UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

"ANALISI DI SICUREZZA DI UNA PRESA SMART"

Relatore: Prof. Mauro Migliardi

Laureando: Alberto Cappelletto

ANNO ACCADEMICO 2021 – 2022

17 Marzo 2022

Abstract

Il mercato dei cosiddetti dispositivi intelligenti, o smart, è in sempre maggiore espansione, e comprende un'enormità di dispositivi diversi che svolgono funzioni differenti. Visto l'aumento, sempre maggiore, di dispositivi utilizzati un ambito quotidiano sono sorte le prime domande circa la sicurezza di suddetti sistemi.

Lo scopo di questa tesi consiste nel verificare l'effettiva sicurezza di un particolare dispositivo IoT: una smart plug. Per verificarla innanzitutto verranno individuate e classificate le minacce e successivamente si procederà al controllo effettivo, tramite vari attacchi informatici, della sicurezza della presa in questione.

Premessa

Con l'introduzione degli oggetti intelligenti, o smart in inglese, ci si ritrova di fronte ad un mondo sempre più digitalizzato e connesso. Questo aumento però se, da un lato porta ad un mondo maggiormente accessibile e comodo, da un altro, costringe a porre l'attenzione sulla loro sicurezza.

Con l'aumento dei dispositivi connessi infatti, sono aumentati anche i possibili vettori con cui gli hackers possono violare un sistema, ma, questo, passa inosservato ai più. Ciò è dovuto anche al fatto che l'utente medio, interessato principalmente ai benefici che un determinato apparecchio comporta, ignora che un prodotto IoT sia sottoposto alle stesse minacce degli altri dispositivi informatici con una connessione ad Internet. Infatti si pensa, erroneamente, che sia meno soggetto a violazioni informatiche, considerato che ha delle funzionalità ridotte rispetto ad un computer piuttosto che ad uno smartphone, e quindi si sottovalutano i possibili pericoli.

Per sicurezza informatica si intende l'insieme di processi con cui si proteggono le informazioni, i dispositivi e le risorse personali. Questo insieme si basa su tre elementi, a cui si fa comunemente riferimento con l'acronimo CIA[1] (fig 1.1):

- Confidenzialità: proteggere le proprie informazioni sensibili da utenti non autorizzati.
- Integrità: verificare che le informazioni non siano state alterate in alcun modo.
- Accesso: controllare che sia possibile avere accesso alle informazioni quando necessario.

Oltre a questi tre elementi è necessaria una dose di buon senso e intelligenza da parte degli utenti.

[1] Confidenzialità, integrità e disponibilità (triade CIA): https://tecnologico.wiki/riservatezza-integrita-e-disponibilita-triade-cia/ , 20 Novembre 2021.

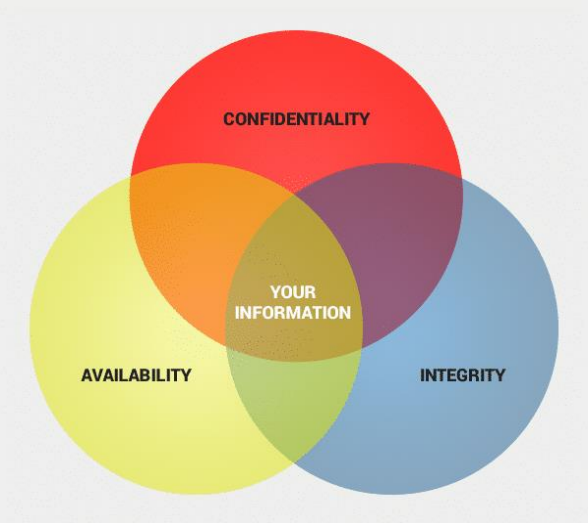

Figura 1.1 I tre elementi della triade CIA

Con questo elaborato si cercherà di valutare quanto sia vulnerabile una presa intelligente. Questa valutazione si strutturerà in più parti. Una parte introduttiva darà una panoramica sul mondo dell'Internet delle cose e sulla presa in questione, nella successiva si discuterà della metodologia usata per identificare le possibili minacce per poi giudicarne la pericolosità e infine, nella terza parte, si passerà all'effettivo controllo delle sopra-menzionate vulnerabilità, attraverso l'utilizzo di attacchi informatici.

È comunque importante far notare, a prescindere dai possibili risultati qui esposti, che non esiste un sistema informatico inviolabile, ma che, in un modo o in un altro, ogni singolo dispositivo informatico può essere potenzialmente compromesso.

1 Introduzione

1.1 Breve introduzione all'IoT

Con il termine Internet delle Cose o Internet degli Oggetti (Internet of things, IoT in inglese) si intende l'insieme delle connessioni ad internet di tutti quegli oggetti che vengono utilizzati quotidianamente, e per estensione (magari impropria) anche agli oggetti stessi che effettuano la connessione.

La quantità di connessioni che compongono l'Internet delle Cose è un numero enorme proprio perché i dispositivi che effettuano tali connessioni sono molteplici e possono spaziare tra innumerevoli tipologie (ad esempio può essere un termostato, un'automobile, un dispositivo medico, un tostapane e così via…). Il motivo per cui questi dispositivi effettuano delle connessioni è quello di scambiare informazioni, acquisire dati ed interpretarli ed è anche ciò che rende questi dispositivi smart.

Un tipico oggetto smart (molte volte viene usato come sinonimo di IoT) effettua una connessione, il più delle volte tale connessione è wireless, fornisce e riceve dati dall'applicazione che ne gestisce i comandi (se esiste) o dai propri sensori; una volta ricevuti, tali dati vengono interpretati e analizzati (manualmente con l'intervento umano o attraverso sistemi di intelligenza artificiale e machine learning)[2].

1.2 Sicurezza IoT

Vista l'enorme quantità di dispositivi IoT in circolazione, che secondo uno studio di IoT Analytics nel 2020 erano circa 11.3 miliardi destinati a crescere secondo le stime fino ad arrivare a circa 27.1 miliardi nel 2025[3] (figura 1.2), e considerata anche la fascia di prezzi facilmente accessibili in cui la maggioranza di questi dispositivi ricade, uno degli aspetti più importanti, molto spesso trascurato agli occhi dell'utente medio, è ciò che concerne la sicurezza degli stessi.

---

[2] Che cos'è l'Internet of Things (IoT)?: https://www.redhat.com/it/topics/internet-of-things/what-is-iot , 20 Novembre 2021.

[3] State of IoT 2021: Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion: https://iot-analytics.com/number-connected-iot-devices/ , 20 Novembre 2021.

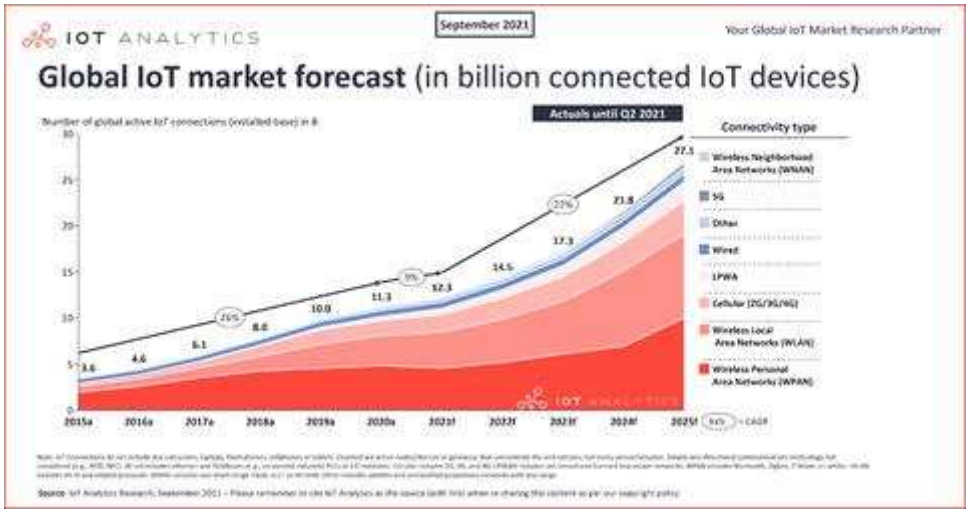

Figura 1.2 Previsione di IoT Analytics riguardo il numero di connessioni IoT

Un oggetto smart, pensato principalmente come uno strumento per facilitare o rendere più automatizzate alcune azioni, risulta essere in molti casi meno protetto di un normale computer o di uno smartphone, nonostante sia, di fatto, soggetto allo stesso numero di pericoli visto che, proprio come un computer, possiede la capacità di connettersi ed interagire con altri prodotti informatici.

Ciò dipende anche dal fatto che, nonostante si parli da diversi anni di IoT, è solo in tempi molto recenti che l'internet delle cose è esploso ed è diventato parte della quotidianità delle persone. Infatti, come solitamente succede ogni qual volta un'innovazione viene prima presentata e poi utilizzata, emergono delle criticità o vulnerabilità che la riguardano.

1.3 Amazon smart plug

Questa tesi si focalizza su un particolare prodotto appartenente al mondo IoT, una presa smart (smart plug) Amazon[4]. Questa presa può essere controllata tramite uno smartphone (sia iOS che android) collegandola ad una applicazione specifica. È possibile configurare la presa in modo tale da utilizzare i comandi vocali di Amazon Alexa per comandarla, è inoltre possibile impostare varie routines per eseguire determinati comandi in determinati periodi temporali.

