοιχεία είναι σωστά σύμφωνα με τα δεδομένα του πίνακα της βάσης δεδομένων, τότε η εφαρμογή επιστρέφει "response: Accounting-Response". Σε οποιαδήποτε άλλη περίπτωση θα επιστρέψει "No response from Server" και σε περίπτωση μη αυθεντικοποίησης, επιστρέφει "response: Access-Rejected".

Radius Server/port :192.168.1.12/1813
Reply timeout (sec) :3
Retries: 3
Radius Secret Key: Password2
Request type: Accounting (Start, Stop, Interim-Update, On ,Off)

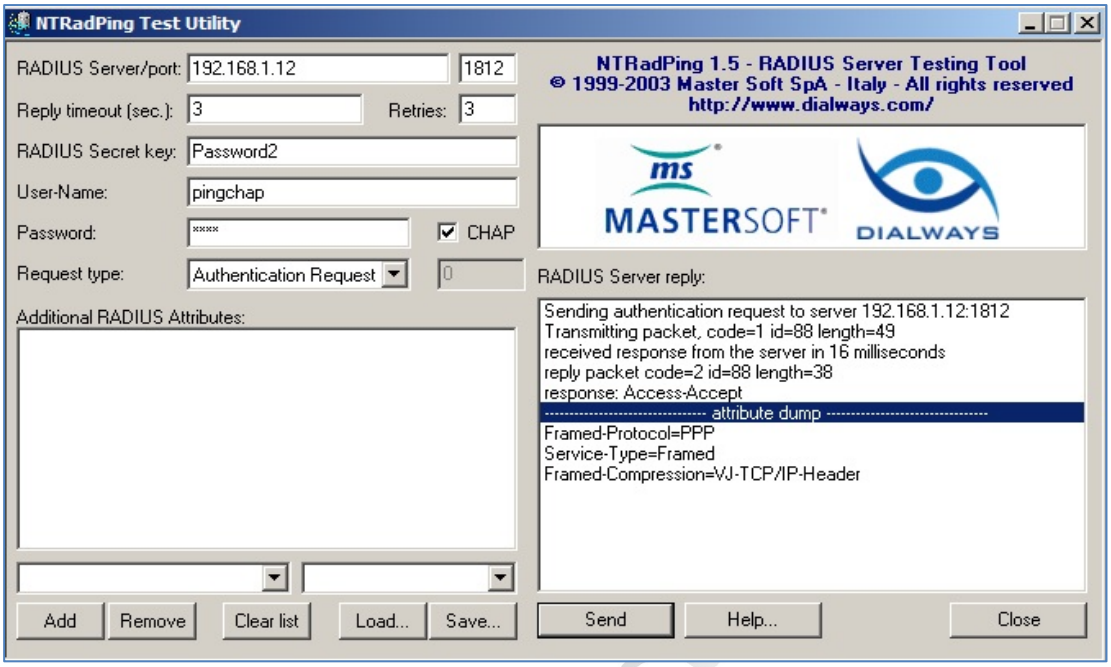
Στην εικόνα 1-15 παρουσιάζονται οι ρυθμίσεις της εφαρμογής NTRadPing Test Utility:



Εικόνα 1—15: NTRadPing Test Utility Accounting

1.22.2 Παραμετροποίηση NTRadPing Test με αυθεντικοποίηση CHAP

Στη βάση δεδομένων *radius*, ο χρήστης *pingchap* έχει παραμετροποιηθεί ώστε να μπορεί να αυθεντικοποιηθεί στην πόρτα 1812 μέσω πρωτοκόλλου CHAP και η παραμετροποίηση που πρέπει να γίνει στην εφαρμογή για επιτυχή αυθεντικοποίηση, παρουσιάζεται παρακάτω (εικ. 1-16).



Εικόνα 1—16: NTRadPing Test Utility CHAP Authentication

Radius Server/port :192.168.1.12/1812
Reply timeout (sec) :3
Retries: 3
Radius Secret Key: Password2
User-Name: pingchap
Password: 1234
CHAP: Yes
Request type: Authentication Request

Κατά την αυθεντικοποίηση σε CHAP πρωτ, παρατηρούμε ότι στον πίνακα radpostauth καταγράφεται το password σε κωδικοποιημένη μορφή (π.χ. 0xe34f15d5227e19455e4a68f00cc9730a0f).

1.22.3 Παραμετροποίηση NTRadPing Test με αυθεντικοποίηση PAP

Αντίστοιχα στη βάση δεδομένων *radius*, ο χρήστης *pingpap* έχει παραμετροποιηθεί ώστε να μπορεί να αυθεντικοποιηθεί στην πόρτα 1812 μέσω πρωτοκόλλου PAP και η παραμετροποίηση που πρέπει να γίνει στην εφαρμογή για επιτυχή αυθεντικοποίηση, παρουσιάζεται οι παρακάτω:

Radius Server/port :192.168.1.12/1812
Reply timeout (sec) :3
Retries: 3
Radius Secret Key: Password2
User-Name: pingchap
Password: 1234
CHAP: No
Request type: Authentication Request

Κατά την αυθεντικοποίηση σε PAP πρωτόκολλο , παρατηρούμε ότι στον πίνακα radpostauth καταγράφεται το password σε μη κωδικοποιημένη μορφή (π.χ. 1234).

2 Εγκατάσταση & Παραμετροποίηση FreeRADIUS-OpenLDAP

2.1 Εισαγωγή

Ο FreeRADIUS server είναι μία αρθρωτή, RADIUS σουίτα ανεπτυγμένη και κατανεμημένη υπό την άδεια του GNU General Public License, v2 και είναι ελεύθερη για φόρτωμα και χρήση σε Η/Υ. Περιλαμβάνει τον RADIUS server, μια RADIUS client βιβλιοθήκη άδειας BSD , μία PAM βιβλιοθήκη, μία υπομονάδα Apache, πολλές πρόσθετες εφαρμογές σχετικές με το RADIUS και βιβλιοθήκες ανάπτυξης. Ο FreeRADIUS server είναι ο πιο δημοφιλής ανοικτού κώδικα RADIUS server και ο πιο διαδεδομένος RADIUS server στον κόσμο.

Η εφαρμογή καταλόγου χρηστών και ομάδων που χρησιμοποιείται για τον FreeRADIUS server είναι η OpenLDAP. Η διαχείριση του OpenLDAP server γίνεται μέσω της εφαρμογής phpLDAPadmin.

Το LDAP (Lightweight Directory Access Protocol) είναι ένα πρωτόκολλο ανοικτού προτύπου για την πρόσβαση σε υπηρεσίες καταλόγου X.500. Το πρωτόκολλο τρέχει πάνω από το επίπεδο μεταφοράς ενός δικτύου, στην περίπτωση του Διαδικτύου αυτό είναι το TCP. Χρησιμοποιεί τη δικτυακή διαστρωμάτωση TCP/IP για τα επίπεδα δικτύου και μεταφοράς, σε αντίθεση με την περίπλοκη διαστρωμάτωση του μοντέλου OSI. Επίσης υιοθετεί και άλλες απλουστεύσεις, όπως η αναπαράσταση τιμών γνωρισμάτων και δομές πληροφορίας του πρωτοκόλλου ως αλφαριθμητικά κειμένου (strings), τα οποία σχεδιάστηκαν ώστε να γίνεται η υλοποίηση περισσότερο απλή και εύκολη. Οι υπηρεσίες καταλόγου βασίζονται σε μία βάση δεδομένων, η οποία οργανώνει εγγραφές, και είναι βελτιστοποιημένη για διαδικασίες ανάγνωσης και αναζήτησης δεδομένων.

Στο κεφάλαιο αυτό αυτή περιγράφεται η διαδικασία εγκατάστασης του FreeRADIUS server, έκδοσης 2.1.12 σε Λειτουργικό Σύστημα Ubuntu 11.10 (32-bit).

Το Λειτουργικό Σύστημα έχει εγκατασταθεί σε Virtual Machine στην εφαρμογή VMWARE Workstation 8 με τα παρακάτω χαρακτηριστικά: μνήμη RAM 768 MB, 1 core CPU και 8 GB χωρητικότητας σκληρού δίσκου. Ο χρήστης που έχει δημιουργηθεί για την εγκατάσταση της εφαρμογής είναι ο “radius” και έχει δικαιώματα administrator. Στην κάρτα δικτύου έχουν οριστεί οι παρακάτω παράμετροι για την επικοινωνία του VM με το υπόλοιπο VLAN:

IP Address: 192.168.1.12, Netmask: 255.255.255.0, Gateway**:** 192.168.1.1, DNS Server: 192.168.1.1

2.2 Εγκατάσταση του OpenLDAP server

Για την εγκατάσταση του OpenLDAP server εγκαθίσταται το πακέτο slapd. Χρησιμοποιώντας τη γραμμή εντολών εγκαθίσταται η εφαρμογή OpenLDAP:

sudo bash
apt-get install slapd

Δίνεται το password του Administrator:

Administrator password: setupRADIUS

Μετά την εγκατάσταση του OpenLDAP γίνεται η εγκατάσταση των utilities του OpenLDAP:

apt-get install ldap-utils

Στη συνέχεια γίνεται επαναδιαμόρφωση των ρυθμίσεων του OpenLDAP server ακολουθώντας τα παρακάτω βήματα στη γραμμή εντολών:

sudo dpkg-reconfigure slapd

Διάλογος Εγκατάστασης:

Omit openLDAP server configuration? :> No
DNS domain name: => unipi.gr
Organization name: =>unipi
Administrator password: =>setupRADIUS
Database backend to use: =>HDB
Do you want the database to be removed when slapd is purged? : => No
Move old database? : => Yes
Allow LDAPv2 protocol? :=> No

2.3 Εγκατάσταση phpLDAPadmin

Για μεγαλύτερη ευκολία στην διαχείριση του OpenLDAP server και την ύπαρξη γραφικού περιβάλλοντος, γίνεται εγκατάσταση της εφαρμογής phpLDAPadmin.

Στη γραμμή εντολών δίνονται οι παρακάτω εντολές για την εγκατάσταση:

radius@RADIUS:~$ sudo bash
root@RADIUS:~# apt-get install phpldapadmin

Παρακάτω γίνεται διαμόρφωση στις αρχικές ρυθμίσεις του αρχείου config.php ώστε να εισάγονται αυτόματα στο portal του phpLDPadmin το domain name και το username πρόσβασης.

gedit /etc/phpldapadmin/config.php

Γίνεται αλλαγή στις παρακάτω γραμμές του αρχείου:

```
$servers->setValue('server','host','192.168.1.12'); // your address

$servers->setValue('server','base',array('dc=unipi,dc=gr')); //your domain name

$servers->setValue('login','auth_type','session');

$servers->setValue('login','bind_id','cn=unipi,dc=server,dc=gr'); // your DN
```

Αφού αποθηκευθεί το αρχείο γίνεται επανεκκίνηση στον apache και στον LDAP server με τις παρακάτω εντολές:

sudo /etc/init.d/apache2 restart

sudo /etc/init.d/slapd restart

Στη συνέχεια γίνεται διαμόρφωση του αρχείου ldap.conf με την εντολή:

gedit/etc/ldap/ldap.conf

Γίνεται αλλαγή του domain name που ανήκει ο LDAP server και η IP διεύθυνση του:

```
#BASE   dc=UNIPI,dc=gr

#URI    ldap://192.168.1.12
```

Κατά την δοκιμαστική εισαγωγή στο portal του phpLDAPAdmin παρατηρήθηκε ότι στη δημιουργία νέων χρηστών σε νέο LDAP κατάλογο, υπάρχουν 2 σφάλματα στα πεδία uidNumber / gidNumber, όπου δεν επιτρέπουν την ολοκλήρωση της δημιουργίας του και πρέπει να γίνει διόρθωση στα 2 αρχεία posixAccount.xml & posixGroup.xml : http://wiki.debian.org/PhpLdapAdmin

gedit /etc/phpldapadmin/templates/creation/posixAccount.xml

Πρέπει να μαρκαριστεί με σχόλιο το readonly attribute στο uidNumber:

```
<attribute id="uidNumber">
        <display>UID Number</display>
        <icon>terminal.png</icon>
        <order>6</order>
        <page>1</page>
<!--    <readonly>1</readonly> -->
        <value>=php.GetNextNumber(/;uidNumber)</value>
</attribute>
```

και διόρθωση στο:

gedit /etc/phpldapadmin/templates/creation/posixGroup.xml

Πρέπει να μαρκαριστεί με σχόλιο το readonly attribute στο gidNumber:

```
<attribute id="gidNumber">
        <display>GID Number</display>
        <order>2</order>
        <page>1</page>
<!--    <readonly>1</readonly> -->
        <spacer>1</spacer>
        <value>=php.GetNextNumber(/;gidNumber)</value>
        <!--
<value><![CDATA[=php.GetNextNumber(/;gidNumber;false;(&(objectClass=posix
Group));*2,+1000)]]></value> -->
</attribute>
```

Μετά τις αλλαγές αυτές είναι εφικτό πλέον να εισαχθούν τα πρώτα uidNumber / gidNumber και τα υπόλοιπα θα υπολογίζονται αυτόματα,

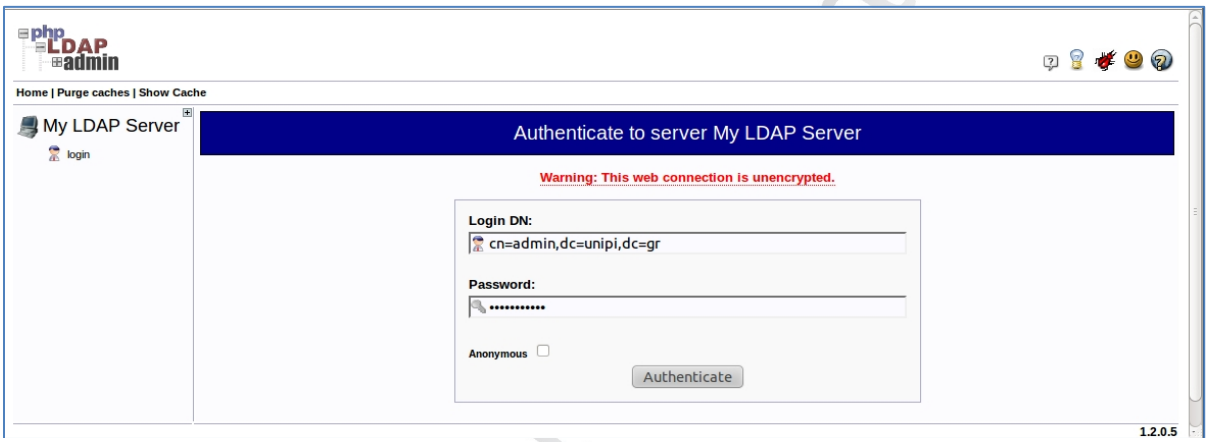
2.4 Παραμετροποιήση του LDAP Directory

Για την παραμετροποίηση της βάσης δεδομένων του LDAP Directory μέσω του phpLDAPserver πρέπει να εισαχθεί σε ένα περιηγητή ιστοσελίδων την παρακάτω διεύθυνση URL:

http://192.168.1.12/phpldapadmin/ ή http://localhost/phpldapadmin/

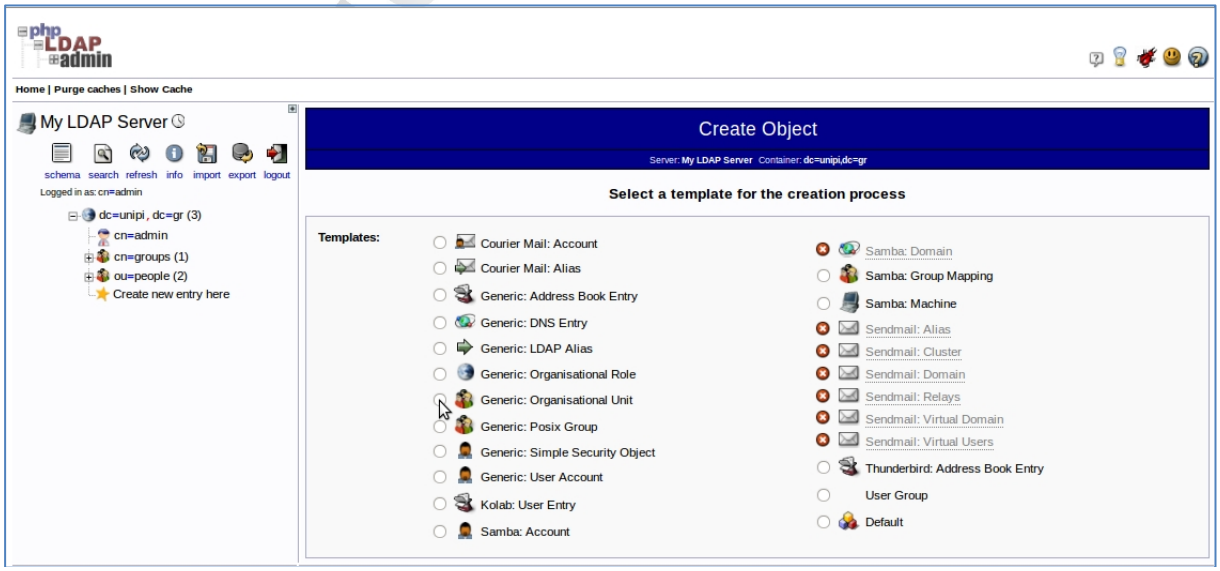
Στα δύο πεδία (εικ. 2-1) συμπληρώνονται το username και το password που δόθηκαν κατά την εγκατάσταση του LDAP server και στη συνέχεια το κομβίο «Authenticate» για εισαγωγή στο LDAP:

Log in: *cn=admin,dc=unipi,dc=gr* Password: setupRADIUS



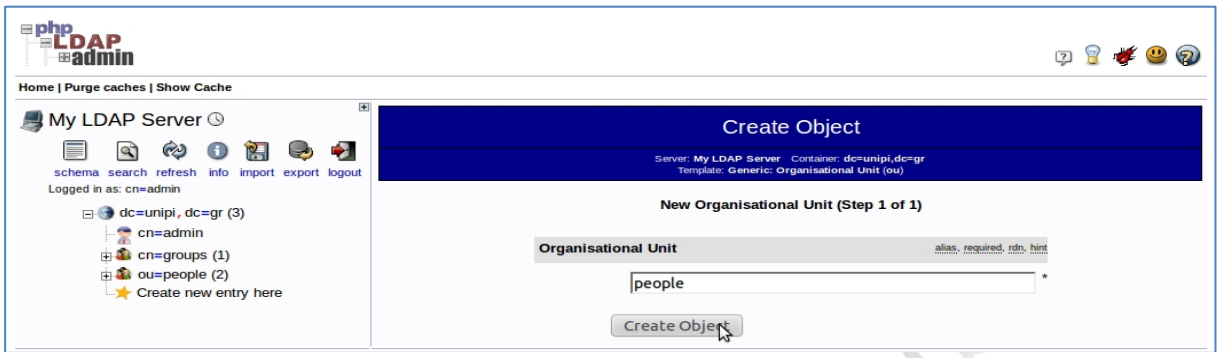
Εικόνα 2—1: Αυθεντικοποίηση στο LDAP με phpLDAPadmin

Στη συνέχεια γίνονται οι απαραίτητες ενέργειες για τη δημιουργία δύο χρηστών " user1" και "user2" όπου θα ενταχθούν στο Organization Unit "people". Επιλέγουμε κάτω από το δέντρο *dc=unipi,dc=gr,* το "*Create new entry here*" και στη συνέχεια επιλογή του "*Generic: Organisational Unit" (εικ. 2-2)*



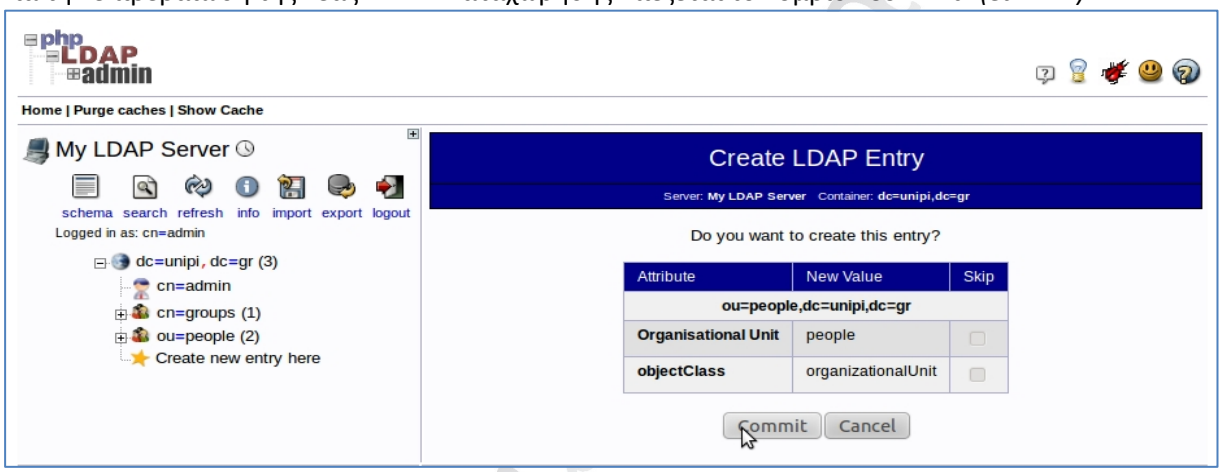
Εικόνα 2—2: Δημιουργία νέας καταχώρησης

Στο επόμενο βήμα δημιουργείται το organization unit *"people"* με το *"Create Object" (εικ. 2-3)*



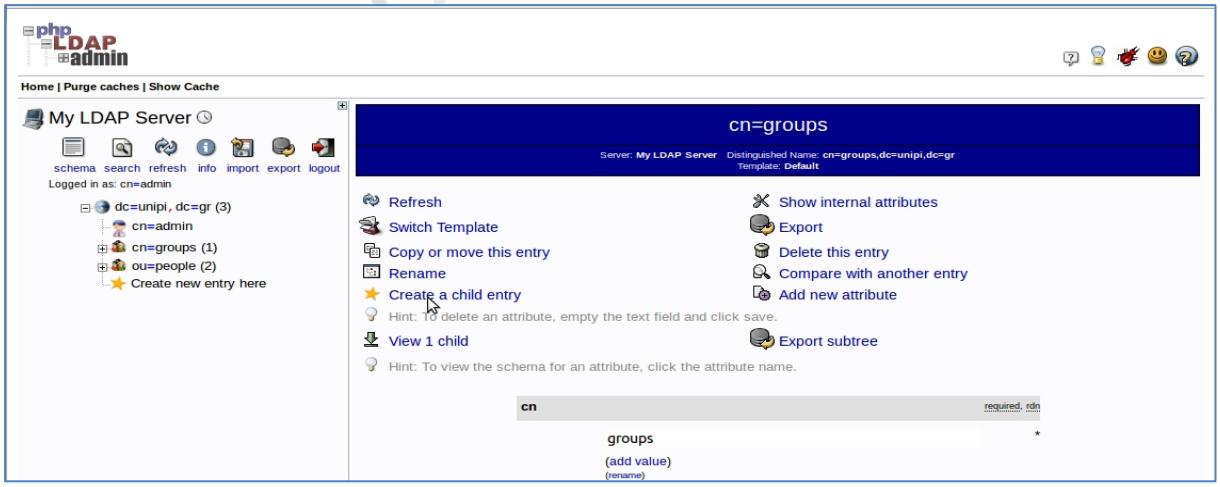
Εικόνα 2—3: Δημιουργία αντικειμένου OU "people"

Για την επιβεβαίωση της νέας LDAP καταχώρησης πιέζεται το κομβίο "Commit" *(εικ. 2-4)*



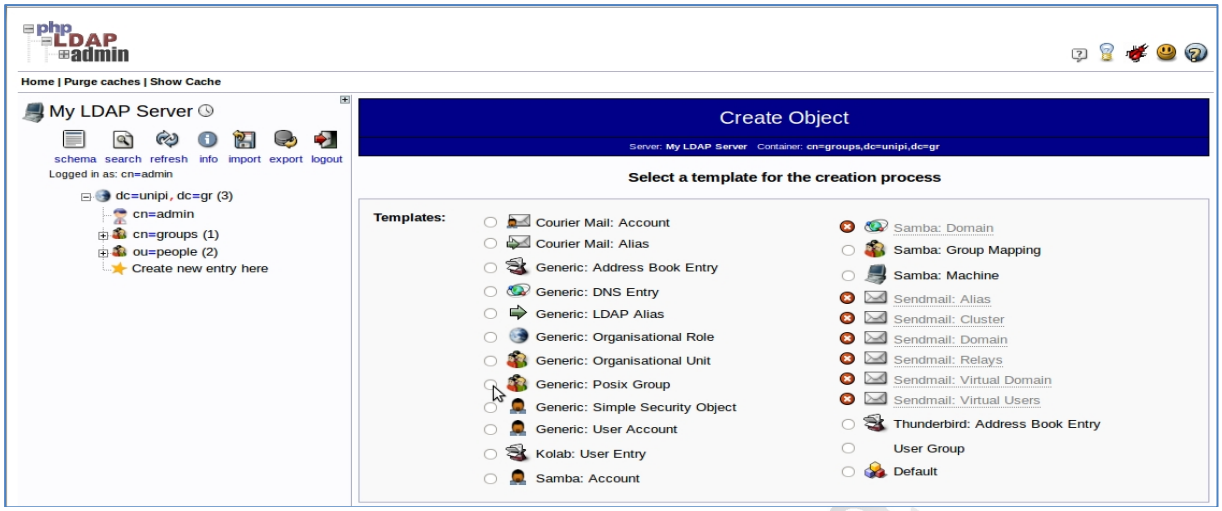
Εικόνα 2—4: Δημιουργία LDAP καταχώρησης

Στη συνέχεια ακολουθώντας την πιο πάνω διαδικασία, δημιουργείται ένα νέο organization unit με την ονομασία *"groups"* και έπειτα δημιουργείται κάτω από το groups μία νέα καταχώρηση, επιλέγοντας το *"Create a child entry" (εικ. 2-5).*



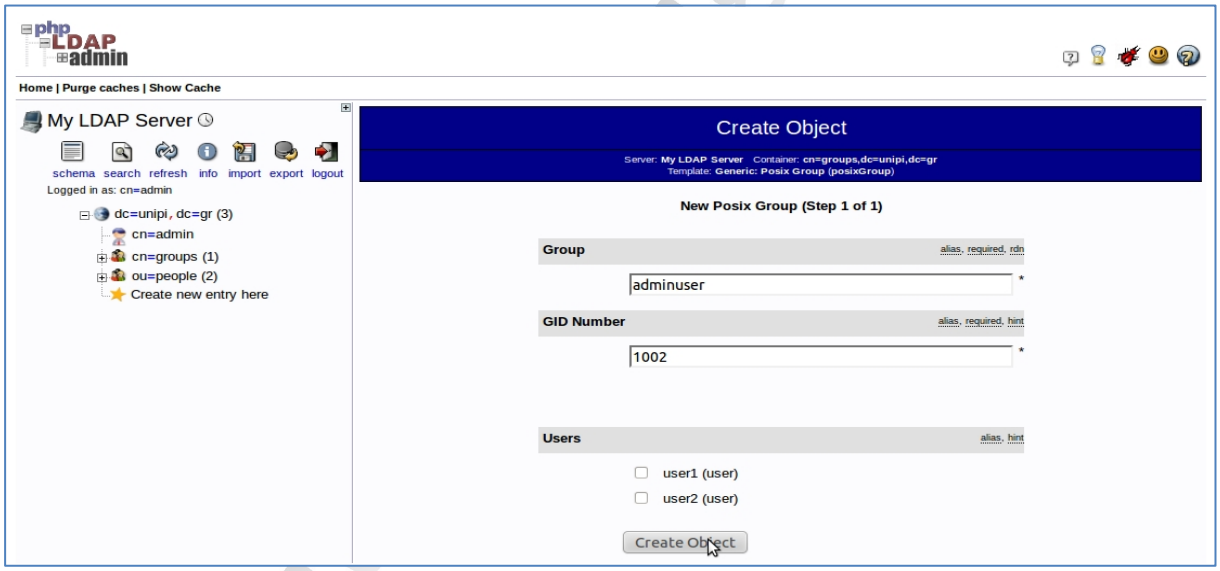
Εικόνα 2—5: Δημιουργία αντικειμένου OU "groups" και δημιουργία "Child entry"

Στη συνέχεια επιλέγεται το *" Generic: Posix Group"* , όπου δημιουργείται κάτω από το *"cn=groups" (εικ. 2-6)*



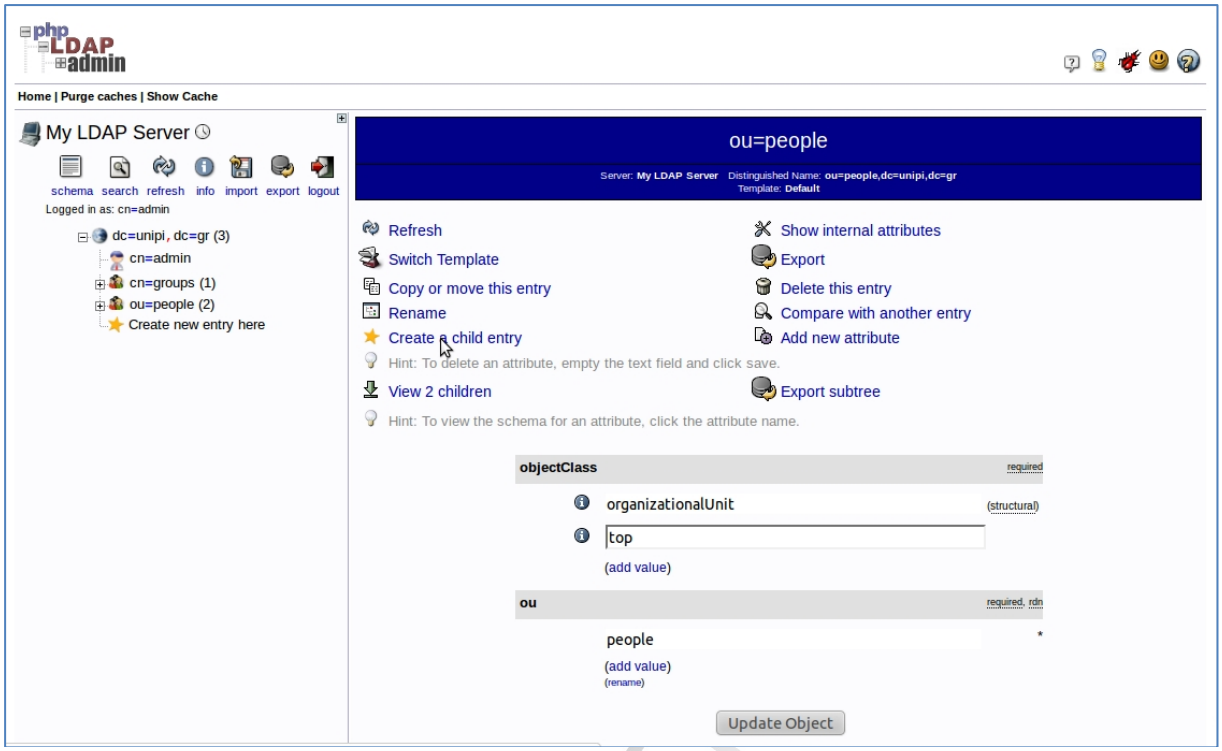
Εικόνα 2—6: Δημιουργία Generic: Posix Group

Στο επόμενο βήμα δίνεται η ονομασία του Posix Group ως *"adminuser"* και στο πεδίο GID Number δίνεται μία αρχική αυθαίρετη τιμή *"1002"* και όλα τα υπόλοιπα GID Number, στη συνέχεια θα μετρούνται αυτόματα *(εικ. 2-7).*



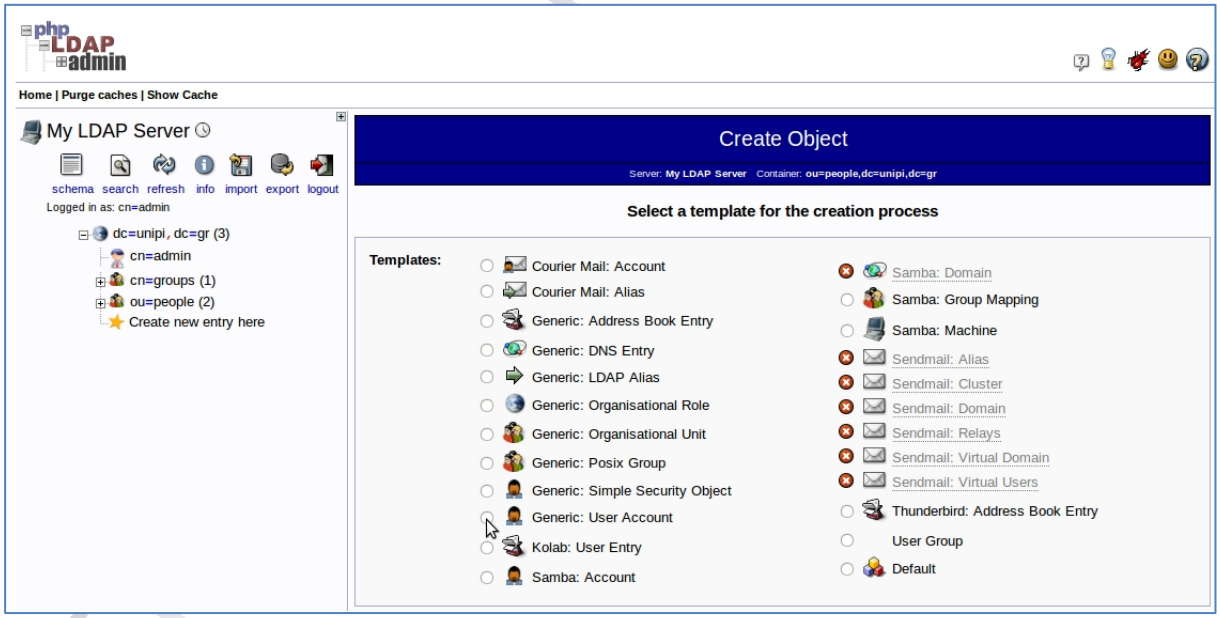
Εικόνα 2—7 : Δημιουργία του αντικειμένου Posix Group "adminuser"

