ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΑ SNIFFER, ΟΙ ΕΦΑΡΜΟΓΕΣ ΤΟΥΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΩΝ ΤΜΗΜΑΤΩΝ ΤΟΥΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΑΓΓΕΛΙΔΗ ΜΑΡΙΟΥ

Επιβλέποντες :

ΣΤΑΜΟΥΛΗΣ ΓΕΩΡΓΙΟΣ	ΤΣΟΜΠΑΝΟΠΟΥΛΟΥ ΠΑΝΑΓΙΩΤΑ
ΚΑΘΗΓΗΤΗΣ	ΑΝΑΠΛΗΡΩΤΗΣ ΚΑΘΗΓΗΤΡΙΑ

Εγκρίθηκε από την διμελή εξεταστική επιτροπή την

(Υπογραφή)

.....

ΚΥΡΙΟΣ ΕΠΙΒΛΕΠΩΝ

(Υπογραφή)

.....

ΔΕΥΤΕΡΕΥΩΝ ΕΠΙΒΛΕΠΩΝ

(Υπογραφή)

.....

ΑΓΓΕΛΙΔΗΣ ΜΑΡΙΟΣ

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστημίου Θεσσαλίας

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τους διδάσκοντες μου για τα εφόδια που μου παρείχαν όλα αυτά τα χρόνια των σπουδών μου και ιδιαιτέρως τον καθηγητή κ. Σταμούλη Γεώργιο για την υποστήριξη στην διπλωματική μου εργασία. Τέλος, ευχαριστώ τους γονείς μου και τους φίλους μου για την αμέριστη υποστήριξη τους στην προσπάθεια εκπλήρωσης των χρόνων φοίτησης μου.

Πίνακας περιεχομένων

ΕΙΣΑΓΩΓΗ

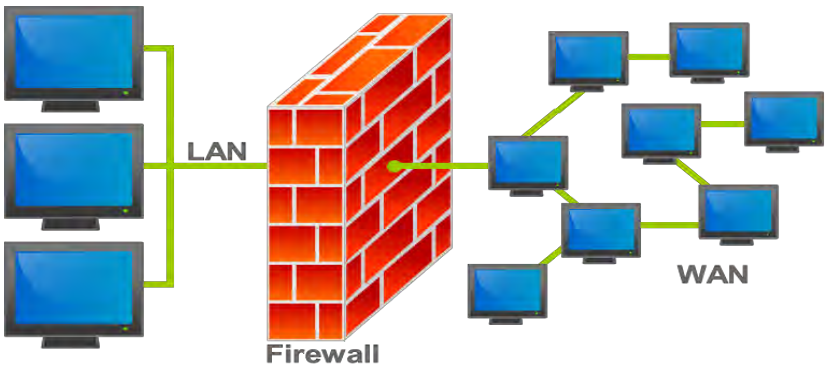
Μετά το πέρας της δεύτερης χιλιετίας η λέξη 'ίντερνετ', μια έννοια που φάνταζε σχεδόν άγνωστη στους περισσότερους ανθρώπους, σήμερα ίσως είναι ανάμεσα στις πιο κοινότυπες και καθημερινά χρησιμοποιούμενες λέξεις στο μεγαλύτερο μέρος της υφηλίου. Τί είναι όμως το ίντερνετ ή αλλιώς διαδίκτυο? Το διαδίκτυο [1] είναι ένα παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών, οι οποίοι χρησιμοποιούν καθιερωμένη ομάδα πρωτοκόλλων, η οποία συχνά αποκαλείται "TCP/IP" (αν και αυτή δεν χρησιμοποιείται από όλες τις υπηρεσίες του Διαδικτύου) για να εξυπηρετεί εκατομμύρια χρήστες καθημερινά σε ολόκληρο τον κόσμο. Οι διασυνδεδεμένοι ηλεκτρονικοί υπολογιστές ανά τον κόσμο, οι οποίοι βρίσκονται σε ένα κοινό δίκτυο επικοινωνίας, ανταλλάσσουν μηνύματα (ιντερνετικά πακέτα) με τη χρήση διαφόρων πρωτοκόλλων (τυποποιημένοι κανόνες επικοινωνίας), τα οποία υλοποιούνται σε επίπεδο υλικού και λογισμικού.

Με τον όρο ιντερνετικό πακέτο [2] εννοούμε την βασική μονάδα πληροφορίας που μεταφέρεται μέσα στα δίκτυα δηλαδή ένα καλά καθορισμένο μπλοκ απο bytes που αποτελείται από κεφαλίδα, δεδομένα και τρέιλερ. Μέσα σε ένα δίκτυο υπολογιστών τα δεδομένα που στέλνονται δεν μεταφέρονται με την μορφή μιας τυχαίας ακολουθίας απο συνεχόμενα bit αλλά το διαδικτυακό σύστημα διαιρεί τα δεδομένα σε μικρές ενότητες (πακέτα - packets) τις οποίες στέλνει μεμονωμένα. Επειδή η τεχνολογία αυτή χρησιμοποιείται απο τα δίκτυα υπολογιστών ονομάζονται και δίκτυα μεταγωγής πακέτων (packet switching networks). Απο τους δύο αυτούς ορισμούς μπορούμε εύκολα να συμπεράνουμε οτι με την κατάλληλη τεχνογνωσία των ιντερνετικών συστημάτων και με ορισμένα εργαλεία – προγράμματα τα πακέτα αυτά, που διακινούνται απο τον έναν χρήστη στον άλλον σε ένα δίκτυο υπολογιστών, μπορούνε να καταγραφθούν απο κάποιο υπολογιστικό σύστημα και στη συνέχεια να γίνει περαιτέρω ανάλυση αυτών ώστε να αντληθούν διάφορες πληροφορίες που μπορεί να είναι ευαίσθητες ή μη των εμπλεκώμενων χρηστών.

Με βάση τα προηγούμενα είναι απολύτως λογικό σε κάποιο χρήστη να δημιουργηθεί η απορία του κατα πόσο είναι ασφαλές να χρησιμοποιεί το διαδίκτυο και ποιά μέτρα θα πρέπει να λάβει έτσι ώστε να προβεί στη διαφύλαξη της ιδιωτικότητας του ώστε να μειώσει στο ελάχιστο τις περιπτώσεις μη θεμιτής παρακολούθησης των κινήσεων του στο διαδίκτυο και προστασίας υποκλοπής ευαίσθητων στοιχείων που μπορούν να του προκαλέσουν κακό.

Για τον λόγο αυτό ειδικά τα τελευταία χρόνια παρατηρούμε οτι δίνεται ιδιαίτερα μεγάλη έμφαση απο τις εταιρίες παραγωγής λογισμικών στην τελειοποίηση προγραμμάτων που αποτρέπουν την εύκολη παρακολούθηση των διαδικτυακών κινήσεων των χρηστών τα οποία ονομάζονται τείχη προστασίας ή

Firewall. Τα Firewall [3] μπορούν να αποτελέσουν ένα αποτελεσματικό μέσο προστασίας ενός τοπικού συστήματος ή δικτύου συστημάτων από απειλές ασφάλειας που βασίζονται σε δίκτυο, ενώ ταυτόχρονα παρέχουν πρόσβαση στον έξω κόσμο μέσω δικτύων ευρείας περιοχής και του Διαδικτύου.



Θεωρώντας την ιδιωτικότητα των χρηστών πολύτιμη στις μέρες μας εδώ και αρκετά χρόνια εκτός απο τα τείχη προστασίας που χρησιμοποιούνται απο τα υπολογιστικά συστήματα έχει ενσωματωθεί και η τέχνη της κρυπτογραφίας στα μηνύματα που διακινούνται μεταξύ τών χρηστών ώστε σε περίπτωση που κάποιο μήνυμα παραληφθεί απο κάποιον τρίτο χρήστη να είναι σχεδόν αδύνατο να διαβαστεί χωρίς να είναι γνωστός ο αλγόριθμος κρυπτογράφησης του αρχικού μηνύματος.

Παρά την τεράστια προσπάθεια των εταιριών για ελαχιστοποίηση της παραβίασης της ιδιωτικότητας των χρηστών αυτό δεν κατέστει ποτέ δυνατό στο 100% μέχρι σήμερα καθότι στον ιντερνετικό χώρο πάντοτε καραδωκούν κίνδυνοι και χρήστες (χάκερ) που με τις κατάλληλες γνώσεις μπορούνε σχεδόν πάντα να παρακάμψουν τα εμπόδια που διαθέτει ένα υπολογιστικό σύστημα στο διαδίκτυο και να χρησιμοποιήσουν προς όφελος τους ή μη, πληροφορίες που θα αποσπάσουν χωρίς να γίνουν αντιληπτοί τις περισσότερες φορές.

Αυτή η δυσκολία στην επίτευξη της πλήρους προστασίας της ιδιωτικότητας των χρηστών είναι λογικό να υπάρχει διότι ένα πρόγραμμα που έχει φτιαχτεί απο κάποιον άνθρωπο με γνώσεις σε υπολογιστικά συστήματα σίγουρα είναι εφικτό να μπορεί με κατάλληλά βήματα και εργαλεία να διασπαστεί και να παρακαμφθούν βασικές λειτουργίες του καθιστώντας το άχρηστο.

Φυσικά, αυτή η δυνατότητα παρακολούθησης της ιντερνετικής κυκλοφορίας των πληροφοριών δε θα πρέπει μονόπλευρα να φοβίζει τους χρήστες του ίντερνετ καθότι τα μέσα παρακολούθησης έχουν αναπτυχθεί εκτός απο τρίτους χρήστες, και απο υπηρεσίες όπως η αστυνομία όπου μέσω αυτών μπορούν να προλαμβάνουν και να διαλευκάνουν εγκληματικές και τρομοκρατικές ενέργειες που σκοπό έχουν την πλήξη πολιτικών ομάδων και κυβερνήσεων. Ένα παράδειγμα τέτοιου λογισμικού είναι το Carnivore [4] που μετονομάστηκε σε

DCS1000 και δημιουργήθηκε απο το FBI τον Οκτώβριο του 1997. Το σύστημα αυτό σχεδιάστηκε για τον έλεγχο των email και της ηλεκτρονικής επικοινωνίας χρησιμοποιώντας ένα προσαρμοζώμενο sniffer πακέτων όπου μπορούσε να ελέγχει όλη την ιντερνετική κυκλοφορία ενος συγκεκριμένου στοχευμένου χρήστη.

Απο όλα αυτά καταλαβαίνουμε πως απο τα πιο βασικά εργαλεία στην παρακολούθηση της ιντερνετικής κυκλοφορίας είναι αυτό της καταγραφής των πακέτων για ανάλυση και εξαγωγή πληροφοριών και για αυτό στη μεγαλύτερη έκταση της εργασίας αυτής θα ασχοληθούμε με τους αναλυτές πακέτων ή Sniffers και θα εξετάσουμε τον τρόπο με τον οποίο λειτουργούν και κατασκευάζονται.

ΚΕΦΑΛΑΙΟ 1

1.1 Ορισμός του αναλυτή πακέτων ή Sniffer.

Ένας αναλυτής πακέτων [5] (επίσης γνωστός και ως διαδικτυακός αναλυτής ή αναλυτής πρωτοκόλλων ή packet sniffer, ή για συγκεκριμένους τύπους δικτύων , ένα Ethernet sniffer ή Wireless sniffer) είναι ένα πρόγραμμα του υπολογιστή ή ένα μέρος hardware του υπολογιστή που μπορεί να λαμβάνει και να καταγράφει την κυκλοφορία που περνάει πάνω απο ένα ψηφιακό δίκτυο ή κομμάτι ενός δικτύου. Τα packet sniffer είναι δηλαδή εφαρμογές που χρησιμοποιούνται για να συλλέγουν τα πακέτα που κινούνται μέσα σε μια ζώνη δικτύου του πρωτόκολλου ελέγχου μετάδοσης/ιντερνετικού πρωτοκόλλου (TCP/IP). Καθώς τα δεδομένα ρέουν μέσα σε ένα δίκτυο ένα sniffer 'πιάνει' κάθε ένα πακέτο και εάν είναι απαιτούμενο αποκρυπτογραφεί τα δεδομένα του πακέτου δείχνοντας τις τιμές διαφόρων πεδίων του πακέτου είτε τα αναλύει ανάλογα πάντα με συγκεκριμένα κριτήρια, προσδιορισμούς και τεχνικές.

1.2 Η τεχνική του Sniffing και η χρήση της.

Η τεχνική του sniffing [6], όπως εύκολα μπορούμε να μαντέψουμε και απο τον παραπάνω ορισμό του packet sniffer, είναι μια μέθοδος αφαίρεσης κάθε πακέτου καθώς ρέει στο δίκτυο δηλαδή, είναι μια τεχνική στην οποία ο χρήστης αποκτά δεδομένα που ανήκουν σε άλλους χρήστες του δικτύου. Ένα packet sniffer μπορεί να λειτουργήσει ως ένα εργαλείο διαχειριστή ενός δικτύου ή για κακεντρεχείς σκοπούς όπως για παράδειγμα απο χάκερς. Με απλά λόγια βρίσκεται στην πρόθεση του χρήστη για το πώς θα χρησιμοποιήσει ένα τέτοιο εργαλείο απο τη στιγμή που το κατέχει. Για παράδειγμα οι διαχειριστές των δικτύων τα χρησιμοποιούν για τον έλεγχο και την πιστοποίηση της διαδικτυακής κυκλοφορίας καθώς μπορούν και να διεξάγουν έρευνες σε ορισμένες των περιπτώσεων. Απο την άλλη πλευρά για ευνόητους λόγους τα sniffer αποτελούν εξαιρετικά χρήσιμα εργαλεία για επίδοξους χάκερς που σκοπός τους είναι να αποσπάσουν ευαίσθητες πληροφορίες ή να εισβάλλουν σε κάποιο υπολογιστικό σύστημα όπως επίσης για επαγγελματίες που ασχολούνται με την ασφάλεια των δικτύων. Επίσης τα sniffer εκμεταλλεύονται χαρακτηριστικά απο διάφορες τεχνολογίες συνδέσεων δεδομένων (data-link), οι οποίες περιλαμβάνουν το Token Ring και κυρίως το Ethernet. Ο όρος token ring [7] ή δακτύλιος με σκυτάλη

αναφέρεται σε ένα τύπο τοπικού δικτύου υπολογιστών όπου στην πράξη υλοποιείται από ένα σύνολο υπολογιστών με συνδέσεις από σημείο σε σημείο. Μειονέκτημα του είναι ότι αν υπάρξει διακοπή στο καλώδιο τότε ο δακτύλιος πεθαίνει , αλλά αυτό το πρόβλημα λύνεται με την χρήση κέντρου καλωδίωσης. Αντίστοιχα το Ethernet [8] είναι το συνηθέστερα χρησιμοποιούμενο πρότυπο δικτύου υπολογιστών ενσύρματης τοπικής δικτύωσης υπολογιστών και αποτελείται από υλικό και λογισμικό που συνεργάζονται μαζί για τη διακίνηση ψηφιακών δεδομένων μεταξύ ενός δικτύου υπολογιστών. Μπορούμε να παρομοιάσουμε ένα sniffer με ένα κοριό τηλεφώνου. Κάποιος μπορεί να υποκλέψει μια τηλεφωνική γραμμή εάν θέλει να παρακολουθήσει κάποιο άλλο άτομο, έτσι και τα packet sniffer μπορούν να χρησιμοποιηθούν για να πάρουν δεδομένα άλλων χρηστών μέσα σε ένα δίκτυο υπολογιστών.

Ορισμένες απο τις χρήσεις των sniffing προγραμμάτων περιλαμβάνουν :

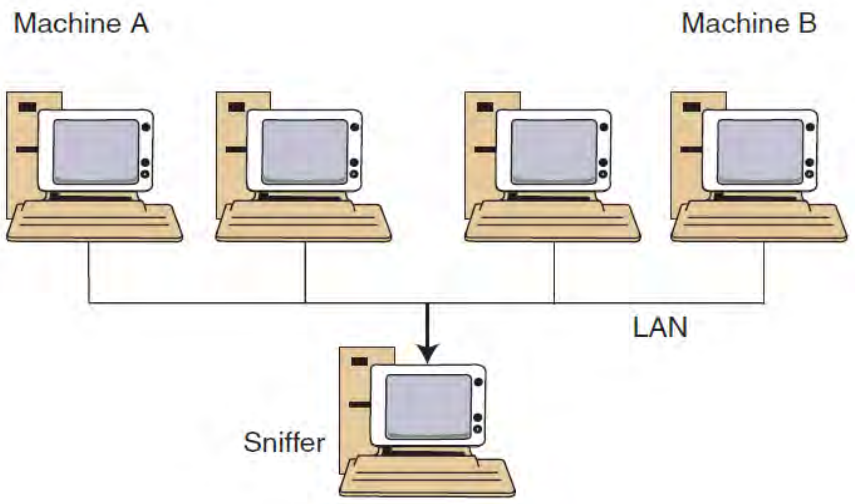
- Καταγραφή της διαδικτυακής κυκλοφορίας
- Επίλυση προβλημάτων επικοινωνίας όπως : να βρίσκουμε γιατί ο υπολογιστής Α δεν μπορεί να επικοινωνήσει με τον υπολογιστή Β (π.χ. Η επικοινωνία μπορεί να μην είναι εφικτή για διάφορους λόγους όπως ένα γενικό πρόβλημα σε κάποιο απο τα δύο συστήματα ή ένα πρόβλημα στο μέσο μετάδοσης)
- Ανάλυση της απόδοσης του δικτύου: Με αυτόν τον τρόπο μπορούν να ανακαλυφθούν τα σημεία συμφόρησης που υπάρχουν στο δίκτυο ή ένα μέρος του δικτύου όπου τα δεδομένα χάνονται (λόγω της συμφόρηση του δικτύου).
- Ανάκτηση ονομάτων χρηστών και κωδικούς πρόσβασης των ατόμων που πραγματοποιούν σύνδεση το δίκτυο.
- Προσδιορισμός πιθανών επιθέσεων ή κακόβουλης δραστηριότητας.
- Καθορισμός του ποιός ή τί χρησιμοποιεί το διαθέσιμο εύρος ζώνης.
- Προσδιορισμός των μέγιστων χρόνων χρήσης του δικτύου.
- Εύρεση μη ασφαλών εφαρμογών.

Υπάρχουν διάφοροι τύποι προγραμμάτων sniffer, συμπεριλαμβανομένων των ελεύθερων και εμπορικών. Κάθε ένα απο αυτά τα προγράμματα έχει σχεδιαστεί με διαφορετικό στόχο ανάλογα με την χρήση που θα ήθελε να έχει η εταιρία ή το

άτομο που το παρήγαγε .

1.3 Δυνατότητες των Sniffer.

Ένα packet sniffer 'ακούει' τα δεδομένα που καταφθάνουν σε μια κάρτα διεπαφής δικτύου (Network Interface Card – NIC). Παρ' όλα αυτά ένα sniffer δεν περιορίζεται μόνο σε κάποιο τοπικό δίκτυο (LAN) καθότι υπάρχουν παρόμοια sniffer που λειτουργούν σε ένα δίκτυο ευρείας περιοχής (WAN). Εάν ένα μηχάνημα βρίσκεται ανάμεσα σε κάποια άλλα δύο τυχαία συνδεδεμένα μηχανήματα A και B σε ένα δίκτυο WAN τότε το συγκεκριμένο μηχάνημα μπορεί να 'ακούσει' την κυκλοφορία των πακέτων που κινούνται ανάμεσα σε αυτά τα δύο μηχανήματα όπως φαίνεται και στην παρακάτω εικόνα που μας δείχνει ένα τυπικό packet sniffer το οποίο λειτουργεί σε τοπολογία Ethernet.



Πολλές συχνά χρησιμοποιούμενες εφαρμογές όπως Telnet, FTP, POP (Post Office Protocol που χρησιμοποιείται για e-mail) ακόμα και εφαρμογές web διαβιβάζουν τους κωδικούς και άλλα ευαίσθητα δεδομένα χωρίς καμία κρυπτογράφηση πράγμα που σημαίνει ότι οποιοσδήποτε κακόβουλος χρήστης μπορεί να χρησιμοποιήσει ένα sniffer και να συλλέξει εύκολα αυτά τα δεδομένα και τους κωδικούς. Οι χάκερς που καταφέρνουν και εισβάλλουν σε ένα σύστημα-στόχο συνήθως εγκαθιστούν μια εφαρμογή sniffer στο προσβαλλόμενο μηχάνημα η οποία παίζει τον ρόλο του 'φρουρού' για τον χάκερ καθώς συλλέγει πληροφορίες και κωδικούς τα οποία σώζει σε τοπικά αρχεία ή τα διαβιβάζει πίσω

στον χρήστη ο οποίος τα χρησιμοποιεί για να εισχωρήσει και σε περισσότερα υπολογιστικά μηχανήματα και δημιουργώντας μια αλυσιδωτή κακόβουλη επίθεση.

Μερικές φορές τα sniffer μπορούν να διατηρήσουν την ανωνυμία τους και είναι πιο δύσκολο να εντοπιστούν εάν εκτελεσθούν μέσα απο ένα άλλο μηχάνημα στο διαδίκτυο. Τα πακέτα μπορούν να καταγραφθούν απο κάποιο εξυπηρετητή στον οποίο έχει πρόσβαση το άτομο που εκτελεί την εφαρμογή του sniffer.

Σε ενσύρματα δίκτυα LAN, όπως τα Ethernet και Token Ring ανάλογα την δομή του δικτύου ένα sniffer μπορεί να λαμβάνει την κυκλοφορία των δεδομένων σε ένα μέρος ή και σε όλη την έκταση του δικτύου. Μερικές φορές για λόγους παρακολούθησης δικτύων είναι επιθυμητό να ελέγχονται όλα τα πακέτα δεδομένων σε ένα LAN χρησιμοποιώντας ένα διακόπτη δικτύου που καλείται θύρα παρακολούθησης και αντιγράφει όλα τα πακέτα που περνούν απο όλες τις θύρες του διακόπτη όταν τα συστήματα είναι συνδεδεμένα σε ένα διακόπτη παρακολούθησης.

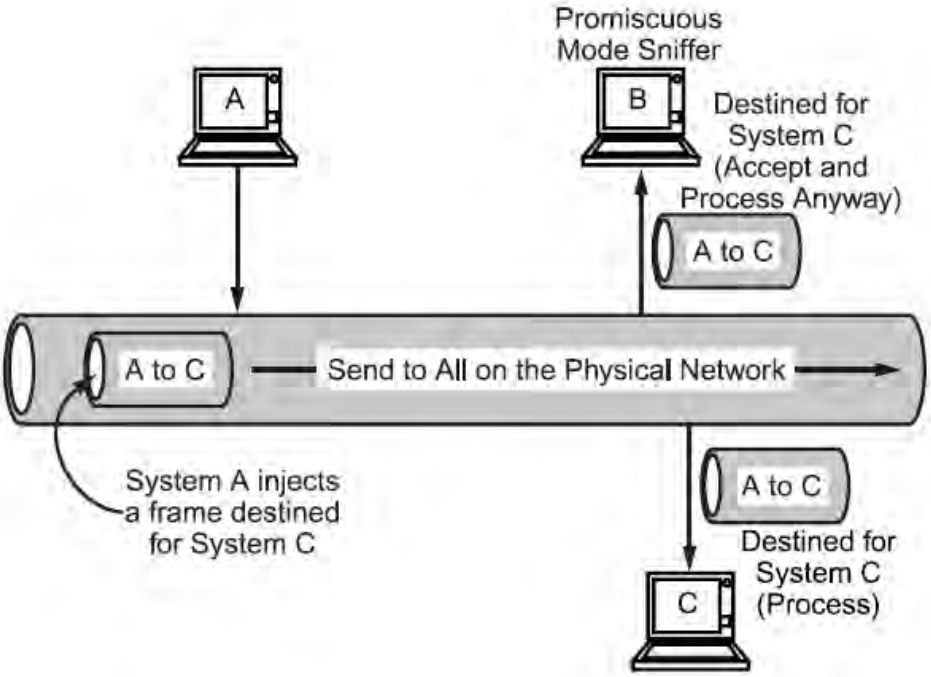
Οι υπολογιστές που λειτουργούν σε ένα περιβάλλον Ethernet όταν ένα σύστημα χρειάζεται να μεταδώσει δεδομένα, περιμένει μια ευκαιρία όταν κανένα άλλο σύστημα δεν μεταδίδει. Σε περίπτωση που δύο συστήματα επιχειρούν ταυτόχρονη εισαγωγή δεδομένων στο δίκτυο, τα ηλεκτρικά σήματα συγκρούονται μέσα στο καλώδιο. Αυτή η σύγκρουση αναγκάζει αμφότερα τα συστήματα να περιμένουν για ένα απροσδιόριστο χρονικό διάστημα πριν από την αναμετάδοση. Ο τομέας στον οποίο συμμετέχει μια ομάδα συστημάτων αναφέρεται μερικές φορές ως τομέας σύγκρουσης, επειδή όλα τα συστήματα στο τμήμα βλέπουν τις συγκρούσεις. Ως εκ τούτου, οποιοδήποτε σύστημα σε ένα κοινόχρηστο τμήμα δικτύου είναι κρυφό σε όλες τις επικοινωνίες σε αυτό το συγκεκριμένο τμήμα. Καθώς τα δεδομένα διασχίζουν ένα δίκτυο, όλες οι συσκευές στο δίκτυο αυτό μπορούν να δουν τα δεδομένα και να ενεργήσουν σε ορισμένες ιδιότητες πάνω σε αυτά για να παρέχουν υπηρεσίες επικοινωνίας. Ένα sniffer μπορεί να βρίσκεται σε θέσεις-κλειδιά του δικτύου και να ελέγχει τις λεπτομέρειες της ίδιας ροής δεδομένων. Σε ασύρματα δίκτυα WirelessLan-WLAN μπορούμε να συλλάβουμε την κυκλοφορία δεδομένων σε ένα συγκεκριμένο κανάλι ή και σε περισσότερα χρησιμοποιώντας περισσότερους ασύρματους προσαρμογείς.

