ΕΝΟΤΗΤΑ ΙΙ

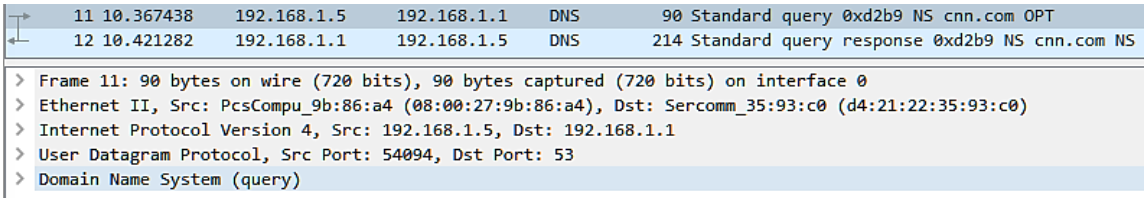
3. ΚΑΤΑΝΟΗΣΗ PROTOCOL DATA UNITS ΜΕΣΑ ΑΠΟ ΤΟ ΠΡΩΤΟΚΟΛΛΟ DNS

Παρακάτω θα αναλύσουμε ερωτήματα και αποκρίσεις του πρωτοκόλλου DNS το οποίο ανήκει στο 5ο επίπεδο (εφαρμογής - application). Θα κατανοήσουμε πως τα **δεδομένα (data)** που αποτελούν το PDU στο L5, μεταφέρονται σε **τμήματα (segments)** στο L4 (μεταφοράς - transport), τα οποία εγκιβωτίζονται σε **πακέτα IPv4** (IPv4 packet encapsulation) στο L3 (δικτύου - network), τα οποία με την σειρά τους ενθυλακώνονται μέσα σε **πλαίσια (frames)** που είναι το PDU στο L2 (ζεύξης δεδομένων–data link). Για την καλύτερη κατανόηση των PDUs θα σας βοηθήσει να έχετε διαθέσιμη μπροστά σας την εικόνα 1.10 από το κεφάλαιο 1, όπου υπάρχουν τα επίπεδα και τα αντίστοιχα PDU του TCP/IP Protocol Stack.

3.1 Ανάλυση καταγεγραμμένης επικοινωνίας DNS

3.1.1 Εργασία με αποθηκευμένο αρχείο καταγραφής

Κατεβάστε το αρχείο **dns_capture.pcapng** από τον ιστότοπο του μαθήματος, ξεκινήστε το Wireshark και αντί για καταγραφή ανοίξτε το αρχείο που μόλις κατεβάσατε από το μενού File ->Open. Περιορίστε την εμφάνιση στις γραμμές που υπάρχει στην στήλη protocol το DNS, χρησιμοποιώντας το κατάλληλο φίλτρο. Για να μελετήσετε τις δυνατότητες των φίλτρων στο Wireshark μπορείτε να ξεκινήσετε από την σελίδα https://wiki.wireshark.org/DisplayFilters Εκεί υπάρχουν παραδείγματα, μέσα στα οποία υπάρχει και το DNS.



Εικόνα 3.1: Καταγεγραμμένο ζεύγος ερωτήματος και απόκρισης DNS

Στο πρώτο (επάνω) panel πρέπει να εμφανίζονται μόνο οι γραμμές που σας ενδιαφέρουν και μέσα σε αυτές επιλέξτε μια που το περιεχόμενο στην στήλη Info ξεκινάει από “Standard query…”. Κατόπιν εστιάστε στο δεύτερο (κάτω αριστερά) panel του Wireshark. Στην δεύτερη γραμμή εμφανίζεται το πλαίσιο (frame) του πρωτοκόλλου 2ου επιπέδου (Data Link) στο οποίο ενθυλακώνεται ένα πακέτο (packet) του πρωτοκόλλου του 3ου επιπέδου (Network). Το πακέτο εμφανίζεται στην αμέσως επόμενη γραμμή (τρίτη), και με την σειρά του ενθυλακώνει ένα τμήμα (segment) του πρωτοκόλλου του 4ου επιπέδου (Transport) το οποίο εμφανίζεται στην τέταρτη γραμμή. Στο segment ενθυλακώνεται ένα μήνυμα του πρωτοκόλλου του 5ου επιπέδου (Application), που στην περίπτωση μας είναι DNS και εμφανίζεται στην πέμπτη γραμμή.

Άσκηση 3.1: Εφαρμογή φίλτρων

Εφαρμόστε φίλτρα πάνω στην αποθηκευμένη καταγραφή ώστε:

1. Να εμφανιστούν μόνο οι γραμμές στο πάνω πάνελ που αφορούν το πρωτόκολλο DNS
2. Το φίλτρο που εφαρμόσατε στο Wireshark με το περιεχόμενο ποιας στήλης ταιριάζει;
3. Να φιλτράρετε περαιτέρω τις γραμμές ώστε να εμφανίζονται αυτές που αφορούν το DNS αλλά έχουν και διεύθυνση source 192.168.1.5

Άσκηση 3.2: Ανάλυση ερωτήματος και αντίστοιχης απόκρισης

Εργαστείτε πάνω στην καταγραφή ώστε να είστε σε θέση να απαντήσετε τα παρακάτω ερωτήματα. Μπορείτε να αναπτύξετε την κάθε γραμμή στο δεύτερο (κάτω αριστερά) panel και να δείτε δομημένη πληροφορία για ένα PDU που είναι γνωστό στο Wireshark.

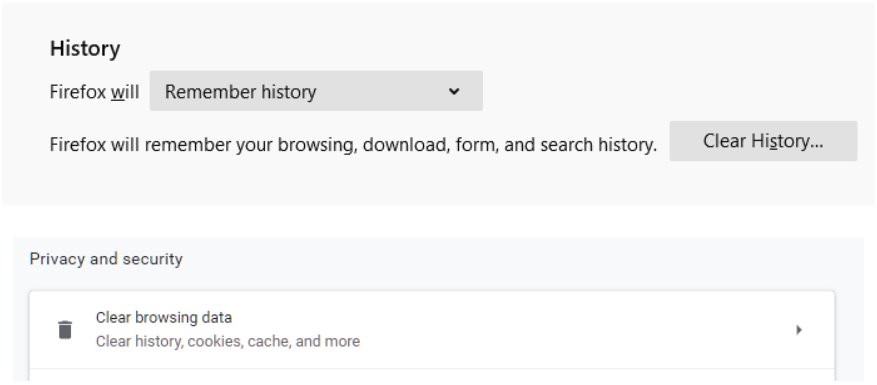
1. Το μήνυμα που στέλνει ένας DNS client (πελάτης) σε έναν DNS server (εξυπηρετητή) ονομάζεται DNS query (ερώτημα) και στο Wireshark υπάρχει στο Info ως **"Standard query {ένας δεκαεξαδικός αριθμός} …".** Εμφανίστε και καταγράψτε τα πεδία του δεύτερου DNS query.
2. Στην συγκεκριμένη γραμμή καταγραφής και για κάθε επίπεδο του μοντέλου πρωτοκόλλων του TCP/IP, καταγράψτε τα πρωτόκολλα που πήραν μέρος στην επικοινωνία με τον DNS server.
3. Καταγράψτε την τοπική (local) και την απομακρυσμένη (remote) IP διεύθυνση της επικοινωνίας, υποθέτοντας ότι το query αποστέλλεται από τον υπολογιστή μας.
4. Καταγράψτε τα port προέλευσης (source) και προορισμού (destination) που χρησιμοποιήθηκαν για το DNS query που εστάλη στον DNS server.
5. Με βάση τις απαντήσεις των δύο παραπάνω (3 και 4) σημειώστε το socket από το οποίο απέστειλε τα segments η τοπική διεργασία στον υπολογιστή μας και το socket που τα παρέλαβε η απομακρυσμένη διεργασία του DNS server.
6. Ποιο είναι το πρωτόκολλο με το οποίο μεταφέρονται τα δεδομένα του πρωτόκολλου επιπέδου εφαρμογής DNS; Δηλαδή ποιο είναι το πρωτόκολλο επιπέδου μεταφοράς;
7. Σε ποιο προκαθορισμένο αριθμό port αναμένει queries ένας DNS server;
8. Εντοπίστε το κείμενο της ερώτησης που γίνεται προς τον DNS server, μέσω του συγκεκριμένου query που αναλύουμε. Σε ποιο πεδίο το βρίσκετε;
9. Το μήνυμα που επιστρέφει ένας DNS server σε έναν DNS client ονομάζεται DNS response (απόκριση) και στο Wireshark υπάρχει στο info ως **"Standard query response {ένας δεκαεξαδικός αριθμός} …"**. Εντοπίστε την γραμμή στο πρώτο πάνελ για το αντίστοιχο response στο query που έχουμε ήδη αναλύσει.
10. Καταγράψτε τα socket προέλευσης (source) και προορισμού (destination) που χρησιμοποιήθηκαν για την μεταφορά της απόκρισης από τον DNS server προς τον DNS client.
11. Εμφανίστε και καταγράψτε τα πεδία του DNS response. Μέσα σε αυτά μπορείτε να βρείτε ποιοι είναι οι δηλωμένοι name servers οι οποίοι περιέχουν τις εγγραφές για το domain που υπάρχει στο request;

12. Μπορείτε να βρείτε που βρίσκεται και ποια είναι η τιμή στο δεκαεξαδικό άθροισμα ελέγχου (checksum) για την ερώτηση και για την απάντηση;
13. Για ποιο λόγο χρειαζόμαστε ένα checksum; Η απάντηση συνδέεται με το πρωτόκολλο και το επίπεδο στο οποίο θα το βρείτε.

3.1.2 Καταγραφή επικοινωνίας DNS που συμβαίνει στο παρασκήνιο και η κρυφή μνήμη DNS

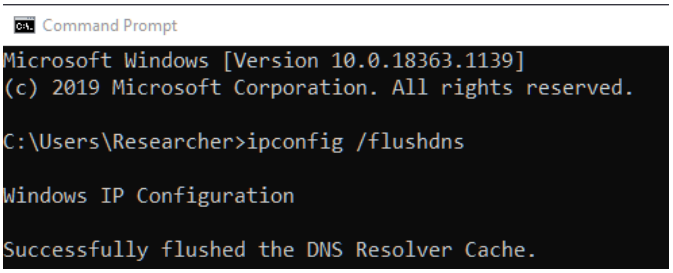
Μπορούμε να αναλύσουμε ερωτήματα DNS που βρίσκονται στο παρασκήνιο από τις διαδικτυακές εφαρμογές που χρησιμοποιούμε. Παρακάτω θα δούμε το αρχικό στάδιο για το κατέβασμα μια ιστοσελίδας κατά το οποίο πρέπει να βρεθεί η διεύθυνση IPv4. Σε επόμενα βήματα σε συνδυασμό με την αντίστοιχη θύρα, το απομακρυσμένο socket χρησιμοποιείται για να μεταφέρει τα δεδομένα του επιπέδου εφαρμογής όπως HTML, CSS, εικόνες, κ.α. από τον εξυπηρετητή Web.

Κλείστε εντελώς όλα τα παράθυρα του browser που χρησιμοποιείτε. Κατόπιν εκκινήστε τον browser και ανοίξτε μια νέα καρτέλα ιδιωτικής περιήγησης (Firefox: Private Window, Chrome: Incognito Window). Αυτό γίνεται ώστε να μπορείτε να επαναλάβετε την διαδικασία, καθώς οι browsers θα επιστρέψουν μια σελίδα που έχετε επισκεφτεί από την τοπική κρυφή μνήμη (browser cache) που συντηρείται στον δίσκο σας. Εναλλακτικά αν δεν καταγράψετε δεδομένα, πρέπει να εκκαθαρίσετε την browser cache με τον αντίστοιχο χειρισμό.



Εικόνα 3.2: Εκκαθάριση ιστορικού από τις ρυθμίσεις Firefox (επάνω) και Chrome (κάτω)

Εκτός αυτού τα λειτουργικά συστήματα έχουν και κρυφή μνήμη για αιτήματα και αποκρίσεις DNS, την DNS cache. Πρέπει να την καθαρίσετε για να εκτελεστεί ξανά το ίδιο ερώτημα DNS. Στα Windows εκτελούμε από την γραμμή εντολών `ipconfig /flushdns` και στο Linux την `sudo system-resolve -flush-caches` για εκκαθάριση και κατόπιν για επιβεβαίωση `sudo systemd-resolve -statistics`.