[4] per il modello specifico si rimanda a: https://www.amazon.it/amazon-smart-plug-presa-intelligente-con-connettivita-wi-fi%C2%A0compatibile-con-alexa/dp/B082YTW968

L’applicazione[5] usata per il controllo permette anche di poter controllare più dispositivi contemporaneamente, di diverse tipologie.

Figura 1.3 Presa smart Amazon

1.4 Obbiettivo

L’obbiettivo di questo elaborato è di verificare l’effettiva solidità delle difese dello smart plug sottoponendola ad alcuni tra i possibili cyber attacchi a cui è esposto normalmente un oggetto appartenente alla stessa categoria. Tali attacchi usati avranno lo scopo di mettere in mostra le criticità del prodotto che potrebbero essere carpite.

1.5 Limitazioni

Il test di controllo delle vulnerabilità avverrà in modalità black box[6], poiché non è fatto in collaborazione con Amazon e quindi non ci è possibile conoscere la struttura del sistema. Questa modalità simula nel modo più verosimile un attacco reale, poiché un hacker solitamente non ha preventivamente accesso a informazioni sull’obbiettivo che intende attaccare, e quindi si trova nelle nostre stesse condizioni. Proprio perché non è una collaborazione con Amazon i test si concentreranno solamente sulla presa smart (software e hardware) che è stata

[5] l’applicazione specifica prende il nome di Amazon Alexa. Si rimanda a: https://play.google.com/store/apps/details?id=com.amazon.dee.app&hl=it&gl=US

[6] modalità di testing in cui il tester non ha informazioni che non siano disponibili pubblicamente, utile per determinare le vulnerabilità che possono essere sfruttate dall’esterno del sistema

regolarmente acquistata e quindi di nostra proprietà e non anche sull'applicazione o su database remoti di proprietà Amazon (poiché illegale). Infine in questa trattazione non verranno considerati attacchi di social engineering (ad esempio phishing/spear phishing) poiché non hanno a che fare con le debolezze della presa, ma con la competenza digitale dell'utente. Infine non verranno effettuati attacchi che si basano sulla previa compromissione di altri dispositivi connessi alla rete o la rete stessa per poter violare la presa perché ciò esula dall'obbiettivo di questa tesi.

1.6 Architettura dispositivi IoT

I dispositivi IoT sono pensati originariamente come dispositivi embedded che si connettono ad una rete (solitamente per le loro dimensioni e funzionalità hanno una quantità limitata di risorse a loro disposizione).

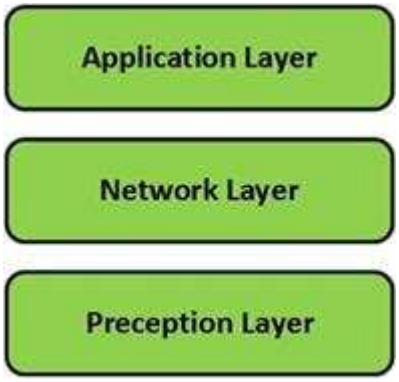

Figura 1.4 Livelli di una struttura IoT

In generale per i sistemi IoT si considera uno stack a tre livelli[7] (o layers): perception, network e application. Tali livelli sono qui brevemente spiegati:

- Perception layer: indica il livello fisico del dispositivo, è composto dai sensori e dai componenti fisici di cui l'oggetto è composto. La sua funzione è quella di collezionare i dati immagazzinati dai sensori, dati che variano a seconda delle funzioni dello strumento (temperatura, posizione, orario…) per poi passarli al livello successivo (network layer) in modo che possano essere trasmessi e interpretati.

[7] S. Pallavi e S. Smruti R. "Internet of Things: Architectures, Protocols, and Applications". (2017).

- Network layer: lo scopo di questo strato è quello di trasferire le informazioni ottenute dal livello precedente (perception layer) al sistema atto all'elaborazione dei dati, trasmissione che può essere sia wired sia wireless, dipende da dispositivo a dispositivo.
- Application layer: si occupa della gestione (e anche dell'interpretazione) delle informazioni che gli vengono trasmesse dal livello precedente (network layer).

Un ulteriore struttura che viene considerata è composta da cinque livelli. La struttura diventa così: perception, transport, processing (anche detto middleware), application e business layer.

Perception e application svolgono le stesse funzioni della struttura a tre livelli, il transport layer trasferisce i dati dei sensori dal livello perception a quello processing, il livello business si occupa dell'intero sistema IoT nel caso in cui in una sola rete vi siano molteplici dispositivi smart e un sistema centralizzato che si occupa della gestione. Lo stack più utilizzato, e a cui si fa maggiormente riferimento rimane comunque a tre strati, basta infatti considerare i due livelli transport e processing come un unico strato (network) e inglobare il livello business all'interno dell'application layer senza avere perdita di generalità.

1.7 Struttura di una smart plug

La struttura base di una presa smart generica consiste di varie componenti (le principali sono illustrate nella figura 1.5):

- Un pulsante per l'accensione o lo spegnimento manuale, che in questa presa corrisponde anche al pulsante adibito al reset.
- Un led con lo scopo di indicare lo stato della presa (collegato, scollegato, spento, in fase di accoppiamento).
- Un dispositivo per regolare la corrente da 240V a 3V (tensione adatta per la presa).
- Una scheda di rete wireless che consente la connessione tramite Wi-Fi tra la presa e l'applicazione.
- Dei bus che permettono alle varie componenti di interfacciarsi tra di loro (uno di questi è il Serial Peripheral Interface o SPI).
- Una spina che permetta di collegare alla corrente la smart plug e un'altra a cui collegare l'apparecchio che si vuole controllare da remoto.
- Un involucro che racchiude in maniera sicura tutte le altre componenti.

- Una scheda, o più, di memoria in cui è salvato il firmware e gli altri programmi che permettono il funzionamento della presa.

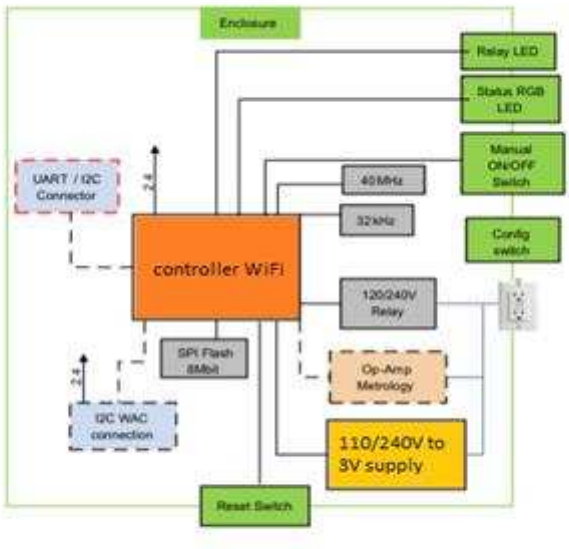

Figura 1.5 Diagramma a blocchi di una generica presa smart

Questi elementi costituiscono lo scheletro di una smart plug. Ogni dispositivo ha, comunque, le sue particolarità; alcuni possono avere oltre a quelli elencati sopra anche degli strumenti atti a misurare la temperatura o il consumo energetico dello stesso, ma nel caso della presa in discussione questi elementi non sono presenti. I diversi tipi di componenti hardware, e la loro disposizione all'interno dell'involucro, variano da modello a modello e da produttore a produttore e, inoltre, all'interno di una singola presa possono essere assemblati componenti di varie aziende.

1.8 Protocolli e standard per la connessione

Prima di iniziare ad eseguire i test è importante capire come la smart plug stabilisca una connessione e i protocolli alla base di tale connessione.

In generale gli smart devices comunicano con il resto dell'ecosistema IoT tramite uno dei vari standard di comunicazione (esempio Wi-Fi, Zigbee, Z-Wave e Bluetooth). Nel nostro caso la presa Amazon comunica da e verso l'applicazione di controllo, tramite Wi-Fi (ho quindi bisogno che dispositivo e strumento di controllo siano connessi sulla stessa rete). Per effettuare una connessione tra due host, o tra un client e un server sono stati definiti vari protocolli con lo scopo di stabilire degli standard e delle regole che qualunque dispositivo deve seguire per potersi connettere ad internet, tra gli standard più famosi e utilizzati vi sono: HTTP, HTTPS, MQTT, DNS, TCP, UDP, SSL, TSL, IP, ICMP ecc... MQTT è un protocollo di messaggistica usato proprio dai dispositivi smart poiché non richiede una quantità elevata di risorse.