Στο επόμενο βήμα γίνεται επιλογή του OU "people" και επιλέγεται το "Create a new child entry" για τη δημιουργία νέων χρηστών κάτω από το ίδιο OU *(εικ. 2-8).*



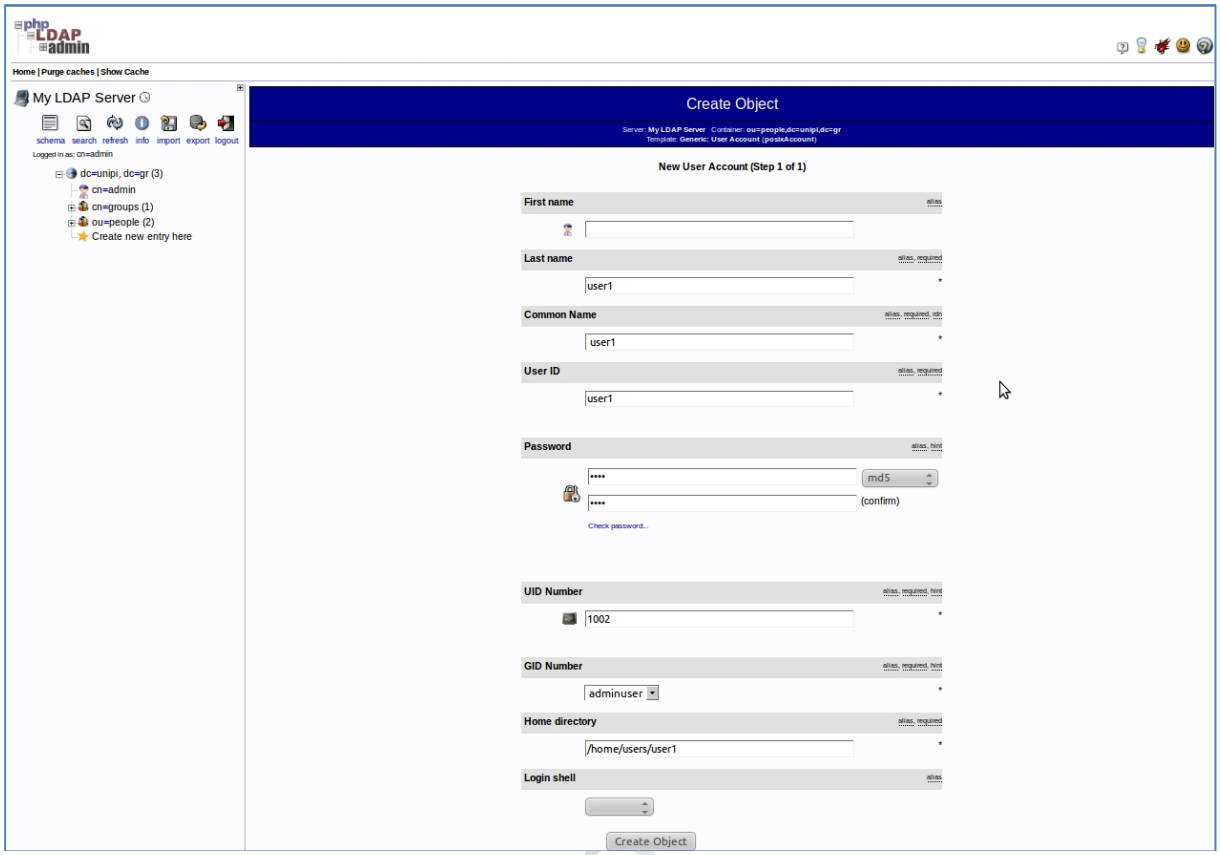
Εικόνα 2—8: Δημιουργία "Child Entry

Στο επόμενο βήμα γίνεται επιλογή δημιουργίας νέου χρήστη, επιλέγοντας το *"Generic:User Account" (εικ. 2-9)*



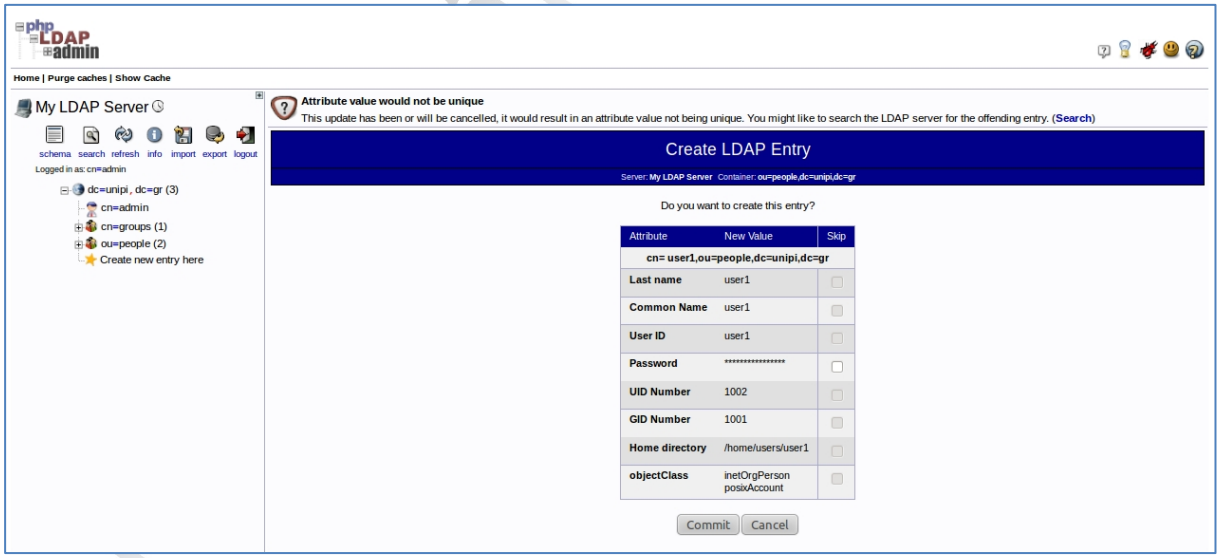
Εικόνα 2—9: Δημιουργία νέου χρήστη στο OU "people"

Στο επόμενο βήμα όπως φαίνεται στην εικ. 2-10 συμπληρώνονται όλα τα πεδία για τη δημιουργία του νέου χρήστη "user1" , (συμπληρώνεται το αρχικό UID Number με αυθαίρετη αρχική τιμή 1002) και στο τέλος πιέζεται το πλήκτρο "Create Object".



Εικόνα 2—10: Δημιουργία χρήστη user1

Στο επόμενο βήμα εμφανίζεται ένα παράθυρο με τα πεδία που εισαχθήκανε στο προηγούμενο βήμα και αν η επαλήθευση επιβεβαιώνεται, τότε πιέζοντας το πλήκτρο “Commit”, δημιουργείται ο νέος χρήστης user1 (Εικ. 2-11).



Εικόνα 2—11: Επιβεβαίωση δημιουργίας του user1

Σύμφωνα με τα ανωτέρω βήματα δημιουργείται και ο χρήστης user2, με τον ίδιο ακριβώς τρόπο που δημιουργήθηκε ο user1.

2.5 Εγκατάσταση FreeRADIUS server

Για την εγκατάσταση του FreeRADIUS server απαιτείται στη γραμμή εντολών να δοθεί η παρακάτω εντολή :

```
apt-get install freeradius
```

Κατά την εγκατάσταση όπου ζητείται, εισάγεται το password.

2.6 Εγκατάσταση του FreeRADIUS με αυθεντικοποίηση LDAP

Για την υποστήριξη του freeRADIUS server ώστε να χρησιμοποιεί μηχανισμό αυθεντικοποίησης LDAP απαιτείται στη γραμμή εντολών να δοθεί η παρακάτω εντολή για την εγκατάσταση:

```
apt-get install freeradius-ldap
```

2.7 Παραμετροποίηση αρχείου clients.conf του FreeRADIUS

Παραμετροποιώντας το αρχείο */etc/freeradius/sites-enabled/clients.conf* δίνεται η δυνατότητα στον FreeRADIUS server να μπορεί να επικοινωνεί με το Access Point όπως και με τον client H/Y που θα επικοινωνεί μαζί του, μέσω του λογισμικού NTRADPingTest.

Παρακάτω δίνεται η εντολή για την παραμετροποίηση του αρχείου:

```
gedit /etc/freeradius/clients.conf
```

Προσθέτω τα παρακάτω δεδομένα μέσα στο αρχείο:

```
client 192.168.1.1 {
    secret          = Password1   -- Μυστικό κλειδί για την αυθεντικοποίηση μεταξύ του AP και τουRadius Server

    shortname       = Access Point PIRELLI
}
client 192.168.1.13 {
    secret          = Password2   -- Μυστικό κλειδί για την αυθεντικοποίηση μεταξύ της εφαρμογής NTRADPINGTEST και τουRadius Server

    shortname       = NTRADPING
}
```

2.8 Παραμετροποίηση αρχείου LDAP του FreeRADIUS

Παρακάτω δίνεται η εντολή για την παραμετροποίηση του αρχείου ldap:

```
gedit /etc/freeradius/modules/ldap
```

```
ldap {
        #
        #  Note that this needs to match the name in the LDAP
        #  server certificate, if you're using ldaps.
        server = "localhost"
```

```
identity = "cn=admin,dc=unipi,dc=gr"  -- Uncommit
password = setupRADIUS                -- Uncommit
basedn = "dc=unipi,dc=gr"
filter = "(uid=%{%{Stripped-User-Name}:-%{User-Name}})"
#base_filter = "(objectclass=radiusprofile)"
```

2.9 Παραμετροποίηση αρχείου DEFAULT του FreeRADIUS

Παραμετροποιώντας το αρχείο */etc/freeradius/sites-enabled/default* δίνεται η δυνατότητα στον FreeRADIUS server να μπορεί να ξεκινήσει, να δημιουργήσει ένα default virtual host και να συνδεθεί στην LDAP βάση δεδομένων.

Με την εκτέλεση της παρακάτω εντολής:

sudo gedit /etc/freeradius/sites-enabled/default

ανοίγει το αρχείο default προς επεξεργασία και θα πρέπει να τροποποιηθούν οι αρχικές ρυθμίσεις όπου χρειάζεται, όπως παρουσιάζεται παρακάτω:

Στο σκέλος *authorize{}*

```
# The ldap module will set Auth-Type to LDAP if it has not
# already been set
ldap   -- (Γίνεται uncomment η ρύθμιση #ldap → ldap)

# Read the 'users' file
#files  --(Γίνεται comment η ρύθμιση files → #files)
```

Στο σκέλος authenticate {

```
Auth-Type LDAP {    -- Γίνεται uncomment

ldap                -- Γίνεται uncomment

}                   -- Γίνεται uncomment
```

2.10 Παραμετροποίηση αρχείου INNER-TUNNEL.CONF του FreeRADIUS

Στο αρχείο inner-tunnel.conf παραμετροποιείται ένας virtual server όπου διαχειρίζεται μόνο τις απαιτήσεις εσωτερικής σήραγγας για τους τύπους EAP-TTLS και PEAP. Η τροποποίηση θα γίνει μόνο στο σημείο που χρειάζεται να βλέπει την LDAP βάση για τη διαχείριση των χρηστών, καταργώντας έτσι το αρχείο users.conf.

Με την εκτέλεση της παρακάτω εντολής:

sudo gedit /etc/freeradius/sites-enabled/ inner-tunnel

ανοίγει το αρχείο *inner-tunnel* προς επεξεργασία και θα πρέπει να τροποποιηθούν οι αρχικές ρυθμίσεις όπου χρειάζεται, όπως παρουσιάζεται παρακάτω:

Στο σκέλος authorize {

```
# Read the 'users' file
#       files     --(Γίνεται comment η ρύθμιση files → #files)

    # The ldap module will set Auth-Type to LDAP if it has not

        # already been set

        ldap      --(Γίνεται uncomment η ρύθμιση ldap → #ldap)
    }

    authenticate {

        Auth-Type LDAP {        -- Γίνεται uncomment

                ldap            -- Γίνεται uncomment

        }                       -- Γίνεται uncomment

    }
```

2.11 Παραμετροποίηση RADIUSD.CONF του FreeRADIUS server

Το αρχείο RADIUS.CONF είναι ο κεντρικός κορμός παραμετροποίησης του Radius Server.

Με την παρακάτω εντολή ανοίγει το αρχείο:

```
sudo gedit /etc/freeradius/radiusd.conf
```

Το “Authentication” του Radius server χρησιμοποιεί την port 1812

```
# Port on which to listen.
# Allowed values are:
#       integer port number (1812)
#       0 means "use /etc/services for the proper port"
port = 1812  ##Αλλαγή σε port 1812
```

Το “Accounting” του Radius server χρησιμοποιεί την port 1813

```
# This second "listen" section is for listening on the accounting
# port, too.
#
listen {
        ipaddr = *
#       ipv6addr = ::
        port = 1813  ##Αλλαγή σε port 1813
        type = acct
#       interface = eth0
#       clients = per_socket_clients
}
```

Στη συνέχεια γίνονται ρυθμίσεις για το logging , username, password κλπ

```
#  Log the full User-Name attribute, as it was found in the request.
#
# allowed values: {no, yes}
#
stripped_names = yes
#  Log authentication requests to the log file.
#
#  allowed values: {no, yes}
#
auth = yes
#  Log passwords with the authentication requests.
#  auth_badpass  - logs password if it's rejected
#  auth_goodpass - logs password if it's correct
#
#  allowed values: {no, yes}
#
auth_badpass = yes      ## Αλλαγή από no σε yes
auth_goodpass = no
```

Στο σκέλος `modules{}` πρέπει το αρχείο clients.conf να είναι ενεργοποιημένο για να μπορεί να διαβάζει από το αρχείο τις IP διευθύνσεις των clients που συνδέονται με τον RADIUS server.

```
$INCLUDE  clients.conf  -- Έλεγχος
```

2.12 Εκκίνηση του Radius Server

Όταν όλες οι ρυθμίσεις έχουν ολοκληρωθεί, μπορεί πλέον ο FreeRADIUS server να ξεκινήσει και σε κανονική λειτουργία με την εντολή *sudo /etc/init.d/freeradius start* και να σταματήσει με την εντολή *sudo /etc/init.d/freeradius stop.* Είναι προτιμότερο όμως να εκτελείται σε debugging mode (με radiusd –X) για συνεχή δυνατότητα ανάγνωσης του log.

Δίνοντας την εντολή:

```
root@RADIUS:/etc# freeradius -X
```

Λαμβάνουμε το παρακάτω σφάλμα:

```
Failed binding to authentication address * port 1812: Address already in use
/etc/freeradius/radiusd.conf[240]: Error binding to port for 0.0.0.0 port 1812
```

Αυτό συμβαίνει ,διότι ο server χρησιμοποιείται ήδη από άλλο session και για να μπορεέσει να ξεκινήσει πρέπει είτε να δοθεί η παρακάτω εντολή:

```
sudo /etc/init.d/freeradius stop
```

Αν αυτό είναι αδύνατο να τερματιστεί το ενεργό process , κάνοντας αρχικά έλεγχο με την παρακάτω εντολή:

```
root@RADIUS:/etc# ps -ef | grep freeradius
```

Εμφανίζοντας το παρακάτω μήνυμα:

```
freerad  27130    1  0 01:06 ?       00:00:00 /usr/sbin/freeradius
root     27772  2097  0 01:56 pts/0   00:00:00 grep --color=auto freeradius
```

Στη συνέχεια εκτελώντας την παρακάτω εντολή για τον τερματισμό του session:

```
root@RADIUS:/etc# kill -9 27130
```

Εκτελώντας ξανά την εντολή, ο server ξεκινάει πλέον κανονικά:

```
root@RADIUS:/etc# freeradius –X
```

2.13 Παραμετροποίηση NTRadPing Test Utility

Το NTRadPing Test Utility όπως περιγράφηκε και στην προηγούμενη ενότητα, είναι μια εφαρμογή για Windows Λ.Σ. όπου μπορεί με κατάλληλη παραμετροποίηση να δοκιμάσει τη λειτουργία του FreeRadius server για Αυθεντικοποίηση.

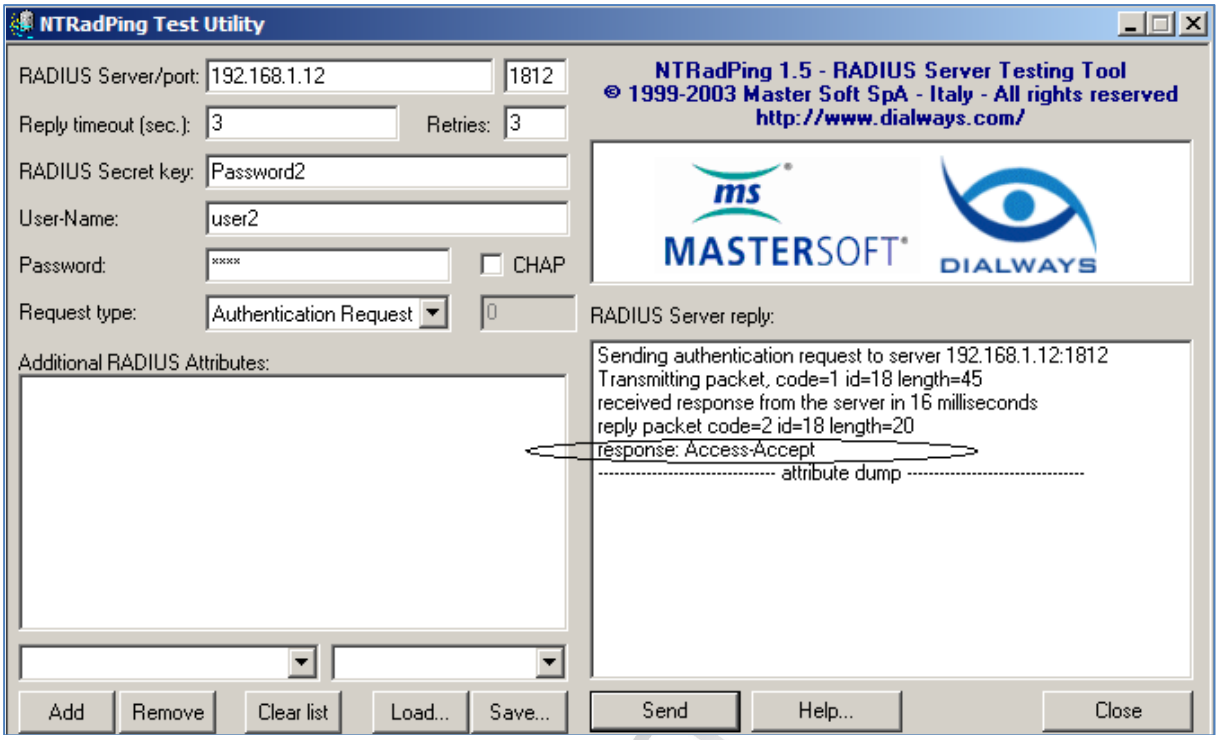
Σε ένα client με εγκατεστημένη την εφαρμογή NTRadPing στο δίκτυο, γίνεται παραμετροποίηση ώστε να χρησιμοποιεί την 1812 port (Authentication) , την IP του RADIUS SERVER, το RADIUS secret key, το username και το password του χρήστη και να επιτυγχάνεται αποστολή αίτησης αυθεντικοποίησης προς τον server.

Αν τα στοιχεία είναι σωστά, σύμφωνα με το αρχείο clients.conf τότε ο χρήστης αυθεντικοποιείται και η εφαρμογή επιστρέφει “response: Access-Accept”. Σε άλλη περίπτωση θα επιστρέψει “No response from Server” και σε περίπτωση μη αυθεντικοποίησης, επιστρέφει “response: Access-Rejected”.

```
Radius Server/port :192.168.1.12/1812
Reply timeout (sec) :3
Retries: 3
Radius Secret Key: Password2
User-Name: user2
Password: 1234
Request type: Authentication Request
```

Όπως φαίνεται στην εικόνα 2-12 , η αυθεντικοποίηση είναι επιτυχής και η απάντηση που λαμβάνει ο client από τον RADIUS server είναι:

```
response: Access-Accept
```



Εικόνα 2—12: NTRadPing Test Utility Authentication

2.13.1 Radius Server Logging

Παρατηρώντας στο log του FreeRADIUS server, βλέπουμε ότι τα δεδομένα που εισαγάγαμε στο client, όπως και η απόκριση που λαμβάνουμε, έχουν καταγραφεί όπως φαίνεται παρακάτω:

```
rad_recv: Access-Request packet from host 192.168.1.13 port 55768, id=18, length=45

User-Name = "user2"

User-Password = "1234"

# Executing section authorize from file /etc/freeradius/sites-enabled/default
+- entering group authorize {...}
++[preprocess] returns ok
++[chap] returns noop
++[mschap] returns noop
++[digest] returns noop
[suffix] No '@' in User-Name = "user2", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
[eap] No EAP-Message, not doing EAP
++[eap] returns noop
[ldap] performing user authorization for user2
[ldap]    expand: %{Stripped-User-Name} ->
[ldap]    ... expanding second conditional
[ldap]    expand: %{User-Name} -> user2
[ldap]    expand: (uid=%{%{Stripped-User-Name}:-%{User-Name}}) -> (uid=user2)
[ldap]    expand: dc=unipi,dc=gr -> dc=unipi,dc=gr
[ldap] ldap_get_conn: Checking Id: 0
[ldap] ldap_get_conn: Got Id: 0
[ldap] performing search in dc=unipi,dc=gr, with filter (uid=user2)
```

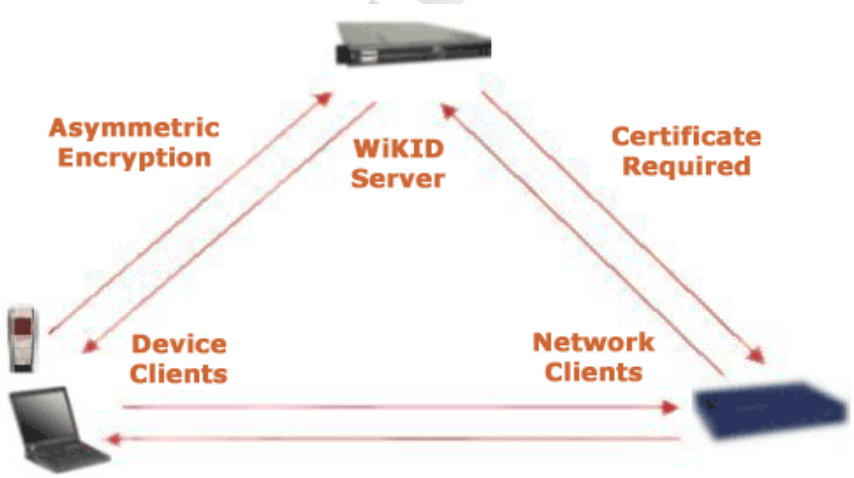
```
[ldap] No default NMAS login sequence
[ldap] looking for check items in directory...
[ldap] userPassword -> Password-With-Header == "{MD5}gdyb21LQTcIANtvYMT7QVQ=="
[ldap] looking for reply items in directory...
[ldap] Setting Auth-Type = LDAP
[ldap] user user2 authorized to use remote access
[ldap] ldap_release_conn: Release Id: 0
++[ldap] returns ok
++[expiration] returns noop
++[logintime] returns noop
[pap] Normalizing MD5-Password from base64 encoding
[pap] WARNING: Auth-Type already set.  Not setting to PAP
++[pap] returns noop
Found Auth-Type = LDAP
# Executing group from file /etc/freeradius/sites-enabled/default
+- entering group LDAP {...}
[ldap] login attempt by "user2" with password "1234"
[ldap] user DN: cn=user2,ou=people,dc=unipi,dc=gr
 [ldap] (re)connect to localhost:389, authentication 1
 [ldap] bind as cn=user2,ou=people,dc=unipi,dc=gr/1234 to localhost:389

 [ldap] waiting for bind result ...

 [ldap] Bind was successful

[ldap] user user2 authenticated succesfully

++[ldap] returns ok
# Executing section post-auth from file /etc/freeradius/sites-enabled/default
+- entering group post-auth {...}
++[exec] returns noop
Sending Access-Accept of id 18 to 192.168.1.13 port 55768

Finished request 5.
Going to the next request
Waking up in 4.9 seconds.
Cleaning up request 5 ID 18 with timestamp +676
Ready to process requests.
```

3 Εγκατάσταση & Παραμετροποίηση WiKID

3.1 Εισαγωγή

Στην παράγραφο αυτή παρουσιάζεται η εγκατάσταση και παραμετροποίηση του συστήματος αυθεντικοποίησης WiKID, το οποίο είναι ένα σύστημα αυθεντικοποίησης δύο παραγόντων (2-Factor) βασισμένο μόνο σε λογισμικό και είναι σχεδιασμένο έτσι ώστε να είναι επεκτάσιμο και να έχει μικρότερο λειτουργικό κόστος αυθεντικοποίησης συγκριτικά με τις συσκευές token, αυξάνοντας επιπλέον την ασφάλεια με την κεντρικοποίηση του έλεγχου αυθεντικοποίησης.

Η αρχιτεκτονική του WiKID βασίζεται στον παρακάτω τρόπο λειτουργίας: (1ος παράγοντας αυθεντικοποίησης) Ο χρήστης επιλέγει τον τομέα WiKID που επιθυμεί να χρησιμοποιήσει και εισάγει το PIN του τομέα, στο WiKID client. Ο Client κρυπτογραφεί το PIN (Personal Identification Number) με το δημόσιο κλειδί του WiKID server, διασφαλίζοντας ότι μόνο ο server μπορεί να το αποκρυπτογραφήσει με το ιδιωτικό κλειδί του. Εάν ο server αποκρυπτογραφήσει επιτυχώς το PIN και είναι σωστό και ο λογαριασμός είναι ενεργός, τότε δημιουργεί ένα κωδικό μιας χρήσης OTP (One-Time-Password) και το κρυπτογραφεί με το δημόσιο κλειδί του client. (2ος παράγοντας αυθεντικοποίησης) Ο χρήστης τότε εισάγει το username του και το OTP σε οποιαδήποτε υπηρεσία χρησιμοποιεί π.χ. VPN , το οποίο προωθεί στον WiKID server για επαλήθευση.



Εικόνα 3—1 Αρχιτεκτονική WiKID Strong Authentication

Το Λογισμικό εγκατάστασης server παρέχεται από το link http://www.wikidsystems.com/downloads και μπορεί να εγκατασταθεί είτε κατευθείαν σε Λειτουργικό Σύστημα RedHat,Centos,Fedora & Ubuntu ή μέσω της μορφής ISO ως έτοιμο Virtual Machine έκδοσης Centos για εγκατάσταση στην εφαρμογή VMWARE, όπου χρειάζεται μόνο εγκατάσταση της εφαρμογής WiKID. Το λογισμικό των clients μπορεί να εγκατασταθεί σε WINDOWS, MAC, LINUX, ANDROID, I-PHONE, BLACKBERRY, WINDOWS MOBILE & SMARTPHONE, όπως και σε pocket PC Palmtops. παρέχεται από το link http://www.wikidsystems.com/downloads/token-clients. Στο server χρησιμοποιήθηκε το Workstation 8 με τα παρακάτω χαρακτηριστικά: μνήμη RAM 768 MB, 1 core CPU και 8 GB χωρητικότητας σκληρού δίσκου, 2 εικονικές κάρτες δικτύου.

3.2 Εγκατάσταση WiKID Strong Authentication Server

Ξεκινώντας την εγκατάσταση της έκδοσης του VM που έχει εγκατασταθεί στο VMWARE, θα χρειαστεί να γίνει μόνο διαμόρφωση των ρυθμίσεων των καρτών δικτύου και στη συνέχεια να ξεκινήσει ο server. Για τη ρύθμιση των καρτών δικτύου πρέπει σε γραμμή εντολών να εκτελεστεί η παρακάτω εντολή:

wikidctl setup

Επιλέγουμε αλλαγή των network settings. Το script εμφανίζει τις υπάρχουσες ρυθμίσεις του δικτύου, και τις τροποποιούμε κατάλληλα:

FQDN=wikid.unipi
Use interface eth0=Yes
IP ADDRESS FOR eth0=192.168.1.15
Use interface sit0=yes
FQDN=wikid2.unipi
IP ADDRESS FOR sit0=192.168.1.16

Έχοντας ολοκληρώσει την παραμετροποίηση των 2 καρτών δικτύου, όπου η πρώτη χρησιμοποιείται για την εσωτερική επικοινωνία του δικτύου και προαιρετικά η δευτερη για εξωτερική διασύνδεση με το διαδίκτυο, στη συνέχεια δημιουργείται το SSL φηφιακό πιστοποιητικό για το server και σε επόμενο βήμα γίνεται εκκίνηση του server με την παρακάτω εντολή:

wikidctl start

Στη συνέχεια από ένα συνδεδεμένο Η/Υ στο ίδιο δίκτυο , εισάγεται η παρακάτω URL διεύθυνση σε Web browser που έχει οριστεί στην εσωτερική κάρτα δικτύου :

https://192.168.1.15/WiKIDAdmin/

Εικόνα 3—2 WiKID Portal

Στα δύο πεδία (εικ. 3-2) γίνεται εισαγωγή των παρακάτω default login credentials:

Username: *WiKIDAdmin* *(mixed-case)*
Password: *2Factor* *(mixed-case)*

Μετά την επιτυχή πρόσβαση στο σύστημα διαχείρισης εμφανίζεται η οθόνη κατάστασης του συστήματος (εικ 3-3). Η οθόνη αυτή παρέχει πληροφορίες σχετικά με στατιστικά των συστημάτων και των υπηρεσιών που παρέχει.

Εικόνα 3—3 Αρχική σελίδα Διαχείρισης

Registered Devices: Ο αριθμός των συσκευών που εξυπηρετούνται από το server.Οι συσκευές αυτές έχουν ολοκληρώσει τη διαδικασία εγγραφής, και μπορούν επιτυχώς να έχουν πρόσβαση σε ασφαλείς πόρους.

Unregistered Devices: Οι συσκευές αυτές έχουν ολοκληρώσει μερικώς τη διαδικασία εγγραφής, αλλά δεν έχουν ολοκληρώσει την αντιστοίχηση της συσκευής σε userid. Σε περίπτωση μη ολοκλήρωσης της εγγραφής, διαγράφονται από το σύστημα μετά από μία ώρα.

Served Domains: Ο αριθμός των ξεχωριστών domains (server codes) που έχουν διαμορφωθεί για αυτό το server.

Network Clients: Ο αριθμός των συστημάτων που χρησιμοποιούν τον server για αυθεντι-κοποίηση.

Protocols Enabled: Ο αριθμός των πρωτοκόλλων που είναι εγκατεστημένα και ενεργο-ποιημένα στον server.

3.3 Εγκατάσταση υποδομής PKI

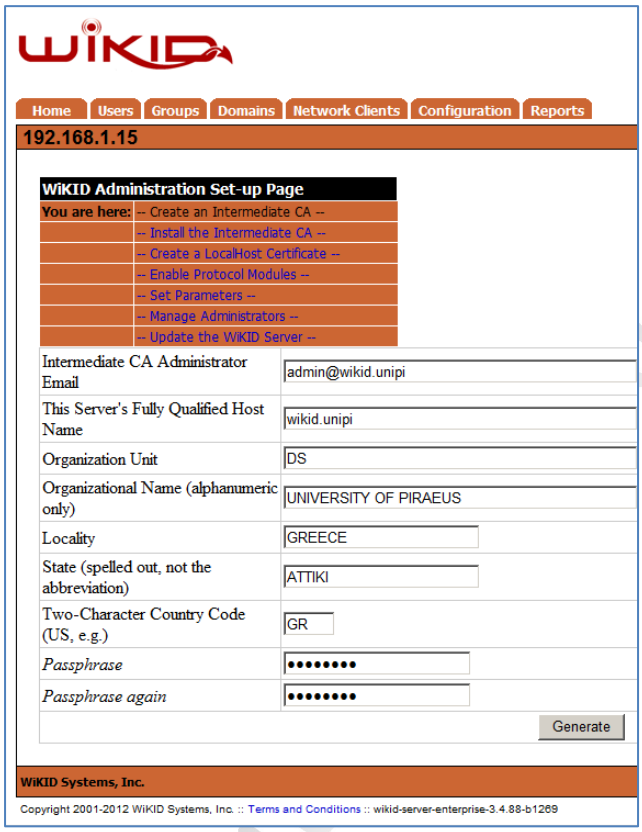
- Κάθε server αυθεντικοποίησης είναι επίσης μια ενδιάμεση αρχή πιστοποίησης
- Κάθε server αυθεντικοποίησης χρησιμοποιεί ψηφιακό πιστοποιητικό για να αναγνωρίσει και να εξουσιοδοτήσει τα clients δικτύου.

Αυτές οι λειτουργίες απαιτούν την εγκατάσταση ενός ψηφιακού πιστοποιητικού πριν ο server γίνει πλήρως λειτουργικός.

