Πανεπιστήμιο Πειραιώς

Τμήμα Ψηφιακών Συστημάτων

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

Τεχνοοικονομική Διοίκηση και Ασφάλεια Ψηφιακών Συστημάτων

«Κατεύθυνση Ασφάλειας Ψηφιακών Συστημάτων»

Μεταπτυχιακή Διπλωματική Εργασία

Υλοποίηση RBAC, με χρήση του AzMan, σε περιβάλλον Windows

Καλουδάς Βασίλειος {MTE1108}

Επιβλέπων: Αναπληρωτής Καθηγητής, Κωνσταντίνος Λαμπρινουδάκης

Πειραιάς, Ιούλιος 2013

Περιεχόμενα

Ευρετήριο Σχημάτων

Ευρετήριο Πινάκων

Πρόλογος

Η παρούσα Διπλωματική Εργασία με θέμα «Υλοποίηση RBAC, με χρήση του AzMan σε περιβάλλον Windows» εκπονήθηκε στο πλαίσιο του Μεταπτυχιακού Προγράμματος σπουδών «Τεχνοοικονομική Διοίκηση και Ασφάλεια Ψηφιακών Συστημάτων», υπό την επίβλεψη του Αναπληρωτή Καθηγητή κ. Κωνσταντίνου Λαμπρινουδάκη. Από τη θέση αυτή θα ήθελα να ευχαριστήσω, τον κ. Λαμπρινουδάκη για την καθοριστική του συμβολή στην επιλογή του θέματος της εργασίας και γενικότερα για την καθοδήγησή του και την εποικοδομητική συνεργασία σε όλη τη διάρκεια της προσπάθειας της εκπόνησης της παρούσας εργασίας. Επίσης, θα ήθελα να ευχαριστήσω την Εθνική Τράπεζα της Ελλάδος και ειδικότερα τα Στελέχη της Διεύθυνσης Πληροφορικής κυρίους Ν. Χριστοδούλου, Μ. Μαυροφοράκη, Η. Τραβασάρο και Δ. Κακαβέτσο, για τη δυνατότητα που μου έδωσαν κατά το διάστημα από Ιούλιο έως Οκτώβριο 2012 να αντιληφθώ πως σχεδιάζει-υλοποιεί και ελέγχει ένας Οργανισμός τέτοιου βεληνεκούς, τον Έλεγχο Πρόσβασης των χρηστών στα συστήματα. Τέλος, ευχαριστώ θερμά την οικογένεια μου για την οικονομική και ηθική υποστήριξή της, καθ' όλη τη διάρκεια των μεταπτυχιακών μου σπουδών.

Περίληψη

Ο Έλεγχος Πρόσβασης απασχολεί, κάθε επιχείρηση που χρησιμοποιεί πληροφοριακά συστήματα στη παραγωγική της γραμμή. Τα Συστήματα που πραγματοποιούν τον έλεγχο πρόσβασης είναι σχεδιασμένα ώστε να ελέγχουν ποιος μπορεί να εκτελέσει ποιες εργασίες και να ποιοι έχουν πρόσβαση σε ποιους πόρους. Επί της ουσίας, για κάθε Πληροφοριακό Σύστημα ή Εφαρμογή πρέπει να πραγματοποιείται έλεγχος πρόσβασης, που αφορά στα δικαιώματα των υποκειμένων και των αντικειμένων. Η εμπειρία μας έχει διδάξει ότι σε μεγάλους Οργανισμούς, που χρησιμοποιούν πληθώρα συστημάτων και απασχολούν μεγάλο αριθμό υπαλλήλων, η προστασία των πόρων αποτελεί πρωταρχικό στόχο. Το πρόβλημα γίνεται ακόμα πιο σύνθετο, αν προστεθούν οι οργανωσιακές ιεραρχίες και η ύπαρξη περιορισμών όπως οι κανόνες σύγκρουσης συμφερόντων, ή όπως είναι γνωστότεροι οι conflict-of-interest rules. Για την αντιμετώπιση και επίλυση του προβλήματος, είναι απαραίτητος ο σχεδιασμός και η υλοποίηση μεθόδων για τον έλεγχο της πρόσβασης των χρηστών στα χρησιμοποιούμενα συστήματα. Ο σχεδιασμός και η εφαρμογή των ελέγχων πρόσβασης παραμένει μια διαδικασία σύνθετη και ιδιαίτερα σημαντική, για την ασφαλή αλλά και παραγωγική λειτουργία ενός Οργανισμού ή μιας Επιχείρησης.

Με την πάροδο του χρόνου η εξυπηρέτηση των εργασιών όλων των επιχειρήσεων και οργανισμών βασίζεται σχεδόν αποκλειστικά στη χρήση Πληροφοριακών συστημάτων, από ολοένα περισσότερους χρήστες, οι οποίοι έχουν διαφορετικά καθήκοντα, εκτελούν διαφορετικού είδους αλλά και επιπέδου ευθύνης, εργασίες. Για τους προαναφερόμενους λόγους, οι οποίοι μάλιστα θα εντείνονται συνεχώς στο μέλλον, μεγιστοποιούνται οι ανάγκες για την απόλυτα εξουσιοδοτημένη πρόσβαση των χρηστών στα λειτουργούντα Πληροφορικά συστήματα.

Στο χώρο της ασφάλειας των ΤΠΕ έχει αναπτυχθεί ένα επιστημονικό πεδίο, το οποίο καλείται να δώσει λύσεις με το σχεδιασμό και την απονομή προσβάσεων στους χρήστες ανάλογα με τη θέση τους και το είδος της εργασίας που εκτελούν στο πλαίσιο της ομαλής λειτουργίας της Επιχείρησης ή του Οργανισμού, που εργάζονται. Στο πλαίσιο του επιστημονικού αυτού πεδίου εκπονήθηκε η παρούσα ΜΔΕ, η οποία συγγράφηκε μετά από:

- τη μελέτη σχετικής βιβλιογραφίας,
- την καθοδήγηση του επιβλέποντα καθηγητή μου και
- την εμπειρία που αποκτήθηκε κατά την παρουσία μου στη Διεύθυνση Πληροφορικής της Εθνικής Τράπεζας.

Μετά τη συστηματική μελέτη των μοντέλων που έχουν αναπτυχθεί για την αντιμετώπιση του ζητήματος του Ελέγχου των προσβάσεων, γίνονται συγκρίσεις και εξάγονται συμπεράσματα.

Ειδικότερα, μελετήθηκε ενδελεχώς το μοντέλο του Ελέγχου πρόσβασης που βασίζεται σε Ρόλους και διατυπώθηκε αρχικά κατά το 1992. Το μοντέλο αυτό παρουσιάζει σημαντικά πλεονεκτήματα στη χρήση του σε μεγάλους και «πολύπλοκους» Οργανισμούς. Το RBAC παρότι είναι ένα μοντέλο που έχει σχεδιαστεί εδώ και πολλά χρόνια εξακολουθεί να είναι αντικείμενο συζητήσεων και έρευνας. Συνεχώς, δημιουργούνται νέα λογισμικά που ακολουθούν τις κατευθυντήριες γραμμές του και εταιρείες κολοσσοί στο χώρο των ΤΠΕ το χρησιμοποιούν σαν βασικό εργαλείο για την επίλυση του προβλήματος του ελέγχου πρόσβασης.

Στην ΜΔΕ εκτός από τη περιγραφή του μοντέλου, υπάρχει αναφορά:

- στο πρότυπο ANSI/INCITS 359-2004,
- στις πολλές και χρήσιμες επεκτάσεις του,
- στις βέλτιστες πρακτικές που πρέπει να χρησιμοποιούνται καθώς και
- σε εταιρείες κολοσσούς στο χώρο της ΤΠΕ που δημιουργούν καθημερινά όλο και περισσότερα λογισμικά τα οποία εφαρμόζουν τον Έλεγχο Πρόσβασης Βασισμένο σε Ρόλους.

Ο Έλεγχος Πρόσβασης στα Χρηματοπιστωτικά Ιδρύματα αποτελούσε πάντα μια σπουδαία πρόκληση. Ανάλογες της πρόκλησης είναι και οι δυσκολίες που προκύπτουν εξ' αιτίας του μεγέθους των Ιδρυμάτων αυτών, του πλήθους των χρηστών και των εφαρμογών που χρησιμοποιούν καθώς και της μεγάλης πολυπλοκότητάς τους. Έτσι, ο σχεδιασμός ενός συστήματος και η υλοποίηση μιας ρόλο-κεντρικής Πολιτικής Ασφαλείας για ένα υποκατάστημα τράπεζας έδωσε το έναυσμα για το πρακτικό μέρος της εργασίας.

Για να δομηθεί σωστά και να λειτουργήσει αποτελεσματικά ένα σύστημα διαχείρισης πρόσβασης βασισμένο σε ρόλους πρέπει να γίνουν συγκεκριμένα βήματα και να εφαρμοστούν οι διεθνείς βέλτιστες πρακτικές.

Στην ΜΔΕ, θα γίνει ανάλυση απαιτήσεων, θα σχεδιαστεί και υλοποιηθεί μια πολιτική ελέγχου πρόσβασης η οποία θα είναι άμεσα εφαρμόσιμη (στο περιβάλλον ενός τραπεζικού υποκαστήματος) και θα έχει σαν στόχο την αξιοποίηση των πλεονεκτημάτων του RBAC

Λέξεις κλειδιά: **Έλεγχος Πρόσβασης, RBAC, Υλοποίηση RBAC σε Windows, Authorization Manager**

1 Εισαγωγή

1.1 Ορισμός Προβλήματος

Η προστασία της πληροφορίας ήταν, είναι και θα είναι πάντα επιτακτική και αναγκαία, άλλωστε η ακεραιότητα της πληροφορίας είναι ένας από τους τρεις βασικούς πυλώνες των ΤΠΕ. Από τη στιγμή που αναπτύχθηκαν και λειτούργησαν πληροφοριακά συστήματα, αναδείχθηκε η ανάγκη προστασίας της ψηφιακής πληροφορίας. Ένας από τους βασικότερους τρόπους προστασίας της πληροφορίας στο χώρο των ΤΠΕ είναι ο Έλεγχος Πρόσβασης.

Ο Έλεγχος Πρόσβασης μπήκε από πολύ νωρίς στο χώρο των Η/Υ και των Πληροφοριακών Συστημάτων καθώς οι πρώτες του αρχές θεμελιώθηκαν το 1975 από τους Salzer & Shroeder (Salzer & Shroeder, 1975). Σύμφωνα με τους Sandhu & Samarati, «*Ο σκοπός του ελέγχου πρόσβασης είναι να περιορίσει τις δράσεις και ενέργειες που ένας νόμιμος χρήστης, ενός συστήματος Η/Υ μπορεί να εκτελέσει*» (Sandhu R., Samarati P., 1994).

Με την πάροδο του χρόνου, αναπτύχθηκαν διάφορα μοντέλα, δημιουργήθηκαν πρότυπα και η επιστημονική κοινότητα παρουσίασε διαφορετικές τάσεις, λύσεις και προτάσεις γύρω από τη συγκεκριμένη θεματική ενότητα. Σήμερα, υπάρχουν προτάσεις που καλύπτουν τις περισσότερες ανάγκες της αγοράς. Έτος σταθμός για τον Έλεγχο Πρόσβασης είναι το 1992 όταν οι Feraiolo και Kuhn θεμελιώνουν **τον Έλεγχο Πρόσβασης βασισμένο σε Ρόλους, το γνωστό σε όλους ως RBAC.**

Το μοντέλο αυτό είναι μια προέκταση του μοντέλου του υποχρεωτικού ελέγχου MAC. Τα πλεονεκτήματα που προσδίδει το RBAC είναι πολλά. Η επικράτησή του σε σχέση με τα άλλα βασικά μοντέλα αργεί αλλά με την πάροδο του χρόνου γίνεται καθολική. Μέρα με τη μέρα, όλο και περισσότεροι δημιουργοί λογισμικών που ασχολούνται με τον Έλεγχο Πρόσβασης των χρηστών σε πόρους και πληροφορίες δημιουργούν λογισμικά, που είναι βασισμένα στο RBAC και μπορούν να εφαρμόσουν τον Έλεγχο Πρόσβασης βασισμένο σε Ρόλους.

Όμως, όσο η πολυπλοκότητα των επιχειρήσεων αυξάνεται, αναζητούνται συνθετότερες λύσεις που να αξιοποιούν τα θετικά από όλα τα μοντέλα και να παρακάμπτουν τις δυσκολίες. Οι τράπεζες είναι οργανισμοί με πολύ μεγάλη πολυπλοκότητα όσον αναφορά στη διαχείριση των χρηστών και στη διενέργεια του ελέγχου πρόσβασης.

Βάσει αυτών των συνθηκών, η παρούσα μελέτη θα προσπαθήσει να δώσει λύσεις, σχεδιάζοντας και υλοποιώντας μια ρόλο-κεντρική πολιτική ασφάλειας η οποία θα μπορεί να εφαρμοστεί ώστε να πραγματοποιείται ο έλεγχος της πρόσβασης των υπαλλήλων, των

υποκαταστημάτων μιας τράπεζας, έχοντας ως οδηγό την πολιτική ασφάλειας του οργανισμού.

1.2 Σκοπός - Στόχοι

Η παρούσα μελέτη έχει ως αντικείμενο τη μελέτη των μέχρι σήμερα γνωστών μοντέλων, τη εμβάθυνση στον Έλεγχο Πρόσβασης βασισμένου σε Ρόλους, την εξαγωγή και παράθεση συμπερασμάτων.

Οι Θεωρητικοί Στόχοι της μελέτης αφορούν στην:

- Πλήρη επισκόπηση του πεδίου του Ελέγχου Πρόσβασης.
- Ανάλυση του Ελέγχου Πρόσβασης Βασισμένου σε Ρόλους
- Ανάλυση του προτύπου του ANSI/INCITS 359-2004

Εκτός όμως από τη μελέτη της βιβλιογραφίας και την εξαγωγή των συμπερασμάτων χρειάζεται να τεθούν και στόχοι που αφορούν στο πρακτικό κομμάτι της εργασίας. Η λειτουργία ενός τραπεζικού υποκαταστήματος βασίζεται στην πρόσβαση των χρηστών στα Πληροφοριακά Συστήματα του Οργανισμού. **Ως εκ τούτου στόχος της παρούσας ΜΔΕ είναι ο σχεδιασμός και υλοποίηση μιας πολιτικής η οποία θα εφαρμόζει τον έλεγχο πρόσβασης βασισμένο σε ρόλους και η δημιουργία της απαραίτητης σχετικής εφαρμογής.**

1.3 Μεθοδολογία

Ό Έλεγχος Πρόσβασης δεν είναι μια καινούργια έννοια. Απεναντίας υπάρχει για πολλά χρόνια στο χώρο της Ασφάλειας των ΤΠΕ. Ως εκ τούτου για τη συγγραφή της παρούσας ΜΔΕ μελετήθηκαν επιστημονικά κείμενα τα οποία συνεγράφησαν από το 1975 μέχρι και το 2012. Κυριότερο βάρος δόθηκε στους συγγραφείς θεμελιωτές του Role Based Access Control: Ferraiollo, Khun, Sahndu, καθώς και σε σύγχρονα επιστημονικά κείμενα της Sammarati, η οποία σήμερα θεωρείται ίσως η πλέον καταξιωμένη ερευνήτρια στο χώρο του Ελέγχου Πρόσβασης.

Σύμφωνα με τα παραπάνω, η παρούσα ΜΔΕ μελετά τα χαρακτηριστικά των μοντέλων του Ελέγχου Πρόσβασης, εξηγεί τον τρόπο λειτουργίας τους και παραθέτει τις βασικές τους αρχές. Στη συνέχεια γίνεται εμβάθυνση στο μοντέλο του Ελέγχου Πρόσβασης βασισμένου σε Ρόλους. Το συγκεκριμένο μοντέλο αναλύεται πλήρως, παρουσιάζεται η πορεία του στο χρόνο μέσω επιλεγμένων προεκτάσεων του και παρουσιάζεται το πρότυπο που έχει προτείνει ο NIST και υιοθετήσει ο ANSI. Τέλος, η εργασία ολοκληρώνεται με την παράθεση της

πειραματικής διαδικασίας, η οποία εκπονήθηκε και αφορά στη λειτουργία τραπεζικού υποκαταστήματος για τη διενέργεια συγκεκριμένων εργασιών.

Η ΜΔΕ αποτελείται από 5 κεφάλαια.

Το κεφάλαιο 1 είναι το εισαγωγικό της εργασίας. Σε αυτό γίνεται ο ορισμός του προβλήματος το οποίο πραγματεύεται η μελέτη, παρουσιάζεται ο σκοπός, τίθενται οι στόχοι και περιγράφεται η δομή της εργασίας.

Στο κεφάλαιο 2 αναφέρεται η μελέτη των πλέον γνωστών μοντέλων Ελέγχου Πρόσβασης.

Στο κεφάλαιο 3 περιγράφεται η παρουσίαση του ελέγχου πρόσβασης που βασίζεται σε ρόλους. Επεξηγείται με λεπτομέρεια ο τρόπος λειτουργίας, παρουσιάζεται η πορεία του στο χρόνο, αναφέρονται οι πλέον ενδιαφέρουσες και χρήσιμες επεκτάσεις του και τέλος παρουσιάζεται το πρότυπο ANSI/INCITS 359-2004.

Στο κεφάλαιο 4 καταγράφεται το πρακτικό μέρος της παρούσας ΜΔΕ. Αρχικώς πραγματοποιείται η ανάλυση απαιτήσεων για τη μελέτη περίπτωσης, παρουσιάζεται με σαφήνεια και ακρίβεια το εργαλείο του Authorization Manager μέσω του οποίου χτίστηκε η εφαρμογή και τέλος παρουσιάζεται η υλοποίησή της.

Στο κεφάλαιο 5, που είναι και το τελευταίο της ΜΔΕ καταγράφονται τα συμπεράσματα που προέκυψαν μέσα από την εκπόνηση της εργασίας και δίνονται μελλοντικές κατευθύνσεις για τη συνέχισή της και την αξιοποίησή της σε διαφορετικά πεδία.

Η εργασία ολοκληρώνεται με τη παράθεση των βιβλιογραφικών πηγών και του παραρτήματος όπου περιέχεται ο πηγαίος κώδικας της εφαρμογής που σχεδιάστηκε.

2 Μηχανισμοί Ελέγχου Πρόσβασης

2.1 Εισαγωγή

Αρχικά θα πρέπει να δοθεί ένας ορισμός για το τι είναι έλεγχος πρόσβασης ή έλεγχος προσπέλασης. Ανατρέχοντας στη διεθνή βιβλιογραφία συναντάμε ένα πλήθος ορισμών. Ενδεικτικά αναφέρονται τέσσερις. «*Ο Έλεγχος Πρόσβασης αξιολογεί τις αιτήσεις πρόσβασης στους πόρους από τους εξουσιοδοτημένους χρήστες και, με βάση ορισμένους κανόνες πρόσβασης, καθορίζει το αν πρέπει να χορηγήσει ή να απαγορεύσει την πρόσβαση*» (Samarati, 2008). *«Η δυνατότητα να επιτρέπεται μόνο σε εξουσιοδοτημένους χρήστες ή προγράμματα ή διεργασίες, η πρόσβαση σε πόρους του συστήματος.» «Η χορήγηση ή άρνηση, σύμφωνα με ένα συγκεκριμένο μοντέλο ασφάλειας, δικαιωμάτων πρόσβασης σε ένα σύνολο πόρων.»* (Syracuse University, 2010) *«Ένα σύνολο διαδικασιών που εκτελούνται από το υλικό (hardware), το λογισμικό (software) και τους διαχειριστές, για την παρακολούθηση της πρόσβασης, τον εντοπισμό των χρηστών που ζητούν πρόσβαση, την καταγραφή των προσπαθειών απόκτησης πρόσβασης, και τη χορήγηση ή την άρνηση άδειας πρόσβασης με βάση καθιερωμένους κανόνες.»* (Moore, 2001). «*Ο Έλεγχος Προσπέλασης αφορά τους συγκεκριμένους μηχανισμούς που υλοποιεί κάποιο υπολογιστικό σύστημα, με σκοπό να προστατεύσει από μη εξουσιοδοτημένη προσπέλαση τα δεδομένα εκείνα, τα οποία επηρεάζουν την ασφάλεια του συστήματος και είναι αποθηκευμένα στη μνήμη, στους καταχωρητές ή σε οποιοδήποτε άλλο τμήμα του.*» (Λαμπρινουδάκης, 2012)

Ο Έλεγχος Πρόσβασης χωρίζεται σε δύο μεγάλες ενότητες. Στον Φυσικό Έλεγχο Πρόσβασης και στον Λογικό Έλεγχο Πρόσβασης. Στην παρούσα εργασία θα ασχοληθούμε αποκλειστικά με τη δεύτερη κατηγορία. Σε αυτό το κεφάλαιο θα γίνει μια ιστορική αναδρομή στον Έλεγχο Πρόσβασης, θα παρουσιαστούν οι βασικές αρχές και οι βασικές δομές που ακολουθούνται από το παρελθόν μέχρι και σήμερα. Στη συνέχεια θα παρουσιαστούν τα διαφορετικά μοντέλα με τις παραμέτρους τους και θα εξεταστεί η πορεία των μοντέλων στο χρόνο.

Ό Έλεγχος Πρόσβασης απασχολεί τους οργανισμούς από τότε που άρχισαν να διακινούν πληροφορίες μέσα από πληροφοριακά συστήματα. Το παρόν κεφάλαιο έχει ως στόχο τη μελέτη των βασικών εννοιών, των βασικών αρχών της θεματικής ενότητας, καθώς και των βασικών δομών και πολιτικών Ελέγχου Πρόσβασης.

Σήμερα οι επιχειρήσεις ζουν στη κοινωνία τη πληροφορίας. Για να διασφαλιστεί η ομαλή παραγωγική τους διαδικασία απαιτείται οι εργαζόμενοι να έχουν πρόσβαση τόσο σε

πληροφορίες όσο και σε πόρους. Σύμφωνα με το NIST ο Έλεγχος Πρόσβασης ήρθε να δώσει απαντήσεις στα εξής κρίσιμα ερωτήματα :

- ✓ What - Γιατί κάποιος να έχει πρόσβαση;
- ✓ Who – Ποιος έχει πρόσβαση;
- ✓ Where – Που έχει πρόσβαση;

Τέλος αξίζει να αναφερθεί ότι στις ΤΠΕ υπάρχουν δύο τύποι ελέγχου οι οποίοι χρησιμοποιούνται κατά κανόνα σε διαφορετικού τύπου υπολογιστικά συστήματα. Για την επιβολή ελέγχου πρόσβασης στους πόρους ενός συστήματος θα πρέπει να έχει προσδιορισθεί:

- ✓ Τι μπορεί να κάνει ένα υποκείμενο
- ✓ Τι μπορεί να γίνει σε ένα αντικείμενο

Η πρώτη προσέγγιση συνήθως βρίσκει εφαρμογή στα συστήματα διαχείρισης βάσεων δεδομένων και σε συστήματα διαχείρισης χρηστών, ενώ η δεύτερη συναντάται περισσότερο στα κλασσικά λειτουργικά συστήματα.

2.2 Βασικές Έννοιες

Στην πορεία του παρόντος κεφαλαίου, όπως και όλης της ΜΔΕ, θα υπάρχει συνεχής αναφορά σε συγκεκριμένες έννοιες που είναι δομικά στοιχεία του Ελέγχου Πρόσβασης. Με στόχο την ομαλή πορεία της μελέτης, κρίνεται σκόπιμο, να υπάρξει αναφορά και ορισμός αυτών των Βασικών Εννοιών.

- **Χρήστης**

Είναι το άτομο που αλληλεπιδρά με τον Η/Υ και έχει πρόσβαση σε πόρους του συστήματος.

- **Ρόλος**

Πρόκειται για ένα σύνολο ενεργειών-ευθυνών οι οποίες σχετίζονται με συγκεκριμένη λειτουργία (π.χ. Ταμίας).

- **Υποκείμενο**

Υποκείμενο είναι μια οντότητα, που για να λειτουργήσει χρειάζεται να έχει δικαίωμα πρόσβασης στο σύστημα, για παράδειγμα οι χρήστες και οι Διεργασίες. (Λαμπρινουδάκης, 2004)

- **Αντικείμενο**

Με τον όρο αντικείμενο αντιπροσωπεύονται τα αρχεία και οι άλλοι υπολογιστικοί πόροι όπως: μνήμη, εκτυπωτές και άλλοι κόμβοι ενός δικτύου υπολογιστών. (Λαμπρινουδάκης, 2004)

- **Πολιτική Ασφάλειας**

Ως Πολιτική Ασφάλειας ορίζεται το έγγραφο το οποίο με επίσημο και δομημένο τρόπο καθορίζει τις γενικές αρχές που πρέπει να ισχύουν για την ασφάλεια του συστήματος που εφαρμόζεται. Μια πολιτική ασφάλειας εκφράζεται με κανόνες που ρυθμίζουν τον τρόπο που ελέγχονται τα συμμετέχοντα μέρη και πως λαμβάνονται οι αποφάσεις για προσπέλαση. Συνήθως επιβάλλονται από διάφορους μηχανισμούς ασφάλειας, οι οποίοι, μπορούν να καταταγούν στις παρακάτω κατηγορίες: (Πάγκαλος, 2002)

- ✓ Αναγνώριση (Identification)
- ✓ Αυθεντικοποίηση (Authentication)
- ✓ Εξουσιοδότηση (Authorization)
- ✓ Έλεγχο πρόσβασης (Access Control)
- ✓ Ακεραιότητα (Integrity)
- ✓ Συνέπειαα (Consistency)
- ✓ Επιθεώρηση (Auditing)

- **Μοντέλο Ασφάλειας**

Ορίζει επίσημα τόσο τις προδιαγραφές όσο και την επιβολή του Ελέγχου Πρόσβασης. (Samarati, 2008)

- **Μηχανισμός Ασφάλειας**

Εφαρμόζει τις πολιτικές ασφάλειας μέσω χαμηλού επιπέδου (λογισμικού και hardware) διαδικασιών

Αφού ορίστηκαν οι βασικές έννοιες, κρίνεται χρήσιμο να εξηγηθούν οι βασικές διαφορές μεταξύ των Μοντέλων και των Μηχανισμών Ασφάλειας. Ο διαχωρισμός αυτός, σύμφωνα με τη Samarati, μας επιτρέπει να :

I. Εξετάσουμε τις απαιτήσεις Πρόσβασης ανεξάρτητα από την εφαρμογή τους
II. Συγκρίνουμε τις διαφορετικές πολιτικές ελέγχου πρόσβασης και τους διαφορετικούς μηχανισμούς που εμπεριέχονται στην ίδια πολιτική

III. Σχεδιάσουμε μηχανισμούς που είναι σε θέση να επιβάλουν πολλαπλές πολιτικές

Η κατηγορία III είναι αυτή που δίνει τη δυνατότητα να υλοποιηθεί η θεωρητική ιδέα της ΜΔΕ.

2.3 Βασικές Αρχές Ασφάλειας για τους Μηχανισμούς Ελέγχου Πρόσβασης

Αρκετές είναι οι βασικές αρχές ασφάλειας οι οποίες εφαρμόζονται από τους Μηχανισμούς και τα Μοντέλα Ελέγχου Πρόσβασης. Οι αρχές αυτές έχουν σαν στόχο να μην επιτρέπεται η παραβίαση των καθορισμένων απαιτήσεων ασφάλειας που αφορούν τα δικαιώματα πρόσβασης.

Οι αρχές αυτές, αρχικά δημοσιεύτηκαν το 1975 από τους Jerome Saltzer και Michael Schroeder και είναι οι παρακάτω:

- Economy of Mechanism (Οικονομία Μηχανισμού)

Σύμφωνα με το Bishop, ως αρχή της Οικονομίας του Μηχανισμού ορίζεται η τάση της διατήρησης του σχεδιασμού στην απλούστερη δυνατή μορφή. Η συγκεκριμένη αρχή υλοποιείται κυρίως στα κατώτερα στρώματα και έχει ως αποτέλεσμα την καλύτερη προστασία αυτών των στρωμάτων.

- Fail Safe Defaults (Προεπιλογή της Ασφάλειας)

Σύμφωνα με αυτή την αρχή η χορήγηση πρόσβασης βασίζεται περισσότερο στη παροχή συγκεκριμένων δικαιωμάτων παρά στον αποκλεισμό. (University of California, 2012)

- Complete Mediation (Πλήρους Μεσολάβησης)

Κάθε πρόσβαση σε κάθε αντικείμενο, πρέπει να ελέγχεται από το διαχειριστή και εν συνεχεία να παραχωρείται δικαίωμα πρόσβασης. Με τη συστηματική εφαρμογή της παρούσας αρχής θεμελιώνεται η πρωταρχική ασφάλεια του συστήματος, καθώς για κάθε αντικείμενο πρέπει να αιτείται ο χρήστης πρόσβαση κάθε φορά που χρειάζεται να το χρησιμοποιήσει. (Saltzer, 1975)

- Open Design (Ανοιχτού Σχεδιασμού)

Από το 1975 οι Saltzer και Shroeder υποστήριξαν ότι η ασφάλεια ενός μηχανισμού ασφάλειας δεν θα πρέπει να εξαρτάται από το αν είναι γνωστός ή όχι. Χαρακτηριστικά γίνεται αναφορά στο ότι, αν οι σχεδιαστές μοιράζονται με τους χρήστες όλη τη γνώση τότε

οι τελευταίοι αισθάνονται περισσότερο ασφαλείς στο σύστημα που σχεδίασαν οι πρώτοι. Σήμερα αυτή η αρχή ακολουθείται στους κόλπους της Ασφάλειας των ΤΠΕ κυρίως με τους κρυπτογραφικούς αλγορίθμους.

- Separation of Privileges (Διαχωρισμού Προνομίων)

Ο Διαχωρισμός Προνομίων είναι μια αρχή που υποστηρίζει ότι το σπάσιμο ενός ενιαίου προνομίου σε επιμέρους, ανεξαρτήτως των συστατικών και των ανθρώπων, είναι απαραίτητο έτσι ώστε να υπάρξουν πολλαπλές συμφωνίες για την ολοκλήρωση μιας ενέργειας. (Meunier, 2008)

- Least Common Mechanism (Λιγότερο Συνήθης Μηχανισμός)

Περισσότεροι του ενός χρήστες δεν θα πρέπει να μοιράζονται τον ίδιο μηχανισμό για να αποκτήσουν πρόσβαση σε ένα πόρο. (Collberg, 2012)

- Psychological Acceptability (Ψυχολογική Αποδεκτικότητα)

Τα συστήματα και οι διεπαφές θα πρέπει να είναι κατανοητές. Επίσης δεν θα πρέπει να θέτουν προ εκπλήξεως τους χρήστες και θα πρέπει να είναι προεγκατεστημένες στη κατάσταση που επιθυμεί να τις βρει ο χρήστης. (Collberg, 2012)

- Least Privileges (Ελάχιστων Προνομίων ή Ελάχιστων Δικαιωμάτων)

Από αυτές τις αρχές κρίνεται σκόπιμο να ορίσουμε και να περιγράψουμε αυτή των «Ελαχίστων Προνομίων» περισσότερο από τις προηγούμενες, καθώς είναι δομικό στοιχείο, του ελέγχου πρόσβασης βασισμένου σε ρόλους.

Η αρχή των «Ελαχίστων Προνομίων», σύμφωνα με τους Saltzer & Schroeder (1975) ορίζεται ως: «*Κάθε πρόγραμμα και κάθε εξουσιοδοτημένος χρήστης του συστήματος θα πρέπει να λειτουργεί, χρησιμοποιώντας τον ελάχιστο αριθμό από Δικαιώματα ο οποίος είναι απαραίτητος, για την επιτυχή και ολοκληρωμένη περάτωση της εργασίας του*». Βασικός στόχος της αρχής είναι η διατήρηση της ακεραιότητας. Εφαρμόζοντας την αρχή υποχρεωτικά θα πρέπει να αναγνωριστεί ποια είναι η εργασία που καλείται ο χρήστης να φέρει εις πέρας. Εν συνεχεία του χορηγείται πρόσβαση στους ελάχιστους πόρους ώστε να είναι εφικτή η ευχερής εκτέλεση της εργασίας του. Με την άρνηση χορήγησης επιπλέον δικαιωμάτων πρόσβασης διασφαλίζεται η μη παράκαμψη της εφαρμοζόμενης πολιτικής ασφάλειας.

2.4 Βασικές Δομές Ασφάλειας στον Έλεγχο Πρόσβασης

Στην παράγραφο αυτή θα γίνει περιγραφή των βασικών δομών ασφάλειας που υπάρχουν στον Έλεγχο Πρόσβασης. Αξίζει να αναφερθεί ότι θα παρουσιαστούν τα ακόλουθα μοντέλα:

- ✓ Πίνακας Ελέγχου Προσπέλασης
- ✓ Λίστες Ελέγχου Δικαιωμάτων Προσπέλασης
- ✓ Λίστες Δυνατοτήτων

Πριν όμως γίνει η παρουσίαση των μοντέλων πρέπει να καταστεί σαφές ότι τα δικαιώματα προσπέλασης διαφέρουν ανάλογα με το υπολογιστικό σύστημα.

2.4.1 Πίνακας Ελέγχου Προσπέλασης (Access Matrix)

Για 40 και πλέον χρόνια ο Πίνακας Ελέγχου Προσπέλασης αποτελεί έναν βασικό τρόπο απεικόνισης των δικαιωμάτων πρόσβασης. Δημιουργήθηκε το 1971 από το Lampson και βελτιώθηκε από τους Graham(1972) και Denning (1971). Χάρη στην απλή δομή του και την εύκολη εξαγωγή πληροφορίας από αυτόν, χρησιμοποιείται ακόμα και σήμερα.

	BIBLIOG	TEMP	F	HELP.TXT	C_COMP	LINKER	SYS_CLOCK	PRINTER
USER A	ORW	ORW	ORW	R	X	X	R	W
USER B	R	-	-	R	X	X	R	W
USER S	RW	-	R	R	X	X	R	W
USER T	-	-	-	R	X	X	R	W
SYS_MGR	-	-	-	RW	OX	OX	ORW	O
USER_SVCS	-	-	-	O	X	X	R	W

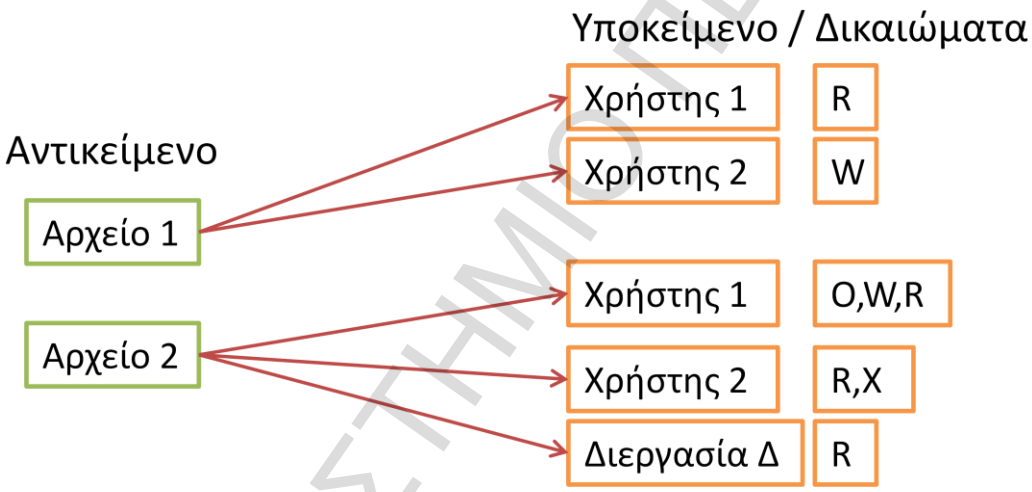
Εικόνα 1 Παράδειγμα Πίνακα Ελέγχου Προσπέλασης

Η εικόνα 1 παρουσιάζει έναν ενδεικτικό Πίνακα Ελέγχου Προσπέλασης. Ο Πίνακας αποτυπώνει όλες τις πιθανές καταστάσεις που αφορούν τόσο τους χρήστες όσο και τις διεργασίες. Οι χρήστες και οι διεργασίες αποτελούν τα υποκείμενα και απεικονίζονται στις σειρές. Τα αντικείμενα είναι τα αρχεία και απεικονίζονται στις στήλες. Ενδεικτικό παράδειγμα για τον τρόπο ανάγνωσης του παραπάνω πίνακα είναι ότι ο χρήστης USER A έχει δικαιώματα ιδιοκτησίας, ανάγνωσης και εγγραφής στο αρχείο BIBLIOG, ενώ για το αντίστοιχο αντικείμενο το υποκείμενο USER B έχει μόνο το δικαίωμα της ανάγνωσης. Ο Πίνακας Ελέγχου Πρόσβασης αποτυπώνει την εικόνα ενός συστήματος μια συγκεκριμένη χρονική στιγμή.

Βασικό πλεονέκτημα του Πίνακα είναι η δυνατότητα που δίνει ώστε να γίνει σαφής προσδιορισμός των προσβάσεων του κάθε χρήστη σε ένα υπολογιστικό σύστημα.

2.4.2 Λίστες Ελέγχου Δικαιωμάτων Προσπέλασης

Οι Λίστες Ελέγχου Δικαιωμάτων Προσπέλασης, μια μορφή υλοποίησης του Πίνακα Ελέγχου Προσπέλασης είναι πολύ διαδεδομένες, ιδιαίτερα στο περιβάλλον των Microsoft Windows. Οι Λίστες Ελέγχου Δικαιωμάτων προσπέλασης είναι πολύ γνωστές με τη συντομογραφία τους ACL (Access Control Lists). Πρόκειται για ένα σύνολο δεδομένων, που ενημερώνει το λειτουργικό σύστημα αναφορικά με τις άδειες προσπέλασης ή τα δικαιώματα πρόσβασης που έχει το υποκείμενο σε σχέση με τα αντικείμενα του συστήματος. Κάθε αντικείμενο έχει ένα μοναδικό χαρακτηριστικό ασφάλειας, το οποίο εντοπίζει τους χρήστες που έχουν πρόσβαση σε αυτό. Η Λίστα Ελέγχου Δικαιωμάτων Προσπέλασης είναι μια λίστα συνδεδεμένη με το κάθε αντικείμενο και τα αντίστοιχα δικαιώματα πρόσβασης που έχει ο χρήστης πάνω σε αυτό. Μαθηματικά η παραπάνω πρόταση αποτυπώνεται από την ακόλουθη σχέση: $l = \{ (s, a) : s \in S, a \subseteq A \}$.



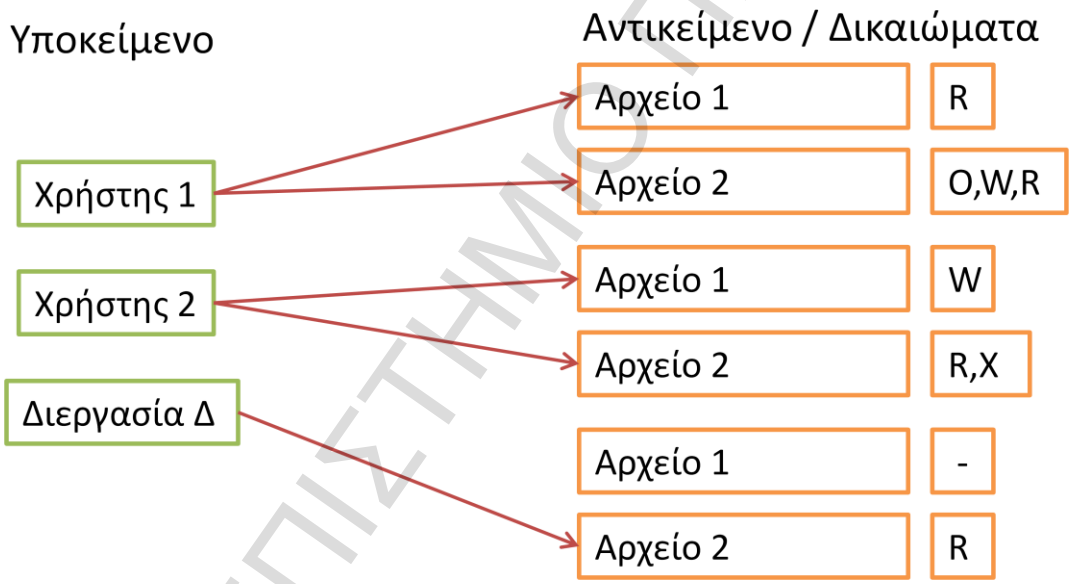
Εικόνα 2 Παράδειγμα Λίστας Ελέγχου Δικαιωμάτων Προσπέλασης

Οι Λίστες Ελέγχου Δικαιωμάτων Προσπέλασης έχουν κάποια συγκριτικά πλεονεκτήματα σε σχέση με τον Πίνακα Ελέγχου Προσπέλασης. Το βασικότερο πλεονέκτημα είναι η εύκολη εύρεση και το είδος των δικαιωμάτων των χρηστών σε σχέση με κάποιο αντικείμενο. Όπως φαίνεται από την εικόνα 2, για το Αρχείο 1 ο Χρήστης 1 έχει δικαίωμα Ανάγνωσης και ο Χρήστης 2 δικαίωμα Εγγραφής. Επιπλέον μπορούν τα υποκείμενα να αντικατασταθούν από ομάδες ή σύνολα υποκειμένων ώστε να μειωθεί το μέγεθός τους. Συνήθως οι Λίστες Ελέγχου Δικαιωμάτων Προσπέλασης απαντώνται σε υλοποιήσεις του μοντέλου του Διακριτικού Ελέγχου Πρόσβασης που θα περιγραφεί παρακάτω.

2.4.3 Λίστες Δυνατοτήτων

Επί της ουσίας, οι Λίστες Δυνατοτήτων εφαρμόζουν την αντίθετη προσέγγιση από τις αντίστοιχες των Δικαιωμάτων. Στη διεθνή ορολογία τις συναντάμε ως Capability Lists. Φέρνοντας στο μυαλό μας τον Πίνακα Ελέγχου Πρόσβασης οι Λίστες Δυνατοτήτων λειτουργούν διαβάζοντας τον Πίνακα ανά γραμμή. Στόχος τους είναι η παρουσίαση όλων των αδειών και των δικαιωμάτων που έχει ένα υποκείμενο. Όπως αναφέρεται στη βιβλιογραφία, «*Κάθε υποκείμενο του συστήματος συσχετίζεται με μία ομάδα από ζεύγη, κάθε ζεύγους αποτελούμενου από ένα αντικείμενο και ένα σύνολο λειτουργιών προσπέλασης - δικαιωμάτων προσπέλασης. Το συσχετιζόμενο υποκείμενο έχει τη δυνατότητα να προσπελάσει το συγκεκριμένο αντικείμενο με οποιαδήποτε από τις αναφερόμενες λειτουργίες προσπέλασης.*» (Λαμπρινουδάκης, 2012)

Μαθηματικά η Λίστα Δυνατοτήτων ορίζεται ως εξής: $c = \{ (o, a) : o \in O, a \subseteq A \}$.



Εικόνα 3 Παράδειγμα Λίστας Δυνατοτήτων

Η Εικόνα 3 αναπαριστά την Εικόνα 2 σε μορφή λίστας δυνατοτήτων. Για να διευκολυνθεί η διαχείριση των υπολογιστικών συστημάτων σπάνια συναντάμε αυτή τη προσέγγιση σαν αποτύπωση της πολιτικής ασφάλειας του συστήματος. Τα πλεονεκτήματα των Λιστών Δυνατοτήτων εμφανίζονται στα κατανεμημένα συστήματα καθώς δεν χρειάζεται η επαναλαμβανόμενη αυθεντικοποίηση. Αυτό επιτρέπει σε ένα υποκείμενο να αυθεντικοποιηθεί μία φορά, να λάβει τις δυνατότητες του και μετά να λάβει τις αντίστοιχες υπηρεσίες από τους servers του συστήματος (Sandhu & Samarati, 1994).

2.5 Μοντέλα Ελέγχου Προσπέλασης

Με την πάροδο του χρόνου έχει αναπτυχθεί πληθώρα μοντέλων, τα οποία έχουν ως στόχο την συμβολική απεικόνιση μιας πολιτικής, που πρέπει να επιβληθεί σε ένα υπολογιστικό σύστημα. Ερευνητές της Ασφάλειας των ΤΠΕ τα ονομάζουν και φορμαλιστικά μοντέλα (Κάτσικας, 2004). Εν συνεχεία λοιπόν θα παρουσιαστούν κάποια από τα βασικότερα μοντέλα ελέγχου πρόσβασης. Αυτά είναι:

- ✓ Bell-LaPadula
- ✓ Biba
- ✓ Σινικού Τείχους
- ✓ Graham - Denning
- ✓ Harrison-Ruzzo-Ulman

2.5.1 Μοντέλο Bell – LaPadula

Το μοντέλο Bell – LaPadula ανήκει στα φορμαλιστικά μοντέλα ελέγχου πρόσβασης. Αναπτύχθηκε από τους David Elliot Bell και Len La Padula την περίοδο 1972-1975 (Bell, 2005). Σύμφωνα με τους δημιουργούς πρόκειται για ένα μοντέλο ασφάλειας το οποίο παράγει εννοιολογικά μέσα για την ανάλυση και το σχεδιασμό ασφαλών υπολογιστικών συστημάτων. Αξίζει να αναφερθεί ότι δεν δημιουργήθηκε έχοντας ως στόχο τη χρησιμοποίηση του ως μοντέλου ελέγχου προσπέλασης. Πιο συγκεκριμένα, στόχος του ήταν η διασφάλιση ασφαλών ροών πληροφορίας (Κάτσικας, 2004).

Σήμερα χρησιμοποιείται κυρίως σε συστήματα ελέγχου πρόσβασης κυβερνητικών και στρατιωτικών υπηρεσιών (Hansche, 2003). Ο λόγος που χρησιμοποιείται το μοντέλο των Bell και LaPadula είναι το ότι δίνει έμφαση στα θέματα εμπιστευτικότητας και στη προστασία απόρρητων πληροφοριών.

Στο παρόν μοντέλο οι οντότητες του συστήματος χωρίζονται αποκλειστικά σε υποκείμενα και αντικείμενα. Στο Bell-LaPadula συναντώνται οι καταστάσεις και ορίζεται η έννοια της «ασφαλούς κατάστασης». Αποδεικνύεται ότι κάθε μετάβαση διατηρεί την ασφάλεια του συστήματος αν και μόνο αν τόσο η αρχική όσο και η τελική κατάσταση διατηρούν την οντότητά τους και δεν μπορεί να υποστούν οποιαδήποτε μεταβολή. Με χρήση του επαγωγικού συλλογισμού αποδεικνύεται ότι το σύστημα ανταποκρίνεται στους βασικούς στόχους ασφαλείας του μοντέλου.

Στο σημείο αυτό θα γίνει μια σύντομη παρουσίαση του τρόπου λειτουργίας του μοντέλου Bell-LaPadula. Αρχικά ορίζεται μια ασφαλής κατάσταση του συστήματος αν και μόνο αν οι προσβάσεις των υποκειμένων στα αντικείμενα συμφωνούν με την ορισμένη πολιτική ασφάλειας του συστήματος. Για να προσδιοριστεί αν μια συγκεκριμένη λειτουργία επιτρέπεται, συγκρίνεται το clearance του υποκειμένου με την τάξη του αντικειμένου (συνδυάζοντας την ταξινόμηση και το σύνολο των στοιχείων που συνθέτουν το επίπεδο ασφάλειας) ώστε να καθοριστεί εάν το υποκείμενο έχει εξουσιοδοτηθεί για τη συγκεκριμένη λειτουργία. Το σύστημα σύγκρισης του clearance/ τάξης εκφράζει επί της ουσίας τους όρους ενός πλέγματος ασφάλειας. Το μοντέλο αυτό συνδυάζει αρχές τόσο του υποχρεωτικού (MAC) όσο και του διακριτικού (DAC) ελέγχου πρόσβασης που περιγράφονται αναλυτικά στα δύο επόμενα υποκεφάλαια 2.6 & 2.7 αντίστοιχα.