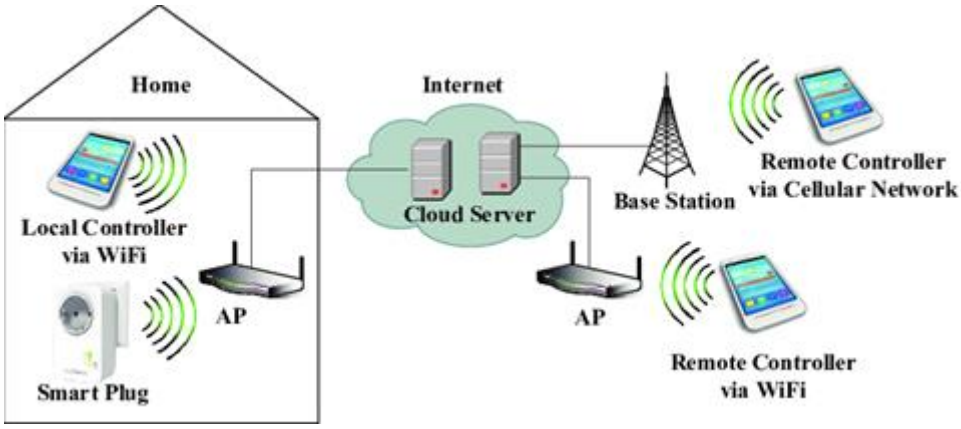

Figura 1.6 Scheletro di una connessione IoT

La connessione tra l'apparecchio che controlla la presa attraverso l'applicazione e la presa stessa passa in ogni caso attraverso il cloud che poi ritrasmette loro le informazioni. Come illustrato nella figura 1.6 una volta che lo smartphone si connette ad un access point, che può essere il router del Wi-Fi casalingo o un Wi-Fi pubblico se non si è in casa, o si connette attraverso i dati mobili del proprio fornitore di servizio e cerca di comunicare con la smart plug, le informazioni trasmesse arrivano ai cloud server appositi, che nel caso in questione sono quelli del cloud AWS (Amazon Web Services) e successivamente vengono ritrasmessi alla presa che esegue i comandi inviatele. Una volta eseguiti i comandi vengono inviati dei messaggi di risposta che nuovamente passano dapprima attraverso il cloud e successivamente arrivano all'applicazione, che aggiorna lo stato della presa visualizzato nello schermo a seconda della risposta.

Nella seconda parte di questa tesi, in cui ci concentreremo sull'effettivo testing della sicurezza dovremmo quindi tenere in considerazione i vari protocolli che il dispositivo usa e le loro possibili criticità (combinate con le debolezze del firmware o più in generale del software della presa) per stabilire quali sono i punti più vulnerabili (e quindi sfruttabili) del nostro sistema.

2 Ethical hacking

2.1 Metodologia usata

Il procedimento che useremo per testare la sicurezza del nostro oggetto sarà quello di definire un modello delle minacce che elenchi i maggiori punti deboli individuati e ne quantifichi nella maniera più oggettiva possibile il livello di pericolosità. Dopodiché procederemo, utilizzando vari approcci e strategie, a sfruttare le debolezze individuate per compromettere il nostro dispositivo e osservare i risultati che i nostri attacchi hanno prodotto.

Per comprendere cosa sia il cosiddetto "threat modeling" ci siamo documentati leggendo "Threat Modeling: a Systematic Literature Review" di Wenjun Xiong e Robert Lagerström, 2019. E, citando testualmente:

> "Threat modeling is proposed as a solution for secure application development and system security evaluations. Its aim is to be more proactive and make it more difficult for attackers to accomplish their malicious intents."[8]

É chiaro quindi che lo scopo sia quello di prevenire o quantomeno complicare il lavoro dei possibili attaccanti: ciò è possibile perché con questo sistema si cambia il modo di pensare, ci si focalizza infatti sui possibili modi in cui si abusa del sistema piuttosto che sulle possibili applicazioni e ambiti di utilizzo del sistema.

Quest'ultima parte è particolarmente importante per gli sviluppatori del sistema, più che per gli utenti, ma nel nostro caso ci aiuta ad osservare con occhio critico la situazione dal punto di vista dei possibili usi dannosi, che è il punto focale della nostra tesi.

Per identificare i vari tipi di minacce in cui la nostra presa smart può incorrere e per classificarli in categorie ci siamo basati sul modello STRIDE[9]:

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

[8] W. Xiong e R. Lagerström. "Threat Modeling: A Systematic Literature Review". (2019).
[9] A. Shostack. "Experiences Threat Modeling at Microsoft". (2008). pp. 4-5.

- Elevation of privilege

Tale modello, sviluppato da Praerit Garg e Loren Kohnfelder, è usato per identificare i vari tipi di minacce e ci aiuta a capire le reali vulnerabilità.

Una volta che le varie fragilità sono state individuate abbiamo utilizzato un altro modello per classificare il livello (alto/medio/basso) di pericolo di un attacco che sfrutta tale debolezza; il modello usato è il modello DREAD[10]:

- 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

Ad ognuna delle cinque categorie si assegna un punteggio che va da uno a tre a seconda che il pericolo sia basso, medio o alto rispettivamente, una volta fatto i punteggi vengono sommati e se il punteggio totale è tra 5-7 il rischio è basso, 8-11 rischio medio e 12-15 rischio alto. Per informazioni più dettagliate visionare la tabella 2.1 alla pagina successiva.

Per finire una volta valutato il livello di pericolo delle minacce individuate useremo i tipi di attacchi principali e più comunemente usati per sfruttarle e violare l'oggetto di questa tesi.

[10] D. LeBlanc e M. Howard. "Writing *Secure Code". (*2nd edition). (2002).

Categoria	Alto (3)	Medio (2)	Basso (1)
Damage	Può ottenere il pieno controllo del sistema	Può esserci una fuga di informazioni sensibili	Può esserci la fuga di informazioni non sensibili
Reproducibility	L'attacco è sempre riproducibile	L'attacco può essere riprodotto solo in specifiche condizioni o in uno specifico periodo temporale	È molto difficile riprodurre l'attacco anche con specifiche informazioni sulla vulnerabilità
Exploitability	Permette ad un utente alle prime armi di eseguire l'attacco	Un utente esperto può effettuare l'attacco ripetutamente	Permette ad un utente esperto e con una vasta conoscenza sulla vulnerabilità di effettuare l'attacco
Affected users	Colpisce tutti gli utenti o dispositivi del sistema, o comunque un numero molto elevato	Colpisce alcuni utenti o dispositivi	Colpisce una piccola percentuale di utenti e dispositivi
Discoverability	Può essere facilmente trovata la spiegazione di come sfruttare la fragilità per un attacco	La fragilità influisce su una funzione usata raramente e un utente avrebbe bisogno di una soluzione creativa per sfruttarla	È oscura e poco probabile che un utente scopra un metodo per sfruttare la fragilità

Tabella 2.1 Spiegazioni dei punteggi della modellizzazione DREAD

2.2 Struttura ecosistema IoT

Se vogliamo individuare i punti deboli dello smart plug Amazon da sfruttare dobbiamo capire la struttura del nostro sistema, suddividendola nelle varie parti che lo compongono, successivamente elencheremo le possibili funzionalità sfruttabili.

Di seguito sono elencate le componenti del sistema.

App Amazon Alexa

L'unica applicazione con cui si può connettere e controllare il dispositivo è "Amazon Alexa" sviluppata da Amazon e disponibile e per IOS e android (è disponibile anche una versione desktop per PC).

Tramite l'applicazione si possono controllare più dispositivi e si dispone anche (come si poteva intuire dal nome) delle funzionalità di Alexa, l'IA dell'ecosistema Amazon. L'accoppiamento dispositivo applicazione si può effettuare anche solamente connettendo i dispositivi a internet e scansionando il codice a barre sul retro della presa, senza il bisogno di inserire password o eseguendo altri comandi.

Wi-Fi

Il nostro dispositivo smart si connette ad internet utilizzando la tecnologia Wi-Fi che si basa sui protocolli IEEE 802.11 (standard che permettono a dei dispositivi di essere connessi tra loro in modo wireless).

La presa in particolare è predisposta per funzionare solamente con Wi-Fi a 2.4GHz, dove 2.4GHz è un certo tipo di frequenza delle onde che vengono impiegate per trasmettere le informazioni.

Smart plug Amazon

È composta da due elementi principali, il software, che nella nostra presa è la penultima versione rilasciata (versione 204000017), e l'hardware.

Il software è una collezione di istruzioni che spiegano al dispositivo come compiere un determinato compito, un software particolare è il firmware che ha l'incarico di avviare il sistema e permettere la comunicazione tra il software vero e proprio e l'hardware.

L'hardware è l'insieme delle componenti fisiche del dispositivo, dai sensori alla scheda madre fino ad arrivare al pulsante per l'accensione o lo spegnimento manuale.

Cloud

Amazon Web Services (AWS) è Il servizio di cloud proprio di casa Amazon, che può essere utilizzato anche da terze parti, in questo caso viene utilizzato per inviare comandi alla presa e per scambiare informazioni con essa. Ogni singola informazione da e verso la presa passa prima attraverso il cloud, che ne salva il contenuto al suo interno e poi se è necessario la ritrasmette.

Viene qui accennato solo per esaustività nonostante non verranno prese in considerazione minacce riguardanti questa componente dell'ecosistema poiché (come già spiegato nella sezione 1.5) trattasi di un servizio privato del quale non si dispongono i permessi per lanciare uno o più attacchi legalmente.

2.3 Identificazione minacce

Dopo aver separato le varie componenti, e quindi aver compreso meglio le dinamiche coinvolte è il momento di identificare le possibili minacce a cui può essere sottoposta la presa.