3.3.1 Βήμα 1: Δημιουργία Ενδιάμεσου CA

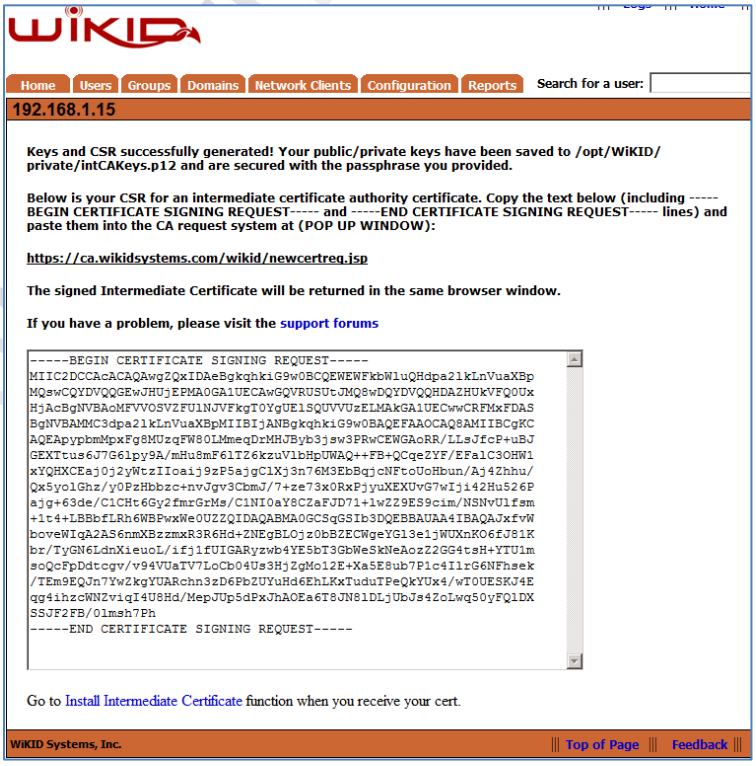
Το πρώτο βήμα στη διαδικασία είναι η δημιουργία Δημόσιων/Ιδιωτικών κλειδιών που θα χρησιμοποιούνται για την αναγνώριση του server και για την ασύμμετρη κρυπτογράφηση των δεδομένων μέσω SSL. Μέσω αυτής της διαδικασίας δημιουργούνται τα κλειδιά και παράγεται το

CSR πιστοποιητικό. Επιλέγουμε “Create an Intermediate CA” συμπληρώνουμε τα στοιχεία και τέλος επιλέγουμε το πλήκτρο “Generate” (εικ 3-4).



Εικόνα 3—4 Δημιουργία Intermediate Certificate Authority

Στη συνέχεια εμφανίζεται το CSR (εικ 3-5), όπου πρέπει να αντιγραφεί για το επόμενο βήμα.



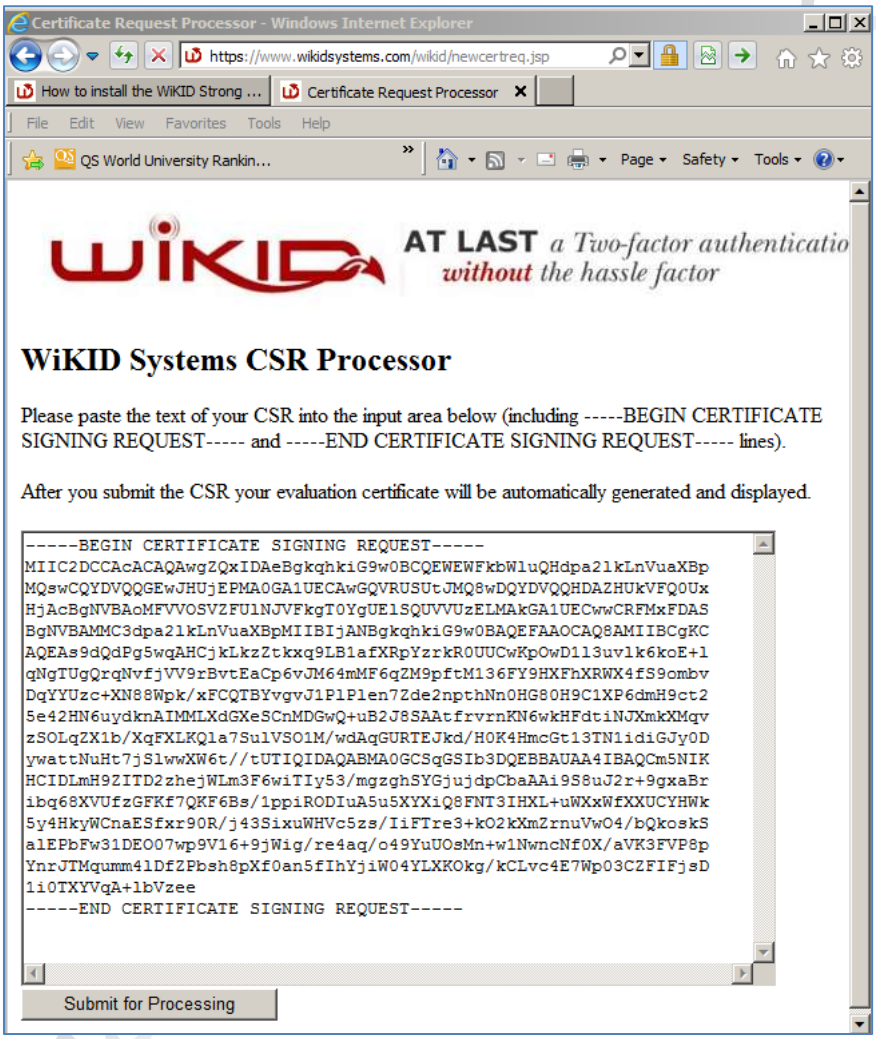
Εικόνα 3—5 CSR

3.3.2 Βήμα 2: Υποβολή του CSR για Υπογραφή

Στη συνέχεια εισάγουμε σε ένα browser το παρακάτω link της WiKID Systems CA:

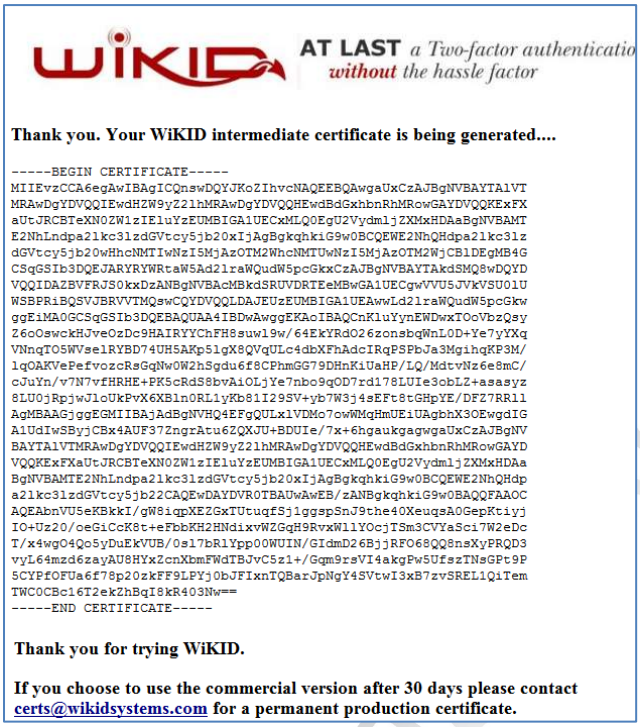
https://www/wikidsystems.com/wikid/newcertreq.jsp

επικολλούμε στον κειμενογράφο το CSR που αντιγράφηκε πριν και τέλος πιέζουμε το πλήκτρο «Submit for Processing" για να σταλεί για υπογραφή (εικ. 3-6).



Εικόνα 3—6 Υποβολή CSR

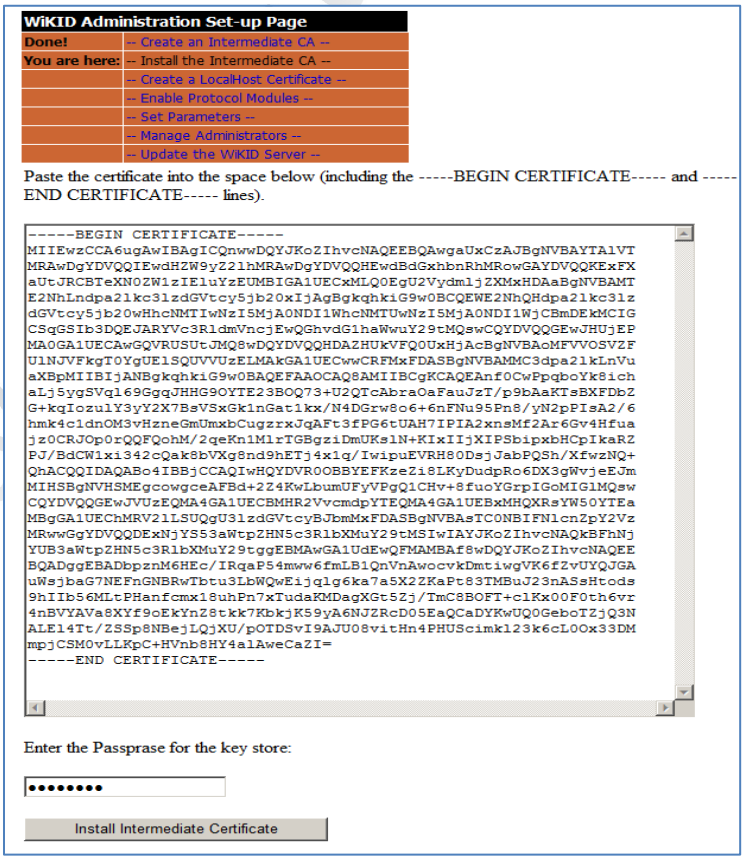
Το υπογεγραμμένο ψηφιακό πιστοποιητικό επιστρέφει στο ίδιο παράθυρο του browser (εικ. 3-7) και αντιγράφουμε το κείμενο του ψηφιακού πιστοποιητικού.



Εικόνα 3—7 Υπογεγραμμένο Ψηφιακό Πιστοποιητικό

3.3.3 Βήμα 3: Εγκατάσταση του Ψηφιακού Πιστοποιητικού

Αφού το ψηφιακό πιστοποιητικό έχει ληφθεί, επιστρέφουμε στην οθόνη διαχείρισης πιστοποιητικών μέσω του πλήκτρου «Install Intermediate CA» και επιλέγουμε το πλήκτρο *"Install"*.



Εικόνα 3—8 Εγκατάσταση Ενδιάμεσου Πιστοποιητικού

Επικολλούμε το ψηφιακό πιστοποιητικό μέσα στο χώρο κειμένου που παρέχεται όπως φαίνεται στην εικ. 3-8. Εισάγουμε το password που χρησιμοποιήσαμε για να ασφαλίσουμε το ιδιωτικό κλειδί στο βήμα 1 και εγκαθιστούμε το ενδιάμεσο πιστοποιητικό. Εάν δε συμφωνεί το password πρέπει η διαδικασία να επαναληφθεί από το βήμα 1. Η εγκατάσταση έχει ολοκληρωθεί επιτυχώς (εικ. 3-9).

Εικόνα 3—9 Εγκατεστημένο CA

3.3.4 Βήμα 4: Δημιουργία Ψηφιακού Πιστοποιητικού Localhost

Όλα τα συστήματα που επικοινωνούν άμεσα με τον server αυθεντικοποίησης, απαιτούν ένα έγκυρο πιστοποιητικό που εκδίδεται από εκείνο τον server. Προτού να μπορέσουν να έχουν πρόσβαση οι clients στον server πρέπει να δημιουργηθεί ένα πιστοποιητικό που υπογράφεται από αυτόν ενδιάμεσο CA πιστοποιητικό. Αυτό αποτρέπει οποιαδήποτε μη εξουσιοδοτημένα συστήματα από την επικοινωνία με τον server αυθεντικοποίησης.

Κάποια πρωτόκολλα όπως *το RADIUS*, το *LDAP*, *το wAuth*, κ.λπ. δεν παρέχουν ευκολίες για αυθεντικοποίηση με πιστοποιητικά ή κρυπτογράφηση μεταφοράς δεδομένων. Ο WiKID server περιέχει υπομονάδες πρωτοκόλλων που μετατρέπουν διαφανώς αυτά τα πρωτόκολλα σε ασφαλείς επικοινωνίες. Αυτό σημαίνει ότι η το LDAP στον WiKID server απαιτεί ένα πιστοποιητικό για να αυθεντικοποιήσει τα διαπιστευτήρια ακόμα κι αν το RADIUS δεν έχει καμία αρχή πιστοποιητικών.

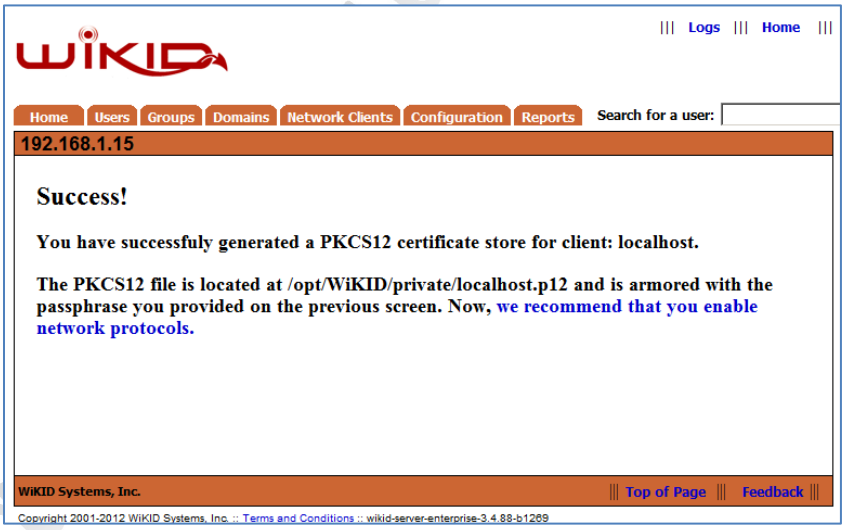
Οι υπομονάδες πρωτοκόλλων που εκτελούνται τοπικά στον WiKID Server, μπορούν να μοιράζονται ένα ανεξάρτητο πιστοποιητικό για το *localhost*.

Μπορούμε να δημιουργήσουμε το πιστοποιητικό αυτό καθορίζοντας την ονομασία ως το FQDN στην οθόνη δημιουργίας πιστοποιητικών. Αυτό παρουσιάζεται στην εικόνα 3-10 συμπληρώνοντας και τα υπόλοιπα στοιχεία που απαιτούνται όπως *Organization Name, Locality, State, Country Code, Client PKCS12 Passphrase,Server keystroke Passphrase.*

Client PKCS12 Passphrase *: P@ssw0rd* Server keystroke Passphrase *: P@ssw0rd*

Εικόνα 3—10 Δημιουργία Πιστοποιητικού

Επιλέγοντας το πλήκτρο “Generate” (εικ. 3-10), θα πρέπει να εμφανισθεί το αποτέλεσμα της εικόνας 3-11.



Εικόνα 3—11 Ολοκλήρωση του PKCS πιστοποιητικού

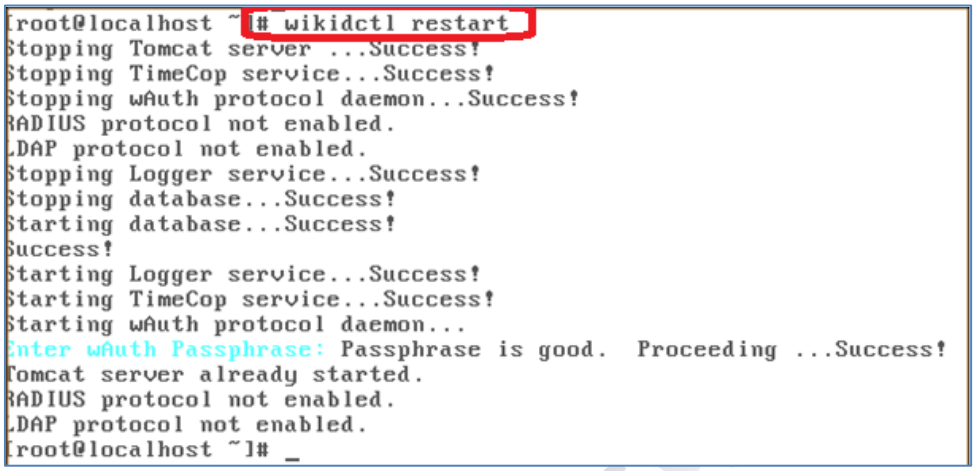
Το μήνυμα αυτό αναφέρει την τοποθεσία αποθήκευσης του ολοκληρωμένου πιστοποιητικού. Εάν αυτό το πιστοποιητικό είναι για τα localhost services, τότε έχει εγκατασταθεί στοκατάλληλο path.

3.3.5 Βήμα 5: Επανεκκίνηση του WiKID Server

Κάνουμε Login στη γραμμή εντολών του server ως root και πληκτρολογούμε:

wikidctl restart

Αυτή η εντολή θα σταματήσει τα services του WiKID server. Θα ζητηθεί ο κωδικός του wAuth. Αυτός είναι ο κωδικός που δημιουργήθηκε στο βήμα 1 για το ενδιάμεσο πιστοποιητικό. Η είσοδος του σωστού κωδικού θα επιτρέψει στον server για να ξεκινήσει, χρησιμοποιώντας το νέο πιστοποιητικό για την αυθεντικοποίηση των clients.



Εικόνα 3—12 Επανεκκίνηση WIKID

Η υποδομή ψηφιακών πιστοποιητικών είναι το στοιχείο κλειδί στην εξασφάλιση των επικοινωνιών μεταξύ του WiKID server και των υπηρεσιών δικτύου που απαιτούν two-factor αυθεντικοποίηση. Έπειτα θα δημιουργήσουμε ένα WiKID domain αυθεντικοποίησης.

3.4 Δημιουργία μιας ενός 2-Factor Τομέα Αυθεντικοποίησης WiKID

Το σύστημα αυθεντικοποίησης WiKID υιοθετεί την έννοια των τομέων (domains) αυθεντικοποίησης. Ένας τομέας αυθεντικοποίησης είναι μια κατάτμηση της αρχής αυθεντικοποίησης. Οποιαδήποτε δεδομένη συσκευή που χρησιμοποιεί το σύστημα, μπορεί να συμμετέχει σε οποιοδήποτε αριθμό τομέων αυθεντικοποίησης. Αυτοί οι τομείς μπορούν να υπάρξουν σε έναν μεμονωμένο WiKID server ή μπορούν να υπάρξουν στους ξεχωριστούς και διακριτούς servers (ή οποιοδήποτε συνδυασμό). Αντιθέτως, ένας WiKID server μπορεί να παρέχει τις υπηρεσίες αυθεντικοποίησης για οποιοδήποτε αριθμό διακριτών τομέων. Αυτοί οι τομείς μπορούν να είναι αποκλειστικοί ή συμπεριλαμβανόμενοι από οποιαδήποτε ομάδα συσκευών.

Ένας τομέας αυθεντικοποίησης καθορίζεται αρχικά από τον κωδικό 12 ψηφίων που χρησιμοποιείται στην διαμόρφωση των clients των συσκευών. Αυτός ο κωδικός επιτρέπει σε οποιαδήποτε μη διαμορφωμένη, ανεξάρτητη συσκευή, να εντοπίσει και να καταχωρήσει με έναν συγκεκριμένο WiKID server και τομέα. Στην πράξη, ο κωδικός 12 ψηφίων δηλώνει μια zero-padded IP διεύθυνση που είναι προσβάσιμη από το διαδίκτυο. Προαιρετικά, μπορεί να υποδείξει ένα πρόθεμα στον τομέα του wikidsystems.net. Για παράδειγμα, ένας WiKID server με τη public IP 83.212.239.150, θα ήταν άμεσα προσβάσιμος μέσω του κωδικού 12 ψηφίων 083212239150. Χρησιμοποιώντας την υπηρεσία του wikidsystem.net, οι κωδικοί που δηλώνουν τις μη δρομολογήσιμες IP διευθύνσεις μπορούν να χρησιμοποιηθούν, σαν 999888777666. Μπορούμε επίσης να αλλάξουμε τις DNS ρυθμίσεις, διαμορφώνοντας το αρχείο jw.properties με ένα token.

Επιλέγοντας την καρτέλα Domains, θα εμφανίσει τους τρέχοντες τομείς που εξυπηρετούνται από τον WiKID server (Εικ. 3-13).

Εικόνα 3—13 Διαμόρφωση Domain

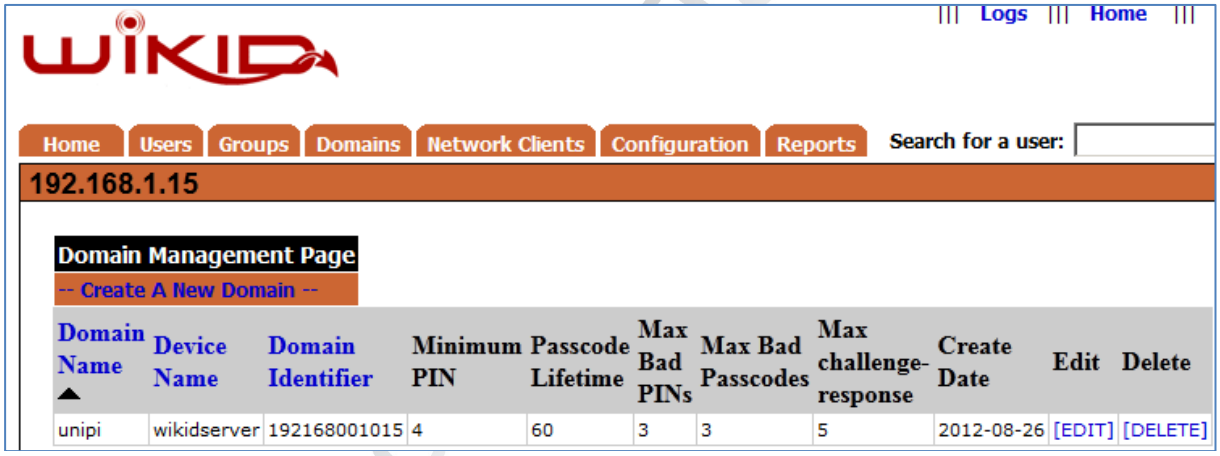
Επιλέγοντας το [Create New Domain] στην ίδια οθόνη, επιτρέπει στον διαχειριστή να εγκαταστήσει ένα νέο τομέα αυθεντικοποίησης για αυτό το server (Εικ. 3-14).

Εικόνα 3—14 Διαμόρφωση Παραμέτρων Domain

Στα απαιτούμενα πεδία έχουν συμπληρωθεί τα παρακάτω δεδομένα για τον server της εργασίας αυτής :

Domain Name : *unipi*
Device Domain Name: *wikidserver*
Registered URL: - *(Χρησιμοποιείται για mutual authentication)*
Server Code: *192168001015*
Minimum PIN Length: *4*
Passcode Lifetime: 60
Max Bad PIN Attempts: *3*
Max Bad Passcode Attempts: *3*
Max Sequential Offlines: *5*
Token types: *Allow all token types*
Use TACACS: -

Μετά τον καθορισμό αυτών των παραμέτρων, επιλέγουμε το πλήκτρο Create για την προσθήκη του νέου domain. Η εικόνα 3-15 παρουσιάζει την επιτυχή δημιουργία του τομέα.



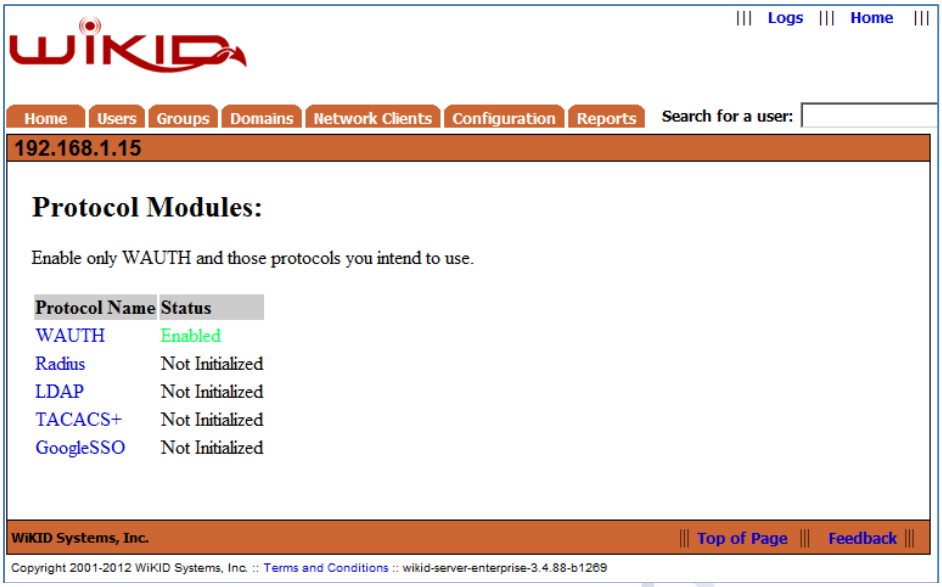
Εικόνα 3—15 Ισχύοντα Domains

3.5 Ενεργοποίηση Πρωτοκόλλων Επικοινωνίας

Οι υπομονάδες πρωτοκόλλων επιτρέπουν στον WiKID Server να παρέχει υπηρεσίες αυθεντικοποίησης σε διάφορους τύπους client δικτύου. Η υποστήριξη παρέχεται για RADIUS, LDAP, TACACS+ και wAuth. Στην εργασία αυτή χρησιμοποιούνται μόνο το wAuth και το RADIUS πρωτόκολλο.

Το πρωτόκολλο wAuth είναι το τοπικό μέσο σύνδεσης στον WiKID server. Χρησιμοποιεί SSL και αυθεντικοποίηση ψηφιακού πιστοποιητικού, για να επιτρέψει στους κατανεμημένους ή τοπικούς clients να επικοινωνήσουν με τα δεδομένα αυθεντικοποίησης πέρα από ένα επισφαλές δίκτυο.

Για την αρχικοποίηση διαμόρφωσης των πρωτοκόλλων επιλέγουμε στην καρτέλα *Configuration,* την επιλογή *[Protocol Modules]*. Παρουσιάζεται μία λίστα από υπομονάδες πρωτοκόλλων, που είναι διαθέσιμα στον server όπως φαίνεται στην εικόνα 3-16.



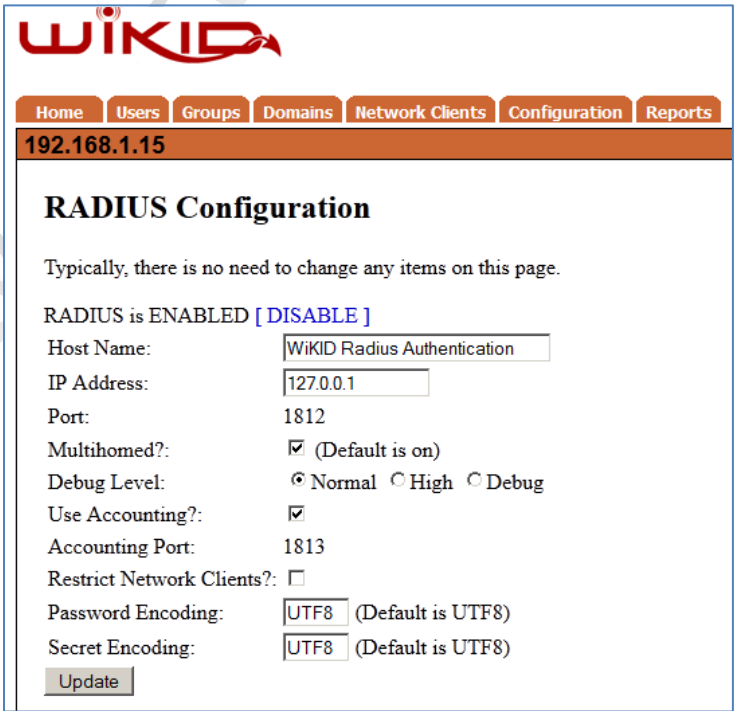
Εικόνα 3—16 Μη αρχικοποιημένα Πρωτόκολλα

Αξίζει να σημειωθεί εδώ ότι το wAuth πρωτόκολλο είναι ενεργοποιημένο αυτόματα, διότι κανένα άλλο πρωτόκολλο δεν μπορεί να λειτουργήσει χωρία αυτό.

3.5.1 Παραμετροποίηση RADIUS

Για την χρησιμοποίηση του RADIUS πρωτοκόλλου πρέπει να γίνει αρχικοποίηση και ενεργοποίηση του RADIUS για συσκευές που βασίζονται στο πρωτόκολλο RADIUS, για την επικοινωνία με τον Authentication server του WiKID server. Ουσιαστικά γίνεται ενεργοποίηση ενός ενσωματωμένου RADIUS server.

Επιλέγουμε το πρωτόκολλο RADIUS από την εικόνα 3-16 για να ανοίξει το παράθυρο αρχικοποίησης παραμέτρων , όπως φαίνεται στην εικόνα 3-17.



Εικόνα 3—17 Παραμετροποίηση RADIUS

Οι παραμετροποίηση μπορεί να παραμείνει όπως είναι οι αρχικές παράμετροι, εάν δεν υπάρχει ανάγκη αλλαγής των παραμέτρων.

Μετά την ενεργοποίηση οποιουδήποτε πρωτοκόλλου, πρέπει να γίνει επανεκκίνηση του WiKID server, όπως έγινε στο Βήμα 5 (παρ. 3.3.5) δίνοντας την εντολή:

wikidctl restart

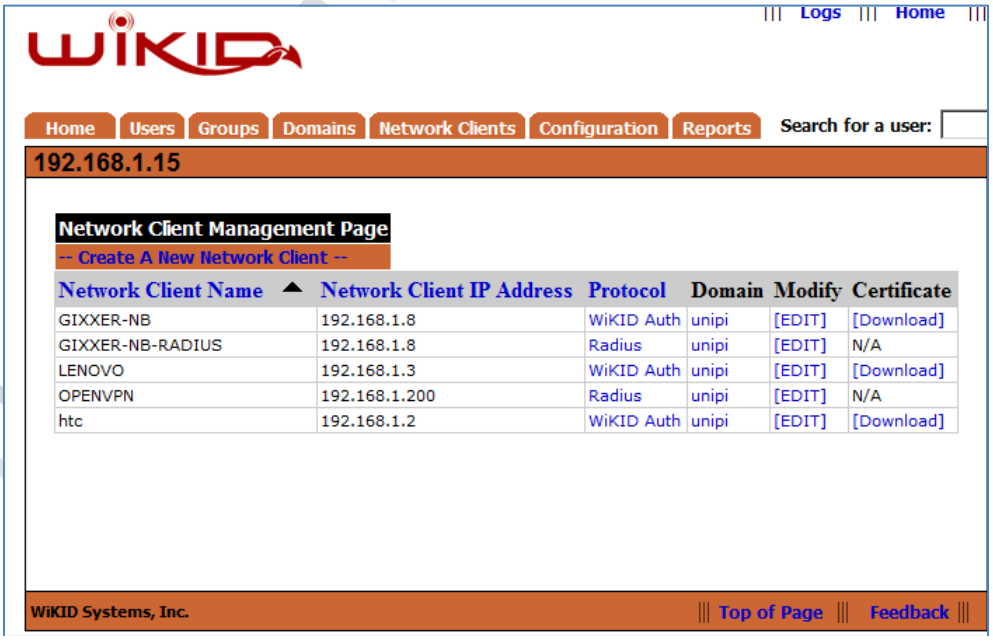
Χρησιμοποιώντας το πρωτόκολλο RADIUS πρέπει να οριστεί ένα "shared secret" που χρησιμοποιείται για την ασφαλή επικοινωνία του πρωτοκόλλου, όπως αναφέρθηκε και στα κεφάλαια 1 και 2.

Το πρωτόκολλο RADIUS στον WiKID server δέχεται μόνο κωδικοποίηση: PAP.CHAP,MSCHAP και MSCHAPV2.

3.5.2 Δημιουργία Network Clients

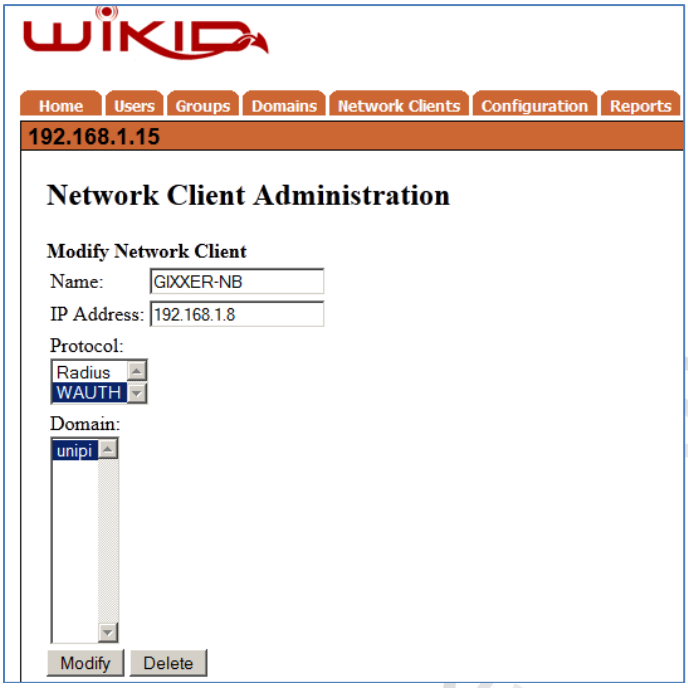
Τα Network Clients είναι συστήματα που ζητούν τη επικύρωση κωδικού πρόσβασης μίας χρήσης OTP από ένα WiKID server. Αυτά τα συστήματα ενεργούν με μια ικανότητα πληρεξούσιου, που δέχεται τις ερωτηθέντες πληροφορίες από τους χρήστες και που επικοινωνεί με τον WiKID server για επιβεβαίωση. Τα Network Clients χρησιμοποιούν μια από τις εγκατεστημένες υπομονάδες πρωτοκόλλου. Η υπομονάδα πρωτοκόλλου πρέπει να εγκατασταθεί, αρχικοποιηθεί και να ενεργοποιηθεί προτού να γίνει προσθήκη ενός Network Client.

Κάθε Network Client πρέπει να παραμετροποιηθεί στον WiKID Server πριν επιτρέψει στον client να ζητήσει επικύρωση. Για τους wAuth clients, αυτό θα απαιτήσει την παραγωγή ενός πιστοποιητικού για το network client.Εξαιρείται ο localhost client, όπου είναι προ-εγκατεστημένος. Η εικόνα 3-18 παρουσιάζεται η αρχικοποίηση των Network Clients.



