Università degli Studi di Camerino

Scuola di Scienze e Tecnologie

Corso di Laurea in Informatica (Classe L-31)

Automated tools for Penetration Testing

Laureando
Alessandro Buccolini

Matricola 101748

Relatore
Fausto Marcantoni

A.A. 2019/2020

Indice

Abstract

L'informatica è il settore protagonista, nell'era moderna in cui ci troviamo, di una crescita esponenziale che riguarda, non solo l'innovazione delle tecnologie, ma anche la mole di persone che ne utilizzano i suoi prodotti.
Miliardi di informazioni circolano nel mondo sotto forma di byte e milioni di persone lavorano per creare un mondo sempre più interconnesso. Nel 2020 circa 27 milioni di programmatori hanno dato vita a nuovi software e, di conseguenza, infinite vulnerabilità che un hacker può sfruttare a suo favore.

La sicurezza è il ramo del settore informatico che cerca di trovare queste vulnerabilità. Lo scopo è informare il proprietario del software ed evitare che dati sensibili vengano esposti. Le modalità con cui opera consistono nel simulare i medesimi attacchi che un malintenzionato potrebbe mettere in atto.

Vulnerability assessment e *penetration testing* sono le principali azioni che un hacker esegue nei sistemi informatici. Le suddette pratiche vengono tradizionalmente eseguite in modo "manuale" ma, negli ultimi anni, sono stati sviluppati numerosi strumenti per renderle automatiche. Il seguente elaborato ha come obiettivo spiegarne il funzionamento ed analizzarne le funzionalità, fornendo un'analisi sui possibili sviluppi futuri.

1. Introduzione

La crescente interconnessione del mondo digitale tramite l'uso di Internet ha cambiato il modo in cui le imprese, i governi e gli individui operano ed ha portato a significativi benefici sociali . Questa migliore comunicazione e disponibilità, tuttavia, ha anche creato maggiori opportunità per i criminali informatici di lanciare attacchi dannosi nella speranza di ottenere l'accesso ai dati sensibili per il proprio guadagno. Questi azioni offensive possono variare da massicci attacchi sponsorizzati dallo stato, a semplici attacchi contro singoli individui nella speranza di ottenere password o dettagli della carta di credito per guadagni monetari. Sempre più organizzazioni e individui si affidano a sistemi informatici connessi a livello globale e la capacità di proteggersi dai suddetti attacchi sta diventando sempre più importante.

L'ascesa di nuove tecnologie è quindi seguita dalla sfida di fornire un ambiente sicuro. Uno studio del 2005 condotto dal Federal Bureau of Investigation (FBI) suggerisce che oltre l'87% delle aziende statunitensi sia stato vittima di attacchi dannosi ogni anno. Lo studio indica inoltre che le perdite complessive potrebbero raggiungere i 67 milioni di dollari all'anno. Il verificarsi di attacchi dannosi è aumentato enormemente come si evince dal grafico del Computer Emergency Response Team USA (CERT)[Cer] in figura 1.1:

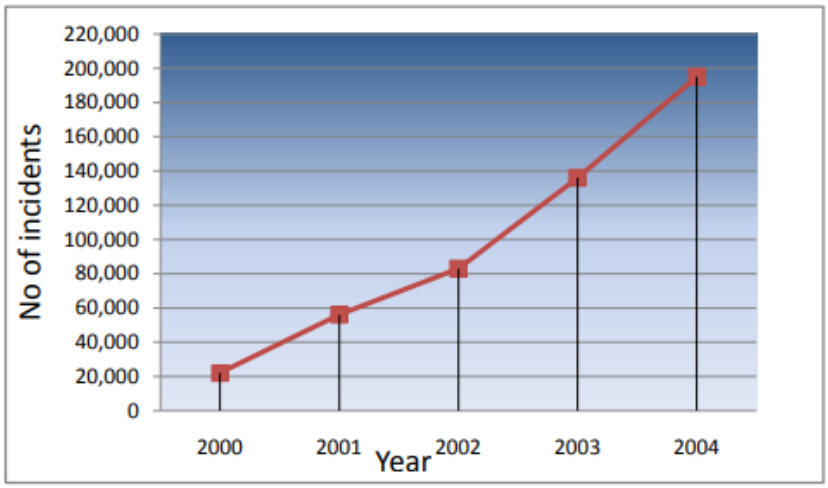

Figura 1.1: Grafico sul numero di incidenti annuali [CERT]

Il CERT italiano, cosciente del pericolo che può comportare una carente attenzione sulla sicurezza informatica, in seguito al DPCM 8 Agosto 2019 ha creato il Cert-AgID, una struttura che si occupa di mantenere e sviluppare servizi di sicurezza preventivi in ambito informatico utili alle pubbliche amministrazioni.