Ne sono state individuate 6:

1) Sfruttare una debolezza del software per caricare codice malevolo da eseguire all'interno della presa o per prenderne il controllo
2) Manomettere l'hardware della presa in modo tale che non funzioni correttamente
3) Rendere la presa inutilizzabile, o fare in modo che non risponda più ai comandi
4) Cercare di carpire informazioni eseguendo un attacco Man In The Middle (MITM)
5) Impersonare l'utente in modo da fare far eseguire alla presa dei comandi
6) Utilizzare un attacco a dizionario per scoprire le credenziali dell'utente che comunica con la presa

Si sottolinea come tali minacce non siano tutte quelle realmente possibili, ma solamente quelle che sono state individuate da noi tenendo in considerazione le limitazioni elencate nella sezione 1.5, e le risorse disponibili.

2.4 Classificazione minacce

Per valutare il livello di rischio delle minacce appena trovate useremo il modello DREAD prima citato (sez. 2.1). Di seguito i punteggi relativi:

minaccia #1: “Si potrebbe sfruttare una debolezza del software per caricare codice malevolo da eseguire all’interno della presa o per prenderne il controllo”	Punteggio
Damage	3
Reproducibility	2
Exploitability	2
Affected users	2
Discoverability	2
Rischio totale: medio	Somma punteggi: 11

Tabella 2.2 Punteggi DREAD assegnati alla minaccia 1

Minaccia #2: “Si potrebbe manomettere l’hardware della presa in modo tale che non funzioni correttamente”	Punteggio
Damage	3
Reproducibility	1
Exploitability	1
Affected users	1
Discoverability	1
Rischio totale: basso	Somma punteggi: 7

Tabella 2.3 Punteggi DREAD assegnati alla minaccia 2

Minaccia #3: “Si potrebbe rendere la presa inutilizzabile, o fare in modo che non risponda più ai comandi”	Punteggio
Damage	2
Reproducibility	3
Exploitability	3
Affected users	1
Discoverability	3
Rischio totale: alto	Somma punteggi: 12

Tabella 2.4 Punteggi DREAD assegnati alla minaccia 3

Minaccia #4: “Si potrebbe cercare di carpire informazioni eseguendo un attacco Man In The Middle (MITM)”	Punteggio
Damage	2
Reproducibility	2
Exploitability	3
Affected users	3
Discoverability	3
Rischio totale: alto	Somma punteggi: 13

Tabella 2.5 Punteggi DREAD assegnati alla minaccia 4

Minaccia #5: “Si potrebbe impersonare l’utente in modo da fare far eseguire alla presa dei comandi”	Punteggio
Damage	3
Reproducibility	3
Exploitability	3
Affected users	3
Discoverability	3
Rischio totale: alto	Somma punteggi: 15

Tabella 2.6 Punteggi DREAD assegnati alla minaccia 5

Minaccia #6:” Si potrebbe utilizzare un attacco a dizionario per scoprire le credenziali dell’utente che comunica con la presa”	Punteggio
Damage	3
Reproducibility	3
Exploitability	2
Affected users	3
Discoverability	3
Rischio totale: alto	Somma punteggi: 14

Tabella 2.7 Punteggi DREAD assegnati alla minaccia 6

2.5 Penetration testing

Per tutti gli attacchi di seguito illustrati è stata usata una macchina con Kali Linux[11] integrato. Si è scelta questa particolare distribuzione proprio perché pensata per testare la sicurezza informatica; in particolare è utilizzata per effettuare penetration testing, e perché ha già preinstallati la maggior parte, se non tutti, gli strumenti necessari per effettuare i tentativi qui esposti.

2.5.1 Sfruttare il software

Si passa all'analisi del primo possibile pericolo individuato ossia, ossia sfruttare una debolezza del software per caricare codice malevolo da eseguire all'interno della presa o per prenderne il controllo.

Per questa tipologia di attacchi si ha bisogno di avere una versione ufficiale del codice del software caricato sul dispositivo, un programma, o più programmi, che siano in grado di effettuare reverse engineering ed eventualmente un altro software da sostituire a quello ufficiale che permette di eseguire azioni non consentite da quello ufficiale.

Visitando il sito ufficiale di Amazon è possibile trovare alcuni file sorgente dei vari software per dispositivi smart di casa Amazon. Purtroppo al momento della scrittura di questa tesi tra i vari file non vi è presente quello relativo alla smart plug Amazon. Si è provato a scaricare e analizzare un paio di altri softwares e utilizzando il tool Ghidra[12], un software di reverse engineering sviluppato dall'NSA, il cui scopo è quello di tradurre un file da linguaggio macchina in linguaggio ad alto livello, in modo che sia capibile da un umano, il codice risultante risultava offuscato in molti punti (ad esempio i nomi delle variabili o delle funzioni venivano visualizzati come una serie casuale di caratteri alfanumerici), rendendo la comprensione molto difficile, se non impossibile.

Come ulteriore tentativo si potrebbe cercare di analizzare il codice sorgente dell'applicazione per smartphone, sfruttando i vari file con denominazione .apk disponibili online, ma si

[11] Distribuzione GNU/Linux pensata per la sicurezza informatica e per effettuare penetration testing. Si rimanda a: https://www.kali.org/
[12] cfr. https://github.com/NationalSecurityAgency/ghidra/releases

sconsiglia tale pratica, visto che il codice non è open source e quindi si sfocia nell'illegalità. Da notare comunque come anche quest'ultimo approccio rischia con molta probabilità di soffrire della stessa problematica già descritta, ovvero il codice sarà quasi sicuramente oscurato se non allo stesso modo in un modo molto simile a quello usato per i sorgenti dei vari dispositivi smart Amazon.

Preme sottolineare che, anche se i file fossero leggibili, sarebbe richiesto un livello molto elevato di conoscenze per analizzare il codice, capirne il funzionamento e individuare le vulnerabilità dello stesso.

Questo tipo di approccio è utilizzato per raccogliere informazioni sul funzionamento del dispositivo, informazioni che faciliteranno all'hacker la comprensione dei meccanismi del dispositivo e che potranno servire per scoprire bug o fragilità all'interno del codice che poi potranno essere sfruttate in seguito attraverso un'altra tipologia di attacco. Un ulteriore rischio consiste nel fatto che è possibile, dopo averne compreso il funzionamento, modificare il codice sorgente in modo che funzioni erroneamente e poi ricaricarlo sulla presa ed eseguirlo, facendo sì che si abbia il pieno controllo sulla presa, come se fosse un normale aggiornamento del software.

2.5.2 Manomissione dell'hardware

Navigando sul web, si trovano vari tutorial su come modificare il dispositivo in modo che si possa ottenerne il controllo senza averne le credenziali di accesso, o per permettere che svolga delle azioni per cui non era stato pensato e che potenzialmente possono essere pericolose. Alcuni esempi potrebbero essere: modificare le componenti hardware per aggiungere funzionalità non proprie della presa originale, cambiare alcune componenti originali con altre che permettono il controllo della presa anche ad utenti non autorizzati o aggiungere parti che consentano la creazione di una backdoor o di altri metodi per ottenere il pieno controllo del dispositivo.

L'obbiettivo che si ottiene con questo tipo di tentativi è principalmente quello di modificare la presa, per fare eseguire comandi, anche dannosi per la presa stessa, o per permettere all'hacker di assumere il totale comando del dispositivo e delle sue funzioni.

Tuttavia, ogni metodo richiede: un enorme livello di conoscenze per poter riuscire nell'effettiva manomissione, varie apparecchiature, di aver fisicamente accesso alla presa e

una quantità di tempo non indifferente. Quindi, sebbene un tale approccio dia effettivamente il pieno controllo dell'oggetto ad un malintenzionato, permettendo di comandare la presa a svolgere vari tipi di azioni, anche azioni che potrebbero risultare dannose per la presa stessa, siamo arrivati alla conclusione che il rischio collegato a tale attacco sia basso, tenendo in considerazione la mole di lavoro e di tempo richiesto per manomettere un singolo dispositivo.

2.5.3 Attacco Denial of Service

Uno degli attacchi più usati è il cosiddetto attacco Denial of Service o attacco DoS, e viene usato per impedire a degli utenti autorizzati di usufruire di un servizio, questo si ottiene in molti casi riuscendo a consumare o occupare le risorse disponibili di un processo, in modo da impedire a chiunque altro di averne accesso per utilizzarle, quindi, di fatto, rendendo il servizio inutilizzabile.

SYN flood è una tipologia di attacco DoS che sfrutta il protocollo TCP/IP, un protocollo utilizzato per stabilire una connessione tra due macchine che vogliono comunicare tra loro. L'attacco consiste, come lascia intendere il nome, nell'inondare di richieste SYN (synchronize) l'obbiettivo e nel non chiudere la connessione rispondendo alle richieste SYN-ACK (synchronize-acknowledgment) inviate dal sistema sotto attacco, il risultato prodotto è quello di rendere il dispositivo incapace di soddisfare le richieste di altri dispositivi, e quindi rendendolo irraggiungibile.

