Αλεξάνδρειο Τεχνολογικό Ίδρυμα Θεσσαλονίκης

Τμήμα Μηχανικών
Πληροφορικής ΑΤΕΙΘ

ΠΤΥΧΙΑΚΉ ΕΡΓΑΣΙΑ

«Ανάπτυξη στοιχείων λογισμικού (plug-in) για την ενσωμάτωσή τους σε υπηρεσίες ηλεκτρονικού ταχυδρομείου τύπου Thunderbird και Outlook για το σύστημα «Arachnoid»»

Βρύνιος Παναγιώτης

04/2654

Επιβλέπων: Ηλιούδης Χρήστος

Θεσσαλονίκη, 2014

Ευχαριστίες

Για την υλοποίηση της παρούσας Πτυχιακής Εργασίας θέλω να ευχαριστήσω ιδιαιτέρως τον επιβλέποντα καθηγητή κ. Ηλιούδη Χρήστο για την καθοδήγησή του κατά τη συγγραφή της εργασίας και τις χρήσιμες παρατηρήσεις του.

Τους γονείς μου, Χρήστο και Ευαγγελία, χωρίς την παρότρυνση των οποίων δε θα ήταν δυνατή η ολοκλήρωση αυτής της Πτυχιακής και την αδελφή μου Δήμητρα για συνεχή συμπαράσταση και την βοήθεια.

Τέλος, θα ήθελα να ευχαριστήσω θερμά την Φυγαλία Διαμαντοπούλου για την υποστήριξή της καθ' όλη την διάρκεια της συγγραφής και την πολύτιμη βοήθεια της στην διαδικασία των πειραμάτων.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Περίληψη

Living in the technological century, we encounter more and more issues related to technology. One of these is the size that the problem of the unwanted emails has taken. One of the most successful attempts to solve the above problem is the creation and use of filters that have the potential to discern the wanted from the unwanted emails. In this assignment we thoroughly present spamming, the ways in which it rapidly spreads through the internet as well as the mechanisms that were developed in order to cope with this expansive tendency of spamming. The Mozilla solution is being studied in terms of the organization of all emails, Thunderbird too, and a filter extension is being coded using development tools provided to us.

Καθώς διανύουμε τον αιώνα της τεχνολογίας εμφανίζονται ολοένα και περισσότερα προβλήματα που σχετίζονται με αυτήν. Ένα από αυτά είναι η διάσταση που έχει λάβει το πρόβλημα των ανεπιθύμητων ηλεκτρονικών μηνυμάτων. Μία από τις πιο επιτυχημένες προσπάθειες επίλυσης του άνωθεν προβλήματος είναι η δημιουργία και χρήση φίλτρων, τα οποία έχουν την δυνατότητα να ξεχωρίζουν τα επιθυμητά από τα ανεπιθύμητα ηλεκτρονικά μηνύματα. Σε αυτή την εργασία παρουσιάζεται αναλυτικά το Spam, οι τρόποι με τους οποίους διαδίδεται ταχύτατα στο διαδίκτυο, καθώς και οι μηχανισμοί που έχουν αναπτυχθεί για να καταπολεμηθεί αυτή η επεκτατική τάση της ανεπιθύμητης αλληλογραφίας. Μελετούμε την λύση της Mozilla για την οργάνωση της ηλεκτρονικής αλληλογραφίας, το Thunderbird, και αναπτύσσουμε μία επέκταση φίλτρου για αυτό χρησιμοποιώντας τα εργαλεία ανάπτυξης που μας παρέχει.

Εισαγωγή

Με την αυξανόμενη σημασία του Διαδικτυακού εμπορίου, τα ανεπιθύμητα μηνύματα αποτελούν ένα πρόβλημα το οποίο καλούνται να αντιμετωπίσουν καθημερινά οι χρήστες του διαδικτύου. Διάφοροι άλλοι όροι για αυτά είναι «*Junk*» και «*Bulk*» μηνύματα ή εκούσια ηλεκτρονικά μηνύματα, αλλά ο πιο ευρέως διαδεδομένος όρος είναι ο «*Spam*». Εστιάζοντας στις υπηρεσίες ηλεκτρονικής αλληλογραφίας γίνεται άμεσα αντιληπτό ότι τα αξιόπιστα μηνύματα ηλεκτρονικού ταχυδρομείου ενός χρήστη, χάνονται μέσα στον πλήθος των *Spam* μηνυμάτων που βομβαρδίζουν τα Εισερχόμενά του. Επιπλέον, η αυξημένη χρήση πόρων για την μεταφορά των δεδομένων, τα καθυστερημένα μηνύματα και η χαμένη παραγωγικότητα αποτελούν τεράστια προβλήματα τα οποία καλείται να αντιμετωπίσει οποιοσδήποτε παρέχει υπηρεσίες ηλεκτρονικού ταχυδρομείου. Τα ανεπιθύμητα μηνύματα αποτελούν παραβίαση της ιδιωτικής ζωής και πολύ συχνά αναγκάζουν τον χρήστη να δει ανεπιθύμητο υλικό ακόμα και πορνογραφία. Εκτός από την οπτική παρενόχληση υπάρχουν και τεχνικά προβλήματα τα οποία προέρχονται από αυτό. Πολύ συχνά το *Spam* είναι επικίνδυνο, περιέχοντας ιούς, δούρειους ίππους και άλλα επικίνδυνα λογισμικά, προκαλώντας κενά ασφαλείας σε ηλεκτρονικούς υπολογιστές και δίκτυα. Το *Spam* είναι μία από τις κύριες μεθόδους εξαπόλυσης επιθέσεων «*phishing*» στις οποίες ο χρήστης ξεγελάτε ώστε να δώσει ευαίσθητα δεδομένα σε τρίτους.

Οι περισσότεροι χρήστες πιστεύουν ότι η εξάλειψη του *Spam* είναι ευθύνη του Παρόχου Υπηρεσιών Διαδικτύου. Δυστυχώς, όμως, δεν υπάρχει τρόπος εξάλειψης του παρά μόνο διάφορα μέσα για την αντιμετώπισή του. Σε αυτό το κομμάτι, ενεργό ρόλο έχουν και οι ίδιοι οι χρήστες. Είναι πολύ σημαντικό να γνωρίζουν τι είναι *Spam*, πως λειτουργεί, ποια είναι τα προγράμματα προστασίας, που δρουν συμπληρωματικά στα ήδη υπάρχοντα συστήματα αντιμετώπισης των παρόχων, και πώς να τα εγκαταστήσουν. Ένα από τα δωρεάν προγράμματα που βοηθά τον χρήστη να οργανώσει τα Εισερχόμενα μηνύματα του λογαριασμού ηλεκτρονικού ταχυδρομείου είναι το *Mozilla Thunderbird*. Μέσα στο πρόγραμμα λειτουργεί ήδη υπάρχον φίλτρο προστασίας ανεπιθύμητων μηνυμάτων, δίνεται όμως η δυνατότητα στον χρήστη να εγκαταστήσει «Addons» ή αλλιώς «Επεκτάσεις», οι οποίες δρουν είτε συμπληρωματικά είτε αντικαθιστώντας το

προ-εγκατεστημένο φίλτρο. Στα πλαίσια αυτής της Πτυχιακής αναπτύχτηκε μια τέτοια επέκταση για να βοηθήσει τον χρήστη να αντιμετωπίσει το *Spam* πιο αποτελεσματικά.

Η επέκταση με το όνομα «*Arachnoid*» έχει σαν στόχο έχει να ανιχνεύει και να απομακρύνει όλα τα μηνύματα *Spam* ενός χρήστη καθώς αυτά φθάνουν στον *Thunderbird*, κάνοντας την εμπειρία χρήσης του ηλεκτρονικού ταχυδρομείου πιο ευχάριστη και πιο αξιόπιστη. Λειτουργώντας στο παρασκήνιο και απαιτώντας ελάχιστες ρυθμίσεις από πλευράς χρήστη, η επέκταση μειώνει δραστικά τα Εισερχόμενα *Spam*. Σε πολλές περιπτώσεις κάποια μηνύματα *Spam* καμουφλάρονται εύκολα σαν συνηθισμένα μηνύματα και περνούν απαρατήρητα. Αναγνωρίζοντας λέξεις κλειδιά στον τίτλο του μηνύματος ή και ακόμα στο κυρίως κείμενο, ο χρήστης μπορεί να τις προσθέσει στην λίστα του φίλτρου μέσα από τις ρυθμίσεις, εμπλουτίζοντας με αυτό τον τρόπο τις δυνατότητες του φίλτρου. Επιπλέον, το φίλτρο λειτουργεί με βαθμό ευαισθησίας, στις λέξεις κλειδιά, ορισμένο από τον ίδιο τον χρήστη. Αλλάζοντας, τον βαθμό ευαισθησίας βελτιώνεται η αποτελεσματικότητα της επέκτασης. Παράλληλα με την ανίχνευση, το φίλτρο διαθέτει λίστες αποκλεισμού επιλεγμένων διευθύνσεων και Domains από τα οποία ο χρήστης δεν θέλει να λαμβάνει κανένα μήνυμα, είτε *Spam* είτε όχι.

Σκοπός της Πτυχιακής Εργασίας

Η παρούσα εργασία διαπραγματεύεται το θέμα του *Spam* και των τρόπων αντιμετώπισής του. Συγκεκριμένα στο πρώτο κεφάλαιο, μαθαίνουμε από πού προήλθε ο όρος «*Spam*» και πως ορίζεται στην Ελληνική Γλώσσα. Αναλύνται τα χαρακτηριστικά του, το πρόβλημα του *Spam* γενικότερα όσον αφορά στο τρόπο διάδοσης του καθώς και ο τρόπος δράσης των *Spammers*. Με αυτό τον τρόπο ο χρήστης έχει την δυνατότητα να ενημερωθεί σχετικά με το μέγεθος του προβλήματος και τις επιπτώσεις του στην καθημερινότητα του. Τέλος, ο χρήστης μαθαίνει πως μπορεί να αποφύγει και όπου αυτό δεν είναι εφικτό, να αντιμετωπίσει το εισερχόμενο *Spam*.

Έπειτα, στο δεύτερο κεφάλαιο παρουσιάζονται οι μηχανισμοί αντιμετώπισης που χρησιμοποιούνται ενάντια στα *spam*-μηνύματα ηλεκτρονικού ταχυδρομείου, όπως ο απλοϊκός ταξινομητής *Bayes* και οι λίστες φιλτραρίσματος διευθύνσεων αποστολέων. Στη συνέχεια παρουσιάζονται τα διάφορα χαρακτηριστικά των τεχνικών αντιμετώπισης και αναλύεται ο τρόπος λειτουργίας τους.

Στο τρίτο κεφάλαιο, σχεδιάζεται ένα φίλτρο αντιμετώπισης ανεπιθύμητων μηνυμάτων. Το φίλτρο προγραμματίστηκε υπό την μορφή επέκτασης για το *Mozilla Thunderbird* και αναπτύχθηκε κάνοντας χρήση των εργαλείων ανάπτυξης και της βιβλιοθήκης γνώσεων της *Mozilla*. Έπειτα, αναλύονται οι λειτουργικές και οι μη λειτουργικές απαιτήσεις της εφαρμογής και δίνονται οι περιπτώσεις χρήσης στις οποίες πρέπει να ανταπεξέλθει η εφαρμογή.

Στο τέταρτο κεφάλαιο γίνεται παρουσίαση της επέκτασης «*Arachnoid*» και του τεχνολογικού περιβάλλοντος ανάπτυξης της. Ο χρήστης έχει την δυνατότητα να κατανοήσει τι είναι μία επέκταση, τι προσπαθεί να επιτύχει και από τι αποτελείται. Για την καλύτερη κατανόηση του χρήστη, γίνεται παρουσίαση της δομής της επέκτασης και περιγράφονται αναλυτικά τα διάφορα αρχεία που την απαρτίζουν. Τέλος, αναφέρονται εκτενέστερα οι γλώσσες που χρησιμοποιήθηκαν στην ανάπτυξη του προγράμματος και τα εργαλεία αποσφαλμάτωσης που απαιτήθηκαν να σιγουρευτεί η ορθή λειτουργία του.

Στο πέμπτο κεφάλαιο σχολιάζεται η υλοποίηση της εφαρμογής και στην συνέχεια γίνεται χρήση εικόνων για την οπτική αναπαράσταση των λειτουργιών που αναπτύχθηκαν. Ο χρήστης ενημερώνεται για της τελικές δυνατότητες του φίλτρου, την παραμετροποίηση που διαθέτει και τον κώδικα που τις υλοποιεί.

Στο έκτο κεφάλαιο παρουσιάζονται τα κριτήρια κατά τα οποία ένα μήνυμα ηλεκτρονικού ταχυδρομείου κατατάσσεται ως *Spam* και οι τεχνικές αξιολόγησης των διάφορων μεθόδων αντιμετώπισης. Πιο συγκεκριμένα, παρατηρούμε από τι αποτελείται ένα ολοκληρωμένο σύστημα αξιολόγησης ενός ταξινομητή μηνυμάτων, περιγράφοντας τι είναι μια συλλογή μηνυμάτων δοκιμών, ποιες συλλογές είναι διαθέσιμες σήμερα και ποια τα πλεονεκτήματα και μειονεκτήματα αυτών. Επίσης, γίνεται εκτενής αναφορά στην διαδικασία εκτέλεσης μια δοκιμής, σχετικά με το τι απαιτείται και ποιες μετρήσεις είναι αυτές που τελικά χαρακτηρίζουν την απόδοση ενός ταξινομητή. Στη συνέχεια, περιγράφονται οι τρόποι με τους οποίους γίνεται η αναπαράσταση των αποτελεσμάτων και η σύγκριση των διάφορων λύσεων ταξινόμησης που μελετούνται. Τέλος, δίνεται η

περιγραφή του πειράματος σχετικά με τον τρόπο που θα αξιολογηθεί το φίλτρο που αναπτύχθηκε στην παρούσα εργασία, και εξάγονται τα αποτελέσματα με χρήση πινάκων που οδηγούν στα συμπεράσματα της πειραματικής διαδικασίας.

Στο τελευταίο κεφάλαιο γίνεται αναφορά στα μελλοντικά σχέδια ως προς την λειτουργικότητα και τις υπηρεσίες που θα μπορούσαν να εφαρμοστούν στην επέκταση «*Arachnoid*» και που θα βελτίωναν την απόδοση της.

1 *Το Πρόβλημα Spamming*

1.1 Φαινόμενο Spamming

Ηλεκτρονικό *Spamming* είναι η ανεπιθύμητη και απρόσκλητη μαζική αποστολή πληθώρα μηνυμάτων που απευθύνονται σε ένα σύνολο παραληπτών του διαδικτύου [2].

Ο όρος *spam* έκανε την εμφάνισή του στη Μεγάλη Βρετανία, τη δεκαετία του 1960 και προήλθε από το εμπορικό όνομα αμερικανικού προϊόντος κρέατος σε κονσέρβα, το οποίο εισαγόταν στη χώρα σε μεγάλες ποσότητες. Το 1970, επαναλαμβανόταν στην εκπομπή «Το ιπτάμενο τσίρκο των Μόντυ Πάιθον», ένα τραγούδι των «*Monty Python*», στο πλαίσιο του σκετς τους με όνομα «*Spam*», με μόνα λόγια το «spam spam spam...ωραίο spam...εξαίρετο spam». Έπειτα, στη δεκαετία του 1980 η έννοιά του ήταν συνυφασμένη με τον όρο Sales Promotion And Marketing (Προώθηση Πωλήσεων και Μάρκετινγκ) από τις εταιρείες αποστολής τέτοιων μηνυμάτων, εξού και το ακρωνύμιο S.P.A.M. Αργότερα, το New Oxford Dictionary of English, και συγκεκριμένα το 1998, συμπεριέλαβε τον συγκεκριμένο όρο σημαίνοντας «άσχετα ή μη αποδεκτά μηνύματα που στέλνονται στον Ίντερνετ σε μεγάλο αριθμό ομάδων ειδήσεων ή χρηστών [1].

Το *Spam* λαμβάνονται ακούσια από τους παραλήπτες μέσα από το ηλεκτρονικό τους γραμματοκιβώτιο, με τη μορφή ενημερωτικών ή διαφημιστικών μηνυμάτων για προϊόντα ή υπηρεσίες. Το χαμηλό κόστος αποστολής τους δικαιολογεί τον μεγάλο αριθμό αποδεκτών, στον οποίο απευθύνεται. Αυτό το παγκόσμιο φαινόμενο υπολογίζεται πως μόνο το 2011 έχουν ληφθεί επτά τρισεκατομμύρια ανεπιθύμητα μηνύματα *spam*, παρόλο που σε αρκετές χώρες η αποστολή *spam* θεωρείται ποινικό αδίκημα.

Στην Ελληνική γλώσσα, το συγκεκριμένο είδος αλληλογραφίας αποδίδεται με τους όρους **απρόκλητη** ή **ανεπιθύμητη αλληλογραφία.** Τα κυριότερα χαρακτηριστικά του *Spam* θεωρούνται τα ακόλουθα:

- **Απρόκλητο:** Η επικοινωνία που επιχειρείται είναι απρόκλητη, με την έννοια ότι δεν υπάρχει κάποια σχέση μεταξύ παραλήπτη και αποστολέα που θα δικαιολογούσε ή θα προκαλούσε την επικοινωνία αυτή.
- **Εμπορικό:** Πολλές φορές το *spam* αφορά την αποστολή μηνυμάτων εμπορικού σκοπού με σκοπό την προβολή και την διαφήμιση προϊόντων και υπηρεσιών για την προσέλκυση πελατών και την πραγματοποίηση πωλήσεων.
- **Μαζικό:** Το *spam* συνίσταται στην μαζική αποστολή μεγάλων ποσοτήτων ίδιων ή ελαφρά διαφοροποιημένων μηνυμάτων από τον αποστολέα σε ένα πλήθος παραληπτών. [5]

1.2 Μέθοδοι Διάδοσης

Πρωταρχική τακτική που ακολουθεί μια επικερδής επιχείρηση *spam* είναι να συλλέξει πραγματικές διευθύνσεις ηλεκτρονικού ταχυδρομείου πελατών. Οι διαθέσιμες πληροφορίες για τον τρόπο συλλογής αυτών των διευθύνσεων είναι ελλιπείς. Ωστόσο, φαίνεται πως η συγκομιδή πραγματικών διευθύνσεων γίνεται από mailing list archives, ροές usenet, δικτυακούς τόπους κτλ., και συγκεκριμένα, είτε άμεσα από τα συστήματα των *spammer*, είτε μέσω ενός συνόλου συστημάτων *«botnet»*[1], αθώων χρηστών, που είναι κάτω από τον έλεγχο του *spammer* [Αντωνόπουλος, 2009].

Ο *spammer* μπορεί να ενισχύσει αυτή τη μέθοδο εξαπλώνοντας κακόβουλα προγράμματα όπως ιούς, δούρειους ίππους και rootkits. Αυτά τα προγράμματα μπορούν να διαβάσουν τα βιβλία διευθύνσεων, όπως και τα παλιότερα μηνύματα, ανυποψίαστων χρηστών για τη συλλογή νέων διευθύνσεων που αντιστοιχούν σε πραγματικούς χρήστες, ενώ παράλληλα δίνουν τον έλεγχο αυτών των συστημάτων στο *spammer* διευρύνοντας του μέγεθος του *botnet*. Μια προηγμένη μέθοδος είναι ο συνδυασμός ενός μέρους μιας διεύθυνσης που υπάρχει ήδη στη βάση του *spammer* με άλλα *domains*.

[1] Ως botnet ορίζεται ένα δίκτυο υπολογιστών, το οποίο ελέγχεται εξ αποστάσεως από τον λεγόμενο botmaster χωρίς τη γνώση ή την έγκριση των κατόχων των μεμονωμένων υπολογιστών. Τα σύνολα αυτά συχνά χρησιμοποιούνται για κακόβουλες ενέργειες, όπως κατανεμημένες επιθέσεις άρνησης υπηρεσίας DDoS. [4]

Παραδείγματος χάριν, έστω ότι η διεύθυνση *john@example.com* είναι πραγματική. Τότε, είναι αρκετά πιθανό να υπάρχει και η διεύθυνση *john@example2.com* και να αντιστοιχεί σε ένα πραγματικό χρήστη. Η τεχνική αυτή καλείται επίθεση *«λεξικού»* ή *«Rumpelstiltskin»* επίθεση. Αν η επίθεση αυτή είναι επιτυχής, τότε η καινούρια διεύθυνση (john@example2.com) καταγράφεται στη βάση δεδομένων του *spammer*.

Μια άλλη γνωστή μέθοδος είναι η αγορά λιστών διευθύνσεων ηλεκτρονικού ταχυδρομείου από παράνομες πηγές. Οι λίστες αυτές περιέχουν διευθύνσεις που έχουν συλλεχθεί με ένα συνδυασμό από τις παραπάνω τεχνικές, αλλά και άλλων αγνώστων, και το κόστος καθορίζεται από την ποιότητά τους, δηλαδή από το ποσοστό πραγματικών διευθύνσεων που περιέχουν.

Από τη στιγμή που ο *spammer* έχει συλλέξει σημαντικό αριθμό πραγματικών διευθύνσεων, μπορεί να αρχίσει την αποστολή των μηνυμάτων. Για την αποθήκευση των μηνυμάτων και των αποδεκτών, αλλά και για την αποστολή των μηνυμάτων, απαιτούνται υπολογιστικοί πόροι, όπως και *bandwidth*. Το *bandwidth* θεωρείται ο πιο σημαντικός παράγοντας για την όσο το δυνατόν ταχύτερη αποστολή μεγάλου όγκου μηνυμάτων. Ο *spammer* έχει δύο επιλογές, είτε την αγορά *bandwidth* από κάποιον πάροχο, είτε τη χρήση ενός *botnet*. Η πρώτη περίπτωση είναι η πιο αποτελεσματική, καθώς ο αποστολέας έχει τον πλήρη έλεγχο της αποστολής των μηνυμάτων. Στην περίπτωση του *botnet,* παρότι είναι πιο εύκολη η αποστολή μεγάλου όγκου μηνυμάτων χωρίς να είναι άμεσα συνδεδεμένα με τον αποστολέα, είναι σχεδόν αδύνατη η παραμετροποίηση σε περίπτωση που αλλάξει η πολιτική που χρησιμοποιούν οι πάροχοι των παραληπτών. Χαρακτηριστικό παράδειγμα είναι αν κάποιος πάροχος αλλάξει τη θύρα 25, η οποία χρησιμοποιείται συνήθως για τη λήψη ηλεκτρονικού ταχυδρομείου [Αντωνόπουλος, 2009].

1.3 Επιπτώσεις των *Spam*

Ιδιαίτερο πρόβλημα αντιμετωπίζουν οι χρήστες που χρησιμοποιούν για μεγάλα διαστήματα της ημέρας το ηλεκτρονικό ταχυδρομείο και είναι αναγκασμένοι να σβήνουν όλη αυτή την ανεπιθύμητη αλληλογραφία. Τα μηνύματα αυτά για αρκετούς

χρήστες φθάνουν να είναι πολλές φορές εκατοντάδες σε μια ημέρα. Η αναγκαιότητα για την αντιμετώπιση του Spam εντοπίζεται στα ακόλουθα σημεία:

- **Είναι φαινόμενο δυσάρεστο, ενοχλητικό και απαράδεκτο από τους παραλήπτες.** Πολλές φορές προβάλει αμφίβολης ποιότητας προϊόντα και υπηρεσίες, ενώ συνηθισμένη είναι η προβολή ύποπτων οικονομικών δραστηριοτήτων τύπου πυραμίδων κλπ. Άλλα μηνύματα πιθανόν να περιέχουν ή διαφημίζουν σεξουαλικό περιεχόμενο.
- **Οδηγεί σε κατάχρηση πόρων του Διαδικτύου.** Η κατάχρηση αυτή επιβαρύνει τα δίκτυα με κατανάλωση εύρους ζώνης, αποθηκευτικών και υπολογιστικών πόρων στα κεντρικά συστήματα διανομής αλληλογραφίας (*e-mail servers*). Αντίστοιχα προβλήματα προκαλεί στην πρόσβαση και στα συστήματα των χρηστών.
- **Θέτει σε κίνδυνο την ασφάλεια και την αξιοπιστία του διαδικτύου:** Οι *spammers* βρίσκονται σε συνεχή αναζήτηση συστημάτων, τα οποία θα μπορούσαν να χρησιμοποιήσουν για την αποστολή των μηνυμάτων τους. Πολλά μηνύματα αυτής της κατηγορίας μεταφέρουν επισυναπτόμενα αρχεία, τα οποία μπορεί να είναι **ιοί** ή **δούρειοι ίπποι**, τα οποία θέτουν σε κίνδυνο την ασφάλεια των συστημάτων. Το τελευταίο διάστημα, μεγάλο ποσοστό ανεπιθύμητης και επικίνδυνης αλληλογραφίας είναι αποτέλεσμα της δράσης ιών που έχουν προσβάλει διάφορα συστήματα που συνδέονται στο Διαδίκτυο. [5]

1.4 Τρόποι Αποφυγής *Spam*

Μερικές από τις τακτικές για την αποφυγή ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου *(spam)* είναι οι ακόλουθες:

1. Η δημιουργία μίας διεύθυνσης ηλεκτρονικού ταχυδρομείου αποκλειστικά για συναλλαγές στον Ιστό.

