UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

"ANALISI DELLA SICUREZZA DI UN DISPOSITIVO IOT"

Relatore: Prof. / Dott Mauro Migliardi

Laureando/a: Matteo Cischele

ANNO ACCADEMICO 2021-2022
Data di laurea 21/09/2022

Abstract

Con l'elaborato si vuole andare ad analizzare la sicurezza informatica di un dispositivo IoT tramite test pratici su un dispositivo reale. Inizialmente si dovranno definire i concetti di dispositivo IoT e di sicurezza informatica con un focus sul penetration testing e quindi sugli attacchi informatici. Seguirà l'introduzione al test che si incentrerà su attacchi informatici che verranno propriamente descritti come il dispositivo reale sul quale questi saranno eseguiti Ogni attacco sarà descritto da un punto di vista tecnico ma anche da metriche che riescano a mettere i vari attacchi a confronto concentrandosi sull' efficienza di ciascuno di questi. Alla fine si confronteranno le varie metriche per capire quali attacchi siano stati più efficaci e come il dispositivo reale si sia comportato quando esposto alle diverse minacce.

Sommario

Introduzione

La sicurezza è uno dei problemi fondamentali nei dispositivi informatici. La crescita di connessioni e complessità di queste fa sì che ogni anno la sicurezza informatica sia un maggiore problema rispetto al passato. Per valutare la sicurezza dei dispositivi prima di un rilascio sul mercato o anche semplicemente per verificare eventuali falle nella sicurezza di una rete domestica si eseguono dei penetration test.

Se per una rete domestica è solo una questione di controllo per un'azienda la parte informatica diventa di anno in anno un asset fondamentale e bisogna assicurarsi sia ben difeso e resistente ad ogni possibile attacco conosciuto. Qui il penetration testing diventa fondamentale e parte della filiera di vendita e mantenimento di reti complesse. I sistemi informatici dovranno passare vari livelli di testing per verificarne la sicurezza e quindi il loro valore sul mercato.

La sicurezza diventa tanto più fondamentale quando si parla di domotica e/o dispositivi Iot progettati per essere parte di una smart home. In questo caso infatti i dispositivi che subiscono l'attacco sono molto vicini a una persona reale e ciò rende il potenziale attacco più pericoloso. La sicurezza dei dispositivi Iot è fondamentale.

L'IoT (Internet of Things) si sta diffondendo rapidamente su una moltitudine di domini differenti: salute personale, monitoraggio ambientale, automazione della casa, mobilità smart e Industria 4.0. Il numero e la varietà di dispositivi IoT è cresciuto rapidamente con stime che indicavano come avremmo superato la soglia dei 50 miliardi di dispositivi connessi a Internet già nel 2020. [1]

Di conseguenza sempre più dispositivi IoT sono presenti in ambienti pubblici e privati diventando con il passare del tempo oggetti comuni nella vita di tutti i giorni. Chiaramente in uno scenario del genere la cybersecurity diventa fondamentale per evitare diffusione di informazioni private, attacchi denial of service e accesso ai dispositivi da parte di persone non autorizzate. Sfortunatamente molti dispositivi IoT di basso livello non supportano pesanti meccanismi di cybersecurity quindi sono spesso bersagliati o usati per compiere attacchi informatici.

Come anticipato nel riassunto, con l'elaborato si vuole andare ad analizzare la sicurezza informatica di un dispositivo IoT tramite test pratici su un dispositivo reale. Nel mondo in cui viviamo diventa di giorno in giorno più fondamentale rendersi conto di quanto le nuove tecnologie possano essere un'arma a doppio taglio. Da un lato queste innovazioni ci offrono modi nuovi di affrontare la vita di tutti i giorni facilitando azioni prima complesse ma che allo stesso tempo ci pongono nuove sfide. Lo scopo degli attacchi è quello di violare un dispositivo IoT pensato per essere parte di una smart home quindi a stretto contatto con la vita privata di ciascuno. Ciò dovrebbe far capire quanto il mondo informatico ci arrivi vicino ogni giorno senza che ce ne si accorga e che spesso l'essere inconsapevoli che la tecnologia sia un rischio è il principale motivo per cui si ha un crescendo di crimini informatici. I progettisti infatti possono creare sistemi molto complessi e con sicurezza adeguata ma la negligenza di un solo utente può compromettere l'integrità del sistema stesso. Sarebbe importante per tutti gli

utenti costruire una cultura sulla sicurezza informatica anche imparando e andando a fondo sui tecnicismi di ciascun dispositivo. Capendo che chi mantiene la sicurezza il dispositivo sia in continua lotta con chi sta cercando di approfittare in maniera illecita del dispositivo stesso.

Nell'elaborato si inizierà parlando dei due blocchi fondamentali del test di sicurezza che si farà poi sul dispositivo IoT: penetration testing e dispositivi IoT. Verrà spiegato in cosa consiste un penetration test e le motivazioni che ci spingono verso questo approccio alla sicurezza informatica per poi andare a descrivere più nel dettaglio il soggetto del test stesso, il dispositivo IoT, parlando della diffusione di questa tecnologia e dei possibili rischi che porta con sé. Prima del test vero e proprio si passerà poi in rassegna la scaletta precisa dei passaggi che verranno svolti durante il test. Il test svolto infatti dovrà essere pensato per essere il più possibile schematico, modulare e ripetibile. Così che sia facile trarre conclusioni dai risultati prodotti nel test e che tutto il procedimento sia facilmente replicabile in un secondo momento in maniera schematica. Si arriva così alla parte più importante quella di test che contiene tanto una parte di spiegazione dei test effettuati quanto l'uso di metriche per trarre poi conclusioni sui test effettuati. Nell' ultima fase la sicurezza del dispositivo potrà finalmente essere valutata a partire da un'analisi dei parametri delle metriche così da poter trarre delle conclusioni. Nella fase di testing la tipologia di attacco è spiegata anche in funzione dei danni che potrebbe fare se portata su scale diverse ma principalmente in che modo riesce a produrre questi danni. Si otterrà questo con spiegazioni dettagliate dell'attacco utilizzato in tutte le sue forme e modi d'uso. Per capire al meglio la vulnerabilità degli oggetti che ci circondano nella vita di tutti i giorni.

1 Penetration testing

Come detto nell'introduzione è la metodologia che verrà utilizzata per valutare la sicurezza della rete analizzata e dei dispositivi presenti in questa. [2]

Nello specifico il penetration testing è una serie di attività svolte per verificare e sfruttare eventuali vulnerabilità della sicurezza. Aiuta a confermare l'efficacia o inefficacia delle misure di sicurezza che sono state implementate.

Il penetration testing non va confuso con i test funzionali di sicurezza: questi garantiscono il corretto funzionamento del sistema mentre il penetration testing dimostra la difficoltà di accesso alle funzionalità di sistema per un utente non autorizzato.
Il penetration testing viene svolto per identificare e possibilmente risolvere vulnerabilità nel sistema prima che malintenzionati possano sfruttarle.

1.1 Strategie di penetration test

Tre tipologie di penetration test a seconda del grado di conoscenza della rete all' inizio del test

- Black box
- White box
- Gray box

Il penetration test Black Box è quello che si avvicina più alla situazione reale di un malintenzionato che provi ad accedere alla rete senza nessuna conoscenza pregressa della rete stessa e dei dispositivi a questa connessi.
Al contrario nel penetration test white box chi si occupa di mantenere la rete sicura passa volontariamente informazioni sulla rete stessa a chi va ad eseguire il test di modo che il penetration test sia più capillare.
Avendo maggiori informazioni sulla rete infatti il compito di chi esegue il penetration test diventa non più quello di impersonare un malintenzionato ma più quello di testare ogni specifica difesa.
Nel penetration test grey box chi si occupa di mantenere la rete sicura passa a chi esegue il penetration test informazioni parziali sulla rete questo velocizza il processo che comunque resta più vicino all' idea di provare ad accedere alla rete come utente non autorizzato.
Le informazioni passate a chi esegue il penetration test potrebbero anche essere date in fasi avanzate del test per valutare la solidità di solo alcuni punti della rete.

1.2 Tipologie di penetration test

Una rete aziendale o domestica è composta di vari dispositivi, connessioni e servizi garantiti dalla rete stessa. Il penetration testing opera in tre aree diverse:

- Struttura fisica del sistema
- Struttura logica del sistema
- Risposta del sistema

Ogni area avrà sarà soggetta a tipologie di attacco differenti. Un esempio di attacco fisico può essere un penetration testing a livello di network: nell' attacco si andranno a colpire le connessioni tra i vari dispositivi all' interno di una rete. Chi svolge il test andrà a svolgere un'analisi delle varie connessioni trovando eventuali brecce nella sicurezza di queste e sfruttandole.

Attaccare la struttura logica vuol dire compiere attacchi a livello di applicazione stando distanti dal mondo fisico delle comunicazioni e dell'hardware come avevamo visto nell' attacco precedente e concentrandosi solo sul software. L'application penetration testing è infatti una simulazione di attacco pensata per dimostrare l'efficacia dei controlli di sicurezza di un'applicazione mettendo in risalto i rischi causati da possibili abusi del sistema.

1.3 Come eseguire un penetration test

Il penetration testing non è solamente un uso di tool automatizzati seguita dalla redazione di report tecnici.
Dovrebbe invece provvedere un'indicazione concisa ed efficace su come si possa migliorare la sicurezza della rete in seguito cercando di minimizzare le vulnerabilità che sono state esposte durante la fase di testing.
Un punto critico sta nell' approccio metodico al test.
Dovrà essere usato un approccio sistematico e scientifico per far sì che la documentazione prodotta al fine del test sia tanto logica quanto efficace per questo il test va diviso in diverse fasi.

1.4 Fasi penetration test

- Test preparation
- Information Gathering
- Vulnerability Analysis
- Vulnerability Exploits
- Test Analysis

Nella fase di test preparation si va a scegliere quale strategia e quale tipologia di test verranno utilizzate durante il test ma anche quali sono a grandi linee gli obiettivi del test stesso e le tempistiche di questo.
A livello aziendale in questa fase si procederà anche con la raccolta di tutto l'insieme della documentazione legale necessaria per svolgere il test.
Le fasi di information gathering, vulnerability analysis e vunerability exploits combinate costituiscono il penetration test vero e proprio, la parte centrale del processo. Il test parte con la fase di information gathering. Fase resa molto diversa a seconda di che strategia di penetration testing abbiamo deciso di eseguire ovviamente sarà una fase molto complessa in caso di penetration testing black box mentre sarà quasi saltata nell' approccio white box.
La fase di information gathering richiede di analizzare sia la struttura logica che fisica della superficie d'attacco ossia il dispositivo o l'insieme di dispositivi che saranno soggetti al penetration test.
L'information gathering è una fase fondamentale in quanto il successo del test dipende dalla scoperta di più informazioni e quindi vulnerabilità possibili della superficie di attacco per garantire più opzioni possibili nelle fasi successive.
Una volta capito che sistema si ha davanti si può fare un'analisi delle sue vulnerabilità e provare a sfruttare queste attraverso l'uso di tool automatizzati e specifici per ogni tipologia di vulnerabilità nelle fasi di Vulnerability analysis e Vulnerability Explots. Le due fasi sono tanto più lunghe e tanto più di successo in base a quanto si è scoperto sul dispositivo nella fase precedente.
Si riflette poi sui risultati del test nella fase di test analysis con la compilazione di un report che deve fornire informazioni tecniche ma allo stesso tempo essere fruibile dalla maggior gamma di persone possibili e non solo da chi ha un back ground tecnico.
Nel report otre che esporre i problemi chi esegue il test può anche aggiungere possibili soluzioni per aiutare chi legge il report ad attuare una risposta immediata a vulnerabilità ritenute particolarmente d'interesse.

