Ατομική Διπλωματική Εργασία

ΑΝΙΧΝΕΥΣΗ ΚΑΚΟΒΟΥΛΩΝ ΕΦΑΡΜΟΓΩΝ ΣΕ ΥΠΟΛΟΓΙΣΤΕΣ ΚΑΙ ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ

Ρέα Αρχαίου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΙΧΝΕΥΣΗ ΚΑΚΟΒΟΥΛΩΝ ΕΦΑΡΜΟΓΩΝ ΣΕ ΥΠΟΛΟΓΙΣΤΕΣ ΚΑΙ ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ

Ρέα Αρχαίου

Επιβλέπων Καθηγητής
Βάσος Βασιλείου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2010

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Δρ. Βάσο Βασιλείου, για την καθοδήγηση και την υποστήριξη που μου προσέφερε κατά τη διάρκεια της υλοποίησης της παρούσας εργασίας. Θερμές ευχαριστίες πρέπει να δοθούν και στη διδακτορική φοιτήτρια του τμήματος, Χριστιάνα Ιωάννου, για την πολύτιμη βοήθεια και καθοδήγηση που μου έχει προσφέρει για την επίτευξη της συγκεκριμένης εργασίας. Εύχομαι και στους δύο κάθε επιτυχία και ευτυχία τόσο σε επαγγελματικό, όσο και οικογενειακό επίπεδο. Επίσης ευχαριστίες πρέπει να δοθούν και στην ερευνητική ομάδα του εργαλείου Anubis, που μου παρείχε τις κακόβουλες εφαρμογές που χρησιμοποιήθηκαν για την εκπόνηση της συγκεκριμένης εργασίας.

Συνεχίζοντας θα ήθελα να ευχαριστήσω την οικογένεια μου για τη συμπαράσταση και κατανόηση που μου έδειχναν τα τελευταία τέσσερα χρόνια, καθώς επίσης και την καλύτερη μου φίλη. Τέλος θα ήθελα να ευχαριστήσω τους συμφοιτητές και συμφοιτήτριες μου για τις ευχάριστες στιγμές που μοιραστήκαμε τα τέσσερα χρόνια των σπουδών μας.

Περίληψη

Ο κακόβουλος κώδικας αποτελεί μια μεγάλη απειλή για τα υπολογιστικά συστήματα. Στις μέρες μας, που η χρησιμοποίηση του διαδικτύου αυξάνεται όλο και περισσότερο και όλο και περισσότερες εφαρμογές δημιουργούνται για την διευκόλυνση της ανταλλαγής δεδομένων, είναι επιτακτική ανάγκη η ύπαρξη κάποιου ισχυρού συστήματος ασφάλειας.

Υπάρχουν διάφορα συστήματα τα οποία παρέχουν προστασία από κακόβουλες εφαρμογές, όμως η λειτουργία τους στηρίζεται στην ανίχνευση κακόβουλου κώδικα, ο οποίος είναι ήδη γνωστός. Αυτά τα συστήματα δεν παρέχουν ασφάλεια σε καινούριες επιθέσεις.

Σκοπός της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός εργαλείου, το οποίο ακολουθώντας την μέθοδο ανίχνευσης ανωμαλιών και με βάση τα systems calls θα μπορεί να ανιχνεύει νέων ειδών επιθέσεις.

Αρχικά παρουσιάζεται η γενική θεωρία που αφορά τα θέματα ασφαλείας σε υπολογιστές και δίκτυα υπολογιστών. Αυτό παρουσιάζεται με βάση τις αρχές ασφάλειας που πρέπει να διέπουν τα συστήματα. Αργότερα παρουσιάζονται τα είδη επιθέσεων, τα είδη κακόβουλων εφαρμογών, αλλά και πώς μπορεί να σύστημα να προστατευτεί. Περισσότερη σημασία δόθηκε στην ανίχνευση εισβολών και συγκεκριμένα στα συστήματα που είναι υλοποιημένα με τη μέθοδο ανίχνευσης ανωμαλιών, όπως και σε μελέτες που έγιναν για τη δημιουργία ενός ισχυρού συστήματος ανίχνευσης εισβολών.

Ο σχεδιασμός και η υλοποίηση του μοντέλου περιλαμβάνει αρχικά τη συλλογή των δεδομένων που θα χρησιμοποιηθούν. Ακολουθεί ο διαχωρισμός της φυσιολογικής συμπεριφοράς μελετώντας τα system calls από την ασυνήθιστη. Για το σκοπό αυτό χρησιμοποιείται το Logistic Regression σε συνδυασμό με το Principal Component Analysis. Με τη χρησιμοποίηση των πιο πάνω μεθόδων δημιουργήθηκαν διαφορετικά εργαλεία ανίχνευσης εισβολών και αξιολογήθηκε η επίδοση τους. Από ότι διαφάνηκε μπορούν να δημιουργηθούν εργαλεία ανίχνευσης ανωμαλιών τα οποία μπορούν να έχουν υψηλό ποσοστό πρόβλεψης και μικρό ποσοστό λάθους.

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή……1

1.1 Καθορισμός προβλήματος…….……...1

Κεφάλαιο 2 Αρχές Ασφάλειας ……..3

Εισαγωγή……3

2.1 Έννοια ασφάλειας…….3

2.2 Αρχές σχεδιασμού συστήματος……6

2.3 Αρχές σχεδιασμού συστημάτων ασφαλείας……8

Κεφάλαιο 3 Κακόβουλες Εφαρμογές…….....11

Εισαγωγή……..11

3.1 Επιθέσεις στο σύστημα…….……11

3.2 Τύποι Εισβολέων…….……..16

3.3 Κακόβουλος Κώδικας…….……...…17

3.3.1 Ιός……..……...19

3.3.2 Worms……..……22

3.3.3 Trojan Horses……..……24

3.3.4 Λογική Βόμβα……..……27

3.3.5 Trapdoor…….…….……27

3.3.6 Zombie……..28

3.3.7 Rabbit……...29

3.4 Τρόποι επισύναψης ιών σε προγράμματα ……...……..29

Κεφάλαιο 4 Τρόποι εξασφάλισης ασφάλειας.……. 31

Εισαγωγή……...31

4.1 Τρόποι Προστασίας……31

4.1.1 Κρυπτογραφία……..32

4.1.2 Εξακρίβωση Αυθεντικότητας……...34

4.1.3 Προστασία Ηλεκτρονικού Ταχυδρομείου……35

4.1.4 Προστασία TCP συνδέσεων……36

4.1.5 Προστασία επιπέδου δικτύου……..37
4.1.6 Ασύρματη Ασφάλεια……..38
4.1.7 Firewalls……..40
4.2 Συστήματα Ανίχνευσης Εισβολής……42
4.3 Παρουσίαση μοντέλου ανίχνευσης εισβολής……...45
4.4 Προηγούμενες μελέτες για ανίχνευση ανωμαλιών……...51
4.4.1 Ασφάλεια συστήματος και Βιολογική Ανοσιολογία……..51
4.4.2 Καταγραφή Συμπεριφοράς προγραμμάτων……54
4.4.3 Εξερχόμενη κίνηση διαδικτύου……..56
4.4.4 Ανίχνευση Ανωμαλιών σε επιθέσεις διαδικτύου……58

Κεφάλαιο 5 Σχεδιασμός και Υλοποίηση Εργαλείου Ασφάλειας……..61
Εισαγωγή……63
5.1 Παρουσίαση προηγούμενης ερευνητικής εργασίας……...64
5.2 Προτεινόμενη Λύση…….……65
5.2.1 Συλλογή Δεδομένων…….…..66
5.2.2 Ανάλυση Δεδομένων…….….69
5.3 Υλοποίηση Εργαλείου Ανίχνευσης…….….72
5.3.1 Υλοποίηση πρώτου εργαλείου ανίχνευσης……..…73
5.3.2 Υλοποίηση δεύτερου εργαλείου ανίχνευσης……..78
5.4 Παρατηρήσεις……...82

Κεφάλαιο 6 Συμπεράσματα και μελλοντική εργασία……...83
6.1 Συμπεράσματα……...83
6.2 Μελλοντική εργασία……85

Βιβλιογραφία…….86
Παράρτημα Α Παρουσίαση προγράμματος WinAPIOverride32……...89
Παράρτημα Β Πρόγραμμα σε JAVA για ανάλυση system calls……...94
Παράρτημα Γ Πρόγραμμα σε MATLAB για το Principal Component Analysis..98

Κεφάλαιο 1

Εισαγωγή

1.1 Προσδιορισμός προβλήματος

1.1 Προσδιορισμός προβλήματος

Η συνεχής και ραγδαία ανάπτυξη των ηλεκτρονικών υπολογιστών φέρνει συνεχώς στην επιφάνεια νέες εφαρμογές και νέες προκλήσεις. Οι νέες εφαρμογές που παρουσιάζονται αφορούν αποθήκευση δεδομένων και ταυτόχρονη χρήση από πολλούς χρήστες. Για τέτοιου είδους εφαρμογές απαιτούνται οι κατάλληλοι μηχανισμοί οι οποίοι θα προστατεύουν το σύστημα από διάφορες κακόβουλες ενέργειες.

Μέχρι σήμερα τα διάφορα προγράμματα που είναι σχεδιασμένα για να προστατεύουν τα συστήματα, παρέχουν μόνο προστασία από κακόβουλα προγράμματα, τα οποία έχουν ήδη αναγνωριστεί. Αυτά τα προγράμματα ασφάλειας ονομάζονται συστήματα ανίχνευσης κατάχρησης και χρησιμοποιούν βάσεις, όπου αποθηκεύουν τις πληροφορίες για τους αναγνωρισμένους ιούς. Έτσι χρειάζεται να ενημερώνονται καθημερινά για τα νέα κακόβουλα προγράμματα που έχουν αναγνωριστεί, ώστε να μπορούν να παρέχουν προστασία.

Κάθε μέρα δημιουργούνται όλο και περισσότεροι ιοί από τους διάφορους εισβολείς, οι οποίοι βρίσκουν συνεχώς αδυναμίες των συστημάτων που μπορούν να τις εκμεταλλευτούν. Η συνεχής δημιουργία νέων κακόβουλων προγραμμάτων μπορεί να προκαλέσουν ζημιά στα συστήματα, αφού μπορεί να μην προλάβουν να καταχωρηθούν στη βάση και να ενημερωθεί το σύστημα ανίχνευσης ώστε να είναι σε θέση να αναγνωρίζει τον κακόβουλος κώδικας που πραγματοποιεί επίθεση σε κάποιο σύστημα. Έτσι η πιο πάνω μέθοδος που ακολουθούσαν τα μέχρι σήμερα προγράμματα ασφάλειας

δεν είναι η πιο αποτελεσματική. Δεν μπορεί να αναγνωρίζει καινούριο κακόβουλο κώδικα.

Για την αντιμετώπιση των πιο πάνω μειονεκτημάτων, μπορεί να χρησιμοποιηθεί η μέθοδος ανίχνευσης ανωμαλιών, η οποία μπορεί να ανιχνεύει και καινούριου τύπου κακόβουλες εφαρμογές. Αυτό μπορεί να πραγματοποιηθεί με το διαχωρισμό των φυσιολογικής δραστηριότητας από εκείνη που παρακινείται από κάποιο κακόβουλο κώδικα. Καθορίζεται ένα όριο, έτσι ώστε να μπορεί να αναγνωρίζεται τι είναι φυσιολογικό και τι όχι. Η μεθοδολογία για αυτό τον διαχωρισμό, γίνεται με βάση τον ερευνητή.

Η μέθοδος ανίχνευσης ανωμαλιών παρουσιάζει το μειονέκτημα ότι ενεργοποιεί πολύ συχνά λάθος συναγερμούς αν δεν γίνει σωστά ο διαχωρισμός των κακόβουλων προγραμμάτων από τα φυσιολογικά. Έτσι για να αποφευχθεί αυτό χρειάζονται να βρεθούν οι κατάλληλες τεχνικές και να εξεταστούν τα κατάλληλα στοιχεία.

Στην παρούσα εργασία θα ελεγχθεί κατά πόσο μπορεί να δημιουργηθεί κάποιο μοντέλο ανίχνευσης εισβολών στηριζόμενο στη ανίχνευση ανωμαλιών, το οποίο θα παρουσιάζει υψηλά ποσοστά πρόβλεψης τόσο σε ήδη αναγνωρισμένες, όσο και σε νέες επιθέσεις. Είναι πολύ σημαντικό τα ποσοστά λάθους να διατηρηθούν χαμηλά.

Κεφάλαιο 2

Αρχές Ασφάλειας

Εισαγωγή
2.1 Έννοια ασφάλειας
2.2 Αρχές σχεδιασμού συστήματος
2.3 Αρχές σχεδιασμού συστημάτων ασφαλείας

Εισαγωγή

Η ασφάλεια ενός συστήματος είναι ένα θέμα που απασχολεί όλα τα υπολογιστικά συστήματα μικρών ή μεγάλων οργανισμών. Ένα σύστημα πρέπει να παρέχει ασφάλεια και οι χρήστες να νιώθουν ασφαλείς να το χρησιμοποιήσουν. Τα υπολογιστικά συστήματα και τα συστήματα ασφάλειας πρέπει να σχεδιάζονται προσεκτικά ώστε να παρέχουν ασφάλειας και να μην μπορεί εύκολα κάποιος εισβολέας να εκμεταλλευτεί τις οποιοδήποτε αδυναμίες του συστήματος. Σε αυτό το κεφάλαιο θα αναλυθεί η έννοια της ασφάλειας καθώς επίσης και κάποιες αρχές σχεδιασμού που πρέπει να διέπουν τόσο τα υπολογιστικά συστήματα, όσο και τα συστήματα ασφάλειας.

2.1 Έννοια ασφάλειας

Η ασφάλεια της πληροφορίας ήταν πάντοτε ένα πολύ σημαντικό θέμα, αφού μόνο συγκεκριμένα άτομα μπορούν να έχουν πρόσβαση στην πληροφορία και δεν πρέπει ποτέ να υποπέσει στα χέρια ατόμων τα οποία έχουν ως αποσκοπούν στην πρόκληση μιας ανεπιθύμητης κατάστασης.

Με την εισαγωγή του ηλεκτρονικού υπολογιστή παρουσιάστηκε και η ανάγκη για τη δημιουργία αυτοματοποιημένων εργαλείων, τα οποία θα μπορούσαν να παρέχουν προστασία στην πληροφορία που είναι αποθηκευμένη σε ένα σύστημα. Η ανάγκη για προστασία παρουσιάζεται επιτακτική ειδικά σε συστήματα, που παρουσιάζουν πολλαπλούς χρήστες και πρόσβαση σε δεδομένα. Έτσι η διαδικασία εύρεσης των

κατάλληλων εργαλείων που θα προστατεύουν τα δεδομένα και δεν θα επιτρέπουν την πρόσβαση σε κακόβουλα άτομα, ονομάζεται ασφάλεια ηλεκτρονικού υπολογιστή.

Η παρουσία των διαμοιραζόμενων συστημάτων και η χρησιμοποίηση των δικτύων για επικοινωνία και ανταλλαγή δεδομένων μεταξύ συστημάτων και μεταξύ συστήματος και χρήστη παρουσίασε και την ανάγκη για την ύπαρξη εργαλείων, τα οποία θα προστατεύουν τα δεδομένα κατά την μετάδοσής τους. Αυτή η κατάσταση οδήγησε στην προστασία του δικτύου.

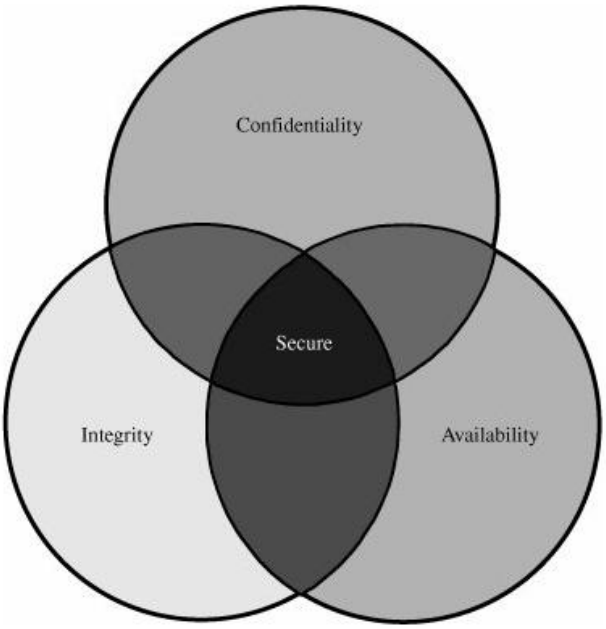
Πολλοί οργανισμοί έχοντας κατανοήσει το πόσο πολύτιμα είναι τα δεδομένα που έχουν αποθηκευμένα στα συστήματά τους και πόσο ευάλωτοι είναι οι πόροι του συστήματος, έχουν πάρει τα ανάλογα μέτρα ώστε να εξασφαλίσουν την προστασία της πληροφορίας που περιέχουν. Αντίθετα, άλλοι οργανισμοί δεν έχουν λάβει τα απαραίτητα μέτρα ασφαλείας του συστήματος, αφού δεν αναγνωρίζουν την αναγκαιότητα της προστασίας του συστήματος τους. Η απουσία προστασίας είναι πολύ χειρότερη όταν δεν υπάρχει η γνώση του πόσο ευάλωτο είναι ένα σύστημα και τα δεδομένα που περιέχει σε επιθέσεις

Ένα σύστημα για να μπορεί να αποκαλείται ασφαλές πρέπει να ικανοποιεί 3 στόχους:

- **Εμπιστευτικότητα:** Μόνο άτομα εξουσιοδοτημένα πρέπει να έχουν πρόσβαση στο σύστημα. Λέγεται επίσης μυστικότητα.

- **Ακεραιότητα:** Οποιαδήποτε πληροφορία ή πόρος του συστήματος μπορεί να τροποποιηθεί μόνο από κάποιο εξουσιοδοτημένα άτομα και σύμφωνα με τους τρόπους που του επιτρέπονται.

- **Διαθεσιμότητα:** Κάθε εξουσιοδοτημένο άτομο που επιθυμεί να έχει πρόσβαση στο σύστημα, αυτή πρέπει να του επιτρέπεται. Συχνά αυτό το χαρακτηριστικό αναφέρεται και με την αντίθετη έννοια του «άρνηση χρήσης».



Σχ. 2.1 Σχέση Εμπιστευτικότητας, Ακεραιότητας και Διαθεσιμότητας

Υπάρχει πρόκληση στο να ικανοποιηθούν αυτοί οι 3 στόχοι στη δημιουργία ενός ασφαλούς συστήματος. Αυτό οφείλεται κυρίως στο γεγονός ότι τα 3 πιο πάνω χαρακτηριστικά αρκετές φορές συγκρούονται μεταξύ τους και πρέπει να υπάρχει κάποια ισορροπία μεταξύ τους.

Σε ορισμένα βιβλία αναφέρεται επίσης και ένας τέταρτος στόχος. Αυτός είναι η αυθεντικότητα [18]. Ένα σύστημα πρέπει να είναι σε θέση να επαληθεύσει την ταυτότητα του χρήστη.

Ένα υπολογιστικό σύστημα αποτελείται από διάφορα στοιχεία, όπως είναι το υλικό, το λογισμικό, τα δεδομένα και τα δίκτυα και οι γραμμές επικοινωνίας. Στον πιο κάτω πίνακα παρουσιάζονται οι απειλές που μπορεί να παρουσιαστούν σε κάθε ένα από τα στοιχεία με βάση τους 3 πιο πάνω στόχους.

	Διαθεσιμότητα	**Εμπιστευτικότητα**	**Ακεραιότητα**
Υλικό	Ο εξοπλισμός κλάπηκε ή είναι απενεργοποιημένος με αποτέλεσμα να μην αρνείται την εξυπηρέτηση		

Λογισμικό	Προγράμματα διαγράφηκαν ή αρνούνται πρόσβαση στους χρήστες	Γίνεται ένα μη εξουσιοδοτημένο αντίγραφο.	Ένα εκτελέσιμο πρόγραμμα τροποποιείται είτε για να αποτύχει η εκτέλεσή του ή για να προκαλέσει άθελα την εκτέλεση κάποιου έργου.
Δεδομένα	Αρχεία διαγράφηκαν ή αρνούνται πρόσβαση στους χρήστες	Λαμβάνει μέρος μη εξουσιοδοτημένο διάβασμα αρχείου.	Υπάρχοντα αρχεία τροποποιούνται ή νέα αρχεία παραποιούνται
Δίκτυα και γραμμές επικοινωνίας	Μηνύματα καταστράφηκαν ή διαγράφηκαν. Γραμμές επικοινωνίας και δίκτυα δεν είναι διαθέσιμα.	Τα μηνύματα διαβάζονται. Η κίνηση των μηνυμάτων στο δίκτυο παρακολουθείται	Τα μηνύματα τροποποιούνται, καθυστερούν, αναδιοργανώνεται η σειρά τους ή δημιουργούνται αντίγραφά τους. Δημιουργούνται και ψεύτικα μηνύματα.

Πίνακας 2.1 Απειλές στα διάφορα στοιχεία [18]

2.2 Αρχές Σχεδιασμού Συστήματος

Κάθε σύστημα παρουσιάζει τη δική του λειτουργικότητα. Υπάρχουν συστήματα τα οποία δεν παρουσιάζουν κάποια μορφή ασφάλειας και ο οποιοσδήποτε χρήστης έχει πρόσβαση σε όλες τις πληροφορίες του συστήματος. Αντίθετα άλλα συστήματα ελέγχουν ποιος έχει πρόσβαση και τι δικαιώματα σε κάθε μέρος των δεδομένων που είναι αποθηκευμένα στο σύστημα. Κάποια άλλα συστήματα περιορίζουν ακόμα περισσότερο την πρόσβαση σε δεδομένα είτε με το να περιορίζουν δυνατότητα πρόσβασης κάποιες συγκεκριμένες χρονικές στιγμές είτε με το να ελέγχουν κατά κάποιο τρόπο τις κινήσεις κάποιου εξουσιοδοτημένου χρήστη, έτσι ώστε να αποφευχθεί η διαρροή της πληροφορίας σε τρίτους. Άλλα συστήματα απομονώνουν τους χρήστες τους σε κάποιο συγκεκριμένο προσωπικό χώρο εργασίας. Σε τέτοιου είδους συστήματα τα δεδομένων που είναι αποθηκευμένα στο σύστημα μοιράζονται στους χρήστες.

Όποια όμως και να είναι η λειτουργία του συστήματος, εναπόκειται στο ίδιο το σύστημα να αποτρέψει παραβιάσεις ασφάλειας. Ο σχεδιασμός ενός συστήματος που να είναι σε θέση να αποτρέπει όλες τις μη εξουσιοδοτημένες ενέργειες είναι πολύ

δύσκολος, όμως μπορεί να στηριχτεί σε κάποιες αρχές, οι οποίες θα βοηθήσουν στην ελαχιστοποίηση των αδυναμιών.

Οι αδυναμίες είναι κάποια λάθη που υπάρχουν στο σύστημα από το σχεδιασμό του, τα οποία δεν είναι εμφανή. Όταν κάποιος ανακαλύψει αυτές τις αδυναμίες μπορεί να τις εκμεταλλευτεί για να εισέλθει στο σύστημα χωρίς εξουσιοδότηση και για να επιτεθεί σε ένα σύστημα.

Πιο κάτω παρατίθενται οι αρχές σχεδιασμού ενός συστήματος χωρίς αδυναμίες [16].

- Ο σχεδιασμός πρέπει να είναι όσο πιο απλός και μικρός γίνεται έτσι ώστε να μπορούν να ανιχνεύονται εύκολα ανεπιθύμητες προσβάσεις σε διάφορα μονοπάτια κατά την κανονική λειτουργία του συστήματος.

- Η πρόσβαση στο σύστημα πρέπει να απαγορεύεται, αλλά το πρότυπο ασφάλειας να μπορεί να αναγνωρίζει τις συνθήκες υπό τις οποίες η πρόσβαση μπορεί να επιτρέπεται, δηλαδή αν ο χρήστης είναι εξουσιοδοτημένος ή όχι. Η προσπάθεια εύρεσης των συνθηκών υπό τις οποίες θα πρέπει να απαγορεύεται η πρόσβαση παρουσιάζει προβλήματα.

- Η κάθε πρόσβαση σε κάποιο αντικείμενο του συστήματος πρέπει να ελέγχεται για εξουσιοδότηση. Πρέπει να υπάρχει ανανέωση στις λίστες ελέγχου μόλις γίνει κάποια τροποποίηση. Αυτή η αρχή αποτελεί το βασικό στοιχείο σε κάποιο σύστημα ασφάλειας.

- Οι διάφοροι μηχανισμοί του συστήματος δεν πρέπει να στηρίζονται σε άγνοια των πιθανών εισβολέων για το σχεδιασμό, αλλά πρέπει να στηρίζονται στην κατοχή συγκεκριμένων και κλειδιών και κωδικών που παρέχουν πιο εύκολα ασφάλεια.

- Ένα μέτρο προστασίας που είναι πιο αποτελεσματικό είναι ο διαχωρισμός δικαιωμάτων. Δηλαδή για πρόσβαση σε κάποιο πόρο ή σε κάποια δεδομένα του συστήματος, θα πρέπει να απαιτείται η παρουσίαση 2 «κλειδιών» από 2 διαφορετικά άτομα παρά μόνο από 1.

- Κάθε πρόγραμμα και κάθε χρήστης του συστήματος πρέπει να έχουν τα ελάχιστα δικαιώματα, τα οποία χρειάζονται για να ολοκληρώσουν για να ολοκληρώσουν την εκτέλεσής τους. Αυτό μπορεί να περιορίσει τη ζημιά από κάποιο λάθος, αλλά επίσης μειώνει και τον αριθμό των αλληλοεπιδράσεων μεταξύ των διάφορων προγραμμάτων.

- Το ποσοστό των μηχανισμών που είναι κοινό σε περισσότερους από ένα χρήστη πρέπει να είναι πολύ περιορισμένο. Αυτό περιορίζει τυχόν μονοπάτια πληροφορίας που μπορεί να δημιουργηθούν μεταξύ χρηστών και έτσι δεν τίθεται σε κίνδυνο η ασφάλεια του συστήματος.

- Η διαπροσωπεία του συστήματος πρέπει να είναι εύκολη στη χρήση έτσι ώστε οι χρήστες του συστήματος να μπορούν να διαλέξουν σωστά τους μηχανισμούς ασφάλειας.

Εκτός από τις πιο πάνω αρχές πρέπει να ληφθούν υπόψη και 2 άλλοι παράγοντες.

- Ο υπολογισμός του χρόνου που χρειάζεται κάποιος για να ανακαλύψει τον τρόπο που του επιτρέπει πρόσβαση στο σύστημα μπορεί να υπολογιστεί πολύ εύκολα. Ο μηχανισμός ασφάλειας που θα χρησιμοποιεί το σύστημα δεν πρέπει να δέχεται ήττα με εύκολο και άμεσο τρόπο.

- Μπορούν να εφαρμοστούν μηχανισμοί οι οποίοι να καταγράφουν πότε κάποια πληροφορία έχει εκτεθεί σε κάποιο χρήστη. Έτσι έστω και αν υπάρχει ευκολία στην απόκτηση πληροφορίας από κάποιο τρίτο, αυτή η ενέργεια θα καταγραφεί στο σύστημα και έτσι την επόμενη φορά που θα ενωθεί κάποιος πραγματικός χρήστης του συστήματος θα ανακαλύψει εύκολα την κατάχρηση ή την υποκλοπή της πληροφορίας.

2.3 Αρχές σχεδιασμού συστημάτων ασφαλείας

Πέρα από τους πιο πάνω τρόπους προστασίας είναι απαραίτητη και η ύπαρξη κάποιου συστήματος ασφαλείας. Πιο κάτω περιγράφονται κάποιες αρχές σχεδιασμού συστημάτων ασφάλειας οι οποίες χρειάζονται να ληφθούν υπόψη [20] .

- **Υποστήριξη του σκοπού του οργανισμού:** Σκοπός ενός συστήματος ασφάλειας είναι να προστατεύει τους πόρους ενός οργανισμού (πληροφορία, λογισμικό, υλικό). Οι κανόνες ασφάλειας δεν πρέπει να επιλέγονται πρόχειρα, αλλά πρέπει να δίνεται μεγάλη σημασία σε αυτούς. Αρχικά πρέπει να καθορίζεται ο σκοπός ενός οργανισμού και μετά να καθοριστούν τα χαρακτηριστικά που πρέπει να διέπουν το σύστημα ασφάλειας, ώστε να ικανοποιείται ο σκοπός. Με αυτό τον τρόπο βελτιώνονται και οι υπηρεσίες του συστήματος που παρέχονται στο χρήστη.

- **Αποτελεί στοιχείο για Σωστή Διαχείριση:** Τα συστήματα αποτελούν ένα σημαντικό μέρος για την υποστήριξη του σκοπού ενός οργανισμού. Είναι απαραίτητο να ληφθούν υπόψη οι συνέπειες που μπορεί να υπάρξουν σε ένα σύστημα με την εισαγωγή των μηχανισμών ασφάλειας, αν και κάτι τέτοιο δεν αποκλείει την πιθανότητα να προκληθεί ζημιά στο σύστημα. Όταν είναι σύστημα είναι συνδεδεμένο και με άλλα εξωτερικά συστήματα πρέπει να ληφθεί υπόψη το είδος της ασφάλειας που έχουν και ότι παρέχουν ασφάλεια και στα συστήματα του οργανισμού.

- **Ικανοποιητικό κόστος:** Το κόστος εγκατάστασης ενός συστήματος ασφάλειας πρέπει να ληφθεί υπόψη, τόσο από άποψη χρημάτων όσο και από άποψη επίδοσης. Η ασφάλεια σε ένα σύστημα πρέπει να ικανοποιεί τις ανάγκες ενός συστήματος. Επενδύοντας σε μέτρα ασφάλειας μπορούν να μειωθούν οι απώλειες που αφορούν θέματα ασφάλειας. Έτσι το κόστος για την εγκατάσταση ελέγχων ασφαλείας πρέπει να είναι τέτοιο ώστε να αντιμετωπίζονται αποτελεσματικά τυχόν απειλές, αλλά να μην επηρεάζεται αρνητικά η λειτουργία και η επίδοση του συστήματος.

- **Ενημέρωση εξωτερικών χρηστών:** Πρέπει να υπάρχει ενημέρωση των εξωτερικών χρηστών ενός συστήματος για την ύπαρξη μέτρων ασφάλειας στο σύστημα, ώστε να υπάρχει η αίσθηση προστασίας. Ενημερώνοντας όμως τους χρήστες για την ύπαρξη μηχανισμών ασφάλειας δεν πρέπει να τίθεται σε κίνδυνο το σύστημα.

- **Σαφείς ευθύνες:** Οι ευθύνες των ιδιοκτητών του συστήματος και των χρηστών πρέπει να είναι ρητά καθορισμένες. Σε ένα οργανισμό το μέγεθος του

προγράμματος ασφάλειας είναι άμεσα συνυφασμένο με το μέγεθος του οργανισμού. Έτσι πρέπει να δημιουργείται ένα έγγραφο που να καθορίζει την πολιτική του οργανισμού και να δηλώνει ρητά τις ευθύνες που αφορούν την ασφάλεια.

- **Περιεκτική προσέγγιση:** Ένα αποτελεσματικό σύστημα ασφάλειας στηρίζεται σε μια περιεκτική προσέγγιση, όπου λαμβάνεται υπόψη όλος ο κύκλος της πληροφορίας. Για να λειτουργεί αποτελεσματικά το σύστημα ασφάλειας πρέπει να ληφθούν υπόψη οι λειτουργίες ενός συστήματος και ποιες είναι οι αλληλεξαρτήσεις τους και άλλα θέματα όπως η διαχείριση του συστήματος, ώστε το σύστημα ασφάλειας να ενσωματωθεί ομαλά στο υπάρχον σύστημα.

- **Συντήρηση:** Τα συστήματα ασφάλειας πρέπει να ελέγχονται περιοδικά. Η τεχνολογία, τα συστήματα, η ροή πληροφορίας παρουσιάζουν συνεχώς αλλαγές, οι οποίες αντανακλούν και αλλαγές στα συστήματα ασφάλειας. Έτσι είναι απαραίτητο να γίνονται κάποιες μετατροπές στο σύστημα ασφάλειας. Επίσης ένα σύστημα ασφάλειας δεν είναι ποτέ τέλειο με αποτέλεσμα συνεχώς να ανακαλύπτονται τρόποι εισβολής σε ένα σύστημα. Πρέπει να υπάρχει συνεχής έλεγχος ότι το σύστημα ασφάλειας μπορεί να προστατέψει τα πληροφορικά συστήματα.

- **Υποστήριξη κοινωνικών παραγόντων:** Οι τρόποι με τους οποίους ένα σύστημα ασφαλείας πρέπει να είναι σχεδιασμένο περιορίζονται και από κάποιους κοινωνικούς παράγοντες. Τα μέτρα ασφαλείας που πρέπει να λαμβάνονται πρέπει να είναι σύμφωνα με τα δικαιώματα των χρηστών και πρέπει να είναι τα τέτοια ώστε να είναι σύμφωνα με την νομοθεσία.

Κεφάλαιο 3

Κακόβουλες Εφαρμογές

Εισαγωγή

3.1 Επιθέσεις στο σύστημα

3.2 Τύποι Εισβολέων

3.3 Κακόβουλος Κώδικας

3.4 Τρόποι επισύναψης ιών σε προγράμματα

Εισαγωγή

Στο προηγούμενο κεφάλαιο έγινε αναφορά στην έννοια της ασφάλειας και στο πώς πρέπει να είναι σχεδιασμένο ένα σύστημα ώστε να μπορεί να ικανοποιεί τους στόχους εμπιστευτικότητα, ακεραιότητα, διαθεσιμότητα και αυθεντικότητα. Επίσης έγινε αναφορά και στις αρχές που πρέπει να διέπουν ένα σύστημα ασφάλειας.

Σε αυτό το κεφάλαιο θίγονται τα θέματα που αφορούν τις κακόβουλες εφαρμογές. Γίνεται αναφορά στους τρόπους που μπορεί κάποιος να επιτεθεί σε ένα σύστημα, στο τι εννοούμε με τον όρο κακόβουλος κώδικας και σε ποιες κατηγορίες χωρίζεται. Τέλος παρατίθενται οι τρόποι με τους οποίους ένα ιός μπορεί να επισυναφτεί σε ένα σύστημα.

3.1 Επιθέσεις στο σύστημα

Κάθε μέρος υπολογιστικού συστήματος μπορεί να αποτελέσει στόχος κάποιου εγκλήματος. Υπάρχει μια αρχή για εύκολη διείσδυση, η οποία αναφέρει ότι κάποιος εισβολέας είναι αναμενόμενο ότι θα χρησιμοποιήσει όλα τα μέσα για να εισβάλει σε ένα σύστημα. Μπορεί να μην γίνει με τον πιο προφανή τρόπο και όχι από το μέρος όπου είναι εγκατεστημένη η πιο ισχυρή άμυνα. Και σίγουρα δεν θα είναι με τον τρόπο, από τον οποίο περιμένουμε τον εισβολέα ότι θα επιτεθεί.

Οι πιθανές παραβάσεις ασφαλείας χωρίζονται σε 3 κατηγορίες:

- Μη εξουσιοδοτημένη πρόσβαση σε πληροφορία: Ένα άτομο χωρίς εξουσιοδότηση έχει πρόσβαση σε πληροφορίες που είναι αποθηκευμένες σε ένα ηλεκτρονικό υπολογιστή. Μπορεί να είναι σε θέση να δει μόνο κάποια στοιχεία από τα δεδομένα, αλλά με αυτόν τον τρόπο να καταλάβει το περιεχόμενο τους. Επίσης μπορεί να έχει μη εξουσιοδοτημένη πρόσβαση και χρήση σε κάποιο πρόγραμμα του συστήματος.

- Μη εξουσιοδοτημένη τροποποίηση πληροφορίας: Ένα μη εξουσιοδοτημένο άτομο μπορεί να είναι σε θέση να τροποποιήσει αποθηκευμένες πληροφορίες του συστήματος. Σημαντικό είναι το γεγονός ότι οι μετατροπές μπορούν να γίνουν και χωρίς το άτομο να γνωρίζει τα αρχικά στοιχεία.

