

Όνοματεπώνυμο: Παναγιώτης Σταματόπουλος		Ομάδα: 6
Όνομα PC/ΛΣ: TakisAsus/Windows 10		Ημερομηνία: 27/11/23
Διεύθυνση IP: 192.168.2.2	Διεύθυνση Mac: D4-5D-64-59-27-5F	

AM: el20096

Εργαστηριακή Άσκηση 7

Πρωτόκολλα TCP και UDP

Άσκηση 1:

*Στον προσωπικό υπολογιστή με σύνδεση στο δίκτυο της σχολής με IP: 147.102.238.148
MAC: 70-66-55-41-70-C3

- 1.1: host 147.102.238.148
- 1.2: ip.dst == 1.1.1.1 or ip.dst == 2.2.2.2 or ip.dst == 147.102.40.1
- 1.3: Θύρα 23 Telnet Protocol
- 1.4: tcp.port == 23
- 1.5: SYN
- 1.6: 5 προσπάθειες σε κάθε περίπτωση
- 1.7: Η χρονική απόσταση από την πρώτη προσπάθεια είναι σε δυνάμεις του 2: $1 \rightarrow 2 \rightarrow 4 \rightarrow 8$
- 1.8: Δεν παρατηρείται κάποια σημαντική διαφορά
- 1.9: Μόνο το πρώτο βήμα
- 1.10: Απλώς εγκαταλείπει την προσπάθεια
- 1.11: tcp.port == 23 and ip.addr == 147.102.40.1
- 1.12: 5 προσπάθειες
- 1.13: Οι προσπάθειες γίνονται μετά από 0.5 sec περίπου από την τελευταία προσπάθεια
- 1.14: Accurate ECN, Congestion Window Reduced, ECN-Echo, Urgent, Acknowledgement, Push, Reset, Syn και Fin
- 1.15: RST
- 1.16: 20 bytes header, 0 bytes data
- 1.17: Source Port: 16 bits - Destination Port: 16 bits
Sequence Number: 32 bits
Acknowledgement Number: 32 bits
Header Length: 4 bits - Flags: 12 bits - Window: 16 bits

Checksum: 16 bits - Urgent Pointer: 16 bits

- 1.18: Το Data Offset είναι ο αριθμός των 32-bit λέξεων της επικεφαλίδας TCP. Στο Wireshark το πεδίο αυτό είναι το Header Length.
- 1.19: Η τιμή στα περιεχόμενα του πακέτου πολλαπλασιάζεται με 32 bit = 4 byte, στην περίπτωση μας $4 * 5 = 20$
- 1.20: Δεν υπάρχει
- 1.21: Από το Total Length πεδίο του IPv4 αφαιρούμε το Header Length του IPv4. Για τα δεδομένα αφαιρούμε και το Header Length του TCP
- 1.22: 32 bytes
- 1.23: Υπάρχει λόγω του πεδίου Options που εμφανίζεται στο τεμάχιο που στέλνει ο υπολογιστής

Άσκηση 2:

*Στον προσωπικό υπολογιστή

- 2.1: tcp and ip host edu-dy.cn.ntua.gr
- 2.2: Port 21: FTP Control
- 2.3: Port 20: FTP Data Transfer
- 2.4: tcp.port == 21
- 2.5: 3 τεμάχια
- 2.6: SYN και Acknowledgment
- 2.7: 32 bytes, 32 bytes και 20 bytes
- 2.8: 0 και για τα 3
- 2.9: 0.23118
- 2.10: Συμφωνεί
- 2.11: Υπολογιστής: 0/2663449732
Server: 0/1072058939
- 2.12: Sequence Number του τεμαχίου που έλαβε +1
- 2.13: SYN No: Sequence Number του πρώτου τεμαχίου +1
ACK No: Sequence Number του δεύτερου τεμαχίου +1
- 2.14: 0 σε κάθε τεμάχιο
- 2.15: Το μέγεθος του πεδίου είναι 32 bits επομένως η μέγιστη τιμή είναι $2^{32} - 1$
- 2.16: tcp.port == 21 and tcp.len == 0 and ((tcp.seq == 0 and tcp.ack == 0) or (tcp.seq == 0 and tcp.ack == 1) or (tcp.seq == 1 and tcp.ack == 1))
- 2.17: Control: 8192
Data Transfer: 65535
- 2.18: Control: 65535
Data Transfer: 65535 → 1030
- 2.19: Window
- 2.20: Control: 0 → 8
Server: 6
- 2.21: TCP Option – Window scale
- 2.22: 1460 bytes
- 2.23: MSS = MTU – 40 = 1500 – 40 = 1460 bytes
- 2.24: TCP Option – Maximum segment size
- 2.25: 536 bytes

