



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Αναζήτηση , μελέτη και υλοποίηση μεθόδων πιστοποίησης και παρακολούθησης χρηστών (AAA – Authentication , Authorization , Accounting) για ασύρματα δίκτυα WLAN.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέξανδρος, Δ. Γιαμάς

Επιβλέπων : Μιχαήλ, Ε., Θεολόγου
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2005



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

Αναζήτηση , μελέτη και υλοποίηση μεθόδων πιστοποίησης και παρακολούθησης χρηστών (AAA – Authentication , Authorization , Accounting) για ασύρματα δίκτυα WLAN.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέξανδρος, Δ. Γιαμάς

Επιβλέπων : Μιχαήλ, Ε., Θεολόγου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 11^η Ιουλίου 2005.

.....
Μιχαήλ Θεολόγου
Καθηγητής Ε.Μ.Π.

.....
Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.

.....
Γεώργιος Στασινόπουλος
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2005

.....
Αλέξανδρος, Δ Γιαμάς

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Αλέξανδρος, Γιαμάς, 2005

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

ΠΕΡΙΛΗΨΗ

Η ακόλουθη διπλωματική εκπονήθηκε στον Τομέα Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου.

Το θέμα της διπλωματικής είναι η Αναζήτηση , μελέτη και υλοποίηση μεθόδων πιστοποίησης και παρακολούθησης χρηστών (AAA – Authentication , Authorization , Accounting) για ασύρματα δίκτυα WLAN.

Τα ασύρματα δίκτυα WLAN είναι ένας ραγδαία αναπτυσσόμενος κλάδος που συνδυάζει τεχνολογίες τόσο από τον τομέα της Πληροφορικής όσο και από τον τομέα των Τηλεπικοινωνιών. Λόγω της ταχείας ανάπτυξης τους κρίθηκε απαραίτητη η αναζήτηση και μελέτη λύσεων από μηδενική βάση. Μελετώντας προσεκτικά τις διαθέσιμες υλοποιήσεις καταλήξαμε σε μια λύση που συνδυάζει ένα ευρύ φάσμα τεχνολογιών. Χρησιμοποιώντας έναν router σαν authentication-proxy , τον IAS, την SQL Server, το Active Directory και το Visual Studio .NET 2003 υλοποιήσαμε μια πλατφόρμα πιστοποίησης και παρακολούθησης χρηστών. Με τις κατάλληλες ρυθμίσεις δημιουργήσαμε στην ουσία την πλατφόρμα ενός παροχέα internet ο οποίος θα ήθελε να εγκαταστήσει ένα ή περισσότερα hotspots με κεντρική διαχείριση και λογιστική χρέωση.

Η παρούσα διπλωματική εργασία αποτελείται από πέντε κεφάλαια συν τα παραρτήματα. Στο πρώτο κεφάλαιο παρουσιάζουμε το θεωρητικό υπόβαθρο του πρωτόκολλου 802.11 το οποίο είναι απαραίτητο για την πληρέστερη κατανόηση των επόμενων κεφαλαίων. Στο δεύτερο κεφάλαιο παρουσιάζονται συνοπτικά κάποια θέματα σχετικά με την ασφάλεια των ασύρματων δικτύων τύπου WLAN. Στο τρίτο κεφάλαιο παρουσιάζεται το AAA Framework πάνω στο οποίο βασίστηκε η υλοποίηση της πλατφόρμας καθώς και το ανταγωνιστικό του Cisco Service Selection Gateway. Στο τέταρτο κεφάλαιο εμβαθύνουμε στην αρχιτεκτονική της πλατφόρμας πιστοποίησης και παρακολούθησης χρηστών με έμφαση στην διαμόρφωση των διαφόρων τεχνολογιών. Τέλος, στο πέμπτο κεφάλαιο παρουσιάζεται η αρχιτεκτονική του συστήματος το οποίο φιλοξενεί τον IAS, την SQL Server και την εφαρμογή διαχείρισης και λογιστικής χρέωσης Account++. Στο παράρτημα Α παρατίθεται ο πηγαίος κώδικας της εφαρμογής Account++. Στο παράρτημα Β παρατίθεται ένα research paper το οποίο σχετίζεται με την διπλωματική εργασία μου και στο οποίο είχα την τιμή να συμμετάσχω και να εγκριθεί για δημοσίευση και παρουσίαση στο συνέδριο ICAS'05 και ICNS'05. Τέλος, στο παράρτημα Γ παρατίθεται η βιβλιογραφία.

Σε αυτό το σημείο θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον υπεύθυνο για την εκπόνηση της διπλωματικής καθηγητή κ. Μιχάλη Θεολόγου για την πολύτιμη καθοδήγηση που προσέφερε καθώς και στον υποψήφιο διδάκτορα κ. Δημήτρη Νικητόπουλο για την σημαντική βοήθεια και την αγαστή συνεργασία που είχαμε. Επίσης θα ήθελα να ευχαριστήσω τους υποψήφιους διδάκτορες κ. Νίκο Παπαουλάκη και κ. Άγγελο Τράκο για τις πολύτιμες συμβουλές τους που προσέφεραν απλόχερα σε κάθε στάδιο της διπλωματικής μου εργασίας.

Αλέξανδρος Γιαμάς
Αθήνα, Ιούνιος 2005

Λέξεις κλειδιά: ασύρματο δίκτυο , εξουσιοδότηση , λογιστική χρέωση , πιστοποίηση

ABSTRACT

The following diploma thesis project was implemented in the Division of Communication, Electronic and Information Engineering of the school of Electrical and Computer Engineering of the National Technical University of Athens.

This project's main subject is the seeking, study and implementation of various methods for Authentication, Authorization and Accounting under WLAN wireless networks.

WLAN wireless networking is a rapidly evolving area of study which combines theoretical substratum from areas such as those of Computer Science and Telecommunication Engineering. Due to its rapid evolution it was decided to seek and study on various solutions from the ground up. After an insightful study of the available implementations we headed towards a solution which combines a wide spectrum of available technologies. By using a router as an authentication-proxy and combining the strengths of IAS, SQL Server, Active Directory and Visual Studio .NET 2003 we implemented a platform which could be used by an ISP that would like to deploy one or more hotspots under a platform of central administration and accounting.

This diploma thesis project is composed of five chapters plus three appendixes. In the first chapter we present the theoretical substratum of the 802.11 protocol which is required for the intimate understanding of the following chapters. In the second chapter we present a synopsis of the greater issues regarding WLAN security. The third chapter is about the AAA Framework, on which was based the implementation of the AAA platform. There is also a presentation of the Cisco Service Selection Gateway, an antagonistic platform to the AAA Framework. In the fourth chapter we go further into the details of the architecture of the AAA platform with an emphasis on the configuration of the various technologies that were used for the implementation. Finally, the fifth chapter is dealing about the system architecture. This system is hosting IAS, SQL Server and the administration and accounting application Account++. In the appendix A we present the source code of the application Account++. The Appendix B contains a research paper which is related to the study field of my diploma thesis project. To my greatest happiness this paper to which I have contributed was accepted for publication and presentation at the ICAS'05 and ICNS'05 conference. Finally, the appendix C contains bibliography and references.

At this point I would like to express my great appreciation to the professor in charge of my diploma thesis project Mr. Michael Theologou for the invaluable guidance he offered to me. Also, I would like to sincerely thank my tutor in this project, Ph.D. candidate Mr. Dimitris Nikitopoulos for the studious support and help and the great collaboration we had. I would also like to thank the Ph.D. candidates Mr. Nikos Papaoulakis and Mr. Angelos Trakos for their valued advices that they freely gave to me in every stage of my project.

Alexandros Giamas
Athens, June 2005

Key words: .NET , 802.11, 802.11a , 802.11b , 802.11g , AAA Framework , accounting , ACL , active directory , authentication , authorization , C++ , cisco, IAS , ICAS , ICNS , managed code , OOP , radius , router, service selection gateway , SQL server, SSG , UML , unmanaged code, Visual Studio , WEP , WLAN , WPA

Αφιερώνεται στην οικογένεια και στους φίλους μου που με στηρίζουν σε κάθε μου βήμα

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1

1.Θεωρητικό υπόβαθρο πάνω στο 802.11 πρωτόκολλο	12
1.1 Εισαγωγή.....	12
1.2 Το πρωτόκολλο 802.11:.....	12
1.2.1 Προδιαγραφές 802.11	13
1.2.2 Πάκετο δεδομένων τύπου 802.11	14
1.3 Παραλλαγές του 802.11	17
1.4 Τα κυριότερα εμπορικά πρωτόκολλα σήμερα:	19
1.5 Διαμόρφωση σήματος στα κυριότερα πρωτόκολλα:	21
1.5.1 Διαμόρφωση σήματος στο 802.11b:	21
1.5.2 Διαμόρφωση σήματος στο 802.11a:	23
1.5.3 Διαμόρφωση σήματος στο 802.11g:	25
1.6 Το μέλλον είναι εδώ.....	26

ΚΕΦΑΛΑΙΟ 2

2. Ασφάλεια Ασύρματων Δικτύων (WLAN)	30
2.1 Εισαγωγή.....	30
2.2 Ανάλυση μηχανισμών ασφαλείας σε WLAN.....	31
2.3 Πρωτόκολλο κρυπτογράφησης WEP (isn't?)	32
2.3.1 Τρόπος λειτουργίας.....	32
2.3.2 Προβλήματα ασφαλείας WEP.....	33
2.4 WPA	34

ΚΕΦΑΛΑΙΟ 3

3. AAA Framework.....	36
3.1 Authentication (Πιστοποίηση χρήστη)	36
3.1.1 Εισαγωγή.....	36
3.1.2 Τι είναι το AAA framework.....	36
3.2 Μέθοδοι πιστοποίησης με τη χρήση του AAA framework.....	36
3.3 Μέθοδοι πιστοποίησης χωρίς τη χρήση του AAA framework.....	38
3.4 Authorization (εξουσιοδότηση)	44
3.4.1 Μέθοδοι εξουσιοδότησης	44
3.5 Εξουσιοδότηση τύπου Radius	44
3.5.1 Εισαγωγή.....	44
3.5.2 Πως λειτουργεί	45
3.5.3 Μοντέλο Challenge/Response	46
3.5.4 Χρήση proxy	47
3.6 Accounting	48
3.7 Cisco Service Selection Gateway.....	53
3.7.1 Εισαγωγή.....	53

3.7.2 Τύποι υπηρεσιών Cisco SSG	53
3.7.3 Χαρακτηριστικά του Cisco SSG	54
3.7.4 One platform to rule them all.....	56

ΚΕΦΑΛΑΙΟ 4

4 Αρχιτεκτονική πλατφόρμας	57
4.1 Τοπολογία δικτύου	57
4.2 router (ACL , interfaces).....	59
4.3 active directory	62
4.4 IAS	65

ΚΕΦΑΛΑΙΟ 5

5 Αρχιτεκτονική συστήματος	68
5.1 Δομή βάσης SQL server	68
5.1.1 Έισαγωγή.....	68
5.1.2 Βήματα παραμετροποίησης.....	68
5.2 Δομή εφαρμογής.....	72
5.2.1 Έισαγωγή.....	72
5.2.2 Απαιτήσεις εφαρμογής	73
5.3 Περιγραφή υλοποίησης εφαρμογής	73
5.4 Περιγραφή των σημαντικότερων λειτουργιών της εφαρμογής (περιγραφή σε επίπεδο methods που δημιουργήσα)	75
5.5 Περιγραφή αρχιτεκτονικής λογισμικού με χρήση UML Diagrams.....	76
5.6 Περιγραφή γραφικής διεπαφής χρήστη	84
Appendix A	93
Source code	93
Appendix B	180
Research paper.....	180
Appendix C.....	189
Βιβλιογραφία	189

Κατάλογος σχημάτων:

Σχήμα 1.Ι Στρώματα OSI του 802.11	14
Σχ. 1.ΙΙ 802.11 MAC Frame Format	14
Σχ. 1.ΙΙΙ Περιγραφές υποπεδίων του 802.11 πακέτου.....	15
Σχ. 1.ΙV Οι διάφορες παραλλαγές του 802.11 πρωτοκόλλου.....	19
Σχ. 1.V Παραλλαγές του 802.11	19
Σχ. 1.VΙ Ρυθμαπόδοση των παραλλαγών του 802.11 σε συνάρτηση με την απόσταση.....	20
Σχ. 1.VΙΙ Κωδικοποίηση παραμέτρων φάσεων στο 802.11b	22
Σχ. 1.VΙΙΙ Διαφορική διαμόρφωση QPSK.....	22
Σχ. 1.ΙΧ Απόκριση συχνότητας, εξάπλωση φάσματος.....	22
Σχ. 1.Χ Κανάλια επικοινωνίας στο 802.11b.....	23
Σχ. 1.ΧΙ Διαμόρφωση στο 802.11a.....	25
Σχ. 1.ΧΙΙ Κανάλια και ισχύς εκπομπής στο 802.11a.....	25
Σχ. 1.ΧΙΙΙ Λειτουργία σε μικτό περιβάλλον b/g.....	26
Σχ. 1.ΧΙV ΜΙΜΟ with spatial multiplexing	28
Σχ. 2.Ι Η βιομηχανία 802.11	30
Σχ. 2.ΙΙ Κρυπτογράφηση WEP	32
Σχ. 2.ΙΙΙ Αποκρυπτογράφηση WEP	32
Σχ. 3.Ι 802.1X πριν την πιστοποίηση (pre-auth).....	41
Σχ. 3.ΙΙ 802.1X μετά από επιτυχημένη πιστοποίηση χρήστη.....	42
Σχ 3.ΙΙΙ Βήματα για μια επιτυχημένη πιστοποίηση στο 802.1X.....	43
Σχ. 3.ΙV Βήματα για πιστοποίηση με χρήση proxy διακομιστή.....	48
Σχ. 3.V Αρχιτεκτονική συστοιχίας πολλαπλών διακομιστών πρόσβασης.....	49
Σχ. 3.VΙ Αρχιτεκτονική Cisco SSG.....	53
Σχ. 3.VΙΙ Τύποι υπηρεσιών Cisco SSG.....	54
Σχ 3.VΙΙΙ Αρχιτεκτονική κοινής δημιουργίας και διαχείρισης υπηρεσιών	56
Σχ. 4.Ι Η αρχική τοπολογία του δικτύου	57
Σχ. 4.ΙΙ Η τελική τοπολογία του δικτύου	58
Σχ. 4.ΙΙΙ Κεντρικός έλεγχος πιστοποίησης χρηστών με χρήση του Active Directory	63
Σχ. 4.ΙV Καρτέλα Dial-in της εφαρμογής Active Directory Users and Computers	64
Σχ. 4.V Properties χρήστη της εφαρμογής Active Directory Users and Computers	64
Σχ. 4.VΙ Επικοινωνία IAS - router.....	65
Σχ. 4.VΙΙ Επικοινωνία του IAS με τον SQL server	66
Σχ. 4.VΙΙΙ IAS Remote Access Policies	67

Σχ. 4.IX Ρυθμίσεις <i>Connection Request Policies</i>	67
Σχ. 5.I <i>Report Event.sql</i>	70
Σχ. 5.II Στιγμιότυπο του πίνακα <i>event_data</i>	70
Σχ. 5.III Στιγμιότυπο του πίνακα <i>event_main</i>	71
Σχ. 5.IV Στιγμιότυπο του πίνακα <i>Subscriber</i>	71
Σχ. 5.V Στιγμιότυπο του πίνακα <i>Subscriber2</i>	71
Σχ. 5.VI Πίνακας συναρτησιακών εξαρτήσεων.....	72
Σχ. 5.VII Κλάσεις της εφαρμογής <i>Account++</i>	73
Σχ. 5.VIII Διάγραμμα <i>Use case</i>	77
Σχ. 5.IX UML Διάγραμμα κλάσεων	80
Σχ. 5.X Κεντρικό μενού εφαρμογής <i>Account++</i>	85
Σχ. 5.XI Επιλογή <i>Print</i> της εφαρμογής	85
Σχ. 5.XII Επιλογή <i>Pertimeperiod</i> της εφαρμογής	86
Σχ. 5.XIII Επιλογή <i>Peruser</i> της εφαρμογής.....	87
Σχ. 5.XIV Επιλογή <i>ExportData</i> της εφαρμογής	88
Σχ. 5.XV Επιλογή <i>Add User</i> της εφαρμογής	89
Σχ. 5.XVI Επιλογή <i>Change User</i> της εφαρμογής.....	90
Σχ. 5.XVII Επιλογή <i>Remove User</i> της εφαρμογής	90
Σχ. 5.XVIII Επιλογή <i>About</i> της εφαρμογής	91

1.Θεωρητικό υπόβαθρο πάνω στο 802.11 πρωτόκολλο

1.1 Εισαγωγή

Ο αιώνας που διανύουμε έχει χαρακτηριστεί αιώνας της πληροφορίας. Τομείς της οικονομίας οι οποίοι παλιότερα αποτελούσαν την κινητήρια δύναμη της ανάπτυξης όπως είναι η πρωτογενής και η δευτερογενής παραγωγή υποσκελίζονται από την τριτογενή παραγωγή (υπηρεσίες) η οποία αποτελεί πλέον το μεγαλύτερο μέρος της οικονομικής δραστηριότητας των περισσότερων ανεπτυγμένων χωρών.

Τα δίκτυα επικοινωνιών είναι το μέσο και εργαλείο ανάπτυξης της κοινωνίας της πληροφορίας. Σήμερα μπορούμε να επικοινωνούμε με ανθρώπους από όλον τον πλανήτη με κόστος μόνο την πρόσβαση στο διαδίκτυο. Διανύθηκαν νέες προοπτικές για το εμπόριο με την αναβίωση των καταστημάτων ταχυδρομικής αποστολής με τη μόνη διαφορά ότι η επιλογή των αγαθών δεν γίνεται πλέον μέσα από έναν κατάλογο που ταχυδρομήθηκε πριν ένα μήνα το λιγότερο, αλλά μέσα από μια σελίδα του διαδικτύου που ενημερώνεται το λιγότερο κάθε μέρα.

Η πρόοδος στον τομέα των τηλεπικοινωνιών και της ψηφιακής επεξεργασίας σήματος μας έχει επιτρέψει να διακινούμε δεδομένα με ρυθμούς Gbps (Δισεκατομύρια bit ανά δευτερόλεπτο) τόσο με ενσύρματες, όσο και με ασύρματες ζεύξεις. Η διαδικτύωση που συντελείται σήμερα μας επιτρέπει να ισχυριστούμε ότι σε μερικά χρόνια δεν θα υπάρχει αστική περιοχή χωρίς πρόσβαση στο διαδίκτυο. Για την καλύτερη κάλυψη των αναγκών των χρηστών σε ασύρματη επικοινωνία έχουν αναπτυχθεί διάφορα πρότυπα ασύρματης επικοινωνίας. Ενδεικτικά, αλλά όχι μόνο, έχουμε το Bluetooth, 802.11, GPRS/UMTS. Αυτα τα πρότυπα διαφέρουν στην περιοχή κάλυψης και στο ρυθμό μετάδοσης δεδομένων.

Στην παρούσα διπλωματική εργασία θα αναφερθούμε σε δίκτυα WLAN. Τα δίκτυα WLAN λειτουργούν σύμφωνα με το πρότυπο 802.11 και έχουν το πλεονέκτημα ότι συνδυάζουν υψηλές ταχύτητες πρόσβασης με ικανοποιητική εμβέλεια κάλυψης. Ένας αρκετά σημαντικός τομέας που θα επικεντρωθούμε είναι το AAA(authentication, authorization, accounting). Είναι δέοντως σημαντικό για ένα οποιοδήποτε δίκτυο επικοινωνιών να έχει τα μέσα για να πιστοποιεί τους χρήστες που πρέπει, να τους εξουσιοδοτεί με τα κατάλληλα δικαιώματα και να έχει στατιστικά στοιχεία ώστε να μπορεί να διενεργήσει στατιστική ανάλυση των δεδομένων και να εφαρμόσει μοντέλο χρέωσης.

1.2 Το πρωτόκολλο 802.11: Τι είναι το IEEE 802.11 πρωτόκολλο;

Το standard 802.11 είναι το πρώτο standard για WLAN και εώς τώρα το μοναδικό που βρίσκεται στην αγορά. Η υλοποίηση του standard ξεκίνησε το 1987 σαν μέρος του IEEE 802.4 token bus standard με τον αριθμό γκρουπ IEEE 802.4L. Το IEEE 802.4 πρωτόκολλο είναι το αντίστοιχο των IEEE 802.3 και IEEE 802.5 τα οποία έχουν σχεδιαστεί με γνώμονα το βιομηχανικό περιβάλλον. Ένα από τα βασικά κίνητρα για την ανάπτυξη των WLAN's ήταν για να χρησιμοποιηθούν από τη βιομηχανία στην επικοινωνία μεταξύ διαφόρων μηχανημάτων. Για αυτό το λόγο μεγάλες εταιρείες όπως η GM(General Motors) συμμετείχαν ενεργά στην ανάπτυξη του 802.4L ειδικά κατά τα πρώτα στάδια της ανάπτυξης του. Το 1990 η ομάδα εργασίας του 802.4L μετονομάστηκε σε IEEE 802.11 δημιουργώντας ένα ανεξάρτητο 802.11 standard ώστε να ορίσει το φυσικό στρώμα και το MAC στρώμα για WLAN's. Το πρώτο IEEE 802.11 standard για ταχύτητες 1 και 2 Mbps ολοκληρώθηκε το 1997 υποστηρίζοντας DSSS, FHSS και φυσικό

στρώμα διάχυτων υπέρυθρων ακτίνων (DFIR). Από την ολοκλήρωση αυτού του standard , καινούργιες υλοποιήσεις του φυσικού στρώματος που υποστηρίζουν 11Mbps χρησιμοποιώντας CCK (IEEE 802.11b) και 54Mbps χρησιμοποιώντας OFDM (IEEE 802.11a) έχουν υλοποιηθεί. Και οι τρεις αυτές εκδόσεις του 802.11 μοιράζονται το ίδιο στρώμα MAC που χρησιμοποιεί CSMA/CA για αποθήκευση δεδομένων , μηχανισμό αίτησης αποστολής / αποδεκτής αποστολής (RTS/CTS) για να προσπεράσουν το πρόβλημα κρυμμένου τερματικού και έναν προαιρετικό μηχανισμό που λέγεται λειτουργία συντονισμού σημείων (PCF) για να υποστηρίξει εφαρμογές ευαίσθητες στην απόκριση χρόνου. Το IEEE 802.11 standard υποστηρίζει τόσο WLAN's βασισμένα στη λογική του πελάτη – εξυπηρετητή όσο και ad hoc δίκτυα με ισότιμους peers.

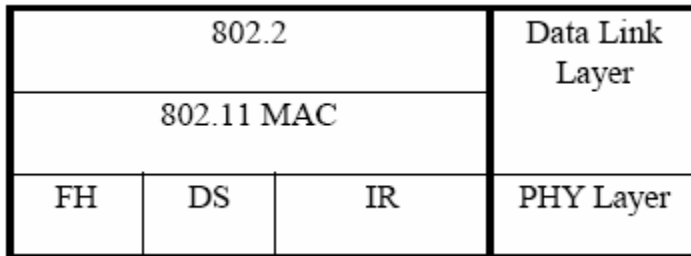
Το IEEE 802.11 standard ήταν το πρώτο standard ασύρματων δικτύων που είχε να αντιμετωπίσει την πρόκληση να ορίσει ένα συστηματικό standard για ασύρματα ευρυζωνικά δίκτυα. Σε σύγκριση με τα ενσύρματα δίκτυα τύπου LAN , τα WLAN's λειτουργούν κάτω από αντίξοες συνθήκες μέσου μεταφοράς (αέρας) και έχουν μεγάλες απαιτήσεις σε φορητότητα και ασφάλεια. Το ασύρματο μέσο μεταφοράς έχει μεγάλους περιορισμούς στο εύρος ζώνης και περιοριστικούς κανονισμούς όσον αφορά τις συχνότητες που μπορεί να χρησιμοποιήσει. Επιπλέον , έχει το πρόβλημα της παρεμβολής από ανακλάσεις του σήματος (multipath fading). Το WLAN υπόκειται σε παρεμβολές από άλλα γειτονικά WLAN's ή γενικά συσκευές ραδιοεπικοινωνίας ή μη (φούρνοι μικροκυμάτων , παλιά ασύρματα τηλέφωνα). Τα standard ασύρματης επικοινωνίας πρέπει να είναι σχεδιασμένα ώστε να υποστηρίζουν φορητότητα του χρήστη , χαρακτηριστικό που δεν υποστηρίζεται από κανένα άλλο standard τύπου LAN. Η ομάδα εργασίας του IEEE 802.11 έπρεπε να εξετάσει τη διαχείριση σύνδεσης , διαχείριση διασφάλισης σταθερότητας της σύνδεσης και διαχείριση εξοικονόμησης ενέργειας · κανένα από αυτά δεν υφίσταται σε κάποιο άλλο πρωτόκολλο της σειράς 802. Επίσης τα WLANs δεν έχουν φυσικά όρια (όριο του WLAN είναι τα σημεία όπου εξασθενεί τόσο το σήμα ώστε να είναι αδύνατη η πρόσβαση σε αυτό) και συνήθως επικαλύπτονται με γειτονικά WLANs έτσι η ομάδα που τυποποίησε τα standard έπρεπε να βρουν κάποιο τρόπο ώστε να οριστεί ένα ισχυρό επίπεδο ασφάλειας μεταξύ των ζεύξεων. Για όλους τους παραπάνω αναφερθέντες λόγους συν των διαφόρων ανταγωνιστικών standards χρειάστηκαν σχεδόν 10 χρόνια για την υλοποίηση του IEEE 802.11 χρόνος που είναι αρκετά μεγαλύτερος από αυτόν που απαιτήθηκε για άλλα standard τύπου 802 που σχεδιάστηκαν για ενσύρματα μέσα. Μόλις παρουσιάστηκε το γενικό πλαίσιο χρειάστηκε αρκετά μικρότερος χρόνος για να αναπτυχθούν οι επεκτάσεις IEEE 802.11b και 802.11a .

1.2.1 Προδιαγραφές 802.11

Το πρωτόκολλο 802.11 υποστηρίζει ρυθμούς μετάδοσης δεδομένων της τάξεως των 1Mbps και 2Mbps. Η μετάδοση του σήματος γίνεται είτε στην ISM ζώνη συχνοτήτων (2.4GHz – 2.4835GHz), είτε με υπέρυθρη ακτινοβολία μήκους κύματος 850nm. Για την μετάδοση του σήματος στην ISM ζώνη χρησιμοποιείται διαμόρφωση FSK 2 – επιπέδων για ρυθμούς 1Mbps και FSK 4 – επιπέδων για ρυθμούς 2Mbps. Για την επικοινωνία μέσω υπέρυθρων χρησιμοποιείται διαμόρφωση PPM (Pulse Position Modulation). Για μεγαλύτερη ανθεκτικότητα στον θόρυβο στενής ζώνης το σήμα κωδικοποιείται με μεθόδους απλωμένου φάσματος. Το πρωτόκολλο υποστηρίζει την μέθοδο εναλλαγής συχνότητας (FHSS) και ευθείας ακολουθίας (DSSS) για αυτό το σκοπό. Η μέγιστη εκπεμπόμενη ισχύς καθορίζεται από τους περιορισμούς που υπάρχουν για την χρήση της ISM ζώνης συχνοτήτων και περιορίζεται στα 20dBm ενώ η ευαισθησία του δέκτη, ορίζεται από το πρωτόκολλο, ότι πρέπει να είναι μικρότερη ή ίση των –80dBm για FER της τάξης του 3%.

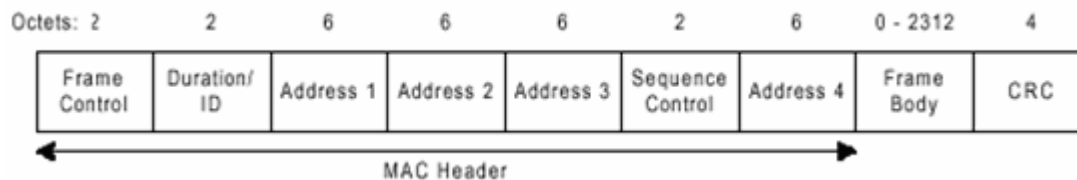
1.2.2 Πακέτο δεδομένων τύπου 802.11

Ένα πακέτο δεδομένων τύπου 802.11 βρίσκεται στο στρώμα ζεύξης του μοντέλου OSI. Η επιτροπή καθιέρωσης standards του IEEE 802 έχει ορίσει δύο διαφορετικά layers, το Logical Link Control (LLC) και το στρώμα media access control, για το στρώμα ζεύξης (Data-Link layer) του μοντέλου OSI. Το IEEE 802.11 standard ασύρματης δικτύωσης ορίζει τις προδιαγραφές για το φυσικό επίπεδο και το στρώμα (media access control (MAC)) το οποίο επικοινωνεί μέχρι το LLC επίπεδο.



Σχήμα 1.1 Στρώματα OSI του 802.11

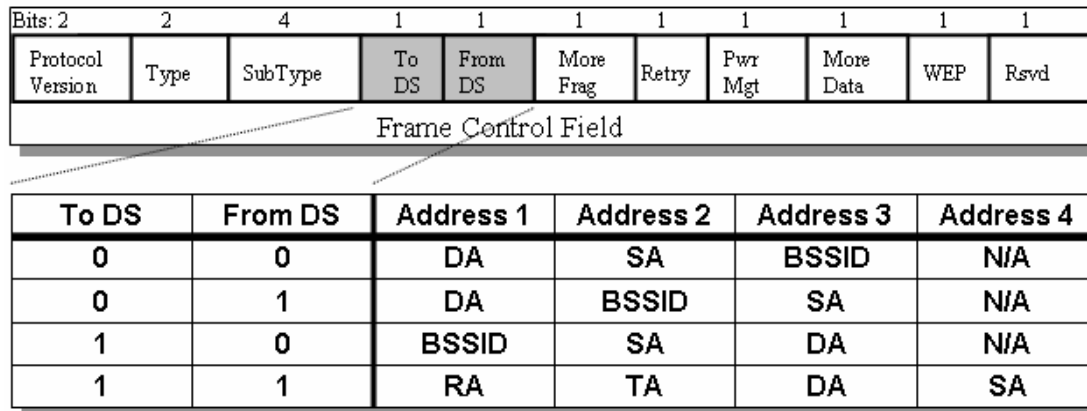
Όλα τα στοιχεία της αρχιτεκτονικής του 802.11 εμπίπτουν είτε στο υποστρώμα του MAC είτε στο φυσικό επίπεδο. Το πλαίσιο MAC του 802.11, αποτελείται από την επικεφαλίδα MAC, το σώμα του πλαισίου και ένα FCS(frame check sequence). Τα νούμερα στο παρακάτω σχήμα αντιπροσωπεύουν τον αριθμό των octets(bytes) σε κάθε πεδίο.



Σχ. 1.11 802.11 MAC Frame Format

Το πεδίο Frame Control, το οποίο φαίνεται στο επόμενο σχήμα περιέχει πληροφορίες ελέγχου που χρησιμοποιούνται για να οριστεί ο τύπος του 802.11 MAC πλαισίου και μπορεί να δώσει απαραίτητη στα επόμενα πεδία πληροφορία ώστε να χειριστούν σωστά το πλαίσιο MAC. Οι αριθμοί στο επόμενο σχήμα αναπαριστούν τον αριθμό των bits για κάθε πεδίο.

πεδίο Frame Control



Σχ. 1.ΙΙΙ Περιγραφές υποπεδίων του 802.11 πακέτου

Οι περιγραφές κάθε υποπεδίου του πεδίου Frame Control είναι οι ακόλουθες:

- **Protocol Version:** Προσδιορίζει την έκδοση του 802.11 πρωτοκόλλου που χρησιμοποιείται. Τα STA που λαμβάνουν το πλαίσιο χρησιμοποιούν αυτήν την τιμή για να προσδιορίσουν αν η έκδοση του πρωτοκόλλου με το οποίο είναι σύμφωνο το συγκεκριμένο πλαίσιο υποστηρίζεται από το STA.
- **Type and Subtype:** Ορίζει την λειτουργία του πλαισίου. Υπάρχουν τρία διαφορετικοί τύποι πεδίων πλαισίου: control, data και management. Υπάρχουν πολλαπλά υποπεδία για κάθε τύπο πλαισίου. Κάθε υποτύπος προσδιορίζει τη συγκεκριμένη λειτουργία που πρέπει να εκτελεστεί για τον τύπο πλαισίου.
- **To DS και From DS:** Ορίζει αν το πλαίσιο εισέρχεται ή εξέρχεται από ένα DS(distributed system) και χρησιμοποιείται μόνο σε πλαίσια τύπου δεδομένων STAs που συνεργάζονται με AP(Access Points).
- **More Fragments:** Ορίζει αν πρόκειται να ακολουθήσουν επιπλέον κομμάτια του πλαισίου είτε πλαίσια δεδομένων είτε πλαίσια τύπου management (διαχείρισης).
- **Retry:** Ορίζει αν ή όχι το πλαίσιο επαναμεταδίδεται, είτε για σκοπούς ακεραιότητας δεδομένων είτε για σκοπούς διαχείρισης.
- **Power Management:** Ορίζει αν το STA που στέλνει το πλαίσιο βρίσκεται ή μη σε κατάσταση ελαχιστοποίησης κατανάλωσης ενέργειας.
- **More Data:** Υποδεικνύει στο STA που βρίσκεται σε κατάσταση ελαχιστοποίησης κατανάλωσης ενέργειας ότι το AP έχει και άλλα πλαίσια να μεταδώσει. Επίσης, χρησιμοποιείται για APs ώστε να υποδείξει ότι επιπλέον broadcast/multicast πλαίσια πρόκειται να ακολουθήσουν.
- **WEP:** Υποδεικνύει αν χρησιμοποιείται πιστοποίηση και κρυπτογράφηση σε αυτό το πλαίσιο. Μπορεί να οριστεί στην ίδια τιμή για όλα τα πλαίσια δεδομένων ή διαχείρισης τα οποία έχουν τον κατάλληλο υποτύπο ορισμένο για πιστοποίηση.
- **Order:** Υποδεικνύει ότι όλα τα πλαίσια δεδομένων που πρόκειται να παραληφθούν πρέπει να επεξεργαστούν σύμφωνα με τη σειρά μετάδοσης.

Πεδίο Duration/ID

Αυτό το πεδίο χρησιμοποιείται για όλα τα πλαίσια τύπου ελέγχου εκτός από αυτά που περιέχουν τον υποτύπο Power Save (PS) Poll, ώστε να υποδείξουν την απαραίτητη χρονική διάρκεια που χρειάζεται μέχρι να παραληφθεί το επόμενο πλαίσιο. Όταν ο υποτύπος είναι PS Poll, το πεδίο περιέχει το association identity (AID) του STA που μεταδίδει.

Πεδία Address

Ανάλογα με τον τύπο του πλαισίου, τα τέσσερα πεδία διευθύνσεων (address) περιέχουν ένα συνδυασμό των ακόλουθων τύπων διευθύνσεων:

- **BSS Identifier (BSSID):** Το BSSID αναγνωρίζει μοναδικά κάθε BSS. Όταν το πλαίσιο προέρχεται από κάποιο STA από την υποδομή του BSS, το BSSID είναι η MAC διεύθυνση του AP. Όταν το πλαίσιο προέρχεται από κάποιο STA από την υποδομή ενός IBSS, το BSSID είναι η τυχαία παραγόμενη, τοπικά διαχειριζόμενη MAC διεύθυνση του STA που εκκίνησε το IBSS.
- **Destination Address (DA):** Το DA υποδεικνύει τη MAC διεύθυνση του τελικού προορισμού του πλαισίου.
- **Source Address (SA):** Το SA υποδεικνύει τη MAC διεύθυνση της αρχικής πηγής που δημιούργησε και μετέδωσε το πλαίσιο.
- **Receiver Address (RA):** Το RA υποδεικνύει τη MAC διεύθυνση του επόμενου STA πάνω στον ασύρματο δίαυλο μεταφοράς που πρόκειται να παραλάβει το πλαίσιο.

Transmitter Address (TA): Το TA υποδεικνύει τη MAC διεύθυνση του STA το οποίο μετέδωσε το πλαίσιο στον ασύρματο δίαυλο.

Πεδίο Sequence Control

Το πεδίο Sequence Control περιέχει δύο υποπεδία, το Fragment Number field και το Sequence Number. Το υποπεδίο Sequence Number υποδεικνύει τον αριθμό σειράς κάθε πλαισίου. Ο αριθμός σειράς είναι ο ίδιος για κάθε πλαίσιο το οποίο στέλνεται σαν μέρος ενός τεμαχισμένου πλαισίου αλλιώς ο αριθμός αυξάνεται κατά ένα μέχρι να φτάσει το 4095 και μετά ξεκινάει πάλι από το μηδέν.

Το υποπεδίο Fragment Number υποδεικνύει τον αριθμό σειράς κάθε πλαισίου το οποίο στέλνεται ως μέρος ενός τεμαχισμένου πλαισίου. Η αρχική τιμή είναι 0 και κατόπιν αυξάνεται κατά ένα για κάθε επακόλουθο πλαίσιο το οποίο στέλνεται ως μέρος του τεμαχισμένου πλαισίου.

Frame Body (Σώμα πλαισίου)

Το σώμα του πλαισίου περιέχει τα δεδομένα ή την απαιτούμενη πληροφορία είτε σε τύπο management (διαχείρισης), είτε σε πλαίσια τύπου δεδομένων.

Frame Check Sequence

Το μεταδίδων STA χρησιμοποιεί τον κυκλικό έλεγχο πλεονασμού (cyclic redundancy check (CRC)) για όλα τα πεδία της επικεφαλίδας MAC και το σώμα του πλαισίου ώστε να δημιουργήσει την τιμή FCS. Το λαμβάνων STA κατόπιν επαναλαμβάνει τον ίδιο υπολογισμό ώστε να υπολογίσει την δική του τιμή για το πεδίο FCS και κατόπιν με τη μέθοδο της σύγκρισης να διαπιστώσει εάν το πλαίσιο αλλοιώθηκε κατά τη μετάδοση.

1.3 Παραλλαγές του 802.11

802.11a – OFDM in 5GHz Band

Το 802.11a αποτελεί ένα πρωτόκολλο για το φυσικό επίπεδο ενός ασυρμάτου δικτύου το οποίο καθορίζει την λειτουργία αυτού στην ζώνη UNII των 5GHz. Χρησιμοποιείται διαμόρφωση ορθογώνιας διαίρεσης συχνότητας (OFDM) και υποστηρίζει ρυθμούς μετάδοσης μέχρι και 54Mbps. Λόγω της λειτουργίας του στην ζώνη UNII όπου υπάρχουν πολύ λιγότερες παρεμβολές από την ζώνη ISM και του υψηλότερου ρυθμού μετάδοσης προσφέρει πολύ καλύτερες επιδόσεις τόσο από το κλασικό 802.11 όσο και από το νεότερο και ευρύτερα εξαπλωμένο 802.11b.

802.11b – High Rate DSSS

Το task group B είχε αναλάβει την εργασία, να επεκτείνει τον τρόπο κωδικοποίησης DSSS του φυσικού επιπέδου του 802.11 ώστε να υποστηρίζει ρυθμούς μετάδοσης της τάξης των 5.5Mbps και 11Mbps. Για να το πετύχει αυτό, τροποποιήθηκε ο τρόπος διαμόρφωσης του σήματος. Έτσι για την επίτευξη των νέων ρυθμών χρησιμοποιήθηκε διαμόρφωση CCK, ενώ για τους ρυθμούς των 1Mbps και 2Mbps, ώστε να κρατηθεί η συμβατότητα με το 802.11, χρησιμοποιήθηκε διαμόρφωση DBPSK (Differential Binary) και DQPSK (Differential Quadratic) αντίστοιχα.

802.11c – Bridge Op Procedures

Το 802.11c παρέχει απαραίτητες πληροφορίες για να διασφαλιστεί η σωστή λειτουργία των bridges. Οι πληροφορίες που περιέχονται σε αυτό το πρωτόκολλο χρησιμοποιούνται κυρίως από τους κατασκευαστές σημείων πρόσβασης ώστε να εξασφαλίζεται η διαλειτουργικότητά τους με συσκευές άλλων κατασκευαστών.

802.11d – Global Harmonization

Το task group D έχει αναλάβει την εργασία να καθορίσει τις απαιτήσεις του φυσικού επιπέδου καθώς και να καταγράψει το νομικό πλαίσιο που ισχύει για την χρησιμοποίηση ραδιοσυχνοτήτων σε διάφορες χώρες ώστε να μπορούν να κατασκευαστούν προϊόντα που θα λειτουργούν σε διάφορες γεωγραφικές περιοχές.

802.11e – MAC Enhancements for QoS

Χωρίς καλό QoS (Quality of Service) το αρχικό πρωτόκολλο 802.11 δεν βελτιστοποιεί την μετάδοση φωνής και video. Αυτό ακριβώς το μειονέκτημα έρχεται να καλύψει το task group E τροποποιώντας το υποεπίπεδο MAC και βελτιώνοντας το QoS του πρωτοκόλλου.

802.11f – Inter Access Point Protocol

Η αρχική ομάδα εργασίας του 802.11 σκοπίμως δεν προσδιορίζει την επικοινωνία μεταξύ σημείων πρόσβασης με σκοπό την υποστήριξη της περιαγωγής των χρηστών από ένα σημείο πρόσβασης σε ένα άλλο. Η επιλογή αυτή δίνει ευελιξία όταν χρησιμοποιούνται διάφορα distribution system. Το πρόβλημα, όμως που ανακύπτει είναι ότι τα σημεία πρόσβασης από διαφορετικούς κατασκευαστές μπορεί να μην λειτουργούν ομαλά μεταξύ τους όταν υποστηρίζουν λειτουργίες περιαγωγής. Το 802.11f έρχεται ακριβώς σε αυτό το σημείο, να φτιάξει μια προδιαγραφή που θα παρέχει στα σημεία πρόσβασης της απαραίτητες πληροφορίες για να γίνει μια περιαγωγή με επιτυχία και να εξασφαλιστεί η ομαλή λειτουργία του συστήματος.

802.11g – Union of .11a and .11b

Η παραλλαγή αυτή του 802.11 έχει ως αντικείμενο εργασίας να προσφέρει ρυθμούς μετάδοσης της τάξης των 54Mbps, όπως και το 802.11a διατηρώντας όμως την συμβατότητα με το διαδεδομένο 802.11b. Λειτουργεί στην ISM ζώνη συχνοτήτων όπως το 802.11b αλλά χρησιμοποιεί διαμόρφωση OFDM όπως το 802.11a για να πετύχει υψηλούς ρυθμούς μετάδοσης. Χάριν συμβατότητας με το 802.11b υποστηρίζεται και η διαμόρφωση CCK.

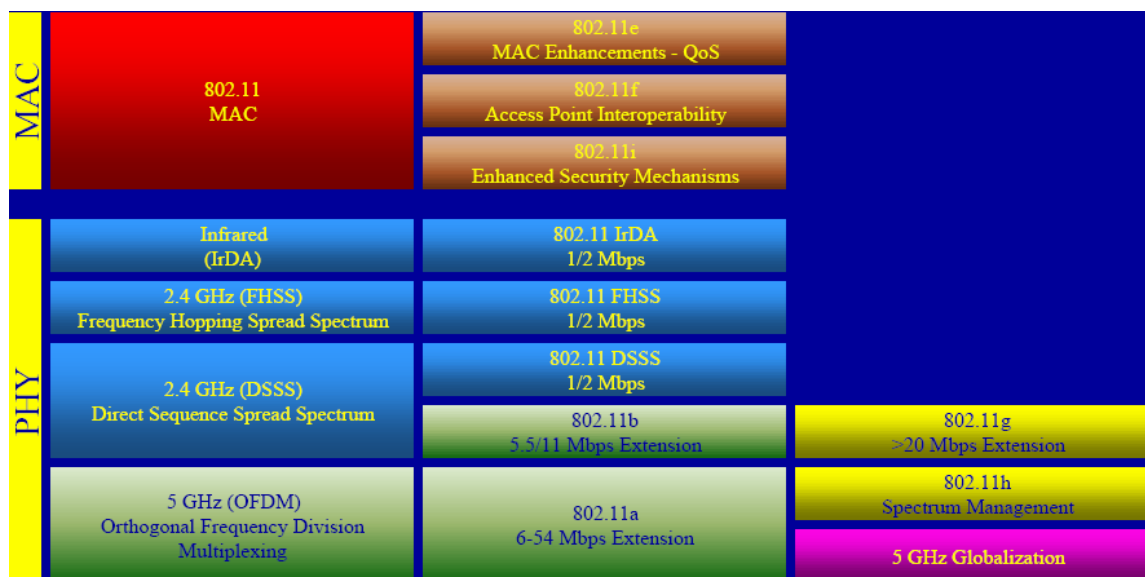
802.11h – UNII for Europe

Η προδιαγραφή αυτή είναι συμπληρωματική του υποεπιπέδου MAC και συμμορφώνεται με τους ευρωπαϊκούς κανονισμούς για την χρήση της ζώνης συχνοτήτων στα 5GHz. Συγκεκριμένα οι ευρωπαϊκοί κανονισμοί απαιτούν για τις συσκευές που λειτουργούν σε αυτή την ζώνη συχνοτήτων να έχουν δυνατότητες ελέγχου εκπεμπόμενης ισχύος (Transmission Power Control) και δυναμικής επιλογής συχνότητας (Dynamic Frequency Selection).

802.11i – Enhanced Security

Η προδιαγραφή αυτή έρχεται να καλύψει πολλά από τα κενά σε θέματα ασφαλείας που βρέθηκαν στο πρωτόκολλο κρυπτογράφησης WEP του 802.11. Ο αλγόριθμος RC4 της RCA που χρησιμοποιείται αποδείχτηκε ανεπαρκής, με πολλά σφάλματα και παραλήψεις, κάνοντας τα ασύρματα δίκτυα εύκολο στόχο σε διάφορα είδη επιθέσεων. Με την νέα προδιαγραφή καθορίζονται πρωτόκολλα για τα κλειδιά κρυπτογράφησης όπως τα TKIP (Temporal Key Integrity Protocol) και AES (Advanced Encryption Standard).

Στο παρακάτω σχήμα παρουσιάζονται μερικές από τις παραλλαγές του 802.11 σε σχέση με την λειτουργία τους και την θέση τους στο μοντέλο αναφοράς OSI.



Σχ. 1.IV Οι διάφορες παραλλαγές του 802.11 πρωτοκόλλου

1.4 Τα κυριότερα εμπορικά πρωτόκολλα σήμερα:

Σήμερα στην αγορά κυριαρχούν υλοποιήσεις βασισμένες στα 802.11a , 802.11b και 802.11g . Οι τεχνικές προδιαγραφές τους επισημαίνονται στον παρακάτω πίνακα:

Table 1: IEEE 802.11 Specifications

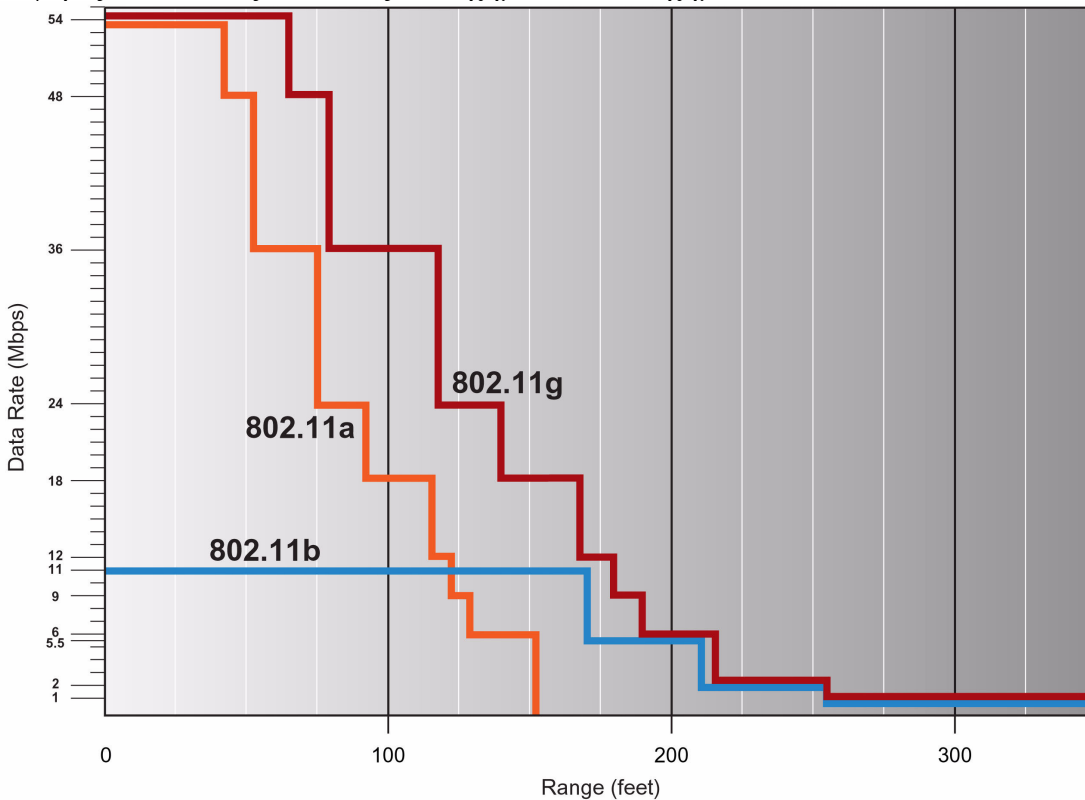
	802.11b	802.11a	802.11g
Standard approved	July 1999	July 1999	June 2003
Maximum data rate	11 Mbps	54 Mbps	54 Mbps
Modulation	CCK	OFDM	OFDM and CCK
Data rates	1, 2, 5.5, 11 Mbps	6, 9, 12, 18, 24, 36, 48, 54 Mbps	CCK: 1, 2, 5.5, 11 Mbps OFDM: 6, 9, 12, 18, 24, 36, 48, 54 Mbps
Frequencies	2.4–2.497 GHz	5.15–5.35 GHz 5.425–5.675 GHz	2.4–2.497 GHz

Σχ. 1.V Παραλλαγές του 802.11

Το πρωτόκολλο 802.11 υποστηρίζει διάφορους ρυθμούς ταχυτήτων ώστε να επιτρέπει στους χρήστες να επικοινωνούν με την καλύτερη δυνατή ταχύτητα. Η επιλογή ρυθμού ταχύτητας είναι μια χρυσή τομή μεταξύ της λήψης του υψηλότερου δυνατού ρυθμού ταχύτητας προσπαθώντας να ελαχιστοποιήσει τον αριθμό λαθών επικοινωνίας. Όταν υπάρχει λάθος στα απεσταλθέντα δεδομένα, το σύστημα ασύρματης επικοινωνίας πρέπει να ξοδεύει χρόνο ώστε να αναμεταδοθούν τα στοιχεία έως ότου μεταδοθούν επιτυχώς. Η κάρτα πρόσβασης του κάθε χρήστη 802.11 μόλις συνδεθεί με το σημείο πρόσβασης ακολουθεί μια διαδικασία για να επιλέξει το καλύτερο ποσοστό στοιχείων. Οι χρήστες του πρωτόκολλου 802.11g μπορούν να επιλέξουν ρυθμό μετάδοσης από τους διαθέσιμους με διαμόρφωση OFDM 54 ..48 ..36 ..24 ..18 ..12 ..9, και 6 Mbps, και τους ρυθμούς μετάδοσης με διαμόρφωση CCK 11, 5,5, 2, και 1 Mbps

Όσο η απόσταση από το σημείο πρόσβασης αυξάνεται, τα βασισμένα στο 802.11 προϊόντα προσαρμόζουν προς τα κάτω την ταχύτητα σύνδεσης για να διατηρήσουν τη σύνδεση. Το πρότυπο 802.11g έχει τα ίδια χαρακτηριστικά διάδοσης με το 802.11b, επειδή εκπέμπει στην ίδια ζώνη συχνότητας των 2.4-GHz. Επειδή τα προϊόντα 802.11b και 802.11g έχουν τα ίδια χαρακτηριστικά διάδοσης, οι υλοποιήσεις καρτών ασύρματης πρόσβασης και σημείων πρόσβασης παρέχουν κατά προσέγγιση την ίδια μέγιστη ακτίνα ραδιοκάλυψης στην ίδια ταχύτητα μεταφοράς. Επειδή τα ραδιοκύματα στα 5- GHz δεν έχουν τα ίδια χαρακτηριστικά διάδοσης όπως στα 2.4-GHz, η ακτίνα ραδιοκάλυψης των προϊόντων βασισμένων στο 802.11a είναι περιορισμένη συγκριτικά με όσα είναι βασισμένα στο 802.11b ή 802.11g.

Η απόσταση του τερματικού από το σημείο πρόσβασης (access point) επηρεάζει σημαντικά την αναμενόμενη ρυθμαπόδοση (throughput) της σύνδεσης. Η θεωρητική ρυθμαπόδοση για διάφορες αποστάσεις απεικονίζεται σχηματικά στο σχήμα 1.VI:



Σχ. 1.VI Ρυθμαπόδοση των παραλλαγών του 802.11 σε συνάρτηση με την απόσταση

1.5 Διαμόρφωση σήματος στα κυριότερα πρωτόκολλα:

1.5.1 Διαμόρφωση σήματος στο 802.11b:

Όπως φαίνεται και στον πίνακα X το πρωτόκολλο 802.11b χρησιμοποιεί διαμόρφωση CCK(Complementary Code Keying) για τη μετάδοση της πληροφορίας.

Η διαμόρφωση CCK αποτελείται από ένα σύνολο 64 οκτάμπιτων λέξεων κώδικα που χρησιμοποιούνται για να κωδικοποιήσουν τα στοιχεία ώστε να πετύχουν ταχύτητες μεταφοράς δεδομένων της τάξης των 5.5 ή 11Mbps στο φάσμα συχνοτήτων των 2.4GHz . Οι λέξεις κώδικα έχουν κάποιες μοναδικές μαθηματικές ιδιότητες που τους επιτρέπουν να διακρίνονται μεταξύ τους από τον δέκτη ακόμα και όταν υπάρχει θόρυβος και πολυδιαδρομική παρέμβολή. Η διαμόρφωση CCK λειτουργεί μόνο σε συνεργασία με την τεχνολογία DSSS(direct-sequence spread spectrum) που συμπεριλαμβάνεται στο πρότυπο του 802.11. Το CCK εφαρμόζει περίπλοκους μαθηματικούς τύπους στην κωδικοποίηση του DSSS, που επιτρέπει την αποστολή μεγαλύτερου όγκου πληροφοριών ανά κύκλο ρολογιού. Πρακτικά δηλαδή η συσκευή αποστολής ραδιοσημάτων μπορεί να εφαρμόσει πολυπλεξία στην πληροφορία που αποστέλεται με κάθε DSSS κώδικα, ώστε να επιτύχει ταχύτητα μετάδοσης ίση με 11Mbps αντί για 2Mbps που καθορίζονταν στα αρχικά πρότυπα.

Αναλυτικότερα , η κωδικοποίηση DSSS χρησιμοποιείται για το σχήμα διαμόρφωσης υψηλού ρυθμού οπότε οι συμπληρωματικοί κώδικες του CCK αναφέρονται σαν κώδικες εξάπλωσης (spreading codes) επειδή χρησιμοποιούνται για να εξαπλώσουν το απασχολούμενο εύρος ζώνης της κυματομορφής τύπου DSSS.

Οι συμπληρωματικοί κώδικες εξάπλωσης που ορίζονται από το IEEE 802.11 έχουν μήκος κώδικα 8 και ρυθμό chipping 11Mchips/sec. Είναι γνωστό ότι 8 σύνθετα chips απαρτίζουν ένα σύμβολο. Ορίζοντας τον ρυθμό συμβόλων στα 1.375MSymbols/sec η κυματομορφή των 11Mbps καταλαμβάνει περίπου το ίδιο εύρος ζώνης όπως η κυματομορφή των 2Mbps με διαμόρφωση QPSK στο πρότυπο 802.11 επιτρέποντας έτσι τρία μη επικαλυπτόμενα κανάλια στην μπάντα συχνοτήτων ISM. Αυτό είναι σημαντικό για τη μεγιστοποίηση της ρυθμαπόδοσης στα δίκτυα WLAN και είναι ένας από τους λόγους που επιλέχθηκε η διαμόρφωση τύπου CCK.

Οι οκτάμπιτες κωδικές λέξεις σχηματίζονται σύμφωνα με την ακόλουθη φόρμουλα:

$$c = \left\{ e^{j(\varphi_1 + \varphi_2 + \varphi_3 + \varphi_4)}, e^{j(\varphi_1 + \varphi_3 + \varphi_4)}, e^{j(\varphi_1 + \varphi_2 + \varphi_4)}, \right. \\ \left. -e^{j(\varphi_1 + \varphi_4)}, e^{j(\varphi_1 + \varphi_2 + \varphi_3)}, e^{j(\varphi_1 - \varphi_3)}, -e^{j(\varphi_1 + \varphi_2)}, e^{j\varphi_1} \right\} \quad (\text{EQ. 2})$$

Όπου c είναι η κωδική λέξη με πρώτο το LSB και τελευταίο το MSB(Least και Most Significant Bit αντίστοιχα). Αυτή η φόρμουλα χρησιμοποιείται για να παραγάγει τα σύνολα κωδικών λέξεων για ρυθμούς μετάδοσης τόσο 11Mbps όσο και 5.5Mbps , δηλαδή ένα υποσύνολο των κωδικών λέξεων για τα 11Mbps χρησιμοποιείται για τα 5.5Mbps. Οι παράμετροι $\varphi_1 \dots \varphi_4$ μας δείχνουν τη φάση των μιγαδικών συνόλων κωδικών λέξεων και ορίζονται από το standard 802.11 . Για ρυθμό μετάδοσης 11Mbps κάθε σύμβολο αναπαριστά 8 bit πληροφορίας ενώ για 5.5Mbps μεταδίδονται 4bit ανά σύμβολο. Τώρα μπορούμε να δούμε συνοπτικά τη μετάδοση για ρυθμό 11Mbps. Καταρχάς εισάγουμε μία σειριακή ακολουθία bit σε κάποιον επεξεργαστή σήματος βασικής ζώνης. Η ροή των bits χωρίζεται σε bytes σύμφωνα με τη σειρά (d7 , d6 , d5 ... d0) όπου d0 είναι το LSB και το πρώτο χρονικά. Τα 8 bits που προέκυψαν χρησιμοποιούνται για να κωδικοποιήσουμε τις παραμέτρους φάσης $\varphi_1 \dots \varphi_4$ σύμφωνα με το σχήμα 1.VII. Η κωδικοποίηση βασίζεται σε διαφορεική διαμόρφωση τύπου QPSK όπως ορίζεται στο σχήμα 1.VIII.

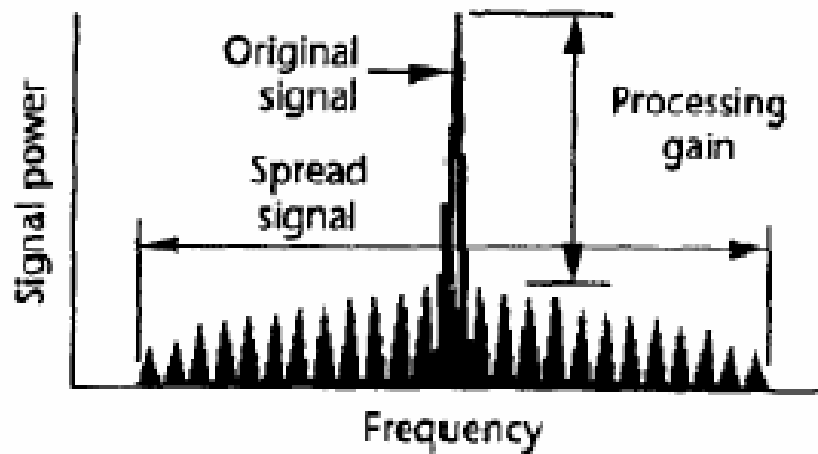
DIBIT	PHASE PARAMETER
(d1, d0)	φ_1
(d3, d2)	φ_2
(d5, d4)	φ_3
(d7, d6)	φ_4

Σχ. 1.VII Κωδικοποίηση παραμέτρων φάσεων στο 802.11b

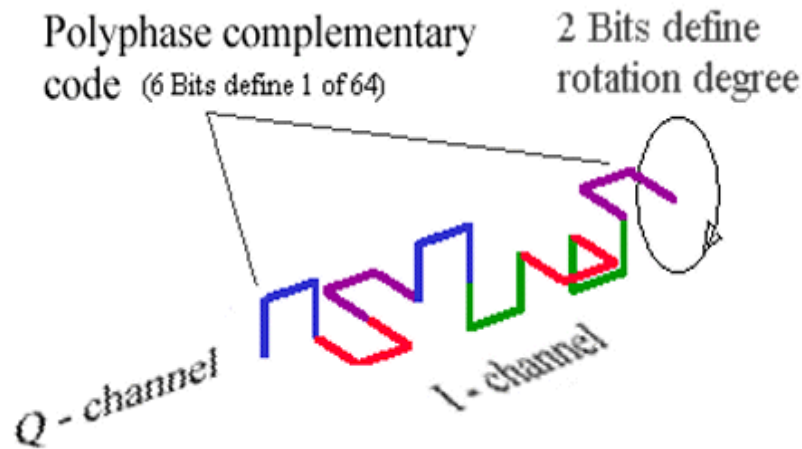
DIBIT (d_{i+1}, d_i)	PHASE
00	0
01	π
10	$\pi/2$
11	$-\pi/2$

Σχ. 1.VIII Διαφορική διαμόρφωση QPSK

Με αυτόν τον τρόπο έχουμε τη μιγαδική κωδική λέξη που χρησιμοποιεί ο επεξεργαστής σήματος βασικής ζώνης ώστε να διαμορφώσει το φέρον και να εξαπλώσει το εύρος ζώνης της κυματομορφής. Το τελικό και το αρχικό σχήμα φαίνονται στο σχήμα 1.IX όπου βλέπουμε επίσης και το κέρδος επεξεργασίας του σήματος.

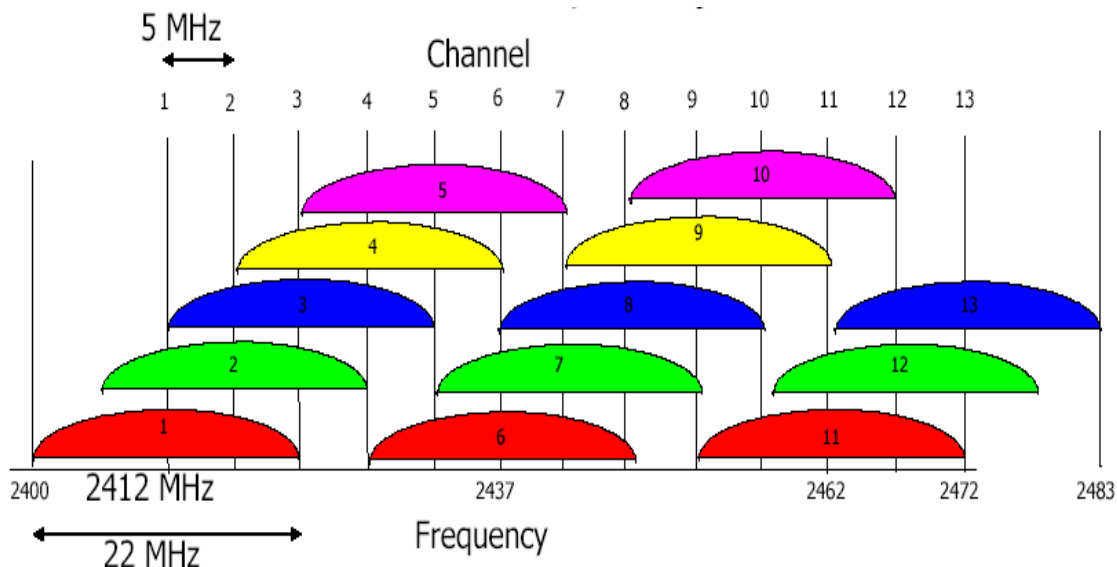


Σχ. 1.IX Απόκριση συχνότητας, εξάπλωση φάσματος



Η τεχνική εξάπλωσης φάσματος DSSS που χρησιμοποιείται για την μετάδοση δεδομένων στο 802.11b συνοπτικά ορίζεται σαν τη λειτουργία του διαμορφωτή ραδιοσήματος με ένα από τα 14 δυνατά φέροντα το οποίο διαμορφώνεται κατά DQPSK με εύρος φάσματος 22MHz.