2 Dispositivi IoT

L’Internet of things (IoT) [3] è un paradigma di comunicazione che punta a collegare un numero progressivo di oggetti della nostra vita di tutti i giorni ad internet così da poter aver accesso a dati generati da questi oggetti direttamente a contatto con il mondo reale questi solitamente sono: sensori, applicativi controllati a distanza, veicoli, controllori ambientali ecc…
I dispositivi smart nella nostra casa diventano sempre più comuni e molti di questi ai fini di garantire un servizio più personalizzato chiedono spesso all’ utente di inserire informazioni personali sensibili. Bisognerebbe che la progettazione di questi dispositivi IoT prendesse in seria considerazione la privacy dell’utente ma ciò non avviene in tutti i casi.
Per prima cosa molti di questi dispositivi per limitazioni hardware quali la dimensione o la capacità della batteria non possono applicare molti dei sistemi di protezione più standard che sono invece stati pensati per dispositivi con capacità hardware superiori. Un'altra problematica inoltre è la percezione che si ha del dispositivo e dei rischi informatici che questo porta. Se sia a livello aziendale che privato negli anni si è sviluppata una cultura sulla cybersecurity per quanto riguarda computer e smartphone questa attenzione alla sicurezza non si è ancora sviluppata per i dispositivi IoT. Questi ultimi sono lontani da essere inattaccabili le aziende produttrici tal volta sono negligenti sui protocolli usati per criptare le connessioni tra dispositivi IoT ma allo stesso modo è estremamente comune che gli utenti non cambino la password di fabbrica per accedere al dispositivo IoT da remoto.
Come discusso gli attacchi su dispositivi Iot sono spesso semplici e facili da condurre.
Ad esempio potrebbero essere eseguiti per creare disservizi ma anche per violare la privacy dell’utente e diffondere informazioni sensibili. I dati contenuti in un dispositivo Iot infatti possono andare da semplici dati riguardo la temperatura di una stanza fino a informazioni molto più sensibili sulle abitudini dell’utente. Un’altra tipologia di attacco non mira a estrarre informazioni dal dispositivo ma ben sì attaccare il dispositivo più vulnerabile della rete domestica o aziendale per poi sfruttare le connessioni della rete stessa per accedere a un dispositivo con dati più sensibili utilizzando i permessi garantiti a un dispositivo appartenente al network domestico o aziendale.
Per essere valutato sul piano della sicurezza ogni dispositivo Iot ha dei requisiti da rispettare.

2.1 Security Requirements

I requisiti sono presentati a diversi livelli operazionali:

- Informazione
- Accesso
- Funzionalità

2.1.1 Livello di informazione

A questo livello la sicurezza dovrebbe garantire questi requisiti

- Integrità
  I dati a ricevitore non devono essere alterati durante la trasmissione
- Anonimato
  L'identità di una fonte di dati dovrebbe rimanere nascosta a terzi
- Confidenzialità
  I dati non possono essere acquistati da terzi
- Privacy
  Le informazioni personali del client non dovrebbero essere rivelate durante uno scambio di dati.

2.1.2 Livello di accesso

Specifica alcuni meccanismi per controllare l'accesso al network del quale il dispositivo IoT fa parte

- Controllo accessi
  Garantisce che solo utenti con i giusti permessi di accesso possano accedere al network per task amministrativi come controllo del dispositivo o del network da remoto
- Autentificazione
  Verifica che un dispositivo abbia il diritto di accedere ad un network ma anche che il network abbia il permesso di connettersi al dispositivo
- Autorizzazione
  Garantisce che solo i dispositivi autorizzati possano accedere alle informazioni e ai servizi che il network offre

2.1.3 Livello funzionale

A questo livello si definiscono i requisiti seguenti

- Resilienza
  Si riferisce alla capacità di un network di garantire la sicurezza ai dispositivi connessi anche in caso di attacchi o problemi con alcune componenti a causa di malfunzionamenti
- Auto organizzazione
  Denota la capacità di un sistema IoT di rimanere operativo anche in caso di fallimenti di alcune parti per malfunzionamenti periodici o attacchi

3 Penetration Test in un dispositivo IoT

I vari dispositivi IoT anche se molti diversi l'uno dall'altro presentano spesso proprietà, funzionalità e modi di utilizzo similari[4]. Questi aspetti comuni si traducono in vulnerabilità comuni tra i vari dispositivi. Ciò non significa che il dispositivo in quanto IoT avrà solo delle vulnerabilità specifiche ma che essendo le più comuni conviene partire da queste. Iniziando il penetration test su queste vulnerabilità infatti ci garantirà maggiore probabilità di trovare falle nella sicurezza del sistema.

Un penetration test su dispositivo Iot è più complesso di un semplice penetration test per tutte le diverse componenti e framework usati nel dispositivo stesso

L'architettura di un dispositivo Iot può essere infatti divisa in 3 macro categorie

- Dispositivo Embendded
- Firmware software e applicazioni
- Rete Comunicazioni radio

3.1 Dispositivo Embedded

Il dispositivo stesso sul quale viene eseguito il penetration test è il "thing" della sigla IoT.
Le possibili vulnerabilità riscontrabili a livello di dispositivo sono:

- Porte seriali esposte
- Meccanismo di autenticazione di porta scadente
- Possibilità di scaricare il firmware con Jtag o flash drive
- Attacchi external media-based
- Analisi alimentazione e attacchi side-channel

Tramite la comprensione di quali siano le funzionalità e i dati ai quali il dispositivo ha accesso possiamo valutare quale di queste vulnerabilità può risultare più dannosa

3.2 Firmware, software e applicazioni

Firmware è la parte logica del dispositivo composta dal codice "software" e le applicazioni che vengono eseguite dal o sul dispositivo per garantire le sue funzionalità

Spesso tramite reverse engineering sull'applicazione o sul firmware si possono scoprire vulnerabilità a livello di dispositivo o di rete che prima non era possibile sfruttare. Vulnerabilità che emergono

dalla scoperta di come due dispositivi comunicano tra loro e quali protocolli stiano usando per la comunicazione stessa. Le componenti software di un dispositivo smart home includono solitamente:

- Applicazione mobile
- Dash board Web-based
- Interfacce network
- Firmware

Applicazione che ci permette di controllare il dispositivo da un dispositivo mobile, solitamente scritta per Android o IOS. Si posso svolgere svariati tipi di attacco su questa sfruttando possibili mancanze dell'applicazione molti progettisti infatti si fidano della forza del linguaggio e del codice di terzi copiando librerie. Un'app progettata in questo modo potrebbe lasciare vulnerabilità soprattutto se non scaricata da un app store ma da un sito di terze parti. In questo caso infatti nessuna delle norme standard di sicurezza potrebbe essere implementata nell'applicazione in quanto quest'ultima non dovrebbe adattarsi a nessuna delle linee guida dello store.

Un accesso al sorgente dell'app può essere un grave problema in quanto può svelare chiavi di criptazione e tecnologie di comunicazione. Informazioni che se non protette potrebbero esporre falle di sicurezza a livello di dispositivo e rete.

Riassumendo possibili vulnerabilità dell' applicazione sono:

- Reverse engineering
- Scaricare codice sorgente dell'app
- Verifiche di autenticazione e autorizzazione insufficienti
- accesso tramite glitch e bug
- Leak di dati da interfacce di terze parti
- Comunicazione network poco sicura

Dash board Web-based

Se il dispositivo IoT è da operare con una web app non ci sono restrizioni né store quindi il sito potrebbe essere operativo ma allo stesso tempo completamente insicuro permettendo a chiunque di accedere a dettagli sul dispositivo

Riassumendo le possibili vulnerabilità della dash board sono:

- Script e richieste cross-site
- Verifiche di autenticazione e autorizzazione insufficienti
- Ingnezione di codice client side

3.3 Rete e comunicazioni radio

Comunicazioni Radio
Solitamente non considerate a livello di sicurezza quindi facilmente attaccabili
Categorie di comunicazione radio utili a livello di penetration test includono:

- SDR (Software Define Radio)
- Attacchi al protocollo ZigBee
- Exploit BLE

A seconda del dispositivo posso avere vulnerabilità molto diverse tra loro a seconda del protocollo di comunicazione usato infatti il dispositivo potrebbe essere molto sicuro rispetto a una tipologia di attacco ma molto fragile a un'altra

- Attacchi man in the middle (MITM)
- Attacchi replay
- Denial of Service (DoS)
- Mancanza di criptazione
- Possibilità di estrarre informazioni sensibili dalla comunicazione
- Intercettamento e modifica del segnale

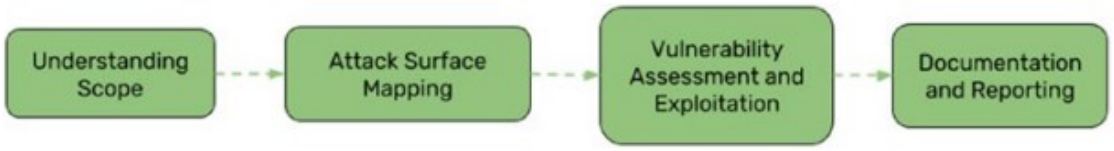
Network interfaces

Anche se applicazione o la web app sono state progettate ad hoc per prevenire ogni forma di problema di sicurezza queste potrebbero utilizzare interfacce di terzi scadenti o che non vengono aggiornate con patch di sicurezza tanto di frequente come l'applicazione. La cosa potrebbe condurre brecce nella sicurezza del dispositivo.

4 Test

Come preannunciato in questa fase si andrà ad eseguire il penetration test di un dispositivo IoT e della rete domestica nel quale è inserito.

Seguendo un paradigma di penetration testing specifico per i dispositivi embedded e IoT lo svolgimento del test sarà diviso in quattro fasi:



- **Understanding scope**

  Nella fase di understanding scope si dovrà definire cosa si vuole ottenere dal test e darsi dei limiti su come questo verrà svolto con attenzione particolare a: tempistiche, insieme di dispositivi target, livello di invasività del test, tipologia di test (white, grey e black box).

  A partire da queste definizioni si può farsi un'idea generica di che tipologia di attacchi si andranno a svolgere così da concentrarsi solo su una o più specifiche macroaree durante le fasi successive.

  Simile alla fase di test preparation nel paradigma generico di penetration test ma con un'attenzione specifica alle fragilità del dispositivo che viene attaccato. Come spiegato ci si basa spesso su vulnerabilità generiche per ottimizzare il test.

- **Attack Surface Mapping**

  Durante questa fase si cercheranno di ottenere maggiori informazioni possibili sul dispositivo IoT e sulla rete di cui fa parte.

  Tramite questa ricerca si cercheranno di carpire informazioni su:

  1. Componenti del dispositivo
  2. Architettura della CPU
  3. Protocolli di comunicazione usati
  4. Dettagli su applicazioni mobili collegate
  5. Processo di aggiornamento del firmware

  Fase speculare dell'information gathering nel paradigma generico di penetration test

- **Vulnerability Assessment and Exploitation**
  Unione delle due fasi di Vulnerability Analysis Vulnerability Exploits del paradigma generico di penetration test. Da notare come nei dispositivi IoT spesso si vada a fare un'analisi anticipata delle vulnerabilità del sistema in quanto queste, come già spiegato, sono spesso comuni a vari dispositivi IoT. L'attenzione non è tanto sulla scoperta della vulnerabilità ma su un test della stessa. Il test andrà ad attaccare il dispositivo e la rete di cui questo fa parte in modi che nella maggior parte dei casi si rivelano efficaci a causa di vulnerabilità comuni.
  La fase è infatti di vulnerability assesment la vulnerabilità non è scoperta ma piuttosto provata. Solo in seguito a una prova che garantisca che la vulnerabilità comune è presente anche nel dispositivo sotto analisi si continuerà con l'attacco analizzando i danni che questo potrebbe provocare alla rete.

