ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

ΚΑΤΕΥΘΥΝΣΗ: ΔΙΚΤΥΟΚΕΝΤΡΙΚΑ ΣΥΣΤΗΜΑΤΑ

Ανάπτυξη συστήματος μετα επεξεργασίας συνεγερμών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Χαρούλη Αναστάσιου

Επιβλέπων: Κωνσταντίνος Λαμπρινουδάκης
Επίκουρος Καθηγήτής

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

Ανάπτυξη συστήματος μετα επεξεργασίας συνεγερμών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Χαρούλη Αναστάσιου

Επιβλέπων: Κωνσταντίνος Λαμπρινουδάκης
Επίκουρος Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 28η Φεβρουαρίου 2012.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Κ. Λαμπρινουδάκης	Σ. Κάτσικας	Χ. Ξενάκης
Επίκουρος Καθηγητής	**Καθηγητής**	**Επίκουρος Καθηγητής**

Περίληψη

Η χρήση των συστημάτων ανίχνευσης εισβολών (Intrusion Detection Systems, IDS) είναι ευρέως διαδεδομένη για την προστασία ενός δικτύου. Παρά την επιτυχία αυτών των τεχνολογιών, κοινό πρόβλημα σχεδόν σε όλες τις κατηγορίες IDSs αποτελεί ο τεράστιος αριθμός των συνεγερμών που παράγουν καθώς και το υψηλό ποσοστό των ψευδών συνεγερμών. Γενικά τα IDSs παράγουν πάρα πολλές ειδοποιήσεις σε σχέση με το μέγεθος του συστήματος που προστατεύουν. Ένα IDS που χρησιμοποιείται για την προστασία ενός μεσαίου μεγέθους δικτύου παράγει χιλιάδες συνεγερμούς κάθε ημέρα, οι οποίοι απαιτούν υπερβολική προσπάθεια από ένα διαχειριστή δικτύου, προκειμένου να αναλυθούν καθώς και να ελεγχθούν. Εκτός αυτού, πολλές από αυτές τις ειδοποιήσεις είναι συνήθως ψευδείς συνεγερμοί, πρόκειται για ειδοποιήσεις που δεν έχουν προκληθεί από πραγματικές εισβολές, αλλά μπορεί να είναι το αποτέλεσμα μίας φυσιολογικής λειτουργίας του συστήματος ή το αποτέλεσμα ενός μη σωστού ρυθμισμένου IDS. Στην παρούσα εργασία επιχειρείται να αναπτυχθεί κατάλληλο λογισμικό του οποίου ο στόχος είναι η μετα επεξεργασία των συνεγερμών που παράγονται από ένα σύστημα ανίχνευσης εισβολών έτσι ώστε ο αναλυτής ασφαλείας να έχει ένα πιο ποιοτικό σύνολο συνεγερμών στη διάθεσή του να επεξεργασθεί.

Λέξεις κλειδιά: Σύστημα ανίχνευσης εισβολών, συνεγερμός, μετα-συνεγερμός

Abstract

Intrusion detection Systems (IDS) are commonly used in order to increase the level of security in a computer network. However successful these technologies may be, a common problem of almost all categories of IDSs is the huge number of alerts they produce and the high percentage of false ones. Generally IDSs produce too many alerts compared to the size of the system they protect; an IDS protecting an average-sized network produces thousands of alerts per day, which may require excessive effort from a single network administrator, in order to analyze and check them all. Besides, many of these alerts are usually false ones; they are alerts which have not been triggered by real intrusions, but can either be the outcome of normal system operations which trigger one of the IDS signatures, or the outcome of a misconfigured IDS. This thesis attempts to develop suitable software whose objective is the treatment of alerts generated by an intrusion detection system so that the security analyst to have a more quality set of alerts available to elaborate.

Keywords: Intrusion Detection Systems, IDS, alert, meta-alert, alert set.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Καθηγητή κ. Κωνσταντίνο Λαμπρινουδάκη για την ανάθεση της εργασίας, την παρακολούθηση και παροχή κατευθύνσεων για την υλοποίηση της. Επίσης ευχαριστώ τον συνεπιβλέποντα κ. Γιώργο Σπαθούλα για τη συνεργασία και στενή παρακολούθηση των φάσεων της εργασίας και τις εποικοδομητικές παρατηρήσεις του.

Με αφορμή, τέλος την ολοκλήρωση των σπουδών μου, θα ήθελα να ευχαριστήσω την οικογένειά μου για την υποστήριξη και προσφορά από τα πρώτα μου βήματα στη μάθηση.

Κατάλογος περιεχομένων

1 Εισαγωγή στο κεφάλαιο

1.1 Εισαγωγή

Στον τομέα της δικτύωσης, ο χώρος της ασφάλειας δικτύων, αποτελείται από τις διατάξεις και πολιτικές έχουν έγκριθεί από το διαχειριστή του δικτύου για την πρόληψη και την παρακολούθηση της μη εξουσιοδοτημένης πρόσβασης, κακής χρήσης, τροποποίησης, ή στέρηση του δικτύου ηλεκτρονικών υπολογιστών και τους διαθέσιμους πόρους του δικτύου. Η ασφάλεια των δικτύων προϋποθέτει την έγκριση της πρόσβασης σε δεδομένα σε ένα δίκτυο, η οποία ελέγχεται από το διαχειριστή του δικτύου. Οι χρήστες επιλέγουν ή τους αποδίδεται ένα αναγνωριστικό και ένας κωδικός πρόσβασης ή άλλες πληροφορίες ελέγχου ταυτότητας που τους επιτρέπει την πρόσβαση σε πληροφορίες και προγραμμάτων στο πλαίσιο της εξουσίας τους. Η ασφάλεια των δικτύων καλύπτει μια ποικιλία δικτύων υπολογιστών, τόσο των δημόσιων όσο και ιδιωτικών, που χρησιμοποιούνται στις καθημερινές εργασίες για τη διεξαγωγή των συναλλαγών και της επικοινωνίας μεταξύ επιχειρήσεων, κυβερνητικών υπηρεσιών και ιδιωτών. Τα δίκτυα μπορεί να είναι ιδιωτικά, όπως στο εσωτερικό μιας εταιρίας, και άλλα που μπορεί να είναι ανοικτά για την πρόσβαση του κοινού. Η ασφάλεια των δικτύων εμπλέκεται με οργανισμούς, επιχειρήσεις και άλλους τύπους θεσμικών οργάνων. Όπως και ο τίτλος του εξηγεί: προστατεύει το δίκτυο, προστατεύοντας και εποπτεύοντας τις διάφορες ενέργειες που επιτελούνται. Ο πιο κοινός και απλός τρόπος για την προστασίας ενός πόρου του δικτύου αποτελεί η ανάθεση ενός μοναδικού ονόματος και αντίστοιχου κωδικού.

Ένα από τα εργαλεία που χρησιμοποιείται στον τομέα της ασφάλειας των δικτύων αποτελούν τα συστήματα ανίχνευσης εισβολών (Intrusion Detection System IDS). Τα συστήματα ανίχνευσης εισβολών (IDSs) είναι λογισμικό ή / και hardware που εντοπίζει κακόβουλη συμπεριφορά στα συστήματα που συνδέεται για να προστατεύει και παράγει σχετικές ειδοποιήσεις. Τα IDSs μπορεί να χρησιμοποιούνται για την προστασία ενός μόνο κεντρικού υπολογιστή ή ενός μεγάλου δίκτυου υπολογιστών. Ταξινομούνται ανάλογα με τα δεδομένα που χρησιμοποιούν για την ανίχνευση εισβολών. Τα Network based IDSs βασίζονται στην κυκλοφορία και τη χρήση του δικτύου, τα Host based όπου χρησιμοποιούν αρχεία καταγραφής γεγονότων του συστήματος και κλήσεις συστήματος στο μηχάνημα που φιλοξενούνται και τέλος τα Protocol based IDSs που βασίζονται σε πληροφορίες σχετικά με τη λειτουργία των ειδικών πρωτοκόλλων. Τα IDSs ταξινομούνται επίσης σύμφωνα με τη λογική που χρησιμοποιούν για την ανίχνευση εισβολών από τα δεδομένα αυτά. Τα Signature based IDSs βασίζονται στην αξιοποίηση συγκεκριμένων μοντέλων επιθέσεων και προσπαθούν να αντιστοιχίσουν αυτά τα μοντέλα με τα δεδομένα. Εάν μία αντιστοίχιση διαπιστωθεί, τότε μια προειδοποίηση που υποδεικνύει την ύπαρξη της αντίστοιχης επίθεσης

παράγεται. Τα Anomaly based όπου χρησιμοποιείται ένα μοντέλο της κανονικής λειτουργίας του συστήματος και προσπαθείται να εντοπιστούν σημαντικές αποκλίσεις από αυτό το μοντέλο στα παραγόμενα δεδομένα κατά τη διάρκεια της κανονικής δραστηριότητας του συστήματος. Αν μια τέτοια απόκλιση εντοπίζεται, τότε μια ειδοποίηση παράγεται, μαζί με πληροφορίες για τη φύση της απόκλισης. Η ερευνητική δραστηριότητα στον χώρο των IDS συνεχώς εξελίσσεται προκειμένου να δημιουργηθούν ισχυρές και αποτελεσματικές τεχνολογίες έτσι ώστε να είναι δυνατόν να ταξινομηθεί κάθε δραστηριότητα σε ένα σύστημα με ένα αποδεκτό ποσοστό. Παρά την επιτυχία αυτών των τεχνολογιών, κοινό πρόβλημα σχεδόν σε όλες τις κατηγορίες IDSs αποτελεί ο τεράστιος αριθμός των συνεγερμών που παράγουν καθώς και το υψηλό ποσοστό των ψευδών συνεγερμών. Γενικά τα IDSs παράγουν πάρα πολλές ειδοποιήσεις σε σχέση με το μέγεθος του συστήματος που προστατεύουν. Ένα IDS που χρησιμοποιείται για την προστασία ενός μεσαίου μεγέθους δικτύου παράγει χιλιάδες συνεγερμούς κάθε ημέρα, οι οποίοι απαιτούν υπερβολική προσπάθεια από ένα διαχειριστή δικτύου, προκειμένου να αναλυθούν καθώς και να ελεγχθούν. Εκτός αυτού, πολλές από αυτές τις ειδοποιήσεις είναι συνήθως ψευδείς συνεγερμοί, πρόκειται για ειδοποιήσεις που δεν έχουν προκληθεί από πραγματικές εισβολές, αλλά μπορεί να είναι το αποτέλεσμα μίας φυσιολογικής λειτουργίας του συστήματος ή το αποτέλεσμα ενός μη σωστού ρυθμισμένου IDS. Το πρόβλημα της ταξινόμησης σε ένα περιβάλλον που επηρεάζεται από πολλούς παράγοντες είναι πολύπλοκο, συνεπώς οι ψευδείς συναγερμοί είναι αναπόφευκτοι σε οποιαδήποτε IDS. Εκτός αυτού, οι μεθοδολογίες εισβολής καθώς και οι στρατηγικές επίθεσης εξελίσσονται παράλληλα με τα συστήματα τεχνολογίας υπολογιστών, για αυτό το λόγο και ένα IDS που είχε ικανοποιητικές επιδόσεις πριν από πέντε χρόνια μπορεί να είναι ακατάλληλο σήμερα. Δεν είναι περίεργο, επομένως, ότι η έρευνα στην ανίχνευση εισβολών έχει επικεντρωθεί τον τελευταίο καιρό στη μετά επεξεργασία των συνεγερμών, έτσι ώστε να παράγεται ένα πιο ποιοτικό σύνολο συναγερμών, το οποίο θα είναι πολύ πιο χρήσιμο για τον αναλυτή. Πολλές μέθοδοι έχουν προταθεί για την μείωση των ψευδών θετικών ενδείξεων. Ορισμένες εξ αυτών προτείνουν διαφορετικές ρυθμίσεις των IDSs, ενώ περισσότερες από αυτές προτείνουν την μετά επεξεργασία των συνεγερμών.

1.2 Αντικείμενο της διπλωματικής

Στα πλαίσια της παρούσας διπλωματικής εργασίας θα αναπτυχθεί κατάλληλο λογισμικό του οποίου ο στόχος είναι η μετα επεξεργασία των συνεγερμών που παράγονται από ένα Intrusion Detection System έτσι ώστε ο αναλυτής ασφαλείας να έχει ένα πιο ποιοτικό σύνολο συνεγερμών στη διάθεσή του να επεξεργασθεί. Ένας τρόπος που προτείνεται στη βιβλιογραφία για τη μετα επεξεργασία των συνεγερμών που παράγονται από ένα IDS [14] αποτελεί η εφαρμογή μίας σειράς από στοιχείων (components) στο σύνολο των παραχθέντων συνεγερμών τα οποία με βάση ένα συγκεκριμένο αλγόριθμο αποκλείουν κάποιους από τους συνεγερμούς. Επειδή όμως κανένα από τα γνωστά components δεν είναι δυνατόν να μειώσουν το μέγεθος του συνόλου των συνεγερμών επαρκώς απαιτείται συνδυασμός αυτών έτσι ώστε να παράγονται σύνθετα φίλτρα. Θα κατασκευαστεί λοιπόν ένα λογισμικό όπου θα είναι δυνατόν να δημιουργηθούν διάφορα components τα οποία υλοποιούν συγκεκριμένη λογική και κατόπιν τα συγκεκριμένα components θα συνδυάζονται έτσι ώστε να δημιουργηθεί ένα σύνθετο φίλτρο. Το παραγόμενο φίλτρο θα είναι δυνατόν να αποθηκευτεί σε μορφή xml με σκοπό την εφαρμογή του σε διαφορετικό σύνολο συνεγερμών και τη σύγκρισή του με άλλα φίλτρα που έχουν παραχθεί. Για την υλοποίηση της εφαρμογής θα χρησιμοποιηθεί η γλώσσα προγραμματισμού Java. Η διαδικασία ανάπτυξης της εφαρμογής θα περιλαμβάνει τα ακόλουθα βήματα:

- Εξοικίωση με το πρόβλημα και τις σχετικές τεχνολογίες.
- Βασικός σχεδιασμός. Το τμήμα αυτό περιλαμβάνει το βασικό σχεδιασμό της αρχιτεκτονικής του συστήματος όπως για παράδειγμα το σχεδιασμό των xml αρχείων στα οποία θα αποθηκεύεται το παραγόμενο φίλτρο.
- Λεπτομερής σχεδιασμός.
- Συγγραφή κώδικα και ανάπτυξη της εφαρμογής.

1.3 Οργάνωση τόμου

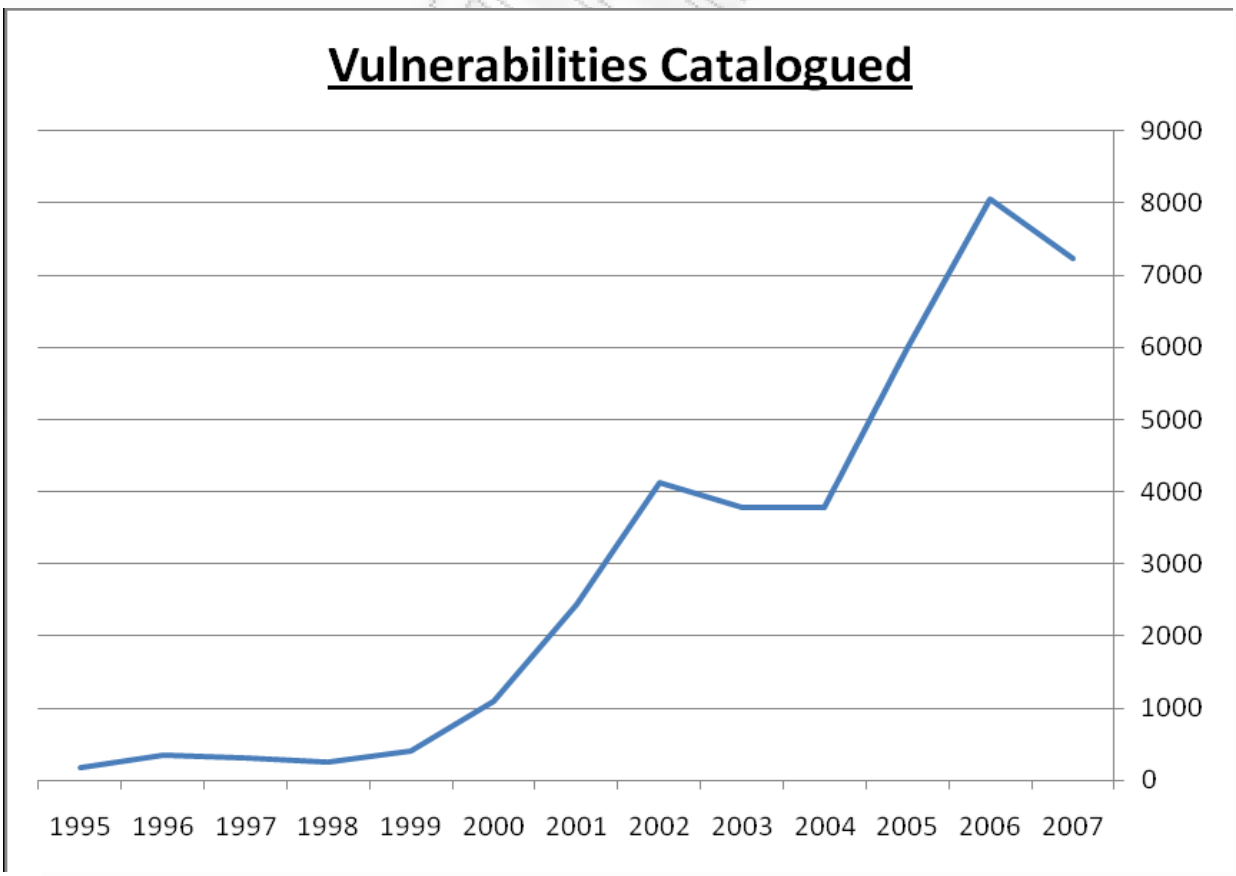
Στο κεφάλαιο που ακολουθεί παρουσιάζεται το απαραίτητο θεωρητικό υπόβαθρο για τη μελέτη και την ανάπτυξη του συστήματος. Επιχειρείται να δωθεί μία εισαγωγή στην ασφάλεια δικτύων και πιο συγκεκριμένα στα συστήματα ανίχνευσης εισβολών. Κατόπιν παρουσιάζονται διάφορες μεθόδοι προερχόμενες από την επιστημονική κοινότητα που επιχειρούν να λύσουν το πρόβλημα των εσφαλμένων συνεγερμών που μπορεί να παράξει ένα σύστημα ανίχνευσης εισβολών. Στο κεφάλαιο 3 παρουσιάζεται αναλυτικά η σχεδίαση και η υλοποίηση του συστήματος. Περιγράφονται οι απαιτήσεις του συστήματος η αρχιτεκτονική καθώς και οι επιλογές που πραγματοποιήθηκαν κατά την υλοποίησή του.

2 θεωρητικό Υπόβαθρο

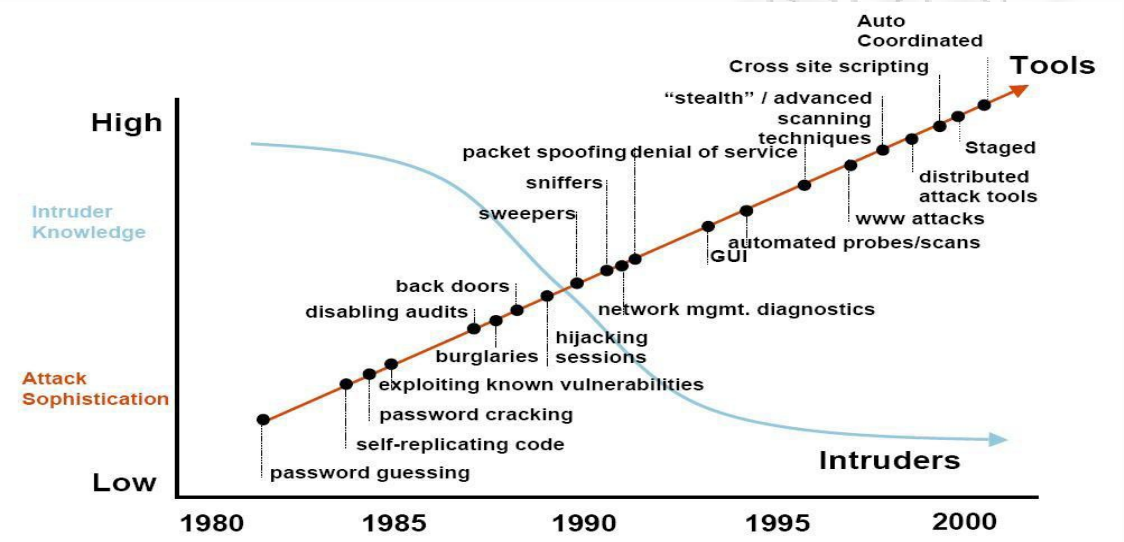
Στο παρόν κεφάλαιο παρέχεται το απαραίτητο θεωρητικό υπόβαθρο για τη μελέτη του αντικειμένου της διπλωματικής. Η παράγραφος 2.1 αναφέρεται γενικά στην ασφάλεια δικτύων. Στην παράγραφο 2.2 επιχειρείται μία εισαγωγή στα συστήματα ανίχνευσης εισβολών ενώ στην παράγραφο 2.3 παρουσιάζονται διάφορες αρχιτεκτονικές συστημάτων ανίχνευσης εισβολών. Στις παραγράφους 2.4 έως και 2.9 περιγράφονται μεθόδοι που έχουν αναπτυχθεί από την επιστημονική κοινότητα για τη μείωση των εσφαλμένων συνεγερμών στα συστήματα ανίχνευσης εισβολών.

2.1 Ασφάλεια Δικτύων

Οι τρεις άξονες της ασφάλειας των δεδομένων είναι η εμπιστευτικότητα, η ακεραιότητα και η διαθεσιμότητα. Η διατήρησή τους σε ένα δίκτυο υπολογιστών αποτελεί πεδίο της ασφάλειας των δικτύων. Η διασφάλιση της ασφάλειας του δικτύου απαιτεί μια πολυεπίπεδη προσέγγιση, επιτήρηση και τακτική ενημέρωση για την προστασία από τις τελευταίες απειλές. Τα τρωτά σημεία έχουν εμφανιστεί σε αυξανόμενους αριθμούς, όπως φαίνεται και από το σχήμα που ακολουθεί τα στοιχεία του οποίου προέρχονται από το CERT (Computer Emergency Response Team).



Η εκτεταμένη παρουσία των τρωτών σημείων και η διαθεσιμότητα στους χάκερ των εργαλείων, όπως nmap [11] και μηχανισμών όπως είναι το Metasploit [12] και το w3af, αυτοματοποιούν την διαδικασία hacking, αφού συνδυάζοντάς τα κάνουν τις επιθέσεις σχετικά απλές να πραγματοποιηθούν. Phishing toolkits, κατανεμημένη άρνηση υπηρεσίας πακέτα (DDoS), και οδηγοί για τη δημιουργία ιων trojans και άλλων βρίσκονται διαθέσιμα στο διαδίκτυο. Στο σχήμα που ακολουθεί φαίνεται η αυξανόμενη πολυπλοκότητα της επίθεσης έναντι του ρυθμού μείωσης στις δεξιότητες που απαιτούνται.



2.2 Εισαγωγή στα Intrusion Detection Systems

Ένα σύστημα ανίχνευσης εισβολών μπορεί να εγκατασταθεί σε συνδυασμό με ένα τείχος προστασίας για να παρέχει ένα συμπληρωματικό στρώμα στην ασφάλεια των δικτύων. Ένα σύστημα ανίχνευσης εισβολών υλοποιεί γνωστοποίηση των επιθέσεων που τυχόν πραγματοποιούνται στα συστήματα πληροφορικής με την πρόθεση της παραβίασης του απορρήτου, της ακεραιότητας ή της διαθεσιμότητας του συνόλου ή μέρους του συστήματος. Ένα σύστημα ανίχνευσης εισβολών (IDS) θα υποβάλει αναφορά σχετικά με τις επιθέσεις είτε επιτυχημένες είτε όχι, προκειμένου να παρέχει στο διαχειριστή του δικτύου, εικόνα σχετικά με τους κινδύνους που διατρέχει το δίκτυό του, προσθέτοντας άλλο ένα σημαντικό επίπεδο στην ασφάλεια του δικτύου. Το πρώτο μοντέλο ανίχνευσης εισβολών δημοσιεύτηκε το 1986 από τον Dorothy E. Denning [5] και τα συστήματα ανίχνευσης εισβολών έχουν γίνει, τα τελευταία χρόνια [6], μια δημοφιλή επιλογή για την προστασία σε ένα δίκτυο επιπλέον του τείχους προστασίας. Ενώ, όπως αναφέρθηκε προηγουμένως, τα τείχη προστασίας χρησιμοποιούνται για να αποκλεισθούν εισβολές περιμετρικά του δικτύου με βάση ορισμένα χαρακτηριστικά της κυκλοφορίας δεν είναι δυνατόν να εμποδιστούν οι

επιθέσεις που στοχεύουν σε νόμιμες υπηρεσίες. Για παράδειγμα, οι εταιρείες συχνά διαμορφώνουν το τείχος προστασίας τους για να επιτρέπει την πρόσβαση στη θύρα 80 (HTTP) σε ένα web server που βρίσκεται μέσα στο δίκτυό τους από το διαδίκτο. Εάν ένας εισβολέας επιτεθεί σε αυτό τον web server και αποκτήσει πρόσβαση στο δίκτυο, ένα IDS μπορεί να ανιχνεύσει την επίθεση χρησιμοποιώντας στοιχεία που προκύπτουν από την εσωτερική κίνηση κατά την επίθεση (μετά την εισβολή, οι εισβολείς ή το κακόβουλο λογισμικό μπορεί να προσπαθήσει να εξαπλωθεί σε άλλους υπολογιστές στο εσωτερικό δίκτυο). Τα συστήματα ανίχνευσης εισβολών παρέχουν μια περισσότερο λεπτομερή εικόνα της κυκλοφορίας στο δίκτυο από ότι τα τείχη προστασίας. Εφαρμόζουν πιο εξελιγμένα σύνολα κανόνων και κάνουν αναλυτική επιθεώρηση του κάθε πακέτου που, εκτός από την εξέταση των TCP, UDP και IP παραμέτρων της επικεφαλίδας, μπορεί να επιθεωρήσουν το ωφέλιμο φορτίο του πακέτου για την απόδειξη των επιθέσεων και να λαμβάνουν αποφάσεις με βάση το περιεχόμενο της. Οι αναφορές ασφαλείας που δημιουργούνται σε κάθε σήμα συναγερμού διανέμονται στους διαχειριστές του δικτύου οι οποίοι πρέπει να ανταποκριθούν αναλόγως, ώστε το σύστημα να επωφεληθεί από αυτές τις ενδείξεις.

Τα κύρια κίνητρα για την υιοθέτηση ενός IDS έχουν ως εξής:

- Ανίχνευση επιθέσεων – Πρωταρχική λειτουργεία.
- Εκτέλεση των πολιτικών ασφαλείας - για να εξασφαλιστεί ότι ένα δίκτυο να χρησιμοποιείται μόνο για τον επιδιωκόμενο σκοπό.
- Ο διαχειριστής δικτύου έχει πληροφορίες για το πώς μια επίθεση πραγματοποιήθηκε και τις μεθόδους που χρησιμοποιούνται για να υπονομευθεί ένα σύστημα.
- Επιπλέον πληροφορίες για την ασφάλεια - Χρήσιμες για την ανακάλυψη του πόσο επαρκώς οι λοιποί μηχανισμοί ασφαλείας λειτουργούν.

2.3 Έννοιες και Αρχιτεκτονικές ανίχνευσης εισβολών

Αυτή η ενότητα παρουσιάζει αρχιτεκτονικές συστημάτων ανίχνευσης εισβολών (network, host, hybrid) καθώς και πλεονεκτήματα και μειονεκτήματα της κάθε μίας. Επίσης, περιγράφονται οι κοινές μεθόδοι ανίχνευσης: στηριγμένο σε κανόνες και ανάλυση με βάση την ανίχνευση ανωμαλίας.

Network IDS

Τα συστήματα ανίχνευσης εισβολών μπορούν να ταξινομηθούν ανάλογα με την τοποθεσία πού το ίδιο το σύστημα είναι εγκατεστημένο. Ένα Network IDS (NIDS) παρακολουθεί την κυκλοφορία στο δίκτυο, εντοπίζοντας ενδείξεις επιθέσεων οι οποίες στη συνέχεια αναφέρονται σε έναν διαχειριστή. Από τα πιο δημοφιλή NIDS είναι το Snort [7]. Ένα

Network IDS έχει το πλεονέκτημα της παρακολούθησης μιας ομάδας μηχανημάτων από μία φυσική τοποθεσία στο δίκτυο, πράγμα που σημαίνει ότι απαιτείται ελάχιστη προσπάθεια συντήρησης αφού είναι απαραίτητη μόνο μία ενημερωμένη έκδοση για να καλύψει το σύνολο του δικτύου με έναν νέο κανόνα ή επιλογή διαμόρφωσης. Ένα Network IDS είναι συνήθως αόρατο στον εισβολέα που είτε δεν συνειδητοποιεί ότι αυτό το στρώμα της προστασίας υπάρχει, ή έχει άγνοια των ειδικών κανόνων και, επομένως, αν η παρουσία του στο δίκτυο έχει εντοπιστεί. Μειονεκτήματα περιλαμβάνουν το σχετικό ενιαίο σημείο της αποτυχίας και το ενδεχόμενο της υπερφόρτωσης με μεγάλη κίνηση σε μεγαλύτερα δίκτυα. Όπως συμβαίνει για οποιοδήποτε επίπεδο ασφάλειας ενός συστήματος του δικτύου, το Network IDS παρακολουθεί μόνο hosts ενώ βρίσκονται συνδεδεμένοι με το σχετικό δίκτυο, αφήνοντας ανοιχτό το ενδεδόμενο επίθεσης σε φορητές συσκευές, που τυχόν λειτουργούν σε λιγότερο ασφαλή δίκτυα. Μία τέτοια μόλυνση με κακόβουλο λογισμικό αποτελεί μια πραγματική απειλή για την ασφάλεια των δικτύων καθώς οι συγκεκριμένες συσκευές παρακάμπτουν τα τείχη προστασίας του δικτύου κατά την επανασύνδεση τους στο δίκτυο.

