UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

"Sicurezza IoT"

Relatore: Prof. Mauro Migliardi

Laureando: Vittorio Cardillo

ANNO ACCADEMICO 2021 – 2022

Data di laurea 21.09.2022

Abstract

La continua diffusione dell'Internet of Things (cd IoT) [1] e l'aumento della quantità di dati personali ai quali hanno accesso tali dispositivi può rivelarsi un problema per la sicurezza degli utilizzatori e per la loro privacy.

Per la verifica della sicurezza di tali dispositivi risulta fondamentale la figura del penetration tester, che, tramite l'utilizzo di tecniche manuali ed automatiche, ricerca le vulnerabilità in grado di minare la sicurezza dei dispositivi e di causare conseguentemente problemi e rischi ai fruitori dei dispositivi medesimi e all'azienda che li produce.

In base alle necessità di un'azienda e alle informazioni da essa fornite in ordine all'oggetto dell'analisi, il penetration tester potrà avvalersi di diverse strategie operative e aiutarsi tramite l'utilizzo di molteplici software, presenti sul mercato in forma open source, gratuita o commerciale. [2]

Il penetration tester ha a disposizione numerose e differenti metodologie per eseguire l'analisi dei dispositivi, e ciascuna di queste può fornire al tester linea guida durante tutto il processo di analisi. [3]

Con questa tesi vengono presentate le diverse metodologie di analisi e gli strumenti a disposizione del penetration tester, concludendosi con un esempio di penetration test eseguito su una Smart TV con l'utilizzo di tecniche manuali, software open source e gratuiti.

Contents

1. Introduzione

I rischi per la sicurezza di privati, aziende, organizzazioni ed Enti che lavorano o utilizzano dati sensibili, del settore pubblico o meno, rendono la sicurezza uno dei problemi principali dei sistemi informatici.

Il problema della sicurezza informatica è in costante crescita a causa dell'aumento della connettività dei dispositivi e della conseguente necessità che hanno le aziende di proteggere le risorse aziendali, i dati dei clienti e/o degli utenti finali, al fine di evitare complicazioni con le Organizzazioni governative responsabili della supervisione del settore industriale e con tutti i soggetti direttamente o indirettamente coinvolti in un progetto o nell'attività di un'azienda (cd stakeholder). [4]

Effettuare un penetration test permette di verificare la base hardware, software e umana di cui è composto un sistema.

Esso permette anche di comprendere se il team di sicurezza sta svolgendo il suo lavoro in maniera appropriata e funzionale, e pertanto è in grado di identificare le vulnerabilità di sicurezza, in modo tale che esse vengano eliminate prima che utenti non autorizzati possano sfruttarle. [4]

Eseguire questi test quindi aiuta i diversi Enti a prevenire perdite finanziarie e preservare l'immagine aziendale. Le aziende infatti si vedono costrette a spendere milioni di dollari quando il proprio sistema di sicurezza viene violato a causa dei costi di notifica, di riparazione e dalla diminuzione della produttività lavorativa.

Secondo l'ultimo rapporto sulla violazione dei dati di IBM del 2021 [5], in media il costo che devono affrontare le aziende per riprendersi da un attacco informatico alla sicurezza aziendale è di 4.24 milioni di dollari; somma che rappresenta il valore medio più alto negli ultimi diciassette anni.

È fondamentale verificare anche la sicurezza di tutti i dispositivi che sono interconnessi tra loro. Il rischio è che un continuo aumento dei dati forniti a questi dispositivi possa portare a gravi problemi, non solo di privacy ma che possano impattare anche il mondo fisico. Ad esempio nel 2020 un gruppo di ricercatori sono riusciti ad accedere ad una tesla Model X ad insaputa del proprietario sfruttando la connessione Bluetooth che il veicolo utilizza per sbloccare la macchina all'avvicinarsi del proprietario. [6]

Con questa tesi si vuole fornire una visione globale riguardo gli strumenti più utilizzati, le principali strategie e le principali metodologie a disposizione dei penetration tester, così da offrire ai medesimi un approccio sistematico e scientifico, fondamentale per portare a termine positivamente l'analisi.

La tesi offre altresì un esempio di penetration testing su una Smart TV [7], il quale vuole mostrare come sia possibile mettere in pratica le tecniche descritte nei paragrafi precedenti ad esso, mostrando oltre ai risultati ottenuti anche come sarebbe possibile replicare tali risultati nel caso si utilizzassero gli stessi strumenti.

In particolare la tesi presenta nella sezione 2 il concetto di penetration test e la sua struttura generale, mentre nella sezione 3 il concetto di Internet of Things e i rischi che si corrono con la loro incontrollata espansione; nella sezione 4 vengono descritti i tipi di attacchi informatici più comunemente eseguiti; nella sezione 5 vengono descritte le possibili strategie che un penetration tester può utilizzare per testare un dispositivo o un sistema; nella sezione 6 vengono descritti le diverse aree che possono venire analizzate dai tester; nella sezione 7 viene mostrata la tipica struttura di un penetration test, completa delle operazioni che vanno eseguite per portarlo a compimento; nella sezione 8 vengono presentati gli strumenti più popolari che possono supportare il lavoro del tester; nella sezione 9 vengono presentate le metodologie più popolari che possono guidare i penetration tester durante l'analisi di un sistema, fornendo una linea guida a tale operazione; nella sezione 10 viene eseguito un penetration test su una Smart TV e nella sezione 11 viene fornita una conclusione al lavoro presentato.

2. Cos’è il Penetration Testing

Il penetration testing, anche detto test di penetrazione, consiste in una serie di attività intraprese per identificare le vulnerabilità di un sistema o di una rete (cd network), permettendo così di verificare l’efficacia o l’inefficacia delle misure di sicurezza che sono state implementate.

Solitamente la sicurezza informatica di un Ente (pubblico o privato che sia) viene garantita da alcuni meccanismi di protezione che vanno dalla fase di prevenzione a quella del rilevamento della minaccia, fino a quella della risposta, o reazione alla minaccia rilevata.

La fase di prevenzione consiste nel cercare di impedire l’accesso di un malintenzionato alle risorse del sistema;

la fase del rilevamento consiste invece nel riuscire ad identificare la presenza di un malintenzionato che sia riuscito ad accedere nel sistema;

la fase della risposta, invece, consiste nell’adottare quei meccanismi di difesa e reazione a fronte del fallimento delle misure di protezione attivate nelle due fasi precedenti al fine di fermare e/o prevenire futuri danni o l’accesso alle risorse.

Ebbene quando si vuole analizzare il sistema di sicurezza di un Ente (rappresentato dalle tre fasi appena descritte) si rende necessario eseguire appunto un penetration test, che può venire effettuato simulando un accesso di un utente non autorizzato, il quale attacca il sistema utilizzando strumenti automatici, metodi manuali o una combinazione di entrambe le tecniche.

Il penetration test prevede quindi un’analisi del sistema, al fine di riconoscere qualsiasi potenziale vulnerabilità, ivi compresa la sussistenza di una configurazione del sistema scadente o impropria e la presenza di difetti hardware e software. [8]

Un penetration test determina la difficoltà che dovrà affrontare un malintenzionato, che intenda oltrepassare i controlli di sicurezza di un Ente, effettuando un accesso non autorizzato alle relative informazioni e ai relativi sistemi informativi.

3. Cos'è l'Internet of Things [9]

Per Internet of Things ci si riferisce a tutte quelle tecnologie in grado di raccogliere, trasferire e gestire informazioni digitali tramite reti wireless, senza necessità di intervento umano.

L’obbiettivo di questa tecnologia è espandere la connessione ad internet a più dispositivi possibili, ad esempio telefoni, telecamere, televisori, lavatrici ecc. in modo tale che sia possibile usufruirne in maniera più semplice, efficace e anche a distanza. Questo tipo di tecnologia è in espansione in diversi settori, dalla sanità, all’industria automobilistica, ai sistemi embedded.

Il funzionamento di un sistema IoT comprende apparecchiature digitali che comunicano grazie ad un server, ottenendo un proprio indirizzo IP all’interno della rete. Alcuni dispositivi IoT sono stati realizzati per non essere accessibili solo all’interno di una rete privata ma anche attraverso internet.

A questa enorme espansione dell’utilizzo di dispositivi connessi alla rete non corrisponde un altrettanto considerevole investimento nella sicurezza. Il problema non appartiene solamente a dispositivi progettati da zero, ma spesso vengono dotati di connessione ad internet dispositivi che sono stati progettati inizialmente per rimanere isolati, e non utilizzati a distanza.

4. Tipi di attacchi comuni

Solitamente gli attacchi ad un sistema di sicurezza sono diretti a leggere, danneggiare o rubare dati e possono venir classificati come segue [3]:

1) Denial of Service (DoS)
2) Remote to User (R2L)
3) User to Root (U2R)
4) Probing

- **Denial of Service (DoS)**, che si verifica quanto un utente malintenzionato cerca di appesantire e conseguentemente rallentare le risorse di elaborazione di un sistema, al fine di rendere ingestibili le richieste legittime;
- **Remote to User (R2L)**, che si verifica quanto un utente malintenzionato, che non dispone di un account per accedere ad un sistema remoto, sfrutta alcune vulnerabilità si quel sistema per ottenerne l'accesso locale come utente;
- **User to Root (U2R)**, che si verifica quando un malintenzionato accede ad un sistema utilizzando un normale account utente ed è in grado di sfruttare le vulnerabilità del sistema medesimo per ottenere l'accesso ad esso come amministratore (cd root).

- **Probing**, che si verifica quando un utente malintenzionato esegue la scansione di una rete di computer per raccogliere informazioni o trovare vulnerabilità della stessa.

5. Strategie di Penetration Testing

In base alla quantità di informazioni fornite al penetration tester, possiamo distinguere tre diverse strategie di penetration testing [8]:

1) Black Box
2) White Box
3) Gray Box

5.1 Black Box

Quando mancano informazioni in ordine alla struttura informatica aziendale, i dispositivi e le tecnologie adottate. In questo caso, i tester sono costretti a cercare da zero delle falle nel sistema, agendo simulando le azioni e procedure di attacchi reali.

5.2 White Box

Quando, contrariamente al black box, ai tester vengono fornite a priori tutte le informazioni necessarie in ordine alla struttura informatica aziendale, i dispositivi e le tecnologie adottate. In questo caso, il team di tester e l'organizzazione lavorano insieme ai test. Questo approccio è quello che fornisce i migliori risultati, dato che ad una maggiore quantità di informazioni nelle mani dei tester corrisponde una maggiore qualità dell'analisi. In termini di costo, questo tipo di test risulta essere più dispendioso, ma fornisce un analisi più accurata e in profondità.

5.3 Gray Box

Trattasi di una combinazione dei due casi precedenti. In questo caso, i tester devono recuperare solo parzialmente le informazioni necessarie prima di poter condurre il test. La maggior parte delle volte al tester sarà fornita una quantità di informazioni maggiore delle conoscenze

possedute dai vari individui dell'azienda, ad esempio possono venire fornite oltre ad informazioni sul sistema anche delle credenziali di accesso.

In base agli obbiettivi che si intendono raggiungere, ci sono due strategie di penetration testing [8]:

1) External Testing
2) Internal Testing

5.4 External Testing

L'obbiettivo del tester in questo caso è quello di capire se un attaccante esterno all'azienda possa penetrare all'interno della stessa ed eventualmente quanto possa spingersi avanti una volta ottenuto l'accesso.

5.5 Internal Testing

In questo caso l'obbiettivo del tester è quello di stimare quanti danni possa arrecare un dipendente scontento. Il test interno è incentrato sulla comprensione di cosa potrebbe accadere se il sistema aziendale venisse violato da un utente interno autorizzato con privilegi di accesso standard.

6. Aree di Penetration Testing

Il penetration test può essere eseguito su tre aree di intervento a seconda dell'obiettivo che ci si pone, esse sono [3]:

1) Sulla rete o network
2) Sul software o applicazione
3) Sulla componente umana dell'azienda (cd ingegneria sociale)

Queste tre aree definiscono l'obbiettivo e il tipo di penetration testing che si intende svolgere:

6.1 Test sulla rete o network

Il test di penetrazione della rete o network è un modo etico e sicuro per individuare lacune o difetti nel sistema di sicurezza, intervenendo sulla progettazione, implementazione o funzionamento della rete aziendale. I tester eseguono analisi ed esperimenti per valutare se modem e ogni altro dispositivo di accesso remoto possono essere utilizzati per violare il sistema di sicurezza.