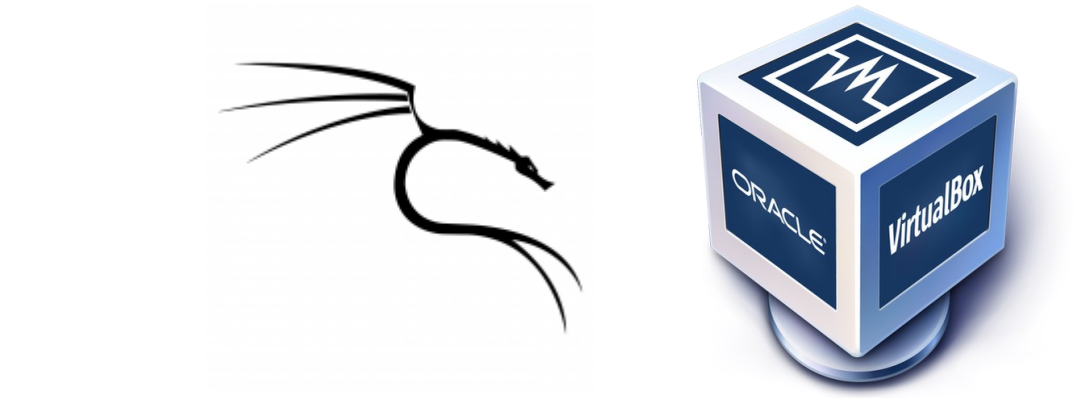
- **Documentation and reporting**
  Parallelo della fase di Test Analysis nel modello astratto di penetration testing

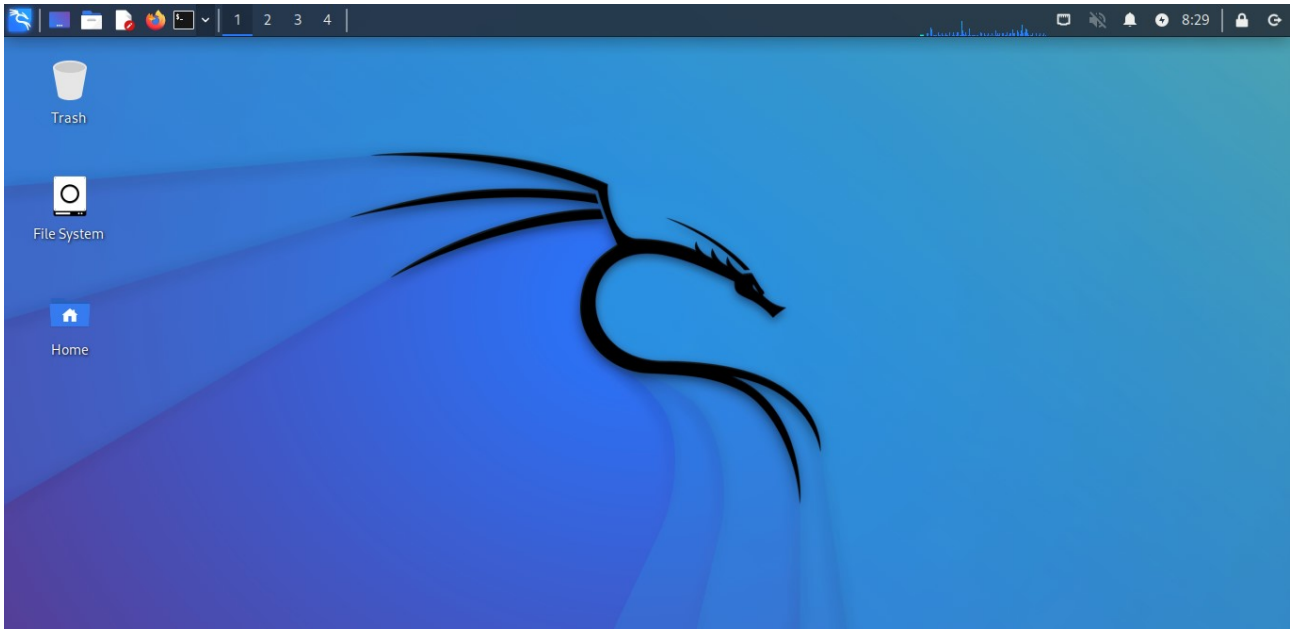
4.1 Understanding Scope

Tra tutte le aree che esponevano vulnerabilità si è deciso di concentrarsi sulla parte di reti e comunicazioni radio. Attacchi troppo legati all' hardware del dispositivo IoT avrebbero probabilmente provocato danni alle sue componenti. Sulla superficie del dispositivo non sono infatti presenti modi per accedere agli elementi circuitali senza rovinare ciò che li incapsula.
Accedere al firmware o alle applicazioni del dispositivo è sicuramente il metodo migliore. Un accesso da questa macro area garantisce un controllo remoto completo ma è allo stesso tempo il più complesso da eseguire. Il fatto che entrambe le applicazioni con le quali connettersi al dispositivo siano scaricabili dall' app store ci confermano che entrambe implementino le più recenti linee guida sulla sicurezza. Inoltre il codice sorgente di entrambe non è disponibile. La probabilità di riscontrare vulnerabilità in questa macroarea sono perciò minime.
Per ottenere dei risultati nel minor tempo possibile si dovrà attaccare le vulnerabilità di un'altra area, un'area nella quale si hanno maggiori chance di trovare faglie nella sicurezza.
Si è deciso di concentrarsi quindi sulla rete e le comunicazioni radio cercando di quanto meno dare una buona idea di come il dispositivo IoT potrebbe essere un problema di sicurezza non solo per sè stesso ma anche per l'intera rete alla quale è connesso. La rete è composta da dispositivi eterogeni che potrebbero presentare problemi di sicurezza quando comunicano l'uno con l'altro. Le metodologie di attacco via rete e comunicazioni radio sono le più antiche e diffuse. Non è difficile immaginare che nella maggior parte dei casi se si dovesse subire un attacco alla rete domestica sarebbe fatto sfruttando vulnerabilità di rete e comunicazioni radio. Testare soprattutto quest'area può darci sicuramente un'idea di quanto la rete sia solida rispetto agli attacchi più comuni.

4.1.1 Tool Utilizzati

Per effettuare e controllare lo stato di tutti gli attacchi svolti si dovranno usare dei tool.
Dopo aver cercato a lungo nel web non si è riusciti a trovare tutti i tool necessari per eseguire gli attacchi su windows quindi si è deciso di installare Kali Linux su macchina virtuale. Kali Linux è una distribuzione Linux open source basata su Debian orientata a varie attività di sicurezza delle informazioni, come Penetration Testing, Security Research, Computer Forensics e Reverse Engineering [5]. Kali Linux è il sistema operativo più usato da chi svolge penetration testing a livello professionale. Risulta perfetto per eseguire tutte le tipologie di attacco alla rete e alle comunicazioni radio in quanto esistono comandi specifici per ogni attacco preinstallati in Kali Linux. Non volendo partizionare il disco si è deciso di usare una macchina virtuale in questo caso Virtual Box.Le macchine virtuali simulano le caratteristiche di un computer reale per poter eseguire sistemi operativi senza dover cambiare il sistema operativo del proprio terminale.





Anteprima terminale

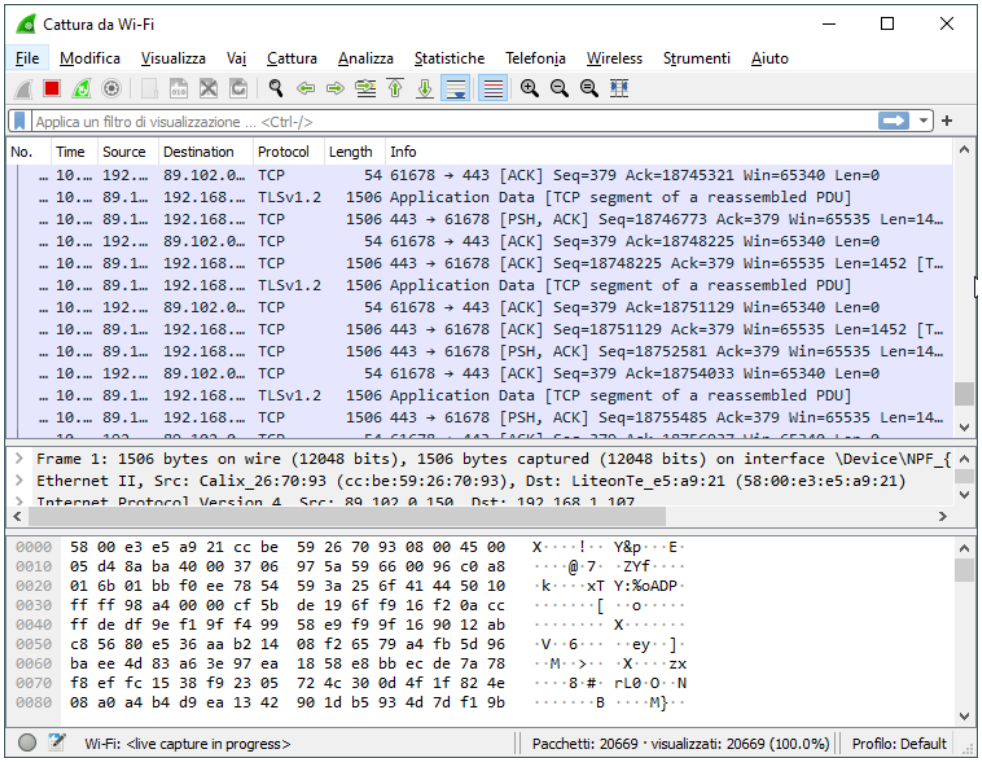
Per controllare come l’attacco procede ma avere maggiori dettagli sui messaggi che i vari dispositivi scambiano all’ interno della rete domestica si è scelto di usare l’applicazione per Windows WireShark.

Wireshark è l'analizzatore di protocolli di rete più diffuso e utilizzato al mondo. Ti consente di vedere cosa sta succedendo sulla tua rete a livello microscopico ed è lo standard de facto (e spesso de jure) in molte imprese commerciali e senza scopo di lucro, agenzie governative e istituzioni educative. Lo sviluppo di Wireshark prospera grazie ai contributi volontari di esperti di networking in tutto il mondo ed è la continuazione di un progetto avviato da Gerald Combs nel 1998. [6]
Con l’uso di questa applicazione siamo in grado di analizzare tutte le reti che si trovano nelle vicinanze del nostro terminale.
Grazie all’ utilizzo dell’applicazione si riusciranno ad avere informazioni sulla rete ma soprattutto sui pacchetti che viaggiano all’ interno di questa. Nello specifico le informazioni che si potranno trarre da ogni pacchetto trasmesso nella rete saranno:

- Orario d’invio
- IP trasmettitore
- IP ricevitore
- Lunghezza del pacchetto
- Protocollo della comunicazione
- Info sul pacchetto

Tutte queste informazioni sono essenziali tanto in una fase di prima analisi per capire con quali protocolli di connessione i vari dispositivi appartenenti alla rete comunichino tra loro tanto in una fase successiva di penetration test. Nel penetration testing infatti come si è già spiegato è di vitale

importanza fare un'analisi precisa dei risultati. Un attacco svolto sulla rete potrà avere successo o fallire ma in entrambi i casi lo scopo dell'attacco non è principalmente di vedere la risposta alla minaccia ma di capire come il sistema risponda alla minaccia cosa che si può vedere solo in parte con solo dati empirici come il rallentamento della connessione.