L'attacco si basa sulla procedura utilizzata per effettuare una connessione TCP, che prende il nome di stretta di mano a tre vie (three-way handshake). Tale connessione si ha quando un sistema cerca di effettuare una connessione TCP con un altro sistema. Per iniziare la connessione il primo invia un pacchetto SYN al secondo che risponde inviando un pacchetto SYN-ACK, dopo aver ricevuto questo pacchetto il primo sistema risponde con un pacchetto ACK e una volta che viene ricevuto dal secondo dispositivo la connessione è completata ed è possibile scambiarsi informazioni.

Il tentativo che verrà illustrato è stato condotto solamente contro la presa e non anche contro lo smartphone che la controlla da remoto ed è stato effettuato avendo accesso alla rete Wi-Fi su cui comunica il dispositivo (si è quindi in possesso di nome e password della rete).

Il metodo qui usato per sfruttare il processo di connessione e provocarne dei malfunzionamenti è un attacco diretto, il quale consiste nell'inviare una quantità massiva si richieste SYN in un

piccolo lasso di tempo al sistema vittima, e nel non rispondere (inviando i pacchetti ACK) ai pacchetti SYN-ACK inviati dal sistema sotto attacco. Il risultato che si ottiene è l'instaurazione di moltissimi tentativi di connessione, senza chiuderli (le connessioni rimangono nello stato half-open), costringendo la vittima ad utilizzare tutte le proprie risorse per rimanere in attesa di una risposta che consenta di portare a termine il collegamento e impedendo che tali risorse possano essere utilizzate da altri dispositivi per stabilire una connessione, di fatto rendendo il sistema sotto attacco irraggiungibile.

Per questo tentativo si è usato il framework Metasploit[13], uno strumento open source sviluppato proprio per il penetration testing, e Wireshark[14] per visualizzare il traffico in entrata e in uscita dal dispositivo sotto attacco. Una volta avviato il framework, per effettuare l'attacco è necessario conoscere l'indirizzo IP della vittima, trovato l'IP è possibile modificare vari parametri per personalizzare l'attacco (ad esempio scegliere la porta da "inondare"), in questo specifico caso sono state lasciate le impostazioni predefinite.

Una volta che il cyberattacco è stato lanciato, abbiamo verificato che la presa viene sommersa di pacchetti TCP, che sono stati catturati con Wireshark per visualizzare il traffico effettivo a cui viene sottoposta la vittima (fare riferimento alla figura 2.1), e, di conseguenza, non risponde a nessun tipo di comando, risultando completamente irraggiungibile dall'utente per tutta la durata dell'attacco. L'applicazione non notifica il problema all'utente finché non è l'utente stesso a provare ad accendere, da remoto, alla presa almeno una volta. Per interrompere questo attacco è necessario bloccare la fonte che instaura tutte queste connessioni half-open, scollegare e ricollegare la presa non risolve la situazione, perché la macchina attaccante continua a inviare pacchetti, che cadono nel vuoto finché la presa non è ricollegata, ma nel momento in cui ritorna connessa si ripresenta la situazione iniziale.

[13] cfr. https://www.metasploit.com/
[14] cfr. https://www.wireshark.org/

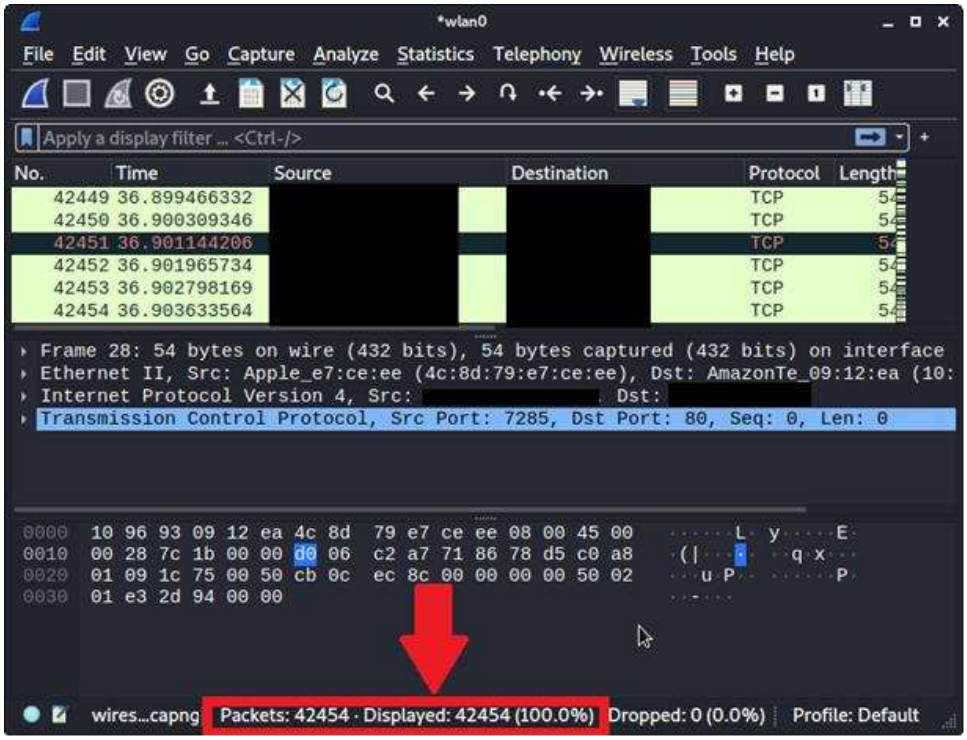

Figura 2.1 Numero di pacchetti TCP inviati durante l'attacco SYN flood

2.5.4 Attacco di deautenticazione

Un secondo tipo di attacco che mira a scollegare un dispositivo è un attacco di deautenticazione Wi-Fi (Wi-Fi deauthentication attack).

L'obbiettivo di tale attacco è quello di intromettersi tra le comunicazioni di un dispositivo e il router e si basa su una particolarità dello standard IEEE 802.11 che permette l'invio da un punto di accesso (access point o AP) di un frame di deautenticazione non criptato. Sfruttando questa funzionalità è possibile, fingendo di essere il dispositivo vittima, inviare una richiesta di terminazione della connessione al router che di conseguenza impedisce alla vittima di accedere ad internet, e, nel caso di una presa smart, di funzionare.

Per questo attacco è stata utilizzata la suite Aircrack-ng[15], composta da una moltitudine di strumenti per verificare la sicurezza di una rete Wi-Fi. Prima di cominciare con l'attacco vero e proprio è stato necessario predisporre la macchina attaccante in modalità monitor, la quale permette di ricevere tutto il traffico presente in una rete, anche quello che non è destinato al proprio dispositivo, e che quindi non è possibile vedere. Importante sottolineare come questa modalità funzioni solamente con connessioni wireless, mentre invece la modalità promiscua

[15] cfr. https://www.aircrack-ng.org/

che viene utilizzata solitamente per lo sniffing dei pacchetti funzioni sia con reti cablate che con reti senza fili. Dopo averlo fatto è stato necessario trovare l'indirizzo MAC del router e quello del dispositivo che si voleva disconnettere; per trovarli si possono usare vari modi, in questo caso sono stati trovati utilizzando la stessa suite con cui poi è stato lanciato l'attacco.

Un indirizzo MAC (anche detto BSSID) ha lo scopo di identificare in maniera univoca la scheda di rete presente su ogni dispositivo in grado di connettersi; è costituito da 12 caratteri (lettere o numeri) del tipo "xx:xx:xx:yy:yy:yy" (e può essere ad esempio: C4:F8:88:3F:26:44) ed è diviso in due parti:

- La prima parte è xx:xx:xx (nel caso dell'esempio appena fatto: C4:F8:88) ed identifica il produttore della scheda di rete.
- La seconda parte è yy:yy:yy (nel caso dell'esempio: 3F:26:44) identifica esattamente quella specifica scheda, similmente a come avviene per la targa di una macchina.

Una volta trovati i due indirizzi necessari, è possibile effettuare l'attacco lanciando il comando in figura 2.2 dal terminale:

```
aireplay-ng --deauth 0 -c [DEVICES MAC ADDRESS] -a [ROUTERS MAC ADDRESS] wlan0mon
```

Figura 2.2 Comando per eseguire deauthentication attack

Dove "deauth" indica un attacco di deautenticazione, "0" serve ad indicare che il numero di frame di deautenticazione inviati è infinito (l'attacco continua finché l'attaccante non lo interrompe), "c" indica il dispositivo che si vuole attaccare, "a" indica il router e "wlan0mon" indica il nome della rete Wi-Fi in modalità monitor.

In seguito al lancio del comando sopra descritto, si è effettivamente sperimentato che la vittima viene scollegata con successo dalla rete finché l'attacco non viene interrotto dall'attaccante stesso. Il dispositivo risulta disconnesso, e quindi irraggiungibile, per tutta la durata di tale attacco. L'applicazione non notifica questa situazione finché l'utente non tenta almeno una volta di farle eseguire un'azione.

Da notare come, per eseguire questo attacco, non sia necessario essere connessi alla rete della vittima, infatti, una volta in modalità monitor, la macchina attaccante, anche nel caso in cui

fosse connessa alla rete, si sconnette automaticamente per poter visualizzare il traffico trasmesso nel canale wireless.