2.5.2 Μοντέλο Biba

Το μοντέλο Biba δημιουργήθηκε το 1977 από τον Kenneth Biba και μέσω της πολιτικής ελέγχου πρόσβασης που προτείνει, έχει ως άμεσο στόχο τη διασφάλιση της ακεραιότητας (Biba,1977). Επί της ουσίας αυτό το μοντέλο κατασκευάστηκε κατά αντιστοιχία του Bell – LaPadula (Κάτσικας, 2004), καθώς ορίζει βαθμούς ακεραιότητας της πληροφορίας, ανάλογους με τους βαθμούς ευαισθησίας.

Σε αυτό το μοντέλο τα υποκείμενα και τα αντικείμενα ομαδοποιούνται σε διατεταγμένα επίπεδα ακεραιότητας. Ο σχεδιασμός του είναι τέτοιος ώστε τα υποκείμενα να μην μπορούν να αλλοιώσουν τα αντικείμενα τα οποία έχουν καταταχθεί σε υψηλότερο επίπεδο ή και το αντίστροφο.

Σε γενικές γραμμές, η προσπάθεια διασφάλισης της ακεραιότητας των δεδομένων έχει τρεις στόχους:

- Να αποτραπεί η τροποποίηση δεδομένων από μη εξουσιοδοτημένες οντότητες.
- Να αποτραπεί η τροποποίηση δεδομένων από εξουσιοδοτημένες οντότητες χαμηλότερου επιπέδου.
- Να διατηρηθεί η εσωτερική και εξωτερική συνοχή.

2.5.3 Μοντέλο Σινικού Τείχους

Το μοντέλο του Σινικού Τείχους, ή Chinese Wall, όπως συναντάται στη βιβλιογραφία, δημιουργήθηκε με στόχο τη λύση κάποιων προβλημάτων που υπήρχαν συχνά εξ' αιτίας των γενικών μοντέλων και πολιτικών που περιγράφηκαν προηγουμένως. Στόχος των συγγραφέων

της πολιτικής ήταν η αντιμετώπιση είτε ενδεχομένων Σύγκρουσης Συμφερόντων, είτε ακούσιας αποκάλυψης κρίσιμων πληροφοριών από εξωτερικούς συνεργάτες του έργου (Brewer & Nash, 1989).

Ο βασικός τρόπος λειτουργίας του μοντέλου επιτρέπει στους χρήστες να αποκτούν δικαιώματα πρόσβασης σε πόρους και δεδομένα, τα οποία δεν βρίσκονται σε σύγκρουση με οποιαδήποτε άλλη πληροφορία που ήδη κατέχουν. Το μοντέλο του Σινικού Τείχους έχει ως βασικό στόχο τη διαφοροποίηση από τη γενική προσέγγιση και τη δημιουργία ειδικών προσεγγίσεων πολιτικών ελέγχου πρόσβασης, οι οποίες θα δημιουργούνται για συγκεκριμένους οργανισμούς και θα επιδιώκουν λύση συγκεκριμένων προβλημάτων.

Ως εκ τούτου, η πολιτική ασφάλειας του μοντέλου του Σινικού Τείχους βασίζεται σε 3 επίπεδα (Young, 2012):

Αντικείμενα όπως π.χ. τα αρχεία. Αντικείμενα που περιέχουν πληροφορίες σχετικά με ένα μόνο οργανισμό.

Ομάδες Χρηστών που συλλέγουν όλα τα αντικείμενα που αφορούν ένα συγκεκριμένο οργανισμό.

Συγκρουόμενες Κλάσεις, cluster ομάδων χρηστών που αφορά ανταγωνιστικές εταιρείες.

2.5.4 Μοντέλο Graham – Denning

Το μοντέλο αυτό βασίζεται στον Πίνακα Ελέγχου Προσπέλασης. Η ουσία και ο στόχος του μοντέλου είναι η παροχή οδηγιών και κατευθυντήριων γραμμών για την ορθή και ασφαλή δημιουργία και διαγραφή υποκειμένων και αντικειμένων. Η προσέγγιση ασφάλειας αυτού του μοντέλου σχετίζεται με τον καθορισμό ενός βασικού συνόλου δικαιωμάτων σχετικά με το πως τα υποκείμενα θα εφαρμόζουν λειτουργίες ασφάλειας στα αντικείμενα. Όπως αναφέρεται, το μοντέλο χρησιμοποιεί οκτώ βασικές πράξεις ώστε να ελέγξει τη ροή των πληροφοριών μεταξύ των δομικών στοιχείων ενός υπολογιστικού συστήματος (Κάτσικας, 2004).

Οι πράξεις αυτές είναι:

- ✓ Ασφαλής Δημιουργία Αντικειμένου
- ✓ Ασφαλής Δημιουργία Υποκειμένου
- ✓ Ασφαλής Διαγραφή Αντικειμένου
- ✓ Ασφαλής Διαγραφή Υποκειμένου

- ✓ Ασφαλής Παραχώρηση του Δικαιώματος Προσπέλασης της Ανάγνωσης
- ✓ Ασφαλής Παραχώρηση του Δικαιώματος Προσπέλασης της Εκχώρησης
- ✓ Ασφαλής Παραχώρηση του Δικαιώματος Προσπέλασης της Ανάκλησης
- ✓ Ασφαλής Παραχώρηση του Δικαιώματος Προσπέλασης της Μεταβίβασης

2.5.5 Μοντέλο Harrison – Ruzzo - Ulman

Μια φορμαλιστική διατύπωση και παραλλαγή του μοντέλου του Πίνακα Ελέγχου Πρόσβασης, όπως αυτός παρουσιάστηκε από τους Graham και Denning (1972) και όχι της αρχικής μορφή του Lampson (1971), είναι το μοντέλο των Harrison-Ruzzo-Ullman (Harrison, Ruzzo et al., 1976).

Στο μοντέλο αυτό προτείνεται η χρήση γλώσσας για τον ακριβή προσδιορισμό της πολιτικής ασφάλειας που περιέχει τη χρήση δικαιωμάτων προσπέλασης. Σύμφωνα με τη βιβλιογραφία «*Το παρόν μοντέλο ορίζει ως σύστημα προστασίας ένα σύνολο γενικευμένων δικαιωμάτων R και ένα σύνολο εντολών C. Ως κατάσταση του συστήματος προστασίας ορίζεται το τρι-άνυσμα (S, O, A) όπου S το σύνολο των υποκειμένων, O το αντίστοιχο των αντικειμένων και A ο πίνακας προσπέλασης, στον οποίο κάθε γραμμή αντιστοιχεί σε ένα υποκείμενο του S και κάθε στήλη σε ένα αντικείμενο του O*» (Κάτσικας 2004).

2.6 Υποχρεωτικός (Κατά απαίτηση) Έλεγχος Πρόσβασης – Mandatory Access Control MAC

Ο υποχρεωτικός έλεγχος πρόσβασης αποτελεί κορωνίδα της θεματικής ενότητας του access control. Το μοντέλο βασίζεται σε ένα σχήμα διαβάθμισης, το οποίο καθορίζει τη σημαντικότητα που θα είχε για την επιχείρηση/οργανισμό, η μη εξουσιοδοτημένη αποκάλυψη, τροποποίηση, μη διαθεσιμότητα ή καταστροφή κάποιου πληροφοριακού πόρου (Μάγκος, 2011). Το MAC εφαρμόζει τον έλεγχο πρόσβασης σύμφωνα με προκαθορισμένες ετικέτες ασφάλειας που έχουν τόσο τα υποκείμενα του υπολογιστικού συστήματος όσο και τα αντικείμενα (Sandhu et al, 1996). Επί της ουσίας, οι ετικέτες αυτές απεικονίζουν αντίστοιχα την σημαντικότητα των αντικειμένων καθώς και την εξουσιοδότηση (clearance) των υποκειμένων. Βασικό χαρακτηριστικό του MAC είναι ότι, ένα υποκείμενο δεν είναι σε θέση να μεταβιβάσει σε άλλο δικαιώματα και προνόμια πρόσβασης. Η απονομή των προνομίων γίνεται αποκλειστικά από τον διαχειριστή του συστήματος. Η αυστηρότητα του συγκεκριμένου μοντέλου έχει οδηγήσει στην εφαρμογή του, μόνο σε περιβάλλοντα που λειτουργούν με τον τρόπο need-to-know (Κάτσικας, 2004). Το MAC το συναντάμε σε περιβάλλοντα που η αξία της πληροφορίας είναι μεγάλη όπως συμβαίνει για πληροφορίες

κυβερνητικών και στρατιωτικών υπηρεσιών. Στην πορεία του χρόνου έχει αποδειχθεί ότι για την ανάπτυξη πολιτικών υποχρεωτικού ελέγχου πρόσβασης, χρησιμοποιείται το μοντέλο Bell-LaPadula.

Ολοκληρώνοντας τη περιγραφή του MAC θα παρατεθούν τα θετικά και τα αρνητικά στοιχεία του συγκεκριμένου μοντέλου:

- ✓ Ισχυρή Ασφάλεια
- ✓ Ύπαρξη Μοντέλων Υλοποίησης
- × Δύσκολη Διαχείριση
- × Δύσκολη Παραμετροποίηση
- × Δύσκολος Σχεδιασμός

2.7 Διακριτικός Έλεγχος Πρόσβασης – Discretionary Access Control DAC

Ο υποχρεωτικός έλεγχος πρόσβασης ήταν, επί της ουσίας, η πρώτη πολιτική ασφάλειας που υλοποιήθηκε στο χώρο των πληροφοριακών συστημάτων. Η απλότητα των συστημάτων και η μη ύπαρξη πολλών χρηστών, του επέτρεπαν τα πρώτα χρόνια της εφαρμογής του, να μένει μόνος χωρίς «αντίπαλο» στον κόσμο του ελέγχου πρόσβασης. Όμως, όπως περιγράφηκε και στη προηγούμενη παράγραφο το MAC έφερε και αρνητικά στοιχεία, τα οποία δεν μπορούσαν να είναι συμβατά με τα πληροφοριακά συστήματα, εκτός αυτών που αφορούσαν το στρατιωτικό ή το κυβερνητικό περιβάλλον. Έτσι δημιουργήθηκε ένα νέο μοντέλο, αυτό του Διακριτικού Ελέγχου Πρόσβασης.

Το μοντέλο αυτό, βασίζεται στην έννοια της κατοχής-ιδιοκτησίας. Η μεταβίβαση-ανάκληση των δικαιωμάτων πρόσβασης είναι στην ευχέρεια των μεμονωμένων χρηστών-υποκειμένων του συστήματος, οι οποίοι «κατέχουν» ένα αντικείμενο (πληροφοριακό πόρο) του συστήματος (Μάγκος, 2011). Η απλότητα και η ευχρηστία του συγκεκριμένου μοντέλου, γρήγορα το οδήγησαν στην κορυφή και στην υιοθέτησή του από τους κατασκευαστές λειτουργικών συστημάτων. Χαρακτηριστικό παράδειγμα είναι τα λειτουργικά συστήματα Unix και Windows, τα οποία ακόμα και σήμερα λειτουργούν με αυτό τον τρόπο. Το DAC, υλοποιείται κυρίως μέσω των λιστών ελέγχου προσπέλασης (ACLs). Με την πάροδο του χρόνου οι πολιτικές DAC κατηγοριοποιήθηκαν, για την παροχή μεγαλύτερης ευελιξίας αλλά και την αύξηση της ασφάλειας, στις εξής επιμέρους πολιτικές:

- Strict DAC (Αυστηρό)
- Liberal DAC (Φιλελεύθερο)

Όμως, το DAC είχε και έχει αρκετά αδύνατα σημεία. Σημεία που για την Ασφάλεια της πληροφορίας είναι πολύ σημαντικά. Το DAC παρουσιάζει αδυναμία στη διαδικασία ασφάλισης της ροής της πληροφορίας, καθώς τόσο η παραχώρηση δικαιωμάτων πρόσβασης όσο και η ανάκληση αυτών είναι μεταβατική. Είναι μεγάλο μειονέκτημα ο χρήστης 1 να παραχωρεί στο χρήστη 2 δικαίωμα πρόσβασης στο αρχείο Χ και ο χρήστης 2 να παραχωρεί το δικαίωμα του στο χρήστη 3, χωρίς να υπάρχει καμία σχετική ενημέρωση προς τον χρήστη 1. Επιπλέον σημειώνεται ότι το DAC παρουσιάζει τρωτότητα απέναντι σε επιθέσεις Trojan Horse[1] (Mao Z., et al, 2009). Ολοκληρώνοντας τη περιγραφή και αυτής της πολιτικής, παρατίθενται τα βασικά πλεονεκτήματα και μειονεκτήματά της.

- ✓ Μεγάλη Ευελιξία
- ✓ Πολλά παραδείγματα χρήσης
- × Κενά Ασφαλείας

2.8 Σύνοψη

Το κεφάλαιο 2 αποτελεί τον πρώτο θεμέλιο λίθο της εργασίας. Παρουσιάστηκαν τα βασικά μοντέλα και οι βασικές αρχές του ελέγχου πρόσβασης. Όπως έγινε αντιληπτό, τα μειονεκτήματα που έχουν τα MAC και DAC δεν μπορούσαν να μην απασχολήσουν τη διεθνή κοινότητα έρευνας στον τομέα του Ελέγχου Πρόσβασης. Τα κενά αυτά έρχεται να συμπληρώσει μια απλή και ταυτόχρονα σύνθετη ιδέα. Ο Έλεγχος Πρόσβασης Βασισμένος σε Ρόλους, αντικείμενο που θα μας απασχολήσει ως προς το θεωρητικό του μέρος στο επόμενο κεφάλαιο και ως προς το πρακτικό του μέρος, στο μεθεπόμενο. Τα συμπεράσματα που προκύπτουν από το δεύτερο κεφάλαιο στο οποίο καταγράφεται η βιβλιογραφική επισκόπηση, αποτυπώνουν μια γνώριμη αλήθεια για τον κόσμο της Πληροφορικής: **Η εξέλιξη είναι συνεχής, όμως οι λύσεις του παρελθόντος μπορούν να δώσουν λύσεις σε ζητήματα του σήμερα, καθώς τα δομικά στοιχεία δεν αλλάζουν ποτέ.**

[1] Ο δούρειος ίππος (trojan horse ή απλά trojan) είναι ένα κακόβουλο πρόγραμμα που ξεγελάει τον χρήστη και τον κάνει να πιστεύει ότι εκτελεί κάποια χρήσιμη λειτουργία ενώ στα κρυφά εγκαθιστά στον υπολογιστή του άλλα κακόβουλα προγράμματα.

3 Έλεγχος Πρόσβασης Βασισμένος σε Ρόλους

3.1 Εισαγωγή και Ορισμός

Σε αυτό το κεφάλαιο θα παρουσιαστεί και θα αναλυθεί ο Έλεγχος Πρόσβασης Βασισμένος σε Ρόλους.

Έχοντας ορίσει, τι είναι τόσο ο έλεγχος πρόσβασης, όσο και το τι είναι ρόλος στο προηγούμενο κεφάλαιο της ΜΔΕ, θα πρέπει τώρα να δοθεί ορισμός και για το RBAC. Οι ορισμοί που συναντώνται στη διεθνή βιβλιογραφία είναι πολλοί, όμως ένας πληρέστατος ορισμός δίνεται τον καθηγητή Σωκράτη Κάτσικα « *Η πολιτική Ρόλο-Κεντρικού Ελέγχου Πρόσβασης βασίζει τις αποφάσεις για τη χορήγηση άδεις προσπέλασης στις ενέργειες που ένας χρήστης επιτρέπεται να εκτελέσει, ενώ δεν αφήνεται στη διακριτική ευχέρεια των χρηστών η μεταβίβαση δικαιωμάτων προσπέλασης σε άλλους χρήστες.*» (Κάτσικας, 2004)

Το RBAC αναφέρεται για πρώτη φορά από τους Ferraiolo και Kuhn το 1992 και στόχο είχε την επίλυση σύνθετων προβλημάτων ασφάλειας.Τα προβλήματα προέρχονταν κυρίως από την πολυπλοκότητα διαχείρισης των χρηστών, που αντιμετώπιζαν μεγάλοι οργανισμοί (Ferraiolo et al, 1995). Όπως αναφέρουν οι ίδιοι στο paper που δημοσιεύθηκε τότε, το RBAC είναι μια διαφορετικού τύπου εφαρμογή του MAC, πολύ πιο αποτελεσματική και ταυτόχρονα, ευκολότερα διαχειρίσιμη. Σήμερα, η χρήση του συναντάται συχνά, καθώς όλο και περισσότεροι κατασκευαστές λογισμικού προσπαθούν να δημιουργήσουν και να εξελίξουν τα προϊόντα τους με βάση τις κατευθυντήριες γραμμές που δίνονται από το RBAC.

Τι νέο λοιπόν έφερε το RBAC; Η βασική αρχή του RBAC είναι ότι: Στον έλεγχο πρόσβασης βασισμένο σε ρόλους, **κεντρικό στοιχείο παύει να είναι ο χρήστης**, όπως ήταν στα προηγούμενα μοντέλα MAC και DAC, και **πλέον είναι ο ρόλος.**

Όλες οι ενέργειες που γίνονται είναι συνυφασμένες με ρόλους. Στους χρήστες ανατίθενται ρόλοι, στους οποίους ρόλους, έχουν ήδη ανατεθεί προνόμια και δικαιώματα. Ο χρήστης μπορεί να ανήκει σε περισσότερους από έναν ρόλους. Η διάκριση των ρόλων καθώς και η τροφοδότησή τους με τα κατάλληλα δικαιώματα πρέπει να γίνεται με βάση τις ανάγκες του οργανισμού και το έργο που επιθυμεί να επιτελεί ο κάθε ρόλος στην παραγωγική διαδικασία του οργανισμού.

Το RBAC είναι μια πολιτική η οποία χαρακτηρίζεται ως ουδέτερη (policy neutral) και υποστηρίζει τρεις βασικές αρχές ασφάλειας, όπως αυτές ορίστηκαν από τον Saltzer, και περιγράφηκαν στο προηγούμενο κεφάλαιο. Αυτές είναι:

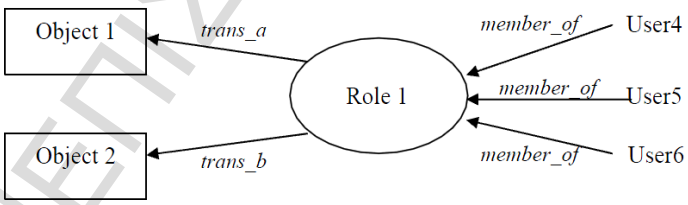
- η αρχή των ελαχίστων προνομίων,
- η αρχή του διαχωρισμού καθηκόντων και
- η αρχή της αφαίρεσης δεδομένων.

Η αρχή των ελαχίστων προνομίων ικανοποιείται, επειδή για τη λειτουργία του συστήματος απαιτείται να δοθούν στο χρήστη ελάχιστα προνόμια, καθώς αυτά δίνονται στους ρόλους.

Ο διαχωρισμός καθηκόντων αποτελεί ολόκληρη ενότητα του RBAC και θα περιγραφεί αναλυτικά παρακάτω. Εδώ αρκεί να τονιστεί ότι το ρολο-κεντρικό μοντέλο επιτρέπει την ύπαρξη αμοιβαίως αποκλειόμενων ρόλων σε ζητήματα ευαίσθητων εργασιών.

Τέλος, data abstraction, επιτυγχάνεται χάρη στη δυνατότητα που δίνει το RBAC, στους ρόλους, ως αφηρημένες έννοιες, να ξεφύγουν από τα κλασσικά δικαιώματα πρόσβασης και να δημιουργούνται νέα, ανάλογα με τις πραγματικές ανάγκες του συστήματος.

Η επόμενη εικόνα αποτυπώνει το βασικό τρόπο λειτουργίας του RBAC. Ο ρόλος Role 1 έχει δικαιώματα πρόσβασης στα αντικείμενα Object 1 & Object 2. Οι χρήστες User 4, User 5 & User 6 είναι μέλη του Role 1. Αυτό σημαίνει ότι αυτόματα, μόλις τους ανατεθεί ο ρόλος, αποκτούν τα δικαιώματα πρόσβασης του. Αν πάλι, αλλάξουν οι ανάγκες του οργανισμού, οι διαχειριστές αλλάζουν τα δικαιώματα του ρόλου, και όχι του κάθε χρήστη χωριστά. Αν ένας χρήστης, πάψει να είναι εργαζόμενος ή αλλάξει τμήμα, απλώς διαγράφεται από μέλος του ρόλου, χωρίς αυτό να βλάψει τόσο την παραγωγική διαδικασία όσο και την ασφάλεια του συστήματος.

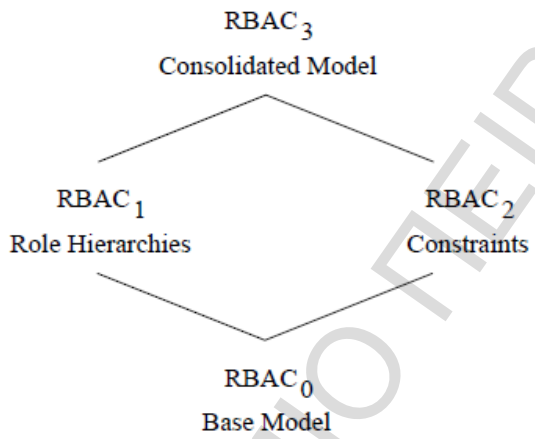


Εικόνα 4 Ανάθεση Χρηστών & Δικαιωμάτων στους Ρόλους (Ferrailo & Kuhn, 1992)

3.2 Μοντέλα

Ο Ravi Sahndu το 1996, παρουσίασε τα τέσσερα μοντέλα τα οποία αναπαριστούν τις βασικές αρχές του RBAC (Sahndu et al, 1996). Το σύνολο των μοντέλων αυτών ονομάστηκε RBAC Models 96.

Το $RBAC_0$ είναι το βασικό μοντέλο πού πρέπει να ικανοποιείται σε κάθε σύστημα που υποστηρίζει το μοντέλο RBAC. Τα $RBAC_1$ και $RBAC_2$ εμπεριέχουν και τα δυο το μοντέλο $RBAC_0$ αλλά προσθέτουν σ' αυτό επιπλέον στοιχεία. Το $RBAC_1$ θέτει την αρχή της ιεραρχίας των ρόλων, δηλαδή τη δυνατότητα των ρόλων να κληρονομούν άδειες από άλλους ρόλους. Το $RBAC_2$ εισάγει και την έννοια των περιορισμών (constraints) σχετικά με το πώς μπορεί να οριστεί κάποιο συγκεκριμένο RBAC μοντέλο. Τέλος το $RBAC_3$ μοντέλο είναι ο συνδυασμός των $RBAC_1$ και $RBAC_2$, και εμμέσως του $RBAC_0$ μοντέλου.



Εικόνα 5 RBAC 96 Models

3.3 Πρότυπο κατά ANSI/INCITS 359-2004

Τον Απρίλιο του 2004 το American National Standards Institute ενέκρινε και δημοσίευσε ως πρότυπο 359-2004 την δημοσίευση του NIST για το RBAC (ANSI, 2004). Το πρότυπο βασίζεται αρχικά στο RBAC 96 μοντέλο και καθορίζει τα βασικά συστατικά του RBAC. Αυτά δεν είναι άλλα από τους ρόλους, τους χρήστες, τα δικαιώματα, τις λειτουργίες και τα αντικείμενα. Τα βασικά συστατικά του προτύπου είναι 4:

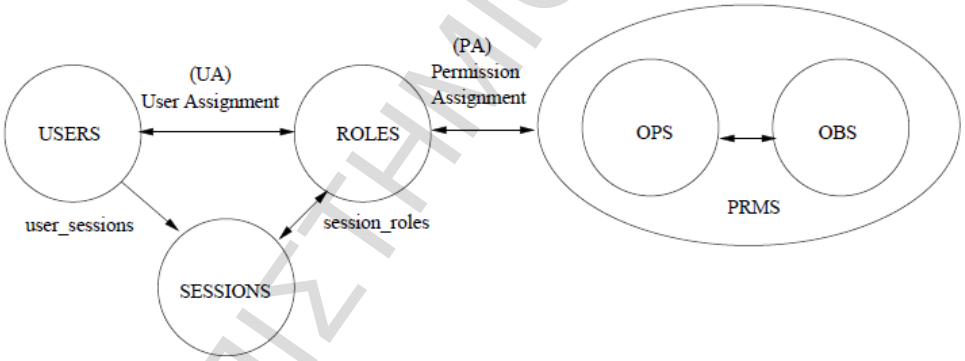
- Core RBAC (Πυρήνας ή Βασικό RBAC)
- Hierarchical RBAC (Ιεραρχικό RBAC)
- Static Separation of Duty Relations (Στατικός διαχωρισμός καθηκόντων)
- Dynamic Separation of Duty Relations (Δυναμικός διαχωρισμός καθηκόντων)

3.3.1 Βασικό RBAC

Το Core RBAC παρουσιάζει τα βασικά χαρακτηριστικά που πρέπει να ικανοποιεί ένα RBAC σύστημα. Η συμμόρφωση με αυτό αποτελεί την ελάχιστη απαίτηση ώστε να μπορεί ένα σύστημα να ισχυρίζεται ότι παρέχει έλεγχο πρόσβασης βασισμένο σε ρόλους. Επί της ουσίας είναι το $RBAC_0$ όπως ορίστηκε από τον Sahndu. Το Core RBAC περιέχει τα ακόλουθα στοιχεία (Ferraiolo et al, 2001):

- U, R, P, and S (χρήστες, ρόλους, προνόμια και συνεδρίες (sessions) αντίστοιχα)
- PA ⊆ P × R, πολλαπλή (many-to-many) σχέση ανάθεσης προνομίων σε ρόλους
- UA ⊆ U × R, πολλαπλή (many-to-many) σχέση ανάθεσης χρηστών σε ρόλους

- U, R, P, and S (χρήστες, ρόλους, προνόμια και συνεδρίες (sessions) αντίστοιχα)
- PA ⊆ P × R, πολλαπλή (many-to-many) σχέση ανάθεσης προνομίων σε ρόλους
- UA ⊆ U × R, πολλαπλή (many-to-many) σχέση ανάθεσης χρηστών σε ρόλους



Εικόνα 6 Core RBAC (Ferraiolo et al, 2001)

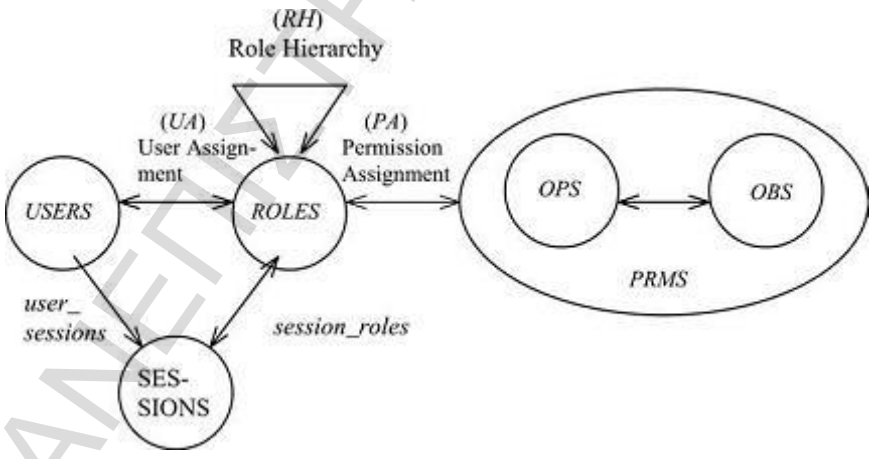
Στο RBAC, συνήθως η έννοια των χρηστών αφορά σε υποκείμενα αλλά αυτό δε σημαίνει ότι ένα αντικείμενο, πχ πρόγραμμα, δεν μπορεί να θεωρηθεί user. Τα προνόμια δεν είναι τίποτα διαφορετικό από την αποτύπωση της εξουσιοδοτημένης πρόσβασης σε κάποιο αντικείμενο που θα αναγνωρίζει το σύστημα το οποίο χρησιμοποιεί το μοντέλο αυτό. Τελευταίο βασικό στοιχείο του core RBAC είναι η έννοια της συνεδρίας. Συνεδρία είναι η σύνδεση ενός χρήστη με ένα σύνολο από ρόλους. Η συνεδρία αρχικοποιείται όταν ο χρήστης ζητήσει ένα αντικείμενο και ορίσει ένα σύνολο ρόλων. Υπάρχει στη διεθνή βιβλιογραφία και ένα στοιχείο το οποίο μπορεί να ενταχθεί στο βασικό πυρήνα του ρόλο-κεντρικού μοντέλου. Αυτό το στοιχείο είναι η έννοια του καθήκοντος. Η έννοια αυτή υποδηλώνει τις υποχρεώσεις του

ρόλου ώστε να μπορεί να λειτουργήσει φυσιολογικά και να φέρει σε πέρας μια ορισμένη εργασία.

3.3.2 Ιεραρχικό RBAC

Το Ιεραρχικό RBAC εισάγει την έννοια των Ιεραρχιών. Ιεραρχία είναι μαθηματικά, μια μερική ακολουθία με στόχο τον καθορισμό μίας συσχέτισης μεταξύ των ρόλων, όπου οι ανώτεροι ρόλοι αποκτούν δικαιώματα και άδειες από τους κατώτερους και οι κατώτεροι αποκτούν τους χρήστες των ανώτερων ρόλων. Φορμαλιστικά το Ιεραρχικό RBAC, που επί της ουσίας είναι το $RBAC_1$, ορίζεται από τους παρακάτω κανόνες (Ferraiolo et al, 2001):

- Τα σύνολα U, R, P, S, PA, UA και η συνάρτηση user() είναι ορισμένα όπως ακριβώς και στο $RBAC_0$ μοντέλο
- $RH \subseteq R \times R$, είναι η μερική διάταξη πάνω στο R και ονομάζεται ιεραρχία ρόλων ή σχέση κυριαρχίας ρόλων (role dominance relation) που επίσης μπορεί να αποδοθεί και ως $\geq$
- $roles : S \rightarrow 2^R$, έχει τροποποιηθεί από το $RBAC_0$ μοντέλο ώστε $roles(si) \subseteq \{r \mid (\exists\ r' \geq r)[(user(si), r') \in UA]\}$ (που μπορεί να αλλάζει με τον χρόνο) και κάθε συνεδρία si έχει άδειες $\bigcup_{r \in roles(si)} \{p \mid (\exists\ r'' \leq r)\ [(p,r'') \in PA]\}$.



Εικόνα 7 Hierarchical RBAC (Ferraiolo et al, 2001)

Ο τρόπος λειτουργίας του Ιεραρχικού RBAC διευκολύνει το σχεδιασμό της πολιτικής του ελέγχου πρόσβασης σε πολύπλοκους οργανισμούς. Μάλιστα διαπιστώνεται ότι είναι ιδιαίτερα χρήσιμο όταν υπάρχουν αμοιβαίως αποκλειόμενοι ρόλοι. Το πρακτικό μέρος της παρούσας ΜΔΕ εφαρμόζει πλήρως τις κατευθυντήριες οδηγίες του Ιεραρχικού RBAC, και οι

ανώτεροι ιεραρχικά ρόλοι κληρονομούν τα δικαιώματα των υφισταμένων. Αξίζει να σημειωθεί ότι το συγκεκριμένο μοντέλο χωρίζεται στις εξής δύο υποκατηγορίες:

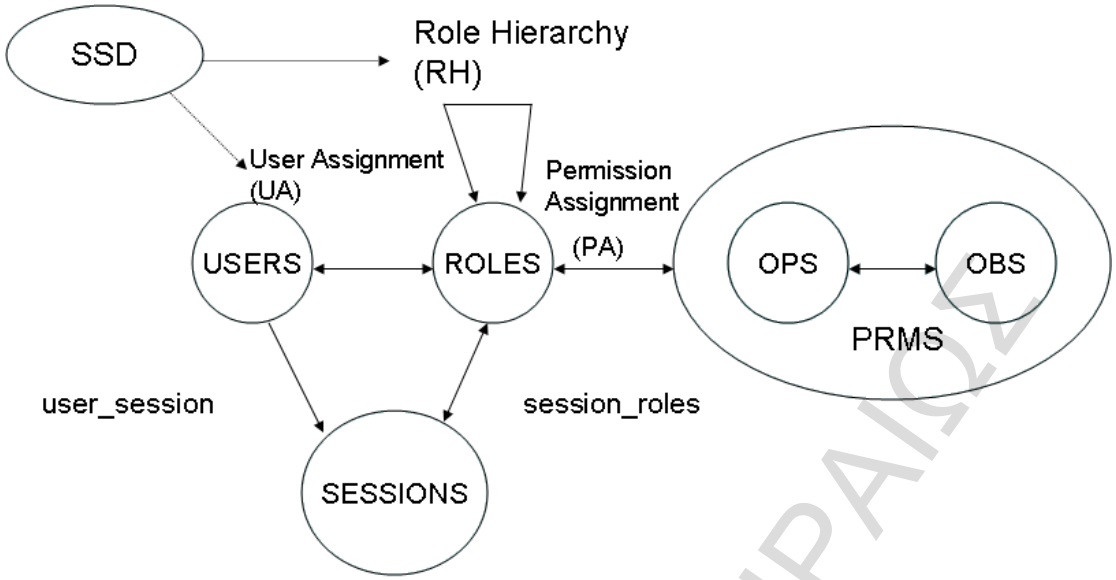
- General Hierarchical RBAC
- Limited Hierarchical RBAC

3.3.3 Στατικός Διαχωρισμός Καθηκόντων

Ο Στατικός Διαχωρισμός Καθηκόντων είναι ένας περιορισμός και επί της ουσίας, δεν επιτρέπει σε ένα χρήστη, να κατέχει παραπάνω από έναν ρόλο (σε περίπτωση που ο σκοπός των ρόλων αλληλοσυγκρούεται) και ο ρόλος αυτός να κατέχει παραπάνω δικαιώματα από αυτά που χρειάζεται για να φέρει σε πέρας την εργασία του (Simon & Zukro, 1997).

Η χρήση του Στατικού Διαχωρισμού Καθηκόντων μειώνει σημαντικά την έκθεση του οργανισμού σε κινδύνους που προέρχονται από εξαπάτηση, μη εξουσιοδοτημένη πρόσβαση και σύγκρουση συμφερόντων. Επίσης, εξασφαλίζει ότι κρίσιμες επιχειρησιακές διαδικασίες δεν θα βασίζονται μόνο σε ένα άτομο.

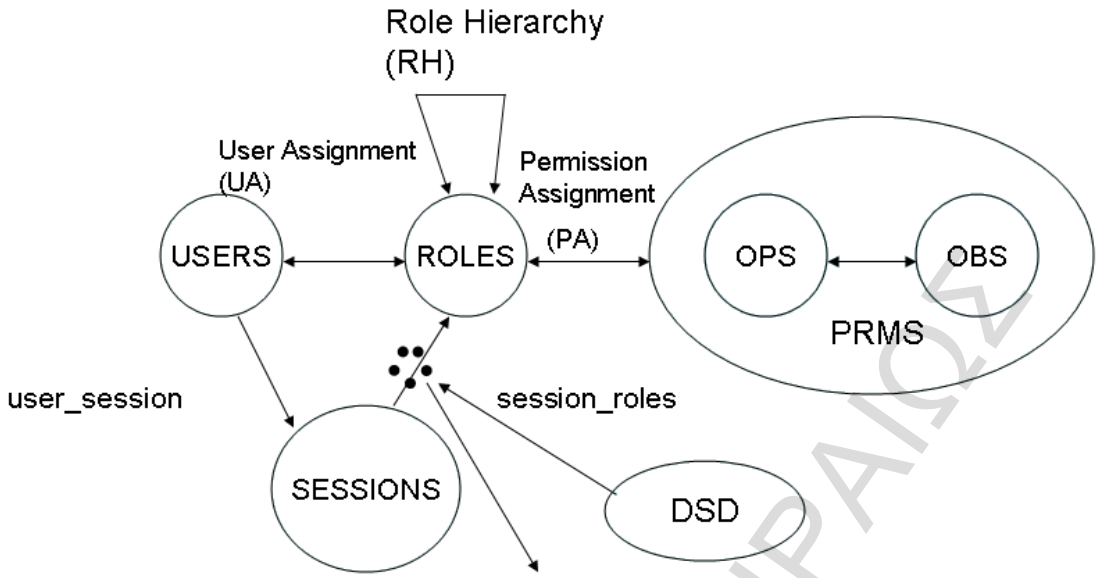
Ο ρολο-κεντρικός έλεγχος πρόσβασης, όπως αναφέρθηκε και στην παράγραφο 3.1, έχει στη φύση του και στον τρόπο λειτουργίας του την υποστήριξη ως προς την εφαρμογή του στατικού διαχωρισμού καθηκόντων. Οι ρόλοι ορίζονται σύμφωνα με τα δικαιώματα πρόσβασης και τις απαγορεύσεις πρόσβασης. Μπορεί να επιβληθεί μια πολιτική όπου ένας ρόλος δεν μπορεί ταυτόχρονα να αιτείται και να εγκρίνει δαπάνες. Ομοίως το άτομο το οποίο ανήκει στο ρόλο που διενεργεί αλλαγές και είναι υπεύθυνος για τη λειτουργία του firewall, δεν μπορεί παράλληλα να υπηρετεί το ρόλο του εσωτερικού επιθεωρητή. Αυτός είναι στην πράξη ο στατικός διαχωρισμός καθηκόντων, ένας από τους δύο που μπορούν να εφαρμοστούν στο RBAC. Ο άλλος είναι ο δυναμικός διαχωρισμός καθηκόντων που θα μελετηθεί στην επόμενη παράγραφο. Η πολιτική αυτή καθορίζεται σε κεντρικό επίπεδο και εν συνεχεία συγκεκριμενοποιείται για κάθε ρόλο.



Εικόνα 8 Static Separation of Duty Relations (Ferraiolo et al, 2001)

3.3.4 Dynamic Separation of Duty (DSD)

Στην προηγούμενη παράγραφο περιγράφηκε ο απλός διαχωρισμός καθηκόντων. Ο Δυναμικός Διαχωρισμός Καθηκόντων, ανατρέπει τα δεδομένα που παρουσιάστηκαν στην προηγούμενη παράγραφο και επιτρέπει στον ίδιο χρήστη να βρίσκεται σε αλληλοσυγκρουόμενους ρόλους. Όμως σύμφωνα με όσα περιγράψαμε αυτό θα έπρεπε να εμποδίζει τη διενέργεια του διαχωρισμού καθηκόντων και να αφήνει κενά ασφαλείας στο σύστημα. Κάτι τέτοιο όμως δεν συμβαίνει στη πράξη, καθώς εδώ έρχεται και παρεμβαίνει η έννοια του Δυναμικού Διαχωρισμού Καθηκόντων, η οποία επιτρέπει σε ένα άτομο να κατέχει δύο αμοιβαίως αποκλειόμενους ρόλους αλλά δεν μπορεί να χρησιμοποιήσει και τους δύο ταυτόχρονα. Στην πράξη, έστω ότι ένας χρήστης κατέχει το ρόλο του αιτούντα δαπανών και του εγκρίνοντα δαπάνες, και αιτείται μιας Χ δαπάνης. Η εφαρμογή των αρχών του Δυναμικού Διαχωρισμού Καθηκόντων δεν θα του επιτρέψει να εγκρίνει τις δικές του αιτήσεις παρά μόνο άλλων μελών αυτού του ρόλου. Για τις δικές του αιτήσεις δαπανών απαιτείται έγκριση από διαφορετικό χρήστη που ανήκει στο ρόλο των εγκρινόντων. **Στο Δυναμικό Διαχωρισμό Καθηκόντων αυτό που αλλάζει είναι το ότι ο διαχωρισμός των καθηκόντων σχετίζεται άμεσα με την έννοια της συνεδρίας. Σε αυτήν τη πολιτική οι περιορισμοί είναι έγκυροι και έχουν ισχύ για την παρούσα συνεδρία από τη στιγμή που ενεργοποιούνται.**



Εικόνα 9 Dynamic Separation of Duty (DSD) (Ferraiolo et al, 2001)

3.4 Επεκτάσεις RBAC

Το RBAC μετράει 21 χρόνια ζωής και είναι ένα μοντέλο το οποίο δεν έχει πάψει παρόλη την ηλικία του να εξελίσσεται. Υπάρχουν δεκάδες επεκτάσεις του μοντέλου οι οποίες είτε ανέδειξαν διάφορες αδυναμίες και είχαν ως στόχο να τις καλύψουν, είτε απλώς πρόσθεσαν διάφορα χαρακτηριστικά ώστε να κάνουν το μοντέλο πιο λειτουργικό. Παρακάτω θα παρουσιαστούν οι πιο ενδιαφέρουσες επεκτάσεις του μοντέλου.

3.4.1 TRBAC Temporal RBAC

Αρχικά, η εκκίνηση δίνεται από την επέκταση που εντάσσει στο μοντέλο την έννοια των χρονικών περιορισμών. Δημιουργείται το 2001 και εισάγει την έννοια των ενεργών ρόλων (Bertino et al, 2001). Τι ακριβώς όμως σημαίνει η έννοια ενεργός ρόλος; Σημαίνει ότι μπαίνει ένας νέος περιορισμός, χρονικής φύσεως, ο οποίος άλλοτε επιτρέπει και άλλοτε απαγορεύει στο ρόλο να είναι ενεργός σύμφωνα με τους χρονικούς περιορισμούς που έχουν τεθεί στο Π.Σ. όπου εφαρμόζεται το RBAC. Η συγκεκριμένη επέκταση μπορεί να λειτουργήσει σε χώρους όπου υπάρχει συνεχόμενη χρήση ενός Π.Σ. με βάση ένα σταθερό και ορισμένο πρόγραμμα. Για παράδειγμα, μπορεί να λειτουργήσει σε ένα σχολικό περιβάλλον με τους ρόλους καθηγητών, μαθητών και υπεύθυνων εργαστηρίων να είναι ενεργοί τις καθημερινές ημέρες από τις 8:00 έως τις 14:00.

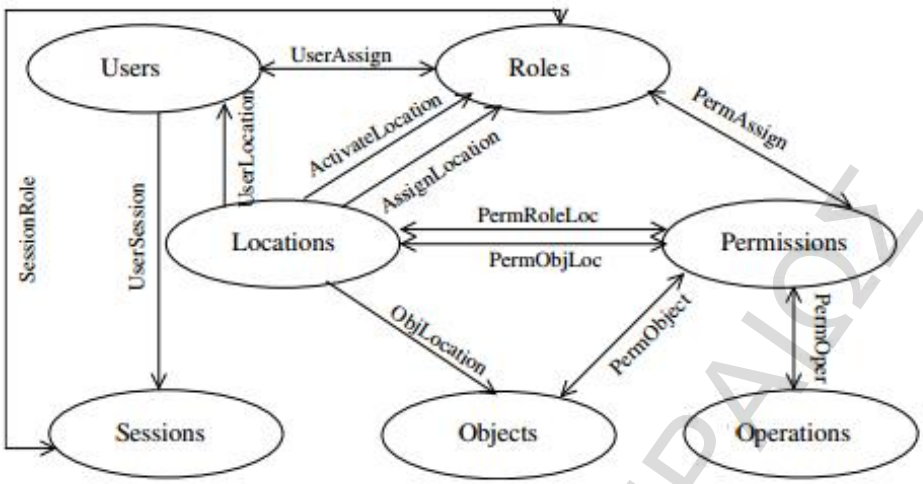
Σε αυτό το σημείο αξίζει να αναφερθεί ότι η συγκεκριμένη επέκταση παρουσίασε μεγάλο ενδιαφέρον, τόσο που κατά την πάροδο του χρόνου υπήρξαν αρκετές και αξιόλογες επεκτάσεις της.

3.4.2 Generalized Role-Based Access Control

Μια ακόμα επέκταση του RBAC, που παρουσιάζει αρκετό ενδιαφέρον, είναι αυτή του γενικευμένου ρόλο-κεντρικού μοντέλου. Προτάθηκε το 2000 για να καλύψει μελλοντικές ανάγκες και κυρίως τις ανάγκες ενός «έξυπνου» σπιτιού (Covington et al, 2000). Από τη στιγμή λοιπόν που η πηγή της έμπνευσης ήταν η λειτουργία του smart home, προέκυψε και η ανάγκη για αλλαγή στο μέχρι τότε τρόπο λειτουργίας και σχεδιασμού του RBAC. Το κλασσικό RBAC, όπως αναλύθηκε στις προηγούμενες ενότητες, λειτουργεί με εκχώρηση ρόλων στα υποκείμενα. Το GRBAC έρχεται να προτείνει την παράλληλη ανάθεση προνομίων και στα αντικείμενα. Στην πράξη, στο παράδειγμα του smart home, αυτό έχει σαν αποτέλεσμα να ομαδοποιούνται τα αντικείμενα σε ρόλους. Παραδείγματος χάρη, τα πλυντήρια ρούχων και πιάτων καθώς και ο θερμοσίφωνας εντάσσονται στις ενεργοβόρες συσκευές οι οποίες θα είχαν δικαίωμα στην ηλεκτροδότηση μόνο τις ώρες του μειωμένου τιμολογίου, ενώ οι ψυχαγωγικές συσκευές θα είχαν δικαίωμα λειτουργίας μόνο κατά τις απογευματινές ώρες. Όμως, αυτή δεν είναι η μόνη καινοτομία που φέρνει αυτή η γενικευμένη μορφή του RBAC. Εισάγεται και μια ακόμα έννοια, αυτή του περιβάλλοντος. Με βάση αυτή την έννοια μπορεί να προσαρμοστεί ο ρόλος των αντικειμένων ανάλογα με το περιβάλλον που εργάζεται το υποκείμενο προσαρμόζοντας, για παράδειγμα τη λειτουργία των συσκευών κλιματισμού ανάλογα με τη περίοδο, τη θερμοκρασία και τη μέρα είναι μέσα στη βδομάδα.

3.4.3 A Location-Aware RBAC

Το RBAC μπορεί να λειτουργήσει με ευχέρεια σε οποιοδήποτε κλασσικό δικτυακό περιβάλλον, και να αποτελέσει τη βάση των εφαρμογών του ελέγχου πρόσβασης. Σήμερα, με τη ραγδαία ανάπτυξη των ασύρματων και κινητών δικτύων έχει γεννηθεί και η ανάγκη του ελέγχου πρόσβασης βάσει τοποθεσίας. Το 2006 δημοσιεύθηκε άλλη μια επέκταση του RBAC η οποία ονομάστηκε Location Aware RBAC. Η χρήση των πληροφοριών που προκύπτουν από την τοποθεσία, μπορούν να χρησιμοποιηθούν σε εφαρμογές που διενεργούν έλεγχο πρόσβασης. Ειδικότερα στο πεδίο των κρίσιμων υποδομών και των κρίσιμων εφαρμογών, μια τέτοια επέκταση του μοντέλου , θα μπορούσε να προσφέρει ακόμα μεγαλύτερη ασφάλεια (Indrakshi et al, 2006). Η αρχιτεκτονική του μοντέλου παρουσιάζεται στην παρακάτω εικόνα στην οποία απεικονίζεται η έννοια της τοποθεσίας με όλα τα βασικά συστατικά του RBAC. Για την εφαρμογή του Location-Aware RBAC στη πράξη χρειάζονται πέρα από ασύρματο δίκτυο και ειδικές συσκευές που θα καταγράφουν την τοποθεσία του χρήστη. Στη συνέχεια, το σύστημα, αυτοματοποιημένα θα εντοπίζει ποιο ρόλο υπηρετεί ο χρήστης και αν βάσει της τοποθεσίας του θα ενεργοποιείται ή όχι η πρόσβαση στα δικαιώματα του ρόλου του.



