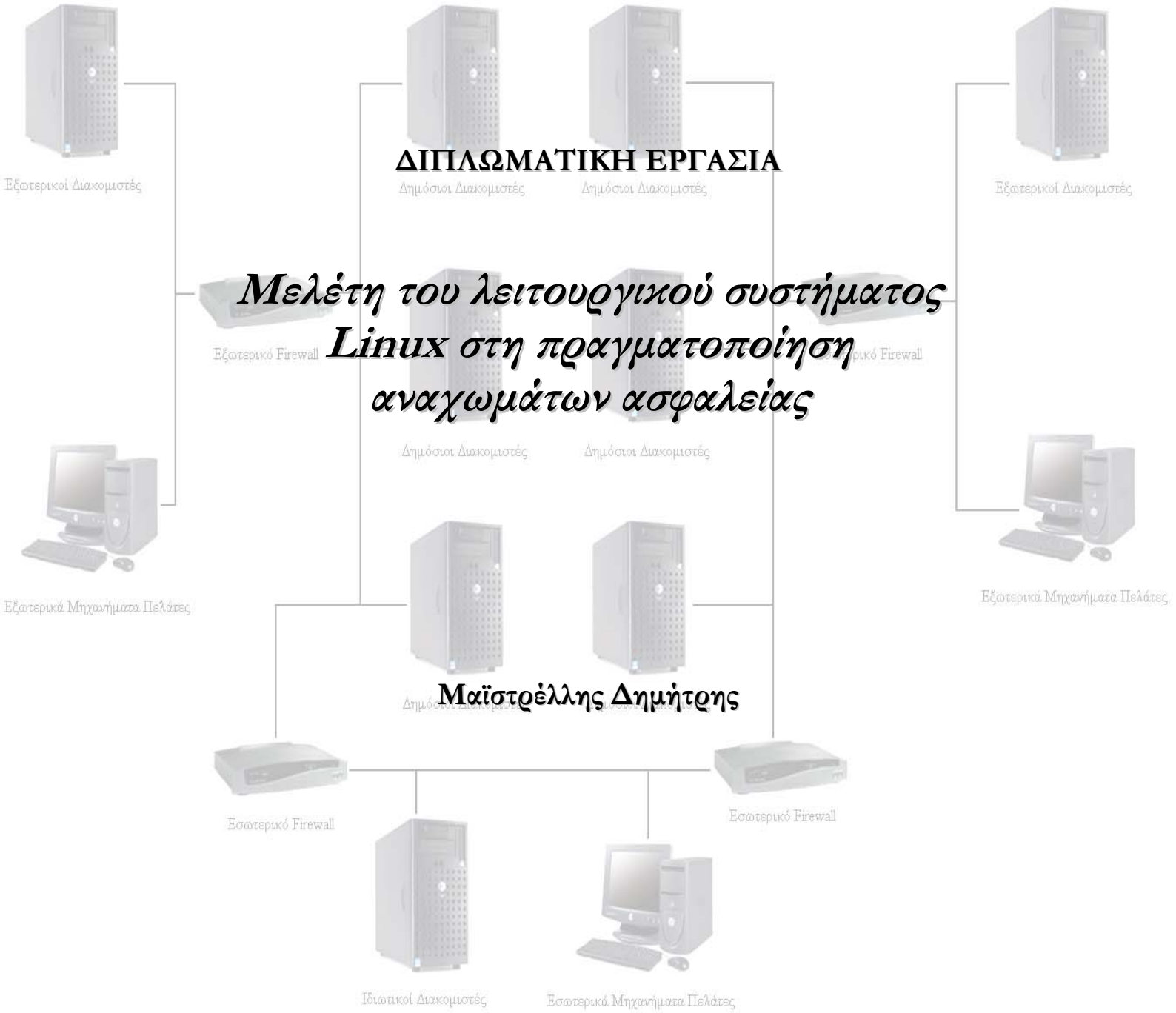
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΔΙΔΑΚΤΙΚΗΣ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μελέτη του λειτουργικού συστήματος Linux στη πραγματοποίηση αναχωμάτων ασφαλείας

Μαϊστρέλλης Δημήτρης

Επιβλέπων καθηγητής: Κάτσικας Σ.

ΠΕΙΡΑΙΑΣ 2009

Περιεχόμενα

Εισαγωγή

Η διπλωματική αυτή εργασία πραγματεύεται θέματα δικτυακής ασφάλειας υπολογιστών και το πώς κάτι τέτοιο μπορεί να επιτευχθεί με τη χρήση ενός κατεξοχήν δικτυακού λειτουργικού συστήματος, του Linux. Μιλώντας τόσο σε θεωρητικό επίπεδο, αναλύοντας τις δυνατότητες και τις εφαρμογές του, αλλά και σε πρακτικό με οδηγίες και συμβουλές για την κατασκευή ενός Firewall και Proxy Server για την βελτίωση της ασφάλειας της ασφάλειας δικτύων.

Ένας firewall server αποτελείται από δυο ή περισσότερους προσαρμογείς δικτύου (κάρτες Ethernet) οι οποίοι συνδέουν διαφορετικά δίκτυα μεταξύ τους. Μπορεί να φιλτράρει IP πακέτα με βάση τις τιμές του περιεχομένου της επικεφαλίδας του πακέτου, όπως η διεύθυνση αφετηρίας και προορισμού. Επίσης μπορεί να παραμετροποιηθεί ώστε να επιτρέπει πρόσβαση μόνο σε συγκεκριμένα πακέτα, να επιτρέπει την πραγματοποίηση συνδέσεων μόνο από συγκεκριμένους υπολογιστές και να μπλοκάρει την πρόσβαση μη εξουσιοδοτημένων υπολογιστών. Ένας Proxy server είναι μια εφαρμογή η οποία τρέχει στο firewall και αναμεταδίδει την κίνηση μεταξύ αυτού και του προορισμού. Έτσι αντί να επιτρέψουμε σε δύο υπολογιστές να επικοινωνήσουν μεταξύ τους κατευθείαν, τους αναγκάζουμε να επικοινωνήσουν μέσω ενός άλλου υπολογιστή ο οποίος λέγεται εξυπηρετητής (Proxy).Πιο εξειδικευμένα εξυπηρετητές μπορούν να χειριστούν πιο πολύπλοκες διαδικασίες οι οποίες είναι πέρα από τις δυνατότητες των firewalls. Μπορούν να καταλάβουν την εφαρμογή και το περιεχόμενό της ώστε να εκτελέσουν διεργασίες όπως το φιλτράρισμα mail ανάλογα με την προέλευσή του ή μπορεί να έχουν ακόμα υψηλότερου επιπέδου φιλτράρισμα ώστε να ψάχνουν για πορνογραφικό υλικό ή ακόμα και για συντακτικά και ορθογραφικά λάθη. Το κλειδί για αυτό είναι ο server να μπορεί να καταλάβει το περιεχόμενο που μεταφέρεται ακόμα και αν χρειαστεί να το αναγνωρίσει αναλύοντας το πρωτόκολλο.

Μερικά οφέλη από τα οποία μπορεί να προσφέρει η πραγματοποίηση ενός τέτοιου συστήματος, πέρα από την προστασία των host του εσωτερικού δικτύου, είναι η χρησιμοποίηση λιγότερου bandwidth σύνδεσης internet, η μείωση του χρόνου φόρτωσης των σελίδων, η συλλογή στατιστικών για την κίνηση του δικτύου, η παρεμπόδιση χρηστών από το να επισκέπτονται συγκεκριμένα site (π.χ. πορνογραφικού υλικού κτλ), η βεβαίωση ότι μόνο εξουσιοδοτημένοι χρήστες μπορούν να σερφάρουν στο internet, και η βελτίωση της ασφάλειας των χρηστών φιλτράροντας ευαίσθητες πληροφορίες.

Κεφάλαιο 1

Περί Firewall και Proxy διακομιστών

1.1 Γιατί χρειαζόμαστε τα firewall;

Ο ρόλος των firewall είναι η προστασία των δικτύων. Πιο συγκεκριμένα επειδή ο όρος δίκτυο δεν είναι απόλυτα σαφής μπορεί να μην είναι και κατανοητό και για ποιο λόγο το δίκτυο χρειάζεται προστασία.

Από πλευράς επιχειρήσεων τα πολύτιμα στοιχεία ενός δικτύου είναι:

- Η εμπιστευτικότητα των δεδομένων που μεταφέρονται
- Η αξιοπιστία μεταφοράς δεδομένων
- Η διαθεσιμότητα του δικτύου

Εμπιστευτικότητα δεδομένων

Μερικά δεδομένα είναι πολύτιμα γιατί δεν είναι ευρέως γνωστά. Για παράδειγμα θα ήταν πολύτιμο αν γνωρίζαμε τις αυριανές τιμές του χρηματιστηρίου. Αν όμως είχαν όλοι πρόσβαση σε αυτές τις πληροφορίες τότε πιθανόν να ήταν άνευ αξίας. Πολλές εταιρίες κατέχουν εμπιστευτικά δεδομένα αποθηκευμένα σε αρχεία υπολογιστών. Αυτοί οι υπολογιστές πρέπει να προστατευτούν από επιθέσεις, μη εξουσιοδοτημένη χρήση και από άλλα γεγονότα τα οποία μπορούν να οδηγήσουν σε διαρροές δεδομένων.

Αξιοπιστία δεδομένων

Η αξιοπιστία των δεδομένων είναι ο βαθμός στον οποίο κάποιος μπορεί να είναι σίγουρος ότι τα δεδομένα θα είναι πλήρη και ακριβή. Η αξιοπιστία των δεδομένων είναι σημαντική γιατί η αξία τους μειώνεται η χάνεται αν τα περιεχόμενά τους μεταβληθούν ή είναι ανακριβή.

Διαθεσιμότητα του δικτύου

Μερικά δίκτυα υποστηρίζουν επιχειρήσεις ή παρέχουν υπηρεσίες, έτσι μία δυσλειτουργία τους μπορεί να είναι πολλές φορές καταστροφική. Για παράδειγμα το «Ebay» έχει γίνει πολλές φορές θύμα επιθέσεων λόγω του ότι παρέχει On-Line συναλλαγές. Η αναστολή της λειτουργίας του για κάποιο διάστημα μπορεί να αποφέρει σημαντικές απώλειες κερδών.

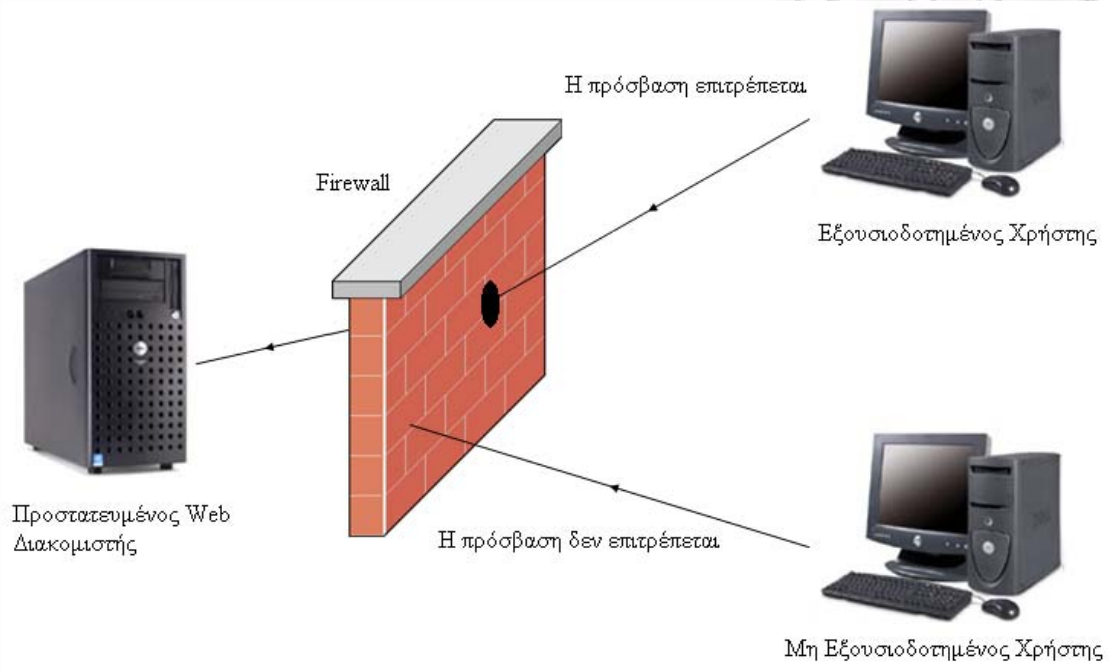
1.2 Πως απειλούνται τα δίκτυα

Πριν την εξάπλωση του internet οι οργανισμοί και οι επιχειρήσεις διατηρούσαν ιδιωτικά δίκτυα υπολογιστών τα οποία δεν παρείχαν απομακρυσμένη πρόσβαση. Έτσι η επίθεση σε έναν υπολογιστή ή γενικότερα σε ένα δίκτυο απαιτούσε φυσική επαφή με το στόχο. Ο επιτιθέμενος έπρεπε να είχε εξουσιοδοτημένη πρόσβαση ή με κάποιον τρόπο να εισβάλει παράνομα στην εταιρία με σκοπό την πρόσβαση σε κάποιο τερματικό. Έτσι οι περισσότερες επιθέσεις γίνονταν από υπαλλήλους. Με τη χρήση των modem ανοίχτηκε μία νέα οδό επιθέσεων και αρκούσε μία τηλεφωνική κλήση από απομακρυσμένη περιοχή αντί για την παραβίαση φυσικών μέτρων ασφαλείας. Στις μέρες μας όλες οι επιχειρήσεις έχουν δημόσια δίκτυα τα οποία περιέχουν και δεδομένα τα οποία χρειάζονται προστασία. Ένας επιτιθέμενος ο οποίος μπορεί να αποκτήσει παραπάνω προνόμια πρόσβασης μπορεί να προκαλέσει από απώλεια δεδομένων μέχρι κατάρρευση του δικτύου.

1.3 Τι κάνει το firewall

Το firewall ενός αυτοκινήτου είναι σχεδιασμένο για να εμποδίζει την εξάπλωση της φωτιάς από το χώρο της μηχανής στην καμπίνα των επιβατών. Σκοπός του είναι ο περιορισμός. Ένα firewall δικτύου είναι επίσης μία συσκευή περιορισμού. Ένα firewall λειτουργεί με το να διαιρεί ένα δίκτυο σε πολλαπλές ζώνες και να περιορίζει τη λειτουργία σε αυτές. Αυτό γίνεται με το να εμποδίζει τη μη εξουσιοδοτημένη κίνηση να εισέρχεται ή να εξέρχεται. Αν ρυθμιστεί σωστά μπορεί να εμποδίσει μια επίθεση να φτάσει στον προορισμό της.

Το παρακάτω σχήμα δείχνει ένα απλό τυπικό firewall το οποίο επιτρέπει μόνο σε συγκεκριμένους χρήστες να προσπελάσουν τον Web Server. Ένας επιτιθέμενος για να αποκτήσει πρόσβαση στον Web Server πρέπει πρώτα να νικήσει το firewall.



Ένα firewall κάνει περισσότερα από το να μπλοκάρει απλά τη μη εξουσιοδοτημένη πρόσβαση. Το firewall έχει δύο πρωταρχικούς ρόλους: παρεμπόδιση και ανίχνευση. Η πολιτική ασφαλείας καθορίζει τις λειτουργίες που ένας χρήστης είναι εξουσιοδοτημένος να εκτελεί. Το firewall εμποδίσει τις επιθέσεις με το να φιλτράρει την κίνηση βασιζόμενο στην πολιτική ασφαλείας και επίσης ανιχνεύει επιθέσεις. Η ανίχνευση αυτή γίνεται με το να κρατάει αρχείο (log) για τις προσπάθειες και της πραγματοποιήσεις συνδέσεων με μηχανήματα του δικτύου και να ενημερώνει τους διαχειριστές για ύποπτες ενέργειες. Προφανώς το firewall δεν μπορεί να εμποδίσει μια επίθεση την οποία απέτυχε να εντοπίσει. Άρα η ανίχνευση προηγείται της παρεμπόδισης. Ένα firewall μπορεί να φιλτράρει την κίνηση με διάφορους τρόπους. Οι πιο κοινοί είναι μέσω των:

- IP διευθύνσεων
- Υπηρεσιών

Η κίνηση του δικτύου σηματοδοτείται από τις διευθύνσεις IP, οι οποίες δείχνουν την αφετηρία και τον προορισμό του host. Η κίνηση η οποία απαιτεί πρόσβαση σε μια υπηρεσία καθορίζεται από τον αριθμό του port το οποίο συνδυαζόμενο με την διεύθυνση IP καθορίζει την υπηρεσία. Επίσης ένα firewall πρέπει να μπορεί να προστατεύσει τον κεντρικό υπολογιστή του δικτύου από επιθέσεις οι οποίες προέρχονται και από τους υπολογιστές του τοπικού δικτύου. Κάτι τέτοιο μπορεί να μοιάζει ασήμαντο αλλά φανταστείτε αν ο επιτιθέμενος αποκτώντας πρόσβαση, χρησιμοποιήσει αυτόν σαν μεσάζοντα της επιθέσεώς του. Αυτό σημαίνει ότι ένα firewall πρέπει να ελέγχει την εισερχόμενη και την εξερχόμενη κίνηση.

1.4 Τύποι firewall

Σε γενικές γραμμές υπάρχουν τρεις τύποι firewall:

Προσωπικά firewall

Προστατεύουν τον υπολογιστή ενός μικρού δικτύου, δηλαδή ενός δικτύου με σύνδεση DSL ή καλωδιακών modem. Σε αυτήν την περίπτωση δεν χρησιμοποιούνται διακομιστές firewall αλλά κάποιο απλό λογισμικό του εμπορίου με εύκολη χρήση και συντήρηση γιατί η χρήση ενός διακομιστή firewall απαιτεί προχωρημένες γνώσεις θεμάτων ασφαλείας.

Τμηματικά firewall

Σε γενικές γραμμές προστατεύουν περισσότερους υπολογιστές από ότι ένα προσωπικό firewall. Οι υπολογιστές οι οποίοι προστατεύονται είναι πιθανό να παρέχουν περισσότερες και πιο εξελιγμένες υπηρεσίες και επιπλέον να μπορούν να χειριστούν μεγαλύτερο όγκο πληροφοριών. Τα τμηματικά firewall είναι συνδυασμοί πολλών firewall με σκοπό να παρέχουν αυξημένες υπηρεσίες ασφαλείας.

Επιχειρησιακά firewall

Αποτελούνται από πολλά τμηματικά firewall τα οποία φιλτράρουν και κρατάνε αρχείο της εισερχόμενης και εξερχόμενης κίνησης (log). Η διαφορά τους είναι ότι χρειάζονται περισσότερη γνώση και συντήρηση για τη σωστή λειτουργία τους. Επειδή διαχειρίζονται μεγάλο

όγκο πληροφοριών απαιτούν αυτοματοποιημένες διαδικασίες για την παρακολούθηση και την ανάλυση των αρχείων καταγραφής κίνησης.

1.5 Μειονεκτήματα των firewall

Τα firewall παρουσιάζουν διάφορα μειονεκτήματα. Στις περισσότερες περιπτώσεις τα οφέλη υπερέχουν των μειονεκτημάτων, παρόλα αυτά πρέπει να γνωρίζουμε τα μειονεκτήματα με σκοπό να τα ελαχιστοποιήσουμε ή να τα υπερπηδήσουμε. Τα μειονεκτήματα πηγάζουν από το γεγονός ότι το firewall μετατρέπεται σε ένα στενό πέρασμα επηρεάζοντας το δίκτυο με τους τρεις παρακάτω τρόπους:

- Αξιοπιστία
- Απόδοση
- Ευκαμψία

Η αποτυχία λειτουργίας ενός firewall μειώνει την αξιοπιστία του δικτύου. Αυτό αποφεύγεται με τη χρήση πολλαπλών firewall ρυθμιζόμενα έτσι ώστε η αποτυχία λειτουργίας του ενός να ενεργοποιήσει το άλλο διαπερνόντας το οποίο τελικά η κίνηση να φτάσει στον προορισμό της. Με παρόμοιο τρόπο μπορεί να μειωθεί και η απόδοση του δικτύου. Επειδή όλη η κίνηση πρέπει να περάσει μέσα από το firewall η απόδοση του δικτύου επηρεάζεται από την ικανότητα χειρισμού του όγκου πληροφοριών από το firewall και επιπλέον για να χειριστεί ένα νέο τύπο δεδομένων το firewall πρέπει να ρυθμιστεί ξανά.

1.6 Τεχνολογίες firewall

Οι τεχνολογίες οι οποίες χρησιμοποιούνται κατά την δόμηση των firewall περιλαμβάνουν την προώθηση πακέτων και το φιλτράρισμά τους, τους εξυπηρετητές εφαρμογών και τέλος πιο σύγχρονες τεχνολογίες όπως τα λεπτομερούς επιθεώρησης και τα υβριδικά firewall. Επιπλέον υπάρχουν κάποιες άλλες σημαντικές τεχνολογίες οι οποίες χρησιμοποιούνται

σε συνδυασμό με τα firewall, όπως η μετάφραση διευθύνσεων δικτύου (NAT) και η χρήση εικονικών ιδιωτικών δικτύων (VPNs).

Προώθηση Πακέτων

Ο όρος προώθηση (δρομολόγηση) αναφέρεται σε διαδικασίες μετακίνησης πακέτων από ένα δίκτυο σε ένα άλλο. Αυτό συχνά γίνεται από μονάδες που ονομάζονται δρομολογητές. Στον κόσμο του Linux είναι πολύ πιθανό να δούμε έναν υπολογιστή να εκτελεί χρέη δρομολογητή. Ένα firewall μπορεί να πραγματοποιηθεί με το συνδυασμό τεχνολογιών δρομολόγησης και άλλων λειτουργιών. Ένας δρομολογητής αποτελείται από δυο ή περισσότερους προσαρμογείς δικτύου οι οποίοι συνδέουν διαφορετικά δίκτυα μεταξύ τους. Κάθε δρομολογητής περιλαμβάνει ένα πακέτο δρομολόγησης με ένα σύνολο κανόνων το ποίο καθορίζει ποια πακέτα θα προωθηθούν και που. Το πότε και που θα προωθηθούν τα πακέτα καθορίζεται από:

- Τον προσαρμογέα δικτύου στον οποίο φτάνει το πακέτο
- Τη διεύθυνση αφετηρίας του πακέτου
- Τη διεύθυνση προορισμού του πακέτου

Φιλτράρισμα πακέτων

Θεωρήστε το firewall σαν μία συσκευή διέλευσης η οποία διαχειρίζεται την κυκλοφορία. Τα firewall υλικού αποτελούνται από δρομολογητές ή από υπολογιστές οι οποίοι φέρουν το κατάλληλο λογισμικό. Οι δρομολογητές λειτουργούν σε επίπεδο δικτύου και μπορούν να φιλτράρουν IP πακέτα βασιζόμενα στις τιμές του περιεχομένου της επικεφαλίδας του πακέτου όπως η διεύθυνση αφετηρίας και προορισμού. Οι δρομολογητές μπορούν να παραμετροποιηθούν ώστε να επιτρέπουν πρόσβαση μόνο σε συγκεκριμένα πακέτα, να επιτρέπουν την πραγματοποίηση συνδέσεων μόνο από συγκεκριμένους υπολογιστές και να μπλοκάρουν την πρόσβαση των μη εξουσιοδοτημένων μηχανημάτων. Αυτή η διαδικασία συχνά αναφέρεται σαν φιλτράρισμα πακέτων (Packet Filtering).

Εξυπηρετητές εφαρμογών

Ένα firewall στρώματος εφαρμογών είναι ένας proxy διακομιστής ο οποίος παρέχει ένα άλλο στρώμα ασφαλείας το οποίο δεν παρέχουν τα firewall φιλτραρίσματος πακέτων. Βασικά ένας εξυπηρετητής εφαρμογών είναι μια εφαρμογή η οποία τρέχει στο firewall και αναμεταδίδει

την κίνηση μεταξύ αυτού και του προορισμού. Το πλεονέκτημα εδώ είναι ότι η κίνηση ανάμεσα στις δύο πλευρές μπορεί να ελεγχθεί μέσω τρίτων εφαρμογών.

Λεπτομερούς επιθεώρησης

Αυτά είναι τα τρίτης γενιάς firewall τα οποία σχετίζονται με τη μέθοδος του φιλτραρίσματος πακέτων αλλά επεκτείνουν τις λειτουργίες του firewall με το να συνεχίζουν να επιθεωρούν τα πακέτα τη στιγμή που περνάνε μέσα από το firewall.

Υβριδικά

Είναι τα τέταρτης γενιάς firewall τα οποία είναι συνδυασμός όλων των προηγουμένων και δίνουν σε όλους τους χρήστες περισσότερο έλεγχο.

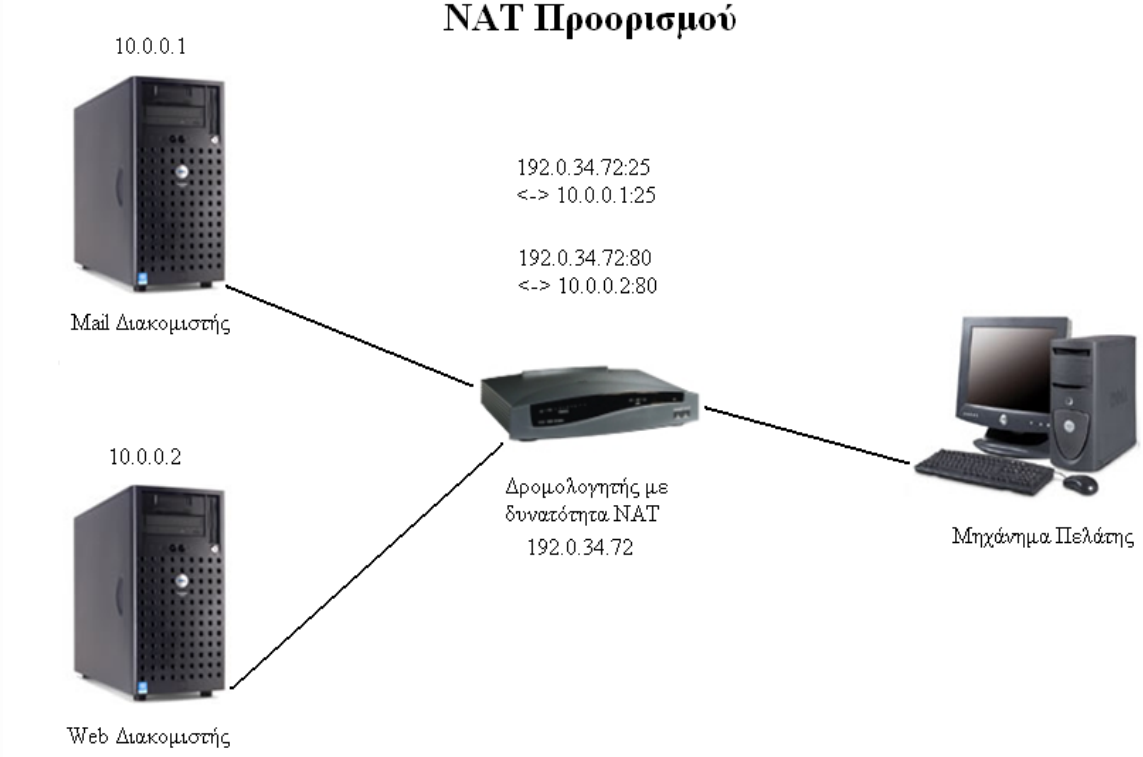
1.7 Άλλες σημαντικές τεχνολογίες

Μετάφραση διευθύνσεων δικτύου (NAT)

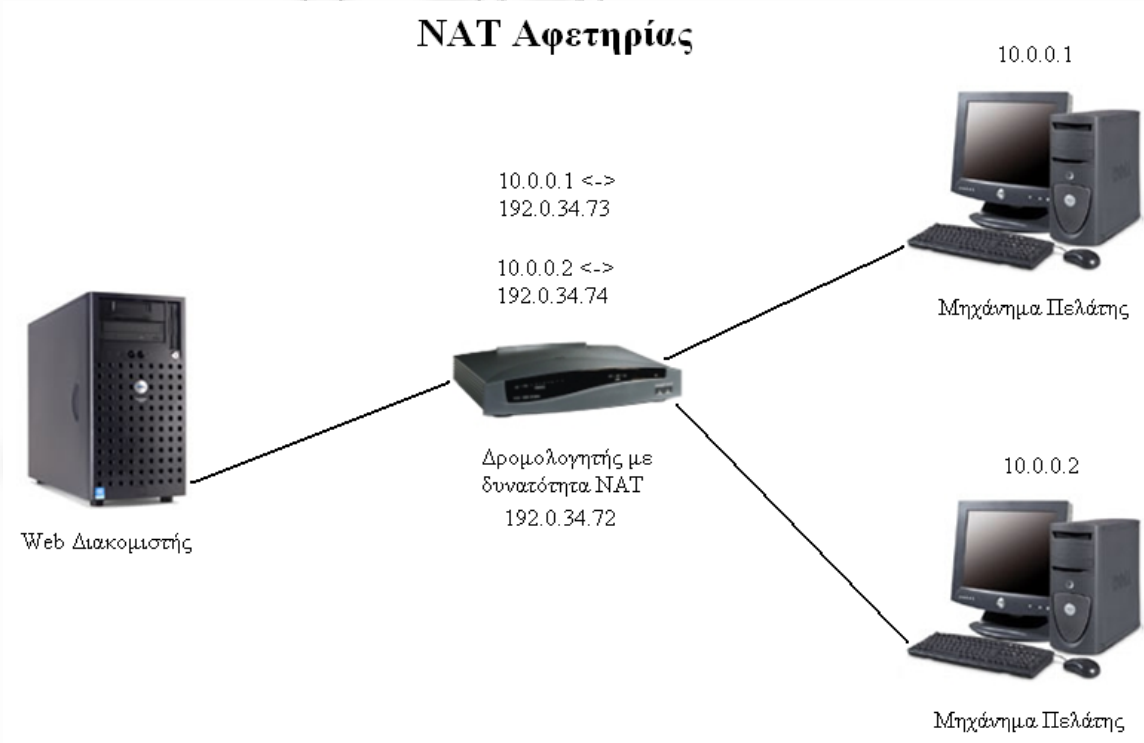
Η μετάφραση διευθύνσεων δικτύου σχεδιάστηκε για να επιτρέπει σε πολλαπλούς host να μοιράζονται μια IP διεύθυνση. Το NAT είναι μία απλή λειτουργία φιλτραρίσματος πακέτων η οποία πραγματοποιείται από δρομολογητές ή firewall κατά την οποία η διεύθυνση προορισμού ή αφετηρίας μεταβάλλεται.

Με χρήση DNAT (destination) μεταβάλλεται η διεύθυνση προορισμού ενώ με χρήση SNAT (source) μεταβάλλεται η διεύθυνση αφετηρίας. Με τη χρήση NAT επιτυγχάνεται η οικονομία διευθύνσεων IP. Για παράδειγμα ο πελάτης καλώντας την ίδια IP διεύθυνση μπορεί να συνδεθεί μέσο του δρομολογητή με πολλούς διακομιστές ανάλογα με την υπηρεσία για την οποία έχει κάνει αίτηση.

Στο παρακάτω σχήμα παρατηρούμε ένα παράδειγμα NAT προορισμού. Το μηχάνημα πελάτης (client) δρομολογείται μέσω του δρομολογητή είτε στον mail διακομιστή, είτε στον web διακομιστή, ανάλογα με την αίτηση.



Στο παρακάτω σχήμα παρατηρούμε ένα παράδειγμα NAT αφετηρίας.



Ιδιωτικά εικονικά δίκτυα (VPNs)

Συχνά είναι σημαντική η υποστήριξη απομακρυσμένων συνδέσεων έτσι ώστε ένας υπάλληλος να μπορεί να προσπελάσει το δίκτυο μια εταιρίας ακόμα και από το σπίτι. Χωρίς κάποια συγκεκριμένη λειτουργία που να το επιτρέπει το firewall θα εμπόδιζε την πρόσβαση. Ένας δημοφιλής τρόπος υποστήριξης αυτών των χρηστών είναι μέσω VPNs. Το VPN πραγματοποιεί τη σύνδεση, τη λεγόμενη tunnel, μεταξύ δυο δικτύων ή ανάμεσα σε ένα host και ένα δίκτυο. Το tunnel παρέχει εξειδικευμένη δρομολόγηση των δεδομένων. Τα VPN γενικά κρυπτογραφούν τα δεδομένα έτσι ώστε να ταξιδεύσουν ασφαλή μέσω ενδιάμεσων δικτύων.

Τύποι VPNs

Τα VPN χωρίζονται σε 3 κατηγορίες:

- Hardware VPN
- Firewall VPN
- Stand Alone εφαρμογές VPN

Το κάθε ένα από αυτά τα συστήματα έχει προτερήματα αλλά και μειονεκτήματα τα οποία θα αναλύσουμε λεπτομερέστερα παρακάτω.

Hardware VPN Συστήματα

Ο μεγαλύτερος όγκος των Hardware VPN συστημάτων είναι στην πραγματικότητα δρομολογητές κρυπτογράφησης. Όλη η κρυπτογράφηση γίνεται μέσω υλικού, το οποίο είναι πιο γρήγορο από αυτή του λογισμικού. Αυτοί οι δρομολογητές είναι πολύ εύκολοι στη χρήση, στην εγκατάσταση, και παρέχουν αυξημένη ασφάλεια. Μερικά hardware VPN περιλαμβάνουν και λογισμικό πελάτη για απομακρυσμένη εγκατάσταση και έχουν ενσωματωμένα μερικά από τα χαρακτηριστικά ελέγχου πρόσβασης τα οποία περιλαμβάνονται σε firewall και άλλους εξοπλισμούς ασφαλείας. Τα συστήματα αυτά παρουσιάζουν δύο κύρια προβλήματα. Πρώτον δεν παρουσιάζουν εύκολη παραμετροποίηση όπως αυτά του λογισμικού, και δεύτερον είναι εξαιρετικά ακριβά, πράγμα το οποίο τα κάνει απροσπέλαστα για ένα μέσο χρήστη και αναφέρονται κυρίως για μεγάλα εταιρικά δίκτυα.

Firewall VPN

Τα firewall VPN συστήματα αναφέρονται και σε εταιρική αλλά και σε προσωπική χρήση. Εκμεταλλεύονται τα προτερήματα ασφαλείας που προσφέρουν τα firewall περιλαμβάνοντας ελεγχόμενη πρόσβαση σε εσωτερικό δίκτυο, NAT και άλλα. Επιπλέον παρέχουν αυξημένο έλεγχο πρόσβασης καθώς και λειτουργίες εγγραφών (logging). Σε αυτήν την εγκατάσταση υπάρχουν πολλά μειονεκτήματα. Ένα από αυτά είναι ότι το λειτουργικό που τρέχει το VPN σύστημα πρέπει να είναι όσο πιο ασφαλές γίνεται. Αν το λειτουργικό είναι μη ασφαλές το δίκτυο ή το VPN μπορούν να παραβιαστούν.

VPN λογισμικού

Τα VPN λογισμικού παρέχουν περισσότερες λειτουργίες, αλλά είναι πιο δύσκολο να εγκατασταθούν και να διαχειριστούν σε σύγκριση με τα VPN του υλικού. Τα περισσότερα VPN λογισμικού μπορούν να δρομολογούν την κίνηση βασιζόμενα στην διεύθυνση ή στο πρωτόκολλο. Η δρομολόγηση μόνο συγκεκριμένου τύπου κίνησης μας δίνει τη δυνατότητα να διαχειριστούμε τον όγκο της. Τα μειονέκτημα των VPN λογισμικού είναι το ότι είναι δύσκολα στη διαχείριση τους, αλλά και το ότι απαιτούν αλλαγές στους πίνακες δρομολόγησης και στα προσχέδια διευθύνσεων δικτύου.

1.8 Αρχιτεκτονικές Firewall

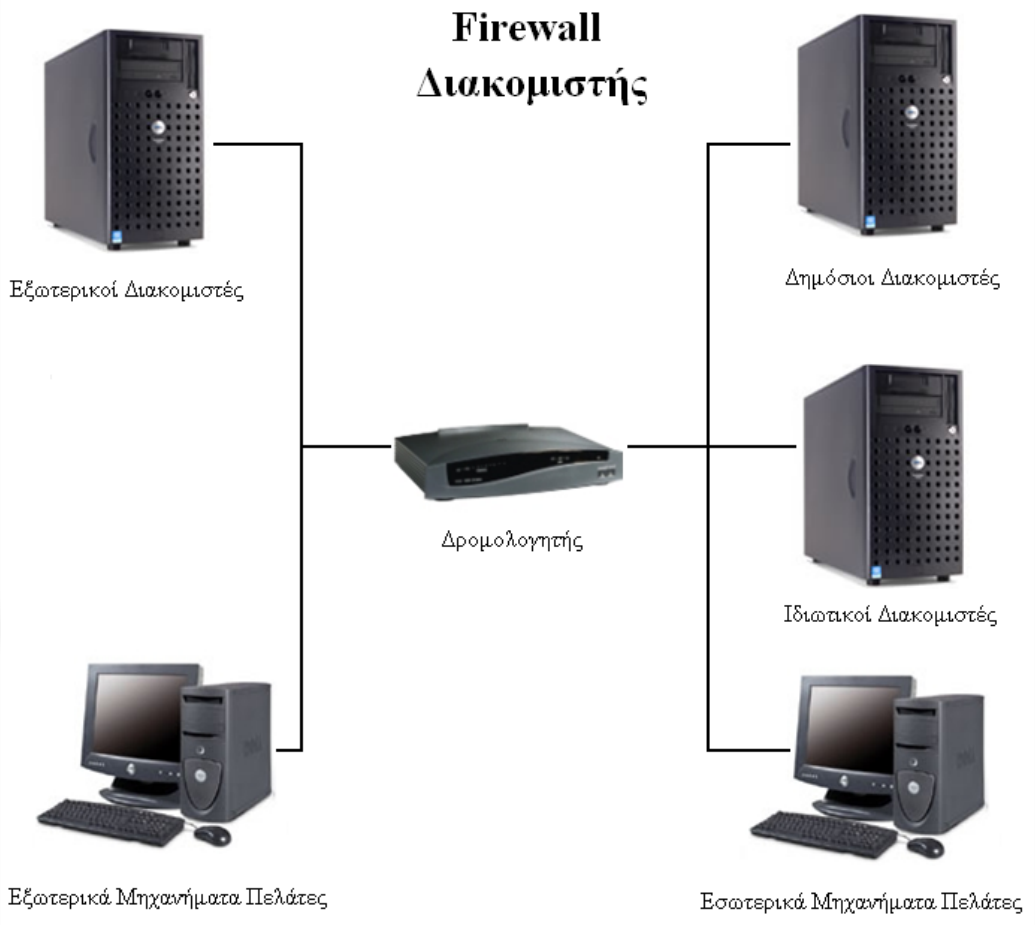
Ο σχεδιασμός ενός firewall περιλαμβάνει δύο βασικές διεργασίες.

- Σχεδιασμός των κανόνων
- Καθορισμός της τοποθέτησης του firewall ή των firewalls

Η τοποθέτηση του firewall σε ένα δίκτυο ονομάζεται αρχιτεκτονική. Η αρχιτεκτονική του firewall είναι πολύ στενά συνδεδεμένη με την αρχιτεκτονική του δικτύου. Για αυτό και ο σχεδιασμός του firewall συνδέεται με το σχεδιασμό του δικτύου. Ο σχεδιασμός του firewall είναι μια διεργασία του σχεδιασμού δικτύων. Αυτός ο τομέας περιγράφει μερικές κοινές αρχιτεκτονικές firewall.