6.2 Test sul software o applicazione

Il penetration test eseguito sull'applicazione consiste in una simulazione di attacco intesa a verificare l'efficacia del sistema di sicurezza di un'applicazione o di un dispositivo, mettendo in luce i rischi derivanti da vulnerabilità effettivamente sfruttabili da soggetti malintenzionati. Sebbene le aziende utilizzino firewall e sistemi di monitoraggio per proteggere le informazioni aziendali, la sicurezza può comunque essere compromessa poiché il traffico potrebbe comunque passare attraverso il firewall.

6.3 Test sulla componente umana o ingegneria sociale

L'ingegneria sociale sfrutta la componente umana di una azienda per ottenere informazioni sui sistemi informatici aziendali od ottenere l'accesso ad essi. Questo test viene utilizzato per determinare il livello di consapevolezza dei dipendenti di un'azienda proprietaria del sistema analizzato. Pertanto, questo è un test incentrato sul flusso di lavoro dell'organizzazione.

7. Come condurre un Penetration Test

Per svolgere un penetration test va usato un approccio sistematico e scientifico, per poi documentare il test e creare dei report rivolti a diversi livelli di gestione all'interno dell'azienda.

Generalmente, un penetration test si divide in tre fasi [10]:

1) Test preparation

2) Test
3) Test analysis

7.1 Test preparation

Prevede la preparazione di tutti i documenti necessari al test; i tester e l'azienda devono decidere l'ambito, gli obbiettivi, le tempistiche e la durata del test. In questa fase vengono discussi e regolamentati alcuni aspetti problematici quali la diffusione di informazioni riservate e il tempo di inattività dell'azienda; inoltre si procede alla definizione e alla firma di altri documenti legali ritenuti importanti.

Tra le informazioni da definire in via preliminare troviamo:

- Le informazioni di base relative all'azienda prima dell'esecuzione del test (black box, white box, gray box);
- L'aggressività del test, ovvero se si intende identificare le principali vulnerabilità di un sistema aziendale o se si intende analizzare tutti i possibili attacchi al sistema medesimo;
- Scopo, che viene determinato a seconda che si intenda analizzare un network, una applicazione o la componente umana di un'azienda;
- Tecniche e metodologie che verranno usate nel penetration test.

7.2 Test

Trattasi della fase principale del penetration test, durante la quale possono venire utilizzati una grande varietà di strumenti automatici o di tecniche manuali.

Questa fase prevede i seguenti punti:

- Information gathering. In questa fase il tester analizza la struttura e l'organizzazione aziendale e acquisisce tutte le informazioni necessarie alla fase successiva. Più precisamente il tester cerca di identificare i servizi e i protocolli di un'azienda, nonché i dispositivi e le applicazioni o software utilizzati dall'azienda medesima.
- Vulnerability analysis. L'obbiettivo di questa fase è quello di analizzare le vulnerabilità del network e delle applicazioni, sulla base delle informazioni che sono state raccolte dal tester o fornite dall'azienda.

- Vulnerability exploits. Questa fase consente al tester di determinare come sfruttare le vulnerabilità rilevate nei passaggi precedenti. Quando a seguito di tale procedura non si giunge ad una soluzione soddisfacente, si riparte dalla fase precedente.

7.3 Test analysis

In questa fase vengono analizzati i risultati delle fasi precedenti e viene altresì elaborato un piano di mitigazione, ovvero un piano atto a correggere e mitigare le falle del sistema di sicurezza aziendale. Viene quindi elaborato un report da consegnare all'azienda e pertanto esso deve essere il più completo e sistematico possibile.

La preparazione di un piano di mitigazione rappresenta la fase finale del penetration test.

8. Strumenti principali utilizzati in un penetration test

Come già affermato, nella fase di test si possono utilizzare strumenti automatici utili per l'analisi e l'identificazione delle vulnerabilità del sistema informatico di sicurezza aziendale.

Tra gli strumenti automatici più popolari troviamo i seguenti:

1) Wireshark [open source]
2) Metasploit [free + commercial]
3) Nessus [free + commercial]
4) Nmap [free]
5) W3af [open source]
6) Zed Attack Proxy (ZAP) [open source]
7) Acunetix WS [commercial]

8.1 Wireshark [11]

Trattasi di un software open source e multi piattaforma che analizza il traffico (e la relativa frequenza) delle informazioni che attraversano il network aziendale. Si tratta quindi di un programma capace di intercettare il traffico all'interno del network, convertendo il linguaggio

di trasmissione delle informazioni da un formato binario ad un formato comprensibile dall'uomo. [12]

Questo software è popolare [2] in quanto è in grado di fornire dettagli in ordine al protocollo di una rete e alle informazioni sui pacchetti intercettati; inoltre è in grado di eseguire la decriptazione dei pacchetti intercettati e di filtrare e di intercettare solamente il traffico a cui si è interessati. [11]

Può essere utilizzato sia tramite linea di comando che tramite un'interfaccia grafica.

8.2 Metasploit [13]

È uno dei framework più popolari [2] utilizzabili per i penetration test. È stato fondato con il concetto di "exploit"[1], ovvero il codice che può sorpassare le misure di sicurezza ed entrare in un certo sistema. Una volta entrato questo script esegue un "payload", ovvero delle operazioni nella macchina oggetto di penetration test. Esso può essere utilizzato in applicazioni web, networks, servers ecc. È costituito sia da una interfaccia da linea di comando, che da una interfaccia grafica funzionante su Linux, Apple Mac OS X e Microsoft Windows. [14]

8.3 Nessus [15]

Trattasi del più popolare scanner di vulnerabilità, ovvero un programma progettato per accedere ad un computer, una rete o un applicazione per scoprirne le debolezze. È costituito da più di 169.000 plugins. Tra le attività che può svolgere troviamo la ricerca di

- Vulnerabilità che possono portare ad accessi non autorizzati a dati sensibili;
- Configurazioni errate, tra le quali la mancanza di patches e aggiornamenti;
- Password di default e di alcune password base;
- Vulnerabilità ad attacchi DoS.

Inoltre può servirsi del software esterno Hydra (software utilizzato per la ricerca di password tramite "brute force") per poter svolgere un dictionary attack.

Si ritiene che tale software abbia il minor numero di falsi positivi rispetto alla concorrenza.

[1] Un exploit identifica una tipologia di script, che sfrutta un bug o una vulnerabilità per creare comportamenti non previsti in software, hardware, o in sistemi elettronici

Da notare che quando nel 2005 Nessus chiuse la licenza di distribuzione, nacque un fork noto come OpenVAS, ovvero un ulteriore scanner delle vulnerabilità con licenza open source.

Lo scanner è costituito da un client, ovvero il programma che viene utilizzato dall'utente e un server, ovvero il programma in remoto che effettua le varie analisi e invia successivamente i risultati al client.

A seguito dell'installazione sarà possibile accedere al client tramite interfaccia web collegandosi alla porta selezionata per svolgere le analisi desiderate.

Le vulnerabilità trovate verranno distinte in quattro livelli: critical, high, medium e low; Inoltre vengono fornite numerose spiegazioni riguardanti le vulnerabilità e le informazioni trovate, tra le quali possibili soluzioni.

8.4 Nmap [16]

Trattasi di un software gratuito creato per effettuare il port scanning, cioè l'individuazione di porte aperte su una rete, inviando pacchetti e analizzandone la risposta. Tra le features troviamo la possibilità di

- Identificare l'host in una rete;
- Fare una lista delle porte aperte in un dato host;
- Determinare il nome di un servizio in una rete e il suo numero di versione;
- Determinare il sistema operativo e le caratteristiche hardware di un dispositivo di rete;
- Utilizzare interazioni scriptabili con il target.

Questo software viene quindi principalmente utilizzato nelle fasi di information gathering, permettendo all'utilizzatore di conoscere la struttura della rete che sta analizzando. Utilizzando questo software sarà possibile selezionare diverse modalità di analisi, in base ad aggressività e risultati cercati. Lo stato delle porte analizzate viene classificato con sei valori differenti: open, closed, filtered, unfiltered, open|filtered e closed|filtered. Gli ultimi due stati indicano una non esatta definizione riguardo alla condizione.

Una porta viene definita "closed" quando è accessibile ma non possiede alcuna applicazione collegata ad essa.

Una porta viene definita “filtered” nel caso Nmap non fosse capace di determinare se essa è chiusa o meno perché i pacchetti inviati non sono giunti a destinazione a causa di un filtro. Tale filtraggio potrebbe venire effettuato da un firewall, o da regole assegnate ad un router.

Una porta viene definita “unfiltered” se è accessibile, ma Nmap non è in grado di determinare se è aperta o chiusa.

8.5 W3af [17]

È un software open source progettato per controllare la sicurezza delle applicazioni web. Fornisce uno scanner delle vulnerabilità e uno strumento per l’exploitation. È accessibile sia tramite linea di comando, che tramite interfaccia grafica. È importante osservare che le sue funzionalità possono venire estese tramite numerosi plugins.

8.6 Zed Attack Proxy (ZAP) [18]

Trattasi di un security scanner per applicazioni web. È uno dei progetti dell’Open Web Application Security Project (OWASP) più attivi. Tra le varie funzioni esso può:

- Fungere da server proxy, manipolando e controllando così il traffico che passa attraverso di esso;
- Effettuare scansioni automatiche;
- Utilizzare una serie di strumenti che permettono di individuare manualmente le vulnerabilità.

8.7 Acunetix WS [19]

È un software utile per scansionare automaticamente le vulnerabilità web e fornire un report su 7000 possibili vulnerabilità, incluso OWASP Top 10 e tutte le varianti di SQL Injection e XSS. Assiste il penetration tester svolgendo automaticamente mansioni che richiederebbero ore di lavoro manuale, fornendo risultati accurati senza falsi positivi. Supporta HTML5, JavaScript, Single-page application e sistemi CMS.

Esso include anche strumenti manuali avanzati. Tra le varie funzionalità vi è quella di mostrare le righe di codice (cd state line) che presentano problemi e vanno corrette, senza quindi bisogno

di cercarle; fornisce inoltre ai tester tutte le informazioni necessarie per correggere i problemi e permette anche di effettuare scansioni programmate automatiche.

9. Modelli e metodologie di Penetration Test

Per l'esecuzione di un penetration test si possono utilizzate diverse metodologie, tra le principali troviamo le seguenti:

9.1 OSSTMM [20]

È una metodologia standard per i test sulla sicurezza, sviluppata dall'Institute for Security and Open Methodologies (ISECOM). Viene utilizzata soprattutto per attività di Network Penetration Test, nonostante l'ambito di applicazione non si limiti a questo.

Questa metodologia può essere applicata a tutti i tipi di test di sicurezza. Qualunque sia il target del test (un sistema, una location, una persona, un processo, ecc.) questa metodologia ne garantisce un esecuzione il più efficiente possibile.

Tale metodologia detta linee guida per tre tipi di verifica della sicurezza:

- COMSEC (canale di sicurezza delle comunicazioni)
- PHYSSEC (canale di sicurezza fisica)
- SPECSEC (canale di sicurezza dello spettro)

Questi tipi o classi di verifica sono suddivise a loro volta in "canali" a seconda che abbiano ad oggetto la componente umana o fisica di una azienda, le reti wireless, di telecomunicazione e dati.

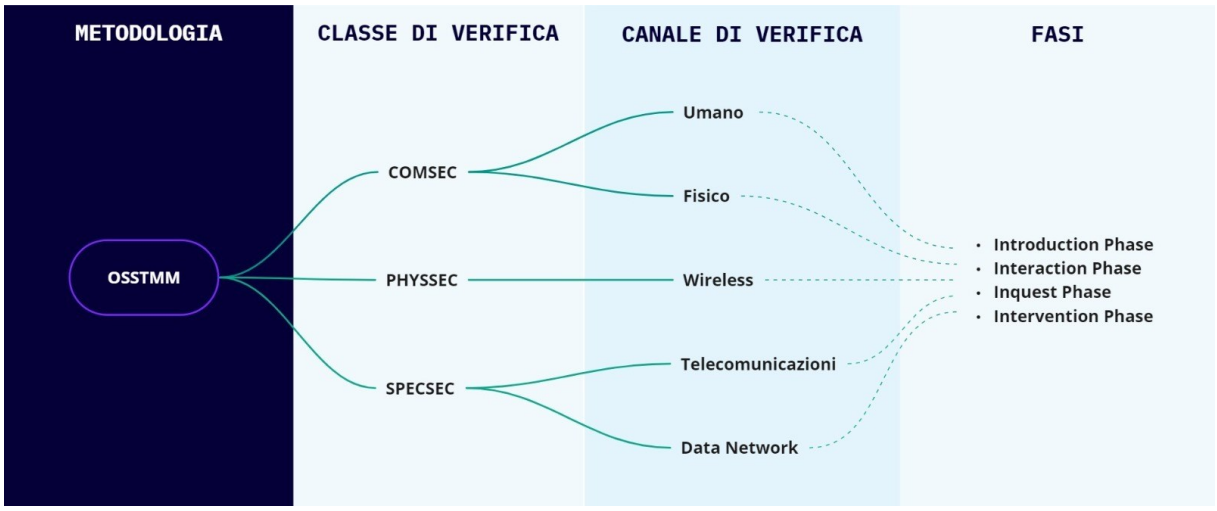

Figura 1: Struttura della metodologia OSSTMM

Non ci sono strumenti indicati per il processo di test, ma solo informazioni sulle attività da eseguire per ciascun canale, dando così molta libertà al tester.

