



ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 12: ΑΣΦΑΛΕΙΑ



17 ΙΑΝΟΥΑΡΙΟΥ, 2023

ΘΟΔΩΡΗΣ ΑΡΑΠΗΣ – EL18028

Όνοματεπώνυμο: Θεodorής Αράπης	Ομάδα: 2
Όνομα PC/ΛΣ: pc-b09/ WINDOWS 95	Ημερομηνία: 17/1/2023
Διεύθυνση IP: 147.102.38.109	Διεύθυνση MAC: 78:45:C4:26:46:83

Άσκηση 1: Πιστοποίηση αυθεντικότητας στο πρωτόκολλο HTTP

1.1

Η απόκριση του εξυπηρετητή στο αρχικό μήνυμα HTTP τύπου GET, έχει status code 401 και φράση Authorization Required.

No.	Time	Source	Destination	Protocol	Length	Info
5	0.000607	147.102.38.109	147.102.40.15	HTTP	373	GET /auth/ HTTP/1.1
7	0.001638	147.102.40.15	147.102.38.109	HTTP	240	HTTP/1.1 401 Authorization Required (text/html)
17	8.810670	147.102.38.109	147.102.40.15	HTTP	416	GET /auth/ HTTP/1.1
19	8.813424	147.102.40.15	147.102.38.109	HTTP	122	HTTP/1.1 200 OK (text/html)
21	8.938280	147.102.38.109	147.102.40.15	HTTP	289	GET /favicon.ico HTTP/1.1
32	8.939178	147.102.40.15	147.102.38.109	HTTP	319	HTTP/1.1 200 OK (image/x-icon)

1.2

Το όνομα της επικεφαλίδας είναι WWW-Authenticate και υποδεικνύει τη μέθοδο Basic authentication.

```

Hypertext Transfer Protocol
> HTTP/1.1 401 Authorization Required\r\n
  Date: Thu, 12 Jan 2023 09:27:33 GMT\r\n
  Server: Apache/2.2.22 (FreeBSD) mod_ssl/2.2.22 OpenSSL/1.1.1\r\n
  WWW-Authenticate: Basic realm="Edu-DY TEST"\r\n
> Content-Length: 401\r\n
  Keep-Alive: timeout=5, max=100\r\n
  Connection: Keep-Alive\r\n
  Content-Type: text/html; charset=iso-8859-1\r\n
, ,

```

1.3

Το όνομα της σχετικής επικεφαλίδας είναι Authorization.

1.4

Τα διαπιστευτήρια Basic ZWR1LWR5OnBhc3N3b3Jk.

```

▼ Hypertext Transfer Protocol
> GET /auth/ HTTP/1.1\r\n
Host: edu-dy.cn.ntua.gr\r\n
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:52.0
Accept: text/html,application/xhtml+xml,application/javascript;q=0.9,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
> Authorization: Basic ZWR1LWR5OjBhc3N3b3Jk\r\n

```

1.5

Το αποτέλεσμα της αποκωδικοποίησης είναι: edu-dy:password.

1.6

Διαπιστώνουμε πως ο μηχανισμός πιστοποίησης αυθεντικότητας που παρέχει το HTTP και βασίζεται στην κωδικοποίηση Base64 δεν είναι καθόλου ασφαλής και αυτό διότι αρκεί κάποιος να καταφέρει να υποκλέπτει τα πακέτα της σύνδεσης μεταξύ των 2 άκρων, καθώς μετά μπορεί εύκολα να αποκωδικοποιήσει οποιαδήποτε ευαίσθητα δεδομένα εστάλησαν κατά την επικοινωνία αυτή. Συνεπώς, εφόσον ο αποστολέας και ο “κανονικός” παραλήπτης δεν είναι οι μόνοι που μπορούν να κατανοούν το περιεχόμενο της σύνδεσης δεν υπάρχει εμπιστευτικότητα.

