ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 8: ΠΡΩΤΟΚΟΛΛΑ TCP ΚΑΙ UDP

6 ΔΕΚΕΜΒΡΙΟΥ, 2022

ΘΟΔΩΡΗΣ ΑΡΑΠΗΣ – EL18028

Ονοματεπώνυμο: Θοδωρής Αράπης	**Ομάδα:** 2
Όνομα PC/ΛΣ: pc-a37/ WINDOWS 95	**Ημερομηνία:** 6/12/2022
Διεύθυνση IP: 147.102.38.97 (Άσκηση 1 και 3) 147.102.38.96 (Άσκηση 2)	**Διεύθυνση MAC:** 00:11:25:F8:F9:0C

Άσκηση 1: TELNET

1.1

Το πρωτόκολλο εφαρμογής TELNET χρησιμοποιεί το πρωτόκολλο μεταφοράς TCP.

1.2

Χρησιμοποιούνται τα ports 1149 και 23.

1.3

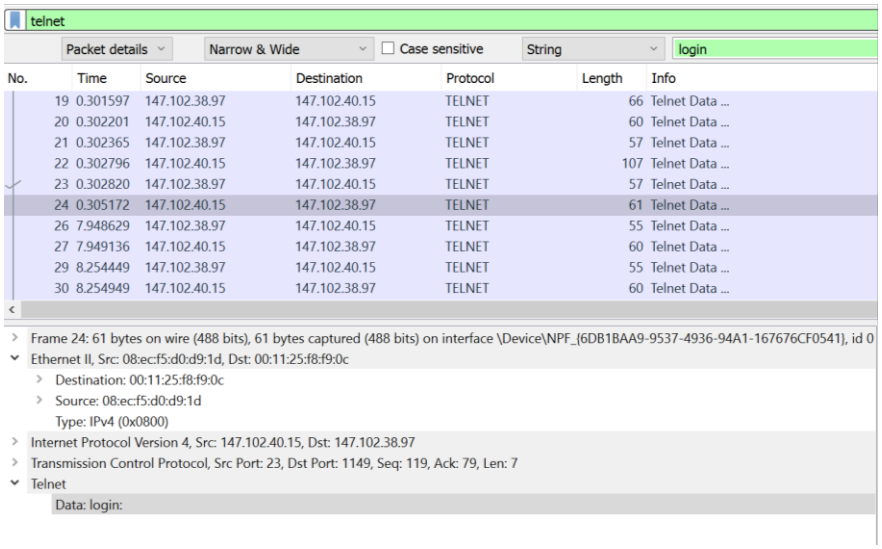
Η θύρα 23 αντιστοιχεί στο πρωτόκολλο TELNET.

1.4

Το φίλτρο απεικόνισης είναι το εξής: «telnet»

1.5

Ακολουθούμε την διαδικασία που υποδεικνύεται και βρίσκουμε το πακέτο με αριθμό 24:



Αναζητούμε τώρα εντολές Telnet τύπου echo στα πακέτα που προηγούνται του πακέτου 24 και βρίσκουμε:

- **Τεμάχιο 16:** Εντολή ***Do Echo*** από 147.102.40.15 προς 147.102.38.97
- **Τεμάχιο 19:** Εντολή ***Will Echo*** από 147.102.38.97 προς 147.102.40.15
- **Τεμάχιο 20:** Εντολή ***Don't Echo*** από 147.102.40.15 προς 147.102.38.97
- **Τεμάχιο 20:** Εντολή ***Will Echo*** από 147.102.40.15 προς 147.102.38.97

- **Τεμάχιο 21:** Εντολή ***Won't Echo*** από 147.102.38.97 προς 147.102.40.15
- **Τεμάχιο 22:** Εντολή ***Do Echo*** από 147.102.38.97 προς 147.102.40.15

1.6

Ναι, ο edu-dy.cn.ntua.gr ζητάει από τον υπολογιστή μας να επαναλαμβάνει τους χαρακτήρες που λαμβάνει (τεμάχιο 16: Do Echo) και ο υπολογιστής μας δέχεται (τεμάχιο 19: Will Echo).

1.7

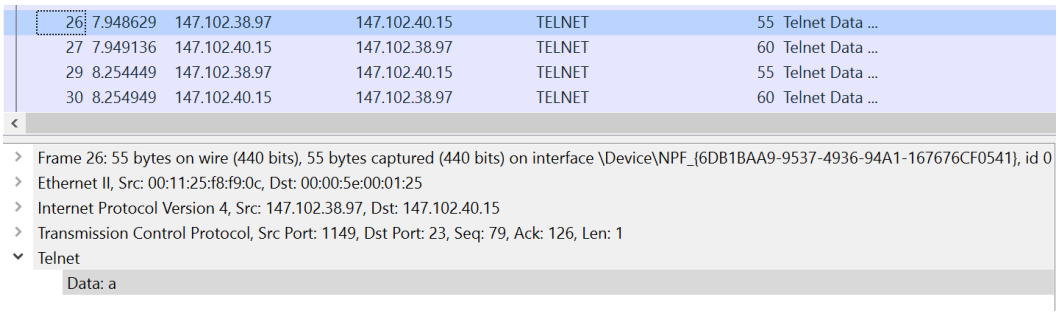
Ναι, ο edu-dy.cn.ntua.gr ζητάει από τον υπολογιστή μας να μην επαναλαμβάνει τους χαρακτήρες που λαμβάνει (τεμάχιο 20: Don't Echo) και ο υπολογιστής μας δέχεται (τεμάχιο 21: Won't Echo).

