Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Triennale in
Ingegneria Informatica

Penetration test di un dispositivo IoT

Relatore:
Prof. Mauro Migliardi

Laureando:
Alberto Castagnaro
1219641

Anno Accademico 2021/2022
Data di laurea 20 luglio 2022

A mio padre, che ha sempre sognato di vedermi laureato

Abstract

L'internet of Things e i dispositivi IoT stanno promettendo un'importante innovazione tecnologica, in quanto l'interconnessione dei dispositivi e la loro crescente presenza stanno impattando la vita quotidiana delle persone.

E' però necessario considerare varie tematiche nello sviluppo di questi dispositivi, a partire dalla sicurezza e dalla privacy che devono garantire. La possibilità di rendere qualunque oggetto di uso comune smart e connesso alla rete deve fare i conti con una maggior complessità nel progettare dispositivi e sistemi sicuri e solidi contro attacchi informatici che hanno il potenziale per rubare dati sensibili e interferire nelle attività delle persone.

Uno strumento potente per testare la sicurezza dei dispositivi IoT è il penetration testing: dei test, svolti da ethical hackers e professionisti, che deliberatamente cercano vulnerabilità e le sfruttano per dimostrare il grado di sicurezza dei dispositivi. Questo riesce a simulare gli attacchi reali che potrebbero essere eseguiti da utenti malintenzionati e trova vulnerabilità a cui si può porre rimedio in fase di sviluppo.

Questo studio esemplifica il processo di penetration testing di un dispositivo IoT attuando un attacco Man In The Middle e un attacco Distributed Denial of Service sul dispositivo Amazon Echo Dot di quarta generazione, analizzando i risultati e le loro possibili implicazioni.

Indice

Capitolo 1

Internet of Things

L'internet of Things (IoT) è un importante argomento che sta acquisendo sempre più importanza nell'industria tecnologica di oggi e che ha innumerevoli ambiti in cui trova applicazione.

Ma cosa si intende per Internet of Things? Il termine *Internet of Things* [1], usato per la prima volta da Kevin Ashton nel 1999, si riferisce al processo di connessione a Internet di molteplici dispositivi, sensori, oggetti di uso quotidiano e perfino di interi luoghi (le così dette *smart cities*). L'estensione di Internet a questo nuovo mondo comporta anche l'acquisizione di un'identità digitale da parte di questi dispositivi che permette loro di comunicare.

L'invenzione dei primi dispositivi IoT si ha circa dieci anni prima del primo uso del termine, nel 1990, quando la maturità raggiunta nelle tecnologie wireless permette la creazione di soluzioni Machine-To-Machine (M2M), in cui due dispositivi collegati alla rete sono in grado di scambiarsi dati e eseguire azioni senza l'intervento manuale da parte di un umano. Queste soluzioni erano inizialmente proprietarie e costruite appositamente per ambiti e reti aziendali, ma con il passare del tempo si iniziò a sperimentare la connessione di dispositivi di uso comune tramite l'IP (*Internet Protocol*) che crearono un nuovo ambito di ricerca sullo "smart object networking" e che getto' le fondamenta dell'Internet of Things di oggi.

Negli ultimi anni, sebbene l'idea di dispositivo connesso a internet quindi non sia nuova, vi è stata una crescita esponenziale delle tipologie e di ciò che riesce a fare un dispositivo IoT. L'implementazione su larga scala di dispositivi IoT promette di rivoluzionare la vita quotidiana delle persone e attira sempre più aziende su questo mercato. Oggi infatti i dispositivi IoT nel mondo sono oltre 10 miliardi [2] e si stima che arriveranno ad essere oltre 25 miliardi nel 2030,

generando un immenso volume d'affari.

I fattori che hanno permesso la diffusione esponenziale dei dispositivi IoT oggi sono molteplici e si devono allo sviluppo tecnologico e al crescente interesse in questo mercato:

- Connessioni sempre più diffuse che offrono un'alta velocità, un basso costo e fanno sì che ogni cosa possa essere connessa
- Diffusione globale e utilizzo dell'IP come protocollo standard di rete che fornisce una piattaforma ampiamente sviluppata.
- Miniaturizzazione di microchip e sensori che possono essere incorporati in dispositivi molto piccoli con un costo relativamente basso
- Efficienza nel produrre microchip che negli anni ha portato ad un aumento della potenza di calcolo, riducendo le dimensioni e il consumo energetico
- Progressi nell'analisi dei dati che riescono a garantire aggregazione e analisi di dati in maniera veloce ed efficiente
- Sviluppo del *Cloud Computing* che permette la distribuzione di potenti servizi di calcolo a dispositivi che altrimenti non ne disporrebbero

La crescita del mondo dell' *Internet of Things* quindi, in parallelo all'aumento dei dispositivi IoT in circolazione e alla mole di dati scambiati e del traffico Internet che essi generano, presenta delle tematiche [3] che vanno prese in considerazione nell'analisi e nello sviluppo di queste tecnologie:

1. Sicurezza: i dispositivi IoT lanciano delle nuove sfide dal punto di vista della sicurezza. Poter garantire la sicurezza dalle vulnerabilità dei servizi e dei prodotti IoT agli utenti deve essere una priorità fondamentale, soprattutto considerando quanto pervasiva questa tecnologia stia diventando nella vita di tutti i giorni. Eventuali vulnerabilità potrebbero rappresentare una perdita di dati sensibili e un accesso per futuri attacchi informatici. Questa sfida viene poi amplificata da vari fattori, tra cui l'interconnessione dei dispositivi, lo sviluppo su larga scala di dispositivi omogenei che hanno l'abilità di connettersi ad altri dispositivi e la scarsa sicurezza degli ambienti in cui questi dispositivi vengono installati. Gli sviluppatori e gli stessi utenti hanno l'obbligo collettivo di garantire che i dispositivi e i sistemi IoT non

vengano esposti a minacce che potrebbero comprometterli. Di conseguenza, sarà necessario un approccio collettivo per sviluppare soluzioni sicure e solide che siano adeguate alla portata e alla complessità dell'argomento.

2. Privacy: è necessario rispettare le scelte individuali in materia di privacy nel realizzare dispositivi IoT sempre più smart, considerando che le scelte individuali hanno molte sfaccettature e tutte devono essere rispettate. L'IoT sta ridefinendo il modo in cui i dati vengono raccolti, analizzati, usati e protetti. Questo crea un dibattito sulla privacy dell'utente e sull'utilizzo dei dati, in quanto la sensazione di privacy degli utenti è parte integrante della fiducia e della sicurezza che gli utenti stessi ripongono nei dispositivi e nei sistemi IoT. Il tracciamento e il monitoraggio dei dati che porta a offrire innovazione e servizi più efficiente e mirati si contrappone alla preoccupazione degli utenti di essere costantemente sorvegliati e che questi dati possano trapelare a malintenzionati. E' perciò necessario sviluppare strategie che trovino un compromesso e che garantiscano la privacy individuale, continuando a offrire l'innovazione di queste tecnologie.

3. Interoperabilità : un ambiente frammentato composto da dispositivi IoT proprietari con un'alta complessità può limitare l'integrazione tra dispositivi e peggiorare in generale l'esperienza dell'utente. Inoltre dispositivi IoT mal progettati o mal configurati possono scatenare conseguenze negative per la rete a cui si connettono. E' quindi consigliabile adottare degli standard, dei modelli di riferimento e delle best practices che contribuiscano a creare maggiori vantaggi per l'utente e migliori opportunità economiche per le aziende.

4. Questioni di sviluppo ed economie emergenti: L' *Internet of Things* promette applicazioni e benefici in svariati ambiti che non riguardano semplicemente il singolo utente, ma interi settori, tra cui l'agricoltura e l'industrializzazione sostenibile, la gestione ambientale, l'assistenza sanitaria, il monitoraggio della qualità e dell'uso dell'acqua. Queste sfide ad ampia portata che i dispositivi IoT possono affrontare non deve essere esclusiva dei paesi già industrializzati, ma regioni in via di sviluppo devono riuscire ad implementarle sopperendo alla mancanza di infrastrutture, competenze tecniche e mancanza di incentivi e investimenti.

