ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
Τμήμα Μηχανικών Πληροφοριακών Και Επικοινωνιακών Συστημάτων

Συγκριτική αξιολόγηση επιθέσεων MITM (Man In The Middle) σε Επικοινωνιακά Συστήματα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπουσα: **Ελισάβετ Κωνσταντίνου, Επίκουρη Καθηγήτρια**

Μικέδης Θεμιστοκλής

Σάμος,Ιούνιος 2021

Περίληψη

Η εργασία αυτή αναφέρεται σε διαφορετικούς τρόπους επίθεσης σε συστήματα ασφαλείας βασισμένα στη τεχνική Man-In-The-Middle (MITM). Κάθε επίθεση διαφοροποιείται αρκετά από τις υπόλοιπες, παρόλο που χρησιμοποιούν την ίδια τεχνική. Για παράδειγμα μια MITM επίθεση βασισμένη σε DNS Spoofing και μια MITM επίθεση στο TCP/IP πρωτόκολλο λειτουργούν με διαφορετικό τρόπο. Για την καλύτερη κατανόηση του τρόπου λειτουργίας τέτοιων επιθέσεων είναι αναγκαία η κατανόηση της βάσης και της λογικής της κρυπτογραφίας, καθώς και του τρόπου επικοινωνίας μεταξύ πελάτη-εξυπηρετητή. Αυτό καθίσταται δυνατό με την ανάλυση της διαδικασίας χειραψίας (Handshake process), την εισαγωγή στη σουίτα αλγορίθμων κρυπτογράφησης και στην αναγκαιότητα έκδοσης πιστοποιητικών.

Πιο συγκεκριμένα, το πρώτο κεφάλαιο είναι εισαγωγικό και το δεύτερο αναφέρεται στα είδη και στους αλγόριθμους κρυπτογράφησης RSA και AES. Επίσης, χρησιμοποιούμε τη βιβλιοθήκη OpenSSL για να απεικονίσουμε τη λειτουργία δημιουργίας κλειδιών, αλλά και τον τρόπο κρυπτογράφησης-αποκρυπτογράφησης αρχείων από τους αλγόριθμους αυτούς. Το τρίτο κεφάλαιο κάνει μια εισαγωγή στη γενική κατανόηση του Man-In-The-Middle και αναλύει τους τρόπους με τους οποίους πραγματοποιείται η επίθεση αυτή σε διάφορα πρωτόκολλα επικοινωνίας, ασφάλειας και μεταφοράς αρχείων. Μερικά από αυτά αποτελούν τα SSL/TLS , FTP, DNS. Το τέταρτο κεφάλαιο αποσκοπεί στην πρακτική υλοποίηση μιας επίθεσης MITM, αναλύοντας τα εργαλεία που χρησιμοποιήθηκαν κατά τη διάρκεια της, τα οποία συμπεριλαμβάνουν τα Ettercap,Wireshark,Nmap με λειτουργικό σύστημα Kali Linux. Τα εργαλεία αυτά περιέχουν πολλές διαφορετικές τεχνικές διεξαγωγής επιθέσεων και τρόπους σάρωσης του δικτύου, είτε αυτό αποτελεί τοπικό ή όχι. Στην παρούσα διπλωματική εργασία, η πρακτική υλοποίηση αποτελείται από μια ARP Poisoning επίθεση και λαμβάνει μέρος σε τοπικό δίκτυο.

Τέλος με τη συνολική χρήση των εργαλείων αυτών και την ολοκλήρωση της εργασίας, ο αναγνώστης θα είναι σε θέση να κατανοήσει και να διαχωρίζει τον τρόπο λειτουργίας των μεθόδων κρυπτογράφησης, τον λόγο που προτιμήθηκαν και τον τρόπο που εφαρμόστηκαν. Επιπρόσθετα θα έχει αποκτήσει τις βασικές γνώσεις και ικανότητες που θα τον βοηθήσουν να κατηγοριοποιήσει τις επιθέσεις MITM, αλλά και να αναπαραστήσει μία από αυτές.

Abstract

This Bachelor thesis refers to different ways of attacking at security systems based on the Man-in-the-Middle (MitM) technique. Every attack differs from one to another, for example a Man-in-the-middle-attack using DNS Spoofing method or a MitM attack at TCP/IP protocol. To better understand how such attacks work, it is necessary to understand how cryptography functions, as well as the creation of communication between client-server. This is made possible by the analysis of the handshake process, the introduction of encryption algorithms into the suite and the need of creating certificates. More specifically, the first chapter is introductory and the second one refers to the cipher suites and the mathematical background of RSA and AES encryption algorithms. Moreover, in the second chapter we make use of the OpenSSL library to illustrate the key generation function, but also how to encrypt-decrypt files from these particular algorithms (RSA-AES). The third chapter is introductory to the general understanding of MitM and analyses many different methods of this attack in various communication, security and file transfer protocols. Some of them are SSL/TLS, FTP, DNS. The last chapter comprises the practical implementation of this Bachelor thesis by using an ARP Poisoning attack on a local network, which requires different tools of Kali Linux operating system. These tools are Ettercap, Wireshark and Nmap and they are highly recommended for the recreation of this attack. With the completion of this thesis, the reader will be able to understand how to distinguish these different techniques of MitM attacks and will have acquired the basic knowledge and skills to reproduce one of them.

Περιεχόμενα

Κεφάλαιο 1ο

Εισαγωγή

1.1 Εισαγωγικά Στοιχεία

Η κυβερνοασφάλεια(Cybersecurity) είναι ένα είδος ασφάλειας που εφαρμόζεται στην τεχνολογία της πληροφορίας(Information technology).Αυτό περιλαμβάνει οποιαδήποτε τεχνολογία που αποθηκεύει ή μετακινεί δεδομένα ,όπως υπολογιστές ,και γενικότερα οποιαδήποτε συσκευή είναι συνδεδεμένη σε ένα δίκτυο (π.χ. δρομολογητής, μεταγωγέας). Το πεδίο της κυβερνοασφάλειας γίνεται όλο και πιο σημαντικό λόγω της αναγκαίας εμπιστοσύνης στα συστήματα υπολογιστών, στο διαδίκτυο ,στα πρότυπα ασύρματου δικτύου(Wi-fi,Bluetooth) καθώς και λόγω της ραγδαίας ανάπτυξης ¨έξυπνων¨ συσκευών (smartphones,smart TV's) και των συσκευών που αποτελούν το Internet of Things (IoT). Η ασφάλεια στον κυβερνοχώρο αποτελεί την πρακτική μέθοδο αποτροπής ψηφιακών επιθέσεων σε αυτά. Αυτές οι επιθέσεις στοχεύουν συνήθως στη πρόσβαση ,την αλλαγή ή την καταστροφή ευαίσθητων πληροφοριών και πραγματοποιούνται κυρίως από crackers. Επίσης υπάρχουν άτομα που εξασκούν ethical hacking και στοχεύουν στην βελτιστοποίηση του συστήματός με τη χρήση παρόμοιων τρόπων εισβολής με των crackers. Οι επιθέσεις αυτές διαφέρουν μεταξύ τους και ποικίλουν όσον αφορά τις τακτικές που χρησιμοποιούνται.

Για παράδειγμα, οι επιθέσεις Denial of Service (DoS) και Distributed Denial of Service(DdoS) όπου έχουν ως στόχο την άρνηση πρόσβασης των νόμιμων χρηστών στους πόρους τους,SQL Injection που έχει ως στόχο την παράκαμψη των αλγορίθμων σύνδεσης (login algorithms) για να σαμποτάρει δεδομένα[17]. Τα παραπάνω παραδείγματα που αναφέραμε αντιπροσωπεύουν κάποιες από τις απειλές σε web διακομιστές. Επιπρόσθετα αξίζει να αναφέρουμε μια από τις πιο συνηθισμένες επιθέσεις που ονομάζεται Phising. Σε αυτή τη περίπτωση αποστέλλονται ψευδή e-mail που μοιάζουν να έχουν σταλθεί από αξιόπιστες πηγές. Στόχος τους είναι να κλέψουν ευαίσθητα στοιχεία όπως αριθμούς πιστωτικών καρτών και στοιχεία σύνδεσης. Συνεπώς η ασφάλεια στον κυβερνοχώρο είναι πολύ σημαντική διότι περιλαμβάνει όλα όσα σχετίζονται με προστασία ευαίσθητων δεδομένων, προσωπικά αναγνωρίσιμων πληροφοριών(PII), προστατευόμενων πληροφοριών για την υγεία(PHI), πνευματική ιδιοκτησία, κυβερνητικά και βιομηχανικά συστήματα πληροφοριών.

1.2 Στόχοι Εργασίας

Για να γίνει κατανοητός ο τρόπος λειτουργίας των επιθέσεων MitM πρέπει ο αναγνώστης να γνωρίζει τις βάσεις της κρυπτογραφίας και τον τρόπο λειτουργίας της. Για το λόγο αυτό ως αρχικός στόχος είναι η κατανόηση των τρόπων κρυπτογράφησης σε βάθος , αναλύοντας το μαθηματικό υπόβαθρο βασικών αλγορίθμων που χρησιμοποιούν τα πρωτόκολλα επικοινωνίας και κρυπτογράφησης(RSA,AES).Η ανάλυση αυτή περιέχει αντίστοιχα παραδείγματα καθώς και εφαρμογή δημιουργίας κλειδιών, κρυπτογράφηση-αποκρυπτογράφηση αρχείων με βιβλιοθήκη OpenSSL(χρησιμοποιείται από το SSL/TLS πρωτόκολλο).Έπειτα, η εργασία αυτή αναλύει τη σημαντικότητα της χρήσης πιστοποιητικών και αναφέρει τη χρησιμότητα της σουίτας αλγορίθμων(Cipher Suite) η οποία διαφέρει σε κάθε πρωτόκολλο. Για να ολοκληρωθούν οι βασικές προαπαιτούμενες γνώσεις για την δημιουργία επικοινωνίας μεταξύ πελάτη-εξυπηρετητή η εργασία αυτή αναλύει τη διαδικασία χειραψίας(Handshake Process) χρησιμοποιώντας ως παράδειγμα το πρωτόκολλο SSL/TLS. Όταν έχουν γίνει κατανοητές οι παραπάνω διαδικασίες, έχουν εδραιωθεί οι βασικές γνώσεις για να ξεκινήσει η ανάλυση των επιθέσεων Man In The Middle. Οι επιθέσεις αυτές είναι διαφορετικές μεταξύ τους, και η επιλογή τους διαφέρει ανάλογα με το σκοπό της επίθεσης (π.χ. Spoofing) Αυτές οι διαφορετικές τεχνικές που αναγράφονται αποτελούν τις πιο διαδεδομένες και συχνές επιθέσεις σε επικοινωνιακά συστήματα. Αξίζει να αναφερθεί ότι δεν αναλύονται εκτενώς επιθέσεις σε Web Services εκτός του HTTP. Τέλος, καταλήγουμε στο πρακτικό κομμάτι της εργασίας το οποίο αποτελεί μια ARP Poisoning επίθεση σε τοπικό ενσύρματο δίκτυο. Για την επιτυχή περάτωση της επίθεσης αυτής χρησιμοποιούνται τα εργαλεία Ettercap,Nmap και Wireshark, τα οποία παρέχονται δωρεάν από το Kali Linux λογισμικό. Η επίθεση αυτή επιτρέπει την υποκλοπή στοιχείων από ιστότοπο που χρησιμοποιεί το HTTP πρωτόκολλο. Κατά την ολοκλήρωση της εργασίας αυτής, ο αναγνώστης θα έχει εκλάβει τις απαραίτητες γνώσεις και ικανότητες για να αναπαραστήσει την επίθεση αυτή. Επίσης θα υπάρχουν οι απαραίτητες γνώσεις και βάσεις για την επέκταση και υλοποίηση άλλων επιθέσεων εκτός αυτής που υλοποιήθηκε στην εργασία, διότι τα εργαλεία που χρησιμοποιήθηκαν διαθέτουν και άλλες δυνατότητες και εφόσον ο αναγνώστης θα έχει εξοικειωθεί με αυτά, μακροπρόθεσμα θα μπορεί να τα αξιοποιήσει και για άλλου τύπου επιθέσεων εκτός της MitM.

1.3 Δομή Εργασίας

Τα κεφάλαια της διπλωματικής εργασίας αυτής αναλύονται ως εξής:

Κεφάλαιο 2ο :
Στο 2ο κεφάλαιο αναφερόμαστε στα είδη της κρυπτογραφίας(Ασύμμετρη,Συμμετρική), καθώς και στους βασικούς αλγόριθμους κρυπτογράφησης(RSA,AES). Η ανάλυση αυτών περιέχει το μαθηματικό υπόβαθρο του RSA αλγόριθμου για τη διαδικασία δημιουργίας κλειδιών(Public key,Private Key) και στην μέθοδο κρυπτογράφησης του AES. Επίσης για την καλύτερη κατανόηση της μεθοδολογίας κρυπτογράφησης από αυτούς τους αλγόριθμους χρησιμοποιείται η βιβλιοθήκη OpenSSL του πρωτόκολλου SSL/TLS με την οποία πραγματοποιείται η δημιουργία κλειδιών και η κρυπτογράφηση-αποκρυπτογράφηση αρχείου. Επίσης, στο κεφάλαιο αυτό περιλαμβάνεται η ανάλυση της σουίτας αλγορίθμων, η αναγκαιότητα της χρήσης πιστοποιητικών και η δημιουργία αυτών χρησιμοποιώντας ως παράδειγμα τον αλγόριθμο ECDSA.

Κεφάλαιο 3ο :
Το 3ο κεφάλαιο αναλύει τις επιθέσεις Man-in-the-Middle σε πρωτόκολλα ασφάλειας, επικοινωνίας και μεταφοράς πακέτων. Οι επιθέσεις αυτές αποτελούνται από διαφορετικές μεθόδους Spoofing(DNS,ARP,IP,Blind/No Blind,ICMP,SNP,DHCP), επιθέσεις στο HTTP πρωτόκολλο στο οποίο περιέχονται διαφορετικές μεθοδολογίες όπως Cookie insertion, cache poisoning, HTTPS Stripping/Splitting. Στη συνέχεια αναγράφονται επιθέσεις στο FTP και SSH πρωτόκολλο που αφορούν μεθόδους Sniffing,Binary Patching και Session Hijacking.Το τέλος του κεφαλαίου αυτού αποτελείται από την ανάλυση επίθεσης Man In The Middle στο SSL/TLS πρωτόκολλο χρησιμοποιώντας τη μέθοδο Renegotiation η οποία αφορά τη διαδικασία χειραψίας όπου αναλύεται διεξοδικά.

Κεφάλαιο 4ο :
Το 4ο κεφάλαιο αφορά αποκλειστικά την διεξαγωγή υλοποίησης επίθεσης Man In The Middle με τη μέθοδο ARP Poisoning. Για την επίθεση αυτή χρησιμοποιήθηκαν τα εργαλεία Ettercap,Nmap,Wireshark που παρέχονται από τα Kali Linux. Επίσης πραγματοποιείται ανάλυση των εργαλείων αυτών, αλλά και εναλλακτικά εργαλεία που μπορούν να χρησιμοποιηθούν. Στη συνέχεια εξηγείται ο λόγος που προτιμήθηκαν αλλά και τα βήματα που ακολουθούνται για την διαδικασία εφαρμογής της επίθεσης αυτής. Με την επιτυχημένη διεξαγωγή της επίθεσης και την υποκλοπή στοιχείων, χρησιμοποιούνται διαγράμματα για την εμφάνιση αποτελεσμάτων από αρχείο τύπου .pcap δημιουργημένο μέσω του Wireshark.

Κεφάλαιο 2ο

Βασικοί Κρυπτογραφικοί Αλγόριθμοι

2.1 Ασύμμετρη/ Συμμετρική Κρυπτογραφία

Η κρυπτογράφηση των δεδομένων είναι απαραίτητη για την ασφαλή επικοινωνία στο διαδίκτυο. Αν τα δεδομένα αυτά δεν έχουν κρυπτογραφηθεί τότε μπορούν να κλαπούν και να αναλυθούν από τον καθένα. Για την αποτροπή αυτής της πιθανότητας έχουν δημιουργηθεί κάποιες μέθοδοι κρυπτογράφησης. Η πιο ασφαλής μέθοδος ονομάζεται ασύμμετρη κρυπτογραφία(Asymmetrical Cryptography) και χρειάζεται δύο κλειδιά(Public key,Private key) ,τα οποία αποτελούν κομμάτια πληροφοριών(συνήθως πολύ μεγάλοι αριθμοί), για να λειτουργήσει σωστά. Από τα δύο αυτά κλειδιά το δημόσιο κλειδί χρησιμοποιείται για την κρυπτογράφηση και αντιστοίχως το ιδιωτικό κλειδί για την αποκρυπτογράφηση. Σκεφτείτε το δημόσιο κλειδί ως πληροφορίες σχετικά με τη θέση ενός κλειδωμένου γραμματοκιβωτίου με μια υποδοχή στο μπροστινό μέρος και το ιδιωτικό κλειδί ως το κλειδί που ξεκλειδώνει το γραμματοκιβώτιο. Όποιος γνωρίζει πού βρίσκεται το γραμματοκιβώτιο μπορεί να βάλει ένα μήνυμα σε αυτό. Αλλά για οποιονδήποτε άλλο να το διαβάσει, χρειάζονται το ιδιωτικό κλειδί.

Επειδή η ασύμμετρη κρυπτογραφία περιλαμβάνει δύσκολα μαθηματικά προβλήματα, απαιτεί πολλούς υπολογιστικούς πόρους. Αυτοί οι πόροι είναι τόσο πολλοί που αν κρυπτογραφούσαμε όλες τις πληροφορίες σε μια συνεδρία επικοινωνίας, η λειτουργία του υπολογιστή καθώς και η σύνδεση θα σταματούσε. Το TLS αντιμετωπίζει αυτό το πρόβλημα χρησιμοποιώντας μόνο ασύμμετρη κρυπτογραφία στην αρχή μιας συνεδρίας επικοινωνίας για να κρυπτογραφήσει τη συνομιλία. Ο διακομιστής και ο πελάτης(server-client) πρέπει να συμφωνήσουν για ένα μόνο κλειδί συνεδρίας που θα χρησιμοποιούν και οι δύο για να κρυπτογραφήσουν τα πακέτα που θα προωθηθούν. Η κρυπτογράφηση που χρησιμοποιεί ένα κοινόχρηστο κλειδί ονομάζεται συμμετρική(Symmetrical Cryptography) και είναι πολύ λιγότερο εντατική υπολογιστικά από την ασύμμετρη. Επειδή το κλειδί της συνεδρίας δημιουργήθηκε χρησιμοποιώντας ασύμμετρη κρυπτογραφία, η επικοινωνία της στο σύνολο είναι πολύ πιο ασφαλής από ότι θα ήταν διαφορετικά.

2.1

2.2 Πιστοποιητικά –Σουίτα Αλγορίθμων-ECDSA

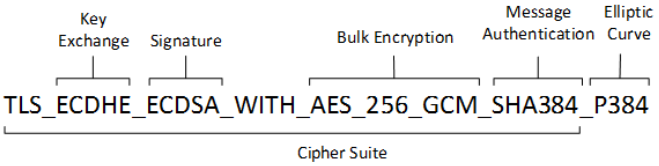
Το Digital Certificate είναι απαραίτητο για την έναρξη ασφαλών συνδέσεων διότι παρέχει το δημόσιο κλειδί στον πελάτη που είναι αναγκαίο για την κρυπτογράφηση. Επίσης όπως αναφέραμε στη προηγούμενη ενότητα, άλλος ένας λόγος που είναι τόσο σημαντικό είναι επειδή καθιστά δυνατό να πιστοποιηθεί ότι το δημόσιο κλειδί συνδέεται με έναν πραγματικό οργανισμό που προσφέρει το πρωτόκολλο αυτό στον πελάτη.Τα πιστοποιητικά αυτά εκδίδονται από αρχές πιστοποιητικών (Certificate Authorities-CA) και μπορούν να παρομοιαστούν με ένα διαβατήριο. Οι οργανισμοί που θέλουν να προσφέρουν υπηρεσίες κρυπτογραφημένες μέσω του TLS πρέπει να αγοράσουν πιστοποιητικά από τα CA, οι οποίες με τη σειρά τους επαληθεύουν την ταυτοποίηση των οργανισμών. Για παράδειγμα , εάν κάποιος θέλει να αγοράσει ένα πιστοποιητικό για την ασφάλεια σε έναν ιστότοπο example.com, θα πρέπει να λάβει ορισμένα μέτρα για να αποδείξει ότι αυτό το domain του ανήκει. Με αυτόν τον τρόπο, εάν κάποιος συνδεθεί στο example.com και λάβει έγκυρο SSL πιστοποιητικό που έχει εκδοθεί από μια αξιόπιστη αρχή έκδοσης πιστοποιητικών, μπορεί να είναι βέβαιος ότι επικοινωνεί με τον νόμιμο κάτοχο του example.com. Αυτή η διαδικασία μπορεί να αποτρέψει μια man-in-the-middle επίθεση.