Infine, lo svolgimento del test si conclude con il Security Test Audit Report (STAR), che contiene i dati ottenuti durante le attività.

Classe	Canale	Descrizione
PHYSSEC	**Umano**	Trattasi del test della componente umana di un'azienda, con la quale vi può essere una interazione fisica o psicologica. Lo scopo di questo tipo di testing è quello di valutare la sicurezza delle interazioni tra le persone e queste ultime con la tecnologia (per esempio rispetto delle policy aziendali o il problema del phishing).
	Fisico	È il testing della componente tangibile dell'azienda, con ciò si intende la parte strutturale dell'azienda medesima, ovvero le cose (per esempio porte ed altre vie di ingresso) ed i luoghi aziendali (si pensi alla verifica di accessi non autorizzati ai locali, ai keylogger, alle spie ed alle intercettazioni ambientali).
SPECSEC	**Wireless**	È il testing di sicurezza relativo alle comunicazioni senza fili come i segnali radio (per esempio la ricerca di reti aperte, reti non autorizzate, password deboli, errori di architettura e di configurazione)

Classe	Canale	Descrizione
COMSEC	**Telecomunicazioni**	È un test che viene eseguito su tutte le reti di telecomunicazione, digitali o analogiche, in cui l'interazione avviene su linee telefoniche stabilite o linee di rete simili a quelle telefoniche. Alcuni esempi delle verifiche da effettuare sono la verifica della sicurezza degli switch, delle regole di firewall, della crittografia e dell'autenticazione.
	Data Networks	È il testing sulla sicurezza delle reti. Si tratta della valutazione della sicurezza dei sistemi elettronici che si occupano di distribuire o smistare dati. Questo testing include tutti i sistemi elettronici e le reti di dati in cui l'interazione avviene su un cavo stabilito e linee di rete cablata (per esempio test sulla configurazione, aggiornamento e vulnerabilità del codice).

Applicati a questi cinque canali ci sono diciassette moduli, i quali differiscono tra loro per le task da svolgere. Il risultato di uno dei moduli può servire come punto di partenza per uno o più moduli successivi, rendendo fondamentale l'esecuzione di tutte le task. Il non riuscire a concludere un modulo potrebbe causare un effetto a catena che ci impedisce di portare a termine i moduli successivi. Questo significa che nessuna fase è meno importante di altre.

Questi diciassette moduli si spartiscono tra quattro fasi ovvero:

1) Introduction Phase
2) Interaction Phase
3) Inquest Phase
4) Intervention Phase

- **Introduction Phase**

L'obbiettivo di questa fase è la comprensione dei requisiti, dello scopo e dei vincoli dell'analisi. La decisione del tipo di test da eseguire viene presa in seguito alla conclusione di questa fase.

- **Interaction Phase**

Questa fase ha lo scopo di definire l'ambito del penetration test.

- **Inquest Phase**

Fase in cui vengono valutate le diverse informazioni che il tester scopre, al fine di comprendere se le stesse possano essere mal riposte o mal gestite.

- **Intervention Phase**

In questa fase vengono individuati i test pratici da eseguire a seconda delle informazioni precedentemente raccolte. I test avranno ad oggetto le varie risorse del target (ovvero dell'oggetto di indagine), che possono essere modificate e/o sovraccaricate per fornire ai tester una possibilità di penetrazione o interruzione del sistema.

Questa è spesso la fase finale di un test di sicurezza; ciò in quanto:

- Le informazioni per effettuare questi test potrebbero non essere note fino a quando non saranno state eseguite altre fasi
- È necessario garantire che le interruzioni non influiscano sulle risposte dei test meno invasivi.

Il modulo finale di questa fase (il modulo D.17, di Alert and Log Review), è necessario per verificare i test precedenti che non hanno fornito alcuna interattività al tester.

9.2 ISSAF [21]

La metodologia ISSAF (Information Systems Security Assessment Framework) ha l'obbiettivo di valutare la sicurezza di reti, sistemi e applicazioni.

Tale metodologia si compone di tre aree (leggasi fasi) principali: quella della pianificazione e preparazione, della valutazione e reportistica, della pulizia e distruzione dei manufatti.

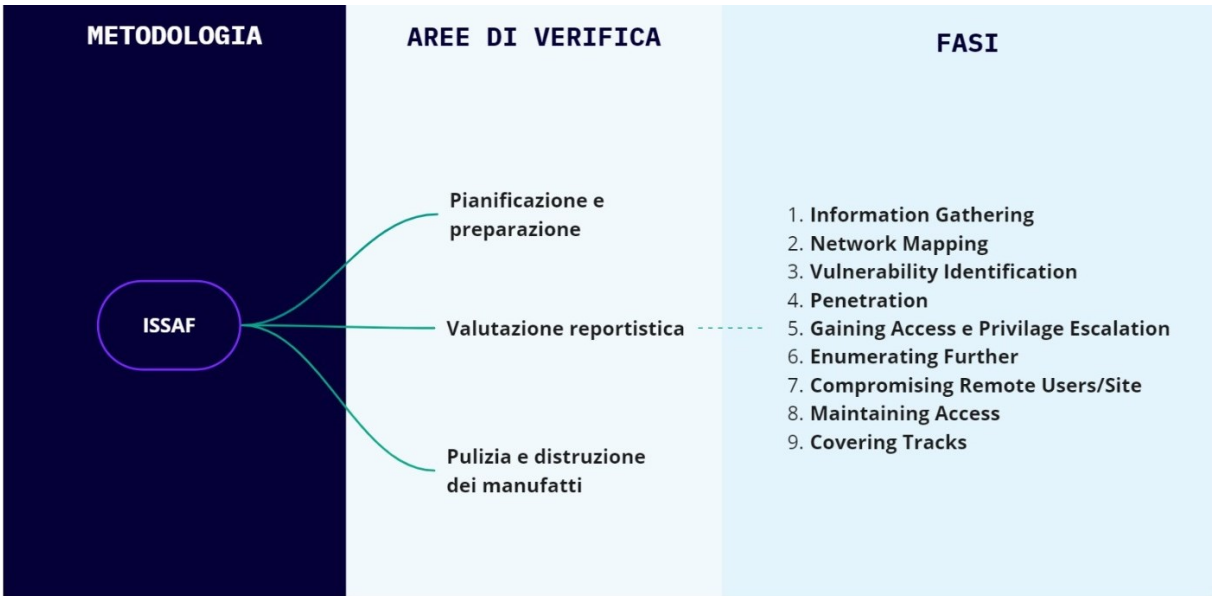

Figura 2: Struttura della metodologia ISSAF

L'area della pianificazione e preparazione riguarda i passaggi necessari per impostare l'ambiente di test, gli strumenti di test, i contratti e gli aspetti legali, la definizione del team di coinvolgimento, le scadenze, i requisiti e la struttura dei report finali.

L'area di valutazione e reportistica è il fulcro della metodologia, in cui vengono eseguiti i test di sicurezza. Questa fase è costituita da nove attività principali:

1) **Information Gathering**: ricerca di informazioni importanti riguardo il target;
2) **Network Mapping**: identificazione di tutte le risorse del sistema internamente ad un dato network;
3) **Vulnerability Identification**: indagine delle vulnerabilità;
4) **Penetration**: ottenimento dell'accesso baipassando le misure di sicurezza;
5) **Gaining Access e Privilage Escalation**: ottenimento delle autorizzazioni amministrative;
6) **Enumerating Further**: ottenimento di nuove informazioni riguardo ai processi del sistema con l'obbiettivo di sfruttarle;
7) **Compromising Remote Users/Sites**: compromissione (attraverso l'exploit) della relazione di fiducia che esiste tra gli utenti e l'azienda;
8) **Maintaining Access**: tentativo di garantirsi un continuo accesso al sistema tramite canali nascosti (backdoors);
9) **Covering Tracks**: eliminazione di tutti i segni di compromissione, nascondendo i file, cancellando i registri, superando i controlli di integrità e neutralizzando il software antivirus.

Nella fase di report vengono proposte due modalità di report: verbale e scritta.

Il report verbale è riservato alle sole problematiche critiche e/o urgenti. Tale report dovrebbe essere limitato ai casi in cui il test di penetrazione riveli uno stato di vulnerabilità rispetto alla quale si renda necessario un intervento immediato o addirittura venga rilevata una attività illegale sulla rete e sui sistemi, che renda necessario dover contattare le autorità legali prima (o anche senza) informare il cliente.

La relazione scritta è il resoconto formale del test di penetrazione. Può essere strutturata in maniera diversa a seconda del destinatario del report medesimo e può anche includere informazioni sulle questioni già discusse nella relazione (o report) verbale.

La metodologia ISSAF non fornisce un modello di report dettagliato come la metodologia OSSTMM e si concentra sugli aspetti informatici del sistema.

L'area della pulizia e distruzione dei manufatti (Cleen-Up and Destroy) è piuttosto breve e si concentra sulla rimozione di eventuali artefatti rimasti dal penetration test. Lascia il tester libero di scegliere come crittografare, disinfettare e distruggere i dati creati durante il penetration test.

9.3 PTES [22]

Lo scopo della metodologia PTES (Penetration Testing Execution Standard) non è quello di stabilire schemi rigidi per il test di penetrazione.

Le linee guida tecniche aiutano a definire le procedure da seguire durante il Penetration test, fornendo una struttura di base per avviare e condurre il test medesimo.

La metodologia si compone di sette fasi, dette sezioni:

- **Interazioni pre-engagement**, che definiscono l'ambito del test (obiettivo, target, tipo di test, data e ora). Questa metodologia infatti prevede che il test debba cominciare con una fase di comunicazione tra penetration tester e azienda in cui vengano comunicati i dettagli dell'operazione di indagine, utili all'azienda per capire quali sono i rischi concreti in caso di un attacco informatico. Questa fase permetterà al penetration tester di potersi concentrare nella frase di exploitation e post exploitation;
- **Raccolta di informazioni**, nonché enumerazione e scansione delle informazioni del sistema di destinazione;

- **Modellazione delle minacce**, dove i vettori di attacco vengono analizzati a partire dalle informazioni ottenute nelle fasi precedenti;
- **Analisi di vulnerabilità**, che si occupa del rilevamento delle vulnerabilità del sistema analizzato (o target);
- **Exploit**, utilizzato per sfruttare le vulnerabilità rilevate;
- **Post-exploitation**, che copre le tracce create ed esegue anche exploitation aggiuntivi;
- **Reporting**, che consiste nel report finale da inviare al cliente.

Oltre ad una descrizione metodologica per lo svolgimento di un penetration test il PTES fornisce anche una guida tecnica molto dettagliata, che tratta anche l'utilizzo di alcuni specifici strumenti automatizzati per la ricerca delle vulnerabilità (tra i quali troviamo i sopra citati OpenVAS, Nessus, Wireshark, Metasploit e Nmap).

9.4 NIST [23]

La metodologia proposta dal NIST (National Institute of Standards and Technology) è stata inizialmente introdotta come GNST (Guideline on Network Security Testing). I target principali che vengono presi in considerazione da questa metodologia fanno parte del settore bancario, di quello energetico e delle comunicazioni. Il framework fornito da questa metodologia è organizzato in cinque funzioni chiave:

1) Identify
2) Protect
3) Detect
4) Respond
5) Recover

- **Identify**

Consiste nell'individuazione di tutte le risorse, i sistemi e i network dell'azienda, classificando le informazioni ricavate e permettendo così di decidere a quali elementi dare la priorità.