Una considerazione ulteriore riguardante l'attacco appena descritto è che per la sua esecuzione non si è sfruttato uno specifico bug, ma la particolarità di un protocollo pensato per essere utilizzato con questa modalità. Quindi, si può dire che non solamente questa specifica tipologia di presa ad essere una potenziale vittima, ma ogni dispositivo che ha bisogno di essere connesso ad una rete Wi-Fi per funzionare.

2.5.5 Attacco Man in The Middle

Con il nome Man In The Middle (MITM), si intende una tipologia di attacchi informatici in cui un malintenzionato segretamente riceve e ritrasmette, anche alterandole, le informazioni che due sistemi si scambiano credendo di comunicare tra loro in modo diretto. Per impedire che tali tipologie di attacco siano efficaci sono stati implementati diversi protocolli di crittografia, alcuni più sicuri di altri, che rendono i messaggi scambiati dai due sistemi inutilizzabili a meno che non si sia in possesso della chiave di decrittazione, quindi rendendo un attacco MITM inutile, poiché anche nel caso in cui si riesca ad inserirsi nella linea di comunicazione, le informazioni che vengono intercettate sono incomprensibili e indecifrabili da chiunque non possegga la chiave o le chiavi usate per criptare e decriptare.

Per queste tipologie di attacco è necessario avere preliminarmente accesso alla rete su cui comunicano i dispositivi di cui si intendono intercettare le comunicazioni, altrimenti non ha alcun senso anche solo tentare di eseguire tale cyberattacco.

Un tipo di attacco Man In The Middle molto utilizzato è chiamato avvelenamento ARP (ARP poisoning) e sfrutta una debolezza del protocollo ARP.

Il protocollo ARP (Addres Resolution Protocol) si occupa della mappatura tra indirizzi IP e indirizzi MAC. Tale protocollo associa ad ogni indirizzo IP un indirizzo MAC e salva queste informazioni in una tabella (ARP cache). Ogni qualvolta un sistema informatico vuole conoscere l'indirizzo MAC di un sistema di cui ha l'indirizzo IP invia una ARP request che controlla nella tabella qual è il MAC associato all'IP fornito, lo restituisce con una ARP response e adesso il sistema che voleva questa informazione può inviare il messaggio all'altro. L'enorme falla di questo protocollo consiste nel fatto che chiunque può fornire l'associazione IP-MAC e salvarla nella tabella, senza che ne venga verificata l'attendibilità; quindi, anche un

hacker può manipolare la mappatura tra indirizzi presente nella tabella e, di conseguenza, ogni volta che viene effettuata una ARP request viene restituito l'ultimo indirizzo MAC salvato, ma che potrebbe benissimo essere stato manipolato ad hoc.

Per questo attacco si è utilizzato dsniff[16], un insieme di strumenti per l'analisi e lo sniffing (annusare) del traffico in una rete. Per iniziare è necessario avvelenare la tabella ARP con il seguente comando:

```
┌──(kali㉿kali)-[~]
└─$ sudo arpspoof -i wlan0 -t [TARGET IP ADDRESS] [HOST IP ADDRESS]
```

Figura 2.3 Comando per avviare ARP poisoning

Dove "arpspoof" è lo strumento utilizzato per effettuare l'ARP poisoning, "-i" indica l'interfaccia utilizzata, in questo caso la connessione è wireless ed è chiamata "wlan0", "-t" indica l'obbiettivo dell'avvelenamento e "HOST IP ADDRESS" è l'indirizzo IP della macchina attaccante.

Questo comando va eseguito due volte su due finestre di terminale separate, in una gli indirizzi sono: indirizzo IP vittima e indirizzo IP attaccante; nell'altra gli indirizzi sono: indirizzo IP router e indirizzo IP attaccante. Dopo aver eseguito i comandi la tabella contenete gli indirizzi risulterà avvelenata, e quindi farà credere al router che l'attaccante è la vittima e alla vittima che l'attaccante è il router. L'ultimo passaggio da eseguire è quello di permettere alla macchina attaccante di ritrasmettere le informazioni ricevute, in modo che la connessione risulti attiva e funzionante agli occhi della vittima.

In seguito a queste operazioni tutte le informazioni che vengono scambiate tra il router e la smart plug passano attraverso il dispositivo dell'attaccante che può ritrasmetterle così come sono o modificarle come meglio ritiene opportuno.

Purtroppo, sebbene, in questo caso, si sia riusciti a catturare le informazioni scambiate durante l'attacco, esse sono inutili, poiché per comunicare, sia la presa, che l'applicazione utilizzano il protocollo crittografico Transport Layer Security (TLS). Tale protocollo permette che i dati non siano leggibili da terzi crittografandoli, e conferma, attraverso certificati di sicurezza, che

[16] cfr. https://www.kali.org/tools/dsniff/

le parti che comunicano sono effettivamente chi affermano di essere, controllando che i dati non siano stati alterati.

Quindi l'attacco Man In The Middle non è riuscito, ma si è riusciti ad origliare la conversazione tra i due dispositivi, si tratta infatti di un caso di eavesdropping.

Per ovviare a questo problema si è provato ad utilizzare un metodo chiamato sslstrip[17] pensato da Matthew Rosenfeld, il metodo consiste nel forzare le parti che stanno comunicando ad usare una connessione non crittografata. Per ottenere questo risultato, dopo aver effettuato l'attacco MITM come in precedenza si utilizza uno script (chiamato proprio sslstrip) che trasforma la richiesta di connessione sicura nel suo equivalente senza il protocollo crittografico (ad esempio passare da HTTPS ad HTTP).

Anche dopo questi accorgimenti abbiamo verificato che la comunicazione tra le parti risulta ancora indecifrabile, poiché sia la presa che l'applicazione continuano a criptare i loro messaggi. Ci sono vari modi con cui è possibile ottenere questo risultato, un esempio è quello di implementare il protocollo HSTS (HTTP Strict Transport Security), un protocollo utilizzato appositamente per evitare il downgrade da HTTPS a HTTP delle connessioni, un altro metodo è quello di mantenere un livello di sicurezza in ogni singola parte della connessione o di abilitare cookies protetti (secure cookies) che limitano l'utilizzo di tali cookies solamente a canali sicuri.

In conclusione, non si riesce ad accedere o modificare le informazioni scambiate poiché viene utilizzato un buon livello di crittografia e dunque l'attacco MITM ha avuto esito negativo. Si potrebbero utilizzare altri metodi per riuscire nell'intento di carpire informazioni, come per esempio generare un gemello cattivo (evil twin) del router e forzare i dispositivi a connettersi ad esso così da riuscire ad aggirare le misure di sicurezza come il protocollo TLS, oppure si potrebbe caricare un certificato fasullo all'interno dei dispositivi vittima in modo che la macchina attaccante risulti attendibile e quindi le vittime si fidino ad inviarle i messaggi. Tali metodi però sfruttano le disattenzioni o le debolezze dell'utente più che quelle del dispositivo smart e quindi non verranno trattati.

[17] cfr. https://github.com/moxie0/sslstrip

2.5.6 Attacco replay

Nel caso in cui si voglia manipolare le informazioni che si scambiano due dispositivi (per esempio nel caso di MITM), ma non si riesca a decifrare le informazioni criptate, si potrebbe comunque tentare di inviare nuovamente le informazioni catturate, senza decriptarle, per comunicare con uno dei due dispositivi illudendolo di essere quello con cui stava comunicando in precedenza.

Un attacco replay consiste proprio in questo, ovvero nel catturare delle credenziali, o dei messaggi, che vengono scambiati in una connessione tra due sistemi e successivamente, anche quando la connessione legittima è stata già chiusa, in maniera asincrona quindi, inviarli alla vittima cercando di instaurare una connessione per carpire informazioni o farle eseguire comandi. La differenza principale tra un attacco replay e uno MITM è appunto il fatto che il primo sia riproducibile anche se la connessione è stata chiusa, mentre il secondo sia eseguibile solamente in modo live, ovvero se, e solo se, i due dispositivi stanno comunicando in quel momento, altrimenti non si è in grado di intercettare nulla proprio perché non c'è nulla da intercettare.

Nel caso della presa in esame questo tentativo viene usato per catturare i messaggi contenenti i comandi di accensione e spegnimento, messaggi che poi potranno essere inviati a piacere e che permetteranno, di fatto, di assumere il controllo della presa accendendola o spegnendola quando si vuole.

Per la prima parte di questo attacco, ovvero la cattura delle informazioni che poi andranno ritrasmesse, lo si è fatto restando connessi alla rete Wi-Fi, mentre per la seconda parte, considerato proprio che la particolarità di questo tentativo è quella di poter essere eseguito anche a connessione chiusa, non è necessario essere connessi alla stessa rete della presa (l'utilità degli oggetti smart consta infatti nella possibilità di essere controllati anche se l'utente in quel momento non è a casa e quindi non ha accesso al Wi-Fi casalingo).

Per la cattura dei pacchetti di informazione si è usato il comando sottostante:

Figura 2.4 Comando per catturare i pacchetti trasmessi e scriverli all'interno di samples.pcap

Dove “tcpdump" è il comando che intercetta e cattura i pacchetti sulla rete (che in questo caso è wlan0), “w” indica che i pacchetti catturati andranno scritti in un file e “samples.pcap” è il nome del file su cui sono stati trascritti i pacchetti.