2. Η χρήση κάποιας διαδικτυακής υπηρεσίας δωρεάν ηλεκτρονικού ταχυδρομείου, για τη δημιουργία ενός λογαριασμού που μπορεί να χρησιμοποιηθεί στις ηλεκτρονικές συναλλαγές, θεωρείται σημαντική. Με αυτόν τον τρόπο, διασφαλίζεται η διατήρηση

της διεύθυνσης ηλεκτρονικού ταχυδρομείου που έχει εκχωρηθεί από τον εκάστοτε πάροχο υπηρεσιών Διαδικτύου (ISP) ή της απόρρητης διεύθυνσης, αν έχει παραχωρηθεί από το επαγγελματικό περιβάλλον.

3. Οι χρήστες οφείλουν να δίνουν την προσωπική τους διεύθυνση ηλεκτρονικού ταχυδρομείου μόνο σε άτομα που εμπιστεύονται.

4. Η καταχώρηση της διεύθυνση ηλεκτρονικού ταχυδρομείου σε μεγάλους καταλόγους του Διαδικτύου, ακόμα και σε προσωπική διαδικτυακή τοποθεσία, μπορεί να «διευκολύνει» τους *spammers*.

5. Προτείνεται η διεύθυνση ηλεκτρονικού ταχυδρομείου να παραμένει προσωπική.

6. Η χρήση μιας «καμουφλαρισμένης» διεύθυνσης συνιστάται, όταν δίνεται η διεύθυνση του χρήστη σε ομάδα συζήτησης, σε κανάλι συνομιλίας ή σε ηλεκτρονικό πίνακα ανακοινώσεων. Για παράδειγμα, θα μπορούσε να δοθεί μια ηλεκτρονική διεύθυνση ως "s0me0ne@example.c0m" χρησιμοποιώντας "0" (μηδέν) αντί για το "o." Κάποιο πρόσωπο μπορεί να καταλάβει τη διεύθυνση, αλλά τα αυτοματοποιημένα προγράμματα που χρησιμοποιούν οι αποστολείς μηνυμάτων *spam*, δεν μπορούν.

7. Επιβάλλεται, επίσης, προσοχή στα πλαίσια επιλογών σε διαδικτυακές συναλλαγές.

8. Κατά την αγορά αντικειμένων από το Διαδίκτυο, οι εταιρείες συνήθως προσθέτουν ένα πλαίσιο επιλογής (προεπιλεγμένο!), το οποίο υποδεικνύει ότι συμφωνεί ο πελάτης να πωληθεί ή να δοθεί η διεύθυνση του ηλεκτρονικού ταχυδρομείου του σε «υπεύθυνα πρόσωπα». Η αποεπιλογή της κρίνεται αναγκαία.

9. Έλεγχος στις πολιτικές απορρήτου των διαδικτυακών τοποθεσιών.

10. Σε περιπτώσεις εγγραφής σε υπηρεσίες που βασίζονται στον Ιστό, όπως ηλεκτρονικές τραπεζικές συναλλαγές, αγορές ή δελτία ενημέρωσης, πρέπει να εξετάζεται προσεκτικά η πολιτική απορρήτου, προτού αποκαλυφθεί η διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη. Η πολιτική απορρήτου θα εξηγεί τους όρους και τις περιπτώσεις σχετικά με το εάν ή το πώς η τοποθεσία θα κοινοποιήσει τα

δεδομένα του. Εάν δεν διαβάσει μάλιστα κάποια δήλωση, πιθανόν να "συμφωνήσει" στην κοινοποίηση των προσωπικών του δεδομένων, χωρίς να το γνωρίζει.

11. Εάν κάποια διαδικτυακή τοποθεσία δεν διαθέτει δήλωση απορρήτου, ο χρήστης οφείλει να επικοινωνήσει πρώτα με τους ιδιοκτήτες της τοποθεσίας, προτού κοινοποιήσει σημαντικές πληροφορίες.

12. Εάν η διαδικτυακή τοποθεσία δεν εξηγεί τον τρόπο με τον οποίο θα χρησιμοποιήσει τα προσωπικά δεδομένα, δεν υποχρεούται να τα δώσει. Πρέπει να γνωστοποιηθεί πως πολλές εταιρείες - ακόμη και νόμιμες - ενδέχεται να κοινοποιήσουν τα προσωπικά δεδομένα με ανεπιθύμητους, πολλές φορές, τρόπους [Καρεκλάς, 2005].

1.5 Τρόποι Αντιμετώπισης Spam

Γενικά μπορούμε να πούμε ότι υπάρχουν δύο τρόποι προστασίας από *spam*:

1. Μέσω εφαρμογών *spam blockers* και

2. Με τη χρήση *spam* φίλτρων.

Παρακάτω θα αναφερθούμε σε αυτούς τους δύο τρόπους και τη προστασία που παρέχουν στους χρήστες έναντι του *spam.*

Το *spam blocker* μπορεί να αποδειχθεί ένας αποτελεσματικός τρόπος για την καταπολέμηση ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου. Αυτού του είδους το λογισμικό διαφέρει από τα προγράμματα *spam filter,* αφού μπλοκάρει όλα σχεδόν τα εισερχόμενα *spam* μηνύματα. Στην πραγματικότητα δηλαδή, πρόκειται για ένα σύστημα το οποίο αποτρέπει τη λήψη μηνυμάτων *spam.* Συνήθως, αποτρέπει ένα ποσοστό 90% ανεπιθύμητων μηνυμάτων να φθάσουν στο ηλεκτρονικό ταχυδρομείο ενός χρήστη, και μπορεί να περιέχουν ποικίλους ιούς ή άλλα κακόβουλα λογισμικά. Η απόδοση σε μερικά συστήματα μπορεί να φθάσει και το 99% [Βαβίτσας, 2009]. Πώς δουλεύει, όμως μία εφαρμογή *spam blocker*;

Μία τέτοια εφαρμογή λειτουργεί μέσω του διακομιστή που είναι υπεύθυνος για το ηλεκτρονικό ταχυδρομείο, ελέγχοντας το λογαριασμό ενός χρήστη για ανεπιθύμητα μηνύματα και σβήνοντάς τα, ώστε αυτά να μην παραδοθούν. Όταν ψάχνουμε για ένα *spam blocker*, θα πρέπει να ελέγξουμε να είναι συμβατό με τη υπηρεσία ηλεκτρονικού ταχυδρομείου που διαθέτουμε, να έχει μεγάλο ποσοστό αποτροπής ανεπιθύμητων μηνυμάτων, να είναι εύκολο στην εγκατάσταση, αλλά και το κόστος αγοράς του. Μερικά από τα πλεονεκτήματα εγκατάστασης και χρήσης μιας εφαρμογής *spam blocking* είναι ότι εγκαθίσταται σχετικά εύκολα και δε χρειάζεται περαιτέρω διαμόρφωση για να λειτουργήσει, επιτρέπει στο να διατηρήσει την διεύθυνση ηλεκτρονικού ταχυδρομείου που διαθέτει, ενώ επειδή σβήνει τα *spam* μηνύματα, ελαχιστοποιεί το χρόνο που ο χρήστης ασχολείται με αυτά καθώς και τη πιθανότητα να μολυνθεί ο υπολογιστής από κάποιο κακόβουλο λογισμικό. Μία άλλη λύση για την αντιμετώπιση των ανεπιθύμητων μηνυμάτων, είναι η χρήση των λεγόμενων *spam* φίλτρων [Βαβίτσας, 2009].

Ένα τέτοιο φίλτρο είναι ένα λογισμικό το οποίο μπορεί και μπλοκάρει τα ανεπιθύμητα μηνύματα με τρεις βασικούς τρόπους:

- **με την εγκαθίδρυση λευκών και μαύρων λιστών (white/black lists):** οι οποίες μπορούν και δημιουργούν μία λίστα με αποδεκτές διευθύνσεις, όπου όλα τα μηνύματα από αυτές γίνονται δεκτά, και μία λίστα με ανεπιθύμητες διευθύνσεις, όπου τα μηνύματα που λαμβάνονται από αυτές αποθηκεύονται σε ένα ξεχωριστό κατάλογο.
- **Μπλοκάρισμα των λεγόμενων «sporm»:** τα *spam* φίλτρα μπορούν και μπλοκάρουν ένα μεγάλο ποσοστό από *spam* που είναι σχετικά με πορνογραφία, καθώς επίσης και εισερχόμενα μηνύματα τα οποία έχουν περιεχόμενο σχετικό με ενηλίκους, όπως εικόνες ή κείμενα.
- **Οργάνωση των μηνυμάτων:** αυτές οι εφαρμογές επιτρέπουν στους χρήστες να δημιουργήσουν φακέλους, ώστε να μπορούν να αποθηκεύονται μηνύματα ανάλογα με τη κατηγορία στην οποία ανήκουν (από φίλους, οικονομικά, προσωπικά, σχετικά με παιχνίδια). Τα εισερχόμενα μηνύματα αυτόματα κατηγοριοποιούνται στον κατάλληλο φάκελο, ώστε ο χρήστης να μπορέσει να διαλέξει το ποια θα διαβάσει. Αυτό που θα πρέπει να τονιστεί είναι ότι ο

χρήστης είναι αυτός που θέτει τα διάφορα φίλτρα, ανάλογα με τους κανόνες που επιλέγει, ώστε να γίνει η κατηγοριοποίηση των εισερχόμενων μηνυμάτων.

Μία σημαντική διαφορά του *spam blocker* από το *spam filter* είναι πως μέσω του *spam filter* έχουμε την οργάνωση των μηνυμάτων σε καταλόγους ανάλογα με το περιεχόμενο, ώστε ο χρήστης να δει μετά τι χρειάζεται και να απαντήσει ή να σβήσει όσα δε χρειάζεται. Παράλληλα, ένα πρόγραμμα *spam blocker* σβήνει όλα τα εισερχόμενα *spam* μηνύματα διευκολύνοντας τον χρήστη.

Η αντιμετώπιση, συνεπώς, ενός μεγάλου αριθμού ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου μπορεί να πραγματοποιηθεί μέσα από τη προσεκτική παρακολούθηση και ανακάλυψη κάποιων βασικών χαρακτηριστικών που έχουν τα *spam* μηνύματα, είτε στη διεύθυνση του αποστολέα, είτε μέσα στο κείμενο [Βαβίτσας, 2009].

2 *Μηχανισμοί Αντιμετώπισης - Components Φίλτρου*

Το φιλτράρισμα ενός μηνύματος ηλεκτρονικού ταχυδρομείου είναι μια σειρά ελέγχων μέσω ταξινομητών, αλγόριθμων απόδοσης, tokenizers και άλλων εργαλείων για την ταυτοποίηση των ανεπιθύμητων μηνυμάτων *Spam.* Το κάθε συστατικό του φίλτρου χρησιμοποιεί τις δικές του υλοποιήσεις κώδικα. Αυτό επιτρέπει στο φίλτρο να τα χρησιμοποιεί με οποιαδήποτε σειρά ανάλογα με τον τρόπο που θέλει να χειριστεί το εκάστοτε μήνυμα. Ο συνδυασμός διαφορετικών συστατικών για την αναγνώριση ενός *Spam* οδηγεί σε μεγαλύτερη απόδοση του φίλτρου σε σχέση με κάθε τεχνική ξεχωριστά.

2.1 Τεχνικές Φιλτραρίσματος

2.1.1 Μέθοδος Ανάλυσης Περιεχομένου

Οι τεχνικές Ανάλυσης Περιεχομένου στηρίζονται στην ανάλυση του περιεχομένου των μηνυμάτων, με σκοπό την ομαδοποίησή τους βάσει κοινών χαρακτηριστικών. Αυτές οι τεχνικές χωρίζονται σε δύο μεγάλες κατηγορίες: στην στατική ανάλυση περιεχομένου και στην δυναμική ανάλυση περιεχομένου.

Η πρώτη εξετάζει τα περιεχόμενα ενός μηνύματος ηλεκτρονικού ταχυδρομείου ψάχνοντας για λέξεις ενδεικτικές σε *Spam* μηνύματα, τα λεγόμενα *tokens*. Αυτές οι λέξεις πιθανόν ανασύρονται από ένα δυναμικό λεξικό το οποίο τροφοδοτείτε συνεχώς με λέξεις που περιέχονται σε μηνύματα *Spam.*

Η δεύτερη στηρίζεται κυρίως σε *Bayesian* φίλτρα και σε μηχανισμούς λήψης αποφάσεων. Η τεχνική των *Bayesian* φίλτρων βασίζεται στο θεώρημα του *Bayes* το

οποίο και αναφέρει ότι η πιθανότητα ένα *email* να είναι *spam* είναι ίση με την πιθανότητα να βρεθούν συγκεκριμένες λέξεις σε ένα *email spam* επί την πιθανότητα κάθε *email* να είναι *spam* δια την πιθανότητα να βρεθούν οι συγκεκριμένες λέξεις σε οποιοδήποτε *email* [12]. Ο μαθηματικός τύπος που προκύπτει είναι ο ακόλουθος:

$$\Pr(\text{spam}|\text{words}) = \frac{\Pr(\text{words}|\text{spam})\Pr(\text{spam})}{\Pr(\text{words})}$$

Ένα από τα πλεονεκτήματα του θεωρήματος *Bayes* είναι πως οι πιθανότητες των λέξεων να θεωρηθούν *spam* είναι εξατομικευμένες για κάθε χρήστη, έτσι ώστε να μπορούν να διορθωθούν με το πέρασμα του χρόνου, καθώς εξαρτώνται από το πόσα και τι είδους email δέχεται ο χρήστης, με αποτέλεσμα την αυξημένη αξιοπιστία των πιθανοτήτων του πότε μια λέξη να θεωρηθεί *spam* ή όχι. Τα αποτελέσματα των πιθανοτήτων των λέξεων αποθηκεύονται σε ένα σημείο το οποίο είναι προσβάσιμο και από τους υπόλοιπους χρήστες του συστήματος. Ένα απλό παράδειγμα είναι, εάν ο χρήστης είναι εγγεγραμμένος σε κάποια λίστα ηλεκτρονικής ενημέρωσης, η λίστα αυτή θα έχει μικρές πιθανότητες να θεωρηθεί ως *spam* [12].

Ένα ακόμα πλεονέκτημα είναι ότι μπορεί μια λέξη να θεωρηθεί ως *spam*, αλλά όχι όλο το *email*, διότι κατά το φιλτράρισμα κάθε λέξη ελέγχεται ξεχωριστά. Για παράδειγμα εάν ο χρήστης λάβει ένα *email* με θέμα «*Festival against the drugs*» και περιεχόμενο «*On Monday at Syntagma square, festival agaist drugs. Be there*», η λέξη *drugs* έχει αυξημένες πιθανότητες να θεωρηθεί ως *spam* διότι χρησιμοποιείτε από τους *spammers* για να διαφημίσουν χάπια και άλλα παρόμοια, αλλά όχι οι υπόλοιπες λέξεις.

Παράλληλα, το μειονέκτημα του θεωρήματος του *Bayes* είναι ένα από τα πλεονεκτήματα του. Ένας *spammer* μπορεί να στείλει ένα *email* με θέμα «*drug festival*» και περιεχόμενο «*drug viagra festival, everywhere delivery at your place, buy very cheap*». Οι λέξεις *festival* και *drug(s)* θα είχαν πολλές πιθανότητες να θεωρηθούν ως *spam*, ωστόσο επειδή το προηγούμενο *email* που τις ανέφερε δεν θεωρήθηκε spam, πιθανότατα να μην θεωρηθεί ούτε και αυτό [12].

2.1.2 *Checksum-based* φιλτράρισμα

Πολλά από τα *spam emails* στέλνονται πολλές φορές στον ίδιο χρήστη, αλλά έχουν διαφορετική διεύθυνση αποστολέα και θέμα, με αποτέλεσμα κάποιο από όλα αυτά να τα διαβάσει. Μία μέθοδος καταπολέμησης αυτού του τύπου *spam* είναι το φιλτράρισμα *checksum*, το οποίο ελέγχει τις λέξεις των *emails* σε κάθε εισερχόμενο μήνυμα και εάν βρει λέξεις οι οποίες έχουν θεωρηθεί ως *spam*, αυτόματα τις μαρκάρει και δημιουργεί ένα μοναδικό αριθμό κλειδί με βάση τις λέξεις που βρήκε για το email αυτό και το αποθηκεύει για μελλοντική χρήση. Από την πλευρά του χρήστη, τα *emails* μπορούν να δηλωθούν σαν *spam* από τον ίδιο, μέσω μίας επιλογής του προγράμματος που χρησιμοποιεί. Το πλεονέκτημα με αυτή την μέθοδο είναι ότι ο χρήστης έχει τον έλεγχο για το ποια *emails* να θεωρούνται spam και ποια όχι. Το μειονέκτημα, όμως, είναι ότι οι κακόβουλοι χρήστες γνωρίζοντας αυτή τη τεχνική, στέλνουν τα email τους προσθέτοντας περισσότερες λέξεις με αποτέλεσμα να δημιουργούνται καινούρια μοναδικά κλειδιά [12].

Από την πλευρά των παρόχων υπηρεσιών *Internet* και ηλεκτρονικού ταχυδρομείου, το φιλτράρισμα γίνεται πριν φτάσει το *email* στον χρήστη και έπειτα, χρησιμοποιείται η ίδια τεχνική. Η διαφορά είναι ότι ο διαχειριστής του συστήματος, πρέπει να δηλώσει το εκάστοτε *email* σαν *spam*, και ότι το μοναδικό κλειδί που δημιουργείται αποθηκεύεται σε κάποια τοποθεσία στο *Internet* στην οποία, όμως, έχουν πρόσβαση όλοι.

2.1.3 *DNS Blacklists*

Άλλη μια πολύ γνωστή μέθοδος καταπολέμησης του *email spam* είναι τα *DNS Blacklists (DNSBL)*, τα οποία βασίζονται στο Σύστημα Ονομάτων Τομέα *(Domain Name System, DNS).* Τα *DNSBLs* τρέχουν από ανεξάρτητους οργανισμούς, οι οποίοι είναι σύμμαχοι στην καταπολέμηση των *spam emails.* Η λειτουργία τους είναι πολύ απλή. Όταν το σύστημα διαχείρισης ηλεκτρονικού ταχυδρομείου καταλάβει ότι δέχεται πολλά *emails* από μια διεύθυνση IP του *Internet* και τα θεωρήσει *spam emails*, τότε στέλνει την διεύθυνση αυτή σε ένα ή περισσότερα *DNSBLs*, με αποτέλεσμα την επόμενη φορά που θα φτάσει κάποιο *email* στο σύστημα διαχείρισης ηλεκτρονικού

ταχυδρομείου, να θεωρηθεί αυτόματα ως *spam* χωρίς περαιτέρω διεργασίες που καταλαμβάνουν επεξεργαστική ισχύ. Γνωστά *DNSBL* είναι τα *Spamhaus* και τα *Spam and Open Relay Blocking System (SORBS).* Το πλεονέκτημα με τα *DNSBLs* είναι ότι οι διευθύνσεις IP που στέλνουν spam emails υπάρχουν διαθέσιμες σε κάποιο συγκεκριμένο σημείο και έχουν πρόσβαση εκατομμύρια συστήματα διαχείρισης ηλεκτρονικού ταχυδρομείου, με αποτέλεσμα την πιο αξιόπιστη και με πιο μεγάλη ακρίβεια αποφυγή των spam emails. Το μειονέκτημα μέχρι στιγμής, από την άλλη πλευρά, είναι ότι για κάθε καινούριο email που παραλαμβάνει το σύστημα διαχείρισης ταχυδρομείου, και το οποίο δεν έχει θεωρηθεί spam πρέπει να συγκρίνεται με κάποια από τις λίστες, έτσι ώστε να εξακριβωθεί εάν είναι *spam* ή όχι [12].

2.1.4 Μαύρες Λίστες

Ο μηχανισμός των Μαύρων Λιστών χρησιμοποιεί μια γνωστή λίστα διευθύνσεων των οποίων τα μηνύματα ηλεκτρονικής αλληλογραφίας απορρίπτονται άμεσα ως *Spam* από το εκάστοτε φίλτρο. Το βασικό μειονέκτημα τόσο αυτής της μεθόδου είναι ο μεγάλος αριθμός διευθύνσεων στην λίστα λόγω λανθασμένων αποδοχών ή λανθασμένων απορρίψεων. Για την αντιμετώπιση αυτού του προβλήματος προέκυψε η μέθοδος φήμης[7].

2.1.5 Μέθοδος Φήμης

Τα συστήματα φήμης στηρίζονται στην απόδοση ενός βαθμού αξιοπιστίας σε κάθε αποστολέα ηλεκτρονικής αλληλογραφίας[9]. Όταν ο βαθμός αυτός είναι υψηλότερος από ένα όριο, το οποίο μπορεί να ορίσει το φίλτρο, τα μηνύματα ηλεκτρονικού ταχυδρομείου του συγκεκριμένου χρήστη γίνονται αποδεκτά. Αν είναι χαμηλότερος, τα μηνύματά του είτε απορρίπτονται, είτε τους εφαρμόζονται άλλες τεχνικές *Anti Spam* για την περαιτέρω εξακρίβωση της φύσης τους. Η μέθοδος αυτή εξομοιώνει αποτελεσματικά και ταυτόχρονα τις μεθόδους μαύρης αλλά και άσπρης λίστας [Γεώργιζα, 2009].

2.2 *Tokenizers*

Για να γίνει σάρωση ενός μηνύματος ηλεκτρονικού ταχυδρομείου, το κείμενο του μηνύματος θα πρέπει πρώτα να χωριστεί σε μικρά τμήματα κειμένου, τα οποία αναφέρονται ως *tokens*. Παρόλο που ο ορισμός του *token* ποικίλει μεταξύ τεχνικών και εφαρμογών, παραδοσιακά είναι μια σειρά από τυπικούς αλφαριθμητικούς χαρακτήρες, τελείες και αποστρόφου. Οποιοσδήποτε χαρακτήρας δεν ανήκει σε αυτές τις κατηγορίες θεωρείται διαχωριστής *token*. Τα περισσότερα προγράμματα αγνοούν tokens τα οποία αποτελούνται από σειρές αριθμών άλλα και σχόλια κώδικα πχ στην *HTML* και την *Javascript* και τα οποία μπορεί να περιλαμβάνονται στο μήνυμα[15].

Η γραμματοσειρά ενός χαρακτήρα σε ένα *token* πολλές φορές θεωρείται ξεχωριστό token. Π.χ το «ΓΕΙΑ» είναι διαφορετικό token από το «γεια». Το μειονέκτημα σε αυτήν την περίπτωση είναι οι πολλαπλές καταχωρήσεις στις βιβλιοθήκες βάσεων δεδομένων των *tokens*. Πολλά προγράμματα χρησιμοποιούν αλγόριθμους για την μετατροπή των χαρακτήρων σε μία ενιαία γραμματοσειρά πριν την σάρωση προς αποφυγή αυτού το προβλήματος. Μερικοί δημοφιλείς μηχανισμοί για *tokenization* είναι οι *Whitespace* και *Krusty Tokenizers* [15].

2.2.1 *Whitespace Tokenizer*

Ο *Whitespace Tokenizer* είναι η πιο απλή μορφή *tokenization*, όπου ο διαχωριστής κειμένου είναι αυστηρά το κενό. Κάθε σημείο στίξης όπως και κάθε συντακτικός χαρακτήρας, δεν θεωρείται κομμάτι του *token* κατά την ανάλυση και απομακρύνεται αυτόματα. Για παράδειγμα σε μια γραμμή ενός μηνύματος:

*
<img> src=http://mail.teithe.gr//img.gif/*

Θα χωριζόταν σε δύο μόνο tokens με τον εξής τρόπο:

brimg
srchttpmailteithegrimggif

Ένα πλεονέκτημα αυτής της μεθόδου είναι ο μειωμένος όγκος πληροφορίας που χρειάζεται να αποθηκευτεί. Ένα μειονέκτημα είναι ότι δεν παράγει μεγάλο επίπεδο ακρίβειας ως προς την εκπαίδευση μιας βάσης [15].