1.8

Ο edu-dy.cn.ntua.gr προτίθεται να επαναλαμβάνει τους χαρακτήρες που λαμβάνει από τον υπολογιστή μας (τεμάχιο 20: Will Echo).

1.9

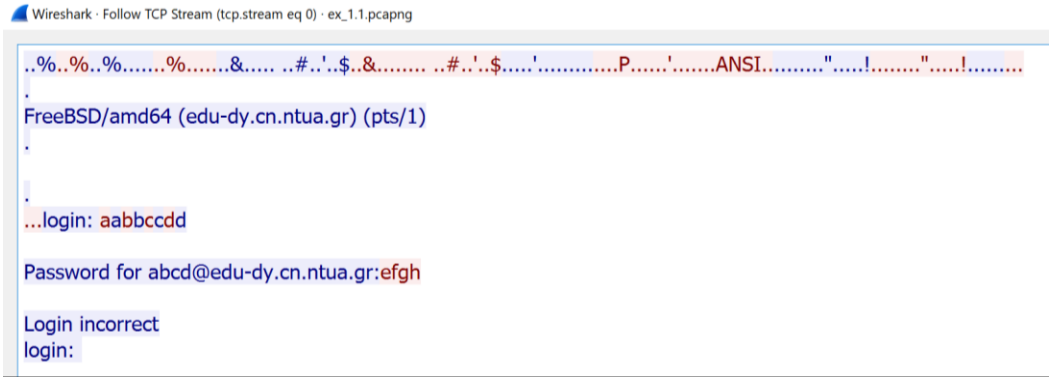
Αναζητούμε μεταξύ των τεμαχίων που έχουν ως πηγή τον υπολογιστή μας και με αύξοντα αριθμό μεγαλύτερο του 23. Βρίσκουμε το ζητούμενο στο πακέτο 26:



Προηγουμένως (ερώτημα 1.5), ο υπολογιστής μας έχει ζητήσει την επανάληψη των χαρακτήρων από τον edu-dy.cn.ntua.gr (τεμάχιο 22: Do Echo).

1.10

Η ροή κίνησης TCP είναι η εξής:Μετά την προτροπή login



Μετά την προτροπή login παρατηρούμε, αρχικά, την εισαγωγή του χαρακτήρα ‘a’ (κόκκινο χρώμα) εκ μέρους μας (τεμάχιο 26) και την εμφάνισή του επίσης στον σέρβερ (μπλε χρώμα). Το ίδιο συμβαίνει και για τους υπόλοιπους χαρακτήρες που εισάγουμε κατά το login (b, c και d), δηλαδή τους πληκτρολογούμε και αυτοί εμφανίζονται επίσης στον edu-dy.cn.ntua.gr.

1.11

Όσα παρατηρήσαμε, δικαιολογούνται, καθώς όπως είδαμε νωρίτερα, ο edu-dy.cn.ntua.gr προτίθεται να επαναλαμβάνει (τεμάχιο 20) τους χαρακτήρες που του στέλνουμε και επιπλέον ο δικός μας υπολογιστής του έχει ζητήσει να το κάνει (τεμάχιο 22).

1.12

Εφαρμόζουμε το φίλτρο απεικόνισης: «ip.src==147.102.38.97 and ip.dst==147.102.40.15 and telnet»

1.13

Χρειάζονται 4 πακέτα (υπ’ αριθμόν 26, 29, 32, 35), ένα για κάθε χαρακτήρα.

1.14

Επίσης, για τον κωδικό efgh χρειάζονται επίσης 4 πακέτα (43, 45, 47, 49) .

1.15

Όχι, ο εξυπηρετητής δε στέλνει την ηχώ των χαρακτήρων efgh του κωδικού χρήστη προς τον πελάτη.

1.16

Ενώ πριν την εισαγωγή των χαρακτήρων για το login, βλέπουμε πως ο υπολογιστής μας στέλνει Do Echo (τεμάχιο 22), δε παρατηρούμε κάποια εντολή Don't Echo πριν τη μεταφορά του κωδικού.

1.17

Υπάρχει περίπτωση ένα κακόβουλο λογισμικό (ή ακόμη και κάποιος άνθρωπος) να μπορεί να διαβάσει την οθόνη όσο εισάγεται ο κωδικός και να αποκτήσει πρόσβαση ενώ δε θα έπρεπε.