1.4 Promiscuous Mode και Monitor Mode.

Κάθε διακομιστής σε ένα Ethernet - στην πραγματικότητα, κάθε διακομιστής Ethernet στον κόσμο - έχει μια μοναδική διεύθυνση Ethernet. Το Ethernet βασίζεται σε μια διεύθυνση ελέγχου πρόσβασης πολυμέσων (Media Access Control - MAC), συνήθως 48 bits που αντιστοιχούν σε μία κάρτα διεπαφής δικτύου (NIC). Οι διευθύνσεις Ethernet εκτυπώνονται τυπικά σε μια μορφή που ο άνθρωπος μπορεί να διαβάσει ως μια ακολουθία έξι αριθμών που χωρίζονται από ':'. Για παράδειγμα η 08:00:2b:e4:b1:02 είναι η ανθρώπινη αναγνώσιμη αναπαράσταση της διεύθυνσης :

00001000 00000000 00101011 11100100 10110001 00000010. Κάθε πακέτο που μεταδίδεται σε ένα Ethernet λαμβάνεται από κάθε προσαρμογέα συνδεδεμένο σε αυτό το Ethernet. Κάθε προσαρμογέας αναγνωρίζει τα πακέτα που απευθύνονται στη διεύθυνσή του και μεταδίδει μόνο αυτά τα πακέτα στον διακομιστή.

Ένα τυπικό sniffer λειτουργεί σε promiscuous mode. Promiscuous mode είναι μια κατάσταση στην οποία το NIC δέχεται όλα τα πακέτα, ανεξάρτητα από τη διεύθυνση MAC προορισμού τους. Η δυνατότητα υποστήριξης του promiscuous mode είναι απαραίτητη προϋπόθεση για τη χρήση του NIC ως sniffer, καθώς αυτό του επιτρέπει να συλλέγει και να διατηρεί όλα τα πακέτα που διασχίζουν το δίκτυο. Για τα sniffer που βασίζονται σε λογισμικό, το εγκατεστημένο NIC πρέπει να υποστηρίζει promiscuous mode για τη λήψη όλων των δεδομένων του τμήματος. Για τα sniffer που βασίζονται σε hardware - αποκλειστικό εξοπλισμό του οποίου ο μοναδικός σκοπός είναι να συλλέξει όλα τα δεδομένα - η εγκατεστημένη NIC πρέπει να υποστηρίζει τη λειτουργία promiscuous για να είναι αποτελεσματική. Η υλοποίηση ενός εξοπλισμού χωρίς τη δυνατότητα να λειτουργεί σε promiscuous mode θα ήταν σχεδόν άχρηστη διότι ο συγκεκριμένος εξοπλισμός δεν έχει σκοπό χρήσης τις φυσιολογικές επικοινωνίες μέσα σε κάποιο δίκτυο.

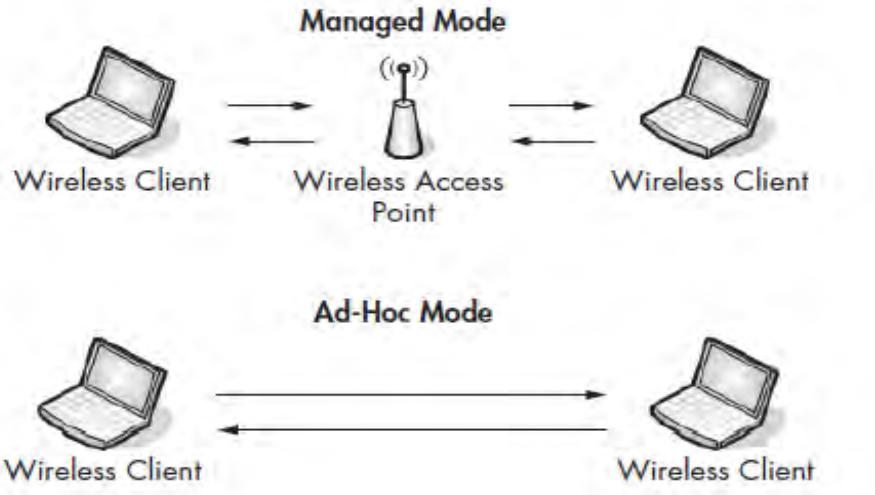


Υπάρχει μια περίπτωση όσων αφορά το ethernet κατά την οποία ένα σύστημα δεν γνωρίζει τη διεύθυνση MAC προορισμού ή χρειάζεται να επικοινωνήσει με όλα τα συστήματα του δικτύου. Μια εκπομπή συμβαίνει όταν ένα σύστημα εισάγει απλά ένα πακέτο που θα επεξεργαστεί κάθε άλλο σύστημα. Μια ενδιαφέρουσα πτυχή των μεταδόσεων είναι ότι ένα sniffer μπορεί να λειτουργεί σε nonpromiscuous mode και να έχει τη δυνατότητα να λαμβάνει εκπομπές από άλλα τμήματα. Παρόλο που οι πληροφορίες αυτές δεν είναι τυπικά ευαίσθητες, ένας εισβολέας μπορεί να χρησιμοποιήσει τις πληροφορίες για να μάθει περισσότερες πληροφορίες σχετικά με το δίκτυο στο οποίο επιδρά. [9]

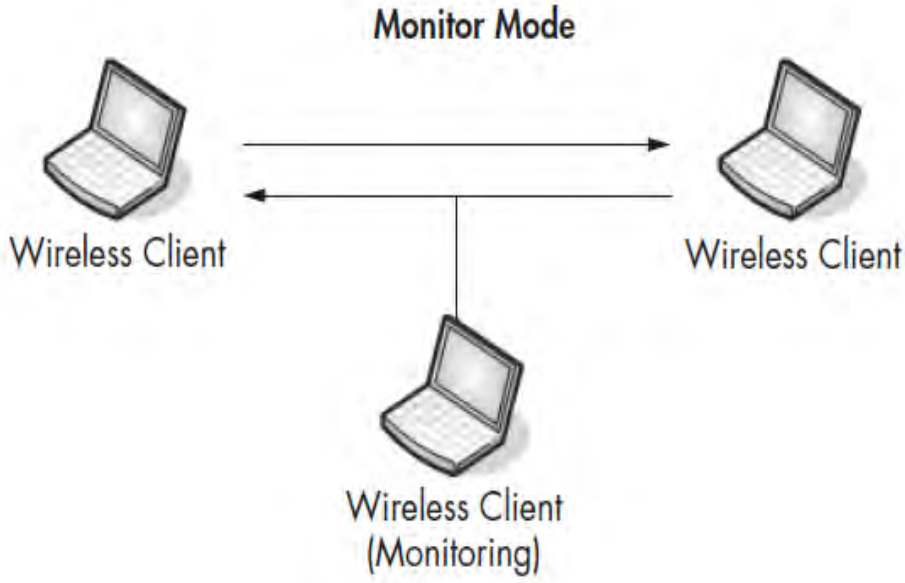
Εκτός όμως απο τα ενσύρματα δίκτυα υπάρχουν και τα ασύρματα δίκτυα τα γνωστά σε όλους wireless. Στην εν λόγω περίπτωση συναντούμε μια άλλη λειτουργία των ασύρματων συστημάτων την monitor mode. Η monitor mode ή αλλιώς RFMON (Radio Frequency MONitor) επιτρέπει σε έναν υπολογιστή με ελεγκτή διεπαφής ασύρματου δικτύου (WNIC) να παρακολουθεί όλη την κίνηση που λαμβάνεται από το ασύρματο δίκτυο. Σε αντίθεση με τη λειτουργία promiscuous , η οποία χρησιμοποιείται επίσης για τη συλλογή πακέτων, ο τρόπος παρακολούθησης επιτρέπει τη λήψη πακέτων χωρίς να χρειάζεται να συνδεθεί πρώτα με ένα σημείο πρόσβασης (Wireless Access Point-WAP) ή με ένα δίκτυο ad-hoc. Managed mode είναι η λειτουργία όπου ο ασύρματος υπολογιστής-πελάτης συνδέεται απευθείας με ένα σημείο ασύρματης πρόσβασης (WAP) ενώ

η λειτουργία adhoc χρησιμοποιείται όταν έχουμε μια ρύθμιση ασύρματου δικτύου στην οποία οι συσκευές συνδέονται απευθείας μεταξύ τους. [10]

Η



λειτουργία monitor χρησιμοποιείται όταν θέλουμε η ασύρματη κεραία μας να σταματήσει να μεταδίδει και να λαμβάνει δεδομένα και να συλλέγει μόνο τα πακέτα που πετούν στον αέρα. Να προσθέσουμε ότι η λειτουργία monitor ισχύει μόνο για ασύρματα δίκτυα, ενώ η promiscuous λειτουργία μπορεί να χρησιμοποιηθεί τόσο σε ενσύρματα όσο και σε ασύρματα δίκτυα.

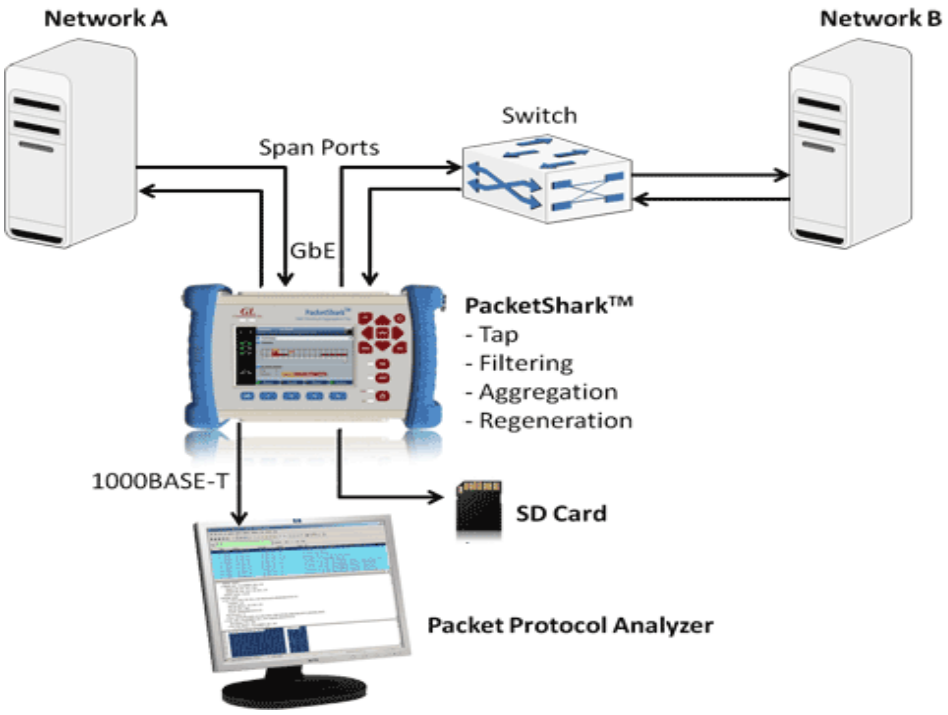


Χρήσιμο είναι να αναφερθεί ότι τα περισσότερα λειτουργικά συστήματα (συμπεριλαμβανομένων των Windows) δεν θα μας επιτραπεί να χρησιμοποιήσουμε μια κάρτα NIC σε promiscuous mode, εκτός αν έχουμε

αυξημένα προνόμια χρήστη. Εάν δεν μπορούμε νομίμως να λάβουμε αυτά τα προνόμια σε ένα σύστημα, είναι πολύ πιθανό ότι δεν θα πρέπει να εκτελούμε οποιαδήποτε διαδικασία συλλογής πακέτων στο συγκεκριμένο δίκτυο.

1.5 Τα βασικά μέρη ενός Sniffer.

- ***Υλικό (Hardware):*** Τα περισσότερα προϊόντα λειτουργούν από τυπικούς προσαρμογείς δικτύου, αν και μερικοί απαιτούν ειδικό υλικό. Αν χρησιμοποιούμε ειδικό υλικό, μπορούμε να αναλύσουμε σφάλματα υλικού όπως σφάλματα CRC, προβλήματα τάσης, καλωδιακά προβλήματα, dribbles, jitter, σφάλματα διαπραγμάτευσης κ.ο.κ.
- ***Πρόγραμμα Οδήγησης (Capture Driver):*** Αυτό είναι το πιο σημαντικό κομμάτι ενός sniffer. Καταγράφει την κίνηση του δικτύου μέσω του καλωδίου, τη φιλτράρει για τη συγκεκριμένη χρήση που τη θέλουμε και στη συνέχεια αποθηκεύει τα δεδομένα σε ένα buffer.
- ***Ενδιάμεση Μνήμη (Buffer):*** Μόλις ληφθούν τα πακέτα από το δίκτυο, αποθηκεύονται σε ένα buffer. Υπάρχουν δύο τρόποι καταγραφής των πακέτων: σύλληψη μέχρι να γεμίσει ο buffer, ή χρησιμοποιούμε το buffer με τη μέθοδο "round robin" όπου τα νεότερα δεδομένα αντικαθιστούν τα παλαιότερα δεδομένα.
- ***Ανάλυση σε πραγματικό χρόνο:*** Αυτό το χαρακτηριστικό κάνει κάποια μικρή ανάλυση των πακέτων καθώς αυτά βγαίνουν από το καλώδιο. Με αυτόν τον τρόπο είμαστε σε θέση να βρούμε τα προβλήματα επιδόσεων του δικτύου και τα σφάλματα κατά τη λήψη.
- ***Αποκρυπτογράφος (Decoder):*** Το μέρος αυτό εμφανίζει τα περιεχόμενα της κυκλοφορίας δικτύου με περιγραφικό κείμενο, έτσι ώστε να είναι σε θέση ένας αναλυτής δικτύου να καταλάβει τι συμβαίνει.
- ***Επεξεργασία / μετάδοση πακέτων:*** Ορισμένα προϊόντα περιέχουν λειτουργίες που μας επιτρέπουν να επεξεργαστούμε τα δικά μας πακέτα δικτύου και να τα μεταδώσουμε ξανά στο δίκτυο.


Network A
Network B
Switch
Span Ports
GbE
PacketShark™
- Tap
- Filtering
- Aggregation
- Regeneration
1000BASE-T
SD Card
Packet Protocol Analyzer

ΚΕΦΑΛΑΙΟ 2

Αφού είδαμε κάποια βασικά πράγματα για τους αναλυτές πακέτων, τον τρόπο λειτουργίας και την χρήση τους ήρθε η ώρα να γίνει αναφορά στην μορφή αυτών των πακέτων που διακινούνται και στα κυρίαρχα πρωτόκολλα επικοινωνίας που υπάρχουν την σημερινή εποχή στα διάφορα δίκτυα ανά τον κόσμο. Έτσι θα μας είναι πιο εύκολη η κατανόηση της ύπαρξης του κλάδου της ανάλυσης των πακέτων και θα είμαστε σε θέση να λύσουμε ποικίλες απορίες που είναι πιθανό να μας έχουν δημιουργηθεί καθώς και τους λόγους που οι διάφορες εταιρίες-κολοσσοί λογισμικών σπαταλούν υπέρογκα χρηματικά ποσά στον τομέα της προστασίας για τα υπολογιστικά τους συστήματα που έχουν πρόσβαση στο διαδίκτυο.

Η επικοινωνία μεταξύ δύο ατόμων καθίσταται δυνατή μέσω της αμοιβαίας συμφωνίας τους σε έναν κοινό τρόπο μεταφοράς ιδεών από το ένα άτομο στο άλλο. Κάθε άτομο πρέπει να γνωρίζει ακριβώς πώς να επικοινωνεί με το άλλο για να είναι αυτό επιτυχές. Η επικοινωνία μπορεί να έχει τη μορφή λεκτικής ή γραπτής γλώσσας. Θα μπορούσε να λάβει τη μορφή φυσικών χειρονομιών, όπως η νοηματική γλώσσα. Μπορεί ακόμη να γίνει μέσω εικόνων ή μουσικής. Ανεξάρτητα από τη μορφή της επικοινωνίας, είναι πρωταρχικής σημασίας ότι η έννοια ενός στοιχείου, για παράδειγμα μια λέξη, έχει το ίδιο νόημα και για τα δύο εμπλεκόμενα μέρη. Το μέσο που χρησιμοποιείται για την επικοινωνία είναι επίσης σημαντικό. Και τα δύο μέρη πρέπει να έχουν πρόσβαση στο ίδιο μέσο επικοινωνίας. Δεν μπορεί κανείς να μιλήσει με κάποιον άλλο μέσω τηλεφώνου, εάν μόνο ένα άτομο έχει τηλέφωνο.

Με τους υπολογιστές, η επικοινωνία μέσω δικτύων καθίσταται δυνατή από τα γνωστά ως πρωτόκολλα. Ένα πρωτόκολλο είναι μια σαφώς καθορισμένη μορφή μηνύματος. Η μορφή μηνύματος ορίζει τι σημαίνει καθετί σε κάποια θέση στο μήνυμα. Μια πιθανή μορφή μηνύματος θα μπορούσε να ορίσει τα πρώτα 4 bits ως τον αριθμό έκδοσης (version number), τα επόμενα 4 bits ως μήκος της κεφαλίδας (header length) και στη συνέχεια 8 bits για την υπηρεσία που χρησιμοποιείται. Όσο οι δύο υπολογιστές συμφωνούν με αυτή τη μορφή επικοινωνίας τότε και η επικοινωνία μπορεί να πραγματοποιηθεί. Οι επικοινωνίες δικτύων χρησιμοποιούν περισσότερα από ένα πρωτόκολλα. Τα σύνολα των πρωτοκόλλων που χρησιμοποιούνται μαζί είναι γνωστά ως πακέτα πρωτοκόλλων (protocol suites) ή πρωτόκολλα με στρώματα (layered protocols).

2.1 Internet Protocol (IP)

Το πρωτόκολλο Internet ή IP, είναι ένα πρωτόγονο και ανεξάρτητο από την εφαρμογή πρωτόκολλο που χρησιμοποιείται για την διευθυνσιοδότηση και τη δρομολόγηση πακέτων δεδομένων εντός ενός δικτύου. Είναι το "IP" στο TCP / IP, το πακέτο πρωτοκόλλου που αποτελεί και τον ορισμό του ίντερνετ. Προορίζεται για χρήση σε ένα σχετικά επίπεδο, πλεγματικό, μεταδιδόμενο, χωρίς σύνδεση όπου χρησιμοποιείται μεταγωγή πακέτων δίκτυο όπως το ίντερνετ.

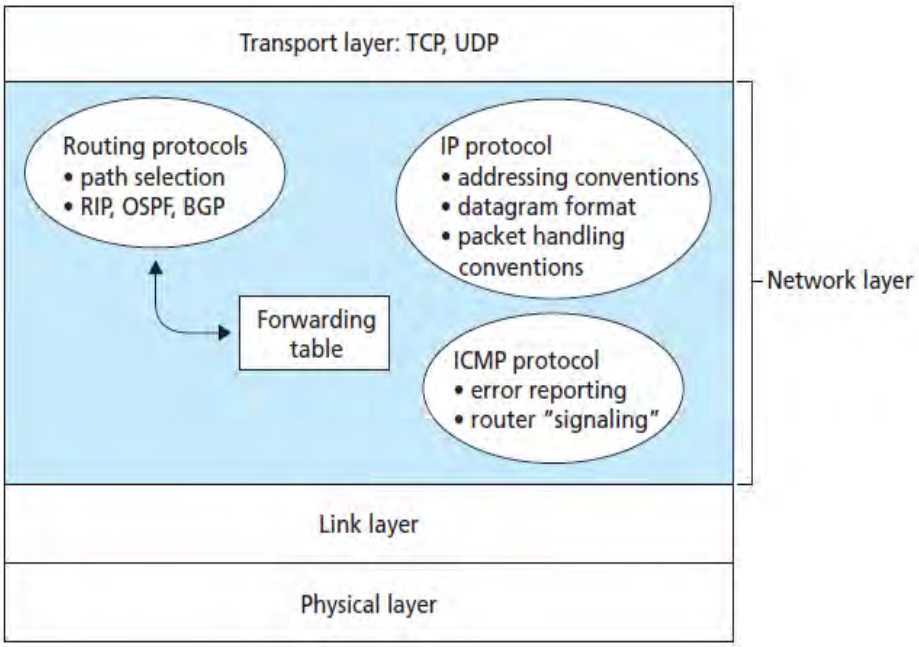
Εάν μας επιτρέπεται να παρομοιάσουμε το πρωτόκολλο IP τότε αυτό θα ήταν με μια ταχυδρομική κάρτα που στέλνεται τον 18ο αιώνα. Ο αποστολέας έγραψε το μήνυμα στη μία πλευρά της κάρτας και την διεύθυνση αποστολής και επιστροφής αντίστοιχα από την άλλη. Τότε το έδωσε σε κάποιον που κατευθυνόταν στη γενική κατεύθυνση του αποδέκτη. Φυσικά, το συγκεκριμένο μήνυμα δεν ήταν εμπιστευτικό και όλοι όσοι το χειριζόταν θα μπορούσαν να το διαβάσουν και θα μπορούσαν στην χείριστη των περιπτώσεων να κάνουν μια μη ανιχνευμένη αλλαγή σε αυτό.

Το IP είναι ένα πρωτόκολλο που δεν εγγυάται την παράδοση μηνυμάτων και δεν παρέχει κανένα στοιχείο για την επιτυχή παράδοση του. Αποτελεί πρωτόκολλο χωρίς έλεγχο καθότι ο δέκτης δεν γνωρίζει αν έλαβε ολόκληρο το επιθυμητό μήνυμα και αν είναι σωστό ή όχι. Οι διευθύνσεις είναι αναξιόπιστες και ως εκ τούτου ο αποστολέας δεν μπορεί να είναι σίγουρος ότι το μήνυμα θα πάει μόνο εκεί όπου προτίθεται ή ακόμα και μέσα στο χρονικό περιθώριο που σκοπεύει να παραδοθεί. Ο δέκτης δεν μπορεί να είναι σίγουρος ότι το μήνυμα προήλθε από τη διεύθυνση που καθορίστηκε ως η διεύθυνση επιστροφής στο πακέτο. Το πρωτόκολλο δεν παρέχει έλεγχο ή απόκρυψη. Εάν η εφαρμογή απαιτεί τα συγκεκριμένα χαρακτηριστικά θα πρέπει αυτά να προσδιορίζονται κάπου αλλού, συνήθως σε ένα υψηλότερο (δηλαδή κάπου 'πιο κοντά' στην εφαρμογή) στρώμα πρωτοκόλλου.

Ένα άλλο χαρακτηριστικό του IP είναι ότι καθορίζει τις διευθύνσεις της συσκευής υλικού αποστολής ή λήψης αλλά για την περίπτωση που αυτή η συσκευή υποστηρίζει πολλαπλές εφαρμογές, το IP δεν καθορίζει για ποια από αυτές προορίζονται.