Αξίζει να σημειωθεί ότι χρησιμοποιήσαμε παραπάνω τη φράση «αξιόπιστη αρχή». Ο καθένας μπορεί να δημιουργηθεί ως αρχή έκδοσης πιστοποιητικών. Είναι δύσκολο να καταλάβουμε ποιος έχει την εξουσία για τον έλεγχο ταυτότητας των πελατών. Η διαδικασία αντιμετώπισης αυτού του προβλήματος πραγματοποιείται κυρίως από τους κατασκευαστές λογισμικού. Το ίδρυμα Mozilla διατηρεί μια λίστα CA που θα εμπιστευτεί ο Firefox. Η Apple και η Microsoft διατηρούν επίσης λίστες που εφαρμόζουν σε επίπεδο λειτουργικού συστήματος για Windows, macOS και iOS, τις οποίες χρησιμοποιεί το Chrome σε αυτές τις πλατφόρμες. Επίσης το πρότυπο που ορίζει τα πιστοποιητικά ονομάζεται X.509 και επιτρέπει στα πιστοποιητικά να μεταφέρουν πολλές πληροφορίες εκτός του δημόσιου κλειδιού και της επιβεβαιωμένης ταυτότητας του κατόχου του πιστοποιητικού.

Τα Cipher Suite είναι αλγόριθμοι που αποτελούν ένα σύνολο βημάτων που απαιτούνται για την εκτέλεση μιας κρυπτογραφικής λειτουργίας (π.χ. κρυπτογράφηση, αποκρυπτογράφηση, κατακερματισμός, ψηφιακές υπογραφές). Σήμερα οι αλγόριθμοι αυτοί εξαρτώνται από τις πλέον προηγμένες δυνατότητες επεξεργασίας των υπολογιστών. Παλαιότερα δεν υπήρχαν αυτές οι επιλογές, ωστόσο η λογική της κρυπτογράφησης είναι η ίδια , δηλαδή να μπορέσει ένα μήνυμα να κρυπτογραφηθεί με έναν τρόπο τέτοιο ώστε να μπορεί να αποκρυπτογραφηθεί μόνο από αυτόν ή αυτούς που προορίζεται.

Όπως αναφέραμε, οι αλγόριθμοι αυτοί είναι ένα σύνολο βημάτων που χρησιμοποιούνται για την εκτέλεση μιας συγκεκριμένης μαθηματικής λειτουργίας. Η κρυπτογράφηση που πραγματοποιούν οι αλγόριθμοι αυτοί χρησιμοποιούν κλειδιά(public key, private key, shared key). Ανάλογα λοιπόν με τον αλγόριθμο κρυπτογράφησης που χρησιμοποιείται , είτε οι τιμές που βρίσκονται εντός του αλγόριθμου είτε αυτές που εισέρχονται στον αλγόριθμο αποτελούν κλειδιά.

Κατά τη διάρκεια της handshake διαδικασίας ο πελάτης ζητά από τον διακομιστή να του αποστείλει τη λίστα που κατέχει σχετικά με τα Cipher Suites που υποστηρίζει η εκδοχή του.Γενικότερα υπάρχουν αρκετοί συνδυασμοί αλγόριθμων κρυπτογράφησης οι οποίοι αποτελούνται από τους Key Exchange Algorithms (RSA, DH, DHE, ECDHE, PSK), Authentication/DigitalSignatureAlgorithms (RSA, ECDSA, DSA), BulkEncryptionAlgorithms (AES, CHACHA20, Camellia, ARIA) και Message Authentication Code Algorithms (SHA-256, POLY1305). Στο παρακάτω παράδειγμα θα χρησιμοποιήσουμε την 1.2 εκδοχή του TLS με μια Cipher Suite για να κατανοήσουμε καλύτερα το πώς διαχωρίζονται μεταξύ τους οι κάθε είδους αλγόριθμοι.



2.2

Γενικότερα εμείς θεωρούμε ότι χρησιμοποιούμε ως παράδειγμα την TLS 1.2 έκδοση η οποία υπάρχει περίπου δέκα χρόνια και υποστηρίζει 37 διαφορετικές Cipher Suites. Επίσης από τη στιγμή που υποστηρίζονται τόσοι πολλοί αλγόριθμοι, αυτομάτως υπάρχουν πάρα πολλοί συνδυασμοί που μπορούν να χρησιμοποιηθούν, όπως παρατηρήσαμε στην εικόνα 1.3.Υπάρχουν τόσες επιλογές που αξίζουν να αναφερθούν, αλλά εμείς θα ασχοληθούμε με τους αλγόριθμους RSA(Key Exchange Algorithm), AES(Bulk Encryption Algorithm) και ECDSA (Authentication-Digital Signature Algorithm).

Ο ECDSA(Elliptic Curve Digital Signature Algorithm) αποτελεί έναν Authentication/Signature αλγόριθμο ο οποίος έχει πλέον αντικαταστήσει τον DSA. Γενικώς, ο αλγόριθμος αυτός λειτουργεί πολύ παρόμοια με τον RSA(για τη συγκεκριμένη λειτουργία), αλλά με μία σημαντική διαφορά. Ενώ και οι δύο μέθοδοι ελέγχουν το πιστοποιητικό με τον ίδιο τρόπο, όταν η Diffie Hellman χρησιμοποιείται, το πραγματικό τμήμα ανταλλαγής κλειδιών δεν μπορεί να χρησιμοποιηθεί για να αποδείξει την κατοχή του ιδιωτικού κλειδιού. Αντιθέτως, ο διακομιστής παίρνει τα δύο τυχαία (client και server) καθώς και τις παραμέτρους που έχει επιλέξει (pre-master key) και τις κρυπτογραφεί με το ιδιωτικό κλειδί που κατέχει. Το αποτέλεσμα της κρυπτογράφησης αυτής χρησιμεύει ως ψηφιακή υπογραφή του. Ο πελάτης θα χρησιμοποιήσει το δημόσιο κλειδί για να επαληθεύσει την υπογραφή , και επομένως την ιδιοκτησία του δημόσιου κλειδιού.

Αναφέραμε παραπάνω ότι ECDSA έχει αντικαταστήσει τον DSA. Ο Digital Signature Algorithm DSA έχει αφαιρεθεί εντελώς από το TLS 1.3 και υπάρχουν αρκετές συζητήσεις σχετικά με το πόσο ασφαλές είναι ακόμα . Ο κύριος λόγος όμως που αφαιρέθηκε ήταν το μέγεθος των κλειδιών που χρησιμοποιούσε. Χρησιμοποιεί κλειδιά αντίστοιχου μεγέθους με του RSA (1024-bit,2048-bit,3096-bit) όπου καλύψαμε ότι είναι αρκετά ακριβά ως προς τον υπολογισμό τους. Αντιθέτως το ECDSA χρησιμοποιεί κλειδιά πολύ μικρότερου μεγέθους σε bit (224-bit,256-bit).

2.3 RSA-AES

2.3.1 Μαθηματικό υπόβαθρο RSA

Ο RSA πήρε το όνομα του από τους δημιουργούς του (Rivest,Shamir,Adleman) και αποτελεί το πιο κοινό ασύμμετρο κρυπτογραφικό αλγόριθμο. Στο SSL/TLS ο RSA χρησιμοποιείται συνήθως στο πλαίσιο ανταλλαγής κλειδιών και βασίζεται στη νοοτροπία του ότι είναι πιο εύκολο να πολλαπλασιάσεις μεγάλους αριθμούς και στο ότι είναι πιο δύσκολο να βρούμε από ποιο γινόμενο αποτελείται αυτός ο αριθμός. Για παράδειγμα 31*37=1147 αλλά 1147= x*y. Συνεπώς σε πιο μεγάλους αριθμούς υπάρχουν άπειροι συνδυασμοί γινομένων , κάτι που καθιστά πολύ δύσκολο να βρεθούν τα σωστά x και y.

Στον RSA, το δημόσιο κλειδί δημιουργείται με τον πολλαπλασιασμό δύο πρώτων αριθμών p και q (π.χ. 2,3,5,7,11).Το ιδιωτικό κλειδί αντίστοιχα δημιουργείται μέσω μιας διαφορετικής διαδικασίας που αφορά και αυτή τα p και q.

Όσον αφορά το μαθηματικό υπόβαθρο που χρειάζεται για την κατανόηση του RSA , πρέπει να αναφέρουμε ότι χρησιμοποιεί πολύ modular αριθμητική , το Θεώρημα Euler καθώς και την Euler's totient function ή αλλιώς phi function(ϕ).

Σχετικά με τα βήματα που ακολουθά ο RSA , ξεκινάει με τον δέκτη, ο οποίος διαλέγει δύο prime αριθμούς p και q , με το αποτέλεσμα του γινομένου τους να είναι n= p*q , όπου n αποτελεί ουσιαστικά το μισό δημόσιο κλειδί. Έπειτα υπολογίζει το Φ(p*q)=(p-1)(q-1) και διαλέγει έναν αριθμό e ο οποίος είναι σχετικά prime(δηλαδή ο μέγιστος κοινός διαιρέτης είναι ίσος με 1) με το ϕ(p*q). Στην πράξη ο αριθμός e έχει συνήθως εύρος ίσο με 2^{16} + 1, ωστόσο έχει υπάρξει και με εύρος ίσο με 3.Στη συνέχεια υπολογίζει το Modular Inverse d του e mod ϕ(n), δηλαδή με άλλα λόγια d*e ≡ 1 (mod ϕ(n)) , όπου d είναι το ιδιωτικό κλειδί. Τέλος, ο δέκτης αποστέλλει και τα δύο μέρη του δημόσιου κλειδιού(e και n) αλλά κρατά το ιδιωτικό (d) κρυφό.Από τη στιγμή που και τα δύο κλειδιά έχουν δημιουργηθεί , μπορούν να χρησιμοποιηθούν όσες φορές χρειαστεί[6].

Για να κατανοήσουμε πως μεταδίδεται ένα μήνυμα, θα χρησιμοποιήσουμε ως παράδειγμα μια γνωστή και πολύ συχνή μετατροπή που χρησιμοποιεί το αλφάβητο ASCII.

A	B	C	D	E	F	G	H	I	J	K	L	M
65	66	67	68	69	70	71	72	73	74	75	76	77
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
78	79	80	81	82	83	84	85	86	87	88	89	90

2.3

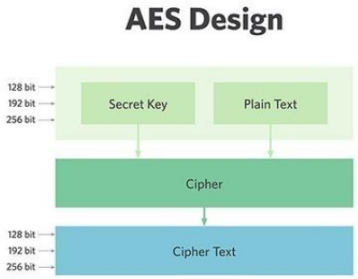
Για αρχή , ο αποστολέας μετατρέπει το μήνυμα του σε έναν αριθμό m. Για παράδειγμα το μήνυμα “HELLO” θα ήταν αντίστοιχο του 7269767679. Είναι πολύ σημαντικό να ισχύει m < n, αλλιώς το μήνυμα θα χαθεί κατά τη διάρκεια της πράξης mod n. Αν n<m τότε το μήνυμα θα σταλεί σε κομμάτια. Στη συνέχεια ο αποστολέας υπολογίζει $c \equiv m^e \pmod n$, όπου c αποτελεί το ciphertext ή αλλιώς το κρυπτογραφημένο μήνυμα. Αξίζει να αναφερθεί ότι το ciphertext αποτελεί τη μόνη πληροφορία που μπορεί να κλαπεί από κάποιον επιτιθέμενο. Ο δέκτης με τη σειρά του όταν λάβει το μήνυμα αυτό , πραγματοποιεί την αποκρυπτογράφηση του με τη χρήση του ιδιωτικού κλειδιού d με την βοήθεια του θεωρήματος Euler και την αντίστοιχη πράξη $c^d \equiv m \pmod n$.Ας θεωρήσουμε ότι έχουμε n=187 (μισό δημόσιο κλειδί), d=107 (private key) και e=3(μισό δημόσιο κλειδί). Όπως αναφέραμε παραπάνω ο χρήστης θέλει να στείλει το μήνυμα “HELLO”. Από τη στιγμή που το n είναι τόσο μικρό, ο αποστολέας θα χρειαστεί να αποστείλει το μήνυμα του χαρακτήρα-χαρακτήρα. Το ‘H’ στην ASCII αντιστοιχεί στο νούμερο 72, άρα m=72. Έπειτα υπολογίζει το $m^e = 72^3 \equiv 183 \pmod{187}$, κάνοντας το ciphertext c=183. Εν συνεχεία ο δέκτης υπολογίζει $c^d = 183^{107} \equiv 72 \pmod{187}$, και έτσι λαμβάνει τη πληροφορία ότι m=72, άρα μεταγλωττίζει το 72 σε ‘H’. Τα υπόλοιπα γράμματα ακολουθούν ακριβώς την ίδια διαδικασία.Γενικά, λόγω της μεγάλης δυσκολίας να «σπάσει» ο RSA , χρησιμοποιείται σχεδόν οπουδήποτε απαιτείται κρυπτογράφηση(ανταλλαγή κωδικών πρόσβασης, τραπεζικές αλλαγές, ηλεκτρονικές αγορές , ακόμη και στην καλωδιακή τηλεόραση).Το RSA χρησιμοποιείται επίσης για να διασφαλίσει ότι οι ιστότοποι είναι νόμιμοι, καθώς μόνο ο πραγματικός ιστότοπος θα έχει το ιδιωτικό κλειδί του. Αποφεύγει επομένως τις επιθέσεις man-in-the-middle , στις οποίες ένας εισβολέας παρεμποδίζει μια σύνδεση και προβάλει στον χρήστη κάτι πειστικό αλλά σχεδόν εντελώς ψεύτικο. Συνεπώς μια ευπάθεια στον αλγόριθμο RSA θα είχε καταστροφικές συνέπειες για την ασφάλεια. Η δύναμη του RSA μετριέται σε μέγεθος κλειδιού, που είναι ο αριθμός των bits σε n=p*q. Πλέον, ο RSA των 512-bit(155 ψηφία) δεν θεωρείται ασφαλής , καθώς οι σύγχρονες επιθέσεις brute force μπορούν να εξαγάγουν ιδιωτικά κλειδιά σε λίγες μόνο ώρες. Επίσης το 2010 συνέβη παρόμοια επίθεση όπου κατάφεραν να εξαγάγουν ιδιωτικό κλειδί εύρους 768-bit(232 ψηφία). Από το 2016 και μετά, τα κλειδιά 1024-bit (309 ψηφία) θεωρούνται επικίνδυνα και αξίζει να αναφέρουμε ότι τα περισσότερα κλειδιά που δημιουργήθηκαν πρόσφατα είναι 4096-bit(1234 ψηφία).

Μια συχνή επίθεση στον RSA είναι αυτή που παρακάμπτει εντελώς τον αλγόριθμο. Ένας υπολογιστής μπορεί γρήγορα να υπολογίσει τον μέγιστο κοινό διαιρέτη δύο αριθμών χρησιμοποιώντας τον αλγόριθμο του Ευκλείδη. Ο εισβολέας μπορεί να εκτελέσει τον αλγόριθμο αυτό σε δύο δημόσια κλειδιά. Εάν ο μέγιστος κοινός διαιρέτης τους δεν είναι ίσος με 1, τότε ο εισβολέας έχει βρει έναν prime αριθμό που διαιρεί και τα δύο κλειδιά, συνεπώς καταφέρνει και τα «σπάει» ταυτόχρονα. Όσον αφορά την χρήση του RSA ως Authentication/Signature αλγόριθμο, ο πελάτης (κάποιες φορές και ο server αν χρησιμοποιείται SSL πιστοποιητικό από τον πελάτη) ελέγχει την αυθεντικότητα του πιστοποιητικού που του παρουσιάζεται εκτελώντας μια σειρά ελέγχων. Εξετάζει την αλυσίδα πιστοποιητικών ακολουθώντας τις ψηφιακές υπογραφές που εμπιστεύεται. Επίσης ελέγχει τις ημερομηνίες ισχύος και την κατάσταση ανάκλησης του πιστοποιητικού. Στη συνέχεια χρησιμοποιεί το συσχετισμένο δημόσιο κλειδί για να επαληθεύσει την υπογραφή του ιδιωτικού κλειδιού[9]. Η τελική επιβεβαίωση ότι ο server διαθέτει ιδιωτικό κλειδί πραγματοποιείται κατά την ανταλλαγή κλειδιών, όταν ο πελάτης κρυπτογραφεί το pre-master key και όταν ο server το αποκρυπτογραφεί αντίστοιχα με το ιδιωτικό κλειδί

2.3.2 Λειτουργία AES

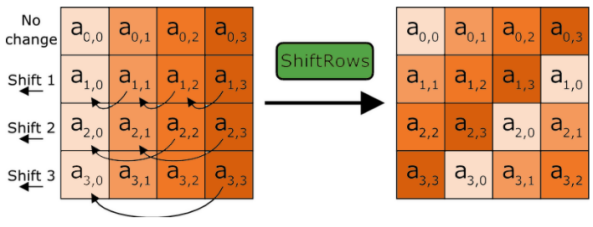
Το AES(Advanced Encryption Standard) ή αλλιώς Rijndael, αποτελεί έναν αλγόριθμο κρυπτογράφησης(Bulk Algorithm) ο οποίο έχει εγκριθεί από το NIST(National Institute of Standards and Technology) και κατέχει ένα block μεγέθους 128 bits, καθώς και συμμετρικά κλειδιά μήκους 128,192 ή 265 ψηφίων. Στην πραγματικότητα αποτελεί το πρώτο και μόνο διαθέσιμο στο κοινό αλγόριθμο που έχει εγκριθεί από την NSA για την κρυπτογράφηση δεδομένων. Η AES είναι διάδοχος του Data Encryption Standard (DES), το οποίο δημοσιεύτηκε για πρώτη φορά το 1977[10].

Το AES λειτουργεί με έναν αρκετά ενδιαφέρον τρόπο. Χρησιμοποιεί πίνακες block 4x4 οι οποίοι ονομάζονται «state». Όπως αναφέραμε παραπάνω η AES είναι ένα block κρυπτογράφησης με τα υπο-block του να αποτελούνται από 128 bits. Το μέγεθος των κλειδιών που χρησιμοποιεί εξαρτάται από τον αριθμό των γύρων (rounds) που θα χρειαστεί το κείμενο να περάσει για να κρυπτογραφηθεί. Οι επιλογές που υπάρχουν είναι 128-bit key= 10 Rounds , 192-bit key= 12 Rounds και 256-bit key =14 Rounds. Η παρακάτω εικόνα μας δείχνει απλά και περιεκτικά το σχέδιο του AES.



2.4

Ο αλγόριθμος κρυπτογράφησης AES καθορίζει πολλούς μετασχηματισμούς που πρόκειται να εκτελεστούν σε δεδομένα που είναι αποθηκευμένα σε έναν πίνακα. Ως πρώτο βήμα της διαδικασίας κρυπτογράφησης αποτελεί η τοποθέτηση των δεδομένων σε μια συστοιχία. Ως δεύτερο βήμα πραγματοποιούνται οι μετασχηματισμοί κρυπτογράφησης οι οποίοι επαναλαμβάνονται ανάλογα με τα rounds. Ο πρώτος μετασχηματισμός είναι η αντικατάσταση δεδομένων χρησιμοποιώντας έναν πίνακα αντικατάστασης . Ο δεύτερος μετασχηματισμός αλλάζει σειρές δεδομένων και ο τρίτος συνδυάζει τις στήλες. Ο τελευταίος μετασχηματισμός πραγματοποιείται σε κάθε στήλη χρησιμοποιώντας ένα διαφορετικό μέρος του κλειδιού κρυπτογράφησης. Τα μεγαλύτερα κλειδιά χρειάζονται περισσότερους γύρους για να ολοκληρωθούν.