Una volta arrivati a questo punto è stato utilizzato il tool tcpreplay[18], già preinstallato in Kali Linux, strumento open source per l’editing e la ritrasmissione di parti di traffico di rete precedentemente catturate. Per avviare l’attacco è necessario impartire il seguente comando:

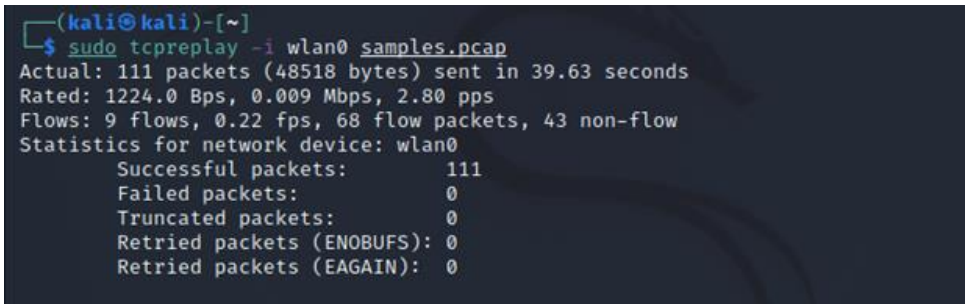

Figura 2.5 Comando per inviare i pacchetti catturati al dispositivo vittima e risultati prodotti

In questo caso “wlan0” è l’interfaccia mentre “samples.pcap” è il file contenente il traffico catturato in precedenza. Come si può osservare dall’immagine, in questo tentativo tutti i pacchetti sono stati inviati con successo, e, anche controllando in tempo reale con Wireshark, sono arrivati correttamente alla presa, ciononostante l’attacco non ha avuto alcun successo. La presa infatti ha scartato tutti i pacchetti inviati dall’attaccante e non ha risposto ad alcuno dei comandi.

Questo significa che la smart plug è in grado di distinguere tra richieste legittime e non. Se così non fosse il rischio che si avrebbe sarebbe quello di avere una presa in balia di un hacker, che, in qualunque momento può inviarle i pacchetti catturati, i quali contengono le informazioni necessarie per farle eseguire un determinato comando, facendo sì che l’utente legittimo perda il controllo sul proprio dispositivo, anche a sua insaputa, visto che, proprio per il modo in cui è stato pensato l’attacco, non è necessario che l’utente legittimo sia connesso, ma solo che la presa sia connessa.

Ci sono diverse contromisure per far sì che un dispositivo sia in grado di resistere ad un attacco replay, una di queste è l’utilizzo di chiavi di sessione generate pseudocasualmente che

[18] cfr. https://tcpreplay.appneta.com/

verranno cambiate ad ogni nuova connessione, così facendo anche se si volesse inviare un messaggio catturato ad una connessione precedente esso verrebbe scartato poiché utilizza una chiave crittografica errata.

Un'altra contromisura consiste nell'utilizzo di timestamps, ovvero una sequenza di caratteri rappresentanti un orario che certificano l'avvenimento di un fatto in un determinato attimo temporale. Grazie a queste sequenze, inserite nel corpo delle richieste il mittente è in grado di capire quali messaggi sono legittimi e quali vanno ignorati perché fasulli e possibilmente dannosi.

2.5.7 Password cracking

Quando un hacker attacca un sistema informatico lo fa molte volte per ottenere informazioni riservate, che poi riutilizza per i propri scopi, tra queste informazioni riservate le più ambite sono quelle riguardanti le credenziali di uno o più utenti, ovvero il nome e la password.

Ci sono diversi modi per ottenere tali informazioni, per esempio utilizzando un approccio MITM, ma, nel caso in cui non funzioni, si può tentare il cosiddetto password cracking. Tale procedimento, che consiste nel recupero di password tramite informazioni ottenute da un sistema informatico, può avvenire in diversi modi. Gli approcci maggiormente utilizzati sono: il metodo forza bruta (brute force) o l'utilizzo di dizionari. Ognuno di questi metodi ha i suoi pro e i suoi contro, il tipo di metodo utilizzato varia a seconda delle informazioni possedute e del tempo e risorse informatiche di cui dispone l'attaccante.

Il metodo forza bruta consiste nel tentare ogni singola combinazione possibile di caratteri finché non viene trovata la password corretta, tale tentativo ha il vantaggio di funzionare sempre, non importa che tipo di password sia. L'enorme punto dolente però consiste nell'enormità di tempo richiesto per trovare la password corretta, ciò è ovviamente dovuto al numero astronomico di combinazioni possibili di caratteri che la compongono.

Per ovviare a questo problema, tenendo anche conto della psicologia umana, infatti la maggior parte delle persone utilizza parole di senso compiuto come password e magari, se è richiesto, inserendo la prima lettera maiuscola e un numero alla fine, viene molto utilizzato un altro tentativo, quello di avvalersi di dizionari contenenti migliaia di password, che sono ritenute più probabili. Con questo sistema si riduce moltissimo il tempo necessario, ma non si ha più la certezza che la password verrà trovata, per aumentare la percentuale di successo sono stati

sviluppati softwares che, mentre testano se una parola può essere o meno la password, tentano varianti della stessa parola, magari sostituendo le vocali con dei numeri o con altri simboli (ad esempio al posto della lettera a utilizzano il numero 4). Il risultato ottenuto è un risparmio enorme di tempo a discapito però della certezza assoluta di successo che si aveva con il metodo brute force.

Nel caso dell'applicazione per gestire la presa è necessario inserire la mail associata all'account e la password. Se si tenta di creare un nuovo account, reinserendo una mail già utilizzata appare un messaggio di errore che notifica come quella specifica mail sia già in uso e quindi è necessario cambiarla, questo perché la mail deve essere unica per ogni account. Tramite questa semplice notifica però un malintenzionato riesce a scoprire quali mail sono già in uso, e quindi quali utenti attaccare. Dopo aver scoperto la mail è il momento della password, provando ad inserire password casuali, dopo cinque tentativi errati lo schermo dell'applicazione chiede di inserire la mail per la verifica in due passaggi, ovvero dopo aver inviato la mail verrà chiesto di inserire il codice di sei cifre che è stato appena inviato alla mail selezionata. È molto importante evidenziare però che chiudendo e riaprendo l'applicazione è immediatamente possibile ritentare altre cinque volte l'inserimento della password, prima che si venga bloccati di nuovo. In questo secondo caso però viene richiesto il completamento di un test CAPTCHA che ha lo scopo di impedire attacchi brute force e a dizionario automatizzati.

Come appena detto sopra, un attacco online, ovvero un attacco dove è necessario essere connessi alla rete, per poter accedere all'applicazione ed inserire le credenziali, non è attuabile, ma è possibile tentare un attacco offline, cioè, dopo essere riusciti a catturare la sequenza criptata della password, si utilizza un software. Un esempio molto famoso è John the Ripper[19] (software open source per il password cracking), che attraverso dizionari, tabelle arcobaleno (rainbow tables), brute force o una loro combinazione ricava il corrispettivo criptato di ogni password tentata e lo confronta con quello catturato, se coincidono la password corretta è stata trovata (è detto offline perché non è necessario essere connessi alla rete per il cracking effettivo della password).

Questo procedimento è necessario visto il modo in cui sono crittografate le password. Infatti le password, attraverso funzioni di hash, vengono trasformate in sequenze di caratteri della stessa lunghezza, ma la particolarità di queste funzioni è di non essere invertibili, ovvero una volta ottenuto l'hash di una parola non è possibile decriptarlo per risalire alla parola originale, l'unica

[19] cfr. https://www.kali.org/tools/john/

cosa possibile è calcolare, con lo stesso standard crittografico, l'hash di una parola e confrontarlo con quello originale, visto che parole uguali hanno hash uguali.

Per catturare l'hash della password esistono vari metodi: un hacker potrebbe tentare con un attacco MITM, ma ciò non ha successo in questo specifico caso, poiché non si è in grado di ricavare un hash leggibile intercettando le comunicazioni da e verso la presa, si è verificato infatti che viene utilizzato il protocollo TLS per rendere le informazioni indecifrabili (fare riferimento alla sezione 2.5.5).

Esistono anche altri metodi per ottenere l'hash, come per esempio attaccare, con successo, il database in cui sono contenute, o violare il sistema di un utente che ha l'autorità per accedervi. Come si può intuire tuttavia tali tentativi sono illegali e pertanto non verranno presi in considerazione. Un'ultima precisazione da fare sugli hash è che esistono metodi per impedire, o quantomeno rallentare, gli attacchi che si basano su di essi per individuare le password, uno di questi consiste nell'aggiungere il sale (salt) alla password, questo sale consiste in una sequenza casuale di bit che viene apposta alla fine della password, per ottenere un hash che è molto più difficile, se non impossibile da forzare, data la natura casuale di una propria parte.

Nel caso qui trattato, se si riuscisse a catturare e poi decifrare la password, si otterrebbe l'accesso all'applicazione e quindi essere in grado di controllare, senza alcun problema, non solamente la smart plug ma anche ogni altro dispositivo memorizzato nell'applicazione, e si avrebbe inoltre il controllo sull'account hackerato e, possibilmente, su ogni altro account che utilizza le stesse credenziali.