Οι περισσότεροι απο τους ανθρώπους που αποτελούν χρήστες του διαδικτύου σίγουρα θα έχουν παρατηρήσει πως οι διευθύνσεις IP είναι της μορφής 'xxx.xxx.xxx.xxx' όπου x αριθμητικό απο το 0 έως το 9. Παρόλο που η ασφάλεια δεν είναι ένα απο τα δυνατά χαρακτηριστικά της IP, το πρωτόκολλο αυτό είναι υπεύθυνο για μεγάλο μέρος της επιτυχίας του παγκόσμιου διαδικτύου. Είναι γρήγορο και απλό και στην πράξη οι περιορισμοί ασφαλείας του IP απλώς δεν έχουν μεγάλη σημασία καθώς οι εφαρμογές βασίζονται σε πρωτόκολλα υψηλότερου επιπέδου για την ασφάλεια. [9] Υπάρχουν δύο εκδόσεις του IP που χρησιμοποιούνται σήμερα. Θα εξετάσουμε πρώτα την ευρέως αναπτυχθείσα έκδοση πρωτοκόλλου IP πρωτόκολλο σειρά 4, η οποία συνήθως αναφέρεται απλώς ως Ipv4 και μετά θα αναφερθούμε στην έκδοση που έπεται να διαδεχθεί

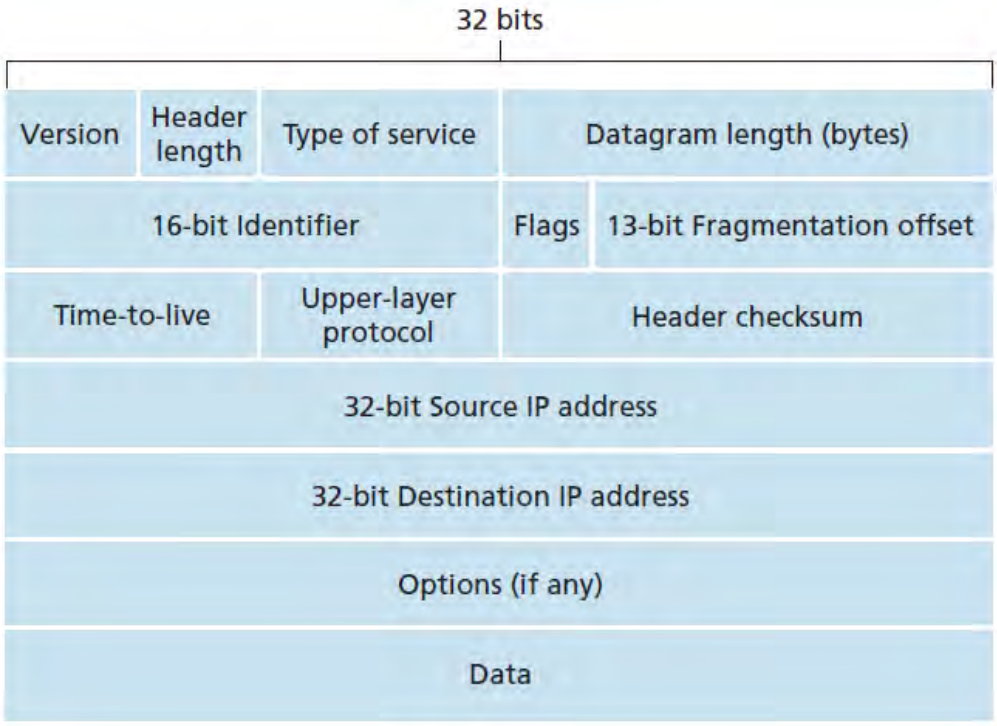
την Ipv4 η οποία ονομάζεται Ipv6.



2.1.1 IPv4

Το δεδομενόγραμμα (datagram) είναι το πακέτο των πληροφοριών IP που πρέπει να διαβιβάζονται μέσω του διαδικτυακού πρωτοκόλλου. Καταλαμβάνει τα 46 έως 1500 bytes δεδομένων σε ένα πακέτο ethernet και ανάλογα με το μέγεθός του μπορεί να διαμοιραστεί σε περισσότερα από δύο ή περισσότερα ξεχωριστά πακέτα ethernet. [11]Μπορεί να σκεφτόμαστε ότι τίποτα δεν θα μπορούσε να είναι πιο ασήμαντο από την μορφή και την σημασιολογία των bits ενός πακέτου. Παρόλα αυτά, το πακέτο διαδραματίζει κυρίαρχο ρόλο στο Internet και κάθε φοιτητής στον τομέα της πληροφορικής είτε επαγγελματίας μηχανικός πρέπει να το παρατηρήσει, να το κατανοήσει και να το κατακτήσει γνωστικά. Έχωντας τις απαραίτητες γνώσεις θα είναι σε θέση να κατανοήσει τον τρόπο λειτουργίας του διαδικτύου εις βάθος, θα μπορεί να επιλύει προβλήματα γνωρίζοντας ή προβλέπωντας τα αίτια τους καθώς και θα κατέχει προβάδισμα στον τομέα της προστασίας και ασφάλειας συστημάτων.

Για να κατανοήσουμε καλύτερα τα πράγματα θα αναλύσουμε το δεδομενόγραμμα του Ipv4 όπως φαίνεται στην παρακάτω εικόνα.



Επεξήγηση των πεδίων του δεδομενογράμματος:

- ***Version:*** Αυτά τα 4 bits καθορίζουν την έκδοση πρωτοκόλλου IP του datagram. Παρατηρώντας τον αριθμό έκδοσης, ο δρομολογητής (router) μπορεί να καθορίσει τον τρόπο διερμηνείας του υπολοίπου του πακέτου. Οι διαφορετικές εκδόσεις του IP χρησιμοποιούν διαφορετικό datagram.

- ***Header length:*** Επειδή ένα datagram IPv4 μπορεί να περιέχει έναν μεταβλητό αριθμό επιλογών (που περιλαμβάνονται στην κεφαλίδα δεδομένων του datagram), αυτά τα 4 bits είναι απαραίτητα για να προσδιορίσουν πού στην πραγματικότητα αρχίζουν τα δεδομένα μέσα στο datagram. Τα περισσότερα datagrams IP δεν περιέχουν επιλογές, οπότε το τυπικό πακέτο IP έχει μια κεφαλίδα 20 byte.

- ***Type of service:*** Τα bits τύπου υπηρεσίας (TOS) συμπεριελήφθησαν στην κεφαλίδα IPv4 για να μπορούν να διακριθούν μεταξύ τους διαφορετικοί τύποι datagrams IP (π.χ. datagrams που απαιτούν ιδιαίτερα καθυστέρηση, υψηλή απόδοση ή αξιοπιστία). Για παράδειγμα, μπορεί να είναι χρήσιμη η διάκριση των datagrams σε πραγματικό χρόνο (όπως αυτά που χρησιμοποιούνται από μια εφαρμογή IP τηλεφωνίας) από μη πραγματικού χρόνου κυκλοφορίας (π.χ. το FTP). Το συγκεκριμένο επίπεδο υπηρεσίας που πρέπει να παρέχεται είναι ένα θέμα πολιτικής που καθορίζεται από το διαχειριστή του δρομολογητή.

- ***Datagram length:*** Αυτό είναι το συνολικό μήκος του πακέτου IP (η κεφαλίδα συν τα δεδομένα) που μετριέται σε bytes. Δεδομένου ότι το πεδίο αυτό έχει μήκος 16 bits, το θεωρητικό μέγιστο μέγεθος του datagram IP είναι 65.535 byte. Ωστόσο, τα datagrams είναι σπάνια μεγαλύτερα από 1.500 bytes.

- ***Identifier, flags, fragmentation offset:*** Αυτά τα τρία πεδία έχουν να κάνουν με τον αποκαλούμενο κατακερματισμό IP. Είναι ενδιαφέρον ότι η νέα έκδοση του IP, IPv6, δεν επιτρέπει κατακερματισμό σε δρομολογητές.

- ***Time-to-live:*** Το πεδίο χρόνος-ζωής (TTL) συμπεριλαμβάνεται για να διασφαλιστεί ότι τα πακέτα δεδομένων δεν κυκλοφορούν για πάντα (εξαιτίας, για παράδειγμα, ενός μακρόβιου βρόχου δρομολόγησης) στο δίκτυο. Αυτό το πεδίο μειώνεται κατά ένα κάθε φορά που το datagram επεξεργάζεται από ένα δρομολογητή. Αν το πεδίο TTL φτάσει στο 0, το datagram πρέπει να απορριφθεί.

- ***Protocol:*** Αυτό το πεδίο χρησιμοποιείται μόνο όταν ένα πακέτο δεδομένων IP φτάσει στον τελικό του προορισμό. Η τιμή αυτού του πεδίου υποδεικνύει το συγκεκριμένο πρωτόκολλο στρώματος μεταφοράς στο οποίο πρέπει να περάσει το τμήμα δεδομένων αυτού του πακέτου. Αξίζει να σημειωθεί ότι ο αριθμός πρωτοκόλλου στο πακέτο δεδομένων IP έχει ένα ρόλο που είναι ανάλογος του ρόλου του πεδίου αριθμού θύρας στο τμήμα μεταφοράς. Ο αριθμός πρωτοκόλλου είναι η 'κόλλα' που συνδέει το δίκτυο και τα στρώματα μεταφοράς μαζί, ενώ ο αριθμός θύρας είναι η κόλλα που συνδέει τα στρώματα μεταφοράς και εφαρμογής μαζί.

- ***Header checksum:*** Το ελεγκτικό άθροισμα κεφαλίδας βοηθά ένα δρομολογητή στο να ανιχνεύει σφάλματα δυαδικών ψηφίων σε ένα ληφθέν IP datagram. Το ελεγκτικό άθροισμα κεφαλίδας υπολογίζεται με επεξεργασία κάθε 2 bytes στην κεφαλίδα ως ενός αριθμού και αθροίζοντας αυτούς τους αριθμούς χρησιμοποιώντας αριθμητική συμπλήρωση 1s. Ένας δρομολογητής υπολογίζει το άθροισμα ελέγχου κεφαλίδας για κάθε ληφθέν IP datagram και ανιχνεύει μια κατάσταση σφάλματος εάν το άθροισμα ελέγχου που μεταφέρεται στην κεφαλίδα της δέσμης δεδομένων δεν είναι ίσο με το υπολογισμένο άθροισμα ελέγχου. Οι δρομολογητές τυπικά απορρίπτουν τα datagrams για τα οποία έχει εντοπιστεί σφάλμα. Να σημειώσουμε ότι το άθροισμα ελέγχου πρέπει να ξαναυπολογιστεί και να αποθηκευτεί ξανά σε κάθε δρομολογητή, καθώς μπορεί να αλλάξει το πεδίο TTL και ενδεχομένως και το πεδίο επιλογών.

- ***Source and destination IP addresses:*** Όταν μια πηγή δημιουργεί ένα datagram, εισάγει τη διεύθυνση IP στο πεδίο διεύθυνσης IP προέλευσης και εισάγει τη διεύθυνση του τελικού προορισμού στο πεδίο διεύθυνσης IP προορισμού. Συχνά ο αρχικός διακομιστής καθορίζει τη διεύθυνση προορισμού μέσω μιας αναζήτησης DNS.

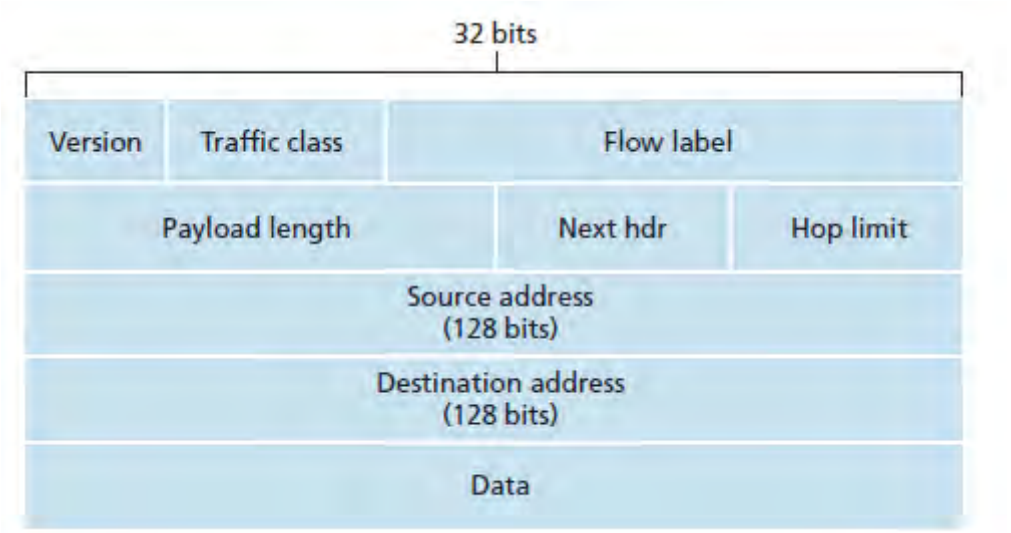
- ***Options:*** Τα πεδία επιλογών επιτρέπουν την επέκταση μιας κεφαλίδας IP. Οι επιλογές επικεφαλίδων προορίζονταν να χρησιμοποιούνται σπανίως και ως εκ τούτου πάρθηκε η απόφαση να εξοικονομηθούν πρόσθετες επιβαρύνσεις με το να μην συμπεριλαμβάνονται πληροφορίες στα πεδία επιλογών σε κάθε κεφαλίδα του datagram. Ωστόσο, η απλή ύπαρξη επιλογών περιπλέκει περισσότερο τα πράγματα δεδομένου ότι οι κεφαλίδες των πακέτων δεδομένων μπορούν να έχουν μεταβλητό μήκος και έτσι δεν μπορεί κανείς να προσδιορίσει εκ των προτέρων το σημείο εκκίνησης του πεδίου δεδομένων. Επίσης, δεδομένου ότι ορισμένα datagrams ενδέχεται να απαιτούν επεξεργασία των επιλογών και άλλα όχι, ο χρόνος που απαιτείται για την επεξεργασία ενός πακέτου IP σε ένα δρομολογητή μπορεί να ποικίλει σημαντικά. Αυτές οι εκτιμήσεις καθίστανται ιδιαίτερα σημαντικές για την επεξεργασία IP σε δρομολογητές και κεντρικούς υπολογιστές υψηλής απόδοσης.

- ***Data (payload):*** Τέλος, φτάνουμε στο τελευταίο και πιο σημαντικό πεδίο - το raison d'être για το datagram δηλαδή τον βασικό λόγο ύπαρξης του! Στις περισσότερες περιπτώσεις, το πεδίο δεδομένων του πακέτου δεδομένων IP περιέχει το τμήμα στρώματος μεταφοράς (TCP ή UDP) που πρέπει να παραδοθεί στον προορισμό. Ωστόσο, το πεδίο δεδομένων μπορεί να μεταφέρει άλλους τύπους δεδομένων, όπως μηνύματα ICMP.

2.1.2 IPv6

Στις αρχές της δεκαετίας του 1990, η ομάδα εργασίας του Internet Engineering ξεκίνησε μια προσπάθεια να αναπτύξει έναν διάδοχο του πρωτοκόλλου IPv4. Ένα πρωταρχικό κίνητρο για αυτή την προσπάθεια ήταν η συνειδητοποίηση ότι ο χώρος διεύθυνσης IP 32-bit αρχίζει να εξαντλείται, με νέα υποδίκτυα και κόμβους IP να συνδέονται στο Διαδίκτυο (έχοντας εκχωρηθεί μοναδικές διευθύνσεις IP) με ρυθμό ανεπανάληπτο . Για να ανταποκριθεί σε αυτή την ανάγκη για ένα μεγάλο χώρο διεύθυνσης IP, αναπτύχθηκε ένα νέο πρωτόκολλο IP, το IPv6. Οι σχεδιαστές του IPv6 έλαβαν επίσης την ευκαιρία να τροποποιήσουν και να ενισχύσουν άλλες πτυχές του IPv4, με βάση τη συσσωρευμένη λειτουργική εμπειρία με το Ipv4.

Η μορφή του datagram IPv6 εμφανίζεται ακριβώς παρακάτω.



Οι πιο σημαντικές αλλαγές που εισήχθησαν στο IPv6 είναι εμφανείς στη μορφή του δεδομενογράμματος:

1. ***Διευρυμένες δυνατότητες διευθυνσιοδότησης:*** Το IPv6 αυξάνει το μέγεθος της διεύθυνσης IP από 32 σε 128 bits. Αυτό εξασφαλίζει ότι ο κόσμος δεν θα ξεμείνει απο διευθύνσεις IP. Τώρα, κάθε κόκκος άμμου στον πλανήτη μπορεί να είναι IP-διευθυνσιοδοτούμενος. Εκτός από τις διευθύνσεις unicast και multicast, το IPv6 εισήγαγε έναν νέο τύπο διεύθυνσης, που ονομάζεται διεύθυνση anycast, η οποία επιτρέπει την παράδοση ενός datagram σε οποιαδήποτε ομάδα διακομιστών.

2. ***Μια βελτιωμένη επικεφαλίδα 40 byte:*** Όπως αναλύεται παρακάτω, πολλά πεδία IPv4 έχουν καταργηθεί ή έχουν γίνει προαιρετικά. Η προκύπτουσα κεφαλίδα καθορισμένου μήκους 40 byte επιτρέπει την ταχύτερη επεξεργασία του πακέτου IP. Μια νέα κωδικοποίηση των επιλογών επιτρέπει την πιο ευέλικτη επεξεργασία τους.

3. ***Επισήμανση ροής και προτεραιότητα:*** Το IPv6 έχει έναν αόριστο ορισμό μιας ροής. Για παράδειγμα, η μετάδοση ήχου και βίντεο μπορεί να είναι μπορεί να χαρακτηριστεί ως ροή. Από την άλλη πλευρά, οι πιο παραδοσιακές εφαρμογές, όπως η μεταφορά αρχείων και το ηλεκτρονικό ταχυδρομείο, ενδέχεται να μην αντιμετωπίζονται ως ροές. Είναι πιθανό ότι η κίνηση που μεταφέρεται από ένα χρήστη υψηλής προτεραιότητας (για παράδειγμα, κάποιος που πληρώνει για καλύτερη εξυπηρέτηση της κυκλοφορίας του) μπορεί επίσης να αντιμετωπιστεί ως ροή. Ωστόσο, είναι σαφές ότι οι σχεδιαστές του IPv6 προβλέπουν την ενδεχόμενη ανάγκη να είναι δυνατή η διαφοροποίηση μεταξύ των ροών, ακόμη και αν δεν έχει ακόμη προσδιοριστεί η ακριβής σημασία μιας ροής. Η κεφαλίδα IPv6

διαθέτει επίσης ένα πεδίο κυκλοφορίας της τάξης των 8 bit. Αυτό το πεδίο, όπως το πεδίο TOS στο IPv4, μπορεί να χρησιμοποιηθεί για να δώσει προτεραιότητα σε ορισμένα datagrams μέσα σε μια ροή ή μπορεί να χρησιμοποιηθεί για να δώσει προτεραιότητα σε datagrams από ορισμένες εφαρμογές.

Όπως σημειώθηκε παραπάνω, μια σύγκριση των σχημάτων αποκαλύπτει την απλούστερη, πιο εξορθολογισμένη δομή του datagram IPv6. Τα ακόλουθα πεδία ορίζονται στο Ipv6:

- ***Version:*** Αυτό το πεδίο 4-bit αναγνωρίζει τον αριθμό έκδοσης IP. Δεν αποτελεί έκπληξη το γεγονός ότι το IPv6 έχει τιμή 6 στο πεδίο αυτό. Σημειώστε ότι η τοποθέτηση ενός 4 σε αυτό το πεδίο δεν δημιουργεί ένα έγκυρο πακέτο Ipv4.

- ***Traffic class:*** Αυτό το πεδίο 8-bit είναι παρόμοιο σε πνεύμα με το πεδίο TOS που είδαμε στο Ipv4.

- ***Flow label:*** Όπως αναφέρθηκε παραπάνω, αυτό το πεδίο 20-bit χρησιμοποιείται για την αναγνώριση ροής των datagrams.

- ***Payload length:*** Αυτή η τιμή 16-bit αντιμετωπίζεται ως ένας μη-υπογεγραμμένος ακέραιος που δείχνει τον αριθμό των bytes στο πακέτο IPv6 που ακολουθεί την κεφαλίδα πακέτου δεδομένων σταθερού μήκους 40 bytes.

- ***Next header:*** Αυτό το πεδίο προσδιορίζει το πρωτόκολλο στο οποίο θα παραδοθούν τα περιεχόμενα (πεδίο δεδομένων) αυτού του datagram (για παράδειγμα, στο TCP ή UDP). Το πεδίο χρησιμοποιεί τις ίδιες τιμές με το πεδίο πρωτοκόλλου στην κεφαλίδα Ipv4.

- ***Hop limit:*** Τα περιεχόμενα αυτού του πεδίου μειώνονται κατά ένα από κάθε δρομολογητή που προωθεί το datagram. Εάν ο αριθμός του ορίου μεταπήδησης φτάσει στο μηδέν, το πακέτο δεδομένων απορρίπτεται.

- ***Source and destination addresses:*** Οι διάφορες μορφές της 128-bit Ipv6 διεύθυνσης.

- ***Data:*** Αυτό είναι το τμήμα ωφέλιμου φορτίου του πακέτου IPv6. Όταν το datagram φτάσει στον προορισμό του, τα χρήσιμα δεδομένα (payload) θα αφαιρεθούν από το πακέτο IP και θα μεταφερθούν στο πρωτόκολλο που καθορίζεται στο επόμενο πεδίο κεφαλίδας.

Η παραπάνω συζήτηση προσδιορίζει το σκοπό των πεδίων που περιλαμβάνονται στο πακέτο δεδομένων Ipv6. Παρατηρούμε ότι πολλά πεδία που εμφανίζονται στο πακέτο δεδομένων Ipv4 δεν υπάρχουν πλέον στο πακέτο δεδομένων Ipv6 και τα αναλύουμε στη συνέχεια:

- ***Κατακερματισμός/επανασυναρμολόγηση(Fragmentation/Reassembly):*** Το IPv6 δεν επιτρέπει τον κατακερματισμό και την επανασυναρμολόγηση στους ενδιάμεσους δρομολογητές. Αυτές οι λειτουργίες μπορούν να πραγματοποιηθούν μόνο από την πηγή και τον προορισμό. Εάν ένα πακέτο δεδομένων IPv6 που λαμβάνεται από έναν δρομολογητή είναι πολύ μεγάλο για να προωθηθεί μέσω του εξερχόμενου συνδέσμου, ο δρομολογητής απλά αποσύρει το datagram και στέλνει ένα μήνυμα σφάλματος ICMP "Packet Too Big" πίσω στον αποστολέα. Ο αποστολέας μπορεί στη συνέχεια να ξαναστείλει τα δεδομένα, χρησιμοποιώντας ένα μικρότερο μέγεθος δεδομένων IP. Ο κατακερματισμός και η επανασυναρμολόγηση είναι μια χρονοβόρα λειτουργία, για αυτό τον λόγο η κατάργηση αυτής της λειτουργικότητας από τους δρομολογητές και η τοποθέτησή της στα τελικά συστήματα επιταχύνει σημαντικά την προώθηση IP μέσα στο δίκτυο.

- ***Header checksum*:** Επειδή τα πρωτόκολλα στρώματος μεταφοράς (για παράδειγμα, TCP και UDP) και τα στρώματα συνδέσεων (για παράδειγμα, Ethernet) στα στρώματα του Διαδικτύου πραγματοποιούν ελέγχους, οι σχεδιαστές της IP πιθανώς θεώρησαν ότι αυτή η λειτουργικότητα ήταν επαρκώς περιττή στο επίπεδο δικτύου και θα μπορούσε να αφαιρεθεί. Για άλλη μια φορά, η ταχεία επεξεργασία των πακέτων IP αποτελούσε βασική μέριμνα της ομάδας εργασιών.