Tutto il mondo è ormai consapevole di quanto sia rischioso trascurare il campo della sicurezza informatica. Lo stesso AgID ha dichiarato nel proprio piano triennale per l'Informatica nella Pubblica Amministrazione: "In un momento storico nel quale la minaccia cibernetica cresce continuamente in quantità e qualità e i servizi informatici e telematici erogati dalla Pubblica Amministrazione diventano sempre più cruciali per il funzionamento del sistema Paese, la sicurezza informatica riveste un ruolo fondamentale in quanto garantisce non solo la disponibilità, l'integrità e la riservatezza delle informazioni proprie del Sistema informativo della Pubblica Amministrazione, ma anche la resilienza della complessa macchina amministrativa. Essa è inoltre direttamente collegata ai principi di privacy previsti dall'ordinamento giuridico." [Dig19]

La Cyber Security, ovvero la salvaguardia dei sistemi informatici da attacchi di persone non autorizzate, è ora una questione di importanza e interesse globale. Molte nazioni hanno stabilito politiche ufficiali per quanto concerne la sicurezza informatica ed alcune di esse stanno investendo somme significative in questo campo. Ne è portavoce il governo australiano che, nel 2018, ha destinato investimenti per circa 50 milioni di dollari su questo fronte. [Comunicato [Lau]]

Questa maggiore attenzione e investimento da parte dei governi e delle grandi organizzazioni sottolinea la grave minaccia rappresentata dai crimini informatici. È importante che metodi e tecnologie efficaci vengano sviluppati per proteggere i sistemi tecnologici da questi pericoli.

Uno dei metodi più comunemente applicati per valutare la sicurezza di un sistema è un test di penetrazione (*pentesting*). Il *pentesting* comporta l'esecuzione di un attacco controllato e autorizzato ad un sistema al fine di scovare eventuali vulnerabilità di sicurezza potenzialmente sfruttabili da un attaccante. Si tratta essenzialmente di una simulazione di come gli aggressori agirebbero nel mondo reale. Nonostante l'efficacia di questa tecnica, essa presenta degli svantaggi rilevanti: ha un costo elevato in termini di tempo e richiede molta abilità per eseguirla.

Per quanto concerne l'elevato costo, esso sta diventando un problema sempre più grande ora che i sistemi digitali sono cresciuti esponenzialmente in dimensioni, complessità e quantità. Le forti carenze di professionisti della sicurezza fanno sì che la loro domanda sia in costante aumento. A causa della necessità di pentesters sta diventando cruciale che vengano sviluppati strumenti e metodi per rendere il pentesting più efficiente. Nel 2015 *Cisco*, uno dei leader mondiali nel settore IT e società di *networking*, nel comunicato [Cis] ha stimato che ci fossero più di 1 milione di posti di lavoro nel settore della sicurezza in tutto il mondo.

La figura 1.2 sotto riportata mostra un grafico ripreso dal sito *Statista* [Sta] che mette in luce la crescita del mercato *cyber* a livello mondiale.