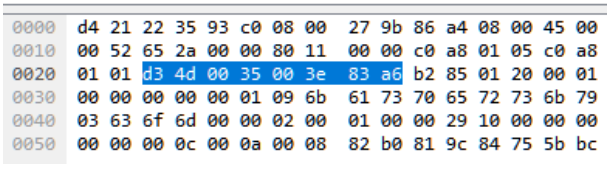


Εικόνα 3.3: Εκκαθάριση της dns cache σε Windows

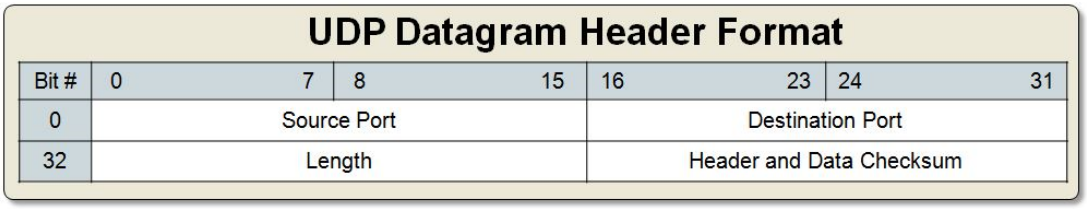
Άσκηση 3.3: Διαδικασία DNS resolution (αποσαφήνισης) κατά την λειτουργία του web browser: (1/2) -DNS Query

Ξεκινήστε μια νέα καταγραφή με το Wireshark. Γράψτε στη γραμμή URL **www.google.com** και αφού εμφανιστεί η σελίδα σταματήστε την καταγραφή του Wireshark. Εφαρμόστε το κατάλληλο φίλτρο για να βλέπετε μόνο τις γραμμές DNS. Αν δεν εμφανιστούν μετά την εφαρμογή του φίλτρου, είναι επειδή βρέθηκε σε κάποιες από τις cache. Αναλύστε την καταγραφή:

1. Καταγράψτε την IPv4 διεύθυνση και τη φυσική διεύθυνση (MAC address) του υπολογιστή σας από το πρώτο DNS query που υπάρχει.
2. Καταγράψτε την IPv4 διεύθυνση και την φυσική διεύθυνση (MAC address) του DNS server που διεκπεραίωσε το αίτημα σας.
3. Τρέξτε την εντολή `ipconfig /all` και βρείτε την διεύθυνση IPv4 (από την απάντηση στο ερώτημα 2) μέσα στις ρυθμίσεις ενός από τα NIC του συστήματος σας. Παρατηρείτε την ίδια διεύθυνση και σε κάποιο άλλο πεδίο; Δοκιμάστε και στο σπίτι σας αν ισχύει. Αν ναι, με ποια διεύθυνση στο δίκτυο σας;
4. Αναπτύξτε την γραμμή που εμφανίζει το πρωτόκολλο του επιπέδου μεταφοράς (transport). Για χρήση με το DNS είναι <u>πάντα</u> το UDP. Έχει μικρή κεφαλίδα όπως φαίνεται στις εικόνες 3.4 και 3.5



Εικόνα 3.4: Η κεφαλίδα του UDP σε raw δεκαεξαδική μορφή, υπερφωτίζεται στο τρίτο panel (κάτω δεξιά) του Wireshark εφόσον κάνουμε κλικ στην αντίστοιχη γραμμή για το πρωτόκολλο μεταφοράς



Εικόνα 3.5: Η διαμόρφωση των 64 bits της κεφαλίδας UDP με τα πεδία από-ως bit, τα οποία εμφανίζει το Wireshark προς τον χρήστη σε δομημένη μορφή

5. Συμπληρώστε τα παρακάτω πεδία που εμφανίζονται στο Wireshark, εντοπίζοντας τα παράλληλα και στα raw bytes του τρίτου panel (κάτω δεξιά):

Source IP Address		Source Port	
Destination IP Address		Destination Port	
Source MAC Address			
Destination MAC Address			
Frame Size:			

Άσκηση 3.4: Διαδικασία DNS resolution (αποσαφήνισης) κατά την λειτουργία του web browser: (2/2) DNS response

Εντοπίστε το κατάλληλο DNS response για το query που έγινε κατά την εκτέλεση της προηγούμενης άσκησης 3.3 και αφορούσε το **www.google.com**. Παρατηρείστε ότι η απάντηση είναι πάντα μεγαλύτερη (σε bytes) από το ερώτημα που θέσατε.

1. Ποιες είναι τώρα οι φυσικές διευθύνσεις του αποστολέα και του παραλήπτη;
2. Σε ποιες συσκευές αντιστοιχούν (αντιπαραβάλετε τις πληροφορίες σε σχέση με το DNS query στον πίνακα της ερώτησης 5 της προηγούμενης άσκησης).
3. Τι παρατηρείτε για τις διευθύνσεις IPv4 μεταξύ αποστολέα και παραλήπτη; Ισχύει το ίδιο και για τις ports που χρησιμοποιήθηκαν ;
4. Ποια πιστεύετε πως είναι η χρησιμότητα του UDP ως πρωτόκολλο μεταφοράς για το DNS σε σχέση με το TCP που ονομάζει τη σουίτα των πρωτοκόλλων του TCP/IP; Αφού προσπαθήσετε να δώσετε την απάντηση, κοιτάξτε το υπόμνημα από το βιβλίο της θεωρίας στην παράγραφο 3.5.1.

3.2 Ανάλυση επικοινωνίας DHCP με Wireshark

3.2.1 Το Πρωτόκολλο Δυναμικής Διαμόρφωσης Υπολογιστών

Ένας διαχειριστής συστήματος διαμορφώνει συνήθως τις διευθύνσεις IP μέσα στον δρομολογητή (συχνά απομακρυσμένα). Οι διευθύνσεις υπολογιστών μπορούν επίσης να αποδοθούν με το χέρι, αλλά συνήθως αυτό γίνεται με τη χρήση του Πρωτοκόλλου Δυναμικής Διαμόρφωσης Υπολογιστών (**Dynamic Host Configuration Protocol, DHCP**) [RFC 2131].
Οι περισσότεροι χρήστες δεν αντιλαμβάνονται τις τεχνικές λεπτομέρειες της δικτύωσης και οι ρυθμίσεις του TCP/IP για να συνδεθούν σε ένα δίκτυο, τους φαίνονται πολύπλοκες. Το DHCP δίνει τη δυνατότητα σ' αυτούς τους χρήστες να συνδεθούν εύκολα στο δίκτυο και στο διαχειριστή το πλεονέκτημα της κεντρικής διαχείρισης των ρυθμίσεων και την ευκολία υποστήριξης των χρηστών και συντήρησης του δικτύου.

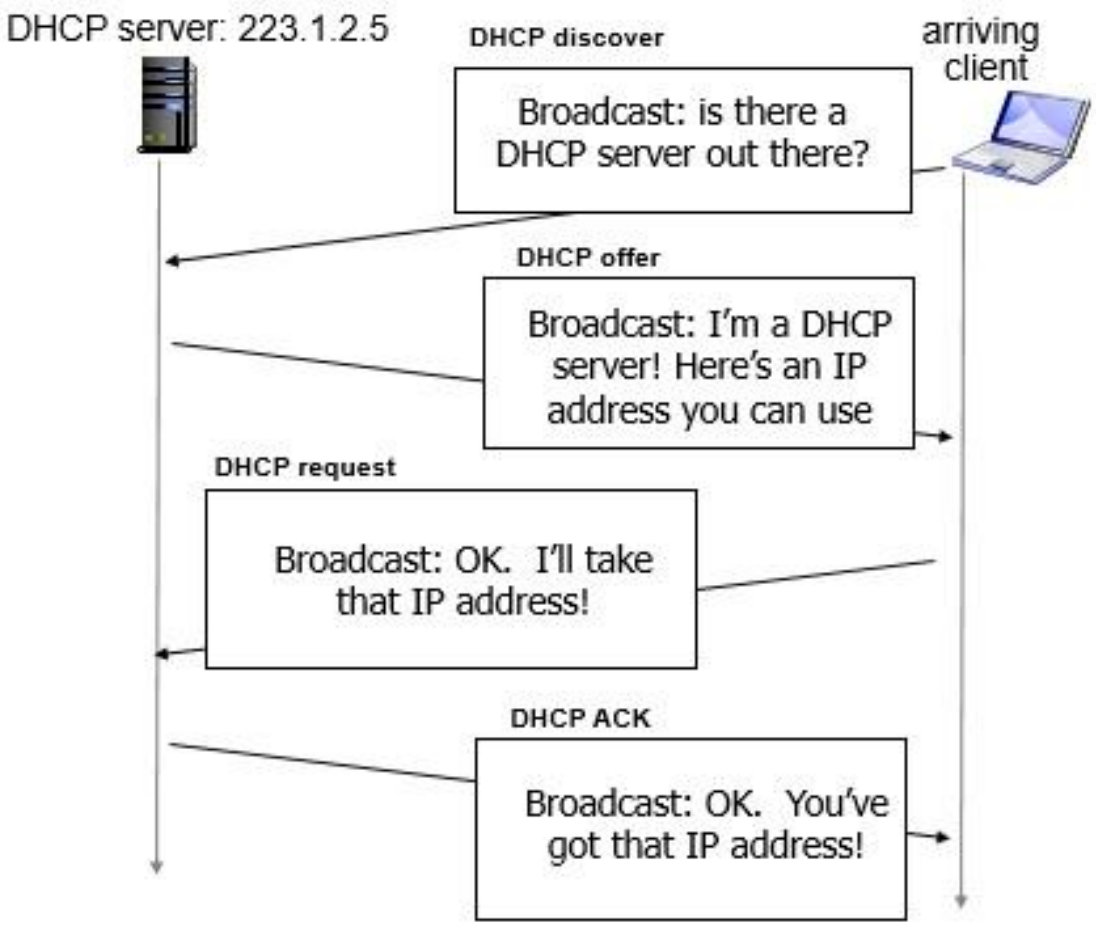
Το DHCP επιτρέπει σ' έναν υπολογιστή να αποκτήσει (να του εκχωρηθεί) μια διεύθυνση IP αυτόματα. Δίνει επίσης τη δυνατότητα να μάθει πρόσθετες πληροφορίες, όπως η μάσκα υποδικτύου, η διεύθυνση του δικού του δρομολογητή πρώτου-άλματος (συχνά αποκαλούμενη προεπιλεγμένη πύλη) και η διεύθυνση του τοπικού του εξυπηρετητή DNS. Επιτρέπει σε έναν υπολογιστή να αποκτά τις ρυθμίσεις που χρειάζεται σε ένα μόνο μήνυμα.
Καθορίζει τρεις τύπους εκχώρησης διευθύνσεων:

- μη αυτόματη ρύθμιση (manual configuration), στην οποία ο διαχειριστής ορίζει συγκεκριμένες διευθύνσεις που θα πάρουν συγκεκριμένοι υπολογιστές.
- αυτόματη ρύθμιση (automatic configuration), κατά την οποία ο διακομιστής DHCP εκχωρεί μια μόνιμη διεύθυνση σε έναν υπολογιστή ο οποίος συνδέεται πρώτη φορά.
- δυναμική ρύθμιση (dynamic configuration) κατά την οποία ο διακομιστής δανείζει ή μισθώνει μια διεύθυνση σε έναν υπολογιστή για περιορισμένο χρόνο.

Η δυναμική ρύθμιση είναι και η πιο συχνά χρησιμοποιούμενη.

Ένας υπολογιστής, ρυθμισμένος να χρησιμοποιεί την υπηρεσία DHCP, αμέσως μετά την εκκίνησή του:

- Δημιουργεί ένα πακέτο UDP **DHCP DISCOVER** με θύρα προορισμού 67. Το ενθυλακώνει σε πακέτο IPv4 με διεύθυνση προέλευσης 0.0.0.0 και διεύθυνση προορισμού τη διεύθυνση εκπομπής 255.255.255.255. Στη συνέχεια το ενθυλακώνει σε ένα πλαίσιο με διεύθυνση προέλευσης τη δική του φυσική διεύθυνση και διεύθυνση προορισμού τη διεύθυνση εκπομπής FF-FF-FF-FF-FF-FF και στέλνεται στο τοπικό δίκτυο.
- Εάν υπάρχουν εξυπηρετητές DHCP ανταποκρίνονται ο καθένας με ένα πακέτο **DHCP OFFER** στη θύρα 68, ενθυλακωμένο σε πακέτο IP εκπομπής και πλαίσιο εκπομπής (διευθύνσεις προορισμού 255.255.255.255 και FF-FF-FF-FF-FF-FF αντίστοιχα). Όταν είναι εφικτό, αποφεύγουν να απαντούν με πλαίσια εκπομπής.
- Ο πελάτης υπολογιστής επιλέγει τις ρυθμίσεις που προσφέρονται από έναν από τους εξυπηρετητές και το δηλώνει αποστέλλοντας ένα πακέτο εκπομπής **DHCP REQUEST** στο οποίο ζητά τις προσφερόμενες ρυθμίσεις.
- Ο εξυπηρετητής DHCP που προσέφερε τις ρυθμίσεις επιβεβαιώνει την προσφορά του με ένα πακέτο **DHCP ACK**.



