Όνοματεπώνυμο: Μάρκος Δεληγιάννης	Ομάδα: 3
Όνομα PC/ΛΣ: DESKTOP-SCJFUE1 / W10	Ημερομηνία: 2 / 1 / 2023
Διεύθυνση IP: 147.102.131.44	Διεύθυνση MAC: 00-ff-ee-de-8a-a1

Εργαστηριακή Άσκηση 12
Ασφάλεια

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

1

1.1 Κώδικας κατάστασης: 401 Φράση: Authorization Required

1.2 Όνομα επικεφαλίδας: “WWW-Authenticate” Μέθοδος: “Basic”

1.3 Όνομα επικεφαλίδας: “Authorization”

1.4 Authorization: **Basic ZWR1LWR5OnBhc3N3b3Jk\r\n**”

1.5 “edu-dy:password”

1.6 Αυτός ο μηχανισμός παρόλο που παρέχει τις υπηρεσίες της πιστοποίησης αυθεντικότητας και ελέγχου πρόσβασης δεν παρέχει εμπιστευτικότητα, καθώς τα δεδομένα δεν κρυπτογραφούνται ούτε χρησιμοποιείται hashing. Έτσι, ένας κακόβουλος τρίτος μπορεί να συλλάβει το μήνυμα http καθώς αυτό πηγαίνει στον προορισμό και έτσι να μάθει τα credentials του χρήστη.

2

2.1 Το TCP.

2.2 Client: 64506 Server: 22

2.3 Η θύρα 22.

2.4 “ssh”

2.5 Έκδοση πρωτοκόλλου: 2.0 Έκδοση λογισμικού: OpenSSH_6.6.1_hpn13v11
Σχόλια: FreeBSD-20140420

2.6 Έκδοση πρωτοκόλλου: 2.0 Έκδοση λογισμικού: PuTTY_Release_0.78
Σχόλια: -

2.7 Πλήθος: 19
Οι πρώτοι δύο είναι: **sntrup761x25519-sha512@openssh.com** και **curve448-sha512**

2.8 Πλήθος: 9
Οι πρώτοι δύο είναι: **ssh-ed448** και **ssh-ed25519**

2.9 Οι πρώτοι δύο είναι: **aes256-ctr** και **aes256-cbc**

2.10 Οι πρώτοι δύο είναι: **hmac-sha2-256** και **hmac-sha1**

2.11 Οι πρώτοι δύο είναι: **none** και **zlib**

2.12 Βάσει του κανόνα που αναφέρεται προκύπτει ότι ο αλγόριθμος που θα χρησιμοποιηθεί είναι ο **curve25519-sha256@libssh.org**. Το wireshark εμφανίζει αυτήν την πληροφορία στην καρτέλα Key Exchange εντός παρενθέσεων και στα δύο μηνύματα SSH (χρησιμοποιούνται παρενθέσεις διότι η πληροφορία δεν δίνεται ρητά, αλλά μπορεί να εξαχθεί).

2.13 Θα χρησιμοποιηθεί ο αλγόριθμος **aes256-ctr**.

2.14 Θα χρησιμοποιηθεί ο αλγόριθμος **hmac-sha2-256**.

2.15 Επιλέγεται **None, οπότε δεν χρησιμοποιείται αλγόριθμος συμπίεσης**.

2.16 Ναι, στο παράθυρο με τις λεπτομέρειες, στην καρτέλα με την έκδοση του πρωτοκόλλου SSH μέσα σε παρενθέσεις (αφού η πληροφορία δεν δίνεται ρητά αλλά μπορεί να εξαχθεί).