Router Firewall

Το παρακάτω σχήμα μας δείχνει την αρχιτεκτονική η οποία ονομάζεται router firewall. Στην πραγματικότητα αυτή η αρχιτεκτονική δεν περιλαμβάνει firewall. Αντί για αυτό ένας δρομολογητής αναλαμβάνει την επικοινωνία του εξωτερικού δικτύου με το εσωτερικό. Το ότι ο δρομολογητής παρέχει λειτουργίες προώθησης πακέτων είναι μια απλή μορφή φιλτραρίσματος πακέτων η οποία μπορεί να χρησιμοποιηθεί για να προστατεύσει ένα δίκτυο. Το μέγεθος προστασίας είναι μικρό διότι η προώθηση πακέτων επιθεωρεί μόνο την διεύθυνση IP του πακέτου. Η άμυνα που παρέχει αυτή η αρχιτεκτονική δεν είναι επαρκής επειδή έχει μόνο ένα επίπεδο ασφαλείας. Ο δρομολογητής δεν μπορεί να προγραμματιστεί για να μπλοκάρει ή να δέχεται πακέτα βασιζόμενος στο port. Έτσι κάθε υπηρεσία που παρέχεται από το εσωτερικό δίκτυο είναι διαθέσιμη και ως προς τρίτους. Για αυτό δεν είναι δυνατό να παρέχουμε ιδιωτικές υπηρεσίες. Παρά τις αδυναμίες της αρχιτεκτονικής αυτής μπορούμε να την επιλέξουμε λόγο του ελάχιστου κόστους και της μέγιστης απόδοσης. Η προώθηση πακέτων αν και παρέχει χαμηλή ασφάλεια είναι η πιο γρήγορη από τις άλλες τεχνολογίες firewall.

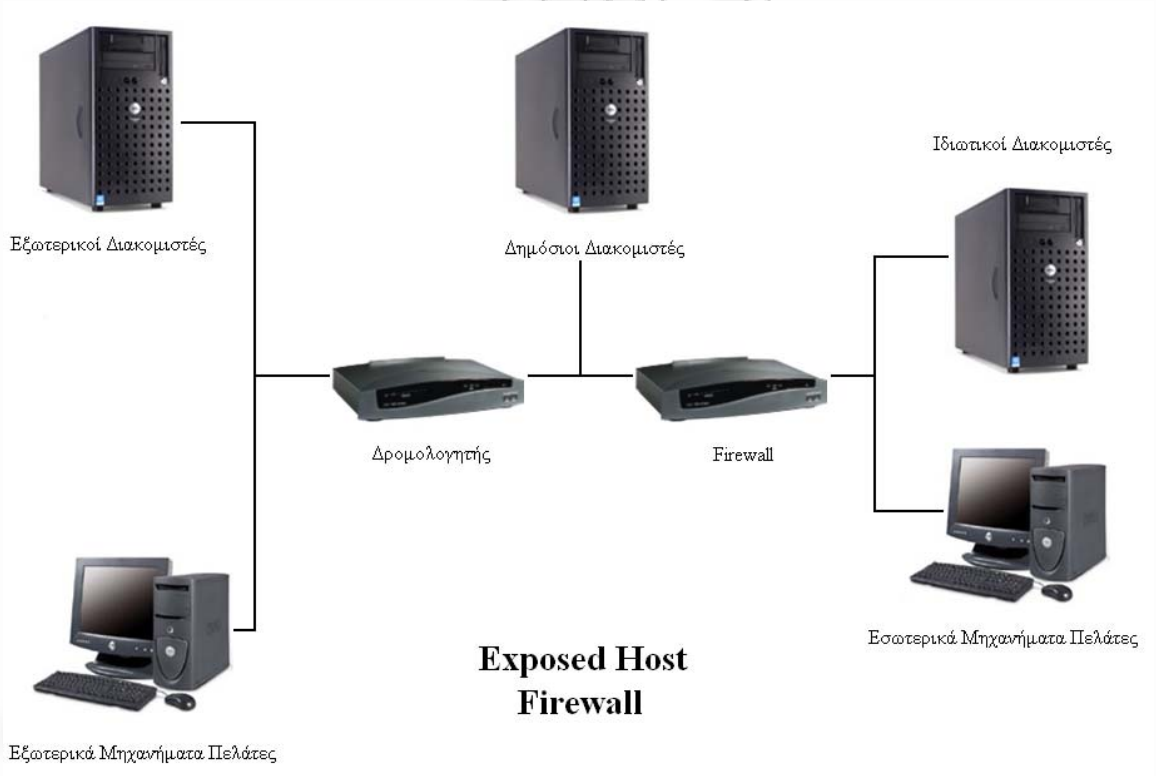


Single Host firewall

Αυτός ο τομέας εξηγεί τις αρχιτεκτονικές firewall οι οποίες πραγματοποιούνται μόνο με ένα firewall φιλτραρίσματος πακέτων ή proxy. Τα firewall φιλτραρίσματος πακέτων είναι πιο δημοφιλή από τα proxy γιατί μπορούν να στεγάσουν μεγαλύτερη ποικιλία πρωτοκόλλων. Εντούτοις τα proxy firewall είναι ικανά να πραγματοποιήσουν πιο εξειδικευμένο φιλτράρισμα από αυτά των φιλτραρίσματος πακέτων. Έχοντας ένα φιλτραρίσματος πακέτων ή proxy firewall διαιρούμε το δίκτυο σε δυο υποδίκτυα:

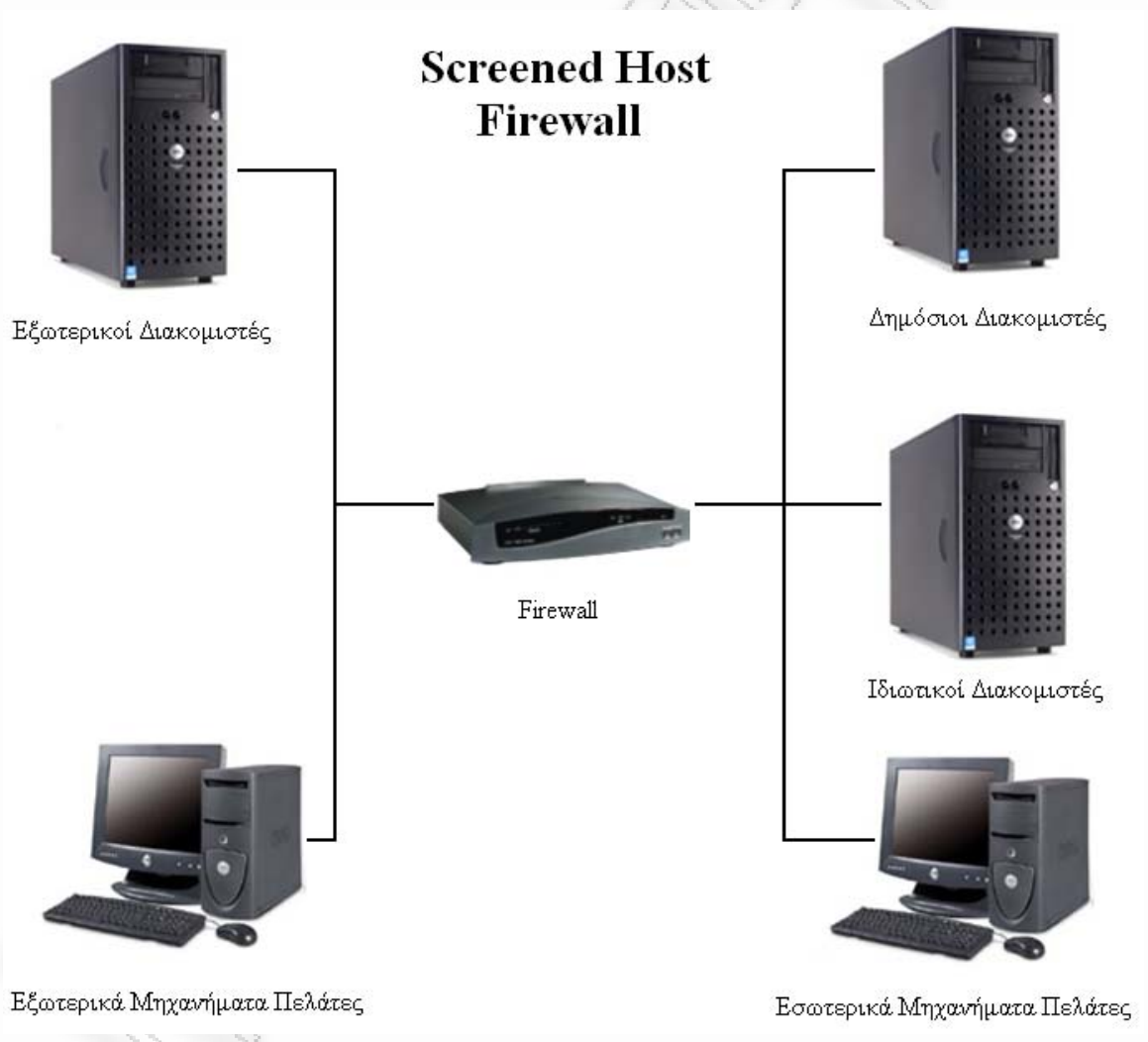
- Το εσωτερικό ιδιωτικό δίκτυο
- Το περιμετρικό δίκτυο, γνωστό σαν DMZ (DeMilitarized Zone)

Το παρακάτω σχήμα απεικονίζει μια απλή αρχιτεκτονική single host firewall γνωστή σαν exposed host firewall.



Το firewall μπορεί όχι μόνο να προωθήσει πακέτα αλλά και να φιλτράρει από και προς το εσωτερικό δίκτυο. Οι κανόνες του firewall μπορεί να είναι πιο εξειδικευμένοι από ότι στα firewall δρομολογήσεως. Αν συνδυαστούν με αυξημένη host-based ασφάλεια ένα exposed

firewall παρέχει ικανοποιητική προστασία στο εσωτερικό δίκτυο. Το αδύνατο σημείο αυτής της αρχιτεκτονικής είναι οι εκτεθειμένοι hosts. Αυτή η αρχιτεκτονική τοποθετεί τους δημόσιους διακομιστές σε μια ευαίσθητη θέση και έτσι είναι εκτεθειμένοι σε επιθέσεις από το εξωτερικό δημόσιο δίκτυο. Ένα single host firewall παρέχει μόνο δύο δίκτυα και έτσι οι δημόσιοι διακομιστές μπορεί να είναι μόνο σε μια από τις δύο πλευρές: στο περιμετρικό δίκτυο ή στο εσωτερικό ιδιωτικό δίκτυο. Η αρχιτεκτονική exposed host firewall τοποθετεί τους δημόσιους διακομιστές στο περιμετρικό δίκτυο. Μια άλλη single host αρχιτεκτονική η οποία ονομάζεται screened host firewall τοποθετεί τους δημόσιους διακομιστές στο εσωτερικό ιδιωτικό δίκτυο.



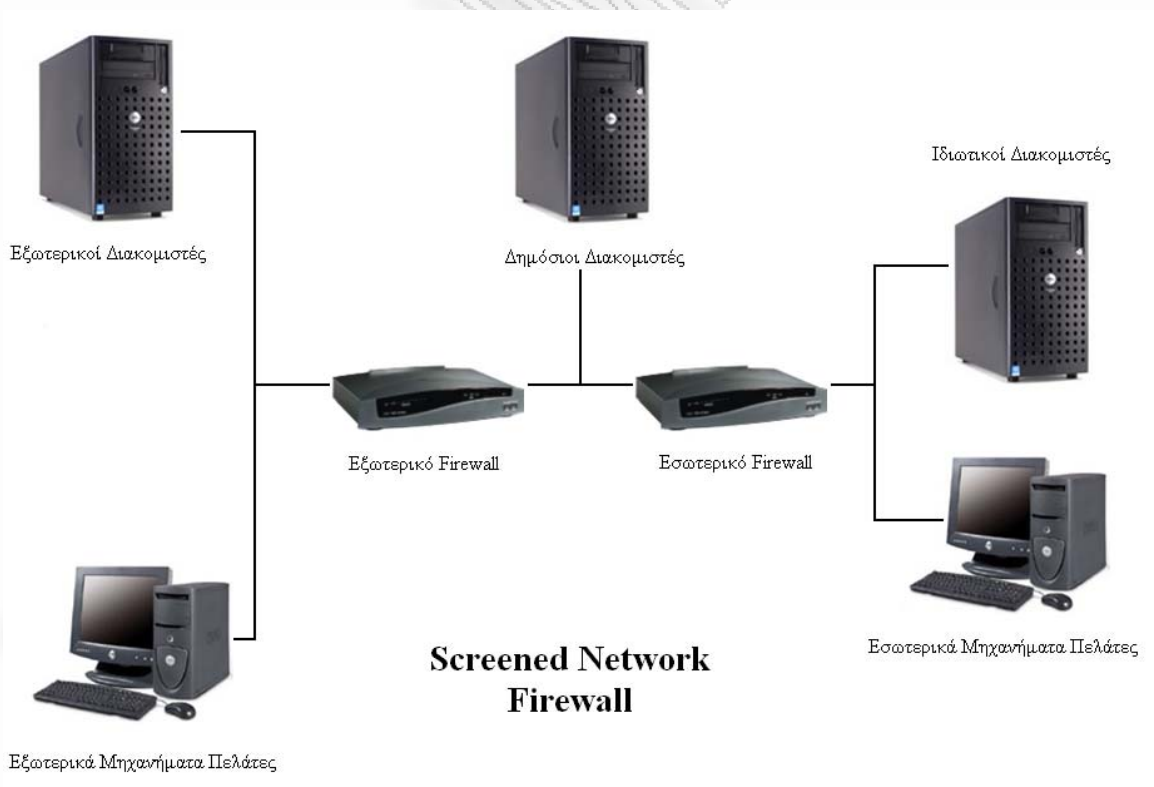
Η screened host αρχιτεκτονική μοιάζει με αυτή του δρομολογητή firewall. Η μόνη διαφορά είναι η αντικατάσταση του δρομολογητή προώθησης πακέτων με έναν φιλτραρίσματος

πακέτων ή proxy firewall. Μεταφέροντας τους δημόσιους διακομιστές πίσω από το firewall έχουμε τη δυνατότητα να φιλτράρουμε την κίνηση που κατευθύνεται από και προς τους διακομιστές και έτσι μειώνουμε την ευαισθησία τους σε επιθέσεις. Αυτή η αρχιτεκτονική είναι γενικά πιο ασφαλείς γιατί αφήνουμε εκτεθειμένο μόνο το firewall το οποίο είναι λιγότερο ευπαθή σε επιθέσεις από ότι ο host.

Multi-Host firewall

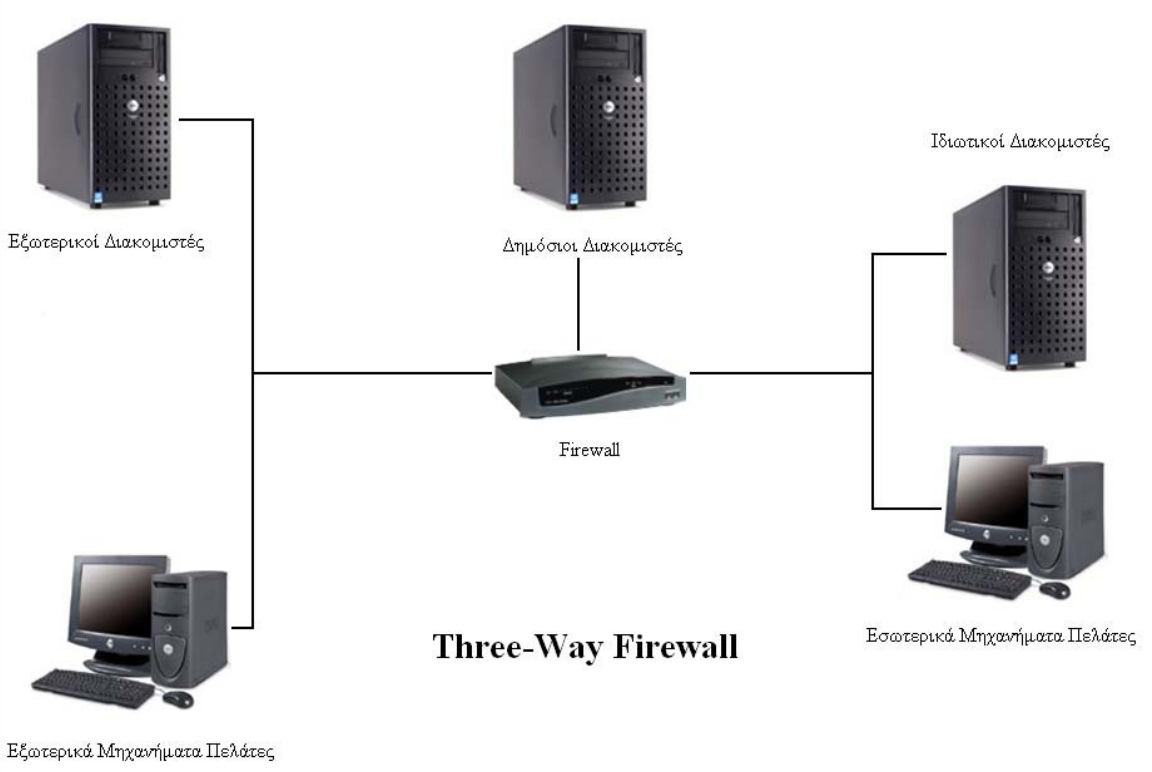
Με τα multi-host firewall μπορούμε να υπερνικήσουμε τους περιορισμούς ασφαλείας των single-host firewall. Το επόμενο σχήμα μας δείχνει μια multi-host firewall αρχιτεκτονική γνωστή σαν screened network firewall. Αυτή η αρχιτεκτονική περιλαμβάνει δύο firewall:

- Ένα εξωτερικό firewall γνωστό σαν gateway firewall
- Εσωτερικό firewall γνωστό σαν choke firewall



Η διαφορά με τα exposed host-firewall είναι ότι αντικαθιστάμε τον δρομολογητή με ένα δεύτερο firewall. Το δεύτερο firewall μπορεί να προστατεύσει τους δημόσιους διακομιστές από επιθέσεις. Κάθε Host προστατεύεται από ένα firewall. Έτσι αυτή η αρχιτεκτονική παρέχει υψηλό βαθμό ασφαλείας. Μπορούμε να προσομοιώσουμε αυτήν την αρχιτεκτονική χρησιμοποιώντας ένα single, multi-homed host.

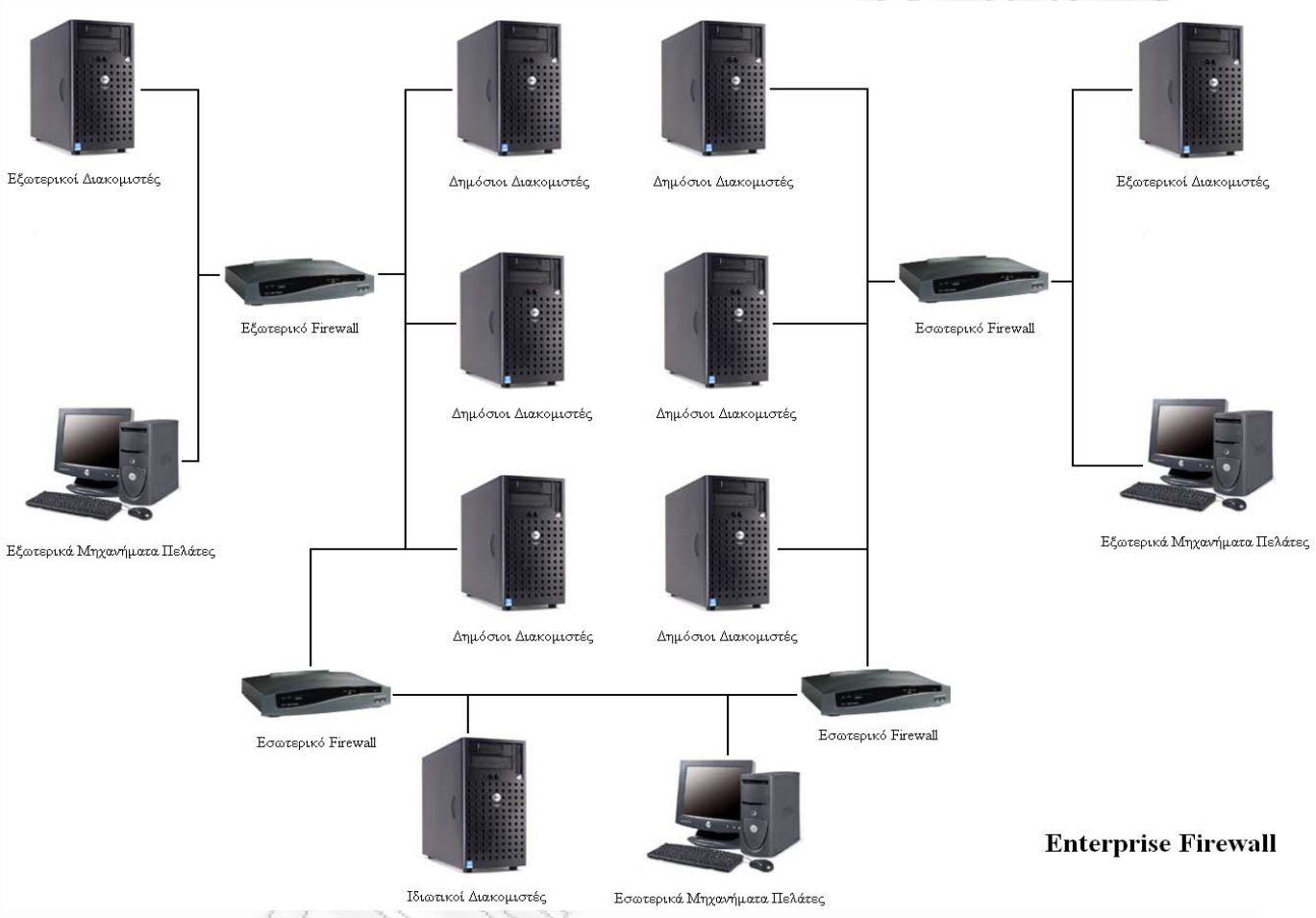
Το αποτέλεσμα απεικονίζεται παρακάτω και είναι γνωστό σαν three-way firewall γιατί έχει τρεις προσαρμογείς δικτύου.



Όπως στην screened network firewall αρχιτεκτονική έτσι και στην three-way firewall αρχιτεκτονική τοποθετούμε τους δημόσιους διακομιστές και τους ιδιωτικούς host πίσω από το firewall παρέχοντας ικανοποιητικό βαθμό ασφαλείας. Ακόμα πιο εξειδικευμένες αρχιτεκτονικές είναι πιθανές.

Μεγάλες επιχειρήσεις χρειάζονται πιο εξειδικευμένες αρχιτεκτονικές. Για παράδειγμα μια επιχείρηση που επικοινωνεί με πολλαπλά εξωτερικά δίκτυα μπορεί να χρησιμοποιήσει μια αρχιτεκτονική όπως η παρακάτω. Η αρχιτεκτονική αυτή προσφέρει υπεράριθμη δρομολόγηση από το εσωτερικό ιδιωτικό δίκτυο, στο εξωτερικό δημόσιο δίκτυο. Συνεπώς η πρόσβαση του

δικτύου είναι αξιόπιστη. Επίσης επειδή η αρχιτεκτονική αυτή βασίζεται στην screened network το επίπεδο ασφαλείας είναι αυξημένο. Ένα παράδειγμα της αρχιτεκτονικής αυτής διακρίνεται στο παρακάτω σχήμα.

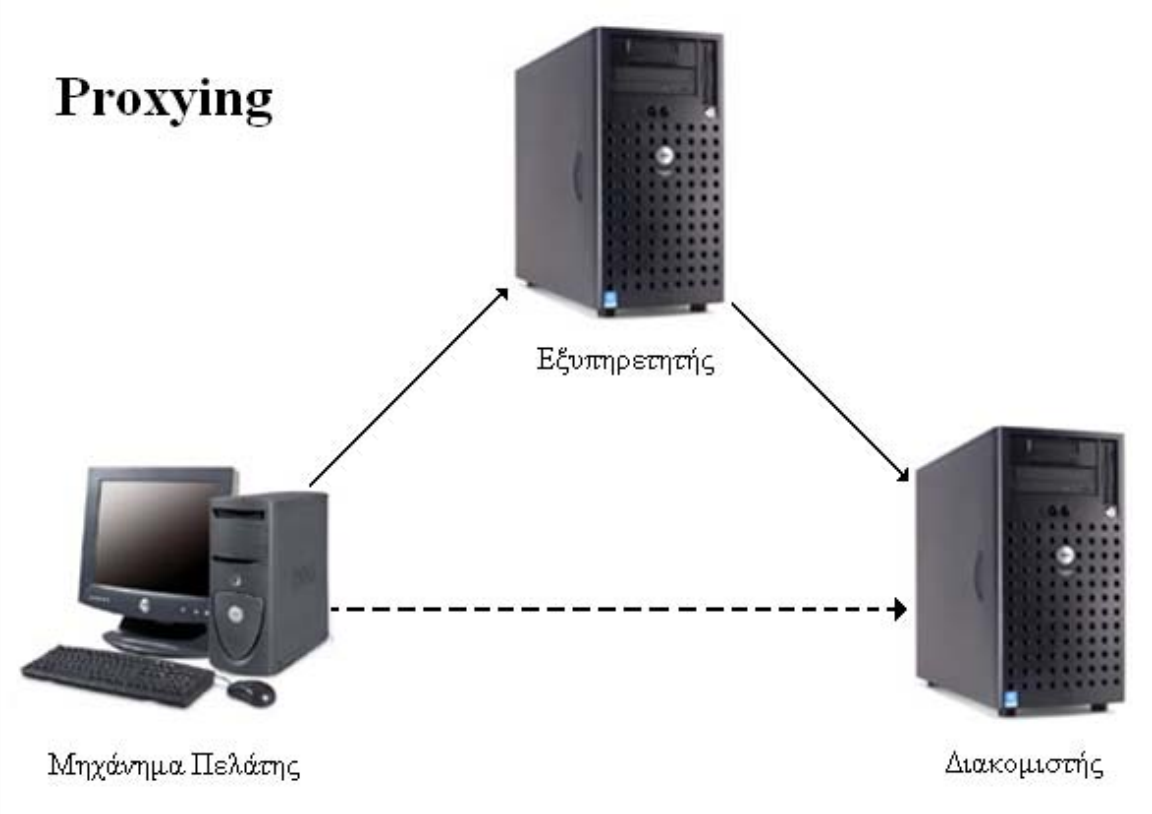


Enterprise Firewall

Η αρχιτεκτονική αυτή παρέχει πλήρης δρομολόγηση από το εσωτερικό ιδιωτικό δίκτυο στο εξωτερικό δίκτυο και έτσι η πρόσβαση είναι αξιόπιστη. Επιπλέον επειδή η μορφοποίηση βασίζεται σε screened network αρχιτεκτονική το επίπεδο ασφαλείας είναι υψηλό.

1.9 Χρήση Εξυπηρετητών (Proxying)

Μία άλλη σημαντική τεχνολογία κατασκευής firewall είναι γνωστή σαν Proxying. Αντί να επιτρέψουμε σε δύο υπολογιστές να επικοινωνήσουν μεταξύ τους κατευθείαν, τους αναγκάζουμε να επικοινωνήσουν μέσω ενός άλλου υπολογιστή ο οποίος λέγεται εξυπηρετητής (Proxy). Μερικές φορές ο εξυπηρετητής αναφέρεται και ως πύλη (Gateway). Ο εξυπηρετητής μπορεί να χρησιμοποιηθεί για να προστατεύσει ένα πελάτη ή ένα διακομιστή. Ο εξυπηρετητής ο οποίος προστατεύει ένα διακομιστή πολλές φορές αναφέρεται σαν αντίστροφος εξυπηρετητής (Reverse Proxy).



Αν συγκρίνουμε τη χρήση εξυπηρετητή με την προώθηση πακέτων θα δούμε ότι έχουν τα ίδια χαρακτηριστικά. Και οι δύο περιλαμβάνουν ένα μεσάζοντα ο οποίος επιτρέπει στους host να επικοινωνούν. Ο πιο κοινός διαχωρισμός είναι ότι η προώθηση πακέτων περνάει τα πακέτα του πελάτη στο χρήστη ενώ με τη χρήση εξυπηρετητή στέλνεται από αυτόν ένα νέο πακέτο εκ μέρους του πελάτη.

Τεχνολογίες εξυπηρετητών:

- Επιπέδου εφαρμογών (Application Level)

Πιο εξειδικευμένα εξυπηρετητές επιπέδου εφαρμογών (Application-Level) μπορούν να χειριστούν πιο πολύπλοκες διαδικασίες οι οποίες είναι πέρα από τις δυνατότητες του δρομολογητή υλικού. Μπορούν να καταλάβουν την εφαρμογή και το περιεχόμενό της ώστε να εκτελέσουν «έξυπνες» διεργασίες όπως το φιλτράρισμα mail ανάλογα με την προέλευσή του. Μελλοντικά firewall μπορεί να έχουνε ακόμα υψηλότερου επιπέδου φιλτράρισμα ώστε να ψάχνουν για πορνογραφικό υλικό ή ακόμα και για συντακτικά και ορθογραφικά λάθη. Το κλειδί για αυτό είναι το firewall να μπορεί να καταλάβει το περιεχόμενο που μεταφέρεται ακόμα και αν χρειαστεί να αναγνωρίσει το πρωτόκολλο αναλύοντας το περιεχόμενο.

- Επιπέδου κυκλώματος (Circuit Level)

Ένας εξυπηρετητής επιπέδου κυκλώματος είναι το λογισμικό το οποίο λειτουργεί σε επίπεδο σύνδεσης. Ιδρύει μία σύνδεση με την εφαρμογή η οποία τη ζήτησε και απλά προωθεί τα πακέτα και στις δυο κατευθύνσεις, χωρίς να παρεμβαίνει στο πρωτόκολλο επιπέδου εφαρμογών. Η διακομιστές επιπέδου εφαρμογών έχουν το μειονέκτημα ότι μπορούν να χειριστούν μόνο προκαθορισμένα πρωτόκολλα. Νέα ή άγνωστα πρωτόκολλα δεν λειτουργούν δια μέσω ενός διακομιστή επιπέδου εφαρμογών χωρίς να αναβαθμιστεί πρώτα το λογισμικό του. Οι διακομιστές επιπέδου κυκλώματος έχουν το προτέρημα ότι είναι γενικευμένοι έτσι ώστε να διαχειρίζονται κάθε πρωτόκολλο χωρίς ο διακομιστής να απαιτεί προηγούμενη γνώση αυτών.

Το SQUID είναι ένας δημοφιλής HTTP proxy που μπορεί να δουλέψει ως προς το συμφέρον των διακομιστών και των client. Αντιλαμβάνεται το HTTP πρωτόκολλο και μπορεί να διαχειριστεί URL ή άλλες αιτήσεις από τα μηχανήματα πελάτες. Μπορεί επίσης και εξετάζει προσεκτικά τα πακέτα και μπλοκάρει ορισμένα που είναι επικίνδυνα για το δίκτυο.

1.10 Γενικές ιδιότητες των εξυπηρετητών

Διαφανής εξυπηρετητής (Transparent Proxy)

Ο όρος διαφανής εξυπηρετητής σημαίνει ότι ο χρήστης δεν βλέπει καμία διαφορά είτε ή αίτηση γίνεται απευθείας στο διακομιστή είτε μέσω του εξυπηρετητή. Με αυτήν την έννοια

όλοι οι εξυπηρετητές είναι διαφανείς. Με τη χρήση διαφανή εξυπηρετητή ο δρομολογητής είναι προγραμματισμένος να κατευθύνει τις αιτήσεις προς τον εξυπηρετητή. Έτσι το λογισμικό του πελάτη αγνοεί εντελώς την ύπαρξη ενός μεσάζοντα εξυπηρετητή.

Άλλες ιδιότητες

- Η επιλογή της χρήσης proxy ή όχι από το χρήστη
- Ο διακομιστής προέλευσης αγνοεί την ύπαρξη του εξυπηρετητή

1.11 Τύποι εξυπηρετητών

Γενικοί Proxy διακομιστές

Οι γενικοί proxy διακομιστές είναι ο πιο κοινός τύπος των εξυπηρετητών. Χειρίζονται την κίνηση web περιλαμβάνοντας τα πρωτόκολλα HTTP, FTP κτλ. Τα χαρακτηριστικά τους είναι πλούσια. Παρέχουν πολλούς τρόπους πρόσβασης, ελέγχου, φιλτραρίσματος, logging και caching. Οι firewall proxy διακομιστές δέχονται αιτήσεις μέσα από το firewall, το οποίο τις προωθεί στο internet και επιστρέφει τα αποτελέσματα στον πελάτη. Η χρήση cache γίνεται συχνά από αυτούς τους proxy έτσι ώστε οι αιτήσεις να μην προωθούνται από το διακομιστή προέλευσης, αλλά από την cache.

Τμηματικοί Proxy διακομιστές

Οι τμηματικοί proxy διακομιστές είναι όμοιοι με τους firewall proxy διακομιστές. Το λογισμικό το οποίο χρησιμοποιείται είναι ίδιο με διαφορετικές παραμέτρους. Για παράδειγμα κάποια τμήματα μπορεί να έχουν πιο αυστηρές μεθόδους ελέγχου και ο τρόπος πρόσβασης να ποικίλει από το ένα τμήμα στο άλλο.

Αλυσωτοί Proxy διακομιστές

Οι πελάτες μπορούν να κάνουν αίτηση για δεδομένα μέσω ενός τμηματικού εξυπηρετητή ο οποίος είναι συνδεδεμένος αλυσωτά με τον firewall proxy διακομιστή. Αλυσωτά σημαίνει ότι ο τμηματικός proxy διακομιστής αναλαμβάνει τις αιτήσεις μέσω ενός άλλου proxy διακομιστή, σε αυτήν την περίπτωση του firewall proxy. Ο ένας proxy ωφελείται από τον άλλο

γιατί αν γίνει μια αίτηση για ένα αντικείμενο, αυτό χρησιμοποιείται από την cache του κοντινότερου εξυπηρετητή στην οποία υπάρχει. Η χρήση αλυσωτών εξυπηρετητών μειώνει το φόρτο εργασίας του κεντρικού firewall εξυπηρετητή. Μόνο οι αιτήσεις, τα περιεχόμενα των οποίων δεν βρίσκονται στην cache των τμηματικών εξυπηρετητών, προωθούνται στον κεντρικό firewall εξυπηρετητή.

Προσωπικοί proxy διακομιστές

Οι προσωπικοί proxy διακομιστές τρέχουν στον ίδιο host με το λογισμικό του πελάτη. Ο διαχωρισμός των χαρακτηριστικών του λογισμικού πελάτη και του προσωπικού διακομιστή είναι ασαφής. Στην πραγματικότητα κάποιος μπορεί να ισχυριστεί ότι οι προσωπικοί διακομιστές θα έπρεπε να ήταν ενσωματωμένοι με το λογισμικό του πελάτη.

Εξειδικευμένοι διακομιστές

Οι εξειδικευμένοι proxy διακομιστές λειτουργούνε σαν ομάδα και εκτελούν ειδικές λειτουργίες. Ένα καλό παράδειγμα είναι ο proxy διακομιστής ενσωματωμένος σε ένα λογισμικό ενός υπολογιστή παλάμης. Αυτός ο τύπος του proxy θα μπορούσε να μειώσει την ποιότητα εικόνας και τον αριθμό των χρωμάτων ώστε μετατρέποντας το format της να γίνεται αντιληπτή από τον υπολογιστή παλάμης. Με αυτόν τον τρόπο μειώνει τον bandwidth το οποίο απαιτείται, το οποίο είναι περιορισμένο για έναν υπολογιστή παλάμης και την ίδια στιγμή διαμορφώνει τα δεδομένα να είναι κατάλληλα για το λογισμικό και υλικό που απευθύνονται.

Αντίστροφοι Proxy διακομιστές

Ο όρος αντίστροφος proxy αναφέρεται στις ρυθμίσεις κατά τις οποίες ο proxy εκτελείται με τέτοιο τρόπο ώστε να εμφανίζεται στον πελάτη σαν ένας κανονικός web διακομιστής. Έτσι οι πελάτες συνδέονται με αυτόν θεωρώντας τον σαν έναν κανονικό διακομιστή προέλευσης χωρίς να ξέρουν αν οι αιτήσεις τους θα καθυστερηθούν περισσότερο μέσο ενός άλλου διακομιστή ή ακόμα και ενός άλλου proxy. Ο ορισμός αυτός αναφέρεται στον αντίστροφο ρόλο του proxy διακομιστή. Σε κανονική μορφή ο proxy λειτουργεί για τον πελάτη και η αίτηση γίνεται εκ μέρους του. Σε αντίστροφη μορφή ο αντίστροφος proxy λειτουργεί για τον διακομιστή και οι proxy υπηρεσίες κάνουνε αιτήσεις εκ μέρους του διακομιστή.

1.12 Γιατί οι εξυπηρετητές δεν είναι μέρος των διακομιστών web

Ο λόγος δεν είναι τόσο τεχνικός όσο το ότι υπάρχουν διαφορές στις απαιτήσεις χρηστών. Ο εξυπηρετητής και ο διακομιστής web συχνά έχουν διαφορετική βάση χρηστών. Οι διακομιστές web μπορεί να είναι εστιασμένοι σε ολόκληρο το Internet ή σε μια εταιρία ενώ οι proxy διακομιστές για εξειδικευμένη χρήση από μια εταιρία ή από ένα τμήμα της. Σε γενικές γραμμές χρησιμοποιούνται από διαφορετικά είδη ανθρώπων. Πρακτικά είναι δυνατόν να φτιάξουμε ένα διακομιστή web που να λειτουργεί και σαν proxy. Ετούτης υπάρχουν διάφοροι λόγοι για τους οποίους πρέπει να υπάρχει διαχωρισμός.

Βελτιωμένη ασφάλεια

Οι web διακομιστές δεν χρειάζεται να πραγματοποιούν συνδέσεις στο εσωτερικό δίκτυο. Έτσι το firewall μπορεί να ρυθμιστεί ώστε να κόβει συνδέσεις οι οποίες ξεκινούν από τον web διακομιστή. Αυτό προστατεύει το εσωτερικό δίκτυο αν ο web διακομιστής δεσμευτεί από επίθεση. Ακόμα και αν ένας εισβολέας αποκτήσει πρόσβαση στον web διακομιστή δεν θα μπορεί να συνδεθεί με τους host μέσα στο firewall. Οι proxy διακομιστές από την άλλη δεν χρειάζεται να είναι ικανοί να δεχτούν νέες συνδέσεις προερχόμενες από το εξωτερικό δίκτυο. Αυτό σημαίνει ότι ο proxy μπορεί να είναι προστατευμένος από εισβολείς. Οι ρυθμίσεις του firewall δρομολογητή μπορούν να κόβουν κάθε προσπάθεια σύνδεσης από την proxy διακομιστή.

Ευκολία διαχείρισης

Διαχωρίζοντας τον web διακομιστή και τον proxy διακομιστή γίνεται πιο εύκολη η διαχείρισή τους. Αυτό μειώνει τις πιθανότητες εσφαλμένης λειτουργίας. Για παράδειγμα αν ο έλεγχος πρόσβασης είναι λανθασμένα ρυθμισμένος στον διακομιστή web αυτό δεν επηρεάζει τον proxy διακομιστή και το αντίστροφο.

