Ονοματεπώνυμο: Ανδρέας Στάμος (03120***)	**Ομάδα:** 1
Όνομα PC/ΛΣ: linux / Ubuntu 22.04.2 LTS (με VPN στο δίκτυο του Πολυτεχνείου)	**Ημερομηνία:** 28/11/2023
Διεύθυνση IP: 147.102.131.218	**Διεύθυνση MAC:** DE–3F–DC–B2–E0–D0

Εργαστηριακή Άσκηση 8
TELNET, FTP και TFTP

Απαντήστε στα ερωτήματα στον χώρο που σας δίνεται παρακάτω και στην πίσω σελίδα εάν δεν επαρκεί. Το φυλλάδιο αυτό θα παραδοθεί στον επιβλέποντα.

Η εργασία υλοποιήθηκε με σύνδεση στο δίκτυο VPN του Πολυτεχνείου.

Άσκηση 1

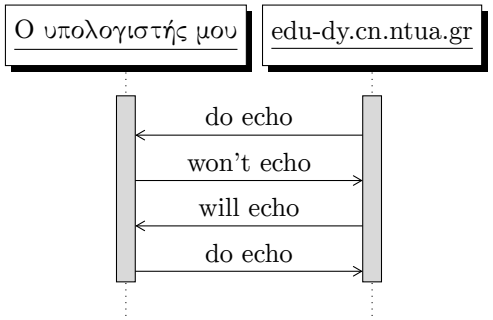
1.1 TCP

1.2 Θύρα 23 τον εξυπηρετητή και Θύρα 33096 στον υπολογιστή μου

1.3 Η 23

1.4 `telnet`

1.5



1.6 Ναι, και ο υπολογιστής μας δεν το δέχεται.

1.7 Όχι δεν το ζητά.

1.8 Ναι.

1.9 Ναι το `do echo` που έστειλε ο υπολογιστής μας σε συνέχεια του `will echo` που έστειλε ο edu-dy.cn.ntua.gr, που έδειξε την πρόθεση του να επαναλαμβάνει τους χαρακτήρες που του στέλνουμε.

1.10 Χαρακτηρά προς χαρακτήρα που στέλνουμε για το login μας τον στέλνει (echoing) ξανά πίσω προς τα εμάς.

1.11 Ο edu-dy.cn.ntua.gr μας πρότεινε να επαναλαμβάνει χαρακτήρες, και εμείς του είπαμε όντως να το κάνει.

1.12 `ip.dst == 147.102.40.15`

1.13 Ο υπολογιστής μας αποστέλλει 10 πακέτα συνολικά.

Ο υπολογιστής μας αποστέλλει 5 χαρακτήρες (`abcd` και το Carriage Return). Κάθε χαρακτήρας μπαίνει σε ξεχωριστό πακέτο (5 πακέτα για αυτό). Επίσης ο εξυπηρετητής κάνει echo ό,τι του στέλνουμε. Οπότε ο υπολογιστής μας πρέπει να απαντήσει με ACK στα πακέτα που του στέλνει ο εξυπηρετητής (5 πακετά για αυτό). Οπότε συνολικά απαιτούνται 10 πακέτα.

Συμπλήρωση: Προκειμένου, το τερματικό να φαίνεται αποκρίσιμο στον χρήστη, ο αλγόριθμος Nagle του TCP δεν χρησιμοποιείται και τα δεδομένα τοποθετούνται απευθείας στην σύνδεση. (αυτό δηλώνεται με το socket option `SO_OOBINLINE` που μπορείς κανείς να δει ότι το Telnet πράγματι θέτει εκτελώντας το Telnet με `strace`). Αυτός είναι ο λόγος που αποστέλλεται 1 IP πακέτο για κάθε χαρακτήρα. Διαφορετικά ο αλγόριθμος του Nagle θα προσπαθούσε να τα συμπτύξει.

1.14 Ο υπολογιστής μας αποστέλλει 6 πακέτα συνολικά.