2.17 Client → Server: Elliptic Curve Diffie-Hellman Key Exchange Init (κωδικός 30)
Server → Client: Elliptic Curve Diffie-Hellman Key Exchange Reply (κωδικός 31)
& New Keys (κωδικός 21)
Client → Server: New Keys (κωδικός 21)

2.18 **Όχι**, διότι μετά την ανταλλαγή κλειδιών μεταξύ server και client όλη η ανταλλασσόμενη πληροφορία είναι κρυπτογραφημένη. Εάν με τη χρήση του wireshark μπορούσαμε να εξάγουμε αυτήν την πληροφορία τότε το ίδιο θα μπορούσε να γίνει και από έναν κακόβουλο τρίτο, το οποίο θα σήμαινε ότι το SSH δεν προσφέρει ασφάλεια.

2.19 Η υπηρεσία SSH προσφέρει authentication και access control, δεδομένου ότι απαιτείται εισαγωγή username και password. Σε αντίθεση όμως με άλλα πρωτόκολλα, όπως το TELNET, που υλοποιούν την ταυτοποίηση χρηστών, το SSH κρυπτογραφεί τα δεδομένα με ένα κλειδί το οποίο δεν μπορεί να γνωρίζει κανείς άλλος πέρα από τον server και client. Έτσι παρέχεται εμπιστιστευτικότητα, αφού κανένας τρίτος δεν μπορεί να αποκρυπτογραφήσει τα μηνύματα που ανταλλάσσονται. Τέλος, παρέχεται και ακεραιότητα των δεδομένων, αφού χρησιμοποιείται αλγόριθμος MAC. Δεδομένου ότι κανείς άλλος δεν μπορεί να υπογράψει με έγκυρο τρόπο το μήνυμα ο έλεγχος του MAC κατηγορηματικά πιστοποιεί ότι το πακέτο δεν έχει αλλοιωθεί με κανέναν τρόπο.

3

3.1 “host bbb2.cn.ntua.gr”

3.2 “tcp.flags.syn==1 and tcp.flags.ack==0”

3.3 Στις θύρες 80 και 443.

3.4 Η 80 αντιστοιχεί στο πρωτόκολλο HTTP και η 443 στο HTTPS.

3.5 6 συνδέσεις στην περίπτωση HTTP και 2 στην περίπτωση HTTPS.

3.6 Θύρες πηγής: 49366, 49367

3.7 Content type (μήκος 1 byte), Version (μήκος 2 bytes) και Length (μήκος 2 bytes).

3.8 1) Handshake (22) 2) Change Cipher Spec (20) 3) Application data (23)

3.9 Η έκδοση που αναφέρει ο client στο μήνυμα TLS “Client Hello” είναι η 1.0 (0x0301) (παρατηρούμε όμως και την 1.2 με τιμή 0x0303), αλλά αυτή που αναφέρει ο server και χρησιμοποιείται τελικά είναι η έκδοση 1.2 (0x0303).

3.10 1) Client Hello (1) 2) Server Hello (2) 3) Certificate (11)
4) Server Key Exchange (12) 5) Server Hello Done (14) 6) Client Key Exchange (16)
7) New Session Ticket (4)

3.11 Ο πελάτης έστειλε 2 μηνύματα Client Hello που αντιστοιχούν ένα προς ένα με τις συνδέσεις TCP στη θύρα 443 που καταγράψαμε στο ερώτημα 3.6.

3.12 Στο πρώτο μήνυμα Client Hello δηλώνεται η έκδοση TLSv1.2 (0x0303), όπως αναφέρθηκε στο 3.9.

3.13 Δηλώνονται οι εκδόσεις **1.2 και 1.3**. Η αριθμητική τιμή της έκδοσης 1.3 είναι **0x0304**.

3.14 Δηλώνονται τα πρωτόκολλα **HTTP/2** (αναγράφεται h2) **και HTTP/1.1**.

3.15 Το μήκος του τυχαίου αριθμού είναι 32 bytes. Τα 4 πρώτα bytes είναι:
4f3ff705 και αντιστοιχούν σε **ημερομηνία**.