- 2.26: $MSS = MTU - 40 = 576 - 40 = 536$ bytes
- 2.27: $TCP = MSS + TCP\ Header\ Length = 536 + 20 = 556$ bytes
- 2.28: Fin
- 2.29: O Server
- 2.30: 4
- 2.31: 20 bytes σε όλα
- 2.32: 0 bytes σε όλα
- 2.33: IPv4 = 20 bytes Header Length + 20 bytes TCP
Ethernet = 14 bytes γιατί το Wireshark κάνει capture πριν το padding του frame
- 2.34: IPv4 = 20 bytes Header Length + 20 bytes TCP
Ethernet = 20 bytes = 14 bytes Header + 6 bytes padding
(Ethernet frame \geq 60 bytes)
- 2.35: Από τον υπολογιστή: 929 bytes
Από τον Server: 973 bytes
- 2.36: Υπολογιστή: tcp.port == 21 and ip.src == 147.102.40.15
Server: tcp.port == 21 and ip.dst == 147.102.40.15
Προσθέτουμε το Length στα πακέτα για κάθε περίπτωση
- 2.37: tcp.port == 20
- 2.38: Server: 536 bytes
Υπολογιστής: 1460 bytes
- 2.39: 556 bytes = 536 bytes MTU + 20 bytes TCP Header
- 2.40: 0.000122 seconds
- 2.41: Όχι
- 2.42: 115 πακέτα
- 2.43: 59 πακέτα
- 2.44: $512 * 256 = 131072$ bytes
- 2.45: Όχι, το Window size αυξάνεται για να δεχθεί τα δεδομένα που θα μεταφερθούν μέσω του FTP
- 2.46: Αλλάζει μόνο στην αποσύνδεση του FTP και γίνεται 130560
- 2.47: Θα περίμενε τον υπολογιστή να στείλει μήνυμα για να συνεχίσει τη μεταφορά δεδομένων
- 2.48: 590 bytes Frame, 14 bytes Ethernet, 20 bytes IPv4, 20 bytes TCP
- 2.49: 536 bytes ναι είναι στο όριο της MTU

2.50: Θα πρέπει ο αποστολέας να κάνει fragment τα πακέτα πριν τα αποστείλει

2.51: Server \rightarrow Computer: $114 * 536 + 510 = 61614 \text{ bytes}$

Computer \rightarrow Server: 0 bytes δεδομένων

2.52: $\frac{61614}{0.114542} = 525.3088114 \text{ kbyte/sec}$

2.53: Δεν υπήρχε Retransmission TCP πακέτο, άρα δεν είχαμε αναμετάδοση

Άσκηση 3:

*Στον προσωπικό υπολογιστή

3.1: `tcp.port == 20`

3.2: 94.65.141.44

3.3: 0.014626 sec σημαντικά μεγαλύτερο από το 2.40

3.4: Παρατηρούμε ότι τα πακέτα στέλνονται σε ριπές αυξανόμενου μεγέθους

3.5: Στέλνει 4 τεμάχια, όπως προβλέπει το RFC 5681 για $SMSS \leq 1095$ bytes

3.6: 6, 10, 16 τεμάχια αντίστοιχα

3.7: 1, 2, 3 τεμάχια αντίστοιχα: 4 τεμάχια \rightarrow 1 ACK, 6 τεμάχια \rightarrow 2 ACK, 10 τεμάχια \rightarrow 3 ACK σύμφωνα με το RFC 3465 θέτουμε το όριο $L = 1 * SMSS$ (536) bytes, δηλαδή στέλνουμε 1 ACK ανά 4 segments

3.8: Η καταγραφή είναι παρόμοια με την προηγούμενη, με τη διαφορά ότι στέλνονται 2 τεμάχια για κάθε ACK με μια μικρή χρονική καθυστέρηση, όχι σαν τις ριπές της προηγούμενης καταγραφής

Άσκηση 4:

*Στον προσωπικό υπολογιστή

4.1: udp

4.2: Source Port: 2 bytes

Destination Port: 2 bytes

Length: 2 bytes

Checksum: 2 bytes

4.3: 8 bytes

4.4: Από το IPv4 Total Length = 70 bytes

4.5: Το μήκος όλου του δεδομενογράμματος UDP
(Header + Data)

4.6: Min Length = UDP Header = 8 bytes

4.7: Min Data = 0 bytes

Max Data = Max IPv4 Total Length – Min IPv4 Header

Length - UDP Header Length = $2^{16} - 1 - 20 - 8 = 65507$

4.8: Max Data = $576 - 60(\text{Max IPv4 Header Length}) -$
 $8(\text{UDP Header Length}) = 508 \text{ bytes}$

4.9: Όχι, εμφανίστηκαν μόνο DNS πρωτόκολλα

4.10: dns

4.11: 192.168.2.1

4.12: Source Port: 49913

Destination Port: 53

4.13: Source Port: 53

Destination Port: 49913

4.14: Port 53