- Μη εξουσιοδοτημένη άρνηση χρήσης: Μη εξουσιοδοτημένο άτομο μπορεί να μην επιτρέπει σε κάποιο εξουσιοδοτημένο χρήστη του συστήματος να έχει πρόσβαση ή να τροποποιήσει πληροφορίες. Δεν είναι απαραίτητο ο εισβολέας να γνωρίζει το περιεχόμενο των πληροφοριών.

Από τις πιο πάνω παραβάσεις φαίνεται ότι για να μπορεί να γίνει εφικτή η πραγματοποίηση τους σημαίνει προηγήθηκαν κάποιες επιθέσεις στο σύστημα. Οι επιθέσεις παρουσιάζονται σε τέσσερις γενικές κατηγορίες [18].

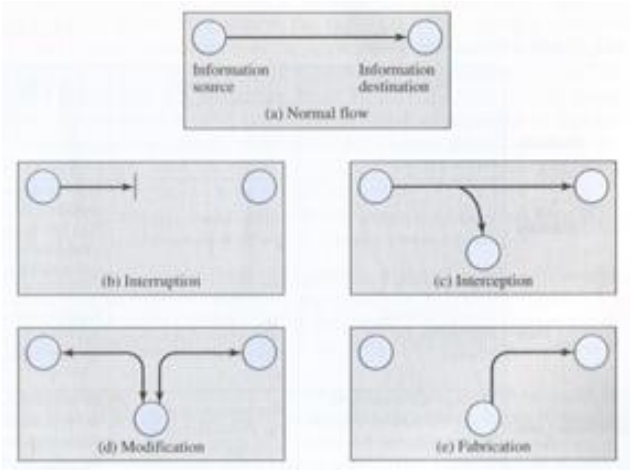
- Διακοπή (Interruption): Ένα στοιχείο – πόρος του συστήματος καταστρέφεται ή δεν είναι πλέον διαθέσιμος . Αυτή η επίθεση στοχεύει στην διαθεσιμότητα που αναφέρθηκε στο προηγούμενο κεφάλαιο. Παράδειγμα είναι καταστροφή μέρους του υλικού ή το κόψιμο μιας γραμμής επικοινωνίας.

- Αναχαίτιση (Interception): Ένα πρόσωπο χωρίς εξουσιοδότηση αποκτά πρόσβαση σε ένα πόρο του συστήματος. Το πρόσωπο μπορεί να είναι ένα άτομο, ένα πρόγραμμα ή ένας ηλεκτρονικός υπολογιστής. Αυτή η επίθεση αφορά την εμπιστευτικότητα.

- Τροποποίηση (Modification): Ένα μη εξουσιοδοτημένο πρόσωπο αποκτά πρόσβαση, αλλά επίσης μπορεί να τροποποιήσει και κάποιο πόρο – στοιχείο του συστήματος. Αυτή η επίθεση αναφέρεται στην ακεραιότητα.

- Σκευωρία (Fabrication): Ένα μη εξουσιοδοτημένο άτομο εισάγει πλαστά αντικείμενα στο σύστημα. Αυτή η επίθεση στοχεύει στην αυθεντικότητα.

Πιο κάτω παρουσιάζονται σχηματικά οι κατηγορίες επιθέσεων σε σύγκριση με μια φυσιολογική ροή πληροφορίας.

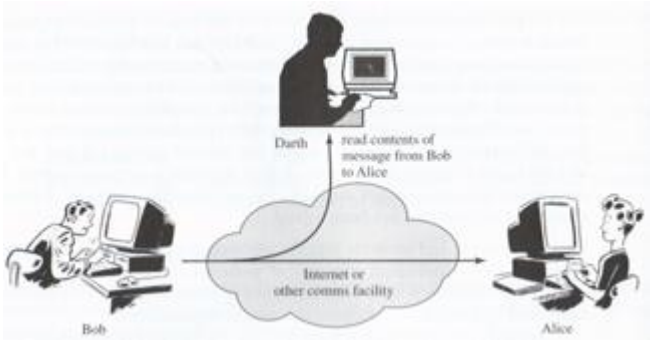


Σχήμα 3.1 Επιθέσεις κατά του συστήματος

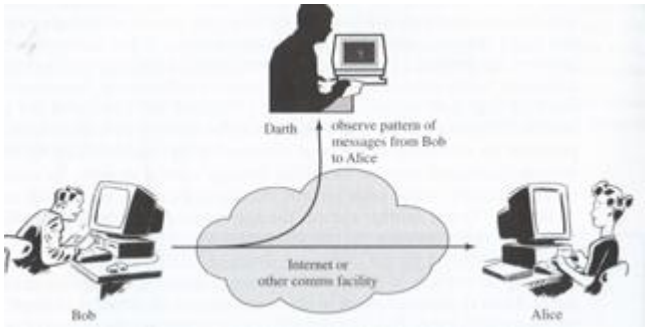
Οι επιθέσεις που γίνονται εναντίον ενός συστήματος μπορεί να είναι είτε παθητικές είτε ενεργές [19].

Στις παθητικές επιθέσεις γίνεται προσπάθεια υποκλοπής πληροφορίας από το σύστημα, η οποία μεταδίδεται, αλλά δεν επηρεάζονται οι πόροι του συστήματος. Οι παθητικές επιθέσεις παρουσιάζουν 2 τύπους επιθέσεων. Η πρώτη αφορά την γνωστοποίηση των περιεχομένων ενός μηνύματος που μεταδίδεται, το οποίο μπορεί να περιέχει πολύ σημαντικές πληροφορίες. Ο δεύτερος τύπος παθητικής επίθεσης είναι η ανάλυση της κίνησης ενός δικτύου, η οποία είναι πιο δυσδιάκριτη. Αν χρησιμοποιείται κάποιος τρόπος προστασίας των δεδομένων, όπως είναι η κρυπτογράφηση, τότε δεν μπορεί κάποιος να μάθει την πληροφορία που κινείται μέσα σε ένα σύστημα. Αντίθετα μπορεί να ανακαλύψει τα πρόσωπα της επικοινωνίας και μέσω του μεγέθους των δεδομένων και της συχνότητας με την οποία ανταλλάζουν μηνύματα να μπορέσει να ανακαλύψει το λόγο της επικοινωνίας. Οι παθητικές επιθέσεις είναι δύσκολο να ανιχνευτούν, γιατί η δράση τους δεν εμπεριέχει μετατροπή των δεδομένων. Κατά τη διάρκεια μιας επικοινωνίας μεταξύ 2 ατόμων δεν γίνεται αντιληπτό ότι κάποιος μπορεί να γνωρίζει το περιεχόμενο των μηνυμάτων που ανταλλάζουν ή να παρακολουθεί την κυκλοφορία των

μηνυμάτων. Η αντιμετώπιση των παθητικών επιθέσεων μπορεί να γίνει με τρόπους πρόληψης. Πιο κάτω παρουσιάζονται τα 2 είδη παθητικών επιθέσεων.



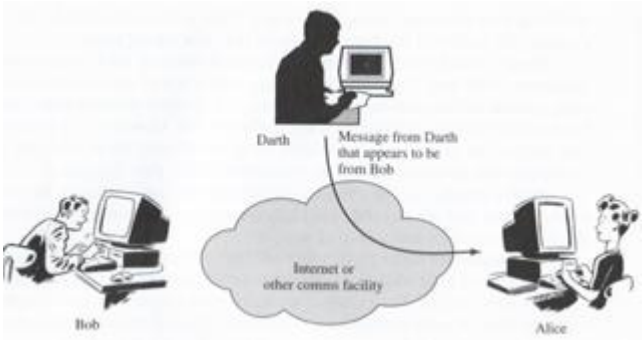
Σχήμα 3.2 Γνωστοποίηση πληροφοριών



Σχήμα 3.3 Ανάλυση κίνησης δικτύου

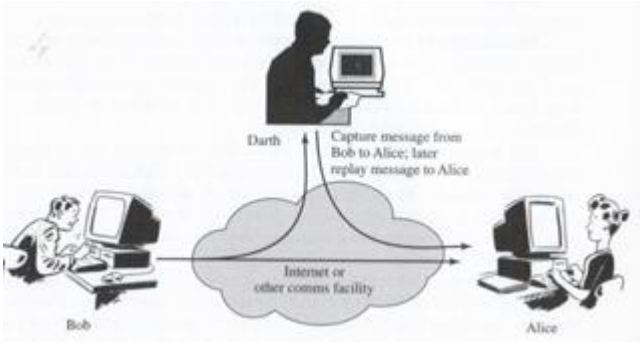
Αντίθετα οι ενεργές επιθέσεις παρουσιάζουν διαφορές στον τρόπο με τον οποίο επιτίθενται σε ένα σύστημα και περιλαμβάνουν τροποποίηση δεδομένων ή δημιουργία λανθασμένων δεδομένων. Χωρίζονται σε 4 κατηγορίες: πλαστοπροσωπία, επανάληψη, τροποποίηση μηνυμάτων και άρνηση χρήσης.

Η πλαστοπροσωπία υπάρχει όταν ένα άτομο προσποιείται κάποιο άλλο με αποτέλεσμα να αποκτά τα προνόμια και τα δικαιώματα που έχει το συγκεκριμένο άτομο. Συνήθως αυτού του είδους ενεργής επίθεσης γίνεται με την ανάμειξη και κάποιας άλλης κατηγορίας.



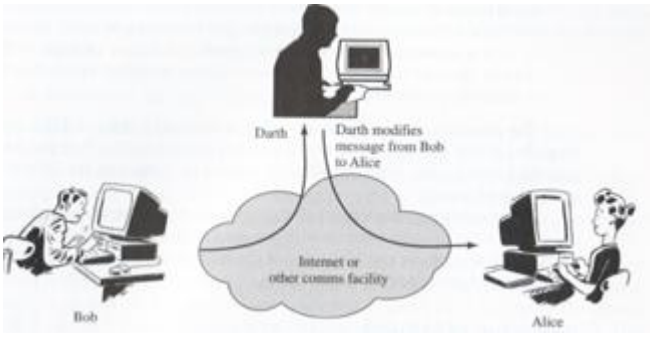
Σχήμα 3.4 Πλαστοπροσωπία

Η επανάληψη συμπεριλαμβάνει τη δέσμευση δεδομένων και την των δεδομένων με σκοπό να δημιουργήσει ένα μη εξουσιοδοτημένο γεγονός.



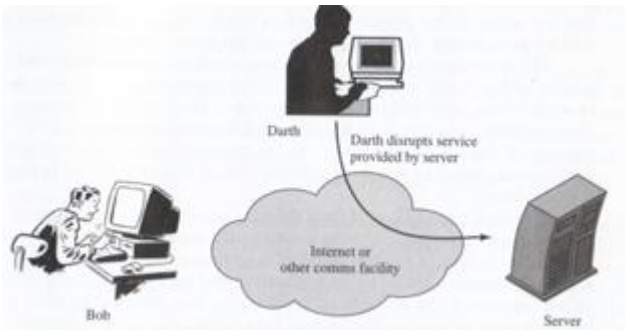
Σχήμα 3.5 Επανάληψη

Η τροποποίηση κάποιου μηνύματος περιλαμβάνει αλλαγές σε κάποια μηνύματα ή καθυστέρηση μετάδοσης τους με αποτέλεσμα να γίνεται κάτι που δεν επιτρέπεται.



Σχήμα 3.6 Τροποποίηση

Η άρνηση χρήσης όπως αναφέρθηκε και πιο πάνω παρεμποδίζει την ομαλή λειτουργία του συστήματος και των επικοινωνιών. Μπορεί να παρεμποδίζει τη μετάδοση μηνυμάτων προς συγκεκριμένο προορισμό ή να υπερφορτώνει το δίκτυο με μηνύματα, έτσι ώστε να μειώνεται η επίδοση του συστήματος.



Σχήμα 3.7 Άρνηση χρήσης

Οι ενεργές επιθέσεις παρουσιάζουν αντίθετο χαρακτήρα από ότι οι παθητικές. Δεν μπορούν να εφαρμοστούν μέτρα πρόληψης όλων των ενεργών επιθέσεων που μπορεί να λάβουν μέρος σε ένα σύστημα, για αυτό και η καλύτερη αντιμετώπιση τους είναι η ανίχνευση.

Η συνεχής χρήση του διαδικτύου για την εύρεση πληροφοριών μπορεί να αποτελέσει εύκολο στόχο για την εισαγωγή κακόβουλου κώδικα στο σύστημα. Οι επιθέσεις μπορούν να πραγματοποιηθούν παντού από το πιο ασθενές στο πιο δυνατό σύστημα προστασίας.

3.2 Τύποι Εισβολέων

Ο εισβολέας ή ευρέως γνωστός ως hacker είναι εξίσου επικίνδυνος, όπως και ο κακόβουλος κώδικας. Οι επιθέσεις τους μπορεί να είναι αθώες, αλλά μπορεί να προκαλέσουν και μεγάλες ζημιές στο σύστημα. Υπάρχουν 3 διαφορετικοί τύποι εισβολέων.

- **Masquerader:** Είναι το άτομο που δεν είναι εξουσιοδοτημένο για να χρησιμοποιεί το σύστημα, αλλά βρίσκει τρόπο να διαπεράσει τους ελέγχους

πρόσβασής σε αυτό και να διερευνήσει το λογαριασμό ενός αυθεντικού χρήστη. Δεν έχει σχέση με το σύστημα.

- **Misfeasor:** Είναι ένας αυθεντικός χρήστης του συστήματος, ο οποίος έχει πρόσβαση σε δεδομένα, προγράμματα ή πόρους για τους οποίους δεν είναι εξουσιοδοτημένος. Μπορεί ακόμα να έχει εξουσιοδότηση να τα χρησιμοποιεί, αλλά να κακομεταχειρίζεται τα δικαιώματά του.

- **Clandestine User:** Είναι ένα πρόσωπο που αποκτά τον έλεγχο του συστήματος και το χρησιμοποιεί για έχει πρόσβαση σε audit records. Μπορεί να είναι χρήστης του συστήματος ή να είναι κάποιο εξωτερικό πρόσωπο.

3.3 Κακόβουλος Κώδικας

Οι επιθέσεις μπορούν να γίνουν με τη χρήση κακόβουλων εφαρμογών, οι οποίες εκτελούν διάφορες διεργασίες χωρίς να το γνωρίζει ο χρήστης. Η εισαγωγή κακόβουλου κώδικα στο σύστημα μπορεί να γίνει με διάφορους τρόπους. Πέραν από επιθέσεις που επιδιώκουν την πρόσβαση στο σύστημα, κάποιος χρήστης του συστήματος μπορεί να επιτρέψει σε κακόβουλο κώδικα να εγκατασταθεί στο σύστημα, μετά από λανθασμένες εκτιμήσεις. Για παράδειγμα στην προσπάθεια του να αποκτήσει κάποιο συγκεκριμένο λογισμικό μέσω του διαδικτύου μπορεί να κατεβάσει κάποιο πακέτο εφαρμογών το οποίο πέρα από τα προγράμματα που αναφέρει ότι περιέχει να περιέχει και κακόβουλο κώδικα. Παρόμοιο παράδειγμα με το πιο πάνω είναι η περίπτωση στην οποία για την πλήρη εμφάνιση μιας ιστοσελίδας χρειάζεται η εγκατάσταση συγκεκριμένων προγραμμάτων. Έτσι άθελα του ο χρήστης μπορεί να επιτρέψει και την εισαγωγή κακόβουλων εφαρμογών. Υπάρχει μια αβεβαιότητα στο κατά πόσο κάτι που ο χρήστης κατεβάζει από το διαδίκτυο είναι ασφαλές και δεν περιέχει κακόβουλο κώδικα. Τα προγράμματα και τα δεδομένα που ανταλλάζονται είναι πολλά, όπως επίσης και οι διάφορες τροποποιήσεις που μπορούν να γίνουν σε κάποια αρχεία. Όλα αυτά μπορούν να συμβούν χωρίς τη συναίνεση του χρήστη ή την ενημέρωσή του.

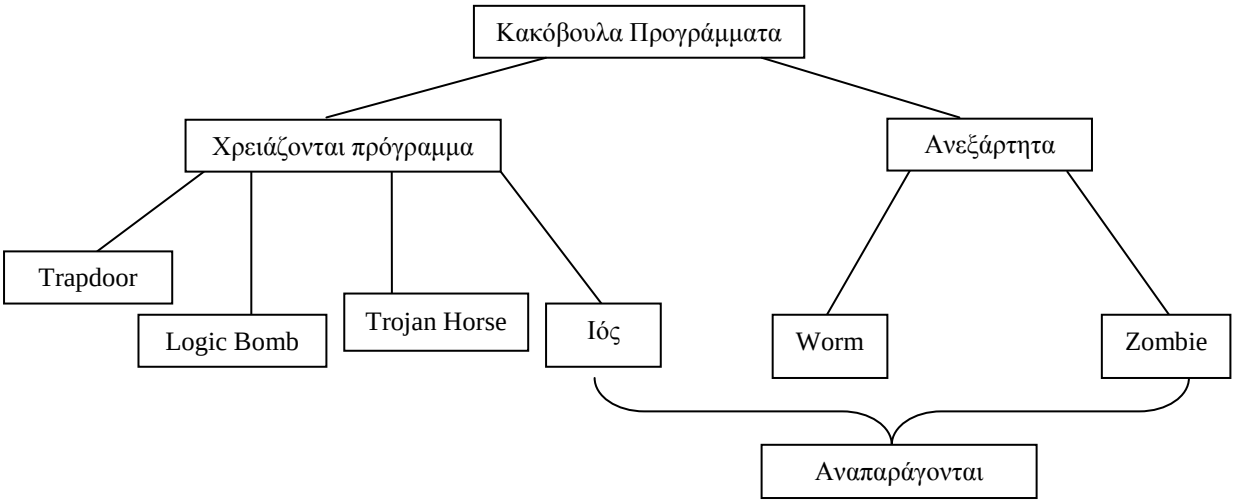
Ο κακόβουλος κώδικας συμπεριφέρεται με τρόπο που δεν αναμένεται. Δεν έχει κάποιο σταθερό τρόπο δράσης, γιατί αυτός συνήθως εξαρτάται από τα σχέδια και τις προθέσεις του προγραμματιστή κακόβουλων εφαρμογών. Ένα κακόβουλο πρόγραμμα μπορεί να κάνει οτιδήποτε μπορεί και ένα συνηθισμένο πρόγραμμα ή μπορεί να κάνει διαφορετικά πράγματα κάθε φορά. Μπορεί να παρουσιάσει ένα μήνυμα, να παράξει ένα ήχο, να πάρει προσωπικά δεδομένα, να τροποποιήσει τη λειτουργία υπάρχοντων προγραμμάτων και να τροποποιήσει ή να καταστρέψει δεδομένα. Ακόμα μπορεί να αποκτήσει τα ίδια δικαιώματα με κάποιο χρήστη. Επίσης με την εισαγωγή της στο σύστημα, μια κακόβουλη εφαρμογή μπορεί και να μην κάνει τίποτα. Μπορεί να εισέρχεται στο σύστημα με σκοπό να βρίσκεται εκεί απαρατήρητο μέχρι ορισμένα συμβάντα να την ενεργοποιούν ώστε να προκαλέσει τη ζημιά για την οποία είναι σχεδιασμένη. Η ενεργοποίηση μπορεί να γίνεται με βάση κάποια ημερομηνία ή ώρα, κάποιο γεγονός, όπως η εκτέλεση συγκεκριμένου προγράμματος, την ικανοποίηση κάποιας συνθήκης ή με βάση κάποια τυχαία κατάσταση. Η ενεργοποίηση μπορεί να γίνει και με το συνδυασμό των πιο πάνω.

Γενικά η έννοια κακόβουλος κώδικας ή κακόβουλο πρόγραμμα είναι το γενικό όνομα για απρόβλεπτες και ανεπιθύμητες επιπτώσεις σε προγράμματα ή μέρη προγραμμάτων, οι οποίες δημιουργούνται από κάποιο συντελεστή, του οποίου στόχος είναι η πρόκληση ζημιάς. Ο συντελεστής που αναφέρθηκε πιο πάνω είναι ο συγγραφέας του προγράμματος ή το άτομο που προκαλεί τη διανομή του. Ο συγκεκριμένος ορισμός δεν περιλαμβάνει λάθη που παρουσιάζονται χωρίς πρόθεση, όπως λάθη σε λογισμικό, αν και αυτά μπορεί να έχουν αρνητικές συνέπειες. Επίσης δεν περιλαμβάνει σύμπτωση, όπου δύο προγράμματα μπορεί να συνδυαστούν και να οδηγήσουν σε μια επίσης αρνητική συνέπεια. Έτσι λάθη χωρίς κάποια πρόθεση από κακόβουλες εφαρμογές μπορεί να οδηγήσουν στα ίδια αποτελέσματα με τα κακόβουλα προγράμματα. Τα περισσότερα κακόβουλα προγράμματα μπορούν να εκτελεστούν σε συγκεκριμένα λειτουργικά συστήματα και κάποιες φορές σε συγκεκριμένο υλικό. Αυτό οφείλεται στο γεγονός ότι είναι σχεδιασμένα για να εκμεταλλευτούν αδυναμίες συγκεκριμένων συστημάτων, όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο.

Ο κακόβουλος κώδικας είναι κοινώς γνωστός ως ιός, αλλά στην πραγματικότητα χωρίζεται σε διάφορες κατηγορίες, οι οποίες ενώ παρουσιάζουν πάρα πολλά κοινά έχουν κάποιες μικροδιαφορές που τις διαφοροποιούν μεταξύ τους. Κατηγοριοποιούνται

κυρίως με βάση τη μέθοδο μεταφοράς τους. Οι κατηγορίες κακόβουλου κώδικα που παρουσιάζονται είναι οι ιοί, τα worms, τα Trojan horses, τα trapdoors, τα logic bombs, τα rabbit και τα zombie.

Οι πιο πάνω κατηγορίες κακόβουλου κώδικα μπορούν να διαχωριστούν μεταξύ τους σε δύο άλλες κατηγορίες, στις απειλές που χρειάζονται ένα πρόγραμμα για να μπορούν να εκτελεστούν και σε κείνες που είναι ανεξάρτητες. Η πρώτη κατηγορία είναι μέρη προγράμματος που δεν μπορούν να υπάρχουν ανεξάρτητα από κάποια εφαρμογή ή πρόγραμμα του συστήματος. Η δεύτερη κατηγορία είναι προγράμματα που μπορούν να υπάρξουν χωρίς τη στήριξη κάποιου προγράμματος και μπορούν να χρονοδρομολογηθούν και να εκτελεστούν από το λειτουργικό σύστημα. Επίσης μπορεί να γίνει και ένας άλλος διαχωρισμός των διαφόρων ειδών κακόβουλου κώδικα. Μπορούν να διαχωριστούν σε εκείνα που μπορούν να αναπαραχθούν και σε κείνα που δεν μπορούν. Τα είδη κακόβουλου κώδικα που μπορούν να αναπαραχθούν πέρα από την εκτέλεση τους, παράγουν και άλλα αντίγραφα του εαυτού τους. Πιο κάτω παρουσιάζεται η ταξινομία όπως αναφέρθηκε [19].



Σχήμα 3.8

3.3.1 Ιός

Ο ιός είναι ένα πρόγραμμα, το οποίο συμπεριφέρεται όπως οι βιολογικοί ιοί. Είναι ένα από τα μεγαλύτερα προβλήματα για την ασφάλεια ενός υπολογιστικού συστήματος.

Ο ιός είναι ένα πρόγραμμα και είναι αρκετά ενδιαφέρον ο τρόπος με τον οποίο λειτουργεί, γιατί έχει τη δυνατότητα να προσκολληθεί σε άλλα προγράμματα και να τα

κάνει να συμπεριφέρονται και αυτά σαν ιούς. Συγκεκριμένα τροποποιεί τα προγράμματα με τέτοιο τρόπο ώστε αυτά να έχουν ένα αντίγραφο του. Αυτό έχει ως αποτέλεσμα την γρήγορη εξάπλωσή του σε όλο το σύστημα ή το δίκτυο χρησιμοποιώντας τις εξουσιοδοτήσεις κάθε χρήστη για να «μεταδοθεί» και σε άλλα προγράμματα.[3]

Κάποιοι ιοί στοχεύουν σε συγκεκριμένο λογισμικό, ενώ άλλοι «μολύνουν» όποιο εκτελέσιμο αρχείο βρουν. Ένας ιός μπορεί να είναι είτε προσωρινός, δηλαδή εκτελείται όταν εκτελείται και το πρόγραμμα στο οποίο είναι προσκολλημένος, είτε μπορεί να είναι μόνιμος τοποθετώντας τον εαυτό του στη μνήμη με αποτέλεσμα να είναι παραμένει ενεργός και να μπορεί να εκτελείται ακόμα και αν το πρόγραμμα στο οποίο είναι προσκολλημένο τελειώσει την εκτέλεση του. Όταν ένας ιός χρησιμοποιείται σε συνδυασμό με ένα Trojan ότι θα προκαλέσει εξαπλωμένα denial of services και μη εξουσιοδοτημένες μεθοδεύσεις δεδομένων. Επίσης ένας πρόγραμμα μπορεί να έχει την ιδιότητα εξάπλωσης ενός ιού, αλλά να μην είναι επικίνδυνο, παρά μόνο ένα συνηθισμένο πρόγραμμα. Τέτοιου είδους προγράμματα ζητούν πάντα άδεια πριν να αρχίσουν την εκτέλεση τους.

Κατά τη διάρκεια ζωής του ένας ιός περνά από 4 διαφορετικά στάδια [18]:

1. Φάση Αδράνειας: Ο ιός είναι αδρανής. Δεν περνούν όλοι οι ιοί από τη συγκεκριμένη φάση. Κάποια χρονική στιγμή ο ιός θα ενεργοποιηθεί από κάποιο γεγονός (ημερομηνία, παρουσία κάποιου άλλου προγράμματος κ.τ.λ.).

2. Φάση Αναπαραγωγής: Ο ιός τοποθετεί ένα δικό του αντίγραφο σε άλλα προγράμματα ή συγκεκριμένα μέρη στο δίσκο του συστήματος. Έτσι το κάθε «μολυσμένο» πρόγραμμα περιέχει ένα αντίγραφο του ιού και μετά και εκείνο θα βρεθεί στην φάση αναπαραγωγής.

3. Φάση Ενεργοποίησης: Ο ιός ενεργοποιείται για να εκτελέσει την ενέργεια για την οποία προοριζόταν. Όπως αναφέρθηκε και πιο πάνω, η ενεργοποίηση οφείλεται στην πραγματοποίηση κάποιου γεγονότος ή στην ικανοποίηση κάποιας συνθήκης.

4. Φάση εκτέλεσης: Ο ιός εκτελείται. Η ενέργεια που πραγματοποιεί μπορεί να είναι αβλαβής ή αρκετά επιζήμια για το σύστημα.

Ο ιός μπορεί να διαχωριστεί περαιτέρω σε 7 τύπους [18] [23]:

- Ο ιός παράσιτο είναι η παραδοσιακή και πιο συχνή μορφή ιού. Ο παρασιτικός ιός προσκολλάται σε εκτελέσιμα αρχεία και αναπαράγεται όταν τα «μολυσμένα» αρχεία εκτελούνται, βρίσκοντας άλλα εκτελέσιμα αρχεία για να «μολύνει».

- Ο ιός memory – resident τοποθετεί τον εαυτό του στη μνήμη του συστήματος, ως μέρος ενός προγράμματος του συστήματος. Με αυτό τον τρόπο μπορεί να «μολύνει» κάθε πρόγραμμα που εκτελείται.

- Ο file ιός ελέγχει το file system του λειτουργικού συστήματος στο οποίο βρίσκεται. Μπορεί να «μολύνει» αρχεία, να δημιουργήσει αντίγραφα από υπάρχοντα αρχεία, να πολλαπλασιάσει τον εαυτό του σε διαφορετικούς καταλόγους ή να χρησιμοποιήσει τις ιδιότητες του file system για δικούς του σκοπούς.

- Ο ιός script είναι υποσύνολο του file ιού και αναφέρεται κυρίως σε αρχεία τα οποία είναι γραμμένα σε κάποια scripting language.

- Ο boot sector ιός μπορεί να μολύνει μόνο boot ή το master sector και μπορεί να εξαπλωθεί όταν το σύστημα κάνει εκκίνηση από το σκληρό δίσκο που περιέχει τον ιό. Ο συγκεκριμένος τύπος ιού έχει εξαφανισθεί από τότε που παρουσιάστηκε ο 32-bit επεξεργαστής.

- Ο ιός macro όπως φανερώνει και το όνομα του πραγματοποιείται όταν κάποιο macro του συστήματος λάβει μέρος. Τα macro είναι εκτελέσιμα προγράμματα τα οποία είναι ενσωματωμένα σε αρχεία. Χρησιμοποιούνται για την αυτοματοποίηση κάποιων επανειλημμένων ενεργειών με στόχο την διευκόλυνση του χρήστη. Συνήθως οι πιο γνωστοί macro ιοί αναφέρονται στα προϊόντα της Microsoft Office. Οι ιοί macro αποτελούν μια σχετικά υψηλού κινδύνου απειλή γιατί είναι ανεξάρτητα κάποια αρχιτεκτονικής ή κάποιου λειτουργικού συστήματος, αφού στοχεύουν σε συγκεκριμένες εφαρμογές.

Συνήθως «μολύνουν» κανονικά αρχεία - έγγραφα και όχι εκτελέσιμα, αφού αυτά αποτελούν το μεγαλύτερο ποσοστό πληροφορίας που εισέρχεται σε ένα σύστημα. Επίσης, μπορούν να εξαπλωθούν πολύ εύκολα. Ο ιός μπορεί να «μολύνει» macro τα οποία λειτουργούν αυτόματα από το σύστημα και έτσι όταν εκτελείται ένα macro, εκτελείται και ο ιός. Επίσης ο συγκεκριμένος τύπος ιού μπορεί να αναφέρεται και σε macro τα οποία πρέπει να θέσει σε λειτουργία ο χρήστης. Έτσι ανάλογα με το εκτελείται το macro, εκτελείται και ο συγκεκριμένος ιός.

- Ο ιός stealth είναι σχεδιασμένος για να μην είναι εμφανής από κάποιο λογισμικό ασφάλειας. Ο συγκεκριμένος όρος μπορεί να αναφερθεί περισσότερο ως τεχνική κάποιου ιού, παρά τύπος ιού.

- Ο πολυμορφικός ιός είναι ο ιός που μεταβάλλεται μετά από κάθε «μόλυνση». Κατά την αναπαραγωγή του δημιουργεί αντίγραφα, τα οποία έχουν την ίδια λειτουργικότητα αλλά διαφορετικό δομή των bits. Ο πολυμορφικός ιός έχει ως στόχο να παραπλανεί τα συστήματα ασφαλείας. Η αναγνώριση του είναι δύσκολη, ειδικά για συστήματα ασφαλείας που χρησιμοποιούν το signature του κάθε ιού, μιας και το κάθε αντίγραφο θα διαφέρει.

3.3.2 Worms

Τα worms όπως και οι ιοί έχουν τη δυνατότητα να αναπαράγονται μόνα τους. Η διαφορά τους όμως παρουσιάζεται στο ότι ενώ ο ιός προσπαθεί να προκαλέσει ζημιά στο σύστημα, το οποίο έχει μολύνει, το worm χρησιμοποιεί τις συνδέσεις δικτύου. Επίσης το worm μπορεί να υπάρξει και από μόνο του, ενώ ο ιός πρέπει να είναι προσκολλημένος σε κάποιο άλλο πρόγραμμα του συστήματος στο οποίο βρίσκεται. Τα worms όπως και οι ιοί παρουσιάζουν τα ίδια στάδια, με τη διαφορά ότι στη φάση αναπαραγωγής υπάρχει διαφορετική διαδικασία. Αρχικά γίνεται αναζήτηση άλλων συστημάτων από τους πίνακες δρομολόγησης, εγκαθιδρύεται σύνδεση με το σύστημα και δημιουργεί αντίγραφο του εαυτού του στο άλλο σύστημα και προκαλεί την εκτέλεση του. Ανάλογα με το πρόγραμμα στο οποίο επιτίθενται, χωρίζονται σε 5 τύπους: email, instant messaging, IRC, Internet και P2P [23].

- Το email worm εξαπλώνεται διαμέσου του ηλεκτρονικού ταχυδρομείου. Το worm στέλνει ένα αντίγραφο του εαυτού του ως επισύναψη σε ένα ηλεκτρονικό μήνυμα ή ένα σύνδεσμο(URL) σε ένα πόρο δικτύου(ιστοσελίδα με «μολυσμένο» αρχείο). Στην πρώτη περίπτωση ο κώδικας του worm ενεργοποιείται όταν η επισύναψη ανοίγεται, ενώ στη δεύτερη περίπτωση ενεργοποιείται όταν επιχειρείται πρόσβαση στο συγκεκριμένο σύνδεσμο. Τα email worms χρησιμοποιούν διάφορες μεθόδους για να αποστείλουν «μολυσμένα» μηνύματα, όπως είναι η απευθείας σύνδεση σε ένα SMTP εξυπηρετητή χρησιμοποιώντας τον κατάλογο ηλεκτρονικού ταχυδρομείου που είναι ενσωματωμένος στον κώδικα του worm, χρησιμοποιώντας τις υπηρεσίες του MS Outlook και άλλες. Επίσης τα email worms χρησιμοποιούν και διάφορες τρόπους για να βρουν ηλεκτρονικές διευθύνσεις στις οποίες θα αποστείλουν «μολυσμένα» μηνύματα, όπως είναι ο κατάλογος διευθύνσεων στο MS Outlook, το Windows Address Book(επιτρέπει σε χρήστες να έχουν μια ενιαία λίστα επαφών που χρησιμοποιείται από διάφορα προγράμματα), διάφορα αρχεία της μορφής .txt που περιέχουν ηλεκτρονικές διευθύνσεις ή από τα ηλεκτρονικά μηνύματα που υπάρχουν στα εισερχόμενα.

- Το Instant-Messaging (IM) worm εξαπλώνεται μέσω των προγραμμάτων για στιγμιαία μηνύματα όπως το (ICQ, MSN Messenger κτλ). Αποστέλλει ένα σύνδεσμο (URL), που περιέχει ένα αρχείο με τον κώδικα του worm, σε μια λίστα επαφών. Η τακτική που χρησιμοποιεί είναι η ίδια με εκείνη των Email Worms.

- Το IRC worm εξαπλώνεται διαμέσου της αναμετάδοσης ηλεκτρονικής συνομιλίας (Internet Relay Chat). Υπάρχουν 2 τρόποι «μόλυνσης». Ο ένας είναι η αποστολή ενός συνδέσμου (URL), που περιέχει ένα αρχείο με τον κώδικα του worm και ο άλλος η απευθείας αποστολή ενός μολυσμένου αρχείου σε ένα χρήστη της αναμετάδοσης ηλεκτρονικής συνομιλίας. Όμως με την απευθείας αποστολή πρέπει ο παραλήπτης να δεχτεί το αρχείο, να το αποθηκεύσει και να το ανοίξει (εκτελέσει).

- Το Internet Worm αναπαράγεται χρησιμοποιώντας τα δίκτυα των ηλεκτρονικών υπολογιστών. Δεν χρειάζεται κάποια ενέργειας του χρήστη για να εξαπλωθεί. Ο

συγκεκριμένος τύπος worm συνήθως ψάχνει κάποιες αδυναμίες στο λογισμικό που τρέχει σε διασυνδεδεμένους υπολογιστές για να εξαπλωθεί. Συγκεκριμένα στέλνει ένα πακέτο μέσω του δικτύου, το οποίο περιέχει τον κώδικα του worm, και έτσι εισέρχεται στον υπολογιστή του θύματος και ενεργοποιείται. Κάποιες φορές το συγκεκριμένο πακέτο περιέχει μόνο κάποιο μέρος του κώδικα worm, το οποίο θα κατεβάσει ένα αρχείο που θα περιέχει το κυρίως μέρος του worm που θα εκτελεστεί.

- Το P2P Worm διαδίδεται μέσω ενός peer-to-peer αρχείο που διαμοιράζεται μεταξύ των δικτύων. Διαδίδεται σε ένα δίκτυο P2P, εισάγοντας ένα αντίγραφο στον κατάλογο του διαμοιρασμού αρχείων και μετά η διάδοση γίνεται μέσω του P2P δικτύου όπως οποιουδήποτε άλλου αρχείου. Υπάρχουν πιο περίπλοκα P2P worm, τα οποία ανταποκρίνονται θετικά σε αναζητήσεις προτείνοντας ένα αντίγραφο του P2P worm ως ταύτιση.