5. Legislazioni e diritti: la diffusione e l'uso a livello globale dei dispositivi IoT solleva nuove questioni normative e legali. Un esempio può essere un utente

che si sposta in uno stato differente: il dispositivo IoT potrebbe trasmettere dati in un paese in cui le leggi sulla protezione dei dati sono diverse. La rapida evoluzione della tecnologia IoT spesso non è seguita e supportata dalle capacità politiche e normative degli stati. Altri problemi legali legati a questo comprendono il conflitto tra la sorveglianza e l'uso dei dati raccolti da parte delle forze dell'ordine e i diritti civili dei cittadini, la conservazione e la distruzione dei dati. Sono questioni molto ampie e complesse e per questo bisogna adottare linee guida che promuovano le capacità dell'utente di interagire e che ne preservino i diritti.

L' *Internet of Things* promette quindi una vera e proprio rivoluzione, un mondo "smart" interconnesso con dispositivi che collaborano con gli utenti e con l'ambiente esterno. Questo enorme cambiamento, che sta ridefinendo il concetto stesso di Internet, comunemente associato dalle persone al World Wide Web, deve però tenere conto di una miriade di sfide nel suo sviluppo mirato a offrire sempre più benefici agli utenti, alla società e all'economia.

1.1 Dispositivi domotici

I dispositivi IoT possono essere oggetti fisici, sensori, apparecchiature e altre macchine che raccolgono, scambiano e utilizzano dati per offrire dei servizi agli utenti. Questi, tra le varie applicazioni che hanno e che dopo analizzeremo, trovano un impiego significativo nelle abitazioni.

La domotica [4] è la scienza che si occupa di studiare tecnologie che migliorino la qualità della vita nelle case. E' proprio in questo contesto che i dispositivi IoT possono giocare un ruolo chiave: l'avere un ambiente interconnesso, in cui i vari dispositivi monitorano la casa, può offrire all'utente vari servizi come il controllo automatico della temperatura, assistenti vocali, il risparmio di energia elettrica e altro ancora. E' proprio in questo contesto che l'utente vede i benefici che l'*Internet of Things* può creare, ma potrebbe anche aumentare la percezione che tutti i dati che vengono raccolti dai dispositivi portino a un continuo monitoraggio sulla vita dell'utente. E' perciò fondamentale creare un ambiente e dei dispositivi sicuri che stimolino fiducia e rispettino i diritti e le scelte dell'utente.

1.1.1 Tipologia di dispositivi

Le categorie e le tipologie in cui i dispositivi IoT vengono suddivisi non sono universali. Spesso accade che le organizzazioni strutturino una propria tassonomia e categorizzazione dei dispositivi IoT in modo congeniale alle loro ricerche e analisi.

I dispositivi IoT possono quindi essere categorizzati in maniera differente, per esempio in base alla tipologia di dispositivo, all'ambiente in cui viene inserito o alla funzione che un dispositivo deve svolgere. In un report del McKinsey Global Institute [5] (figura 1.1) viene descritto l'ampio range di possibili applicazioni che i dispositivi IoT offrono per poter beneficiare gli utenti e le industrie.

Setting		Description	Examples
	Human	Devices attached to or inside the human body	Devices (wearables and ingestibles) to monitor and maintain human health and wellness; disease management, increased fitness, higher productivity
	Home	Buildings where people live	Home controllers and security systems
	Retail environments	Spaces where consumers engage in commerce	Stores, banks, restaurants, arenas—anywhere consumers consider and buy; self-checkout, in-store offers, inventory optimization
	Offices	Spaces where knowledge workers work	Energy management and security in office buildings; improved productivity, including for mobile employees
	Factories	Standardized production environments	Places with repetitive work routines, including hospitals and farms; operating efficiencies, optimizing equipment use and inventory
	Worksites	Custom production environments	Mining, oil and gas, construction; operating efficiencies, predictive maintenance, health and safety
	Vehicles	Systems inside moving vehicles	Vehicles including cars, trucks, ships, aircraft, and trains; condition-based maintenance, usage-based design, pre-sales analytics
	Cities	Urban environments	Public spaces and infrastructure in urban settings; adaptive traffic control, smart meters, environmental monitoring, resource management
	Outside	Between urban environments (and outside other settings)	Outside uses include railroad tracks, autonomous vehicles (outside urban locations), and flight navigation; real-time routing, connected navigation, shipment tracking

SOURCE: McKinsey Global Institute analysis

Figura 1.1: possibili applicazioni di dispositivi IoT

E' comunque evidente che, sebbene non ci siano categorie universali standard in cui i dispositivi IoT possono essere suddivisi, l'*Internet of Things* è una tecnologia che può essere estesa ad ogni aspetto della vita delle persone.

1.1.2 Interconnessione e dati scambiati

L'interconnessione e l'interoperabilità dei dispositivi Iot è uno dei punti chiave di questa tecnologia. La possibilità di un dispositivo di raccogliere dati e analizzarli si basa spesso su dei protocolli di comunicazione che il dispositivo IoT utilizza per collegarsi con server e database esterni [6]. Il ragionamento si estende quando i dispositivi, oltre a collaborare con server e database esterni, collaborano attivamente con altri dispositivi IoT. Analizzando i principali modelli di comunicazione, ne emergono prevalentemente quattro:

- Device-To-Device communications: in questo modello due dispositivi comunicano direttamente tra di loro senza un applicativo o un server intermediario. La comunicazione avviene su diversi tipi di rete (ad esempio IP su Internet, protocollo BlueTooth), la quale permette ai dispositivi di scambiare dati in modo da svolgere le loro funzioni.

- Device-To-Cloud communications: in questo modello un dispositivo IoT si connette direttamente ad un servizio cloud per scambiare e monitorare il traffico dati. La connessione avviene solitamente sfruttando sistemi di comunicazione già presenti, come connessioni Ethernet e Wi-fi, per collegare il dispositivo alla rete IP e successivamente al servizio cloud.

- Device-To-Gateway communications: in questo modello un dispositivo IoT si connette ad un *application-layer gateway* (ALG) che raccoglie i dati e fa da intermediario tra il dispositivo e il servizio cloud. L' ALG fornisce inoltre sicurezza ed altre funzioni come il trasferimento dei dati (Un esempio di questo modello puo' essere un fitness tracker che scambia dati con lo smartphone che a sua volta si connette al servizio cloud. In questo esempio è lo smartphone ad agire da *application-layer gateway*).

- Back-End sharing model: in questo modello viene permesso agli utenti di esportare i dati raccolti dai dispositivi IoT dal cloud in combinazione a dati raccolti da altre fonti. Questo modello estende il *Device-To-Cloud* model in modo che i dati siano accessibili da terze parti autorizzate che a loro volta raccolgono ed elaborano i dati.

Questi quattro modelli di comunicazione stanno alla base dell'interoperabilità dei dispositivi IoT, ma la tematica non si ferma qui, essendo molto complessa, e richiede un'analisi ad ampio spettro. Mentre Internet come lo conosciamo og-

gi si basa su protocolli standard e l'interoperabilità è una sua pietra miliare, nell'*Internet of Things* vi sono vari fattori che entrano in gioco.

La completa interoperabilità tra dispositivi e sistemi IoT avviene in misura variabile in differenti livelli dei protocolli di comunicazione dei dispositivi. Inoltre, l'interconnessione completa in ogni aspetto tra dispositivi non sempre è fattibile, ne' necessaria e richiesta (in aggiunta se questa viene imposta potrebbe perfino disincentivare l'innovazione e gli investimenti). Adottare standard e protocolli che definiscano la comunicazione tra dispositivi potrebbe tuttavia aumentare l'interoperabilità e di conseguenza incoraggiare l'innovazione e lo sviluppo di nuovi modelli di business sui dispositivi e sistemi IoT. Gli utenti poi sarebbero facilitati nello scegliere dispositivi IoT diversi che, indipendentemente da costo e servizi offerti, sarebbero in grado di integrare con altri dispositivi IoT e funzionare insieme.