Schermata utilizzo WireShark

4.1.2 Metriche usate

Si è scelto di valutare e così essere in grado di analizzare al meglio le vulnerabilità trovate durante la fase successiva di Vulnerability Assessment and Exploitation usando i modelli:

- Modello STRIDE [7]
- Modello DREAD [8]

Modello Stride

Per etichettare il tipo di minaccia associata alla vulnerabilità del dispositivo IoT e/o della rete nella quale questo è inserito usiamo questo modello che divide i vari attacchi nelle categorie

- Spoofing
- Tampering
- Repudiation
- Information disclosure
- Denial of service
- Elevation of privilege

Modello Dread

Assegna per ogni vulnerabilità un livello di pericolo per il dispositivo e la rete sulla quale quest' ultimo è connesso nel caso questa vulnerabilità fosse sfruttata in un attacco.
Il livello di pericolosità di un attacco viene misurato dividendo il rischio di attacco in varie categorie e assegnando a queste un valore da 1 a 3 indicando rispettivamente un rischio: basso, medio e alto.
Le categorie sono:

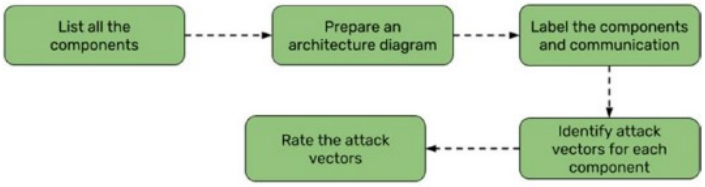
- Damage
  quanto grave sarebbe l'attacco
- Reproducibility
  quanto è facile riprodurre l'attacco
- Exploitability
  quanto è facile sfruttare l'attacco
- Affected users
  quanti utenti vengono colpiti dall'attacco
- Discoverability
  quanto facilmente si può scoprire la minaccia

Si sommano poi i risultati ottenuto nelle varie categorie per valutare il livello di rischio complessivo dell'attacco

- Basso
  Totale tra 5-7
- Medio
  Totale tra 8-11
- Alto
  Totale tra 12-15

4.2 Attack Surface Mapping

I vari passi per effettuare un efficace attack Surface Mapping sono



4.2.1 Elencare le componenti

Lista dei componenti della rete

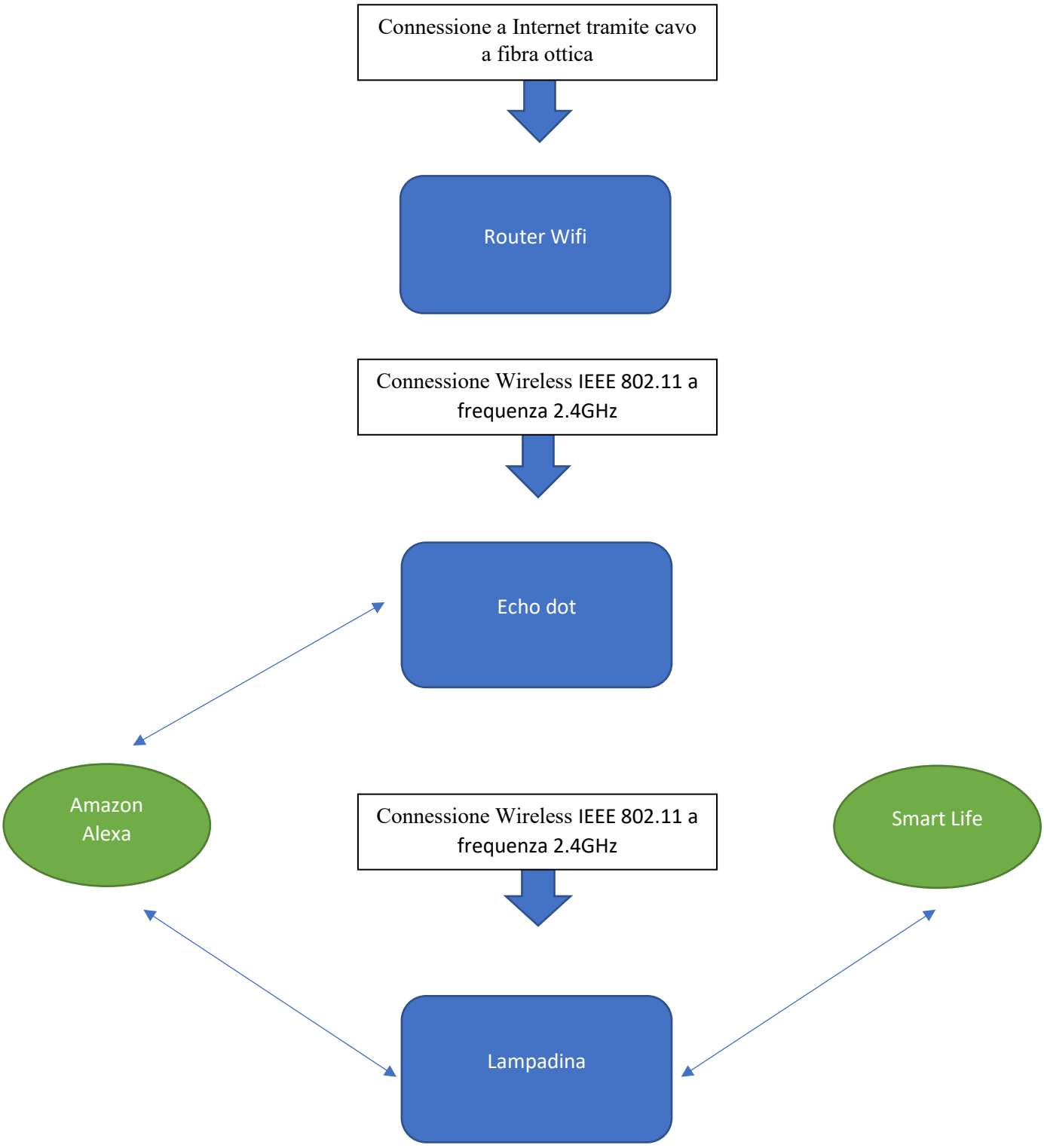
- **Router Wifi**
  Garantisce un collegamento ad Internet a tutte le altre componenti della rete

- **Echo dot**
  Amazon Echo dot (3ª generazione) – Altoparlante intelligente con integrazione Alexa
  L'unico dispositivo direttamente connesso ad internet attraverso il router.
  Può eseguire comandi anche molto complessi a partire da input che posso arrivare a livello fisico con comandi vocali o con azioni sui 4 tasti presenti sulla scocca del dispositivo. Ma anche da remoto tramite l'app Amazon Alexa.

- **Lampadina**
  Bakibo Lampadina Wifi Intelligente Led Smart Dimmerabile 9W 1000Lm
  Non direttamente connessa ad internet ma connessa al Echo dot che ne controlla le funzionalità.
  Per eseguire i comandi si può agire a livello fisico passando il comando al dispositivo Echo dot oppure da remoto usando sia l'app Alexa sia un'app proprietaria della casa produttrice della lampadina stessa chiamata Smart Life.

4.2.2 Preparare un diagramma dell'architettura e etichettare componenti e comunicazioni

Nella rappresentazione ogni componente è fondamentale: i dispositivi sono rappresentati in blu, le connessioni in bianco mentre le app sono in verde



4.2.3 Identificare i vettori d'attacco per ogni componente e valutare ciascuno

Si tengono in considerazione le vulnerabilità e quindi i vettori di attacco in funzione della lampadina ossia il dispositivo IoT vero e proprio.

- Sottrarre informazioni sull'utente o sulla connessione andando a carpire informazioni ascoltando lo scambio di informazioni tra i dispositivi se questo scambio non fosse criptato.
  In caso le connessioni non fossero protette questo vettore di attacco potrebbe essere estremamente dannoso.
- Il funzionamento del dispositivo IoT non è indipendente dal funzionamento degli altri dispositivi presenti nella rete. In particolare se solo uno degli altri due dispositivi appartenenti alla dovesse subire delle manomissioni il funzionamento della lampadina non sarebbe garantito. Non solo tutti i dispositivi devono essere operativi anche le connessioni tra questo devono esserlo. La manomissione di solo una tra le connessioni crea un potenziale disservizio della lampadina.
  Il mancato funzionamento di uno dei dispositivi è facilmente individuabile. Un attacco efficace in quest'area deve essere meno improntato ad agire di nascosto ma ad essere difficile da fermare una volta che è in corso.
- Da un'analisi approfondita dell'hardware del dispositivo IoT si nota che questo non ha una memoria non volatile. La memoria di un dispositivo si dice volatile quando quest'ultimo non è in grado di mantenere le informazioni in assenza di corrente.
  Questa vulnerabilità potrebbe essere sfruttata per prendere il completo controllo del dispositivo IoT.
- Provare a impersonare l'utente replicando le comunicazioni di relazione tra i dispositivi o le comunicazioni di comando mandate tramite applicazione.
  Se ciò fosse possibile si potrebbero far eseguire comandi al dispositivo IoT.
- Tutte le comunicazioni avvengono solo dopo che i vari dispositivi sono stati precedentemente associati.
  Cancellando o modificando l'informazione dell'associazione da un dispositivo gli si impedisce di effettuare comunicazioni future.

4.3 Vulnerability Assessment and Exploitation

Ad ogni vulnerabilità trovata attraverso il punto precedente associamo una tipologia di attacco utilizzando il modello Stride e una valutazione della minaccia che comporta alla rete con il modello Dread. In seguito si spiegherà al meglio la tipologia di attacco che si andrà a svolgere per capire non solo come questo avvenga ma anche per dare un'idea precisa di come questo funzioni. Sarà così possibile trarre maggiori informazioni dall'attacco specifico eseguito e contestualizzarle.

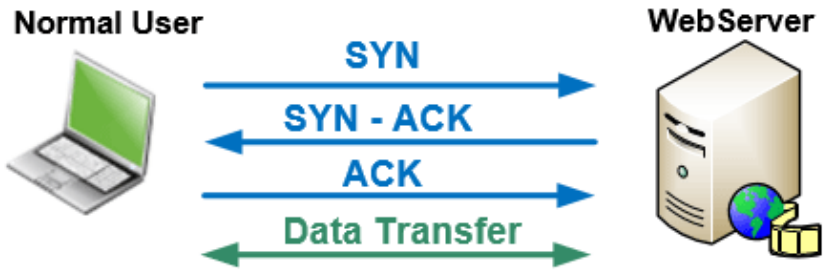
4.3.1 Denial of Service [9]

L'attacco va a sfruttare la seconda vulnerabilità della rete ossia che il funzionamento della lampadina sia subordinato al corretto funzionamento di tutti i dispositivi e di tutte le connessioni della rete.
Dal modello stride con questo attacco si cerca di rendere indisponibile un processo/servizio. Con questo penetration test cercheremo di impedire il corretto funzionamento della lampadina andando a creare un malfunzionamento in un'altra parte della rete.
La valutazione della minaccia nel modello Dread è:

	Punteggio
Damage	2
Reproducibility	3
Explitability	1
Affected users	2
Discoverability	1
Rischio tot	**9 (medio)**

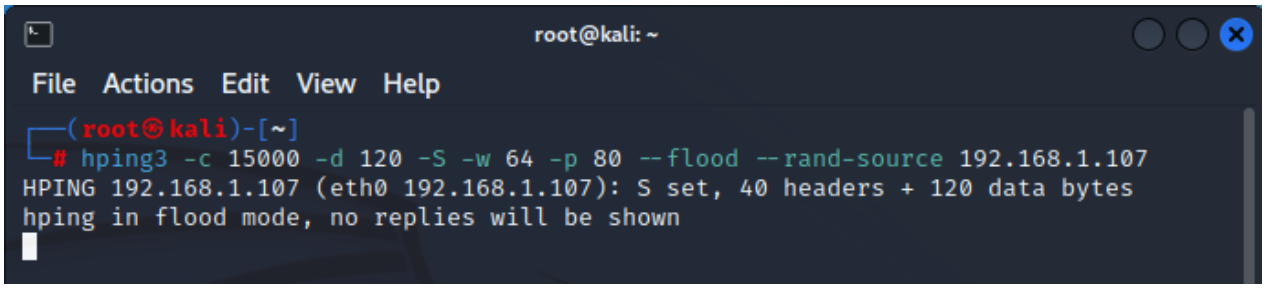
Gli attacchi Denial of Service (DoS) sono diventati uno dei maggiori rischi per le moderni reti di computer. Sono attacchi che esistono e sono conosciuti dalla nascita delle connessioni ma per lungo tempo non sono mai stati utilizzati da malintenzionati e sono diventati un vero problema solo più avanti. Gli attacchi DoS disponibili su Internet generalmente prendono il controllo di un dispositivo target saturando le sue risorse. Risorse che possono essere qualsiasi cosa legato alla rete di dispositivi che si sta cercando di prendere di mira. Esempi di risorse da saturare sono: larghezza di banda, buffer connessioni TCP, buffer di applicazioni e servizi, cicli CPU, etc. Ma si possono anche sfruttare vulnerabilità specifiche della rete ma sempre allo scopo di rendere indisponibile un processo/servizio che normalmente la rete provvederebbe.
Negli anni sta diventando progressivamente più complicato saturare le risorse di una rete con l'uso di un solo terminale soprattutto se questa rete provvede servizi per un'intera azienda. Per compiere l'attacco nonostante le difficoltà i moderni attacchi DoS sono lanciati da un grosso numero di terminali host distribuiti non solo sulla rete ma spesso anche geograficamente. Gli attacchi DoS così