1.18

Το Telnet υστερεί από άποψη ασφαλείας, καθώς αρκεί κάποιος να μπορεί να "ακούει" την επικοινωνία μεταξύ 2 κόμβων για να υποκλέψει ευαίσθητα δεδομένα. Συγκεκριμένα, εφόσον η επικοινωνία δεν είναι κρυπτογραφημένη, με έναν αναλυτή πακέτων όπως το Wireshark και όπως είδαμε, είναι εύκολο να αναγνωστούν τα δεδομένα αυτά.

Άσκηση 2: FTP

2.1

Χρησιμοποιήσαμε το φίλτρο σύλληψης: «host edu-dy.cn.ntua.gr»

2.2

Το όρισμα -d ενεργοποιεί την αποσφαλμάτωση (enables debugging).

2.3

Το FTP πρωτόκολλο εφαρμογής χρησιμοποιεί το TCP πρωτόκολλο μεταφοράς.

2.4

Οι ζητούμενες θύρες πηγής και προορισμού είναι:

Αριθμός πακέτου	Πηγής	Προορισμού
1	1155	21
2	21	1155
28	20	5001
29	5001	20

Από τα παραπάνω γνωρίζουμε πως η θύρα 21 χρησιμοποιείται για τις εντολές ελέγχου, ενώ η θύρα 20 για τις εντολές δεδομένων (για ενεργό FTP τρόπο λειτουργίας).

2.5

Η TCP σύνδεση για τη μεταφορά δεδομένων γίνεται από τον εξυπηρετητή προς τον πελάτη.

2.6

Στάλθηκαν οι εξής εντολές FTP από τον πελάτη:

- **Τεμάχιο 6**: Εντολή ***USER anonymous***
- **Τεμάχιο 9**: Εντολή ***PASS labuser@cn***
- **Τεμάχιο 12**: Εντολή ***HELP***
- **Τεμάχιο 25**: Εντολή ***PORT 147,102,38,96,19,137***
- **Τεμάχιο 27**: Εντολή ***NLST***
- **Τεμάχιο 38**: Εντολή ***QUIT***

2.7

Όπως μπορούμε να δούμε για παράδειγμα παρακάτω, οι εντολές αυτές εμφανίζονται στις πληροφορίες αποσφαλμάτωσης στην οθόνη του προγράμματος φλοιού ftp με ένα βέλος μπροστά τους:

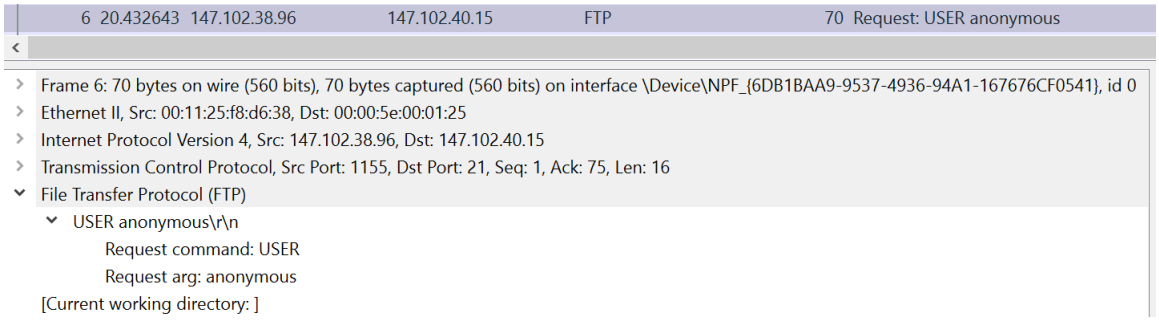
```
User (edu-dy.cn.ece.ntua.gr:(none)): anonymous
---> USER anonymous
331 Anonymous login ok, send your complete email address as your password
Password:
---> PASS labuser@cn
230 Anonymous access granted, restrictions apply
ftp> help
Commands may be abbreviated.  Commands are:

!               delete          literal         prompt          send
?               debug           ls              put             status
append          dir             mdelete         pwd             trace
ascii           disconnect      mdir            quit            type
bell            get             mget            quote           user
binary          glob            mkdir           recv            verbose
bye             hash            mls             remotehelp
cd              help            mput            rename
close           lcd             open            rmdir
ftp> remotehelp
---> HELP
214-The following commands are recognized (* =>'s unimplemented):
```

2.8

Με την εντολή ***USER***.

2.9

Απαιτείται ένα πακέτο (Αυτό με αριθμό 6 συγκεκριμένα, όπως φαίνεται παρακάτω).



2.10

Με την εντολή ***PASS***.

2.11

Χρειάζεται επίσης ένα μόνο IPv4 πακέτο για να μεταφερθεί ο κωδικός.