Εικόνα 3.6: Αλληλεπίδραση πελάτη εξυπηρετητή DHCP

Από τη λήψη της επιβεβαίωσης DHCP ACK και στη συνέχεια ο υπολογιστής λειτουργεί με τις δικτυακές ρυθμίσεις που πήρε (κατάσταση Δεσμευμένος - BOUND). Η διεύθυνση IPv4 παραχωρείται στον υπολογιστή για συγκεκριμένο χρονικό διάστημα και χαρακτηρίζεται ως μίσθωση (lease).

Από τη στιγμή αυτή, ο υπολογιστής αρχίζει τη σχετική μέτρηση χρόνου ώστε να προβεί στις κατάλληλες ενέργειες παράτασης της μίσθωσης της διεύθυνσης πριν τη λήξη της. Αν θέλει να κάνει ανανέωση του χρόνου στον ίδιο server εκτελεί πάλι ένα **DHCP REQUEST** απευθυνόμενος μόνο σε αυτόν με (unicast αποστολή) και λαμβάνοντας ένα πακέτο **DHCP ACK** γίνεται η ανανέωση (RENEW) της μίσθωσης.

Όταν ο υπολογιστής τερματίζει τη λειτουργία του ομαλά πριν λήξει η μίσθωση της διεύθυνσης, τότε απελευθερώνει την διεύθυνσή του στέλνοντας πριν τον τερματισμό, στον διακομιστή DHCP, ένα πακέτο **DHCP RELEASE**.

Το πρωτόκολλο DHCP προτάθηκε ως επέκταση του BOOTP, αρχικά στα RFC1531, 1541 τα οποία έχουν αντικατασταθεί από το RFC2131. Όλες οι πληροφορίες σχετικά με αυτό βρίσκονται στο RFC2131 και στα συμπληρωματικά του.

3.2.2 Εργασία με αρχείο καταγραφής DHCP

ΑΣΚΗΣΗ 3.5: Ανάλυση καταγραφής DHCP από αποθηκευμένο αρχείο

Κατεβάστε το αρχείο **dhcp.pcap** από τον ιστότοπο του μαθήματος, ξεκινήστε το Wireshark και αντί για καταγραφή ανοίξτε το αρχείο που μόλις κατεβάσατε από το μενού File ->Open. Οι γραμμές της λίστας περιέχουν την επικοινωνία μεταξύ ενός DHCP client και DHCP server όπως και της εικόνας 3.6

1. Τα μηνύματα DHCP στέλνονται χρησιμοποιώντας ως πρωτόκολλο μεταφοράς το UDP ή το TCP;
2. Για κάθε πλαίσιο εντοπίστε τις πόρτες πηγής και προορισμού. Είναι οι πόρτες αυτές ίδιες με αυτές που αναφέρθηκαν πιο πάνω (3.2.1 σελ 5-6)? Ποιες είναι για το REQUEST και το ACK;
3. Ποιες τιμές μέσα στο μήνυμα DHCP DISCOVER διαφέρουν σε σχέση με το μήνυμα DHCP REQUEST;
4. Ποια είναι η τιμή του Transaction ID; Είναι ίδια σε όλα; Ποιος είναι ο σκοπός αυτού του πεδίου;
5. Για κάθε μήνυμα εντοπίστε και καταγράψτε τις διευθύνσεις IP της πηγής και του προορισμού.

ΑΣΚΗΣΗ 3.6: Καταγραφή μηνυμάτων DHCP

Ξεκινήστε ανοίγοντας ένα Windows Command Prompt όπως έχουμε χρησιμοποιήσει και παλαιότερα. Εκτελέστε την εντολή `ipconfig /release` . Έτσι ο υπολογιστής σας αποκτά την διεύθυνση IP 0.0.0.0 . Ξεκινήστε μια νέα καταγραφή στο Wireshark. Επιστρέψτε στο Windows Command Prompt. Εκτελέστε την εντολή `ipconfig /renew` . Με αυτή την εντολή ο υπολογιστής σας παίρνει νέες ρυθμίσεις και φυσικά διεύθυνση IPv4. Αφού εκτελεστεί πρέπει να την εκτελέσετε `ipconfig /renew` ξανά άλλη μια φορά. Μετά την εκτέλεση δεύτερης φοράς της εντολής, πρέπει να εκτελέσετε την εντολή `ipconfig /release` για να ελευθερωθεί η ρύθμιση που είχατε λάβει. Τελευταία εντολή για εκτέλεση είναι η `ipconfig /renew`. Σταματήστε την καταγραφή και απαντήστε στις ερωτήσεις.

1. Πόσα πλαίσια καταγράφηκαν;
2. Ποια είναι η διεύθυνση IPv4 του δικού σας DHCP server ;
3. Ποια είναι η διεύθυνση IPv4 του DHCP server που δίνει τις ρυθμίσεις στον υπολογιστής σας στο μήνυμα DHCP OFFER; Εντοπίστε ποιο μήνυμα προσφέρει στον υπολογιστή την προσφερόμενη διεύθυνση.
4. Ένας υπολογιστής ζητά τις ρυθμίσεις συμπεριλαμβανομένης και της διεύθυνση IPv4 σε ένα μήνυμα DHCP REQUEST. Συμβαίνει το ίδιο και στην δική σας καταγραφή;
5. Εντοπίστε και εξηγείστε το πεδίο lease time. Ποια είναι η τιμή στη δική σας καταγραφή;
6. Ποιος είναι ο σκοπός του μηνύματος DHCP RELEASE; Ο εξυπηρετητής δίνει μια επαλήθευση παραλαβής του μηνύματος από τον πελάτη στο συγκεκριμένο μήνυμα; Τι θα γίνει αν χαθεί το μήνυμα από τον πελάτη DHCP RELEASE;

3.3 Ανάλυση επικοινωνίας ARP με Wireshark

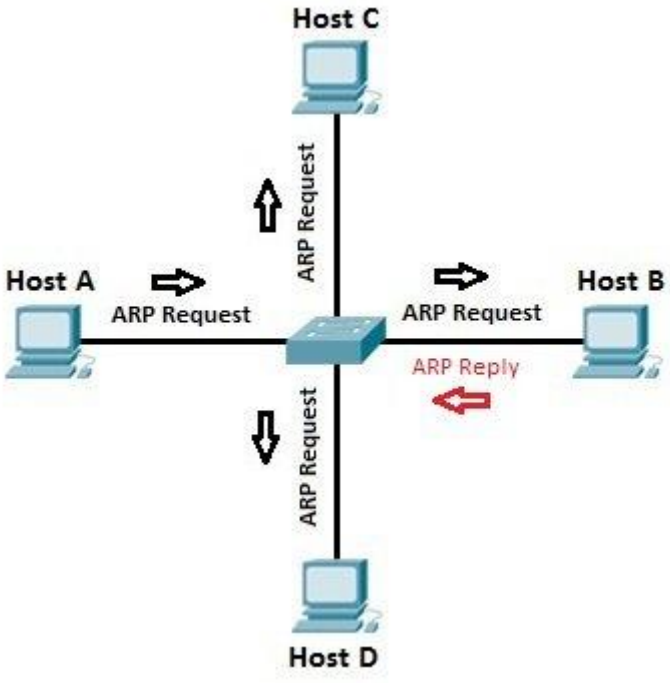
3.3.1 Το Πρωτόκολλο Ανάλυσης Διευθύνσεων

Σε έναν κόμβο ο οποίος επιθυμεί να αποστείλει δεδομένα σε κάποιον άλλο, το επίπεδο εφαρμογής ξεκινά τη διαδικασία ενθυλάκωσης και κάθε επίπεδο είναι υπεύθυνο να προσθέσει τις δικές του διαχειριστικές πληροφορίες στη PDU μονάδα του. Το επίπεδο δικτύου :

- δημιουργεί ένα πακέτο IP ενθυλακώνοντας τα δεδομένα που του παραδόθηκαν από το παραπάνω επίπεδο μεταφοράς L4
- τοποθετεί στα αντίστοιχα πεδία της επικεφαλίδας του τις **διευθύνσεις IPv4** προέλευσης και προορισμού - καθώς και ό,τι άλλο απαιτείται
- το παραδίδει στο αμέσως κατώτερο επίπεδο L2.

Το επίπεδο ζεύξης δεδομένων όμως δε γνωρίζει τίποτα από διευθύνσεις IPv4 παρά μόνο για **διευθύνσεις υλικού** ή **φυσικές** ή **διευθύνσεις MAC**. Για να το παραδώσει στον παραλήπτη θα πρέπει να γνωρίζει σε ποια φυσική διεύθυνση βρίσκεται ο κόμβος με τη διεύθυνση IPv4 που υπάρχει στο αντίστοιχο πεδίο του πακέτου.

Τον συνδετικό κρίκο ανάμεσα στα δυο επίπεδα, απαντώντας στο ερώτημα “ποια είναι η φυσική διεύθυνση (MAC) του κόμβου με τη συγκεκριμένη διεύθυνση IPv4;” αναλαμβάνει το πρωτόκολλο ανάλυσης διευθύνσεων (**Address Resolution Protocol, ARP**).



Εικόνα 3.7: Λειτουργία του ARP

Το **ερώτημα ARP** (ARP request) απευθύνεται στο τοπικό δίκτυο Ethernet με ένα πλαίσιο εκπομπής (broadcast) με διεύθυνση Ethernet προορισμού FF-FF-FF-FF-FF-FF. Αυτό σημαίνει ότι το ερώτημα φτάνει σε όλους τους κόμβους.

Οι κόμβοι οι οποίοι δεν έχουν την διεύθυνση IP η οποία περιλαμβάνεται στο ερώτημα, απλά το αγνοούν. Ο κόμβος ο οποίος αναγνωρίζει την δική του διεύθυνση IPv4 αποστέλλει μια **απάντηση ARP** (ARP Reply) με ένα πλαίσιο με προορισμό την διεύθυνση Ethernet του ερωτούντος απευθυνόμενος μόνο σε αυτόν (unicast).

Τώρα πια είναι γνωστή η φυσική διεύθυνση του παραλήπτη και μπορεί να ολοκληρωθεί το πλαίσιο Ethernet και να αποσταλεί στον παραλήπτη.

Για να αποφευχθούν τα συχνά ερωτήματα προς το τοπικό δίκτυο με πλαίσια εκπομπής (αυξημένη δικτυακή κίνηση), οι σταθμοί διατηρούν προσωρινά τις απαντήσεις που έλαβαν σε έναν πίνακα αντιστοιχίας διευθύνσεων IP σε διευθύνσεις Ethernet στην τοπική μνήμη (**arp cache**). Έτσι πριν υποβάλλουν νέο ερώτημα ελέγχουν τον πίνακά τους arp και υποβάλλουν ερώτημα μόνο όταν δεν υπάρχει κατάλληλη καταχώριση σε αυτόν. Υπάρχει ένας πίνακας ARP για κάθε δικτυακή διασύνδεση (NIC) ενός υπολογιστή.