3.3.3 Trojan Horse

Το όνομα του προέρχεται από την ελληνική ιστορία, από τον Τρωικό Πόλεμο και συγκεκριμένα αναφέρεται στον τρόπο με τον οποίο οι Έλληνες κατάφεραν να ξεγελάσουν τους Τρώες με τον Δούρειο Ίππο και να κατακτήσουν την Τροία. Αυτό οφείλεται στο γεγονός ότι το Trojan horse εκτός από κάποιο τρόπο λειτουργίας που παρουσιάζεται ότι έχει, έχει και ένα δεύτερο σκοπό, ο οποίος στηρίζεται σε κάποια κακόβουλη ενέργεια και δεν είναι εμφανής.

Συγκεκριμένα τα Trojan horses παρουσιάζονται στο χρήστη ως αυθεντικά λογισμικά, όμως στο παρασκήνιο εκτελούνται κάποιες άγνωστες ενέργειες, οι οποίες έχουν ως στόχο την υποκλοπή κάποιων πληροφοριών από το σύστημα και όχι την πραγματοποίηση της εκτέλεσης που παρουσιάζεται αρχικά στο χρήστη. Τα Trojan horses συλλέγουν διάφορες πληροφορίες από το σύστημα και τις αποστέλλουν πίσω στα άτομα, τα οποία έστειλαν από την αρχή τον κακόβουλο κώδικα. Επίσης μπορούν να χρησιμοποιηθούν για την καταστροφή αρχείων. Μπορεί να παρουσιάζεται στο χρήστη ότι εκτελεί κάποια χρήσιμη λειτουργία, ενώ στην ουσία καταστρέφει τα αρχεία του χρήστη χωρίς να είναι αντιληπτό. Υπάρχουν 16 τύποι Trojan horses ανάλογα με την λειτουργία την οποία έχουν, όπως είναι η εκτέλεση κώδικα, η διαγραφή αρχείων, η κλοπή συνθηματικών και άλλα, οι οποίοι είναι: Backdoors, PSW Trojans, Trojan

Clickers, Trojan Downloaders, Trojan Droppers, Trojan Proxies, Trojan Spies, Trojan Notifiers, ArcBombs, Rootkits, Trojan Banker, Trojan Mailfinder, Trojan GameThief, Trojan DDoS, Trojan IM και exploits [23].

- Το backdoor μπορεί να θεωρηθεί και ως διαφορετική κατηγορία ιών. Αναφέρεται εκτενώς πιο κάτω.

- Τα Trojan Droppers είναι σχεδιασμένα για να εγκαθιστούν μυστικά κακόβουλα προγράμματα, που είναι ενσωματωμένα στον κώδικα τους, στους ηλεκτρονικούς υπολογιστές των θυμάτων τους. Αυτού του είδους το πρόγραμμα αποθηκεύει μια ποσότητα αρχείων στο δίσκο του θύματος τα οποία εκτελεί χωρίς κάποια προειδοποίηση. Αυτά τα προγράμματα χρησιμοποιούνται για την μυστική εγκατάσταση Trojan Horses ή ιών και για να προστατεύσουν κακόβουλα προγράμματα από το να ανιχνευτούν από προγράμματα ασφαλείας.

- Το Trojan Proxy είναι πρόγραμμα το οποίο είναι σχεδιασμένο για να δίνει σε κακόβουλους χρήστες πρόσβαση σε διάφορους πόρους του διαδικτύου διαμέσου του ηλεκτρονικού υπολογιστή του θύματος.

- Το Trojan Clicker είναι σχεδιασμένο για να έχει πρόσβαση σε πόρους του διαδικτύου (ιστοσελίδες). Αυτό συμβαίνει είτε αποστέλλοντας εντολές στη μηχανή πλοήγησης είτε αντικαθιστώντας αρχεία συστήματος που παρέχουν σταθερές διευθύνσεις σε πόρους του διαδικτύου. Αυτά τα προγράμματα έχουν ως σκοπό την αύξηση των επισκέψεων σε ιστοσελίδες για την αύξηση παρουσίασης των διαφημίσεων του διαδικτύου, την πραγματοποίηση μιας επίθεσης Denial of Service σε ένα εξυπηρετητή και την οδήγηση θυμάτων σε ιούς ή Trojan Horses.

- Το Trojan-PSW (Password Stealing Ware) είναι για να είναι σχεδιασμένο για να κλέβει πληροφορίες σχετικά με τον λογαριασμό του χρήστη από «μολυσμένους» ηλεκτρονικούς υπολογιστές. Ψάχνει στα αρχεία του συστήματος για απόρρητες πληροφορίες.

- Τα Trojan-Banker είναι προγράμματα σχεδιασμένα για να κλέβουν δεδομένα λογαριασμών που αφορούν πληρωμές μέσω διαδικτύου. Αυτά τα δεδομένα αποστέλλονται πίσω στο δημιουργό του Trojan.

- Το Trojan-Mailfinder είναι σχεδιασμένο για να κλέβει ηλεκτρονικές διευθύνσεις από ένα ηλεκτρονικό υπολογιστή και να τις αποστέλλει στο δημιουργό του Trojan με διάφορες μεθόδους.

- Το Trojan-GameThief είναι σχεδιασμένο για να κλέβει δεδομένα που αφορούν λογαριασμούς του χρήστη για online παιχνίδια.

- Το Trojan-Ransom τροποποιεί δεδομένα σε ένα σύστημα έτσι ώστε ο χρήστης του να μην μπορεί να τα χρησιμοποιήσει, ή δεν επιτρέπει στο σύστημα να λειτουργήσει σωστά. Ο χρήστης θα λάβει ένα αίτημα για λεφτά για να μπορεί να χρησιμοποιήσει τα τροποποιήσει τα τροποποιημένα δεδομένα. Μόλις αυτό γίνει εφικτό ο δημιουργός του Trojan αποστέλλει ένα πρόγραμμα για επαναφορά των δεδομένων στην αρχική τους κατάσταση.

- Το Trojan-IM είναι σχεδιασμένο για να κλέβει δεδομένα που αφορούν λογαριασμούς του χρήστη που αφορούν προγράμματα στιγμιαίων μηνυμάτων (ICQ, MSN Messenger).

- Το Trojan-Downloader κατεβάζει και αντικαθιστά νέες εκδοχές κακόβουλων προγραμμάτων (Trojan και Adware) στα συστήματα των θυμάτων. Μόλις εγκατασταθούν εκτελούνται ή ενσωματώνονται σε μια λίστα προγραμμάτων που εκτελείται αυτόματα κατά την εκκίνηση του συστήματος.

- Το Trojan-Notifier αποστέλλει μηνύματα για να ενημερώνει τον κακόβουλο χρήστη όποτε ένας μολυσμένος ηλεκτρονικός υπολογιστής είναι online αποστέλλοντας του και τις απαραίτητες πληροφορίες.

- Το Trojan-Spy είναι προγράμματα που χρησιμοποιούνται για να κατασκοπεύουν τις ενέργειες ενός χρήστη. Οι πληροφορίες που συλλέγονται αποστέλλονται στον δημιουργό του Trojan.

- Τα Trojan-ArcBomb είναι συμπιεσμένα αρχεία που έχουν ως σκοπό την μείωση της επίδοσης ή την υπερφόρτωση του δίσκου με ένα μεγάλο μέγεθος αχρείαστων δεδομένων, όταν γίνεται προσπάθεια αποσυμπίεσης τους . Η υπερφόρτωση μπορεί να έχει ως αποτέλεσμα την κατάρρευση του συστήματος.

- Το Trojan-DDoS πραγματοποιεί επίθεση Denial of Service από ένα μολυσμένο υπολογιστή σε μια καθορισμένη από πριν διεύθυνση. Για να είναι επιτυχής μια επίθεση αποστέλλεται ένας μεγάλος αριθμός αιτημάτων στην μηχανή-στόχο.

- Τα exploits είναι προγράμματα που περιέχουν κώδικα που μπορεί να εκμεταλλευτεί τις αδυναμίες ενός λογισμικού που τρέχει στο «μολυσμένο» σύστημα. Αυτό έχει ως στόχο την εγκατάσταση κακόβουλου κώδικα.

- Τα rootkits είναι προγράμματα που έχουν ως στόχο την απόκρυψη συγκεκριμένων αντικειμένων και δραστηριοτήτων στο σύστημα, όπως registry keys, αρχεία, φάκελοι διεργασίες, όπως και κακόβουλος κώδικας Στόχος των συγκεκριμένων προγραμμάτων είναι να αποτρέψουν την ανίχνευση των κακόβουλων προγραμμάτων και έτσι να αυξήσουν τον χρόνο που μπορούν να εκτελούνται σε μια «μολυσμένη μηχανή».

3.3.4 Λογική Βόμβα

Η λογική βόμβα [18][14] είναι από τα πρώτα είδη απειλών που εμφανίστηκαν στα συστήματα. Είναι κώδικας ενσωματωμένος σε κάποιο πρόγραμμα που ενεργοποιείται όταν πραγματοποιηθούν συγκεκριμένες συνθήκες ή γεγονότα στο σύστημα. Μοιάζει με την λειτουργία των macro ιών. Μπορεί να είναι χρονική βόμβα, όταν η συνθήκη η οποία πρέπει να πραγματοποιηθεί είναι κάποια ώρα ή ημερομηνία. Όταν ενεργοποιηθεί μπορεί να προκαλέσει ζημιά στο σύστημα, να τροποποιήσει ή να διαγράψει δεδομένα ή ολόκληρα αρχεία.

3.3.5 Trapdoor ή Backdoor

Το trapdoor [14][18] αναφέρεται σε μια μυστική είσοδο μέσα σε ένα πρόγραμμα, το οποίο επιτρέπει σε κάποιο που γνωρίζει για αυτή την είσοδο να αποκτήσει πρόσβαση στο πρόγραμμα, χωρίς να είναι απαραίτητο να περάσει από τις συνηθισμένες διαδικασίες πρόσβασης. Επίσης μπορεί να αποκτήσει κάποια επιπλέον προνόμια.

Μπορεί να παραλάβει και να αποστείλει δεδομένα, να δημιουργήσει και να διαγράψει αρχεία, να εκτελέσει αρχεία ή ακόμα και να επανεκκινήσει την μηχανή. Αυτή η ξεχωριστή πρόσβαση του δίνει την ευκαιρία να μην αφήσει ίχνη για παρουσία κακομεταχείρισης του προγράμματος και είναι δύσκολο να ανιχνευτεί η παρουσία του. Αρχικά τα trapdoors χρησιμοποιούνταν από τους προγραμματιστές μιας εφαρμογής ως διευκόλυνση για την πρόσβασή τους στο σύστημα στην προσπάθεια τους να βρίσκουν λάθη στο πρόγραμμα και να το ελέγχουν. Επίσης χρησίμευε και στο ότι υπήρχε τρόπος πρόσβασης στο σύστημα όταν κάτι δεν λειτουργούσε σωστά με τη διαδικασία πρόσβασης. Η απειλή παρουσιάστηκε όταν χρήστες προσπαθούσαν να αποκτήσουν πρόσβαση σε προγράμματα και αρμοδιότητες για τις οποίες δεν ήταν εξουσιοδοτημένοι. Επίσης είναι δύσκολο να σχεδιαστούν έλεγχοι για εντοπισμό των trapdoors. Το trapdoor μπορεί να θεωρηθεί και ως τύπος Trojan Horse.

3.3.6 Zombie

Το zombie [18] είναι ένα είδος κακόβουλου κώδικα το οποίο χωρίς να γίνει αντιληπτό καταλαμβάνει ένα άλλο ηλεκτρονικό υπολογιστή που είναι ενωμένος με το διαδίκτυο και στέλνει επιθέσεις που είναι δύσκολο να γίνουν αντιληπτές. Συνήθως χρησιμοποιούνται σε επιθέσεις που αφορούν denial-of-service που έχουν ως στόχο ιστοσελίδες. Το zombie ενσωματώνεται σε πολύ μεγάλο αριθμό υπολογιστών τρίτων προσώπων. Στόχος του είναι να υπερφορτώσει μια συγκεκριμένη ιστοσελίδα – στόχο δημιουργώντας υπερβολική κίνηση στο διαδίκτυο.

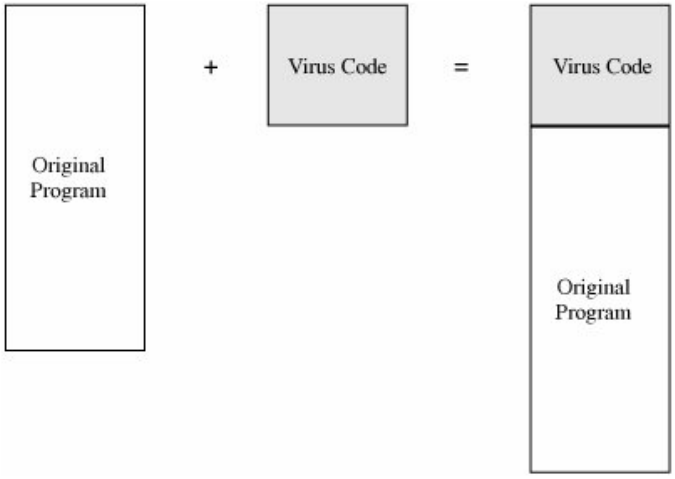
3.3.7 Rabbit

Το rabbit [14] είναι ένας ιός ή ένα worm το οποίο έχει τη δυνατότητα να πολλαπλασιάζεται χωρίς κάποιο όριο και έχει ως στόχο την εξόντωση κάποιου πόρου του συστήματος. π.χ Ένα rabbit δημιουργεί αντίγραφα του εαυτού του και τα αποθηκεύει στο σκληρό δίσκο μέχρι αυτός να γεμίσει εντελώς.

3.4 Τρόποι επισύναψης ιών σε προγράμματα

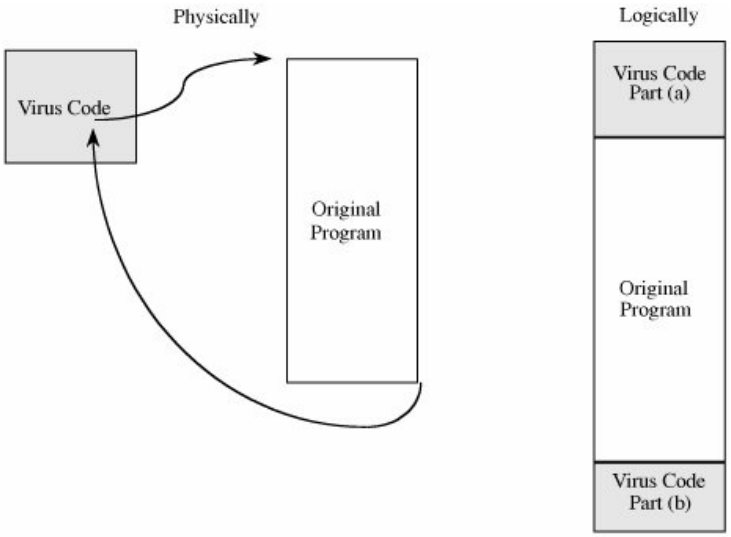
Ο ιός όπως αναφέρθηκε είναι πρόγραμμα, το οποίο έχει τη δυνατότητα να προσκολληθεί σε άλλα προγράμματα και να τα κάνει να συμπεριφέρονται και αυτά σαν ιούς. Συγκεκριμένα τροποποιεί τα προγράμματα με τέτοιο τρόπο ώστε αυτά να έχουν ένα αντίγραφο του. Υπάρχουν διάφοροι που ένας ιός μπορεί να προσκολληθεί σε ένα πρόγραμμα. Πιο κάτω παρουσιάζονται οι τρόποι επισύναψης ενός ιού [14].

Πρόσθεση ιού: Ο ιός προσθέτει ένα αντίγραφο του εαυτό του στο εκτελέσιμο πρόγραμμα πριν από την εκτέλεση της πρώτης εντολής του προγράμματος. Αρχικά εκτελείται ο ιός και μετά το τέλος της εκτέλεσης του, μπορεί να εκτελεστεί κανονικά το πρόγραμμα. Το «μολυσμένο» πρόγραμμα λειτουργεί σαν ο μεταφορέας του ιού. Είναι ο πιο απλός τρόπος επισύναψης ιού σε πρόγραμμα και ο πιο αποτελεσματικός, αφού ο δημιουργός ενός ιού δεν χρειάζεται να γνωρίζει κάτι για το πρόγραμμα. Ο συγκεκριμένος τρόπος επισύναψης είναι ο πιο συνηθισμένος.



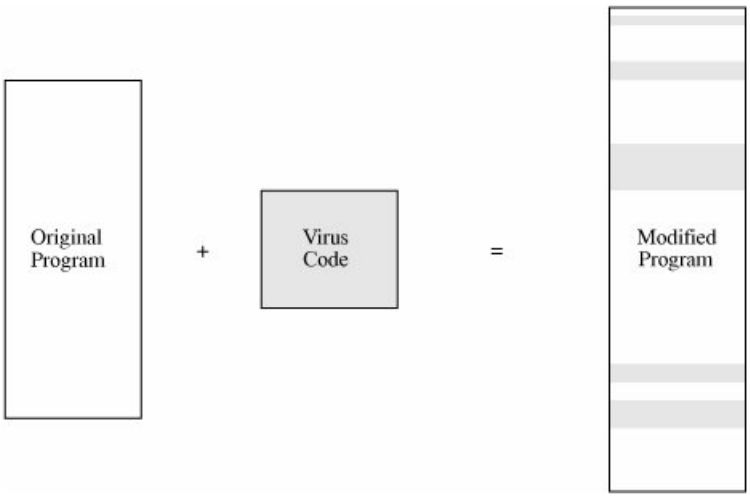
Σχήμα 3.9 Πρόσθεση ιού

Περικύκλωση προγράμματος: Ο ιός περικυκλώνει το πρόγραμμα. Το πρόγραμμα εκτελείται, αλλά ο ιός έχει τον έλεγχο στην αρχή και στο τέλος της εκτέλεσης του. Αυτός ο τρόπος προσκόλλησης παρουσιάζεται όταν δεν θέλει ο δημιουργός του ιού να γίνει αντιληπτός ο ιός. Έτσι ο ιός στο τέλος της εκτέλεσης του προγράμματος μπορεί να διαγράψει οποιαδήποτε πληροφορία μπορεί να φανερώσει την ύπαρξη του.



Σχήμα 3.10 Περικύκλωση προγράμματος

Ενσωμάτωση ιού και Αποκατάσταση: Ο συγκεκριμένος τύπος προσκόλλησης παρουσιάζεται όταν ο ιός ενσωματώνεται μέσα στον αρχικό κώδικα του προγράμματος. Ο ιός μπορεί να αντικαταστήσει και ολόκληρο το πρόγραμμα, αγνοώντας το αποτέλεσμα του προγράμματος και εκτελώντας μόνο τον ιό. Σε αυτή την περίπτωση υπάρχει απώλεια του αρχικού προγράμματος και ο δημιουργός του ιού πρέπει να γνωρίζει τη δομή του προγράμματος για να μπορεί ο ιός να ενσωματωθεί.



Σχήμα 3.11 Ενσωμάτωση ιού

Κεφάλαιο 4

Τρόποι εξασφάλισης ασφάλειας

Εισαγωγή

4.1 Τρόποι Προστασίας

4.2 Συστήματα Ανίχνευσης Εισβολής

4.3 Παρουσίαση μοντέλου ανίχνευσης εισβολής

4.4 Προηγούμενες μελέτες για ανίχνευση ανωμαλιών

Εισαγωγή

Στο προηγούμενο κεφάλαιο έγινε αναφορά στις διάφορες επιθέσεις που μπορούν να γίνουν σε ένα σύστημα, στους τύπους εισβολέων και στους τύπους κακόβουλου κώδικα που υπάρχουν και πώς αυτοί επισυνάπτονται στα προγράμματα.

Σε αυτό το κεφάλαιο θα αναφερθούν οι τρόποι προστασίας ενός συστήματος και οι τρόποι ανίχνευσης κακόβουλου κώδικα με στόχο ένα σύστημα να μπορεί να αντεπεξέλθει στις διάφορες επιθέσεις.

4.1 Τρόποι Προστασίας συστημάτων

Ένα σύστημα χρειάζεται προστασία ώστε να μπορεί να ικανοποιεί τους στόχους που παρουσιάστηκαν σε προηγούμενο κεφάλαιο. Ο καλύτερος τρόπος προστασίας ενός συστήματος είναι η απομόνωση του, δηλαδή να μην έχει επικοινωνία με άλλα συστήματα για την ανταλλαγή δεδομένων. Επίσης να μην επιτρέπει την τροποποίηση προγραμμάτων και δεδομένων. Ένα τέτοιο σύστημα όμως δεν μπορεί να υπάρχει, αφού τα πιο πάνω είναι αδύνατον να μην συμβαίνουν. Έτσι είναι απαραίτητο να υπάρχουν κάποιοι άλλοι τρόποι που να παρέχουν προστασία σε ένα σύστημα.

Πιο κάτω παρουσιάζονται τρόποι προστασίας των δεδομένων και του συστήματος που αφορούν όλα τα επίπεδα αρχιτεκτονικής δικτύου .

4.1.1 Κρυπτογραφία

Η κρυπτογραφία [11] [12] είναι ένας τρόπος προστασίας των δεδομένων κατά την αποστολή τους, ώστε κάποιος εισβολέας να μην μπορεί να υποκλέψει πληροφορίες. Σημαντικό όμως είναι όπως ο παραλήπτης γνωρίζει τον τρόπο να ανακτήσει τα αρχικά δεδομένα από τα κρυπτογραφημένα. Αποτελείται από τα στάδια της κρυπτογράφησης και αποκρυπτογράφησης. Χρησιμοποιείται και στα Virtual Private Networks (VPN) και αποτελεί ένα σημαντικό μέρος του πως αυτά λειτουργούν.

Η κρυπτογράφηση και αποκρυπτογράφηση βασίζονται σε κλειδιά. Τα κλειδιά είναι μυστικές τιμές και διαφέρουν σε μέγεθος, ανάλογα με την ασφάλεια που πρέπει να έχει το μήνυμα. Μπορεί να είναι συμμετρικά ή ασύμμετρα.

Το αρχικό κείμενο πριν κρυπτογραφηθεί ονομάζεται cleartext ή plain text και το κρυπτογραφημένο ciphertext. Η μέθοδος που χρησιμοποιείται για την κρυπτογράφηση/αποκρυπτογράφη είναι αλγόριθμοι κρυπτογράφησης. Πιο κάτω θα γίνει αναφορά σε μερικούς.

4.1.1.1 Συμμετρικό – Διαμοιραζόμενο κλειδί

Το συμμετρικό ή διαμοιραζόμενο κλειδί χρησιμοποιεί την ίδια τιμή κλειδιού για την κρυπτογράφηση και την αποκρυπτογράφηση. Αυτό προϋποθέτει την εκ των προτέρων μυστική ανταλλαγή του κλειδιού. Αυτή η μέθοδος είναι γρήγορη γιατί δεν είναι όσο περίπλοκη όσο οι μέθοδοι με τα ασύμμετρα κλειδιά, όμως υπάρχει το μειονέκτημα ότι η ανταλλαγή του κλειδιού πρέπει να γίνει μυστικά και πρέπει κάποιος να είναι σίγουρος για την αυθεντικότητα του συνομιλητή του. Η ανταλλαγή προϋποθέτει την ύπαρξη κάποιου ασφαλούς καναλιού για να μεταδοθεί το κλειδί, όμως κάτι τέτοιο δεν είναι δυνατόν. Έτσι η εμπιστευτικότητα των δεδομένων είναι εξασφαλισμένη εφόσον διατηρείται μυστικό το κλειδί. Υπάρχουν διάφοροι αλγόριθμοι κρυπτογράφησης αυτού του είδους όπως ο Data Encryption Standard(DES), ο 3DES, ο Rijndael, ο Blowfish και ο International Data Encryption Algorithm (IDEA). Ο καθένας παρέχει ένα συγκεκριμένο βαθμό ασφαλείας. Για παράδειγμα ο DES που χρησιμοποιεί ένα 56-bit κλειδί έχει αποδειχτεί ότι κάποιος μπορεί εύκολα να τον σπάσει, ενώ αντίθετα ο 3DES που χρησιμοποιεί 3 διαφορετικά 56-bit κλειδιά είναι πιο ασφαλές. Η καλύτερη λύση θεωρείται ο Rijndael που όμως δεν χρησιμοποιείται ευρέως.

4.1.1.2 Ασύμμετρο – Διαμοιραζόμενο Ιδιωτικό κλειδί

Οι αλγόριθμοι ασύμμετρου κλειδιού λειτουργούν διαφορετικά από ότι η προηγούμενη κατηγορία. Χρησιμοποιούνται 2 διαφορετικά κλειδιά, ένα διαμοιραζόμενο και ένα ιδιωτικό. Το διαμοιραζόμενο κλειδί χρησιμοποιείται για την κρυπτογράφηση και το μυστικό για την αποκρυπτογράφηση. Η πολυπλοκότητα της συγκεκριμένης μεθόδου έγκειται στο γεγονός ότι ενώ το ciphertext μπορεί να γίνει από οποιονδήποτε έχει το διαμοιραζόμενο κλειδί, ο μόνος που μπορεί να το αποκρυπτογραφήσει είναι εκείνος που έχει το ιδιωτικό κλειδί. Όμως η συγκεκριμένη μέθοδος είναι πιο αργή και χρειάζεται περισσότερη επεξεργασία. Παραδείγματα αλγορίθμων αυτής της κατηγορίας είναι ο Pretty Good Privacy (PGP), ο Diffie Hellman και ο RSA public-key. Ο PGP αναφέρεται σε επικοινωνία ατόμων που μπορεί να μην έχουν συναντηθεί ποτέ. Χρησιμοποιεί εξυπηρετητές, όπου κάποιος έχει την επιλογή να δημοσιοποιήσει το διαμοιραζόμενο κλειδί και κάποιος που θέλει να επικοινωνήσει μαζί του μπορεί να το αναζητήσει και μετά να αρχίσει η μεταξύ τους επικοινωνία. Πολλές φορές είναι απαραίτητη και η χρήση κάποιου κωδικού για την εξασφάλιση του κλειδιού. Ο Diffie Hellman χρησιμοποιείται στα VPN και αρχικά χρησιμοποιεί ένα συμμετρικό αλγόριθμο όπως ο DES και 3DES για να μεταφερθούν αρχικά οι διαμοιραζόμενες πληροφορίες και οι ρυθμίσεις για να μπορεί να γίνει η σύνδεση.

Ο συνδυασμός των 2 πιο πάνω τύπων αλγορίθμων αποδεικνύεται πολύ αποτελεσματικός. Χρησιμοποιούνται οι συμμετρικοί αλγόριθμοι που μπορούν να παρέχουν γρήγορη κρυπτογράφηση και οι ασύμμετροι αλγόριθμοι για την ασφαλή ανταλλαγή των κλειδιών.

4.1.1.3 Ψηφιακές υπογραφές και hash αλγόριθμοι

Οι πιο πάνω μέθοδοι κρυπτογράφησης που αναφέρθηκαν πιο πάνω παρέχουν την εμπιστευτικότητα μεταξύ όσων επικοινωνούν μεταξύ τους. Η κρυπτογραφία όμως χρησιμοποιείται και για να αποδείξει την ακεραιότητα των δεδομένων, την αυθεντικότητά τους. Για αυτό το σκοπό χρησιμοποιούνται οι ψηφιακές υπογραφές και οι hash συναρτήσεις.

Οι ψηφιακές υπογραφές χρησιμοποιούνται για να αποδείξουν ότι τα δεδομένα προέρχονται από ένα συγκεκριμένο πρόσωπο και η απόδειξη της αυθεντικότητας

παρέχεται με την χρήση της κρυπτογραφίας. Δεν χρησιμοποιείται η συμμετρική κρυπτογράφηση λόγω του μειονεκτήματος της ανταλλαγής του κλειδιού, αλλά η ασύμμετρη. Μπορεί να γίνει κρυπτογράφηση με τη χρήση του μυστικού κλειδιού και η αποκρυπτογράφηση με τη χρήση του διαμοιραζόμενου, αποδεικνύοντας ότι τα δεδομένα προέρχονται από το άτομο που έχει τα κλειδιά, αφού οποιαδήποτε αλλαγή στα κρυπτογραφημένα δεδομένα δεν οδηγεί στο σωστό αρχικό κείμενο. Η χρήση όμως του διαμοιραζόμενου κλειδιού είναι ένα μειονέκτημα, αφού οποιοσδήποτε μπορεί να το χρησιμοποιήσει για να αποκρυπτογραφήσει τα δεδομένα προκαλώντας έτσι την απώλεια της εμπιστευτικότητας. Επίσης στις ψηφιακές υπογραφές παρουσιάζεται και το μειονέκτημα του ασύμμετρου κλειδιού που χρειάζεται αρκετή ώρα επεξεργασίας. Αυτή η καθυστέρηση δεν είναι πρακτική.

Οι hash αλγόριθμοι χρησιμοποιούνται για να καλύπτονται τα πιο πάνω μειονεκτήματα. Δημιουργούν ένα «αποτύπωμα» (fingerprint) από κάποιο δεδομένο. Χρησιμοποιώντας αυτό το «αποτύπωμα» που ονομάζεται hash ή message direct μπορεί να αποδειχτεί ότι το δεδομένο δεν τροποποιήθηκε. Έτσι αποδεικνύεται η ακεραιότητα της πληροφορίας χωρίς τον έλεγχο της ανά bit. Οι 2 πιο γνωστοί αλγόριθμοι hash είναι ο Message Direct 5 (MD5) και ο Secure Hash Algorithm (SHA-1), εκ των οποίων ο SHA-1 μπορεί να παρέχει μεγαλύτερη ασφάλεια, ενώ ο MD5 είναι πιο γρήγορος.

4.1.2 Εξακρίβωση Αυθεντικότητας

Η εξακρίβωση αυθεντικότητας είναι η διαδικασία κατά την οποία εξακριβώνεται η ταυτότητα κάποιου, δηλαδή εξακριβώνεται ότι είναι αυτός που ισχυρίζεται ότι είναι. Όταν 2 πρόσωπα επικοινωνούν μεταξύ τους, ανταλλάζοντας δεδομένα είναι απαραίτητο να γίνεται η συγκεκριμένη ασφάλεια για να παρέχεται ασφάλεια στα πρόσωπα.

Η συγκεκριμένη διαδικασία πρέπει να γίνεται την ώρα που αρχίζει η επικοινωνία. Έτσι στην αρχή της επικοινωνίας πρέπει πρώτα να χρησιμοποιείται ένα πρωτόκολλο εξακρίβωσης αυθεντικότητας πριν αρχίσει η χρήση άλλων πρωτοκόλλων για την ανταλλαγή δεδομένων. Υπάρχουν διάφορα εκδοχές πρωτοκόλλων που χρησιμοποιούνται.

Αρχικά η πιο απλή μορφή πρωτοκόλλου είναι όταν το ένα πρόσωπο εκ των δύο αποστέλλει ένα μήνυμα και όπου λέει ποιος είναι. Το άλλο πρόσωπο όμως που αποστέλλει το μήνυμα δεν είναι σίγουρο ότι όντως είναι αυτό που ισχυρίζεται.

Για την σωστή και έγκυρη εξακρίβωση της αυθεντικότητας χρησιμοποιούνται πιο περίπλοκοι αλγόριθμοι, οι οποίοι χρησιμοποιούν την κρυπτογραφία. Ένα παράδειγμα εξακρίβωσης της αυθεντικότητας είναι το εξής. Ο ένας εκ των δύο προσώπων που λαμβάνουν μέρος σε μια επικοινωνία αποστέλλει μήνυμα στον άλλο για να αρχίσει η μεταξύ τους συνομιλία. Τότε ο δεύτερος αποστέλλει ένα μήνυμα, το οποίο περιέχει ένα αριθμό, ο οποίος χρησιμοποιείται για να καθορίζεται η σειρά των μηνυμάτων και να αποφεύγεται η αναπαραγωγή από κάποιο εισβολέα. Μετά το πρώτο πρόσωπο κρυπτογραφεί το μήνυμα χρησιμοποιώντας το ιδιωτικό του κλειδί και το αποστέλλει στο δεύτερο. Το δεύτερο άτομο με τη σειρά του χρησιμοποιεί το διαμοιραζόμενο κλειδί για να αποκρυπτογραφήσει το μήνυμα του πρώτου. Έτσι αν το μήνυμα περιέχει τον αριθμό που απέστειλε τότε το δεύτερο άτομο εξακριβώνει την ταυτότητα του άλλου προσώπου.

Η κρυπτογραφία είναι ένας από τους πιο βασικούς τρόπους προστασίας. Η εξακρίβωση αυθεντικότητας χρησιμοποιεί την κρυπτογραφία και έτσι σε συνδυασμό χρησιμοποιούνται για να παρέχουν προστασία στα δεδομένα που διακινούνται σε ένα δίκτυο. Υπάρχει η ανάγκη παροχής προστασίας τόσο στα υψηλότερα επίπεδα, όσο και στα χαμηλότερα. Στα υποκεφάλαια που ακολουθούν θα περιγραφούν τρόποι προστασίας που μπορούν να παρατηρηθούν σε κάθε επίπεδο [11].

4.1.3 Προστασία Ηλεκτρονικού Ταχυδρομείου

Η μέθοδος προστασίας που θα περιγραφεί πιο κάτω αναφέρεται στο επίπεδο εφαρμογών και να αναφέρεται στους τρόπους που μια εφαρμογή μπορεί να είναι ασφαλής, δηλαδή να προσφέρει εμπιστευτικότητα, ακεραιότητα και αυθεντικότητα [11].

Αρχικά πρέπει να καθοριστεί πώς θα ικανοποιηθούν οι πιο πάνω στόχοι. Η εμπιστευτικότητα ικανοποιείται με την εξασφάλιση ότι η ανταλλαγή δεδομένων δεν φτάνει στα χέρια τρίτων. Απαραίτητη είναι η εξασφάλιση αυθεντικότητας των

προσώπων που επικοινωνούν, ότι τα πρόσωπα είναι όντως αυτά που υποστηρίζουν ότι είναι. Τέλος πρέπει να εξασφαλιστεί η ακεραιότητα των δεδομένων.

Η εμπιστευτικότητα μπορεί να εξασφαλιστεί μέσω της κρυπτογράφησης και αποκρυπτογράφησης των δεδομένων. Το συμμετρικό κλειδί είναι μια λύση, όμως παρουσιάζει το πρόβλημα που αναφέρθηκε πιο πάνω. Θεωρείται καλύτερη λύση τη μέθοδο RSA public key, όπου γνωστοποιείται το διαμοιραζόμενο κλειδί του παραλήπτη, ο αποστολέας κρυπτογραφεί τα δεδομένα με αυτό και τα αποστέλλει και τότε ο παραλήπτης τα αποκωδικοποιεί χρησιμοποιώντας το ιδιωτικό κλειδί του. Αυτή η μέθοδος όμως δεν θεωρείται αποδοτική, ειδικά για μεγάλα μηνύματα. Έτσι ως καλύτερη λύση κρυπτογράφησης θεωρείται ο αλγόριθμος PGP, που επεξηγήθηκε πιο πάνω και προσφέρει ασφάλεια στην ανταλλαγή των κλειδιών.

Για την εξασφάλιση της αυθεντικότητας και της ακεραιότητας των δεδομένων χρησιμοποιούνται οι ψηφιακές υπογραφές και hash αλγόριθμοι.

Έχοντας αναφέρει τα πιο πάνω, μπορεί να θεωρηθεί ότι αυτό το μοντέλο ασφάλειας για το ηλεκτρονικό ταχυδρομείο μπορεί να ικανοποιήσει τους στόχους της ασφάλειας και να παρέχει ικανοποιητικά αποτελέσματα.

