Ονοματεπώνυμο: Ιωάννης Γιαννούκος			**Ομάδα:** 3
Όνομα PC/ΛΣ: JohnJohn / Windows 11		**Ημερομηνία:**	**16/1/2023**
Διεύθυνση IP:	**147.102.200.45**	**Διεύθυνση MAC:**	**3C-6A-A7-9A-B3-CF**

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.

Άσκηση 1

1.1) Στο πρώτο αίτημα επιστρέφεται ο κωδικός 401 (Authorization Required) με φράση επιστροφής “Authorization Required”.

1.2) WWW-Authenticate: Basic realm="Edu-DY TEST"

1.3) Επικεφαλίδα Authorization.

1.4) Στην επικεφαλίδα περιέχεται το εξής:
Credentials → edu-dy:password

1.5) ZWR1LWR5OnBhc3N3b3Jk → edu-dy:password

1.6) Συμπεραίνουμε ότι, ακόμα και αν υπάρχει κρυπτογράφηση δεδομένων, αυτή δεν είναι πολύ δύσκολο να σπάσει, γεγονός που κάνει όλη τη διαδικασία σχετικά ανούσια.

Άσκηση 2

2.1) Το SSH χρησιμοποιεί το πρωτόκολλο μεταφοράς TCP.

2.2) Οι θύρες που χρησιμοποιούνται (σε πακέτα που στέλνονται από τον υπολογιστή μου) είναι: Source Port = 54836, Destination Port = 22.

2.3) Στο πρωτόκολλο εφαρμογής SSH αντιστοιχεί η θύρα 22.

2.4) Φίλτρο απεικόνισης: “ssh”

2.5) Protocol: SSH-2.0-OpenSSH_6.6.1_hpn13v11 FreeBSD-20140420
Δεν περιέχονται περαιτέρω σχόλια.

2.6) Protocol: SSH-2.0-PuTTY_Release_0.78
Δεν περιέχονται περαιτέρω σχόλια.

2.7) kex_algorithms length: 470. Οι δύο πρώτοι εξ αυτών είναι οι sntrup761x25519-sha512@openssh.com και curve448-sha512.

2.8) server_host_key_algorithms length: 123. Οι δύο πρώτοι εξ αυτών είναι οι ssh-ed448 και ssh-ed25519.

2.9) Οι δύο πρώτοι εξ αυτών είναι οι aes256-ctr και aes256-cbc.

2.10) Οι δύο πρώτοι εξ αυτών είναι οι hmac-sha2-256 και hmac-sha1.

2.11) Οι δύο πρώτοι εξ αυτών είναι οι none και zlib.

2.12) Ναι, οι αλγόριθμοι κρυπτογράφησης που θα ακολουθήσουν τα δύο μέρη φαίνονται στην (αναπτυσσόμενη) επικεφαλίδα SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none), όπως την εμφανίζει το *Wireshark*.

2.13) Θα χρησιμοποιηθεί ο aes256-ctr (όπως φαίνεται και στην (αναπτυσσόμενη) επικεφαλίδα SSH Version 2 (encryption:aes256-ctr …) που εμφανίζει το *Wireshark*).

2.14) Θα χρησιμοποιηθεί ο hmac-sha2-256 (όπως φαίνεται και στην (αναπτυσσόμενη) επικεφαλίδα SSH Version 2 (… mac:hmac-sha2-256 …) που εμφανίζει το *Wireshark*).

2.15) Δεν θα χρησιμοποιηθεί αλγόριθμος (none) (όπως φαίνεται και στην (αναπτυσσόμενη) επικεφαλίδα SSH Version 2 (… compression:none) που εμφανίζει το *Wireshark*).

2.16) Όχι, το *Wireshark* δεν τους εμφανίζει σε κάποιο σημείο.

2.17) Καταγράφηκαν επίσης οι τύποι SSH “Elliptic Curve Diffie-Hellman Key Exchange Init” και “Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys” και “New Keys”.