2.2.2 *Krusty Tokenizer*

Ο *Krusty Tokenizer* είναι παρόμοιος με τον *Whitespace Tokenizer* με μία διακριτή διαφορά. Αντί να απομακρύνει τα σημεία στίξης, μαζί με το κενό, τα χρησιμοποιεί για τον διαχωρισμό των *tokens*. Οπότε στο προηγούμενο παράδειγμα του 2.2.1 στην γραμμή ενός μηνύματος:

*
<img> src=http://mail.teithe.gr//img.gif/*

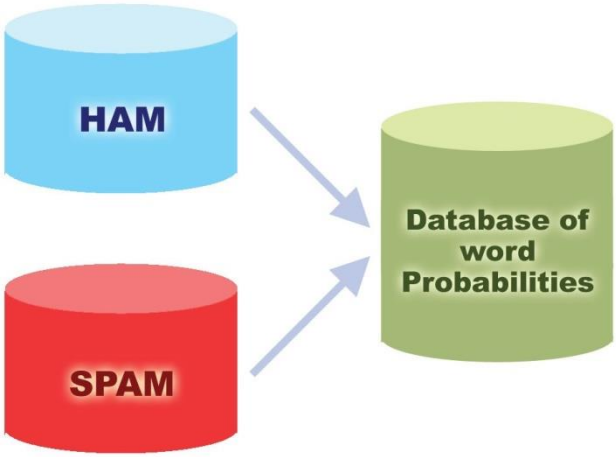
Θα χωριζόταν στα εξής *tokens*:
br
img
src
http
mail
teithe
gr
img
gif

Αυτή η μέθοδος παράγει σημαντικά περισσότερα *tokens* και αποθηκεύει μεγαλύτερο όγκο πληροφορίας αλλά επιτρέπει μεγαλύτερη ακρίβεια και παράγει πιο σωστά αποτελέσματα [15].

3

Σχεδιασμός της Επέκτασης «Arachnoid»

Για την αντιμετώπιση των ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου θα σχεδιαστεί φίλτρο με την μορφή επέκτασης που θα ενσωματωθεί στο πρόγραμμα ανάγνωσης μηνυμάτων ηλεκτρονικού ταχυδρομείου της *Mozilla* με το όνομα *Thunderbird*. Το φίλτρο που θα αναλυθεί παρακάτω θα ονομάζεται «*Arachnoid*». Το προϋπάρχον φίλτρο του *Thunderbird* αποτελεί υλοποίηση του θεωρήματος *Bayes*. Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, πρόκειται για μία από τις πιο διαδεδομένες αλλά και αποτελεσματικές μεθόδους αντιμετώπισης *Spam*. Το μειονέκτημα του φίλτρου της *Mozilla* είναι ότι απαιτεί μια περίοδο εκπαίδευσης του φίλτρου, ως προς τα μηνύματα που λαμβάνει ο εκάστοτε χρήστης, μέχρι να αρχίσει να αποδίδει[12]. Το «Arachnoid» φίλτρο που θα αναπτυχθεί θα επεκτείνει την απόδοση του προϋπάρχοντος φίλτρου χρησιμοποιώντας μια ανεξάρτητη βιβλιοθήκη λέξεων κλειδιών ορισμένων από το χρήστη. Θα χρησιμοποιεί την βιβλιοθήκη και με την χρήση κανόνων θα χαρακτηρίζει τα μηνύματα ως ανεπιθύμητα.



[Εικόνα 3.1] Λέξεις κλειδιά και πιθανότητα Spam

3.1 Λειτουργικές απαιτήσεις επέκτασης

Η επέκταση που θα αναπτυχθεί θα απαρτίζεται από ένα πλήθος λειτουργιών και υπηρεσιών οι οποίες θα λειτουργούν είτε αυτόματα είτε κατόπιν εντολής του χρήστη.

3.1.1 Ανίχνευση μηνυμάτων Spam

Το φίλτρο θα πρέπει να είσαι σε θέση να ανιχνεύει τα νέα μηνύματα Spam άμεσα και κατά την άφιξή τους χωρίς την παρέμβαση του χρήστη. Η διαδικασία αυτή θα πρέπει να γίνεται αυτόματα. Επιπρόσθετα, μπορεί να υπάρξει επιλογή αποστολής στον κάδο ανεπιθύμητων ή και απενεργοποίησης της αυτόματης ανίχνευσης.

3.1.2 Ανίχνευση μηνυμάτων Απάτης

Το φίλτρο θα πρέπει να είσαι σε θέση να ανιχνεύει τα νέα μηνύματα απάτης άμεσα και κατά την άφιξή τους χωρίς την παρέμβαση του χρήστη. Η διαδικασία αυτή θα πρέπει να γίνεται αυτόματα. Επιπρόσθετα, μπορεί να υπάρξει επιλογή αποστολής στον κάδο ανεπιθύμητων ή και απενεργοποίησης της αυτόματης ανίχνευσης

3.1.3 Σάρωση μηνυμάτων

Το φίλτρο θα πρέπει να είσαι σε θέση να σαρώσει ήδη στα υπάρχοντα μηνύματα, ένα ή πολλά μαζί, και να ανιχνεύσει τα Spam. Η διαδικασία αυτή θα εκκινείται κατόπιν εντολής του χρήστη και θα παρουσιάζει τα αποτελέσματα της σάρωσης σε αντίστοιχο παράθυρο-μήνυμα.

3.1.4 Σάρωση φακέλων

Το φίλτρο θα πρέπει να είσαι σε θέση να σαρώσει ολόκληρους φακέλους μηνυμάτων και να ανιχνεύσει τα Spam. Η διαδικασία αυτή θα εκκινείται κατόπιν εντολής του χρήστη και θα παρουσιάζει τα αποτελέσματα της σάρωσης σε αντίστοιχο παράθυρο-μήνυμα.

3.1.5 Μαύρη Λίστα Αποστολέων

Το φίλτρο θα διαθέτει υπηρεσίες μαύρης λίστας μεμονωμένων διευθύνσεων ηλεκτρονικού ταχυδρομείου ή ολόκληρων Domain. Τα μηνύματα με αποκλεισμένη προέλευση θα χαρακτηρίζονται αυτόματα ως Ανεπιθύμητα.

3.1.6 Προσθήκη στην Μαύρη Λίστα

Η πρόσθεση μιας διεύθυνσης ή ενός Domain στην μαύρη λίστα θα γίνεται είτε με μέσω επιλογής στο μενού επιλογών του Thunderbird είτε μέσα από το παράθυρο ρυθμίσεων της επέκτασης. Εκεί ο χρήστης θα είναι σε θέση να διαγράφει μια προηγούμενη καταχώρηση.

3.1.7 Λευκή Λίστα Αποστολέων

Το φίλτρο θα διαθέτει υπηρεσίες λευκής λίστας μεμονωμένων διευθύνσεων ηλεκτρονικού ταχυδρομείου ή ολόκληρων Domain. Τα μηνύματα με αξιόπιστη προέλευση δεν θα σαρώνονται από το φίλτρο.

3.1.8 Προσθήκη στην Λευκή Λίστα

Η πρόσθεση μιας διεύθυνσης ή ενός Domain στην λευκή λίστα θα γίνεται είτε με μέσω επιλογής στο μενού επιλογών του Thunderbird είτε μέσα από το παράθυρο ρυθμίσεων της επέκτασης. Εκεί ο χρήστης θα είναι σε θέση να διαγράφει μια προηγούμενη καταχώρηση.

3.1.9 Προσθήκη στην Λίστα Λέξεων Κλειδιών

Ο χρήστης θα έχει την δυνατότητα να προσθέσει λέξεις και φράσεις κλειδιά σε μια λίστα την οποία θα συμβουλεύεται το φίλτρο κατά την σάρωση για την ανίχνευση νέων Spam μηνυμάτων. Επιπλέον, θα είναι δυνατή και η διαγραφή τους.

3.1.10 Δείκτης επιπέδου Spam

Το φίλτρο θα χρησιμοποιεί έναν δείκτη επιπέδου Spam για κάθε λέξη κλειδί που ανιχνεύει. Ο χρήστης θα είναι σε θέση να μεταβάλει αυτόν τον δείκτη σε επιθυμητά επίπεδα.

3.1.11 Δείκτης χαρακτηρισμού Spam

Το φίλτρο θα χρησιμοποιεί έναν δείκτη Spam. Για κάθε λέξη κλειδί που θα ανιχνεύει το φίλτρο κατά την σάρωση, θα αθροίζεται σε αυτόν τον δείκτη ο δείκτης επιπέδου Spam της λέξης που ανιχνεύτηκε. Στο τέλος της σάρωσης, αν ο δείκτης Spam είναι μεγαλύτερος του δείκτη χαρακτηρισμού Spam, τότε το μήνυμα θα χαρακτηρίζεται ως ανεπιθύμητο. Ο χρήστης θα είναι σε θέση να μεταβάλει αυτόν τον δείκτη χαρακτηρισμού επηρεάζοντας με αυτόν τον τρόπο την ευαισθησία του φίλτρου.

3.2 Μη Λειτουργικές απαιτήσεις επέκτασης

3.2.1 Χρόνος Απόκρισης

Το φίλτρο πρέπει πάντοτε να ανταποκρίνεται άμεσα με αντίστοιχα μηνύματα αλλά ο χρόνος σάρωσης μπορεί να επηρεαστεί από τον αριθμό των μηνυμάτων που έχουν τεθεί ως προς αυτήν καθώς και το μέγεθος του εκάστοτε μηνύματος. Οι δύο τελευταίες μεταβλητές είναι αδύνατον να προβλεφτούν.

3.2.2 Απαιτήσεις συστήματος

Το φίλτρο θα πρέπει να λειτουργεί σε όλες τις διαθέσιμες εκδόσεις του Thunderbird και να αναβαθμίζεται εύκολα. Η λειτουργία του δεν θα πρέπει να επηρεάζεται από το λειτουργικό σύστημα στο οποίο είναι εγκατεστημένο.

3.3 Περιπτώσεις Χρήσης

3.3.1 Αυτόματη Σάρωση κατά την λήψη

Η διαδικασία έχει ως εξής:

1. Ο χρήστης λαμβάνει νέο μήνυμα.
2.
    2.1. Κατάσταση 1 (Η αυτόματη ανίχνευση είναι ενεργοποιημένη) :
    Το φίλτρο σαρώνει για την εξακρίβωση της προέλευσης-αποστολέα.
        2.1.1. Κατάσταση 1.1 (Ο αποστολέας είναι στην Λευκή Λίστα) : Το φίλτρο παραβλέπει το μήνυμα.
        2.1.2. Κατάσταση 1.2 (Ο αποστολέας είναι στην Μαύρη Λίστα) : Το φίλτρο χαρακτηρίζει το μήνυμα ως *Spam*.
        2.1.3. Κατάσταση 1.3 (Ο αποστολέας δεν είναι σε Λίστα) : Το φίλτρο σαρώνει το μήνυμα για λέξεις κλειδιά.
            2.1.3.1. Κατάσταση 1.3.1 (Το μήνυμα ανιχνεύεται ως Spam) : Το μήνυμα χαρακτηρίζεται Spam.
                2.1.3.1.1. Κατάσταση 1.3.1α (Η αυτόματη διαγραφή είναι ενεργοποιημένη) : Το φίλτρο διαγράφει το μήνυμα.
                2.1.3.1.2. Κατάσταση 1.3.1β (Η αυτόματη διαγραφή είναι απενεργοποιημένη) : Το φίλτρο δεν μετακινεί το μήνυμα.
            2.1.3.2. Κατάσταση 1.3.2 (Το μήνυμα ανιχνεύεται ως Απάτη) : Το μήνυμα χαρακτηρίζεται Απάτη.
                2.1.3.2.1. Κατάσταση 1.3.2α (Η αυτόματη διαγραφή είναι ενεργοποιημένη) : Το φίλτρο διαγράφει το μήνυμα.
                2.1.3.2.2. Κατάσταση 1.3.2β (Η αυτόματη διαγραφή είναι απενεργοποιημένη) : Το φίλτρο δεν μετακινεί το μήνυμα.
    2.2. Κατάσταση 2 (Η αυτόματη ανίχνευση είναι απενεργοποιημένη) : Το φίλτρο παραβλέπει το μήνυμα.

3.3.2 Χειροκίνητη Σάρωση

Η διαδικασία χειροκίνητης σάρωσης μηνυμάτων έχει ως εξής:

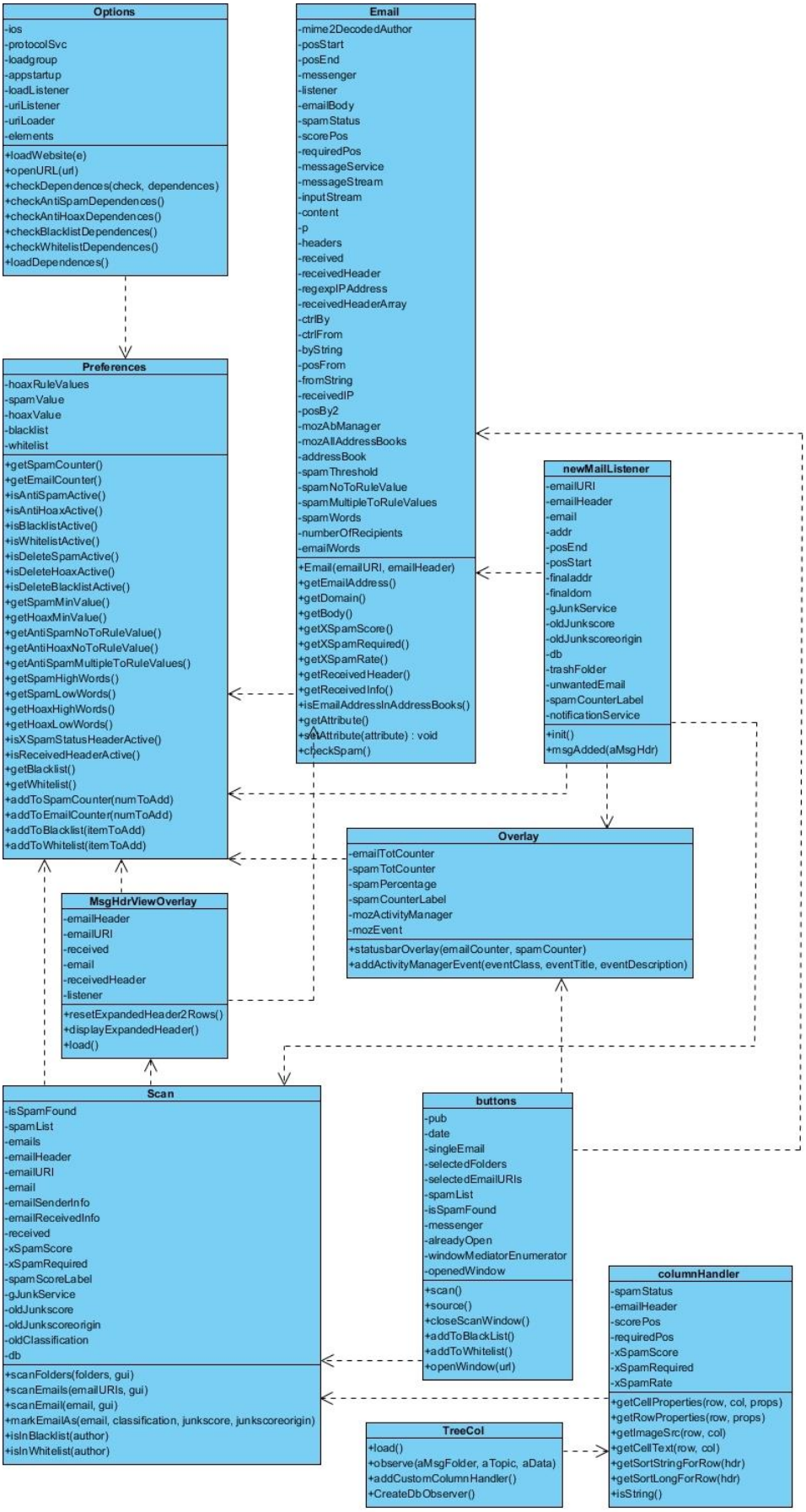
1. Ο χρήστης επιλέγει ένα ή περισσότερα μηνύματα.
2.
    2.1. Κατάσταση 1 (Ο χρήστης επέλεξε ένα μήνυμα) : Το φίλτρο σαρώνει για την εξακρίβωση της προέλευσης-αποστολέα.
        2.1.1. Κατάσταση 1.1 (Ο αποστολέας είναι στην Λευκή Λίστα) : Το φίλτρο παραβλέπει το μήνυμα.
        2.1.2. Κατάσταση 1.2 (Ο αποστολέας είναι στην Μαύρη Λίστα) : Το φίλτρο χαρακτηρίζει το μήνυμα ως *Spam*.
        2.1.3. Κατάσταση 1.3 (Ο αποστολέας δεν είναι σε Λίστα) : Το φίλτρο σαρώνει το μήνυμα για λέξεις κλειδιά.
            2.1.3.1. Κατάσταση 1.3.1 (Το μήνυμα ανιχνεύεται ως Spam) : Το μήνυμα χαρακτηρίζεται Spam.
            2.1.3.2. Κατάσταση 1.3.2 (Το μήνυμα ανιχνεύεται ως Απάτη) : Το μήνυμα χαρακτηρίζεται Απάτη.
    2.2. Κατάσταση 2 (Ο χρήστης επέλεξε περισσότερα μηνύματα) : Το φίλτρο σαρώνει για την εξακρίβωση της προέλευσης-αποστολέα.
        2.2.1. Κατάσταση 1.1 (Ο αποστολέας είναι στην Λευκή Λίστα) : Το φίλτρο παραβλέπει το μήνυμα και σαρώνει το επόμενο μήνυμα.
        2.2.2. Κατάσταση 1.2 (Ο αποστολέας είναι στην Μαύρη Λίστα) : Το φίλτρο χαρακτηρίζει το μήνυμα ως *Spam* και σαρώνει το επόμενο μήνυμα.
        2.2.3. Κατάσταση 1.3 (Ο αποστολέας δεν είναι σε Λίστα) : Το φίλτρο σαρώνει το μήνυμα για λέξεις κλειδιά.
            2.2.3.1. Κατάσταση 1.3.1 (Το μήνυμα ανιχνεύεται ως Spam) : Το μήνυμα χαρακτηρίζεται Spam και το φίλτρο σαρώνει το επόμενο μήνυμα.
            2.2.3.2. Κατάσταση 1.3.2 (Το μήνυμα ανιχνεύεται ως Απάτη) : Το μήνυμα χαρακτηρίζεται Απάτη και το φίλτρο σαρώνει το επόμενο μήνυμα.
3. Εμφανίζεται παράθυρο με τα αποτελέσματα της σάρωσης.

Η διαδικασία χειροκίνητης σάρωσης φακέλου έχει ως εξής:

1. Ο χρήστης επιλέγει ένα φάκελο.
2.
   2.1. Κατάσταση 1 (Ο φάκελος περιέχει ένα μήνυμα) : Το φίλτρο σαρώνει για την εξακρίβωση της προέλευσης-αποστολέα.
      2.1.1. Κατάσταση 1.1 (Ο αποστολέας είναι στην Λευκή Λίστα) : Το φίλτρο παραβλέπει το μήνυμα.
      2.1.2. Κατάσταση 1.2 (Ο αποστολέας είναι στην Μαύρη Λίστα) : Το φίλτρο χαρακτηρίζει το μήνυμα ως *Spam*.
      2.1.3. Κατάσταση 1.3 (Ο αποστολέας δεν είναι σε Λίστα) : Το φίλτρο σαρώνει το μήνυμα για λέξεις κλειδιά.
         2.1.3.1. Κατάσταση 1.3.1 (Το μήνυμα ανιχνεύεται ως Spam) : Το μήνυμα χαρακτηρίζεται Spam.
         2.1.3.2. Κατάσταση 1.3.2 (Το μήνυμα ανιχνεύεται ως Απάτη) : Το μήνυμα χαρακτηρίζεται Απάτη.
   2.2. Κατάσταση 2 (Ο φάκελος περιέχει περισσότερα μηνύματα) : Το φίλτρο σαρώνει για την εξακρίβωση της προέλευσης-αποστολέα.
      2.2.1. Κατάσταση 1.1 (Ο αποστολέας είναι στην Λευκή Λίστα) : Το φίλτρο παραβλέπει το μήνυμα και σαρώνει το επόμενο μήνυμα.
      2.2.2. Κατάσταση 1.2 (Ο αποστολέας είναι στην Μαύρη Λίστα) : Το φίλτρο χαρακτηρίζει το μήνυμα ως *Spam* και σαρώνει το επόμενο μήνυμα.
      2.2.3. Κατάσταση 1.3 (Ο αποστολέας δεν είναι σε Λίστα) : Το φίλτρο σαρώνει το μήνυμα για λέξεις κλειδιά.
         2.2.3.1. Κατάσταση 1.3.1 (Το μήνυμα ανιχνεύεται ως Spam) : Το μήνυμα χαρακτηρίζεται Spam και το φίλτρο σαρώνει το επόμενο μήνυμα.
         2.2.3.2. Κατάσταση 1.3.2 (Το μήνυμα ανιχνεύεται ως Απάτη) : Το μήνυμα χαρακτηρίζεται Απάτη και το φίλτρο σαρώνει το επόμενο μήνυμα.
3. Εμφανίζεται παράθυρο με τα αποτελέσματα της σάρωσης.

3.4 Διάγραμμα κλάσεων

Παρακάτω παρουσιάζεται το διάγραμμα κλάσεων της επέκτασης και περιγράφεται η εσωτερική δομή της, οι μεταβλητές και οι λειτουργίες της κάθε κλάσης καθώς και οι περιορισμοί όσον αφορά τον τρόπο με τον οποίο συνεργάζονται μεταξύ τους.


Options
-ios
-protocolSvc
-loadgroup
-appstartup
-loadListener
-uriListener
-uriLoader
-elements
+loadWebsite(e)
+openURL(url)
+checkDependences(check, dependences)
+checkAntiSpamDependences()
+checkAntiHoaxDependences()
+checkBlacklistDependences()
+checkWhitelistDependences()
+loadDependences()
Email
-mime2DecodedAuthor
-posStart
-posEnd
-messenger
-listener
-emailBody
-spamStatus
-scorePos
-requiredPos
-messageService
-messageStream
-inputStream
-content
-p
-headers
-received
-receivedHeader
-regexpIPAddress
-receivedHeaderArray
-ctrlBy
-ctrlFrom
-byString
-posFrom
-fromString
-receivedIP
-posBy2
-mozAbManager
-mozAllAddressBooks
-addressBook
-spamThreshold
-spamNoToRuleValue
-spamMultipleToRuleValues
-spamWords
-numberOfRecipients
-emailWords
+Email(emailURI, emailHeader)
+getEmailAddress()
+getDomain()
+getBody()
+getXSpamScore()
+getXSpamRequired()
+getXSpamRate()
+getReceivedHeader()
+getReceivedInfo()
+isEmailAddressInAddressBooks()
+getAttribute()
+setAttribute(attribute) : void
+checkSpam()
Preferences
-hoaxRuleValues
-spamValue
-hoaxValue
-blacklist
-whitelist
+getSpamCounter()
+getEmailCounter()
+isAntiSpamActive()
+isAntiHoaxActive()
+isBlacklistActive()
+isWhitelistActive()
+isDeleteSpamActive()
+isDeleteHoaxActive()
+isDeleteBlacklistActive()
+getSpamMinValue()
+getHoaxMinValue()
+getAntiSpamNoToRuleValue()
+getAntiHoaxNoToRuleValue()
+getAntiSpamMultipleToRuleValues()
+getSpamHighWords()
+getSpamLowWords()
+getHoaxHighWords()
+getHoaxLowWords()
+isXSpamStatusHeaderActive()
+isReceivedHeaderActive()
+getBlacklist()
+getWhitelist()
+addToSpamCounter(numToAdd)
+addToEmailCounter(numToAdd)
+addToBlacklist(itemToAdd)
+addToWhitelist(itemToAdd)
newMailListener
-emailURI
-emailHeader
-email
-addr
-posEnd
-posStart
-finaladdr
-finaldom
-gJunkService
-oldJunkscore
-oldJunkscoreorigin
-db
-trashFolder
-unwantedEmail
-spamCounterLabel
-notificationService
+init()
+msgAdded(aMsgHdr)
Overlay
-emailTotCounter
-spamTotCounter
-spamPercentage
-spamCounterLabel
-mozActivityManager
-mozEvent
+statusbarOverlay(emailCounter, spamCounter)
+addActivityManagerEvent(eventClass, eventTitle, eventDescription)
MsgHdrViewOverlay
-emailHeader
-emailURI
-received
-email
-receivedHeader
-listener
+resetExpandedHeader2Rows()
+displayExpandedHeader()
+load()
Scan
-isSpamFound
-spamList
-emails
-emailHeader
-emailURI
-email
-emailSenderInfo
-emailReceivedInfo
-received
-xSpamScore
-xSpamRequired
-spamScoreLabel
-gJunkService
-oldJunkscore
-oldJunkscoreorigin
-oldClassification
-db
+scanFolders(folders, gui)
+scanEmails(emailURIs, gui)
+scanEmail(email, gui)
+markEmailAs(email, classification, junkscore, junkscoreorigin)
+isInBlacklist(author)
+isInWhitelist(author)
buttons
-pub
-date
-singleEmail
-selectedFolders
-selectedEmailURIs
-spamList
-isSpamFound
-messenger
-alreadyOpen
-windowMediatorEnumerator
-openedWindow
+scan()
+source()
+closeScanWindow()
+addToBlackList()
+addToWhitelist()
+openWindow(url)
columnHandler
-spamStatus
-emailHeader
-scorePos
-requiredPos
-xSpamScore
-xSpamRequired
-xSpamRate
+getCellProperties(row, col, props)
+getRowProperties(row, props)
+getImageSrc(row, col)
+getCellText(row, col)
+getSortStringForRow(hdr)
+getSortLongForRow(hdr)
+isString()
TreeCol
+load()
+observe(aMsgFolder, aTopic, aData)
+addCustomColumnHandler()
+CreateDbObserver()

4

Τεχνολογικό Περιβάλλον Ανάπτυξης

Το *Arachnoid Spam Filter* είναι ένα *Add-on/plug-in*, γνωστά ως *externsions*, για τον *Mozilla Thunderbird* που επιτρέπει στον *Mail Client* της *Mozilla* και στον χρήστη να φιλτράρει πιο αποτελεσματικά την ληφθείσα αλληλογραφία. Η ανάπτυξη της εφαρμογής έγινε με την χρήση των λογισμικών *Notepad++* και *NetBeans* 8, τα οποία είναι ολοκληρωμένα περιβάλλοντα σύνταξης κειμένου και κώδικα. Τα περιβάλλοντα αυτά καθώς και όλα τα προγράμματα που χρησιμοποιήθηκαν είναι ελεύθερης διανομής (*freeware*) και ανοικτού κώδικα (*open source*). Η γλώσσες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής είναι η *Javascript* και η *XUL* ή αλλιώς ***X****ML* ***U****ser Interface* ***L****anguage*. Απαιτούμενη προϋπόθεση για την χρήση των προγραμμάτων είναι η εγκατάσταση των εργαλείων της γλώσσας προγραμματισμού *Java, Java Development Kit (JDK)*. Για την αποσφαλμάτωση της εφαρμογής χρησιμοποιήθηκε το ο ενσωματωμένος *Debugger* του *Mozilla Firefox* και το *Error Console* που βρίσκεται ενσωματωμένο στο *Thunderbird* κάτω από το μενού επιλογών *Tools*.