A questo si devono aggiungere importanti considerazioni che aumentano la complessità dell'argomento. Innanzitutto alcuni produttori di dispositivi trovano un vantaggio di mercato nel creare un ecosistema IoT proprietario e chiuso ad altri dispositivi esterni. Questi produttori possono limitare l'interoperabilità dei dispositivi ai prodotti del proprio marchio. La scelta può essere estesa anche ai dati raccolti, con incompatibilità dei servizi cloud e del formato dei dati salvati. Il tutto può essere considerato come un vantaggio o uno svantaggio: da una parte si ostacola la concorrenza e l'innovazione, dall'altra un ecosistema chiuso fornisce un protocollo adattabile più rapidamente alle esigenze tecniche e di mercato.

Inoltre, un'altra considerazione che si può fare riguarda i costraint e i problemi tecnici ed economici. I produttori devono tenere in conto vari fattori, tra cui risorse tecnologiche limitate, costi che possono lievitare per implementare l'interoperabilità, scadenze strette e pressanti per l'uscita di nuovi dispositivi e test prolungati dei dispositivi stessi interconnessi ad altri. Queste tematiche portano le aziende a fare un'analisi dei trade-off necessari per l'economia dell'azienda stessa, che spesso vanno in conflitto con l'implementazione di una forte interoperabilità. La creazione di standard globali, sebbene possa essere un processo costoso per le aziende, aiuta a creare nuovi dispositivi aumentando il range di conoscenza tecnica accessibile e diminuendo i costi di sviluppo.

In aggiunta, è necessario considerare la legacy dei dispositivi già in circolazione, che devono rapportarsi a nuovi dispositivi. Anche in questo caso l'interconnessione tra dispositivi dovrebbe essere garantita, trovando un compromesso tra la compatibilità con dispositivi datati e una maggiore interoperabilità con nuovi

dispositivi. Bisogna considerare poi che gli utenti possono trovare delle difficoltà nel rapportarsi e configurare una sempre crescente quantità di dispositivi; per questo è importante rendere semplice e intuitivo la modifica delle configurazioni di dispositivi collegati a una stessa rete.

1.2 Sicurezza e privacy dei dispositivi

La sicurezza e la privacy che i dispositivi promettono sono alla base della fiducia che gli utenti hanno verso questi. E' quindi necessario che entrambi questi punti siano adeguatamente sviluppati e testati. Ogni giorno nuove vulnerabilità e metodi di exploit vengono scoperti, ed insieme al crescente numero di dispositivi IoT interconnessi, questo porta a un crescente rischio di cyber-attacchi che possono rubare dati sensibili, provocare malfunzionamenti nei dispositivi e nell'intera rete. Conseguentemente anche la fiducia degli utenti è a rischio ed eventuali vulnerabilità possono creare enormi danni economici e utenti che abbandonano i dispositivi IoT. Diventa quindi fondamentale creare sistemi IoT sicuri, resistenti agli attacchi e mantenibili nel lungo periodo, soprattutto considerando la crescente interconnessione dei dispositivi a livello globale e l'aumento dell'utilizzo giornaliero da parte degli utenti di questi sistemi. Dispositivi mal progettati dal punto di vista della sicurezza, datati o non aggiornati possono essere un punto di entrata per attacchi informatici e diffondere la vulnerabilità ad altri dispositivi fino a raggiungere un livello globale.

E' di primaria importanza seguire delle linee di principio e considerare dei fattori che rendano la sicurezza un requisito chiave per i dispositivi IoT. I dispositivi IoT possono avere diversi gradi di resistenza e resilienza agli attacchi informatici, da quasi nulla a molto solida, ma è importante che eventuali vulnerabilità restino localizzate nel singolo dispositivo e non si propaghino ad altri e nella rete. E' comunque utopistico pensare che ogni dispositivo possa essere esente da attacchi, ma è necessario studiare dei trade-off che valutino quanto un dispositivo sarà compromesso, cosa questa vulnerabilità causerà e le risorse e tempo che dovranno essere impiegate per riacquisire un adeguato livello di protezione. Diventa inoltre opportuno valutare, attraverso uno studio sulla mitigazione del rischio, gli eventuali danni economici che una vulnerabilità può scatenare. Investire sulla sicurezza e resistenza agli attacchi di un dispositivo IoT può diventare estremamente oneroso per un'azienda, rendendo quindi necessaria una valutazione complessiva di tutti questi fattori. Andiamo quindi ad analizzare le principali

tematiche riguardanti la sicurezza dell'*Internet of Things* [7]:

- La quantità di dispositivi interconnessi su larga scala aumenterà drasticamente, andando a superare notevolmente quella dei dispositivi connessi a internet tradizionali. Questo, unito alla possibile abilità di molti dispositivi di collegarsi ad altri in modo indipendente, dinamico e a volte imprevedibile, potrebbe richiedere modifiche alle usuali strategie e metodi legati alla sicurezza.

- L'implementazione di standard potrebbe portare ad una omogeneità dei dispositivi IoT con caratteristiche affini. Un'eventuale vulnerabilità avrebbe un impatto molto più amplificato su gruppi di dispositivi IoT che utilizzano gli stessi protocolli o che presentano caratteristiche di implementazione e design simile.

- La durata di vita dei dispositivi IoT, come gli eventuali aggiornamenti, presentano delle criticità. Dispositivi che vengono progettati per durare molto, con eventuali problematiche nell'aggiornamento e riconfigurazione, devono essere progettati in modo da poter resistere nel loro intero ciclo di vita a possibili minacce informatiche, che altrimenti potrebbero perdurare a lungo. Lo stesso discorso vale per i dispositivi per cui non vengono previsti eventuali aggiornamenti e upgrade, dove l'impossibilità o la macchinosità di doverli aggiornare, unita al mancato supporto da parte degli utenti, potrebbe portare i dispositivi ad essere soggetti ad attacchi informatici senza poi riuscire a sanare queste vulnerabilità.

- Molti dispositivi IoT nascondono la maggior parte del loro funzionamento e dei flussi di dati all'utente. Alcune vulnerabilità che portano a dei flussi di dati anomali o ad ulteriori funzioni non volute ma nascoste all'utente potrebbero passare inosservate. Inoltre i produttori potrebbero rilasciare aggiornamenti che modificano la raccolta dati senza farlo sapere agli utenti. Problematiche simili si possono verificare fisicamente: dispositivi IoT installati in luoghi non sicuri potrebbero fornire un accesso fisico per dei possibili attacchi, mentre dispositivi progettati per essere integrati discretamente nell'ambiente, dove l'utente non li controlla direttamente, possono trovare difficoltà a segnalare violazioni della sicurezza o passare inosservati quando questi vengono utilizzati da utenti malintenzionati.

Si evidenzia quindi come la tematica della sicurezza sia fondamentale per l'espansione del'*Internet of Things* e come la questione sia ampia e complessa nel suo insieme.

Alla sicurezza poi si lega la privacy, un argomento molto dibattuto negli ultimi anni a cui gli utenti pongono una grande attenzione. I dispositivi IoT, per offrire i i propri servizi, raccolgono dati nell'ambiente in cui si trovano: questi dati sono spesso correlati anche agli utenti stessi. Gli utenti potrebbero poi non essere a conoscenza della raccolta dati effettuata dai dispositivi. La questione si amplia quando dati raccolti da dispositivi differenti vengono aggregati: più dispositivi apparentemente indipendenti che monitorano diversi aspetti della vita di un utente possono delineare, utilizzandoli insieme, un ritratto molto più chiaro e nitido della vita dell'utente stesso. La raccolta e analisi dei dati beneficia l'utente, offrendo servizi mirati, ma anche gli sviluppatori, che possono utilizzare quei dati per migliorare i loro prodotti o perfino per condividerli con terze parti. Si delineano quindi fattori da tenere in considerazione nello sviluppo e uso dei dispositivi IoT [8].