2.12

Αναφορικά με τη μεταφορά ονόματος/κωδικού με τα πρωτόκολλα TELNET και FTP παρατηρούμε πως ενώ το πρώτο απαιτεί ένα τεμάχιο για κάθε χαρακτήρα του ονόματος/κωδικού, το ftp στέλνει ολόκληρο το όνομα/κωδικό σε ένα πακέτο. Αυτό που έχουν κοινό είναι πως και στο FTP αλλά και στο TELNET όπως είδαμε πριν, οι πληροφορίες αυτές δε στέλνονται κρυπτογραφημένες.
Επιπλέον

2.13

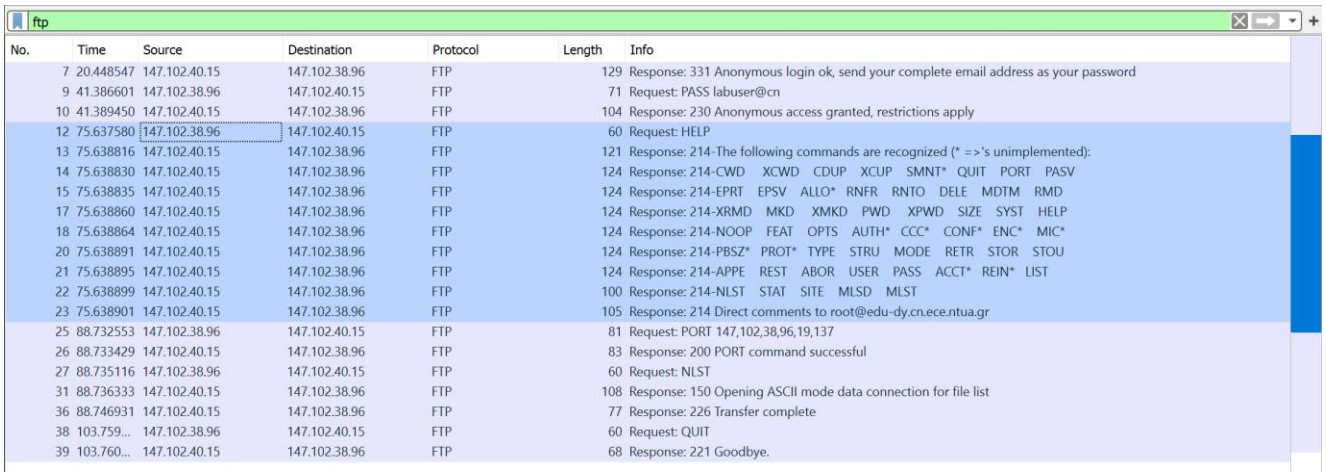
Όπως παρατηρούμε από το screenshot του ερωτήματος 2.7, η εντολή help του προγράμματος φλοιού δε μεταφράζεται σε εντολή του πρωτοκόλλου FTP, αφού δεν εκτυπώνεται στο τερματικό μήνυμα από τον debugger. Ωστόσο, αυτή που μεταφράζεται είναι η εντολή remotehelp, η οποία και μεταφράζεται στην εντολή HELP.

2.14

Δύο εντολές FTP που δεν υποστηρίζονται από τον FTP εξυπηρετητή είναι η PORT και η PBSZ.

2.15

Όπως βλέπουμε, ο υπολογιστής μας έστειλε 1 (πακέτο 12), ενώ ο εξυπηρετητής 9 πακέτα (13 έως 23) σχετικά με την εντολή remotehelp.