2.18) Παρόλο ου φαίνεται ότι μεταφέρεται κάποια πληροφορία, η αποκωδικοποίηση των πληροφοριών login και password δεν είναι πλέον δυνατές κάπου, καθώς με τον αλγόριθμο κρυπτογράφησης το *Wireshark* δεν μπορεί απλώς να τις αποκωδικοποιήσει.

2.19) Το πρωτόκολλο SSH φαίνεται να είναι πάρα πολύ αξιόπιστο, κυρίως επειδή διαθέτει πολλούς αλγόριθμους κρυπτογράφησης, πράγμα που κάνει ακόμα δυσκολότερο το «σπάσιμό» τους.

Άσκηση 3

3.1) Φίλτρο σύλληψης: “host 147.102.40.19”

3.2) Φίλτρο απεικόνισης: “tcp.seq == 0 and tcp.ack == 0”

3.3) Οι συνδέσεις γίνονται στις θύρες 80 και 443 του εξυπηρετητή.

3.4) Η θύρα 80 είναι η συνήθης θύρα του πρωτοκόλλου HTTP και η 443 είναι αυτή του HTTPS.

3.5) Στην περίπτωση του HTTP έγιναν 4 συνδέσεις TCP και σε αυτήν του HTTPS έγινε μία.

3.6) Η σύνδεση που έγινε στην περίπτωση του HTTPS έγινε από την θύρα 54853 του υπολογιστή μου.

3.7) Τα 3 πεδία που κάθε εγγραφή TLS έχει στην αρχή του είναι αυτά με επικεφαλίδες Content Type, Version και Length.

3.8) Οι διαφορετικοί τύποι εγγραφών TLS που καταγράφηκαν είναι οι εξής:
Handshake (22), Application Data (23), Change Cipher Spec (20).

3.9) Η έκδοση του πρωτοκόλλου TLS φαίνεται στο πεδίο Version: 1.2 (0x0303).

3.10) Οι διαφορετικοί τύποι χειραψιών που καταγράφηκαν είναι οι εξής:
Client Hello(1), Server Hello(2), Certificate(11), Server Key Exchange(12), Server Hello Done(14), Client Key Exchange(16), New Session Ticket(4).

3.11) Ο πελάτης (εγώ) έστειλε 1 μήνυμα *Client Hello* στον εξυπηρετητή, δηλαδή τόσα μηνύματα όσες συνδέσεις TCP για την επικοινωνία μέσω HTTPS, που, όπως είδαμε, συνέβη 1 σύνδεση TCP.

3.12) Η έκδοση του πρωτοκόλλου TLS στο μήνυμα από τον πελάτη *Client Hello* είναι 1.0 (0x0301), η οποία δεν συμφωνεί με την τιμή που βρήκαμε στο υποερώτημα 3.9.

3.13) Ο πλοηγός μου είναι συμβατός με 2 επιπλέον εκδόσεις, τις 1.2 (0x0303) και 1.3 (0x0304).

3.14) Ο πλοηγός μου δεν είναι συμβατός με την έκδοση HTTP/2.

3.15) Το μήκος του τυχαίου αριθμού του μηνύματος *Client Hello* είναι 32 bytes. Τα τέσσερα πρώτα bytes του έχουν τιμές {c9, b6, ea, a5} και βρίσκονται σε ξεχωριστό υποπεδίο που λέγεται GNU Unix Time και ουσιαστικά παριστάνει μία ημερομηνία. Αυτή η τιμή, λογικά, συμβάλλει στην δημιουργία του αριθμού αυτού.

3.16) Ο πελάτης δηλώνει ότι υποστηρίζει 17 σουίτες κωδικών (cipher suites), οι δύο πρώτες των οποίων έχουν τιμές (0x1301) και (0x1303).

3.17) Από το μήνυμα *Server Hello* φαίνεται ότι θα χρησιμοποιηθεί η έκδοση TLS 1.2.