Παρακάτω φαίνεται ένας **πίνακας arp** (cache) ενός υπολογιστή με Windows 10. Οι δυναμικές καταχωρήσεις προέρχονται από ερωτήματα arp ενώ οι στατικές είναι προκαθορισμένα ρυθμισμένες. Προσέξτε ότι η διεύθυνση IPv4 εκπομπής αντιστοιχεί σε διεύθυνση Ethernet εκπομπής.

```
Interface: 192.168.1.104 --- 0x19
  Internet Address      Physical Address      Type
  192.168.1.31          fc-d5-d9-ee-96-2d     dynamic
  192.168.1.232         84-25-19-0e-2b-9b     dynamic
  192.168.1.254         7c-77-16-26-ff-30     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.2             01-00-5e-00-00-02     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Οι δυναμικές καταχωρίσεις του πίνακα arp μετά την παρέλευση ορισμένου χρόνου χωρίς να χρησιμοποιηθούν, διαγράφονται. Ο χρόνος ποικίλει από μερικά δευτερόλεπτα μέχρι μερικά λεπτά (συνήθως 1-5 λεπτά) και μπορεί να ρυθμιστεί από τον διαχειριστή του συστήματος.

3.3.2 Εργασία με αρχείο καταγραφής ARP

ΑΣΚΗΣΗ 3.7: Ανάλυση καταγραφής ARP από αποθηκευμένο αρχείο

Κατεβάστε το αρχείο **dns_capture.pcapng** από τον ιστότοπο του μαθήματος, ξεκινήστε το Wireshark και αντί για καταγραφή ανοίξτε το αρχείο που μόλις κατεβάσατε από το μενού File->Open. Περιορίστε την εμφάνιση στις γραμμές που υπάρχει στην στήλη protocol το ARP, χρησιμοποιώντας το κατάλληλο φίλτρο. Τα ερωτήματα έχουν την ερώτηση “Who has…” στη στήλη Info. Οι απαντήσεις ξεχωρίζουν γιατί στην ίδια στήλη δεν υπάρχει η προηγούμενη ερώτηση. Εντοπίστε το πρώτο ζευγάρι ερώτησης και απόκρισης.

1. Ποιες είναι οι δεκαεξαδικές τιμές για τις διευθύνσεις προέλευσης και προορισμού στο πλαίσιο Ethernet που περιλαμβάνεται στο ερώτημα;
2. Ποιος είναι ο δεκαεξαδικός κωδικός του πεδίου Type στο ίδιο πλαίσιο;
3. Στο ερώτημα ARP περιλαμβάνεται η διεύθυνση IPv4 του αποστολέα;
4. Στην απόκριση του συγκεκριμένου ερωτήματος που εμφανίζεται η «απάντηση»;
5. Υπάρχουν και άλλα ερωτήματα; Γιατί δεν υπάρχουν αντίστοιχες απαντήσεις;
6. Ποια είναι η τιμή του πεδίου opcode μέσα στα δεδομένα του ARP στο ερώτημα; Ποια είναι στην απόκριση;

ΑΣΚΗΣΗ 3.8: Καταγραφή μηνυμάτων ARP

Η άσκηση αυτή μπορεί να πραγματοποιηθεί στους οικιακούς υπολογιστές σας μόνο και όχι στο εργαστήριο λόγω δικαιωμάτων.

Ξεκινήστε ανοίγοντας ένα Windows Command Prompt με **δικαιώματα διαχειριστή**. Πηγαίνετε στο δίσκο C:\Windows\System32. Εντοπίστε το **cmd.exe**, με δεξί κλικ στο ποντίκι επιλέξτε **Run as Administrator**. Εκτελέστε την εντολή `arp` για να δείτε τις παραμέτρους της συγκεκριμένης εντολής καθώς και παραδείγματα.

1. Εκτελέστε την εντολή `arp -a` και γράψτε τα περιεχόμενα της κρυφής μνήμης **arp cache** του υπολογιστή σας. Ποια είναι η σημασία των τιμών σε κάθε στήλη;
2. Εκτελέστε την εντολή `arp -d *` για να καθαριστεί η κρυφή μνήμη. Εκτελέστε την προηγούμενη εντολή για να διαπιστώσετε ότι άδειασε.

Εκτελέστε την εντολή `ipconfig /flushdns`. Ξεκινήστε μια νέα καταγραφή στο Wireshark. Ανοίξτε ένα ιδιωτικό παράθυρο και πληκτρολογήστε τη διεύθυνση του εξυπηρετητή ιστού του τμήματός μας. Σταματήστε την καταγραφή και περιορίστε τις γραμμές με φίλτρο για το πρωτόκολλο ARP.

3. Εντοπίστε την πρώτη ερώτηση και απάντηση. Ποιος είναι ο δεκαεξαδικός κωδικός του πεδίου Type στο ερώτημα; Είναι διαφορετικό από την προηγούμενη άσκηση (3.7-2);
4. Ποιες είναι οι δεκαεξαδικές τιμές για τις διευθύνσεις προέλευσης και προορισμού στο πλαίσιο Ethernet που περιλαμβάνεται στην απόκριση;

3.4 Χρήσιμες τεχνικές για την αποδοτικότερη χρήση του Wireshark

Capture Filter

Για να περιορίσουμε την καταγραφή μας μπορούμε να εφαρμόσουμε ένα φίλτρο καταγραφής (Capture Filter) είτε κατά την εκκίνηση του Wireshark είτε αφού έχει γίνει μια καταγραφή και πριν πατήσουμε το πλήκτρο ώστε να ξεκινήσει η επόμενη.



Εικόνα 3.8: Πεδίο Capture Filter κατά την εκκίνηση του Wireshark

Στην δεύτερη περίπτωση επιλέγουμε το μενού Capture ->Options ώστε να εμφανιστεί το παράθυρο "Wireshark–Capture Interfaces" και γράφουμε στο πεδίο Capture filter for selected intefaces : **port 53.** Με την συγκεκριμένη επιλογή θα καταγράψει μόνο μεταφορά από την port του DNS. Ανάλογα θα μπορούσαμε να κάνουμε για οποιοδήποτε άλλο πρωτόκολλο χρησιμοποιώντας την port στην οποία λειτουργεί, από την λίστα γνωστών ports[1].

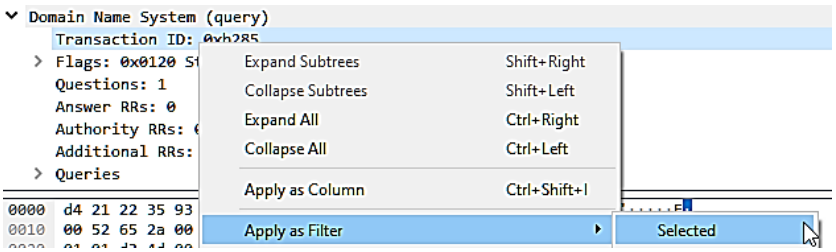


Εικόνα 3.9: Πεδίο Capture Filter από το παράθυρο Wireshark – Capture Interfaces

Apply as Filter

Για να βρούμε το response από τον DNS server πηγαίνουμε στο μεσαίο panel αφού έχουμε επιλέξει request και επεκτείνουμε τις πληροφορίες τους επιπέδου εφαρμογής. Βρίσκουμε το πεδίο **Transaction ID** και ενώ το έχουμε επιλεγμένο κάνουμε δεξί κλικ στο ποντίκι ώστε να εμφανιστεί το context menu και από τις επιλογές διαλέγουμε **Apply as Filter** και μετά **Selected.** Αυτός ο χειρισμός μπορεί να γίνει πάνω σε οποιοδήποτε πεδίο πληροφορίας για οποιοδήποτε πρωτόκολλο αναγνωρίζει το Wireshark.

[1] https://www.wikiwand.com/en/List_of_TCP_and_UDP_port_numbers



Εικόνα 3.10: Το context menu που εμφανίζεται στα πεδία πληροφοριών πρωτοκόλλου και η εφαρμογή φίλτρου από πληροφορία που εμφανίζει το Wireshark

Μπορούμε τώρα να απομονώσουμε τις αποκρίσεις επιλέγοντας από μια απόκριση το **Flags** και κατόπιν **Apply as Filter->Selected**

Apply as Column

Για να καταγράψουμε τους χρόνους απόκρισης στα ερωτήματα DNS μπορούμε να επιλέξουμε μια απόκριση και να βρούμε το χρόνο στο πεδίο [Response In: {χρόνος}]. Από το context menu των πεδίων πληροφορίας μπορούμε να επιλέξουμε **Apply as Column** και έτσι να δούμε το χρόνο απόκρισης σε όλες τις απαντήσεις των queries σε μια νέα στήλη στο πρώτο panel. Αφαιρώντας τα φίλτρα οι στήλες παραμένουν και έτσι μπορούμε να δούμε όλους τους χρόνους απόκρισης στα DNS queries.

Edit Column

Μπορούμε επίσης να αλλάξουμε την εμφάνιση και τα περιεχόμενα μιας στήλης, π.χ. της Time σε ότι επιθυμούμε. Αφού τοποθετήσουμε τον δείκτη πάνω στην κεφαλίδα της στήλης και κάνουμε δεξί κλικ με το ποντίκι εμφανίζεται ένα context menu για τις στήλες όπου υπάρχει η επιλογή **Edit Column**. Εκεί στο πεδίο Title μπορείτε να δώσετε τον επιθυμητό τίτλο.