4.1.4 Προστασία TCP συνδέσεων

Σε αυτό το σημείο αναφέρονται τρόποι ενός πιο χαμηλού επιπέδου της αρχιτεκτονικής δικτύου, το οποίο αναφέρεται στο TCP πρωτόκολλο [11]. Χρησιμοποιείται η κρυπτογραφία για την προστασία των TCP συνδέσεων για την παροχή εμπιστευτικότητας, ακεραιότητας και αυθεντικότητας. Η εκδοχή του TCP με ενσωματωμένη τη μέθοδο της κρυπτογραφίας είναι το Secure Sockets Layer(SSL), όπως και μια άλλη τροποποιημένη εκδοχή του το Transport Layer Security (TLS).

Το SSL σχεδιάστηκε αρχικά από το Netscape, αλλά απέκτησε μεγάλη φήμη και χρησιμοποιείται σχεδόν από όλους τους εξυπηρετητές διαδικτύου και στα προγράμματα πλοήγησης. Η ενσωμάτωση του SSL είναι αρκετά σημαντική. Χωρίς εμπιστευτικότητα κάποιος μπορεί να έχει πρόσβαση σε υψίστης σημασίας δεδομένα που αποστέλλονται με αποτέλεσμα να τα χρησιμοποιήσει εις βάρους του χρήστη. Χωρίς την παροχή ακεραιότητας, ένας εισβολέας μπορεί να τροποποιήσει τα δεδομένα που

αποστέλλονται. Επίσης χωρίς την εξακρίβωση αυθεντικότητας μπορεί ένας εισβολέας να ισχυριστεί ότι είναι κάποιος άλλος με αποτέλεσμα ένας χρήστης να αποστέλλει πληροφορίες σε αυτόν.

Το πρωτόκολλο SSL χρησιμοποιείται για κυρίως για την προστασία εφαρμογών που χρησιμοποιούν το πρωτόκολλο HTTP και μπορεί να χρησιμοποιηθεί σε οποιαδήποτε εφαρμογή που χρησιμοποιεί το πρωτόκολλο TCP. Κατά την αρχή της επικοινωνίας τα πρόσωπα που λαμβάνουν μέρος μπορούν να καθορίσουν τον αλγόριθμο κρυπτογραφίας που θα χρησιμοποιήσουν και μετά κρυπτογραφούν τα δεδομένα που αποστέλλουν.

4.1.5 Προστασία επιπέδου δικτύου (IP)

Η ασφάλεια που παρέχεται στο επίπεδο δικτύου, είναι ένα πρωτόκολλο γνωστό ως IPsec [11], το οποίο αποτελείται από ένα σύνολο πρωτοκόλλων. Η ύπαρξη του είναι αρκετά σημαντική.

Αρχικά η εμπιστευτικότητα εξασφαλίζεται με την κρυπτογράφηση των IP διαγραμμάτων. Έτσι κάθε φορά που αποστέλλεται ένα διάγραμμα IP, πρώτα κρυπτογραφείται με τη χρησιμοποίηση διάφορων αλγόριθμων. Αν εφαρμοζόταν σε όλα τα διαγράμματα που κυκλοφορούν στα δίκτυα τότε δεν θα μπορούσαν κάποια τρίτα άτομα που «ακούνε» το δίκτυο να έχουν πρόσβαση σε αυτά. Επίσης κάποιος που παραλαμβάνει ένα IP διάγραμμα πρέπει να είναι σίγουρος για την αυθεντικότητα του αποστολέα.

Αυτό γίνεται με τα δύο κύρια πρωτόκολλα που υπάρχουν στο IPsec, το Authentication Header(AH) και το Encapsulation Security Payload(ESP) πρωτόκολλο. Το πρωτόκολλο AH παρέχει εξακρίβωση αυθεντικότητας της πηγής και ακεραιτότητα, ενώ το πρωτόκολλο ESP παρέχει και εμπιστευτικότητα. Το πρωτόκολλο ESP είναι πιο πολύπλοκο και χρειάζεται περισσότερη επεξεργασία.

Τα 2 πρωτόκολλα δημιουργούν μια λογική σύνδεση δικτύου, το οποίο ονομάζεται Security Association (SA), το οποίο δεν είναι αμφίδρομο. Για κάθε χρήστη που θέλει να αποστείλει με ασφάλεια τα δεδομένα, δημιουργείται ένα διαφορετικό SA. Το κάθε SA αναγνωρίζεται από μια 32-bit παράμετρο, Security Parameter Index (SPI), που

περιέχεται στην επικεφαλίδα όλων των διαγραμμάτων του IPsec. Όλα όσα αφορούν ένα συγκεκριμένα SA έχουν το ίδιο SPI.

4.1.6 Ασύρματη Ασφάλεια

Η ασφάλεια των ασύρματων δικτύων είναι ένα σημαντικό θέμα, αφού τα ραδιοκύματα που μεταφέρουν τα πακέτα πληροφορίας μπορούν να μεταδοθούν και πέρα από την ασύρματη βάση.

4.1.6.1 Wired Equivalent Privacy (WEP)

Στο πρωτόκολλο IEEE 802.11 χρησιμοποιείται μέθοδος συμμετρικού κλειδιού, το Wired Equivalent Privacy (WEP) πρωτόκολλο για την προστασία των δεδομένων που διακινούνται σε ένα ασύρματο δίκτυο. Παρέχει αυθεντικότητα και κρυπτογράφηση δεδομένων μεταξύ ενός προσώπου και του σημείου πρόσβασης στο ασύρματο δίκτυο. Ο τρόπος που συμφωνούν για το κλειδί γίνεται με μια μέθοδο εκτός δικτύου. Η διαδικασία αυθεντικότητας περιέχει 4 στάδια. Αρχικά ένας ασύρματος κόμβος ζητά αυθεντικότητα από ένα σημείο πρόσβασης. Το σημείο πρόσβασης απαντά στο αίτημα με ένα 128-byte εξειδικευμένο μήνυμα. Τότε ο ασύρματος κόμβος κωδικοποιεί το μήνυμα χρησιμοποιώντας το συμμετρικό κλειδί. Το σημείο πρόσβασης το αποκωδικοποιεί και αν ταιριάζει με το αρχικό, τότε ο κόμβος αποκτά αυθεντικότητα. Το συγκεκριμένο πρωτόκολλο προστασίας περιλαμβάνει αρκετές αδυναμίες που μπορεί κάποιος να τις εκμεταλλευτεί και να επιτεθεί στο δίκτυο.

4.1.6.2 IEEE 802.11i

Η παρουσία ενός πρωτόκολλου με καλύτερους μηχανισμούς ασφάλειας ήταν απαραίτητη και έτσι σχεδιάστηκε το πρωτόκολλο IEEE802.11i, το οποίο παρέχει πιο δυνατές μορφές κρυπτογραφίας, διευρυμένους μηχανισμούς απόκτησης αυθεντικότητας και ένα μηχανισμό διανομής του κλειδιού. Πέρα από τον ασύρματο κόμβο και το σημείο πρόσβασης παρουσιάζεται και ένας εξυπηρετητής αυθεντικότητας, που επικοινωνεί με το σημείο πρόσβασης. Ο εξυπηρετητής μπορεί να είναι ενωμένος με διάφορα σημεία πρόσβασης ενώ οι αποφάσεις για την αυθεντικότητα λαμβάνονται

συγκεντρωτικά. Με αυτόν τον τρόπο περιορίζεται η πολυπλοκότητα ενός σημείου πρόσβασης και το κόστος. Το συγκεκριμένο πρωτόκολλο περιλαμβάνει 4 φάσεις.

Αρχικά είναι η φάση ανακάλυψης, όπου το σημείο πρόσβασης «διαφημίζει» την παρουσία του και τις μορφές αυθεντικότητας και κρυπτογραφίας που παρέχει σε ένα ασύρματο κόμβο. Ο ασύρματος κόμβος επικοινωνεί με το σημείο πρόσβασης και καθορίζει τις ρυθμίσεις που επιθυμεί και αν και υπάρχει επικοινωνία με το σημείο πρόσβασης δεν έχει διευκρινιστεί η αυθεντικότητα του για να επικοινωνήσει με κάποιο κόμβο του δικτύου.

Ακολουθεί η φάση της αυθεντικότητας και της παραγωγής ενός κυρίως κλειδιού. Αποφασίζεται η αυθεντικότητα του κινητού κόμβου μέσω του εξυπηρετητή αυθεντικότητας και το σημείο πρόσβασης λειτουργεί σαν ενδιάμεσος κόμβος εφαρμόζοντας μια διαδικασία αναγνώρισης και παράγεται και ένα κλειδί που είναι γνωστό μεταξύ τους.

Μετά είναι η φάση για την παραγωγή ενός pairwise κυρίως κλειδιού. Το κυρίως κλειδί που παράγεται στην προηγούμενη φάση είναι γνωστό μόνο μεταξύ του εξυπηρετητή και του ασύρματου κόμβου, οι οποίο το χρησιμοποιούν για να καθορίσουν ένα άλλο κλειδί το οποίο κοινοποιείται στο σημείο πρόσβασης. Αυτό το σημείο καλύπτει μια από τις αδυναμίες του WEP πρωτοκόλλου.

Τέλος ακολουθεί η φάση των προσωρινών κλειδιών, όπου ασύρματος κόμβος και σημείο πρόσβασης παράγουν προσωρινά κλειδιά που θα χρησιμοποιηθούν για κρυπτογράφηση των δεδομένων για την μεταξύ τους επικοινωνία.

4.1.6.3 Wireless Access Protocol (WAP)

Το WAP είναι ένα άλλο πρωτόκολλο που παρέχει προστασία στα ασύρματα δίκτυα και στηρίζεται στο WEP πρωτόκολλο. Χρησιμοποιεί την κρυπτογραφία για την κωδικοποίηση των δεδομένων, ενώ ταυτόχρονα πραγματοποιεί ελέγχους για να σιγουρεύεται ότι το κλειδί του δικτύου δεν έχει τροποποιηθεί. Επίσης ελέγχει για την αυθεντικότητα των ασύρματων κόμβων και εξασφαλίζει ότι μόνο εξουσιοδοτημένα πρόσωπα έχουν πρόσβαση στο δίκτυο.

Το WPA2 προσφέρει περισσότερη ασφάλεια από το WPA και βασίζεται στο 802.11i πρωτόκολλο. Υπάρχει διανομή διαφορετικών κλειδιών σε κάθε χρήστη. Επιπλέον, μπορεί να χρησιμοποιηθεί και ένα διαμοιραζόμενο κλειδί όπου όλοι οι χρήστες λαμβάνουν τον ίδιο τρόπο πρόσβασης. Το WPA2 είναι η προτιμότερη μέθοδος προστασίας που θα χρησιμοποιηθεί στο πρωτόκολλο 802.11n.

4.1.7 Firewalls

Η ύπαρξη του firewall στα συστήματα είναι πολύ σημαντική. Πλέον όλοι είναι οργανισμοί είναι διασυνδεδεμένοι με το διαδίκτυο, αφού τα πλεονεκτήματα που παρέχει είναι πολλά. Όμως είναι έντονη και η απειλή που υπάρχει από πρόσωπα εκτός του οργανισμού που προσπαθούν να φτάσουν στο δίκτυο του και να εκμεταλλευτούν τους πόρους του. Σε ένα οργανισμό κάθε ηλεκτρονικός υπολογιστής και εξυπηρετητής του δικτύου έχει κάποιους μηχανισμούς ασφάλειας, όμως κάτι τέτοιο δεν είναι πρακτικό, αφού σε περίπτωση που ανακαλυφθεί μια αδυναμία στο σύστημα ασφάλειας πρέπει όλα τα επηρεαζόμενα μέλη του δικτύου πρέπει να αναβαθμιστούν. Ο ρόλος του firewall αρχίζει εδώ.

Το firewall είναι ένας συνδυασμός υλικού και λογισμικού και στόχος του είναι η απομόνωση του δικτύου του οργανισμού από το Διαδίκτυο, με σκοπό να το προστατέψει από τις επιθέσεις μέσω Διαδικτύου. Επιτρέπει μόνο σε ορισμένα πακέτα να εισέλθουν στο δίκτυο του οργανισμού, ενώ εμποδίζει την είσοδο άλλων.

Ο υπεύθυνος του δικτύου του οργανισμού καθορίζει τις ρυθμίσεις του firewall ανάλογα με την πολιτική ασφάλειας που ακολουθεί ο οργανισμός. Έτσι μπορεί να ελέγχει και να καθορίζει την πρόσβαση προσώπων εκτός οργανισμού σε πόρους του δικτύου, ελέγχοντας την κίνηση των πακέτων από και προς τους πόρους του δικτύου. Επίσης ανάλογα με την πολιτική ασφάλειας του δικτύου μπορεί να δημιουργηθούν πολλαπλά επίπεδα firewall.

Το firewall είναι σχεδιασμένο για να ικανοποιεί 3 στόχους:

- Όλη η κίνηση που παρατηρείται εκτός δικτύου προς αυτό και αντίθετα πρέπει να περνά διαμέσου του firewall.

- Μόνο εξουσιοδοτημένη κίνηση, όπως καθορίζεται από την πολιτική ασφάλειας, επιτρέπεται να περάσει στο δίκτυο.
- Το firewall είναι άτρωτο απέναντι σε κάποια επίθεση.

Υπάρχουν 3 διαφορετικοί τύποι firewall:

- Packet filter: Έχει ένα δρομολογητή που ενώνει το δίκτυο με το Διαδίκτυο και εφαρμόζει ένα σύνολο κανόνων σε κάθε εισερχόμενο ή εξερχόμενο IP πακέτο. Ελέγχει το κάθε διάγραμμα και καθορίζει κατά πόσο πρέπει να του επιτραπεί να περάσει ή αν θα απορριφθεί βάση των κανόνων. Οι αποφάσεις στηρίζονται στη IP διεύθυνση της πηγής και του προορισμού, στο τύπο πρωτοκόλλου, στις διευθύνσεις πηγής και προορισμού που αφορούν το επίπεδο μεταφοράς και στη διεπιφάνεια των δρομολογητών.
- Stateful Packet Filters: Στον προηγούμενο τύπο firewall η απόφαση για το κάθε πακέτο γινόταν ατομικά. Σε αυτό τον τύπο λαμβάνονται υπόψη και τα περιεχόμενα του πιο ψηλού επιπέδου. Δηλαδή προσπαθεί να εντοπίσει όλες τις TCP συνδέσεις τις οποίες καταχωρεί σε ένα πίνακα για να αποφασίσει. Επιτρέπει την είσοδο σε πακέτα που ταιριάζουν με τα στοιχεία που υπάρχουν στον πίνακα που δημιουργεί.
- Application-Level Gateway: Ονομάζεται και εξυπηρετητής proxy. Σε αυτού του είδους το firewall πέρα από τις επικεφαλίδες IP/TCP/UDP, οι αποφάσεις στηρίζονται στα δεδομένα των εφαρμογών. Η είσοδος σε κάθε εφαρμογή μπορεί να ρυθμιστεί έτσι ώστε να υποστηρίζει συγκεκριμένα χαρακτηριστικά για μια εφαρμογή. Αν ο κώδικας proxy που δίνει ένα πρόσωπο δεν μπορεί να ταυτιστεί με μια συγκεκριμένη εφαρμογή, τότε σημαίνει ότι δεν μπορεί το πακέτο να εγκριθεί και απορρίπτεται. Αυτός ο τύπος firewall παρέχει μεγαλύτερη ασφάλεια από τα άλλα, αλλά προσθέτει επιπλέον φόρτο για την επεξεργασία και χρειάζεται διαφορετικό gateway για κάθε εφαρμογή.

Γενικά το firewall παρέχει ένα σημείο του δικτύου που μπορεί να κρατήσει μη εξουσιοδοτημένους χρήστες εκτός του δικτύου, να αποτρέψει υπηρεσίες που έχουν αδυναμίες από το να εισέλθουν ή να εξέλθουν από το σύστημα και να παρέχει ασφάλεια από διάφορες επιθέσεις εξαπάτησης IP και από επιθέσεις στις δρομολογήσεις. Ουσιαστικά είναι ένα μέρος που μπορούν να παρακολουθούνται

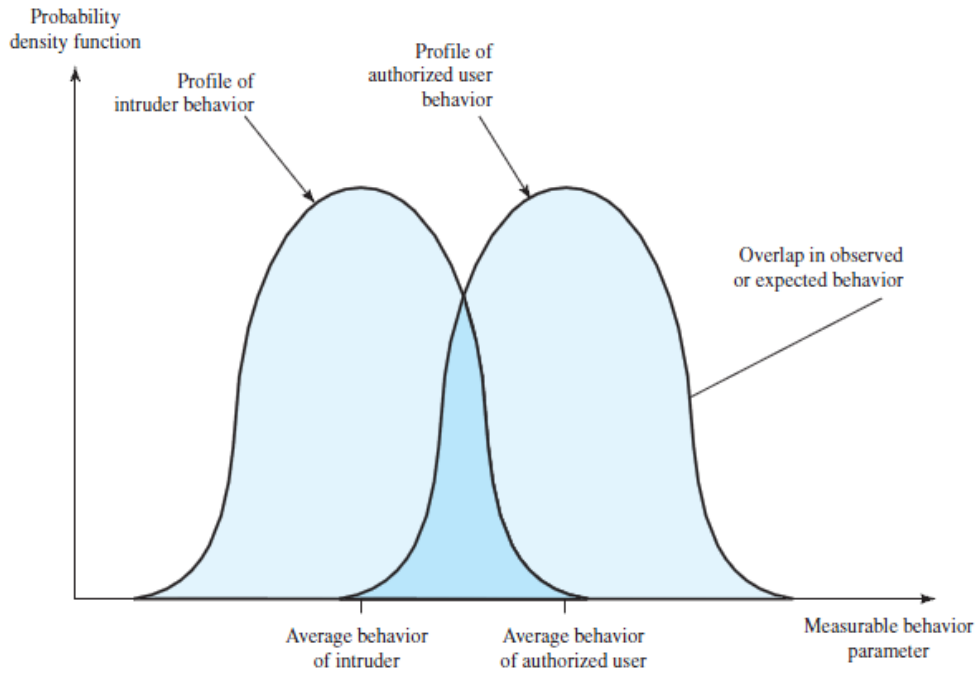
συμβάντα που έχουν σχέση με την ασφάλεια, ενώ ταυτόχρονα μπορεί να αποτελέσει πλατφόρμα για διαδικτυακές ενέργειες που δεν έχουν σχέση με την ασφάλεια. Όμως το firewall δεν μπορεί να προστατέψει το δίκτυο, όταν μια επίθεση καταφέρνει να το διαπεράσει, όπως και ούτε από εσωτερικές επιθέσεις. Επίσης δεν μπορεί να προστατέψει το δίκτυο από την μεταφορά «μολυσμένων» προγραμμάτων και αρχείων, αφού δεν είναι εύκολο να ελέγχει για κακόβουλο κώδικα.

4.2 Συστήματα Ανίχνευσης Εισβολής (Intrusion Detection)

Οι τρόποι προστασίας σε ένα σύστημα που αναφέρθηκαν πιο πάνω είναι σημαντικό να υλοποιούνται σε ένα σύστημα. Όμως εξίσου σημαντικό είναι να υπάρχει η δυνατότητα ανίχνευσης όταν γίνεται μια εισβολή στο σύστημα.

Η ανίχνευση εισβολών είναι απαραίτητο να υπάρχει σε ένα σύστημα. Είναι ο τρόπος με τον οποίο μπορεί να γίνει ανίχνευση ότι το σύστημα δέχεται επιθέσεις, ακόμα και όταν αυτές δεν είναι αντιληπτές παρά μόνο όταν προκαλέσουν ζημιά στο σύστημα. Όταν μια εισβολή στο σύστημα ανιχνευτεί γρήγορα και έγκαιρα, τότε ο εισβολέας μπορεί να αναγνωριστεί και να διωχθεί από το σύστημα πριν προκαλέσει μεγάλες ζημιές σε αυτό ή αλλοιώσει δεδομένα.[12]

Τα συστήματα ανίχνευσης εισβολής είναι σχεδιασμένα έτσι ώστε να ελέγχουν την κίνηση των δεδομένων και των ενεργειών που γίνονται σε ένα σύστημα και να είναι σε θέση να αναγνωρίζουν επιθέσεις εναντίον του. Στόχος τους είναι να μπορούν να χειριστούν τις απειλές εναντίον του συστήματος αποτελεσματικά. Στηρίζονται στην υπόθεση ότι οι ενέργειες κάποιου εισβολέα διαφέρουν από κάποιου αυθεντικού χρήστη του συστήματος και μπορούν να αναγνωριστούν. Φυσικά αυτό δεν είναι απόλυτο, γιατί μπορεί να μην υπάρχει κάποιος εμφανής διαχωρισμός και μπορεί να υπάρχει και κάποια επικάλυψη στις μεταξύ τους ενέργειες. Πιο κάτω παρουσιάζεται πώς αλληλεπιδρούν οι συμπεριφορές ενός εισβολέα του συστήματος και ενός εξουσιοδοτημένου χρήστη.



Σχήμα 4.1 Συμπεριφορά εισβολέα και εξουσιοδοτημένου χρήστη [18]

Συνήθως οι ενέργειες ενός χρήστη και ενός εισβολέα διαφέρουν. Παρόλα αυτά μπορεί και να υπάρχουν κάποια σημεία στα οποία συγκλίνουν. Έτσι στην προσπάθεια καθορισμού των ορίων της συμπεριφοράς ενός εισβολέα για να γίνεται πιο εύκολη η ανίχνευση, μπορεί να παρουσιαστούν false positives και false negatives. Τα false positives παρουσιάζονται όταν η ανίχνευση γίνεται σε μεγάλη εμβέλεια και σε σημεία επικάλυψης. Κάποια φυσιολογική ενέργεια αναγνωρίζεται ως επίθεση. Αντίθετα, τα false negatives παρουσιάζονται όταν το σύστημα ανίχνευσης έχει περιορισμένη εμβέλεια με αποτέλεσμα το σύστημα να αδυνατεί να αναγνωρίσει κάποια επίθεση. [12] Είναι δύσκολη η αναγνώριση της ύπαρξης false negatives, γιατί δεν υπάρχει γνώση του πότε παρουσιάζονται. Αυτό το φαινόμενο στηρίζεται στο ποσοστό συγκεκριμενοποίησης. Όσο πιο συγκεκριμένη παρουσιάζεται η συμπεριφορά ενός εισβολέα τότε παρουσιάζονται πολλά false negatives, αλλά όταν αναφέρεται σε πιο μεγάλη εμβέλεια τότε αυξάνεται ο αριθμός των false positives. Είναι δύσκολος ο καθορισμός ενός ορίου για τη διάκριση των κακόβουλων ενεργειών από τις φυσιολογικές, έτσι ο σχεδιασμός ενός συστήματος ανίχνευσης πρέπει να γίνεται προσεκτικά και μετά από εκτενή μελέτη.

Χρησιμοποιούνται δύο διαφορετικά συστήματα για την ανίχνευση γνωστών και νέων επιθέσεων εναντίον του συστήματος, ο ανιχνευτής εύρεσης προτύπου και ο ανιχνευτής ανωμαλιών.

Ο ανιχνευτής εύρεσης προτύπου χρησιμοποιεί τον αλγόριθμο εύρεσης για να ανιχνεύει γνωστές επιθέσεις. Στηρίζεται στο γεγονός ότι κάθε επίθεση έχει μοναδικό πρότυπο αναγνώρισης, ένα fingerprint. Το fingerprint δίνεται από το RFC χρησιμοποιώντας το Message Digest Method (MD5), μόλις παρουσιαστεί κάποιος νέος ιός στο διαδίκτυο. Το πρότυπο αναγνώρισης εισάγεται σε βάση του ανιχνευτή και για κάθε πακέτο δεδομένων που λαμβάνεται από το σύστημα, ελέγχεται το πρότυπο με τα πρότυπα που υπάρχουν στη βάση. Αν υπάρχει κάποια συσχέτιση τότε το πακέτο αναγνωρίζεται ως κακόβουλο. Για να δημιουργηθεί το fingerprint πρέπει πρώτα η επίθεση να λάβει μέρος και έτσι αυτός ο τύπος ανιχνευτή αναγνωρίζει γνωστές επιθέσεις. Σε αυτού του τύπου τον ανιχνευτή στηρίζονται τα περισσότερα συστήματα ασφάλειας. Σημαντικό είναι επίσης να αναφερθεί ότι τέτοιου είδους συστήματα πρέπει να ενημερώνονται συνεχώς, έτσι ώστε να περιέχουν στη βάση τους τα πρότυπα όλων των κακόβουλων εφαρμογών που έχουν αναγνωριστεί, μιας και συνεχώς αναγνωρίζονται νέοι.

Αντίθετα με τον ανιχνευτή εύρεσης προτύπου, ο ανιχνευτής ανωμαλιών ανιχνεύει και νέου τύπου επιθέσεις. Ανιχνεύει διάφορες ανωμαλίες που μπορεί παρουσιαστούν στο δίκτυο και τις παρουσιάζει σαν επιθέσεις. Ο ανιχνευτής ανωμαλιών διαθέτει και αυτός μια βάση μέσα στην οποία εισάγει διάφορα φυσιολογικά πρότυπα. Οτιδήποτε μπορεί να παρουσιαστεί στο σύστημα που να μην μπορεί να ταυτιστεί με κάποιο από τα πρότυπα που περιέχονται στη βάση χαρακτηρίζεται ως επίθεση. Για να μπορεί να λειτουργήσει αυτός ο τύπος ανιχνευτή χρειάζονται 2 στάδια, το στάδιο μάθησης και το στάδιο εφαρμογής.

Στο στάδιο μάθησης εισάγονται στη βάση διάφορα πρότυπα, που δεν περιέχουν κακόβουλο κώδικα, έτσι ώστε να οριστεί το τι είναι φυσιολογική είσοδος δεδομένων. Φυσιολογική είσοδος δεδομένων μπορεί να οριστεί οτιδήποτε και να χαρακτηρίζεται από πολλά στοιχεία, όπως system calls και HTTP traffic. Μετά την εισαγωγή των διάφορων προτύπων, ο ανιχνευτής θέτει κάποιο όριο, ένα threshold. Στο στάδιο αξιολόγησης, ο ανιχνευτής είναι σε θέση να αναγνωρίσει τη φυσιολογική είσοδο

δεδομένων. Οτιδήποτε δεν ανήκει είναι εκτός του ορίου που τέθηκε χαρακτηρίζεται ως κακόβουλο.

Η μέθοδος ανίχνευσης ανωμαλιών, ενώ μπορεί να ανιχνεύσει νέες επιθέσεις παρουσιάζει κάποια μειονεκτήματα, τα οποία πρέπει να λαμβάνονται υπόψη κατά το σχεδιασμό συστημάτων που υποστηρίζουν τη συγκεκριμένη μέθοδο. Μπορεί να παρουσιαστεί ένα υψηλό ποσοστό λάθους, τόσο false negatives όσο και false positives. Η παρουσία πολλών false positives απαιτεί τον συνεχή έλεγχο του υπεύθυνου του συστήματος, ώστε να μπορούν φυσιολογικές εφαρμογές να εκτελεστούν. Αντίθετα η παρουσία πολλών false negatives μπορεί να αντιμετωπιστεί με τη χρήση πολλών συστημάτων ανίχνευσης σε διάφορα επίπεδα του συστήματος. Έτσι όταν μια κακόβουλη εφαρμογή δεν ανιχνευτεί από ένα σύστημα ανίχνευσης, να υπάρχει η δυνατότητα αναγνώρισης του από κάποιο άλλο. Δεν υπάρχει εγγύηση ότι ένα σύστημα ασφάλειας μπορεί να αναγνωρίσει όλες τις νέες επιθέσεις.

Ένα δεύτερο μειονέκτημα της ανίχνευσης ανωμαλιών είναι ο καθορισμός του τι είναι φυσιολογικό. Πρέπει ο σχεδιασμός ενός τέτοιου συστήματος να γίνεται προσεκτικά ώστε να μπορεί να καθοριστεί ένα αντιπροσωπευτικό όριο. Ακόμα κάποιο σύστημα ασφάλειας μπορεί να είναι εκπαιδευμένο ώστε να θεωρεί ότι οι κακόβουλες εφαρμογές είναι φυσιολογικές. Κάτι τέτοιο μπορεί να αποφευχθεί με το σχεδιασμό ενός συστήματος ασφάλειας από έμπιστα άτομα.

Τα καλύτερα συστήματα ανίχνευσης εισβολών είναι αυτά που χρησιμοποιούν και τις 2 μεθόδους, και την ανίχνευση εύρεσης προτύπου και την ανίχνευση ανωμαλιών. Με αυτό τον τρόπο η ανίχνευση εύρεσης προτύπου μπορεί να εντοπίσει γνωστές επιθέσεις, ενώ η ανίχνευση ανωμαλίας μπορεί να εντοπίσει καινούριες επιθέσεις αφαιρώντας το μειονέκτημα της πρώτης μεθόδου.

4.3 Παρουσίαση μοντέλου ανίχνευσης εισβολής

Το audit record χρησιμοποιείται ως είσοδος σε ένα σύστημα ανίχνευσης εισβολής. Μπορούν να δημιουργηθούν 2 διαφορετικά σενάρια με τα audit records. Το ένα αφορά τα audit records όπως αυτά δημιουργούνται από το σύστημα. Όλα τα συστήματα περιέχουν ένα λογισμικό, το οποίο παράγει στατιστικά στοιχεία με βάση τις δραστηριότητες ενός χρήστη στο συγκεκριμένο σύστημα. Έτσι χρησιμοποιώντας αυτά

τα στατιστικά στοιχεία υπάρχει το πλεονέκτημα ότι δεν χρειάζεται κάποιο επιπλέον λογισμικό και τα audit records χρησιμοποιούνται ως είσοδος στο σύστημα ανίχνευσης εισβολής. Όμως χρησιμοποιώντας τα συγκεκριμένα audit records, υπάρχει το μειονέκτημα ότι μπορεί να μην περιέχουν την απαραίτητη πληροφορία ή μπορεί η πληροφορία να μην εμφανίζεται με βολικό και κατανοητό τρόπο.

Ο δεύτερος τρόπος χρησιμοποίησης των audit records είναι η ανίχνευση συγκεκριμένων audit records. Για το συγκεκριμένο τρόπο απαιτείται η χρησιμοποίηση ενός προγράμματος – συλλέκτης, ο οποίος να επιλέγει μόνο τα audit records που περιέχουν την απαραίτητη πληροφορία που χρειάζεται από τον σύστημα ανίχνευσης εισβολής. Αυτή η μέθοδος έχει το πλεονέκτημα ότι το συγκεκριμένο πρόγραμμα μπορεί να θεωρηθεί ανεξάρτητο συστήματος και να χρησιμοποιηθεί σε διάφορες πλατφόρμες. Παρόλα αυτά όμως υπάρχει και το μειονέκτημα ότι προστίθεται και ένα επιπλέον φόρτος εργασίας στο σύστημα, αφού πέρα από το λογισμικό που υπολογίζει στατιστικά στοιχεία γενικά με τις δραστηριότητες του χρήστη, υπάρχει και το λογισμικό που διαχωρίζει τα audit records με βάση τη χρησιμότητα τους.

Πιο κάτω περιγράφεται ένα intrusion – detection μοντέλο, το οποίο στηρίζεται στο ότι μπορεί να ανιχνεύει ασυνήθιστη παραβιάσεις ασφαλείας στηριζόμενο στα audit records για ασυνήθιστη χρησιμοποίηση του συστήματος, το οποίο υλοποιήθηκε από την Dorothy Denning.

Intrusion – Detection model [4]: Σκοπός του μοντέλου είναι να ανιχνεύει παραβάσεις ασφαλείας από εξωτερικούς, αλλά και από εσωτερικούς χρήστες του συστήματος. Η δημιουργία ενός τέτοιου μοντέλου οφείλεται σε 4 παράγοντες:

Τα περισσότερα συστήματα σήμερα έχουν πολλές ατέλειες όσο αφορά την ασφάλειά τους και τα καθιστά ευάλωτα σε παραβιάσεις, εισβολές και άλλης μορφής κακομεταχείρισης. Τα συστήματα με ατέλειες δεν μπορούν να αντικατασταθούν εύκολα με καινούρια και αυτό οφείλεται κυρίως σε οικονομικούς λόγους.

Είναι σχεδόν αδύνατον να σχεδιαστούν απόλυτα ασφαλή συστήματα. Ακόμα και τα πιο ασφαλή συστήματα είναι επιρρεπή σε κακομεταχείριση από τους εσωτερικούς χρήστες, οι οποίοι χρησιμοποιούν λάθος τα δικαιώματά τους.

Το μοντέλο είναι σχεδιασμένο βασισμένο στην ιδέα ότι η εξερεύνηση για εύρεση ελαττωμάτων και αδυναμιών στο σύστημα αποτελεί μη συνηθισμένη χρήση του συστήματος. Με αυτόν τον τρόπο μπορούν να παρατηρηθούν παραβιάσεις ασφαλείας όταν υπάρχει ασυνήθιστη χρήση του συστήματος. Ασυνήθιστες συμπεριφορές στο σύστημα παρατηρούνται για παράδειγμα όταν κάποιος προσπαθεί να μπει μέσα στο σύστημα και εισάγει ένα μεγάλο αριθμό λανθασμένων συνθηματικών που αφορά ένα συγκεκριμένο λογαριασμό ή όταν κάποιος από τους χρήστες του συστήματος ενώνεται με το σύστημα ασυνήθιστες για αυτόν ώρες και προσπαθεί να χρησιμοποιήσει δεδομένα, τα οποία συνήθως δεν χρησιμοποιεί και άλλα πολλά. Τέτοιου είδους συμπεριφορές όμως μπορεί να μην αποτελούν κάποια παραβίαση της ασφάλειας του συστήματος. Μπορεί να αφορούν ενέργειες οι οποίες σχετίζονται με αλλαγή καθηκόντων ενός χρήστη. Έτσι είναι πολύ σημαντικό να καθοριστούν ποιες ενέργειες και ποιες στατιστικές μετρήσεις μπορούν να έχουν αποτελεσματική ανίχνευση κάποιας εισβολής στο σύστημα και να μειώνουν τις λανθασμένες προειδοποιήσεις.

Το μοντέλο αποτελεί ένα πρότυπο για μιας γενικής χρήσης intrusion detection σύστημα. Είναι σχεδιασμένο έτσι ώστε να μπορεί να εφαρμοστεί σε οποιοδήποτε σύστημα και δεν εξαρτάται από κάποια συγκεκριμένης αρχιτεκτονική, ούτε από κάποια συγκεκριμένη εφαρμογή, αλλά και ούτε από κάποιο συγκεκριμένο τύπο εισβολής.

Το μοντέλο αποτελείται από 6 διαφορετικά στοιχεία: τα υποκείμενα (subjects), τα αντικείμενα (subjects), τα audit records, τα περιγράμματα (profiles), οι καταγραφές ανωμαλιών (anomaly records) και οι κανόνες ενεργειών (activity rules). Πιο κάτω αναλύονται τα διάφορα μέρη ξεχωριστά.

- **Υποκείμενα (subjects):** Είναι οι υποκινητές των ενεργειών στο σύστημα. Συνήθως είναι οι χρήστες του συστήματος, όμως μπορεί να είναι και διεργασία που ενεργεί εκ μέρους των χρηστών ή ακόμα και το ίδιο το σύστημα. Όλες οι ενέργειες σε ένα σύστημα ξεκινούν από κάποιο υποκείμενο. Τα υποκείμενα μπορούν να κατηγοριοποιηθούν σε διάφορες κλάσεις για να μπορεί να είναι πιο εύκολος ο έλεγχος πρόσβασης στα αντικείμενα μέσα στο σύστημα.

- **Αντικείμενα (objects):** Είναι οι παραλήπτες των ενεργειών και συνήθως περιλαμβάνουν αρχεία, προγράμματα, μηνύματα, εγγραφές, εκτυπωτές και άλλα. Αντικείμενα μπορούν να θεωρηθούν και υποκείμενα όταν σε κάποια ενέργεια γίνονται παραλήπτες. Τα αντικείμενα κατηγοριοποιούνται με βάση τον τύπο τους.