3.18) Το μήκος του τυχαίου αριθμού του μηνύματος *Server Hello* είναι 32 bytes. Τα τέσσερα πρώτα bytes του έχουν τιμές {7c, 7e, 67, c8}. Οι τιμές αυτές σε σύγκριση με αυτές του υποερωτήματος 3.15 δεν φαίνεται να έχουν κάποια σύνδεση, επομένως οι τιμές αυτές είναι πιθανό να παράγονται τυχαία.

3.19) Όχι, πελάτης και εξυπηρετητής δεν χρησιμοποιούν κάποια μέθοδο συμπίεσης, πράγμα που φαίνεται από την τιμή null (0) των σχετικών πεδίων Compression Method.

3.20) Οι παράμετροι οι οποίες εν τέλει επιλέχθηκαν για χρήση είναι οι εξής:
Σουίτα κωδικών:
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f).
Αλγόριθμος ανταλλαγής κλειδιών: ECDHE (EC Diffie-Hellman Server Params)
Αλγ. Πιστοποίησης ταυτότητας: RSA
Αλγ. Κρυπτογράφησης: GCM
Αλγ. Συνάρτησης Κατακερματισμού: SHA

3.21) Το μήκος του μηνύματος *Certificate* είναι 4276 bytes, που φαίνεται στο πεδίο Length της εγγραφής TLS.

3.22) Μεταφέρονται 3 πιστοποιητικά (certificates) με αντίστοιχα μήκη 1574, 1306 και 1380 bytes.

3.23) Για την παραπάνω εγγραφή χρειάστηκαν 4 πλαίσια Ethernet.

3.24) Το μήκος του δημοσίου κλειδιού του πελάτη φαίνεται στο πεδίο με επικεφαλίδα Pubkey Length, μέσα στην (αναπτυσσόμενη) επικεφαλίδα EC Diffie-Hellman Client Params.
Το μήκος του δημοσίου κλειδιού του εξυπηρετητή φαίνεται στο παραπάνω πεδίο και έχει τιμή 32 bytes.

3.25) Το μήκος της εγγραφής αυτής έχει μήκος 6 bytes και το μήκος του μηνύματος που μεταφέρει είναι μόνο 1 byte.

3.26) Το μήκος του μηνύματος *Encrypted Handshake Message* από την πλευρά του πελάτη είναι 40 bytes.

3.27) Ναι, στάλθηκαν οι εγγραφές με μηνύματα *Change Cipher Spec* και *Encrypted Handshake Message* από την πλευρά του εξυπηρετητή.

3.28) Σύμφωνα με το *Wireshark*, μεταφέρονται δεδομένα με το πρωτόκολλο HTTP.

3.29) Όχι, δεν καταγράφηκαν εγγραφές TLS του πρωτοκόλλου Alert (Encrypted Alert).

3.30) (Δεν στάλθηκαν τέτοιες εγγραφές, όπως απαντήθηκε στο ερώτημα 3.29).

3.31) Στην περίπτωση του HTTP, η εύρεση της φράσης (ή και οποιασδήποτε άλλης) “BigBlueButton” είναι επιτυχημένη, ενώ στην περίπτωση του HTTPS είναι αποτυχημένη. Αυτό συμβαίνει επειδή, όπως είδαμε στην άσκηση αυτή, στην δεύτερη περίπτωση υπάρχει κρυπτογράφηση όλης της πληροφορίας που μεταδίδεται, και έτσι δεν είναι εμφανίσιμη σε οποιοδήποτε ενδιάμεσο μέσο, παρά μόνο στον πλοηγό.

3.32) Όπως είδαμε, το πρωτόκολλο HTTP δεν εφαρμόζει κάποια μέθοδο για ασφάλεια οποιουδήποτε τομέα, κάνοντας την υποκλοπή πληροφοριών δυνατή από οποιονδήποτε. Αντίθετα, το πρωτόκολλο HTTPS είναι πολύ αξιόπιστο, καθώς χρησιμοποιεί αλγορίθμους κρυπτογράφησης πληροφορίας σχεδόν σε όλα τα σημαντικά θέματα.