Ονοματεπώνυμο: ΔΗΜΗΤΡΙΟΣ ΓΕΩΡΓΟΥΣΗΣ	ΑΜ: 03119005	Ομάδα: 4
Όνομα PC/ΛΣ: dimitris-Laptop / WINDOWS 11 HOME	Ημερομηνία: 24/11/2022	
Διεύθυνση IPv4: 192.168.2.5	Διεύθυνση MAC: 7C-8A-E1-C3-47-5C	
Διεύθυνση IPv6: fe80::1fce:88b8:c432:df51%8 (link local)		

1 Μετάδοση δεδομένων με TCP

1.1: Σύνταξη φίλτρου σύλληψης: ip host 192.168.2.5

1.2: Φίλτρο απεικόνισης: ip.dst == 1.1.1.1 or ip.dst == 2.2.2.2 or ip.dst == 147.102.40.1

1.3: Port (άλλου υπολογιστή) = 23 (χρησιμοποιείται για το telnet)

1.4: Φίλτρο απεικόνισης: tcp.port = 23 ή

(ip.dst == 1.1.1.1 or ip.dst == 2.2.2.2 or ip.dst == 147.102.40.1) and tcp.port == 23, αν θέλουμε μόνο τα εξερχόμενα πακέτα προς αυτούς τους προορισμούς

1.5: Σημαία που είναι ενεργοποιημένη για την εκκίνηση της επικοινωνίας: Syn

1.6: Κάνει την πρώτη προσπάθεια και 4 retransmissions άρα 5 προσπάθειες

1.7: Είναι 1, 2, 4, 8 seconds

1.8: Και στις δύο περιπτώσεις τα πακέτα των προσπαθειών σύνδεσης είναι ίδια. Τα πακέτα των A, B είναι διαφορετικά μεταξύ τους, αφού έχουν διαφορετικό Sequence Number (raw)

1.9: Γίνεται μόνο το πρώτο βήμα (ο υπολογιστής μας στέλνει το Syn)

1.10: Απλώς εγκαταλείπει την προσπάθεια (δεν βλέπουμε να στάλθηκε Fin από τον υπολογιστή μας)

1.11: Φίλτρο απεικόνισης: tcp and ip.addr == 147.102.40.1

1.12: Κάνει πάλι 5 προσπάθειες

1.13: Διαφορές σε σχέση με τις περιπτώσεις A, B:

1 Υπάρχει απάντηση από τον 147.102.40.1 η οποία όμως έχει ενεργοποιημένη την σημαία της απόρριψης (Rst)

2 Κάθε επαναπροσπάθεια γίνεται 0.5 sec μετά την λήψη της απάντησης – απόρριψης

1.14: Περιλαμβάνει τις σημαίες Acknowledgement και Reset

1.15: Η Reset δηλώνει την άρνηση της εγκατάστασης σύνδεσης TCP

1.16: Header Length = 20 bytes, Data = 0 bytes

1.17: Στον παρακάτω πίνακα φαίνονται τα πεδία και τα μεγέθη τους σε bit ή byte

<table>
<tr><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="16">Source Port</td><td colspan="16">Destination Port</td></tr>
<tr><td colspan="32">Sequence Number</td></tr>
<tr><td colspan="32">Acknowledgement Number</td></tr>
<tr><td colspan="4">Header Length</td><td colspan="12">Flags</td><td colspan="16">Window</td></tr>
<tr><td colspan="16">Checksum</td><td colspan="16">Urgent Pointer</td></tr>
</table>

Το πεδίο Flags περιέχει τα εξής πεδία:

- Reserved = 3 bit
- (N) Accurate ECN = 1 bit
- (C) Congestion Window Reduced = 1 bit
- (E) ECN – Echo = 1 bit
- (U) URG = 1 bit
- (A) ACK = 1 bit
- (P) PSH = 1 bit
- (R) RST = 1 bit
- (S) SYN = 1 bit
- (F) FIN = 1 bit