Εικόνα 3—18 Αρχικοποίηση Network Client

Επιλέγοντας το πλήκτρο " Create new Network Client" εμφανίζεται η οθόνη της εικόνας 3-19.



Εικόνα 3—19 Ιδιότητες Network Client

Αυτές οι τιμές απαιτούνται για κάθε Network Client που διαμορφώνεται, ανεξάρτητα από το πρωτόκολλο που επιλέγεται. Οι ιδιότητες παρουσιάζονται παρακάτω:

Name: Ονομασία του Network Client

IP Address: IP Διεύθυνση του Network Client

Protocol: Το πρωτόκολλο επικοινωνιών που χρησιμοποιείται από αυτόν τον Network Client. Μόνο τα πρωτόκολλα που επιτρέπονται προηγουμένως θα είναι διαθέσιμα. Η επιλογή πρωτοκόλλου θα υπαγορεύσει τις πρόσθετες ιδιότητες που πρέπει να καθοριστούν για αυτόν τον client

Domain: Ο τομέας αυθεντικοποίησης WiKID, στον οποίο ο client θα ζητήσει την επικύρωση της πιστοποίησης του

Εάν ο Network Client δημιουργηθεί για το wAuth πρωτόκολλο, θα πρέπει να δημιουργηθεί ένα πιστοποιητικό για τον Network Client. Ολοκληρώνουμε τις απαραίτητες πληροφορίες όπως φαίνεται στην εικόνα 3-20. Το πεδίο Client's FQDN δεν απαιτεί ένα πραγματικό FQDN. Είναι αποδεκτό να χρησιμοποιηθεί ένα όνομα υπολογιστών ή ένα ψευδώνυμο.

Στα απαιτούμενα πεδία έχουν συμπληρωθεί τα παρακάτω δεδομένα:

Client's FQDN: *GIXXER-NB*
Organization: *UNIPI*
Locality: *PIRAEUS*
State: *ATTIKI*
Country Code: *GR*
Clients PKCS12: *P@ssw0rd*
Passphrase: *P@ssw0rd*
Server Keystore Password: *P@ssw0rd*

Εικόνα 3—20 Δημιουργία Πιστοποιητικού για wAuth Network Client

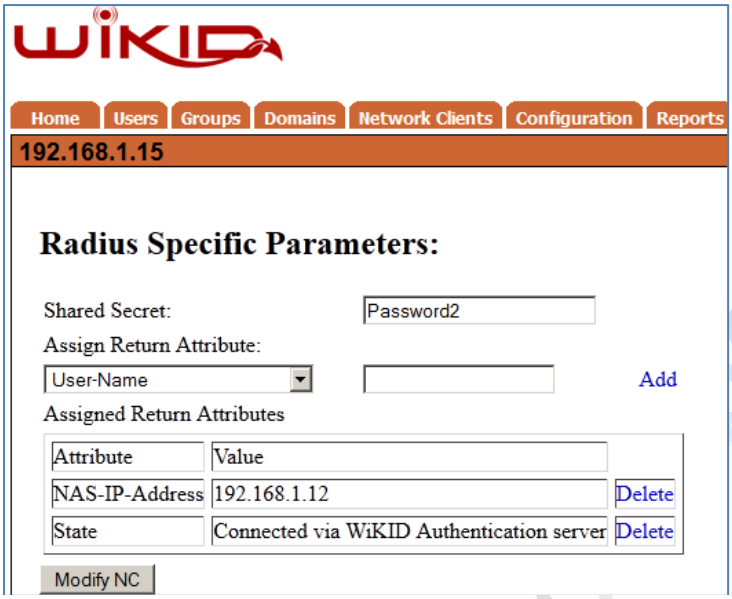
Δημιουργώντας τον RADIUS network client (διαφορετικό πρωτόκολλο) για την ίδια IP, θα πρέπει να δημιουργηθεί νέος Network client , αλλά με επιλογή πρωτοκόλλου RADIUS (εικ. 3-21).

Εικόνα 3—21 Ιδιότητες Network Client RADIUS

Στη συνέχεια επιλέγοντας το πρωτόκολλο RADIUS και το πλήκτρο *Modify* ,εμφανίζεται η οθόνη της εικόνας 3-22 και εισάγονται τα παρακάτω δεδομένα:

Shared secret*: P@ssw0rd2* State: *Connected via WikID Authentication server*

Υπάρχει δυνατότητα να προστεθεί οποιαδήποτε άλλη πληροφορία για απάντηση στον client, μέσω προσθήκης των πεδίων Assigned Return Attributes.

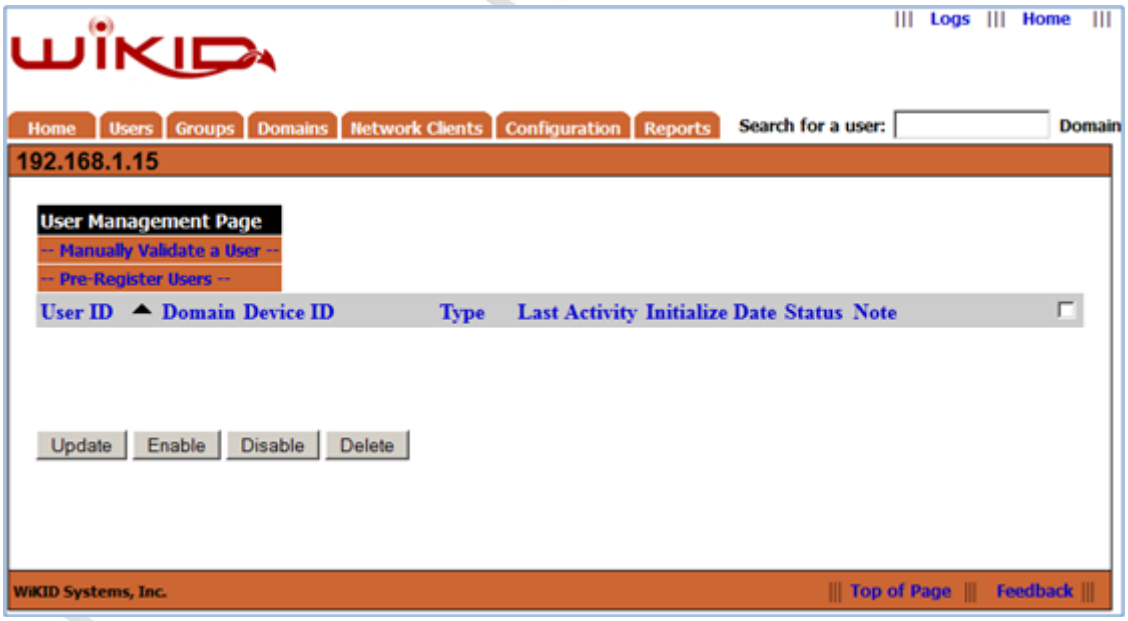


Εικόνα 3—22 Δημιουργία Network Client RADIUS- Shared Secret

3.6 Διαχείριση Χρηστών

Στο σημείο αυτό δε μπορεί να δημιουργηθεί κάποιος χρήστης. Ο χρήστης δημιουργείται κατευθείαν από τον client και στη συνέχεια γίνεται αυτόματη επικύρωση του χρήστη, δίνοντας απλώς ένα username στο χρήστη.

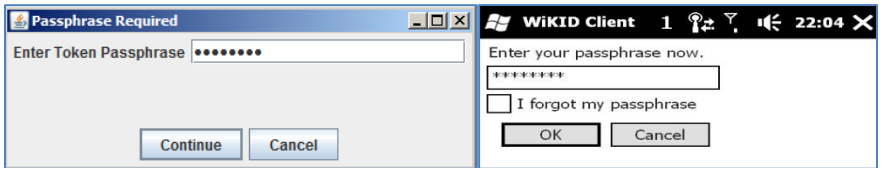
Για τη δημιουργία ενός χρήστη θα πρέπει να επιλεχθεί η καρτέλα *Users.*



Εικόνα 3—23 User Management

Στη συνέχεια πρέπει να ξεκινήσει ο χρήστης την εφαρμογή WiKID Software Token σε Η/Υ ή σε smartphone που παρέχεται από το URL http://www.wikidsystems.com/downloads/token-clients

Η client εφαρμογή που χρησιμοποιείται στην εργασία είναι η wikidtoken-3.1.22.jar στον Η/Υ όπως και σε smartphone Windows Mobile και είναι εκτελέσιμη μέσω JAVA client. Εκτελώντας την εφαρμογή (PC/Palm) ζητείται να δημιουργηθεί ένας κωδικός για την πρόσβαση σε αυτή (εικ 3-24).



Εικόνα 3—24 Κωδικός Token Client

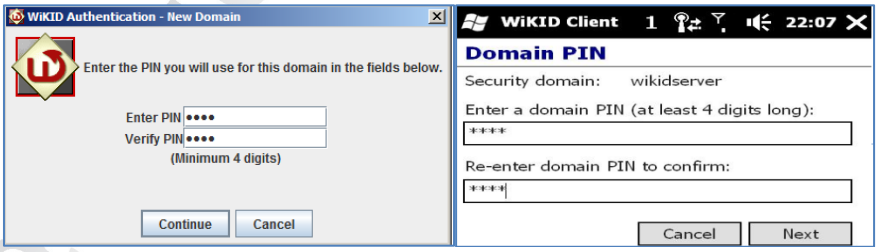
Στη συνέχεια ο χρήστης επιλέγοντας *Actions-> Create New Domain* για Η/Υ και New Domain για το Smartphone δημιουργεί στον client το domain που θέλει να χρησιμοποιεί ως authentication WiKID Server (εικ. 3-25).

Εικόνα 3—25 Δημιουργία νέου Domain στον client

Σε επόμενο βήμα εισάγεται ο 12 ψήφιος κωδικός του WiKID server που αντιστοιχεί στο domain που έχει δημιουργηθεί στον server.

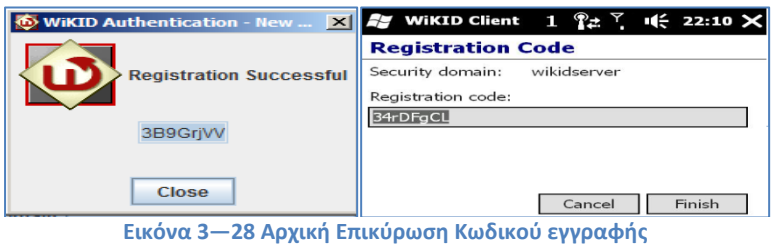
Εικόνα 3—26 Εισαγωγή του Domain Code

Εισάγεται ένα νέο PIN και επιβεβαιώνεται , όπου θα χρησιμοποιεί ο χρήστης αυτός τη συγκεκριμένη συσκευή για το domain αυτό.



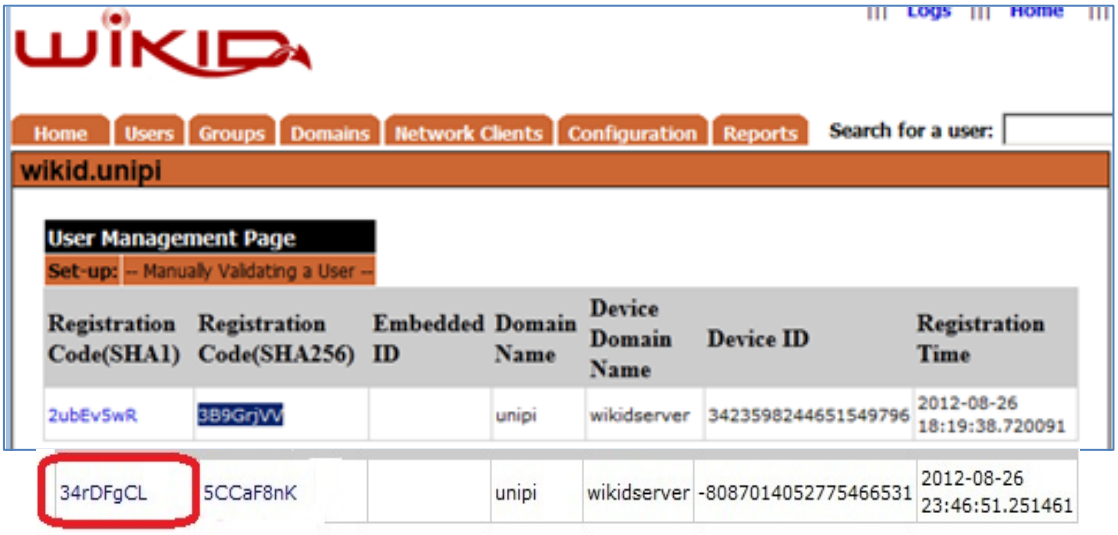
Εικόνα 3—27 Εισαγωγή PIN

Επιστρέφει πίσω ένας κωδικός εγγραφής και χρησιμοποιείται μόνο για την αρχική διαδικασία επικύρωσης του χρήστη.



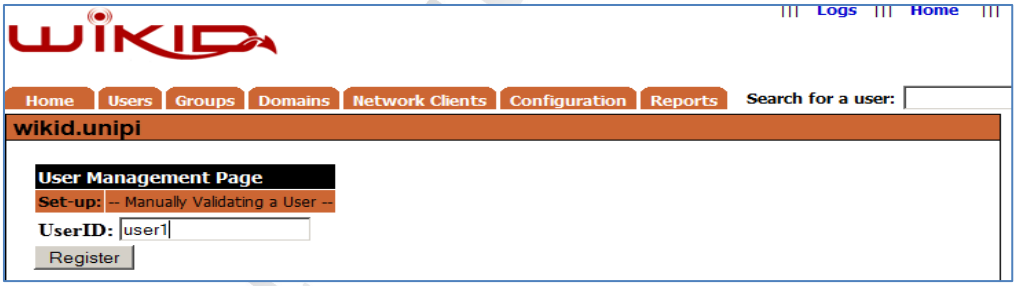
Εικόνα 3—28 Αρχική Επικύρωση Κωδικού εγγραφής

Στην οθόνη διαχείρισης χρηστών WiKID (εικ. 3-23) επιλέγοντας το "Manually Validating a User", εμφανίζεται η οθόνη της εικόνας 3-29 όπου βλέπουμε τους κωδικούς εγγραφής που απαριθμούνται στα SHA1 και SHA256. Εξ ορισμού ένας κωδικός εγγραφής μπορεί να επικυρωθεί οποτεδήποτε μέσα σε 24 ώρες αφότου δημιουργείται. Ο διαχειριστής μπορεί να ελέγξει αυτήν την διάρκεια χρόνου ζωής με την αλλαγή της τιμής UnRegDeviceTTL στις ρυθμίσεις παραμέτρων.



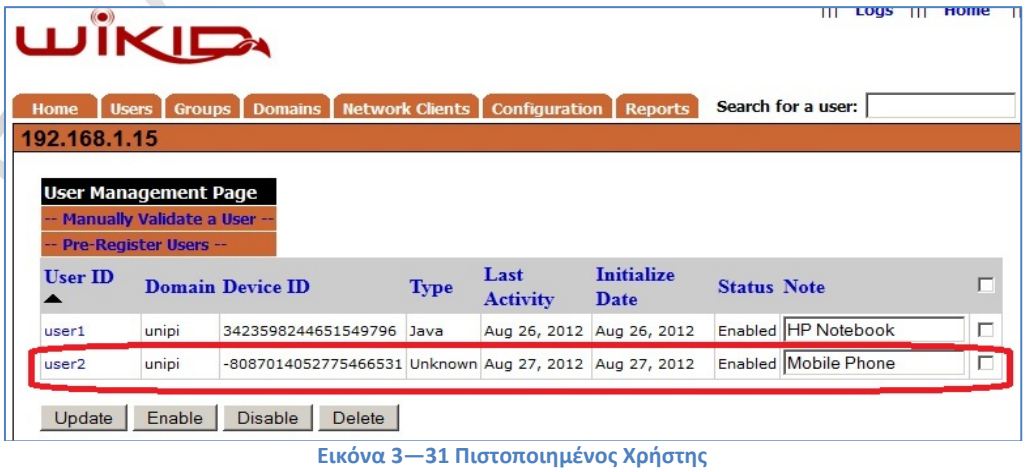
Εικόνα 3—29 Χειροκίνητη Επικύρωση Χρήστη

Έχοντας επιλέξει και επιβεβαιώσει το σωστό Registration Code, εισάγουμε το κατάλληλο user name για τον user1 του χρήστη Η/Υ και User2 για χρήστη Smartphone, όπως φαίνεται στην εικόνα 3-30.



Εικόνα 3—30 Εισαγωγή User name

Επιστρέφοντας στην κύρια οθόνη «User Management» εμφανίζονται οι πιστοποιημένοι χρήστες user1 και user2.



Εικόνα 3—31 Πιστοποιημένος Χρήστης

3.6.1 Έλεγχος One-Time Passcodes στον WiKID Server με πρωτόκολλο wAuth

Για να επιβεβαιώσουμε ότι wAuth λειτουργεί χρησιμοποιώντας το πιστοποιητικό localhost, θα πρέπει να κάνουμε δύο αλλαγές στον κώδικα της example.jsp σελίδας να συνδεθούμε με έναν κωδικό μιας χρήσης OTP. Στη γραμμή εντολών του WiKID server, πληκτρολογούμε την παρακάτω εντολή:

vi /opt/WiKID/tomcat/webapps/WiKIDAdmin/example.jsp

Στη γραμμή 43 γίνεται αλλαγή του WiKID server domain code σε:

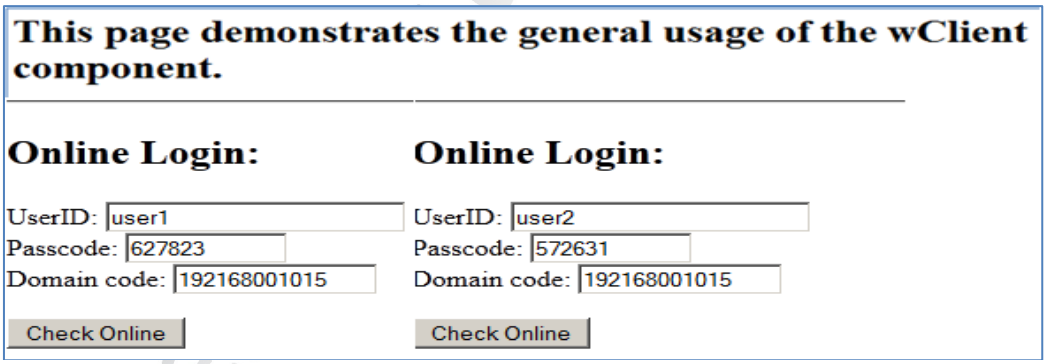
String defaultservercode = "192168001015"

Και στη γραμμή 48 αλλάζει το password αλλάζει σε : *P@ssw0rd*

```
<%
    String defaultservercode = "192168001015";
    String status = "";
    String chall;
    wClient wc;
    if (session.getServletContext().getAttribute("wClient") == null) {
        wc = new wClient("127.0.0.1", 8388, Config.getValue("BASEPATH") + "priva
te/localhost.p12", "P@ssw0rd",
```

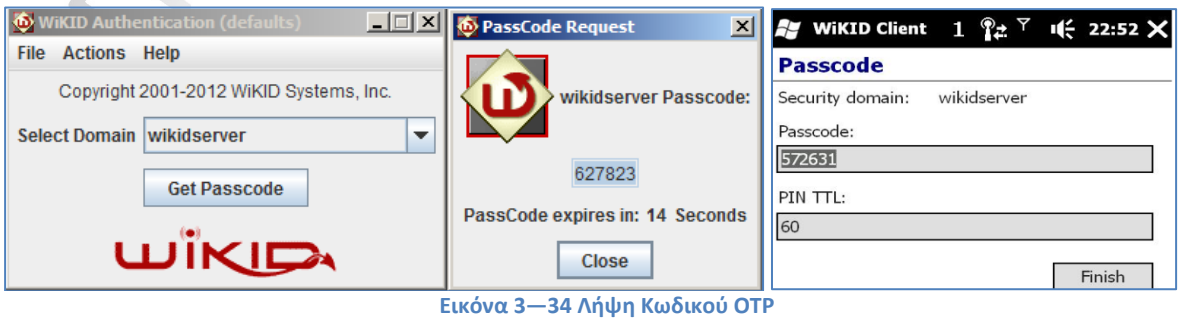
Εικόνα 3—32 Κώδικας Σελίδα Δοκιμής example.jsp

Μετά την αποθήκευση του αρχείου εισάγεται σε ένα browser το URL, https://192.168.1.15/WiKIDAdmin/example.jsp ,εισάγονται τα διαπιστευτήρια του χρήστη WiKIDAdmin και εμφανίζεται η ιστοσελίδα της εικόνας 3-33.



Εικόνα 3—33 Κώδικας Σελίδα Δοκιμής example.jsp

Στα αριστερά πεδία έχουν εισαχθεί τα στοιχεία του user1 με Passcode που έχει ληφθεί από τον client του Η/Υ αφού έχει δοθεί πρώτα το PIN του Domain και στα δεξιά πεδία τα στοιχεία του user2 με Passcode που έχει ληφθεί από τον client του Smartphone.



Εικόνα 3—34 Λήψη Κωδικού OTP

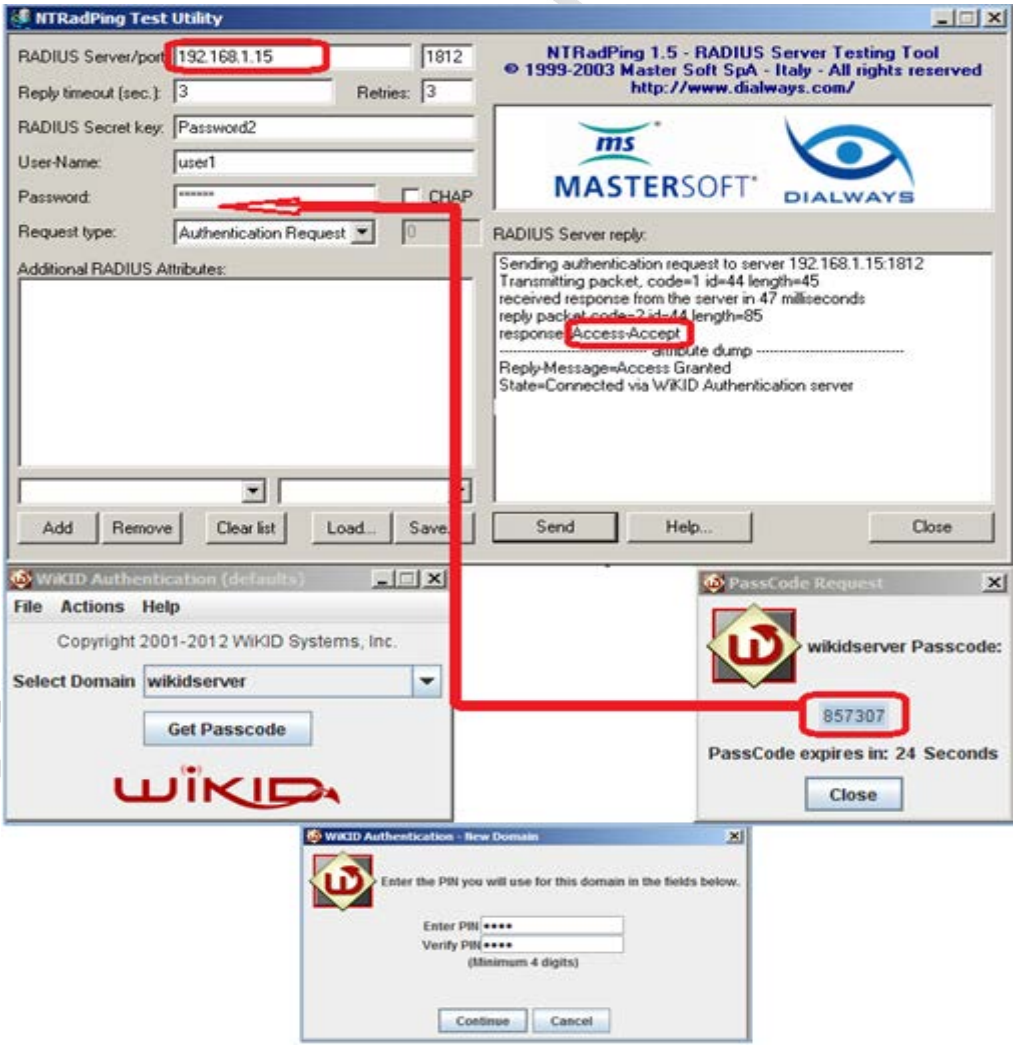
Αν το OTP εισαχθεί εντός του ορισμένου χρόνου 60 sec και είναι σωστό, τότε πατώντας Check Online η αυθεντικοποίηση είναι επιτυχής και το αποτέλεσμα και στις δύο περιπτώσεις είναι:

Success

Ο WiKID Server λειτουργεί πλέον ως σύστημα αυθεντικοποίησης δύο παραγόντων.

3.6.2 Έλεγχος One-Time Passcodes στον WiKID Server με πρωτόκολλο RADIUS

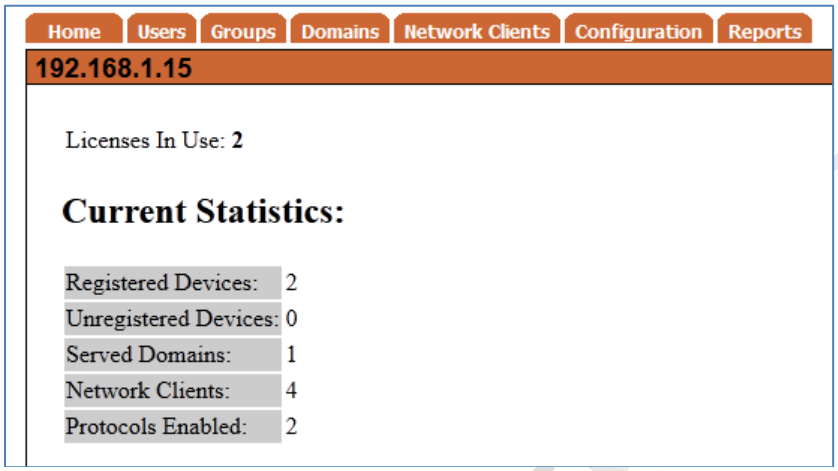
Χρησιμοποιώντας την εφαρμογή NTRadPing Test Utility και έχοντας παραμετροποιήσει την εφαρμογή όπως φαίνεται στην εικόνα 3-35 για RADIUS πρωτόκολλο, εισάγοντας την IP του WikID server με πόρτα 1812 ή 1813 (Accounting), με secret key αυτό που ορίστηκε στην παρ. 3.5.2 και χρησιμοποιώντας είτε τον χρήστη user1 ή τον user2, εισάγοντας κάθε φορά αρχικά το PIN του Domain στην εφαρμογή του Token (1ος παράγοντας) και έπειτα στο Password από το OTP Passcode από το token της εφαρμογής (2ος παράγοντας), μέσα σε 60 secs, τότε η αυθεντικοποίηση γίνεται επιτυχώς και το αποτέλεσμα που επιστρέφει επιτυχώς ο RADIUS είναι Access-Accept όπως φαίνεται στην εικόνα 3-35. Domain PIN: 1234, Token Passcode: *857307*, αντιγραφή NTRadPing Pass: *857307*



Εικόνα 3—35 Αυθεντικοποίηση OTP με RADIUS

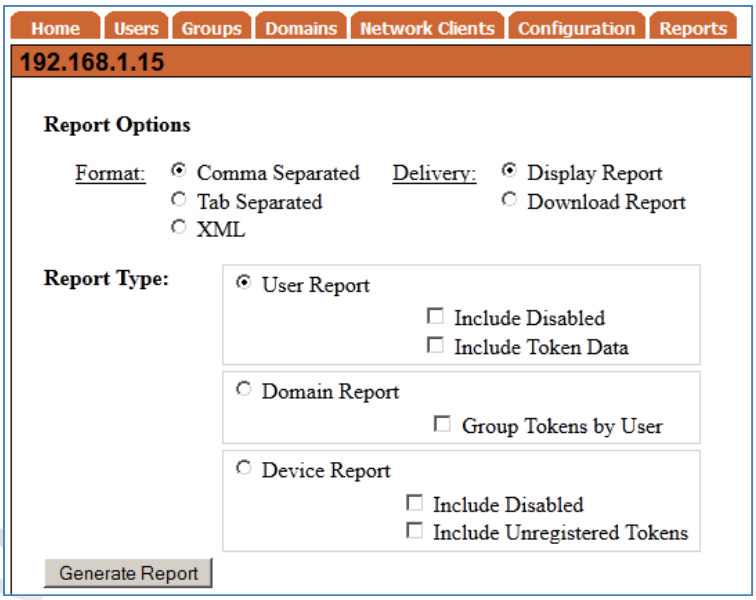
3.7 Στατιστικά και Report

Επιλέγοντας την καρτέλα *"Home"* παρατηρούμε στα τελικά στατιστικά των καταχωρημένων συσκευών, τη μεταβολή ορισμένων στοιχείων μετά την ολοκλήρωση των καταχωρήσεων.



Εικόνα 3—36 Τρέχοντα Στατιστικά

Επιλέγοντας την καρτέλα "Reports" δίνεται η επιλογή να δημιουργηθεί μία αναφορά ανάλογα με την επιλογή που θα γίνει για User,Domain ή Device Report.



Εικόνα 3—37 Δημιουργία Αναφοράς

Πιέζοντας το πλήκτρο *Generate Report* για User Report, λαμβάνουμε την παρακάτω αναφορά για τους δύο χρήστες user1 και user2:

```
username,badPasscodes,userCreation,userStatus,tokenDeviceID,tokenStatus,bad
PINs,tokenExpiration,tokenCreation,domainCode,domainName,deviceDomainName
user1,0,2012-08-26,enabled,3423598244651549796,enabled,2,2015-08-26,2012-
08-26,192168001015,unipi,wikidserver
user2,0,2012-08-27,enabled,-8087014052775466531,enabled,0,2015-08-27,2012-
08-26,192168001015,unipi,wikidserver
```

4 Εγκατάσταση και Παραμετροποίηση OpenVPN

4.1 Εισαγωγή

Το *OpenVPN* είναι μια εφαρμογή λογισμικού ανοιχτού κώδικα που εφαρμόζει τεχνικές εικονικών ιδιωτικών δικτύων (VPN), για τη δημιουργία ασφαλών συνδέσεων «point to point» ή «site to site» σε δρομολογημένες ή γεφυρωμένες υπηρεσίες και σε υπηρεσίες απομακρυσμένης πρόσβασης. Χρησιμοποιεί ένα προσαρμοσμένο πρωτόκολλο ασφάλειας που χρησιμοποιεί SSL/TLS για την ανταλλαγή κλειδιών. Έχει την δυνατότητα να χρησιμοποιει Network Address Translators (NATs) και Firewalls. Η εφαρμογή έχει γράφτεί από τον James Yonan και έχει δημοσιευθεί υπό την άδεια της GNU General Public License (GPL). Το OpenVPN επιτρέπει στις συσκευές που συνδέονται μεταξύ τους, να αυθεντικοποιήσουν η μια την άλλη χρησιμοποιώντας ένα προ-διαμοιραζόμενο μυστικό κλειδί (Pre-Shared Key),ψηφιακά πιστοποιητικά, ή όνομα χρήστη/κωδικό πρόσβασης. Όταν χρησιμοποιείται σε μια διαμόρφωση από multiclient-servers, επιτρέπει στον server να εκδίδει ένα ψηφιακό πιστοποιητικό αυθεντικοποίησης για κάθε πελάτη, χρησιμοποιώντας την Αρχή Ψηφιακών Υπογραφών και Πιστοποιητικών (CA). Χρησιμοποιεί εκτενώς τη βιβλιοθήκη κρυπτογράφησης OpenSSL, καθώς επίσης και το πρωτόκολλο SSLv3/TLSv1 και περιέχει πολλά χαρακτηριστικά λειτουργίες ασφάλειας και ελέγχου.

Κρυπτογράφηση

Το OpenVPN χρησιμοποιεί τη βιβλιοθήκη OpenSSL για να παρέχει κρυπτογράφηση των δεδομένων και των καναλιών ελέγχου. Αυτό επιτρέπει στο OpenSSL να κάνει όλη την εργασία κρυπτογράφησης και αυθεντικοποίησης, επιτρέποντας στο OpenVPN να χρησιμοποιήσει όλους τους κρυπτογραφικούς αλγόριθμους που υπάρχουν διαθέσιμοι στο πακέτο OpenSSL. Μπορεί επίσης να χρησιμοποιήσει τη λειτουργία αυθεντικοποίησης HMAC, για να προσθέσει ένα πρόσθετο στρώμα ασφάλειας στη σύνδεση (που αναφέρεται ως «HMAC Firewall» από το δημιουργό).

Αυθεντικοποίηση

Το OpenVPN έχει διάφορους τρόπους να αυθεντικοποιήσει το κάθε σημείο σύνδεσης στο άλλο. Το OpenVPN διαθέτει λειτουργίες αυθεντικοποίησης με χρήση προ-διαμοιραζόμενων κλειδιών, ψηφιακών πιστοποιητικών και όνομα χρήστη/κωδικό πρόσβασης. Το προ-διαμοιραζόμενο μυστικό κλειδί είναι ο ευκολότερος τρόπος αυθεντικοποίησης, ενώ το ψηφιακό πιστοποιητικό παρέχει τον πιο ασφαλή τρόπο αυθεντικοποίησης και είναι από τις σημαντικότερες λειτουργίες.