Host IDS

Ένα Host IDS έχει πρόσβαση όχι μόνο στα δεδομένα κίνησης του δικτύου, αλλά σε όλες τις τοπικές κλήσεις συστήματος, αρχεία καταγραφής πυρήνα λειτουργικού συστήματος, αρχεία καταγραφής εφαρμογής, αρχεία καταγραφής εξοπλισμού δικτύου κλπ. Οποιαδήποτε γεγονότα που συνέβησαν στον κεντρικό υπολογιστή είναι δυνατόν να καταγραφούν και ως εκ τούτου ένα HIDS έχει περισσότερες πληροφορίες για την ανίχνευση επιθέσεων. Αυτό μπορεί να περιλαμβάνει παρακολούθηση της κυκλοφορίας της τοπικής διασύνδεσης δικτύου όπως και ένα Network IDS. Το κύριο όφελος ενός HIDS είναι η διάθεση περισσότερων πληροφοριών από ότι η κίνηση του δικτύου και μόνο. Ένα άλλο πλεονέκτημα είναι η συνεχής προστασία για τις φορητές συσκευές, καθώς και η συνεχής παροχή ασφαλείας ακόμα και όταν βρίσκονται εκτός δικτύου σε μη ασφαλή δίκτυα. Καθώς όλο και αυξάνονται οι συσκευές δικτύου κινητής τηλεφωνίας, αναμένεται ότι ο αριθμός των επιθέσεων που απειλούν τις φορητές συσκευές να αυξηθεί επίσης. Έτσι, οι κινητές και ασύρματες τεχνικές ανίχνευσης επίθεσης θα αποκτούν ολοένα και μεγαλύτερη σημασία [8]. Η διανομή της προστασίας σε όλο το δίκτυο είναι ένα άλλο όφελος, με αποτέλεσμα τα κοινά χαρακτηριστικά επεξεργαστή και απαιτήσεων μνήμης για όλους τους hosts επιτρέποντας έτσι την κλιμάκωση σε μεγαλύτερου μεγέθους δίκτυα. Μειονεκτήματα περιλαμβάνουν τη διαμόρφωση, ενημερώσεις, την εγκατάσταση και διορθώσεις σφαλμάτων τα οποία, όταν απαιτείται, θα είναι ενδεχομένως αναγκαία για τις εκατοντάδες των συστημάτων σε ένα δίκτυο.

Υβριδικά IDS

Ένα υβριδικό σύστημα συγχωνεύει τις δύο προηγούμενες αρχιτεκτονικές, προσφέροντας τα πλεονεκτήματα και των δύο τύπων του συστημάτων για τη μέγιστη κάλυψη απειλής. Ένα παράδειγμα υβριδικού συστήματος είναι το Prelude [65] που συνυπολογίζει τα αποτελέσματα από μια σειρά συστημάτων για ανάλυση σε μία κεντρική τοποθεσία, είτε host- είτε network-based.

Ανίχνευση βάσει υπογραφής

Μια άλλη μέθοδος για την κατηγοριοποίηση των IDS έγκειται στο μηχανισμό τους, για την ανίχνευση εισβολών. Βασισμένο σε κανόνες ανίχνευσης, γνωστό και ως υπογραφή εντοπισμού, ταιριάζει παρατηρήσεις έναντι υπογραφών ή σε κανόνες σε μια βάση δεδομένων γνωστών επιθέσεων, οι οποίες στην περίπτωση των NIDS είναι με τη μορφή των ακολουθιών πακέτων δικτύου ή του περιεχομένου τους. Όπως και στα προγράμματα εντοπισμού ιών που βασίζονται στην υπογραφή, τα μειονεκτήματα της συγκεκριμένης προσέγγισης είναι ότι μπορεί να ανιχνευφθούν μόνο γνωστές απειλές καθώς επίσης και ότι η βάση δεδομένων υπογραφών πρέπει να επικαιροποιείται. Ένα πλεονέκτημα είναι ο χαμηλός αριθμός των ψευδώς θετικών αποτελεσμάτων (false positives), δεδομένου ότι οι ειδοποιήσεις παράγονται με βάση συγκεκριμένους κανόνες. Ψευδώς θετικές ειδοποιήσεις (false positives) μπορεί ακόμη να προκύψουν ωστόσο μέσα από ακατάλληλη ή πλημμελούς κατασκευής αρχεία κανόνων. Ένα παράδειγμα signature-based IDS είναι το Snort.

Ανίχνευση βάσει Ανωμαλίας

Ανωμαλία ή ανίχνευση που βασίζεται στη συμπεριφορά μαθαίνει πώς ένα σύστημα συμπεριφέρεται συνήθως και αποκλίσεις πάνω από ένα ορισμένο όριο από τις αποδεκτές μετρήσεις αναφοράς θεωρούνται δυνητικά απειλητικές και αναφέρονται. Μπορούν να καθοριστούν όρια όσον αφορά την χρήση της CPU, χρήση μνήμης, τους τύπους πακέτων δικτύου, το ρυθμό με τον οποίο ο χρήστης πληκτρολογεί κλπ. [10]. Ένα ζήτημα με την ανίχνευση βάσει ανωμαλίας είναι ότι ένα σύστημα μπορεί να επανεκπαιδεύεται κακόβουλα με την πάροδο του χρόνου από έναν εισβολέα έτσι ώστε να αποδεχθεί την ανώμαλία στη συμπεριφορά ως φυσιολογική, εξαλείφοντας έτσι την πιθανότητα ανίχνευσης. Ένα εργαλείο για παράδειγμα που μπορεί να χρησιμοποιειθεί για την υλοποίηση του ανωτέρω είναι το rrdtool [11]. Τα δύο προαναφερθέντα συστήματα δύναται να συνδυαστούν. Η προσέγγιση αυτή υιοθετήθηκε στην ανάπτυξη ενός συστήματος από την SRI International [12]. Εδώ και των δύο κατηγοριών τα πλεονεκτήματα και μειονεκτήματα κληρονόμούνται, όπως για παράδειγμα η δυνατότητα παρακολούθησης για γνωστές υπογραφές, σε συνδυασμό με τη σύλληψη των δυνητικών νέων, αταξινόμητων επιθέσεων.

Ψευδώς θετικά & ψευδώς αρνητικά.

Ένα πρόβλημα που αντιμετωπίζουν τα IDS είναι ο όγκος των ανεφερθέντων ψευδών συνεγερμών. Ψευδώς θετικά (ψευδείς συναγερμοί) είναι οι ειδοποιήσεις που δημιουργούνται από το IDS και προέρχονται από φυσιολογική κίνηση δεδομένων στο δίκτυο. Ψευδώς αρνητικά αποτελέσματα (απώλειες) από την άλλη πλευρά είναι οι επιθέσεις που δεν ανιχνεύονται από το IDS. Οι κύριες απαιτήσεις ενός συστήματος ανίχνευσης εισβολής είναι οι χαμηλό ποσοστό ψευδώς θετικών και το υψηλό ποσοστό ανίχνευσης των πραγματικών επιθέσεων [13]. Πολλές μέθοδοι έχουν προταθεί για την μείωση των ψευδών θετικών, αλλά το θέμα παραμένει σημαντικό πρόβλημα [14]. Τα σφάλματα θα είναι σύνηθες φαινόμενο όταν ένα σύστημα εγκαθίσταται για πρώτη φορά, αλλά μέσω της ρύθμισης του συνόλου των κανόνων και των ρυθμίσεων, οι ψευδείς συνεγερμοί μπορούν να μειωθούν. Καθώς ο αριθμός των ψευδώς θετικών αυξάνεται, η πιθανότητα ο διαχειριστής IDS να χάνει πραγματικές επιθέσεις αυξάνεται ταυτόχρονα. Αποφάσεις σχετικά με τα γεγονότα που απαιτούν τη λήψη ενέργειων είναι στα χέρια του διαχειριστή και θα ποικίλουν ανάλογα με το εν λόγω δίκτυο [15].

Παράδειγματα συστημάτων

Υπάρχουν πολλά συστήματα IDS, τόσο ανοιχτού κώδικα όσο και εμπορικά που διατίθενται σήμερα. Τα δύο πιο δημοφιλή, σύμφωνα με Insecure.org [16] είναι το Snort και το Open Source Security, Host-Based Intrusion Detection System (OSSEC) [70]. Και τα δύο είναι, συστήματα ανοικτού κώδικα και εκτελούν ανίχνευση υπογραφής στο δίκτυο και σε επίπεδο host αντίστοιχα. Το OSSEC HIDS εκτελεί ανάλυση αρχείων καταγραφής, έλεγχο ακεραιότητας, ειδοποίηση βασισμένη στο χρόνο και ενεργή ανταπόκριση. Αρχιτεκτονικά, το σύστημα γενικά εγκαθίσταται σε όλα τα μηχανήματα που προορίζονται για έλεγχο, προώθώντας τους συνεγερμούς σε ένα σταθμό διαχείρισης για ανάλυση. Στην αγορά διατίθεται ως συμπλήρωμα και όχι ως υποκατάστατο των NIDS.

Το Snort παρακολουθεί και εκτελεί έλεγχο των πακέτων δεδομένων της κίνησης του δικτύου σε πραγματικό χρόνο ώστε να εντοπίσει τα πρότυπα που ορίζει ο διαχειριστής. Το σύνολο των κανόνων μπορεί να ενημερώνεται εύκολα και μπορεί να είναι τόσο ευρύ ή συγκεκριμένο αν και τροποποιήσεις έχουν άμεση επίδραση στον αριθμό των συνεγερμών που δημιουργούνται. Το Snort μπορεί να ρυθμιστεί να εκτελεί αποτροπή εισβολών (IP) αν είναι επιθυμητό, αποβάλλοντας τα πακέτα που επιχειρούν «μη αποδεκτή» συμπεριφορά, όπως stealth scanning. Τα αποτελέσματα μπορεί να δύσκολο να αποκρυπτογραφηθούν και πιθανά add-ons περιλαμβάνουν ανάλυση και απεικόνιση της βάσης δεδομένων και εργαλεία επεξεργασίας αρχείων. Εργαλεία καταγραφής πακέτων μπορούν να χρησιμοποιηθούν για να ανιχνεύσουν πακέτα δικτύου, επιτρέποντας στο διαχειριστή να προβάλει με μη αυτόματο τρόπο δεδομένα για απειλές. Η καταγραφή όλων των πακέτων για έλεγχο διασφαλίζει ότι το σύνολο της κίνησης ανιχνεύεται, αλλά η ανάλυση απαιτεί ιδιαίτερες ικανότητες. Αυτή η

προσέγγιση δεν είναι πρακτική σε βάση πλήρους απασχόλησης λόγω του όγκου της κίνησης που παράγεται από μία συσκευή, πόσο μάλλον για ένα ολόκληρο δίκτυο, αλλά είναι χρήσιμα για την παρακολούθηση για άγνωστες απειλές ή για την αξιολόγηση μετά από μια επίθεση. Δημοφιλή εργαλεία για την καταγραφή πακέτων αποτελούν το tcpdump [18]και το wireshark [19].

2.4 Μείωση των ψευδών συνεγερμών στο επίπεδο του HTTP μέσω της μεθόδου Procedure Analysis

Πολλές προσπάθειες έχουν γίνει από την ερευνητική κοινότητα για τη μείωση των ψευδώς θετικών ειδοποιήσεων που παράγονται από ένα IDS [20], εκ των οποίων ακολουθούν μερικές:

Fingerprinting Λειτουργικού συστήματος: Αυτό περιλαμβάνει false positives που συμβαίνουν σε ένα σενάριο όπου το περιβάλλον δεν είναι ευάλωτο. Αυτό το σενάριο υπάρχει επειδή τα περισσότερα IDS δικτύου δεν λαμβάνουν υπ' όψιν την ευπάθεια του προφίλ των host κατά την ανίχνευση επιθέσεων. Ένα πιθανό κλειδί για τον περιορισμό των false positives στο συγκεκριμένο σενάριο είναι η παραγωγή ενός περιβάλλοντος όπου οι πληροφορίες του host στόχου εντάσσονται στο πλαίσιο ανίχνευσης. Ως αποτέλεσμα, οι προς παρακολούθηση λεπτομέρειες των πακέτων του δικτύου μπορούν να συγκριθούν με πληροφορίες που είναι αποθηκευμένες στο προφίλ του λειτουργικού συστήματος του host στόχου. Εάν το αποτέλεσμα της σύγκρισης είναι θετικό, τότε θα μπορούσε να γίνει ανάλυση ανίχνευσης, διαφορετικά, το πακέτο δικτύου απορρίπτεται [20].

Εξάλειψη πλημμύρας Συναγερμών: Πρόκειται για μια προειδοποίηση για την ίδια εισβολή που διαδίδεται συνεχώς σε ολόκληρο το προς παρακολούθηση δίκτυο. Ένα παράδειγμα περιλαμβάνει την MS Blaster ή SQL Slammer outbreak [8] , με αποτέλεσμα το IDS δικτύου να παράγει συναγερμούς κατ 'επανάληψη για την ίδια εισβολή, προκαλώντας μια πλημμύρα σημάτων. Μια πιθανή λύση θα είναι να προ επεξεργαστούν "πιθανές ειδοποιήσεις" πριν από την έγερση βάσει κανόνων που χρησιμοποιούν παράμετρο που θα λαμβάνει υπόψη τον τύπο της ειδοποίησης, διευθύνσεις IP προέλευσης και προορισμού. Οι παράμετροι αυτές θα επιτρέψουν στο IDS να καταστέλλουν ταυτόσημες ειδοποιήσεις και απλώς να καταγράφονται για τις στατιστικές αναλύσεις [21].

Συσχέτιση Μετα-Alert: Πρόκειται για συνεγερμούς παράγονται από την συσχέτιση δύο ή περισσοτέρων καταχωρήσεων, ενδεχομένως, από διαφορετικούς αισθητήρες ανίχνευσης. Η

συσχέτιση Μετα-Alert επιτρέπει τη δημιουργία ενός συναγερμού υψηλότερης προτεραιότητας όταν ορισμένες προϋποθέσεις που συνδέονται με χαμηλότερου επίπεδου ειδοποιήσεις πληρούνται. Η απουσία των μετα-alert συσχέτισης θα οδηγήσει στη προβολή επιθετικών δραστηριοτήτων ως μεμονωμένα και ξεχωριστά γεγονότα, ενδεχομένως απορρίπτοντας μεμονωμένα συμβάντα που, όταν συσχετίζονται, θα μπορούσαν να οδηγήσουν σε έναν επιθετικό γεγονός. Παραμέτροι συσχέτισης Μετα-Συνεγερμών θα περιλαμβάνουν το χρονικό παράθυρο, τον αριθμό των γεγονότων και τον τύπο τους, IP διεύθυνση και αριθμό θύρας [20]. Μια πιθανή λύση για την αντιμετώπιση του υψηλού αριθμού των ψευδώς θετικών ειδοποιήσεων θα μπορούσε να είναι η πιο έξυπνη σχεδίαση των μηχανών ανίχνευσης επιθετικών ενεργειών, που ενδέχεται να περιλαμβάνουν ενιαίο πλαίσιο για την ανίχνευση για ανάλυση, για την παραγωγή συναγερμών και για τους παραμετρικούς κανόνες και το συσχετισμό και την ομαδοποίηση των κανόνων. Μια άλλη εναλλακτική λύση στον περιορισμό των ψευδώς θετικών ειδοποιήσεων θα μπορούσε να μοντελοποιήσει το πρωτόκολλο επικοινωνίας και στη συνέχεια να χρησιμοποιηθεί η σύνταξη ή / και η σημασιολογία που σχετίζονται με αυτό το πρωτόκολλο για το σχεδιασμό επιθετικών υπογραφών. Το τελευταίο περιγράφεται στην παρούσα παράγραφο. Τα περισσότερα βάσει δικτύου συστήματα ανίχνευσης εισβολών (NIDS) τυπικά βασίζονται στην αντιστοίχιση υπογραφής. Παραδείγματα αποτελούν τα Snort και Bro [22], τα οποία προσφέρουν έναν απλό τρόπο για να καταγραφούν οι υπογραφές και να περιοριστούν οι υπηρεσίες σε μια σειρά από αξιόπιστες διευθύνσεις. Ωστόσο, είναι λίγο πιο δύσκολο να αποφορτιστεί ο διαχειριστής του δικτύου από το έργο της διατήρησης των υπογραφών και της ενημέρωση από την παρακολούθηση της κίνησης . Συστήματα όπως ADAM [10], NIDES [23], και Emerald [24] κάνουν ακριβώς αυτό. Τα προαναφερθέντα συστήματα χρησιμοποιούν μια βάση δεδομένων έμπειρου συστήματος που αποτελείται από επιθετικές υπογραφές, κωδικοποιημένες με τη γνώση που αποκτήθηκε από εμπειρογνώμονες ασφάλειας σε αρχεία δοκιμής ή από την κυκλοφορία του δικτύου για πρότυπα που είναι γνωστό ότι συμβαίνουν σε επιθέσεις. Ως εκ τούτου ελάχιστες διακυμάνσεις της μεθόδου επίθεσης μπορούν να νικήσουν πολλές φορές τέτοια συστήματα. Κανένα από αυτά τα IDS δεν αποσκοπούν στην ενίσχυση της τεχνικής ανίχνευσης για τη μείωση των ψευδώς θετικών ειδοποιήσεων, αλλά απλώς εφαρμόζουν καθιερωμένου μοντέλου τεχνικές. αντιστοίχισης υπογραφών. Οι ερευνητικές εργασίες που διεξάγονται στο [25], μπορεί να εκφραστεί σε δύο στάδια: πρώτον, πραγματοποιείται στατιστική ανάλυση της κανονικής και εχθρικής κυκλοφορίας. Τα πειραματικά αποτελέσματα της ανάλυσης αυτής αποκαλύπτουν ότι ορισμένα στοιχεία που προέρχονται από HTTP αιτήσεις μπορούν να χρησιμοποιηθούν για τη διάκριση μη ομαλής (και ως εκ τούτου, ύποπτης) κυκλοφορίας που αντιστοιχεί σε σωστές κανονικές συνδέσεις. Το δεύτερο μέρος της έρευνάς παρουσιάζει μια νέα τεχνική βασισμένη στον εντοπισμό ανωμαλίας για την ανίχνευση επιθέσεων που πραγματοποιήθηκαν κατά την HTTP κυκλοφορία. Η τεχνική εισήγαγε τη στατιστική και

κάνει χρήση των Markov αλυσίδων [26] στη μοντελοποίηση της HTTP κυκλοφορίας στο δίκτυο. Η εισερχόμενη κίνηση HTTP αποτελεί παράμετρο για την αξιολόγηση με βάση το ωφέλιμο φορτίο σε ένα πακέτο δικτύου. Έτσι, το ωφέλιμο φορτίο κάθε πακέτου στο δίκτυο κάθε σύνδεσης HTTP κατακερματίζεται σε έναν ορισμένο αριθμό συνεχόμενων μπλοκ, τα οποία είναι συνεχώς κβαντισμένα σύμφωνα με μια προηγουμένως εκπαιδευμένη βαθμωτή κωδικοσειρά. Τέλος, η χρονική ακολουθία των συμβόλων που λαμβάνονται αξιολογείται μέσω ενός μοντέλου Markov που προκύπτει κατά τη διάρκεια της φάσης της εκπαίδευσής τους. Ένας απλός οπτικός έλεγχος διενεργείται στο μήκος του ωφέλιμου φορτίου για τον υπολογισμό του ιστογράμματος του ωφέλιμου φορτίου και της τυπικής απόκλισης, μέσω ομαδοποίησης της κυκλοφορίας του δικτύου σε πρωτόκολλο και σε υπηρεσίες. Τέλος εξετάζονται τα δεδομένα που συλλαμβάνονται από έναν NetHost-αισθητήρα για να εκτελέσει μια ανάλυση διαδικασίας(procedure analysis) τεχνική που μοντελοποιεί τα δεδομένα του δικτύου στο επίπεδο του HTTP για την ανίχνευση και τη μείωση των ψευδώς θετικών ειδοποιήσεων. Οι εργασίες που διεξάγονται εκφράζονται σε δύο κύρια στάδια: πρώτον, μοντελοποίηση δεδομένων στο επίπεδο του HTTP, και δεύτερον, με βάση το μοντέλο δεδομένων στο επίπεδο του HTTP που δημιουργήθηκε ένα πρωτόκολλο διαδικασίας που χρησιμοποιεί τυπική σύνταξη και η σημασιολογία σχεδιάζεται αποσκοπώντας στη δημιουργία επιθετικών υπογραφών για τη μείωση των ψευδώς θετικών ειδοποιήσεων.

2.5 Χρήση Neuro-Fuzzy τεχνικών για τη μείωση των ψευδών συνεγερμών στα συστήματα ανίχνευσης εισβολών

Το IDS που εξετάστηκε πιο προσεκτικά σε αυτή τη μελέτη [27], είναι το Snort, που είναι ένα σύστημα ανίχνευσης εισβολων δικτύων (NIDS) που βασίζεται σε κανόνες. Το snort είναι ένα ανεξάρτητης-πλατφόρμας, ελαφρύ εργαλείο εντοπισμού εισβολων δικτύων που μπορεί να εγκατασταθεί για την παρακολούθηση μικρών δικτύων TCP / IP και να εντοπίσει μια μεγάλη ποικιλία από ύποπτη κυκλοφορία του δικτύου καθώς και επιθέσεων. Το Ινστιτούτο SANS ανέφερε επίσης για το Snort ότι θα καταστεί το πρότυπο μεταξύ των εμπειρογνωμόνων ανίχνευσης εισβολής λόγω του γεγονότος ότι είναι ανοιχτού κώδικα, που ενημερώνεται συχνά, και διατίθεται δωρεάν. Το πρόβλημα των ψευδών ειδοποιήσεων στο Snort που όπως έχει ήδη αναφερθεί είναι ένα από τα από τα κύρια προβλήματα στους υπάρχοντες αισθητήρες ασφαλείας αποτελεί η τάση τους να παράγουν υψηλά ποσοστά ψευδών θετικών ειδοποιήσεων. Συχνά, ένας λανθασμένος συναγερμος δημιουργείται όταν στην πραγματικότητα το γεγονός που προκάλεσε το συναγερμό μπορεί

να θεωρηθεί ακίνδυνο. Αυτή η κατάσταση επιδεινώνεται όταν ο επιτιθέμενος έχει κάποια εκ των προτέρων γνώση των τεχνικών που χρησιμοποιούνται από τον αισθητήρα ασφάλειας, έτσι εσκεμμένα δημιουργεί δεδομένα δικτύου για να προκαλέσει αυτούς τους εσφαλμένους συναγερμούς. Αυτό όχι μόνο θα επιτρέψει σε έναν εισβολέα να αποκτήσει τον έλεγχο των αισθητήρων ασφαλείας, αλλά και θα αποτρέψει την ικανότητα του αισθητήρα ασφαλείας να λειτουργήσει σωστά λόγω του μεγάλου ποσού της κίνησης που έχει να αντιστοιχίσει με τους κανόνες του ή άλλιώς προκαλώντας τους μηχανισμούς συναγερμού, και ως εκ τούτου, να έχουμε σπατάλη των πόρων επεξεργασίας. Αν και το Snort είναι ένα εξαιρετικό εργαλείο, έχει τρία σημαντικά προβλήματα:

- Packet Dropping.
- False Positive Alerts.
- False Negative Alerts.
-

Το Snort δεν μπορεί να επεξεργαστεί όλα τα πακέτα, λόγω θεμάτων ταχύτητας σε ένα δίκτυο. Άλλοι παράγοντες που μπορούν να επηρεάσουν Snort σε αυτό το τρόπο είναι η ταχύτητα της διεπαφής δικτύου και η υλοποίηση της στοίβας του λειτουργικού συστήματος. Είναι σημαντικό να σημειωθεί ότι το Snort είναι δυνατόν να δεχθεί πλημμύρα πακέτων που κάνει τότε την ανίχνευση εισβολών πιο δύσκολη. Ψευδώς θετικές ειδοποιήσεις συμβαίνουν όταν το Snort στέλνει ειδοποιήσεις ενώ δεν θα έπρεπε, με άλλα λόγια ένα ψευδή συναγερμό. Αυτό μπορεί να συμβεί για διάφορους λόγους. Ορισμένοι από αυτούς περιλαμβάνουν:

Τοποθέτηση του Snort εκτός της της περιμέτρου ασφάλειας: Σε αυτή την περίπτωση το Snort λαμβάνει σαρώσεις DNS, σαρώσεις proxy web και διάφορες άλλες καλοήθεις ενέργειες στο πληροφοριακό δίκτυο που θα προκαλέσουν υπερφόρτωση για το διαχειριστή του συστήματος.
Πολιτική ιστοσελίδας που επιτρέπει δραστηριότητα που προκαλεί IDS συναγερμούς: Για παράδειγμα, χρησιμοποιώντας την προεπιλεγμένη ρύθμιση για το το Snort που θα αυξήσει την εισροή δεδομένων σε ένα μη διαχειρίσιμο επίπεδο.
Η έλλειψη ενημέρωσης των εφαρμογών που φιλοξενούνται σε ένα δίκτυο στο IDS: Η μη γνώση των υπηρεσιών που λειτουργούν στους hosts, όπως IIS επιθέσεις σε Apache διακομιστές Web θα μπορούσε να οδηγήσει σε ψευδείς συναγερμούς. Εσφαλμένα θετικά (False Positives) συμβαίνουν λόγω οποιασδήποτε επίθεσης δεν αντιστοίχιθεί μια υπογραφή στη βάση δεδομένων με τις «γνωστές επίθεσεις». Αυτό μπορεί να συμβεί λόγω του κακού σχεδιασμού κανόνων, κρυπτογραφημένη ή με άλλο τρόπο έξυπνη [2] συγκεκαλυμμένη κυκλοφορία, ή απλώς επειδή η επίθεση είναι νέα και δεν είναι

αντιστοιχισμένη η υπογραφή της. Η προτεινόμενη λύση βασίζεται σε Τεχνικές Τεχνητής Νοημοσύνης, που αναμένεται να βελτιώσει το ποσοστό της μείωσης των ψευδώς θετικών ειδοποιήσεων. Επίσης, η λύση θα πρέπει να είναι σε θέση να καλύψει το κύριο πρόβλημα της Neuro - Fuzzy τεχνικής, η οποία δεν θα μπορούσε να μειώσει τον αριθμό των ψευδώς αρνητικών αρκετά σημαντικά. Ο πρώτος και κύριος στόχος είναι ο σχεδιασμός και η η υλοποίηση μιας ευφυούς τεχνικής που επιτρέπει στο σύστημα (IDS) τη μείωση των εσφαλμένων συναγερμών. Δεύτερον, το σύστημα θα πρέπει να ρυθμιστεί με λεπτομέρεια έτσι ώστε ο αριθμός των ψευδώς αρνητικών να είναι επίσης μειωμένος. Η Τεχνητή Νοημοσύνη είναι ένας τομέας της επιστήμης των υπολογιστών που προσπαθεί να μιμηθεί ή να αντιγράψει την ανθρώπινου τύπου σκέψη και δράση. Σε αντίθεση με την απλή επεξεργασία των πληροφοριών με την επιλογή δηλώσεων και μνήμη εργασίας, η τεχνητή νοημοσύνη επιχειρεί να επαναλάβει τη διαδικασία της σκέψης, όπως η λογική, διαίσθηση, μαθαίνοντας από το παρελθόν, δοκιμής και του λάθους, και γενικεύσεις [28]. Αν και δύσκολο, κάποια επιτυχία στην αναπαραγωγή της ανθρώπινης νοημοσύνης έχει επιτευχθεί με τα αποκαλούμενα ως έμπειρα συστήματα. Συνήθως τα συστήματα αυτά βρίσκονται σε πολύ ισχυρά μηχανήματα που λειτουργούν σε εξαιρετικά υψηλές ταχύτητες και τα προγράμματα από μόνα τους είναι πολύ πολύπλοκα. Τα έμπειρα συστήματα ανήκουν στην πραγματικότητα σε μια κατηγορία τεχνητής νοημοσύνης γνωστά ως συστήματα που βασίζονται σε κανόνες [28]. Όσο περισσότερο ένα σύστημα ανίχνευσης εισβολών (IDS) γνωρίζει για το δίκτυο που προσπαθεί να προστατεύσει, τόσο καλύτερα θα είναι σε θέση να προστατεύσει το δίκτυο. Αυτή είναι η βασική αρχή που διέπει τα συστήματα ανίχνευσης εισβολής που προορίζονται για ένα συγκεκριμένο δίκτυο, το IDS να γνωρίζει τους hosts του δικτύου. Το σύστημα ανίχνευσης εισβολών Snort διαθέτει κάποια χαρακτηριστικά μη ευρέως διαδεδομένα που εάν χρησιμοποιηθούν είναι δυνατόν το σύστημα ανίχνευσης εισβολών να προσαρμοστεί στο δίκτυο για τον εντοπισμό μη ομαλής συμπεριφοράς. Μέρος από το φόρτο εργασίας των υπευθύνων ασφάλειας μπορεί να εξαλειφθεί αρχικά μαθαίνοντας για ένα δίκτυο και μετρώντας αντιδράσεις από τους υπευθύνους ασφαλείας για τη μείωση των ψευδών θετικών, και δεύτερον, με την προσαρμογή μεταβολών στο δίκτυο για τον εντοπισμό νέων επιθέσεων. Υπάρχουν πολλές διαφορετικές τεχνικές και αλγόριθμοι που μπορεί να χρησιμοποιηθούν με επιτυχία για την ανίχνευση εισβολών.

Αυτές οι τεχνικές περιλαμβάνουν:

- Ασαφής Λογική.
- Πιθανοτική Λογικής.
- Νευρωνικά Δίκτυα.
- Γενετικοί Αλγόριθμοι.

Οι συνδυασμοί αυτών μπορούν επίσης να χρησιμοποιηθούν. Για παράδειγμα, γενετικοί αλγόριθμοι μπορεί να χρησιμοποιηθούν για να οικοδομήσουμε ένα νευρωνικό δίκτυο και πιθανοτική λογική μπορεί να βασίζεται στην ασαφή λογική. Ένα νευρο-ασαφές δίκτυο μπορεί να οριστεί ως ένα συγκεχυμένο συστήμα εκπαιδευμένο με κάποιο αλγόριθμο που προέρχεται από τη θεωρία νευρωνικών δικτύων. Η ολοκλήρωση των νευρωνικών δικτύων και των ασαφών συστημάτων στοχεύει στην δημιουργία ενός πιο εύρωστου, αποτελεσματικού και εύκολα ερμηνεύσιμου συστήματος, όπου τα πλεονεκτήματα του κάθε μοντέλου διατηρούνται και αφαιρούνται τα πιθανά μειονεκτήματα τους. Ορισμένα νευρωνικά μοντέλα δικτύου, όπως το MLP [29] έχουν εφαρμόστεί με επιτυχία στην εκπαίδευση των νευρο-ασαφών δικτύων. Το μοντέλο NEFCLASS που προτάθηκε από τον Nauck και τον Kruse [30] βασίζεται σε ένα τριών επιπέδων feedforward νευρωνικό δίκτυο [29] και τα τη FuNN (Fuzzy Neural Network) που προτάθηκε από τον Kasabov είναι ένα πέντε στρωμάτων feedforward νευρωνικό δίκτυο. Και στα δύο δίκτυα χρησιμοποιούνται τροποποιημένες εκδόσεις του αλγορίθμου πίσω-διάδοσης έτσι ώστε να προσαρμοστούν οι λειτουργίες των μελών (λειτουργίες ενεργοποίησης) και τα βάρη σύνδεσης των μονάδων επεξεργασίας. Σύγχρονες νευρο-ασαφές προσεγγίσεις είναι της μορφής: Ένα νευρωνικό δίκτυο και ένα ασαφές του σύστημα συνδυάζονται σε μια ομοιογενή αρχιτεκτονική. Το σύστημα μπορεί να ερμηνευθεί είτε ως ειδικό νευρωνικό δίκτυο με ασαφείς παραμέτρους, ή ως ένα ασαφές σύστημα υλοποιημένο σε κατανεμημένη παράλληλη μορφή. Μερικές από αυτές τις προσεγγίσεις είναι τύπου εκμάθησης που είναι ιδιαίτερα κατάλληλο για για τον έλεγχο εργασιών, άλλα είναι πολλαπλών χρήσεων μοντέλων όπου χρησιμοποιείται η επιβλεπόμενη μάθηση, και μπορεί να χρησιμοποιηθεί για την ανάλυση δεδομένων, όπως κατά την προσέγγιση NEFCLASS. Ένα neuro fuzzy σύστημα έχει τα παρακάτω χαρακτηριστικά:

1) Ένα νευρο-ασαφές σύστημα είναι ένα σύστημα ασαφές που έχει εκπαιδευτεί από έναν (heuristical) αλγόριθμο μάθησης που προέρχεται από (συνήθως) νευρωνικά δίκτυα.