- L'interfaccia grafica dei dispositivi IoT non sempre è chiara e intuitiva per gli utenti, alcune volte non è neanche presente. Dispositivi che non presentano un'interfaccia con cui l'utente accetta la condivisione dei propri dati modificano l'idea tradizionale dove l'utente accetta dei servizi tramite cookie o permessi dello smartphone. Bisogna inoltre considerare che un utente che si relaziona a svariati dispositivi IoT durante la giornata non sarà sempre incline a interagire direttamente con i dispositivi per decidere l'utilizzo dei suoi dati. E' necessario delineare dei principi di protezione della privacy che siano scalabili alla moltitudine di dispositivi IoT, rimanendo comunque chiari e comprensibili agli utenti.

- Le potenzialità che hanno i dispositivi IoT nel raccogliere dati possono minacciare la privacy delle persone. I dispositivi IoT raccolgono dati della vita privata degli utenti, dove quest'ultimi si sentono solitamente "sicuri" e protetti. Alcuni sono poi in grado di raccogliere dati su più utenti contemporaneamente, senza che magari tutti ne siano a conoscenza.

- Le scienze e i processi che si occupano dell'analisi dati (ad esempio *Big Data Analytics*) sono in grado di utilizzare dati aggregati per analizzare la vita delle persone. Questo, amplificato dalla portata che hanno i dispositivi IoT, rappresenta un rischio per la privacy degli utenti. L'uso smodato dei

dispositivi IoT poi potrebbe far abituare gli utenti a questi ultimi e creare un finto senso di sicurezza, disinibendoli alla divulgazione dei loro dati sensibili e privati.

Capitolo 2

Penetration testing

Il termine *Penetration testing* (o anche *PenTesting*, abbreviato) indica un'analisi che si focalizza sull'individuazione di vulnerabilità in applicazioni e software mirate a dimostrare eventuali metodi di exploit per poi proporre soluzioni e misure che possano migliorare il livello di sicurezza informatica dei soggetti del test [9]. Si può quindi definire un penetration test come un tentativo autorizzato e legale di trovare metodi di exploit e vulnerabilità che permettano l'acquisizione di risorse o il controllo di un sistema, con l'obiettivo poi di renderlo più sicuro. L'idea generale è quella di provare a cercare problemi di sicurezza utilizzando gli stessi strumenti disponibili ad eventuali utenti malintenzionati, in modo da proporre un attacco fedele alla realtà.

E' importante sottolineare la differenza tra *Penetration testing* e *Vulnerability assessment*: il *Vulnerability assessment* è un processo che si limita all'analisi e alla ricerca di vulnerabilità nei sistemi, mentre il *Penetration testing* include metodi effettivi che provano l'esistenza delle suddette vulnerabilità, valindando o meno i protocolli di sicurezza usati e fornendo metodi di hacking utilizzabili.

Il penetration testing si propone quindi come uno strumento di analisi molto importante per la sicurezza di sistemi e software, con applicazioni importanti anche nel mondo dell'*Internet of Things*, dove la sicurezza dei dispositivi IoT sempre più interconnessi gioca un ruolo chiave nella loro evoluzione e diffusione.

2.1 Tipologia di attacchi

Le tipologie di attacchi che possono comporre un penetration test sono innumerevoli e frammentate. La stessa definizione di penetration test porta a questa diversificazione, dove gli obiettivi e le vulnerabilità variano da sistema a sistema

e dove l'accordo tra sviluppatori, proprietari e penetration testers determina la natura e la tipologia stessa del test che viene effettuato.

Si possono comunque delineare delle categorie di attacchi effettuati in un penetration test che si basano su vari fattori, come ad esempio le tipologie di funzionalità, dispositivi e software che si vanno a testare o la conoscenza preliminare del soggetto del test [10] [11]. Una prima classificazione dei penetration test può essere costruita su quanto i testers collaborano e hanno accesso a conoscenza e risorse del target, creando così tre approcci differenti:

- Black Box Penetration Test: questo test si basa su un approccio "black box" (scatola nera). I tester non dispongono di informazioni sull'hardware, sulle tecnologie utilizzate ed eventuali configurazioni. Si identifica quindi come un test fatto su un soggetto inizialmente sconosciuto, su cui tipicamente si attua una prima analisi automatizzata e successivamente manuale per delineare struttura e tecnologie usate. Il principale approccio si basa su tentativi ed errori per trovare vulnerabilità e difetti, impiegando però una notevole quantità di tempo.

- White Box Penetration Test: questo test si avvale di una conoscenza totale del soggetto del penetration test, attraverso informazioni varie tra cui codice sorgente, diagrammi architetturali e altro che vengono fornite ai testers prima dell'inizio dell'effettivo test. In questo modo si possono attuare test più specifici che accelerano il processo e risultano più mirati.

- Gray Box Penetration Test: quest test si presenta come una mediazione tra l'approccio Black Box e l'approccio White Box. I testers sono in possesso di informazioni parziali e non complete sul soggetto del penetration test, che vengono utilizzate come punto di inizio dei test per poi ampliare a funzionalità e metodi che richiedono informazioni non conosciute a priori.

Un'altra classificazione può essere fatta sui testers, coloro che eseguono il penetration test. In base al loro ruolo e all'eventuale ausilio del reparto IT dell'azienda cliente si possono differenziare vari tipi di penetration testers.

- External Testing: i penetration test esterni hanno l'obiettivo di capire quanto un hacker malintenzionato può arrivare ad ottenere l'accesso a un software o sistema e quanto in profondità può spingersi. Questi test utilizzano attacchi che solitamente usano un approccio Black Box, con informazioni e risorse che si cercano in rete.

- Internal Testing: i test interni vengono solitamente effettuati da testers interni all'organizzazione, in modo da analizzare situazioni ed eventuali implicazioni in cui attackers entrino in possesso di credenziali di accesso di persone interne all'azienda.

- Targeted Testing: i target test vengono effettuati da testers e dal dipartimento IT, in modo da analizare insieme al dipartimento IT la prospettiva e le strategie solitamente utilizzate da attackers esterni.

- Blind Testing: nel blind testing i testers operano esternamente all'azienda e simulano più fedelmente eventuali attacchi, in quanto i testers sono in possesso di pochissime se non nessuna informazione. La prospettiva "Blind" può interessare anche il dipartimento IT dell'azienda, il quale può essere notificato della natura dei test o può rimanere ignaro del fatto che si sta eseguendo un penetratrion test.

In aggiunta a queste due, si può stabilire una importante classificazione che categorizza le tipologie di vulnerabilità che vengono testate nel processo di penetration testing. I test effettuabili sono molteplici e si articolano generalmente in undici diverse categorie.

- Information Gathering: questi test si limitano alla raccolta di informazioni sul target del penetration test, utilizzando strumenti di analisi manuali e automatici.

- Configuration and Deployment Management Testing: questi test verificano la corretta configurazione degli strumenti utilizzati nello sviluppo del sistema e del dispositivo/applicativo. Configurazioni lasciate di default o mal impostate possono creare potenziali vulnerabilità.

- Authentication Testing: questi test analizzano la corretta procedura di autenticazione, controllando come l'autenticazione degli utenti è gestita e protetta da tentativi di brute-force o al furto di credenziali.

- Authorization Testing: questi test verificano l'adeguata protezione delle risorse in seguito ad attacchi eseguiti da utenti che non hanno l'autorizzazione per potervi accedere, garantendo un corretto access control.

- Identity Management Testing: è importante che gli utenti abbiano accesso solamente alle risorse a cui dovrebbero avere accesso. E' quindi importante

testare che vengano rispettati ruoli e privilegi di ogni utente, per evitare potenziali rischi che minino la sicurezza dell'intero sistema.