Σε επίπεδο συχνοτήτων έχουμε 13 κανάλια(Europe) ή τα πρώτα 11 κανάλια(United States) από 2412MHz έως τα 2472MHz. Όπως παρατηρούμε μόνο 3 (U.S. 1,6,11) ή 4 (E.U. 1,5,9,13) κανάλια είναι μη επικαλυπτόμενα ενώ το εύρος φάσματος κάθε καναλιού επικοινωνίας είναι 22MHz.



Σχ. 1.X Κανάλια επικοινωνίας στο 802.11b

1.5.2 Διαμόρφωση σήματος στο 802.11a:

Όπως φαίνεται στον πίνακα X το πρωτόκολλο 802.11a χρησιμοποιεί διαμόρφωση OFDM(Orthogonal Frequency Division Multiplexing) για τη μετάδοση της πληροφορίας. Η διαμόρφωση τύπου OFDM λειτουργεί χωρίζοντας το ραδιοσήμα σε πολλά μικρότερα υποσήματα

τα οποία μεταδίδονται ταυτόχρονα σε διαφορετικές συχνότητες στον αποδέκτη. Η διαμόρφωση OFDM έχει το πλεονέκτημα ότι ελαττώνει την ομοδιαυλική παρεμβολή η οποία στον αποδέκτη φαίνεται σαν θόρυβος.

Η διαμόρφωση που χρησιμοποιείται στο πρότυπο IEEE 802.11a είναι η OFDM (Orthogonal Frequency Division Multiplexing), που μοιάζει πολύ με τη διαμόρφωση που χρησιμοποιείται στις γραμμές ADSL (Asymmetrical Digital Subscriber Loop) (βλ. την ITU G.992.1 και G.992.2) και αποκαλείται DMT (Discrete Multi Tone), όπου το σύστημα μετάδοσης στέλνει παράλληλα διάφορα χρησιμοποιώντας τον αντίστροφο γρήγορο μετασχηματισμό Fourier (IFFT), και που λαμβάνουν τα υποφέροντα χρησιμοποιώντας το γρήγορο μετασχηματισμό Fourier (FFT).

Τα πρότυπα IEEE 802.11a ορίζει το φυσικό στρώμα OFDM (PHY) που χωρίζει το σήμα πληροφορίας σε 52 ξεχωριστά υποφέροντα για να παράσχει μεταφορά δεδομένων σε ρυθμούς 6, 9, 12, 18, 24, 36, 48, ή 54 Mbps. Οι ρυθμοί μεταφοράς 6, 12, και 24 Mbps είναι υποχρεωτικό να υλοποιούνται από τις συσκευές που ακολουθούν το πρωτόκολλο 802.11a. Τέσσερα από τα υποφέροντα είναι πειραματικά υποφέροντα που το σύστημα χρησιμοποιεί ως αναφορά για να αγνοήσει μετατοπίσεις συχνότητας ή φάσης του σήματος κατά τη διάρκεια της μετάδοσης.

Στο πρότυπο 802.11a, μια ψευδο δυαδική ακολουθία στέλνεται μέσω των πειραματικών υποκαναλιών για να αποτρέψει την δημιουργία spectral lines (φασματικών γραμμών). Στο 802.11a, τα υπόλοιπα 48 υποφέροντα είναι ξεχωριστές ασύρματες διαδρομές για την παράλληλη αποστολή των πληροφοριών. Το προκύπτον διάστημα συχνότητας των υπο-φερόντων είναι 0.3125 MHz (για 20 MHz ανά κανάλι με 64 δυνατές σχισμές συχνότητας υπο-φερόντων).

Το στρώμα OFDM PHY αποτελείται από δύο λειτουργίες πρωτοκόλλου: καταρχήν μια λειτουργία σύγκλισης PHY, η οποία προσαρμόζει τις ικανότητες του Physical Medium Dependent (PMD) συστήματος στην υπηρεσία PHY. Αυτή η λειτουργία υποστηρίζεται από τη φυσική διαδικασία σύγκλισης στρώματος (PLCP), η οποία καθορίζει μια μέθοδο για να αντιστοιχίσουμε τις IEEE 802.11 PHY Sublayer Service Data Units (PSDU) σε ένα πλαίσιο κατάλληλο για να στείλουμε και να λάβουμε δεδομένα των χρηστών και πληροφορίες διαχείρισης μεταξύ δύο ή περισσότερων σταθμών χρησιμοποιώντας το σχετικό σύστημα PMD. Τέλος, ένα σύστημα PMD του οποίου η λειτουργία καθορίζει τα χαρακτηριστικά και τη μέθοδο αποστολής και λήψης δεδομένων μέσω ενός ασύρματου μέσου μεταξύ δύο ή περισσότερων σταθμών εκ των οποίων κάθε ένας χρησιμοποιεί το σύστημα OFDM.

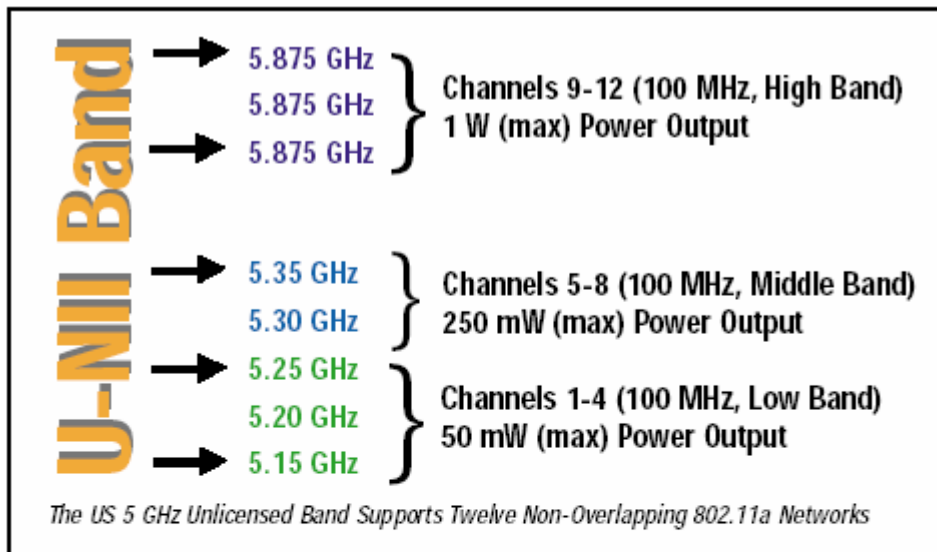
Ο διαμορφωτής τύπου 802.11a μετατρέπει το δυαδικό σήμα σε αναλογική κυματομορφή μέσω της χρήσης διαφόρων τύπων διαμόρφωσης, ανάλογα με το ποιος ρυθμός έχει επιλεγεί. Παραδείγματος χάριν για ρυθμό μετάδοσης 6 Mbps, το PMD χρησιμοποιεί τη binary phase shift keying (BPSK, δυαδική διαμόρφωση μετατόπισης φάσης), η οποία μετατοπίζει τη φάση της κεντρικής συχνότητας μετάδοσης για να αντιπροσωπεύσει διάφορα πρότυπα bits δεδομένων. Οι υψηλότεροι ρυθμοί μετάδοσης, όπως 54 Mbps, υιοθετούν τη quadrature amplitude modulation (QAM) για να αντιπροσωπεύσουν bits δεδομένων με το να μεταβάλλουν την κεντρική συχνότητα μετάδοσης με διαφορετικά επίπεδα εύρους ζώνης μετάδοσης και να εκτελούν μετατόπιση φάσης. Όλα αυτά φαίνονται παραστατικά στο σχήμα 1.XI.

Data Rate (Mbps)	Modulation	Coding Rate	Coded bits per subcarrier	Coded bits per OFDM symbol	Data bits per OFDM symbol
6	BPSK	1/2	1	48	24
9	BPSK	3/4	1	48	36
12	QPSK	1/2	2	96	48

18	QPSK	3/4	2	96	72
24	16-QAM	1/2	4	192	96
36	16-QAM	3/4	4	192	144
48	16-QAM	2/3	6	288	192
54	64-QAM	3/4	6	288	216

Σχ. 1.XI Διαμόρφωση στο 802.11a

Τα διαθέσιμα κανάλια επικοινωνίας φαίνονται στο σχήμα 1.XI. Παρατηρούμε ότι υπάρχουν διαφορές στα κανάλια ανάλογα με τη χώρα και τους κανονισμούς εκπομπής που έχει υιοθετήσει για τις διάφορες συχνότητες. Η μπάνα συχνότητων που χρησιμοποιεί το 802.11a είναι όπως και στο 802.11b ελεύθερη προς χρήση από το κοινό με περιορισμούς όμως ως προς την ισχύ εκπομπής. Η μέγιστη ισχύς εκπομπής καθώς και οι μπάνες συχνότητων ανάλογα με τη χώρα φαίνονται στο ακόλουθο σχήμα:



5 GHz Frequency Band	US and Canada	Europe*	France*	Spain *	Japan
5.150 – 5.250	40 mW	200 mW	200 mW	200 mW	200 mW
5.250 – 5.350	200 mW	200 mW	200 mW	200 mW	
5.470 – 5.725		1 W			
5.725 – 5.875	1 W				

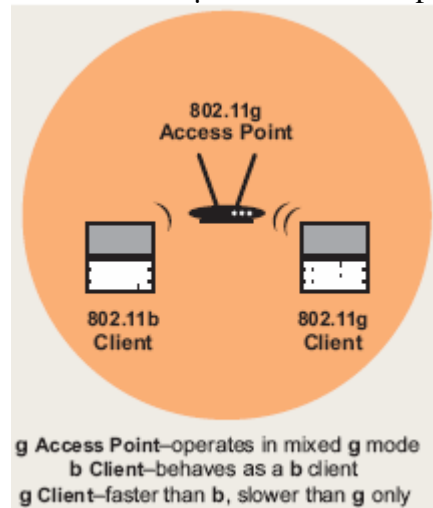
*Note: * Not available to 802.11a compliant devices*

Σχ. 1.XII Κανάλια και ισχύς εκπομπής στο 802.11a

1.5.3 Διαμόρφωση σήματος στο 802.11g:

Το πρωτόκολλο 802.11g χρησιμοποιεί συνδυασμό των τεχνικών OFDM και CCK για να επιτύχει προς τα πίσω συμβατότητα και μεγάλες ταχύτητες μεταφοράς δεδομένων. Έτσι όταν το access point λειτουργεί με χρήστες εφοδιασμένους με κάρτες 802.11g χρησιμοποιεί διαμόρφωση OFDM στην ζώνη συχνοτήτων των 2.4GHz ενώ όταν οι κάρτες των χρηστών υποστηρίζουν μόνο το 802.11b μεταπηδά σε διαμόρφωση CCK ώστε να μπορεί να επικοινωνήσει με τους χρήστες σύμφωνα με το πρωτόκολλο 802.11b.

Η προς τα πίσω συμβατότητα που παρέχει το πρωτόκολλο 802.11g έχει αντίκτυπο στη ρυθμαπόδοση όταν υπάρχουν κινητά τερματικά με 802.11g και 802.11b κάρτες πρόσβασης που επικοινωνούν με το ίδιο access point.



Σχ. 1.XIII Λειτουργία σε μικτό περιβάλλον b/g

Όταν υπάρχει περιβάλλον όπως στο Σχήμα 1.XIII το access point καταλαβαίνει ότι υπάρχουν χρήστες και των δύο πρωτοκόλλων και διατάζει τις κάρτες ασύρματης πρόσβασης με τεχνολογία 802.11g να χρησιμοποιήσουν έναν μηχανισμό προστασίας από παρεμβολές των ραδιοσημάτων των χρηστών με τεχνολογία 802.11b.

Με αυτόν τον τρόπο επιτυγχάνεται απρόσκοπτη επικοινωνία μεταξύ των χρηστών και του access point με αντίκτυπο όμως στην ταχύτητα επικοινωνίας. Πειραματικά δεδομένα μας έχουν δείξει ότι η ρυθμαπόδοση σε περιβάλλον μεικτών χρηστών τεχνολογίας 802.11b και 802.11g πέφτει στα 15Mbps περίπου που και πάλι όμως είναι μεγαλύτερη από αυτήν των χρηστών 802.11b που επιτυγχάνουν ρυθμαπόδοση της τάξης των 5-6 Mbps σε πραγματικές συνθήκες. Επίσης από μετρήσεις γνωρίζουμε ότι η μέγιστη ρυθμαπόδοση σε πραγματικές συνθήκες λειτουργίας για access points βασισμένα σε τεχνολογία 802.11g δεν ξεπερνάει τα 24-25Mbps, τιμή που διαφέρει αισθητά από το θεωρητικό μέγιστο που είναι τα 54Mbps.

1.6 Το μέλλον είναι εδώ

Προτρέχοντας ελαφρά μπορούμε να διακρίνουμε τρεις άξονες στην ανάπτυξη των ασύρματων δικτύων (WLAN's).

1) Η βελτίωση της ασφάλειας των δικτύων αποτρέποντας επιθέσεις DDOS (Distributed Denial of Service attacks) και υποκλοπής των διακινούμενων δεδομένων. Οι πρώτες μπορούν να γίνουν από κινητά τερματικά που προσπαθούν να «φιμώσουν» το σημείο πρόσβασης (access point). Η υποκλοπή των διακινούμενων δεδομένων γίνεται μέσω εργαλείων sniffing που «συλλαμβάνουν» τα πακέτα άλλων χρηστών ενώ βρίσκονται στον αέρα και κατόπιν

αποκρυπτογραφώντας τα ώστε να αποκαλύψουν τυχόν ευαίσθητες πληροφορίες χρηστών (αριθμοί πιστωτικών καρτών , κωδικοί πρόσβασης κ.α.)

2) Η αύξηση της μέσης και μέγιστης ταχύτητας των ασύρματων δικτύων μέσω τεχνικών δέσμευσης μεγαλύτερου εύρους φάσματος και χρήσης έξυπνων κεραιών. Τα κυριότερα εμπόδια στην προσπάθεια για αύξηση της ταχύτητας των ασύρματων δικτύων που βασίζονται στην τεχνολογία 802.11 είναι το πεπερασμένο εύρος φάσματος και η διακαναλική παρεμβολή.

Το πεπερασμένο εύρος φάσματος αποτελεί τον κυριότερο ανασταλτικό παράγοντα για την βελτίωση του bandwidth των ασύρματων επικοινωνιακών δικτύων. Τα προβλήματα που δημιουργεί αφορούν θεωρητικά άνω όρια στην μετάδοση δεδομένων με συγκεκριμένη ισχύ εκπομπής/λήψης και για κάποια συγκεκριμένη κεραία εκπομπής/λήψης. Αυτοί οι περιορισμοί του διαθέσιμου φάσματος πρέπει να υπερκεραστούν με κάποιο διαφορετικό τρόπο για να βελτιωθεί η ταχύτητα πρόσβασης.

Η διακαναλική παρεμβολή αποτελεί τον δεύτερο κυριότερο παράγοντα παρεμβολής στην αύξηση της διαθέσιμης ταχύτητας των ασύρματων δικτύων τύπου 802.11 . Όπως είδαμε και παραπάνω (Βλ. Σελ. 21) τα διαθέσιμα κανάλια επικοινωνίας στο φάσμα που λειτουργεί το 802.11 είναι μόλις τρία. Σε περίπτωση που σε κάποια περιοχή βρίσκονται περισσότερα από τρία hotspots οι παρεμβολές μεταξύ των σημάτων είναι αναπόφευκτες αφού η πληροφορία του ενός hotspot αποτελεί θόρυβο για όποιο άλλο hotspot λειτουργεί είτε στο ίδιο κανάλι , είτε σε επικαλυπτόμενο κανάλι.

Η τεχνική δέσμευσης μεγαλύτερου εύρους φάσματος για την βελτίωση της ταχύτητας αποτελεί μια απλή και αποδοτική μέθοδο. Σε αυτήν η κεραία εκπομπής και λήψης χρησιμοποιούν δύο ή και περισσότερα κανάλια του διαθέσιμου φάσματος για την αποστολή και λήψη δεδομένων. Η μεθοδολογία κωδικοποίησης και μετάδοσης παραμένει αναλλοίωτη αφού το μόνο που μεταβάλλεται είναι το δεσμευμένο φάσμα. Συνήθως ο εξοπλισμός αποστολής και λήψης πρέπει να είναι του ίδιου κατασκευαστή ώστε να εκμεταλλεύεται το επιπλέον φάσμα . Το κυριότερο μειονέκτημα της μεθόδου είναι οι διακαναλικές παρεμβολές που πιθανόν να συμβούν σε κάποια περιοχή με πολλά AP's και για αυτό το λόγο αυτή η μέθοδος δεν χρησιμοποιείται ευρέως.

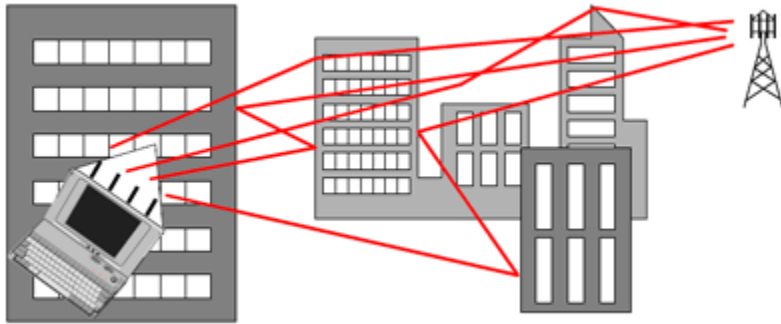
Η τεχνική χρήσης έξυπνων κεραιών τύπου MIMO(Multiple Input Multiple Output) αποτελεί το κυριότερο πεδίο έρευνας για την βελτίωση της ταχύτητας πρόσβασης. Με τη χρήση κεραιών τύπου MIMO το AP χρησιμοποιεί πολλαπλές κεραιές εκπομπής. Υπάρχουν πολλές διαφορετικές τεχνικές εκπομπής με χρήση κεραιών MIMO , μερικές από τις οποίες χρησιμοποιούν τεχνικές βελτιστοποίησης τόσο στο AP όσο και στο κινητό τερματικό (MT) .

Η πιο απλή προσέγγιση αφορά τη χρήση πολλαπλών κεραιών εκπομπής στο AP χωρίς να χρησιμοποιείται κάποια τεχνική βελτιστοποίησης στο MT. Οι πολλαπλές κεραιές εκπομπής καλύπτουν επικαλυπτόμενες περιοχές και έτσι το MT επιλέγει την κεραία με το καλύτερο σήμα. Αυτή η προσέγγιση βελτιώνει τη μέση ταχύτητα πρόσβασης δεν βελτιώνει όμως την μέγιστη ταχύτητα.

Οι πολυκατευθυντικές κεραιές αποτελούν μια παραλλαγή του πρώτου μοντέλου. Σε αυτό το μοντέλο χρησιμοποιούνται πολυκατευθυντικές κεραιές με περιορισμένο λοβό εκπομπής. Ανάλογα με τη θέση του, το MT επιλέγει το λοβό της κεραίας που προσφέρει το μεγαλύτερο λόγο σήματος προς θόρυβο. Η επεξεργασία σήματος που απαιτείται από την πλευρά του κινητού τερματικού είναι σχετικά απλή αφού ο δέκτης αρκεί να ψάχνει για τον καλύτερο λοβό κάθε μερικά δευτερόλεπτα.

Μια πιο έξυπνη τεχνική συνίσταται στη σταθμισμένη και συνδυασμένη λήψη σήματος από πολλαπλές κεραιές(MIMO-SM). Όταν ο χρήστης λάβει δεδομένα από N διαφορετικές κεραιές εκπομπής έχει N διαφορετικούς τρόπους να τα συνδυάσει για να μεγιστοποιήσει την ταχύτητα

μετάδοσης. Αυτό μπορεί να γίνει με το να θεωρήσει το σήμα απ' την πρώτη κεραία σαν πληροφορία και το σήμα των υπολοίπων κεραιών σαν θόρυβο, κατόπιν το σήμα της δεύτερης κεραιάς σαν πληροφορία και το σήμα των υπολοίπων κεραιών σαν θόρυβο κ.ο.κ. Αυτή η τεχνική αναφέρεται και ως MIMO με χωρική πολυπλεξία και παρουσιάζεται σχηματικά στο Σχ. 1.XIV



Σχ. 1.XIV MIMO with spatial multiplexing

Ο ρυθμός μετάδοσης δεδομένων μπορεί να αυξηθεί N φορές διατηρώντας την ίδια ισχύ εκπομπής και εύρος ζώνης. Δηλαδή, χρησιμοποιώντας τεχνικές ψηφιακής επεξεργασίας σήματος μπορούμε να αυξήσουμε το ρυθμό μετάδοσης σε οποιοδήποτε επίπεδο επιθυμούμε προσθέτοντας έξυπνες κεραιές τόσο στο AP όσο και στο κινητό τερματικό. Το κυριότερο μειονέκτημα αυτής της τεχνικής είναι ότι απαιτεί N κεραιές τόσο στον εκπομπό όσο και στον δέκτη με N αλυσίδες μετάδοσης σε RF περιβάλλον καθώς και αρκετά πολύπλοκη υλοποίηση επεξεργασίας σήματος. Έτσι, είναι προφανές ότι η τεχνική MIMO-SM δεν μπορεί να υλοποιηθεί με την υπάρχουσα υποδομή και κυρίως χωρίς να χρειάζεται αλλαγή των ISO προτύπων.

3) Η αύξηση της περιοχής κάλυψης με χρήση λίγων μόνο σημείων πρόσβασης. Τελικός στόχος είναι η ανάπτυξη WISP's (Wireless Internet Service Provider's). Αυτοί οι πάροχοι internet θα αντικαθιστούν τα χάλκινα καλώδια στο "last mile" (ουσιαστικά δηλαδή από το τοπικό κέντρο του ΟΤΕ μέχρι τον καταναλωτή της πολυκατοικίας μας και το σπίτι μας) με ασύρματες ζεύξεις που θα παρέχουν περίπου 15Mbps σε ακτίνα τριών χιλιομέτρων από το σημείο πρόσβασης. Όταν αυτό γίνει πραγματικότητα οι μεγάλες πόλεις θα είναι εξ' ολοκλήρου MAN's (Metropolitan Area Network's) με ευρυζωνικές συνδέσεις οπουδήποτε οποτεδήποτε. Σήμερα, ένα κονσόρτσιουμ εταιρειών (147) από διάφορες χώρες προωθούν ένα τέτοιο πρότυπο με την ονομασία WIMAX(802.16)

Η ομάδα εργασίας του IEEE 802.16 έχει αναπτύξει ασύρματα πρότυπα πρόσβασης ευρείας ζώνης σημείο προς πολλαπλά σημεία για συστήματα που λειτουργούν σε συχνότητες 10-66 GHz και κάτω από τα 11 GHz. Τα πρότυπα καλύπτουν τόσο τον έλεγχο πρόσβασης μέσου (MAC) όσο και το φυσικό στρώμα (PHY).

Διάφορες σκέψεις που αφορούν το φυσικό στρώμα λήφθηκαν υπόψη για την επίτευξη του στόχου. Στις υψηλότερες συχνότητες, η οπτική επαφή είναι απαραίτητη. Ο λόγος είναι ότι θέλουμε να ελαττώσουμε την επίδραση πολλαπλών διαδρομών (multipath fading), επιτρέποντας κανάλια ευρύτερου φάσματος, συνήθως μεγαλύτερα από 10 MHz.

Αυτό δίνει στο πρότυπο IEEE 802.16 τη δυνατότητα να παρέχει ζεύξεις πολύ υψηλής ταχύτητας τόσο στο uplink όσο και στο downlink της σύνδεσης. Για συχνότητες κάτω από 11 GHz

η απαίτηση που έχουμε είναι το πρότυπο να λειτουργεί αποτελεσματικά σε συνθήκες μη οπτικής επαφής(NLOS).

Αρχικά το υπόστρωμα MAC του IEEE 802.16 επεκτάθηκε για να φιλοξενήσει διαφορετικές PHYs και υπηρεσίες, οι οποίες καλύπτουν τις ανάγκες των διαφορετικών περιβαλλόντων εργασίας. Το πρότυπο έχει σχεδιαστεί για να εξυπηρετήσει τόσο ανάπτυξη εφαρμογών βασισμένες σε Time Division Duplexing(TDD) όσο και Frequency Division Duplexing(FDD) , επιτρέποντας τεμαχικά που λειτουργούν τόσο full duplex όσο και half duplex στην περίπτωση του FDD. το υπόστρωμα MAC σχεδιάστηκε συγκεκριμένα για το rmp ασύρματο περιβάλλον πρόσβασης.

Υποστηρίζει υψηλότερα πρωτόκολλα στρώματος ή μεταφοράς όπως το πρωτόκολλο (IP), το πρωτόκολλο Ethernet ή το ATM, και έχει ως σκοπό να προσαρμοσθεί εύκολα σε μελλοντικά πρωτόκολλα που δεν έχουν αναπτυχθεί ακόμα. Η MAC έχει σχεδιαστεί για πολύ υψηλούς ρυθμούς μετάδοσης δεδομένων (μέχρι και 268 mbps ανά κατεύθυνση) σε επίπεδο φυσικού στρώματος , ενώ μπορεί να υλοποιήσει συμβατή με ATM ποιότητα υπηρεσίας(QoS) , UGS, rtPS, nrtPS, και Best Effort(βέλτιστη προσπάθεια για τη μεγαλύτερη δυνατή ταχύτητα χωρίς εγγύηση υπηρεσίας).

Η δομή του πλαισίου επιτρέπει στα τεμαχικά να ορίζουν δυναμικά uplink και downlink προφίλ σύνδεσης ανάλογα με τις συνθήκες στις οποίες πραγματοποιείται η ζεύξη μεταξύ τους. Αυτό επιτρέπει μια ανταλλαγή μεταξύ του εύρους ζώνης της σύνδεσης και της ευρωστίας της σύνδεσης σε πραγματικό χρόνο, και παρέχει κατά προσέγγιση μία αύξηση κατά δύο φορές στο εύρος ζώνης μιας σύνδεσης όταν συγκριθεί με συστήματα χωρίς προσαρμοστικότητα στις συνθήκες περιβάλλοντος , ενώ παράλληλα διατηρεί τη ζεύξη ενεργή.

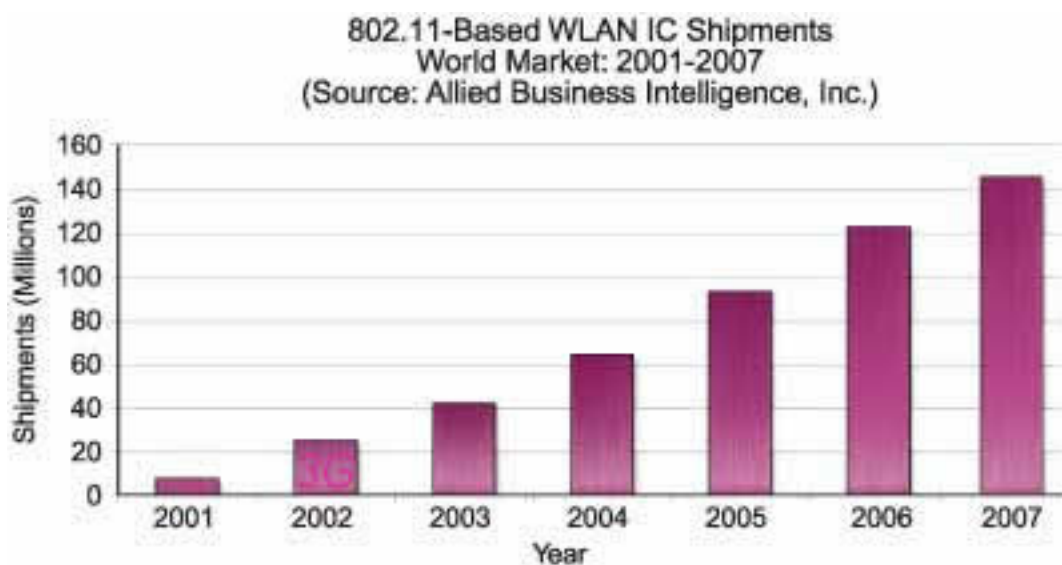
Το υπόστρωμα MAC του 802.16 χρησιμοποιεί μια μεταβλητού μεγέθους Protocol Data Unit (PDU) μαζί με διάφορες άλλες υλοποιήσεις που αυξάνουν πολύ την αποδοτικότητα του πρότυπου. Πολλαπλές MAC PDUs μπορούν να συνδεθούν σε ένα ενιαίο πλαίσιο ώστε να ελαττώσουμε την πλεονάζουσα πληροφορία (overhead) στο φυσικό στρώμα. Επιπλέον, πολλαπλές Service Data Units (SDU) για την ίδια υπηρεσία μπορούν να συνενωθούν σε μία ενιαία MAC PDU, γλιτώνοντας μεγάλο μέρος του overhead σε επίπεδο MAC. Ο τεμαχισμός επιτρέπει σε πολύ μεγάλες SDUs να στείλουν διαμέσω ορίων πλαισίων ώστε να εγγραφεί την QoS ανταγωνιστικών υπηρεσιών. Επίσης, η καταπίεση φορτίου επικεφαλίδας μπορεί να χρησιμοποιηθεί για να μειώσει το overhead που προκαλείται από redundant κομμάτια των επικεφαλίδων SDU.

Το υπόστρωμα MAC χρησιμοποιεί ένα αυτό-επιδιορθούμενο πλάνο request/grant διαχείρισης εύρους ζώνης που πρακτικά εξαλείφει το overhead και την αργοπορία των επιβεβαιώσεων (acknowledgements) , ενώ παράλληλα επιτρέπει καλύτερη διαχείριση ποιότητας υπηρεσίας από τα υπάρχοντα σχήματα. Τα τεμαχικά έχουν ποικιλία επιλογών διαθέσιμες για να απαιτήσουν εύρος ζώνης ανάλογα με το QoS και τις παραμέτρους διακίνησης δεδομένων των υπηρεσιών τους. Τα τεμαχικά μπορούν να ερωτηθούν ξεχωριστά ή σε ομάδες. Μπορούν να ζητήσουν παραπάνω εύρος ζώνης και να κάνουν αιτήσεις για να επερωτηθούν. Τέλος, μπορούν να διαχειριστούν αιτήσεις για μεγαλύτερο εύρος ζώνης.

2. Ασφάλεια Ασύρματων Δικτύων (WLAN)

2.1 Εισαγωγή

Τα τελευταία χρόνια η αγορά προϊόντων ασύρματης δικτύωσης παρουσιάζει εκρηκτική άνοδο. Όπως δείχνει και το γράφημα (Σχήμα XX) το 2004 υπολογίζεται ότι θα διατεθούν στην αγορά πάνω από 60 εκατομμύρια κομμάτια προϊόντων ασύρματης δικτύωσης. Ο Edward A. Rerisi, Διευθυντής στο τμήμα R&D της ABI corp. προβλέπει ότι η βιομηχανία που σχετίζεται με το WLAN θα συνεχίσει να αναπτύσσεται με εκρηκτικούς ρυθμούς όσο προϊόντα σε νέα τμήματα της αγοράς βγαίνουν στην κυκλοφορία. Τα νέα τμήματα της αγοράς είναι σπίτια, μεσαίου μεγέθους γραφεία, ακαδημαϊκές εγκαταστάσεις, υπηρεσίες μεταφορών, εγκαταστάσεις υγείας, βιομηχανικά κέντρα ακόμα και τοπικά εστιατόρια ή καφετέριες. Επιπλέον τα πρωτόκολλα 802.11a και 802.11b και 802.11g δρουν καταλυτικά στη διείσδυση προϊόντων WLAN σε νέα τμήματα της αγοράς με τις αυξημένες δυνατότητες για ταχύτητα που προσφέρουν. Έρευνα της ABI προβλέπει κύκλο εργασιών για την βιομηχανία WLAN της τάξης των \$1.67 δισεκατομμύρια δολάρια για το 2003.



Σχ. 2.1 Η βιομηχανία 802.11

Η αγορά των προϊόντων ασύρματης δικτύωσης λοιπόν είναι μια μεγάλη και δυναμικά αναπτυσσόμενη αγορά. Σε λίγα χρόνια ασύρματα δίκτυα WLAN θα βρίσκονται σε κάθε πόλη και σε κάθε γειτονιά. Το ζήτημα της ασφάλειας των δικτύων WLAN είναι πλέον πολύ σοβαρό αφού ευαίσθητα προσωπικά δεδομένα (αριθμοί πιστωτικών καρτών, κωδικοί πρόσβασης, ευαίσθητα έγγραφα) διακινούνται καθημερινά πάνω σε ανασφαλές μέσο (αέρας). Η ανασφάλεια του μέσου μεταφοράς μάλιστα προβληματίζει πολλές επιχειρήσεις για το αν θα πρέπει να υλοποιήσουν λύσεις ασύρματης δικτύωσης και εν τέλει να αντικαταστήσουν κατά το μεγαλύτερο μέρος ή εντελώς τα ενσύρματα δίκτυα για τα οποία παραδοσιακά υπάρχει μεγαλύτερη εμπιστοσύνη.

Το IEEE κατά την επικύρωση του πρωτοκόλλου 802.11 (09/99) ασύρματης δικτύωσης λαμβάνοντας υπόψη τις ανησυχίες και προσπαθώντας να σχεδιάσει ένα ασφαλές πρωτόκολλο ασύρματης δικτύωσης ενσωμάτωσε το μηχανισμό ασφαλείας WEP (Wired Equivalent Privacy). Σήμερα, μετά από πέντε χρόνια η φράση που κυριαρχεί σχετικά με το επίπεδο ασφάλειας που παρέχει το WEP είναι:

WEP isn't. (το WEP ΔΕΝ προσφέρει αντίστοιχο επίπεδο ασφάλειας όσο ένα ενσύρματο δίκτυο)

Μαζί με το WEP το IEEE ενσωμάτωσε πολλούς άλλους μηχανισμούς ασφαλείας οι οποίοι λίγο έως πολύ θεωρούνται επίσης παρωχημένοι. Ας τους εξετάσουμε όμως πιο αναλυτικά.

2.2 Ανάλυση μηχανισμών ασφαλείας σε WLAN

Κατά την επικύρωση του 802.11 το IEEE υποστήριξε ότι τα παρακάτω είναι δικλίδες ασφαλείας για την εχεμύθεια των δεδομένων.

1. **διάχυση φάσματος (spread spectrum):** Κατά τη διάρκεια του δευτέρου παγκοσμίου πολέμου η τεχνική διάχυσης φάσματος χρησιμοποιήθηκε ευρέως από τους συμμάχους ώστε να μην μπορεί ο εχθρός να μπλοκάρει τις ραδιοεπικοινωνίες. Πρακτικά , συνίσταται από μεταβλητούς κώδικες διάχυσης που μεταβάλλονται κατά τρόπο ώστε μόνο όποιος γνωρίζει τον κώδικα να μπορεί να παρακολουθήσει τις επικοινωνίες. Για λόγους συμβατότητας και ανοιχτού προτύπου όμως το IEEE δημοσιοποίησε λεπτομέρειες του κώδικα διάχυσης ώστε να μπορούν ανεξάρτητοι κατασκευαστές να παράγουν προϊόντα 802.11. Τελικά , κάθε επίδοξος εισβολέας μπορεί να χρησιμοποιήσει μια απλή κάρτα 802.11 και να καταρρίψει αυτήν τη δικλίδα ασφάλειας.
2. **SSID (Service Set Identifier):** Το SSID περιγράφεται από το 802.11 σαν ένα είδος κωδικού πρόσβασης για τον χρήστη ώστε να συνδεθεί με κάποιο WLAN. Για να γίνει εφικτή η επικοινωνία πρέπει το AP και ο χρήστης (NIC) να έχουν το ίδιο SSID. Στην πραγματικότητα το SSID είναι η μόνη **υποχρεωτική** δικλίδα ασφαλείας που θέσπισε το IEEE κατά την επικύρωση του πρωτοκόλλου. Το πρόβλημα είναι ότι το SSID μεταδίδεται ανοιχτά στον αέρα πολλές φορές το δευτερόλεπτο από το AP μέσα σε κάθε πλαίσιο διαφήμισης (beacon frame). Κάθε επίδοξος εισβολέας μπορεί να χρησιμοποιήσει κάποιο από τα πολλά εργαλεία ανάλυσης ασύρματων δικτύων που υπάρχουν(Airmagnet , Aironet , NetStumbler) και να μάθει το SSID που «κυκλοφορεί» στον αέρα. Ευτυχώς πολλοί κατασκευαστές έχουν την επιλογή να απενεργοποιηθεί το πλαίσιο διαφήμισης (να μην αποστέλλεται δηλαδή πλαίσιο για εντοπισμό χρηστών στο χώρο) όμως και πάλι ο επίδοξος εισβολέας μπορεί να περιμένει υπομονετικά μέχρι να αναγκαστεί το AP να στείλει πλαίσιο διαφήμισης λόγω της μεταγωγής κάποιου χρήστη από άλλο AP(roaming) ή λόγω εισόδου κάποιου νέου χρήστη στο δίκτυο.
3. **MAC :** Το 1999 που παρουσιάστηκε το πρωτόκολλο ήταν ισχυρή δικλίδα ασφαλείας η πρόσβαση στο δίκτυο μόνο υπολογιστών με διευθύνσεις MAC που όριζε ο διαχειριστής του δικτύου. Σήμερα , πολλές κάρτες WLAN επιτρέπουν με άμεσους ή πλάγιους τρόπους να αλλάξει αυτή η διεύθυνση οπότε δεν αποτελεί πλέον μέσο ενίσχυσης της ασφαλείας των WLAN's αφού δεν αποτρέπει την μη-εξουσιοδοτημένη πρόσβαση χρηστών. Επιπλέον , αυξάνει κατά πολύ το διαχειριστικό φόρτο ειδικά σε περιπτώσεις μεγάλων δικτύων (πάνω από 50-100 χρήστες) και καθιστά το δίκτυο σχεδόν στατικό αφού κάθε χρήστης που θέλει να εισέλθει νόμιμα στο δίκτυο πρέπει να ειδοποιήσει τον διαχειριστή ώστε να του επιτρέψει την είσοδο μεταβάλλοντας τις ρυθμίσεις του AP.

Εκτός από τις δικλίδες ασφαλείας που είχε παρουσιάσει το 802.11 και αποδείχθηκαν ανεπαρκείς οι διαχειριστές των ασύρματων δικτύων κάποιες φορές εφαρμόζοντας πρακτικές των ενσύρματων δικτύων διευκολύνουν το έργο κακόβουλων εισβολέων. Συνηθισμένη πρακτική είναι να λειτουργεί DHCP (Dynamic Host Configuration Protocol) διακομιστής ώστε να αποδίδει αυτόματα διευθύνσεις IP σε όποιον χρήστη περάσει τις υπόλοιπες δικλίδες ασφαλείας. Με αυτόν τον τρόπο ο εισβολέας αποφεύγει το πρόβλημα του να βρει τις διευθύνσεις που είναι έγκυρες και μπορεί μόλις συσχετιστεί με το AP να αποκτήσει πρόσβαση σε πόρους του δικτύου.

2.3 Πρωτόκολλο κρυπτογράφησης WEP (isn't?)

Το WEP κρυπτογραφεί το σώμα κάθε πλαισίου. Παραπάνω ισχυριστήκαμε αξιωματικά ότι το WEP δεν προσφέρει το επίπεδο ασφάλειας που υπόσχεται. Εδώ θα το αποδείξουμε πιο παραστατικά.

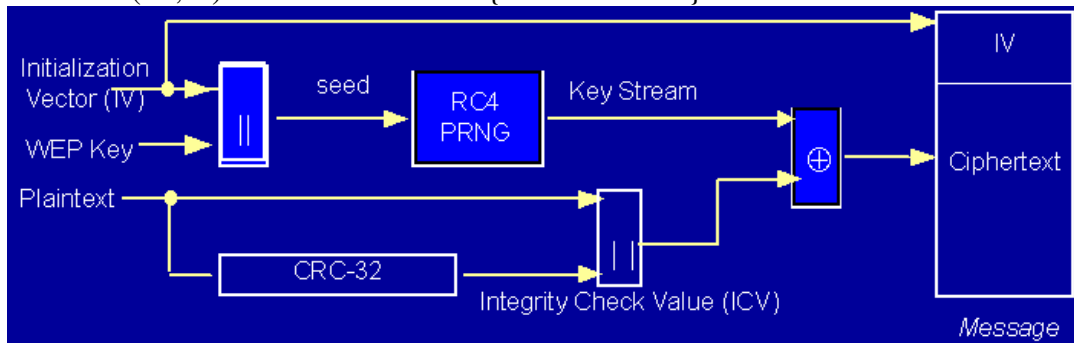
2.3.1 Τρόπος λειτουργίας

Οι υπηρεσίες που φιλοδοξούσε να προσφέρει το WEP είναι τρεις:

- Εμπιστευτικότητα δεδομένων – αποτροπή διαρροής ευαίσθητων προσωπικών δεδομένων
- Ακεραιότητα δεδομένων – σιγουριά ότι τα δεδομένα που μεταδίδονται είναι έγκυρα και όπως εστάλησαν
- Πιστοποίηση χρήστη – περιορισμός πρόσβασης και τελικά απαγόρευση σε μη πιστοποιημένους χρήστες

Κατά την κρυπτογράφηση των δεδομένων για την ενθυλάκωση τους σε πλαίσιο κρυπτογραφημένης πληροφορίας WEP πραγματοποιούνται τα βήματα που φαίνονται παρακάτω:

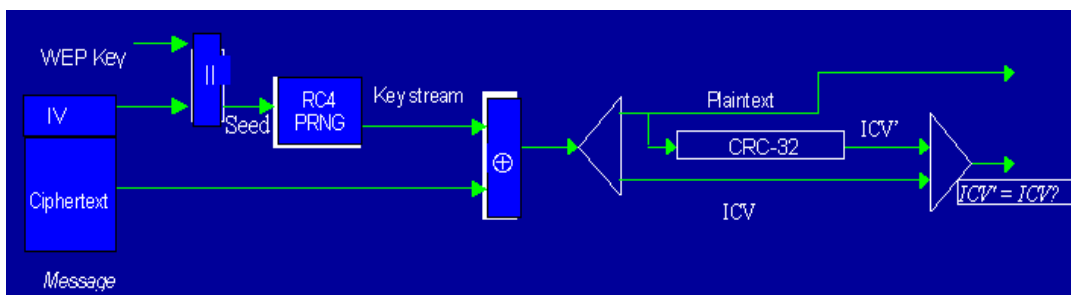
1. $P = \langle M \parallel \text{checksum}(M) \rangle$ {p=plaintext}
2. $\text{KeyStream} = \text{RC4}(\text{IV} \parallel k)$ {k=shared-key}
3. $C = \text{XOR}(P, \text{KeyStream})$ {c=ciphertext}
4. $\text{Transmit}(\text{IV}, C)$ {IV=init-vector}



Σχ. 2.Π Κρυπτογράφηση WEP

Ενώ κατά την αποκρυπτογράφηση (αποθυλάκωση) των δεδομένων από το κρυπτογραφημένο πλαίσιο WEP τα βήματα είναι τα επόμενα:

1. $\text{KeyStream} = \text{RC4}(\text{IV} \parallel k)$
2. $P' = \text{XOR}(C, \text{KeyStream}) = \langle M \parallel \text{checksum}(M) \rangle$
3. If $\text{checksum}(M') = (\text{checksum}(M))'$ Then P' is accepted



Σχ. 2.ΙΙΙ Αποκρυπτογράφηση WEP

Η λειτουργία του WEP διαδραματίζεται στο επίπεδο MAC (medium access control) . Αν ο χρήστης ενεργοποιήσει το WEP και το υποστηρίζει και το σημείο πρόσβασης , η κάρτα ασύρματου δικτύου του χρήστη κρυπτογραφεί το φορτίο αποστολής (δηλαδή το σώμα του πλαισίου και τα bit CRC) κάθε πλαισίου τύπου 802.11 πριν την αποστολή χρησιμοποιώντας κώδικα RC4. Ο αποδέκτης πραγματοποιεί την αποκρυπτογράφηση με την παραλαβή του κάθε πλαισίου.

Σαν μέρος της διαδικασίας κρυπτογράφησης , το WEP προετοιμάζει τον προγραμματισμό κλειδιών (seed) συγχωνεύοντας το κοινό μυστικό κλειδί που δίνεται από τον αποστολέα με ένα τυχαία δημιουργούμενο 24-bit διάνυσμα αρχικοποίησης (IV). Το IV επεκτείνει τη διάρκεια ζωής του μυστικού κλειδιού επειδή ο σταθμός αποστολής μπορεί να αλλάξει το IV για κάθε αποστολή πλαισίου. Το WEP χρησιμοποιεί τον προγραμματισμό κλειδιών σαν είσοδο σε μια γεννήτρια ψευδοτυχαίων αριθμών που παράγει μια ροή κλειδιών (keystream) ισοδύναμη με το μήκος του φορτίου πλαισίου συν ένα 32-bit ICV (Integrity Check Value)

Το ICV είναι ένας αριθμός ελέγχου που ο αποδέκτης επαναυπολογίζει και τον συγκρίνει με αυτόν που του στάλθηκε για να διαπιστώσει αν τα δεδομένα μεταβλήθηκαν κατά την μεταφορά πάνω από το ανασφαλές μέσο. Αν ο αποδέκτης συγκρίνει τα δύο ICV και τα βρει διαφορετικά μπορεί να απορρίψει το πλαίσιο ή να σημειώσει τον χρήστη ως ύποπτο.

Το WEP προσδιορίζει ένα κοινό μυστικό κλειδί μήκους 40 ή 64 bit για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων. Κάποιοι κατασκευαστές υλοποιούν κλειδιά 128-bit τα οποία όμως όπως θα δούμε παρακάτω δεν προσφέρουν μεγαλύτερη ασφάλεια. Κάθε κάρτα ασύρματης πρόσβασης και σημείο πρόσβασης πρέπει φυσικά να ρυθμιστούν χειροκίνητα ώστε να χρησιμοποιούν το ίδιο κλειδί.

Πριν ξεκινήσει η μετάδοση το WEP συνδυάζει τη ροή κλειδιών με το φορτίο/ICV με μια διαδικασία XOR όπως φαίνεται στο σχήμα XX , που παράγει κρυπτογραφημένα δεδομένα(ciphertext). Το WEP συμπεριλαμβάνει το IV μη κρυπτογραφημένο μέσα στα πρώτα bytes του σώματος πλαισίου. Ο αποδέκτης χρησιμοποιεί αυτό το IV μαζί με το κοινό μυστικό κλειδί που παρέχεται από τον χρήστη του αποδέκτη σταθμού για να αποκρυπτογραφήσει το φορτίο του σώματος πλαισίου.

Στις περισσότερες περιπτώσεις ο αποστολέας χρησιμοποιεί διαφορετικό IV για κάθε πλαίσιο (αν και δεν απαιτείται από το πρωτόκολλο 802.11). Όταν τα μεταδιδόμενα μηνύματα έχουν κοινή αρχή , η αρχή κάθε κρυπτογραφημένου πλαισίου είναι ίδια όταν χρησιμοποιείται το ίδιο κλειδί. Μετά την κρυπτογράφηση των δεδομένων , η αρχή αυτών των πλαισίων θα είναι ίδιες , προσφέροντας πρόσφορο έδαφος για επίδοξους εισβολείς. Αφού το IV είναι διαφορετικό για τα πιο πολλά πλαίσια , το WEP προσφέρει ικανοποιητική ασφάλεια σε τέτοιου τύπου επιθέσεις. Η συχνή αλλαγή του IV επίσης βελτιώνει την ικανότητα του WEP να προστατεύεται έναντι επιθέσεων.

2.3.2 Προβλήματα ασφάλειας WEP

Τρεις ερευνητές από το πανεπιστήμιο της Καλιφόρνιας στο Μπέρκλεϊ (Nikita Borisov, Ian Goldberg , David Wagner) ανακάλυψαν μια σημαντική τρύπα ασφάλειας στην κρυπτογράφηση του WEP το 2001. Επιπλέον τον Αύγουστο του 2001 οι κρυπτογράφοι Scott Fluhrer, Itsik Mantin και Adi Shamir δημοσίευσαν ένα paper για τις αδυναμίες της κρυπτογράφησης RC4 πάνω στις οποίες βασίζεται το WEP. Λίγο αργότερα στα τέλη Αυγούστου του ίδιου έτους ένας φοιτητής του πανεπιστημίου του Rice και δύο υπάλληλοι των AT&T Labs - Research (Adam Stubblefield, John Ioannidis και Aviel D. Rubin) υλοποίησαν επιτυχώς τις ιδέες που είχαν εκφραστεί στις δύο προηγούμενες δημοσιεύσεις.

Προηγουμένως περιγράψαμε τη δημιουργία του διανύσματος αρχικοποίησης (IV). Η «κερκόπορτα» ασφάλειας του WEP βρίσκεται στην φτωχή υλοποίηση του IV. Αν για παράδειγμα ένας επίδοξος εισβολέας χρησιμοποιήσει συνάρτηση XOR για να συσχετίσει δύο πακέτα που επεξεργάστηκαν με τα ίδια IV, δηλαδή, ταυτόσημα κλειδιά RC4, τότε μπορεί εύκολα να υπολογίσει το μυστικό κλειδί.

Αφού το IV είναι 24 bits, σε ένα AP που λειτουργεί στη μέγιστη ταχύτητα λειτουργίας – 1500 byte ανά πακέτα με ρυθμαπόδοση 11Mbps – θα επαναχρησιμοποιηθεί το ίδιο μετά από λιγότερο από πέντε ώρες. Κατά την διάρκεια των πέντε ωρών μεταδίδονται 24GB. Είναι εφικτό λοιπόν να καταγραφούν οι μεταφορές δεδομένων για αρκετές ώρες με τη χρήση φορητού υπολογιστή και τελικά να έχουμε πακέτα με ίδια IV και συνεπώς ίδια RC4 κλειδιά.

Το κενό που αφήνει το πρωτόκολλο σχετικά με τη δημιουργία του IV, οδηγεί πολλούς κατασκευαστές στο να μην χρησιμοποιούν και τα 24-bit για το IV. Με αυτόν τον τρόπο επαναλαμβανόμενα IV μπορεί να εμφανιστούν ακόμα γρηγορότερα.

Οι Fluhrer, Martin and Shamir επίσης βρήκαν ότι υπάρχουν διανύσματα αρχικοποίησης που δίνουν ενδείξεις για ένα byte του κλειδιού με βεβαιότητα 5%. Μετά την καταγραφή τεσσάρων με έξι εκατομμυρίων πακέτων (περίπου 8.5GB) υπάρχουν αρκετά αδύναμα διανύσματα αρχικοποίησης IV ώστε να βρεθεί το WEP κλειδί.

Επιπλέον, αν το software του AP δέχεται ακολουθίες string αντί για Hex για το κλειδί του WEP ο αριθμός των πιθανών συνδυασμών ελαττώνεται. Σε αποτέλεσμα η βεβαιότητα 5% που αναφέρθηκε προηγουμένως μεγαλώνει ούτως ώστε να χρειάζονται μόλις δύο εκατομμύρια πακέτα για να βρεθεί το κλειδί.

Αυτές οι αδυναμίες οδήγησαν στην ανάπτυξη του WPA (Wi-Fi Protected Access) το 2003, ενώ ένα χρόνο αργότερα το IEEE επικύρωσε το πλήρες πρωτόκολλο 802.11i (WPA2).

2.4 WPA

Το πρωτόκολλο WPA στην ουσία είναι μια προέκδοση της τρέχουσας έκδοσης του 802.11i πρωτοκόλλου, η οποία περιλαμβάνει το πρωτόκολλο Temporal Key Integrity Protocol (TKIP) και μηχανισμούς που ανήκουν στο 802.1x. Ο συνδυασμός αυτών των μηχανισμών παρέχει κρυπτογράφηση με χρήση δυναμικού κλειδιού και αμοιβαία πιστοποίηση, κάτι που αναμφίβολα χρειάζεται σε WLANs.

Όπως και με το WEP, το TKIP χρησιμοποιεί το RC4 που παρέχεται από την υπηρεσία ασφάλειας RSA για να κρυπτογραφήσει το σώμα των πλαισίων και το CRC κάθε 802.11 πλαισίου πριν από τη μετάδοση. Τα ζητήματα ασφαλείας με το WEP δεν σχετίζονταν με τον RC4 αλγόριθμο κρυπτογράφησης. Αντιθέτως, τα προβλήματα αφορούσαν πρώτιστα τη δημιουργία κλειδιών και το πως υλοποιείται η κρυπτογράφηση.

Το TKIP προσθέτει τις ακόλουθες τροποποιήσεις ασφαλείας στο WEP:

*** διάνυσμα αρχικοποίησης 48-bit.**

Το WEP δημιουργεί αυτό που αναφέρεται σαν "keyschedule" με το να συνενώνει στο κλειδί κοινού μυστικού (shared secret) με ένα τυχαία-παραγόμενο διάνυσμα αρχικοποίησης(IV) 24-bit. Το WEP εισάγει το keyschedule που παράγεται σε μια γεννήτρια ψευδοτυχαίων αριθμών και παράγει μια ροή κλειδιών(keystream) ίση σε μήκος με το φορτίο του πλαισίου του 802.11 πακέτου. Χρησιμοποιώντας όμως το 24-bit IV διάνυσμα, το WEP τελικά χρησιμοποιεί το ίδιο διάνυσμα για διαφορετικά πακέτα δεδομένων. Σε ένα τυπικό σενάριο χρήσης το ίδιο διάνυσμα IV μπορεί να επανεμφανιστεί μέσα σε μία ώρα ή και λιγότερο. Αυτό οδηγεί στη μετάδοση πλαισίων με αρκετά όμοια κρυπτογράφηση ώστε κάποιος κακόβουλος χρήστης να μπορεί να συλλέξει

πλαίσια που περιέχουν το ίδιο διάνυσμα IV και υπολογίζοντας τις κοινές τιμές τους να φτάσει στην αποκρυπτογράφηση των πλαισίων του 802.11. Το WPA με τη χρήση του TKIP, χρησιμοποιεί το διάνυσμα 48-bit IV τα οποία μειώνουν σημαντικά την επαναχρησιμοποίησή τους και την πιθανότητα ότι ο κακόβουλος χρήστης θα καταφέρει να συλλέξει αρκετά πλαίσια 802.11 δεδομένων ώστε να "σπάσει" την κρυπτογράφηση.

*** Ανα πακέτο κατασκευή και διανομή κλειδιών.**

Το WPA παράγει αυτόματα ένα νέο μοναδικό κλειδί κρυπτογράφησης περιοδικά για κάθε χρήστη. Το WPA χρησιμοποιεί ένα μοναδικό κλειδί για κάθε 802.11 πλαίσιο. Αυτό αποτρέπει τη χρήση του ίδιου κλειδιού για μεγάλο χρονικό διάστημα όπως συμβαίνει με το WEP.

*** Κώδικας ακεραιότητας μηνυμάτων.**

Το WPA εφαρμόζει τον κώδικα ακεραιότητας μηνυμάτων (mic), για να προστατευτεί ενάντια στις επιθέσεις παραποίησης. Το WEP επισυνάπτει μια τιμή ελέγχου ακεραιότητας(ICV) 4-bytes στο ωφέλιμο φορτίο του πακέτου 802.11 . Ο δέκτης πρέπει να υπολογίσει το ICV μόλις λάβει το πλαίσιο για να αποφασίσει αν ταιριάζει με αυτό στο πλαίσιο. Εάν ταιριάζουν, τότε υπάρχει κάποια διασφάλιση ότι το πακέτο δεν αλλοιώθηκε κατά την μεταφορά(από κάποιον τρίτο). Αν και το WEP κρυπτογραφεί το ICV, ένας κακόβουλος χρήστης μπορεί να αλλάξει τα bits στο κρυπτογραφημένο ωφέλιμο φορτίο του πακέτου και να αλλοιώσει το κρυπτογραφημένο ICV χωρίς να μπορεί να ανιχνευτεί από το δέκτη. Το WPA λύνει αυτό το πρόβλημα με τον υπολογισμό του 8-bytes MIC που βρίσκεται αμέσως πριν από το ICV στο πακέτο.

Για την πιστοποίηση, το WPA χρησιμοποιεί έναν συνδυασμό της τεχνικής ανοικτού συστήματος(open system) και της πιστοποίησης 802.1x. Αρχικά, ο ασύρματος χρήστης πιστοποιείται με το Access Point, το οποίο επιτρέπει στον πελάτη να στείλει πλαίσια στο Access Point. Έπειτα, το WPA εκτελεί πιστοποίηση σε επίπεδο χρήστη μέσω του 802.1x. Το WPA συνήθως συνεργάζεται με κάποιον διακομιστή πιστοποίησης (authentication server), όπως είναι ο RADIUS ή το LDAP. Το WPA μπορεί επίσης να λειτουργήσει σε κατάσταση "pre-shared key" αν δεν υπάρχει κάποιος διακομιστής πιστοποίησης.

Παρότι οι βελτιώσεις του WPA πρωτοκόλλου προσφέρουν βελτιωμένη ασφάλεια έναντι του WEP, έχουν βρεθεί αρκετές αδυναμίες του WPA. Κυριότερες εξ'αυτών είναι επιθέσεις με χρήση λεξικού (dictionary attacks) και επιθέσεις ενάντια στο MIC [1] [2]. Για αυτούς τους λόγους έχει υλοποιηθεί ήδη το WPA2 , το οποίο βασίζεται στην τελική έκδοση του 802.11i .

3. AAA Framework

3.1 Authentication (Πιστοποίηση χρήστη)

3.1.1 Εισαγωγή

Το πρώτο βήμα για την παροχή υπηρεσιών ασύρματου δικτύου είναι αναμφίβολα η πιστοποίηση του κινητού τερματικού. Με τη χρήση διαφόρων μεθόδων πιστοποιούμε την ταυτότητα του χρήστη και επικυρώνουμε την φυσική υπόσταση προτού τον εξουσιοδοτήσουμε να χρησιμοποιήσει υπηρεσίες του δικτύου.

Οι μέθοδοι πιστοποίησης χωρίζονται σε δύο μεγάλες κατηγορίες. Αυτές που ανήκουν στο πλαίσιο του AAA(AAA framework) και αυτές που ενεργούν αυτόνομα. Στο παρόν κείμενο θα αναφέρουμε όλες τις δυνατές μεθόδους προτού καταλήξουμε σε αυτήν που θα χρησιμοποιήσουμε.

3.1.2 Τι είναι το AAA framework

Το AAA framework είναι ένα αρχιτεκτονικό πλαίσιο για την ρύθμιση και ενοποίηση ενός συνόλου τριών ανεξάρτητων λειτουργιών ασφαλείας με ένα συνεπή και ενιαίο τρόπο. Το AAA framework αποτελεί μέρος του Cisco IOS. Τα μέρη του AAA framework είναι η πιστοποίηση, η εξουσιοδότηση και η χρέωση.

3.2 Μέθοδοι πιστοποίησης με τη χρήση του AAA framework

Το AAA framework παρέχει επτά μεθόδους πιστοποίησης της ταυτότητας του χρήστη οι οποίες εφαρμόζονται μέσω της λίστας μεθόδων του διακομιστή πρόσβασης. Αυτές είναι οι:

- Πιστοποίηση εισόδου με ενεργοποίηση κωδικού πρόσβασης
- Πιστοποίηση εισόδου με χρήση του Kerberos
- Πιστοποίηση εισόδου με χρήση κωδικού πρόσβασης τύπου γραμμής
- Πιστοποίηση εισόδου με χρήση τοπικού κωδικού πρόσβασης
- Πιστοποίηση εισόδου με χρήση ομάδας RADIUS
- Πιστοποίηση εισόδου με χρήση ομάδας TACACS+
- Πιστοποίηση εισόδου με χρήση ονόματος ομάδας

Αναλυτικότερα για την καθεμία από αυτές έχουμε:

Πιστοποίηση εισόδου με ενεργοποίηση κωδικού πρόσβασης

Με βάση αυτήν την μέθοδο ορίζουμε έναν default κωδικό πρόσβασης για τη χρήση όλων των κινητών τερματικών. Ο αλγόριθμος κωδικοποίησης του κωδικού πρόσβασης είναι αρκετά παλιός και για αυτό το λόγο συνίσταται αυτή η μέθοδος πιστοποίησης να χρησιμοποιείται σε συνδυασμό με την ενεργοποίηση μυστικού κλειδιού.

Πιστοποίηση εισόδου με χρήση του Kerberos

Αυτή η μέθοδος πιστοποίησης είναι διαφορετική από τις πιο πολλές άλλες μεθόδους. Ο λόγος είναι ότι ο κωδικός πρόσβασης δεν στέλνεται ποτέ στον διακομιστή απομακρυσμένης πρόσβασης. Αντ' αυτού η κεντρική «ιδέα» του πρωτοκόλλου είναι η χρησιμοποίηση ενός «εμπιστευόμενου τρίτου προσώπου» που στην περίπτωση μας είναι το access point. Το «εμπιστευόμενο τρίτο πρόσωπο» ονομάζεται κέντρο διανομής κλειδιών (KDC).

Με αυτήν τη μέθοδο ο χρήστης του κινητού τερματικού μεταδίδει στον διακομιστή μόνο το όνομα χρήστη. Ο διακομιστής (Kerberos) ελέγχει αν υπάρχει αυτό το όνομα χρήστη στη λίστα νόμιμων χρηστών του δικτύου και αν ναι δημιουργεί ένα κρυπτογραφημένο κουπόνι επιβεβαίωσης (TGT) – με περιορισμένη διάρκεια ζωής - και το αποστέλλει στον router του δικτύου. Κατόπιν ο χρήστης αποστέλλει τον κωδικό πρόσβασης στον router και αυτός επιχειρεί να τον αποκωδικοποιήσει με τη χρήση του κουπονιού επιβεβαίωσης (TGT) . Εφόσον η διαδικασία πετύχει ο χρήστης πιστοποιείται και το TGT αποθηκεύεται στον προσωρινό χώρο αποθήκευσης του χρήστη μέσα στο router. Η διαδικασία αυτή πραγματοποιείται μόνο την 1^η φορά κατά τη διάρκεια ενός session αφού έπειτα η πιστοποίηση μπορεί να γίνει με το κρυπτογραφημένο αντίτυπο του κωδικού πρόσβασης που βρίσκεται αποθηκευμένο στον προσωρινό χώρο αποθήκευσης.

Το πρωτόκολλο αναπτύχθηκε στο Ινστιτούτο Τεχνολογίας της Μασσαχουσέτης (MIT) και χρησιμοποιεί τον αλγόριθμο κρυπτογράφησης DES.

Πιστοποίηση εισόδου με χρήση κωδικού πρόσβασης τύπου γραμμής

Αυτή η μέθοδος πιστοποίησης είναι παρόμοια με την πιστοποίηση εισόδου με ενεργοποίηση κωδικού πρόσβασης που προαναφέρθηκε. Και εδώ ορίζεται ένας κωδικός πρόσβασης για τη χρήση των κινητών χρηστών.