2) Έναν νευρο-ασαφές σύστημα μπορεί να αναπαρασταθεί από μία feedforward νευρωνική αρχιτεκτονική δικτύου. Ωστόσο, αυτό δεν αποτελεί προϋπόθεση για κατάρτιση, είναι απλώς μια διευκόλυνση για να απεικονίστει η δομή και η ροή των δεδομένων.

3) Ένα νευρο-ασαφές σύστημα μπορεί να ερμηνεύεται πάντα μέ όρους των ασαφών if-then κανόνες.

4) Η διαδικασία κατάρτισης ενός νευρο-ασαφούς συστήματος έχει τη σημασιολογία του υποκείμενου ασαφούς μοντέλου υπόψη της διατήρησης της γλωσσική interpretability του μοντέλου.

5) Ένα νευρο-ασαφές σύστημα εκτελεί (ειδικές περιπτώσεις) λειτουργίες προσεγγίσεως. Δεν έχει τίποτε να κάνει με την ασαφή λογική με τη στενή έννοια, δηλαδή γενικευμένους κανόνες λογικής.

Για να μειωθεί το ποσοστό των ψευδών θετικών ειδοποιήσεων ενός IDS, χρειαζόμαστε μια μέθοδο, η οποία είναι σε θέση να αντιμετωπίσει την αβεβαιότητα στο δίκτυο κίνηση και να προβλέψει απρόβλεπτα και θορυβώδη δεδομένα με ακρίβεια. Επιπλέον,οιπληροφορίες που παρέχονται για τις ειδοποιήσεις μέσω δεδομένα ελέγχου και αρχείων καταγραφής δεν διαθέτουν επαρκή στοιχεία σχετικά με τα χαρακτηριστικά των συνδέσεων που πραγματοποιούνται στο δίκτυο. Συστήματα που βασίζονται σε ασαφείς κανόνες παρέχουν τη δυνατότητα εξήγησης των ασαφών προτύπων των χαρακτηριστικών των σημάτων. Ωστόσο, αυτά τα χαρακτηριστικά των ειδοποιήσεων χρησιμοποιούνται για την εκμάθηση των ασαφών κανόνων του IDS είναι συνήθως υψηλά σε διαστάσεις. Για παράδειγμα, οι ειδοποιήσεις που δημιουργούνται από το DARPA 1999 σύνολο δεδομένων περιέχει πολλά χαρακτηριστικά που πρεπει να αναλυθούν. Κάθε γνώρισμα έχει ένα αριθμό από διάφορες πιθανές τιμές που κυμαίνονται από μικρό αριθμό δυνατών τιμών (π.χ. ο αριθμός των πρωτοκόλλων) στον τεράστιο αριθμό από πιθανές τιμές (π.χ. η διεύθυνση IP). Ως εκ τούτου, δεν είναι εύκολο να προσδιοριστεί ρητά οι συναρτήσεις συμμετοχής για τους ασαφείς κανόνες. Για αυτόν τον τύπο γνωστικού υπόβαθρου μία προσέγγιση, νευρωνικού δικτύου είναι αποδεκτή ως μια ισχυρή μέθοδος μάθησης για να μάθουν από το μηδέν. Για τους λόγους αυτούς, η προσέγγιση νευρωνικού δικτύου μπορεί να αποδειχθεί χρήσιμη προσέγγιση μάθησης για να βελτιώσει τα Ασαφή Σύνολα και τη συνάρτηση ένταξης στην ώστε να είναι κατάλληλη με το σύνολο δεδομένων.

2.6 Ανάπτυξη προσαρμοσμένων φίλτρων ανίχνευσης εισβολής με με τεχνικές εξόρυξης δεδομένων

Στόχος είναι να προσδιοριστούν τα πρότυπα των ψευδών ειδοποιήσεων που προέρχονται από τα συστήματα ανίχνευσης εισβολής. Χρησιμοποιώντας γενικευμένα συχνά επεισόδια, μια τεχνική εξόρυξη δεδομένων, για την ανάλυση εξόδου συστήματος ανίχνευσης εισβολής. Αυτό προσδιορίζει κοινές, επαναλαμβανόμενες αλληλουχίες των συναγερμών για μια δεδομένη τοποθεσία. Αυτά μπορούν να αναλύονται με μη αυτόματο τρόπο ώστε να καθορίσουν εάν αυτά προκύπτουν από τις συνήθεις εργασίες σε εκείνη την τοποθεσία. Αυτό θα επιτρέψει την ανάπτυξη site-specific φίλτρων για τη μείωση της ροής των πληροφοριών από τα συστήματα ανίχνευσης εισβολής. Δεδομένου ότι οι ακολουθίες συναγερμού είναι γνωστό ότι είναι κοινές, η ανταμοιβή (από την άποψη της μείωσης των λανθασμένων συναγερμών) είναι υψηλή. Πολλά συμβατικά συστήματα ανίχνευσης εισβολής βασίζονται σε υπογραφές σχετικά με τις επίθεσεις: μοντέλα της κίνησης του δικτύου που ταιριάζουν με γνωστή ή πιθανή εισβολή. Αυτό λειτουργεί καλά για εύκολες περιπτώσεις, όπως κοινά διαθέσιμα προγράμματα που εκμεταλλεύονται γνωστά κενά ασφαλείας. Ωστόσο, ο

εντοπισμός περισσότερων προηγμένων επιθέσεων απαιτεί γενικότερες υπογραφές. Αναγνωρίζονται πρότυπα της δραστηριότητας που ενδέχεται να σχετίζονται με επιθέσεις, αντί να τα ταιριάζονται σε γνωστές επιθέσεις. Τέτοιες υπογραφές μπορεί επίσης να προκληθούν από τις συνήθεις εργασίες, με αποτέλεσμα ψευδείς συναγερμούς. Στα συστήματα ανίχνευσης εισβολών υπάρχει μια αντίστροφη σχέση μεταξύ της ευαισθησίας, προσδιορίζοντας όλες τις εισβολές, και τον αποκλεισμό κανονικής συμπεριφορά ως ύποπτης. Εμπορικά εργαλεία βελτιστοποιούν την σχέση αυτή για τυπικά περιβάλλοντα. Για ένα σύνηθες περιβάλλον, αυτό έχει ως αποτέλεσμα σε ένα αποδεκτό επίπεδο ψευδών συναγερμών και χαμένων εισβολών. Ωστόσο, κανένα περιβάλλον δεν είναι στην πραγματικότητα πρότυπο. Αν και τα εμπορικά συστήματα ανίχνευσης εισβολών επιτρέπουν σε ορισμένους χρήστες ευελιξία προσαρμόζοντας την ανταλλαγή μεταξύ ανίχνευσης εισβολών - ψευδών συναγερμών, σε πολλά περιβάλλοντα το ποσοστό ψεύτικων συναγερμών παραμένει σε πολύ υψηλά επίπεδα. Η προσέγγισή στη συγκεκριμένη δημοσίευση είναι να αναπτυχθούν προσαρμοσμένα φίλτρα που μειώνουν το ρεύμα ψευδών συναγερμών με βάση γνωστή "φυσιολογική συμπεριφορά" σε ένα συγκεκριμένο περιβάλλον. Χρησιμοποιούνται εμπορικά συστήματα ανίχνευσης εισβολών, αλλά φιλτράρονται οι παραγόμενοι συναγερμοί που ταιριάζουν σε ένα πρότυπο που προκαλείται από την κανονική λειτουργία σε ένα συγκεκριμένο site. Η δυσκολία με αυτήν την προσέγγιση είναι η οικοδόμηση αυτών των φίλτρων, και ο προσδιορισμός του τι είναι η κανονική λειτουργία σε μια τοποθεσία. Μολονότι είναι πολύ λιγότερο δαπανηρή ενέργεια από την κατασκευή ενός πλήρους συστήματος ανίχνευσης εισβολών και αυτή η ενέργεια απαιτεί σημαντική ανθρώπινη προσπάθεια. Η προσέγγισή που ακολουθείται για τη μείωση αυτής της προσπάθειας είναι η χρήση τεχνολογίας εξόρυξης δεδομένων για την ανακάλυψη ειδοποιήσεων που προκαλούνται από τις συνήθεις λειτουργίες. Αναπτύσσονται φίλτρα με βάση ακολουθίες συναγερμών. Η ιδέα είναι ότι μια σειρά από λειτουργίες που είναι φυσιολογικές σε ένα συγκεκριμένο περιβάλλον μπορεί να περιέχουν λειτουργίες που φαίνονται σαν μια πιθανή εισβολή. Ωστόσο, η πλήρης αλληλουχία είναι απίθανο να αναπαραχθεί σε μια εισβολή, έτσι συναγερμοί που είναι μέρος της πλήρης ακολουθίας μπορούν να αγνοηθούν. Το πρόβλημα έγκειται στην αναγνώριση αυτών των κανονικών ακολουθιών. Ο λόγος που είναι σημαντικό να βρεθούν τέτοια συχνά επεισόδια; Στόχος είναι ο εντοπισμός ακολουθιών συναγερμών που οφείλονται σε κανονική λειτουργία. Συχνά επεισόδια είναι οι ακολουθίες των συναγερμών που συμβαίνουν συχνά - αυτό μας δίνει δύο πράγματα:

1. Μια κοινή ακολουθία των συναγερμών δεν είναι πιθανώς το αποτέλεσμα των πραγματικών προσπαθειών εισβολής - εισβολείς πιθανότατα να μην προσπαθήσουν κατ 'επανάληψη την ίδια μέθοδο. Ωστόσο, η κανονική λειτουργία επαναλαμβάνεται. Κατά συνέπεια, μια συχνά εμφανιζόμενη ακολουθία των συναγερμών είναι ένας καλός

υποψήφιος να έχει προκληθεί από την κανονική λειτουργία.
2. Περιμένουμε ένα φυσικό πρόσωπο για να καθορίσει εάν μια ακολουθία είναι αποτέλεσμα κανονικής λειτουργίας. Αναλύοντας συχνές ακολουθίες είναι εγγυημένο να έχει ως αποτέλεσμα τη μεγαλύτερη μείωση του ρεύματος των ψευδών συναγερμών, εάν οι ακολουθίες μπορούν να φιλτραριστούν. Έτσι η εφαρμογή της ανθρώπινης προσπάθειας αποφέρει το μεγαλύτερο όφελος.

2.7 Ομαδοποίησης συνεγερμών σε συστήματα ανίχνευσης εισβολών για την υποστήριξη Root cause ανάλυσης

Στο επίκεντρο αυτής της προσέγγισης είναι η αντίληψη ότι κάθε συναγερμός ενεργοποιείται για έναν λόγο, που αναφέρεται ως γενεσιουργά αίτια του συναγερμού. Αυτό το άρθρο επισημαίνει ότι σε λίγες δεκάδες των μάλλον επίμονων βασικών αιτιών γενικά αντιπροσωπεύουν άνω του 90% των συναγερμών που ένα σύστημα ανίχνευσης εισβολών ενεργοποιεί. Ως εκ τούτου, υποστηρίζεται ότι συναγερμοί πρέπει να αντιμετωπίζονται με τον εντοπισμό και την αφαίρεση των πιο κυρίαρχων και επίμονων αιτιών. Για να γίνει αυτό το μοντέλο εφικτό, προτείνεται μία νέα μέθοδος ομαδοποίησης συναγερμών που υποστηρίζει τον αναλυτή για τον εντοπισμό της αίτιας. Σε πειράματα που αφορούν πραγματικά σενάρια συστημάτων ανίχνευσης εισβολών φαίνεται πώς η ομαδοποίηση συναγερμών συνετέλεσε στον εντοπισμό των βασικών αιτιών. Επιπλέον, δείχνεται ότι το φορτίο των συναγερμών μειώνεται αρκετά σημαντικά αν τα ταυτοποιημένα βασικά αίτια εξαλειφθούν έτσι ώστε να μην ενεργοποιούν πλέον συναγερμούς στο μέλλον. Παρουσιάζεται διαγραφή ημι-διαγραφή μία προσέγγιση για το χειρισμό ανίχνευσης εισβολών συναγερμών effciently. Στο επίκεντρο αυτής της προσέγγισης είναι η έννοια της βασικής αιτίας του συναγερμού. Διαισθητικά, η πρωταρχική αιτία του συναγερμούς είναι ο λόγος για τον οποίο συμβαίνει. Πραγματοποιείται η βασική παρατήρηση ότι στα περισσότερα περιβάλλοντα, υπάρχει ένας σχετικά μικρός αριθμός εξαιρετικών κυρίαρχων αιτιών. Παρατηρήθηκε ότι μερικές δεκάδες βασικές αιτίες γενικά έχουν ως αποτέλεσμα το άνω του 90% όλων των συναγερμών. Επιπλέον, αυτά είναι συνεχή, δηλαδή δεν εξαφανίζονται, εκτός αν κάποιος τα αφαιρέσει. Αποτελούν ιδιαίτερο πρόβλημα επειδή προκαλούν ψευδείς συναγερμούς που εμποδίζουν τον αναλυτή ανίχνευσης εισβολών από τον εντοπισμό των πραγματικών επιθέσεων. Ερευνήσαμε ως εκ τούτου τεχνικές που για αποτελεσματικά χειρίζονται τέτοιες μεγάλες ομάδες περιττών συναγερμών. Το αποτέλεσμα της έρευνάς αυτής ήταν μια ημι-αυτόματη διαδικασία που αποτελείται από δύο στάδια: Στο

πρώτο βήμα, το οποίο συμβατικά ονομάζεται root cause ανάλυση, αναγνωρίζεται η βασική αίτια που ευθύνεται για το μεγάλο αριθμό των συναγερμών. Στο δεύτερο βήμα πραγματοποιεί την αφαίρεση αυτών και ως εκ τούτου μειώνει σημαντικά το μελλοντικό φορτίο των συνεγερμών. Τα πειράματα που παρουσιάζονται σε αυτό το άρθρο δείχνουν πως η μία προσπάθεια για τον εντοπισμό και την εξάλειψη των περιττών συνεγερμών αποδίδει, μειώνοντας το μελλοντικό φορτίο των συναγερμών κατά 87%. Αυτό είναι σημαντικό επειδή επιτρέπει στον αναλυτή εντοπισμού εισβολέων να επικεντρωθεί στο υπόλοιπο 13% των συναγερμών. Το άρθρο αυτό επικεντρώνεται στο πρώτο από τα παραπάνω δύο βήματα, δηλαδή με την αναγνώριση των κύριων αιτιών των περιττών συνεγερμών. Για αυτό το βήμα, αναπτύχθηκε μια μέθοδος ομαδοποίησης των συνεγερμών. Το κίνητρο για τη μέθοδο αυτή πηγάζει από την παρατήρηση ότι οι συναγερμοί μίας δεδομένης αιτίας είναι γενικά \ παρόμοιοι. Η μέθοδος ομαδοποίησης αντιστρέφει αυτή την εμπλοκή ομαδοποιώντας παρόμοιους συναγερμούς , θεωρώντας ότι οι αυτοί οι συναγερμοί μοιράζονται επίσης την ίδια πρωταρχική αιτία. Για κάθε συστάδα του συναγερμών, ορίζουμε ένα αποκαλούμενο γενικευμένο συναγερμό. Ένας γενικευμένος συναγερμός είναι ένα πρότυπο με το οποίο θα πρέπει να αντιστοιχιθεί ένας συναγερμός έτσι, ώστε να ανήκει στο αντίστοιχο σύμπλεγμα. Η γνώση γενικευμένων συναγερμών απλουστεύει κατά πολύ την ανάλυση. Ακολουθεί ένα παράδειγμα ομαδοποίσης συνεγερμών και ανάλυσης αρχικών αιτιών. Ας θεωρήσουμε την αιτία μιας σπασμένης TCP / IP στοίβας, που τεμαχίζει το σύνολο της εξερχόμενης IP κυκλοφορίας και ως εκ τούτου προκαλεί Fragmented IP alarms. Επιπλέον, ας υποθέσουμε ότι η TCP / IP στοίβα ανήκει σε ένα δημοφιλή διακομιστή Web που χρησιμοποιείται κυρίως κατά τις εργάσιμες ημέρες. Σαφώς, όλα τα Fragmented IP alarms έχουν την ίδια διεύθυνση IP προέλευσης (δηλαδή η διεύθυνση IP του Web server) και την ίδια θύρα θύρα (ήτοι 80). Οι προορισμοί των συναγερμών είναι θύρες των διαφόρων web clients Web. Δεδομένου ότι ο διακομιστής Web ως επί το πλείστον χρησιμοποιείται τις εργάσιμες ημέρες, προκύπτει ότι η πλειοψηφία των συναγερμών λαμβάνει χώρα κατά τις εργάσιμες ημέρες. Τέλος, οι Fragmented IP alarms ενεργοποιούνται κάθε φορά που ο διακομιστής Web ανταποκρίνεται σε αίτημα πελάτη. Με βάση την παραδοχή ότι ο διακομιστής Web είναι δημοφιλής και ως εκ τούτου χρησιμοποιείται πολύ, προκύπτει ότι πλήθος από Fragmented IP alarms. Η μέθοδος ομαδοποίησης των συναγερμών ομαδοποιεί τα Fragmented IP alarms και τα αναφέρει ως ένα γενικευμένο συναγερμό. Αυτός ο γενικευμένος συναγερμος δηλώνει ότι η θύρα προέλευσης 80 του διακομιστή Web ενεργοποιεί πολλούς Fragmented IP Alarms κατά τις εργάσιμες ημέρες κατά των μη προνομιούχων θυρών των πελατών Ιστού. Είναι σαφές ότι ένας γενικευμένος συναγερμός, όπως ο παραπάνω διευκολύνει την ταύτιση των βαθύτερων αιτίων (root causes), αλλά η ανθρώπινη παρέμβαση είναι ακόμη απαραίτητη. Ως εκ τούτου, ο εντοπισμός συστάδων συναγερμών υποστηρίζει μόνο root cause ανάλυση, αλλά δεν αποτελεί μία πλήρως αυτοματοποιημένη διαδικασία. Επιπλέον, υπενθυμίζεται ότι η root

cause ανάλυση είναι μόνο το πρώτο βήμα σε μια διαδικασία δύο σταδίων. Το δεύτερο βήμα, είναι η λήψη ενεργειών για τα ταυτοποιημένα βασικά αίτια. Στην ιδανική περίπτωση, κανείς θα αφαιρέσει τις βαθύτερες αιτίες (π.χ. η περίπτωση που περιγράφηκε ανωτέρω με την TCP / IP stack). Δυστυχώς, ορισμένες βασικές αιτίες δεν είναι υπό τον έλεγχό μας ή έχουν υψηλό κόστος για να αφαιρεθούν. Στη συνέχεια, ειδικά προσαρμοσμένοι κανόνες φιλτράρισματος (που απορρίπτουν αυτόματα τέτοιους συναγερμούς) ή τους κανόνες συσχέτισης (οι οποίοι έξυπνα ομάδοποιούν και συνοψίζουν τέτοιους συναγερμούς) μπορεί να αποτελέσουν εναλλακτική λύση. Σαφώς, αυτό το δεύτερο βήμα απαιτεί την προσοχή καθώς και την ανθρώπινη κρίση. Η νέα συμβολή αυτού του άρθρου είναι τετραπλή: Πρώτον, δείχνεται ότι μερικές αιτίες προκαλούν συνήθως την πλειοψηφία των συναγερμών σε ένα αρχείο καταγραφής συναγερμών. Δεύτερον, παρουσιάστηκε μία μέθοδος ομαδοποίησης συναγερμών που ομαδοποιεί παρόμοιους συναγερμούς, και τους συνοψίζει σε ένα ενιαίο γενικευμένο συναγερμό. Τρίτον, δείχνεται ότι η γνώση γενικευμένων συναγερμών απλουστεύει κατά πολύ την αναγνώριση των βαθύτερων αιτιών. Τέλος, δείχνεται ότι η άρση των βαθύτερων αιτιών μπορεί να μειώσει σημαντικά το μελλοντικό φορτίο συναγερμών, συνεπώς επιτρέπεται μια πιο εμπεριστατωμένη ανάλυση των υπολοίπων συναγερμών.

2.8 Εξόρυξη δεδομένων και μηχανική μάθηση για τη μείωση των ψευδών συνεγερμών σε σύστημα ανίχνευσης εισβολών

Στην ενότητα αυτή παρουσιάζονται δύο ορθογώνιες και συμπληρωματικές προσεγγίσεις για τη μείωση του αριθμού ψευδών συνεγερμών στον τομέα της ανίχνευσης εισβολών με τη χρήση μετεπεξεργασίας συνεγερμών. Η βασική ιδέα είναι να χρησιμοποιηθούν τα υφιστάμενα IDSs σαν πηγή συνεγερμών και στη συνέχεια να εφαρμοστεί είτε off-line (χρησιμοποιώντας την εξόρυξη δεδομένων) ή on-line (χρησιμοποιώντας μηχανική μάθηση) να ευαισθητοποιήσουν μετεπεξεργασία συνεγερμών για τη μείωση του αριθμού των λανθασμένων ενδείξεων. Επιπλέον, λόγω του συμπληρωματικού χαρακτήρα των, δύο προσεγγίσεων δύναται επίσης να χρησιμοποιηθούν μαζί. Η πρώτη προσέγγιση για την αντιμετώπιση του προβλήματος των λανθασμένων ενδείξεων είναι η χρήση της εξόρυξης δεδομένων σε παρελθόντα αρχεία καταγραφής συναγερμών με offline επεξεργασία για να ανακαλυφθούν κοινοί λόγοι για τους μεγάλους αριθμούς των σημάτων [32,31]. Πίσω από αυτή τη μέθοδο βρίσκεται η ιδέα ότι για κάθε παρατηρούμενο συναγερμό υπάρχει μία βασική αιτία, η οποία είναι ο λόγος για την παραγωγή του. Η

υπόθεση, η οποία επαληθεύτηκε από τη δημοσίευση [19], και καθιστά την προσέγγιση αυτή ενδιαφέρουσα είναι 1) ότι οι μεγάλες ομάδες συνεγερμών έχουν μια κοινή αιτία, 2) ότι μερικά από αυτά τα βαθύτερα αίτια ευθύνονται για ένα μεγάλο όγκο σημάτων, και 3) ότι αυτές οι αιτίες είναι (σχετικά) σταθερές μέσα στο χρόνο. Αυτό σημαίνει ότι η ανακάλυψη και η αφαίρεση των αιτίων που δε σχετίζονται με κακόβουλες ενέργειες μπορεί με ασφάλεια ναοδηγήσει σε σημαντική μείωση του αριθμού των συνεγερμών που ο αναλυτής πρέπει να χειριστεί. Ο στόχος της CLARAty (CLustering Alerts for Root Cause Analysis) είναι ως εκ τούτου η αποτελεσματική ανίχνευση μεγάλων συστάδων συνεγερμών και η περιγραφή τους με ένα γενικευμένο τρόπο, προκειμένου να οριστούν τα κοινά χαρακτηριστικά μεταξύ διαφορετικών συνεγερμών ρητά και με σαφή τρόπο για έναν αναλυτή. Ιδανικά, οι γενικευμένες αυτές περιγραφές αντιστοιχούν στα βασικά αίτια και θα είναι συνεπώς γνωστές περιπτώσεις για τους αναλυτές ασφαλείας IDS. Γενικά, οι συνεγερμοί περιγράφονται ως πλειάδες για τη μέτρηση της γνώρισματα με τιμές ζεύγη. Οι συνεγερμοί είναι δυνατόν να έχουν πολλά διαφορετικά χαρακτηριστικά ανάλογα με τον τύπο του, αλλά στην πράξη, είναι ορισμένα βασικά χαρακτηριστικά που αποτελούν πάντα μέρος του συναγερμού, π.χ., μια χρονική σήμανση, διευθύνσεις αφετηρίας και προορισμού, καθώς και μια περιγραφή ή ο τύπος του συνεγερμού. Για να καταστεί δυνατή η ομαδοποίηση των συνεγερμών, ένα μέτρο για τη μέτρηση της ομοιότητας των συνεγερμών απαιτείται. Για να προσδιοριστεί το συγκεκριμένο μέτρο χρησιμοποιούμε τη γνώση μας για το συγκεκριμένο περιβάλλον και τα γενικά χαρακτηριστικά του. Αυτό το γνωστικό υπόβαθρο αναπαρίσταται μέσω γενικευμένων ιεραρχιών από τα σημαντικά χαρακτηριστικάτων συνεγερμών. Για παράδειγμα, η τοπολογία του δικτύου ενός συγκεκριμένου περιβάλλοντος μπορούν να αποτυπωθούν σε μια ιεραρχία από περιγραφές των διευθύνσεων IP. Άλλα χαρακτηριστικά θα έχουν διαφορετικές γενικευμένες ιεραρχίες, ανάλογα με τον τύπο . Για παράδειγμα, οι θύρες προέλευσης και προορισμού συνδέσεων IP μπορούν να γενικευτούν σε προνομιακές (1 - 1024) και μη (1.025 έως 65.535). Και ένα χαρακτηριστικό timestamp θα μπορούσε να γενικευθεί σε εργάσιμη ημέρα και αργία, ή επίσης σε ώρες γραφείου και μη. Γενικευμένες ιεραρχίες όπως οι παραπάνω είναι στατικές, αλλά δυναμικές ιεραρχίες είναι επίσης δυνατό να οριστούν, για παράδειγμα συχνά επαναλβανόμενες συμβολοσειρές χαρακτηριστικών ελεύθερης μορφής, που μπορεί να παράγονται κατά τη διάρκεια εξόρυξης δεδομένων. Αν και οι γενικευμένες ιεραρχίες που περιγράφονται εδώ παρουσιάζονται ως δέντρα, το CLARAty μπορεί να επεκταθεί για να επιτρέψει τη χρήση κατευθυνόμενων ακυκλικών γράφων (DAG) [31]. Πολλές διαφορετικές τεχνικές εξόρυξης δεδομένων υπάρχουν για ανάλυση συστάδων και η καταλληλότητα των διαφορετικών μεθόδων εξαρτάται σε μεγάλο βαθμό από τον τομέα της εφαρμογής και τις ιδιότητές του. Για το πρόβλημα της ομαδοποίησης συναγερμών, όπου αναζητούνται ανθρωπίνως κατανοητές περιγραφές από συστάδες συναγερμών σε αντιστοιχία με τις αιτίες που τους προκαλούν, η μέθοδος της επαγωγή προσανατολισμένη στα χαρακτηριστικά (attribute-oriented induction

AOI) [34,33] με επεκτάσεις για τη συγκεκριμένη εφαρμογή είναι η πιο αρμόζουσα. Ο τροποποιημένος AOI αλγόριθμος, όπως περιγράφεται στο [32] χρησιμοποιεί τις ιεραρχίες γενίκευσης που περιγράφουν το γνωστικό υπόβαθρο για να συνδυαστούν συνεγερμοί σε γενικευμένους συνεγερμούς με επαναληπτικό τρόπο. Αυτοί οι γενικευμένοι συνεγερμοί περιέχουν τουλάχιστον εν μέρει γενικευμένα χαρακτηριστικά, δηλαδή, χαρακτηριστικά που έχουν γενικευτεί μέσω των παραπάνω ιεραρχιών πέρα από το χαμηλότερο επίπεδο. Θεωρητικά, ο τροποποιημένος AOI αλγόριθμος λειτουργεί ως εξής: Θεωρώντας ένα μεγάλο σύνολο συνεγερμών με χαρακτηριστικά A1 έως An μια γενικευμένη ιεραρχία Gi για κάθε χαρακτηριστικό των συνεγερμών και μια παράμετρο Nmin, η διαδικασία της ομαδοποίησης των συνεγερμών για τη δημιουργία ενόςγενικευμένου συναγερμού αποτελείται από τα ακόλουθα βήματα:

1. Επιλογή ενός χαρακτηριστικού Ai προς γενίκευση (ευριστικά).
2. Για όλους τους συνεγερμούς: αντικαθίσταται η τιμή του χαρακτηριστικού Ai τη γονική αξία της αντίστοιχης στην ιεραρχία γενίκευσης για την Ai.
3. Ομαδοποίηση όλων των συνεγερμών που έχουν γίνει ταυτόσημοι μετά από αυτή την γενίκευση και διατήρηση μια σχετικής αρίθμησης των αντίστοιχων πρωτότυπων συνεγερμών.
4. Επανάληψη των ανωτέρω βημάτων έως ότου μία από τις γενικευμένες ειδοποιήσεις να έχει πλήθος μεγαλύτερο από Nmin. Η παράμετρος ελέγχου Nmin πρέπει να προσδιοριστεί από το χρήστη. Ως αποτέλεσμα του παραπάνω αλγορίθμου έχουμε έναν γενικευμένο συνεγερμό που καλύπτει τουλάχιστον Nmin αρχικούς συνεγερμούς, η τιμή της συγκεκριμένης παράμετρου πρέπει να επιλέγεται προσεκτικά: αν επιλεγεί πολύ μεγάλη τιμή, διαφορετικά αίτια συγχωνεύονται και, επομένως, το αποτέλεσμα θα είναι πολύ γενικευμένο. Εάν έχει επιλεγεί πολύ μικρή τιμή, μία αιτία μπορεί να αντιπροσωπεύεται από πολλούς γενικευμένους συνεγερμούς. Αυτός ο αλγόριθμος θα κατασκευάσει ένα δ γενικευμένο συνεγερμό που αντιπροσωπεύει nes ένα σύμπλεγμα από τουλάχιστον Nmin συνεγερμούς. Επαναλήψεις του παραπάνω αλγορίθμου θα αποφέρει πολλαπλούς γενικευμένους συνεγερμούς, περιγράφοντας ένα μεγάλο μέρος του αρχικού συνόλου των συνεγερμών με συμπαγή τρόπο. Όπως είναι φανερό, αυτή η προσέγγιση εστιάζει στον εντοπισμό βασικών αιτιών για μεγάλες ομάδες συνεγερμών, που συνήθως αντιστοιχούν σε προβλήματα στην υποδομής του δικτύου που οδηγούν σε πολλούς ψευδείς συνεγερμούς (με την πιθανή εξαίρεση της μεγάλης κλίμακας αυτοματοποιημένων επιθέσεων). Δεν αναζητούνται επιθέσεις στα αρχεία καταγραφής συνεγερμών, αλλά ο στόχος είναι να μειωθεί ο θόρυβος από τις από τους ακατέργαστους συνεγερμούς για να γίνει ευκολότερο να εντοπιστούν οι πραγματικές επιθέσεις στη μετέπειτα ανάλυση.