2.5

Παρατηρούμε στο σχήμα 2.5 ότι οι γραμμές του πίνακα αυτού αλλάζουν διαφορετικά . Για παράδειγμα η πρώτη γραμμή δεν αλλάζει καθόλου, ενώ οι υπόλοιπες αλλάζουν με βήμα αντίστοιχο των 1 , 2 και 3. Για την αποκρυπτογράφηση των μηνυμάτων που έχουν κρυπτογραφηθεί με τον αλγόριθμο AES πρέπει να επαναληφθούν ακριβώς τα ίδια βήματα με την αντίθετη σειρά, έτσι ώστε να μετακινηθούν τα block στη θέση που ήταν εξαρχής, αλλιώς θα είναι αδύνατο να βρεθεί το σωστό μήνυμα.

2.4 Υλοποίηση Κρυπτογράφησης με OpenSSL

Για να υλοποιηθεί το πρακτικό κομμάτι της εργασίας, χρειάστηκε να εγκατασταθεί μια κρυπτογραφική βιβλιοθήκη που χρησιμοποιεί το πρωτόκολλο SSL/TLS. Η βιβλιοθήκη αυτή ονομάζεται OpenSSL και παρέχεται δωρεάν. Η έκδοση που χρησιμοποιήσαμε είναι η OpenSSL 1.1.1g και οι αλγόριθμοι κρυπτογράφησης που μας βοήθησαν να κρυπτογραφήσουμε δεδομένα είναι ο RSA και ο AES. Πρέπει να αναφέρουμε επίσης ότι η διαδικασία αυτή πραγματοποιείται μέσω της γραμμής εντολών (Command Prompt).
Αφότου εγκαταστήσαμε τη βιβλιοθήκη, επαληθεύουμε την εγκατάσταση της με την εντολή openssl version –a. Στη δική μας περίπτωση η έκδοση που εγκαταστάθηκε είναι η 1.1.1g .

2.4.1 Δημιουργία Κλειδιών

Όπως αναφέραμε στο προηγούμενο κεφάλαιο, το πρωτόκολλο SSL/TLS χρησιμοποιεί πολύ συχνά τον αλγόριθμο RSA για την ανταλλαγή κλειδιών. Ως πρώτο βήμα για την υλοποίηση της διαδικασίας, πρέπει να δημιουργήσουμε το ιδιωτικό κλειδί. Αυτό πραγματοποιείται με την εντολή openssl genrsa -out example.key 2048.
Το κομμάτι της εντολής genrsa ζητά να δημιουργηθεί το κλειδί με τη χρήση του αλγόριθμου RSA . Στη συνέχεια με τη χρήση του -out example.key δημιουργούμε ένα αρχείο που θα περιέχει το κλειδί αυτό. Τέλος, χρησιμοποιήσαμε τον αριθμό 2048 , οποίος αναφέρεται στο μέγεθος που θα του αντιστοιχεί.
Το αποτέλεσμα της εντολής που χρησιμοποιήσαμε μας επαληθεύει ότι δημιουργήθηκε ένα ιδιωτικό κλειδί με μέγεθος 2048 bit.
Εάν ανοίξουμε τον φάκελο αυτό μπορούμε να δούμε το κρυπτογραφημένο κλειδί.

example.key

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAtXAfjlOaHuVURRnGjJ0s3hXNgFMcTnJcKgJNvhw/6mn6MZR4
ow4MEs5tCWQ6jd9QLNVa7B+FRB9CzjNiW9mwXhvy/bg9mXkjHrHjqIBVRfKIXjwz
jT724QuoxaBtW6qkSd9cT0RkjjXn4VqEfFlOAvkhud8zGbFrDty8AbR40bxxH6nK
lOvPzRmcrESgubiU8HFWeVCfG6vWdUXNJu3pH0fuFAlOAaArbIQwVQjUYuahY/Ff
PFkLXZTnn7P5R5Y/hiel4zv0uloW9dWtpYDGXlLYbFc822qijdLFNEKt5JkCtfzb
d/Dlkh3sMUsab/RrZpxJJZYxx9TKdC5O981z+wIDAQABAoIBAFYpIrMh/GSDnxsZ
fHI2CaKFVAuGbEDMifvhEC4hmDCnlwqIaSlleEJOCPWWxT+uNpKzlvOenFiXR9BI
EoNvwxNNznyyS35gSk8/Rs8z8kq/PAcUK579sZt5P/Jp1sUbKdlylpFF90KZUr5v
j87FUgiTanKsDcY7vIlYQWfF3kIqnwJpfOApqs3JoTIjPHRLcwejDgYDGZ3xd514
Gs0T0DKgEBTNRyoZBgoMj6eHhYghdQi85utMJJ4vHR+D9J1WRlo6GSCAURCIHS88
D94rA62kbqVfkqDIYPjoTmBcZYTSaixRdOqJZh9+Jp0ZFKwhcvXWUj4VNvY49NgR
TzzEVBECgYEA7rj6stIl7HfI2Yj840OI9Zr0FaB93sT6c4ud/FI6lPfaNYr8EyMF
iYvDMb+UjJ6wSwXWvUSOF08Duf21T8SaoCNT7WSjiPiRjZr9hT3rpI3zP5aWezYh
wxl5laPXFhEg6BBLnntlCF6m70OO2v8DCL2VKVg02DRJXpfuWWC0Rg0CgYEAwpHI
UdLpVTKh5HQrCOOdgbqdgDV8EnBdVwK60Jt2CW2f4BPS+oOgVv+l4DKFuf3Ee/UY
aiFbUKBLsXxHggi97JrMlgpWiWY32VSVJGvm6cdvldve/nrdXXIlw8Cy4L65eUUC
tjdxaXmDNeO4Dfplg5GjznrQs2ObH6YsQZyH6CcCgYBocXPlwq2m7KlqR13lDIoi
cp3xzFrlmSa+YMpJN6y7uki0/k7VDJCCoixPWrNiZpdY64ANJK5KEl7eA9tMwAaq
o2o3xb0q0x3TZ4kdv36WayiOnJLP6loQJJggWRlkZTARd/etqfQFKmw86EDbHZEl
RM565wWPxiiHg09buUyGDQKBgB5PGVZttXlZGtD8q6vgh4uqSlgOfNpeeJv5SAmw
/e0wD4SXcfkmQfbrdIiNNjmn5lCr+Han5ZKktcXbJ0SBWhyebJo+2AYlTzFisizv
RhRKxf4hNxaL5yqAWpOo0/SgoUHbizVNY7Rbgm4FoClFR8MMD+ldRl97d7GxcBl0
adNnAcGAZtNhe4iwDoiZNbXxJFxyr0mEqv3FamUl/ClRl+NNvZ3xUgJWWDevZSjw
cMUy/mdIrDwpmI29Xqyl1Q7D6UMZYI8GTP+KLCO04RUt86x5grGwOhunEtXYhXwB
iJLR5pvSl7nnzgqqzRoz3QlemgEfKl2W829jaGRctM/wZmnFTOw=
-----END RSA PRIVATE KEY-----
```

2.6

Για να μπορέσουμε στη συνέχεια να εξάγουμε το δημόσιο κλειδί από το ιδιωτικό χρησιμοποιούμε την εντολή openssl rsa -in example.key –pubout -out examplekey_public.key.
Ουσιαστικά, χρησιμοποιούμε μια εντολή του rsa η οποία χρησιμοποιεί ως είσοδο τον αρχείο example.key. Έπειτα με τη χρήση του –pubout εξάγει το δημόσιο κλειδί και δημιουργεί ένα αρχείο με όνομα examplekey_public που θα το περιέχει. Το περιεχόμενου του αρχείου αυτού παρατηρείται παρακάτω.

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtXAfjlOaHuVURRnGjJ0s
3hXNgFMcTnJcKgJNvhw/6mn6MZR4ow4MEs5tCWQ6jd9QLNVa7B+FRB9CzjNiW9mw
Xhvy/bg9mXkjHrHjqIBVRfKIXjwzjT724QuoxaBtW6qkSd9cT0RkjjXn4VqEfFlO
Avkhud8zGbFrDty8AbR40bxxH6nKlOvPzRmcrESgubiU8HFWeVCfG6vWdUXNJu3p
H0fuFAlOAaArbIQwVQjUYuahY/FfPFkLXZTnn7P5R5Y/hiel4zv0uloW9dWtpYDG
XlLYbFc822qijdLFNEKt5JkCtfzbd/Dlkh3sMUsab/RrZpxJJZYxx9TKdC5O981z
+wIDAQAB
-----END PUBLIC KEY-----
```

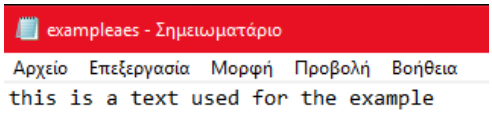
,

2.7

2.4.2 Κρυπτογράφηση-Αποκρυπτογράφηση

Όσον αφορά τη διαδικασία κρυπτογράφησης-αποκρυπτογράφησης χρησιμοποιούμε τον αλγόριθμο AES.

Ως πρώτο βήμα θα δημιουργήσουμε ένα αρχείο με όνομα exampleaes.txt το οποίο περιέχει μια τυπική πρόταση.

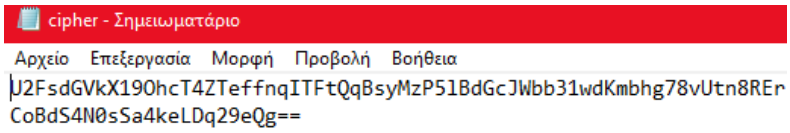


2.8

Αφότου δημιουργήθηκε το αρχείο μπορούμε να ξεκινήσουμε τη διαδικασία κρυπτογράφησης του. Χρησιμοποιούμε τη γραμμή εντολών και εισάγουμε εντολή
enc –aes-256-ecb –a –salt –in exampleaes.txt –out cipher.txt .

Το enc αντιπροσωπεύει το encoding και ως επιλογή χρησιμοποιεί τον αλγόριθμο AES με μέγεθος αλφαριθμητικού ίσο με 256 bit. Επίσης περιέχουμε και τις επιλογές
-a και -salt οι οποίες ενισχύουν την δύναμη της κρυπτογράφησης του αλγόριθμου. Ως είσοδο εισάγουμε το αρχείο exampleaes.txt και το αποτέλεσμα της κρυπτογράφησης του αποθηκεύεται σε ένα άλλο αρχείο τύπου .txt με όνομα cipher.

Τέλος, το σύστημα απαιτεί να εισαχθεί ένας κωδικός πρόσβασης που θα προστατεύει αυτό το κρυπτογραφημένο αρχείο. Όταν εισάγουμε τον κωδικό που θέλουμε να χρησιμοποιήσουμε η κρυπτογράφηση ολοκληρώνεται και το περιεχόμενο του αρχείου cipher.txt έχει πλέον κρυπτογραφηθεί.



2.9

Ακολούθως, η διαδικασία αποκρυπτογράφησης πραγματοποιείται με την εντολή
end -d -aes-257-ecb -a -salt -in cipher.txt -out plaintext.txt.

Η διαφορά της εντολής αποκρυπτογράφησης με αυτήν της κρυπτογράφησης είναι η χρήση του -d, που ενημερώνει το σύστημα για αυτό το συγκεκριμένο σκοπό.

Επίσης ως είσοδο δέχεται το κρυπτογραφημένο αρχείο cipher.txt και το αποτέλεσμα της αποκρυπτογράφησης του αποθηκεύεται σε ένα νέο αρχείο με όνομα plaintext.txt . Αξίζει να αναφερθούμε στο γεγονός ότι το αρχικό μας αρχείο exampleaes.txt δεν μας χρησιμεύει κάπου πλέον. Θα μπορούσε να είχε διαγραφεί , αλλά το περιεχόμενο του μπορεί να ανακτηθεί από το κρυπτογραφημένο αυτό αρχείο. Τέλος, επαληθεύουμε την εντολή αποκρυπτογράφησης με τη χρήση του κωδικού που δημιουργήσαμε για την κρυπτογράφηση του. Όταν ανοίξουμε το αρχείο plaintext.txt επιβεβαιώνουμε το μήνυμα που αποκρυπτογραφήθηκε.

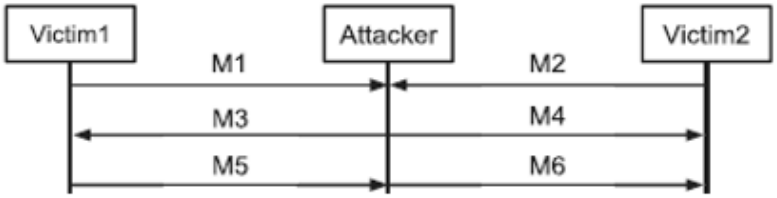
Κεφάλαιο 3ο

Επιθέσεις Man in the Middle (MitM)

3.1 Εισαγωγή στο MitM

Η επίθεση Man-in-the-Middle αποτελεί μια από τις πιο γνωστές επιθέσεις στον τομέα της ασφάλειας όσον αφορά τα υπολογιστικά και επικοινωνιακά συστήματα. Καθιστά το έργο να διατηρηθούν τα δεδομένα ασφαλή και ιδιωτικά ιδιαίτερα προκλητικό , καθώς οι επιθέσεις μπορούν να πραγματοποιηθούν από απομακρυσμένους υπολογιστές με ψευδείς διευθύνσεις. Η MitM στοχεύει στα δεδομένα που ρέουν μεταξύ των τελικών σημείων μιας επικοινωνίας, καθώς και στην εμπιστευτικότητα (confidentiality) και ακεραιότητα(integrity) τους. Πιο συγκεκριμένα εκμεταλλεύεται τις αδυναμίες στον έλεγχο ταυτοποίησης των πρωτόκολλων που χρησιμοποιούνται και από τα δύο μέρη της επικοινωνίας αυτής. Δεδομένου ότι ο έλεγχος ταυτότητας γενικά παρέχεται από τρίτους που εκδίδουν πιστοποιητικά , το σύστημα δημιουργίας πιστοποιητικών αποτελεί άλλη μια πηγή πιθανής αδυναμίας. Οι κατηγορίες της επίθεσης MitM διακρίνονται τουλάχιστον σε τρείς, οι οποίες αποτελούνται από τεχνικές πλαστοπροσωπίας(Impersonation) , κατηγορίες βασισμένες στο κανάλι επικοινωνίας και σε αυτές που χρησιμοποιούν τη θέση του επιτιθέμενου αλλά και των στόχων του στο δίκτυο. Η MitM είναι ένα είδος επίθεσης στο οποίο ένα κακόβουλο τρίτο μέρος αποκτά κρυφά τον έλεγχο καναλιού επικοινωνίας μεταξύ δύο ή περισσότερων τελικών σημείων[1]. Ο επιτιθέμενος μπορεί να παρακολουθήσει, να τροποποιήσει, να αλλάξει ή και να αντικαταστήσει τη ροή επικοινωνίας του θύματος.

Η παρακάτω εικόνα 1.4 απεικονίζει ένα συνηθισμένο σενάριο επίθεσης MitM όπου περιλαμβάνονται δύο τελικά σημεία (Victims) και ο επιτιθέμενος.



3.1

Πιο συγκεκριμένα ,ο επιτιθέμενος έχει καταφέρει να έχει πρόσβαση στο κανάλι επικοινωνίας των θυμάτων και μπορεί να χειριστεί τα μηνύματά τους. Τα M1 και M2 αποτελούν δημόσια κλειδιά που στέλνονται από τα θύματα στη προσπάθεια να αρχικοποιήσουν μια ασφαλή επικοινωνία μεταξύ τους. Ωστόσο, ο επιτιθέμενος αλλάζει τα κλειδιά M1,M2 από τη στιγμή που βρίσκεται στη μέση της επικοινωνίας τους και τα αντικαθιστά με τα μηνύματα M3,M4. Όταν ο Victim1 λάβει το M3, κρυπτογραφεί το μήνυμα του με βάσει το δημόσιο κλειδί που έλαβε και αποστέλλει το νέο κρυπτογραφημένο μήνυμα M5 στο Victim2. Ο επιτιθέμενος όμως παρακολουθεί το μήνυμα αυτό και το αποκρυπτογραφεί χρησιμοποιώντας ένα γνωστό ιδιωτικό κλειδί (Private key).Έπειτα κρυπτογραφεί το απλό δημόσιο κλειδί του Victim2 και του το αποστέλλει πίσω (M6 μήνυμα).Αυτή η διαδικασία που πραγματοποιεί ο επιτιθέμενος έχει ως αποτέλεσμα να πειστούν τα θύματα ότι το κανάλι που χρησιμοποιούν είναι ασφαλές, ενώ στην πραγματικότητα έχει αποκτήσει πρόσβαση σε όλα τα κρυπτογραφημένα μηνύματα. Καταλαβαίνουμε λοιπόν ότι ένας από τους λόγους που αυτό το είδος επίθεσης είναι τόσο δύσκολο να ανιχνευθεί είναι επειδή είναι ουσιαστικά ¨αόρατο¨ και κανένα από τα θύματα δεν μπορεί να καταλάβει ότι τα κρυπτογραφημένα μηνύματα τους δεν είναι ασφαλή και ότι έχουν παραβιαστεί[2]. Ωστόσο έχουν υπάρξει προσπάθειες αποφυγής τέτοιου είδους επίθεσης . Για παράδειγμα το Double Signature Lock Protocol το οποίο χρησιμοποιεί τεχνική κατακερματισμού όπου οι δυο υπογραφές που χρησιμοποιούνται στο κανάλι επικοινωνίας ανταλλάζονται με τη βοήθεια κατακερματισμένων ψηφίων για να μπορέσουν να επαληθευτούν με ασφάλεια. Αν οι υπογραφές δεν ταιριάζουν μεταξύ τους τότε η επικοινωνία δεν πραγματοποιείται και ο επιτιθέμενος που έχει εγκατασταθεί στη μέση της σύνδεσης μπορεί να αποφευχθεί διότι δεν γνωρίζει τα κατακερματισμένα αυτά στοιχεία που χρησιμοποιήθηκαν για την καθιέρωση της σύνδεσης. Αυτή η τεχνική ωστόσο δεν θεωρήθηκε επιτυχημένη επειδή ο επιτιθέμενος ήταν ακόμα ενεργός στο σχετικό δίκτυο. Αξίζει να αναφέρουμε ότι αυτού του είδους η επίθεση έχει χρησιμοποιηθεί στο παρελθόν ακόμα και από εταιρείες που προσπαθούσαν να ¨κυνηγήσουν¨ τους υπαλλήλους τους. Η ανακάλυψη που συνέβη το 2015 σχετικά με τους υπολογιστές Lenovo, όπου βγήκαν στην αγορά με προ-εγκατεστημένο adware ονομασμένο «Superfish» το οποίο εισάγει διαφημίσεις σε προγράμματα περιήγησης(Google Chrome , Internet Explorer). Πιο συγκεκριμένα εγκαθιστά ένα αυτό-δημιουργημένο root πιστοποιητικό στο αντίστοιχο πιστοποιητικό των Windows. Στη συνέχεια αποθηκεύει και αντικαθιστά όλα τα SSL πιστοποιητικά που παρουσιάζονται από HTTPS sites με το δικό του. Αυτό θα μπορούσε να επιτρέψει στους crackers/hackers να κλέψουν ευαίσθητα δεδομένα.

3.2 MitM-Spoofing επιθέσεις

Η MITM που βασίζεται σε Spoofing (spoofing-based MITM), είναι μια επίθεση κατά την οποία ο επιτιθέμενος παρακολουθεί μια νόμιμη επικοινωνία μεταξύ δύο μερών μέσω της Spoofing επίθεσης και ελέγχει τα δεδομένα που μεταφέρονται, ενώ οι δύο πλευρές δεν γνωρίζουν την ύπαρξη του ενδιάμεσου (middle man). Σε ορισμένες περιπτώσεις (π.χ. Spoofing στο DNS), ο επιτιθέμενος πραγματοποιεί Spoofing στις συσκευές μεταξύ των θυμάτων, ενώ σε άλλες περιπτώσεις, για παράδειγμα στο ARP Spoofing (Address Resolution Protocol), ο επιτιθέμενος κάνει απευθείας Spoofing στις συσκευές του θύματος.

Υπάρχουν διάφοροι τύποι επίθεσης Spoofing που μπορούν να χρησιμοποιήσουν τα κακόβουλα μέρη, όπως το ARP Spoofing, το DNS Spoofing (Domain Name System), το DHCP Spoofing (Dynamic Host Configuration Protocol) και το IP Spoofing (Internet Protocol). Σε όλους τους τύπους Spoofing, οι επιτιθέμενοι χρησιμοποιούν την ίδια αδυναμία των πρωτοκόλλων, την έλλειψη επικύρωσης των μηνυμάτων προέλευσης και προορισμού.