Πιστοποίηση εισόδου με χρήση τοπικού κωδικού πρόσβασης

Μέσω αυτής της μεθόδου ο διακομιστής πρόσβασης χρησιμοποιεί την τοπική του βάση δεδομένων για την πιστοποίηση των χρηστών. Αυτή η βάση δεδομένων περιέχει ζεύγη όνομα χρήστη – κωδικού πρόσβασης τα οποία συγκρίνει με την είσοδο του χρήστη.

Πιστοποίηση εισόδου με χρήση ομάδας RADIUS

Η πιστοποίηση με χρήση ομάδας RADIUS διακομιστών είναι ίσως η πιο διαδεδομένη μέθοδος πιστοποίησης απομακρυσμένων χρηστών. Το σύστημα RADIUS είναι καταναμημένο σε πελάτη / εξυπηρετητή και προστατεύει το δίκτυο από μη εξουσιοδοτημένη πρόσβαση. Το πρωτόκολλο RADIUS είναι τύπου open source (ανοιχτού λογισμικού) και ο πηγαίος κώδικας του είναι διαθέσιμος στο ευρύ κοινό ώστε να μπορεί να προσαρμοστεί σε κάθε σύστημα ασφαλείας που είναι διαθέσιμο στην αγορά.

Η πιστοποίηση του χρήστη με χρήση ομάδας RADIUS γίνεται με τα ακόλουθα βήματα:
1) Ο απομακρυσμένος χρήστης εισάγει το όνομα χρήστη και τον κωδικό πρόσβασης του.

2) Το απομακρυσμένο σύστημα κρυπτογραφεί τον κωδικό πρόσβασης και μαζί με το όνομα χρήστη τα αποστέλλει μέσω του δικτύου στον διακομιστή RADIUS.

3) Ο διακομιστής RADIUS αποκρίνεται με ένα από τα ακόλουθα σήματα:

- ACCEPT - ο χρήστης πιστοποιήθηκε
- REJECT - ο χρήστης ΔΕΝ πιστοποιήθηκε και του ζητείται να υποβάλλει ξανά το όνομα χρήστη και τον κωδικό πρόσβασης του
- CHALLENGE – ο διακομιστής RADIUS ζητά επιπλέον στοιχεία από τον χρήστη
- CHANGE PASSWORD - ο διακομιστής RADIUS ζητά από τον χρήστη να αλλάξει τον κωδικό πρόσβασης του

Όταν ο διακομιστής RADIUS αποκριθεί με REJECT ή ACCEPT αποστέλλει επιπλέον στοιχεία που χρησιμοποιούνται για EXEC ή για την εξουσιοδότηση του απομακρυσμένου χρήστη. Είναι απαραίτητο να πιστοποιηθεί πρώτα ο απομακρυσμένος χρήστης πρωτού εξουσιοδοτηθεί πρόσβαση σε μέρος ή όλες τις υπηρεσίες του δικτύου. Τα επιπλέον στοιχεία είναι τα παρακάτω:

- Υπηρεσίες τις οποίες ο χρήστης έχει πρόσβαση όπως telnet, PPP , slip ή EXEC.
- Παράμετροι σύνδεσης όπως είναι η IP διεύθυνση του πελάτη ή / και του εξυπηρετητή , η access list(λίστα πρόσβασης) και το timeout που έχουμε ορίσει για αυτόν το χρήστη.

Υπάρχουν ωστόσο περιπτώσεις στις οποίες δεν μπορούμε να χρησιμοποιήσουμε RADIUS διακομιστή για να διασφαλίσουμε την ασφάλεια του δικτύου , όπως:

- Όταν θέλουμε να χρησιμοποιήσουμε διαφορετικά πρωτόκολλα όπως ARAP (Appletalk Remote Access Protocol) , NBFCP (NetBios Frame Protocol Control Protocol) , NASI (Netware Asynchronous Service Interface και συνδέσεις τύπου X.25 PAD τα οποία δεν υποστηρίζει το πρωτόκολλο RADIUS
- Επικοινωνία δρομολογητή με άλλον δρομολογητή αφού το RADIUS δεν υποστηρίζει αμφίδρομη πιστοποίηση
- Δίκτυα που χρησιμοποιούν ποικιλία διαθέσιμων υπηρεσιών αφού το RADIUS γενικά περιορίζει τους πιστοποιημένους χρήστες στις υπηρεσίες που τους δίνει πρόσβαση κατα την πιστοποίηση (???)

Πιστοποίηση εισόδου με χρήση ομάδας TACACS+

Με αυτή τη μέθοδο χρησιμοποιείται ο TACACS+ ως μέθοδος πιστοποίησης της ταυτότητας του χρήστη. Ο TACACS+ είναι μία εφαρμογή που προσφέρει κεντρική διαχείριση και πιστοποίηση του ελέγχου πρόσβασης σε διακομιστή πρόσβασης.

Κατά την πιστοποίηση σημαντική παράμετρος είναι μπορούν να ζητηθούν εκτός από το όνομα χρήστη και τον κωδικό πρόσβασης και άλλα στοιχεία όπως email , ταχυδρομικό κώδικα και οτιδήποτε παραστεί αναγκαίο για επιπλέον ασφάλεια. Ο τρόπος που λειτουργεί ο TACACS+ είναι απλός.

Αρχικά, ζητείται από τον χρήστη να εισάγει όνομα χρήστη. Εφόσον το όνομα χρήστη υπάρχει στη βάση δεδομένων του TACACS+ ζητείται ο κωδικός πρόσβασης. Αν και ο κωδικός πρόσβασης υπάρχει στη βάση και εφόσον ο TACACS+ δεν ζητεί επιπλέον πληροφορίες πιστοποίησης επιτρέπεται η είσοδος του χρήστη στο σύστημα. Ο αλγόριθμος κωδικοποίησης με τον οποίο μεταδίδονται τα δεδομένα είναι ο MD5.

Πιστοποίηση εισόδου με χρήση ονόματος ομάδας

Αυτή η μέθοδος ουσιαστικά συνίσταται στο να χρησιμοποιήσουμε ομάδα διακομιστών RADIUS ή TACACS+ για την πιστοποίηση των χρηστών. Με αυτόν τον τρόπο μπορούμε να διαμοιράσουμε το φόρτο εργασίας και να επιταχύνουμε τη διαδικασία πιστοποίησης του χρήστη.

3.3 Μέθοδοι πιστοποίησης χωρίς τη χρήση του AAA framework

Το AAA framework παρέχει επτά μεθόδους πιστοποίησης της ταυτότητας του χρήστη οι οποίες εφαρμόζονται μέσω της λίστας μεθόδων του διακομιστή πρόσβασης. Αυτές είναι οι:

- Πιστοποίηση εισόδου με χρήση κωδικού πρόσβασης τύπου γραμμής
- Πιστοποίηση εισόδου με πιστοποίηση ονόματος χρήστη
- Πιστοποίηση εισόδου με χρήση CHAP ή PAP μεθόδου
- Πιστοποίηση εισόδου με χρήση MSCHAP μεθόδου
- Πιστοποίηση εισόδου με χρήση 802.1X

Πιστοποίηση εισόδου με χρήση κωδικού πρόσβασης τύπου γραμμής

Με αυτήν τη μέθοδο ορίζουμε έναν κωδικό πρόσβασης τον οποίο ο χρήστης πρέπει να εισάγει σε γραμμή τερματικού.

Πιστοποίηση εισόδου με πιστοποίηση ονόματος χρήστη

Με βάση αυτό το σχήμα πιστοποίησης χρήστη η μόνη πληροφορία πιστοποίησης είναι το όνομα χρήστη. Το σκεπτικό της μεθόδου είναι η υλοποίηση συστήματος πιστοποίησης χρήστη σε συστήματα που δεν υποστηρίζουν το TACACS με κρυπτογραφημένο κωδικό πρόσβασης ή να επιτρέψουν την είσοδο σε χρήστες με «ειδικά» προνόμια όπως οι guest λογαριασμοί.

Πιστοποίηση εισόδου με χρήση CHAP ή PAP μεθόδου

Ένα από τα πιο γνωστά πρωτόκολλα που χρησιμοποιούνται σήμερα από τους παροχείς Internet (ISP's) είναι το PPP(Point-to-Point Protocol). Σε αυτό οι απομακρυσμένοι χρήστες συνδέονται με τον διακομιστή πρόσβασης για να αρχίσουν μία PPP σύννοδο. Μόλις διαπραγματευτεί το PPP οι απομακρυσμένοι χρήστες συνδέονται με το δίκτυο κορμού του ISP και κατ' επέκταση και το Internet.

Επειδή οι ISP's θέλουν μόνο οι δικοί τους χρήστες να συνδέονται στον διακομιστή πρόσβασης, οι απομακρυσμένοι χρήστες πιστοποιούνται προτού αρχίσουν την σύννοδο PPP. Αυτό γίνεται συνήθως εισάγοντας όνομα χρήστη και κωδικό πρόσβασης όταν τους ζητηθεί από τον διακομιστή. Παρότι αυτή είναι μια αποδεκτή λύση είναι δύσκολη στη διαχείριση για αυτό και μπορούμε να χρησιμοποιήσουμε μια εναλλακτική μέθοδο.

Αυτή είναι να χρησιμοποιήσουμε τα πρωτόκολλα πιστοποίησης που είναι ενσωματωμένα στο PPP. Σε αυτήν την περίπτωση ο απομακρυσμένος χρήστης συνδέεται με τον διακομιστή πρόσβασης και ξεκινά μια ελάχιστη(minimal) σύννοδο PPP με τον διακομιστή. Αυτή η ελάχιστη σύννοδος δεν δίνει στον χρήστη δικαιώματα πρόσβασης στο δίκτυο, στην πραγματικότητα μόλις που είναι αρκετή για την μεταξύ τους επικοινωνία.

Το PPP μέχρι σήμερα υποστηρίζει δύο πρωτόκολλα πιστοποίησης. Το PAP (Password Authentication Protocol) και το CHAP (Challenge Handshake Authentication Protocol) τα οποία προσδιορίζονται στο RFC 1334 και υποστηρίζονται τόσο από σύγχρονες όσο και από ασύγχρονες διεπαφές. Η πιστοποίηση μέσω PAP ή CHAP είναι ισοδύναμη με του ζεύγους όνομα χρήστη / κωδικού πρόσβασης. Το CHAP θεωρείται γενικά πιο ασφαλές αφού ο κωδικός πρόσβασης δεν μεταδίδεται σε καμία περίπτωση μέσω της ζεύξης.

Για να χρησιμοποιήσουμε PAP ή CHAP πρέπει να έχουμε ενεργοποιήσει την ενθυλάκωση PPP. Η διαδικασία πιστοποίησης του χρήστη με χρήση PAP είναι αρκετά απλή και συμπυκνώνεται στην αποστολή από τον απομακρυσμένο χρήστη της αίτησης πιστοποίησης. Αν το όνομα χρήστη και ο κωδικός πρόσβασης που βρίσκονται στην αίτηση πιστοποίησης γίνουν

αποδεκτά από τον διακομιστή πρόσβασης στέλνεται θετική απάντηση πιστοποίησης και επιτρέπεται η είσοδος του χρήστη.

Η διαδικασία πιστοποίησης με χρήση CHAP είναι κάπως πιο σύνθετη, λογικό αφού πρόκειται για μέθοδο που προσφέρει μεγαλύτερη ασφάλεια. Όταν το CHAP είναι ενεργοποιημένο σε κάποια διεπαφή και κάποια απομακρυσμένη συσκευή προσπαθήσει να συνδεθεί καταρχάς ο διακομιστής πρόσβασης στέλνει ένα πακέτο τύπου CHAP στην συσκευή. Το πακέτο «προκαλεί» τη συσκευή να απαντήσει. Αυτό το πακέτο περιέχει πληροφορίες όπως το ID, έναν τυχαίο αριθμό και το όνομα host του τοπικού δρομολογητή.

Μόλις η απομακρυσμένη συσκευή δεχτεί το πακέτο «πρόκλησης» συμπυκνώνει το ID, τον τυχαίο αριθμό και το όνομα host του τοπικού δρομολογητή και τα κρυπτογραφεί όλα μαζί χρησιμοποιώντας τον κωδικό πρόσβασης που θα χρησιμοποιούσε για να πιστοποιηθεί στον διακομιστή πρόσβασης. Κατόπιν στέλνει τα αποτελέσματα στον διακομιστή πρόσβασης μαζί με το όνομα χρήστη που αποτελεί μέρος του ζεύγους όνομα χρήστη/κωδικός πρόσβασης του οποίου τον κωδικό πρόσβασης χρησιμοποίησε για την κρυπτογράφηση του πακέτου «πρόκλησης».

Όταν ο διακομιστής πρόσβασης δεχτεί το πακέτο από την απομακρυσμένη συσκευή χρησιμοποιεί το (μη κρυπτογραφημένο) όνομα χρήστη για να βρει από τα στοιχεία της βάσης του τον κωδικό πρόσβασης που του αντιστοιχεί. Ο κωδικός πρόσβασης που βρίσκει σε αυτήν την φάση πιστοποίησης πρέπει να είναι ο ίδιος με αυτόν που χρησιμοποίησε η απομακρυσμένη συσκευή για να κρυπτογραφήσει το πακέτο «πρόκλησης». Κατόπιν χρησιμοποιεί αυτόν τον κωδικό πρόσβασης για να κρυπτογραφήσει εκ νέου το πακέτο «πρόκλησης» και συγκρίνει το νέο πακέτο με αυτό που του έστειλε η απομακρυσμένη συσκευή. Εφόσον είναι ταυτόσημα η πιστοποίηση πετυχαίνει.

Η δοσοληψία τύπου CHAP συμβαίνει μόνο για να εγκαθιδρυθεί η σύνδεση. Κατά την διάρκεια της συνόδου που ακολουθεί ο διακομιστής πρόσβασης δεν ζητεί τον κωδικό πρόσβασης. Ωστόσο, η απομακρυσμένη συσκευή μπορεί να απαντήσει σε αιτήσεις άλλων συσκευών κατά τη διάρκεια της κλήσης.

Πιστοποίηση εισόδου με χρήση MSCHAP μεθόδου

Το MSCHAP (MicroSoft Challenge Handshake Authentication Protocol) είναι η έκδοση του CHAP από την Microsoft και αποτελεί επέκταση της πρότασης του RFC 1394. Όπως και το αυθεντικό CHAP πρωτόκολλο έτσι και αυτό χρησιμοποιείται για PPP πιστοποίηση.

Οι διαφορές του MSCHAP με το CHAP είναι οι ακόλουθες:

- Το MS-CHAP ενεργοποιείται με την διαπραγμάτευση CHAP αλγορίθμου 0x80 σε επιλογή LCP 3 πρωτόκολλου πιστοποίησης.
- Το πακέτο απάντησης του MS-CHAP βρίσκεται σε μορφή συμβατή με Windows NT 3.5 και 3.51, Windows 95 και Microsoft LAN Manager 2.x. Σε αυτήν τη μορφή δεν απαιτείται η αποθήκευση του κωδικού πρόσβασης.
- Το MS-CHAP παρέχει έναν μηχανισμό διαδοχικών προσπαθειών για αποτυχημένες προσβάσεις στο σύστημα.
- Το MS-CHAP παρέχει ένα μηχανισμό για την αλλαγή του κωδικού πρόσβασης στο σύστημα
- Το MS-CHAP προσδιορίζει ένα σύνολο κωδικών με την ονομασία “reason-for-failure” σε περίπτωση λάθους που ενσωματώνονται στο πεδίο μηνύματος του πακέτου αποτυχίας.

Ανάλογα με τα πρωτόκολλα ασφαλείας που θέλουμε να υλοποιήσουμε η πιστοποίηση χρήστη μέσω PPP και MS-CHAP μπορεί να χρησιμοποιήσει ή όχι το πλαίσιο AAA. Αν έχουμε

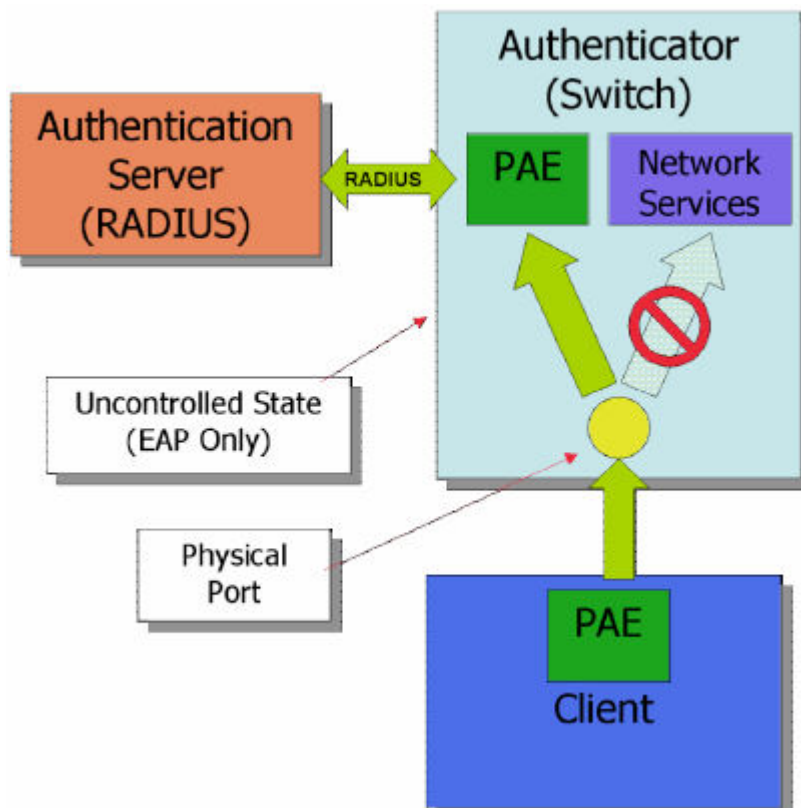
ενεργοποιήσει το AAA η πιστοποίηση PPP μπορεί να γίνει με συνδυασμό TACACS+ και RADIUS διακομιστών.

Πιστοποίηση εισόδου με χρήση 802.1X

Το στάνταρ του IEEE 802.1X είναι σχεδιασμένο για να επεκτείνει την ασφάλεια ασύρματων (και όχι μόνο) δικτύων που ακολουθούν το 802.11 στάνταρ. Αποτελεί ένα πλαίσιο ασφαλείας που επιτρέπει στον απομακρυσμένο χρήστη να πιστοποιηθεί από κάποια κεντρική αρχή του δικτύου. Ο αλγόριθμος πιστοποίησης είναι οποιοσδήποτε αποφασίσει ο σχεδιαστής του δικτύου και υπάρχει η δυνατότητα να συνδυαστούν πολλοί διαφορετικοί αλγόριθμοι για καλύτερο αποτέλεσμα.

Το 802.1X χρησιμοποιεί το πρωτόκολλο EAP(Extensible Authentication Protocol , RFC 2284) για την επικοινωνία απομακρυσμένου χρήστη – διακομιστή πρόσβασης κατά τη διαδικασία πιστοποίησης

Αρχικά η κατάσταση του δικτύου έχει όπως στο σχήμα 3.1:



Σχ. 3.1 802.1X πριν την πιστοποίηση (pre-auth)

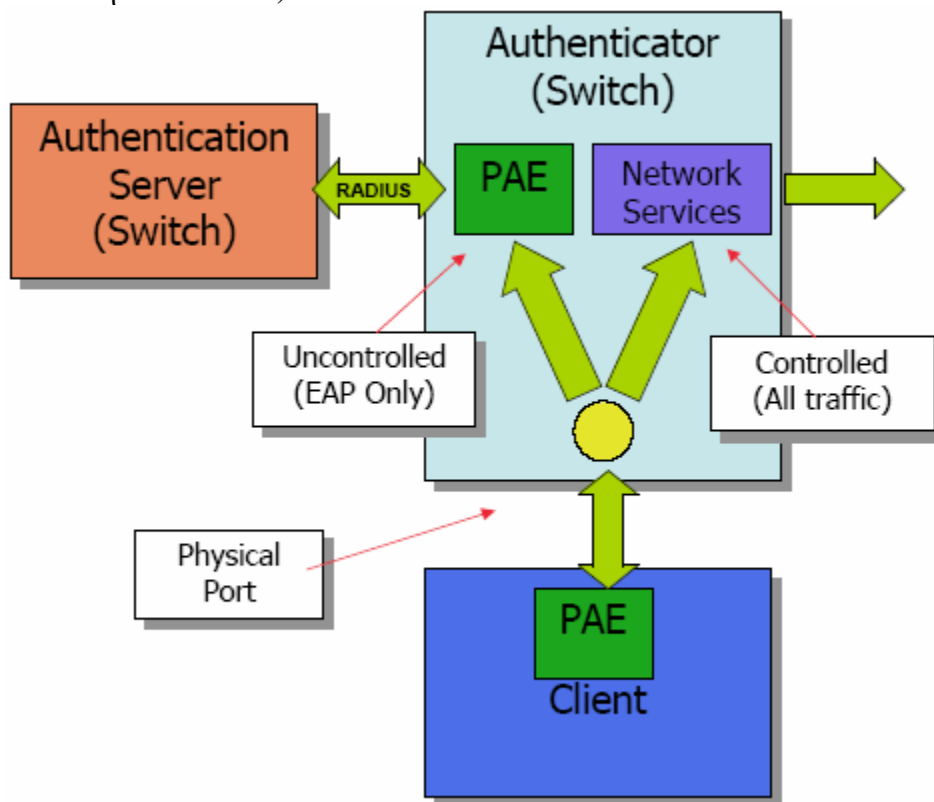
Όπου (PAE) Port Access Entry = το σημείο (port) επικοινωνίας ανάμεσα στον χρήστη και το σημείο πρόσβασης.

Η διαδικασία πιστοποίησης ξεκινά με τον απομακρυσμένο χρήστη να ζητά πρόσβαση απο το σημείο πρόσβασης. Αυτό αποκρίνεται αναγκάζοντάς τον να εισέλθει σε μη πιστοποιημένη κατάσταση που του επιτρέπει να στείλει μόνο EAP μήνυμα έναρξης διαπραγμάτευσης σύνδεσης και του στέλνει (του χρήστη) ένα μήνυμα που του ζητάει να δώσει πληροφορίες πιστοποίησης.

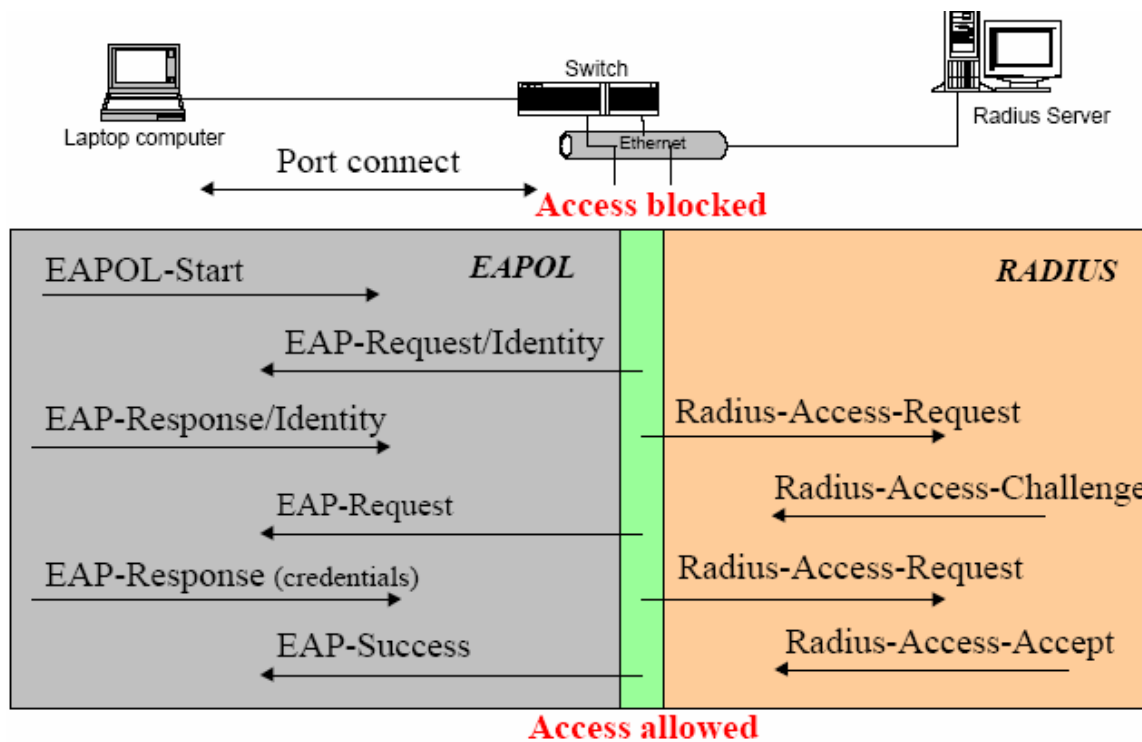
Εφόσον ο χρήστης αποκριθεί με έγκυρα στοιχεία για τον διακομιστή πρόσβασης επιστρέφει απόκριση τύπου ACCEPT και ξεκινά η σύνδεση του χρήστη με το δίκτυο.

Πλεονέκτημα του 802.1X είναι ότι επιτρέπει στους απομακρυσμένους χρήστες που συνδέονται με το σημείο πρόσβασης να χρησιμοποιούν διαφορετικά, δυναμικά εναλλασσόμενα κλειδιά κρυπτογράφησης. Χρησιμοποιώντας το 802.1X μπορούμε μέσω του EAP να στείλουμε τα κλειδιά κρυπτογράφησης στον απομακρυσμένο χρήστη και αυτός να υπολογίσει τα κλειδιά κρυπτογράφησης που θα χρησιμοποιηθούν για την ασφαλή σύνδεση με το σημείο πρόσβασης. Αυτό προσπερνά το πρόβλημα του ενσωματωμένου πρωτόκολλου στο 802.11 WEP (Wired Equivalent Privacy) αναφορικά με την ασφάλεια του.

Όπως φαίνεται με τη χρήση του 802.1X ο χρήστης του σημείου πρόσβασης πρέπει να πιστοποιηθεί προτού του επιτραπεί οποιαδήποτε μεταφορά δεδομένων (συμπεριλαμβανομένων και των αιτήσεων DHCP) στο δίκτυο.



Σχ. 3.11 802.1X μετά από επιτυχημένη πιστοποίηση χρήστη



Σχ 3.III Βήματα για μια επιτυχημένη πιστοποίηση στο 802.1X

Κυριότερα πλεονεκτήματα του 802.1X

- Χρήση ανοιχτών πρωτοκόλλων
Επεκτείνει τα υπάρχοντα standards EAP (RFC 2284) και RADIUS (RFC 2138 , 2139)
Επιτρέπει κεντρική διαχείριση πιστοποίησης χρήστη , διαχείριση κλειδιών ασφαλείας και αναγνώριση ταυτότητας χρήστη
- Αναγνώριση ταυτότητας χρήστη
Η αναγνώριση χρήστη με χρήση αναγνώρισης πρόσβασης δικτύου(Network Access Identifier , RFC 2486) επιτρέπει υποστήριξη περιαγωγής(roaming) πρόσβασης χρήστη (RFC 2607)
- Δυναμική διαχείριση κλειδιών πρόσβασης και επικοινωνίας
- Κεντρική διαχείριση χρήστη
Υποστήριξη του RADIUS (RFC 2138 , 2139) ώστε να έχουμε κεντρική διαχείριση πιστοποίησης , εξουσιοδότησης και χρέωσης των χρηστών
Υποστήριξη του RADIUS/EAP που επιτρέπει ενσωμάτωση EAP πακέτων στο RADIUS.
- Εκτεταμένη υποστήριξη πιστοποίησης
Το EAP είναι σχεδιασμένο ώστε ο διαχειριστής του δικτύου να μπορεί να επιτρέψει διάφορες μεθόδους πιστοποίησης χωρίς αλλαγές στον χρήστη ή την κάρτα ασύρματης πρόσβασής του.
Η υποστήριξη του RFC 2284 συμπεριλαμβάνει υποστήριξη για πιστοποίηση κωδικών με χρήση EAP-MD5 ή κωδικών μίας χρήσης (OTP)

Το 802.1X ελέγχει ανά προκαθορισμένο χρόνο μίας ώρας αν ο χρήστης βρίσκεται στο δίκτυο κατά τρόπο διαφανή στον χρήστη.

3.4 Authorization (εξουσιοδότηση)

3.4.1 Μέθοδοι εξουσιοδότησης

Μόλις πιστοποιήσουμε την ταυτότητα του απομακρυσμένου χρήστη που ζητά πρόσβαση στο δίκτυο μας επόμενο βήμα είναι να τον εξουσιοδοτήσουμε να χρησιμοποιήσει μέρος ή όλες τις υπηρεσίες του δικτύου.

Στη διάθεση μας έχουμε πέντε τρόπους για να εξουσιοδοτήσουμε τον χρήστη:

- TACACS+ Ο διακομιστής πρόσβασης ανταλλάσει πληροφορίες με τον ασφαλή διακομιστή TACACS+. Η βάση δεδομένων του περιέχει πληροφορίες για κάθε χρήστη με ζεύγη χαρακτηριστικό-αξία δηλαδή για κάθε χαρακτηριστικό εξουσιοδότησης έχει και την ανάλογη αξία για το αν επιτρέπεται στον συγκεκριμένο χρήστη να την χρησιμοποιήσει ή όχι.

- If-εξουσιοδότηση Εφόσον ο χρήστης πιστοποιηθεί πρόσβαση μπορεί να χρησιμοποιήσει όποια υπηρεσία του δικτύου ζητήσει.

- Καμία Ο διακομιστής πρόσβασης δεν ζητάει πληροφορίες εξουσιοδότησης , η εξουσιοδότηση δεν συμβαίνει σε αυτό το στάδιο / διεπαφή.

- Τοπική Ο διακομιστής πρόσβασης συμβουλευεται την τοπική βάση δεδομένων του και με βάση το όνομα του απομακρυσμένου χρήστη αποδίδει τα ανάλογα δικαιώματα εξουσιοδότησης. Ένα μειονέκτημα είναι ότι μόνο ένα μικρό σετ λειτουργιών μπορούν να διαχειριστούν μέσω της τοπικής βάσης δεδομένων.

- RADIUS Ο διακομιστής πρόσβασης ζητάει πληροφορίες εξουσιοδότησης από τον ασφαλή διακομιστή RADIUS. Η εξουσιοδότηση μέσω RADIUS προσδιορίζει επακριβώς τα δικαιώματα κάθε χρήστη συσχετίζοντας τα χαρακτηριστικά που βρίσκονται αποθηκευμένα στον ασφαλή διακομιστή RADIUS με τον κάθε απομακρυσμένο χρήστη.

3.5 Εξουσιοδότηση τύπου Radius

3.5.1 Εισαγωγή

Η διαχείριση πολλών χρηστών ταυτόχρονα που καταναλώνουν πόρους του δικτύου μπορεί να δημιουργήσει την ανάγκη για υποστήριξη διαχείρισης. Συνήθως οι χρήστες συνδέονται μέσω «δεξαμενών» modems (στην περίπτωση μας βέβαια δεν έχουμε modems αλλά ασύρματες συνδέσεις το οποίο όμως είναι ίδιο σαν αντιμετώπιση) και αποτελούν τον σύνδεσμό μας με τον έξω κόσμο , έτσι χρειάζεται ιδιαίτερη φροντίδα στην ασφάλεια , εξουσιοδότηση και πιστοποίηση των χρηστών. Αυτό μπορεί να γίνει με τη διαχείριση μίας και μόνο μίας «βάσης δεδομένων» χρηστών η οποία μας επιτρέπει να διαχειριστούμε την εξουσιοδότηση (στοιχεία χρήστη , κωδικός πρόσβασης) όσο και πληροφορίες διαχείρισης όσον αφορά τον τύπο υπηρεσίας που χρησιμοποιεί ο χρήστης – επιτρέπεται να χρησιμοποιήσει.

Τα κύρια χαρακτηριστικά της εξουσιοδότησης τύπου Radius είναι:

1) Μοντέλο πελάτη / εξυπηρετητή

Ο διακομιστής πρόσβασης δικτύου (NAS) λειτουργεί ως πελάτης του εξυπηρετητή radius. Ο πελάτης είναι αρμόδιος για τη διαβίβαση των πληροφοριών χρηστών στους υπεύθυνους εξυπηρετητές Radius, και έπειτα να ενεργήσει μόλις δεχθεί απάντηση. Οι εξυπηρετητές Radius είναι αρμόδιοι για τη λήψη των αιτημάτων σύνδεσης χρηστών, την πιστοποίηση του χρήστη, και τέλος την επιστροφή όλων των πληροφοριών διαμόρφωσης που είναι απαραίτητοι για τον πελάτη ώστε να υλοποιήσει την υπηρεσία στον χρήστη. Ένας εξυπηρετητής radius μπορεί να ενεργήσει ως proxy πελάτης σε άλλους εξυπηρετητές radius ή άλλα είδη εξυπηρετητών πιστοποίησης

2) Ασφάλεια δικτύου

Οι δοσοληψίες μεταξύ του πελάτη και του εξυπηρετητή radius πιστοποιούνται μέσω της χρήσης ενός "κοινού μυστικού", το οποίο δεν στέλνεται ποτέ πέρα από τη μεταξύ τους ζεύξη. Επιπλέον, οποιοδήποτε κωδικός πρόσβασης χρηστών στέλνονται κρυπτογραφημένοι μεταξύ του πελάτη και του εξυπηρετητή radius, για να εξαλείψουν την πιθανότητα ότι κάποιος θα μπορούσε να διεισδύσει στο ανασφαλές δίκτυο και να υποκλέψει τον κωδικό πρόσβασης κάποιου χρήστη.

3) Ευέλικτοι μηχανισμοί πιστοποίησης

Ο εξυπηρετητής radius μπορεί να υποστηρίξει ποικίλες μεθόδους για να πιστοποιήσει έναν χρήστη. Όταν του δοθεί το ζεύγος όνομα χρήστη / κωδικού πρόσβασης από τον χρήστη μπορεί να υποστηρίξει PPP PAP CHAP UNIX login , και διάφορες άλλες μεθόδους πιστοποίησης

4) Δυνατότητα επέκτασης του πρωτοκόλλου

Όλες οι δοσοληψίες αποτελούνται από μεταβλητού μήκους μηνύματα. Νέες τιμές ιδιοτήτων μπορούν να προστεθούν χωρίς να διαταραχθούν οι υπάρχουσες εφαρμογές του πρωτοκόλλου.

3.5.2 Πως λειτουργεί

Όταν ένας πελάτης χρησιμοποιεί εξουσιοδότηση τύπου radius, οποιοσδήποτε χρήστης του πελάτη οφείλει να δώσει πληροφορίες πιστοποίησης στον πελάτη. Αυτό μπορεί να είναι με μια προτροπή εισόδου, όπου ο χρήστης οφείλει να πληκτρολογήσει το όνομα χρήστη και τον κωδικό πρόσβασής του. Εναλλακτικά, ο χρήστης μπορεί να χρησιμοποιήσει ένα πρωτόκολλο ζεύξης πλαισίων όπως είναι το από σημείο σε σημείο πρωτόκολλο (PPP), το οποίο χρησιμοποιεί πακέτα πιστοποίησης που φέρνουν αυτές τις πληροφορίες. Μόλις λάβει ο πελάτης αυτές τις πληροφορίες, μπορεί να επιλέξει να πιστοποιήσει τον χρήστη του μέσω radius. Για να γίνει αυτό, ο πελάτης δημιουργεί ένα αίτημα "Access-Request" που περιέχει πληροφορίες όπως το όνομα χρήστη, ο κωδικός πρόσβασης του, η ταυτότητα του πελάτη και η ταυτότητα της port που ζητά πρόσβαση ο πελάτης. Όταν υπάρχει κωδικός πρόσβασης στο μήνυμα, κωδικοποιείται με τη μέθοδο RSA Digest Algorithm MD5.

Το αίτημα "Access-Request" υποβάλλεται στον εξυπηρετητή radius μέσω του δικτύου. Εάν καμία απάντηση δεν επιστραφεί μέσα σε ένα χρονικό διάστημα, το αίτημα επαναυποβάλλεται για κάποιες φορές ακόμα. Ο πελάτης μπορεί επίσης να διαβιβάσει τα αιτήματα σε εναλλακτικούς εξυπηρετητές σε περίπτωση που ο αρχικός εξυπηρετητής δεν αποκρίνεται (δεν λειτουργεί ή το δίκτυο του είναι απρόσιτο). Ένας εναλλακτικός εξυπηρετητής μπορεί να χρησιμοποιηθεί είτε αφότου αποτύχει κάποιες φορές στον κεντρικό εξυπηρετητή, ή με μία διαδικασία round-robin. Σήμερα αλγόριθμοι επανα-προσπάθειας και υπαναχώρησης μελετούνται και αποτελούν ένα σημαντικό μέρος της ερευνητικής προσπάθειας στον τομέα.

Μόλις λάβει ο εξυπηρετητής radius το αίτημα, προσπαθεί να επικυρώσει την ταυτότητα του αποστολέα πελάτη. Οποιοδήποτε αίτημα από κάποιον πελάτη για τον οποίο ο εξυπηρετητής radius

δεν έχει το κοινό μυστικό μήνυμα ΠΡΕΠΕΙ να απορριφθεί χωρίς να δοθεί απάντηση. Εάν ο πελάτης υπάρχει και είναι έγκυρος στο δίκτυο, ο εξυπηρετητής radius συμβουλεύεται μια βάση δεδομένων των χρηστών για να βρεί το χρήστη του οποίου το όνομα ταιριάζει με αυτό του αιτήματος. Η είσοδος χρηστών στη βάση δεδομένων περιέχει μια λίστα απαιτήσεων που πρέπει να ικανοποιηθούν για να επιτραπεί η πρόσβαση στον χρήστη. Αυτό περιλαμβάνει πάντα την επαλήθευση του κωδικού πρόσβασης, αλλά μπορεί επίσης να συμπεριλαμβάνει τον πελάτη ή το port στο οποίο ο χρήστης εξουσιοδοτείται να έχει πρόσβαση.

Ο εξυπηρετητής radius ΜΠΟΡΕΙ να υποβάλλει αιτήματα άλλων εξυπηρετητών, οπότε σ'αυτή την περίπτωση ενεργεί ως πελάτης. Εάν οποιεσδήποτε ιδιότητες Proxy-State ήταν παρούσες στο αίτημα "access-request", ΠΡΕΠΕΙ να αντιγραφεί χωρίς τροποποιήσεις και με την ίδια σειρά στο πακέτο απάντησης. Άλλες ιδιότητες μπορούν να τοποθετηθούν πριν από, μετά από, ή ακόμα και μεταξύ των ιδιοτήτων του Proxy.

Εάν οποιοσδήποτε όρος δεν ικανοποιείται, ο εξυπηρετητής radius στέλνει μια "Access-Reject" απάντηση που υποδεικνύει ότι αυτό το αίτημα χρήστη είναι άκυρο. Εάν επιδιώκεται, ο εξυπηρετητής ΜΠΟΡΕΙ να περιλαμβάνει ένα μήνυμα κειμένου στο μήνυμα Access-Reject που ΜΠΟΡΕΙ να επιδειχθεί από τον πελάτη στο χρήστη. Καμία άλλη ιδιότητα (εκτός από το Proxy-State) δεν επιτρέπεται σε ένα μήνυμα Access-Reject.

Εάν όλοι οι όροι ικανοποιούνται και ο εξυπηρετητής radius θέλει να εκδώσει μια πρόκληση στην οποία ο χρήστης πρέπει να ανταποκριθεί, ο εξυπηρετητής radius στέλνει ένα μήνυμα "Access-Challenge". Αυτό μπορεί να περιλαμβάνει ένα μήνυμα κειμένου που δείχνει ο πελάτης στον χρήστη για να απαντήσει στην πρόκληση, και ΜΠΟΡΕΙ να περιλαμβάνει κάποια ιδιότητα κατάστασης.

Εάν ο πελάτης λάβει ένα μήνυμα τύπου Access-Challenge και υποστηρίζει το πρότυπο πρόκληση/απάντηση αυτό ΜΠΟΡΕΙ να επιδείξει το μήνυμα κειμένου, ενδεχομένως, στο χρήστη, και να προτρέψει έπειτα το χρήστη για μια απάντηση. Ο πελάτης έπειτα επανα-υποβάλλει το αρχικό μήνυμα Access-Request με μια νέα ταυτότητα χρήστη, αντικαθιστώντας την ιδιότητα User-Password από την κρυπτογραφημένη απάντηση και συμπεριλαμβάνοντας την ιδιότητα κατάστασης από το μήνυμα Access-Challenge, αν υπάρχει. Μόνο περιπτώσεις 0 ή 1 της ιδιότητας κατάστασης ΠΡΕΠΕΙ να είναι παρούσες σε ένα αίτημα. Ο εξυπηρετητής μπορεί να αποκριθεί σε αυτό το νέο μήνυμα Access-Request είτε με Access-Accept, είτε με Access-Reject είτε με κάποιο άλλο μήνυμα Access-Challenge.

Εάν όλοι οι όροι ικανοποιούνται, η λίστα των τιμών διαμόρφωσης της σύνδεσης για το χρήστη τοποθετούνται σε μια απάντηση "Access-Accept". Αυτές οι τιμές συμπεριλαμβάνουν τον τύπο υπηρεσίας (παραδείγματος χάριν: SLIP, PPP, Login User) και όλες τις απαραίτητες τιμές για να παραδοθεί η επιθυμητή υπηρεσία. Παραδείγματος χάριν για τα SLIP ή τα PPP πρωτόκολλα, αυτό μπορεί να συμπεριλάβει τιμές όπως η διεύθυνση IP, η μάσκα υποδικτύου, το MTU, η επιθυμητή συμπίεση και τα επιθυμητά αναγνωριστικά φίλτραρίσματος πακέτων.

3.5.3 Μοντέλο Challenge/Response

Στην πιστοποίηση με τη μέθοδο Challenge/Response, στο χρήστη δίνεται ένας απρόβλεπτος αριθμός και προκαλείται να τον κρυπτογραφήσει και να επιστρέψει το αποτέλεσμα. Οι εξουσιοδοτημένοι χρήστες είναι εξοπλισμένοι με ειδικές συσκευές όπως έξυπνες κάρτες ή λογισμικό που διευκολύνουν τον υπολογισμό της σωστής απάντησης με ευκολία. Οι χρήστες χωρίς δικαίωμα πιστοποίησης, που στερούνται την κατάλληλη συσκευή ή το λογισμικό και που στερούνται τη γνώση του μυστικού κλειδιού απαραίτητου για να μιμηθούν μια τέτοια συσκευή ή λογισμικό, μπορούν μόνο να υποθέσουν στην απάντηση.

Το πακέτο Access-Challenge περιέχει τυπικά ένα μήνυμα απάντησης συμπεριλαμβανομένης μιας πρόκλησης που επιδεικνύεται στο χρήστη, όπως μια αριθμητική αξία απίθανη να επαναληφθεί. Τυπικά αυτό λαμβάνεται από έναν εξωτερικό εξυπηρετητή που ξέρει ποιον τύπο μηχανήματος ή λογισμικού πιστοποίησης κατέχει ο εξουσιοδοτημένος χρήστης και μπορεί επομένως να επιλέξει έναν τυχαίο ή ψευδοτυχαίο αριθμό μη-επανάληψης με μια κατάλληλη βάση και μήκος.

Ο χρήστης έπειτα εισάγει την πρόκληση στη συσκευή του (ή το λογισμικό) και υπολογίζει μια απάντηση, την οποία εισάγει στον πελάτη (το μηχάνημα που χρησιμοποιεί) που το διαβιβάζει εξυπηρετητή radius μέσω ενός δεύτερου μηνύματος Access-Request. Εάν η απάντηση ταιριάζει με την αναμενόμενη απάντηση ο εξυπηρετητής radius απαντά με μήνυμα Access-Accept, αλλιώς με Access-Reject.

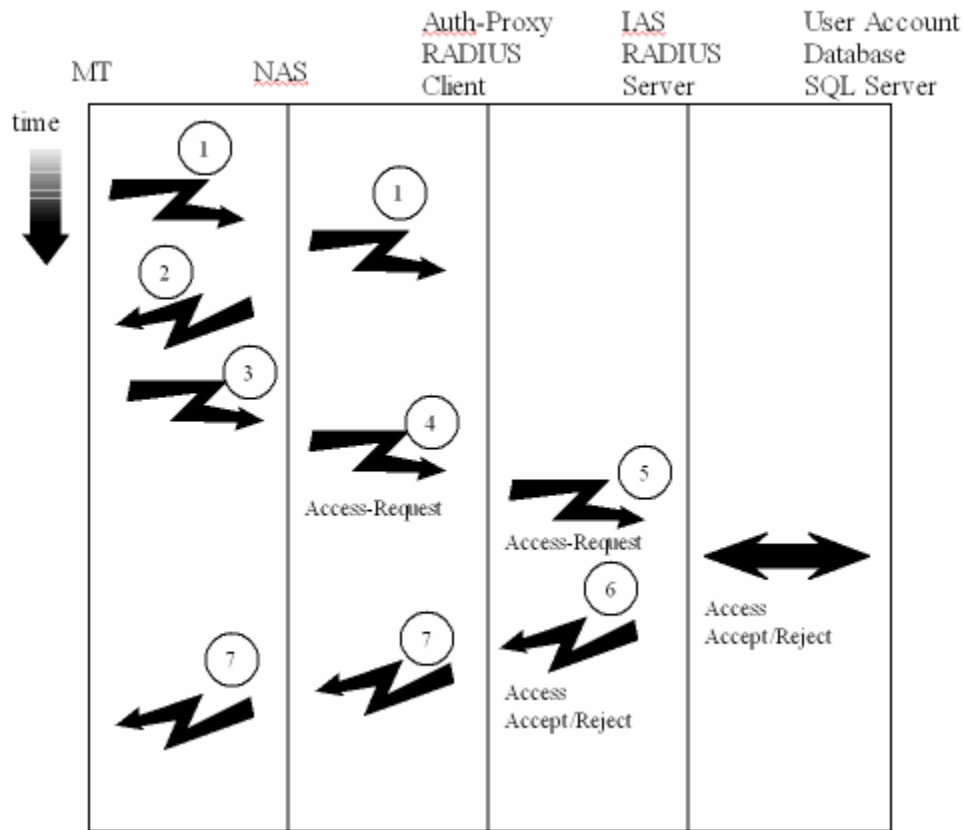
Σαν Παράδειγμα μπορούμε να υποθέσουμε ότι ο NAS στέλνει ένα πακέτο Access-Request στον εξυπηρετητή radius με το αναγνωριστικό του, την port που χρησιμοποιεί, το όνομα χρήστη και τον κωδικό πρόσβασης του (που μπορεί να είναι ένα σταθερό string όπως "πρόσκληση"). Ο εξυπηρετητής απαντάει με ένα μήνυμα Access-Challenge με έλεγχο κατάστασης και ένα μήνυμα απάντησης του τύπου: "πρόκληση(Challenge) 13021981, εισάγετε την απάντησή σας στην πρόσκληση" το οποίο ο NAS παρουσιάζει στον χρήστη. Ο nas περιμένει για την απάντηση και στέλνει ένα NEO Access-Request μήνυμα στον εξυπηρετητή (με νέα ταυτότητα) με τα στοιχεία που έστειλε πριν πλην όμως ότι ο κωδικός χρήστη τώρα είναι η κρυπτογραφημένη απάντηση που έδωσε ο χρήστης και την ιδιότητα κατάστασης που ήρθε με το μήνυμα Access-Challenge. Ο εξυπηρετητής έπειτα απαντάει είτε με ένα μήνυμα τύπου Access-Accept είτε με ένα τύπου Access-Reject ανάλογα με το αν η απάντηση του χρήστη ταιριάζει με την απαιτούμενη τιμή, ή ακόμα μπορεί να στείλει ένα νέο Access-Challenge μήνυμα ώστε να ζητήσει περαιτέρω πληροφορίες πιστοποίησης από τον χρήστη.

3.5.4 Χρήση proxy

Με τη χρήση proxy εξυπηρετητή radius, ο εξυπηρετητής radius λαμβάνει την αίτηση πιστοποίησης χρήστη (ή στοιχεία χρέωσης) από έναν πελάτη radius (όπως είναι ο NAS) και προωθεί την αίτηση σε έναν απομακρυσμένο εξυπηρετητή radius, λαμβάνει την απάντηση από τον απομακρυσμένο εξυπηρετητή και την αποστέλει στον πελάτη του, πιθανώς με αλλαγές ώστε να αντικατοπτρίσει τοπικές πολιτικές διαχείρισης. Μια συνήθης χρήση για proxy εξυπηρετητή radius είναι το roaming. Το Roaming επιτρέπει σε δύο ή παραπάνω δίκτυα να επιτρέπουν την πρόσβαση συνδρομητών ενός δικτύου και στα υπόλοιπα και να έχουν προνόμια πρόσβασης σε υπηρεσίες.

Ο NAS στέλνει το μήνυμα Access-Request στον εξυπηρετητή προώθησης ο οποίος το διαβιβάζει στον απομακρυσμένο εξυπηρετητή. Ο απομακρυσμένος εξυπηρετητής στέλνει μια απάντηση (Access-Accept, Access-Reject, ή Access-Challenge) πίσω στον εξυπηρετητή προώθησης, ο οποίος το στέλνει πίσω στον NAS. Η ιδιότητα όνομα χρήστη ΜΠΟΡΕΙ να περιέχει ένα αναγνωριστικό δικτύου (NAI) για τη διευκόλυνση των λειτουργιών του proxy Radius εξυπηρετητή. Η επιλογή του ποιος εξυπηρετητής θα λάβει την προωθημένη αίτηση ΠΡΕΠΕΙ να βασίζεται στην πιστοποίηση Realm(περιοχής). Η πιστοποίηση Realm ΜΠΟΡΕΙ να είναι το Realm μέρος ενός αναγνωριστικού δικτύου(ένα "ονομαζόμενο realm"). Εναλλακτικά, η επιλογή του ποιος εξυπηρετητής θα λάβει το προωθημένο μήνυμα ΜΠΟΡΕΙ να βασίζεται σε οποιαδήποτε άλλα κριτήρια ο εξυπηρετητής προώθησης έχει τροποποιηθεί να χρησιμοποιεί όπως για παράδειγμα η ταυτότητα του καλούντος σταθμού (ένα "αριθμησιμο Realm").

Για παράδειγμα, ο χρήστης AAA που συνδέεται στο δίκτυο BBB θα λάβει πιστοποίηση χρήστη της μορφής AAA@BBB. Σε αυτήν την περίπτωση το BBB αποτελεί το realm του χρήστη και μπορεί να χρησιμοποιηθεί για την αναγνώρισή του σε περίπτωση roaming.

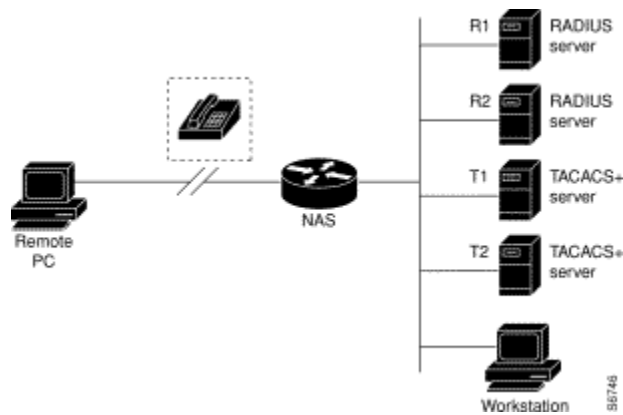


Σχ. 3.IV Βήματα για πιστοποίηση με χρήση proxy διακομιστή

3.6 Accounting

Σε αυτό το κεφάλαιο θα εξετάσουμε την χρέωση χρηστών κυρίως με τη χρήση radius server αφού η υλοποίηση θα βασιστεί πάνω σε αυτό το πρωτόκολλο. Συνοπτικά τα πακέτα χρέωσης αποστέλλονται από τον NAS (Network Access Server) προς τον radius server και περιέχουν πληροφορίες η IP Address του NAS , το όνομα χρήστη στον οποίο αναφέρονται οι πληροφορίες σύνδεσης , ο τύπος του πρωτοκόλλου που χρησιμοποιείται , τα εισερχόμενα / εξερχόμενα πακέτα κ.λ.π.

Όπως και με την πιστοποίηση και εξουσιοδότηση χρήστη τα δεδομένα μπορούν να στέλνονται σε ομάδες radius servers ή και TACACS+ servers και γενικότερα ότι τύπου server έχουμε επιλέξει για την παρακολούθηση της χρέωσης των χρηστών του σημείου πρόσβασης.



Σχ. 3.V Αρχιτεκτονική συστοιχίας πολλαπλών διακομιστών πρόσβασης

Τα πακέτα χρέωσης RADIUS μπορούν να χωριστούν στις ακόλουθες κατηγορίες:

1. Πακέτο τύπου Accounting-Start που περιέχει το όνομα χρήστη , το αναγνωριστικό του NAS server/την ipaddress του και ότι άλλες πληροφορίες δέχεται το NAS.
2. Πακέτο τύπου Accounting-Stop που περιέχει το όνομα χρήστη , το αναγνωριστικό του NAS server/την ipaddress του και ότι άλλες πληροφορίες δέχεται το NAS.
3. Πακέτο τύπου Accounting-On που περιέχει το όνομα χρήστη , το αναγνωριστικό του NAS server/την ipaddress του και υποδεικνύει ότι το συγκεκριμένο NAS έκανε επανεκκίνηση.
4. Πακέτο τύπου Accounting-Off που περιέχει το όνομα χρήστη , το αναγνωριστικό του NAS server/την ipaddress του και υποδεικνύει ότι το NAS σταμάτησε να λειτουργεί
5. Εγγραφή τύπου Accounting-Interim που είναι μια εγγραφή χρέωσης που στέλνεται από το NAS για κάθε χρήστη που έχει πιστοποιηθεί και εξουσιοδοτηθεί πρόσβαση στο σύστημα. Αυτή η υπηρεσία υποστηρίζεται γενικά μόνο από νεώτερες εκδόσεις του NAS.

Οι λίστες μεθόδων χρέωσης είναι συγκεκριμένοι για τον τύπο αίτησης χρέωσης. Το πλαίσιο AAA υποστηρίζει έξι διαφορετικούς τύπους λογιστικής χρέωσης:

- * Network -παρέχει τις πληροφορίες για όλες το PPP, SLIP ή τις ARAP συνδέσεις , συμπεριλαμβανομένων μετρήσεων των πακέτων και bytes που διακινούνται.
- * EXEC-παρέχει τις πληροφορίες για τις συνόδους χρηστών EXEC του NAS στο δίκτυο.
- * Commands-παρέχει τις πληροφορίες για τις εντολές τύπου EXEC που ένας χρήστης ζητάει. Αυτή η μέθοδος παράγει αρχεία λογιστικής χρέωσης για όλες τις εντολές τύπου EXEC, συμπεριλαμβανομένων των εντολών διαμόρφωσης σε global επίπεδο , που συνδέονται με ένα συγκεκριμένο επίπεδο πιστοποίησης χρήστη.
- * Connection-παρέχει τις πληροφορίες για όλες τις εξερχόμενες συνδέσεις που γίνονται από τον NAS στο δίκτυο, όπως Telnet, local-area transport (LAT), TN3270, packet assembler/disassembler (pad), και rlogin.
- * System-παρέχει τις πληροφορίες για γεγονότα σε επίπεδο συστήματος.
- * Resource-παρέχει εγγραφές έναρξης και διακοπής για τις κλήσεις απομακρυσμένων χρηστών που έχουν πιστοποιηθεί, και παρέχει εγγραφές διακοπής για τις κλήσεις που αποτυγχάνουν να πιστοποιηθούν.

Ένα τυπικό πακέτο τύπου Network παρουσιάζεται παρακάτω:

```
Wed Jun 27 04:47:46 2001
NAS-IP-Address = "172.16.25.15"
NAS-Port = 5
User-Name = "fgeorge"
Client-Port-DNIS = "4327528"
Caller-ID = "562"
Acct-Status-Type = Stop
Acct-Authentic = RADIUS
Service-Type = Framed
Acct-Session-Id = "0000000E"
Framed-IP-Address = "10.1.1.2"
Framed-Protocol = PPP
Acct-Input-Octets = 3075
Acct-Output-Octets = 167
Acct-Input-Packets = 39
Acct-Output-Packets = 9
Acct-Session-Time = 171
Acct-Delay-Time = 0
User-Id = "fgeorge"
NAS-Identififier = "172.16.25.15"
```

Ένα τυπικό πακέτο τύπου Connection παρουσιάζεται παρακάτω:

```
Wed Jun 27 04:28:00 2001
NAS-IP-Address = "172.16.25.15"
NAS-Port = 2
User-Name = "fgeorge"
Client-Port-DNIS = "4327528"
Caller-ID = "5622329477"
Acct-Status-Type = Start
Acct-Authentic = RADIUS
Service-Type = Login
Acct-Session-Id = "00000008"
Login-Service = Telnet
Login-IP-Host = "171.68.202.158"
Acct-Delay-Time = 0
User-Id = "fgeorge"
NAS-Identififier = "172.16.25.15"
```

EXEC μέθοδος χρέωσης:

Η μέθοδος χρέωσης EXEC παρέχει πληροφορίες για τις εισόδους χρηστών EXEC σε shell (γραμμή εντολών) στον NAS, συμπεριλαμβανομένων των στοιχείων:

ονόμα χρήστη, ημερομηνία, έναρξη και λήξη της συνόδου και της IP του διακομιστή πρόσβασης.

Το ακόλουθο παράδειγμα παρουσιάζει τις πληροφορίες που περιλαμβάνονται σε ένα αρχείο λογιστικής τύπου RADIUS EXEC για έναν απομακρυσμένο χρήστη:

```
Wed Jun 27 04:26:23 2001
NAS-IP-Address = "172.16.25.15"
NAS-Port = 1
User-Name = "fgeorge"
Client-Port-DNIS = "4327528"
Caller-ID = "5622329483"
```

```
Acct-Status-Type = Start
Acct-Authentic = RADIUS
Service-Type = Exec-User
Acct-Session-Id = "00000006"
Acct-Delay-Time = 0
User-Id = "fgeorge"
NAS-Identifier = "172.16.25.15"
```

System μέθοδος χρέωσης:

Η μέθοδος χρέωσης system παρέχει πληροφορίες για όλα τα συμβάντα επιπέδου συστήματος (παραδείγματος χάριν, όταν το σύστημα κάνει επανεκκίνηση ή όταν η λογιστική χρέωσης σταματά ή ξεκινά).

Command μέθοδος χρέωσης

Η μέθοδος χρέωσης Command παρέχει πληροφορίες για τις εντολές επιπέδου shell EXEC για ένα συγκεκριμένο επίπεδο πιστοποίησης πρόσβασης που εκτελούνται στο NAS. Κάθε εντολή μεθόδου χρέωσης τύπου Command περιλαμβάνει μια λίστα των εντολών που εκτελούνται για εκείνο το επίπεδο πρόσβασης, καθώς επίσης και την ημερομηνία και το χρόνο που κάθε εντολή εκτελέστηκε, και ο χρήστης που τον εκτέλεσε.

Resource μέθοδος χρέωσης

Η υλοποίηση που προτείνει η cisco για τη χρέωση με βάση το πλαίσιο AAA παρέχει υποστήριξη εγγραφών έναρξης και λήξης για τις κλήσεις που έχουν πιστοποιηθεί. Το πρόσθετο χαρακτηριστικό γνώρισμα της παραγωγής των εγγραφών λήξης για τις κλήσεις που αποτυγχάνουν να πιστοποιηθούν υποστηρίζεται επίσης ως μέρος της πιστοποίησης χρηστών. Τέτοιες εγγραφές είναι απαραίτητες για τους διαχειριστές δικτύων ώστε να έχουν μια συνολική εικόνα του δικτύου τους.

Επεκτάσεις της λογιστικής χρέωσης του πλαισίου AAA:

Το πλαίσιο AAA υποστηρίζει δύο επεκτάσεις ώστε να διευκολύνει υπηρεσίες που χρησιμοποιούνται ευρέως. Αυτές είναι η αποστολή πακέτων πληροφοριών χρέωσης σε πολλούς αποδέκτες (broadcast) και η διαχείριση των συνδέσεων client-server μέσω της MIB του Simple Network Management Protocol (SNMP). Αναλυτικότερα έχουμε:

AAA Broadcast Accounting

Το AAA broadcast accounting επιτρέπει τις πληροφορίες χρέωσης να στέλνονται σε πολλούς servers τύπου AAA ταυτόχρονα. Αυτή η λειτουργία επιτρέπει στους παροχείς υπηρεσιών να στέλνουν πληροφορίες λογιστικής χρέωσης στους δικούς τους servers και στους AAA servers των πελατών τους. Παρέχει επίσης τις απαραίτητες πληροφορίες για εφαρμογές φωνής (VoIP).

AAA Session MIB

Το AAA Session MIB επιτρέπει στους χρήστες να ελέγχουν και να σταματούν τις πιστοποιημένες συνδέσεις των clients χρησιμοποιώντας το Simple Network Management Protocol(SNMP). Τα δεδομένα του client παρουσιάζονται έτσι ώστε να συσχετίζονται άμεσα με τις πληροφορίες χρέωσης που αναφέρονται είτε από τον server RADIUS είτε από τον server τύπου TACACS+. Η σύνοδος MIB τύπου AAA παρέχει τις ακόλουθες πληροφορίες:

- * Στατιστικά για κάθε λειτουργία του AAA(όταν χρησιμοποιείται από κοινού με την εντολή show radius statistics)

- * Κατάσταση των servers που παρέχουν λειτουργίες τύπου AAA

- * Ταυτότητες των εξωτερικών servers τύπου AAA

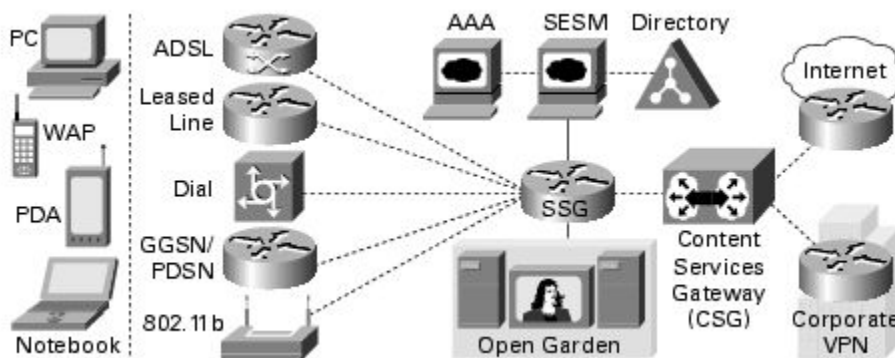
- * Πληροφορίες σε πραγματικό χρόνο(όπως οι χρόνοι idle των clients), που παρέχουν τα πρόσθετα κριτήρια προς χρήση από τα δίκτυα SNMP για την αξιολόγηση εάν πρέπει να ολοκληρωθεί μια κλήση σε εξέλιξη.

3.7 Cisco Service Selection Gateway

3.7.1 Εισαγωγή

Η Cisco Service Selection Gateway (SSG) είναι μια επιπρόσθετη βιβλιοθήκη του λογισμικού του Cisco IOS® που επιτρέπει στους φορείς παροχής υπηρεσιών να δημιουργήσουν νέες υπηρεσίες βασισμένες στην βασισμένη στην χρήση χρέωση.