Δικτύωση

Το OpenVPN μπορεί να εκτελεστεί πάνω από τα πρωτόκολλα UDP ή TCP, εφαρμόζοντας πολυπλεξία στα SSL tunnels σε μία μονή πόρτα TCP/UDP. Έχει τη δυνατότητα να λειτουργεί μέσω των περισσότερων proxy servers (συμπεριλαμβανομένου του HTTP) και λειτουργεί σε πολύ καλό βαθμό μέσω NAT και μέσω firewalls. Ο Server έχει τη δυνατότητα να “προωθεί” ορισμένες επιλογές παραμετροποίησης δικτύου στους clients. Αυτές περιλαμβάνουν τις IP διευθύνσεις, τις εντολές δρομολόγησης, και μερικές επιλογές σύνδεσης. Το OpenVPN προσφέρει δύο τύπους διασύνδεσης για τη δικτύωση μέσω του Universal driver TUN/TAP. Μπορεί να δημιουργήσει είτε ένα LAYER-3 βασισμένο σε σήραγγα IP (TUN), είτε LAYER-2 βασισμένο σε Ethernet TAP, που μπορούν να

μεταφέρουν οποιοδήποτε τύπο δεδομένων Ethernet. Η πόρτα 1194 είναι ο επίσημος ορισμένος αριθμός πόρτας από τον IANA για το OpenVPN.

Ασφάλεια

Το OpenVPN προσφέρει διάφορες εσωτερικές λειτουργίες ασφαλείας. Εκτελείται στο *userspace* (εικονική μνήμη στον πυρήνα), αντί απαίτησης λειτουργίας σε IP stack (και επομένως στον πυρήνα). Το OpenVPN έχει τη δυνατότητα να σταματήσει *τα προνόμια του root*, να χρησιμοποιήσει *mlockall* για να αποτρέψει την ανταλλαγή των ευαίσθητων στοιχείων στο δίσκο, να εισαγάγει *chroot jail* μετά από την έναρξη και να εφαρμόσει ένα πλαίσιο SELinux μετά την έναρξη. Το OpenVPN τρέχει ένα πρωτόκολλο ασφάλειας συνήθειας βασισμένο σε SSL και TLS. Επίσης παρέχει υποστήριξη έξυπνων καρτών μέσω PKCS#11 βάσισμένο σε κρυπτογραφικά tokens.

Επεκτασιμότητα

Το OpenVPN μπορεί να επεκταθεί με εξωτερικά plug-ins ή scripts που μπορούν να κληθούν στα καθορισμένα σημεία εισόδων. Ο σκοπός αυτών δίνει τη δυνατότητα να επεκταθεί το OpenVPN με πιο προηγμένη δυνατότητα καταγραφής (logging), πιο προηγμένη δυνατότητα αυθεντικοποίησης με διαπιστευτήρια, πιο δυναμικές ενημερώσεις των αντιπυρικών ζωνών, ενσωμάτωση RADIUS κλπ. Στο πηγαίο κώδικα του OpenVPN περιλαμβάνονται plug-ins για αυθεντικοποίηση PAM, LDAP ή SQL.

Στο κεφάλαιο αυτό περιγράφεται η διαδικασία εγκατάστασης του λογισμικού OpenVPN server, έκδοσης 2.3-alpha3 σε Λειτουργικό Σύστημα Ubuntu 11.10 (32-bit). Το λογισμικό παρέχεται από το URL http://openvpn.net/index.php/open-source/downloads.html για απεριόριστη χρονική χρήση και αριθμό χρηστών. Το λειτουργικό σύστημα είναι επίσης διαθέσιμο σε virtual machine για Windows και ESXi (περιορισμένης άδειας 2 χρηστών με γραφικό περιβάλλον), όπως και σε αρχείο εγκατάστασης για λειτουργικά RedHat, Fedora, CentOs σε έκδοση ανοικτής άδειας GNU.

Για τον server χρησιμοποιήθηκε το Workstation 8 με τα παρακάτω χαρακτηριστικά: μνήμη RAM 768 MB, 1 core CPU και 2 GB χωρητικότητας σκληρού δίσκου και μία εικονική κάρτα δικτύου.

Ο χρήστης που έχει δημιουργηθεί για την εγκατάσταση της εφαρμογής είναι ο “radius” και έχει δικαιώματα administrator. Στην κάρτα δικτύου έχουν οριστεί οι παρακάτω παράμετροι για την επικοινωνία του VM με το υπόλοιπο VLAN:

IP Address: 192.168.1.200, Netmask: 255.255.255.0, Gateway**:** 192.168.1.1, DNS Server: 192.168.1.1

4.2 Εγκατάσταση OpenVPN Server

Σε VM Ubuntu 11.10 έχοντας παραμετροποιηθεί η κάρτα δικτύου με την IP 192.168.1.200, και έχοντας πρόσβαση στο διαδίκτυο δίνεται η παρακάτω εντολή για να γίνει εγκατάσταση του OpenVPN server:

```
apt-get install openvpn
```

Εναλλακτικά μπορεί να γίνει εγκατάσταση της επιθυμητής έκδοσης **openvpn-[version].tar.gz** server από το URL που αναφέρεται στην εισαγωγή.

4.3 Δημιουργία Ψηφιακών Πιστοποιητικών

Το πρώτο βήμα στην παραμετροποίηση του OpenVPN server είναι η εγκαθίδρυση μίας υποδομής PKI. Το OpenVPN υποστηρίζει αμφίδρομη αυθεντικοποίηση βασισμένο σε ψηφιακά πιστοποιητικά για αμοιβαία εμπιστοσύνη.

Για να γίνει η δημιουργία του ψηφιακού πιστοποιητικού του OpenVPN server, πρέπει να δημιουργηθεί το πιστοποιητικό ρίζας ROOT CA χρησιμοποιώντας τα παρεχόμενα easy-rsa scripts που βρίσκονται στη διαδρομή /usr/share/doc/openvpn/examples/easy-rsa/2.0.

Στη γραμμή εντολών δίνεται η παρακάτω εντολή για μετάβαση στο φάκελο:

```
cd /usr/share/doc/openvpn/examples/easy-rsa/2.0
```

4.3.1 Δημιουργία ROOT CA Certificate

Για να γίνει η αρχικοποίηση του PKI, δίνονται οι παρακάτω εντολές και ξεκινά η παραγωγή του πιστοποιητικού ROOT CA (ca.key & ca.crt) συμπληρώνοντας τα απαραίτητα πεδία του ψηφιακού πιστοποιητικού:

```
. ./vars
./clean-all
./build-ca
```

Εμφανίζεται στην οθόνη:

```
Generating a 1024 bit RSA private key
.....++++++
.++++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:GR
State or Province Name (full name) [CA]:ATTIKI
```

```
Locality Name (eg, city) [SanFrancisco]:PIRAEUS
Organization Name (eg, company) [Fort-Funston]:UNIVERSITY OF PIRAEUS
Organizational Unit Name (eg, section) []:DIGITAL SYSTEMS
Common Name (eg, your name or your server's hostname) [Fort-Funston CA]:OpenVPN-CA
Name []:
Email Address [me@myhost.mydomain]:
```

4.3.2 Δημιουργία OpenVPN server certificate

Για τη δημιουργία του server Certificate και κλειδιού (server.key, server.crt), στη γραμμή εντολών δίνεται η παρακάτω εντολή και γίνεται η συμπλήρωση τω πεδίων αντίστοιχα:

```
./build-key-server server
```

Εμφανίζεται στην οθόνη:

```
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:GR
State or Province Name (full name) [CA]:ATTIKI
Locality Name (eg, city) [SanFrancisco]:PIRAEUS
Organization Name (eg, company) [Fort-Funston]:UNIVERSITY OF PIRAEUS
Organizational Unit Name (eg, section) []:DIGITAL SYSTEMS
Common Name (eg, your name or your server's hostname) [server]:SERVER
Name []:
Email Address [me@myhost.mydomain]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1234

Sign the certificate? [y/n]:Y
1 out of 1 certificate requests certified, commit? [y/n]Y
Data Base Updated
```

4.3.3 Δημιουργία OpenVPN client certificate

Για τη δημιουργία των client Certificates και κλειδιών που θα έχει η κάθε απομακρυσμένη συσκευή (LAPTOP.key, LAPTOP.crt), στη γραμμή εντολών δίνεται η παρακάτω εντολή για την κάθε client συσκευή που θέλει να συνδεθεί στον OpenVPN server και γίνεται η συμπλήρωση των πεδίων αντίστοιχα:

./build-key LAPTOP

Εμφανίζεται στην οθόνη:

```
Generating a 1024 bit RSA private key
.....++++++
.....++++++
writing new private key to 'LAPTOP.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:GR
State or Province Name (full name) [CA]:ATTIKI
Locality Name (eg, city) [SanFrancisco]:PIRAEUS
Organization Name (eg, company) [Fort-Funston]:UNIVERSITY OF PIRAEUS
Organizational Unit Name (eg, section) []:DIGITAL SYSTEMS
Common Name (eg, your name or your server's hostname) [server]:LAPTOP
Name []:
Email Address [me@myhost.mydomain]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:1234

Sign the certificate? [y/n]:Y
1 out of 1 certificate requests certified, commit? [y/n]Y
Data Base Updated
```

4.3.4 Δημιουργία αρχείου Diffie-Helman 1024-bit παραμέτρων

Για τη δημιουργία του αρχείου dh.pem μήκους 1024-bit , πρέπει να δοθεί η παρακάτω εντολή:

./build-dh

Εμφανίζεται στην οθόνη:

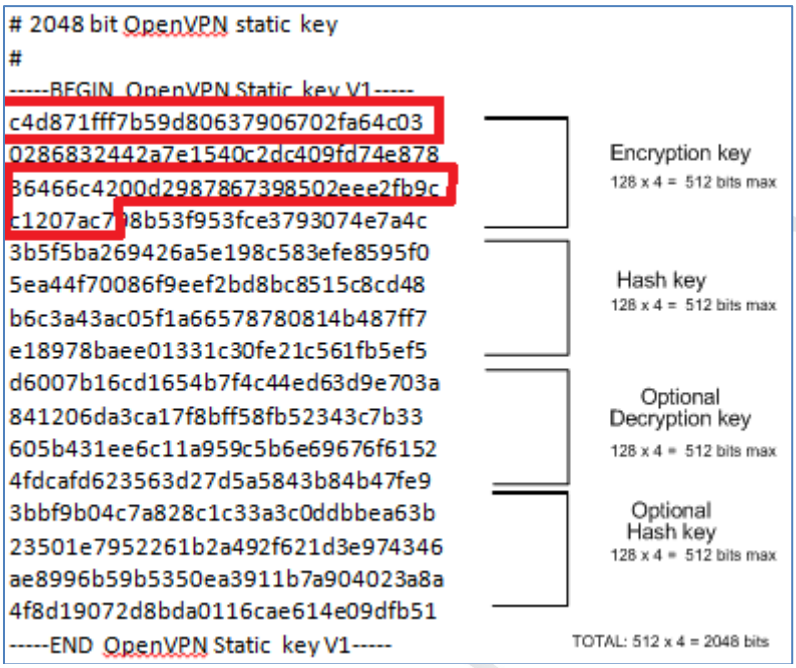
```
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
.....+.....+.....+.....+.....
```

4.3.5 Δημιουργία ψηφιακού πιστοποιητικού HMAC-Firewall (Pre-Shared key)

Για τη μεγαλύτερη ασφάλεια του OpenVPN server από επιθέσεις DoS στην πόρτα UDP, port scanning, buffer overflow, Αρχικοποίηση SSL/TLS από μη εξουσιοδοτημένες συσκευές, χρησιμοποιείται το παρακάτω κλειδί όπου ενσωματώνεται στον server και στον κάθε client που πρόκειται να συνδεθεί μαζί του. Δίνοντας την παρακάτω εντολή δημιουργείται το Pre-Shared Key.

openvpn --genkey --secret ta.key

Δημιουργώντας το παρακάτω κλειδί:



Εικόνα 4—1 PreShared Key

Το πρώτο μέρος του κλειδιού χρησιμοποιείται για να κρυπτογραφεί τα δεδομένα και το δεύτερο για τον αλγόριθμό Hash του κλειδιού. Για την κρυπτογράφηση και τη αποκρυπτογράφηση, χρησιμοποιείται το ίδιο κλειδί και τι ίδιο ακριβώς και για τον αλγόριθμο Hash. Το τρίτο και το τέταρτο κομμάτι του κλειδιού χρησιμοποιείται σε περίπτωση που θέλουμε να χρησιμοποιήσουμε διαφορετικά κλειδιά για την αποκρυπτογράφηση και διαφορετικά κλειδιά για το Hash της αποκρυπτογράφησης. Οι δεκαεξαδικοί χαρακτήρες που είναι έξω από τα κόκκινα πλαίσια, μπορούν να είναι διαφορετικοί στα δύο OpenVPN αρχεία. Στα κόκκινα πλαίσια βρίσκονται οι χαρακτήρες που χρησιμοποιούνται για τα κλειδιά.

4.3.6 Μεταφορά των ψηφιακών πιστοποιητικών

Μετά την ολοκλήρωση όλων των απαραίτητων ψηφιακών πιστοποιητικών και κλειδιών γίνεται μεταφορά στον κύριο φάκελο που είναι εγκατεστημένος ο OpenVPN server /etc/openvpn.

Δίνεται η παρακάτω εντολή:

cp /usr/share/doc/openvpn/examples/easy-rsa/2.0/keys/ca.crt ca.key server.crt server.key dh1024.pem ta.key /etc/openvpn

4.4 Δημιουργία αρχείου παραμετροποίησης του OpenVPN Server

Για να μπορέσει να λειτουργήσει ο OpenVPN server χρειάζεται ένα βασικό αρχείο το server.conf, που μέσα περιέχει τις βασικές λειτουργίες παραμετροποίησης του server, όπως και τις εντολές που καλεί τα modules και τα ψηφιακά πιστοποιητικά που χρειάζεται κάθε φορά.

4.4.1 Μεταφορά του αρχείου server.conf

Με την παρακάτω εντολή, γίνεται μεταφορά του αρχείου server.conf από τον φάκελο δειγμάτων στον κύριο φάκελο /etc/openvpn ,που είναι εγκατεστημένος ο OpenVPN server.

```
cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf  /etc/openvpn
```

4.4.2 Παραμετροποίηση αρχείου server.conf για αυθεντικοποίηση σε RADIUS Server

Δίνοντας την παρακάτω εντολή ξεκινά η παραμετροποίηση στο αρχείο *server.conf*:

```
sudo gedit /etc/openvpn/server.conf
```

Στο αρχικό αρχείο που παρουσιάζεται παρακάτω προσθέτονται εν συνεχεία αρκετές εντολές από το αρχικό, σύμφωνα με τις ανάγκες του δικτύου και της ασφάλειας που χρειάστηκαν στην εγκατάσταση:

```
################ Τα σχόλια μέσα στο αρχείο σημειώνονται με '#' ή με ';' ####################
################ ###########################################################################

#Στην επόμενη εντολή επιλέγεται ως μέθοδος επικοινωνίας του server, το πρωτόκολλο TLS.

    mode server
    tls-server

# Καθορίζεται σε ποια τοπική IP διεύθυνση θα πρέπει ο OPENVPN να ακούει μόνο (Προαιρετικό)

    local 192.168.1.200

# Σε ποια TCP/UDP πόρτα θα πρέπει να ακούει ο OpenVPN. Η προκαθορισμένη είναι η 1194 αλλά
#για λόγους ασφάλειας έχει αλλάξει σε 4596.

    port 4596

# Χρήση TCP ή UDP πόρτας. Επιλέγεται η UDP πόρτα, γιατί η TCP μπορεί να προκαλέσει
#υποβάθμιση της απόδοσηςεπικοινωνίας

    proto udp
    ;proto tcp

# Οι δύο παρακάτω εντολές dev tun και dev tap επιτρέπουν να χρησιμοποιηθεί από το πρωτόκολλο
# OSI, είτε το Επίπεδο #2 (Ζεύξης δεδομένων) ή το Επίπεδο 3 (Δικτύου). Έχει γίνει επιλογή του
#Layer-3, γιατί το Layer 2 μέχρι και αυτή τη στιγμή χρησιμοποιείται μόνο από Windows Clients.

# Η εντολή "dev tun" δημιουργεί ένα tunnel δρομολόγησης IP.
```

```
# Η εντολή "dev tap" δημιουργεί ένα tunnel Ethernet.

        dev tun
        ;dev tap

#Δηλώνονται το SSL/TLS root certificate (ca.crt), το server certificate (server.crt ) και το ιδιωτικό
#κλειδί (server.key). Ο server και όλοι οι clients πρέπει να χρησιμοποιούν το ίδιο ca αρχείο.

        ca ca.crt
        cert server.crt
        key server.key  # This file should be kept secret

# Αρχείο παραμέτρων Diffie Hellman. Η δημιουργία του αναφέρεται στην παρ. 4.3.4.

        dh dh1024.pem

# Παραμετροποίηση του τρόπου λειτουργίας του server και παροχή ενός VPN υποδικτύου για το
#OpenVPN, για να απονέμει διευθύνσεις στου clients που αποκτούν πρόσβαση.
#Ο server θα χρησιμοποιήσει για το VPN δίκτυο τη διεύθυνση 10.8.0.1 για τον ίδιο, και οι
#υπόλοιπες 255 (από 10.8.0.0-10.8.0.255) θα είναι διαθέσιμες για απόδοση στους clients ώστε να
#συνδεθούν με τον server στη διεύθυνση 10.8.0.1.

        server 10.8.0.0 255.255.255.0

#Η εντολή αυτή διατηρεί σε ένα αρχείο μια εγγραφή του κάθε client και συσχετίζεται με την
#εικονική διεύθυνση IP που του έχει αποδοθεί. Εάν το OpenVPN πέσει ή κάνει επανεκκίνηση, στους
#clients που ήταν συνδεδεμένοι θα αποδοθούν οι ίδιες διευθύνσεις που χρησιμοποιούσαν.

        ifconfig-pool-persist ipp.txt

#Σε περίπτωση που θέλουμε να δρομολογήσουμε περεταίρω υπό-δίκτυα πίσω από τον OpenVPN
#server πρέπει να δηλωθούν οι παρακάτω εντολές ( Δεν έχουν χρησιμοποιηθεί στην εργασία) :

        ;push "route 192.168.10.0 255.255.255.0"
        ;push "route 192.168.20.0 255.255.255.0"

# Εάν ενεργοποιηθεί η παρακάτω παράμετρος,  όλοι οι clients που θα συνδεθούν στο VPN θα τους
#αποδοθεί IP Δρομολόγησης αυτή που καθορίζει ο OpenVPN Server, ώστε να χρησιμοποιούν αυτή
#ως προεπιλεγμένη για έξοδο στο Internet και DNS lookup. (Δεν έχει χρησιμοποιηθεί η παράμετρος)

        # push "redirect-gateway 192.168.1.1 bypass-dhcp"

# Με την παρακάτω εντολή μπορούν να αποδοθούν στις ρυθμίσεις της εικονικής κάρτας δικτύου
#που χρησιμοποιούν οι clients για το VPN, καθορισμένες εξωτερικές  DNS διευθύνσεις όπως ο DNS
#της Google που ορίστηκε  και μπορεί  γίνεται name resolve από εκεί.


        push "dhcp-option DNS 8.8.8.8"
```

 # Η εντολή duplicate-cn δεν επιτρέπει σε ένα μηχάνημα που χρησιμοποιεί το ίδιο common name με #ένα άλλο μηχάνημα στο ψηφιακό πιστοποιητικό/κλειδί τους να έχουν το ίδιο common name.Αυτό #σημαίνει ότι κάθε client πρέπει να χρησιμοποιεί μοναδικό cn.

```
duplicate-cn
```

 # Η εντολή keepalive 10 120 προκαλεί αποστολή μηνυμάτων ping μπροστά και πίσω στη σύνδεση. #Έτσι ώστε κάθε πλευρά να γνωρίζει πότε η άλλη πλευρά είναι κάτω. Το ping γίνεται κάθε 10 #δευτερόλεπτα και σημαίνει ότι η απέναντι πλευρά είναι κάτω αν δε δεχθεί αντίστοιχα ping, σε #διάστημα 120 δευτερολέπτων .

```
keepalive 10 120
```

#Ενεργοποίηση του HMAC-Firewall χρησιμοποιώντας το pre-shared key (ta.key) που περιγράφεται #στην παράγραφο 4.3.5. Ο server και κάθε client που συνδέεται στο server, πρέπει να έχει το κλειδί. # Η δεύτερη παράμετρος καθορίζεται ως 0, όπου καθορίζει το server και 1 ορίζεται στα clients.

```
tls-auth ta.key 0 # Μυστικό Κλειδί
```

#Γίνεται επιλογή κρυπτογραφικού αλγορίθμου τρέχοντας την εντολή #openvpn --show-ciphers # Επιλέγεται ο ανώτερος AES-256-CBC. Η ίδια παραμετροποίηση πρέπει να γίνει και στο αρχείο του #client.

```
cipher AES-256-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES
```

 # Γίνεται επιλογή του αλγορίθμου Hash,τρέχοντας την εντολή #openvpn --show-tls # Επιλέγεται ο ανώτερος RSA-SHA256. Η ίδια παραμετροποίηση πρέπει να γίνει και στο αρχείο του #client.

```
auth RSA-SHA256
```

 # Ενεργοποιείται συμπίεση τύπου LZO, στην σύνδεση VPN. # Πρέπει να ενεργοποιηθεί αντίστοιχα και στους clients.

```
comp-lzo
```

 # Επιλέγεται ο μέγιστος αριθμός 30 συνδεόμενων ταυτόχρονα clients

```
max-clients 30
```

#Για την ασφαλέστερη λειτουργία και ελαχιστοποίηση των δικαιωμάτων πρόσβασης στον daemon #του OpenVPN μετά την αρχικοποίηση, χρησιμοποιείται η εντολή user nobody ,όχι όμως και η # εντολή group nobody

```
user nobody
#group nobody
```

 # Οι δύο παρακάτω εντολές χρησιμοποιούνται για την αποφυγή της πρόσβασης σε κάποιους #ιδιαίτερους πόρους στην επανεκκίνηση, που θα μπορούσαν να μην είναι πλέον προσβάσιμοι #εξαιτίας υποβάθμισης προνομίων.

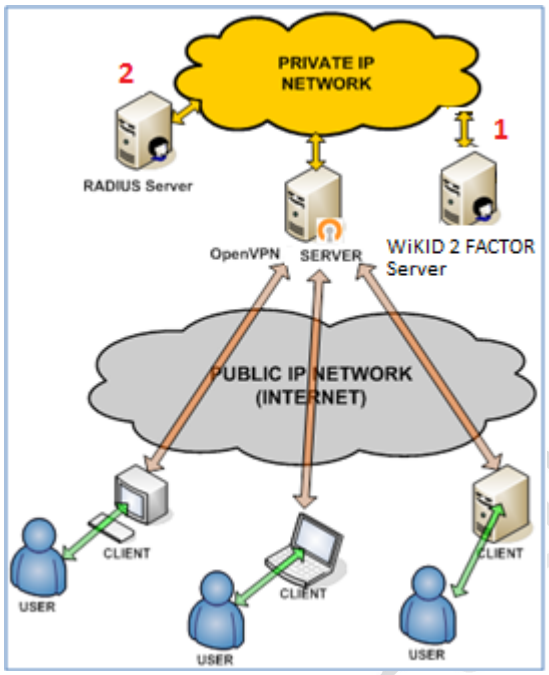
```
        persist-key
        persist-tun

# Η παρακάτω εντολή εξάγει ένα log αρχείο, που αναγράφει τις ενεργές συνδέσεις VPN κάθε ένα
#λεπτό.

        status openvpn-status.log

#Σε περίπτωση που δε χρειαζόμαστε αμοιβαία αυθεντικοποίηση με τους clients (Σε περίπτωση που
#δεν έχουν εκδοθεί ψηφιακά πιστοποιητικά για clients), αφαιρείται το σχόλιο στις δύο επόμενες
#εντολές. Στην περίπτωση της εργασίας αυτής χρησιμοποιείται αμοιβαία αυθεντικοποίηση.

        #client-cert-not-required
        #username-as-common-name

# Plugin για το πρωτόκολλο RADIUS. Στη γραμμή αυτή δηλώνεται σε ποιο σημείο βρίσκεται το plug-
#in αρχείο radiusplugin.so και radiusplugin.cnf για την ενσωμάτωση του πρωτοκόλλου RADIUS στον
#OpenVPN  server. Το plug-in μέσω PAM πρωτοκόλλου παραμένει απενεργοποιημένο.

        plugin /etc/openvpn/radiusplugin.so /etc/openvpn/radiusplugin.cnf
        ;plugin /usr/share/openvpn/plugin/lib/openvpn-auth-pam.so openvpn

# Ρυθμίζεται το επίπεδο αριθμού σχολίων στο log ως 3 που είναι για σχετικά γενική χρήση.
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose

        verb 3
########################################## Τέλος  ######################################
```

Τελειώνοντας τη διαδικασίας παραμετροποίησης του αρχείου server.conf γίνεται αποθήκευση και το αρχείο είναι έτοιμο.

4.5 Παραμετροποίηση FreeRADIUS plug-in για FreeRADIUS & OpenVPN

Για να μπορέσει ο OpenVPN server να αυθεντικοποιήσει τους χρήστες που χρησιμοποιούν τους clients, πρέπει να συνδεθεί σε κάποια βάση δεδομένων που θα περιέχει τους χρήστες. Αυτό επιτυγχάνεται είτε μέσω πρωτοκόλλου LDAP ή PAM ή RADIUS. Μπορεί να χρησιμοποιηθεί μόνα ένα πρωτόκολλο κάθε φορά. Στην εργασία αυτή γίνεται επιλογή του RADIUS πρωτοκόλλου. Αυτό επιτυγχάνεται με δύο τρόπους. Το FreeRADIUS plugin επικοινωνεί μέσω RADIUS πρωτοκόλλου αρχικά με τον WiKID Server που περιγράφηκε στο κεφάλαιο 3 και έχει σαν δευτερεύουσα επιλογή σε περίπτωση που ο WiKID Server δεν είναι διαθέσιμος, μετά από 1 δευτερόλεπτο να συνδεθεί με τον FreeRADIUS server μέσω FreeRADIUS πρωτοκόλλου που χρησιμοποιεί LDAP βάση, όπως περιγράφηκε στο κεφάλαιο 2. Αυτό έχει σαν αποτέλεσμα τη συνένωση με έμμεσο τρόπο των τριών αυτών συστημάτων.

Στην εικόνα 4-2 παρουσιάζεται ο τρόπος λειτουργίας, με συνεργασία των τριών συστημάτων, OpenVPN-WiKID-FreeRADIUS αναμένοντας να συνδεθούν οι clients στον OpenVPN server (εικ.4-2)



Εικόνα 4—2 OpenVPN Server με εξωτερικές βάσεις σε WiKID & RADIUS servers

Για να μπορέσει ο OpenVPN server να χρησιμοποιήσει τα αρχεία που έχουν οριστεί στο server.conf,

"plugin /etc/openvpn/radiusplugin.so /etc/openvpn/radiusplugin.cnf" ,πρέπει το RADIUS plug-in να εγκατασταθεί.

Αρχικά εγκαθίσταται η βιβλιοθήκη libgcrypt11-dev με την παρακάτω εντολή:

sudo apt-get install libgcrypt11-dev

Στη συνέχεια γίνεται φόρτωση του αρχείου RADIUSplugin v2.1 με την εντολή:

wget http://www.nongnu.org/radiusplugin/radiusplugin_v2.1a_beta1.tar.gz

Γίνεται αποσυμπίεση με την εντολή:

tar xvfz radiusplugin_v2.1.tar.gz

Μετακίνηση στο φάκελο που αποσυμπιέστηκε το αρχείο:

cd radiusplugin_v2.1

Και στη συνέχεια γίνεται η δημιουργία του αρχείου radiusplugin.so με την εντολή:

make

Μεταφέρονται τα 2 αρχεία του φακέλου στο σημείο που βρίσκεται ο OpenVPN server:

cp radiusplugin.so /etc/openvpn/
cp radiusplugin.cnf /etc/openvpn/

Στη συνέχεια γίνεται παραμετροποίηση του αρχείου *radiusplugin.cnf* , με την παρακάτω εντολή:

gedit /etc/openvpn/radiusplugin.cnf

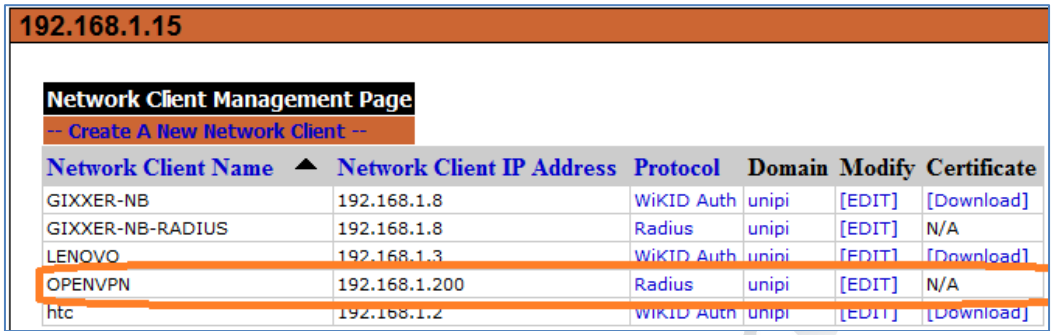
Στο αρχείο αυτό γίνεται παραμετροποίηση, ώστε να μπορεί ο OpenVPN server να συνδεθεί μέσω πρωτοκόλλου RADIUS πρώτα στον WiKID Server και στη συνέχεια στον FreeRADIUS Server.Οι πόρτες που χρησιμοποιούνται για "Authentication" είναι η 1812 και για "Accounting" η 1813.

```
OpenVPNConfig=/etc/openvpn/server.conf
overwriteccfiles=true
#client-cert-not-required (if the option is used or not) #Θέτω ως σχόλιο
#username-as-common-name (if the option is used or not) #Θέτω ως σχόλιο
# A radius server definition, there could be more than one.
# The priority of the server depends on the order in this file. The first one has the highes.
server
{
        # The UDP port for radius accounting.
        acctport=1813
        # The UDP port for radius authentication.
        authport=1812
        # The name or ip address of the radius server.
        name=192.168.1.15  ##############################FOR WiKIDServer
        # How many times should the plugin send the if there is no response?
        retry=1
        # How long should the plugin wait for a response?
        wait=1
        # Χρησιμοποιείται το shared secret που είχε οριστεί στον WiKID Server
        sharedsecret=Password2
}
server
{
        # The UDP port for radius accounting.
        acctport=1813
        # The UDP port for radius authentication.
        authport=1812
        # The name or ip address of the radius server.
        name=192.168.1.12    ############################## For RADIUS SERVER
        # How many times should the plugin send the if there is no response?
        retry=1
        # How long should the plugin wait for a response?
        wait=1
        # Χρησιμοποιείται το shared secret που είχε οριστεί στον FreeRADIUS Server
        sharedsecret=Password2
}
```

Για να μπορέσει να υπάρξει επικοινωνία του OpenVPN server με τον WiKID Server, θα πρέπει να δημιουργηθεί ένα Network Client για πρωτόκολλο RADIUS στον WiKID Server, όπως περιγράφεται στην παράγραφο παράγραφο 3.5.2.

Τα στοιχεία που πρέπει να παραμετροποιηθούν στον WiKID server είναι τα παρακάτω (εικ. 4-3):

Network Client name: *OPENVPN*
Network Client IP Address: *192.168.1.200*
Protocol: *RADIUS*
Shared Secret: *P@ssword2*



Εικόνα 4—3 Εισαγωγή OpenVPN server στο WiKID server

Επίσης θα πρέπει να ορίστεί στο αρχείο clients.conf του FreeRADIUS Server όπως περιγράφεται στην παράγραφο 2.7, η IP διεύθυνση του OpenVPN Server .

Προσθέτω στο αρχείο clients.conf μέσα την παρακάτω εγγραφή:

```
client 192.168.1.200 {
        secret          = Password2  -- Μυστικό κλειδί για την αυθεντικοποίηση μεταξύ
του OpenVPN και τουRadius Server
        shortname       = OpenVPN Server
}
```

4.6 Εκκίνηση OpenVPN server

Μετά το τέλος των ανωτέρων βημάτων ο OpenVPN server είναι έτοιμος για εκκίνηση, δίνοντας την παρακάτω εντολή για να υπάρχει ενεργή καταγραφή κατά τη διάρκεια των δοκιμών :

openvpn /etc/openvpn/ server.conf

Εναλλακτικά μπορούν να γίνει χρήση των εντολών sudo /etc/init.d/openvpn restart-stop-start,χωρίς άμεση καταγραφή συμβάντων.
Στη γραμμή εντολών εμφανίζεται το παρακάτω log όπου η τελευταία εγγραφή επιβεβαιώνει ότι ο server έχει εκκινηθεί χωρίς πρόβλημα και είναι έτοιμος να δεχθεί συνδέσεις από τους clients.

Log εκκίνησης OpenVPN server

```
Tue Sep 11 01:59:09 2012 Incoming Control Channel Authentication: Using 256 bit message hash 'SHA256' for HMAC authen
Tue Sep 11 01:59:09 2012 TLS-Auth MTU parms [ L:1570 D:178 EF:78 EB:0 ET:0 EL:0 ]
Tue Sep 11 01:59:09 2012 Socket Buffers: R=[114688->131072] S=[114688->131072]
Tue Sep 11 01:59:09 2012 ROUTE default_gateway=192.168.1.1
Tue Sep 11 01:59:09 2012 TUN/TAP device tun0 opened
Tue Sep 11 01:59:09 2012 TUN/TAP TX queue length set to 100
Tue Sep 11 01:59:09 2012 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Tue Sep 11 01:59:09 2012 /sbin/ifconfig tun0 10.8.0.1 pointopoint 10.8.0.2 mtu 1500
Tue Sep 11 01:59:09 2012 /sbin/route add -net 10.8.0.0 netmask 255.255.255.0 gw 10.8.0.2
Tue Sep 11 01:59:09 2012 Data Channel MTU parms [ L:1570 D:1450 EF:70 EB:135 ET:0 EL:0 AF:3/1 ]
Tue Sep 11 01:59:09 2012 UID set to nobody
Tue Sep 11 01:59:09 2012 UDPv4 link local (bound): [AF_INET]192.168.1.200:4596
Tue Sep 11 01:59:09 2012 UDPv4 link remote: [undef]
Tue Sep 11 01:59:09 2012 MULTI: multi_init called, r=256 v=256
```

```
Tue Sep 11 01:59:09 2012 ifconfig_pool_read(), in='user2,10.8.0.8', TODO: IPv6
Tue Sep 11 01:59:09 2012 succeeded -> ifconfig_pool_set()
Tue Sep 11 01:59:09 2012 IFCONFIG POOL LIST
Tue Sep 11 01:59:09 2012 user2,10.8.0.8
Tue Sep 11 01:59:09 2012 Initialization Sequence Completed
```

4.7 Εγκατάσταση και παραμετροποίηση του OpenVPN client

Για την εγκατάσταση του OpenVPN client σε άλλους Η/Υ που θα χρειαστεί να συνδεθούν στον server, χρησιμοποιείται το ίδιο αρχείο εγκατάστασης που χρησιμοποιείται και για τον server. Η μόνη διαφορά του client με τον server, είναι ότι η παραμετροποίηση του αρχείου server.conf είναι διαφορετική και τα ψηφιακά πιστοποιητικά του που έχουν εκδοθεί για τους clients όπως αναφέρεται στο βήμα 4.3.3.