4.1 Τα *extensions* του *Mozilla Thunderbird*

Τα extensions είναι μικρά προγράμματα που μπορούν να εγκατασταθούν από τους χρήστες για να προστεθούν χαρακτηριστικά ή να βελτιστοποιήσουν τον Thunderbird. Οι επεκτάσεις προσφέρουν νέα λειτουργικότητα στον Mail Client, η οποία είναι ένα απλό κουμπί εργαλειοθήκης ή ακόμα και ένα εντελώς νέο χαρακτηριστικό. Επιτρέπουν με αυτόν τον τρόπο στον χρήστη να προσαρμόσει το πρόγραμμα ανάλογα με τις δικές του ανάγκες και προτιμήσεις.

Οι επεκτάσεις συσκευάζονται και διανέμονται μέσω αρχείων *ZIP* με την κατάληξη *xpi*. Η δομή του περιεχομένου του *XPI* αρχείου έχει ως εξής [8][10]:

extension.xpi:

/install.rdf

/defaults/

/defaults/preferences/.js*

/chrome.manifest

/chrome/

/chrome/content/

/chrome/skin

/chrome/locale

/chrome/modules

4.1.1 Το Αρχείο *Install.rdf*

Το αρχείο *install.rdf* είναι ένα αρχείο απλού κειμένου το οποίο περιέχει μετα-δεδομένα που δίνουν πληροφορίες στην εφαρμογή σχετικά με την επέκταση [8][10]. Τα δεδομένα αυτά ανήκουν σε 3 κατηγορίες:

- Πληροφορίες που χρησιμεύουν στο μοναδικό προσδιορισμό της επέκτασης από την εφαρμογή με μια μοναδική προσδιοριστική συμβολοακολουθία με το *ID* την έκδοση της επέκτασης.
- Πληροφορίες χρήσιμες στο χρήστη όπως το όνομα, μια περιγραφή της επέκτασης και τον συγγραφέα της επέκτασης.
- Πληροφορίες συμβατότητας, όπως η εφαρμογή στην οποία απευθύνεται και η ελάχιστη-μέγιστη έκδοση της εφαρμογής με την οποία παραμένει λειτουργική η επέκταση.

4.1.2 Το Αρχείο *chrome.manifest*

Το αρχείο *chrome.manifest* είναι ένα αρχείο που ενημερώνει τον *Thunderbird* για την δομή του πακέτου συσκευασίας των αρχείων *chrome* για τη συγκεκριμένη επέκταση. Το *chrome* είναι το σύνολο των στοιχείων της διεπαφής του χρήστη τα οποία

βρίσκονται έξω από την περιοχή του περιεχομένου του παραθύρου της εφαρμογής. Η *Mozilla* έχει δημιουργήσει μια νέα κατηγορία *URI*, τα «*chrome*://» URI, μέσω των οποίων μπορούν να φορτώνονται εύκολα τα αρχεία *chrome*, έτσι ώστε η εγκατεστημένη εφαρμογή να έχει την δυνατότητα να βρει και χειριστεί αυτά τα αρχεία. Πιο συγκεκριμένα, στο αρχείο chrome.manifest καταχωρούνται πληροφορίες πέντε ειδών:

- Το *overlay* και *style*: Ένα *overlay* επιτρέπει την προσθήκη νέου περιεχομένου σε ένα ήδη υπάρχον έγγραφο της εφαρμογής. Με τον τρόπο αυτόν λοιπόν δηλώνονται τα έγγραφα που πρόκειται να τροποποιηθούν, και τα έγγραφα της επέκτασης που περιέχουν τις τροποποιήσεις που θα γίνουν, αρχεία γλώσσας και τύπου *XUL*.
- Το *content*: Πρόκειται για τον κατάλογο όπου περιέχονται όλα τα *XUL* αρχεία που ορίζουν τα περιεχόμενα των παραθύρων και των διαλόγων της επέκτασης, και τα *JavaScript* αρχεία που καθορίζουν τη λειτουργικότητά της.
- Το *locale*: Εδώ προσδιορίζεται ο κατάλογος που περιέχει όλα τα αρχεία όπου περιέχονται οι μεταφράσεις των μηνυμάτων που απευθύνονται προς το χρήστη σε διάφορες γλώσσες.
- Το *skin*: Εδώ προσδιορίζεται ο κατάλογος των διάφορων εικόνων που χρησιμοποιούνται και των αρχείων που περιέχουν όλους τους προσδιορισμούς σχετικά με την εμφάνιση των διαφόρων στοιχείων της διεπαφής χρήστη, όπως τα αρχεία τύπου *css*.
- Το *resource*: Εδώ προσδιορίζεται ο κατάλογος των διάφορων *javascript* αρχείων που χρησιμοποιούνται για την διαμοιρασμένη χρήση κώδικα ανάμεσα στα στοιχεία της επέκτασης.

4.1.3 Ο Κατάλογος αρχείων *chrome*

Ο κατάλογος αυτός αποτελείται από τέσσερις υποκαταλόγους, *content*, *locale*, *modules* και *skin*, οι οποίοι αντιστοιχούν στα είδη των αρχείων που περιγράφηκαν παραπάνω στις καταχωρήσεις του αρχείου *chrome.manifest*. Από αυτούς, μόνο ο υποκατάλογος content είναι υποχρεωτικό να υπάρχει για να μπορεί να γίνει η εγκατάσταση της εφαρμογής, καθώς περιέχει όλο το περιεχόμενο της επέκτασης. Ο

συγγραφέας της επέκτασης μπορεί να προσθέσει δικούς του καταλόγους με αρχεία τα οποία μπορεί να καλεί μέσα από τα αρχεία του content[8].

4.1.4 Ο Κατάλογος αρχείων *default*

Εδώ περιέχονται τα αρχεία που ορίζουν και αρχικοποιούν τις μεταβλητές των προτιμήσεων του χρήστη. Συγκεκριμένα, οι αρχικοποιήσεις αυτές γίνονται σε ένα αρχείο *JavaScript* ώστε να φορτωθούν αυτόματα από το σύστημα κατά την εκκίνηση του *Thunderbird* [8].

4.2 Η Γλώσσα Προγραμματισμού *XUL*

Η *XUL (XML-based User Interface Language*), με τύπο αρχείων **.xul*, είναι μια γλώσσα διεπαφής χρήστη βασισμένη στην *XML*. Αναπτύχθηκε από την *Mozilla* και επιτρέπει στους προγραμματιστές να σχεδιάσουν *cross-platform* εφαρμογές που μπορούν να τρέχουν *online* αλλά και *offline*. Είναι προσαρμόσιμη χρησιμοποιώντας διαφορετικά γραφικά, κείμενο και σχεδιαγράμματα για να μπορεί να υποστηρίξει τοπικές και διεθνής αγορές [6]. Όλα τα στοιχεία που αποτελούν τη διεπαφή χρήστη στα προγράμματα και τις επεκτάσεις της *Mozilla* είναι γραμμένα σε γλώσσα *XUL*. Η *cross-platform* ιδιότητα της γλώσσας είναι και το μεγάλο πλεονέκτημα της, επιτρέποντας στους προγραμματιστές να δημιουργούν εφαρμογές μεταφέρσιμες σε οποιοδήποτε σύστημα. Επίσης, η *XUL* έχει όλα τα πλεονεκτήματα των γλωσσών *XML*, όπως να εμφωλεύει σε αυτήν στοιχεία από άλλες γλώσσες *XML*. Με την βοήθεια αρχείων *Javascrict* είναι εφικτό να δημιουργηθούν και να τροποποιηθούν δυναμικά τα στοιχεία της *XUL* [7].

Η *XUL* ορίζει διάφορα είδη στοιχείων, όπως:

- στοιχεία ανώτερων επιπέδων, π.χ. *window, page, dialog* κλπ.
- *widgets*, π.χ. *label, button, textbox, listbox, menulist, menupopup, checkbox, menu, toolbar, groupbox, tabbox, colorpicker, spacer, splitter* κλπ.
- *events* και *scripts*

• πηγές δεδομένων

• και άλλα όπως overlay, iframe κ.λ.π.

4.3 Η Γλώσσα Προγραμματισμού *JavaScript*

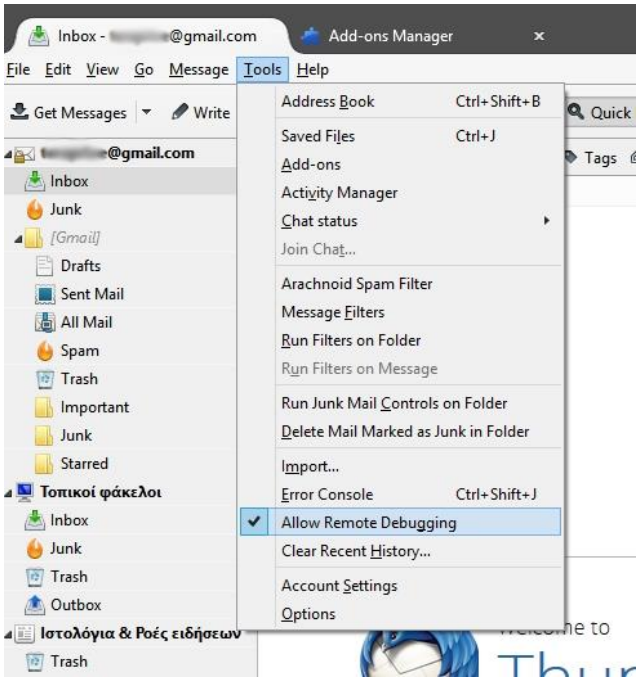
Η *JavaScript*, με τύπο αρχείων **.js*, είναι μια αντικειμενοστραφής scripting γλώσσα, ανεξάρτητη πλατφόρμας, η οποία συνήθως χρησιμοποιείται την για ανάπτυξη *client-side* εφαρμογών. Το επίσημο όνομά της είναι *ECMAScript*. Η *Javascript* έχει επηρεαστεί από πολλές γλώσσες προγραμματισμού και έχει αρκετές ομοιότητες με τη *Java* με σκοπό να είναι οικεία στους προγραμματιστές. Αρχικά αναπτύχθηκε από τον *Brendan Eich* στη *Netscape* και είχε το όνομα *Mocha*. Στη συνέχεια μμετονομάστηκε σε *LiveScript* και τελικά σε *JavaScript*. Για πρώτη φορά παρουσιάστηκε και ενσωματώθηκε στον περιηγητή της Netscape το 1995.

Η *JavaScript* υποστηρίζει τη σύνταξη δομημένου προγραμματισμού της *C* (π.χ. εντολές *if, switch,* βρόγχοι *while* κλπ.). Οι μεταβλητές της δεσμεύονται σε συγκεκριμένους τύπους και τα αντικείμενα σε αυτή αντιμετωπίζονται ως συσχετιζόμενοι πίνακες, έτσι ώστε τα ονόματα των γνωρισμάτων των αντικειμένων να αποτελούν κλειδιά για τους πίνακες. Ένα χαρακτηριστικά της *JavaScript* είναι ότι τα αντικείμενα της δε βασίζονται σε κλάσεις αλλά σε «πρωτότυπα». Επιπλέον, δε γίνεται διαχωρισμός μεταξύ των συναρτήσεων και των μεθόδων παρά μόνο κατά την κλήση μιας συνάρτησης και υποστηρίζει κανονικές εκφράσεις. Η βασικότερη χρήση της *JavaScript* είναι στη συγγραφή συναρτήσεων που βρίσκονται ενσωματωμένες σε *HTML* σελίδες και αλληλεπιδρούν με το μοντέλο *DOM* της σελίδας. Ο κώδικάς της μπορεί να εκτελείται τοπικά στον browser του χρήστη με αποτέλεσμα ναι ανταποκρίνεται πολύ γρήγορα στις ενέργειες του χρήστη, κάνοντας την εκάστοτε εφαρμογή ιδιαίτερα υποκρίσιμη. Ένα άλλο χαρακτηριστικό της είναι ότι μπορεί να ανιχνεύσει τις ενέργειες του χρήστη, όπως τις μεμονωμένες πιέσεις πλήκτρων. Η *JavaScript* χρησιμοποιείται στις επεκτάσεις της *Mozilla* για να παρέχει λειτουργικότητα κατά το «χρόνο εκτέλεσης» και αλληλεπίδρασης με το χρήστη, την αποθήκευση των προτιμήσεων του χρήστη και για τη δημιουργία στοιχείων *XPCOM*.

4.4 Ο *Mozilla Firefox Debugger*

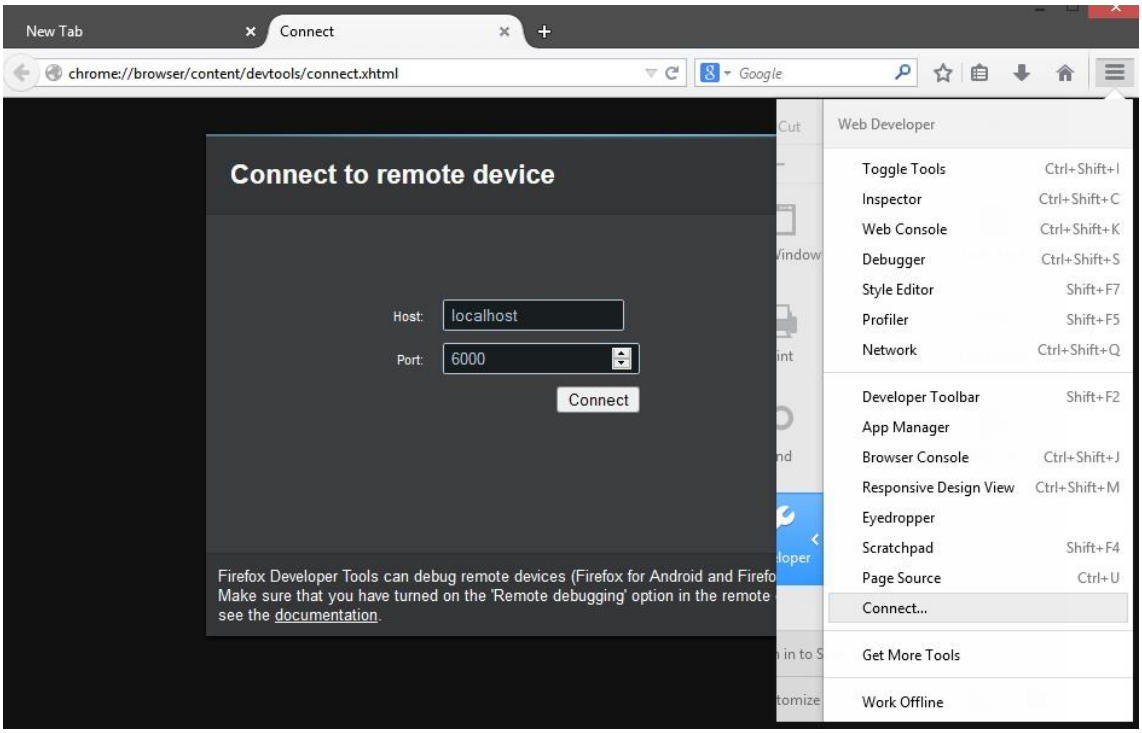
Η έλλειψη επίσημων εργαλείων ανάπτυξης προεκτάσεων για το Thunderbird από την *Mozilla* λειτουργεί κατασταλτικά για τους προγραμματιστές και τους αποθαρρύνει. Η βιβλιοθήκη της *Mozilla* είναι κατακερματισμένη και γίνεται ακόμα προσπάθεια συλλογής όλων των πληροφοριών σε ένα ενιαίο περιβάλλον. Τις ελλείψεις αυτές προσπαθεί να εξαλείψει το επίσημο *Forum* της *Mozilla* προσφέροντας βοήθεια και κώδικα αλλά και η χρήση του ενσωματωμένου *debugger* του *Mozilla Firefox* στον *Thunderbird*.

Για να μπορέσει να κάνει χρήση του *debugger*, ο προγραμματιστής πρέπει να κάνει μια σειρά ρυθμίσεων στα δύο προγράμματα. Ο προγραμματιστής πρέπει αρχικά να ενεργοποιήσει στο *Mozilla Thunderbird* το πρωτόκολλο απομακρυσμένης αποσφαλμάτωσης, που θα του επιτρέψει να κάνει χρήση των εργαλείων *Web Developer* του *Firefox* για να χειριστεί το *Thunderbird*. Η ενεργοποίηση γίνεται στην Επιλογή του μενού *Tools, Allow Remote Debugging*[Εικόνα 4.1]. Η προεπιλεγμένη ρύθμιση εκκινεί έναν εξυπηρετητή αποσφαλμάτωσης στο *port* 6000. Ο χρήστης μπορεί να αλλάξει το *port* ανοίγοντας το προχωρημένο επεξεργαστή ρυθμίσεων και να αλλάξει την τιμή στην επιλογή «*devtools.debugger.remote-port*» [9].



[Εικόνα 4.1] Ενεργοποίηση του Remote Debugging

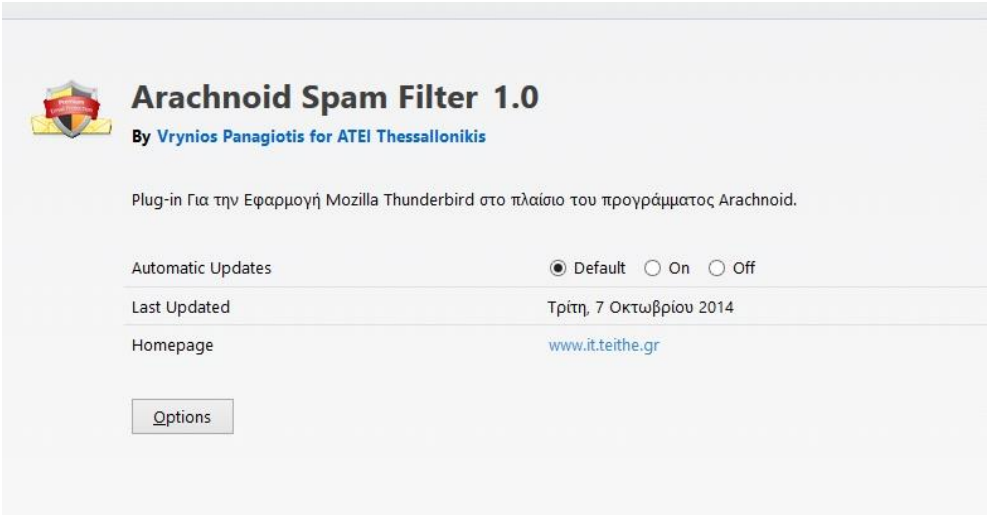
Ο *Mozilla Firefox* λειτουργεί ως client και προσφέρει το γραφικό περιβάλλον για την χρήση των εργαλείων ανάπτυξης για το *Thunderbird*. Στην συνέχεια λοιπον, ο προγραμματιστής ανοιγει το *Toolbox* του *Firefox*, πατά *Settings* και επιλέγει «*Enable remote debugging*».Έπειτα στο μενου του *Firefox* επιλέγει το υπομενού *Developer* και πατά *Connect*[Εικόνα 4.2].