fatti sono detti Distributed Denial of Service (DDoS) denominazione che va ad esplicitare la natura distribuita dei dispositivi che svolgono l'attacco.
Comparati con i più convenzionali attacchi DoS che possono essere facilmente gestiti con migliore security ad esempio prevenendo l'accesso alla rete a dispositivi interni o esterni che stanno saturando le risorse della rete stessa gli attacchi DDoS sono più difficili da contrastare.
Siccome l'attacco DDoS è eseguito da molti dispositivi diversi e sparsi è difficile se non impossibile distinguerli e quindi prevenirne l'accesso alla rete.
Nel penetration test ci si è limitati ad attacchi da un unico dispositivo vista le scarse risorse da saturare che offre una rete domestica ma è anche da considerare il fatto che un attacco su più larga scala e con meno opzioni difensive potrebbe sicuramente avere effetti maggiori.
Esistono vare metodologie per effettuare un attacco DoS sicuramente il più conosciuto e usato è il TCP SYN Floading.
Questo è un tipo di attacco che sfrutta i protocolli di stato delle connessioni perché questi protocolli consumano risorse per mantenere uno stato. Il TCP SYN Floading si basa su una delle connessioni più comuni quella tra client e server. Quando un client infatti tenta di stabilire una connessione TCP con un server manda un messaggio di sincronizzazione (SYN) per informare il server che sta tentando di stabilire una connessione. Ricevuto questo segnale il server se disponibile manda una risposta SYN-ACK per comunicare al client che è disponibile alla connessione. Il client a questo punto sottoscrive la creazione della connessione con l'invio di un messaggio di conferma (ACK). La vulnerabilità sta nel fatto che il server deve aspettare la conferma (ACK) del client prima di poter effettuare la connessione ma nel mentre dovrà mantenere salvata l'informazione che il client precedentemente gli ha inviato un messaggio di sincronizzazione (SYN). Se il server dovesse ricevere troppe richieste di sincronizzazione tutte insieme non potendone scartare non potrebbe più accettare nuove connessioni creando possibili disservizi. Ovviamente sistemi moderni implementano time-out che scartano connessioni pendenti da troppo tempo ma allo stesso tempo chi esegue gli attacchi ha trovato modi più efficienti per inondare di pacchetti la vittima in modo che la frequenza dei parchetti ricevuti sia maggiore della frequenza dei time out che li vanno a scartare. Comunque il TCP SYN anche se il metodo in uso da maggior tempo non è l'unico con il moltiplicarsi delle connessioni negli anni si sono sviluppati conseguentemente attacchi che sfruttano le vulnerabilità uniche di ciascuna.
La vulnerabilità del protocollo TCP è ovviamente, come prima spiegato, la three way handshake ossia la conferma di avvenuta connessione con tre messaggi separati.



Three way handshake TCP prima del trasferimento dati

Il modo più semplice per performare un attacco di questo tipo è usando Kali Linux più nello specifico il tool *hping3*. Quest'ultimo è tool specifico per performare penetration test al protocollo TCP già incluso in Kali Linux. Per accedere al comando e quindi al tool hping3 si dovrà comunque accedere a Kali Linux come root in quanto il comando usa dei permessi non garantiti a tutti gli user.
Chi attacca usa hping o un qualsiasi altro tool per generare indirizzi IP casuali per mandare poi richieste TCP SYN. Il dispositivo attaccato risulterà così inondato da richieste da un numero altissimo di IP diversi creando così dei disservizi o quanto meno dei rallentamenti.
Il comando da eseguire è



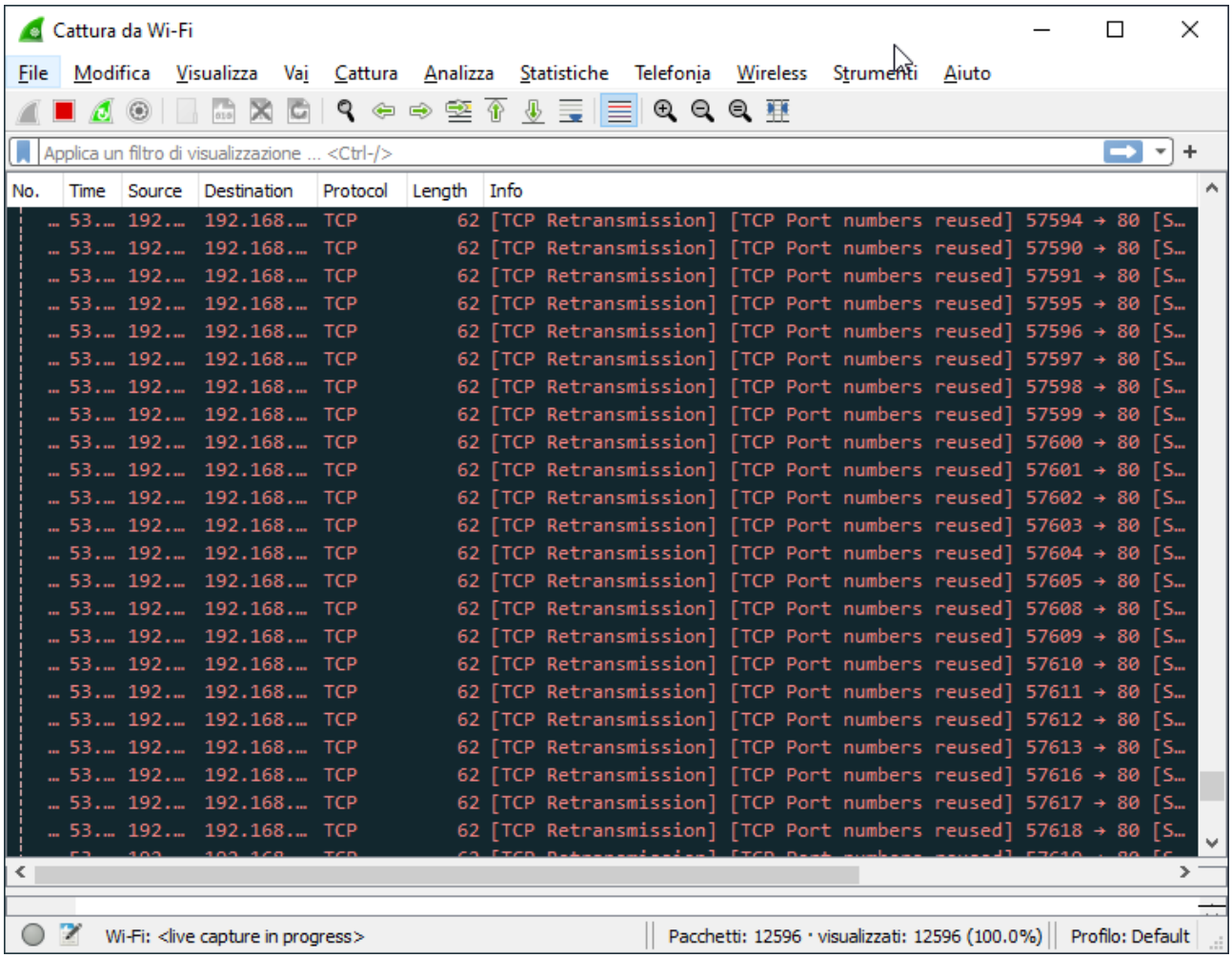
I vari parametri passati al comando stanno per

- -c 15000 → 15000 pacchetti inviati
- -d 120 → la grandezza del pacchetto è 120 byte
- -S → invio di pacchetti SYN TCP
- -w 64 → grandezza finestra di risposta TCP
- -p 80 → l'attacco avverrà sul port 80
- fload → pacchetti inviati con maggior frequenza
- rand source → IP sorgente della richiesta randomizzato
- 192.168.1.1 → IP target

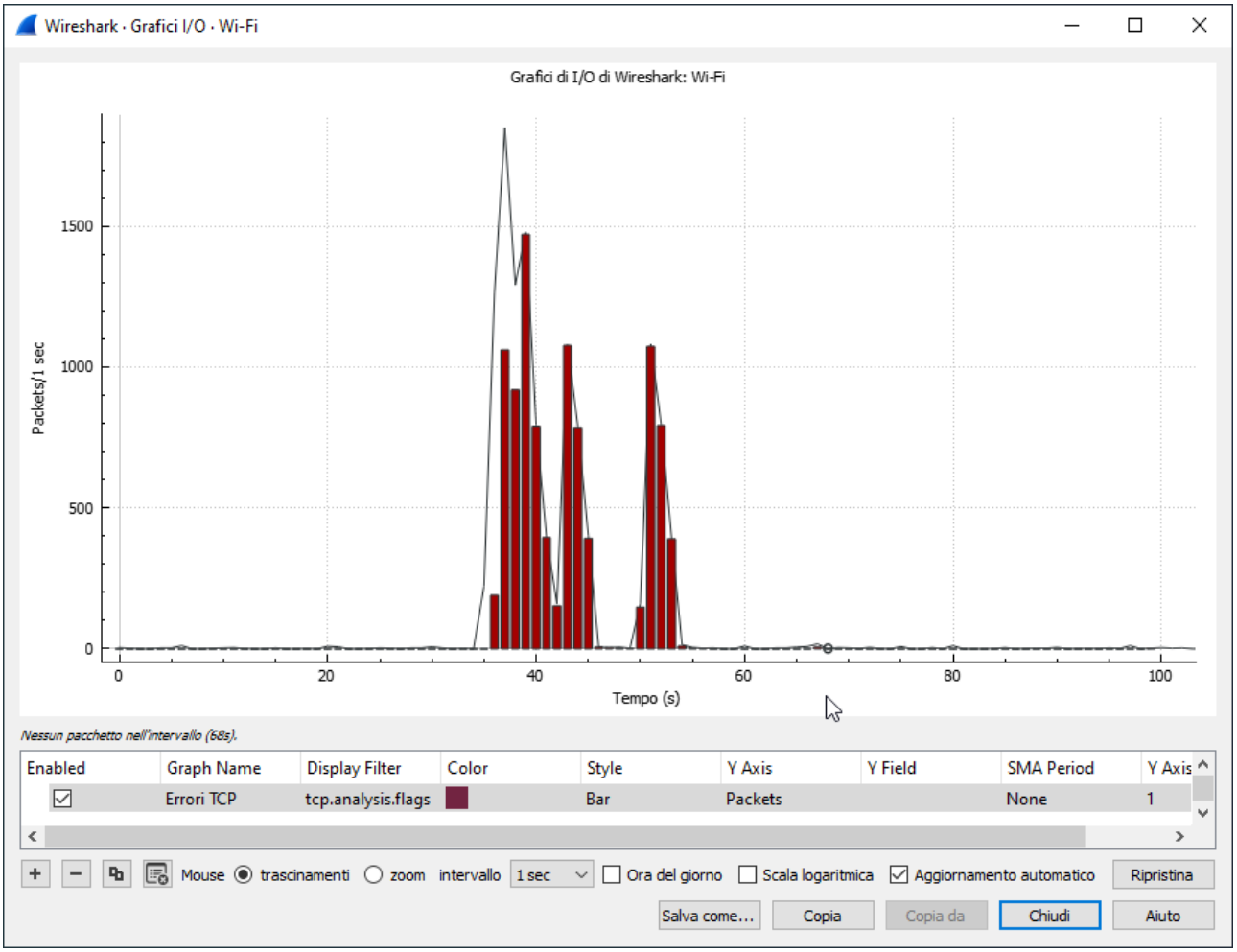
Da notare come riuscendo ad inviare le richieste da indirizzi IP casuali non c'è modo per chi subisce l'attacco di tracciare la fonte di questo e quindi avere chance di fermarlo. L'attacco è da vedersi come un intermedio tra DOS e DDOS in quanto gli IP di chi attacca sonno diversi ad ogni pacchetto ma allo stesso modo tutti i pacchetti vengono dalla stessa area cosa che potrebbe far individuare chi sta svolgendo l'attacco.