2.6 Risultati prodotti e considerazioni ulteriori

I vari attacchi presi in considerazione hanno prodotto vari risultati, che sono riassunti brevemente nella tabella 2.8 seguente.

Tipologia di attacco	Esito	Considerazioni
Sfruttare il software	Il successo o il fallimento dipende dalle capacità dell'attaccante	L'attacco può avere successo o meno, ma ciò dipende dal livello di esperienza dell'hacker
Manomissione dell'hardware	Il successo o il fallimento dipende dalle capacità dell'attaccante e dalle risorse a sua disposizione	L'attacco può avere successo o meno, a seconda dell'esperienza dell'hacker. Tuttavia è stato ritenuto a basso rischio perché richiede molto tempo per essere portato a termine su un singolo dispositivo
Attacco Denial of Service	Successo	L'attacco è stato efficace e non ha bisogno di conoscenze elevate per essere portato a termine
Attacco di deautenticazione	Successo	L'attacco è stato efficace e non ha bisogno di conoscenze elevate per essere portato a termine
Attacco Man In The Middle	Fallimento	Nonostante si riescano a catturare i pacchetti scambiati, l'attacco non ha avuto successo perché le informazioni ottenute sono criptate
Attacco replay	Fallimento	L'attacco non ha avuto successo nonostante diversi tentativi
Password cracking	Il successo o il fallimento dipende dalle capacità dell'attaccante e dal tempo a sua disposizione	L'attacco può avere successo o meno, ma ciò dipende anche dal buon senso dell'utente di non scegliere una password facile da decriptare e dal tempo a disposizione dell'hacker

Tabella 2.8 Esiti e considerazioni degli attacchi effettuati

Con l'eccezione dell'attacco replay e di quello Man In The Middle, dove si sa per certo che l'esito ottenuto è negativo, gli altri attacchi sono stati, o possono potenzialmente essere, effettuati con successo.

Nel caso dell'attacco MITM, nonostante non sia avvenuto con successo, si è comunque riusciti nell'eavesdropping (si è infatti riusciti ad intercettare le informazioni scambiate dai due dispositivi). Ma tali informazioni però risultano inutili. Infatti non sono utilizzabili poiché illeggibili a causa del protocollo di sicurezza che crittografa i pacchetti scambiati.

I tentativi che hanno avuto effettivamente successo, come per esempio l'attacco Denial of Service, erano stati valutati ad alto rischio durante la classificazione delle minacce, e non hanno richiesto una grande quantità di tempo e conoscenze per essere attuati, anzi, è stato relativamente facile portarli a termine.

I tentativi che non sono stati portati a termine se effettuati da un'hacker esperto probabilmente avrebbero avuto successo, considerato che i tentativi si sono interrotti per delle mancanze da parte dell'attaccante (mancanza di componenti hardware, di conoscenze) o perché richiedeva azioni illegali e non a causa dell'effettiva sicurezza della presa. Di fatto, quindi, la smart plug risulta potenzialmente vulnerabile a questi approcci, e nonostante tali azioni avrebbero potuto permettere il controllo totale della presa bisogna però specificare che la loro difficoltà di esecuzione è senza dubbio maggiore.

2.7 Conclusioni

Se si osservano i risultati ottenuti, dopo aver effettuato la modellizzazione delle minacce e aver tentato di sfruttarle attraverso degli attacchi informatici, si può convenire che la presa sia tutt'altro che invulnerabile. Solo due tra gli attacchi proposti sono stati effettivamente bloccati, mentre gli altri o sono stati portati a termine con successo, riuscendo nel loro intento, o sono stati interrotti nel mezzo del tentativo per cause esterne non riguardanti la sicurezza del dispositivo. Come si può intuire, dal punto di vista della protezione e della difesa del dispositivo stesso questi risultati non sono incoraggianti, se visti con gli occhi del consumatore, e anzi evidenziano una carenza di protezione, che dovrebbe perlomeno suscitare un minimo di preoccupazione.

Se si considera che gli attacchi che hanno avuto successo erano stati classificati come aventi un rischio alto, e la facilità d'esecuzione, il quadro diventa ancora più grave, tenuto conto anche che il numero di possibili dispositivi compromissibili sia molto elevato per ogni singolo attacco.

I rischi aumentano considerevolmente se la smart plug viene usata in ambito commerciale, ovvero se viene usata in luoghi dove è presente una connessione Wi-Fi pubblica, dove non è necessario conoscere password o altre credenziali. Per esempio se venisse usata in un'azienda, connessa ad un Wi-Fi dove tutti hanno accesso, allora sarebbe praticamente immediato eseguire uno tra gli attacchi dove sono necessarie le credenziali della connessione, proposti nella sezione 2.5, e una volta violata la presa metterebbe a rischio l'intera rete aziendale.

Gli attacchi eseguiti erano solamente alcuni tra i più utilizzati dagli hacker, e, così come le vulnerabilità prese in esame, non comprendono la totalità delle possibili minacce. Quindi, è da sottolineare come ci siano altri aspetti del dispositivo che non sono ancora stati testati e che possono risultare vulnerabili.

Con quanto è stato scritto in questa tesi, si spera dunque di essere riusciti nell'intento di aumentare la consapevolezza riguardo i pericoli a cui sono sottoposti tutta questa categoria di dispositivi.

Bibliografia

1. H Bidgoli. “Handbook of Information Security, Key Concepts, Infrastructure, Standards, and Protocols” (Vol. 1). (2006). John Wiley and Sons
2. B. Bhushan, G. Sahoo e A. K. Rai. “Man-in-the-middle attack in wireless and computer networking — A review”. (2017). 2017 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA), Dehradun, DOI: https://doi.org/10.1109/ICACCAF.2017.8344724
3. A. N. Duc, R. Jabangwe, P. Paul, e P. Abrahamsson. “Security challenges in IoT development: a software engineering perspective”. (2017). In Proceedings of the XP2017 Scientific Workshops
4. P. Engebretson. “The basics of hacking and penetration testing: ethical hacking and penetration testing made easy”. (2013). Elsevier.
5. A. Guzman e A. Gupta. “IoT Penetration Testing Cookbook: Identify vulnerabilities and secure your smart devices”. (2017). Packt Publishing Ltd
6. D. LeBlanc e M. Howard. “Writing *Secure Code”. (*2nd edition). (2002). Microsoft Press.
7. S. Pallavi e S. Smruti R. “Internet of Things: Architectures, Protocols, and Applications”. (2017). In *Journal of Electrical and Computer Engineering*, vol. 2017. DOI: https://doi.org/10.1155/2017/9324035
8. S. Phithakkitnukoon, R. Dantu e E. A. Baatarjav.“Voip security—attacks and solutions. Information Security Journal: A Global Perspective”. (2008). In ResearchGate.
9. R. Pillay. “Learn Penetration Testing: Understand the art of penetration testing and develop your white hat hacker skills”. (2019). Packt Publishing Ltd.
10. A. Shostack. “Experiences Threat Modeling at Microsoft”. (2008). pp. 4-5 URL: http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-413/paper12.pdf
11. C. Teodoro. “Software reverse engineering education”. (2009). In San Jose State University. DOI: https://doi.org/10.31979/etd.4ppy-2cjg URL: https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=4730&context=etd_theses
12. G. Weidman. “Penetration testing: a hands-on introduction to hacking”. (2014). No Starch Press.

13. W. Xiong e R. Lagerström. “Threat Modeling: A Systematic Literature Review”. In Science Direct. (2019). pp. 53–69. DOI: https://doi.org/10.1016/j.cose.2019.03.010 URL: https://www.sciencedirect.com/science/article/pii/S0167404818307478

Sitografia

1. Red Hat. “Che cos’è l’Internet of Things (IoT)?”. URL: https://www.redhat.com/it/topics/internet-of-things/what-is-iot 20 Novembre 2021.
2. S. Sinha. “State of IoT 2021: Number of connected IoT devices growing 9% to 12.3 billion globally, cellular IoT now surpassing 2 billion”. URL: https://iot-analytics.com/number-connected-iot-devices/ 20 Novembre 2021.
3. Tecnologico_editor. “Confidenzialità, integrità e disponibilità (triade CIA)”. URL: https://tecnologico.wiki/riservatezza-integrita-e-disponibilita-triade-cia/ 20 Novembre 2021.
4. H. Poston. “What are black box, grey box, and white box penetration testing?”. URL: https://resources.infosecinstitute.com/topic/what-are-black-box-grey-box-and-white-box-penetration-testing/ 20 Novembre 2021.
5. https://www.amazon.it/amazon-smart-plug-presa-intelligente-con-connettivita-wi-fi%C2%A0compatibile-con-alexa/dp/B082YTW968
6. https://play.google.com/store/apps/details?id=com.amazon.dee.app&hl=it&gl=US
7. https://www.kali.org/
8. https://github.com/NationalSecurityAgency/ghidra/releases
9. https://www.metasploit.com/
10. https://www.wireshark.org/
11. https://www.aircrack-ng.org/
12. https://www.kali.org/tools/dsniff/
13. https://github.com/moxie0/sslstrip
14. https://tcpreplay.appneta.com/
15. https://www.kali.org/tools/john/