[Εικόνα 4.2] Debugging μέσω του Mozilla Firefox

5 *Υλοποίηση Εφαρμογής*

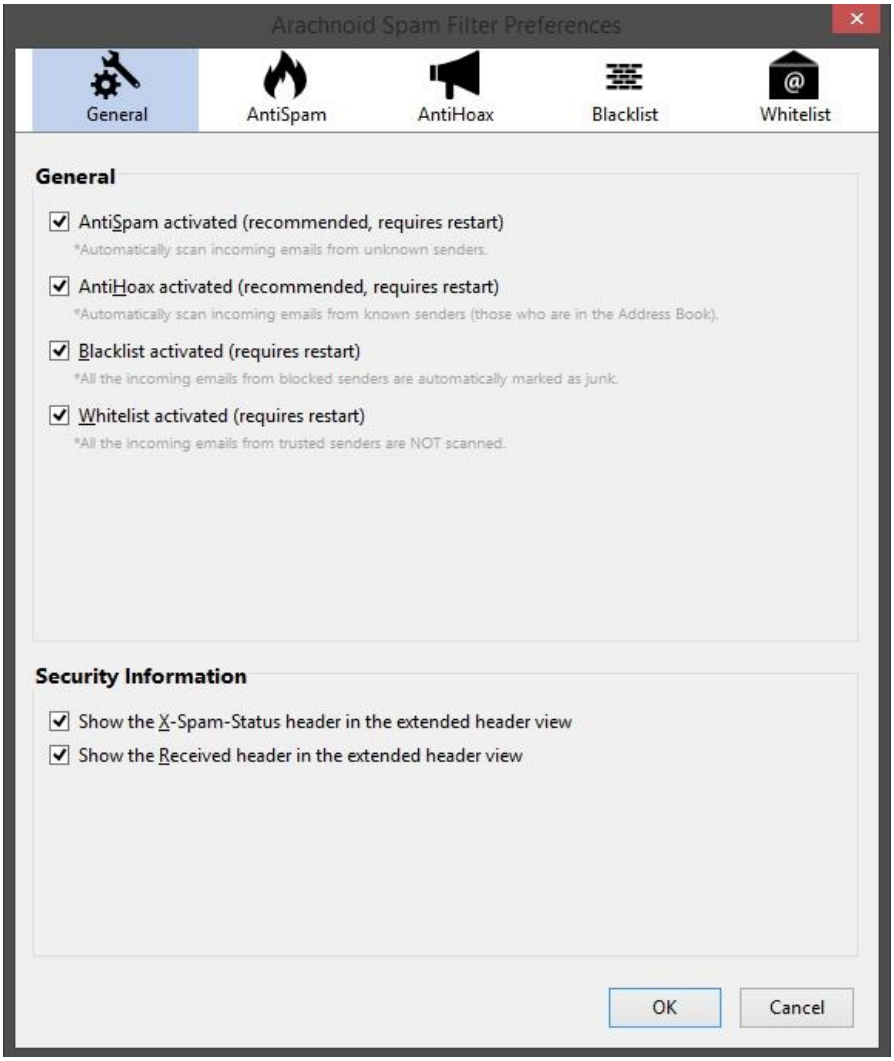
Η προέκταση *Arachnoid Spam Filter* απαιτεί εγκατάσταση από τον χρήστη κάνοντας *drag-n-drop* το αρχείο *arachnoidspamfilter.xpi* στο μενού *Tools* και επιλογή *Add-ons*. Μετά την εγκατάσταση ο χρήστης θα προσέξει το Τίτλο, τον συγγραφέα και τα σχόλια του συγγραφέα σχετικά με την προέκταση[Εικόνα 5.1]. Ο χρήστης μπορεί να επιλέξει να απενεργοποιήσει τις αυτόματες ενημερώσεις καθώς και να επεξεργαστεί τις ρυθμίσεις της εφαρμογής πατώντας στο πλήκτρο *Options.*



[Εικόνα 5.1] Το Description του Extension

5.1 Η Καρτέλα επιλογών *General*

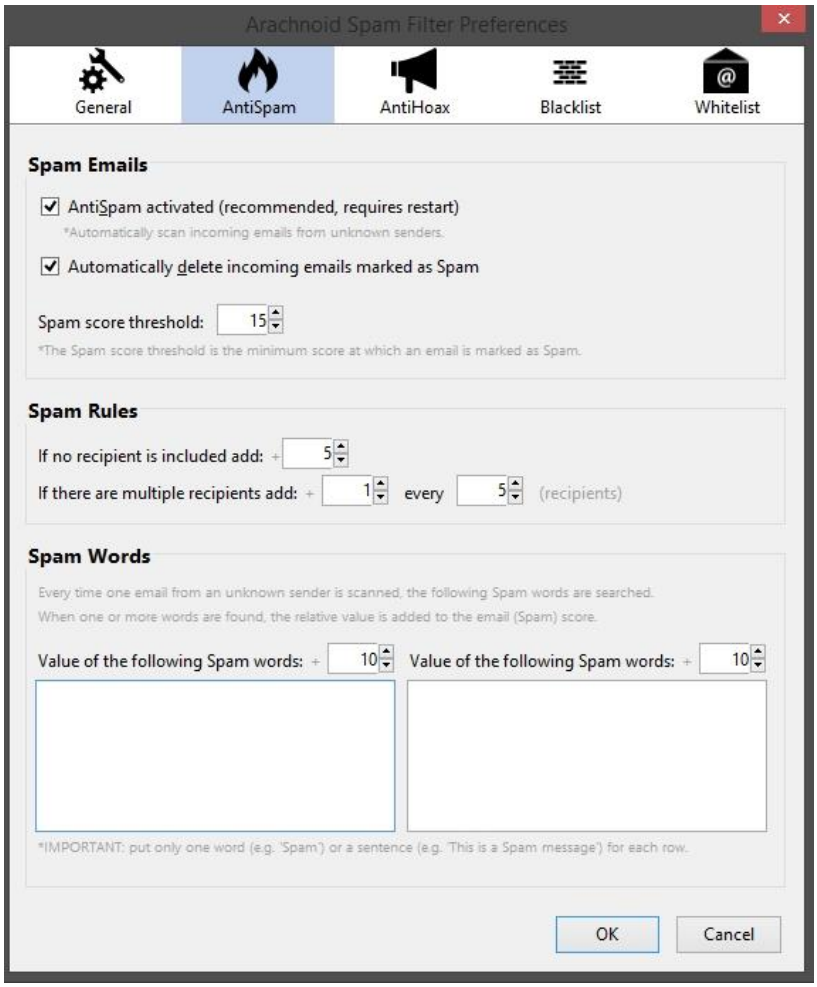
Ανοίγοντας το παράθυρο επιλογών παρουσιάζεται στον χρήστη η καρτέλα *General* με τις γενικές επιλογές του φίλτρου [Εικόνα 5.2]. Εδώ ο χρήστης μπορεί να Ενεργοποιήσει την προστασία *AntiSpam* και *AntiHoax* για την ανίχνευση των *Spam* μηνυμάτων και των μηνυμάτων απάτης κατά την άφιξή τους. Επιπλέον, επιλέγει αν θέλει να κάνει χρήση της Μαύρης Λίστας αποστολέων ή και της Άσπρης Λίστας αποστολέων. Στις πληροφορίες Ασφάλειας ο χρήστης επιλέγει αν θέλει να εμφανίζονται στα *Headers* του ληφθέντος μηνύματος το *X-Spam-Status*, ένας δείκτης κατάστασης spam για το συγκεκριμένο μήνυμα, και αν θέλει να συμπεριληφθεί το *Header* ληφθέντα στην εμφάνιση του επεκταμένου *Header*.



[Εικόνα 5.2] Το μενού επιλογών «*General*»

5.2 Η Καρτέλα επιλογών *AntiSpam*

Στην καρτέλα *AntiSpam* ο χρήστης μπορεί να επιλέξει τα μηνύματα που ανιχνεύονται ως *Spam* να διαγράφονται αυτόματα από τον φάκελο ληφθέντων και να μεταφέρονται στον κάδο ανακύκλωσης του *Thunderbird* [Εικόνα 5.3]. Ο αλγόριθμος ανίχνευσης χρησιμοποιεί πόντους για κάθε λέξη κλειδί που ανιχνεύεται. Οι πόντοι αθροίζονται και αν ξεπεράσουν το *Spam Score Threshold* που ορίζει εδώ ο χρήστης, το μήνυμα χαρακτηρίζεται ως *Spam*. Στους κανόνες *Spam* ο χρήστης μπορεί να αλλάξει τους πόντους που δίνονται σε περίπτωση που το μήνυμα δεν έχει παραλήπτη ή αντιθέτως έχει πολλαπλούς παραλήπτες. Στο *Spam Words* δίνονται οι λέξεις και οι πόντοι αυτών, οι οποίες λειτουργούν ως βιβλιοθήκη με *tokens* για τον αλγόριθμο ανίχνευσης. Ο χρήστης μπορεί να προσθέσει δικές του λέξεις ή φράσεις κλειδιά και να αλλάξει την τιμή του δείκτη *Spam* για αυτές τις λέξεις και φράσεις.



[Εικόνα 5.3] Το μενού επιλογών «*AntiSpam*»

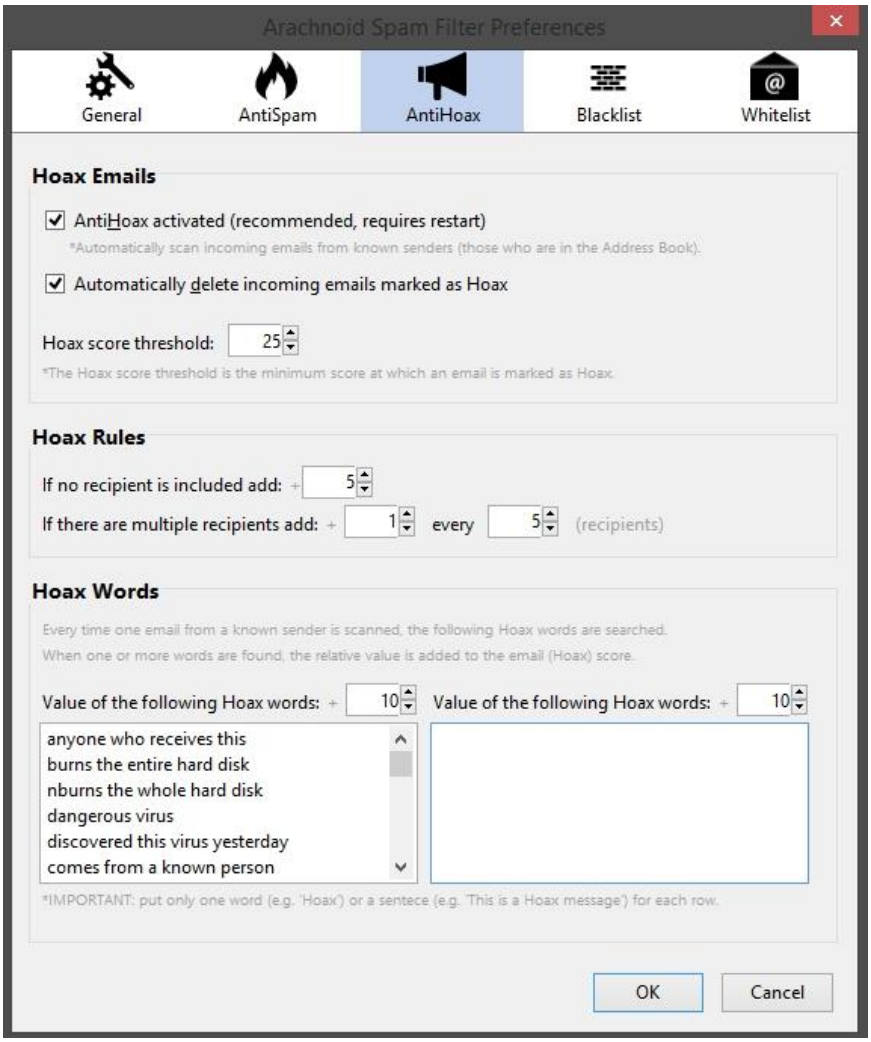
Κατά την άφιξη ενός μηνύματος αρχικά ελέγχεται αν είναι ενεργοποιημένος ο Έλεγχος *AntiSpam*. Στην περίπτωση που είναι ενεργοποιημένος το μήνυμα σαρώνεται όπως φαίνεται στο Παράρτημα 1.

Αν βρεθούν αρκετές λέξει κλειδιά, τότε το μήνυμα χαρακτηρίζεται ως *Spam* εκτελώντας τον κώδικα του Παραρτήματος 2.

Εφόσον ο χρήστης το έχει επιλέξει, το μήνυμα στην συνέχεια διαγράφεται εκτελώντας τον κώδικα του Παραρτήματος 3.

5.3 Η Καρτέλα επιλογών *AntiHoax*

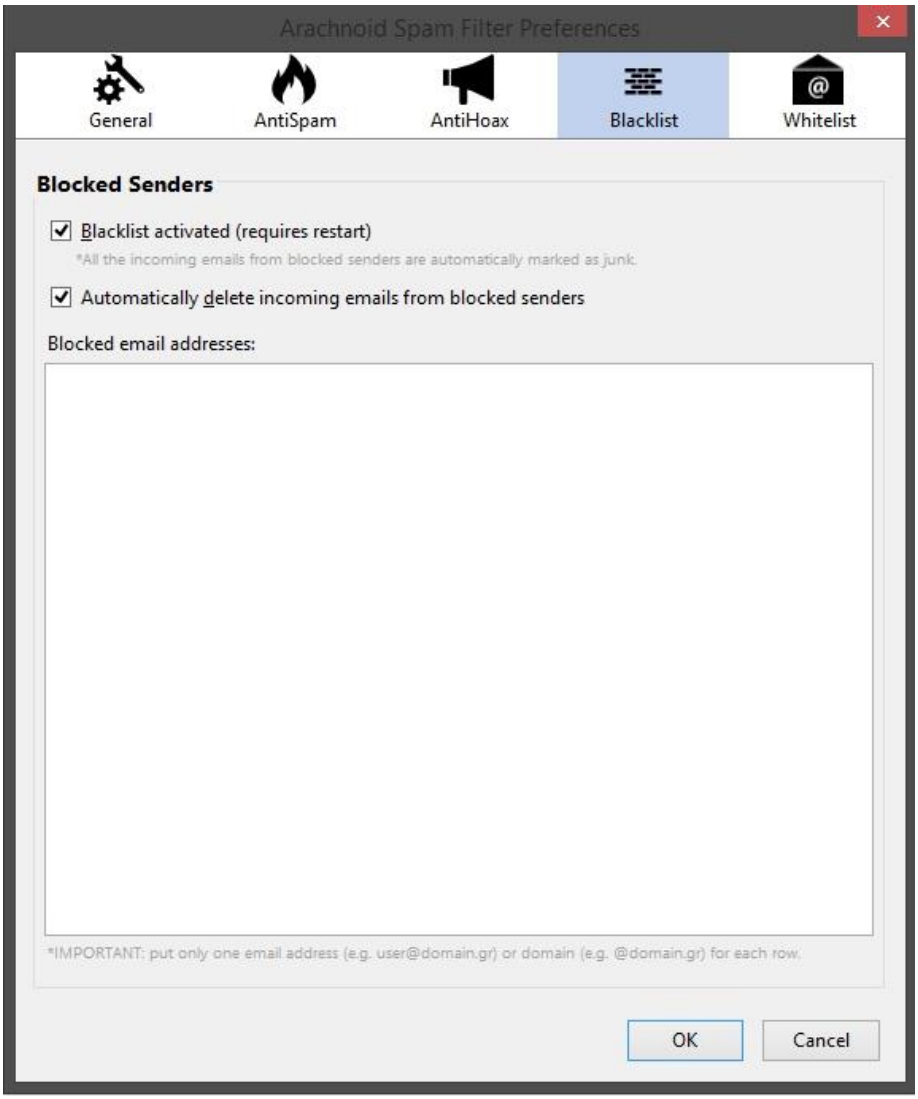
Στην καρτέλα *AntiHoax* ο χρήστης μπορεί να επιλέξει τα μηνύματα που ανιχνεύονται ως *Hoax* να διαγράφονται αυτόματα από τον φάκελο ληφθέντων και να μεταφέρονται στον κάδο ανακύκλωσης του *Thunderbird* [Εικόνα 5.4]. Ο αλγόριθμος ανίχνευσης χρησιμοποιεί και εδώ πόντους για κάθε λέξη κλειδί που ανιχνεύεται. Οι πόντοι αθροίζονται και αν ξεπεράσουν το *Hoax Score Threshold* που ορίζει ο χρήστης, το μήνυμα χαρακτηρίζεται ως *Hoax*. Στους κανόνες *Hoax* ο χρήστης μπορεί να αλλάξει τους πόντους που δίνονται σε περίπτωση που το μήνυμα δεν έχει παραλήπτη ή έχει πολλαπλούς παραλήπτες. Στο *Hoax Words* δίνονται οι λέξεις και οι πόντοι αυτών για τον αλγόριθμο ανίχνευσης. Ο χρήστης μπορεί να προσθέσει δικές του λέξεις ή φράσεις κλειδιά και να αλλάξει την τιμή του δείκτη *Hoax* για αυτές τις λέξεις και φράσεις. Για τον έλεγχο των εισερχόμενων μηνυμάτων εκτελείται ο ίδιος κώδικας όπως και κατά το έλεγχο *AntiSpam*.



[Εικόνα 5.4] Το μενού επιλογών «*AntiHoax*»

5.4 Η Καρτέλα επιλογών *Blacklist*

Στην καρτέλα *Blacklist* ο χρήστης μπορεί να επιλέξει τα μηνύματα των οποίων ο αποστολέας βρίσκεται σε Μαύρη Λίστα να διαγράφονται αυτόματα από τον φάκελο ληφθέντων και να μεταφέρονται στον κάδο ανακύκλωσης του *Thunderbird*[Εικόνα 5.5]. Ο χρήστης μπορεί επίσης να διαγράψει ή να προσθέσει δικές του διευθύνσεις ηλεκτρονικού ταχυδρομείου και *domains*.



[Εικόνα 5.5] Το μενού επιλογών «*Blacklist*»

Κατά την άφιξη ενός μηνύματος αρχικά ελέγχεται αν ο αποστολέας είναι στην *Blacklist* όπως φαίνεται στο Παράρτημα 4.

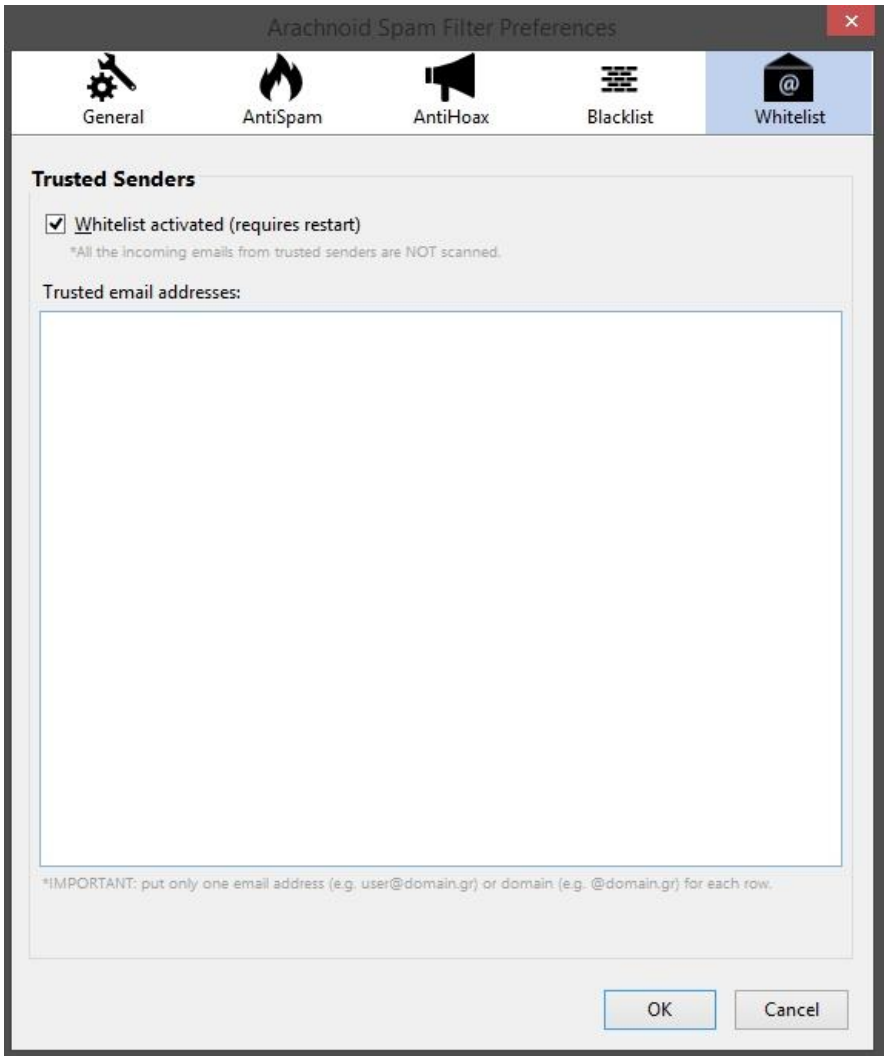
Αν βρεθεί η διεύθυνση του αποστολέα στην λίστα, τότε το μήνυμα χαρακτηρίζεται ως *Spam* όπως φαίνεται στο Παράρτημα 5.

Εφόσον ο χρήστης το έχει επιλέξει, το μήνυμα του μπλοκαρισμένου αποστολέα διαγράφεται εκτελώντας τον κώδικα του Παραρτήματος 6.

Όταν ο χρήστης επιλέγει να προσθέσει έναν αποστολέα στην *Blacklist* εκτελείται το κομμάτι κώδικα του Παραρτήματος 7.

5.5 Η Καρτέλα επιλογών *Whitelist*

Στην καρτέλα *Whitelist* ο χρήστης μπορεί να επιλέξει τα μηνύματα των οποίων ο αποστολέας εξαιρείται από την ανίχνευση για *Spam* και *Hoax* [Εικόνα 5.6]. Ο χρήστης μπορεί επίσης να διαγράψει ή να προσθέσει δικές του διευθύνσεις ηλεκτρονικού ταχυδρομείου και *domains*.



[Εικόνα 5.6] Το μενού επιλογών «*Whitelist*»

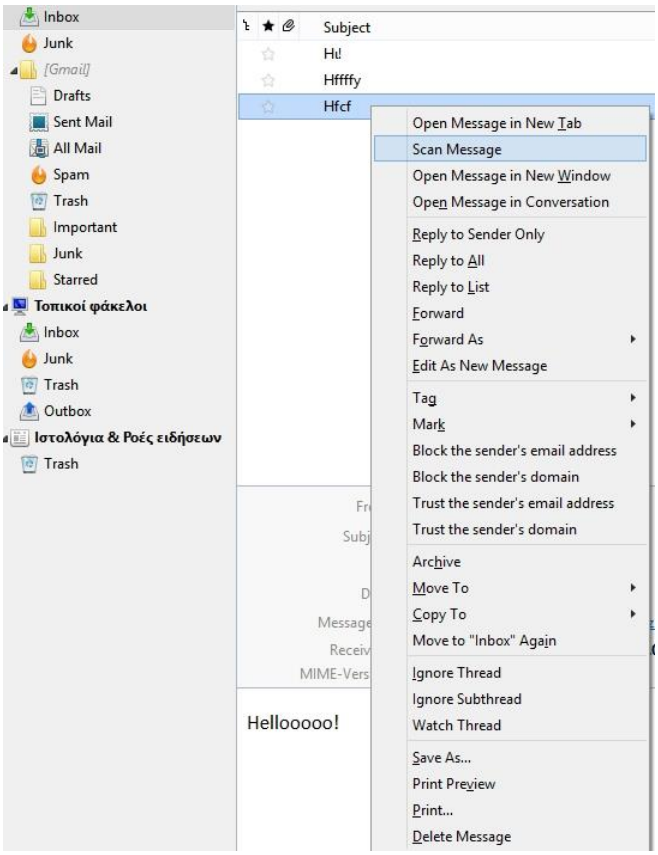
Κατά την άφιξη ενός μηνύματος αρχικά ελέγχεται αν ο αποστολέας είναι στην *Whitelist* όπως φαίνεται στο Παράρτημα 8.

Εφόσον βρεθεί ότι βρίσκεται στην *Whitelist* ο έλεγχος σταματά εκτελώντας τον κώδικα του Παραρτήματος 9.

Όταν ο χρήστης επιλέγει να προσθέσει έναν αποστολέα στην *Whitelist* εκτελείται το κομμάτι κώδικα του Παραρτήματος 10.

5.6 Το μενού Επιλογών

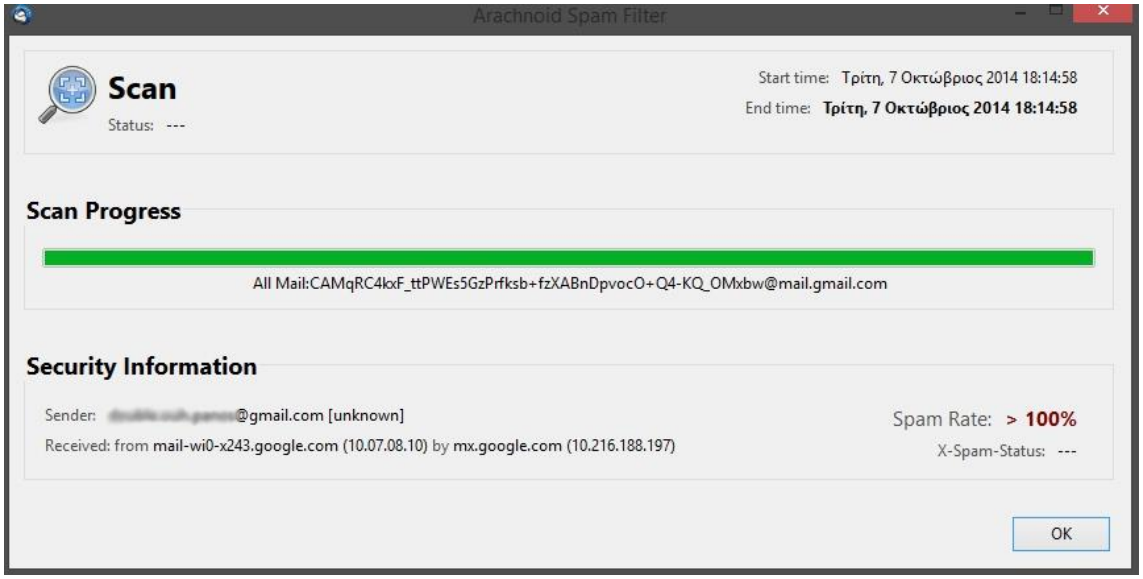
Με την επιλογή ενός ή πολλαπλών μηνυμάτων ή φακέλων ο χρήστης μπορεί, κάνοντας δεξί κλικ στα επιλεγμένα, να εκκινήσει διαδικασία σάρωσης για ανίχνευση μηνυμάτων *Spam*. Στην περίπτωση των μηνυμάτων, έχει επίσης την δυνατότητα να προσθέσει την διεύθυνση ηλεκτρονικού ταχυδρομείου ή το *domain* του αποστολέα στην Μαύρη Λίστα ή στην Άσπρη Λίστα [Εικόνα 5.7].