Sort Column

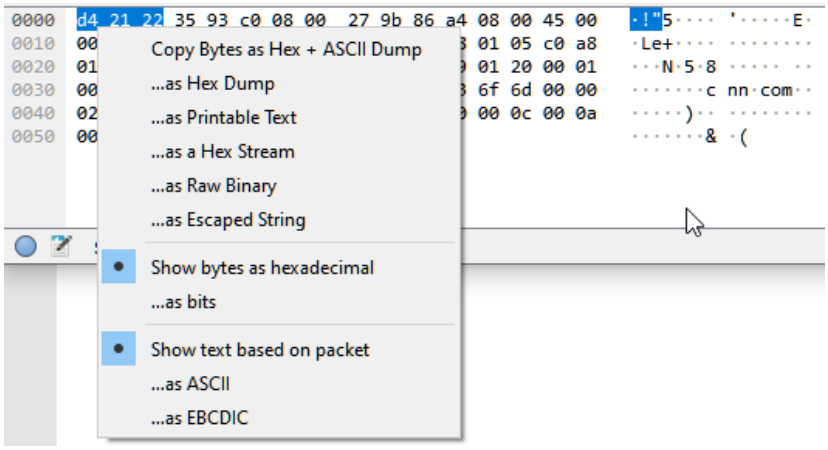
Με αριστερό κλικ πάνω στην κεφαλίδα της στήλης ταξινομείται είτε σε φθίνουσα ή αύξουσα σειρά η οποία γίνεται αντιληπτή από ένα βέλος που δείχνει προς την μικρότερη τιμή

No.	No.
1	76
2	75

Εικόνα 3.11: Αύξουσα και φθίνουσα ταξινόμηση στην στήλη με τον αύξων αριθμό του καταγεγραμμένου frame

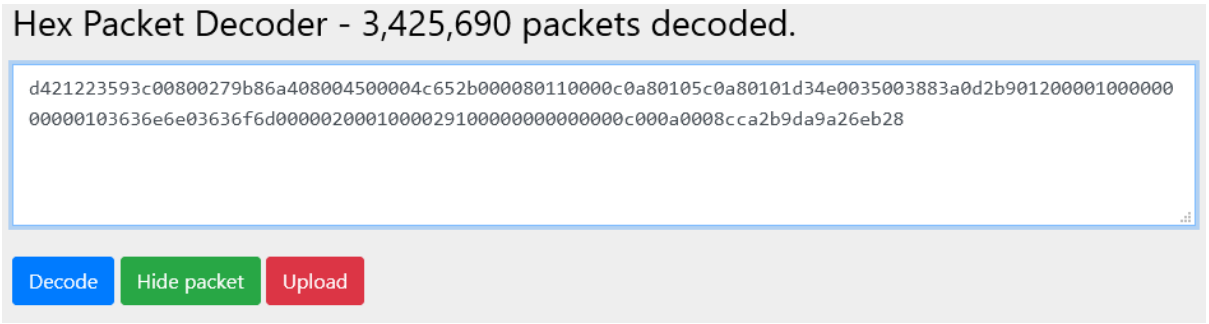
3.5 Ανάλυση ενός πλαισίου (frame) σε δεκαεξαδική μορφή

Η κάθε γραμμή στο Wireshark αντιστοιχεί σε ένα πλαίσιο (frame) που διακινήθηκε από και προς το επιλεγμένο NIC στο οποίο έτρεξε η καταγραφή. Μπορούμε να αντιγράψουμε τα bytes ή octets του σε δεκαεξαδική μορφή τοποθετώντας τον δείκτη του ποντικιού πάνω στην δεκαεξαδική εμφάνιση τους στο (κάτω δεξιά) panel, δεξί κλικ για το context menu, επιλογή (**Copy) … as Hex Stream**.



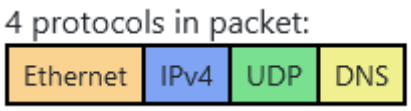
Εικόνα 3.12: Εξαγωγή καταγεγραμμένων bytes από το Wireshark

Κατόπιν μπορούμε να επικολλήσουμε την δεκαεξαδική μορφή σε οποιοδήποτε επεξεργαστή κειμένου ώστε να την αποθηκεύσουμε σε αρχείο. Θα χρησιμοποιήσουμε το online εργαλείο ανάλυσης πλαισίου Hex Packet Decoder https://hpd.gasmi.net/ στο οποίο μπορούμε να επικολλήσουμε απευθείας ένα οποιοδήποτε string που περιέχει hex bytes.



Εικόνα 3.13: Επικόλληση δεκαεξαδικής ροής στο εργαλείο ανάλυσης πακέτων HPD

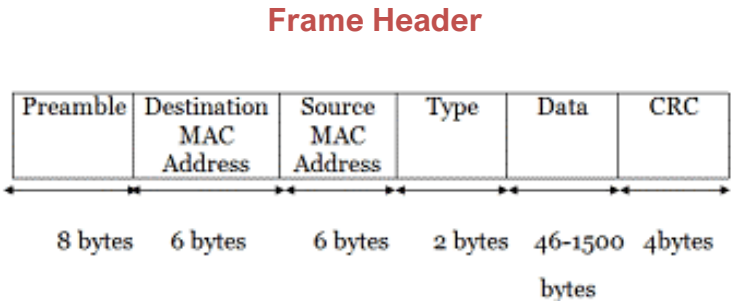
Εφόσον το string αντιστοιχεί σε ένα έγκυρο πακέτο θα εμφανίσει τον εγκιβωτισμό των PDUs στα διαφορετικά επίπεδα, χρωματίζοντας τα bytesπου προστίθενται από αυτά ξεκινώντας από τα δεδομένα του επιπέδου εφαρμογής (L5) προς το επίπεδο ζεύξης (L2). Θα εμφανιστεί ένα υπόμνημα όπως το παρακάτω.



Εικόνα 3.14: Χρωματισμός που επιδεικνύει τον εγκιβωτισμό των PDUs στο Εφαρμογής (κίτρινο), Μεταφοράς (πράσινο), Δικτύου (μπλε) και Ζεύξης (πορτοκαλί).

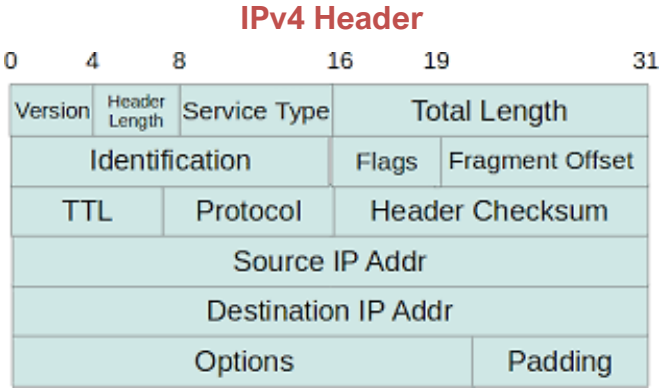
Άσκηση 3.9: Ενημέρωση στα πεδία των κεφαλίδων που υπάρχουν σε PDUs

Κατεβάστε το αρχείο **frame_example.pdf** από τον ιστότοπο του μαθήματος αντιγράψτε τα bytes στο online εργαλείο HPD. Προσπαθήστε να βρείτε την θέση των πεδίων αντίστοιχα με τις περιγραφές των κεφαλίδων των PDUs που υπάρχουν στις εικόνες 3.13, 3.14, 3.15 και 3.5 (για UDP). Κινήστε το ποντίκι πάνω στα bytes. Ποιες πληροφορίες μπορείτε να πάρετε από αυτό το εργαλείο;



Εικόνα 3.15: Bytes σε κάθε πεδίο για πλαίσιο Ethernet

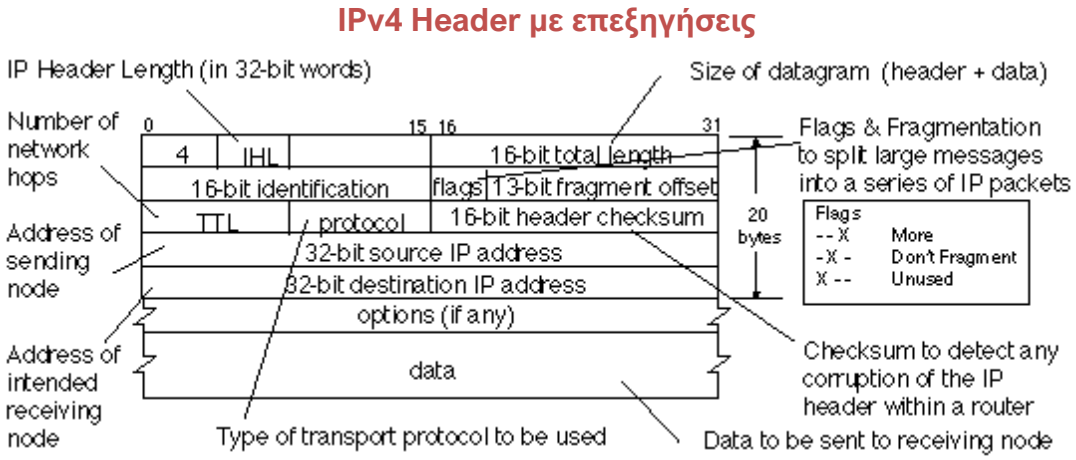
Σημειώνεται ότι στα δεδομένα που διέρχονται από ένα NIC και καταγράφονται με χρήση λογισμικού δεν υπάρχει το Preamble και το CRC.



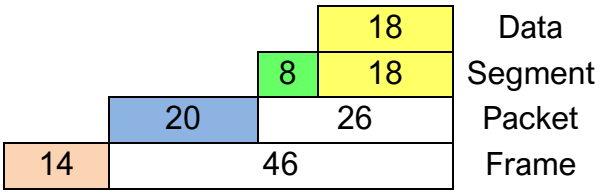
Εικόνα 3.16: Bytes σε κάθε πεδίο για πακέτο IPv4, με κάθε γραμμή να αντιστοιχεί σε 32 bytes

1. Ποιο είναι το μέγεθος των δεδομένων του επιπέδου εφαρμογής; Από ποιο πεδίο μπορούμε να το βρούμε. Υπάρχει η δυνατότητα να συνδυάσουμε και άλλο πεδίο και να βρούμε την ίδια απάντηση;
2. Μπορούμε από τις πληροφορίες που διαβάζουμε να κρίνουμε ότι αποστολέας και παραλήπτης βρίσκονται στο ίδιο υποδίκτυο; Τι χρειαζόμαστε για να είμαστε σίγουροι;

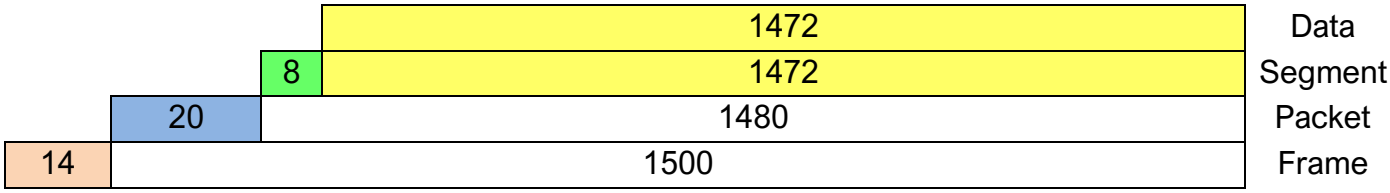
3. Γνωρίζοντας την διεύθυνση IPv4 του αποστολέα και του παραλήπτη, μπορούμε να έχουμε την πληροφορία του default gateway από την πλευρά του αποστολέα ή του παραλήπτη συνδυάζοντας πληροφορίες από το πλαίσιο που εξετάζουμε;



Εικόνα 3.17: Bytes σε κάθε πεδίο για πακέτο IPv4, με επεξηγήσεις



Εικόνα 3.18: Ελάχιστος αριθμός Bytes σε κάθε επίπεδο με UDP στο L4



Εικόνα 3.19: Μέγιστος αριθμός Bytes σε κάθε επίπεδο με UDP στο L4

3.6 Υπόμνημα: Πρωτόκολλο UDP

Το UDP ως πρωτόκολλο μεταφοράς παρέχει :

- ✓ γρήγορη εγκαθίδρυση της συνόδου (session) μεταφοράς,
- ✓ γρήγορη απόκριση,
- ✓ ελάχιστη επιβάρυνση σε πλήθος bytes,
- ✓ δεν χρειάζεται επαναπροσπάθειες (επειδή θεωρούμε αισιόδοξα ότι θα μεταφερθεί χωρίς προβλήματα),
- ✓ δεν υπάρχει επανασυναρμολόγηση (επειδή δεν γίνεται κατακερματισμός) και
- ✓ δεν απαιτεί παραλαβή μιας επιβεβαίωσης λήψης (acknowledgement).

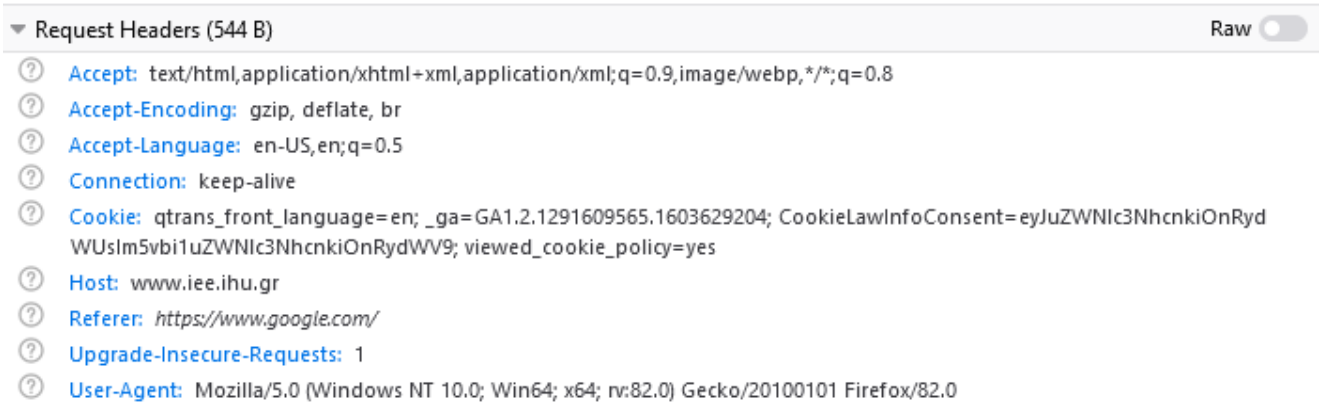
4. ΚΑΤΑΝΟΗΣΗ ΤΗΣ ΜΕΤΑΦΟΡΑΣ ΜΕ TCP ΜΕΣΑ ΑΠΟ ΤΟ ΠΡΩΤΟΚΟΛΛΟ HTTP

4.1 Τα επίπεδα της στοίβας TCP/IP στο world wide web

4.1.1 Το επίπεδο εφαρμογής

Μία διεργασία, που υλοποιεί κάποιο πρωτόκολλο επιπέδου εφαρμογής (application - L5) διακινεί δεδομένα μέσω του δικτύου, έχοντας ως αποδέκτη μια άλλη διεργασία που υλοποιεί το ίδιο πρωτόκολλο επιπέδου εφαρμογής. Οι σχέση των δύο διεργασιών ως προς την μεταξύ τους επικοινωνία περιγράφεται από το μοντέλο **πελάτη-εξυπηρετητή (client-server),** μια κατανεμημένη αρχιτεκτονική εφαρμογών κατά την οποία ο server προσφέρει υπηρεσίες που εξυπηρετούν τον client.

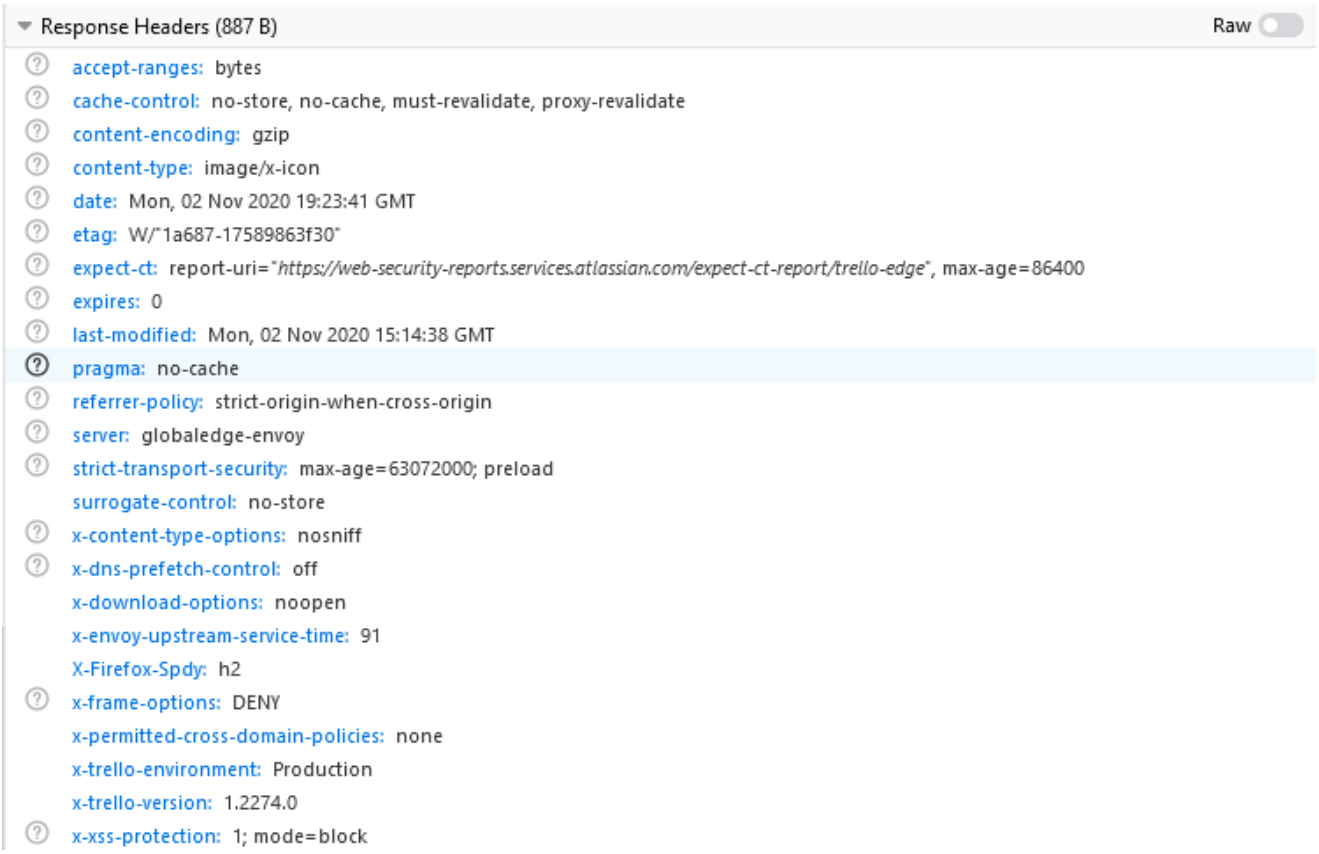
Η μονάδα πληροφορίας πρωτοκόλλου PDU στο επίπεδο εφαρμογής είναι τα δεδομένα (data) που ανταλλάσσονται μεταξύ client και server. Τα δεδομένα αυτά μπορεί να είναι δυαδικά (binary), π.χ. εικόνες JPEG/PNG, απλό κείμενο (text), κείμενο κάποιας δομημένης μορφής, π.χ. HTML, κομμάτι ροής πολυμέσων, π.χ. videoframes, κ.α. Τα PDU σε κάθε επίπεδο, επισυνάπτουν κεφαλίδες (headers) με πληροφορίες οργανωμένες σε πεδία, οι οποίες χρησιμοποιούνται για τον έλεγχο του διαλόγου μεταξύ των διεργασιών. Στην περίπτωση του πρωτοκόλλου HTTP το οποίο χρησιμοποιείται στον παγκόσμιο ιστό για κατέβασμα ιστοσελίδων από τον server (έναν ιστότοπο) σε έναν client (έναν browser) έχουμε προκαθορισμένα πεδία στα αιτήματα και στις αποκρίσεις, αλλά και προσαρμοσμένα πεδία που προσθέτουν οι εφαρμογές web.



Εικόνα 4.1: HTTP request headers όταν ανοίγουμε την ιστοσελίδα του τμήματος από τα αποτελέσματα αναζήτησης στο Google. Το πεδίο referrer δείχνει σε ποιο URL υπήρχε ο σύνδεσμος που ακολουθήσαμε ώστε να δημιουργηθεί το αίτημα προς τον ιστότοπο που καθορίζεται στο πεδίο Host