Η εγκατάσταση του client έχει γίνει σε Η/Υ με λειτουργικό Windows 7 64-bit και το αρχείο εγκατάστασης βρίσκεται στο παρακάτω URL:

http://swupdate.openvpn.org/community/releases/openvpn-install-2.3_alpha3-I001-x86_64.exe

Αφού γίνει η εγκατάσταση της εφαρμογής OpenVPN GUI το σημείο εγκατάστασης βρίσκεται στο C:\Program Files\OpenVPN.

4.7.1 Μεταφορά των ψηφιακών πιστοποιητικών

Γίνεται αντιγραφή των τεσσάρων αρχείων *ca.crt LAPTOP.key LAPTOP.csr ta.key* , από το παρακάτω σημείο του OpenVPN server:

/usr/share/doc/openvpn/examples/easy-rsa/2.0/keys/

στο παρακάτω σημείο που είναι εγκατεστημένος ο client:

C:\Program Files\OpenVPN\config\

4.7.2 Μεταφορά αρχείου παραμετροποίησης του OpenVPN Client

Για να μπορέσει να λειτουργήσει ο OpenVPN client όπως και ο server, χρειάζεται το βασικό αρχείο client.ovpn, που μέσα περιέχει τις βασικές λειτουργίες παραμετροποίησης του client, όπως και τις εντολές που καλεί τα ψηφιακά πιστοποιητικά που χρειάζεται κάθε φορά.

Γίνεται μεταφορά του αρχείου :

C:\Program Files\OpenVPN\sample-config\client.ovpn

στο σημείο:

C:\Program Files\OpenVPN\config\

4.7.3 Παραμετροποίηση αρχείου OpenVPN client

Ανοίγουμε τα αρχείο C:\ProgramFiles\OpenVPN\config\client.ovpn και ξεκινάει η παραμετροποίηση του client.opvn:

```
##############################################################################
# Αυτό το αρχείο παραμετροποίησης μπορεί να  χρησιμοποιηθεί από πολλαπλούς clients,αλλά    #
#οκάθε client πρέπει να έχει το δικό το πιστοποιητικό και κλειδί                          #
##############################################################################
#Γίνεται καθορισμός ότι το αρχείο χρησιμοποιείται ως client και θα δέχεται εντολές από το server

        client

# Χρησιμοποιείται η ίδια ρύθμιση με του server dev.tun
# Πρέπει να γίνει απενεργοποίηση του firewall στην κάρτα TUN στο λειτουργικό σύστημα
        dev tun

# Χρήση UDP πόρτας. Επιλέγεται  η UDP πόρτα όπως και στο server

        ;proto tcp
        proto udp

# Ορίζεται το  hostname/IP  και η πόρτα του server.
# Μπορούν να δηλωθούν πολλαπλοί servers για load-balance.

        remote 192.168.1.200 4596

# Δηλώνουμε το όνομα του που αναγράφεται στο common name του πιστοποιητικού του server για
#την ταυτοποίηση του. Γίνεται απόρριψη σε περίπτωση διαφορετικού ονόματος του SERVER.

        tls-remote SERVER

#Διατηρεί αόριστα την host name  του  OpenVPN server.  Πολύ χρήσιμο σε μηχανές που δεν είναι
#μόνιμα συνδεδεμένες στο Internet όπως  οι φορητοί Η/Υ.

        resolv-retry infinite

# Οι περισσότεροι clients δε χρειάζεται να δεσμεύσουν μια συγκεκριμένη τοπική πόρτα,
#χρησιμοποιείται η παρακάτω εντολή

        nobind

# Διατηρεί την κατάσταση ελαχιστοποίησης δικαιωμάτων πρόσβασης, σε επανεκκινήσεις

        persist-key
        persist-tun

#Δηλώνονται το SSL/TLS root certificate (ca.crt), το client certificate (server.crt ) και το ιδιωτικό
#κλειδί (server.key). Ο server και όλοι οι clients πρέπει να χρησιμοποιούν το ίδιο ca #αρχείο.

        ca ca.crt
        cert laptop.crt
        key laptop.key

#Η εντολή  auth-user-pass χρησιμοποιείται για να εμφανιστεί το πλαίσιο με username/password
        auth-user-pass
```

 # Δηλώνουμε το όνομα του που αναγράφεται στο common name του πιστοποιητικού του server

```
ns-cert-type SERVER
```

#Το shared key κρυπτογράφησης/αποκρυπτογράφησης ta.key πρέπει να δηλωθεί εδώ με τιμή 1, #υποδηλώνοντας ότι είναι client.

```
tls-auth ta.key 1
```

#Επιλέγεται ο κρυπτογραφικός αλγορίθμος AES-256-CBC. Η ίδια παραμετροποίηση έχει γίνει και #στο αρχείο του server

```
cipher AES-256-CBC   # AES
```

 # Επιλέγεται ο αλγόριθμος Hash RSA-SHA256. Η ίδια παραμετροποίηση έχει γίνει και στο αρχείο #του server.

```
auth RSA-SHA256
```

 # Ενεργοποιείται συμπίεση τύπου LZO, στην σύνδεση VPN.Έχει ενεργοποιηθεί και στον server

```
comp-lzo
```

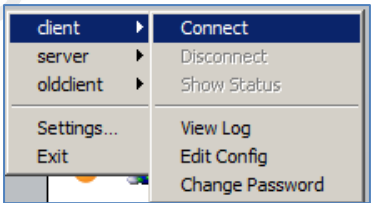
 # Ρυθμίζεται το επίπεδο αριθμού σχολίων στο log ως 3 που είναι για σχετικά γενική χρήση.

```
verb 3
```

Το αρχείο αποθηκεύεται και ο client είναι έτοιμος προς εκκίνηση.

4.8 Εκκίνηση OpenVPN client

Έχοντας εκκινήσει τους OpenVPN server, WiKIDServer και FreeRADIUS server (προαιρετικά μπορούν να ανοίξουν και οι δύο) και έχοντας εκκινήσει την εφαρμογή OpenVPN Client GUI για Windows, για να επιτευχθεί σύνδεση στο server επιλέγουμε ***client → Connect*** (Εικ. 4-4)



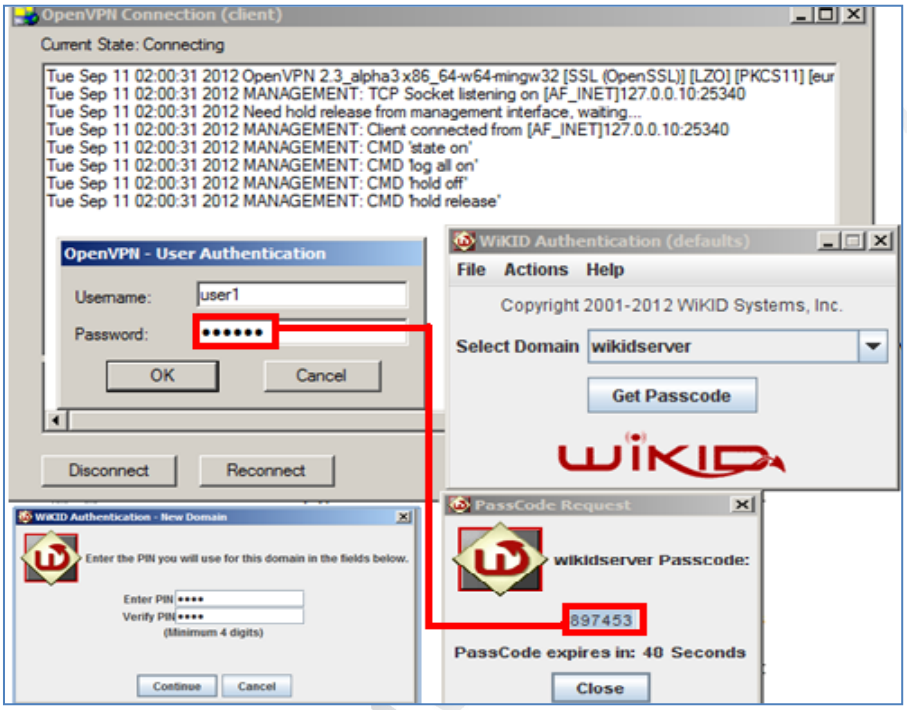
Εικόνα 4—4 OpenVPN Client GUI Windows

4.8.1 Αυθεντικοποίηση Client σε OpenVPN Server και χρήστη σε WiKID Server

Έχοντας ανοίξει το WiKID Client από τον Η/Υ που έχει συσχετιστεί με τον user1 στον WiKID server και πληκτρολογώντας επιτυχώς τον πρώτο παράγοντα αυθεντικοποίησης PIN στο WiKID Client για το Domain που έχει δημιουργηθεί (αναφορά για WiKID client στην παρ. 3.6.2), παράγεται το Passcode στο WiKID client, για να αντιγραφεί στο πεδίο password του OpenVPN client, αφού έχει εισαχθεί και το αντίστοιχο username user1. Πιέζοντας το πλήκτρο OK μέσα σε 60 δευτερόλεπτα από τη δημιουργία του OTP, γίνεται επιτυχής αυθεντικοποίηση στον WiKID Server και στη συνέχεια ο WiKID Server επιστρέφει μήνυμα επιτυχούς πρόσβασης προς τον OpenVPN server:

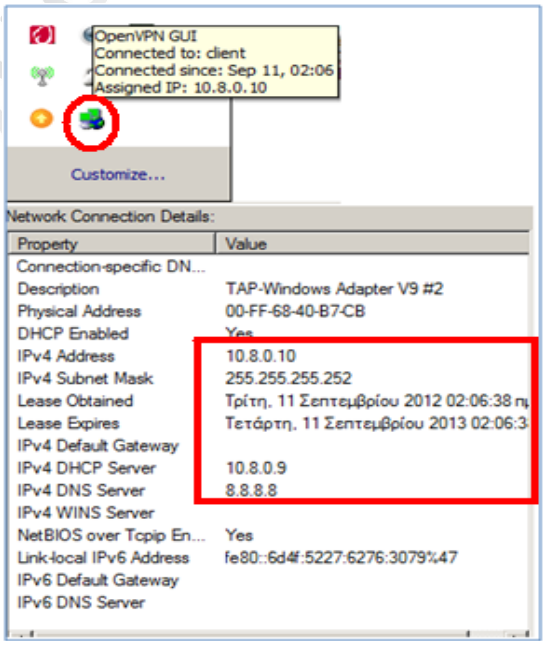
```
RADIUS-PLUGIN: BACKGROUND AUTH: Reply-Message:Access Granted
```

Η διαδικασία αυτή επιτυγχάνεται εφόσον γίνει αμοιβαία αυθεντικοποίηση του OpenVPN client στον OpenVPN server και αντίστροφα. Ουσιαστικά η αυθεντικοποίηση είναι συνδυασμός τριών παραγόντων, όπου οι δύο παράγοντες είναι από το WiKID (PIN και Username/Passcode των 60 δευτ.) για το χρήστη και ο τρίτος παράγοντας από το ψηφιακό πιστοποιητικό του OpenVPN client για τον client.



Εικόνα 4—5 Αυθεντικοποίηση στο OpenVPN μέσω WiKID server

Η επιτυχής σύνδεση του OpenVPN client στον OpenVPN server, αποδίδει μέσω του DHCP server του OpenVPN server στην εικονική κάρτα δικτύου VPN (TAP-Adapter) τα παρακάτω χαρακτηριστικά:IP διεύθυνση 10.8.0.10,DNS Server 8.8.8.8,Lease Time. Το γραφικό GUI client (Windows) αλλάζει χρώμα από κόκκινο σε πράσινο.



Εικόνα 4—6 Επιτυχής Σύνδεση OpenVPN client μέσω WiKID

Το Log του OpenVPN client κατά την επιτυχή σύνδεση παρουσιάζεται παρακάτω:

Log OpenVPN Client

```
Tue Sep 11 02:00:31 2012 OpenVPN 2.3_alpha3 x86_64-w64-mingw32 [SSL (OpenSSL)] [LZO] [PKCS11] [eurephia]
[PF_INET6] [IPv6 payload 20110522-1 (2.2.0)] built on Jul 24 2012
Tue Sep 11 02:00:31 2012 MANAGEMENT: TCP Socket listening on [AF_INET]127.0.0.10:25340
Tue Sep 11 02:00:31 2012 Need hold release from management interface, waiting...
Tue Sep 11 02:00:31 2012 MANAGEMENT: Client connected from [AF_INET]127.0.0.10:25340
Tue Sep 11 02:00:31 2012 MANAGEMENT: CMD 'state on'
Tue Sep 11 02:00:31 2012 MANAGEMENT: CMD 'log all on'
Tue Sep 11 02:00:31 2012 MANAGEMENT: CMD 'hold off'
Tue Sep 11 02:00:31 2012 MANAGEMENT: CMD 'hold release'
Tue Sep 11 02:01:32 2012 MANAGEMENT: CMD 'username "Auth" "user1"'
Tue Sep 11 02:01:32 2012 MANAGEMENT: CMD 'password [...]'
Tue Sep 11 02:01:32 2012 WARNING: Make sure you understand the semantics of --tls-remote before using it (see the man
page).
Tue Sep 11 02:01:32 2012 NOTE: OpenVPN 2.1 requires '--script-security 2' or higher to call user-defined scripts or executables
Tue Sep 11 02:01:33 2012 Control Channel Authentication: using 'ta.key' as a OpenVPN static key file
Tue Sep 11 02:01:33 2012 Outgoing Control Channel Authentication: Using 256 bit message hash 'SHA256' for HMAC
authentication
Tue Sep 11 02:01:33 2012 Incoming Control Channel Authentication: Using 256 bit message hash 'SHA256' for HMAC
authentication
Tue Sep 11 02:01:33 2012 Socket Buffers: R=[8192->8192] S=[8192->8192]
Tue Sep 11 02:01:33 2012 UDPv4 link local: [undef]
Tue Sep 11 02:01:33 2012 UDPv4 link remote: [AF_INET]192.168.1.200:4596
Tue Sep 11 02:01:33 2012 MANAGEMENT: >STATE:1347318093,WAIT,,,
Tue Sep 11 02:01:33 2012 MANAGEMENT: >STATE:1347318093,AUTH,,,
Tue Sep 11 02:01:33 2012 TLS: Initial packet from [AF_INET]192.168.1.200:4596, sid=2c807672 1643262b
Tue Sep 11 02:01:33 2012 WARNING: this configuration may cache passwords in memory -- use the auth-nocache option to
prevent this
Tue Sep 11 02:01:33 2012 VERIFY OK: depth=1, C=GR, ST=ATTIKI, L=PIRAEUS, O=UNIVERSITY OF PIRAEUS,
OU=DIGITAL SYSTEMS, CN=OpenVPN-CA, emailAddress=me@myhost.mydomain
Tue Sep 11 02:01:33 2012 VERIFY OK: nsCertType=SERVER
Tue Sep 11 02:01:33 2012 Validating certificate key usage
Tue Sep 11 02:01:33 2012 ++ Certificate has key usage  00a0, expects 00a0
Tue Sep 11 02:01:33 2012 VERIFY KU OK
Tue Sep 11 02:01:33 2012 Validating certificate extended key usage
Tue Sep 11 02:01:33 2012 ++ Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server
Authentication
Tue Sep 11 02:01:33 2012 VERIFY EKU OK
Tue Sep 11 02:01:33 2012 VERIFY X509NAME OK: C=GR, ST=ATTIKI, L=PIRAEUS, O=UNIVERSITY OF PIRAEUS,
OU=DIGITAL SYSTEMS, CN=SERVER, emailAddress=me@myhost.mydomain
Tue Sep 11 02:01:33 2012 VERIFY OK: depth=0, C=GR, ST=ATTIKI, L=PIRAEUS, O=UNIVERSITY OF PIRAEUS,
OU=DIGITAL SYSTEMS, CN=SERVER, emailAddress=me@myhost.mydomain
Tue Sep 11 02:01:33 2012 Data Channel Encrypt: Cipher 'AES-256-CBC' initialized with 256 bit key
Tue Sep 11 02:01:33 2012 Data Channel Encrypt: Using 256 bit message hash 'SHA256' for HMAC authentication
Tue Sep 11 02:01:33 2012 Data Channel Decrypt: Cipher 'AES-256-CBC' initialized with 256 bit key
Tue Sep 11 02:01:33 2012 Data Channel Decrypt: Using 256 bit message hash 'SHA256' for HMAC authentication
Tue Sep 11 02:01:33 2012 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 1024 bit RSA
Tue Sep 11 02:01:33 2012 [SERVER] Peer Connection Initiated with [AF_INET]192.168.1.200:4596
Tue Sep 11 02:01:34 2012 MANAGEMENT: >STATE:1347318094,GET_CONFIG,,,
Tue Sep 11 02:01:35 2012 SENT CONTROL [SERVER]: 'PUSH_REQUEST' (status=1)
Tue Sep 11 02:01:35 2012 PUSH: Received control message: 'PUSH_REPLY,dhcp-option DNS 8.8.8.8,route 10.8.0.1,topology
net30,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5'
Tue Sep 11 02:01:35 2012 OPTIONS IMPORT: timers and/or timeouts modified
Tue Sep 11 02:01:35 2012 OPTIONS IMPORT: --ifconfig/up options modified
Tue Sep 11 02:01:35 2012 OPTIONS IMPORT: route options modified
Tue Sep 11 02:01:35 2012 OPTIONS IMPORT: --ip-win32 and/or --dhcp-option options modified
Tue Sep 11 02:01:35 2012 do_ifconfig, tt->ipv6=0, tt->did_ifconfig_ipv6_setup=0
Tue Sep 11 02:01:35 2012 MANAGEMENT: >STATE:1347318095,ASSIGN_IP,,10.8.0.6,
Tue Sep 11 02:01:35 2012 open_tun, tt->ipv6=0
Tue Sep 11 02:01:35 2012 TAP-WIN32 device [Local Area Connection 7] opened: \\.\Global\{6840B7CB-CD84-4064-B7E5-
60ABE93D6D21}.tap
Tue Sep 11 02:01:35 2012 TAP-Windows Driver Version 9.9
Tue Sep 11 02:01:35 2012 Notified TAP-Windows driver to set a DHCP IP/netmask of 10.8.0.6/255.255.255.252 on interface
{6840B7CB-CD84-4064-B7E5-60ABE93D6D21} [DHCP-serv: 10.8.0.5, lease-time: 31536000]
Tue Sep 11 02:01:35 2012 Successful ARP Flush on interface [47] {6840B7CB-CD84-4064-B7E5-60ABE93D6D21}
Tue Sep 11 02:01:40 2012 TEST ROUTES: 1/1 succeeded len=1 ret=1 a=0 u/d=up
Tue Sep 11 02:01:40 2012 MANAGEMENT: >STATE:1347318100,ADD_ROUTES,,,
Tue Sep 11 02:01:40 2012 C:\Windows\system32\route.exe ADD 10.8.0.1 MASK 255.255.255.255 10.8.0.5
Tue Sep 11 02:01:40 2012 ROUTE: CreateIpForwardEntry succeeded with dwForwardMetric1=30 and dwForwardType=4
Tue Sep 11 02:01:40 2012 Route addition via IPAPI succeeded [adaptive]
Tue Sep 11 02:01:40 2012 Initialization Sequence Completed
```

```
Tue Sep 11 02:01:40 2012 MANAGEMENT: >STATE:1347318100,CONNECTED,SUCCESS,10.8.0.6,192.168.1.200
```

Στο Log του client προτελευταία εγγραφή δηλώνει ότι η σύνδεση VPN έχει επιτευχθεί με το server.

```
Initialization Sequence Completed
```

Το Log του OpenVPN server σε επιτυχή σύνδεση με το OpenVPN client και αυθεντικοποίηση του user1 στον WiKID server παρουσιάζεται παρακάτω:

Log OpenVPN server σε επιτυχή σύνδεση με τον WIKID server

```
Tue Sep 11 02:01:32 2012 MULTI: multi_create_instance called
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 Re-using SSL/TLS context
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 LZO compression initialized
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 Control Channel MTU parms [ L:1570 D:178 EF:78 EB:0 ET:0 EL:0 ]
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 Data Channel MTU parms [ L:1570 D:1450 EF:70 EB:135 ET:0 EL:0 AF:3/1 ]
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 Local Options hash (VER=V4): '8a3b3cca'
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 Expected Remote Options hash (VER=V4): '73e43c96'
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 TLS: Initial packet from [AF_INET]192.168.1.8:51510, sid=b584d91c eca66d19
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 VERIFY OK: depth=1,
/C=GR/ST=ATTIKI/L=PIRAEUS/O=UNIVERSITY_OF_PIRAEUS/OU=DIGITAL_SYSTEMS/CN=OpenVPN-
CA/emailAddress=me@myhost.mydomain
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 VERIFY OK: depth=0,
/C=GR/ST=ATTIKI/L=PIRAEUS/O=UNIVERSITY_OF_PIRAEUS/OU=DIGITAL_SYSTEMS/CN=LAPTOP/emailAddress=me@
myhost.mydomain
Tue Sep 11 02:01:32 2012 RADIUS-PLUGIN: FOREGROUND THREAD: Auth_user_pass_verify thread started.
Tue Sep 11 02:01:32 2012 RADIUS-PLUGIN: FOREGROUND THREAD: New user.
Tue Sep 11 02:01:32 2012 RADIUS-PLUGIN: No attributes Acct Interim Interval or bad length.
Tue Sep 11 02:01:32 2012 RADIUS-PLUGIN: BACKGROUND AUTH: Reply-Message:Access Granted
Tue Sep 11 02:01:32 2012 RADIUS-PLUGIN: Client config file was not written, overwriteccfiles is false
.Tue Sep 11 02:01:32 2012 RADIUS-PLUGIN: FOREGROUND THREAD: Add user to map.
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 PLUGIN_CALL: POST
/etc/openvpn/radiusplugin.so/PLUGIN_AUTH_USER_PASS_VERIFY status=0
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 TLS: Username/Password authentication succeeded for username 'user1'
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 Data Channel Encrypt: Cipher 'AES-256-CBC' initialized with 256 bit key
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 Data Channel Encrypt: Using 256 bit message hash 'SHA256' for HMAC
authentication
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 Data Channel Decrypt: Cipher 'AES-256-CBC' initialized with 256 bit key
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 Data Channel Decrypt: Using 256 bit message hash 'SHA256' for HMAC
authentication
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 Control Channel: TLSv1, cipher TLSv1/SSLv3 DHE-RSA-AES256-SHA, 1024 bit
RSA
Tue Sep 11 02:01:32 2012 192.168.1.8:51510 [LAPTOP] Peer Connection Initiated with [AF_INET]192.168.1.8:51510
Tue Sep 11 02:01:32 2012 LAPTOP/192.168.1.8:51510 MULTI_sva: pool returned IPv4=10.8.0.6,
IPv6=dc62:1809:107e:1609:500:0:8a5:e0bf
Tue Sep 11 02:01:32 2012 LAPTOP/192.168.1.8:51510 PLUGIN_CALL: POST
/etc/openvpn/radiusplugin.so/PLUGIN_CLIENT_CONNECT status=0
Tue Sep 11 02:01:32 2012 LAPTOP/192.168.1.8:51510 OPTIONS IMPORT: reading client specific options from:
/tmp/openvpn_cc_02da53ba18e82b17a41563e240daf97a.tmp
Tue Sep 11 02:01:32 2012 LAPTOP/192.168.1.8:51510 MULTI: Learn: 10.8.0.6 -> LAPTOP/192.168.1.8:51510
Tue Sep 11 02:01:32 2012 LAPTOP/192.168.1.8:51510 MULTI: primary virtual IP for LAPTOP/192.168.1.8:51510: 10.8.0.6
Tue Sep 11 02:01:34 2012 LAPTOP/192.168.1.8:51510 PUSH: Received control message: 'PUSH_REQUEST'
Tue Sep 11 02:01:34 2012 LAPTOP/192.168.1.8:51510 send_push_reply(): safe_cap=960
Tue Sep 11 02:01:34 2012 LAPTOP/192.168.1.8:51510 SENT CONTROL [LAPTOP]: 'PUSH_REPLY,dhcp-option DNS
8.8.8.8,route 10.8.0.1,topology net30,ping 10,ping-restart 120,ifconfig 10.8.0.6 10.8.0.5' (status=1)
```

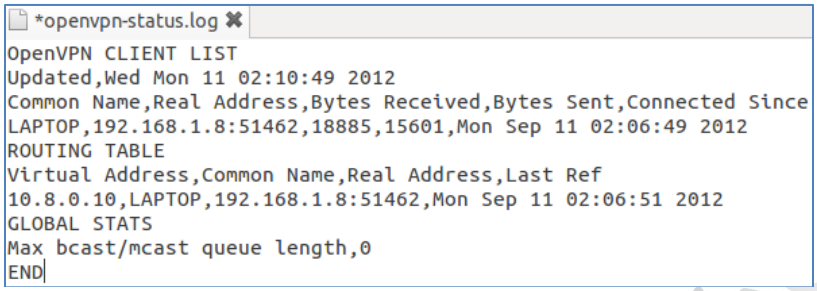
Αν από το σταθμό του Cient που έχει λάβει εικονική IP 10.8.0.6 στην εικονική του κάρτα δικτύου VPN, γίνει ping στην εικονική IP 10.8.0.1 που αντιστοιχεί στην εικονική κάρτα VPN δικτύου του OpenVPN server, τότε θα υπάρχει θετική ανταπόκριση.

Στο αρχείο status.log του server καταγράφεται η λίστα με τις τρέχουσες συνδέσεις των clients που είναι συνδεδεμένοι πάνω στον server και ανανεώνεται κάθε 1 λεπτό.

Πληκτρολογώντας την εντολή:

```
gedit /etc/openvpn/openvpn-status.log
```

Λαμβάνουμε την εγγραφή με την σύνδεση του OpenVPN client με τα στοιχεία σύνδεσής του .



Εικόνα 4—7 OpenVPN Status Log

Αφού έχουν διασφαλιστεί όλοι οι παράμετροι αυθεντικοποίησης, η VPN σύνδεση μεταξύ server-client μέσω του tunnel ,είναι απολύτως ασφαλής για λειτουργία εικονικού ιδιωτικού δικτύου και η κρυπτογράφηση/αποκρυπτογράφηση γίνεται με το προ-διαμοιραζόμενο κλειδί "ta.key".

4.8.2 Αυθεντικοποίηση Client σε OpenVPN Server και χρήστη σε FreeRADIUS Server

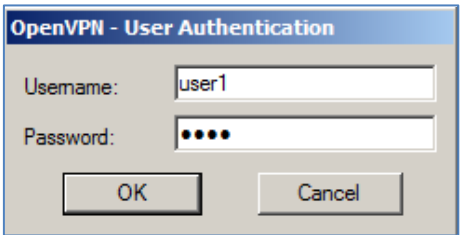
Σε περίπτωση που ο WiKID Server δεν είναι διαθέσιμος ή κλείσει, Ο OpenVPN server μέσω του αρχείου */etc/openvpn/radiusplugin.cnf* επιλέγει ως δεύτερη επιλογή τον server FreeRADIUS αυθεντικο-ποίησης χρηστών και πάλι μέσω του πρωτοκόλλου RADIUS . Εμφανίζεται και πάλι το πλαίσιο με τα δύο πεδία username και password του OpenVPN client. Στα δύο αυτά πεδία ο χρήστης για να μπορέσει να αυθεντικοποιηθεί, θα πρέπει να χρησιμοποιήσει το ίδιο username αλλά ως password αυτό που έχει οριστεί στη βάση δεδομένων LDAP του FreeRADIUS server όπως φαίνεται στην εικόνα 2-10 στο κεφάλαιο 2. Αυτό σημαίνει ότι ο χρήστης θα πρέπει να γνωρίζει το password που αντιστοιχεί στο χρήστη του.

Τα στοιχεία αυθεντικοποίησης μπορούν να είναι οποιαδήποτε εγγραφή χρήστη περιέχεται στον LDAP server για εξουσιοδότηση.

Πιέζοντας το πλήκτρο OK (εικ.4-8) γίνεται επιτυχής αυθεντικοποίηση στον FreeRADIUS server και στη συνέχεια ο FreeRADIUS Server επιστρέφει μήνυμα επιτυχούς πρόσβασης προς τον OpenVPN server:

```
RADIUS-PLUGIN: BACKGROUND AUTH: Reply-Message:Access Granted
```

Η διαδικασία αυτή επιτυγχάνεται εφόσον γίνει αμοιβαία αυθεντικοποίηση του OpenVPN client στον OpenVPN server και αντίστροφα. Ουσιαστικά η αυθεντικοποίηση είναι δύο παραγόντων, όπου ο ένας παράγοντας είναι το username/password και ο δεύτερος παράγοντας το ψηφιακό πιστοποιητικό του OpenVPN client.

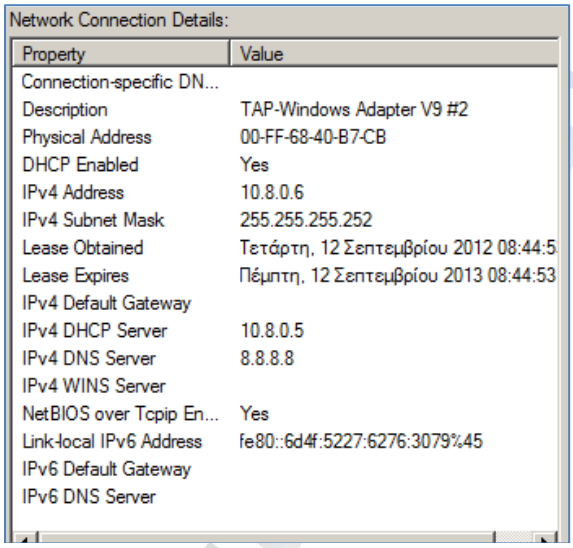


Εικόνα 4—8 Αυθεντικοποίηση στο OpenVPN μέσω FreeRADIUS server

Τα logs του Client και του Server είναι ακριβώς τα ίδια με της σύνδεσης με τον WiKID server για αυτό και δε θα αναφερθούν εκτενώς. Παρατηρούμε απλώς τις δύο τελευταίες εγγραφές του client log, όπου αναφέρουν επιτυχή σύνδεση.

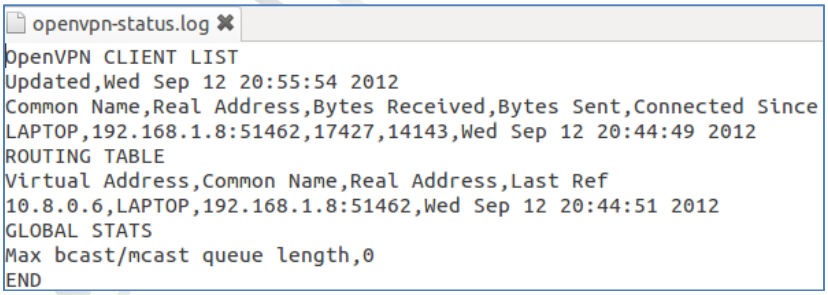
```
Wed Sep 12 20:44:58 2012 Initialization Sequence Completed
Wed Sep 12 20:44:58 2012 MANAGEMENT: >STATE:1347471898,CONNECTED,SUCCESS,10.8.0.6,192.168.1.200
```

Η επιτυχής σύνδεση του OpenVPN client στον OpenVPN server, αποδίδει μέσω του DHCP server του OpenVPN server στην εικονική κάρτα δικτύου VPN (TAP-Adapter) τα παρακάτω χαρακτηριστικά:IP διεύθυνση 10.8.0.6,DNS Server 8.8.8.8,Lease Time.



Εικόνα 4—9 Επιτυχής Σύνδεση OpenVPN client μέσω FreeRADIUS

Λαμβάνουμε την εγγραφή με την σύνδεση του OpenVPN client με τα στοιχεία σύνδεσής του



Εικόνα 4—10 OpenVPN Status Log

Παρατηρούμε ότι σε σχέση με την προηγούμενη σύνδεση, ο DHCP server έχει δώσει διαφορετική IP στον εαυτό του, όπως και στον client.