[Εικόνα 5.7] Η επιλογή «*Scan*»

5.7 Η Επιλογή *Scan*

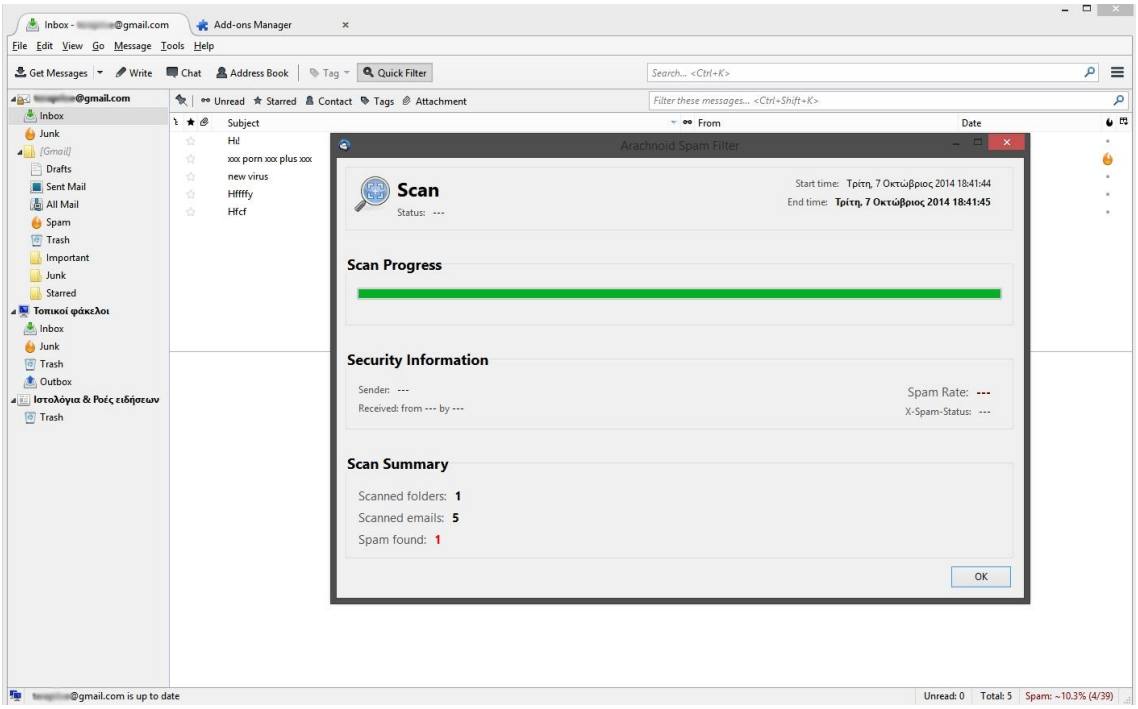
Η σάρωση ενός μηνύματος ολοκληρώνεται με την εμφάνιση σχετικού παράθυρου διεπαφής ενημέρωσης του χρήστη [Εικόνα 5.8]. Στο παράθυρο διεπαφής εμφανίζονται οι ημερομηνίες και ώρες εκκίνησης και τερματισμού της σάρωσης. Στο κέντρο βρίσκεται η μπάρα ολοκλήρωσης και κάτω από αυτή αναγράφεται ο φάκελος και ολόκληρη η διεύθυνση του μηνύματος που σαρώθηκε. Στις πληροφορίες Ασφαλείας αναγράφεται η διεύθυνση ηλεκτρονικού ταχυδρομείου του αποστολέα καθώς και οι διευθύνσεις προέλευσης του μηνύματος μαζί με τις *IP* τους. Το *Spam Rate* δηλώνει το ποσοστό της πιθανότητας το μήνυμα να είναι spam και το *X-Spam-Status* είναι ο δείκτης κατάστασης *Spam* του μηνύματος.



[Εικόνα 5.8] Το παράθυρο σάρωσης Μηνυμάτων

Στην σάρωση ενός φακέλου ολοκληρώνεται με την εμφάνιση αντίστοιχου παράθυρου διεπαφής ενημέρωσης του χρήστη [Εικόνα 5.9]. Στο παράθυρο διεπαφής εμφανίζονται οι ημερομηνίες και ώρες εκκίνησης και τερματισμού της σάρωσης. Στο κέντρο βρίσκεται η μπάρα ολοκλήρωσης με τις πληροφορίες Ασφαλείας να είναι κενές αφού δεν πρόκειται για μήνυμα και δεν υπάρχει αποστολέας. Τέλος, αναγράφεται το

αποτέλεσμα της σάρωσης με τον αριθμό των φακέλων και των μηνυμάτων που σαρώθηκαν και τον αριθμό των μηνυμάτων που ανιχνεύθηκαν ως *Spam*.

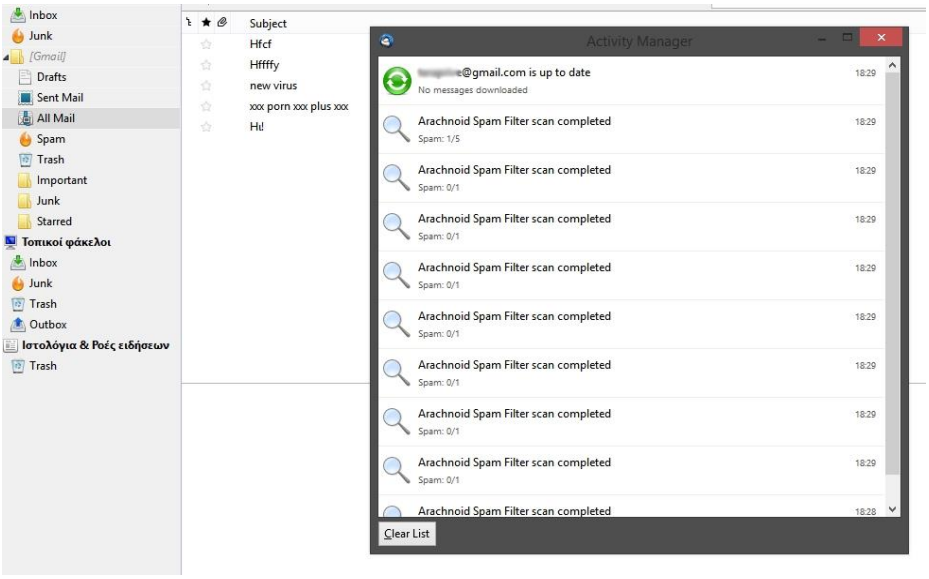


[Εικόνα 5.9] Το παράθυρο σάρωσης Φακέλων

Ο κώδικας που εκτελείται κατά την ενέργεια της σάρωσης φαίνεται στο Παράρτημα 11.

5.8 Ο Διαχειριστής Ενεργειών

Μετά την ολοκλήρωση κάθε ενέργειας, η εφαρμογή προσθέτει ανάλογο μήνυμα στον ενσωματωμένο *Activity Manager* όπου ο χρήστης μπορεί να ενημερωθεί για τις τελευταίες ενέργειες του φίλτρου καθώς και την ώρα που ολοκληρώθηκαν [Εικόνα 5.10].

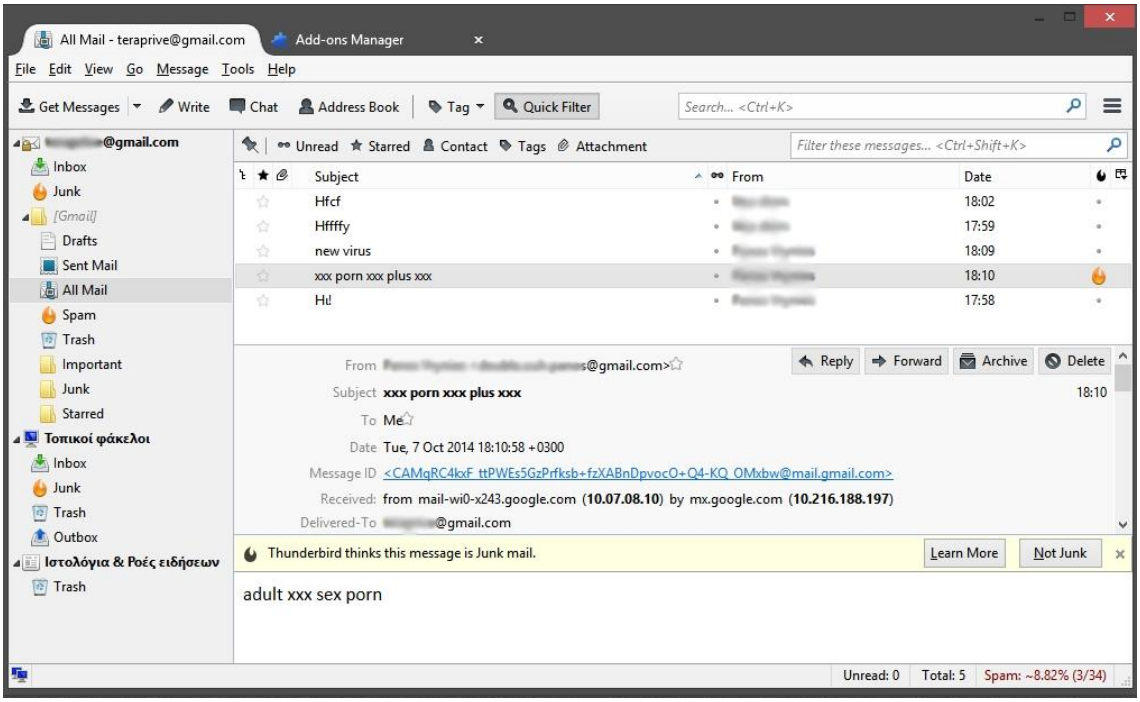


[Εικόνα 5.10] Ο Διαχειριστής Ενεργειών

Ο κώδικας που εκτελεί την προσθήκη των ενεργειών στον Διαχειριστή Ενεργειών παρουσιάζεται στο Παράρτημα 12.

5.9 Ο Μετρητής *Spam*

Με την άφιξη κάθε μηνύματος η εφαρμογή αυξάνει τον μετρητή ληφθέντων μηνυμάτων στην γραμμή κατάστασης του *Mozilla Thunderbird*. Σε αναλογία με τα μηνύματα *Ham* ο μετρητής μετρά και των αριθμών των ληφθέντων μηνυμάτων τα οποία χαρακτηρίστηκαν ως *Spam* και παρουσιάζει ένα ποσοστό λήψης *Spam* στον χρήστη [Εικόνα 5.11].



[Εικόνα 5.11] Ο μετρητής «*Spam*» δεξιά στο «*Status bar*»

Ο κώδικας που αυξάνει τον counter μετά την άφιξη κάθε μηνύματος ηλεκτρονικού ταχυδρομείου παρουσιάζεται στο Παράρτημα 13.

6

Αξιολόγηση Φίλτρων Spam

Ειδυλλιακά ένα φίλτρο Spam θα έπρεπε να αξιολογείται σε μια μεγάλη βάση δεδομένων με γνωστά Spam και Ham μηνύματα ηλεκτρονικού ταχυδρομείου. Κυρίως γιατί για λόγους εμπιστευτικότητας καλές και δημόσιες βάσης δεδομένων για δοκιμαστικούς σκοπούς δεν υπάρχουν. Πολλές από τις πιο γνωστές βάσεις περιέχουν μηνύματα που χρονολογούνται αρκετά χρόνια πριν.

Δυστυχώς, υπάρχει μία τάση στους χρήστες να συμφωνούν με τις αποφάσεις του φίλτρου, ειδικά σε μηνύματα τα οποία χαρακτηρίζονται ως Ham ή Spam οριακά και στα οποία ο χρήστης είναι αδιάφορος. Επιπλέον, ο αριθμός των λανθασμένα χαρακτηρισμένων ως Spam μηνυμάτων υποτιμάται από τους χρήστες αφού πολλές φορές διαγράφονται αυτόματα ή το τοποθετούνται στον φάκελο με τα υπόλοιπα Spam.

6.1 Αξιολόγηση συστήματος ταξινομητή

Η αξιολόγηση ενός συστήματος ταξινομητή είναι ένα κρίσιμο σημείο. Καμία πρόοδος δεν μπορεί να αναμένεται, αν δεν υπάρχει ένας τρόπος για να την δοκιμαστούν οι ήδη υπάρχουσες. Απαιτούνται τυποποιημένες μετρήσεις, συλλογές και διαδικασίες, οι οποίες να επιτρέπουν την σύγκριση διαφορετικών ερευνητικών έργων. Έχουν καθιερωθεί πολλές μέθοδοι αξιολόγησης, αλλά έχουν προσαρμοστεί και να βελτιωθεί καθώς αντιμετωπίζουν αυτό που ίσως είναι η κύρια δυσκολία στο φιλτράρισμα ανεπιθύμητης αλληλογραφίας, το πρόβλημα του ασύμμετρου κόστος εσφαλμένης ταξινόμησης. Το γεγονός ότι ένα νόμιμο μήνυμα ταξινομείται ως το spam, ένα ψευδώς θετικό, είναι πολύ πιο επιβλαβής από ότι ένα spam που ταξινομείται ως νόμιμο και

αξιόλογο, ένα ψευδώς αρνητικό. Αυτό συνεπάγεται ότι οι μετρήσεις αξιολόγησης πρέπει να αποδίδουν μεγαλύτερα βάρη σε χειρότερα λάθη.

Οι επιστημονικές αξιολογήσεις έχουν σαφώς καθορισμένες διαδικασίες, ενοποιημένες μετρήσεις, και δημόσιες συλλογές τεστ που κάνουν την σύγκριση των αποτελεσμάτων διαφορετικών ερευνητικών έργων σχετικά εύκολη. Η βάση των επιστημονικών πειραμάτων είναι να μπορούν να αναπαραχθούν. Υπάρχει ένας μεγάλος αριθμός από βιομηχανικές αξιολογήσεις, που εκτελούνται από τους ίδιους τους προγραμματιστές φίλτρων, αλλά και άλλες που εκτελούνται από εξειδικευμένα περιοδικά υπολογιστών. Οι περιορισμοί της βιομηχανικής αξιολόγησης περιλαμβάνουν αυτό-καθορισμένα κριτήρια, ιδιωτικές συλλογές δοκιμών, και αυτό-οριζόμενες μετρικές απόδοσης. Τα κύρια θέματα στην αξιολόγηση των φίλτρα *spam* είναι οι συλλογές δοκιμών, η λειτουργική διαδικασία, και οι μετρήσεις αξιολόγησης.

6.2 Συλλογές Δοκιμών

Μια συλλογή δοκιμών είναι ένα σύνολο ήδη διαβαθμισμένων μηνυμάτων που αποστέλλονται σε ένα ταξινομητή προκειμένου να μετρηθεί η αποτελεσματικότητά του, όσον αφορά τις επιτυχίες και τα λάθη. Είναι σημαντικό οι δοκιμαστικές συλλογές να είναι διαθέσιμες στο κοινό, ώστε να επιτρέπεται η σύγκριση των προσεγγίσεων και η βελτίωση της τεχνολογίας. Από την άλλη πλευρά, οι συλλογές μπορεί να περιλαμβάνουν ιδιωτικά μήνυμα, οπότε η προστασία της ιδιωτικής ζωής είναι ένα ζήτημα. Υπάρχει ένας αριθμός από εργασίες στις οποίες η συλλογή δοκιμής που χρησιμοποιείται φυλάσσονται ιδιωτικά, όπως είναι τα προσωπικά ηλεκτρονικά μηνύματα ενός συγγραφέα ή κάποιου που τα δώρισε σε ένα έργο αξιολόγησης με την προϋπόθεση να παραμείνουν απόρρητα [Graham, 2003]. Το πρόβλημα της ιδιωτικής ζωής μπορεί να λυθεί με διαφορετικές τρόπους:

- Εξυπηρετώντας μια επεξεργασμένη μορφή των μηνυμάτων που δεν επιτρέπει την ανακατασκευή τους. Αυτή η προσέγγιση έχει ακολουθηθεί στις *SpamBase*, *PU1* και *ECMLPKDD* 2006 *Discovery Challenge* δημόσιες συλλογές.

• Κατασκευάζοντας την συλλογή χρησιμοποιώντας μόνο μηνύματα από δημόσιες πηγές. Αυτή είναι η προσέγγιση στις *Lingspam* και *SpamAssassin* δημόσιες συλλογές.

• Κρατώντας την συλλογή στην ιδιοκτησία ενός έγκριτου φορέα που εκτελεί την δοκιμή για λογαριασμό των ερευνητών. Ο διαγωνισμός *TREC Spam Track* είναι ένα τέτοιο θεσμικό όργανο. Προσφέρει κάποιες συλλογές δοκιμών δημόσια, ενώ διατηρεί κάποιες άλλες ιδιωτικές.

Στις επόμενες παραγράφους, παρουσιάζονται οι συλλογές δοκιμών που είναι διαθέσιμες στο κοινό, επιλύοντας το πρόβλημα της ιδιωτικής ζωής:

• Το SpamBase23 είναι μια συλλογή μηνυμάτων ηλεκτρονικού ταχυδρομείου που περιέχει 4601 μηνύματα, από τα οποία τα 1813 (39%) έχουν χαρακτηριστεί ως *spam*. Η συλλογή παρέχεται σε προ-επεξεργασμένη μορφή, και περιπτώσεις της έχουν παρασταθεί ως διανύσματα 58 διαστάσεων. Τα πρώτα 48 χαρακτηριστικά είναι λέξεις που προέρχονται από τα αρχικά μηνύματα και επιλέχθηκαν ως οι πιο μη ισορροπημένες λέξεις για την κλάση *UCE*. Τα επόμενα 6 χαρακτηριστικά είναι τα ποσοστά εμφάνισης των ειδικών χαρακτήρων "?", "(", "[", "$" και "#" "!". Τα ακόλουθα 3 χαρακτηριστικά αντιπροσωπεύουν διαφορετικές εμφανίσεις κεφαλαίων γραμμάτων στο κείμενο των μηνυμάτων. Τέλος, το τελευταίο χαρακτηριστικό είναι η ετικέτα κλάσης. Το κύριο πρόβλημα της συλλογής είναι ότι έχει περάσει προ-επεξεργασία και κατά συνέπεια δεν είναι δυνατόν να καθοριστούν και να δοκιμάσουν άλλα χαρακτηριστικά εκτός από εκείνες που έχουν ήδη συμπεριληφθεί [Zelkowitz, 2011].

• Η συλλογή *PU1*, αποτελείται από 1099 μηνύματα, από τα οποία τα 481 (43%) είναι *spam* και τα 618 νόμιμα. Έχοντας παραληφθεί από τον Ίων Ανδρουτσόπουλο, έχουν υποστεί επεξεργασία για την αφαίρεση των συνημμένων και των ετικετών *HTML*. Προς σεβασμό για την ιδιωτική ζωή, στην δημοσία διαθέσιμη έκδοση του *PU1*, πεδία διαφορετικά από το "Θέμα:" των μηνυμάτων έχουν αφαιρεθεί, και κάθε διακριτικό (λέξη, αριθμός, σύμβολο στίξης, κλπ) στο κυρίως κείμενο ή στα θέματα των μηνυμάτων έχει αντικατασταθεί από ένα μοναδικό αριθμό, που παραμένει ο ίδιος σε όλα τα μηνύματα. Αυτός ο

μηχανισμός κρυπτογράφησης *Hash* κάνει αδύνατο να εκτελεστούν πειράματα με άλλες τεχνικές *tokenization* και χαρακτηριστικά πέρα αυτών που περιλαμβάνονται από τους συγγραφείς [Androutsopoulos, 2000γ].

• Η συλλογή *CML-PKDD 2006 Discovery Challenge* έχει συλλεχθεί από τους διοργανωτές του διαγωνισμού, προκειμένου να δοκιμαστεί το πώς μπορούν να βελτιωθούν οι ταξινομήσεις των *spam* χρησιμοποιώντας δεδομένα που δεν έχουν χαρακτηριστεί ακόμα [Bickel, 2006]. Είναι διαθέσιμη σε μεταποιημένη μορφή και οι λέξεις που εμφανίζονται λιγότερες από τέσσερις φορές στην συλλογή έχουν αφαιρεθεί. Κάθε μήνυμα αντιπροσωπεύεται από ένα διάνυσμα που υποδεικνύει τον αριθμό των εμφανίσεων του κάθε χαρακτηριστικού στο μήνυμα.

• Η *Lingspam* συλλογή δοκιμών που χρησιμοποιείται σε πολλές πρόσφατες μελέτες έχει κατασκευαστεί από την ανάμιξη *spam* μηνυμάτων με μηνύματα που εξάγονται από δημόσια αρχεία λιστών αλληλογραφίας χωρίς *spam* [Androutsopoulos, 2000α]. Πιο συγκεκριμένα τα νόμιμα μηνύματα έχουν εξαχθεί από τη λίστα *Linguist*, μία διατηρήσιμη και ως εκ τούτου, χωρίς *spam* λίστα, για το επάγγελμα και την επιστήμη της γλωσσολογίας. Ο αριθμός των νόμιμων μηνυμάτων είναι 2412 και ο αριθμός των μηνυμάτων ανεπιθύμητης αλληλογραφίας είναι 481 (16%).

• Η δημόσια συλλογή *SpamAssassin* έχει συλλεχθεί από τον *Justin Mason*, έναν από τους προγραμματιστές του, με τις δημόσιες συνεισφορές πολλών άλλων και αποτελείται από 6047 μηνύματα, από τα οποία τα 1897 (31%) είναι *spam*. Τα νόμιμα μηνύματα έχουν διαιρεθεί περαιτέρω σε εύκολα και δύσκολα, εφόσον κάνουν χρήση HTML, έγχρωμου κειμένου και λέξεων που θα μπορούσαν να συσχετίζονται με *spam*. Καθώς είναι σχετικά μεγάλη, ρεαλιστική και δημόσια, έχει γίνει το πρότυπο για την αξιολόγηση φίλτρων *spam*.

Εκτός από αυτές τις συλλογές, το *TREC Spam Track* διαθέτει πολλές δημόσιες και ιδιωτικές συλλογές δοκιμών, όπως η Δημόσια Συλλογή *TREC*, η «*trec05p-1*». Οι περισσότερες συλλογές *TREC* έχουν δύο πολύ μοναδικές και ενδιαφέρουσες ιδιότητες:

1. Τα μηνύματα είναι ταξινομημένα με χρονολογική σειρά, επιτρέποντας την δοκιμή της σταδιακής μάθησης και που αναφέρεται ως διαδικτυακή δοκιμή

2. Έχουν κατασκευαστεί χρησιμοποιώντας μια σταδιακή διαδικασία στην οποία τα μηνύματα έχουν χαρακτηριστεί χρησιμοποιώντας διάφορα φίλτρα *anti-spam*, και όπου αυτό δεν ήταν δυνατό, από τους ίδιους τους χρήστες χειροκίνητα [Cormack, 2005].

Οι συλλογές *TREC* είναι επίσης πολύ μεγάλες σε σύγκριση με τις προηγούμενες, επιτρέποντας στους ερευνητές να φθάνουν σε πιο αξιόπιστα συμπεράσματα. Το *TREC Spam Track* έχει καθορίσει το πρότυπο αξιολόγησης για τα *learning-based* φίλτρα *anti-spam*.

6.3 Η Διαδικασίας Δοκιμής

Η πιο συχνή διαδικασία αξιολόγησης για τα φίλτρα *spam* είναι η αξιολόγηση παρτίδας. Το φίλτρο εκπαιδεύτηκε σε ένα σύνολο μηνυμάτων και εφαρμόστηκε σε ένα διαφορετικό σύνολο μηνυμάτων για δοκιμή. Τα δοκιμαστικά μηνύματα επισημαίνονται ως *Ham* ή *Spam*, και είναι δυνατόν να συγκριθεί η απόφαση του φίλτρου υπολογίζοντας τις επιτυχίες και τα λάθη. Είναι σημαντικό, η συλλογή δοκιμής να είναι παρόμοια, αλλά και διαφορετική από αυτήν που εκπαίδευσε το φίλτρο ώστε να εξομοιώνει όσο πιο πιστά γίνεται τις πραγματικές συνθήκες. Αυτή είναι η διαδικασία δοκιμής που χρησιμοποιείται στις περισσότερες αξιολογήσεις χρησιμοποιώντας τις *SpamBase, PU1* και *Linspam* συλλογές δοκιμής.

Μία βελτίωση της αξιολόγησης αυτής είναι να εκτελεστεί μία διασταυρωμένη επικύρωση Ν φορές. Αντί της χρήσης μιας χωριστής συλλογής δοκιμής, τμήματα της ήδη επισημασμένης χρησιμοποιούνται μερικές φορές για την εκπαίδευση καθώς και την δοκιμή. Εν ολίγοις, η επισημασμένη συλλογή διαιρείται τυχαία σε Ν ομάδες ή πτυχώσεις και στην συνέχεια εκτελούνται Ν δοκιμές, χρησιμοποιώντας Ν-1 ομάδες ως σύνολα εκπαίδευσης και η τελευταία ως σύνολο δοκιμής. Μετά από Ν δοκιμές λαμβάνεται ο μέσος όρος των αποτελεσμάτων, οδηγώντας σε πιο έγκυρα στατιστικά στοιχεία, αφού το πείραμα δεν εξαρτάται από τα απρόβλεπτα χαρακτηριστικά των δεδομένων. Τα ίδια δεδομένα χρησιμοποιούνται για την εκπαίδευση και την δοκιμή.