- **Audit Records:** Απεικονίζουν τις ενέργειες που γίνονται στο σύστημα από υποκείμενα σε αντικείμενα. Η περιγραφή τους γίνεται με 6 πεδία. Δύο πεδία αναφέρονται στο υποκείμενο και το αντικείμενο και ένα άλλο στην ενέργεια που λαμβάνει μέρος. Ένα άλλο πεδίο αναφέρεται σε ένα μοναδικό χρόνο και ημερομηνία, που υποδηλώνει πότε έγινε η ενέργεια, ενώ ένα άλλο πεδίο αναφέρει τις πόρους του συστήματος που χρησιμοποιήθηκαν και σε ποια ποσότητα. Ένα τελευταίο πεδίο αναφέρει αν μια ή περισσότερες συνθήκες εξαίρεσης παρουσιάστηκαν κατά τον τερματισμό της ενέργειας. Επίσης αν τα audit records μαζεύονται από διάφορα συστήματα πρέπει να υπάρχει και ένα επιπλέον πεδίο για την αναγνώριση του κάθε συστήματος.

  Για κάθε audit record δημιουργείται μια αντιστοιχία στον πίνακα «audit matrix» για το υποκείμενο με το αντικείμενο (γραμμές – υποκείμενα και στήλες - αντικείμενα). Ο συγκεκριμένος πίνακας έχει μια αναλογία με τον πίνακα πρόσβασης, στον οποίο καθορίζονται τα δικαιώματα του κάθε υποκειμένου για πρόσβαση σε ένα αντικείμενο. Το μοντέλο της ανίχνευσης εισβολής παρατηρεί την ενέργεια που λαμβάνει μέρος χωρίς να ελέγχει κατά πόσο η συγκεκριμένη ενέργεια είναι εξουσιοδοτημένη με την υπόθεση ότι για να εκτελείται η ενέργεια σημαίνει ότι οι έλεγχοι πρόσβασης του συστήματος το επιτρέπουν. Έτσι ο ρόλος της ανίχνευσης εισβολής είναι να καθορίσει αν η ενέργεια είναι συνηθισμένη ή όχι ώστε να δημιουργήσει υποψίες ότι μπορεί να αποτελεί εισβολή στο σύστημα.

  Οι περισσότερες ενέργειες αφορούν πολλά αντικείμενα, αλλά το μοντέλο τις αποσυνθέτει, έτσι ώστε κάθε ενέργεια να αφορά μόνο ένα αντικείμενο. Αυτό έχει ως στόχο να διατηρούνται ακέραιοι οι μηχανισμού ασφάλειας έτσι ώστε να μπορούν να αναγνωριστούν εύκολα τυχόν ανωμαλίες. Επίσης το μοντέλο είναι πιο απλό και η εφαρμογή του είναι πιο εύκολη από ότι αν ήταν σύνθετες οι

ενέργειες. Τέλος η αποσύνθεση των ενεργειών είναι χρήσιμη για τα audit records, αφού η δομή τους περιλαμβάνει ένα αντικείμενο.

Το σύστημα είναι υπεύθυνο για τη δημιουργία των audit records και την αποστολή τους στο μοντέλο ανίχνευσης εισβολής για ανάλυση. Ο χρόνος που δημιουργούνται τα audit records καθορίζει τι τύπου δεδομένα είναι διαθέσιμα, όπως για παράδειγμα μπορούν να μετρηθούν οι επιτυχείς και ανεπιτυχείς προσπάθειες που έγιναν για να εκτελεστεί μια ενέργεια, οι πόροι οι οποίοι χρησιμοποιήθηκαν από την εκτέλεση μια ενέργειας ή οι συνθήκες εξαίρεσης, οι οποίες προκάλεσαν τον ανώμαλο τερματισμό μιας ενέργειας. Έτσι στο τέλος εκτέλεσης μια ενέργειας υπάρχουν περισσότερες πληροφορίες, όμως κάτι τέτοιο δεν επιτρέπει την άμεση ανίχνευση εισβολών στο σύστημα. Για τον έγκαιρο εντοπισμό τους είναι καλύτερη η δημιουργία audit records τόσο στην αρχή της εκτέλεσης μιας ενέργειας, όσο και στον τερματισμό της. Αυτό είναι απαραίτητο σε ορισμένες ενέργειες, οι οποίες μπορούν να προκαλέσουν μεγάλη ζημιά όταν υπάρχει εισβολή στο σύστημα. Δεν είναι σαφές πόσο συχνά πρέπει να δημιουργούνται τα audit records. Η συνεχής όμως δημιουργία τους μπορεί να μειώσει την επίδοση του συστήματος, αλλά ταυτόχρονα να υπερφορτώνει και το σύστημα ανίχνευσης εισβολής.

Τέλος θα ήταν προτιμότερο όπως δημιουργηθεί μια ενιαία αναπαράσταση των audit records,αφού το καθένα έχει τη δική του δομή και χρειάζεται γνώση για την αναπαράστασης τους, ώστε να μπορεί να γίνει σωστή ανάλυση. Έτσι το σύστημα ανίχνευσης εισβολής να είναι ανεξάρτητο του συστήματος και να μπορεί να χρησιμοποιείται σε διάφορα συστήματα.

Οι περισσότερες ενέργειες αφορούν πολλά αντικείμενα, αλλά το μοντέλο τις αποσυνθέτει, έτσι ώστε κάθε ενέργεια να αφορά μόνο ένα αντικείμενο. Αυτό έχει ως στόχο να διατηρούνται ακέραιοι οι μηχανισμού ασφάλειας έτσι ώστε να μπορούν να αναγνωριστούν εύκολα τυχόν ανωμαλίες. Επίσης το μοντέλο είναι πιο απλό και η εφαρμογή του είναι πιο εύκολη από ότι αν ήταν σύνθετες οι ενέργειες. Τέλος η αποσύνθεση των ενεργειών είναι χρήσιμη για τα audit records, αφού η δομή τους περιλαμβάνει ένα αντικείμενο.

- **Περίγραμμα ενέργειας (activity profile):** Το περίγραμμα χαρακτηρίζει τη συμπεριφορά ενός υποκειμένου προς ένα αντικείμενο. Αποτελεί δηλαδή, μια περιγραφή της φυσιολογικής ενέργειας όσον αφορά το υποκείμενο και το αντικείμενο. Μια συμπεριφορά που γίνεται αντικείμενο παρατήρησης, καθορίζεται από διάφορες μετρικές που αφορούν μια χρονική περίοδο, όπως πόσα audit records εκτελέστηκαν, ο χρόνος μεταξύ 2 γεγονότων ή η ποσότητα πόρων που χρησιμοποιήθηκαν από μια δραστηριότητα. Οι παρατηρήσεις που λαμβάνονται από τα audit records χρησιμοποιούνται από στατιστικά μοντέλα για να καθοριστεί κατά πόσο μια νέα παρατήρηση είναι ασυνήθιστη. Τα περιγράμματα δημιουργούνται αυτόματα και αρχικοποιούνται με τα πρότυπα περιγραμμάτων που υπάρχουν.

- **Εγγραφή ανωμαλίας (anomaly record):** Είναι οι εγγραφές που δημιουργούνται όταν ανιχνευτεί κάποια ανωμαλία στο σύστημα. Με βάση τους κανόνες ενέργειας το σύστημα ανίχνευσης εισβολής, ενημερώνει το περιγράμματα ενέργειας και ελέγχει για ασυνήθιστες συμπεριφορές κάθε φορά που δημιουργείται audit record ή μια χρονική περίοδος τελειώνει. Αν ανιχνευτεί κάποια ανωμαλία στο σύστημα τότε δημιουργείται μια εγγραφή ανωμαλίας, η οποία περιλαμβάνει το γεγονός που τη δημιούργησε, το μοναδικό χρόνο που αναγράφεται στο audit records και το περίγραμμα ενέργειας στο οποίο παρουσιάστηκε.

- **Κανόνες Ενέργειας (Activity Rules):** Είναι κανόνες, οι οποίοι καθορίζουν την ενέργεια που θα πραγματοποιηθεί όταν ένα audit record ή εγγραφή ανωμαλίας δημιουργούνται ή όταν μια χρονική περίοδος τελειώνει. Ο κανόνας αποτελείται από 2 μέρη, τη συνθήκη που πρέπει να ικανοποιηθεί και την ενέργεια του κανόνα που θα εκτελεστεί. Υπάρχουν 4 τύποι κανόνων. Αρχικά είναι ο audit record κανόνας, ο οποίος ενεργοποιείται όταν υπάρχει σύνδεση μεταξύ ενός καινούριου audit record και ενός περιγράμματος ενέργειας. Ανανεώνει το περίγραμμα και ελέγχει για ασυνήθιστη συμπεριφορά. Ο περιοδικός κανόνας ανανέωσης ενέργειας ενεργοποιείται στο τέλος κάθε χρονικής περιόδου. Ανανεώνει το περίγραμμα ενέργειας και ελέγχει για ασυνήθιστη συμπεριφορά. Ο κανόνας εγγραφής ανωμαλίας ενεργοποιείται όταν δημιουργείται μια εγγραφή ανωμαλίας, όπου η ανωμαλία που παρουσιάζεται στο σύστημα παρουσιάζεται

στον τεχνικό της ασφάλειας του συστήματος για να ληφθεί υπόψη. Τέλος ο περιοδικός κανόνας για ανάλυση ανωμαλίας, ενεργοποιείται στο τέλος κάθε συγκεκριμένης χρονικής περιόδου και παράγει συνοπτικές αναφορές για τις ανωμαλίες που παρουσιάστηκαν στο σύστημα.

Το παραπάνω μοντέλο που περιγράφηκε για σύστημα ανίχνευσης εισβολής αποτελεί πρότυπο για την ανάπτυξη ενός συστήματος ανίχνευσης εισβολής πραγματικού χρόνου, ικανό να ανιχνεύσει διαφόρων ειδών εισβολές στο σύστημα. Αυτό μπορεί να γίνει χωρίς να πρέπει να υπάρχει γνώση για τις αδυναμίες που μπορεί να υπάρχουν στο σύστημα που πρέπει να προστατευθεί. Παρόλα αυτά υπάρχουν διάφορα ερωτήματα που αφορούν τη λειτουργία του συγκεκριμένου μοντέλου και κατά πόσο μπορεί να φέρει τα επιθυμητά αποτελέσματα. Μπορεί για παράδειγμα να μην μπορεί να ανιχνεύσει ένα άτομο που σταδιακά αλλάζει τις δραστηριότητες του στο σύστημα ή μορφές εισβολής που χρησιμοποιούν χαμηλού επιπέδου χαρακτηριστικά του συστήματος με αποτέλεσμα να μην είναι δυνατή η ανίχνευση τους. Έτσι υπάρχει η πιθανότητα να μην ανιχνευτεί μια τέτοιου είδους εισβολή, παρά μόνο όταν ένας εισβολέας παρουσιάζει ασυνήθιστη συμπεριφορά και έχει δυνατότητα πρόσβασης σε διάφορα μέρη του συστήματος που πριν δεν ήταν δυνατόν. Τέλος, το παραπάνω μοντέλο ακόμα και αν αποδειχθεί ότι μπορεί να ανιχνεύσει τις περισσότερες εισβολές, δεν σχεδιάστηκε για την αντικατάσταση άλλων ελέγχων ασφαλείας, γιατί κάτι τέτοιο μπορεί να αφήσει το σύστημα εκτεθειμένο σε εισβολές που θα παρουσιαστούν μια φορά και δεν θα αφήσουν κάποια στοιχεία μετά. Σκοπός του μοντέλου είναι να αποτελέσει ένα επιπλέον στρώμα ασφάλειας, το οποίο ενισχύει περαιτέρω την ασφάλεια του συστήματος.

4.4 Προηγούμενες μελέτες για ανίχνευση ανωμαλιών

Η μέθοδος ανίχνευσης ανωμαλιών προκαλεί το ενδιαφέρον διάφορων ερευνητών, οι οποίοι μελέτησαν διάφορους αλγόριθμους για την εύρεση του ιδανικού συστήματος. Σε αυτό το υποκεφάλαιο θα παρουσιαστούν διάφορες μελέτες που έγιναν.

4.4.1 Ασφάλεια Συστήματος και Βιολογική Ανοσολογία

Πολλές μελέτες έγιναν με βάση το συγκεκριμένο θέμα, λόγω των ομοιοτήτων που υπάρχουν στα ανοσοποιητικά συστήματα των βιολογικών οργανισμών και συστήματα

ασφάλειας. Παρά τις διαφορές που έχουν, οι μέθοδοι ανίχνευσης και αντιμετώπισης των βιολογικών ασθενειών, μπορούν να χρησιμοποιηθούν για την ανίχνευση κακόβουλων προγραμμάτων. Αυτή η αναλογία χρησιμοποιήθηκε για την ανάπτυξη ενός συστήματος ανίχνευσης με τη χρήση system calls, όπως αναφέρεται στα άρθρα "Computer Immunology" [5] και "Intrusion Detection using Sequences of System Calls"[8].

Ένα βιολογικό σώμα για να είναι υγιές έχει τη δυνατότητα να διαχωρίζει τον «εαυτό του» από τον «μη - εαυτό του». Χρησιμοποιώντας τον όρο «εαυτό» γίνεται αναφορά στο κύτταρα του οργανισμού που είναι φυσιολογικά και υγιή, ενώ ο όρος «μη – εαυτός» αναφέρεται σε οποιαδήποτε ξένα αντικείμενα που δεν ανήκουν στην κατηγορία «εαυτός». Σε ένα ανθρώπινο σώμα οτιδήποτε αναγνωρίζεται ως «μη – εαυτός» καταστρέφεται. Έτσι ένα υπολογιστικό σύστημα πρέπει να έχει ως παράδειγμα ένα βιολογικό οργανισμό ώστε να μπορεί να αναγνωρίζει οτιδήποτε γίνεται από κάποια κακόβουλη εφαρμογή. Ο καθορισμός του «εαυτού» γίνεται από τους δημιουργούς των συστημάτων ανίχνευσης.

Ένα βιολογικό σώμα, πέρα από το καθορισμό του «εαυτού», για να εμποδίσει κάποιο ξένο αντικείμενο να εισέλθει σε αυτό, χρησιμοποιεί διάφορα επίπεδα προστασίας. Το ίδιο πρέπει να παρουσιάζεται και στα συστήματα ασφάλειας, έτσι ώστε όταν αποτύχει κάποιο σύστημα ανίχνευσης να αναγνωρίζει μια κακόβουλη εφαρμογή, τότε το σύστημα που βρίσκεται στο πιο ψηλό επίπεδο μπορεί να την ανιχνεύσει.

Ακόμα ένα υπολογιστικό σύστημα πρέπει να έχει περισσότερα από ένα σημεία ελέγχου για την ανίχνευση κακόβουλου κώδικα. Όλα τα σημεία πρέπει να αλληλεπιδρούν μεταξύ τους ώστε να μπορούν να διαχειρίζονται σωστά όλους τους πόρους του συστήματος. Όταν υπάρξει μια εισβολή στο σύστημα και την αναγνωρίσει ένα από τα συστήματα ασφαλείας, τότε μπορούν να ειδοποιηθούν και τα υπόλοιπα ή οι διαχειριστές του συστήματος, ώστε να την απομονώσουν.

Η δημιουργία μοναδικών συστημάτων ασφάλειας μπορεί να παρέχει περισσότερη ασφάλεια. Ο κάθε οργανισμός έχει μοναδικό σύστημα προστασίας για να μην μπορούν όλοι να επηρεάζονται το ίδιο από κάποιο ιό. Το ίδιο ισχύει και για τα υπολογιστικά συστήματα. Αν όλα είχαν εγκατεστημένα τα ίδια προγράμματα ασφάλειας, τότε στην

περίπτωση που κάποιος κακόβουλος κώδικας μπορούσε να τα διαπεράσει, όλα τα υπολογιστικά συστήματα θα ήταν εκτεθειμένα.

Στην περίπτωση προηγούμενων επιθέσεων, το ανθρώπινο σύστημα αντιδρά άμεσα και γρήγορα, γιατί θυμάται τη συγκεκριμένη επίθεση. Στην περίπτωση νέων επιθέσεων χρειάζεται περισσότερη ώρα ανταπόκρισης αλλά μπορεί να ανταποκριθεί αποτελεσματικά. Με αυτό τον τρόπο πρέπει να είναι σχεδιασμένοι και οι ανιχνευτές εισβολών. Πρέπει να ανταποκρίνονται γρήγορα σε αναγνωρισμένες κακόβουλες εφαρμογές, αλλά και να μπορούν να αντιμετωπίσουν νέες επιθέσεις.

Το ανθρώπινο ανοσοποιητικό σύστημα χρησιμοποιεί την ατελή ανίχνευση και αποτελείται από 2 φάσεις, την φάση μάθησης και τη φάση διάδοσης. Αρχικά το ανοσοποιητικό σύστημα μαθαίνει για κάποια ασθένεια με την πρώτη εμφάνισή της στο σύστημα και μετά τη διαδίδει σε όλο το ανθρώπινο σύστημα. Η ατελής ανίχνευση ταυτίζεται με την ανίχνευση κατάχρησης των συστημάτων ασφάλειας. Μόλις ανιχνευτεί κάποια κακόβουλη εφαρμογή, τότε πρέπει να γίνεται άμεση ενημέρωση της βάσης από τους κατασκευαστές κάποιου συστήματος ανίχνευσης, αλλά ταυτόχρονα να ενημερώνουν και τους υπόλοιπους πελάτες του για να μπορούν να προστατέψουν τα συστήματά τους.

Με βάση όλα τα χαρακτηριστικά του ανοσοποιητικού συστήματος, υλοποιήθηκε μια μέθοδος ανίχνευσης εισβολής, όπου ο «εαυτός» καθορίζεται από τα system calls που δημιουργούνται από τις διεργασίες. Οι ερευνητές υποστήριξαν ότι ένα σύνολο από 6 ακολουθίες system calls μπορούσαν να έχουν μια ολοκληρωμένη εικόνα του «εαυτού». Επόμενο βήμα ήταν η δημιουργία βάσεων, όπου κάθε βάση περιείχε μια ακολουθία system calls για ένα συγκεκριμένο πρόγραμμα. Κάθε βάση αναφέρεται σε συγκεκριμένη αρχιτεκτονική, λογισμικό και ρυθμίσεις. Έτσι ένα δίκτυο μπορεί να περιέχει διάφορες βάσεις, ανάλογα με το αριθμό των προγραμμάτων που χρησιμοποιεί το δίκτυο. Ο στόχος είναι η σύγκριση των system calls που καλούνται με την εκτέλεση προγραμμάτων με τις βάσεις. Αν οι ακολουθίες των system calls δεν υπάρχουν στις βάσεις, θεωρούνται ως ασυνήθιστες.

Για την υλοποίηση του ανιχνευτή με την μέθοδο ανίχνευσης ανωμαλιών, χρειάζονται 2 φάσεις. Αρχικά είναι η φάση μάθησης του «εαυτού», όπου ο ανιχνευτής δέχεται διάφορα system calls από φυσιολογικές εφαρμογές. Συγκεκριμένα δημιουργήθηκαν 3

διαφορετικές βάσεις με την χρησιμοποίηση 3 διαφορετικών προγραμμάτων του λειτουργικού UNIX. Η φάση αξιολόγησης γίνεται με ακολουθίες μη φυσιολογικής συμπεριφοράς, τα οποία συγκρίνονται με τα δεδομένα των βάσεων.

Το μοντέλο που δημιουργήθηκε εξέτασε τις ακολουθίες άγνωστων προγραμμάτων και τα σύγκρινε με τη φυσιολογική συμπεριφορά της βάσης ενός από τα προγράμματα. Ο ανιχνευτής είχε 40- 80% αποτυχημένους συνδυασμούς. Επίσης χρησιμοποιήθηκαν και ακολουθίες από επιθέσεις, οι οποίες έγιναν αντιληπτές από τον ανιχνευτή.

Με βάση τα συγκεκριμένα αποτελέσματα, οι δημιουργοί του συγκεκριμένου εργαλείου κατέληξαν στο συμπέρασμα ότι μικρές ακολουθίες system calls μπορούν να καθορίσουν την φυσιολογική συμπεριφορά ενός προγράμματος. Ακόμα και αν το συγκεκριμένο δεν πληροί όλα τα χαρακτηριστικά που παρουσιάστηκαν πιο πάνω, οι δημιουργοί του πίστευαν ότι κατάφεραν να δημιουργήσουν ένα εργαλείο πραγματικού χρόνου, το οποίο μπορούσε να αναγνωρίσει άγνωστες επιθέσεις.

4.4.2 Καταγραφή Συμπεριφοράς Προγραμμάτων

Στο άρθρο “Detecting Anomalous and Unknown Intrusions Against Programs”[7], ερευνητές κατέγραψαν system calls τα οποία κλήθηκαν από συγκεκριμένα προγράμματα και χρησιμοποίησαν την ανίχνευση ανωμαλιών για την αναγνώριση νέων επιθέσεων. Επικεντρώθηκαν σε συγκεκριμένα system calls μιας και ο κακόβουλος κώδικας στοχεύει σε συγκεκριμένο λογισμικό. Άλλες μελέτες επικεντρώνονται στο επίπεδο συμπεριφοράς του χρήστη, όμως κάτι τέτοιο μπορεί να δημιουργήσει πολλά false positives, όταν χρησιμοποιηθούν λειτουργίες που δεν είναι συνηθισμένες. Έτσι η υλοποίηση ενός ανιχνευτή σε επίπεδο λογισμικού επιτρέπει τη χρησιμοποίηση του για διάφορους χρήστες.

Στο συγκεκριμένο άρθρο έγινε η χρήση νευρωνικών δικτύων και συγκεκριμένα το backpropagation δίκτυο, 3 διαφορετικά σύνολα δεδομένων και το πρόγραμμα Linux lpr. Το νευρωνικό δίκτυο backpropagation χρησιμοποιήθηκε, γιατί μπορεί να αναγνωρίζει αποκλίσεις από τη φυσιολογική συμπεριφορά. Αποτελείται από διάφορα επίπεδα και έτσι μπορεί να παράγει σωστά αποτελέσματα για τις εισόδους που χρησιμοποιούνται για τη φάση εκπαίδευσης. Όμως ένα τέτοιο δίκτυο είναι περίπλοκο και χρειάζεται αρκετή ώρα για την εκπαίδευσή του.

Τα σύνολα εκπαίδευσης αποτελούνται από ένα σύνολο φυσιολογικής συμπεριφοράς, ένα από κακόβουλες εφαρμογές και ένα με τυχαία δεδομένα. Το σύνολο της φυσιολογικής συμπεριφοράς δημιουργήθηκε με την δυναμική ανάλυση των διεργασιών κάτω από κανονικές συνθήκες λειτουργίας.

Διεξήχθηκαν 6 διαφορετικά πειράματα, τα οποία χρησιμοποίησαν ένα ή 2 σύνολα ή όλα για να καθοριστεί ποια από τα σύνολα χρειάζονται για τον σχεδιασμό ενός αποτελεσματικού ανιχνευτή ανωμαλιών.

Το πρόγραμμα Linux lpr χρησιμοποιήθηκε λόγω της αδυναμίας που έχει και ο buffer μπορεί να υπερφορτώνεται. Εκμεταλλευόμενοι αυτήν την αδυναμία, δημιουργήθηκε μια επίθεση για να ελεγχθεί κατά πόσο μπορεί να ανιχνευτεί η ασυνήθιστη συμπεριφορά του συγκεκριμένου προγράμματος.

Από τα 6 διαφορετικά πειράματα που έγιναν μόνο τα 2 ολοκληρώθηκαν επιτυχώς, το 3 και το 4 συγκεκριμένα. Το πείραμα 3 υπέδειξε ότι η ανίχνευση ανωμαλιών μπορεί να είναι χρήσιμη στην ανίχνευση άγνωστων επιθέσεων. Χρησιμοποιεί 2 σύνολα δεδομένων, το σύνολο με τα δεδομένα φυσιολογικής συμπεριφοράς και το σύνολο με τα τυχαία δεδομένα. Μόνο 0.9% ποσοστό λάθους παρουσιάστηκε στο $3^{ο}$ πείραμα. Στο $4^{ο}$ πείραμα όπου χρησιμοποιήθηκαν και τα 3 σύνολα δεδομένων παρουσιάστηκαν τα ίδια αποτελέσματα με το πείραμα 3.

Στο $1^{ο}$ πείραμα, όπου εξετάστηκαν μόνο τα δεδομένα φυσιολογικής συμπεριφοράς τα αποτελέσματα δεν ήταν ικανοποιητικά με 20% ποσοστό λάθους. Το ίδιο και στο $2^{ο}$ πείραμα όπου εξετάστηκαν σε συνδυασμό με το δεδομένα ασυνήθιστης συμπεριφοράς. Έτσι μπορεί να θεωρηθεί ότι το σύνολο τυχαίων δεδομένων είναι απαραίτητο για την αναγνώριση κακόβουλων εφαρμογών.

Από τη συγκεκριμένη μελέτη αποδείχθηκε ότι η μέθοδος ανίχνευσης ανωμαλιών υλοποιημένη στο επίπεδο των διεργασιών μπορεί να αναγνωρίσει κακόβουλο κώδικα. Επίσης ο συγκεκριμένος ανιχνευτής μπορεί να χρησιμοποιηθεί με οποιεσδήποτε ρυθμίσεις χρηστών.

4.4.3 Εξερχόμενη κίνηση διαδικτύου

Η ασφάλεια ενός δικτύου είναι ένα θέμα που απασχολεί πολύ. Για την ασφάλεια των δικτύων χρησιμοποιούνται firewalls και proxy εξυπηρετητές, που δεν επιτρέπουν την απευθείας πρόσβαση στο εσωτερικό του δικτύου.

Ένας εισβολέας μπορεί να έχει πρόσβαση σε ένα τέτοιο δίκτυο εφαρμόζοντας την εξής διαδικασία. Αρχικά πρέπει να προηγηθεί πρόσβαση ενός χρήστη σε κάποιο κακόβουλο αρχείο ή ιστοσελίδα, ώστε να μολυνθεί η μηχανή του χρήστη. Πρέπει μετά να δημιουργηθεί ένα μονοπάτι επικοινωνίας με το δίκτυο. Αυτό θα γινόταν με την εγκατάσταση κάποιου backdoor προγράμματος, επειδή όμως η οποιαδήποτε εισερχόμενη κίνηση παρεμποδίζεται, ο μόνος τρόπος επικοινωνίας είναι με την εξερχόμενη κίνηση του δικτύου, που μπορεί να γίνει με εξυπηρετητή ηλεκτρονικού ταχυδρομείου ή με proxy εξυπηρετητή.

Το Web Tap [1] είναι ένα σύστημα ανίχνευσης ανωμαλιών σε επίπεδο δικτύου, το οποίο ελέγχει τα αιτήματα χρηστών εξερχόμενης HTTP κίνησης για τυχόν επιθέσεις με τη διαρροή πληροφοριών. Ο λόγος που ελέγχεται η εξερχόμενη κίνηση και όχι η εισερχόμενη είναι γιατί η εισερχόμενη είναι πιο δύσκολο να ελεγχθεί, αφού τα πακέτα που εισέρχονται σε ένα δίκτυο είναι πάρα πολλά. Επιπλέον η εισερχόμενη κίνηση ελέγχεται όπως αναφέρθηκε και πιο πάνω με proxy εξυπηρετητών και firewall.

Αρχικά για την υλοποίηση ενός ανιχνευτή ανωμαλιών πρέπει να καθοριστεί το μοντέλο επιθέσεων. Το μοντέλο επιθέσεων καθορίζει τα πεδία που μπορεί να ανιχνεύσει το Web Tap. Περιλαμβάνει τα κανάλια HTTP, τα backdoor προγράμματα και τα spyware. Τα κανάλια HTTP χρησιμοποιούνται από τα backdoor προγράμματα, ώστε να αποστείλουν πληροφορίες μέσω εξουσιοδοτημένης κίνησης, το οποίο μπορεί να αναγνωριστεί από το Web Tap. Επίσης τα backdoor προγράμματα και τα spyware χρησιμοποιούν μηχανισμούς πλοήγησης ή POST επικεφαλίδες για να διαχωριστούν από τα άλλα προγράμματα. Έτσι το Web Tap είναι εκπαιδευμένο για να καταγράφει τις επικεφαλίδες και να αναφέρει τις ασυνήθιστες.

Ακολουθεί η διαδικασία καθορισμού των 6 διαφορετικών φίλτρων που χρησιμοποιούνται από το Web Tap. Το πρώτο είναι η διάταξη της επικεφαλίδας. Κάθε μηχανισμός αναζήτησης παρουσιάζει μοναδικό χαρακτηριστικό επικεφαλίδας. Η

διάταξη της επικεφαλίδας μπορεί να χρησιμοποιηθεί για να εξεταστεί κατά πόσο ένα αίτημα έγινε από κάποιο μη μηχανισμό αναζήτησης ή είναι τροποποιημένη από κάποιο εισβολέα. Επίσης μπορεί να χρησιμοποιηθεί για τον έλεγχο αιτημάτων που γίνονται από χρήστες.

Το δεύτερο φίλτρο είναι η καθυστέρηση χρόνου. Στη φάση εκμάθησης διατηρούνται log αρχεία για την καθυστέρηση χρόνου σε κάθε ιστοσελίδα για κάθε χρήστη. Έτσι αν παρουσιάζεται περισσότερη καθυστέρηση χρόνου, τότε θεωρείται ανωμαλία.

Το μέγεθος των αιτημάτων είναι ένα άλλο στοιχείο που ελέγχεται. Τα αιτήματα που γίνονται σε ιστοσελίδες είναι μικρά και δεν μεταφέρουν πολλή πληροφορία, ενώ αιτήματα που γίνονται από εισβολείς περιέχουν μεγάλο μέγεθος πληροφορίας και έτσι είναι μεγαλύτερα από ότι συνήθως. Έτσι υπολογίστηκε ένα όριο 3KB, έτσι ώστε όταν υπερβαίνεται να θεωρείται ως κακόβουλο. Το ίδιο γίνεται και για το εύρος ζώνης. Το εξερχόμενο εύρος ζώνης ελέγχεται γιατί οι εισβολείς χρειάζονται μεγάλο μέγεθος εύρους ζώνης. Έτσι αν το εύρος ζώνης υπερβαίνει το όριο που καθορίζεται, τότε αναγνωρίζεται ως κακόβουλο.

Η περιοδικότητα των αιτημάτων επίσης ελέγχεται. Τα μη κακόβουλα αιτήματα παρουσιάζονται κατά μικρά χρονικά διαστήματα. Οι εισβολείς συνήθως αποστέλλουν κακόβουλα αιτήματα με μεγαλύτερη διαφορά χρόνου. Έτσι παρατηρώντας την περιοδικότητα των αιτημάτων μπορούν να αναγνωριστούν τα κακόβουλα αιτήματα. Τέλος χρησιμοποιείται και η χρονική στιγμή που γίνονται τα αιτήματα. Κατά τη φάση της εκμάθησης το Web Tap καταγράφει την χρονική στιγμή που παρουσιάζεται η περισσότερη εξερχόμενη κίνηση. Αν υπάρχει αίτημα από κάποιο χρήστη σε ασυνήθιστη χρονική στιγμή τότε αναγνωρίζεται ως κακόβουλο. Ο καθορισμός του συγκεκριμένου ορίου είναι ιδιαίτερα δύσκολος γιατί οι χρονικές στιγμές τείνουν να διαφέρουν.

Το Web Tap ενσωματώνεται σε εξυπηρετητές proxy και τα αιτήματα περνούν πρώτα από το Web Tap πριν να φτάσουν στον προορισμό τους. Περνούν από τα φίλτρα που αναφέρθηκαν πιο πάνω και αν γίνει υπέρβαση κάποιου ορίου, τότε αναγνωρίζεται ως κακόβουλο.

Για τη φάση εκμάθησης και αξιολόγησης χρειάστηκαν 40 μέρες. Κατά τη φάση εκμάθησης το Web Tap κατέγραψε τη συμπεριφορά 30 διαφορετικών χρηστών. Κατά

το τέλος της εκμάθησης, τέθηκαν τα απαραίτητα όρια και αξιολογήθηκαν τα δεδομένα. Το Web Tap δημιούργησε διαφορετικές προειδοποιήσεις από διαφορετικά φίλτρα, σύμφωνα με την επίθεση που συνέβαινε. Κατά τη φάση αξιολόγησης χρησιμοποιήθηκαν οι ίδιοι 30 χρήστες και τα ποσοστά λάθους κυμάνθηκαν μεταξύ 0 – 32 % ανάλογα με τα όρια που καθορίζονται και τα φίλτρα. Οι εισβολείς κάποιες φορές μπορεί να βρουν τρόπους να εξαπατήσουν τα φίλτρα, αλλά τοποθετούνται εμπόδια για να ανατρέπονται οι ενέργειες των εισβολέων.

4.4.4 Ανίχνευση ανωμαλιών σε επιθέσεις διαδικτύου

Οι εξυπηρετητές και οι εφαρμογές διαδικτύου αποτελούν διάσημους στόχους επιθέσεων. Για την ανίχνευση των επιθέσεων διαδικτύου χρησιμοποιούνται συστήματα ανίχνευσης υλοποιημένα με τη μέθοδο ανίχνευσης κατάχρησης. Είναι δύσκολο όμως να διατηρούνται οι βάσεις των συστημάτων ενημερωμένες, λόγω των αδυναμιών που ανακαλύπτονται καθημερινά στα συστήματα και στις εφαρμογές. Έτσι η ανίχνευση κατάχρησης είναι καλύτερο να συνδυαστεί με τη μέθοδο ανίχνευσης ανωμαλιών. Δυστυχώς δεν υπάρχουν συστήματα ανίχνευσης που να είναι υλοποιημένα για την ανίχνευση επιθέσεων σε εφαρμογές διαδικτύου.

Το συγκεκριμένο άρθρο [10] παρουσιάζει ένα σύστημα ανίχνευσης ανωμαλιών εξετάζοντας τα HTTP αιτήματα, χρησιμοποιώντας διάφορα μοντέλα. Συγκεκριμένα εξετάζονται τα αιτήματα GET που αποστέλλονται από συγκεκριμένα προγράμματα με 6 διαφορετικά μοντέλα. Το κάθε μοντέλο είναι ένα σύνολο από διαδικασίες που είτε εξετάζει ένα συγκεκριμένο χαρακτηριστικό κάποιας ιδιότητας του αιτήματος, είτε ένα συγκεκριμένο χαρακτηριστικό του αιτήματος συνολικά. Το κάθε μοντέλο υπολογίζει την πιθανότητα παρουσίας του αιτήματος ως συνόλου ή κάποιας ιδιότητας του αιτήματος και το συγκρίνει με το την είσοδο. Με τον υπολογισμό της πιθανότητας, το όριο μπορεί να καθοριστεί. Αν ένα αίτημα ξεφεύγει από το όριο τότε θεωρείται κακόβουλο.

Τα 6 μοντέλα είναι το μέγεθος της ιδιότητας, η κατανομή χαρακτήρων των ιδιοτήτων, η δομή της διεπιφάνειας, ο ευρέτης συμβόλου, η παρουσία ή η απουσία ιδιότητας και η σειρά ιδιότητας. Το κάθε μοντέλο αποτελείται από 2 φάσεις, τη φάση μάθησης και τη φάση ανίχνευσης. Η φάση μάθησης αποτελείται από 2 διαφορετικά στάδια. Αρχικά δημιουργούντα τα προφίλ για κάθε πρόγραμμα του εξυπηρετητή και καθορίζονται τα

χαρακτηριστικά του. Μετά, σύμφωνα με τα προφίλ που δημιουργούνται καθορίζονται τα όρια.

Το μοντέλο για το μήκος της ιδιότητας εξετάζει το μέγεθος κάποιου χαρακτηριστικού του αιτήματος. Συνήθως το μήκος κάποιου χαρακτηριστικού του αιτήματος είναι σταθερό ή είναι μικρού μήκους και εισάγεται από το χρήστη. Έτσι στη φάση μάθησης, υπολογίζεται ο μέσος όρος του μεγέθους του χαρακτηριστικού και η διασπορά του από φυσιολογικά αιτήματα τα οποία χρησιμοποιούνται από τη μέθοδο ανισότητας Chebyshev για να καθοριστεί τι είναι φυσιολογικό και ποιο είναι το όριο διαχωρισμού. Στη φάση ανίχνευσης για κάθε αίτημα που εισέρχεται στο σύστημα, το μήκος μιας συγκεκριμένης ιδιότητας συγκρίνεται με το όριο. Αν είναι μεγαλύτερο τότε το αίτημα αναγνωρίζεται ως κακόβουλο. Η μέθοδος ανισότητας Chebyshev παράγει μικρό ποσοστό false positives, αλλά αδυνατεί να αναγνωρίσει όλα τα κακόβουλα πακέτα.