Εικόνα 10 Αρχιτεκτονική του Location Aware RBAC (Πηγή: Indrakshi et al, 2006)

3.4.4 Context-Aware Role Based Access Control

Τελευταία επέκταση στην οποία θα γίνει αναφορά, είναι αυτή του Context – Aware Role Based Access Control. Η επέκταση αυτή παρουσιάζει ιδιαίτερο ενδιαφέρον. Αν και δομικά δεν παρουσιάζει ομοιότητες με τη μελέτη περίπτωσης που θα παρουσιαστεί στο επόμενο κεφάλαιο κρίνεται σκόπιμο να γίνει αναφορά και σε αυτή την ιδέα. Το Context Aware RBAC διαφέρει αρκετά από το κλασσικό RBAC. Πρώτη και κύρια διαφορά είναι ο τρόπος με τον οποίο ορίζεται ο ρόλος. Στο CARBAC ο ρόλος αποκτά δυναμικά στοιχεία, αντίθετα με τη σταθερότητα που παρουσιάζει στο πρότυπο του ANSI. Σε αυτό το μοντέλο ο ρόλος ορίζεται σαν μέρος της εφαρμογής που σχεδιάζεται (Kulkarni, 2008). Πρόκειται για ένα μοντέλο το οποίο μπορεί να βρει εφαρμογή σε πεδία όπου ένα σύστημα χρησιμοποιείται για πολλές εργασίες.

3.5 Βέλτιστες Πρακτικές για την Εφαρμογή του RBAC στο επίπεδο της επιχείρησης.

Όπως γίνεται αντιληπτό, η εφαρμογή του RBAC είναι ένα σύνθετο project με συγκεκριμένες δυσκολίες, τόσο στο σχεδιασμό όσο και στην υλοποίηση. Αν οι επιμέρους δυσκολίες δεν αντιμετωπιστούν με επιτυχία, τότε η εφαρμογή μπορεί να μην έχει επιτυχές αποτέλεσμα.. Για αυτό το λόγο στη συνέχεια παρατίθενται οι πρακτικές με την εφαρμογή των οποίων διασφαλίζονται τα βέλτιστα αποτελέσματα (Simeio Solutions, 2009).

1. Καλώς Ορισμένοι Στόχοι

2. Διενέργεια Ανάλυσης Απαιτήσεων/Role Engineering

3. Καθιέρωση Διαχείρισης του Κύκλου Ζωής των Ρόλων

4. Ορισμός Εφικτού Χρονοδιαγράμματος

5. Προσδιορισμός Ρόλων και Αρμοδιοτήτων

6. Κατάλληλη επιλογή συνεργατών

7. Σχεδιασμός Υλοποίησης

8. Ύπαρξη Κέντρου Υποστήριξης και Συνεργασίας των Εργαζόμενων Ομάδων

9. Τήρηση του Χρονοδιαγράμματος

10. Κατάλληλη προετοιμασία, ώστε να υπάρχει η δυνατότητα υλοποίησης μη αναμενόμενων αλλαγών.

3.6 Πλεονεκτήματα

Ο έλεγχος πρόσβασης αποτελεί μια επίπονη διαδικασία για τους πολύπλοκους οργανισμούς. Ήταν και είναι μια επιχειρησιακή λειτουργία, η οποία κοστίζει σε χρόνο και προσπάθεια προκειμένου να σχεδιαστεί σωστά , να εφαρμοστεί και να συντηρείται με τρόπο που να εγγυάται την ομαλή λειτουργία. Έτσι, οι οργανισμοί καλούνται να βρουν τη βέλτιστη λύση. Στους μεγάλους και πολύπλοκους οργανισμούς, ως ενδεδειγμένη λύση θεωρείται η υλοποίηση πολιτικών Ελέγχου Πρόσβασης που είναι βασισμένες στο RBAC.

Η χρήση συστημάτων που εφαρμόζουν την πολιτική του RBAC, επιφέρει οφέλη και πλεονεκτήματα στους οργανισμούς. Αυτά συνοψίζονται στη λίστα που ακολουθεί:

- ✓ Αύξηση Ασφάλειας: Τα προφίλ των ρόλων και τα δικαιώματα αυτών μπορούν άμεσα, με την απαιτούμενη βέβαια εξουσιοδότηση, να παραμετροποιηθούν από τους διαχειριστές. Η αλλαγή πολιτικών και η ενημέρωση στα προφίλ των ρόλων έχει σαν αποτέλεσμα τη διατήρηση υψηλού επιπέδου ασφάλειας.
- ✓ Διασφάλιση Ασφάλειας σε Πολύπλοκους Οργανισμούς: Η συμμόρφωση με τις πολιτικές του ελέγχου πρόσβασης βασισμένου σε ρόλους, προσφέρει στους Πολύπλοκους Οργανισμούς Ασφάλεια μέσω της δημιουργίας των κατάλληλων ρόλων και τη διαχείριση αυτών από μικρή ομάδα διαχειριστών. Οι αλλαγές εφαρμόζονται άμεσα αν υποτεθεί ότι ο Οργανισμός αλλάξει την πολιτική Ασφάλειάς του.

- ✓ Μείωση Πολυπλοκότητας: Απαιτούνται λιγότεροι διαχειριστές και εξασφαλίζεται μεγάλη ευκολία στις αλλαγές
- ✓ Μείωση Κόστους (O'Connor & Loomis, 2010): Μειώνονται οι διαχειριστικές δαπάνες, λόγω του ότι απαιτείται ολιγομελής ομάδα υποστήριξης του συστήματος
- ✓ Μεγάλη Ευελιξία: Επιτυγχάνεται γρήγορη και άμεση εφαρμογή των αλλαγών όπως περιγράφηκε και προηγουμένως.
- ✓ Αύξηση Αποδοτικότητας στις επιχειρησιακές διαδικασίες: Μειώνεται ο απαιτούμενος χρόνος απονομής προσβάσεων σε νέους εργαζομένους.
- ✓ Αύξηση Αποδοτικότητας στην χρήση των Π.Σ: Το RBAC δίνει τη δυνατότητα στους οργανισμούς να χρησιμοποιούν λιγότερα Π.Σ. τα οποία είναι ευκολότερα διαχειρίσιμα όσον αναφορά τη διενέργεια του ελέγχου πρόσβασης
- ✓ Αύξηση της Συμμόρφωσης: Διευκολύνεται η επίτευξη της Συμμόρφωσης, μέσω της εξουσιοδότησης ρόλων αντί μεμονωμένων εφαρμογών και στοιχείων. Η δημιουργία προτυποποιημένης διαδικασίας για κάθε επιχειρησιακή διαδικασία, διευκολύνει τη διενέργεια του ελέγχου Συμμόρφωσης.

Η επιβολή του ρόλο-κεντρικού ελέγχου πρόσβασης περιορίζει την πρόσβαση στο σύστημα μόνο σε εξουσιοδοτημένους χρήστες, ενώ ταυτόχρονα απλοποιεί τη διαχείριση των προνομίων των χρηστών καθώς γίνεται μαζική διαχείριση των αντίστοιχων προσβάσεων των ρόλων. Ο σχεδιασμός των ρόλων γίνεται με βάση τις ανάγκες του κάθε οργανισμού λαμβάνοντας υπ' όψιν όλα τα απαραίτητα στοιχεία και τους περιορισμούς. Με τη χρήση του RBAC οι νέοι χρήστες των Π.Σ. του οργανισμού εντάσσονται εύκολα στους υφιστάμενους ρόλους με βάση τις απαιτήσεις της δουλειάς τους και αποκτούν άμεσα πρόσβαση στους απαιτούμενους πόρους του συστήματος. Για την ορθή εφαρμογή του μοντέλου απαιτείται να γίνονται συχνά έλεγχοι από τους διαχειριστές που θα έχουν σαν στόχο την εκχώρηση και ανάκληση προνομίων σύμφωνα με τις ανάγκες του οργανισμού. Όμως, το μεγάλο πλεονέκτημα του RBAC είναι το ότι επιτρέπει στους διαχειριστές να διαχειρίζονται και να επεξεργάζονται μόνο ένα σύνολο ρόλων και όχι το σύνολο των χρηστών.

3.7 Προκλήσεις

Το RBAC δεν είναι η απόλυτη λύση για όλα τα προβλήματα. Παρά τα πλεονεκτήματα του, παρά τη διεθνή αναγνώριση αναφορικά με τη χρηστική και λειτουργική του αξία, υπάρχουν σοβαρές προκλήσεις οι οποίες καθιστούν την ορθή εφαρμογή του, μια όχι και τόσο εύκολη

υπόθεση. Η εφαρμογή του RBAC απαιτεί την ικανοποίηση πολλών και ποικίλων παραμέτρων και προϋποθέσεων διότι:

- Δύσκολα εφαρμόζεται μεμονωμένα

Αν ένας οργανισμός επιλέξει να εφαρμόσει το RBAC πρέπει να λάβει υπόψη του ότι αυτό δεν μπορεί να εφαρμοστεί μεμονωμένα αλλά πρέπει να αλλάξει η γενική θεώρηση γύρω από τον τρόπο με τον οποίο διεξάγεται ο έλεγχος πρόσβασης. Επίσης μια υλοποιημένη μορφή RBAC σε ένα οργανισμό δεν μπορεί να εφαρμοστεί σε έναν άλλο (Northcutt, 2012). Αυτό ισχύει όχι μόνο για διαφορετικής μορφής οργανισμούς, αλλά και για οργανισμούς με ομοιότητες ως προς το αντικείμενο εργασιών και την οργανωτική δομή . Η υλοποίηση στατικού RBAC σε ένα συγκεκριμένο οργανισμό ήταν, είναι και θα είναι μοναδική.

- Απαιτεί επιμονή

Το RBAC δεν είναι ένα έργο το οποίο μπορεί να υλοποιηθεί σε σύντομο χρονικό διάστημα. Απαιτεί αρκετό χρόνο, πολλούς πόρους και είναι μια επένδυση αρκετά δαπανηρή για τον οργανισμό που αποφασίζει να επενδύσει σε αυτό.

- Απαιτεί αλλαγή συμπεριφοράς

Ο τρόπος που σχεδιάζεται και εφαρμόζεται ο κλασσικός έλεγχος πρόσβασης είναι αρκετά διαφορετικός από το σχεδιασμό και την υλοποίηση του RBAC. Θα πρέπει να καταστεί σαφές και να γίνει κατανοητό, ότι οι λειτουργικές διαφορές μεταξύ των διαφορετικών μοντέλων και η μετάβαση στη διαχείριση του ελέγχου πρόσβασης των χρηστών, πρέπει να είναι ομαλή και να μη δημιουργήσει προβλήματα στην παραγωγική διαδικασία του οργανισμού.

- Απαιτεί αλλαγή τρόπου διοίκησης

Θα πρέπει να αλλάξει ο τρόπος της διοίκησης του ελέγχου πρόσβασης. Δεν θα υπάρχει πλέον αλληλεπίδραση με τους χρήστες. Θα εξαλειφτούν οι προσωπικές επαφές και οι αιτήσεις για νέες προσβάσεις, από τους χρήστες των οργανισμών. Το μόνο το οποίο θα μένει ως καθήκον για τους διαχειριστές θα είναι η διαρκής εργασία με στόχο τη βελτίωση των ρόλων και των υφιστάμενων προνομίων τους.

- Απαιτεί τη συμβολή πολλών τμημάτων του οργανισμού.

Όπως αναφέρθηκε και παραπάνω, η εφαρμογή του RBAC δεν είναι εύκολη υπόθεση. Το RBAC είναι ιδιαίτερα διαδομένο σε μεγάλους και πολύπλοκους οργανισμούς λόγω των πλεονεκτημάτων του. Όμως για να φανούν και να αξιοποιηθούν τα πλεονεκτήματα πρέπει να γίνει σωστός σχεδιασμός και σωστή ανάλυση απαιτήσεων. Για να γίνει αυτό δεν αρκεί να

εργαστεί μόνο το τμήμα του IT Security του οργανισμού. Χρειάζεται συνεργασία μεταξύ αυτού και των τμημάτων της Διοίκησης Ανθρωπίνων Πόρων ώστε να γίνει το πλέον λειτουργικό role engineering και ασφαλώς με το Οργανωτικό τμήμα του οργανισμού ώστε να μην αλλάξει επί της ουσίας ο τρόπος λειτουργίας του οργανισμού.

- Δεν λύνει όλα τα προβλήματα – Δεν μπορεί να εφαρμοστεί παντού

Το RBAC δεν είναι η λύση σε όλα τα προβλήματα που αφορούν τον έλεγχο πρόσβασης εντός ενός οργανισμού. Δεν μπορεί και δεν πρέπει να εφαρμόζεται σε όλους τους τύπους των οργανισμών. Όπως παρουσιάστηκε αναλυτικά έχει και αρνητικά σημεία τα οποία θα πρέπει να λαμβάνονται υπόψη, ώστε οι οργανισμοί που δεν το χρειάζονται να μη μπουν στον πειρασμό της εφαρμογής του.

3.8 Συμπεράσματα

Όπως έχει αναφερθεί παραπάνω, ο έλεγχος πρόσβασης διενεργείται χορηγώντας ή/και αφαιρώντας προσβάσεις από τους χρήστες. Ακολουθώντας τις βασικές αρχές του RBAC επιτυγχάνεται η ευκολότερη διαχείριση, καθώς μπορεί να πραγματοποιείται μαζικά και για όλους τους χρήστες μέσω μόνο ενός συστήματος ή μιας εφαρμογής. Αυτό το στρατηγικό πλεονέκτημα του ρόλο-κεντρικού ελέγχου πρόσβασης έχει δύο άμεσα αποτελέσματα, τα οποία είναι τόσο η αύξηση του επιπέδου της ασφάλειας όσο και η μείωση της πολυπλοκότητας του συστήματος.

Λαμβάνοντας υπόψη, τα πλεονεκτήματα και γνωρίζοντας τις ανάγκες για τη διενέργεια, ασφαλούς ελέγχου πρόσβασης με το μικρότερο δυνατό διαχειριστικό κόστος, στους σύνθετους οργανισμούς, θα ακολουθήσει στα επόμενα κεφάλαια η παρουσίαση μιας λύσης που θα αφορά το σχεδιασμό και την υλοποίηση μιας πολιτικής ασφάλειας βασισμένης στο RBAC. Ο σύνθετος οργανισμός που επιλέχθηκε να μελετηθεί είναι ένα υποκατάστημα τράπεζας. Όπως θα αποδειχθεί και παρακάτω τα πλεονεκτήματα του σχεδιασμού και της υλοποίησης μιας RBAC πολιτικής στο προαναφερθέντα παραγωγικό χώρο, είναι αρκετά.

Το RBAC είναι μια προσέγγιση, η οποία έχει ως στόχο την επίτευξη Defense in Depth και αυτό απαιτεί διενέργεια του ελέγχου πρόσβασης σε επιχειρησιακή βάση. Το σημαντικότερο ζήτημα που πρέπει να επιλυθεί είναι να βρεθεί τρόπος σύνδεσης μεταξύ των εννοιών ατόμου και χρήστη. Για να μπορέσει λοιπόν να θεωρηθεί το RBAC ως ένα πλαίσιο (DiD), πρέπει να είμαστε σε θέση να συμπεριλάβουμε σε αυτό υπολογιστικά και πληροφοριακά συστήματα, δίκτυα, καθώς επίσης και να επεκτείνουμε τη στενή έννοια των αντικειμένων όπως αυτά είναι γνωστά σαν δεδομένα, εφαρμογές, βάσεις δεδομένων κ.α.

Λειτουργικά το RBAC μπορεί να καλύψει όλες τις πιθανές επιχειρησιακές λειτουργίες, οι οποίες μπορούν να ελεγχθούν από τη δημιουργία των ρόλων, την ανάθεση δικαιωμάτων και αδειών προσπέλασης στα υποκείμενα, στα αντικείμενα και στους ρόλους.

Σήμερα όλο και περισσότεροι κατασκευαστές λογισμικού δημιουργούν λογισμικά που να υλοποιούν RBAC πολιτικές όπως η Cisco και η Secure Computing. Λειτουργικά συστήματα όπως είναι τα Windows (θα δειχθεί υλοποίηση στο επόμενο κεφάλαιο), HP, Solaris, έχουν σήμερα ενσωματωμένες προτάσεις που υποστηρίζουν το RBAC. Ακόμα όμως και συστήματα με διαφορετική αρχιτεκτονική όπως είναι τα Unix μπορούν σήμερα να υποστηρίξουν RBAC εφαρμογές μέσω κάποιων επεκτάσεων που εγκαθίστανται στο πυρήνα. Τέτοιες επεκτάσεις για το λειτουργικό Unix είναι το GrSecurity και το SELinux.

4 Σχεδιασμός, Υλοποίηση & Έλεγχος Πολιτικής Ελέγχου Πρόσβασης

4.1 Εισαγωγή

Ό Έλεγχος Πρόσβασης Βασισμένος σε Ρόλους καλύπτει ένα πολύ μεγάλο τμήμα των προσεγγίσεων των μεγάλων οργανισμών αναφορικά με τη εφαρμογή ενός πλήρους σχεδίου Διαχείρισης Προσβάσεων Χρηστών. Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο το RBAC προσφέρει πολλά διαχειριστικά πλεονεκτήματα και σήμερα έχει επικρατήσει ως το πλέον συμφέρον μοντέλο για την διενέργεια του ελέγχου πρόσβασης.

Τα περισσότερα λογισμικά σχεδιάζονται με βάση τις κατευθυντήριες γραμμές που δίνονται από αυτό αλλά δαπανάται μεγάλο χρονικό διάστημα τόσο στη φάση της ανεύρεσης των κατάλληλων ρόλων όσο και στο σχεδιασμό καθώς και την υλοποίηση τέτοιου είδους συστημάτων.

Με βάση τις εμπειρίες που αποκόμισα κατά την πρακτική μου άσκηση, που πραγματοποιήθηκε το χρονικό διάστημα από τον Ιούλιο έως τον Οκτώβρη 2012 στη Διεύθυνση Λειτουργιών και Υποδομών Πληροφορικής στην Εθνική Τράπεζα της Ελλάδος μπορεί να διατυπωθεί η άποψη ότι τα συγκριτικά πλεονεκτήματα και η ευελιξία που προσφέρει το μοντέλο RBAC το καθιστούν, σχεδόν μονόδρομο, για το σχεδιασμό και την εφαρμογή νέων πολιτικών ασφάλειας Ελέγχου Πρόσβασης.

Στο κεφάλαιο αυτό παρουσιάζεται μια πλήρης λύση που αφορά το σχεδιασμό, την υλοποίηση και τη διενέργεια του Ελέγχου Πρόσβασης βασισμένου σε Ρόλους σε έναν πολύπλοκο οργανισμό. Δίδεται μια ολοκληρωμένη λύση για το πώς θα μπορέσει να τεθεί σε εφαρμογή ένα κοστοβόρο και δύσκολο σχέδιο που όμως, αν σχεδιαστεί ορθά και υλοποιηθεί σύμφωνα με τις κατευθυντήριες οδηγίες του προτύπου ANSI/INCITS 359-2004, θα προσφέρει πολλαπλά οφέλη στον οργανισμό.

4.2 Το προς επίλυση Πρόβλημα

Σύμφωνα με πολλούς μελετητές ασφάλειας και συγκεκριμένα σύμφωνα με αυτούς που καταπιάνονται με τη θεματική ενότητα του Identity Management και του Role Engineering θεωρείται ότι ο Έλεγχος Πρόσβασης των χρηστών, δεν είναι ένα πρόβλημα που αφορά αποκλειστικά το τμήμα πληροφορικής του κάθε οργανισμού. Πρόκειται για ένα ζήτημα το οποίο θα πρέπει να απασχολεί τον οργανισμό τόσο στο επιχειρησιακό, όσο και στο οργανωτικό- σχεδιαστικό και το εκτελεστικό του τμήμα.

Ένας χρηματοπιστωτικός οργανισμός απασχολεί αρκετές εκατοντάδες έως και μερικές χιλιάδες εργαζομένους ανάλογα με το μέγεθός του. Ο έλεγχος πρόσβασης όπως αντιλαμβάνεται ο καθένας, αποτελεί ένα ζήτημα, το οποίο χρίζει σοβαρής αντιμετώπισης έχοντας ως στόχο την ασφαλή και αποτελεσματική λειτουργία του οργανισμού. Είναι δεδομένο, ότι οι εργαζόμενοι στα υποκαταστήματα των σύγχρονων τραπεζικών μονάδων, έχουν ανάγκη από υπολογιστικούς πόρους και συστήματα, για την ομαλή περάτωση των εργασιών τους. Το ζήτημα της απονομής μεμονωμένων προσβάσεων σε ένα τόσο μεγάλο αριθμό χρηστών, ενδέχεται να καταστήσει το σύστημα μη διαχειρίσιμο, μη λειτουργικό και πιθανότατα μη ασφαλές. Η παρούσα ΜΔΕ παραθέτει ένα σύνολο προτάσεων και παρουσιάζει μια λύση για το πώς πρέπει να σχεδιάζεται και πως μπορεί να υλοποιηθεί η σχεδιασθείσα πολιτική ασφάλειας. **Το μοντέλο με βάση το οποίο θα γίνει ο σχεδιασμός και η υλοποίηση είναι το RBAC.**

Το κεφάλαιο θα χωριστεί σε τρία μέρη, στα οποία περιγράφονται όσα χρειάζονται για να γίνει λειτουργική και αποτελεσματική μια RBAC λύση.

Στο πρώτο μέρος θα γίνει αναγνώριση των αναγκών σε επίπεδο ρόλων, ώστε να μπορεί να λειτουργήσει ένα υποκατάστημα τράπεζας. Στο ίδιο μέρος, γίνεται και η γενικότερη ανάλυση απαιτήσεων του συστήματος, σχετικά με τις εργασίες που πρέπει να εκτελούνται, ποιοι ρόλοι πρέπει να εκτελούν τις εκάστοτε εργασίες και πως πρέπει αυτές να πραγματοποιούνται.. Η αρχιτεκτονική του συστήματος παρουσιάζεται στο τέλος του πρώτου μέρους.

Εν συνεχεία, περνώντας στο δεύτερο μέρος, παρουσιάζεται το εργαλείο Authorization Manager μιας built in εφαρμογής των Windows η οποία προσφέρει τη δυνατότητα να σχεδιαστούν πολιτικές RBAC.

Στο τελευταίο μέρος του κεφαλαίου παρουσιάζεται ένας τρόπος, μέσω του οποίου, μπορούν να δημιουργούνται εφαρμογές στις οποίες θα καλείται η πολιτική που σχεδιάστηκε και υλοποιήθηκε στο δεύτερο μέρος.

Έτσι προτείνεται μια ολοκληρωμένη λύση στο πρόβλημα του Ελέγχου Πρόσβασης και της Διαχείρισης Χρηστών για όλα τα υποκαταστήματα μιας τράπεζας, λύνοντας σημαντικά προβλήματα διαχείρισης, διευκολύνοντας τη διαχειριστική λειτουργία, αυξάνοντας το επίπεδο ασφάλειας και επιδεικνύοντας συμμόρφωση με το διεθνές πρότυπο του RBAC.

4.3 Σενάριο

Για την ολοκλήρωση της ιδέας της ΜΔΕ γίνεται επαρκής περιγραφή του υποθετικού σεναρίου στο οποίο βασίζεται η επίλυση του προβλήματος. Καλούμαστε να μελετήσουμε μια

περίπτωση μεγάλου χρηματοπιστωτικού ιδρύματος και να βελτιστοποιήσουμε τη διαδικασία μέσω της οποίας γίνεται ο έλεγχος πρόσβασης για τους υπαλλήλους των καταστημάτων.

Οι μεγάλοι τραπεζικοί οργανισμοί εδώ και χρόνια προσπαθούν να στραφούν στο RBAC. Όμως, εξαιτίας της πολυπλοκότητας τους ακόμα και αυτή η προσέγγιση, που όπως παρουσιάστηκε στα προηγούμενα κεφάλαια είναι αποδοτικότερη και ευκολότερα διαχειρίσιμη από το MAC και το DAC, δεν είναι εύκολα υλοποιήσιμη. Επί της ουσίας, η εργασία σκοπεύει να παραθέσει μια προσέγγιση RBAC η οποία είναι σχετικά απλή, χρησιμοποιεί απλά εργαλεία και μπορεί με μικρές αλλαγές να εφαρμοστεί σε πραγματικό περιβάλλον. Έχοντας ως βασικό στόχο την αύξηση του επιπέδου ασφαλείας του υπολογιστικού συστήματος αλλά και τη βελτιστοποίηση της παραγωγικής διαδικασίας προτείνεται η παρακάτω υλοποίηση, η οποία στηρίζεται στον έλεγχο πρόσβασης βασισμένο σε ρόλους και δανείζεται βασικά στοιχεία του διακριτικού ελέγχου πρόσβασης, ο οποίος χρησιμοποιείται ως προεπιλεγμένος στα λειτουργικά συστήματα της Microsoft.

Για να προχωρήσουμε στη παρουσίαση της υλοποίησης για τη βελτιστοποίηση της διαδικασίας του ελέγχου πρόσβασης πρέπει να αναφέρουμε τα ακόλουθα:

- Ο οργανισμός χρησιμοποιεί λειτουργικά Microsoft
- Υπάρχει η δυνατότητα μερικής διαχείρισης χρηστών ανά κατάστημα

Η παρούσα μελέτη χωρίζεται σε τρία βασικά στάδια.

1. Αρχικά πρέπει να γίνει το κατάλληλο role & operations engineering ώστε να αποτυπωθούν οι απαραίτητοι ρόλοι και οι εργασίες που πραγματοποιούν οι χρήστες για την πλήρη λειτουργία ενός καταστήματος.

2. Εν συνεχεία περιγράφεται η υλοποίηση της πολιτικής ασφάλειας, στο κομμάτι του ελέγχου πρόσβασης, του οργανισμού καθώς και η δημιουργία των κατάλληλων ρόλων και εργασιών, σε περιβάλλον MS Windows με χρήση του Authorization Manager.

3. Τέλος δημιουργείται μια εφαρμογή, την οποία θα χρησιμοποιούν οι χρήστες και ο έλεγχος πρόσβασης θα πραγματοποιείται βάσει της υφιστάμενης πολιτικής που έχει αποτυπωθεί στο Authorization Manager.

Όμως θα παραμένει το ερώτημα αν η πολιτική ασφάλειας του Οργανισμού έχει αποτυπωθεί κατάλληλα και επαρκώς μέσω του Authorization Manager. Για να μπορέσουμε να πάρουμε απάντηση σε αυτό το πολύ κρίσιμο ερώτημα για την ασφάλεια του συστήματος, πρέπει να

αποτυπωθεί η πολιτική ασφάλειας σε ένα xml αρχείο και να γίνεται σύγκριση με το παραγόμενο xml του Authorization Manager.

Βασικοί στόχοι του σεναρίου είναι:

- ✓ Η Επιτυχής ανάλυση απαιτήσεων
- ✓ Ο Επιτυχής έλεγχος Πρόσβασης
- ✓ Ο Σχεδιασμός RBAC πολιτικής ελέγχου πρόσβασης βάση του βασικού μοντέλου RBAC και του Ιεραρχικού RBAC
- ✓ Η Ικανοποίηση των βασικών αρχών ασφάλειας που ενσωματώνει το RBAC
- ✓ Η Διευκόλυνση του Audit
- ✓ Η συγγραφή εγχειριδίου χρήστη ώστε να είναι δυνατές οι αλλαγές στην πολιτική από τους μελλοντικούς σχεδιαστές
- ✓ Η Διευκόλυνση στη διαχείριση
- ✓ Η Αύξηση του επιπέδου ασφάλειας

4.4 Ανάλυση Απαιτήσεων – Role Engineering

Στην παρούσα παράγραφο περιγράφεται η μελέτη των αναγκών και η ανάλυση των απαιτήσεων που πρέπει να γίνει για να σχεδιαστεί μια ρόλο-κεντρική πολιτική ελέγχου πρόσβασης. Η πολιτική που θα σχεδιαστεί και θα υλοποιηθεί αφορά σε ένα υποκατάστημα τράπεζας. Οι εργασίες που πραγματοποιούνται σε τράπεζες αφορούν χρηματοοικονομικές συναλλαγές, οπότε ο έλεγχος πρόσβασης των υπαλλήλων πρέπει να αντιμετωπίζεται με τη δέουσα προσοχή.

Ένα υποκατάστημα τράπεζας, για να καλύψει τις ανάγκες του χρειάζεται τη δημιουργία κάποιων βασικών ρόλων. Οι ανάγκες του αφορούν ως επί το πλείστον στις συναλλαγές μεταξύ του τραπεζικού καταστήματος και των πελατών. Οι συναλλαγές χωρίζονται σε δύο επιμέρους κατηγορίες: α) τις Εγχρήματες και β) τις Ενημερωτικές – Μη Εγχρήματες συναλλαγές. Βασικοί στόχοι ενός υποκαταστήματος τράπεζας, από πλευράς ελέγχου πρόσβασης, είναι η εξουσιοδοτημένη πρόσβαση των υπαλλήλων στα κατάλληλα συστήματα και στους κατάλληλους πόρους ώστε να εξυπηρετηθούν οι ανάγκες τόσο των πελατών όσο και της Τράπεζας.

4.4.1 Ανάλυση Εργασιών

Πολύ σημαντικό κομμάτι στην ανάλυση των απαιτήσεων, είναι η σαφής καταγραφή των λειτουργιών του υποκαταστήματος και η ταυτοποίηση - καταγραφή των αναγκών πρόσβασης σε πόρους. Οι εργασίες - που καλούνται να εκτελέσουν οι υπάλληλοι θα πάρουν την μορφή Operation στο Authorization Manager. Παράλληλα θα συνοδεύονται και από έναν μοναδικό αριθμό οποίος θα τις χαρακτηρίζει.

Μελετώντας λοιπόν προσεκτικά τον τρόπο λειτουργίας ενός υποκαταστήματος τράπεζας, προέκυψαν τα ακόλουθα συμπεράσματα όσον αναφορά τις εργασίες που καλούνται να εκτελέσουν οι χρήστες. Πρόκειται για εννέα ενότητες συναλλαγών, οι οποίες περιέχουν 13 εργασίες που θα διαχωριστούν, θα ομαδοποιηθούν και θα ανατεθούν σε ρόλους.

1. **Μη εγχρήματες συναλλαγές**. Οι μη εγχρήματες συναλλαγές έχουν ως στόχο τόσο την εξυπηρέτηση της πελατείας, παρέχοντας πληροφορίες και ενημέρωση, όσο και την εξυπηρέτηση εσωτερικών διαδικασιών . Οι συναλλαγές αυτές υποστηρίζουν τις ακόλουθες 4 επιμέρους ομάδες εργασιών:

    a) **Πληροφόρηση και ενημέρωση των πελατών** – Απαιτείται πρόσβαση και δικαίωμα ανάγνωσης, στο πληροφοριακό σύστημα της τράπεζας και ειδικότερα σε συγκεκριμένες συναλλαγές που παρέχουν πληροφορίες για τους λογαριασμούς της πελατείας.

    b) **Εκτέλεση διαδικασιών δημιουργίας και διαχείρισης καταθετικών λογαριασμών** – Απαιτείται πρόσβαση και δικαίωμα εκτέλεσης εγχρήματων συναλλαγών (καταθέσεων, αναλήψεων κ.λ.π.), στο πληροφοριακό σύστημα των λογαριασμών.

    c) **Προώθηση τραπεζικών προϊόντων και διαβίβαση αιτήσεων για χορήγηση τραπεζικών προϊόντων, χωρίς εξασφαλίσεις** – Απαιτείται πρόσβαση καθώς και δικαίωμα ανάγνωσης και εκτέλεσης εγγραφής, στο σύστημα υποβολής αιτήσεων καταναλωτικών δανείων και πιστωτικών καρτών.

2. **Εγχρήματες συναλλαγές με όριο.** Οι εγχρήματες συναλλαγές μέχρι ενός προκαθορισμένου ορίου χωρίζονται σε δύο επιμέρους εργασίες:

    a) **Πιστωτικές συναλλαγές με όριο μέχρι 20.000 νομισματικών μονάδων (€)** – Απαιτείται πρόσβαση και δικαίωμα εγγραφής με όριο, στο πληροφοριακό σύστημα χρεώσεων-πιστώσεων λογαριασμών.

b) **Χρεωστικές συναλλαγές με όριο μέχρι 1.000 νομισματικών μονάδων(€)** – Απαιτείται πρόσβαση και δικαίωμα εγγραφής με όριο, στο πληροφοριακό σύστημα χρεώσεων-πιστώσεων λογαριασμών.

3. **Εγκρίσεις εγχρήματων συναλλαγών που υπερβαίνουν τα παραπάνω όρια**. Η εργασία αφορά στην έγκριση χρεωστικών εγχρήματων συναλλαγών της 2ης κατηγορίας, ποσού μεγαλύτερου των **1.000 νομισματικών μονάδων(€) και μέχρι του ποσού των 5.000 νομισματικών μονάδων(€).** . – Απαιτείται πρόσβαση στο πληροφοριακό σύστημα χρεώσεων-πιστώσεων λογαριασμών και δικαίωμα εγγραφής με όριο.

4. **Εγκρίσεις κάθε τύπου εγχρήματων συναλλαγών χωρίς όριο** – Η εργασία αφορά στην έγκριση κάθε τύπου εγχρήματων συναλλαγών τα ποσά των οποίων υπερβαίνουν τα όρια που αναφέρονται πιο πάνω. Απαιτείται πρόσβαση και δικαίωμα εγγραφής, στο πληροφοριακό σύστημα χρεώσεων-πιστώσεων λογαριασμών.

5. **Έλεγχος σύνθετων μη εγχρήματων εργασιών**. Πριν από τη διαβίβαση των στοιχείων στο κεντρικό σύστημα εσωτερικής επικοινωνίας – Απαιτείται πρόσβαση στο σύστημα εσωτερικής επικοινωνίας και δικαίωμα εγγραφής.

6. **Άντληση πληροφοριών**. Η εργασία πραγματοποιείται με στόχο τη διενέργεια κατασταλτικών ελέγχων, σε συναλλαγές που έχουν πραγματοποιηθεί από τους κατώτατους ιεραρχικά χρήστες - Απαιτείται πρόσβαση και δικαίωμα ανάγνωσης στα αρχεία, που έχουν δημιουργηθεί από όλες τις συναλλαγές (εγχρήματες και μη) των Tellers

7. **Λήψη εμπιστευτικών εγγράφων**. Η εργασία αυτή αφορά στη λήψη εμπιστευτικών και απόρρητων εγγράφων που προορίζεται για το υποκατάστημα – Απαιτείται πρόσβαση και δικαίωμα ανάγνωσης, στο σύστημα απόρρητης εσωτερικής επικοινωνίας.

8. **Άντληση σύνθετων πληροφοριών**. Η εργασία πραγματοποιείται με στόχο τη διενέργεια κατασταλτικών ελέγχων, σε ενέργειες που έχουν πραγματοποιηθεί από τους Tellers & Chief Tellers του υποκαταστήματος - Απαιτείται πρόσβαση και δικαίωμα ανάγνωσης στα αρχεία των Tellers & Chief Tellers..

9. **Απονομή Ρόλων**. Η εργασία αυτή αφορά την απονομή ρόλων σε χρήστες που εργάζονται στο υποκατάστημα – Απαιτείται πρόσβαση στο σύστημα διαχείρισης προσβάσεων χρηστών.

4.4.2 Ρόλοι

Αφού έγινε ο ορισμός και η περιγραφή των εργασιών που πραγματοποιούνται από τους χρήστες (στελέχη και υπαλλήλους) ενός υποκαταστήματος, έφτασε η στιγμή να παρουσιαστούν οι απαιτούμενοι ρόλοι.

Ένα υποκατάστημα τράπεζας μπορεί να λειτουργήσει εύρυθμα και με πλήρη ασφάλεια μόλις με 3 ρόλους. Τα πλεονεκτήματα του RBAC αναφέρθηκαν αναλυτικά στο προηγούμενο κεφάλαιο, στη συγκεκριμένη περίπτωση τα πλεονεκτήματα συνοψίζονται παρακάτω. Είναι εύκολα αντιληπτό ότι τόσο το **διαχειριστικό όφελος**, όσο και το **όφελος ασφάλειας** είναι τεράστιο. Είναι πολύ πιο εύκολο να παραμετροποιούνται και να αλλάζουν τα δικαιώματα πρόσβασης σε 3 ρόλους, παρά σε χιλιάδες εργαζομένους. Από πλευράς Ασφάλειας, το βασικό πλεονέκτημα της λύσης προκύπτει από τη δυνατότητα ορθότερης, ευκολότερης και αποτελεσματικότερης χορήγησης δικαιωμάτων πρόσβασης σε 3 ρόλους, καθώς μπορεί να γίνει καλύτερη εκτίμηση για το πια δικαιώματα είναι πραγματικά αναγκαία για την ορθή λειτουργία της τράπεζας. Σπουδαίο πλεονέκτημα υπάρχει και από πλευράς ελέγχου, καθώς διευκολύνεται το έργο των ελεγκτών, ο οποίοι πρέπει να ελέγχουν μόνο αν τα δικαιώματα πρόσβασης των 3 ρόλων έχουν χορηγηθεί με βάση την ισχύουσα πολιτική ασφάλειας.

Η περιγραφή των ρόλων θα γίνει Ιεραρχικά καθώς έτσι θα σχεδιαστεί και η RBAC πολιτική ελέγχου πρόσβασης.

1. Ρόλος: **Teller.** Είναι ο απλούστερος ρόλος του συστήματος. Αποτελεί τη βάση της πυραμίδας και σε αυτόν απονέμονται τα δικαιώματα εκτέλεσης απλών εργασιών. Συνήθως στελεχώνεται από νέους χρήστες και σε όλους σχεδόν τους απλούς υπαλλήλους εκχωρείται ο συγκεκριμένος ρόλος.

2. Ρόλος: **Chief Teller**. Είναι επικεφαλής υπαλλήλων που έχουν το ρόλο του Teller και κληρονομεί όλα τα προνόμια και τα δικαιώματα που έχει ο ρόλος του απλού Teller. Διαθέτει περισσότερα και πιο σύνθετα δικαιώματα και εκχωρείται σε ικανότερους και πιο έμπειρους χρήστες.

3. Ρόλος: **Brach Manager**. Κληρονομεί όλα τα προνόμια και τα δικαιώματα των δύο παραπάνω ρόλων. Διαθέτει ειδικά καθώς και διαχειριστικά προνόμια και έχει όλες τις προσβάσεις που είναι απαραίτητες για την ομαλή και ασφαλή λειτουργία του υποκαταστήματος. Το ρόλο τον αναλαμβάνει ο επικεφαλής του κάθε υποκαταστήματος. Έτσι επιτυγχάνεται και ο διαχωρισμός καθηκόντων που είναι βασικό δομικό στοιχείο του προτύπου ANSI/INCITS 359-2004 του RBAC.

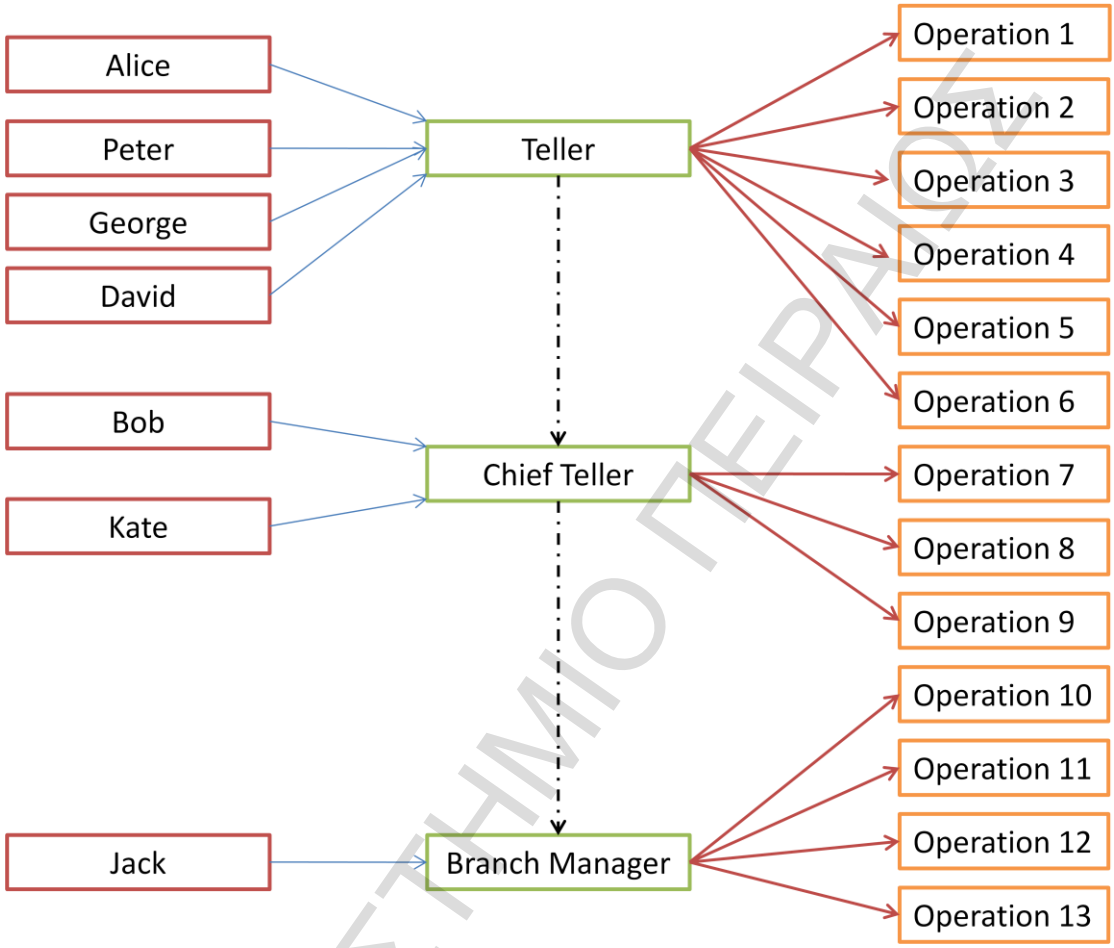
4.4.3 Σύνδεση Εργασιών & Ρόλων

Οι ρόλοι που σχεδιάστηκαν στην προηγούμενη παράγραφο πρέπει να αποκτήσουν: α) σύνδεση με τις εργασίες και β) τα κατάλληλα δικαιώματα. Το αποτέλεσμα της συγκεκριμένης σύνδεσης αποτυπώνεται στον παρακάτω πίνακα. Στην παρένθεση αναγράφεται ο μοναδικός αριθμός της κάθε εργασίας, ο οποίος και θα την ακολουθεί καθ' όλη την πειραματική διαδικασία.

Εργασίες	Access Rights	Teller	Chief Teller	Brach Manager
Πληροφόρηση και ενημέρωση των πελατών(1)	**R**	✓	✓	✓
Εκτέλεση διαδικασιών δημιουργίας καταθετικών λογαριασμών(2)	**W,X**	✓	✓	✓
Προώθηση τραπεζικών προϊόντων και αιτήσεις για τραπεζικά προϊόντα χωρίς εξασφαλίσεις(3)	**W,X**	✓	✓	✓
Διαβίβαση εσωτερικών πληροφοριακών στοιχείων(4)	**W,X**	✓	✓	✓
Πιστωτικές εργασίες με όριο 20.000 μονάδων(5)	**W**	✓	✓	✓
Χρεωστικές με όριο 1.000 μονάδων(6)	**W**	✓	✓	✓
Εγκρίσεις εγχρήματων συναλλαγών με μεγαλύτερο όριο(7)	**W**		✓	✓
Έλεγχος σύνθετων μη εγχρήματων εργασιών(8)	**W**		✓	✓
Άντληση πληροφοριών(9)	**W**		✓	✓
Εγκρίσεις κάθε τύπου εγχρήματων συναλλαγών χωρίς όριο(10)	**W**			✓
Λήψη εμπιστευτικών εγγράφων(11)	**R**			✓
Άντληση σύνθετων πληροφοριών(12)	**R**			✓

Απονομή Ρόλων(13)	**R,W,X**			✓

Πίνακας 1 Ανάθεση Προνομίων σε Ρόλους



Εικόνα 11 Αποτύπωση Ιεραρχίας και Δικαιωμάτων

Το αποτέλεσμα αυτής της διαδικασίας είναι από τα σημαντικότερα στοιχεία αυτής της εργασίας. Πραγματοποιήθηκε ανασχεδιασμός επιχειρησιακής διαδικασίας και η αύξηση της ασφάλειας μέσου του ελέγχου πρόσβασης αποτέλεσε το βασικό πυλώνα αυτού του συστήματος. Η απονομή προνομίων στους ρόλους τους συστήματος, υλοποιεί τις κατευθυντήριες γραμμές που τίθενται από το ISO 27001:2005 ως προς το Access Control και καλύπτει πλήρως τον οργανισμό σε θέματα συμμόρφωσης, ως προς τον έλεγχο πρόσβασης, με το πρότυπο.

4.5 Συμμόρφωση με το Πρότυπο ANSI/INCITS 359-2004

Αυτό το κομμάτι δείχνει τη συμμόρφωση με το πρότυπο ANSI/INCITS 359-2004. Ο σχεδιασμός είναι βασισμένος στις κατευθυντήριες γραμμές του προτύπου και η υλοποίηση

είναι σύμφωνη με το βασικό RBAC. Αναλυτικά με την υλοποίηση επιτυγχάνονται τα παρακάτω:

- ✓ Επιτρέπεται η δημιουργία και διαγραφή των χρηστών
- ✓ Επιτρέπεται η δημιουργία, η διαγραφή και η επεξεργασία των ρόλων.
- ✓ Η Εκχώρηση χρηστών σε ρόλους.
- ✓ Η ανάθεση των προνομίων πρόσβασης σταματάει να γίνεται στο κάθε χρήστη ατομικά αλλά δημιουργούνται ρόλοι.
- ✓ Η Ιεραρχική δομή που ορίζει το Hierarchical RBAC υποστηρίζεται απόλυτα καθώς ο σχεδιασμός των ρόλων γίνεται ιεραρχικά και με παρόμοιο τρόπο εφαρμόζεται η υλοποίηση.
- ✓ Επιτυγχάνεται ο Στατικός Διαχωρισμός Καθηκόντων μέσω της μοναδικότητας των ρόλων που μπορεί να κατέχει ένας χρήστης. Ο έλεγχος που πραγματοποιούν οι ανώτεροι χρήστες προς τους υφισταμένους τους δεν επιτρέπει σε κάποιον χρήστη να κατέχει κατώτερο ρόλο.

Ο Richard Fernadez το 2005, δημοσίευσε ποια είναι τα στοιχεία τα οποία πρέπει να ικανοποιούνται σε ένα σύστημα RBAC (Fernadez, 2005). Τα στοιχεία αφορούν τη συμμόρφωση με τις κατά ελάχιστες απαιτήσεις και παρατίθενται στο παρακάτω πίνακα.

Βασικές Λειτουργίες	
Add User	✓
Add Role	✓
Delete Role	✓
AssignUser	✓
Deassign User	✓
Grant Permission	✓
Revoke Permission	✓
Supporting System Functions for Core RBAC	✓

Add Active Role	✓
Drop Active Role	✓
Check Access	✓
Review Functions for Core RBAC	✓
Assigned Users	✓
Assigned Roles	✓
Advanced Review Functions for Core RBAC	✓
Role Permissions	✓
User Permissions	✓
Session Roles	✓
Session Permissions	✓
Role Operations OnObject	✓
UserOperationsOnObject	✓
Compliance with Hierarchical RBAC	✓

Πίνακας 2 Συμμόρφωση προτεινόμενου συστήματος με το πρότυπο ANSI/INCITS 359-2004

4.6 Εγχειρίδιο Χρήσης του Authorization Manager

Βασικό εργαλείο για την εκπόνηση της παρούσας μελέτης είναι το Authorization Manager. Το λειτουργικό MS Windows, στις διάφορες εκδόσεις του, ακολουθεί τις βασικές αρχές του διακριτικού ελέγχου πρόσβασης. Όμως, με το πέρασμα των χρόνων προέκυψε η ανάγκη να υπάρξει μια επέκταση στις πολιτικές ελέγχου πρόσβασης στο εν λόγω λειτουργικό σύστημα. Για αυτό το λόγο αναπτύχθηκε το εργαλείο Authorization Manager το οποίο το συναντάμε από την έκδοση του Windows Server 2003 και έπειτα. Πρόκειται για ένα εργαλείο, το οποίο επιτρέπει στο διαχειριστή του συστήματος να εφαρμόσει μια role based πολιτική ελέγχου πρόσβασης που έχει σχεδιαστεί από το IT Security τμήμα του οργανισμού. Το Authorization Manager ακολουθεί πιστά της κατευθυντήριες γραμμές που τίθενται από το πρότυπο του RBAC. Η χρήση του είναι σχετικά απλή και η εξέλιξη του συνεχής. Παρακάτω γίνεται μια μικρή παρουσίαση του εν λόγω εργαλείου.