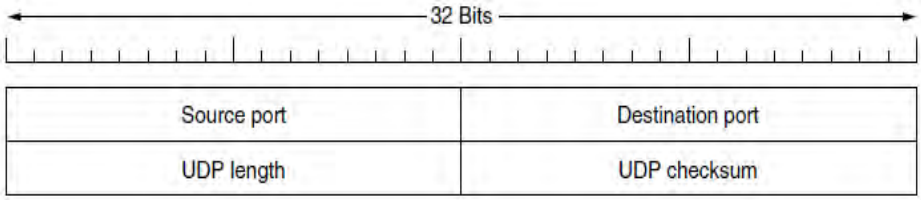
- ***Options:*** Ένα πεδίο επιλογών δεν αποτελεί πλέον μέρος της τυπικής κεφαλίδας IP. Ωστόσο, δεν έχει φύγει παντελώς απο αυτή. Αντίστοιχα, το πεδίο επιλογών είναι μία από τις πιθανές επόμενες κεφαλίδες που υποδεικνύονται από την κεφαλίδα IPv6. Δηλαδή, ακριβώς όπως οι κεφαλίδες πρωτοκόλλων TCP ή UDP μπορούν να είναι η επόμενη κεφαλίδα μέσα σε ένα πακέτο IP, έτσι μπορεί και ένα πεδίο επιλογών. Η κατάργηση του πεδίου επιλογών έχει ως αποτέλεσμα μια επικεφαλίδα IP σταθερού μήκους 40 bytes. [12]

	Internet Protocol version 4 (IPv4)	Internet Protocol version 6 (IPv6)
Deployed	1981	1999
Address Size	32-bit number	128-bit number
Address Format	Dotted Decimal Notation: 192.149.252.76	Hexadecimal Notation: 3FFE:F200:0234:AB00: 0123:4567:8901:ABCD
Prefix Notation	192.149.0.0/24	3FFE:F200:0234::/48
Number of Addresses	2^{32} = ~4,294,967,296	2^{128} = ~340,282,366, 920,938,463,463,374, 607,431,768,211,456

2.2 Το πρωτόκολλο μεταφοράς UDP

Το διαδίκτυο έχει δύο κύρια πρωτόκολλα στο στρώμα μεταφοράς, ένα πρωτόκολλο δικτύου χωρίς σύνδεση (connectionless protocol) και ένα με προσανατολισμένη όπως λέγεται σύνδεση (connection-oriented) τα οποία αλληλοσυμπληρώνουν το ένα το άλλο. Το πρωτόκολλο χωρίς σύνδεση που θα εξετάσουμε στη συνέχεια είναι το UDP (User Datagram Protocol) και δεν κάνει σχεδόν τίποτα πέρα από την αποστολή πακέτων μεταξύ των εφαρμογών, αφήνοντας τις εφαρμογές να δημιουργήσουν τα δικά τους πρωτόκολλα στην κορυφή, όπως απαιτείται. Το πρωτόκολλο προσανατολισμού σύνδεσης TCP (Transmission Control Protocol) που είναι και το πιο ουσιαστικό, κάνει σχεδόν τα πάντα, δηλαδή εκτελεί συνδέσεις και προσθέτει αξιοπιστία στις αναμεταδόσεις, ταυτόχρονα με τον έλεγχο ροής και τον έλεγχο συμφόρησης, και όλα αυτά για λογαριασμό των εφαρμογών που το χρησιμοποιούν και θα αναλυθεί στο επόμενο υποκεφάλαιο.

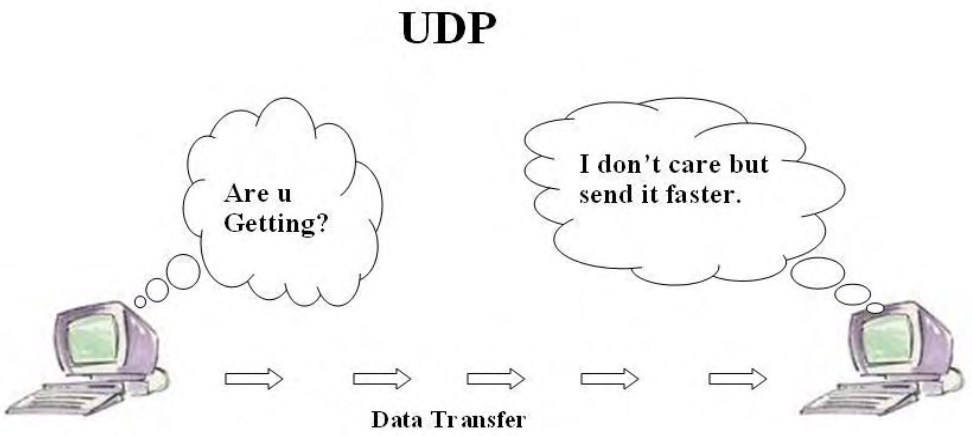
Το UDP παρέχει έναν τρόπο για τις εφαρμογές να στέλνουν ενσωματωμένα IP datagrams χωρίς να χρειάζεται να δημιουργηθεί μια σύνδεση. Το συγκεκριμένο πρωτόκολλο μεταδίδει τμήματα που αποτελούνται από μια επικεφαλίδα 8 bytes ακολουθούμενη από το ωφέλιμο φορτίο bytes όπως φαίνεται στη συνέχεια.



Οι δύο θύρες χρησιμεύουν για τον εντοπισμό των τελικών σημείων μέσα στις μηχανές προέλευσης και προορισμού. Όταν φτάσει ένα πακέτο UDP, το ωφέλιμο φορτίο (payload) του παραδίδεται στη διαδικασία που συνδέεται με τη θύρα προορισμού. Ας σκεφτούμε τις θύρες ως γραμματοκιβώτια που οι εφαρμογές μπορούν να νοικιάσουν για να λαμβάνουν πακέτα. Στην πραγματικότητα, η κύρια τιμή του UDP με τη χρήση της πρώτης IP είναι η προσθήκη των θυρών προέλευσης και προορισμού. Χωρίς τα πεδία θυρών, το στρώμα μεταφοράς δεν θα ξέρει πως να διαχειριστεί το κάθε εισερχόμενο πακέτο ενώ με αυτά παραδίδει το ενσωματωμένο τμήμα στη σωστή εφαρμογή.

Αξίζει να αναφέρουμε ρητά μερικά από τα πράγματα που το UDP δεν εκτελεί. Δεν εκτελεί έλεγχο ροής, έλεγχο συμφόρησης ή αναμετάδοση σε περίπτωση παραλαβής ενός κακού τμήματος. Όλα αυτά εξαρτώνται από τις διαδικασίες του χρήστη. Αυτό που κάνει είναι να παρέχει μια διασύνδεση στο πρωτόκολλο IP με την προστιθέμενη λειτουργία της αποπολυπλεξίας πολλαπλών διαδικασιών χρησιμοποιώντας τις θύρες και την προαιρετική ανίχνευση σφάλματος από άκρο σε άκρο. Για εφαρμογές που απαιτούν ακριβή έλεγχο της ροής πακέτων, έλεγχο σφαλμάτων ή χρονισμού, το UDP παρέχει ακριβώς αυτό που χρειάζονται. Μια περίπτωση όπου είναι ιδιαίτερα χρήσιμη είναι σε καταστάσεις πελάτη-

διακομιστή(client-server). Συχνά, ο πελάτης στέλνει ένα σύντομο αίτημα στο διακομιστή και αναμένει σύντομη απάντηση. Εάν χαθεί είτε η αίτηση είτε η απάντηση, ο πελάτης μπορεί απλώς να σταματήσει και να ξαναπροσπαθήσει. Όχι μόνο ο κώδικας που χρησιμοποιείται είναι απλός, αλλά απαιτούνται λιγότερα μηνύματα (ένα σε κάθε κατεύθυνση) από ότι σε ένα πρωτόκολλο όπως το TCP. **[13]**

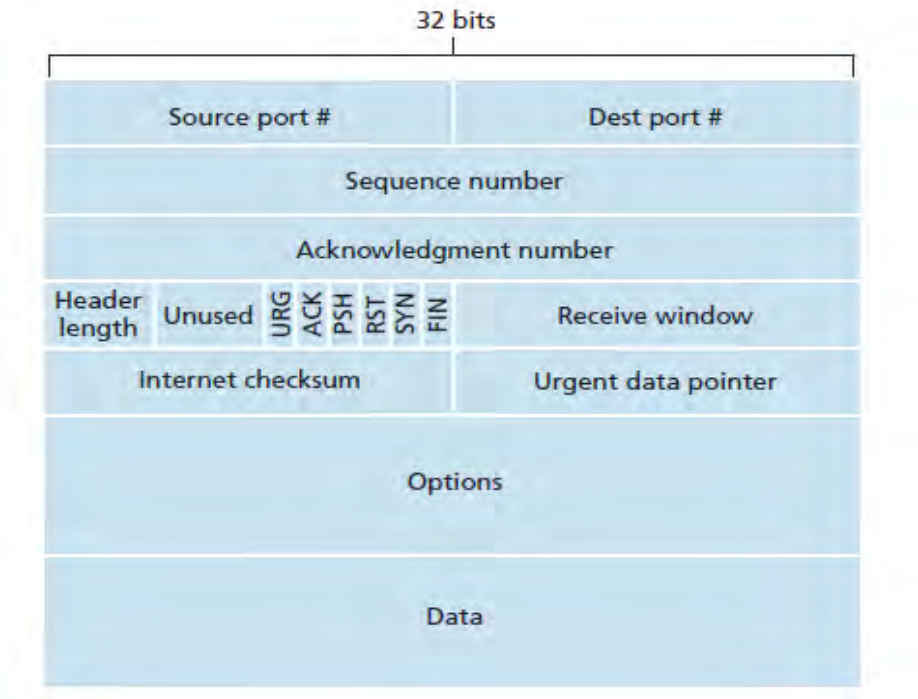


2.3 Το πρωτόκολλο μεταφοράς TCP

Μετά απο αναφορά στο πρωτόκολλο UDP ήρθε η ώρα να αναλύσουμε το αξιοσήμαντο και αξιόπιστο πρωτόκολλο μεταφοράς TCP. Σε αυτήν την παράγραφο θα παρατηρήσουμε ότι για να παρέχει αξιόπιστη μεταφορά δεδομένων, το TCP βασίζεται σε πολλές βασικές αρχές των δικτύων υπολογιστών συμπεριλαμβανομένων αυτών της ανίχνευσης σφαλμάτων, αναμετάδοσης, αθροιστικών αναγνωρίσεων, χρονιστών και πεδίων κεφαλίδας για αριθμούς ακολουθίας και επιβεβαίωσης.

Το TCP λέγεται ότι είναι προσανατολισμένο προς την σύνδεση, διότι πριν μια διαδικασία εφαρμογής μπορεί να αρχίσει η αποστολή δεδομένων σε μια άλλη, οι δύο διαδικασίες πρέπει πρώτα να πράξουν "χειραψία" (handshake) δηλαδή, πρέπει να στείλουν κάποια προκαταρκτικά τμήματα μεταξύ τους για να καθορίσουν τις παραμέτρους πρίν την επακόλουθη μεταφορά δεδομένων. Στο πλαίσιο της δημιουργίας σύνδεσης TCP, και οι δύο πλευρές της σύνδεσης θα αρχικοποιήσουν πολλές μεταβλητές κατάστασης TCP που συσχετίζονται με τη σύνδεση TCP.

Η TCP 'σύνδεση' δεν αποτελεί ένα εικονικό κύκλωμα καθώς η κατάσταση σύνδεσης εξαρτάται εξ ολοκλήρου απο τα δύο τελικά συστήματα. Επειδή το πρωτόκολλο TCP εκτελείται μόνο στα τελικά συστήματα και όχι στα ενδιάμεσα στοιχεία του δικτύου (δρομολογητές και διακόπτες στρώματος συνδέσμων), τα ενδιάμεσα στοιχεία του δικτύου δεν διατηρούν την κατάσταση σύνδεσης TCP. Αφού εξετάσαμε εν συντομία τη σύνδεση TCP ας εξετάσουμε και τη δομή του τμήματος TCP. Το τμήμα TCP αποτελείται από πεδία κεφαλίδων και πεδίο δεδομένων όπου το πεδίο δεδομένων περιέχει ένα κομμάτι δεδομένων εφαρμογής. Το μέγιστο μέγεθος τμήματος (maximum segment size- MSS) περιορίζει το μέγιστο μέγεθος του πεδίου δεδομένων ενός τμήματος όπως ορίζει και το όνομα του. Όταν το TCP στέλνει ένα μεγάλο αρχείο, όπως μια εικόνα ως μέρος μιας ιστοσελίδας, συνήθως σπάει το αρχείο σε κομμάτια μεγέθους MSS (εκτός από το τελευταίο κομμάτι, το οποίο συχνά είναι μικρότερο από το MSS). Οι διαδραστικές εφαρμογές, ωστόσο, συχνά μεταδίδουν κομμάτια δεδομένων που είναι μικρότερα από το MSS. Για παράδειγμα, σε εφαρμογές απομακρυσμένης σύνδεσης όπως το Telnet, το πεδίο δεδομένων στο τμήμα TCP είναι συχνά μόνο 1 byte. Επειδή η κεφαλίδα TCP είναι συνήθως 20 bytes (12 bytes περισσότερο από την κεφαλίδα UDP), τα τμήματα που αποστέλλονται από το Telnet μπορεί να έχουν μήκος μόνο 21 byte.



Στήν παραπάνω εικόνα απεικονίζεται η δομή του τμήματος TCP , όπως συμβαίνει και με το UDP, η κεφαλίδα περιλαμβάνει τους αριθμούς θύρας προέλευσης και προορισμού, οι οποίοι χρησιμοποιούνται για την πολυπλεξία / αποπολυπλεξία δεδομένων από / προς εφαρμογές ανώτερου επιπέδου. Επιπρόσθετα, η κεφαλίδα

περιλαμβάνει ένα πεδίο ελέγχου. Μια κεφαλίδα τμήματος TCP περιέχει επίσης τα ακόλουθα πεδία:

- Τα πεδία ακολουθίας αριθμού καθώς και αριθμού επιβεβαίωσης (**sequence number, acknowledgement number**) 32-bit το καθένα χρησιμοποιούνται από τον αποστολέα και τον δέκτη TCP για την υλοποίηση μιας αξιόπιστης υπηρεσίας μεταφοράς δεδομένων.

- Το πεδίο παραθύρου λήψης (**receive window**) μεγέθους 16-bit χρησιμοποιείται για τον έλεγχο ροής.

- Το πεδίο μήκους κεφαλίδας (**header length**) 4-bit καθορίζει το μήκος της κεφαλίδας TCP σε λέξεις 32-bit. Η κεφαλίδα TCP μπορεί να έχει μεταβλητό μήκος λόγω του πεδίου επιλογών TCP. (Συνήθως, το πεδίο επιλογών είναι κενό, έτσι ώστε το μήκος της τυπικής κεφαλίδας TCP να είναι 20 byte.)

- Το προαιρετικό και μεταβλητού μήκους πεδίο επιλογών (**options**) χρησιμοποιείται όταν ένας αποστολέας και δέκτης διαπραγματεύονται το μέγιστο μέγεθος τμήματος ή ως συντελεστής κλιμάκωσης παραθύρων για χρήση σε υψηλής ταχύτητας δίκτυα.

- Το πεδίο σημαίας περιέχει 6 bits. Το bit **ACK** χρησιμοποιείται για να υποδείξει ότι η τιμή που μεταφέρεται στο πεδίο επιβεβαίωσης είναι έγκυρη. Δηλαδή, το τμήμα περιέχει μια επιβεβαίωση για ένα τμήμα που έχει ληφθεί επιτυχώς. Τα bits **RST**, **SYN** και **FIN** χρησιμοποιούνται για τη ρύθμιση σύνδεσης και τον τερματισμό. Η ρύθμιση του bit **PSH** υποδηλώνει ότι ο δέκτης θα πρέπει να μεταβιβάσει αμέσως τα δεδομένα στο ανώτερο στρώμα. Τέλος, το bit **URG** χρησιμοποιείται για να υποδείξει ότι υπάρχουν δεδομένα που η οντότητα ανώτερου στρώματος της αποστολής έχει επισημάνει ως "επείγουσα". Η θέση του τελευταίου byte αυτών των επειγόντων δεδομένων υποδεικνύεται από το επείγον 16 bit πεδίο δείκτη δεδομένων. Το TCP πρέπει να ενημερώνει την οντότητα ανώτερου στρώματος της πλευράς λήψης όταν υπάρχουν επείγοντα δεδομένα και να περάσει έναν δείκτη στο τέλος των επειγόντων δεδομένων. Στην πράξη τα PSH, URG και ο επείγων δείκτης δεδομένων δεν χρησιμοποιούνται αλλά τα αναφέρουμε για λόγους έρευνας.[13]

2.4 Το πρωτόκολλο ελέγχου ICMP

Θυμόμαστε ότι το στρώμα δικτύου του Internet αποτελείται απο τρία βασικά στοιχεία: το πρωτόκολλο IP, τα πρωτόκολλα δρομολόγησης Internet και το ICMP, το οποίο αποτελεί το αντικείμενο αυτής της ενότητας.

Παρόλο που η IP είναι ένα πρωτόκολλο απολύτως 'πρόθυμο' να απορρίπτει τα datagrams όταν η μετάβαση γίνεται πιο σκληρή (για παράδειγμα, όταν ένας δρομολογητής δεν ξέρει πώς να προωθήσει το datagram ή όταν ένα κομμάτι ενός datagram αποτυγχάνει να φθάσει στον προορισμό του) η απόρριψη αυτή δεν συμβαίνει σιωπηλά χωρίς καμία ένδειξη. Το IP είναι πάντα διαμορφωμένο με ένα συνοδευτικό πρωτόκολλο, γνωστό και ως Internet Control Message Protocol (ICMP), το οποίο ορίζει μια συλλογή μηνυμάτων σφάλματος που αποστέλλονται ξανά στον κεντρικό υπολογιστή προέλευσης κάθε φορά που ένας δρομολογητής ή ένας κεντρικός υπολογιστής δεν είναι σε θέση να επεξεργαστεί επιτυχώς ένα πακέτο δεδομένων IP.

Το ICMP ορίζει μηνύματα σφάλματος που υποδεικνύουν ότι ο κεντρικός υπολογιστής προορισμού δεν είναι προσβάσιμος (ίσως εξαιτίας μιας αποτυχίας σύνδεσης), ότι η διαδικασία επανασύνδεσης απέτυχε, ότι ο TTL είχε φτάσει στο 0, ότι η κεφαλίδα ελέγχου IP απέτυχε και ούτω καθεξής. Το ICMP θεωρείται συχνά μέρος της IP, αλλά αρχιτεκτονικά βρίσκεται ακριβώς πάνω από το IP, καθώς τα μηνύματα ICMP μεταφέρονται μέσα σε IP datagrams. Δηλαδή, τα μηνύματα ICMP μεταφέρονται ως ωφέλιμο φορτίο IP, ακριβώς όπως τα τμήματα TCP ή UDP . Ομοίως, όταν ένας κεντρικός υπολογιστής λαμβάνει ένα πακέτο IP με το ICMP που ορίζεται ως το πρωτόκολλο ανώτερου στρώματος, αποπολυπλέκει τα περιεχόμενα του datagram στο ICMP, ακριβώς όπως θα αποπολυπλέξει το περιεχόμενο ενός datagram σε TCP ή UDP.

Επιπλέον, το ICMP ορίζει ένα σύνολο μηνυμάτων ελέγχου που ένας δρομολογητής μπορεί να στείλει πίσω σε έναν κεντρικό υπολογιστή προέλευσης. Ένα από τα πιο χρήσιμα μηνύματα ελέγχου, που ονομάζεται ανακατεύθυνση-ICMP (ICMP-Redirect), αναφέρει στον κεντρικό υπολογιστή προέλευσης ότι διατίθεται μια καλύτερη διαδρομή προς τον προορισμό. Οι ανακατευθύνσεις ICMP χρησιμοποιούνται στην ακόλουθη περίπτωση που δίνουμε ως παράδειγμα κατανόησης. Ας υποθέσουμε ότι ένας κεντρικός υπολογιστής είναι συνδεδεμένος σε ένα δίκτυο που έχει δύο δρομολογητές συνδεδεμένους σε αυτόν, τους R1 και R2, όπου ο κεντρικός υπολογιστής χρησιμοποιεί τον R1 ως προεπιλεγμένο δρομολογητή του. Σε περίπτωση που το R1 λάβει κάποια στιγμή ένα datagram από τον κεντρικό υπολογιστή, όπου με βάση τον πίνακα προώθησής του γνωρίζει ότι το R2 θα ήταν καλύτερη επιλογή για μια συγκεκριμένη διεύθυνση προορισμού, στέλνει ένα μήνυμα ICMP-Redirect πίσω στον κεντρικό υπολογιστή δίνοντάς του εντολή να χρησιμοποιεί το R2 για όλα τα μελλοντικά πακέτα δεδομένων που απευθύνονται στον συγκεκριμένο προορισμό. Στη συνέχεια ο κεντρικός υπολογιστής προσθέτει αυτή τη νέα διαδρομή στον πίνακα προώθησης.

Τα μηνύματα ICMP έχουν ένα πεδίο τύπου και πεδίο κωδικού αντίστοιχα και περιέχουν την κεφαλίδα και τα πρώτα 8 byte του πακέτου IP που προκάλεσαν το μήνυμα ICMP που δημιουργήθηκε στην πρώτη θέση (έτσι ώστε ο αποστολέας να μπορεί να καθορίσει το datagram που προκάλεσε το σφάλμα).[14] Οι επιλεγμένοι τύποι μηνυμάτων ICMP παρουσιάζονται στην εικόνα παρακάτω. Να σημειωθεί ότι τα μηνύματα ICMP δεν χρησιμοποιούνται μόνο για συνθήκες σηματοδότησης σφάλματος.

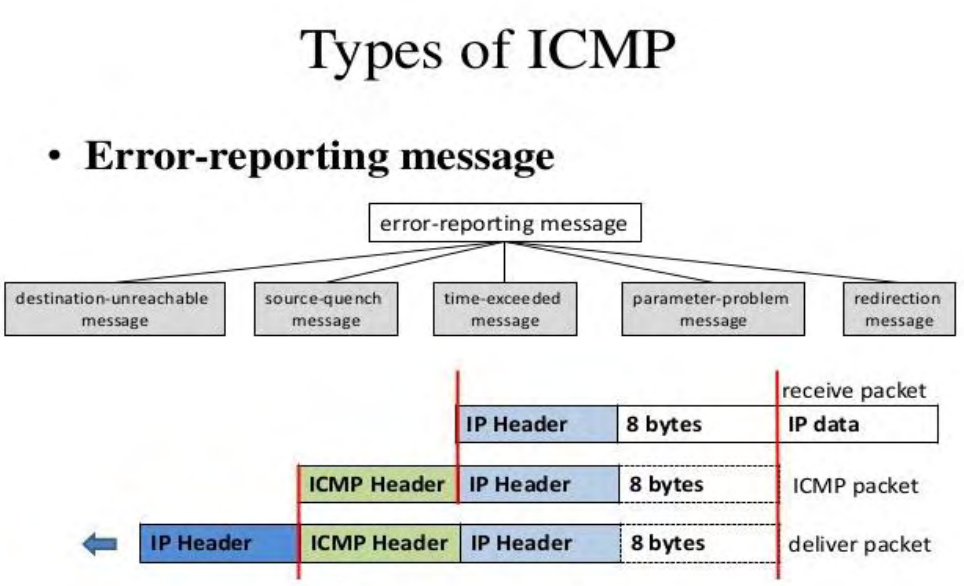
ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

Το ICMP παρέχει επίσης τη βάση για δύο ευρέως χρησιμοποιούμενα εργαλεία εντοπισμού σφαλμάτων, το ping και το traceroute για το οποίο θα αναλύσουμε λίγα πράγματα στη συνέχεια. Το ping χρησιμοποιεί μηνύματα echo ICMP για να καθορίσει εάν ένας κόμβος είναι προσβάσιμος και ζωντανός. Το traceroute χρησιμοποιεί μια ελαφρώς μη διαισθητική τεχνική για να καθορίσει το σύνολο των δρομολογητών κατά μήκος της διαδρομής προς έναν προορισμό. Για να προσδιορίσει τα ονόματα και τις διευθύνσεις των δρομολογητών μεταξύ της πηγής και του προορισμού, η πηγή Traceroute στέλνει μια σειρά από συνηθισμένα datagrams IP στον προορισμό. Κάθε ένα από αυτά τα datagrams φέρει ένα τμήμα UDP με έναν απίθανο αριθμό θύρας UDP. Το πρώτο από αυτά τα datagrams έχει TTL 1, το δεύτερο 2, το τρίτο 3, και ούτω καθεξής. Η πηγή ξεκινά επίσης χρονοδιακόπτες για κάθε ένα από τα datagrams. Όταν το ν-οστό datagram φτάνει στον ν-οστό δρομολογητή, ο ν-οστός δρομολογητής παρατηρεί ότι το TTL του datagram μόλις έληξε. Σύμφωνα με τους κανόνες του πρωτοκόλλου IP, ο δρομολογητής απορρίπτει το datagram και στέλνει ένα

μήνυμα προειδοποίησης ICMP στην πηγή (κωδικός 0, τύπος 11). Αυτό το προειδοποιητικό μήνυμα περιλαμβάνει το όνομα του δρομολογητή και τη διεύθυνση IP του. Όταν αυτό το μήνυμα ICMP επανέλθει στην πηγή, η πηγή αποκτά τον χρόνο γύρου από το χρονοδιακόπτη και το όνομα και τη διεύθυνση IP του ν-οστού δρομολογητή από το μήνυμα ICMP.