Σε ένα αίτημα HTTP (HTTP request) προς μια διεργασία web server το πεδίο κεφαλίδας `User-Agent` γνωστοποιεί τον τύπο της διεργασίας client, δηλαδή ποιον web browser χρησιμοποιεί ο χρήστης που ζητάει την ιστοσελίδα. Στην εικόνα 4.1 φαίνεται ότι το αίτημα προέρχεται από Firefox v82.0. Αυτό μπορεί στην συνέχεια να χρησιμοποιηθεί από την πλευρά του server ώστε

να στείλει διαφορετική απόκριση (HTTP response) ανάλογα με το αν έχουμε έναν web browser για desktop ή για mobile.



Εικόνα 4.2: HTTP response headers για το εικονίδιο της web εφαρμογής Trello. Στο πεδίο content-type καθορίζεται ότι τα δεδομένα που κατεβαίνουν στον browser είναι εικόνας (δυαδικά) και τύπου .ico. Παρατηρούμε ότι η εφαρμογή προσθέτει τα προσαρμοσμένα πεδία x-trello-environment και x-trello-version

Στην κεφαλίδα του HTTP response στο πεδίο `Server` υπάρχει το είδος και η έκδοση του web server που εξυπηρετεί το αίτημα. Από την σκοπιά του τομέα της Ασφάλειας Ιστού (Web Security) μπορεί να χρησιμοποιηθεί από κάποιον για να ελέγξει αν ένας web server είναι πλήρως ενημερωμένος στην τελευταία έκδοση. Στην εικόνα 4.2 παρατηρούμε ότι δεν βλέπουμε τον τύπο του πραγματικού webserver (Apache/IIS/Nginx) αλλά `globaledge-envoy` το οποίο είναι ένα proxy που αναλαμβάνει να στείλει την απάντηση αποκρύπτοντας τον πραγματικό server.

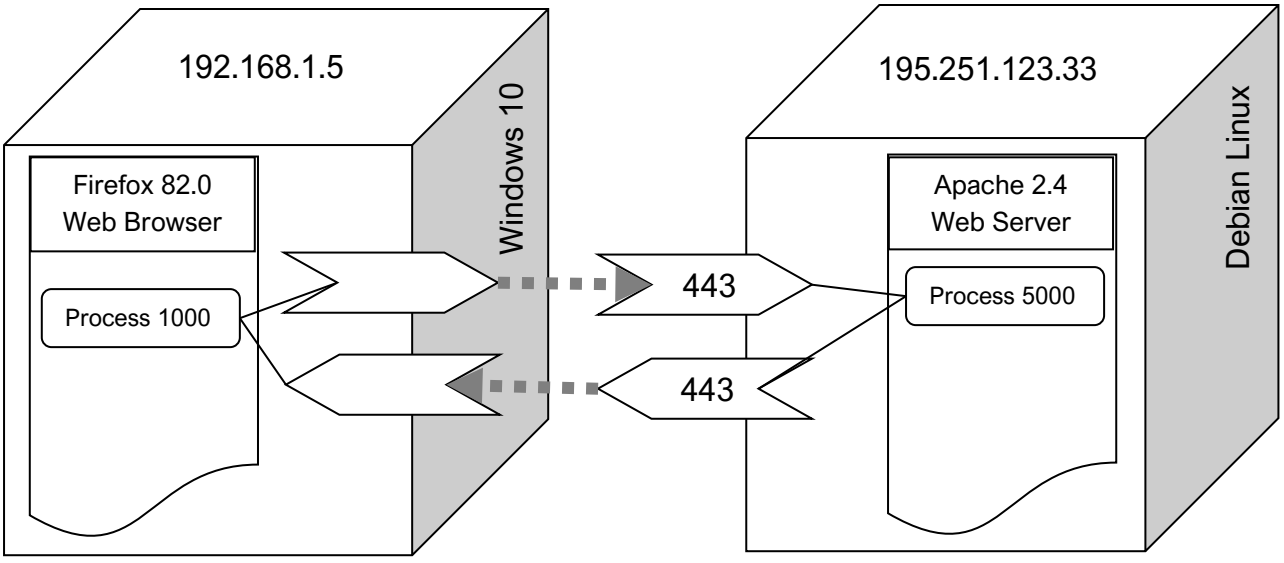
Μετά την αποστολή της απόκρισης ο web server τελειώνει τον κύκλο λειτουργίας του χωρίς να «θυμάται» πληροφορίες για τον client και το αίτημα, δηλαδή δεν διατηρεί κατάσταση (state). Όταν στον ίδιο web server αιτείται ο ίδιος web client μια δεύτερη ιστοσελίδα τότε αντιμετωπίζεται ξανά ως «νέος» αιτούμενος. Ο περιορισμός αυτός του πρωτοκόλλου HTTP, παρακάμπτεται με χρήση cookies για την ταυτοποίησης μιας συνεδρίας (session) χρήστη.

Γενικά οι κεφαλίδες υποδεικνύουν αν το μήνυμα είναι μία ερώτηση, ή απάντηση, αλλά και την μορφή των δεδομένων που ανταλλάσσουν οι δύο διεργασίες. Οποιοδήποτε πρωτόκολλο

επιπέδου εφαρμογής ορίζει ρητά το/τα πρωτόκολλο(-α) που χρησιμοποιούνται για την μεταφορά των δεδομένων μέσω του επιπέδου μεταφοράς.

4.1.2 Το επίπεδο μεταφοράς

Τα πρωτόκολλα επιπέδου μεταφοράς (transport - L4) εξυπηρετούν την μεταφορά δεδομένων που ζητείται από τα πρωτόκολλα επιπέδου εφαρμογής. Τα PDU του L4 επισυνάπτουν πληροφορίες πάνω στα δεδομένα του L5. Οι υπηρεσίες του L4 είναι υλοποιημένες ως τμήμα του λειτουργικού συστήματος (operating system - OS). Καλώντας αυτές από ανώτερου επιπέδου (higher level) γλώσσες προγραμματισμού τα δεδομένα μεταφέρονται από την διεργασία που τρέχει στον τοπικό υπολογιστή, προς μια διεργασία που τρέχει στον απομακρυσμένο υπολογιστή. Μια εφαρμογή έχει πολλές διεργασίες για να διεξάγει πολυεπεξεργασία, έχοντας μια διεργασία για να υλοποιήσει την γραφική διεπαφή χρήστη (GUI) και τουλάχιστον μια όταν θέλει παράλληλα και διαδικτυακή επικοινωνία.



Εικόνα 4.3: Σχηματική απεικόνιση μεταφοράς δεδομένων μεταξύ δύο διεργασιών. Στα αριστερά μια διεργασία του Firefox σε λειτουργικό Windows 10 και στα δεξιά μια διεργασία του server του ιστότοπου της σχολής. Οι αριθμοί πάνω στα βέλη αντιστοιχούν σε τοπικές και απομακρυσμένες θύρες (ports).

Ενώ στο επίπεδο δικτύου γίνεται προσπάθεια για δρομολόγηση πακέτων (packets) μεταξύ δύο hosts, στο επίπεδο μεταφοράς γίνεται μεταφορά τμημάτων (segments) μεταξύ διεργασιών που τρέχουν μέσα στους hosts. Αυτή η επέκταση λέγεται **πολύπλεξη (multiplexing)** και **αποπολύπλεξη επιπέδου μεταφοράς (transport-layer demultiplexing).** Για την αναγνώριση των διεργασιών που ανταλλάσσουν segments στο επίπεδο μεταφοράς υπάρχουν τα sockets. Σύμφωνα με το αρχικό πρότυπο RFC 147 προβλέπεται ένας 32bit αριθμός αναγνώρισης, αλλά θεωρούνται επίσης στην βιβλιογραφία ως συνδυασμός `{Διεύθυνση IP}:{Αριθμός Port}`, ο οποίος χρησιμοποιείται στην ανάπτυξη εφαρμογών TCP/IP. (Βλέπε και 1.2.4 1ης ενότητας)

Το επίπεδο δικτύου δεν εγγυάται την ακεραιότητα των πληροφοριών που μεταφέρονται, ούτε την μεταφορά των πακέτων με την σωστή σειρά. Υλοποιεί δηλαδή μια **υπηρεσία καλύτερης προσπάθειας (best-effort delivery service)**. Αυτά υλοποιούνται στο επίπεδο μεταφοράς όπως φαίνεται στην παρακάτω εικόνα

Επίπεδο Μεταφοράς (L4)

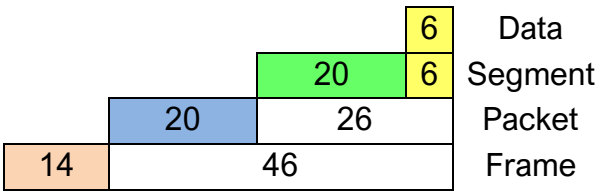
	TCP	UDP
Παράδοση από διεργασία σε διεργασία	ΝΑΙ	ΝΑΙ
Έλεγχος σφαλμάτων	ΝΑΙ	ΝΑΙ
Εγγύηση για την σωστή παραλαβή των δεδομένων του επιπέδου L5 εφαρμογής	ΝΑΙ	ΟΧΙ
Εγγύηση για την σωστή σειρά λήψης των τμημάτων (PDU στο L4)	ΝΑΙ	ΟΧΙ
Έλεγχος συμφόρησης	ΝΑΙ	ΟΧΙ

Επίπεδο Δικτύου (L3)

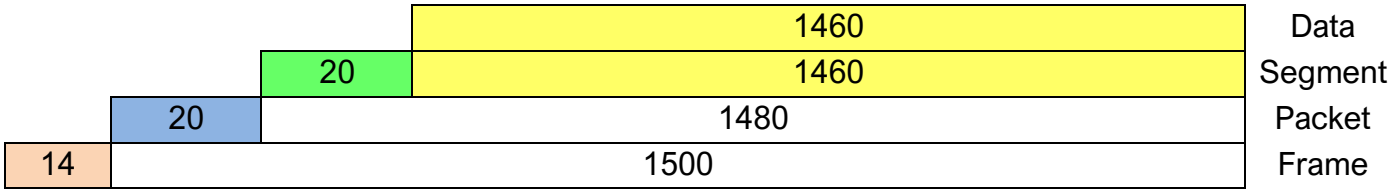
	IP
Παράδοση από σύστημα σε σύστημα	ΝΑΙ
Έλεγχος σφαλμάτων	Προαιρετικός
Εγγύηση για την σωστή παραλαβή των τμημάτωντου επιπέδου L4 μεταφοράς	ΟΧΙ
Εγγύηση για την σωστή σειρά λήψης των πακέτων (PDU στο L3)	ΟΧΙ
Έλεγχος συμφόρησης	ΟΧΙ

Εικόνα 4.4: Σύγκριση χαρακτηριστικών των πρωτοκόλλωνTCP, UDP και IP στο επίπεδο μεταφοράς (επάνω) και στο επίπεδο δικτύου (κάτω)

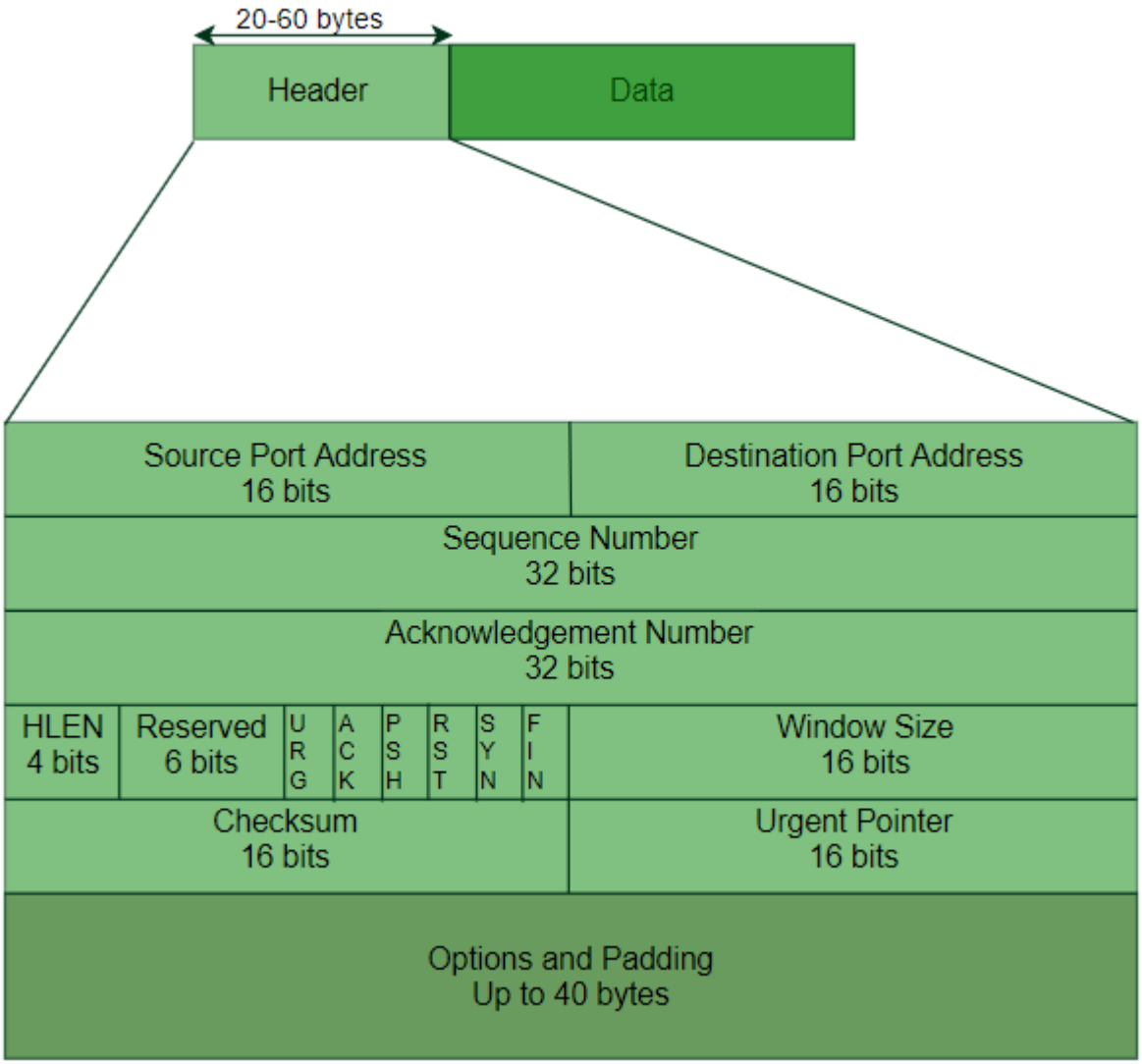
Επιπρόσθετα το πρωτόκολλο TCP μπορεί να κατατμήσει τα δεδομένα σε πολλά τμήματα (segments) τα οποία εν δυνάμει φτάνουν στο άλλο άκρο της επικοινωνίας χωρίς να κρατούν την σωστή σειρά. Στον παραλήπτη διασφαλίζεται από το TCP ότι τα δεδομένα του 5ου επιπέδου θα επανασυναρμολογηθούν στην σωστή σειρά από τα διάφορα τμήματα τους.



Εικόνα 4.5: Ελάχιστος αριθμός Bytes σε κάθε επίπεδο με TCP στο L4



Εικόνα 4.6: Μέγιστος αριθμός Bytes σε κάθε επίπεδο με TCP στο L4



Εικόνα 4.7: TCP header

4.2 Ανάλυση επικοινωνίας HTTP με Wireshark

4.2.1 Εργασία με αποθηκευμένο αρχείο καταγραφής

Κατεβάστε το αρχείο **http_capture.pcapng** από τον ιστότοπο του μαθήματος, ξεκινήστε το Wireshark και αντί για καταγραφή ανοίξτε το αρχείο που μόλις κατεβάσατε από το μενού File ->Open. Εντοπίστε τις γραμμές που περιέχουν στην στήλη protocol το HTTP με χρήση του κατάλληλου φίλτρου. Επιλέξτε την γραμμή που στην στήλη πληροφορίες (Info) υπάρχει το μήνυμα "**`GET / HTTP/1.1`**".

