ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΡΟΗ Δ - ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ (COMPUTER NETWORKS)

ΓΕΩΡΓΑΚΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ 03120827

ΑΝΑΦΟΡΑ 12ΗΣ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ

Ομάδα: 2
Λογισμικο: Linux Ubuntu 20.04
Ονομα PC: glaptop
Διεύθυνση IP: 147.102.200.145
Διεύθυνση MAC: 70:9c:d1:03:b0:15

ΑΣΚΗΣΗ 1: Πιστοποίηση αυθεντικότητας στο πρωτόκολλο HTTP

1.1 Η απόκριση στο αρχικό αίτημα HTTP (GET) είναι:HTTP/1.1 401 Authorization Required Authorization Required

1.2 Το όνομα της σχετική επικεφαλίδας είναι: WWW-Authenticate και υποδεικνύει την μέθοδο Basic realm=”Edu-DY TEST”

1.3 Στο δεύτερο αίτημα HTTP τύπου GET το όνομα της σχετικής επικεφαλίδας για τα διαπιστευτήριά του πελάτη, είναι: Authorization

1.4 Η μέθοδος πιστοποίησης αυθεντικότητας και τα σχετικά διαπιστευτήρια εμφανίζονται ως εξής: Basic ZWR1LWR5OnBhc3N3b3Jk

1.5 Το αποτέλεσμα του DECODE είναι: edu-dy:password (Φαίνεται και στο wireshark)

1.6 Επομένως συμπεραίνω ότι η ασφάλεια του βασικού μηχανισμού πιστοποίησης αυθεντικότητας του HTTP είναι πολυ απλή και εύκολα μπορεί κάποιος να αποκρυπτογραφήσει τα δεδομένα που στέλνονται.

ΑΣΚΗΣΗ 2: Υπηρεσία SSH – Secure SHell

2.1 Το SSH χρησιμοποιεί ως πρωτόκολλο μεταφοράς το TCP

2.2 Οι θύρες του πρωτοκόλλου μεταφοράς που χρησιμοποιήθηκαν είναι: 51792 (Client - MyPC) και 22 (Server)

2.3 Η θύρα που αντιστοιχεί στο πρωτόκολλο εφαρμογής SSH είναι η 22.

2.4 Το φίλτρο σύληψης που χρησιμοποίησα είναι: ssh

2.5 Η έκδοση του πρωτοκόλλου SSH που χρησιμοποιεί ο πελάτης είναι: SSH-2.0 και η έκδοση λογισμικού είναι: OpenSSH_8.9p1. Στα σχόλια αναφέρονται τα εξής: Ubuntu-3ubuntu0.6

2.6 Η έκδοση του πρωτοκόλλου SSH που χρησιμοποιεί ο εξυπηρετητής είναι: SSH-2.0 και η έκδοση λογισμικού είναι: OpenSSH_6.6.1_hpn13v11. Στα σχόλια αναφέρονται τα εξής: FreeBSD-20140420

2.7 Το μήκος της συμβολοσειράς kex-algorithms είναι 305 bytes. Το πλήθος

των αλγορίθμων είναι 12 (Σύμφωνα με το Show Packet Bytes). Οι πρώτοι δύο είναι: curve25519-sha256 και curve25519-sha256@libssh.org.

2.8 Το πλήθος των αλγορίθμων παραγωγής κλειδιών είναι 14. Οι πρώτοι δυο είναι: ssh-ed25519-cert-v01@openssh.com και ecdsa-sha2-nistp256-cert-v01@openssh.com.

2.9 Το πλήθος των αλγορίθμων κρυπτογράφησης είναι 5. Οι πρώτοι δύο είναι: chacha20-poly1305@openssh.com και aes128-ctr,aes192-ctr.

2.10 Το πλήθος των αλγορίθμων πιστοποίησης αυθεντικότητας μηνυμάτων είναι 10. Οι δύο πρώτοι είναι: umac-64-etm@openssh.com και umac-128-etm@openssh.com.

2.11 Το πλήθος των αλγορίθμων συμνπίεσης είναι 3. Οι πρώτοι δύο είναι: none και zlib@openssh.com.

2.12 Ο αλγόριθμος που θα ακολουθήσουν τα 2 μέρη είναι ο curve25519-sha256@libssh.org. Αυτός φαίνεται και στο πεδίο Key Exchange

2.13 Ο αλγόριθμος παραγωγής κλειδιών που τελικά θα χρησιμοποιηθεί είναι ο ssh-ed25519. (Η επιλογή έγινε οπως και στο ερώτημα 2.12)

2.14 Ο αλγόριθμος κρυπτογράφησης που τελικά θα χρησιμοποιηθεί είναι ο chacha20-poly1305@openssh.com.