Το πεδίο Flags αναπτύσσεται όπως παρακάτω:

<table>
<tr><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="4">Reserved</td><td>N</td><td>C</td><td>E</td><td>U</td><td>A</td><td>P</td><td>R</td><td>S</td><td>F</td></tr>
</table>

1.18: Με βάση το Network Sorcery το πεδίο αυτό λέγεται Data Offset, στο Wireshark λέγεται Header Length

1.19: Το πεδίο Header Length αναπαριστά τον αριθμό των 32 – bit λέξεων στο TCP header. Το μήκος της επικεφαλίδας σε byte προκύπτει ως εξής:

(Header Length) * 32 / 8 = (Header Length) * 4

1.20: Όχι, δεν υπάρχει τέτοιο πεδιο

1.21: Η επικεφαλίδα IPv4 έχει τα πεδία Header Length και Total Length, ενώ η επικεφαλίδα TCP έχει το πεδίο Header Length. Συνεπώς, μπορούμε να πούμε ότι

Data(TCP) = Total Length(IPv4) – Header Length(IPv4) – Header Length(TCP)

1.22: Header Length = 32 bytes

1.23: Η διαφορά οφείλεται στην ύπαρξη του πεδίου Options που είναι 12 bytes στο τεμάχιο TCP που στέλνει ο υπολογιστής μου

2 Εγκατάσταση σύνδεσης, μεταφορά δεδομένων και απόλυση σύνδεσης TCP

2.1: Φίλτρο σύλληψης: host edu-dy.cn.ntua.gr and tcp

Εγκατάσταση σύνδεσης

2.2: Ο υπολογιστής μου προσπαθεί να συνδεθεί στη θύρα 21

2.3: Για την μεταφορά των δεδομένων συνδεόμαστε με τη θύρα 20

2.4: Φίλτρο απεικόνισης: tcp.port == 21

2.5: Ανταλλάσσονται 3 τεμάχια TCP για την εγκατάσταση της σύνδεσης

2.6: Χρησιμοποιούνται 2 σημαίες. Οι Syn και Ack

2.7: Syn από εμάς = 32 bytes
Syn/Ack απάντηση σε εμάς = 32 bytes
Ack απάντηση από εμάς = 20 bytes

2.8: Data = 0 bytes και στα 3 τεμάχια

2.9: Διάρκεια τριπλής χειραψίας = 12.235ms

2.10: Ναι

2.11: Sequence Number (αιτήματός μας) = 0 (relative)
Sequence Number (απάντησης) = 0 (relative)

2.12: Το Acknowledgement Number είναι 1. Αυτό δηλώνει ότι ο εξυπηρετητής FTP περιμένει να δεχτεί πακέτο από εμάς με Sequence Number = 1. Είναι ο αύξων αριθμός του επόμενου byte στα δεδομένα που αναμένει ο εξυπηρετητής FTP να του στείλουμε. Γίνεται 1 γιατί το πρώτο byte δεδομένων που περιμένει να λάβει έχει αύξων αριθμό (Initial Sequence Number + 1) = 0 + 1 = 1

2.13: Sequence Number — Ο αύξων αριθμός του πρώτου byte δεδομένων που στέλνουμε

Acknowledgement Number — Ο αύξων αριθμός του πρώτου byte δεδομένων που περιμένουμε να λάβουμε

2.14: Το μήκος δεδομένων τους είναι 0

2.15: max{Seq/Ack Number} = 2^32 – 1 = 4,294,967,295

2.16: Φίλτρο απεικόνισης: tcp.len == 0 and tcp.port == 21 and (tcp.flags.syn == 1 or tcp.flags.ack == 1 or (tcp.flags.syn == 1 and tcp.flags.ack == 1)) and (tcp.seq == 0 or tcp.seq == 1)