Η δεύτερη προσέγγιση αντιμετωπίζει το πρόβλημα των ψευδών συνεγερμών στον τομέα

της ανίχνευσης εισβολών με την κατασκευή ενός συνεγερμού ταξινομητή που αναφέρει τους αληθείς από τους ψευδείς συνεγερμούς. Ως ταξινομητής συναγερμός ορίζεται ως η η επισύναψη μιας ετικέτας από ένα σύνολο ετικετών που έχουν οριστεί από το χρήστη. Στην απλούστερη περίπτωση, οι συνεγερμοί είναι ταξινόμημένοι σε ψευδείς και θετικούς, αλλά η ταξινόμηση μπορεί να επεκταθεί και για να αναδείξει την κατηγορία μίας επίθεσης, τις αιτίες ενός ψευδούς συνεγερμού ή οτιδήποτε άλλο. Οι συνεγερμοί ταξινομούνται από ένα συνεγερμό ταξινομητή. Ταξινομητές συνεγερμοί μπορεί να κατασκευαστούν αυτόματα με τη χρήση της τεχνικής μηχανικής μάθησης ή μπορούν να κατασκευαστούν με μη αυτόματο τρόπο. Το Adaptive Learner for Alert Classication (ALAC) που περιγράφεται στο [36] κάνει χρήση της προηγούμενης προσέγγισης. Το σημαντικότερο είναι πως, το ALAC εκπαιδεύει τους ταξινομητές συνεγερμών ώστε να διαθέτουν μία συγκεκριμένη λογική ταξινόμησης και ο αναλυτής να επιθεωρεί και να επιβεβαιώνει την ορθότητα τους. Το ALAC ταξινομεί τους συνεγερμούς σε πραγματικούς και ψευδείς και αναφέρει τις συγκεκριμένες κατηγοριοποιήσεις στον αναλυτή του συστήματος εισβολών. Σε αντίθεση με τα συνηθισμένα συστήματα ανίχνευσης εισβολών το ALAC χρησιμοποιεί το input του χρήστη-αναλυτή που ταξινομεί τους συνεγερμούς για να δημιουργήσει ετικέτες συνεγερμούς. Οι συγκεκριμένοι ετικέτες συνεγερμοί χρησιμοποιούνται από το σύστημα για να παραχθούν εκπαιδευτικά παραδείγματα που χρησιμοποιούνται από τις τεχνικές εκμάθησης για να κατασκευασθεί και να ανανεώνεται ο ταξινομητής. Κατόπιν ο ταξινομητής χρησιμοποιείται για να ταξινομήσει νέους συνεγερμούς. Ο αναλυτής μπορεί να επιθεωρήσει τον ταξινομητή οποιαδήποτε στιγμή. Πιο συγκεκριμένα το ALAC έχει δύο τρόπους λειτουργίας. Κατά τον πρώτο τρόπο που αποκαλείται recommended mode το σύστημα δεν επεξεργάζεται κανένα συνεγερμό αλλά μόνο προβλέπει τις ετικέτες και προωθεί τους συνεγερμούς στον αναλυτή μαζί με μία σύσταση. Αυτό σημαίνει ότι ο αναλυτής πρέπει να επιθεωρήσει όλους τους συνεγερμούς όπως προηγουμένως αλλά μπορεί να χρησιμοποιηθεί και η σύσταση από το σύστημα. Ο δεύτερος τρόπος λειτουργίας αποκαλείται agent mode και το σύστημα αξιολογεί την εμπιστοσύνη της ταξινόμηση και εάν είναι ανώτερη μίας προκαθορισμένης τιμής το σύστημα επεξεργάζεται τους συνεγερμούς αυτόματα. Στην περίπτωση ψευδών συνεγερμών αυτοί θα αγνοούνταν. Στην περίπτωση των αληθινών συνεγερμών το σύστημα είτε το αναφέρει στον αναλυτή είτε πραγματοποιεί μία ενέργεια. Σε αντίθεση με την προσέγγιση εξόρυξη δεδομένων που χρησιμοποιείται από το CLARAty, η προσέγγιση αυτή εξαρτάται από την ικανότητα του αναλυτή να χαρακτηρίσει τους συνεγερμούς σωστά. Αυτή η παραδοχή είναι αιτιολογείται επειδή ο αναλυτής πρέπει να είναι ειδικός στον τομέα της ανίχνευσης εισβολών για να εκτελέσει ανάλυση συμβάντων και να δρομολογήσει τις κατάλληλες απαντήσεις. Αυτό εγείρει το ερώτημα γιατί οι αναλυτές δεν ορίζουν μία συγκεκριμένη λογική ταξινόμησης ή δεν ανανεώνουν αυτή τη λογική πιο συχνά. Μια εξήγηση αυτών των ζητημάτων μπορεί να βασίζεται στα ακόλουθα στοιχεία:

Οι αναλυτές είναι δύσκολο να πραγματοποιούν γενικεύσεις, δηλαδή, να διατυπώνουν

γενικότερους κανόνες, με βάση μία επιμέρους συγκεκριμένη λογική ταξινόμησης. Για παράδειγμα, ο αναλυτήςθα μπορούσε να χαρακτηρίσει κάποιους μεμονωμένους συνεγερμούς ως ψευδείς , αλλά δενείναι σε θέση να γράψει ένα γενικό κανόνα που χαρακτηρίζει το σύνολο των συνεγερμών αυτών.Τα περιβάλλοντα είναι δυναμικά. Στο πραγματικά περιβάλλοντα, τα χαρακτηριστικά των συνεγερμών μεταβάλλονται, π.χ., δι αφορετικοί συνεγερμοί εμφανίζονται εάν νέοι ηλεκτρονικοί υπολογιστές και υπηρεσίες εγκαθίστανται ή ορισμένα worms ή επιθέσεις πραγματοποιούνται πιο συχνά ή αραιά. Η ταξινόμηση ενός συνεγερμού μπορεί επίσης να αλλάξει. Ως εκ τούτου, οι κανόνες πρέπει να συντηρούνται και να διαχειρίζονται. Αυτή η διαδικασία είναι απαιτεί εργασία και είναι επιρρεπής σε λάθη.

Όπως αναφέρθηκε στην εισαγωγή, το CLARAty και το ALAC είναι δύο συμπληρωματικές προσεγγίσεις και μπορούν να χρησιμοποιηθούν μαζί σε ένα σύστημα φιλτραρίσματος και ταξινόμησης με δύο στάδια. Ένα τέτοιο σύστημα θα χρησιμοποιεί το CLARAty στο πρώτο στάδιο έτσι ώστε περιοδικά να εξορύσονται πρωτογενείς συνεγερμοί να διερευνούνται τα γεννεσιουργά αιτιά τους και να προβαίνουμε σε είτε αφαίρεση τους ή εγκατάσταση φίλτρων συναγερμού. Η έξοδος από το πρώτο στάδιο θα τροφοδοτούσε το ALAC. Το πλεονέκτημα αυτής της προσέγγισης είναι ότι το CLARAty αφαιρεί τους πιο διαδεδομένους και μη ενδιαφέροντες ψευδείς συνεγερμούς γεγονός που βελτιώνει την κατανομή της ταξινόμησης για τους συνεγερμούς που εισέρχονται στο δεύτερο στάδιο. Επιπλέον, το ALAC λαμβάνει λιγότερους συνεγερμούς για επεξεργασία, που είναι σημαντικό από άποψη πόρων.

2.9 Μείωση ψευδών συνεγερμών σε συστήματα ανίχνευσης εισβολών

Στη συγκεκριμένη παράγραφο παρουσιάζεται μία μέθοδος μετεπεξεργασίας συνεγερμών [14]. Πρόκειται για ένα φίλτρο το οποίο μπορεί να κατασκευαστεί εάν βασιστεί κανείς σε συγκεκριμένα χαρακτηριστικά που παρουσιάζονται στους αληθείς και στους ψευδείς συνεγερμούς. Το προτεινόμενο σύστημα φιλτραρίσματος βασίζεται στις ακόλουθες βασικές παρατηρήσεις για τους αληθείς ή ψευδείς συνεγερμούς:

• Οι πραγματικοί συνεγερμοί παρατηρούνται συνήθως σε batches συνεγερμών, που παρουσιάζουν ομοιότητες όσον αφορά τις διευθύνσεις IP προέλευσής και προορισμού. Οι πραγματικοί συνεγερμοί παρατηρούνται σε υψηλότερη συχνότητα σε ότι αφορά την υπογραφή της επίθεσης σε σύγκριση με τη μέση συχνότητα της αντίστοιχης υπογραφής.
• Για ένα συγκεκριμένο δίκτυο, κάθε υπογραφή έχει μια συγκεκριμένη πιθανότητα

παραγωγής ψευδών συνεγερμών, που εξαρτάται από την τοπολογία του δικτύου.

Όταν μια πραγματική επίθεσης συμβαίνει, συνήθως παράγεται μια έκρηξη συνεγερμών, που είναι κατά κάποιον τρόπο σχετιζόμενοι με την επίθεση. Η σχέση αυτή μπορεί να αναφέρεται ρητά από τις διευθύνσεις IP προέλευσης και προορισμού των συνεγερμών. Από την άλλη πλευρά, οι ψευδείς συνεγερμοίείναι ψευδείς συνεγερμοί που κατανέμονται ομοιόμορφα σε όλο το τεράστιο ποσό των συνεγερμών που παράγονται από τα IDS. Ως εκ τούτου, η απόφαση για το αν ένας συναγερμός είναι ψευδής ή αληθής μπορεί να βασίζεται στον αριθμό συνεγερμών που συμβαίνουν σε ένα χρονικό παράθυρο γύρω από ένα συγκεκριμένο συναγερμό και ότι έχει κοινές τιμές στα πεδία διευθύνσεων IP προορισμού και προέλευσης με το συγκεκριμένο συνεγερμό. Μια στατιστική επισκόπηση των δεδομένων που παρέχονται με την εκτέλεση του Snort κατά τη 2η εβδομάδα του συνόλου δεδομένων DARPA δείχνει σαφώς ότι η κατανομή του αριθμού των γειτονικών συνεγερμών ποικίλλει σημαντικά στους ψευδείς και στους αληθείς συνεγερμούς.
Ενώ οι περισσότεροι ψευδείς συνεγερμοί έχουν έως και 40 γειτονικούς σχετιζόμενους συνεγερμούς οι αληθείς συνεγερμοί έχουν περισσότερους από 100 γειτονικούς σχετιζόμενους συνεγερμούς και πολλοί από αυτούς έχουν χιλιάδες.
Για ένα συγκεκριμένο συνεγερμό με χρονοσήμανση t0 και υπογραφή S0, η σχετιζόμενη με τη συχνότητα υπογραφή είναι η συχνότητα με την οποία οι συνεγερμοί με υπογραφή S0 εμφανίζονται σε ένα χρονικό παράθυρο t0. Είναι πιο πιθανόν για ένα συνεγερμό να είναι μια πραγματική επίθεση εάν εμφανίζεται σε υψηλότερη συχνότητα σε σύγκριση με τη μέση συχνότητα των συνεγερμών που περιγράφουν την ίδια επίθεση (έχουν την ίδια υπογραφή). Με άλλα λόγια, όταν πραγματοποιείται μια επίθεση αυξάνεται η παραπάνω συχνότητα υπογραφής. Σε αυτό το σημείο θα πρέπει να σημειωθεί ότι υπάρχουν περιπτώσεις συνεγερμών που αποκλίνουν από τον παραπάνω κανόνα. Πραγματικές επιθέσεις συνήθως συνοδεύονται από συνεγερμούς με την ίδια υπογραφή οι οποίοι απέχουν κοντά στο χρόνο (η χρονική διαφορά είναι μικρότερη από δύο δευτερόλεπτα). Μια άλλη σημαντική παρατήρηση σχετικά με την εμφάνιση ψευδών συνεγερμών είναι ότι συνήθως προέρχονται από τις ίδιες αιτίες που σχετίζονται με την τοπολογία του δικτύου, hosts που δεν έχουν ρυθμιστεί σωστά ή διάφορες υπηρεσίες και εργασίες που εμφανίζονται στο δίκτυο. Όλες αυτές οι αιτίες είναι σταθερά και επαναλαμβανόμενα πρότυπα που παράγουν ψευδείς συνεγερμούς. Αν μία περίοδος χωρίς εκδηλούμενες επιθέσεις είναι διαθέσιμη, αυτά τα πρότυπα των συνήθων ψευδώς συνεγερμών δύναται να εξαχθούν και στη συνέχεια να χρησιμοποιηθούν για να αναγνωριστούν και να αποκλειστούν οι ψευδείς συνεγερμοί.
Το προτεινόμενο φίλτρο αποτελείται από τρεις συνιστώσες, και συγκεκριμένα τη συνιστώσα Neighboring Related Alerts (NRA) τη συνιστώσα High Alert Frequency (HAF) και τη συνιστώσα Usual False Positives (UFP). Το σύνολο των συναγερμών που παράγεται

από ένα IDS είναι η είσοδος σε κάθε ένα από τα τρία αυτά στοιχεία. Κάθε στοιχείο παράγει ένα αποτέλεσμα για κάθε συναγερμό, που αντιπροσωπεύει την πιθανότητα ότι ο συγκεκριμένος συνεγερμός είναι πραγματική επίθεση. Αυτά τα τρία μεγέθη συσχετίζονται για να παραχθεί η τελική απόφαση για το εάν ο συγκεκριμένος συνεγερμός είναι πραγματική επίθεση ή όχι.

3 Ανάλυση και Σχεδίαση

Στο παρόν κεφάλαιο παρουσιάζονται τα σχετικά με τη σχεδίαση του συστήματος ζητήματα. Ο τρόπος παρουσίασης αυτών γίνεται μέσω τυποποιημένων εγγράφων τεχνολογίας λογισμικού. Συγκεκριμένα τα έγγραφα αυτά είναι βασισμένα στα αντίστοιχα έγγραφα Προσδιορισμού Απαιτήσεων από το Λογισμικό και Αρχιτεκτονικής της διαδικασίας ανάπτυξης λογισμικού Rational Unified Process (RUP). Στην πρώτη ενότητα περιγράφονται οι λειτουργίες του συστήματος μέσω του εγγράφου με τίτλο «Έγγραφο Προσδιορισμού Απαιτήσεων από το Λογισμικό». Στη δεύτερη ενότητα παρουσιάζεται ο βασικός και λεπτομερής σχεδιασμός του συστήματος μέσω των εγγράφων «Έγγραφο Περιγραφής της Αρχιτεκτονικής» και «Έγγραφο Περιγραφής του Λεπτομερούς Σχεδίου»

3.1 Περιγραφή Λειτουργιών-Έγγραφο Προδιαγραφών Απαιτήσεων από το Λογισμικό

Η περιγραφή των λειτουργιών του συστήματος παρουσιάζεται στην ενότητα αυτή μέσω του «Εγγράφου Προδιαγραφών Απαιτήσεων από το Λογισμικό» το οποίο ακολουθεί.

3.1.1 Έγγραφο Προδιαγραφών Απαιτήσεων από το Λογισμικό

Προδιαγραφή απαιτήσεων από το λογισμικό

Εισαγωγή

Το έγγραφο αυτό περιλαμβάνει τις προδιαγραφές του λογισμικού συστήματος "Λογισμικό μετα-επεξεργασίας Συνεγερμών". Η παρουσίαση και ανάλυση των προδιαγραφών γίνεται με βάση τις λειτουργικές απαιτήσεις του συστήματος σύμφωνα με το μοντέλο των περιπτώσεων χρήσης.

Σκοπός

Σκοπός του εγγράφου είναι η περιγραφή της συμπεριφοράς που θα εκδηλώνει το λογισμικό προς το περιβάλλον του, καθώς και η απαρίθμηση των γνωρισμάτων και των λειτουργιών του.

Εμβέλεια

Το παρόν έγγραφο παρουσιάζει τις περιπτώσεις χρήσης του συστήματος και περιλαμβάνει περαιτέρω ανάλυση αυτών με τη βοήθεια κατάλληλων διαγραμμάτων σε UML.

Ορισμοί, ακρωνύμια και συντομογραφίες

component: ο συγκεκριμένος όρος μπορεί να έχει διάφορους ορισμούς ανάλογα το πλαίσιο που χρησιμοποιείται. Στο παρόν έγγραφο χρησιποιείται κυρίως για να δηλώσει δύο έννοιες. Η μία έννοια είναι ένα αυτόνομο μέρος λογισμικού. Χρησιμοποιείται επίσης για να δηλώσει ένα component που μπορεί να χρησιμοποιηθεί για τη δημιουργία ενός φίλτρου.
φίλτρο: με αυτόν τον όρο περιγράφεται μία δομή από components την οποία σχεδιάζει ο χρήστης και αποτελεί τη λύση.
RUP: Rational Unified Process.
Alert: Συνεγερμός
Συνεγερμός: Ένδειξη που λαμβάνεται από το IDS/IPDS για ενδεχόμενη εισβολή.
IDS/IPDS: Intrusion Detection System / Intrusion Prevention Detection System.

Περίληψη

Στην ενότητα 2 δίνεται η γενική περιγραφή του μοντέλου περιπτώσεων χρήσης, δηλαδή της περιγραφής της συμπεριφοράς του λογισμικού. Στην ενότητα 3 δίνονται οι τίτλοι και οι γενικές περιγραφές των περιπτώσεων χρήσης του συστήματος και οι ειδικές απαιτήσεις αυτού.

Γενική Περιγραφή του συστήματος

Το λογισμικό που θα αναπτυχθεί ονομάζεται «Λογισμικό μετα επεξεργασίας συνεγερμών». Ο βασικός χρήστης του συστήματος είναι ο αναλυτής που είναι υπεύθυνος για την ανάλυση και αξιολόγηση των συνεγερμών που παράγονται από ένα IDS/IPDS σύστημα. Αφού έχει ολοκληρωθεί η συλλογή των συνεγερμών που έχουν παραχθεί από ένα IDS/IPDS το λογισμικό παρέχει δύο βασικές λειτουργίες:

1. Δημιουργία και εκτέλεση φίλτρου
Ο χρήστης σχεδιάζει ένα φίλτρο συνδυάζοντας έναν αριθμό από τα διαθέσιμα components που βρίσκονται εγκατεστημένα στο σύστημα. Κατόπιν πραγματοποιείται εισαγωγή στο σύστημα των συλλεγμένων συνεγερμών, εκτελείται η λογική του κάθε component και απομένει στο τέλος ένα σύνολο από συνεγερμούς το οποίο συγκρίνεται με ένα σύνολο απο εγγραφές που αποτελούν τις πραγματικές επιθέσεις. Η αποδοτικότητα του σχεδιασμένου φίλτρου είναι ανάλογη της αντιστοιχίας μεταξύ των πραγματικών επιθέσεων και εναπομείναντων συνεγερμών.

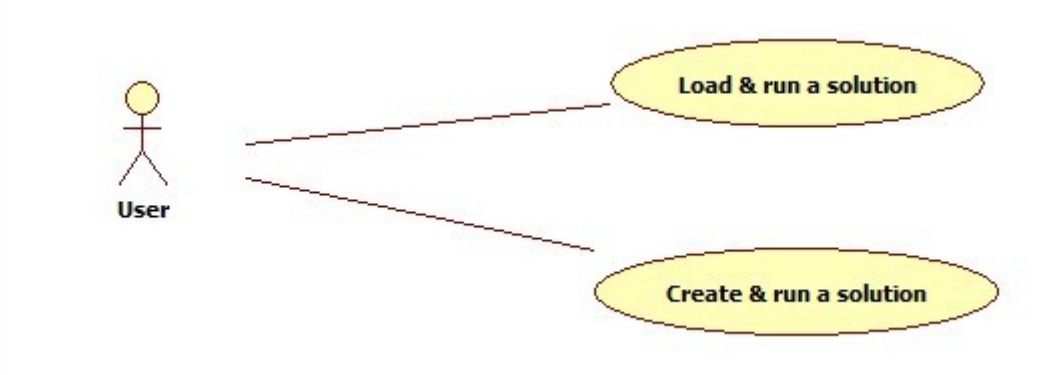
2. Εισαγωγή και εκτέλεση φίλτρου.

Η συγκεκριμένη λειτουργία παρουσιάζει ομοιότητες με την λειτουργία 1 εκτός του ότι η δομή του φίλτρου στη συγκεκριμένη περίπτωση δεν σχεδιάζεται. Η δομή του φίλτρου βρίσκεται αποθηκευμένη σε xml αρχείο το οποίο εισάγεται στο σύστημα. Κατόπιν ο χρήστης έχει τη δυνατότητα να τροποποιήσει τη δομή του φίλτρου καθώς και να το εκτελέσει.

Το μοντέλο περιπτώσεων χρήσης

Περιγραφή του μοντέλου

Το μοντέλο αποτελείται από 2 περιπτώσεις χρήσεις, οι οποίες παρουσιάζονται σε διάγραμμα περιπτώσεων χρήσης (use case diagram) που παρατίθεται στη συνέχεια. Κάθε μία από τις περιπτώσεις χρήσης αναλύεται περαιτέρω σε ξεχωριστό έγγραφο.



Όπως προκύπτει από το παραπάνω use-case διάγραμμα, το σύστημα περιλαμβάνει 2 βασικές περιπτώσεις χρήσης:

1. Σχεδιασμός και εκτέλεση φίλτρου
2. Τροποποίηση και εκτέλεση υπάρχοντος φίλτρου.

Απαιτήσεις

Υπάρχουν πολλά διαφορετικά είδη απαιτήσεων. Ένας τρόπος κατηγοριοποίησης τους περιγράφεται ως το μοντέλο FURPS + [GRA92], χρησιμοποιώντας το αρκτικόλεξο FURPS για να περιγράψει τις μεγάλες κατηγορίες των απαιτήσεων με υποκατηγορίες, όπως φαίνεται παρακάτω.

- Functionality (Λειτουργικές απαιτήσεις)
- Usability (Απαιτήσεις χρήσης)
- Reliability (Απαιτήσεις αξιοπιστίας)

- Performance (Απαιτήσεις επιδόσεων)
- Supportability (Απαιτήσεις υποστήριξης συστήματος)

Το "+" υπενθυμίζει να συμπεριληφθούν και οι κατηγορίες:

- Περιορισμοί σχεδίασης
- Περιορισμοί υλοποίησης
- Περιορισμοί διεπαφών
- Περιορισμοί υλικού

Λειτουργικές απαιτήσεις

Το σύστημα θα πρέπει να παρέχει τη δυνατότητα:

- Δημιουργίας, εκ του μηδενός, μίας δομής για το φίλτρο. Ο χρήστης θα μπορεί να επιλέγει τα μέρη που θα αποτελούν τη φίλτρο.
- Αποθήκευσης του φίλτρου.
- Εισαγωγή ήδη υπάρχοντος φίλτρου και δυνατότητα τροποποίησης αυτού.
- Εκτέλεσης του φίλτρου από κονσόλα και γραφικό περιβάλλον.
- Αποθήκευση των αποτελεσμάτων σε αρχεία.
- Εισαγωγή στο σύστημα alertset από βάση δεδομένων.
- Εισαγωγή στο σύστημα έγκυρων συνεγερμών (alerts) από xml αρχεία.

Χρηστικές Απαιτήσεις

- Εκμάθηση. Το λογισμικό δεν πρέπει να απαιτεί καμία ιδιαίτερη εκμάθηση από το χρήστη ο οποίος γνωρίζει το αντικείμενό του. Τα περιεχόμενα του αρχείου βοηθείας που θα είναι διαθέσιμα στο χρήστη πρέπει να είναι αρκετά για το σκοπό αυτό.
- Γενική αίσθηση. Η γενική αίσθηση θα είναι αυτή μιας εφαρμογής JAVA.

Απαιτήσεις αξιοπιστίας

- Οι δημιουργηθέντα από το χρήστη φίλτρα θα πρέπει να εκτελούνται με προβλέψιμο τρόπο. Τούτο σημαίνει ότι δύο διαδοχικές εκτελέσεις μίας συγκεκριμένης δομής φίλτρου με ένα σύνολο υποψήφιων συνεγερμών (alertset) και ένα σύνολο από έγκυρα alerts θα πρέπει να παράγει πάντα το ίδιο αποτέλεσμα.

Απαιτήσεις επιδόσεων

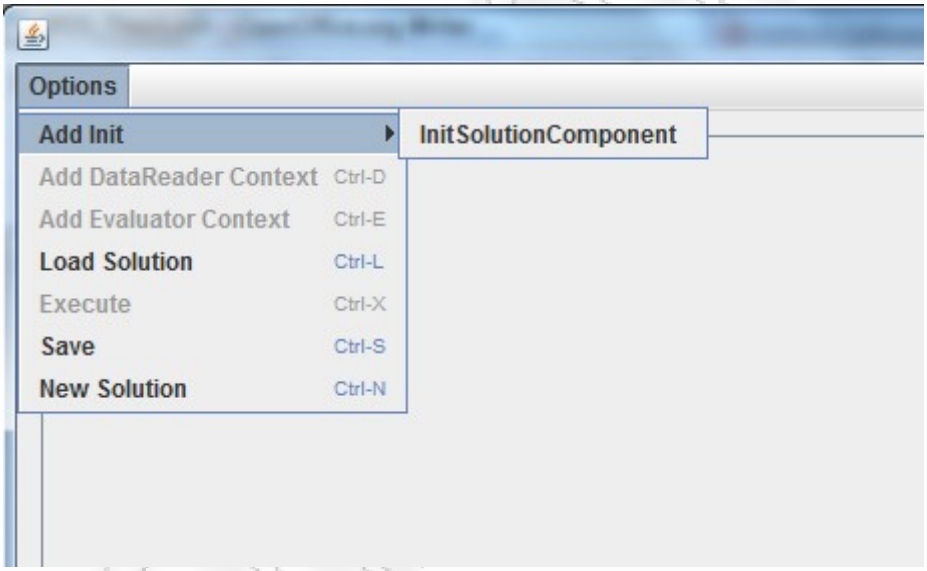
- Η εισαγωγή και η εκτέλεση μίας δομής φίλτρου δε θα πρέπει να απαιτεί μεγάλο χρονικό διάστημα για την ολοκλήρωσή της. Σε αυτό το σημείο θα πρέπει να

αναφερθεί ότι η συγκεκριμένη απαίτηση εξαρτάται σε μεγάλο βαθμό από την υλοποίηση των επιμέρους components του φίλτρου που έχει ορίσει ο χρήστης.

Απαιτήσεις υποστήριξης συστήματος

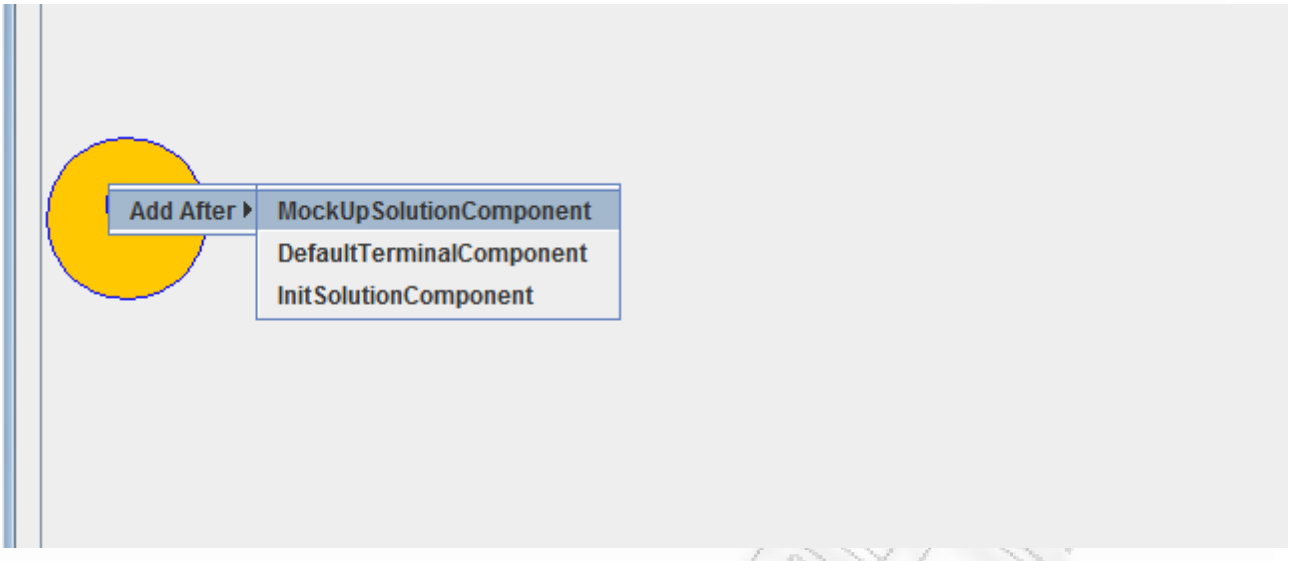
- Η διαδικασία της εγκατάστασης και της παραμετροποίησης της εφαρμογής δε θα πρέπει να είναι σύνθετη.
- Δυνατότητα απομονομένου ελέγχου των επιμέρους στοιχείων που συνθέτουν το λογισμικό. (Testability)
- Επεκτασιμότητα. Ο χρήστης θα πρέπει να μπορεί να ορίσει και να εισάγει στο σύστημα καινούργια components για τη δημιουργία σύνθετων φίλτρων.

Interfaces – Διεπαφές (Διεπαφή χρήστη)

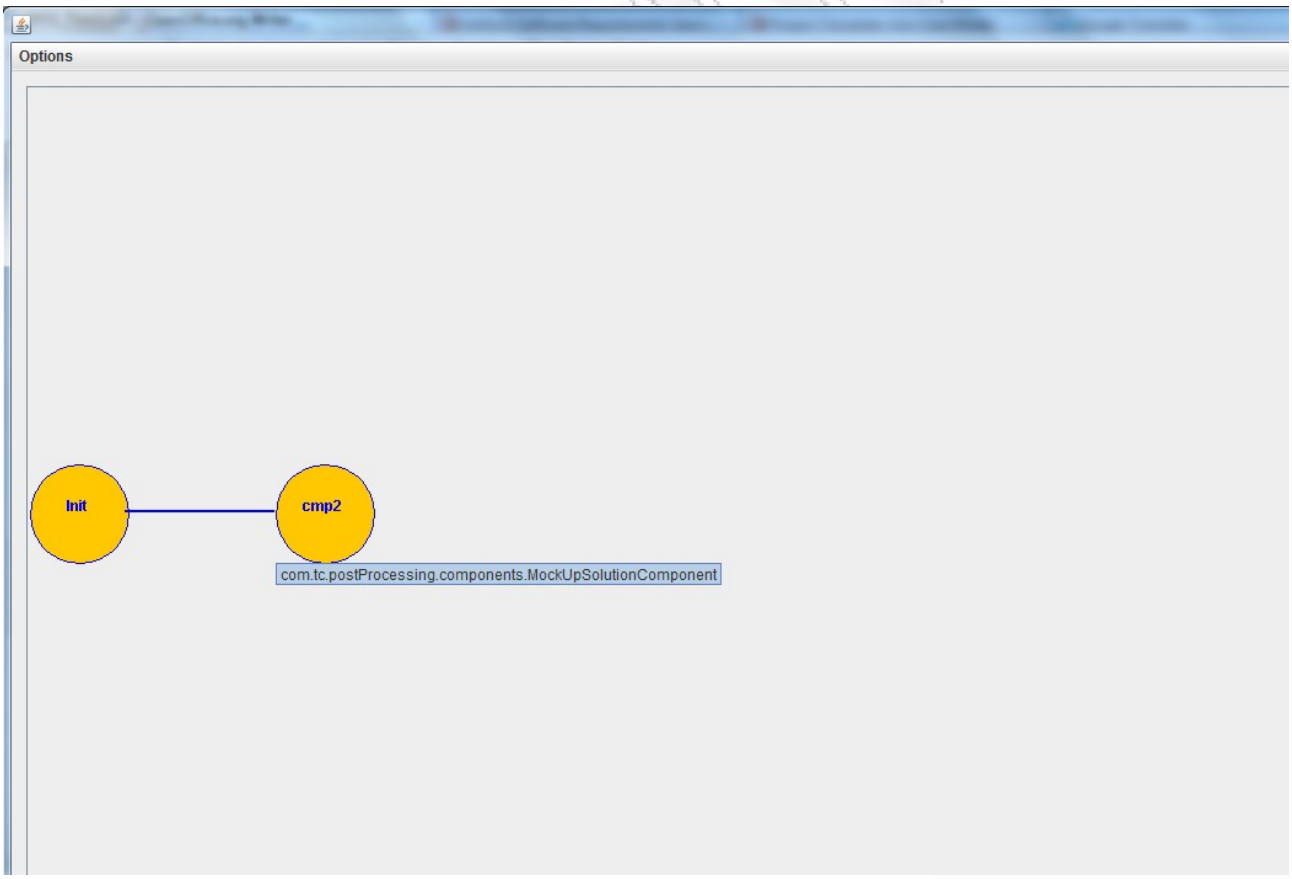


Στη συνέχεια παρατίθενται οι βασικές οθόνες επικοινωνίας με το χρήστη του συστήματος και περιγράφεται σύντομα η λειτουργία τους.
Στην παραπάνω οθόνη φαίνεται το μενού κατά την αρχικοποίηση του φίλτρου. Στο menu item "Add Init" δίνεται η δυνατότητα στο χρήστη να προσθέσει το πρώτο component του φίλτρου. Στο μενού θα εμφανιστούν όλα τα διαθέσιμα components του συγκεκριμένου τύπου. Πρόκειται για στοιχεία τα οποία επιτρέπεται να οριστούν ως αρχικά σε ένα φίλτρο.