3.2.1 IP-Blind/No Blind-ICMP-TCP/SNP-ARP-DNS-DHCP

IP Protocol

Η IP (Internet Protocol) αποτελεί το πρωταρχικό πρωτόκολλο στο Internet και λειτουργεί στο επίπεδο δικτύου του μοντέλου OSI. Έχει ως στόχο την παράδοση πακέτων από τον host πηγής στον host προορισμού αποκλειστικά και μόνο με βάση τις διευθύνσεις IP στις κεφαλίδες πακέτων (packet headers). Η IP ορίζει τις δομές των πακέτων που ενσωματώνουν τα δεδομένα που πρόκειται να παραδοθούν. Ορίζει επίσης τις μεθόδους διευθυνσιοδότησης (addressing) που χρησιμοποιούνται για την επισήμανση του datagram με πληροφορίες πηγής (source) και προορισμού (destination). Η IP χρησιμοποιεί ένα μοντέλο χωρίς σύνδεση (connectionless model), που σημαίνει ότι δεν υπάρχουν πληροφορίες σχετικά με την κατάσταση της συναλλαγής, η οποία χρησιμοποιείται για την δρομολόγηση πακέτων σε ένα δίκτυο. Επιπλέον, η IP δεν καθορίζει καμία μέθοδο για την επικύρωση της αυθεντικότητας της πηγής ενός πακέτου. Αυτό σημαίνει ότι ο επιτιθέμενος θα μπορούσε να πλαστογραφήσει τη διεύθυνση πηγής (source address) ώστε να είναι οποιαδήποτε αυτός επιθυμεί. Μπορούμε να συμπεράνουμε λοιπόν ότι δεν είναι ένα αξιόπιστο πρωτόκολλο επειδή δεν μπορεί να εγγυηθεί την παράδοση πακέτων στον προορισμό και επειδή δεν παρέχει έλεγχο ροής και ανίχνευση/διόρθωση σφαλμάτων. Η επίθεση MITM που βασίζεται σε IP Spoofing είναι μια επίθεση στην οποία ένα κακόβουλο μέρος παρακολουθεί μια νόμιμη επικοινωνία μεταξύ δύο μη-κακόβουλων μερών. Η κακόβουλη οντότητα ελέγχει τη ροή της επικοινωνίας και μπορεί να εξαλείψει ή να τροποποιήσει τις πληροφορίες που έστειλε ο ένας από τους αρχικούς συμμετέχοντες, χωρίς να γνωρίζει κανένα από τα αρχικά-τελικά σημεία.

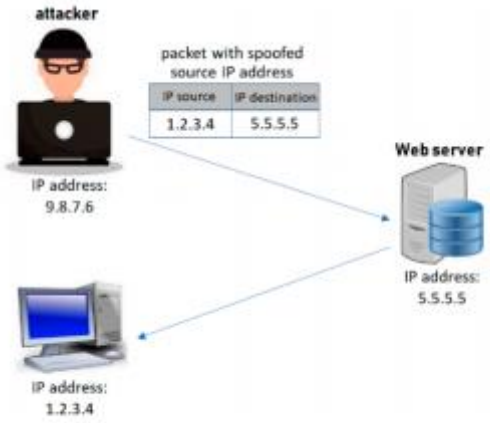


Figure 1. Transmission of spoofed IP packets.

3.2

Blind-Non-Blind Spoofing/ICMP Spoofing

Το Blind spoofing απαιτεί από έναν επιτιθέμενο να στείλει αρχικά αιτήματα σε ένα δίκτυο και στη συνέχεια να είναι σε θέση να αναλύσει τη σειρά μετάδοσης.
Αντιθέτως, στο Non-Blind spoofing ο επιτιθέμενος βρίσκεται στο ίδιο υποδίκτυο με το θύμα, γεγονός που του δίνει την ευκαιρία να κάνει Sniffing σε αριθμούς ακολουθίας και επιβεβαίωσης. Και οι δύο μέθοδοι που αναφέραμε χρησιμοποιούνται συνήθως για επιθέσεις DdoS , αλλά εκτελούνται ως φάση εξόρυξης δεδομένων (Data mining phase) για την επίθεση MITM που βασίζεται σε IP-Spoofing.

Το ICMP (Internet Control Message Protocol) αποτελεί ένα πρωτόκολλο που χρησιμοποιείται στο επίπεδο διαδικτύου του TCP/IP .Η IP χρησιμοποιεί το ICMP (για την αποστολή μηνυμάτων μονής κατεύθυνσης (one-way) για την εφαρμογή διάφορων δυνατοτήτων αναφοράς σφαλμάτων , feedback και test. Επίσης το ICMP κατέχει μηνύματα ανακατεύθυνσης (redirecting) , τα οποία χρησιμοποιούνται συνήθως για να ειδοποιούν τους δρομολογητές για μια καλύτερη διαδρομή. Τα μηνύματα αυτά ενδέχεται να καταχραστούν για την εκτέλεση επίθεσης MiTM , εφόσον το ICMP δεν παρέχει μηχανισμούς ελέγχου ταυτότητας. Επομένως ο επιτιθέμενος πραγματοποιεί spoofing στα μηνύματα ICMP redirect για να ανακατευθύνει την κυκλοφορία του θύματος μέσω του δρομολογητή του , όπου μπορεί να τα παρακολουθεί και να τα τροποποιεί

TCP Sequence-Number Prediction

Το πρωτόκολλο TCP (Transmission Control Protocol) λειτουργεί πάνω από το IP και είναι υπεύθυνο για τη διάσπαση της ροής δεδομένων σε τμήματα πριν τα μεταβιβάσει στην IP και τα επανασυναρμολογεί στον προορισμό. Το TCP πρωτόκολλο θεωρείται αξιόπιστο επειδή εγγυάται την παράδοση πακέτων και ενσωματώνει μηχανισμούς ανίχνευσης και διόρθωσης σφαλμάτων. Υπάρχουν διαφορετικοί μηχανισμοί που χρησιμοποιούνται από το TCP για τη διασφάλιση της παράδοσης πακέτων, όπως αριθμοί ακολουθίας, αναγνώριση, χειραψία τρίτων (Three-way handshake) και χρονόμετρα (Timers).

Το Three-way handshake αποτελεί τη διαδικασία που χρησιμοποιείται για την δημιουργία σύνδεσης μεταξύ των δύο πλευρών που συμμετέχουν σε μια σύνδεση TCP. Το three-way αντιπροσωπεύει τα : SYN,SYN-ACK,ACK. Δηλαδή, ο πελάτης στέλνει ένα SYNchronized packet στον Server. Ακολούθως , ο Server λαμβάνει αυτό το πακέτο και αποστέλλει με τη σειρά του ένα SYNchronized ACKnowledged packet πίσω στον πελάτη ο οποίος με τη σειρά του το λαμβάνει[18].

Το TCP πρωτόκολλο όπως αναφέραμε παραπάνω χρησιμοποιεί αριθμούς ακολουθιών για την επιβεβαίωση δεδομένων. Οι αριθμοί αυτοί βοηθούν το πρωτόκολλο να μειώσει την απώλεια δεδομένων για τον προσδιορισμό πακέτων εκτός σειράς με τον ίδιο τρόπο που διασφαλίζουν την αξιοπιστία. Οι αριθμοί παράγονται με ψευδοτυχαίο τρόπο που είναι γνωστός και στα δύο μέρη. Η ιδέα της πρόβλεψης της ακολουθίας των αριθμών (Sequence-Number Prediction) είναι να βρει τον αλγόριθμο της δημιουργίας των αριθμών ακολουθιών και στη συνέχεια να χρησιμοποιήσει αυτή τη γνώση για να αναχαιτίσει μια υπάρχουσα συνεδρία (Session).

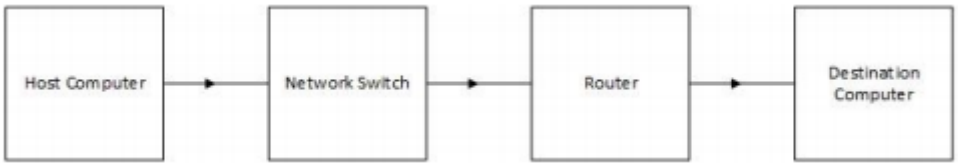
ARP Spoofing

Το πρωτόκολλο ARP (Address Resolution Protocol) χρησιμοποιείται από τις συσκευές δικτύου για να αντιστοιχήσουν τις διευθύνσεις του δικτύου τους σε διευθύνσεις ελέγχου της πρόσβασης των πολυμέσων (MAC-Media access Control). Το ARP είναι πάρα πολύ σημαντικό στις επικοινωνίες LAN (Local Area Network), επειδή κάθε πλαίσιο (Frame) που αφήνει έναν κεντρικό υπολογιστή πρέπει να περιέχει μια διεύθυνση προορισμού MAC[4].

Σε μια κανονική επικοινωνία ARP, ο κεντρικός υπολογιστής θα στείλει ένα πακέτο που περιέχει την IP διεύθυνση της πηγής και του προορισμού. Στη συνέχεια θα το μεταδώσει σε όλες τις συσκευές που είναι συνδεδεμένες στο δίκτυο αυτό. Η συσκευή που κατέχει την διεύθυνση προορισμού θα αποστείλει την ARP απάντηση με περιεχόμενο μόνο την MAC διεύθυνση του και εν συνεχεία πραγματοποιείται η επικοινωνία[8].

Το πρωτόκολλο αυτό δεν είναι ασφαλές επειδή στην ARP μνήμη δεν περιέχεται κάποιος ανθεκτικός μηχανισμός αποτροπής εισβολών , κάτι το οποίο αποτελεί μεγάλο πρόβλημα. Πιο συγκεκριμένα , τα ARP requests και replies δεν απαιτούν κάποιον έλεγχο ταυτότητας, αφού όλοι οι κεντρικοί υπολογιστές στο δίκτυο εμπιστεύονται όλες τις απαντήσεις που πραγματοποιεί το ARP[12].

Στη παρακάτω φωτογραφία 1.2 απεικονίζεται μια φυσιολογική επικοινωνία μεταξύ δύο υπολογιστών



3.3

Το πακέτο απάντησης του ARP μπορεί εύκολα να κλαπεί και μπορεί να σταλεί στη συσκευή που πραγματοποίησε το ARP αίτημα χωρίς να γνωρίζει ότι αυτή η συσκευή χρησιμοποιείται από κάποιον τρίτο με σκοπό την υποκλοπή δεδομένων. Αυτό συμβαίνει επειδή ο πίνακας προσωρινής μνήμης του πρωτόκολλου ARP ενημερώνεται όπως θέλει ο εισβολέας, με αποτέλεσμα όλο το network traffic να είναι ορατό από αυτόν[13]. Αυτού του είδους η επίθεση αποτελεί τον καλύτερο και πιο αποτελεσματικό τρόπο εισβολής στα LAN.

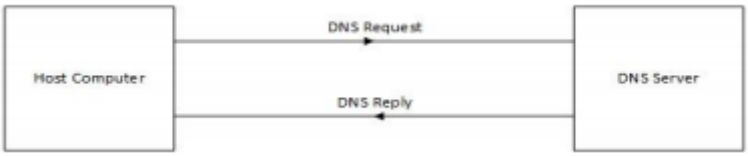
DNS Spoofing

Το DNS (Domain Name System) αποτελεί ένα ιεραρχικό σύστημα ονομασίας για το διαδίκτυο που βασίζεται σε μια αρχιτεκτονική πελάτη-διακομιστή. Η κύρια λειτουργία ενός διακομιστή DNS είναι η εκτέλεση ανάλυσης του URL (διαδικασία μετάφρασης διεύθυνσης URL ή ονόματος τομέα). Οι διακομιστές DNS και τα ονόματα τομέων είναι οργανωμένα ιεραρχικά σε τομείς ανώτατου επιπέδου, δευτερεύοντος επιπέδου και σε αυτούς του κατώτερου επιπέδου[15]. Για παράδειγμα στο icsd.aegean.gr το gr αποτελεί το ανώτατο επίπεδο, το Aegean το δευτερεύον και το icsd το κατώτερο.

Μια από τις πιο επικίνδυνες επιθέσεις εναντίον του DNS αποτελεί το DNS Spoofing, το οποίο εκτελείται μέσω του poisoning της κρυφής μνήμης (το DNS spoofing πολλές φορές αναφέρεται και ως DNS poisoning). Η υπηρεσία DNS χρησιμοποιεί το σύστημα κρυφής μνήμης για τη βελτίωση της απόδοσης, αλλά έχει διάφορα αδύναμα σημεία. Το DNS Spoofing έχει ως αποτέλεσμα την αποθήκευση από το DNS των μη έγκυρων ή κακόβουλων χαρτογραφήσεων μεταξύ συμβολικών ονομάτων και διευθύνσεων IP[14].

Οι επιτιθέμενοι, για να μπορέσουν να εκτελέσουν την DNS Spoofing, πρέπει να παρακάμψουν τις καταχωρήσεις δρομολόγησης του τοπικού DNS με ψεύτικα δεδομένα. Το DNS ανανεώνει την κρυφή μνήμη του, ανάλογα με το TTL(Time To Live) των καταχωρήσεων, οπότε ανά τακτά χρονικά διαστήματα ζητάει άλλες DNS για ενημερώσεις(updates)[19].Ο επιτιθέμενος μπορεί να το χρησιμοποιήσει αυτό για την εκτέλεση της επίθεσης του.

Για την καλύτερη κατανόηση της MiTM επίθεσης στο DNS θα χρησιμοποιήσουμε ένα παράδειγμα όπου ο επιτιθέμενος έχει δημιουργήσει μια ψευδή ιστοσελίδα της τράπεζας του θύματος. Όταν ο χρήστης επισκεφτεί την ιστοσελίδα της τράπεζάς του, ανακατευθύνεται στη σελίδα που έχει δημιουργήσει ο επιτιθέμενος , όπου στόχος του είναι να κλέψει τα στοιχεία του χρήστη. Γενικότερα, κατά τη έναρξη της διαδικασίας περιήγησης σε μια ιστοσελίδα από τον υπολογιστή, ένα DNS αίτημα αποστέλλεται στον διακομιστή(DNS Server) όπου με τη σειρά του ανταποκρίνεται στο αίτημα αυτό με μια απάντηση (DNS Reply).Αυτή η επικοινωνία απεικονίζεται στην εικόνα 3.4.



3.4

Το αίτημα και η απάντηση DNS αντιστοιχίζεται με έναν μοναδικό αριθμό αναγνώρισης. Εάν ο εισβολέας καταφέρει και υποκλέψει αυτόν τον αριθμό, μπορεί να αποστείλει ένα κακόβουλο πακέτο που περιέχει το αναγνωριστικό αυτό στον χρήστη. Αυτό έχει ως συνέπεια την επιτυχημένη ανακατεύθυνση του χρήστη από την ιστοσελίδα της τράπεζας σε αυτήν που έχει δημιουργήσει ο επιτιθέμενος. Πιο συγκεκριμένα, ως απάντηση στο αίτημα DNS, αποστέλλεται το ψευδές πακέτο που έχει δημιουργηθεί.

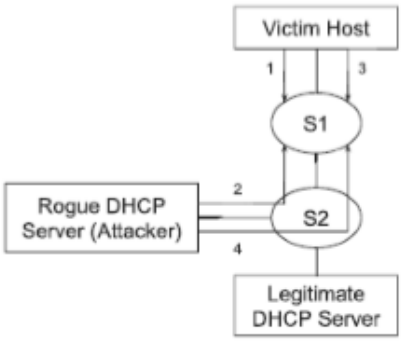
3.5

Στη παραπάνω εικόνα 3.5 ο κεντρικός υπολογιστής(Host Computer) θέλει να συνδεθεί στην ιστοσελίδα της τράπεζας, οπότε αποστέλλει ένα DNS αίτημα σύνδεσης στον εξυπηρετητή. Όμως λόγω της επίθεσης MiTM , ο επιτιθέμενος αναχαιτεί το μήνυμα αυτό και στέλνει μια ψευδή απάντηση (DNS response) στον υπολογιστή[20]. Ο δεν μπορεί να γνωρίζει αν η απάντηση αυτή είναι νόμιμη ή όχι ,και έτσι εγκαθιστά την επικοινωνία με την κακόβουλη ιστοσελίδα.

DHCP Spoofing

Το DCHP (Dynamic Host Configuration Protocol), είναι ένα πρωτόκολλο που παρέχει παραμέτρους διαμόρφωσης (configuration parameters) του δικτύου με πρόσφατα συνδεδεμένους εξυπηρετητές. Οι παράμετροι αυτοί περιλαμβάνουν τη διεύθυνση IP, τη μάσκα του υποδικτύου (subnet mask), την προεπιλεγμένη πύλη εισόδου (default gateway), τον διακομιστή DNS και τον μισθωμένο χρόνο(leased time). Το DHCP παρέχει μια δομή πελάτη–διακομιστή, στην οποία έχουν ανταλλαγεί τέσσερα πακέτα DHCP μεταξύ του εξυπηρετητή DHCP και των hosts για την αυτόματη ανάθεση των παραπάνω παραμέτρων. Το DHCP αποτελεί σημαντικό ρόλο στη διαχείριση δικτύου. Ωστόσο, υπάρχουν μερικά γνωστά μειονεκτήματα όσον αφορά την ασφάλεια που παρέχει. Το DHCP από μόνο του δεν διαθέτει πιστοποίηση προέλευσης μηνυμάτων DHCP. Από τη μία πλευρά, οι πελάτες DHCP δεν μπορούν να εγγυηθούν ότι είναι συνδεδεμένοι με έναν αξιόπιστο διακομιστή DHCP, ενώ από την άλλη πλευρά ο διακομιστής DHCP δεν είναι σε θέση να διασφαλίσει ότι επικοινωνεί με έναν νόμιμο πελάτη. Επιπρόσθετα, κάθε μήνυμα DHCP μεταδίδεται σε ένα σαφές κείμενο (cleartext)[15]. Η MiTM επίθεση με βάση το DHCP spoofing είναι δυνατή μέσω του διακομιστή DHCP. Ο επιτιθέμενος προσπαθεί να απαντήσει σε μια αίτηση DHCP γρηγορότερα από τον νόμιμο διακομιστή DHCP του τοπικού δικτύου.

Σαν παράδειγμα έχουμε ένα δίκτυο το οποίο αποτελείται από τον host Victim, τον νόμιμο DHCP Server, τον Rogue DHCP Server και δύο διακόπτες S1,S2. Όταν το θύμα συνδεθεί στο δίκτυο, εμφανίζεται η επικοινωνία που βλέπουμε στην παρακάτω εικόνα.



3.6

Για αρχή, ο υπολογιστής-πελάτης μεταδίδει το DHCP Discovery. Στη συνέχεια ο Server του επιτιθέμενου αποστέλλει το DHCP Offer(Unicast).Ο πελάτης με τη σειρά του μεταδίδει το αίτημα DHCP, με τον Server του επιτιθέμενου να λαμβάνει και να αποστέλλει γρήγορα ένα DHCP ACK(Unicast) πακέτου[16].

Επιπρόσθετα, ο επιτιθέμενος μπορεί να εκτελέσει μια επίθεση DoS στο νόμιμο διακομιστή DHCP για να εξασφαλίσει ότι ο host του θύματος δεν θα λάβει απάντηση από αυτό.

3.3 MitM Επιθέσεις στο HTTP

Το HTTP (HyperText Transfer Protocol) αποτελεί ένα πρωτόκολλο το οποίο είναι σχεδιασμένο για κατανεμημένα και συνεργατικά συστήματα πληροφοριών. Πιο απλοϊκά , το πρωτόκολλο αυτό είναι υπεύθυνο για την συλλογή και προβολή ιστοσελίδων από τους εξυπηρετητές με τη βοήθεια των περιηγητών ιστού (Firefox , Internet Explorer). Το HTTP είναι αρκετά παλιό. Η πρώτη έκδοση του (HTTP/1.0) δημιουργήθηκε το 1996 και έπειτα ακολούθησε το HTTP/1.1 το 1999. Αξίζει να αναφέρουμε ότι η Google πρότεινε πρόσφατα μέσω ενός project (SPDY) μια νέα έκδοση με την ονομασία HTTP/2. Ωστόσο θα επικεντρωθούμε στην 1.1 έκδοση.