Ο υπολογιστής μας αποστέλλει 5 χαρακτήρες (`efgh` και το Carriage Return). Κάθε χαρακτήρας μπαίνει σε ξεχωριστό πακέτο (5 πακέτα για αυτό). Ο εξυπηρετητής κάνει echo μόνο το Carriage Return, προσθέτοντας και ένα Line Feed ώστε να γίνει αλλαγή σειράς – η λογική του Telnet είναι πως στέλνουμε προς τον εξυπηρετητή ότι συμβαίνει στο πληκτρολόγιο και δείχνουμε στην οθόνη μόνο ότι στείλει ο εξυπηρετητής, με τον εξυπηρετητή εδώ να επιλέγει να μην δείχνει το password για λόγους ασφαλείας.

Ο υπολογιστής μας κατ' επεκτάση πρέπει να απαντήσει με ACK στο (μοναδικό) πακέτο που του στέλνει ο εξυπηρετητής (1 πακετά για αυτό). Οπότε συνολικά απαιτούνται 6 πακέτα.

1.15 Όχι. Λογικά για λόγους ασφαλείας, ώστε να μην εμφανιστεί στην οθονή – όπως είπαμε στο ερώτημα 1.14 το Telnet δείχνει στην οθόνη μόνο ότι του στείλει ο εξυπηρετητής.

1.16 Όχι.

1.17 Σύμφωνα με το RFC 857, που ορίζει τα σχετικά με το Telnet Echo Option, ο εξυπηρετητής δεν είναι υποχρεωμένος να στέλνει ακριβώς ό,τι λαμβάνει. Ο τρόπος λειτουργίας του Telnet, σύμφωνα με το RFC 854 (The Network Virtual Terminal) είναι πως αποστέλλει στον εξυπηρετητή ότι πατάει ο χρήστης στο πληκτρολόγιο, όμως δείχνει στην οθόνη μόνο ότι λαμβάνει από τον εξυπηρετητή, με σκόπο την λογική του τι φαίνεται στην οθόνη να την αναλάβει ο εξυπηρετητής, σαν να ήταν η οθόνη και το πληκτρολόγιο συνδεδεμένα σε αυτόν με σκοπό την δημιουργία ενός "δικτυακού τερματικού".

1.18 Τα δεδομένα κυκλοφορούν σε plain text στο δίκτυο. Εκτός των απλών δεδομένων, σε plain text κυκλοφορούν και τα passwords. Προφανώς ένας middle man μπορεί απλά να κοιτάζει τα πακέτα που διασχίζουν το δίκτυο και έτσι να διαβάσει το password. Αυτό είναι αρκετά σημαντική απειλή ασφαλείας. Για αυτό και κατά βάση σήμερα, το ssh, που τα δεδομένα μεταδίδονται κρυπτογραφημένα, έχει αντικαταστήσει το telnet.

Άσκηση 2

2.1 `host 147.102.40.15`

2.2 Enable Debugging. Συγκεκριμένα το tnftp (ο συνήθης ftp client σε Linux) θα εκτυπώνει τις FTP εντολές που στέλνει στην σύνδεση ελέγχου, εκτυπώνοντας μπροστά τους το >

2.3 TCP

2.4 Θύρα 21 του εξυπηρετητή και τυχαίου αριθμού θύρα του υπολογιστή μας για την σύνδεση ελέγχου.

Θύρα 20 του εξυπηρετητή και τυχαίου αριθμού θύρα του υπολογιστή μας για την σύνδεση δεδομένων.

2.5 Θύρα 21 του εξυπηρετητή για την σύνδεση ελέγχου και θύρα 20 του εξυπηρετητή για την σύνδεση δεδομένων.

2.6 Από τον εξυπηρετητή.

2.7 Οι εντολές είναι:

- `USER`
- `PASS`
- `SYST`
- `FEAT`
- `HELP`
- `EPRT`
- `LIST`

2.8 Ναι, και μπροστά τους εκτυπώνεται το > προκειμένου να γνωρίζουμε πως πρόκειται για εντολές που στέλνει ο υπολογιστής μας προς τον εξυπηρετητή.

2.9 `USER`

2.10 1

2.11 `PASS`

2.12 1

2.13 Μια ομοιότητα είναι πως το όνομα χρήστη και ο κωδικός χρήστης μεταδίδονται σε plain text.