Στο μοντέλο κατανομής χαρακτήρων των ιδιοτήτων καταγράφεται η συχνότητα εμφάνισης των χαρακτήρων. Στηρίζεται στο γεγονός ότι οι ιδιότητες έχουν μια καθορισμένη δομή και περιέχουν χαρακτήρες που μπορούν να είναι κατανοητοί από τον άνθρωπο. Συνήθως οι χαρακτήρες προέρχονται από ένα σύνολο 256 πιθανών 8-bit τιμών και έτσι μπορούν να υπάρχουν ομοιότητες στη συχνότητα χαρακτήρων στις παραμέτρους ενός αιτήματος. Η κατανομή χαρακτήρων στηρίζεται στις ταξινομημένες σχετιζόμενες συχνότητες χαρακτήρων κάποιας ιδιότητας και ενδιαφέρει η συχνότητα με την οποία παρουσιάζονται οι χαρακτήρες και όχι ο χαρακτήρας. Σε ένα φυσιολογικό αίτημα αναμένεται ότι οι συχνότητες των χαρακτήρων μειώνονται σταδιακά. Μια κατανομή χαρακτήρων των ιδιοτήτων φυσιολογικού αιτήματος αναφέρεται ως ιδανική κατανομή χαρακτήρων. Στη φάση της μάθησης υπολογίζεται η ιδανική κατανομή χαρακτήρων και στη φάση ανίχνευσης χρησιμοποιείται μαζί με την μέθοδο χ^2 για να υπολογιστεί κατά πόσο η συγκεκριμένη ιδιότητα είναι κακόβουλη.

Στο μοντέλο δομής της διεπιφάνειας αναλύεται η δομή μιας παραμέτρου. Συνήθως όταν ένα αίτημα είναι κακόβουλο, οι παράμετροι του έχουν μεγάλες τιμές και οι χαρακτήρες δεν μπορούν να διαβαστούν. Οι εισβολείς κατάφεραν να συμπεριλάβουν τις συγκεκριμένες παραμέτρους στα αιτήματα με τέτοιο τρόπο ώστε να είναι δύσκολος ο εντοπισμός τους. Στη φάση μάθησης χρησιμοποιείται το μη-ντετερμινιστικό πεπερασμένο αυτόματο για να υπολογίζεται η πιθανότητα εμφάνισης για τη γραμματική

της δομής της παραμέτρου. Υπολόγισαν τις πιθανότητες γραμματικής εισάγοντας διάφορα αιτήματα στο αυτόματο. Το μη-ντετερμινιστικό πεπερασμένο αυτόματο αποτελείται από διάφορες καταστάσεις, οι οποίες καθορίζουν μια έξοδο και την πιθανότητα παρουσίασης της. Πιθανότητες ανατίθενται και σταις μεταβάσεις μεταξύ των καταστάσεων. Το αποτέλεσμα του μη-ντετερμινιστικού πεπερασμένου αυτόματου είναι η πιθανότητα παρουσίασης μια συγκεκριμένης γραμματικής. Αυτό το αποτέλεσμα μπορεί να θεωρηθεί και ως μοντέλο Marcov. Στη φάση ανίχνευσης το μοντέλο Marcov χρησιμοποιείται για καθορίσει τη πιθανότητα της ιδιότητας του αιτήματος. Αν η συγκεκριμένη τιμή δεν μπορεί να προέλθει από τη δοσμένη γραμματική, τότε θεωρείται κακόβουλη.

Το μοντέλο ευρέτης συμβόλου εξετάζει κατά πόσο συγκεκριμένες τιμές προέρχονται από κάποια σύνολα συμβόλων ή από στοιχεία αρίθμησης. Οι σπάνιες τιμές υποδηλώνουν επίθεση. Συνήθως οι μηχανές αναζήτησης διαδικτύου χρησιμοποιούν κάποιες πιθανές τιμές για συγκεκριμένες ιδιότητες αιτημάτων. Κατά τη φάση μάθησης, διάφορα αιτήματα περνούν μέσα από ένα σύνολο εξισώσεων, για να υπολογιστεί αν υπάρχει κάποια αρίθμηση. Αν ναι, τότε η γραμματοσειρά αποθηκεύεται και χρησιμοποιείται στη φάση ανίχνευσης. Στη φάση ανίχνευσης, αν η τιμή είναι σπάνια ή η αρίθμηση δεν είναι αποθηκευμένη, τότε το αίτημα θεωρείται κακόβουλο.

Το μοντέλο ύπαρξης ή απουσίας ιδιότητας στηρίζεται στις αδυναμίες των εισβολέων. Στο άρθρο υποστηρίζεται ότι συνήθως οι εισβολείς εκμεταλλεύονται τις αδυναμίες που παρουσιάζουν τα αιτήματα των χρηστών, όμως δεν δίνουν μεγάλη σημασία στη σειρά ή στην σωστή ολοκλήρωση των παραμέτρων. Στη φάση μάθησης δημιουργείται ένα σύνολο από επιτρεπτά αιτήματα και στην φάση ανίχνευσης γίνεται σε κάθε αίτημα που παρουσιάζεται ως είσοδος έλεγχος για ταύτιση με σύνολο που έχει καθοριστεί. Αν δεν υπάρχει ταύτιση, τότε το αίτημα είναι κακόβουλο.

Το τελευταίο μοντέλο, το μοντέλο σειράς ιδιότητας σχετίζεται με το πιο πάνω. Οι δημιουργοί επιθέσεων δεν δίνουν πολλή σημασία στην τοποθέτηση των παραμέτρων, ενώ τα προγράμματα των εξυπηρετητών τείνουν να παρουσιάζουν συγκεκριμένες τοποθετήσεις. Έτσι στο στάδιο μάθησης χρησιμοποιείται ο αλγόριθμος Tarjan και κατευθυνόμενοι γράφοι, ώστε να καθοριστούν οι περιορισμοί στη σειρά της κάθε

ιδιότητας. Στη φάση ανίχνευσης γίνεται έλεγχος αν το αίτημα εισόδου ικανοποιεί τους περιορισμούς που τέθηκαν.

Για την αξιολόγηση των μοντέλων χρησιμοποιήθηκαν 3 διαφορετικά σύνολα δεδομένων, ένα από το Google και 2 από πανεπιστήμια. Το σύνολο δεδομένων της Google παρουσίασε το μεγαλύτερο ποσοστό false positives. Επίσης χρησιμοποιήθηκε και ένα σύνολο από επιθέσεις, όπου παρατηρήθηκε ότι το σύστημα μπορούσε να ανιχνεύσει ένα σημαντικό ποσοστό επιθέσεων, που μπορούσαν να περάσουν απαρατήρητες από το σύστημα ανίχνευσης κατάχρησης. Το συγκεκριμένο σύστημα έχει ένα περιορισμό. Η φάση μάθησης στηρίζεται σε log εξυπηρετητών διαδικτύου, τα οποία δεν πρέπει να είναι «μολυσμένα», γιατί τότε δεν μπορεί να γίνει σωστά η εκμάθηση του συστήματος.

4.4.5 Σύστημα ανίχνευσης εισβολών δικτύου

Τα συστήματα ανίχνευσης εισβολής δικτύου καταγράφουν την κίνηση που υπάρχει σε ένα δίκτυο. Τα συστήματα ανίχνευσης εισβολής αδυνατούν να ελέγχουν μεγάλης ταχύτητας συνδέσεις διαδικτύου και παρέχουν περιορισμένο χρόνο στον διαχειριστή του δικτύου για τον έλεγχο του δικτύου. Για να αντιμετωπιστούν τα πιο πάνω προβλήματα δημιουργήθηκε το NATE (Network Analysis of Anomalous Traffic) [21], όπως αναφέρεται στο συγκεκριμένο άρθρο, που είναι μια χαμηλού κόστους προσέγγιση για την ανίχνευση εισβολής σε δίκτυα. Στόχος ήταν η δημιουργία ενός ελαφρού για το σύστημα εργαλείου, το οποίο χρησιμοποιώντας τη μέθοδο ανίχνευσης ανωμαλίας θα μπορούσε να αναγνωρίζει τις επιθέσεις που μπορούν να ανιχνευτούν από τις πληροφορίες της επικεφαλίδας.

Το NATE ενσωματώνει ένα μοναδικό σύνολο χαρακτηριστικών, που δεν είχε παρουσιαστεί πιο πριν σε λύσεις ανίχνευσης εισβολής. Στοχεύει στην μείωση του μεγέθους δεδομένων που χρειάζονται για την ανίχνευση κάποιας επίθεσης. Συγκεκριμένα για τον διαχωρισμό της φυσιολογικής από την κακόβουλη συμπεριφορά χρησιμοποιούνται τα TCP χαρακτηριστικά, όπως τα στοιχεία TCP P(Push) και Ack(Acknowledge), ο μέσος όρος και ο συνολικός αριθμός των bytes που μεταφέρθηκαν και το ποσοστό των στοιχείων ελέγχου.

Οι μέθοδοι που χρησιμοποιήθηκαν για τη δημιουργία του μοντέλου ήταν διάφορες. Χρησιμοποιήθηκε η ομαδοποίηση των δεδομένων, έτσι ώστε να ομαδοποιηθεί η φυσιολογική κίνηση και να καθοριστεί η φυσιολογική συμπεριφορά στο δίκτυο. Επίσης χρησιμοποιήθηκε αντιπροσωπευτική στρατηγική για τη συλλογή των δεδομένων που να καλύπτει σε μεγάλη εμβέλεια την συμπεριφορά του δικτύου. Συλλέχθηκαν αντιπροσωπευτικά δεδομένα για κάθε εφαρμογή που δημιουργεί TCP κίνηση, που στηρίζονταν τόσο στη συχνότητα εμφάνισης της εφαρμογής, όσο και στην κατανομή των ιδιοτήτων τους. Για τον διαχωρισμό των δεδομένων και τον καθορισμό ενός ορίου εξετάστηκαν διάφορες μέθοδοι, όπως η ευκλείδεια απόσταση και η απόσταση Mahalanobis και η μέθοδος της ανισότητας Chebyshev.

Για την αξιολόγηση του εργαλείου χρησιμοποιήθηκαν προσομοιωμένα σύνολα δεδομένων από τα MIT Lincoln Labs, τα οποία είχαν ικανοποιητικά αποτελέσματα, όμως παρουσίαζαν και κάποια προβλήματα. Έτσι χρησιμοποιήθηκαν αληθινά σύνολα δεδομένων από ένα λειτουργικό δίκτυο, όπου λάμβαναν μέρος κυρίως λειτουργίες διαδικτύου και ηλεκτρονικού ταχυδρομείου. Στο δίκτυο υπήρχαν και firewalls προσομοιωμένα να επιτρέπουν μόνο συγκεκριμένη κίνηση. Η συλλογή δεδομένων διήρκησε 10 μέρες. Τα πραγματικά δεδομένα περιέχουν πιθανόν και κακόβουλα δεδομένα, έτσι λήφθηκαν υπόψη 2 παράμετροι. Μη φυσιολογικά δεδομένα μπορεί να έχουν ελλειπή χαρακτηριστικά TCP ή να αποτελούνται από μεγάλα πακέτα δεδομένων. Με προσοχή ελέχθησαν τα δεδομένα και χρησιμοποιήθηκαν οι μέθοδοι διαχωρισμού.

Από τα αποτελέσματα φάνηκε ότι το εργαλείο είναι απαραίτητο να ελέγχεται τόσο με προσομοιωμένα δεδομένα όσο και με πραγματικά, ώστε να υπάρχει μια πιο αντιπροσωπευτική ανάλυση. Το NATE μπορεί να χρησιμοποιηθεί για την ανίχνευση επιθέσεων που μπορεί να γίνονται σε ένα δίκτυο από «μολυσμένες μηχανές». Επίσης μπορεί να ανιχνεύσει και επιθέσεις, οι οποίες κατάφεραν να διαπεράσουν από το firewall.

Κεφάλαιο 5

Σχεδιασμός και Υλοποίηση Εργαλείου Ασφάλειας

Εισαγωγή
5.1 Παρουσίαση προηγούμενης ερευνητικής εργασίας
5.2 Προτεινόμενη Λύση
5.3 Υλοποίηση Εργαλείου Ανίχνευσης
5.4 Παρατηρήσεις

Εισαγωγή

Στο προηγούμενο κεφάλαιο έγινε αναφορά στους 2 κυριότερους τρόπους ανίχνευσης που χρησιμοποιούνται στα συστήματα ανίχνευσης εισβολής. Αρχικά έχουμε την ανίχνευση κατάχρησης, η οποία χρησιμοποιείται από τα περισσότερα συστήματα ασφαλείας και η οποία μπορεί να ανιχνεύσει ήδη αναγνωρισμένες επιθέσεις. Αντίθετα η ανίχνευση ανωμαλιών είναι σχεδιασμένη ώστε να ανιχνεύει και νέες επιθέσεις.

Η ανίχνευση ανωμαλιών παρουσιάζει 2 στάδια, το στάδιο της εκπαίδευσης και το στάδιο της αξιολόγησης. Το στάδιο της εκπαίδευσης χρησιμοποιείται για να βρεθεί ένα όριο, έτσι ώστε να διαχωριστούν οι κακόβουλες εφαρμογές από τις συνηθισμένες. Το όριο υπολογίζεται με τέτοιο τρόπο έτσι ώστε όταν παρουσιαστεί τιμή μεγαλύτερη του ορίου, τότε αυτό να υποδηλώνει την ύπαρξη επίθεσης. Επίσης πρέπει να είναι καλά καθορισμένο ώστε να μην υπάρχει μεγάλο ποσοστό λάθους.

Ο σχεδιασμός και η υλοποίηση του μοντέλου ακολούθησε την μεθοδολογία που χρησιμοποιήθηκε στην Ερευνητική Εργασία της Χριστιάνας Ιωάννου “The Hunt for Viral Processes” [9], της οποία τα αποτελέσματα ήταν ικανοποιητικά.

5.1 Παρουσίαση προηγούμενης ερευνητικής εργασίας

Η Ερευνητική Εργασία της Χριστιάνας Ιωάννου "The Hunt for Viral Processes" [9] ασχολήθηκε με την ανάπτυξη ενός εργαλείου για την ανίχνευση κακόβουλων εφαρμογών με μικρό ποσοστό λάθος στηριζόμενο στη μέθοδο ανίχνευσης ανωμαλιών χρησιμοποιώντας τα system calls. Ορισμός των system calls δίνεται πιο κάτω.

Η πρώτη διαδικασία ήταν η συλλογή των δεδομένων. Τα δεδομένα έπρεπε να περιλαμβάνουν system calls, τα οποία έχουν καλεστεί από φυσιολογικές εφαρμογές και από κακόβουλες. Συγκεκριμένα χρησιμοποιήθηκαν 45 φυσιολογικές εφαρμογές και 67 κακόβουλες. Κατά την εκτέλεση των εφαρμογών εξετάστηκε η συχνότητα με την οποία γίνονταν κλήσεις σε ένα σύνολο 58 διαφορετικών system calls. Οι εφαρμογές εκτελέστηκαν σε ελεγχόμενο περιβάλλον.

Μετά τη συλλογή των δεδομένων εξετάστηκε κατά πόσο μπορούσαν να διαχωριστούν σε φυσιολογικές και κακόβουλες εφαρμογές και να ομαδοποιηθούν με τέτοιο τρόπο ώστε να καθοριστεί ένα όριο. Για το συγκεκριμένο σκοπό χρησιμοποιήθηκαν και εξετάστηκαν διαφορετικές μέθοδοι. Αρχικά εξετάστηκε ο διαχωρισμός τον δεδομένων με τη μέθοδο του Principal Component Analysis, για να μειωθούν οι διαστάσεις των δεδομένων και να βρεθούν κάποια πρότυπα σε αυτά. Η συγκεκριμένη μέθοδος μπορούσε να ομαδοποιήσει τις φυσιολογικές εφαρμογές, όμως δεν μπορούσε να διαχωρίσει και τις κακόβουλες. Έτσι χρησιμοποιήθηκαν 2 μέθοδοι ομαδοποίησης, η ευκλείδεια απόσταση και ο γωνιακός διαχωρισμός. Τα αποτελέσματα δεν ήταν καθόλου ικανοποιητικά, έτσι εξετάστηκε μια άλλη μέθοδος το Logistic Regression.

Το Logistic Regression είναι ένα στατιστικό εργαλείο, το οποίο χρησιμοποιείται για την υλοποίηση ενός μοντέλου. Το μοντέλο μπορεί να αξιολογήσει μια εφαρμογή χρησιμοποιώντας ένα σύνολο ανεξάρτητων μεταβλητών και να καθορίσει κατά πόσο η συγκεκριμένη εφαρμογή είναι κακόβουλη ή όχι. Αποτελείται από 2 στάδια, το στάδιο εκπαίδευσης και το στάδιο αξιολόγησης. Χρησιμοποιήθηκαν 2 διαφορετικά σύνολα, τα οποία περιέχουν φυσιολογικές και κακόβουλες εφαρμογές, στα οποία εφαρμόστηκε αρχικά η μέθοδος του Principal Component Analysis. Το πρώτο σύνολο εφαρμογών χρησιμοποιήθηκε στη φάση εκπαίδευσης ώστε να δημιουργηθεί το μοντέλο ανίχνευσης και το δεύτερο χρησιμοποιήθηκε για την αξιολόγησή του. Για κάθε εφαρμογή του

δεύτερου συνόλου ελέγχθηκε η πιθανότητα να είναι κακόβουλη. Αν η πιθανότητα ήταν μεγαλύτερη από 0.5 τότε ήταν, αλλιώς δεν ήταν.

Η συγκεκριμένη εργασία έδειξε ότι η μέθοδος του Logistic Regression είναι αρκετά υποσχόμενη και μπορεί να χρησιμοποιηθεί για το διαχωρισμό φυσιολογικών και κακόβουλων εφαρμογών. Τα αποτελέσματα ήταν αρκετά ικανοποιητικά αφού το ποσοστό λάθους δεν ήταν πολύ μεγάλο. Στα συμπεράσματα αναφέρθηκε ότι ένα διαφορετικό σύνολο system calls και ένας μεγαλύτερος αριθμός εφαρμογών μπορούν να οδηγήσουν σε καλύτερα αποτελέσματα.

5.2 Προτεινόμενη Λύση

Όπως αναφέρθηκε και πιο πάνω ο σχεδιασμός και η υλοποίηση του μοντέλου ακολούθησε την μεθοδολογία που χρησιμοποιήθηκε στην Ερευνητική Εργασία της Χριστιάνας Ιωάννου [9].

Στόχος είναι ο σχεδιασμός ενός ανιχνευτή επιθέσεων στηριζόμενος στη μέθοδο ανίχνευσης ανωμαλιών χρησιμοποιώντας τα system calls, οποίος να παρουσιάζει μικρό ποσοστό εμφάνισης λάθους. Η υλοποίηση του μοντέλου γίνεται με τη χρήση του Logistic Regression σε συνδυασμό με το Principal Component Analysis.

Τα system calls είναι ειδικοί κώδικες της μηχανής που χρησιμοποιούνται από μια εφαρμογή για να ζητήσει συγκεκριμένες υπηρεσίες ή πόρους από το Λειτουργικό Σύστημα. Οι διεργασίες δεν έχουν άμεση επαφή με το Λειτουργικό Σύστημα και τον πυρήνα του συστήματος, έτσι τα system calls μπορούν να έχουν πρόσβαση στον πυρήνα και επιτρέπουν σε μια διεργασία να εκτελέσει κάποιες ενέργειες. Ουσιαστικά τα system calls αποτελούν τον τρόπο με τον οποίο μια διεργασία και ένα λειτουργικό σύστημα επικοινωνούν, έτσι ώστε να εκτελεστεί μια διεργασία.

Για την υλοποίηση του μοντέλου χρησιμοποιήθηκε ένα σύνολο από 87 διαφορετικών system calls και εκτελέστηκαν 56 φυσιολογικά προγράμματα ευρείας χρήσης και 100 κακόβουλων προγραμμάτων.

Πιο κάτω επεξηγείται αναλυτικά η διαδικασία.

5.2.1 Συλλογή δεδομένων

Όπως αναφέρθηκε και πιο πάνω για τη δημιουργία του εργαλείου χρησιμοποιείται ένα σύνολο system calls, ένα σύνολο φυσιολογικών εφαρμογών και ένα σύνολο κακόβουλων εφαρμογών. Συγκεκριμένα με την εκτέλεση των εφαρμογών, οι διεργασίες που δημιουργούνται χρησιμοποιούν τα system calls για να έχουν πρόσβαση σε λειτουργίες και πόρους του Λειτουργικού Συστήματος.

Η διαδικασία συλλογής των δεδομένων καταγράφει την εμφάνιση των system calls κατά την εκτέλεση μιας διεργασίας σύμφωνα με το σύνολο των system calls που έχει καθοριστεί. Η επιλογή των system calls, που αποτελούν το σύνολο που χρησιμοποιήθηκε, στηρίχτηκε στα system calls που είχαν επιλεγεί για την Ερευνητική Εργασία "The Hunt for Viral Processes", αλλά και με βάση κάποια στατιστικά στοιχεία που υποδείκνυαν ποια system calls καλούνται αρκετά συχνά κατά την εκτέλεση των προγραμμάτων. Τα system calls χρησιμοποιήθηκαν παρουσιάζονται στον πιο κάτω πίνακα.

System Calls με Αλφαβητική Σειρά			
accept	DeleteFileA	getsocketname	RegOpenKeyW
CloseHandle	DeleteFileW	GetSystemTime	RegQueryValueA
closesocket	ExitProcess	GetSystemTimeAsFileTime	RegQueryValueW
coCreateGuid	ExitThread	GetTempFileNameA	RegSetValueA
coCreateInstance	FindFirstFileA	GetTempFileNameW	RegSetValueExA
coGetClassObject	FindFirstFileW	GetTickCount	RegSetValueExW
coGetTreatAsClass	FindNextFileA	GlobalAddAtomA	RegSetValueW
coInitialize	FindNextFileW	GlobalAddAtomW	send
coMarshalInterface	FreeLibrary	GlobalAlloc	SetCurrentDirectoryA
CopyFileA	GetCommandLineA	InitializeCriticalSection	SetCurrentDirectoryW
CopyFileW	GetCommandLineW	LoadLibraryA	SetFilePointer
coRegisterClassObject	GetCurrentProcess	LoadLibraryW	Sleep
coUnmarshalInterface	GetCurrentThreadId	lstrlenA	socket
CreateEventA	GetDateFormatA	lstrlenW	TerminateProcess
CreateEventW	GetDateFormatW	QueryPerformanceCounter	TerminateThread
CreateFileA	gethostbyname	recv	UnhandledExceptionFilter
CreateFileW	GetModuleFileNameA	RegCloseKey	VirtualAlloc
CreateMutexA	GetModuleFileNameW	RegCreateKeyA	WriteFile
CreateMutexW	GetModuleHandleA	RegCreateKeyW	WSACleanup
CreateProcessA	GetModuleHandleW	RegDeleteKeyA	WSARecv
CreateProcessW	getpeername	RegDeleteKeyW	WSAStartup
CreateThread	GetProcAddress	RegOpenKeyA	

Πίνακας 5.1 System Calls που χρησιμοποιήθηκαν

Η διαδικασία συλλογής δεν πραγματοποιείται με το ίδιο πρόγραμμα που χρησιμοποιήθηκε στην ερευνητική εργασίας της Χρ. Ιωάννου. Αυτό οφείλεται στο γεγονός ότι το συγκεκριμένο πρόγραμμα έχει αποσυρθεί με αποτέλεσμα να μην είναι δυνατή η χρήση του. Έτσι ήταν απαραίτητη η εύρεση ενός νέου προγράμματος, το οποίο θα ήταν σε θέση να καταγράφει τα system calls που καλούνται κατά την εκτέλεση μιας διεργασίας.

Έτσι χρησιμοποιήθηκε το WinAPIOverride32 [24]. Το συγκεκριμένο πρόγραμμα επιτρέπει τον καθορισμό των system calls, που θέλουμε να ελέγξουμε κατά πόσο κλήθηκαν από το πρόγραμμα που εκτελείται και καταγράφει τα system calls που καλούνται δημιουργώντας log αρχεία.

Το κάθε log αρχείο που δημιουργείται περιέχει όλες τις φορές που κλήθηκε ένα system calls με διάφορες πληροφορίες. Κάθε φορά που καλείται ένα system call αναφέρεται το όνομα του, η κλήση της συνάρτησης του με τις παραμέτρους, την ημερομηνία και ώρα κλήσης, τη βιβλιοθήκη στην οποία ανήκει το συγκεκριμένο system call και άλλες πληροφορίες.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	1	I	Process 0x67C (chrome.exe) successfully hooked										22:29:41:957:884,6							
2	2	In	InitializeC	0x0000000(	0x71AA16	WS2HELP.	0x0000067	0x000003E	0x0000000	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:51:4	18	kernel32.(	InitializeC	C:\WINDOWS\system32\WS2HELP.dll			
3	3	In	InitializeC	0x0000000(	0x71AB96(	ws2_32.dl	0x0000067	0x000003E	0x0000000	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:51:4	17	kernel32.(	InitializeC	C:\WINDOWS\system32\ws2_32.dll			
4	4	In	InitializeC	0x0000000(	0x71AB96(	ws2_32.dl	0x0000067	0x000003E	0x0000000	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:51:4	17	kernel32.(	InitializeC	C:\WINDOWS\system32\ws2_32.dll			
5	5	In	InitializeC	0x0000000(	0x71AB97(	ws2_32.dl	0x0000067	0x000003E	0x0000000	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:51:4	17	kernel32.(	InitializeC	C:\WINDOWS\system32\ws2_32.dll			
6	6	Out	GetSysten	0x01CAE1(	0x004309E	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0x00(	EAX=0x01(	-1.#IND	22:29:57:4	47	kernel32.(	GetSysten	C:\Documents and Settings\VMuser\Local Se			
7	7	In	GetCurrer	0x0000023	0x004309C	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:57:4	47	kernel32.(	GetCurrer	C:\Documents and Settings\VMuser\Local Se			
8	8	In	GetTickCo	0x00483C3	0x004309E	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:57:4	46	kernel32.(	GetTickCo	C:\Documents and Settings\VMuser\Local Se			
9	9	Out	QueryPer	0x0000000(	0x004309E	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:57:4	15	kernel32.(	QueryPer	C:\Documents and Settings\VMuser\Local Se			
10	10	In	GetModul	0x7C8000(	0x0042F85	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0x00(	EAX=0x7cl	-1.#IND	22:29:57:4	343	kernel32.(	GetModul	C:\Documents and Settings\VMuser\Local Se			
11	11	In	GetProcA(	0x0000000(	0x0042F8E	chrome.e	0x0000067	0x0000023	0x0000007	EAX=0x7cl	EAX=0x00(	-1.#IND	22:29:57:4	53	kernel32.(	GetProcA(	C:\Documents and Settings\VMuser\Local Se			
12	12	In	GetProcA(	0x0000000(	0x0042F8C	chrome.e	0x0000067	0x0000023	0x0000007	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:57:4	48	kernel32.(	GetProcA(	C:\Documents and Settings\VMuser\Local Se			
13	13	In	GetProcA(	0x0000000(	0x0042F8E	chrome.e	0x0000067	0x0000023	0x0000007	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:57:4	48	kernel32.(	GetProcA(	C:\Documents and Settings\VMuser\Local Se			
14	14	In	GetProcA(	0x0000000(	0x0042F8E	chrome.e	0x0000067	0x0000023	0x0000007	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:57:4	48	kernel32.(	GetProcA(	C:\Documents and Settings\VMuser\Local Se			
15	15	In	GetModul	0x7C8000(	0x0042F5C	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0xfff	EAX=0x7cl	-1.#IND	22:29:57:4	200	kernel32.(	GetModul	C:\Documents and Settings\VMuser\Local Se			
16	16	In	GetModul	0x0040000	0x0042F48	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:57:4	45	kernel32.(	GetModul	C:\Documents and Settings\VMuser\Local Se			
17	17	In	GetProcA(	0x7C9139(	0x0042F51	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0x00(	EAX=0x7cl	-1.#IND	22:29:57:4	55	kernel32.(	GetProcA(	C:\Documents and Settings\VMuser\Local Se			
18	18	In	GetModul	0x7C8000(	0x0042F5C	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0xfff	EAX=0x7cl	-1.#IND	22:29:57:4	200	kernel32.(	GetModul	C:\Documents and Settings\VMuser\Local Se			
19	19	In	GetModul	0x0040000	0x0042F48	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:57:4	46	kernel32.(	GetModul	C:\Documents and Settings\VMuser\Local Se			
20	20	In	GetProcA(	0x7C9139(	0x0042F51	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0x00(	EAX=0x7cl	-1.#IND	22:29:57:4	51	kernel32.(	GetProcA(	C:\Documents and Settings\VMuser\Local Se			
21	21	In	GetModul	0x7C8000(	0x0042F5C	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0xfff	EAX=0x7cl	-1.#IND	22:29:57:4	204	kernel32.(	GetModul	C:\Documents and Settings\VMuser\Local Se			
22	22	In	GetModul	0x0040000	0x0042F48	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0x00(	EAX=0x00(	-1.#IND	22:29:57:4	46	kernel32.(	GetModul	C:\Documents and Settings\VMuser\Local Se			
23	23	In	GetProcA(	0x7C9139(	0x0042F51	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0x00(	EAX=0x7cl	-1.#IND	22:29:57:4	52	kernel32.(	GetProcA(	C:\Documents and Settings\VMuser\Local Se			
24	24	In	GetModul	0x7C8000(	0x0042F5C	chrome.e	0x0000067	0x0000023	0x0000000	EAX=0xfff	EAX=0x7cl	-1.#IND	22:29:57:4	200	kernel32.(	GetModul	C:\Documents and Settings\VMuser\Local Se			

Πίνακας 5.2 Ενδεικτικό παράδειγμα του Log File

Το κάθε log αρχείο εισάγεται σε ένα φύλλο της MS Excel και το καινούριο αρχείο περνά από ένα πρόγραμμα, όπου η έξοδος του είναι μια ανάλυση του πόσες φορές καθένα από τα system calls εκτελέστηκε. Το συγκεκριμένο πρόγραμμα είναι υλοποιημένο σε JAVA και αποτέλεσμα είναι η δημιουργία μιας συνοπτικής ανάλυσης των αποτελεσμάτων όπου παρουσιάζεται ο συνολικός αριθμός παρουσίασης του κάθε system call για κάθε εφαρμογή.

	A	B	C	D	E	F	G	H	I	J	K	L
1		Adobe Rea	Azureus	BitLord	bsplayer	calculator	camfrog cl	cometbird	FoxitReadi	Gimp	GOMplaye	Google Ch
2	accept	0	3	0	0	0	0	2	0	0	0	0
3	CloseHanc	85	130	578	268	3	535	394	1	16	505	169
4	closesock	0	3	8	0	0	0	8	0	0	0	2
5	coCreateG	0	0	0	0	0	0	0	0	0	0	0
6	coCreateIn	0	0	0	0	0	0	0	0	0	0	0
7	coGetClas	0	0	0	0	0	0	0	0	0	0	0
8	coGetTrea	0	0	0	0	0	0	0	0	0	0	0
9	coInitialize	0	0	0	0	0	0	0	0	0	0	0
10	coMarshal	0	0	0	0	0	0	0	0	0	0	0
11	CopyFileA	0	0	0	0	0	0	0	0	0	0	0
12	CopyFileW	0	0	0	0	0	1	0	0	0	0	1
13	coRegister	0	0	0	0	0	0	0	0	0	0	0
14	coUnmarsl	0	0	0	0	0	0	0	0	0	0	0
15	CreateEve	2	193	11	40	0	13	11	1	9	9	12
16	CreateEve	6	65	471	0	0	407	122	0	0	414	62
17	CreateFile	1	18	22	36	0	8	5	0	0	6	0
18	CreateFile	29	50	37	146	0	40	102	1	0	106	40
19	CreateMut	2	1	6	9	0	8	8	0	0	4	2
20	CreateMut	2	0	7	56	0	11	4	2	0	6	5
21	CreatePro	0	0	0	0	0	0	0	0	0	0	0
22	CreatePro	0	0	0	0	0	0	0	0	0	0	0
23	CreateThre	1	31	14	1	0	7	29	1	0	1	10
24	DeleteFile	0	1	3	4	0	0	0	0	0	0	0
25	DeleteFile	8	2	0	0	0	10	8	0	0	0	4
26	ExitProces	1	0	1	1	0	1	1	1	0	0	1
27	ExitThread	0	0	0	0	0	0	0	0	0	0	0
28	FindFirstFi	0	45	0	24	0	0	3	0	0	0	0
29	FindFirstFi	307	159	6	0	0	2	15	1	0	3	0
30	FindNextF	0	5	0	53	0	0	0	0	0	0	0

Πίνακας 5.3 Ενδεικτικό παράδειγμα της συνοπτικής ανάλυσης

Η πιο πάνω διαδικασία γίνεται για ένα σύνολο συνηθισμένων εφαρμογών και για ένα σύνολο κακόβουλων εφαρμογών, που όπως αναφέρθηκε και πιο πάνω είναι 57 συνηθισμένες εφαρμογές και 100 κακόβουλες. Η κάθε εφαρμογή εκτελείται σε ένα συγκεκριμένο χρονικό διάστημα μετά τη φόρτωση της, συγκεκριμένα 1 λεπτό. Σε αυτό το χρονικό διάστημα καταγράφονται στα log αρχεία, όσα system calls εκτελούνται.

Σημαντικό είναι επίσης να αναφερθεί ότι τα προγράμματα εκτελούνται σε ένα ελεγχόμενο περιβάλλον. Συγκεκριμένα τα προγράμματα εκτελούνται σε μια εικονική μηχανή με λειτουργικό Windows XP. Κάθε εφαρμογή εγκαθίσταται ξεχωριστά από τα άλλα και μετά την εκτέλεσή της και την δημιουργία του log αρχείου, η εικονική μηχανή επαναφέρεται σε ρυθμίσεις πριν την εγκατάσταση. Έτσι η μηχανή «καθαρίζεται» μετά την εκτέλεση κάποιας κακόβουλης εφαρμογής.

Μετά την ολοκλήρωση της εκτέλεσης όλων των προγραμμάτων έχουμε ως αποτέλεσμα 2 σύνολα. Κάθε σύνολο περιέχει ένα αριθμό vector, τον συνολικό αριθμό των εφαρμογών που εκτελέστηκαν. Κάθε vector αντιπροσωπεύει τον αριθμό εμφάνισης των system calls για την κάθε εφαρμογή. Έτσι μπορούν να απεικονιστούν ως 2 δισδιάστατοι πίνακες όπου το B αναφέρεται στις συνηθισμένες εφαρμογές και το V στις κακόβουλες.

$$B= (b_{i,1}, b_{i,2}, b_{i,3} \ldots\ldots b_{i,n})$$

όπου το $b_{i,k}$ αναφέρεται σε φυσιολογική εφαρμογή, το i =1…56 και το n=87

$$V = (v_{j,1}, v_{j,2}, v_{j,3} \ldots\ldots v_{j,n})$$

όπου το $v_{i,k}$ αναφέρεται σε κακόβουλη εφαρμογή, το j =1…100 και το n=87

5.2.2 Ανάλυση Δεδομένων

Η ανάλυση των δεδομένων για να γίνει ο διαχωρισμός τους σε φυσιολογικές και κακόβουλες εφαρμογές, ώστε να ομαδοποιηθούν και να μπορεί να καθοριστεί ένα όριο μπορεί να γίνει με διάφορες μεθόδους. Όμως σύμφωνα με την προηγούμενη ερευνητική εργασία, όπου εξετάστηκαν το Principal Component Analysis και μέθοδοι ομαδοποίησης, ο γωνιακός διαχωρισμός και η ευκλείδεια απόσταση φάνηκε ότι η κατάλληλη μέθοδος είναι το Logistic Regression σε συνδυασμό με τη μέθοδο Principal Component Analysis.