- **Protect**

Consiste nel salvaguardare i sistemi e i dati da accessi non autorizzati o modifiche, includendo misure di sicurezza quali firewalls, intrusion detector e software anti-virus.

- **Detect**

Consiste nell'essere in grado di scoprire attività dannose o comportamenti anomali per rispondere prontamente a potenziali minacce.

- **Respond**

Consiste nell'essere in grado di adottare le misure necessarie quando è stata compromessa la sicurezza; misure consistenti nel contenimento, salvaguardia e tutela delle informazioni aziendali.

- **Recover**

Consiste nell'essere in grado di ripristinare le normali operazioni e funzioni aziendali dopo che si è verificato una compromissione al sistema di sicurezza aziendale.

La metodologia NIST tratta più in particolare il penetration testing nelle pubblicazioni 800-42 e 800-115 (dal titolo "Technical Guide to Information Security Testing and Assessment"). Sostanzialmente il processo viene diviso in quattro fasi:

- **Planning**, dove il sistema viene analizzato al fine di determinare la priorità dei target da analizzare;
- **Discovery**, in cui il tester cerca le vulnerabilità nel sistema;
- **Attack**, in cui il tester verifica se le vulnerabilità rilevate possono essere sfruttate;
- **Report**, dove viene riportato ogni risultato delle azioni intraprese nel passaggio precedente.

Queste fasi vengono bene rappresentate nell'immagine seguente.

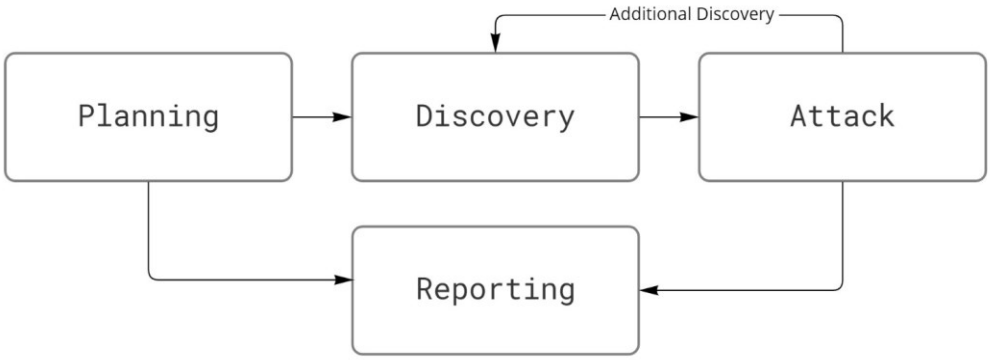

Figura 3: Processo di penetration testing proposto da NIST

Nella fase di attacco sono presenti anche le seguenti attività: ottenere l'accesso, aumentare i privilegi, navigare nel sistema e installare strumenti aggiuntivi.

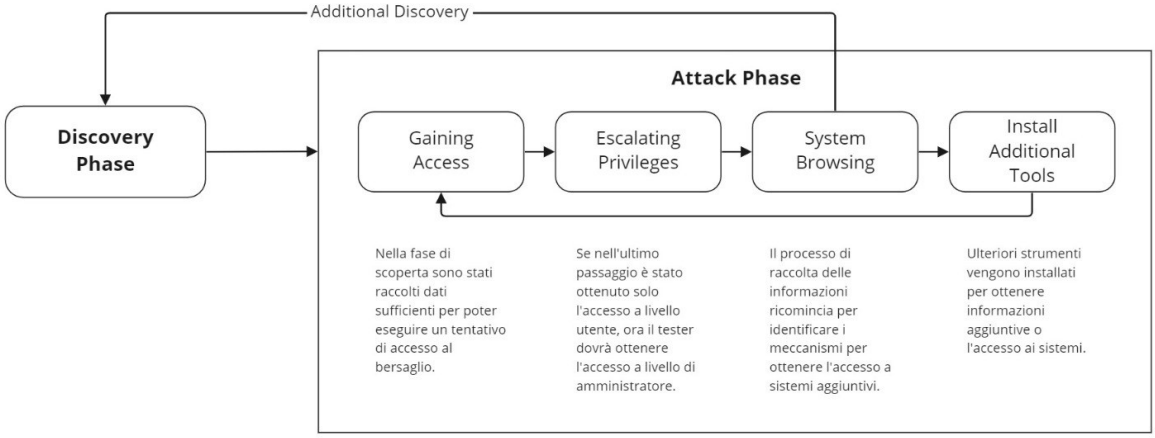

Figura 4: Fase di attacco proposta da NIST

Il testing e l'analisi di vari sistemi viene rappresentato dal loop mostrato nell'immagine sopra riportata tra la discovery phase e la attack phase.

9.5 OWASP [24]

Il progetto OWASP (Open Web Application Security Project) detta regole per il testing del sistema di sicurezza aziendale e lo sviluppo sicuro delle applicazioni web.

La maggior parte delle aziende che si occupano di sviluppo dei software, non si occupano dei problemi legati alla sicurezza nel processo di sviluppo del software.

Quindi, la metodologia considera l'uso dei test di sicurezza come mezzo di consapevolezza e si basa su altri progetti forniti dall'OWASP come Code Review Guide e Development Guide.

Questa metodologia è suddivisa in tre fasi principali:

10 La fase introduttiva, che tratta i presupposti per il test delle applicazioni web e anche l'ambito del test;
11 La fase intermedia, che presenta l'OWASP Testing Framework con le sue tecniche e attività legate alle diverse fasi del Software Development Life Cycle;
12 La fase conclusiva, che descrive come le vulnerabilità vengono testate da Code Review e Penetration Testing.

Questo standard è estremamente dettagliato, e comprende anche numerosi esempi per ogni singola fase. Esso può essere estremamente utile per aziende che necessitano di una forte presenza sul web.

In particolare, per quanto riguarda il penetration testing, le fasi proposte dello standard OWASP sono 12 e sono le seguenti:

A. Information Gathering

Trattasi del processo di raccolta di informazioni riguardo l'applicazione considerata.

B. Configuration and Deployment Management Testing

In questa fase è necessario comprendere le configurazioni del server di hosting, poiché alcuni errori di configurazione potrebbero compromettere l'applicazione.

C. Identity Management Testing

L'obbiettivo è quello di identificare e documentare i vari ruoli degli utenti previsti dall'applicazione, per i quali si cerca di ottenere l'accesso, validando inoltre il processo di registrazione.

D. Authentication Testing

Consiste nella verifica della sicurezza dei processi di autenticazione degli utenti

E. Authorization Testing

In questa fase si verifica il modo in cui i vari utenti possono interagire con l'applicazione e con quali limitazioni.

F. Session Management Testing

In questa fase si verifica il modo in cui il sito interagisce con l'utente, ad esempio tramite cookies, session variables, cross site request e logout functionality.

G. Input Validation Testing

In questa fase si verificano i vari tipi di input che l'applicazione accetta. Questa fase è fondamentale poiché tra le più comuni vulnerabilità delle applicazioni web c'è proprio il fallimento nella validazione degli input, che rende il sistema vulnerabile a diverse tecniche di attacco quali per esempio l'SQL injection o il cross site scripting (XSS).

H. Testing for Error Handling

L'obbiettivo di questa fase è l'identificazione e l'analisi dei possibili output di errore, cercando di individuarne la gestione non appropriata.

I. Testing for Weak Cryptography

L'obbiettivo di questa fase è la verifica della Criptografia che viene utilizzata nell'applicazione. Tra le varie verifiche che vengono descritte vi è la valutazione della capacità criptografica e della validità del certificato digitale.

J. Business Logic Testing

In questa fase si vuole valutare se le logiche di business implementate nell'applicazione (ovvero il nucleo di elaborazione, sotto forma di codice sorgente, che rende operativa un'applicazione) funzionano come atteso o contengono vulnerabilità.

K. Client-side Testing

Trattasi dell'esecuzione di codice nel client, tipicamente in modo nativo all'interno di un browser web o tramite un plugin del browser medesimo. Esempi di ciò sono HTML e CSS injection o Client-side URL Redirection.

L. API Testing

Questa fase comprende il Testing di GraphQL, e consiste nel validare tutti i possibili input rispetto ad attacchi generici, e verificando che tutti i relativi controlli di sicurezza vengano applicati.

9.6 OWASP IoT: ISVS [25]

Data la grande popolarità guadagnata dalla metodologia OWASP sopra descritta, la quale permetteva una accurata analisi della sicurezza di applicazioni web, ne è stata realizzata una versione anche per i dispositivi IoT dal nome Internet of Things Security Verification Standard (ISVS). Attualmente nella fase di pre-release (Pre-release 1.0RC) e nata dall'unione della

community, la quale si è dedicata alla stesura di questa metodologia, si prefigge il compito di definire vari requisiti e best practice di sicurezza che possono essere verificati in più fasi dello sviluppo, tra le quali la progettazione, lo sviluppo e il testing del dispositivo. Data la grande varietà di dispositivi IoT esistenti, è importante considerare questa metodologia come una base per sviluppare dei propri test. Inoltre vengono indicate solamente delle verifiche da effettuare, non vengono presentati metodi pratici per effettuare le verifiche, elemento presente in altre metodologie già presentate.

Il modello è suddiviso in cinque fasi:

1) IoT Ecosystem
2) User Space Application
3) Software Platform
4) Communication
5) Hardware Platform

Come è possibile osservare nell'immagine seguente, rappresentante la struttura del modello, ogni fase è suddivisa in più sezioni, ovvero ambiti diversi che sarà necessario analizzare.

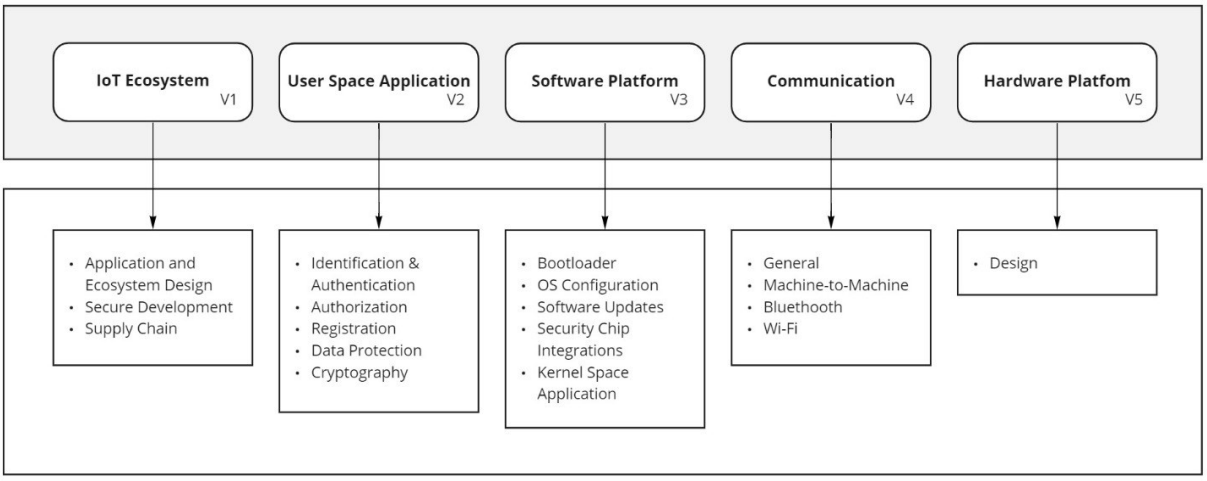

Figura 5: Struttura della metodologia ISVS

- **IoT Ecosystem (V1)**

Questa fase fornisce una serie di requisiti riguardanti l'intero ecosistema di dispositivi IoT.

La prima sezione tratta l'analisi della sicurezza nella progettazione, essa è necessaria per creare un prodotto dall'architettura sicura. Ad esempio il verificare che tutte le applicazioni di un ecosistema IoT rispettino le linee guida relative alla sicurezza delle applicazioni stesse.

La seconda sezione tratta la sicurezza del processo di sviluppo, andando ad identificare e documentare tutte le informazioni sensibili richieste al sistema. Ad esempio in caso di sviluppo in C e C++, verificare che tutte le funzioni vietare, quali memcpy e strcpy siano state sostituite con versioni più sicure come l'utilizzo della Safe String library.