Μια διαφορά είναι πως στο FTP το όνομα χρήστη και ο κωδικός χρήστης μεταδίδονται με ειδικές εντολές του FTP, που ο πελάτης αποφασίζει ότι θέλει να στείλει. Μάλιστα κάθε αυτοτελή εντολή αφού προκύπτει "ενιαία", τοποθετείται τελικά σε ένα IP πακέτο. Αντίθετα, στο TELNET το πρόγραμμα-πελάτης και γενικότερα το πρωτόκολλο του TELNET δεν γνωρίζουν για την διαδικασία αυθεντικοποίησης, και απλά μεταδίδουν τους χαρακτήρες που ο χρήστης πατά στο πληκτρολόγιο και εμφανίζουν ότι στέλνει ο εξυπηρετητής. Δηλαδή, πρακτικά, ο εξυπηρετητής αναλαμβάνει να στείλει ένα login prompt και έπειτα να ερμηνεύσει ότι λάβει ως όνομα χρήστη και κωδικό χρήστη. Προκειμένου, μάλιστα, το τερματικό να φαίνεται αποκρίσιμο στον χρήστη, ο αλγόριθμος Nagle του TCP δεν χρησιμοποιείται και τα δεδομένα τοποθετούνται απευθείας στην σύνδεση. (αυτό δηλώνεται με το socket option `SO_OOBINLINE` που μπορείς κανείς να δει ότι το Telnet πράγματι θέτει εκτελώντας το Telnet με `strace`)

2.14 Όχι.

2.15 `PBSZ` και `PROT`

2.16 1 πακέτο στάλθηκε από τον υπολογιστή μας και 9 πακέτα στάλθηκαν από τον εξυπηρετητή.

2.17 Το γεγονός ότι η απάντηση στην εντολή `HELP` αποτελείται από πολλές γραμμές, δηλώνεται στην πρωτή γραμμή με μια παύλα (hyphen) μεταξύ του κωδικού του FTP Reply και του Text που την ακολουθεί και επίσης με την ένα κενό (αυστηρά <Space>) μεταξύ του κωδικού του κωδικού FTP Reply και του Text της τελευταίας γραμμής.

2.18 Το default στο tnftp, που είναι ο συνήθης FTP client στο Linux, όταν χρησιμοποιείται σε Active Mode, είναι να χρησιμοποιεί την εντολή `EPRT` αντί της εντολής `PORT` προκειμένου να είναι συμβάτος με το IPv6. Προκειμένου να εμφανισθεί η εντολή `PORT`, που αναφέρεται στην εκφώνηση, εκτελέστηκε η εντολή tnftp (του προγράμματος-πελάτη) `epsv`, που δηλώνει να μην γίνει χρήση της εντολής EPRT.

Οι 4 πρώτοι δεκαδικοί αριθμοί (διαχωρίζονται με τον χαρακτήρα `','`) αναπαριστούν Την IP διεύθυνση του πελάτη δοσμένη byte προς byte, όπου κάθε byte δίνεται ως δεκαδικός αριθμός (0-255) με το κάθε δεκαδικό ψηφίο να δίνεται ως ASCII χαρακτήρας.

2.19 Ακολουθείται η ίδια σύμβαση που ακολουθείται για την IP διεύθυνση, όπως περιγράφηκε στο ερώτημα 2.18. Συγκεκριμένα ο αριθμός TCP θύρας, γνωρίζουμε πως αποτελείται από 2 bytes. Έτσι κάθε byte δίνεται σε αναπαράσταση δεκαδικού αριθμού (0-255), με το κάθε δεκαδικό ψηφίο να δίνεται ως ASCII χαρακτήρας. Μάλιστα, τα bytes δίνονται με την ίδια σειρά πως θα μπαίνανε στην TCP επικεφαλίδα. (με big-endian σειρά, όπως ορίζει και το RFC 1700 για το network byte order)

2.20 `LIST`

2.21 Σύμφωνα με την παράγραφο 3.2, ο πελάτης πρέπει να κάνει "listen" στην θύρα της σύνδεσης δεδομένων, προτού στείλει μια εντολή που αφορά σε μεταφορά δεδομένων (η `LIST` είναι τέτοια εντολη). Αν δεν σταλεί η εντολή `PORT` πριν την `LIST`, τότε ο εξυπηρετητής θα θεωρήσει ότι ισχύει η default FTP θύρα δεδομένων, δηλαδή η 20, οπότε θα προσπαθήσει να ανοίξει σύνδεση με τον υπολογιστή μας στην θύρα 20. Στο `tnftp` μπορεί να απενεργοποιηθεί η αποστολή εντολής `PORT` με χρήση της tnftp εντολής (εντολή του προγράμματος-πελάτη) `sendport`. Όμως, ακόμα και τότε, το `tnftp` από την λογική της υλοποίησής του, δεν κάνει bind στην θύρα 20. (Και καλά κάνει, αφού στην θύρα 20 θα μπορούσε να κάνει bind κάποιος FTP εξυπηρετητής που πιθανώς λετουργεί επίσης στον υπολογιστή)