- Session Management Testing: questi test vanno ad analizzare come la gestione delle sessioni utente viene effettuata, verificando il corretto funzionamento e la protezione della sessione che deve essere implementata insieme ai requisiti comportamentali richiesti dagli sviluppatori.

- Input Validation Testing: questi test controllano la corretta gestione e sanificazione degli input ricevuti dall'utente. La gestione dell'input è una delle debolezze più comuni, in quanto la mancata sanificazione di input può portare a exploit, come le SQL injections o cross-site scripting, che minacciano la sicurezza dei dati.

- Client Side Testing: questi test vanno ad simulare ed analizzare vari metodi di exploit applicabili che si eseguono sul lato client (ossia dal lato dell'utente).

- Error Handling: questi test vanno a verificare la corretta gestione degli errori. Messaggi di errore o errori mal gestiti possono essere indicatori di potenziali vulnerabilità sfruttabili da utenti malintenzionati. E' quindi importante che questo aspetto venga correttamente gestito.

- Weak Cryptography: questi test vanno ad analizzare la robustezza e la sicurezza dei metodi e dei protocolli di crittografia utilizzati, in modo da prevenire che dati sensibili possano essere facilmente decriptati.

- Business Logic Testing: questi test hanno lo scopo di prevenire errori e difetti nella logica dei processi, solitamente si testano delle funzioni fornendo input ed esaminando l'output risultante.

2.2 Tools utilizzabili

La gamma di tools sviluppata per i penetration test che si trova al momento nel mercato è molto ampia. Vi sono svariati software, sia a pagamento che open source, ideati per eseguire differenti attacchi, come sistemi operativi adatti specificatamente al penetration testing.

E' importante considerare questa moltitudine di strumenti come un valore aggiunto per i penetration testers, in quanto l'uso di questi, anche in modo complementare, porta a dei risultati efficaci e veloci. Deve essere il tester che, in base

agli obiettivi del penetration test, sceglie i più opportuni strumenti per creare un test quanto più efficace e realistico possibile.

2.2.1 Sistemi operativi

Gli strumenti base che permettono a una persona di effettuare un penetration test sono i sistemi operativi. Un sistema operativo che viene sviluppato con l'intento di eseguire penetration test si differenza dai soliti sistemi operativi come Windows e MacOS, in quanto presenta strumenti e configurazioni ad hoc usabili in penetration test ed ethical hacking sessions.

Questi sistemi operativi sono solitamente open source, si basano sul kernel linux e integrano numerosi hacking tools. Alcuni esempi di questi sistemi operativi sono:

- Kali Linux [12]: è un sistema operativo open source basato su una distribuzione linux Debian studiato appositamente per i penetration test e l'analisi forense. E' mantenuto e aggiornato dalla Offensive Security Ltd, permette l'installazione su vari dispositivi, contiene più di 300 programmi per i penetration test pre-installati, offre un kernel personalizzato ed è conforme allo standard *Filesystem Hierarchy Standard* per individuare file binari, file di supporto e librerie.

- BlackBox [13]: è anche questo un sistema operativo open source appositamente sviluppato per i penetration test, ma si basa su una distribuzione Linux Ubuntu. Progettato con un design minimale e funzionale per essere più veloce possibile, offre numerosi vantaggi come repository costantemente aggiornate e numerosi strumenti che spaziano in molti ambiti di penetration testing. Si presenta inoltre come una delle prime piattaforme che supporta il cloud per il penetration testing.

2.2.2 Softwares open-source

I software votati al penetration testing rappresentano dei tools utili per i penetration testers che possono utilizzarli in diverse situazioni, selezionando quelli che si adattano meglio alla tipologia di test che si vuole effettuare. Parallelamente ai sistemi operativi votati al penetration testing, i software (sia open source che a pagamento) offrono degli strumenti automatizzati e configurabili che solitamente non coprono l'ampio spettro delle possibili vulnerabilità che un software o un

dispositivo potrebbero presentare, ma si specializzano in alcuni ambiti in cui possono andare in profondità con l'accuratezza e l'efficenza dei test che propongono. In particolare i software open source possono essere uno strumento utile per chi si vuole approcciare al mondo del penetration testing, in quanto non presentano licenze a pagamento e spesso la gamma di tools che offrono rappresenta un valido aiuto nell'esecuzione dei test e nell'analisi dei risultati, automatizzando molti processi che richiederebbero un'importante quantità di tempo. Analizzeremo alcuni software per il penetration testing open source che offrono funzionalità diverse e che non sono in comparazione tra loro, ma possono essere usati in modo complementare per performare un test più ampio ed efficace.

- Nmap [14]: applicazione gratuita e open source utilizzata per l'analisi delle reti e la verifica della loro sicurezza; rientra nella categoria dei software *port scanner*. Nmap, utilizzando dei pacchetti IP in modo innovativo, riesce a determinare gli host disponibili nella rete, i servizi che offrono, che tipo di filtri sono in uso e molte altre caratteristiche. E' progettato per la scansione rapida di reti di grandi dimensioni ma permette anche analisi più approfondite di singoli host.

- Wireshark [15]: uno dei software più utilizzati al mondo per l'analisi dei protocolli di rete che rientra nella categoria *Web vulnerability scanner*. Possiede varie funzionalità: può ispezionare centinaia di protocolli approfonditamente, supporta i filtri e la lettura e scrittura in diversi formati di acqusizione, offre la lettura dei dati inviati in tempo reale e la decodifica di molti protocolli comuni. Aiuta l'utente fornendo un'interfaccia grafica ben definita ed è supportato da più piattaforme

- Metasploit [16]: è un framework open source della categoria *Vulnerability exploitation framework* che fornisce una serie di strumenti per eseguire penetration test in un sistema, automatizzando molti test che manualmente richiederebbero un'importante quantità di tempo. Questo strumento viene utilizzato per scoprire vulnerabilità, raccogliere informazioni e testare le difese contro exploit.

- Burp Suite [17]: software che rientra nella categoria *Net Scanner*, viene utilizzato per intercettare le richieste e le risposte tra browser e le applicazioni target. In versione gratuita vengono forniti altri strumenti utili per generare attacchi come *cross-site request forgery*(CSRF), brute force di password e enumeration degli username.

- John the Ripper [18]: tool della categoria *password cracking*, è uno strumento che permette attacchi a dizionario per identificare le vulnerabilità delle password in una rete, il cracking di password offline e attacchi brute force e rainbow crack (due tipologie di attacchi che si basano sull'andare per tentativi).

2.3 Procedura di penetration testing

La procedura standard di penetration testing (PTES) [19] consiste in sette fasi principali che coprono tutto ciò che riguarda un penetration test: dalle motivazioni iniziali che stanno alla base del test, alle fasi di raccolta dati e delineazione delle possibili minacce (in cui i *testers*, coloro che eseguono il test, lavorano dietro le quinte per comprendere meglio ciò che devono testare e le possibili vulnerabilità che presenta), fino ai metodi di exploitation e al report finale, in cui le competenze e le abilità dei testers entrano in gioco per sfruttare questi exploit e creare un report che catturi l'intero processo fornendo al cliente più informazioni utili possibili.

1. Pre-engagement Interactions: in questa fase, che avviene prima dell'effettivo inizio del penetration test, si discute con il cliente cosa esattamente verrà testato, definendo il progetto di penetration testing e il costo del servizio.

2. Intelligence Gathering: in questa fase si passa alla raccolta di informazioni sull'obiettivo, che principalmente si articola su tre livelli:

   - Livello 1, in cui la raccolta di informazioni avviene tramite tools automatizzati
   - Livello 2, in cui la raccolta di informazioni avviene tramite alcuni tools del livello 1 e alcune analisi manuali
   - Livello 3, in cui viene svolta una profonda analisi e ricerca di informazioni manualmente. Questo livello solitamente richiede molto più tempo.