Το συγκεκριμένο API είναι αυτόματα διαθέσιμο στις μεταγενέστερες εκδόσεις των Windows XP. Για να χρησιμοποιηθεί από τα Windows XP θα πρέπει να εγκατασταθούν τα Administrative Tools που χορηγεί δωρεάν η Microsoft μέσα από τη διαδικτυακή της βιβλιοθήκη.

4.6.1 Οντολογία του AzMan

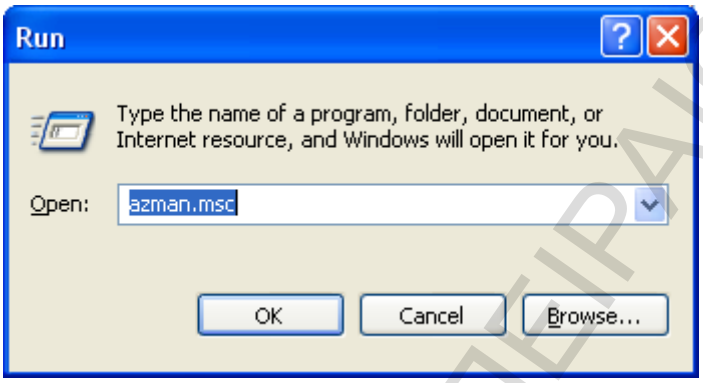
Το AzMan αποτελείται από 6 βασικά στοιχεία τα οποία ορίζονται και περιγράφονται παρακάτω.

- ✓ **Operation**: Είναι η δομική μονάδα του AzMan και αποτελεί το μέσο ορισμού της ασφάλειας. Η κάθε λειτουργία συνδέεται με έναν μοναδικό αριθμό, ο οποίος καλείται από τις εφαρμογές και η πρόσβαση σε κάθε συγκεκριμένη λειτουργία δίνεται με βάση τους κανόνες που έχουν οριστεί στο περιβάλλον του AzMan.
- ✓ **Task**: Το Task έχει τη δυνατότητα να ομαδοποιεί ένα σύνολο λειτουργιών. Αποτελεί παρόμοια έννοια με το Operation και επιπρόσθετα μπορεί να ορίζεται και από άλλα Tasks. Υπάρχει τέλος η δυνατότητα να εκτελούνται μέσα στα Tasks scripts ώστε να αποτυπώνονται επιχειρησιακοί κανόνες A task can also be based on other tasks.
- ✓ **Authorization Store**: Authorization store είναι ο χώρος στον οποίο αποθηκεύονται όλοι οι κανόνες ασφάλειας, δηλαδή επί της ουσίας η πολιτική ελέγχου πρόσβασης. Μπορεί να έχει διάφορες μορφές όπως: απλό xml αρχείο ή να είναι «στημένο» σε SQL Server, LDAP και Active Directory για να λειτουργήσει σε πραγματικό περιβάλλον.
- ✓ **Application**: Ίσως το στοιχείο που παρουσιάζει το μεγαλύτερο ενδιαφέρον στο AzMan και είναι αρμόδιο για τη δημιουργία operations, δημιουργίας ρόλων και ανάθεσης αυτών των συγκεκριμένων operation σε ρόλους. Επίσης εδώ πραγματοποιείται και η εκχώρηση χρηστών σε ρόλους.
- ✓ **Scope** : Ένα application μπορεί να διαιρεθεί σε διαφορετικά scopes τα οποία μπορούν να έχουν τα δικά τους operations, tasks και ρόλους.
- ✓ **Role**: Ο ρόλος στο AzMan δεν έχει καμία απολύτως διαφορά από την έννοια που έχει περιγραφεί αναλυτικά στο κεφάλαιο 2. Μέσω του AzMan δημιουργούνται, διαγράφονται και είναι διαχειρίσιμοι ρόλοι. Όπως σε κάθε εργαλείο που βασίζεται στο RBAC έτσι και εδώ ο ρόλος παίζει πρωτεύουσα σημασία.

4.6.2 Περιγραφή και εικόνες

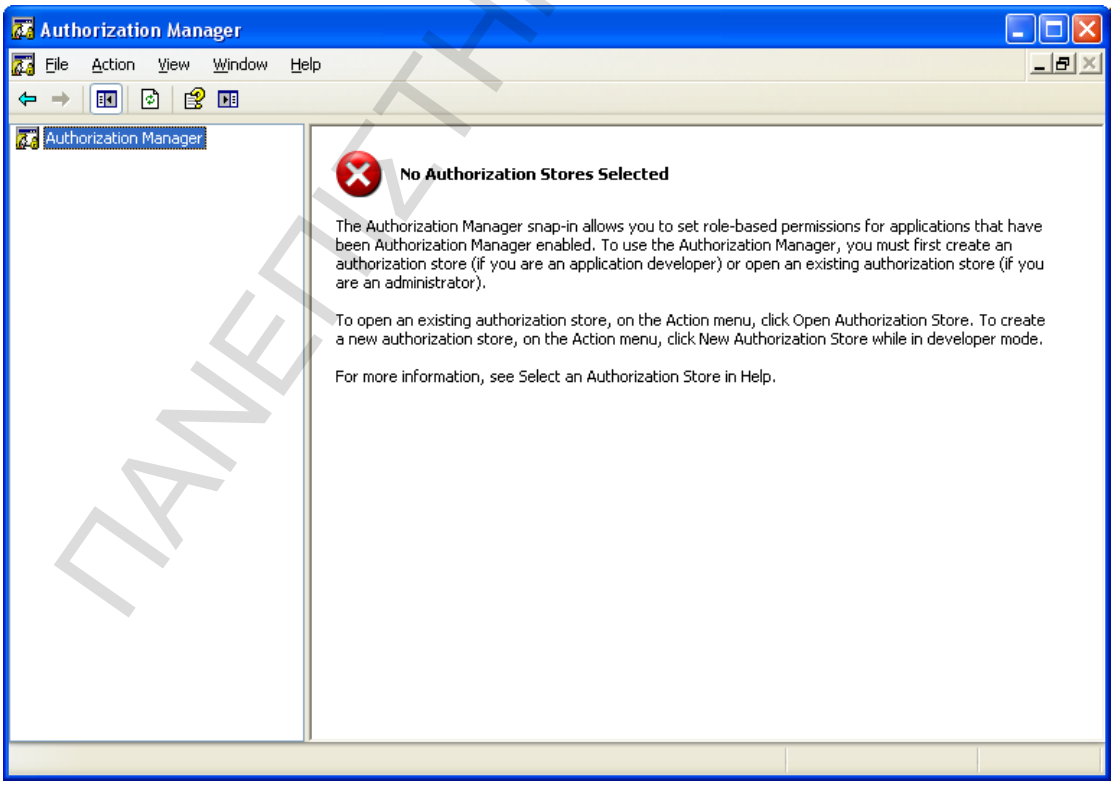
Για την υλοποίηση μιας εφαρμογής RBAC με τη χρήση του AzMan ακολουθείται η παρακάτω διαδικασία, η οποία εκτός από τη λεκτική περιγραφή παρουσιάζεται και σχηματικά με την παράθεση των σχετικών εικόνων.

Αρχικά ο χρήστης μεταβαίνει στο Run και δίνει την εντολή azman.msc.



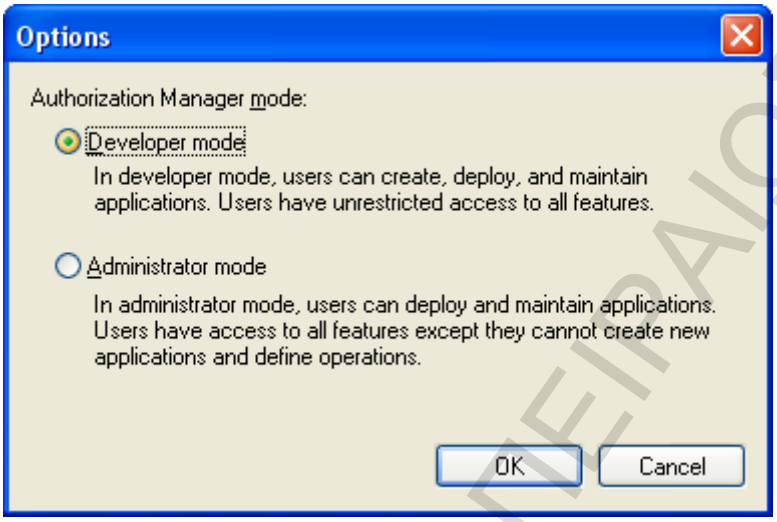
Εικόνα 12 Κλήση Authorization Manager

Το Authorization Manager ή AzMan (Follette, 2007) ανοίγει στην μορφή που φαίνεται στην εικόνα 13



Εικόνα 13 Αρχική Εικόνα Authorization Manager

Το AzMan, κατά την εκκίνηση του βρίσκεται σε κατάσταση Administrator mode. Για να μπορέσει κάποιος να υλοποιήσει μια πολιτική ασφάλειας θα πρέπει να αλλάξει την κατάσταση του συστήματος σε Developer mode. Αυτό επιτυγχάνεται επιλέγοντας από το μενού Action την κατάλληλη επιλογή.



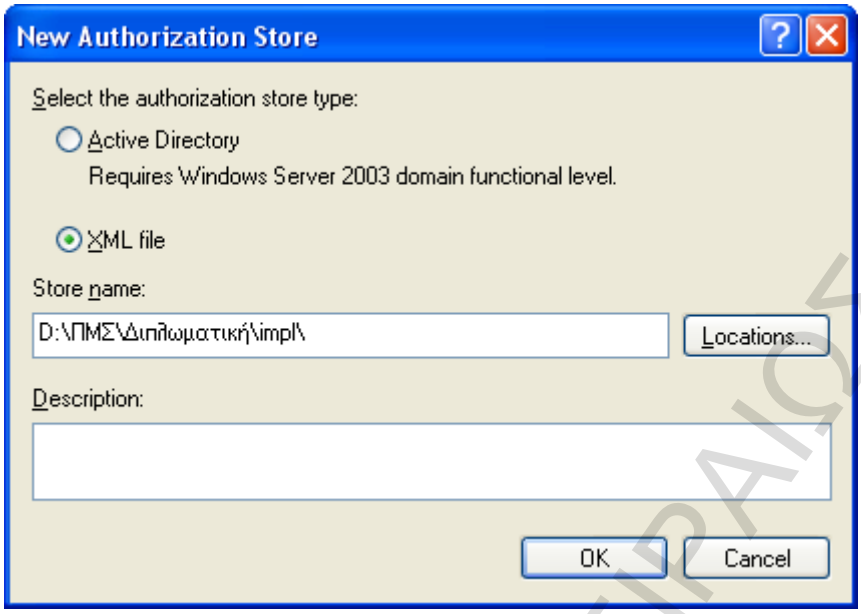
Εικόνα 14 Αλλαγή προκαθορισμένων Ρυθμίσεων

Για την έναρξη της δημιουργίας μιας πολιτικής, με το AzMan θα πρέπει να δημιουργηθεί ένα Authorization Store όπως δείχνει και η παρακάτω εικόνα. Στο περιβάλλον του Windows Server 2003, που πραγματοποιήθηκε η παρούσα μελέτη, οι δυνατοί τύποι Authorization Store είναι δύο.

I. Active Directory

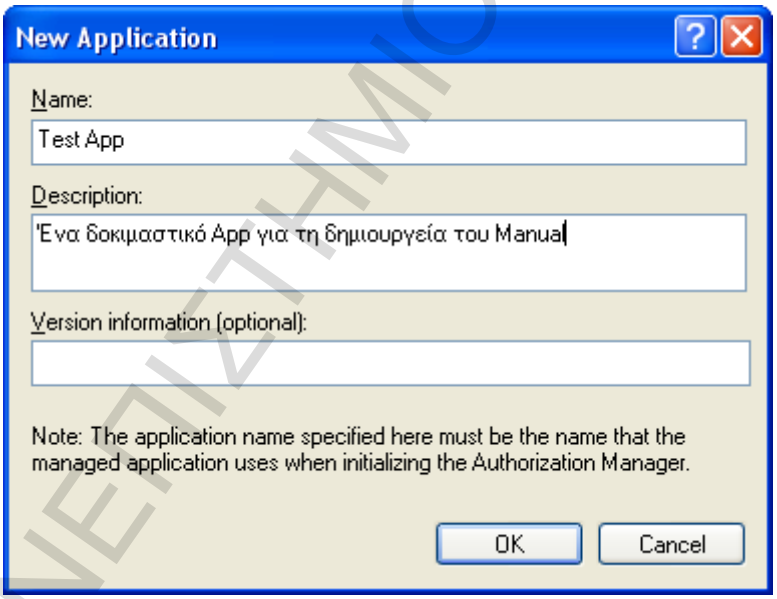
II. Αρχείο XML

Για τους λόγους που αναφέρθηκαν στη παράγραφο 4.3 επιλέγεται η δημιουργία XML Authorization Store.



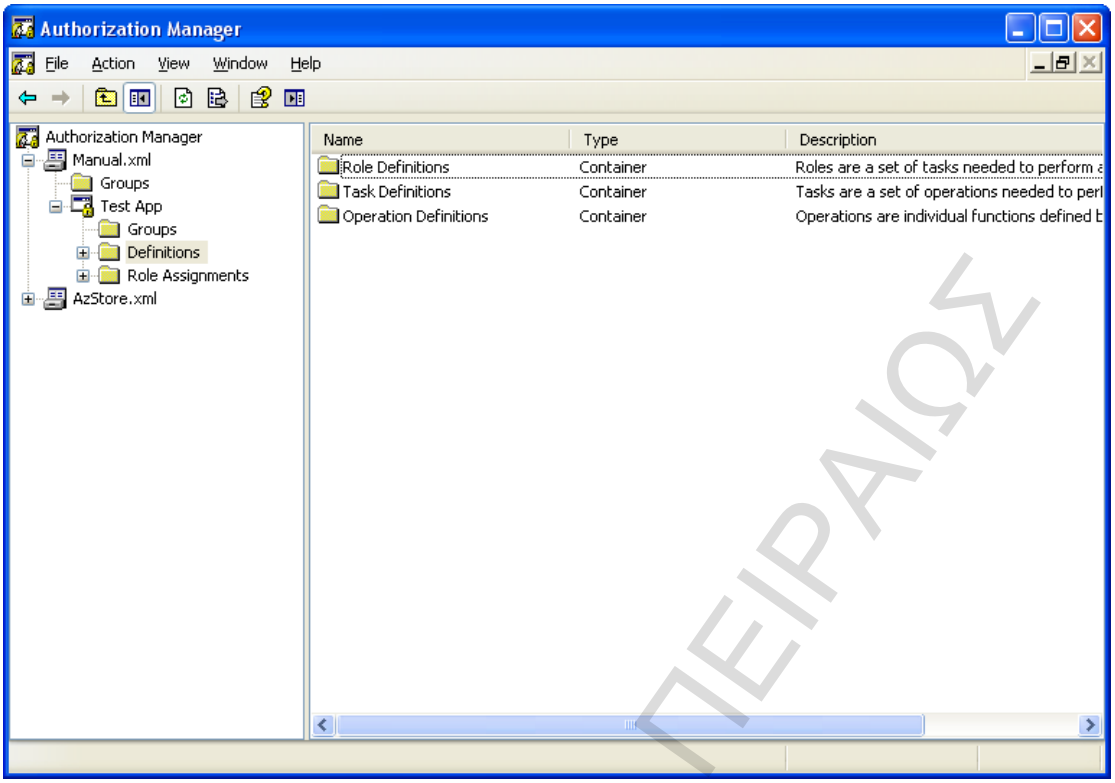
Εικόνα 15 Δημιουργία Authorization Store

Αφού έχει δημιουργηθεί επιτυχώς το Store πρέπει να δημιουργηθεί πρώτα μια εφαρμογή ώστε να αρχίσει η υλοποίηση μιας ρόλο-κεντρικής πολιτικής.



Εικόνα 16 Δημιουργία αρχικής εφαρμογής

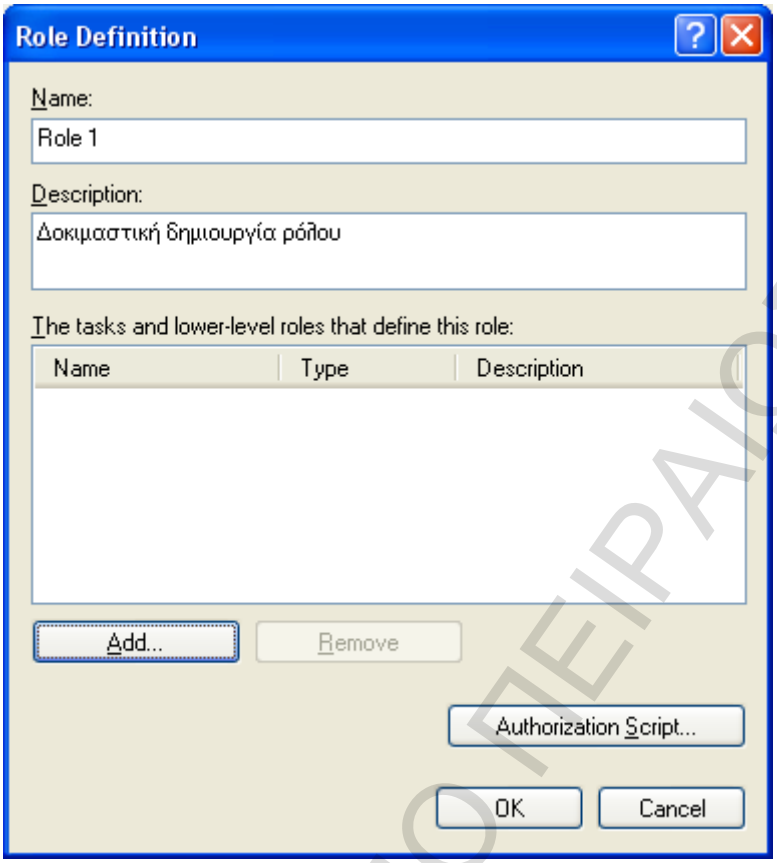
Αφού δημιουργηθεί το αρχικό App το αποτέλεσμα εμφανίζεται στην εικόνα 16. Από το σημείο αυτό μπορεί επί της ουσίας να ξεκινήσει η υλοποίηση μιας πολιτικής RBAC. Επιλέγοντας το φάκελο Definitions ανοίγονται δεξιά οι επιλογές όπως φαίνονται στην εικόνα 17. Εκεί μπορεί ο developer να δημιουργήσει τους ρόλους, τις διεργασίες (Operation) που αντιπροσωπεύουν τα δικαιώματα του κλασσικού RBAC, καθώς και τις εργασίες (Task) τις οποίες μπορούν να πραγματοποιήσουν οι ρόλοι.



Εικόνα 17 Authorization Manager Definitions Menu

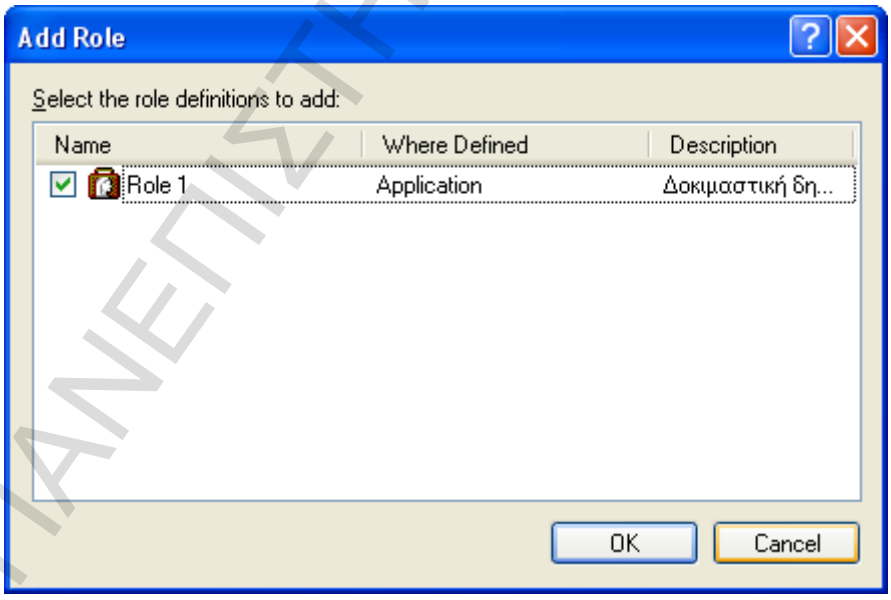
Ο developer κάνοντας δεξί κλικ στο φάκελο role definitions μπορεί να επιλέξει τη δημιουργία ενός νέου ρόλου. Η φόρμα που καλείται να συμπληρώσει είναι η εμφανιζόμενη στην εικόνα 18 και τα πεδία που συμπληρώνονται είναι:

- το όνομα του ρόλου,
- η περιγραφή του ρόλου,
- η δυνατότητα απονομής μεμονωμένων προνομίων πρόσβασης,
- η σύνδεση με ένα υφιστάμενο Authorization Script.



Εικόνα 18 Δημιουργία Ρόλου

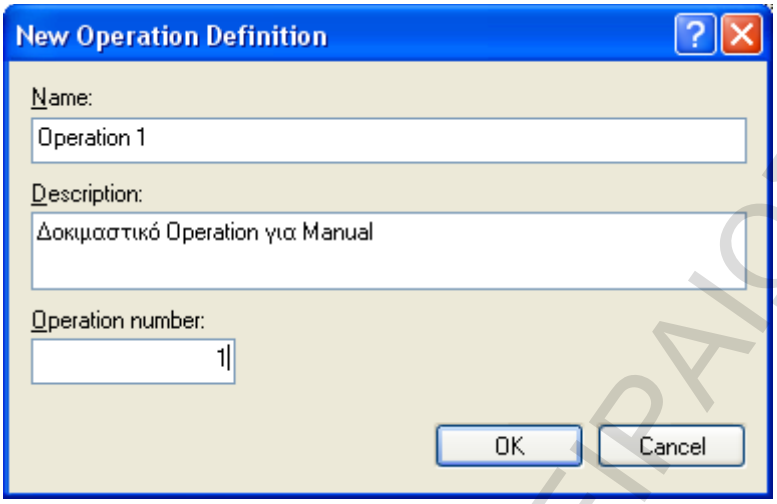
Η εικόνα 19 αποτυπώνει το αποτέλεσμα της δημιουργίας ενός ρόλου.



Εικόνα 19 Αποτύπωση δημιουργίας Νέου Ρόλου

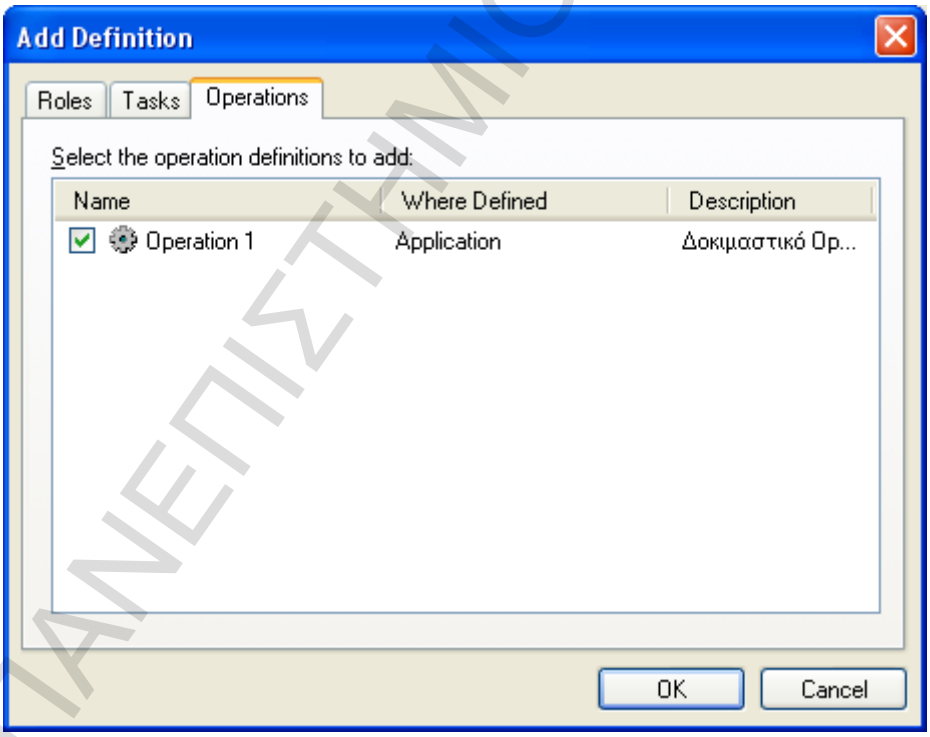
Βασικό κομμάτι για την ορθή υλοποίηση της πολιτικής είναι η σωστή δημιουργία των Operations. Τα Operations δημιουργούνται μόνο από τον developer κάνοντας δεξί κλικ στο φάκελο Operation Definition και επιλέγοντας τη δημιουργία νέου Operation. Η φόρμα που

εμφανίζεται έχει τη μορφή της εικόνας 20. Επιλέγεται το όνομα, δίνεται μια σύντομη περιγραφή και ένας **μοναδικός αριθμός** στο Operation που δημιουργείται.



Εικόνα 20 Δημιουργία Operation

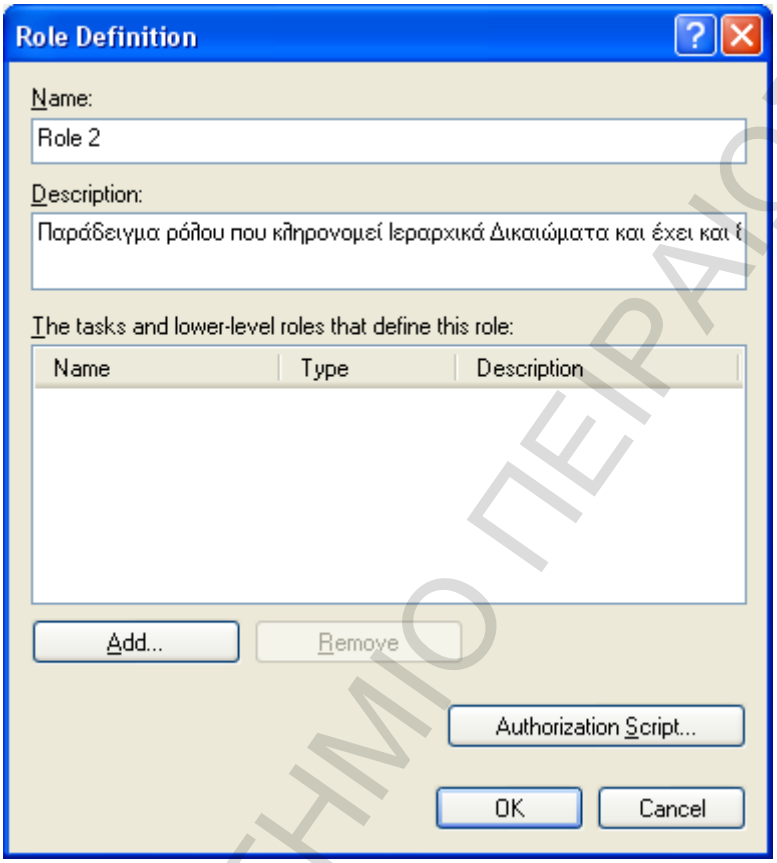
Μετά από τη δημιουργία η εικόνα όπως διαμορφώνεται για τα υπάρχοντα operation είναι η παρακάτω (εικόνα 21)



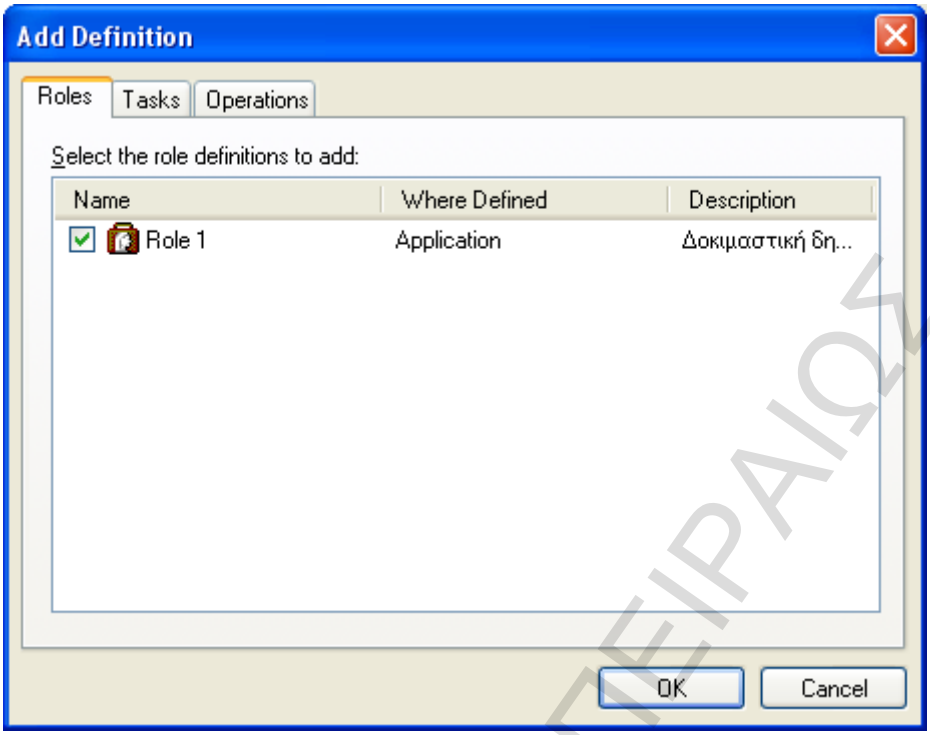
Εικόνα 21 Διαθέσιμα Operations

Ένα πολύ σημαντικό πλεονέκτημα του Authorization Manager είναι το ότι δίνει τη δυνατότητα, όπως φαίνεται από τις εικόνες που ακολουθούν ώστε να επιτυγχάνεται συμμόρφωση με τις κατευθυντήριες γραμμές του ιεραρχικού ρόλο-κεντρικού μοντέλου. Πιο

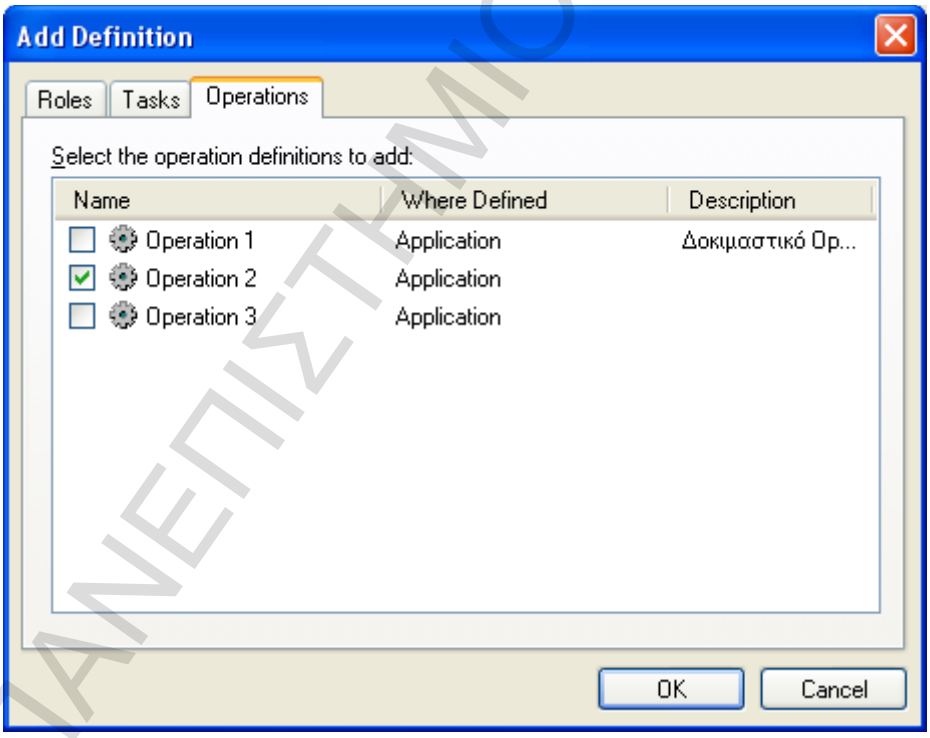
συγκεκριμένα το API προσφέρει τη δυνατότητα, ρόλοι να κληρονομούν ρόλους, κληρονομώντας το σύνολο των δικαιωμάτων που έχουν οι δεύτεροι.



Εικόνα 22 Δημιουργία ρόλου για την εφαρμογή του Ιεραρχικού RBAC

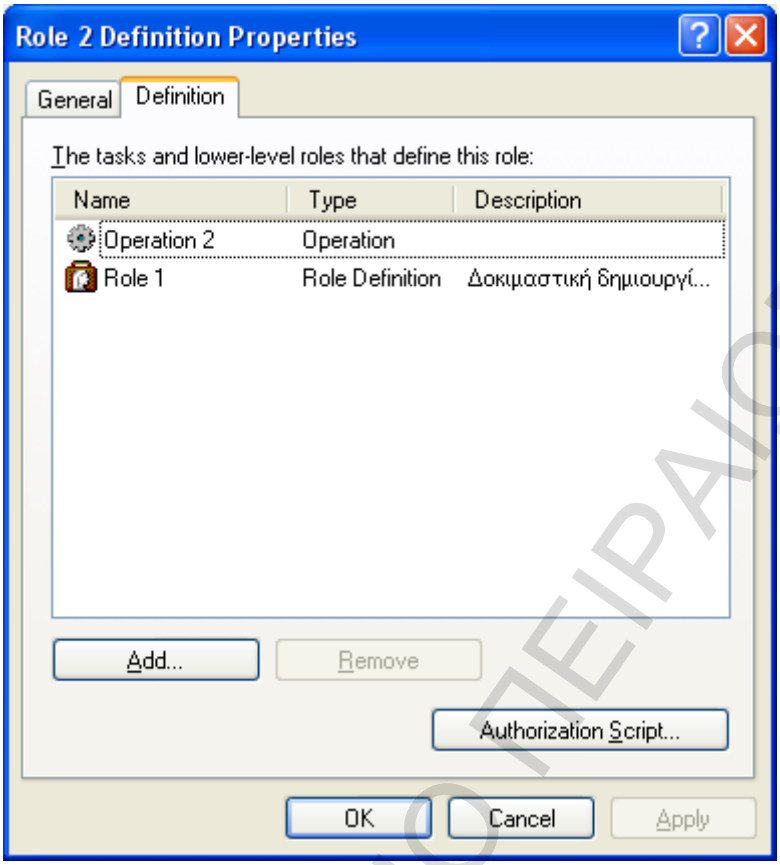


Εικόνα 23 Στιγμιότυπο εκχώρησης ρόλου σε ρόλο



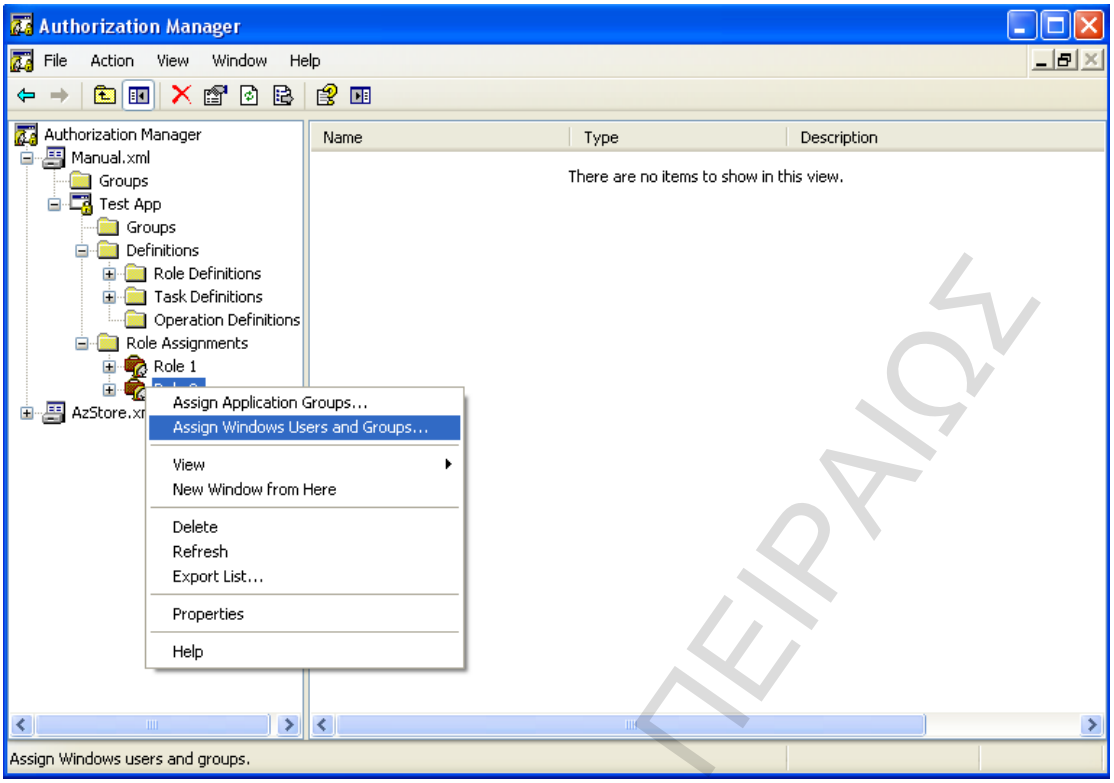
Εικόνα 24 Στιγμιότυπο εκχώρησης μεμονωμένων δικαιωμάτων πρόσβασης

Το αποτέλεσμα που προκύπτει από τις παραπάνω ενέργειες (εικόνες 23,24) είναι το ακόλουθο. Ο Role 2 έχει δικαίωμα πρόσβασης στο Operation 2 και έχει κληρονομήσει όλες τις προσβάσεις του Role 1.

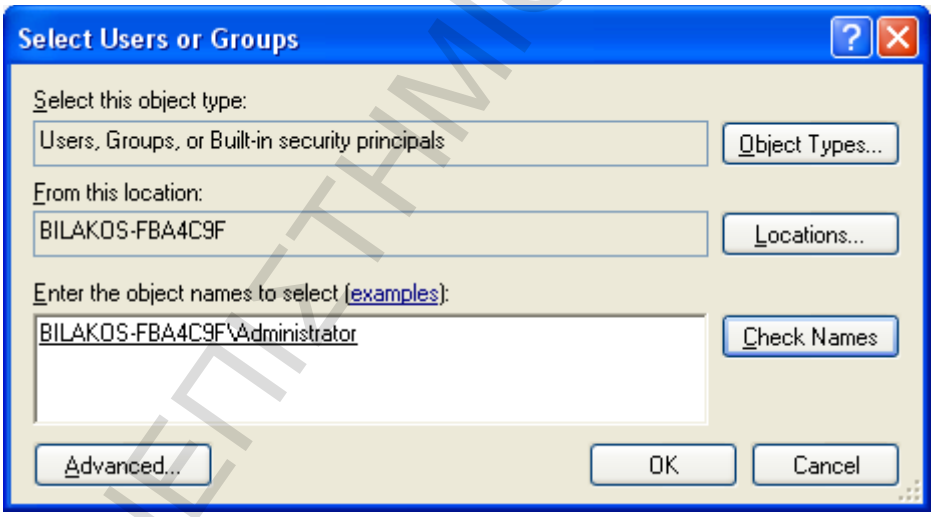


Εικόνα 25 Στιγμιότυπο Προσβάσεων Ιεραρχικού Ρόλου

Τέλος, η σύντομη παρουσίαση του Authorization Manager ολοκληρώνεται με την αναφορά στη σύνδεση των χρηστών με τους ρόλους, το γνωστό role assignment. Το AzMan παρέχει αυτή τη λειτουργία τόσο στο Developer mode τόσο και στο Administrative mode. Άλλωστε η εκχώρηση των χρηστών με ρόλους είναι η κύρια εργασία του διαχειριστή. Οι δυνατότητες σύνδεσης αφορούν υφιστάμενους μεμονωμένους χρήστες ή ομάδες χρηστών (Groups) του Active Directory με υπάρχοντες ρόλους. Η ανάθεση ρόλου πραγματοποιείται κάνοντας δεξί κλικ στο ρόλο και επιλέγοντας Assign Windows Users and Groups.

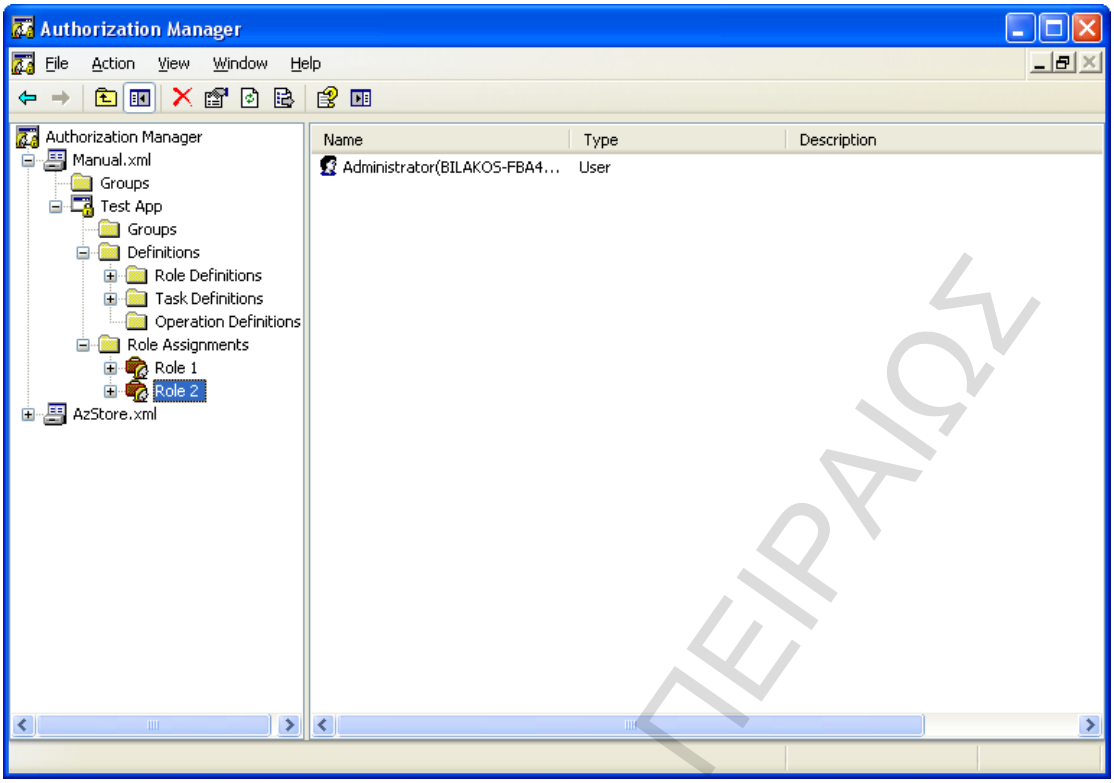


Εικόνα 26 Ανάθεση Ρόλου σε Χρήστη (-ες) [1/2]



Εικόνα 27 Ανάθεση Ρόλου σε Χρήστη (-ες) [2/2]

Τα αποτελέσματα των ενεργειών που απεικονίζονται στις παραπάνω εικόνες, φαίνονται στη εικόνα 28 όπου ο Role 2 είναι συνδεδεμένος με το χρήστη Administrator.



Εικόνα 28 Αποτύπωση Role Assignment

Πλεονεκτήματα του AzMan

Το Authorization Manager είναι ένα εργαλείο, το οποίο είναι ιδιαίτερα εύχρηστο και παρέχει στο χρήστη τη δυνατότητα της υλοποίησης μιας RBAC πολιτικής ελέγχου πρόσβασης. Συνοπτικά τα κυριότερα χαρακτηριστικά-πλεονεκτήματά του είναι:

- ✓ Built in στα Windows
- ✓ Φιλική Διεπαφή
- ✓ Εύκολο στη Χρήση
- ✓ Συμμόρφωση με το Βασικό και Ιεραρχικό RBAC
- ✓ Διευκόλυνση δημιουργίας εφαρμογών που αξιοποιούν την πολιτική ελέγχου πρόσβασης
- ✓ Πολλαπλοί τύποι αρχείων αποθήκευσης του AzStore
- ✓ Εύκολη μεταφορά από το δοκιμαστικό σε πραγματικό περιβάλλον εργασίας

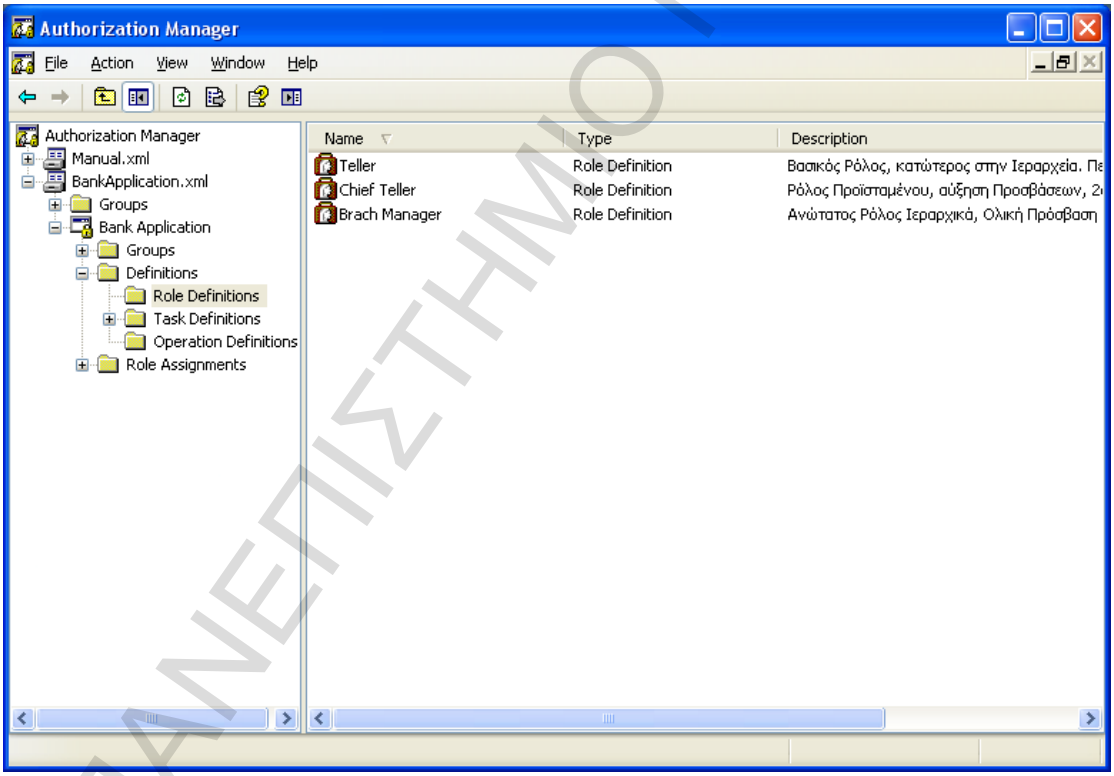
4.7 Υλοποίηση της σχεδιασθείσας πολιτικής στο Authorization Manager

Στα υποκεφάλαια 4.3 & 4.4 γίνεται εκτενής τεκμηρίωση και ανάλυση των δεδομένων της λειτουργίας ενός τραπεζικού καταστήματος. Προσδιορίζονται οι ρόλοι, οι προσβάσεις και τα δικαιώματα κάθε ρόλου. Για την υλοποίηση όσων περιγράφονται στα προαναφερόμενα αναπτύχθηκε η εφαρμογή, της οποίας η υλοποίηση ακολουθεί.