2.22 `QUIT`

2.23 `221 Goodbye` (στο τέλος υπάρχει Carriage Return και Line Feed – έτσι συμβολίζεται η αλλαγή σειράς στα Windows)

2.24 `tcp.flags.fin == 1`

2.25 Και η σύνδεση ελέγχου και η σύνδεση δεδομένων, απολύεται πρώτα από την πλευρά του εξυπηρετητή.

2.26 `tcp.flags.syn ==1`

2.27 Για τις εντολές ελέγχου χρησιμοποιείται η TCP θύρα 21 του εξυπηρετητή και 40548 του υπολογιστή μας. Για την μεταφορά δεδομένων χρησιμοποιείται η TCP θύρα 47736 του εξυπηρετητή και 39135 του υπολογιστή μας.

2.28 Για την μεταφορά δεδομένων χρησιμοποιείται η TCP θύρα 47736 του εξυπηρετητή και 39135 του υπολογιστή μας. Η σύνδεση ανοίγει από τον υπολογιστή μας.

2.29 Οι εντολές που έστειλε ο υπολογιστής μας είναι:

- `AUTH TLS`
- `AUTH SSL`
- `USER`
- `PASS`
- `SYST`
- `FEAT`
- `OPTS`
- `PWD`
- `TYPE`
- `PASV`
- `MLSD`

Όνομα χρήστη: anonymous

Κωδικός χρήστη: anonymous@example.com

2.30 `MLSD`

2.32 `227 Entering Passive Mode (147,102,40,15,186,120).`

2.33 Όμοια με την ερώτηση 2.19 ο προτελευταίος και τελευταίος αριθμός είναι το 1ο και 2ο byte της θύρας δεδομένων (big-endian σειρά βάση και του RFC 1700) του εξυπηρετητή εκφρασμένο σε δεκαδική μορφή με το κάθε δεκαδικό ψηφίο να δίνεται ως ASCII κωδικό.

Πιο συγκεκριμένα: 186=0χba, 120=0x78. Άρα ο αριθμός θύρας είναι 0xba78=47736, όπως πράγματι παρατηρήσαμε προηγουμένως.

2.34 Επιλέγεται τυχαία από τον πελάτη. Ο πελάτης είναι που ανοίγει την σύνδεση. Οπότε εκείνος στέλνει πρώτος προς τον εξυπηρετητή. Πρακτικά, ο εξυπηρετητής μαθαίνει τν θύρα του πελάτη μέσω του TCP SYN που στέλνει ο πελάτης.

2.35 Στάλθηκαν 9 μηνύματα, συνολικού μεγέθους 4409 bytes (τα πρώτα 8 είναι 524 bytes το καθένα και το τελευταίο είναι 217 bytes).

2.36 Καθορίζεται από το `MSS` του εξυπηρετητή, που γνωρίζουμε από παλαιότερη εργαστηρική άσκηση πως αντιστοιχεί σε μέγιστο IP πακέτο μεγέθους 576 bytes. Εν προκειμένω είναι 576 bytes - 20 bytes IP επικεφαλίδα - 32 bytes TCP επικεφαλίδα = 524 bytes μέγεθος τεμαχίου.

Η πιθανή σύμπτυξη των δεδομένων, ώστε τελικά να συμπληρωνόνται πλήρη τεμάχια TCP, αν το FTP παραδίδει τμηματικά τα δεδομένα στο TCP, υλοποιείται – μάλλον – από τον αλγόριθμο Nagle του TCP.

2.37 Από τον εξυπηρετητή.

2.38 Από τον πελάτη.