ftp

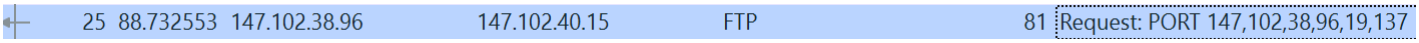
No.	Time	Source	Destination	Protocol	Length	Info
7	20.448547	147.102.40.15	147.102.38.96	FTP	129	Response: 331 Anonymous login ok, send your complete email address as your password
9	41.386601	147.102.38.96	147.102.40.15	FTP	71	Request: PASS labuser@cn
10	41.389450	147.102.40.15	147.102.38.96	FTP	104	Response: 230 Anonymous access granted, restrictions apply
12	75.637580	147.102.38.96	147.102.40.15	FTP	60	Request: HELP
13	75.638816	147.102.40.15	147.102.38.96	FTP	121	Response: 214-The following commands are recognized (* =>'s unimplemented):
14	75.638830	147.102.40.15	147.102.38.96	FTP	124	Response: 214-CWD XCWD CDUP XCUP SMNT* QUIT PORT PASV
15	75.638835	147.102.40.15	147.102.38.96	FTP	124	Response: 214-EPRT EPSV ALLO* RNFR RNTO DELE MDTM RMD
17	75.638860	147.102.40.15	147.102.38.96	FTP	124	Response: 214-XRMD MKD XMKD PWD XPWD SIZE SYST HELP
18	75.638864	147.102.40.15	147.102.38.96	FTP	124	Response: 214-NOOP FEAT OPTS AUTH* CCC* CONF* ENC* MIC*
20	75.638891	147.102.40.15	147.102.38.96	FTP	124	Response: 214-PBSZ* PROT* TYPE STRU MODE RETR STOR STOU
21	75.638895	147.102.40.15	147.102.38.96	FTP	124	Response: 214-APPE REST ABOR USER PASS ACCT* REIN* LIST
22	75.638899	147.102.40.15	147.102.38.96	FTP	100	Response: 214-NLST STAT SITE MLSD MLST
23	75.638901	147.102.40.15	147.102.38.96	FTP	105	Response: 214 Direct comments to root@edu-dy.cn.ece.ntua.gr
25	88.732553	147.102.38.96	147.102.40.15	FTP	81	Request: PORT 147,102,38,96,19,137
26	88.733429	147.102.40.15	147.102.38.96	FTP	83	Response: 200 PORT command successful
27	88.735116	147.102.38.96	147.102.40.15	FTP	60	Request: NLST
31	88.736333	147.102.40.15	147.102.38.96	FTP	108	Response: 150 Opening ASCII mode data connection for file list
36	88.746931	147.102.40.15	147.102.38.96	FTP	77	Response: 226 Transfer complete
38	103.759...	147.102.38.96	147.102.40.15	FTP	60	Request: QUIT
39	103.760...	147.102.40.15	147.102.38.96	FTP	68	Response: 221 Goodbye.

2.16

Βλέποντας το παραπάνω στιγμιότυπο, το πρώτο μήνυμα (πακέτο 15) από τον εξυπηρετητή περιλαμβάνει το μήνυμα “214-The following commands are recognized…”. Ο εξυπηρετητής, δηλώνει πως τελείωσε η αποστολή πακέτων στέλνοντας ένα πακέτο, το μήνυμα του οποίου ξεκινάει με τον ίδιο κωδικό (214 εν προκειμένω), ακολουθείται από κενό και έχει ενδεχομένως κάποιο κείμενο, όπως και επαληθεύεται παραπάνω (πακέτο 23).

2.17

Περιγράφουν την IP του υπολογιστή μας.



25	88.732553	147.102.38.96	147.102.40.15	FTP	81	Request: PORT 147,102,38,96,19,137

2.18

Στο ερώτημα 2.4 βρήκαμε πως ο υπολογιστής μας δέχεται δεδομένα στη θύρα 5001. Αυτό, προκύπτει από τους τελευταίος δεκαδικούς αριθμούς ως εξής: Πολλαπλασιάζουμε τον πρώτο από τους 2 με 256 και προσθέτουμε τον δεύτερο. Άρα, στην περίπτωση μας: 19 * 256 + 137 = 5001.

2.19

Τα αρχεία του τρέχοντος καταλόγου εμφανίζονται με την εντολή φλοιού ls, η οποία αντιστοιχεί στην εντολή πρωτοκόλλου FTP: NLST.

2.20

Αυτό συμβαίνει γιατί όπως είδαμε, ο υπολογιστής μας λέει, πριν την εντολή NLST, ότι ακούει για δεδομένα στο PORT 5001, θέλουμε δηλαδή να γίνει η σύνδεση των ports πριν την μεταφορά δεδομένων.

2.21

H bye μεταφράζεται στην QUIT.

2.22

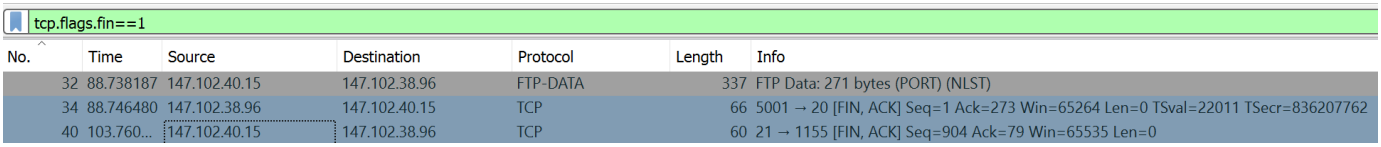
Ο εξυπηρετητής αποκρίνεται στο Request: QUIT με Response: 221 Goodbye.

2.23

Φίλτρο απεικόνισης: «tcp.flags.fin==1».