Αρχικά δημιουργείται ένα Authorization Store σε xml μορφή. Ονομάζεται Bank Application. Για την ομαλή ροή της διαδικασίας δημιουργείται το βασικό App του συστήματος το Bank Application.

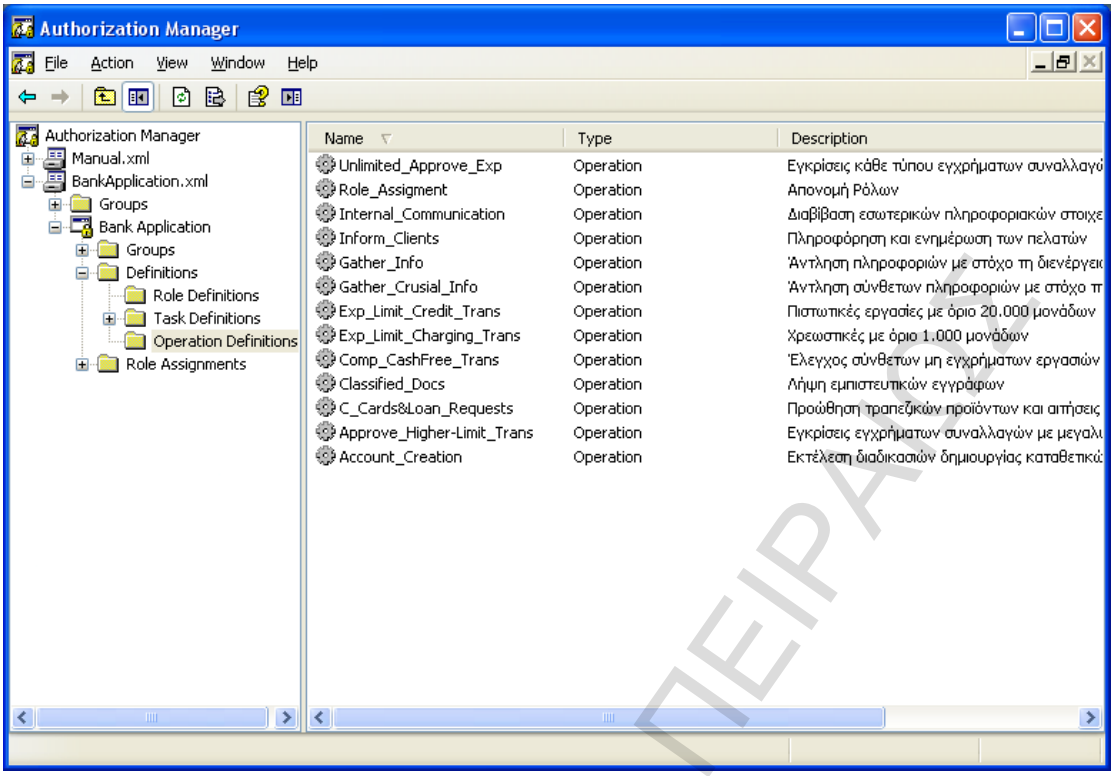
Πάνω σε αυτό το App θα αναπτυχθεί και η εφαρμογή η οποία θα πραγματοποιεί τον έλεγχο πρόσβασης σύμφωνα με τα όσα πρόκειται δημιουργηθούν στο API του AzMan.

Εν συνεχεία, όπως φαίνεται στο στιγμιότυπο της εικόνας 29 δημιουργούνται οι 3 ρόλοι του συστήματος όπως περιγράφηκαν στη παράγραφο 4.4.2.



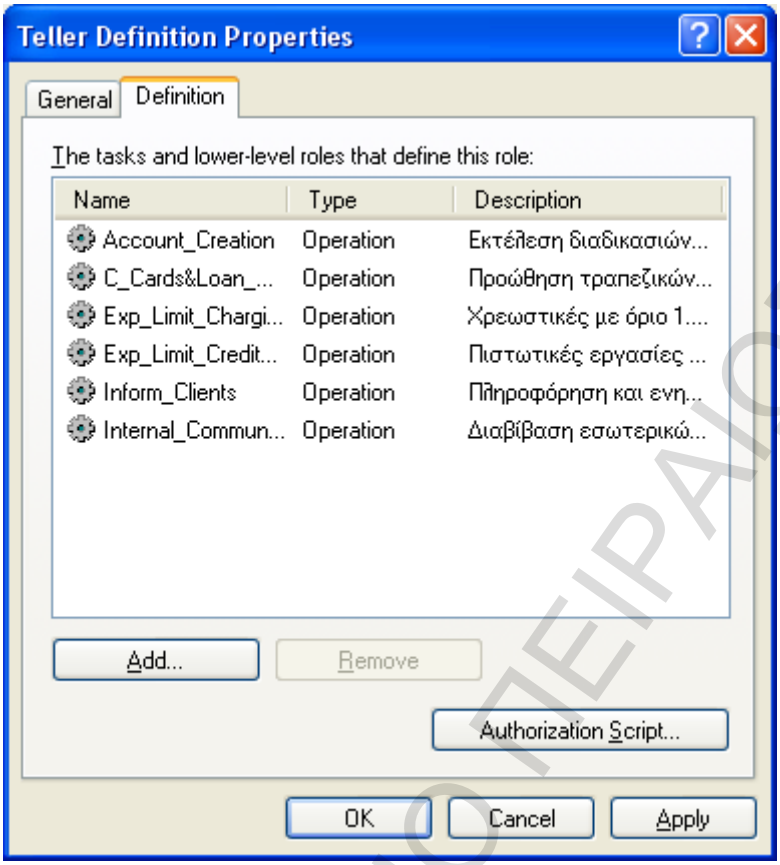
Εικόνα 29 Στιγμιότυπο αποτύπωσης Ρόλων

Το επόμενο βήμα είναι η ένταξη των εργασιών στο σύστημα. Δημιουργούνται λοιπόν οι 13 εργασίες που περιγράφηκαν στην παράγραφο 4.4.1 ως χωριστά Operations και παίρνουν μοναδικούς αύξοντες αριθμούς από το 1 έως το 13



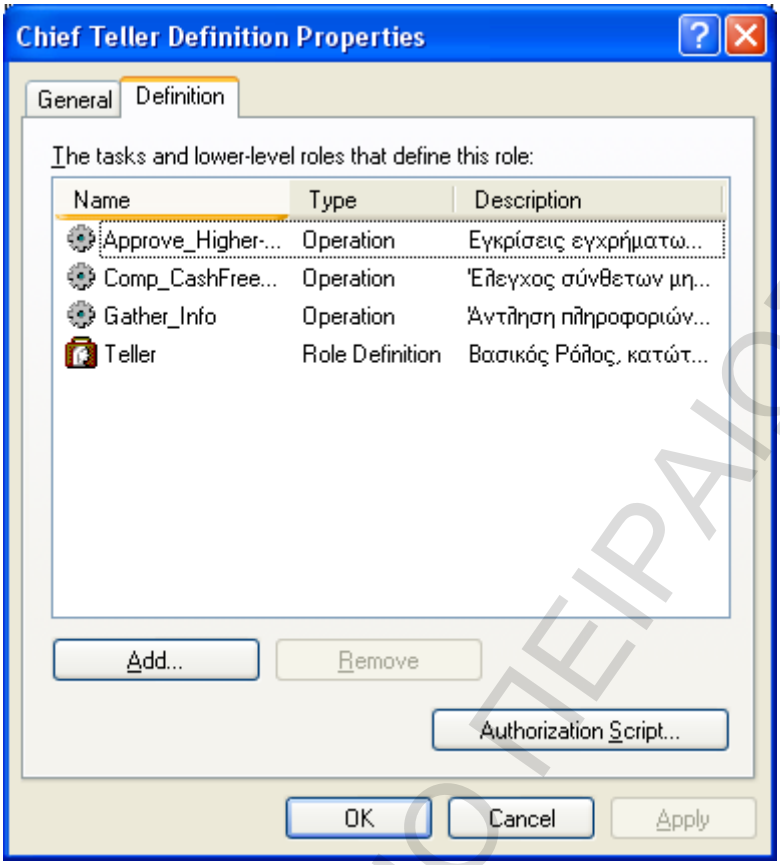
Εικόνα 30 Στιγμιότυπο αποτύπωσης Εργασιών

Η συνέχεια της διαδικασίας αφορά στην ανάθεση προνομίων στους ρόλους. Η ανάθεση όπως έχει τονιστεί και νωρίτερα, θα πραγματοποιηθεί ιεραρχικά. Η αρχή γίνεται με την ανάθεση των προνομίων στο βασικό ρόλο του Teller, ο οποίος αποκτά προσβάσεις στα 6 Operations. Το αποτέλεσμα αποτυπώνεται στην εικόνα 31.



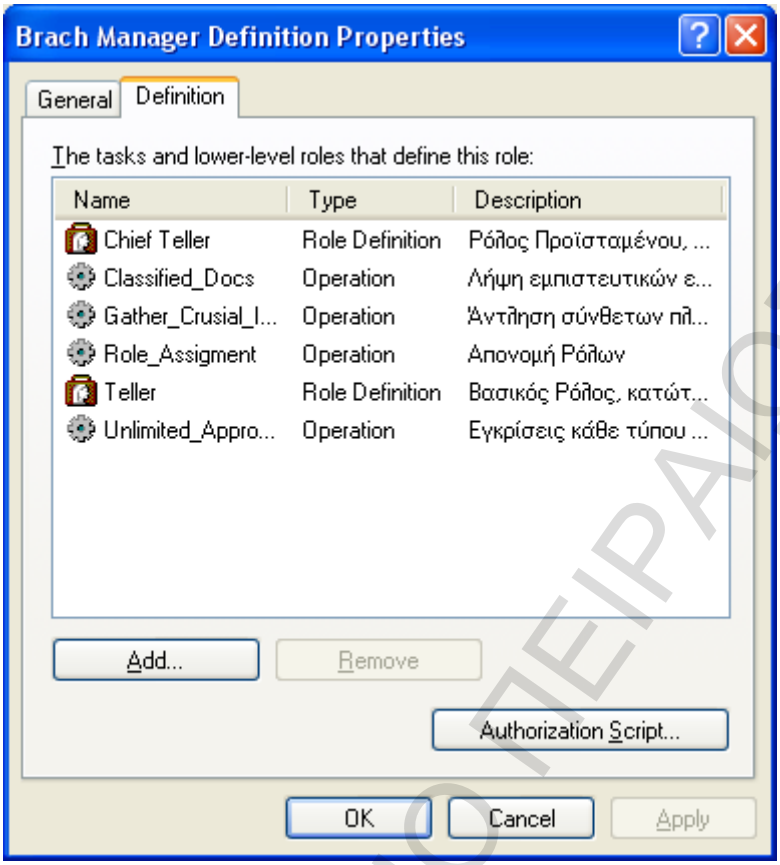
Εικόνα 31 Αποτύπωση Προνομίων Teller

Προχωρώντας στην ανάθεση των προνομίων στο ρόλο του Chief Teller, αρχικά επιλέγεται η σύνδεση του με το ρόλο του Teller. Επί της ουσίας ο 2^{ος} τη τάξει ρόλος κληρονομεί όλα τα προνόμια του βασικού ρόλου. Εν συνεχεία, του ανατίθενται επιπλέον προνόμια και το αποτέλεσμα αποτυπώνεται στην εικόνα 32.



Εικόνα 32 Αποτύπωση Προνομίων Chief Teller

Τέλος, το στάδιο της ανάθεσης προνομίων ολοκληρώνεται με τον εμπλουτισμό του ρόλου Branch Manager. Ο διαχειριστικός ρόλος, αποκτά πρόσβαση σε όλες τις εργασίες που εκτελούνται από το σύστημα κληρονομώντας τα προνόμια των υφισταμένων ρόλων και μέσω της απευθείας ανάθεσης των υψηλά ισταμένων προνομίων. Το αποτέλεσμα παρουσιάζεται στην εικόνα 33.



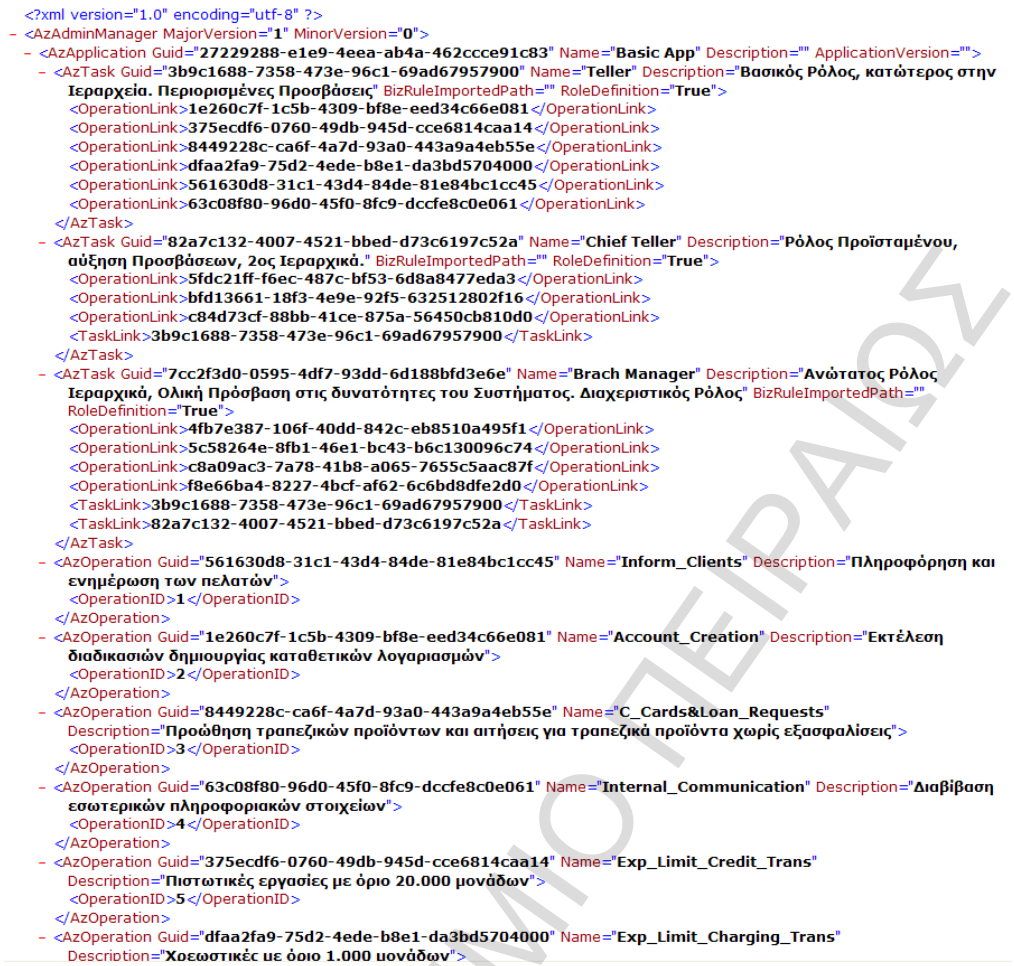
Εικόνα 33 Αποτύπωση Προνομίων Branch Manager

Με βάση λοιπόν αυτή τη πολιτική ασφάλειας ελέγχου πρόσβασης η οποία υλοποιήθηκε με χρήση του AzMan στην παράγραφο 4.9, θα παρουσιαστεί η υλοποίηση της εφαρμογής η οποία και θα πραγματοποιεί τον έλεγχο πρόσβασης βάσει της πολιτικής.

Τελευταίο βήμα της παρούσας διαδικασίας, είναι η ανάθεση ρόλων σε υφιστάμενους χρήστες των windows. Τη διαδικασία αυτή μπορεί να τη διαμορφώσει ο σχεδιαστής και στη συνεχεία ο διαχειριστής του συστήματος μέσα από την εφαρμογή, μπορεί να την παραμετροποιεί.

4.8 Συμμόρφωση με τη πολιτική ελέγχου πρόσβασης του οργανισμού

Στις παρακάτω εικόνες παρατίθεται το xml αρχείο που παράγει το AzMan μετά την ολοκλήρωση της συγγραφής της πολιτικής. Αρχικά ορίζει την εφαρμογή που θα τρέχουν οι χρήστες και εν συνεχεία ορίζει τις προσβάσεις, μέσω της σύνδεσης με το Operation Link, που θα έχει ο κάθε ρόλος. Στην αποτύπωση ακολουθούνται τα operations με το μοναδικό SID και το Operation Link τους. Τέλος στην εικόνα 34 φαίνεται το περιεχόμενο των χρηστών που έχουν αντιστοιχηθεί με τους υφιστάμενους ρόλους.



```
<?xml version="1.0" encoding="utf-8" ?>
- <AzAdminManager MajorVersion="1" MinorVersion="0">
  - <AzApplication Guid="27229288-e1e9-4eea-ab4a-462ccce91c83" Name="Basic App" Description="" ApplicationVersion="">
    - <AzTask Guid="3b9c1688-7358-473e-96c1-69ad67957900" Name="Teller" Description="Βασικός Ρόλος, κατώτερος στην
      Ιεραρχεία. Περιορισμένες Προσβάσεις" BizRuleImportedPath="" RoleDefinition="True">
      <OperationLink>1e260c7f-1c5b-4309-bf8e-eed34c66e081</OperationLink>
      <OperationLink>375ecdf6-0760-49db-945d-cce6814caa14</OperationLink>
      <OperationLink>8449228c-ca6f-4a7d-93a0-443a9a4eb55e</OperationLink>
      <OperationLink>dfaa2fa9-75d2-4ede-b8e1-da3bd5704000</OperationLink>
      <OperationLink>561630d8-31c1-43d4-84de-81e84bc1cc45</OperationLink>
      <OperationLink>63c08f80-96d0-45f0-8fc9-dccfe8c0e061</OperationLink>
    </AzTask>
    - <AzTask Guid="82a7c132-4007-4521-bbed-d73c6197c52a" Name="Chief Teller" Description="Ρόλος Προϊσταμένου,
      αύξηση Προσβάσεων, 2ος Ιεραρχικά." BizRuleImportedPath="" RoleDefinition="True">
      <OperationLink>5fdc21ff-f6ec-487c-bf53-6d8a8477eda3</OperationLink>
      <OperationLink>bfd13661-18f3-4e9e-92f5-632512802f16</OperationLink>
      <OperationLink>c84d73cf-88bb-41ce-875a-56450cb810d0</OperationLink>
      <TaskLink>3b9c1688-7358-473e-96c1-69ad67957900</TaskLink>
    </AzTask>
    - <AzTask Guid="7cc2f3d0-0595-4df7-93dd-6d188bfd3e6e" Name="Brach Manager" Description="Ανώτατος Ρόλος
      Ιεραρχικά, Ολική Πρόσβαση στις δυνατότητες του Συστήματος. Διαχεριστικός Ρόλος" BizRuleImportedPath=""
      RoleDefinition="True">
      <OperationLink>4fb7e387-106f-40dd-842c-eb8510a495f1</OperationLink>
      <OperationLink>5c58264e-8fb1-46e1-bc43-b6c130096c74</OperationLink>
      <OperationLink>c8a09ac3-7a78-41b8-a065-7655c5aac87f</OperationLink>
      <OperationLink>f8e66ba4-8227-4bcf-af62-6c6bd8dfe2d0</OperationLink>
      <TaskLink>3b9c1688-7358-473e-96c1-69ad67957900</TaskLink>
      <TaskLink>82a7c132-4007-4521-bbed-d73c6197c52a</TaskLink>
    </AzTask>
    - <AzOperation Guid="561630d8-31c1-43d4-84de-81e84bc1cc45" Name="Inform_Clients" Description="Πληροφόρηση και
      ενημέρωση των πελατών">
      <OperationID>1</OperationID>
    </AzOperation>
    - <AzOperation Guid="1e260c7f-1c5b-4309-bf8e-eed34c66e081" Name="Account_Creation" Description="Εκτέλεση
      διαδικασιών δημιουργίας καταθετικών λογαριασμών">
      <OperationID>2</OperationID>
    </AzOperation>
    - <AzOperation Guid="8449228c-ca6f-4a7d-93a0-443a9a4eb55e" Name="C_Cards&Loan_Requests"
      Description="Προώθηση τραπεζικών προϊόντων και αιτήσεις για τραπεζικά προϊόντα χωρίς εξασφαλίσεις">
      <OperationID>3</OperationID>
    </AzOperation>
    - <AzOperation Guid="63c08f80-96d0-45f0-8fc9-dccfe8c0e061" Name="Internal_Communication" Description="Διαβίβαση
      εσωτερικών πληροφοριακών στοιχείων">
      <OperationID>4</OperationID>
    </AzOperation>
    - <AzOperation Guid="375ecdf6-0760-49db-945d-cce6814caa14" Name="Exp_Limit_Credit_Trans"
      Description="Πιστωτικές εργασίες με όριο 20.000 μονάδων">
      <OperationID>5</OperationID>
    </AzOperation>
    - <AzOperation Guid="dfaa2fa9-75d2-4ede-b8e1-da3bd5704000" Name="Exp_Limit_Charging_Trans"
      Description="Χοεωστικές με όριο 1.000 μονάδων">
```


Εικόνα 34 Αποτύπωση παραγόμενου xml αρχείου [1/2]

```
- <AzOperation Guid="375ecdf6-0760-49db-945d-cce6814caa14" Name="Exp_Limit_Credit_Trans"
    Description="Πιστωτικές εργασίες με όριο 20.000 μονάδων">
    <OperationID>5</OperationID>
  </AzOperation>
- <AzOperation Guid="dfaa2fa9-75d2-4ede-b8e1-da3bd5704000" Name="Exp_Limit_Charging_Trans"
    Description="Χρεωστικές με όριο 1.000 μονάδων">
    <OperationID>6</OperationID>
  </AzOperation>
- <AzOperation Guid="bfd13661-18f3-4e9e-92f5-632512802f16" Name="Approve_Higher-Limit_Trans"
    Description="Εγκρίσεις εγχρήματων συναλλαγών με μεγαλύτερο όριο">
    <OperationID>7</OperationID>
  </AzOperation>
- <AzOperation Guid="c84d73cf-88bb-41ce-875a-56450cb810d0" Name="Comp_CashFree_Trans" Description="Ελεγχος
    σύνθετων μη εγχρήματων εργασιών">
    <OperationID>8</OperationID>
  </AzOperation>
- <AzOperation Guid="5fdc21ff-f6ec-487c-bf53-6d8a8477eda3" Name="Gather_Info" Description="Άντληση πληροφοριών
    με στόχο τη διενέργεια κατασταλτικών ελέγχων, σε ενέργειες που έχουν πραγματοποιηθεί από τους Tellers, που
    εποπτεύει">
    <OperationID>9</OperationID>
  </AzOperation>
- <AzOperation Guid="c8a09ac3-7a78-41b8-a065-7655c5aac87f" Name="Unlimited_Approve_Exp" Description="Εγκρίσεις
    κάθε τύπου εγχρήματων συναλλαγών χωρίς όριο">
    <OperationID>10</OperationID>
  </AzOperation>
- <AzOperation Guid="f8e66ba4-8227-4bcf-af62-6c6bd8dfe2d0" Name="Classified_Docs" Description="Λήψη
    εμπιστευτικών εγγράφων">
    <OperationID>11</OperationID>
  </AzOperation>
- <AzOperation Guid="4fb7e387-106f-40dd-842c-eb8510a495f1" Name="Gather_Crusial_Info" Description="Άντληση
    σύνθετων πληροφοριών με στόχο τη διενέργεια κατασταλτικών ελέγχων, σε ενέργειες που έχουν πραγματοποιηθεί
    από τους Tellers & Chief Tellers του υποκαταστήματος">
    <OperationID>12</OperationID>
  </AzOperation>
- <AzOperation Guid="5c58264e-8fb1-46e1-bc43-b6c130096c74" Name="Role_Assigment" Description="Απονομή Ρόλων">
    <OperationID>13</OperationID>
  </AzOperation>
- <AzRole Guid="4beb7873-ef89-46dd-9d72-6c26dcc37ca0" Name="Brach Manager">
    <TaskLink>7cc2f3d0-0595-4df7-93dd-6d188bfd3e6e</TaskLink>
    <Member>S-1-5-21-1123561945-562591055-1801674531-1003</Member>
  </AzRole>
- <AzRole Guid="4b85c6a4-0ec1-433b-9f28-939cc169acf6" Name="Chief Teller">
    <TaskLink>82a7c132-4007-4521-bbed-d73c6197c52a</TaskLink>
    <Member>S-1-5-21-1123561945-562591055-1801674531-1008</Member>
  </AzRole>
- <AzRole Guid="20add645-4232-4204-92a9-0b52143c594c" Name="Teller">
    <TaskLink>3b9c1688-7358-473e-96c1-69ad67957900</TaskLink>
    <Member>S-1-5-21-1123561945-562591055-1801674531-1009</Member>
  </AzRole>
</AzApplication>
</AzAdminManager>
```

Εικόνα 35 Αποτύπωση παραγόμενου xml αρχείου [2/2]

Πέρα όμως από την ολοκλήρωση της συγγραφής και της υλοποίησης του ελέγχου πρόσβασης μέσω του AzMan και της εφαρμογής που θα χρησιμοποιούν οι εργαζόμενοι, απομένει ένα σημαντικό κομμάτι για θεωρηθεί η πολιτική πλήρης. Θα πρέπει λοιπόν, να είναι σύμφωνη με την υφιστάμενη πολιτική ασφάλειας του οργανισμού. Αυτή η πολιτική ασφάλειας σε τέτοιου τύπου οργανισμούς αποτυπώνεται σε εμπιστευτικά εσωτερικά έγγραφα και συντάσσεται από τον security officer του οργανισμού. Ως εκ τούτου δεν είναι δυνατή η παρουσίαση των εν λόγω κειμένων και αντί αυτών παρουσιάζεται το xml που δημιουργείται βάσει αυτών.

```
<?xml version="1.0" encoding="utf-8" ?>
- <Access_Control_Policy>
  - <Role_Definition>
      <Name>Teller</Name>
    - <AccessIn>
        <OperationLink>Inform_Clients</OperationLink>
        <OperationLink>Account_Creation</OperationLink>
        <OperationLink>C_Cards_Loan_Requests</OperationLink>
        <OperationLink>Internal_Communication</OperationLink>
        <OperationLink>Exp_Limit_Credit_Trans</OperationLink>
        <OperationLink>Exp_Limit_Charging_Trans</OperationLink>
      </AccessIn>
    </Role_Definition>
  - <Role_Definition>
      <Name>Chief_Teller</Name>
    - <AccessIn>
        <OperationLink>Inform_Clients</OperationLink>
        <OperationLink>Account_Creation</OperationLink>
        <OperationLink>C_Cards_Loan_Requests</OperationLink>
        <OperationLink>Internal_Communication</OperationLink>
        <OperationLink>Exp_Limit_Credit_Trans</OperationLink>
        <OperationLink>Exp_Limit_Charging_Trans</OperationLink>
        <OperationLink>Approve_Higher-Limit_Trans</OperationLink>
        <OperationLink>Comp_CashFree_Trans</OperationLink>
        <OperationLink>Gather_Info</OperationLink>
      </AccessIn>
    </Role_Definition>
  - <Role_Definition>
      <Name>Branch_Manager</Name>
    - <AccessIn>
        <OperationLink>Inform_Clients</OperationLink>
        <OperationLink>Account_Creation</OperationLink>
        <OperationLink>C_Cards_Loan_Requests</OperationLink>
        <OperationLink>Internal_Communication</OperationLink>
        <OperationLink>Exp_Limit_Credit_Trans</OperationLink>
        <OperationLink>Exp_Limit_Charging_Trans</OperationLink>
        <OperationLink>Approve_Higher-Limit_Trans</OperationLink>
        <OperationLink>Comp_CashFree_Trans</OperationLink>
        <OperationLink>Gather_Info</OperationLink>
        <OperationLink>Unlimited_Approve_Exp</OperationLink>
        <OperationLink>Classified_Docs</OperationLink>
        <OperationLink>Gather_Crusial_Info</OperationLink>
        <OperationLink>Role_Assigment</OperationLink>
      </AccessIn>
    </Role_Definition>
</Access_Control_Policy>
```

Εικόνα 36 Πολιτική Ελέγχου Πρόσβασης Οργανισμού

Μελετώντας τα δύο αρχεία μπορεί να εξαχθεί ένα συμπέρασμα. Η ζητούμενη πολιτική ελέγχου πρόσβασης του οργανισμού ικανοποιείται πλήρως από την υλοποίηση που έγινε μέσω του AzMan. Οι συντακτικές διαφορές δεν παίζουν ρόλο και η αυτοματοποιημένη σύγκριση είναι μη εφικτή μιας και πολλά από τα παραγόμενα στοιχεία του πρώτου xml προκύπτουν αυτόματα, ενώ του δεύτερου μετά από λογική επεξεργασία κειμένων. Σε κάθε περίπτωση η σύγκριση από μελετητή του χώρου της ΤΠΕ είναι εύκολη, καθώς σε κάθε δομική αλλαγή της πολιτικής μπορεί να γίνεται σύγκριση των xml.

Επί της καθημερινής παραγωγικής διαδικασίας κάτι τέτοιο δεν έχει νόημα καθώς οι διαχειριστές του συστήματος έχουν περιορισμένες προσβάσεις. Αυτό σημαίνει ότι δεν μπορούν να αλλάξουν τα διαθέσιμα operation και επομένως η λειτουργία του συστήματος μένει ως έχει.

Σχετικά με τους ρόλους, οι διαχειριστές έχουν πλήρη δικαιώματα, καθώς μπορούν να δημιουργήσουν ρόλο να διαγράψουν ήδη υπάρχοντα και να παραμετροποιήσουν το σύστημα ανάλογα με τις ανάγκες του υποκαταστήματος τους. Αυτό είναι και το πολύ μεγάλο πλεονέκτημα της προτεινόμενης εφαρμογής.

Υπάρχει δομημένη μια ρόλο-κεντρική πολιτική ασφάλειας η οποία δίνεται σε όλο το δίκτυο των καταστημάτων ως κατευθυντήρια γραμμή.

Παρόλα αυτά παραμένει ένα μεγάλο ερώτημα ανοιχτό. Πώς μπορεί να φανούν χρήσιμες οι πολιτικές που δημιουργούνται μέσω του AzMan; Σε αυτό το ερώτημα δίνει απάντηση η επόμενη παράγραφος κατά την οποία δημιουργείται μια εφαρμογή, στην οποία ο έλεγχος πρόσβασης διενεργείται με βάση την υφιστάμενη πολιτική του AzMan.

4.9 Εφαρμογή

Βασικός στόχος του υπεύθυνου ασφάλειας σε έναν οργανισμό είναι να μπορεί να ελέγχει εύκολα και απλά αν η σχεδιασθείσα πολιτική είναι σύμφωνη με την κεντρική πολιτική ασφάλειας του συστήματος. Η εφαρμογή που δημιουργήθηκε επιτρέπει τη διενέργεια του ελέγχου πρόσβασης, με βάση την πολιτική που υλοποιήθηκε στην παράγραφο 4.7. Επιτρέπει στο διαχειριστή να εκχωρεί χρήστες σε ρόλους, να δημιουργεί νέους χρήστες χωρίς όμως να μπορεί να επεξεργαστεί την πολιτική ελέγχου πρόσβασης. Με τον τρόπο αυτό εξασφαλίζεται η διατήρηση της συμφωνίας με την πολιτική ασφάλειας του οργανισμού. Βασικοί στόχοι της εφαρμογής, είναι:

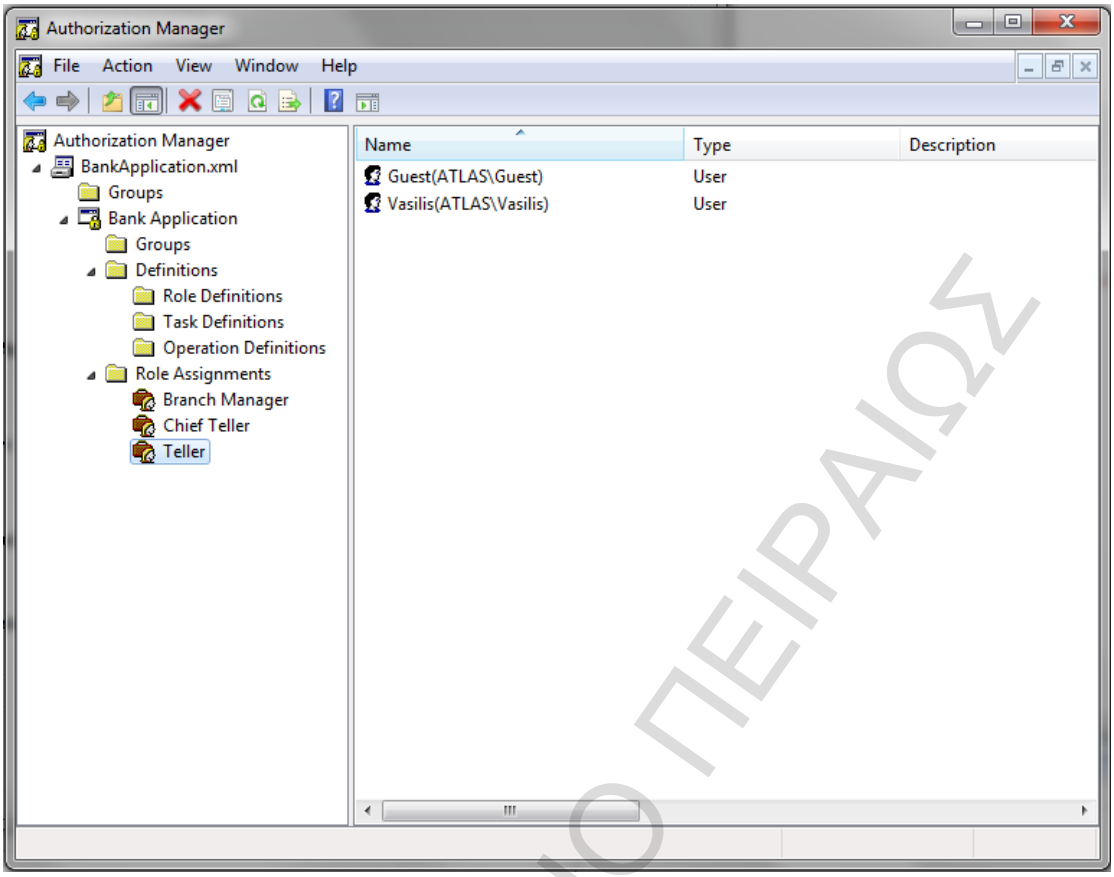
- ✓ Η Διενέργεια Ελέγχου Πρόσβασης
- ✓ Η Ρόλο-κεντρική υλοποίηση
- ✓ Η Αύξηση του Επιπέδου Ασφαλείας
- ✓ Η Μείωση Διαχειριστικής Πολυπλοκότητας

Υποθέτουμε ότι όλοι οι υπάλληλοι για να μπορέσουν να έχουν πρόσβαση σε πόρους, χρησιμοποιούν μια εφαρμογή η οποία τους δίνει πρόσβαση στους πόρους ώστε να πραγματοποιήσουν τις εργασίες τους. Η ιδέα για τη δημιουργία της εφαρμογής βασίζεται στη χρήση του token που προκύπτει μετά την είσοδο του χρήστη στο Λ.Σ. των windows. Στον κώδικα της εφαρμογής χρησιμοποιείται αυτό το token ώστε να συνδεθεί ο χρήστης με την υφιστάμενη πολιτική, που έχει υλοποιηθεί στο authorization manager. Στην παρούσα παράγραφο λοιπόν, παρουσιάζεται μια εφαρμογή που γράφτηκε σε γλώσσα C# στη πλατφόρμα .NET του Visual Studio και πραγματοποιεί έλεγχο πρόσβασης βασισμένο στη πολιτική που υλοποιήθηκε. Ο χρήστης αναγνωρίζεται αυτόματα από το log in που κάνει στο λειτουργικό σύστημα και μόλις ανοίξει την εφαρμογή αποκτά τις προσβάσεις του ρόλου που ανήκει.

Έστω λοιπόν ότι ο χρήστης Vasilis ανήκει στο ρόλο Teller. Μόλις ανοίξει την εφαρμογή θα έχει διαθέσιμες τις προσβάσεις που φαίνονται στη παρακάτω εικόνα και αντιστοιχούν στα «ενεργά» πλήκτρα (operations 1έως και 6)

Εικόνα 37 Προσβάσεις εφαρμογής ρόλου Teller

Όπως βλέπουμε, στην εφαρμογή υπάρχουν και τα 13 operations που δημιουργήθηκαν στη πολιτική και χρειάζονται για τη λειτουργία του υποκαταστήματος. Η επόμενη εικόνα δείχνει ότι ο χρήστης ανήκει στο ρόλο Teller, (στο φάκελο Role Assignments, είναι «ενεργό» το πλήκτρο Teller).



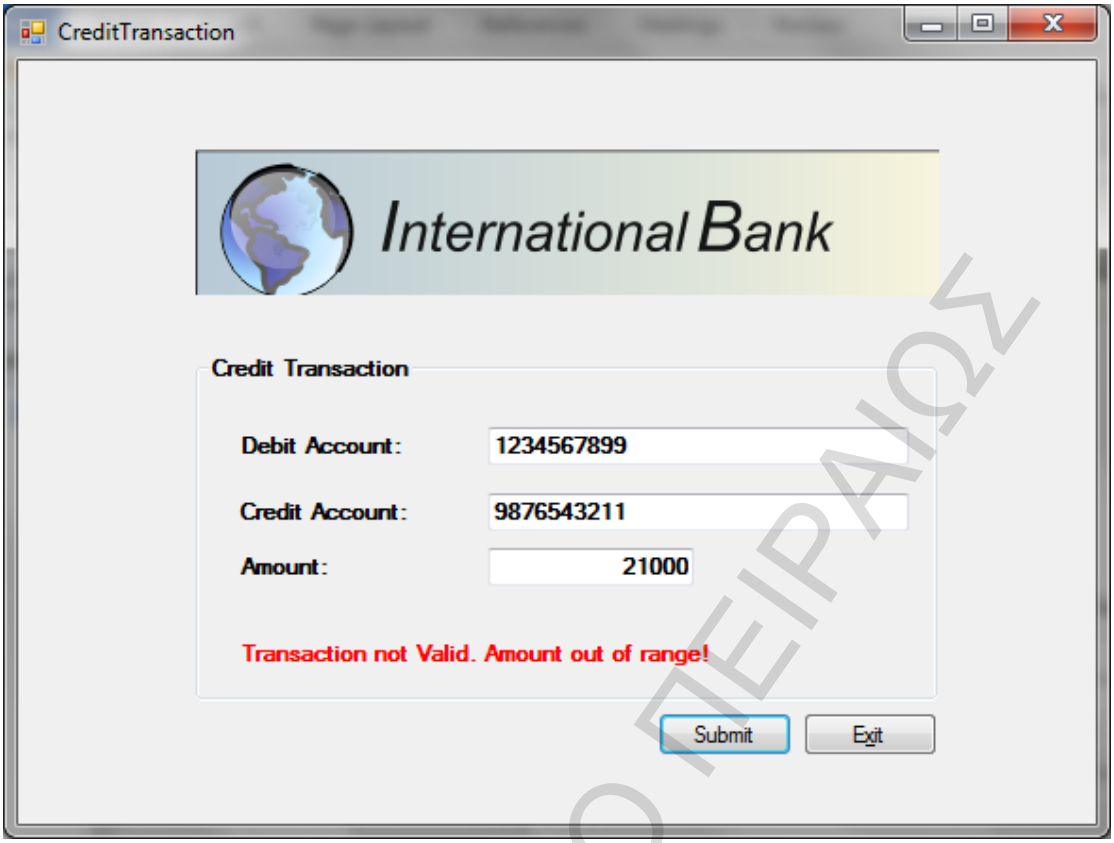
Εικόνα 38 Ανάθεση στο χρήστη Vasilis του ρόλου Teller

Ενδεικτικά, δημιουργήθηκαν κάποιες από τις εφαρμογές που μπορεί να εκτελέσει ο ρόλος Teller. Για παράδειγμα, επιλέγοντας το Operation 1 θα του ανοίξει το ακόλουθο παράθυρο, το οποίο εμφανίζει πληροφορίες για την τράπεζα.

Εικόνα 39 Στιγμιότυπο εκτέλεσης Operation 1

Επίσης, δημιουργήθηκαν φόρμες και για τα Operation 2 & 5. Το Operation 2 επιτρέπει τη δημιουργία λογαριασμού και το Operation 5 τη διενέργεια πιστωτικής συναλλαγής με όριο. Κατά το Operation 5 διενεργείται και έλεγχος που αφορά το ποσό.

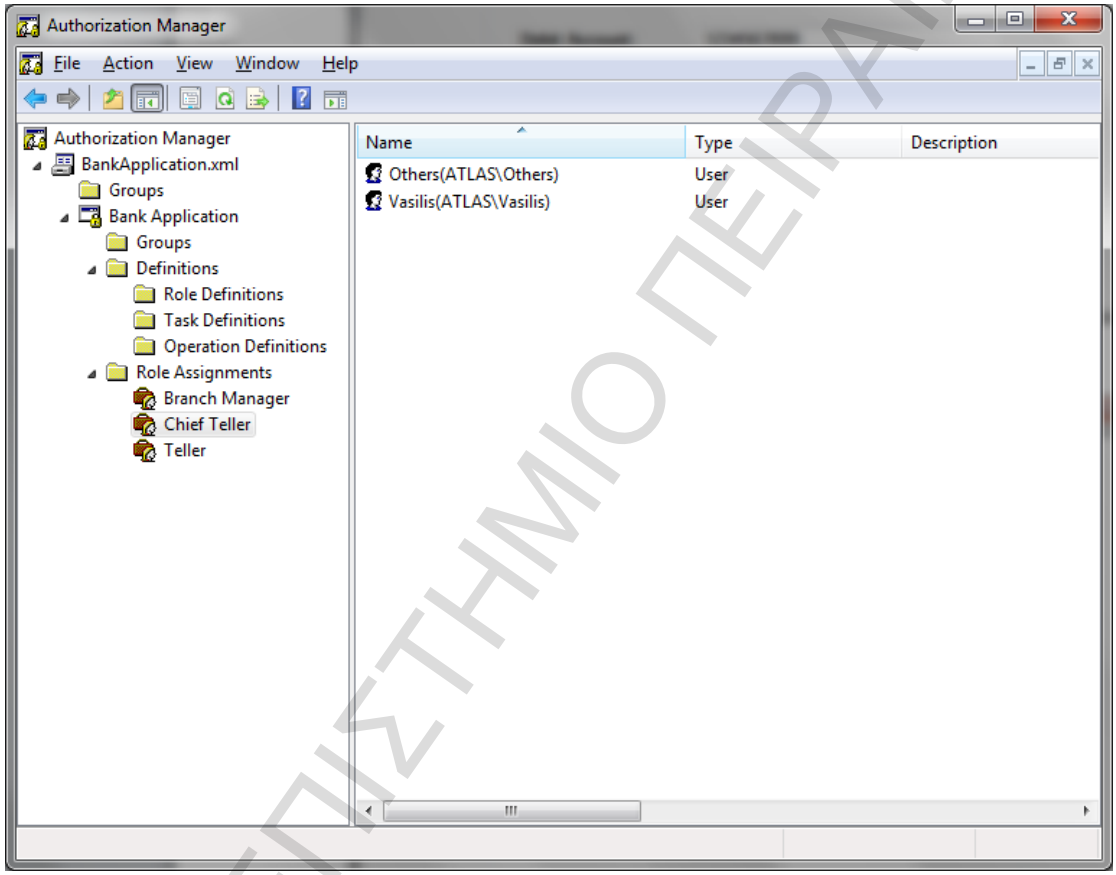
Εικόνα 40 Στιγμιότυπο εκτέλεσης Operation 2



Εικόνα 41 Στιγμιότυπο εκτέλεσης Operation 5 [1/2]

Εικόνα 42 Στιγμιότυπο εκτέλεσης Operation 5 [2/2]

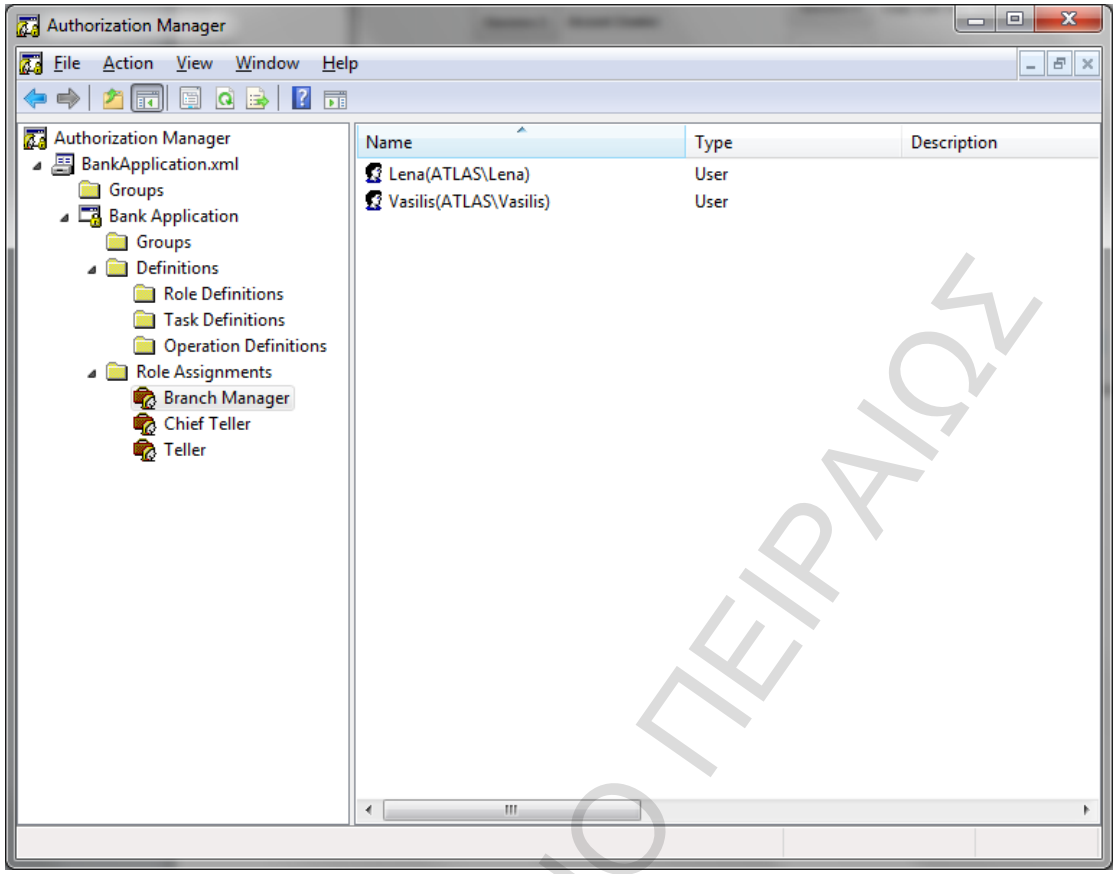
Στη συνέχεια παρουσιάζονται τα βήματα που απαιτούνται ώστε να διαγραφεί ένας χρήστης (π.χ. ο χρήστης Vasilis) από το ρόλο του Teller και να του ανατεθεί ο ρόλος του Chief Teller. Αφού κάνουμε refresh το Authorization Store Bank Application και ξανατρέξουμε την εφαρμογή, ο χρήστης αποκτά τα δικαιώματα προσβάσεων του Chief Teller όπως αυτά αποτυπώθηκαν στηω πολιτική που δημιουργήθηκε από το AzMan. Όπως φαίνεται παρακάτω (Εικόνες 43 και 44) και χωρίς καμία αλλαγή στον κώδικα της εφαρμογής, φορτώνεται η Πολιτική Ελέγχου Πρόσβασης σύμφωνα με τα δικαιώματα του νέου ρόλου, στον οποίο ανήκει ο χρήστης.