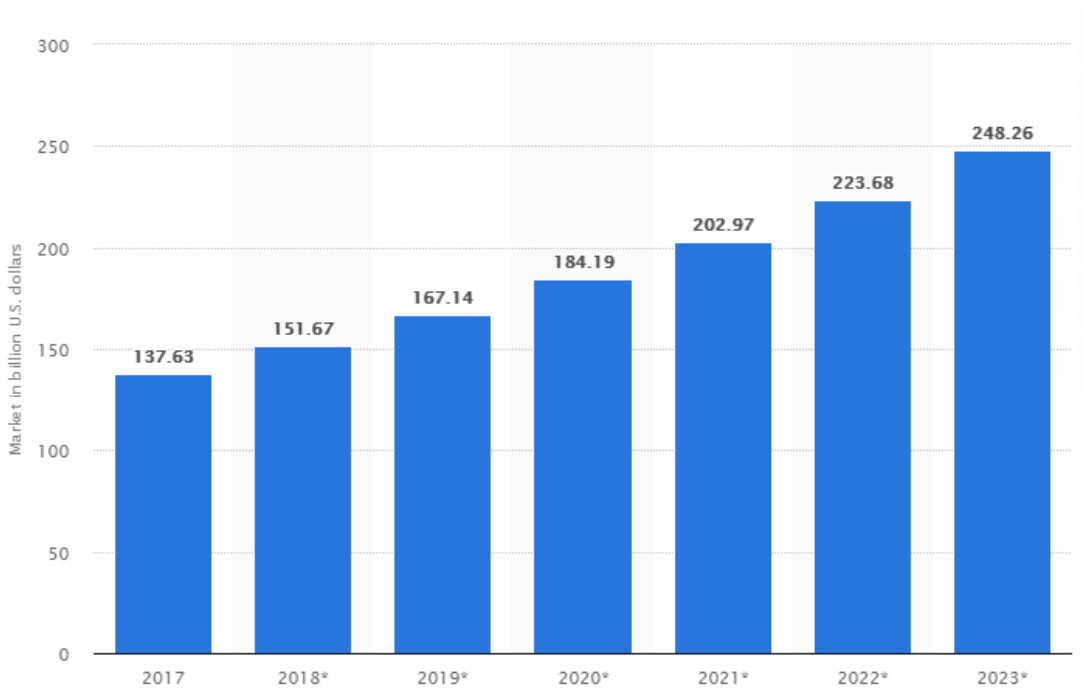

Figura 1.2: Crescita del mercato cyber security secondo Statista

Un risultato degli investimenti in questo settore è la nascita di nuovi strumenti, sempre più affidabili ed efficienti, per il *penetration testing*. Se tempo fa, un addetto alla sicurezza informatica, per verificare la sicurezza di un sistema, avrebbe verificato manualmente la presenza di vulnerabilità, ad oggi esistono software che lo agevolano nel suo lavoro. Molti dei compiti necessari al *penetration testing* sono automatizzati senza perdere in efficacia.

In questa tesi si vogliono indagare le tecnologie, nascenti o già confermate nel mercato, utilizzate ai fini dell'automatizzazione della sicurezza dei sistemi IT. Si pongono in rilievo considerazioni circa l'utilità e l'affidabilità, comparando diversi prodotti per capire le differenze dei vari software.

Nel Capitolo 1 illustreremo prima le motivazioni che ci hanno spinto a perseguire l'obiettivo descritto e quindi la struttura della tesi.

1.1 Obiettivo

L'obiettivo di questa tesi è quello di ottenere informazioni riguardanti i software di pentesting automatico. Vogliamo capire il funzionamento di questi programmi, quanto siano più efficienti rispetto ad un penetration test manuale e cercare di stabilire quanto siano più affidabili in determinate condizioni. Inoltre stabiliremo quali sono le tecnologie più ricorrenti utilizzate nei tools automatici e quale linguaggio sia più conveniente per il loro sviluppo. L'attenta analisi che produrremo di questi strumenti ci servirà poi come base per un focus circa il loro futuro e la loro evoluzione verso l'utilizzo di tecnologie moderne come il Machine Learning e le Intelligenze Artificiali.

1.2 Struttura della Tesi

La tesi è strutturata in 4 capitoli. Nel capitolo 1 abbiamo introdotto l'importanza della cyber security nel mondo e come grandi investimenti nel settore abbiano portato allo sviluppo di nuove tecnologie automatiche per il pentesting.

Nel capitolo 2 spiegheremo che Penetration Testing e Vulnerability Assessment sono le attività principali nella messa in sicurezza di un sistema e ne daremo una descrizione sommaria di cosa sono e come vengono eseguite. Ci concentreremo poi sugli strumenti automatici, protagonisti della tesi, anticipando cosa sono e comparandoli con il pentesting manuale, per stabilire quale sia il migliore.

Il capitolo 3 è il fulcro della tesi. Inizialmente elencheremo una serie di strumenti commerciali per il pentesting automatico che sono destinati a grandi realtà come aziende o banche, cercando di ottenere più informazioni possibili al riguardo tramite la loro documentazione. In seguito metteremo mano sulla parte opensource di questi software, utilizzandoli per test ed analizzandone la struttura.

In chiusura, nel capitolo 4, faremo un excursus su quanto elaborato precedentemente, traendo le conclusioni della nostra analisi e discutendo di come gli strumenti automatici per il pentesting potranno migliorarsi in futuro.