La terza sezione tratta la sicurezza della catena di distribuzione, ovvero la sicurezza di quel processo fondamentale che permette di portare sul mercato un prodotto. Ad esempio la verifica che il codice di terze parti e i vari componenti vengano analizzate per assicurarsi che non siano state introdotte backdoor.

- **User Space Application (V2)**

Questa fase fornisce una serie di requisiti relativi allo spazio che interessano maggiormente l'utente, ovvero a livello dell'applicazione, andando a verificare che l'accesso al dispositivo venga effettuato in maniera sicura, andando a proteggere i dati sensibili.

La prima sezione tratta l'autenticazione e l'identificazione nel dispositivo, andando a verificare che avvengono in maniera sicura. Ad esempio bisogna verificare che tutte le password degli utenti autenticati superino la lunghezza di dodici caratteri e che siano abbastanza complesse.

La seconda sezione tratta le autorizzazioni all'interno del dispositivo, verificando che un certo utente abbia il permesso di accedere e visualizzare alcuni dati o servizi.

La terza sezione tratta la protezione dei dati sensibili come le credenziali di accesso o dati di fatturazione. Ad esempio bisogna assicurarsi che tutte le informazioni mantenute in memoria vengano sovrascritte da zero quando non vengono più ritenute necessarie.

La quarta fase tratta la crittografia, è infatti necessario selezionare i metodi giusti per crittografare i dati, ad esempio bisogna verificare che segreto e chiave crittografica siano unici pero ogni dispositivo o verificare che le librerie di crittografazione utilizzate dal dispositivo siano conformi a standard crittografici riconosciuti.

- **Software Platform (V3)**

Questa fase fornisce una serie di requisiti riguardanti la piattaforma in uso.

La prima sezione tratta il boot del dispositivo, ovvero il suo avvio. Gli sviluppatori del firmware del dispositivo sono responsabili della sua sicurezza, è quindi importante verificarne l'effettiva affidabilità. Ad esempio verificare che il firmware sia mantenuto in uno spazio crittografato.

La seconda sezione riguarda il sistema operativo utilizzato e il suo kernel, ciò è necessario perché esso svolge delle importanti funzioni primitive. Ad esempio è necessario verificare che vengano bloccati eventuali accessi non autorizzati alla RAM.

La terza sezione riguarda gli aggiornamenti dei software, ad esempio permettendo che i dispositivi possano ricevere aggiornamenti periodici ed automatici.

La quarta sezione riguarda l'utilizzo di chip di sicurezza, ovvero dei circuiti integrati che gestiscono le funzioni di scurezza, i quali devono essere forniti da venditori approvati e sicuri.

- **Communication (V4)**

Questa fase fornisce una serie di requisiti riguardanti i metodi di comunicazione che vengono forniti al dispositivo.

La prima sezione tratta delle verifiche generali a tutti i mezzi di comunicazione. Ad esempio la verifica che esse avvengano attraverso canali sicuri.

La seconda sezione tratta la comunicazione Machine-to-Machine, ovvero quella tra più dispositivi. Uno dei requisiti presentati in questa sezione consiste nel assicurarsi che le uniche trasmissioni non crittare non riguardino dati sensibili.

La terza sezione tratta la comunicazione tramite Bluetooth. Tra i vari requisiti, viene richiesto che venga utilizzata la modalità di sicurezza il più aggiornata possibile per il dispositivo in uso.

La quarta sezione tratta la comunicazione tramite Wi-Fi. Tra i vari requisiti viene richiesto che lo stato di connettività Wi-Fi sia attivato solamente quando effettivamente richiesto.

- **Hardware Platform (V5)**

Questa fase fornisce una serie di requisiti riguardanti le componenti hardware della piattaforma.

Tale fase è costituita da una sola sezione, nella quale si cerca di fornire delle linee guida per progettare un hardware il più sicuro possibile, poiché rappresenta la base dell'intero dispositivo.

Tutte le fasi sopra discusse sono costituite da vari controlli da effettuare, i quali, vengono affiancati da un livello rappresentante il tipo di dispositivi che richiedono tali verifiche.

Il modello ISVS introduce tre livelli di sicurezza in base al tipo di dispositivo che si vuole analizzare:

- **ISVS Level 1 (L1)**

Questo livello di sicurezza è costituito da dispositivi che non rappresentano un rischio alla sicurezza se compromesso, ad esempio poiché non utilizzano alcun dato personale, ad esempio una Smart TV.

- **ISVS Level 2 (L2)**

Questo livello di sicurezza è costituito da dispositivi la quale compromissione dovrebbe essere evitata, ad esempio telecamere di sicurezza o sistemi di allarme.

- **ISVS Level 3 (L3)**

Questo livello di sicurezza è costituito da dispositivi la quale compromissione è fondamentale venga evitata, sono esempio di dispositivi appartenenti a tale livello i dispositivi medici.

10. Penetration Test eseguito su un dispositivo IoT

Si è deciso di eseguire un penetration test ad un dispositivo posseduto, una Smart TV, ovvero un televisore la quale principale caratteristica consiste nell'introduzione di funzioni e servizi legati ad internet. [7]

10.1 Dispositivi utilizzati

Per l'esecuzione di tale test sono stati utilizzati i seguenti dispositivi:

- Televisore Samsung HD T4300 2020

Trattasi della televisione presa in esame, essa tra le varie funzioni permette la connessione ad un dispositivo mobile tramite l'applicazione Samsung SmartThing [26], ovvero una applicazione sviluppata dall'azienda Samsung che permette un controllo remoto degli apparati elettronici direttamente dal proprio smartphone, permettendo la connessione a dispositivi quali monitor, televisioni, elettrodomestici, termostati e altro.

Grazie a questa applicazione viene permesso agli utenti un monitoraggio costante dei dispositivi connessi e la possibilità di programmare il loro funzionamento in modo automatico.

- Telefono Samsung Galaxy S9+

Trattasi dello smartphone utilizzato per usufruire dell'applicazione SmartThings, con esso si farà sempre la vece del proprietario del televisore Samsung HD T4300 2020, esso sarà necessario nei test che richiederanno l'utilizzo di due dispositivi mobile collegati simultaneamente al televisore.

- Telefono Huawei Mate 20

Trattasi dello smartphone utilizzato per usufruire dell'applicazione SmartThings, con esso si farà sempre la vece di un ipotetico malintenzionato, intento ad ottenere il controllo del televisore Samsung HD T4300 2020.

- ASUS FX553V

Trattasi del computer utilizzato per eseguire la maggior parte dei test sotto riportati. Esso presenta come sistema operativo Windows 10. Verrà utilizzato principalmente per servirsi di software per la verifica della sicurezza.

10.2 Programmi utilizzati

Per effettuare il penetration test lo scrivente si è avvalso di alcuni software open source e commerciali:

- Metasploit [13]
- Nessus [15]
- Nmap [16]

10.3 Approccio utilizzato

Per quanto riguarda l'approccio adottato in base alle informazioni disponibili, ovvero in base alla distinzione sopra discussa [8] tra black box, gray box e white box, alcuni test effettuati possono riferirsi a tutti gli approcci. Nonostante ciò, si ritiene che il procedimento seguito possa essere maggiormente assimilabile ad un approccio gray box, poiché in alcuni casi sono stati effettuati dei test partendo dal presupposto di possedere l'accesso alle credenziali dell'account Samsung collegato all'applicazione SmartThings.

Si è inoltre deciso di effettuare il test simulando l'azione di un individuo esterno al dispositivo, ovvero il già trattato external testing [3], poiché è stato ritenuto più valido effettuare una verifica osservando quali danni potesse arrecare una persona non avente normalmente accesso ad esso o se fosse possibile ottenerne in qualche modo il controllo, ad insaputa del proprietario del dispositivo.

È possibile inoltre concludere che l'area dei seguenti test è riguardante il "software o l'applicazione", ovvero la verifica dell'efficacia del sistema di sicurezza di un'applicazione o di un dispositivo.

10.4 Information gathering

Successivamente ad una fase di test preparation, a seguito della quale è stato possibile decidere quale sarebbe stato lo scopo del test in esame, si apre la fase fondamentale del information gathering, ovvero la ricerca delle informazioni riguardante il dispositivo. Nel caso il test fosse stato eseguito in collaborazione all'azienda produttrice del televisore, durante la fase di preparazione sarebbe stato necessario definire anche delle regole di riservatezza e altri documenti legali ritenuti importanti da parte dell'azienda. La fase di ricerca delle informazioni è essenziale per lo svolgimento di un penetration test, poiché è necessario comprendere lo stato del sistema che si vuole attaccare.

La seguente fase è stata effettuata raccogliendo informazioni seguendo due modalità, in modo passivo e in modo attivo.

Per quanto riguarda la ricerca di informazioni attiva, è stato utilizzato il software gratuito Nmap per effettuare una scansione della rete alla quale era connessa la televisione. L'obbiettivo infatti è quello di conoscere quale fosse l'indirizzo del dispositivo in esame.

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-08 15:04 ora legale Europa occidentale
Nmap scan report for H388X.home (192.168.1.1)
Host is up (0.0090s latency).
MAC Address: A8:02:DB:D7:31:68 (zte)
Nmap scan report for dlinkap.home (192.168.1.2)
Host is up (0.0030s latency).
MAC Address: 90:8D:78:2A:6F:2A (D-Link International)
Nmap scan report for Host-003.home (192.168.1.11)
Host is up (0.12s latency).
MAC Address: 32:06:A9:43:E4:5B (Unknown)
Nmap scan report for Samsung.home (192.168.1.14)
Host is up (0.044s latency).
MAC Address: 80:8A:BD:53:28:93 (Samsung Electronics)
Nmap scan report for hewlett-packard.home (192.168.1.15)
Host is up (0.013s latency).
MAC Address: 34:68:95:08:B8:FB (Hon Hai Precision Ind.)
Nmap scan report for LAPTOP-5113VFS7.home (192.168.1.33)
Host is up (0.066s latency).
MAC Address: F8:5E:A0:E8:70:6F (Intel Corporate)
Nmap scan report for DESKTOP-8MLFS95.home (192.168.1.9)
Host is up.
Nmap done: 256 IP addresses (7 hosts up) scanned in 2.99 seconds
```

Figura 6: Scoperta degli host tramite il comando "nmap -sn 192.168.1.0/24

Il comando –sn permette di effettuare una scansione più rapida della rete, andando ad evitare di controllare quali sono le porte aperte per ogni dispositivo collegato, fornendoci unicamente una lista dei dispositivi attualmente collegati. Gli indirizzi IP che sono stati verificati sono stati definiti dal comando 192.168.1.0/24, permettendo quindi a Nmap di verificare tutti gli indirizzi compresi tra 192.168.1.0 e 192.168.1.255. [16]

È possibile osservare che Nmap ha rilevato 7 dispositivi connessi alla rete in questione, tra di essi è possibile individuare l'indirizzo IP della televisione "Samsung.home", ovvero 192.168.1.14.

Ottenuto l'indirizzo IP del dispositivo interessato, sarà ora possibile poter effettuare una scansione per verificare quali porte aperte possiede.