2.17: Ο υπολογιστής μου ανακοινώνει Window = 8192

2.18: Ο εξυπηρετητής ανακοινώνει Window = 65535

2.19: Στο πεδίο Window

2.20: Εμείς — Window Scale = 0 (επί 1)
Εξυπηρετητής — Window Scale = 6 (επί 64)

2.21: Στα Options υπάρχει το πεδίο Windows Scale που είναι το σχετικό πεδίο

2.22: MSS που ανακοινώνει ο υπολογιστής μου = 1460 bytes

2.23: MSS = MTU – 40 = 1500 – 40 = 1460 bytes

2.24: Στα Options υπάρχει το σχετικό πεδίο Maximum segment size

2.25: MSS (edu-dy.cn.ntua.gr) = 536 bytes

2.26: MTU = MSS + 40 = 536 + 40 = 576 bytes

2.27: Το μεγαλύτερο TCP τεμάχιο (μαζί με την επικεφαλίδα) είναι TCP header + MSS = 20 + 536 = 556. Τα δεδομένα, προφανώς, είναι 553 bytes

Απόλυση σύνδεσης

2.28: Ενεργοποιείται η σημαία Fin

2.29: Φίλτρο απεικόνισης: tcp.port == 21 and tcp.flags.fin == 1

2.30: Ο εξυπηρετητής εκκινεί την διαδικασία απόλυσης

2.31: Ανταλλάσσονται 2 τεμάχια για την απόλυση της σύνδεσης στο τέλος

2.32: Header Length = 20 bytes

2.33: Στα πακέτα λήξης της σύνδεσης στο τέλος Data = 0 bytes

2.34: Για το IPv4 πακέτο αυτό έχουμε Total Length = 40 bytes. Είναι 20 bytes το IPv4 Header και άλλα 20 bytes το TCP Header

2.35: Το ίδιο με πριν

2.36: Ο υπολογιστής μου έστειλε 114 bytes και έλαβε 375 bytes

2.37: Ξέρουμε από τα προηγούμενα ότι η αρίθμηση των Data byte ξεκινάει από το 1. Δηλαδή, το πρώτο byte που θα στείλουμε είναι το 1. Επομένως, τα bytes που μεταδόθηκαν από κάθε πλευρά θα είναι τα Sequence Numbers κάθε πλευράς.

Μεταφορά σύνδεσης

2.38: Φίλτρο απεικόνισης: tcp.port == 20

2.39:

	MSS (bytes)
147.102.40.15	536
Ο υπολογιστής μου	1460

2.40: Είναι TCP Header + minimum MSS = 20 + 536 = 556 bytes
Αν μας νοιάζει το MSS μόνο τότε, προφανώς, είναι το 536

2.41: RTT = 11.884ms

2.42: Όχι

2.43: Ο εξυπηρετητής έστειλε 118 τεμάχια με δεδομένα

2.44: Έστειλε 50 πακέτα ACK για τα πακέτα που έλαβε

2.45: Ανακοινώνει 8193 (Calculated window size: 2097408)

2.46: Όχι, πιθανώς έχει διαφορετικό διαθέσιμο bandwidth και το γνωστοποιεί αυτό στον εξυπηρετητή

2.47: Το window size δεν μεταβάλλεται

2.48: Σημαίνει ότι ο buffer της υποδοχής δεδομένων είναι γεμάτος και δεν μπορεί να δεχτεί νέα δεδομένα. Ο εξυπηρετητής ή θα περίμενε ή θα έστελνε πακέτο με την σημαία Psh ενεργοποιημένη για να εξωθήσει τον υπολογιστή μου να αδειάσει τον χώρο αποθήκευσης και να διαβάσει τα δεδομένα

2.49:

	Header (bytes)
Ethernet	14
IPv4	20
TCP	32

Το πλαίσιο έχει μέγεθος 590 bytes

2.50: Τα δεδομένα είναι 524 bytes. Ναι, γιατί η τιμή 536 bytes που είχαμε βρει προηγουμένως είναι για τα δεδομένα αν η επικεφαλίδα έχει την μικρότερη δυνατή τιμή (20 bytes). Στη περίπτωση αυτή η επικεφαλίδα TCP είναι 32 bytes άρα παίρνει 12 bytes από το MSS