Η ανάλυση των δεδομένων έχει ως στόχο τη δημιουργία μοντέλου με μικρό σύνολο ενδεικτικών μεταβλητών με μεγάλο ποσοστό πρόβλεψης

5.2.2.1 Logistic Regression

Το Logistic Regression [2] [6] είναι μια στατιστική μέθοδος, η οποία μπορεί να χρησιμοποιηθεί για να προβλέψει μια εξαρτημένη μεταβλητή στηριζόμενη σε ένα σύνολο ανεξάρτητων μεταβλητών. Χρησιμοποιείται για να καθορίσει τη σχέση του ανεξάρτητων μεταβλητών με την εξαρτημένη, όπως και να καθορίσει τη σημαντικότητα της κάθε ανεξάρτητης μεταβλητής. Οι ανεξάρτητες μεταβλητές μπορεί να είναι κατηγοριοποιημένες ή συνεχείς.

Η μέθοδος Logistic Regression χρησιμοποιείται κυρίως σε ιατρικές έρευνες για να καθοριστεί κατά πόσο ένας ασθενής πάσχει ή όχι από κάποιο νόσημα. Η εξαρτημένη μεταβλητή παίρνει την τιμή 1 όταν πάσχει από κάποιο νόσημα και την τιμή 0 όταν δεν πάσχει. Οι εξαρτημένες μεταβλητές αποτελούνται από τα διάφορες μετρικές-συμπτώματα που μπορεί να οδηγούν στην εμφάνιση κάποιου νοσήματος.

Στην συγκεκριμένη εργασία χρησιμοποιείται για να καθοριστεί αν μία εφαρμογή είναι κακόβουλη ή όχι. Η εξαρτημένη μεταβλητή παίρνει την τιμή 1 και 0 ανάλογα με το αν η εφαρμογή είναι κακόβουλη ή όχι αντίστοιχα. Οι ανεξάρτητες μεταβλητές είναι ένα μικρό σύνολο από συγκεκριμένες τιμές που δημιουργούνται με την μέθοδο Principal Component Analysis.

Η συγκεκριμένη μέθοδος αποτελείται από 2 φάσεις, τη φάση εκπαίδευσης και τη φάση αξιολόγησης. Η φάση εκπαίδευσης δημιουργεί ένα μοντέλο το οποίο θα χρησιμοποιηθεί στη φάση αξιολόγησης. Καθορίζει κατά πόσο η συγκεκριμένη μέθοδος μπορεί να χρησιμοποιηθεί για την διάκριση κακόβουλων εφαρμογών από φυσιολογικές και ποιες ανεξάρτητες μεταβλητές είναι οι πιο σημαντικές για να χρησιμοποιηθούν. Η φάση αξιολόγησης εξετάζει τη λειτουργικότητα του μοντέλου με τη χρησιμοποίηση ενός διαφορετικού συνόλου δεδομένων. Πιο κάτω παρουσιάζεται το μοντέλο που μπορεί να δημιουργηθεί από τη φάση εκπαίδευσης.

$$P = \frac{e^{\alpha + b_1 x_1 + \ldots + b_n x_n}}{1 + e^{\alpha + b_1 x_n + \ldots + b_n x_n}}$$

όπου το n είναι ο αριθμός των ανεξάρτητων μεταβλητών, το x_i αντιπροσωπεύει την τιμή της i–οστής ανεξάρτητης μεταβλητής, το a είναι η σταθερά και το b_i είναι ο συντελεστής της i-οστής ανεξάρτητης μεταβλητής. Το α και το b καθορίζονται κατά τη φάση εκπαίδευσης.

Στόχος του Logistic Regression είναι η δημιουργία ενός μοντέλου, το οποίο θα περιέχει τα απαραίτητα στοιχεία ώστε να αντικατοπτρίζονται όσο καλύτερα γίνονται τα δεδομένα για να είναι όσο μεγαλύτερο γίνεται το goodness of fit. Για να υπολογιστεί το goodness of fit χρησιμοποιείται το maximum likelihood, το οποίο είναι η μεγαλύτερη πιθανότητα εμφάνισης συγκεκριμένης τιμής της εξαρτημένης μεταβλητής βασιζόμενη στο σύνολο των ανεξάρτητων μεταβλητών.

Ένας τρόπος επιλογής των κατάλληλων ανεξάρτητων, ώστε να επιτευχθεί όσο το δυνατό μεγαλύτερη πιθανότητα εμφάνισης μιας μεταβλητής, είναι η μέθοδος trial and error. Η συγκεκριμένη μέθοδος ασχολείται με τη δοκιμή διάφορων συνδυασμών ώστε να βρεθεί το κατάλληλο σύνολο ανεξάρτητων μεταβλητών. Η καλύτερη μέθοδος όμως

που μπορεί να χρησιμοποιηθεί είναι η αξιολόγηση της σημαντικότητας των συντελεστών. Για την εργασία χρησιμοποιήθηκε το p-value και το χ^2 Wald test για να εξεταστεί η υπόθεση του κενού, δηλαδή ότι κάποιος συντελεστής μπορεί να έχει την τιμή 0. Η συγκεκριμένη μέθοδος μπορεί να καθοδηγήσει την επιλογή των ανεξάρτητων μεταβλητών ώστε να επιλεγούν οι πιο αντιπροσωπευτικές του αρχικού συνόλου. Συγκεκριμένα επιλέχτηκαν τα σύνολα των ανεξάρτητων μεταβλητών που παρουσιάζουν σημαντικότητα μικρότερη του 0.1 σε όλες τις μεταβλητές. Αν κάποια μεταβλητή παρουσίαζε μεγαλύτερη σημαντικότητα, τότε το συγκεκριμένο σύνολο δεν χρησιμοποιούνταν.

Πιο κάτω στην υλοποίηση του εργαλείου εξετάστηκαν διάφορα σύνολα για την επιλογή του καλύτερου.

5.2.2.2 Principal Component Analysis (PCA)

Το Principal Component Analysis [17] είναι μια στατιστική μέθοδος, η οποία μειώνει τις διαστάσεις των δεδομένων χωρίς να υπάρχει απώλεια πληροφορίας και βρίσκει πρότυπα σε αυτά ώστε να καθορίσει τις ομοιότητες και τις διαφορές τους.

Η μέθοδος PCA παίρνει ένα σύνολο δεδομένων όπως το B ή το V που ορίστηκαν πιο πάνω. Καθορίζει το X που είναι ο ανάστροφος πίνακας του αρχικού συνόλου, ώστε κάθε γραμμή του πίνακα να αντιστοιχεί σε κάθε εφαρμογή και οι στήλες στον αριθμό εμφάνισης του κάθε system call. Στόχος του PCA είναι η εύρεση του πίνακα Y, όπου οι μεταβλητές είναι γραμμικές και είναι ταξινομημένες ανάλογα με τη σημαντικότητά τους.

$$Y = A \times \Lambda^T$$

όπου Λ: κανονικοποιημένος πίνακας X και A: ανάστροφος ταξινομημένος πίνακας eigenvectors

Αρχικά υπολογίζεται ο πίνακας Λ, που όπως αναφέρθηκε είναι ο πίνακας X κανονικοποιημένος. Έτσι αρχικά για κάθε system call θα υπολογιστεί ο μέσος όρος εμφάνισης της σε όλες τις εφαρμογές. Έτσι χρησιμοποιείται ο πιο κάτω τύπος, όπου το j= 1….87 (αριθμός system calls) και το m είναι ο αριθμός των εφαρμογών.

$$\bar{x}_j = \frac{1}{m}\sum_{i=1}^{m} x_{i,j}$$

Για τον υπολογισμό του Λ χρησιμοποιείται ο πιο κάτω τύπος.

$\Lambda = [\lambda_{i,j}]_{m\times n}$, όπου $\lambda_{i,j} = x_{i,j} - \bar{x}_j$

Για τον υπολογισμό του πίνακα Α θα ακολουθηθεί η πιο κάτω διαδικασία. Αρχικά υπολογίζουμε το variance, δηλαδή πώς 2 μεταβλητές μεταβάλλονται μαζί. Το $X^{<i>}$ αντιστοιχεί σε κάποια στήλη του πίνακα Χ, έτσι:

$$C = [c_{i,j}]_{n\times n} = \left[\frac{X^{<i>}\cdot X^{<j>}}{n} - \left(\bar{x}_i \cdot \bar{x}_j\right)\right]_{n\times n}$$

Με βάση τον πίνακα C υπολογίζονται 2 πίνακες. Ο πίνακας Ε και ο πίνακας D. Ο πίνακας Ε αντιστοιχεί στα eigenvectors και ο πίνακας D στα eigenvalues. Το επόμενο βήμα είναι η ταξινόμηση του πίνακα Ε με βάση τα eigenvalues από την μεγαλύτερη τιμή στη μικρότερη. Αυτό έχει ως στόχο να δοθούν τα στοιχεία του Ε με βάση τη σημαντικότητά τους. Σε αυτό το σημείο μπορεί να παραληφθούν τα σημεία που είναι λιγότερο σημαντικά και να γίνει μείωση των διαστάσεων. Στην εργασία χρησιμοποιούνται όλα τα στοιχεία χωρίς να γίνεται μείωση των διαστάσεων για να υπάρχει καλύτερη εικόνα του διαχωρισμού των δεδομένων και το αποτέλεσμα να είναι πιο αντιπροσωπευτικό. Το Α υπολογίζεται από τον ανάστροφο ταξινομημένο πίνακα Ε.

Μετά και τον υπολογισμό του Α μπορούμε να υπολογίσουμε το Y με τον τύπο που αναφέρθηκε πιο πάνω.

Η διαδικασία του PCA υλοποιήθηκε με κώδικα στη Matlab.

5.3 Υλοποίηση Εργαλείου Ανίχνευσης

Για την υλοποίηση του εργαλείου ανίχνευσης όπως αναφέρθηκε και πιο πάνω χρησιμοποιήθηκε η μέθοδος Logistic Regression σε συνδυασμό με την μέθοδο του Principal Component Analysis. Για τη δημιουργία του μοντέλου ανίχνευσης

χρησιμοποιείται το στατιστικό πρόγραμμα SPSS. Επίσης θα χρησιμοποιηθούν τα 2 σύνολα των φυσιολογικών και κακόβουλων εφαρμογών που δημιουργήθηκαν κατά τη διαδικασία συλλογής δεδομένων. Αρχικά τα στοιχεία των 2 συνόλων τροποποιούνται, ώστε σε κάθε εφαρμογή να παρουσιάζεται το ποσοστό εμφάνισης κάθε system call. Με αυτή την τροποποίηση έχουμε τους 2 πιο κάτω πίνακες.

$$T = (t_{i,1}, t_{i,2}, t_{i,3} \ldots\ldots t_{i,n})$$

όπου το $t_{i,k}$ αναφέρεται στην κανονικοποιημένη εφαρμογή, το i =1…56 και το n=87

$$U = (u_{i,1}, u_{i,2}, u_{i,3} \ldots\ldots u_{i,n})$$

όπου το $u_{i,k}$ αναφέρεται στην κανονικοποιημένη κακόβουλη εφαρμογή, το j =1…100 και το n=87

Μετά την τροποποίηση των συνόλων γίνεται τυχαίος διαχωρισμός των συνόλων ώστε να δημιουργηθούν 2 άλλα για την εκπαίδευση και αξιολόγηση του εργαλείου. Τα 2 σύνολα αποτελούνται από φυσιολογικές εφαρμογές και από κακόβουλες.

$$S1= (t_{i,1},t_{i,2},\ldots.t_{28,n},u_{j,1},u_{j,2},\ldots..u_{50,n})$$

$$S2= (t_{i,1},t_{i,2},\ldots.t_{28,n},u_{j,1},u_{j,2},\ldots..u_{50,n})$$

Τα 2 πιο πάνω σύνολα θα χρησιμοποιηθούν για την δημιουργία 2 διαφορετικών μοντέλων. Αρχικά χρησιμοποιείται το σύνολο S1 για την εκπαίδευση και το σύνολο S2 για την αξιολόγηση, αλλά αργότερα χρησιμοποιείται το σύνολο S2 για εκπαίδευση και το S1 για την αξιολόγηση.

5.3.1 Υλοποίηση Πρώτου Εργαλείου Ανίχνευσης

Για την υλοποίηση του πρώτου εργαλείου ανίχνευσης χρησιμοποιείται το σύνολο S1 για την εκπαίδευση και το σύνολο S2 για την αξιολόγηση του μοντέλου. Συγκεκριμένα καθορίζονται οι εξής πίνακες:

$$X_{Training=} S1^{T}$$

$$X_{Evaluation=} S2^{T}$$

5.3.1.1 Φάση Εκπαίδευσης:

Κατά τη φάση εκπαίδευσης δημιουργείται το μοντέλο που θα χρησιμοποιηθεί στη φάση αξιολόγησης. Καθορίζει ποιες ανεξάρτητες μεταβλητές είναι οι πιο σημαντικές για να χρησιμοποιηθούν και καθορίζονται και οι τιμές των συντελεστών και της σταθεράς.

Αρχικά εφαρμόζεται η μέθοδος Principal Component Analysis στον πίνακα $X_{Training}$ για να βρεθεί ο πίνακας $Y_{Training}$, που περιέχει γραμμικές μεταβλητές ταξινομημένες ανάλογα με τη σημαντικότητά τους. Η διαδικασία που ακολουθείται για την εύρεση του πίνακα $Y_{Training}$ περιγράφηκε πιο πάνω.

Οι μεταβλητές του πίνακα $Y_{Training}^{T}$ χρησιμοποιήθηκαν ως οι ανεξάρτητες μεταβλητές του μοντέλου. Επίσης χρησιμοποιήθηκε και η εξαρτημένη μεταβλητή viral που καθορίζει κατά πόσο μια εφαρμογή είναι κακόβουλη ή όχι για να δημιουργηθεί το μοντέλο. Έτσι με τη μέθοδο που περιγράφηκε και πιο πάνω για την επιλογή των ανεξάρτητων μεταβλητών παρουσιάζονται 2 διαφορετικά σύνολα με υψηλό ποσοστό πρόβλεψης, αλλά και με το μικρότερο σύνολο ανεξάρτητων μεταβλητών. Επιλέγονται 2 διαφορετικά μοντέλα για να παρατηρηθεί η συμπεριφορά τους κατά τη φάση αξιολόγησης.

Πιο κάτω παρουσιάζονται τα 2 μοντέλα που δημιουργήθηκαν.

Στο πρώτο σύνολο παρουσιάζεται πρόβλεψη κακόβουλων εφαρμογών κατά 98% με 2% false negatives. Μπορούν να ανιχνευτούν όλες οι κακόβουλες εφαρμογές εκτός από μια. Επίσης παρουσιάζεται ένα ποσοστό 3,6% false positives. Σύμφωνα με τα αποτελέσματα πρόβλεψης μπορεί να θεωρηθεί ότι το μοντέλο που παράγεται θα είναι σε θέση να ανιχνεύει καινούριες επιθέσεις που παρουσιάζονται στο σύστημα.

Στους πίνακες που ακολουθούν παρουσιάζονται τα αποτελέσματα που αναφέρθηκαν, το σύνολο των ανεξάρτητων μεταβλητών που χρησιμοποιήθηκαν για την κατασκευή του μοντέλου καθώς και οι τιμές των συντελεστών και τις σταθεράς.

Classification Table[a]

Observed			Predicted		
			Viral		Percentage Correct
			0	1	
Step 1	Viral	0	27	1	96,4
		1	1	49	98,0
	Overall Percentage				97,4

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	Y2	109,048	37,746	8,346	1	,004	2,285E47
	Y3	128,018	45,588	7,886	1	,005	3,959E55
	Y4	109,954	38,695	8,074	1	,004	5,654E47
	Y10	80,052	26,958	8,818	1	,003	5,838E34
	Constant	1,953	1,042	3,510	1	,061	7,049

a. Variable(s) entered on step 1: Y2, Y3, Y4, Y10.

Πίνακας 5.4 και 5.5 Επίδοση μοντέλου πρόβλεψης και Μεταβλητές της εξίσωσης

Με βάση τα πιο πάνω παρουσιάζεται το μοντέλο:

$$P = \frac{e^{\text{constant} + b2Y2 + b3Y3 + b4Y4 + b10Y10}}{1 + e^{\text{constant} + b2Y2 + b3Y3 + b4Y4 + b10Y10}}$$

Όπου το b_i αναφέρεται στο συντελεστή της κάθε μεταβλητής i του μοντέλου και το Yi αναφέρεται στην ανεξάρτητη μεταβλητή που μπορεί να βρεθεί από το πίνακα Y^T.

Στο δεύτερο σύνολο παρουσιάζεται πρόβλεψη κακόβουλων εφαρμογών κατά 94% με 6% false negatives. Μπορούν να ανιχνευτούν όλες οι κακόβουλες εφαρμογές εκτός από 3. Επίσης παρουσιάζεται ένα ποσοστό 3,6% false positives το οποίο είναι το ίδιο με το πρώτο μοντέλο. Σύμφωνα με τα αποτελέσματα πρόβλεψης το δεύτερο μοντέλο παρουσιάζει μεγαλύτερο ποσοστό false negatives από το πρώτο, όμως μπορεί να θεωρηθεί ότι και το δεύτερο μοντέλο μπορεί να ανιχνεύσει ένα αριθμό νέων κακόβουλων εφαρμογών.

Στους πίνακες που ακολουθούν παρουσιάζονται τα αποτελέσματα που αναφέρθηκαν, το σύνολο των ανεξάρτητων μεταβλητών που χρησιμοποιήθηκαν για την κατασκευή του μοντέλου καθώς και οι τιμές των συντελεστών και της σταθεράς.

Classification Table[a]

Observed			Predicted		
			Viral		Percentage Correct
			0	1	
Step 1	Viral	0	27	1	96,4
		1	3	47	94,0
	Overall Percentage				94,9

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	Y1	9,871	4,960	3,961	1	,047	19369,253
	Y2	30,117	16,044	3,524	1	,061	1,201E13
	Y3	35,234	19,216	3,362	1	,067	2,005E15
	Y4	28,115	16,406	2,937	1	,087	1,622E12
	Y30	-1044,905	529,541	3,894	1	,048	,000
	Constant	1,628	1,094	2,213	1	,137	5,095

a. Variable(s) entered on step 1: Y1, Y2, Y3, Y4, Y30.

Πίνακας 5.6 και 5.7 Επίδοση μοντέλου πρόβλεψης και Μεταβλητές της εξίσωσης

Με βάση τα πιο πάνω παρουσιάζεται το μοντέλο:

$$P= \frac{e^{\text{constant} + b1Y1 + b2Y2 + b3Y3 + b4Y4 + b30Y30}}{1 + e^{\text{constant} + b1Y1 + b2Y2 + b3Y3 + b4Y4 + b30Y30}}$$

Όπου το b_i αναφέρεται στο συντελεστή της κάθε μεταβλητής i του μοντέλου και το Yi αναφέρεται στην ανεξάρτητη μεταβλητή που μπορεί να βρεθεί από το πίνακα Y^T.

Το όριο που ορίζεται και στα 2 μοντέλα είναι το 0.5. Αν η πιθανότητα παρουσίασης είναι μεγαλύτερη από 0.5 τότε η εφαρμογή θεωρείται κακόβουλη, ενώ αν είναι μικρότερη θεωρείται φυσιολογική.

5.3.1.2 Φάση Αξιολόγησης:

Κατά τη φάση αξιολόγησης εξετάζεται κατά πόσο το μοντέλο που δημιουργήθηκε είναι λειτουργικό με τη χρησιμοποίηση ενός διαφορετικού συνόλου δεδομένων.

Αρχικά εφαρμόζεται η μέθοδος Principal Component Analysis στον πίνακα $X_{Evaluation}$ για να βρεθεί ο πίνακας $Y_{Evaluation}$, που περιέχει γραμμικές μεταβλητές ταξινομημένες ανάλογα με τη σημαντικότητά τους. Για την εύρεση του πίνακα $Y_{Evaluation}$ ακολουθείται

η διαδικασία που περιγράφηκε πιο πάνω, όμως χρησιμοποιείται ο πίνακας $A_{Training,}$ για να συσχετιστούν οι μεταβλητές αξιολόγησης με τις μεταβλητές εκπαίδευσης.

Για την αξιολόγηση του μοντέλου χρησιμοποιούνται οι συντελεστές που υπολογίστηκαν κατά τη φάση εκπαίδευσης. Επίσης από τον πίνακα $Y_{Evaluation}^{T}$ θα χρησιμοποιηθούν το ίδιο σύνολο μεταβλητών που χρησιμοποιήθηκε για την κατασκευή των μοντέλων.

Πιο κάτω παρουσιάζονται τα αποτελέσματα των 2 μοντέλων.

		Prediction		
		VIRAL Cut 0.5		%Correct
Observed		0	1	
Viral	0	25	3	89.3
	1	32	18	36

Πίνακας 5.8 Αποτελέσματα Αξιολόγησης πρώτου μοντέλου

		Prediction		
		VIRAL Cut 0.5		%Correct
Observed		0	1	
Viral	0	25	3	89.3
	1	2	48	96

Πίνακας 5.9 Αποτελέσματα Αξιολόγησης δεύτερου μοντέλου

Από τα πιο πάνω αποτελέσματα παρατηρούμε ότι το πρώτο μοντέλο δεν ανταποκρίνεται στην πρόβλεψη που είχε γίνει κατά την φάση εκπαίδευσης. Συγκεκριμένα παρουσιάζει πρόβλεψη κακόβουλων εφαρμογών με ποσοστό 36%. Παρουσιάζεται ένα μεγάλο ποσοστό λάθους για τα false negatives με αποτέλεσμα πολλές κακόβουλες εφαρμογές να περνούν απαρατήρητες. Το ποσοστό των false positives αυξήθηκε σε σχέση με το ποσοστά πρόβλεψης, όμως είναι σχετικά μικρό με ποσοστό 10.7%.

Στο δεύτερο μοντέλο παρουσιάζονται καλύτερα αποτελέσματα από ότι στο πρώτο μοντέλο. Συγκεκριμένα παρουσιάζεται πρόβλεψη κακόβουλων εφαρμογών κατά 96%

με ποσοστό λάθους 4% για false negatives και 10,7% για false positives. Το ποσοστό false positives μπορεί να είναι μεγαλύτερο από ότι στο μοντέλο πρόβλεψης, όμως δεν αποτελεί κάποιο σοβαρό πρόβλημα.

5.3.2 Υλοποίηση Δεύτερου Εργαλείου Ανίχνευσης

Για την υλοποίηση του δεύτερου εργαλείου ανίχνευσης χρησιμοποιείται το σύνολο S2 για την εκπαίδευση και το σύνολο S1 για την αξιολόγηση του μοντέλου. Συγκεκριμένα καθορίζονται οι εξής πίνακες:

$$X_{Training=} S2^{T}$$

$$X_{Evaluation=} S1^{T}$$

5.3.2.1 Φάση Εκπαίδευσης:

Κατά τη φάση εκπαίδευσης δημιουργείται το μοντέλο που θα χρησιμοποιηθεί στη φάση αξιολόγησης. Καθορίζει όπως αναφέρθηκε και πιο πάνω ποιες ανεξάρτητες μεταβλητές είναι οι πιο σημαντικές για να χρησιμοποιηθούν και καθορίζονται και οι τιμές των συντελεστών και της σταθεράς.

Αρχικά εφαρμόζεται η μέθοδος Principal Component Analysis στον πίνακα $X_{Training}$ για να βρεθεί ο πίνακας $Y_{Training}$, που περιέχει γραμμικές μεταβλητές ταξινομημένες ανάλογα με τη σημαντικότητά τους. Η διαδικασία που ακολουθείται για την εύρεση του πίνακα $Y_{Training}$ περιγράφηκε πιο πάνω.

Οι μεταβλητές του πίνακα $Y_{Training}^{T}$ χρησιμοποιήθηκαν ως οι ανεξάρτητες μεταβλητές του μοντέλου. Επίσης χρησιμοποιήθηκε και η εξαρτημένη μεταβλητή viral που καθορίζει κατά πόσο μια εφαρμογή είναι κακόβουλη ή όχι για να δημιουργηθεί το μοντέλο. Έτσι με τη μέθοδο που περιγράφηκε και πιο πάνω για την επιλογή των ανεξάρτητων μεταβλητών παρουσιάζονται 2 διαφορετικών συνόλων με υψηλό ποσοστό πρόβλεψης και τα μικρότερα σύνολα ανεξάρτητων μεταβλητών.

Πιο κάτω παρουσιάζονται τα 2 μοντέλα που δημιουργήθηκαν.

Στο πρώτο σύνολο παρουσιάζεται πρόβλεψη κακόβουλων εφαρμογών κατά 96% με 4% false negatives. Επίσης παρουσιάζεται ένα ποσοστό 3,6% false positives. Σύμφωνα με

τα αποτελέσματα πρόβλεψης μπορεί να θεωρηθεί ότι το μοντέλο που παράγεται θα είναι σε θέση να ανιχνεύει καινούριες επιθέσεις που παρουσιάζονται στο σύστημα σε αρκετά ικανοποιητικό βαθμό.

Στους πίνακες που ακολουθούν παρουσιάζονται τα αποτελέσματα που αναφέρθηκαν, το σύνολο των ανεξάρτητων μεταβλητών που χρησιμοποιήθηκαν για την κατασκευή του μοντέλου καθώς και οι τιμές των συντελεστών και τις σταθεράς.

Classification Table[a]

Observed			Predicted		
			Viral		
			0	1	Percentage Correct
Step 1	Viral	0	27	1	96,4
		1	2	48	96,0
	Overall Percentage				96,2

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	Y1	5,193	2,358	4,848	1	,028	179,948
	Y2	-22,351	8,917	6,284	1	,012	,000
	Y6	14,807	6,902	4,603	1	,032	2695235,325
	Constant	,892	,676	1,740	1	,187	2,439

a. Variable(s) entered on step 1: Y1, Y2, Y6.

Πίνακας 5.10 και 5.11 Επίδοση μοντέλου πρόβλεψης και Μεταβλητές της εξίσωσης

Με βάση τα πιο πάνω παρουσιάζεται το μοντέλο:

$$P = \frac{e^{\text{constant} + b1Y1 + b2Y2 + b6Y6}}{1 + e^{\text{constant} + b1Y1 + b2Y2 + b6Y6}}$$

Όπου το b_i αναφέρεται στο συντελεστή της κάθε μεταβλητής i του μοντέλου και το Yi αναφέρεται στην ανεξάρτητη μεταβλητή που μπορεί να βρεθεί από το πίνακα Y^T.

Στο δεύτερο σύνολο παρουσιάζεται πρόβλεψη κακόβουλων εφαρμογών κατά 96% με 4% false negatives. Μπορούν να ανιχνευτούν όλες οι κακόβουλες εφαρμογές εκτός από 2. Επίσης παρουσιάζεται ένα ποσοστό 7.1% false positives. Τα δύο μοντέλα παρουσιάζουν το ίδιο ποσοστό πρόβλεψης κακόβουλων εφαρμογών, αλλά διαφέρουν στην αναγνώριση των φυσιολογικών εφαρμογών. Το δεύτερο μοντέλο παρουσιάζει μεγαλύτερο ποσοστό false positives. Στους πίνακες που ακολουθούν παρουσιάζονται τα

αποτελέσματα που αναφέρθηκαν, το σύνολο των ανεξάρτητων μεταβλητών που χρησιμοποιήθηκαν για την κατασκευή του μοντέλου καθώς και οι τιμές των συντελεστών και τις σταθεράς.

Classification Table[a]

Observed			Predicted		
			Viral		
			0	1	Percentage Correct
Step 1	Viral	0	26	2	92,9
		1	2	48	96,0
	Overall Percentage				94,9

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	Y2	-60,027	16,214	13,707	1	,000	,000
	Y6	37,777	10,477	13,000	1	,000	2,548E16
	Y24	-299,295	116,196	6,635	1	,010	,000
	Constant	2,264	,787	8,268	1	,004	9,625

a. Variable(s) entered on step 1: Y2, Y6, Y24.

Πίνακας 5.12 και 5.13 Επίδοση μοντέλου πρόβλεψης και Μεταβλητές της εξίσωσης

Με βάση τα πιο πάνω παρουσιάζεται το μοντέλο:

$$P = \frac{e^{\text{constant} + b2Y2 + b6Y6 + b24Y24}}{1 + e^{\text{constant} + b2Y2 + b6Y6 + b24Y24}}$$

Όπου το b_i αναφέρεται στο συντελεστή της κάθε μεταβλητής i του μοντέλου και το Yi αναφέρεται στην ανεξάρτητη μεταβλητή που μπορεί να βρεθεί από το πίνακα Y^T.

Το όριο που ορίζεται και στα 2 μοντέλα είναι το 0.5. Αν η πιθανότητα παρουσίασης είναι μεγαλύτερη από 0.5 τότε η εφαρμογή θεωρείται κακόβουλη, ενώ αν είναι μικρότερη θεωρείται φυσιολογική.

5.3.2.2 Φάση Αξιολόγησης:

Κατά τη φάση αξιολόγησης εξετάζεται κατά πόσο το μοντέλο που δημιουργήθηκε είναι λειτουργικό με τη χρησιμοποίηση ενός διαφορετικού συνόλου δεδομένων.

Αρχικά εφαρμόζεται η μέθοδος Principal Component Analysis στον πίνακα $X_{Evaluation}$ για να βρεθεί ο πίνακας $Y_{Evaluation}$, που περιέχει γραμμικές μεταβλητές ταξινομημένες ανάλογα με τη σημαντικότητά τους. Για την εύρεση του πίνακα $Y_{Evaluation}$ ακολουθείται η διαδικασία που περιγράφηκε πιο πάνω, όμως χρησιμοποιείται ο πίνακας $A_{Training,}$ για να συσχετιστούν οι μεταβλητές αξιολόγησης με τις μεταβλητές εκπαίδευσης.

Για την αξιολόγηση του μοντέλου χρησιμοποιούνται οι συντελεστές που υπολογίστηκαν κατά τη φάση εκπαίδευσης. Επίσης από τον πίνακα $Y_{Evaluation}^{T}$ θα χρησιμοποιηθούν το ίδιο σύνολο μεταβλητών που χρησιμοποιήθηκε για την κατασκευή των μοντέλων.

Πιο κάτω παρουσιάζονται τα αποτελέσματα των 2 μοντέλων.

		Prediction		
		VIRAL Cut 0.5		%Correct
Observed		0	1	
Viral	0	26	2	92.9
	1	5	45	90

Πίνακας 5.14 Αποτελέσματα Αξιολόγησης πρώτου μοντέλου

		Prediction		
		VIRAL Cut 0.5		%Correct
Observed		0	1	
Viral	0	23	5	82.2
	1	6	44	88

Πίνακας 5.15 Αποτελέσματα Αξιολόγησης δεύτερου μοντέλου

Από τα πιο πάνω αποτελέσματα παρατηρούμε ότι το πρώτο μοντέλο ανταποκρίνεται σε μεγάλο βαθμό στην πρόβλεψη που είχε γίνει κατά την φάση εκπαίδευσης. Συγκεκριμένα παρουσιάζει πρόβλεψη κακόβουλων εφαρμογών με ποσοστό 90%. Το ποσοστό των false positives παρουσιάζει ποσοστό ίσο με 7.1%.

Στο δεύτερο μοντέλο δεν παρουσιάζονται τόσο καλά αποτελέσματα όπως το πρώτο μοντέλο. Συγκεκριμένα παρουσιάζεται πρόβλεψη κακόβουλων εφαρμογών κατά 88%, ενώ κατά τη πρόβλεψη εκπαίδευσης παρουσιάστηκε ποσοστό 96%. Το ποσοστό false positives είναι 17.8% και είναι αρκετά μεγάλο σε σχέση με τα υπόλοιπα αποτελέσματα που βρέθηκαν.

5.4 Παρατηρήσεις

Από τα πιο πάνω αποτελέσματα παρατηρούμε αρκετά υψηλά ποσοστά πρόβλεψης. Η μόνη περίπτωση που το ποσοστό πρόβλεψης δεν είναι καθόλου ικανοποιητικό είναι στην πρώτη περίπτωση του πρώτου μοντέλου, όπου το ποσοστό φτάνει μόλις το 36%.

Χρησιμοποιώντας τα 2 δύο σύνολα δεδομένων, τη μια φορά ως σύνολα εκπαίδευσης και την άλλη ως σύνολα αξιολόγησης, τα αποτελέσματα δεν είναι τα ίδια. Για κάθε σύνολο βρίσκονται οι ανεξάρτητες μεταβλητές που αντικατοπτρίζουν καλύτερα τα δεδομένα που δίνονται για την εκπαίδευση του μοντέλου. Τα δεδομένα που χρησιμοποιούνται δεν είναι τα ίδια με αποτέλεσμα να διαφέρουν και τα χαρακτηριστικά τους.

Επίσης μπορούμε να παρατηρήσουμε ότι ένα μικρό σύνολο ανεξάρτητων μεταβλητών μπορεί να μην οδηγήσει σε καλά ποσοστά πρόβλεψης, ενώ αντίθετα ένα εξίσου μικρό σύνολο με κάποια επιπλέον ανεξάρτητη μεταβλητή μπορεί να οδηγήσει σε ακόμα καλύτερα αποτελέσματα, αφού συμπεριλαμβάνει πιο ενδεικτική πληροφορία. Ακόμα μπορούμε να παρατηρήσουμε ότι τα μοντέλα με μεγαλύτερη πιθανότητα πρόβλεψης δεν παρουσιάζουν και το μεγαλύτερο ποσοστό πρόβλεψης στη φάση αξιολόγησης. Μοντέλο με μικρότερη πιθανότητα πρόβλεψης μπορεί να παρουσιάσει καλύτερα αποτελέσματα.

Σε σύγκριση με την προηγούμενη ερευνητική εργασία μπορεί να γίνει η παρατήρηση ότι ένα μεγαλύτερο μέγεθος δεδομένων και ένα μεγαλύτερο σύνολο system calls μπορούν να οδηγήσουν σε καλύτερα αποτελέσματα. Δεν χρησιμοποιήθηκαν ούτε οι ίδιες εφαρμογές ούτε τα ίδια system calls, όμως μπορούμε να υποθέσουμε ότι η επιλογή των δεδομένων για τη συγκεκριμένη εργασία ήταν καλύτερη.

Κεφάλαιο 6

Συμπεράσματα και Μελλοντική Εργασία

6.1 Συμπεράσματα
6.2 Μελλοντική Εργασία

6.1 Συμπεράσματα

Τα συμπεράσματα που μπορούν να εξαχθούν από τη συγκεκριμένη εργασία στηρίζονται κυρίως στα αποτελέσματα που παρουσιάστηκαν με τη δημιουργία των διάφορων μοντέλων.

Αρχικά όπως διαφάνηκε και από προηγούμενες μελέτες, τα system calls μπορούν να χρησιμοποιηθούν για την ανίχνευση άγνωστων προς το σύστημα επιθέσεων. Υλοποιώντας τη μέθοδο ανίχνευσης ανωμαλιών και χρησιμοποιώντας τις κατάλληλες μεθόδους διαχωρισμού των δεδομένων μεταξύ τους, μπορεί να δημιουργηθεί ένα εργαλείο, το οποίο να μπορεί να διακρίνει κατά πόσο μια εφαρμογή είναι κακόβουλη ή όχι. Το Logistic Regression και το Principal Component Analysis είναι 2 μέθοδοι που αποδεικνύουν ότι μπορούν να αντεπεξέλθουν στην πρόκληση για την εύρεση ενός κατάλληλου ορίου.

Η επιλογή των κατάλληλων ανεξάρτητων μεταβλητών που θα χρησιμοποιηθούν για τον καθορισμό ενός διαχωριστικού ορίου μεταξύ φυσιολογικών και κακόβουλων εφαρμογών είναι ένα από τα σημαντικότερα θέματα κατά την υλοποίηση του εργαλείου. Η επιλογή τους πρέπει να γίνεται με τη χρήση κάποιας μεθόδου, ώστε να μπορούν να οδηγήσουν σε ικανοποιητικά αποτελέσματα. Αν και η δημιουργία του μοντέλου ανίχνευσης εισβολών πρέπει να στηρίζεται σε μικρό σύνολο ανεξάρτητων μεταβλητών με το μεγαλύτερο ποσοστό πρόβλεψης, μπορεί κάτι τέτοιο να μην αποφέρει τα κατάλληλα αποτελέσματα. Μεγαλύτερου μεγέθους σύνολα ανεξάρτητων μεταβλητών μπορούν να οδηγήσουν σε καλύτερα αποτελέσματα, μιας και μπορεί να

περιέχουν πιο ενδεικτική πληροφορία των εφαρμογών και να μπορούν να τις διαχωρίζουν σε κακόβουλες και φυσιολογικές με μικρά ποσοστά λάθους. Επίσης μοντέλα που αποτελούνται από τον ίδιο αριθμό ανεξάρτητων μεταβλητών μπορεί να μην παρέχουν τα ίδια ποσοστά πρόβλεψης. Αυτό οφείλεται στην πληροφορία που περιέχει η κάθε ανεξάρτητη μεταβλητή που χρησιμοποιείται για την κατασκευή του μοντέλου ανίχνευσης επιθέσεων.