Αυτή η διαδικασία έχει πραγματοποιηθεί αρκετές φορές στην αξιολόγηση φίλτρων ανεπιθύμητης αλληλογραφίας [Gomez, 2002].

Μια μεγάλη κριτική σε αυτή την προσέγγιση είναι ότι η συνηθισμένη λειτουργία των φίλτρων *Spam* τους επιτρέπει να μαθαίνουν από τα λάθη που έκαναν αν ο χρήστης τους τα εντοπίσει και τα αναφέρει σαν λανθασμένα θετικά. Η παρτίδα αξιολόγησης δεν επιτρέπει τα φίλτρα για να μάθουν καθώς κατατάσσουν, και αγνοεί την χρονολογική σειρά. Αντίθετα, οι αξιολογητές τύπου *TREC* προσέγγισαν τις δοκιμές φίλτρων ως μία διαδικτυακή διεργασία μάθησης στην οποία τα μηνύματα παρουσιάζονται στο φίλτρο, ένα κάθε φορά, σε χρονολογική σειρά. Για κάθε μήνυμα, το φίλτρο προβλέπει την τάξη, *spam* ή *ham,* υπολογίζοντας μια βαθμολογία S η οποία συγκρίνεται με ένα σταθερό αλλά αυθαίρετο κατώφλι T. Αμέσως μετά την πρόβλεψη, η πραγματική τάξη παρουσιάζεται στο φίλτρο, έτσι ώστε να μπορεί να χρησιμοποιήσει αυτή την πληροφορία σε μελλοντικές προβλέψεις.

Η παραπάνω διαδικασία αξιολόγησης υποστηρίζεται από ένα συγκεκριμένο σύνολο εντολών που απαιτεί από ένα φίλτρο να ενσωματώσει τις παρακάτω λειτουργίες γραμμής εντολών:

- *Initialize* - δημιουργεί όλα τα αρχεία ή τους διακομιστές που είναι απαραίτητοι για τη λειτουργία του φίλτρου.
- *Classify message* - επιστρέφει *Ham/Spam* ταξινόμηση και βαθμολογία *spamminess* για το μήνυμα.
- *Train ham message* - ενημερώνει το φίλτρο για την σωστή *Ham* ταξινόμηση προηγούμενων μηνυμάτων.
- *Train spam message* - ενημερώνει το φίλτρο για την σωστή *Spam* ταξινόμηση προηγούμενων μηνυμάτων.
- *Finalize* - αφαιρεί όλα τα αρχεία ή τους διακομιστές που δημιουργήθηκαν από το φίλτρο.

Αυτή είναι η τυπική διαδικασία που χρησιμοποιείται στην αξιολόγηση *TREC*. Ένα ανοικτό ερώτημα είναι αν οι δύο μέθοδοι είναι ισοδύναμες, με βάση τα αποτελέσματα που λαμβάνονται λειτουργώντας σε πραγματικές συνθήκες. Οι *Cormack* και *Bratko* πραγματοποίησαν μια συστηματική σύγκριση μεταξύ αλγόριθμων με

κορυφαίες επιδόσεις, ακολουθώντας και τις δύο διαδικασίες, και έφτασαν στο συμπέρασμα ότι οι καλύτερες είναι οι μέθοδοι συμπίεσης, και ότι η on-line διαδικασία είναι πιο κατάλληλη, διότι είναι πιο κοντά σε λειτουργικές ρυθμίσεις [Cormack, 2006].

6.4 Μετρικές Αξιολόγησης

Οι μετρικές που χρησιμοποιούνται στις μελέτες δοκιμής Φίλτρων *Spam* χωρίζονται σε τρεις ομάδες: η βασική μετρική που χρησιμοποιείται στα αρχικά έργα, η μέθοδος *ROCCH* ως μία προχωρημένη μέθοδος για πρωταρχικά λάθη, και οι μετρικές *TREC* ως το ισχύον πρότυπο.

6.4.1. Βασικές Μετρικές

Η αποτελεσματικότητα των συστημάτων φιλτραρίσματος ανεπιθύμητης αλληλογραφίας μετράται με γνώμονα τον αριθμό των σωστών και των λανθασμένων αποφάσεων. Ας υποθέσουμε ότι το φίλτρο ταξινομεί ένα δεδομένο αριθμό μηνυμάτων. Μπορούμε να συνοψίσουμε τη σχέση ανάμεσα στις ταξινομήσεις συστήματος και τις σωστές αποφάσεις σε έναν πίνακας συσχέτισης [Εικόνα 6.1]. Κάθε καταχώρηση στο πίνακα καθορίζει τον αριθμό των εγγράφων με το συγκεκριμένο αποτέλεσμα. Για το πρόβλημα φιλτραρίσματος του *Spam*, λαμβάνουμε το *Spam* ως την θετική κλάση (+), και το *Ham* ως την αρνητική (-). Σε αυτόν τον πίνακα, το κλειδί "*TP*" σημαίνει "αριθμός των αληθινών θετικών αποφάσεων», και τα "*tn*", "*fp*" και "*fn*" αναφέρονται στον αριθμό των «αληθινά αρνητικών», «ψευδώς θετικών» και «Ψευδώς αρνητικών» αποφάσεων, αντίστοιχα. Οι περισσότερες παραδοσιακές μετρικές αξιολόγησης *TC* μπορούν να οριστούν με βάση τις εγγραφές του πίνακα συσχέτισης.

	+	-
+	tp	fp
-	fn	tn

[Εικόνα 6.1] Τυπικός Πίνακας Συσχέτισης

Η F_1 είναι μία μέτρηση που δίνει ίση σημασία στην ανάκληση και την ακρίβεια [Sebastiani, 2002]. Η Ανάκληση, *Recall*, ορίζεται ως το ποσοστό των μελών της κατηγορίας που έχουν ανατεθεί σε μια κατηγορία από ένα ταξινομητή. Η Ακρίβεια, *Precision*, ορίζεται ως το ποσοστό των ορθώς ταξινομημένων εγγράφων σε μια κατηγορία. Λαμβάνοντας υπόψη μία γραφική παράσταση όπως αυτή που φαίνεται στον πίνακα, η ανάκληση (R), η ακρίβεια (P) και η F1 υπολογίζονται με τους ακόλουθους τύπους:

$$R = \frac{tp}{tp + fn}$$

$$R = \frac{tp}{tp + fp}$$

$$F_1 = \frac{2RP}{R + P}$$

Οι μετρικές Ανάκληση και Ακρίβεια έχουν χρησιμοποιηθεί σε διάφορα έργα σχετικά με το φιλτράρισμα *Spam* [Androutsopoulos, 2000α, Androutsopoulos 2000β, Androutsopoulos, 2000γ]. Άλλα έργα κάνουν χρήση των τυπικών μετρικών ML, όπως η Ακρίβεια, *Accuracy* και το Λάθος, *Error* [Provost, 1999]. Για τον τελικό χρήστη τα λάθη ταξινόμησης δεν έχουν όλα την ίδια σημασία. Τυπικά, το σφάλμα της ταξινόμησης ενός έννομου μηνύματος ως ανεπιθύμητη αλληλογραφία, ένα ψευδώς θετικό λάθος, είναι πολύ πιο επικίνδυνο από την ταξινόμηση ενός μηνύματος *Spam ως* νόμιμο, ψευδώς αρνητικό. Αυτή η παρατήρηση μπορεί να ξανά-εκφραστεί ως το κόστος ενός ψευδώς θετικού είναι μεγαλύτερο από το κόστος ενός ψευδώς αρνητικού στο πλαίσιο της ταξινόμησης *Spam*. Τα κόστη εσφαλμένης ταξινόμησης συνήθως εκπροσωπούνται ως ένας πίνακας συσχέτισης του κόστους στην οποία η καταχώρηση C (A, B) σημαίνει το κόστος της λήψης μίας A απόφασης όταν η σωστή απόφαση είναι B, που είναι το κόστος του A δεδομένου B (κόστος (A | B)). Για παράδειγμα, η C (+, -) είναι το κόστος μιας ψευδώς θετικής απόφασης και η C (-, +) είναι το κόστος μιας ψευδώς αρνητικής απόφασης.

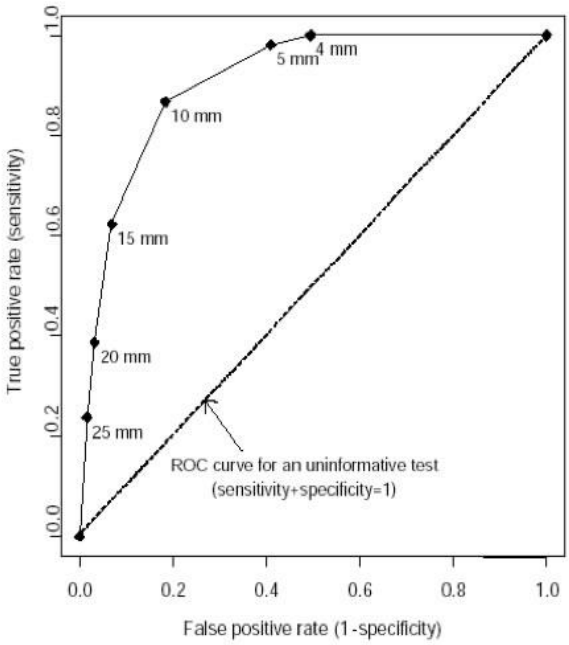
6.4.2. Η μέθοδος *ROCCH*

Η ανάλυση Λειτουργικών Χαρακτηριστικών Δέκτη (*Receiver Operating Characteristics - ROC*) είναι μια μέθοδος για την αξιολόγηση και τη σύγκριση των επιδόσεων των ταξινομητών. Έχει χρησιμοποιηθεί εκτενώς στην ανίχνευση σημάτων, και παρουσιάστηκε και επεκτάθηκε από τους *Provost* και *Fawcett* στην κοινότητα Μηχανικής Μάθησης [Provost, 1997]. Στην ανάλυση *ROC*, αντί για μια απλή τιμή της ακρίβειας, ένα ζευγάρι τιμών καταγράφεται για διαφορετικές συνθήκες τάξης και κόστους στις οποίες μαθαίνει ένας ταξινομητής. Οι τιμές που καταγράφονται είναι το ποσοστό ψευδώς θετικών αποτελεσμάτων (*FP*) και το ποσοστό πραγματικά θετικών (*TP*), που ορίζονται στον πίνακα συσχέτισης ως:

$$FP = \frac{fp}{fp + tn}$$

$$TP = \frac{tp}{tp + fn}$$

Το ποσοστό TP είναι ισοδύναμη με την ανάκληση της θετικής κλάσης, ενώ ο ρυθμός FP είναι που ισοδυναμεί με 1 μείον την ανάκληση της αρνητικής τάξης. Κάθε (FP, TP) ζεύγος παρίσταται ως σημείο στο χώρο ROC. Οι περισσότεροι αλγόριθμοι ML παράγουν διαφορετικούς ταξινομητές σε διαφορετικές συνθήκες τάξης και κόστους. Για αυτούς τους αλγορίθμους, οι συνθήκες μεταβάλλονται για να αποκτήσουν μια Καμπύλη ROC [Εικόνα 6.2].



[Εικόνα 6.2] Καμπύλη *ROC*

Ένα σημείο σε ένα διάγραμμα *ROC* κυριαρχεί ένα άλλο αν είναι πάνω και προς τα αριστερά, δηλαδή έχει υψηλότερο *TP* και χαμηλότερο *FP*. Κυριαρχία συνεπάγεται ανώτερη απόδοση για μια ποικιλία κοινών μέτρων απόδοσης, συμπεριλαμβανομένων του αναμενόμενου κόστους , της Ανάκλησης και άλλων. Λαμβάνοντας υπόψη ένα σύνολο από καμπύλες *ROC* για διάφορους αλγορίθμους *ML*, αυτός που είναι πιο κοντά στην αριστερή ανώτερη γωνία του χώρου *ROC* αντιπροσωπεύει τον καλύτερο αλγόριθμο.

Η κυριαρχία σπάνια αποκτάται κατά τη σύγκριση των καμπυλών *ROC*. Αντ 'αυτού, είναι δυνατόν να υπολογιστεί μια σειρά από προϋποθέσεις υπό τις οποίες ένας αλγόριθμος *ML* θα παράγει τουλάχιστον καλύτερα αποτελέσματα από τους άλλους αλγορίθμους. Αυτό γίνεται μέσω της μεθόδου *ROC Convex Hull* [Provost, 1997]. Συνοπτικά, δίνεται ένα σύνολο από (*FP, TP*) σημεία, έτσι ώστε να μην βρίσκονται στο άνω κυρτό σημείο που αντιστοιχεί σε ικανοποιητικούς ταξινομητές για οποιεσδήποτε συνθήκες τάξης και το κόστους. Κατά συνέπεια, δεδομένης μιας καμπύλης *ROC*, μόνο το άνω κυρτό σημείο μπορεί να είναι βέλτιστο, και τα υπόλοιπο σημεία μπορούν να απορριφθούν. Επίσης, για μια σειρά από καμπύλες *ROC*, μόνο ένα μικρό κομμάτι της καθεμίας που βρίσκεται στο άνω κυρτό σημείο τους διατηρείται, οδηγώντας σε ένα

εύρος κλίσης στην οποία ο αλγόριθμος *ML* που αντιστοιχεί στην καμπύλη παράγει τους πιο καλούς ταξινομητές απόδοσης. Ένα παράδειγμα των καμπυλών *ROC* παρουσιάζεται στην εικόνα 6.4.2.

Η *ROC* ανάλυση επιτρέπει μια οπτική σύγκριση των επιδόσεων ενός συνόλου *ML* αλγορίθμων, ανεξάρτητα από τις συνθήκες τάξης και κόστους. Με αυτό τον τρόπο, η απόφαση του ποιος είναι ο καλύτερος ταξινομητής ή αλγόριθμος *ML* μπορεί να καθυστερήσει μέχρι οι στοχευόμενες συνθήκες, του πραγματικό κόσμου, να γίνουν γνωστές, και ταυτόχρονα να μπορούν να ληφθούν πολύτιμες πληροφορίες. Στην πιο πλεονεκτική περίπτωση, ένας αλγόριθμος θα είναι κυρίαρχος σε όλο το εύρος κλίσης. Συνήθως, διάφοροι αλγόριθμοι ML θα οδηγούν σε ταξινομητές που βέλτιστοι, μεταξύ αυτών που δοκιμάζονται, για διαφορετικά εύρη κλίσης, που αντιστοιχούν σε διαφορετικές συνθήκες τάξης και κόστους.

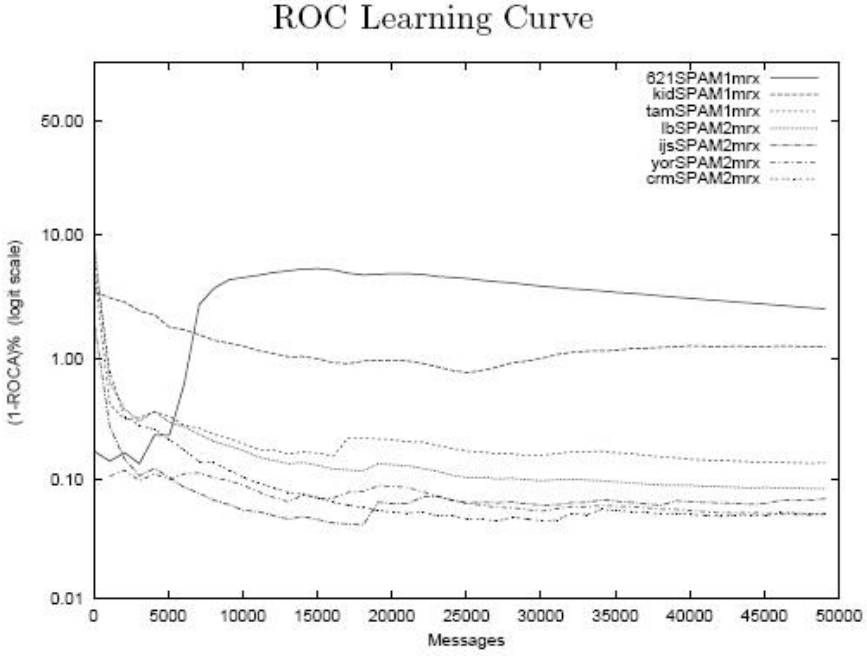
Λειτουργικά, η μέθοδος *ROCCH* αποτελείται από τα ακόλουθα στάδια:

1. Για κάθε αλγόριθμο *ML*, λαμβάνεται μια καμπύλη ROC και σχεδιάζεται ολόκληρη ή μόνο το κυρτό της σημείο στο χώρος *ROC*.
2. Λαμβάνεται το κυρτό σημείο του συνόλου των καμπυλών ROC που είχαν σχεδιαστεί νωρίτερα.
3. Υπολογίζεται το εύρος κλίσης για το οποίο κάθε καμπύλη *ROC* βρίσκεται στο κυρτό σημείο.
4. Σε περίπτωση που οι στοχευόμενες συνθήκες είναι γνωστές, υπολογίζεται η αντίστοιχη τιμή κλίσης και παράγεται ο καλύτερος αλγόριθμος. Σε άλλη περίπτωση, παράγονται όλες οι τιμές και οι καλύτεροι τοπικοί αλγόριθμοι ή ταξινομητές.

Έχει γίνει χρήση της μεθόδου *ROCCH* για την αξιολόγηση μιας ποικιλίας *ML* αλγορίθμων για το πρόβλημα του φιλτραρίσματος του spam στην έρευνα του *Gomez* [Gomez, 2002]. Αυτή ήταν και η πρώτη φορά που *ROC* καμπύλες χρησιμοποιήθηκαν σε δοκιμές για την αξιολόγηση φίλτρων ανεπιθύμητων μηνυμάτων. Έκτοτε έχουν γίνει ένα πρότυπο στις *TREC* αξιολογήσεις.

6.4.3. Μετρικές TREC

Οι μετρικές *TREC Spam Track* θεωρούνται ένα πρότυπο στην αξιολόγηση των φίλτρων *Spam* [Cormack, 2005]. Η κύρια βελτίωση σε σχέση με τη μέθοδο *ROC* είναι η προσαρμογή της στην διαδικτυακή διαδικασίας αξιολόγησης. Καθώς η διαδικτυακή αξιολόγηση επιτρέπει την απευθείας εκμάθηση του φίλτρου από τις λανθασμένες ταξινομήσεις, οι ρυθμοί (*TP*, *FP*) του αλλάζουν συνεχώς, και οι τιμές τους παρουσιάζουν βελτίωση με το χρόνο. Το γράφημα *ROC* μετασχηματίζεται σε ένα ενιαίο αριθμητικό σχήμα, υπολογίζοντας την περιοχή κάτω από την καμπύλη *ROC* (*Area Under ROC - AUC*). Καθώς τα φίλτρα προς αξιολόγηση είναι εξαιρετικά αποτελεσματικά, είναι καλύτερα να αναφέρετε η αντίστροφη 1-*AUC* τιμή, η οποία υπολογίζεται διαχρονικά όταν η διαδικτυακή διαδικασία μάθησης-δοκιμής είναι ενεργή. Με αυτό τον τρόπο, είναι δυνατόν να εξεταστεί πόσο γρήγορα μαθαίνουν τα συστήματα, καθώς και σε ποία επίπεδα σφάλματος οι επιδόσεις φτάνουν σε τέλμα. Στην Εικόνα 6.3 παρουσιάζεται ένα γράφημα μιας *ROC* καμπύλης εκμάθησης [Cormack, 2005]. Αρχικά τα φίλτρα διαπράττουν ένα σχετικά υψηλό αριθμό λαθών κατά την έναρξη της *ROC* καμπύλης μάθησης και στην συνέχεια βελτιώνουν το αποτέλεσμά τους, φτάνοντας, τελικά, στην επίτευξη ενός μέσου επιπέδου απόδοσης.



[Εικόνα 6.3] *ROC* Καμπύλη μάθησης

6.5 Πειραματική Αξιολόγηση Φίλτρου *Arachnoid*

Σε αυτό το κεφάλαιο παρουσιάζονται τα αποτελέσματα των πειραμάτων που εκτελέστηκαν στη διάρκεια της πτυχιακής εργασίας και αναλύονται τα συμπεράσματα που προέκυψαν. Οι δοκιμές έγιναν με διαφορετικές ρυθμίσεις ευαισθησίας του φίλτρου και σε συνδυασμό με το ενσωματωμένο φίλτρο του *Thunderbird* ή χωρίς. Στα πειράματα της παρούσας εργασίας χρησιμοποιήθηκε το σύνολο μηνυμάτων *Enron-Spam* που περιέχει επιθυμητά και ανεπιθύμητα μηνύματα έξι ψευδό-χρηστών. Στα μηνύματα της παραπάνω συλλογής αξιολογούμε μόνο το σώμα και το θέμα του κάθε μηνύματος, αγνοώντας συνημμένα αρχεία και ετικέτες HTML και κεφαλίδες.

6.5.1 Περιγραφή Πειραματικής μεθόδου

Για την αναπαράσταση των αποτελεσμάτων θα χρησιμοποιηθούν πίνακες συσχέτισης. Αυτό το είδος αναπαράστασης αποτελεσμάτων εστιάζει στην προγνωστική ικανότητα του μοντέλου και όχι στο πόσο γρήγορα εκτελεί την ταξινόμηση, την επεκτασιμότητα του, κ.λ.π..

Ο πίνακας συσχέτισης παρουσιάζεται από έναν πίνακα όπου κάθε σειρά αντιπροσωπεύει τις περιπτώσεις σε μια προβλεπόμενη κατηγορία, ενώ κάθε στήλη αντιπροσωπεύει την πραγματική κατηγορία [Εικόνα 6.4]. Ο πίνακας δείχνει επίσης την ακρίβεια του ταξινομητή ως προς το ποσοστό των ορθώς ταξινομημένων μηνυμάτων σε μία δεδομένη κατηγορία διαιρούμενο με τον συνολικό αριθμό των μηνυμάτων σε αυτή την κατηγορία. Η συνολική (μέση) ακρίβεια του ταξινομητή επίσης αξιολογείται με τη χρήση του πίνακα συσχέτισης [16].

	Spam (Prediction)	Ham (Prediction)	Accuracy
Spam (Actual)	True Positive	False Negative	Percentage
Ham (Actual)	False Positive	True Negative	Percentage
Overall Accuracy	--	--	Overall Percentage

[Εικόνα 6.4] Πίνακας Συσχετίσεων ενός μοντέλου ταξινομητή *Spam*

Από τα δυαδικά προβλήματα ταξινόμησης, όπως στην περίπτωση των πειραμάτων της παρούσας εργασίας, μπορούν να προκύψουν δύο εξισώσεις μέσα από τις μετρικές *TP, FP, FN, TN*. Οι εξισώσεις που προκύπτουν ονομάζονται ευαισθησία (*Sensitivity - TPR*) και αντιπροσωπεύει το ποσοστό ανάκλησης ανεπιθύμητων μηνυμάτων *(Spam Recall, SR)*, και σαφήνεια (*Specificity - TNR*), το ποσοστό ανάκλησης επιθυμητών μηνυμάτων *(Ham Recall, HR)*. Συνήθως, χρησιμοποιούνται για την αξιολόγηση κάθε δυαδικού ταξινομητή [16].

Η σαφήνεια (*TNR*) μετρά το ποσοστό των μηνυμάτων που είναι αρνητικά (*TN*) επί του συνόλου των μηνυμάτων που είναι πραγματικά αρνητικά (*TN + FP*). Μπορεί να θεωρηθεί ως η πιθανότητα το μήνυμα να έχει ταξινομηθεί ως αρνητικό, δεδομένου ότι δεν περιέχει αρνητικές λέξεις. Με υψηλότερη σαφήνεια, λιγότερα θετικά μηνύματα χαρακτηρίζονται ως αρνητικά [16].

Από την άλλη πλευρά, η ευαισθησία (*TPR*) είναι η αναλογία των μηνυμάτων που είναι θετικά (*TP*) επί του συνόλου των μηνυμάτων που είναι πραγματικά θετικά (*TP + FN*). Μπορεί να θεωρηθεί ως η πιθανότητα το μήνυμα να είναι θετικό, δεδομένου

ότι περιέχει θετικές λέξεις. Με υψηλότερη ευαισθησία, λιγότερα πραγματικά μηνύματα χαρακτηρίζονται ως αρνητικά [16].