2.51: Με την προϋπόθεση ότι επιτρέπεται το fragmentation θα είχαμε θρυμματισμό των δεδομένων και θα στέλνονταν σε ξεχωριστά IP πακέτα

2.52:

	Data (bytes)
Εγώ	0
Εξυπηρετητής	61440

Για να βρούμε τα δεδομένα (και στο ερώτημα 2.36) χρησιμοποιούμε τα Sequence Numbers όπως εξηγήσαμε αλλά πρέπει να αφαιρέσουμε το πλήθος των αυξόντων αριθμών byte που δεν αντιστοιχούν σε πραγματικά byte δεδομένων που στέλνονται αλλά υπάρχουν στα πακέτα σύνδεσης/λήξης της σύνδεσης

2.53: Ρυθμός μεταφοράς δεδομένων (εξυπηρετητής → εγώ) = 568.89 kbyte/sec

2.54: Δεν υπήρξαν αναμεταδόσεις πακέτων

Αποφυγή συμφόρησης στο TCP

3.1: Φίλτρο απεικόνισης: tcp.port == 20

3.2: IPv4 (υπολογιστής που κατέβασε τα δεδομένα) = 94.65.141.44

3.3: RTT = 14.674ms. Είναι λίγο μεγαλύτερη

3.4: Το πλήθος των πακέτων που στέλνονται ανά RTT αυξάνονται εκθετικά

3.5: Έστειλε 4 τεμάχια. Στο πρότυπο βλέπουμε ότι για SMSS < 1095 bytes το παράθυρο δεν μπορεί να είναι μεγαλύτερο από 4 τεμάχια. Αυτό συμφωνεί με ό,τι βλέπουμε στο Wireshark

3.6:

RTT	Τεμάχια
2ο	6
3ο	10
4ο	16

3.7:

RTT	ACK
2ο	1
3ο	2
4ο	3

3.8:

RTT	Τεμάχια
2ο	10
3ο	11
4ο	12

Στην παράγραφο 2 του RFC 6928 βλέπουμε ότι υπάρχει και η ιδέα το Initial Window να είναι 10 * MSS, το οποίο συμφωνεί με την δική μου καταγραφή. Στην δική μου καταγραφή η αύξηση είναι γραμμική, στην έτοιμη καταγραφή η αύξηση ακολουθεί αυτή που περιγράφεται στο RFC 5681.

4 Μετάδοση δεδομένων με UDP

4.1: Φίλτρο σύλληψης: udp

4.2:

Field	Size (bytes)
Source Port	2
Destination Port	2
Length	2
Checksum	2

4.3: Header Length = 8 bytes

Στην καταγραφή μου ενθυλακώνονται σε IPv6 πακέτα

4.4: 90 bytes (Payload) + 8 bytes (UDP Header)

4.5: Το πεδίο Length της επικεφαλίδας UDP εκφράζει το συνολικό μήκος του δεδομενογράμματος UDP (για το παραπάνω πακέτο είναι 98)

4.6: min{Length} = 8

4.7: min = 28 bytes (τα 2 headers χωρίς data) και max = 65507 bytes (65535 bytes είναι το μέγιστο για το IPv4 και αφαιρούμε τα 2 headers (20 + 8 bytes))

4.8: 576 – 20 – 8 = 548 bytes (payload data στο UDP δεδομενόγραμμα)

4.9: Όχι

4.10: Φίλτρο απεικόνισης: dns

4.11: IPv6 (απάντησης) = fe80::1 (είναι ο DNS server μου)

	4.12	4.13
προέλευση	50117	53
προορισμός	53	50117

4.14: Η θύρα που αντιστοιχεί στο πρωτόκολλο εφαρμογής DNS είναι η 53