Δόμηση μέσω υπομονάδων

Από πλευράς ανάπτυξης λογισμικού ο διαχωρισμός των λειτουργιών κάνει την κατασκευή λειτουργικού πιο εύκολη. Διαχωρίζοντας τις λειτουργίες η σταθεροποίηση, η ανάπτυξη, η δοκιμή γίνονται πιο εύκολα και το μέγεθος του λειτουργικού γίνεται μικρότερο.

Marketing

Από πλευράς πωλήσεων λογισμικού είναι φυσικά προτιμότερο να υπάρχουν περισσότερα πακέτα. Από πλευράς αγοραστικού κοινού η δυνατότητα να αγοράσεις μόνο το λογισμικό το οποίο χρειάζεσαι είναι καλό γιατί αποφεύγεις το επιπλέον κόστος υπηρεσιών.

Κεφάλαιο 2

Απειλές και Αρχές της Άμυνας Δικτύων

2.1 Απειλές στην ασφάλεια του δικτύου

Μια αποτελεσματική στρατηγική για την άμυνα του δικτύου θα πρέπει να μπορεί να αντιμετωπίσει καινούρια είδη επιθέσεων. Ωστόσο η πλειονότητα των επιθέσεων είναι σχετικά προβλέψιμες. Έτσι πρέπει κανείς να εξασφαλίσει υψηλή ασφάλεια στο δίκτυο του για να μπορεί να επικεντρωθεί σε λιγότερο γνώριμες μορφές απειλών.

2.2 Κατηγορίες επιτιθέμενων

Σε αντίθεση με το κυρίαρχο στερεότυπο των νεαρών εισβολέων η ειδικοί της ασφάλειας αναγνωρίζουν δυο κατηγορίες εισβολέων. Τους blackhats και script kiddies.

Οι blackhats εξαιτίας της μεγάλης εξειδίκευσης τους είναι ικανοί να εξαπολύσουν μία μεγάλη γκάμα επιθέσεων εναντίων μιας μεγάλης γκάμας συστημάτων. Ωστόσο οι περισσότερες επιθέσεις εξαπολύονται από τους script kiddies που επιτίθενται με προγράμματα που κατεβάζουν από το internet. Τέτοια προγράμματα μπορεί να είναι τα scanners τα οποία εξετάζουν host και δίκτυα για ευάλωτα σημεία. Όταν βρεθεί ένα ευάλωτο σημείο οι script kiddies εκτελούν ένα πρόγραμμα με την ονομασία exploit το οποίο παραβιάζει την ασφάλεια του host. Όταν η ασφάλεια παραβιαστεί εγκαθιστούν ένα rootkit το οποίο καλύπτει την εισβολή και δίνει στον εισβολέα τον έλεγχο του host.

Οι blackhats επίσης μπορεί να χρησιμοποιήσουν αυτά τα εργαλεία, αλλά τα χρησιμοποιούν με ένα πιο εξελιγμένο τρόπο σε αντίθεση με τους script kiddies που επιτίθενται λιγότερο διακριτικά.

Παράλληλα ένας αυξανόμενος αριθμός επιθέσεων εξαπολύονται αυτόματα από προγράμματα γνωστά ως worms. Ένα τέτοιο worm εκμεταλλεύεται την ευαισθησία ενός Microsoft SQL διακομιστή για να πάρει τον έλεγχο του host θύματος. Έτσι σε σύντομο χρόνο το worm προσβάλει χιλιάδες συστήματα και δημιουργεί μια «πίσω πόρτα» από όπου ένας εισβολέας μπορεί να ελέγξει ένα μολυσμένο σύστημα. Έτσι οι script kiddies βρίσκουν ήδη μολυσμένα συστήματα και τα χρησιμοποιούν για δικούς τους σκοπούς.

2.3 Κίνητρα των εισβολέων

Οικονομικό ή προσωπικό όφελος

Οι εισβολείς υπολογιστών συχνά στοχεύουν βάσεις δεδομένων που περιέχουν πληροφορίες πιστωτικών καρτών. Ένας εισβολέας που μπαίνει σε μια τέτοια βάση δεδομένων μπορεί να έχει πρόσβαση σε χιλιάδες πιστωτικές κάρτες. Ωστόσο η επίθεση μπορεί να είναι και για προσωπικό όφελος και ένα παράδειγμα είναι οι φοιτητές που εισβάλουν σε βάσεις για την αλλαγή βαθμών.

Πρόσβαση σε υπολογιστικούς πόρους

Ορισμένες φορές οι εισβολείς θέλουν απλά να έχουν πρόσβαση στους υπολογιστικούς πόρους των συστημάτων που στοχεύουν . Για παράδειγμα ένας blackhat που θέλει να μολύνει ένα σύστημα μπορεί πρώτα να επιτεθεί σε διάφορα άλλα μη σχετιζόμενα συστήματα και να τα χρησιμοποιήσει για να καλύψει την ταυτότητα του.

Αναδημιουργία και εξαπάτηση

Μερικοί εισβολείς και πιο συγκεκριμένα οι script kiddies αντιμετωπίζουν τις εισβολές τους σαν σπορ και συχνά ο μοναδικός τους σκοπός δεν είναι άλλος από το να εισβάλουν στον σύστημα. Απολαμβάνουν την εκτίμηση των άλλων ανάλογα με τον αριθμό των συστημάτων που έχουν εισβάλει και συχνά ανακοινώνουν στο δίκτυο τα επιτεύγματα τους.

Πολιτικές σκοπιμότητες

Σε ορισμένες περιπτώσεις ο στόχος των εισβολέων είναι να ενοχλήσουν ολικά πρόσωπα που έχουν διαφορετικές απόψεις από τις δικές τους, αλλά και δημόσιους οργανισμούς ή εταιρίες που αντιπαθούν. Για παράδειγμα πολλοί blackhats μπορεί να αντιτίθενται στην βιομηχανία εμπορικής ασφάλειας και για αυτό επιχειρούν να παραβιάσουν αυτές τις εταιρίες και ειδικότερα όσες υποστηρίζουν ότι είναι ιδιαίτερα αξιόπιστες.

2.4 Τύποι επιθέσεων

Αυτή η ενότητα εξετάζει μερικούς από τους τρόπους με τους οποίους διαπράττονται παράνομες δραστηριότητες στο Internet. Αυτές κυμαίνονται από ενέργειες οι οποίες απλά εκμεταλλεύονται την απλή ανθρώπινη αδυναμία έως αυτές που απαιτούν εξειδικευμένες τεχνολογικές γνώσεις και βαθιά κατανόηση της δομής του Internet. Ωστόσο, πριν εξετάσουμε αυτούς τους τρόπους με τους οποίου μπορεί να απειληθεί ένα σύστημα, αξίζει να δούμε τους τύπους απειλών που μπορεί να αντιμετωπίσει ένα Web Site.

- *Απειλές ακεραιότητας δεδομένων.* Αυτές οι απειλές αφορούν την παραποίηση αποθηκευμένων δεδομένων από έναν εισβολέα όπως την αλλαγή στοιχείων πιστωτικών καρτών σε μια βάση δεδομένων ή την παραποίηση στοιχείων κατά την μεταφορά τους όπως την μεταβολή ενός μηνύματος κατά τη μεταφορά του.
- *Απειλές εμπιστευτικών δεδομένων.* Αυτές οι απειλές αφορούν την ανάγνωση σημαντικών αποθηκευμένων δεδομένων από μη εξουσιοδοτημένα άτομα όπως π.χ. διοικητικά μυστικά εταιρειών κ.λ.π.
- *Απειλές άρνησης υπηρεσιών (Denial of Service - DoS).* Αυτές οι απειλές αφορούν το πλημμύρισμα ενός Web server με μεγάλο αριθμών αιτημάτων ώστε να μην μπορεί πλέον αυτός να λειτουργήσει λόγω έλλειψης πόρων.
- *Απειλές πιστοποίησης χρηστών.* Σε τέτοιου είδους απειλές ο εισβολέας προσποιείται πως είναι ένας χρήστης ενώ δεν είναι.

Μη-τεχνολογικές επιθέσεις

Αυτές είναι επιθέσεις οι οποίες είτε βασίζονται σε κάποια αδυναμία μιας επιχείρησης ή οργανισμού είτε απαιτούν ελάχιστες γνώσεις υπολογιστών για να γίνουν. Μερικά παραδείγματα είναι:

- Να μαντέψει κανείς το password κάποιου άλλου και να αποκτήσει έτσι πρόσβαση στα αρχεία του. Συχνά τα passwords επιλέγονται ώστε να είναι ευκολομνημόνευτα, για παράδειγμα το όνομα της συζύγου του κατόχου του λογαριασμού, των παιδιών του κ.λ.π. Όταν τα password είναι ευκολομνημόνευτα, μπορεί να μην είναι τελικά τόσο δύσκολο για κάποιον να το μαντέψει.
- Να κλέψει κανείς password το οποίο είναι εύκολο να βρεθεί με φυσικό τρόπο, π.χ. θα μπορούσε να είναι γραμμένο σε ένα πίνακα, σε ένα ημερολόγιο ή σε ένα κομμάτι χαρτί στο συρτάρι σας.
- Να εκμεταλλευτεί κανείς ελλιπείς φυσικούς η λειτουργικούς ελέγχους. Υπάρχουν για παράδειγμα αρκετές ιστορίες υπαλλήλων τραπεζών οι οποίοι εκδίδανε πιστωτικές κάρτες σε μη υπάρχοντες πελάτες, κρατούσαν την κάρτα οι ίδιοι και την χρησιμοποιούσαν. Αυτός ο τύπος απειλής είναι βέβαια πολύ μακριά από μια τεχνολογική απειλή και απλά βασίζεται σε εσωτερικές αδυναμίες ενός οργανισμού.
- Να γράψει ένα μικρό πρόγραμμα το οποίο παρουσιάζει ένα παράθυρο το οποίο ζητά από τον χρήστη κάποιες σημαντικές πληροφορίες όπως στοιχεία πιστωτικών καρτών, password κ.λ.π. Ένα συνηθισμένο παράθυρο ήταν κάποιο το οποίο έλεγε πως έχει πέσει η σύνδεση τους και ότι πρέπει να συνδεθούν ξανά δίνοντας ένα password. Οι γλώσσες Java και JavaScript μπορούν να κάνουν τη συγγραφή ενός τέτοιου προγράμματος αρκετά εύκολη.

Καταστροφικές επιθέσεις

Αυτά είναι τα ηλεκτρονικά αντίστοιχα των βομβών: απαιτούν πολύ λίγες γνώσεις αλλά τα αποτελέσματα τους μπορεί να είναι αρκετά σημαντικά.

Είδη επιθέσεων DoS (Denial of Service)

Τον τελευταίο καιρό έχουν γίνει πολλές συζητήσεις για το θέμα των επιθέσεων DoS χάρη στις οποίες είναι δυνατή η σχετικά εύκολη απενεργοποίηση ενός κόμβου Internet, ακόμη και από άτομα με μικρές τεχνικές γνώσεις. Παρακάτω περιγράφουμε τα κυριότερα είδη αυτών

των επιθέσεων οι οποίες έχουν ως κοινό χαρακτηριστικό την αποστολή στο θύμα (τον server που αποτελεί τον στόχο της επίθεσης) ενός τόσο μεγάλου αριθμού πλαστών αιτημάτων σύνδεσης ώστε δεν μπορεί πλέον να διαχειριστεί το παραμικρό και διακόπτει τη λειτουργία του.

Οι κυριότερες μορφές επιθέσεων αυτής της μορφής είναι:

Ping of Denial (γνωστή και ως Ping of Death)

Πρόκειται για την παλαιότερη και πιο διαδεδομένη μορφή επίθεσης. Για καθαρά τεχνικούς λόγους κάθε server συνδεδεμένος με το Internet πρέπει να μπορεί να δεχθεί μηνύματα ping στα οποία απαντά αποστέλλοντας μια σειρά από μηνύματα pong χάρη στα οποία μετράται η ταχύτητα ανταπόκρισής του.

Για την επίθεση αυτή ο επιτιθέμενος απλώς αποστέλλει πάρα πολλά μηνύματα ping τα οποία ο server είναι υποχρεωμένος να απαντήσει, δαπανώντας φυσικά υπολογιστική ισχύ και bandwidth. Αν τα μηνύματα ping είναι πάρα πολλά τότε ο αποδέκτης τους καθυστερεί σημαντικά στην εκτέλεση άλλων εργασιών (π.χ. αποστολή web σελίδων) διότι είναι πολύ απασχολημένος στέλνοντας pong, ενώ αν ο φόρτος γίνει πολύ μεγάλος είναι πιθανό να διακόψει τελείως τη λειτουργία του.

Δυστυχώς, η άμυνα από το Ping of Death είναι εξαιρετικά δυσχερής, καθώς το Ping αποτελεί τη μέθοδο με την οποία ένας Η/Υ δηλώνει ότι είναι ενεργός μέσα στο δίκτυο. Δεν είναι λοιπόν δυνατόν να αρνηθεί να απαντήσει σε όποιο Ping δέχεται. Τελευταία, μερικά συστήματα έχουν αποκτήσει μεγάλους Ping buffers και έτσι μπορούν είτε να απαντούν σε πολλά Ping χωρίς να επηρεάζεται η λειτουργία τους, είτε να επεξεργάζονται όσα Ping δέχονται, αναγνωρίζοντας και αγνοώντας αυτόματα όποια από αυτά αποτελούν προϊόν επιθέσεως.

ICMP

Το πρωτόκολλο ICMP χρησιμοποιείται για την επικοινωνία μεταξύ υπολογιστών, χωρίς να χρειάζεται η υλοποίηση ενός ισχυρότερου και πιο περίπλοκου (συνεπώς και πιο αργού) πρωτοκόλλου όπως το TCP. Το ICMP μεταφέρει πολύ λίγες πληροφορίες οι οποίες ενημερώνουν κάθε υπολογιστή για την κατάσταση της σύνδεσής του με άλλα μηχανήματα. Για να "κλείσει" τη σύνδεση ενός Η/Υ ο επιτιθέμενος δεν έχει παρά να του στείλει μέσω του ICMP ένα από τα ακόλουθα μηνύματα:

- Destination Unreachable
- Time to Live Exceeded
- Parameter Problem
- Packet Too Big
- Source Quench

Ουσιαστικά δηλαδή, ο επιτιθέμενος δηλώνει απλώς στο θύμα πως υπάρχει πρόβλημα και ο αποδέκτης διακόπτει τη σύνδεσή του!

Θεωρητικά, αυτό το πρόβλημα μπορεί να αντιμετωπιστεί αν απενεργοποιηθεί (κλείσει) η ICMP port (στα Windows αυτή είναι η default επιλογή). Δυστυχώς, με τον τρόπο αυτό μειώνεται η ταχύτητα σύνδεσης ενός Η/Υ με το δίκτυο. Έτσι, πολλοί administrators προτιμούν να διατηρούν αυτή τη δυνατότητα εν λειτουργία, παρ' όλους τους κινδύνους που συνεπάγεται κάτι τέτοιο.

Fragmentation

Αποτελεί τον πιο μοντέρνο, και γι' αυτό πιο δύσκολο στην αντιμετώπιση, τρόπο επιθέσεως. Όταν δύο Η/Υ επικοινωνούν με το πρωτόκολλο TCP/IP ουσιαστικά ανταλλάσσουν αρχεία τα οποία ο αποστολέας "τεμαχίζει" σε μικρότερα κομμάτια τα οποία και αποστέλλει στον παραλήπτη ο οποίος τα συναρμολογεί, ανασυνθέτοντας το αρχικό αρχείο.

Τα πακέτα αυτά περιέχουν μια σειρά από στοιχεία ελέγχου μέσω των οποίων ο παραλήπτης ελέγχει αν τα δεδομένα έφτασαν σε καλή κατάσταση. Σε περίπτωση που διαπιστωθεί κάποιο πρόβλημα, τότε ο παραλήπτης επικοινωνεί με τον αποστολέα και του ζητάει να ξαναστείλει τα πακέτα που αλλοιώθηκαν κατά τη μεταφορά.

Εκμεταλλευόμενος αυτό το χαρακτηριστικό, ο επιτιθέμενος στέλνει συνεχώς πακέτα με λανθασμένα στοιχεία ελέγχου. Έτσι, υποχρεώνει τον παραλήπτη να σπαταλά υπολογιστική ισχύ και bandwidth, ζητώντας συνεχώς την επανάληψη της αποστολής τους.

Αν και έχουν βρεθεί κάποια αντίμετρα για τις επιθέσεις Fragmentation, κανένα από αυτά δεν είναι αρκετά αποτελεσματικό. Αν η επίθεση συνεχιστεί για μεγάλο χρονικό διάστημα ή αν γίνεται από μια γρήγορη γραμμή, τελικά το θύμα θα υποχρεωθεί να αποσυνδεθεί από το δίκτυο.

E-mail bombing

Αν και από πολλούς δεν θεωρείται ως Denial of Service Attack, το E-mail bombing είναι πολύ αποτελεσματικό όταν χρησιμοποιείται εναντίον υπολογιστών οι οποίοι διαχειρίζονται mail. Το μόνο που έχει να κάνει κανείς είναι να τους στείλει τόσα πολλά (σε μέγεθος και αριθμό) e-mail μηνύματα, ώστε ο φόρτος των εργασιών διαχείρισής τους να οδηγήσει το σύστημα σε κατάρρευση.

Port Flooding

Όλοι οι υπολογιστές διαθέτουν μια σειρά από λογικές πόρτες μέσω των οποίων μπορεί ένα άλλο μηχάνημα να συνδεθεί μαζί τους για να εκτελέσει μια σειρά από εργασίες. Για παράδειγμα, στο UNIX η πόρτα 25 χρησιμοποιείται από την υπηρεσία Sendmail. Ο επιτιθέμενος μπορεί να γράψει ένα πρόγραμμα για οποιαδήποτε πόρτα το οποίο θα ζητάει να ανοιχθούν μέσω αυτής όσο γίνεται περισσότερες (πλαστές φυσικά) συνδέσεις. Όσο αυξάνει ο αριθμός των συνδέσεων, τόσο μεγαλύτερος γίνεται και ο φόρτος για το μηχάνημα το οποίο επιβαρύνεται όλο και περισσότερο μέχρι που τελικά "κολλάει" και διακόπτει τη λειτουργία του.

SYN Flooding

Το πρωτόκολλο SYN-ACK αποτελεί τη βάση κάθε έναρξης σύνδεσης μέσα στο Internet. Όταν ένας Η/Υ θέλει να συνδεθεί με έναν άλλο του αποστέλλει ένα πακέτο SYN στο οποίο ο server απαντάει με ένα πακέτο ACK (acknowledge). Όταν ο Η/Υ που ζήτησε τη σύνδεση λάβει το ACK θεωρεί ότι η σύνδεση έχει ολοκληρωθεί και αρχίζει τη μετάδοση των δεδομένων. Σε μια επίθεση SYN Flooding ο επιτιθέμενος στέλνει συνεχώς πακέτα SYN αλλά όχι ACK. Έτσι, ο server που δέχεται την επίθεση είναι υποχρεωμένος για περιμένει για κάποιο χρονικό διάστημα το ACK, δεσμεύοντας φυσικά με την αναμονή αυτή ένα μέρος των διαθέσιμων πόρων του. Αν το ACK δεν έρθει ποτέ, το θύμα θα τερματίσει την αναμονή και θα ασχοληθεί με άλλες δραστηριότητες. Γι' αυτό ο επιτιθέμενος συνήθως στέλνει έναν συνεχώς αυξανόμενο αριθμό πακέτων SYN, δεσμεύοντας όλο και περισσότερους πόρους του θύματος στην αναμονή των ACK τους και οδηγώντας αργά τον server του θύματος στην κατάρρευση. Δυστυχώς δεν είναι πρακτικά δυνατόν να αμυνθεί κανείς σε επιθέσεις αυτής της μορφής, καθώς το πρωτόκολλο SYN-ACK αποτελεί τη βάση κάθε σύνδεσης μέσα στο Internet. Γι' αυτό και η μόνη απάντηση που έχει προταθεί μέχρι σήμερα είναι να θέτει ο administrator κάθε μηχανήματος ένα όριο στον αριθμό των συνδέσεων τις οποίες θα εξυπηρετεί ο server,

αγνοώντας πολλά από τα SYN που λαμβάνει. Δυστυχώς όμως, με τον τρόπο αυτό κινδυνεύουν να μην εξυπηρετηθούν και οι νόμιμοι χρήστες του συστήματος αφού δεν είναι εύκολο να ξεχωρίσει κανείς τα αποδεκτά από τα κακόβουλα SYN.

Οι επιθέσεις Denial of Service ως μέσο εισβολής σε ένα σύστημα

Για τους περισσότερους ειδικούς, οι επιθέσεις DoS θεωρούνται πολύ ενοχλητικές, αλλά όχι θανάσιμα επικίνδυνες για ένα σύστημα. Ο επιτιθέμενος μπορεί ίσως να διακόψει τη λειτουργία ενός μηχανήματος για μερικές ώρες, αλλά δεν έχει τη δυνατότητα να αποκτήσει πρόσβαση σε αυτό και να τροποποιήσει δεδομένα σε κάποιο από τα άλλα μηχανήματα τα οποία μοιράζονται το ίδιο δίκτυο.

Δυστυχώς, αυτό δεν είναι αλήθεια. Μια καλή και δοκιμασμένη τεχνική (χρησιμοποιήθηκε από τον Kevin Mitnick για να εισβάλει στο σύστημα του διώκτη του κ. Tsutomu Shinomura) είναι η εισβολή σε ένα δίκτυο μέσω επίθεσης DoS σε έναν από τους servers του. Όταν το μηχάνημα αυτό πάψει να λειτουργεί, τότε ο επιτιθέμενος επικοινωνεί με άλλα μηχανήματα του ίδιου δικτύου "υποκρινόμενος" ότι τα πακέτα που στέλνει προέρχονται από το αχρηστεμένο και εκτός λειτουργίας πλέον μηχάνημα. Με τον τρόπο αυτό αυξάνονται σημαντικά οι πιθανότητες να επιτευχθεί πρόσβαση στα άλλα μηχανήματα του δικτύου, καθώς η εντολή πρόσβασης δεν δίνεται από έναν τρίτο, αλλά από μια έμπιστη πηγή (ένα μηχάνημα εντός του δικτύου).

Ιοί

Οι ιοί είναι ένα είδος κακόβουλα γραμμένου κώδικα που θέτει σε κίνδυνο την ασφαλή λειτουργία του συστήματος και μπορούν να χρησιμοποιηθούν για μια ποικιλία διαφορετικών επιθέσεων. Περιγράφονται ξεχωριστά εδώ καθώς απαιτούν αρκετά υψηλό επίπεδο τεχνικών γνώσεων. Οι συνηθισμένες επιθέσεις άρνησης υπηρεσιών είναι απλές και όχι ιδιαίτερα εξεζητημένες ενώ η επίθεση με ιό απαιτεί μεγαλύτερο βαθμό τεχνικών γνώσεων.

Ένας **ιός** είναι ένα πρόγραμμα το οποίο επισυνάπτει τον εαυτό του σε αρχεία τα οποία υπάρχουν στον υπολογιστή - μια διαδικασία που είναι γνωστή ως **μόλυνση** Αφού ένας ιός εγκατασταθεί σε έναν υπολογιστή, μπορεί να αντιγράψει τον εαυτό του και σε άλλα αρχεία στον υπολογιστή.

Κατηγορίες ιών

Υπάρχουν τρεις κύριες κατηγορίες ιών: εκτελέσιμοι ιοί, ιοί δεδομένων και ιοί οδηγών συσκευών. Ένας **εκτελέσιμος ιός** είναι ένας ιός ο οποίος προστίθεται σε ένα εκτελέσιμο αρχείο, το οποίο όταν εκτελεστεί θα έχει ως αποτέλεσμα να εκτελεστεί και ο κώδικας του ιού. Αυτός ο κώδικας στη συνέχεια θα κάνει κάποια κακόβουλη ενέργεια όπως να διαγράψει κάποια σημαντικά αρχεία. Ένας **ιός δεδομένων** είναι ένας ιός ο οποίος μολύνει ένα αρχείο που περιέχει δεδομένα αντί για εκτελέσιμο κώδικα. Συχνά τα δεδομένα αυτά είναι συνδεδεμένα με κάποιο πρόγραμμα, το οποίο χρειάζεται τα δεδομένα για να εκτελέσει τη λειτουργία του. Για παράδειγμα, πολλά προγράμματα χρειάζονται ένα **startup αρχείο** το οποίο αρχικοποιεί το πρόγραμμα και ορίζει βασικές παραμέτρους της λειτουργίας του. Ένας ιός δεδομένων θα μπορούσε να μολύνει ένα τέτοιο αρχείο και να αλλάξει τα δεδομένα σε ένα τέτοιο αρχείο ώστε το πρόγραμμα δεν θα μπορεί να λειτουργήσει ή η λειτουργία του θα τεθεί σε κίνδυνο. Ένας άλλος τύπος ιού δεδομένων θα μπορούσε να προσθέσει μια καταχώρηση σε ένα αρχείο με password κι έτσι θα επέτρεπε πρόσβαση σε ένα εισβολέα. Άλλο ένα παράδειγμα είναι αυτό ενός ιού δεδομένων για έναν επεξεργαστή κειμένου, που θα μπορούσε επίσης να γραφτεί και εύκολα και που θα μπορούσε να αλλάζει τα περιεχόμενα κάθε αρχείου που ανοίγει από τον επεξεργαστή κειμένου ή ακόμη χειρότερα να το σβήνει. Μια τρίτη κατηγορία είναι οι **ιοί οδηγών συσκευών**. Αυτοί επηρεάζουν τους οδηγούς συσκευών ενός λειτουργικού συστήματος που χρησιμοποιούνται για τον χειρισμό διαφόρων στοιχείων του υπολογιστή όπως δίσκος. Ευτυχώς αυτός ο τύπος ιού εμφανιζόταν κυρίως σε παλιότερα λειτουργικά συστήματα όπως το MS-DOS.

Υπάρχει επίσης ένας τρόπος κατηγοριοποίησης των ιών βάσει του τρόπου που χρησιμοποιούν για να κρύψουν την παρουσία τους στον υπολογιστή. Βάσει του κριτηρίου αυτού υπάρχουν δυο ειδών ιοί, οι **stealth ιοί** και οι **πολυμορφικοί ιοί**.

Πριν περιγράψουμε αυτούς τους δυο τύπους ιών είναι χρήσιμο να μιλήσουμε για το πως τα προγράμματα εντοπισμού ιών λειτουργούν. Τα προγράμματα αυτά λειτουργούν ελέγχοντας τα αρχεία που υπάρχουν αποθηκευμένα ψάχνοντας σε αυτά είτε γνωστούς ιούς είτε αλλαγές σε σημαντικά αρχεία, π.χ. αλλαγές σε αρχεία κώδικα του λειτουργικού συστήματος παρότι δεν υπήρξε άμεση αναβάθμιση του.

Οι ιοί Stealth κρύβουν την παρουσία τους χρησιμοποιώντας μερικές διάφορες τεχνικές, για παράδειγμα αλλάζοντας τις ημερομηνίες αλλαγής ή το πραγματικό μέγεθος των αρχείων

ώστε το πρόγραμμα εντοπισμού ιών να μην μπορεί να εντοπίσει κάποια αλλαγή και να μην θεωρεί ύποπτα αρχεία τα οποία στην πραγματικότητα είναι μολυσμένα.

Οι πολυμορφικοί ιοί μπορούν να αλλάζουν συχνά τα χαρακτηριστικά τους - για παράδειγμα το μέγεθος τους - μια διαδικασία που είναι γνωστή ως **μετάλλαξη**. Αυτό σημαίνει ότι είναι πολύ πιο δύσκολο για τα προγράμματα εντοπισμού ιών να τους εντοπίσουν βασιζόμενοι μόνο στα γνωστά χαρακτηριστικά τους.

Δημιουργία ιών

Υπάρχουν διάφοροι τρόποι δημιουργίας ιών. Μπορούν να δημιουργηθούν από την αρχή χρησιμοποιώντας μια γλώσσα όπως η C ή assembly. Χρησιμοποιούνται τέτοιες γλώσσες γιατί πρέπει ο κώδικας του ιού να είναι όσο το δυνατόν μικρότερος για να μπορεί να αποφεύγει τον εντοπισμό από προγράμματα anti-virus. Οι γλώσσες αυτές επίσης παρέχουν αρκετές δυνατότητες σχετικά χαμηλού επιπέδου που δεν προσφέρονται από άλλες γλώσσες όπως κάποιες λειτουργίες εισόδου / εξόδου. Υπάρχουν επίσης κάποια εργαλεία κατασκευής ιών τα οποία μπορούν να βρεθούν σε διάφορα μέρη στο internet.

Ένας τυπικός ιός

Θα συμπληρώσουμε την ενότητα των ιών εξετάζοντας τη λειτουργία ενός συγκεκριμένου ιού βλέποντας έτσι πόσο πονηροί μπορεί να είναι οι κατασκευαστές ιών. Ο ιός αυτός είναι γνωστός σας ιός οικογένειας και φίλων. Χρησιμοποιεί επισυναπτόμενα αρχεία σε e-mails για να διαδοθεί ωστόσο το κάνει με ένα ιδιαίτερα πονηρό τρόπο.

Υπάρχει ένας ιδιαίτερα μεγάλος αριθμός ιών (εκτελέσιμων ιών ή ιών δεδομένων) που διαδόθηκαν μέσω e-mail. Το μόνο που πρέπει να κάνει ο παραλήπτης ενός μολυσμένου e-mail είναι να ανοίξει ένα επισυναπτόμενο αρχείο. Το αποτέλεσμα θα είναι να εκτελεστεί ένα πρόγραμμα που θα μολύνει τον υπολογιστή του παραλήπτη. Συχνά τέτοια e-mail έχουν μια απλή επικεφαλίδα όπως 'Γεια' ή 'Πως πάει;' που ίσως αποτελεί ένδειξη ότι ο αποστολέας είναι γνωστός του παραλήπτη και το αρχείο πιθανότατα μπορεί με ασφάλεια να ανοιχθεί. Μια άλλη ανάλογη μέθοδος είναι μέσω ηλεκτρονικών ευχετήριων καρτών, όπου τα ύποπτα επισυναπτόμενα αρχεία μπορεί να φαίνονται σαν ευχετήριες κάρτες.

Υπήρξε έντονη δημοσιότητα σχετικά με τη διάδοση ιών μέσω e-mail και σαν συνέπεια πολλοί χρήστες του Internet είναι διστακτικοί να ανοίξουν επισυναπτόμενα αρχεία από χρήστες που δεν γνωρίζουν. Αυτό είναι το ιδιαίτερο σημείο στο οποίο οι ιοί του τύπου "φίλοι και

οικογένεια" αποδεικνύονται τόσο ύπουλοι. Ο ιός μπορεί μολύνει κάποιον υπολογιστή χρησιμοποιώντας κάποιο άλλο μέσο και όχι e-mail, για παράδειγμα μπορεί να μολύνει κάποιο υπολογιστή όταν ο χρήστης του κατεβάζει κάποιο δωρεάν πρόγραμμα από το κάποιο ftp site. Ανεξάρτητα από τον τρόπο με τον οποίο έγινε η μόλυνση, ο ιός στη συνέχεια θα δει τη λίστα διευθύνσεων του χρήστη και θα στείλει e-mails σε όλα τα άτομα που είναι καταχωρημένα στη λίστα διευθύνσεων του χρήστη προσποιούμενος πως είναι ο χρήστης του υπολογιστή. Το e-mail που στέλνεται θα περιέχει και ένα επισυναπτόμενο αρχείο. Οι χρήστες οι οποίοι δεν θα άνοιγαν κάποιο επισυναπτόμενο αρχείο από άγνωστο αποστολέα κατά πάσα πιθανότητα θα ανοίξουν το αρχείο που προέρχεται κατά τα φαινόμενα από κάποιον που γνωρίζουν. Συνεπώς ο ιός θα μολύνει όλους τους υπολογιστές των ατόμων που θα ανοίξουν το αρχείο και θα στείλει ακόμα περισσότερα e-mails χρησιμοποιώντας το νέο κατάλογο διευθύνσεων κ.ο.κ.

Ανιχνευτές (Scanners)

Ένα scanner είναι ένα πρόγραμμα το οποίο ανιχνεύει αδυναμίες ασφάλεια σε υπολογιστικά συστήματα. Είναι λίγο αμφιλεγόμενο αν θα έπρεπε να μπει αυτό το θέμα σε μια ενότητα σχετική με επιθέσεις αφού τα προγράμματα αυτά αναπτύχθηκαν για να βοηθήσουν τους διαχειριστές συστημάτων να εντοπίσουν αδυναμίες. Ωστόσο κάποια από αυτά μπορούν να χρησιμοποιηθούν για να διερευνήσουν τρόπους εισβολής σε ένα δίκτυο.

SATAN

Πιθανότατα το πιο γνωστό scanner είναι το SATAN. Όταν κυκλοφόρησε το 1995 δημιούργησε σάλο καθώς ήταν το πρώτο πρόγραμμα το οποίο μπορούσε να εντοπίσει προβλήματα ενός δικτύου λειτουργώντας έξω από το δίκτυο. Υπήρχαν άλλοι δυο λόγοι για τους οποίους δημιουργήθηκε τόσος θόρυβος για το πρόγραμμα αυτό: ο πρώτος είναι ότι όταν εντόπιζε κάποια αδυναμία εμφάνιζε και ένα κατάλληλο και αρκετά αυστηρό μήνυμα σχετικά με τους κινδύνους της αδυναμίας αυτής και το δεύτερο ήταν το ίδιο το όνομα του προγράμματος, ήταν μια ένδειξη κακόβουλων προθέσεων εκ μέρους του εκάστοτε χρήστη του. Το SATAN δημιουργήθηκε από δυο σύμβουλους ασφαλείας, τον Dan Farmer και τον Weitse Venema, στο UNIX. Τα αρχικά SATAN σημαίνουν Security Administrator's Tool for Analyzing Networks.

Ένα **scanner** είναι ένα πρόγραμμα το οποίο ερευνά τα διάφορα στοιχεία ενός λειτουργικού συστήματος και ελέγχει αν είναι ασφαλή, για παράδειγμα μερικοί scanners για το UNIX μπορούν να ελέγχουν αν η δημοφιλής εφαρμογή *sendmail* είναι αρκετά ασφαλής για να

αποτρέψει την εισβολή; άλλοι scanners μπορούν να ελέγξουν την ανθεκτικότητα ενός ftp server, για παράδειγμα βρίσκοντας αν ένα πολύ μεγάλο password θα μπλοκάρει τον ftp server.

Τα scanners συνήθως είναι γραμμένα στο UNIX, αλλά τα τελευταία χρόνια έχουν δημιουργηθεί αντίστοιχα και για άλλα λειτουργικά συστήματα όπως τα Windows NT.

Σπάσιμο κωδικών (Password crackers)

Ένα password cracker είναι ένα πρόγραμμα το οποίο προσπαθεί να βρει το password κάποιου χρήστη ή το όνομα του χρήστη που αντιστοιχεί σε κάποια passwords που υπάρχουν αποθηκευμένα σε ένα αρχείο με passwords σε κάποιο υπολογιστή. Τα εργαλεία αυτά χρησιμοποιήθηκαν αρχικά από διαχειριστές συστημάτων ώστε να σιγουρευτούν ότι τα passwords που επέλεξαν οι χρήστες τους δεν μπορούσαν να εντοπιστούν εύκολα. Ωστόσο, χρησιμοποιήθηκαν επίσης κακόβουλα, για παράδειγμα για να αποκτήσουν πρόσβαση σε συστήματα όπου οι χρήστες είχαν εύκολα passwords όπως 'system' ή 'admin'.

Τα περισσότερα password crackers είτε προσπαθούν να ανακαλύψουν ένα password χρησιμοποιώντας μια μεγάλη λίστα λέξεων που επιλέγουν συχνά οι χρήστες ως passwords και δοκιμάζουν πολλά από αυτά είτε επιχειρούν να αποκτήσουν απευθείας πρόσβαση στο αρχείο των password.

I0phtCrack

Αυτό είναι ένα password cracker για Windows. Λειτουργεί με δυο τρόπους. Ο ένας είναι να ελέγχει τα passwords σε ένα δίκτυο χρησιμοποιώντας ένα αρχείο με passwords που έδωσε ο χρήστης. Ο δεύτερος τρόπος είναι να δοκιμάζει όλα τα δυνατά passwords χρησιμοποιώντας ένα περιορισμένο σύνολο χαρακτήρων: όλα τα γράμματα , κεφαλαία και πεζά καθώς και τα ψηφία από το 0 έως το 9.

Προγράμματα Υποκλοπής (Sniffers)

Αυτά είναι εργαλεία τα οποία υποκλέπτουν πακέτα δεδομένων τα οποία ταξιδεύουν στο δίκτυο. Υπάρχει μια νόμιμη χρήση τους από τους διαχειριστές συστημάτων καθώς μπορούν να χρησιμοποιηθούν για να εντοπίσουν αδυναμίες ενός δικτύου, για παράδειγμα μπορούν να χρησιμοποιηθούν για να εντοπίσουν σημεία πολύ έντονης κυκλοφορίας όπου μπορεί να υπάρχει πρόβλημα. Χρησιμοποιούνται επίσης από προγραμματιστές κατανεμημένων συστημάτων για να πάρουν μια ιδέα της αναμενόμενης κυκλοφορίας στο δίκτυο και να προσαρμόσουν την εφαρμογή τους σε αυτή.

Ωστόσο, έχουν συχνά επίσης χρησιμοποιηθεί για την υποκλοπή σημαντικών δεδομένων. Ένας εισβολέας μπορεί να εγκαταστήσει ένα sniffer σε ένα σημαντικό σημείο ενός δικτύου, για παράδειγμα σε μια πύλη και να διαβάσει τα μηνύματα καθώς αυτά περνάνε από αυτή. Ένας πετυχημένος sniffer μπορεί να εντοπίσει εκατοντάδες, αν όχι χιλιάδες passwords μέσα σε λίγες ώρες και να τα στείλει σε ένα απομακρυσμένο υπολογιστή από όπου κάποιος μη εξουσιοδοτημένος χρήστης θα μπορεί να τα χρησιμοποιήσει για να εισβάλει στο σύστημα.

Οι επιθέσεις με sniffer είναι παραδόξως όχι πολύ συνηθισμένες, ωστόσο όταν συμβαίνουν μπορεί να θέσουν σε κίνδυνο την ασφάλεια πολλών υπολογιστών και χρηστών. Για παράδειγμα, μια πρόσφατη επίθεση με sniffer είχε ως αποτέλεσμα 268 sites (όχι υπολογιστές αλλά sites!) να έχουν σοβαρά προβλήματα ασφάλειας.

Δούρειοι ίπποι (Trojan horses)

Ένας δούρειος ίππος είναι ένα κακόβουλο κομμάτι κώδικα το οποίο υπάρχει μέσα σε ένα κατά τα άλλα αθώο πρόγραμμα και το οποίο επιχειρεί να κάνει κάτι το οποίο ο χρήστης δεν περιμένει να κάνει. Για παράδειγμα, ένα ελεύθερης πρόσβασης πρόγραμμα το οποίο παρέχει σε ένα διαχειριστή συστημάτων πληροφορίες σχετικά με τη χρήση των αρχείων σε ένα δικτυακό σύστημα, αλλά το οποίο μετά από κάποια στιγμή υποκλέπτει πληροφορίες ή αλλάζει αρχεία είναι ένας δούρειος ίππος.

Οι δούρειοι ίπποι μπορούν να χρησιμοποιηθούν για διάφορους λόγους όπως την υποκλοπή passwords και άλλων πληροφοριών ή για να καταστρέψουν πόρους (π.χ. αρχεία) και να προκαλέσουν κατάρρευση ενός συστήματος.