Το γενικό πρόβλημα με τα παλιά πρωτόκολλα είναι ότι δεν είναι σχεδιασμένα με βάση την ασφάλεια. Αυτό σημαίνει ότι το HTTP δεν χρησιμοποιεί καμία μέθοδο κρυπτογράφησης ή πιστοποίησης. Συνεπώς, αν κάποιος παρακολουθεί το traffic του δικτύου , μπορεί να δει ποιες σελίδες είναι προσβάσιμες και έτσι μπορεί να παραβιάσει τη σύνδεση σχετικά εύκολα αν το θέλει. Για να αποφευχθεί αυτό το σενάριο , δημιουργήθηκε το HTTPS , το οποίο χρησιμοποιεί το πρωτόκολλο SSL ή TLS για την προστασία των δεδομένων του[5].
Παρακάτω θα αναλύσουμε κάποιες επιθέσεις και αποτελέσματα επιθέσεων MiTM στο HTTP πρωτόκολλο.

3.3.1 Sniffing-Cookie Insertion-Cache poisoning

Ο πιο απλός τρόπος να πραγματοποιηθεί επίθεση στο HTTP είναι να παρακολουθείται η σύνδεση με σκοπό την υποκλοπή των POST αιτημάτων. Αν ο χρήστης προσπαθήσει να συνδεθεί σε μια ιστοσελίδα με το username και το password του, τότε από τη στιγμή που τα στοιχεία αυτά δεν κρυπτογραφούνται, ο επιτιθέμενος μπορεί να κλέψει εύκολα. Επίσης ο επίθεση αυτή μπορεί να συνδυαστεί με τον διαχωρισμό του SSL από το HTTP με σκοπό την υποκλοπή περισσότερων δεδομένων. Ένα εργαλείο που μπορεί να το καταφέρει αυτό είναι το mitmproxy, το οποίο μπορεί να εμφανίζει το HTTP traffic.Αξίζει να αναφέρουμε ότι αν ο server αναβαθμιστεί έτσι ώστε να χρησιμοποιεί HTTPS, τότε καθιστά την επίθεση αυτή εύκολα στο να αποφευχθεί.

Σε κάποιες εφαρμογές των προγραμμάτων περιήγησης, ο επιτιθέμενος είναι δυνατό να παρακάμψει ή να ορίσει cookies με το ίδιο όνομα. Το HTTPS καθορίζει τα cookies που έχουν το ίδιο όνομα με αυτά που έχει ορίσει το HTTP. Σε μια τυπική σύνδεση HTTP , ο επιτιθέμενος μπορεί επίσης να εισάγει cookies. Για να επιτευχθεί αυτό , ο εισβολέας πρέπει να προσθέσει cookie κεφαλίδες σε αιτήματα απόκρισης HTTP.Αυτή η επίθεση μπορεί να πραγματοποιηθεί χάρη σε ένα σφάλμα (bug) που υπάρχει σε κάποιους web browsers.

Γενικότερα , οι χρήστες των browser πρέπει να ενημερώνουν το πρόγραμμα περιήγησης τους και οι προμηθευτές πρέπει να κάνουν περιστασιακούς ελέγχους. Επίσης οι χρήστες έχουν τη δυνατότητα να διαγράψουν cookies από τους browsers , ωστόσο αυτό μπορεί να οδηγήσει σε crash των ιστοσελίδων. Επομένως αποτελεί έναν τρόπο αποφυγής αυτής της επίθεσης , αλλά δεν είναι εντελώς αποτελεσματικός. Κατά τη διάρκεια περιήγησης στο διαδίκτυο πρέπει να γίνεται συχνά λήψη δεδομένων, το οποίο είναι χάσιμο χρόνου και bandwidth . Για την αποτροπή αυτού , τα προγράμματα περιήγησης εφαρμόζουν μια προσωρινή μνήμη περιεχομένου. Το HTTP παρέχει κεφαλίδες με τις οποίες ένας διακομιστής μπορεί να ελέγχει αυτή τη προσωρινή αποθήκευση. Τα μεγαλύτερα δεδομένα (εικόνες κ.λπ.) αποθηκεύονται προσωρινά στη κρυφή μνήμη για να βελτιωθεί ο χρόνος φόρτωσης.
Ωστόσο, όπως αναφέραμε, το HTTP αποτελεί ένα πρωτόκολλο απλού κειμένου , συμπεριλαμβανομένων των κεφαλίδων ελέγχου κρυφής μνήμης. Αυτό καθιστά δυνατή την εισχώρηση κακόβουλου κώδικα Javascript σε αρχεία , τα οποία στη συνέχεια θα αποθηκευτούν στη κρυφή μνήμη από το πρόγραμμα περιήγησης . Η επιτυχία αυτής της επίθεσης θεωρείται καταστροφική όσο αφορά την ασφάλεια. Ο λόγος που θεωρείται αυτό είναι γιατί ο επιτιθέμενος είναι σε θέση να διαβάσει όλα τα input στη σελίδα , να προβάλλει ότι θέλει στον χρήστη, αλλά και να διαρρεύσει σημαντικές πληροφορίες. Άλλο ένα σοβαρό πρόβλημα είναι η δυνατότητα του επιτιθέμενου να φορτώσει ένα BeFF hook. Αυτό μπορεί να χρησιμοποιηθεί μαζί με το Metasploit για να εκμεταλλευτεί τα μειονεκτήματα του browser με απώτερο σκοπό την εισβολή στο σύστημα. Ωστόσο αυτού του είδους η επίθεση δεν αφορά αποκλειστικά την χρήση Javascript σε αρχεία.Η αποφυγή της επίθεσης αυτής μπορεί να πραγματοποιηθεί με την αναβάθμιση του HTTP σε HTTPS, καθώς και με την εκκαθάριση αρχείων της κρυφής μνήμης του browser.

3.3.2 Binary Patching -HTTPS Stripping/Splitting

Γενικώς είναι γνωστό ότι το να 'κατεβάζουμε' εκτελέσιμα αρχεία είναι ριψοκίνδυνο επειδή πολλοί ιστότοποι παρέχουν λογισμικά προς εγκατάσταση που είναι 'πειραγμένα'(crapware). Ωστόσο, ακόμα και αν καταφέρει ο χρήστης να βρει την επίσημη ιστοσελίδα που παρέχει το πρόγραμμα προς εγκατάσταση που αναζητά, δεν σημαίνει ότι είναι εντελώς ασφαλές. Αυτό συμβαίνει επειδή ο επιτιθέμενος μπορεί να παραμετροποιήσει τα δυαδικά αρχεία μέσω proxies. Ουσιαστικά αλλάζει τον κώδικα στο αρχικό εκτελέσιμο αρχείο με έναν της αρέσκειας του. Συνεπώς, μπορεί να κάνει κατάχρηση τω λήψεων που συνέβησαν μέσω του HTTP. Υπάρχουν ενδείξεις ότι κακόβουλοι κόμβοι TOR (nodes) χρησιμοποιούν αυτή τη μέθοδο. Επίσης, αυτού του είδους επίθεσης εφαρμόζεται πλέον σε BDFProxies, ένα βοηθητικό πρόγραμμα που περιτυλίγει αυτόματα ένα εκτελέσιμο αρχείο. Εάν αυτή η επίθεση εφαρμοστεί με επιτυχία, ο επιτιθέμενος μπορεί να δημιουργήσει μια backdoor είσοδο.
Παρόμοια με τις προηγούμενες επιθέσεις, μπορούμε να το αποφύγουμε με το να αναβαθμίσουμε το HTTP σε HTTPS . Επίσης ο χρήστης μπορεί να χρησιμοποιήσει προγράμματα ανίχνευσης κακόβουλων λογισμικών(antivirus programs).

Κατά τη δημιουργία ιστοσελίδων με HTTPS, οι προγραμματιστές πρέπει να σιγουρευτούν ότι κάθε στοιχείο είναι προστατευμένο με το πρωτόκολλο αυτό. Όταν ένα δεδομένο είναι δημιουργημένο με βάση το HTTP, ο επιτιθέμενος είναι ικανός να το παραμετροποιήσει. Αλλάζοντας αυτά τα στοιχεία, η ακεραιότητα όλης της ιστοσελίδας βρίσκεται σε ρίσκο, με χειρότερο σενάριο την συνεχή εμφάνιση σφαλμάτων σε όλη τη σελίδα(Κάθε browser προβάλλει διαφορετικά σφάλματα).

Για την αποφυγή αυτής της επίθεσης έχουν δημιουργηθεί οι επεκτάσεις CSP(Content Security Policy) και HSTS(HTTP Strict Transport Security). Η CSP δίνει τη δυνατότητα στον προγραμματιστή να καθορίσει από ποια domain μπορεί να φορτωθεί το περιεχόμενο έτσι ώστε να αποφευχθεί η πιθανότητα cross-siting, ενώ με την HSTS ο προγραμματιστής είναι σε θέση να προσδιορίσει ποια δεδομένα της ιστοσελίδας, ακόμα και αυτά που υπο φυσιολογικές συνθήκες θα χρησιμοποιούσαν HTTP για να φορτωθούν , πρέπει να χρησιμοποιήσουν HTTPS.Οι παραπάνω επεκτάσεις πρέπει να ενεργοποιηθούν και να χρησιμοποιηθούν από τους admin του server και από τους προμηθευτές των προγραμμάτων περιήγησης. Είναι σημαντικό να χρησιμοποιηθούν και από τις δύο μεριές , αλλιώς τα πλεονεκτήματα τους δεν μπορούν να αξιοποιηθούν.

Αξίζει να αναφέρουμε ότι τα πιο δημοφιλή προγράμματα περιήγησης (Google Chrome, Firefox,Internet Explorer , Opera) ενημερώνουν τους χρήστες τους όταν φορτώνεται μεικτό περιεχόμενο. Ωστόσο , το ποιο περιεχόμενο θεωρείται επιβλαβές για τον χρήστη διαφέρει μεταξύ των browser.

Το HTTPS Stripping αποτελεί μια επίθεση που υποβαθμίζει μια HTTPS σύνδεση σε μια HTTP με τη βοήθεια proxy server. Μέσω μιας έξυπνης επανεγγραφής που πραγματοποιείται από τον proxy server, παρουσιάζονται στο θύμα εκδόσεις HTTP των σελίδων που επισκέπτεται (και όχι HTTPS).Αν και αυτή η επίθεση από μόνη της δεν μπορεί να επηρεάσει το θύμα , επιτρέπει στον εισβολέα να αποφύγει την ασφάλεια του HTTPS και έτσι μπορεί να υλοποιήσει το σχέδιο του. Αυτή η επίθεση διαδόθηκε από το εργαλείο SSLstrip (υλοποιημένο από τον Moxie Marlinspike).

Για την αποφυγή αυτής της επίθεσης , οι χρήστες μπορούν να χρησιμοποιήσουν ένα plugin (HSTS) το οποίο εξασφαλίζει ότι ο ιστότοπος θα παραμείνει σε HTTPS σύνδεση. Εάν μια ιστοσελίδα βρίσκεται σε HTTP σύνδεση, τότε αυτόματα αλλάζει σε HTTPS.

Ακριβώς όπως το HTTPS Stripping , έτσι και το HTTPS Splitting αποτελεί μια μέθοδο ανάκτησης δυνατοτήτων παραβίασης. Το HTTPS Splitting λειτουργεί με Proxy servers. Το θύμα συνδέεται στον proxy server και με τη σειρά του ο server παρουσιάζει στον χρήστη ένα ψεύτικο certificate για οποιοδήποτε ιστότοπο θέλει να συνδεθεί. Ο server, από τη στιγμή που είναι σε θέση να αποκρυπτογραφήσει οποιοδήποτε αίτημα πραγματοποιεί ο χρήστης , εγκαθιστά μια σύνδεση HTTPS με όποιο server ζητά να συνδεθεί το θύμα(SSLsplit αποτελεί και πάλι το εργαλείο που το καθιστά δυνατό). Η επίθεση αυτή διαφέρει με αυτή που αναφέραμε παραπάνω. Η διαφορά είναι ότι με αυτή την επίθεση ο επιτιθέμενος διατηρεί την SSL/TLS σύνδεση με το θύμα αλλά και με τον server. Επίσης ο θύτης δεν χρειάζεται να αλλάζει το certificate του για κάθε ασφαλή σύνδεση που πραγματοποιεί ο χρήστης.

3.4 Επιθέσεις σε FTP-SSH

Στην υποενότητα αυτή αναφέρονται οι επιθέσεις Man In The Middle στο πρωτόκολλο μεταφοράς αρχείων FTP, καθώς και στο πρωτόκολλο SSH το οποίο έχει δημιουργηθεί για την αποτροπή επιθέσεων sniffing. Επίσης , αξίζει να σημειωθεί ότι το περιεχόμενο της υποενότητας αυτής είναι πολύ σημαντικό διότι το FTP χρησιμοποιείται πολύ συχνά και επειδή το SSH έχει δημιουργηθεί αποκλειστικά για την αποτροπή επιθέσεων, αλλά ακόμα υπάρχουν περιπτώσεις που η επίθεση MitM πραγματοποιείται με επιτυχία.

3.4.1 File Transfer Protocol (FTP)

Το FTP (File Transfer Protocol) αποτελεί ένα πρωτόκολλο που χρησιμοποιείται για την μεταφορά αρχείων στο διαδίκτυο. Επίσης , είναι το πιο διαδεδομένο πρωτόκολλο που αφορά το upload αρχείων σε ιστοσελίδες αλλά και για τα download.Γενικότερα υπάρχουν αρκετοί τρόποι σύνδεσης σε έναν FTP server. Αυτοί κυμαίνονται από απευθείας συνδέσεις σε FTP servers, με σύνδεση FTPS όπου συνδυάζεται με το SSL πρωτόκολλο για την καλύτερη ασφάλεια και ακεραιότητα της σύνδεσης. Ένας άλλος τρόπος είναι να διοχετευθεί το traffic της σύνδεσης μέσω του SSH πρωτόκολλου (SFTP) . Χρησιμοποιώντας το SFTP , καθιστά μη αναγκαία την απευθείας σύνδεση του FTP server στο διαδίκτυο.

Το FTP από μόνο του δεν περιέχει μεθόδους κρυπτογράφησης ή πιστοποίησης , και για αυτό μια επίθεση MiTM είναι πιθανό να πραγματοποιηθεί. Για αυτό το λόγο δημιουργήθηκαν τα πρωτόκολλα FTPS και SFTP όπου παρέχουν την αναγκαία ασφάλεια. Παρόλα αυτά ακόμα και σήμερα πολλοί server δεν έχουν αναβαθμιστεί και χρησιμοποιούν μόνο το FTP[11].

Παρακάτω θα σχολιάσουμε επιθέσεις στο FTP πρωτόκολλο που αφορούν το Sniffing το Binary Patching και επιθέσεις τύπου injection σε NAT,FTPS εξυπηρετητές καθώς και injection με τερματισμό TCP.Από τη στιγμή που το FTP δεν περιέχει τρόπους κρυπτογράφησης , είναι ευάλωτο σε επιθέσεις sniffing. Ο επιτιθέμενος χρησιμοποιώντας κατάλληλα εργαλεία (π.χ. Metasploit) μπορεί να υποκλέψει κωδικούς και άλλα σημαντικά δεδομένα. Για την αποφυγή αυτής της επίθεσης πρέπει να αναβαθμιστεί το πρωτόκολλο αυτό σε FTPS ή SFTP. Ωστόσο και τα δύο αυτά πρωτόκολλα είναι ευάλωτα σε splitting επιθέσεις.

Από τη στιγμή που το πρωτόκολλο αυτό δεν περιέχει μεθόδους ασφάλειας ο επιτιθέμενος μπορεί να υποκλέψει και να παραμετροποιήσει αρχεία που αποστέλλονται από τον server στο χρήστη , αλλά και το αντίστροφο. Το θύμα μπορεί να δεχτεί επίθεση μέσω μολυσμένων εκτελέσιμων αρχείων που έχει ‘κατεβάσει’ από το διαδίκτυο. Όσον αφορά τον server , μπορεί να δεχτεί επίθεση με την εισαγωγή κώδικα (php scripts) , ή όπως αναφέραμε παραπάνω με το upload παραμετροποιημένων εκτελέσιμων αρχείων σε αυτόν.

Για την αποφυγή της επίθεσης αυτής πρέπει να αναβαθμιστεί το πρωτόκολλο αυτό όπως αναφέραμε στην περίπτωση του Sniffing. Εάν ένας server έχει ως μοναδικό σκοπό την παροχή αρχείων στους χρήστες , τότε με την δημοσίευση του κώδικα κατακερματισμού (hash codes) σε αυτούς , τους δίνει τη δυνατότητα να ελέγξουν εάν ο κώδικας των αρχείων που ‘κατέβασαν’ είναι διαφορετικός ή όχι. Ωστόσο , πρέπει να σιγουρευτούμε ότι ο κώδικας που έχει σταλθεί από τον server είναι ασφαλής και δεν έχει κλαπεί από τον θύτη.

Η επίθεση injection είναι πιθανή μόνο σε FTPS servers οι οποίοι χρησιμοποιούν ενεργό NAT (Network Address Translation). Αν και δύσκολο, είναι πιθανό ένας επιτιθέμενος να εισάγει δεδομένα της αρέσκειας του σε μια FTPS session. Εξαιτίας της χρήσης NAT , ο server αναγκάζεται να υποβαθμίσει την FTPS σύνδεση που έχει δημιουργήσει σε μία πιο ευάλωτη και λιγότερο ασφαλή έτσι ώστε να μπορέσει το NAT να αναδιοργανώσει τις PASV και PORT εντολές. Κατά τη διάρκεια αυτής της αλλαγής ο server είναι ευάλωτος και μπορεί να πέσει θύμα επίθεσης. Η αποφυγή της επίθεσης αυτής μπορεί να αποφευχθεί αναθέτοντας στατικά θύρες στον server έτσι ώστε το NAT να μην χρειάζεται να τροποποιήσει αυτά τα πακέτα. Αυτό επιτρέπει στη σύνδεση να είναι ασφαλής καθ' όλη τη διάρκεια της.

Στη περίπτωση επίθεσης injection με χρήση TCP τερματισμού, ο επιτιθέμενος μπορεί να διακόψει προσωρινά τη σύνδεση στον FTPS server χρησιμοποιώντας το TCP πρωτόκολλο. Κατά τη διάρκεια προσπάθειας επανασύνδεσης το επίπεδο ασφάλειας υποβαθμίζεται δραματικά. Έτσι ο επιτιθέμενος μπορεί να 'ανεβάσει' ότι δεδομένα θέλει στον server.Ειδικότερα, αυτή η επίθεση είναι δυνατή εξαιτίας του χειρισμού των τερματικών του TCP. Ένας πελάτης πρέπει να στείλει μήνυμα TLSShutdown για να τερματιστεί η σύνδεση , ωστόσο οι περισσότεροι δεν το κάνουν αυτό. Για αυτό το λόγο οι FTP server θεωρούν φυσιολογικό να τερματίζεται η σύνδεση, ακόμα και αν αυτό συμβαίνει απροειδοποίητα.
Η μόνη λύση για την αποφυγή αυτού του προβλήματος είναι να μην επιτρέπεται η επανασύνδεση μετά από απροειδοποίητους τερματισμούς της σύνδεσης, ακόμα και αν αυτό προκαλέσει προβλήματα με κάποιους πελάτες.

3.4.2 SSH

Το SSH (Secure Shell) πρωτόκολλο είναι σχετικά καινούργιο και δημιουργήθηκε με σκοπό την αποτροπή επιθέσεων password sniffing σε ένα δίκτυο. Αρχικός σκοπός του ήταν να αντικαταστήσει τα πρωτόκολλα απλού κειμένου (Plaintext protocols) όπως τα telnet και rlogin. Πλέον , το πρωτόκολλο αυτό έχει αναπτυχθεί σε κάτι παραπάνω από ένα τυπικό login πρωτόκολλο ασφάλειας. Εκτός του remote shell , χρησιμοποιείται για μεταφορά αρχείων , ακόμα και για δημιουργία συνδέσεων VPN.