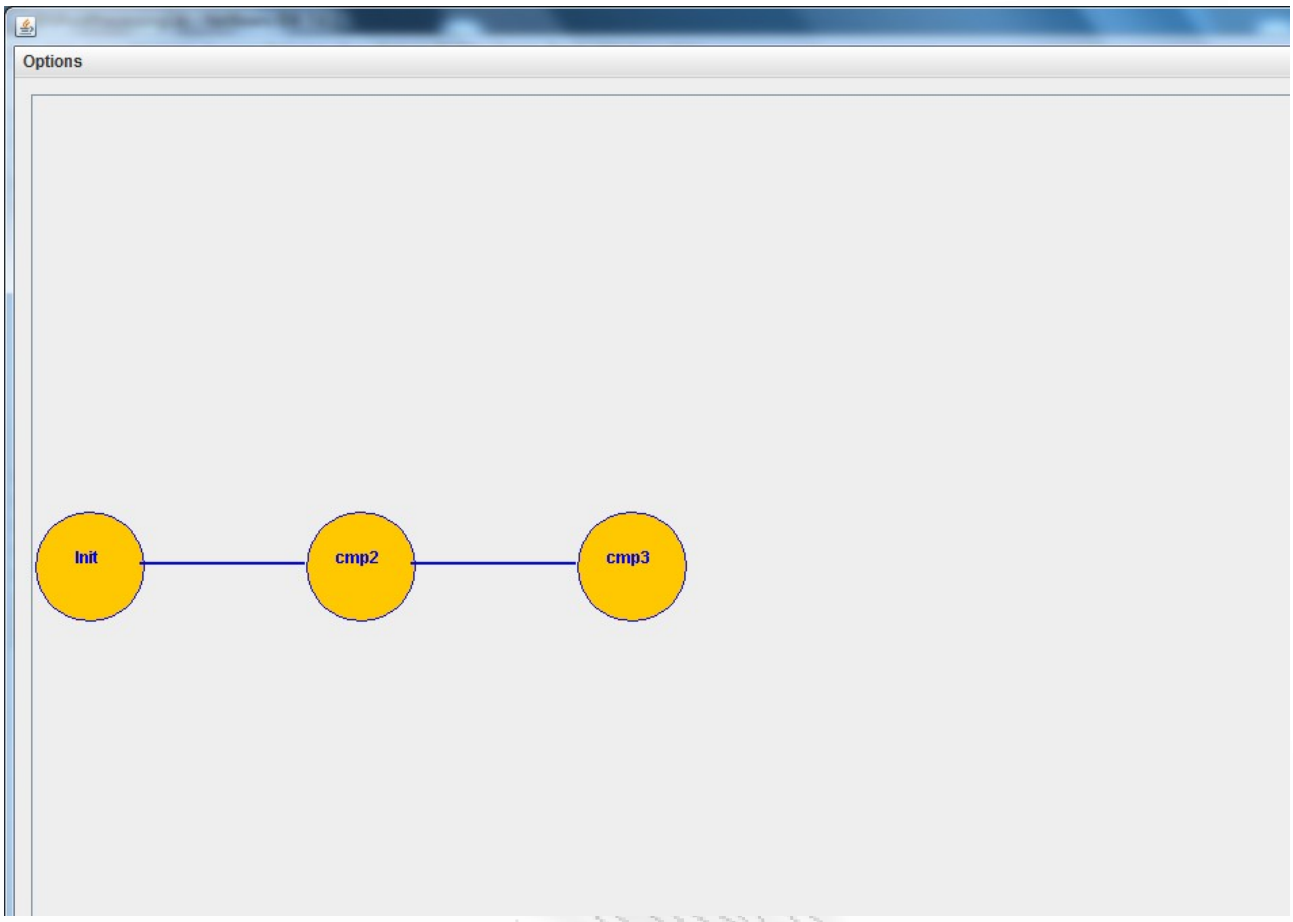
Στη συγκεκριμένη οθόνη έχει προστεθεί ένα αρχικό component.

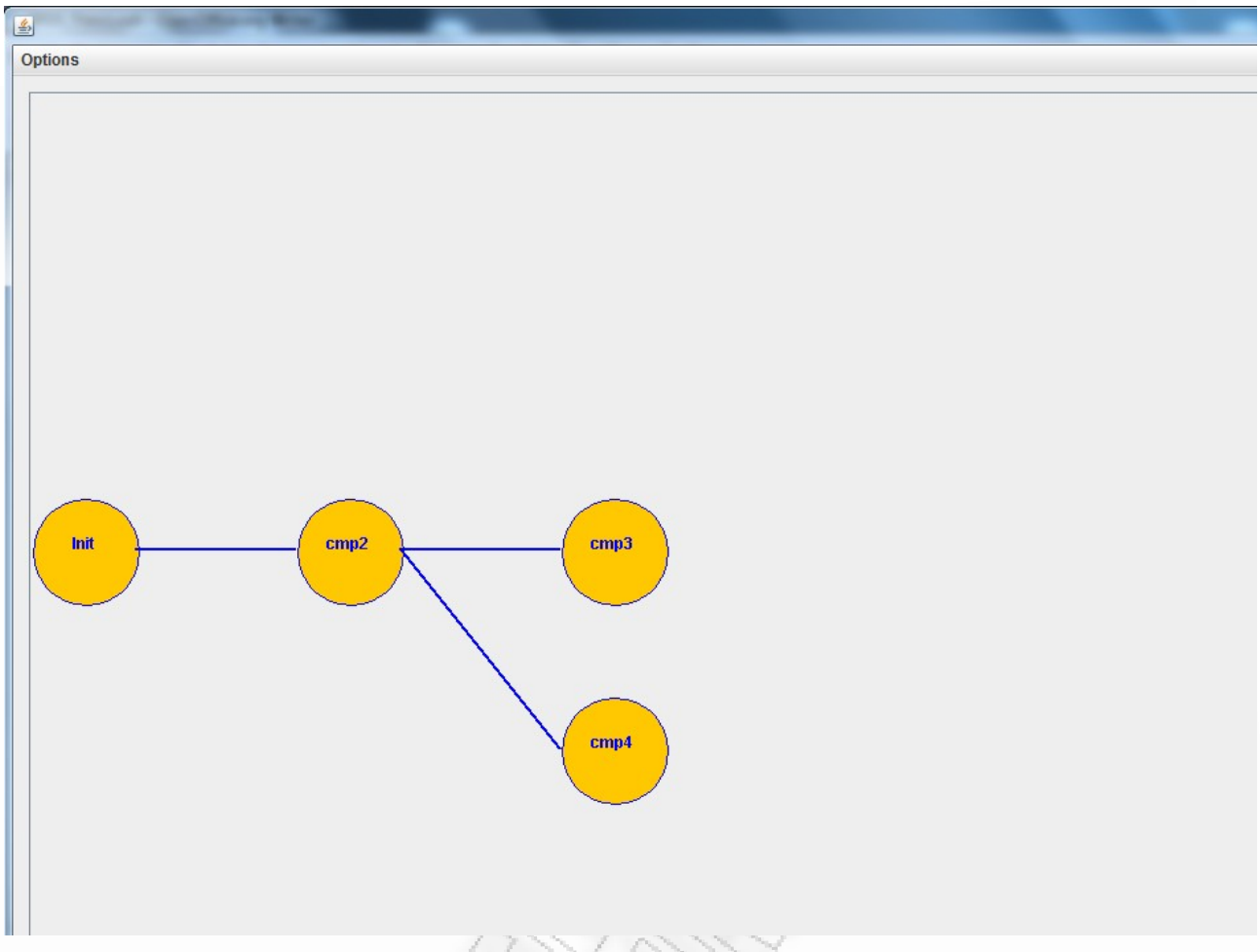


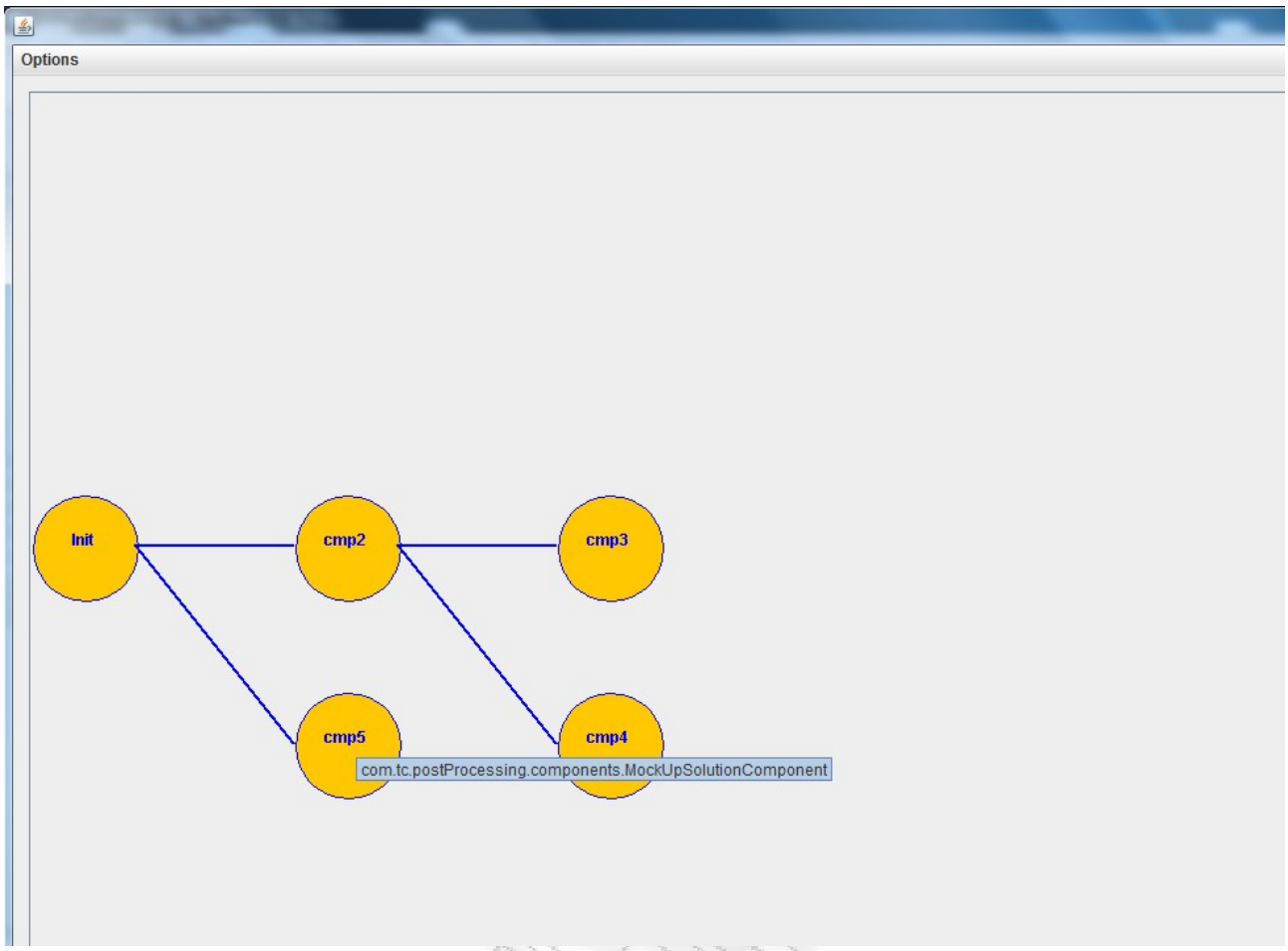
Σε αυτή την οθόνη φαίνονται τα components που μπορούν να τοποθετηθούν μετά το αρχικό.

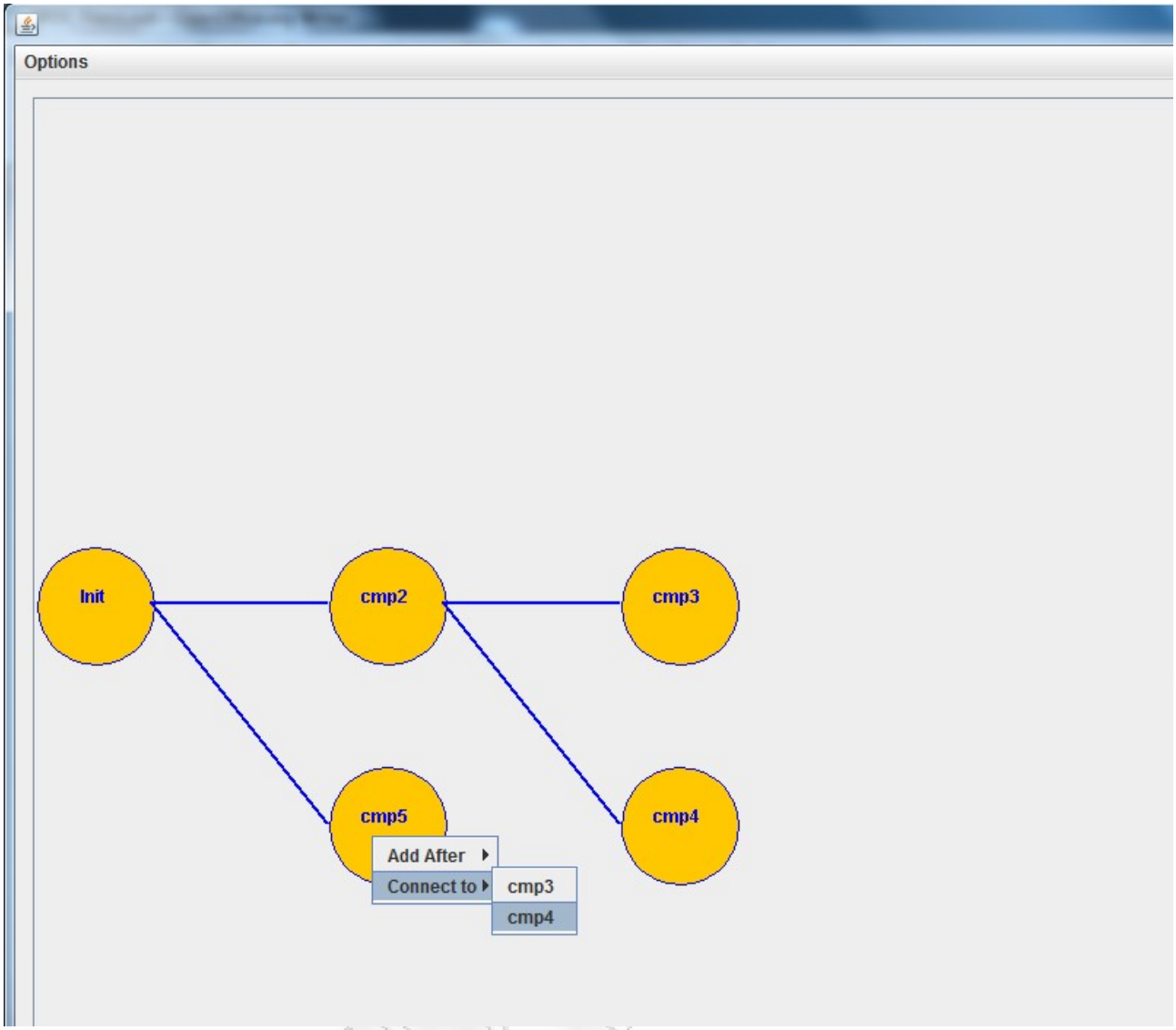


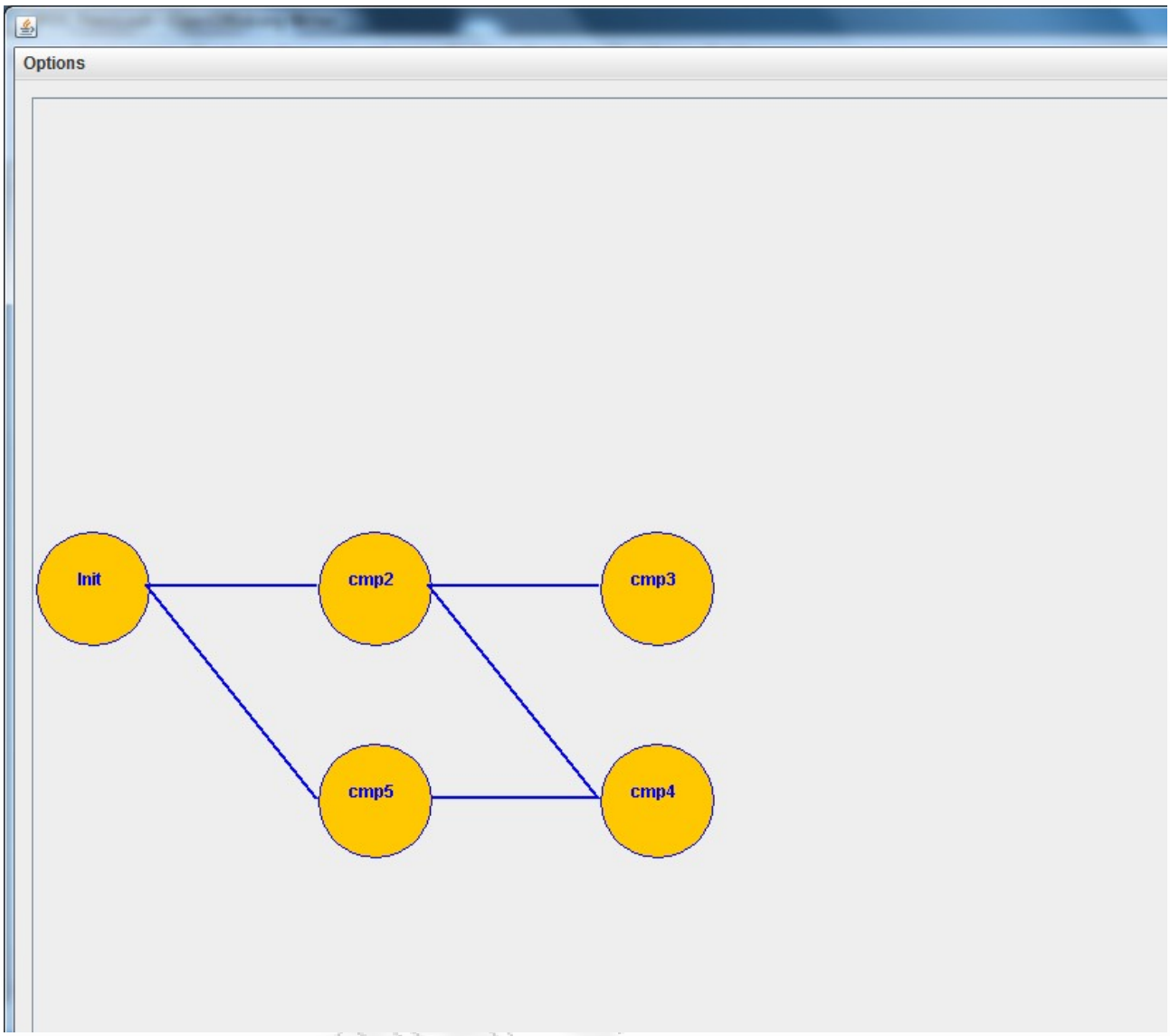
Στις οθόνες που ακολουθούν φάινεται η δημιουργία ενός φίλτρου.

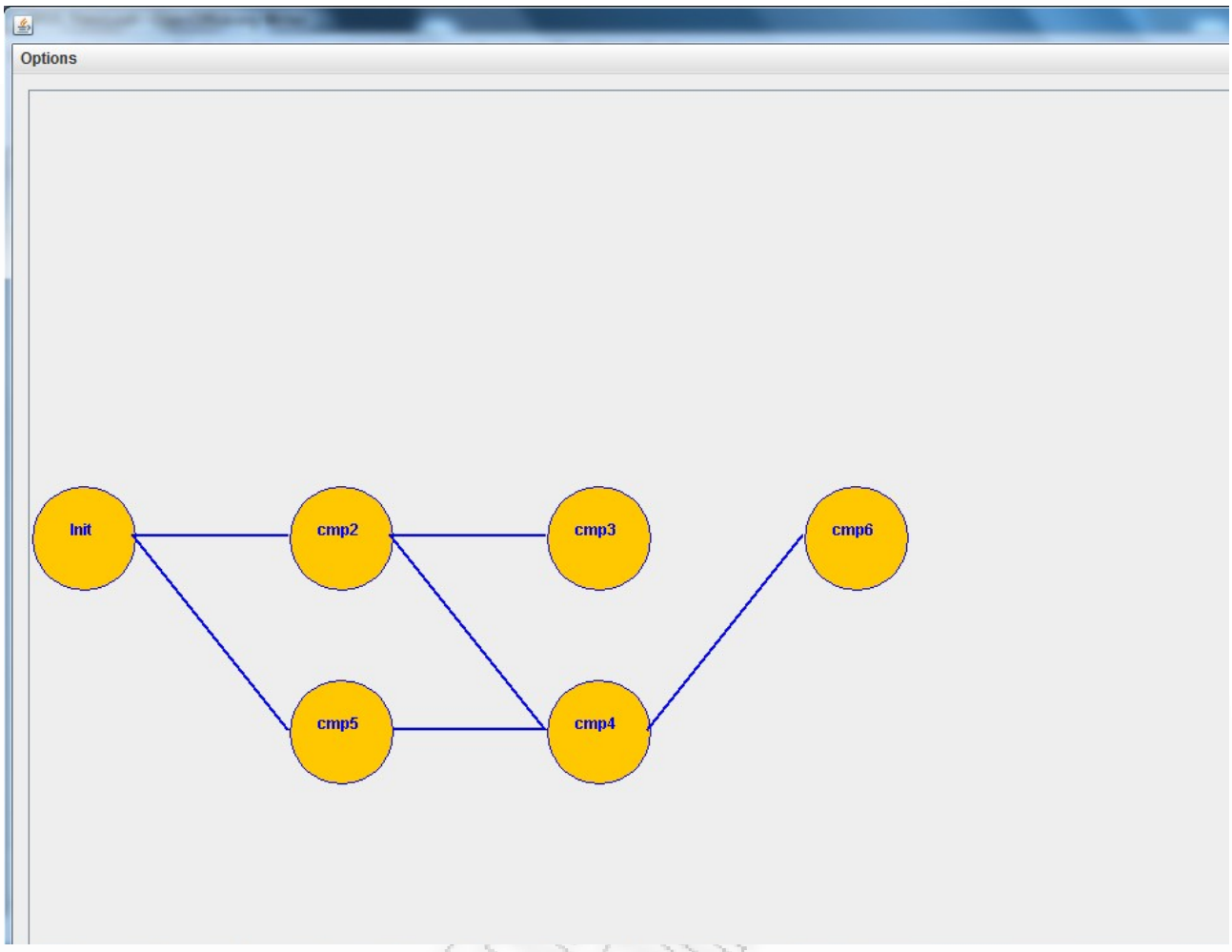

Options
Init
cmp2
cmp3

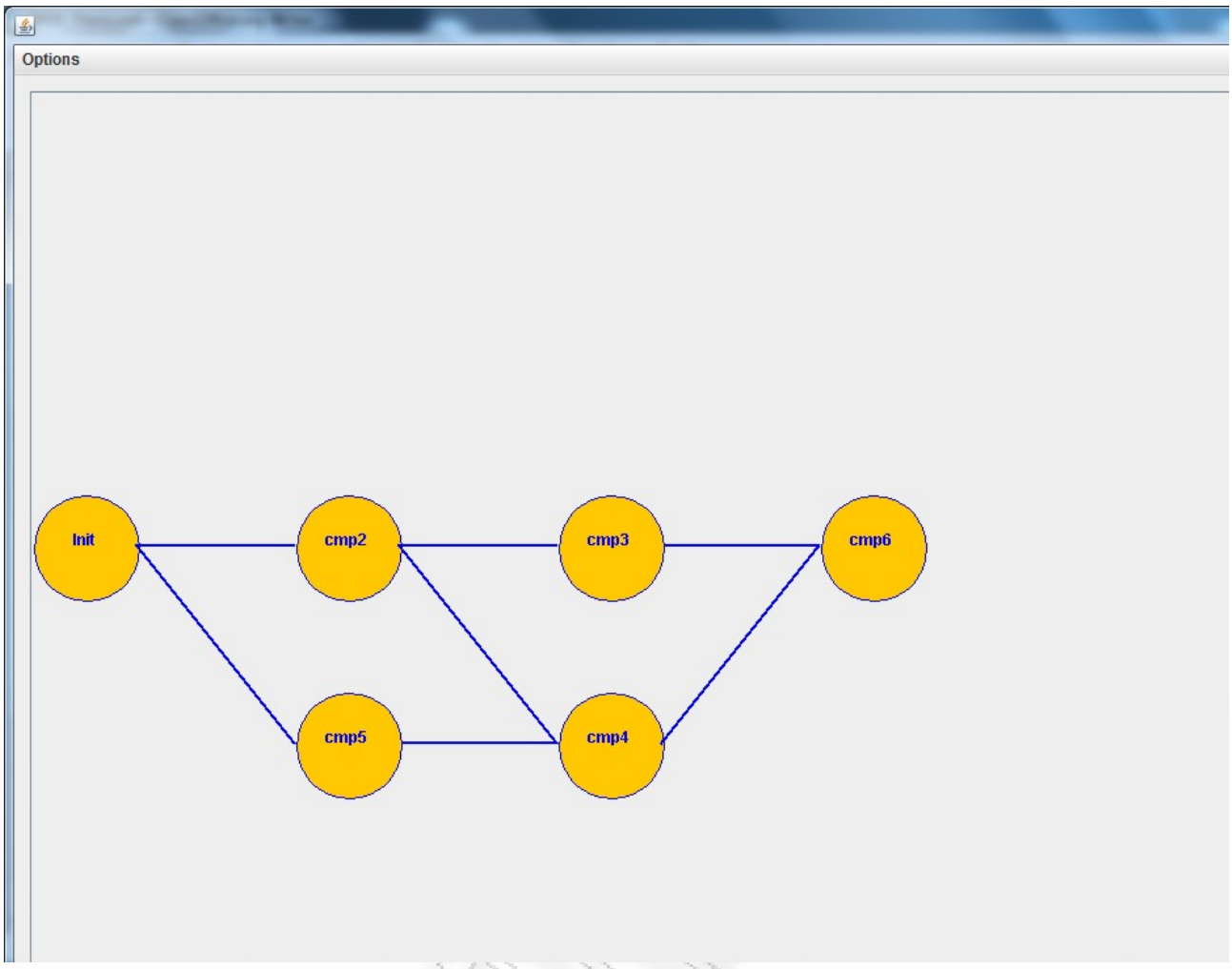

Options
Init
cmp2
cmp3
cmp4

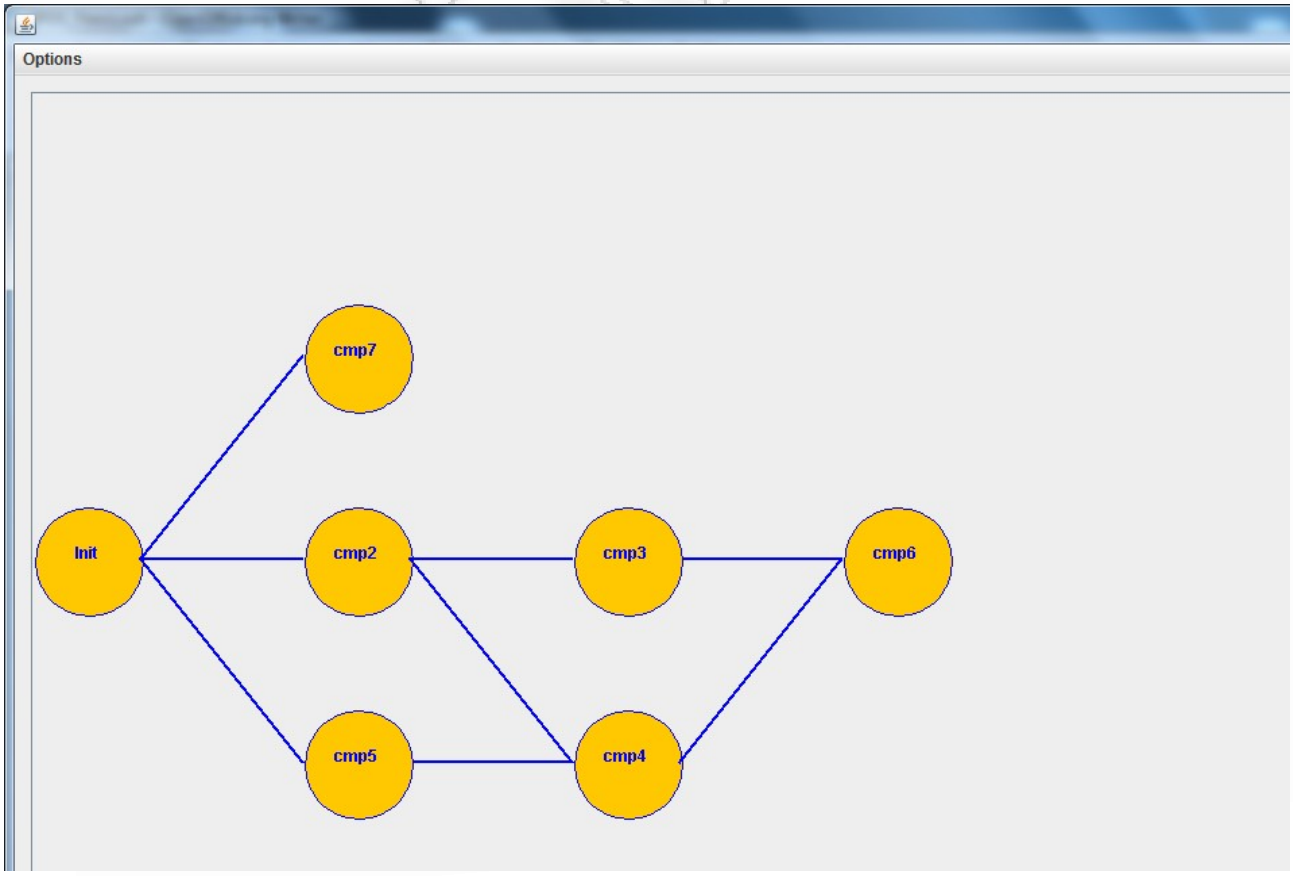

Options
Init
cmp2
cmp3
cmp5
cmp4
com.tc.postProcessing.components.MockUpSolutionComponent