Το κύριο πρόβλημα με τους δούρειους ίππους είναι ότι είναι πολύ δύσκολο να εντοπιστούν. Οι λόγοι είναι δυο: ο πρώτος είναι ότι συχνά παίρνουν τη μορφή ιδιαίτερα συνηθισμένων εργαλείων ή εργαλείων που απαιτούν την χειροκίνητη εγκατάσταση τους από το χρήστηΟ δεύτερος λόγος για τον οποίο είναι δύσκολο να εντοπιστούν είναι ότι υπάρχουν σε κάποιο υπολογιστή με τη μορφή ενός μεταφρασμένου προγράμματος το οποίο είναι δύσκολο να ελεγχθεί τι ακριβώς κάνει.

Spoofing

Αυτός είναι ένας όρος ο οποίος χρησιμοποιείται για να περιγράψει την κατάσταση κατά την οποία ένας εισβολέας χρησιμοποιεί κάποιο υπολογιστή προσποιούμενος στο σύστημα στο οποίο επιτίθεται ότι ο υπολογιστής που χρησιμοποιεί είναι κάποιος άλλος τον οποίο το σύστημα εμπιστεύεται και συνεπώς μπορεί να εκτελέσει λειτουργίες που κανονικά δεν θα επιτρεπόταν. Το spoofing δεν απαιτεί πολλές γνώσεις σχετικά με passwords και μεθόδους πιστοποίησης χρηστών όπως οι προηγούμενες μέθοδοι. Έχει σχέση μόνο με το να νομίζει το δίκτυο ότι ο υπολογιστής που χρησιμοποιεί ο εισβολέας είναι κάποιος άλλος υπολογιστής που το δίκτυο εμπιστεύεται.

Για να καταλάβουμε πως λειτουργεί το spoofing μπορούμε να δούμε μια συγκεκριμένη μορφή της τεχνικής αυτής που λέγεται **IP spoofing**. Αυτή η επίθεση χρησιμοποιεί το πρωτόκολλο TCP-IP για να παρακάμψει τις κανονικές λειτουργίες πιστοποίησης σε ένα σύστημα και γίνεται χρησιμοποιώντας έναν υπολογιστή που ισχυρίζεται πως έχει μια έμπιστη IP διεύθυνση.

Cookies

Ένα cookie είναι ένα αρχείο που τοποθετείται στον υπολογιστή ενός χρήστη από έναν browser και που συνήθως περιέχει στοιχεία συναλλαγών του χρήστη με συγκεκριμένους δικτυακούς τόπους. Για παράδειγμα, ένα cookie μπορεί να περιέχει στοιχεία για τα προϊόντα που επέλεξε και θα χρησιμοποιείται στο τέλος της συναλλαγής για να υπολογίσει το τελικό κόστος. Τέτοια cookies είναι παροδικά, ωστόσο υπάρχουν άλλα που είναι περισσότερο μόνιμα και μένουν στον δίσκο του χρήστη για πολύ καιρό. Μια συνηθισμένη χρήση τέτοιων cookies είναι να κρατάνε στοιχεία πιστωτικών καρτών για παράδειγμα ώστε να μην απαιτείται ο χρήστης να επανεισάγει τα στοιχεία του κάθε φορά που θέλει να κάνει μια συναλλαγή. Τα cookies είναι όμως απειλή για προσωπικά σας στοιχεία τα οποία πιθανόν δεν θέλετε να είναι γνωστά σε άλλους: είναι σχετικά εύκολο να μαζέψει κανείς στοιχεία σχετικά με τις συνήθειες σας, τις προτιμήσεις σας κ.λ.π. Αν αισθάνεστε άβολα με μια τέτοια κατάσταση υπάρχει απλή λύση: να απενεργοποιήσετε την επιλογή του browser σχετικά με την χρησιμοποίηση των cookie. Ωστόσο, αυτό μπορεί μερικές φορές να μην είναι βολικό καθώς πολλά sites θα απαιτούν τη χρήση των cookies.

Όταν ένας υπολογιστής ανοίγει μια σύνδεση με έναν άλλο χρησιμοποιώντας TCP-IP, ο πελάτης στέλνει ένα TCP πακέτο με έναν αρχικό ακέραιο αριθμό. Ο λαμβάνων υπολογιστής (ο διακομιστής) επιστρέφει ένα πακέτο το οποίο περιλαμβάνει έναν άλλο ακέραιο, οι αριθμοί αυτοί είναι γνωστοί ως αριθμοί ακολουθίας. Επίσης στέλνει μια επιβεβαίωση η οποία είναι ο αριθμός ακολουθίας του πελάτη συν ένα. Ο πελάτης στη συνέχεια πρέπει να επιστρέψει μια επιβεβαίωση

η οποία περιλαμβάνει τον αριθμό του ακολουθίας του διακομιστή στην ένα. Από τη στιγμή αυτή, ο πελάτης και ο διακομιστής μπαίνουν σε μια διαδικασία διαλόγου στην οποία ο πελάτης και ο διακομιστής στέλνουν πακέτα τα οποία περιέχουν αριθμούς ακολουθίας τους οποίους η άλλη πλευρά πρέπει να επιστρέψει για να πιστοποιήσει ότι είναι αυτή που ισχυρίζεται. Οι αριθμοί ακολουθίας προσδιορίζονται από έναν αλγόριθμο του TCP-IP. Το κύριο πρόβλημα για να επιτευχθεί το IP spoofing είναι ότι ο εισβολέας θα πρέπει να γνωρίζει τους αριθμούς ακολουθιών που δημιούργησε και έστειλε ο διακομιστής κατά την αρχική εγκατάσταση της επικοινωνίας: ο διακομιστής θα λαμβάνει πακέτα από έναν υπολογιστή που ισχυρίζεται ότι έχει μια IP διεύθυνση αλλά τα πακέτα που στέλνει θα μεταφέρονται από το δίκτυο στον υπολογιστή που πραγματικά έχει την διεύθυνση αυτή και συνεπώς ο εισβολέας δεν θα παίρνει τις απαντήσεις για να δει τον αριθμό ακολουθίας που πρέπει να στείλει. Επειδή ο εισβολέας πρέπει να απαντήσει με πακέτα τα οποία περιέχουν τον κατάλληλο αριθμό ακολουθίας, κάθε πακέτο το οποίο θα λαμβάνεται στον διακομιστή και δεν περιέχει τον κατάλληλο αριθμό ακολουθίας θα θεωρείται ύποπτο και θα παρουσιάζεται το κατάλληλο μήνυμα.

Για να κάνει μια επίθεση IP spoofing, ο εισβολέας πρέπει να πετύχει τα εξής:

- Ο πραγματικός υπολογιστής που θα προσποιηθείτε ότι είστε πρέπει να είναι εκτός λειτουργίας. Αυτό συνήθως επιτυγχάνεται με μια επίθεση άρνησης υπηρεσίας.
- Ο υπολογιστής που θα χρησιμοποιηθεί για την επίθεση πρέπει να πάρει την IP διεύθυνση του υπολογιστή που θα προσποιηθεί ότι είναι.
- Ο υπολογιστής του εισβολέα τότε θα πρέπει να συνδεθεί με τον διακομιστή και να ξεκινήσει έναν διάλογο προσποιούμενος ότι είναι κάποιος άλλος υπολογιστής.
- Ο υπολογιστής του εισβολέα πρέπει με κάποιο τρόπο να ανακαλύψει τον αριθμό ακολουθίας που δημιούργησε ο διακομιστής. Αυτό είναι αρκετά δύσκολο αλλά όχι ακατόρθωτο. Σε μερικά τοπικά δίκτυα μπορεί επίσης να γίνει αρκετά εύκολα.

Μερικές φορές πάντως, οι αριθμοί ακολουθίας που φτιάχνει κάποιος ευάλωτος διακομιστής μπορούν να βρεθούν απλά με δοκιμή πολλών διαφορετικών σε μια σειρά πολλών διαφορετικών προσπαθειών για σύνδεση.

Συνήθως αφού κάποιος εισβολέας καταφέρει να μπει στο σύστημα, βρίσκει ένα πιο απλό και βολικό τρόπο για να συνεχίσει τη δραστηριότητα του, αλλάζοντας κάποιο password ή κάποια ρύθμιση στον διακομιστή κ.λ.π.

Αυτή είναι μια μόνο μορφή spoofing, υπάρχουν και άλλες. Το **ARP spoofing** για παράδειγμα. Τα αρχικά ARP σημαίνουν Address Resolution Protocol. το πρωτόκολλο ARP είναι το κομμάτι του TCP-IP, που συνδέει φυσικές διευθύνσεις υπολογιστών (κάρτας δικτύου π.χ.) με IP διευθύνσεις. Το τμήμα του λειτουργικού συστήματος το οποίο αποθηκεύει τα απαραίτητα στοιχεία για να γίνεται ξ απαραίτητη μετατροπή διευθύνσεων είναι γνωστό ως **ARP cache**. Μια επίθεση ARP spoofing πραγματοποιείται μεταβάλλοντας την cache ώστε η IP διεύθυνση ενός υπολογιστή που ο διακομιστής εμπιστεύεται στην πραγματικότητα θα ισοδυναμεί με τη φυσική διεύθυνση του υπολογιστή του εισβολέα.

Άλλη μια μορφή spoofing είναι το **DNS spoofing**. Αυτό είναι μια λιγότερο σημαντική απειλή από τα δυο προηγούμενα καθώς μπορεί σχετικά εύκολα να εντοπιστεί.. Ωστόσο, κάποιες επιθέσεις αυτού του είδους εξακολουθούν να συμβαίνουν ενίοτε. Σε μια επίθεση DNS spoofing μεταβάλλονται τα στοιχεία ενός DNS server ώστε να αντιστοιχεί το συμβολικό όνομα κάποιου υπολογιστή που εμπιστεύονται οι χρήστες στην IP διεύθυνση ενός υπολογιστή που χρησιμοποιείται από το άτομο που στήνει το κόλπο. Αυτό σημαίνει ότι οι υπολογιστές που θα προσπαθούν να συνδεθούν με τον υπολογιστή που εμπιστεύονται θα συνδέονται στην πραγματικότητα με κάποιον άλλο υπολογιστή, κατά τη διάρκεια της επικοινωνίας με τον οποίο θα μπορούσαν να αντληθούν σημαντικά δεδομένα. Για παράδειγμα θα μπορούσε ο χρήστης να δώσει τον αριθμό της πιστωτικής του κάρτας νομίζοντας πως πραγματικά η άλλη πλευρά θα το χρησιμοποιήσει απλά για να φέρει εις πέρας μια επιθυμητή συναλλαγή.

Επιθέσεις βασισμένες σε κενά ασφαλείας νέων τεχνολογιών

Υπάρχουν είδη επιθέσεων που εκμεταλλεύονται κενά ασφαλείας σε νέες τεχνολογίες. Συνήθως τεχνολογίες που έχουν σχέση με εφαρμογές απομακρυσμένων εφαρμογών είναι αρκετά ευάλωτες σε κενά ασφαλείας. Υπάρχει ένας τόσο μεγάλος αριθμός σφαλμάτων ασφαλείας σε νέες τεχνολογίες που χρησιμοποιούνται στο Internet που αν αυτή η ενότητα ασχολούνταν με όλες αυτές θα είχε μέγεθος δυσανάλογο από τη σημασία της καθώς πολλά από τα λάθη αυτά έχουν ανακαλυφθεί και πολλά από αυτά διορθώθηκαν γρήγορα μετά την ανακάλυψη τους. Στην ενότητα αυτή θα επικεντρωθούμε στα προβλήματα ασφαλείας δυο συγκεκριμένων τεχνολογιών της Java και του Active X και θα δούμε παραδείγματα επιθέσεων βάσει αυτών.

Java

Όταν παρουσιάστηκε η Java η προσοχή του κόσμου εστιάστηκε στα applets. Αυτά είναι μικρά προγράμματα Java που εισάγονται σε HTML σελίδες και τα οποία εκτελούνται στον πελάτη.

Τα applets παρείχαν σημαντικές βελτιώσεις στην εμφάνιση των Web σελίδων: επέτρεπαν animations, image maps και επεξεργασία φορμών στον πελάτη. Ωστόσο, το μειονέκτημα ήταν ότι αποτελούσαν ένα μέσο με το οποίο έξυπνοι προγραμματιστές της Java θα μπορούσαν να θέσουν σε κίνδυνο την ασφάλεια ενός υπολογιστή που τρέχει applets. Πριν αναφερθούμε σε λεπτομέρειες για αυτό, πρέπει να αναφέρουμε ότι πολλά από τα αρχικά προβλήματα ασφαλείας της Java αντιμετωπίστηκαν γρήγορα με αλλαγές στον κώδικα της Java από την Sun Microsystems.

Υπήρχαν κάποια αρχικά προβλήματα με την Java τα οποία εντοπίστηκαν νωρίς από τους ερευνητές:

- Τα applets μπορούσαν να χρησιμοποιηθούν για επιθέσεις άρνησης υπηρεσιών σε συνδυασμό με συγκεκριμένους browsers.
- Κάποιος browser ήταν ευάλωτος σε επιθέσεις με applets τα οποία μπορούσαν να γράψουν αρχεία σε συστήματα που χρησιμοποιούσαν Windows 95.
- Υπήρχε κάποιο applet που προκαλούσε επανεκκίνηση των Windows 95.
- Σε κάποια έκδοση του Netscape Navigator ένα applet παγιδεύσει μια σελίδα που περιέχει κάποια φόρμα, να διαβάσει κάποια δεδομένα και να τα στείλει σε κάποιο απομακρυσμένο υπολογιστή.
- Μερικές εκδόσεις του Netscape Navigator και του Internet Explorer μπορούν να επιτρέψουν τον εντοπισμό IP διευθύνσεων σε ένα κλειστό δίκτυο από applets.

Πολλά από τα αρχικά προβλήματα της Java εξαλείφθηκαν, ωστόσο το δίδαγμα από τις αρχικές της ατέλειες ήταν πως κάθε νέα τεχνολογία θα έχει λάθη τα οποία μπορούν να θέσουν σοβαρά σε κίνδυνο την ασφάλεια.

Active X

Αυτή είναι μια τεχνολογία από την Microsoft που είναι παρόμοια με τα applets. Επειδή το Active X, όπως και τα applets της Java, είναι μια τεχνολογία η οποία αφορά

εκτελέσιμο κώδικα ενσωματωμένο σε μια ιστοσελίδα έχει και αυτό πολλά από τα προβλήματα ασφαλείας που έχουν και τα applets. Για παράδειγμα, τον Ιανουάριο του 1996 δυο Γερμανοί έφτιαξαν ένα Active X αντικείμενο το οποίο μπορούσε να μεταφέρει χρήματα μεταξύ τραπεζικών λογαριασμών.

Κεφάλαιο 3

Windows VS Linux

Αναμφίβολα η πλέον φανατική αντιπαράθεση στο χώρο της πληροφορικής έχει να κάνει με τον πόλεμο των λειτουργικών συστημάτων. Η κόντρα Windows-Linux που κλιμακώνεται τα τελευταία χρόνια, έχει αγγίξει τα όρια του «οπαδισμού» με συχνά ακραίες αντιπαραθέσεις, οι οποίες κάνουν τις αντίστοιχες παλαιότερες κόντρες Amiga-Atari και PC-Mac να μοιάζουν με αθώες διαλέξεις φιλολογικού ενδιαφέροντος.

3.1 Από το 0 έως τα… Windows

Σύμφωνα με την ιστορία, το Σεπτέμβριο του 1982 τα PCs μπαίνουν σε παραθυρικό περιβάλλον με το VisiOn της VisiCorp, το οποίο όμως «εξαφανίζεται» κάτω από το βάρος των Windows 1.0 δύο χρόνια μετά. Το τελευταίο δεν είναι παρά ένα στοιχειώδες γραφικό περιβάλλον στο παλιό καλό MS-DOS (και αρκετά παρωχημένο τόσο προς την εμφάνιση όσο και ως προς τη λειτουργία σε σχέση με το παραθυρικό OS του Macintosh). Έως το 1990, έτος κυκλοφορίας των Windows 3.0, οι χρήστες των PCs παραμένουν λίγο πολύ στην «πρωτόγονη» εποχή της command line του MS-DOS. Πλέον όμως τα πράγματα έχουν ωριμάσει, το ίδιο και το γραφικό λειτουργικό της Microsoft. Η ευρεία υποστήριξη καταρχάς της ίδιας της Microsoft κυρίως με το Office καθιστά τα Windows (3.0 και αργότερα τα 3.11 for Workgroups) στάνταρ στις οθόνες των PCs τόσο στο γραφείο όσο και στο σπίτι. Η σημερινή κατάσταση διαμορφώνεται με τα Windows XP, οπότε ουσιαστικά παύουν να αποτελούν ένα πρόγραμμα με γραφικό περιβάλλον που τρέχει πάνω στο MS-DOS. Σημαίνουν επίσης την είσοδο στο multitasking. Τα Windows από το 1995 έως σήμερα προέρχονται από δύο γραμμές: τη Win9x (Win95/98/Me) και τη WinNT (Win NT/2000/XP). Έως τα Windows NT το λειτουργικό της Microsoft λειτουργούσε ουσιαστικά ως κέλυφος (shell) στο MS-DOS, διατηρώντας τους

τεχνικούς περιορισμούς του 30χρονου συστήματος (διαχείριση μνήμης κ.α.). Τα Windows NT, με τον εντελώς νέο πυρήνα τους αρχίζουν να ξεφεύγουν από τα δεσμά του MS-DOS, κάτι που ολοκληρώνεται στη συνέχεια με τα Windows 2000 και τα Windows XP. Με την πρόσφατη εμφάνιση των Windows XP Media Center η Microsoft μπαίνει και στο χώρο της οικιακής ψυχαγωγίας έχοντας ως βάση τα Windows XP, ενώ πλέον έχουμε περάσει στην επόμενη γενιά με ονομασία Vista.

3.2 Από το 0 έως το… Linux

Πίσω από το Linux κρύβεται το παλιό, καλό Unix. Ο δημιουργός του, Linus Torvalds, προέρχεται από τη χώρα των «χιλίων λιμνών». Εργαζόμενος ως τεχνικός υπολογιστών, ο Torvalds επιχειρεί το 1991 να δημιουργήσει έναν Unix-Κλώνο στο PC του. Όχι μόνο κατάφερε να φτιάξει το εναλλακτικό OS, αλλά έσπασε τον πάγο για το ελεύθερο λογισμικό, δημιουργώντας μια νέα σχολή που αγγίζει τα όρια του… φιλοσοφικού ρεύματος. Καθώς ο πυρήνας έχει τις ρίζες του στο 1960 και σχεδιάζεται πάνω σε ένα 386 PC, το Linux γίνεται διάσημο για την λειτουργικότητά του ακόμα και σε πολύ «αδύναμα» συστήματα. Πέρα όμως από αυτά, είναι συμβατό με μία τεράστια γκάμα συστημάτων, από Z series mainframes της IBM έως και μίνι υπολογιστές με επεξεργαστές ARM. Η τελική μορφή του λειτουργικού παρουσιάζει διαφοροποιήσεις ανάλογα με τις διανομές του, σε σημεία όπως το περιβάλλον χρήσης, το ενσωματωμένο software και η διαδικασία εγκατάστασης. Ο πυρήνας παραμένει ο ίδιος, ενώ υπάρχουν εκδοχές για desktops και servers. Τα KDE και GNOME περιβάλλοντα παραπέμπουν στα Windows. Μάλιστα σε δύο περιπτώσεις (Lindows και XPde) η εμφάνιση είναι σχεδόν όμοια με αυτή των Windows.

3.3 Το «One-on-One» των λειτουργικών συστημάτων

Εγκατάσταση

Τα Windows XP απαιτούν «καθαρή» εγκατάσταση, κάτι που σημαίνει format στο σκληρό δίσκο και εγκατάσταση σε αυτόν (ή σε partition αυτού) και μόνο του λειτουργικού. Η είναι εξαιρετικά απλη και εύκολη (απλά κάποια κλικ του ποντικιού).

Η εγκατάσταση του Linux εξαρτάται από τη διανομή και συνήθως απαιτεί (όπως και στα Windows) μισή με μία ώρα. Παρόλο που στις περισσότερες διανομές η διαδικασία έχει αυτοματοποιηθεί σημαντικά, χρειάζεται κάποια εξοικείωση, καθώς δεν λείπουν οι χειροκίνητες ρυθμίσεις τη στιγμή μάλιστα που η ορολογία είναι πιο στρυφνή και απευθύνεται σε γνώστες. Η εγκατάσταση σε «καθαρό» σύστημα (χωρίς άλλο λειτουργικό) είναι σαφώς πιο εύκολη. Άλλη μια διαφοροποίηση σε σχέση με τα Windows είναι το ότι σε κάποιες εκδοχές του, το Linux τρέχει μέσα από το CD (π.χ. Knoppix). Σε αυτή την περίπτωση δεν χρειάζεται εγκατάσταση στο σκληρό δίσκο.

Σταθερότητα

Εδώ τα Windows έχουν κακή ιστορία. Σαφώς και τα Windows XP είναι θεαματικά πιο σταθερά σε σχέση με τα Windows 98, όμως η πολιτική της Microsoft και το παιχνίδι του marketing έχουν αφήσει ουκ ολίγα προβλήματα ακάλυπτα (δεν ξεχνάμε τις μπλε οθόνες οι οποίες δεν μας έχουν εγκαταλείψει τελείως). Αφενός η βιασύνη να κυκλοφορήσει γρήγορα το προϊόν και αφετέρου οι εταιρικές διαδικασίες δημιουργίας patches συχνά οδηγούν στην παρουσίαση δυσλειτουργιών, αλλά και στην καθυστέρηση της επίλυσης τους.

Αποτελεί γενικότερη πεποίθηση ότι το Linux παρουσιάζει λιγότερες δυσλειτουργίες σε σχέση με τα Windows, κάτι που σε γενικές γραμμές είναι σωστό. Το σίγουρο είναι ότι επειδή το Linux υποστηρίζεται από μια τεράστια κοινότητα (έξω από το παιχνίδι του marketing), τα όποια προβλήματα παρουσιάζουν οι διανομές του επιλύονται ταχύτερα. Το γεγονός και μόνο ότι ο κώδικας είναι ανοιχτός δίνει τη δυνατότητα σε πολλούς χρήστες να εντοπίζουν και να διορθώνουν τις δυσλειτουργίες του.

Interface

Το interface των Windows XP αποτελεί σημαντική βελτίωση σε σχέση με τα Windows 98. Δημιουργημένο για τους πιο αρχάριους είναι περισσότερο εργονομικό και «φαντεζί». Στον τομέα της εργονομίας παραμένουν κάποιες ελλείψεις (κυρίως εξαιτίας των διάσπαρτων μενού και των κρυφών δυνατοτήτων), αλλά γενικότερα η φιλικότητα είναι επαρκείς. Για τους πιο παλιούς η command line παραμένει και συχνά απαιτείται στις πιο ειδικές διαδικασίες.

Το interface είναι ένα ισχυρό όπλο του Linux. Όπως αναφέραμε έχουμε να κάνουμε με ένα σταθερό πυρήνα Unix και με πολλαπλά περιβάλλοντα. Τα δύο πιο καθιερωμένα GUI είναι το KDE και το GNOME, που λίγο-πολύ θυμίζουν τα Windows (π.χ. το κουμπί Start, ο κάδος

ανακύκλωσης κ.α.). Από την άλλη μεριά το GUI μπορεί κάλλιστα να απουσιάζει, ενώ ανάλογα με την ισχύ του μηχανήματος όπου εγκαθίσταται το Linux μπορούν να χρησιμοποιηθούν πιο «light» GUI. Φυσικά η command line διατηρεί τον πρωταρχικό ρόλο της στον έλεγχο του λειτουργικού.

Πολλαπλοί χρήστες

Έπρεπε να φτάσουν τα Windows 2000/XP για να αποκαλέσουμε τα Windows «πολυχρηστικά», καθώς μέχρι τότε ο όρος user ήταν περισσότερο σχήμα λόγου (στα Windows 98 ήταν εξαιρετικά εύκολη η παράκαμψη του παραθύρου εισόδου στο λειτουργικό σύστημα με πληκτρολόγηση password). Το σύστημα του administrator και των επιμέρους χρηστών με συγκεκριμένους κανόνες χρήσης είναι αρκετά εκλεπτυσμένο, αν και απαιτεί κάποιο χρόνο εξοικείωσης, δεν παρέχει ωστόσο απόλυτη ασφάλεια.

Κάθε διανομή Linux διαθέτει στάνταρ δυνατότητα «πολυχρησίας». Μπορεί η μέθοδος να φαίνεται «πρωτόγονη» σε σχέση με τα Windows XP, καθώς όμως ο καθορισμός των χρηστών γίνεται ουσιαστικά βάση των περιοχών του directory στις οποίες έχουν πρόσβαση, όμως το πιο σοβαρό σύστημα διαχείρισης προστατεύει τον υπολογιστή από πιθανές βλάβες ή ιούς. Μια επιπλέον διαφοροποίηση έναντι των Windows έχει να κάνει με τον καθορισμό ομάδων χρηστών στο Linux.

Ασφάλεια

Ομολογουμένως τα Windows υστερούν έναντι του Linux στο θέμα της ασφάλειας για διάφορους λόγους. Ο κυριότερος είναι ότι αφενός ο σχεδιασμός τους αφήνει ανοιχτές πολλές «πόρτες» για την εισβολή ιών, worms κ.λ.π. και αφετέρου η διείσδυσή τους είναι ακόμα μεγαλύτερη, άρα οι χρήστες τους γίνονται πιο εύκολα στόχοι hackers, crackers και συναφών… επιδρομέων. Ειδικά η έκδοση server έχει αποδειχθεί πολύ πιο ευπαθής από τις αντίστοιχες πλατφόρμες του Linux (π.χ. RedHat, Slackware, Suse). Τέλος όταν παρουσιάζονται κενά, η Microsoft δεν αντιδρά το ίδιο γρήγορα με την ενεργή παγκόσμια κοινότητα του Linux.

Στον τομέα ασφάλεια το Linux βασιλεύει, καθώς συγκρινόμενο με τα Windows μπορεί να θεωρηθεί virus free. Ο κώδικας και γενικότερα ο σχεδιασμός του είναι συμπαγείς και δύσκολα αφήνουν ανοίγματα στους κάθε λογής εισβολείς. Ας σκεφτούμε μονάχα ότι οι Apache Servers που βασίζονται στο Linux είναι περισσότεροι, ακόμη και μέσα από αυτούς προσβάλλονται πολύ περισσότερα συστήματα Windows. Υπάρχει μάλιστα η πεποίθηση ότι

ακριβώς επειδή ο κώδικας του Linux είναι ανοιχτός, δεν είναι πρόκληση για τους hackers, οι οποίοι προτιμούν τα «δύσκολα» του «κλειδωμένου» κώδικα των Windows.

Software

Ένα από τα μεγαλύτερα πλεονεκτήματα των Windows, καθώς το λειτουργικό της Microsoft τυγχάνει τεράστιας υποστήριξης. Οι διαδικασίες εγκατάστασης είναι παντού λίγο πολύ ίδιες και σε μεγάλο βαθμό πλήρως αυτοματοποιημένες, απλούστερες ακόμα και για τον πλέον αρχάριο χρήστη. Εδώ βέβαια τίθεται ακόμα ένα ζήτημα μονοπωλίου, καθώς η Microsoft έχει κατορθώσει σχεδόν σε όλους τους τομείς του software που παράγει (σουίτες Office, web browsers κ.α.) να επιβληθεί (ας μην ξεχνάμε άλλωστε, ότι ο περίφημος δικαστικός αγώνας ξεκίνησε με αφορμή της πολιτική προώθησης του Internet Explorer). Παραμένει βέβαια το μειονέκτημα του αυξημένου κόστους αγοράς, αν και η open source κοινότητα παρέχει αρκετές δελεαστικές προτάσεις και για το λειτουργικό της Microsoft.

Παρά τη ραγδαία ανάπτυξη της open source κοινότητας και την αυξανόμενη υποστήριξη του Linux από το εμπορικό λογισμικό, στον τομέα του software το Linux παραμένει υποδεέστερο έναντι των Windows. Βέβαια για μια τυπική έως και απαιτητική χρήση γραφείου, όχι μόνο δεν λείπει τίποτα από το Linux, αλλά και δεν είναι λίγες οι διανομές που παρέχουν ένα υπερεπαρκές πακέτο εφαρμογών. Το δε open source λογισμικό διατηρεί τα πλεονεκτήματα του πολύ χαμηλού έως μηδενικού κόστους απόκτησης και χρήσης, αν και στις περισσότερες περιπτώσεις τα ίδια προγράμματα παρέχονται και για Windows. Εκεί όπου χρειάζεται ακόμα δουλειά είναι η φιλικότητα κατά την εγκατάσταση, αφού εδώ σημειώνονται σημαντικές διαφοροποιήσεις ανάλογα με τη διανομή, ενώ παράλληλα η διαδικασία δεν είναι το ίδιο αυτόματη όπως στα Windows.

Hardware

Χωρίς αμφιβολία τα Windows εγγυώνται την πλέων ομαλή συνύπαρξη με τη μεγαλύτερη γκάμα περιφερειακών, απλούστατα λόγω της μακροχρόνιας επικράτησής τους που αναγκάζει τους κατασκευαστές να σχεδιάζουν drivers πρωτίστως για το συγκεκριμένο λειτουργικό σύστημα. Το σύστημα plug'n'play το οποίο καθιερώθηκε από τα Windows 95 και παραμένει μέχρι σήμερα εξυπηρετεί τον αρχάριο χρήστη, καθώς η εγκατάσταση των περισσότερων συσκευών γίνεται αυτόματα χωρίς σχεδόν την παραμικρή παρέμβαση εκ μέρους

τους. Βέβαια πρέπει να σημειωθεί ότι με το πέρασμα από τη μία έκδοση Windows στην άλλη, αλλάζουν και οι drivers, κάτι που δεν ισχύει για το Linux.

Αν και έχουν αλλάξει πολλά σε σχέση με τα προηγούμενα χρόνια, η συνεργασία του Linux με τα διάφορα περιφερειακά επέχει πολύ από το να χαρακτηριστεί απρόσκοπτη. Ειδικά σε ότι αναφορά τους μικρότερους κατασκευαστές η ενδεχόμενη απουσία υποστήριξης μόνο αμελητέα δεν μπορεί να θεωρηθεί (ευτυχώς η κοινότητα του Linux καλύπτει σε μεγάλο βαθμό τα κενά). Ακόμα όμως και με τα chipsets καρτών γραφικών παρατηρούνται προβλήματα. Τεχνολογίες όπως το WiFi και το Bluetooth, αλλά και όλη η κατηγορία των laptops, τυγχάνουν περισσότερο θεωρητικής παρά πρακτικής υποστήριξης. Το Linux παίρνει το αίμα του πίσω χάρη στην εγγενή υποστήριξη που παρέχει σε μία τεράστια γκάμα αρχιτεκτονικών (επίσης χάρη στον πυρήνα Unix).

Δικτύωση

Τα Windows έχουν κάνει άλματα σε αυτόν τον τομέα. Αρκετά φιλικά και αυτοματοποιημένα ως προς τη χρήση τους, με λειτουργικούς wizards και εκτενείς διαδικασίες ορισμού πολλαπλών χρηστών στο δίκτυο καλύπτουν κάθε ανάγκη δικτύωσης. Ειδική μνεία πρέπει να γίνει στην υποστήριξη των νεότερων προτύπων ασύρματης δικτύωσης (WiFi, Bluetooth). Σημειώνεται ότι τα Windows «βλέπουν» συστήματα Linux μέσω δικτύου.

Το Linux είναι εγγενώς… δικτυακό. Με πυρήνα το κατεξοχήν δικτυακό Unix, διαδικασίες όπως ο διαμοιρασμός αρχείων και η σχέση client-server αποτελούν αναπόσπαστο τμήμα του κώδικά του, το ίδιο και η εξ αποστάσεως διαχείριση. Όμως υπάρχει ακόμα δρόμος μέχρι την πλήρη συνεργασία με τα ασύρματα πρότυπα, η οποία παραμένει στα χαρτιά εν πολλοίς. Τέλος, μέσω του Samba, η δικτύωση με συστήματα Windows είναι αρκετά εύκολη.

Κόστος

Εδώ παρατηρείται η μεγάλη διαφοροποίηση μεταξύ των δύο αντιπάλων, καθώς τα Windows ήταν, είναι και θα είναι ένα σχετικά ακριβό λειτουργικό σύστημα (ειδικά στις εκδόσεις server). Με τη μείωση των τιμών στο hardware, το κλάσμα της τιμής του συστήματος είναι πλέον σημαντικό. Ιδιαίτερα στο εξωτερικό ολοένα αυξάνονται οι κατασκευαστές που προκειμένου να μειώσουν την τιμή του υπολογιστή επιλέγουν Linux αντί για Windows ως προεγκατεστημένο λειτουργικό.

Όσο αναφορά το Linux τα πράγματα ακολουθούν άλλη πολιτική. Στις περισσότερες περιπτώσεις μπορούμε να κατεβάσουμε ολόκληρη τη διανομή από το internet χωρίς το παραμικρό κόστος. Οι διανομές προσφέρονται εμπορικά και σε CD, με το κόστος να διατηρείται σε χαμηλά επίπεδα. Όσο αναφορά στο κατά πόσο, σε βάθος χρόνου, το κόστος χρήσης των δύο λειτουργικών διαφέρει πραγματικά, έχει δημιουργηθεί ένα πολύ σοβαρό debate, όμως τουλάχιστον όσο αναφορά στην οικιακή χρήση το πλεονέκτημα του Linux παραμένει πέρα από κάθε αμφισβήτηση. Άλλωστε έχουμε να κάνουμε με ουκ ολίγες εναλλακτικές και όχι με μία, όπως στα Windows.

3.4 Η θεωρία του «τζάμπα»

Η άποψη ότι το Linux έρχεται δωρεάν δεν είναι απόλυτα ακριβής, τουλάχιστον σε επαγγελματικό επίπεδο. Τα τελευταία χρόνια είναι σε εξέλιξη ένα σημαντικό debate αναφορικά με το κατά πόσο τελικά το Linux είναι φθηνότερο από τα Windows. Ένα debate στο πλαίσιο του οποίου δεν έχουν λείψει οι διεθνείς στατιστικές μελέτες, τα μεγάλα λόγια από δημιουργούς και κατασκευαστές και βέβαια το πανταχού παρόν παιχνίδι του marketing.

Η πιο περίφημη μελέτη προέρχεται από την IDC κατόπιν αίτησης της Microsoft και αφορά στο λειτουργικό κόστος των Windows 2000 σε 104 εταιρίες. Σύμφωνα με τα πορίσματα της μελέτης, υπό συνθήκες και σε βάθος πενταετίας, το κόστος χρήσης του λειτουργικού της Microsoft είναι μικρότερο από του μεγάλου ανταγωνιστή της. Βέβαια δεδομένου του κόστους των επαγγελματικών λύσεων της Microsoft, το γεγονός αυτό προκύπτει ξεκάθαρα από τα λειτουργικά έξοδα των τεχνικών ομάδων IT των εταιριών. Πράγματι λόγω της διάδοσης των Windows και τουλάχιστον σε εταιρικό επίπεδο, η υποστήριξη ενός φθηνού εναλλακτικού OS όπως το Linux, ενδέχεται να αποδειχθεί αρκετά δαπανηρή εξαιτίας των αναγκών περαιτέρω εκπαίδευσης και εξειδίκευσης του τεχνικού προσωπικού. Αναμφίβολα η διείσδυση του επαγγελματικού λογισμικού επηρεάζει ευθέως ανάλογα το κόστος χρήσης του, κάτι άλλωστε που αποδεικνύεται και από το χώρο του Web, όπου η επικράτηση των λύσεων του Linux (για διάφορους λόγους, ξεκινώντας από το αρχικό κόστος και φτάνοντας στην παρεχόμενη ασφάλεια), έχουν αντιστρέψει την κατάσταση, αφήνοντας τη Microsoft δεύτερη.

Κεφάλαιο 4

Σχεδιασμός του Firewall

4.1 Κύκλος ζωής του firewall

Στην ανάπτυξη λογισμικού έχει καθιερωθεί ένα μοντέλο λεγόμενο «ο κύκλος ζωής του λογισμικού». Αν και το firewall δεν είναι ακριβώς λογισμικό η διαδικασία δημιουργίας του είναι παρόμοια. Και οι δυο διαδικασίες παρουσιάζουν ίδια βήματα σε ίδια σειρά. Όπως ο κύκλος ζωής λογισμικού είναι ένας χρήσιμος οδηγός στην ανάπτυξή του, έτσι και ο κύκλος ζωής του firewall είναι ένας χρήσιμος οδηγός στην ανάπτυξη του firewall. Ο κύκλος ζωής του firewall αποτελείται από τα εξής βήματα:

- Καθορισμός απαιτήσεων
- Δικαιολόγηση
- Αρχιτεκτονικός σχεδιασμός
- Καθορισμός πολιτικής
- Υλοποίηση
- Δοκιμή
- Διαχείριση & συντήρηση

Καθορισμός απαιτήσεων

Ο καθορισμός απαιτήσεων καθορίζει τη λειτουργία του firewall. Αναλύει τις απειλές και τους κίνδυνους τους οποίους το firewall πρόκειται να μετριάσει. Ο καθορισμός απαιτήσεων δίνει έμφαση περισσότερο στο τι κάνει το firewall παρά στο πώς. Σκεφτείτε τον καθορισμό απαιτήσεων σαν ερμηνεία των στόχων του firewall.

Δικαιολόγηση

Η κατασκευή ενός firewall περιλαμβάνει την δαπάνη οργανωτικών και οικονομικών πόρων. Τα άτομα της διοίκησης τα οποίοι έχουν αναλάβει την εξουσιοδότηση για την κατασκευή του firewall πρέπει να πειστούν ότι οι απώλειές τους θα αντισταθμιστούν με τα κέρδη από την κατασκευή του firewall. Συχνά οι απειλές και τα ρίσκα τα οποία σχετίζονται με επιθέσεις υπολογιστών δεν μοιάζουν σημαντικά σε μέλη διοικητικά μέλη τα οποία μπορεί να μην είναι πρόθυμα να χρηματοδοτήσουν το project. Αντίστοιχα οι διαχειριστές δικτύου μπορεί να έχουν έλλειψη ικανοτήτων στο να πείσουν τα μέλη της διοίκησης για τη σπουδαιότητα της ασφάλειας δικτύου.

Αρχιτεκτονικός σχεδιασμός

Ο αρχιτεκτονικός σχεδιασμός του firewall αποτελείται από την απόφαση του ποιες βασικές αρχιτεκτονικές είναι κατάλληλες, οι οποίες έπειτα αν τροποποιηθούν να πληρούν τις απαιτήσεις. Ο αρχιτεκτονικός σχεδιασμός περιλαμβάνει τέσσερα βήματα.