Σε αυτό το σημείο θα μπορούσε κάποιος να ερωτηθεί το πως μια πηγή Traceroute γνωρίζει το πότε θα πρέπει να σταματήσει να στέλνει πακέτα ανίχνευσης UDP. Να φέρουμε στο μυαλό μας ότι η πηγή αυξάνει το πεδίο TTL για κάθε datagram που στέλνει. Έτσι, ένα από τα datagrams θα καταλήξει τελικά σε όλη τη διαδρομή προς τον κεντρικό υπολογιστή προορισμού. Επειδή αυτό το datagram περιέχει ένα τμήμα UDP με έναν απίθανο αριθμό θύρας, ο κεντρικός υπολογιστής προορισμού αποστέλλει πίσω στην πηγή ένα μήνυμα μη αποδεκτής θύρας ICMP (κωδικός 3, τύπος 3). Όταν ο κεντρικός υπολογιστής-πηγή λάβει αυτό το συγκεκριμένο μήνυμα ICMP, γνωρίζει ότι δεν χρειάζεται να στείλει πρόσθετα πακέτα ανίχνευσης.

Με αυτόν τον τρόπο, ο υπολογιστής-πηγή μαθαίνει τον αριθμό και την ταυτότητα των δρομολογητών που βρίσκονται μεταξύ αυτού και του υπολογιστή-προορισμού καθώς και τον χρόνο γύρου ταξιδιού μεταξύ τους. Σημειώστε ότι το πρόγραμμα-πελάτης Traceroute πρέπει να είναι σε θέση να καθοδηγήσει το λειτουργικό σύστημα να παράγει datagrams UDP με συγκεκριμένες τιμές TTL και πρέπει επίσης να μπορεί να ειδοποιηθεί από το λειτουργικό του σύστημα όταν φθάσουν τα μηνύματα ICMP. [12]



ΚΕΦΑΛΑΙΟ 3

Αφού αναλύσαμε ορισμένα απο τα βασικότερα πρωτόκολλα τα οποία αποτελούν στηλοβάτες για το παγκόσμιο διαδίκτυο είναι η κατάλληλη στιγμή να αναφερθούμε στους τρόπους που χρησιμοποιούν τα sniffer ηθικά και μή άτομα που εμπλέκονται με την σύλληψη, την καταγραφή και την ανάλυση των διαδικτυακών πακέτων. Το Sniffing διακρίνεται σε δύο κατηγορίες, το ενεργό και το παθητικό.

Στο παθητικό sniffing, η κίνηση καταγράφεται αλλά δεν μεταβάλλεται με κανέναν τρόπο. Το παθητικό sniffing επιτρέπει μόνο την ακρόαση και λειτουργεί με συσκευές διανομής (Hubs). Σε μια συσκευή διανομέα, η κυκλοφορία στέλνεται σε όλες τις θύρες. Σε ένα δίκτυο που χρησιμοποιεί κόμβους για τη σύνδεση συστημάτων, όλοι οι κεντρικοί υπολογιστές στο δίκτυο μπορούν να δουν την επισκεψιμότητα. Επομένως, ένας εισβολέας μπορεί εύκολα να καταγράψει την κίνηση που διέρχεται. Τα καλά νέα είναι ότι οι κόμβοι είναι σχεδόν παρωχημένοι στις μέρες μας και τα περισσότερα σύγχρονα δίκτυα χρησιμοποιούν διακόπτες. Ως εκ τούτου, το παθητικό sniffing δεν είναι αποτελεσματικό και δεν χρησιμοποιείται ιδιαίτερα.

Απο την άλλη πλευρά στο ενεργό sniffing η κυκλοφορία όχι μόνο καταγράφεται και παρακολουθείται, αλλά μπορεί επίσης να μεταβληθεί με κάποιο τρόπο όπως καθορίζεται από την επίθεση. Χρησιμοποιείται για να καταγράψουμε την κυκλοφορία σε ένα δίκτυο βασισμένο σε διακόπτες και περιλαμβάνει την εισαγωγή πακέτων ανάλυσης διευθύνσεων (ARP) σε ένα δίκτυο στόχο ώστε να πλημμυρίσει ο πίνακας μνήμης που απευθύνεται σε περιεχόμενο (Content-addressable memory-CAM). Το CAM παρακολουθεί ποιος κεντρικός υπολογιστής είναι συνδεδεμένος σε ποια θύρα. Ορισμένες τεχνικές sniffing και επιθέσεις είναι οι MAC Flooding, MiddleMan SSL Attack, DHCP Attacks, DNS Cache Poisoning, Spoofing Attacks,ARP Poisoning, SNMP Monitoring, RTT Detection, Ping Method ή ICMP Flood εκ των οποίων ορισμένες θα τις αναλυθούν στα επόμενα υποκεφάλαια.

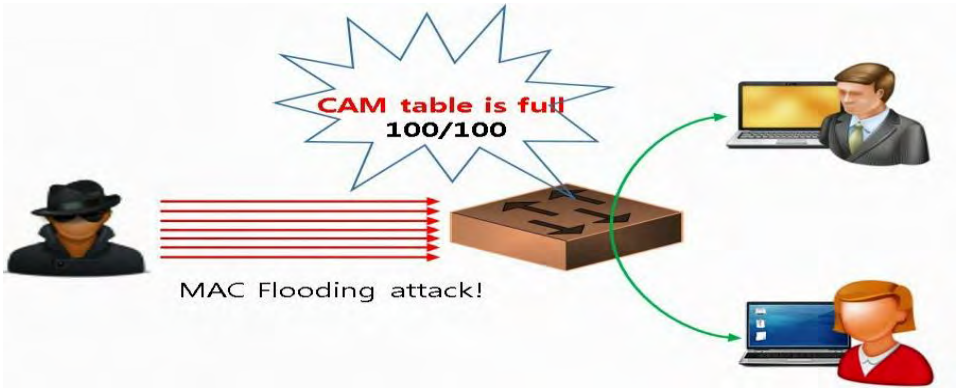
3.1 MAC Flooding

Οι περισσότερες εταιρίες αντικαθιστούν τους κόμβους (hubs) με διακόπτες (switches) για να αποτρέψουν τις απεριόριστες επιθέσεις με sniffers. Μια sniffing επίθεση επιτρέπει σε έναν εισβολέα να συνδέσει μια κάρτα διασύνδεσης δικτύου σε μια αχρησιμοποίητη πρίζα τοίχου και να καταγράψει δεδομένα. Εάν χρησιμοποιείται ένας κόμβος, ο εισβολέας μπορεί να συλλάβει δεδομένα που

μετακινούνται μέσω του κόμβου ενώ αντίθετα εάν χρησιμοποιείται ένας διακόπτης, ο εισβολέας δεν είναι σε θέση να καταγράψει μεγάλο όγκο δεδομένων. Ένας διακόπτης Ethernet καθορίζει πού να αποσταλεί η κυκλοφορία σε ένα LAN με βάση τη διεύθυνση ελέγχου πρόσβασης πολυμέσων (MAC). Η διεύθυνση MAC δηλώνει το μοναδικό υλικό διασύνδεσης δικτύου για κάθε σύστημα συνδεδεμένο στο τοπικό δίκτυο.

Ένας διακόπτης Ethernet παρακολουθεί την κίνηση σε ένα LAN για να μάθει ποια βύσματα στον διακόπτη συνδέονται με ποιές διευθύνσεις MAC. Για παράδειγμα, ο διακόπτης θα δει την κυκλοφορία που έρχεται από τη διεύθυνση MAC: BB: CC: DD: EE: FF στο βύσμα νούμερο ένα. Ο διακόπτης θα θυμάται αυτές τις πληροφορίες και θα στέλνει δεδομένα προοριζόμενα για αυτή τη διεύθυνση MAC μόνο στο πρώτο βύσμα του διακόπτη. Ομοίως, ο διακόπτης θα εντοπίσει αυτομάτως τις διευθύνσεις MAC που σχετίζονται με τις άλλες διεπαφές δικτύου στο τοπικό δίκτυο και θα στείλει τα κατάλληλα δεδομένα σε αυτές.

Μία από τις απλούστερες, ενεργές τεχνικές sniffing περιλαμβάνει το 'πλημμύρισμα' του LAN με κίνηση που αποτελείται απο ψεύτικες διευθύνσεις MAC. Ο εισβολέας χρησιμοποιεί ένα πρόγραμμα που είναι εγκατεστημένο σε ένα μηχάνημα στο LAN για να δημιουργήσει πακέτα με τυχαίες διευθύνσεις MAC και να τα τροφοδοτήσει στον διακόπτη. Ο διακόπτης θα προσπαθήσει να θυμάται όλες τις διευθύνσεις MAC κατά την άφιξή τους. Τελικά, η χωρητικότητα μνήμης του διακόπτη θα εξαντληθεί απο τις υπεράριθμες ψευδείς διευθύνσεις MAC. Ορισμένοι διακόπτες όταν η μνήμη τους γεμίσει λέμε ότι αποτυγχάνουν και τίθενται σε μια κατάσταση όπου η κυκλοφορία στέλνεται σε όλα τα μηχανήματα που είναι συνδεδεμένα στο τοπικό δίκτυο και ουσιαστικά λειτουργούνε ως hubs. Με τη χρήση της MAC Flooding τεχνικής, ένας εισβολέας μπορεί να βομβαρδίσει αδιάκοπα έναν διακόπτη έτσι ώστε αυτός να στείλει όλη την κίνηση σε όλα τα μηχανήματα του τοπικού δικτύου. Ο επιτιθέμενος μπορεί στη συνέχεια να χρησιμοποιήσει ένα παραδοσιακό sniffer για να συλλέξει τα δεδομένα από το τοπικό δίκτυο. [15]



3.2 SNMP Monitoring

Ένα ολοκληρωμένο σύστημα διαχείρισης δικτύου SNMP είναι ένα σύστημα που αξιοποιεί το SNMP για να παρέχει μια πλήρη εικόνα ενός δικτύου. Οι

ολοκληρωμένες πλατφόρμες παρακολούθησης συλλέγουν ενεργά πληροφορίες δικτύου από συσκευές δικτύου και αναλύουν τα δεδομένα. Οι πιο ολοκληρωμένες πλατφόρμες παρέχουν γραφικές αναφορές, παρακολούθηση και διαχείριση δικτύων που επιτρέπουν εύκολη διαχείριση και ρύθμιση της παρακολούθησης. Τα ολοκληρωμένα συστήματα παρακολούθησης παρέχουν λεπτομερή και ολοκληρωμένη ανάλυση της απόδοσης του δικτύου. Τα κοινά χαρακτηριστικά απόδοσης περιλαμβάνουν τη χρήση του εύρους ζώνης, τη διαθεσιμότητα, το χρόνο απόκρισης και τους ρυθμούς σφαλμάτων, καθώς και τη χρήση του CPU και της μνήμης του εξοπλισμού δικτύου και των διακομιστών. Εκτός από την παρακολούθηση και την αναφορά σε πραγματικό χρόνο, τα συστήματα διαχείρισης δικτύου SNMP παρέχουν επίσης και τους πελάτες που βασίζονται σε οποιοδήποτε από τα χαρακτηριστικά παρακολούθησης που έχουν παρακολουθείται. Υπάρχουν πολλά και διάφορα εργαλεία τα οποία χρησιμοποιούνται για την αναγνώριση και για την παρακολούθηση ενός δικτύου (πολλά από τα οποία χρησιμοποιούν τυπικές εγκαταστάσεις δικτύων όπως ICMP (traceroute), DNS και SNMP). Ορισμένα από αυτά τα εργαλεία είναι περισσότερο ανιχνεύσιμα από άλλα (και επομένως, ίσως πιο χρήσιμα μέσα σε μια διαδικτυακή περίμετρο) και επιταχύνουν τη διαδικασία συλλογής δεδομένων τοπολογίας δικτύου.

Μια από τις πιο επιτυχημένες επιθέσεις στο πλαίσιο της κατηγορίας άρνησης εξυπηρέτησης (DenialofService-DoS) περιλαμβάνει την αφαίρεση ή τροποποίηση των κρίσιμων πληροφοριών διαμόρφωσης μιας συσκευής, για παράδειγμα σε δρομολογητή, διακόπτη ή συσκευή απομακρυσμένης πρόσβασης. Ένα παράδειγμα αυτού του τύπου επίθεσης θα ήταν αν ένας εισβολέας καταφέρει να χρησιμοποιήσει το πρωτόκολλο Simple Network Management Protocol (SNMP), έχοντας γνώση της SNMP Private Community String (η οποία μπορεί συχνά να είναι brute-forced ή να έχει μαντευτεί μέσω μιας επίθεσης με λεξικό) για να διαβάσει και να προκαλέσει αλλαγές στη διαμόρφωση της συσκευής. Ένας εισβολέας θα μπορούσε απλώς να τροποποιήσει τις πληροφορίες ρύθμισης παραμέτρων της συσκευής που ανακατευθύνουν πακέτα ή σειριακά δεδομένα που διέρχονται από τη συσκευή σε άλλα συστήματα ή σε μια εικονική τοποθεσία, αποτρέποντας έτσι τη σωστή λειτουργία της συσκευής. Άλλη μια περίπτωση είναι οτι θα μπορούσε επίσης να αντικαταστήσει το αρχείο διαμόρφωσης με ένα κενό αρχείο, αποτρέποντας την κανονική λειτουργία της συσκευής και κατά συνέπεια αχρηστεύοντας οποιεσδήποτε υπηρεσίες μπορεί να εξαρτώνται από τη συσκευή για σωστή λειτουργία. Ένα παράδειγμα αυτού μπορεί να είναι ένας εισβολέας που εκδίδει μια εντολή "Write Erase" σε μια συσκευή στην οποία έχει πραγματοποιηθεί επίθεση, απομακρύνοντας αποτελεσματικά τις πληροφορίες διαμόρφωσης της.

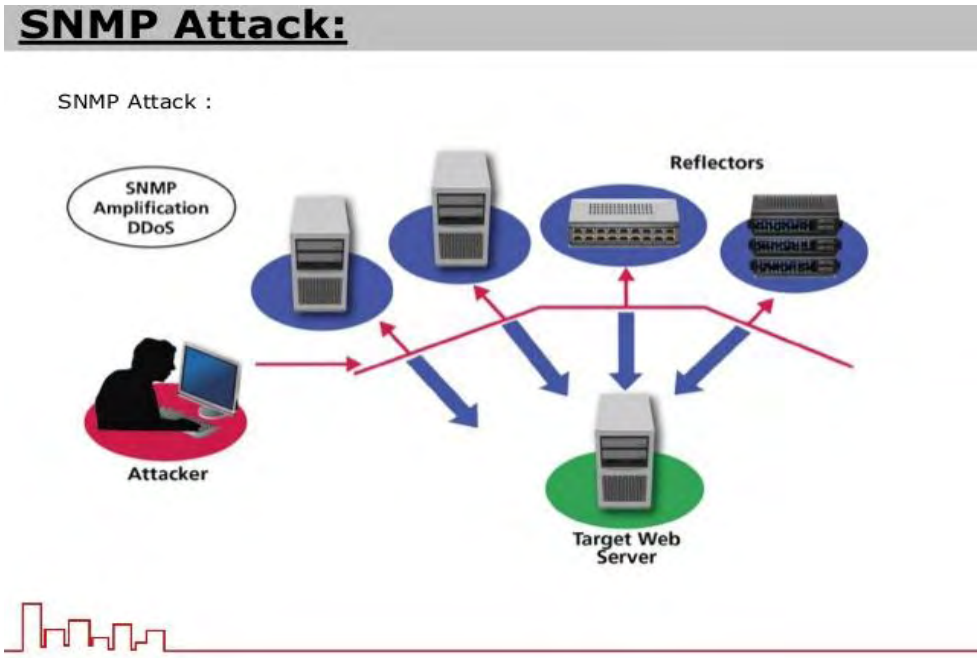
Ορισμένα αντικείμενα δικτύου θα απενεργοποιήσουν τις υπηρεσίες τους ή ακόμα και θα τερματίσουν τελείως όταν γίνει λήψη συγκεκριμένων εντολών διαχείρισης μέσω του δικτύου. Εάν ένας εισβολέας είναι σε θέση να στείλει ένα

πακέτο που έχει δημιουργήσει ζητώντας επανεκκίνηση ή τερματισμό της συσκευής τότε ο επιτιθέμενος αποτρέπει αποτελεσματικά τη λειτουργία της συσκευής και αρνείται σε όλους τους χρήστες να έχουν πρόσβαση στη πηγή της συσκευής για ορισμένο χρονικό διάστημα. Η παραπάνω περίπτωση επίθεσης μπορεί να επιτευχθεί είτε μέσω εξουσιοδοτημένης είτε μη εξουσιοδοτημένης πρόσβασης μέσω πρωτοκόλλου διαχείρισης όπως το Simple Network Management Protocol (SNMP) ή μέσω υπερχείλισης buffer.

Οι οργανισμοί που δεν εφαρμόζουν κάποια μέθοδο επαλήθευσης ταυτότητας μεταξύ ενός συστήματος διαχείρισης δικτύου και συγκεκριμένων συσκευών είναι επιρρεπείς σε αυτόν τον τύπο επίθεσης. Ένας επιτιθέμενος χρήστης μπορεί να είναι σε θέση να σχεδιάσει ένα παρωχημένο πακέτο πρωτοκόλλου διαχείρισης (δηλ. SNMP), προειδοποιώντας το σύστημα διαχείρισης για κάποιο ψευδές "κρίσιμο συμβάν" που υπάρχει σε μια συσκευή. Για παράδειγμα, ένας εισβολέας μπορεί να στείλει ένα παρωχημένο πακέτο SNMP που δηλώνει ότι τερματίζει μια συσκευή, η οποία όταν ληφθεί από το προσωπικό του δικτύου επιχειρήσεων (NOC) ενδέχεται να οδηγήσει σε αποστολή πόρων για τη διόρθωση του ψευδούς συμβάντος. Αυτός ο τύπος επίθεσης λειτουργεί καλύτερα σε συστήματα που φιλοξενούνται εξ αποστάσεως χωρίς απομακρυσμένη πρόσβαση, απαιτώντας επιπλέον χρόνο για την επιβεβαίωση του συμβάντος. Με προνομιακή πρόσβαση σε μια δεδομένη συσκευή μπορούμε να κάνουμε τροποποιήσεις στις λίστες πρόσβασης μιας συσκευής επιτιθέμενοι είτε με τις υπηρεσίες SNMP ή με την χρήση μη ασφαλούς κονσόλας ή με την απομακρυσμένη πρόσβαση. Αυτές οι τροποποιήσεις ενδέχεται να περιλαμβάνουν απλή τροποποίηση των κανόνων ή πλήρη απομάκρυνση της ίδιας της λίστας πρόσβασης. Όχι μόνο οι επιτιθέμενοι μπορούν να καταργήσουν τη λίστα πρόσβασης από μια συσκευή αλλά μπορούν επίσης να καταργήσουν την διεπαφή εντολών στην οποία εφαρμόζεται η λίστα πρόσβασης κάνωντας το να φαίνεται ότι η πρόσβαση στον κατάλογο πρόσβασης εξακολουθεί να εφαρμόζεται σωστά όταν στην πραγματικότητα δεν λειτουργεί σωστά.

Η συντριπτική πλειοψηφία των συσκευών υποδομής δικτύου υλοποιεί SNMP για σκοπούς διαχείρισης. Δυστυχώς, η SNMP έκδοση 1 (SNMPv1) δεν έχει άλλες επιλογές ασφάλειας εκτός από τη χρήση μη συνηθισμένων συμβολοσειρών της ευρούς κοινότητας. Το SNMPv1 στέλνει αυτές τις συμβολοσειρές με απόλυτη σαφήνεια έτσι ώστε να μην είναι απαραίτητο αυτές να βασίζονται στην εξουσιοδότηση. Αυτό αναγκάζει τους διαχειριστές δικτύου να εφαρμόζουν λίστες πρόσβασης για να περιορίζουν τις συσκευές που έχουν πρόσβαση στα δεδομένα SNMP στις συσκευές υποδομής δικτύου. Λόγω τεκμηριωμένων ανησυχιών σχετικά με την ασφάλεια με το SNMPv1και SNMPv2, συνιστάται να απενεργοποιήσετε το SNMP σε εξωτερικά προσβάσιμο εξοπλισμό δικτύου ή τουλάχιστον να τοποθετήσετε λίστες πρόσβασης που απαγορεύουν την κυκλοφορία SNMP σε εξωτερικές διεπαφές. Τα περισσότερα εργαλεία υποδομής δικτύου μπορούν να διαμορφωθούν ώστε να λειτουργούν ως ένας πελάτης SNMP.

Όταν η υπηρεσία SNMP είναι ενεργοποιημένη στο εργαλείο, τα εργαλεία διαχείρισης δικτύου μπορούν να την χρησιμοποιήσουν για τη συλλογή πληροφοριών σχετικά με τη διαμόρφωση, την διαδικτυακή κυκλοφορία και άλλα. Όπως αναφέρθηκε παραπάνω, οι εκδόσεις 1 και 2 του SNMP δεν θεωρούνται ασφαλείς εξαιτίας της έλλειψης παραχώρησης αυθεντικότητας. Εξαιτίας αυτού, το SNMP θα πρέπει να χρησιμοποιείται μόνο σε εσωτερικά ή προστατευμένα δίκτυα.[16]



3.3 DNS Cache Poisoning

Το DNS έχει εξελιχθεί σε έναν εύθραυστο σύνδεσμο στην ασφάλεια του διαδικτύου επειδή ουσιαστικά παρέχει έναν ενιαίο κατάλογο πληροφοριών για πλοήγηση στα σημερινά δίκτυα. Εάν καταργήσουμε τους δημόσιους διακομιστές DNS ενός οργανισμού ή χειριστούμε τα δεδομένα που φιλοξενούνται σε αυτούς τους διακομιστές ενδέχεται να μειώσουμε την ύπαρξη του στο Internet. Επίσης εάν καταστρέψουμε την προσωρινή μνήμη DNS σε μια πύλη Internet ουσιαστικά απαγορεύουμε στον οργανισμό να χρησιμοποιεί την πύλη αυτή για πρόσβαση σε περιοχές του Διαδικτύου ώστε να χρησιμοποιήσει μια πληθώρα υπηρεσιών. Σε περίπτωση που διεισδύσουμε στην ιδιωτική υποδομή DNS ενός οργανισμού

μπορούμε να καταστρέψουμε την ικανότητά του να είναι σε θέση να πλοηγηθεί σε συγκεκριμένους διακομιστές. Το DNS είναι ένας βολικός στόχος για μια σειρά επιθέσεων άρνησης εξυπηρέτησης (DoS) που μπορούν να καταργήσουν με χειρουργική ακρίβεια έναν οργανισμό από το τοπίο του Internet ή να θέσει σε κίνδυνο την ιντερνετική συνδεσιμότητα.

Από την πλευρά του χάκερ, το DNS αποτελεί έναν ελκυστικό στόχο επειδή είναι μια υπηρεσία που εγγυάται ότι θα χρησιμοποιηθεί από οργανισμούς προσιτούς στο Διαδίκτυο και πηγή μεγάλης χρήσιμης αναγνώρισης IP και τοπολογίας δικτύων. Τελικά το DNS είναι ένας πηγαίος κατάλογος για όλα τα είδη πληροφοριών που σχετίζονται με έναν κεντρικό υπολογιστή (διευθύνσεις IP, ονόματα, υπηρεσίες κ.λπ.) δηλαδή ένα πραγματικό "τηλεφωνικό βιβλίο" στο Διαδίκτυο και το δίκτυο.

Μια επίθεση 'DNS cache poisoning' προσπαθεί να δημιουργήσει μια ψεύτικη παραποιημένη απάντηση από ένα εξουσιοδοτημένο nameserver(NS), αναγκάζοντας έτσι έναν αναδρομικό NS να αποθηκεύσει πλαστές πληροφορίες στην εσωτερική του μνήμη. Για το λόγο αυτό η επίθεση ονομάζεται «δηλητηρίαση της μνήμης cache». Ως αποτέλεσμα της δηλητηρίασης της εσωτερικής μνήμης, του αναδρομικού NS όλα τα επακόλουθα ερωτήματα επιλύονται με τις πλαστές πληροφορίες που έχουν δημιουργηθεί.