Άσκηση 2: Υπηρεσία SSH – Secure SHell

No.	Time	Source	Destination	Protocol	Length	Info
4	0.012778	147.102.40.15	147.102.38.109	SSHv2	103	Server: Protocol (SSH-2.0-OpenSSH_6.6.1_hpn13v11 FreeBSD-20140420)
6	0.757040	147.102.38.109	147.102.40.15	SSHv2	82	Client: Protocol (SSH-2.0-PuTTY_Release_0.60)
8	0.757073	147.102.38.109	147.102.40.15	SSHv2	158	Client: Key Exchange Init
13	0.759474	147.102.40.15	147.102.38.109	SSHv2	534	Server: Key Exchange Init
14	0.759597	147.102.38.109	147.102.40.15	SSHv2	70	Client: Diffie-Hellman Group Exchange Request (Old)
15	0.766237	147.102.40.15	147.102.38.109	SSHv2	590	Server: Diffie-Hellman Group Exchange Group
17	0.876489	147.102.38.109	147.102.40.15	SSHv2	70	Client: Diffie-Hellman Group Exchange Init
22	0.896546	147.102.40.15	147.102.38.109	SSHv2	86	Server: Server: Diffie-Hellman Group Exchange Reply, New Keys
24	2.859042	147.102.38.109	147.102.40.15	SSHv2	70	Client: New Keys
25	2.859192	147.102.38.109	147.102.40.15	SSHv2	106	Client: Encrypted packet (len=52)
27	2.859615	147.102.40.15	147.102.38.109	SSHv2	106	Server: Encrypted packet (len=52)
29	14.091078	147.102.38.109	147.102.40.15	SSHv2	122	Client: Encrypted packet (len=68)
30	14.096543	147.102.40.15	147.102.38.109	SSHv2	122	Server: Encrypted packet (len=68)
31	14.096671	147.102.38.109	147.102.40.15	SSHv2	154	Client: Encrypted packet (len=100)
32	14.100421	147.102.40.15	147.102.38.109	SSHv2	154	Server: Encrypted packet (len=100)
34	19.137929	147.102.38.109	147.102.40.15	SSHv2	350	Client: Encrypted packet (len=296)
35	19.139291	147.102.40.15	147.102.38.109	SSHv2	122	Server: Encrypted packet (len=68)
36	19.139426	147.102.38.109	147.102.40.15	SSHv2	154	Client: Encrypted packet (len=100)
37	19.141631	147.102.40.15	147.102.38.109	SSHv2	154	Server: Encrypted packet (len=100)
39	45.685310	147.102.38.109	147.102.40.15	SSHv2	122	Client: Encrypted packet (len=68)

2.1

Το SSH χρησιμοποιεί το πρωτόκολλο μεταφοράς TCP.

- > Transmission Control Protocol, Src Port: 22, Dst Port: 1667, Seq: 1, Ack: 1, Len: 49
- > SSH Protocol

2.2

Χρησιμοποιούνται οι θύρες 22 (εξυπηρετητής) και 1667 (εμείς) του πρωτοκόλλου μεταφοράς TCP.

2.3

Η θύρα 22 αντιστοιχεί στο πρωτόκολλο εφαρμογής SSH.

22	Yes	Assigned	Yes ^[12]	Secure Shell (SSH), ^[11] secure logins, file transfers (scp, sftp) and port forwarding
----	-----	----------	---------------------	---

2.4

Η σύνταξη του φίλτρου είναι «ssh».

2.5

Το περιεχόμενο είναι γραμμένο στα Αγγλικά και τα Ελληνικά.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.012778	147.102.40.15	147.102.38.109	SSHv2	103	Server: Protocol (SSH-2.0-OpenSSH_6.6.1_hpn13v11 FreeBSD-20140420)
<div>> Frame 4: 103 bytes on wire (824 bits), 103 bytes captured (824 bits) on interface \Device\NPF_{6D4...}</div> <div>> Ethernet II, Src: 08:ec:f5:d0:d9:1d, Dst: 78:45:c4:26:46:83</div> <div>> Internet Protocol Version 4, Src: 147.102.40.15, Dst: 147.102.38.109</div> <div>> Transmission Control Protocol, Src Port: 22, Dst Port: 1667, Seq: 1, Ack: 1, Len: 49</div> <div>> SSH Protocol</div> <div>Protocol: SSH-2.0-OpenSSH_6.6.1_hpn13v11 FreeBSD-20140420</div>						

Βλέπουμε πως ο εξυπηρετητής χρησιμοποιεί την έκδοση SSH-2.0, την έκδοση λογισμικού OpenSSH_6.6.1_hpn13v11 και στα σχόλια εντοπίζουμε το FreeBSD-201404.