1) Αναγνωρίζει την υποψήφια αρχιτεκτονική και τεχνολογία
2) Κατανόηση του πώς οι υποψήφιες αρχιτεκτονικές και τεχνολογίες θα λειτουργήσουν στην συγκεκριμένη περίσταση
3) Επιλογή της υποψήφιας αρχιτεκτονικής και τεχνολογίας η οποία είναι η πιο κατάλληλη
4) Βελτίωση της επιλεγμένης αρχιτεκτονικής για καλύτερη απόδοση

Καθορισμός πολιτικής

Ο καθορισμός πολιτικής περιλαμβάνει τις πολιτικές και τους κανόνες οι οποίοι θα κατευθύνουν την λειτουργία του firewall. Οι διεργασίες οι οποίες περιλαμβάνει είναι οι εξής:

1) Αναγνώριση των host οι οποίοι θα έχουν εξουσιοδότησε σε συγκεκριμένες υπηρεσίες
2) Αναγνώριση των χαρακτηριστικών της κάθε υπηρεσίας
3) Προετοιμασία εγγράφων για το πώς το firewall χειρίζεται τα δεδομένα

Στο σχεδιασμό ενός μικρού firewall ο σχεδιαστής συνήθως συνδυάζει τον σχεδιασμό πολιτικής με την πραγματοποίηση. Έτσι όταν αυτό το μέρος πραγματοποιηθεί το firewall παίρνει την τελική του μορφή. Εντούτοις είναι γενικά χρήσιμο να δημιουργούμε έγγραφα

σχεδιασμού ακόμα και για μικρά firewall. Αυτά τα έγγραφα είναι λιγότερο ογκώδη και πιο εύκολο να διαβαστούν.

Υλοποίηση firewall

Η υλοποίηση του firewall περιλαμβάνει τη μετατροπή των πολιτικών του σε μορφή τελικού firewall. Συχνά αυτή η διαδικασία περιλαμβάνει την μετάφραση του σχεδιασμού του firewall σε εντολές και συνταχτικές παραμέτρους οι οποίες είναι κατανοητές από της εκάστοτε χρησιμοποιούμενη τεχνολογία. Εντούτοις κάποια σύγχρονα firewall περιλαμβάνουν γραφικά περιβάλλοντα αντί για χρήση γλώσσας εντολών. Σε αυτές τις περιπτώσεις η υλοποίηση του firewall αποτελείται από την παραμετροποίηση του προϊόντος ώστε να εκτελεί τις καθορισμένες λειτουργίες από τον σχεδιασμό πολιτικής του firewall.

Δοκιμή

Αφού το firewall έχει υλοποιηθεί ο έλεγχος της σωστής λειτουργίας του είναι απαραίτητος. Η δοκιμή του firewall περιλαμβάνει την δημιουργία δεδομένων εγγράφων για το πώς ανταποκρίνεται το firewall σε συγκεκριμένες λειτουργίες. Με την εκτέλεση προγραμμάτων ελέγχου firewall και συγκρίνοντας έπειτα τα αποτελέσματα με αυτά των εγγράφων προσδιορίζεται η σωστή λειτουργία του firewall. Ακόμα και οι πιο προσεκτικοί άνθρωποι είναι πιθανόν να κάνουν λάθη. Ένα μικρό λάθος μπορεί να εξασφαλίσει στον επιτιθέμενο την ευκαιρία να παραβιάσει ακόμα και το πιο εξεζητημένο firewall. Έτσι δεν πρέπει να βασιζόμαστε σε firewall μέχρις ότου η λειτουργία τους δοκιμαστεί και επαληθευτεί.

Διαχείριση και συντήρηση

Τα firewall χρειάζονται συνεχή διαχείριση και συντήρηση. Πιο συγκεκριμένα η χρήση των firewall logs μπορεί να μας ενημερώσει για επιθέσεις δίνοντάς μας την ευκαιρία να βελτιώσουμε την άμυνα του δικτύου απέναντι σε καθορισμένους τύπους επιθέσεων.

4.2 Κόστος και οφέλη

Οι αποφάσεις διαχείρισης συχνά βασίζονται σε οικονομικές παραμέτρους. Στην υπόδειξη ενός προσχεδίου firewall πρέπει να καθοριστούν το κόστος και τα οφέλη με οικονομικές ορολογίες. Ένα συχνό λάθος στην αναφορά του κόστους είναι η θεώρηση μόνο των αρχικών δαπανών, δηλαδή του σχεδιασμού και της υλοποίησης του firewall. Ακόμα και το πιο απλό firewall απαιτεί κάποια διαρκή διαχείριση και συντήρηση. Για αυτόν το λόγο η εκτίμηση πρέπει να περιλαμβάνει μια πρόβλεψη και των μελλοντικών δαπανών.

4.3 Επιλογή Λογισμικού και υλικού

Ο δεύτερος στόχος του σχεδιασμού του firewall είναι η επιλογή της τεχνολογίας. Πρέπει να έχουμε στο μυαλό μας ότι η επιλογή τεχνολογιών μπορεί να είναι συμπληρωματική. Ένα μικρό δίκτυο μπορεί να πραγματοποιηθεί με τη χρήση μιας μόνο τεχνολογίας αλλά ένα μεγάλο δίκτυο μπορεί να ωφεληθεί από ένα σχεδιασμό ο οποίος μπορεί να ενσωματώνει πολλαπλές τεχνολογίες. Γενικά οι σχεδιαστές δεν επιλέγουν μια τεχνολογία αλλά ένα προϊόν που ενσωματώνει περισσότερες από μια τεχνολογίες. Σαν χαρακτηριστικά επιλογής τεχνολογιών έχουμε τα εξής:

- Κόστος
- Χαρακτηριστικά όπως NAT, VPN, Logs κτλ.
- Τεχνική υποστήριξη και τεκμηρίωση με έγγραφα
- Ευκολία χρήσης
- Σταθερότητα
- Απόδοση

Σε αυτό το κεφάλαιο αναφέρουμε τα δημοφιλέστερα προϊόντα firewall δίνοντας έμφαση στις τεχνολογίες που ενσωματώνουν. Τα προϊόντα αυτά που θα μας απασχολήσουν είναι τα παρακάτω:

- IPTables
- IPChains
- TIS Firewall Toolkit
- Firewall-1
- Firewall Υλικού

IPChains

Το RedHat Linux 6.2 και οι προηγούμενες εκδόσεις του υποστήριζαν το IPChains. Το IPChains παρέχει stateless packet filtering. Συνεπώς το IPChains firewall πρέπει να δέχεται τις εισερχόμενες συνδέσεις άσχετα με την προέλευσή τους. Έτσι τα IPChains firewall είναι σχετικά ευπαθή σε σαρώσεις που έχουν το TCP flag δηλωμένο σαν ACK. Τα firewall αυτά δεν μπορούν να εμποδίσουν χρήστες ή ύποπτο λογισμικό όπως Trojans από το να παρακολουθούνε εισερχόμενες συνδέσεις. Το IPChains υποστηρίζει μασκάρισμα και προώθηση πόρτων. Η προώθηση πόρτων είναι μια ειδική περίπτωση του NAT προορισμού στην οποία ο αριθμός του Port και όχι η διεύθυνση IP είναι το αντικείμενο προς τροποποίηση. Το IPChains περιλαμβάνει καταγραφή πακέτων (logs) με τη λειτουργία syslog.

Χαρακτηριστικά του IPChains

Χαρακτηριστικά	Περιγραφή
Τεχνολογία	Stateless Packet Filtering
Μασκάρισμα	NAI
NAT	Περιορισμένο
Logging	Syslog
Περιβάλλον Χρήστη	Γραμμή Εντολών

IPTables

Το RedHat Linux 9 και οι επόμενες εκδόσεις περιλαμβάνουν το IPTables το οποίο είναι ο διάδοχος του IPChains. Η ασφάλεια που παρέχει το IPTables είναι ανώτερη του IPChains γιατί το IPTables χρησιμοποιεί Statefull Packet Filtering. Έτσι μπορεί να μπλοκάρει

εισερχόμενα πακέτα τα οποία δεν είναι μέρη ή δεν σχετίζονται με Established συνδέσεις. Συνεπώς το IPTables αντίθετα με το IPChains μπορεί να μπλοκάρει ανεπιθύμητη κίνηση σε συγκεκριμένα port. Επιπλέον υποστηρίζει μασκάρισμα IP και NAT διεύθυνσης και προορισμού. Ένα δίκτυο προστατευμένο από το IPTables μπορεί να αντισταθείς σε ACK σαρώσεις και μπορεί να περιορίσει την ικανότητα χρηστών ή ύποπτου λογισμικού να παρακολουθεί ή να δέχεται εισερχόμενες συνδέσεις.

Χαρακτηριστικά του IPTables

Χαρακτηριστικά	Περιγραφή
Τεχνολογία	Statefull Packet Filtering
Μασκάρισμα	NAI
NAT	NAI
Logging	Syslog
Περιβάλλον Χρήστη	Γραμμή Εντολών

TIS Firewall Toolkit

Το TIS Firewall Toolkit αναπτύχθηκε στα τέλη της δεκαετίας του 90 υπό την εποπτεία του DARPA (US Defence Advanced Research Projects Agency). Το πακέτο συνεχίζει να χρησιμοποιείται και να παραμένει δημοφιλές μέχρι σήμερα. Όπως μας παραπέμπει και το όνομα είναι μια συλλογή εργαλείων για την δημιουργία firewalls. Επειδή είναι μια συλλογή εργαλείων και όχι ένα έτοιμο προς χρήση firewall η κατασκευή ενός firewall με τη χρήση του απαιτεί παραμετροποίηση και προγραμματισμό. Αντίθετα με το IPChains και το IPTables το TIS Firewall Toolkit παρέχει Proxy-Based firewalling δίνοντας μας αυξημένο επίπεδο ασφαλείας και ευκαμψίας. Το TIS Firewall Toolkit περιλαμβάνει SMPT, HTTP, FTP και πολλούς άλλους proxy και ακόμα παρέχει την δυνατότητα κατασκευής proxy για ακόμα περισσότερα πρωτόκολλα.

Χαρακτηριστικά του TIS Firewall Toolkit

Χαρακτηριστικά	Περιγραφή
Τεχνολογία	Proxy
Μασκάρισμα	NAI
NAT	NAI
Logging	Syslog
Περιβάλλον Χρήστη	Γραμμή Εντολών

CheckPoint Firewall-1

Το Firewall-1 από την εταιρία CheckPoint είναι ένα δημοφιλές εμπορικό firewall που τρέχει κάτω από πολλές πλατφόρμες περιλαμβάνοντας και το RedHat Linux. Μερικοί πολλοί ειδικοί θεωρούν το Firewall-1 το κυρίαρχο εμπορικό προϊόν firewall. Το Firewall-1 παρέχει stateful packet filtering και NAT. Επιπλέον έχει πολλές αξιοσημείωτες δυνατότητες:

- Υψηλή διαθεσιμότητα
- Γραφικό περιβάλλον
- Φιλτράρισμα περιεχομένου σε επίπεδο εφαρμογών
- Υποστήριξη proxy για πολλά πρωτόκολλα
- Υποστήριξη multimedia πρωτοκόλλων

Η υψηλή διαθεσιμότητα του Firewall σημαίνει ότι είναι σχεδιασμένο να λειτουργεί συνεχώς. Για να παρέχει υψηλή διαθεσιμότητα το Firewall-1 μπορεί να πραγματοποιηθεί με multi-host αρχιτεκτονική στην οποία ένα δευτερεύον firewall αυτόματα αναλαμβάνει τον έλεγχο όταν το πρωτεύον βρεθεί εκτός λειτουργίας. Το φιλτράρισμα περιεχομένου σε επίπεδο εφαρμογών μπορεί να πραγματοποιήσει λειτουργίες όπως μπλοκάρισμα επικίνδυνων e-mail. Επιπλέον το Firewall-1 παρέχει χρήση Proxy για πρωτόκολλα όπως RealVideo, Windows Media και H323 τα οποία χρησιμοποιούνται για VoIP, Netmeeting κτλ. Το Firewall-1 δεν χρησιμοποιεί την Syslog λειτουργία, αλλά αντί για αυτό έχει μια ιδιωτική μέθοδο καταγραφών.

Χαρακτηριστικά του Firewall-1

Χαρακτηριστικά	Περιγραφή
Τεχνολογία	Stateful Packet Filtering
Μασκάρισμα	NAI
NAT	NAI
Logging	Ιδιωτική Μέθοδος
Περιβάλλον Χρήστη	Γραφικό

Firewall Υλικού

Ένας εναλλακτικός τύπος firewall είναι αυτά που στεγάζονται σε μονάδες υλικού όπως για παράδειγμα τα Cisco PIX firewalls. Η αξιοπιστία τους MTBF (Mean Time Between Failure) είναι υψηλότερη από αυτή των υπόλοιπων firewall. Τα PIX περιλαμβάνουν Stateful Packet Filtering με περιορισμένη υποστήριξη για proxy όπως H323. Είναι διαθέσιμα σε πολλά μοντέλα έχοντας υποστήριξη για bandwidth από 10MBps έως 1GBps. Επειδή οι μονάδες PIX δεν έχουν εσωτερικούς δίσκους πρέπει να κάνουν τις καταγραφές τους σε ένας ξεχωριστό διακομιστή καταγραφών. Αυτός μπορεί να είναι ένας Syslog server ή μια μονάδα υλικού σχεδιαζόμενη από τη Cisco σαν συμπλήρωμα στο PIX.

Χαρακτηριστικά των PIX Firewalls

Χαρακτηριστικά	Περιγραφή
Τεχνολογία	Stateful Packet Filtering
Μασκάρισμα	NAI
NAT	NAI
Logging	Εξωτερικό
Περιβάλλον Χρήστη	Γραφικό και Γραμμής Εντολών

Κεφάλαιο 5

Linux Firewalls με χρήση IPTables

Η ασφάλεια του δικτύου είναι το πρώτο πράγμα που λαμβάνουμε υπόψη για την πραγματοποίηση ενός Web Site αφού οι απειλές κάθε μέρα γίνονται όλο και περισσότερες. Ένας τρόπος για την παροχή πρόσθετης ασφάλειας είναι η πραγματοποίηση ενός firewall. Λόγω του μειωμένου κόστους μπορούμε να κατασκευάσουμε ένα firewall χρησιμοποιώντας το λογισμικό **IPTables** που περιλαμβάνεται στο λειτουργικό σύστημα **Linux**.

5.1 Τι είναι το IPTables;

Πρωταρχικά το πιο δημοφιλές πακέτο firewall που έτρεχε στο Linux λεγόταν **IPChains**, αλλά είχε μία σειρά από μειονεκτήματα. Για να διορθωθούν όλα αυτά κατασκευάστηκε ένα νέο προϊόν επονομαζόμενο **IPTables** το οποίο παρέχει:

- Καλύτερη ενσωμάτωση με τον πυρήνα του Linux, παρέχοντας την δυνατότητα επιλογής modules σχεδιασμένα για βελτιωμένη ταχύτητα και αξιοπιστία.
- Λεπτομερείς επιθεώρηση πακέτων. Αυτό σημαίνει ότι το firewall κρατά ίχνος για κάθε σύνδεση που πραγματοποιείται και σε κάποιες περιπτώσεις επιθεωρεί τα περιεχόμενα των δεδομένων που ρέουν και προσπαθεί να περιμένει την επόμενη κίνηση του συγκεκριμένου πρωτοκόλλου.
- Φιλτράρισμα πακέτων βασιζόμενη στις MAC διευθύνσεις και στις καταστάσεις των flags στην επικεφαλίδα TCP. Αυτό είναι χρήσιμο στην παρεμπόδιση επιθέσεων χρησιμοποιώντας δύσμορφα πακέτα και περιορίζοντας την πρόσβαση τοπικών διακομιστών σε άλλα δίκτυα.

- Καταγραφή αρχείων εγγραφής (**logs**) που παρέχουν την επιλογή προσαρμογής σε επίπεδο λεπτομέρειας και αναφοράς.
- Καλύτερη χρήση μετάφρασης διευθύνσεων δικτύου (**NAT**).
- Υποστήριξη για διαφανείς ενσωμάτωση με Web εξυπηρετητές όπως το **Squid**.
- Χαρακτηριστικά που εμποδίζουν την πραγματοποίηση επιθέσεων τύπου **DoS**.

5.2 Εγκατάσταση του πακέτου IPTables

Πριν ξεκινήσουμε πρέπει να βεβαιωθούμε ότι είναι εγκατεστημένο το πακέτο IPTables. Το όνομα του πακέτου είναι κάτι σαν `iptables-1.2.9-1.0.i386.rpm`. Ο έλεγχος γίνεται με την εντολή **`rpm`** `-q iptables-1.2.9-1.0.i386.rpm` ή στην περίπτωση που δεν γνωρίζουμε το ακριβές όνομα του πακέτου **`rpm`** `-qa | grep -i iptables`.

5.3 Εκτέλεση του IPTables

Μπορούμε να ξεκινήσουμε να σταματήσουμε ή να επανεκκινήσουμε το iptables χρησιμοποιώντας τις εξής εντολές:

```
[root@Avatar.gr]# service iptables start
[root@Avatar.gr]# service iptables stop
[root@Avatar.gr]# service iptables restart
```

Για την έναρξη του IPTables κατά τη διαδικασία boot χρησιμοποιούμε την εντολή `chkconfig`.

```
[root@Avatar.gr]# chkconfig iptables on
```

5.4 Προσδιορίζοντας την κατάσταση του IPTables.

Μπορούμε να προσδιορίσουμε αν το IPTables τρέχει ή όχι μέσω την εντολής:

```
[root@Avatar.gr]# service iptables status
```

Ο πυρήνας θα μας δώσει ένα μήνυμα όπως το εξής:

```
[root@Avatar.gr]# service iptables restart
Firewall is stopped.
[root@Avatar.gr]#
```

5.5 Επεξεργασία πακέτων

Όλα τα πακέτα επιθεωρούνται καθώς περνάνε μέσω μιας σειράς φίλτρων (**tables**). Κάθε φίλτρο έχει ως αντικείμενο την ανάλυση της δραστηριότητας ενός συγκεκριμένου πακέτου και ελέγχεται από τη λεγόμενη αλυσίδα φιλτραρίσματος. Υπάρχουν τρεις τύποι tables συνολικά. Ο πρώτος είναι το **mangle** table το οποίο είναι υπεύθυνο για την αλλαγή της κατάστασης των bit υπηρεσίας στην επικεφαλίδα TCP. Το δεύτερο table είναι το **filter** το οποίο είναι υπεύθυνο για το φιλτράρισμα πακέτων. Περιλαμβάνει τρεις αλυσίδες στις οποίες τοποθετούνται οι κανόνες πολιτικής του firewall.

- **Forward chain**: Φιλτράρει τα πακέτα τα οποία προορίζονται για τους διακομιστές τους οποίους προστατεύει το firewall.
- **Input chain**: Φιλτράρει τα πακέτα που προορίζονται για το firewall
- **Output chain**: Φιλτράρει τα πακέτα που προέρχονται από το firewall.

Το τρίτο table είναι το **NAT** το οποίο είναι υπεύθυνο για την μετάφραση διευθύνσεων δικτύου. Περιλαμβάνει δύο αλυσίδες:

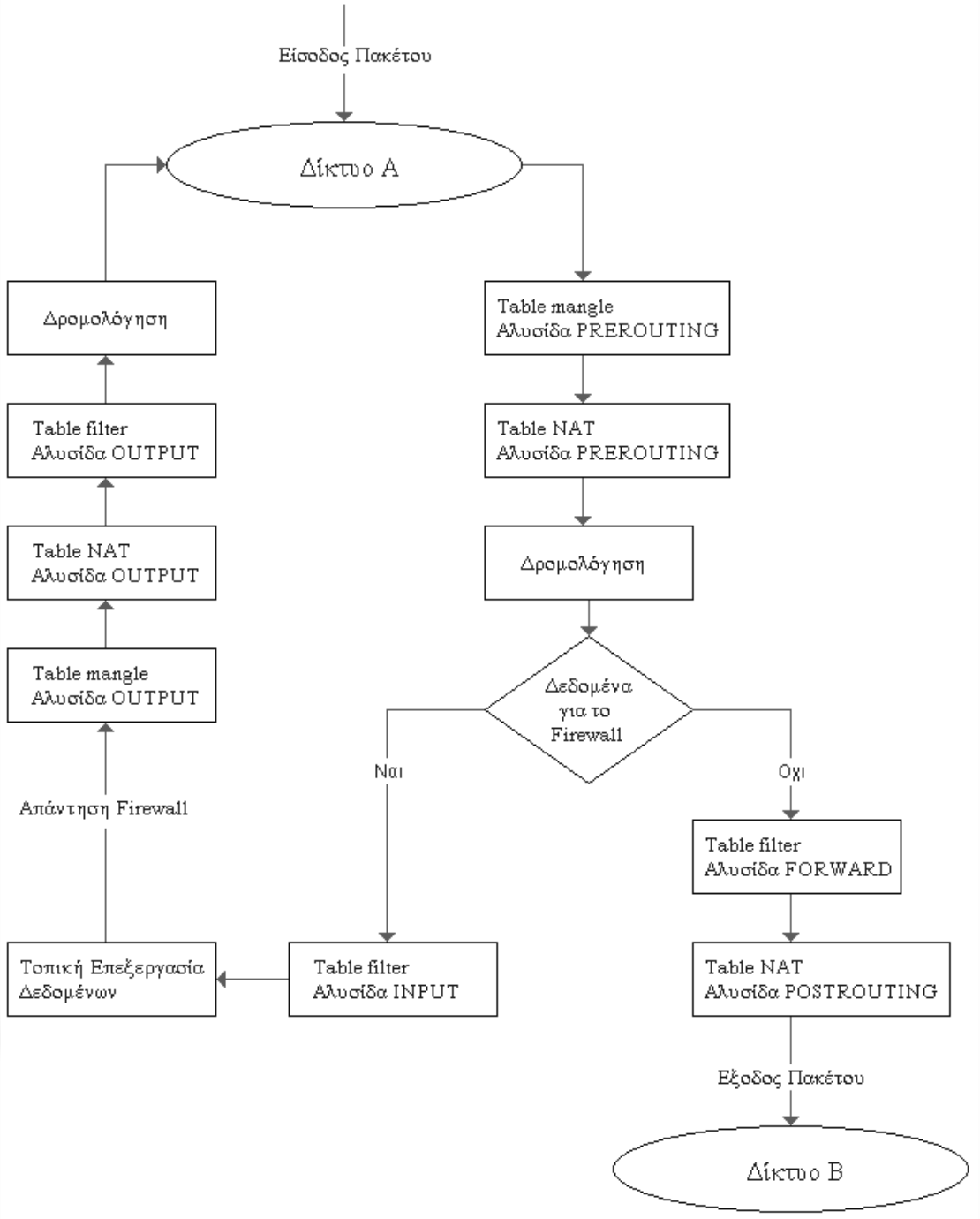
- **Prerouting chain**: Αναφέρεται σε πακέτα τα οποία η διεύθυνση προορισμού χρειάζεται να αλλάξει
- **Postrouting chain**: Αναφέρεται σε πακέτα τα οποία η διεύθυνση αφετηρίας χρειάζεται να αλλάξει

Τύπος Table	Λειτουργία Table	Τύποι Αλυσίδων	Λειτουργία Αλυσίδων
Filter	Φιλτράρισμα Πακέτων	Forward	Φιλτράρισμα πακέτων σε διακομιστές οι οποίοι είναι προσπελάσιμοι από άλλου τύπου NIC ή firewall
		Input	Φιλτράρισμα πακέτων με προορισμό το firewall
		Output	Φιλτράρισμα πακέτων προερχόμενα από το firewall
NAT	Μετάφραση Διευθύνσεων Δικτύου	Prerouting	Η μετάφραση διευθύνσεων πραγματοποιείται πριν την δρομολόγηση. Επίσης γνωστό και σαν DNAT (Destination NAT)
		Postrouting	Η μετάφραση διευθύνσεων πραγματοποιείται μετά την δρομολόγηση. Επίσης γνωστό και σαν SNAT (Source NAT)
		Output	Μετάφραση διευθύνσεων δικτύου για πακέτα που παράγονται από το firewall
Mangle	Αλλαγή Κεφαλίδας TCP	Prerouting Postrouting Output Input Forward	Μετατροπή των bits του τομέα Quality of Service του πακέτου TCP πριν πραγματοποιηθεί η δρομολόγηση

Πρέπει να ορίσουμε το table και την αλυσίδα για κάθε κανόνα του firewall που δημιουργούμε. Οι περισσότεροι κανόνες σχετίζονται με το table filtering, έτσι το IPTables υποθέτει ότι κάθε αλυσίδα χωρίς ορισμό θα είναι μέρος του filter table. Συνεπώς το filter table ορίζεται σαν προεπιλογή.

Το πακέτο πρώτα εξετάζεται από τους κανόνες του table mangle και της αλυσίδας prerouting, αν υπάρχουν. Έπειτα εξετάζεται από τους κανόνες του table NAT και της αλυσίδας prerouting για να ελεγχθεί αν το πακέτο χρειάζεται DNAT και μετά δρομολογείται. Αν το

πακέτο προορίζεται για ένα προστατευμένο δίκτυο τότε φιλτράρεται από τους κανόνες της αλυσίδας forward του table filter. Αν χρειαστεί το πακέτο υποβάλλεται σε SNAT πριν φτάσει στο δίκτυο B. Όταν ο διακομιστής προορισμού αποφασίσει να απαντήσει το πακέτο υποβάλλεται στην ίδια σειρά βημάτων.



Αν ο προορισμός του πακέτου είναι το ίδιο το firewall τότε φιλτράρεται από τους κανόνες την αλυσίδας input του table filter πριν χρησιμοποιηθεί από την εφαρμογή. Σε μερικές περιπτώσεις το firewall χρειάζεται να απαντήσει. Η απάντηση επιθεωρείται από τους κανόνες της αλυσίδας output του table mangle, αν υπάρχουν. Οι κανόνες της αλυσίδας output του table NAT καθορίζουν πότε απαιτείται μετάφραση διευθύνσεως και οι κανόνες της αλυσίδας output του table filter ελέγχονται πριν το πακέτο δρομολογηθεί πίσω στο internet.

Είναι τώρα ώρα να συζητήσουμε τον τρόπο που προσθέτουμε κανόνες σε αυτές τις αλυσίδες.

5.6 Στόχοι και Άλματα

Κάθε κανόνας του firewall επιθεωρεί το IP πακέτο και προσπαθεί να το αναγνωρίσει σαν το στόχο κάποιου τύπου λειτουργίας. Όταν ο στόχος αναγνωριστεί το πακέτο χρειάζεται να κάνει άλμα και να μεταφερθεί στη μετέπειτα διαδικασία.

Στόχος	Περιγραφή	Κοινές Επιλογές
ACCEPT	Το iptables σταματάει την περαιτέρω διαδικασία. Το πακέτο μεταφέρεται για να χρησιμοποιηθεί από την εφαρμογή η το λειτουργικό σύστημα.	
DROP	Το iptables σταματάει την περαιτέρω διαδικασία. Το πακέτο μπλοκάρεται.	
LOG	Η πληροφορίες του πακέτου στέλνονται στο δαίμονα syslog για να καταγραφούν. Το iptables συνεχίζει την διαδικασία με τον επόμενο κανόνα του table. Εφόσον δεν μπορούμε να κάνουμε εγγραφή και να απορρίψουμε το πακέτο την ίδια στιγμή είναι φυσικό να έχουμε δύο ίδιους κανόνες σε σειρά. Ο πρώτος κάνει log το πακέτο και ο δεύτερος το απορρίπτει.	`--log –prefix "string"` Λέει στο iptables να χρησιμοποιήσει σαν πρόθεμα για μήνυμα log το συγκεκριμένο string που ορίστηκε από το χρήστη. Συνήθως χρησιμοποιείται για να εξηγήσει γιατί το πακέτο απορρίφθηκε.

REJECT	Λειτουργεί σαν το DROP αλλά επιστρέφει και ένα μήνυμα σφάλματος στον host που έστειλε το πακέτο, λέγοντας ότι το πακέτο απορρίφθηκε.	`--reject -with qualifier`
		
		Ορίζει τον τύπο του μηνύματος reject που επιστέφεται. Τα qualifiers περιλαμβάνουν:
		
		`icmp-port-unreachable`
		`icmp-net-unreachable`
		`icmp-host-unreachable`
		`icmp-proto-unreachable`
		`icmp-net-prohibited`
		`icmp-host-prohibited`
		`tcp-reset`
		`echo-reply`

DNAT	Χρησιμοποιείται για να κάνει μετάφραση διευθύνσεως προορισμού επανεγράφοντας την διεύθυνση προορισμό της διεύθυνσης IP του πακέτου.	`--to -destination ipaddress` Λέει στο iptables ποιος θα είναι ο προορισμός της διεύθυνσης IP.
SNAT	Χρησιμοποιείται για να κάνει μετάφραση διευθύνσεως αφετηρίας επανεγράφοντας την διεύθυνση αφετηρίας της διεύθυνσης IP του πακέτου.	`--to -source <address> [-<address>] [:<port>-<port>]` Η αφετηρία της διεύθυνσης IP ορίζεται από το χρήστη. Καθορίζονται η διεύθυνση IP και τα Port που θα χρησιμοποιηθούν από το SNAT.
MASQUERADE	Χρησιμοποιείται για να κάνει μετάφραση της διεύθυνσης αφετηρίας.	`[--to -ports <port> [-<port>]]` Καθορίζει την εμβέλεια των ports αφετηρίας στα οποία το αρχικό port αφετηρίας τα χαρτογραφηθεί.

5.7 Βασικές εντολές και παράμετροι

Κάθε εντολή του IPTables script δεν έχει μόνο το άλμα, αλλά έχει και ένα αριθμό από εντολές και παραμέτρους οι οποίες χρησιμοποιούνται για να προσαρτήσουν τους κανόνες στις αλυσίδες που ταιριάζουν στα χαρακτηριστικά του πακέτου που έχει οριστεί, όπως η διεύθυνση IP αφετηρίας και το TCP port. Υπάρχουν επίσης επιλογές που χρησιμοποιούνται για να καθαρίσουν μια αλυσίδα ώστε να ξεκινήσουμε ξανά από την αρχή. Ο παρακάτω πίνακας δείχνει τις παρακάτω επιλογές.

Παράμετροι Εντολών	**Περιγραφή Εντολών**
`-t <table>`	Το table filter είναι προκαθορισμένο αν δεν ορίσουμε κάποιο άλλο. Τα πιθανά tables είναι: filter, NAT, mangle
`-j <target>`	Άλμα στον συγκεκριμένο στόχο της αλυσίδας όταν το πακέτο ταιριάζει στον τρέχον κανόνα.
`-A`	Προσάρτηση κανόνα στο τέλος της αλυσίδας
`-F`	Διαγράφει όλους τους κανόνες στο επιλεγμένο table
`-p <protocol-type>`	Ταιριάζει το πρωτόκολλο. Τύποι που περιλαμβάνονται ICMP, TCP, UDP, ALL
`-s <ipaddress>`	Ταιριάζει τη διεύθυνση IP αφετηρίας
`-d <ipaddress>`	Ταιριάζει τη διεύθυνση IP προορισμού
`-i <interface-name>`	Ταιριάζει την κάρτα δικτύου εισόδου στην οποία το πακέτο εισέρχεται
`-o <interface-name>`	Ταιριάζει την κάρτα δικτύου εξόδου στην οποία το πακέτο εξέρχεται

Παράδειγμα:

```
[root@Avatar.gr]# iptables -A INPUT -s 0/0 -i eth0 -d
192.168.1.1 -p TCP -j ACCEPT
```

Το IPTables παραμετροποιήθηκε για να επιτρέπει στο firewall να δεχθεί TCP πακέτα που προέρχονται από την κάρτα δικτύου *eth0*, από οποιαδήποτε IP address και προορίζονται για την διεύθυνση 192.168.1.1.

Παράμετρος	Περιγραφή
`-p tcp --sport <port>`	TCP port αφετηρίας.
`-p tcp --dport <port>`	TCP port προορισμού.
`-p tcp --syn`	Χρησιμοποιείται για να αναγνωρίσει αίτηση νέας σύνδεσης.
`-p udp --sport <port>`	UDP port αφετηρίας
`-p udp --dport <port>`	UDP port προορισμού

Παράδειγμα:

```
[root@Avatar.gr]# iptables -A FORWARD -s 0/0 -i eth0 -d
192.168.1.58 -o eth1 -p TCP \ --sport 1024:65535 --dport 80 -j
ACCEPT
```

Το IPTables παραμετροποιείται για να επιτρέψει στο firewall να δέχεται TCP πακέτα για δρομολόγηση όταν εισέρχονται στην κάρτα δικτύου *eth0*, από οποιαδήποτε IP διεύθυνση και έχουν προορισμό την διεύθυνση 192.168.1.58 η οποία είναι προσπελάσιμη μέσω της κάρτας δικτύου *eth1*. Το port αφετηρίας έχει εμβέλεια από 1024 έως 65535 και το port προορισμού είναι το port 80 (WWW/HTTP).

Επιλογή με βάση ICMP τύπο	Περιγραφή
`--icmp -type <type>`	Οι πιο χρησιμοποιούμενοι τύποι είναι τα echo-request και echo-reply

Παράδειγμα:

```
[root@Avatar.gr]# iptables -A OUTPUT -p icmp --icmp-type echo-
request -j ACCEPT
[root@Avatar.gr]# iptables -A INPUT -p icmp --icmp-type echo-
reply -j ACCEPT
```

Το IPTables παραμετροποιήθηκε για να επιτρέπει στο firewall να στέλνει ICMP echo requests (**pings**) και να δέχεται ICMP echo replies.

Θεωρήστε το παρακάτω παράδειγμα:

```
[root@Avatar.gr]# iptables -A INPUT -p icmp --icmp-type echo-
request \ -m limit --limit 1/s -i eth0 -j ACCEPT
```

Η επιλογή `limit` καθορίζει το μέγιστο αριθμό αιτήσεων που θα επιτραπούν ανά δευτερόλεπτο. Σε αυτήν την περίπτωση τα ICMP echo requests έχουν περιοριστεί να μην είναι περισσότερα από 1 ανά δευτερόλεπτο. Αν ρυθμιστεί σωστά αυτή η επιλογή μα επιτρέπει να φιλτράρουμε ασυνήθιστα μεγάλο όγκο πληροφοριών ο οποίος χαρακτηρίζει επιθέσεις τύπου *denial of service* και *internet worms*.

```
[root@Avatar.gr]# iptables -A INPUT -p tcp --syn -m limit --
limit 5/s -i eth0 -j ACCEPT
```

Μπορούμε να επεκτείνουμε την επιλογή `limit` ώστε να μειώσουμε ευαισθησία σε συγκεκριμένου τύπου επιθέσεις *denial of service*. Στη συγκεκριμένη περίπτωση η άμυνα για SYN flood επιθέσεις δημιουργήθηκε περιορίζοντας την αποδοχή TCP τεμαχίων με το SYN bit ορισμένων σε λιγότερα από 5 ανά δευτερόλεπτο.

Παράμετρος	Περιγραφή
-m multiport --sport <port, port>	Πλήθος TCP/UDP port αφετηρίας
-m multiport --dport <port, port>	Πλήθος TCP/UDP port προορισμού
-m multiport --ports <port, port>	Πλήθος TCP/UDP port
-m --state <state>	Οι πιο συχνές καταστάσεις είναι: ESTABLISHED: το πακέτο είναι μέρος της σύνδεσης που βλέπει πακέτα και από τις δύο κατευθύνσεις NEW: το πακέτο είναι η αρχή μιας νέας σύνδεσης RELATED: το πακέτο ξεκινάει μια νέα δευτερεύουσα σύνδεση INVALID: το πακέτο δεν μπόρεσε να αναγνωριστεί

Παράδειγμα:

```
[root@Avatar.gr]# iptables -A FORWARD -s 0/0 -i eth0 -d
192.168.1.58 -o eth1 -p TCP \ --sport 1024:65535 -m multiport --
dport 80,443 -j ACCEPT

[root@Avatar.gr]# iptables -A FORWARD -d 0/0 -o eth0 -s
192.168.1.58 -i eth1 -p TCP \ -m state --state ESTABLISHED -j
ACCEPT
```

Εδώ το IPTables έχει παραμετροποιηθεί για να επιτρέπει στο firewall να δέχεται TCP πακέτα για δρομολόγηση όταν εισέρχονται στην κάρτα δικτύου *eth0* από οποιαδήποτε IP διεύθυνση και με προορισμό την IP διεύθυνση 192.168.1.58 η οποία είναι προσπελάσιμη μέσω της *eth1*. Το port αφετηρίας έχει εμβέλεια από 1024 έως 65535 και τα port προορισμού είναι τα port 80 (HTTP) και το port 443 (HTTPS). Τα πακέτα επιστροφής από της 192.168.1.58 επιτρέπονται επίσης. Αντί να ορίσουμε τα port αφετηρίας και προορισμού μπορούμε πιο απλά

να επιτρέψουμε τα πακέτα τα οποία σχετίζονται με την παρούσα σύνδεση, χρησιμοποιώντας τα `-m state` και `-state ESTABLISHED`.

5.8 Αλυσίδες προσδιοριζόμενες από το χρήστη

Μπορούμε να ρυθμίσουμε το IPTables ώστε να έχουμε αλυσίδες προσδιοριζόμενες από το χρήστη. Αυτό το χαρακτηριστικό σχήμα χρησιμοποιείται για να απλοποιήσει την επεξεργασία πακέτων. Για παράδειγμα αντί να χρησιμοποιήσουμε μια αλυσίδα για όλα τα πρωτοκόλλα, μπορούμε να ορίσουμε μια αλυσίδα για κάθε τύπο πρωτοκόλλου του πακέτου και μετά να επεξεργαστούμε αυτές τις αλυσίδες στο table filter. Με άλλα λόγια μπορούμε να αντικαταστήσουμε μια μεγάλη αλυσίδα από μια η οποία μας παραπέμπει σε άλλες. Για παράδειγμα:

```
[root@Avatar.gr]# iptables -A INPUT -i eth0 -d 206.229.110.2 -j
fast-input-queue

[root@Avatar.gr]# iptables -A OUTPUT -o eth0 -s 206.229.110.2 -j
fast-output-queue

[root@Avatar.gr]# iptables -A fast-input-queue -p icmp -j icmp-
queue-in

[root@Avatar.gr]# iptables -A fast-output-queue -p icmp -j icmp-
queue-out

[root@Avatar.gr]# iptables -A icmp-queue-out -p icmp --icmp-type
echo-request \ -m state --state NEW -j ACCEPT

[root@Avatar.gr]# iptables -A icmp-queue-in -p icmp --icmp-type
echo-reply -j ACCEPT
```

Αλυσίδα	Περιγραφή
INPUT	Η κανονική input αλυσίδα του iptables
OUTPUT	Η κανονική output αλυσίδα του iptables
fast-input-queue	Αλυσίδα Input αφοσιωμένη στην αναγνώριση συγκεκριμένων πρωτοκόλλων και στη μετατόπιση πακέτων σε συγκεκριμένες αλυσίδες
Fast-output-queue	Αλυσίδα output αφοσιωμένη στην αναγνώριση συγκεκριμένων πρωτοκόλλων και στη μετατόπιση πακέτων σε συγκεκριμένες αλυσίδες
icmp-queue-out	Αλυσίδα output αφοσιωμένη στο ICMP
icmp-queue-in	Αλυσίδα input αφοσιωμένη στο ICMP

5.9 Αποθηκεύοντας το IPTables Script

Η εντολή `service iptables save` σώζει την παραμετροποίηση του IPTables στο αρχείο `/etc/sysconfig/iptables`. Όταν το σύστημα κάνει επανεκκίνηση το `iptables-restore` πρόγραμμα διαβάζει τις ρυθμίσεις και τις κάνει ενεργές. Η μορφοποίηση του αρχείου `/etc/sysconfig/iptables` είναι ελαφρός διαφορετική από αυτή του script. Η εισαγωγή των αλυσίδων είναι αυτόματη και το string `iptables` παραλείπεται. Παρακάτω έχουμε ένα παράδειγμα του `/etc/sysconfig/iptables` που επιτρέπει ICMP, IPSec συνδέσεις καταστάσεως established και εισερχόμενο SSH.

```
[root@Avatar.gr]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.2.9 on Mon Jun 9 11:00:07 2008
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [144:12748]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -p icmp -m icmp --icmp-type 255 –j ACCEPT
-A RH-Firewall-1-INPUT -p esp -j ACCEPT
-A RH-Firewall-1-INPUT -p ah -j ACCEPT
-A RH-Firewall-1-INPUT -m state –state RELATED, ESTABLISHED -j
ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW –m tcp --dport
22 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-
prohibited
COMMIT
# Completed on Mon Jun 9 11:00:07 2008
[root@Avatar.gr]#
```

Δεν είναι καλή ιδέα να επέμβουμε άμεσα σε αυτό το script γιατί δεν σώζει καθόλου σχόλια, πράγμα το οποίο μας δυσκολεύει να το κατανοήσουμε. Για αυτό το λόγο είναι καλύτερα να χρησιμοποιήσουμε ένα προσαρμοσμένο script και έπειτα χρησιμοποιώντας την εντολή `service iptables save` να το κάνουμε μόνιμο.