L’attacco è da vedersi come un successo come vediamo da WireShark si è chiaramente creato un picco molto alto di traffico di pacchetti TCP SYN durante l’attacco. La rete Wifi non ha avuto disservizi completi ma ha subito dei sostanziali rallentamenti. L’attacco è stato svolto da una sola macchina ma ha avuto comunque un risultato ben visibile. Da quello che sappiamo sulla tipologia d’attacco è facile pensare che

Da un’analisi dei pacchetti vediamo come per tutta la durata del test quasi la totalità dei pacchetti in arrivo sia composta da richieste TCP. Andando poi a leggere i dettagli di ogni richiesta si può evidenziare che come richiesto tutti gli indirizzi IP che hanno mandato i pacchetti contenenti richieste TCP sono diversi l’uno dall’altro.



Usando lo strumento grafico presente in WireShark possiamo apprezzare al meglio il picco anomalo di pacchetti in transito nella rete Wi-Fi. Il traffico dati nonostante nettamente aumentato non ha creato completo disservizio. Bisogna anche notare come la rete Wi-Fi al momento dell'attacco fosse soggetta a un traffico molto basso che nel grafico appare quasi nullo comparato al traffico durante l'attacco. Ovviamente un attacco più prolungato e su una rete con un traffico più elevato avrebbe potuto creare danni maggiori. Ma il test è un mezzo per capire come la rete reagisca all'attacco e non un modo per danneggiare la rete. Così si è deciso di interrompere l'attacco una volta che questo ha creato rallentamenti apprezzabili a livello di velocità di esecuzione di comandi sul dispositivo IoT senza spingersi oltre.

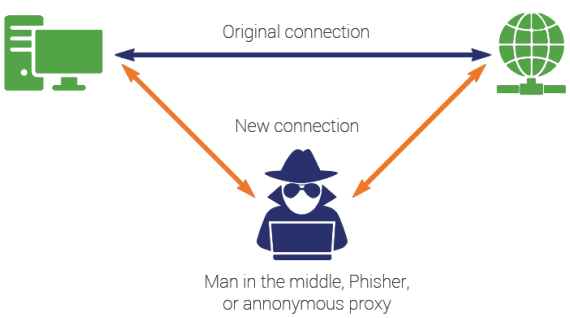


4.3.2 Man in the Middle [10]

L'attacco Man in the middle (MITM) è uno degli attacchi più conosciuti e diffusi nella sicurezza informatica, prende di mira la connessione tra due dispositivi e mette a rischio la sicurezza delle informazioni trasportate e la coerenza dei dati trasmessi.
Nel modello Stride è generalmente è definito spoofing. Ma alcuni tipo di attacchi MITM potrebbero essere classificati come Information Disclosure, Denial of Service ma anche Tampering il tutto a seconda di che connessione viene attaccata e di cosa si decide di fare con i pacchetti ottenuti.
La valutazione della minaccia nel modello Dread è:

	Punteggio
Damage	2
Reproducibility	1
Explitability	3
Affected users	2
Discoverability	3
Rischio tot	**11 (medio)**

Nell'attacco man in the middle chi esegue l'attacco si intromette nella comunicazione tra due o più dispositivi appartenenti alla rete senza che questi ne siano consapevoli. Chi attacca potrebbe ricevere passivamente informazioni scambiate tra i due partecipanti alla comunicazione o diventare un elemento attivo e interferire con lo scambio di dati modificando o cancellando le informazioni che i due dispositivi cercano di scambiare. L'attacco è in circolazione fino dagli anni 80' quindi si è fatta ricerca in maniera estensiva su di esso per far sì che i sistemi moderni riescano al meglio a contrastarlo con nuove tecniche di criptazione e mascheramento.
Esistono due metodologie principali nelle quali l'attacco è eseguito: creare reti false controllate da chi attacca o modificare illegittimamente le informazioni che si scambiano tra loro i due dispositivi.
Mentre il primo è un attacco che è diventato tristemente comune e molto utilizzato nelle reti Wi-Fi pubbliche spesso di bar e ristoranti il secondo è sicuramente un attacco che richiede molte più conoscenze tecniche per essere eseguito e diventa man mano più complicato più il livello di criptazione della comunicazione è elevato.



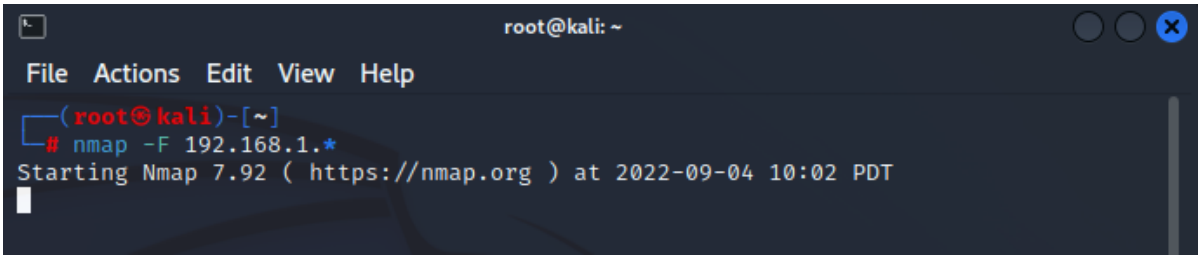
Lo scopo in entrambi i casi per chi attacca è quello di porsi nel mezzo della comunicazione e far credere a entrambe le parti di star comunicando tra loro quando in realtà è chi esegue l'attacco ad avere pieno controllo sulla comunicazione.
Come nella maggior parte delle trasmissioni possiamo usare per descrivere la comunicazione tra i due dispositivi il modello client server. Modello nel quale il primo dispositivo (client) idealmente invia solo richieste al secondo dispositivo (server) che risponde con informazioni a seconda della richiesta inviata. Questo canale di comunicazione nel quale passano richieste e risposte viene distrutto durante un attacco MITM per venire sostituito da un nuovo canale di comunicazione con le stesse funzionalità del precedente ma controllato da chi esegue l'attacco. Chi esegue l'attacco si pone così al centro del canale di comunicazione e lo controlla come se si trattasse di un proxy ingannando il client fingendosi il server ma anche il server che si comporterà come se stesse continuando a comunicare direttamente con il client. Così quando il client manda una richiesta è chi esegue l'attacco a riceverla e non il server. A quel punto il MITM non fa altro che inoltrare la richiesta del client al server per poi risponderà come se stesse comunicando direttamente con il client quando invece manda la risposta a chi attacca. Risposta che una volta ricevuta dal MITM viene inoltrata al client che resta così ignaro del fatto che la connessione sia stata in qualche modo alterata.
Essere nel mezzo della comunicazione tra client e server da a chi svolge l'attacco l'accesso ai pacchetti che stanno vendendo trasferiti tra i due dispositivi. Questi pacchetti potrebbero contenere informazioni sensibili quali: passwords, login, credenziali etc. Chi svolge l'attacco ha completo controllo su dati pacchetti e a seconda di quale sia lo scopo dell'attacco può: scartarli per impedire delle specifiche funzionalità del server, analizzarli per carpire informazioni su uno dei due dispositivi oppure modificare i pacchetti per accedere ad ulteriori funzionalità.
Ci sono due tipi di attacchi MITM: attacchi passivi e attacchi attivi. Negli attacchi passivi chi svolge l'attacco riceve i pacchetti e li inoltra senza apportare nessuna modifica a questi. Negli attacchi attivi al contrario i pacchetti vengono manipolati da chi sta svolgendo l'attacco prima di essere inoltrati.
Possiamo ulteriormente suddividere gli attacchi MITM in sottocategorie a seconda del protocollo usato per svolgere l'attacco:

- ARP Poisoning-IP spoofing
- DNS Spoofing
- DHCP Spoofing
- Wi-Fi Eavesdropping
- SSL Stripping
- HTTPS Spoofing

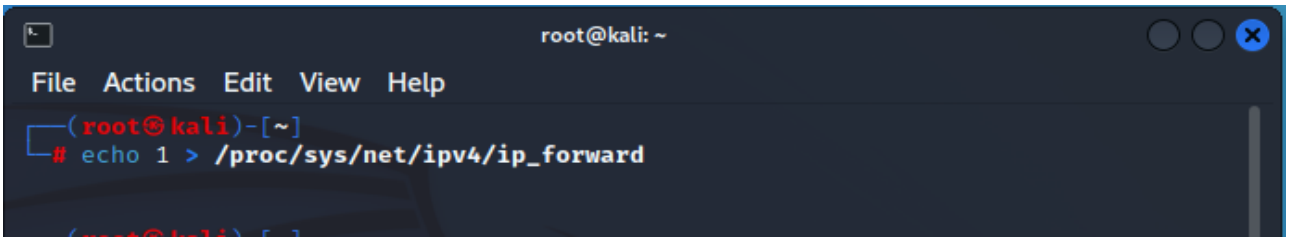
Per il test si è scelto di usare la prima tipologia ossia l'ARP Poisoning che è l'attacco MITM più comunemente usato, questo accade per la scarsa sicurezza del protocollo ARP e perché è il modo più semplice di eseguire questa tipologia d'attacco

Il protocollo ARP (Address Resolution Protocol) è un protocollo che si occupa di effettuare la mappatura tra l'indirizzo MAC e l'IP. Questi protocolli funzionano usando due tipologie di messaggi: richiesta e risposta. La comunicazione contiene due parti: host sorgente e hosts destinazione. La richiesta ARP è trasmessa ed è usata per capire quale indirizzo MAC si riferisce a un dato IP. Il messaggio di richiesta è trasmesso a tutti gli host in ascolto ma sono quello con l'indirizzo MAC corrispondente all' IP trasmesso nella richiesta risponde. Per abbassare il traffico dati ogni host ha una cache ARP nella quale salva in quali indirizzi MAC vengono mappati gli indirizzi IP di ogni host connesso alla rete.
Nell' ARP Poisoning si va proprio a avvelenare (poining) la ARP chace sfruttando la principale vulnerabilità del protocollo ossia che il protocollo ARP non prevede che il dispositivo salvi uno stato nel quale si trova per eseguire il protocollo e quindi distinguere tra messaggi leciti ed illeciti. Il dispositivo non ha uno stato nel quale non è in ascolto per richieste o risposte ARP questo significa che un dispositivo andrà a scrivere nella chace ARP qualsiasi risposta ARP gli arrivi anche se non ha mai trasmesso una richiesta ARP. Siccome le richieste ARP sono trasmesse a tutti gli host in ascolto qualsiasi dispositivo connesso alla rete è identificato come host e può quindi ricevere la rischista ARP. Chi attacca manda una risposta mandando un indirizzo MAC copiato e attacca così entrambe le parti della comunicazione. Tramite e risposte ARP sì modifica la chace ARP sia di client che del server. Da lato client chi attacca, con una risposta ARP, assocerà il proprio indirizzo MAC all' IP del server. Mentre al contrario dal lato server chi attacca, con una risposta ARP, assocerà il proprio indirizzo MAC all' IP questa volta del client Ogni comunicazione che verrà scambiata tra i due dispositivi sarà così disponibile a chi sta svolgendo l'attacco.