2.6

Εδώ παρατηρούμε τα εξής:

- **Έκδοση:** SSH-2.0
- **Λογισμικό:** PuTTY_Release_0.60

6	0.757040	147.102.38.109	147.102.40.15	SSHv2	82	Client: Protocol (SSH-2.0-PuTTY_Release_0.60)
> Frame 6: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface \Device\NPF_{6D416F...}						
> Ethernet II, Src: 78:45:c4:26:46:83, Dst: 00:00:5e:00:01:25						
> Internet Protocol Version 4, Src: 147.102.38.109, Dst: 147.102.40.15						
> Transmission Control Protocol, Src Port: 1667, Dst Port: 22, Seq: 1, Ack: 50, Len: 28						
SSH Protocol						
Protocol: SSH-2.0-PuTTY_Release_0.60						
[Direction: client-to-server]						
0000	00 00 5e 00 01 25 78 45 c4 26 46 83 08 00 45 00	...^...%xE &F...E				
0010	00 44 1b bf 40 00 80 06 69 ac 93 66 26 6d 93 66	...D...@...j...f&m f				
0020	28 0f 06 83 00 16 9f d4 22 63 e5 2f 24 e5 50 18	...(...)"c/\$ P...				
0030	ff ce 72 a8 00 00 53 53 48 2d 32 2e 30 2d 50 75	...r...SS H-2.0-Pu				
0040	54 54 59 5f 52 65 6c 65 61 73 65 5f 30 2e 36 30	TTY_ Release_0.60				
0050	0d 0a	..				

2.7

Όπως βλέπουμε, εμφανίζονται 4 αλγόριθμοι ανταλλαγής κλειδιών. Οι 2 πρώτοι εξ αυτών είναι οι diffie-hellman-group-exchange-sha256 και diffie-hellman-group-exchange-sha1.

147.102.38.109	147.102.40.15	SSHv2	158	Client: Key Exchange Init
kex_algorithms string: diffie-hellman-group-exchange-sha256,diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1				

2.8

Στο ίδιο πακέτο, εντοπίζουμε τους 2 αλγορίθμους παραγωγής κλειδιών ssh-rsa και ssh-dss

147.102.38.109	147.102.40.15	SSHv2	158	Client: Key Exchange Init
server_host_key_algorithms string: ssh-rsa,ssh-dss				

2.9

Οι 2 πρώτοι αλγόριθμοι κρυπτογράφησης που υποστηρίζει ο πελάτης με κατεύθυνση client to server είναι aes256-ctr, aes256-cbc.

147.102.38.109	147.102.40.15	SSHv2	158	Client: Key Exchange Init
encryption_algorithms_client_to_server string: aes256-ctr,aes256-cbc,rjndael-cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-ctr,aes128-cbc,blowfish-ctr,blowfish-cbc,3des-ctr,3des-cbc,arcfour256,arcfour128				

2.10

Αντίστοιχα, για τους αλγορίθμους πιστοποίησης αυθεντικότητας (mac), έχουμε τους hmac-sha1 και hmac-sha1-96.

147.102.38.109	147.102.40.15	SSHv2	158	Client: Key Exchange Init
mac_algorithms_client_to_server string: hmac-sha1,hmac-sha1-96,hmac-md5				

2.11

Αντίστοιχα για τους αλγορίθμους συμπίεσης (compression), έχουμε τους none και zlib.

147.102.38.109	147.102.40.15	SSHv2	158 Client: Key Exchange Init
compression_algorithms_client_to_server string: none,zlib			

2.12

Βλέποντας τη λίστα αλγορίθμων ανταλλαγής κλειδιών του εξυπηρετητή και συγκρίνοντας με αυτή του πελάτη, αναμένουμε να χρησιμοποιηθεί ο αλγόριθμος diffie-hellman-group-exchange-sha256, αφού είναι ο πρώτος κοινός αλγόριθμος στις δύο λίστες. Πράγματι, το επαληθεύουμε από το πεδίο Key Exchange όπως βλέπουμε παρακάτω.

Key Exchange (method:diffie-hellman-group-exchange-sha256)

2.13

Στην λίστα αλγορίθμων κρυπτογράφησης βλέπουμε πρώτο τον αλγόριθμο aes256-ctr. Παρατηρούμε πως είναι ο πρώτος από τη λίστα του πελάτη οπότε θα χρησιμοποιηθεί αυτός.

2.14

Με την ίδια διαδικασία εντοπίζουμε πως ο πρώτος κοινός αλγόριθμος πιστοποίησης αυθεντικότητας των δύο λιστών είναι ο hmac-sha1, οπότε θα χρησιμοποιηθεί αυτός.

2.15

Δε χρησιμοποιείται κανένας αλγόριθμος συμπίεσης (και οι δύο λίστες περιέχουν none ως πρώτη επιλογή).

2.16

Όχι δεν εμφανίζεται.