2.15 Ο αλγόριθμος πιστοποίησης αυθεντικότητας μηνυμάτων που θα χρησιμοποιηθεί είναι ο umac-64-etm@openssh.com

2.16 Ο αλγόριθμος συμπίεσης που θα χρησιμοποιηθεί είναι ο none

2.17 Σχετικά με την φάση παραγωγής κοινού μυστικού παρατήρησα και τους εξής τύπους μηνυμάτων:
Message Code: Elliptic Curve Diffie-Hellman Key Exchange Init (30)
Message Code: Elliptic Curve Diffie-Hellman Key Exchange Reply (31)
Message Code: New Keys (21)
Στην συνέχεια παρατηρώ μόνο Encrypted Packet

2.18 Ναι, οι αλγόριθμοι που επιλέχθηκαν φαίνονται και σε παρενθέσεις δίπλα στο SSH Version 2 (encryption:chacha20-poly1305@openssh.com mac <implicit> compression:none)

2.19 Οχι, δεν μπορώ να εντοπίσω τα πακέτα όπου μεταφέρεται η πληροφορία για την προτροπή login και password καθως όλα τα πακέτα που παρουσιάζει το wireshark είναι κρυπτογραφημένα. Μόνο τα 2 άκρα μπορούν να τα διαβάσουν

2.20 Η ασφάλεια της υπηρεσίας SSH σε σχέση με άλλα πρωτόκολλα ανταλλαγής δεδομένων είναι σαφώς υψηλοτερη. Χάρη στην κρυπτογράφηση των δεδομένων, την χρήση κλειδιών και την προσθήκη σύνοψης που παράγεται από τα περιεχόμενα του μηνύματος, έχουμε πιστοποίηση της αυθεντικότητας, εμπιστευτικότητα και ακεραιότητα των δεδομένων στην επικοινωνία μας.

ΑΣΚΗΣΗ 3: Υπηρεσία HTTPS

3.1 Το φίλτρο σύλληψης που χρησιμοποίησα είναι: host 147.102.222.246
(Η IPv4 διεύθυνση του www.noc.ntua.gr βρέθηκε με την εντολή ping)

3.2 Το φίλτρο απεικόνισης που χρησιμοποίησα είναι:
tcp.flags.syn == 1 and tcp.flags.ack == 0

3.3 Οι συνδέσεις γίνονται στις (πασίγνωστες) θύρες: 80 και 443

3.4 Η θύρα 80 αντιστοιχεί στο πρωτόκολλο http και η 443 στο https

3.5 Μεταξύ του υπολογιστή μου και του εξυπηρετητή ιστού ανοίχθηκαν:
2 συνδέσεις στην περίπτωση http
8 συνδέσεις στην περίπτωση https

3.6 Οι θύρες πηγής που καταγράφηκαν για την περίπτωση https είναι:
42060 / 42076 / 42090 / 42100 / 42102 / 42104 / 33288 / 33294

3.7 Τα τρία πρώτα πεδία που είναι κοινά στις επικεφαλίδες Στρώματος Εγγραφών TLS είναι:
Content Type (1 byte) / Version (2 bytes) / Length (2 bytes)

3.8 Σχετικά με το Content Type παρατηρούμε τους εξής διαφορετικούς τύπους εγγραφών:
Handshake (22) / Change Cipher Spec (20) / Application Data (23) / Alert (21)

3.9 Οι διαφορετικοί τύποι μηνυμάτων χειραψίας που παρατήρησα είναι:
Client Hello (1) / Server Hello (2) / Certificate(11)/
Server Key Exchange (12)/ Server Hello Done (14)/
Client Key Exchange (16) /Encrypted Handshake Message

3.10 Ο πελάτης έστειλε 8 μηνύματα Client Hello (1 για κάθε TCP σύνδεση)

3.11 Στην εγγραφή TLS με το πρώτο μήνυμα Client Hello που στέλνει ο πελάτης δηλώνεται η έκδοση: Version: TLS 1.0 (0x0301)

3.12 Στο πρώτο μήνυμα Client Hello που στέλνει ο πελάτης δηλώνεται η εξής έκδοση του πρωτοκόλλου TLS: Version: TLS 1.2 (0x0303. Οχι δεν ειναι ταυτόσημη με αυτή στην εγγραφή TLS

3.13 Το μήκος σε byte του τυχαίου αριθμού (Random) που περιέχει το μήνυμα Client Hello είναι: 32 bytes. Τα πρώτα 4 byte είναι: ab 99 64 70

3.14 Το πλήθος των σουίτων κωδικών που υποστηρίζει ο πελάτης είναι 16. Οι πρώτες 2 από αυτές είναι: 0x4a4a / 0x1301

3.15 Ο πλοηγός μου είναι συμβατός με την έκδοση TLS1.3 του πρωτοκόλλου και στην αντίστοιχη επικεφαλίδα δηλώνονται 3:
Reserved (GREASE) (0xeaea) / TLS 1.3 (0x0304) / TLS 1.2 (0x0303).