Το Cisco SSG επιτρέπει στους συνδρομητές να επιλέξουν δυναμικά τις υπηρεσίες on-demand που επιθυμούν. Αυτό βελτιώνει την ευελιξία και την ευκολία για τους συνδρομητές και τους δίνει την δυνατότητα να εκτελούν πολλαπλές υπηρεσίες βασισμένες στην ίδια πλατφόρμα ταυτόχρονα. Παραδείγματος χάριν, η πρόσβαση Διαδικτύου μπορεί να είναι υπηρεσία σταθερής χρέωσης, αλλά πρόσθετες κατόπιν παραγγελίας υπηρεσίες όπως η εταιρική τηλεργασία, τυχερά παιχνίδια κ.α. είναι πρόσφορα για χρέωση πέραν της μηνιαίας σταθερής χρέωσης



Σχ. 3.VI Αρχιτεκτονική Cisco SSG

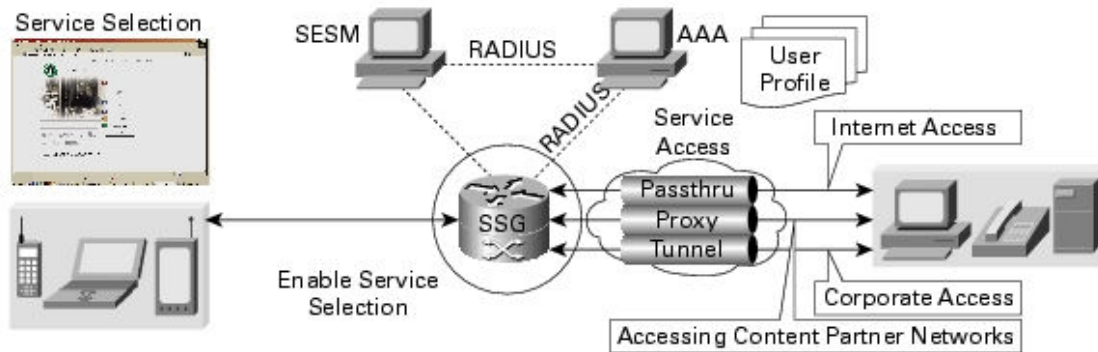
Μαζί με το Cisco SSG, ο Cisco Subscriber Edge Services Manager (SESM) επιτρέπει στον φορέα παροχής υπηρεσιών για να δημιουργήσει ένα web portal που παρουσιάζει στους συνδρομητές τις επιλογές των υπηρεσιών, επιτρέποντάς τους να συνδεθούν και να αποσυνδεθούν από τις διάφορες υπηρεσίες χρησιμοποιώντας τον browser τους. Έχοντας τη δύναμη να προσφέρει πολλαπλές υπηρεσίες σε μια ενοποιημένη δικτυακή πύλη, οι παροχείς υπηρεσιών και οι παροχείς διαδικτύου (ISPs) μπορούν να υλοποιήσουν πρωτοποριακές υπηρεσίες με ανταγωνιστική χρέωση.

3.7.2 Τύποι υπηρεσιών Cisco SSG

- Pass-through-Traffic- εξυπηρετείται από το συνήθες routing ή από πίνακα επόμενου κόμβου. Το Cisco SSG διαδραματίζει το ρόλο του AAA αλλά δεν εκτελεί υπηρεσίες μετάφρασης διευθύνσεων (NAT) Αυτός ο τύπος υπηρεσιών χρησιμοποιείται συνήθως στην τυποποιημένη πρόσβαση Διαδικτύου.

- Tunnel- Το Cisco SSG εκκινεί το Layer 2 Tunneling Protocol (L2TP) στο μακρινό διακομιστή πρόσβασης L2TP (LNS). Το NAT εκτελείται ανάμεσα στην IP διεύθυνση που ορίζεται στον χρήστη και στην διεύθυνση που ορίζεται από τον LNS Αυτός ο τύπος υπηρεσιών είναι ιδανικός για τις υπηρεσίες που είναι ήδη εξοπλισμένες με LNS όπως είναι οι υπηρεσίες βασισμένες σε εταιρικές ή τρίτου πηγές δεδομένων.

- Proxy- ο μακρινός διακομιστής εκτελεί το AAA και υποστηρίζει το πρωτόκολλο NAT όταν ορίζει μια IP διεύθυνση. Αυτός ο τύπος υπηρεσιών είναι καλά ταιριαγμένος όταν μια υπηρεσία πρέπει να χρησιμοποιήσει το AAA framework.



Σχ. 3.VII Τύποι υπηρεσιών Cisco SSG

3.7.3 Χαρακτηριστικά του Cisco SSG

Ταυτόχρονος ή διαδοχικός τρόπος πρόσβασης υπηρεσιών

Οι υπηρεσίες Cisco SSG μπορούν να διαμορφωθούν για ταυτόχρονη ή διαδοχική πρόσβαση. Η ταυτόχρονη πρόσβαση επιτρέπει στους χρήστες να συνδεθούν στην υπηρεσία ενώ βρίσκονται ταυτόχρονα συνδεδεμένοι με άλλες υπηρεσίες. Η διαδοχική πρόσβαση απαιτεί από τον χρήστη να αποσυνδεθεί από όλες τις υπόλοιπες υπηρεσίες (της πλατφόρμας) προτού αποκτήσει πρόσβαση σε μια υπηρεσία που έχει καθοριστεί για διαδοχική πρόσβαση. Επιπλέον, οι υπηρεσίες Cisco SSG μπορούν να διαμορφωθούν σε αμοιβαία αποκλειόμενες ομάδες, όπου οι υπηρεσίες μέσα σε μία ομάδα είναι διαδοχικής πρόσβασης, ενώ οι υπηρεσίες έξω από την ομάδα μπορούν να είναι ταυτόχρονης πρόσβασης.

TCP Redirect

Το Cisco SSG επιτρέπει στους χρήστες να πιστοποιηθούν χωρίς να γνωρίζουν τη URL διεύθυνση του Cisco SESM portal. Εάν ένας χρήστης που δεν έχει πιστοποιηθεί στείλει πακέτα σε μια προκαθορισμένη ομάδα από TCP ports, το Cisco SSG στέλνει εκείνα τα πακέτα σε μια ομάδα διακομιστών υπεύθυνων για τη λειτουργία του web portal . Το web portal διαχειρίζεται τα εισερχόμενα πακέτα κατά τον κατάλληλο τρόπο επιστρέφοντας τη σελίδα αίτησης πρόσβασης.

Domain Name System Redirection

Όταν το Cisco SSG λαμβάνει ένα αίτημα τύπου Domain Name System(dns), εκτελεί ταίριασμα ονόματος domain χρησιμοποιώντας το χαρακτηριστικό ονόματος domain. Εάν βρεθεί μια αντιστοιχία, το αίτημα προωθείται στο διακομιστή dns για την αντίστοιχη υπηρεσία. Εάν δεν βρεθεί αντιστοιχία και ο χρήστης έχει συνδεθεί σε μια υπηρεσία που έχει πρόσβαση στο διαδίκτυο, το αίτημα προωθείται στην πρώτη υπηρεσία στη λίστα των υπηρεσιών που έχει πρόσβαση ο χρήστης και η οποία διαθέτει πρόσβαση στο διαδίκτυο. Εάν δεν βρεθεί αντιστοιχία και ο χρήστης δεν έχει συνδεθεί σε κάποια υπηρεσία με πρόσβαση στο διαδίκτυο , η αίτηση προωθείται στο διακομιστή DNS που ορίζεται στη στοιβιά TCP/IP του χρήστη.

ανοχή ελαττωμάτων DNS

Το Cisco SSG μπορεί να διαμορφωθεί για να συνεργαστεί με έναν κεντρικό διακομιστή DNS ή δύο διακομιστές σε διαμόρφωση ανοχής σε αστοχία υλικού. Με βάση έναν εσωτερικό αλγόριθμο,

οι αιτήσεις τύπου dns προωθούνται στον δευτερεύον διακομιστή εάν ο κεντρικός διακομιστής αποτύχει να απαντήσει με απάντηση τύπου DNS μέσα σε ορισμένο χρονικό πλαίσιο.

Auto-Logon

Η αυτόματη-σύνδεση συνδέει τους χρήστες στις υπηρεσίες που έχουν προεπιλέξει μόλις συνδεθούν στο SSG. Οι χρήστες μπορούν να προσπεράσουν το Cisco SESM και να έχουν απευθείας πρόσβαση στις προεπιλεγμένες υπηρεσίες τους.

Auto-Domain Αυτόματος-περιοχή

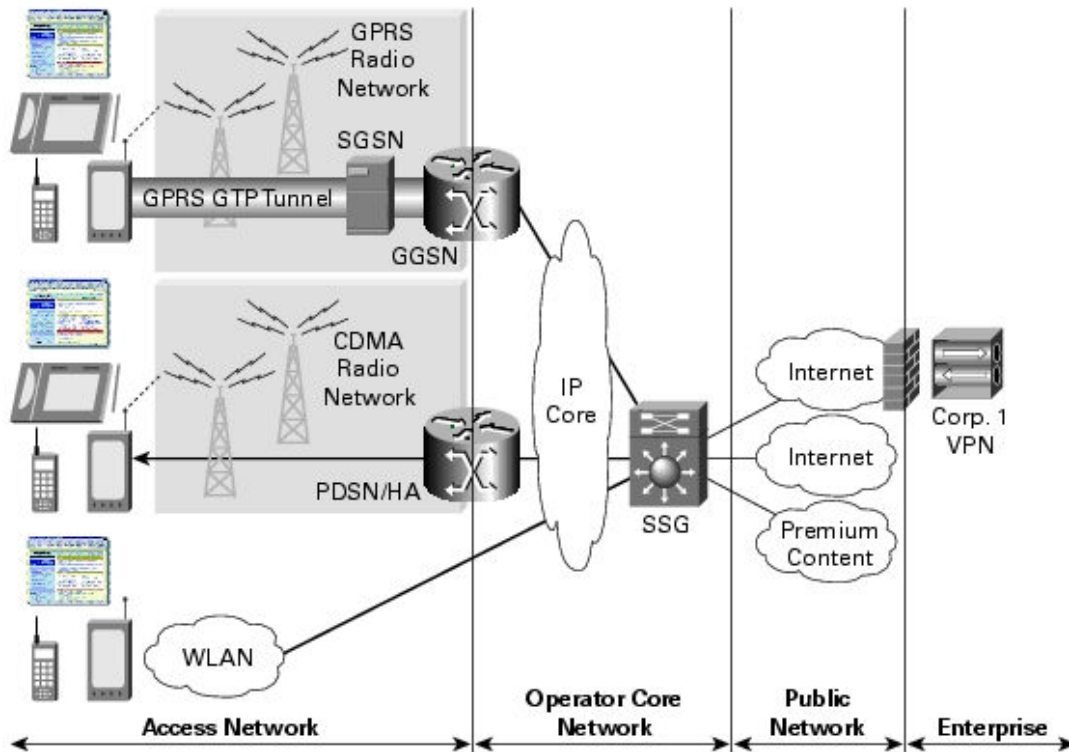
Χρησιμοποιώντας το χαρακτηριστικό Auto-Domain στο Cisco SSG, είναι δυνατό να συνδεθεί αυτόματα κάποιος χρήστης βάσει είτε του ονόματος του Access point μέσω του οποίου προσπαθεί να συνδεθεί, είτε βάσει του user@service ονόματος χρήστη που εισάγει ο χρήστης κατά την πρόσβασή του στο δίκτυο. Οι χρήστες μπορούν να παρακάμψουν το Cisco SESM και να έχουν πρόσβαση σε κάποια υπηρεσία, όπως ένα εταιρικό intranet ή το δίκτυο κάποιου φορέα υπηρεσιών διαδικτύου (isp). Το Auto-Domain χαρακτηριστικό υποστηρίζεται επίσης από το Cisco SESM, εάν ο χρήστης εισάγει κάποιο σωστά δομημένο όνομα χρήστη κατά τη διαδικασία πρόσβασης στο λογαριασμό του (login). Το Auto-Domain στο Cisco SSG επιτρέπει σε έναν χρήστη να συνδεθεί με υπηρεσίες είτε τύπου proxy είτε τύπου L2TP.

Open Garden - Ανοικτός κήπος

Όλοι οι μη πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση στις υπηρεσίες τύπου Open Garden. Ο διαχειριστής του δικτύου μπορεί να χρησιμοποιήσει αυτό το χαρακτηριστικό ώστε να έχει διαχειριστικές υπηρεσίες, να παρέχει υπηρεσίες σε δοκιμαστική μορφή ή να διαφημίσει υπηρεσίες στους τελικούς χρήστες. Το προφίλ υπηρεσίας για τις Open Garden υπηρεσίες διαμορφώνεται τοπικά (local-profile), ή remotely σε κάποια βάση δεδομένων τύπου RADIUS,LDAP ή ORACLE. Τα δεδομένα που στέλνονται/λαμβάνονται από υπηρεσίες Open Garden δεν υπόκεινται σε λογιστική χρέωση.

3.7.4 One platform to rule them all

Το Cisco SSG μπορεί να συναθροίσει και φιλοξενήσει τη δημιουργία υπηρεσιών και διαχείριση για πολλαπλούς τύπους δικτύων, συμπεριλαμβανομένων των General Packet Radio Service (GPRS), Universal Mobile Telecommunications Service (UMTS), Code Division Multiple Access (CDMA), Integrated Digital Enhanced Network (iDEN), Circuit Switched Data (CSD), Wireless LAN (WLAN) και άλλων, που προσφέρει τελικά σε όλους τους χρήστες μια ενιαία εμπειρία υπηρεσίας ανεξάρτητα από τον τύπο πρόσβασης. Από την οπτική γωνία του παροχέα υπηρεσιών, αυτό επιτρέπει την κοινή διαχείριση της υπηρεσίας παρότι αυτή προσφέρεται σε πολλαπλούς τύπους δικτύων. Ένα δίκτυο βασισμένο στο Cisco SSG προσφέρει ευελιξία επέκτασης καθώς νέες υπηρεσίες μπορούν να προστεθούν χωρίς μεγάλες τροποποιήσεις του υφιστάμενου δικτύου.

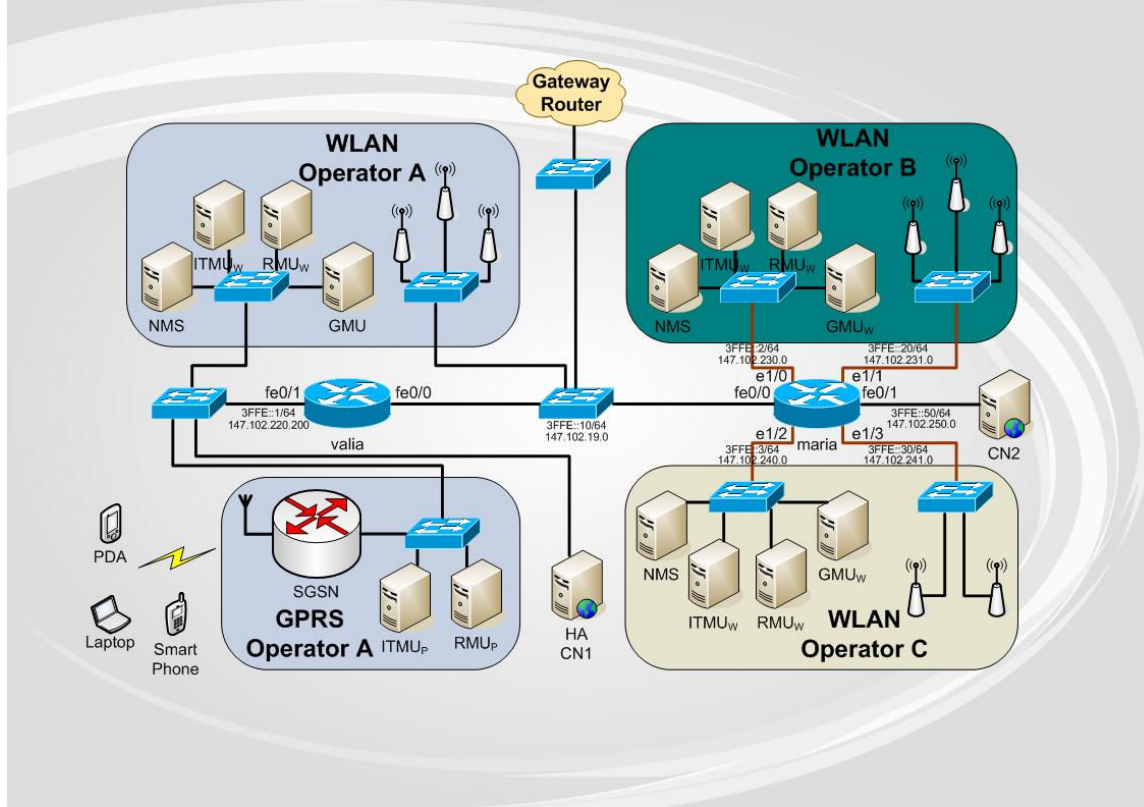


Σχ 3.VIII Αρχιτεκτονική κοινής δημιουργίας και διαχείρισης υπηρεσιών

4. Αρχιτεκτονική πλατφόρμας

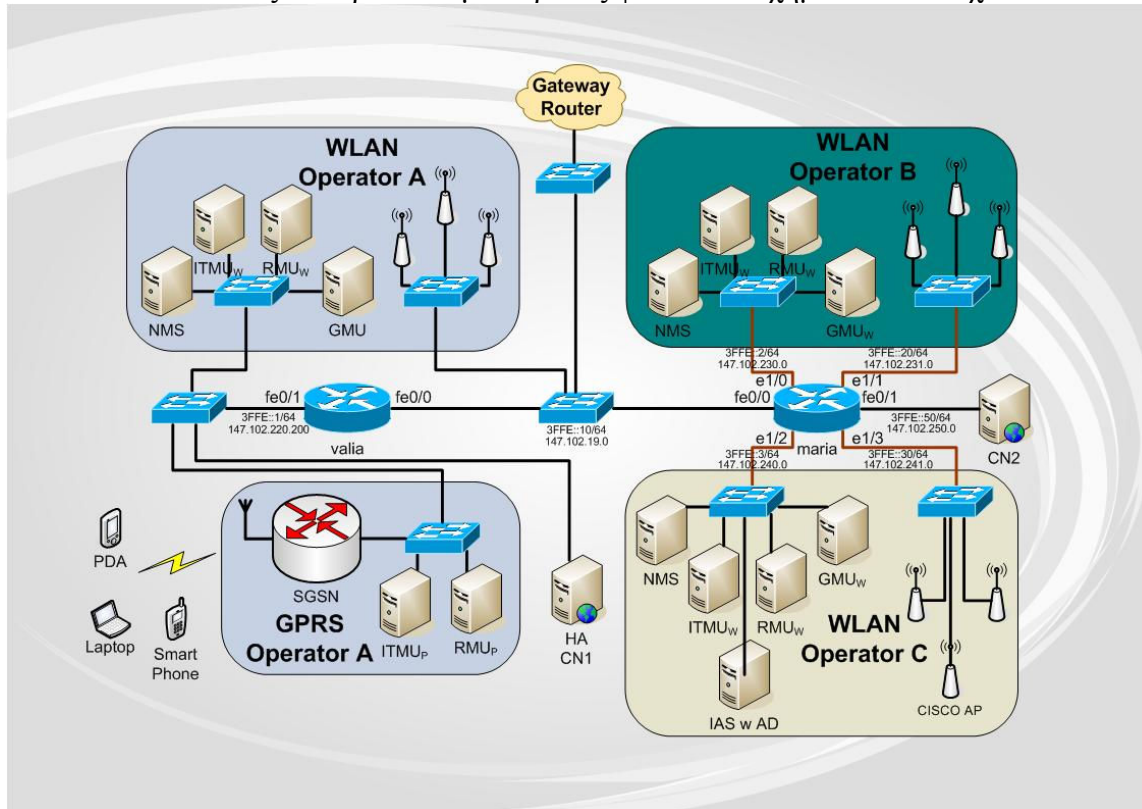
4.1 Τοπολογία δικτύου

Για τις ανάγκες της διπλωματικής χρησιμοποιήθηκε η πλατφόρμα WONDERLAND η τοπολογία της οποίας φαίνεται στο Σχ. 4.1



Σχ. 4.1 Η αρχική τοπολογία του δικτύου

Το access point (cisco Aironet) βρίσκεται στο subnet 147.102.241.0 , ο web server που χρησιμοποιήθηκε για δοκιμές της πλατφόρμας AAA (πιστοποίηση, εξουσιοδότηση, λογιστική χρέωση) στην διεύθυνση 147.102.250.21 , ενώ ο router πιστοποίησης με χρήση radius server βρίσκεται στο subnet 147.102.240.0 στην IP διεύθυνση 147.102.240.200. Η εφαρμογή παρακολούθησης και λογιστικής χρέωσης βρίσκεται στο μηχάνημα “IAS w AD” στην διεύθυνση 147.102.240.2 . Όλες οι παραπάνω μετατροπές φαίνονται σχηματικά στο Σχ. 4.II



Σχ. 4.II Η τελική τοπολογία του δικτύου

Όπως φαίνεται στο Σχήμα 2, με την προσθήκη στην τοπολογία του δικτύου του μηχανήματος με τον IAS, επιτύχαμε την κεντρική πιστοποίηση των χρηστών με χρήση windows authentication. Ο router στο 147.102.240.200 συνεργάζεται με τον IAS στο 147.102.240.2 ο οποίος (IAS) επιστρέφει έγκριση πιστοποίησης χρήστη εφόσον ο χρήστης είναι εγγεγραμμένος στο active directory του μηχανήματος. Παράλληλα όλα τα στοιχεία λογιστικής χρέωσης εγγράφονται στον SQL server στο μηχάνημα του IAS server ώστε να είναι δυνατή η περαιτέρω επεξεργασία τους από το πρόγραμμα παρακολούθησης χρηστών και λογιστικής χρέωσης Account++.

4.2 router (ACL , interfaces)

Ο cisco router που χρησιμοποιήθηκε στην πλατφόρμα είναι το μοντέλο 2611XM. Πρόκειται για έναν μεσαίων δυνατοτήτων router στη σειρά προϊόντων της cisco , ο οποίος όμως τελικά όμως αποδείχθηκε ικανός για οποιαδήποτε σχεδόν εργασία απαιτήσε η πλατφόρμα AAA. Όπως φαίνεται και στο σχήμα 2 , η επικοινωνία διεξάγεται στα interfaces e1/2 , e1/3 και fe0/1. Πιο συγκεκριμένα, οι πληροφορίες για προσπάθεια εισόδου χρήστη στο wlan έρχεται από το interface 1/3 όπου βρίσκεται το access point. Κατόπιν, ερωτάται ο IAS στο interface 1/2 αν μπορεί να επιτραπεί η είσοδος στον χρήστη και αν επιτύχει, ο χρήστης έχει πρόσβαση στα interfaces e1/2 και fe0/1 (όπου βρίσκεται ο δοκιμαστικός web server στον οποίο ζητεί πρόσβαση ο χρήστης στο υποθετικό σενάριο λειτουργίας)

Το configuration του router ώστε να επιτευχθεί η επικοινωνία μεταξύ των διαφόρων interfaces είναι το εξής:

```
!  
! Last configuration change at 14:28:28 ATH Thu Dec 23 2004 by test  
! NVRAM config last updated at 14:42:18 ATH Thu Dec 23 2004 by test  
!  
version 12.3  
service timestamps debug datetime msec  
service timestamps log datetime msec  
no service password-encryption  
!  
hostname maria  
!  
boot-start-marker  
boot-end-marker  
!  
enable secret 5 $1$J5C8$cfHFc98n7wRctNHkAI1pN0  
enable password enable  
!  
clock timezone ATH 2  
no network-clock-participate slot 1  
no network-clock-participate wic 0  
aaa new-model  
!  
!  
aaa authentication login default group radius  
aaa authentication login console_line none  
aaa authorization auth-proxy default group radius  
aaa accounting auth-proxy default start-stop group radius  
aaa session-id common  
ip subnet-zero  
!  
!  
ip name-server 147.102.220.11  
no ip dhcp conflict logging  
ip dhcp excluded-address 147.102.231.51 147.102.231.255  
ip dhcp excluded-address 147.102.241.51 147.102.241.255  
!  
ip dhcp pool Operator_B  
  network 147.102.231.0 255.255.255.0  
  default-router 147.102.231.200  
  dns-server 147.102.220.11  
  domain-name operb.ntua.gr  
!  
ip dhcp pool Operator_C  
  network 147.102.241.0 255.255.255.0  
  default-router 147.102.241.200  
  dns-server 147.102.220.11  
  domain-name operc.ntua.gr  
!  
ip cef  
ip auth-proxy auth-proxy-banner http _ Welcome to Wonderland Operator 3 _  
ip auth-proxy inactivity-timer 10
```



```

half-duplex
ipv6 address 3FFE:0:0:30::/64 eui-64
ipv6 nd reachable-time 1800000
ipv6 nd ns-interval 1000
!
ip http server
ip http access-class 61
ip http authentication aaa
no ip http secure-server
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0 147.102.19.200
ip route 147.102.220.0 255.255.255.0 FastEthernet0/0 147.102.19.202
!
!
access-list 61 deny any
access-list 100 permit udp any any eq bootps
access-list 100 permit ip 147.102.231.0 0.0.0.255 any
access-list 101 permit udp any any eq bootps
access-list 101 permit ip 147.102.241.0 0.0.0.255 any
access-list 116 permit tcp any any eq domain
access-list 116 permit udp any any eq bootps
access-list 116 permit udp any any eq domain
access-list 116 deny tcp any any
access-list 116 deny udp any any
access-list 116 deny icmp any any
ipv6 route 3FFE:0:0:1::/64 FastEthernet0/0 3FFE::10:20C:CEFF:FEC9:98A0
!
!
snmp-server community public RO
snmp-server enable traps tty
radius-server host 147.102.240.2 auth-port 1645 acct-port 1646 key telecomnikiyam5566
!
!
!
!
dial-peer cor custom
!
!
!
!
!
line con 0
 password consolemaria
 logging synchronous
line aux 0
line vty 0 4
 exec-timeout 0 0
 password vtermmaria
!
ntp master
!
!
end

```

4.3 active directory

Σημαντικό ρόλο στην απρόσκοπτη επικοινωνία της πλατφόρμας διαδραμάτισε το active directory. Η πολιτική πιστοποίησης χρηστών μέσω της χρήσης windows authentication υποδείκνυε τη χρήση ενός κεντρικού συστήματος για την προσθήκη/μεταβολή στοιχείων/διαγραφή χρηστών. Το αρχικό σύστημα των windows 2003 server με τη χρήση των windows user names and accounts δεν προσέφερε την ευελιξία και ασφάλεια του κατά πολύ βελτιωμένου active directory. Για αυτούς τους λόγους υλοποιήσαμε μια πλατφόρμα active directory στο μηχάνημα που βρίσκεται εγκατεστημένος ο IAS και ο SQL server (147.102.240.2).

Αρχικά, από το CLI των windows 2003 server enterprise με την εντολή dcpromo δημιουργήσαμε τον domain controller ADServer.ntua.gr στο καινούργιο domain ntua.gr .Το καινούργιο domain δημιουργήθηκε μέσα σε ένα καινούργιο forest . Το όνομα του καινούργιου domain forest είναι το ίδιο με το όνομα του domain name/controller (ADServer.ntua.gr). Σαν DNS server ορίστηκε το localhost, δηλαδή το ίδιο μηχάνημα στο οποίο ορίστηκε ο domain controller. Τέλος, για λόγους συμβατότητας με προηγούμενες εκδόσεις των windows (προ 2000) , ώστε σε περίπτωση που επιθυμούμε να μπορούν να συνδεθούν στην πλατφόρμα AAA δημιουργήσαμε το NetBIOS όνομα για το domain ADSERVER.

Αναλυτικά, οι ρυθμίσεις που πραγματοποιήθηκαν ήταν οι εξής:
Configure this server as the first domain controller in a new forest of domain trees.

The new domain name is ADServer.ntua.gr. This is also the name of the new forest.

The NetBIOS name of the domain is ADSERVER

*Database folder: E:\ADDATABASE
Log file folder: E:\ADDATABASE
SYSVOL folder: E:\ADDATABASE\SYSVOL*

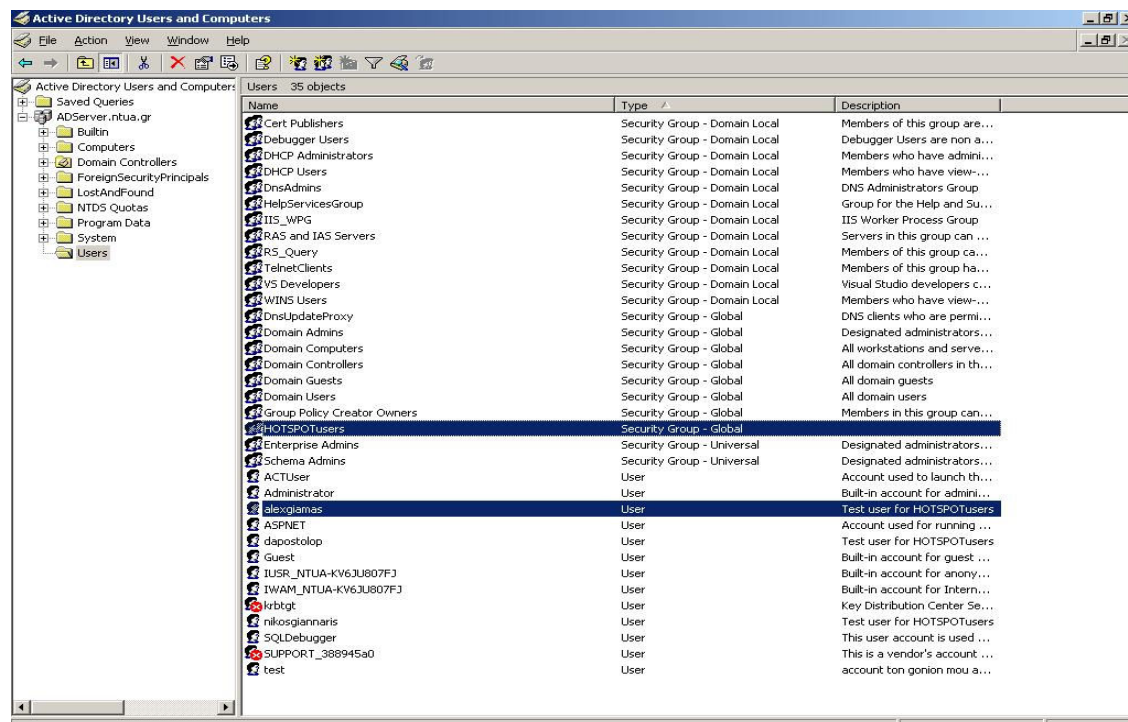
The DNS service will be installed and configured on this computer. This computer will be configured to use this DNS server as its preferred DNS server.

The password of the new domain administrator will be the same as the password of the administrator of this computer.

Active Directory is now installed on this computer for the domain ADServer.ntua.gr.

This domain controller is assigned to the site Default-First-Site-Name. Sites are managed with the Active Directory Sites and Services administrative tool.

Με τη χρήση του active directory μπορούμε να έχουμε τον κεντρικό έλεγχο της πιστοποίησης των χρηστών όπως φαίνεται στο Σχ. 4.III

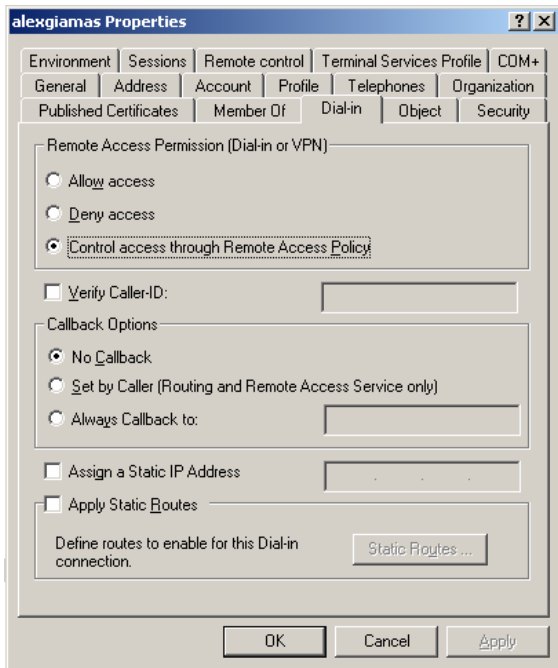


Σχ. 4.III Κεντρικός έλεγχος πιστοποίησης χρηστών με χρήση του Active Directory

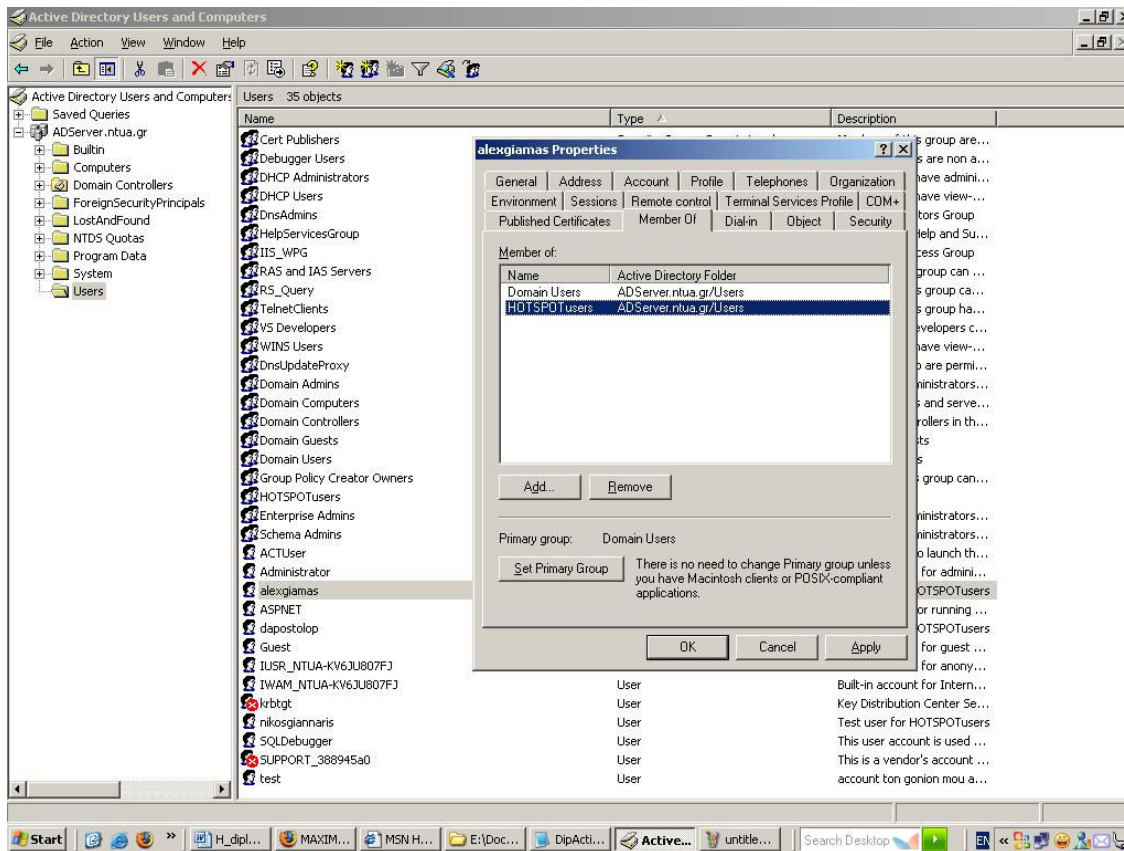
Στην εικόνα φαίνονται ο χρήστης που δημιουργήσαμε με την εφαρμογή Account++ “alexgiannas” καθώς και η ομάδα χρηστών HOTSPOTusers στην οποία θέλουμε να προσθέτουμε τον εκάστοτε χρήστη ώστε να υπόκειται σε group policy όσον αφορά την πιστοποίησή του στο δίκτυο.

Στο Σχ.4.IV φαίνεται παραστατικά το πως ο τυχαίος χρήστης που προσθέσαμε μέσω της εφαρμογής Account++ γίνεται αυτόματα μέλος του group HOTSPOTusers . Παρατηρούμε επίσης τις 16 καρτέλες για παραμετροποίηση και προσθήκη χαρακτηριστικών του χρήστη. Στην καρτέλα Dial-in ορίζουμε με την προσθήκη του χρήστη στο Active Directory να ορίζεται η πιστοποίηση του χρήστη μέσω του Remote Access Policy.

Το Remote Access Policy που είναι μια επιπλέον δυνατότητα που παρέχεται με το Active Directory , παρέχει κεντρική διαχείριση ανά group των υπηρεσιών πιστοποίησης που θα παρέχονται στον τελικό χρήστη με τρόπο διαφανή προς τον χρήστη.



Σχ. 4.IV Καρτέλα Dial-in της εφαρμογής Active Directory Users and Computers



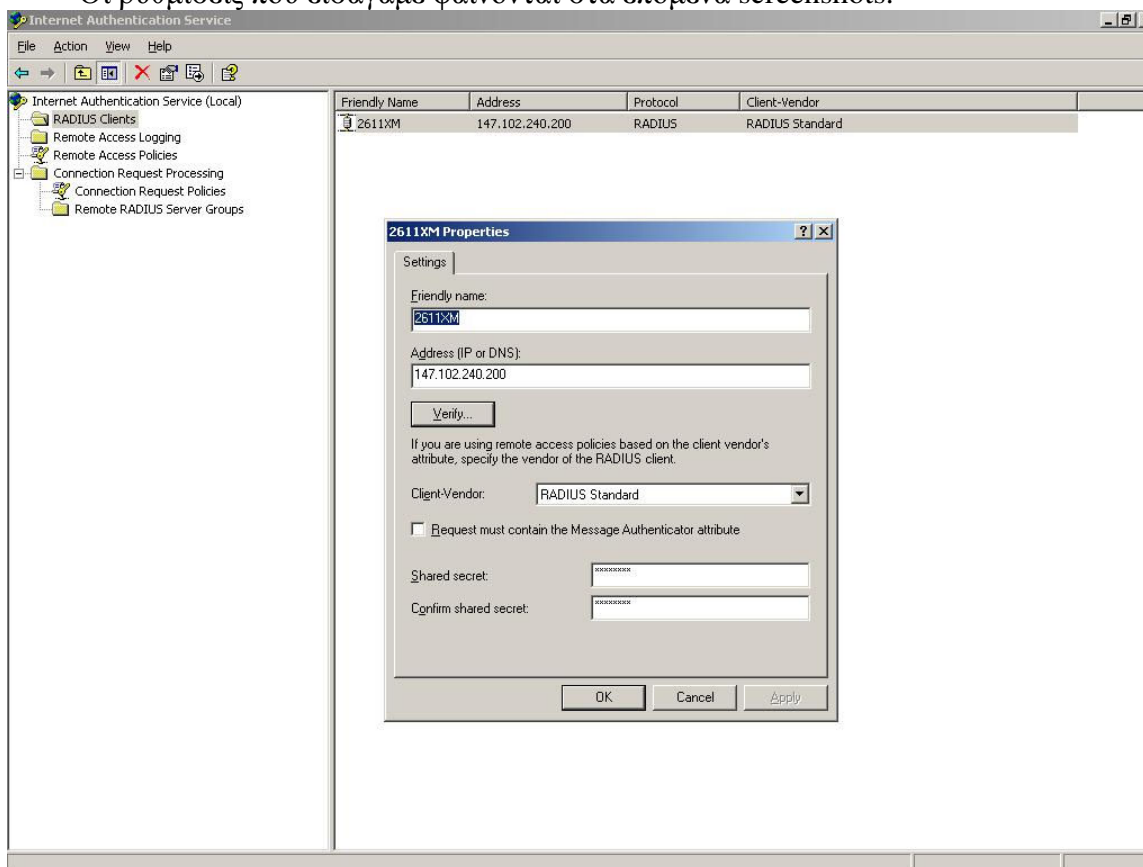
Σχ. 4.V Properties χρήστη της εφαρμογής Active Directory Users and Computers

4.4 IAS

Ο Internet Authentication Service (IAS) είναι μια εφαρμογή που περιέχεται στα windows NT 4 και μεταγενέστερα , η οποία επεκτείνει τις δυνατότητες πιστοποίησης του λειτουργικού συστήματος (authentication). Ο IAS υλοποιεί το πρότυπο του Remote Authentication Dial-In User Service (RADIUS).

Στην πλατφόρμα που χρησιμοποιούμε ο IAS είναι το μέσο πιστοποίησης των χρηστών. Τα πακέτα ερώτησης πιστοποίησης από τον cisco router πηγαίνουν στον IAS και αυτός απαντάει είτε με πακέτο πιστοποίησης πρόσβασης είτε με πακέτο άρνησης πρόσβασης. Ο ρόλος του IAS είναι πολύπλευρος. Εκτός από τις απαντήσεις πιστοποίησης που οφείλει να δώσει , κρατάει στατιστικά για την κίνηση του δικτύου και επίσης οφείλει να αρνηθεί την πρόσβαση σε όσους χρήστες δεν τηρούν τα κριτήρια που έχουμε ορίσει.

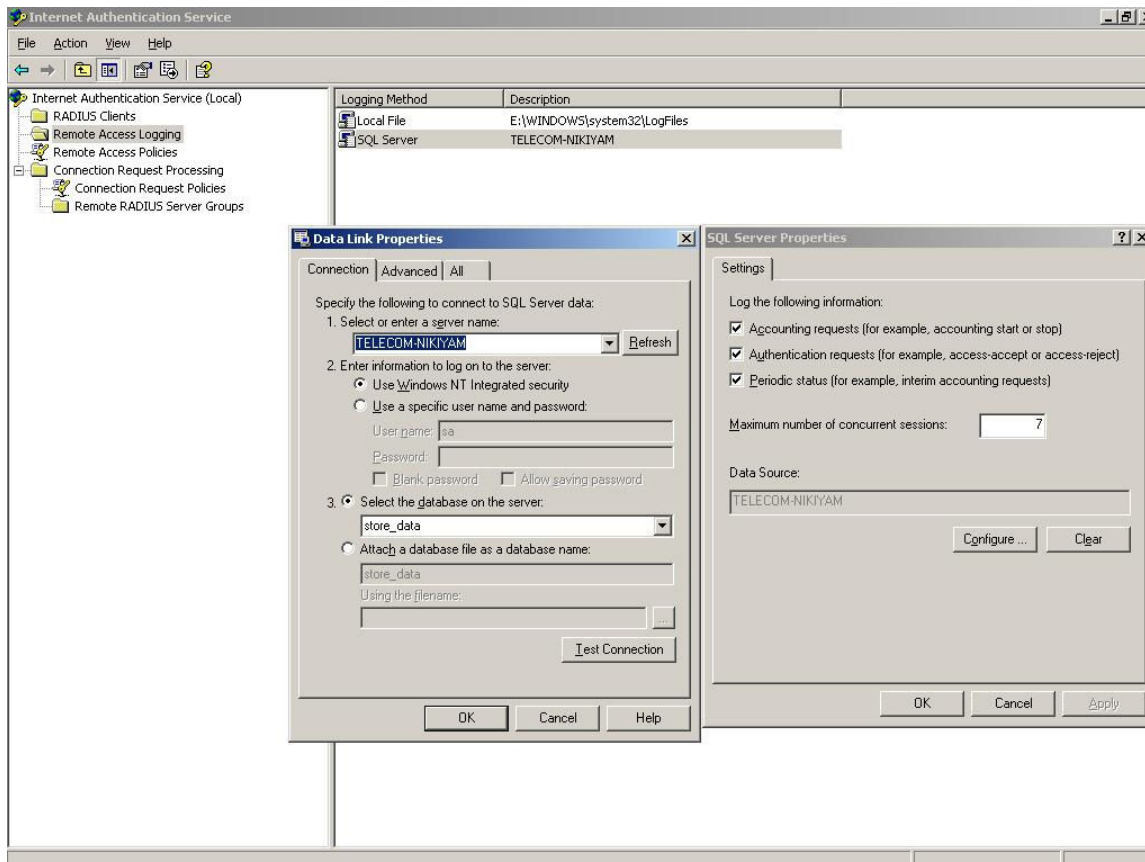
Οι ρυθμίσεις που εισάγαμε φαίνονται στα επόμενα screenshots.



Σχ. 4.VI Επικοινωνία IAS - router

Στο παραπάνω σχήμα βλέπουμε τις ρυθμίσεις που κάναμε ώστε να επικοινωνεί ο IAS με τον router. Η επικοινωνία τους στηρίζεται στο shared secret το οποίο δρα σαν password για την μεταξύ τους επικοινωνία.

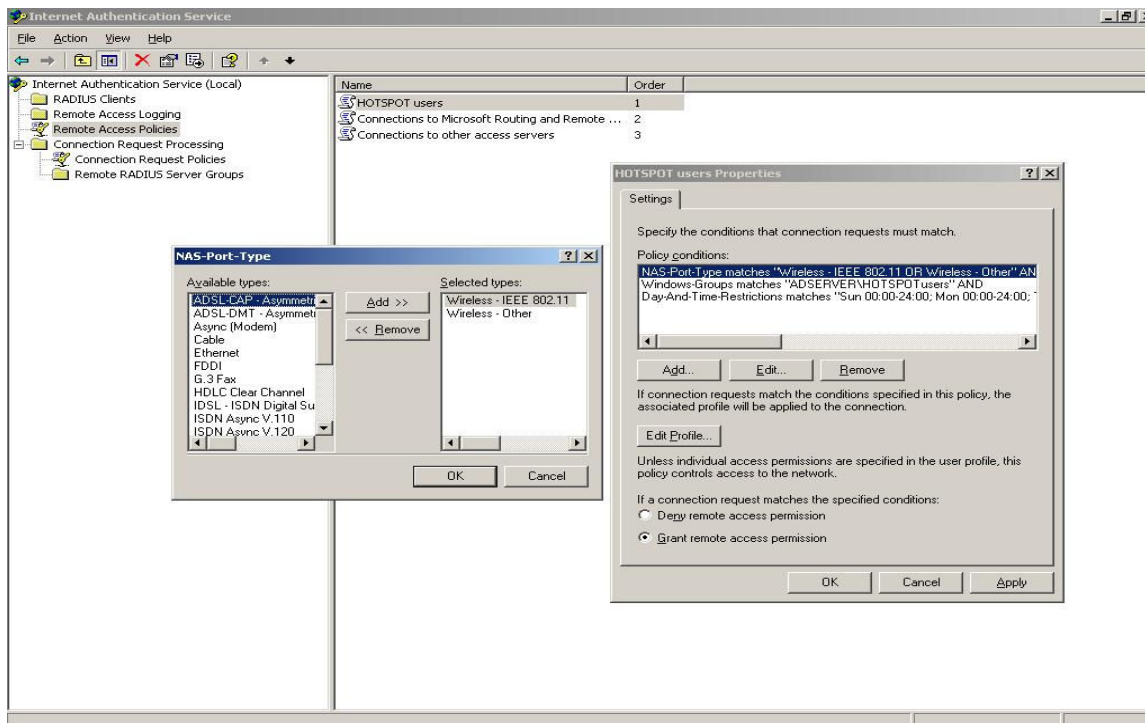
Στην επόμενη οθόνη βλέπουμε το configuration της σύνδεσης με τον SQL server. Έχουμε ρυθμίσει να κάνουμε log το σύνολο των πληροφοριών που μπορούμε. Το όνομα του SQL server είναι το όνομα του υπολογιστή αφού επικοινωνούν μέσω named pipes στο localhost. Έχουμε δημιουργήσει μια καινούργια database στον SQL server με το όνομα store_data στην οποία εγγράφουμε τα δεδομένα που έρχονται από τον router.



Σχ. 4.VII Επικοινωνία του IAS με τον SQL server

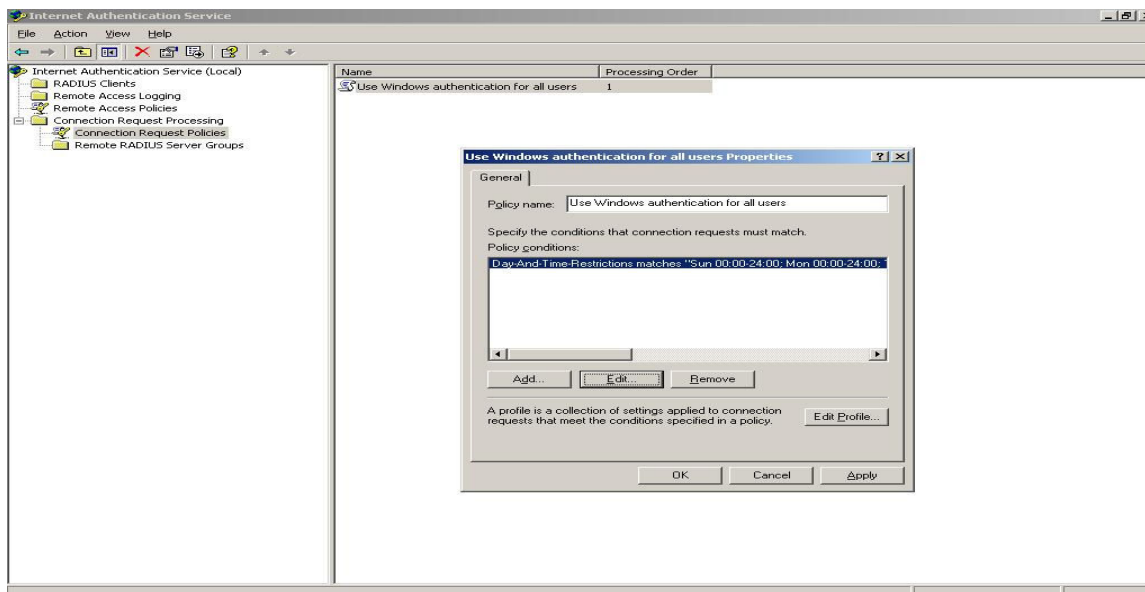
Στο Σχήμα 4.VIII βλέπουμε τις ρυθμίσεις στο remote access policies. Έχουμε εισάγει τρεις κανόνες για να επιτρέψουμε την είσοδο του χρήστη στο δίκτυο. Πρώτος και θεμελιώδης κανόνας είναι οι αιτήσεις να προέρχονται από το wireless interface του router (e1/3). Αυτό το επιλέγουμε στο NAS-Port-Type ίσο με Wireless-Other/IEEE 802.11 . Ο δεύτερος και εξίσου σημαντικός κανόνας είναι ο χρήστης να ανήκει στο windows group ADSERVER\HOTSPOTusers. Το group ορίζεται από το όνομα του active directory domain (ADSERVER) και το όνομα του group (HOTSPOTusers). Αυτός ο κανόνας χρησιμεύει στο να χρησιμοποιείται το windows authentication ώστε να μπορούμε να προσθέτουμε και αφαιρούμε από χρήστες το δικαίωμα πρόσβασης ανάλογα με το windows group στο οποίο ανήκουν.

Τέλος, ο τρίτος κανόνας δείχνει απλά ότι δεν θα υπάρχουν χρονικοί περιορισμοί όσον αφορά το σε ποιες ώρες και ημέρες της εβδομάδας θα επιτρέπεται η είσοδος σε χρήστες που πληρούν τα δύο παραπάνω κριτήρια. Θα μπορούσαμε βέβαια αν θέλαμε να βάλουμε κάποιο χρονικό περιορισμό ώστε π.χ. να επιτρέπεται η είσοδος στους χρήστες του windows group HOTSPOTusers κατά τις καθημερινές ενώ τα σαββατοκύριακα να επιτρέπεται η είσοδος των WEEKENDusers ώστε να υλοποιήσουμε διαφορετικές χρεώσεις ή πολιτικές πρόσβασης στο δίκτυο ανάλογα με την ημέρα της εβδομάδας.



Σχ. 4.VIII IAS Remote Access Policies

Στην τελευταία εικόνα βλέπουμε τις ρυθμίσεις στο Connection Request Policies όπου έχουμε επίσης ορίσει να μην υπάρχει κανένας περιορισμός στην ώρα και ημέρα της εβδομάδας κατά την οποία θα επιτρέπεται η είσοδος στους χρήστες. Η ρύθμιση εδώ έχει καθολική (global) ισχύ, δηλαδή όποιος κανόνας εφαρμοστεί στο Connection Request Policies αφορά το σύνολο των χρηστών που θέλουμε να πιστοποιήσουμε. Με αυτήν την παράμετρο μπορούμε να περιορίσουμε το σύνολο των χρηστών σε πρόσβαση π.χ. κατά τις καθημερινές ή όποια άλλη χρονική περίοδο επιθυμούμε.



Σχ. 4.IX Ρυθμίσεις Connection Request Policies

5. Αρχιτεκτονική συστήματος

5.1 Δομή βάσης SQL server

5.1.1 Εισαγωγή

Σημαντικό ρόλο στην επιτυχία της πλατφόρμας πιστοποίησης και λογιστικής χρέωσης χρηστών διαδραμάτισε ο SQL server. Η άρρηκτη επικοινωνία του με τον IAS είναι ακρογωνιαίος λίθος στην απρόσκοπτη λειτουργία της πλατφόρμας αφού όλα τα στατιστικά στοιχεία αποθηκεύονται στον SQL server με τη μορφή records και ανά πάσα στιγμή μπορούν να ανακτηθούν με χρήση απλών SQL queries.

Όλα τα στοιχεία λογιστικής χρέωσης που στέλνει ο authentication proxy (router) αφού επεξεργαστούν από τον IAS αποθηκεύονται με την χρήση customized procedures στον SQL server σε τέσσερις πίνακες που έχουν δημιουργηθεί για αυτόν τον σκοπό. Στους ίδιους πίνακες αποθηκεύονται επίσης στοιχεία που αφορούν τους λογαριασμούς των χρηστών της πλατφόρμας. Τα στοιχεία αυτά αφορούν στοιχεία πιστοποίησης /στατιστικής επεξεργασίας όπως διεύθυνση,τηλέφωνο, email κλπ όσο και στοιχεία που αφορούν το λογαριασμό του χρήστη όπως διαθέσιμες ώρες στο λογαριασμό του και τύπος συνδρομής (προπληρωμένη/αορίστου χρόνου).

5.1.2 Βήματα παραμετροποίησης

Αρχικά, δημιουργήσαμε την SQL βάση και τους πίνακες στους οποίους σκοπεύαμε να αποθηκεύσουμε τα δεδομένα με τις εξής SQL εντολές.

```
create database store_data
CREATE TABLE dbo.Subscriber (
event_id nvarchar(64) PRIMARY KEY NOT NULL,
name1 nvarchar(64) NOT NULL,
surname1 nvarchar(128) NOT NULL,
Telno nvarchar (14) NOT NULL,
Email nvarchar(100) NOT NULL,
Con nvarchar (14) NOT NULL
)
GO
CREATE TABLE dbo.Subscriber2 (
event_id nvarchar(64) NOT NULL
REFERENCES Subscriber (event_id)
ON UPDATE CASCADE
ON DELETE CASCADE,
Mobno nvarchar (14),
address nvarchar(128),
)
GO
```

Το επόμενο βήμα ήταν να δημιουργήσουμε customized procedures ώστε να αποθηκεύει ο IAS στον SQL server τα πακέτα χρέωσης/πιστοποίησης που στέλνει ο authentication proxy.

Report Event.sql

```
IF EXISTS (
SELECT * FROM dbo.sysobjects
WHERE (
id = OBJECT_ID('dbo.event_data') AND
OBJECTPROPERTY(id, 'IsUserTable') = 1
)
```

```

)
DROP TABLE dbo.event_data

IF EXISTS (
SELECT * FROM dbo.sysobjects
WHERE (
id = OBJECT_ID('dbo.event_main') AND
OBJECTPROPERTY(id, 'IsUserTable') = 1
)
)
DROP TABLE dbo.event_main

IF EXISTS (
SELECT * FROM dbo.sysobjects
WHERE (
id = OBJECT_ID('dbo.report_event') AND
OBJECTPROPERTY(id, 'IsProcedure') = 1
)
)
DROP PROCEDURE dbo.report_event

GO

CREATE TABLE dbo.event_main (
event_id uniqueidentifier NOT NULL
PRIMARY KEY CLUSTERED,
record_timestamp datetime NOT NULL
)

GO

CREATE TABLE dbo.event_data (
event_id uniqueidentifier NOT NULL
REFERENCES event_main (event_id)
ON UPDATE CASCADE
ON DELETE CASCADE,
attribute_type nvarchar (64) NOT NULL,
attribute_value nvarchar (1024) NULL
)

GO

CREATE PROCEDURE dbo.report_event
@doc ntext
AS

SET NOCOUNT ON

DECLARE @idoc int
EXEC sp_xml_preparedocument @idoc OUTPUT, @doc

DECLARE @event_id uniqueidentifier
SET @event_id = NEWID()

```

```

DECLARE @record_timestamp datetime
SET @record_timestamp = GETUTCDATE()

BEGIN TRANSACTION

INSERT dbo.event_main VALUES (
@event_id,
@record_timestamp
)

INSERT dbo.event_data
SELECT
@event_id,
attribute_type,
attribute_value
FROM OPENXML(@idoc, '/Event/*')
WITH (
attribute_type varchar(64) '@mp:localname',
attribute_value nvarchar(1024) 'child::text()'
)

COMMIT TRANSACTION

EXEC sp_xml_removedocument @idoc

GO

```

Σχ. 5.I Report Event.sql

Στα Σχ 5.II – 5.V βλέπουμε τυπικές καταστάσεις της βάσης μέσω του εργαλείου SQL query analyzer

	event_id	attribute_type	attribute_value
1	6F9619FF-8B86-D011-B42D-00C...	username	agiamas
2	6F9619FF-8B86-D011-B42D-00C...	username	dnikit
3	6F9619FF-8B86-D011-B42D-00C...	username	test1
4	6F9619FF-8B86-D011-B42D-00C...	username	test2
5	9178E9AE-A8C3-4BAD-8231-31D...	Computer-Name	TELECOM-NIKIYAM
6	9178E9AE-A8C3-4BAD-8231-31D...	Event-Source	RAS
7	9178E9AE-A8C3-4BAD-8231-31D...	NAS-IP-Address	169.254.201.247
8	9178E9AE-A8C3-4BAD-8231-31D...	Acct-Session-Id	0
9	9178E9AE-A8C3-4BAD-8231-31D...	Acct-Status-Type	7
10	9178E9AE-A8C3-4BAD-8231-31D...	Client-IP-Address	169.254.201.247
11	9178E9AE-A8C3-4BAD-8231-31D...	Provider-Type	2
12	9178E9AE-A8C3-4BAD-8231-31D...	Packet-Type	4
13	9178E9AE-A8C3-4BAD-8231-31D...	Reason-Code	0
14	C2E4A8E4-CF42-45BE-8CD6-2AB...	Computer-Name	TELECOM-NIKIYAM
15	C2E4A8E4-CF42-45BE-8CD6-2AB...	Event-Source	RAS
16	C2E4A8E4-CF42-45BE-8CD6-2AB...	NAS-IP-Address	147.102.19.198
17	C2E4A8E4-CF42-45BE-8CD6-2AB...	Acct-Session-Id	0
18	C2E4A8E4-CF42-45BE-8CD6-2AB...	Acct-Status-Type	7
19	C2E4A8E4-CF42-45BE-8CD6-2AB...	Client-IP-Address	147.102.19.198
20	C2E4A8E4-CF42-45BE-8CD6-2AB...	Provider-Type	2
21	C2E4A8E4-CF42-45BE-8CD6-2AB...	Packet-Type	4
22	C2E4A8E4-CF42-45BE-8CD6-2AB...	Reason-Code	0
23	E3C8575A-3638-4D83-9511-C65...	Computer-Name	TELECOM-NIKIYAM
24	E3C8575A-3638-4D83-9511-C65...	Event-Source	RAS
25	E3C8575A-3638-4D83-9511-C65...	NAS-IP-Address	147.102.19.198
26	E3C8575A-3638-4D83-9511-C65...	Acct-Session-Id	0
27	E3C8575A-3638-4D83-9511-C65...	Acct-Status-Type	8
28	E3C8575A-3638-4D83-9511-C65...	Client-IP-Address	147.102.19.198
29	E3C8575A-3638-4D83-9511-C65...	Provider-Type	2

Σχ. 5.II Στιγμιότυπο του πίνακα event_data

	event_id	record_timestamp
1	6F9619FF-8B86-D011-B42D-00C04FC964AA	2005-05-28 00:00:00.000
2	6F9619FF-8B86-D011-B42D-00C04FC964AB	2005-05-28 00:00:00.000
3	6F9619FF-8B86-D011-B42D-00C04FC964AC	2005-05-28 00:00:00.000
4	6F9619FF-8B86-D011-B42D-00C04FC964AD	2005-05-28 00:00:00.000
5	6F9619FF-8B86-D011-B42D-00C04FC964AE	2005-05-28 00:00:00.000
6	6F9619FF-8B86-D011-B42D-00C04FC964FA	2005-01-05 00:00:00.000
7	6F9619FF-8B86-D011-B42D-00C04FC964FB	2005-05-28 00:00:00.000
8	6F9619FF-8B86-D011-B42D-00C04FC964FC	2005-06-02 13:12:14.123
9	6F9619FF-8B86-D011-B42D-00C04FC964FF	1980-01-01 00:00:00.000
10	C2E4A8E4-CF42-45BE-8CD6-2ABC29C0BA0	2005-06-07 08:38:39.127
11	9178E9AE-A8C3-4BAD-8231-31DF2867837C	2005-06-06 23:04:02.457
12	E3C8575A-3638-4D83-9511-C65B95F3B493	2005-06-07 08:54:53.830

Σχ. 5.III Στιγμιότυπο του πίνακα event_main

	event_id	name1	surname1	Telno	Email
1	alekosgiamas	alekos	giamas	32238032809	dcodoidcio
2	alekoulisyamas	alekoulis	yamas	2117508123	agiamas@hotmail
3	alexgiamas	alex	giamas	6945102539	agiamas@hotmail
4	dapostolop	d	apostolop	2102343212	apostold@photor
5	dnikit	d	nikit	2109343506	dnikit@telecom.
6	nikosgiannaris	nikos	giannaris	2106699696	nnickgian@hotmail
7	nikospapazoglou	nikos	papazoglou	6972304293	nikospapazoglou

Σχ. 5.IV Στιγμιότυπο του πίνακα Subscriber

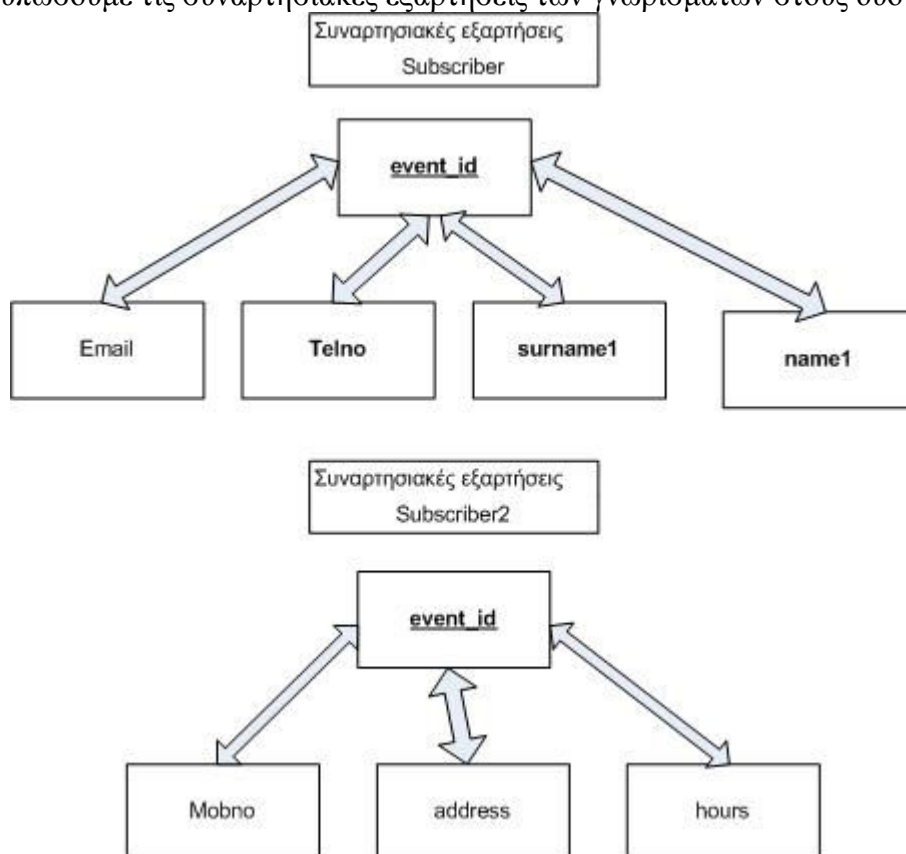
	event_id	Mobno	address	hours
2	alekoulisyamas	023083048	sdodsdsioioj 293	10
3	alekosgiamas	7798798789	hiohiohi	7070
4	dnikit	33202390	ddiosdjio	10
5	alexgiamas	2902902902	g.zografou 8	20
6	nikosgiannaris	6686689689	goggogu	700

Σχ. 5.V Στιγμιότυπο του πίνακα Subscriber2

Αμέσως παρατηρούμε ότι τα στοιχεία του συνδρομητή αποθηκεύονται σε δύο πίνακες αντί για έναν. Αυτό έχει γίνει για να βρίσκεται η βάση σε κανονικοποιημένη μορφή. Μια σχεσιακή βάση δεδομένων (RDBMS) που βρίσκεται σε κανονικοποιημένη μορφή έχει το πλεονέκτημα ότι ελαχιστοποιούνται οι επαναλήψεις κατά τις λειτουργίες της βάσης. Επίσης, με το να βρίσκεται η βάση σε κανονικοποιημένη μορφή ελαχιστοποιούνται οι ανωμαλίες κατά την εισαγωγή, διαγραφή και τροποποίηση των δεδομένων της βάσης.