2.17

Παρατηρούμε τους εξής 6 τύπους μηνυμάτων:

- Diffie-Hellman Group Exchange Request (Old)
- Diffie-Hellman Group Exchange Group
- Diffie-Hellman Group Exchange Init
- Server: Diffie-Hellman Group Exchange Reply, New Keys

- **New Keys**
- **Encrypted Packet**

14	0.759597	147.102.38.109	147.102.40.15	SSHv2	70	Client: Diffie-Hellman Group Exchange Request (Old)
15	0.766237	147.102.40.15	147.102.38.109	SSHv2	590	Server: Diffie-Hellman Group Exchange Group
17	0.876489	147.102.38.109	147.102.40.15	SSHv2	70	Client: Diffie-Hellman Group Exchange Init
22	0.896546	147.102.40.15	147.102.38.109	SSHv2	86	Server: Server: Diffie-Hellman Group Exchange Reply, New Keys
24	2.859042	147.102.38.109	147.102.40.15	SSHv2	70	Client: New Keys
25	2.859192	147.102.38.109	147.102.40.15	SSHv2	106	Client: Encrypted packet (len=52)

2.18

Παρατηρούμε πως δε γίνεται αντιληπτό ποια πακέτα αφορούν το login και το password στην περίπτωση του SSH και ο λόγος είναι πως τα πακέτα αυτά είναι κρυπτογραφημένα.

2.19

Αναφορικά με την ασφάλεια του SSH:

- **Πιστοποίηση αυθεντικότητας**: Έχουμε authentication μέσω public-private keys, από τις ασφαλέστερες δηλαδή μεθόδους.
- **Εμπιστευτικότητα**: Λόγω της κρυπτογράφησης, το περιεχόμενο γίνεται κατανοητό μόνο από τον εξυπηρετητή και τον πελάτη.
- **Ακεραιότητα των δεδομένων**: Παρέχονται hashing αλγόριθμοι για data-integrity (MAC).

Κρίνεται, επομένως, ως μια ασφαλής επιλογή.

Άσκηση 3: Υπηρεσία HTTPS

3.1

Χρησιμοποιήσαμε το φίλτρο σύλληψης: «host bbb2.cn.ntua.gr».

3.2

Με το φίλτρο απεικόνισης:

«tcp.len==0 and ((tcp.seq==0 and tcp.ack==0) or (tcp.seq==0 and tcp.ack==1) or (tcp.seq==1 and tcp.ack==1))»

Εμφανίζονται όλες οι τριπλές χειραψίες που έγιναν. Εμφανίζονται 21 πακέτα TCP χειραψίας (και 7 Duplicates), επομένως συμπεραίνουμε πως έγιναν 7 TCP συνδέσεις.

3.3

Οι συνδέσεις έγιναν στις θύρες 80 (HTTP) και 443 (HTTPS) του εξυπηρετητή.

3.4

80-HTTP, 443-HTTPS.

3.5

Ανοίχτηκαν 6 συνδέσεις HTTP και 1 σύνδεση HTTPS.

3.6

Χρησιμοποιήθηκε η θύρα 1715.

3.7

Παρατηρούμε τα εξής πεδία:

- **Content Type (1 Byte)**
- **Version (2 Bytes)**
- **Length (2 Bytes)**

3.8

Καταγράφουμε τις παρακάτω τιμές:

- **Handshake (22)**
- **Change Cipher Spec (20)**
- **Application Data (23)**
- **Alert (21)**

3.9

Η έκδοση του πρωτοκόλλου Στρώματος Εγγραφών και η αριθμητική της τιμή είναι: TLS 1.2 (0x0303)

3.10

Καταγράφουμε τους εξής τύπους μηνυμάτων χειραψίας:

- **Client Hello (1)**
- **Server Hello (2)**
- **Certificate (11)**
- **Server Key Exchange (12)**
- **Server Hello Done (14)**

- **Client Key Exchange (16)**
- **Encrypted Handshake Message**
- **New Session Ticket (4)**

3.11

Ο πελάτης έστειλε 1 Client Hello, όσες και οι HTTPS συνδέσεις.

3.12

Η μέγιστη υποστηριζόμενη έκδοση που δηλώνεται από τον client είναι η TLS 1.0 (0x0301), η οποία είναι παλαιότερη από αυτήν του ερωτήματος 3.9.

3.13

Δεν εντοπίζουμε την επικεφαλίδα επέκτασης supported_versions.

