

Εθνικό Μετσόβιο Πολυτεχνείο
Δίκτυα Υπολογιστών
Χειμερινό Εξάμηνο 2022-2023

Όνομα:

Ιωάννης Μπασδέκης - 03119198



7η Εργαστηριακή Αναφορά

Μέρος 1ο

1.1) host 147.102.131.131

1.2) ip.addr == 1.1.1.1 or ip.addr == 2.2.2.2 or ip.addr == 147.102.40.1

1.3) port = 23 . Είναι η θύρα για telnet

1.4) tcp.port == 23

1.5) SYN flag

1.6) 11 σε κάθε περίπτωση (μία αρχική και 10 tcp retransmission)

1.7) 1 sec, 2 sec, 4 sec, 8 sec, 16 sec, 32 sec.

1.8) Το μόνο που αλλάζει είναι το Sequence Number(raw)
(προφανώς αλλάζει και το checksum αφού εξαρτάται από τα
στοιχεία του header)

1.9) Μόνο το πρώτο βήμα [SYN] , όπου seq = 0, ack = 0.

1.10) Σταματάει την προσπάθεια αφού δεν έχουμε κάποιο TCP
μήνυμα με FIN flag για την απόλυση της σύνδεσης

1.11) tcp and ip.addr == 147.102.40.1

1.12) Μία

1.13) Σε αυτή την περίπτωση λαμβάνουμε TCP μήνυμα με [RST,ACK] flag που σημαίνει ότι ο προορισμός μας έλαβε και απέρριψε την σύνδεση

1.14) [RST,ACK]

1.15) RST

1.16) TCP header length = 20 bytes , data = 0 bytes

1.17) Source Port : 16 bits

Destination Port : 16 bits

Sequence Number : 32 bits

Acknowledgment Number : 32 bits

Header Length : 4 bits

Flags : 12 bits

Windows : 16 bits

Checksum : 16 bits

Urgent Pointer : 16 bits

1.18) Το πεδίο αυτό ονομάζεται DO (Data Offset) αλλά στο wireshark αναγράφεται ως Header Length

1.19) Το header length (η HEX τιμή) δείχνει από πόσες τετράδες από bytes (δηλαδή για κάθε 32 bit +1 header length) αποτελείται το header. Στην προκειμένη περίπτωση header length =5 . Άρα $5 \times 4 = 20$ bytes

1.20) Όχι

1.21) Από το πεδίο Total Length του IP header

1.22) 44 bytes

1.23) Υπάρχει διαφορά η οποία οφείλεται στο πεδίο Options που έχει το μήνυμα που στείλαμε εμείς και περιέχει τις ρυθμίσεις σύνδεσης

Μέρος 2ο

2.1) tcp and host 147.102.40.15

2.2) Στην θύρα 21 που είναι η πασίγνωστη για FTP

2.3) Στην θύρα 52283 (είμαι σε Mac και χρησιμοποιώ ncftp αντί για ftp και δεν χρησιμοποιείται η θύρα 20 για το transfer)

2.4) tcp.port == 21

2.5) 3

2.6) SYN , ACK

2.7) 1ο τεμάχιο = 44 bytes , 2ο = 40 bytes , 3ο = 32 bytes

2.8) 0 bytes

2.9) 0.005212

2.10) Ναι

2.11) Το πρώτο μήνυμα(που στο στέλνουμε εμείς) : απόλυτος = 3684828693. σχετικός = 0

Για το δεύτερο (το πρώτο που λαμβάνουμε) : απόλυτος = 3492578101. σχετικός = 0

2.12) Αφού το πρώτο μήνυμα που του στείλαμε έχει seq = 0 σημαίνει ότι θα του στείλουμε μέχρι και το 0 byte (στην ουσία δεν στέκεται κάτι) . Ο ftp εξυπηρετητής το λαμβάνει και απαντάει ότι έλαβε μέχρι και το 0 byte και περιμένει το επόμενο δηλαδή το 1 . Άρα για αυτό το ACK = 1 στη μήνυμα αποδοχής

2.13) Λειτουργεί με τον ίδιο τρόπο. Στο δεύτερο μήνυμα ο εξυπηρετητής περιμένει να του στείλουμε από το 1 byte και μετά οπότε για αυτό seq = 1 στο τρίτο μήνυμα. Για το ack = 1 είναι η ίδια περίπτωση με το 2.12. Αφού μας στέλνει μέχρι και το 0 byte(στην ουσία τίποτα) με το ack = 1 του απαντάμε ότι λάβαμε μέχρι το 0 byte και περιμένουμε από το 1 και μετά.