3. Threat Modeling: in questa fase si crea un modello delle possibili minacce, tenendo conto di due elementi chiave: le risorse e i possibili attacchi. Si fa un'analisi di quali risorse sono più importanti di altre e quali minacce sono più rilevanti. Si cerca poi di creare un modello che emuli i tools e gli strumenti che un possibile *attacker* ha a disposizione e i possibili obiettivi

primari che possono essere soggetti ad attacchi. E' inoltre opportuno creare un modello che tenga conto di minacce *black-box*, in cui gli *attackers* non hanno effettuano indagini sul dispositivo/organizzazione da colpire, e un modello basato invece su un'analisi e una ricerca di informazioni svolta a priori dagli *attackers* stessi.

4. Vulnerability Analysis: in questa fase si attua il processo di ricerca di falle e vulnerabilità nel sistema soggetto al penetration test che potrebbero essere sfruttate da un eventuale *attacker*. La tipologia di vulnerabilità varia molto da caso a caso, come anche la riposta a tali vulnerabilità che bisogna testare. In linea di principio va effettuata nel test un'adeguata analisi che spazi sufficientemente in profondità e in ampiezza. L'analisi in profondità deve testare ogni singolo parametro di una determinata funzione in tutti i comportamenti possibili per cercare ogni possibile vulnerabilità. L'analisi deve essere poi sufficientemente ampia, andando a testare l'ampio spettro di opzioni e funzionalità che possono provocare lo stesso comportamento anomalo.

5. Exploitation: in questa fase ci si concentra esclusivamente sull'accesso a un sistema o a delle risorse aggirando le restrizioni di sicurezza. Questa fase dovrebbe essere ben pianificata e includere attacchi precisi, basandosi sull'analisi delle vulnerabilità eseguita precedentemente. Sfruttando le vulnerabilità riscontrate, si dovrebbe raccogliere una lista di obiettivi ad alto valore, come una considerazione sulla probabilità di successo degli attacchi e sul possibile impatto sul soggetto del test.

6. Post Exploitation: in questa fase si determinano il valore delle risorse e delle funzionalità a cui si ha avuto accesso nella fase di exploitation e se ne mantiene il controllo per analizzare come, a partire da queste, si possa sfruttare ulteriori vulnerabilità per accedere ad altre risorse e funzionalità, analizzando così l'effettiva portata dei metodi di exploitation. Le risorse e le funzionalità a cui si ha avuto accesso sono valutate in base alla sensibilità dei dati che contengono e alla capacità di ottenere l'accesso ad altri obiettivi partendo da queste.

7. Reporting: in questa fase si compone il report sul penetration test fatto. E' consigliato dalla PTES di dividere il report in due parti: una di sintesi, contenente gli obiettivi principali e i risultati del penetration test, e una

tecnica, contenente tutti i dettagli dei risultati e le raccomandazioni su possibili azioni correttive applicabili.

2.4 Standard e certificazioni

Sono stati creati vari standard di penetration testing come strumento di supporto utile all'individuazione di vulnerabilità al fine di porvi rimedio prima di eventuali attacchi esterni.

La standardizzazione delle procedure ha potuto individuare i principali fornitori di servizi di penetration testing, esaminandoli tecnicamente per garantire la qualità e migliorare la rapidità e la distribuzione di questi servizi alle utenze. Alla standardizzazione dei penetration test partecipano enti governativi in collaborazione ad organismi professionali.

I risultati dei penetration test variano in base allo standard e alle metodologie utilizzate: attualmente esistono cinque standard per i penetration test e sono: Open Source Security Testing Methodology Manual (OSSTMM), Open Web Application Security Project (OWASP), National Institute of Standards and Technology (NIST00), Information System Security Assessment Framework (ISSAF), and Penetration Testing Methodologies and Standards (PTES).

Esiste inoltre uno standard strettamente legato ai penetration test: lo standard ISO/IEC 27001 [20]. Questo standard contiene i requisiti per sviluppare e gestire un sistema di gestione della sicurezza delle informazioni e i penetration test rappresentano un indispensabile strumento nel rispettarlo.

Capitolo 3

Penetration Test sul dispositivo Amazon Echo Dot

L'analisi fatta sui dispositivi IoT e su cos'è e come si esegue un penetration test vuole mettere in correlazione quanto la diffusione di questi dispositivi debba essere legata a una sicurezza testata e garantita, in modo da rassicurare gli utenti sul fatto che i loro dati sensibili siano al sicuro da utenti malintezionati. Per esemplificare questo, si vuole realizzare un penetration test sul dispositivo Amazon Echo Dot di 4a generazione, eseguendo due attacchi: *Man In The Middle* e *Distributed Denial of Service*. Si andranno poi ad analizzare i risultati ottenuti, confrontandoli con le aspettative iniziali del test, per controllare che lo sviluppo del dispositivo sia conforme in sicurezza e protezione dei dati.

Le funzioni del dispositivo includono varie azioni eseguibili attraverso l'interazione vocale, tra cui: la riproduzione di musica, la creazione di elenchi di cose da fare, l'impostazione di sveglie, lo streaming di podcast e la riproduzione di audiolibri, oltre a fornire informazioni sul meteo, sul traffico e altre informazioni in tempo reale. Amazon Echo Dot, nella modalità predefinita, rimane in ascolto continuato e si attiva tramite la parola sentinella, ma i microfoni possono essere disattivati tramite tasto fisico *mute*. Il dispositivo si basa su una architettura Device-To-Cloud, utilizzando i *Amazon Web Services*(AWS). Si configura tramite app, sempre sviluppata da Amazon, e richiede una connessione ad Internet Wireless.

3.1 Dispositivi Amazon Echo

Amazon Echo è la gamma di autoparlanti e dispositivi intelligenti realizzati da Amazon che integra al proprio interno il software di assistenza vocale *Alexa*. I vari dispositivi, che si differenziano in base alla taglia, alla qualità del suono riprodotto e alla possibilità di avere un display o meno, hanno rappresentato un punto di svolta per i dispositivi IoT e una vera e propria diffusione di massa di questi dispositivi con assistenti vocali integrati. I dispositivi della gamma Echo, tramite il software di assistenza vocale, sono in grado di attivarsi tramite la parola sentinella "Alexa" (parola che può essere cambiata in "Amazon" o "Echo"). e possono eseguire azioni e rispondere a domande poste dagli utenti (sono comunque necessarie una connessione a Internet e i permessi dati dall'app dello smartphone con cui si configurano). I dispositivi sono a loro volta in grado di dialogare con altri dispositivi della stessa gamma, creando così un vero e proprio ecosistema digitale.

3.1.1 Storia e sviluppo del dispositivo

Il progetto di Amazon Echo ha inizio nel 2011, a cura della divisione di ricerca e sviluppo Lab126 di Amazon. Il nome originale doveva essere *Amazon Flash* e la parola di attivazione "Amazon", successivamente cambiata in "Alexa", il nome dell'intelligenza artificiale che anima i dispositivi. Viene rilasciato nel 2014 il primo dispositivo IoT della gamma Echo *Amazon Echo 1a gen*, un piccolo smart speaker con intregato il software di intelligenza artificale Alexa.

Il continuo miglioramente dell'intelligenza artificiale porta Amazon ad ampliare la gamma e la tipologia di dispositivi della gamma Echo, partendo dai piccoli speakers *Dot* (con nuove versioni 2,3 e 4 annunciati negli anni a seguire) e spaziando fino a dispositivi dotati di fotocamera e schermi touch screen con hardware migliorati [21].

Il dispositivo soggetto del penetration test, l *Amazon Echo Dot 4a gen* e' stato rilasciato al pubblico il 22 ottobre 2020.

3.1.2 Amazon e Amazon Alexa

L'azienda proprietaria della gamma Echo è Amazon, una compagnia tech multinazionale, fondata da Jeff Bezos nel 1994, che si occupa di e-commerce, cloud computing, intelligenza artificale e streaming digitale.