2.24

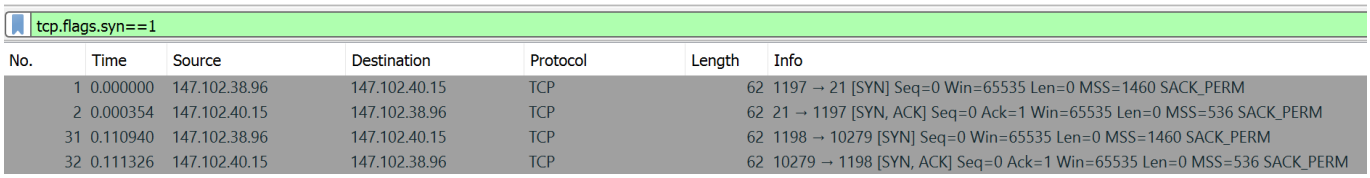
Παρατηρούμε πως η απόλυση των συνδέσεων έγινε από την πλευρά του σέρβερ όσον αφορά τις εντολές ελέγχου FTP (πακέτο 40) και από την πλευρά του πελάτη όσον αφορά τα μηνύματα δεδομένων (πακέτο 34).


tcp.flags.fin==1

No.	Time	Source	Destination	Protocol	Length	Info
32	88.738187	147.102.40.15	147.102.38.96	FTP-DATA	337	FTP Data: 271 bytes (PORT) (NLST)
34	88.746480	147.102.38.96	147.102.40.15	TCP	66	5001 → 20 [FIN, ACK] Seq=1 Ack=273 Win=65264 Len=0 TSval=22011 TSecr=836207762
40	103.760...	147.102.40.15	147.102.38.96	TCP	60	21 → 1155 [FIN, ACK] Seq=904 Ack=79 Win=65535 Len=0

2.25

Όπως βλέπουμε, οι θύρες πηγής/προορισμού είναι οι 1197/21 για τις εντολές ελέγχου και οι θύρες πηγής/προορισμού για τη μεταφορά δεδομένων είναι οι 1198/10279.


tcp.flags.syn==1

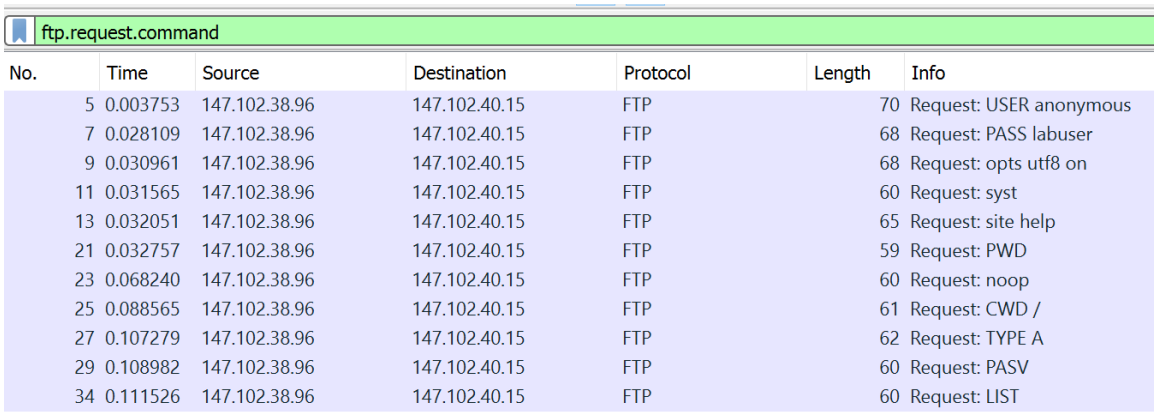
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	147.102.38.96	147.102.40.15	TCP	62	1197 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
2	0.000354	147.102.40.15	147.102.38.96	TCP	62	21 → 1197 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=536 SACK_PERM
31	0.110940	147.102.38.96	147.102.40.15	TCP	62	1198 → 10279 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM
32	0.111326	147.102.40.15	147.102.38.96	TCP	62	10279 → 1198 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=536 SACK_PERM

2.26

Παρατηρούμε τις εξής εντολές:

- ***Request: USER anonymous***
- ***Request: PASS labuser***
- ***Request: opts utf8 on***
- ***Request: syst***

- ***Request: site help***
- ***Request: PWD***
- ***Request: noop***
- ***Request: PWD***
- ***Request: TYPE A***
- ***Request: PASV***
- ***Request: LIST***



ftp.request.command

No.	Time	Source	Destination	Protocol	Length	Info
5	0.003753	147.102.38.96	147.102.40.15	FTP	70	Request: USER anonymous
7	0.028109	147.102.38.96	147.102.40.15	FTP	68	Request: PASS labuser
9	0.030961	147.102.38.96	147.102.40.15	FTP	68	Request: opts utf8 on
11	0.031565	147.102.38.96	147.102.40.15	FTP	60	Request: syst
13	0.032051	147.102.38.96	147.102.40.15	FTP	65	Request: site help
21	0.032757	147.102.38.96	147.102.40.15	FTP	59	Request: PWD
23	0.068240	147.102.38.96	147.102.40.15	FTP	60	Request: noop
25	0.088565	147.102.38.96	147.102.40.15	FTP	61	Request: CWD /
27	0.107279	147.102.38.96	147.102.40.15	FTP	62	Request: TYPE A
29	0.108982	147.102.38.96	147.102.40.15	FTP	60	Request: PASV
34	0.111526	147.102.38.96	147.102.40.15	FTP	60	Request: LIST

2.27

Στην περίπτωσή μας, χρησιμοποιήθηκε το όνομα χρήστη «anonymous» και ο κωδικός χρήστη «labuser».