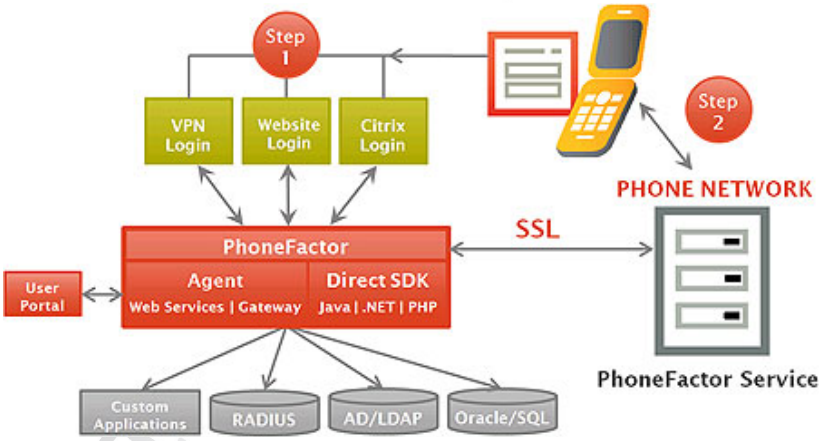
5 Εγκατάσταση και Παραμετροποίηση PhoneFACTOR

5.1 Εισαγωγή

Το PhoneFACTOR είναι ένα σύστημα αυθεντικοποίησης πολλαπλών παραγόντων για τη διασφάλιση της εισόδου των λογαριασμών των χρηστών σε διάφορες υπηρεσίες, χρησιμοποιώντας αυτοματοποιημένες τηλεφωνικές κλήσεις για την εξακρίβωση της ταυτότητας. Με την επιλογή πολλαπλών μεθόδων (φωνητική κλήση, SMS μήνυμα τηλεφώνου και την εφαρμογή Push για Smartphones) και μίας μεθόδου OATH για One-Time-Password, το PhoneFACTOR παρέχει ευελιξία στους χρήστες για την επιλογή μεθόδου αυθεντικοποίησης τους. Δεδομένου ότι ο κάθε χρήστης κατέχει πλέον μια τηλεφωνική συσκευή, το PhoneFACTOR μπορεί να επιτρέψει τη γρήγορη επέκταση της αυθεντικοποίησης δύο παραγόντων σε διάφορες εφαρμογές που την απαιτούν.

Αυτή η διαδικασία παρέχει αυθεντικοποίηση δύο βημάτων, μέσω δύο ξεχωριστών καναλιών (Η/Υ και τηλέφωνο). Αυτά είναι κάτι που ο χρήστης γνωρίζει, τον κωδικό του και κάτι που ο χρήστης έχει, το τηλέφωνο του. Το PhoneFACTOR παρέχει και τρίτο παράγοντας αυθεντικοποίησης μέσω βιομετρικών χαρακτηριστικών, δηλαδή κάτι που ο χρήστης είναι, δηλαδή η φωνή του.

Στην εικόνα 5-1 παρουσιάζεται η αρχιτεκτονική αυθεντικοποίησης δύο παραγόντων σε δύο βήματα, μέσω του PhoneFACTOR σε διάφορες εφαρμογές.



Σχήμα 5-1 Αρχιτεκτονική του PhoneFACTOR

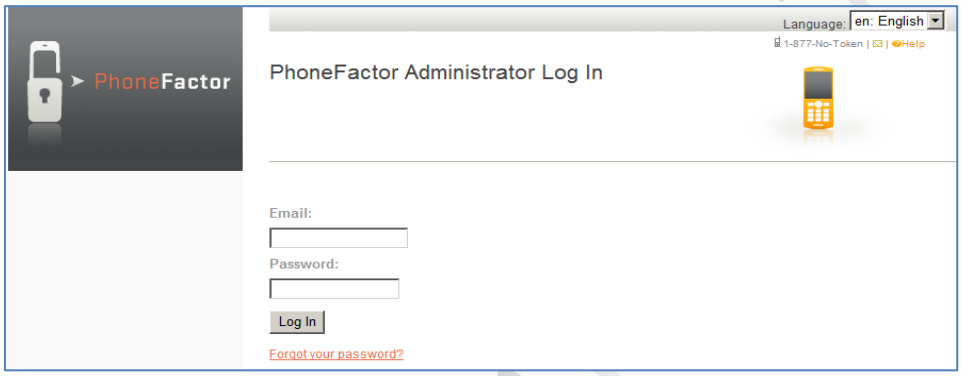
Η αρχιτεκτονική λειτουργίας του είναι η εξής: Μόλις ο χρήστης αυθεντικοποιηθεί επιτυχώς με τον πρώτο παράγοντα username/password σε μία εφαρμογή π.χ. VPN Login, που αυτή επικοινωνεί με εφαρμογή agent του PhoneFACTOR που είναι εγκατεστημένος σε ένα Η/Υ του διαχειριστή των χρηστών, τότε αυτός επικοινωνεί μέσω SSL με το κέντρο δεδομένων της εταιρείας PhoneFACTOR, και στη συνέχεια ο χρήστης δέχεται αυτόματη τηλεφωνική κλήση από server της εταιρείας όπου του ζητά το δεύτερο παράγοντα, που είναι το PIN (Personal Identification Number) του λογαριασμού χρήστη στο PhoneFACTOR και τον τρίτο παράγοντα που είναι η φωνή του χρήστη. Μετά την επιτυχή αυθεντικοποίηση και των άλλων δύο παραγόντων, ο server της PhoneFACTOR επικοινωνεί με τον agent PhoneFACTOR και ο agent επιτρέπει την πρόσβαση του χρήστη στην εφαρμογή VPN.

Εναλλακτικά από τον Agent(μόνο για Windows), μπορεί να χρησιμοποιηθεί το Direct SDK, όπου ενσωματώνεται σε διαδικτυακές εφαρμογές ως plug-in για .NET, Java, PHP, Rubu και Perl ή PhoneFACTOR Agent για την υπηρεσία LogMeIn.

5.2 Διαχείριση Λογαριασμών στο PhoneFACTOR

Για τη δημιουργία λογαριασμών της κεντρικής διαχειριστικής κονσόλας του PhoneFACTOR, όπως και τη λήψη του Phonefactor Agent για μετέπειτα εγκατάσταση σε Η/Υ, θα πρέπει να γίνει εγγραφή του διαχειριστή στο URL https://pfweb.phonefactor.com/register/ σε 4 βήματα *(1.Sign Up- 2.Authenticate- 3.Security Questions- 4.Download)*

Στη συνέχεια μεταβαίνοντας στο URL https://pfweb.phonefactor.com/ γίνεται αυθεντικοποίηση με τα στοιχεία εγγραφής (εικ. 5-1).

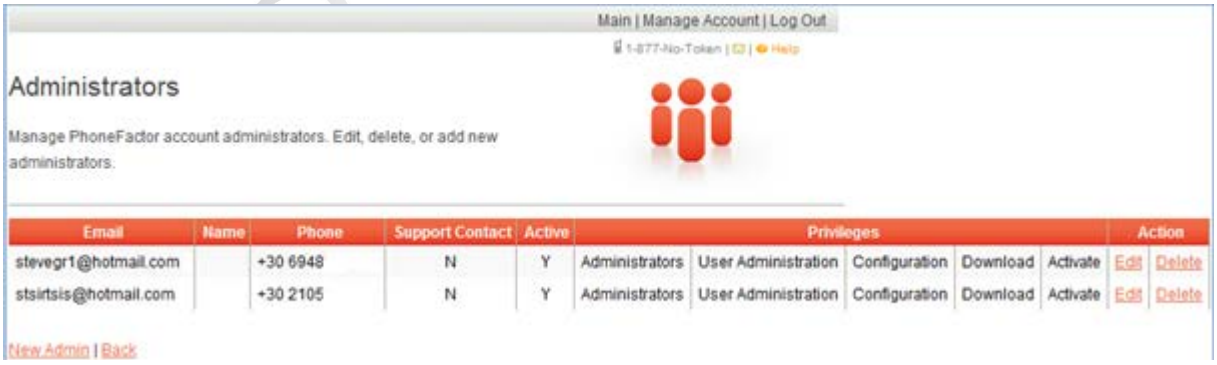


Εικόνα 5—1 Πρόσβαση Διαχείρισης PhoneFACTOR

Η υπηρεσία τηλεφωνικών κλήσεων και SMS μηνυμάτων αυθεντικοποίησης που αποστέλλονται από το κέντρο δεδομένων της PhoneFACTOR, παρέχεται στις Ευρωπαικές χώρες δωρεάν σε σταθερό τηλέφωνο και σε κινητό τηλέφωνο με 0.10$/κλήση ή 0,05$/SMS. Ο πίνακας τιμών ανά χώρα παρέχεται στο URL https://pfweb.phonefactor.com/reports/global_rates_report. Στην πρώτη εγγραφή ενός χρήστη πιστώνεται ένα ποσό αξίας 5$, για δοκιμαστικό σκοπό. Στις επόμενες τρεις παραγράφους, περιγράφονται οι βασικότερες λειτουργίες του PhoneFACTOR Management Portal.

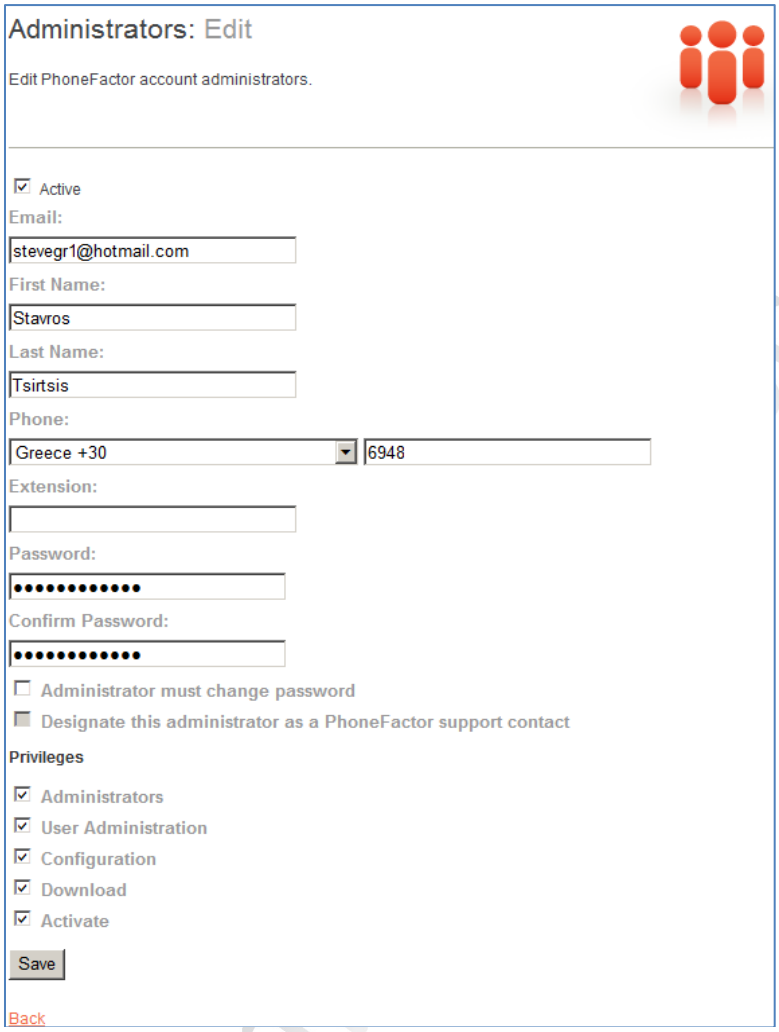
5.2.1 Παραμετροποίηση Διαχειριστών

Για την προσθήκη πάνω από έναν διαχειριστές από το κεντρικό μενού, επιλέγεται το πλήκτρο *Administrators* και εμφανίζεται η λίστα των διαχειριστών του PhoneFACTOR, με τα στοιχεία αυθεντικοποίησης τους και τα δικαιώματα του κάθε διαχειριστή (εικ 5-2).



Εικόνα 5—2 Λίστα Διαχειριστών

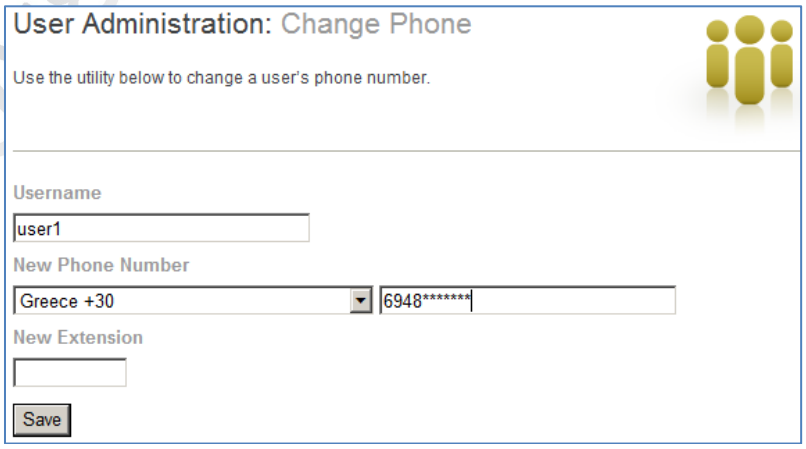
Επιλέγοντας στην ίδια σελίδα το πλήκτρο *NewAdmin* δημιουργείται νέος διαχειριστής με όλα τα στοιχεία που απαιτούνται για την αυθεντικοποίηση του, και όλα τα δικαιώματα διαχείρισης, όπως φαίνεται στην εικόνα 5-3.



Εικόνα 5—3 Δημιουργία Διαχειριστή

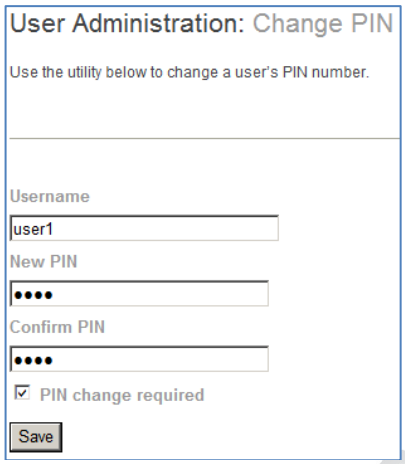
5.2.2 Διαχείριση Χρηστών PhoneFACTOR

Η διαχείριση χρηστών προϋποθέτει αρχικά την δημιουργία τους από το PhoneFACTOR Agent που θα αναλυθεί αργότερα. Από το κεντρικό μενού επιλέγοντας *User Administration → Change Phone,* εισάγοντας το Username που πρέπει να γνωρίζουμε, γίνεται η αλλαγή του τηλεφωνικού αριθμού (Εικ. 5-4).



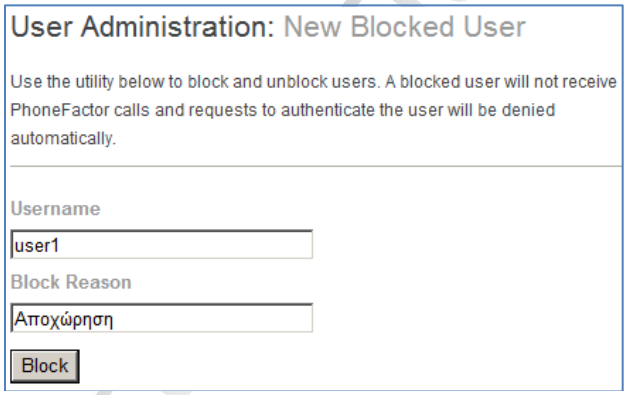
Εικόνα 5—4 Αλλαγή τηλεφωνικού αριθμού χρήστη

Από το κεντρικό μενού επιλέγοντας *User Administration → Change PIN,* εισάγοντας το Username, γίνεται η αλλαγή του PIN. (Εικ. 5-5)



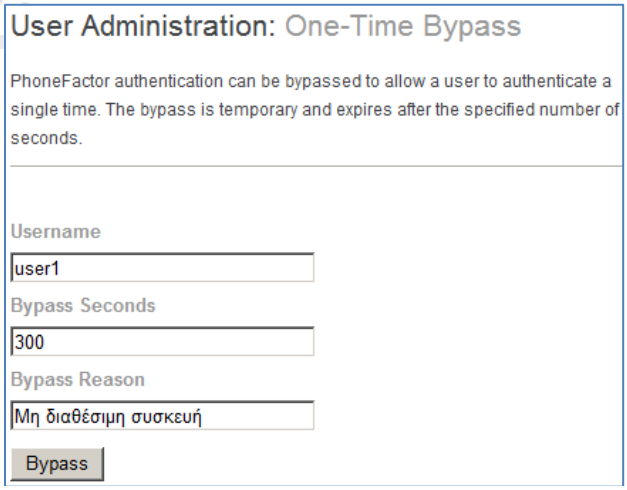
Εικόνα 5—5 Αλλαγή PIN χρήστη

Από το κεντρικό μενού επιλέγοντας *User Administration → Block/Unblock Users,* εισάγοντας το Username, γίνεται ο αποκλεισμός ενός χρήστη (Εικ. 5-6).



Εικόνα 5—6 Αποκλεισμός χρήστη

Από το κεντρικό μενού επιλέγοντας *User Administration → One-Time Bypass,* εισάγοντας το username και τον επιθυμητό χρόνο, γίνεται για μία φόρα μόνο αυθεντικοποίηση του χρήστη χωρίς χρήση του PhoneFACTOR (Εικ. 5-7).

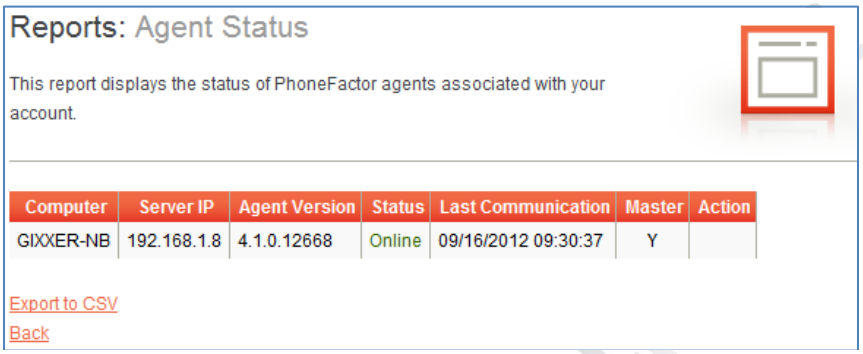


Εικόνα 5—7 One-Time Bypass

5.2.3 Διαχείριση Αναφορών PhoneFACTOR

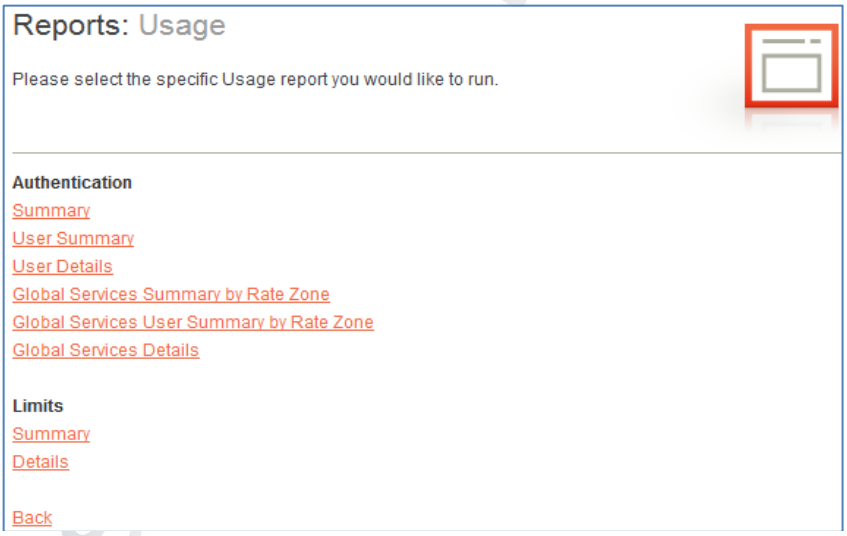
Για τη διαχείριση διάφορων αναφορών του PhoneFACTOR γίνεται χρήση των παρακάτω επιλογών:

Από το κεντρικό μενού επιλέγοντας *Reports→ Agent Status* παρουσιάζεται η εικόνα 5-8, αναφέροντας τους Η/Υ που είναι συνδεδεμένοι με το λογαριασμό, αναφέροντας την IP διεύθυνση του Η/Υ ,την έκδοση του Agent, αν είναι συνδεδεμένοι On-line, τελευταία επικοινωνία κλπ.



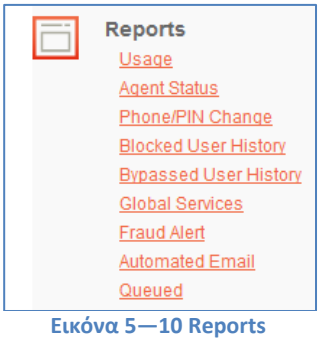
Εικόνα 5—8 Agent Status

Από το κεντρικό μενού επιλέγοντας *Reports→ Usage* εμφανίζεται μια λίστα επιλογών με αναφορές, σχετικά με τους διάφορους χρόνους χρήσης της υπηρεσίας του κάθε χρήστη.



Εικόνα 5—9 Reports:Usage

Στα υπόλοιπα υπό-μενού των *Reports,* εμφανίζονται διάφορες άλλες αναφορές σχετικά με την αλλαγή κωδικού, αποκλεισμός χρηστών κλπ. (Εικ- 5-10)



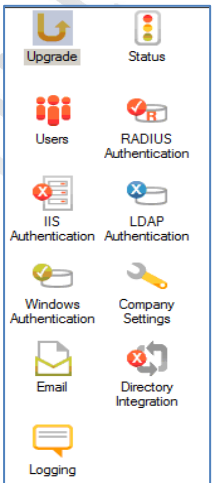
Εικόνα 5—10 Reports

5.3 Παραμετροποίηση PhoneFACTOR Agent

Το PhoneFACTOR Agent είναι ένα λογισμικό που χρησιμοποιείται ως διαμεσολαβητής για το κέντρο δεδομένων της PhoneFACTOR.Μετά την εγκατάσταση του PhoneFACTOR Agent που γίνεται από το λογισμικό που φορτώθηκε από το 4ο βήμα του URL https://pfweb.phonefactor.com/register/, πρέπει να γίνει ενεργοποίηση του από το πλήκτρο *"Activate"* μέσω του λογαριασμού αυθεντικοποίησης του διαχειριστή (Εικ. 5-11) που δημιουργήθηκε στο 1ο βήμα της διαχειριστική σελίδα του PhoneFACTOR στην παρ. 5.2.

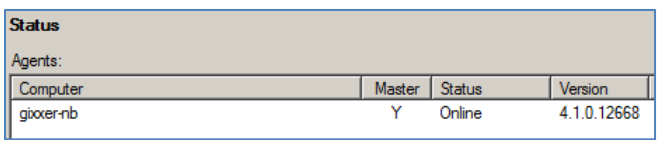
Εικόνα 5—11 Ενεργοποίηση PhoneFACTOR Agent

Μετά την αυθεντικοποίηση του διαχειριστή δίνεται η δυνατότητα παραμετροποίησης διάφορων πρωτοκόλλων και υπηρεσιών αυθεντικοποίησης όπως: RADIUS,IIS,LDAP,Windows, Directory Integration όπως εμφανίζονται στην εικ. 5-12. Η παραμετροποίηση που θα αναλυθεί στη συνέχεια, γίνεται μόνο για RADIUS Authentication και για Windows Authentication.



Εικόνα 5—12 Μενού PhoneFACTOR Agent

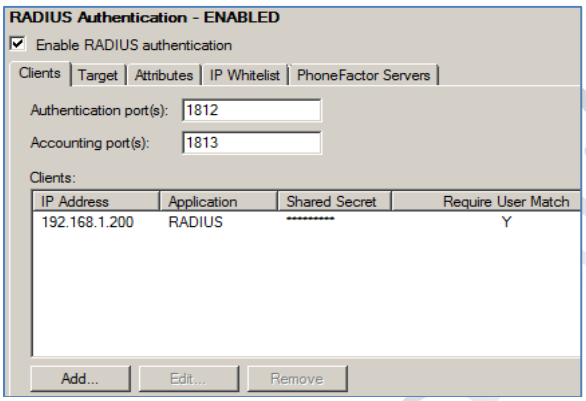
Επιλέγοντας το πλήκτρο "Status" εμφανίζονται οι Η/Υ που είναι διαθέσιμοι ως Agents και η κατάσταση διασύνδεσης τους με το κέντρο δεδομένων της PhoneFACTOR μέσω διαδικτύου.



Εικόνα 5—13 Status

5.3.1 Διαχείριση RADIUS Authentication

Από το κεντρικό μενού, επιλέγοντας το πλήκτρο *RADIUS Authentication* → *Clients* όπως φαίνεται στην εικ. 5-14, γίνονται οι παρακάτω ρυθμίσεις του πίνακα 5-1 για κάθε έναν client που χρειάζεται να χρησιμοποιήσει τον Agent ως PROXY, για να μπορέσει να κατευθυνθεί σε κάποιον RADIUS server.Στις παρακάτω ρυθμίσεις έχει παραμετροποιηθεί ως client, ο OpenVPN server.



Εικόνα 5—14 RADIUS Clients

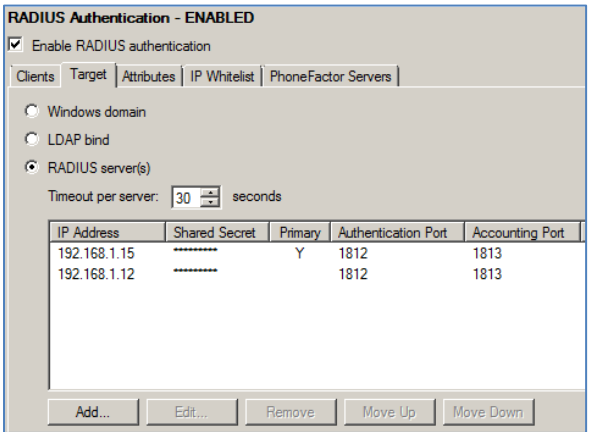
IP Address	Application	Shared Secret	Require User Match	Authentication Port	Accounting Port
192.168.1.200	RADIUS	Password2	Y	1812	1813

Πίνακας 5—1 Radius Clients

Επιλέγοντας τη δεύτερη καρτέλα *RADIUS Authentication* → *Target (εικ 5-15)* γίνονται οι παρακάτω ρυθμίσεις για κάθε server αυθεντικοποίησης που θα είναι διαθέσιμος κατά προτεραιότητα. Στην παρακάτω ρύθμιση έχει οριστεί ως πρωτόκολλο επικοινωνίας το RADIUS, ως πρωτεύων server ο WiKID server και ως δευτερεύων ο FreeRADIUS server με LDAP. Σε περίπτωση μη διαθεσιμότητας του πρώτου server μέσα σε 30 δευτερόλεπτα, γίνεται διαθέσιμος ο δεύτερος server.

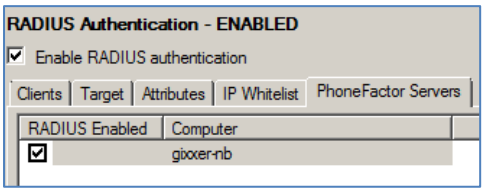
Target	IP Address	Shared Secret	Primary	Authentication Port	Accounting Port
RADIUS server(s)	192.168.1.15	Password2	Y	1812	1813
	192.168.1.12	Password2		1812	1813

Πίνακας 5—2 RADIUS Server(s)



Εικόνα 5—15 RADIUS Target

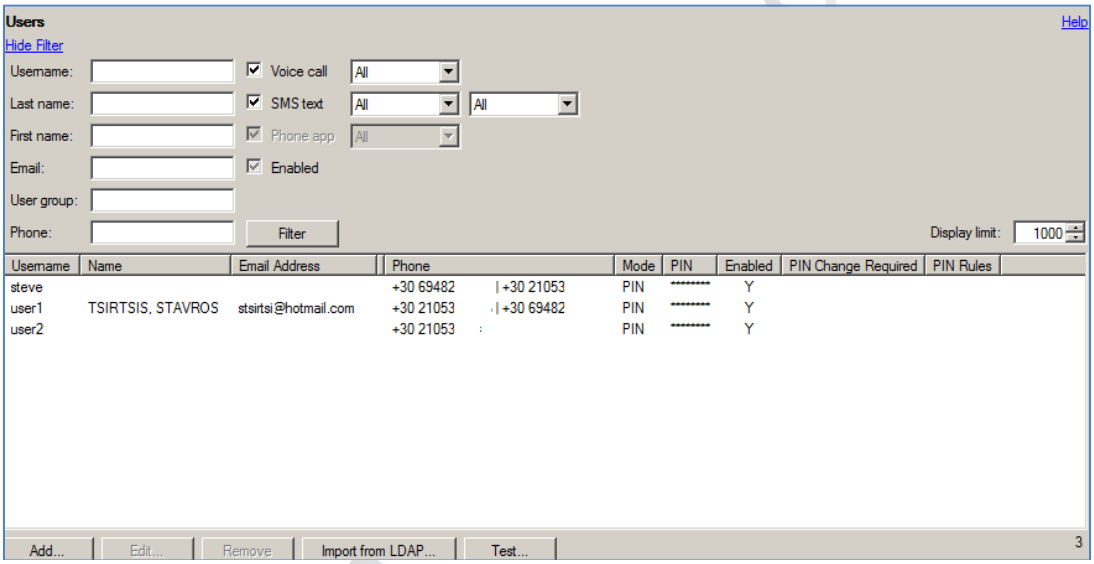
Επιλέγοντας την καρτέλα *RADIUS Authentication → PhoneFACTOR Servers* παρατηρούμε τους Η/Υ που έχουν το RADIUS πρωτόκολλο ενεργοποιημένο και είναι συσχετισμένοι με τον Agent.



Εικόνα 5—16 RADIUS PhoneFACTOR Servers

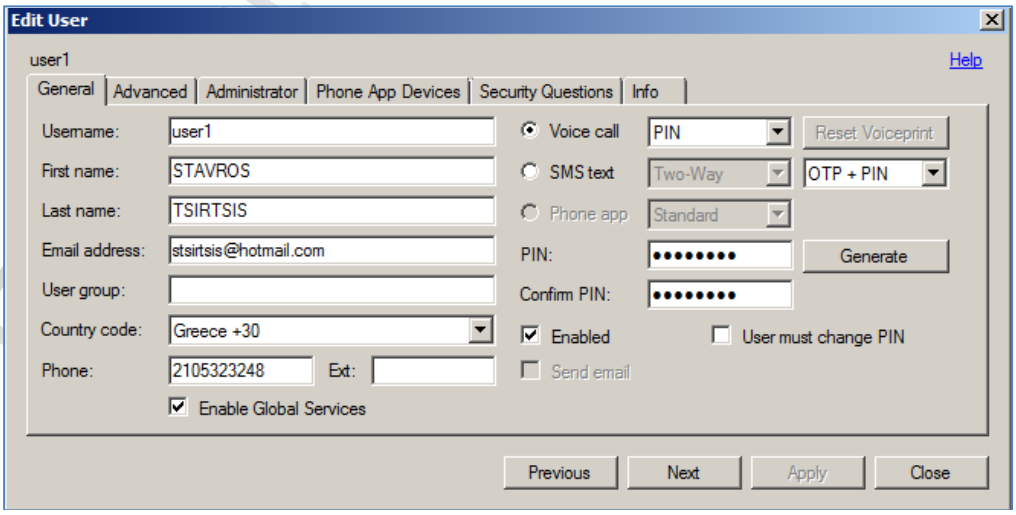
5.3.2 Διαχείριση Χρηστών

Οι χρήστες που μπορούν να αυθεντικοποιηθούν στα αντίστοιχα πρωτόκολλα ή υπηρεσίες, παρουσιάζονται στη λίστα της εικόνας 5-17, επιλέγοντας το πλήκτρο "Users".



Εικόνα 5—17 Λίστα Χρηστών PhoneFACTOR

Επιλέγοντας το πλήκτρο *"Add"* δημιουργείται ο χρήστης user1 όπως φαίνεται στην εικ. 5-18.

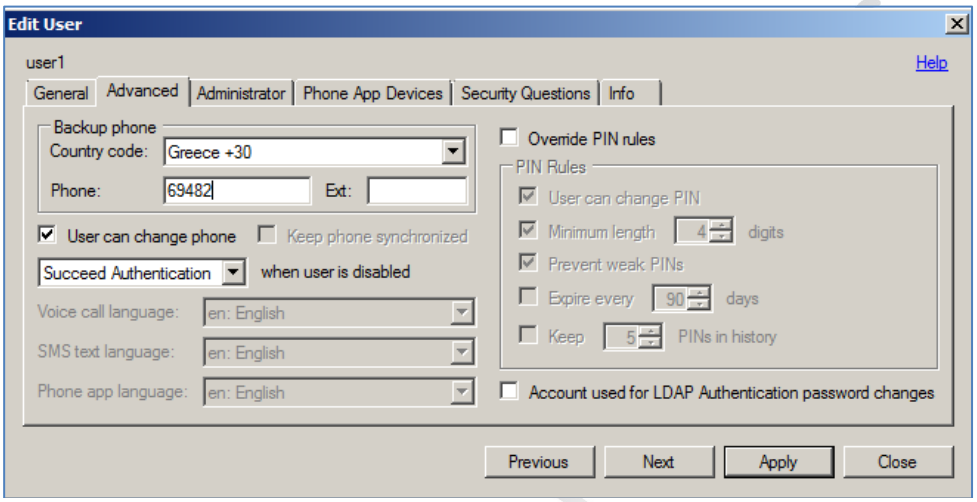


Εικόνα 5—18 Δημιουργία Χρήστη

Στα αριστερά πεδία συμπληρώνονται όλα τα απαραίτητα στοιχεία, με υποχρεωτικό πεδία τα *Username,Email address,Country Code, Phone* (πρώτος παράγοντας αυθεντικοποίησης). Στα δεξιά

πεδία επιλέγεται η αυθεντικοποίηση μέσω φωνητικής κλήσης και επιλογή δεύτερου παράγοντα αυθεντικοποίησης μέσω PIN (Voice Call → PIN).Στα 2 πεδία PIN συμπληρώνεται ο επιθυμητός αριθμός που θα χρησιμοποιείται (άνω των 4 ψηφίων) και για την ενεργοποίηση του λογαριασμού επιλέγεται το *"Enable".*

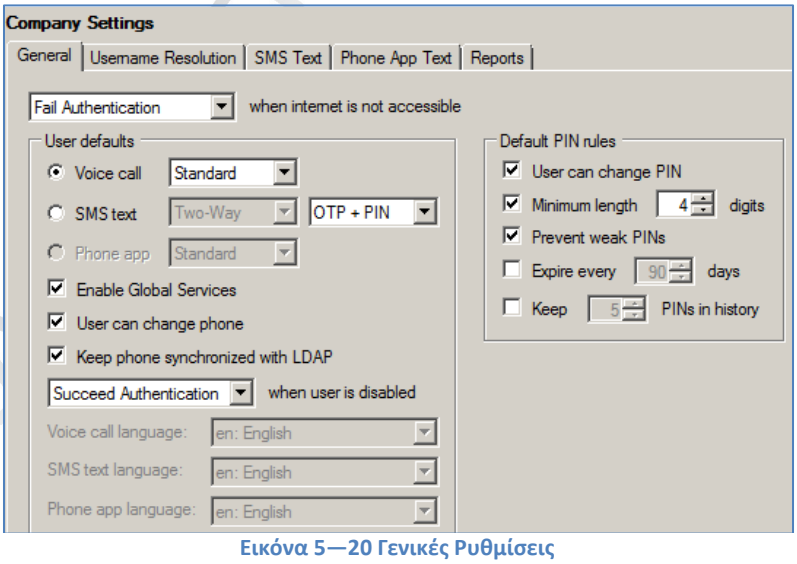
Επιλέγοντας την καρτέλα *"Advanced"* εισάγεται εναλλακτικός τηλεφωνικός αριθμός κλήσης προς το χρήστη σε περίπτωση μη διαθεσιμότητας του πρώτου αριθμού (σελ 5-19).