Το SSH πρωτόκολλο αποτελείται από τρία επίπεδα, το επίπεδο μεταφοράς(Transport Layer), το επίπεδο ταυτοποίησης(Authentication Layer) και το επίπεδο σύνδεσης(Connection Layer). Το επίπεδο μεταφοράς παρέχει στο πρωτόκολλο την ασφάλεια που χρειάζεται για την δημιουργία σύνδεσης ενώ στο επίπεδο ταυτοποίησης πραγματοποιείται η ταυτοποίηση του πελάτη-χρήστη. Ο έλεγχος αυτός μπορεί να επιτευχθεί με πολλούς τρόπους , συμπεριλαμβανομένης της γνωστής μεθόδου χρήσης κωδικών , αλλά και της μεθόδου ελέγχου ταυτότητας με δημόσια κλειδιά. Τέλος το επίπεδο σύνδεσης είναι υπεύθυνο για τα κανάλια επικοινωνίας και για τις υπηρεσίες που παρέχει το SSH. Μία σύνδεση μπορεί να εκτελεί πολλά κανάλια ταυτόχρονα και ας είναι διαφορετικού τύπου.
Το SSH χωρίζεται σε δύο μέρη, τα SSH1 και SSH2. Όσον αφορά το SSH1, είναι γεμάτο με ευπάθειες και για αυτό το λόγο δημιουργήθηκε το SSH2 το 2006. Η ανάπτυξη του SSH2 βοήθησε δραματικά στην υποστήριξη του αλλά και ως προς την επιλογή του σε αντίθεση με αυτή του SSH1. Ωστόσο ακόμα και σήμερα υπάρχουν αρκετοί server που χρησιμοποιούν το SSH1.

Παρακάτω αναφέρονται οι επιθέσεις sniffing και session hijacking στο SSH πρωτόκολλο.Η υποκλοπή κωδικών πρόσβασης είναι πιθανή στο πρωτόκολλο SSH. Ο επιτιθέμενος μπορεί να παραμετροποιήσει τη συμβολοσειρά έκδοσης του πρωτόκολλου με σκοπό την υποβάθμιση του σε SSH1, που όπως αναφέραμε παραπάνω , το επίπεδο ασφάλειας του είναι πολύ χαμηλό. Γενικότερα είναι αρκετά εύκολο να επιτευχθεί αυτό , καθώς υπάρχουν και αρκετά εργαλεία που το καθιστούν δυνατό(π.χ. ettercap, dsniff). Ωστόσο , αυτή η επίθεση προϋποθέτει ο SSH server να υποστηρίζει την SSH1 έκδοση.

Στη περίπτωση που ο εξυπηρετητής υποστηρίζει την SSH2 έκδοση , οι πιθανότητες η επίθεση πετύχει είναι πολύ μικρότερες. Το μόνο διαθέσιμο εργαλείο για επιθέσεις MiTM στο SSH2 είναι το jmitm2 και με τη χρήση του είναι πιθανό να υποκλαπούν κωδικοί. Ωστόσο το θύμα ενημερώνεται αυτομάτως για την επίθεση αυτή μέσω του server. Επίσης , εάν για την καθιέρωση της σύνδεσης χρησιμοποιούνται δημόσια κλειδιά (public keys) τότε η επίθεση αυτή δεν είναι δυνατή.

Εάν κάποιος καταφέρει πετυχημένα να αποκτήσει MiTM δικαιώματα μπορεί να παραμετροποιήσει τις εντολές που αποστέλλει ο χρήστης στον server. Για παράδειγμα ο επιτιθέμενος μπορεί να παραβιάσει την σύνδεση πελάτη-διακομιστή (μετά από πετυχημένη αυθεντικοποίηση) μέσω αποσυνδέσεων του χρήστη από τον server. Αυτού του είδους η επίθεση είναι πιθανή μόνο για όσους εξυπηρετητές υποστηρίζουν SSH1. Όσον αφορά το SSH2 δεν υπάρχουν διαθέσιμα εργαλεία, οπότε η επίθεση αυτή δεν είναι δυνατή.

3.5 Επιθέσεις σε SSL/TLS

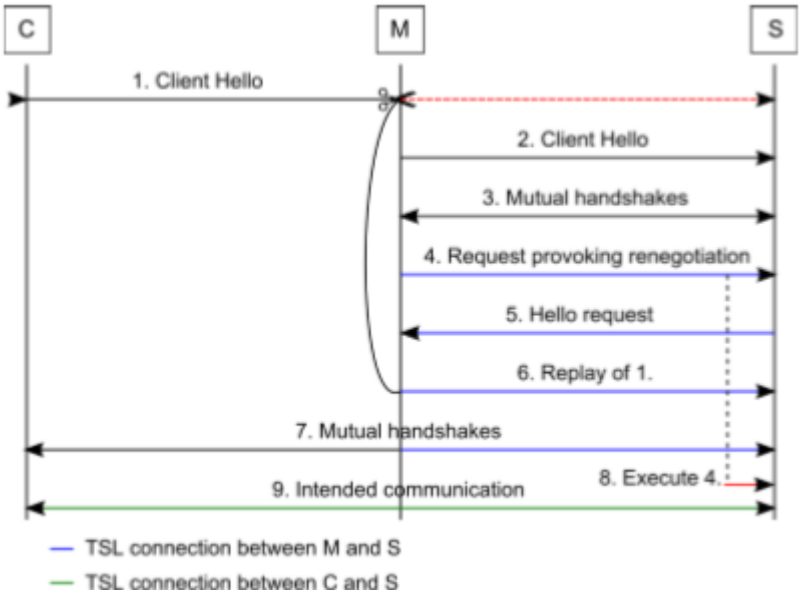
Πολλά πρωτόκολλα δεν έχουν σχεδιαστεί για να είναι ασφαλή όταν χρησιμοποιούνται στο διαδίκτυο. Το 1993 εμφανίστηκε το Secure Network Programming με σκοπό την ασφάλεια μεταφοράς δεδομένων στα ήδη υπάρχοντα πρωτόκολλα. Στη συνέχεια δημιουργήθηκαν τα SSL 1.0 , SSL 2.0 και τέλος το SSL 3.0. Το TLS πρωτοεμφανίστηκε το 1999 ως διάδοχος του SSL 3.0 και το αποκατέστησε τον Ιούνιο του 2015. Η τωρινή έκδοση του TLS είναι η 1.2.
Το SSL/TLS πρωτόκολλο παρέχει μια ασφαλή σήραγγα η οποία με τη σειρά της επιτρέπει τη σύνδεση μεταξύ σημείων χρησιμοποιώντας αλγόριθμους κρυπτογράφησης και τα αντίστοιχα certificates. Με αυτόν τον τρόπο λειτουργίας το πρωτόκολλο αυτό παρέχει εμπιστευτικότητα και ακεραιότητα για τα δεδομένα που αποστέλλονται μεταξύ αυτών των δύο σημείων. Ο γενικός στόχος του SSL/TLS είναι να προστατεύσει τα δεδομένα άλλων πρωτόκολλων που δεν έχουν τη δυνατότητα να το καταφέρουν από μόνα τους. Για παράδειγμα, μια απλή σύνδεση HTTP μπορεί να διοχετευτεί μέσω του SSL/TLS με αποτέλεσμα να αποτραπούν επιθέσεις MitM.Συνεπώς η κρυπτογραφία είναι απαραίτητη για την ομαλή επικοινωνία στο διαδίκτυο και το SSL/TLS είναι μια αξιοπρεπής λύση για αυτό.

3.5.1 Renegotiation Επίθεση

Μόλις εδραιωθεί μια SSL σύνδεση, είναι δυνατό τόσο για το πελάτη όσο και για το διακομιστή, να απαιτήσουν μια επαναδιαπραγμάτευση που ονομάζεται και re-handshake process. Αυτό έχει ως αποτέλεσμα την πραγματοποίηση νέας διαδικασίας χειραψίας μέσω του κρυπτογραφημένου καναλιού. Ο πελάτης μπορεί να ξεκινήσει μια επαναδιαπραγμάτευση στέλνοντας απλά ένα μήνυμα “Client Hello” μέσω του κρυπτογραφημένου καναλιού. Ομοίως, ο διακομιστής μπορεί να στείλει ένα “Hello Request”, με αποτέλεσμα ο πελάτης να αποκρίνεται με ένα νέο “Client Hello”. Ο λόγος που πραγματοποιείται επαναδιαπραγμάτευση είναι για να επιτραπεί η ανανέωση των κλειδιών αυξάνοντας ταυτοχρόνως την ισχύ της χρησιμοποιούμενης Cipher Suite , ή σε περίπτωση ελέγχου ταυτότητας του πελάτη στο διακομιστή, ή για οποιονδήποτε άλλο λόγο.

Έχοντας ξεκινήσει την διαδικασία , η επαναδιαπραγμάτευση γίνεται ακριβώς όπως η αρχική handshake process που αναφέραμε σε προηγούμενη ενότητα. Η νέα χειραψία συνεπάγεται νέα διαπραγμάτευση αλγορίθμων και δημιουργία νέων κλειδιών. Κάθε επαναδιαπραγμάτευση οδηγεί σε ένα νέο αναγνωριστικό περιόδου σύνδεσης. Είναι σημαντικό να επισημανθεί ότι ούτε οι προδιαγραφές του SSL, αλλά ούτε και του TLS απαιτούν την εφαρμογή της επαναδιαπραγμάτευσης. Επομένως, τα δύο μέρη επιτρέπεται να αγνοούν απλώς ένα τέτοιο αίτημα με ένα μήνυμα τύπου “No_renegotiation”. Όμως αν εφαρμοστεί η διαδικασία επαναδιαπραγμάτευσης , θα πρέπει να έχει προτεραιότητα από τα δεδομένα εφαρμογής. Αυτό σημαίνει ότι όλες οι αιτήσεις που λήφθηκαν κατά τη διάρκεια της επαναδιαπραγμάτευσης, συμπεριλαμβανομένου του αιτήματος ενεργοποίησης θα εκτελεστούν μόνο εάν η διαδικασία είναι πετυχημένη.

Γενικότερα έχουμε αναφερθεί στον τρόπο με τον οποίο υλοποιείται μια MitM επίθεση. Πρόσφατα όμως ανακαλύφθηκε μια νέα MitM επίθεση στο SSL πρωτόκολλο η οποία στοχεύει στη διαδικασία επαναδιαπραγμάτευσης. Για τη καλύτερη κατανόηση πρέπει να δούμε το παρακάτω σχήμα.

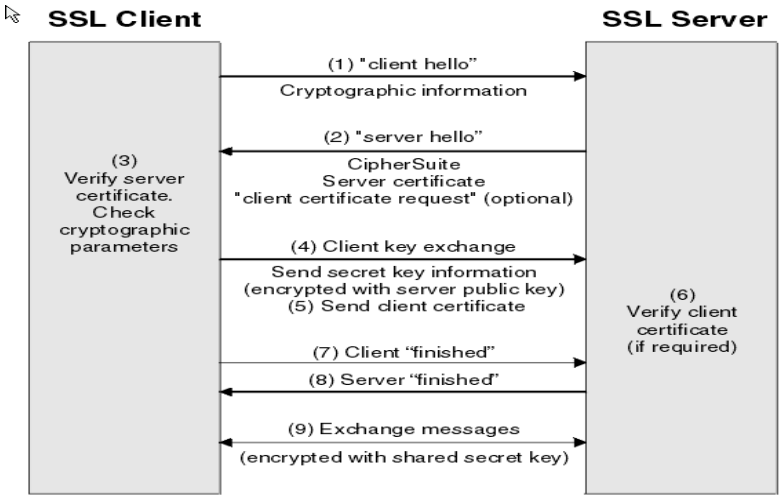


3.7

Αρχικά , ο πελάτης C θέλει να επικοινωνήσει με τον server S μέσω μιας ασφαλής σύνδεσης SSL, και για να ξεκινήσει αυτή η διαδικασία αποστέλλει ένα ClientHello μήνυμα για να πραγματοποιηθεί η διαδικασία χειραψίας, το οποίο όμως μπλοκάρεται από τον επιτιθέμενο M. Αυτός με τη σειρά του στέλνει στον server ένα μήνυμα ClientHello με αποτέλεσμα να εγκατασταθεί μια ασφαλής σύνδεση SSL μεταξύ των M και S. Όμως η διαδικασία επαναδιαπραγμάτευσης καλείται να υλοποιηθεί, είτε επειδή ο M αποστέλλει ένα επιπλέον ClientHello στον S , είτε επειδή πραγματοποίησε κάποιο αίτημα και αναγκάστηκε ο Server να ξεκινήσει από μόνος του τη διαδικασία αυτή. Συνεπώς ο S στέλνει ένα μήνυμα HelloRequest στον M για να εδραιωθεί η επαναδιαπραγμάτευση. Όπως αναφέραμε παραπάνω, η handshake διαδικασία έχει προτεραιότητα έναντι οποιασδήποτε άλλης λειτουργίας, οπότε οτιδήποτε λήφθηκε από τον S έχει υλοποιηθεί σε αυτό. Στη συνέχεια, αντί ο M να απαντήσει στο Hellorequest, ο M επαναπροωθεί το ClientHello(1) που έχει στείλει εξαρχής ο C. Έτσι δημιουργείται μια συνεδρία SSL μεταξύ του C και S πάνω από την σύνδεση του M και S. Όμως ο M δεν μπορεί να διαβάσει ή να παραμετροποιήσει τα δεδομένα που ανταλλάζονται μεταξύ των C και S. Το πρόβλημα όμως εδώ είναι ότι τα δεδομένα που έχει επεξεργαστεί ο S είναι προ-επεξεργασμένα από τον M τα οποία τα είχε αποστείλει στον S κατά τη διάρκεια της προηγούμενης συνεδρίας τους. Άλλο ένα πολύ σοβαρό πρόβλημα αποτελεί το γεγονός ότι ο C δεν γνωρίζει ότι ο S έχει επεξεργαστεί δεδομένα τα οποία δεν έχουν προέλθει από εκείνον. Ουσιαστικά αυτό που έχει καταφέρει ο M είναι να ξεγελάσει τον Server με το να μπερδέψει το αρχικό Handshake process με το renegotiation handshake process[3]. Το πώς αυτού του είδους η επίθεση μπορεί να ενισχυθεί εξαρτάται από πολλούς παράγοντες , όπως ποιος αλγόριθμος κρυπτογράφησης χρησιμοποιείται ή ποια σουίτα αλγορίθμων υποστηρίζεται.

3.5.2 Διαδικασία Χειραψίας

Η διαδικασία χειραψίας(Handshake process) του πρωτόκολλου SSL/TLS αποτελεί μια διαπραγμάτευση μεταξύ δύο πλευρών σε ένα δίκτυο(π.χ. ιστότοπος με διακομιστή) για να προσδιοριστούν οι λεπτομέρειες της σύνδεσής τους. Κατά τη διάρκεια της διαπραγμάτευσης αυτής καθορίζεται ποια έκδοση του SSL/TLS θα χρησιμοποιηθεί κατά τη συνεδρία, ποια σουίτα κρυπτογράφησης(Cipher Suite) θα αναλάβει την κρυπτογράφηση της επικοινωνίας, επαληθεύει την ταυτότητα του διακομιστή(κάποιες φορές και τον πελάτη) και τέλος διαπιστώνει ότι υπάρχει ασφαλής σύνδεση πριν από τη μεταφορά δεδομένων.



3.8

Στην εικόνα 1.2 παρατηρούμε ότι πραγματοποιούνται εννέα βήματα στο σύνολο. Ωστόσο, υπάρχουν διαφορετικές παραλλαγές οι οποίες εξαρτώνται από το τι επιτρέπει το πρωτόκολλο. Οι κοινές διαδικασίες που παρατηρούνται σε όλες αυτές τις παραλλαγές ξεκινούν με τον πελάτη να επικοινωνεί με τον διακομιστή και να ζητά να εδραιωθεί μια ασφαλής σύνδεση μεταξύ τους. Ο διακομιστής απαντά σε αυτό το αίτημα με μια λίστα από το Cipher Suite(αλγόριθμοι κρυπτογράφησης) που διαθέτει. Ο πελάτης με τη σειρά του συγκρίνει τη λίστα που έλαβε με αυτή που διαθέτει για να βρει τους κοινούς αλγόριθμους που υποστηρίζονται και από τους δύο με σκοπό την επιλογή μόνο ενός που θα χρησιμοποιήσουν, και έπειτα ενημερώνει τον διακομιστή. Ο διακομιστής αφότου λάβει το μήνυμα , αποστέλλει το δημόσιο κλειδί που διαθέτει (public key)μαζί με το πιστοποιητικό του (Digital certificate) το οποίο αποτελεί ένα ηλεκτρονικό έγγραφο(Digital certificate). Ο πελάτης στη συνέχεια επιβεβαιώνει την αυθεντικότητα του πιστοποιητικού και αναλαμβάνει την κρυπτογράφηση ενός ιδιωτικού κλειδιού με τη βοήθεια του δημόσιου κλειδιού(του διακομιστή), καθώς και την αποστολή του. Ο διακομιστής λαμβάνει το κλειδί αυτό και το αποκρυπτογραφεί με τη χρήση του ιδιωτικού κλειδιού που κατέχει. Τέλος, ενημερώνει τον πελάτη για την επιτυχημένη αποκρυπτογράφηση του κλειδιού. Σε αυτό το σημείο υπάρχει ένα ιδιωτικό κλειδί το οποίο γνωρίζουν μόνο ο πελάτης και ο διακομιστής. Για αυτό το λόγο ονομάζεται κοινό κρυφό κλειδί και χρησιμοποιείται για την κρυπτογράφηση δεδομένων που ανταλλάζονται καθ' όλη τη διάρκεια της συνεδρίας. Από τη στιγμή που το κλειδί αυτό είναι κοινό , η μέθοδος κρυπτογράφησης είναι συμμετρική και αποτελεί μια άλλη διαδικασία εκτός της χειραψίας[7].

Αξίζει να αναφερθούμε στο γεγονός ότι στο σχήμα 1.2 υπάρχει το βήμα (2) και (5) στα οποία ο διακομιστής και ο πελάτης ζητούν ανταλλαγή των πιστοποιητικών για την ταυτοποίηση και των δύο πλευρών. Ο πελάτης όμως καλείται να αποστείλει το πιστοποιητικό του μόνο αν ζητηθεί από τον διακομιστή και αυτό συμβαίνει όταν η συναλλαγή που πρόκειται να πραγματοποιηθεί χρειάζεται επιπλέον ασφάλεια.(π.χ. Συναλλαγή ποσών μεταξύ δύο τραπεζών.

Κεφάλαιο 4ο

Επίθεση ARP Spoofing

4.1 Εισαγωγή

Για την επιτυχή περάτωση της επίθεσης Man-in-the-middle θα χρησιμοποιήσουμε κάποια εργαλεία που διαθέτει το λογισμικό Kali Linux. Το λογισμικό αυτό είναι δωρεάν και περιέχει περισσότερα από 600 εργαλεία για δοκιμές επιθέσεων στον χώρο της ασφάλειας, αλλά και γενικότερα για την ανάλυση και αξιολόγηση συστημάτων στο δίκτυο. Επιπλέον, αν και τα εργαλεία είναι μεταγλωττισμένα στην Αγγλική διάλεκτο , τα Kali Linux περιλαμβάνουν πολύγλωσση υποστήριξη, επιτρέποντας σε περισσότερους χρήστες να λειτουργούν στη μητρική τους γλώσσα με αποτέλεσμα να εντοπίζουν τα εργαλεία που χρειάζονται πιο εύκολα. Επιπρόσθετα, αποτελεί ένα πλήρως προσαρμόσιμο λογισμικό. Δηλαδή, κάθε χρήστης μπορεί να παραμετροποιήσει το μοντέλο σχεδίασης του σύμφωνα με τις δικές του προτιμήσεις. Στη δική μας περίπτωση θα χρησιμοποιήσουμε τα Kali Linux με τις αρχικές του ρυθμίσεις.
Όπως αναφέραμε στην προηγούμενη παράγραφο, υποστηρίζονται περισσότερα από 600 εργαλεία. Για την υλοποίηση της εργασίας μας θα χρησιμοποιήσουμε τρία από αυτά.

Τα εργαλεία αυτά είναι τα Wireshark, Ettercap και nmap. Η υλοποίηση αυτή λαμβάνει μέρος σε τοπικό ενσύρματο δίκτυο και περιλαμβάνει τις συσκευές που είναι συνδεδεμένες σε αυτό, καθώς και τον δρομολογητή. Χρησιμοποιείται η μέθοδος ARP-Poisoning(Spoofing) στην οποία έχουμε αναφερθεί και αναλύσει στην υποενότητα 3.2.1. Για την καλύτερη κατανόηση της διαδικασίας αυτής πρέπει να γνωρίζουμε για τον τρόπο λειτουργείας των εργαλείων που χρησιμοποιούνται, αλλά και τον λόγο που προτιμήθηκαν.