3.14

Δηλώνονται τα παρακάτω ALPN πρωτόκολλα:

```

▼ Extension: application_layer_protocol_negotiation (len=14)
  Type: application_layer_protocol_negotiation (16)
  Length: 14
  ALPN Extension Length: 12
  ▼ ALPN Protocol
    ALPN string length: 2
    ALPN Next Protocol: h2
    ALPN string length: 8
    ALPN Next Protocol: http/1.1

```

3.15

Το μήκος του τυχαίου αριθμού είναι 32 bytes, με τα πρώτα 4 εξ αυτών να είναι τα 6a 52 93 00. Τα bytes αυτά δηλώνουν το GMT Unix Time.

3.16

Καταγράφονται 15 Cipher Suites, ενώ οι 2 πρώτες εξ αυτών είναι οι Cipher Suite:

- **TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)**
- **TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)**

3.17

Από τον εξυπηρετητή θα χρησιμοποιηθεί η έκδοση TLS 1.2 (0x0303).

3.18

Και εδώ περιέχονται 32 bytes στον τυχαίο αριθμό. Τα πρώτα 4 bytes του Random είναι τα 45 1e dc 25.

3.19

Όχι δεν χρησιμοποιείται compression method αφού το αντίστοιχο πεδίο έχει τιμή null (0).

Compression Method: null (0)

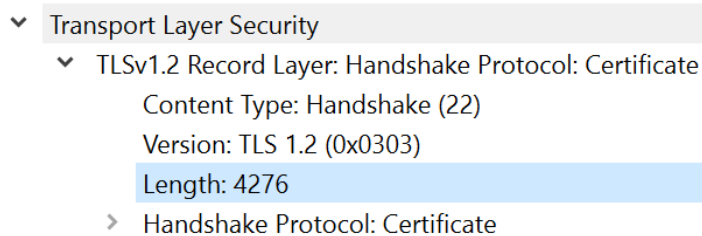
3.20

Τα ζητούμενα βρίσκονται στο πεδίο Cipher Suite, το οποίο στην περίπτωση μας έχει τιμή «TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256», η οποία είναι η σουίτα που επιλέχθηκε. Ειδικότερα, από το όνομα αυτό εξαγάγουμε τα εξής:

- Αλγόριθμος ανταλλαγής κλειδιών: ECDHE
- Αλγόριθμος πιστοποίησης ταυτότητας: RSA
- Αλγόριθμος κρυπτογράφησης: AES(128bits)
- Αλγόριθμος συνάρτησης κατακερματισμού: SHA(256bits)

3.21

Είναι 4276 bytes, όπως φαίνεται ακολούθως.



3.22

Μεταφέρονται 3 πιστοποιητικά:

- id-at-commonName = bbb2.cn.ntua.gr, Length: 1574
- id-at-commonName = R3, Length: 1306
- id-at-commonName = ISRG ROOT X1, Length: 1380

3.23

Χρειάστηκαν 4 πλαίσια Ethernet, ώστε να μεταφερθεί η παραπάνω εγγραφή TLS.

3.24

Ο πελάτης αποστέλλει δημόσιο κλειδί μήκους 32 bytes (5 αρχικά γράμματα: c5a80), όσα bytes αποστέλλει δηλαδή και ο εξυπηρετητής (5 πρότερα γράμματα: d5f2f).

3.25

Το μήκος της εγγραφής είναι 6 bytes, ενώ το μήκος του μηνύματος είναι 1.

The image shows a Wireshark packet capture of a TLSv1.2 Change Cipher Spec message. The packet list on the left shows 'Transport Layer Security' expanded, with 'TLSv1.2 Record Layer: Handshake Protocol: Client Key Exchange' and 'TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec' selected. The packet details on the right show 'Content Type: Change Cipher Spec (20)', 'Version: TLS 1.2 (0x0303)', and 'Length: 1'. The packet bytes on the right show the hex value '14 03 03 00 01 01' highlighted in blue.

3.26

Το μέγεθος του μηνύματος είναι 40 bytes.

The image shows a Wireshark packet capture of a TLSv1.2 Encrypted Handshake Message. The packet list on the left shows 'TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message' selected. The packet details on the right show 'Content Type: Handshake (22)', 'Version: TLS 1.2 (0x0303)', and 'Length: 40' highlighted in blue. The packet bytes on the right show the hex value '14 03 03 00 01 01' highlighted in blue.

3.27

Ναι παρατηρήσαμε.