5.10 Φορτώνοντας τα απαιτούμενα Kernel Modules

Η εφαρμογή IPTables απαιτεί το φόρτωμα κάποιων συγκεκριμένων kernel modules τα οποία ενεργοποιούν τις λειτουργίες του. Για παράδειγμα όποτε απαιτείται χρήση NAT το module `iptable_nat` πρέπει να φορτωθεί. Το `ip_conntrack_ftp` module χρειάζεται για να υπάρχει υποστήριξη FTP και πρέπει να φορτώνεται πάντα μαζί με το `ip_conntrack` το οποίο ανιχνεύει τις καταστάσεις της TCP σύνδεσης. Το `ip_nat_ftp` module επίσης χρειάζεται να φορτωθεί για FTP servers που βρίσκονται πίσω από NAT firewalls. Δυστυχώς το `/etc/sysconfig/iptables` αρχείο δεν υποστηρίζει το φόρτωμα modules. Έτσι πρέπει να προσθέσουμε τις καταχωρίσεις μας στο αρχείο `/etc/rc.local` το οποίο τρέχει στο τέλος κάθε boot. Για παράδειγμα:

```
# File: /etc/rc.local
# Module to track the state of connections
```

```
modprobe ip_conntrack
# Load the iptables active FTP module, requires ip_conntrack
modprobe ip_conntrack_ftp
# Load iptables NAT module when required
modprobe iptable_nat
# Module required for active an FTP server using NAT
modprobe ip_nat_ftp
```

5.11 Παραδείγματα IPTables Scripts

Αυτός ο τομέας παρέχει κάποια παραδείγματα που μπορούν να βοηθήσουν στην παραμετροποίηση του iptables.

Βασική άμυνα του λειτουργικού συστήματος.

Μπορούμε να κάνουμε πολλά πράγματα πριν χρησιμοποιήσουμε το firewall script για να αυξήσουμε την ανθεκτικότητα του firewall. Για παράδειγμα το Linux περιλαμβάνει έναν αριθμό μηχανισμών προστασίας που πρέπει να ενεργοποιηθούν τροποποιώντας τις παραμέτρους του kernel στο /proc μέσω του /etc/sysctl.conf αρχείου. Παράδειγμα παραμετροποίησης:

```
# File: /etc/sysctl.conf

#-------------------------------------------------------------
# Disable routing triangulation. Respond to queries out
# the same interface, not another. Helps to maintain state
# Also protects against IP spoofing
#-------------------------------------------------------------

net/ipv4/conf/all/rp_filter = 1

#-------------------------------------------------------------
# Enable logging of packets with malformed IP addresses
```

```
#--------------------------------------------------------------

net/ipv4/conf/all/log_martians = 1

#--------------------------------------------------------------
# Disable redirects
#--------------------------------------------------------------

net/ipv4/conf/all/send_redirects = 0

#--------------------------------------------------------------
# Disable source routed packets
#--------------------------------------------------------------

net/ipv4/conf/all/accept_source_route = 0

#--------------------------------------------------------------
# Disable acceptance of ICMP redirects
#--------------------------------------------------------------

net/ipv4/conf/all/accept_redirects = 0

#--------------------------------------------------------------
# Turn on protection from Denial of Service (DoS) attacks
#--------------------------------------------------------------

net/ipv4/tcp_syncookies = 1

#--------------------------------------------------------------
# Disable responding to ping broadcasts
#--------------------------------------------------------------

net/ipv4/icmp_echo_ignore_broadcasts = 1

#--------------------------------------------------------------
```

```
# Enable IP routing.
#-----------------------------------------------------------------

net/ipv4/ip_forward = 1
```

Αυτή η παραμετροποίηση θα γίνει ενεργή στην επόμενη επανεκκίνηση του συστήματος, εκτός εάν εκτελέσουμε την εντολή `sysctl –p`

```
[root@Avatar.gr]# sysctl -p
...
...
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.tcp_syncookies = 1
net.ipv4.icmp_echo_ignore_broadcasts = 1

[root@Avatar.gr]#
```

Βασική εκκίνηση του IPTables

Η αρχική απόρριψη πακέτων (**drop**) των αλυσίδων input, forward και output του table filter είναι η καλύτερη πολιτική ασφαλείας. Επίσης δεν είναι καλή πολιτική να κάνουμε τα tables NAT και mangle να απορρίπτουν πακέτα. Αυτά τα tables εξετάζονται πριν το table filter και έτσι αν τα πακέτα δεν ταιριάζουν με τους κανόνες στα tables NAT και mangle απορριφθούν δεν θα φτάσουν στις αλυσίδες input forward και output για επεξεργασία.

```
#-----------------------------------------------------------------
# Load modules for FTP connection tracking and NAT
#-----------------------------------------------------------------
modprobe ip_conntrack
modprobe ip_nat_ftp
```

```
modprobe ip_conntrack_ftp
modprobe iptable_nat
#-----------------------------------------------------------------
# Initialize all the chains by removing all the rules
# tied to them
#-----------------------------------------------------------------
iptables --flush
iptables -t nat --flush
iptables -t mangle --flush
#-----------------------------------------------------------------
# Now that the chains have been initialized, the user defined
# chains should be deleted. We'll recreate them in the next step
#-----------------------------------------------------------------
iptables --delete-chain
iptables -t nat --delete-chain
iptables -t mangle --delete-chain
#-----------------------------------------------------------------
# If a packet doesn't match one of the built in chains, then
# the policy should be to drop it
#-----------------------------------------------------------------
iptables --policy INPUT   DROP
iptables --policy OUTPUT  DROP
iptables --policy FORWARD DROP
iptables -t nat --policy POSTROUTING ACCEPT
iptables -t nat --policy PREROUTING ACCEPT
#-----------------------------------------------------------------
# The loopback interface should accept all traffic
# Necessary for X-Windows and other socket based services
#-----------------------------------------------------------------
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

Προχωρημένη χρήση του iptables

Μπορούμε επίσης να προσθέσουμε κάποια πιο εξειδικευμένα βήματα στο script. Για παράδειγμα μπορούμε να ελέγξουμε τον όγκο της κίνησης ή να συμπεριλάβουμε ελέγχους για επιθέσεις χρησιμοποιώντας μη έγκυρα TCP flags. Το παρακάτω scrip χρησιμοποιεί πολλαπλές αλυσίδες από το χρήστη για να γίνει πιο σύντομο και πιο γρήγορο καθώς οι αλυσίδες μπορούν να προσπελαστούν με επανάληψη. Αυτό απαλείφει την ανάγκη για επανάληψη των ίδιων δηλώσεων ξανά και ξανά.

```
#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=
#
# Define networks
#
#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=

EXTERNAL_INT="eth0"            # External Internet interface
EXTERNAL_IP="97.158.253.25"    # Internet Interface IP address

#----------------------------------------------------------------
# Initialize our user-defined chains
#----------------------------------------------------------------

iptables -N valid-src
iptables -N valid-dst

#----------------------------------------------------------------
# Verify valid source and destination addresses for all packets
#----------------------------------------------------------------

iptables -A INPUT   -i $EXTERNAL_INT -j valid-src
iptables -A FORWARD -i $EXTERNAL_INT -j valid-src
iptables -A OUTPUT  -o $EXTERNAL_INT -j valid-dst
iptables -A FORWARD -o $EXTERNAL_INT -j valid-dst

#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#
```

```
#
# Source and Destination Address Sanity Checks
#
# Drop packets from networks covered in RFC 1918 (private nets)
# Drop packets from external interface IP
#
#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#=#

iptables -A valid-src -s $10.0.0.0/8     -j DROP
iptables -A valid-src -s $172.16.0.0/12  -j DROP
iptables -A valid-src -s $192.168.0.0/16 -j DROP
iptables -A valid-src -s $224.0.0.0/4    -j DROP
iptables -A valid-src -s $240.0.0.0/5    -j DROP
iptables -A valid-src -s $127.0.0.0/8    -j DROP
iptables -A valid-src -s 0.0.0.0/8       -j DROP
iptables -A valid-src -d 255.255.255.255 -j DROP
iptables -A valid-src -s 169.254.0.0/16  -j DROP
iptables -A valid-src -s $EXTERNAL_IP    -j DROP
iptables -A valid-dst -d $224.0.0.0/4    -j DROP
```

Επιτρέποντας DNS πρόσβαση στο firewall

Σχεδόν σίγουρα θα επιθυμούμε το firewall να κάνει DNS αιτήσεις στο internet. Οι παρακάτω δηλώσεις δεν εφαρμόζονται μόνο σε firewall που λειτουργούνε μόνο σαν DNS clients, αλλά και σε firewall που λειτουργούνε και σαν κανονικοί DNS διακομιστές.

```
#-------------------------------------------------------------
# Allow outbound DNS queries from the FW and the replies too
#
# - Interface eth0 is the internet interface
#
# Zone transfers use TCP and not UDP. Most home networks
# / websites using a single DNS server won't require TCP
statements
```

```
#
#--------------------------------------------------------------------

iptables -A OUTPUT -p udp -o eth0 --dport 53 --sport 1024:65535 \
-j ACCEPT

iptables -A INPUT -p udp -i eth0 --sport 53 --dport 1024:65535 \
-j ACCEPT
```

Επιτρέποντας WWW και SSH πρόσβαση στο firewall

```
#--------------------------------------------------------------------
# Allow previously established connections
# - Interface eth0 is the internet interface
#--------------------------------------------------------------------
iptables -A OUTPUT -o eth0 -m state --state ESTABLISHED,RELATED \
-j ACCEPT
#--------------------------------------------------------------------
# Allow port 80 (www) and 22 (SSH) connections to the firewall
#--------------------------------------------------------------------
iptables -A INPUT -p tcp -i eth0 --dport 22 --sport 1024:65535 \
-m state --state NEW -j ACCEPT
iptables -A INPUT -p tcp -i eth0 --dport 80 --sport 1024:65535 \
-m state --state NEW -j ACCEPT
```

Επιτρέποντας στο firewall να έχει πρόσβαση στο internet

Το παρακάτω script επιτρέπει στο χρήστη του firewall να χρησιμοποιήσει ένας web browser για να αποκτήσει πρόσβαση στο internet. Το HTTP χρησιμοποιεί το TCP port 80 και το HTTPS το port 443.

Το HTTPS (Secure HTTP) χρησιμοποιείται συνήθως για συναλλαγές πιστωτικών καρτών όπως επίσης και από διακομιστές του RedHat Linux που τρέχουν την εφαρμογή Up2Date

```
#---------------------------------------------------------------
# Allow port 80 (www) and 443 (https) connections from the
firewall
#---------------------------------------------------------------
iptables -A OUTPUT -j ACCEPT -m state \
--state NEW,ESTABLISHED,RELATED -o eth0 -p tcp \
-m multiport --dport 80,443 -m multiport --sport 1024:65535

#---------------------------------------------------------------
# Allow previously established connections
# - Interface eth0 is the internet interface
#---------------------------------------------------------------

iptables -A INPUT -j ACCEPT -m state --state
ESTABLISHED,RELATED  \
-i eth0 -p tcp
```

Αν θέλουμε όλη η κίνηση TCP που προέρχεται από το firewall να είναι αποδεχτή αφαιρούμε την παρακάτω γραμμή.

```
-m multiport --dport 80,443 -m multiport --sport 1024:65535
```

Επιτρέποντας στο τοπικό δίκτυο να αποκτήσει πρόσβαση στο firewall

Στο παρακάτω παράδειγμα η *eth1* είναι συνδεδεμένη με τοπικό δίκτυο χρησιμοποιώντας την IP 192.168.1.0. Η κίνηση δεδομένων μεταξύ αυτού του δικτύου και του firewall υποτίθεται ότι είναι έμπιστη και επιτρεπόμενη.

```
#--------------------------------------------------------------
# Allow all bidirectional traffic from your firewall to the
# protected network
# - Interface eth1 is the private network interface
#--------------------------------------------------------------

iptables -A INPUT   -j ACCEPT -p all -s 192.168.1.0/24 -i eth1
iptables -A OUTPUT  -j ACCEPT -p all -d 192.168.1.0/24 -o eth1
```

Masquerading (πολλά σε ένα NAT)

Ο όρος μασκάρισμα (masquerading) είναι μια άλλη ονομασία για αυτό που πολλοί αποκαλούν πολλά σε ένα NAT. Με άλλα λόγια η κίνηση από όλες τις συσκευές προς ένα ή περισσότερα δίκτυα θα εμφανίζεται στο firewall σαν να πηγάζει από την ίδια IP διεύθυνση.

Το IPTables χρειάζεται το module `iptables_nat` να φορτωθεί με την εντολή `modprobe` για να λειτουργήσει το μασκάρισμα. Το μασκάρισμα επίσης εξαρτάται στο λειτουργικό Linux που έχει παρεματροποιηθεί για να υποστηρίζει δρομολόγηση μεταξύ του internet και του τοπικού δικτύου. Αυτό γίνεται με την ενεργοποίηση του IP forwarding δίνοντας στο αρχείο `/proc/sys/net/ipv4/ip_forward` την τιμή 1 σε αντίθεση με την προεπιλεγμένη τιμή που είναι 0. Αφού το μασκάρισμα έχει γίνει με την αλυσίδα postrouting του NAT table πρέπει να ρυθμίσουμε το NAT table να επιτρέπει την ροή πακέτων μεταξύ των 2 καρτών δικτύου. Για να πραγματοποιηθεί αυτό χρησιμοποιούμε την αλυσίδα forward του table filter. Ειδικότερα τα πακέτα που σχετίζονται με new και established συνδέσεις θα επιτρέπεται να εξέρχονται προς το Internet, αλλά μόνο τα πακέτα που σχετίζονται με established συνδέσεις θα επιτρέπεται να εισέρχονται.

```
#--------------------------------------------------------------
# Load the NAT module
#--------------------------------------------------------------

modprobe iptable_nat
#--------------------------------------------------------------
```

```
# Enable routing by modifying the ip_forward /proc filesystem
file
#-------------------------------------------------------------

echo 1 > /proc/sys/net/ipv4/ip_forward

#-------------------------------------------------------------
# Allow masquerading
# - Interface eth0 is the internet interface
# - Interface eth1 is the private network interface
#-------------------------------------------------------------

iptables -A POSTROUTING -t nat -o eth0 -s 192.168.1.0/24 -d 0/0 \
-j MASQUERADE

#-------------------------------------------------------------
# Prior to masquerading, the packets are routed via the filter
# table's FORWARD chain.
# Allowed outbound: New, established and related connections
# Allowed inbound : Established and related connections
#-------------------------------------------------------------

iptables -A FORWARD -t filter -o eth0 -m state \
--state NEW,ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -t filter -i eth0 -m state \
--state ESTABLISHED,RELATED -j ACCEPT
```

Αν παραμετροποιηθεί το firewall ώστε να κάνει μασκάρισμα, τότε πρέπει να χρησιμοποιηθεί σαν η προεπιλεγμένη πύλη για όλους τους διακομιστές του δικτύου.

Port Forwarding τύπου NAT (DHCP DSL)

Στις περισσότερες περιπτώσεις οι χρήστε παίρνουν μία DHCP δημόσια IP διεύθυνση από τον παροχέα υπηρεσιών internet (ISP). Αν το Linux firewall είναι το μέσω διασύνδεσης με το internet και επιθυμούμε να δημιουργήσουμε ένα Web Site σε έναν από τους NAT προστατευμένους διακομιστές, τότε πρέπει να χρησιμοποιήσουμε port forwarding. Παρακάτω ο συνδυασμός της διεύθυνσης IP του firewall, της διεύθυνσης IP του απομακρυσμένου διακομιστή και το port αφετηρίας/προορισμού κίνησης μπορεί να χρησιμοποιηθεί για να αναγνωρίσει τη ρέουσα κίνηση. Όλη η κίνηση η οποία ικανοποιεί τον συνδυασμό των συγκεκριμένων παραγόντων μπορεί να προωθηθεί σε έναν διακομιστή του εσωτερικού δικτύου. Η δρομολόγηση πρέπει να επιτραπεί στο IPTables μέσω της αλυσίδας forward, και να συμπεριλάβει όλες τις new εισερχόμενες συνδέσεις από το Internet οι οποίες ταιριάζουν με το port προώθησης και επιπλέον όλα τα μελλοντικά πακέτα τα οποία σχετίζονται με established συνδέσεις από και προς τις δύο κατευθύνσεις.

```
#------------------------------------------------------------------
# Load the NAT module
#------------------------------------------------------------------

modprobe iptable_nat
#------------------------------------------------------------------
# Get the IP address of the Internet interface eth0
#------------------------------------------------------------------

external_int="eth0"
external_ip="`ifconfig $external_int | grep 'inet addr' | \
awk '{print $2}' | sed -e 's/.*://'`"
#------------------------------------------------------------------
# Enable routing by modifying the ip_forward /proc filesystem
file
#------------------------------------------------------------------

echo 1 > /proc/sys/net/ipv4/ip_forward

#------------------------------------------------------------------
```

```
# Allow port forwarding for traffic destined to port 80 of the
# firewall's IP address to be forwarded to port 8080 on server
# 192.168.1.200
#
# - Interface eth0 is the internet interface
# - Interface eth1 is the private network interface
#-----------------------------------------------------------------

iptables -t nat -A PREROUTING -p tcp -i eth0 -d $external_ip \
--dport 80 --sport 1024:65535 -j DNAT --to 192.168.1.200:8080

#-----------------------------------------------------------------
# After DNAT, the packets are routed via the filter table's
# FORWARD chain.
# Connections on port 80 to the target machine on the private
# network must be allowed.
#-----------------------------------------------------------------

iptables -A FORWARD -p tcp -i eth0 -o eth1 -d 192.168.1.200 \
--dport 8080 --sport 1024:65535 -m state --state NEW -j ACCEPT

iptables -A FORWARD -t filter -o eth0 -m state \
--state NEW,ESTABLISHED,RELATED -j ACCEPT

iptables -A FORWARD -t filter -i eth0 -m state \
--state ESTABLISHED,RELATED -j ACCEPT
```

Στατικό NAT

Στο παρακάτω παράδειγμα η κίνηση προς μία δημόσια IP διεύθυνση, όχι απλά προς ένα συγκεκριμένο port, μεταφράζεται σε έναν διακομιστή του προστατευμένου υποδικτύου. Επειδή το firewall έχει περισσότερες από μια IP διευθύνσεις δεν συνίσταται το μασκάρισμα. Αντί για αυτό χρησιμοποιώντας το SNAT ορίζουμε την εναλλακτική IP διεύθυνση η οποία θα

χρησιμοποιηθεί για συνδέσεις οι οποίες δημιουργούνται από όλους τους διακομιστές του προστατευμένου δικτύου.

Αν και το table NAT μεταφράζει όλη την κίνηση προς τους διακομιστές μόνο οι συνδέσεις στα port 80, 443 και 22 επιτρέπονται μέσω της αλυσίδας forward. Επίσης σημειώνουμε ότι πρέπει να χρησιμοποιήσουμε την επιλογή `-m multiport` *όποτε χρειαζόμαστε να ταιριάξουμε πολλαπλά μη συνεχή port.*

Το παρακάτω παράδειγμα:

- Χρησιμοποιεί ένα «1 προς 1» NAT ώστε ο διακομιστής 192.168.1.100 του τοπικού δικτύου να εμφανίζεται στο internet με την IP διεύθυνση 97.158.253.29
- Δημιουργεί ένα «πολλά σε ένα» NAT για την διεύθυνση 192.168.1.0 του τοπικού δικτύου για την οποία όλοι οι διακομιστές εμφανίζονται στο internet με την IP διεύθυνση 97.158.253.29

```
#------------------------------------------------------------------
# Load the NAT module
#------------------------------------------------------------------

modprobe iptable_nat

#------------------------------------------------------------------
# Enable routing by modifying the ip_forward /proc filesystem
file
#------------------------------------------------------------------

echo 1 > /proc/sys/net/ipv4/ip_forward

#------------------------------------------------------------------
# NAT ALL traffic:
#
# TO:              FROM:           MAP TO SERVER:
# 97.158.253.26    Anywhere         192.168.1.100 (1:1 NAT -
Inbound)
```

```
# Anywhere        192.168.1.100   97.158.253.26 (1:1 NAT -
Outbound)
# Anywhere        192.168.1.0/24  97.158.253.29 (FW IP)
#
# SNAT is used to NAT all other outbound connections initiated
# from the protected network to appear to come from
# IP address 97.158.253.29
#
# POSTROUTING:
#   NATs source IP addresses. Frequently used to NAT connections
from
#   your home network to the Internet
#
# PREROUTING:
#   NATs destination IP addresses. Frequently used to NAT
#   connections from the Internet to your home network
#
# - Interface eth0 is the internet interface
# - Interface eth1 is the private network interface
#-----------------------------------------------------------------

# PREROUTING statements for 1:1 NAT
# (Connections originating from the Internet)
iptables -t nat -A PREROUTING -d 97.158.253.26 -i eth0 \
     -j DNAT --to-destination 192.168.1.100
# POSTROUTING statements for 1:1 NAT
# (Connections originating from the home network servers)

iptables -t nat -A POSTROUTING -s 192.168.1.100 -o eth0 \
-j SNAT --to-source 97.158.253.26

# POSTROUTING statements for Many:1 NAT
# (Connections originating from the entire home network)

iptables -t nat -A POSTROUTING -s 192.168.1.0/24 \
```

```
        -j SNAT -o eth0 --to-source 97.158.253.29

# Allow forwarding to each of the servers configured for 1:1 NAT
# for connections originating from the Internet.

iptables -A FORWARD -p tcp -i eth0 -o eth1 -d 192.168.1.100 \
-m multiport --dport 80,443,22 \
-m state --state NEW -j ACCEPT

# Allow forwarding for all New and Established SNAT connections
# originating on the home network AND already established
# DNAT connections

iptables -A FORWARD -t filter -o eth0 -m state \
--state NEW,ESTABLISHED,RELATED -j ACCEPT

# Allow forwarding for all 1:1 NAT connections originating on
# the Internet that have already passed through the NEW
forwarding
# statements above

iptables -A FORWARD -t filter -i eth0 -m state \
--state ESTABLISHED,RELATED -j ACCEPT
```

5.12 Troubleshooting IPTables

Ένας μεγάλος αριθμός από εργαλεία υπάρχει στην διάθεσή μας για τον έλεγχο και τη διόρθωση των προβλημάτων του IPTables. Μια από τις καλύτερες μεθόδους είναι η καταγραφή (log) όλων των πακέτων που απορρίφθηκαν (drop) στο αρχείο `/var/log/messages`.

Έλεγχος των εγγραφών του firewall

Μπορούμε να ανιχνεύσουμε τα πακέτα που περνάνε μέσα από τους κανόνες του IPTables χρησιμοποιώντας το Target LOG. Πρέπει να έχουμε υπόψη μας ότι το LOG target:

- Καταγράφει όλη την κίνηση που ταιριάζει στους κανόνες του IPTables που τοποθετήθηκε.
- Δημιουργεί αυτόματα μια καταχώριση στο αρχείο /var/log/messages και έπειτα εκτελεί τον επόμενο κανόνα.

```
#---------------------------------------------------------------
# Log and drop all other packets to file /var/log/messages
#---------------------------------------------------------------

iptables -A OUTPUT -j LOG
iptables -A INPUT -j LOG
iptables -A FORWARD -j LOG

iptables -A OUTPUT -j DROP
iptables -A INPUT -j DROP
iptables -A FORWARD -j DROP
```

Εδώ παραθέτουμε μερικά παραδείγματα της εξόδου του αρχείου:

- Το firewall αρνείται να απαντήσει σε DNS αιτήσεις (UDP port 53) που προορίζονται για το διακομιστή 192.168.1.102 του τοπικού δικτύου.

```
Feb 23 20:33:50 Avatar kernel: IN=wlan0 OUT=
MAC=00:06:25:09:69:80:00:a0:c5:e1:3e:88:08:00 SRC=192.42.93.30
DST=192.168.1.102 LEN=220 TOS=0x00 PREC=0x00 TTL=54 ID=30485
PROTO=UDP SPT=53 DPT=32820 LEN=200
```

- Το firewall αρνείται κίνηση Windows NetBIOS (UDP port 138)

```
Feb 23 20:43:08 Avatar kernel: IN=wlan0 OUT=
MAC=ff:ff:ff:ff:ff:ff:00:06:25:09:6a:b5:08:00 SRC=192.168.1.100
DST=192.168.1.255 LEN=241 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF
PROTO=UDP SPT=138 DPT=138 LEN=221
```

- Το firewall αρνείται το πρωτόκολλο Network Time (NTP UDP port 123)

```
Feb 23 20:58:48 Hnet kernel: IN= OUT=wlan0 SRC=192.168.1.102
DST=207.200.81.113 LEN=76 TOS=0x10 PREC=0x00 TTL=64 ID=0 DF
PROTO=UDP SPT=123 DPT=123 LEN=56
```

Κεφάλαιο 6

Παραδείγματα Firewall

Παρακάτω παραθέτονται δύο firewall scripts. Το πρώτο είναι ένα single host firewall για ένα διακομιστή ο οποίος παρέχει web υπηρεσίες και το δεύτερο είναι ένα Network firewall για προστασία ενός δικτύου πολλαπλών διακομιστών ενσωματώνοντας τεχνολογία NAT. Με τις κατάλληλες μετατροπές και τροποποιήσεις μπορούν να καλύψουν τις ανάγκες των περισσοτέρων χρηστών.

6.1 Παράδειγμα Single Host Firewall

```
#!/bin/sh
#
#--------------------------------------------------------------------------
# Παράδειγμα single host firewall
#
# Copyright © 2008 Δημήτρης Μαϊστρέλλης
#
#--------------------------------------------------------------------------
#
#
#
#--------------------------------------------------------------------------
# Προσδιορισμοί
#--------------------------------------------------------------------------
# Ip διεύθυνση του Firewall
IP="xxx.xxx.xxx.xxx"

# Λίστα παρεχόμενων υπηρεσιών
PING=""
SSH="0.0.0.0/0"
WWW="0.0.0.0/0"

# Περιορισμοί
LOGOPT="--log-level=3 -m limit --limit 1/minute --limit-burst 10"
SYNOPT="-m limit --limit 5/second --limit-burst 10"

# Οι παρακάτω προσδιορισμοί δεν χρειάζονται τροποποιήσεις
BADIP="$extbase $extbcast $intbase $intbcast 0.0.0.0/8 10.0.0.0/8
      127.0.0.0/8 192.0.2.0/24 255.255.255.255"
```

```
SHUNIP=""
LO="127.0.0.1"
SSH="$SSH $LO"
WWW="$WWW $LO"
ipt=/sbin/iptables


#-------------------------------------------------------------------------------
# Σβήσιμο προϋπάρχουσων κανόνων
#-------------------------------------------------------------------------------

if [ ! -x $iptables ]
then
      die "firewall: can't execute $iptables"
fi

$ipt -P INPUT DROP            #set default policy to drop
$ipt -P OUTPUT DROP
$ipt -P FORWARD DROP
$ipt -F                       #Flush all chains
$ipt -X                       #Delete all chains

for table in filter nat mangle
do
      $ipt -t $table -F       #Delete the table's rules
      $ipt -t $table -X       #Delete the table's chains
      $ipt -t $table -Z       #Delete the table's counters
done


#-------------------------------------------------------------------------------
# Αλυσίδα καταγραφών
#-------------------------------------------------------------------------------

$ipt -N LDROP
$ipt -A LDROP   -j LOG --log-prefix "IPT Drop:    "$LOGOPT
$ipt -A LDROP                                    -j DROP

$ipt -N LBADIP
$ipt -A LBADIP  -p tcp --dport 137:139                          -j DROP
$ipt -A LBADIP  -p udp --dport 137:139                       -j DROP
$ipt -A LBADIP  -j LOG --log-prefix "IPT Bad:     "$LOGOPT
$ipt -A LBADIP                                   -j DROP

$ipt -N LSHUN
$ipt -A LSHUN   -j LOG --log-prefix "IPT Shun:    "$LOGOPT
$ipt -A LSHUN                                    -j DROP

$ipt -N LFLOOD
$ipt -A LFLOOD   -j LOG --log-prefix "IPT Flood:  "$LOGOPT
$ipt -A LFLOOD                                   -j DROP

$ipt -N LFLAGS
$ipt -A LFLAGS   -j LOG --log-prefix "IPT Flags:  "$LOGOPT
$ipt -A LFLAGS                                   -j DROP

#-------------------------------------------------------------------------------
```

```
# Bad IPs
#------------------------------------------------------------------------------


$ipt -N BADIP
for ip in $BADIP; do
      $ipt -A BADIP  -s $ip   -j LBADIP
      $ipt -A BADIP  -d $ip   -j LBADIP
done

#------------------------------------------------------------------------------
# Απαγορευμένοι Hosts
#------------------------------------------------------------------------------

$ipt -N SHUN
for ip in $SHUNIP; do
      $ipt -A SHUN  -s $ip    -j LSHUN
      $ipt -A SHUN  -d $ip    -j LSHUN
done



#------------------------------------------------------------------------------
# Syn Flood Προστασία
#------------------------------------------------------------------------------

$ipt -N FLOOD
$ipt -A FLOOD $SYNOPT   -j RETURN
$ipt -A FLOOD             -j LFLOOD


#------------------------------------------------------------------------------
# TCP flag επικύρωση
#------------------------------------------------------------------------------

$ipt -N TCP_FLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags ack,fin fin            -j LFLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags ack,psh psh              -j LFLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags ack,urg urg            -j LFLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags fin,rst fin,rst          -j LFLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags syn,fin syn,fin          -j LFLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags syn,rst syn,rst        -j LFLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags all all          -j LFLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags all none               -j LFLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags all fin,psh,urg        -j LFLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags all syn,fin,psh,urg    -j LFLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags all syn,rst,ack,fin,urg  -j LFLAGS


#------------------------------------------------------------------------------
# Εισερχόμενα TCP/UDP datagrams
#------------------------------------------------------------------------------
$ipt -N IN
$ipt -A IN -m state -- state INVALID            -j DROP
$ipt -A IN -p tcp --syn                    -j FLOOD
$ipt -A IN -p tvp                          -j FLAGS
```

```
$ipt -A IN -m state --state ESTABLISHED,RELATED  -j ACCEPT
$ipt -A IN -s $IP                              -j DROP

# Αποδοχή εισερχόμενων νέων συνδέσεων

for sip in $ssh; do
      $ipt -A IN -p tcp -s $sip --dport 22 -m state --state NEW -j ACCEPT
done

for sip in $WWW; do
      $ipt -A IN -p tcp -s $sip --dport 80 -m state --state NEW -J ACCEPT
done

# Απόρριψη AUTH αιτήσεων

$ipt -A IN -p tcp --dport 113 -j REJECT --reject-with tcp-reset

# Πρόσθετοι κανόνες για εξουσιοδοτημένη κίνηση μπορούν να προστεθούν εδώ
# Κάθε άλλη μορφή μη εξουσιοδοτημένης κίνησης θα απορρίπτεται και θα
# καταγράφεται

#-------------------------------------------------------------------------------
# Εξερχόμενα TCP/UDP datagrams
#-------------------------------------------------------------------------------

$ipt -N OUT
$ipt -A OUT -p tcp -j FLAGS
$ipt -A OUT -s ! $IP -j DROP
$ipt -A OUT -m state --state ESTABLISHED, RELATED -j ACCEPT

# Αυτό το firewall έχει ρυθμιστεί για να εμποδίζει εξερχόμενες συνδέσεις.
# Για επιτρέψετε κάθε έξοδο η οποία δεν μπλοκάρετε ρητά κάντε uncomment την
# παρακάτω γραμμή

# $ipt -A OUT -m state --state NEW -j ACCEPT


# Αποδοχή εισερχόμενων νέων συνδέσεων

$ipt - A OUT  -m state --state NEW  -p tcp --dport 21   -j ACCEPT # ftp
$ipt - A OUT  -m state --state NEW  -p tcp --dport 22   -j ACCEPT # ssh
$ipt - A OUT  -m state --state NEW  -p tcp --dport 25   -j ACCEPT # smtp
$ipt - A OUT  -m state --state NEW  -p tcp --dport 43   -j ACCEPT # whois
$ipt - A OUT  -m state --state NEW  -p tcp --dport 53   -j ACCEPT # domain
$ipt - A OUT  -m state --state NEW  -p tcp --dport 80   -j ACCEPT # http
$ipt - A OUT  -m state --state NEW  -p tcp --dport 443  -j ACCEPT # https
$ipt - A OUT  -m state --state NEW  -p tcp --dport 873  -j ACCEPT # rsync

$ipt - A OUT  -m state --state NEW  -p udp --dport 53   -j ACCEPT # domain

# Πρόσθετοι κανόνες για εξουσιοδοτημένη κίνηση μπορούν να προστεθούν εδώ
# Κάθε άλλη μορφή μη εξουσιοδοτημένης κίνησης θα απορρίπτεται και θα
# καταγράφεται


#-------------------------------------------------------------------------------
# Εισερχόμενα ICMP
```

```
#------------------------------------------------------------------------


$ipt -N IN_ICMP
for sip in $PING; do
       $ipt -A IN_ICMP  -p icmp --icmp-type echo-request -s $sip  -j ACCEPT
       $ipt -A IN_ICMP  -p icmp --icmp-type echo-reply -s $sip    -j ACCEPT
done

$ipt -A IN_ICMP  -p icmp --icmp-type destination-unreachable  -j ACCEPT
$ipt -A IN_ICMP  -p icmp --icmp-type source-quench          -j ACCEPT
$ipt -A IN_ICMP  -p icmp --icmp-type time-exceeded          -j ACCEPT
$ipt -A IN_ICMP  -p icmp --icmp-type parameter-problem        -j ACCEPT


#------------------------------------------------------------------------
# Εξερχόμενα ICMP
#------------------------------------------------------------------------

$ipt -N OUT_ICMP
for dip in $PING; do
       $ipt -A OUT_ICMP   -p icmp --icmp-type echo-request -s $dip  -j ACCEPT
       $ipt -A OUT_ICMP   -p icmp --icmp-type echo-reply -s $dip    -j ACCEPT
done

$ipt -A OUT_ICMP  -p icmp --icmp-type destination-unreachable  -j ACCEPT
$ipt -A OUT_ICMP  -p icmp --icmp-type source-quench          -j ACCEPT
$ipt -A OUT_ICMP  -p icmp --icmp-type parameter-problem        -j ACCEPT
$ipt -A OUT_ICMP  -p icmp --icmp-type fragmentation-needed     -j ACCEPT

#------------------------------------------------------------------------
# Κανόνες για built-in αλυσίδες
#------------------------------------------------------------------------

$ipt -A INPUT  -i lo          -j ACCEPT
$ipt -A INPUT                 -j BADIP
$ipt -A INPUT                 -j SHUN
$ipt -A INPUT  -p ! ICMP      -J IN
$ipt -A INPUT  -p   ICMP      -J IN_ICMP
$ipt -A INPUT                 -j LDROP

$ipt -A OUTPUT -o lo          -j ACCEPT
$ipt -A OUTPUT                -j BADIP
$ipt -A OUTPUT                -j SHUN
$ipt -A OUTPUT -p ! ICMP      -J OUT
$ipt -A OUTPUT -p   ICMP      -J OUT_ICMP
$ipt -A OUTPUT                -j LDROP
```

6.2 Παράδειγμα δικτυακού Firewall

```
#------------------------------------------------------------------------
# Δικτυακό Firewall
#
# Copyright © 2008 Δημήτρης Μαϊστρέλλης
#
#------------------------------------------------------------------------
#!/bin/sh
#
# Απαιτήσεις
# - sysctls in /etc/sysctl.conf
# - modprobs in /etc/rc./rc.local
#------------------------------------------------------------------------
# προσδιορισμοί
#------------------------------------------------------------------------

# Αλλάξτε τους παρακάτω προσδιορισμούς ώστε να ταιριάζουν στην αρχιτεκτονική του
# δικτύου σας

extdev=eth0
extip="143.233.176.200"
extbase="143.233.176.201"
extbcast="143.233.176.255"
extgate="143.233.176.254

intdev=eth1
intip="10.0.0.1"
intbase="10.0.0.0"
intbcast=10.0.0.255"
intent="10.0.0.0/8"

# Διευθύνσεις IP των hosts και δικτύων που επιτρέπεται να κάνουν ping στο
# firewall
Ping="143.233.176.100 $intent"


# Διευθύνσεις IP των hosts και δικτύων που επιτρέπεται να κάνουν SSH στο
# firewall
ssh="143.233.0.0"

# Διευθύνσεις IP του κάθε δημόσιου διακομιστή του εσωτερικού δικτύου
smtpip="10.0.0.2"
dnspip="10.0.0.2"
httpip="10.0.0.2"
popip="10.0.0.2"
authip="10.0.0.2"
imapip="10.0.0.2"

# IP addresses of hosts and networks not to be communicated with
SHUN""

# The following assignments should not generally need to be changed
BADIP="$extbase $extbcast $intbase $intbcast 0.0.0.0/8 10.0.0.0/8
      127.0.0.0/8 192.0.2.0/24 255.255.255.255"
```

```
ipt=/sbin/iptables
LOGOPT="--log-level=3 -m limit --limit 3/minute --limit-burst 3"
SYNOPT="-m limit --limit 5/second --limit-burst 10"


#-----------------------------------------------------------------------------
# Σβήσιμο προϋπάρχουσων κανόνων
#-----------------------------------------------------------------------------

if [ ! -x $iptables ]
then
      die "firewall: can't execute $iptables"
fi

$ipt -P INPUT DROP            #set default policy to drop
$ipt -P OUTPUT DROP
$ipt -P FORWARD DROP
$ipt -F                       #Flush all chains
$ipt -X                       #Delete all chains

for table in filter nat mangle
do
      $ipt -t $table -F       #Delete the table's rules
      $ipt -t $table -X       #Delete the table's chains
      $ipt -t $table -Z       #Delete the table's counters
done

#-----------------------------------------------------------------------------
# Bad TCP flags
#-----------------------------------------------------------------------------

$ipt -N BADFLAGS
$ipt -A BADFLAGS -j log --log-prefix "ipt badflags: "$logopt
$ipt -A BADFLAGS -j drop

#-----------------------------------------------------------------------------
# TCP flag επικύρωση
#-----------------------------------------------------------------------------

$ipt -N TCP_FLAGS
$ipt -A TCP_FLAGS -p  --tcp-flags ack,fin fin            -j badflags
$ipt -A TCP_FLAGS -p  --tcp-flags ack,psh psh              -j badflags
$ipt -A TCP_FLAGS -p  --tcp-flags ack,urg urg            -j badflags
$ipt -A TCP_FLAGS -p  --tcp-flags fin,rst fin,rst          -j badflags
$ipt -A TCP_FLAGS -p  --tcp-flags syn,fin syn,fin          -j badflags
$ipt -A TCP_FLAGS -p  --tcp-flags syn,rst syn,rst        -j badflags
$ipt -A TCP_FLAGS -p  --tcp-flags all all          -j badflags
$ipt -A TCP_FLAGS -p  --tcp-flags all none               -j badflags
$ipt -A TCP_FLAGS -p  --tcp-flags all fin,psh,urg        -j badflags
$ipt -A TCP_FLAGS -p  --tcp-flags all syn,fin,psh,urg    -j badflags
$ipt -A TCP_FLAGS -p  --tcp-flags all syn,rst,ack,fin,urg  -j badflags