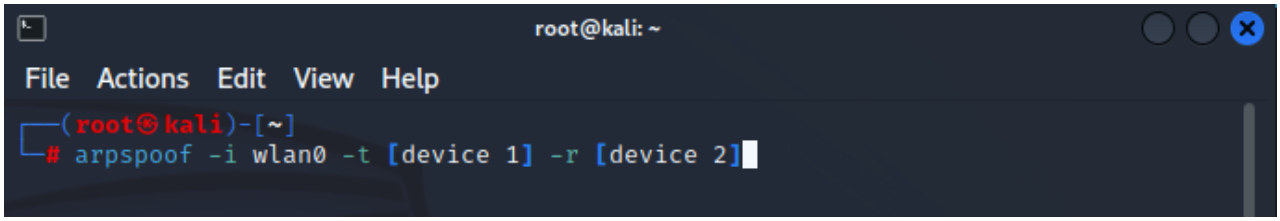
L'attacco sarà eseguito sempre su Kali Linux sfruttando tool preinstallati in questo. [11]
Per cercare gli indirizzi IP e MAC della rete pr eseguire l'attacco esiste un comando *nmap* la particolarità del comando è che gli si può passare un indirizzo generico come 192.168.1.* e il comando cercherà in automatico tra tutti gli indirizzi inviando segnali di vari protocolli. A seconda se l'indirizzo risponderà o meno e a quale messaggio risponderà si ha la possibilità di individuare gli indirizzi IP associati a dispositivi e fare delle assunzioni sul tipo di dispositivo connesso in base al protocollo alla quale ha risposto.



Purtroppo questo metodo esterno svolto totalmente da Kali Linux non ha portato ai risultati sperati, gli IP che sappiamo associati a dispositivi reali dagli attacchi precedenti e dall'analisi dei requisiti non hanno risposto. Si è stati quindi costretti a cercare di ottenere gli IP tramite WireShark.
Quando eseguiamo l'attacco stiamo solamente confondendo la rete su quale nodo corrisponde a quale computer dobbiamo mantenere il traffico della rete costante per non far notare che ci si è inseriti nella comunicazione. Dobbiamo quindi cambiare un parametro della scheda di rete per far si che il dispositivo sia in grado di rinviare i pacchetti ricevuti.



Ci si pone nel mezzo della connessione tra echo dot e lampadina con il comando arpspoof

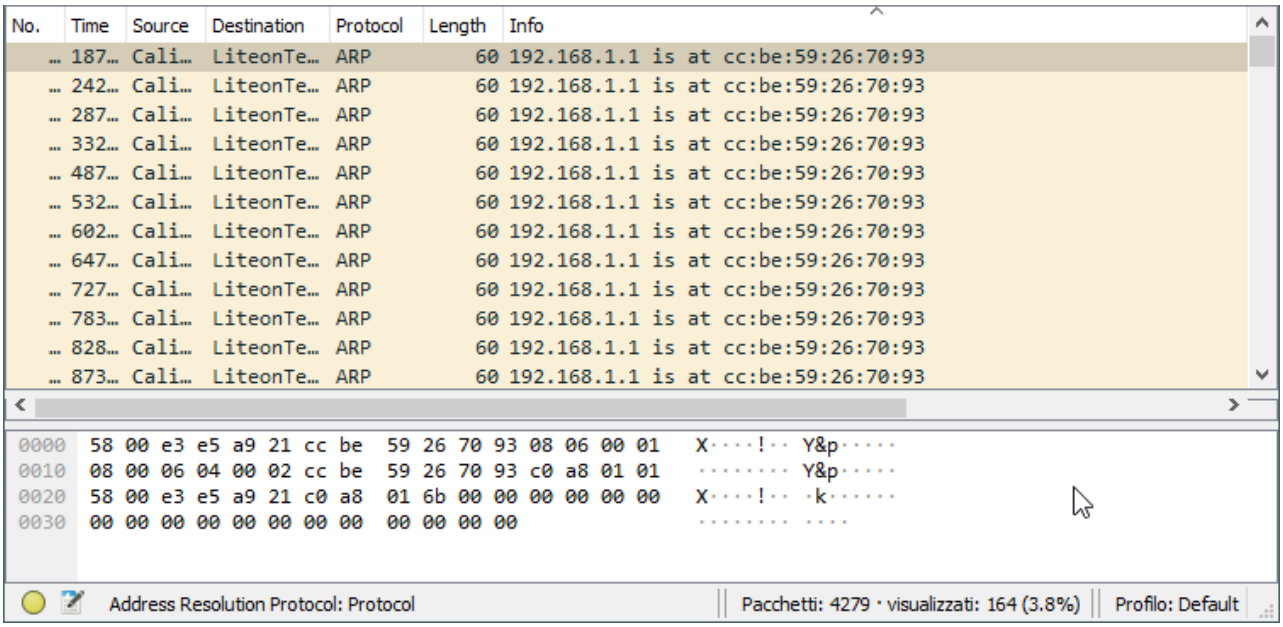


I vari flag stanno per;

- -i wlan0 → connessione utilizzata del tipo wlan0
- -t → dispositivo target
- -r → dispositivo attaccante

Il comando formula una richiesta ARP per tutti i dispositivi in ascolto che il nuovo IP del dispositivo target è quello del dispositivo attaccante poi con i cambi effettuati a livello di scheda di rete il dispositivo attaccante invierà tutti i pacchetti ricevuti al dispositivo target.

Bisognerà eseguire il comando due volte per avvelenare sia il registro arp della lampadina sia quello di echo dot ponendosi così al centro della connessione tra i due dispositivi. Controlliamo poi da WireShark il traffico di richieste ARP.



Kali Linux offre il comando dsniff per elaborare i dati immagazzinati dai pacchetti in passaggio per il disposqitivo attaccante sperando di carpire informazioni sensibili da questi.
Intercettando la comunicazione che avviene tra echo dot ed internet forse potremmo carpire informazioni sensibili ma ponendo la concentrazione su la comunicazione che avviene tra lampadina ed echo dot questa non ne contiene quasi nessuna. Gli unici dati che passano sono parametri per l'attivazione e lo spegnimento e la gradazione della luce.
L'attacco è stato un fallimento il canale tra i due dispositivi veniva infatti dopo poco rispristinato. Si è anche provato ad usare ARP poisoning per creare un disservizio andando a sostituire l'IP della lampadina nella ARP chace degli altri dispositivi così che non potessero arrivare a questa i comandi. Ma nemmeno ciò ha portato a risultati evidenti.
Da WireShark risulta un arrivo di richieste arp durante l'attacco ma evidentemente queste sono state scartate come illecite dalla rete che ha risposto bene a questa minaccia. Il comando Linux usato è uno dei più standard per l'ARP poisoning quindi è facile pensare che avere contro misure per renderlo inefficace siano alla base della sicurezza di un sistema moderno.

4.3.3 Replay [12]

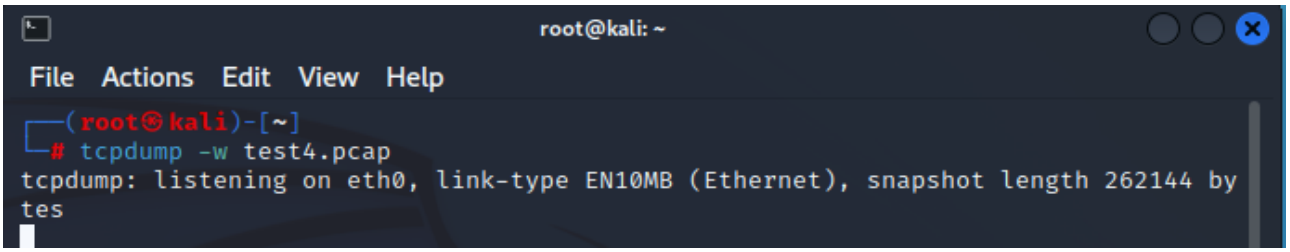
Non sempre gli attacchi MITM funzionano a causa della criptazione della comunicazione che avviene tra i due dispositivi. Se i messaggi scambiati fossero criptati infatti chi esegue l'attacco non potrebbe acquisire i dati necessari per porsi in mezzo alla comunicazione tra i due dispositivi ignari della presenza di chi svolge l'attacco. Per queste situazioni esiste l'attacco replay.
In maniera molto simile per quanto definito per l'attacco MITM nel modello Stride è generalmente l'attacco Replay è definito spoofing. Ma alcuni tipo di attacchi Replay potrebbero anche dare Information Disclosure, Denial of Service ma anche Tampering il tutto a seconda di che connessione viene attaccata e soprattutto cosa si decide di fare con i messaggi ottenuti.
La valutazione della minaccia nel modello Dread è:

	Punteggio
Damage	2
Reproducibility	1
Explitability	3
Affected users	2
Discoverability	3
Rischio tot	**11 (medio)**

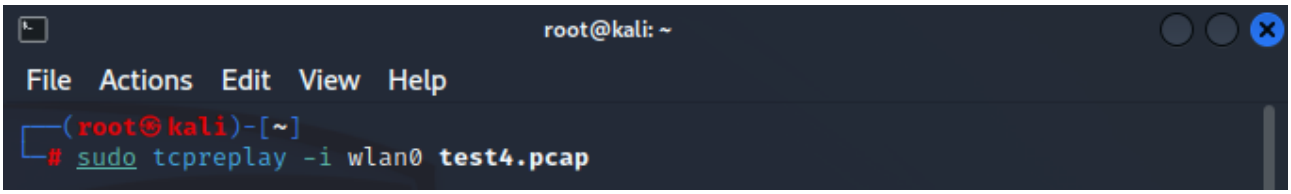
I protocolli criptografici impiegano al loro interno della crittografia per garantire delle funzioni di sicurezza. Ma per molti di questi protocolli la struttura e quindi la sicurezza degli algoritmi usati per la crittografia non sono considerati come parte del protocollo stesso. Questi algoritmi sono considerati perciò come infallibili all' interno del protocollo. Chi esegue il l'attacco dovrà girare attorno al problema usando i messaggi scambiati tra altri dispositivi anche se non sa né leggere né riprodurre i messaggi stessi.

Non essendo riusciti a porsi nel mezzo della comunicazione non si può aver accesso a tutti i pacchetti come se l'attacco MITM avesse avuto pieno successo quindi si dovrà solo ascoltare i messaggi dopo un eventuale accensione della lampadina per poi ritrasmetterli.

A questo punto anche senza aver controllo completo del canale di comunicazione si potranno rinviare i comandi svolti dalla lampadina così da farle svolgere quanto registrato ma in un secondo momento. Per fare questo si usano i comandi *tcpdump* e *tcpreplay* di Kali Linux già preinstallati sul sistema operativo per svolgere gli attacchi replay.
Si esegue prima il comando tcpdump per raccogliere i dati all' interno di un file. Il file nel quale si andranno ad immagazzinare i dati può essere specificato con il flag -w come facciamo in questo caso.



Si rinviano poi tutti i pacchetti prima raccolti con il comando tcpreplay, con il flag i si va a comunicare l'interfaccia da usare e gli si passa poi il file dal quale prendere i comandi precedentemente salvati



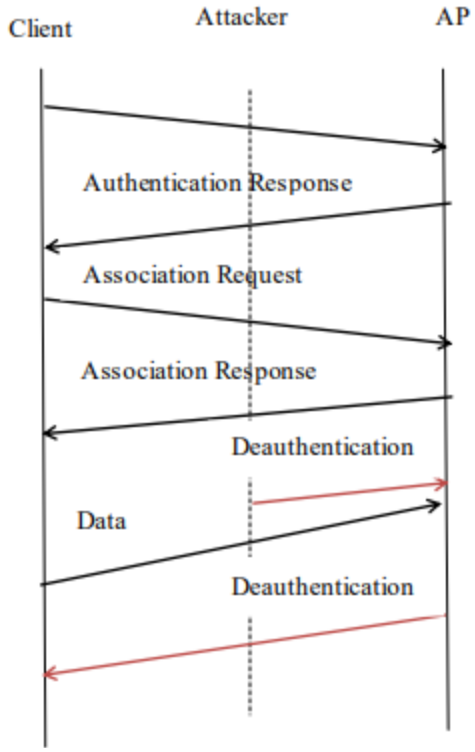
Purtroppo anche questo attacco è stato un insuccesso ma come avevamo appurato da i tentativi di attacco MITM la comunicazione tra i due dispositivi probabilmente riesce a distinguere tra richieste lecite e non lecite tramite sistemi di criptazione che rendono la rete sicura dagli attacchi più comuni. Sia per l'attacco MITM sia per l'attacco replay sono stati usati comandi standard e molto conosciuti per i quali ogni sistema moderno dovrebbe avere contromisure. Per ottenere risultati migliori si dovrebbero andare a testare altri protocolli di comunicazione e associazione tra i dispositivi della rete. Permettendo quindi una migliore comprensione dei messaggi inviati durante un attacco MITM che porterebbe a un conseguente attacco replay più facile e probabilmente più efficace.