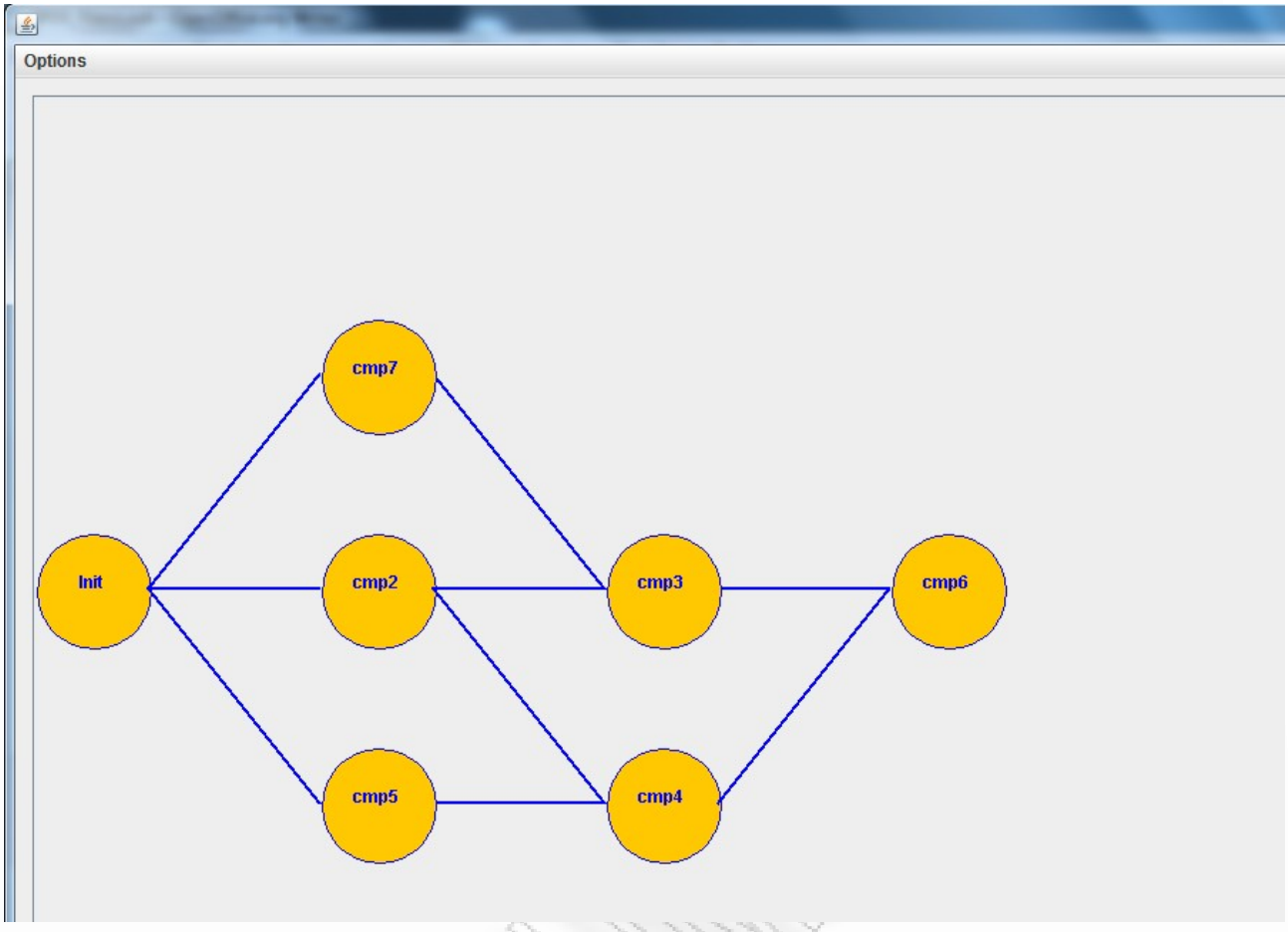

Options
Init
cmp2
cmp3
cmp5
cmp4
Add After
Connect to
cmp3
cmp4

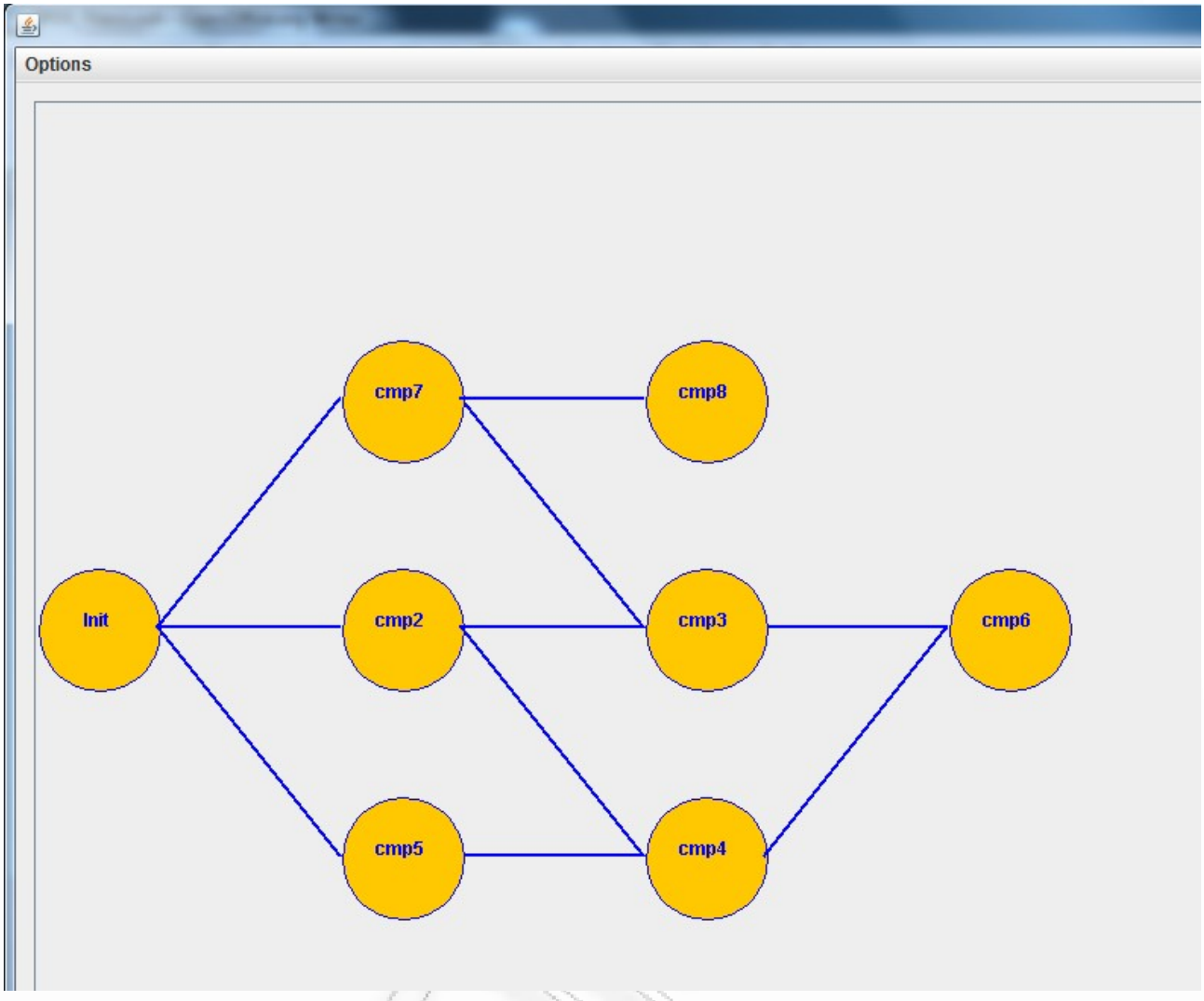

Options
Init
cmp2
cmp3
cmp5
cmp4

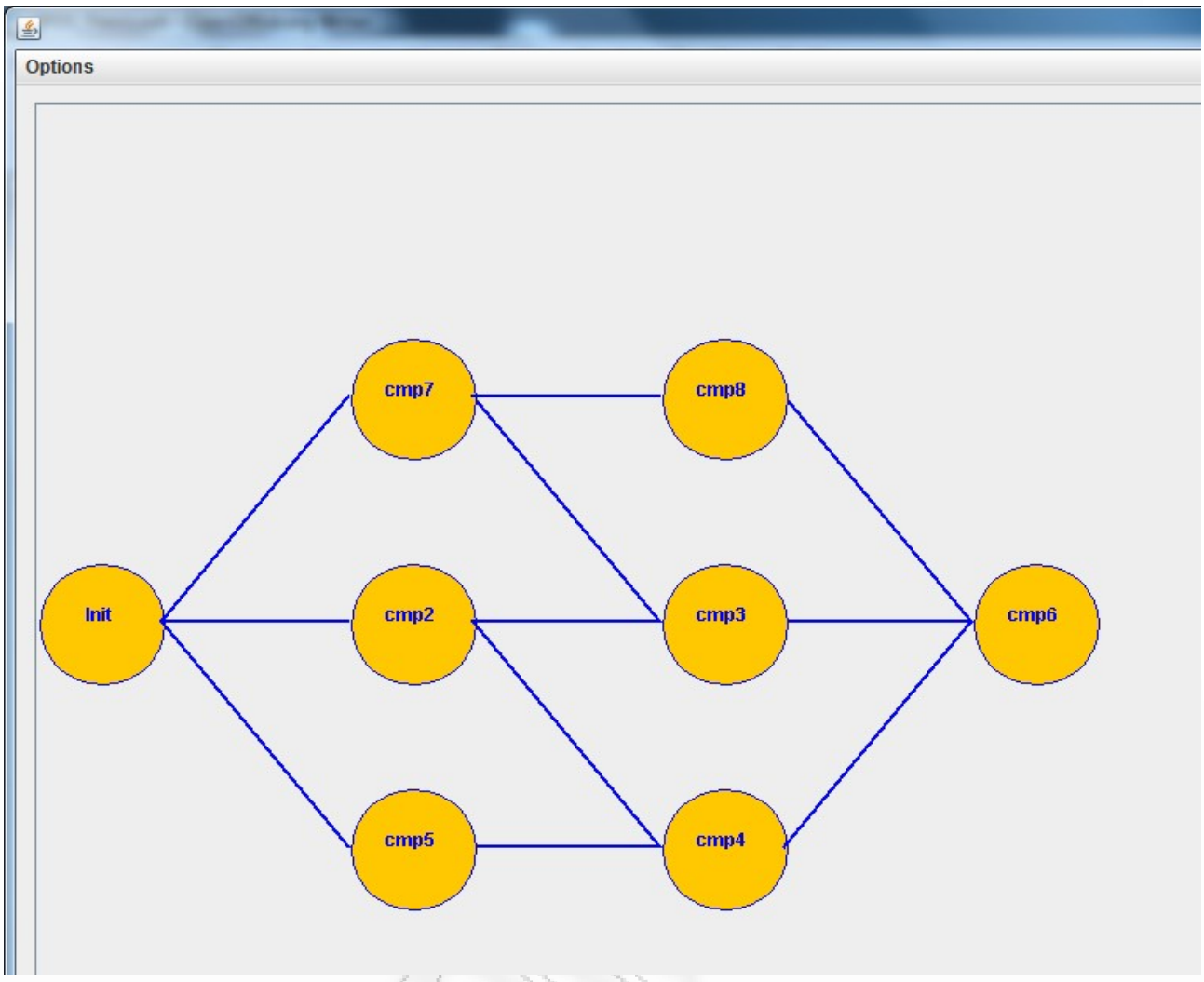

Options
Init
cmp2
cmp3
cmp6
cmp5
cmp4

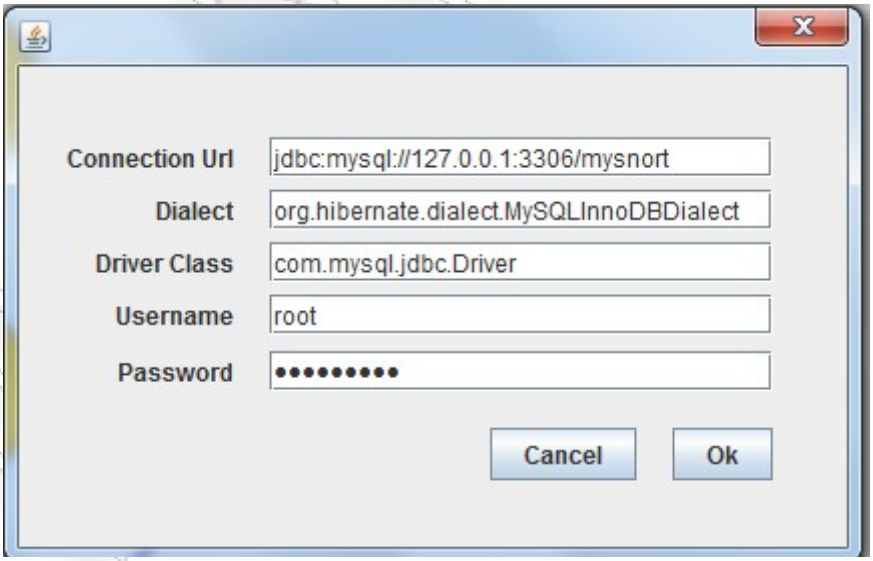

Options
Init
cmp2
cmp3
cmp6
cmp5
cmp4


Options
cmp7
Init
cmp2
cmp3
cmp6
cmp5
cmp4


Options
cmp7
Init
cmp2
cmp3
cmp6
cmp5
cmp4

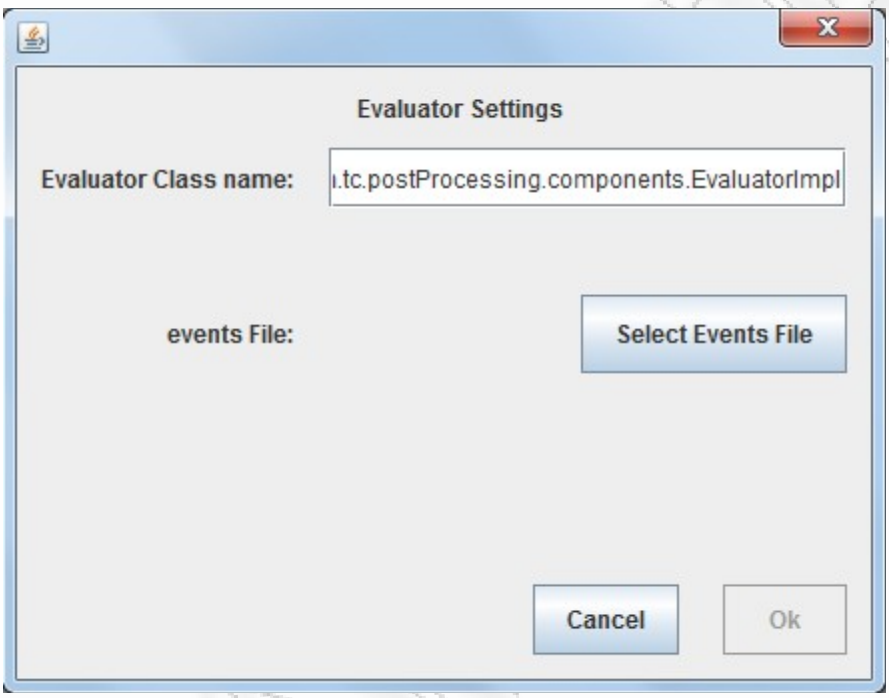

Options
cmp7
cmp8
Init
cmp2
cmp3
cmp6
cmp5
cmp4





Στην παραπάνω οθόνη φαίνονται οι παράμετροι που πρέπει να οριστούν για τη σύνδεση με τη βάση δεδομένων για να εισαχθούν στον σύστημα οι υποψήφιοι συνεγερμοί (alerts). Για τη σύνδεση με τη βάση δεδομένων χρησιμοποιήθηκε το ORM Framework Hibernate. Οι παράμετροι που πρέπει να οριστούν είναι:

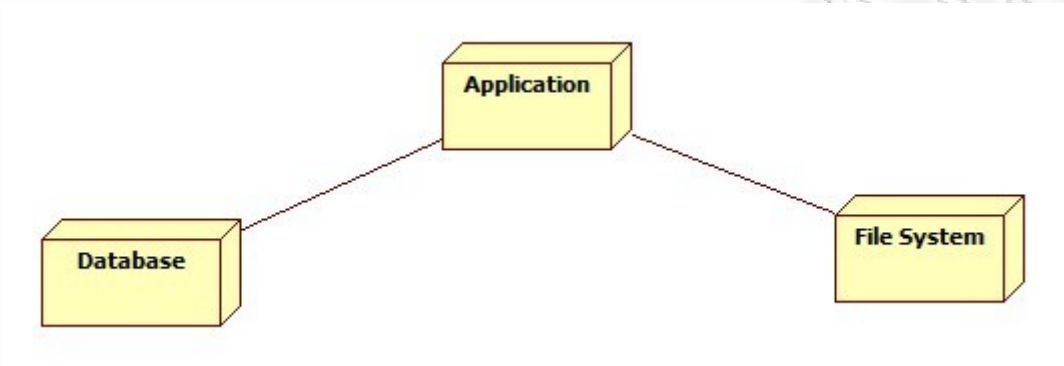
- Connection url για τη σύνδεση με τη βάση δεδομένων.
- Διάλεκτος για τη βάση δεδομένων.
- Java κλάση του database driver.
- Συνθηματικό χρήστη.
- Κωδικός χρήστη.



Στην παραπάνω οθόνη φαίνονται οι ρυθμίσεις που απαιτούνται για το στοιχείο που πραγματοποιεί την αξιολόγηση των εναπομείναντων συνεγερμών. Παρατηρούμε ότι απαιτείται το όνομα της Java κλάσης που περιέχει τη λογική της αξιολόγησης καθώς επίσης και ένα ή περισσότερα αρχεία που περιέχουν τις πραγματικές επιθέσεις.

Διεπαφές Λογισμικού

Στο ακόλουθο παραταξιακό διάγραμμα UML παρουσιάζονται τα συστατικά στοιχεία με τα οποία επικοινωνεί το λογισμικό.



3.1.2 Έγγραφο Περιγραφής Περίπτωσης χρήσης 1: Σχεδιασμός φίλτρου

Η παρούσα περίπτωση χρήσης αφορά τη δημιουργία ενός φίλτρου εκ του μηδενός. Ο χρήστης σχεδιάζει τη δομή επιλέγοντας ένα προς ένα τα components τα οποία θα την αποτελούν, αποθηκεύει και προαιρετικά εκτελεί το φίλτρο.

Ροή γεγονότων

Βασική ροή

1. Ο χρήστης μέσω του μενού options/Add Init επιλέγει κάποιο από τα διαθέσιμα αρχικά components.
2. Ο χρήστης συνδέει στο αρχικό component ένα από τα διαθέσιμα και επιτρεπόμενα component τα οποία είναι εγκατεστημένα στο σύστημα.
3. Ο χρήστης διαμορφώνει στο σχεδιαστικό περιβάλλον τη δομή του φίλτρου.
4. Ο χρήστης επιλέγει ένα από τα εγκατεστημένα στο σύστημα τελικά components.
5. Μετά την ολοκλήρωση του σχεδιασμού το φίλτρου ο χρήστης καθορίζει τις παραμέτρους για τη σύνδεση με τη βάση δεδομένων όπου θα εισαχθούν οι προς εξέταση υποψήφιοι συνεγερμοί.
6. Ο χρήστης επιλέγει το xml αρχείο που περιέχει με τις πραγματικές επιθέσεις.
7. Αποθήκευση του φίλτρου.
8. Εκτέλεση του φίλτρου.

Εναλλακτικές Ροές

Οι εναλλακτικές ροές προκύπτουν από τα ενδεχόμενα σφάλματα που μπορεί να παρουσιαστούν κατά τη χρήση του συστήματος.

Πρώτη εναλλακτική ροή

1. Σφάλμα στη σύνδεση με τη βάση δεδομένων.
2. Εμφάνιση μήνύματος λάθους.

Δεύτερη εναλλακτική ροή

1. Το αρχείο με τις πραγματικές επιθέσεις δεν περιέχει την κατάλληλη μορφοποίηση.
2. Εμφάνιση μηνύματος λάθους.

Τρίτη εναλλακτική ροή

1. Η σχεδιασμένη δομή δεν είναι έγκυρη.
2. Εμφάνιση κατάλληλου μηνύματος λάθους.

3.1.3 Έγγραφο Περιγραφής Περίπτωσης χρήσης 2: Τροποποίηση υπάρχοντος φίλτρου.

Η παρούσα περίπτωση χρήσης αφορά την τροποποίηση και κατόπιν εκτέλεσης ενός ήδη υπαρχόντος φίλτρου.

Ροή γεγονότων

Βασική ροή

1. Ο χρήστης μέσω της επιλογής Load του μενού Options επιλέγει το xml αρχείο στο οποίο έχει αποθηκευτεί το φίλτρο.
2. Ο χρήστης τροποποιεί τη δομή του φίλτρου στο σχεδιαστικό περιβάλλον.
3. Μετά την ολοκλήρωση του σχεδιασμού του φίλτρου ο χρήστης καθορίζει τις παραμέτρους για τη σύνδεση με τη βάση δεδομένων όπου θα εισαχθούν οι προς εξέταση υποψήφιοι συνεγερμοί.
4. Ο χρήστης επιλέγει το xml αρχείο που περιέχει με τις πραγματικές επιθέσεις.
5. Αποθήκευση του φίλτρου.
6. Εκτέλεση του φίλτρου.

Εναλλακτικές Ροές

Οι εναλλακτικές ροές προκύπτουν από τα ενδεχόμενα σφάλματα που μπορεί να παρουσιαστούν κατά τη χρήση του συστήματος.

Πρώτη εναλλακτική ροή

1. Το φίλτρο που εισάγεται στο σύστημα δεν έχει έγκυρη δομή.
2. Εμφάνιση κατάλληλου μηνύματος λάθους.

Δεύτερη εναλλακτική ροή

1. Σφάλμα στη σύνδεση με τη βάση δεδομένων.
2. Εμφάνιση μήνύματος λάθους.

Τρίτη εναλλακτική ροή

1. Το αρχείο με τις πραγματικές επιθέσεις δεν περιέχει την κατάλληλη μορφοποίηση.

2. Εμφάνιση μηνύματος λάθους.

Τέταρτη εναλλακτική ροή

1. Η σχεδιασμένη δομή δεν είναι έγκυρη.
2. Εμφάνιση κατάλληλου μηνύματος λάθους.

3.2 Περιγραφή Αρχιτεκτονικής-Έγγραφο Περιγραφής της Αρχιτεκτονικής

Το περιεχόμενο της ενότητας αυτής αφορά τη βασική και λεπτομερή σχεδίαση του συστήματος, η οποία παρουσιάζεται μέσω των εγγράφων:
«Έγγραφο Περιγραφής της Αρχιτεκτονικής» και «Έγγραφο Περιγραφής του Λεπτομερούς Σχεδίου», που ακολουθούν.

3.2.1 Έγγραφο Περιγραφής της Αρχιτεκτονικής

Σκοπός

Σκοπός αυτού του εγγράφου είναι η παρουσίαση της αρχιτεκτονικής του λογισμικού μετα επεξεργασίας συνεγερμών. Αποτελεί προϊόν της Βασικής Σχεδίασης του συστήματος.

Εμβέλεια

Αποδέκτες του εγγράφου είναι μηχανικοί λογισμικού που μπορεί να ασχοληθούν με την επέκταση του συστήματος καθώς και οι χρήστες του συστήματος.

Ορισμοί, ακρωνύμια, συντομογραφίες

GUI: Graphic User Interface: Γραφική Διεπαφή Χρήστη
JVM: Java Virtual Machine
Use Case Diagram: Διάγραμμα Περιπτώσεων Χρήσης
XML Parsing: Δομική Ανάλυση Αρχείου XML

Πλατφόρμα

Υλισμική πλατφόρμα του συστήματος είναι το PC που λειτουργεί η εφαρμογή και ο υπολογιστής στον οποίο είναι εγκατεστημένη η βάση δεδομένων που περιέχει τους προς εξέταση συνεγερμούς.

Σχολή /Γλώσσα προγραμματισμού

Ως σχολή προγραμματισμού θα χρησιμοποιηθεί η αντικειμενοστρεφής σχολή προγραμματισμού.
Ως γλώσσα προγραμματισμού θα χρησιμοποιηθεί η Java.

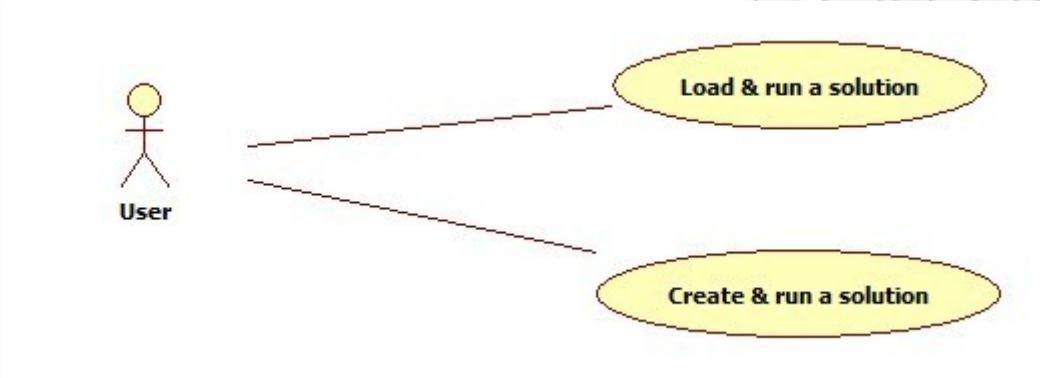
Κωδική γλώσσα

Η κωδική γλώσσα είναι η Java συν τις διαπροσωπείες (interfaces), που απαιτούνται από τις απαραίτητες βιβλιοθήκες για τη σύνδεση με τη βάση δεδομένων (Hibernate), την υποστήριξη γραφικού περιβάλλοντος (SWING), την αναπαράσταση του γράφου του φίλτρου ως ένα spring configuration αρχείου.

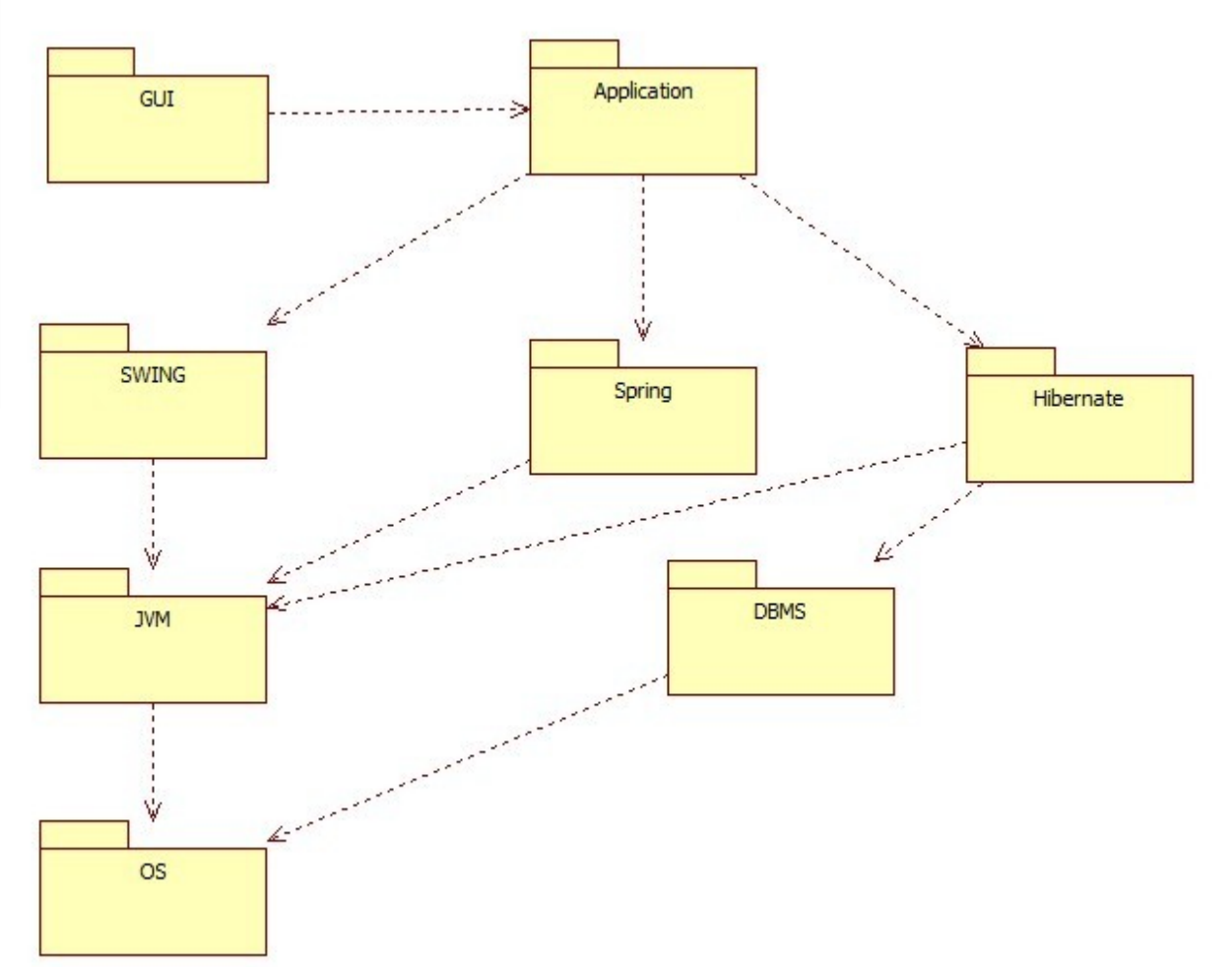
3.2.2 Αρχιτεκτονικές όψεις

Περιβαλλοντική /Υπηρεσιακή

Σ ’ αυτή την όψη φαίνονται οι δράστες (actors), που συνεργάζονται με το σύστημα. Ακολουθεί σχετικό Use Case Diagram.



Δομική



Σ’ αυτήν την όψη φαίνονται τα μέρη του συστήματος σε μακροσκοπικό επίπεδο καθώς και ποιες είναι οι εξαρτήσεις τους.

Στο υψηλότερο επίπεδο επίπεδο πραγματοποιείται η επικοινωνία εφαρμογής και χρήστη μέσω του γραφικού περιβάλλοντος. Στο επόμενο επίπεδο φαίνονται οι εξαρτήσεις της εφαρμογής από τα διάφορα frameworks. Το framework SWING είναι απαραίτητο για τη δημιουργία του γραφικού περιβάλλοντος χρήστη, το framework Hibernate για την επικοινωνία της εφαρμογής με τη βάση δεδομένων και το framework Spring για τη διαχείριση των διάφορων αντικειμένων που απαιτούνται από την εφαρμογή.