Εικόνα 5—19 Προχωρημένες Παράμετροι Χρήστη

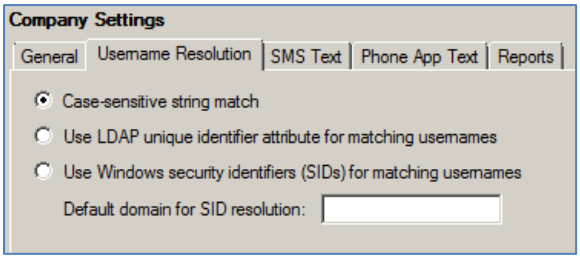
5.3.3 Διαχείριση Ρυθμίσεων

Επιλέγοντας το πλήκτρο *Company Settings → General* ,γίνεται παραμετροποίηση των παρακάτω παραμέτρων σχετικά με την προεπιλεγμένη εμφάνιση για τύπο αυθεντικοποίησης (φωνητική κλήση ή SMS)αριθμός ελάχιστων ψηφίων PIN κλπ (εικ. 5-20).



Εικόνα 5—20 Γενικές Ρυθμίσεις

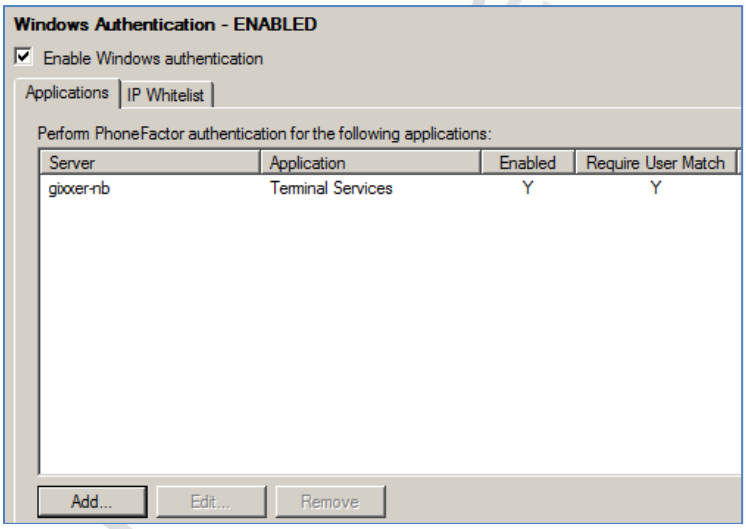
Επιλέγοντας το πλήκτρο *Company Settings →Username Resolution* επιλέγεται ο τύπος αναγνώρισης του username του χρήστη. Γίνεται επιλογή του *"Case-sensitive string match"* ,ώστε να μη συσχετίζεται το username με " SID ή unique identifier attribute" (εικ. 5-21).



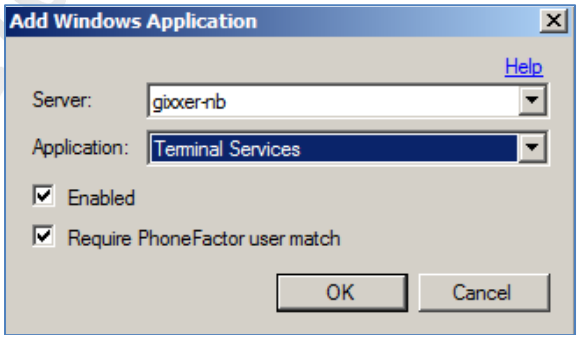
Εικόνα 5—21 Ρυθμίσεις Username

5.3.4 Διαχείριση Windows Authentication

Επιλέγοντας το πλήκτρο *Windows Authentication →Applications* γίνεται προσθήκη του Windows server που πρέπει να έχει εγκατεστημένο τον Agent (εικ. 5-22), για να μπορεί να χρησιμοποιεί για την αυθεντικοποίηση των χρηστών μέσω Terminal Services (Remote Desktop) εκτός από Username/Password και την αυθεντικοποίηση (επιλέγοντας Require User Match) στη συνέχεια με PhoneFACTOR (εικ. 5-23).



Εικόνα 5—22 Windows Authentication



Εικόνα 5—23 Προσθήκη Windows Application

Κατά την εφαρμογή, κάνοντας αρχικά επιτυχή αυθεντικοποίση μέσω πρώτου παράγοντα των διαπιστευτηρίων χρήστη σε απομακρυσμένο Λ.Σ. Windows (οποιοσδήποτε επιλεχθεί), γίνεται στη συνέχεια αυθεντικοποίηση δεύτερου παράγοντα μέσω του PhoneFACTOR για επιτυχή πρόσβαση.

5.4 Σενάριο Αυθεντικοποίησης Πολλαπλών Παραγόντων

Το PhoneFACTOR χρησιμοποιώντας τη λειτουργία των δύο παραγόντων αυθεντικοποίησης (τηλεφωνικής κλήσης σε αριθμό και PIN χρήστη) PhoneFACTOR, μπορεί να λειτουργήσει σε συνδυασμό με το σενάριο δύο παραγόντων αυθεντικοποίησης του WiKID, με PIN του τομέα WiKID και του OTP, που χρησιμοποιήθηκε στην παράγραφο 4.8.1, με αποτέλεσμα τον ασφαλέστερο συνδυασμό αυθεντικοποίησης χρήστη τεσσάρων παραγόντων και προσθέτοντας ενός παράγοντα αυθεντικοποίησης του WiKID client, τότε το σύνολο γίνεται 5 παραγόντων. Ουσιαστικά συνδυάζεται η λειτουργία, του PhoneFACTOR και του WiKID που περιγράφηκε στα προηγούμενα κεφάλαια ως υπηρεσίες αυθεντικοποίησης, για την λειτουργία του OpenVPN Client-Server. Στο PhoneFACTOR Agent έχει γίνει τέτοια παραμετροποίηση, ώστε σε περίπτωση μη διαθεσιμότητας του WiKID server, εντός 30 δευτερολέπτων να γίνεται αυθεντικοποίηση ως δεύτερη επιλογή, στον FreeRADIUS server με LDAP, όπου χρησιμοποιήθηκε στην παράγραφο 2, χρησιμοποιώντας όμως 3 παράγοντες αυθεντικοποίησης όπως περιγράφεται στην παρ. 5.4.2 (Θα πρέπει να το γνωρίζει ο χρήστης ότι σε περίπτωση που δεν μπορεί να αυθεντικοποιηθεί με το WiKID σε 30 δευτερόλεπτα, θα πρέπει να χρησιμοποιήσει το username/password του LDAP).

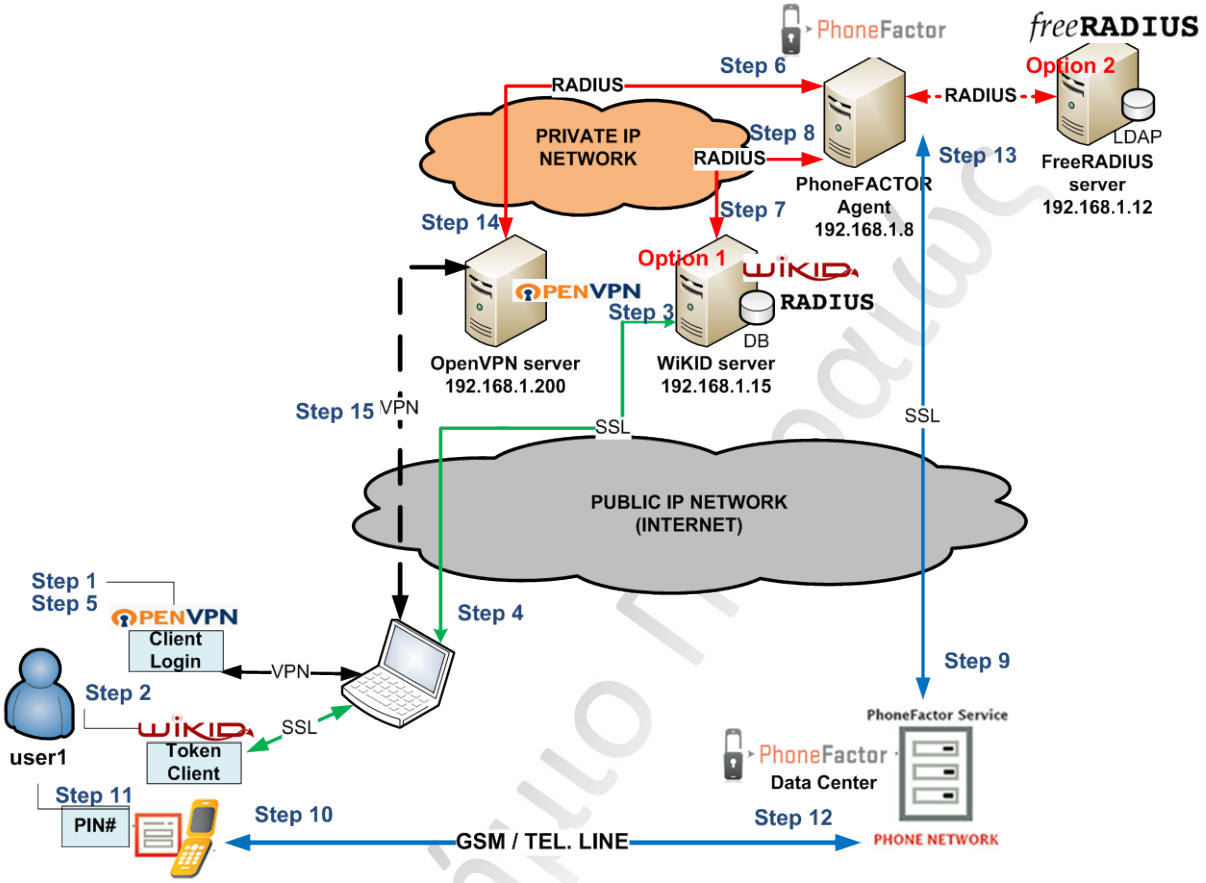
Προϋπόθεση για την επιτυχή ολοκλήρωση των βημάτων του σεναρίου που περιγράφεται στις δύο επόμενες παραγράφους, είναι να γίνει αρχικά η παραμετροποίηση του αρχείου /etc/openvpn/radiusplugin.cnf *του* OpenVPN server , έτσι ώστε ο server να μπορεί να χρησιμοποιεί ως PROXY server τον Agent του PhoneFACTOR, σύμφωνα με τον παρακάτω κώδικα του αρχείου:

```
server
{
        # The UDP port for radius accounting.
        acctport=1813
        # The UDP port for radius authentication.
        authport=1812
        # The name or ip address of the PhoneFACTOR
        name=192.168.1.8
        # How many times should the plugin send the if there is no response?
        retry=2
        # How long should the plugin wait for a response?
        wait=60
        # The shared secret.
        sharedsecret=Password2
}
```

Ουσιαστικά γίνεται αντικατάσταση της IP διεύθυνσης με αυτή του PhoneFACTOR Agent, ο αριθμός επανάληψης κλήσης και αντικατάσταση του χρόνου αναμονής *wait=60* του plugin για την τηλεφωνική κλήση του PhoneFACTOR.

5.4.1 Αυθεντικοποίηση (4+1) Παραγόντων OpenVPN-WiKID-PhoneFACTOR

Το παραπάνω σενάριο παρουσιάζεται σχηματικά στο σχήμα 5-2.

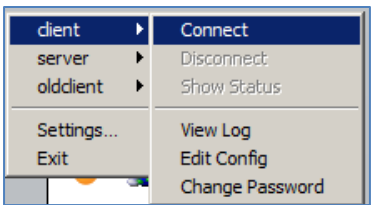


Σχήμα 5-2 Πρόσβαση στο OpenVPN server με αυθεντικοποίηση σε WiKID & PhoneFACTOR

Έχοντας παραμετροποίησει το PhoneFACTOR Agent όπως περιγράφηκε στην παρ. 5.3, στη συνέχεια για την επιτυχή αυθεντικοποίηση του OpenVPN Client στον OpenVPN server, εκτελούνται τα παρακάτω βήματα:

Βήμα 1

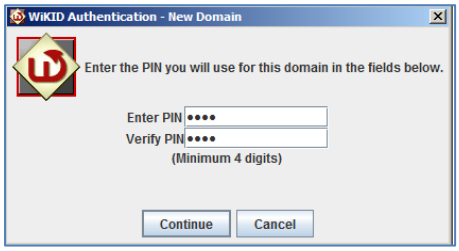
Γίνεται εκκίνηση του OpenVPN client για σύνδεση προς τον OpenVPN server,όπως περιγράφεται στην παρ. 4.8.



Εικόνα 5—24 Εκκίνηση OpenVPN Client

Βήμα 2

Έχοντας ανοίξει το WiKID Client από τον Η/Υ που έχει συσχετιστεί με τον user1 στον WiKID server, πληκτρολογείται από το χρήστη επιτυχώς, ο πρώτος παράγοντας αυθεντικοποίησης PIN στο WiKID Client για το Domain που έχει δημιουργηθεί (αναφορά για WiKID client στην παρ. 3.6.2).



Εικόνα 5—25 Εισαγωγή PIN του Domain

Βήμα 3

Το PIN αποστέλλεται στον WiKID server.

Βήμα 4

Παράγεται το Passcode από τον WiKID server και αποστέλλεται πίσω στο WiKID client.

Εικόνα 5—26 Δημιουργία OTP κωδικού

Βήμα 5

Αντιγράφεται το Passcode του WiKID client στο πεδίο password του OpenVPN client, αφού έχει εισαχθεί και το αντίστοιχο username user1 και αποστέλλεται στον OpenVPN server.

Εικόνα 5—27 Εισαγωγή του OTP κωδικού στον OpenVPN client

Βήμα 6

Αποστέλλεται το username/password από το OpenVPN server, στο PhoneFACTOR που λειτουργεί και ως PROXY RADIUS server.

Βήμα 7

Το PhoneFACTOR αποστέλλει username/password στον WiKID server.Έχοντας πιέσει το πλήκτρο OK μέσα σε 60 δευτερόλεπτα από τη δημιουργία του OTP του WiKID Client, γίνεται επιτυχής αυθεντικοποίηση στον WiKID Server, εφόσον είναι διαθέσιμος και ο WiKIDServer στον PhoneFACTOR μέσα σε 60 δευτερόλεπτα (Ορισμός πεδίου *Timeout per server: 60 seconds* στον PhoneFACTOR Agent)

Βήμα 8

Στη συνέχεια ο WiKID Server επιστρέφει μήνυμα επιτυχούς πρόσβασης προς τον PhonePACTOR Agent που λειτουργεί ως PROXY.

```
RADIUS-PLUGIN: BACKGROUND AUTH: Reply-Message:Access Granted
```

Η διαδικασία αυτή επιτυγχάνεται εφόσον γίνει αμοιβαία αυθεντικοποίηση του OpenVPN client στον OpenVPN server και αντίστροφα. Ουσιαστικά η αυθεντικοποίηση αυτού του βήματος είναι συνδυασμός τριών παραγόντων, όπου οι δύο παράγοντες είναι από το WiKID (PIN και Username/Passcode των 60 δευτ.) για το χρήστη και ο τρίτος παράγοντας από το ψηφιακό πιστοποιητικό του OpenVPN client για τον client.

Βήμα 9

Ο PhoneFACTOR Agent μέσω Internet επικοινωνεί σε ασφαλές κανάλι SSL με το κέντρο δεδομένων της PhoneFACTOR και ζητά αυθεντικοποίηση του χρήστη user1 μέσω τηλεφωνικής κλήσης.

Βήμα 10

Το κέντρο δεδομένων της PhoneFACTOR καλεί τον τηλεφωνικό αριθμό που αντιστοιχεί στο χρήστη user1, ως τρίτος παράγοντας αυθεντικοποίησης του χρήστη και μέσω φωνητικών μηνυμάτων ζητάει την πληκτρολόγηση του PIN.

Βήμα 11

Ο χρήστης user1 εισάγει το PIN και στο τέλος το σύμβολο #.

Βήμα 12

Η τηλεφωνική συσκευή επιστρέφει το PIN στο κέντρο δεδομένων ως τέταρτος παράγοντας αυθεντικοποίησης χρήστη.

Βήμα 13

Το κέντρο δεδομένων επιστρέφει στον PhoneFACTOR Agent μέσω SSL, επιτυχή αυθεντικοποίηση.

Βήμα 14

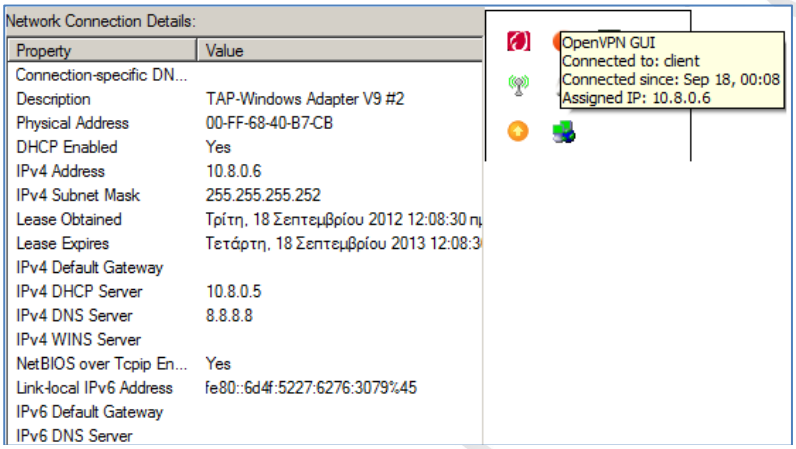
Αν ο χρόνος εισαγωγής του OTP Client έχει δοθεί εντός του χρόνου 60 δευτερολέπτων (Ο χρόνος αυτός έχει οριστεί στο αρχείο *radiusplugin.conf* του OpenVPN server στο πεδίο *wait=60*), Ο

PhoneFACTOR Agent αποστέλλει το παρακάτω μήνυμα στον OpenVPN server, για να αυθεντικοποιήσει το χρήστη user1.

```
RADIUS-PLUGIN: BACKGROUND AUTH: Reply-Message:Access Granted
```

Βήμα 15

Ο χρήστης user1 έχει αυθεντικοποιηθεί και ο OpenVPN server εγκαθιστά VPN σύνδεση με τον OpenVPN client.



Εικόνα 5—28 Επιτυχής Σύνδεση OpenVPN client μέσω WiKID & PhoneFACTOR

Δοκιμάζοντας την επικοινωνία του OpenVPN client δίνοντας *ping 10.8.0.1* στον OpenVPN server, παρατηρούμε ότι υπάρχει ανταπόκριση του server οπότε το VPN λειτουργεί κανονικά (Εικ. 5-29).

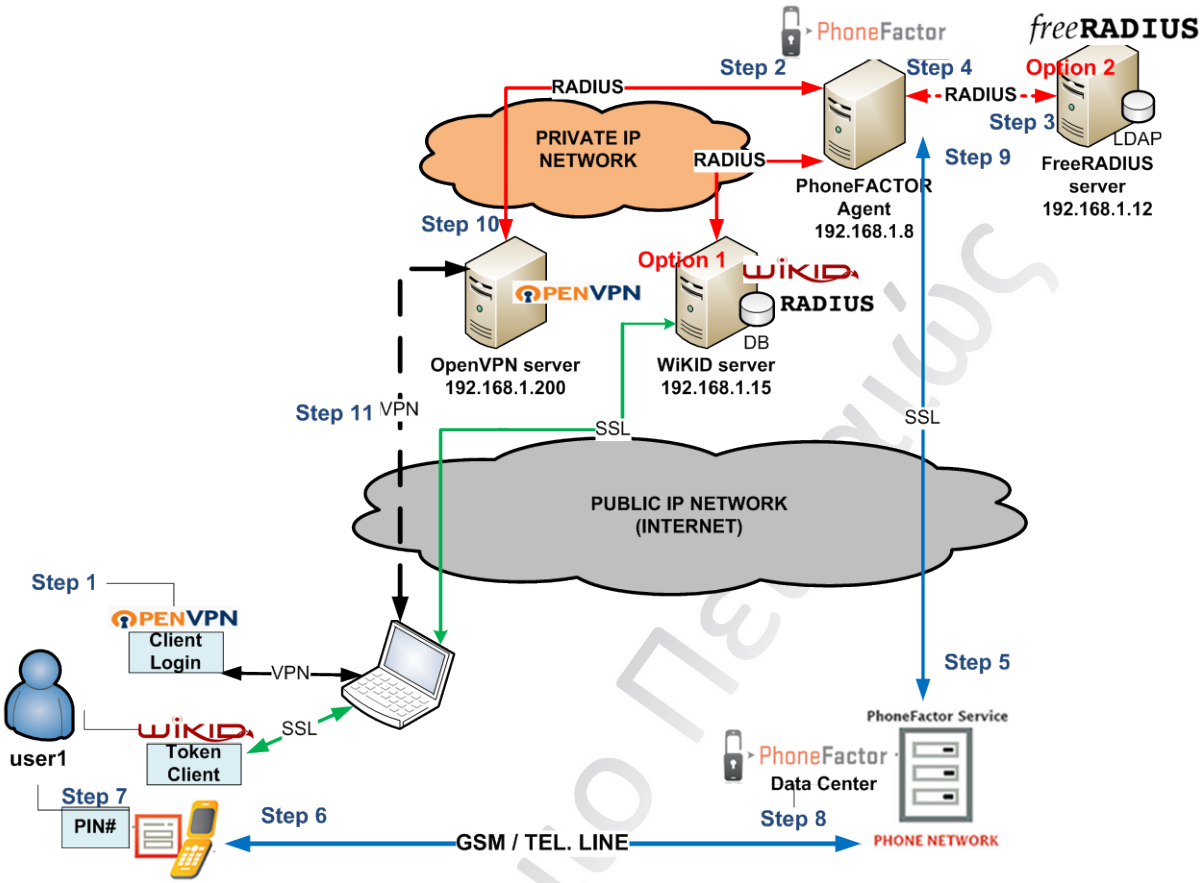
```
C:\Users\steve>ping 10.8.0.1

Pinging 10.8.0.1 with 32 bytes of data:
Reply from 10.8.0.1: bytes=32 time=1ms TTL=64
Reply from 10.8.0.1: bytes=32 time=1ms TTL=64
Reply from 10.8.0.1: bytes=32 time=1ms TTL=64
Reply from 10.8.0.1: bytes=32 time=1ms TTL=64

Ping statistics for 10.8.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Εικόνα 5—29 Δοκιμή VPN σύνδεσης

5.4.2 Αυθεντικοποίηση (3+1) Παραγόντων OpenVPN-FreeRADIUS-PhoneFACTOR

Το PhoneFACTOR Agent έχει παραμετροποιηθεί έτσι ώστε σε περίπτωση μη διαθεσιμότητας του WiKID server εντός 30 δευτερολέπτων να προσπαθήσει να αυθεντικοποιήσει με ένα παράγοντα (username/password) τον χρήστη του OpenVPN server στη βάση LDAP του FreeRADIUS server. Χρησιμοποιώντας τη λειτουργία των δύο παραγόντων αυθεντικοποίησης τηλεφωνικής κλήση σε αριθμό και PIN χρήστη του PhoneFACTOR, επιτυγχάνεται αυθεντικοποίηση χρήστη τριών παραγόντων και προσθέτοντας ενός παράγοντα αυθεντικοποίησης του WiKID client, τότε το σύνολο γίνεται τεσσάρων παραγόντων. Ουσιαστικά συνδυάζεται η λειτουργία, του PhoneFACTOR και του FreeRADIUS server με FreeLDAP, όπως περιγράφηκε στις προηγούμενες παραγράφους ως υπηρεσία αυθεντικοποίησης, για την λειτουργία του OpenVPN Client-Server.

Το παραπάνω τμήμα του σεναρίου παρουσιάζεται σχηματικά στο σχήμα 5-3.

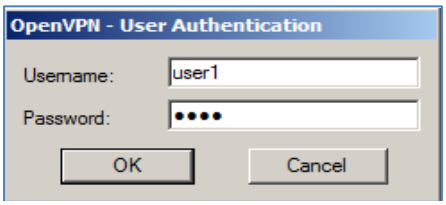


Σχήμα 5-3 Πρόσβαση χρήστη στο OpenVPN server με αυθεντικοποίηση σε FreeRADIUS LDAP & PhoneFACTOR

Έχοντας παραμετροποίησει το PhoneFACTOR Agent όπως περιγράφηκε στην παρ. 5.3, στη συνέχεια για την επιτυχή αυθεντικοποίηση του OpenVPN Client στον OpenVPN server, εκτελούνται τα παρακάτω βήματα:

Βήμα 1

Γίνεται εκκίνηση του OpenVPN client για σύνδεση προς τον OpenVPN server,όπως περιγράφεται στην παρ. 4.8 και στα πεδία Username/ Password εισάγονται τα στοιχεία που είναι καταχωρημένα στη βάση του LDAP (user1/ 1234) στον FreeRADIUS και αποστέλλονται από το OpenVPN client στον OpenVPN server.



Εικόνα 5—30 Εισαγωγή Username/Password στον OpenVPN client

Βήμα 2

Αποστέλλεται το username/password από το OpenVPN server, στο PhoneFACTOR που λειτουργεί και ως PROXY RADIUS server.

Βήμα 3

Το PhoneFACTOR αποστέλλει username/password στον FreeRADIUS server, εφόσον είναι διαθέσιμος και ο FreeRADIUS στον PhoneFACTOR μέσα σε 60 δευτερόλεπτα (Ορισμός πεδίου *Timeout per server: 60 seconds* στον PhoneFACTOR Agent)

Βήμα 4

Στη συνέχεια ο FreeRADIUS server επιστρέφει μήνυμα επιτυχούς πρόσβασης προς τον PhonePACTOR Agent που λειτουργεί ως PROXY.

RADIUS-PLUGIN: BACKGROUND AUTH: Reply-Message:Access Granted

Η διαδικασία αυτή επιτυγχάνεται εφόσον γίνει αμοιβαία αυθεντικοποίηση του OpenVPN client στον OpenVPN server και αντίστροφα. Ουσιαστικά η αυθεντικοποίηση αυτού του βήματος είναι συνδυασμός δύο παραγόντων, όπου ο πρώτος παράγοντας είναι από το FreeRADIUS (Username/Password) για το χρήστη και ο δεύτερος παράγοντας από το ψηφιακό πιστοποιητικό του OpenVPN client για τον client.

Βήμα 5

Ο PhoneFACTOR Agent μέσω Διαδικτύου επικοινωνεί σε ασφαλές κανάλι SSL με το κέντρο δεδομένων της PhoneFACTOR και ζητά αυθεντικοποίηση του χρήστη user1 μέσω τηλεφωνικής κλήσης.

Βήμα 6

Το κέντρο δεδομένων της PhoneFACTOR καλεί τον τηλεφωνικό αριθμό που αντιστοιχεί στο χρήστη user1 ως τρίτος παράγοντας αυθεντικοποίησης του χρήστη και ζητάει PIN.

Βήμα 7

Ο χρήστης user1 εισάγει το PIN και στο τέλος το σύμβολο #.

Βήμα 8

Η τηλεφωνική συσκευή επιστρέφει το PIN στο κέντρο δεδομένων ως τέταρτος παράγοντας αυθεντικοποίησης χρήστη.

Βήμα 9

Το κέντρο δεδομένων επιστρέφει στον PhoneFACTOR Agent μέσω SSL, επιτυχή αυθεντικοποίηση.

Βήμα 10

Αν ο χρόνος εισαγωγής του OTP Client έχει δοθεί εντός του χρόνου 60 δευτερολέπτων (Ο χρόνος αυτός έχει οριστεί στο αρχείο *radiusplugin.conf* του OpenVPN server στο πεδίο *wait=60*), Ο PhoneFACTOR Agent αποστέλλει το παρακάτω μήνυμα στον OpenVPN server, για να αυθεντικοποιήσει το χρήστη user1.

RADIUS-PLUGIN: BACKGROUND AUTH: Reply-Message:Access Granted

Βήμα 11

Ο χρήστης user1 έχει αυθεντικοποιηθεί και ο OpenVPN server εγκαθιστά VPN σύνδεση με τον OpenVPN client επιτυχώς.

6 Βιβλιογραφία

1) Εγκατάσταση και ρύθμιση του λογισμικού FreeRADIUS server για αυθεντικοποίηση και ανταλλαγή κλειδιών (EAP methods) - *Τσιρτσής-Ζάγουρας-Πρέτσιος, Εργασία Παν. Πειραιά*
2) HOWTO: EAP/TLS Setup for FreeRADIUS and Windows XP Supplicant Version 1.0.1 April 18, 2002 - *Ken Roser Version 1.0*
3) 802.11 Wireless Networks- Security and Analysis - *(SPRINGER) ALAN HOLT –CHI-YU HUANG*
4) https://help.ubuntu.com/community/ApacheMySQLPHP
5) http://www.mydeveloperblog.com/linux-tutorial/radius/radius-servers-installation-guide-freeradius-ubuntu-mysql/
6) http://www.youtube.com/watch?v=5GPWvD0ralQ
7) http://www.docstoc.com/docs/22211496/How-to-configure-FreeRadiusnet-to-work-with-Alcatel-Lucent-OmniSwitch
8) http://en.wikipedia.org/wiki/FreeRADIUS
9) http://oss.sgi.com/LDP/HOWTO/LDAP-Implementation-HOWTO/radius.html
10) http://www.keller.com/wifi/CNIT107HW7.html
11) http://www.linuxhomenetworking.com/wiki/index.php/Quick_HOWTO_:_Ch31_:_Centralized_Logins_Using_LDAP_and_RADIUS
12) http://www.youtube.com/watch?v=DM_UQVVVtoY&feature=related
13) http://ubuntuforums.org/showthread.php?t=1760830
14) http://www.youtube.com/watch?v=9E-ZlPjaT0Y
15) http://wiki.debian.org/PhpLdapAdmin
16) http://castix.wordpress.com/2009/04/05/openldap-installation-and-adding-objects-using-phpldapadmin-and-ldapadd/
17) http://vuksan.com/linux/dot1x/802-1x-LDAP.html
18) http://el.wikipedia.org/wiki/LDAP
19) http://www.wikidsystems.com/support/wikid-support-center/how-to/using-wikid-strong-authentication-with-openvpn
20) http://www.wikidsystems.com/downloads/download_images/wikid-support-center/installation-how-tos
21) http://www.wikidsystems.com/downloads/download_images/downloads/token-clients/
22) http://www.wikidsystems.com/support/wikid-support-center/installation-how-tos/wikid-quickstart-installation-cheatsheet-version-3.x

23) http://www.wikidsystems.com/downloads/download_images/support/wikid-support-center/manual/how-to-install-the-wikid-strong-authentication-server
24) http://www.wikidsystems.com/support/wikid-support-center/manual/how-to-install-the-wikid-strong-authentication-server/referencemanual-all-pages
25) http://www.youtube.com/watch?v=LajxJanyHEE
26) http://www.slideshare.net/nowen/wi-kid-systems-incversion-532
27) http://www.wikidsystems.com/learn-more/technology/overview
28) https://www.wikidsystems.com/
29) http://en.wikipedia.org/wiki/OpenVPN
30) http://openvpn.net/index.php/access-server/download-openvpn-as-vm/164.html?osfamily=Virtual%20Appliance%20(VMWare)&ex=1
31) http://noc.auth.gr/services/connectivity/vpn/
32) http://openvpn.net/images/pdf/OpenVPN_Access_Server_Sysadmin_Guide_Rev.pdf
33) http://pkgs.org/ubuntu-12.04/ubuntu-main-amd64/libpam0g-dev_1.1.3-7ubuntu2_amd64.deb.html
34) http://freeradius.org/pam_radius_auth/
35) http://www.wikidsystems.com/support/wikid-support-center/how-to/using-wikid-strong-authentication-with-openvpn/support/wikid-support-center/how-to/pam-radius-how-to/
36) http://www.wikidsystems.com/support/wikid-support-center/how-to/using-wikid-strong-authentication-with-openvpn
37) http://pkgs.org/ubuntu-10.04/ubuntu-main-i386/libpam-radius-auth_1.3.17-0ubuntu3_i386.deb.html
38) http://community.openvpn.net/openvpn/wiki/Easy_Windows_Guide#CertificatesandKeys
39) http://openvpn.net/index.php/open-source/documentation/howto.html
40) https://help.ubuntu.com/community/OpenVPN
41) https://help.ubuntu.com/11.10/serverguide/openvpn.html
42) http://safesrv.net/setup-freeradius-plugin-and-openvpn-source/
43) http://openmaniak.com/openvpn_tutorial.php
44) http://openmaniak.com/openvpn_static.php
45) http://en.wikipedia.org/wiki/PhoneFactor
46) http://www.phonefactor.com/how-it-works/
47) http://www.phonefactor.com/solutions/ssl-vpn-authentication/
48) http://www.openldap.org/doc/admin22/intro.html