Μια παραποιημένη απάντηση πρέπει να πληροί αυτές τις απαιτήσεις ώστε να γίνει δεκτή από ένα αναδρομικό NS:

1. Παρέχεται στο αναδρομικό NS πριν από την απάντηση της έγκυρης NS.

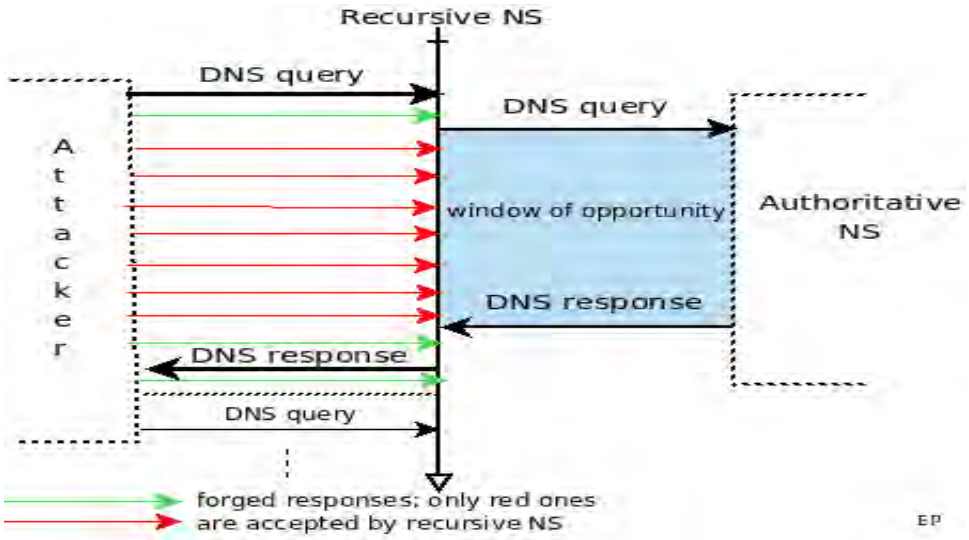
2. Περιέχει ένα "Τμήμα ερωτήσεως" από το αρχικό ερώτημα του αναδρομικού NS.

3. Το αναγνωριστικό (ID) της απόκρισης ταιριάζει με το αναγνωριστικό του ερωτήματος.

4. Η διεύθυνση προέλευσης της παραποιημένης απόκρισης αντιστοιχεί στη διεύθυνση προορισμού του αντίστοιχου ερωτήματος.

5. Η θύρα / διεύθυνση στόχος της παραποιημένης απάντησης ταιριάζει με τη θύρα προέλευσης / διεύθυνση του αναδρομικού ερωτήματος NS.

Η ακόλουθη εικόνα εμφανίζει το ερώτημα DNS στο κύριο NS:



Η επόμενη εικόνα δείχνει μια παραποιημένη απάντηση DNS του εισβολέα που πληροί όλες τις απαιτήσεις που αναφέραμε παραπάνω:

```
▷ Internet Protocol, Src: 193.0.0.236 (193.0.0.236), Dst: 192.168.1.4 (192.168.1.4)
▷ User Datagram Protocol, Src Port: domain (53), Dst Port: 6661 (6661)
▽ Domain Name System (response)
    Transaction ID: 0xb16a
  ▷ Flags: 0x8580 (Standard query response, No error)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 1
    Additional RRs: 1
  ▽ Queries
    ▽ 3-269f-25035.example.net: type A, class IN
        Name: 3-269f-25035.example.net
        Type: A (Host address)
        Class: IN (0x0001)
  ▽ Answers
    ▷ 3-269f-25035.example.net: type A, class IN, addr 1.2.3.4
  ▽ Authoritative nameservers
    ▷ example.net: type NS, class IN, ns utocnikuv.example.net
  ▽ Additional records
    ▷ utocnikuv.example.net: type A, class IN, addr 1.2.3.4
```

Προκειμένου ο εισβολέας να ξεκινήσει την αποστολή πλαστών απαντήσεων, πρέπει να υπάρχει ένα αναδρομικό ερώτημα NS για τον συγκεκριμένο τομέα. Επομένως, ο εισβολέας στέλνει ένα ερώτημα DNS στο αναδρομικό NS, τότε ένα αναδρομικό NS βρίσκει έγκυρα NS για τον δεδομένο τομέα και στέλνει ένα ερώτημα σε ένα από αυτά. Με αυτήν την τελευταία ερώτηση ανοίγει ένα "παράθυρο ευκαιρίας" για την παράδοση παραποιημένων απαντήσεων και κλείνει με την έγκυρη απάντηση από ένα έγκυρο NS. Εάν μια παραποιημένη απόκριση παραδοθεί πριν από την έγκυρη απόκριση NS και ικανοποιεί όλες τις απαιτήσεις που αναφέρονται παραπάνω ταυτόχρονα, η μνήμη cache του αναδρομικού NS είναι δηλητηριασμένη και η επίθεση είναι επιτυχής.

Αυτός ο τύπος επίθεσης είναι αρκετά παλαιός και άμα εξαιρέσουμε ότι τα τρωτά σημεία σε ένα NS έχουν κατα μεγάλο βαθμό διορθωθεί, είναι επίσης εξαιρετικά ανεπαρκής. Η αιτιολογία σε όλο αυτό βρίσκεται στο ότι ένα αναδρομικό NS θυμάται το αποτέλεσμα της απάντησης και μια περαιτέρω επίθεση είναι δυνατή μόνο όταν λήξει η ισχύς (χρόνος ζωής) του αποθηκευμένου αρχείου.[17]

3.4 ARP Poisoning-Spoofing

Ενώ ορισμένοι διακόπτες (switches) αποτυγχάνουν κάτω από συνθήκες ενός MAC flooding και βρίσκονται σε μια κατάσταση όπου στέλνουν όλη την ιντερνετική κυκλοφορία σε όλα τα συστήματα στο LAN ορισμένοι άλλοι διακόπτες δεν το κάνουν. Κατά τη διάρκεια μιας πλημμύρας, αυτοί οι διακόπτες θυμούνται το αρχικό σύνολο διευθύνσεων MAC που εντοπίστηκαν αυτόματα στο τοπικό δίκτυο και χρησιμοποιούν αυτές τις διευθύνσεις καθ' όλη τη διάρκεια της πλημμύρας. Ο επιτιθέμενος δεν μπορεί να ξεκινήσει ένα MAC flooding για να θέσει τον διακόπτη σε κατάσταση αποστολή της κυκλοφορίας σε όλα τα συστήματα. Ωστόσο, ένας εισβολέας μπορεί ακόμα να υπονομεύσει ένα τέτοιο τοπικό δίκτυο LAN, εισάγοντας έναν άλλο τύπο κίνησης που βασίζεται στο πρωτόκολλο επίλυσης διευθύνσεων (Address Resolution Protocol – ARP).

Το ARP χρησιμοποιείται για τη χαρτογράφηση διευθύνσεων πρωτοκόλλου Internet (IP) σε διευθύνσεις MAC σε ένα τοπικό δίκτυο. Όταν ένα μηχάνημα διαθέτει δεδομένα για αποστολή σε άλλο σύστημα στο τοπικό δίκτυο, σχηματίζει ένα πακέτο για τη διεύθυνση IP του προορισμού. Ωστόσο, η διεύθυνση IP είναι απλώς μια ρύθμιση διαμόρφωσης στο μηχάνημα προορισμού. Πώς όμως η μηχανή αποστολής με το πακέτο για παράδοση καθορίζει την κατεύθυνση της συσκευής υλικού στο τοπικό δίκτυο για να στείλει το πακέτο; Η απάντηση είναι το πρωτόκολλο ARP.

Ας υποθέσουμε ότι ένα μηχάνημα στο LAN έχει ένα πακέτο που προορίζεται για διεύθυνση IP 10.1.2.3. Το μηχάνημα με το πακέτο θα στείλει ένα αίτημα ARP στο τοπικό δίκτυο, ζητώντας ποια διεπαφή δικτύου συσχετίζεται με τη διεύθυνση IP 10.1.2.3. Το μηχάνημα με αυτή τη διεύθυνση IP θα μεταδώσει μια απάντηση ARP ότι "η διεύθυνση IP 10.1.2.3 συνδέεται με τη διεύθυνση MAC AA: BB: CC: DD: EE: FF." Όταν ένα σύστημα λαμβάνει μια απάντηση ARP, αποθηκεύει τη χαρτογράφηση της διεύθυνσης IP στη διεύθυνση MAC σε έναν τοπικό πίνακα, που ονομάζεται πίνακας ARP, για μελλοντική αναφορά. Στη συνέχεια, το πακέτο θα παραδοθεί στη διασύνδεση δικτύου με αυτή τη διεύθυνση MAC. Με αυτό τον τρόπο, το ARP χρησιμοποιείται για τη μετατροπή διευθύνσεων IP σε διευθύνσεις MAC έτσι ώστε τα πακέτα να μπορούν να μεταφερθούν στην κατάλληλη διεπαφή δικτύου στο τοπικό δίκτυο. Τα αποτελέσματα αποθηκεύονται στον πίνακα ARP του συστήματος για να ελαχιστοποιηθεί η ανάγκη για επιπλέον κίνηση ARP σε ένα τοπικό δίκτυο.

Το ARP περιλαμβάνει υποστήριξη για μια δυνατότητα που ονομάζεται "gratuitous ARP". Με ένα gratuitous ARP, μια μηχανή μπορεί να στείλει μια απάντηση ARP παρόλο που καμία μηχανή δεν έστειλε ένα αίτημα ARP. Τα περισσότερα συστήματα 'διψούν' για εγγραφές στους πίνακες ARP τους έτσι ώστε να βελτιώσουν την απόδοση στο τοπικό δίκτυο. Σε μια άλλη μορφή ενεργού sniffing, ένας εισβολέας χρησιμοποιεί ψευδή μηνύματα ARP για ανακατεύθυνση της κινήσεως δεδομένων μέσα σε ένα LAN.

Αναφέρονται τα βήματα μιας τέτοιας επίθεσης για να κατανοήσουμε την εικόνα παρακάτω όπου ο επιτιθέμενος παρουσιάζεται ως χρήστης με ένα μαύρο καπέλο :

1. Ο εισβολέας ενεργοποιεί την προώθηση IP στη μηχανή του στο τοπικό δίκτυο. Οποιοδήποτε πακέτο κατευθύνεται από το διακόπτη προς το καπέλο θα μεταφερθεί στον προεπιλεγμένο δρομολογητή για το τοπικό δίκτυο.

2. Ο εισβολέας στέλνει ένα μήνυμα gratuitous ARP στο μηχάνημα-στόχο. Ο επιτιθέμενος θέλει να κάνει λήψη της κίνησης που αποστέλλεται από αυτό το μηχάνημα στον έξω κόσμο. Το δωρεάν μήνυμα ARP θα αντιστοιχίσει τη διεύθυνση IP του προεπιλεγμένου δρομολογητή για το LAN στη διεύθυνση MAC της μηχανής του εισβολέα. Το μηχάνημα-στόχος δέχεται αυτό το ψεύτικο μήνυμα ARP και το εισάγει στον πίνακα ARP του. Ο πίνακας ARP του στόχου έχει πλέον 'δηλητηριαστεί' με την ψευδή καταχώρηση.

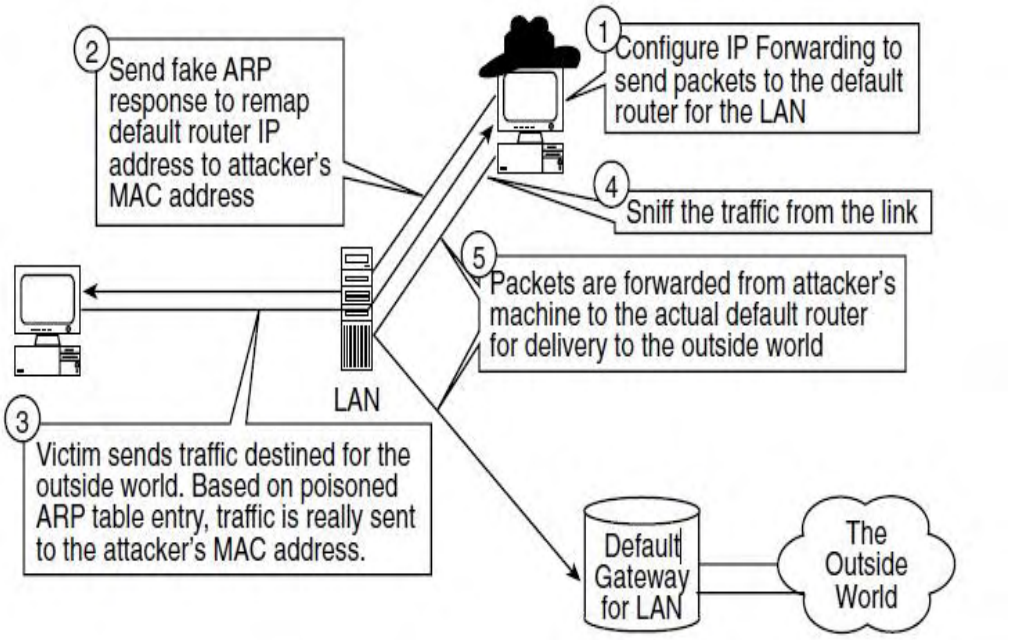
3. Το μηχάνημα-στόχος στέλνει κίνηση προορισμένη για τον έξω κόσμο. Συμβουλεύει τον πίνακα του ARP για να προσδιορίσει τη διεύθυνση MAC που σχετίζεται με τον προεπιλεγμένο δρομολογητή για το τοπικό δίκτυο. Η διεύθυνση MAC που βρίσκει στον πίνακα ARP είναι η διεύθυνση του εισβολέα. Όλα τα δεδομένα για τον έξω κόσμο αποστέλλονται στο μηχάνημα του εισβολέα.

4. Ο επιτιθέμενος κάνει λήψη της κίνησης από τη γραμμή.

5. Η προώθηση IP που ενεργοποιήθηκε στο Βήμα 1 ανακατευθύνει όλη την κίνηση από το μηχάνημα του εισβολέα στον προεπιλεγμένο δρομολογητή για το τοπικό δίκτυο. Ο προεπιλεγμένος δρομολογητής προωθεί την κίνηση στον εξωτερικό κόσμο. Με αυτόν τον τρόπο, το θύμα θα είναι σε θέση να στείλει κυκλοφορία στον έξω κόσμο, αλλά θα περάσει από το μηχάνημα του επιτιθέμενου ώστε αυτό να την καταγράψει κατά την έξοδο του.

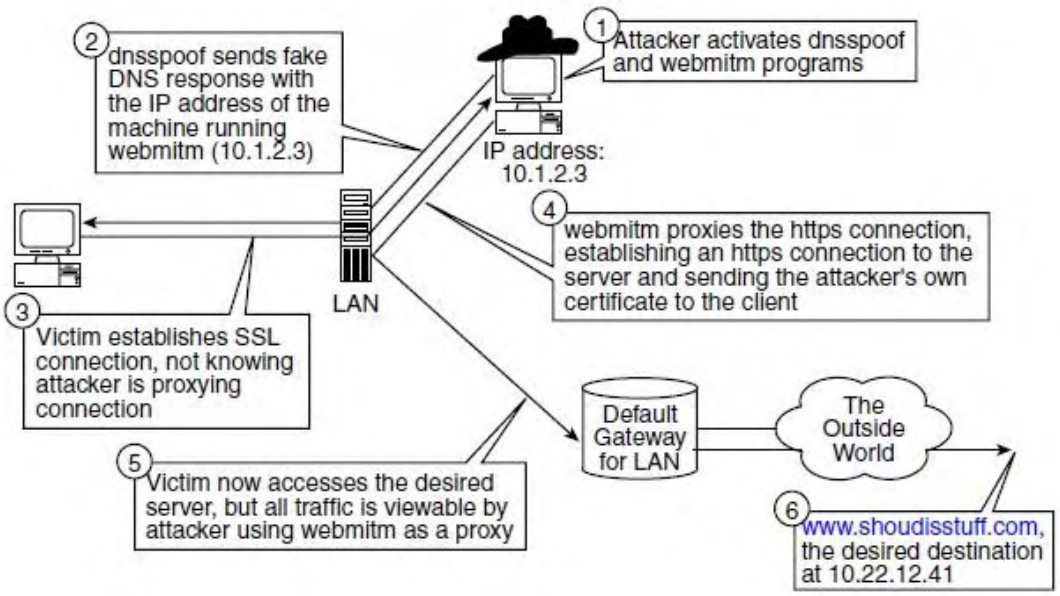
Αυτή η παραπάνω ακολουθία βημάτων επιτρέπει στον εισβολέα να βλέπει όλη την κίνηση πακέτων προς τον έξω κόσμο από το σύστημα προορισμού. Να

τονιστεί ότι για αυτή την τεχνική, ο εισβολέας δεν τροποποιεί καθόλου τον διακόπτη. Ο επιτιθέμενος είναι σε θέση να καταγράψει την κίνηση στο LAN χειριζόμενος τον πίνακα ARP του θύματος και επειδή η κυκλοφορία ARP και οι σχετικές πληροφορίες διεύθυνσης MAC μεταδίδονται μόνο μέσω LAN, αυτή η τεχνική λειτουργεί μόνο εάν ο εισβολέας ελέγχει ένα μηχάνημα στο ίδιο LAN με το σύστημα προορισμού.



3.5 Middle-Man SSL Attack

Η έγχυση-τοποθέτηση ψεύτικων αποκρίσεων DNS σε ένα δίκτυο είναι μια ιδιαίτερα ισχυρή τεχνική όταν χρησιμοποιείται για να δημιουργήσει μια επίθεση ενάντια στα κρυπτογραφικά πρωτόκολλα όπως το SSL, το οποίο χρησιμοποιείται συνήθως για ασφαλή πρόσβαση στο διαδίκτυο. Ουσιαστικά, ο εισβολέας στέλνει μια ψεύτικη απάντηση DNS στον στόχο, έτσι ώστε να δημιουργηθεί μια νέα περίοδος SSL μέσω της μηχανής του εισβολέα. Όπως επισημαίνεται στην εικόνα πιο κάτω ο εισβολέας χρησιμοποιεί ένα εξειδικευμένο εργαλείο αναμετάδοσης για να δημιουργήσει δύο κρυπτογραφικές συνεδρίες: μία μεταξύ του πελάτη και του εισβολέα και άλλη μία μεταξύ του εισβολέα και του διακομιστή. Ενώ τα δεδομένα μετακινούνται μεταξύ αυτών των δύο περιόδων σύνδεσης ο εισβολέας μπορεί να καταγράψει την ισχύουσα κυκλοφορία πακέτων και να την επεξεργαστεί. Όπως και πριν θα αναφέρουμε τα συγκεκριμένα στάδια για αυτή την επίθεση ώστε να γίνει απλούστερη στην κατανόηση :



1. Ο εισβολέας ενεργοποιεί ένα πρόγραμμα όπως για παράδειγμα το dnsspoof του Dsniff, ένα εργαλείο που στέλνει ψεύτικες απαντήσεις DNS. Επιπλέον, ο εισβολέας ενεργοποιεί ένα άλλο εργαλείο Dsniff που ονομάζεται "webmitm", μια συντομογραφία για το Web Man-in-the-Middle. Αυτό το εργαλείο υλοποιεί μια εξειδικευμένη σύνδεση SSL.

2. Ο επιτιθέμενος παρατηρεί ένα ερώτημα DNS από το μηχάνημα του θύματος και στέλνει μια ψεύτικη απάντηση DNS. Η ψεύτικη απάντηση DNS περιέχει τη διεύθυνση IP της μηχανής του εισβολέα.

3. Το θύμα λαμβάνει την απάντηση DNS και δημιουργεί μια περίοδο σύνδεσης SSL με τη διεύθυνση IP που περιλαμβάνεται στην απάντηση.

4. Το εργαλείο webmitm που εκτελείται στη μηχανή του εισβολέα δημιούργησε μια περίοδο SSL με το μηχάνημα του θύματος και μια άλλη περίοδο SSL με τον πραγματικό διακομιστή Web στον οποίο ο πελάτης θέλει να έχει πρόσβαση.

5. Το θύμα στέλνει δεδομένα μέσω της σύνδεσης SSL. Το εργαλείο webmitm αποκρυπτογραφεί την κυκλοφορία από τη σύνδεση SSL του θύματος, το εμφανίζει στον εισβολέα και κρυπτογραφεί την κίνηση για τη μετάβαση στον εξωτερικό διακομιστή Web. Ο εξωτερικός διακομιστής Web λαμβάνει την κίνηση, χωρίς να συνειδητοποιήσει ότι συμβαίνει μια επίθεση στη μηχανή-στόχο.

Ενώ αυτή η τεχνική είναι αρκετά αποτελεσματική, έχει έναν περιορισμό για την μεριά του εισβολέα. Κατά την εγκατάσταση της σύνδεσης SSL μεταξύ του θύματος και της μηχανής του εισβολέα, ο εισβολέας πρέπει να στείλει στο θύμα ψηφιακό πιστοποιητικό SSL που ανήκει στον εισβολέα. Για να αποκρυπτογραφήσει όλα τα δεδομένα που αποστέλλονται από το στόχο, ο εισβολέας πρέπει να χρησιμοποιήσει το δικό του ψηφιακό πιστοποιητικό και όχι το πιστοποιητικό από τον πραγματικό διακομιστή Web προορισμού. Όταν ο περιηγητής ιστού του θύματος λάβει το πλαστό πιστοποιητικό από τον εισβολέα, θα εμφανίσει ένα προειδοποιητικό μήνυμα στον χρήστη. Το πρόγραμμα περιήγησης θα υποδείξει ότι το πιστοποιητικό που παρουσίασε ο διακομιστής υπογράφηκε από μια αρχή έκδοσης πιστοποιητικών, η οποία δεν είναι αξιόπιστη από το πρόγραμμα περιήγησης. Το πρόγραμμα περιήγησης δίνει στον χρήστη τη δυνατότητα να δημιουργήσει τη σύνδεση κάνοντας απλά κλικ σε ένα κουμπί με την ένδειξη "OK" ή "Connect." Οι περισσότεροι χρήστες δεν κατανοούν τα προειδοποιητικά μηνύματα από τα προγράμματα περιήγησης και θα συνεχίσουν τη σύνδεση χωρίς δεύτερη σκέψη. Ο περιηγητής θα είναι ικανοποιημένος ότι έχει δημιουργήσει μια ασφαλή σύνδεση επειδή ο χρήστης του είπε να δεχτεί το πιστοποιητικό του εισβολέα. Μετά τη συνέχιση της σύνδεσης, ο επιτιθέμενος θα είναι σε θέση να συγκεντρώσει όλη την κίνηση από τη περίοδο σύνδεσης SSL. Στην ουσία, ο επιτιθέμενος βασίζεται στο γεγονός ότι οι αποφάσεις εμπιστοσύνης σχετικά με τα πιστοποιητικά SSL παραμένουν στα χέρια του χρήστη.

Η ίδια βασική τεχνική λειτουργεί και ενάντια στο πρωτόκολλο Secure Shell (SSH) που χρησιμοποιείται για την πρόσβαση σε κάποιο μηχάνημα μέσω απομακρυσμένων εντολών. Παρομοίως με την επίθεση SSL, ένα εργαλείο μπορεί να δημιουργεί δύο συνδέσεις SSH: μία μεταξύ του θύματος και του εισβολέα και μία άλλη μεταξύ του εισβολέα και του διακομιστή προορισμού. Επίσης, όπως και ο περιηγητής ιστού 'διαμαρτυρήθηκε' για το τροποποιημένο πιστοποιητικό SSL, ο πελάτης SSH θα παραπονεθεί ότι δεν αναγνωρίζει το δημόσιο κλειδί που χρησιμοποιείται από το διακομιστή SSH. Ωστόσο, ο πελάτης SSH θα επιτρέψει στον χρήστη να αντικαταστήσει την προειδοποίηση και να δημιουργήσει τη συνεδρία SSH έτσι ώστε ο εισβολέας να μπορεί να δει όλη την κίνηση και να την καταγράψει.[9]

ΚΕΦΑΛΑΙΟ 4

Στα πλαίσια της διπλωματικής αυτής εργασίας παράξαμε κώδικα στην ευρύτερα κοινή γλώσσα προγραμματισμού C ώστε να μπορούμε να δούμε στην πράξη την προσομοίωση της λειτουργίας ενός sniffer και να κατανοήσουμε καλύτερα τα όσα προαναφέρθηκαν στα προηγούμενα κεφάλαια, όπως και να μπορεί ο οποιοσδήποτε άνθρωπος, ακόμη και άσχετος με τον τομέα της πληροφορικής, να αντιληφθεί την δύναμη που παρέχεται σε έναν χρήστη, ο οποίος στην περίπτωση μας είναι ένας χειριστής-sniffer και μπορεί να υποκλέψει κάθε είδους ασύρματη και ενσύρματη επικοινωνία μεταξύ υπολογιστών και να την επεξεργαστεί με μέσα που διαθέτει.