3.3 Έγγραφο Περιγραφής του Λεπτομερούς Σχεδίου

3.3.1 Εισαγωγή

Σκοπός

Σκοπός αυτού του εγγράφου είναι η παρουσίαση του Λεπτομερούς Σχεδίου του συστήματος μετα επεξεργασίας συνεγερμών. Το παρόν έγγραφο αποτελεί προϊόν της Λεπτομερούς Σχεδίασης του συστήματος.

Ενδιαφερόμενοι

Ενδιαφερόμενοι αυτού του εγγράφου είναι ο αρχιτέκτων του συστήματος, σχεδιαστές, προγραμματιστές και ελεγκτές.

Ορισμοί, ακρωνύμια, συντομογραφίες

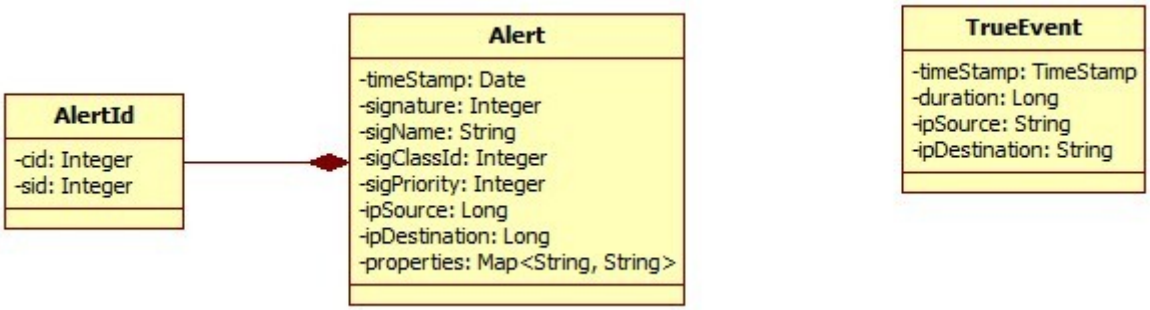
Σχεδιαστικές αποφάσεις

Οι σχεδιαστικές αποφάσεις που πρέπει να ληφθούν αφορούν την επεκτασιμότητα του συστήματος. Το σύστημα θα πρέπει να δίνει τη δυνατότητα στο χρήστη να μπορεί να ορίσει νέα components που θα μπορούν να χρησιμοποιηθούν για τη δημιουργία νέων φίλτρων. Επιπλέον σχεδιαστική απόφαση αποτελεί η επιλογή της δομής που αναπαριστά ένα συνεγερμό καθώς θα πρέπει να είναι ευέλικτη και να επιτρέπει την προσθήκη νέων γνωρισμάτων. Μία ακόμη σχεδιαστική απόφαση είναι μορφή με την οποία θα αποθηκεύεται το παραγόμενο από το χρήστη φίλτρο έτσι ώστε να μπορεί να το χρησιμοποιήσει αργότερα με διαφορετικά δεδομένα στο σύστημα. Επίσης μέρος της σχεδίασης αποτελεί και η σχεδίαση του γραφικού περιβάλοντος χρήστη. Τέλος βασικό τμήμα της σχεδίασης αποτελεί και η λεπτομερής σχεδίαση των τμημάτων κώδικα για την υλοποίηση του συστήματος.

3.3.2 Σχεδιαστικές Όψεις

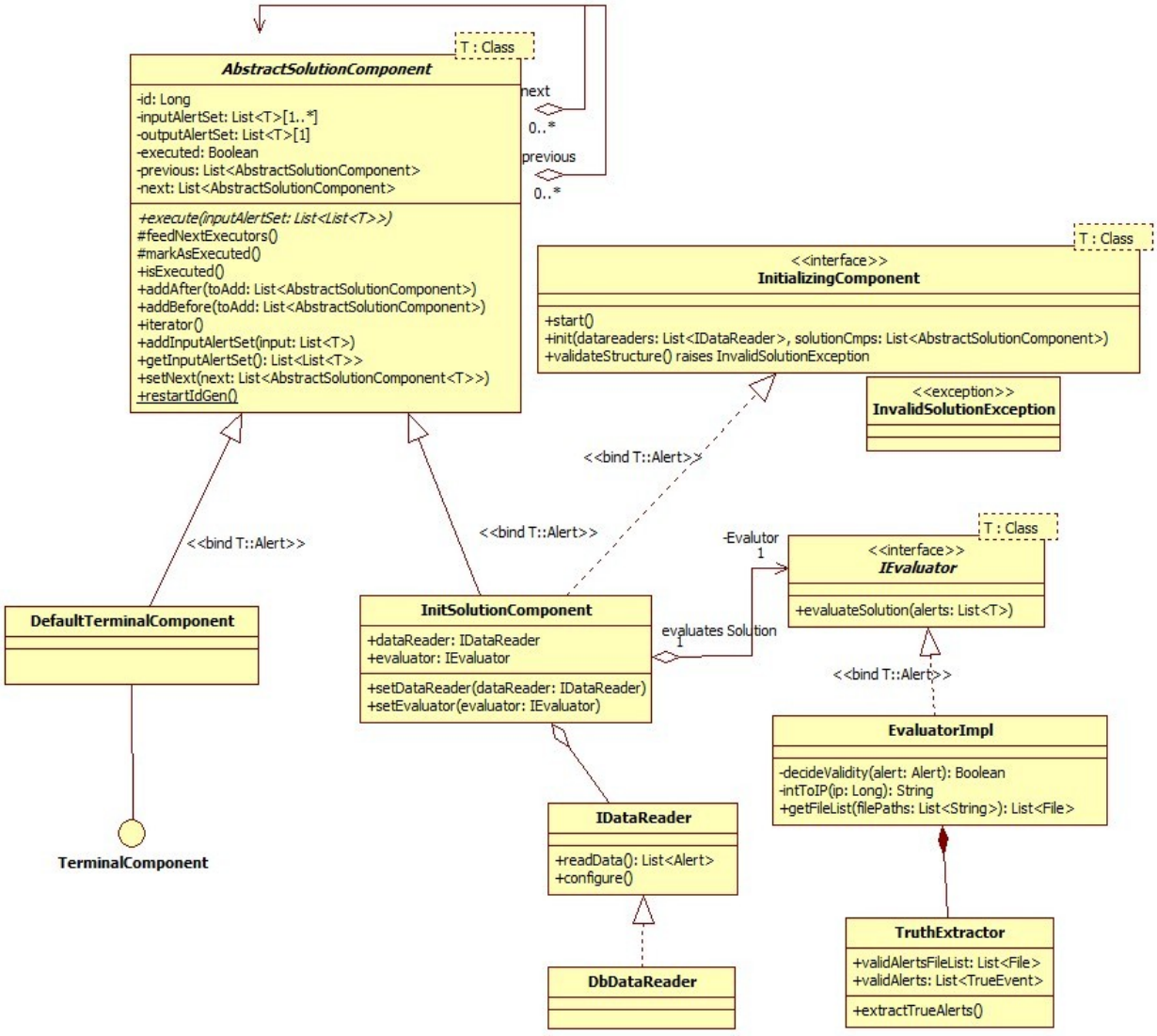
Αποσυνθετική

Στα σχήματα που ακολουθούν παρατίθενται τα διαγράμματα κλάσεων της εφαρμογής. Συγκεκριμένα στο παρακάτω σχήμα φαίνεται η μοντελοποίηση του υποψήφιου συνεγερμού καθώς και του έγκυρου συνεγερμού

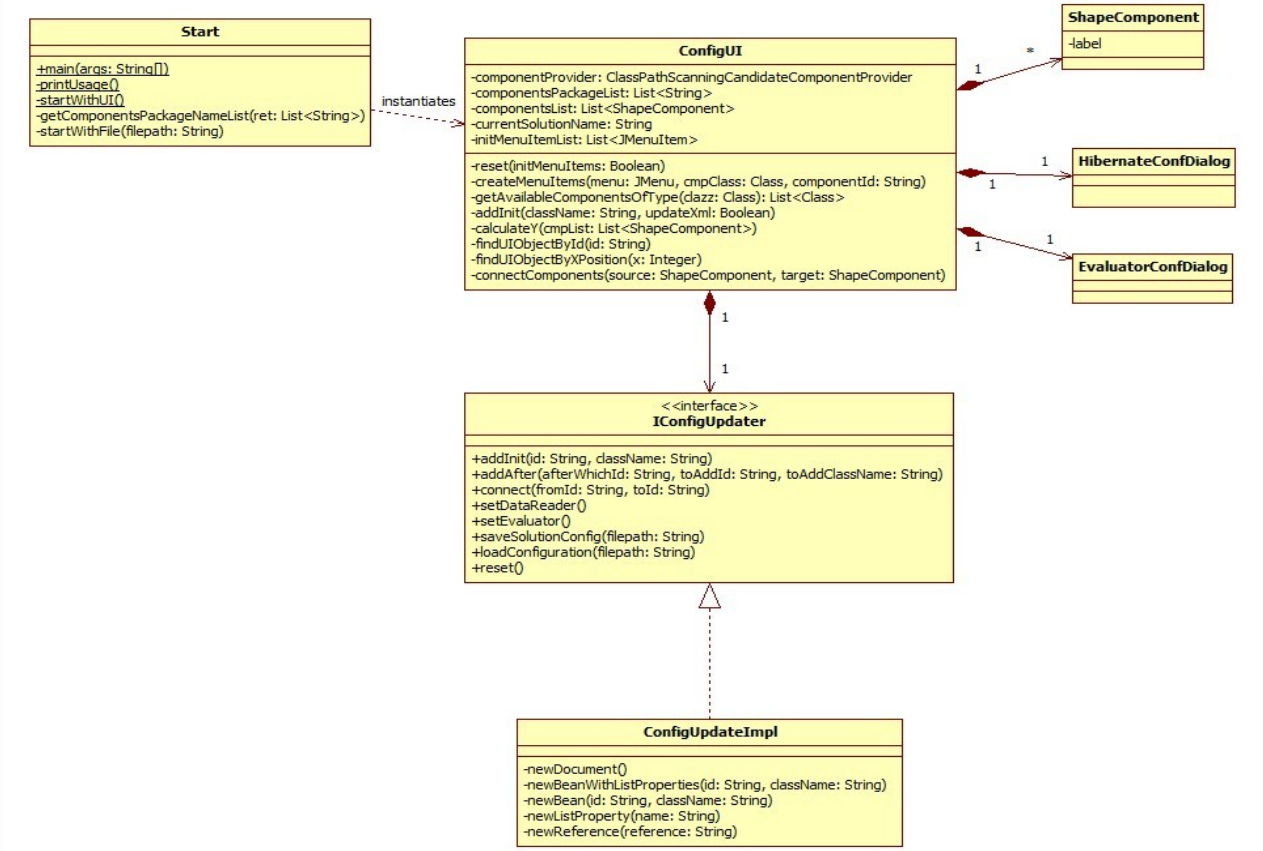


.

Ακολουθεί το διάγραμμα κλάσεων ενός αφηρημένου component ενός φίλτρου.



Τέλος παρουσιάζεται το διάγραμμα κλάσεων που αφορά τις κλάσεις που συνθέτουν το γραφικό περιβάλλον της εφαρμογής.



Διαπροσωπειακή

Ακολουθεί αναλυτική παρουσίαση των κλάσεων και των αντίστοιχων διαπροσωπειών τους, για τις οποίες πρέπει να γραφεί κώδικας σε Java.

Κλάση Start.

Η συγκεκριμένη κλάση είναι η main class της εφαρμογής.

Μεθόδοι

Start() : Ο κατασκευαστής της κλάσης.

printUsage() : Εμφανίζει ένα μήνυμα στην κονσόλα που πληροφορεί το χρήστη σχετικά με τις διαθέσιμες παραμέτρους με τις οποίες μπορεί να κληθεί η εφαρμογή.

startWithUI() : Πραγματοποιείται εκκίνηση της εφαρμογής μέσω γραφικού περιβάλλοντος.

getComponentsPackageNameList() : Διαβάζει από τα αρχεία παραμετροποίησης της εφαρμογής το όνομα των πακέτων (package names) στα οποίο αναμένεται να βρίσκονται τα διαθέσιμα components για τη δημιουργία ενός φίλτρου.

startWithFile() : Πραγματοποιεί εκτέλεση μίας συγκεκριμένης δομής φίλτρου.

Κλάση ConfigUI : Αποτελεί τη βασική κλάση για την υλοποίηση του γράφικού περιβάλλοντος χρήστη.

Μεθόδοι

Στον κατασκευστή της κλάσης καλούνται όλες οι μεθόδοι για την αρχικοποίηση του γραφικού περιβάλοντος.

reset() : Η συγκεκριμένη μέθοδος αρχικοποιεί την περιοχή σχεδίασης στο γραφικό περιβάλλον.

createMenuItems() : Με την κλήση αυτής της μεθόδου δημιουργούνται τα μενού με τις διαθέσιμες επιλογές.

getAvailableComponentsOfType(Class clazz) : Βρίσκει όλες τις κλάσεις που κληρονομούν από εκείνη που δίδεται ως όρισμα. Χρησιμοποιείται για να προσδιοριστούν οι κλάσεις των components που ο χρήστης έχει ορίσει.

addInit(): Προσθέτει έναν αρχικό κόμβο στο φίλτρο και ενημερώνει την περιοχή απεικόνισης φίλτρου στο γραφικό περιβάλλον.

add() : Προσθέτει έναν τυχαίο κόμβο στο φίλτρο και ενημερώνει την περιοχή απεικόνισης φίλτρου στο γραφικό περιβάλλον.

calculateY() : Βοηθητική μέθοδος που υπολογίζει το ύψος στο οποίο θα πρέπει να τοποθετηθέι ένα νέο component στο γραφικό περιβάλλον που απεικονίζει το φίλτρο.

findUIObjectById(String id) : Προσδιορίζει το αντικείμενο που αντιπροσωπεύει γραφικά το component με το δοθέν αναγνωριστκό.

findUIObjectByXPosition(int x) : Βρίσκει τα γραφικά αντικείμενα που βρίσκονται στο δοθέν μήκος της περιοχής απεικόνισης του φίλτρου στο γραφικό περιβάλλον.

Διεπαφή IConfigUpdater

Το παρόν interface ορίζει τις μεθόδους που χρησιμοποιούνται για το χειρισμό του xml αρχείου που αντιπροσωπεύει το φίλτρο.

Μεθόδοι

addInit(String id, String className) : Προσθέτει στο xml αρχείο του φίλτρου τον κατάλληλο xml κώδικα που αντιπροσωπεύει έναν αρχικό κόμβο.

addAfter(String afterWhichId, String toAddId, String toAddClassName) : Προσθέτει έναν κόμβο φίλτρου στο xml αρχείο του φίλτρου.

connect(String fromId, String toId) : Προσθέτει στο xml αρχείο του φίλτρου τον κατάλληλο xml κώδικα που αναπαριστά τη σύνδεση μεταξύ δύο components του φίλτρου. Τα components αναγνωρίζονται από τα αναγνωριστικά τους

setDataReader() : Η παρούσα μέθοδος ενημερώνει το xml αρχείο με τα απαραίτητα δεδομένα για τη σύνδεση με τη βάση δεδομένων που περιέχει τους προς εξέταση συνεγερμούς.

setEvaluator() : Η παρούσα μέθοδος ενημερώνει το xml που αναπαριστά το φίλτρο με τα απαραίτητα δεδομένα έτσι ώστε αφού εκτελεστεί το φίλτρο να αξιολογηθούν οι εναπομείναντες συνεγερμοί.

saveSolutionConfig(String filepath) : Με την κλήση της παρούσας μεθόδου πραγματοποιείται η αποθήκευση του xml που αναπαριστά το φίλτρο.

loadConfiguration(String filepath) : Πραγματοποιείται επεξεργασία του xml που βρίσκεται στη δοθείσα διαδρομή στο δίσκο και εφόσον είναι έγκυρο πραγματοποιείται εισαγωγή της παρούσας δομής του φίλτρου στο σύστημα.

void reset() : Επαναφέρει το xml document στην αρχική κατάσταση δηλαδή κενό φίλτρο. (φίλτρο που δεν περιέχει κανένα component).

Κλάση ConfigUpdateImpl. Η παρούσα κλάση υλοποιεί το interface IconfigUpdater. Ακολουθεί η περιγραφή των μεθόδων που δεν έχουν αναφερθεί προηγουμένως στην περιγραφή του interface IconfigUpdater.

newDocument() : Επιστρέφει ένα καινούργιο xml Document.

newBeanWithListProperties() : Εισάγει ένα νέο bean στο xml που αναπαριστά το φίλτρο που περιέχει μία λίστα ως γνώρισμα.

newBean() : Εισάγει ένα καινούργιο bean στο xml που αναπαριστά το φίλτρο.

newListProperty() : Προσθέτει στο bean ένα νέο γνώρισμα τύπου λίστας.

newReference() : Εισαγωγή ενός νέου bean reference.

locateElement() : Εύρεση ενός xml στοιχείου με χρήση Xpath.

Κλάση ShapeComponent. Αναπαριστά γραφικά ένα component ενός φίλτρου.

Στον κατασκευαστή της κλάσης δίνεται το μήκος και το πλάτος στη περιοχή γραφικής απεικόνισης του φίλτρου στο οποίο θα τοποθετηθεί το αντικείμενο.

Μεθόδοι

drawCircle() : Πραγματοποιεί την απεικόνιση του component του φίλτρου σε σχήμα κύκλου.

Κλάση AbstractSolutionComponent: Αποτελεί την αφηρημένη μορφή component ενός φίλτρου. Όλα τα προερχόμενα από το χρήστη components θα πρέπει να κληρονομούν από τη συγκεκριμένη κλάση.

Μεθόδοι

Κατασκευαστής: Πραγματοποιείται αρχικοποίηση του αντικειμένου.

execute(List<List<T>> inputAlertSet) : η αφηρημένη μέθοδος που θα πρέπει να υλοποιηθεί από τις κλάσεις που κληρονομούν τη συγκεκριμένη κλάση. Στη συγκεκριμένη μέθοδο θα πρέπει να υλοποιηθεί λογική του εκάστοτε δομικού στοιχείου.

feedNextExecutors() : Τροφοδοτεί τα επόμενα components της δομής ενός φίλτρου με μία λίστα που περιέχει συνεγερμούς οι οποίοι με βάση το συγκεκριμένο component θεωρούνται έγκυροι.

addAfter() : Με τη χρήση της συγκεκριμένης μεθόδου μπορούμε να προσθέσουμε ένα νέο κόμβο σε μία δομή ύστερα από ένα συγκεκριμένο με προγραμματιστικό τρόπο.

addBefore() : Με τη χρήση της συγκεκριμένης μεθόδου μπορούμε να προσθέσουμε ένα νέο κόμβο σε μία δομή πριν από ένα συγκεκριμένο με προγραμματιστικό τρόπο.

iterator() : Η συγκεκριμένη μέθοδος επιστρέφει τα components του φίλτρου ταξινομημένα με την ίδια ακριβώς σειρά που θα εκτελεστούν.

getOutputAlertSet() : Επιστρέφει μία λίστα με τους συνεγερμούς τους οποίους το συγκεκριμένο component έχει θεωρήσει ως έγκυροους.

addInputAlertSet() : Προσθέτει μία λίστα με συνεγερμούς τους οποίους θα επεξεργαστεί ο συγκεκριμένος κόμβος.

getInputAlertSet() : Επιστρέφει μία λίστα με όλους τους συνεγερμούς που θα επεξεργαστεί ο συγκεκριμένος κόμβος του φίλτρου.

setNext() : Θέτει τους κόμβους του φίλτρου που θα εκτελεστούν μετά από το συγκεκριμένο κόμβο.

getNext() : Επιστρέφει τους κόμβους του φίλτρου που θα εκτελεστούν μετά από το συγκεκριμένο κόμβο.

setPrevious() : Θέτει τους κόμβους του φίλτρου που θα εκτελεστούν πριν από το συγκεκριμένο κόμβο.

restartIdGen() : Επανεκκινεί το μηχανισμό παραγωγής μοναδικών αναγνωριστικών για τους κόμβους του φίλτρου.

Διεπαφή InitializingComponent : Διεπαφή που θα πρέπει να υλοποιεί ένας κόμβος ο οποίος έχει οριστεί ως αρχικός στη δομή του φίλτρου.

Κλάση InitSolutionComponent : Αποτελεί την υλοποίηση ενός κόμβου που μπορεί να οριστεί ως αρχικός σε μία δομή ενός φίλτρου.

Διεπαφή TerminalComponent : Διεπαφή που θα πρέπει να υλοποιεί ένας κόμβος ο οποίος έχει οριστεί ως τελικός στη δομή του φίλτρου.

Κλάση DefaultTerminalComponent : Αποτελεί την υλοποίηση ενός κόμβου που μπορεί να οριστεί ως τελικός σε μία δομή ενός φίλτρου.

Διεπαφή IDataReader : Η διεπαφή που ορίζει τον τρόπο με τον οποίο θα εισάγονται προς εξέταση συνεγερμοί στο σύστημα.

readData() : Η κλήση της συγκεκριμένης μεθόδου εισάγει στο σύστημα τους προς εξέταση συνεγερμούς.

configure() : Με την κλήση της συγκεκριμένης μεθόδου πραγματοποιείται η παραμετροποίηση για τη σύνδεση με τη βάση δεδομένων που περιέχει τους προς εξέταση συνεγερμούς..

Κλάση DbDataReader : Αποτελεί την υλοποίηση της διεπαφής IDataReader. Υποστηρίζεται η σύνδεση με διάφορες βάσεις δεδομένων.

Διεπαφή IEvaluator : Δειπαφή που θα πρέπει να υλοποιεί μία κλάση που πραγματοποιεί αποτίμηση του φίλτρου.

evaluateSolution() : Η κλήση της συγκεκριμένης μεθόδου πραγματοποιεί την αξιολόγηση του φίλτρου και αποθηκεύει τα αποτελέσματα σε κάποιο μέσο.

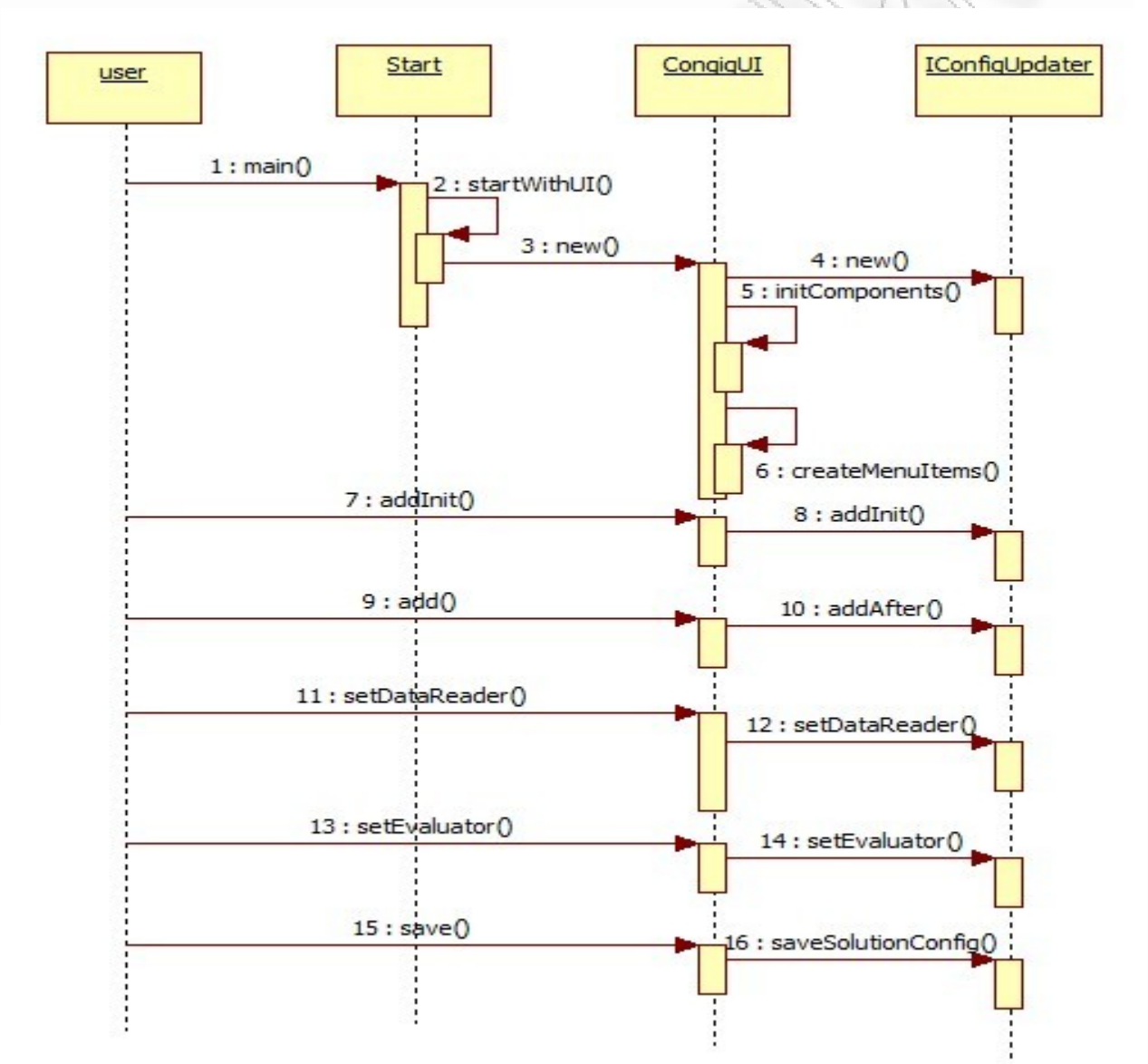
Κλάση EvaluatorImpl : Πρόκειται για υλοποίηση της διεπαφής IEvaluator όπου τα αποτελέσματα της αξιολόγησης αποθηκεύονται σε αρχεία.

Με τη βοήθεια των δύο παρακάτω sequence diagrams παρουσιάζεται μία αναλυτική

περιγραφή των λειτουργιών αποθήκευση και εκτέλεση ενός φίλτρου.

3.3.3 Δημιουργία και αποθήκευση φίλτρου.

Στο παρακάτω sequence diagram ο χρήστης κατασκευάζει ένα φίλτρο με δύο κόμβους. Κατόπιν καθορίζει τις παραμέτρους σύνδεσης με τη βάση δεδομένων, επιλέγει το xml αρχείο που περιέχει τις έγκυρες επιθέσεις και τέλος αποθηκεύει το φίλτρο σε μορφή xml στο δίσκο.



Στη συνέχεια δίδεται μία σύντομη περιγραφή των βημάτων που απεικονίζονται στο παραπάνω sequence diagram.

1: Ο χρήστης εκκινεί την java εφαρμογή οπότε πραγματοποιείται κλήση της μεθόδου main().

2: Η κλήση της συγκεκριμένης μεθόδου πραγματοποιεί αρχικοποίηση του γραφικού περιβάλλοντος.

3: Δημιουργία (instantiation) ενός αντικειμένου της κλάσης ConfigUI.

4: Αρχικοποίηση ενός αντικειμένου IConfigUpdater.

5: Με την κλήση της μεθόδου initComponents πραγματοποιείται η αρχικοποίηση της κλάσης ConfiUI.

6: Δημιουργία και αρχικοποίηση του μενού επιλογών.

7: Ο χρήστης προσθέτει έναν αρχικό κόμβο στη δομή του φίλτρου.

8: Προσθήκη xml κώδικα που αναπαριστά έναν αρχικό κόμβο.

9: Ο χρήστης προσθέτει εναν επιπλέον κόμβο στη δομή του φίλτρου. Το παρόν βήμα επαναλαμβάνεται όσες φορές χρειαστεί μέχρι να ολοκληρωθεί ο σχεδιασμός της δομής του φίλτρου από το χρήστη.

10: Ενημέρωση του xml που αναπαριστά το συγκεκριμένο φίλτρο.

11: Ο χρήστης πραγματοποιεί τη ρύθμιση των παραμέτρων για τη σύνδεση με τη βάση δεδομένων.

12: Πραγματοποιείται ρύθμιση των παραμέτρων για τη σύνδεση με τη βάση δεδομένων στο xml αρχείο.

13: Ο χρήστη πραγματοποιεί ρύθμιση των παραμέτρων για την αξιολόγηση του φίλτρου.

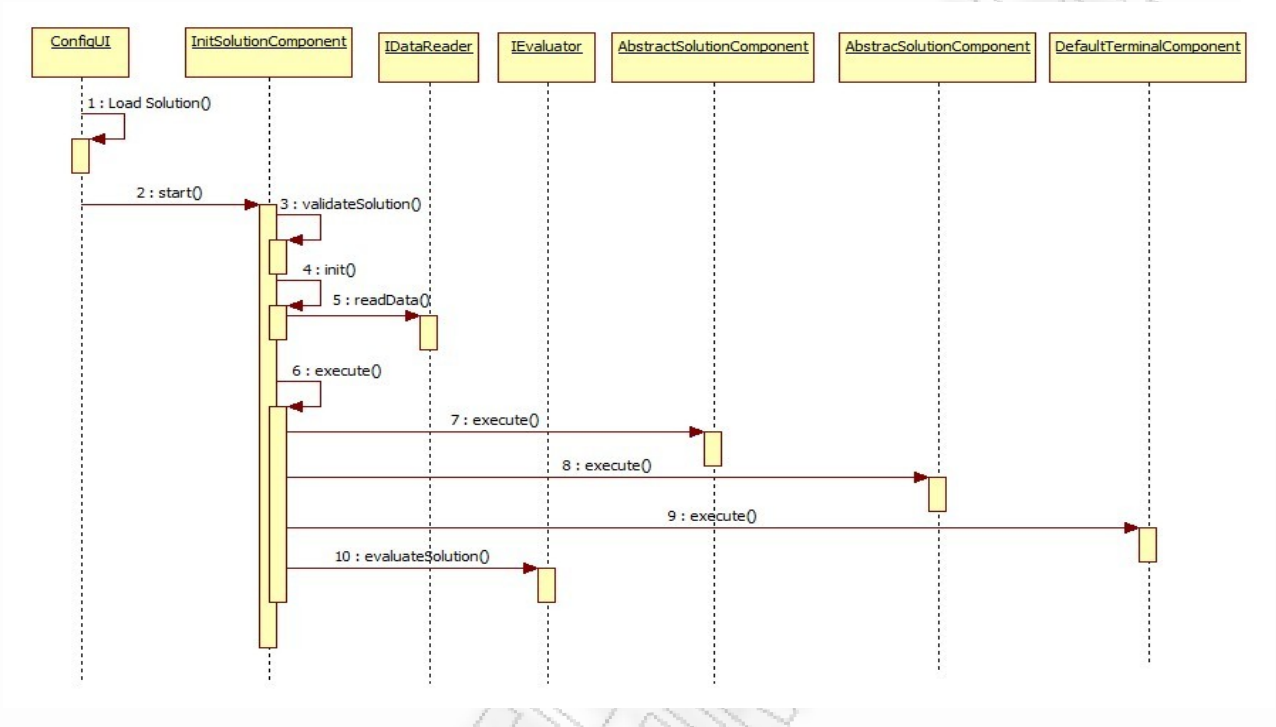
14: Οι επιλογές του χρήστη αποθηκεύονται στο xml.

15: Ο χρήστης επιλέγει την αποθήκευση του φίλτρου.

16: Αποθηκεύεται το xml που αναπαριστά το φίλτρο στο File System.