Το γνώρισμα κλειδί των δύο πινάκων είναι το event_id το οποίο είναι μοναδικό για κάθε συνδρομητή. Αφού το γνώρισμα κλειδί δεν έχει πολλαπλά γνωρίσματα και καμία σχέση δεν έχει με ατομικά γνωρίσματα ή εμφωλευμένες σχέσεις η βάση βρίσκεται σίγουρα σε δεύτερη κανονική μορφή (2NF).

Για να εξακριβώσουμε αν η βάση βρίσκεται σε τρίτη κανονική μορφή (3NF) χρειάζεται να αποτυπώσουμε τις συναρτησιακές εξαρτήσεις των γνωρισμάτων στους δύο πίνακες.



Σχ. 5.VI Πίνακας συναρτησιακών εξαρτήσεων

Από τον πίνακα συναρτησιακών εξαρτήσεων παρατηρούμε ότι δεν υπάρχει μεταβατική εξάρτηση από το πρωτεύον κλειδί σε ένα γνώρισμα που δεν ανήκει στο κλειδί. Δηλαδή δεν υπάρχει στις εξαρτήσεις ένα γνώρισμα που δεν είναι μέρος του κλειδιού και το οποίο να προσδιορίζεται συναρτησιακά από ένα άλλο γνώρισμα που δεν αποτελεί κλειδί. Έτσι τελικά προσδιορίζουμε ότι η βάση μας βρίσκεται σε Τρίτη κανονική μορφή (3NF).

Επιπλέον, χωρίζοντας τα δεδομένα σε δύο πίνακες ανάλογα με το αν θεωρούνται υποχρεωτικά ή προαιρετικά καταφέρνουμε να ελαττώσουμε το φόρτο της βάσης σε εντολές αναζήτησης, προσθήκης, τροποποίησης ή διαγραφής δεδομένων. Ένας πίνακας με λιγότερα columns χρειάζεται σαφώς λιγότερο χρόνο για να διασχισθεί από έναν πιο «βαρύ» πίνακα με περισσότερα columns, πάντα αναφερόμενοι στον ίδιο αριθμό πλειάδων(records).

5.2 Δομή εφαρμογής

5.2.1 Εισαγωγή

Η εφαρμογή Account++ αποτελεί τον συνδεδετικό κρίκο και την εποπτεύουσα αρχή των ανομοιογενών τεχνολογιών που συνεργάστηκαν για τη λειτουργία της πλατφόρμας πιστοποίησης και λογιστικής χρέωσης. Αφενός επηρεάζει κάθε λειτουργία της πλατφόρμας αφετέρου μπορεί να μας δώσει χρήσιμες πληροφορίες αρκετά γρήγορα όσον αφορά στατιστικά δεδομένα των εγγεγραμμένων χρηστών της πλατφόρμας.

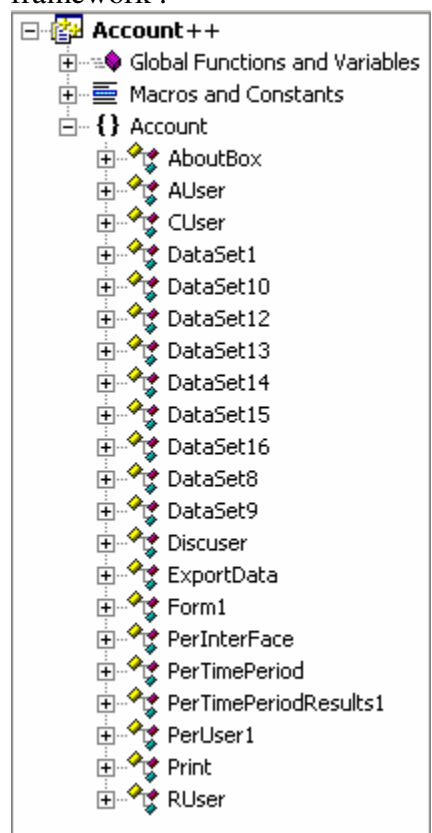
Η ανάπτυξη της εφαρμογής έγινε με χρήση του Microsoft Visual Studio .NET 2003TM και πιο συγκεκριμένα η γλώσσα προγραμματισμού στην οποία αναπτύχθηκε η εφαρμογή ήταν η C++ τόσο στην παραδοσιακή unmanaged code μορφή της , όσο και με τη χρήση των managed extensions for .NET[®] framework.

5.2.2 Απαιτήσεις εφαρμογής

Για την σωστή εκτέλεση της εφαρμογής απαιτείται να είναι εγκατεστημένα (και σωστά παραμετροποιημένα) στο μηχάνημα του χρήστη τα διάφορα components που αποτελούν την πλατφόρμα. Τα κυριότερα εξ αυτών είναι ο SQL server 2000 ή νεότερος, ο IAS Windows 2003 Server Enterprise Edition και το .NET[®] Framework v.1.1 της Microsoft. Το .NET[®] Framework διανέμεται προς ελεύθερη χρήση από την ιστοσελίδα της Microsoft και η εγκατάστασή του αποτελεί απαραίτητη προϋπόθεση για τη δυνατότητα εκτέλεσης (σχεδόν) οποιασδήποτε εφαρμογής έχει αναπτυχθεί με κάποια από τις γλώσσες προγραμματισμού που συμπεριλαμβάνονται στο περιβάλλον του Visual Studio .NET 2003. Σημειώνεται ότι στο cd που συνοδεύει την παρούσα διπλωματική εργασία περιέχεται το .NET[®] Framework v.1.1 το οποίο όμως πρέπει να εγκατασταθεί ξεχωριστά από τον χρήστη.

5.3 Περιγραφή υλοποίησης εφαρμογής

Η εφαρμογή Account++ στηρίχθηκε στις αρχές του αντικειμενοστρεφούς προγραμματισμού με τη χρήση τόσο C++ native code όσο και εντολών περιβάλλοντος .NET framework .



Σχ. 5.VII Κλάσεις της εφαρμογής Account++

Η εφαρμογή Account++ αποτελείται από έναν αριθμό κλάσεων οι οποίες συνεργάζονται και αλληλεπιδρούν για να επιτελέσουν τις διάφορες διεργασίες. Παρακάτω παρατίθεται μια σύντομη περιγραφή των κλάσεων:

- **Form1**

Αν και οι κλάσεις παρατίθενται αλφαβητικά, κατ' εξαίρεση πρώτη αναφέρεται η Form1. Η Form1 αποτελεί την πρώτη οθόνη που παρουσιάζεται στον χρήστη και ως εκ τούτου την «βιτρίνα» της εφαρμογής μας. Όπως και όλες οι υπόλοιπες κλάσεις η Form1 κληρονομεί τα χαρακτηριστικά της System::Windows::Forms::Form κλάσης αφού όλες οι κλάσεις υλοποιούν κάποιο παράθυρο με τη χρήση του πακέτου Windows Forms. Η Form1 διαθέτει με τη μορφή μενού όλες τις επιλογές διαχείρισης της εφαρμογής.

- **AboutBox**

Η κλάση AboutBox είναι η πιο απλή κλάση της εφαρμογής. Το μόνο που επιτελεί είναι να παρουσιάζει με τη μορφή κειμένου πληροφορίες για την εφαρμογή, το δημιουργό κ.α.

- **AUser**

Η κλάση AUser αποτελεί τον πυρήνα της προσθήκης χρηστών τόσο στο Active Directory όσο και στον SQL server. Παρέχει τις μεθόδους για την προσθήκη χρηστών με όρισμα το μοναδικό userid του χρήστη.

- **CUser**

Μέσω της κλάσης CUser μπορούμε να τροποποιήσουμε τα στοιχεία ενός χρήστη. Ανάμεσα στις μεθόδους που περιέχει ξεχωρίζουν οι δύο μέθοδοι που ανανεώνουν τα στοιχεία στον SQL Server και στο Active Directory αντίστοιχα.

- **DataSet1...DataSet16**

Το μεγάλο πλήθος των DataSet οφείλεται στην ανάγκη για offline διαχείριση δεδομένων που αντλούμε από τον SQL Server. Δημιουργούνται αυτόματα από το Visual Studio και περιέχουν πληροφορίες που αφορούν το σχήμα των δεδομένων που έχουν αποθηκευμένα.

- **ExportData**

Η κλάση ExportData έχει ρόλο τόσο εισόδου όσο και εξόδου δεδομένων. Είσοδος δεδομένων γίνεται όταν επιλέγουμε κάποιο sql query για προεπισκόπηση. Τα δεδομένα περνάνε προς την έξοδο όταν αποφασίσουμε να αποθηκεύσουμε τα δεδομένα που επιλέξαμε σε αρχείο τύπου XML.

- **PerTimePeriod**

Η κλάση PerTimePeriod είναι η πρώτη από τις δύο κλάσεις που ασχολούνται με την προβολή στο χρήστη δεδομένων ανάλογα με τη χρονική περίοδο που αυτός θα επιλέξει. Σε αυτήν την κλάση ο χρήστης επιλέγει ημέρα και ώρα έναρξης και λήξης της παρακολούθησης του δικτύου ώστε να επιλεγούν μόνο εκείνες οι εγγραφές που ανήκουν σε αυτό το χρονικό παράθυρο. Αυτήν την έξοδο την προωθεί στην κλάση PerTimePeriodResults1 για περαιτέρω επεξεργασία.

- **PerTimePeriodResults1**

Στην δεύτερη από τις δύο κλάσεις που προαναφέραμε, η κλάση πέρνει σαν είσοδο από την PerTimePeriod την ώρα και ημέρα που επέλεξε ο χρήστης. Σαν έξοδο στον χρήστη επιστρέφει ένα datagrid με τα αποτελέσματα της αναζήτησης.

- **PerUser1**

Στην κλάση PerUser1 μπορούμε να φιλτράρουμε τα δεδομένα μας με βάση τον χρήστη στον οποίο αναφέρονται. Έτσι, έχουμε μια εποπτική εικόνα για τις κινήσεις κάποιου χρήστη ώστε να μπορούμε να εξαγάγουμε αναλυτικά στατιστικά στοιχεία.

- **Print**

Η Print κλάση επιτρέπει στον χρήστη της εφαρμογής να εκτυπώσει στον εκτυπωτή ή σε τοπικό αρχείο τα δεδομένα που έχει κάνει εξαγωγή με την επιλογή της κλάσης export data.

- **RUser**

Η κλάση RUser πέρνει από τον χρήστη το όρισμα του χρήστη που θέλει να διαγράψει με χρήση του μοναδικού userid του και τον διαγράφει από το Active Directory της πλατφόρμας. Με αυτήν τη μέθοδο ο χρήστης στερείται την πρόσβαση αλλά τα στοιχεία του παραμένουν αποθηκευμένα στη βάση για πιθανή μελλοντική στατιστική επεξεργασία.

5.4 Περιγραφή των σημαντικότερων λειτουργιών της εφαρμογής (περιγραφή σε επίπεδο methods που δημιουργήσα)

Σε αυτήν την ενότητα θα περιγράψουμε τις κυριότερες λειτουργίες της εφαρμογής σε επίπεδο μεθόδων. Συνοπτικά λοιπόν έχουμε:

- `Int AUser::ADWrite(String __gc *a)`

Με όρισμα εισόδου το μοναδικό userid του χρήστη που θέλουμε να εγγράψουμε στο Active Directory αυτή η μέθοδος διεκπεραιώνει την εγγραφή του χρήστη στο Active Directory και επιστρέφει το ανάλογο MessageBox στον χρήστη ανάλογα με το αν η εγγραφή πέτυχε ή απέτυχε.

- `Int AUser::SQLWrite(void)`

Χωρίς να χρειάζεται κάποιο όρισμα εισόδου, αυτή η μέθοδος κάνει την κατάλληλη εγγραφή στον SQL Server και ανήκει στην κλάση AUser. Χρησιμοποιώντας τις δηλώσεις των textboxes, ανοίγει πρώτα μια sqlconnection με τη βάση δεδομένων και περνάει τα δεδομένα με τη χρήση ενός sql query.

- `Int CUser::SQLUpdate(void)`

Όπως και στην AUser class, έτσι και εδώ για τις αλλαγές στη βάση δεδομένων δεν χρειάζεται να δώσουμε όρισμα στη μέθοδο. Όλες οι παράμετροι αντλούνται από τα ανάλογα textboxes του παραθύρου και ενημερώνουν τη βάση με μια εντολή τύπου SQL update.

- `Bool PerTimePeriod::CheckForms(int helper)`

Με αυτήν τη μέθοδο εκτελείται ένας στοιχειώδης έλεγχος του αν έχουν επιλεγεί και τα τέσσερα στοιχεία που απαιτούνται για να οριστεί πλήρως ένα χρονικό παράθυρο (ώρα/ημέρα έναρξης, ώρα/ημέρα λήξης παρακολούθησης). Το όρισμα που επιστρέφει είναι τύπου Boolean και πρέπει να γίνει true για να συνεχιστεί η εκτέλεση της εφαρμογής.

- `Int PerTimePeriodResults1::SelectPtptr(String *a, String *b, String *c, String *d)`

Αυτή η μέθοδος της κλάσης PerTimePeriodResults1 διενεργεί τις κύριες λειτουργίες της κλάσης. Αφού πάρει σαν όρισμα τα τέσσερα στοιχεία που ορίζουν πλήρως το χρονικό παράθυρο στην PerTimePeriod κλάση, τα χρησιμοποιεί για να δημιουργήσει το ανάλογο DataSet δεδομένων με τα δεδομένα που μας ενδιαφέρουν.

- `Private: void Print::pd_PrintPage(System::Object * sender, System::Drawing::Printing::PrintPageEventArgs * ev)`

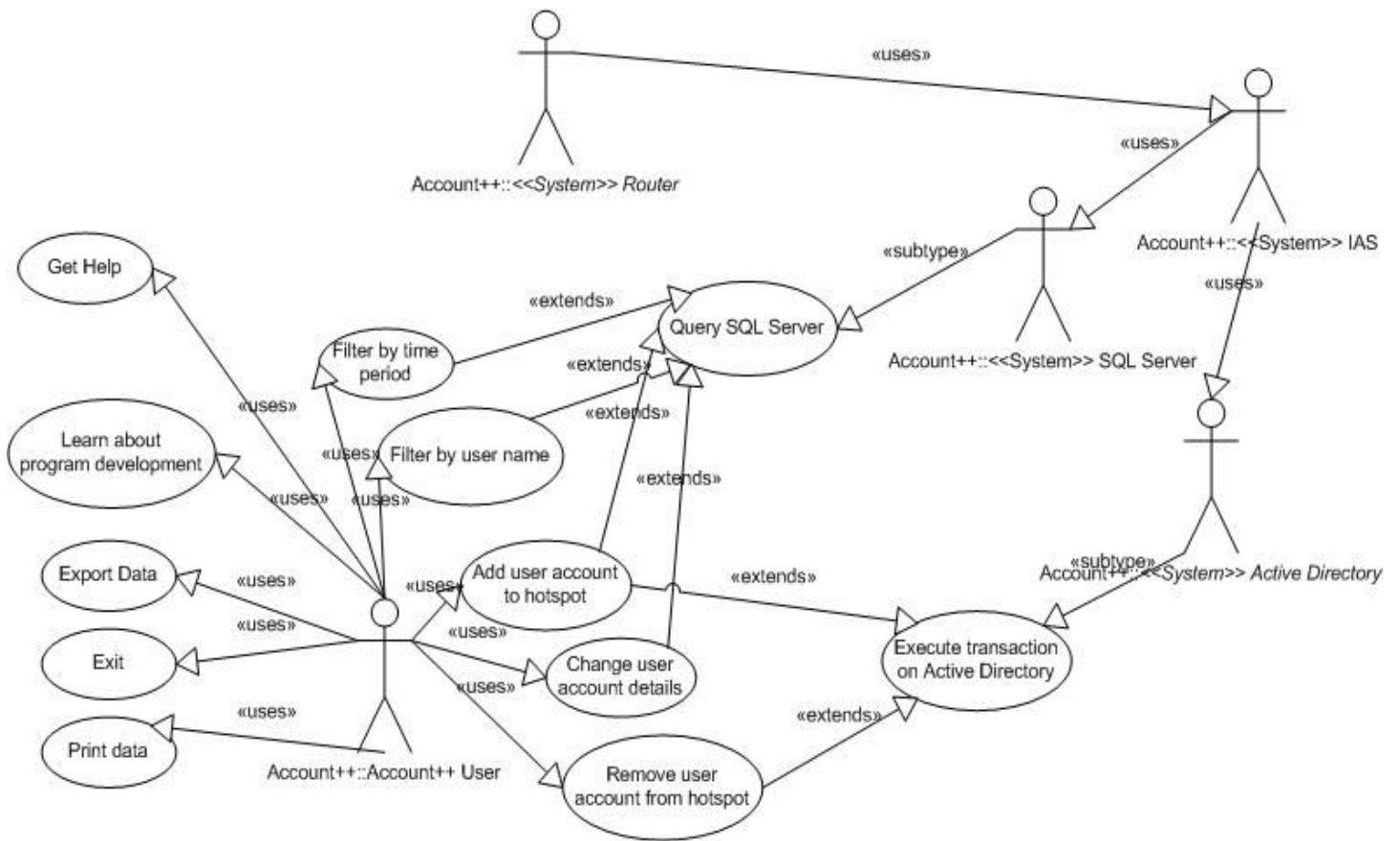
Η κλάση pd_PrintPage της κλάσης Print εκτυπώνει μία ακριβώς σελίδα στον εκτυπωτή ή αρχείο κειμένου που έχουμε επιλέξει σαν στόχο εκτύπωσης. Η διαχείριση των σελίδων προς εκτύπωση γίνεται από την υπόλοιπη κλάση, αυτή η μέθοδος απλά εκτυπώνει μία μόνο σελίδα.

- `Public: int RUser::ADDelete(String *a)`

Η μέθοδος ADDelete της κλάσης RUser διαγράφει έναν χρήστη από το Active Directory. Το όρισμα String που δέχεται αποτελεί το μοναδικό userid του χρήστη. Ανάλογα με την έκβαση της διαγραφής επιστρέφεται το ανάλογο μήνυμα στον χρήστη και συνεχίζεται ή μη η εκτέλεση του προγράμματος.

5.5 Περιγραφή αρχιτεκτονικής λογισμικού με χρήση UML Diagrams

Με τη χρήση διαγραμμάτων UML μπορούμε να έχουμε μια εποπτική εικόνα των διαφόρων ενεργειών που επιτελεί το πρόγραμμα. Συγκεκριμένα, το διάγραμμα Use case χρησιμεύει στο να καταλάβουμε πως οι διάφορες ενέργειες που επιτελεί ο χρήστης προκαλούν την αλληλεπίδραση άλλων νοητών οντοτήτων ώστε να φτάσουμε στο επιθυμητό αποτέλεσμα. Το διάγραμμα Class αποτυπώνει τις κλάσεις του προγράμματος και τις συσχετίσεις και εξαρτήσεις που έχουν μεταξύ τους. Και τα δύο διαγράμματα χρησιμεύουν σαν προσχέδιο του τελικού προγράμματος και δεν αντιπροσωπεύουν μια πιστή αναπαράσταση της τελικής μορφής αφού για λόγους ευχέρειας των μηχανικών λογισμικού δεν εμφανίζονται κλάσεις και μέθοδοι που επιτελούν βοηθητικές λειτουργίες. Στα δύο επόμενα σχήματα παρουσιάζονται τα διαγράμματα:



Σχ. 5.VIII Διάγραμμα Use case



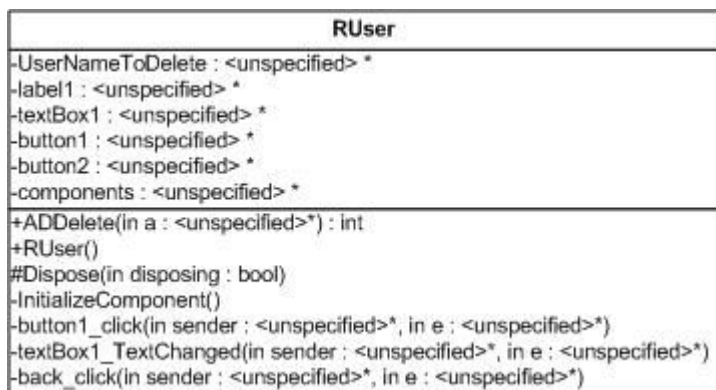
AUser
<pre> -a : <unspecified> * -b : <unspecified> * -e : <unspecified> * -f : <unspecified> * -sqlDataAdapter1 : <unspecified> * -sqlSelectCommand1 : <unspecified> * -sqlInsertCommand2 : <unspecified> * -sqlInsertCommand3 : <unspecified> * -sqlConnection1 : <unspecified> * -sqlCommand1 : <unspecified> * -sqlCommand2 : <unspecified> * -dataSet131 : <unspecified> * -button2 : <unspecified> * -label2 : <unspecified> * -label1 : <unspecified> * -textBox1 : <unspecified> * -label3 : <unspecified> * -label4 : <unspecified> * -label5 : <unspecified> * -label6 : <unspecified> * -label7 : <unspecified> * -label8 : <unspecified> * -textBox2 : <unspecified> * -textBox3 : <unspecified> * -textBox4 : <unspecified> * -textBox5 : <unspecified> * -textBox6 : <unspecified> * -radioButton1 : <unspecified> * -radioButton2 : <unspecified> * -textBox7 : <unspecified> * -label9 : <unspecified> * -button1 : <unspecified> * -label10 : <unspecified> * -components : <unspecified> * </pre>
<pre> +SQLWrite() : int +ADWrite(in a : <unspecified>*) : int +AUser() #Dispose(in disposing : bool) -InitializeComponent() -label1_Click(in sender : <unspecified>*, in e : <unspecified>*) -button1_click(in sender : <unspecified>*, in e : <unspecified>*) -textBox1_TextChanged(in sender : <unspecified>*, in e : <unspecified>*) -sqlConnection1_InfoMessage(in sender : <unspecified>*, in e : <unspecified>*) -sqlDataAdapter1_RowUpdated(in sender : <unspecified>*, in e : <unspecified>*) -back_click(in sender : <unspecified>*, in e : <unspecified>*) </pre>

CUser					
<pre> -sqlUpdateCommand1 : <unspecified> * -button2 : <unspecified> * -sqlUpdateCommand2 : <unspecified> * -label1 : <unspecified> * -label2 : <unspecified> * -label3 : <unspecified> * -label4 : <unspecified> * -label5 : <unspecified> * -label6 : <unspecified> * -label7 : <unspecified> * -label8 : <unspecified> * -radioButton1 : <unspecified> * -radioButton2 : <unspecified> * -textBox1 : <unspecified> * -textBox2 : <unspecified> * -textBox3 : <unspecified> * -textBox4 : <unspecified> * -textBox5 : <unspecified> * -textBox6 : <unspecified> * -textBox7 : <unspecified> * -label9 : <unspecified> * -button1 : <unspecified> * -label10 : <unspecified> * -label11 : <unspecified> * -textBox8 : <unspecified> * -sqlDataAdapter1 : <unspecified> * -sqlSelectCommand1 : <unspecified> * -sqlConnection1 : <unspecified> * -components : <unspecified> * +SQLUpdate() : int +CUser() #Dispose(in disposing : bool) -InitializeComponent() -label3_Click(in sender : <unspecified>*, in e : <unspecified>*) -label5_Click(in sender : <unspecified>*, in e : <unspecified>*) -label9_Click(in sender : <unspecified>*, in e : <unspecified>*) -button1_click(in sender : <unspecified>*, in e : <unspecified>*) -back_click(in sender : <unspecified>*, in e : <unspecified>*) </pre>	<table border="1"> <thead> <tr> <th colspan="2" style="text-align: center;">ExportData</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"> <pre> -label11 : bool -label22 : bool -button3 : <unspecified> * -saveFileDialog1 : <unspecified> * -textBox1 : <unspecified> * -label1 : <unspecified> * -label2 : <unspecified> * -radioButton2 : <unspecified> * -button1 : <unspecified> * -sqlConnection1 : <unspecified> * -sqlDataAdapter1 : <unspecified> * -dataGrid1 : <unspecified> * -sqlSelectCommand1 : <unspecified> * -dataSet11 : <unspecified> * -button2 : <unspecified> * -components : <unspecified> * +ExportData() #Dispose(in disposing : bool) -InitializeComponent() -label1_Click(in sender : <unspecified>*, in e : <unspecified>*) -label2_Click(in sender : <unspecified>*, in e : <unspecified>*) -textbox_update(in sender : <unspecified>*, in e : <unspecified>*) -button_click(in sender : <unspecified>*, in e : <unspecified>*) -buttonGO_click(in sender : <unspecified>*, in e : <unspecified>*) -back_click(in sender : <unspecified>*, in e : <unspecified>*) </pre> </td> <td style="vertical-align: top;"> <pre> </pre> </td> </tr> </tbody> </table>	ExportData		<pre> -label11 : bool -label22 : bool -button3 : <unspecified> * -saveFileDialog1 : <unspecified> * -textBox1 : <unspecified> * -label1 : <unspecified> * -label2 : <unspecified> * -radioButton2 : <unspecified> * -button1 : <unspecified> * -sqlConnection1 : <unspecified> * -sqlDataAdapter1 : <unspecified> * -dataGrid1 : <unspecified> * -sqlSelectCommand1 : <unspecified> * -dataSet11 : <unspecified> * -button2 : <unspecified> * -components : <unspecified> * +ExportData() #Dispose(in disposing : bool) -InitializeComponent() -label1_Click(in sender : <unspecified>*, in e : <unspecified>*) -label2_Click(in sender : <unspecified>*, in e : <unspecified>*) -textbox_update(in sender : <unspecified>*, in e : <unspecified>*) -button_click(in sender : <unspecified>*, in e : <unspecified>*) -buttonGO_click(in sender : <unspecified>*, in e : <unspecified>*) -back_click(in sender : <unspecified>*, in e : <unspecified>*) </pre>	<pre> </pre>
ExportData					
<pre> -label11 : bool -label22 : bool -button3 : <unspecified> * -saveFileDialog1 : <unspecified> * -textBox1 : <unspecified> * -label1 : <unspecified> * -label2 : <unspecified> * -radioButton2 : <unspecified> * -button1 : <unspecified> * -sqlConnection1 : <unspecified> * -sqlDataAdapter1 : <unspecified> * -dataGrid1 : <unspecified> * -sqlSelectCommand1 : <unspecified> * -dataSet11 : <unspecified> * -button2 : <unspecified> * -components : <unspecified> * +ExportData() #Dispose(in disposing : bool) -InitializeComponent() -label1_Click(in sender : <unspecified>*, in e : <unspecified>*) -label2_Click(in sender : <unspecified>*, in e : <unspecified>*) -textbox_update(in sender : <unspecified>*, in e : <unspecified>*) -button_click(in sender : <unspecified>*, in e : <unspecified>*) -buttonGO_click(in sender : <unspecified>*, in e : <unspecified>*) -back_click(in sender : <unspecified>*, in e : <unspecified>*) </pre>	<pre> </pre>				

Form1
<pre> -mainMenu1 : <unspecified> * -menultem1 : <unspecified> * -menultem2 : <unspecified> * -menultem3 : <unspecified> * -menultem4 : <unspecified> * -menultem6 : <unspecified> * -menultem7 : <unspecified> * -menultem8 : <unspecified> * -menultem9 : <unspecified> * -menultem10 : <unspecified> * -menultem11 : <unspecified> * -menultem12 : <unspecified> * -printDialog1 : <unspecified> * -dataGridTextBoxColumn1 : <unspecified> * -sqlDataAdapter1 : <unspecified> * -sqlSelectCommand1 : <unspecified> * -sqlInsertCommand1 : <unspecified> * -sqlConnection1 : <unspecified> * -dataSet11 : <unspecified> * -dataSet81 : <unspecified> * -pictureBox1 : <unspecified> * -menultem13 : <unspecified> * -menultem14 : <unspecified> * -menultem15 : <unspecified> * -components : <unspecified> * +Form1() #Dispose(in disposing : bool) -InitializeComponent() -menultem3_Click(in sender : <unspecified>*, in e : <unspecified>*) -menultem1_Click(in sender : <unspecified>*, in e : <unspecified>*) -help_ab(in sender : <unspecified>*, in e : <unspecified>*) -listView1_SelectedIndexChanged(in sender : <unspecified>*, in e : <unspecified>*) -label1_Click(in sender : <unspecified>*, in e : <unspecified>*) -menultem5_Click(in sender : <unspecified>*, in e : <unspecified>*) -menultem7_Click(in sender : <unspecified>*, in e : <unspecified>*) -menultem6_Click(in sender : <unspecified>*, in e : <unspecified>*) -menultem2_Click(in sender : <unspecified>*, in e : <unspecified>*) -menultem11_Click(in sender : <unspecified>*, in e : <unspecified>*) -dataGrid1_Navigate(in sender : <unspecified>*, in ne : <unspecified>*) -menultem9_Click(in sender : <unspecified>*, in e : <unspecified>*) -menultem15_click(in sender : <unspecified>*, in e : <unspecified>*) -menultem14_click(in sender : <unspecified>*, in e : <unspecified>*) -menultem13_click(in sender : <unspecified>*, in e : <unspecified>*) </pre>

PerTimePeriod
<pre> -ga : bool -gb : bool -gc : bool -gd : bool -ge : bool -button2 : <unspecified> * -dateTimePicker1 : <unspecified> * -dateTimePicker3 : <unspecified> * -label1 : <unspecified> * -label2 : <unspecified> * -listBox1 : <unspecified> * -listBox2 : <unspecified> * -label3 : <unspecified> * -label4 : <unspecified> * -button1 : <unspecified> * -components : <unspecified> * -dtp1 : <unspecified> * -dtp3 : <unspecified> * -dtp11 : <unspecified> * -dtp33 : <unspecified> * +PerTimePeriod() #Dispose(in disposing : bool) -InitializeComponent() -timer1_Tick(in sender : <unspecified>*, in e : <unspecified>*) -timer2_Tick(in sender : <unspecified>*, in e : <unspecified>*) -timer3_Tick(in sender : <unspecified>*, in e : <unspecified>*) -timer4_Tick(in sender : <unspecified>*, in e : <unspecified>*) -timer5_Tick(in sender : <unspecified>*, in e : <unspecified>*) -CheckForms(in helper : int) : bool -back_click(in sender : <unspecified>*, in e : <unspecified>*) </pre>
PerTimePeriodResults1
<pre> -sqlSelectCommand1 : <unspecified> * -sqlDataAdapter1 : <unspecified> * -sqlConnection1 : <unspecified> * -sqlCommand1 : <unspecified> * -dataGrid1 : <unspecified> * -label1 : <unspecified> * -sqlInsertCommand1 : <unspecified> * -dataset141 : <unspecified> * -components : <unspecified> * +PerTimePeriodResults1(in a : <unspecified>*, in b : <unspecified>*, in c : <unspecified>*, in d : <unspecified>*) +SelectPtr(in a : <unspecified>*, in b : <unspecified>*, in c : <unspecified>*, in d : <unspecified>*) : int #Dispose(in disposing : bool) -InitializeComponent() </pre>

PerUser1
<pre> -performanceCounter1 : <unspecified> * -dataSet161 : <unspecified> * -sqlDataAdapter2 : <unspecified> * -sqlSelectCommand2 : <unspecified> * -button3 : <unspecified> * +listbox2value : <unspecified> * -listbox1 : <unspecified> * -label1 : <unspecified> * +button1 : <unspecified> * -button2 : <unspecified> * -dataGrid1 : <unspecified> * -listBox2 : <unspecified> * -sqlDataAdapter1 : <unspecified> * -sqlSelectCommand1 : <unspecified> * -sqlConnection1 : <unspecified> * -dataSet151 : <unspecified> * -dataSet152 : <unspecified> * -components : <unspecified> * +PerUser1() #Dispose(in disposing : bool) -InitializeComponent() -listBox1_SelectedIndexChanged(in sender : <unspecified>*, in e : <unspecified>*) -label1_Click(in sender : <unspecified>*, in e : <unspecified>*) -PerUser1_Load(in sender : <unspecified>*, in e : <unspecified>*) -button1_Click(in sender : <unspecified>*, in e : <unspecified>*) -sqlConnection1_InfoMessage(in sender : <unspecified>*, in e : <unspecified>*) -sqlDataAdapter1_RowUpdated(in sender : <unspecified>*, in e : <unspecified>*) -sqlConnection1_InfoMessage_1(in sender : <unspecified>*, in e : <unspecified>*) -sqlConnection1_InfoMessage_2(in sender : <unspecified>*, in e : <unspecified>*) -sqlDataAdapter2_RowUpdated(in sender : <unspecified>*, in e : <unspecified>*) -sqlDataAdapter1_RowUpdated_1(in sender : <unspecified>*, in e : <unspecified>*) -dataGrid1_Navigate(in sender : <unspecified>*, in ne : <unspecified>*) -sqlDataAdapter1_RowUpdated_2(in sender : <unspecified>*, in e : <unspecified>*) -textBox1_TextChanged(in sender : <unspecified>*, in e : <unspecified>*) -dataGrid1_Navigate_1(in sender : <unspecified>*, in ne : <unspecified>*) -listBox2_SelectedIndexChanged(in sender : <unspecified>*, in e : <unspecified>*) -sqlConnection1_InfoMessage_3(in sender : <unspecified>*, in e : <unspecified>*) -dataView1_ListChanged(in sender : <unspecified>*, in e : <unspecified>*) -button2_click(in sender : <unspecified>*, in e : <unspecified>*) -back_click(in sender : <unspecified>*, in e : <unspecified>*) </pre>
Print
<pre> -xmlfile : bool -printoverview : bool -printDialog1 : <unspecified> * -printDocument1 : <unspecified> * -printPreviewDialog2 : <unspecified> * -openFileDialog1 : <unspecified> * -button2 : <unspecified> * -printPreviewDialog1 : <unspecified> * -streamToPrint : <unspecified> * -label1 : <unspecified> * -radioButton1 : <unspecified> * -printFont : <unspecified> * -button1 : <unspecified> * -components : <unspecified> * +Print() #Dispose(in disposing : bool) -InitializeComponent() -PrintXmlFile_click(in sender : <unspecified>*, in e : <unspecified>*) -Print_click(in sender : <unspecified>*, in e : <unspecified>*) -pd_PrintPage(in sender : <unspecified>*, in ev : <unspecified>*) -close_window(in sender : <unspecified>*, in e : <unspecified>*) </pre>



Σχ. 5.IX UML Διάγραμμα κλάσεων και αναλυτικό διάγραμμα κάθε κλάσης

5.6 Περιγραφή γραφικής διεπαφής χρήστη

Σε αυτήν την ενότητα θα επεξηγήσουμε τη λειτουργία της εφαρμογής Account++ . Η εφαρμογή Account++ αποτελεί τον συνδεδετικό κρίκο στη λειτουργία της πλατφόρμας AAA. Αλληλεπιδρώντας με όλες τις τεχνολογικές λύσεις οι οποίες αποτελούν την πλατφόρμα υλοποίησης, (η εφαρμογή) αποτελεί ένα πολύ χρήσιμο εργαλείο για την απρόσκοπτη συνεχή λειτουργία της.

Η εφαρμογή ξεκινώντας , παρουσιάζει στον τελικό χρήστη το κεντρικό παράθυρο μέσα από το οποίο ο χρήστης μπορεί να επιλέξει όποια από τις διαθέσιμες εφαρμογές επιθυμεί. Οι επιλογές των κυρίων menu είναι File, View, Tools και Help. Η επιλογή File(Αρχείο) διαθέτει τις υπο-επιλογές Print και eXit. Η πρώτη υπο-επιλογή μας οδηγεί στο παράθυρο εκτύπωσης. Η δεύτερη υπο-επιλογή τερματίζει το πρόγραμμα. Η επιλογή View με τις υποεπιλογές Per time period και pEr user μας επιτρέπει να δούμε στατιστικά των χρηστών είτε ανά χρονική περίοδο που εμείς ορίζουμε, είτε ανά χρήστη.

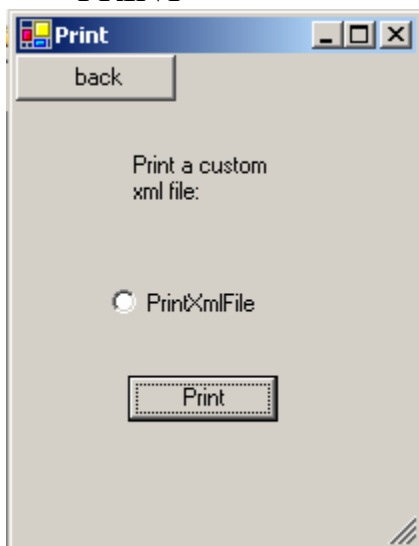
Στην επιλογή του menu Tools , έχουμε τέσσερις επιλογές. Αυτές είναι οι Export data, Add user, Change user και Remove user. Η επιλογή Export data ανοίγει ένα νέο παράθυρο μέσω του οποίου μπορούμε να εξαγάγουμε τα δεδομένα που επεξεργάζεται η εφαρμογή σε αρχεία τύπου xml. Το xml είναι ένα δημοφιλές πρότυπο αποθήκευσης δεδομένων με κύρια χαρακτηριστικά τη διασυνδεσιμότητα και το ότι είναι ανοιχτό πρότυπο. Οι πιθανές χρήσεις της εφαρμογής υπέδειξαν τη χρησιμοποίηση αυτού του προτύπου. Με την δεύτερη επιλογή, την Add user, μέσω της σύνδεσης με το Active Directory μπορούμε να εγγράψουμε κάποιο χρήστη στο Active Directory της πλατφόρμας και κατεπέκταση να τον πιστοποιήσουμε για χρήση της πλατφόρμας. Η επιλογή Change user, όπως και η προηγούμενη αλληλεπιδρά με το Active Directory για την τροποποίηση των δεδομένων κάποιου χρήστη χρησιμοποιώντας το μοναδικό userid του. Εκτός από το όνομα χρήστη μπορούμε να τροποποιήσουμε πολλά άλλα στοιχεία που περιέχονται στην καρτέλα του χρήστη όπως διεύθυνση, τηλέφωνο, email κλπ. Τέλος, με την επιλογή Remove user διαγράφουμε κάποιον χρήστη από τη βάση δεδομένων και το Active Directory με βάση το μοναδικό userid του. Αφαιρώντας την εγγραφή του χρήστη από το Active Directory στερούμε το δικαίωμα σε αυτόν τον χρήστη να αποκτήσει πρόσβαση στην πλατφόρμα AAA.



Σχ. 5.X Κεντρικό μενού εφαρμογής Account++

Στην υπο-επιλογή Print του menu File έχουμε την επιλογή να εκτυπώσουμε κάποιο xml αρχείο που έχουμε αποθηκεύσει προηγουμένως. Έτσι επιλέγοντας το radiobutton PrintXmlFile και πατώντας κατόπιν το κουμπί Print παρουσιάζεται το γνωστό παράθυρο εκτύπωσης από όπου μπορούμε να επιλέξουμε τις επιλογές εκτύπωσης.

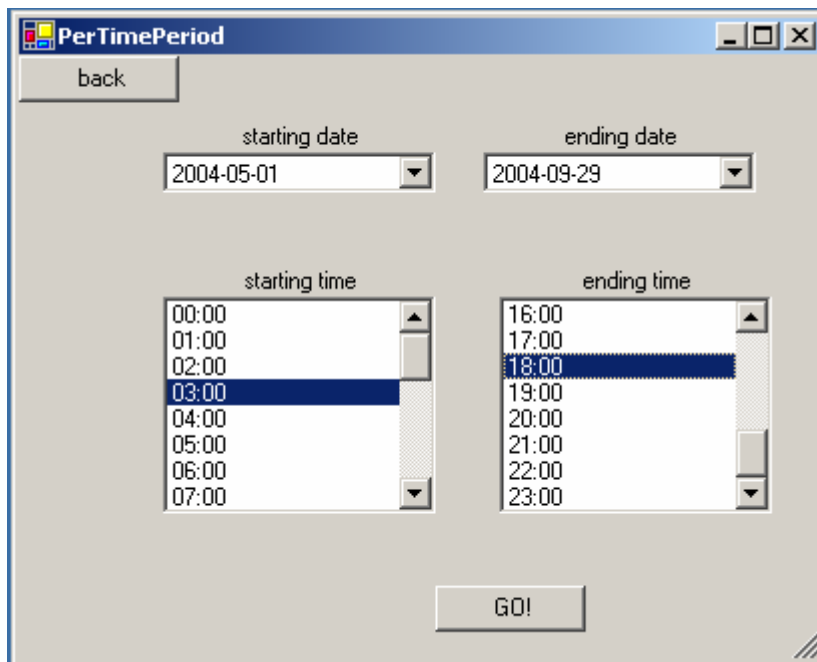
PRINT



Σχ. 5.XI Επιλογή Print της εφαρμογής

Στην υπο-επιλογή Per time period μπορούμε να δούμε τις ενέργειες χρηστών που κατέγραψε ο authentication-proxy σύμφωνα με την ώρα και την ημέρα εκτέλεσης. Η ημερομηνίες εισάγονται στην μορφή YYYY-MM-DD ενώ οι ώρες σύμφωνα με εικοσιτετράωρη αρίθμηση.

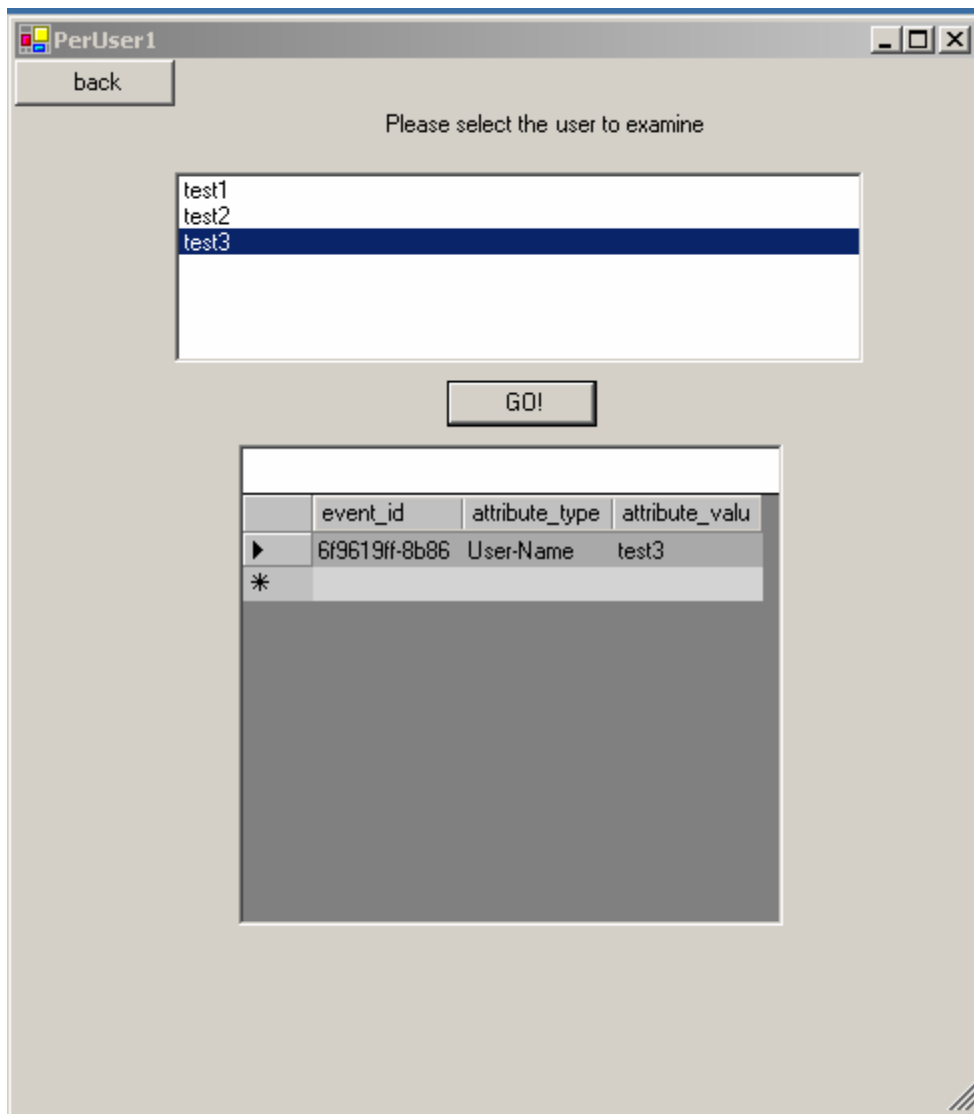
PER TIME PERIOD



Σχ. 5.XII Επιλογή PerTimeperiod της εφαρμογής

Στην επιλογή pEr user βλέπουμε τα καταγεγραμμένα στοιχεία με κριτήριο επιλογής τον χρήστη στον οποίο οφείλονται. Με ένα απλό κλικ στο όνομα του χρήστη βλέπουμε ακριβώς από κάτω τις διάφορες ενέργειές του σε ζεύγη attribute_type/attribute_value.

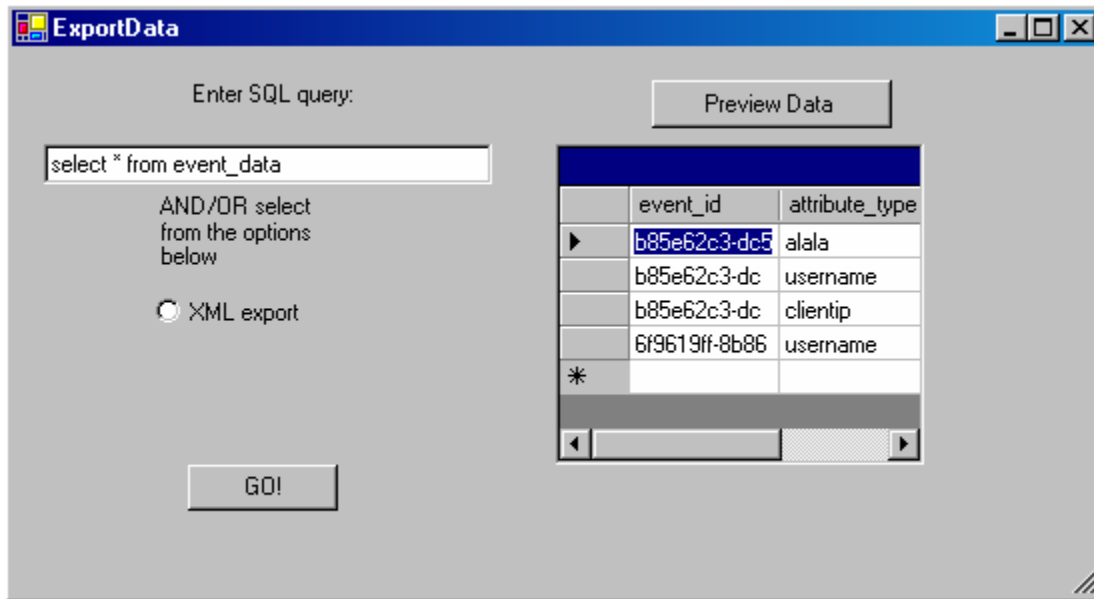
PER USER



Σχ. 5.XIII Επιλογή Peruser της εφαρμογής

Με την υπο-επιλογή Export data του μενού Tools μπορούμε να εξάγουμε δεδομένα του προγράμματος με τη μορφή αρχείων τύπου xml. Εισάγωντας ένα απλό SQL query με το κουμπί Preview data μπορούμε να δούμε το αποτέλεσμα και αν μας ικανοποιεί να το αποθηκεύσουμε με το όνομα της επιλογής μας.

EXPORT DATA



Σχ. 5.XIV Επιλογή ExportData της εφαρμογής

Η πρώτη υποεπιλογή που αφορά την αλληλεπίδραση της εφαρμογής με το Active Directory αλλά και με τον SQL server είναι η Add user. Εδώ μπορούμε να εισάγουμε υποχρεωτικά αλλά και προαιρετικά στοιχεία του χρήστη που θέλουμε να πιστοποιήσουμε στην πλατφόρμα AAA. Η εγγραφή του χρήστη γίνεται τόσο στο Active Directory όσο και στον SQL server. Με την εγγραφή στον SQL server έχουμε ανά πάσα στιγμή διαθέσιμα τα στοιχεία πιστοποίησης του χρήστη, ενώ με την εγγραφή στο Active Directory τον πιστοποιούμε σε πρόσβαση στην πλατφόρμα AAA.

ADD USER

Add a new user

back

Enter the appropriate details for the user you want to add to the system

Name ** Type of Connection **

Surname ** Prepaid Subscriber

Fixed Telephone number ** How many minutes?

Mobile number

Email **

Address

fields marked with ** are compulsory

Σχ. 5.XV Επιλογή Add User της εφαρμογής

Η υπο-επιλογή Change user τροποποιεί τα στοιχεία του χρήστη στον SQL server. Οι ανάλογες εγγραφές στην βάση δεδομένων μας τροποποιούνται σύμφωνα με το event_id του

χρήστη που θέλουμε να τροποποιήσουμε. Η εγγραφή του στο AD παραμένει αναλλοίωτη αφού το AD χρησιμεύει στην πιστοποίηση ή μη κάποιου χρήστη μέσω του IAS.

CHANGE USER

Change user credentials

back

Change user credentials

Please enter the unique identifier name of the user you want to modify. In case this is incorrect no modifications will be made to the database.

unique userid: textBox8

Name *** textBox1

Surname *** textBox2

Fixed telephone number *** textBox3

Mobile Number textBox4

Email *** textBox5

Address textBox6

Type of connection ***

Prepaid Subscriber

How many minutes? textBox7

Apply Changes!

Σχ. 5.XVI Επιλογή Change User της εφαρμογής

Στην επιλογή Remove user χρησιμοποιώντας το μοναδικό userid (username) ενός χρήστη μπορούμε να τον αφαιρέσουμε από τη λίστα πιστοποιημένων χρηστών του Active Directory (ομάδα HOTSPOTusers). Οι εγγραφές του χρήστη στον SQL server παραμένουν ανέπαφες για λόγους στατιστικής επεξεργασίας.

REMOVE USER

Remove a user from the system

back

Enter username to delete

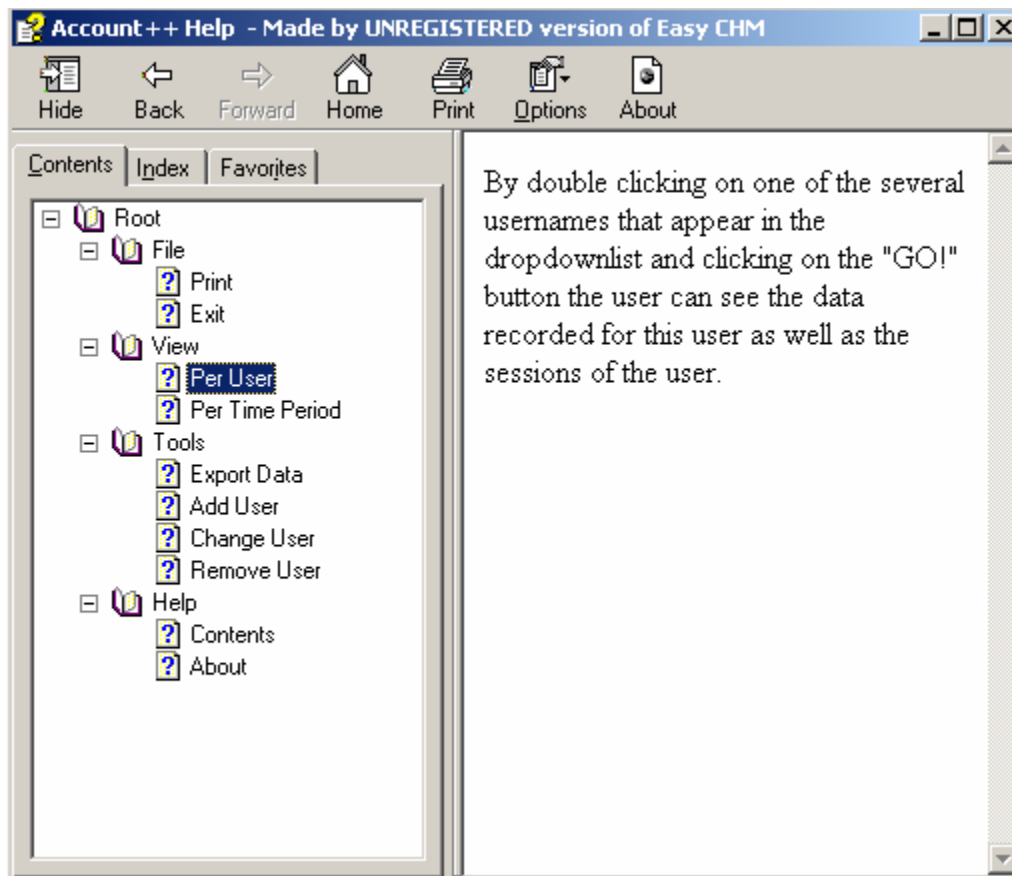
alexojamas

Remove him!

Σχ. 5.XVII Επιλογή Remove User της εφαρμογής

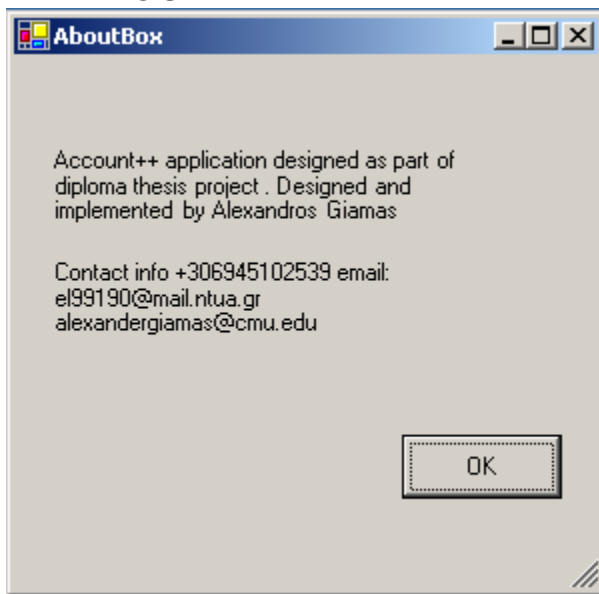
CONTENTS

Στην επιλογή contents βλέπουμε τη βοήθεια της εφαρμογής. Αυτή είναι ιεραρχικά δομημένη σύμφωνα με τα μενού και τις επιλογές της εφαρμογής.



Τέλος, σε αυτό το απλό παράθυρο παρουσιάζονται πληροφορίες για την εφαρμογή και την υλοποίησή της.

ABOUT



Σχ. 5.XVIII Επιλογή About της εφαρμογής

Appendix A

Source code

Παρακάτω παρατίθεται μέρος του κώδικα της εφαρμογής Account++. Σκόπιμα και χάριν ευχρηστίας στην ανάγνωση έχουν παραλειφθεί δευτερεύουσες κλάσεις που χρησιμεύουν στη διαχείριση των δεδομένων (Datasets) και σε εσωτερικές διεργασίες του .NET .