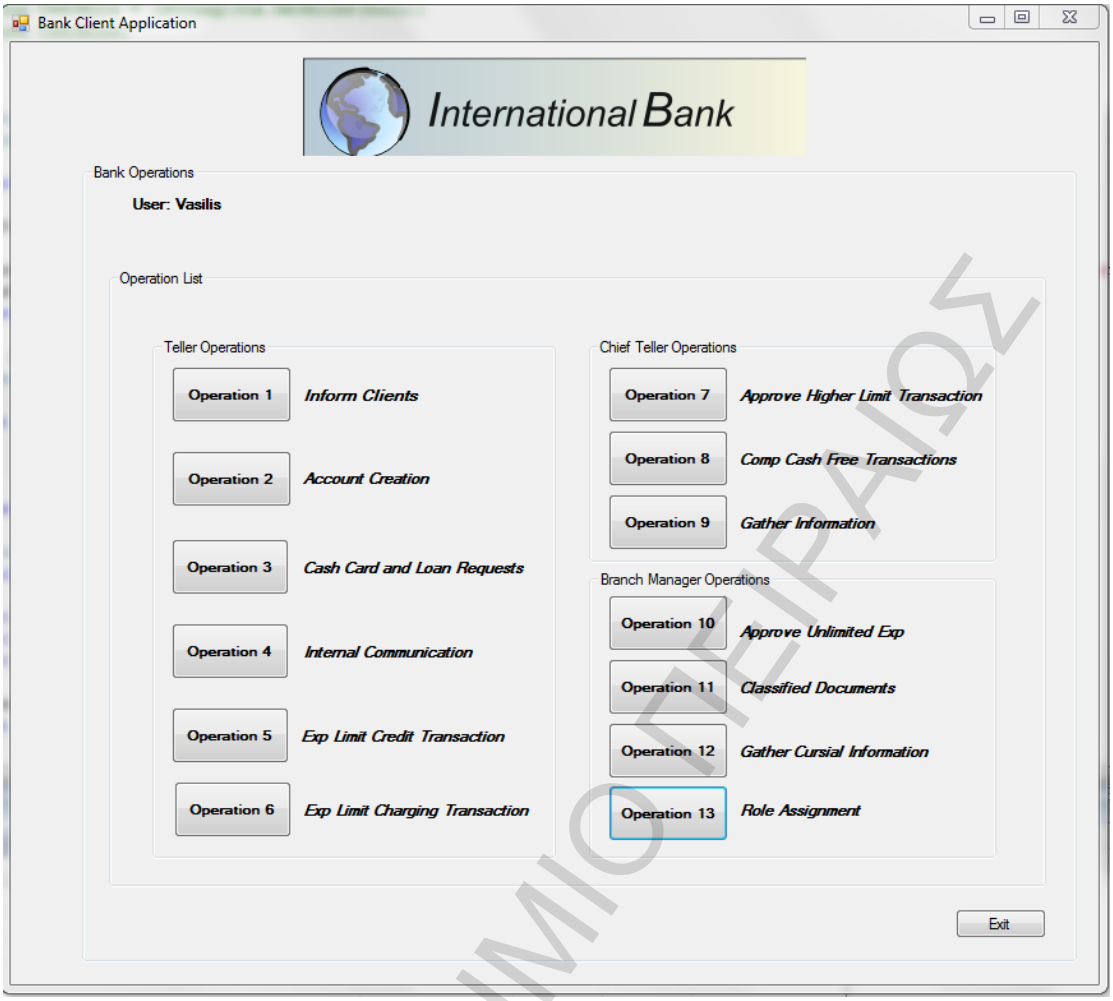
Εικόνα 43 Εκχώρηση στο χρήστη Vasilis του ρόλου Chief Teller

Εικόνα 44 Προσβάσεις εφαρμογής ρόλου Chief Teller

Ολοκληρώνοντας την παρουσίαση της εφαρμογής, εντάσσουμε το χρήστη στο ρόλο Bank Manager. Η διαδικασία επαναλαμβάνεται όπως ακριβώς περιγράφηκε στο προηγούμενο βήμα και τα αποτελέσματα φαίνονται στις εικόνες 45 και 46 που ακολουθούν.



Εικόνα 45 Εκχώρηση στο χρήστη Vasilis του ρόλου Branch Manager



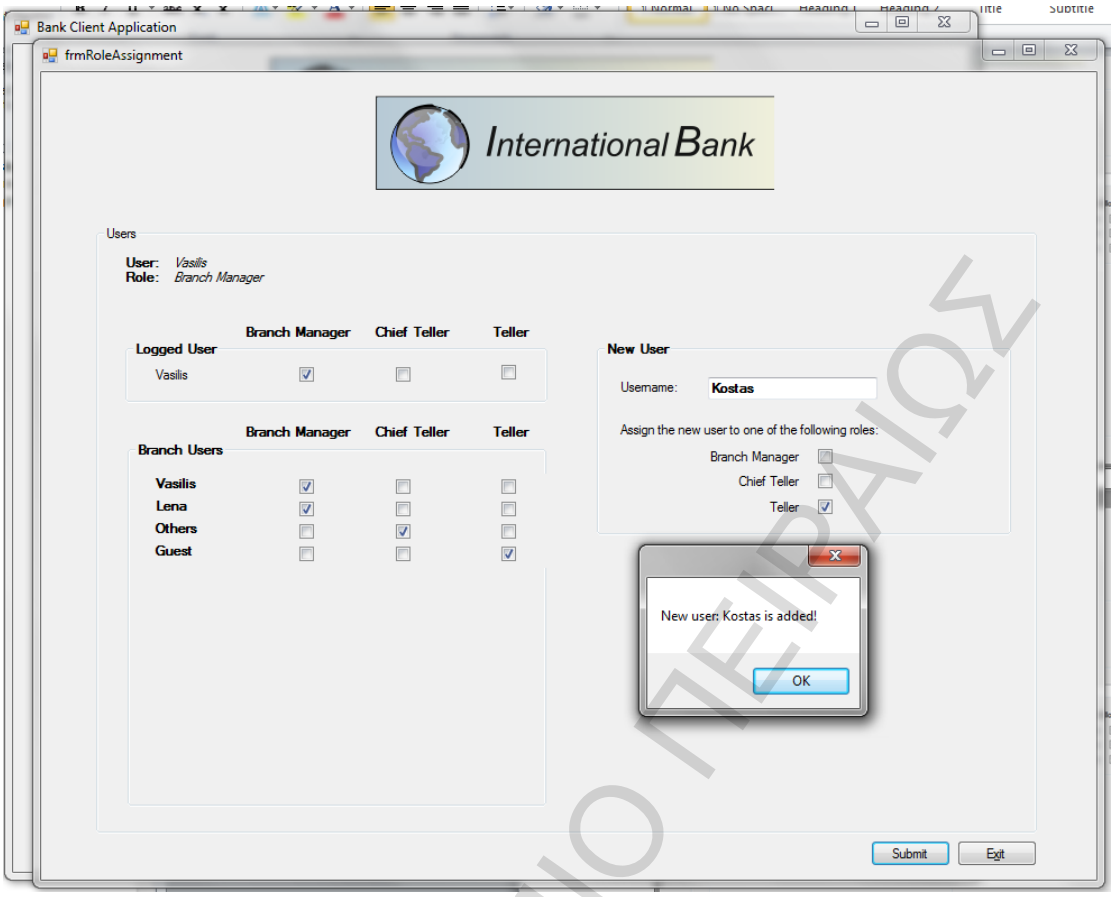
Εικόνα 46 Προσβάσεις εφαρμογής ρόλου Branch Manager

Ο χρήστης Vasilis, μέλος του ρόλου Branch Manager έχει πρόσβαση σε όλα τα operations της εφαρμογής. Σαν Branch Manager μπορεί να εκτελέσει τη λειτουργία 13 που είναι το Role Assignment. Η λειτουργία αποτυπώνεται στην εικόνα που ακολουθεί.

Εικόνα 47 Στιγμιότυπο εκτέλεσης Operation 13 [1/]

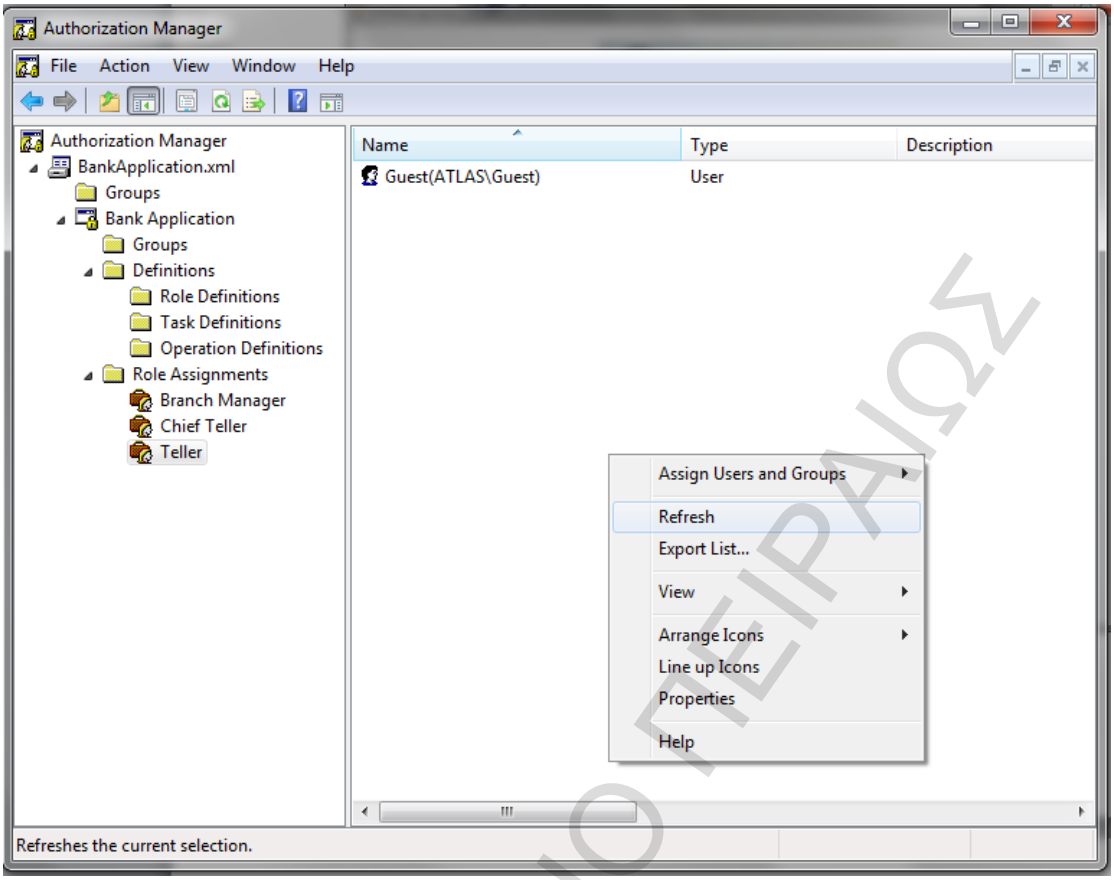
Τα πεδία που υπάρχουν στην λειτουργία 13 της εφαρμογής είναι 4. Το πρώτο δείχνει ποιος χρήστης έχει συνδεθεί, χρησιμοποιεί την εφαρμογή και ποιος ρόλος του αντιστοιχεί. Στο δεύτερο πεδίο φαίνονται όλοι οι χρήστες του συστήματος και οι ρόλοι που έχουν. Ο Branch Manager είναι ο μόνος, ο οποίος μπορεί να επεξεργαστεί και να αλλάξει τους ρόλους. Η αλλαγή ρόλου και κατά συνέπεια των προσβάσεων, ενεργοποιείται για το χρήστη στην επόμενη είσοδό του στο σύστημα. Το τρίτο τμήμα βρίσκεται στο δεξί μέρος της εφαρμογής (εικόνας 47) όπου φαίνονται τα πεδία τα οποία χρησιμοποιεί ο διαχειριστής για να εντάξει ένα νέο χρήστη στο σύστημα και να του αποδώσει έναν από τους τρεις ρόλους. Ακριβώς από κάτω, στο τέταρτο πεδίο, επιλέγει το ρόλο που θα έχει. Στο παράδειγμα μας, θα δημιουργηθεί ο χρήστης Kostas και θα του εκχωρηθεί ο ρόλος του Teller. Όπως θα δείξουμε στις παρακάτω εικόνες (48 έως 53) ο χρήστης δημιουργείται επιτυχώς, αποκτά προσβάσεις Teller, εντάσσεται στο σύστημα και στην πολιτική ασφάλειας που τρέχει στο AzMan .

Εικόνα 48 Στιγμιότυπο εκτέλεσης Operation 13 [1/]

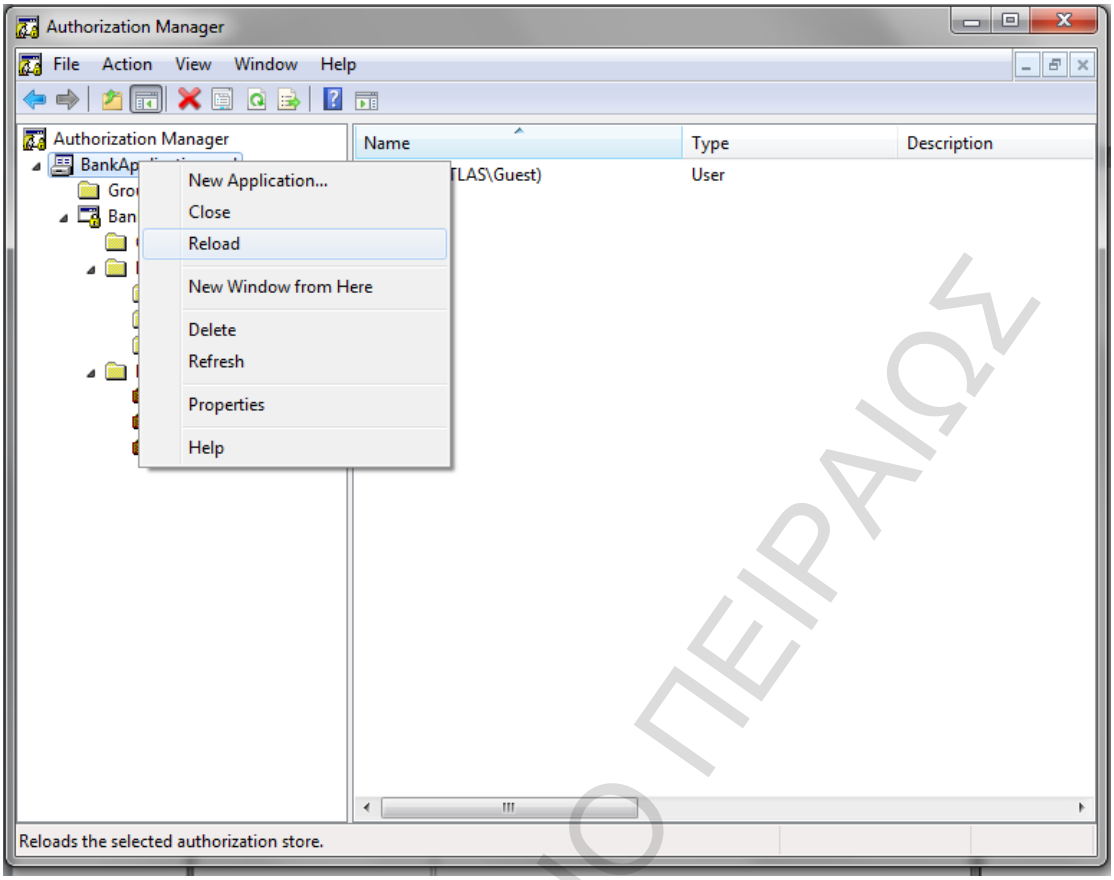


Εικόνα 49 Στιγμιότυπο εκτέλεσης Operation 13 [1/]

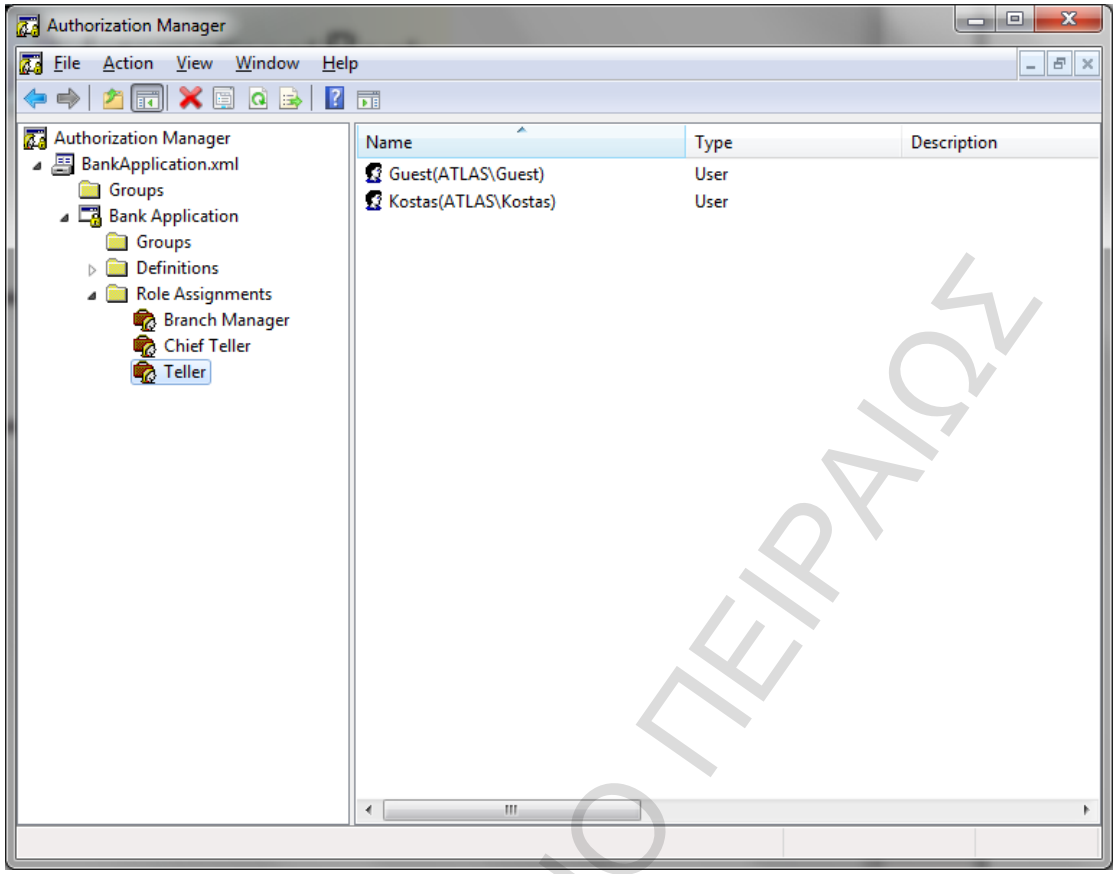
Εικόνα 50 Στιγμιότυπο εκτέλεσης Operation 13 [1/]



Εικόνα 51 Στιγμιότυπο της κονσόλας του AzMan πριν την ανανέωση[1/2]



Εικόνα 52 Στιγμιότυπο της κονσόλας του AzMan πριν την ανανέωση[2/2]



Εικόνα 53 Στιγμιότυπο της κονσόλας του AzMan μετά την ανανέωση

Η υλοποίηση του Operation 13,όπως δείχνουν και οι παραπάνω εικόνες, σημαίνει ότι η εφαρμογή μπορεί να λειτουργήσει τοπικά σε ένα κατάστημα περιορίζοντας το χρόνο που απαιτείται για την ολοκλήρωση της απονομής προσβάσεων σε χρήστες. Οι διαχειριστές του συστήματος, χωρίς να έρχονται σε άμεση επαφή με το εργαλείο που δημιουργεί τη πολιτική (Authorization Manager), μπορούν αναθέτουν στους υπαλλήλους ρόλους, χωρίς να επηρεάζουν την Πολιτική Ελέγχου Πρόσβασης. Επί της ουσίας απλά την υλοποιούν.

Οι παραπάνω εικόνες πιστοποιούν ότι έχει υλοποιηθεί και λειτουργεί ένα ρολο-κεντρικό μοντέλο ελέγχου πρόσβασης, βάσει της πολιτικής ασφάλειας που σχεδιάστηκε.

4.10 Συμμόρφωση με τις Βέλτιστες Πρακτικές

Όταν μια επιχείρηση αποφασίζει να στραφεί στο RBAC, πρέπει να είναι προετοιμασμένη για την υλοποίηση ενός δύσκολου έργου. Αυτό έχει καταστεί σαφές από όλη την επισκόπηση της βιβλιογραφίας. Λόγω της πολυπλοκότητας στο σχεδιασμό και την υλοποίηση ενός τέτοιου έργου, αναζητήθηκαν και παρουσιάστηκαν οι 10 βέλτιστες πρακτικές για την όσο το δυνατόν ομαλότερη εξέλιξη του έργου. Στον πίνακα που ακολουθεί, καταγράφεται το κατά πόσο η λύση που σχεδιάστηκε και υλοποιήθηκε, ικανοποιεί, έστω και σε πειραματικό επίπεδο, τις πρακτικές όπως αυτές αποτυπώθηκαν στην παράγραφο 3.5.

Βέλτιστες Πρακτικές	**Αποτέλεσμα**	**Αιτιολογία**
1. Καλώς Ορισμένοι Στόχοι	✓	Οι στόχοι ήταν καλώς ορισμένοι, καθώς αποτελούνταν από τη σχεδίαση και υλοποίηση μιας πολιτικής ελέγχου πρόσβασης σύμφωνα με τις βασικές αρχές του RBAC.
2.Διενέργεια Ανάλυσης Απαιτήσεων/Role Engineering	✓	Πραγματοποιήθηκε επιτυχώς ανάλυση απαιτήσεων, ώστε να δημιουργηθούν οι κατάλληλοι ρόλοι και να μπορεί να διασφαλιστεί ομαλή πρόσβαση των ρόλων στους πόρους.
3. Καθιέρωση Διαχείρισης του Κύκλου Ζωής των Ρόλων	×	Δεν μπορεί να ικανοποιηθεί σε πειραματικό στάδιο.
4. Ορισμός Εφικτού Χρονοδιαγράμματος	×	Δεν μπορεί να ικανοποιηθεί σε πειραματικό στάδιο.
5. Προσδιορισμός Ρόλων και Αρμοδιοτήτων	✓	Υπήρξε σαφής προσδιορισμός τόσο των ρόλων όσο και των αρμοδιοτήτων αυτών.
6. Κατάλληλη επιλογή συνεργατών	×	Δεν μπορεί να ικανοποιηθεί σε πειραματικό στάδιο.
7. Σχεδιασμός Υλοποίησης	✓	Υπήρξε αναλυτικός σχεδιασμός για την πορεία υλοποίησης και το τελικό αποτέλεσμα είναι πλήρες.
8. Ύπαρξη Κέντρου Υποστήριξης και	×	Δεν μπορεί να ικανοποιηθεί

Συνεργασίας των Εργαζόμενων Ομάδων		σε πειραματικό στάδιο.
9. Τήρηση του Χρονοδιαγράμματος	×	Δεν υφίσταται χρονοδιάγραμμα σε πειραματικό περιβάλλον στο οποίο εκπονήθηκε η παρούσα εργασία.
10. Κατάλληλη προετοιμασία, ώστε να υπάρχει η δυνατότητα υλοποίησης μη αναμενόμενων αλλαγών.	×	Η προετοιμασία, για την εκπόνηση της εργασίας, έγκειται στη μελέτη της βιβλιογραφίας, στο σχεδιασμό της, στην εκπόνηση της απαραίτητης εφαρμογής και τέλος στη συγγραφή του τεύχους παρουσίασής της. Υπ' αυτή την έννοια υλοποιήθηκαν όσες αλλαγές χρειάστηκε, στην πορεία, να γίνουν

Πίνακας 3 Συμμόρφωση Πειραματικής Διαδικασίας με τις Βέλτιστες Πρακτικές

4.11 Συμπεράσματα

Κατά την πειραματική διαδικασία τηρήθηκαν όλα τα βήματα που προτείνουν οι διεθνείς βέλτιστες πρακτικές, ώστε να σχεδιαστεί και να υλοποιηθεί μια RBAC πολιτική ασφάλειας. Η ανάλυση απαιτήσεων και το Role Engineering παρά το ότι είναι μια δύσκολη και χρονοβόρα διαδικασία εκτιμάται ότι προσεγγίστηκε με επιτυχία . Κατά την εκπόνηση της παρούσας εργασίας, έγιναν παραδοχές και απλουστεύσεις μιας και **ο κύριος στόχος ήταν η υλοποίηση μιας ρόλο-κεντρικής πολιτικής σε πειραματικό περιβάλλον**. Αυτό προέκυψε με την εκπόνηση ενός απλουστευμένου role engineering, το σχεδιασμό της πολιτικής μέσω του AzMan και τέλος της υλοποίησης της πολιτικής που σχεδιάστηκε και εφαρμόστηκε. Οι χρήστες λοιπόν με την εισαγωγή τους στο σύστημα, ανοίγουν μια κοινή για όλους εφαρμογή, ό έλεγχος πρόσβασης πραγματοποιείται σύμφωνα με τη συνδεμένη πολιτική που υπάρχει στο AzStore. Συνοψίζοντας τονίζονται τα παρακάτω Συμπεράσματα:

Από τη λειτουργία της υλοποιησθείσας εφαρμογής προκύπτει ότι εξασφαλίζεται:

- ✓ Η Διενέργεια πλήρους ελέγχου πρόσβασης μέσω της RBAC πολιτικής ασφάλειας του Οργανισμού.
- ✓ Η Εύκολη Παραμετροποίηση.
- ✓ Ο Συνδυασμός Εργαλείων.
- ✓ Το «κλείδωμα» των δομικών στοιχείων της πολιτικής, ώστε να είναι αδύνατη οποιαδήποτε αλλαγή, από τους τοπικούς διαχειριστές.
- ✓ Η Συμμόρφωση με το Core RBAC.
- ✓ Η Συμμόρφωση με το Ιεραρχικό RBAC.
- ✓ Η Ευκολία μετατροπής της από πειραματική σε πραγματική.
- ✓ Η Αξιοποίησή της, ως παράδειγμα, για υλοποίηση RBAC πολιτικών σε άλλους οργανισμούς.

5 Επίλογος

5.1 Μελλοντικές κατευθύνσεις

Η εφαρμογή που δημιουργήθηκε αποδεικνύει τη δυνατότητα αξιοποίησης του εργαλείου AzMan για την υλοποίηση εφαρμογών ρόλο-κεντρικών πολιτικών. Οι συγκεκριμένες λοιπόν πολιτικές μπορεί υλοποιηθούν σε διάφορες εφαρμογές που απαιτούν Έλεγχο Πρόσβασης σε περιβάλλον Windows.

Η πειραματική εφαρμογή που παρουσιάστηκε διαφέρει από μία εφαρμογή παραγωγής σε ένα πραγματικό περιβάλλον υποκαταστήματος τράπεζας. Στην εφαρμογή αυτή αποτυπώθηκε το ελάχιστο εύρος των εργασιών και της πολυπλοκότητας, παρόλα αυτά, μπορεί εύκολα να αποκτήσει πλήρη λειτουργικότητα σε πραγματικό περιβάλλον εργασίας εφαρμόζοντας τις συγκεκριμένες αρχές.

Η προτεινόμενη εφαρμογή μπορεί, με μικρές αλλαγές, να πραγματοποιεί έλεγχο πρόσβασης σε δικτυακό περιβάλλον, να στηθεί δηλαδή σε server και να λειτουργήσει αποτελεσματικά.

Πεδίο εφαρμογής μπορεί να είναι κάθε είδους οργανισμός που επιδιώκει να επιτύχει Έλεγχο Πρόσβασης και λειτουργεί σε δικτυακό περιβάλλον Windows. Οι εφαρμογές αυτού του είδους μπορεί να συνδεθούν με βάσεις δεδομένων και να γίνουν ακόμα περισσότερο λειτουργικές.

Το Authorization Manager είναι ένα χρήσιμο εργαλείο που προσφέρεται από τα Windows. Το εργαλείο αυτό αν και ιδιαίτερα χρήσιμο δεν μπορεί να χαρακτηριστεί πλήρες καθώς υπάρχουν, περιθώρια βελτίωσης τόσο χρηστικής όσο και μηχανογραφικής.

- ✓ **Ο διαχωρισμός καθηκόντων**: Μπορεί να υλοποιηθεί **μόνο λογικά** με βάση τον τρόπο που στήνεται η πολιτική και **όχι αυτοματοποιημένα**.
- ✓ **Δεν υπάρχει τρόπος να οριστούν αμοιβαίως αποκλειόμενοι ρόλοι**, πράγμα το οποίο κρίνεται ιδιαίτερα χρήσιμο.

Με βάση αυτές τις ελλείψεις, κρίνεται σκόπιμο να προταθεί η εξέλιξη του υπάρχοντος εργαλείου ή δημιουργία ενός νέου, που θα διατηρεί τις καινοτομίες και τα πλεονεκτήματα του AzMan απαλείφοντας όμως τόσο τις προαναφερθείσες ελλείψεις, όσο και αυτές που ενδεχομένως θα εντοπιστούν μελλοντικά.

5.2 Συμπεράσματα

Η τελευταία παράγραφος και επί της ουσίας επίλογος της παρούσας ΜΔΕ, αφορά την καταγραφή των συμπερασμάτων που εξήχθησαν καθ' όλη την εκπόνηση της μελέτης.

Η ενότητα του Ελέγχου Πρόσβασης παραμένει ένα ανοιχτό ζήτημα προς περεταίρω διερεύνηση. Ο υπογράφων την εργασία θεωρεί ότι, πολύ δύσκολα θα βρεθεί μια κοινά αποδεχτή λύση από τους ερευνητές και θα σταματήσει η ερεύνα γύρω από το συγκεκριμένο τομέα.

Αντιθέτως, αναμένεται ραγδαία αύξηση του ενδιαφέροντος καθώς τόσο η τεχνολογική εξέλιξη όσο και οι απαιτήσεις του περιβάλλοντος των πολύπλοκων οργανισμών συνεχώς επιζητούν:

- ✓ Περισσότερη Ασφάλεια
- ✓ Μεγαλύτερη Ευελιξία
- ✓ Μικρότερο Διαχειριστικό Κόστος
- ✓ Περισσότερη Αποτελεσματικότητα

Λόγω των παραπάνω απαιτήσεων η έρευνα στο πεδίο του Ελέγχου Πρόσβασης εντείνεται με συνέπεια να προτείνονται:

- ✓ Νεότερες Λύσεις
- ✓ Επεκτάσεις και συμπληρώσεις των υφιστάμενων μοντέλων και πολιτικών
- ✓ Εξολοκλήρου νέα μοντέλα που έχουν στόχο να προτείνουν νέους τρόπους αντιμετώπισης του ζητήματος του ελέγχου πρόσβασης
    - ➢ Με συνέπεια να φιλοδοξούν να ρίξουν από την κορυφή το RBAC

Το RBAC, ακόμα και σήμερα, 21 χρόνια μετά την πρώτη του επίσημη παρουσίαση θεωρείται ως το πληρέστερο μοντέλο ελέγχου πρόσβασης. Σήμερα όλο και περισσότεροι κατασκευαστές δημιουργούν νέα λογισμικά τα οποία βασίζονται σε αυτό. Οι πολύπλοκοι οργανισμοί το προτιμούν, αλλά για την εξυπηρέτηση των ιδιαίτερων χαρακτηριστικών του κάθε οργανισμού απαιτούνται συνεχείς επεκτάσεις και βελτιώσεις στη λειτουργία του.

Ειδικότερα σημειώνεται ότι με βάση το RBAC και τη χρήση σύγχρονων εργαλείων μπορεί εύκολα:

a) Να λυθούν προβλήματα

b) Να δημιουργηθούν πολιτικές

c) Να στηθούν εφαρμογές που θα υλοποιούν τον έλεγχο πρόσβασης

Τελικά αυτό που μπορούμε με σιγουριά να συμπεράνουμε είναι ότι και το RBAC, όπως και όλη η θεματική ενότητα του ελέγχου πρόσβασης είναι ένα ανοιχτό θέμα για το οποίο:

- ✓ Πραγματοποιείται συνεχώς έρευνα
- ✓ Διατυπώνονται νέες προτάσεις
- ✓ Σχεδιάζονται νέες επεκτάσεις
- ✓ Αναμένονται νέες τάσεις

Το βήμα από το οποίο συνήθως παρουσιάζονται οι βελτιώσεις, οι νέες ιδέες και οι εξελίξεις που αφορούν το συγκεκριμένο αντικείμενο παρουσιάζονται, στο ετήσιο Συμπόσιο των Μηχανισμών Ελέγχου Πρόσβασης.

Το RBAC μπορεί να περιγραφεί ως ένα «οργανισμός» που ζει και εξελίσσεται, ένα εργαλείο που χρησιμοποιείται ευρέως. Η υλοποίηση RBAC πολιτικών **προτείνεται** να ακολουθούν τις κατευθυντήριες γραμμές του προτύπου του ANSI/INCITS 359-2004 και να ενσωματώνουν τις απαραίτητες επεκτάσεις και προσαρμογές ανάλογα με τις ανάγκες που κάθε φορά υπάρχουν.

Το πρότυπο ANSI/INCITS 359-2004 πρέπει, κατά την άποψη του υπογράφοντος την παρούσα ΜΔΕ, να ακολουθείται πιστά κατά το σχεδιασμό και την υλοποίηση μιας RBAC λύσης. Εάν το πρότυπο δεν τηρηθεί συνεπάγεται:

- ✓ Αύξηση της πολυπλοκότητας
- ✓ Μείωση της λειτουργικότητας, χωρίς ανάλογη αύξηση του επιπέδου ασφάλειας του συστήματος

Τελικά προτείνεται ότι, κατά το σχεδιασμό μιας πολιτικής ελέγχου πρόσβασης πρέπει να εξασφαλίζεται η μεγαλύτερη δυνατή ισορροπία μεταξύ:

- ✓ Του επιπέδου της Ασφάλειας
- ✓ Του κόστους υλοποίησης
- ✓ Της ευελιξίας
- ✓ Της λειτουργικότητας

Βιβλιογραφία

ANSI., "*American National Standard for Information Technology – Role Based Access Control*", ANSI

Bell D. E., "Looking back at the Bell-La Padula model", *Computer Security Applications Conference, 21st Annual*, pp. 15-27, 5-9 December 2005

Bell D. E., LaPadula L., "*Secure Computer Systems: Mathematical Foundations*", MITRE Technical Report 2547, 1973

Bertino E., Bonatti A., Ferrari E., "TRBAC: A Temporal Role-Based Access Control Model", *ACM Transactions on Information and System Security*, Vol.4, No. 3, pp. 191 - 223September 2001

Biba, K., "*Integrity Considerations for Secure Computer Systems*", MTR-3153, The Mitre Corporation, April 1977

Bishop M., "*Computer Security: Art and Science*", MA: Addison-Wesley, Boston 2003.

Brewer D., Nash M., "The Chinese Wall Security Policy", *Proceedings of IEEE Symposium on Security and Privacy*, pp. 206-214. IEEE, 1989

Collberg C., "*Computer Security, 2: Introduction — Mechanisms*", University of Arizona, 2012

Covington M., Moyer M., Ahamad M., "Generalized Role-Based Access Control for Securing Future Applications", *College of Computing, Georgia Institute of Technology*, Atlanta, Georgia, 2000

Denninng P., "Third Generation Computer Systems", *ACM Computing Surveys*, Vol.3, No.4, pp. 175-216, 1971

Fernandez R., "*Enterprise Dynamic Access Control (EDAC) Compliance with the American National Standards Institute (ANSI) Role Based Access Control (RBAC)*" May 1, 2005

Ferraiolo D., Kuhn R., "Role-Based Access Controls," *Proceedings of the 15th National Computer Security Conference*, pp. 554-563, Baltimore, October, 1992

Ferraiolo D., Janet A., Cugini, D., Kuhn R., "Role-Based Access Control (RBAC): Features and Motivations*" 11th Annual Computer Security Applications Proceedings*, (1995)

Ferraiolo D., Sandhu R., Gavrila S., Kuhn R, Chandramouli R., "Proposed NIST standard for role-based access control", *ACM Transactions on Information and Systems Security*, 4(3):pp. 224–274, August, 2001.

Follette D., "*Demystified Series: Getting Started with AzMan", http://channel9.msdn.com*, [Online] March 07, 2007 [Accessed: 10 February 2013], http://channel9.msdn.com/Blogs/donovanf/Demystified-Series-Getting-Started-with-AzMan

Graham G., Denning P., "Protection – Principles and Practice", *Spring Joint Computer Conference Proceedings*, Col.40, pp. 417-429, 1972

Hansche S., Berti J., Hare J., "*Official (ISC)2 Guide to the CISSP Exam*", Auerbach Publications, New York, 2003

Indrakshi R., Mahendra K., Lijun Y., "LRBAC: A Location-Aware Role-Based Access Control Model", *Proceedings of: Information Systems Security, Second International Conference*, pp.147-161, Kolkata, India, December 19-21, 2006

Kulkarni D., Tripathi A., "Context-aware role-based access control in pervasive computing systems", *SACMAT '08 Proceedings of the 13th ACM symposium on Access control models and technologies*, pp. 113-122, USA, New York, 2008

Lampson, B. W., "Protection", *Proceedings of the 5th Princeton Conference on Information Sciences and Systems*, pp. 437, 1971

Mao Z., Li N., Chen H., Jiang X., "*Trojan Horse Resistant DAC*", Proceedings of the 14th ACM symposium on Access Control Models and Technologies, 2009

Meunier P., *"Confusion of Separation of Privilege and Least Privilege", http://www.cerias.purdue.edu/site/blog*, [Online] January 15, 2008 [Accessed: 10 February 2013], http://www.cerias.purdue.edu/site/blog/post/confusion-of-separation-of-privilege-and-least-privilege/

Moore J., "Access Control", *Information Security Operations, Information Systems Security Association*, Atlanta 2001.

Northcutt S., "*Role Based Access Control to Achieve Defense in Depth", http://www.sans.edu*, [Online] December 3, 2012 [Accessed: 10 February 2013], http://www.sans.edu/research/security-laboratory/article/311

O'Connor A., Loomis P., "Economic Analysis of Role-Based Access Control", *Technical report, National Institute of Standards and Technology* (*NIST*), 2010.

Saltzer J., Schroeder M., "The protection of information in computer systems", *Proceedings of the IEEE*, Vol. 63, No. 9, September 1975, p.1278-1308

Samarati P., "*Access Control*", Università degli Studi di Milano, FOSAD, Milan 2008

Sandhu R., Samarati P., "Access control: principle and practice", *Communications Magazine IEEE*, Vol. 32, Issue 9, pp 40-48, 1994

Sandhu R., Coyne E., Feinstein H., Youman C., *"*Role-based Access Control Models*"*, *IEEE Computer*, Vol. 29, No. 2, pp38-47, February 1996

Simeio Solutions, *"Top 10 Steps to a Successful Enterprise Role Management Deployment", http://www.simeiosolutions.com*, [Online] June 16, 2009 [Accessed: 10 February 2013], http://www.simeiosolutions.com/images/10%20Steps%20to%20a%20Successful%20Enterprise%20Role%20Management%20Deployment.pdf

Simon R., Zukro E., "Separation of Duty in Role-Based Environments", *In Proceedings*

of IEEE Computer Security Foundations Workshop, pp 183 -194, Rockport, December, 1997

University of California Davis, "*Design Principles*", *http://www.cs.ucdavis.edu/*, [Accessed: 10 February 2013], http://cancer.cs.ucdavis.edu/course/ecs235a/f12/scribe/1001.pdf

Young B., "Foundations of Computer Security Lecture 25: The Chinese wall Policy", *University of Texas at Austin, Department of Computer Sciences*, Austin, 2012

Κάτσικας Σ., Γκρίτζαλης Δ., Γκρίτζαλης Σ., «*Ασφάλεια Πληροφοριακών Συστημάτων*», Νέων Τεχνολογιών, Αθήνα 2004

Λαμπρινουδάκης Κ., «*Μηχανισμοί Ελέγχου Προσπέλασης*», Πανεπιστήμιο Πειραιά, Σημειώσεις Μαθήματος: Ασφάλεια, Ιδιωτικότητα και Εμπιστοσύνη Δικτυοκεντρικών Πληροφοριακών Συστημάτων, Ακαδημαϊκό Έτος 2011-2012.

Μάγκος Ε., «*Ασφάλεια Υπολογιστών και Προστασία Δεδομένων*», Ιόνιο Πανεπιστήμιο, Κέρκυρα, Ακαδημαϊκό Έτος 2010-2011.

Πάγκαλος Γ., Μαυρίδης Ι., «*Ασφάλεια Πληροφοριακών Συστημάτων και Δικτύων*», Ανικούλα, Θεσσαλονίκη 2002

Παράρτημα

Στο παράρτημα παρατίθενται τμήματα του πηγαίου κώδικα της εφαρμογής που υλοποιήθηκε και παρουσιάστηκε στο κεφάλαιο 4.