Εστιάστε στο μεσαίο panel του Wireshark. Η πρώτη γραμμή ξεκινάει με την λέξη Frame. Στην δεύτερη εμφανίζεται το πλαίσιο του πρωτοκόλλου Ethernet II του 2ου επιπέδου, στο οποίο ενθυλακώνεται ένα πακέτο του πρωτοκόλλου IPv4 του 3ου επιπέδου (τρίτη γραμμή). Σε αυτό ενθυλακώνεται ένα segment του πρωτοκόλλου TCP του 4ου επιπέδου (τέταρτη γραμμή), στο οποίο περιέχεται ένα αίτημα του πρωτοκόλλου HTTP του 5ου επιπέδου (πέμπτη γραμμή). **Σε συγκεκριμένες καταστάσεις ο αριθμός του επιπέδου (π.χ. 5ο) μπορεί να μην συμπίπτει με την θέση στο μεσαίο πάνελ.**

Αναπτύξτε την αντίστοιχη γραμμή για να πάρετε τις πληροφορίες που ζητούνται από το πρωτόκολλο εφαρμογής. Την επικοινωνία τύπου client-server την ξεκινά το πρόγραμμα πελάτη. Το **HTTP** χρησιμοποιεί **TCP** για την μεταφορά και ο client εκκινεί αρχικά μια σύνδεση TCP με τον εξυπηρετητή. Αφού γίνει η σύνδεση οι διεργασίες του πελάτη και του εξυπηρετητή έχουν προσπέλαση στο TCP μέσω των διεπαφών που έχουν τα socket τους. Έτσι ο πελάτης στέλνει μηνύματα αιτήσεων (requests) HTTP στη διεπαφή socket και λαμβάνει μηνύματα απόκρισης (response) HTTP από τη διεπαφή socket.

4.2.2 Ανάλυση μηνύματος αίτησης HTTP (HTTP request)

Τα μηνύματα αιτήσεων HTTP έχουν τη γενική μορφή (εικόνα 2.8 στη σελίδα 105 του βιβλίου της θεωρίας 7η εκδ. ή στη σελίδα 102 του βιβλίου της θεωρίας 8η εκδ.):

- Γραμμή αίτησης (πρώτη γραμμή)
- Γραμμές κεφαλίδας (πολλαπλές γραμμές)
- Κενή γραμμή
- Σώμα οντότητας (αν υπάρχει, εξαρτάται από τη μέθοδο στη γραμμή αίτησης)

Η μέθοδος (Request Method) ως πεδίο μπορεί να έχει τις τιμές (ανάλογα με την έκδοση):

GET: Ο πελάτης ζητά ένα αντικείμενο από τον εξυπηρετητή.

POST: Ο πελάτης ζητά πάλι ένα αντικείμενο αλλά τα περιεχόμενα της ιστοσελίδας του εξυπηρετητή εξαρτώνται από τα πεδία μιας φόρμας που έχει συμπληρώσει και αποστέλλει.

HEAD: Ο εξυπηρετητής θα αποκριθεί αλλά χωρίς να στείλει το αντικείμενο. Χρησιμοποιείται συνήθως για αποσφαλμάτωση.

PUT: Ο πελάτης φορτώνει ένα αντικείμενο στον εξυπηρετητή, εφόσον ο δεύτερος το επιτρέπει.

DELETE: Ο πελάτης διαγράφει ένα αντικείμενο στον εξυπηρετητή, εφόσον ο δεύτερος το επιτρέπει.

Οι σημαντικότερες γραμμές κεφαλίδας αιτήματος που εμφανίζονται όταν γίνεται επέκταση του Hypertext Transfer Protocol στο Wireshark είναι:

- `Host:` Μας δείχνει την ονομασία του εξυπηρετητή στον οποίο αποστέλλεται το αίτημα (web server)
- `User-Agent:` Μας δείχνει το πρόγραμμα πελάτη (web browser)
- `Connection`: Μας δείχνει αν έχουμε παραμένουσα ή μη παραμένουσα σύνδεση.

Οι γραμμές που αρχίζουν με `Accept` μας δείχνουν τι είναι αποδεκτό από την πλευρά του πελάτη.

4.2.3 Ανάλυση μηνύματος απόκρισης (HTTP response)

Τα μηνύματα αποκρίσεων HTTP έχουν τη γενική μορφή (εικόνα 2.9 στη σελίδα 107 του βιβλίου της θεωρίας 7η εκδ. ή στη σελίδα 104 του βιβλίου της θεωρίας 8η εκδ.):

- Γραμμή κατάστασης (έκδοση, κωδικός κατάστασης, μήνυμα κατάστασης)
- Γραμμές κεφαλίδας (πολλαπλές γραμμές)
- Κενή γραμμή
- Σώμα οντότητας (είναι το ουσιαστικό περιεχόμενο του μηνύματος)

Οι κωδικοί κατάστασης (status code) ως πεδίο, μπορεί να είναι:

- 1xx: Πληροφοριακά μηνύματα
- 2xx: Μηνύματα επιτυχίας
- 3xx: Μηνύματα αλλαγής της διαδρομής του αντικειμένου
- 4xx: Μηνύματα σφάλματος που σχετίζονται με τον πελάτη
- 5xx: Μηνύματα σφάλματος που σχετίζονται με τον εξυπηρετητή

Μια πλήρη λίστα με τους κωδικούς κατάστασης HTTPυπάρχει στο
https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

Οι σημαντικότερες γραμμές κεφαλίδας απόκρισης που εμφανίζονται όταν γίνεται επέκταση του Hypertext Transfer Protocol στο Wireshark είναι:

- `Date`: δηλώνει την ημερομηνία και ώρα της απόκρισης από τον εξυπηρετητή.
- `Server`: το πρόγραμμα εξυπηρετητή.
- `Content-Location`: μας δείχνει τη διαδρομή και το αντικείμενο της απόκρισης.
- `Content-Encoding`: δηλώνει το είδος της κωδικοποίησης του αντικειμένου.
- `Content-Length`: το μέγεθος του αντικειμένου σε bytes.

- `Content-Type`: το είδος του αντικειμένου.
- `Content-Language`: η γλώσσα του αντικειμένου.

Οι γραμμές που αρχίζουν με `Accept` μας δείχνουν τι είναι αποδεκτό από την πλευρά του εξυπηρετητή.

ΑΣΚΗΣΗ 4.1: Ανάλυση αιτήματος HTTP.

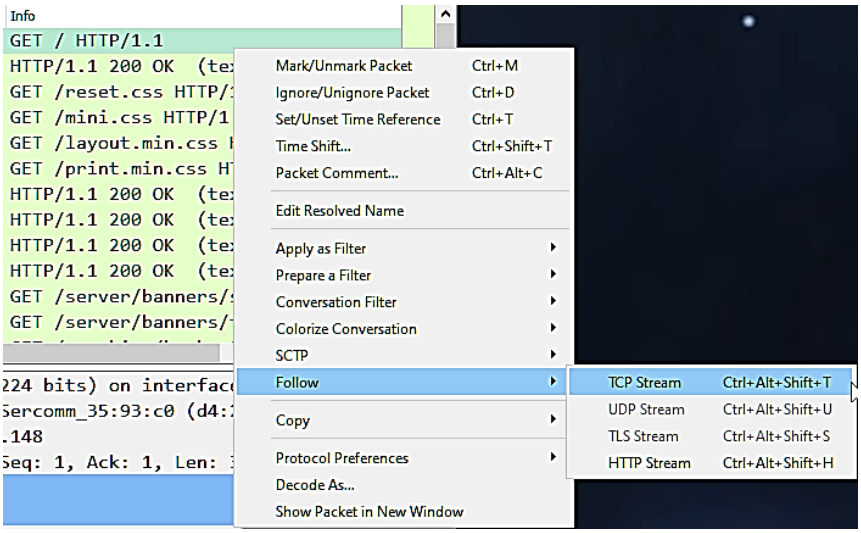
Έχετε ήδη ανοικτό το αρχείο **http_capture.pcapng** και έχετε εντοπίσει την γραμμή για το αίτημα **"`GET / HTTP/1.1`",** σύμφωνα με τις οδηγίες που υπάρχουν στην παράγραφο 4.2.1

1. Καταγράψτε την source και την destination IP διεύθυνση της επικοινωνίας.
2. Καταγράψτε το source και το destination port που χρησιμοποιήθηκαν.
3. Καταγράψτε το socket από το οποίο στέλνει ο web browser το αίτημα προς τον web server και το socket του δεύτερου που το παραλαμβάνει.
4. Παρατηρώντας και άλλες γραμμές αιτημάτων GET, ποιο είναι το σταθερό (γνωστό) port στο οποίο αναμένει αιτήματα ένας web server;
5. Ποιο είναι το πρωτόκολλο επιπέδου μεταφοράς που χρησιμοποιεί το πρωτόκολλο επιπέδου εφαρμογής HTTP;
6. Βρείτε την γραμμή HTTP request που στέλνει ο browser για να κατεβάσει το αρχείο `/reset.css` από τον server. Παρατηρήστε τους HTTP headers και βρείτε σε ποιον HTTP host έγινε το αίτημα.
7. Στην γραμμή που βρίσκεστε από το ερώτημα 6: Συνδυάζοντας την τιμή του πεδίου host και το αρχείο από το αίτημα GET προσπαθήστε να βρείτε το URL με το οποίο μπορούμε να το κατεβάσουμε στο browser. Χρησιμοποιήστε το στον browser σας και δείτε τα περιεχόμενα.

4.2.4 Χρήσιμες τεχνικές για την αποδοτικότερη χρήση του Wireshark στο TCP

Follow -> TCP Stream

Για να δούμε όλη την επικοινωνία μαζί με τα τμήματα TCP και την σύνδεση/αποσύνδεση μπορούμε σε κάποιο αίτημα/απόκριση HTTP να κάνουμε την επιλογή από το context menu (δεξί πλήκτρο) της γραμμής που εμφανίζεται στην παρακάτω εικόνα:



Εικόνα 4.8: Λειτουργία FollowTCPStreamαπό γραμμή αιτήματος HTTP

Κλείνοντας το παράθυρο που εμφανίζει το stream, έχει εφαρμοστεί ένα φίλτρο το οποίο δείχνει αφενός την γραμμή HTTP έχοντας ένα βέλος στην στήλη No και αφετέρου όλα τα τμήματα TCP έχοντας κυκλικές κουκίδες.