4.3.4 Deautenticazione [13]

Perché i vari dispositivi appartenenti alla rete possano comunicare non solo la connessione deve essere attiva ma questi devono sapere con che altro dispositivo stanno comunicando in un dato istante. Questo problema di progettazione è risolto con un'associazione iniziale dei dispositivi che salvano i dettagli l'uno dell'altro per garantire connessioni future più veloci.
In questo penetration test si andrà a cercare di rompere questa associazione tra dispositivi cercando di impedire la connessione tra questi e di eventualmente sostituirsi nella connessione per prendere completo controllo del dispositivo IoT.
Nel modello Stride l'attacco può essere considerato a primo impatto un Denial of Service se si riuscisse a solamente a fermare la connessione tra i due dispositivi con un conseguente disservizio.
Ma se si riuscisse a impersonare un utente o un processo con una successiva nuova associazione a un dispositivo terzo allora si parlerebbe di Spoofing.
La valutazione della minaccia nel modello Dread è:

	Punteggio
Damage	3
Reproducibility	1
Explitability	3
Affected users	2
Discoverability	3
Rischio tot	**12 (alto)**

Se nel mondo dei dispositivi connessi via cavo non c'è mai bisogno di autenticazione tra trasmettitore e ricevitore perché connessi fisicamente nelle reti wireless le cose sono più complicate.
In questo caso infatti la autenticità della comunicazione deve essere controllata esplicitamente usando meccanismi di autentificazione come quelli a chiave condivisa o di sistema.
Inoltre solo chi fornisce il servizio può validare chi lo utilizza ad accedere al servizio stesso.
Chi effettua l'attacco deve porsi in mezzo a connessione come avevamo visto con l'attacco man in the middle precedente. Può ascoltare le comunicazioni che avvengono tra i due dispositivi e prendere controllo remotamente di uno di questi associandosi con un dispositivo terzo e mandando poi richieste di azioni o servizi.
Ma si può anche creare un disservizio come mostrato nell' immagine precedente con un dispositivo che attacca ponendosi in mezzo alla connessione e mandando messaggi di disconnessione appena sono sentiti messaggi di associazione. Questo metodo ovviamente prevede che in precedenza chi progetta il dispositivo che svolge l'attacco abbia ascoltato e registrato il messaggio che era stato trasmesso per associare e dissociarsi tra i due dispositivi.



Schema attacco deautentication generico

Ovviamente una volta che il vero client viene privato dell'autenticazione si può andare a fingere di essere il client o l'AP (access point), inteso come ingresso a internet, a seconda di chi sia l'obiettivo finale dell'attacco. Chi attacca infatti con l'uso delle metodologie di un attacco replay può rimandare i messaggi di autenticazione e associazione e fingersi a tutti gli effetti il dispositivo che ha appena disconnesso.Questa tipologia di attacco ovviamente è resa inefficace se come spesso accade l'associazione di due dispositivi è un operazione svolta con una frequenza estremamente bassa . In questo caso soprattutto se si lavora con due dispositivi che scambiano spesso informazioni è difficile identificare se nello scambio sono avvenute richieste di associazione e risposte a queste.

La rete che prendiamo in esame è invece molto vulnerabile a ciò perché come abbiamo appreso nella parte di identificazione dei vettori di attacco per ogni componente della rete ci si è resi conto di come un'assenza di corrente elettrica porta la lampadina a perdere le informazioni dell'associazione costringendola a doversi riassociare con il dispositivo echo dot.

Se chi esegue l'attacco è abbastanza esperto potrebbe pianificare una disconnessione e dunque riuscire facilmente ad isolare la comunicazione che codifica l'associazione tra i due dispositivi usando poi un attacco replay per disconnettere la lampadina a piacimento.In seguito a una disconnessione tentando di effettuare un replay dell'associazione dei due dispositivi si è scoperta una nuova vulnerabilità del dispositivo IoT. Questo infatti per essere riassociato ha bisogno dell'utilizzo dell'app smart life questa permette di cercare il dispositivo sia con un codice noto solo al proprietario ma anche di cercare dispositivi nelle vicinanze che stiano cercando di associarsi. Cercando i dispositivi nelle vicinanze non si dovrà immettere nessun codice per associarsi alla lampadina. Questo fa si che basti scollegare il dispositivo e l'avvicinarsi successivamente a questo per garantirsi un completo controllo sulle funzionalità.

4.4 Documentation and reporting

Si valuterà ogni attacco e conseguentemente la rete utilizzando metriche di valutazione specifiche

4.4.1 Valutazione Attacchi

Ad ogni attacco viene assegnato un punteggio da 1 a 10 in base a come il dispositivo IoT abbia risposto nelle varie aree, seguendo i requisiti di sicurezza di un dispositivo IoT.

Denial of Service

Requisiti	punteggio
Informazione	10
Integrità	10
Anonimato	10
Confidenzialità	10
Privacy	10
Accesso	10
Controllo accessi	10
Autenticazione	10
Autorizzazione	10
Funzionalità	4
Resilienza	4
Auto organizzazione	4

Man in the Middle

Requisiti	punteggio
Informazione	10
Integrità	10
Anonimato	10
Confidenzialità	10
Privacy	10
Accesso	10
Controllo accessi	10
Autenticazione	10
Autorizzazione	10
Funzionalità	10
Resilienza	10
Auto organizzazione	10

Replay

Requisiti	punteggio
Informazione	10
Integrità	10
Anonimato	10
Confidenzialità	10
Privacy	10
Accesso	10
Controllo accessi	10
Autenticazione	10
Autorizzazione	10
Funzionalità	10
Resilienza	10
Auto organizzazione	10

Deautentication

Requisiti	punteggio
Informazione	10
Integrità	10
Anonimato	10
Confidenzialità	10
Privacy	10
Accesso	5
Controllo accessi	5
Autenticazione	5
Autorizzazione	5
Funzionalità	3
Resilienza	3
Auto organizzazione	3

4.4.2 Valutazione rete e dispositivo

Alla rete e al dispositivo viene assegnato un punteggio da 1 a 10 per ogni categoria calcolato come:

$$Valutazione = \frac{\sum_{Attacco}(valutazione\ attacco) \times (gravità\ attacco)}{\sum_{attacco}(gravità\ attacco)}$$

Requisiti	punteggio
Informazione	10
Integrità	10
Anonimato	10
Confidenzialità	10
Privacy	10
Accesso	8,6
Controllo accessi	8,6
Autenticazione	8,6
Autorizzazione	8,6
Funzionalità	6,7
Resilienza	6,7
Auto organizzazione	6,7

5 Conclusioni

Il dispositivo IoT alla fine del test ottiene un risultato almeno sufficiente in ogni categoria. Per questo, alla fine del test, non possiamo che valutare in maniera positiva la sicurezza della lampadina.
Come si è spiegato i requisiti di sicurezza di un dispositivo IoT sono divisi in varie aree e i vari attacchi sono stati scelti specificatamente per mettere sotto pressione ciascuna di queste. L'attacco Denial of Service doveva andare a comprometterne le funzionalità, gli attacchi MITM e replay dovevano andare ad intaccare l'informazione mentre l'attacco Deauthentication doveva verificare eventuali problemi con l'accessibilità ai servizi offerti dal dispositivo IoT stesso.
L'area nella quale il dispositivo si è dimostrato più sicuro è quella dell'informazione. Le informazioni scambiate con altri elementi della rete domestica sono state impossibili da decifrare e riutilizzare durante il test. Gli altri due attacchi hanno riscosso maggiore successo: con l'attacco Denial of Service che è riuscito a provocare rallentamenti alla connessione e così rendere temporaneamente inaccessibili i servizi offerti dalla lampadina. E l'attacco Deauthentication che ha funzionato però non tramite un comando a distanza ma piuttosto tramite quella che è in realtà a tutti gli effetti una manomissione hardware. Se non fosse infatti per l'assenza di corrente non si riuscirebbe a dissociare il dispositivo IoT con un attacco a distanza e sarebbe impossibile sfruttare poi la vulnerabilità dell'applicazione per entrare in controllo della lampadina. L'attacco risulta da un lato potenzialmente molto rischioso ma ha prerequisiti così difficili da soddisfare prima di essere eseguito che la sua riproducibilità è minima. Queste valutazioni ci portano ad escludere che la minaccia da potenziale possa mai diventare tanto reale da essere un serio rischio alla sicurezza. A livello di progettazione del dispositivo IoT il team di sviluppo si è evidentemente concentrato sui requisiti di informazione. Data la sensibilità dei dati scambiati in un ambiente domestico a contatto con la parte più privata della vita di ciascuno non può che essere una buona scelta. Ovviamente il dispositivo potrebbe andare in contro a dei disservizi a causa di attacchi DoS di successo ma questi non avranno lo stesso effetto potenzialmente catastrofico della perdita di dati personali e saranno solo un fastidio momentaneo per la lampadina.
Come spiegato anche durante i singoli attacchi per svolgere gli stessi sono stati usati tool facilmente ottenibili ed usabili da chiunque. Molti dei comandi usati hanno un flag -h (help) per accedere a un manuale che spiega al meglio la sintassi del comando e come usarlo. Quindi se questi attacchi fossero andati a buon fine il dispositivo e in secondo luogo la rete della quale questo fa parte sarebbero veramente vulnerabili a qualsiasi attacco. Nella spiegazione di ogni attacco ho tenuto a sottolineare come ciascuno di questi abbia poi metodologie per essere eseguito più moderne, diverse e più efficaci. Così anche se il dispositivo e la rete possono sembrare sicure alle più comuni tipologie potrebbero risultare vulnerabili a qualche attacco molto specifico o appena sviluppato per il quale, al momento, non ci sia ancora una difesa

Bibliografia

[1] D. Evans, «The Internet of Things. How the next evolution of the Internet is changing everything,» 4 2011. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf. [Consultato il giorno 8 2022].

[2] A. &. Y. X. &. C. B. &. J. M. Bacudio, «Penetration testing».

[3] M. C. D. Z. M. P. a. A. Z. F. Meneghello, «IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices,» *IEEE Internet of Things Journal ,* vol. 6, n. 5, pp. pp. 8182-8201, 2019.

[4] A. Gupta, The IoT Hacker's Handbook, A. Gupta, 2019.

[5] Kali Linux, «kali.org,» [Online]. Available: https://www.kali.org/. [Consultato il giorno 08 2022].

[6] WireShark, «WireShark.org,» [Online]. Available: https://www.wireshark.org/. [Consultato il giorno 08 2022].

[7] A. Shostack., «Experiences Threat Modeling at Microsoft,» 2008, pp. 4-5.

[8] D. L. e. M. Howard., «Writing Secure Code,» *Microsoft Press,* 2002.

[9] P. L. Qijun Gu, «Denial of Service Attacks,» San Marcos, TX , Park, PA.

[10] D. J. F. Enkli Yllia, «Man in the Middle: Attack and Protection».

[11] charlesreid1, «MITM Labs/Dsniffing Over Wifi,» [Online]. Available: https://charlesreid1.com/wiki/MITM_Labs/Dsniffing_Over_Wifi. [Consultato il giorno 8 2022].

[12] P. Syverson, «A Taxonomy of Replay Attacks,» Washington, DC.

[13] D. B. D. S. S. Rupinder Cheema, «Deauthentication/Disassociation Attack : Implementation and Security in Wireless Mesh Networks,» *International Journal of Computer Applications,* vol. 23, n. 7, p. 0975 – 8887, 2011.