Ο κώδικας του βασικού προγράμματος:

Program.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;
using AZROLESLib;
using System.Security.Principal;
using System.IO;


namespace BankApplication
{
    //Το κυρίως πρόγραμμα, Program, του BankApplication. Στο σημείο
αυτό
    //υλοποιείται η αρχικοποίηση του Authorization Store και οι
συναρτήσεις
    //AccessCheck και MultipleAccessCheck για τους απαιτούμενους
ελέγχους
    //στον Authorization Manager.

    static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            if (AzManInitialization())
            {
                Application.Run(new frmMain());

            }
        }

        public static string LoggedUser() {
            string UserName = WindowsIdentity.GetCurrent().Name;
            return UserName.Substring(UserName.IndexOf('\\') + 1);
        }


        static AzAuthorizationStore AzmanStore;
        static IAzApplication AzManApp;
```

```
        //Συνάρτηση αρχικοποίησης του Authorization Manager

        public static bool AzManInitialization()
        {
            AzmanStore = new AzAuthorizationStore();
            try {
                //Path
                AzmanStore.Initialize(0,
@"msxml://C:\Users\Vasilis\Documents\Visual                    Studio
2008\Projects\BankApplication\BankApplication\BankApplication.xml",
null);
                AzManApp      =      AzmanStore.OpenApplication("Bank
Application", null);
                return true;
            }
            catch (FileNotFoundException) {
                MessageBox.Show("Unable  to  find  the  xml  path  and
initialize AzMan.");
            }
            return false;
        }

        //Συνάρτηση AccessCheck που ελέγχει μέσω του χρήστη που έχει
κάνει log in την πρόσβαση
        //σε κάποιο συγκεκριμένο Operation
        public static bool AccessCheck(string audit, Operations op)
        {
            //Αρχικοποίηση  του  Client  Context  μέσω  του  χρήστη  που
έχει κάνει log in στο σύστημα
            IAzClientContext                    ctx                =
AzManApp.InitializeClientContextFromToken(
                (ulong)WindowsIdentity.GetCurrent().Token.ToInt64(),
null);

            object[] scopes = { "" };
            object[] operations = { (int)op };
            object[]   results   =   (object[])ctx.AccessCheck(audit,
scopes, operations,
                                                          null,  null,
null, null, null);
            int result = (int)results[0];
            return 0 == result;

        }

        //Συνάρτηση  MultipleAccessCheck  πουμέσω  της  κλήσης  της
AccessCheck ελέγχει την πρόσβαση
        //του χρήστη σε όλα τα Operations του Authorization Store
        public  static  bool[]  MultipleAccessCheck(string  audit,
Operations[] ops)
        {
            IAzClientContext                    ctx                =
AzManApp.InitializeClientContextFromToken(
                (ulong)WindowsIdentity.GetCurrent().Token.ToInt64(),
null);


            object[] scopes = new object[ops.Length];
            object[] operations = new object[ops.Length];
            bool[] results = new bool[ops.Length];
            for (int i = 0; i < ops.Length; ++i)
```

```
            {
                scopes[i] = "";
                operations[i] = (int)ops[i];
            }
            object[]  rawResults  =  (object[])ctx.AccessCheck(audit,
scopes, operations,
                                                                null,
null, null, null, null);
            for (int i = 0; i < results.Length; ++i)
            {
                results[i] = (0 == (int)rawResults[i]);
            }
            return results;
        }


        public static void RefreshLocalAuthzCache()
        {
           AzmanStore.UpdateCache(null);
        }
    }
}
```

Operations.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace BankApplication
{
    //Αντιστοίχηση του κάθε Operation με το αντίστοιχο Operation
Number.

    public enum Operations{
        Inform_Clients = 1,
        Account_Creation = 2,
        C_Cards_n_Loan_Requests = 3,
        Internal_Communication = 4,
        Exp_Limit_Credit_Trans = 5,
        Exp_Limit_Charging_Trans = 6,
        Approve_Higher_Limit_Trans = 7,
        Comp_Cash_Free_Trans = 8,
        Gather_Info = 9,
        Unlimited_Approve_Exp = 10,
        Classified_Docs = 11,
        Gather_Cursial_Info = 12,
        Role_Assignment = 13,
    }
}
```

frmMain

[Design]:

frmMain.Designer.cs:

```
namespace BankApplication
{
    partial class frmMain
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should
be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
```

```
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }



        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.btnExit = new System.Windows.Forms.Button();
            this.lblMessage2 = new System.Windows.Forms.Label();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.groupBox5 = new System.Windows.Forms.GroupBox();
            this.lblGatherCursialInfo = new
System.Windows.Forms.Label();
            this.lblClassifiedDocs = new
System.Windows.Forms.Label();
            this.lblUnlimitedApproveExp = new
System.Windows.Forms.Label();
            this.btnGatherCursialInfo = new
System.Windows.Forms.Button();
            this.btnClassifiedDocs = new
System.Windows.Forms.Button();
            this.btnUnlimitedApproveExp = new
System.Windows.Forms.Button();
            this.btnRoleAssignment = new
System.Windows.Forms.Button();
            this.lblRoleAssignment = new
System.Windows.Forms.Label();
            this.groupBox4 = new System.Windows.Forms.GroupBox();
            this.lblGatherInfo = new System.Windows.Forms.Label();
            this.lblCompCashFreeTrans = new
System.Windows.Forms.Label();
            this.btnGatherInfo = new System.Windows.Forms.Button();
            this.btnCompCashFreeTrans = new
System.Windows.Forms.Button();
            this.btnApproveHigherLimitTrans = new
System.Windows.Forms.Button();
            this.label1 = new System.Windows.Forms.Label();
            this.groupBox3 = new System.Windows.Forms.GroupBox();
            this.lblExpLimitChargingTrans = new
System.Windows.Forms.Label();
            this.lblExpLimitCreditTrans = new
System.Windows.Forms.Label();
            this.lblInternalCommunication = new
System.Windows.Forms.Label();
            this.lblCCardNLoadRequests = new
System.Windows.Forms.Label();
            this.btnExpLimitCreditTans = new
System.Windows.Forms.Button();
```

```
            this.btnExpLimitChargingTrans = new
System.Windows.Forms.Button();
            this.btnInternalCommunication = new
System.Windows.Forms.Button();
            this.btnCCardNLoadRequests = new
System.Windows.Forms.Button();
            this.lblAccountCreation = new
System.Windows.Forms.Label();
            this.btnAccountCreation = new
System.Windows.Forms.Button();
            this.lblInformClients = new System.Windows.Forms.Label();
            this.btnInformClients = new
System.Windows.Forms.Button();
            this.lblMessage1 = new System.Windows.Forms.Label();
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.groupBox1.SuspendLayout();
            this.groupBox2.SuspendLayout();
            this.groupBox5.SuspendLayout();
            this.groupBox4.SuspendLayout();
            this.groupBox3.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginI
nit();
            this.SuspendLayout();
            //
            // groupBox1
            //
            this.groupBox1.Controls.Add(this.btnExit);
            this.groupBox1.Controls.Add(this.lblMessage2);
            this.groupBox1.Controls.Add(this.groupBox2);
            this.groupBox1.Controls.Add(this.lblMessage1);
            this.groupBox1.Location = new System.Drawing.Point(56,
95);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(779, 626);
            this.groupBox1.TabIndex = 0;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Bank Operations";
            this.groupBox1.Enter += new
System.EventHandler(this.groupBox1_Enter);
            //
            // btnExit
            //
            this.btnExit.DialogResult =
System.Windows.Forms.DialogResult.Cancel;
            this.btnExit.Location = new System.Drawing.Point(684,
584);
            this.btnExit.Name = "btnExit";
            this.btnExit.Size = new System.Drawing.Size(71, 23);
            this.btnExit.TabIndex = 2;
            this.btnExit.Text = "E&xit";
            this.btnExit.UseVisualStyleBackColor = true;
            this.btnExit.Click += new
System.EventHandler(this.btnExit_Click);
            //
            // lblMessage2
            //
            this.lblMessage2.AutoSize = true;
            this.lblMessage2.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
```

```
System.Drawing.FontStyle.Bold,      System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblMessage2.Location = new System.Drawing.Point(36,
47);
            this.lblMessage2.Name = "lblMessage2";
            this.lblMessage2.Size = new System.Drawing.Size(37, 13);
            this.lblMessage2.TabIndex = 4;
            this.lblMessage2.Text = "Msg2";
            this.lblMessage2.Visible = false;
            //
            // groupBox2
            //
            this.groupBox2.Controls.Add(this.groupBox5);
            this.groupBox2.Controls.Add(this.groupBox4);
            this.groupBox2.Controls.Add(this.groupBox3);
            this.groupBox2.Location = new System.Drawing.Point(21,
83);
            this.groupBox2.Name = "groupBox2";
            this.groupBox2.Size = new System.Drawing.Size(734, 484);
            this.groupBox2.TabIndex = 3;
            this.groupBox2.TabStop = false;
            this.groupBox2.Text = "Operation List";
            //
            // groupBox5
            //
            this.groupBox5.Controls.Add(this.lblGatherCursialInfo);
            this.groupBox5.Controls.Add(this.lblClassifiedDocs);
            this.groupBox5.Controls.Add(this.lblUnlimitedApproveExp);
            this.groupBox5.Controls.Add(this.btnGatherCursialInfo);
            this.groupBox5.Controls.Add(this.btnClassifiedDocs);
            this.groupBox5.Controls.Add(this.btnUnlimitedApproveExp);
            this.groupBox5.Controls.Add(this.btnRoleAssignment);
            this.groupBox5.Controls.Add(this.lblRoleAssignment);
            this.groupBox5.Location = new System.Drawing.Point(376,
236);
            this.groupBox5.Name = "groupBox5";
            this.groupBox5.Size = new System.Drawing.Size(319, 226);
            this.groupBox5.TabIndex = 15;
            this.groupBox5.TabStop = false;
            this.groupBox5.Text = "Branch Manager Operations";
            //
            // lblGatherCursialInfo
            //
            this.lblGatherCursialInfo.AutoSize = true;
            this.lblGatherCursialInfo.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblGatherCursialInfo.Location = new
System.Drawing.Point(115, 135);
            this.lblGatherCursialInfo.Name = "lblGatherCursialInfo";
            this.lblGatherCursialInfo.Size = new
System.Drawing.Size(154, 13);
            this.lblGatherCursialInfo.TabIndex = 16;
            this.lblGatherCursialInfo.Text = "Gather Cursial
Information";
            //
            // lblClassifiedDocs
            //
            this.lblClassifiedDocs.AutoSize = true;
```

```
            this.lblClassifiedDocs.Font             =             new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold          |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblClassifiedDocs.Location           =           new
System.Drawing.Point(115, 85);
            this.lblClassifiedDocs.Name = "lblClassifiedDocs";
            this.lblClassifiedDocs.Size             =             new
System.Drawing.Size(128, 13);
            this.lblClassifiedDocs.TabIndex = 15;
            this.lblClassifiedDocs.Text = "Classified Documents";
            //
            // lblUnlimitedApproveExp
            //
            this.lblUnlimitedApproveExp.AutoSize = true;
            this.lblUnlimitedApproveExp.Font          =          new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold          |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblUnlimitedApproveExp.Location        =        new
System.Drawing.Point(115, 41);
            this.lblUnlimitedApproveExp.Name                      =
"lblUnlimitedApproveExp";
            this.lblUnlimitedApproveExp.Size          =          new
System.Drawing.Size(135, 13);
            this.lblUnlimitedApproveExp.TabIndex = 14;
            this.lblUnlimitedApproveExp.Text  =  "Approve  Unlimited
Exp";
            //
            // btnGatherCursialInfo
            //
            this.btnGatherCursialInfo.Font            =            new
System.Drawing.Font("Microsoft       Sans       Serif",       7.5F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnGatherCursialInfo.Location          =          new
System.Drawing.Point(15, 119);
            this.btnGatherCursialInfo.Name = "btnGatherCursialInfo";
            this.btnGatherCursialInfo.Size            =            new
System.Drawing.Size(94, 44);
            this.btnGatherCursialInfo.TabIndex = 13;
            this.btnGatherCursialInfo.Text = "Operation 12";
            this.btnGatherCursialInfo.UseVisualStyleBackColor = true;
            //
            // btnClassifiedDocs
            //
            this.btnClassifiedDocs.Font             =             new
System.Drawing.Font("Microsoft       Sans       Serif",       7.5F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnClassifiedDocs.Location           =           new
System.Drawing.Point(15, 69);
            this.btnClassifiedDocs.Name = "btnClassifiedDocs";
            this.btnClassifiedDocs.Size = new System.Drawing.Size(94,
44);
            this.btnClassifiedDocs.TabIndex = 12;
            this.btnClassifiedDocs.Text = "Operation 11";
            this.btnClassifiedDocs.UseVisualStyleBackColor = true;
            //
```

```
            // btnUnlimitedApproveExp
            //
            this.btnUnlimitedApproveExp.Font = new
System.Drawing.Font("Microsoft Sans Serif", 7.5F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnUnlimitedApproveExp.Location = new
System.Drawing.Point(15, 19);
            this.btnUnlimitedApproveExp.Name =
"btnUnlimitedApproveExp";
            this.btnUnlimitedApproveExp.Size = new
System.Drawing.Size(94, 44);
            this.btnUnlimitedApproveExp.TabIndex = 9;
            this.btnUnlimitedApproveExp.Text = "Operation 10";
            this.btnUnlimitedApproveExp.UseVisualStyleBackColor =
true;
            //
            // btnRoleAssignment
            //
            this.btnRoleAssignment.Font = new
System.Drawing.Font("Microsoft Sans Serif", 7.5F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnRoleAssignment.Location = new
System.Drawing.Point(15, 168);
            this.btnRoleAssignment.Name = "btnRoleAssignment";
            this.btnRoleAssignment.Size = new System.Drawing.Size(94,
44);
            this.btnRoleAssignment.TabIndex = 6;
            this.btnRoleAssignment.Text = "Operation 13";
            this.btnRoleAssignment.UseVisualStyleBackColor = true;
            this.btnRoleAssignment.Click += new
System.EventHandler(this.btnRoleAssignment_Click);
            //
            // lblRoleAssignment
            //
            this.lblRoleAssignment.AutoSize = true;
            this.lblRoleAssignment.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblRoleAssignment.Location = new
System.Drawing.Point(115, 181);
            this.lblRoleAssignment.Name = "lblRoleAssignment";
            this.lblRoleAssignment.Size = new
System.Drawing.Size(101, 13);
            this.lblRoleAssignment.TabIndex = 8;
            this.lblRoleAssignment.Text = "Role Assignment";
            //
            // groupBox4
            //
            this.groupBox4.Controls.Add(this.lblGatherInfo);
            this.groupBox4.Controls.Add(this.lblCompCashFreeTrans);
            this.groupBox4.Controls.Add(this.btnGatherInfo);
            this.groupBox4.Controls.Add(this.btnCompCashFreeTrans);

this.groupBox4.Controls.Add(this.btnApproveHigherLimitTrans);
            this.groupBox4.Controls.Add(this.label1);
            this.groupBox4.Location = new System.Drawing.Point(376,
54);
```

```
            this.groupBox4.Name = "groupBox4";
            this.groupBox4.Size = new System.Drawing.Size(319, 176);
            this.groupBox4.TabIndex = 14;
            this.groupBox4.TabStop = false;
            this.groupBox4.Text = "Chief Teller Operations";
            //
            // lblGatherInfo
            //
            this.lblGatherInfo.AutoSize = true;
            this.lblGatherInfo.Font                =                new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold          |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblGatherInfo.Location              =              new
System.Drawing.Point(115, 138);
            this.lblGatherInfo.Name = "lblGatherInfo";
            this.lblGatherInfo.Size  =  new  System.Drawing.Size(112,
13);
            this.lblGatherInfo.TabIndex = 11;
            this.lblGatherInfo.Text = "Gather Information";
            //
            // lblCompCashFreeTrans
            //
            this.lblCompCashFreeTrans.AutoSize = true;
            this.lblCompCashFreeTrans.Font           =           new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold          |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblCompCashFreeTrans.Location         =          new
System.Drawing.Point(115, 88);
            this.lblCompCashFreeTrans.Name = "lblCompCashFreeTrans";
            this.lblCompCashFreeTrans.Size           =           new
System.Drawing.Size(176, 13);
            this.lblCompCashFreeTrans.TabIndex = 10;
            this.lblCompCashFreeTrans.Text   =   "Comp   Cash   Free
Transactions";
            this.lblCompCashFreeTrans.Click          +=          new
System.EventHandler(this.lblCompCashFreeTrans_Click);
            //
            // btnGatherInfo
            //
            this.btnGatherInfo.Font               =               new
System.Drawing.Font("Microsoft       Sans       Serif",       7.5F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnGatherInfo.Location             =             new
System.Drawing.Point(15, 122);
            this.btnGatherInfo.Name = "btnGatherInfo";
            this.btnGatherInfo.Size  =  new  System.Drawing.Size(94,
44);
            this.btnGatherInfo.TabIndex = 9;
            this.btnGatherInfo.Text = "Operation 9";
            this.btnGatherInfo.UseVisualStyleBackColor = true;
            //
            // btnCompCashFreeTrans
            //
            this.btnCompCashFreeTrans.Font           =           new
System.Drawing.Font("Microsoft       Sans       Serif",       7.5F,
```

```
System.Drawing.FontStyle.Bold,      System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnCompCashFreeTrans.Location          =          new
System.Drawing.Point(15, 72);
            this.btnCompCashFreeTrans.Name = "btnCompCashFreeTrans";
            this.btnCompCashFreeTrans.Size            =            new
System.Drawing.Size(94, 44);
            this.btnCompCashFreeTrans.TabIndex = 8;
            this.btnCompCashFreeTrans.Text = "Operation 8";
            this.btnCompCashFreeTrans.UseVisualStyleBackColor = true;
            //
            // btnApproveHigherLimitTrans
            //
            this.btnApproveHigherLimitTrans.Font        =          new
System.Drawing.Font("Microsoft        Sans        Serif",        7.5F,
System.Drawing.FontStyle.Bold,      System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnApproveHigherLimitTrans.Location      =       new
System.Drawing.Point(15, 22);
            this.btnApproveHigherLimitTrans.Name                    =
"btnApproveHigherLimitTrans";
            this.btnApproveHigherLimitTrans.Size         =         new
System.Drawing.Size(94, 44);
            this.btnApproveHigherLimitTrans.TabIndex = 5;
            this.btnApproveHigherLimitTrans.Text = "Operation 7";
            this.btnApproveHigherLimitTrans.UseVisualStyleBackColor =
true;
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Font    =    new    System.Drawing.Font("Microsoft
Sans                          Serif",                                8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold           |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.label1.Location = new System.Drawing.Point(115, 38);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(196, 13);
            this.label1.TabIndex = 7;
            this.label1.Text = "Approve Higher Limit Transaction";
            //
            // groupBox3
            //

this.groupBox3.Controls.Add(this.lblExpLimitChargingTrans);
            this.groupBox3.Controls.Add(this.lblExpLimitCreditTrans);

this.groupBox3.Controls.Add(this.lblInternalCommunication);
            this.groupBox3.Controls.Add(this.lblCCardNLoadRequests);
            this.groupBox3.Controls.Add(this.btnExpLimitCreditTans);

this.groupBox3.Controls.Add(this.btnExpLimitChargingTrans);

this.groupBox3.Controls.Add(this.btnInternalCommunication);
            this.groupBox3.Controls.Add(this.btnCCardNLoadRequests);
            this.groupBox3.Controls.Add(this.lblAccountCreation);
            this.groupBox3.Controls.Add(this.btnAccountCreation);
            this.groupBox3.Controls.Add(this.lblInformClients);
            this.groupBox3.Controls.Add(this.btnInformClients);
```

```
            this.groupBox3.Location = new System.Drawing.Point(34,
54);
            this.groupBox3.Name = "groupBox3";
            this.groupBox3.Size = new System.Drawing.Size(316, 408);
            this.groupBox3.TabIndex = 13;
            this.groupBox3.TabStop = false;
            this.groupBox3.Text = "Teller Operations";
            //
            // lblExpLimitChargingTrans
            //
            this.lblExpLimitChargingTrans.AutoSize = true;
            this.lblExpLimitChargingTrans.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblExpLimitChargingTrans.Location = new
System.Drawing.Point(115, 363);
            this.lblExpLimitChargingTrans.Name =
"lblExpLimitChargingTrans";
            this.lblExpLimitChargingTrans.Size = new
System.Drawing.Size(183, 13);
            this.lblExpLimitChargingTrans.TabIndex = 16;
            this.lblExpLimitChargingTrans.Text = "Exp Limit Charging
Transaction";
            //
            // lblExpLimitCreditTrans
            //
            this.lblExpLimitCreditTrans.AutoSize = true;
            this.lblExpLimitCreditTrans.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblExpLimitCreditTrans.Location = new
System.Drawing.Point(113, 305);
            this.lblExpLimitCreditTrans.Name =
"lblExpLimitCreditTrans";
            this.lblExpLimitCreditTrans.Size = new
System.Drawing.Size(166, 13);
            this.lblExpLimitCreditTrans.TabIndex = 15;
            this.lblExpLimitCreditTrans.Text = "Exp Limit Credit
Transaction";
            //
            // lblInternalCommunication
            //
            this.lblInternalCommunication.AutoSize = true;
            this.lblInternalCommunication.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblInternalCommunication.Location = new
System.Drawing.Point(115, 239);
            this.lblInternalCommunication.Name =
"lblInternalCommunication";
            this.lblInternalCommunication.Size = new
System.Drawing.Size(139, 13);
            this.lblInternalCommunication.TabIndex = 14;
            this.lblInternalCommunication.Text = "Internal
Communication";
```

```
            this.lblInternalCommunication.Click        +=        new
System.EventHandler(this.label2_Click);
            //
            // lblCCardNLoadRequests
            //
            this.lblCCardNLoadRequests.AutoSize = true;
            this.lblCCardNLoadRequests.Font            =            new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold          |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblCCardNLoadRequests.Location          =          new
System.Drawing.Point(115, 173);
            this.lblCCardNLoadRequests.Name                      =
"lblCCardNLoadRequests";
            this.lblCCardNLoadRequests.Size            =            new
System.Drawing.Size(179, 13);
            this.lblCCardNLoadRequests.TabIndex = 13;
            this.lblCCardNLoadRequests.Text  =  "Cash  Card  and  Loan
Requests";
            //
            // btnExpLimitCreditTans
            //
            this.btnExpLimitCreditTans.Font            =            new
System.Drawing.Font("Microsoft       Sans       Serif",       7.5F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnExpLimitCreditTans.Location          =          new
System.Drawing.Point(15, 289);
            this.btnExpLimitCreditTans.Name                      =
"btnExpLimitCreditTans";
            this.btnExpLimitCreditTans.Size            =            new
System.Drawing.Size(92, 44);
            this.btnExpLimitCreditTans.TabIndex = 11;
            this.btnExpLimitCreditTans.Text = "Operation 5";
            this.btnExpLimitCreditTans.UseVisualStyleBackColor       =
true;
            this.btnExpLimitCreditTans.Click           +=            new
System.EventHandler(this.btnExpLimitCreditTans_Click);
            //
            // btnExpLimitChargingTrans
            //
            this.btnExpLimitChargingTrans.Font          =          new
System.Drawing.Font("Microsoft       Sans       Serif",       7.5F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnExpLimitChargingTrans.Location        =        new
System.Drawing.Point(17, 347);
            this.btnExpLimitChargingTrans.Name                     =
"btnExpLimitChargingTrans";
            this.btnExpLimitChargingTrans.Size          =          new
System.Drawing.Size(92, 44);
            this.btnExpLimitChargingTrans.TabIndex = 12;
            this.btnExpLimitChargingTrans.Text = "Operation 6";
            this.btnExpLimitChargingTrans.UseVisualStyleBackColor    =
true;
            this.btnExpLimitChargingTrans.Click        +=        new
System.EventHandler(this.button4_Click);
            //
            // btnInternalCommunication
            //
```

```
            this.btnInternalCommunication.Font          =          new
System.Drawing.Font("Microsoft        Sans        Serif",        7.5F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnInternalCommunication.Location        =        new
System.Drawing.Point(15, 223);
            this.btnInternalCommunication.Name                      =
"btnInternalCommunication";
            this.btnInternalCommunication.Size          =          new
System.Drawing.Size(92, 44);
            this.btnInternalCommunication.TabIndex = 10;
            this.btnInternalCommunication.Text = "Operation 4";
            this.btnInternalCommunication.UseVisualStyleBackColor    =
true;
            this.btnInternalCommunication.Click        +=          new
System.EventHandler(this.button2_Click);
            //
            // btnCCardNLoadRequests
            //
            this.btnCCardNLoadRequests.Font          =          new
System.Drawing.Font("Microsoft        Sans        Serif",        7.5F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnCCardNLoadRequests.Location         =         new
System.Drawing.Point(15, 157);
            this.btnCCardNLoadRequests.Name                         =
"btnCCardNLoadRequests";
            this.btnCCardNLoadRequests.Size           =           new
System.Drawing.Size(92, 44);
            this.btnCCardNLoadRequests.TabIndex = 9;
            this.btnCCardNLoadRequests.Text = "Operation 3";
            this.btnCCardNLoadRequests.UseVisualStyleBackColor      =
true;
            //
            // lblAccountCreation
            //
            this.lblAccountCreation.AutoSize = true;
            this.lblAccountCreation.Font            =           new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold          |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblAccountCreation.Location          =           new
System.Drawing.Point(115, 103);
            this.lblAccountCreation.Name = "lblAccountCreation";
            this.lblAccountCreation.Size            =           new
System.Drawing.Size(105, 13);
            this.lblAccountCreation.TabIndex = 4;
            this.lblAccountCreation.Text = "Account Creation";
            //
            // btnAccountCreation
            //
            this.btnAccountCreation.Font            =           new
System.Drawing.Font("Microsoft        Sans        Serif",        7.5F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnAccountCreation.Location          =           new
System.Drawing.Point(15, 88);
            this.btnAccountCreation.Name = "btnAccountCreation";
            this.btnAccountCreation.Size            =           new
System.Drawing.Size(94, 44);
```

```
            this.btnAccountCreation.TabIndex = 3;
            this.btnAccountCreation.Text = "Operation 2";
            this.btnAccountCreation.UseVisualStyleBackColor = true;
            this.btnAccountCreation.Click           +=           new
System.EventHandler(this.btnAccountCreation_Click);
            //
            // lblInformClients
            //
            this.lblInformClients.AutoSize = true;
            this.lblInformClients.Font              =              new
System.Drawing.Font("Microsoft        Sans        Serif",        9F,
((System.Drawing.FontStyle)((System.Drawing.FontStyle.Bold          |
System.Drawing.FontStyle.Italic))),
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblInformClients.Location           =           new
System.Drawing.Point(115, 36);
            this.lblInformClients.Name = "lblInformClients";
            this.lblInformClients.Size = new System.Drawing.Size(96,
15);
            this.lblInformClients.TabIndex = 2;
            this.lblInformClients.Text = "Inform Clients";
            //
            // btnInformClients
            //
            this.btnInformClients.Font              =              new
System.Drawing.Font("Microsoft        Sans        Serif",        7.5F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.btnInformClients.Location           =           new
System.Drawing.Point(15, 22);
            this.btnInformClients.Name = "btnInformClients";
            this.btnInformClients.Size = new System.Drawing.Size(94,
44);
            this.btnInformClients.TabIndex = 1;
            this.btnInformClients.Text = "Operation 1";
            this.btnInformClients.UseVisualStyleBackColor = true;
            this.btnInformClients.Click           +=           new
System.EventHandler(this.btnInformClients_Click);
            //
            // lblMessage1
            //
            this.lblMessage1.AccessibleDescription = "";
            this.lblMessage1.AutoSize = true;
            this.lblMessage1.Font                =                new
System.Drawing.Font("Microsoft        Sans        Serif",        8.25F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblMessage1.Location = new System.Drawing.Point(36,
25);
            this.lblMessage1.Name = "lblMessage1";
            this.lblMessage1.Size = new System.Drawing.Size(37, 13);
            this.lblMessage1.TabIndex = 0;
            this.lblMessage1.Text = "Msg1";
            this.lblMessage1.Click              +=              new
System.EventHandler(this.label1_Click);
            //
            // pictureBox1
            //
            this.pictureBox1.Image                                  =
global::BankApplication.Properties.Resources.img1;
```

```
            this.pictureBox1.InitialImage                                    =
global::BankApplication.Properties.Resources.img1;
            this.pictureBox1.Location = new System.Drawing.Point(229,
12);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(394, 77);
            this.pictureBox1.TabIndex = 1;
            this.pictureBox1.TabStop = false;
            //
            // frmMain
            //
            this.AutoScaleDimensions  =  new  System.Drawing.SizeF(6F,
13F);
            this.AutoScaleMode                                               =
System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(852, 738);
            this.Controls.Add(this.pictureBox1);
            this.Controls.Add(this.groupBox1);
            this.Name = "frmMain";
            this.Text = "Bank Client Application";
            this.Load += new System.EventHandler(this.frmMain_Load);
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            this.groupBox2.ResumeLayout(false);
            this.groupBox5.ResumeLayout(false);
            this.groupBox5.PerformLayout();
            this.groupBox4.ResumeLayout(false);
            this.groupBox4.PerformLayout();
            this.groupBox3.ResumeLayout(false);
            this.groupBox3.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndIni
t();
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.PictureBox pictureBox1;
        internal System.Windows.Forms.Label lblMessage1;
        internal System.Windows.Forms.Button btnExit;
        private System.Windows.Forms.Label lblInformClients;
        private System.Windows.Forms.Button btnInformClients;
        private System.Windows.Forms.GroupBox groupBox2;
        private System.Windows.Forms.Button btnAccountCreation;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Button btnRoleAssignment;
        private                              System.Windows.Forms.Button
btnApproveHigherLimitTrans;
        private System.Windows.Forms.Label lblAccountCreation;
        private System.Windows.Forms.Label lblRoleAssignment;
        private System.Windows.Forms.Label lblMessage2;
        private System.Windows.Forms.Button btnExpLimitCreditTans;
        private System.Windows.Forms.Button btnInternalCommunication;
        private System.Windows.Forms.Button btnCCardNLoadRequests;
        private System.Windows.Forms.GroupBox groupBox5;
        private System.Windows.Forms.GroupBox groupBox4;
        private System.Windows.Forms.GroupBox groupBox3;
        private System.Windows.Forms.Button btnExpLimitChargingTrans;
```

```
        private System.Windows.Forms.Label lblCCardNLoadRequests;
        private System.Windows.Forms.Label lblInternalCommunication;
        private System.Windows.Forms.Label lblExpLimitCreditTrans;
        private System.Windows.Forms.Label lblCompCashFreeTrans;
        private System.Windows.Forms.Button btnGatherInfo;
        private System.Windows.Forms.Button btnCompCashFreeTrans;
        private System.Windows.Forms.Label lblExpLimitChargingTrans;
        private System.Windows.Forms.Label lblGatherInfo;
        private System.Windows.Forms.Button btnUnlimitedApproveExp;
        private System.Windows.Forms.Button btnGatherCursialInfo;
        private System.Windows.Forms.Button btnClassifiedDocs;
        private System.Windows.Forms.Label lblGatherCursialInfo;
        private System.Windows.Forms.Label lblClassifiedDocs;
        private System.Windows.Forms.Label lblUnlimitedApproveExp;
    }
}
```

frmMain.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Security.Principal;
using AZROLESLib;
using System.Threading;
using System.Text.RegularExpressions;
using System.Security;


namespace BankApplication
{
    //Κύρια φόρμα, frmMain, του BankApplication. Μέσω αυτής γίνεται
αρχικοποίηση
    //των πλήκτρων των λειτουργιών για τον χρήστη που έχει κάνει log
in στο σύστημα.

    public partial class frmMain : Form
    {
        public frmMain()
        {
            InitializeComponent();

            UserName = Program.LoggedUser();

        }

        public WindowsIdentity mIdentity;

        string UserName;
```

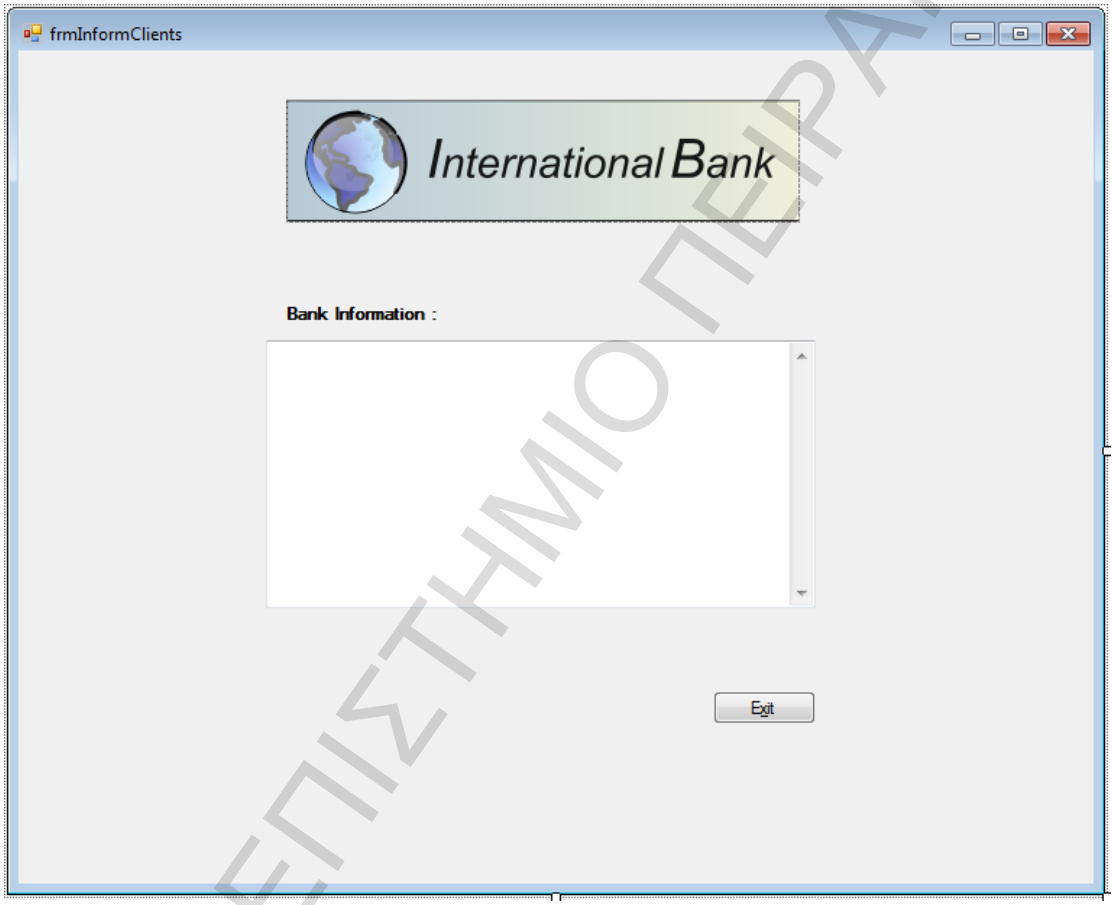
```
        private void frmMain_Load(object sender, EventArgs e) {


            WindowsIdentity userIdentity =
WindowsIdentity.GetCurrent();

            lblMessage1.Text = string.Concat("User: ",
Program.LoggedUser());

            AuthenticationEnableFeatures();


        }

        private void AuthenticationEnableFeatures() {
            Operations[] ops = {
                Operations.Inform_Clients,
                Operations.Account_Creation,
                Operations.C_Cards_n_Loan_Requests,
                Operations.Internal_Communication,
                Operations.Exp_Limit_Credit_Trans,
                Operations.Exp_Limit_Charging_Trans,
                Operations.Approve_Higher_Limit_Trans,
                Operations.Comp_Cash_Free_Trans,
                Operations.Gather_Info,
                Operations.Unlimited_Approve_Exp,
                Operations.Classified_Docs,
                Operations.Gather_Cursial_Info,
                Operations.Role_Assignment,
            };


            //Κλήση της MultipleAccessCheck και ενεργοποίηση των
Operation Buttons για
            //τα εκάστοτε Operation στα οποία έχει πρόσβαση ο
χρήστης.
            bool[] results = Program.MultipleAccessCheck("Enable
buttons", ops);

            btnInformClients.Enabled = results[0];
            btnAccountCreation.Enabled = results[1];
            btnCCardNLoadRequests.Enabled = results[2];
            btnInternalCommunication.Enabled = results[3];
            btnExpLimitCreditTans.Enabled = results[4];
            btnExpLimitChargingTrans.Enabled = results[5];
            btnApproveHigherLimitTrans.Enabled = results[6];
            btnCompCashFreeTrans.Enabled = results[7];
            btnGatherInfo.Enabled = results[8];
            btnUnlimitedApproveExp.Enabled = results[9];
            btnClassifiedDocs.Enabled = results[10];
            btnGatherCursialInfo.Enabled = results[11];
            btnRoleAssignment.Enabled = results[12];
        }

        private void EstablishWindowsIdentity()
        {
            WindowsPrincipal myPrincipal = new
WindowsPrincipal(WindowsIdentity.GetCurrent());
            mIdentity = WindowsIdentity.GetCurrent();
        }
```

```
        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void groupBox1_Enter(object sender, EventArgs e)
        {

        }

        private void frmMain_Load_1(object sender, EventArgs e)
        {

        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button2_Click(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private   void   btnExpLimitCreditTans_Click(object   sender,
EventArgs e)
        {
            new frmCreditTransaction().Show(this);
        }

        private void button4_Click(object sender, EventArgs e)
        {

        }

        private   void   lblCompCashFreeTrans_Click(object   sender,
EventArgs e)
        {

        }

        private void btnRoleAssignment_Click(object sender, EventArgs
e)
        {
            new frmRoleAssignment().Show(this);
        }

        private void btnInformClients_Click(object sender, EventArgs
e)
        {
            new frmInformClients().Show(this);
        }
```

```
        private void btnAccountCreation_Click(object sender,
EventArgs e)
        {
            new frmAccountCreation().Show(this);
        }
    }
}
```

frmInformClients

[Design]:



frmInformClients.Designer.cs:

```
namespace BankApplication
{
    partial class frmInformClients
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
```

```
        /// <param name="disposing">true if managed resources should
be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.txtBxInformClients                    =                new
System.Windows.Forms.TextBox();
            this.lblInformation = new System.Windows.Forms.Label();
            this.btnExit = new System.Windows.Forms.Button();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginI
nit();
            this.SuspendLayout();
            //
            // pictureBox1
            //
            this.pictureBox1.Image                                      =
global::BankApplication.Properties.Resources.img1;
            this.pictureBox1.InitialImage                               =
global::BankApplication.Properties.Resources.img1;
            this.pictureBox1.Location = new System.Drawing.Point(185,
34);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(355, 85);
            this.pictureBox1.TabIndex = 1;
            this.pictureBox1.TabStop = false;
            //
            // txtBxInformClients
            //
            this.txtBxInformClients.Location           =           new
System.Drawing.Point(171, 200);
            this.txtBxInformClients.Multiline = true;
            this.txtBxInformClients.Name = "txtBxInformClients";
            this.txtBxInformClients.ScrollBars                        =
System.Windows.Forms.ScrollBars.Vertical;
            this.txtBxInformClients.Size             =             new
System.Drawing.Size(380, 185);
            this.txtBxInformClients.TabIndex = 2;
            //
            // lblInformation
            //
            this.lblInformation.AutoSize = true;
            this.lblInformation.Font                 =                new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
```

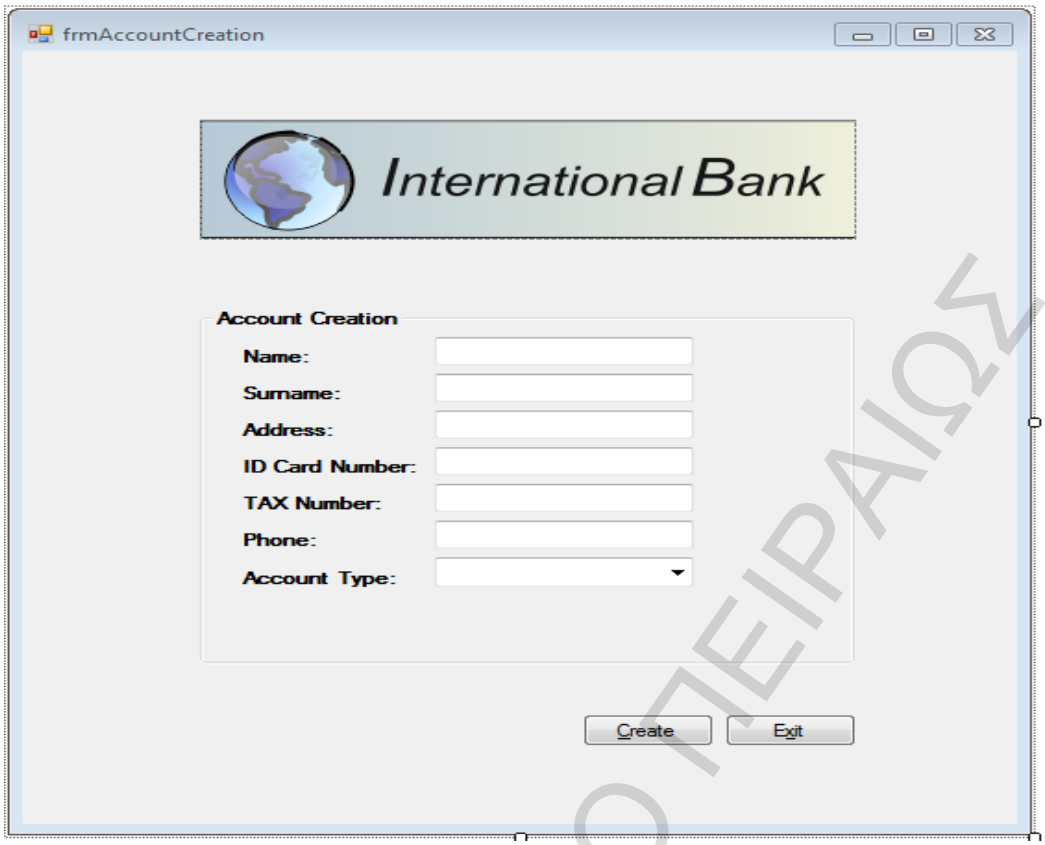
```
            this.lblInformation.Location            =            new
System.Drawing.Point(182, 175);
            this.lblInformation.Name = "lblInformation";
            this.lblInformation.Size  =  new  System.Drawing.Size(111,
13);
            this.lblInformation.TabIndex = 3;
            this.lblInformation.Text = "Bank Information :";
            //
            // btnExit
            //
            this.btnExit.DialogResult                          =
System.Windows.Forms.DialogResult.Cancel;
            this.btnExit.Location   =   new   System.Drawing.Point(480,
442);
            this.btnExit.Name = "btnExit";
            this.btnExit.Size = new System.Drawing.Size(71, 23);
            this.btnExit.TabIndex = 4;
            this.btnExit.Text = "E&xit";
            this.btnExit.UseVisualStyleBackColor = true;
            this.btnExit.Click                 +=                new
System.EventHandler(this.btnExit_Click);
            //
            // frmInformClients
            //
            this.AutoScaleDimensions  =  new  System.Drawing.SizeF(6F,
13F);
            this.AutoScaleMode                                     =
System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(743, 575);
            this.Controls.Add(this.btnExit);
            this.Controls.Add(this.lblInformation);
            this.Controls.Add(this.txtBxInformClients);
            this.Controls.Add(this.pictureBox1);
            this.Name = "frmInformClients";
            this.Text = "frmInformClients";
            this.Load                      +=                    new
System.EventHandler(this.frmInformClients_Load);

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndIni
t();
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.PictureBox pictureBox1;
        private System.Windows.Forms.TextBox txtBxInformClients;
        private System.Windows.Forms.Label lblInformation;
        internal System.Windows.Forms.Button btnExit;
    }
}
```

frmInformClients.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace BankApplication
{
    //Φόρμα frmInformClients. Παρέχει σε textBox πληροφορίες για την τράπεζα.
    //Αντιστοιχει στο Operation 1, Inform Clients, και είναι προσβάσιμη από όλους
    //τους ρόλους του Authorization Store.

    public partial class frmInformClients : Form
    {
        public frmInformClients()
        {
            InitializeComponent();

            this.txtBxInformClients.Text = "International Bank:";
            this.txtBxInformClients.Text += Environment.NewLine;
            this.txtBxInformClients.Text  += "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum";
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

frmAccountCreation:

[Design]:



frmAccountCreation.Designer.cs:

```
namespace BankApplication
{
    partial class frmAccountCreation
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should
be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
```

```
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.btnExit = new System.Windows.Forms.Button();
            this.btnCreateUser = new System.Windows.Forms.Button();
            this.lblName = new System.Windows.Forms.Label();
            this.lblSurname = new System.Windows.Forms.Label();
            this.lblAddress = new System.Windows.Forms.Label();
            this.lblIDCardNumber = new System.Windows.Forms.Label();
            this.lblTaxNumber = new System.Windows.Forms.Label();
            this.lblPhone = new System.Windows.Forms.Label();
            this.lblAccountType = new System.Windows.Forms.Label();
            this.cmbBoxAccountTypes                =                new
System.Windows.Forms.ComboBox();
            this.txtBxName = new System.Windows.Forms.TextBox();
            this.txtBxSurname = new System.Windows.Forms.TextBox();
            this.txtBxIDCardNumber                 =                new
System.Windows.Forms.TextBox();
            this.txtBxTaxNumber = new System.Windows.Forms.TextBox();
            this.txtBxAddress = new System.Windows.Forms.TextBox();
            this.textBox6 = new System.Windows.Forms.TextBox();
            this.groupBox1 = new System.Windows.Forms.GroupBox();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginI
nit();
            this.groupBox1.SuspendLayout();
            this.SuspendLayout();
            //
            // pictureBox1
            //
            this.pictureBox1.Image                                         =
global::BankApplication.Properties.Resources.img1;
            this.pictureBox1.InitialImage                                  =
global::BankApplication.Properties.Resources.img1;
            this.pictureBox1.Location = new System.Drawing.Point(96,
49);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(355, 85);
            this.pictureBox1.TabIndex = 2;
            this.pictureBox1.TabStop = false;
            //
            // btnExit
            //
            this.btnExit.DialogResult                                      =
System.Windows.Forms.DialogResult.Cancel;
            this.btnExit.Location  =  new  System.Drawing.Point(380,
470);
            this.btnExit.Name = "btnExit";
            this.btnExit.Size = new System.Drawing.Size(71, 23);
            this.btnExit.TabIndex = 5;
            this.btnExit.Text = "E&xit";
            this.btnExit.UseVisualStyleBackColor = true;
            this.btnExit.Click                  +=                 new
System.EventHandler(this.btnExit_Click);
            //
            // btnCreateUser
            //
            this.btnCreateUser.DialogResult                                =
System.Windows.Forms.DialogResult.Cancel;
```

```
            this.btnCreateUser.Location = new
System.Drawing.Point(303, 470);
            this.btnCreateUser.Name = "btnCreateUser";
            this.btnCreateUser.Size = new System.Drawing.Size(71,
23);
            this.btnCreateUser.TabIndex = 6;
            this.btnCreateUser.Text = "&Create";
            this.btnCreateUser.UseVisualStyleBackColor = true;
            //
            // lblName
            //
            this.lblName.AutoSize = true;
            this.lblName.Font = new System.Drawing.Font("Microsoft
Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblName.Location = new System.Drawing.Point(20, 27);
            this.lblName.Name = "lblName";
            this.lblName.Size = new System.Drawing.Size(43, 13);
            this.lblName.TabIndex = 7;
            this.lblName.Text = "Name:";
            //
            // lblSurname
            //
            this.lblSurname.AutoSize = true;
            this.lblSurname.Font = new System.Drawing.Font("Microsoft
Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblSurname.Location = new System.Drawing.Point(20,
53);
            this.lblSurname.Name = "lblSurname";
            this.lblSurname.Size = new System.Drawing.Size(60, 13);
            this.lblSurname.TabIndex = 8;
            this.lblSurname.Text = "Surname:";
            //
            // lblAddress
            //
            this.lblAddress.AutoSize = true;
            this.lblAddress.Font = new System.Drawing.Font("Microsoft
Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblAddress.Location = new System.Drawing.Point(20,
79);
            this.lblAddress.Name = "lblAddress";
            this.lblAddress.Size = new System.Drawing.Size(56, 13);
            this.lblAddress.TabIndex = 9;
            this.lblAddress.Text = "Address:";
            //
            // lblIDCardNumber
            //
            this.lblIDCardNumber.AutoSize = true;
            this.lblIDCardNumber.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblIDCardNumber.Location = new
System.Drawing.Point(20, 105);
            this.lblIDCardNumber.Name = "lblIDCardNumber";
            this.lblIDCardNumber.Size = new System.Drawing.Size(101,
13);
            this.lblIDCardNumber.TabIndex = 10;
            this.lblIDCardNumber.Text = "ID Card Number:";
```

```
            //
            // lblTaxNumber
            //
            this.lblTaxNumber.AutoSize = true;
            this.lblTaxNumber.Font                =                new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblTaxNumber.Location = new System.Drawing.Point(20,
131);
            this.lblTaxNumber.Name = "lblTaxNumber";
            this.lblTaxNumber.Size = new System.Drawing.Size(82, 13);
            this.lblTaxNumber.TabIndex = 11;
            this.lblTaxNumber.Text = "TAX Number:";
            //
            // lblPhone
            //
            this.lblPhone.AutoSize = true;
            this.lblPhone.Font  =  new  System.Drawing.Font("Microsoft
Sans       Serif",       8.25F,       System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblPhone.Location   =   new   System.Drawing.Point(20,
157);
            this.lblPhone.Name = "lblPhone";
            this.lblPhone.Size = new System.Drawing.Size(47, 13);
            this.lblPhone.TabIndex = 12;
            this.lblPhone.Text = "Phone:";
            //
            // lblAccountType
            //
            this.lblAccountType.AutoSize = true;
            this.lblAccountType.Font              =                new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblAccountType.Location            =            new
System.Drawing.Point(20, 184);
            this.lblAccountType.Name = "lblAccountType";
            this.lblAccountType.Size   =   new   System.Drawing.Size(90,
13);
            this.lblAccountType.TabIndex = 13;
            this.lblAccountType.Text = "Account Type:";
            //
            // cmbBoxAccountTypes
            //
            this.cmbBoxAccountTypes.FormattingEnabled = true;
            this.cmbBoxAccountTypes.Location          =          new
System.Drawing.Point(127, 176);
            this.cmbBoxAccountTypes.Name = "cmbBoxAccountTypes";
            this.cmbBoxAccountTypes.Size            =            new
System.Drawing.Size(140, 21);
            this.cmbBoxAccountTypes.TabIndex = 14;
            this.cmbBoxAccountTypes.SelectedIndexChanged    +=    new
System.EventHandler(this.cmbBoxAccountTypes_SelectedIndexChanged);
            //
            // txtBxName
            //
            this.txtBxName.Location  =  new  System.Drawing.Point(127,
20);
            this.txtBxName.Name = "txtBxName";
            this.txtBxName.Size = new System.Drawing.Size(140, 20);
```

```
            this.txtBxName.TabIndex = 15;
            //
            // txtBxSurname
            //
            this.txtBxSurname.Location              =              new
System.Drawing.Point(127, 46);
            this.txtBxSurname.Name = "txtBxSurname";
            this.txtBxSurname.Size   =   new   System.Drawing.Size(140,
20);
            this.txtBxSurname.TabIndex = 17;
            //
            // txtBxIDCardNumber
            //
            this.txtBxIDCardNumber.Location           =              new
System.Drawing.Point(127, 98);
            this.txtBxIDCardNumber.Name = "txtBxIDCardNumber";
            this.txtBxIDCardNumber.Size               =              new
System.Drawing.Size(140, 20);
            this.txtBxIDCardNumber.TabIndex = 19;
            //
            // txtBxTaxNumber
            //
            this.txtBxTaxNumber.Location              =              new
System.Drawing.Point(127, 124);
            this.txtBxTaxNumber.Name = "txtBxTaxNumber";
            this.txtBxTaxNumber.Size  =  new  System.Drawing.Size(140,
20);
            this.txtBxTaxNumber.TabIndex = 21;
            //
            // txtBxAddress
            //
            this.txtBxAddress.Location              =              new
System.Drawing.Point(127, 72);
            this.txtBxAddress.Name = "txtBxAddress";
            this.txtBxAddress.Size   =   new   System.Drawing.Size(140,
20);
            this.txtBxAddress.TabIndex = 23;
            //
            // textBox6
            //
            this.textBox6.Location  =  new  System.Drawing.Point(127,
150);
            this.textBox6.Name = "textBox6";
            this.textBox6.Size = new System.Drawing.Size(140, 20);
            this.textBox6.TabIndex = 24;
            //
            // groupBox1
            //
            this.groupBox1.Controls.Add(this.textBox6);
            this.groupBox1.Controls.Add(this.txtBxAddress);
            this.groupBox1.Controls.Add(this.txtBxTaxNumber);
            this.groupBox1.Controls.Add(this.txtBxIDCardNumber);
            this.groupBox1.Controls.Add(this.txtBxSurname);
            this.groupBox1.Controls.Add(this.txtBxName);
            this.groupBox1.Controls.Add(this.cmbBoxAccountTypes);
            this.groupBox1.Controls.Add(this.lblAccountType);
            this.groupBox1.Controls.Add(this.lblPhone);
            this.groupBox1.Controls.Add(this.lblTaxNumber);
            this.groupBox1.Controls.Add(this.lblIDCardNumber);
            this.groupBox1.Controls.Add(this.lblAddress);
            this.groupBox1.Controls.Add(this.lblSurname);
```

```
            this.groupBox1.Controls.Add(this.lblName);
            this.groupBox1.Font = new System.Drawing.Font("Microsoft
Sans       Serif",       8.25F,       System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.groupBox1.Location = new System.Drawing.Point(96,
183);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(354, 252);
            this.groupBox1.TabIndex = 25;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Account Creation";
            //
            // frmAccountCreation
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F,
13F);
            this.AutoScaleMode                                    =
System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(538, 548);
            this.Controls.Add(this.groupBox1);
            this.Controls.Add(this.btnCreateUser);
            this.Controls.Add(this.btnExit);
            this.Controls.Add(this.pictureBox1);
            this.Name = "frmAccountCreation";
            this.Text = "frmAccountCreation";
            this.Load                    +=                    new
System.EventHandler(this.frmAccountCreation_Load);