Previous Packet in History: Όταν πηγαίνουμε σε μια διαφορετική γραμμή στο επάνω panel μπορούμε να επιστρέψουμε στην προηγούμενη με Go ->Previous Packet in History ή πατώντας την συντόμευση Alt + Left Arrow.

ΑΣΚΗΣΗ 4.2: Ανάλυση απόκρισης HTTP και κατάτμησης TCP.

Συνεχίζοντας την ανάλυση από την άσκηση 4.1:

1. Εντοπίστε την απόκριση (response) του HTTP server για το αίτημα GET που ζητάει τον ριζικό κατάλογο του ιστότοπου. (Υπάρχει ένα οπτικό βοήθημα στην στήλη με τίτλο "No.” του Wireshark.)
2. Για κάθε επίπεδο του μοντέλου πρωτοκόλλων του TCP/IP αναγνωρίστε τα πρωτόκολλα που πήραν μέρος στην επικοινωνία σας με τον server. Τι παρατηρείτε σε σχέση με τις γραμμές του μεσαίου panel.
3. Καταγράψετε τα source και destination sockets.
4. Εκτός της αντιστροφής μεταξύ source και destination socket, τι άλλο παρατηρείτε όταν συγκρίνετε την κατεύθυνση της επικοινωνίας του ερωτήματος με την αντίστοιχης απάντησης;
5. Η απόκριση επανασυναρμολογείται στον παραλήπτη από πολλά τμήματα TCP που μεταφέρονται από τον web server. Πόσα είναι αυτά; (Το Wireshark τα ομαδοποιεί σε μια εμβόλιμη γραμμή μεταξύ των γραμμών που αντιστοιχούν στο 4ο και 5ο επίπεδο).
6. Παρατηρήστε τα bytes από τα οποία αποτελείται το καθένα. Ποιο είναι το μέγιστο μέγεθος του τμήματος σε bytes.
7. Ποιο είναι το συνολικό μέγεθος (Length) σε bytes της επανασυναρμογής των τμημάτων TCP;

8. Αν το συνολικό μέγεθος ήταν 28080 bytes πόσα τμήματα θα είχαμε και πόσα bytes θα ήταν στο τελευταίο τμήμα;

ΑΣΚΗΣΗ 4.3: Πλοήγηση στην αλληλουχία TCP segments της απόκρισης HTTP.

Συνεχίζοντας την ανάλυση από την άσκηση 4.2 και με χρήση των τεχνικών της παραγράφου 4.2.3: Καταγράψτε τον αριθμό γραμμής (στήλη No.) για την απάντηση. Ακυρώστε τα φίλτρα εμφάνισης βάζοντας κενό και πατώντας Enter ή κάντε Follow TCP Stream πάνω σε αυτήν την γραμμή. Εντοπίστε το πρώτο τμήμα (για τα bytes 0-1393) από την λίστα των Reassembled TCP Segments και κάντε διπλό κλικ πάνω στον σύνδεσμο που εμφανίζει γραμμή της μορφής [Frame: {Αριθμός},payload: 0-1393 ({Μέγεθος} bytes)]

1. Τι είδους PDU μας εμφανίζει η γραμμή στην οποία μας πήγε το Wireshark και σε ποιο επίπεδο ανήκει;
2. Στην γραμμή που αναφέρεται το ερώτημα 1 καταγράψτε το δεκαεξαδικό άθροισμα ελέγχου (Checksum). Σε τι μας χρησιμεύει;
3. Επιστρέψτε στον αριθμό γραμμής της απάντησης (διπλό κλικ στο link της μορφής [Reassembled PDU in frame: {Αριθμός}] ή Go ->Previous Packet in History ή Alt-Left Arrow) και κρατείστε τους αριθμούς γραμμής του τέταρτου και πέμπτου TCP segment κατά σειρά. Μετακινηθείτε στην γραμμή του τέταρτου segment (απευθείας ή μέσω του link) και καταγράψτε το Next sequence number. Κατόπιν μετακινηθείτε στην γραμμή του πέμπτου segment και δείτε το Sequence number. Ως βοήθημα για τις ερωτήσεις 3, 4, 5 μπορείτε να χρησιμοποιήσετε τον παρακάτω πίνακα:

Σειρά εντός της συναρμογής	No.	Sequence Number	Next sequence number
5			
6			
7			
8			

   Τι συμπέρασμα βγαίνει έχοντας κατά νου ότι τα δύο segment είναι διαδοχικά.
4. Από την γραμμή του πέμπτου segment στο ερώτημα 3, καταγράψτε το Next sequence number. Κινηθείτε στις επόμενες γραμμές της καταγραφής και μέσω των Next sequence number και Sequence number, βρείτε ποιες γραμμές είναι το έκτο και το έβδομο segment εντός της συναρμογής των δεδομένων HTTP.
5. Από την κατανόηση που αποκτήσατε για τα TCP segments στις ερωτήσεις 3 και 4, ποια δομή δεδομένων υλοποιείται μεταξύ των διαδοχικών TCP segments;

4.2.5 Καταγραφή επικοινωνίας HTTP και HTTPS με Wireshark

ΑΣΚΗΣΗ 4.4: Καταγραφή επικοινωνίας HTTP

Κλείστε όλες τις καρτέλες στον browser που χρησιμοποιείτε ή ανοίξτε ένα νέο ιδιωτικό παράθυρο και γράψτε στην γραμμή URL http://testfire.net/login.jsp. Αν δεν εμφανιστούν γραμμές HTTP μετά την εφαρμογή του φίλτρου, είναι επειδή βρέθηκε στην web cache

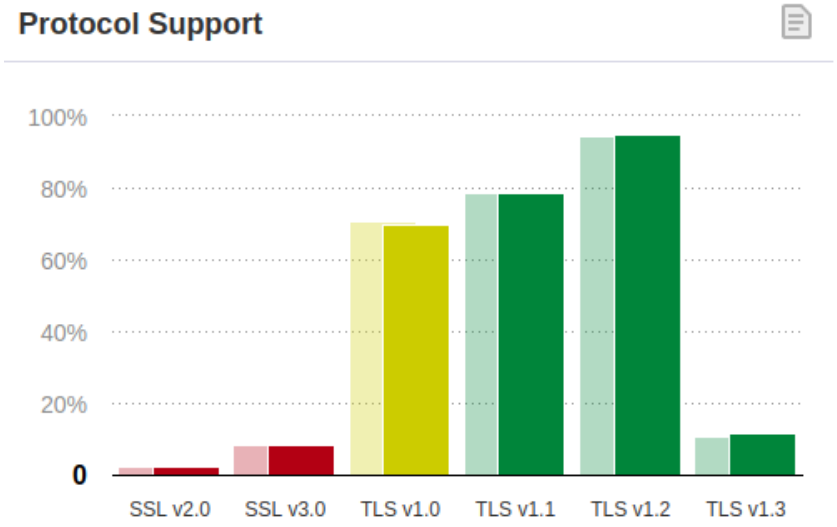
(παράγραφος 3.1.2). Στο πεδίο Username πληκτρολογήστε **jsmith** και στο Password **demo1234**, και μετά κάντε κλικ στο Login. Αφού εμφανιστεί η σελίδα σταματήστε την καταγραφή του Wireshark και εφαρμόστε το κατάλληλο φίλτρο για να βλέπετε μόνο τις γραμμές HTTP.

1. Στο επάνω panel επιλέξτε την γραμμή με το αίτημα τύπου POST /doLogin και στο μεσαίο πάνελ κάντε κλικ στην γραμμή του πρωτοκόλλου HTTP. Διαβάστε το αναγνώσιμο κείμενο στο κάτω panel. Τι παρατηρείτε σε σχέση με τα διαπιστευτήρια που χρησιμοποιήσατε;
2. Μετά την γραμμή του 5ου επιπέδου υπάρχει μια επιπρόσθετη που εμφανίζει τα πεδία της φόρμας που έγινε POST. Εντοπίστε την θέση του στοιχείου “btnSubmit” στο κάτω panel.

ΑΣΚΗΣΗ 4.5: Καταγραφή επικοινωνίας HTTPS

Ξεκινήστε μια καταγραφή στο Wireshark και ανοίξτε τη σελίδα https://login.it.teithe.gr/ όπου θα κάνετε είσοδο με τα στοιχεία σας. Εφαρμόστε το κατάλληλο φίλτρο για να βλέπετε μόνο την καταγραφή HTTPS. Σε αυτήν την περίπτωση η σύνταξη του φίλτρου θα είναι της μορφής `tcp.port==`**`{αριθμός port για HTTPS}`**. Η port που αντιστοιχεί μπορεί να βρεθεί ανάμεσα στις 14 κυριότερες: https://opensource.com/article/18/10/common-network-ports. Τέλος επιλέξτε κάποια γραμμή με Application Data και αναπτύξτε το Transport Layer Security (TLS) στο μεσαίο Panel.

1. Ποια είναι η έκδοση του πρωτοκόλλου TLS; Ποια είναι η τελευταία έκδοση αυτού του πρωτοκόλλου, δείτε στην εικόνα και αναζητήστε περισσότερα online.



Εικόνα 4.9: Οι διαφορετικές εκδόσεις TLS που χρησιμοποιούνται (Φεβρουάριος 2019) Πηγή: https://medium.com/iocscan/transport-layer-security-tls-ssl-8e02b6d1d648

2. Κάντε κλικ στο Encrypted Application Data. Τι παρατηρείτε σε σχέση με την αναγνωσιμότητα των δεδομένων του HTTPS σε σχέση με αυτά που είδαμε σε μια καταγραφή HTTP;
3. Σε ποιο επίπεδο της στοίβας πρωτοκόλλων TCP/IP ανήκει το TLS;

4. Σύμφωνα με την κρίση σας ποια είναι η σημασία της χρήσης ενός πιο ενημερωμένου πρωτοκόλλου για Transport Layer Security; Μπορείτε να ψάξετε περισσότερα online για τις παρωχημένες εκδόσεις ώστε να οδηγηθείτε στην απάντηση.