2.28

Για την εμφάνιση της λίστας αρχείων, χρησιμοποιήθηκε η εντολή FTP πρωτοκόλλου LIST.

2.29

Εφαρμόζουμε το φίλτρο ftp.response και βλέπουμε τα αιτήματα του πελάτη και τις α-ποκρίσεις του εξυπηρετητή. Έτσι βρίσκουμε την απάντηση του σέρβερ, η οποία είναι «Response: 227 Entering Passive Mode (147,102,40,15,40,39)»:

29	0.108982	147.102.38.96	147.102.40.15	FTP	60	Request: PASV
30	0.110723	147.102.40.15	147.102.38.96	FTP	104	Response: 227 Entering Passive Mode (147,102,40,15,40,39).

2.30

Η εγκατάσταση σύνδεσης TCP που αφορούν τα μηνύματα δεδομένων FTP γίνεται από την πλευρά του πελάτη.

2.31

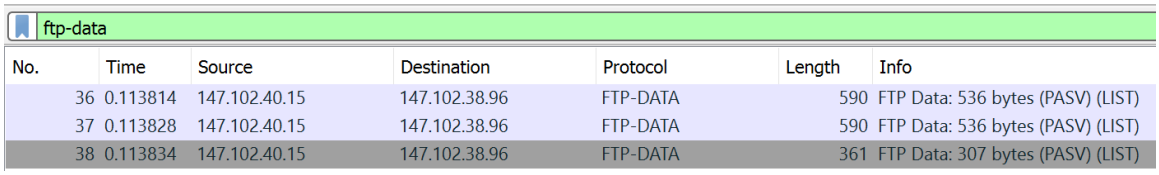
Για τη μεταφορά δεδομένων FTP, ο εξυπηρετητής χρησιμοποιεί τη θύρα 10279 για τη μεταφορά δεδομένων. Παρατηρώντας την απόκριση στο 2.29, ο αριθμός αυτός προκύπτει από τους 2 τελευταίους δεκαδικούς αριθμούς που εμφανίζονται στην απόκριση (40,39) ως εξής: $40 \cdot 256 + 39 = 10279$.

2.32

Αντίστοιχα, από την πλευρά του πελάτη, η θύρα 1198 που χρησιμοποιείται για τη μεταφορά δεδομένων προκύπτει ως η αμέσως επόμενη της θύρας που χρησιμοποιήθηκε για τη σύνδεση ελέγχου (1197).

2.33

Στάλθηκα 3 πακέτα δεδομένων:


ftp-data

No.	Time	Source	Destination	Protocol	Length	Info
36	0.113814	147.102.40.15	147.102.38.96	FTP-DATA	590	FTP Data: 536 bytes (PASV) (LIST)
37	0.113828	147.102.40.15	147.102.38.96	FTP-DATA	590	FTP Data: 536 bytes (PASV) (LIST)
38	0.113834	147.102.40.15	147.102.38.96	FTP-DATA	361	FTP Data: 307 bytes (PASV) (LIST)

2.34

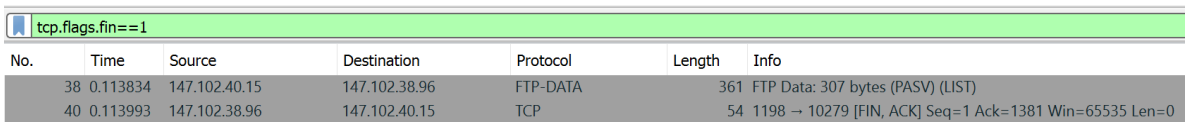
Γνωρίζουμε (από προηγούμενες ασκήσεις) πως ο σέρβερ 147.102.40.15 έχει MTU 576 bytes, άρα συνολικά με την προσθήκη του Ethernet Header έχουμε μέγιστο μέγεθος πακέτου 590 bytes.

2.35

Για την απόλυση σύνδεσης όσον αφορά τις εντολές ελέγχου δεν βλέπουμε να υπάρχει κάποια σχετική καταγραφή, υποθέτουμε όμως ότι γίνεται από τον πελάτη, αφού εμείς κλείνουμε την σύνδεση (κλείνουμε τον file explorer)

2.36

Όπως φαίνεται παρακάτω, η απόλυση σύνδεσης όσον αφορά τα μηνύματα δεδομένων γίνεται από τον εξυπηρετητή.


tcp.flags.fin==1

No.	Time	Source	Destination	Protocol	Length	Info
38	0.113834	147.102.40.15	147.102.38.96	FTP-DATA	361	FTP Data: 307 bytes (PASV) (LIST)
40	0.113993	147.102.38.96	147.102.40.15	TCP	54	1198 → 10279 [FIN, ACK] Seq=1 Ack=1381 Win=65535 Len=0

Άσκηση 3: TFTP

3.1

Το TFTP χρησιμοποιεί το πρωτόκολλο μεταφοράς UDP.