Ακόμα διαφάνηκε ότι χρησιμοποιώντας διαφορετικά σύνολα δεδομένων, τα αποτελέσματα διαφέρουν. Αυτό συμβαίνει γιατί η κατασκευή του μοντέλου στηρίζεται αποκλειστικά στην ανάλυση των δεδομένων που χρησιμοποιούνται. Διαφορετικά σύνολα δεδομένων περιέχουν διαφορετική πληροφορία. Για αυτό το λόγο παρατηρήσαμε ότι όταν αντιστραφήκαν τα σύνολα που χρησιμοποιούνταν για την εκπαίδευση και την αξιολόγηση των μοντέλων, τα αποτελέσματα δεν είχαν κάποια σχέση μεταξύ τους.

Αναφέροντας τα πιο πάνω είναι σημαντικό να τονιστεί ότι όσο πιο πολλά είναι τα δεδομένα που χρησιμοποιούνται κατά τη φάση εκπαίδευσης τόσο πιο αντιπροσωπευτικά είναι και τα αποτελέσματα. Για παράδειγμα αν μπορούσε να κατασκευαστεί ένα μοντέλο που θα χρησιμοποιούσε κατά την φάση εκπαίδευσής του όλα τα προγράμματα που υπάρχουν, αλλά και όλες τις κακόβουλες εφαρμογές, τα αποτελέσματά του θα μπορούσαν να ανιχνεύσουν ορθά μια οποιαδήποτε εφαρμογή, αφού θα ήταν πιο ενδεικτικά.

Είναι επίσης σημαντικό να αναφερθεί ότι και η επιλογή των system calls είναι μια διαδικασία η οποία συμβάλει στα τελικά αποτελέσματα. Τα system calls πρέπει να επιλέγονται προσεκτικά, ώστε να χρησιμοποιούνται αρκετά από τα διάφορα προγράμματα. Συνήθως τα κακόβουλα προγράμματα χρησιμοποιούν ένα συγκεκριμένο σύνολο system calls, τα οποία καλούνται με πολύ πιο μικρή συχνότητα, από όταν καλούνται από κάποια φυσιολογική εφαρμογή.

Τέλος μπορεί να σημειωθεί ότι ένα μεγάλο σύνολο system calls μπορεί να δημιουργήσει ένα πιο αποτελεσματικό μοντέλο ανίχνευσης επιθέσεων. Αυτό οφείλεται στο γεγονός ότι υπάρχει μια πιο λεπτομερής ανάλυση της συμπεριφοράς ενός προγράμματος και έτσι με βάση περισσότερα στοιχεία μπορεί να γίνει καλύτερος διαχωρισμός των κακόβουλων εφαρμογών από τις φυσιολογικές.

6.2 Μελλοντική Εργασία

Σε αυτό το σημείο θα αναφερθούν τα σημεία που θεωρείται ορθό στα πλαίσια μελλοντικών εργασιών ώστε να δημιουργηθούν πιο ολοκληρωμένα και με πολύ χαμηλά ποσοστά λάθους, συστήματα ανίχνευσης επιθέσεων.

Μπορεί να υλοποιηθεί και να εξεταστεί η συμπεριφορά ενός μοντέλου ανίχνευσης κακόβουλου κώδικα, χρησιμοποιώντας διαφορετικά system calls. Όπως φάνηκε και από την παρούσα εργασία ένα διαφορετικό σύνολο μπορεί να οδηγήσει σε διαφορετικά αποτελέσματα. Έτσι περαιτέρω μελέτη πάνω στο θέμα μπορεί να οδηγήσει στην εξαγωγή των πιο αντιπροσωπευτικών system calls, τα οποία θα δημιουργήσουν ένα ισχυρό εργαλείο ανίχνευσης επιθέσεων που θα έχει πάρα πολύ χαμηλό ποσοστό λάθους.

Ακόμα μπορεί να γίνει δημιουργία κάποιου μοντέλου ανίχνευσης εισβολών με περισσότερα δεδομένα, δηλαδή να χρησιμοποιηθούν σύνολα με περισσότερες εφαρμογές τόσο κακόβουλες, όσο και φυσιολογικές.

Τέλος θα μπορούσε να υλοποιηθεί ένα μοντέλο ανίχνευσης εισβολών, το οποίο να μπορεί να αξιολογηθεί σε δίκτυο υπολογιστών. Συγκεκριμένα να ελεγχθεί κατά πόσο μπορεί να γίνει ανίχνευση κάποιας επίθεσης σε ένα δίκτυο, όπου τα δεδομένα διακινούνται συνεχώς και περνούν διαμέσου διάφορων κόμβων.

Βιβλιογραφία

[1] K. Borders, and A. Prakash, "Web Tap: Detecting Covert Web Traffic," *Proceedings of the 11th ACM conference on Computer and communication security*, pp. 110-120, October 2004.

[2] M. T. Brannick. Logistic Regression [Online]. Available: http://luna.cas.usf.edu/~mbrannic/files/regression/Logistic.html Accessed: May 2010

[3] F. Cohen, "Computer Viruses Theory and Experiments", *Computers and Security*, vol. 6, pp. 22-35, 1985.

[4] D. E. Denning, "An Intrusion – Detection Model," *IEEE Transactions on Software Engineering*, volume 13 , pp. 222 – 232, February 1987.

[5] S. Forrest, S. A. Hofmeyr, and A. Somayaji, "Computer Immunology," *Communications of the ACM*, Vol. 40, No 10, pp. 88-96, October 1997.

[6] G. D. Garson, Logistic Regression [Online]. Available: http://faculty.chass.ncsu.edu/garson/PA765/logistic.htm#biblio Accessed: May 2010.

[7] A. K. Ghosh, J. Wanken, and F. Charron, "Detecting Anomalous and Unknown Intrusions Against Programs*." Proceedings of the 14th Annual Computer Security Application*, pp. 259-267. December 1998.

[8] S. A. Hofmeyr, S.Forrest and A.Somayaji, "Intrusion detection using sequences of system calls", *Journal of Computer Security*, Vol.6, pp. 151 – 180, August 1998.

[9] C. Ioannou, "The Hunt for Viral Processes," Master Thesis, Florida Institute of Technology, 2006.

[10] C. Kruegel and G. Vigna, "Anomaly Detection of Web-based Attacks," *Proceedings of the 10th ACM conference on Computer and Communications Security*, pp. 251 – 261, October 2003.

[11] J. F. Kurose and K. W. Ross, *Computer Networking: A top down approach*, 4th Edition, New Yok: Addison Wesley, 2008, pp. 705 – 775.

[12] S. Northcutt, L. Zeltser, S. Winters, K. Kent Frederick, and R. W, Ritchey, *Network Perimeter Security: The Definite Guide to firewalls, VPNS, Routers,*

and Intrusion Detection Systems, 3rd Edition, New Riders, 2003, pp. 161 – 180 & 645 – 650.

[13] S.Northcutt and J. Novak, *Network Intrusion Detection*, 3rd Edition, New Riders, 2002.

[14] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing*, 4th Edition, New Jersey: Pearson Education, 2007

[15] RFC security glossary [Online] http://www.ietf.org/rfc/rfc2828.txt, May 2010

[16] J. H. Saltzer and M. D. Schroeder, "The Protection of Information in Computer Systems", *Forth ACM Symposium on Operating System Principles*, October 1975.

[17] L. Smith, "A tutorial on Principal Components Analysis," Tutorial [Online]. University of Otago. Dunebin, New Zealand. Available: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf Accessed: May 2010.

[18] W. Stallings, *Operating Systems: Internals and Design Principles*, 5th Edition, New Jersey: Prentice Hall, 2005, pp. 677 – 710.

[19] W. Stallings, *Networks Security Essentials: Applications and Standards*, 3rd Edition, New Jersey: Prentice Hall, 2007, pp. 299 – 315.

[20] M.Swanson and B. Guttman, "Generally Accepted Principles and Practices for Securing Information Technology Systems", National Institute of Standards and Technology, Technology Administration, U.S. Department of Commerce, September 1996.

[21] C. Taylor and J. Alves-Foss. "An Empirical Analysis of NATE – Network Analysis of Anomalous Traffic Events," *New Security Paradigms Workshop 2002*, pp. 18 – 26, September 2002.

[22] G. Vigna, F. Valeur, and R. A. Kammerer, "Designing and Implementing a Family of Intrusion Detection Systems," *Proceedings of the 9th European software engineering conference held jointly with 11 ACM SIGSOFT international symposium on Foundations of Software Engineering*, pp. 88-97, September 2003.

[23] VirusList Encyclopedia, http://www.securelist.com/en/threats Accessed: May 2010.

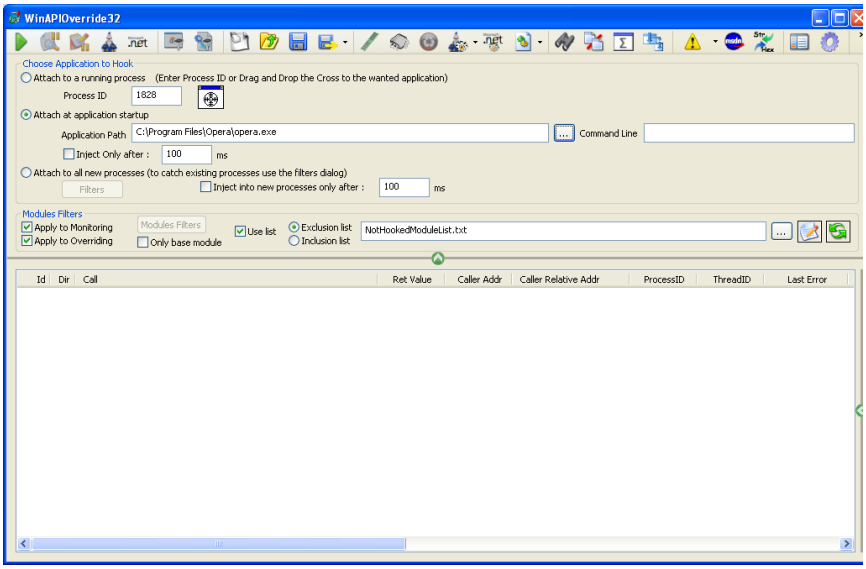
[24] WinAPIOverride32, http://jacquelin.potier.free.fr/winapioverride32/, Accessed: May 2010.

Παράρτημα Α

Παρουσίαση προγράμματος WinAPIOverride32

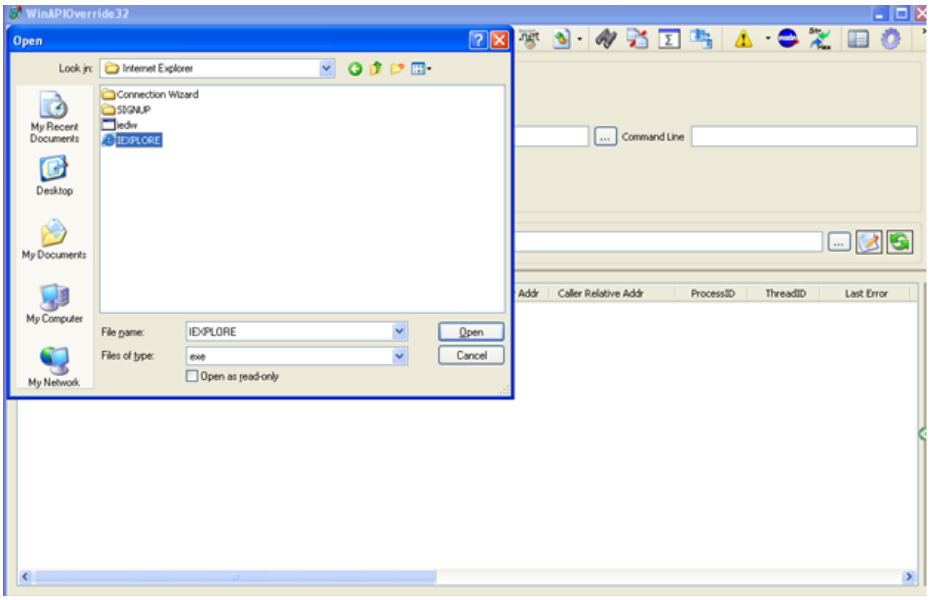
Το πρόγραμμα που χρησιμοποιήθηκε για την καταγραφή των system calls είναι το WinAPIOverride32 [24]. Πιο κάτω παρουσιάζεται ο τρόπος λειτουργίας του.

Αρχικά παρουσιάζεται η οθόνη που εμφανίζεται με την επιλογή του προγράμματος.



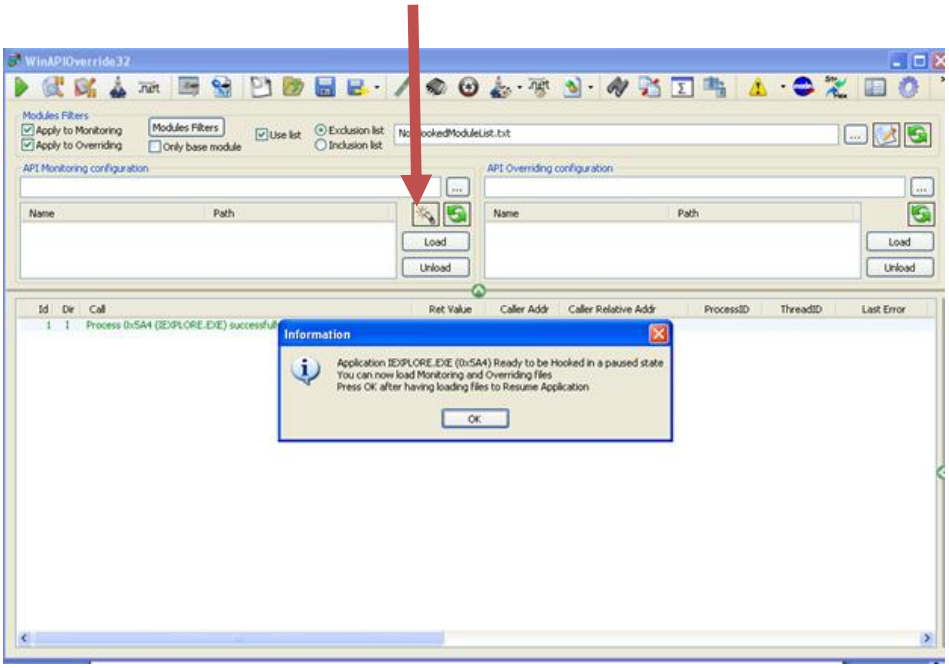
Σχήμα Α-1 Αρχική Οθόνη προγράμματος

Η διεργασία που θα εκτελεστεί καθορίζεται από την επιλογή Attach at application startup και συγκεκριμένα στο Application Path.



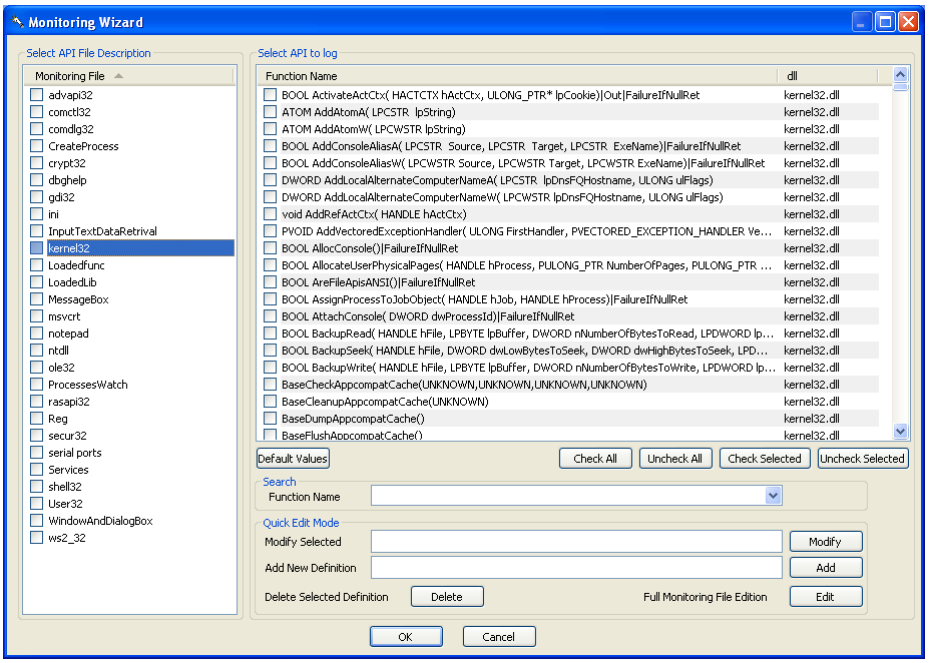
Σχήμα Α-2 Επιλογή Διεργασίας προς εκτέλεση

Μόλις επιλεχθεί η διεργασία που θα εκτελεστεί παρουσιάζεται η πιο κάτω οθόνη. Ο καθορισμός των system calls που θα καταγραφούν γίνεται επιλέγοντας την πιο κάτω επιλογή.



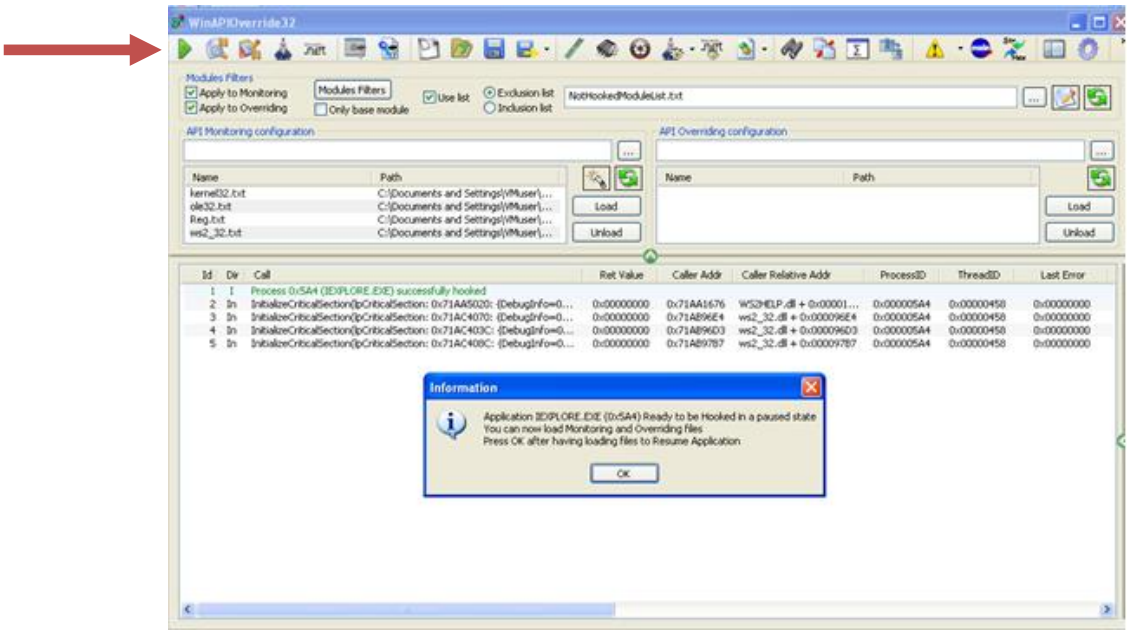
Σχήμα Α-3 Οθόνη μετά τον καθορισμό διεργασίας

Η πιο κάτω οθόνη παρουσιάζεται για την επιλογή των system calls. Τα system calls είναι ταξινομημένα ανάλογα με την βιβλιοθήκη στην οποία υπάγονται.



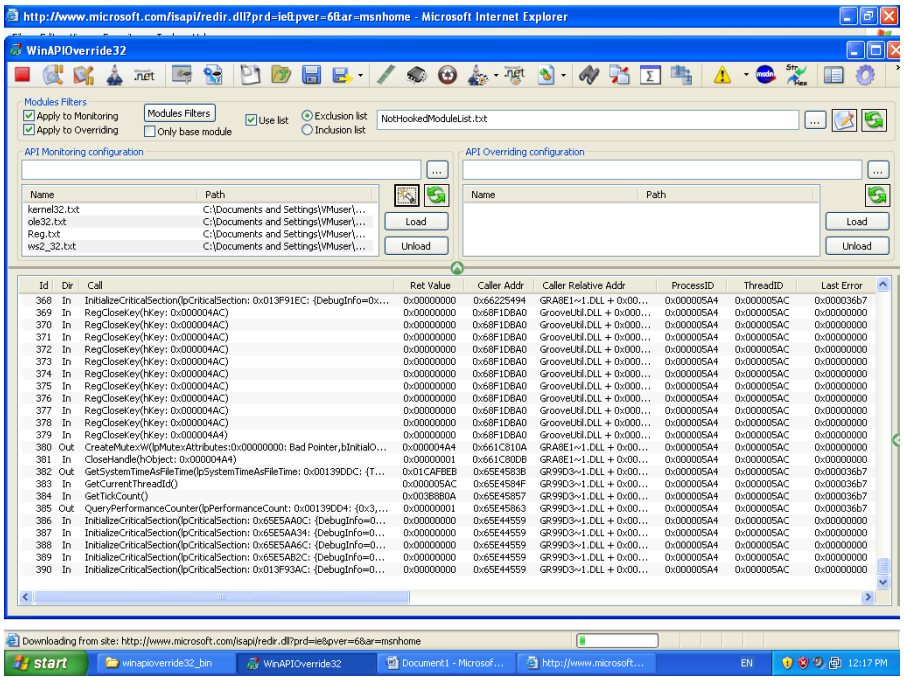
Σχήμα Α-4 Καθορισμός system calls

Μετά την επιλογή των system calls που θα καταγραφούν παρουσιάζεται η πιο κάτω οθόνη. Η εκτέλεση της διεργασίας μπορεί να ξεκινήσει πατώντας το κουμπί που φαίνεται πιο κάτω.

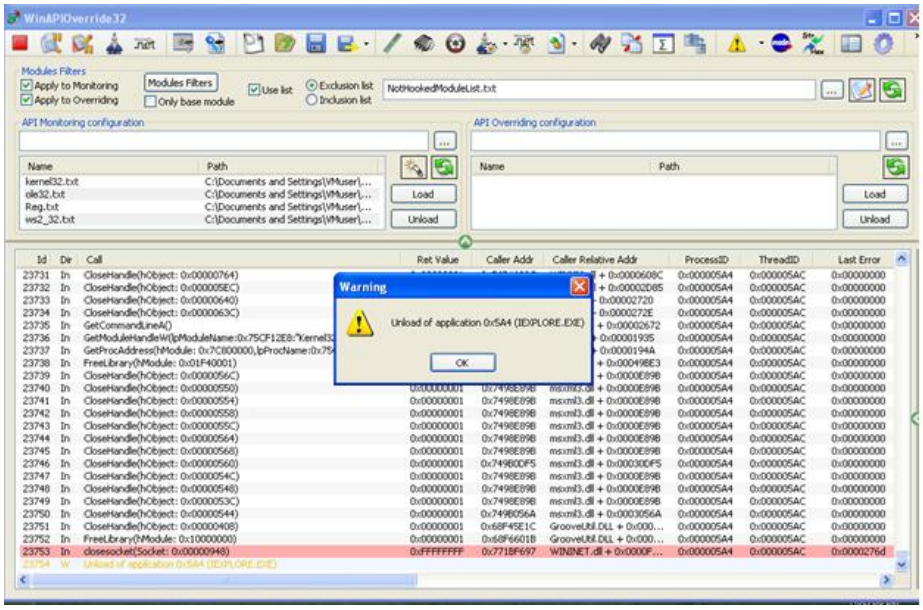


Σχήμα Α-5 Οθόνη πριν εκτέλεση της διεργασίας

Πιο κάτω παρουσιάζεται η οθόνη κατά την καταγραφή και εκτέλεση της διεργασίας και η οθόνη μετά το τέλος εκτέλεσης της διεργασίας.

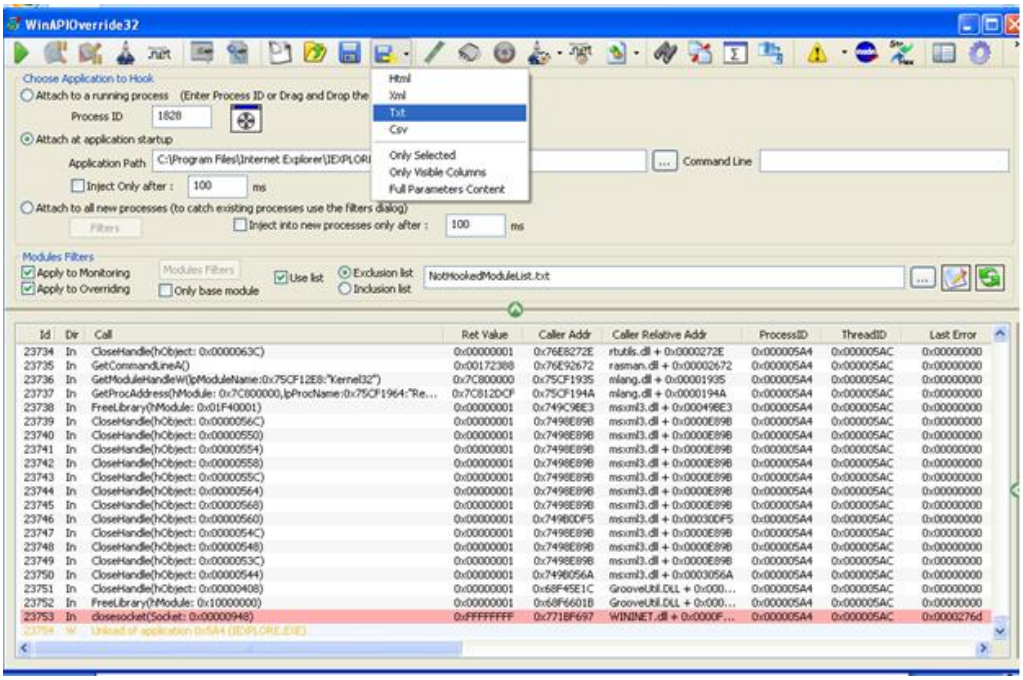


Σχήμα Α-6 Οθόνη κατά την εκτέλεση της διεργασίας

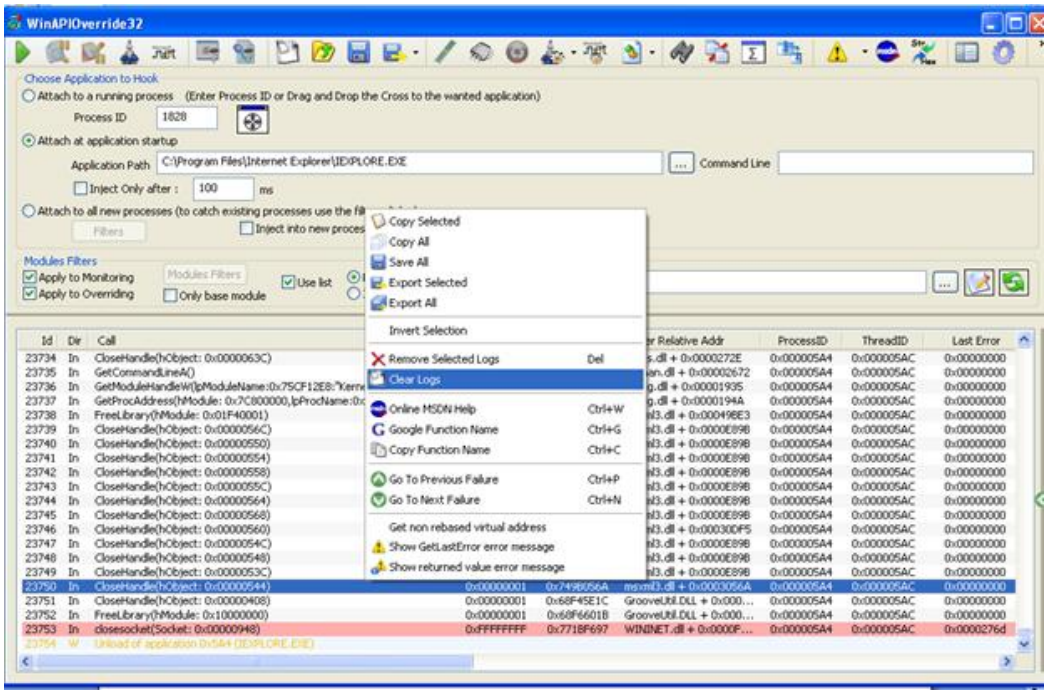


Σχήμα Α-7 Οθόνη μετά τον τερματισμό της διεργασίας

Η καταγραφή των system calls μπορεί να αποθηκευτεί σε log αρχεία με την διαδικασία που φαίνεται πιο κάτω όπως και η αρχικοποίηση του παράθυρου αρχικοποίησης.



Σχήμα Α-8 Οθόνη για δημιουργία log αρχείου



Σχήμα Α-9 Αρχικοποίηση παράθυρου

Παράρτημα Β

Πρόγραμμα σε JAVA για την ανάλυση των system calls

Ο κώδικας χρησιμοποιεί το πακέτο Apache POI για διάβασμα από MS Excel αρχεία.

Κλάση ReadFromFile: Κύρια κλάση

```
/* PROGRAM FOR MS EXCEL ANALYSIS*/


import java.io.*;
import java.util.Vector;

import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.poifs.filesystem.POIFSFileSystem;

//Program that reads data from a MS Excel file and creates an analysis
public class ReadFromExcel {

      //vector of object Node that consists the list of the selected
system calls
      public static Vector<Node> listCalls=new Vector<Node>();

      //Main function
      public static void main(String[] args) {

            HSSFWorkbook wb2 = new HSSFWorkbook();  //creation of
object HSSFWorkbook for the output workbook

            try {
                  //opens MS Excel file and obtains access to its
contents. the name of the file is given as an argument
                  FileInputStream fstream = new
FileInputStream(args[0]);
                  POIFSFileSystem fs = new POIFSFileSystem(fstream);
                HSSFWorkbook wb = new HSSFWorkbook(fs); //creation of
object HSSFWorkbook for the input workbook
                HSSFSheet sheet; //object of class HSSFSheet for the
sheets of the Excel file
                HSSFRow row;     //object of class HSSFRow for the
rows of each sheet
                HSSFCell cell;   //object of class HSSFCell for each
cell

                int rows;           //Number of rows
                int sheets;         //Number of sheets
                String str;   //temporal string
                String sheetName;  //string with the sheet name

                sheets=wb.getNumberOfSheets();  //gets the number of
sheets in the workbook
```

```
                sheet=wb.getSheetAt(0);                //the first
sheet is assigned to object sheet

            rows = sheet.getPhysicalNumberOfRows(); //gets the number
of rows in the sheet

            //this loop gets the names of the system calls in the
first sheet and adds them to vector listCalls
             for(int j = 0; j < rows; j++) {
                   row = sheet.getRow(j);  //assigns a row of the
sheet to object row
                        if(row != null)
                        {
                          cell = row.getCell(0);  //assigns the first
cell of each row to object cell
                          //if the cell is not null, it converts it to
string and adds it to listCalls
                          if(cell != null) {
                               str=cell.toString();
                               listCalls.add(new Node(str));
                          }
                        }
             }

             HSSFSheet sheet2 = wb2.createSheet(); //creation a sheet
in the output workbook
             HSSFRow[] row2=new HSSFRow[listCalls.size()+1]; //creates
the number of rows of the system calls
             row2[0]= sheet2.createRow(0);          //creates the
first row with a null value
             row2[0].createCell(0).setCellValue("");
             //for each system call it creates a row in the output
workbook and puts the name of the system call
             //in the first cell of each row
             for(int k = 1; k <=listCalls.size(); k++)
             {
                   row2[k] = sheet2.createRow(k);

row2[k].createCell(0).setCellValue(listCalls.elementAt(k-1).name);
             }

             //for each sheet in the input workbook it calculates the
occurrence of each system call in a process
                for(int i=1;i<sheets;i++)
                {
                  sheet=wb.getSheetAt(i);         //object sheet is
assigned each sheet of the input workbook
                  sheetName=wb.getSheetName(i);   //gets the name of
the sheet
                  rows = sheet.getPhysicalNumberOfRows(); //gets the
number of rows of each sheet

                  //this loop calculates the occurrence of each system
call in a process
                  for(int j = 0; j < rows; j++) {
                        row = sheet.getRow(j);    //assigns a row of
the sheet to object row
                        if(row != null)
                        {
```

```
                                cell = row.getCell(1);  //assigns the
second cell of each row to object cell
                                //if cell is not null then function
adjustList is called to adjust the sum
                                if(cell != null) {
                                      str=cell.toString();
                                      adjustList(str);    //calls
function adjustList
                                }
                          }
                    }
                   //the first row of the output worksheet contains the
name of the process
                    row2[0].createCell(i).setCellValue(sheetName);
                    //writes the occurrence of each system call in the
appropriate cell  of a specific column
                    for(int m = 1; m <=listCalls.size(); m++)

      row2[m].createCell(i).setCellValue(listCalls.elementAt(m-
1).num);
                   //initialization of vector listCalls
                    for(int r = 0; r <listCalls.size(); r++)
                    {
                          listCalls.elementAt(r).num=0;
                    }
                }
            } catch (Exception e){//Catch exception if any
                  System.err.println("Error: " + e.getMessage());
            }



            //write the object wb2 in an excel file
            try {
            //creates output file
             FileOutputStream fileOut = new
FileOutputStream("workbook.xls");
             wb2.write(fileOut);  //writes output workbook to the
excel file
             fileOut.close();  //closes the file
            } catch (Exception e){//Catch exception if any
                  System.err.println("Error: " + e.getMessage());
            }
      }

//function that adjusts the vector listCalls for each occurrence of a
system call
public static void adjustList(String temp)
{

      int equality;
      //checks if the temp string is equal with the name of a system
call
      for (int i=0; i<listCalls.size();i++)
      {
            equality=listCalls.elementAt(i).getEqual(temp);
listCalls.elementAt(i).setNum(); //calls function getEqual of object
Node
            //if they are equal then the number of the occurrence for
the specific system call is increased
```

```
            if (equality==0)
            {
                  listCalls.elementAt(i).setNum(); //calls function
setNum of object Node
                  return;
            }
      }

}

}
```

Κλάση Node : Βοηθητική

```
//Class node
public class Node {
      //attributes of class Node
      public String name;  //name of system call
      public int num;      //number of occurrence

      //Constructor
      public Node(String str)
      {
            name=str;
            num=0;
      }

      //function that checks in the name is equal to the parameter
      public int getEqual(String s1)
      {
            int check;
            check=s1.compareTo(name);
            return check;
      }

      //function that increases the number
      public void setNum()
      {
            num=num+1;
      }

}
```

Παράρτημα Γ

Πρόγραμμα σε MATLAB για το Principal Component Analysis

Ενδεικτικός κώδικας

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Program for Principal Component Analysis %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%array of processes (different system calls by rows)
initialArray=[
      ...
 ];

%transposes array so that each column is a different system call
X=initialArray';

%calculates the mean of each column - system call
meanArray=mean(X);

%finds the dimensions of array X
[m n]=size(X);

%creates matrix L (normalized data)
L=zeros(m,n);  %initialization of array L
 for j=1:n
     L(:,j)=X(:,j)-meanArray(j);
 end

 LT=L';    %transposes matrix L

%finds the covariance of the elements in the array
covariance=cov(X);

%calculates the eigenvector array and the eigenvalues
[eigenvector,D]=eig(covariance);

%converts diagonal matrix of eigenvalues to a row
eigenvalues=(diag(D))';

%sorts the array of eigenvectors according to eigenvalues
 [junk, rindices] = sort(-1*eigenvalues);
 eigenvalues = eigenvalues(rindices);
 temp = eigenvector(:,rindices);

 %declaration of array A
 A=temp';

%checks if array A is calculated correctly based on I=A*A(T)
res=A';
Id=A*res;
%disp(Id)
```

```
%calculates array Y
Y=A*LT;
%calculates Y(T) array and displays result
YT=Y';
disp(YT);
```