AboutBox.h

```
#include "StdAfx.h"
#include "AboutBox.h"

#pragma once

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

namespace Account
{
    /// <summary>
    /// Summary for AboutBox
    ///This class displays a forms box with credits of the program
    /// </summary>
    public __gc class AboutBox : public System::Windows::Forms::Form
    {
    public:
        AboutBox(void)
        {
            InitializeComponent();
        }

    protected:
        void Dispose(Boolean disposing)
        {
            if (disposing && components)
            {
                components->Dispose();
            }
            __super::Dispose(disposing);
        }

    private: System::Windows::Forms::Label * label2;
    private: System::Windows::Forms::Label * label1;
    private: System::Windows::Forms::Button * button1;

    private:
        /// <summary>
        /// Required designer variable.
        /// </summary>
        System::ComponentModel::Container* components;

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        void InitializeComponent(void)
```

```

    {
        this->label2 = new System::Windows::Forms::Label();
        this->label1 = new System::Windows::Forms::Label();
        this->button1 = new System::Windows::Forms::Button();
        this->SuspendLayout();
        //
        // label2
        //
        this->label2->Location = System::Drawing::Point(18, 101);
        this->label2->Name = S"label2";
        this->label2->Size = System::Drawing::Size(216, 48);
        this->label2->TabIndex = 6;
        this->label2->Text = S"Contact info +306945102539 email: el99190@mail.ntua.gr
alexandergiamas@cmu.edu";
        //
        // label1
        //
        this->label1->Location = System::Drawing::Point(18, 45);
        this->label1->Name = S"label1";
        this->label1->Size = System::Drawing::Size(224, 40);
        this->label1->TabIndex = 5;
        this->label1->Text = S"Account++ application designed as part of diploma thesis project . Designed
and i"
                S"mplemented by Alexandros Giamas";
        //
        // button1
        //
        this->button1->Location = System::Drawing::Point(194, 189);
        this->button1->Name = S"button1";
        this->button1->Size = System::Drawing::Size(80, 32);
        this->button1->TabIndex = 4;
        this->button1->Text = S"OK";
        this->button1->Click += new System::EventHandler(this, button1_Click);
        //
        // AboutBox
        //
        this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
        this->ClientSize = System::Drawing::Size(292, 266);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->button1);
        this->Name = S"AboutBox";
        this->Text = S"AboutBox";
        this->ResumeLayout(false);
    }
private: System::Void button1_Click(System::Object * sender, System::EventArgs * e)
    {
        Close();
    }
};
} // adread.cpp
//

#using <mscorlib.dll>
using namespace System;
using namespace System::Collections;

#using <System.dll>

#using <System.DirectoryServices.dll>
using namespace System::DirectoryServices;

```

```

void main(int argc, char ** argv)
{
    if (argc != 2)
    {
        Console::WriteLine(String::Concat(S"Usage: ", Environment::GetCommandLineArgs()[0], S" <ad_path>"));
        Console::WriteLine();
        Console::WriteLine(S"Press any key to continue...");
        Console::ReadLine();
        return;
    }

    DirectoryEntry* objDirEnt = new DirectoryEntry(argv[1]);
    Console::WriteLine(String::Concat(S"Name      = ", objDirEnt->Name));
    Console::WriteLine(String::Concat(S"Path      = ", objDirEnt->Path));
    Console::WriteLine(String::Concat(S"SchemaClassName = ", objDirEnt->SchemaClassName));
    Console::WriteLine(S"");
    Console::WriteLine(S"Properties:");

    String *Key;
    Object* objValue;

    ICollection* PropCol = objDirEnt->Properties->PropertyNames;
    if (PropCol->Count > 0)
    {
        IEnumerator* Enum1 = PropCol->GetEnumerator();
        while (Enum1->MoveNext())
        {
            Key = Enum1->Current->ToString();
            Console::Write(S"\t{0} = ", Key);
            Console::WriteLine(S"");

            ICollection* ValCol = objDirEnt->Properties->get_Item(Key);
            if (ValCol->Count > 0)
            {
                IEnumerator* Enum2 = ValCol->GetEnumerator();
                while (Enum2->MoveNext())
                {
                    objValue = Enum2->Current;
                    Console::WriteLine(S"\t\t{0}", objValue);
                }
            }
        }
    }
}

```

```

#include "StdAfx.h"
#include ".adwrite.h"
using <mscorlib.dll>

```

```

ADWrite::ADWrite(String *a)
{
}

```

```

ADWrite::~ADWrite(void)
{
}

```

ADWrite.h

```

#pragma once

```

```

class ADWrite
{
public:
    ADWrite(void);
    ~ADWrite(void);
};
#include "stdafx.h"

using namespace System::Reflection;
using namespace System::Runtime::CompilerServices;

//
// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
//
[assembly:AssemblyTitleAttribute(")];
[assembly:AssemblyDescriptionAttribute(")];
[assembly:AssemblyConfigurationAttribute(")];
[assembly:AssemblyCompanyAttribute(")];
[assembly:AssemblyProductAttribute(")];
[assembly:AssemblyCopyrightAttribute(")];
[assembly:AssemblyTrademarkAttribute(")];
[assembly:AssemblyCultureAttribute(")];

//
// Version information for an assembly consists of the following four values:
//
//   Major Version
//   Minor Version
//   Build Number
//   Revision
//
// You can specify all the value or you can default the Revision and Build Numbers
// by using the '*' as shown below:

[assembly:AssemblyVersionAttribute("1.0.*")];

//
// In order to sign your assembly you must specify a key to use. Refer to the
// Microsoft .NET Framework documentation for more information on assembly signing.
//
// Use the attributes below to control which key is used for signing.
//
// Notes:
// (*) If no key is specified, the assembly is not signed.
// (*) KeyName refers to a key that has been installed in the Crypto Service
//   Provider (CSP) on your machine. KeyFile refers to a file which contains
//   a key.
// (*) If the KeyFile and the KeyName values are both specified, the
//   following processing occurs:
//   (1) If the KeyName can be found in the CSP, that key is used.
//   (2) If the KeyName does not exist and the KeyFile does exist, the key
//       in the KeyFile is installed into the CSP and used.
// (*) In order to create a KeyFile, you can use the sn.exe (Strong Name) utility.
//   When specifying the KeyFile, the location of the KeyFile should be
//   relative to the project directory.
// (*) Delay Signing is an advanced option - see the Microsoft .NET Framework
//   documentation for more information on this.
//
[assembly:AssemblyDelaySignAttribute(false)];

```



```
[assembly:AssemblyKeyFileAttribute("");  
[assembly:AssemblyKeyNameAttribute("");
```

AUser.cpp

```
#include "StdAfx.h"
#include "AUser.h"
```

```
#using <System.dll>
```

```
#using <System.DirectoryServices.dll>
#include <tchar.h>
```

AUser.h

```
#pragma once
```

```
using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
using namespace System::DirectoryServices;
```

```
namespace Account
```

```
{
```

```
    /// <summary>
```

```
    /// Summary for AUser
```

```
    /// add a user to the Active Directory and write his account data
```

```
    /// to the SQL Server database
```

```
    /// </summary>
```

```
    public __gc class AUser : public System::Windows::Forms::Form
```

```
    {
```

```
        System::String *a,*b,*e,*f;
```

```
        private: System::Data::SqlClient::SqlDataAdapter * sqlDataAdapter1;
```

```
        private: System::Data::SqlClient::SqlCommand * sqlSelectCommand1;
```

```
        private: System::Data::SqlClient::SqlCommand * sqlInsertCommand2;
```

```
        private: System::Data::SqlClient::SqlCommand * sqlInsertCommand3;
```

```
        private: System::Data::SqlClient::SqlConnection * sqlConnection1;
```

```
        private: System::Data::SqlClient::SqlCommand * sqlCommand1;
```

```
        private: System::Data::SqlClient::SqlCommand * sqlCommand2;
```

```
        private: Account::DataSet13 * dataSet131;
```

```
        private: System::Windows::Forms::Button * button2;
```

```
        private: System::Windows::Forms::Label * label2;
```

```
        public:
```

```
            int SQLWrite(){
```

```
                try{
```

```
                    System::String *con;
```

```
                    System::String *tb1, *tb2, *tb3, *tb5;
```

```
                    System::String *tb4, *tb6, *tb7;
```

```
                    tb4 = this->textBox4->Text->ToString();//mobile
```

```
                    tb6 = this->textBox6->Text->ToString();//adress
```

```
                    tb7 = this->textBox7->Text->ToString();//hours
```

```
                    tb1 = this->textBox1->Text->ToString();//name
```

```
                    tb2 = this->textBox2->Text->ToString();//surname
```

```
                    tb3 = this->textBox3->Text->ToString();//fixed phone
```

```
                    tb5 = this->textBox5->Text->ToString();//email
```

```
                    System::String *event_id;
```

```
                    event_id = String::Concat(tb1, tb2);
```

```
                    if(this->radioButton1->Checked) con="1";
```

```
                    if(this->radioButton2->Checked) con="2"; else con="3";
```

```
                    this->sqlInsertCommand2 = new System::Data::SqlClient::SqlCommand();
```

```
                    this->sqlDataAdapter1->InsertCommand = this->sqlInsertCommand2;
```

```

        this->sqlInsertCommand2->Connection = this->sqlConnection1;
        this->sqlConnection1->Open();
        this->sqlInsertCommand2->CommandText = String::Concat(S"INSERT INTO Subscriber(event_id,
name1, surname1, Telno, Email, Con) VALUES (' ,event_id, S"",",         tb1,         S"",",         tb2,         S"",",         tb3,
        S"",",         tb5,         S"",",         con,         S"");");
        this->sqlInsertCommand2->CommandType = CommandType::Text;
        this->sqlInsertCommand2->ExecuteNonQuery();

// Process results.

        this->sqlInsertCommand2->Connection->Close();

//sqlinsertcommand3

        this->sqlInsertCommand3 = new System::Data::SqlClient::SqlCommand();
        this->sqlDataAdapter1->InsertCommand = this->sqlInsertCommand3;
        this->sqlInsertCommand3->Connection = this->sqlConnection1;
        this->sqlConnection1->Open();
        this->sqlInsertCommand3->CommandText = String::Concat(S"INSERT INTO Subscriber2(event_id,
Mobno, address, hours) VALUES (' ,event_id, S"",",         tb4,         S"",",         tb6,         S"",",         tb7, S"");");
        this->sqlInsertCommand3->CommandType = CommandType::Text;
        this->sqlInsertCommand3->ExecuteNonQuery();
        this->sqlInsertCommand3->Connection->Close();
        //end sqlcommand3
    } catch(Exception *e)
    {
        MessageBox::Show(S"An error has occured please try later");
        MessageBox::Show(String::Concat(S"error:" ,e->ToString()));
        Console::WriteLine("error:" + e);
        exit(0);
    }
    return 0;
} //END OF SQLWRITE FUNCTION()

//String AddUser
int ADWrite(String *a){
    try
    {
        //Build the path, and then bind to computer.
        String *adsPath = String::Format(S"WinNT://{0},computer", Environment::MachineName);
        DirectoryEntry *computerEntry = new DirectoryEntry(adsPath);
        DirectoryEntry *userEntry = computerEntry->Children->Add(a, S"user");
        userEntry->Properties->get_Item(S"Description")->Add(S"Test user for HOTSPOTusers");
        //Set the password, and then commit the changes.
        String *stringArray[] = new String *[1];
            stringArray[0] = S"Agiamas#13"//##12353Abc
        userEntry->Invoke(S"SetPassword", stringArray);
            userEntry->CommitChanges();
        //Add the new user to a group.
        DirectoryEntry *groupEntry = computerEntry->Children->Find(S"HOTSPOTusers");
        stringArray[0] = userEntry->Path;
        groupEntry->Invoke(S"Add", stringArray);
            MessageBox::Show(S"added user to Active Directory successfully");
    }
    catch(Exception *e)
    {
        MessageBox::Show(S"Some kind of error has occured please try later");
        Console::Write(e);
    }
    return 0;
}
AUser(void)
{
    InitializeComponent();
}

```

```

    }
protected:
    void Dispose(Boolean disposing)
    {
        if (disposing && components)
        {
            components->Dispose();
        }
        __super::Dispose(disposing);
    }
private: System::Windows::Forms::Label * label1;
private: System::Windows::Forms::TextBox * textBox1;

private: System::Windows::Forms::Label * label3;
private: System::Windows::Forms::Label * label4;
private: System::Windows::Forms::Label * label5;
private: System::Windows::Forms::Label * label6;
private: System::Windows::Forms::Label * label7;
private: System::Windows::Forms::Label * label8;
private: System::Windows::Forms::TextBox * textBox2;
private: System::Windows::Forms::TextBox * textBox3;
private: System::Windows::Forms::TextBox * textBox4;
private: System::Windows::Forms::TextBox * textBox5;
private: System::Windows::Forms::TextBox * textBox6;
private: System::Windows::Forms::RadioButton * radioButton1;
private: System::Windows::Forms::RadioButton * radioButton2;
private: System::Windows::Forms::TextBox * textBox7;
private: System::Windows::Forms::Label * label9;
private: System::Windows::Forms::Button * button1;
private: System::Windows::Forms::Label * label10;

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container* components;

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        this->label1 = new System::Windows::Forms::Label();
        this->textBox1 = new System::Windows::Forms::TextBox();
        this->label3 = new System::Windows::Forms::Label();
        this->label4 = new System::Windows::Forms::Label();
        this->label5 = new System::Windows::Forms::Label();
        this->label6 = new System::Windows::Forms::Label();
        this->label7 = new System::Windows::Forms::Label();
        this->label8 = new System::Windows::Forms::Label();
        this->textBox2 = new System::Windows::Forms::TextBox();
        this->textBox3 = new System::Windows::Forms::TextBox();
        this->textBox4 = new System::Windows::Forms::TextBox();
        this->textBox5 = new System::Windows::Forms::TextBox();
        this->textBox6 = new System::Windows::Forms::TextBox();
        this->radioButton1 = new System::Windows::Forms::RadioButton();
        this->radioButton2 = new System::Windows::Forms::RadioButton();
        this->textBox7 = new System::Windows::Forms::TextBox();
        this->label9 = new System::Windows::Forms::Label();
    }

```

```

this->button1 = new System::Windows::Forms::Button();
this->label10 = new System::Windows::Forms::Label();
this->label2 = new System::Windows::Forms::Label();
this->sqlDataAdapter1 = new System::Data::SqlClient::SqlDataAdapter();
this->sqlSelectCommand1 = new System::Data::SqlClient::SqlCommand();
this->sqlConnection1 = new System::Data::SqlClient::SqlConnection();
this->sqlCommand1 = new System::Data::SqlClient::SqlCommand();
this->sqlCommand2 = new System::Data::SqlClient::SqlCommand();
this->dataSet131 = new Account::DataSet13();
this->button2 = new System::Windows::Forms::Button();
(__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet131))->BeginInit();
this->SuspendLayout();
//
// label1
//
this->label1->Font = new System::Drawing::Font(S"Microsoft Sans Serif", 10,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point, (System::Byte)0);
this->label1->Location = System::Drawing::Point(8, 32);
this->label1->Name = S"label1";
this->label1->Size = System::Drawing::Size(408, 40);
this->label1->TabIndex = 0;
this->label1->Text = S"Enter the appropriate details for the user you want to add to the system";
this->label1->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
//
// textBox1
//
this->textBox1->Location = System::Drawing::Point(120, 72);
this->textBox1->Name = S"textBox1";
this->textBox1->TabIndex = 1;
this->textBox1->Text = S"alex";
this->textBox1->TextChanged += new System::EventHandler(this, textBox1_TextChanged);
//
// label3
//
this->label3->Location = System::Drawing::Point(0, 104);
this->label3->Name = S"label3";
this->label3->TabIndex = 3;
this->label3->Text = S"Surname **";
//
// label4
//
this->label4->Location = System::Drawing::Point(0, 128);
this->label4->Name = S"label4";
this->label4->Size = System::Drawing::Size(100, 24);
this->label4->TabIndex = 4;
this->label4->Text = S"Fixed Telephone number **";
//
// label5
//
this->label5->Location = System::Drawing::Point(0, 160);
this->label5->Name = S"label5";
this->label5->TabIndex = 5;
this->label5->Text = S"Mobile number";
//
// label6
//
this->label6->Location = System::Drawing::Point(0, 184);
this->label6->Name = S"label6";
this->label6->TabIndex = 6;
this->label6->Text = S"Email **";
//
// label7

```

```

//
this->label7->Location = System::Drawing::Point(0, 216);
this->label7->Name = S"label7";
this->label7->TabIndex = 7;
this->label7->Text = S"Address";
//
// label8
//
this->label8->Location = System::Drawing::Point(280, 72);
this->label8->Name = S"label8";
this->label8->Size = System::Drawing::Size(100, 24);
this->label8->TabIndex = 8;
this->label8->Text = S"Type of Connection **";
//
// textBox2
//
this->textBox2->Location = System::Drawing::Point(120, 96);
this->textBox2->Name = S"textBox2";
this->textBox2->TabIndex = 9;
this->textBox2->Text = S"giamas";
//
// textBox3
//
this->textBox3->Location = System::Drawing::Point(120, 128);
this->textBox3->Name = S"textBox3";
this->textBox3->TabIndex = 10;
this->textBox3->Text = S"textBox3";
//
// textBox4
//
this->textBox4->Location = System::Drawing::Point(120, 152);
this->textBox4->Name = S"textBox4";
this->textBox4->TabIndex = 11;
this->textBox4->Text = S"textBox4";
//
// textBox5
//
this->textBox5->Location = System::Drawing::Point(120, 176);
this->textBox5->Name = S"textBox5";
this->textBox5->TabIndex = 12;
this->textBox5->Text = S"textBox5";
//
// textBox6
//
this->textBox6->Location = System::Drawing::Point(120, 216);
this->textBox6->Name = S"textBox6";
this->textBox6->TabIndex = 13;
this->textBox6->Text = S"textBox6";
//
// radioButton1
//
this->radioButton1->Location = System::Drawing::Point(224, 112);
this->radioButton1->Name = S"radioButton1";
this->radioButton1->TabIndex = 14;
this->radioButton1->Text = S"Prepaid";
//
// radioButton2
//
this->radioButton2->Location = System::Drawing::Point(328, 112);
this->radioButton2->Name = S"radioButton2";
this->radioButton2->TabIndex = 15;
this->radioButton2->Text = S"Subscriber";

```

```

//
// textBox7
//
this->textBox7->Location = System::Drawing::Point(224, 176);
this->textBox7->Name = S"textBox7";
this->textBox7->TabIndex = 16;
this->textBox7->Text = S"textBox7";
//
// label9
//
this->label9->Location = System::Drawing::Point(224, 136);
this->label9->Name = S"label9";
this->label9->Size = System::Drawing::Size(96, 32);
this->label9->TabIndex = 17;
this->label9->Text = S"How many minutes\?";
//
// button1
//
this->button1->Location = System::Drawing::Point(64, 264);
this->button1->Name = S"button1";
this->button1->Size = System::Drawing::Size(328, 40);
this->button1->TabIndex = 18;
this->button1->Text = S"Add new user!";
this->button1->Click += new System::EventHandler(this, button1_click);
//
// label10
//
this->label10->Font = new System::Drawing::Font(S"Microsoft Sans Serif", 8.25F,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point, (System::Byte)0);
this->label10->Location = System::Drawing::Point(8, 312);
this->label10->Name = S"label10";
this->label10->Size = System::Drawing::Size(424, 24);
this->label10->TabIndex = 19;
this->label10->Text = S"fields marked with ** are compulsory";
this->label10->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
//
// label2
//
this->label2->Location = System::Drawing::Point(0, 64);
this->label2->Name = S"label2";
this->label2->Size = System::Drawing::Size(100, 32);
this->label2->TabIndex = 2;
this->label2->Text = S"Name **";
//
// sqlDataAdapter1
//
this->sqlDataAdapter1->SelectCommand = this->sqlSelectCommand1;
System::Data::Common::DataTableMapping* __mcTemp__1[] = new
System::Data::Common::DataColumnMapping*[1];
System::Data::Common::DataColumnMapping* __mcTemp__2[] = new
System::Data::Common::DataColumnMapping*[6];
__mcTemp__2[0] = new System::Data::Common::DataColumnMapping(S"event_id", S"event_id");
__mcTemp__2[1] = new System::Data::Common::DataColumnMapping(S"name1", S"name1");
__mcTemp__2[2] = new System::Data::Common::DataColumnMapping(S"surname1",
S"surname1");
__mcTemp__2[3] = new System::Data::Common::DataColumnMapping(S"Telno", S"Telno");
__mcTemp__2[4] = new System::Data::Common::DataColumnMapping(S"Email", S"Email");
__mcTemp__2[5] = new System::Data::Common::DataColumnMapping(S"Con", S"Con");
__mcTemp__1[0] = new System::Data::Common::DataTableMapping(S"Table", S"Subscriber",
__mcTemp__2);
this->sqlDataAdapter1->TableMappings->AddRange(__mcTemp__1);

```

```

        this->sqlDataAdapter1->RowUpdated += new
System::Data::SqlClient::SqlRowUpdatedEventHandler(this, sqlDataAdapter1_RowUpdated);
        //
        // sqlSelectCommand1
        //
        this->sqlSelectCommand1->CommandText = S"SELECT event_id, name1, surname1, Telno, Email,
Con FROM Subscriber";
        this->sqlSelectCommand1->Connection = this->sqlConnection1;
        //
        // sqlConnection1
        //
        this->sqlConnection1->ConnectionString = S"workstation id=\"TELECOM-NIKIYAM\";packet
size=4096;integrated security=SSPI;data s"
        S"ource=\"TELECOM-NIKIYAM\";persist security info=False;initial catalog=store_data";
        //
        // sqlCommand1
        //
        this->sqlCommand1->CommandText = S"INSERT INTO Subscriber (event_id, name1, surname1,
Telno, Email, Con) VALUES ([1]"
        S", [2], [3], [4], [5], [6])";
        this->sqlCommand1->Connection = this->sqlConnection1;
        //
        // dataSet131
        //
        this->dataSet131->DataSetName = S"DataSet13";
        this->dataSet131->Locale = new System::Globalization::CultureInfo(S"en-US");
        //
        // button2
        //
        this->button2->Location = System::Drawing::Point(0, 0);
        this->button2->Name = S"button2";
        this->button2->Size = System::Drawing::Size(80, 23);
        this->button2->TabIndex = 20;
        this->button2->Text = S"back";
        this->button2->Click += new System::EventHandler(this, back_click);
        //
        // AUser
        //
        this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
        this->ClientSize = System::Drawing::Size(432, 349);
        this->Controls->Add(this->button2);
        this->Controls->Add(this->label10);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->label9);
        this->Controls->Add(this->textBox7);
        this->Controls->Add(this->textBox6);
        this->Controls->Add(this->textBox5);
        this->Controls->Add(this->textBox4);
        this->Controls->Add(this->textBox3);
        this->Controls->Add(this->textBox2);
        this->Controls->Add(this->textBox1);
        this->Controls->Add(this->radioButton2);
        this->Controls->Add(this->radioButton1);
        this->Controls->Add(this->label8);
        this->Controls->Add(this->label7);
        this->Controls->Add(this->label6);
        this->Controls->Add(this->label5);
        this->Controls->Add(this->label4);
        this->Controls->Add(this->label3);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->Name = S"AUser";

```



```

        this->Text = S"Add a new user";
        (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet131))->EndInit();
        this->ResumeLayout(false);

    }
private: System::Void label1_Click(System::Object * sender, System::EventArgs * e)
    {
    }

private: System::Void button1_click(System::Object * sender, System::EventArgs * e)
    {
        SQLWrite();
        ADWrite(String::Concat(this->textBox1->Text->ToString(),this->textBox2->Text-
>ToString()));
    }

private: System::Void textBox1_TextChanged(System::Object * sender, System::EventArgs * e)
    {
    }

private: System::Void sqlConnection1_InfoMessage(System::Object * sender, System::Data::SqlClient::SqlInfoMessageEventArgs
* e)
    {
    }

private: System::Void sqlDataAdapter1_RowUpdated(System::Object * sender,
System::Data::SqlClient::SqlRowUpdatedEventArgs * e)
    {
        MessageBox::Show(S"row updated successfully");
    }

private: System::Void back_click(System::Object * sender, System::EventArgs * e)
    {
        Close();
    }

};
}

```

CUser.cpp

```
#include "StdAfx.h"
#include "CUser.h"
```

CUser.h

```
#pragma once
#include "AUser.h"
#include "RUser.h"
```

```
using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
```

```
namespace Account
```

```
{
    /// <summary>
    /// Summary for CUser
    ///
    /// Change user's details in SQL Server DB and in Active Directory
    /// </summary>
    public __gc class CUser : public System::Windows::Forms::Form
    {
        private: System::Data::SqlClient::SqlCommand * sqlUpdateCommand1;
        private: System::Windows::Forms::Button * button2;
        private: System::Data::SqlClient::SqlCommand * sqlUpdateCommand2;

    public:
        int SQLUpdate(){
            try{
                System::String *con;
                System::String *tb1, *tb2, *tb3, *tb5;
                System::String *tb4, *tb6, *tb7;
                System::String *tb8;
                tb4 = this->textBox4->Text->ToString();//mobile
                tb6 = this->textBox6->Text->ToString();//adress
                tb7 = this->textBox7->Text->ToString();//hours
                tb1 = this->textBox1->Text->ToString();//name
                tb2 = this->textBox2->Text->ToString();//surname
                tb3 = this->textBox3->Text->ToString();//fixed phone
                tb5 = this->textBox5->Text->ToString();//email
                tb8 = this->textBox8->Text->ToString();//event_id (unique)
                System::String *event_id;
                event_id = String::Concat(tb1, tb2);
                if(this->radioButton1->Checked) con="1";
                if(this->radioButton2->Checked) con="2"; else con="3";
                this->sqlUpdateCommand1 = new System::Data::SqlClient::SqlCommand();
                this->sqlDataAdapter1->UpdateCommand = this->sqlUpdateCommand1;
                this->sqlUpdateCommand1->Connection = this->sqlConnection1;
                this->sqlConnection1->Open();
                this->sqlUpdateCommand1->CommandText = String::Concat(S"UPDATE Subscriber SET
event_id=" ,event_id, S", name1=", tb1, S", surname1=", tb2, S", Telno=", tb3, S", Email=",
tb5, S", con=", con, S" WHERE event_id=", tb8, S"");
                this->sqlUpdateCommand1->CommandType = CommandType::Text;
                this->sqlUpdateCommand1->ExecuteNonQuery();
                this->sqlUpdateCommand1->Connection->Close();
            }
            //sqlUpdateCommand2
            this->sqlUpdateCommand2 = new System::Data::SqlClient::SqlCommand();
        }
    };
}
```

```

        this->sqlDataAdapter1->InsertCommand = this->sqlUpdateCommand2;
        this->sqlUpdateCommand2->Connection = this->sqlConnection1;
        this->sqlConnection1->Open();
        this->sqlUpdateCommand2->CommandText = String::Concat(S"UPDATE Subscriber2 SET
event_id=" ,event_id, S"', Mobno="", tb4, S"',address="", tb6, S"', hours="", tb7, S"' WHERE event_id=",
tb8, S"");

        this->sqlUpdateCommand2->CommandType = CommandType::Text;
        this->sqlUpdateCommand2->ExecuteNonQuery();
        this->sqlUpdateCommand2->Connection->Close();
        //end sqlcommand3
    } catch(Exception *e)
    {
        MessageBox::Show(S"An error has occured please try later");
        MessageBox::Show(String::Concat(S"error:" ,e->ToString()));
        exit(0);
    }
    MessageBox::Show(S"Updated SQL server database");
return 0;
} //END OF SQLWRITE FUNCTION()

CUser(void)
{
    InitializeComponent();
}

protected:
void Dispose(Boolean disposing)
{
    if (disposing && components)
    {
        components->Dispose();
    }
    __super::Dispose(disposing);
}

private: System::Windows::Forms::Label * label1;
private: System::Windows::Forms::Label * label2;
private: System::Windows::Forms::Label * label3;
private: System::Windows::Forms::Label * label4;
private: System::Windows::Forms::Label * label5;
private: System::Windows::Forms::Label * label6;
private: System::Windows::Forms::Label * label7;
private: System::Windows::Forms::Label * label8;
private: System::Windows::Forms::RadioButton * radioButton1;
private: System::Windows::Forms::RadioButton * radioButton2;
private: System::Windows::Forms::TextBox * textBox1;
private: System::Windows::Forms::TextBox * textBox2;
private: System::Windows::Forms::TextBox * textBox3;
private: System::Windows::Forms::TextBox * textBox4;
private: System::Windows::Forms::TextBox * textBox5;
private: System::Windows::Forms::TextBox * textBox6;
private: System::Windows::Forms::TextBox * textBox7;
private: System::Windows::Forms::Label * label9;
private: System::Windows::Forms::Button * button1;
private: System::Windows::Forms::Label * label10;
private: System::Windows::Forms::Label * label11;
private: System::Windows::Forms::TextBox * textBox8;
private: System::Data::SqlClient::SqlDataAdapter * sqlDataAdapter1;
private: System::Data::SqlClient::SqlCommand * sqlSelectCommand1;

```

```

private: System::Data::SqlClient::SqlConnection * sqlConnection1;

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container* components;

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        this->label1 = new System::Windows::Forms::Label();
        this->label2 = new System::Windows::Forms::Label();
        this->label3 = new System::Windows::Forms::Label();
        this->label4 = new System::Windows::Forms::Label();
        this->label5 = new System::Windows::Forms::Label();
        this->label6 = new System::Windows::Forms::Label();
        this->label7 = new System::Windows::Forms::Label();
        this->label8 = new System::Windows::Forms::Label();
        this->radioButton1 = new System::Windows::Forms::RadioButton();
        this->radioButton2 = new System::Windows::Forms::RadioButton();
        this->textBox1 = new System::Windows::Forms::TextBox();
        this->textBox2 = new System::Windows::Forms::TextBox();
        this->textBox3 = new System::Windows::Forms::TextBox();
        this->textBox4 = new System::Windows::Forms::TextBox();
        this->textBox5 = new System::Windows::Forms::TextBox();
        this->textBox6 = new System::Windows::Forms::TextBox();
        this->textBox7 = new System::Windows::Forms::TextBox();
        this->label9 = new System::Windows::Forms::Label();
        this->button1 = new System::Windows::Forms::Button();
        this->label10 = new System::Windows::Forms::Label();
        this->label11 = new System::Windows::Forms::Label();
        this->textBox8 = new System::Windows::Forms::TextBox();
        this->sqlDataAdapter1 = new System::Data::SqlClient::SqlDataAdapter();
        this->sqlSelectCommand1 = new System::Data::SqlClient::SqlCommand();
        this->sqlConnection1 = new System::Data::SqlClient::SqlConnection();
        this->button2 = new System::Windows::Forms::Button();
        this->SuspendLayout();
        //
        // label1
        //
        this->label1->Location = System::Drawing::Point(16, 200);
        this->label1->Name = S"label1";
        this->label1->TabIndex = 0;
        this->label1->Text = S"Name **";
        //
        // label2
        //
        this->label2->Location = System::Drawing::Point(16, 240);
        this->label2->Name = S"label2";
        this->label2->TabIndex = 1;
        this->label2->Text = S"Surname **";
        //
        // label3
        //
        this->label3->Location = System::Drawing::Point(16, 272);
        this->label3->Name = S"label3";
        this->label3->TabIndex = 2;
    }

```

```

this->label3->Text = S"Fixed telephone number **";
this->label3->Click += new System::EventHandler(this, label3_Click);
//
// label4
//
this->label4->Location = System::Drawing::Point(16, 304);
this->label4->Name = S"label4";
this->label4->TabIndex = 3;
this->label4->Text = S"Mobile Number";
//
// label5
//
this->label5->Location = System::Drawing::Point(16, 336);
this->label5->Name = S"label5";
this->label5->TabIndex = 4;
this->label5->Text = S"Email **";
this->label5->Click += new System::EventHandler(this, label5_Click);
//
// label6
//
this->label6->Location = System::Drawing::Point(16, 368);
this->label6->Name = S"label6";
this->label6->TabIndex = 5;
this->label6->Text = S"Address";
//
// label7
//
this->label7->Location = System::Drawing::Point(304, 200);
this->label7->Name = S"label7";
this->label7->TabIndex = 6;
this->label7->Text = S"Type of connection **";
//
// label8
//
this->label8->Location = System::Drawing::Point(232, 272);
this->label8->Name = S"label8";
this->label8->Size = System::Drawing::Size(96, 32);
this->label8->TabIndex = 7;
this->label8->Text = S"How many minutes\?";
//
// radioButton1
//
this->radioButton1->Location = System::Drawing::Point(232, 232);
this->radioButton1->Name = S"radioButton1";
this->radioButton1->TabIndex = 8;
this->radioButton1->Text = S"Prepaid";
//
// radioButton2
//
this->radioButton2->Location = System::Drawing::Point(328, 232);
this->radioButton2->Name = S"radioButton2";
this->radioButton2->TabIndex = 9;
this->radioButton2->Text = S"Subscriber";
//
// textBox1
//
this->textBox1->Location = System::Drawing::Point(120, 200);
this->textBox1->Name = S"textBox1";
this->textBox1->TabIndex = 10;
this->textBox1->Text = S"textBox1";
//
// textBox2

```

```

//
this->textBox2->Location = System::Drawing::Point(120, 240);
this->textBox2->Name = S"textBox2";
this->textBox2->TabIndex = 11;
this->textBox2->Text = S"textBox2";
//
// textBox3
//
this->textBox3->Location = System::Drawing::Point(120, 272);
this->textBox3->Name = S"textBox3";
this->textBox3->TabIndex = 12;
this->textBox3->Text = S"textBox3";
//
// textBox4
//
this->textBox4->Location = System::Drawing::Point(120, 304);
this->textBox4->Name = S"textBox4";
this->textBox4->TabIndex = 13;
this->textBox4->Text = S"textBox4";
//
// textBox5
//
this->textBox5->Location = System::Drawing::Point(120, 336);
this->textBox5->Name = S"textBox5";
this->textBox5->TabIndex = 14;
this->textBox5->Text = S"textBox5";
//
// textBox6
//
this->textBox6->Location = System::Drawing::Point(120, 368);
this->textBox6->Name = S"textBox6";
this->textBox6->TabIndex = 15;
this->textBox6->Text = S"textBox6";
//
// textBox7
//
this->textBox7->Location = System::Drawing::Point(232, 312);
this->textBox7->Name = S"textBox7";
this->textBox7->TabIndex = 16;
this->textBox7->Text = S"textBox7";
//
// label9
//
this->label9->Font = new System::Drawing::Font(S"Microsoft Sans Serif", 10,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point, (System::Byte)0);
this->label9->Location = System::Drawing::Point(144, 16);
this->label9->Name = S"label9";
this->label9->Size = System::Drawing::Size(424, 32);
this->label9->TabIndex = 17;
this->label9->Text = S"Change user credentials";
this->label9->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
this->label9->Click += new System::EventHandler(this, label9_Click);
//
// button1
//
this->button1->Location = System::Drawing::Point(120, 400);
this->button1->Name = S"button1";
this->button1->Size = System::Drawing::Size(376, 56);
this->button1->TabIndex = 18;
this->button1->Text = S"Apply Changes!";
this->button1->Click += new System::EventHandler(this, button1_click);
//

```

```

// label10
//
this->label10->Font = new System::Drawing::Font(S"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Underline, System::Drawing::GraphicsUnit::Point, (System::Byte)0);
this->label10->Location = System::Drawing::Point(16, 64);
this->label10->Name = S"label10";
this->label10->Size = System::Drawing::Size(648, 48);
this->label10->TabIndex = 19;
this->label10->Text = S"Please enter the unique identifier name of the user you want to modify. In
case t"

S"his is incorrect no modifications will be made to the database.";
this->label10->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
//
// label11
//
this->label11->Location = System::Drawing::Point(160, 144);
this->label11->Name = S"label11";
this->label11->TabIndex = 20;
this->label11->Text = S"*unique* userid.";
//
// textBox8
//
this->textBox8->Location = System::Drawing::Point(312, 144);
this->textBox8->Name = S"textBox8";
this->textBox8->TabIndex = 21;
this->textBox8->Text = S"textBox8";
//
// sqlDataAdapter1
//
this->sqlDataAdapter1->SelectCommand = this->sqlSelectCommand1;
System::Data::Common::DataTableMapping* __mcTemp__1[] = new
System::Data::Common::DataTableMapping*[1];
System::Data::Common::DataColumnMapping* __mcTemp__2[] = new
System::Data::Common::DataColumnMapping*[6];
__mcTemp__2[0] = new System::Data::Common::DataColumnMapping(S"event_id", S"event_id");
__mcTemp__2[1] = new System::Data::Common::DataColumnMapping(S"name1", S"name1");
__mcTemp__2[2] = new System::Data::Common::DataColumnMapping(S"surname1",
S"surname1");
__mcTemp__2[3] = new System::Data::Common::DataColumnMapping(S"Telno", S"Telno");
__mcTemp__2[4] = new System::Data::Common::DataColumnMapping(S"Email", S"Email");
__mcTemp__2[5] = new System::Data::Common::DataColumnMapping(S"Con", S"Con");
__mcTemp__1[0] = new System::Data::Common::DataTableMapping(S"Table", S"Subscriber",
__mcTemp__2);
this->sqlDataAdapter1->TableMappings->AddRange(__mcTemp__1);
//
// sqlSelectCommand1
//
this->sqlSelectCommand1->CommandText = S"SELECT event_id, name1, surname1, Telno, Email,
Con FROM Subscriber";
this->sqlSelectCommand1->Connection = this->sqlConnection1;
//
// sqlConnection1
//
this->sqlConnection1->ConnectionString = S"workstation id=\\"TELECOM-NIKIYAM\";packet
size=4096;integrated security=SSPI;data s"
S"ource=\\"TELECOM-NIKIYAM\";persist security info=False;initial catalog=store_data";
//
// button2
//
this->button2->Location = System::Drawing::Point(0, 0);
this->button2->Name = S"button2";
this->button2->Size = System::Drawing::Size(80, 23);

```

```

        this->button2->TabIndex = 22;
        this->button2->Text = S"back";
        this->button2->Click += new System::EventHandler(this, back_click);
        //
        // CUser
        //
        this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
        this->ClientSize = System::Drawing::Size(680, 477);
        this->Controls->Add(this->button2);
        this->Controls->Add(this->textBox8);
        this->Controls->Add(this->textBox7);
        this->Controls->Add(this->textBox6);
        this->Controls->Add(this->textBox5);
        this->Controls->Add(this->textBox4);
        this->Controls->Add(this->textBox3);
        this->Controls->Add(this->textBox2);
        this->Controls->Add(this->textBox1);
        this->Controls->Add(this->label11);
        this->Controls->Add(this->label10);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->label9);
        this->Controls->Add(this->radioButton2);
        this->Controls->Add(this->radioButton1);
        this->Controls->Add(this->label8);
        this->Controls->Add(this->label7);
        this->Controls->Add(this->label6);
        this->Controls->Add(this->label5);
        this->Controls->Add(this->label4);
        this->Controls->Add(this->label3);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->Name = S"CUser";
        this->Text = S"Change user credentials";
        this->ResumeLayout(false);
    }
private: System::Void label3_Click(System::Object * sender, System::EventArgs * e)
    {
    }

private: System::Void label5_Click(System::Object * sender, System::EventArgs * e)
    {
    }

private: System::Void label9_Click(System::Object * sender, System::EventArgs * e)
    {
    }

private: System::Void button1_click(System::Object * sender, System::EventArgs * e)
    {SQLUpdate();
//First remove the user from the AD and then rewrite him with his new name...
RUser* rusr1 = new RUser();
rusr1->ADDelete(this->textBox8->Text->ToString());
AUser* ausr1 = new AUser();
ausr1->ADWrite(String::Concat(this->textBox1->Text->ToString(), this->textBox2->Text->ToString()));
MessageBox::Show(S"User's credentials updated successfully", S"Change user's data",
MessageBoxButtons::OK, MessageBoxIcon::Exclamation);
    }

private: System::Void back_click(System::Object * sender, System::EventArgs * e)
    {Close();
    }

```



```
};  
}#pragma once
```

DataSet3.h

```
#using <System.dll>  
#using <System.Data.dll>  
#using <System.Xml.dll>  
#using <microsoft.VisualBasic.dll>
```

```
namespace Account {  
    using namespace System;  
    using namespace System::Data;  
    using namespace System::Xml;  
    using namespace System::Runtime::Serialization;
```

```
    using namespace System;  
    public __gc class DataSet3;
```

```
    [Serializable,  
     System::ComponentModel::DesignerCategoryAttribute(S"code"),  
     System::Diagnostics::DebuggerStepThrough,  
     System::ComponentModel::ToolboxItem(true)]  
    public __gc class DataSet3 : public System::Data::DataSet {  
        public : __gc class event_dataDataTable;  
        public : __gc class event_dataRow;  
        public : __gc class event_dataRowChangeEvent;  
        public : __gc class event_mainDataTable;  
        public : __gc class event_mainRow;  
        public : __gc class event_mainRowChangeEvent;  
  
        private: Account::DataSet3::event_dataDataTable * tableevent_data;  
  
        private: Account::DataSet3::event_mainDataTable * tableevent_main;  
  
        public: DataSet3();  
        protected: DataSet3(System::Runtime::Serialization::SerializationInfo * info,  
System::Runtime::Serialization::StreamingContext context);  
        public: [property: System::ComponentModel::Browsable(false),  
property:  
System::ComponentModel::DesignerSerializationVisibilityAttribute(System::ComponentModel::DesignerSerializationVisibility::C  
ontent)]  
        __property Account::DataSet3::event_dataDataTable * get_event_data();  
  
        public: [property: System::ComponentModel::Browsable(false),  
property:  
System::ComponentModel::DesignerSerializationVisibilityAttribute(System::ComponentModel::DesignerSerializationVisibility::C  
ontent)]  
        __property Account::DataSet3::event_mainDataTable * get_event_main();  
  
        public: virtual System::Data::DataSet * Clone();  
  
        protected: virtual System::Boolean ShouldSerializeTables();  
  
        protected: virtual System::Boolean ShouldSerializeRelations();  
  
        protected: virtual System::Void ReadXmlSerializable(System::Xml::XmlReader * reader);  
  
        protected: virtual System::Xml::Schema::XmlSchema * GetSchemaSerializable();
```

```

public private: System::Void InitVars();

private: System::Void InitClass();

private: System::Boolean ShouldSerializeevent_data();

private: System::Boolean ShouldSerializeevent_main();

private: System::Void SchemaChanged(System::Object * sender, System::ComponentModel::CollectionChangeEventArgs *
e);

public: __delegate System::Void event_dataRowChangeEventHandler(System::Object * sender,
Account::DataSet3::event_dataRowChangeEvent * e);

public: __delegate System::Void event_mainRowChangeEventHandler(System::Object * sender,
Account::DataSet3::event_mainRowChangeEvent * e);

public : [System::Diagnostics::DebuggerStepThrough]
__gc class event_dataDataTable : public System::Data::DataTable, public System::Collections::IEnumerable {

private: System::Data::DataColumn * columnevent_id;

private: System::Data::DataColumn * columnattribute_type;

private: System::Data::DataColumn * columnattribute_value;

public private: event_dataDataTable();
public private: event_dataDataTable(System::Data::DataTable * table);
public: [property: System::ComponentModel::Browsable(false)]
__property System::Int32 get_Count();

public private: __property System::Data::DataColumn * get_event_idColumn();

public private: __property System::Data::DataColumn * get_attribute_typeColumn();

public private: __property System::Data::DataColumn * get_attribute_valueColumn();

public: __property Account::DataSet3::event_dataRow * get_Item(System::Int32 index);

public: __event Account::DataSet3::event_dataRowChangeEventHandler * event_dataRowChanged;

public: __event Account::DataSet3::event_dataRowChangeEventHandler * event_dataRowChanging;

public: __event Account::DataSet3::event_dataRowChangeEventHandler * event_dataRowDeleted;

public: __event Account::DataSet3::event_dataRowChangeEventHandler * event_dataRowDeleting;

public: System::Void Addevent_dataRow(Account::DataSet3::event_dataRow * row);

public: Account::DataSet3::event_dataRow * Addevent_dataRow(System::Guid event_id, System::String * attribute_type,
System::String * attribute_value);

public: System::Collections::IEnumerator * GetEnumerator();

public: virtual System::Data::DataTable * Clone();

protected: virtual System::Data::DataTable * CreateInstance();

public private: System::Void InitVars();

private: System::Void InitClass();

```

```

public: Account::DataSet3::event_dataRow * Newevent_dataRow();

protected: virtual System::Data::DataRow * NewRowFromBuilder(System::Data::DataRowBuilder * builder);

protected: virtual System::Type * GetRowType();

protected: virtual System::Void OnRowChanged(System::Data::DataRowChangeEventArgs * e);

protected: virtual System::Void OnRowChanging(System::Data::DataRowChangeEventArgs * e);

protected: virtual System::Void OnRowDeleted(System::Data::DataRowChangeEventArgs * e);

protected: virtual System::Void OnRowDeleting(System::Data::DataRowChangeEventArgs * e);

public: System::Void Removeevent_dataRow(Account::DataSet3::event_dataRow * row);
};

public : [System::Diagnostics::DebuggerStepThrough]
__gc class event_dataRow : public System::Data::DataRow {

private: Account::DataSet3::event_dataDataTable * tableevent_data;

public private: event_dataRow(System::Data::DataRowBuilder * rb);
public: __property System::Guid get_event_id();
public: __property void set_event_id(System::Guid value);

public: __property System::String * get_attribute_type();
public: __property void set_attribute_type(System::String * value);

public: __property System::String * get_attribute_value();
public: __property void set_attribute_value(System::String * value);

public: System::Boolean Isattribute_valueNull();

public: System::Void Setattribute_valueNull();
};

public : [System::Diagnostics::DebuggerStepThrough]
__gc class event_dataRowChangeEvent : public System::EventArgs {

private: Account::DataSet3::event_dataRow * eventRow;

private: System::Data::DataRowAction eventAction;

public: event_dataRowChangeEvent(Account::DataSet3::event_dataRow * row, System::Data::DataRowAction action);
public: __property Account::DataSet3::event_dataRow * get_Row();

public: __property System::Data::DataRowAction get_Action();
};

public : [System::Diagnostics::DebuggerStepThrough]
__gc class event_mainDataTable : public System::Data::DataTable, public System::Collections::IEnumerable {

private: System::Data::DataColumn * columnevent_id;

private: System::Data::DataColumn * columnrecord_timestamp;

public private: event_mainDataTable();
public private: event_mainDataTable(System::Data::DataTable * table);
public: [property: System::ComponentModel::Browsable(false)]
__property System::Int32 get_Count();

```

```

public private: __property System::Data::DataColumn * get_event_idColumn();

public private: __property System::Data::DataColumn * get_record_timestampColumn();

public: __property Account::DataSet3::event_mainRow * get_Item(System::Int32 index);

public: __event Account::DataSet3::event_mainRowChangeEventHandler * event_mainRowChanged;

public: __event Account::DataSet3::event_mainRowChangeEventHandler * event_mainRowChanging;

public: __event Account::DataSet3::event_mainRowChangeEventHandler * event_mainRowDeleted;

public: __event Account::DataSet3::event_mainRowChangeEventHandler * event_mainRowDeleting;

public: System::Void Addevent_mainRow(Account::DataSet3::event_mainRow * row);

public: Account::DataSet3::event_mainRow * Addevent_mainRow(System::Guid event_id, System::DateTime
record_timestamp);

public: Account::DataSet3::event_mainRow * FindByevent_id(System::Guid event_id);

public: System::Collections::IEnumerator * GetEnumerator();

public: virtual System::Data::DataTable * Clone();

protected: virtual System::Data::DataTable * CreateInstance();

public private: System::Void InitVars();

private: System::Void InitClass();

public: Account::DataSet3::event_mainRow * Newevent_mainRow();

protected: virtual System::Data::DataRow * NewRowFromBuilder(System::Data::DataRowBuilder * builder);

protected: virtual System::Type * GetRowType();

protected: virtual System::Void OnRowChanged(System::Data::DataRowChangeEventArgs * e);

protected: virtual System::Void OnRowChanging(System::Data::DataRowChangeEventArgs * e);

protected: virtual System::Void OnRowDeleted(System::Data::DataRowChangeEventArgs * e);

protected: virtual System::Void OnRowDeleting(System::Data::DataRowChangeEventArgs * e);

public: System::Void Removeevent_mainRow(Account::DataSet3::event_mainRow * row);
};

public : [System::Diagnostics::DebuggerStepThrough]
__gc class event_mainRow : public System::Data::DataRow {

private: Account::DataSet3::event_mainDataTable * tableevent_main;

public private: event_mainRow(System::Data::DataRowBuilder * rb);
public: __property System::Guid get_event_id();
public: __property void set_event_id(System::Guid value);

public: __property System::DateTime get_record_timestamp();
public: __property void set_record_timestamp(System::DateTime value);
};

public : [System::Diagnostics::DebuggerStepThrough]

```

```

__gc class event_mainRowChangeEvent : public System::EventArgs {

    private: Account::DataSet3::event_mainRow * eventRow;

    private: System::Data::DataRowAction eventAction;

    public: event_mainRowChangeEvent(Account::DataSet3::event_mainRow * row, System::Data::DataRowAction action);
    public: __property Account::DataSet3::event_mainRow * get_Row();

    public: __property System::Data::DataRowAction get_Action();
};

};

inline DataSet3::DataSet3() {
    this->InitClass();
    System::ComponentModel::CollectionChangeEventHandler * schemaChangedHandler = new
System::ComponentModel::CollectionChangeEventHandler(this, SchemaChanged);
    this->Tables->CollectionChanged += schemaChangedHandler;
    this->Relations->CollectionChanged += schemaChangedHandler;
}

inline DataSet3::DataSet3(System::Runtime::Serialization::SerializationInfo * info,
System::Runtime::Serialization::StreamingContext context) {
    System::String * strSchema = (__try_cast<System::String * >(info->GetValue(S"XmlSchema", __typeof(System::String))));
    if ((strSchema != 0)) {
        System::Data::DataSet * ds = new System::Data::DataSet();
        ds->ReadXmlSchema(new System::Xml::XmlTextReader(new System::IO::StringReader(strSchema)));
        if ((ds->Tables->Item[S"event_data"] != 0)) {
            this->Tables->Add(new Account::DataSet3::event_dataDataTable(ds->Tables->Item[S"event_data"]));
        }
        if ((ds->Tables->Item[S"event_main"] != 0)) {
            this->Tables->Add(new Account::DataSet3::event_mainDataTable(ds->Tables->Item[S"event_main"]));
        }
        this->DataSetName = ds->DataSetName;
        this->Prefix = ds->Prefix;
        this->Namespace = ds->Namespace;
        this->Locale = ds->Locale;
        this->CaseSensitive = ds->CaseSensitive;
        this->EnforceConstraints = ds->EnforceConstraints;
        this->Merge(ds, false, System::Data::MissingSchemaAction::Add);
        this->InitVars();
    }
    else {
        this->InitClass();
    }
    this->GetSerializationData(info, context);
    System::ComponentModel::CollectionChangeEventHandler * schemaChangedHandler = new
System::ComponentModel::CollectionChangeEventHandler(this, SchemaChanged);
    this->Tables->CollectionChanged += schemaChangedHandler;
    this->Relations->CollectionChanged += schemaChangedHandler;
}

inline Account::DataSet3::event_dataDataTable * DataSet3::get_event_data() {
    return this->tableevent_data;
}

inline Account::DataSet3::event_mainDataTable * DataSet3::get_event_main() {
    return this->tableevent_main;
}

```

```

inline System::Data::DataSet * DataSet3::Clone() {
    Account::DataSet3 * cln = (__try_cast<Account::DataSet3 * >(__super::Clone()));
    cln->InitVars();
    return cln;
};

inline System::Boolean DataSet3::ShouldSerializeTables() {
    return false;
};

inline System::Boolean DataSet3::ShouldSerializeRelations() {
    return false;
};

inline System::Void DataSet3::ReadXmlSerializable(System::Xml::XmlReader * reader) {
    this->Reset();
    System::Data::DataSet * ds = new System::Data::DataSet();
    ds->ReadXml(reader);
    if ((ds->Tables->Item[S"event_data"] != 0)) {
        this->Tables->Add(new Account::DataSet3::event_dataDataTable(ds->Tables->Item[S"event_data"]));
    }
    if ((ds->Tables->Item[S"event_main"] != 0)) {
        this->Tables->Add(new Account::DataSet3::event_mainDataTable(ds->Tables->Item[S"event_main"]));
    }
    this->DataSetName = ds->DataSetName;
    this->Prefix = ds->Prefix;
    this->Namespace = ds->Namespace;
    this->Locale = ds->Locale;
    this->CaseSensitive = ds->CaseSensitive;
    this->EnforceConstraints = ds->EnforceConstraints;
    this->Merge(ds, false, System::Data::MissingSchemaAction::Add);
    this->InitVars();
};

inline System::Xml::Schema::XmlSchema * DataSet3::GetSchemaSerializable() {
    System::IO::MemoryStream * stream = new System::IO::MemoryStream();
    this->WriteXmlSchema(new System::Xml::XmlTextWriter(stream, 0));
    stream->Position = 0;
    return System::Xml::Schema::XmlSchema::Read(new System::Xml::XmlTextReader(stream), 0);
};

inline System::Void DataSet3::InitVars() {
    this->tableevent_data = (__try_cast<Account::DataSet3::event_dataDataTable * >(this->Tables->Item[S"event_data"]));
    if ((this->tableevent_data != 0)) {
        this->tableevent_data->InitVars();
    }
    this->tableevent_main = (__try_cast<Account::DataSet3::event_mainDataTable * >(this->Tables->Item[S"event_main"]));
    if ((this->tableevent_main != 0)) {
        this->tableevent_main->InitVars();
    }
};

inline System::Void DataSet3::InitClass() {
    this->DataSetName = S"DataSet3";
    this->Prefix = S"";
    this->Namespace = S"http://www.tempuri.org/DataSet3.xsd";
    this->Locale = new System::Globalization::CultureInfo(S"en-GB");
    this->CaseSensitive = false;
    this->EnforceConstraints = true;
    this->tableevent_data = new Account::DataSet3::event_dataDataTable();
    this->Tables->Add(this->tableevent_data);
};

```

```

        this->tableevent_main = new Account::DataSet3::event_mainDataTable();
        this->Tables->Add(this->tableevent_main);
    };

    inline System::Boolean DataSet3::ShouldSerializeevent_data() {
        return false;
    };

    inline System::Boolean DataSet3::ShouldSerializeevent_main() {
        return false;
    };

    inline System::Void DataSet3::SchemaChanged(System::Object * sender,
System::ComponentModel::CollectionChangeEventArgs * e) {
        if ((e->Action == System::ComponentModel::CollectionChangeAction::Remove)) {
            this->InitVars();
        }
    };

    inline DataSet3::event_dataDataTable::event_dataDataTable() :
        DataTable(S"event_data") {
        this->InitClass();
    }

    inline DataSet3::event_dataDataTable::event_dataDataTable(System::Data::DataTable * table) :
        DataTable(table->TableName) {
        if ((table->CaseSensitive != table->DataSet->CaseSensitive)) {
            this->CaseSensitive = table->CaseSensitive;
        }
        if ((table->Locale->ToString() != table->DataSet->Locale->ToString())) {
            this->Locale = table->Locale;
        }
        if ((table->Namespace != table->DataSet->Namespace)) {
            this->Namespace = table->Namespace;
        }
        this->Prefix = table->Prefix;
        this->MinimumCapacity = table->MinimumCapacity;
        this->DisplayExpression = table->DisplayExpression;
    }

    inline System::Int32 DataSet3::event_dataDataTable::get_Count() {
        return this->Rows->Count;
    }

    inline System::Data::DataColumn * DataSet3::event_dataDataTable::get_event_idColumn() {
        return this->columnevent_id;
    }

    inline System::Data::DataColumn * DataSet3::event_dataDataTable::get_attribute_typeColumn() {
        return this->columnattribute_type;
    }

    inline System::Data::DataColumn * DataSet3::event_dataDataTable::get_attribute_valueColumn() {
        return this->columnattribute_value;
    }

```

```

inline Account::DataSet3::event_dataRow * DataSet3::event_dataDataTable::get_Item(System::Int32 index) {
    return (__try_cast<Account::DataSet3::event_dataRow * >(this->Rows->Item[index]));
}

inline System::Void DataSet3::event_dataDataTable::Addevent_dataRow(Account::DataSet3::event_dataRow * row) {
    this->Rows->Add(row);
};

inline Account::DataSet3::event_dataRow * DataSet3::event_dataDataTable::Addevent_dataRow(System::Guid event_id,
System::String * attribute_type, System::String * attribute_value) {
    Account::DataSet3::event_dataRow * rowevent_dataRow = (__try_cast<Account::DataSet3::event_dataRow * >(this-
>NewRow()));
    System::Object* __mcTemp__1[] = new System::Object*[3];
    __mcTemp__1[0] = __box(event_id);
    __mcTemp__1[1] = attribute_type;
    __mcTemp__1[2] = attribute_value;
    rowevent_dataRow->ItemArray = __mcTemp__1;
    this->Rows->Add(rowevent_dataRow);
    return rowevent_dataRow;
};

inline System::Collections::IEnumerator * DataSet3::event_dataDataTable::GetEnumerator() {
    return this->Rows->GetEnumerator();
};

inline System::Data::DataTable * DataSet3::event_dataDataTable::Clone() {
    Account::DataSet3::event_dataDataTable * cln = (__try_cast<Account::DataSet3::event_dataDataTable *
>(__super::Clone()));
    cln->InitVars();
    return cln;
};

inline System::Data::DataTable * DataSet3::event_dataDataTable::CreateInstance() {
    return new Account::DataSet3::event_dataDataTable();
};

inline System::Void DataSet3::event_dataDataTable::InitVars() {
    this->columnevent_id = this->Columns->Item[S"event_id"];
    this->columnattribute_type = this->Columns->Item[S"attribute_type"];
    this->columnattribute_value = this->Columns->Item[S"attribute_value"];
};

inline System::Void DataSet3::event_dataDataTable::InitClass() {
    this->columnevent_id = new System::Data::DataColumn(S"event_id", __typeof(System::Guid), 0,
System::Data::MappingType::Element);
    this->Columns->Add(this->columnevent_id);
    this->columnattribute_type = new System::Data::DataColumn(S"attribute_type", __typeof(System::String), 0,
System::Data::MappingType::Element);
    this->Columns->Add(this->columnattribute_type);
    this->columnattribute_value = new System::Data::DataColumn(S"attribute_value", __typeof(System::String), 0,
System::Data::MappingType::Element);
    this->Columns->Add(this->columnattribute_value);
    this->columnevent_id->AllowDBNull = false;
    this->columnattribute_type->AllowDBNull = false;
};

inline Account::DataSet3::event_dataRow * DataSet3::event_dataDataTable::Newevent_dataRow() {
    return (__try_cast<Account::DataSet3::event_dataRow * >(this->NewRow()));
}

```



```

};

inline System::Data::DataRow * DataSet3::event_dataDataTable::NewRowFromBuilder(System::Data::DataRowBuilder *
builder) {
    return new Account::DataSet3::event_dataRow(builder);
};

inline System::Type * DataSet3::event_dataDataTable::GetRowType() {
    return __typeof(Account::DataSet3::event_dataRow);
};

inline System::Void DataSet3::event_dataDataTable::OnRowChanged(System::Data::DataRowChangeEventArgs * e) {
    __super::OnRowChanged(e);
    if ((this->event_dataRowChanged != 0)) {
        this->event_dataRowChanged(this, new
Account::DataSet3::event_dataRowChangeEvent((__try_cast<Account::DataSet3::event_dataRow * >(e->Row)), e->Action));
    }
};

inline System::Void DataSet3::event_dataDataTable::OnRowChanging(System::Data::DataRowChangeEventArgs * e) {
    __super::OnRowChanging(e);
    if ((this->event_dataRowChanging != 0)) {
        this->event_dataRowChanging(this, new
Account::DataSet3::event_dataRowChangeEvent((__try_cast<Account::DataSet3::event_dataRow * >(e->Row)), e->Action));
    }
};

inline System::Void DataSet3::event_dataDataTable::OnRowDeleted(System::Data::DataRowChangeEventArgs * e) {
    __super::OnRowDeleted(e);
    if ((this->event_dataRowDeleted != 0)) {
        this->event_dataRowDeleted(this, new
Account::DataSet3::event_dataRowChangeEvent((__try_cast<Account::DataSet3::event_dataRow * >(e->Row)), e->Action));
    }
};

inline System::Void DataSet3::event_dataDataTable::OnRowDeleting(System::Data::DataRowChangeEventArgs * e) {
    __super::OnRowDeleting(e);
    if ((this->event_dataRowDeleting != 0)) {
        this->event_dataRowDeleting(this, new
Account::DataSet3::event_dataRowChangeEvent((__try_cast<Account::DataSet3::event_dataRow * >(e->Row)), e->Action));
    }
};

inline System::Void DataSet3::event_dataDataTable::Removeevent_dataRow(Account::DataSet3::event_dataRow * row) {
    this->Rows->Remove(row);
};

inline DataSet3::event_dataRow::event_dataRow(System::Data::DataRowBuilder * rb) :
    DataRow(rb) {
    this->tableevent_data = (__try_cast<Account::DataSet3::event_dataDataTable * >(this->Table));
}

inline System::Guid DataSet3::event_dataRow::get_event_id() {
    return (*__try_cast<__box System::Guid * >(this->Item[this->tableevent_data->event_idColumn]));
}
inline void DataSet3::event_dataRow::set_event_id(System::Guid value) {
    this->Item[this->tableevent_data->event_idColumn] = __box(value);
}

inline System::String * DataSet3::event_dataRow::get_attribute_type() {

```

```

    return (__try_cast<System::String * >(this->Item[this->tableevent_data->attribute_typeColumn]));
}
inline void DataSet3::event_dataRow::set_attribute_type(System::String * value) {
    this->Item[this->tableevent_data->attribute_typeColumn] = value;
}

inline System::String * DataSet3::event_dataRow::get_attribute_value() {
    try {
        return (__try_cast<System::String * >(this->Item[this->tableevent_data->attribute_valueColumn]));
    }
    catch (System::InvalidCastException * e) {
        throw new System::Data::StrongTypingException(S"Cannot get value because it is DBNull.", e);
    }
}
inline void DataSet3::event_dataRow::set_attribute_value(System::String * value) {
    this->Item[this->tableevent_data->attribute_valueColumn] = value;
}

inline System::Boolean DataSet3::event_dataRow::Isattribute_valueNull() {
    return this->IsNull(this->tableevent_data->attribute_valueColumn);
};

inline System::Void DataSet3::event_dataRow::Setattribute_valueNull() {
    this->Item[this->tableevent_data->attribute_valueColumn] = System::Convert::DBNull;
};

inline DataSet3::event_dataRowChangeEvent::event_dataRowChangeEvent(Account::DataSet3::event_dataRow * row,
System::Data::DataRowAction action) {
    this->eventRow = row;
    this->eventAction = action;
}

inline Account::DataSet3::event_dataRow * DataSet3::event_dataRowChangeEvent::get_Row() {
    return this->eventRow;
}

inline System::Data::DataRowAction DataSet3::event_dataRowChangeEvent::get_Action() {
    return this->eventAction;
}

inline DataSet3::event_mainDataTable::event_mainDataTable() :
    DataTable(S"event_main") {
    this->InitClass();
}

inline DataSet3::event_mainDataTable::event_mainDataTable(System::Data::DataTable * table) :
    DataTable(table->TableName) {
    if ((table->CaseSensitive != table->DataSet->CaseSensitive)) {
        this->CaseSensitive = table->CaseSensitive;
    }
    if ((table->Locale->ToString() != table->DataSet->Locale->ToString())) {
        this->Locale = table->Locale;
    }
    if ((table->Namespace != table->DataSet->Namespace)) {
        this->Namespace = table->Namespace;
    }
}

```

```

    this->Prefix = table->Prefix;
    this->MinimumCapacity = table->MinimumCapacity;
    this->DisplayExpression = table->DisplayExpression;
}

inline System::Int32 DataSet3::event_mainDataTable::get_Count() {
    return this->Rows->Count;
}

inline System::Data::DataColumn * DataSet3::event_mainDataTable::get_event_idColumn() {
    return this->columnevent_id;
}

inline System::Data::DataColumn * DataSet3::event_mainDataTable::get_record_timestampColumn() {
    return this->columnrecord_timestamp;
}

inline Account::DataSet3::event_mainRow * DataSet3::event_mainDataTable::get_Item(System::Int32 index) {
    return (__try_cast<Account::DataSet3::event_mainRow * >(this->Rows->Item[index]));
}

inline System::Void DataSet3::event_mainDataTable::Addevent_mainRow(Account::DataSet3::event_mainRow * row) {
    this->Rows->Add(row);
};

inline Account::DataSet3::event_mainRow * DataSet3::event_mainDataTable::Addevent_mainRow(System::Guid event_id,
System::DateTime record_timestamp) {
    Account::DataSet3::event_mainRow * rowevent_mainRow = (__try_cast<Account::DataSet3::event_mainRow * >(this-
>NewRow()));
    System::Object* __mcTemp__1[] = new System::Object*[2];
    __mcTemp__1[0] = __box(event_id);
    __mcTemp__1[1] = __box(record_timestamp);
    rowevent_mainRow->ItemArray = __mcTemp__1;
    this->Rows->Add(rowevent_mainRow);
    return rowevent_mainRow;
};

inline Account::DataSet3::event_mainRow * DataSet3::event_mainDataTable::FindByevent_id(System::Guid event_id) {
    System::Object* __mcTemp__1[] = new System::Object*[1];
    __mcTemp__1[0] = __box(event_id);
    return (__try_cast<Account::DataSet3::event_mainRow * >(this->Rows->Find(__mcTemp__1)));
};

inline System::Collections::IEnumerator * DataSet3::event_mainDataTable::GetEnumerator() {
    return this->Rows->GetEnumerator();
};

inline System::Data::DataTable * DataSet3::event_mainDataTable::Clone() {
    Account::DataSet3::event_mainDataTable * cln = (__try_cast<Account::DataSet3::event_mainDataTable *
>(__super::Clone()));
    cln->InitVars();
    return cln;
};

inline System::Data::DataTable * DataSet3::event_mainDataTable::CreateInstance() {
    return new Account::DataSet3::event_mainDataTable();
};

```

```

inline System::Void DataSet3::event_mainDataTable::InitVars() {
    this->columnevent_id = this->Columns->Item[S"event_id"];
    this->columnrecord_timestamp = this->Columns->Item[S"record_timestamp"];
};

inline System::Void DataSet3::event_mainDataTable::InitClass() {
    this->columnevent_id = new System::Data::DataColumn(S"event_id", __typeof(System::Guid), 0,
System::Data::MappingType::Element);
    this->Columns->Add(this->columnevent_id);
    this->columnrecord_timestamp = new System::Data::DataColumn(S"record_timestamp", __typeof(System::DateTime), 0,
System::Data::MappingType::Element);
    this->Columns->Add(this->columnrecord_timestamp);
    System::Data::DataColumn* __mcTemp__1[] = new System::Data::DataColumn*[1];
    __mcTemp__1[0] = this->columnevent_id;
    this->Constraints->Add(new System::Data::UniqueConstraint(S"Constraint1", __mcTemp__1, true));
    this->columnevent_id->AllowDBNull = false;
    this->columnevent_id->Unique = true;
    this->columnrecord_timestamp->AllowDBNull = false;
};

inline Account::DataSet3::event_mainRow * DataSet3::event_mainDataTable::Newevent_mainRow() {
    return (__try_cast<Account::DataSet3::event_mainRow * >(this->NewRow()));
};

inline System::Data::DataRow * DataSet3::event_mainDataTable::NewRowFromBuilder(System::Data::DataRowBuilder *
builder) {
    return new Account::DataSet3::event_mainRow(builder);
};

inline System::Type * DataSet3::event_mainDataTable::GetRowType() {
    return __typeof(Account::DataSet3::event_mainRow);
};

inline System::Void DataSet3::event_mainDataTable::OnRowChanged(System::Data::DataRowChangeEventArgs * e) {
    __super::OnRowChanged(e);
    if ((this->event_mainRowChanged != 0)) {
        this->event_mainRowChanged(this, new
Account::DataSet3::event_mainRowChangeEvent((__try_cast<Account::DataSet3::event_mainRow * >(e->Row)), e->Action));
    }
};

inline System::Void DataSet3::event_mainDataTable::OnRowChanging(System::Data::DataRowChangeEventArgs * e) {
    __super::OnRowChanging(e);
    if ((this->event_mainRowChanging != 0)) {
        this->event_mainRowChanging(this, new
Account::DataSet3::event_mainRowChangeEvent((__try_cast<Account::DataSet3::event_mainRow * >(e->Row)), e->Action));
    }
};

inline System::Void DataSet3::event_mainDataTable::OnRowDeleted(System::Data::DataRowChangeEventArgs * e) {
    __super::OnRowDeleted(e);
    if ((this->event_mainRowDeleted != 0)) {
        this->event_mainRowDeleted(this, new
Account::DataSet3::event_mainRowChangeEvent((__try_cast<Account::DataSet3::event_mainRow * >(e->Row)), e->Action));
    }
};

inline System::Void DataSet3::event_mainDataTable::OnRowDeleting(System::Data::DataRowChangeEventArgs * e) {
    __super::OnRowDeleting(e);
    if ((this->event_mainRowDeleting != 0)) {
        this->event_mainRowDeleting(this, new
Account::DataSet3::event_mainRowChangeEvent((__try_cast<Account::DataSet3::event_mainRow * >(e->Row)), e->Action));
    }
};

```

```

    }
};

inline System::Void DataSet3::event_mainDataTable::Removeevent_mainRow(Account::DataSet3::event_mainRow * row) {
    this->Rows->Remove(row);
};

inline DataSet3::event_mainRow::event_mainRow(System::Data::DataRowBuilder * rb) :
    DataRow(rb) {
    this->tableevent_main = (__try_cast<Account::DataSet3::event_mainDataTable * >(this->Table));
}

inline System::Guid DataSet3::event_mainRow::get_event_id() {
    return (*__try_cast<__box System::Guid * >(this->Item[this->tableevent_main->event_idColumn]));
}
inline void DataSet3::event_mainRow::set_event_id(System::Guid value) {
    this->Item[this->tableevent_main->event_idColumn] = __box(value);
}

inline System::DateTime DataSet3::event_mainRow::get_record_timestamp() {
    return (*__try_cast<__box System::DateTime * >(this->Item[this->tableevent_main->record_timestampColumn]));
}
inline void DataSet3::event_mainRow::set_record_timestamp(System::DateTime value) {
    this->Item[this->tableevent_main->record_timestampColumn] = __box(value);
}

inline DataSet3::event_mainRowChangeEvent::event_mainRowChangeEvent(Account::DataSet3::event_mainRow * row,
System::Data::DataRowAction action) {
    this->eventRow = row;
    this->eventAction = action;
}

inline Account::DataSet3::event_mainRow * DataSet3::event_mainRowChangeEvent::get_Row() {
    return this->eventRow;
}

inline System::Data::DataRowAction DataSet3::event_mainRowChangeEvent::get_Action() {
    return this->eventAction;
}
}