#-----------------------------------------------------------------------------
# SYN Flood προστασία
#-----------------------------------------------------------------------------
```

```
$ipt -N SYNFLOOD
$ipt -A SYNFLOOD -p tcp  --syn $SYNOPT -j RETURN
$ipt -A SYNFLOOD -p !tcp            -j RETURN
$ipt -A SYNFLOOD -p tcp  ! --syn       -j RETURN
$ipt -A SYNFLOOD -j LOG --log-prefix "ipt syn_flood: " $LOGOPT
$ipt -A SYNFLOOD -j DROP

#------------------------------------------------------------------------------
# Bad IP Αλυσίδα
#------------------------------------------------------------------------------

$ipt -N BAD_IP
$ipt -A BAD_IP -j LOG --log-prefix "ipt bad_ip: " $LOGOPT
$ipt -A BAD_IP -j DROP

#------------------------------------------------------------------------------
# Απαγορευμένοι Hosts
#------------------------------------------------------------------------------

$ipt -N SHUN

for ip in $SHUN; do
      $ipt -A SHUN -s $ip -j BAD_IP
      $ipt -A SHUN -d $ip -j BAD_IP
done


#------------------------------------------------------------------------------
# Εισερχόμενα IP Checks
#------------------------------------------------------------------------------

$ipt -N IN_IP_CHECK
for sip in $BADSIP; do
      $ipt -A IN_IP_CHECK -s $sip -j BAD_IP
done

$ipt -A IN_IP_CHECK -i $extdev -s $extip -j BAD_IP
$ipt -A IN_IP_CHECK -i $extdev -s $intnet -j BAD_IP
$ipt -A IN_IP_CHECK -i $intdev -s $extip -j BAD_IP

#------------------------------------------------------------------------------
# Εξερχόμενα IP Checks
#------------------------------------------------------------------------------

$ipt -N OUT_IP_CHECK
for dip in $BADSIP
do
      $ipt -A OUT_IP_CHECK -s $dip -j BAD_IP
done

$ipt -A OUT_IP_CHECK -i $extdev -s $extip -j RETURN
$ipt -A OUT_IP_CHECK -i $intdev -s $intip -j RETURN
$ipt -A OUT_IP_CHECK -ij BAD_IP

#------------------------------------------------------------------------------
# Inbound ICMP
#------------------------------------------------------------------------------
```

```
$ipt -N IN_ICMP
for sip in $PING; do
       $ipt -A IN_ICMP  -p icmp --icmp-type echo-request -s $sip  -j ACCEPT
       $ipt -A IN_ICMP  -p icmp --icmp-type echo-reply -s $sip    -j ACCEPT
done

$ipt -A IN_ICMP  -p icmp --icmp-type destination-unreachable  -j ACCEPT
$ipt -A IN_ICMP  -p icmp --icmp-type source-quench          -j ACCEPT
$ipt -A IN_ICMP  -p icmp --icmp-type time-exceeded          -j ACCEPT
$ipt -A IN_ICMP  -p icmp --icmp-type parameter-problem        -j ACCEPT
$ipt -A IN_ICMP  -j LOG --log-prefix "ipt in ICMP: " $LOGOPT
$ipt -A IN_ICMP  -j DROP

#-------------------------------------------------------------------------------
# Εξερχόμενα ICMP
#-------------------------------------------------------------------------------

$ipt -N OUT_ICMP
for dip in $PING; do
       $ipt -A OUT_ICMP   -p icmp --icmp-type echo-request -s $dip  -j ACCEPT
       $ipt -A OUT_ICMP   -p icmp --icmp-type echo-reply -s $dip    -j ACCEPT
done

$ipt -A OUT_ICMP  -p icmp --icmp-type destination-unreachable  -j ACCEPT
$ipt -A OUT_ICMP  -p icmp --icmp-type source-quench          -j ACCEPT

$ipt -A OUT_ICMP  -p icmp --icmp-type parameter-problem        -j ACCEPT
$ipt -A OUT_ICMP  -j LOG --log-prefix "ipt out ICMP: " $LOGOPT
$ipt -A OUT_ICMP  -j DROP

#-------------------------------------------------------------------------------
# NAT Προορισμού
#-------------------------------------------------------------------------------


if ["$smtpip" != ""]
then
      $ipt -t nat -A PREROUTING -i $extdev -p tcp -d $extip --dport 25 \
          -j DNAT --to-destination #httpip
fi


if ["$dnsip" != ""]
then
      $ipt -t nat -A PREROUTING -i $extdev -p tcp -d $extip --dport 53 \
          -j DNAT --to-destination #httpip
fi


if ["$httpip" != ""]
then
      $ipt -t nat -A PREROUTING -i $extdev -p tcp -d $extip --dport 80 \
          -j DNAT --to-destination #httpip
fi
```

```
if ["$popip" != ""]
then
      $ipt -t nat -A PREROUTING -i $extdev -p tcp -d $extip --dport 110 \
          -j DNAT --to-destination #popip
fi


if ["$authip" != ""]
then
      $ipt -t nat -A PREROUTING -i $extdev -p tcp -d $extip --dport 113 \
          -j DNAT --to-destination #httpip
fi


if ["$imapip" != ""]
then
      $ipt -t nat -A PREROUTING -i $extdev -p tcp -d $extip --dport 143 \
          -j DNAT --to-destination #imapip
fi


#------------------------------------------------------------------------------
# NAT Προέλευσης
#------------------------------------------------------------------------------

$ipt -t nat -A POSTROUTING -o $extdev   -j SNAT --to-source $extip


#------------------------------------------------------------------------------
# Εισερχόμενη κίνηση στο προστατευμένο δίκτυο
#------------------------------------------------------------------------------

$ipt-N IN_NETWORK
$ipt-N IN_NETWORK -p icmp                                -j IN_ICMP
$ipt-N IN_NETWORK -p tcp                                 -j TCP_FLAGS
$ipt-N IN_NETWORK -p tcp --syn                                  -j SYN_FLOOD
$ipt-N IN_NETWORK -p tcp --syn -m state ESTABLISHED,RELATED -j ACCEPT
$ipt-N IN_NETWORK -p udp --syn -m state ESTABLISHED,RELATED -j ACCEPT

if ["$smtpip" != ""]
then
      ipt -A IN_NETWORK -p tcp --syn -d $smtpip --dport 25  -j ACCEPT
fi


if ["$dnsip" != ""]
then
      ipt -A IN_NETWORK -p tcp --syn -d $dnsip  --dport 53  -j ACCEPT
fi


if ["$httpip" != ""]
then
      ipt -A IN_NETWORK -p tcp --syn -d $httpip --dport 80  -j ACCEPT
fi
```

```
if ["$popip" != ""]
then
      ipt -A IN_NETWORK -p tcp --syn -d $popip  --dport 110  -j ACCEPT
fi


if ["$authip" != ""]
then
      ipt -A IN_NETWORK -p tcp --syn -d $authip --dport 113  -j ACCEPT
fi


if ["$imapip" != ""]
then
      ipt -A IN_NETWORK -p tcp --syn -d $imapip --dport 143  -j ACCEPT
fi


#--------------------------------------------------------------------------
# Εξερχόμενη κίνηση από προστατευμένο δίκτυο
#--------------------------------------------------------------------------

$ipt-N OUT_NETWORK
$ipt-A OUT_NETWORK -p icmp                                  -j OUT_ICMP
$ipt-A OUT_NETWORK -p tcp                                   -j TCP_FLAGS
$ipt-A OUT_NETWORK -m state --state ESTABLISHED,RELATED           -j ACCEPT

# Οι παρακάτω κανόνες επιτρέπουν σε πελάτες που υπάρχουν στον προστατευμένο
# δίκτυο να συνδέονται σε απομακρυσμένους διακομιστές. Προσθέστε ή αφαιρέστε
# γραμμές για να τροποποιήσετε την εξουσιοδοτημένη κίνηση.

$ipt -A OUT_NETWORK -m state --state NEW -p tcp --dport 21  -j ACCEPT # ftp
$ipt -A OUT_NETWORK -m state --state NEW -p tcp --dport 22  -j ACCEPT # ssh
$ipt -A OUT_NETWORK -m state --state NEW -p tcp --dport 25  -j ACCEPT # smtp
$ipt -A OUT_NETWORK -m state --state NEW -p tcp --dport 80  -j ACCEPT # http
$ipt -A OUT_NETWORK -m state --state NEW -p tcp --dport 443 -j ACCEPT # https
$ipt -A OUT_NETWORK -m state --state NEW -p tcp --dport 53  -j ACCEPT # domain

#--------------------------------------------------------------------------
# Εισερχόμενη κίνηση προς το firewall
#--------------------------------------------------------------------------

$ipt -N IN_FIREWALL
$ipt -A IN_FIREWALL -p icmp                                 -j IN_ICMP
$ipt -A IN_FIREWALL -p tcp                                  -j TCP_FLAGS
$ipt -A IN_FIREWALL -p tcp --syn                            -j SYN_FLOOD
$ipt -A IN_FIREWALL                                         -j IN_IP_CHECK
$ipt -A IN_FIREWALL -m state --state ESTABLISHED,RELATED    -j ACCEPT

for sip in $ssh
do
     $ipt -A IN_FIREWALL  -p tcp -s $sip   --dport 22 -m state  --state NEW
        -j ACCEPT
done

$ipt -A IN_FIREWALL -j LOG --log-prefix "ipt IN_FIREWALL: " $LOGOPT
```

```
$ipt -A IN_FIREWALL -j DROP

#--------------------------------------------------------------------------------
# Εξερχόμενη κίνηση από το firewall
#--------------------------------------------------------------------------------

$ipt -N OUT_FIREWALL
$ipt -A OUT_FIREWALL -p icmp                                    -j OUT_ICMP
$ipt -A OUT_FIREWALL -p tcp                                     -j TCP_FLAGS
$ipt -A OUT_FIREWALL -m state --state ESTABLISHED,RELATED       -j ACCEPT
$ipt -A OUT_FIREWALL                                            -j OUT_IP_CHECK

# Οι παρακάτω κανόνες επιτρέπουν σε πελάτες που υπάρχουν στον προστατευμένο
# δίκτυο να συνδέονται σε απομακρυσμένους διακομιστές. Προσθέστε ή αφαιρέστε
# γραμμές για να τροποποιήσετε την εξουσιοδοτημένη κίνηση.

$ipt -A OUT_FIREWALL -m state --state NEW -p tcp --dport 21  -j ACCEPT # ftp
$ipt -A OUT_FIREWALL -m state --state NEW -p tcp --dport 22  -j ACCEPT # ssh
$ipt -A OUT_FIREWALL -m state --state NEW -p tcp --dport 25  -j ACCEPT # smtp
$ipt -A OUT_FIREWALL -m state --state NEW -p tcp --dport 80  -j ACCEPT # http
$ipt -A OUT_FIREWALL -m state --state NEW -p tcp --dport 443 -j ACCEPT # https
$ipt -A OUT_FIREWALL -m state --state NEW -p tcp --dport 53  -j ACCEPT # domain

$ipt -A OUT_FIREWALL -j LOG --log-prefix "ipt OUT_FIREWALL: " $LOGOPT
$ipt -A OUT_FIREWALL -j DROP


#--------------------------------------------------------------------------------
# Κύριοι κανόνες του firewall
#--------------------------------------------------------------------------------
$ipt -A FORWARD         -j SHUN
$ipt -A FORWARD -i $extdev    -j IN_NETWORK
$ipt -A FORWARD -i $intdev    -j OUT_NETWORK
$ipt -A FORWARD               -j LOG --log-prefix "ipt forward: " $LOGOPT
$ipt -A FORWARD               -j DROP

$ipt -A INPUT                 -j SHUN
$ipt -A INPUT -i lo           -j ACCEPT
$ipt -A INPUT                 -j IN_FIREWALL
$ipt -A INPUT                 -j LOG --log-prefix "ipt input: " $LOGOPT
$ipt -A INPUT                 -j DROP

$ipt -A OUTPUT                -j SHUN
$ipt -A OUTPUT -o lo          -j ACCEPT
$ipt -A OUTPUT                -j OUT_FIREWALL
$ipt -A OUTPUT                -j LOG --log-prefix "ipt output: " $LOGOPT
$ipt -A OUTPUT                -j DROP
```

SQUID

Κεφάλαιο 7

7.1 Τι είναι το Squid;

Το Squid είναι ένα δωρεάν, υψηλών ταχυτήτων, internet proxying – caching λογισμικό. Αλλά τι σημαίνει ο όρος proxy cache; Οι όροι μεταφράζονται ως εξής:

Proxy: Ένας μεσάζοντας με την εξουσιοδότηση να λειτουργεί για κάποιον άλλον.

Cache: Ένα μέρος αποθήκευσης για τη διαφύλαξη και τη διατήρηση δεδομένων τα οποία χρησιμοποιούνται συχνά από εξουσιοδοτημένους χρήστες.

Το Squid δρα σαν ένας μεσάζοντας, δέχεται αιτήσεις από πελάτες όπως browsers, τις περνάει στον αρμόδιο internet διακομιστή, και αποθηκεύει ένα αντίγραφο των προς επιστροφή δεδομένων σε ένα δίσκο cache. Το μεγάλο προτέρημα είναι ότι το Squid ξαναπαρουσιάζεται στο προσκήνιο όταν τα ίδια δεδομένα ζητηθούν πολλές φορές. Εφόσον ένα αντίγραφο δεδομένων υπάρχει στο δίσκο τα δεδομένα επιστρέφονται στον πελάτη και έτσι γίνεται πιο γρήγορη η πρόσβαση εφόσον σώζονται bandwidth.

Πολλά internet firewall συχνά περιλαμβάνουν έναν proxy. Η διαφορά του Squid proxy με έναν firewall proxy είναι ότι οι firewall proxies δεν αποθηκεύουν αντίγραφα των προς επιστροφή δεδομένων, αλλά αντί για αυτό αναεξάγουν τις αιτήσεις από τον απομακρυσμένο internet διακομιστή.

Χαρακτηριστικά του Squid:

- Χρησιμοποίηση λιγότερου bandwidth της σύνδεσης του internet όταν σερφάρουμε στο web
- Μείωση του χρόνου φόρτωσης των σελίδων web

- Προστασία των host του εσωτερικού δικτύου χρησιμοποιώντας proxy για την κίνηση web
- Συλλογή στατιστικών για την κίνηση του δικτύου
- Παρεμπόδιση χρηστών από το να επισκέπτονται συγκεκριμένα site (π.χ. πορνογραφικού υλικού κτλ)
- Επιβεβαίωση ότι μόνο εξουσιοδοτημένοι χρήστες μπορούν να σερφάρουν στο internet
- Βελτίωση της ασφάλειας των χρηστών φιλτράροντας ευαίσθητες πληροφορίες
- Μείωση του φόρτου στους web διακομιστές
- Μετατροπή κωδικοποιημένων αιτήσεων σε αποκωδικοποιημένες (HTTPS σε HTTP)
- Υποστήριξη πολλών πρωτοκόλλων (Τα firewalls συχνά έχουν συγκεκριμένους proxy για κάθε πρωτόκολλο, πράγμα το οποίο καθιστά δύσκολο την επιβεβαίωση της ασφάλειας κώδικα σε ένα μεγάλο πρόγραμμα)

7.2 Υποστηριζόμενα πρωτόκολλα

Υποστηριζόμενα πρωτόκολλα πελάτη

Το Squid υποστηρίζει τους παρακάτω τύπους πρωτοκόλλων εισερχόμενων αιτήσεων

- Hyper Text Transfer Protocol (HTTP), πάνω στο οποίο βασίζεται το WWW
- File Transfer Protocol (FTP)
- Gopher
- Wide Area Information
- Secure Socket Layer, πάνω στο οποίο βασίζονται οι on-line συναλλαγές

Υποστηριζόμενα πρωτόκολλα διαχείρισης και επικοινωνίας

- HTTP, το οποίο συχνά λαμβάνει αντιγραφές των αντικειμένων από την cache
- Internet cache protocol (ICP). Το ICP χρησιμοποιείται για τον έλεγχο του αν ένα συγκεκριμένα αντικείμενο είναι αποθηκευμένο σε άλλη cache

- Cache Digests. Αυτό το πρωτόκολλο χρησιμοποιείται για την δημιουργία ευρετηρίου των αντικειμένων που είναι αποθηκευμένα σε απομακρυσμένες cache. Όταν η cache δέχεται μία αίτηση για ένα αντικείμενο το οποίο δεν έχει, τότε ελέγχει το ευρετήριο για να βρει ποια cache έχει το αντικείμενο.
- Simple Network Management Protocol (SNMP). Συχνά εργαλεία SNMP χρησιμοποιούνται για την ανάκτηση πληροφοριών σχετικά με την cache.
- HyperText Caching Protocol (HTCP)

7.3 Ορολογία του firewall

Τα firewalls χρησιμοποιούνται από πολλές εταιρίες για να προστατεύσουν τα δίκτυά τους. Το Squid πρέπει να παρεμβάλετε στο firewall για να είναι χρήσιμο. Ένα proxy based firewall δεν δρομολογεί πακέτα μέσω του τελικού χρήστη. Όλα τα εισερχόμενα πακέτα διαχειρίζονται από την IP stack του firewall με προγράμματα τα οποία μεταχειρίζονται την μεταφορά των δεδομένων. Οι proxy δέχονται εισερχόμενα δεδομένα, τα επεξεργάζονται και τα περνάνε σε μηχανήματα του εσωτερικού δικτύου. Το λογισμικό που τρέχει σε ένα proxy level firewall είναι γραμμένο με ασφαλή τρόπο έτσι ώστε για να τον παραβιάσουμε θα πρέπει να βρούμε μια τρύπα στο ίδιο το λογισμικό του firewall αντί για το λογισμικό του εσωτερικού μηχανήματος.

Το firewall φιλτραρίσματος πακέτων αποφασίζει για τη μεταφορά πακέτων από και προς τα εσωτερικά μηχανήματα με βάση του IP πρωτοκόλλου και τα ζεύγη διεύθυνσης IP αφετηρίας και προορισμού και πόρτων.

Με χρήση firewall το δίκτυο διαιρείται σε δύο μέρη: Στους trusted host και στους untrusted host. Οι trusted είναι από τη μια πλευρά του firewall ενώ οι untrusted είναι γενικά το internet. Συχνά μπορεί να έχουμε και μια άλλη ζώνη, τους semi-trusted. Αυτοί οι host στεγάζονται σε ένα άλλο κομμάτι του δικτύου, ξεχωριστά από τα άλλα συστήματα. Δεν είναι trusted από το εσωτερικό δίκτυο, αλλά είναι προστατευμένοι από το firewall. Δημόσιοι servers γενικά τοποθετούνται εδώ. Αυτή η ζώνη ή το τμήμα ονομάζεται demilitarized zone ή DMZ.

Hand-Off

Με έναν επιπέδου proxy firewall τα μηχανήματα πελάτες είναι ρυθμισμένα για να χρησιμοποιούνε την εσωτερική κάρτα δικτύου του firewall σαν proxy διακομιστή. Μερικά firewall μπορούν να κάνουν αυτόματη ανακατεύθυνση των εξωτερικών web requests, αλλά δεν μπορούν να κάνουν πιστοποίηση. Έτσι αν έχουμε έναν μεγάλο αριθμό πελατών ρυθμισμένων να μιλάνε στο firewall σαν proxy η προοπτική του να αλλάξεις της ρυθμίσεις του μπορεί να επηρεάσει το πού θα εγκαταστήσεις τον cache διακομιστή. Σε πολλές περιπτώσεις είναι πιο εύκολο να επαναριθμήσεις το firewall για να επικοινωνεί με τον πελάτη, από το να αλλάξεις τις ρυθμίσεις του proxy διακομιστή σε όλα τα μηχανήματα πελάτες.

Η μεγάλη πλειοψηφία των επιπέδου proxy firewall είναι ικανή να επικοινωνήσει με έναν άλλο Proxy διακομιστή χρησιμοποιώντας το HTTP. Αυτό το χαρακτηριστικό πολλές φορές ονομάζεται hand-off και είναι αυτό το οποίο επιτρέπει στο firewall να επικοινωνεί με υψηλότερου επιπέδου firewall ή με cache διακομιστές μέσω του HTTP. Το hand-off επιτρέπει να δημιουργήσουμε σωρούς από firewall με υψηλότερου επιπέδου firewall τα οποία προστατεύουν ολόκληρες εταιρίες από εξωτερικές επιθέσεις, και με χαμηλότερου επιπέδου firewall για να προστατεύουν τον έναν τομέα από τον άλλο. Όταν το firewall κάνει hand-off σε μία αίτηση για άλλο firewall ή proxy διακομιστή, αυτό λειτουργεί σαν αγωγός μεταξύ του πελάτη και του απομακρυσμένου firewall.

7.4 Βασικές ρυθμίσεις του Squid

Όλα τα αρχεία ρυθμίσεων του Squid φυλάσσονται στον κατάλογο /usr/local/squid/etc. Αν και υπάρχουν πολλά αρχεία σε αυτόν τον κατάλογο, μόνο ένα είναι το πιο σημαντικό για τους διαχειριστές, το squid.conf. Αν και υπάρχουν 125 παράμετροι ρυθμίσεων χρειαζόμαστε μόνο να ρυθμίσουμε 8 ώστε να κάνουμε το Squid να λειτουργήσει. Οι άλλες 117 επιλογές μας δίνουν απίστευτες δυνατότητες, αλλά μπορούμε να τις μάθουμε μόνο εφόσον έχουμε κάνει το Squid να λειτουργεί.

Ρυθμίζοντας το HTTP port του Squid

Η πρώτη επιλογή στο squid.conf ρυθμίζει το http port το οποίο το Squid παρακολουθεί για εισερχόμενες αιτήσεις. Η κάθε υπηρεσία δικτύου αναφέρεται και σε ένα Port.

Τα port κάτω από το 1024 μπορούν να χρησιμοποιηθούν μόνο από τον διαχειριστή του συστήματος και χρησιμοποιούνται από προγράμματα που παρέχουν βασικές λειτουργίες internet: SMTP, POP, DNS, HTTP. Τα port πάνω από το 1024 χρησιμοποιούνται για untrusted υπηρεσίες και για παροδικές συνδέσεις όπως εξερχόμενες αιτήσεις δεδομένων. Τυπικά οι web διακομιστές παρακολουθούν για αιτήσεις web το port 80. Το προεπιλεγμένο http port του squid είναι το 3129. Πολλοί τρέχουν τους cache διακομιστές τους σε ένα Port το οποίο είναι πιο εύκολο στην απομνημόνευση: κάτι σαν 80 ή 8080. Αν επιλέξουμε ένα μικτότερο Port πρέπει να ξεκινήσουμε το Squid σαν root (αλλιώς θεωρούμαστε untrusted και δεν μπορούμε να ξεκινήσουμε το Squid). Αν επιθυμούμε μπορούμε να ορίσουμε πολλαπλά Port στην μεταβλητή http_port. Για παράδειγμα:

```
http_port 8080
```

Χρησιμοποιώντας το port 80

Το HTTP καθορίζει την μορφοποίηση της αίτησης για πληροφορίες και της απάντησης του διακομιστή. Η βασική μορφή του πρωτοκόλλου είναι απλή: Ένας πελάτης, για παράδειγμα το browser, συνδέεται στο Port 80 και ζητάει ένα αρχείο παρέχοντας το πλήρες path και το όνομα του αρχείου που επιθυμεί να κατεβάσει. Ο πελάτης επίσης ορίζει την έκδοση του http που επιθυμεί για τη λήψη.

Για μία αίτηση Proxy η μορφοποίηση είναι λίγο διαφορετική. Ο πελάτης ορίζει ολόκληρο το url αντί για το path του αρχείου. Ο Proxy διακομιστής τότε συνδέεται στον web διακομιστή στο συγκεκριμένο url και στέλνει μια τυπική http αίτηση για αυτή τη σελίδα. Εφόσον η μορφοποίηση μιας proxy αίτησης είναι περίπου ίδια με μια τυπική http αίτηση, δεν πρέπει να μας εκπλήσσει το γεγονός ότι πολλοί web διακομιστές λειτουργούν και σαν Proxy διακομιστές επίσης.

Αποθήκευση δεδομένων στην cache

Τα δεδομένα της cache πρέπει να αποθηκευθούν κάπου. Το Squid δεν μπορεί να εντοπίσει αυτόματα που να αποθηκεύσει τα δεδομένα. Έτσι πρέπει να ορίσουμε εμείς ποιους καταλόγους να χρησιμοποιήσει για την αποθήκευση. Η παράμετρος cache_dir στο Squid.conf αρχείο χρησιμοποιείται για τον καθορισμό των περιοχών αποθήκευσης. Αν χρησιμοποιούμε περισσότερους από ένα δίσκους για την αποθήκευση δεδομένων τότε χρειαζόμαστε

περισσότερο από ένα mount points. Έτσι το Squid μας επιτρέπει να έχουμε περισσότερες από μια cache_dir επιλογές στο αρχείο ρυθμίσεων.

Αν θεωρήσουμε ότι έχουμε μόνο μια cache_dir δήλωση τότε θα είναι της ακόλουθης μορφής:

```
Cache_dir /usr/local/squid/cache/ 100 16 256
```

Η πρώτη επιλογή του cache_dir ορίζει το πού θα αποθηκευτούν τα δεδομένα. Η επόμενη ορίζει το μέγεθος σε MB. Οι άλλες δύο επιλογές είναι πιο πολύπλοκες. Ορίζουν τον αριθμό των υποκαταλόγων που μπορούν τα δημιουργηθούν στον κατάλογο. Το Squid δημιουργεί πολλούς καταλόγους και αποθηκεύει λίγα αρχεία σε κάθε ένα από αυτούς στην προσπάθειά του να επιταχύνει την πρόσβασή τους στο δίσκο.

E-Mail για το διαχειριστή της cache

Αν η λειτουργία του Squid διακοπή τότε μπορεί να σταλεί ένα e-mail στη διεύθυνση που ορίζεται από την επιλογή cache_mgr του αρχείου ρυθμίσεων. Αυτή η διεύθυνση επίσης αναφέρεται και στο τέλος των μηνυμάτων σφάλματος τα οποία λαμβάνουν οι χρήστες, όπως για παράδειγμα το απομακρυσμένο μηχάνημα είναι μη διαθέσιμο.

Πληροφορίες σύνδεσης FTP

Το Squid μπορεί να λειτουργήσει σαν Proxy διακομιστής για πολλά πρωτόκολλα. Το πιο κοινό είναι το http αλλά και το ftp είναι πολύ σημαντικό. Το FTP είναι γραμμένο για εξουσιοδοτημένη μεταφορά αρχείων. Για την παροχή δημόσιας πρόσβασης δημιουργείται ένας ειδικός λογαριασμός με την ονομασία anonymous. Όταν συνδεόμαστε με έναν FTP διακομιστή χρησιμοποιώντας αυτόν τον λογαριασμό, σαν κωδικό εισάγουμε το e-mail μας. Το squid μας επιτρέπει να ορίζουμε μια διεύθυνση e-mail με την παράμετρο ftp_user.

Λίστα ελέγχου πρόσβασης και διαχειριστές ελέγχου πρόσβασης

Το Squid δεν μπορεί να χρησιμοποιηθεί σε περιβάλλον ISP χωρίς το κατάλληλο σύστημα ελέγχου πρόσβασης. Σε πολλές περιπτώσεις απαιτείται ένα βασικό επίπεδο ελέγχου πρόσβασης. Αν έχουμε ένα μικρό δίκτυο και δεν επιθυμούμε να χρησιμοποιήσουμε έλεγχο εξουσιοδότησης ή να μπλοκάρουμε την πρόσβαση με βάση τον domain προορισμό τότε ο

παρακάτω τομέας είναι επαρκής για τη ρύθμιση του τομέα ελέγχου. Ο ποιο απλός τρόπος για να περιορίσουμε την πρόσβαση είναι να την επιτρέπουμε μόνο στις IPs που είναι στο εσωτερικό δίκτυο. Παραδείγματα για τις εισόδους ελέγχου περιλαμβάνονται στο προεγκατεστημένο squid.conf. Αυτό περιλαμβάνει καταχωρίσεις που μπορούν να μας βοηθήσουν να αποφύγουμε μερικά προβλήματα όπως bandwidth-chewing loops, cache tunneling και άλλα παρόμοια προβλήματα.

Ο έλεγχος πρόσβασης επιτυγχάνεται με την ανάλυση κάθε πρωτοκόλλου ξεχωριστά. Όταν το Squid δέχεται HTTP αιτήσεις η λίστα των ελέγχων HTTP αξιοποιείται με παρόμοιο τρόπο όταν μία αίτηση ICP εισέρχεται η λίστα ICP ελέγχεται πριν δοθεί απάντηση.

Ας υποθέσουμε ότι έχουμε μια λίστα από IP διευθύνσεις που έχουν πρόσβαση στην cache. Αν θέλουμε να έχουν πρόσβαση με HTTP και με ICP πρέπει να δώσουμε τη λίστα των IP διευθύνσεων δυο φορές. Έτσι θα έχουμε κάποιες καταχωρήσεις παρόμοιες με τις παρακάτω.

```
http_access deny 10.0.1.0/255.255.255.0
http_access allow 10.0.0.0/255.0.0.0
icp_access allow 10.0.0.0/255.0.0.0
```

Για μεγάλους οργανισμούς η διαχείριση είναι πιο εύκολη αν δημιουργήσουμε classes για τους χρήστες. Έτσι μπορούμε να επιτρέψουμε ή να απορρίψουμε classes χρηστών με ένα πιο πολύπλοκο τρόπο. Στο παρακάτω παράδειγμα έχουμε τροποποιήσει το προηγούμενο χρησιμοποιώντας classes χρηστών.

```
# classes
acl mynetwork src 10.0.0.0/255.0.0.0
acl servernet src 10.0.1.0/255.255.255.0
# what HTTP access to allow classes
http_access deny servernet
http_access allow mynet
# what ICP access to allow classes
icp_access deny servernet
icp_access allow mynet
```

Οι γραμμές που αρχίζουν με acl ονομάζονται acl operators. Ένας acl operator μπορεί είτε να επιτρέψει είτε να αρνηθεί μια αίτηση. Τα acls χρησιμοποιούνται για να καθορίζουν classes. Όταν το Squid δέχεται μια αίτηση ελέγχει τη λίστα των acl operator για τον τύπο της αίτησης. Για παράδειγμα μια http αίτηση απαιτεί τον έλεγχο των http_access καταχωρήσεων. Στο προηγούμενο παράδειγμα χρησιμοποιήσαμε ένα src acl. Αυτό ελέγχει ότι η πηγή της αίτησης θα είναι μέσα από το δεδομένο IP range. Ο τύπος scr acl δέχεται ip διευθύνσεις με διάφορους τρόπους. Προηγουμένως χρησιμοποιήσαμε subnet/netmask. Η cidr (Classless internet domain routing) σημειογραφία μπορεί επίσης να χρησιμοποιηθεί. Το παρακάτω παράδειγμα μας δίνει το ίδιο ip range με διαφορετική σημειογραφία.

```
CIDR εναντίων Netmask Source IP
acl mynet1 src 10.1.0.0/255.0.0.0
acl mynet2 src 10.2.0.0/16
```

Το squid.conf περιλαμβάνει acl καταχωρίσεις οι οποίες απορρίπτουν όλες τις http αιτήσεις. Για να χρησιμοποιήσουμε την cache χρειαζόμαστε σαφώς να επιτρέψουμε τις εισερχόμενες αιτήσεις από το δεδομένο ip range.

```
Πλήρες Παράδειγμα acl λίστας
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
# acls for my network addresses
acl my-iplist-1 src 192.168.1.0/24
acl my-iplist-2 src 10.0.0.0/255.255.0.0
# Check that requests are from users on our network
http_access allow my-iplist-1
http_access allow my-iplist-2
icp_access allow my-iplist-1
icp_access allow my-iplist-2
# allow requests from the local machine (for testing)
http_access allow localhost
# End of locally-inserted rules
http_access deny all
```

Τα ACL always_direct και never_direct καθορίζουν πότε να προωθήσουν μια σύνδεση ή να συνεχίσουν άμεσα. Τα παρακάτω είναι μια λίστα από operators η οποία βασίζεται στην προηγούμενη παράγραφο, αλλά με τη χρήση always_direct και never_direct operators.

Χρήση always_direct και never_direct

```
# acls for my network addresses
acl my-iplist-1 src 192.168.1.0/24
acl my-iplist-2 src 10.0.0.0/255.255.0.0
# Various programs running on the cache box connect to Squid, so
it's
# useful to allow connections from the localhost address.
acl localhost src 127.0.0.1/255.255.255.255
# used to deny all requests: Since the netmask is all 0's, any
request
# matches this acl
acl all src 0.0.0.0/0.0.0.0
# Check that requests are from users on our network
http_access allow my-iplist-1
http_access allow my-iplist-2
icp_access allow my-iplist-1
icp_access allow my-iplist-2
# check the localhost acl as a special case
http_access allow localhost
# If the requests comes from any other IP, deny all access.
http_access deny all
# always go direct to local machines
always_direct allow my-iplist-1
always_direct allow my-iplist-2
# never go direct to other hosts
never_direct allow all
```

Επικοινωνώντας με άλλους Εξυπηρετητές

Το Squid υποστηρίζει ιεράρχηση εξυπηρετητών. Αν ο εξυπηρετητής μας δεν έχει ένα αντικείμενο στο δίσκο του η προεπιλεγμένη λειτουργία είναι να συνδεθεί με τον web διακομιστή και να κάνει λήψη της σελίδας. Με την ιεράρχηση ο εξυπηρετητής μπορεί να επικοινωνήσει με άλλους εξυπηρετητές με τη σκέψη ότι ένας από αυτούς θα έχει τη σχετική σελίδα. Προφανώς η επικοινωνία πρέπει να γίνεται με κοντινούς διακομιστές, αλλιώς μπορεί να καταλήξουμε να καθυστερήσουμε υπερβολικά την πρόσβαση. Επιπλέον αν η πρόσβαση στον αρχικό διακομιστή είναι γρηγορότερη από κάποια γειτονική cache δεν είναι καλή ιδέα να πάρουμε τη σελίδα από την cache.

Η δυνατότητα να χειριστούμε και άλλες cache είναι πολύ χρήσιμη σε μερικές περιπτώσεις. Για παράδειγμα αν επικοινωνούμε συχνά με μια εταιρία, μπορούμε να ρυθμίσουμε την cache να επικοινωνεί με αυτή της εταιρίας. Έτσι μειώνουμε το χρόνο αντίδρασης. Είναι σχεδόν σίγουρο ότι θα είναι πιο γρήγορο να κάνουμε λήψη της σελίδας από εκεί παρά από κάποιο άλλο σημείο της χώρας.

Η επιλογή cache_peer μας επιτρέπει να καθορίσουμε τους εξυπηρετητές με τους οποίους θα επικοινωνεί ο διακομιστής μας. Η πρωταρχική επικοινωνία του squid γίνεται στο Port 3128 και χρησιμοποιεί το icp για να ζητήσει επικοινωνία με τους διακομιστές στο port 3130. Η παραμετροποίηση του squid για τη χρήση περισσοτέρων από έναν εξυπηρετητών είναι πολύ εύκολη: απλά προσθέτουμε μια ακόμα cache_peer καταχώρηση.

```
cache_peer cache.myparent.example parent 3128 3130
cache_peer cache.sibling.example sibling 8080 3130
```

Η cache του ISP

Αν πρέπει να χρησιμοποιήσουμε την cache του παροχέα υπηρεσιών internet πρέπει να παραμετροποιήσουμε το squid για να κάνει αιτήσεις στον συγκεκριμένο υπολογιστή. Το squid θα κάνει προσπάθεια να επικοινωνήσει κάνοντας ICP αιτήσεις. Αυτό μοιάζει με ένα ping. Αν δεν υπάρξει απάντηση στην αίτηση, το squid θα προσπαθήσει να επικοινωνήσει με τον αρχικό διακομιστή, εφόσον δεν μπορεί να επικοινωνήσει με την cache του ISP. Επίσης αν θελήσουμε να μειώσουμε το χρόνο αντίδρασης μπορούμε να προσθέσουμε την default_query επιλογή στο τέλος της cache_peer καταχώρησης.