Η ευαισθησία μπορεί να εκφρασθεί ως εξής:

- *SR = TP / (TP+FN)*

Και η σαφήνεια ως:

- *HR = TN / (TN+FP)*

Όπου:

- *TP* : ***T**rue **P**ositives*, δηλαδή πόσα ανεπιθύμητα μηνύματα κατετάγησαν σωστά
- *TN* : ***T**rue **N**egatives*, δηλαδή πόσα επιθυμητά μηνύματα κατετάγησαν σωστά
- *FP*: ***F**alse **P**ositive*, αριθμός ανεπιθύμητων μηνυμάτων που κατετάγησαν λανθασμένα
- *FN*: ***F**alse **N**egative*, αριθμός επιθυμητών μηνύματα που κατετάγησαν λανθασμένα

Σε γενικές γραμμές, Ευαισθησία TPR σημαίνει ακρίβεια κατά την ταξινόμηση στην αρνητική κατηγορία, και Σαφήνεια TNR νοείται η ακρίβεια κατά την ταξινόμηση στην θετική κατηγορία.

6.5.2 Αποτελέσματα Πειραμάτων

Σε κάθε πείραμα χρησιμοποιήθηκαν 300 μηνύματα, εκ των οποίων 200 ήταν *Spam* και 100 ήταν *Ham*. Στους παρακάτω πίνακες φαίνονται οι ταξινομήσεις του φίλτρου του *Mozilla Thunderbird* χωρίς την προσθήκη της επέκτασης *Arachnoid*, οι ταξινομήσεις της επέκτασης, και οι ταξινομήσεις του φίλτρου του *Mozilla Thunderbird* σε συνδυασμό με το φίλτρο της επέκτασης *Arachnoid*.

	Spam (Prediction)	Ham (Prediction)	Accuracy
Spam (Actual)	163	37	81,5%
Ham (Actual)	26	74	74%
Overall Accuracy	--	--	79%

[Εικόνα 6.5] Πίνακας Συσχετίσεων του φίλτρου του *Mozilla Thunderbird*

	Spam (Prediction)	Ham (Prediction)	Accuracy
Spam (Actual)	159	41	79,5%
Ham (Actual)	28	72	72%
Overall Accuracy	--	--	77%

[Εικόνα 6.6] Πίνακας Συσχετίσεων της επέκτασης *Arachnoid*

	Spam (Prediction)	Ham (Prediction)	Accuracy
Spam (Actual)	167	33	83,5%
Ham (Actual)	31	69	69%
Overall Accuracy	--	--	78,6%

[Εικόνα 6.7] Πίνακας Συσχετίσεων της επέκτασης *Arachnoid* με παραμετροποίηση

	Spam (Prediction)	Ham (Prediction)	Accuracy
Spam (Actual)	169	31	84,5
Ham (Actual)	22	78	78%
Overall Accuracy	--	--	82,3%

[Εικόνα 6.8] Πίνακας Συσχετίσεων του Συνδυασμού των Φίλτρων

6.5.3 Συμπεράσματα Πειραμάτων

Στο πείραμα του πίνακα της Εικόνας 6.5 τα μηνύματα σαρώνονται κατά την άφιξη τους από το προεπιλεγμένο φίλτρο του Thunderbird. Από τα αποτελέσματα προκύπτει:

- *Sensitivity = TP / (TP+FN)* = 163 / (163 + 37) = 0.815 = 81,5%
- *Specificity = TN / (TN+FP)* = 74 / (74 + 26) = 0.74 = 74%

Στο πείραμα του πίνακα της Εικόνας 6.6 τα μηνύματα σαρώνονται κατά την άφιξη τους από την επέκταση *Arachnoid* ενώ το φίλτρο της *Mozilla* έχει απενεργοποιηθεί χειροκίνητα από τις ρυθμίσεις. Στις επιλογές της επέκτασης στην καρτέλα «*AntiSpam*» χρησιμοποιήθηκε «*Spam Score Threshold*» ίσο με 25, «*High Spam Word Value*» ίσο με 7 «*Low Spam Word Value*» ίσο με 1. Από τα αποτελέσματα προκύπτει:

- *Sensitivity = TP / (TP+FN)* = 159 / (159 + 41) = 0.815 = 79,5%
- *Specificity = TN / (TN+FP)* = 72 / (72 + 28) = 0.72 = 72%

Στο πείραμα του πίνακα της Εικόνας 6.7 στις επιλογές της επέκτασης στην καρτέλα «*AntiSpam*» χρησιμοποιήθηκε «*Spam Score Threshold*» ίσο με 100, «*High Spam Word Value*» ίσο με 91 «*Low Spam Word Value*» ίσο με 6. Από τα αποτελέσματα προκύπτει:

- *Sensitivity = TP / (TP+FN)* = 167 / (167 + 33) = 0.815 = 83,5%
- *Specificity = TN / (TN+FP)* = 69 / (69 + 31) = 0.69 = 69%

Παρατηρούμε ότι στο δεύτερο πείραμα της επέκτασης, με τις παραμετροποιημένες και πιο επιθετικές ρυθμίσεις της επέκτασης, ενώ βελτιώθηκε η τιμή του *TP* το φίλτρο λανθασμένα κατηγοριοποιεί περισσότερα επιθυμητά μηνύματα ως *Spam.* Κάτι τέτοιο

αυξάνει τον κίνδυνο να σβηστούν τα μηνύματα πριν διαβαστούν από τον χρήστη, όταν οι επιλογές αυτόματης διαγραφής είναι ενεργοποιημένες.

Στο πείραμα του πίνακα της Εικόνας 6.8 τα μηνύματα σαρώνονται κατά την άφιξη τους από τον συνδυασμό των δύο φίλτρων. Η βάση δεδομένων λέξεων κλειδιών της επέκτασης Arachnoid βοηθά το προεπιλεγμένο φίλτρο να περάσει πιο γρήγορα την φάση της εκπαίδευσης και να αρχίσει να αποδίδει. Από τα αποτελέσματα προκύπτει:

- *Sensitivity = TP / (TP+FN)* = 169 / (169 + 31) = 0.815 = 84,5%
- *Specificity = TN / (TN+FP)* = 78 / (78 + 22) = 0.74 = 78%

Συμπερασματικά, με βάση τις τιμές των μετρικών, αντιλαμβάνεται κάποιος ότι η θετική πρόβλεψη είναι πιο επιθετική, ειδικά με βάση την υψηλή τιμή της σαφήνειας έναντι της ευαισθησίας. Επιπλέον, παρατηρούμε ότι ενώ η απόδοση των φίλτρων ξεχωριστά δεν παρουσιάζει μεγάλες διαφορές, σε συνδυασμό πετυχαίνουν αρκετά καλυτέρα αποτελέσματα και συνολική ευστοχία.

7 *Μελλοντικές Επεκτάσεις*

Στην παρούσα έκδοσή του το *Arachnoid Spam Filter* είναι σε θέση να ξεχωρίσει το μεγαλύτερο ποσοστό των *Ham* μηνυμάτων ηλεκτρονικού ταχυδρομείου από τα *Spam* και να προστατέψει τον χρήστη από απάτες και διαφημίσεις. Επιπρόσθετα, ο χρήστης, εφόσον το επιλέξει, έχει την δυνατότητα να επεκτείνει την βιβλιοθήκη λέξεων και φράσεων κλειδιών της εφαρμογής καθώς και να αναγνωρίσει ως προέλευση *Spam* πολλαπλούς αποστολείς και *domains*.

Κάποιες πιθανές βελτιώσεις του *Arachnoid Spam Filter* είναι:

• Αυτόματη εισαγωγή λέξεων και φράσεων κλειδιών στην βιβλιοθήκη.

• Εξαγωγή στατικών στοιχείων.

• Μαύρη Λίστα διευθύνσεων IP.

Βιβλιογραφία

[1] http://el.wikipedia.org/wiki/%CE%A3%CF%80%CE%B1%CE%BC
[2] http://en.wikipedia.org/wiki/Spamming#Email
[3] http://pdf.aminer.org/000/085/119/spamguru_an_enterprise_anti_spam_filtering_system.pdf
[4] http://tech.in.gr/consult/article/?aid=1231103537
[5] http://www.sch.gr/2010-04-07-09-22-34/-spam?showall=1
[6] http://www.techopedia.com/definition/24392/xml-user-interface-language-xul
[7] https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XUL/Introduction_to_XUL
[8] http://invenio.lib.auth.gr/record/114693/files/ptuxiaki2.ppt
[9] https://developer.mozilla.org/en-US/docs/Tools/Remote_Debugging/Thunderbird
[10] https://developer.mozilla.org/en-US/docs/Bundles
[11] http://www.sch.gr/sch-portlets/static/manual/aboutSpam/index.php?_list=against
[12] http://www.arnos.gr/system/files/2007-2008_2i_ergasia_pli_35-eap_apantiseis_arnos.pdf
[13] http://www.postcastserver.com/help/Internet_Black_and_White_Lists.aspx
[14] http://en.wikipedia.org/wiki/Reputation_system
[15] http://www1.coe.neu.edu/~rwhelan/Spam/Uluski/algorithms_avspam.pdf
[16] http://aimotion.blogspot.gr/2010/08/tools-for-machine-learning-performance.html

[Αντωνόπουλος, 2009]Αντωνόπουλος, Α. (2009). Παροχή ασφαλών υπηρεσιών με φερέγγυες υποδομές (Doctoral dissertation).

[Καρεκλάς, 2005]Καρεκλάς, Ν., & Δενδρινός, Μ. (2005). Θέματα ασφάλειας διακίνησης πληροφοριών στη βιβλιοθηκονομική κοινότητα. Η διαχείρηση της γνώσης για την υποστήριξη της έρευνας στη ακαδημαϊκή κοινότητα: οι βιβλιοθήκες ως ποιοτικά φίλτρα πληροφοριών.

[Γεώργιζα, 2009] Γεώργιζα, Χ. (2009). Anti- Spamming σε publish/ subscribe συστήματα - Προστασία από κακόβουλους Publishers. Αθήνα

[Βαβίτσας, 2009]Βαβίτσας, Γ. (2009). Μοντέλα διάδοσης απειλών σε δίκτυα υπολογιστών: ένα προτεινόμενο μοντέλο (Doctoral dissertation).

[Graham, 2003]Paul Graham. Better Bayesian filtering. In Proceedings of the 2003 spam Conference, Jan 2003.
http://www.paulgraham.com/better.html.

[Gomez, 2002]José María Gómez-Hidalgo. Evaluating cost-sensitive unsolicited bulk email categorization. In Proceedings of SAC-02, 17th ACM Symposium on Applied Computing, pages 615--620, Madrid, ES, 2002.

José María Gómez-Hidalgo, Manuel Maña-López, and Enrique Puertas-Sanz. Evaluating cost-sensitive unsolicited bulk email categorization. In Proceedings of

JADT-02, 6th International Conference on the Statistical Analysis of Textual Data, Madrid, ES, 2002.

[Cormack, 2006]Cormack G.V. and Bratko A., Batch and On-line spam Filter Evaluation, CEAS 2006 - Third Conference on Email and Anti-spam, Mountain View, July 2006.

[Sebastiani. 2002]Machine Learning in Automated Text Categorization, 2002

[Provost, 1999]Jefferson Provost. Naive-bayes vs. rule-learning in classification of email. Technical report, Dept. of Computer Sciences at the U. of Texas at Austin, 1999.

[Provost, 1997]Foster Provost and Tom Fawcett. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, 1997.

[Cormack, 2005]Cormack G.V. and Lynam T.R. TREC 2005 spam Track Overview in Proc. TREC 2005 - the Fourteenth Text REtrieval Conference, Gaithersburg, 2005.

Cormack G.V. and Lynam T.R. spam Corpus Creation for TREC, Proc. CEAS 2005 - The Second Conference on Email and Anti-spam, Palo Alto, July 2005.

[Zelkowitz, 2011]Advances in Computers: Software Development, Marvin Zelkowitz. 2011

[Androutsopoulos00α]I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, G. Paliouras, and C.D. Spyropoulos. 2000. An Evaluation of Naive Bayesian Anti-spam Filtering. In Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML), 9-17,Barcelona, Spain.

[Androutsopoulos00β]I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C.D. Spyropoulos and P. Stamatopoulos. 2000. Learning to Filter spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach". In Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), 1-13, Lyon, France.

[Androutsopoulos00γ]I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, and C.D. Spyropoulos. An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-spam Filtering with Encrypted Personal E-mail Messages. In Proceedings of the 23rd Annual nternational ACM SIGIR Conference on Research and Development in Information Retrieval, 160-167, Athens, Greece.

Παράρτημα

Παράρτημα 1

```
get checkSpam() {
        var spamThreshold;  // Spam score threshold
        var spamNoToRuleValue;  // AntiSpam "no-to" rule value
        var spamMultipleToRuleValues;  // AntiSpam "multiple-to" rule
values: [0: add, 1: every]
        var spamWords = new Array(2);  // Spam words: high and low

        if ( this.isEmailAddressInAddressBooks ) {
            //HOAX:
            spamThreshold =
arachnoidfilter.Preferences.getHoaxMinValue();

            spamNoToRuleValue =
arachnoidfilter.Preferences.getAntiHoaxNoToRuleValue();
            spamMultipleToRuleValues =
arachnoidfilter.Preferences.getAntiHoaxMultipleToRuleValues();

            spamWords[0] =
arachnoidfilter.Preferences.getHoaxHighWords();
            spamWords[1] =
arachnoidfilter.Preferences.getHoaxLowWords();
        }
        else {  // !this.isEmailAddressInAddressBooks()
            //SPAM:
            spamThreshold =
arachnoidfilter.Preferences.getSpamMinValue();

            spamNoToRuleValue =
arachnoidfilter.Preferences.getAntiSpamNoToRuleValue();
            spamMultipleToRuleValues =
arachnoidfilter.Preferences.getAntiSpamMultipleToRuleValues();

            spamWords[0] =
arachnoidfilter.Preferences.getSpamHighWords();
            spamWords[1] =
arachnoidfilter.Preferences.getSpamLowWords();
        }

        if ( this.recipients == "" || this.recipients == "undisclosed-
recipients:;" ) {
            this.spamScore += spamNoToRuleValue;
        }

        var numberOfRecipients = this.cc.match(/@/g);
        if ( numberOfRecipients != null ) {
            this.spamScore += Math.floor(numberOfRecipients.length /
spamMultipleToRuleValues[1]) * spamMultipleToRuleValues[0];
        }

        let emailWords = (this.subject + " " +
this.body).toLowerCase();
```

```
        for(let s=0; s < 2; s++) {  // Spam words: high and low

            for (let i=1; i < spamWords[s].length; i++) {
                let keyPos = emailWords.indexOf( spamWords[s][i] );

                if ( keyPos >= 0 ) {
                    this.spamScore += spamWords[s][0];

                    if ( this.spamScore >= spamThreshold ) {  // Einai
spam
                        return true;
                    }
                }

                lastKeyPos = emailWords.lastIndexOf( spamWords[s][i]
);

                if ( lastKeyPos != keyPos ) {
                    this.spamScore += spamWords[s][0];

                    if ( this.spamScore >= spamThreshold ) {  // Einai
Spam
                        return true;
                    }
                }
            }
        }

        return false;
    }
```

Παράρτημα 2

```
markEmailAs: function(email, classification, junkscore,
junkscoreorigin) {
        let gJunkService =
Components.classes["@mozilla.org/messenger/filter-
plugin;1?name=bayesianfilter"].getService(Components.interfaces.nsIJun
kMailPlugin);
        let oldJunkscore =
email.header.getStringProperty("junkscore");
        let oldJunkscoreorigin =
email.header.getStringProperty("junkscoreorigin");

        let oldClassification =
Components.interfaces.nsIJunkMailPlugin.UNCLASSIFIED;
        if ( oldJunkscoreorigin == "user" ) {
            switch( oldJunkscore ) {
                case "0":
                        oldClassification =
Components.interfaces.nsIJunkMailPlugin.GOOD;
                    break;

                case "100":
```

```
                        oldClassification =
Components.interfaces.nsIJunkMailPlugin.JUNK;
                    break;
                }
            }

            let db = email.folder.msgDatabase;
            db.setStringPropertyByHdr(email.header, "junkscore",
junkscore);
            db.setStringPropertyByHdr(email.header, "junkscoreorigin",
junkscoreorigin);

            if ( classification != oldClassification ) {
                gJunkService.setMessageClassification(email.uri,
oldClassification, classification, null, null);
            }
        }
```

Παράρτημα 3

```
if ( arachnoidfilter.Scan.scanEmail(aMsgHdr, false) ) {
                    if (
(arachnoidfilter.Preferences.isDeleteSpamActive() &&
!isEmailAddressInAddressBooks) ||
(arachnoidfilter.Preferences.isDeleteHoaxActive() &&
isEmailAddressInAddressBooks) ) {

                        let trashFolder =
aMsgHdr.folder.rootFolder.getFolderWithFlags(Components.interfaces.nsM
sgFolderFlags.Trash);
                        let unwantedEmail =
Components.classes["@mozilla.org/array;1"].createInstance(Components.i
nterfaces.nsIMutableArray);
                        unwantedEmail.appendElement(aMsgHdr, false
/*weak*/);

Components.classes["@mozilla.org/messenger/messagecopyservice;1"].getS
ervice(Components.interfaces.nsIMsgCopyService).CopyMessages(aMsgHdr.f
older, unwantedEmail, trashFolder, true /*isMove*/, null, msgWindow,
true /*allowUndo*/);
}
```

Παράρτημα 4

```
isInBlacklist: function(author) {
        for (i=0; i <
arachnoidfilter.Preferences.getBlacklist().length; i++) {
            if ( author ==
arachnoidfilter.Preferences.getBlacklist()[i] ) {
                return true;
```

```
            }
        }

        return false;
    },
```

Παράρτημα 5

```
if ( arachnoidfilter.Preferences.isBlacklistActive() &&
!(arachnoidfilter.Preferences.getBlacklist().length == 1 &&
arachnoidfilter.Preferences.getBlacklist()[0] == "") ) {
                if ( arachnoidfilter.Scan.isInBlacklist( finaladdr )
|| arachnoidfilter.Scan.isInBlacklist(finaldom)) {
                    let gJunkService =
Components.classes["@mozilla.org/messenger/filter-
plugin;1?name=bayesianfilter"].getService(Components.interfaces.nsIJun
kMailPlugin);
                    let oldJunkscore =
aMsgHdr.getStringProperty("junkscore");
                    let oldJunkscoreorigin =
aMsgHdr.getStringProperty("junkscoreorigin");
                    aNewClassification =
Components.interfaces.nsIJunkMailPlugin.JUNK;


                    let db = aMsgHdr.folder.msgDatabase;
                    db.setStringPropertyByHdr(aMsgHdr, "junkscore",
"100");
                    db.setStringPropertyByHdr(aMsgHdr,
"junkscoreorigin", "user");

                    if ( gJunkService.aNewClassification !=
gJunkService.aOldUserClassification ) {
                        gJunkService.setMessageClassification(aMsgURI,
gJunkService.aOldUserClassification, gJunkService.aNewClassification,
null, null);
                    }
                }
```

Παράρτημα 6

```
if( arachnoidfilter.Preferences.isDeleteBlacklistActive() ) {
alert("Blacklist Deletion is Active");

                        let trashFolder =
aMsgHdr.folder.rootFolder.getFolderWithFlags(Components.interfaces.nsM
sgFolderFlags.Trash);
                        let unwantedEmail =
Components.classes["@mozilla.org/array;1"].createInstance(Components.i
nterfaces.nsIMutableArray);
                        unwantedEmail.appendElement(aMsgHdr, false);
```

```
Components.classes["@mozilla.org/messenger/messagecopyservice;1"].getS
ervice(Components.interfaces.nsIMsgCopyService).CopyMessages(aMsgHdr.f
older, unwantedEmail, trashFolder, true /*isMove*/, null, msgWindow,
true /*allowUndo*/);
}
```

Παράρτημα 7

```
addToBlacklist: function(dom) {
            if ( typeof dom == 'undefined' ) {
                dom = false;
            }

            let messenger =
Components.classes["@mozilla.org/messenger;1"].createInstance(Componen
ts.interfaces.nsIMessenger);

            var selectedEmailURIs =
gFolderDisplay.selectedMessageUris;

            for each (let emailURI in selectedEmailURIs) {
                let emailHeader =
messenger.messageServiceFromURI(emailURI).messageURIToMsgHdr(emailURI)
;
            var email = new arachnoidfilter.Email(emailURI,
emailHeader);

                if ( dom ) {
                    if(
!arachnoidfilter.Scan.isInBlacklist(email.authorDomain) ) {
arachnoidfilter.Preferences.addToBlacklist( email.authorDomain );
                    }
                }
                else {
                    if (
!arachnoidfilter.Scan.isInBlacklist(email.author) ) {
arachnoidfilter.Preferences.addToBlacklist( email.author );
                    }
                }
            }
        }
```

Παράρτημα 8

```
isInWhitelist: function(author) {
        for (i=0; i <
arachnoidfilter.Preferences.getWhitelist().length; i++) {
            if ( author ==
arachnoidfilter.Preferences.getWhitelist()[i] ) {
                return true;
```

```
            }
        }
        return false;       }
```

Παράρτημα 9

```
if ( arachnoidfilter.Preferences.isWhitelistActive() &&
!(arachnoidfilter.Preferences.getWhitelist().length == 1 &&
arachnoidfilter.Preferences.getWhitelist()[0] == "") ) {

      if ( arachnoidfilter.Scan.isInWhitelist(finaladdr) ||
arachnoidfilter.Scan.isInWhitelist(finaldom) ) {
            return;
      }
}
```

Παράρτημα 10

```
addToWhitelist: function(dom) {
            if ( typeof dom == 'undefined' ) {
                dom = false;
            }

            let messenger =
Components.classes["@mozilla.org/messenger;1"].createInstance(Componen
ts.interfaces.nsIMessenger);

            var selectedEmailURIs =
gFolderDisplay.selectedMessageUris;

            for each (let emailURI in selectedEmailURIs) {
                let emailHeader =
messenger.messageServiceFromURI(emailURI).messageURIToMsgHdr(emailURI)
;
                var email = new arachnoidfilter.Email(emailURI,
emailHeader);

                if ( dom ) {
                    if(
!arachnoidfilter.Scan.isInWhitelist(email.authorDomain) ) {
                        arachnoidfilter.Preferences.addToWhitelist(
email.authorDomain );
                    }
                }
                else {
      if ( !arachnoidfilter.Scan.isInWhitelist(email.author) ) {
arachnoidfilter.Preferences.addToWhitelist( email.author );
                    }
                }
            }
}
```

Παράρτημα 11

```
scan: function() {
      var singleEmail = false;

      arachnoidfilter.Scan.emailCounter = 0;
      arachnoidfilter.Scan.spamCounter = 0;
      arachnoidfilter.Scan.folderCounter = 1;
      arachnoidfilter.Scan.progressCounter = 0;

      arachnoidfilter.ScanWindow =
window.open('chrome://arachnoidfilter/content/scan.xul','','chrome=yes
,resizable=yes,centerscreen');
arachnoidfilter.ScanWindow.onunload = this.closeScanWindow;

      var selectedFolders = gFolderTreeView.getSelectedFolders();
      var selectedEmailURIs = gFolderDisplay.selectedMessageUris;

      if ( selectedEmailURIs == null ) {
            var spamList =
arachnoidfilter.Scan.scanFolders(selectedFolders, true);
arachnoidfilter.Scan.folderCounter = selectedFolders.length;
            spamList = null;
      }
      else {
            var isSpamFound =
arachnoidfilter.Scan.scanEmails(selectedEmailURIs, true);

      if ( selectedEmailURIs.length == 1 ) {
            singleEmail = true;
      }
      }
      spamCounterLabel =
arachnoidfilter.Overlay.statusbarOverlay(arachnoidfilter.Scan.emailCou
nter, arachnoidfilter.Scan.spamCounter);
document.getElementById("arachnoidfilter-
SpamCounterStat").setAttribute('label', spamCounterLabel);

      this.scanResults(singleEmail);

      arachnoidfilter.Overlay.addActivityManagerEvent("indexMail",
"Arachnoid Spam Filter scan completed", "Spam: " +
arachnoidfilter.Scan.spamCounter + "/" +
arachnoidfilter.Scan.emailCounter);
},
```

Παράρτημα 12

```
arachnoidfilter.Overlay.addActivityManagerEvent("indexMail",
"Arachnoid Spam Filter scan completed", "Spam: " +
arachnoidfilter.Scan.spamCounter + "/" +
arachnoidfilter.Scan.emailCounter);
```

Παράρτημα 13

```
spamCounterLabel =
arachnoidfilter.Overlay.statusbarOverlay(arachnoidfilter.Scan.emailCou
nter, arachnoidfilter.Scan.spamCounter);

document.getElementById("arachnoidfilter-
SpamCounterStat").setAttribute('label', spamCounterLabel);
```