```

DeleteADObject.h

```
#include "StdAfx.h"
#include ".\deleteadobject.h"
#include <mscorlib.dll>
/*****

DeleteADObject()

*****/

HRESULT DeleteADObject(LPOLESTR pwszAdsPath, LPWSTR pwszUsername, LPWSTR pwszPassword)
{
    HRESULT hr;
    IADs *pIADsToDelete = NULL;

    // Bind to the object to be deleted.

    // If a username and password are passed, use ADsOpenObject(), otherwise use ADsGetObject()
    if (!pwszUsername || 0 == *pwszUsername) // No user password passed - use ADsOpenObject.
    {
        hr = ADsGetObject( pwszAdsPath,
                          IID_IADs,
                          (void **)&pIADsToDelete);
    }
    else
    {
        hr = ADsOpenObject( pwszAdsPath,
                          pwszUsername,
                          pwszPassword,
                          ADS_SECURE_AUTHENTICATION,
                          IID_IADs,
                          (void**) &pIADsToDelete);
    }

    if (SUCCEEDED(hr))
    {
        BSTR bsParentPath;

        // Get the parent path.
        hr = pIADsToDelete->get_Parent(&bsParentPath);
        if(SUCCEEDED(hr))
        {
            VARIANT vCNTToDelete;

            VariantInit(&vCNTToDelete);

            // Get the CN property for the object to delete.
            hr = pIADsToDelete->Get(L"cn", &vCNTToDelete);
            if (SUCCEEDED(hr))
            {
                IDirectoryObject *pIDirObjectParent = NULL;

                /*****
                // Bind to the parent.
                // If a username and password are passed, use ADsOpenObject()
                // otherwise use ADsGetObject()
                if (!pwszUsername || 0 == *pwszUsername) // No user password passed - use ADsOpenObject.
                {
                    hr = ADsGetObject( bsParentPath,
                                        IID_IDirectoryObject,
                                        (void **)&pIDirObjectParent);
                }
                *****/
            }
        }
    }
}
```

```

    }
    else
    {
        hr = ADsOpenObject( bsParentPath,
                            pwszUsername,
                            pwszPassword,
                            ADS_SECURE_AUTHENTICATION,
                            IID_IDirectoryObject,
                            (void**)&pIDirObjectParent);
    }
    if (SUCCEEDED(hr))
    {
        // Release the object to delete.
        pIADsToDelete->Release();
        pIADsToDelete = NULL;

        LPWSTR pwszPrefix = L"CN=";
        LPWSTR pwszRDN = new WCHAR[lstrlenW(pwszPrefix) + lstrlenW(vCNTToDelete.bstrVal) + 1];
        if(pwszRDN)
        {
            lstrcpyW(pwszRDN, pwszPrefix);
            lstrcatW(pwszRDN, vCNTToDelete.bstrVal);

            // Instruct the parent to delete the child.
            hr = pIDirObjectParent->DeleteDSObject(pwszRDN);

            delete pwszRDN;
        }
        else
        {
            hr = E_OUTOFMEMORY;
        }

        // Release the Parent Object.
        pIDirObjectParent->Release();
        pIDirObjectParent = NULL;
    }

    VariantClear(&vCNTToDelete);
}

SysFreeString(bsParentPath);
}

// If a IADsObject is held, release it.
if ( pIADsToDelete)
{
    // Release the object to delete.
    pIADsToDelete->Release();
    pIADsToDelete = NULL;
}

return hr;
}

/*
DeleteADObject::DeleteADObject(void)
{

```

```
}  
  
DeleteADObject::~DeleteADObject(void)  
{  
}  
*#pragma once  
  
class DeleteADObject  
{  
public:  
    DeleteADObject(void);  
    ~DeleteADObject(void);  
};
```


ExportData.cpp

```
#include "StdAfx.h"
#include "DisconnectUser.h"
```

```
#include "StdAfx.h"
#include "Discuser.h"
```

```
#include "StdAfx.h"
#include "ExportData.h"
```

ExportData.h

```
#pragma once
```

```
using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
```

```
namespace Account
```

```
{
    /// <summary>
    /// Summary for ExportData
    ///
    /// Export data from the application to xml files
    /// </summary>
    public __gc class ExportData : public System::Windows::Forms::Form
    {
    public:
        ExportData(void)
        {
            InitializeComponent();
        }

    protected:
        void Dispose(Boolean disposing)
        {
            if (disposing && components)
            {
                components->Dispose();
            }
            __super::Dispose(disposing);
        }

    private: System::Windows::Forms::TextBox * textBox1;

    private: System::Windows::Forms::Label * label1;
    private: System::Windows::Forms::Label * label2;

    private: System::Windows::Forms::RadioButton * radioButton2;
    private: System::Windows::Forms::Button * button1;

    private: System::Data::SqlClient::SqlConnection * sqlConnection1;
```

```

private: System::Data::SqlClient::SqlDataAdapter * sqlDataAdapter1;
private: System::Windows::Forms::DataGrid * dataGrid1;
private: System::Data::SqlClient::SqlCommand * sqlSelectCommand1;
private: Account::DataSet1 * dataSet11;
private: System::Windows::Forms::Button * button2;

```

```
private:
```

```

/// <summary>
/// Required designer variable.
/// </summary>
System::ComponentModel::Container* components;

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
void InitializeComponent(void)
{
    this->textBox1 = new System::Windows::Forms::TextBox();
    this->label1 = new System::Windows::Forms::Label();
    this->label2 = new System::Windows::Forms::Label();
    this->radioButton2 = new System::Windows::Forms::RadioButton();
    this->button1 = new System::Windows::Forms::Button();
    this->sqlConnection1 = new System::Data::SqlClient::SqlConnection();
    this->sqlDataAdapter1 = new System::Data::SqlClient::SqlDataAdapter();
    this->sqlSelectCommand1 = new System::Data::SqlClient::SqlCommand();
    this->dataGrid1 = new System::Windows::Forms::DataGrid();
    this->dataSet11 = new Account::DataSet1();
    this->button2 = new System::Windows::Forms::Button();
    this->saveFileDialog1 = new System::Windows::Forms::SaveFileDialog();
    this->button3 = new System::Windows::Forms::Button();
    (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataGrid1))->BeginInit();
    (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet11))->BeginInit();
    this->SuspendLayout();
    //
    // textBox1
    //
    this->textBox1->Location = System::Drawing::Point(16, 48);
    this->textBox1->Name = S"textBox1";
    this->textBox1->Size = System::Drawing::Size(224, 20);
    this->textBox1->TabIndex = 0;
    this->textBox1->Text = S"textBox1";
    this->textBox1->TextChanged += new System::EventHandler(this, textbox_update);
    //
    // label1
    //
    this->label1->Location = System::Drawing::Point(88, 16);
    this->label1->Name = S"label1";
    this->label1->TabIndex = 2;
    this->label1->Text = S"Enter SQL query:";
    this->label1->Click += new System::EventHandler(this, label1_Click);
    //
    // label2
    //
    this->label2->Location = System::Drawing::Point(72, 72);
    this->label2->Name = S"label2";
    this->label2->Size = System::Drawing::Size(100, 40);
    this->label2->TabIndex = 3;
    this->label2->Text = S"select the option below";
}

```

```

this->label2->Click += new System::EventHandler(this, label2_Click);
//
// radioButton2
//
this->radioButton2->Location = System::Drawing::Point(72, 120);
this->radioButton2->Name = S"radioButton2";
this->radioButton2->TabIndex = 5;
this->radioButton2->Text = S"XML export";
this->radioButton2->Click += new System::EventHandler(this, label1_Click);
//
// button1
//
this->button1->Location = System::Drawing::Point(320, 16);
this->button1->Name = S"button1";
this->button1->Size = System::Drawing::Size(120, 24);
this->button1->TabIndex = 6;
this->button1->Text = S"Preview Data";
this->button1->Click += new System::EventHandler(this, button_click);
//
// sqlConnection1
//
this->sqlConnection1->ConnectionString = S"workstation id=\\"TELECOM-NIKIYAM\\";packet
size=4096;integrated security=SSPI;data s
        S"ource=\\"TELECOM-NIKIYAM\\";persist security info=False;initial catalog=store_data";
//
// sqlDataAdapter1
//
this->sqlDataAdapter1->SelectCommand = this->sqlSelectCommand1;
System::Data::Common::DataTableMapping* __mcTemp__1[] = new
System::Data::Common::DataTableMapping*[1];
System::Data::Common::DataColumnMapping* __mcTemp__2[] = new
System::Data::Common::DataColumnMapping*[5];
__mcTemp__2[0] = new System::Data::Common::DataColumnMapping(S"event_id", S"event_id");
__mcTemp__2[1] = new System::Data::Common::DataColumnMapping(S"attribute_type",
S"attribute_type");
__mcTemp__2[2] = new System::Data::Common::DataColumnMapping(S"attribute_value",
S"attribute_value");
__mcTemp__2[3] = new System::Data::Common::DataColumnMapping(S"Expr1", S"Expr1");
__mcTemp__2[4] = new System::Data::Common::DataColumnMapping(S"record_timestamp",
S"record_timestamp");
__mcTemp__1[0] = new System::Data::Common::DataTableMapping(S"Table", S"event_data",
__mcTemp__2);
this->sqlDataAdapter1->TableMappings->AddRange(__mcTemp__1);
//
// sqlSelectCommand1
//
this->sqlSelectCommand1->CommandText = S"SELECT event_data.event_id,
event_data.attribute_type, event_data.attribute_value"
        S", event_main.event_id AS Expr1, event_main.record_timestamp FROM event_data
CROS"
        S"S JOIN event_main";
this->sqlSelectCommand1->Connection = this->sqlConnection1;
//
// dataGrid1
//
this->dataGrid1->DataMember = S"";
this->dataGrid1->HeaderForeColor = System::Drawing::SystemColors::ControlText;
this->dataGrid1->Location = System::Drawing::Point(272, 48);
this->dataGrid1->Name = S"dataGrid1";
this->dataGrid1->Size = System::Drawing::Size(184, 160);
this->dataGrid1->TabIndex = 7;
//

```

```

// dataSet11
//
this->dataSet11->DataSetName = S"DataSet1";
this->dataSet11->Locale = new System::Globalization::CultureInfo(S"en-GB");
//
// button2
//
this->button2->Location = System::Drawing::Point(88, 208);
this->button2->Name = S"button2";
this->button2->TabIndex = 8;
this->button2->Text = S"GO!";
this->button2->Click += new System::EventHandler(this, buttonGO_click);
//
// button3
//
this->button3->Location = System::Drawing::Point(0, 0);
this->button3->Name = S"button3";
this->button3->Size = System::Drawing::Size(80, 23);
this->button3->TabIndex = 9;
this->button3->Text = S"back";
this->button3->Click += new System::EventHandler(this, back_click);
//
// ExportData
//
this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
this->ClientSize = System::Drawing::Size(544, 273);
this->Controls->Add(this->button3);
this->Controls->Add(this->button2);
this->Controls->Add(this->dataGrid1);
this->Controls->Add(this->button1);
this->Controls->Add(this->radioButton2);
this->Controls->Add(this->label2);
this->Controls->Add(this->label1);
this->Controls->Add(this->textBox1);
this->Name = S"ExportData";
this->Text = S"ExportData";
(__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataGrid1))->EndInit();
(__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet11))->EndInit();
this->ResumeLayout(false);
}

private: System::Void label1_Click(System::Object * sender, System::EventArgs * e)
{
    label11=true;
}

private: System::Void label2_Click(System::Object * sender, System::EventArgs * e)
{
    label22=true;
}

private: System::Void textbox_update(System::Object * sender, System::EventArgs * e)
{
    this->textBox1->get_SelectedText();
}

private: System::Void button_click(System::Object * sender, System::EventArgs * e)
{
    try{
        sqlDataAdapter1->SelectCommand->CommandText = textBox1->Text;
        dataSet11->Clear();
        if(dataSet11->Tables->get_Item(S"Results"))
        {
            dataSet11->Tables->get_Item(S"Results")->Columns->Clear();
        }
    }
}

```

```

        sqlDataAdapter1->SelectCommand->Connection->set_ConnectionString(S"workstation
id=wiron;packet size=4096;user id=sa;data source=wiron;persist security info=True;initial
catalog=store_data;password=agiamas13");
        sqlDataAdapter1->Fill(dataSet11, S"Results");
        dataGrid1->SetDataBinding( dataSet11, S"Results");
    }catch(Exception *sqlException)
    {
        MessageBox::Show( sqlException->Message, S"Invalid query", MessageBoxButtons::OK,
MessageBoxIcon::Error);
    }
    //end catch
} //end button_click

private: System::Void buttonGO_click(System::Object * sender, System::EventArgs * e)
{ //the same code as button_click for the case someone clicks first this 1
    if((label11==true || label22==true))
    { //start if
        try{
            sqlDataAdapter1->SelectCommand->CommandText = textBox1->Text;
            dataSet11->Clear();
            if(dataSet11->Tables->get_Item(S"Results"))
            {
                dataSet11->Tables->get_Item(S"Results")->Columns->Clear();
            }
            sqlDataAdapter1->SelectCommand->Connection->set_ConnectionString(S"workstation
id=wiron;packet size=4096;user id=sa;data source=wiron;persist security info=True;initial
catalog=store_data;password=agiamas13");
            sqlDataAdapter1->Fill(dataSet11, S"Results");
            dataGrid1->SetDataBinding( dataSet11, S"Results");
        }catch(Exception *sqlException) {
            MessageBox::Show( sqlException->Message, S"Invalid query", MessageBoxButtons::OK,
MessageBoxIcon::Error);
        } //end catch
        this->saveFileDialog1->ShowDialog();
        dataSet11->WriteXml(String::Concat(S"C:\\Account++\\", this->saveFileDialog1->get_FileName(),
S".xml"));
    } //end if
} //end buttonGO_click

private: System::Void back_click(System::Object * sender, System::EventArgs * e)
{
    Close();
}

};
}

```

Form1.cpp

```
#include "stdafx.h"
#include "Form1.h"
#include <windows.h>

using namespace Account;

int APIENTRY _tWinMain(HINSTANCE hInstance,
    HINSTANCE hPrevInstance,
    LPTSTR lpCmdLine,
    int nCmdShow)
{
    System::Threading::Thread::CurrentThread->ApartmentState = System::Threading::ApartmentState::STA;
    Application::Run(new Form1());
    return 0;
}
```

Form1.h

```
#include "AboutBox.h"
#include "PerInterFace.h"
#include "PerUser1.h"
#include "PerTimePeriod.h"
#include "ExportData.h"
#include "Print.h"
#include "AUser.h"
#include "CUser.h"
#include "RUser.h"

#using <microsoftlib.dll>

//#pragma once

namespace Account
{
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    /// <summary>
    /// Summary for Form1
    ///
    /// WARNING: If you change the name of this class, you will need to change the
    /// 'Resource File Name' property for the managed resource compiler tool
    /// associated with all .resx files this class depends on. Otherwise,
    /// the designers will not be able to interact properly with localized
    /// resources associated with this form.
    /// </summary>
    public __gc class Form1 : public System::Windows::Forms::Form
    {
    public:
        Form1(void)
        {
            InitializeComponent();
        }

    protected:

```

```

void Dispose(Boolean disposing)
{
    if (disposing && components)
    {
        components->Dispose();
    }
    __super::Dispose(disposing);
}

private: System::Windows::Forms::MainMenu * mainMenu1;
private: System::Windows::Forms::MenuItem * menuItem1;
private: System::Windows::Forms::MenuItem * menuItem2;
private: System::Windows::Forms::MenuItem * menuItem3;
private: System::Windows::Forms::MenuItem * menuItem4;

private: System::Windows::Forms::MenuItem * menuItem6;
private: System::Windows::Forms::MenuItem * menuItem7;
private: System::Windows::Forms::MenuItem * menuItem8;
private: System::Windows::Forms::MenuItem * menuItem9;
private: System::Windows::Forms::MenuItem * menuItem10;
private: System::Windows::Forms::MenuItem * menuItem11;
private: System::Windows::Forms::MenuItem * menuItem12;

private: System::Windows::Forms::PrintDialog * printDialog1;
private: System::Windows::Forms::DataGridTextBoxColumn * dataGridTextBoxColumn1;
private: System::Data::SqlClient::SqlDataAdapter * sqlDataAdapter1;
private: System::Data::SqlClient::SqlCommand * sqlSelectCommand1;
private: System::Data::SqlClient::SqlCommand * sqlInsertCommand1;
private: System::Data::SqlClient::SqlConnection * sqlConnection1;
private: Account::DataSet1 * dataSet11;
private: Account::DataSet8 * dataSet81;
private: System::Windows::Forms::PictureBox * pictureBox1;
private: System::Windows::Forms::MenuItem * menuItem13;
private: System::Windows::Forms::MenuItem * menuItem14;
private: System::Windows::Forms::MenuItem * menuItem15;

```

```

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container * components;

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)

```

```

    {
        System::Resources::ResourceManager * resources = new
System::Resources::ResourceManager(__typeof(Account::Form1));
        this->mainMenu1 = new System::Windows::Forms::MainMenu();
        this->menuItem1 = new System::Windows::Forms::MenuItem();
        this->menuItem2 = new System::Windows::Forms::MenuItem();
        this->menuItem3 = new System::Windows::Forms::MenuItem();
        this->menuItem4 = new System::Windows::Forms::MenuItem();
        this->menuItem6 = new System::Windows::Forms::MenuItem();
        this->menuItem7 = new System::Windows::Forms::MenuItem();
        this->menuItem8 = new System::Windows::Forms::MenuItem();
        this->menuItem9 = new System::Windows::Forms::MenuItem();
        this->menuItem13 = new System::Windows::Forms::MenuItem();
        this->menuItem14 = new System::Windows::Forms::MenuItem();
        this->menuItem15 = new System::Windows::Forms::MenuItem();
        this->menuItem10 = new System::Windows::Forms::MenuItem();
        this->menuItem11 = new System::Windows::Forms::MenuItem();
        this->menuItem12 = new System::Windows::Forms::MenuItem();
        this->printDialog1 = new System::Windows::Forms::PrintDialog();
        this->dataGridViewTextBoxColumn1 = new System::Windows::Forms::DataGridViewTextBoxColumn();
        this->sqlDataAdapter1 = new System::Data::SqlClient::SqlDataAdapter();
        this->sqlInsertCommand1 = new System::Data::SqlClient::SqlCommand();
        this->sqlConnection1 = new System::Data::SqlClient::SqlConnection();
        this->sqlSelectCommand1 = new System::Data::SqlClient::SqlCommand();
        this->dataSet81 = new Account::DataSet8();
        this->pictureBox1 = new System::Windows::Forms::PictureBox();
        (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet81))->BeginInit();
        this->SuspendLayout();
        //
        // mainMenu1
        //
        System::Windows::Forms::MenuItem* __mcTemp__1[] = new
System::Windows::Forms::MenuItem*[4];
        __mcTemp__1[0] = this->menuItem1;
        __mcTemp__1[1] = this->menuItem4;
        __mcTemp__1[2] = this->menuItem8;
        __mcTemp__1[3] = this->menuItem10;
        this->mainMenu1->MenuItems->AddRange(__mcTemp__1);
        //
        // menuItem1
        //
        this->menuItem1->Index = 0;
        System::Windows::Forms::MenuItem* __mcTemp__2[] = new
System::Windows::Forms::MenuItem*[2];
        __mcTemp__2[0] = this->menuItem2;
        __mcTemp__2[1] = this->menuItem3;
        this->menuItem1->MenuItems->AddRange(__mcTemp__2);
        this->menuItem1->Text = S"&File";
        this->menuItem1->Click += new System::EventHandler(this, menuItem1_Click);
        //
        // menuItem2
        //
        this->menuItem2->Index = 0;
        this->menuItem2->Text = S"&Print";
        this->menuItem2->Click += new System::EventHandler(this, menuItem2_Click);
        //
        // menuItem3
        //
        this->menuItem3->Index = 1;
        this->menuItem3->Text = S"e&Xit";
        this->menuItem3->Click += new System::EventHandler(this, menuItem3_Click);
        //
    }

```



```

// menuItem4
//
this->menuItem4->Index = 1;
System::Windows::Forms::MenuItem* __mcTemp__3[] = new
System::Windows::Forms::MenuItem*[2];
__mcTemp__3[0] = this->menuItem6;
__mcTemp__3[1] = this->menuItem7;
this->menuItem4->MenuItems->AddRange(__mcTemp__3);
this->menuItem4->Text = S"&View";
//
// menuItem6
//
this->menuItem6->Index = 0;
this->menuItem6->Text = S"p&Er time period";
this->menuItem6->Click += new System::EventHandler(this, menuItem6_Click);
//
// menuItem7
//
this->menuItem7->Index = 1;
this->menuItem7->Text = S"pe&R user";
this->menuItem7->Click += new System::EventHandler(this, menuItem7_Click);
//
// menuItem8
//
this->menuItem8->Index = 2;
System::Windows::Forms::MenuItem* __mcTemp__4[] = new
System::Windows::Forms::MenuItem*[4];
__mcTemp__4[0] = this->menuItem9;
__mcTemp__4[1] = this->menuItem13;
__mcTemp__4[2] = this->menuItem14;
__mcTemp__4[3] = this->menuItem15;
this->menuItem8->MenuItems->AddRange(__mcTemp__4);
this->menuItem8->Text = S"&Tools";
//
// menuItem9
//
this->menuItem9->Index = 0;
this->menuItem9->Text = S"&Export data";
this->menuItem9->Click += new System::EventHandler(this, menuItem9_Click);
//
// menuItem13
//
this->menuItem13->Index = 1;
this->menuItem13->Text = S"&Add user";
this->menuItem13->Click += new System::EventHandler(this, menuItem13_click);
//
// menuItem14
//
this->menuItem14->Index = 2;
this->menuItem14->Text = S"&Change user";
this->menuItem14->Click += new System::EventHandler(this, menuItem14_click);
//
// menuItem15
//
this->menuItem15->Index = 3;
this->menuItem15->Text = S"&Remove User";
this->menuItem15->Click += new System::EventHandler(this, menuItem15_click);
//
// menuItem10
//
this->menuItem10->Index = 3;

```

```

        System::Windows::Forms::MenuItem* __mcTemp__5[] = new
System::Windows::Forms::MenuItem*[2];
        __mcTemp__5[0] = this->menuItem11;
        __mcTemp__5[1] = this->menuItem12;
        this->menuItem10->MenuItems->AddRange(__mcTemp__5);
        this->menuItem10->Text = S"&Help";
        //
        // menuItem11
        //
        this->menuItem11->Index = 0;
        this->menuItem11->Text = S"&Contents";
        this->menuItem11->Click += new System::EventHandler(this, menuItem11_Click);
        //
        // menuItem12
        //
        this->menuItem12->Index = 1;
        this->menuItem12->Text = S"&About Account++";
        this->menuItem12->Click += new System::EventHandler(this, help_ab);
        //
        // printDialog1
        //
        this->printDialog1->ShowHelp = true;
        //
        // dataGridTextBoxColumn1
        //
        this->dataGridTextBoxColumn1->Format = S"";
        this->dataGridTextBoxColumn1->FormatInfo = 0;
        this->dataGridTextBoxColumn1->MappingName = S"";
        this->dataGridTextBoxColumn1->Width = 75;
        //
        // sqlDataAdapter1
        //
        this->sqlDataAdapter1->InsertCommand = this->sqlInsertCommand1;
        this->sqlDataAdapter1->SelectCommand = this->sqlSelectCommand1;
System::Data::Common::DataTableMapping* __mcTemp__6[] = new
System::Data::Common::DataTableMapping*[1];
System::Data::Common::DataColumnMapping* __mcTemp__7[] = new
System::Data::Common::DataColumnMapping*[1];
        __mcTemp__7[0] = new System::Data::Common::DataColumnMapping(S"attribute_value",
S"attribute_value");
        __mcTemp__6[0] = new System::Data::Common::DataTableMapping(S"Table", S"event_data",
__mcTemp__7);
        this->sqlDataAdapter1->TableMappings->AddRange(__mcTemp__6);
        //
        // sqlInsertCommand1
        //
        this->sqlInsertCommand1->CommandText = S"INSERT INTO event_data(attribute_value) VALUES
(@attribute_value)";
        this->sqlInsertCommand1->Connection = this->sqlConnection1;
        this->sqlInsertCommand1->Parameters->Add(new
System::Data::SqlClient::SqlParameter(S"@attribute_value", System::Data::SqlDbType::NVarChar, 1024, S"attribute_value"));
        //
        // sqlConnection1
        //
        this->sqlConnection1->ConnectionString = S"workstation id=WIRON;packet size=4096;user
id=sa;data source=WIRON;persist securi
        S"ty info=False;initial catalog=store_data";
        //
        // sqlSelectCommand1
        //
        this->sqlSelectCommand1->CommandText = S"SELECT event_data.attribute_value FROM
event_data INNER JOIN event_main ON event_"

```

```

        S"data.event_id = event_main.event_id";
        this->sqlSelectCommand1->Connection = this->sqlConnection1;
        //
        // dataSet81
        //
        this->dataSet81->DataSetName = S"DataSet8";
        this->dataSet81->Locale = new System::Globalization::CultureInfo(S"en-GB");
        this->dataSet81->Disposed += new System::EventHandler(this, menuItem5_Click);
        //
        // pictureBox1
        //
        this->pictureBox1->BackColor = System::Drawing::SystemColors::ActiveCaption;
        this->pictureBox1->Image = (__try_cast<System::Drawing::Image * >(resources-
>GetObject(S"pictureBox1.Image")));
        this->pictureBox1->Location = System::Drawing::Point(96, 64);
        this->pictureBox1->Name = S"pictureBox1";
        this->pictureBox1->Size = System::Drawing::Size(96, 88);
        this->pictureBox1->TabIndex = 2;
        this->pictureBox1->TabStop = false;
        //
        // Form1
        //
        this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
        this->ClientSize = System::Drawing::Size(292, 228);
        this->Controls->Add(this->pictureBox1);
        this->Font = new System::Drawing::Font(S"Times New Roman", 8.25F,
System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point, (System::Byte)161);
        this->HelpButton = true;
        this->Icon = (__try_cast<System::Drawing::Icon * >(resources->GetObject(S"$this.Icon")));
        this->Menu = this->mainMenu1;
        this->Name = S"Form1";
        this->Opacity = 0.97;
        this->Text = S"Account++ version 0.92";
        (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet81))->EndInit();
        this->ResumeLayout(false);
    }

```

```

private: System::Void menuItem3_Click(System::Object * sender, System::EventArgs * e)
    {
        exit(0);
    }

```

```

private: System::Void menuItem1_Click(System::Object * sender, System::EventArgs * e)
    {
    }

```

```

private: System::Void help_ab(System::Object * sender, System::EventArgs * e)
    {
        AboutBox* box = new AboutBox();
        box->ShowDialog();
    }

```

```

private: System::Void listView1_SelectedIndexChanged(System::Object * sender, System::EventArgs * e)
    {
    }

```

```

private: System::Void label1_Click(System::Object * sender, System::EventArgs * e)

```

```

        {
        }

private: System::Void menuItem5_Click(System::Object * sender, System::EventArgs * e)
    {PerInterFace* pif = new PerInterFace();
    pif->ShowDialog();
    }

private: System::Void menuItem7_Click(System::Object * sender, System::EventArgs * e)
    {PerUser1* puu = new PerUser1();
    puu->ShowDialog();
    }

private: System::Void menuItem6_Click(System::Object * sender, System::EventArgs * e)
    {PerTimePeriod* ptp = new PerTimePeriod();
    ptp->ShowDialog();
    }

private: System::Void menuItem2_Click(System::Object * sender, System::EventArgs * e)
    { //print click
    Print* prt = new Print();
    prt->ShowDialog();
    }

private: System::Void menuItem11_Click(System::Object * sender, System::EventArgs * e)
    { // Display the index for the help file.
    try {
        Help::ShowHelpIndex(this, "AccountHelp.CHM"/*helpfile*/);
    } catch(Exception *e){ Console::WriteLine(e);}
    }

private: System::Void dataGrid1_Navigate(System::Object * sender, System::Windows::Forms::NavigateEventArgs * ne)
    {
    }

private: System::Void gkjkjbkbjkbjk(System::Object * sender, System::Data::SqlClient::SqlInfoMessageEventArgs * e)
    {
    }

private: System::Void menuItem9_Click(System::Object * sender, System::EventArgs * e)
    {ExportData* exd = new ExportData();
    exd->ShowDialog();
    }

private: System::Void menuItem15_click(System::Object * sender, System::EventArgs * e)
    {RUser* rusr = new RUser();
    rusr->ShowDialog();
    }

private: System::Void menuItem14_click(System::Object * sender, System::EventArgs * e)
    {CUser* cusr = new CUser();
    cusr->ShowDialog();
    }

private: System::Void menuItem13_click(System::Object * sender, System::EventArgs * e)
    {AUser* ausr = new AUser();
    ausr->ShowDialog();
    }

};
}

```


Help_about.cpp

```
#include "StdAfx.h"
#include "help_about.h"
```

Help_about.h

```
#pragma once
```

```
using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
```

```
namespace Account
```

```
{
    /// <summary>
    /// Summary for help_about
    ///
    /// WARNING: If you change the name of this class, you will need to change the
    /// 'Resource File Name' property for the managed resource compiler tool
    /// associated with all .resx files this class depends on. Otherwise,
    /// the designers will not be able to interact properly with localized
    /// resources associated with this form.
    /// </summary>
    public __gc class help_about : public System::Windows::Forms::Form
    {
    public:
        help_about(void)
        {
            InitializeComponent();
        }

    protected:
        void Dispose(Boolean disposing)
        {
            if (disposing && components)
            {
                components->Dispose();
            }
            __super::Dispose(disposing);
        }

    private: System::Windows::Forms::Button * button1;
    private: System::Windows::Forms::DateTimePicker * dateTimePicker1;
    private: System::Windows::Forms::Label * label1;
    private: System::Windows::Forms::Label * label2;

    private:
        /// <summary>
        /// Required designer variable.
        /// </summary>
        System::ComponentModel::Container* components;

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        void InitializeComponent(void)
        {
            this->button1 = new System::Windows::Forms::Button();
```

```

this->dateTimePicker1 = new System::Windows::Forms::DateTimePicker();
this->label1 = new System::Windows::Forms::Label();
this->label2 = new System::Windows::Forms::Label();
this->SuspendLayout();
//
// button1
//
this->button1->Location = System::Drawing::Point(192, 216);
this->button1->Name = S"button1";
this->button1->Size = System::Drawing::Size(80, 32);
this->button1->TabIndex = 0;
this->button1->Text = S"OK";
this->button1->Click += new System::EventHandler(this, button1_Click);
//
// dateTimePicker1
//
this->dateTimePicker1->Location = System::Drawing::Point(48, 40);
this->dateTimePicker1->Name = S"dateTimePicker1";
this->dateTimePicker1->Size = System::Drawing::Size(168, 20);
this->dateTimePicker1->TabIndex = 1;
//
// label1
//
this->label1->Location = System::Drawing::Point(16, 72);
this->label1->Name = S"label1";
this->label1->Size = System::Drawing::Size(224, 40);
this->label1->TabIndex = 2;
this->label1->Text = S"Account++ application designed as part of diploma thesis project . Designed
and i"
                S"mplemented by Alexandros Giamas";
this->label1->Click += new System::EventHandler(this, label1_Click);
//
// label2
//
this->label2->Location = System::Drawing::Point(16, 128);
this->label2->Name = S"label2";
this->label2->Size = System::Drawing::Size(216, 48);
this->label2->TabIndex = 3;
this->label2->Text = S"Contact info +306945102539 email:agiamas@gmail.com";
//
// help_about
//
this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
this->ClientSize = System::Drawing::Size(292, 266);
this->Controls->Add(this->label2);
this->Controls->Add(this->label1);
this->Controls->Add(this->dateTimePicker1);
this->Controls->Add(this->button1);
this->Name = S"help_about";
this->Text = S"help_about";
this->Load += new System::EventHandler(this, help_about_Load);
this->ResumeLayout(false);
    }
private: System::Void help_about_Load(System::Object * sender, System::EventArgs * e)
    {
    }

private: System::Void label1_Click(System::Object * sender, System::EventArgs * e)
    {Application::Exit();
    }
}

```

```
private: System::Void button1_Click(System::Object * sender, System::EventArgs * e)
    {Close();
    }
};
}
```


PerTimePeriod.cpp

```
#include "StdAfx.h"  
#include "PerTimePeriod.h"
```

PerTimePeriod.h

```
#pragma once  
#include "PerTimePeriodResults1.h"
```

```
using namespace System;  
using namespace System::ComponentModel;  
using namespace System::Collections;  
using namespace System::Windows::Forms;  
using namespace System::Data;  
using namespace System::Drawing;
```

```
namespace Account
```

```
{  
    /// <summary>  
    /// Summary for PerTimePeriod  
    ///  
    /// WARNING: If you change the name of this class, you will need to change the  
    ///     'Resource File Name' property for the managed resource compiler tool  
    ///     associated with all .resx files this class depends on. Otherwise,  
    ///     the designers will not be able to interact properly with localized  
    ///     resources associated with this form.  
    /// </summary>  
    public __gc class PerTimePeriod : public System::Windows::Forms::Form  
    {  
        bool ga,gb,gc,gd,ge ;//used to check if I have set the values of datetimepicker 1,2 AND listbox 1,2  
        private: System::Windows::Forms::Button * button2;  
        public:  
            PerTimePeriod(void)  
            {  
                InitializeComponent();  
                System::String * dtPickerFrom;  
                System::String * lb2ValueMember;  
                dtPickerFrom = this->dateTimePicker1->ToString();  
                lb2ValueMember = this->listBox2->get_ValueMember();  
            }  
        protected:  
            void Dispose(Boolean disposing)  
            {  
                if (disposing && components)  
                {  
                    components->Dispose();  
                }  
                __super::Dispose(disposing);  
            }  
        private: System::Windows::Forms::DateTimePicker * dateTimePicker1;  
        private: System::Windows::Forms::DateTimePicker * dateTimePicker3;  
        private: System::Windows::Forms::Label * label1;  
        private: System::Windows::Forms::Label * label2;  
        private: System::Windows::Forms::ListBox * listBox1;  
        private: System::Windows::Forms::ListBox * listBox2;
```

```

private: System::Windows::Forms::Label * label3;
private: System::Windows::Forms::Label * label4;
private: System::Windows::Forms::Button * button1;
private: System::ComponentModel::IContainer * components;

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        System::Configuration::AppSettingsReader * configurationAppSettings = new
System::Configuration::AppSettingsReader();
        this->dateTimePicker1 = new System::Windows::Forms::DateTimePicker();
        this->dateTimePicker3 = new System::Windows::Forms::DateTimePicker();
        this->label1 = new System::Windows::Forms::Label();
        this->label2 = new System::Windows::Forms::Label();
        this->listBox1 = new System::Windows::Forms::ListBox();
        this->listBox2 = new System::Windows::Forms::ListBox();
        this->label3 = new System::Windows::Forms::Label();
        this->label4 = new System::Windows::Forms::Label();
        this->button1 = new System::Windows::Forms::Button();
        this->button2 = new System::Windows::Forms::Button();
        this->SuspendLayout();
        //
        // dateTimePicker1
        //
        this->dateTimePicker1->CustomFormat = (__try_cast<System::String * >(configurationAppSettings-
>GetValue(S"yyyy-MM-dd", __typeof(System::String))));
        this->dateTimePicker1->Format = System::Windows::Forms::DateTimePickerFormat::Custom;
        this->dateTimePicker1->Location = System::Drawing::Point(72, 48);
        this->dateTimePicker1->Name = S"dateTimePicker1";
        this->dateTimePicker1->Size = System::Drawing::Size(136, 20);
        this->dateTimePicker1->TabIndex = 0;
        this->dateTimePicker1->Value = (*__try_cast<__box System::DateTime *
>(configurationAppSettings->GetValue(S"dateTimePicker1.Value", __typeof(System::DateTime))));
        this->dateTimePicker1->ValueChanged += new System::EventHandler(this, timer1_Tick);
        //
        // dateTimePicker3
        //
        this->dateTimePicker3->CustomFormat = (__try_cast<System::String * >(configurationAppSettings-
>GetValue(S"yyyy-MM-dd", __typeof(System::String))));
        this->dateTimePicker3->Format = System::Windows::Forms::DateTimePickerFormat::Custom;
        this->dateTimePicker3->Location = System::Drawing::Point(232, 48);
        this->dateTimePicker3->Name = S"dateTimePicker3";
        this->dateTimePicker3->Size = System::Drawing::Size(136, 20);
        this->dateTimePicker3->TabIndex = 3;
        this->dateTimePicker3->Value = (*__try_cast<__box System::DateTime *
>(configurationAppSettings->GetValue(S"dateTimePicker3.Value", __typeof(System::DateTime))));
        this->dateTimePicker3->ValueChanged += new System::EventHandler(this, timer2_Tick);
        //
        // label1
        //
        this->label1->Location = System::Drawing::Point(72, 32);
        this->label1->Name = S"label1";
        this->label1->Size = System::Drawing::Size(136, 16);

```

```

this->label1->TabIndex = 4;
this->label1->Text = S"starting date";
this->label1->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
//
// label2
//
this->label2->Location = System::Drawing::Point(232, 32);
this->label2->Name = S"label2";
this->label2->Size = System::Drawing::Size(136, 16);
this->label2->TabIndex = 5;
this->label2->Text = S"ending date";
this->label2->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
//
// listBox1
//
this->listBox1->AccessibleName = S"";
this->listBox1->DisplayMember = S"listBox1.ValueMember";
System::Object* __mcTemp__1[] = new System::Object*[24];
__mcTemp__1[0] = S"00:00";
__mcTemp__1[1] = S"01:00";
__mcTemp__1[2] = S"02:00";
__mcTemp__1[3] = S"03:00";
__mcTemp__1[4] = S"04:00";
__mcTemp__1[5] = S"05:00";
__mcTemp__1[6] = S"06:00";
__mcTemp__1[7] = S"07:00";
__mcTemp__1[8] = S"08:00";
__mcTemp__1[9] = S"09:00";
__mcTemp__1[10] = S"10:00";
__mcTemp__1[11] = S"11:00";
__mcTemp__1[12] = S"12:00";
__mcTemp__1[13] = S"13:00";
__mcTemp__1[14] = S"14:00";
__mcTemp__1[15] = S"15:00";
__mcTemp__1[16] = S"16:00";
__mcTemp__1[17] = S"17:00";
__mcTemp__1[18] = S"18:00";
__mcTemp__1[19] = S"19:00";
__mcTemp__1[20] = S"20:00";
__mcTemp__1[21] = S"21:00";
__mcTemp__1[22] = S"22:00";
__mcTemp__1[23] = S"23:00";
this->listBox1->Items->AddRange(__mcTemp__1);
this->listBox1->Location = System::Drawing::Point(72, 120);
this->listBox1->Name = S"listBox1";
this->listBox1->Size = System::Drawing::Size(136, 108);
this->listBox1->TabIndex = 6;
this->listBox1->ValueMember = (__try_cast<System::String * >(configurationAppSettings-
>GetValue(S"listBox1.ValueMember", __typeof(System::String))));
this->listBox1->DoubleClick += new System::EventHandler(this, timer3_Tick);
this->listBox1->SelectedValueChanged += new System::EventHandler(this, timer3_Tick);
this->listBox1->ValueMemberChanged += new System::EventHandler(this, timer3_Tick);
//
// listBox2
//
this->listBox2->DisplayMember = S"listBox2.ValueMember";
System::Object* __mcTemp__2[] = new System::Object*[24];
__mcTemp__2[0] = S"00:00";
__mcTemp__2[1] = S"01:00";
__mcTemp__2[2] = S"02:00";
__mcTemp__2[3] = S"03:00";
__mcTemp__2[4] = S"04:00";

```

```

__mcTemp__2[5] = S"05:00";
__mcTemp__2[6] = S"06:00";
__mcTemp__2[7] = S"07:00";
__mcTemp__2[8] = S"08:00";
__mcTemp__2[9] = S"09:00";
__mcTemp__2[10] = S"10:00";
__mcTemp__2[11] = S"11:00";
__mcTemp__2[12] = S"12:00";
__mcTemp__2[13] = S"13:00";
__mcTemp__2[14] = S"14:00";
__mcTemp__2[15] = S"15:00";
__mcTemp__2[16] = S"16:00";
__mcTemp__2[17] = S"17:00";
__mcTemp__2[18] = S"18:00";
__mcTemp__2[19] = S"19:00";
__mcTemp__2[20] = S"20:00";
__mcTemp__2[21] = S"21:00";
__mcTemp__2[22] = S"22:00";
__mcTemp__2[23] = S"23:00";
this->listBox2->Items->AddRange(__mcTemp__2);
this->listBox2->Location = System::Drawing::Point(240, 120);
this->listBox2->Name = S"listBox2";
this->listBox2->Size = System::Drawing::Size(136, 108);
this->listBox2->TabIndex = 7;
this->listBox2->ValueMember = (__try_cast<System::String * >(configurationAppSettings-
>GetValue(S"listBox2.ValueMember", __typeof(System::String))););
this->listBox2->DoubleClick += new System::EventHandler(this, timer4_Tick);
this->listBox2->ValueMemberChanged += new System::EventHandler(this, timer4_Tick);
//
// label3
//
this->label3->Location = System::Drawing::Point(240, 104);
this->label3->Name = S"label3";
this->label3->Size = System::Drawing::Size(136, 16);
this->label3->TabIndex = 8;
this->label3->Text = S"ending time";
this->label3->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
//
// label4
//
this->label4->Location = System::Drawing::Point(72, 104);
this->label4->Name = S"label4";
this->label4->Size = System::Drawing::Size(136, 16);
this->label4->TabIndex = 9;
this->label4->Text = S"starting time";
this->label4->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
//
// button1
//
this->button1->Location = System::Drawing::Point(208, 264);
this->button1->Name = S"button1";
this->button1->TabIndex = 10;
this->button1->Text = S"GO!";
this->button1->Click += new System::EventHandler(this, timer5_Tick);
//
// button2
//
this->button2->Location = System::Drawing::Point(0, 0);
this->button2->Name = S"button2";
this->button2->Size = System::Drawing::Size(80, 23);
this->button2->TabIndex = 11;
this->button2->Text = S"back";

```

```

        this->button2->Click += new System::EventHandler(this, back_click);
        //
        // PerTimePeriod
        //
        this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
        this->ClientSize = System::Drawing::Size(400, 301);
        this->Controls->Add(this->button2);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->label4);
        this->Controls->Add(this->label3);
        this->Controls->Add(this->listBox2);
        this->Controls->Add(this->label2);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->dateTimePicker3);
        this->Controls->Add(this->dateTimePicker1);
        this->Controls->Add(this->listBox1);
        this->Name = S"PerTimePeriod";
        this->Text = S"PerTimePeriod";
        this->ResumeLayout(false);
    }
private: System::Void timer1_Tick(System::Object * sender, System::EventArgs * e)
    {
        ga=true;
    }

private: System::Void timer2_Tick(System::Object * sender, System::EventArgs * e)
    {
        gb=true;
    }

private: System::Void timer3_Tick(System::Object * sender, System::EventArgs * e)
    {
        gc=true;
    }

private: System::Void timer4_Tick(System::Object * sender, System::EventArgs * e)
    {
        gd=true;
    }

private: System::Void timer5_Tick(System::Object * sender, System::EventArgs * e)
    {
    }

private: System::String *dtp1, *dtp3, *dtp11, *dtp33;
    bool CheckForms(int helper){
        // comments: use of global vars to check if i have values in checkboxes...
        if(helper==5) {if((ga&&gb&&gc&&gd)==true) {ge=true;}}
        if((ga&&gb&&gc&&gd&&ge)==true) {
            dtp11 = dateTimePicker1->Value.ToString();
            dtp1 = dtp11->Substring(0,10);
            dtp33 = dateTimePicker3->Value.ToString();
            dtp3 = dtp33->Substring(0,10);

            PerTimePeriodResults1 *ptpr = new PerTimePeriodResults1(listBox1-
>get_SelectedItem()->ToString(), listBox2->get_SelectedItem()->ToString(),
            dtp1, dtp3);
            ptpr->ShowDialog();
            return true;
        } else
        {
            return false;MessageBox::Show(S"return false!");
        }
    }

```

```
        } //end CheckForms  
private: System::Void back_click(System::Object * sender, System::EventArgs * e)  
        {  
            Close();  
        }  
};  
}
```

PerTimePeriodResults.cpp

```
#include "StdAfx.h"
#include "PerTimePeriodResults.h"
```

PerTimePeriodResults.h

```
#pragma once
```

```
using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
```

```
namespace Account
```

```
{
    /// <summary>
    /// Summary for PerTimePeriodResults
    ///
    /// WARNING: If you change the name of this class, you will need to change the
    /// 'Resource File Name' property for the managed resource compiler tool
    /// associated with all .resx files this class depends on. Otherwise,
    /// the designers will not be able to interact properly with localized
    /// resources associated with this form.
    /// </summary>
    public __gc class PerTimePeriodResults : public System::Windows::Forms::Form
    {
        String __gc *lb1,*lb2,*dtp1,*dtp3;
        System::String *a,*b,*c,*d;
        private: System::Windows::Forms::DataGrid * dataGrid1;
        private: System::Data::SqlClient::SqlDataAdapter * sqlDataAdapter1;

        private: System::Data::SqlClient::SqlConnection * sqlConnection1;
        private: Account::DataSet1 * dataSet11;
        private: System::Data::SqlClient::SqlCommand * sqlSelectCommand1;
        private: System::Data::SqlClient::SqlCommand * sqlInsertCommand1;
        private: System::Data::SqlClient::SqlCommand * sqlUpdateCommand1;
        private: System::Data::SqlClient::SqlCommand * sqlDeleteCommand1;
        private: System::Data::SqlClient::SqlDataAdapter * sqlDataAdapter2;
        private: System::Data::SqlClient::SqlCommand * sqlCommand1;
        public:
            /*void SetPerTimePeriod(String *a, String *b, String *c, String *d) {
                if(a!="") this->lb1 = a; else this->lb1="00:00";
                this->lb2 = b;
                if(c!="") this->dtp1 = c; else this->dtp1="1980/01/01";
                if(d!="") this->dtp3 = d; else this->dtp3="2005/12/12";
                MessageBox::Show(String::Concat(S"nnklnklnkl", dtp1, dtp3));
            }*/

            PerTimePeriodResults(String *a, String *b, String *c,String *d)//edo mporo na antikatastiso ton
            constructor me String *,String *,...,String *)
            {
                InitializeComponent(a,b,c,d);

                try{
                    sqlDataAdapter1->Fill(dataSet11, S"event_data");
                } catch(Exception * ex)
                {
                    Console::WriteLine(ex);
                }
            }
    }
}
```

```

        MessageBox::Show(ex->ToString());
    }

    dataGrid1->SetDataBinding( dataSet11, S"event_data");

}

protected:
void Dispose(Boolean disposing)
{
    if (disposing && components)
    {
        components->Dispose();
    }
    __super::Dispose(disposing);
}

private:
/// <summary>
/// Required designer variable.
/// </summary>
System::ComponentModel::Container* components;

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
void InitializeComponent(String *a , String *b ,String *c , String *d)
{
    if(a!="") this->lb1 = a; else this->lb1="00:00";
    if(b!="") this->lb2 = b; else this->lb2="00:00";
    if(c!="") this->dtp1 = c; else this->dtp1="1980/01/01";
    if(d!="") this->dtp3 = d; else this->dtp3="2005/12/12";
    this->dataGrid1 = new System::Windows::Forms::DataGrid();
    this->dataSet11 = new Account::DataSet1();
    this->sqlDataAdapter1 = new System::Data::SqlClient::SqlDataAdapter();
    this->sqlConnection1 = new System::Data::SqlClient::SqlConnection();
    this->sqlCommand1 = new System::Data::SqlClient::SqlCommand();
    (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataGrid1))->BeginInit();
    (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet11))->BeginInit();
    this->SuspendLayout();
    //
    // dataGrid1
    //
    this->dataGrid1->DataMember = S"";
    this->dataGrid1->DataSource = this->dataSet11;
    this->dataGrid1->HeaderForeColor = System::Drawing::SystemColors::ControlText;
    this->dataGrid1->Location = System::Drawing::Point(8, 64);
    this->dataGrid1->Name = S"dataGrid1";
    this->dataGrid1->Size = System::Drawing::Size(272, 168);
    this->dataGrid1->TabIndex = 0;
    //
    // dataSet11
    //
    this->dataSet11->DataSetName = S"DataSet1";
    this->dataSet11->Locale = new System::Globalization::CultureInfo(S"en-GB");
    //
    // sqlDataAdapter1
    //
    this->sqlDataAdapter1->SelectCommand = this->sqlCommand1;
}

```



```

        System::Data::Common::DataTableMapping* __mcTemp__1[] = new
System::Data::Common::DataTableMapping*[1];
        System::Data::Common::DataColumnMapping* __mcTemp__2[] = new
System::Data::Common::DataColumnMapping*[5];
        __mcTemp__2[0] = new System::Data::Common::DataColumnMapping(S"event_id", S"event_id");
        __mcTemp__2[1] = new System::Data::Common::DataColumnMapping(S"attribute_type",
S"attribute_type");
        __mcTemp__2[2] = new System::Data::Common::DataColumnMapping(S"attribute_value",
S"attribute_value");
        __mcTemp__2[3] = new System::Data::Common::DataColumnMapping(S"Expr1", S"Expr1");
        __mcTemp__2[4] = new System::Data::Common::DataColumnMapping(S"record_timestamp",
S"record_timestamp");
        __mcTemp__1[0] = new System::Data::Common::DataTableMapping(S"Table", S"event_data",
__mcTemp__2);
        this->sqlDataAdapter1->TableMappings->AddRange(__mcTemp__1);
        //
        // sqlConnection1
        //
        this->sqlConnection1->ConnectionString = S"Integrated Security=SSPI;Packet Size=4096;Data
Source=\"TELECOM-NIKIYAM\";Tag with "
        S"column collation when possible=False;Initial Catalog=store_data;Use Procedure fo"
        S"r Prepare=1;Auto Translate=True;Persist Security Info=False;Provider=\"SQLOLEDB.1"
        S"\";Workstation ID=\"TELECOM-NIKIYAM\";Use Encryption for Data=False";
        //
        // sqlCommand1
        //
        this->dtp1 = this->dtp1->Substring(0,10);
        this->dtp3 = this->dtp3->Substring(0,10);
        this->sqlCommand1->CommandText = System::String::Concat(S"SELECT event_data.event_id,
event_data.attribute_type, event_data.attribute_value, event_main.event_id AS Expr1, event_main.record_timestamp FROM
event_data INNER JOIN event_main ON event_data.event_id = event_main.event_id WHERE (event_main.record_timestamp
BETWEEN \"\", dtp1, S" ", lb1, S"\" AND \"\", dtp3, S" ", lb2, S"\"");
        //check this sql command in sql query
        MessageBox::Show(this->lb1);
        MessageBox::Show(this->lb2);

        MessageBox::Show(this->dtp1);
        MessageBox::Show(this->dtp3);

        /\"01/01/1980\" AND \"12/12/2005\"");
        this->sqlCommand1->Connection = this->sqlConnection1;
        //
        // PerTimePeriodResults
        //
        this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
        this->ClientSize = System::Drawing::Size(292, 273);
        this->Controls->Add(this->dataGrid1);
        this->Name = S"PerTimePeriodResults";
        this->Text = S"PerTimePeriodResults";
        (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataGrid1))->EndInit();
        (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet11))->EndInit();
        this->ResumeLayout(false);
    }
};
}
#include "StdAfx.h"
#include "PerTimePeriodResults1.h"

```

```

#pragma once

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;

namespace Account
{
    /// <summary>
    /// Summary for PerTimePeriodResults
    ///
    /// WARNING: If you change the name of this class, you will need to change the
    ///     'Resource File Name' property for the managed resource compiler tool
    ///     associated with all .resx files this class depends on. Otherwise,
    ///     the designers will not be able to interact properly with localized
    ///     resources associated with this form.
    /// </summary>
    public __gc class PerTimePeriodResults1 : public System::Windows::Forms::Form
    {
        private: System::Data::SqlClient::SqlCommand * sqlSelectCommand1;
    public:
        PerTimePeriodResults1(String *a, String *b, String *c, String *d)
        {
            InitializeComponent();
            SelectPtr(a,b,c,d);
        }

        int SelectPtr(String *a, String *b, String *c, String *d)//a=lb1 b=lb2 c=ntp1 d=ntp3
        {
            try{

                try{
                    this->dataSet141->Clear();
                    if(dataSet141->Tables->get_Item(S"Results"))
                    {
                        dataSet141->Tables->get_Item(S"Results")->Columns->Clear();
                    }

                    this->sqlCommand1->set_CommandText(System::String::Concat(S"SELECT event_data.event_id,
event_data.attribute_type, event_data.attribute_value, event_main.event_id AS event_id, event_main.record_timestamp FROM
event_data INNER JOIN event_main ON event_data.event_id = event_main.event_id WHERE (event_main.record_timestamp
BETWEEN \"\",c, S\" \", a, S\"\\ AND \"\", d, S\" \", b, S\"\\)");
                    this->sqlConnection1->Open();
                    this->sqlCommand1->ExecuteReader();
                    this->sqlCommand1->Connection->Close();
                    this->sqlDataAdapter1->Fill(dataSet141, S"Results");
                    this->dataGridView1->SetDataBinding( dataSet141, S"Results");

                }catch(Exception *sqlException) {
                    MessageBox::Show( sqlException->Message, S"Invalid query", MessageBoxButtons::OK,
MessageBoxIcon::Error);
                }
                //end catch

            } catch(Exception *e)
            {
                MessageBox::Show(S"An error has occurred please try later");
                MessageBox::Show(String::Concat(S"error:" ,e->ToString()));
            }
        }
    }
}

```

```

        exit(0);
    }
    MessageBox::Show(S"OK");
return 0;
} //END OF SQLWRITE FUNCTION()

protected:
void Dispose(Boolean disposing)
{
    if (disposing && components)
    {
        components->Dispose();
    }
    __super::Dispose(disposing);
}

private: System::Data::SqlClient::SqlDataAdapter * sqlDataAdapter1;
private: System::Data::SqlClient::SqlConnection * sqlConnection1;
private: System::Data::SqlClient::SqlCommand * sqlCommand1;
private: System::Windows::Forms::DataGrid * dataGrid1;
private: System::Windows::Forms::Label * label1;
private: System::Data::SqlClient::SqlCommand * sqlInsertCommand1;
private: Account::DataSet14 * dataSet141;

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container* components;

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        System::Configuration::AppSettingsReader * configurationAppSettings = new
System::Configuration::AppSettingsReader();
        this->sqlDataAdapter1 = new System::Data::SqlClient::SqlDataAdapter();
        this->sqlCommand1 = new System::Data::SqlClient::SqlCommand();
        this->sqlConnection1 = new System::Data::SqlClient::SqlConnection();
        this->dataGrid1 = new System::Windows::Forms::DataGrid();
        this->dataSet141 = new Account::DataSet14();
        this->label1 = new System::Windows::Forms::Label();
        this->sqlInsertCommand1 = new System::Data::SqlClient::SqlCommand();
        (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataGrid1))->BeginInit();
        (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet141))->BeginInit();
        this->SuspendLayout();
        //
        // sqlDataAdapter1
        //
        this->sqlDataAdapter1->SelectCommand = this->sqlCommand1;
        System::Data::Common::DataTableMapping* __mcTemp__1[] = new
System::Data::Common::DataTableMapping*[1];
        System::Data::Common::DataColumnMapping* __mcTemp__2[] = new
System::Data::Common::DataColumnMapping*[2];
        __mcTemp__2[0] = new System::Data::Common::DataColumnMapping(S"event_id", S"event_id");
        __mcTemp__2[1] = new System::Data::Common::DataColumnMapping(S"record_timestamp",
S"record_timestamp");
        __mcTemp__1[0] = new System::Data::Common::DataTableMapping(S"Table", S"event_data",
__mcTemp__2);
        this->sqlDataAdapter1->TableMappings->AddRange(__mcTemp__1);

```

```

//
// sqlCommand1
//
this->sqlCommand1->CommandText = (__try_cast<System::String * >(configurationAppSettings-
>GetValue(S"sqlCommand1.CommandText", __typeof(System::String)))));
this->sqlCommand1->Connection = this->sqlConnection1;
//
// sqlConnection1
//
this->sqlConnection1->ConnectionString = S"workstation id=\\"TELECOM-NIKIYAM\";packet
size=4096;integrated security=SSPI;data source=\\\"TELECOM-NIKIYAM\";persist security info=False;initial catalog=store_data";
//
// dataGrid1
//
this->dataGrid1->DataMember = S"";
this->dataGrid1->HeaderForeColor = System::Drawing::SystemColors::ControlText;
this->dataGrid1->Location = System::Drawing::Point(40, 80);
this->dataGrid1->Name = S"dataGrid1";
this->dataGrid1->Size = System::Drawing::Size(208, 144);
this->dataGrid1->TabIndex = 0;
//
// dataSet141
//
this->dataSet141->DataSetName = S"DataSet14";
this->dataSet141->Locale = new System::Globalization::CultureInfo(S"en-US");
//
// label1
//
this->label1->Font = new System::Drawing::Font(S"Microsoft Sans Serif", 9,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point, (System::Byte)0);
this->label1->Location = System::Drawing::Point(24, 40);
this->label1->Name = S"label1";
this->label1->Size = System::Drawing::Size(248, 32);
this->label1->TabIndex = 1;
this->label1->Text = S"Search Results:";
this->label1->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
//
// sqlInsertCommand1
//
this->sqlInsertCommand1->CommandText = S"INSERT INTO event_data(event_id, attribute_type,
attribute_value) VALUES (@event_
S" id, @attribute_type, @attribute_value); SELECT event_id, attribute_type, attribu
S" te_value FROM event_data";
this->sqlInsertCommand1->Connection = this->sqlConnection1;
this->sqlInsertCommand1->Parameters->Add(new
System::Data::SqlClient::SqlParameter(S"@event_id", System::Data::SqlDbType::UniqueIdentifier, 16, S"event_id"));
this->sqlInsertCommand1->Parameters->Add(new
System::Data::SqlClient::SqlParameter(S"@attribute_type", System::Data::SqlDbType::NVarChar, 64, S"attribute_type"));
this->sqlInsertCommand1->Parameters->Add(new
System::Data::SqlClient::SqlParameter(S"@attribute_value", System::Data::SqlDbType::NVarChar, 1024, S"attribute_value"));
//
// PerTimePeriodResults1
//
this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
this->ClientSize = System::Drawing::Size(292, 273);
this->Controls->Add(this->label1);
this->Controls->Add(this->dataGrid1);
this->Name = S"PerTimePeriodResults1";
this->Text = S"PerTimePeriodResults";
(__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataGrid1))->EndInit();
(__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet141))->EndInit();

```

```
        this->ResumeLayout(false);  
    };  
}
```

PerUser.cpp

```
#include "StdAfx.h"  
#include "PerUser.h"
```

PerUser.h

```
#pragma once
```

```
using namespace System;  
using namespace System::ComponentModel;  
using namespace System::Collections;  
using namespace System::Windows::Forms;  
using namespace System::Data;  
using namespace System::Drawing;
```

```
namespace Account
```

```
{  
    /// <summary>  
    /// Summary for PerUser  
    ///  
    /// WARNING: If you change the name of this class, you will need to change the  
    ///     'Resource File Name' property for the managed resource compiler tool  
    ///     associated with all .resx files this class depends on. Otherwise,  
    ///     the designers will not be able to interact properly with localized  
    ///     resources associated with this form.  
    /// </summary>  
    public __gc class PerUser : public System::Windows::Forms::Form  
    {  
    public:  
        PerUser(void)  
        {  
            InitializeComponent();  
        }  
  
    protected:  
        void Dispose(Boolean disposing)  
        {  
            if (disposing && components)  
            {  
                components->Dispose();  
            }  
            __super::Dispose(disposing);  
        }  
  
    private: System::Windows::Forms::ListBox * listBox1;  
    private: System::Windows::Forms::Label * label1;  
    private: System::Windows::Forms::Button * button1;  
  
    private:  
        /// <summary>  
        /// Required designer variable.  
        /// </summary>  
        System::ComponentModel::Container* components;  
  
        /// <summary>  
        /// Required method for Designer support - do not modify  
        /// the contents of this method with the code editor.  
        /// </summary>  
        void InitializeComponent(void)  
        {
```

```

        System::Configuration::AppSettingsReader * configurationAppSettings = new
System::Configuration::AppSettingsReader();
        this->listBox1 = new System::Windows::Forms::ListBox();
        this->label1 = new System::Windows::Forms::Label();
        this->button1 = new System::Windows::Forms::Button();
        this->SuspendLayout();
        //
        // listBox1
        //
        System::Object* __mcTemp__1[] = new System::Object*[3];
        __mcTemp__1[0] = S"sdsdnkldnsnklds";
        __mcTemp__1[1] = S"snldslnsdldnkl";
        __mcTemp__1[2] = S"lwenkwenlkewnkwenk";
        this->listBox1->Items->AddRange(__mcTemp__1);
        this->listBox1->Location = System::Drawing::Point(56, 72);
        this->listBox1->Name = S"listBox1";
        this->listBox1->Size = System::Drawing::Size(192, 30);
        this->listBox1->TabIndex = 0;
        this->listBox1->SelectedIndexChanged += new System::EventHandler(this,
listBox1_SelectedIndexChanged);
        //
        // label1
        //
        this->label1->Location = System::Drawing::Point(56, 32);
        this->label1->Name = S"label1";
        this->label1->Size = System::Drawing::Size(192, 32);
        this->label1->TabIndex = 1;
        this->label1->Text = (__try_cast<System::String * >(configurationAppSettings-
>GetValue(S"label1.Text", __typeof(System::String))));
        this->label1->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
        this->label1->Click += new System::EventHandler(this, label1_Click);
        //
        // button1
        //
        this->button1->Location = System::Drawing::Point(104, 112);
        this->button1->Name = S"button1";
        this->button1->Size = System::Drawing::Size(104, 32);
        this->button1->TabIndex = 2;
        this->button1->Text = S"apply";
        this->button1->Click += new System::EventHandler(this, button1_Click);
        //
        // PerUser
        //
        this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
        this->ClientSize = System::Drawing::Size(292, 266);
        this->Controls->Add(this->button1);
        this->Controls->Add(this->label1);
        this->Controls->Add(this->listBox1);
        this->Name = S"PerUser";
        this->Text = S"PerUser";
        this->Load += new System::EventHandler(this, PerUser_Load);
        this->ResumeLayout(false);
    }
private: System::Void listBox1_SelectedIndexChanged(System::Object * sender, System::EventArgs * e)
    {
    }

private: System::Void label1_Click(System::Object * sender, System::EventArgs * e)
    {
    }

```

```
private: System::Void PerUser_Load(System::Object * sender, System::EventArgs * e)
        {
        }

private: System::Void button1_Click(System::Object * sender, System::EventArgs * e)
        {
        }

};

}
```