L'azienda ha sempre avuto un forte impegno verso l'innovazione tecnologica, e proprio da questo è nata l'intelligenza artificiale Alexa. Alexa è stata annunciata nel 2014 parallelamente ai dispositivi Echo, in quanto rappresenta l'anima di questi prodotti. Le capacità di Alexa, chiamate "*skills*" dall'azienda, si sono evolute nel corso del tempo e hanno ampliato il range di azioni che possono compiere. Dalle semplici ricerche sul web, alla riproduzione musicale, ad aggiornare la lista delle cose da fare, gli ingegneri Amazon hanno sviluppato la possibilità di progettare skills specifiche per altre app, in modo da far interagire i dispositivi IoT Echo con software e app terze, per aumentare il livello di interconnessione e le possibilità di utilizzo che un utente ha. La personalizzazione è stata inoltre migliorata nel corso del tempo, ampliando il range di lingue parlate da Alexa e quindi il bacino di utenza a cui i dispositivi Echo sono destinati.

3.1.3 Interazione con altri dispositivi IoT

L'interazione e le capacità dei dispositivi Echo, parallelamente al miglioramento di Alexa, sono aumentate esponenzialmente negli ultimi anni. I dispositivi Echo sono passati da essere dispositivi IoT indipendenti a creare un vero e proprio ecosistema e ad essere anche uno smart hub di controllo dei dispositivi domotici presenti in casa.

Lo sviluppo di nuove skills, come la diffusione dei dispositivi, ha permesso una sempre maggiore interconnessione tra dispositivi della stessa gamma ma anche dispositivi ed app esterne.

I dati e le statistiche confermanno questi trend: ad oggi Alexa ha superato le cento mila skills, come i dispositivi che la supportano, e nel 2020 quasi sette utenti su 10 che utilizzano smart speakers utilizzano Amazon Echo. Ad oggi Alexa rappresenta una delle intelligenze artificiali più avanzate al mondo [22].

3.2 Tools utilizzati e test effettuati

Il penetration test consisterà in un'iniziale fase di scanning, per analizzare e individuare l'indirizzo IP del dispositivo *Dot* all'interno della rete Wi-Fi , e i due successivi attacchi. In questo test non è stato utilizzato Kali Linux ma sono stati utilizzati tre software di penetration testing open source o gratuiti presenti online.

3.2.1 Man In The Middle Attack

Man In The Middle è un attacco informatico in cui un attacker ritrasmette segretamente, con la possibilità di alterare, le comunicazioni tra due parti che credono di star comunicando direttamente tra loro (esemplificato in figura 3.1) [23]. Questo attacco solitamente simula la normale comunicazione tra due utenze (ad esempio un client che si connette a un sito web), lasciando all'attacker il controllo e l'intercettazione attiva dei dati scambiati e delle comunicazioni.

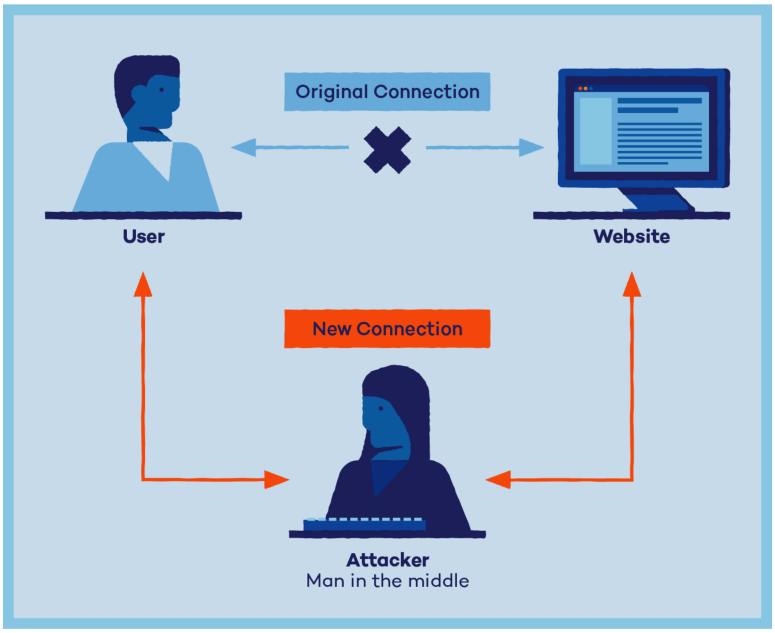

Figura 3.1: raffigurazione di un Man In The Middle Attack

Per contrastarlo, solitamente vengono implementati protocolli e forme di autenticazione degli endpoint, come ad esempio il *Transport Security Layer*. Il software utilizzato per la realizzazione del test è Wireshark, uno strumento di analisi dei protocolli di rete che essenzialmente intercetta e cattura pacchetti che si muovono all'interno della rete.

3.2.2 Distributed Denial Of Service Attack

Un attacco informatico del tipo *Denial of Service* mira a rendere una risorsa di rete (come ad esempio un server o un dispositivo IoT) temporaneamente non disponibile, interropendo i servizi da questa offerti [24]. Questo attacco si attua generalmente inondando la risorsa di richieste inutili e superflue, impendendo a quelle legittime di essere soddisfatte. Si parla di *Distributed Denial of Service*

quando il traffico generato in entrata non proviene da una fonte singola ma da più fonti, mascherando quindi la fonte dell'attacco (vedi figura 3.2).

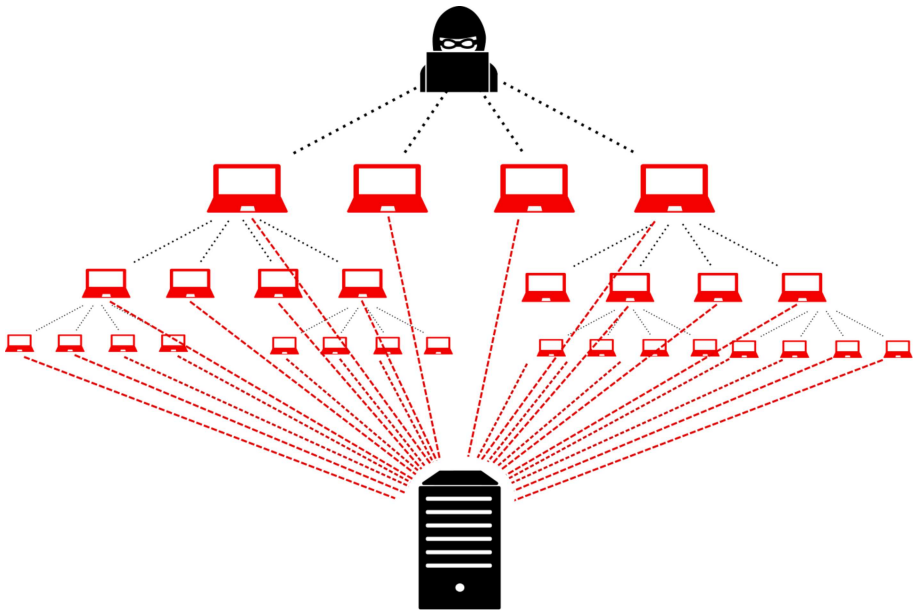

Figura 3.2: raffigurazione di un Distributed Denial of Service Attack

Lo strumento utilizzato per operare l'attacco è Metasploit, un framework open source che permette di scoprire varie vulnerabilità attraverso strategie di exploit e informazione raccolte su vulnerabilità già esistenti.

3.3 Procedura ed esito dei test

Il penetration test inizia con una fase di scanning per individuare l'indirizzo IP e le porte utilizzate dal dispositivo *Dot*, in modo da poter eseguire i due attacchi successivi.

Per realizzare questo è stato utilizzato il software open source Nmap, un'applicazione per la scansione di rete che offre svariate opzioni per la scansione di IP, host e porte. Per simulare un traffico di rete continuo, il dispositivo Echo è stato collegato a Spotify per una riproduzione musicale continua che ha comportato il continuo invio e ricezione di pacchetti dati.