3.3.4 Εισαγωγή στο σύστημα και εκτέλεση του φίλτρου.

Στο παρακάτω sequence diagram παρουσιάζεται η περίπτωση της εισαγωγής στο σύστημα από ένα xml αρχείο και κατόπιν εκτέλεση ενός φίλτρου.



Ακολουθεί μία σύντομη περιγραφή των βημάτων που απεικονίζονται στο παραπάνω διάγραμμα.

1: Επιλογή του αρχείου στο οποίο είναι αποθηκευμένο το φίλτρο και εισαγωγή του στο σύστημα.
2: Κλήση της μεθόδου start όπου εκκινείται η εκτέλεση της δομής που αναπαριστά το φίλτρο.
3: Έλεγχος ορθότητας του φίλτρου.
4: Αρχικοποίηση του πρώτου κόμβου του φίλτρου.
5: Σύνδεση με τη βάση δεδομένων και ανάκτηση των προς εξέταση συνεγερμών.
6: Κλήση της μεθόδου execute() του αρχικού κόμβου της δομής.
7: Προσπέλαση όλων των κόμβων και εκτέλεση της λογικής κάθε ενός ξεχωριστά. Στο συγκεκριμένο βήμα εκτελείται ο δεύτερος κόμβος.
8: Πραγματοποιείται η εκτέλεση της λογικής του δεύτερου κόμβου.
9: Εκτελεση του τελικού κόμβου.
10: Αξιολόγηση των τελικών συνεγερμών με βάση ένα xml αρχείο που περιέχει τις έγκυρες επιθέσεις και αποθήκευση του αποτελέσματος σε αρχείο στο δίσκο.

3.4 Υλοποίηση

Στην παράγραφο που ακολουθεί περιγράφονται οι πλατφόρμες λογισμικού που επιλέχθηκαν καθώς και οι λόγοι που οδήγησαν στην επιλογή τους. Τέλος δίδεται μία περιγραφή για την εγκατάσταση και παραμετροποίηση του συστήματος.

3.4.1 Πλατφόρμες και προγραμματιστικά εργαλεία

JAVA

Η Java είναι μια από τις πιο διαδεδομένες αντικειμενοστρεφείς γλώσσες προγραμματισμού. Είναι ασφαλής και ανεξάρτητη πλατφόρμας και χρησιμοποιείται σήμερα ευρέως σε πολλούς τομείς. Ιδιαίτερα διαδεδομένη είναι η χρήση της στους web servers, τις βάσεις δεδομένων και τα web services. Για την εκτέλεση των προγραμμάτων java απαιτείται η εγκατάσταση μίας java virtual machine (JVM). Μία JVM είναι μια υλοποίηση σε λογισμικό μιας κεντρικής μονάδας επεξεργασίας (CPU) που τρέχει compiled κώδικα java (compiled java bytecode). Ως περιβάλλον ανάπτυξης χρησιμοποιήθηκε η έκδοση 1.6 του standard development kit (sdk) της java και το Integrated Development Environment (IDE) NetBeans 7.0

Rational Unified Process-RUP

Η RUP είναι μία μεθοδολογία ανάπτυξης λογισμικού που αναπτύχθηκε από την εταιρία Rational και βασίζεται στη UML. Δίνει τη δυνατότητα αυτόματης παραγωγής κώδικα μέσα από μία αναλυτική διαδικασία σχεδίασης. Στην παρούσα εφαρμογή τα έγγραφα της σχεδίασης του προς ανάπτυξη λογισμικού συστήματος βασίστηκαν στα αντίστοιχα της RUP, καθώς και το σχεδιαστικό εργαλείο StarUML, χρησιμοποιήθηκε για τη δημιουργία των UML διαγραμμάτων.

MySql

Πρόκειται για ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), που λειτουργεί ως διακομιστής παρέχει πρόσβαση πολλών χρηστών σε μια σειρά από βάσεις δεδομένων. Ο πηγαίος κώδικας της MySQL διατίθεται υπό τους όρους της GNU General Public License, καθώς και κάτω από μια ποικιλία άλλων αδειών. Η MySQL ανήκε στη σουηδική εταιρεία MySQL AB, που τώρα ανήκει στην Oracle Corporation. Οι προς εξέταση συνεγερμοί που εισάγονται στο σύστημα βρίσκονται αποθηκευμένοι σε σχεσιακή βάση δεδομένων MySQL.

Snort

Το Snort είναι ένα ανοικτού κώδικα IPDS (intrusion prevention detection system), που δημιουργήθηκε από τον Martin Roesch το 1998. Το Snort έχει τη δυνατότητα να εκτελέσει σε πραγματικό χρόνο ανάλυση της κυκλοφορίας και καταγραφής πακέτων σε IP δίκτυα. Το Snort εκτελεί ανάλυση πρωτοκόλλου, αναζήτηση περιεχομένου και ταίριασμα περιεχομένου. Το Snort μπορεί να ρυθμιστεί σε τρεις βασικούς τρόπους λετουργιας: 1) ανίχνευση, 2) καταγραφής πακέτων, και 3) ανίχνευσης εισβολών σε επίπεδο δικτύου. Στη λειτουργία ανίχνευσης, το πρόγραμμα θα διαβάσει πακέτα δικτύου και θα τα εμφανίσει στην κονσόλα. Στη λειτουργία καταγραφής πακέτων, το πρόγραμμα θα καταγράψει πακέτα στο δίσκο. Στη λειτουργία ανίχνευσης εισβολής, το πρόγραμμα θα παρακολουθεί την κίνηση του δικτύου και θα την αναλύσει με βάση ένα σύνολο κανόνων που ορίζονται από το χρήστη. Το πρόγραμμα θα εκτελέσει έπειτα μια συγκεκριμένη ενέργεια που βασίζεται σε ρυθμίσεις που έχουν προσδιοριστεί.

Hibernate

Το framework Hibernate αποτελεί ένα εργαλείο αντιστοίχισης ενός αντικειμενοστραφούς μοντέλου σε μία παραδοσιακή σχεσιακή βάση δεδομένων. Πρωταρχικός στόχος της συγκεκριμένης βιβλιοθήκης αποτελεί η προβολή Java κλάσεων σε πίνακες σε μία βάση δεδομένων. Επίσης παρέχονται ευκολίες για τη δημιουργία ερωτημάτων και την ανάκτηση περιεχομένου. Τα ερωτήματα της εφαρμογής δεν πραγματοποιούνται απευθείας στη βάση δεδομένων αλλά γίνονται μέσω του Hibernate. Αυτό έχει ως αποτέλεσμα να είναι δυνατόν να υποστηριχθούν βάσεις δεδομένων που προέρχονται από διαφορετικούς κατασκευαστές χρησιμοποιώντας η εφαρμογή τον ίδιο πηγαίο κώδικα. Η σύνδεση της εφαρμογής με τη βάση δεδομένων πραγματοποιείται με χρήση του συγκεκριμένου framework.

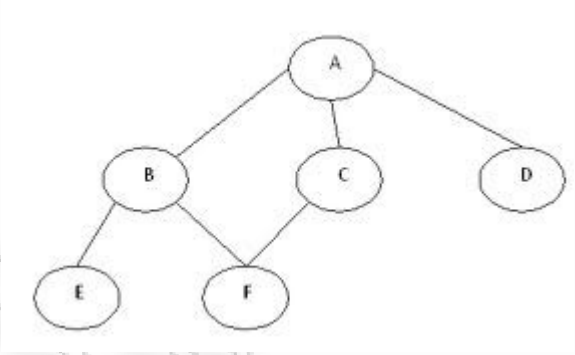
Spring

Ένα ακόμη framework που χρησιμοποιήθηκε για τη δημιουργία της συγκεκριμένης εφαρμογής είναι το spring. Το spring είναι ευρέως διαδεμένο σε εφαρμογές που είναι γραμμένες σε Java. Ένα από τα βασικότερα χαρακτηριστικά που παρέχει είναι το Inversion of Control. Σύμφωνα με αυτό η διαχείριση των Java αντικειμένων καθώς και η διάρκεια ζωής τους πραγματοποιείται από το framework. Τα Java αντικείμενα παραμετροποιούνται και ρυθμίζονται οι μεταξύ τους εξαρτήσεις μέσω του Spring. Στη σύνηθη περίπτωση αυτό γίνεται με χρήση xml αρχείων. Σε τέτοια xml αρχεία αποθηκεύεται το κάθε φίλτρο που δημιουργείται από το χρήστη. Κατά την εκτέλεση του φίλτρου η χρήση του framework δίνει τη δυνατότητα να φορτωθεί απευθείας από το xml αρχείο ο γράφος των Java αντικειμένων

που απαρτίζουν το φίλτρο.

3.4.2 Αναπαράσταση του φίλτρου ως γράφο και διάσχισή του κατά πλάτος

Το φίλτρο που παράγεται κάθε φορά από το χρήστη αναπαρίσταται ως κατευθυνόμενος γράφος από Java αντικείμενα. Το κάθε αντικείμενο αποτελεί ένα component της λύσης που περιέχει τη λογική που θα πρέπει να εκτελεστεί μέσα σε μία μέθοδο που ονομάζεται execute. Κατά την εκτέλεση ενός φίλτρου ο συγκεκριμένος γράφος διασχίζεται κατά πλάτος (Breadth First Search BFS). Κάθε κόμβος που επισκέπτεται εκτελεί τη λογική του λαμβάνοντας ως είσοδο ένα ή περισσότερα σύνολα από συνεγερμούς τα οποία έχουν προκύψει από την εκτέλεση των προηγούμενων κόμβων με εξαίρεση τον αρχικό του οποίου το σύνολο των συνεγερμών έχει προκύψει από ερώτημα στη βάση δεδομένων του Snort. Ακολουθεί ένα παράδειγμα διάσχισης ενός γράφου κατά BFS. Οι κόμβοι του παρακάτω γράφου θα προσπελασθούν με την ακόλουθη σειρά: A,B,C,D, E,F.



3.4.3 Ανίχνευση components στις βιβλιοθήκες της εφαρμογής

Μία βασική απαίτηση κατά τη σχεδίαση αποτέλεσε η δυνατότητα της επέκτασης της εφαρμογής. Είναι επιθυμητό ο χρήστης της εφαρμογής να έχει τη δυνατότητα να εισάγει στο σύστημα τα δικά του components για τα οποία έχει αναπτύξει τη λογική σε πηγαίο κώδικα Java. Αρχικά ορίστηκε η αφηρημένη (abstract) κλάση από την οποία θα πρέπει να κληρονομεί το κάθε component. Δοθέντος του ότι όλα τα components θα κληρονομούν μία συγκεκριμένη κλάση, κατά την εκκίνηση της εφαρμογής πραγματοποιείται ανίχνευση στο classpath και προσδιορίζονται όλες εκείνες οι κλάσεις που κληρονομούν από τη συγκεκριμένη αφηρημένη κλάση. Με αυτόν τον τρόπο φορτώνονται όλα τα διαθέσιμα components κατά την εκκίνηση της εφαρμογής. Ο όρος classpath αναφέρεται σε μία παράμετρο, η οποία μπορεί να προσδιοριστεί είτε από την κονσόλα εντολών είτε ως μεταβλητή περιβάλλοντος του λειτουργικού συστήματος, στην τοποθεσία στην οποία το JVM (Java Virtual Machine) θα αναζητήσει τις από το χρήστη ορισμένες κλάσεις.

3.4.4 Template Method Design Pattern

Για την επίτευξη της επεκτασιμότητας της εφαρμογής έγινε χρήση του design pattern Template Method. Στο συγκεκριμένο design pattern σχεδιάζεται η λογική ενός αλγορίθμου σε μία μέθοδο που αποκαλείται Template Method αφήνοντας ένα ή περισσότερα βήματα αφηρημένα. Το κάθε αφηρημένο βήμα απότελει μία αφηρημένη μέθοδο (abstract method). Οι υποκλάσεις υλοποιώντας τις προαναφερθείσες αφηρημένες μεθόδους είναι δυνατόν να προσδιορίσουν διακριτή συμπεριφορά για κάθε component. Μια μέθοδος πρότυπο καθορίζει το σκελετό προγράμματος ενός αλγορίθμου. Ένα ή περισσότερα από τα βήματα του αλγορίθμου μπορεί να υλοποιηθεί από υποκλάσεις να επιτρέποντας τη διαφορετική συμπεριφορά διασφαλίζοντας παράλληλα ότι ο πρωταρχικός αλγόριθμος ακολουθείται ακόμα. Στον αντικειμενοστρεφή προγραμματισμό, πρώτα μια κλάση δημιουργείται η οποία παρέχει τα βασικά βήματα ενός αλγορίθμου. Αυτά τα βήματα θα σχεδιαστούν ως αφηρημένες μεθόδοι. Αργότερα, υποκλάσεις υλοποιούν τις αφηρημένες μεθόδους για να εφαρμόσουν τη λογική των βημάτων. Έτσι, ο γενικός αλγόριθμος βρίσκεται σε ένα μέρος, αλλά τα συγκεκριμένα βήματα μπορεί να μεταβληθούν από τις υποκλάσεις.
Η μέθοδος template διαχειρίζεται έτσι την ευρύτερη εικόνα της εργασίας, και τις πιο εκλεπτυσμένες λεπτομέρειες υλοποίησης της επιλογής και της ακολουθίας των μεθόδων. Καλούνται αφηρημένες και μη μεθόδοι για την εκτέλεση της εργασίας. Οι μη-αφηρημένες μεθόδοι είναι απόλυτα ελεγχόμενες από τη μέθοδο template, αλλά τις αφηρημένες μέθοδοι, υλοποιούνται εξ'ολοκλήρου στις υποκλάσεις, παρέχουν την εκφραστική δύναμη του προτύπου και του βαθμού της ελευθερίας που αυτό προσφέρει. Ορισμένες ή όλες από τις

αφηρημένες μέθοδοι μπορεί να υλοποιηθούν σε μια υποκλάση, επιτρέποντας σε αυτόν που γράφει την υποκλάση να παρέχει συγκεκριμένη συμπεριφορά, με ελάχιστες τροποποιήσεις. Η μέθοδος template (που είναι μη-αφηρημένη) παραμένει αμετάβλητη, εξασφαλίζοντας ότι οι δευτερεύουσες μη αφηρημένες μέθοδοι και οι αφηρημένες μέθοδοι εκτελούνται με την ίδια ακολουθία. Η μέθοδος πρότυπο εμφανίζεται συχνά, τουλάχιστον στην απλούστερη περίπτωση, όπου μια μέθοδος καλεί μόνο μία αφηρημένη μέθοδο. Εάν κανείς χρησιμοποιεί μια αφηρημένη πολυμορφική μέθοδο, το συγκεκριμένο design pattern μπορεί να είναι μάλλον φυσική συνέπεια. Αυτό συμβαίνει επειδή μια μέθοδος καλώντας μια αφηρημένη ή πολυμορφική μέθοδο είναι απλώς ο λόγος για την ύπαρξη της αφηρημένης ή της πολυμορφικής μεθόδου. Το template design pattern μπορεί να χρησιμοποιηθεί για να προσθέσει άμεσα αξία στο λογισμικό ή με πρωταρχικό στόχο τις βελτιώσεις στο μέλλον.

3.4.5 Αποθήκευση του φίλτρου σαν xml και χειρισμός.

Όπως έχει αναφερθεί το φίλτρο που δημιουργείται από το χρήστη αποθηκεύεται σε μορφή xml. Αυτό δίνει τη δυνατότητα στο χρήστη να εισάγει και να εκτελέσει το φίλτρο που δημιούργησε μελλοντικά. Για να υλοποιηθεί η αποθήκευση του φίλτρου στη συγκεκριμένη μορφή απαιτείται ο κατάλληλος χειρισμός του xml. Για παράδειγμα απαιτείται η προσθαφαίρεση xml στοιχείων πριν ή μετά από συγκεκριμένα στοιχεία. Για τον προσδιορισμό αυτών των στοιχείων μέσα στο στο xml χρησιμοποιήθηκε το Xpath. Το Xpath αποτελεί μία γλώσσα για την έκφραση ερωτημάτων και τον προσδιορισμό στοιχείων μέσα σε ένα XML έγγραφο.

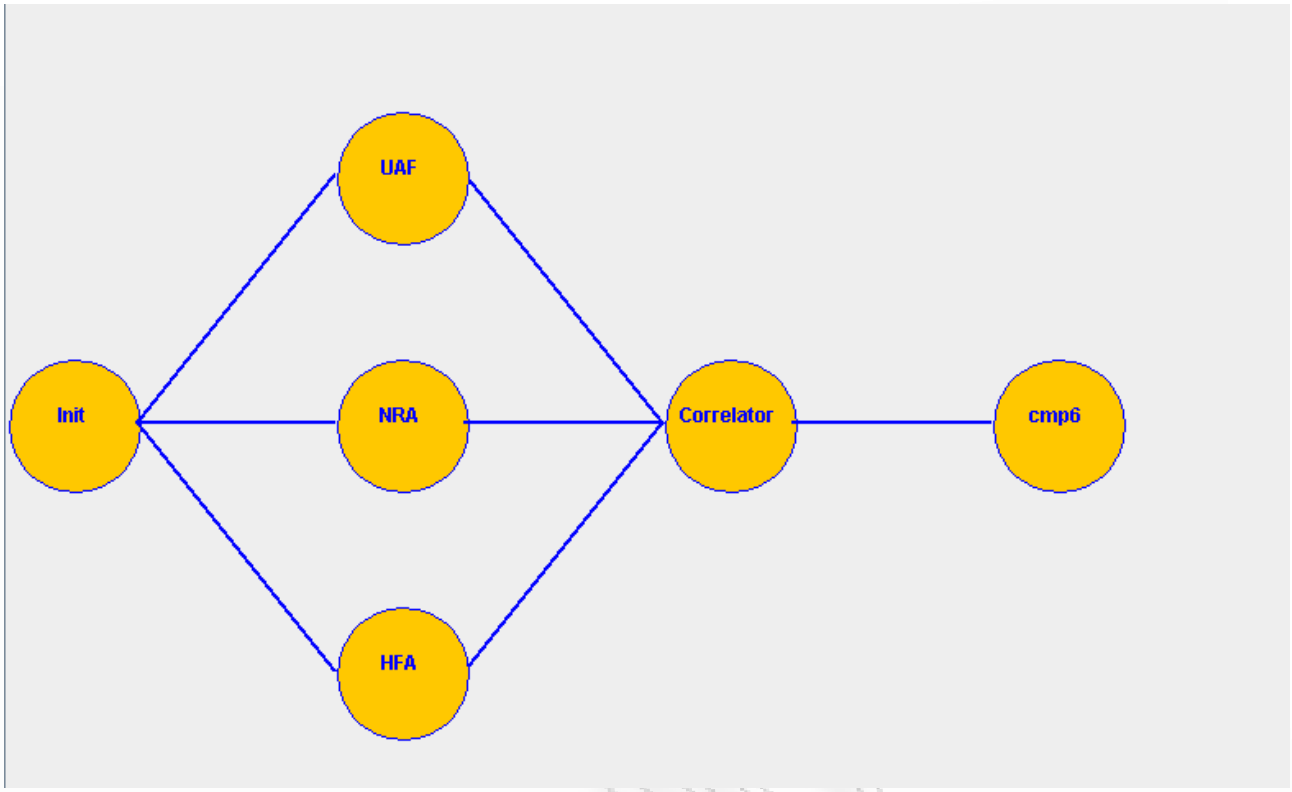
Ακολουθεί το xml κείμενο που αναπαριστά ένα φίλτρο που φαίνεται στην εικόνα που ακολουθεί το xml κείμενο.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-2.5.xsd">
<bean id="dataReader"class="com.tc.postProcessing.components.DbDataReader"/>
<bean id="initCmp"
class="com.tc.postProcessing.components.InitSolutionComponent">
        <property name="previous">
```

```
            <list/>
        </property>
        <property name="next">
            <list>
                <ref bean="cmp2"/>
                <ref bean="cmp3"/>
                <ref bean="cmp4"/>
            </list>
        </property>
        <property name="dataReader" ref="dataReader"/>
        <property name="evaluator" ref="evaluator"/>
    </bean>
    <bean id="cmp2"
class="com.tc.postProcessing.components.MockUpSolutionComponent">
        <property name="previous">
            <list>
                <ref bean="initCmp"/>
            </list>
        </property>
        <property name="next">
            <list>
                <ref bean="cmp5"/>
            </list>
        </property>
        <property name="configMap">
            <map>
                <entry key="name" value="NRA"/>
            </map>
        </property>
    </bean>
<beanid="cmp3"
class="com.tc.postProcessing.components.MockUpSolutionComponent">
        <property name="previous">
            <list>
                <ref bean="initCmp"/>
            </list>
        </property>
        <property name="next">
            <list>
                <ref bean="cmp5"/>
            </list>
```

```
        </property>
        <property name="configMap">
            <map>
                <entry key="name" value="HFA"/>
            </map>
        </property>
    </bean>
    <bean id="cmp4"
class="com.tc.postProcessing.components.MockUpSolutionComponent">
        <property name="previous">
            <list>
                <ref bean="initCmp"/>
            </list>
        </property>
        <property name="next">
            <list>
                <ref bean="cmp5"/>
            </list>
        </property>
        <property name="configMap">
            <map>
                <entry key="name" value="UAF"/>
            </map>
        </property>
    </bean>
    <bean id="cmp5"
class="com.tc.postProcessing.components.MockUpSolutionComponent">
        <property name="previous">
            <list>
                <ref bean="cmp2"/>
                <ref bean="cmp4"/>
                <ref bean="cmp3"/>
            </list>
        </property>
        <property name="next">
            <list>
                <ref bean="cmp6"/>
            </list>
        </property>
        <property name="configMap">
            <map>
```

```
                <entry key="name" value="Correlator"/>
            </map>
        </property>
    </bean>
    <bean id="cmp6"
class="com.tc.postProcessing.components.DefaultTerminalComponent">
        <property name="previous">
            <list>
                <ref bean="cmp5"/>
            </list>
        </property>
        <property name="next">
            <list/>
        </property>
    </bean>
    <bean id="evaluator"
class="com.tc.postProcessing.components.EvaluatorImpl">
        <property name="eventsFileList">
            <list>
                <value>C:\Users\tc\Documents\trueEvents.xml</value>
            </list>
        </property>
    </bean>
</beans>
```



3.4.6 Εγκατάσταση και παραμετροποίηση του συστήματος.

Για την εγκατάσταση του συστήματος απαιτείται μόνο η αντιγραφή του εκτελέσιμου jar αρχείου στο δίσκο μαζί με τις απαιτούμενες βιβλιοθήκες τοποθετημένες σε ένα φάκελο με όνομα lib. Ο συγκεκριμένος φάκελος θα πρέπει να τοποθετηθεί στον ίδιο φάκελο με το εκτελέσιμο αρχείο της εφαρμογής. Επιπροσθέτως θα πρέπει να τοποθετηθούν στο συγκεκριμένο φάκελο και τρία επιπλέον αρχεία :

- hibernate.template.cfg.xml
  Πρότυπο αρχείο για τη σύνδεση με τη βάση δεδομένων του Snort.
- solutionConfigurationTemplate.xml
  Πρότυπο αρχείο για την αποθήκευση του φίλτρου σε μορφή xml.
- conf.txt
  Αρχείο που περιέχει παραμέτρους που αφορούν την τοποθεσία αποθήκευσης των αποτελεσμάτων της εκτέλεσης του φίλτρου, τα ονόματα των πακέτων όπου αναμένεται να βρίσκονται τα components για τη δημιουργία του φιλτρου – λύσης.

3.4.7 Εισαγωγή components στο σύστημα ορισμένων από το χρήστη (User defined components).

Το σύστημα παρέχει τη δυνατότητα στο χρήστη να ορίσει τα δικά του components και να τα χρησιμοποιήσει για τη δημιουργία ενός φίλτρου. Για την υλοποίηση του ανωτέρω θα πρέπει να γίνουν τα ακόλουθα βήματα:

- Δημιουργία νέων components τα οποία κληρονομούν την abstract κλάση AbstractSolutionComponent.
- Υλοποίηση της μεθόδου execute() η οποία δέχεται ως είσοδο μία ή περισσότερες λίστες από alerts και παράγει ως έξοδο μία λίστα από Alerts.
- Τα νέα components να εισαχθούν σε ένα αρχείο τύπου jar (Java Archive).
- Στην περίπτωση όπου το νέο component δέχεται παραμέτρους θα πρέπει να γίνει annotate η κλάση του με το annotation ConfiguredBy. Το τελευταίο δέχεται ως όρισμα ένα String το οποίο θα πρέπει να περιέχει το όνομα της κλάσης η οποία θα υλοποιεί τη διεπαφή ISolutionComponentDialogConfig και θα είναι υπεύθυνη για το όρισμα των συγκεκριμένων παραμέτρων στη μορφή key-value. Ο χρήστης επομένως θα πρέπει να παρέχει μία υλοποίηση της διεπαφής ISolutionComponentDialogConfig. Οι συγκεκριμένοι παράμετροι αποθηκεύονται στο πεδίο configMap του κάθε component (το συγκεκριμένο πεδίο κληρονομείται από το AbstractSolutionComponent).
- Ένταξη στο classpath του jar αρχείου που περιέχει να νέα components κατά την εκτέλεση της εφαρμογής. Για αυτό το βήμα θα πρέπει να αντιγραφεί το νέο αρχείο στο φάκελο lib και στο αρχείο META-INF/MANIFEST.MF, που βρίσκεται μέσα στο java archived αρχείο της εφαρμογής IDSPostProcessingLib.jar, να προστεθεί η εγγραφή "lib/libraryName.jar" στο γνώρισμα Class-Path. Το libraryName θα πρέπει να αντικατασταθεί με το πραγματικό όνομα της βιβλιοθήκης.

4. Βιβλιογραφία

[1] CERT - Carnegie Mellon University. (2006), Vulnerability discovery: Bridging the gap between analysis and engineering. [accessed: 2009, 15 Nov]. Available: www.cert.org/archive/pdf/CERTCC_Vulnerability_Discovery.pdf

[2] Metasploit LLC. (2008, 19 Nov). Metasploit framework. [accessed: 2009, 05 Aug]. Available: http://www.metasploit.com/framework/

[3] G. Lyon. (1997, Sept). Nmap. [accessed: 2009, 02 Oct]. Available: http://nmap.org/

[4] M. Roesch. (1998), Snort. [accessed: 2009, 09 Aug]. Available: http://www.snort.org/

[5] D. E. Denning, "An intrusion-detection model," in *IEEE Symposium on Security and Privacy,* 1986, pp. 118-133.

[6] R. U. Rehman and N. Regina, *Intrusion Detection with SNORT (Bruce Perens' Open Source Series): Advanced IDS Techniques using Snort, Apache, MySQL, PHP, and ACID,* Pearson Education, 2003.

[7] Insecure.org. Top 5 Intrusion Detection Systems [Online]. http://sectools.org/ids.html [accessed: 2009, 04 Nov]

[8] Y. Zhang, W. Lee and Y. Huang, "Intrusion detection techniques for mobile wireless networks," *Wirel.Netw.,* vol. 9, pp. 545-556, 2003.

[9] The Prelude Team. PreludeIDS. Available: https://dev.prelude-ids.com/

[10] R. Goss, M. Botha and R. von Solms, "Utilizing fuzzy logic and neural networks for effective, preventative intrusion detection in a wireless environment," in *SAICSIT* 117 *'07: Proceedings of the 2007 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries,* 2007, pp. 29-35.

[11] T. Oetiker. rrdtool [Online]. http://oss.oetiker.ch/rrdtool/

[12] SRI International, "A Real-Time Intrusion-Detection Expert System (IDES)," 1992, 28 Feb.

[13] Y. Zhang, W. Lee and Y. Huang, "Intrusion detection techniques for mobile wireless networks," *Wirel.Netw.,* vol. 9, pp. 545-556, 2003.

[14] G. P. Spathoulas and S. K. Katsikas, "Reducing false positives in intrusion detection systems," *Comput. Secur.,* 2009.

[15] K. J. Cox., *Managing Security with Snort and IDS Tools,* Sebastopol CA USA: O'Reilly & Associates Inc, 2004.

[16] Insecure.org. Top 5 Intrusion Detection Systems [Online]. http://sectools.org/ids.html

[17] D. B. Cid. OSSEC v 2.2. [accessed: 2009, 06 Nov]. Available: http://www.ossec.net/

[18] V. Jacobson, C. Leres and S. McCanne. (1987), Tcpdump. [accessed: 2009, 06 Nov]. Available: http://www.tcpdump.org/

[19] G. Combs. (1998), Wireshark. Available: http://www.wireshark.org/
[20] A.Yee, "Marking False Positives Go Away",ComputerWorld, www.computerworld.com, 2006
[21] E. Schultz, J.Mellander and D.Peterson "The MS-SQL Slammer Worm", Network Security, V(2003), ISS(3) , pp: 10-14
[22]V.Paxson, "Bro: A System for Detecting Network Intruders in Real-Time", Computer Networks, V(31) Iss (23-24), pp:2435-2463, 14 December 1999
[23] Anderson et al, "Detecting Unusual Program Behaviour Using the Statistical Component of the Next Generation Intrusion Detection Expert System (NIDES)," Computer Science Laboratory SRI-CS2 95-06 , May 1995
[24] P.Neumann and P.Porras, "Experiment with EMERALD to Date", Proceeding 1st USENIX Workshop and Intrusion Detection and Network Monitoring , pp:73-80, 1999
[25] J.M.Estevez-Tapiador, P.Garcia-Teodoro, J.E.Diaz-Verdejo, "Measuring Normality in HTTP Traffic for Anomaly-Based Intrusion Detection", Computer Networks V(45), 175-193, (2004)
[26] G.Benrit, "Application of Markov Chains in an Interactive Information Retrieval System" Information Processing & Management, Vol(41), Iss(4), pp: 843-857,July 2005
[27] Pravesh Gaonjur, N.Z. Tarapore, and S.G. PukaleUsing Neuro-Fuzzy Techniques to Reduce False Alerts in IDS Pravesh Gaonjur, N.Z. Tarapore, and S.G. Pukale
[28] Frank, J., "Artificial Intelligence and Intrusion Detection: Current and Future Directions", Proceedings of the 17th National Computer Security Conference, October 1994.
[29] Alshammari Riyad, Sonamthiang Sumalee, Teimouri Mohsen, Riordan Denis, "Using Neuro-Fuzzy Approach to Reduce False Positive Alerts",
Communication Networks and Services Research, 2007. CNSR '07. Fifth Annual Conference IEEE Press, pg 345 – 349
[30] Nauck D., Nauck U., and Kruse R., "NEFCLASS for JAVA New Learning Algorithms", Proceedings of Fuzzy Information Processing Society NAFIPS) 18th International Conference of the North American,pp. 472-476. July 1999.
[31] Klaus Julisch. Clustering intrusion detection alarms to support root cause analysis. ACM Transactions on Information and System Security (TISSEC), 6(4):443{471, 2003.
[32] Klaus Julisch. Using Root Cause Analysis to Handle Intrusion Detection Alarms. PhD thesis, University of Dortmund, Germany, 2003.
[33]. J. Han, Y. Cai, and N. Cercone. Data-Driven Discovery of Quantitative Rules in Relational Databases. IEEE Transactions on Knowledge and Data Engineering.
[34]. Jiawei Han, Yandong Cai, and Nick Cercone. Knowledge Discovery in Databases: An Attribute-Oriented Approach. In Proceedings 18th International Conference on Very Large Databases (VLDB), Vancouver, Canada, Aug. 1992.