2.14) 0 bytes

2.15) Αφού είναι 32bit η μέγιστη τιμή τους είναι $2^{32} - 1$

2.16) $(\text{tcp.ack} == 0 \text{ or } \text{tcp.ack} == 1) \text{ and } (\text{tcp.seq} == 0 \text{ or } \text{tcp.seq} == 1) \text{ and } \text{tcp.len} == 0 \text{ and } \text{tcp.port} == 21$

2.17) Και στις δύο περιπτώσεις (σύνδεση ελέγχου , μεταφορά δεδομένων) window = 65535

2.18) 65535

2.19) Window

2.20) 6

2.21) Στο πεδίο Options και συγκεκριμένα στο 4ο Option (τιμή Kind = 3) στο πεδίο shift count

2.22) 1460 bytes

2.23) $\text{MSS} = \text{MTU} - 40$

2.24) Στο πεδίο Options και συγκεκριμένα στο 1ο Option (τιμή Kind = 2) στο πεδίο MSS value

2.25) 536

2.26) $\text{MSS} = \text{MTU} - 40 \Rightarrow \text{MSS} = 576 - 40 = 536$

2.27) 536 bytes δεδομένα + 20 bytes TCP header αρα 556 μέγεθος

2.28) FIN flag

2.29) $\text{tcp.port} == 21 \text{ and } \text{tcp.flags.fin} == 1$

2.30) Ο edu-dy.cn.ntua.gr

2.31) 4 (1 FIN,ACK και 1 ACK από τον server και 1 FIN,ACK και 1 ACK από εμάς)

2.32) 32 bytes

2.33) 0 bytes

2.34) 52 bytes = 20 bytes IP header + 32 bytes TCP header

2.35) 52 bytes = 20 bytes IP header + 32 bytes TCP header

2.36) Από τον υπολογιστή μου στον server : 204

Από τον server στον υπολογιστή μου : 1064

2.37) Από το μήνυμα FIN,ACK που έστειλε ο υπολογιστής μας το seq -1 είναι τα bytes που στείλαμε και το ack -1 είναι τα bytes που λάβαμε

2.38) `tcp.port == 52283`

2.39) Ο υπολογιστής μας : 1460 bytes

server: 536 bytes

2.40) 536 data bytes + 20 bytes tcp header αρα συνολικά 556 bytes

2.41) 0.147229 sec

2.42) Όχι

2.43) 118 πακέτα (`tcp.port == 57805 and ip.src == 147.102.40.15 and !tcp.len == 0`)

2.44) 99 πακέτα (`tcp.port == 57805 and ip.dst == 147.102.40.15 and tcp.flags.ack == 1`)

2.45) 2046

2.46) Όχι δεν είναι. Προκύπτει από το την τιμή window πολλαπλασιασμένη με το scale (τιμή scale = 6 , πολ/ζουμε το window με το 2^6)

2.47) Αλλάζει. Μικρότερη τιμή = 1031

2.48) Θα σήμαινε ότι δεν έχει χώρο ο buffer και δεν μπορεί να δεχτεί άλλα δεδομένα ο υπολογιστής μας μέχρι να γίνουν acknowledged.

2.49) Frame = 590 bytes , Ethernet Header = 14 bytes , IP header = 20 bytes , TCP header = 32 bytes

2.50) Ναι

2.51) Θα γίνει θρυμματισμός αλλά με την προϋπόθεση ότι τα θραύσματα δεν θα ξεπερνούν το μέγεθος που έχει οριστεί από την τριπλή χειραψία.

2.52) 61308 bytes απο τον σέρβερ στον υπολογιστή μας και 0 από τον υπολογιστή μας στον σέρβερ

2.53) 1181 kbytes / sec

2.54) Όχι δεν υπήρχαν

Μέρος 3ο

3.1) `tcp.port == 20`

3.2) 94.65.141.44

3.3) `RTT = 0.014626000 sec` . Είναι μικρότερο και σχεδόν υποδεκαπλάσιο αυτού που βρήκα στο προηγούμενο ερώτημα

3.4) Όσο περνάει ο χρόνος ο server στέλνει όλο και περισσότερα segments. Αυτό ουσιαστικά κάνει ο αλγόριθμος slow start

3.5) 4 Packets και δεν ξεπερνάει την προδιαγραφή (max 4)

3.6) 6 , 10 , 16 πακέτα αντίστοιχα

3.7) Στο πρώτο RTT στέλνεται ένα ACK, στο 2ο δύο ACK , στο τρίτο 3 ACK. Παρατηρώ ότι δεν στέλνεται ένα ACK για κάθε πακέτο που στέλνει ο σέρβερ αλλά περιμένει να συσσωρευτούν τα segment και να στείλει λιγότερα ACK

3.8) Είναι παρόμοια αλλά στην καταγραφή που κάναμε εμείς στέλνονται περισσότερα πακέτα ανά RTT

Μέρος 4ο

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) Για IPv4 : 35 bytes

4.5) Εκφράζει το μήκος του UDP header μαζί με το UDP payload

4.6) Το ελάχιστο είναι 8 bytes γιατί μπορεί το payload να είναι 0 άρα θα δείχνει μόνο το header length

4.7) ελάχιστο = 0 bytes (βλέπε 4.6 ερώτημα) και μέγιστο 65535 (max ip packet size) - 20 bytes για το IP header - 8 udp header = 65507 bytes

4.8) $576 - 20 - 8 = 548$ bytes (για τον ίδιο λόγω με 4.7)

4.9) QUIC , SSDP , MDNS

4.10) dns

4.11) IPv4 1.1.1.1 (custom dns ip)

4.12) Source Port : 53971 Destination port : 53

4.13) Destination Port : 53971 Source port : 53

4.14) 53