Άσκηση 3

3.1 UDP

3.2 Read Request, Data και Acknowledgment

3.3 Opcode μεγέθους 2 bytes

3.4 Θύρα πελάτη: 41312

Θύρα εξυπηρετητή: 69

3.5 Θύρα πελάτη: 41312

Θύρα εξυπηρετητή: 38060

3.6 69

3.7 Ο πελάτης στέλνει το αρχικό request στην θύρα 69 του εξυπηρετητή. Κάθε πλευρά διαλέγει μια συγκεκριμένη θύρα με τυχαίο τρόπο για την μετέπειτα επικοινωνία. Η κάθε πλευρά μαθαίνει την θύρα που πρέπει να στέλνει στην άλλη πλευρά από το Source Port της UDP επικεφαλίδας των δεδομενογραμμάτων που λαμβάνει. Η τυχαία επιλογή θύρας γίνεται προκειμένου να είναι χαμηλή η πιθανότητα να μπλεχτούν μεταξύ τους δύο διαφορετικές συνδέσεις.

3.8 Στέλνονται σε netascii, που είναι μια μορφή ASCII που καθορίζεται από το RFC 764 και καθορίζει πως η αλλαγή σειράς πρέπει να συμβάινει με Carriage Return + Line Feed ενώ η απλή επιστροφή στην αρχή της σειράς πρέπει να συμβαίνει με Carriage Return + NUL. (ο ορισμός αυτός επιβάλλεται για λόγους συμβατότητας καθώς σε περιβάλλον Windows η αλλαγή σειράς συμβαίνει με Carriage Return + Line Feed ενώ σε Linux συμβαίνει μόνο με Line Feed)

3.9 Στο αρχικό μήνυμα, δηλαδή στην εντολή Read Request υπάρχει πεδίο `Mode` όπου μπαίνει η συμβολοσειρά `netascii`. (τοποθετείται ως συμβολοσειρά, όχι με κάποιον κωδικό)

3.10 Τα δεδομενογράμματα αριθμούνται με το πρώτο να λαμβάνει αριθμό 1. Ο εξυπηρετητής στην αρχή κάθε δεδομενογράμματος τοποθετεί τον αριθμό του. Ο πελάτης στέλνει επιβεβαιώσεις για κάθε δεδομενόγραμμα που λαμβάνει, αναγράφοντας τον αριθμό του δεδεμονογράμματος που έλαβε. Ο εξυπηρετητής στέλνει με Stop-and-wait, δηλαδή αν δεν ληφθεί επιβεβαίωση για ένα δεδομενόγραμμα δεν μεταδίδει σε επόμενο δεδομενόγραμμα. Ο εξυπηρετητής έχει ρολόι που κάνει timeout αν δε ληφθεί acknowledgment και ξαναστέλνει το ίδιο δεδομενόγραμμα, μέχρι να ληφθεί acknowledgment. Η λήξη μετάδοσης σηματοδοτείται από ποσότητα δεδομένων μικρότερη από 512 bytes, ενώ όσο τα δεδομένα έχουν μέγεθος 512 bytes θεωρείται πως η μετάδοση δεν έχει ολοκληρωθεί.

3.11 Τα δεδομενογράμματα που αποστέλλει ο εξυπηρετητής, δηλαδή αυτά με opcode Data=3, περιέχουν τον αριθμό δεδομενογράμματος (`Block #`). Τα δεδομενογράμματα που απαντά ο πελάτης, δηλαδή τα acknowledgments, με opcode Acknowledgment=4, περιέχουν περιέχουν τον αριθμό δεδεμονογράμματος (`Block #`) που επιβεβαιώνεται πως ελήφθη.

3.12 524 bytes (8 bytes UDP επικεφαλίδα + 4 bytes επικεφαλίδα TFTP + 512 bytes δεδομένων)

3.13 512 bytes

3.14 524 bytes δεδομενόγραμμα UDP + 20 bytes επικεφαλίδα IPv4 + 14 bytes επικεφαλίδα Ethernet II = 558 bytes. (λογικά στο TFTP επιλέχθηκε 512 bytes ώστε να βγαίνει μέγεθος IP πακέτου το πολύ 576 bytes που είναι το ελάχιστο μέγεθος IP πακέτου που όλοι οφείλουν να αποδέχονται)

3.15 Μεταδίδονται λιγότερα από 512 bytes δεδομένων.