Για την προσομοίωση του sniffer χρησιμοποιήσαμε sockets τα οποία παρέχονται σε βιβλιοθήκες που διατίθενται στα λειτουργικά συστήματα Linux και μας δίνουν αρκετές δυνατότητες μέσω συναρτήσεων για την υλοποίηση διαφόρων ειδών project. Στη συνέχεια θα αναφερθούμε σε πληροφορίες για τα sockets όπως επίσης θα γίνει μια πιο λεπτομερειακή παρουσίαση σε σημεία του προγράμματος μας στα οποία αξίζει να δώσουμε προσοχή ώστε να δούμε τα αποτελέσματα και τη ροή της πληροφορίας σε μια επικοινωνία καθώς και τον τρόπο που αποθηκεύεται στο υπολογιστικό σύστημα.

4.1 Τί είναι τα Sockets.

Για κάποιον μη ειδικό με το αντικείμενο ένα socket μπορεί να παρομοιαστεί με ένα τηλέφωνο, δηλαδή ένα πράγμα το οποίο δίνει τη δυνατότητα να υπάρξει επικοινωνία με ένα άλλο τηλέφωνο. Οι περισσότερες συνδέσεις με τηλέφωνα γίνονται μεταξύ δυο χρηστών ενώ συνδέσεις με sockets γίνονται ανάμεσα σε ένα χρήστη και ένα διακομιστή. Ένα βασικό χαρακτηριστικό τους είναι ότι όταν μια σύνδεση με socket κλείσει τότε το socket καταστρέφεται χωρίς να μπορεί να επαναχρησιμοποιηθεί οδηγώντας στην αναγκαιότητα δημιουργίας ενός νέου socket για τη δημιουργία μιας καινούργιας σύνδεσης. Εάν το εξετάσουμε αναλογικά αυτό δεν δείχνει να είναι και τόσο κακό καθώς ένας διακομιστής αποτελεί ένα τηλεφωνικό κέντρο που μπορεί να δεχθεί πλήθος εκατοντάδων κλήσεων την κάθε χρονική στιγμή.

Μια δεύτερη προσέγγιση για τον ορισμό του socket είναι ότι αποτελεί ένα μέσο για επικοινωνία, μπορεί να υπάρξει στο στρώμα πρωτοκόλλου μεταφοράς (transportation layer) καθώς και το ότι υπάρχει η δυνατότητα αποστολής και πρόσληψης δεδομένων σε αυτό με τελευταία και βασικότερη την ικανότητα

πρόσδεσης σε αυτό και καταγραφής της επικοινωνίας. Ένα socket είναι συγκεκριμένο για ένα πρωτόκολλο, ένα μηχάνημα ή μια θύρα και μπορεί να αναφέρεται στην κεφαλίδα ενός διαδικτυακού πακέτου δεδομένων.

Τέλος, ένα socket με απλά λόγια είναι ένα ψευδοαρχείο που αντιπροσωπεύει μια διαδικτυακή σύνδεση. Με την δημιουργία ενός socket, αφού πρώτα χρησιμοποιήσουμε τα κατάλληλα αρχέτυπα και τις κατάλληλες παραμέτρους ώστε να προσδιορίσουμε τον άλλον υπολογιστή-χρήστη μας, οι οποιεσδήποτε εγγραφές δεδομένων μετατρέπονται σε διαδικτυακά πακέτα τα οποία αποστέλλονται και αντίστοιχα τα δεδομένα μπορούν να διαβαστούν από ένα socket. Από μια οπτική γωνία τα sockets μπορούν να παρομοιαστούν με σωλήνες (pipes) οι οποίοι επιτρέπουν την επικοινωνία ενός προγράμματος με ένα άλλο, τοπικά εν μέρει στην περίπτωση των σωληνών και απομακρυσμένα στην δική μας περίπτωση των sockets. Επίσης προσφέρουν την δυνατότητα της αμφίδρομης επικοινωνίας όπως ακριβώς θα παρείχαν και δυο ενωμένοι μεταξύ τους σωλήνες. Όπως αναφέρει και η IEEE στην ιστορική αναδρομή περί συστημάτων BSD-Linux, οι σωλήνες ανακαλύφθηκαν νωρίτερα από τα sockets και επανεμφανίστηκαν χρησιμοποιώντας sockets μόλις αυτά επινοήθηκαν στον κόσμο των υπολογιστών.

4.2 Ανάλυση της εφαρμογής Sniffer

Έχουμε δημιουργήσει κώδικα σε γλώσσα C ώστε να μπορεί να προσομοιώσει τη λειτουργία ενός ωτακουστή δεδομένων – sniffer χρησιμοποιώντας ως βασική αρχή τα sockets. Το πρόγραμμά μας δείχνει τη βασική λειτουργία ενός πραγματικού sniffer, δηλαδή δημιουργούμε ένα socket ώστε να μπορέσουμε να παραλάβουμε όλη την πληροφορία σε μια υπάρχουσα διαδικτυακή σύνδεση μεταξύ δύο υπολογιστών, την επεξεργαζόμαστε ώστε να έχουμε ένα ικανοποιητικό αποτέλεσμα και την αποθηκεύουμε σε ένα αρχείο για ενδεχομένως μεταγενέστερη ανάλυση στο μέλλον ή πιθανή είσοδο σε κάποιο άλλο πρόγραμμα επεξεργασίας πληροφοριών. Φυσικά, για να φτάσουμε στο επιθυμητό αποτέλεσμα είναι σίγουρα λογικό και απαραίτητο, όπως σχεδόν σε όλα τα προγράμματα υπολογιστών, η χρήση ορισμένων έτοιμων βιβλιοθηκών που περιέχουν έτοιμες πληροφορίες, ρουτίνες και συναρτήσεις.

4.2.1 Βιβλιοθήκες που χρησιμοποιήθηκαν

Όπως στα περισσότερα προγράμματα C χρησιμοποιήσαμε τις βασικές βιβλιοθήκες **#include<stdio.h>, #include<stdlib.h>, #include<string.h>** στις οποίες ορίζονται τύποι μεταβλητών, διάφορες μακροεντολές και διάφορες λειτουργίες για την εκτέλεση εισόδου και εξόδου του προγράμματος.

Για να μπορέσουμε να δημιουργήσουμε ένα socket θα πρέπει να συμπεριλάβουμε τις βιβλιοθήκες : **#include <sys/socket.h>** και **#include <sys/types.h>**. Η πρώτη βιβλιοθήκη μας παρέχει τη δυνατότητα ενός socket μέσω μιας συνάρτησης : **int socket(int** *domain***, int** *type***, int** *protocol***)** όπου το όρισμα domain αποτελεί το πρωτόκολλο που θα χρησιμοποιηθεί για την επικοινωνία και αυτά τα πρωτόκολλα ορίζονται μέσα στην ίδια βιβλιοθήκη με κάποια βασικά να είναι τα παρακάτω :

AF_INET: IPv4 Πρωτόκολλα
AF_INET6: IPv6 Πρωτόκολλα
AF_APPLETALK: AppleTalk
AF_PACKET: Διασύνδεση πακέτων χαμηλού επιπέδου

Το όρισμα type αναφέρεται στη σημασιολογία της επικοινωνίας ενώ το protocol στο πρωτόκολλο χρήσης στο socket μας. Η συνάρτηση αυτή θα μας επιστρέψει ένα αρχείο με κάθε επιτυχία ενώ -1 με μια αποτυχία και κατάλληλη ρύθμιση της errno συνάρτησης για διαχείρηση σφαλμάτων στο πρόγραμμα μέσω χρήσης της βιβλιοθήκης **#include<errno.h>**. Ετσι εμείς δημιουργούμε ένα socket με αντίστοιχες διαθέσιμες παραμέτρους: **int sock_raw = socket(AF_PACKET , SOCK_RAW , htons(ETH_P_ALL)) ;** . Χρησιμοποιούμε την συνάρτηση **htons()** για μετατροπή ενός unsigned short ακέραιου μεγέθους 16 bits σε network byte order δηλαδη σε big-endian ετσι ώστε να μην υπάρχει πρόβλημα με τις διάφορες αρχιτεκτονικές υπολογιστών.

Χρησιμοποιούμε επίσης την βιβλιοθήκη **#include <netinet/in.h>** στην οποία οι ακόλουθοι τύποι που ορίζονται μέσω του typedef :
in_port_t: Ένας unsigned ολοκληρωμένος τύπος ακριβώς 16 bit.
In_addr_t: Ένας unsigned ολοκληρωμένος τύπος ακριβώς 32 bit.
Επίσης το συγκεκριμένο header ορίζει τη δομή **sockaddr_in** που περιλαμβάνει τουλάχιστον το ακόλουθο μέρος :
sa_family_t: sin_family
in_port_t: sin_port
struct in_addr: sin_addr
unsigned char: sin_zero[8]
Η δομή **sockaddr_in** χρησιμοποιείται για την αποθήκευση διευθύνσεων πρωτοκόλλου Internet. Χρησιμοποιούμε τις βιβλιοθήκες **#include <sys/types.h>, #include<unistd.h>, #include<arpa/inet.h>** για διαχείριση buffer, διάφορων τύπων και διευθύνσεων ίντερνετ. Τέλος γίνεται χρήση ορισμένων βιβλιοθηκών που περιέχουν ορισμούς και περιγραφή των αντίστοιχων πρωτοκόλλων και διαχείριση των αντίστοιχων headers και ιντερνετικών διευθυνσιοδοτήσεων:
#include<netinet/ip_icmp.h>: Ορισμοί για πρωτόκολλο ICMP
#include<netinet/udp.h>: Ορισμοί για πρωτόκολλο UDP
#include<netinet/tcp.h>: Ορισμοί για πρωτόκολλο TCP
#include<netinet/ip.h>: Ορισμοί για πρωτόκολλο IPV4

#include<netinet/if_ether.h>: Ορισμοί για πρωτόκολλο ETHERNET
#include<net/ethernet.h>: Ορισμοί για πρωτόκολλο ETHERNET

4.2.2 Ανάλυση Και Περιγραφή Των Μερών Της Εφαρμογής

Πριν την βασική μας συνάρτηση main() του προγράμματος ξεκινούμε ορίζοντας δύο μεταβλητές τύπου **struct sockaddr_in** όπου το συγκεκριμένο struct:

```
struct sockaddr_in {
   short          sin_family;   // e.g. AF_INET
   unsigned short   sin_port;     // e.g. htons(3490)
   struct in_addr   sin_addr;     // see struct in_addr, below
   char           sin_zero[8];  // zero this if you want to
};

struct in_addr {
   unsigned long s_addr;  // load with inet_aton()
};
```

χρησιμοποιείται για χειρισμό διευθύνσεων ίντερνετ και έχει την παρακάτω μορφή καθώς και τον ορισμό βιβλιοθηκών και συναρτήσεων όπως φαίνεται παρακάτω:

```
#include<netinet/in.h>
#include<errno.h>
#include<netdb.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<netinet/ip_icmp.h>
#include<netinet/udp.h>
#include<netinet/tcp.h>
#include<netinet/ip.h>
#include<netinet/if_ether.h>
#include<net/ethernet.h>
#include<sys/socket.h>
#include<arpa/inet.h>
#include<sys/types.h>
#include<unistd.h>
```

```
void ProcessPacket(unsigned char* , int);
void print_ip_header(unsigned char* , int);
void print_tcp_packet(unsigned char * , int );
void print_udp_packet(unsigned char * , int );
void print_icmp_packet(unsigned char* , int );
void PrintData (unsigned char* , int);

FILE *logfile;
struct sockaddr_in source,dest;
int Tcp=0,Udp=0,icmp=0,others=0,TotalFrames=0,i,j;
```

Αντίστοιχα, μέσα στη main() ορίζουμε παρομοίως δύο μεταβλητές τύπου **struct sockaddr** το οποίο περιέχει τα εξής πεδία ενώ δεσμεύουμε και χώρο στη μνήμη μας μέσω της κλασσικής συνάρτησης malloc() μεγέθους 1024 bytes που θα μπορούσε να μειωθεί ανάλογα με τις ανάγκες του καθενός:

```
struct sockaddr {
   unsigned short    sa_family;    // address family, AF_xxx
   char              sa_data[14];  // 14 bytes of protocol address
};
```

```
int main()
{
    int addr_size , data_size;
    struct sockaddr saddr;

    unsigned char *buffer = (unsigned char *) malloc(1024);

    logfile=fopen("snifflog.txt","w");
    if(logfile==NULL)
    {
        printf("Unable to create the snifflog.txt file.");
    }

    printf("Sniffing Packets Right Now...\n");
    printf("Push Ctrl + C If You Wish To Stop!!!\n");


    int sock_raw = socket( AF_PACKET , SOCK_RAW , htons(ETH_P_ALL)) ;

    if(sock_raw < 0)
    {
        perror("Socket Error");
        return 1;
    }
    while(1)
    {
        addr_size = sizeof(saddr);
        data_size = recvfrom(sock_raw , buffer , 1024 , 0 , &saddr , (socklen_t*)& addr_size);
        if(data_size <0 )
        {
            printf("Recvfrom error , failed to get packets\n");
            return 1;
        }
        ProcessPacket(buffer , data_size);
    }
    close(sock_raw);
    printf("Finished");
    return 0;
}
```

Εν συνεχεία, ανοίγουμε ένα αρχείο το οποίο στην περίπτωση μας ονομάζουμε **snifflog.txt** το οποίο αποτελεί και το μέσο αποθήκευσης των πληροφοριών που θα συλλέξουμε όταν θα τρέξουμε το πρόγραμμα μας για πιθανόν μελλοντική επεξεργασία. Παρακάτω όπως ο καθένας θα μπορούσε να υποψιαστεί δημιουργούμε το socket μας και είμαστε έτοιμοι να συλλέξουμε δεδομένα, μέσω μιας επαναληπτικής ρουτίνας, με τη βοήθεια της συνάρτησης **recvfrom** η οποία θα επιστρέψει τον αριθμό των bytes που έχουν ληφθεί πραγματικά ή σε οποιαδήποτε άλλη περίπτωση -1 περνώντας τις αντίστοιχες παραμέτρους στη συνάρτηση errno. Επιπλέον καλούμε μια συνάρτηση που έχουμε δημιουργήσει για μια σχετικά απλή επεξεργασία των δεδομένων που έχουν ληφθεί από το socket μας που την ονομάζουμε **ProcessPacket(buffer,data_size)**. Εμφανίζουμε την επαναληπτική ρουτίνα η οποία μπορούμε να πούμε ότι τρέχει επ αόριστον :

```
while(1)
{
    addr_size = sizeof(saddr);
    data_size = recvfrom(sock_raw , buffer , 1024 , 0 , &saddr , (socklen_t*)& addr_size);
    if(data_size <0 )
    {
        printf("Recvfrom error , failed to get packets\n");
        return 1;
    }
    ProcessPacket(buffer , data_size);
}
```

Μέσα στη συνάρτηση ProcessPacket δημιουργούμε ένα **struct iphdr *iph = (struct iphdr*)(buffer + sizeof(struct ethhdr))** για να πάρουμε κατά βάση μόνο το μέρος της κεφαλίδας IPv4 του εκάστωτε πακέτου και όχι όλη την κεφαλίδα Ethernet και μέσω ενός switch-case ελέγχουμε τον τύπο πρωτοκόλλου του πακέτου με τη βοήθεια του πεδίου του struct μας **protocol,** όπου μπορούμε να δόυμε στην παρακάτω εικόνα μια λίστα με το ποιος αριθμός αντιστοιχεί σε ποιό πρωτόκολλο. Φυσικά η λίστα είναι πολύ μεγαλύτερη από ότι παρακάτω, απλά επιλέξαμε να δείξουμε τα πιο γνωστά και κοινότυπα κατά τον κόσμο πρωτόκολλα:

Protocol Number	Host-to-Host Layer Protocol
1	Internet Control Message Protocol (ICMP)
2	Internet Group Management Protocol (IGMP)
4	IP in IP (encapsulation)
6	Transmission Control Protocol (TCP)
17	User Datagram Protocol (UDP)
45	Inter-Domain Routing Protocol (IDRP)
46	Resource Reservation Protocol (RSVP)
47	Generic Routing Encapsulation (GRE)
54	NBMA Next Hop Resolution Protocol (NHRP)

Στη συνέχεια καλούμε κάθε φορά την εκάστοτε συνάρτηση για εμφάνιση των δεδομένων στο αρχείο καταγραφής που δημιουργήσαμε εξ αρχής. Ακόμη, κρατάμε και μετρητές της κάθε μορφής πρωτοκόλλου πακέτου καθώς και ένα συνολικό μετρητή που καταγράφθηκαν κατά τη διάρκεια εκτέλεσης του προγράμματος το οποίο και εμφανίζουμε στην οθόνη του τερματικού με τον τερματισμό του προγράμματος.

```
void ProcessPacket(unsigned char* buffer, int size)
{
    //Get the IP Header part of this packet , excluding the ethernet header
    struct iphdr *iph = (struct iphdr*)(buffer + sizeof(struct ethhdr));
    ++TotalFrames;
    switch (iph->protocol) //Check the Protocol and do accordingly fron protocol table...
    {
        case 1:  //ICMP Protocol
            ++icmp;
            print_icmp_packet( buffer , size);
            break;

        case 6:  //TCP Protocol
            ++Tcp;
            print_tcp_packet(buffer , size);
            break;

        case 17: //UDP Protocol
            ++Udp;
            print_udp_packet(buffer , size);
            break;

        default: //Some Other Protocol like ARP etc.
            ++others;
            break;
    }
    printf("TCP : %d   UDP : %d   ICMP : %d   Others : %d   Total : %d\r", Tcp ,Udp , icmp , others , TotalFrames);
}
```

Από εδώ και πέρα μέσα στις συναρτήσεις εκτύπωσης πακέτων όπως για παράδειγμα στην print_tcp_packet,όπως φαίνεται στην επόμενη εικόνα:

```
void print_tcp_packet(unsigned char* Buffer, int Size)
{
    unsigned short iphdrlen;

    struct iphdr *iph = (struct iphdr *)( Buffer  + sizeof(struct ethhdr) );
    iphdrlen = iph->ihl*4;

    struct tcphdr *tcph=(struct tcphdr*)(Buffer + iphdrlen + sizeof(struct ethhdr));

    int header_size =  sizeof(struct ethhdr) + iphdrlen + tcph->doff*4;

    fprintf(logfile , "\n\n***********************TCP Packet*************************\n");

    print_ip_header(Buffer,Size);

    fprintf(logfile , "\n");
    fprintf(logfile , "TCP Header\n");
    fprintf(logfile , "   |-Source Port      : %u\n",ntohs(tcph->source));
    fprintf(logfile , "   |-Destination Port : %u\n",ntohs(tcph->dest));
    fprintf(logfile , "   |-Sequence Number    : %u\n",ntohl(tcph->seq));
    fprintf(logfile , "   |-Acknowledge Number : %u\n",ntohl(tcph->ack_seq));
    fprintf(logfile , "   |-Header Length      : %d DWORDS or %d BYTES\n" ,(unsigned int)tcph->doff,(unsigned int)tcph->doff*4);
    fprintf(logfile , "   |-Urgent Flag          : %d\n",(unsigned int)tcph->urg);
    fprintf(logfile , "   |-Acknowledgement Flag : %d\n",(unsigned int)tcph->ack);
    fprintf(logfile , "   |-Push Flag            : %d\n",(unsigned int)tcph->psh);
    fprintf(logfile , "   |-Reset Flag           : %d\n",(unsigned int)tcph->rst);
    fprintf(logfile , "   |-Synchronise Flag     : %d\n",(unsigned int)tcph->syn);
    fprintf(logfile , "   |-Finish Flag          : %d\n",(unsigned int)tcph->fin);
    fprintf(logfile , "   |-Window         : %d\n",ntohs(tcph->window));
    fprintf(logfile , "   |-Checksum       : %d\n",ntohs(tcph->check));
    fprintf(logfile , "   |-Urgent Pointer : %d\n",tcph->urg_ptr);
    fprintf(logfile , "\n");
    fprintf(logfile , "                        DATA Dump                         ");
    fprintf(logfile , "\n");

    fprintf(logfile , "IP Header\n");
    PrintData(Buffer,iphdrlen);

    fprintf(logfile , "TCP Header\n");
    PrintData(Buffer+iphdrlen,tcph->doff*4);

    fprintf(logfile , "Data Payload\n");
    PrintData(Buffer + header_size , Size - header_size );

    fprintf(logfile , "\n###########################################################");
}
```

ορίζουμε αντίστοιχα structs τύπου πρωτοκόλλου που θέλουμε να επεξεργαστούμε, στην περίπτωση μας **struct tcphdr,** και εμφανίζουμε στο αρχείο καταγραφής δεδομενογράμματος οποιαδήποτε πεδία του struct μας δίνονται ή θέλουμε να χρησιμοποιήσουμε. Να τονίσουμε ότι πρίν την καταγραφή επεξεργασίας δεδομενογράμματος του πακέτου μέσω της συνάρτήσης PrintData(), την οποία θα αναφέρουμε σε παρακάτω σημείο, καλούμε την συνάρτηση **print_ip_header(Buffer , Size)** όπως φαίνεται ο κώδικας της:

```
void print_ip_header(unsigned char* Buffer, int Size)
{
    struct ethhdr *eth = (struct ethhdr *)Buffer;

    fprintf(logfile , "\n");
    fprintf(logfile , "Ethernet Header\n");
    fprintf(logfile , "   |-Destination Address : %.2X-%.2X-%.2X-%.2X-%.2X-%.2X \n", eth->h_dest[0] , eth->h_dest[1] ,
    eth->h_dest[2] , eth->h_dest[3] , eth->h_dest[4] , eth->h_dest[5] );
    fprintf(logfile , "   |-Source Address      : %.2X-%.2X-%.2X-%.2X-%.2X-%.2X \n", eth->h_source[0] , eth->h_source[1] ,
    eth->h_source[2] , eth->h_source[3] , eth->h_source[4] , eth->h_source[5] );
    fprintf(logfile , "   |-Protocol            : %u \n",(unsigned short)eth->h_proto);

    unsigned short iphdrlen;

    struct iphdr *iph = (struct iphdr *)(Buffer  + sizeof(struct ethhdr) );
    iphdrlen =iph->ihl*4;

    memset(&source, 0, sizeof(source));
    source.sin_addr.s_addr = iph->saddr;

    memset(&dest, 0, sizeof(dest));
    dest.sin_addr.s_addr = iph->daddr;

    fprintf(logfile , "\n");
    fprintf(logfile , "IP Header\n");
    fprintf(logfile , "   |-IP Version        : %d\n",(unsigned int)iph->version);
    fprintf(logfile , "   |-IP Header Length  : %d DWORDS or %d Bytes\n",(unsigned int)iph->ihl,((unsigned int)(iph->ihl))*4);
    fprintf(logfile , "   |-Type Of Service   : %d\n",(unsigned int)iph->tos);
    fprintf(logfile , "   |-IP Total Length   : %d  Bytes(Size of Packet)\n",ntohs(iph->tot_len));
    fprintf(logfile , "   |-Identification    : %d\n",ntohs(iph->id));
    fprintf(logfile , "   |-TTL      : %d\n",(unsigned int)iph->ttl);
    fprintf(logfile , "   |-Protocol : %d\n",(unsigned int)iph->protocol);
    fprintf(logfile , "   |-Checksum : %d\n",ntohs(iph->check));
    fprintf(logfile , "   |-Source IP        : %s\n",inet_ntoa(source.sin_addr));
    fprintf(logfile , "   |-Destination IP   : %s\n",inet_ntoa(dest.sin_addr));
}
```