Una volta identificato l'IP del router della rete Wi-Fi a cui il dispositivo è collegato, tramite Nmap è stata eseguita una prima scansione per individuare i vari host connessi alla rete, e una volta individuato l'IP del dispositivo Dot è stata eseguita una scansione più approfondita che ha permesso di individuare le porte utilizzate dal dispositivo Echo.

3.3.1 Procedura MITM attack

Avendo correttamente individuato l'IP e la porta utilizzata dal dispositivo, è stata possibile l'analisi dei pacchetti scambiati, indivuando i pacchetti che come mittenti o destinatari avevano l'IP individuato, filtrando tutti gli altri pacchetti irrilevanti ai risultati del test. Wireshark permette la visione dei mittenti, destinatari, il protocollo utilizzato, la lunghezza e il contenuto (quest'ultimo se criptato non viene visualizzato in chiaro).

3.3.2 Risultati e possibili implicazioni del MITM attack

I risultati del test hanno evidenziato come l'analisi dei pacchetti, sebbene la loro intercettazione sia riuscita, non ha portato all'individuazione di dati sensibili o utili, in quanto i contenuti dei pacchetti erano criptati. Questo risultato verifica l'uso dichiarato da parte di Amazon del protocollo *TLS 1.2*, che garantisce la sicurezza e la criptazione dei dati scambiati tra i dispositivi Echo e i server AWS.

3.3.3 Procedura DDOS attack

L'attacco DDOS è stato eseguito utilizzando il framework Metasploit. Individuato l'IP del dispositivo, è stato utilizzato un exploit già presente nel framework che ha permesso l'esecuzione dell'attacco. L'esito del test è stato verificato analizzando la rete con Wireshark, per verificare l'effettivo invio delle richieste, e analizzando eventuali anomalie nel funzionamento del dispositivo *Dot*.

3.3.4 Risultati e possibili implicazioni del DDOS attack

I risultati dell'attacco hanno evidenziato la perdita dei pacchetti da parte del dispositivo Echo che ha generato l'impossibilità di processare nuove richieste e di connettersi ai server Amazon, mandando di fatto il dispositivo offline (risultato verificato dai messaggi di errore riprodotti da parte del dispositivo in seguito a nuove richieste).

Le possibili implicazioni di questo risultato possono essere molteplici e dipendono dal grado di complessità dell'ecosistema IoT in cui si trova il dispositivo. Un dispositivo usato singolarmente per la semplice richiesta di informazioni non rappresenta un rischio per l'utente, mentre un dispositivo utilizzato come Hub di comando collegato ad altri dispositivi IoT come serrature smart, lampadine e

altre tipologie rappresenta una minaccia molto più concreta e pericolosa per gli utenti.

Capitolo 4

Conclusione

In questa tesi ho studiato la tecnologia dell'*Internet of Things*, analizzandola nel suo insieme ed approfondendo tematiche che stanno alla base di questa tecnologia, come la sicurezza, la privacy e l'interconnessione tra dispositivi. Ho inizialmente studiato la storia e le motivazioni che hanno contribuito alla diffusione e al miglioramento dei dispositivi IoT. Le potenzialità e i benefici che possono generare, in contrapposizione con problemi e criticità, sono poi stati oggetto di questa tesi, osservando come possibili considerazioni possano influenzare l'innovazione e le strategie di sviluppo del mondo IoT.

In ambito di sicurezza, ho approfondito la tematica del penetration testing, studiando le procedure standard e analizzando programmi operativi, tools e software open source presenti in rete e sviluppati per il penetration testing, con lo scopo di usarne alcuni per eseguire poi un penetration test. Ho categorizzato le tipologie di penetration testing utilizzando diversi criteri, in modo da fornire una panoramica quanto più esaustiva possibile.

Ho infine esemplificato un penetration test eseguendo due attacchi su un dispositivo IoT (Amazon Echo Dot 4a gen), spiegando la procedura di attacco e analizzando i risultati ottenuti, ragionando su possibili implicazioni che essi possono generare.

Bibliografia

[1] Rose, K., Eldridge, S., Chapin, L. (2015). The internet of things: An overview. *The internet society (ISOC)*, 80, 1-50.

[2] *Internet of Things statistics for 2022*, `https://dataprot.net/statistics/iot-statistics/`, ultima consultazione 15/06/2022

[3] Z. -K. Zhang, M. C. Y. Cho, C. -W. Wang, C. -W. Hsu, C. -K. Chen and S. Shieh, *IoT Security: Ongoing Challenges and Research Opportunities*, 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, 2014, pp. 230-234, doi: 10.1109/SOCA.2014.58.

[4] Trisciuoglio, D. (2009). *Introduzione alla domotica. Tecniche nuove.*

[5] Manyika, J., Chui, M., Bisson, P., Woetzel, J., Dobbs, R., Bughin, J., Aharon, D. (2015). *Unlocking the Potential of the Internet of Things.* McKinsey Global Institute, 1.

[6] Kulkarni, S., Kulkarni, S. (2017).*Communication models in internet of things: a survey.* International Journal of Science Technology and Engineering, 3(11), 87-91.

[7] H. Suo, J. Wan, C. Zou and J. Liu, *Security in the Internet of Things: A Review*, 2012 International Conference on Computer Science and Electronics Engineering, 2012, pp. 648-651, doi: 10.1109/ICCSEE.2012.373.

[8] Tawalbeh L, Muheidat F, Tawalbeh M, Quwaider M. *IoT Privacy and Security: Challenges and Solutions.* Applied Sciences. 2020; 10(12):4102. `https://doi.org/10.3390/app10124102`, ultima consultazione 17/06/2022

[9] Engebretson, P. (2013). *The basics of hacking and penetration testing: ethical hacking and penetration testing made easy.* Elsevier.

[10] tipologie di penetration testing https://www.cybersecurity360.it/soluzioni-aziendali/penetration-test-cose-come-funziona-e-a-che-serve/, ultima consultazione 22/06/2022

[11] overview e tipologie di pentesting, https://cyberdivision.net/2021/06/01/penetration-test-pt-cose-e-come-si-esegue/, ultima consultazione 22/06/2022

[12] Kali Linux OS, https://www.kali.org/, ultima consultazione 27/06/2022

[13] BlackBox OS, https://www.backbox.org/, ultima consultazione 27/06/2022

[14] Nmap tool, https://nmap.org/, ultima consultazione 27/06/2022

[15] Wireshark tool, https://www.wireshark.org/, ultima consultazione 27/06/2022

[16] Metasploit framework, https://www.metasploit.com/, ultima consultazione 27/06/2022

[17] Burp suite software, https://portswigger.net/burp/communitydownload, ultima consultazione 27/06/2022

[18] John The Ripper tool, https://www.openwall.com/john/, ultima consultazione 27/06/2022

[19] The Penetration Testing Execution Standard, http://www.pentest-standard.org/index.php/Main_Page, ultima consultazione 25/06/2022

[20] ISO/IEC 27001:2013, https://www.itgovernance.eu/it-it/iso-27001-it, ultima consultazione 30/06/2022

[21] Amazon Echo devices history, https://www.digitaltrends.com/home/history-of-amazon-echo/, ultima consultazione 1/07/2022

[22] Amazon Echo statistics, https://safeatlast.co/blog/amazon-alexa-statistics/#gref, ultima consultazione 2/07/2022

[23] Man In The Middle attack, Gangan, S. (2015). A review of man-in-the-middle attacks. arXiv preprint arXiv:1504.02115.

[24] Distributed Denial Of Service Attack, F. Lau, S. H. Rubin, M. H. Smith and L. Trajkovic, *Distributed denial of service attacks*, Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0, 2000, pp. 2275-2280 vol.3, doi: 10.1109/ICSMC.2000.886455.