4.2 Εργαλεία Υλοποίησης

Τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της επίθεσης αποτελούνται από τα Ettercap, Nmap και Wireshark. Είναι σημαντικό να κατανοήσουμε ότι ο καλύτερος τρόπος περάτωσης μιας επίθεσης Man In The Middle(και οποιασδήποτε άλλης) είναι να χρησιμοποιηθεί εξαρχής ένα λογισμικό το οποίο έχει δημιουργηθεί εξαρχής για προηγμένες δοκιμές διείσδυσης(Penetration Testing) και ελέγχους ασφάλειας. Το λογισμικό αυτό είναι το Kali Linux και διαθέτει περισσότερα από 100 δωρεάν εργαλεία έτοιμα προς χρήση. Από όλη αυτή τη γκάμα εργαλείων τα πιο γνωστά για δοκιμές επιθέσεων MitM είναι τα Ettercap,MITMF, John The Ripper. Ωστόσο η επιθέση αυτή μπορεί να πραγματοποιηθεί και με άλλα λογισμικά Linux(Ubuntu, Raspberry Pi OS).Παρακάτω αναλύονται τα εργαλεία που χρησιμοποιήθηκαν, καθώς και μερικά εναλλακτικά εργαλεία που μπορούν να αναπαραστήσουν την επίθεση που πραγματοποιήθηκε.

Ettercap

Το Ettercap αποτελεί ένα δωρεάν εργαλείο ασφάλειας που επιτρέπει την διεξαγωγή επίθεσης MiTM σε τοπικά δίκτυα και μπορεί να χρησιμοποιηθεί για ανάλυση πρωτοκόλλου δικτύου υπολογιστών καθώς και τον έλεγχο ασφαλείας τους. Επίσης, είναι θέση να παρακολουθεί την κυκλοφορία σε ένα τμήμα του δικτύου, να καταγράφει κωδικούς πρόσβασης και να πραγματοποιεί ενεργή υποκλοπή ενάντια σε ορισμένα κοινά πρωτόκολλα. Το Ettercap υποστηρίζει ενεργή και παθητική ανατομή πολλών πρωτοκόλλων (συμπεριλαμβανομένων των κρυπτογραφημένων) και παρέχει πολλές δυνατότητες για ανάλυση δικτύου και υπολογιστών. Επίσης προσφέρει τέσσερις τρόπους λειτουργίας οι οποίοι είναι βασισμένοι στο IP,MAC,ARP και PublicARP. Ωστόσο το λογισμικό προσφέρει πολλές δυνατότητες στο εργαλείο αυτό με επιπλέον επεκτάσεις που μπορούν να εγκατασταθούν και αυτομάτως να υποστηριχθεί επιπλέον υλικό όπως SSH1,SSL καθώς και συλλογή κωδικών για TELNET,FTP,POP,SNMP.Το Ettercap διαθέτει γραφικό περιβάλλον , κάτι που καθιστά τη χρήση του και τη λειτουργεία του πιο εύκολη συγκριτικά με την εκδοχή του που αφορά καθαρά τη χρήση εντολών. Ωστόσο, η υλοποίηση που θα πραγματοποιήσουμε δεν χρησιμοποιεί το γραφικό περιβάλλον του εργαλείου αυτού, αλλά θα χρησιμοποιήσουμε εντολές που θέτουν σε λειτουργεία το Ettercap μέσω του τερματικού.

Εκτός του εργαλείου αυτού, υπάρχουν και άλλα διαθέσιμα εργαλεία που μπορούν να πραγματοποιήσουν τέτοιου είδους επιθέσεις. Ένα από αυτά είναι το Intercepter-NG, το οποίο χρησιμοποιείται για να κατασκοπεύει συσκευές που είναι συνδεδεμένες στο ίδιο τοπικό δίκτυο. Η εφαρμογή αυτή χρησιμοποιείται κυρίως για την υποκλοπή κωδικών και πληροφοριών από άλλους χρήστες. Επίσης επιτρέπει στον χρήστη να ελέγξει την ασφάλεια του δικτύου με την εύρεση πιθανών εισβολών σε αυτό.

Ένα άλλο εργαλείο που μπορεί να χρησιμοποιηθεί είναι το Pirni Pro το οποίο είναι διαθέσιμο για χρήστες IOS λογισμικού και βοηθάει στη παρακολούθηση ενός τμήματος ασύρματου δικτύου. Η εφαρμογή αυτή επιτρέπει στους χρήστες να καταγράφουν κωδικούς πρόσβασης καθώς και άλλες πληροφορίες (π.χ. κείμενα) που εισάγονται από άλλους χρήστες στο δίκτυο αυτό.

Wireshark

Το Wireshark όπως και το Ettercap αποτελεί ένα εργαλείο μέτρησης το οποίο παρέχεται δωρεάν και είναι προεγκατεστημένο στα Kali Linux. Επίσης , είναι ένα από τα καλύτερα εργαλεία ανάλυσης πακέτων δικτύου που υπάρχουν. Είναι σημαντικό να κατανοήσουμε ότι δεν πραγματοποιεί επιθέσεις αλλά βοηθάει δραματικά στην ανάλυση τους και παρέχει πολλές πληροφορίες που χρειάζονται στον χρήστη ανάλογα με αντίστοιχο σκοπό του. Για παράδειγμα, οι διαχειριστές δικτύου χρησιμοποιούν το Wireshark για την αντιμετώπιση προβλημάτων στο δίκτυο τους. Οι μηχανικοί ασφάλειας δικτύου το χρησιμοποιούν για να εξετάσουν τα αντίστοιχα προβλήματα ασφάλειας. Οι προγραμματιστές το χρησιμοποιούν για τον εντοπισμό σφαλμάτων σε πρωτόκολλα. Ωστόσο, το εργαλείο αυτό δεν αποτελεί σύστημα εντοπισμού εισβολών. Δεν προειδοποιεί όταν κάποιος προσπαθεί να εισβάλει ή έχει καταφέρει να εισβάλει στο δίκτυο. Ο εντοπισμός του εισβολέα εξαρτάται αποκλειστικά από τον χρήστη και τις αντίστοιχες γνώσεις του. Αν το δίκτυο έχει παραβιαστεί, το Wireshark θα περιέχει τις πληροφορίες της επίθεσης αυτής, αλλά ο χρήστης θα πρέπει να είναι σε θέση να καταλάβει ποια πακέτα είναι κακόβουλα και ποια όχι. Είναι εξίσου σημαντικό να καταλάβουμε ότι το Wireshark δεν προωθεί αλλά ούτε λαμβάνει πακέτα στο δίκτυο. Αντιθέτως, παρατηρεί και καταγράφει την ροή των πακέτων σε αυτό, δίνοντας τη δυνατότητα στον χρήστη να πραγματοποιήσει ότι ενέργειες παρέχονται από αυτό.

Εκτός του Wireshark, υπάρχουν και εναλλακτικά εργαλεία για ανάλυση και μεταφορά πακέτων. Ένα από αυτά αποτελεί το Cloudshark, το οποίο αναγνωρίζεται ως μια πλατφόρμα που χρησιμοποιείται για ανάλυση πακέτων που έχουν καταγραφεί στο πρόγραμμα περιήγησης από οποιαδήποτε συσκευή. Με αυτόν τον τρόπο βοηθά στην επίλυση του προβλήματος του δικτύου με ταχύτερο ρυθμό. Τέλος, άλλο ένα εργαλείο που μπορεί να χρησιμοποιηθεί είναι το SmartSniff ο οποίο λειτουργεί σε Windows λογισμικό και συλλέγει δεδομένα από το δίκτυο. Επίσης, περιλαμβάνει μια επέκταση που επιτρέπει την καταγραφή πακέτων σε ασύρματο δίκτυο με την προϋπόθεση ότι η συσκευή του χρήστη υποστηρίζει την ασύρματη σύνδεση. Η επικοινωνία που έχει καταγράψει το SmartSniff προβάλλεται σαν μια επικοινωνία μεταξύ ενός εξυπηρετητή με έναν πελάτη.

Nmap

Το Nmap(Network Mapper) είναι ένα δωρεάν και ανοιχτού κώδικα εργαλείο για την εξερεύνηση του δικτύου καθώς και τον έλεγχο της ασφάλειας του. Λειτουργεί ως σαρωτής ασφαλείας και χρησιμοποιείται για να ανακαλύψει κεντρικούς υπολογιστές και άλλες υπηρεσίες σε ένα δίκτυο υπολογιστών, δημιουργώντας έτσι έναν "χάρτη". Η λειτουργεία του παρέχει στο χρήστη μια αναλυτική εικόνα, φανερώνοντας πιθανά προβλήματα και ελλείψεις ασφάλειας.Το εργαλείο αυτό έχει σχεδιαστεί για να σαρώνει γρήγορα μεγάλα δίκτυα και είναι πολύ αποτελεσματικό.

Ενώ το Nmap χρησιμοποιείται συνήθως για τους ελέγχους ασφαλείας, πολλοί διαχειριστές του δικτύου το θεωρούν χρήσιμο για εργασίες ρουτίνας, όπως η απογραφή του δικτύου.

Όσον αφορά τη λειτουργία , το Nmap χρησιμοποιεί ακατέργαστα IP πακέτα για να καθορίσει ποιοι κεντρικοί υπολογιστές είναι διαθέσιμοι στο δίκτυο, ποιες υπηρεσίες(όνομα εφαρμογής, έκδοση) αυτοί οι υπολογιστές προσφέρουν, το λειτουργικό σύστημα , το είδος του πακέτου και φίλτρων που είναι σε χρήση καθώς και πολλά άλλα χρήσιμα χαρακτηριστικά. Το αποτέλεσμα της εκτέλεσης του Nmap αποτελεί μια λίστα από σαρωμένους στόχους, με συμπληρωματικές πληροφορίες οι οποίες εξαρτώνται από τις επιλογές που χρησιμοποιήθηκαν. Μια βασική πληροφορία μεταξύ αυτών είναι ο πίνακας των θυρών(Ports)Τέλος, αξίζει να αναφέρουμε ότι η γραφική έκδοση του Nmap πρέπει να εγκατασταθεί επιπρόσθετα και δεν είναι προεγκατεστημένη στα Kali Linux. Στην δική μας υλοποίηση θα χρησιμοποιήσουμε το εργαλείο αυτό μόνο μέσω του τερματικού.

Ακόμα ένα δημοφιλές εργαλείο που λειτουργεί σαν το Nmap αποτελεί το Unicornscan και επιλέγεται κυρίως λόγω της δυνατότητας του να υλοποιεί ασύγχρονη σάρωση TCP και UDP πακέτων. Επίσης ανακαλύπτει μοτίβα τα οποία μπορεί ενδεχομένως να προσφέρουν σημαντικές πληροφορίες.

4.3 Εφαρμογή MiTM επίθεσης

Στην πρώτη φάση της υλοποίησης της επίθεσης αυτής εγκαταστήσαμε το λογισμικό Kali Linux σε ένα Virtual Machine(Εικονική μηχανή). Πρέπει να αναφέρουμε ότι για να εγκατασταθεί το λογισμικό αυτό χρειάζεται τουλάχιστον 20 GB ελεύθερου χώρου στον δίσκο. Επίσης ,αξίζει να σημειωθεί ότι η επίθεση αυτή μπορεί να υλοποιηθεί και με λογισμικό Ubuntu αλλά και με χρήση Raspberry Pi. Για το τέλος της πρώτης φάσης πρέπει να εγκαταστήσουμε στο τερματικό τα αντίστοιχα Libraries των εργαλείων αυτών. Κάποια από αυτά μπορεί να είναι ήδη προεγκατεστημένα , αλλά σε περίπτωση τυχόν αναβαθμίσεων είναι προτιμότερο να το πραγματοποιήσουμε.

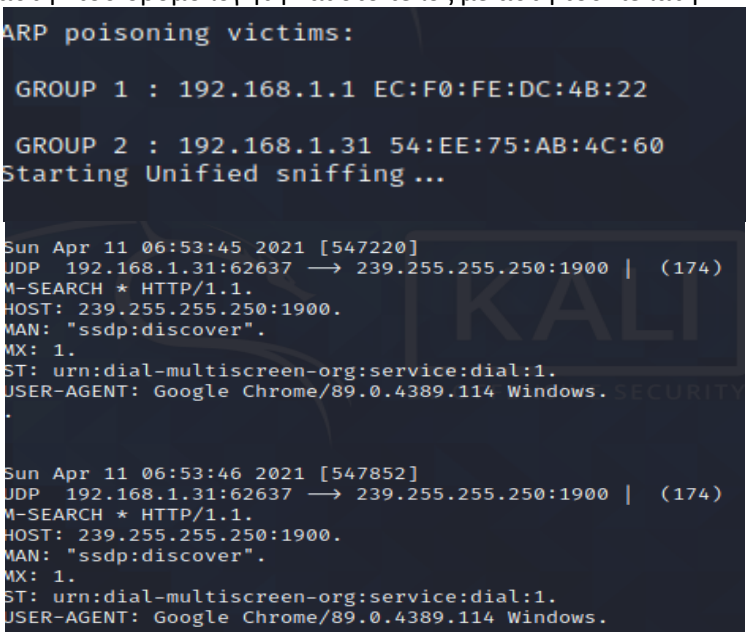
Για την εγκατάσταση τους πρέπει να έχουμε δικαιώματα διαχειριστή, τα οποία τα λαμβάνουμε με τη χρήση της εντολής sudo – i όπου μας ζητάει για επαλήθευση τον κωδικό σύνδεσης μας για τα Linux. Όταν λάβουμε τα δικαιώματα διαχειριστή μπορούμε να προχωρήσουμε με την εγκατάσταση των βιβλιοθηκών με τη χρήση των εντολών apt install wireshark , apt install nmap και apt install ettercap-text-only αντίστοιχα.Για τη χρήση του Ettercap εγκαθιστούμε μόνο την έκδοση που χρησιμοποιείται για τη χρήση κώδικα και όχι για το γραφικό περιβάλλον.Στη συνέχεια βρίσκουμε τον στόχο της επίθεσης. Για να το καταφέρουμε αυτό πρέπει να χρησιμοποιήσουμε το εργαλείο Nmap για να δούμε ποιοι κεντρικοί υπολογιστές είναι ενεργοί στο τοπικό δίκτυο. Για να εκτελέσουμε την Nmap χρειαζόμαστε την διεύθυνση IP του δρομολογητή(Router). Η διεύθυνση αυτή μπορεί να βρεθεί με τη χρήση της εντολής ifconfig για Linux και ipconfig για Windows λογισμικά. Η διεύθυνση αυτή είναι η Default Gateway και στη δική μας περίπτωση είναι η 192.168.1.1.
Εφόσον γνωρίζουμε πλέον το υποδίκτυο που θέλουμε να "χαρτογραφήσουμε" εκτελούμε την εντολή sudo nmap -sn 192.168.1.1/24. Η επιλογή -sn υποδηλώνει τη σάρωση του δικτύου (scan network) και τέλος έχουμε την IP του δρομολογητή καθώς και την επέκταση /24 η οποία αναφέρεται στην Subnet Mask(255.255.255.0) η οποία δηλώνει το πλήθος των IP διευθύνσεων που υποστηρίζεται (το /24 υποστηρίζει έως και 256 IP διευθύνσεις κεντρικών υπολογιστών).

```
┌──(root💀 themis)-[~]
└─# sudo nmap -sn 192.168.1.1/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-11 06:31 CDT
Nmap scan report for speedport-entry-2i.ote.gr (192.168.1.1)
Host is up (0.0032s latency).
MAC Address: EC:F0:FE:DC:4B:22 (zte)
Nmap scan report for huawei_y6_prime_2018-e82c (192.168.1.23)
Host is up.
MAC Address: D0:D7:83:DE:34:A6 (Huawei Technologies)
Nmap scan report for laptop-h3i1ud3v (192.168.1.31)
Host is up (0.00025s latency).
MAC Address: 54:EE:75:AB:4C:60 (Wistron InfoComm(Kunshan)Co.)
Nmap scan report for themis (192.168.1.46)
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 1.93 seconds
```

4.1

Παρατηρούμε ότι βρέθηκαν 4 ενεργοί κεντρικοί υπολογιστές στο τοπικό δίκτυο συμπεριλαμβανομένων τις MAC διευθύνσεις , το όνομα ,τις IP διεθύνσεις καθώς και τον πάροχο τους.
Σε αυτό το κομμάτι της υλοποίησης γνωρίζουμε πλέον την IP διεύθυνση της εικονικής μηχανής που λειτουργούμε (192.168.1.46), την IP διεύθυνση του υπολογιστή(192.168.1.31), την IP διεύθυνση μιας κινητής συσκευής(192.168.1.23) καθώς και του δρομολογητή(192.168.1.1).
Ως στόχο καθιερώνουμε τον κεντρικό υπολογιστή με διεύθυνση 192.168.1.31, ο οποίος θα λειτουργήσει ως πελάτης στην επερχόμενη επικοινωνία.

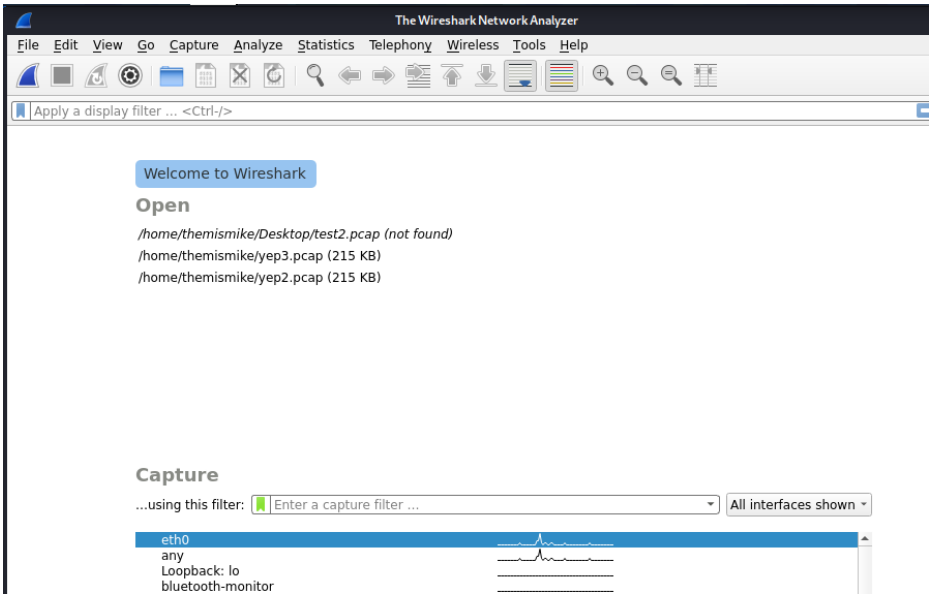
Έπειτα εκτελούμε την man-in-the-middle επίθεση. Όπως έχουμε ήδη αναλύσει, πρέπει να καταφέρουμε να εισβάλλουμε στην μέση της επικοινωνίας του πελάτη(192.168.1.31) και του δρομολογητή(192.168.1.1). Αυτό που θα καταφέρουμε με αυτήν την επίθεση είναι να αναγκάσουμε το δρομολογητή να αποστείλει σε εμάς τα πακέτα που προορίζονταν για τον πελάτη, και αντιστοίχως τον πελάτη να μας αποστείλει τα πακέτα που προορίζονταν για τον δρομολογητή.Αυτό καθίσταται δυνατό με το είδος της επίθεσης που έχουμε επιλέξει(ARP Poisoning).Η επικοινωνία μεταξύ των συσκευών γίνεται με τη χρήση των MAC διευθύνσεων. Με τη χρήση του εργαλείου ettercap και της επίθεσης ARP Poisoning αλλάζουμε την MAC διεύθυνση προορισμού με αυτήν της συσκευής μας (192.168.1.46). Συνεπώς, πριν επικοινωνήσουν ο πελάτης και ο δρομολογητής, όλα τα πακέτα τους θα εμφανίζονται στη δική μας συσκευή πρώτα, και έπειτα θα προωθούνται στον αρχικό τους προορισμό.
Για το πρακτικό κομμάτι της επίθεσης εκτελούμε την εντολή
sudo ettercap -T -S -i eth0 -M arp:remote /192.168.1.1 // /192.168.1.31//.
Η επιλογή -T αναπαριστά το κομμάτι των εντολών(text-only) και όχι το γραφικό περιβάλλον , το -S αποκλείει το SSL πρωτόκολλο από τα πακέτα που θα λαμβάνουμε και το -i δηλώνει τη διεπαφή eth0 που λειτουργούμε(θα εξηγήσουμε παρακάτω πιο αναλυτικά).
Η επιλογή -M υποδηλώνει ότι η επίθεση είναι man-in-the-middle και έπειτα δηλώνουμε το είδος της επίθεσης(arp:remote). Τέλος εισάγουμε τις IP διευθύνσεις που έχουμε ως στόχο, ξεκινώντας με αυτήν του δρομολογητή και στο τέλος με αυτή του πελάτη.