```
Nmap scan report for Samsung.home (192.168.1.14)
Host is up (0.013s latency).
Not shown: 65523 closed tcp ports (reset)
PORT      STATE SERVICE
7678/tcp  open  unknown
8001/tcp  open  vcom-tunnel
8002/tcp  open  teradataordbms
8080/tcp  open  http-proxy
8187/tcp  open  unknown
9197/tcp  open  unknown
15500/tcp open  unknown
26101/tcp open  unknown
33011/tcp open  unknown
34745/tcp open  unknown
56560/tcp open  unknown
60749/tcp open  unknown
MAC Address: 80:8A:BD:53:28:93 (Samsung Electronics)
```

Figura 7: Scoperta delle porte aperte tramite il comando "nmap -p - 192.168.1.14"

Il comando comprende l'indicatore –p, il quale permette a nmap di scansionare tutte le porte presenti negli IP selezionati, in questo caso solamente uno, quello della televisione. [16]

La scansione ha portato alla scoperta di dodici porte aperte, mentre le altre sono chiuse o non in ascolto, e i rispettivi servizi attivi.

La maggior parte delle porte individuate corrispondono a servizi non conosciuti (cd unknown service), infatti possiamo osservare come solamente a tre porte corrispondano servizi conosciuti, i quali sono [27]:

- Vcom-tunnel, corrispondente alla porta 8001
- Teradataordbms, corrispondente alla porta 8002
- Http-proxy, corrispondente alla porta 8080

È importante ricordare che tutte le porte aperte possono fungere da ingresso a malintenzionati durante l'esecuzione di attacchi.

La porta 8002 è la versione HTTPS dell'endpoint HTTP servito nella porta 8001, la quale, come documentato dalla Samsung, fornisce supporto al debug per gli sviluppatori di applicazioni, utilizzabile dopo aver posto la televisione nella modalità di debug. [28]

La porta 8080 invece è un'alternativa al HTTP. [27]

Una scansione più approfondita potrebbe fornirci anche delle informazioni riguardanti il sistema operativo, utilizzando il comando "nmap -T4 –A 192.168.1.14" sono state ricavate le informazioni seguenti. [16]

```
MAC Address: 80:8A:BD:53:28:93 (Samsung Electronics)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
```

Figura 8: Scoperta del sistema operativo tramite il comando "nmap -T4 -A 192.168.1.14"

È stato così scoperto che il dispositivo utilizza Linux, ovvero un sistema operativo ampliamente utilizzato tra i dispositivi IoT.

Il tag –A definisce una scansione aggressiva mentre il tag –T4 definisce un timing template aggressivo, infatti più basso è il numero che segue la "T" e minore sarà l'impatto che avrà Nmap sulla banda e sugli altri utilizzatori della rete. [16]

Per verificare il sistema operativo in utilizzo dal televisore è possibile anche avvalersi del comanda ping, tale comando viene usato per misurare il tempo in millisecondi impiegato da uno o più pacchetti a raggiungere un dispositivo di rete e a ritornare indietro all'origine.

A seguito di tale comando si ottiene il seguente risultato:

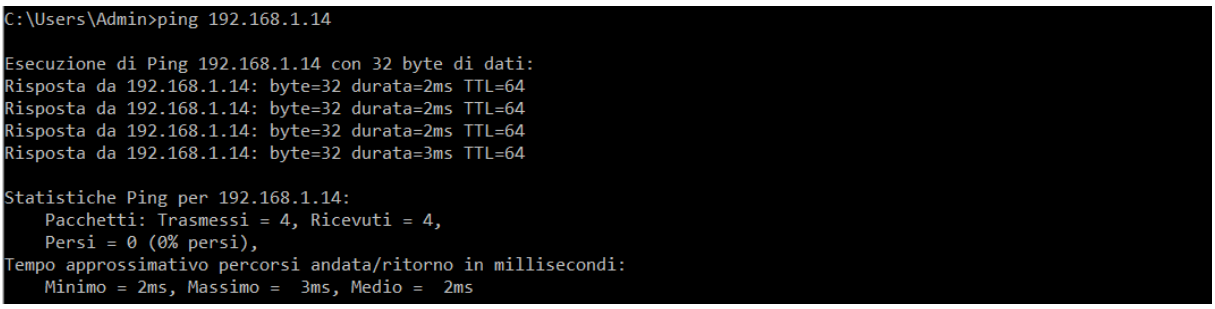

Figura 9: Esecuzione comando "ping 192.168.1.14"

Dal risultato sopra riportato è possibile osservare come il TTL, ovvero il Time To Live, è pari a 64, ciò ci dà un ulteriore riscontro dell'utilizzo di Linux, dato che nel caso dell'utilizzo di Windows tale campo varrebbe 128. [29]

Dato che il televisore è fornito di un web browser, possiamo anche controllare quale sia il browser utilizzato e a quale versione sia aggiornato, in modo tale che si possa verificare la presenza di eventuali exploit già scoperti da terzi.

Per ottenere questa informazione, è possibile osservare il browser fingerprint lasciato accedendo ad un server web. È stato possibile eseguire questa operazione accedendo ad un sito web di mia proprietà e osservare i log, è stato infatti possibile ottenere il seguente risultato:

"GET / HTTP/1.1" 200 15860 "-" "Mozilla/5.0 (SMART-TV; Linux; Tizen 5.5) AppleWebKit/537.36 (KHTML, like Gecko) SamsungBrowser/3.0 Chrome/69.0.3497.106 TV Safari/537.36" "XX.XX.XX.XX"

Così facendo è stato possibile scoprire che il televisore sta utilizzando l'internet browser della Samsung, in particolare la versione 3.0 e la versione di Tizen 5.5.

Per quanto riguarda la raccolta delle informazioni passiva, essa consiste nella ricerca di informazioni di pubblico dominio. In questo caso è stata effettuata una ricerca del dispositivo in rete, la quale ha permesso di entrare a conoscenza di alcuni elementi ritenuti importanti, quali l'esistenza di un'applicazione che permetta di controllare il televisore a distanza, la sopra citata Samsung SmartThings [26].

10.5 Vulnerability analysis

L'obbiettivo di questa fase è quello di analizzare le vulnerabilità del network e delle applicazioni, sulla base delle informazioni che sono state raccolte dal tester nella fase precedente o fornite dall'azienda. Solitamente in questa fase il tester ricorre all'utilizzo di

tecniche manuali ed automatiche per ricercare le vulnerabilità, va comunque ricordato che una ricerca manuale è più lunga e più efficace, escluso il caso in cui si debbano individuare le vulnerabilità di una intera rete aziendale costituita da numerosi host, in questo caso una soluzione automatica verrebbe considerata preferibile.

Lo scrivente ha deciso di avvalersi del software commerciale Nessus, nella sua versione gratuita Nessus Essential [15]. Trattasi di uno scanner delle vulnerabilità, ovvero un programma progettato per accedere ad un computer, una rete o un applicazione per scoprirne le debolezze.

Nessus ha permesso inoltre di confermare i risultati già trovati tramite Nmap sfruttando la funzione di "Host Discovery" effettuato negli indirizzi 192.168.1.0/24.

Host ▾	FQDN
192.168.1.60	Host-001.home
192.168.1.33	LAPTOP-5113VFS7.home
192.168.1.15	hewlett-packard.home
192.168.1.14	Samsung.home

Figura 10: Host discovery tramite Nessus

Effettuando invece una scansione delle vulnerabilità si ricavano i seguenti risultati.

Sev ▾	Score ▾	Name ▴	Family ▴
MEDIUM	6.5	SSL Certificate Cannot Be Trusted	General
MEDIUM	6.4 *	SSL Self-Signed Certificate	General
MEDIUM	5.3	SSL Certificate with Wrong Hostname	General

Figura 11: Vulnerabilità riguardanti l'SSL individuate da Nessus

È possibile osservare che sono stare rilevate tre vulnerabilità classificate di media importanza.

Nessus infatti associa ad ogni vulnerabilità un livello di rischio, il quale può assumere i seguenti valori [15]: critical, high, medium e low.

Tutte le vulnerabilità rilevate riguardano il certificato SSL [30], ovvero un protocollo standard utile alla protezione delle vie telematiche di comunicazione, assicurando così una sicurezza a livello di dati sensibili.

Questa qualità è data dal fatto che ogni tipo di comunicazione viene criptata tra client server e web server, e quest'ultimo inviando il certificato SSL al browser in utilizzo, a seguito di una validazione di esso, ne permette la lettura. [30]

A riguardo il sito ufficiale della Samsung afferma quanto segue:

"To transmit encrypted information between Web browsers and Web servers, Samsung Smart TVs support TLS (Transport Layer Security) versions 1.0, 1.1, and 1.2.

A TV firmware upgrade can be required to patch the POODLE attack vulnerability. Consequently, SSL (Secure Sockets Layer) is no longer supported."

Ciò significa che la Samsung in realtà non utilizza più il certificato SSL nelle proprie Smart TV.

Data l'esistenza dell'applicazione Samsung SmartThing, essa può rappresentare un rischio per la sicurezza del dispositivo, poiché ogni possibile interazione con un dispositivo può tramutarsi in un possibile mezzo di attacco nelle mani di un malintenzionato. Sarà quindi necessario verificarne la sicurezza.

I test che sarà necessario effettuare verificano i seguenti punti:

T1 – Impedire l'accesso all'applicazione da parte dell'utente.

T2 – Assumere il controllo del dispositivo tramite l'applicazione.

Per proseguire nell'analisi delle vulnerabilità è stata sfruttata una delle metodologie precedentemente descritte, ovvero la OWASP ISVS [25].

Innanzitutto è necessario identificare il livello di sicurezza da assegnare al televisore, essendo un dispositivo che non memorizza dati fondamentali, gli sarà assegnato il primo livello (L1). [25]

Di seguito verranno riportati i risultati ottenuti seguendo le indicazioni riportate dal modello seguito. Verranno riportati solamente i punti che hanno generato dei risultati ritenuti interessanti, i quali potrebbero portare a delle falle nella sicurezza.