3.16 Ο πλοηγός μου είναι συμβατός με ε HTTP/2 και στην αντίστοιχη επικεφαλίδα δηλώνονται τα πρωτόκολλα:
ALPN Next Protocol: h2 / ALPN Next Protocol: http/1.1

3.17 Με βάση την επικεφαλίδα της εγγραφής TLS θα χρησιμοποιηθεί η έκδοση:
Version: TLS 1.2 (0x0303)

3.18 Το μήκος σε byte του τυχαίου αριθμού (Random) που περιέχει το μήνυμα

Server Hello είναι: 32 bytes. Τα πρώτα 4 byte είναι: 30 41 be 52. Συγκρίνοντας με την ερώτηση 3.13 παρατηρώ ότι δεν συνδέονται (αρα η παραγωγή τους είναι τυχαία)

3.19 Το όνομα και η δεκαεξαδική τιμή της σουίτας κωδίκων που τελικά επιλέχθηκε είναι:
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)

3.20 Απο την σουίτα αυτή συμπεραίνουμε για τους αλγόριθμους που θα χρησιμοποιηθούν ότι:
Ανταλλαγή κλειδιών: ECDHE
Πιστοποίησης ταυτότητας: RSA
Κρυπτογράφησης: GCM
Συνάρτηση κατακερματισμού: SHA

3.21 Οχι δεν χρησιμοποιείται κάποια μέθοδος συμπίεσης:
Compression Method: null (0)

3.22 Σχετικά με το μήνυμα Certificate που μεταφέρει τα πιστοποιητικά του εξυπηρετητή, το μήκος του είναι: Length: 6209 (σύμφωνα με το πεδίο length της επικεφαλίδας στρώματος εγγραφών TLS)

3.23 Μεταφέρονται 2 πιστοποιητικά με μήκος 6202 το κάθε ενα. (6209-7 λόγω των Handshake Type, Length, Certifiacates Length)

3.24 Για να μεταφερθεί η παραπάνω εγγραφή TLS χρειάστηκαν 5 πλαίσια Ethernet ([5 Reassembled TCP Segments])

3.25 Μήκος του δημόσιου κλειδιού που αποστέλλει ο πελάτης: 65 bytes
Μήκος του δημόσιου κλειδιού που αποστέλλει ο εξυπηρετητής: 65 bytes
Τα 5 πρώτα γράμματα είναι:
Πελάτης: 04 55 b9 ed dc
Εξυπηρετητής: 04 64 eb 93 c6

3.26 Το μήκος της εγγραφής TLS τύπου ChangeCipherSpec είναι 6 bytes και το μήκος του αντίστοιχου μηνύματος είναι 1 byte
(Change Cipher Spec Message)

3.27 Το μήκος του μηνύματος EncryptedHandshakeMessage από την πλευρά του πελάτη είναι 40 bytes

3.28 Ναι, παρατήρησα τέτοια εγγραφή

3.29 Για μια εγγραφή της μορφής Application Data παρατηρώ ότι σύμφωνα με το Wireshark μεταφέρονται δεδομένα του πρωτοκόλλου http-over-tls ([Application Data Protocol: http-over-tls])

3.30 Οι εγγραφές TLS του πρωτοκόλλου Alert (Encrypted Alert) στάλθηκαν απο την πλευρά του εξυπηρετητή.

3.31 Μετά απο αυτές τις εγγραφές παρατηρώ την απόλυση των tcp συνδέσεων. Ουσιαστικά πρόκειται για TLS notifications, στην περίπτωση μας ότι θα διακοπούν οι συνδέσεις.

3.32 Ψάχνοντας να βρώ το πακέτο που μεταφέρει μια φράση με λατινικούς χαρακτήρες παρατηρώ ότι στο http μπορώ να την βρώ μέσω του wireshark, ομως στο https (http-over-tls) δεν μπορώ καθώς τα δεδομένα

κρυπτογραφούνται

3.33 Το https είναι ένα πολυ ασφαλές πρωτόκολλο καθώς χρησιμοποεί certificates (πιστοποίηση της αυθεντικότητας), κρυπτογραφεί τα δεδομένα (εμπιστευτικοτητα) και επίσης κάνει χρηση hash functions (ακεραιότητα των δεδομένων)