4.2

Είναι σημαντικό να κατανοήσουμε ότι η επίθεση είναι ενεργή και ότι τα πακέτα που αποστέλλονται είναι πολλά και έχουν διαφορετικό σκοπό το ένα με το άλλο. Στην εικόνα 4.2 έχουμε ως παράδειγμα δύο πακέτα UDP από τον πελάτη(192.168.1.31) προς έναν κεντρικό υπολογιστή με διεύθυνση 239.255.255.250 . Επειδή όμως για πραγματοποιηθεί αυτό το αίτημα πρέπει να προωθηθεί μέσω του δρομολογητή του τοπικού δικτύου μας , από τη στιγμή που βρισκόμαστε ενδιάμεσα στην επικοινωνία τους μπορούμε και “κλέβουμε” τα πακέτα αυτά.

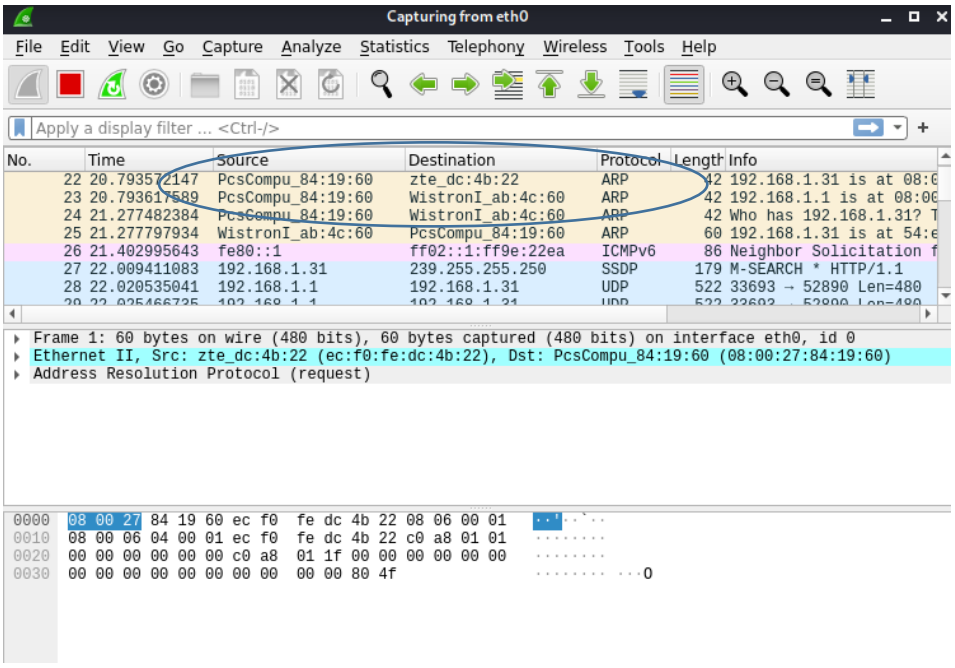
Στη συνέχεια αναλύουμε τα πακέτα αυτά πιο διεξοδικά με τη χρήση του εργαλείου Wireshark. Για να εκτελέσουμε το wireshark ανοίγουμε ένα επιπλέον παράθυρο τερματικού και το ανοίγουμε με την εντολή sudo wireshark(μπορούμε και κατευθείαν από την εφαρμογή).



4.3

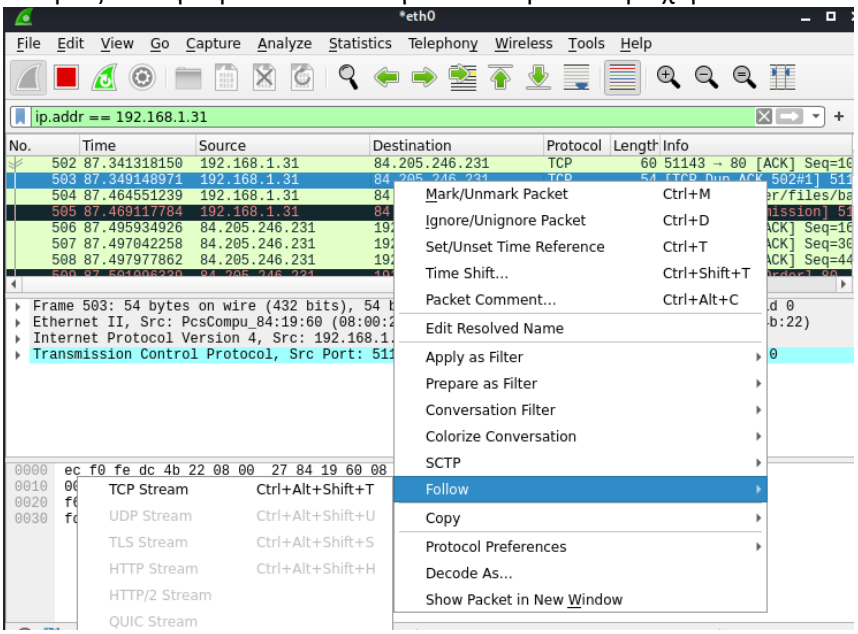
Παρατηρούμε ότι για να ξεκινήσει η διαδικασία ανάλυσης των πακέτων πρέπει να επιλέξουμε τη διεπαφή σύνδεσης μας. Στη δική μας περίπτωση επειδή η σύνδεση είναι ενσύρματη , η διεπαφή είναι η eth0 (η οποία χρησιμοποιήθηκε στη διαδικασία της επίθεσης με το εργαλείο ettercap). Αξίζει να αναφέρουμε ότι εάν η σύνδεση ήταν ασύρματη, η διεπαφή αυτή θα είχε ονομασία wlan0.

Μετά την επιλογή της διεπαφής ξεκινάει η ανάλυση των πακέτων. Μπορούμε να δούμε πιο οργανωμένα τα πακέτα που εμφανίζονταν στο τερματικό. Επίσης μπορούμε να δούμε ξεκάθαρα την πηγή αποστολής, τον προορισμό , καθώς και το πρωτόκολλο που χρησιμοποιήθηκε για την επικοινωνία. Επιπρόσθετα το wireshark μας δίνει πολλές επιλογές για την ανάλυση των πακέτων αυτών με τη δυνατότητα χρήσης αρκετών εργαλείων όπως στατιστικές αναλύσεις , εμφάνιση των αποτελεσμάτων και άλλα.



4.4

Στο επόμενο βήμα της υλοποίησης χρησιμοποιούμε φιλτραρισμένη αναζήτηση για να βρούμε τα πακέτα που μεταφέρθηκαν μέσω του πελάτη (192.168.1.31).
Για να γίνει αυτό πληκτρολογούμε στην αναζήτηση ip.addr == 192.168.1.31. Στη συνέχεια βρίσκουμε ένα τυχαίο πακέτο TCP και το επιλέγουμε. Οι επιλογές που μας δίνονται είναι πολλές, αλλά εμείς επιλέγουμε το “Follow” για να δούμε το περιεχόμενο του πακέτου.



4.5

Όταν επιλεχθεί το “Follow” μπορούμε να δούμε σε ένα αναδυόμενο παράθυρο το περιεχόμενο του πακέτου. Ωστόσο , το TCP πρωτόκολλο κρυπτογραφεί τα δεδομένα της επικοινωνίας αυτής, συνεπώς είναι σχεδόν αδύνατο να υποκλέψουμε το περιεχόμενο.

4.6

Η εικόνα 4.6 περιέχει την κρυπτογραφημένη συνομιλία TCP. Η μόνη πληροφορία που μπορούμε να εκλάβουμε είναι ότι οι κόκκινες γραμμές αντιπροσωπεύουν μηνύματα από τον πελάτη και οι μπλε από τον εξυπηρετητή.

Δυστυχώς, ακόμα και σήμερα υπάρχουν πολλοί ιστότοποι που δεν χρησιμοποιούν πρωτόκολλα ασφαλείας για να κρυπτογραφούν την επικοινωνία τους.
Στην επόμενη φάση της υλοποίησης , βρήκαμε έναν ιστότοπο(Από τη συσκευή του πελάτη), που χρησιμοποιεί HTTP πρωτόκολλο και διαθέτει την επιλογή εισαγωγής στοιχείων. Αφότου εισάγουμε τα λανθασμένα στοιχεία και ολοκληρώσουμε την διαδικασία, βρίσκουμε την επικοινωνία αυτή μέσω του wireshark και τη χρήση της φιλτραρισμένης αναζήτησης ip.addr == 192.168.1.31 && http .
Η αναζήτηση αυτή μας προβάλλει όλα τα πακέτα που χρησιμοποιούν http πρωτόκολλο. Όταν βρούμε το πακέτο αυτό, ακολουθούμε την ίδια διαδικασία που χρησιμοποιήσαμε με το TCP πρωτόκολλο.

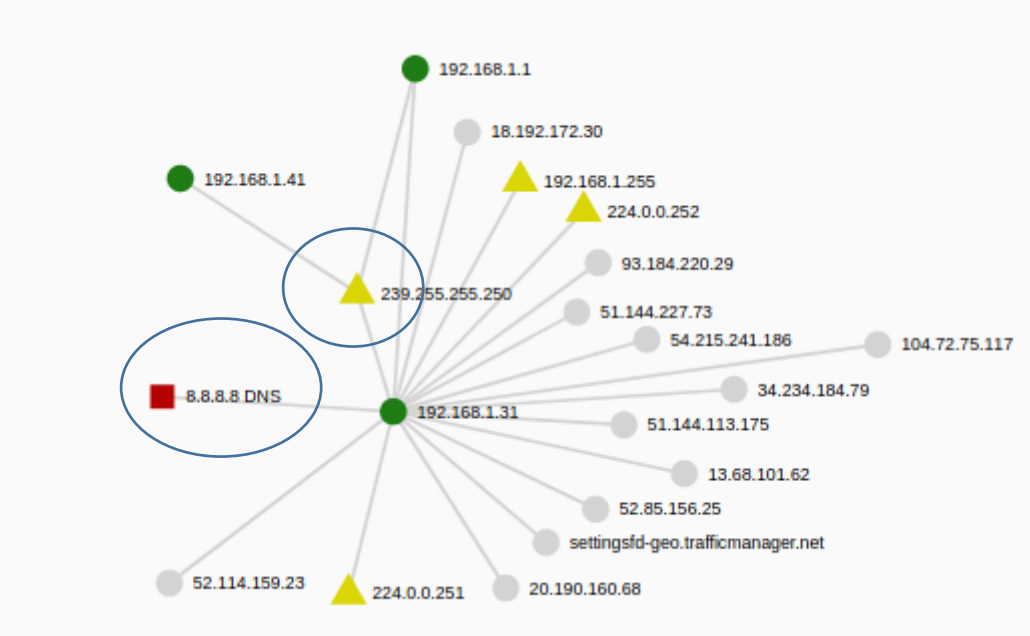
```
POST /secured/newuser.php HTTP/1.1
Host: testphp.vulnweb.com
Connection: keep-alive
Content-Length: 173
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://testphp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/89.0.4389.114 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/
*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://testphp.vulnweb.com/signup.php
Accept-Encoding: gzip, deflate
Accept-Language: el-GR,el;q=0.9,en;q=0.8

uuname=themis&upass=nonprotected1&upass2=nonprotected1&urname=Themis&ucc=5698125789653215&uemail=
notvalidemail%40gmail.com&uphone=9875412698&uaddress=Ellada+23&signup=signupHTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sun, 11 Apr 2021 12:14:14 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
Content-Encoding: gzip

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
loose.dtd">
```

4.7

Αντίθετα με το πρωτόκολλο TCP, οι πληροφορίες δεν είναι κρυπτογραφημένες και μπορούμε να διακρίνουμε ξεκάθαρα το περιεχόμενο. Επίσης μπορούμε να διακρίνουμε τα στοιχεία που εισήγαμε στην ιστοσελίδα. Πιο συγκεκριμένα βλέπουμε ότι σαν όνομα χρήστη είναι το “themis” , ο κωδικός πρόσβασης είναι “nonprotected1” καθώς και τις υπόλοιπες πληροφορίες. Επίσης παρατηρούμε ότι μπορούμε να διαβάσουμε και τον κώδικα που χρησιμοποιήθηκε για τη δημιουργία της ιστοσελίδας αυτής.
Σε αυτή τη φάση της εργασίας έχουμε καταφέρει να εκτελέσουμε με επιτυχία μια man-in-the-middle επίθεση και να υποκλέψουμε σημαντικά πακέτα και στοιχεία χρήστη.

4.4 Συμπεράσματα

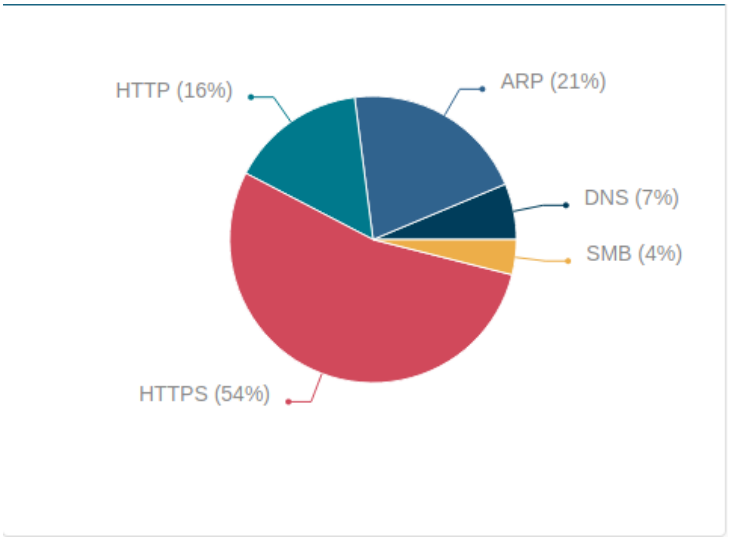
Για τη διεξαγωγή των συμπερασμάτων αποτυπώνουμε τα αποτελέσματα της ανάλυσης πακέτων που υποκλέψαμε μέσω των εργαλείων με τη χρήση διαγραμμάτων. Για να το καταφέρουμε αυτό χρησιμοποιούμε μια από τις επιλογές που μας προσφέρει το wireshark , η οποία μας επιτρέπει να αποθηκεύσουμε την ανάλυση αυτή σε ένα αρχείο τύπου .pcap . Στη συνέχεια χρησιμοποιούμε τον ιστότοπο apackets.com (αποτελεί ιστότοπο ανάλυσης αρχείων .pcap).



4.8

Το παραπάνω σχήμα μας παρουσιάζει τα μονοπάτια που χρησιμοποιήθηκαν κατά τη διάρκεια της διαδικασίας. Πρέπει να σημειωθεί ότι δεν αποτελείται από όλες τις συνδέσεις που πραγματοποιήθηκαν(π.χ. TCP, ICMPv6), διότι το σχήμα θα ήταν σχεδόν αδύνατο να αναλυθεί λόγω του πλήθους των συνδέσεων. Επίσης αξίζει να αναφέρουμε ότι κατά τη διάρκεια της υλοποίησης το σύνολο των πακέτων που μεταφέρθηκαν ήταν 163006. Συνεπώς , μετά τον αποκλεισμό αυτών καταφέραμε να περιορίσουμε και να “χαρτογραφήσουμε” ένα μέρος της διαδικασίας.

Τα πράσινα σημεία αντιπροσωπεύουν τις IP διευθύνσεις των υπολογιστών και τα κίτρινα αποτελούν συνδέσεις UDP πακέτων, όπου παρατηρούμε ότι ο πελάτης, ο δρομολογητής αλλα και η εικονική συσκευή (Virtual machine) έχουν χρησιμοποιήσει τον ίδιο πολυδιανομέα διευθύνσεων. Επίσης στο διάγραμμα αυτό φαίνεται και μία DNS σύνδεση σε μία βάση δεδομένων. Τέλος, οι γκρι συνδέσεις αποτελούν κάποιες άλλες επικοινωνίες με εξυπηρετητές.



4.9

Τέλος, αξίζει να συγκρίνουμε την εμφάνιση συγκεκριμένων πρωτοκόλλων (HTTP, HTTPS, DNS, ARP, SMB). Το SMB αποτελεί πρωτόκολλο δικτύου που χρησιμοποιούν οι εξυπηρετητές για να επιτρέψουν ή να μπλοκάρουν την επικοινωνία με τους πελάτες. Στο διάγραμμα αυτό παρατηρούμε ότι το 16% μεταξύ αυτών των πρωτόκολλων αποτελεί το HTTP. Εάν συμπεριλάβουμε και άλλα πρωτόκολλα στη σύγκριση μας το ποσοστό αυτό θα ήταν μικρότερο. Ωστόσο πρέπει να κατανοήσουμε ότι αν και μικρό ποσοστό, αποτελεί σημαντικό πρόβλημα στο χώρο της ασφάλειας, διότι τα πρωτόκολλα που δεν χρησιμοποιούν κρυπτογραφικές μεθόδους για να προστατέψουν τα δεδομένα τους, η πιθανότητα υποκλοπής αυτών είναι αρκετά μεγάλη.

Βιβλιογραφία

1. M. Conti, N. Dragoni and V. Lesyk, “A Survey of Man In The Middle Attacks,” in IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2027-2051, third quarter 2016.
2. G. Nath Nayak and S. G. Samaddar, “Different flavours of man-in-the- middle attack, consequences and feasible solutions,” in Proc. 3rd IEEE Int. Conf. Comput. Sci. Inf. Technol. (ICCSIT), 2010, vol. 5, pp. 491–495.
3. J. Du, X. Li, and H. Huang, “A study of man-in-the-middle attack based on SSL certificate interaction,” in Proc. 1st Int. Conf. Instrum. Meas. Comput. Commun. Control, 2011, pp. 445–448.
4. S. Y. Nam, S. Djuraev, and M. Park, “Collaborative approach to mitigating ARP poisoning-based man-in-the-middle attacks,” Comput. Netw., vol. 57, no. 18, pp. 3866–3884, 2013.
5. Sebastian Anthony. Chrome finally kills off the HTTP-HTTPS “mixed content” warning. University of Michigan. Fault based attack of RSA Authentication.
6. https://www.ssl.com/article/ssl-tls-handshake-overview/
7. Seung Yeob Nam, Dongwon Kim and Jeongeun Kim, Enhanced ARP: Preventing ARP Poisoning-Based Man-In-The-Middle Attacks, Communication Letters 14: 187-189, 2010.
8. https://brilliant.org/wiki/rsa-encryption/
9. https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard
10. Σπύρος Κοκολάκης, (2000), «Ανάπτυξη και Διαχείριση Ασφάλειας Πληροφοριακών Συστημάτων: Εννοιολογικό Πλαίσιο, Μεθοδολογίες και Εργαλεία», Διδακτορική Διατριβή, Τμήμα Πληροφορικής, Οικονομικό Πανεπιστήμιο Αθηνών.
11. J. Postel, J. Reynolds and ISI. FILE TRANSFER PROTOCOL (FTP). Oct. 1985.
12. C. L. Abad and R. I. Bonilla, “An analysis on the schemes for detecting and preventing ARP cache poisoning attacks,” in Proc. 27th Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW’07), 2007, p. 60.
13. F. Fayyaz and H. Rasheed, “Using JPCAP to prevent man-in-the-middle attacks in a local area network environment,” IEEE Potentials, vol. 31, no. 4, pp. 35–37, Jul./Aug. 2012.
14. N. Alexiou, S. Basagiannis, P. Katsaros, T. Dashpande, and S. A. Smolka, “Formal analysis of the Kaminsky DNS cache-poisoning attack using probabilistic model checking,” in Proc. IEEE 12th Int. Symp. High-Assur. Syst. Eng. (HASE), 2010, pp. 94–103
15. S. Duangphasuk, S. Kungpisdan, and S. Hankla, “Design and implementation of improved security protocols for DHCP using digital certificates,” in Proc. 17th IEEE Int. Conf. Netw. (ICON), 2011, pp. 287–292.
16. Komaram, Shiva Naresh. Ethical Hacking Playbook and Cyber Security : For Absolute Beginners.
17. Wesley Chou “Inside SSL: The secure sockets layer protocol” IEEE IT Professional Volume: 4 Issue: 4, Jul/Aug 2002 Page(s): 47 -52
18. Chappell, Laura. "Inside the TCP Handshake." NetWare Connection (2000).
19. Y. W. Ju, K. H. Song, E. J. Lee, and Y. T. Shin, “Cache poisoning detection method for improving security of recursive DNS,” in Proc. 9th Int. Conf. Adv. Commun. Technol., 2007, vol. 3, pp. 1961–1965.