3.2

Για την πρώτη επικοινωνία πελάτη-εξυπηρετητή TFTP: Θύρα πηγής: 1200 και Θύρα προορισμού: 69.

3.3

Κατά τη μεταφορά δεδομένων, έχουμε Θύρα πελάτη: 1200 και Θύρα εξυπηρετητή: 50031.

3.4

Η θύρα 69 αντιστοιχεί στο πρωτόκολλο TFTP.

3.5

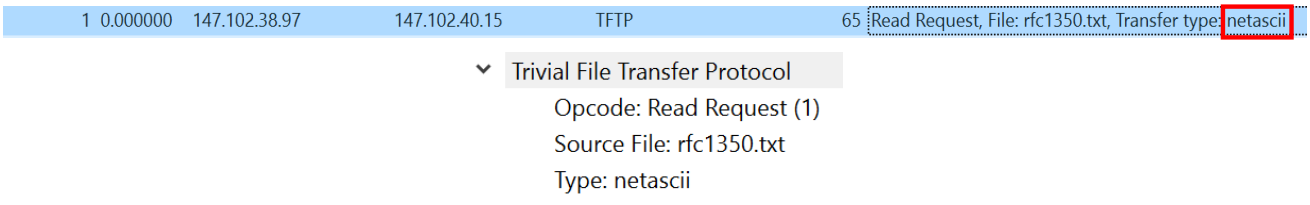
Σύμφωνα με το άρθρο που δίνεται, προκειμένου να δημιουργηθεί μια σύνδεση, κάθε άκρο επιλέγει ένα Transfer Identifier (TID), το οποίο και θα χρησιμοποιείται κατά τη διάρκεια της σύνδεσης. Το κάθε άκρο της επικοινωνίας αυτής επιλέγει τυχαία μία από τις διαθέσιμες θύρες, έτσι ώστε να μειωθεί στο ελάχιστο η πιθανότητα τα 2 άκρα να επέλεξαν ίδια θύρα. Κάθε πακέτο που μεταδίδεται κατά τη σύνδεση αυτή φέρει και τα 2 TID των τερματικών της σύνδεσης, τα οποία και δίνει στο UPD πρωτόκολλο ως Source και Destination Port. Ο κόμβος που κάνει την αρχική αίτηση (εν προκειμένω ο δικός μας, ο οποίος στέλνει RRQ – Read Request), έχει επιλέξει τυχαία τη θύρα που θα χρησιμοποιήσει και στέλνει το αρχικό αίτημα στη θύρα 69 στον εξυπηρετητή. Με τη σειρά του, ο σέρβερ αποκρίνεται, υπό κανονικές συνθήκες με το TID που εκείνος επέλεξε και που διατηρεί για το υπόλοιπο της σύνδεση.

3.6

Το αρχείο rfc1350.txt μεταφέρεται με ASCII.

3.7

Ο τρόπος μεταφοράς καθορίζεται στο πρώτο πακέτο και ειδικότερα στο πεδίο Type της επικεφαλίδας TFTP.



3.8

Καταγράφηκαν οι ακόλουθοι τύποι TFTP μηνυμάτων:

- ***Opcode: Read Request (1)***
- ***Opcode: Data Packet (50)***
- ***Opcode: Acknowledgment (50)***

3.9

Το TFTP λύνει το πρόβλημα αναξιοπιστίας του UDP με τον ακόλουθο τρόπο: κάθε πακέτο που λαμβάνεται με έναν μοναδικό (αύξοντα) αριθμό Block από το ένα άκρο, στέλνεται και ένα TFTP μήνυμα τύπου Acknowledgment για το Block από το άλλο άκρο με τον ίδιο αριθμό προκειμένου να σιγουρευτούμε πως ολοκληρώθηκε επιτυχώς η μεταφορά κάθε datagram.

3.10

Χρησιμοποιείται ο τύπος μηνύματος ***Acknowledgment***, ο οποίος δηλώνεται στο πεδίο Opcode της επικεφαλίδας TFTP.

3.11

Κάθε μήνυμα TFTP που μεταφέρει δεδομένα από τον σέρβερ σε εμάς (πλην του τελευταίου) έχει μέγεθος 516 bytes (4 bytes η επικεφαλίδα TFTP και 512 bytes δεδομένων), ενώ το συνολικό μέγεθος του πακέτου είναι 558 bytes.

3.12

Όπως αναφέραμε μεταφέρονται 512 bytes δεδομένων.

3.13

Ο πελάτης αντιλαμβάνεται το τέλος της μετάδοσης δεδομένων όταν λαμβάνει πακέτο με δεδομένα μεγέθους το πολύ έως 511 bytes.