```
cache_peer cache.myisp.example parent 3128 3130 default no-query
```

Η default επιλογή λέει στο squid «Ψάξε την cache για όλες τις αιτήσεις, Αν δεν βρεθούν επέστρεψε ένα μήνυμα σφάλματος». Η no-query επιλογή λέει στο squid να αγνοήσει το δεδομένο icp port και ποτέ να μην κάνει προσπάθεια σύνδεσης με την cache χρησιμοποιώντας το icp.

Παρεμβολές Firewall

Τα firewall μπορούν να κάνουν την παραμετροποίηση της cache πολύπλοκη. Τα πρωτόκολλα αυτά χρησιμοποιούν πακέτα τα οποία προκαλούν σύγχυση στο firewall. Οι περισσότερες cache χρησιμοποιούν το ICP το οποίο είναι μία στρώση πάνω από το UDP. Το UDP είναι δύσκολο να γίνει ασφαλές και για αυτό το λόγο οι διαχειριστές το απενεργοποιούν, όσο είναι δυνατόν.

Είναι προτεινόμενο να τοποθετούμε τον cache server στον DMZ. Υπάρχουν κάποια προτερήματα για αυτό:

- Ο cache διακομιστής κρατείται ασφαλής
- Το firewall μπορεί να παραμετροποιηθεί για να δέχεται αιτήσεις για τον cache διακομιστή
- Θα μπορούμε να επικοινωνήσουμε με άλλες εξωτερικές cache εφόσον τα DMZ δίκτυα έχουν λιγότερο συμπαγείς κανόνες

Αυτός ο τομέας μας βοηθάει να ρυθμίσουμε το squid και το firewall ώστε να συνεργάζονται. Υπάρχουν κάποιες περιπτώσεις για κάθε τύπο firewall: Η cache μέσα από το firewall, η cache έξω από το firewall, η cache στο DMZ.

7.5 Proxying Firewalls

Αν χρησιμοποιούμε ένα επιπέδου proxy firewall, τα μηχανήματα πελάτες θα είναι πιθανόν ρυθμισμένα να χρησιμοποιούν την εσωτερική IP του firewall σαν proxy διακομιστή. Το firewall πρέπει επίσης να τρέχει σε transparent mode όπου εκείνο αυτόματα θα επιλέγει τις

εξερχόμενες αιτήσεις Web. Αν έχουμε έναν μεγάλο αριθμό μηχανημάτων πελατών δεν επιθυμούμε να τα επαναπαραμετροποιήσουμε. Σε αυτήν την περίπτωση μπορούμε να τοποθετήσομε το firewall έξω από την cache ή στον DMZ και να το παραμετροποιήσουμε να μεταφέρει τις αιτήσεις στην cache.

Η cache μέσα από το firewall

Η cache θεωρείται ένας έμπιστος host και προστατεύεται από το firewall. Χρειάζεται να ρυθμίσουμε τα μηχανήματα πελάτες να χρησιμοποιούνε τον cache διακομιστή στις ρυθμίσεις του περιηγητή proxy. Έτσι όταν γίνει μια αίτηση ο cache διακομιστής θα περάσει την εξερχόμενη αίτηση στο firewall θεωρώντας το firewall σαν parent proxy διακομιστή. Το firewall τότε θα συνδεθεί με το διακομιστή προορισμού. Αν έχουμε έναν μεγάλο αριθμό πελατών ρυθμισμένων να χρησιμοποιούν το firewall σαν proxy διακομιστή θα πρέπει να βάλουμε το firewall να μεταφέρει τις εισερχόμενες http αιτήσεις πίσω στο δίκτυο στον cache διακομιστή. Αυτό είναι λιγότερο αποδοτικό εφόσον η cache θα πρέπει να ξαναπερνάει τις αιτήσεις στο firewall για να εξέλθουν από το δίκτυο χρησιμοποιώντας την επιλογή cache_peer. Εφόσον η κίνηση θα περνάει δυο φορές από το firewall ο φόρτος εργασίας του θα αυξηθεί.

Η cache έξω από το firewall

Υπάρχουν δύο κύριοι λόγοι για να τοποθετήσουμε την cache έξω από το firewall.

- Το squid μπορεί να ρυθμιστεί για να κάνει πιστοποίηση. Το πρόβλημα είναι ότι μπορεί να απαιτήσουμε διπλή πιστοποίηση, δηλαδή και από το firewall και από τον proxy. Έτσι αν θέλουμε να συνεχίσουμε πιστοποιώντας τους χρήστες και στο firewall θα πρέπει να τοποθετήσουμε την cache έξω από αυτό ή στο DMZ. Το firewall θα δέχεται τις αιτήσεις από τους πελάτες, θα τις πιστοποιεί και θα τις περνά στον cache διακομιστή
- Η επικοινωνία με ιεραρχικές cache είναι πιο εύκολη. Οι cache διακομιστές μπορούν να επικοινωνήσουν με άλλα συστήματα χρησιμοποιώντας οποιαδήποτε πρωτόκολλα. Μπορούμε να τοποθετήσουμε την cache έξω από το firewall μόνο αν αυτό υποστηρίζει hand-off. Οι περιηγητές θα συνδέονται με το firewall, θα κάνουν αίτηση για ένα URL και το firewall θα συνδέεται με την εξωτερική cache για να ζητήσει τη σελίδα.

Η cache στο DMZ

Αν μας ενδιαφέρει η ασφάλεια του cache διακομιστή και θέλουμε να επικοινωνούμε με εξωτερικούς cache διακομιστές πρέπει να τοποθετήσουμε την cache στο DMZ. Εφόσον η cache είναι έξω από το firewall δεν χρειάζεται να θεωρεί το firewall σαν parent και απλά δέχεται αιτήσεις από αυτό, ενώ ποτέ δεν μεταφέρει αιτήσεις σε αυτό. Αφού το firewall λειτουργεί σαν φίλτρο μεταξύ της cache και του εξωτερικού δικτύου, πρέπει να ανοίξουμε κάποια ports στο firewall. Η cache πρέπει να μπορεί να συνδεθεί με το port 80 κάθε εξωτερικού μηχανήματος. Εφόσον κάποιοι έγκυροι web διακομιστές τρέχουν και σε Port εκτός του 80 θα πρέπει να επιτρέψουμε τις συνδέσεις σε οποιοδήποτε port στον cache διακομιστή.

Γενικότερα να επιτρέψουμε συνδέσεις σε:

- Port 80 (HTTP αιτήσεις)
- Port 443 (HTTPS αιτήσεις)
- Port >1024 (Μηχανές αναζήτησης)

Αν πρόκειται να επικοινωνήσουμε με cache διακομιστές έξω από το firewall χρειαζόμαστε περισσότερα port ανοιχτά. Αν πρόκειται να επικοινωνήσουμε με ICP πρέπει να επιτρέψουμε UDP κίνηση από και προς τον cache διακομιστή στο Port 3130.

7.6 Εκτελώντας το Squid

Τα προγράμματα τα οποία διαχειρίζονται δικτυακές αιτήσεις συνήθως τρέχουν στο background. Φορτώνονται κατά την εκκίνηση και καταγράφουν τα μηνύματα τους σε ένα αρχείο. Αυτά συχνά αναφέρονται και σαν δαίμονες. (daemons). Το squid είναι και αυτό ένα τέτοιο πρόγραμμα, όταν εκτελέσουμε το binary επιστρέφουμε αμέσως στη γραμμή εντολών. Ενώ μοιάζει σαν να μην έγινε τίποτα το πρόγραμμα δρα στο background περιμένοντας για εισερχόμενες αιτήσεις. Θέλοντας να δούμε ότι το squid κάνει κάτι χρήσιμο αυξήσαμε το επίπεδο debug (χρησιμοποιώντας το -d 1) και του είπαμε να μην εξαφανιστεί στο background (-N). Αν το μηχάνημα μας δεν είναι συνδεδεμένο στο internet πρέπει να χρησιμοποιήσουμε και την επιλογή –D.

Εκτέλεση:

```
cache1:~ # /usr/local/squid/bin/squid -N -d 1 -D
2008/06/12 19:16:20| Starting Squid Cache version 2.2.32008/06/12
19:16:20| Process ID 4121
2008/06/12 19:16:20| With 256 file descriptors available
2008/06/12 19:16:20| helperOpenServers: Starting 'dnsserver'
2008/06/12 19:16:20| Unlinkd pipe opened on FD 13
2008/06/12 19:16:20| Swap maxSize 10240 KB, estimated 787 objects
2008/06/12 19:16:20| Target number of buckets: 15
2008/06/12 19:16:20| Using 8192 Store buckets
2008/06/12 19:16:20| Max Mem  size: 8192 KB
2008/06/12 19:16:20| Max Swap size: 10240 KB
2008/06/12 19:16:20| Rebuilding storage in Cache Dir #0 (DIRTY)
2008/06/12 19:16:20| Loaded Icons.
2008/06/12 19:16:20| Accepting HTTP connections on port 3128
2008/06/12 19:16:20| Accepting ICP messages on port 3130, FD 36.
2008/06/12 19:16:20| Accepting HTCP messages on port 4827, FD 37.
2008/06/12 19:16:20| Ready to serve requests.
2008/06/12 19:16:20| storeRebuildFromDirectory: DIR #0 done!
2008/06/12 19:16:25| Finished rebuilding storage disk.
2008/06/12 19:16:25|         0 Entries read from previous logfile.
2008/06/12 19:16:25|         0 Entries scanned from swap files.
2008/06/12 19:16:25|         0 Invalid entries.
2008/06/12 19:16:25|         0 With invalid flags.
2008/06/12 19:16:25|         0 Objects loaded.
2008/06/12 19:16:25|         0 Objects expired.
2008/06/12 19:16:25|         0 Objects cancelled.
2008/06/12 19:16:25|         0 Duplicate URLs purged.
2008/06/12 19:16:25|         0 Swapfile clashes avoided.
2008/06/12 19:16:25|   Took 5 seconds (   0.0 objects/sec).
2008/06/12 19:16:25| Beginning Validation Procedure
2008/06/12 19:16:26| storeLateRelease: released 0 objects
2008/06/12 19:16:27|   Completed Validation Procedure
2008/06/12 19:16:27|   Validated 0 Entries
2008/06/12 19:16:27|   store_swap_size = 21k
```

Δοκιμάζοντας το Squid

Οι δοκιμές μπορούν να πραγματοποιηθούν με το πρόγραμμα client το οποίο περιλαμβάνεται στον κώδικα του squid στον κατάλογο /usr/local/squid/bin. Το πρόγραμμα αυτό συνδέεται με την cache, κάνει αίτηση για μια σελίδα και εμφανίζει πληροφορίες χρονισμού. Το πρόγραμμα client είναι παραμετροποιημένο να συνδέεται με το local host στο port 3128. Αν εκτελούμε το πρόγραμμα client στον cache server μπορεί να ανακτήσει τη σελίδα με την εντολή:

```
client http://squid.nlanr.net/
```

Αν η cache βρίσκεται σε άλλο υπολογιστή πρέπει να χρησιμοποιήσουμε τις επιλογές –h και –p. Η παρακάτω εντολή μας συνδέει με τον υπολογιστή cache.qualica.comf στο port 8080.

```
cache1:~ $ /usr/local/squid/bin/client -h cache.qualica.com -p
8080 http://www.squid-cache.com/
```

Επίσης μπορούμε να ανακτήσουμε άμεσα μια σελίδα με την εντολή:

```
cache1:~ $ /usr/local/squid/bin/client -h www.squid-cache.com -p
80 /
```

Το πρόγραμμα client επίσης μπορεί να εμφανίσει πληροφορίες σχετικά με τον χρόνο κατεβάσματος της σελίδας. Με αυτήν την επιλογή τα περιεχόμενα της σελίδας δεν εμφανίζονται.

```
cache1:~ $ /usr/local/squid/bin/client -g 0 -h http://www.squid-
cache.com/ -p 80
```

Αρχεία εκκίνησης

Αν χρησιμοποιείται ήδη κάποιο αρχείο εκκίνησης τότε μπορείτε απλά να χρησιμοποιήσετε αυτό για την εκτέλεση την εντολής RunCache στο background, κάτι το οποίο γίνεται με την προσθήκη του & στο τέλος την εντολής.

Παράδειγμα:

```
/usr/local/bin/RunCache &
```

Το πρόγραμμα RunCache προσπαθεί επίσης να επανεκτελέσει το squid αν για κάποιο λόγο διακοπή η λειτουργία του και αποθηκεύει τις καταγραφές του στο *usr/local/squid/squid.out* και στο *syslog*.

7.7 Χρήση ACLS

Οι access control lists (acls) είναι συχνά το πιο δύσκολο μέρος στη ρύθμιση του squid. Η πρωταρχική χρήση του συστήματος acl είναι για την πραγματοποίηση του ελέγχου πρόσβασης. Οι περισσότεροι χρησιμοποιούν το squid μόνο για απλό έλεγχο πρόσβασης, κόβοντας την πρόσβαση σε άτομα εξωτερικών δικτύων. Παρόλα αυτά το squid παρέχει αμέτρητες και πολύ χρήσιμες δυνατότητες παραμετροποίησης με τις οποίες θα ασχοληθούμε σε αυτό το κεφάλαιο.

Access classes και Operators

Υπάρχουν δυο στοιχεία για τον έλεγχο πρόσβασης, τα λεγόμενα classes και operators. Οι classes καθορίζονται με το *acl* tag στο configuration file ενώ οι operators μπορεί να ποικίλουν: Ο πιο κοινός operator είναι το *http_access*.

Classes: Συχνά αναφέρονται σε ομάδες χρηστών. (Μπορούν να αναφέρονται ακόμα και σε destination domains ή επεκτάσεις αρχείων). Αν για παράδειγμα έχουμε μια λίστα 50 ατόμων που θέλουμε να έχουν πρόσβαση στο Internet, μπορούμε να βάλουμε τις IPs τους σε μία λίστα και χρησιμοποιώντας το όνομα της λίστας να έχουμε έλεγχο πρόσβασης.

Operators: Είναι συχνά χρήσιμο να χρησιμοποιούμε μια ομάδα ACLs για το ICP και μια για το HTTP. Με αυτό τον τρόπο μπορούμε να μπορούμε να καθορίσουμε διαφορετικούς κανόνες για διαφορετικά πρωτόκολλα. Για παράδειγμα οι περισσότεροι ISPs δεν θέλουν οι caches να είναι SNMP-queried από όλους τους πελάτες τους ενώ θέλουν όλοι να έχουν πρόσβαση στη χρήση περιηγητών. Με άλλα λόγια θέλουν μια ομάδα acl κανόνων που να

καθορίζουν την http πρόσβαση και μια για το SNMP. Για κάθε πρωτόκολλο υπάρχει και ένας διαφορετικός acl operator. Για παράδειγμα έχουμε τα *http_access, icp_access,* και *snmp_access* tags. Είναι επίσης πολύ σημαντικό να σημειώσουμε ότι δεν υπάρχει *ftp_access* τύπος. Οι ftp αιτήσεις διαχειρίζονται μέσα από το *http_access* tag.

Στο παρακάτω παράδειγμα ένας διαχειριστής εγκαθιστά μια cache και δεν επιθυμεί να έχουν άλλα άτομα πρόσβαση σε αυτή μέχρι το πέρας των εργασιών. (Ο υπολογιστής του διαχειριστή έχει IP διεύθυνση 10.0.0.3)

```
acl myIP src 10.0.0.3/255.255.255.255
acl myNet src 10.0.0.0/255.255.0.0
http_access allow myIP
http_access deny myNet
```

Αν ο διαχειριστής συνδεθεί με την cache το squid κάνει τις παρακάτω ενέργειες:

- Δέχεται την (HTTP) σύνδεση και διαβάζει την αίτηση
- Ελέγχει τη γραμμή *http_access allow myIP*
- Εφόσον οι IP ταιριάζει με αυτές που έχουν οριστεί, επιτρέπει την πρόσβαση

Ενώ αν κάποιος άλλος υπολογιστής του εσωτερικού δικτύου προσπαθήσει να συνδεθεί με την cache τότε:

- Δέχεται την (HTTP) σύνδεση και διαβάζει την αίτηση
- Οι αφετηρία της σύνδεσης δεν ταιριάζει με αυτές στο myIP acl και έτσι ελέγχεται και η επόμενη γραμμή.
- Στις δηλώσεις του myNET ταιριάζει η αφετηρία της σύνδεσης και τελικά απορρίπτεται η πρόσβαση

Αν τώρα κάποιος προσπαθήσει να συνδεθεί με την cache από κάποιο μη δηλωμένο εξωτερικό δίκτυο τότε το squid θα επιτρέψει την πρόσβαση. Ο λόγος που συμβαίνει αυτό είναι λίγο πολύπλοκος. Αν το squid περνώντας μέσα από τον έλεγχο των acl αλυσίδων δεν καταφέρει

να καταλήξει σε κάποια πράξη τότε έχει σαν προεπιλογή να κάνει το αντίθετο από αυτό που όριζε ο τελευταίος acl τελεστής και αν π.χ. ήταν deny access τότε θα επιτρέψει την πρόσβαση. Έτσι αν η τελευταία γραμμή απορρίπτει την πρόσβαση στο εσωτερικό δίκτυο τότε οι προερχόμενοι από αυτό δεν θα έχουν πρόσβαση αλλά όλοι όσοι προέρχονται από το Internet θα έχουν!

Με το παρακάτω παράδειγμα εκμεταλλευόμαστε αυτή την ιδιότητα του squid και καταλήγουμε σε μια σωστή παραμετροποίηση:

```
acl myNet src 10.0.0.0/255.255.0.0
http_access allow myNet
```

Έτσι εφόσον ο τελευταίος τελεστής είναι "allow" squid θα κόβει την πρόσβαση σε όσες αιτήσεις είναι "not match".

Ένα επίσης διορθωμένο παράδειγμα είναι το παρακάτω:

```
acl myIP src 10.0.0.3/255.255.255.255
acl all src 0.0.0.0/0.0.0.0
http_access allow myIP
http_access deny all
```

ACL Lines

Τα προηγούμενα παραδείγματα έδωσαν μια ιδέα της σύνταξης μια acl εντολής. Η γενική σύνταξη έχει την παρακάτω μορφή

```
acl όνομα τύπος (string|"αρχείο1") [string2] [string3]
["αρχείο2"]
```

Μοναδικό όνομα

Κάθε acl δήλωση πρέπει να έχει και ένα μοναδικό όνομα. Αν βρούμε ότι έχουμε κάνει δηλώσεις χρησιμοποιώντας το ίδιο όνομα τότε μπορούμε να προσθέσουμε έναν αριθμό δίπλα από κάθε όνομα, αυτή είναι όμως μια ενέργεια που μπορούμε να αποφύγουμε τοποθετώντας τις όμοιες classes σε ένα αρχείο και περιλαμβάνοντας το αρχείο σε ένα acl.

Τύπος

Μέχρι στιγμής έχουμε συζητήσει για acl που ελέγχουν μόνο την διεύθυνση αφετηρίας του πακέτου. Κάτι τέτοιο όμως μπορεί να μην είναι επαρκές για κάποιους. Έτσι μπορούμε να επιτρέψουμε συνδέσεις μόνο για συγκεκριμένες ώρες, ή για συγκεκριμένα domains, ή από συγκεκριμένους χρήστες ή ακόμα και ένα συνδυασμό όλων των υπολοίπων.

Decision String

Ο acl κώδικας χρησιμοποιεί αυτό το string για να ελέγξει αν το acl ταιριάζει η δεδομένη σύνδεση. Όταν χρησιμοποιούμε αυτό το πεδίο το squid ελέγχει το πεδίο τύπος για να αποφασίσει πως θα χρησιμοποιήσει το string. Το decision string μπορεί να είναι ένα εύρος IP διευθύνσεων, μια έκφραση ή μια λίστα από domains.

Επίσης μπορούμε να χρησιμοποιήσουμε τελεστές OR-AND ή διαχείριση acl εντολών αποθηκευμένων σε αρχείο.

Χρήση πολλαπλών acl αποφάσεων

```
# Εύρος διευθύνσεων IP: 10.0.0.0/255.255.255.0 OR
10.1.0.0/255.255.255.0
acl myNets src 10.0.0.0/255.255.255.0 10.1.0.0/255.255.255.0
acl all src 0.0.0.0/0.0.0.0
http_access allow myNets
http_access deny all
```

Χρήση acl από αρχείο

```
acl myNets src "/usr/local/squid/conf/data/myNets"
acl all src 0.0.0.0/0.0.0.0
http_access allow myNets
http_access deny all
```

Τύποι ACL

Μέχρι στιγμής έχουμε μιλήσει για acl που κάνουν φιλτράρισμα με βάση την διεύθυνση IP αφετηρίας αλλά υπάρχουν και πολλοί άλλοι τύποι acl όπως:

- Αφετηρίας/προορισμού IP διεύθυνση
- Αφετηρίας/προορισμού Domain
- Εκφράσεις που ταιριάζουν σε κάποιο Domain
- Λέξεις μέσα στα URL
- Εκφράσεις που ταιριάζουν σε κάποιο αφετηρίας/προορισμού Domain
- Τρέχουσα ημέρα και ώρα
- Port Προορισμού
- Πρωτόκολλο (FTP,HTTP,SSL)
- Μέθοδος (HTTP GET, HTTP POST)
- Τύπος περιηγητή
- Όνομα (με βάση το Ident πρωτόκολλο)
- Αριθμό αυτόνομου συστήματος
- Ζεύγος ονόματος/κωδικού χρήστη
- SNMP

Παραδείγματα

Φιλτράρισμα IP διεύθυνσης αφετηρίας/προορισμού

```
# Άρνηση πρόσβασης σε ένα τομέα ενός μεγαλύτερου block
acl BadDest dst 10.0.0.0/255.255.0.0
acl NiceDest dst 10.1.0.0/16
http_access deny BadDest
http_access allow NiceDest
```

Φιλτράρισμα Domain αφετηρίας/προορισμού

```
# Φιλτράρισμα ανεπιθύμητων sites
acl badDomains dstdomain adomain.example
acl badIPs dst 10.255.1.2
http_access deny badlist
http_access deny badIPs
http_access allow myNet
http_access deny all
```

Φιλτράρισμα με βάση λέξεις μέσα σε URL

```
# Άρνηση πρόσβασης σε site με τη λέξη "sex" στο URL
acl badURL url_regex -i sex
http_access deny badUrl
http_access allow myNet
http_access deny all
```

Φιλτράρισμα σε κάποιο αφετηρίας/προορισμού Domain

```
# Άρνηση πρόσβασης σε .com, .net domains αν η αίτηση γίνεται από
.za domain
acl bad_dst_TLD dstdom_regex \.com$ \.net$
acl good_src_TLD srcdom_regex \.za$
# allow requests FROM the za domain UNLESS they want to go to
\.com or \.net
http_access deny bad_dst_TLD
http_access allow good_src_TLD
```

Φιλτράρισμα με βάση παραμέτρους ημέρας/ώρας

Βασική σύνταξη και παράμετροι:

```
acl name time [day-list] [start_hour:minute-end_hour:minute]
"S-Sunday M-Monday T-Tuesday W-Wednesday H-Thursday F-Friday A-
Saturday"
acl night time 17:00-24:00
acl early_morning time 00:00-6:00
acl weekends time SA

# Επιτρέποντας  Web πρόσβαση μόνο τα Σαβ/κα.
acl myNet src 10.0.0.0/16
acl workdays time MTWHF
# allow web access only on the weekends!
http_access deny workdays
http_access allow myNet
```

Protocol (FTP, HTTP, SSL)

```
# Άρνηση πρόσβασης σε FTP sites
acl ftp proto FTP
acl myNet src 10.0.0.0/16
acl all src 0.0.0.0/0.0.0.0
http_access deny ftp
http_access allow mynet
http_access deny all
```

Method (HTTP GET, POST or CONNECT)

Μπορούμε να χρησιμοποιήσουμε το squid ακόμα για να εμποδίσουμε τους χρήστες να κάνουν posts σε διάφορα sites.

```
# Μη επιτρέποντας τη χρήση των search engines
acl Post_class method POST
acl myNet src 10.0.0.0/16
acl all src 0.0.0.0/0.0.0.0
# Σταματώντας αιτήσεις πριν επιτραπούν με αφετηρία το εύρος IP
διευθύνσεων μας
http_access deny Post_class
# Επιτρέποντας την πρόσβαση σε sites που δεν έχουν posts
http_access allow myNet
http_access deny all
```

Χρησιμοποιώντας το NCSA module πιστοποίησης

Για να χρησιμοποιήσουμε το NCSA module πιστοποίησης πρέπει να προσθέσουμε την εξής γραμμή στο configuration file:

```
authenticate_program /usr/local/squid/bin/ncsa_auth
/usr/local/squid/etc/passwd
```

Το squid χρησιμοποιεί την ίδια κωδικοποίηση με το password/shadowed file του συστήματος, έτσι μπορούμε να χρησιμοποιήσουμε και το ίδιο password file του συστήματος κάνοντάς το απλά copy.

Άλλοι Acl-operators

Εκτός από τους *http_access* και i*cp_icp_access* operators που αναφέραμε υπάρχουν και άλλοι πιο ειδικευμένοι τους οποίους θα αναφέρουμε επιγραμματικά.

- No_cache
- Ident_lookuo_access
- Miss_access
- Always_direct
- Never_direct
- Snmp_access
- Delay_access
- Broken_posts

7.8 Μέθοδος επιτάχυνσης

Πότε χρησιμοποιείται μέθοδος επιτάχυνσης

Η μέθοδος επιτάχυνσης δεν πρέπει να είναι ενεργοποιημένη εκτός και αν τη χρειαζόμαστε. Υπάρχει ένας ορισμένος αριθμός περιπτώσεων στις οποίες χρειάζεται και που αναφέρουμε παρακάτω.

Επιταχύνοντας έναν αργό διακομιστή

Το squid μπορεί βρεθεί μπροστά από έναν αργό διακομιστή, να κάνει cache στα δεδομένα του και να τα μεταφέρει στους πελάτες. Αυτό είναι πολύ χρήσιμο αν ο διακομιστής είναι πολύ αργός ή είναι πίσω από μία αργεί σύνδεση.

Αντικαθιστώντας έναν cache/web διακομιστή με το Squid

Αν βρισκόμαστε στη διαδικασία να αντικαταστήσουμε έναν cache/web διακομιστή τα μηχανήματα πελάτες πρέπει να ρυθμιστούν για να επικοινωνούν με την cache στο port 80. Αντί να ρυθμίσουμε ξανά όλα τα μηχανήματα μπορούμε να κάνουμε το squid το αναμένει για εισερχόμενες συνδέσεις στο port 80. Όταν το αυτό βρει ότι δέχτηκε μία web αίτηση θα την

προωθήσει στον πρωταρχικό διακομιστή προέλευσης, έτσι ώστε να συνεχίσει να λειτουργεί και σαν web και σαν cache διακομιστής.

Transparent Caching

Το squid μπορεί να ρυθμιστεί για να λειτουργεί με ένα «μαγικό» τρόπο παρεμβάλλοντας στις web αιτήσεις και κάνοντας cache σε αυτές. Εφόσον όμως αυτές οι αιτήσεις είναι σε μορφοποίηση διακομιστή πρέπει να πρέπει να μεταφραστούν σε μορφοποίηση αιτήσεων cache, κάτι το οποίο αναλύουμε περισσότερο σε επόμενο τομέα.

Ασφάλεια

Το squid μπορεί να τοποθετηθεί μπροστά από ένα μη ασφαλή διακομιστή για να τον προστατέψει. Μπορεί όχι μόνο να σταματήσει τους μη επιθυμητούς πελάτες να έχουν πρόσβαση σε αυτόν αλλά και να εμποδίσει κάποιους να παρεμβάλουν bugs στον κώδικα του διακομιστή.

Επιλογές ρυθμίσεων επιτάχυνσης

Η λίστα των ρυθμίσεων επιτάχυνσης είναι πολύ σύντομη και οι ρυθμίσεις τους σχετικά απλές.

Η επιλογή httpd_accel_host

Εδώ δηλώνουμε το όνομα του προς επιτάχυνση διακομιστή. Είναι πιθανό να έχουμε μόνο ένα διακομιστή προορισμού και έτσι θα έχουμε και μόνο μια δήλωση αυτής της επιλογής. Αν πρόκειται να επιταχύνουμε περισσότερους από ένα διακομιστές πρέπει να χρησιμοποιήσουμε τη λέξη *virtual* αντί για το hostname.

Η επιλογή httpd_accel_port

Οι αιτήσεις επιτάχυνσης μπορούν να προωθηθούν μόνο σε ένα port. Το squid θα συνδεθεί με το port που δηλώσαμε στο httpd_accel_port.

Η επιλογή httpd_accel_with_proxy

Αν χρησιμοποιήσουμε την επιλογή http_accel_host το squid θα σταματήσει να αναγνωρίζει cache αιτήσεις. Έτσι για να έχουμε λειτουργία και επιτάχυνσης και web cache πρέπει να θέσουμε αυτή την επιλογή σε κατάσταση *on*.

Η επιλογή httpd_accel_uses_host_header

Μια http αίτηση αποτελείται από τρεις μέρη: τον τύπο μεταφοράς, το path και το όνομα αρχείου που θα μεταφερθεί και τέλος την έκδοση του http. Το http πρωτόκολλο υποστηρίζει ένα ειδικό *host header,* το οποίο μεταφέρεται με κάθε εξερχόμενη http αίτηση. Αυτή η επικεφαλίδα κάνει το transparent caching ευκολότερο. Το squid μπορεί να μεταφράσει τη standard αυτή http αίτηση σε μία cache http αίτηση η οποία μπορεί να διαχειριστεί από τον κώδικα του squid. Αυτή η επιλογή πρέπει να είναι *on* όταν κάνουμε transparent caching.

Έλεγχος πρόσβασης

Έχοντας φτιάξει τις λίστες πρόσβασης με βάση τα προηγούμενα παραδείγματα θα έχετε κόψει την πρόσβαση σε μηχανήματα του εξωτερικού δικτύου. Η προς επιτάχυνση αίτηση μεταχειρίζεται ακριβώς όπως μια κανονική http αίτηση. Έτσι όσοι προέρχονται από το εξωτερικό δίκτυο θα απορριφθούν εφόσον acl κανόνες δεν επιτρέπουν την πρόσβαση στις IPs τους. Χρησιμοποιώντας τον *dst acl* τύπο μπορούμε να κάνουμε συγκεκριμένες εξαιρέσεις στις λίστες πρόσβασης για να επιτρέψουμε την πρόσβαση στον επιταχυμένο διακομιστή.

Πριν τις ρυθμίσεις επιτάχυνσης:

```
acl all src 0.0.0.0/0.0.0.0
acl myNet src 10.0.0.0/255.255.255.0
http_access allow myNet
http_access deny all
Μετά τις ρυθμίσεις επιτάχυνσης
# Ο απομακρυσμένος διακομιστής είναι ο 10.0.0.5, port 80
httpd_accel_host 10.0.0.5
httpd_accel_port 80
acl all src 0.0.0.0/0.0.0.0
acl myNet src 10.0.0.0/255.255.255.0
acl acceleratedHost dst 10.0.0.5
```

```
acl acceleratedPort port 80
# Οι αιτήσεις πρέπει να είναι και από τον σωστό host και από το
# σωστό port για να επιτραπούν
http_access allow acceleratedHost acceleratedPort
http_access allow myNet
http_access deny all
```

Παραδείγματα

Αντικαθιστώντας ένα web/cache διακομιστή

Η πιο κοινή χρίση του accelerator mode είναι για την αντικατάσταση ενός web/cache διακομιστή με το squid. Τότε το squid δέχεται αιτήσεις στο port 80 και τις μεταφέρει στον cache διακομιστή σε ένα διαφορετικό μηχάνημα. Εφόσον θέλουμε το squid να λειτουργεί και σαν επιταχυντής και σαν web διακομιστής πρέπει να χρησιμοποιήσουμε την επιλογή:

```
httpd_accel_with_proxy.
```

Προώθηση αιτήσεων σε διακομιστή

```
http_port 80
# προώθηση εισερχόμενων πακέτων στον localhost, port 8000
httpd_accel_host 127.0.0.1
acl acceleratedHost dst 127.0.0.1/255.255.255.255
httpd_accel_port 8000
acl acceleratedPort port 8000
httpd_accel_with_proxy on
acl all src 0.0.0.0/0.0.0.0
acl myNet src 10.0.0.0/255.255.255.0
# Δεν χρειάζεται να κάνουμε cache στον localhost γιατί είναι
σπατάλη χώρου
no_cache deny acceleratedHost
always_direct allow acceleratedHost
```

```
# Επιτρέπουμε αιτήσεις μόνο όταν είναι για το προς επιτάχυνση
μηχάνημα ΚΑΙ για το # σωστό port
http_access allow acceleratedHost acceleratedPort
http_access allow myNet
http_access deny all
```

Επιταχύνοντας αιτήσεις αργού διακομιστή

```
http_port 80
# προώθηση εισερχόμενων αιτήσεων στον 10.0.0.5, port 80
httpd_accel_host 10.0.0.5
acl acceleratedHost dst 10.0.0.5/255.255.255.255
httpd_accel_port 80
acl acceleratedPort port 8000
httpd_accel_with_proxy on
acl all src 0.0.0.0/0.0.0.0
acl myNet src 10.0.0.0/255.255.255.0
# Επιτρέπουμε αιτήσεις μόνο όταν είναι για το προς επιτάχυνση
μηχάνημα ΚΑΙ για το    # σωστό port
http_access allow acceleratedHost acceleratedPort
http_access allow myNet
http_access deny all
```

7.9 Transparent Caching

Το squid μπορεί να ρυθμιστεί για να λειτουργεί με διαφανή τρόπο (transparent). Σε αυτή τη λειτουργία οι περιηγητές των μηχανημάτων πελατών δεν χρειάζεται να ρυθμιστούν για να έχουν πρόσβαση στην cache, αλλά το squid με μη αντιληπτό τρόπο δέχεται τα πακέτα και κάνει cache τις αιτήσεις. Αυτό λύνει το μεγαλύτερο πρόβλημα με τη χρήση cache: να κάνουμε τους χρήστες να χρησιμοποιήσουνε τον cache διακομιστή. Αυτό είναι κάτι το οποίο ελάχιστοι χρήστες γνωρίζουν, ενώ κάποιοι άλλοι νομίζουν ότι χάνεται τον απόρρητο τους ή ότι καθυστερείτε η σύνδεση εφόσον ένας ακόμα host παρεμβαίνει ανάμεσα σε αυτούς και στο internet.

Προβλήματα

Όταν το squid λειτουργεί με διαφανή τρόπο η IP διεύθυνση αφετηρίας της σύνδεσης αλλάζει. Η αίτηση προέρχεται από τον cache διακομιστή αντί από το μηχάνημα πελάτη. Αυτό μπορεί να προκαλέσει ένα μπέρδεμα σε sites τα οποία χρησιμοποιούν πιστοποίηση μέσο IP διευθύνσεων. Εφόσον η cache αλλάζει τη IP διεύθυνση αφετηρίας της σύνδεσης μερικοί διακομιστές μπορεί να αρνηθούν την πρόσβαση στους χρήστες. Οι ISPs γενικά δεν αντιμετωπίζουν τέτοιο πρόβλημα εφόσον οι περισσότεροι dial-up χρήστες λαμβάνουν διαφορετική IP διεύθυνση κάθε φορά που συνδέονται. Οι ISPs όμως που χρησιμοποιούν transparent caching σε μισθωμένες γραμμές είναι πολύ πιθανό να έχουν προβλήματα. Πρέπει να είναι βέβαιοι ότι όλοι οι χρήστες γνωρίζουν τις τροποποιήσεις που πρέπει να κάνουν, πώς να κάνουν refresh τις σελίδες τους και πώς πρόκειται να αλλάξει η IP διεύθυνση αφετηρίας.

Διαδικασία

Βασικά στοιχεία δρομολογήσεων

Ας δούμε τώρα τι συμβαίνει όταν χρησιμοποιούμε transparency. Πρώτα πρέπει να μάθουμε τι γίνεται με τα IP πακέτα σε επίπεδο Ethernet.

Ένα Ethernet IP πακέτο περιέχει τέσσερις διευθύνσεις.

- Τη διεύθυνση προορισμού mac
- Τη διεύθυνση αφετηρίας mac
- Τη διεύθυνση IP προορισμού
- Τη διεύθυνση IP αφετηρίας

Όταν ένας host θέλει να επικοινωνήσει με ένα μηχάνημα που δεν βρίσκεται στο τοπικό δίκτυο χρησιμοποιεί ένα δρομολογητή για να βρει το path για το δίκτυο. Όταν ο πελάτης θέλει να στείλει ένα πακέτο μέσο του δρομολογητή στέλνει τη διεύθυνση προορισμού mac του πακέτου στο interface του δρομολογητή και τη διεύθυνση προορισμού IP στον host. Όταν ένας δρομολογητής δέχεται ένα πακέτο αποφασίζει σε ποιον host θα το στείλει με βάση τον πίνακα δρομολόγησης.

Ροή πακέτων με Transparent Caches

Το transparent caching παρακολουθεί τις συνδέσεις που προορίζονται για το port 80. Ο cache διακομιστής παρεμβάλει σε αυτά τα πακέτα και τα μεταφέρει στο squid. Όταν το squid στέλνει μια απάντηση στον πελάτη το λειτουργικό σύστημα αλλάζει τη διεύθυνση IP του πακέτου έτσι ώστε ο πελάτης νομίζει ότι είναι συνδεδεμένος στον διακομιστή στον οποίο έστειλε την αίτηση. Μπορούμε δηλαδή απλά να συνδέσουμε μια διαφανή cache σε ένα δίκτυο και να κάνει cache τις σελίδες. Ο cache διακομιστής πρέπει να είναι σε θέση όπου να μπορεί να ξεγελάει τα πακέτα.

Ας δούμε τώρα μια απλή εγκατάσταση transparent cache. Το μηχάνημα πελάτης (10.0.0.50) χειρίζεται το εσωτερικό interface (10.0.0.1) του cache διακομιστή σαν προεπιλεγμένη πύλη. Με αυτό τον τρόπο τα πακέτα φθάνουν στον cache διακομιστή πριν φθάσουν στον internet. Το φίλτρο παρακολουθεί τα πακέτα του port 80, τα μεταφέρει στο squid αλλά επιτρέπει σε όλα τα άλλα πακέτα να περάσουν από το στρώμα δρομολόγησης και να μεταφερθούν στην IP του router. Μόλις η σύνδεση πραγματοποιηθεί το squid πρέπει να επικοινωνήσει με τον πελάτη. Όταν το squid απαντήσει στον πελάτη ο kernel αυτόματα αλλάζει τη διεύθυνση IP του πακέτου έτσι ώστε να εμφανίζεται στον πελάτη ότι ο διακομιστής δρομολογεί τις εξωτερικές αιτήσεις.

Γενικά χρειάζεται να κάνουμε τέσσερα πράγματα για να πετύχουμε transparency:

- Διόρθωση των στρωμάτων δικτύου
- Φιλτράρισμα των κατάλληλων πακέτων
- Kernel transparency: Ανακατεύθυνση των συνδέσεων του port 80 στον squid
- Παραμετροποίηση του squid

Διάταξη δικτύου

Για να γίνει το φιλτράρισμα όλη η κίνηση πρέπει να περάσει μέσα από συσκευή φιλτραρίσματος. Σε μικρά δίκτυα ένα διακομιστής μπορεί να κάνει φιλτράρισμα αλλά σε πολλές άλλες περιπτώσεις χρειαζόμαστε μια δευτερεύουσα συσκευή για αυτή τη χρήση. Αυτές οι συσκευές μπορεί να είναι και οι δρομολογητές, μηχανήματα UNIX ή layer four switches.

Συμπεράσματα

Τα συμπέρασμα που προκύπτει από τη συμβίωση με το Linux είναι οι αστείρευτες δυνατότητες σε συνδυασμό με την πλήρη παραμετροποιησιμότητα του. Κάθε τι όμως έχει και αρνητικά και θετικά στοιχεία. Η πλήρης αυτή παραμετροποιησιμότητα δίνει στο χρήστη τη δυνατότητα να πετύχει αυτό ακριβώς που θέλει, εδώ όμως συνίσταται και το αρνητικό στοιχείο, ο χρήστης πρέπει να γνωρίζει ακριβώς τι θέλει να κάνει καθώς και τον τρόπο με τον οποίο θα πετύχει αυτό το αποτέλεσμα. Αυτό σημαίνει βαθιά γνώση του πρωτοκόλλου επικοινωνίας καθώς και τον διαδικασιών η οποίες είναι αλληλένδετες με τη λειτουργία που θέλει να επιτευχθεί. Όπως αναφέρθηκε η κατασκευή ενός firewall ξεκινάει με την απαγόρευση όλων των κινήσεων εντός και εκτός του δικτύου, αυτό σημαίνει μέγιστη δυνατή ασφάλεια αλλά ταυτόχρονα καθιστά το δίκτυο πλήρως άχρηστο να χειριστεί οποιαδήποτε πληροφορία, το επόμενο βήμα είναι ο καθορισμός της πολιτικής ασφάλειας και στη συνέχεια η πλήρης κατανόηση των αλληλένδετων διαδικασιών του κάθε πρωτοκόλλου που απαιτούνται για την επίτευξη αυτής της πολιτικής.

Από την αντίπερα όχθη, στα windows υπάρχουν έτοιμα πακέτα λογισμικού τα οποία αν παραμετροποιηθούν καταλλήλως μπορούν να μετατρέψουν τον υπολογιστή σε έναν firewall server πολύ πιο εύκολα και με λιγότερες γνώσεις. Το πρόβλημα σε αυτή τη περίπτωση είναι τα κατασκευάστηκα bugs τα οποία πιθανόν να έχει το ίδιο το software καθώς και τα κενά ασφαλείας τα οποία θα δημιουργεί το ίδιο το λειτουργικό το οποίο δεν μπορούμε να παραμετροποιήσουμε ανάλογα με τη χρήση του υπολογιστή όπως στην περίπτωση του Linux.

Βιβλιογραφία

Red Hat® Linux Firewalls

Bill McCarty | 2004, Wiley Publishing, Inc.

Linux Firewalls

Robert L. Ziegler | 2007, New Riders

Firewalls and Internet Security, Repelling the Wily Hacker

William R. Cheswick, Steven M. Bellovin, Aviel D. Rubin | 2007, Addition Wesley

TroubleShooting Linux® Firewalls

Michael Shinn, Scott Shinn | 2005, Addition Wesley

Squid, The Definitive Guide

Duane Wessels | 2004, O'Reilly

Red Hat® Certified Engineer Linux, RHCE

Michael Jang | 2004, McGraw Hill

Index