413	17.347172	147.102.40.19	147.102.38.109	TLSv1.2	312	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
-----	-----------	---------------	----------------	---------	-----	---

3.28

Του HyperText Transfer Protocol 2.

The image shows a Wireshark packet capture of a TLSv1.2 Application Data message. The packet list on the left shows 'Transport Layer Security' expanded, with 'TLSv1.2 Record Layer: Application Data Protocol: HyperText Transfer Protocol 2' selected. The packet details on the right show 'Content Type: Application Data (23)', 'Version: TLS 1.2 (0x0303)', 'Length: 64', and 'Encrypted Application Data: 560392f0609cc549620ca6770b996de3cd0a222cab [Application Data Protocol: HyperText Transfer Protocol 2]'. The packet bytes on the right show the hex value '14 03 03 00 01 01' highlighted in blue.

3.29

Ναι, από την πλευρά του client.

> Frame 790: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface \Device\NPF_{6D416F...}

> Ethernet II, Src: 78:45:c4:26:46:83, Dst: 00:00:5e:00:01:25

> Internet Protocol Version 4, Src: 147.102.38.109, Dst: 147.102.40.19

> Transmission Control Protocol, Src Port: 1715, Dst Port: 443, Seq: 1978, Ack: 339853, Len: 31

▼ Transport Layer Security

▼ TLSv1.2 Record Layer: Encrypted Alert

Content Type: Alert (21)

Version: TLS 1.2 (0x0303)

Length: 26

Alert Message: Encrypted Alert

3.30

Το Alert μήνυμα εδώ λειτουργεί ως warning, προειδοποιώντας των client πως κλείνει η TCP σύνδεση (σταματάει ο client το session), αφού αμέσως μετά ακολουθούν TCP πακέτα με flags [FIN, ACK].

3.31

Στη περίπτωση του HTTP βρίσκουμε πακέτο που έχει ως περιεχόμενο το περιεχόμενο της ιστοσελίδας που ζητήσαμε σε μορφή html, όπως βλέπουμε παρακάτω. Αντίθετα, στα πακέτα HTTPS δε μπορούμε να βρούμε κάποιο πακέτο αναζητώντας το String BigBlueButton και αυτό διότι η πληροφορία μεταφέρεται κρυπτογραφημένη στο https σε αντίθεση με το http.

http

Packet details ▼ Narrow & Wide ▼ ☐ Case sensitive String ▼ BigBlueButton

No.	Time	Source	Destination	Protocol	Length	Info
10	0.001850	147.102.40.19	147.102.38.109	HTTP	1189	HTTP/1.1 200 OK (text/html)

▼ Line-based text data: text/html (285 lines)

```
\n
<!DOCTYPE html>\n
<html>\n
<head>\n
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">\n
\n
<title>BigBlueButton - Open Source Web Conferencing</title>\n
```

3.32

Συγκρίνοντας το HTTP με το HTTPS, μπορούμε να πούμε πως:

- **Πιστοποίηση αυθεντικότητας**: Στο HTTPS, όταν ένας client εκκινεί έναν “διάλογο” επικοινωνίας με έναν εξυπηρετητή, ο εξυπηρετητής επαληθεύει τη γνησιότητα του αντιστοιχίζοντας το private key του με το public key στο SSL/TLS certificate (το οποίο είναι signed από μία έμπιστη αρχή) της σελίδας που επισκεπτόμαστε. Στο HTTP δεν υπάρχει κάποια αντίστοιχη διαδικασία που να εξασφαλίζει την πιστότητα του εξυπηρετητή.
- **Εμπιστευτικότητα**: Στο HTTP τα δεδομένα στέλνονται ως plaintext, επομένως είναι άμεσα αναγνώσιμα από κάποιον που θα καταφέρει να υποκλέψει κάποια πακέτα. Αντιθέτως, το περιεχόμενο στο HTTPS είναι κρυπτογραφημένο, με αποτέλεσμα ακόμα και αν κάποιος υποκλέψει πακέτα να διαβάσει κάτι που δε βγάζει νόημα και από το οποίο δε μπορεί να εξαγει κάτι χρήσιμο.
- **Ακεραιότητα των δεδομένων**: Στο HTTPS είναι αδύνατον να μεταβληθούν τα δεδομένα χωρίς αυτό να γίνει αντιληπτό από τους συμμετέχοντες στη σύνδεση. Αντιθέτως, το HTTP είναι επιρρεπές σε επιθέσεις τύπου Man-In-The-Middle, οι οποίες θα μπορούσαν να αλλοιώσουν το περιεχόμενο των πακέτων.