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndIni
t();
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.PictureBox pictureBox1;
        internal System.Windows.Forms.Button btnExit;
        internal System.Windows.Forms.Button btnCreateUser;
        private System.Windows.Forms.Label lblName;
        private System.Windows.Forms.Label lblSurname;
        private System.Windows.Forms.Label lblAddress;
        private System.Windows.Forms.Label lblIDCardNumber;
        private System.Windows.Forms.Label lblTaxNumber;
        private System.Windows.Forms.Label lblPhone;
        private System.Windows.Forms.Label lblAccountType;
        private System.Windows.Forms.ComboBox cmbBoxAccountTypes;
        private System.Windows.Forms.TextBox txtBxName;
        private System.Windows.Forms.TextBox txtBxSurname;
        private System.Windows.Forms.TextBox txtBxIDCardNumber;
        private System.Windows.Forms.TextBox txtBxTaxNumber;
        private System.Windows.Forms.TextBox txtBxAddress;
        private System.Windows.Forms.TextBox textBox6;
        private System.Windows.Forms.GroupBox groupBox1;
    }
}
```

frmAccountCreation.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace BankApplication
{
    //Φόρμα frmAccountCreation. Πρόκειται για το Operation με ID = 2.
    //Στη φόρμα αυτή ο χρήστης μπορεί να δημιουργήσει λογαριασμό νέου
    //πελάτη στην τράπεζα.

    public partial class frmAccountCreation : Form
    {
        public frmAccountCreation()
        {
            InitializeComponent();
            cmbBoxAccountTypes.Items.Add("Savings");
            cmbBoxAccountTypes.Items.Add("Checkings");
        }

        private void frmAccountCreation_Load(object sender, EventArgs
e)
        {

        }

        private  void  cmbBoxAccountTypes_SelectedIndexChanged(object
sender, EventArgs e)
        {

        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

frmCreditTransaction:

[Design]:

frmCreditTansaction.Designer.cs:

```
namespace BankApplication
{
    partial class frmCreditTransaction
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should
be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
```

```
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.lblDebitAccount = new System.Windows.Forms.Label();
            this.lblCreditAccoung = new System.Windows.Forms.Label();
            this.lblAmount = new System.Windows.Forms.Label();
            this.txtBxDebitAccount = new
System.Windows.Forms.TextBox();
            this.txtBxCreditAccount = new
System.Windows.Forms.TextBox();
            this.txtBxAmount = new System.Windows.Forms.TextBox();
            this.btnExit = new System.Windows.Forms.Button();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.lblMessage1 = new System.Windows.Forms.Label();
            this.lblMessage2 = new System.Windows.Forms.Label();
            this.btnSubmit = new System.Windows.Forms.Button();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginI
nit();
            this.groupBox1.SuspendLayout();
            this.SuspendLayout();
            //
            // pictureBox1
            //
            this.pictureBox1.Image =
global::BankApplication.Properties.Resources.img1;
            this.pictureBox1.InitialImage =
global::BankApplication.Properties.Resources.img1;
            this.pictureBox1.Location = new System.Drawing.Point(94,
47);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(394, 77);
            this.pictureBox1.TabIndex = 2;
            this.pictureBox1.TabStop = false;
            //
            // lblDebitAccount
            //
            this.lblDebitAccount.AutoSize = true;
            this.lblDebitAccount.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblDebitAccount.Location = new
System.Drawing.Point(21, 41);
            this.lblDebitAccount.Name = "lblDebitAccount";
            this.lblDebitAccount.Size = new System.Drawing.Size(92,
13);
            this.lblDebitAccount.TabIndex = 3;
            this.lblDebitAccount.Text = "Debit Account:";
            //
            // lblCreditAccoung
            //
            this.lblCreditAccoung.AutoSize = true;
            this.lblCreditAccoung.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblCreditAccoung.Location = new
System.Drawing.Point(21, 76);
```

```
            this.lblCreditAccoung.Name = "lblCreditAccoung";
            this.lblCreditAccoung.Size = new System.Drawing.Size(95,
13);
            this.lblCreditAccoung.TabIndex = 4;
            this.lblCreditAccoung.Text = "Credit Account:";
            //
            // lblAmount
            //
            this.lblAmount.AutoSize = true;
            this.lblAmount.Font = new System.Drawing.Font("Microsoft
Sans       Serif",       8.25F,       System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblAmount.Location = new System.Drawing.Point(21,
105);
            this.lblAmount.Name = "lblAmount";
            this.lblAmount.Size = new System.Drawing.Size(53, 13);
            this.lblAmount.TabIndex = 5;
            this.lblAmount.Text = "Amount:";
            //
            // txtBxDebitAccount
            //
            this.txtBxDebitAccount.Location = new
System.Drawing.Point(155, 38);
            this.txtBxDebitAccount.MaxLength = 10;
            this.txtBxDebitAccount.Name = "txtBxDebitAccount";
            this.txtBxDebitAccount.Size = new
System.Drawing.Size(223, 20);
            this.txtBxDebitAccount.TabIndex = 6;
            this.txtBxDebitAccount.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.txtBxDebitAccount_KeyP
ress);
            //
            // txtBxCreditAccount
            //
            this.txtBxCreditAccount.Location = new
System.Drawing.Point(155, 73);
            this.txtBxCreditAccount.MaxLength = 10;
            this.txtBxCreditAccount.Name = "txtBxCreditAccount";
            this.txtBxCreditAccount.Size = new
System.Drawing.Size(223, 20);
            this.txtBxCreditAccount.TabIndex = 7;
            this.txtBxCreditAccount.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.txtBxCreditAccount_Key
Press);
            //
            // txtBxAmount
            //
            this.txtBxAmount.Location = new System.Drawing.Point(155,
102);
            this.txtBxAmount.Name = "txtBxAmount";
            this.txtBxAmount.Size = new System.Drawing.Size(109, 20);
            this.txtBxAmount.TabIndex = 8;
            this.txtBxAmount.TextAlign =
System.Windows.Forms.HorizontalAlignment.Right;
            this.txtBxAmount.TextChanged += new
System.EventHandler(this.txtBxAmount_TextChanged);
            this.txtBxAmount.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.txtBxAmount_KeyPress);
            //
            // btnExit
            //
```

```
            this.btnExit.DialogResult                                    =
System.Windows.Forms.DialogResult.Cancel;
            this.btnExit.Location   =   new   System.Drawing.Point(416,
345);
            this.btnExit.Name = "btnExit";
            this.btnExit.Size = new System.Drawing.Size(71, 23);
            this.btnExit.TabIndex = 9;
            this.btnExit.Text = "E&xit";
            this.btnExit.UseVisualStyleBackColor = true;
            this.btnExit.Click                  +=                  new
System.EventHandler(this.btnExit_Click);
            //
            // groupBox1
            //
            this.groupBox1.Controls.Add(this.lblMessage1);
            this.groupBox1.Controls.Add(this.lblMessage2);
            this.groupBox1.Controls.Add(this.txtBxAmount);
            this.groupBox1.Controls.Add(this.txtBxCreditAccount);
            this.groupBox1.Controls.Add(this.txtBxDebitAccount);
            this.groupBox1.Controls.Add(this.lblAmount);
            this.groupBox1.Controls.Add(this.lblCreditAccoung);
            this.groupBox1.Controls.Add(this.lblDebitAccount);
            this.groupBox1.Font  =  new  System.Drawing.Font("Microsoft
Sans       Serif",       8.25F,       System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.groupBox1.Location   =   new   System.Drawing.Point(94,
156);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(393, 183);
            this.groupBox1.TabIndex = 10;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Credit Transaction";
            //
            // lblMessage1
            //
            this.lblMessage1.AutoSize = true;
            this.lblMessage1.ForeColor                                   =
System.Drawing.SystemColors.ActiveCaptionText;
            this.lblMessage1.Location  =  new  System.Drawing.Point(21,
137);
            this.lblMessage1.Name = "lblMessage1";
            this.lblMessage1.Size = new System.Drawing.Size(64, 13);
            this.lblMessage1.TabIndex = 10;
            this.lblMessage1.Text = "Message1";
            this.lblMessage1.Visible = false;
            //
            // lblMessage2
            //
            this.lblMessage2.AutoSize = true;
            this.lblMessage2.ForeColor = System.Drawing.Color.Red;
            this.lblMessage2.Location  =  new  System.Drawing.Point(21,
151);
            this.lblMessage2.Name = "lblMessage2";
            this.lblMessage2.Size = new System.Drawing.Size(64, 13);
            this.lblMessage2.TabIndex = 9;
            this.lblMessage2.Text = "Message2";
            this.lblMessage2.Visible = false;
            //
            // btnSubmit
            //
```

```
            this.btnSubmit.Location = new System.Drawing.Point(339,
345);
            this.btnSubmit.Name = "btnSubmit";
            this.btnSubmit.Size = new System.Drawing.Size(71, 23);
            this.btnSubmit.TabIndex = 11;
            this.btnSubmit.Text = "Submit";
            this.btnSubmit.UseVisualStyleBackColor = true;
            this.btnSubmit.Click                    +=                  new
System.EventHandler(this.btnSubmit_Click);
            //
            // frmCreditTransaction
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F,
13F);
            this.AutoScaleMode                                          =
System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(571, 404);
            this.Controls.Add(this.btnSubmit);
            this.Controls.Add(this.groupBox1);
            this.Controls.Add(this.btnExit);
            this.Controls.Add(this.pictureBox1);
            this.Name = "frmCreditTransaction";
            this.Text = "CreditTransaction";

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndIni
t();
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.PictureBox pictureBox1;
        private System.Windows.Forms.Label lblDebitAccount;
        private System.Windows.Forms.Label lblCreditAccoung;
        private System.Windows.Forms.Label lblAmount;



        private System.Windows.Forms.TextBox txtBxDebitAccount;
        private System.Windows.Forms.TextBox txtBxCreditAccount;
        private System.Windows.Forms.TextBox txtBxAmount;
        internal System.Windows.Forms.Button btnExit;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.Button btnSubmit;
        private System.Windows.Forms.Label lblMessage1;
        private System.Windows.Forms.Label lblMessage2;
    }
}
```

frmCreditTransaction.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
```

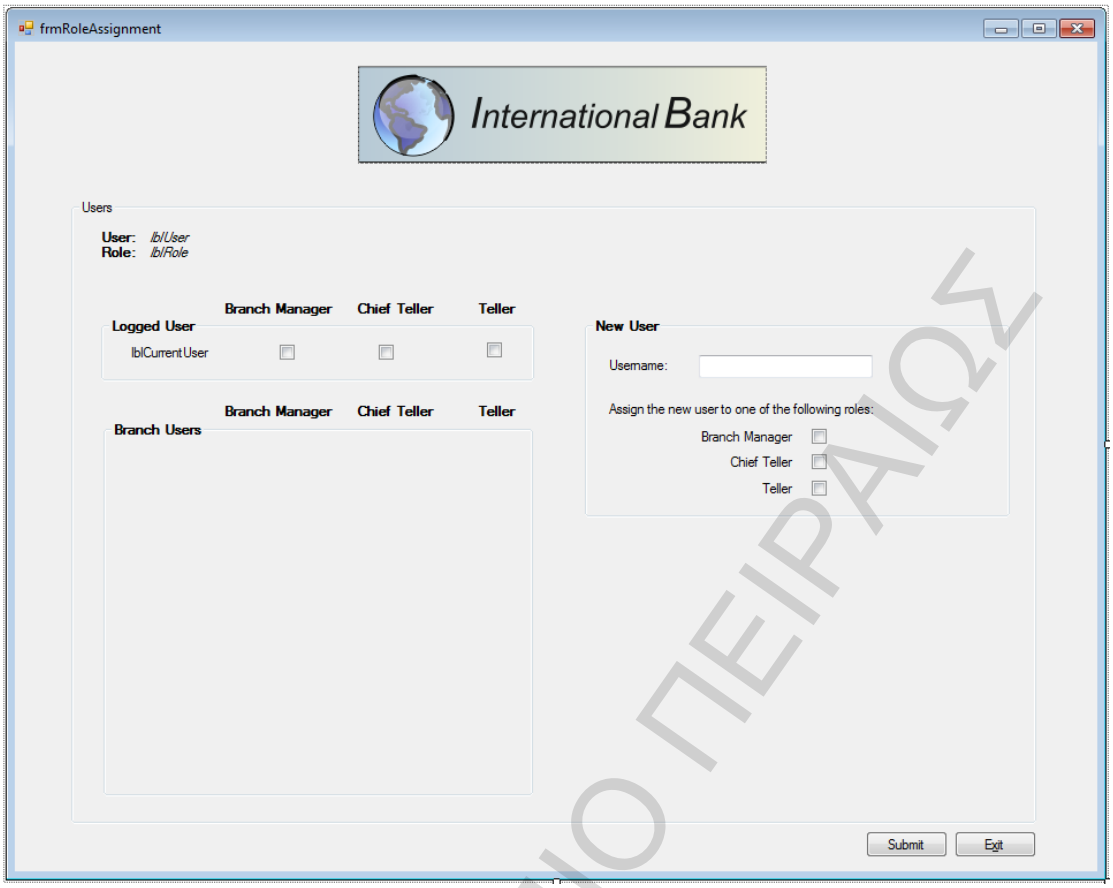
```
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;


namespace BankApplication
{
    //Φόρμα frmCredtiTransaction. Υλοποιεί τη λειτουργία 5 του
Authorization Store.
    //Στη λειτουργία αυτή ο χρήστης μπορεί να υλοποιήσει πιστοτικές
συναλλαγές με
    //μέγιστο όριο τις 20.000 μονάδες.

    public partial class frmCreditTransaction : Form
    {
        public frmCreditTransaction()
        {
            InitializeComponent();
        }

        private void btnSubmit_Click(object sender, EventArgs e)
        {
            this.lblMessage1.Visible = false;
            this.lblMessage2.Visible = false;

            if (CheckRequiredValues())
            {
                bool submit = (CheckAmount());
                if (submit)
                {
                    this.lblMessage1.Visible = true;
                    lblMessage1.Text    =    "Transaction    Succesfully
Submitted";
                }
                else
                {
                    this.lblMessage2.Visible = true;
                    lblMessage2.Text = "Transaction not Valid. Amount
out of range!";
                }
            }

        }

        private bool CheckAmount() {

            bool check = false;
            int amount;


            amount = Convert.ToInt32(txtBxAmount.Text);
            amount = int.Parse(txtBxAmount.Text);


            if (amount <= 20000)
            {
                check = true;
                return check;
            }
            else return check;
```

```
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private      void      txtBxAmount_KeyPress(object      sender,
KeyPressEventArgs e)
        {
            string original = (sender as TextBox).Text;
            if (!char.IsDigit(e.KeyChar))
            {
                e.Handled = true;
            }
            if (e.KeyChar == '.')
            {
                if (original.Contains('.'))
                    e.Handled = true;
                else if (!(original.Contains('.')))
                    e.Handled = false;

            }
            else if (char.IsDigit(e.KeyChar) || e.KeyChar == '\b')
            {
                e.Handled = false;
            }
        }

        private void txtBxAmount_TextChanged(object sender, EventArgs
e)
        {

        }

        private    void    txtBxDebitAccount_KeyPress(object    sender,
KeyPressEventArgs e)
        {
            string original = (sender as TextBox).Text;
            if (!char.IsDigit(e.KeyChar))
            {
                e.Handled = true;
            }
            if (e.KeyChar == '.')
            {
                if (original.Contains('.'))
                    e.Handled = true;
                else if (!(original.Contains('.')))
                    e.Handled = false;

            }
            else if (char.IsDigit(e.KeyChar) || e.KeyChar == '\b')
            {
                e.Handled = false;
            }

        }

        private    void    txtBxCreditAccount_KeyPress(object    sender,
KeyPressEventArgs e)
```

```
        {
            string original = (sender as TextBox).Text;
            if (!char.IsDigit(e.KeyChar))
            {
                e.Handled = true;
            }
            if (e.KeyChar == '.')
            {
                if (original.Contains('.'))
                    e.Handled = true;
                else if (!(original.Contains('.')))
                    e.Handled = false;

            }
            else if (char.IsDigit(e.KeyChar) || e.KeyChar == '\b')
            {
                e.Handled = false;
            }

        }

        private bool CheckRequiredValues()
        {
            bool RequirementsCheck = false;

            if
((string.IsNullOrEmpty(txtBxAmount.Text))||(string.IsNullOrEmpty(txtB
xDebitAccount.Text))||(string.IsNullOrEmpty(txtBxCreditAccount.Text))
||(txtBxCreditAccount.Text.Length                                    <
10)||(txtBxDebitAccount.Text.Length<10)){
                lblMessage2.Visible = true;
                lblMessage2.Text  =  "Please  fill  all  the  required
fields";
                return RequirementsCheck;
            }
            else return RequirementsCheck = true;



        }
    }
}
```

frmRoleAssignment:

[Design]:



frmRoleAssignment.Designer.cs:

```
namespace BankApplication
{
    partial class frmRoleAssignment
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should
be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code
```

```
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.btnExit = new System.Windows.Forms.Button();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.label1 = new System.Windows.Forms.Label();
            this.groupBox3 = new System.Windows.Forms.GroupBox();
            this.label2 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.lblTeller = new System.Windows.Forms.Label();
            this.lblChiefTeller = new System.Windows.Forms.Label();
            this.lblBranchManager = new System.Windows.Forms.Label();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.lblCurrentUser = new System.Windows.Forms.Label();
            this.chcBxTeller = new System.Windows.Forms.CheckBox();
            this.chcBxChiefTeller                    =                    new
System.Windows.Forms.CheckBox();
            this.chcBxBranchManager                  =                    new
System.Windows.Forms.CheckBox();
            this.lblUser = new System.Windows.Forms.Label();
            this.lblRole = new System.Windows.Forms.Label();
            this.lbltxtRole = new System.Windows.Forms.Label();
            this.lbltxtUser = new System.Windows.Forms.Label();
            this.btnSubmit = new System.Windows.Forms.Button();
            this.groupBox5 = new System.Windows.Forms.GroupBox();
            this.label7 = new System.Windows.Forms.Label();
            this.label8 = new System.Windows.Forms.Label();
            this.label9 = new System.Windows.Forms.Label();
            this.txtBxNewUserName                    =                    new
System.Windows.Forms.TextBox();
            this.lblNewUserName = new System.Windows.Forms.Label();
            this.lblNewUserRole = new System.Windows.Forms.Label();
            this.chcBxNewUserAsTeller                =                    new
System.Windows.Forms.CheckBox();
            this.chcBxNewUserAsChiefTeller           =                    new
System.Windows.Forms.CheckBox();
            this.chcBxNewUserAsBranchManager         =                    new
System.Windows.Forms.CheckBox();
            this.pictureBox1 = new System.Windows.Forms.PictureBox();
            this.groupBox1.SuspendLayout();
            this.groupBox2.SuspendLayout();
            this.groupBox5.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginI
nit();
            this.SuspendLayout();
            //
            // btnExit
            //
            this.btnExit.DialogResult                                    =
System.Windows.Forms.DialogResult.Cancel;
            this.btnExit.Location   =   new   System.Drawing.Point(816,
685);
            this.btnExit.Name = "btnExit";
            this.btnExit.Size = new System.Drawing.Size(71, 23);
            this.btnExit.TabIndex = 3;
            this.btnExit.Text = "E&xit";
            this.btnExit.UseVisualStyleBackColor = true;
```

```
            this.btnExit.Click                  +=                  new
System.EventHandler(this.btnExit_Click);
            //
            // groupBox1
            //
            this.groupBox1.Controls.Add(this.groupBox5);
            this.groupBox1.Controls.Add(this.label1);
            this.groupBox1.Controls.Add(this.groupBox3);
            this.groupBox1.Controls.Add(this.label2);
            this.groupBox1.Controls.Add(this.label3);
            this.groupBox1.Controls.Add(this.lblTeller);
            this.groupBox1.Controls.Add(this.lblChiefTeller);
            this.groupBox1.Controls.Add(this.lblBranchManager);
            this.groupBox1.Controls.Add(this.groupBox2);
            this.groupBox1.Controls.Add(this.lblUser);
            this.groupBox1.Controls.Add(this.lblRole);
            this.groupBox1.Controls.Add(this.lbltxtRole);
            this.groupBox1.Controls.Add(this.lbltxtUser);
            this.groupBox1.Location  =  new  System.Drawing.Point(49,
137);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(838, 542);
            this.groupBox1.TabIndex = 4;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Users";
            this.groupBox1.Enter                  +=                  new
System.EventHandler(this.groupBox1_Enter);
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Font  =  new  System.Drawing.Font("Microsoft
Sans       Serif",       8.25F,       System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.label1.Location  =  new  System.Drawing.Point(350,
177);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(39, 13);
            this.label1.TabIndex = 12;
            this.label1.Text = "Teller";
            //
            // groupBox3
            //
            this.groupBox3.Font = new System.Drawing.Font("Microsoft
Sans       Serif",       8.25F,       System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.groupBox3.Location  =  new  System.Drawing.Point(28,
193);
            this.groupBox3.Name = "groupBox3";
            this.groupBox3.Size = new System.Drawing.Size(373, 325);
            this.groupBox3.TabIndex = 9;
            this.groupBox3.TabStop = false;
            this.groupBox3.Text = "Branch Users";
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Font  =  new  System.Drawing.Font("Microsoft
Sans       Serif",       8.25F,       System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
```

```
            this.label2.Location   =   new   System.Drawing.Point(246,
177);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(72, 13);
            this.label2.TabIndex = 11;
            this.label2.Text = "Chief Teller";
            //
            // label3
            //
            this.label3.AutoSize = true;
            this.label3.Font   =   new   System.Drawing.Font("Microsoft
Sans       Serif",       8.25F,       System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.label3.Location   =   new   System.Drawing.Point(130,
177);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(100, 13);
            this.label3.TabIndex = 10;
            this.label3.Text = "Branch Manager";
            //
            // lblTeller
            //
            this.lblTeller.AutoSize = true;
            this.lblTeller.Font = new System.Drawing.Font("Microsoft
Sans       Serif",       8.25F,       System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblTeller.Location = new System.Drawing.Point(350,
88);
            this.lblTeller.Name = "lblTeller";
            this.lblTeller.Size = new System.Drawing.Size(39, 13);
            this.lblTeller.TabIndex = 8;
            this.lblTeller.Text = "Teller";
            //
            // lblChiefTeller
            //
            this.lblChiefTeller.AutoSize = true;
            this.lblChiefTeller.Font                =                new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblChiefTeller.Location            =            new
System.Drawing.Point(246, 88);
            this.lblChiefTeller.Name = "lblChiefTeller";
            this.lblChiefTeller.Size   =   new   System.Drawing.Size(72,
13);
            this.lblChiefTeller.TabIndex = 7;
            this.lblChiefTeller.Text = "Chief Teller";
            //
            // lblBranchManager
            //
            this.lblBranchManager.AutoSize = true;
            this.lblBranchManager.Font              =              new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
System.Drawing.FontStyle.Bold,     System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblBranchManager.Location          =          new
System.Drawing.Point(130, 88);
            this.lblBranchManager.Name = "lblBranchManager";
            this.lblBranchManager.Size = new System.Drawing.Size(100,
13);
            this.lblBranchManager.TabIndex = 6;
```

```
            this.lblBranchManager.Text = "Branch Manager";
            //
            // groupBox2
            //
            this.groupBox2.Controls.Add(this.lblCurrentUser);
            this.groupBox2.Controls.Add(this.chcBxTeller);
            this.groupBox2.Controls.Add(this.chcBxChiefTeller);
            this.groupBox2.Controls.Add(this.chcBxBranchManager);
            this.groupBox2.Font = new System.Drawing.Font("Microsoft
Sans       Serif",       8.25F,       System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.groupBox2.Location = new System.Drawing.Point(26,
103);
            this.groupBox2.Name = "groupBox2";
            this.groupBox2.Size = new System.Drawing.Size(376, 55);
            this.groupBox2.TabIndex = 5;
            this.groupBox2.TabStop = false;
            this.groupBox2.Text = "Logged User";
            this.groupBox2.Enter                    +=                  new
System.EventHandler(this.groupBox2_Enter);
            //
            // lblCurrentUser
            //
            this.lblCurrentUser.AutoSize = true;
            this.lblCurrentUser.Font                =                   new
System.Drawing.Font("Microsoft       Sans       Serif",       8.25F,
System.Drawing.FontStyle.Regular,  System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblCurrentUser.Location            =                   new
System.Drawing.Point(24, 23);
            this.lblCurrentUser.Name = "lblCurrentUser";
            this.lblCurrentUser.Size = new System.Drawing.Size(73,
13);
            this.lblCurrentUser.TabIndex = 5;
            this.lblCurrentUser.Text = "lblCurrentUser";
            //
            // chcBxTeller
            //
            this.chcBxTeller.AutoSize = true;
            this.chcBxTeller.Location = new System.Drawing.Point(335,
21);
            this.chcBxTeller.Name = "chcBxTeller";
            this.chcBxTeller.Size = new System.Drawing.Size(15, 14);
            this.chcBxTeller.TabIndex = 2;
            this.chcBxTeller.UseVisualStyleBackColor = true;
            //
            // chcBxChiefTeller
            //
            this.chcBxChiefTeller.AutoSize = true;
            this.chcBxChiefTeller.Location          =                   new
System.Drawing.Point(241, 23);
            this.chcBxChiefTeller.Name = "chcBxChiefTeller";
            this.chcBxChiefTeller.Size = new System.Drawing.Size(15,
14);
            this.chcBxChiefTeller.TabIndex = 1;
            this.chcBxChiefTeller.UseVisualStyleBackColor = true;
            //
            // chcBxBranchManager
            //
            this.chcBxBranchManager.AutoSize = true;
```

```
            this.chcBxBranchManager.Location = new
System.Drawing.Point(155, 23);
            this.chcBxBranchManager.Name = "chcBxBranchManager";
            this.chcBxBranchManager.Size = new
System.Drawing.Size(15, 14);
            this.chcBxBranchManager.TabIndex = 0;
            this.chcBxBranchManager.UseVisualStyleBackColor = true;
            //
            // lblUser
            //
            this.lblUser.AutoSize = true;
            this.lblUser.Font = new System.Drawing.Font("Microsoft
Sans Serif", 8.25F, System.Drawing.FontStyle.Italic,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblUser.Location = new System.Drawing.Point(66, 26);
            this.lblUser.Name = "lblUser";
            this.lblUser.Size = new System.Drawing.Size(39, 13);
            this.lblUser.TabIndex = 4;
            this.lblUser.Text = "lblUser";
            //
            // lblRole
            //
            this.lblRole.AutoSize = true;
            this.lblRole.Font = new System.Drawing.Font("Microsoft
Sans Serif", 8.25F, System.Drawing.FontStyle.Italic,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lblRole.Location = new System.Drawing.Point(66, 39);
            this.lblRole.Name = "lblRole";
            this.lblRole.Size = new System.Drawing.Size(39, 13);
            this.lblRole.TabIndex = 3;
            this.lblRole.Text = "lblRole";
            //
            // lbltxtRole
            //
            this.lbltxtRole.AutoSize = true;
            this.lbltxtRole.Font = new System.Drawing.Font("Microsoft
Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lbltxtRole.Location = new System.Drawing.Point(23,
39);
            this.lbltxtRole.Name = "lbltxtRole";
            this.lbltxtRole.Size = new System.Drawing.Size(37, 13);
            this.lbltxtRole.TabIndex = 1;
            this.lbltxtRole.Text = "Role:";
            this.lbltxtRole.Click += new
System.EventHandler(this.lblRole_Click);
            //
            // lbltxtUser
            //
            this.lbltxtUser.AutoSize = true;
            this.lbltxtUser.Font = new System.Drawing.Font("Microsoft
Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.lbltxtUser.Location = new System.Drawing.Point(23,
26);
            this.lbltxtUser.Name = "lbltxtUser";
            this.lbltxtUser.Size = new System.Drawing.Size(37, 13);
            this.lbltxtUser.TabIndex = 0;
            this.lbltxtUser.Text = "User:";
            //
            // btnSubmit
```

```
            //
            this.btnSubmit.Location = new System.Drawing.Point(739,
685);
            this.btnSubmit.Name = "btnSubmit";
            this.btnSubmit.Size = new System.Drawing.Size(71, 23);
            this.btnSubmit.TabIndex = 5;
            this.btnSubmit.Text = "Submit";
            this.btnSubmit.UseVisualStyleBackColor = true;
            this.btnSubmit.Click                    +=                  new
System.EventHandler(this.btnSubmit_Click);
            //
            // groupBox5
            //
            this.groupBox5.Controls.Add(this.chcBxNewUserAsTeller);
            this.groupBox5.Controls.Add(this.lblNewUserRole);

this.groupBox5.Controls.Add(this.chcBxNewUserAsChiefTeller);
            this.groupBox5.Controls.Add(this.label7);

this.groupBox5.Controls.Add(this.chcBxNewUserAsBranchManager);
            this.groupBox5.Controls.Add(this.label8);
            this.groupBox5.Controls.Add(this.lblNewUserName);
            this.groupBox5.Controls.Add(this.label9);
            this.groupBox5.Controls.Add(this.txtBxNewUserName);
            this.groupBox5.Font = new System.Drawing.Font("Microsoft
Sans       Serif",       8.25F,       System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.groupBox5.Location = new System.Drawing.Point(446,
103);
            this.groupBox5.Name = "groupBox5";
            this.groupBox5.Size = new System.Drawing.Size(369, 173);
            this.groupBox5.TabIndex = 13;
            this.groupBox5.TabStop = false;
            this.groupBox5.Text = "New User";
            //
            // label7
            //
            this.label7.AutoSize = true;
            this.label7.Font = new System.Drawing.Font("Microsoft
Sans      Serif",      8.25F,      System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.label7.Location = new System.Drawing.Point(151,
141);
            this.label7.Name = "label7";
            this.label7.Size = new System.Drawing.Size(33, 13);
            this.label7.TabIndex = 16;
            this.label7.Text = "Teller";
            //
            // label8
            //
            this.label8.AutoSize = true;
            this.label8.Font = new System.Drawing.Font("Microsoft
Sans      Serif",      8.25F,      System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.label8.Location = new System.Drawing.Point(124,
118);
            this.label8.Name = "label8";
            this.label8.Size = new System.Drawing.Size(60, 13);
            this.label8.TabIndex = 15;
            this.label8.Text = "Chief Teller";
            //
```

```
            // label9
            //
            this.label9.AutoSize = true;
            this.label9.Font = new System.Drawing.Font("Microsoft
Sans Serif", 8.25F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(161)));
            this.label9.Location = new System.Drawing.Point(98, 96);
            this.label9.Name = "label9";
            this.label9.Size = new System.Drawing.Size(86, 13);
            this.label9.TabIndex = 14;
            this.label9.Text = "Branch Manager";
            //
            // txtBxNewUserName
            //
            this.txtBxNewUserName.Location = new
System.Drawing.Point(99, 32);
            this.txtBxNewUserName.Name = "txtBxNewUserName";
            this.txtBxNewUserName.Size = new System.Drawing.Size(151,
20);
            this.txtBxNewUserName.TabIndex = 0;
            //
            // lblNewUserName
            //
            this.lblNewUserName.AutoSize = true;
            this.lblNewUserName.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblNewUserName.Location = new
System.Drawing.Point(18, 34);
            this.lblNewUserName.Name = "lblNewUserName";
            this.lblNewUserName.Size = new System.Drawing.Size(58,
13);
            this.lblNewUserName.TabIndex = 1;
            this.lblNewUserName.Text = "Username:";
            //
            // lblNewUserRole
            //
            this.lblNewUserRole.AutoSize = true;
            this.lblNewUserRole.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(161)));
            this.lblNewUserRole.Location = new
System.Drawing.Point(18, 72);
            this.lblNewUserRole.Name = "lblNewUserRole";
            this.lblNewUserRole.Size = new System.Drawing.Size(237,
13);
            this.lblNewUserRole.TabIndex = 17;
            this.lblNewUserRole.Text = "Assign the new user to one of
the following roles:";
            //
            // chcBxNewUserAsTeller
            //
            this.chcBxNewUserAsTeller.AutoSize = true;
            this.chcBxNewUserAsTeller.Location = new
System.Drawing.Point(197, 141);
            this.chcBxNewUserAsTeller.Name = "chcBxNewUserAsTeller";
            this.chcBxNewUserAsTeller.Size = new
System.Drawing.Size(15, 14);
            this.chcBxNewUserAsTeller.TabIndex = 8;
```

```
            this.chcBxNewUserAsTeller.UseVisualStyleBackColor = true;
            //
            // chcBxNewUserAsChiefTeller
            //
            this.chcBxNewUserAsChiefTeller.AutoSize = true;
            this.chcBxNewUserAsChiefTeller.Location       =       new
System.Drawing.Point(197, 118);
            this.chcBxNewUserAsChiefTeller.Name                       =
"chcBxNewUserAsChiefTeller";
            this.chcBxNewUserAsChiefTeller.Size         =       new
System.Drawing.Size(15, 14);
            this.chcBxNewUserAsChiefTeller.TabIndex = 7;
            this.chcBxNewUserAsChiefTeller.UseVisualStyleBackColor  =
true;
            //
            // chcBxNewUserAsBranchManager
            //
            this.chcBxNewUserAsBranchManager.AutoSize = true;
            this.chcBxNewUserAsBranchManager.Location      =      new
System.Drawing.Point(197, 96);
            this.chcBxNewUserAsBranchManager.Name                     =
"chcBxNewUserAsBranchManager";
            this.chcBxNewUserAsBranchManager.Size        =        new
System.Drawing.Size(15, 14);
            this.chcBxNewUserAsBranchManager.TabIndex = 6;
            this.chcBxNewUserAsBranchManager.UseVisualStyleBackColor
= true;
            //
            // pictureBox1
            //
            this.pictureBox1.Image                                    =
global::BankApplication.Properties.Resources.img1;
            this.pictureBox1.InitialImage                             =
global::BankApplication.Properties.Resources.img1;
            this.pictureBox1.Location = new System.Drawing.Point(298,
21);
            this.pictureBox1.Name = "pictureBox1";
            this.pictureBox1.Size = new System.Drawing.Size(355, 85);
            this.pictureBox1.TabIndex = 0;
            this.pictureBox1.TabStop = false;
            //
            // frmRoleAssignment
            //
            this.AutoScaleDimensions  =  new  System.Drawing.SizeF(6F,
13F);
            this.AutoScaleMode                                        =
System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(940, 720);
            this.Controls.Add(this.btnSubmit);
            this.Controls.Add(this.groupBox1);
            this.Controls.Add(this.btnExit);
            this.Controls.Add(this.pictureBox1);
            this.Name = "frmRoleAssignment";
            this.Text = "frmRoleAssignment";
            this.Load                       +=                        new
System.EventHandler(this.frmRoleAssignment_Load);
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            this.groupBox2.ResumeLayout(false);
            this.groupBox2.PerformLayout();
            this.groupBox5.ResumeLayout(false);
```

```
            this.groupBox5.PerformLayout();

((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndIni
t();
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.PictureBox pictureBox1;
        internal System.Windows.Forms.Button btnExit;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.Label lbltxtRole;
        private System.Windows.Forms.Label lbltxtUser;
        private System.Windows.Forms.Label lblRole;
        private System.Windows.Forms.Label lblUser;
        private System.Windows.Forms.GroupBox groupBox2;
        private System.Windows.Forms.Label lblTeller;
        private System.Windows.Forms.Label lblChiefTeller;
        private System.Windows.Forms.Label lblBranchManager;
        private System.Windows.Forms.CheckBox chcBxTeller;
        private System.Windows.Forms.CheckBox chcBxChiefTeller;
        private System.Windows.Forms.CheckBox chcBxBranchManager;
        private System.Windows.Forms.Label lblCurrentUser;
        private System.Windows.Forms.GroupBox groupBox3;
        private System.Windows.Forms.Button btnSubmit;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.Label label9;
        private System.Windows.Forms.GroupBox groupBox5;
        private System.Windows.Forms.Label lblNewUserName;
        private System.Windows.Forms.TextBox txtBxNewUserName;
        private System.Windows.Forms.Label lblNewUserRole;
        private System.Windows.Forms.CheckBox chcBxNewUserAsTeller;
        private                          System.Windows.Forms.CheckBox
chcBxNewUserAsChiefTeller;
        private                          System.Windows.Forms.CheckBox
chcBxNewUserAsBranchManager;
    }
}
```

frmRoleAssignment.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using AZROLESLib;
```

```
using System.Security.Principal;
using System.Threading;
using System.Text.RegularExpressions;
using System.Security;
using System.DirectoryServices.AccountManagement;
using System.DirectoryServices;
using System.Web.Security;

namespace BankApplication
{
    //Φόρμα frmRoleAssignment. Αντιστοιχεί στο operation 13, Role
Assignment.
    //Παρέχει τη δυνατότητα στον Branch Manager να δειαχειριστεί τους
υπάρχοντες
    //χρήστες του Authorization Store, καθώς επίσης και δυνατότητα
δημιουργίας
    //νέου χρήστη στο Windows Active Directory και αντιστοιχία αυτού
σε έναν
    //από τους τρεις ρόλους του Authorization Store.
    public partial class frmRoleAssignment : Form
    {
        public frmRoleAssignment()
        {
            InitializeComponent();

            userName = Program.LoggedUser();


        }

        string userName;
        int empNum = 0;

        private void frmRoleAssignment_Load(object sender, EventArgs
e)
        {
            EstablishWindowsIdentity();

        }


        private void EstablishWindowsIdentity() {

            WindowsIdentity                userIdentity                =
WindowsIdentity.GetCurrent();
            this.lblUser.Text = Program.LoggedUser();
            string rtnRole = AzmanRoleCheck(userIdentity.Name);
            this.lblRole.Text = rtnRole;
            this.lblCurrentUser.Text = Program.LoggedUser();

            if (rtnRole == "Branch Manager") {
                this.chcBxBranchManager.Checked = true ;
            }
            else if (rtnRole == "Chief Teller")
            {
                this.chcBxChiefTeller.Checked = true;
            }
            else if (rtnRole == "Teller")
            {
                this.chcBxTeller.Checked = true;
            }
```

```
            else
            {
                MessageBox.Show(String.Format("We are sorry {0}, but
your are not authorized for this application!", userIdentity.Name,
MessageBoxButtons.OK));
                this.Close();
            }
            usersCheck(userIdentity.Name);


        }

        //Συνάρτηση AzmanRoleCheck που ελέγχει σε ποιό ρόλο ανήκει ο
logged user
        //μέσω του xml στο Authorization Store. Επιστρέφει σε string
τον ρόλο.
        private string AzmanRoleCheck(string strUserName)
        {

            string strRoleName = string.Empty;
            try
            {
                AzAuthorizationStore        AzmanStore     =     new
AzAuthorizationStore();

                AzmanStore.Initialize(0,
@"msxml://C:\Users\Vasilis\Documents\Visual                    Studio
2008\Projects\BankApplication\BankApplication\BankApplication.xml",
null);
                IAzApplication              AzManApp              =
AzmanStore.OpenApplication("Bank Application", null);

                WindowsIdentity             userIdentity          =
WindowsIdentity.GetCurrent();
                IAzClientContext                ctx               =
AzManApp.InitializeClientContextFromToken((ulong)userIdentity.Token,
null);



                foreach (IAzRole Azrole in AzManApp.Roles)
                {
                    strRoleName = Azrole.Name;
                    foreach         (object         member         in
(object[])Azrole.MembersName)
                    {
                        string strMemberName = member.ToString();
                        string[]  str1  =  Regex.Split(strMemberName,
"@");
                        if    (string.Equals(str1[0],    strUserName,
StringComparison.CurrentCultureIgnoreCase))
                            return strRoleName;
                        else {
                            strRoleName = "Unauthorized";
                            return strRoleName;
                        }
                    }
                }
            }
            catch (Exception)
            {
```

```
                throw;
            }
            return strRoleName;


        }

        //Συνάρτηση usersCheck. Ελέγχει το ρόλο του κάθε χρήστη που έχει δημιουργηθεί
        //στο Acctive Directory των Windows και παράλληλα ανήκει στο Azman Store.
        //Παράλληλα δημιουργεί στη φόρμα στο groupBox Branch Users τη λίστα με όλους
        //τους χρήστες και τον αντίστοιχο ρόλο του σε checkBoxes.
        private void usersCheck(string strUserName)
        {

            string strRoleName = string.Empty;
            try
            {
                AzAuthorizationStore AzmanStore = new AzAuthorizationStore();

                AzmanStore.Initialize(0, @"msxml://C:\Users\Vasilis\Documents\Visual Studio 2008\Projects\BankApplication\BankApplication\BankApplication.xml", null);
                IAzApplication AzManApp = AzmanStore.OpenApplication("Bank Application", null);

                WindowsIdentity userIdentity = WindowsIdentity.GetCurrent();
                IAzClientContext ctx = AzManApp.InitializeClientContextFromToken((ulong)userIdentity.Token, null);


                int i = 0;
                int j = 1;

                foreach (IAzRole Azrole in AzManApp.Roles)
                {

                    strRoleName = Azrole.Name;

                    foreach (object member in (object[])Azrole.MembersName)
                    {
                        string strMemberName = member.ToString();
                        strMemberName= strMemberName.Substring(strMemberName.IndexOf('\\') + 1);


                        Label[] lblUsers = new Label[10];
                        lblUsers[i] = new Label();
                        lblUsers[i].Name = string.Concat("strMemberName",i);
                        lblUsers[i].Text = strMemberName;
                        lblUsers[i].Visible = true;
                        lblUsers[i].AutoSize = true;
```

```
                        lblUsers[i].Location          =          new
System.Drawing.Point(22, 30+(i*20));
                        lblUsers[i].Show();

                        CheckBox[] chcBxRoles = new CheckBox[4];
                        chcBxRoles[1] = new CheckBox();
                        chcBxRoles[1].Name                          =
string.Concat("chcBoxTeller",i);
                        chcBxRoles[1].Visible = true;
                        chcBxRoles[1].Location          =          new
System.Drawing.Point(333, 28+(i*20));
                        chcBxRoles[1].Show();

                        chcBxRoles[2] = new CheckBox();
                        chcBxRoles[2].Name                          =
string.Concat("chcBoxChiefTeller",i);
                        chcBxRoles[2].Visible = true;
                        chcBxRoles[2].Location          =          new
System.Drawing.Point(239, 28 + (i * 20));
                        chcBxRoles[2].Show();

                        chcBxRoles[3] = new CheckBox();
                        chcBxRoles[3].Name                          =
string.Concat("chcBoxBranchManager", i);
                        chcBxRoles[3].Visible = true;
                        chcBxRoles[3].Location          =          new
System.Drawing.Point(153, 28 + (i * 20));
                        chcBxRoles[3].Show();

                        this.groupBox3.Controls.Add(chcBxRoles[1]);
                        this.groupBox3.Controls.Add(chcBxRoles[2]);
                        this.groupBox3.Controls.Add(chcBxRoles[3]);
                        this.groupBox3.Controls.Add(lblUsers[i]);

                        if (strRoleName == "Branch Manager")
                        {
                            chcBxRoles[3].Checked = true;
                        }
                        else if (strRoleName == "Chief Teller")
                        {
                            chcBxRoles[2].Checked = true;
                        }
                        else
                        {
                            chcBxRoles[1].Checked = true;
                        }

                        i++;
                        j++;

                    }
                }
                empNum = i;
                MessageBox.Show(string.Format("Number  of  employees:
{0}", empNum, MessageBoxButtons.OK));
            }
            catch (Exception)
            {
                throw;
            }
```

```
        }


        public static string LoggedUser()
        {
            string UserName = WindowsIdentity.GetCurrent().Name;
            return UserName.Substring(UserName.IndexOf('\\') + 1);
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void groupBox1_Enter(object sender, EventArgs e)
        {

        }

        private void lblRole_Click(object sender, EventArgs e)
        {

        }

        private void groupBox2_Enter(object sender, EventArgs e)
        {

        }

        private void btnSubmit_Click(object sender, EventArgs e)
        {
            //CheckOnSubmitRoles();
            bool WeHaveANewUser = false;
            WeHaveANewUser = CheckForNewUser();
            if (WeHaveANewUser)
            {
                bool                          test                          =
CreateNewUser(this.txtBxNewUserName.Text);
                if (test)
                {

                    AssignUserToRole();
                    MessageBox.Show(string.Format("New  user:  {0}  is
added!", this.txtBxNewUserName.Text, MessageBoxButtons.OK));
                }
                else {


                }
            }

            this.Close();

        }

        private    void    checkedListBox1_SelectedIndexChanged(object
sender, EventArgs e)
        {

        }
```

```
        private void CheckOnSubmitRoles()
        {
            //check for role changes
        }

        private bool CheckForNewUser() {

            bool newuser = false;

            if (this.txtBxNewUserName.Text != "")
            {
                newuser = true;
                return newuser;

            }
            else {
                return newuser;
            }

        }

        //Συνάρτηση CreateNewUser. Δημιουργεί μέσω του textBox στο
groupBox
        //New User νέο χρήστη στο Active Directory των Windows όταν
αυτός δεν
        //υπάρχει.
        public bool CreateNewUser(string sUserName)
        {
            bool newUser = false;

            if (userexists(sUserName) == false)
            {

                PrincipalContext          pc          =          new
PrincipalContext(ContextType.Machine);

System.DirectoryServices.AccountManagement.UserPrincipal   u   =   new
UserPrincipal(pc);

                string password = "";
                string description = "Test user for AzMan";

                u.SetPassword(password);
                u.Name = sUserName;
                u.Description = description;
                u.UserCannotChangePassword = true;
                u.PasswordNeverExpires = true;
                u.Save();


                GroupPrincipal                  gPc                  =
GroupPrincipal.FindByIdentity(pc, "Users");
                gPc.Members.Add(u);
                gPc.Save();

                newUser = true;
                return newUser;
            }
            else
            {
```

```
                MessageBox.Show(string.Format("User          Exists",
MessageBoxButtons.OK));
                return newUser;

            }
         }


        static bool userexists(string strUserName)
        {
            string     adsPath     =     string.Format(@"WinNT://{0}",
System.Environment.MachineName);
            using (DirectoryEntry de = new DirectoryEntry(adsPath))
            {
                try
                {
                    return de.Children.Find(strUserName) != null;
                }
                catch (Exception e)
                {
                    return false;
                }
            }
        }

        //Συνάρτηση AssignUserToRole. Κάνει χρήση του νέου χρήστη
μέσω της συνάρτησης CreateNewUser
        //και τον κατατάσει στον αντίστοιχο ρόλο στο Authorization
Store, ανάλογα με το checkBox που
        //έχει επιλέξει ο Branch Manager.
        private void AssignUserToRole()
        {

            AzAuthorizationStore        AzmanStore        =        new
AzAuthorizationStore();

            AzmanStore.Initialize(0,
@"msxml://C:\Users\Vasilis\Documents\Visual                    Studio
2008\Projects\BankApplication\BankApplication\BankApplication.xml",
null);
            IAzApplication                  AzManApp                =
AzmanStore.OpenApplication("Bank Application", null);

            WindowsIdentity               userIdentity              =
WindowsIdentity.GetCurrent();
            IAzClientContext                    ctx                 =
AzManApp.InitializeClientContextFromToken((ulong)userIdentity.Token,
null);

            NTAccount f = new NTAccount(txtBxNewUserName.Text);
            SecurityIdentifier                  s                   =
(SecurityIdentifier)f.Translate(typeof(SecurityIdentifier));
            String sidString = s.ToString();

            if (chcBxNewUserAsBranchManager.Checked == true){

            IAzRole  azRole  =  AzManApp.OpenRole("Branch  Manager",
null);

            azRole.AddMember(sidString, null);
            azRole.Submit(0, null);
```

```
            AzmanStore.UpdateCache(null);

            }
            else if (chcBxNewUserAsChiefTeller.Checked == true){

                IAzRole  azRole  =  AzManApp.OpenRole("Chief  Teller",
null);

                azRole.AddMember(sidString, null);
                azRole.Submit(0, null);
                AzmanStore.UpdateCache(null);

            }
            else if (chcBxNewUserAsTeller.Checked == true){

                IAzRole azRole = AzManApp.OpenRole("Teller", null);

                azRole.AddMember(sidString, null);
                azRole.Submit(0, null);
                AzmanStore.UpdateCache(null);

            }

            else {

                MessageBox.Show(string.Format("Please  select  role",
MessageBoxButtons.OK));

            }
        }

        private void lblUser1_Click(object sender, EventArgs e)
        {

        }

        private void lblUser2_Click(object sender, EventArgs e)
        {

        }

        private void lblUser7_Click(object sender, EventArgs e)
        {

        }

        private void groupBox4_Enter(object sender, EventArgs e)
        {

        }

        private    void    checkBox24_CheckedChanged(object    sender,
EventArgs e)
        {

        }

        private    void    checkBox17_CheckedChanged(object    sender,
EventArgs e)
        {
```

```
            }
        }
}
```