Παραθέτουμε και τις υπόλοιπες συναρτήσεις για τα πακέτα icmp και udp:

```
void print_icmp_packet(unsigned char* Buffer , int Size)
{
    unsigned short iphdrlen;

    struct iphdr *iph = (struct iphdr *)(Buffer  + sizeof(struct ethhdr));
    iphdrlen = iph->ihl * 4;

    struct icmphdr *icmph = (struct icmphdr *)(Buffer + iphdrlen  + sizeof(struct ethhdr));

    int header_size =  sizeof(struct ethhdr) + iphdrlen + sizeof icmph;

    fprintf(logfile , "\n\n**********************ICMP Packet*************************\n");

    print_ip_header(Buffer , Size);

    fprintf(logfile , "\n");

    fprintf(logfile , "ICMP Header\n");
    fprintf(logfile , "   |-Type : %d",(unsigned int)(icmph->type));

    if((unsigned int)(icmph->type) == 11)
    {
        fprintf(logfile , "  (TTL Expired)\n");
    }
    else if((unsigned int)(icmph->type) == ICMP_ECHOREPLY)
    {
        fprintf(logfile , "  (ICMP Echo Reply)\n");
    }

    fprintf(logfile , "   |-Code : %d\n",(unsigned int)(icmph->code));
    fprintf(logfile , "   |-Checksum : %d\n",ntohs(icmph->checksum));
    fprintf(logfile , "\n");

    fprintf(logfile , "IP Header\n");
    PrintData(Buffer,iphdrlen);

    fprintf(logfile , "UDP Header\n");
    PrintData(Buffer + iphdrlen , sizeof icmph);

    fprintf(logfile , "Data Payload\n");

    //Move the pointer ahead and reduce the size of string
    PrintData(Buffer + header_size , (Size - header_size) );

    fprintf(logfile , "\n###########################################################");
}
```

```
void print_udp_packet(unsigned char *Buffer , int Size)
{

    unsigned short iphdrlen;

    struct iphdr *iph = (struct iphdr *)(Buffer +  sizeof(struct ethhdr));
    iphdrlen = iph->ihl*4;

    struct udphdr *udph = (struct udphdr*)(Buffer + iphdrlen  + sizeof(struct ethhdr));

    int header_size =  sizeof(struct ethhdr) + iphdrlen + sizeof udph;

    fprintf(logfile , "\n\n***********************UDP Packet*************************\n");

    print_ip_header(Buffer,Size);

    fprintf(logfile , "\nUDP Header\n");
    fprintf(logfile , "   |-Source Port      : %d\n" , ntohs(udph->source));
    fprintf(logfile , "   |-Destination Port : %d\n" , ntohs(udph->dest));
    fprintf(logfile , "   |-UDP Length       : %d\n" , ntohs(udph->len));
    fprintf(logfile , "   |-UDP Checksum     : %d\n" , ntohs(udph->check));

    fprintf(logfile , "\n");
    fprintf(logfile , "IP Header\n");
    PrintData(Buffer , iphdrlen);

    fprintf(logfile , "UDP Header\n");
    PrintData(Buffer+iphdrlen , sizeof udph);

    fprintf(logfile , "Data Payload\n");

    //Move the pointer ahead and reduce the size of string
    PrintData(Buffer + header_size , Size - header_size);

    fprintf(logfile , "\n###########################################################");
}
```

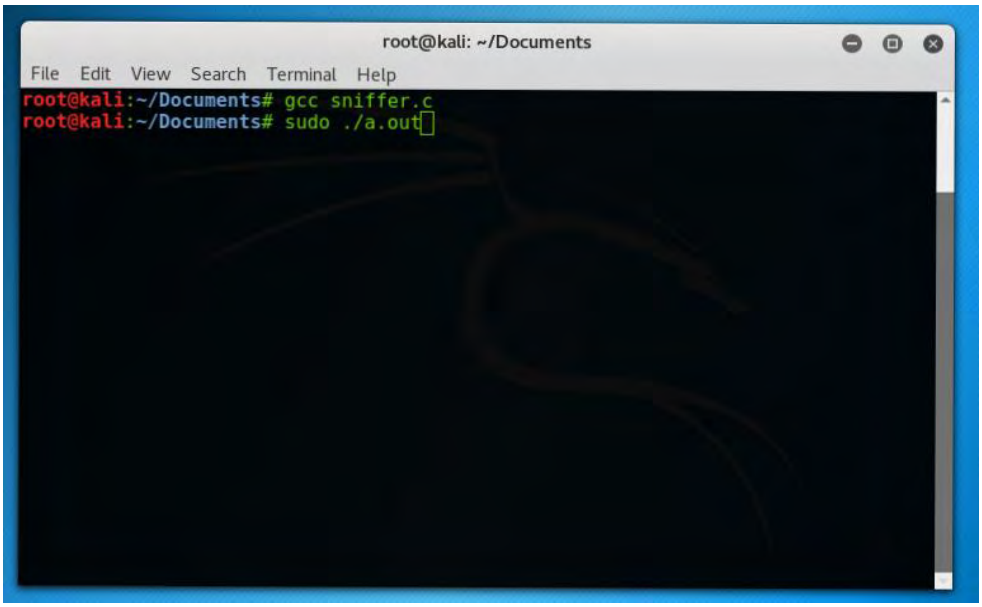
Τέλος, περνάμε στην καταγραφή δεδομενογράμματος στο αρχείο καταγραφής, που προκύπτουν από μια μικρή επεξεργασία των δεδομένων του κάθε πακέτου μέσω της συνάρτησης **void PrintData (unsigned char* data , int Size)**, η οποία είναι να ελέγχουμε κάθε γραμμή στοιχείων σε δεκαεξαδική μορφη και να εμφανίζουμε την αντιστοιχία των δεκαεξαδικών αυτών στοιχείων εφόσων αυτά αντιστοιχούν σε κάποιο γράμμα ή σύμβολο με βάση τον πίνακα ASCII . Στην περίπτωση μας ελέγχουμε για κωδικούς ASCII από 32 έως 128, δηλαδή για γράμματα της αλφαβήτου μικρά ή κεφαλαία καθώς και συμβολοχαρακτήρες έτσι ώστε να καταφέρουμε να αποκωδικοποιήσουμε τουλάχιστον ένα μέρος των δεδομένων του πακέτου που θα είναι ανθρωπίνως κατανοήσιμο. Εάν ο κωδικός ASCII είναι εκτός των δεδομένων ορίων που έχουμε δώσει το πρόγραμμα θα εμφανίσει μια τελεία (.) ώστε να είναι οπτικά πιο όμορφο το αποτέλεσμα. Ο κώδικάς της συνάρτησης φαίνεται παρακάτω :

```
void PrintData (unsigned char* data , int Size)
{
    int i , j;
    for(i=0 ; i < Size ; i++)
    {
        if( i!=0 && i%16==0)   //if one line of hex printing is complete...
        {
            fprintf(logfile , "         ");

        for(j=i-16 ; j<i ; j++)
                {
                if(data[j]>=32 && data[j]<=128)
                    fprintf(logfile , "%c",(unsigned char)data[j]); //print only if its a number or alphabet or special characters

                else fprintf(logfile , "."); //otherwise print a dot
                }

            fprintf(logfile , "\n");
        }

        if(i%16==0) fprintf(logfile , "   ");
            fprintf(logfile , " %02X",(unsigned int)data[i]);

        if( i==Size-1)  //print the last spaces
        {
            for(j=0;j<15-i%16;j++)
            {
              fprintf(logfile , "   "); //extra spaces to fill the gaps of hex line
            }

            fprintf(logfile , "         ");

            for(j=i-i%16 ; j<=i ; j++)
            {
                if(data[j]>=32 && data[j]<=128)
                {
                  fprintf(logfile , "%c",(unsigned char)data[j]);
                }
                else
                {
                  fprintf(logfile , ".");
                }
            }

            fprintf(logfile ,  "\n" );
        }
    }
}
```

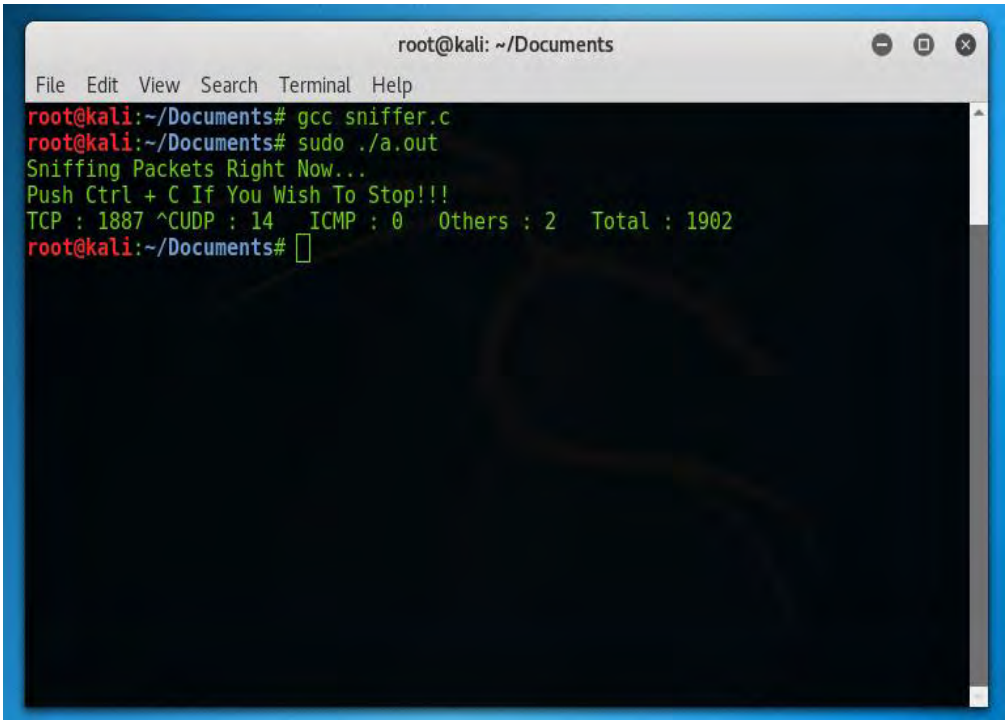
4.2.3 <u>Demo Της Εφαρμογής</u>

Καθότι το πρόγραμμα έχει παραχθεί σε γλώσσα C χρησιμοποιούμε gcc για να το μεταγλωττίσουμε σε ένα Linux machine στο τερματικό με την συνήθη εντολή **>gcc FileName.c** και στη συνέχεια για να καταφέρουμε να το τρέξουμε εκτελούμε την εντολή **>sudo ./a.out** μιας και απαιτούνται προνόμια διαχειριστή. Το πρόγραμμα θα αρχίσει να καταγράφει τα πακέτα και τα δεδομένα τους στο αρχείο καταγραφής κειμένου **Snifflog.txt** το οποίο μπορούμε να ανοίξουμε για μελέτη αφού σταματήσουμε το πρόγραμμα με CTRL + C. Παραθέτουμε εν συνεχεία ορισμένα screenshots από την εκτέλεση του προγράμματος έτσι ώστε να δούμε και στην πράξη τα όσα προείπαμε στο υποκεφάλαιο 4.2.2.

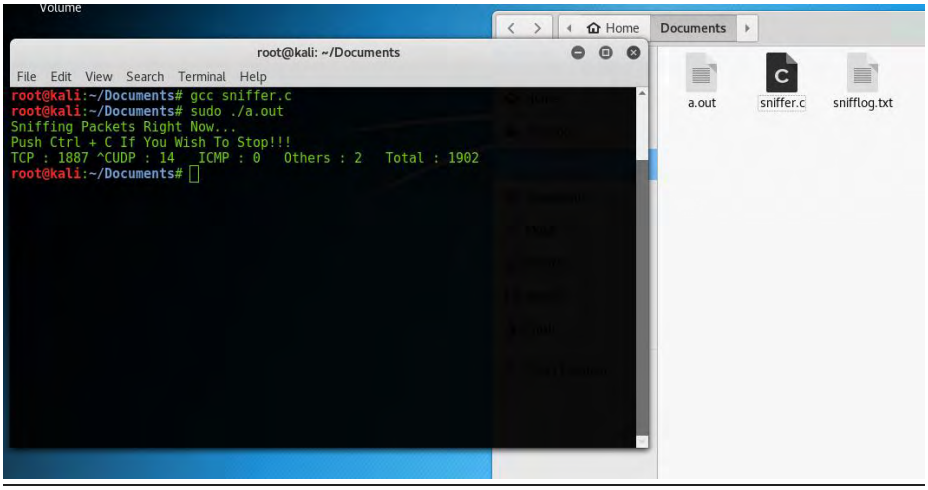
- Μεταγλώττιση & Προετοιμασία Προγράμματος



- Εκκίνηση Προγράμματος και Μετάβαση στο link www.inf.uth.gr από Browser



- Παρατήρηση δημιουργίας του Snifflog.txt



- Ένα τυχαίο πακέτο UDP

```
**********************UDP Packet*************************

Ethernet Header
   |-Destination Address : B0-6E-BF-DB-B0-F8
   |-Source Address      : 20-16-D8-8A-B5-75
   |-Protocol            : 8

IP Header
   |-IP Version        : 4
   |-IP Header Length  : 5 DWORDS or 20 Bytes
   |-Type Of Service   : 0
   |-IP Total Length   : 60  Bytes(Size of Packet)
   |-Identification    : 55376
   |-TTL      : 64
   |-Protocol : 17
   |-Checksum : 57093
   |-Source IP        : 192.168.1.9
   |-Destination IP   : 192.168.1.1

UDP Header
   |-Source Port      : 45598
   |-Destination Port : 53
   |-UDP Length       : 40
   |-UDP Checksum     : 61212

IP Header
    B0 6E BF DB B0 F8 20 16 D8 8A B5 75 08 00 45 00         .n.... ....u..E.
    00 3C D8 50                                             .<.P
UDP Header
    40 00 40 11 DF 05 C0 A8                                 @.@.....
Data Payload
    00 B2 01 00 00 01 00 00 00 00 00 00 03 77 77 77         .............www
    03 69 6E 66 03 75 74 68 02 67 72 00 00 1C 00 01         .inf.uth.gr.....
```

Παρατηρούμε ότι μετά την επεξεργασία του πακέτου και την αποκωδικοποίηση φαίνεται καθαρά το όνομα του συνδέσμου στο οποίο μεταβήκαμε. Επίσης, βασικά στοιχεία για την κεφαλίδα Ethernet όπως τα mac addresses και τον τύπο πρωτοκόλλου στην συνέχεια στοιχεία του Ip heard και τέλος στοιχεία του συγκεκριμένου πακέτου ως UDP.

- Ένα τυχαίο πακέτο TCP

```
**********************TCP Packet*************************

Ethernet Header
   |-Destination Address : B0-6E-BF-DB-B0-F8
   |-Source Address      : 20-16-D8-8A-B5-75
   |-Protocol            : 8

IP Header
   |-IP Version        : 4
   |-IP Header Length  : 5 DWORDS or 20 Bytes
   |-Type Of Service   : 0
   |-IP Total Length   : 376  Bytes(Size of Packet)
   |-Identification    : 42877
   |-TTL      : 64
   |-Protocol : 6
   |-Checksum : 16675
   |-Source IP        : 192.168.1.9
   |-Destination IP   : 194.177.204.124

TCP Header
   |-Source Port      : 51076
   |-Destination Port : 80
   |-Sequence Number    : 2786715212
   |-Acknowledge Number : 180959137
   |-Header Length      : 8 DWORDS or 32 BYTES
   |-Urgent Flag          : 0
   |-Acknowledgement Flag : 1
   |-Push Flag            : 1
   |-Reset Flag           : 0
   |-Synchronise Flag     : 0
   |-Finish Flag          : 0
   |-Window         : 229
   |-Checksum       : 45942
   |-Urgent Pointer : 0
```

Στο επόμενο screenshot το οποίο αποτελεί και συνέχεια της προηγούμενης φωτογραφίας παρουσιάζεται το datagram του πακέτου στο οποίο μπορούμε να αναγνωρίσουμε συγκεκριμένες πληροφορίες.

```
                        DATA Dump
IP Header
    B0 6E BF DB B0 F8 20 16 D8 8A B5 75 08 00 45 00         .n.... ....u..E.
    01 78 A7 7D                                             .x.}
TCP Header
    40 00 40 06 41 23 C0 A8 01 09 C2 B1 CC 7C C7 84         @.@.A#.......|..
    00 50 A6 19 E6 4C 0A C9 37 A1 80 18 00 E5 B3 76         .P...L..7.....v
Data Payload
    47 45 54 20 2F 20 48 54 54 50 2F 31 2E 31 0D 0A         GET / HTTP/1.1..
    48 6F 73 74 3A 20 77 77 77 2E 69 6E 66 2E 75 74         Host: www.inf.ut
    68 2E 67 72 0D 0A 55 73 65 72 2D 41 67 65 6E 74         h.gr..User-Agent
    3A 20 4D 6F 7A 69 6C 6C 61 2F 35 2E 30 20 28 58         : Mozilla/5.0 (X
    31 31 3B 20 4C 69 6E 75 78 20 78 38 36 5F 36 34         11; Linux x86_64
    3B 20 72 76 3A 34 35 2E 30 29 20 47 65 63 6B 6F         ; rv:45.0) Gecko
    2F 32 30 31 30 30 31 30 31 20 46 69 72 65 66 6F         /20100101 Firefo
    78 2F 34 35 2E 30 0D 0A 41 63 63 65 70 74 3A 20         x/45.0..Accept:
    74 65 78 74 2F 68 74 6D 6C 2C 61 70 70 6C 69 63         text/html,applic
    61 74 69 6F 6E 2F 78 68 74 6D 6C 2B 78 6D 6C 2C         ation/xhtml+xml,
    61 70 70 6C 69 63 61 74 69 6F 6E 2F 78 6D 6C 3B         application/xml;
    71 3D 30 2E 39 2C 2A 2F 2A 3B 71 3D 30 2E 38 0D         q=0.9,*/*;q=0.8.
    0A 41 63 63 65 70 74 2D 4C 61 6E 67 75 61 67 65         .Accept-Language
    3A 20 65 6E 2D 55 53 2C 65 6E 3B 71 3D 30 2E 35         : en-US,en;q=0.5
    0D 0A 41 63 63 65 70 74 2D 45 6E 63 6F 64 69 6E         ..Accept-Encodin
    67 3A 20 67 7A 69 70 2C 20 64 65 66 6C 61 74 65         g: gzip, deflate
    0D 0A 43 6F 6F 6B 69 65 3A 20 5F 67 61 3D 47 41         ..Cookie: _ga=GA
    31 2E 32 2E 36 37 33 31 33 32 30 34 37 2E 31 35         1.2.673132047.15
    30 30 30 35 30 38 31 30 0D 0A 43 6F 6E 6E 65 63         00050810..Connec
    74 69 6F 6E 3A 20 6B 65 65 70 2D 61 6C 69 76 65         tion: keep-alive
    0D 0A 0D 0A                                             ....
```

ΕΠΙΛΟΓΟΣ

Μέσα από όλα αυτά τα στοιχεία και τις πληροφορίες που παραθέσαμε για τον τρόπο λειτουργίας του ίντερνετ και τα πρωτόκολλα επικοινωνίας, για επιθέσεις κακόβουλων χρηστών και πληροφορίες για τη βασική ιδέα της εργασίας μας, δηλαδή το sniffer, πιστεύουμε να έγιναν κατανοητές από τον αναγνώστη ο τρόπος λειτουργίας του ίντερνετ και των πακέτων καθώς και ο τρόπος λήψης, επεξεργασίας και διαχείρισης τους. Μπορεί το πρόγραμμά μας να είναι σχετικά απλό σε επίπεδο λειτουργιών αλλά είναι αρκετό για να κατανοήσουμε τον κίνδυνο υποκλοπής στοιχείων και επεξεργασίας τους από έναν καλά διαβασμένο χρήστη έχωντας στην κατοχή του ένα sniffer σε είτε μορφή hardware είτε software.

Μελλοντικά θα μπορούσε το πρόγραμμά μας να τροποποιηθεί ώστε να μπορεί να καταγράφει όσο το δυνατόν περισσότερους τύπους πακέτων όπως επίσης να υλοποιηθεί κάποια συνάρτηση για καλύτερη διαχείριση των datagrams έτσι ώστε να μπορούμε να λαμβάνουμε σε ανθρώπινη κατανοητή μορφή περισσότερες πληροφορίες ενός πακέτου και πιθανής αυτόματης εύρεσης χρήσιμων στοιχείων στο πακέτο. Εάν ο σεβασμός προς τον συνάνθρωπο και η ορθολογική χρήση του ίντερνετ ήταν τα μοναδικά χαρακτηριστικά των χρηστών ηλεκτρονικών υπολογιστών, το μόνο σίγουρο είναι πως ο κάθε χρήστης του διαδικτύου δεν θα είχε καμία ανησυχία χρησιμοποιώντας το και δεν θα άρμοζε ανάγκη για δημιουργία συστημάτων ασφαλείας τόσο σε προσωπικό όσο και σε μαζικό επίπεδο. Δυστυχώς για την ανθρωπότητα η προηγούμενη υπόθεση δεν θα είναι ποτέ εφικτή και οι χρήστες του διαδικτύου είναι αναγκασμένοι να χρήζουν προστασίας από λογισμικά καθώς και να δείχνουν ιδιαίτερη προσοχή στους ιστότοπους που επισκέπτονται καθώς και σε οτιδήποτε εφαρμογή απαιτεί τη χρήση του διαδικτύου για να λειτουργήσει. Ας ελπίσουμε στην ελαχιστοποίηση των κακόβουλων χρηστών του διαδικτύου μελλοντικά έτσι ώστε να μπορεί ολόκληρη η υφήλιος να χρησιμοποιεί χωρίς φόβο ένα από τα μεγαλύτερα επιτεύγματα της ανθρωπότητας.

Τέλος, κλείνουμε με μια φράση του Bill Gates αναφερόμενος τόσο στην σημαντικότητα του διαδικτύου και της ιδιότητας του να ενώνει τους ανθρώπους μεταξύ τους όσο και στη μακροχρόνια συνέχιση ύπαρξης του στο μέλλον.

"The Internet Is Becoming The Town Square For The Global Village Of Tomorrow" **Bill Gates**

Βιβλιογραφία

[1] https://en.wikipedia.org/wiki/Internet
[2] Feibel, W. (1996). *The encyclopedia of networking*. San Francisco, CA: Network Press., p.760
[3] Stallings, W. (2005). *Cryptography and network security principles and practice* (4th ed.). Boston: Prentice Hall., p.622
[4] Meehan, M. (2000, December 11). Is Carnivore Dangerous? Controversy Continues. *Computerworld*, p. 24
[5] Shinder, T. W., & Carasik, A. H. (2003). *Best damn firewall book period*. Rockland: Syngress., p.1189
[6] Sabeel Ansari, Rajeev S.G. And Chandrashekar H.S., *Packet Sniffing: A Bried Introduction*, DECEMBER 2002/JANUARY 2003
[7] https://el.wikipedia.org/wiki/Token_ring
[8] Spurgeon, C. E., & Zimmerman, J. (2014). *Ethernet: the definitive guide*. Beijing: Oreilly., p.24
[9] Tipton, H. F., & Nozaki, M. K. (2014). *Information security management handbook*. Boca Raton: CRC Press., p.274-26,355,668-669
[10] Sanders, C. (2017). *Practical packet analysis: using Wireshark to solve real-world network problems*. San Francisco: No Starch Press., p.218
[11] Tozer, H. F. (2013). *Broadcast engineers reference book*. New York: Focal Press., p.67
[12] Kurose, J. F., & Ross, K. W. (2017). *Computer networking: a top-down approach* (6th ed.). Boston: Pearson., p. 231-235, 333-335, 353-355, 357-359
[13] Tanenbaum, A. S. (2010). *Computer networks* (5th ed.). New Delhi, India: Prentice-Hall of India Private Limited., p.541-543
[14] Peterson, L. L., & Davie, B. S. (2003). *Computer networks: a system approach* (5th ed.). Amsterdam: Morgan Kaufmann. p.235
[15] Gibson, D. (2015). *Managing Risk in information Systems* (1st ed.). Burlington, MA: Jones & Bartlett Learning, p. 216
[16] Nagaraja, M. G., Chittal, R. R., & Kumar, K. (2007). Study of Network Performance Monitoring Tools-SNMP. *IJCSNS International Journal of Computer Science and Network Security,7*, 7th ser.
[17] Petr, E. (2009). *An analysis of the DNS cache poisoning attack*, p.7-9