3.16 Ο client υποστηρίζει 16 σουίτες κωδίκων. Οι δύο πρώτες έχουν τις εξής δεκαεξαδικές τιμές:
1) 0x1301 2) 0x1302 *Δεν λαμβάνουμε υπόψη την αρχική τιμή Reserved (0xfafa)*

3.17 Θα χρησιμοποιηθεί η έκδοση TLSv1.2 (0x0303).

3.18 Το μήκος του τυχαίου αριθμού είναι 32 bytes. Τα 4 πρώτα bytes είναι:
08fd663f και αντιστοιχούν σε **ημερομηνία**.

3.19 **Όχι**, δεδομένου ότι η τιμή της επικεφαλίδας “Compression Method” του Handshake protocol είναι null.

3.20 Το όνομα της σουίτας κωδίκων είναι: **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256**
Η δεκαεξαδική της τιμή είναι: **0xc02f**
Αλγόριθμος ανταλλαγής κλειδιών: **ECDHE**
Αλγόριθμος πιστοποίησης ταυτότητας: **RSA**
Αλγόριθμος κρυπτογράφησης: **AES 128 GCM**
Συνάρτηση κατακερματισμού: **SHA256**

3.21 Το μήκος του είναι 4276 bytes.

3.22 Μεταφέρονται **3** πιστοποιητικά με μήκη **1574, 1306 και 1380**.

3.23 Επιλέγοντας την εγγραφή παρατηρούμε ότι αναφέρεται ότι προκύπτει από “5 reassembled TCP segments”, κάθε ένα από τα οποία μεταφέρεται με ένα πλαίσιο Ethernet. Συνεπώς χρειάστηκαν 5 πλαίσια Ethernet.

3.24 Client: Μήκος κλειδιού: 32 bytes 5 πρώτα γράμματα: d74ed
Server: Μήκος κλειδιού: 32 bytes 5 πρώτα γράμματα: 814af

3.25 Το μήκος του μηνύματος είναι είναι **1 byte** και το μήκος των επικεφαλίδων είναι **5 bytes**, οπότε το συνολικό μήκος της εγγραφής TLS είναι **6 bytes**.

3.26 Μήκος = Length = 40 bytes.

3.27 Ναι, παρατηρήσαμε.

3.28 Μεταφέρονται δεδομένα του πρωτοκόλλου **HTTP/2**.

3.29 Όχι, δεν παρατηρήσαμε.

3.30 Οι εγγραφές αυτές στέλνονται συνήθως για τον τερματισμό της σύνδεσης, όταν δεν υπάρχουν επιπλέον δεδομένα προς αποστολή.

3.31 Στα μηνύματα HTTP μπορούμε να βρούμε όποια φράση θέλουμε από την ιστοσελίδα, ενώ στο HTTPS όχι. Αυτό είναι αναμενόμενο, δεδομένου ότι τα δεδομένα στο HTTPS είναι κρυπτογραφημένα.

3.32 Το πρωτόκολλο HTTP υλοποιεί την πιστοποίηση της αυθεντικότητας του χρήστη, όπως είδαμε στην άσκηση 1. Εντούτοις, αυτό συμβαίνει εφόσον ο εξυπηρετητής το επιβάλει (username και password του χρήστη), ενώ δεν υπάρχει τρόπος να πιστοποιηθεί η αυθεντικότητα του server. Επιπλέον τα δεδομένα δεν είναι κρυπτογραφημένα, οπότε δεν υπάρχει εμπιστευτικότητα ούτε ακεραιότητα των δεδομένων. Αντιθέτως το HTTPS παρέχει και τις τρεις υπηρεσίες, αφού, όπως φαίνεται στο 3.20, χρησιμοποιεί μία σουίτα κωδίκων για ανταλλαγή κλειδιών, πιστοποίηση ταυτότητας, κρυπτογράφηση δεδομένων και κατακερματισμό μηνυμάτων. Συμπερασματικά, το HTTPS είναι πολύ ασφαλέστερο από το HTTP.