Per quanto riguarda i requisiti del user space application (V2) [25], attraverso una breve verifica del sito della Samsung [31], durante la creazione di un account, è possibile verificare i requisiti minimi che deve avere una password. Essa non richiede un minimo di dodici caratteri e non richiede l'inserimento di caratteri speciale, in modo tale da rendere le password sufficientemente complesse.

Ciò significa che i punti 2.1.5 e 2.1.7 [25] non sono stati correttamente rispettati. Verrà quindi successivamente verificata la possibilità di sfruttare questa informazione.

2.1.5	Verify that passwords used for user authentication are at least 12 characters long.
2.1.6	Verify that passwords used for user authentication can be changed by the user and that the password change functionality requires the user's current and new password.
2.1.7	Verify that passwords used for device authentication are sufficiently long and complex.

Figura 12: Requisiti 2.1.5-7 del ISVS

Data la non eccessiva sicurezza delle password, si potrà provare ad eseguire il seguente test:

T3 – Brute force della password dell'applicazione.

Per quanto riguarda la Software Platform (V3), il punto 3.4.2 [25] richiede la verifica che il dispositivo venga aggiornato automaticamente alle nuove versioni. Tale condizione è stata verificata durante la fase di information gathering, infatti nel sito ufficiale della Samsung viene descritto come vengono gestiti gli aggiornamenti [32]: essi vengono installati in background durante la normale visione della televisione e, solamente all'avvio successivo del dispositivo verranno resi utilizzabili; inoltre di default l'aggiornamento automatico è attivo. Tale informazione è stata successivamente verificata da utente ed è corretta.

3.4.2	Verify that devices can be updated automatically upon a pre-defined schedule.

Figura 13: Requisito 3.4.2 del ISVS

Durante la raccolta delle informazioni è stato possibile verificare quali porte aperte abbia il dispositivo. È stato così notato come siano aperte numerose porte TCP, le quali ci forniscono l'opportunità di eseguire il seguente test:

T4 – Attacco SYN Flood.

È stato inoltre possibile osservare come il televisore sia dotato di un web browser, ciò crea ulteriori rischi all'utente, il quale può imbattersi in tentativi di phishing. Sarà quindi necessario verificare anche questa eventualità in un ulteriore test.

T5 – Attacco phishing.

10.6 Vulnerability exploits

L'obbiettivo di questa fase è il determinare come sfruttare le vulnerabilità rilevate durante la fase di vulnerability analysis.

T1 – Impedire l'accesso all'applicazione da parte dell'utente.

Un malintenzionato potrebbe provare a rendere l'applicazione SmartThings [26] inutilizzabile da parte dell'utente, rendendolo quindi virtualmente isolato dai dispositivi connessi a tale applicazione.

Lo svolgimento del test consiste nello sfruttare la comune funzionalità di login che viene richiesta dall'applicazione per poter accedere ad un account SmartThings. Tale test necessita la conoscenza da parte del malintenzionato dell'indirizzo email della vittima.

Un ripetuto inserimento di password sbagliate al momento del login fa in modo che l'account Samsung della vittima venga bloccato, richiedendo un successivo ripristino della password. In particolare si è verificato che il numero di tentativi permessi prima del blocco va dai sette ai nove nel caso di accesso da dispositivo mobile, nel caso di accesso da computer vengono fornite ulteriori possibilità a seguito dello svolgimento di alcune sfide di autenticazione (cd. CAPTCHA [33], ovvero dei test costituiti da una o più domande e risposte per determinare se l'utente sia un umano o meno). Tale blocco vale solamente per i nuovi accessi all'account Samsung, nel caso un utente ne sia già collegato tramite l'applicazione potrà continuare ad usufruirne senza subire le restrizioni del blocco o dover ripristinare la password.

Tale tentativo di isolamento non è comunque sufficiente a creare dei danni al fruitore del servizio, poiché allo stato attuale della digitalizzazione sono pochi i dispositivi accessibili IoT accessibili unicamente tramite applicazione. Ad esempio nel caso in esame il televisore non è accessibile solo tramite dispositivo mobile, ma anche tramite un comune telecomando.

T2 – Assumere il controllo del dispositivo tramite l'applicazione.

Un malintenzionato potrebbe voler prendere il controllo su un dispositivo connesso in rete tramite l'utilizzo di un ulteriore account Samsung, cercando di connettere il dispositivo anche ad un secondo account. Si è quindi provato a connettere la televisione ad un secondo account, seguendo la procedura di connessione fornita dall'applicazione. Prerequisiti di questa tecnica sono la creazione di un account Samsung, l'essere fisicamente vicini al dispositivo, l'avere accesso alla rete Wi-Fi alla quale è connesso il televisore e collegarsi ad esso tramite Bluetooth.

Ho quindi scaricato l'applicazione SmartThings in un secondo dispositivo (telefono Huawei), creare un nuovo account Samsung e provare a ottenere il controllo sul televisore.

Dato che esso è già collegato ad un account vengono offerte due opzioni al malintenzionato, la prima prevede di notificare il proprietario del dispositivo per richiederne l'accesso oppure verificare che si vuole diventare "proprietari". Tale operazione non richiede alcuna notifica al proprietario, ma necessita l'utilizzo del telecomando.

Tutto ciò rende impossibile l'esecuzione di questa operazione.

T3 – Brute force della password dell'applicazione.

Un malintenzionato potrebbe sfruttare tale tecnica per provare ad accedere all'account Samsung della vittima, così da poterne utilizzare tutti i dispositivi connessi ad essa. Nel caso tale operazione fosse possibile da effettuare si correrebbero elevati rischi, si pensi all'ottenere l'accesso a dispositivi connessi all'applicazione che controllano elementi fondamentali della casa quali serrature o telecamere.

Lo scrivente ha quindi provato ad utilizzare questa tecnica, ma si è ritrovato dinnanzi ad una repentina riposta da parte del colosso dell'informatica Samsung. Il malintenzionato si troverà ad affrontare numerosi CHPTCHA [33] e il successivo blocco dell'account, rendendo quindi vana l'esecuzione di questo attacco.

T4 – Attacco SYN Flood.

Il gran numero di porte TCP aperte porta il dispositivo ad essere suscettibile ad attacchi SYN flood.

Questa tecnica consiste in attacchi denial-of-service che mirano a rendere un server indisponibile al traffico legittimo, consumandone tutte le risorse e costringendolo a rifiutare richieste legittime o a ritardarne il più possibile la risposta. Tale condizione si ottiene sfruttando il processo di handshake di una connessione TCP, inviando ripetutamente pacchetti di richieste di connessione (SYN); nel caso il bersaglio dell'attacco non abbia qualche metodo per controllare la congestione, esso verrà sopraffatto dall'elevata mole di richieste alla quale deve rispondere. [34]

Per eseguire questo test è stato utilizzato il software Metasploit [13] e, durante il suo utilizzo è stato verificato il tempo di risposta del server tramite il comando ping.

Una volta aperto Metasploit, è necessario selezionare lo strumento ausiliario "synflood", per fare ciò è stato utilizzato il comando "search synflood", il quale ci permette di ottenere il nome

completo dello strumento ausiliario, il quale come è possibile osservare nell'immagine sottostante è "auxiliary/dos/tcp/synflood". [14]

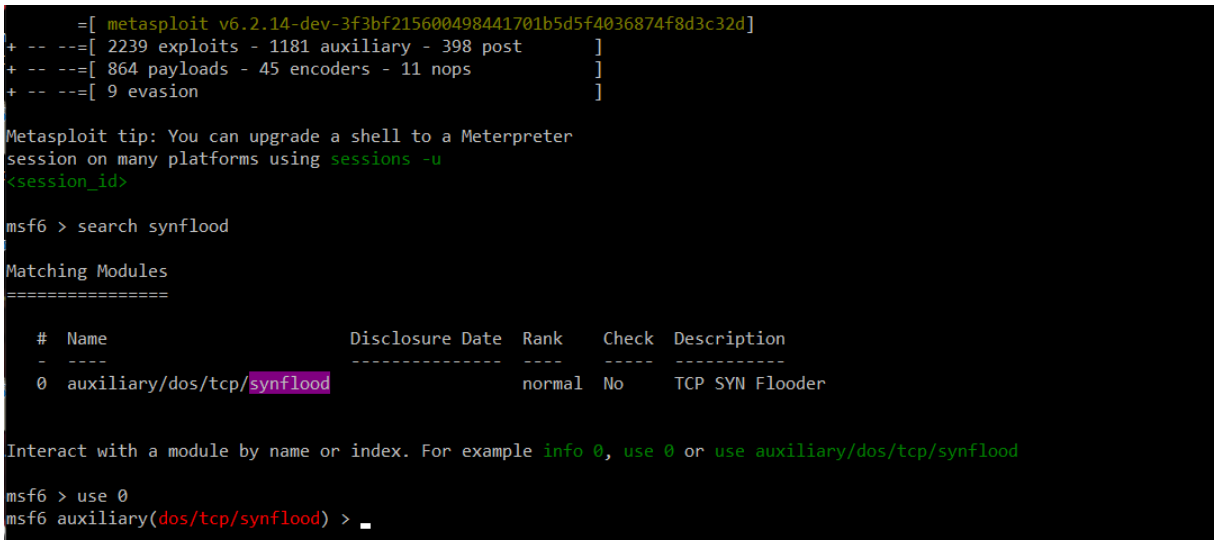

Figura 14: Ricerca e selezione dello strumento ausiliario synflood su Metasploit

Dopo aver selezionato lo strumento desiderato, è stato quindi possibile selezionare l'indirizzo IP del dispositivo bersaglio tramite il comando "set RHOST 192.168.1.14" e la porta desiderata tramite il comando "set RPORT 60749". La porta selezionata è una delle porte TCP aperte ricavate precedentemente tramite il software Nmap. Sarà poi possibile osservare le impostazioni selezionate per l'attacco tramite il comando "show options". [14]

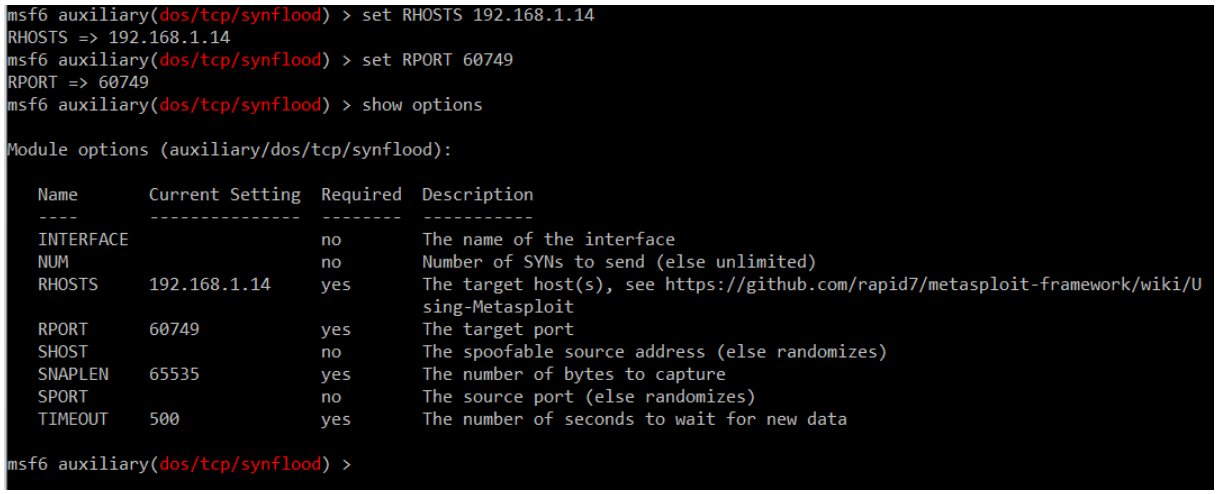

Figura 15: Selezione impostazione per l'attacco synflood su Metasploit

A seguito della selezione delle impostazioni riguardanti l'attacco, sarà necessario eseguirlo, tale operazione è possibile tramite i comando "exploit"

Figura 16: Inizio attacco synflood tramite il comando "exploit"

È ora necessario provare a verificare l'effettiva efficacia ti tale operazione. Per poterne controllare il risultato è stato utilizzato il comando "ping –t 192.168.1.14" sia prima che durante l'esecuzione dell'attacco. L'istruzione "–t" permette di poter eseguire la chiamata continuamente, finché non viene interrotta.