PerUser1.cpp

```
#include "StdAfx.h"
#include "PerUser1.h"
```

PerUser1.h

```
///  
// #pragma once  
#include "Sessions.h"  
using namespace System;  
using namespace System::ComponentModel;  
using namespace System::Collections;  
using namespace System::Windows::Forms;  
using namespace System::Data;  
using namespace System::Drawing;  
  
namespace Account  
{  
    /// <summary>  
    /// Summary for PerUser1  
    ///  
    /// WARNING: If you change the name of this class, you will need to change the  
    ///     'Resource File N' property for the managed resource compiler tool  
    ///     associated with all .resx files this class depends on. Otherwise,  
    ///     the designers will not be able to interact properly with localized  
    ///     resources associated with this form.  
    /// </summary>  
    public __gc class PerUser1 : public System::Windows::Forms::Form  
    {  
    public: int Sessionss(String * username)  
        {  
            return 0;  
        }  
  
    private: System::Diagnostics::PerformanceCounter * performanceCounter1;  
    private: Account::DataSet16 * dataSet161;  
            private: System::Data::SqlClient::SqlDataAdapter * sqlDataAdapter2;  
                    private: System::Data::SqlClient::SqlCommand * sqlCommand2;  
    private: System::Windows::Forms::Button * button3;  
  
    private: Account::DataSet17 * dataSet171;  
    private: System::Data::OleDb::OleDbCommand * oleDbSelectCommand1;  
    private: System::Data::OleDb::OleDbCommand * oleDbInsertCommand1;  
    private: System::Data::OleDb::OleDbCommand * oleDbUpdateCommand1;  
    private: System::Data::OleDb::OleDbCommand * oleDbDeleteCommand1;  
    private: System::Data::OleDb::OleDbDataAdapter * oleDbDataAdapter1;  
  
    public:  
        System::String *listbox2value;  
  
        PerUser1(void)  
        {  
            InitializeComponent();  
            this->sqlDataAdapter1->Fill(dataSet152, S"attribute_value");  
            this->listBox2->DataSource = this->dataSet152->Tables->get_Item(S"attribute_value");  
            this->listBox2->SelectedIndex = 0;  
            this->listBox2->DisplayMember = S"attribute_value";  
            this->listBox2->ValueMember = S"attribute_value";  
            this->listBox2->Update();  
            this->listBox2->Show();  
        }  
  
    protected:  
}
```

```

void Dispose(Boolean disposing)
{
    if (disposing && components)
    {
        components->Dispose();
    }
    __super::Dispose(disposing);
}
private: System::Windows::Forms::ListBox * listBox1;

private: System::Windows::Forms::Label * label1;
public: System::Windows::Forms::Button * button1;
private: System::Windows::Forms::Button * button2;
private: System::Windows::Forms::DataGrid * dataGrid1;
private: System::Windows::Forms::ListBox * listBox2;
private: System::Data::SqlClient::SqlDataAdapter * sqlDataAdapter1;
private: System::Data::SqlClient::SqlCommand * sqlSelectCommand1;
private: System::Data::SqlClient::SqlConnection * sqlConnection1;
private: Account::DataSet15 * dataSet151;
private: Account::DataSet15 * dataSet152;

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container* components;

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        System::Configuration::AppSettingsReader * configurationAppSettings = new
System::Configuration::AppSettingsReader();
        this->label1 = new System::Windows::Forms::Label();
        this->button2 = new System::Windows::Forms::Button();
        this->dataGrid1 = new System::Windows::Forms::DataGrid();
        this->dataSet152 = new Account::DataSet15();
        this->listBox2 = new System::Windows::Forms::ListBox();
        this->sqlDataAdapter1 = new System::Data::SqlClient::SqlDataAdapter();
        this->sqlSelectCommand1 = new System::Data::SqlClient::SqlCommand();
        this->sqlConnection1 = new System::Data::SqlClient::SqlConnection();
        this->dataSet161 = new Account::DataSet16();
        this->button3 = new System::Windows::Forms::Button();
        this->dataSet171 = new Account::DataSet17();
        this->oleDbSelectCommand1 = new System::Data::OleDb::OleDbCommand();
        this->oleDbInsertCommand1 = new System::Data::OleDb::OleDbCommand();
        this->oleDbUpdateCommand1 = new System::Data::OleDb::OleDbCommand();
        this->oleDbDeleteCommand1 = new System::Data::OleDb::OleDbCommand();
        this->oleDbDataAdapter1 = new System::Data::OleDb::OleDbDataAdapter();
        (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataGrid1))->BeginInit();
        (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet152))->BeginInit();
        (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet161))->BeginInit();
        (__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet171))->BeginInit();
        this->SuspendLayout();
        //
        // label1
        //
        this->label1->Location = System::Drawing::Point(168, 16);
        this->label1->Name = S"label1";
        this->label1->Size = System::Drawing::Size(192, 32);
    }
}

```

```

        this->label1->TabIndex = 1;
        this->label1->Text = (__try_cast<System::String * >(configurationAppSettings-
>GetValue(S"label1.Text", __typeof(System::String)))));
        this->label1->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
        this->label1->Click += new System::EventHandler(this, label1_Click);
        //
        // button2
        //
        this->button2->Location = System::Drawing::Point(216, 160);
        this->button2->Name = S"button2";
        this->button2->TabIndex = 5;
        this->button2->Text = S"GO!";
        this->button2->Click += new System::EventHandler(this, button2_click);
        //
        // dataGrid1
        //
        this->dataGrid1->AlternatingBackColor = System::Drawing::Color::LightGray;
        this->dataGrid1->BackColor = System::Drawing::Color::DarkGray;
        this->dataGrid1->CaptionBackColor = System::Drawing::Color::White;
        this->dataGrid1->CaptionFont = new System::Drawing::Font(S"Verdana", 10);
        this->dataGrid1->CaptionForeColor = System::Drawing::Color::Navy;
        this->dataGrid1->DataMember = S"";
        this->dataGrid1->DataSource = this->dataSet152->event_data;
        this->dataGrid1->ForeColor = System::Drawing::Color::Black;
        this->dataGrid1->GridLineColor = System::Drawing::Color::Black;
        this->dataGrid1->GridLineStyle = System::Windows::Forms::DataGridLineStyle::None;
        this->dataGrid1->HeaderBackColor = System::Drawing::Color::Silver;
        this->dataGrid1->HeaderForeColor = System::Drawing::Color::Black;
        this->dataGrid1->LinkColor = System::Drawing::Color::Navy;
        this->dataGrid1->Location = System::Drawing::Point(112, 192);
        this->dataGrid1->Name = S"dataGrid1";
        this->dataGrid1->ParentRowsBackColor = System::Drawing::Color::White;
        this->dataGrid1->ParentRowsForeColor = System::Drawing::Color::Black;
        this->dataGrid1->SelectionBackColor = System::Drawing::Color::Navy;
        this->dataGrid1->SelectionForeColor = System::Drawing::Color::White;
        this->dataGrid1->Size = System::Drawing::Size(272, 240);
        this->dataGrid1->TabIndex = 9;
        //
        // dataSet152
        //
        this->dataSet152->DataSetName = S"DataSet15";
        this->dataSet152->Locale = new System::Globalization::CultureInfo(S"en-US");
        //
        // listBox2
        //
        this->listBox2->AccessibleDescription = S"listbox2";
        this->listBox2->AccessibleName = S"listbox2";
        this->listBox2->Location = System::Drawing::Point(80, 56);
        this->listBox2->Name = S"listBox2";
        this->listBox2->Size = System::Drawing::Size(344, 95);
        this->listBox2->TabIndex = 8;
        this->listBox2->Tag = S"";
        this->listBox2->ValueMember = (__try_cast<System::String * >(configurationAppSettings-
>GetValue(S"listBox2.ValueMember", __typeof(System::String)))));
        this->listBox2->SelectedIndexChanged += new System::EventHandler(this,
listBox2_SelectedIndexChanged);
        //
        // sqlDataAdapter1
        //
        this->sqlDataAdapter1->SelectCommand = this->sqlSelectCommand1;
        System::Data::Common::DataTableMapping* __mcTemp__1[] = new
System::Data::Common::DataTableMapping*[1];

```

```

System::Data::Common::DataColumnMapping* __mcTemp__2[] = new
System::Data::Common::DataColumnMapping*[1];
__mcTemp__2[0] = new System::Data::Common::DataColumnMapping(S"attribute_value",
S"attribute_value");
__mcTemp__1[0] = new System::Data::Common::DataTableMapping(S"Table", S"event_data",
__mcTemp__2);
this->sqlDataAdapter1->TableMappings->AddRange(__mcTemp__1);
//
// sqlSelectCommand1
//
this->sqlSelectCommand1->CommandText = S"SELECT DISTINCT attribute_value FROM
event_data WHERE (attribute_type = \User-Nam
S"e\");
this->sqlSelectCommand1->Connection = this->sqlConnection1;
//
// sqlConnection1
//
this->sqlConnection1->ConnectionString = S"workstation id=\TELECOM-NIKIYAM\;packet
size=4096;integrated security=SSPI;data s
S"ource=\TELECOM-NIKIYAM\;persist security info=False;initial catalog=store_data";
//
// dataSet161
//
this->dataSet161->DataSetName = S"DataSet16";
this->dataSet161->Locale = new System::Globalization::CultureInfo(S"en-US");
//
// button3
//
this->button3->Location = System::Drawing::Point(0, 0);
this->button3->Name = S"button3";
this->button3->Size = System::Drawing::Size(80, 23);
this->button3->TabIndex = 10;
this->button3->Text = S"back";
this->button3->Click += new System::EventHandler(this, back_click);
//
// dataSet171
//
this->dataSet171->DataSetName = S"DataSet17";
this->dataSet171->Locale = new System::Globalization::CultureInfo(S"en-US");
//
// oleDbDataAdapter1
//
this->oleDbDataAdapter1->DeleteCommand = this->oleDbDeleteCommand1;
this->oleDbDataAdapter1->InsertCommand = this->oleDbInsertCommand1;
this->oleDbDataAdapter1->SelectCommand = this->oleDbSelectCommand1;
this->oleDbDataAdapter1->UpdateCommand = this->oleDbUpdateCommand1;
//
// PerUser1
//
this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
this->ClientSize = System::Drawing::Size(440, 525);
this->Controls->Add(this->button3);
this->Controls->Add(this->dataGrid1);
this->Controls->Add(this->listBox2);
this->Controls->Add(this->button2);
this->Controls->Add(this->label1);
this->Name = S"PerUser1";
this->Text = S"PerUser1";
this->Load += new System::EventHandler(this, PerUser1_Load);
(__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataGrid1))->EndInit();
(__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet152))->EndInit();
(__try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet161))->EndInit();

```

```

        (_try_cast<System::ComponentModel::ISupportInitialize * >(this->dataSet171))->EndInit();
        this->ResumeLayout(false);
    }
private: System::Void listBox1_SelectedIndexChanged(System::Object * sender, System::EventArgs * e)
    {
        this->listbox1->Update();
    }
private: System::Void label1_Click(System::Object * sender, System::EventArgs * e)
    {
    }
private: System::Void PerUser1_Load(System::Object * sender, System::EventArgs * e)
    {
    }
private: System::Void button1_Click(System::Object * sender, System::EventArgs * e)
    {
    }
private: System::Void sqlConnection1_InfoMessage(System::Object * sender,
System::Data::SqlClient::SqlInfoMessageEventArgs * e)
    {
    }
private: System::Void sqlDataAdapter1_RowUpdated(System::Object * sender,
System::Data::SqlClient::SqlRowUpdatedEventArgs * e)
    {
    }
private: System::Void sqlConnection1_InfoMessage_1(System::Object * sender,
System::Data::SqlClient::SqlInfoMessageEventArgs * e)
    {
    }
private: System::Void sqlConnection1_InfoMessage_2(System::Object * sender,
System::Data::SqlClient::SqlInfoMessageEventArgs * e)
    {
    }
private: System::Void sqlDataAdapter2_RowUpdated(System::Object * sender,
System::Data::SqlClient::SqlRowUpdatedEventArgs * e)
    {
    }
private: System::Void sqlDataAdapter1_RowUpdated_1(System::Object * sender,
System::Data::SqlClient::SqlRowUpdatedEventArgs * e)
    {
    }
private: System::Void dataGrid1_Navigate(System::Object * sender, System::Windows::Forms::NavigateEventArgs * ne)
    {
    }
private: System::Void sqlDataAdapter1_RowUpdated_2(System::Object * sender,
System::Data::SqlClient::SqlRowUpdatedEventArgs * e)
    {
    }

```

```

private: System::Void textBox1_TextChanged(System::Object * sender, System::EventArgs * e)
    {
    }

private: System::Void dataGridView1_Navigate_1(System::Object * sender, System::Windows::Forms::NavigateEventArgs * ne)
    {
    }

private: System::Void listBox2_SelectedIndexChanged(System::Object * sender, System::EventArgs * e)
    { this->listbox2value = this->listBox2->SelectedValue->ToString();
    }

private: System::Void sqlConnection1_InfoMessage_3(System::Object * sender,
System::Data::SqlClient::SqlInfoMessageEventArgs * e)
    {
    }

private: System::Void dataGridView1_ListChanged(System::Object * sender, System::ComponentModel::ListChangedEventArgs * e)
    {
    }

private: System::Void button2_click(System::Object * sender, System::EventArgs * e)
    {
        this->sqlDataAdapter2 = new System::Data::SqlClient::SqlDataAdapter();
        this->sqlSelectCommand2 = new System::Data::SqlClient::SqlCommand();
        this->sqlDataAdapter2->SelectCommand = this->sqlSelectCommand2;
        this->sqlSelectCommand2->CommandText =
System::String::Concat("SELECT event_id, attribute_type, attribute_value FROM event_data WHERE event_id IN(select
DISTINCT event_id FROM event_data WHERE attribute_value= \'",this->listbox2value ,"\') ORDER by event_id");
        this->sqlSelectCommand2->Connection = this->sqlConnection1;
        System::Data::Common::DataTableMapping* __mcTemp__1[] = new
System::Data::Common::DataTableMapping*[1];
        System::Data::Common::DataColumnMapping* __mcTemp__2[] = new
System::Data::Common::DataColumnMapping*[2];
        __mcTemp__2[0] = new System::Data::Common::DataColumnMapping("attribute_type",
System::String::Concat("attribute_type"));
        __mcTemp__2[1] = new System::Data::Common::DataColumnMapping("attribute_value",
System::String::Concat("attribute_value"));
        __mcTemp__1[0] = new System::Data::Common::DataTableMapping("Table", "event_data",
__mcTemp__2);
        this->sqlDataAdapter2->TableMappings->AddRange(__mcTemp__1);
        sqlDataAdapter2->Fill( dataSet161, "attribute_value");
        dataGridView1->SetDataBinding( dataSet161, "attribute_value");
        this->dataGridView1->Update();
        Sessions * popupsessions = new Sessions(listbox2value);
        popupsessions->ShowDialog();

        //listbox 2 value...if its not the default then run the sql query
    }

private: System::Void back_click(System::Object * sender, System::EventArgs * e)
    {Close();
    }

};

}

```


Print.cpp

```
#include "StdAfx.h"  
#include "Print.h"
```

Print.h

```
#pragma once
```

```
using namespace System;  
using namespace System::ComponentModel;  
using namespace System::Collections;  
using namespace System::Windows::Forms;  
using namespace System::Data;  
using namespace System::Drawing;  
using namespace System::Xml;  
using namespace System::Drawing::Printing;  
using namespace System::IO;  
using namespace System::ComponentModel;  
using namespace System::Drawing::Drawing2D;  
using namespace System::Drawing::Design ;  
using namespace System::Drawing::Text;  
using namespace System::Drawing::Imaging ;
```

```
namespace Account
```

```
{  
    /// <summary>  
    /// Summary for Print  
    ///  
    /// WARNING: If you change the name of this class, you will need to change the  
    ///     'Resource File Name' property for the managed resource compiler tool  
    ///     associated with all .resx files this class depends on. Otherwise,  
    ///     the designers will not be able to interact properly with localized  
    ///     resources associated with this form.  
    /// </summary>  
    public __gc class Print : public System::Windows::Forms::Form  
    {  
  
        bool xmlfile, printoverview;  
        private: System::Windows::Forms::PrintDialog * printDialog1;  
        private: System::Drawing::Printing::PrintDocument * printDocument1;  
        private: System::Windows::Forms::PrintPreviewDialog * printPreviewDialog2;  
  
        private: System::Windows::Forms::OpenFileDialog * openFileDialog1;  
        private: System::Windows::Forms::Button * button2;  
        private: System::Windows::Forms::PrintPreviewDialog * printPreviewDialog1;  
        private: System::IO::StreamReader* streamToPrint;  
        public:  
  
        Print(void)  
        {  
  
            InitializeComponent();  
  
        }  
  
        protected:
```



```

void Dispose(Boolean disposing)
{
    if (disposing && components)
    {
        components->Dispose();
    }
    __super::Dispose(disposing);
}
private: System::Windows::Forms::Label * label1;
private: System::Windows::Forms::RadioButton * radioButton1;
private: System::Drawing::Font *printFont;
private: System::Windows::Forms::Button * button1;

private:

/// <summary>
/// Required designer variable.
/// </summary>
System::ComponentModel::Container* components;

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
void InitializeComponent(void)
{
    System::Resources::ResourceManager * resources = new
System::Resources::ResourceManager(__typeof(Account::Print));
    this->label1 = new System::Windows::Forms::Label();
    this->radioButton1 = new System::Windows::Forms::RadioButton();
    this->button1 = new System::Windows::Forms::Button();
    this->printDialog1 = new System::Windows::Forms::PrintDialog();
    this->printDocument1 = new System::Drawing::Printing::PrintDocument();
    this->printPreviewDialog1 = new System::Windows::Forms::PrintPreviewDialog();
    this->printPreviewDialog2 = new System::Windows::Forms::PrintPreviewDialog();
    this->openFileDialog1 = new System::Windows::Forms::OpenFileDialog();
    this->button2 = new System::Windows::Forms::Button();
    this->SuspendLayout();
    //
    // label1
    //
    this->label1->Location = System::Drawing::Point(56, 48);
    this->label1->Name = S"label1";
    this->label1->Size = System::Drawing::Size(80, 40);
    this->label1->TabIndex = 0;
    this->label1->Text = S"Print a custom xml file:";
    //
    // radioButton1
    //
    this->radioButton1->Location = System::Drawing::Point(48, 112);
    this->radioButton1->Name = S"radioButton1";
    this->radioButton1->TabIndex = 1;
    this->radioButton1->Text = S"PrintXmlFile";
    this->radioButton1->Click += new System::EventHandler(this, PrintXmlFile_click);
    //
    // button1
    //
    this->button1->Location = System::Drawing::Point(56, 160);
    this->button1->Name = S"button1";
    this->button1->TabIndex = 3;
}

```

```

        this->button1->Text = S"Print";
        this->button1->Click += new System::EventHandler(this, Print_click);
        //
        // printDialog1
        //
        this->printDialog1->Document = this->printDocument1;
        //
        // printDocument1
        //
        this->printDocument1->PrintPage += new System::Drawing::Printing::PrintPageEventHandler(this,
pd_PrintPage);
        //
        // printPreviewDialog1
        //
        this->printPreviewDialog1->AutoScrollMargin = System::Drawing::Size(0, 0);
        this->printPreviewDialog1->AutoScrollMinSize = System::Drawing::Size(0, 0);
        this->printPreviewDialog1->ClientSize = System::Drawing::Size(400, 300);
        this->printPreviewDialog1->Enabled = true;
        this->printPreviewDialog1->Icon = (__try_cast<System::Drawing::Icon * >(resources-
>GetObject(S"printPreviewDialog1.Icon")));
        this->printPreviewDialog1->Location = System::Drawing::Point(176, 176);
        this->printPreviewDialog1->MinimumSize = System::Drawing::Size(375, 250);
        this->printPreviewDialog1->Name = S"printPreviewDialog1";
        this->printPreviewDialog1->TransparencyKey = System::Drawing::Color::Empty;
        this->printPreviewDialog1->Visible = false;
        //
        // printPreviewDialog2
        //
        this->printPreviewDialog2->AutoScrollMargin = System::Drawing::Size(0, 0);
        this->printPreviewDialog2->AutoScrollMinSize = System::Drawing::Size(0, 0);
        this->printPreviewDialog2->ClientSize = System::Drawing::Size(400, 300);
        this->printPreviewDialog2->Document = this->printDocument1;
        this->printPreviewDialog2->Enabled = true;
        this->printPreviewDialog2->Icon = (__try_cast<System::Drawing::Icon * >(resources-
>GetObject(S"printPreviewDialog2.Icon")));
        this->printPreviewDialog2->Location = System::Drawing::Point(258, 23);
        this->printPreviewDialog2->MinimumSize = System::Drawing::Size(375, 250);
        this->printPreviewDialog2->Name = S"printPreviewDialog2";
        this->printPreviewDialog2->TransparencyKey = System::Drawing::Color::Empty;
        this->printPreviewDialog2->Visible = false;
        //
        // openFileDialog1
        //
        this->openFileDialog1->DefaultExt = S".xml";
        this->openFileDialog1->FileName = S"results";
        this->openFileDialog1->InitialDirectory = S"c:\\Account++";
        //
        // button2
        //
        this->button2->Location = System::Drawing::Point(0, 0);
        this->button2->Name = S"button2";
        this->button2->Size = System::Drawing::Size(80, 23);
        this->button2->TabIndex = 5;
        this->button2->Text = S"back";
        this->button2->Click += new System::EventHandler(this, close_window);
        //
        // Print
        //
        this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
        this->ClientSize = System::Drawing::Size(200, 245);
        this->Controls->Add(this->button2);
        this->Controls->Add(this->button1);

```

```

        this->Controls->Add(this->radioButton1);
        this->Controls->Add(this->label1);
        this->Name = S"Print";
        this->Text = S"Print";
        this->ResumeLayout(false);
    }
private: System::Void PrintXmlFile_click(System::Object * sender, System::EventArgs * e)
    {xmlfile=true;
    }

private: System::Void Print_click(System::Object * sender, System::EventArgs * e)
    {
        if(xmlfile) {
            this->openFileDialog1->ShowDialog();
printDialog1->Document = printDocument1;
            // Show the dialog
            if (printDialog1->ShowDialog() == DialogResult::OK)
            {
                // Add the page handler
                try {
                    streamToPrint = new StreamReader (this->openFileDialog1->get_FileName());
                    printFont = new System::Drawing::Font("Arial", 10);
                    printDocument1->PrintPage += new PrintPageEventHandler(this,
&Print::pd_PrintPage);
                    // Print the page
                    printDocument1->Print();
                }
                catch(Exception *ex) {MessageBox::Show(S"An error occurred printing the file!!");
                    MessageBox::Show(ex->Message->ToString());
                }
            }
        }
    }
} //end printdialog
} //end(ifxmlfile)
} //end private

//Event fired for each page to print
private: void pd_PrintPage(System::Object * sender, System::Drawing::Printing::PrintPageEventArgs * ev)
    {
        float lpp = 0 ;
        float yPos = 0 ;
        int count = 0 ;
        float leftMargin = ev->MarginBounds.Left;
        float topMargin = ev->MarginBounds.Top;
        String* line=S" ";
        //Work out the number of lines per page
        //Use the MarginBounds on the event to do this
        lpp = (ev->MarginBounds.Height / printFont->GetHeight(ev->Graphics)) ;
        //Now iterate over the file printing out each line
        //NOTE WELL: This assumes that a single line is not wider than the page width
        //Check count first so that we don't read line that we won't print
        while (count < lpp && ((line=streamToPrint->ReadLine()) != S""))
        {
            yPos = topMargin + (count * printFont->GetHeight(ev->Graphics));
            ev->Graphics->DrawString (line, printFont, Brushes::Black, leftMargin, yPos, new
StringFormat());
            count++;
            Console::WriteLine(line); //MessageBox::Show(line);
        }
        //If we have more lines then print another page
        if (line->Trim() != S"")
            ev->HasMorePages = true ;
        else{

```

```
                ev->HasMorePages = false ;
                streamToPrint->Close();
            }
//end customise
    }

private: System::Void close_window(System::Object * sender, System::EventArgs * e)
    {
        Close();
    }

};

};
```

Ruser.cpp

```
//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by app.rc
//
#define IDI_ICON1          101

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        102
#define _APS_NEXT_COMMAND_VALUE        40001
#define _APS_NEXT_CONTROL_VALUE        1000
#define _APS_NEXT_SYMED_VALUE          101
#endif
#endif
#include "StdAfx.h"
#include "RUser.h"
```

Ruser.h

```
#pragma once

using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
using namespace System::DirectoryServices;

namespace Account
{
    /// <summary>
    /// Summary for RUser
    ///
    /// WARNING: If you change the name of this class, you will need to change the
    /// 'Resource File Name' property for the managed resource compiler tool
    /// associated with all .resx files this class depends on. Otherwise,
    /// the designers will not be able to interact properly with localized
    /// resources associated with this form.
    /// </summary>
    public __gc class RUser : public System::Windows::Forms::Form
    {
        System::String *UserNameToDelete;
    public:
        int ADDelete(String *a){
            try
            { //Build the path, and then bind to computer.
                String *adsPath = String::Format(S"WinNT://{0},computer", Environment::MachineName);
                DirectoryEntry *computerEntry = new DirectoryEntry(adsPath);
                DirectoryEntry *userEntry = computerEntry->Children->Find(a, S"user");
                computerEntry->Children->Remove(userEntry);
                MessageBox::Show(S"removed user");
            }
            catch(Exception *e)
            {
                MessageBox::Show(S"error:!");
                Console::Write(e);
            }
        }
    };
}
```

```

}

    return 0;
} //na to allakso se deleteuser

    RUser(void)
    {
        InitializeComponent();
    }

protected:
    void Dispose(Boolean disposing)
    {
        if (disposing && components)
        {
            components->Dispose();
        }
        __super::Dispose(disposing);
    }
private: System::Windows::Forms::Label * label1;
private: System::Windows::Forms::TextBox * textBox1;
private: System::Windows::Forms::Button * button1;
private: System::Windows::Forms::Button * button2;

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container* components;

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        this->label1 = new System::Windows::Forms::Label();
        this->textBox1 = new System::Windows::Forms::TextBox();
        this->button1 = new System::Windows::Forms::Button();
        this->button2 = new System::Windows::Forms::Button();
        this->SuspendLayout();
        //
        // label1
        //
        this->label1->Font = new System::Drawing::Font(S"Microsoft Sans Serif", 10,
System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point, (System::Byte)0);
        this->label1->Location = System::Drawing::Point(72, 16);
        this->label1->Name = S"label1";
        this->label1->Size = System::Drawing::Size(136, 32);
        this->label1->TabIndex = 0;
        this->label1->Text = S"Enter username to delete";
        this->label1->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;
        //
        // textBox1
        //
        this->textBox1->Location = System::Drawing::Point(88, 64);
        this->textBox1->Name = S"textBox1";
        this->textBox1->TabIndex = 1;
        this->textBox1->Text = S"alexgiamas";
        this->textBox1->TextChanged += new System::EventHandler(this, textBox1_TextChanged);
        //

```

```

// button1
//
this->button1->Location = System::Drawing::Point(96, 96);
this->button1->Name = S"button1";
this->button1->Size = System::Drawing::Size(80, 23);
this->button1->TabIndex = 2;
this->button1->Text = S"Remove him!";
this->button1->Click += new System::EventHandler(this, button1_click);
//
// button2
//
this->button2->Location = System::Drawing::Point(0, 0);
this->button2->Name = S"button2";
this->button2->Size = System::Drawing::Size(80, 23);
this->button2->TabIndex = 7;
this->button2->Text = S"back";
this->button2->Click += new System::EventHandler(this, back_click);
//
// RUser
//
this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
this->ClientSize = System::Drawing::Size(256, 141);
this->Controls->Add(this->button2);
this->Controls->Add(this->button1);
this->Controls->Add(this->textBox1);
this->Controls->Add(this->label1);
this->Name = S"RUser";
this->Text = S"Remove a user from the system";
this->ResumeLayout(false);
}
private: System::Void button1_click(System::Object * sender, System::EventArgs * e)
{
    ADDelete(this->textBox1->Text->ToString());
}

private: System::Void textBox1_TextChanged(System::Object * sender, System::EventArgs * e)
{
}

private: System::Void back_click(System::Object * sender, System::EventArgs * e)
{
    Close();
}

};
}

```

Sessions.cpp

```
#include "StdAfx.h"
#include "Sessions.h"
```

Sessions.h

```
#pragma once
```

```
using namespace System;
using namespace System::ComponentModel;
using namespace System::Collections;
using namespace System::Windows::Forms;
using namespace System::Data;
using namespace System::Drawing;
using namespace System::Data;
using namespace System::Data::SqlClient;
```

```
namespace Account
```

```
{
    /// <summary>
    /// Summary for Sessions
    ///
    /// WARNING: If you change the name of this class, you will need to change the
    ///     'Resource File Name' property for the managed resource compiler tool
    ///     associated with all .resx files this class depends on. Otherwise,
    ///     the designers will not be able to interact properly with localized
    ///     resources associated with this form.
    /// </summary>
    public __gc class Sessions : public System::Windows::Forms::Form
    {
    public: String * SessionTime(DateTime start, DateTime end)
        {
            String * sessiontime = S"09:37";
            System::TimeSpan diff1 = end.Subtract(start);
            sessiontime = diff1.ToString();
            return sessiontime;
        }
    public: Sessions(String * username)
        {
            MessageBox::Show(username);
            InitializeComponent();

            this->sqlCommand1->CommandText = System::String::Concat(S"select record_timestamp from event_data inner join
event_main on event_data.event_id=event_main.event_id where event_Data.event_id IN (select event_data.event_id from
event_data inner join event_main on event_data.event_id=event_main.event_id where attribute_type='Acct-Status-Type' AND
attribute_value='1') AND event_data.attribute_type='User-Name' AND event_data.attribute_value='', username, S"" order by
event_data.event_id");
            this->sqlCommand2->CommandText =System::String::Concat(S"select record_timestamp from event_data inner join event_main
on event_data.event_id=event_main.event_id where event_Data.event_id IN (select event_data.event_id from event_data inner join
event_main on event_data.event_id=event_main.event_id where attribute_type='Acct-Status-Type' AND attribute_value='2') AND
event_data.attribute_type='User-Name' AND event_data.attribute_value='', username , S"" order by event_data.event_id");
            SqlDataReader* rdr1;
            SqlDataReader* rdr2;

        try
        {
            this->sqlConnection1->Open();
            this->sqlConnection2->Open();
            rdr1 = this->sqlCommand1->ExecuteReader();
            MessageBox::Show(rdr1->get_HasRows().ToString());
            rdr2 = this->sqlCommand2->ExecuteReader();
            while(rdr1->Read() && rdr2->Read())

```



```

        {
            this->listBox1->Items->Add(S"login");
            this->listBox1->Items->Add(rdr1->get_Item("record_timestamp"));
            this->listBox1->Items->Add(S"logout");
            this->listBox1->Items->Add(rdr2->get_Item("record_timestamp"));
            this->listBox1->Items->Add(System::String::Concat(S"Session time:",
SessionTime(Convert::ToDateTime(rdr1->get_Item("record_timestamp")->ToString()), Convert::ToDateTime(rdr2-
>get_Item("record_timestamp")->ToString()  ))));
        }
        this->sqlConnection1->Close();
        this->sqlConnection2->Close();
    }
    catch (Exception* e)
    {
        MessageBox::Show(e->ToString());
        System::Console::WriteLine(e->ToString());
    }

    this->listBox1->Update();
    this->listBox1->Show();
    }

protected:
    void Dispose(Boolean disposing)
    {
        if (disposing && components)
        {
            components->Dispose();
        }
        __super::Dispose(disposing);
    }

private: System::Windows::Forms::ListBox * listBox1;
private: System::Data::SqlClient::SqlConnection * sqlConnection1;
private: System::Data::SqlClient::SqlConnection * sqlConnection2;
private: System::Data::SqlClient::SqlCommand * sqlCommand1;
private: System::Data::SqlClient::SqlCommand * sqlCommand2;

private:
    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container* components;

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        System::Configuration::AppSettingsReader * configurationAppSettings = new
System::Configuration::AppSettingsReader();
        this->listBox1 = new System::Windows::Forms::ListBox();
        this->sqlConnection1 = new System::Data::SqlClient::SqlConnection();
        this->sqlConnection2 = new System::Data::SqlClient::SqlConnection();
        this->sqlCommand1 = new System::Data::SqlClient::SqlCommand();
        this->sqlCommand2 = new System::Data::SqlClient::SqlCommand();
        this->SuspendLayout();
        //
        // listBox1
        //
    }

```

```

        this->listBox1->Location = System::Drawing::Point(16, 64);
        this->listBox1->Name = S"listBox1";
        this->listBox1->Size = System::Drawing::Size(232, 121);
        this->listBox1->TabIndex = 0;
        //
        // sqlConnection1
        //
        this->sqlConnection1->ConnectionString = S"workstation id=\"TELECOM-NIKIYAM\";packet
size=4096;integrated security=SSPI;data s"
            S"ource=\"TELECOM-NIKIYAM\";persist security info=False;initial catalog=store_data";
        //
        // sqlConnection2
        //
        this->sqlConnection2->ConnectionString = S"workstation id=\"TELECOM-NIKIYAM\";packet
size=4096;integrated security=SSPI;data s"
            S"ource=\"TELECOM-NIKIYAM\";persist security info=False;initial catalog=store_data";
        //
        // sqlCommand1
        //
        this->sqlCommand1->CommandText = (__try_cast<System::String * >(configurationAppSettings-
>GetValue(S"sqlCommand1.CommandText", __typeof(System::String))));
        this->sqlCommand1->Connection = this->sqlConnection1;
        //
        // sqlCommand2
        //
        this->sqlCommand2->CommandText = (__try_cast<System::String * >(configurationAppSettings-
>GetValue(S"sqlCommand2.CommandText", __typeof(System::String))));
        this->sqlCommand2->Connection = this->sqlConnection2;
        //
        // Sessions
        //
        this->AutoScaleBaseSize = System::Drawing::Size(5, 13);
        this->ClientSize = System::Drawing::Size(264, 229);
        this->Controls->Add(this->listBox1);
        this->Name = S"Sessions";
        this->Text = S"Sessions";
        this->ResumeLayout(false);
    }

};
}

// stdafx.cpp : source file that includes just the standard includes
// Account++.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"

// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
#pragma once

#define WIN32_LEAN_AND_MEAN // Exclude rarely-used stuff from Windows headers
// C RunTime Header Files
#include <stdlib.h>
#include <malloc.h>
#include <memory.h>
#include <tchar.h>

// TODO: reference additional headers your program requires here

```

```
#include "DataSet1.h"  
#include "DataSet2.h"  
#include "DataSet3.h"  
#include "DataSet4.h"  
#include "DataSet11.h"  
#include "DataSet5.h"  
#include "DataSet6.h"  
#include "DataSet7.h"  
#include "DataSet8.h"  
#include "DataSet9.h"  
#include "DataSet10.h"  
#include "DataSet12.h"  
#include "DataSet13.h"  
#include "DataSet14.h"  
#include "DataSet15.h"  
#include "DataSet16.h"  
#include "DataSet17.h"  
#include "DataSet18.h"
```

Appendix B

Published Research paper

Παρακάτω παρατίθεται ένα research paper το οποίο σχετίζεται με την διπλωματική μου εργασία και στο οποίο είχα την τιμή να συμμετάσχω και να εγκριθεί για δημοσίευση και παρουσίαση στο συνέδριο ICAS'05 και ICNS'05 της IEEE society.

Authentication platform for seamless handover in heterogeneous environments

Dimitris Nikitopoulos, Nikolaos Papaoulakis, Angelos Trakos, Alexandros Giamas,
Efsthathios Sykas and Michael Theologou

National Technical University of Athens, Greece

School of Electrical and Computer Engineering

[dnikit,npapaoul,atrakos]@telecom.ntua.gr, el99190@mail.ntua.gr, [sykas,theolog]@cs.ntua.gr

Abstract

This work tackles the hot issue of user handover from one network to another, or generally from one Radio Access Technology (RAT) to another, in a 4G heterogeneous mobile and wireless environment without the interruption of an ongoing service (seamless handover) with user authentication support. This paper proposes an authenticated seamless handover mechanism and describes the authentication platform for two network environments: an operator with both GPRS and WLAN networks (the user performs a vertical handover). Two different WLAN operators (the user performs a horizontal handover).

authenticated handover mechanism for 4G heterogeneous environments.

The innovation of this work is the user handover from one network to another, or generally from one Radio Access Technology (RAT) to another, in a 4G heterogeneous mobile and wireless environment without the interruption of an ongoing service (seamless handover) with user authentication support. For example, a user with an ongoing FTP session moves to a foreign network without interruption of the service and he is silently authenticated.

The authors propose an authenticated seamless handover mechanism and describe the authentication platform for two network environments: Firstly, the user moves from the GPRS network of an operator to the WLAN network of the same operator (vertical handover). And secondly, the user moves from the WLAN network of an operator to the WLAN network of another operator (horizontal handover).

This paper is focused on authenticated seamless handover in heterogeneous environments and is organized as follows. Section 1 is the introduction to the problem of seamless handover with authentication support. Section 2 presents the heterogeneous environment and its architecture that used in this work. Section 3 presents the Mobile IPv6 implementation of our testbed. In section 4 the GPRS and WLAN common authentication mechanisms are described. The most important section is section 5 which presents the proposed authentication platform which is going to

1 1. Introduction

Nowadays, many radio access technologies and core networks coexist. The future of “Beyond 3G” mobile and wireless systems are the heterogeneous platforms. Maybe the hottest issue in these heterogeneous environments is the user seamless handover by taking into account also the seamless user authentication. This paper proposes a seamless

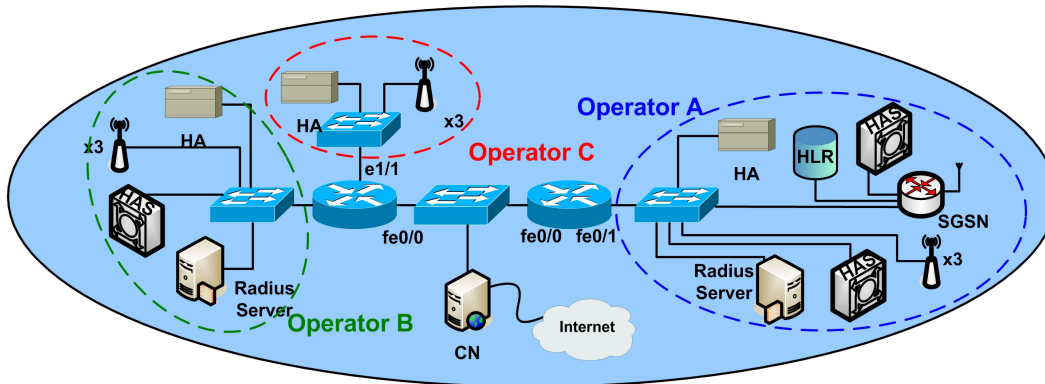


Fig. 1. The WonderLANd testbed schematic: Operator A with WLAN and GPRS support. Operators B and C with WLAN support. Also, Heterogeneous Authentication Servers (HASs), Radius servers and the HLR of the 3 operators are illustrated.

support seamless handover in heterogeneous environments. Finally, section 6 summarizes our work and presents future steps.

2 Heterogeneous platform architecture

The term heterogeneous indicates a multiplatform network that is comprised by several different technology based networks. Thus a heterogeneous environment comprises of GSM, GPRS/EDGE, UMTS, WLAN and other networks. All these type of networks cooperate and offer to users complete and integrated services. Users gain access to all networks and are connected to the preferred network and service anywhere, anytime. So, the quality of service is guaranteed and users have the preferred quality in the appropriate price.

For the design and implementation of the proposed authentication platform, the WonderLANd testbed [1] of the National Technical University of Athens (NTUA) is used. The WonderLANd is a novel 4G wireless wide area network, which covers the NTUA Campus. As it seems in Fig. 1, three different operators have been deployed. Two of them are pure WLAN operators while the third one supports both WLAN and GPRS. Each WLAN operator covers a specific area of the NTUA Campus and all these areas are being overlapped to each other at their borders.

More specific, Operator A consists of four Access Points (APs) which implement a 802.11b network and one SGSN router (emulated) which provides local

GPRS support to the users of the system. The SGSN is directly connected with the HLR (emulated) of Operator A. Operator B and C consists of three different servicing APs each. Even though Operator C has five APs, two of them are used as a wireless backbone for the interconnection of the servicing APs with Operator's C gateway router. All the three WLAN Operators consists of wide area sector antennas in order to long range cover the campus.

The three Operators end up on two Cisco 2611XM multi-service routers which separate them in three different networks and connect them to the internet. The routers act as gateway routers, support and utilize both IPv4 and IPv6 protocol [2], [5]. Each operator has a Home Agent (HA) in order to support Mobile IPv6 user mobility.

The IPv6 enabled GPRS operator is established by the following architecture; due to lack of IPv6 routing support on the live SGSN and in order to have IPv6 addresses and connectivity between MN and core IPv6 network.

A Windows 2003 server is used as the IPv6 SGSN. Analytically, the Windows 2003 Server has two interfaces; the LAN interface that communicates with the core 6bone network of the platform and the GPRS interface which is a GPRS terminal PC card with 4+4

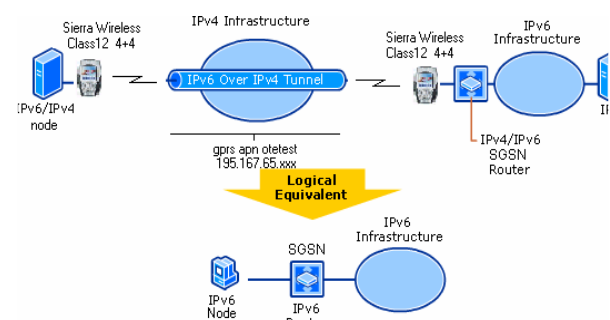


Fig. 2. GPRS IPv6 connectivity.

radio time slots (Class 12) in uplink and downlink. The GPRS terminal on the MN and the GPRS interface of the router are attached to the same GPRS Access Point Network (APN) that has enabled inter-mobile communication feature, in order to have a point to point IPv4 connection. Via this IP connection, a static IPv6 over IPv4 tunnel is established in order to the GPRS terminal of MN gains access to the core IPv6 platform. The LAN interface of the Windows 2003 router belongs to the IPv6 **3ff3::0:1/64** subnet while the **3ffe::0:1:5/80** subnet is created for the GPRS tunneled network (Fig. 2).

Also, the Heterogeneous Authentication Server (HAS) is introduced in this paper and the authors integrate HAS in WonderLANd. HAS is responsible to exchange authentication messages between different networks. HAS has a network type specific part which is communicates with the other network elements depending the network type. In GPRS, HAS communicates with the HA and HLR (see next sections). One HAS per network type is needed.

As described above, the WonderLANd platform implements the IPv6 protocol with mobility support. Mobile IPv6 (MIPv6) is described in the next section.

3 3. Mobile IPv6 implementation

Mobile IPv6 aims to make the mobility transparent to higher level protocols and applications. This is achieved by special routing and signaling mechanisms between the current location of a Mobile Node (MN); the MN's Home Agent (HA) and the Correspondent Nodes (CNs) that MN is communicating with.

Consider that MN has moved from its Home Network to a Foreign Network. MN determines that has moved to a new network, based on the network IPv6 prefixes of router advertisements (RAs) sent by network's IPv6 routers. As the MN moves to a new foreign network, configures a Care-of Address (CoA) and sends a Binding Update (BU) to its HA and also to any CN that it is communicating with. A BU consists of a binding update destination option and also a home address destination option. An IPSec Authentication Header (AH) is used to authenticate the datagram carrying the BU to the HA. This mechanism prevents

unauthorized access to foreign networks, in case of seamless handover need. This operation is illustrated in fig. 3. As the HA of the home network receives successfully the BU, it first sends a proxy neighbor advertisement which redirects the datagrams sent to the home address of the MN to the link layer address of the HA. HA also sends a binding acknowledgement (BA) to the MN.

After that and when a CN that is not aware of the mobility of the MN, sends a datagram to the home address of the MN, it is captured by the HA and tunneled to the CoA of the MN. When the MN receives a datagram tunneled by its HA, it sends a BU to the original sender of the datagram which is the CN to inform it of the new location of the MN. In consequence, CN can send datagrams directly to MN by routing them through the care-of address with a routing header. This operation is illustrated in fig. 4.

Now, when MN communicates with the CN it uses its CoA in the source address of datagrams to avoid the negative effects of triangulation problem. To preserve the higher level connection, home address option is used in all datagrams. In practice, the connections are always established using the home address. The home address is moved to the home address option and CoA is inserted into the source address field of the datagram by the MIPv6 software. When applications in the CN send packets to the MN, they are sent to the home address of the MN. The MIPv6 software then inserts a routing header to the packet and moves the home address to the routing header and inserts the CoA to the destination address field [7].

4 4. Common authentication mechanism

4.1 4.1. Introduction

This paper describes an authentication platform for seamless handover in heterogeneous network environments. The radio access networks that are examined are

GPRS and WLAN. This section describes analytically the common authentication mechanisms used in each network type. These authentication mechanisms are widely used in GPRS and WLAN environments when the user initiates a GPRS attach or a connection to these networks without performing handover from one network to another.

4.2 4.2. GPRS

The user authentication procedures in GPRS are similar to these used in GSM. The difference is that the procedures are executed from the SGSN instead of the MSC. Additionally, the authentication procedure performs the selection of the ciphering algorithm and the synchronization for the ciphering. The authentication mechanism uses “authentication triplets” which are received from the HLR and stored into the SGSN. Authentication triplets consist of 3 parameters (Table 1).

When the user demands access to GPRS network, SGSN must authenticate him. If the SGSN does not have previously stored authentication triplets, sends a Send Authentication Info message to the HLR. Then, HLR responds with a Send Authentication Info Acknowledge message. SGSN has the authentication triplet for that user. When the SGSN has the authentication

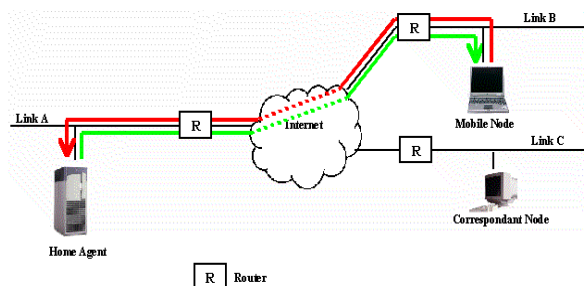


Fig. 3. The Binding Update to Home Agent mechanism; MN sends the BU message and HA sends an acknowledgement.

triplet it sends an Authentication Request message to the mobile subscriber (MS) including the RAND. After that, MS computes SRES from RAND and secret Subscriber authentication key Ki. The MS then sends Authentication Response to the SGSN including SRES and if the computed by the HLR SRES equals to the SRES computed by the MS (fig. 5), the MS is considered to be authenticated to use the network.

It is important to note that all security functions inside the GPRS network are based on the secrecy of the secret key Ki. This key is stored into the SIM (Subscriber Identification Module) card and into the HLR at subscription time and it is not known to the subscriber.

4.3 4.3. WLAN

Authentication in WLAN environments is a complex procedure. Because of the fact that the transport medium is the open air and there is no physical link between the requesting authentication user and the authentication server, the mechanism must ensure security without compromising flexibility and ease-of-use. Authentication using an authentication proxy is a method widely used to meet these two goals.

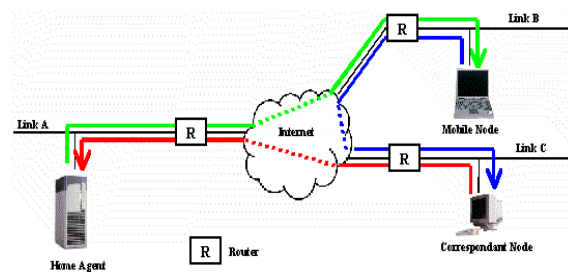


Fig. 4. The Binding Update to Corresponding Node mechanism; CN sends a datagram to the MN's home address, HA captures it and tunnel it to the MN and MN sends a BU to the CN.

TABLE 1
AUTHENTICATION TRIPLETS

Parameter	Description
<i>RAND</i>	random number between 0 and 2128
<i>SRES</i>	signed response which is result of the A3 algorithm used for subscriber authentication
<i>Kc</i>	Ciphering key which computed using the A8 algorithm and it is used by the GPRS Encryption Algorithm (GEA)

Authentication proxy [3] is a solution that provides dynamic per-user authentication and authorization; authenticating users against various industry standard authentication protocols such as TACACS+ and RADIUS. Network administrators have the ability to set an individual, per-user security policy allowing or disallowing services and security restrictions. Using an authentication proxy is a solution that can apply to intranet, extranet, Internet and VPN client users thus creating a uniform platform for authentication and authorization. Finally, one of the major benefits from using an authentication proxy is that it requires no vendor specific client features or software, providing transparent client operation using commonly available desktop browsers. This feature is invaluable for our purpose to provide an authenticated seamless handover between various types of networks.

Part of the WonderLAND platform (fig. 1, WLAN Operator A) is used in order to illustrate the way an authentication proxy functions. This consists of an Access Point (Cisco Aironet), a Cisco 2611XM router running authentication proxy and RADIUS client features, and the Microsoft IAS RADIUS Server. A hypothetical scenario would involve a WLAN user named Joseph who wants to access the Internet while in the

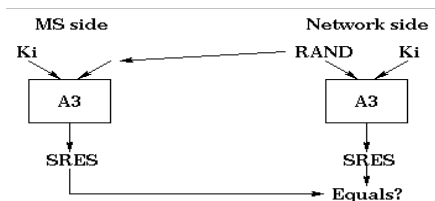


Fig. 5. GPRS SRES computation and comparison.

coverage area of WLAN Operator 1 hotspot. As soon as gets within area of coverage his WLAN card senses the presence of a wireless network and notifies Joseph. Joseph then opens his browser and assuming he has configured no start page, he types in the address bar the site's URL he wants to access. At this point the sequence of events which are visualized in fig. 6 is triggered in a way transparent to Joseph.

First of all, the Mobile Terminal enters the network and as soon as DHCP packets pass through the Authentication Proxy to the router it responds by assigning the Mobile Terminal an IP address from its DHCP pool (step 1). Consequently, the AP forwards Joseph's web page request to authentication proxy (step 2) and receives in return a web page requesting Joseph to authenticate himself using a username/password pair of values (step 3). His username can be either a regular username such as Joseph or a username with a *realm*. A realm is an identifier that represents a company or organization and attaches to a username as a prefix or suffix.

Joseph then enters his credentials and upon submitting them the AP forwards the Access Request to the authentication proxy (step 4). At this stage as the authentication proxy is configured as a RADIUS client to the IAS RADIUS server, it forwards the request to the IAS (step 5) in the form of an Access Request. IAS then gets consulted by its User Account Database, which in our case is an SQL Server, whether the submitted

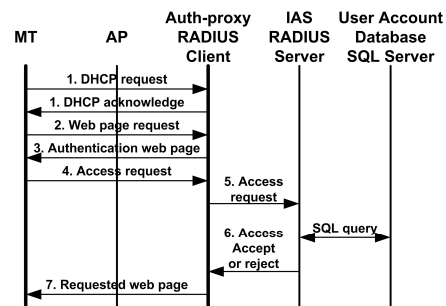


Fig. 6. Authentication proxy mechanism.

username/password pair matches a stored pair and if it matches sends back an Access Accept response. If the pair doesn't match any of the stored pairs in the User Account Database the RADIUS Server returns an Access Reject packet to the authentication proxy asking for denial of authentication (step 6). Depending on the packet received the authentication proxy sends back the appropriate web page to the AP to forward it to the end user (step 7).

5 5. Authenticated seamless handover

5.1 5.1. Introduction

The innovation of this work is the user handover from one network to another, or generally from one Radio Access Technology (RAT) to another, in a 4G heterogeneous mobile and wireless environment without the interruption of an ongoing service (seamless handover) with user authentication support. For example, a user with an ongoing FTP session moves to a foreign network without interruption of the service and he is silent authenticated.

This section proposes an authenticated seamless handover mechanism and describes the authentication platform for two network environments: Firstly, the user moves from the GPRS network of an operator to the WLAN network of the same operator (vertical handover). And secondly, the user moves from the WLAN network of an operator to the WLAN network of another operator (horizontal handover).

5.2 5.2. Proposed mechanism

The proposed heterogeneous authentication mechanism supports seamless handover between heterogeneous 4G mobile and wireless network platforms during non-stop services, like file transfer, video and audio streaming, etc.

The proposed mechanism is based on IPv6 extension header for Mobile IPv6 mobile node authentication. When a mobile node moves from one IPv6 network to another, sends a Binding Update (BU) message to its Home Agent (HA). This message includes an IPsec authentication header [6]. The mobile node generates an authentication key and sends it within the BU message. Then, HA uses this key in order to authenticate the user.

In this paper, a new network component is introduced: the Heterogeneous Authentication Server (HAS). HAS is responsible to communicate with its peer in the user's home network and exchange user authentication and authorization information. In a heterogeneous wireless environment one HAS per network type per operator is needed. HAS is network type specific; in GPRS networks HAS communicates with the HLR and HA, in WLAN environments HAS communicates with the Authentication proxy router, the Radius server and the HA. In GPRS, there is communication between HLR and HAS in order to synchronize the user authentication information. In the HLR an operator defined field is added in order to indicate if a specific user has WLAN access permission.

First of all, the proposed mechanism utilizes HAS which communicates with authentication proxy router in order to get the user information of the new-coming user. Then, HAS communicates with user's home HAS and gets the user permitted services. The

home HAS checks within the network that it is responsible for that user's permissions. Finally, the first HAS authenticates the user. The authentication in WLAN networks is performed with the previously described authentication proxy mechanism.

With this novel authentication mechanism the user is authenticated via the Binding Update IPsec authentication header to the HA. IPsec authentication header (AH) is used to authenticate the datagram carrying the BU. On each WLAN hot spot there is an access router with authentication proxy feature enabled. When the user connects to the WLAN network, takes an IPv6 care of address from the DHCP pool and the prefix advertisement. All the ports of auth-proxy firewall are closed except those for the DHCP, DNS, and the port for the communication between MN and HA. These ports are opened for functional purposes.

The next parts of this section describe the authentication mechanism in term of message exchanged, for two network environments: Firstly, the user moves from the GPRS network of an operator to the WLAN network of the same operator (vertical handover). And secondly, the user moves from the WLAN network of an operator to the WLAN network of another operator (horizontal handover).

5.3 5.3. GPRS – WLAN (Vertical handover)

In this scenario, a user (MN) is moving from his Home Network which is the GPRS Operator A (1) to the foreign WLAN Operator A (1) network. A vertical handover into the same operator (Operator A) is performed. Fig. 7 illustrates all the mechanism and exchanged messages between MN, CN, HAS and network elements. A

precondition is that operator A gives to its subscribers access permission to operator's WLAN hotspots and both HAS have given peer-to-peer authentications.

First of all, the subscriber starts an ongoing data transfer session (e.g. FTP service) and moves from GPRS to WLAN. The WLAN router embeds a DHCP server that gives a care of IPv6 address to the subscriber. Immediately, MN sends a Binding Update message that includes the IPsec authentication header to user's HA and receives the acknowledgment. Then, the HAS of WLAN checks (periodic check) the DHCP pool of network router for new-coming user entries. HAS finds that a new user comes from another network. After that, HAS sends an Heterogeneous Authentication request message to user's home HAS that it (GPRS HAS) checks HA for successful user BU notification and queries the GPRS HLR if the user has WLAN access permission. Home HAS responses to HAS and if the user is an

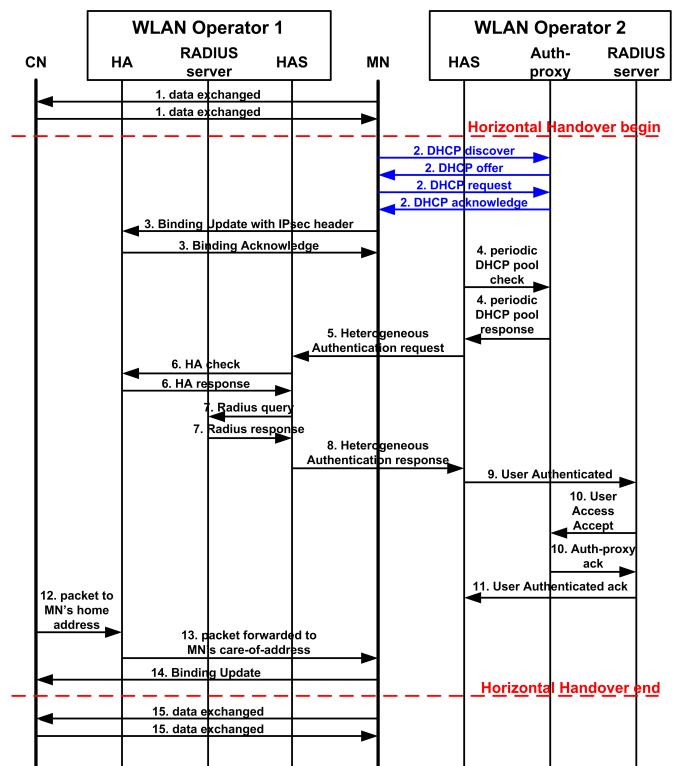


Fig. 8. Message sequence diagram for WLAN Operator A to WLAN Operator B horizontal handover.

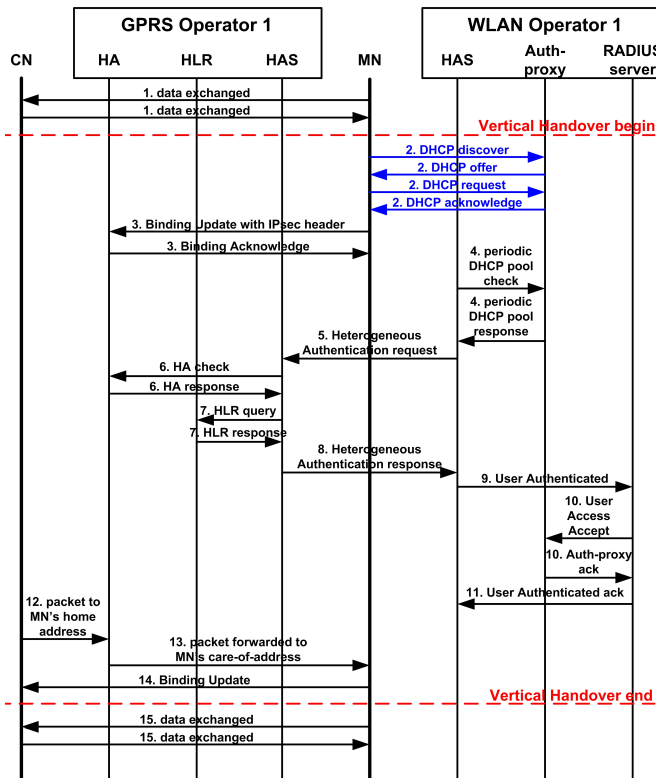


Fig. 7. Message sequence diagram for GPRS to WLAN vertical handover.

authenticated user, HAS notifies Radius server to access grant the user. Radius server sends a User Access Accept message to auth-proxy and Auth-proxy opens the firewall for this user. Finally, acknowledges to Radius server and to HAS are generated.

Possible packets coming from CN to MN's home address are captured by the HA. These captured packets are forwarded by the HA to MN's care of address. MN sends a BU to the CN. The ongoing service is active and it is continuing uninterrupted (FTP session).

5.4 5.4. WLAN – WLAN (Horizontal handover)

In this scenario, a user (MN) is moving from his Home Network which is the WLAN Operator A (1) to the foreign WLAN

Operator B (2) network. A horizontal handover across different operators is performed. Fig. 8 illustrates all the mechanism and exchanged messages between MN, CN, HAS and network elements. A precondition is that the two operators have roaming agreement and gives to the subscribers access permission to both operator's WLAN hotspots and both HAS have gave peer-to-peer authentications.

First of all, the subscriber starts an ongoing data transfer session (e.g. FTP service) and changes operator. The WLAN router embeds a DHCP server that gives a care of IPv6 address to the subscriber. Immediately, MN sends a Binding Update message that includes the IPsec authentication header to user's HA and receives the acknowledgment. Then, the HAS of WLAN checks (periodic check) the DHCP pool of network router for new-coming user entries. HAS finds that a new user comes from another network. After that, HAS sends an Heterogeneous Authentication request message to user's home HAS that it (GPRS HAS) checks HA for successful user BU notification and queries the GPRS HLR if the user has WLAN access permission. Home HAS responses to HAS and if the user is an authenticated user, HAS notifies Radius server to access grant the user. Radius server sends a User Access Accept message to auth-proxy and Auth-proxy opens the firewall for this user. Finally, acknowledges to Radius server and to HAS are generated.

Possible packets coming from CN to MN's home address are captured by the HA. These captured packets are forwarded by the HA to MN's care of address. MN sends a BU to the CN. The ongoing service is active and it is continuing uninterrupted (FTP session).

6 6. Conclusion

Authentication is of prime importance in heterogeneous network environments. This work proposes a novel authentication mechanism in order to achieve this goal. Future work is to implement and validate this platform, to test it in more network and handover types, to measure seamless handover performance and to introduce a heterogeneous billing mechanism.

7 7. Acknowledgement

This work was performed in the framework of European projects, which are funded by the European Commission. The Authors would like to acknowledge their colleagues from the National Technical University of Athens, Greece for their contributions.

8 8. References

- [1] IST-CAUTION++ Deliverable D-5.1, “Trial planning description”, August 2004.
- [2] Regis Desmeules, “Cisco Self-Study: Implementing Cisco IPv6 Networks (IPv6)”, Cisco Press, May 2003.
- [3] Richard A. Deal, “Cisco Router Firewall Security”, Cisco Press, August 10, 2004.
- [4] Greg Bastien, “CCSP SECUR Exam Certification Guide”, Cisco Press, December 22, 2003.

- [5] Mark A. Sportack, “IP Addressing Fundamentals”, Cisco Press, October 31, 2002.
- [6] S. Kent, R. Atkinson, “IP Authentication Header”, RFC 2402, November 1998.
- [7] <http://www.mipl.mediapoli.com/doc>

Appendix C

Βιβλιογραφία

- [1] Visual C++ .NET Bible Wiley; 1 edition (July 15, 2002) ISBN: 0764548379
- [2] Microsoft Visual C++ .Net Deluxe Learning Edition Version 2003 Microsoft Press; Bk edition (June 4, 2003) ISBN: 0735619085
- [3] Visual C++.NET: How to Program Prentice Hall; Bk edition (August 27, 2003) 0134373774
- [4] Microsoft Visual C++ .NET Step by Step--Version 2003 (Step By Step (Microsoft)) Microsoft Press; Bk edition (April 23, 2003) ISBN:0735619077
- [5] Microsoft Visual C++ .NET 2003 Kick Start Sams; 1st edition (October 29, 2003) ISBN: 0672326000
- [6] Visual C++ .NET: A Primer for C++ Developers Peer Information; 1st edition (March, 2002) ISBN: 1861005962
- [7] Programming with Managed Extensions for Microsoft Visual C++ .NET Microsoft Press; 1 edition (July 31, 2002) ISBN: 0735617244
- [8] Microsoft Visual C++ .NET Language Reference Microsoft Press; 1 edition (May 29, 2002) ISBN: 0735615535
- [9] Programming with Microsoft Visual C++ .NET, Sixth Edition (Core Reference) Microsoft Press; 6th Pkg edition (September 25, 2002) ISBN: 0735615497
- [10] Microsoft Visual C++.NET Course Technology; 1 edition (May 24, 2002) ISBN: 0619016574
- [11] CCNA Certification Library (CCNA Self-Study, exam #640-801) Cisco Press; Bk edition (August 14, 2003) ISBN: 1587200953
- [12] Cisco IOS in a Nutshell O'Reilly; 1 edition (December 15, 2001) ISBN: 156592942X
- [13] Cisco Field Manual: Router Configuration Cisco Press; 1st edition (December 14, 2001) ISBN: 1587050242
- [14] Cisco Networking Simplified Cisco Press; 1st edition (June 9, 2003) ISBN: 1587200740
- [15] Mastering Cisco Routers Sybex; 2nd edition (June 15, 2002) ISBN: 0782141072
- [16] Microsoft Windows Internals, Fourth Edition: Microsoft Windows Server(TM) 2003, Windows XP, and Windows 2000 (Pro-Developer) Microsoft Press; 4th edition (December 8, 2004) ISBN: 0735619174
- [17] Active Directory, Second Edition O'Reilly; 2 edition (April 1, 2003) ISBN: 0596004664
- [18] Mastering Active Directory for Windows Server 2003 Sybex Inc; 3 edition (February 20, 2003) ISBN: 0782140793
- [19] Windows Server 2003 Active Directory Design And Implementation: Creating, Migrating, And Merging Networks Packt Publishing (December 30, 2004) ISBN: 1904811086
- [20] Active Directory Bible Wiley; Bk edition (November 20, 2000) ISBN: 0764547623