```
C:\Users\Admin>ping -t 192.168.1.14

Esecuzione di Ping 192.168.1.14 con 32 byte di dati:
Risposta da 192.168.1.14: byte=32 durata=2ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=2ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=3ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=3ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=3ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=4ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=3ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=2ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=2ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=2ms TTL=64

Statistiche Ping per 192.168.1.14:
    Pacchetti: Trasmessi = 10, Ricevuti = 10,
    Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 2ms, Massimo =  4ms, Medio =  2ms
```

Figura 17: Esecuzione del comando ping prima dell'inizio dell'attacco synflood

```
C:\Users\Admin>ping -t 192.168.1.14

Esecuzione di Ping 192.168.1.14 con 32 byte di dati:
Risposta da 192.168.1.14: byte=32 durata=7ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=6ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=15ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=4ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=2ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=9ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=4ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=7ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=6ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=2ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=5ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=2ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=12ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=2ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=6ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=25ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=6ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=9ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=20ms TTL=64
Risposta da 192.168.1.14: byte=32 durata=4ms TTL=64

Statistiche Ping per 192.168.1.14:
    Pacchetti: Trasmessi = 20, Ricevuti = 20,
    Persi = 0 (0% persi),
Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 2ms, Massimo =  25ms, Medio =  7ms
```

Figura 18: Esecuzione del comando ping durante l'esecuzione dell'attacco synflood

Dai risultati sopra riportati è possibile osservare come ci sia un leggero aumento del tempo in millisecondi per rispondere ai pacchetti inviati, la media infatti passa dai due millisecondi ai sette millisecondi.

Tale ritardo non è comunque sufficiente per arrecare un effettivo disagio all'utente del televisore. È quindi possibile affermare che questo test ha portato ad un piccolo successo, è

stato infatti possibile portare a termine il test, ma con degli scarsi risultati, poiché non è stato possibile congestionare la rete.

T5 – Attacco phishing.

Uno dei possibili obbiettivi di un malintenzionato è l'ottenimento di dati personali, finanziari o codici di accesso delle vittime. Ciò è possibile attraverso una tecnica detta phishing, attraverso la quale un malintenzionato cerca di ingannare la vittima, fingendosi un ente affidabile e portandola ad inserire i propri dati in un sito che mira a replicare il più possibile il sito dell'ente selezionato. [35] Sarà così possibile per il malintenzionato ottenere i dati inseriti dalla vittima in un database selezionato o semplicemente in un documento di testo.

Questo tipo di attacco è molto comune, infatti solo nel 2021 sono stati rilevati dal Anti-Phishing Working Group (APWG) [36] un totale di 2.847.773 siti malevoli eseguenti operazioni di phishing e un totale di 484.469 argomenti di email malevoli. Solo nel primo quadrimestre del 2022 sono stati individuati 1.025.968 siti web malevoli, il risultato più alto di sempre.

Questo continuo aumento degli attacchi viene causato anche dall'enorme facilità con la quale è possibile realizzare siti web identici a quelli di enti ufficiali, è infatti sufficiente utilizzare semplici strumenti online per poter clonare in pochi secondi interi siti web, senza la necessità di dover avere conoscenze nella programmazione di siti web.

Per lo svolgimento di questa tecnica è stato necessario appoggiarsi ad un sito web, nel quale sono presenti dei campi di input per poter sottrarre all'utente i propri dati.

Il sito web è stato realizzato in HTML e CSS, utilizzando MySQL per poter memorizzare i dati inseriti nel sito e il PHP per poter interagire con il suddetto database.

È stato successivamente necessario accedere dal web browser del televisore al sito web realizzato e inserire i dati richiesti.

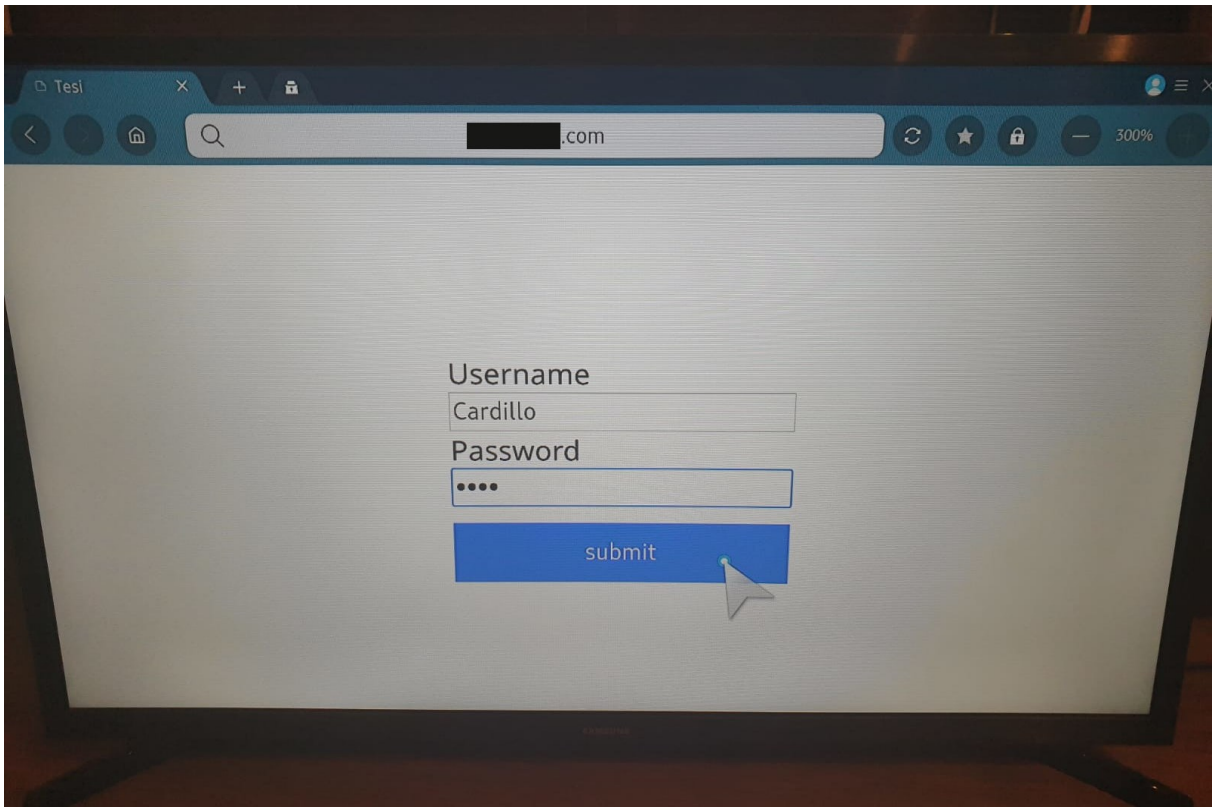

Figura 19: Accesso al sito web di phishing e inserimento dei dati personali

A seguito di tali passaggi sarà possibile accedere al database MySQL desiderato per poter vedere tutti i tentativi di accesso eseguiti dai malcapitati che erano convinti di trovarsi nel sito desiderato.

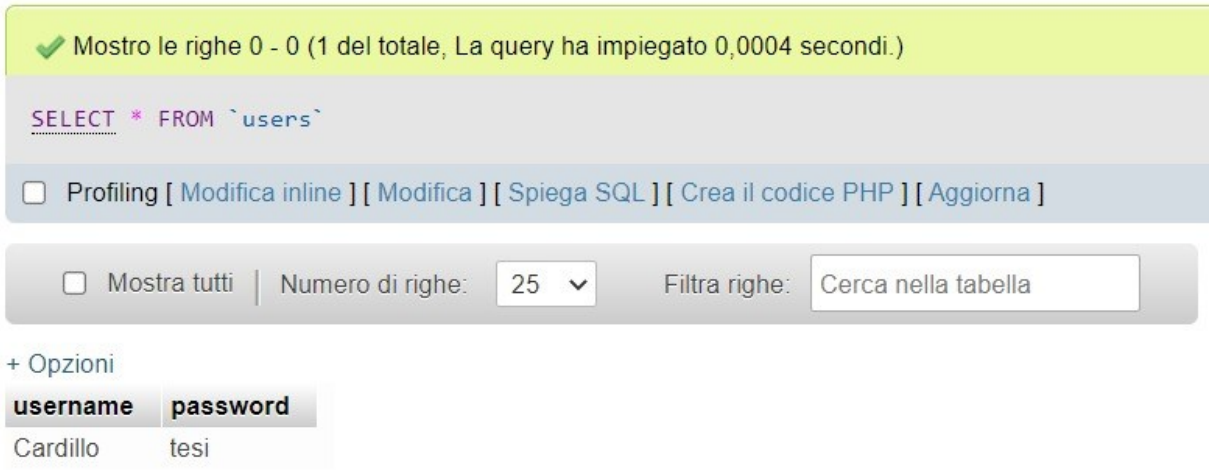

Figura 20: Accesso al database contenente i dati raccolti dal sito di phishing

È possibile osservare come, nel caso fosse stata implementata una interfaccia utente (cd user interface o UI) che voglia ingannare l'utente, sarebbe facilmente possibile portarlo a credere esso stia interagendo con un sito web legittimo, ottenendone così le informazioni riservate.

La difficoltà che presenta tale tecnica riguarda il come attirare le vittime sul sito web malevolo. Tra i metodi più utilizzati ci sono:

Invio di email o SMS riguardo un determinato servizio indicando una necessità, spesso urgente, di accedere al sito web in questione. Accedendo tramite il link presente nella email o nel messaggio si accederà al sito web malevolo. [35]

Registrare il sito web malevolo con un dominio simile al dominio reale del sito che si vuole imitare, ad esempio sostituendo lettere simili, come la "e" e la "a", oppure utilizzando un suffisso differente da quello originale, ad esempio sostituendo ".it" con ".com". Così facendo sarà possibile sfruttare anche eventuali errori di digitazione da parte degli utenti del sito cercato.

Utilizzo di codici QR per portare all'apertura della pagina web senza la necessità di cliccare link.

Utilizzo di social network per contattare direttamente le vittime e trarle in inganno.

11. Conclusione

In questa tesi sono stati analizzati i diversi approcci che un penetration tester può utilizzare per poter verificare la sicurezza di una piattaforma o di un sistema e sono state mostrate le diverse metodologie che possono accompagnare il tester durante la sua analisi, fungendo da linea guida minima alle attività e ai test da eseguire. È stato possibile osservare come esistano molteplici metodologie differenti, le quali hanno come obbiettivo l'analisi di specifiche piattaforme di attacco.

Sono stati inoltre mostrati diversi software che possono essere utilizzati per eseguire delle operazioni automatiche in supporto al tester, essi si distinguono in base alla loro funzione e alle operazioni che svolgono.

È stato inoltre eseguito un penetration test su un televisore smart, seguendo la metodologia ISVS, proposta dalla OWASP. È stato possibile così facendo valutare il livello di sicurezza del dispositivo analizzato, mostrando l'importanza che assume lo svolgere questo tipo di test. L'utilizzo di metodologie, tecniche e strumenti, uniti all'esperienza del tester, permettono di individuare le eventuali debolezze nei dispositivi o nei sistemi, per poi poterli sfruttare per risolvere le falle nella sicurezza prima che le possa utilizzare un malintenzionato.

Il penetration test sul televisore della Samsung ha mostrato la grande utilità nell'utilizzo di una metodologia adeguata all'area di analisi presa in esame, la quale ha permesso uno studio accurato del dispositivo, fornendo lo spunto per l'esecuzione dei test eseguiti successivamente.

Nonostante l'analisi eseguita abbia portato ad alcuni risultati importanti come la vulnerabilità ad attacchi di tipo phishing, è stato generalmente appurato come le misure di sicurezza prese dalla Samsung siano adeguate per permettere il superamento alla quasi completezza dei test eseguiti.

12. Bibliography and Sitography

[1] Che cos'è l'Internet of Things (IoT):

https://www.redhat.com/it/topics/internet-of-things/what-is-iot#:~:text=L'acronimo%20IoT%20indica%20qualsiasi,negli%20oggetti%20dispositivi%20di%20elaborazione

[2] Penetration testing tools: https://sectools.org/?sort=rank

[3] Daniel Dalalana Bertoglio, Avelino Francisco Zorzo, "Overview and open issues on penetration test", Journal of the Brazilian Computer Society, 2017 - https://journal-bcs.springeropen.com/articles/10.1186/s13173-017-0051-1

[4] Importanza dei penetration test:

https://www.getastra.com/blog/security-audit/why-penetration-testing-is-important/

[5] Perdite economiche conseguenti ad attacchi informatici: https://www.ibm.com/security/data-breach

[6] Tesla model X, criticità scoperte: https://www.wired.com/story/tesla-model-x-hack-bluetooth/

[7] Smart TV: https://en.wikipedia.org/wiki/Smart_TV

[8] Aileen G. Bacudio, Xiaohong Yuan, Bei-Tseng Bill Chu, Monique Jones, "AN OVERVIEW OF PENETRATION TESTING", International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.6, November 2011 - https://www.airccse.org/journal/nsa/1111nsa02.pdf

[9] IoT devices (Internet of Things) https://www.techtarget.com/iotagenda/definition/IoT-device

[10] Shebli, Hessa Mohammed Zaher Al and Babak D. Beheshti. “A study on penetration testing process and tools.” 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT), 2018 - https://www.semanticscholar.org/paper/A-study-on-penetration-testing-process-and-tools-Shebli-Beheshti/957180bd049d2966f26c0c101f5867503cc2bb2a

[11] Wireshark: https://www.wireshark.org/

[12] What is Wireshark: https://www.csoonline.com/article/3305805/what-is-wireshark-what-this-essential-troubleshooting-tool-does-and-how-to-use-it.html

[13] Metasploit: https://www.metasploit.com/

[14] Metasploit documentation: https://docs.rapid7.com/metasploit/

[15] Nessus: https://www.tenable.com/products/nessus

[16] Nmap: https://nmap.org/

[17] W3af: http://w3af.org/

[18] Zed Attack Proxy: https://www.zaproxy.org/

[19] Acunetix WS: https://www.acunetix.com/

[20] Pete Herzog, “OSSTMM 3 – The Open Source Security Testing Methodology Manual”, ISECOM, pg. 33-40, 2010 - https://www.isecom.org/OSSTMM.3.pdf

[21] “Information Systems Security Group Information Systems Security Assessment Framework (ISSAF) draft 0.2.1”, OISSG, April 30, 2006 - https://untrustednetwork.net/files/issaf0.2.1.pdf

[22] PTES: http://www.pentest-standard.org/index.php/Main_Page

[23] NIST: https://csrc.nist.gov/Projects/cybersecurity-framework/nist-cybersecurity-framework-a-quick-start-guide

[24] OWASP: https://owasp.org/

[25] ISVS: https://github.com/OWASP/IoT-Security-Verification-Standard-ISVS

[26] Samsung SmartThings: https://www.samsung.com/it/apps/smartthings/

[27] Service Name and Transport Protocol Port Number Registry:

https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

[28] Smart TV debugging mode:

https://developer.samsung.com/smarttv/develop/extension-libraries/smart-view-sdk/receiver-apps/debugging.html

[29] Default TTL (Time To Live) Values of Different OS: https://subinsb.com/default-device-ttl-values/

[30] SSL: https://ssl.com

[31] Samsung: https://www.samsung.com/

[32] Smart TV aggiornamento del software:
https://www.samsung.com/us/support/answer/ANS00062224/

[33] Captcha: https://en.wikipedia.org/wiki/CAPTCHA

[34] B, Prabadevi & Nagamalai, Jeyanthi. “A Review on Various Sniffing Attacks and its Mitigation Techniques”, Indonesian Journal of Electrical Engineering and Computer Science, Vol. 12, No. 3, pg. 1117 – 1125, 2018 -

https://www.researchgate.net/publication/329467167_A_Review_on_Various_Sniffing_Attacks_and_its_Mitigation_Techniques

[35] Phishing, Polizia Postale e delle Comunicazioni: https://www.commissariatodips.it/approfondimenti/phishing/phishing-che-cose/index.html

[36] APWG phishing trends report: https://apwg.org/

Kevin M. Henry, “Penetration Testing Protecting networks and systems”, IT Governance Publishing, UK, 2012 - https://scholar.google.com/scholar_lookup?title=Penetration%20testing%3A%20protecting%20networks%20and%20systems&publication_year=2012&author=Henry%2CKM