2. Overview

2.1 Penetration testing

Il test di penetrazione è un attacco informatico simulato che gli hacker etici professionisti lanciano per penetrare nelle reti aziendali e trovare i punti deboli prima che lo facciano gli aggressori.
Questo processo utilizza strumenti e tecniche disponibili agli hacker malintenzionati. Può essere automatizzato con applicazioni software o eseguito manualmente.

Il processo prevede la raccolta di informazioni sull'obiettivo prima del test, l'identificazione di possibili punti di ingresso, il tentativo di irrompere, virtualmente o per davvero, e riportare i risultati. L'obiettivo principale dei test di penetrazione è identificare i punti deboli della sicurezza.
Può anche essere utilizzato per testare la *policy* di sicurezza di un'organizzazione, la sua aderenza ai requisiti di conformità, la consapevolezza della sicurezza dei suoi dipendenti e la sua capacità di identificare e rispondere agli incidenti di sicurezza. Ad esempio, se la sicurezza di un'azienda si concentra sulla prevenzione e il rilevamento di un attacco ai propri sistemi, potrebbe non includere un processo per espellere un hacker.

I report generati da un penetration test forniscono il feedback necessario a un'organizzazione per dare la priorità agli investimenti che intende fare nella sua sicurezza e agli sviluppatori di applicazioni per creare app più sicure.

Il processo di penetration test, come illustrato nella figura 2.1 (Fonte: [Nay20]), può essere suddiviso nelle seguenti cinque fasi:

- **Pianificazione e ricognizione**
- **Scansione**
- **Ottenere accesso**
- **Mantenere l'accesso**
- **Configurazione analisi e WAF (Web application firewall)**

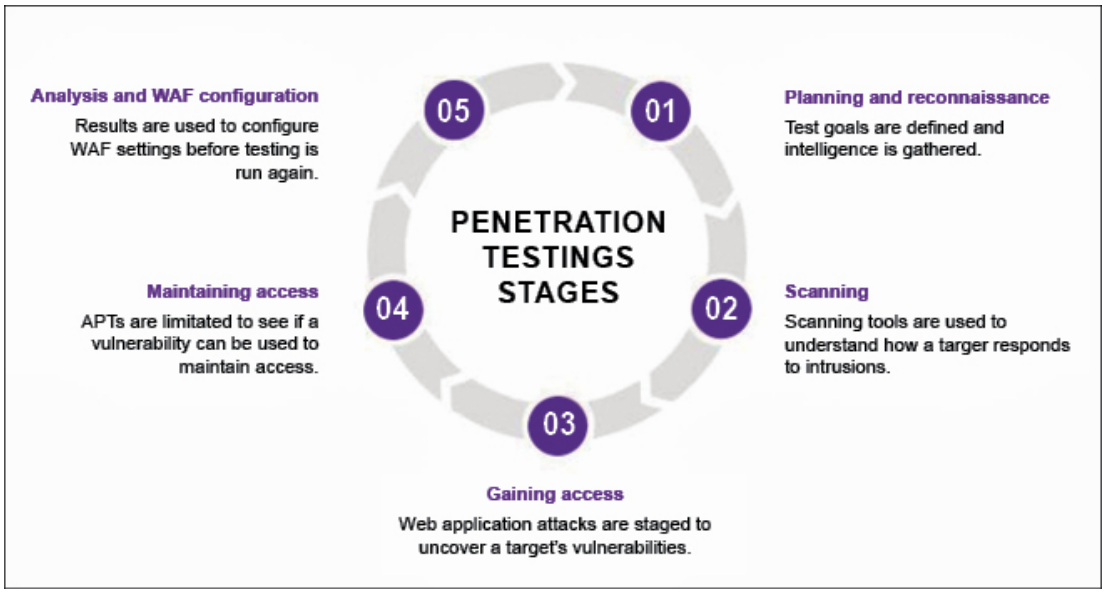

Figura 2.1: Fasi del pentesting

Ci sono numerosi vantaggi per l'utilizzo di penetration test in prospettiva sia commerciale che tecnica:

- **Problemi di sicurezza** Problemi di sicurezza come attacchi di malware, intrusioni di rete, e il furto di dati possono comportare l'interruzione del servizio e processi di sistema inaffidabili. Ciò potrebbe portare alla potenziale perdita di fedeltà del cliente e influenzare il valore di mercato dell'azienda. I test di penetrazione possono queste conseguenze eliminando minacce persistenti e inaspettate.
- **Proteggere le informazioni** Le aziende utilizzano diversi meccanismi di sicurezza per la salvaguardia delle informazioni come metodi di controllo degli accessi, firewall, crittografia, sistemi di rilevamento delle intrusioni, ecc. Tuttavia, con nuovi attacchi scoperti ogni giorno è difficile proteggere costantemente le informazioni dell'utente o del sistema.
- **Dare la priorità ai rischi per la sicurezza** L'uso di test di penetrazione come pratica di sicurezza standard non solo aiuta a comprendere i problemi di sicurezza, ma può anche dare la priorità a questi problemi.Questi sforzi possono portare a un'allocazione efficiente del budget per i problemi di sicurezza informatica.
- **Perdita finanziaria** Il test di penetrazione aiuta a mitigare la perdita di ricavi dovuti a tempi di fermo del servizio derivanti da attacchi dannosi. Può anche prevenire o ridurre multe o azioni legali derivanti da pratiche scorrette in materia di sicurezza.

2.2 Vulnerability assessment

Al giorno d'oggi, vi è una notevole confusione tra professionisti del settore e ricercatori riguardo alle differenze significative tra la scansione delle vulnerabilità e il test di penetrazione. Tuttavia, come sottolineato da Imperva [Impa], leader del settore della Cyber Security, i loro esatti significati tecnici e le loro implicazioni sono molto diversi.

Una valutazione delle vulnerabilità identifica e segnala semplicemente le vulnerabilità rilevate, mentre i test di penetrazione tentano di sfruttare le vulnerabilità per determinare se è possibile un accesso non autorizzato o un'attività dannosa. La maggior parte delle intrusioni ad un server avviene tramite lo sfruttamento di vulnerabilità i cui rimedi sono già conosciuti.

In ambito informatico, per Vulnerability Assessment si intende quindi quel processo finalizzato a identificare e classificare i rischi e le vulnerabilità, in termini di sicurezza, dei sistemi informativi aziendali.

Il Vulnerability Assessment è un'analisi di sicurezza che ha come obiettivo l'identificazione di tutte le vulnerabilità potenziali dei sistemi e delle applicazioni valutando il danno potenziale che l'eventuale "attaccante" può infliggere all'unità produttiva.

Queste attività hanno lo scopo di scovare all'interno o all'esterno di un'organizzazione gli eventuali errori di programmazione o di errate configurazioni, commessi durante un'installazione o un upgrade dei sistemi informativi. Uno degli aspetti chiave di questa tipologia di analisi è l'isolamento tempestivo delle vulnerabilità evidenziate che potrebbero causare un blocco temporale o una grave perdita di dati.

Un buon strumento di Vulnerability Assessment permette all'utente di avere una situazione aggiornata del livello di sicurezza degli asset IT. Ovviamente, questo è il punto di partenza per ottimizzare tutti gli sforzi di security management.

Esistono diversi tipi di valutazioni della vulnerabilità. Questi includono:

- **Valutazione dell'host** la valutazione dei server critici, che possono essere vulnerabili agli attacchi se non adeguatamente testati o non generati da un'immagine della macchina testata.
- **Valutazione di rete e wireless** La valutazione di politiche e pratiche per prevenire l'accesso non autorizzato a reti pubbliche o private e risorse accessibili in rete.
- **Valutazione del database** La valutazione di database o sistemi di big data per vulnerabilità e configurazioni errate, identificazione di database non autorizzati o ambienti di sviluppo/test non sicuri e classificazione dei dati sensibili nell'infrastruttura di un'organizzazione.
- **Scansioni delle applicazioni** L'identificazione delle vulnerabilità di sicurezza nelle applicazioni Web e del loro codice sorgente mediante scansioni automatiche sul front-end o analisi statica/dinamica del codice sorgente.

Il processo di scansione della sicurezza si compone di quattro fasi: test, analisi, valutazione e rimedio. In figura 2.2 una rappresentazione del processo sopra delineato. [Impb]

Figura 2.2: Fasi di una vulnerability assessment

- **1. Vulnerability identification (testing)** L'obiettivo di questo passaggio è redigere un elenco completo delle vulnerabilità di un'applicazione. Gli analisti della sicurezza testano l'integrità della sicurezza di applicazioni, server o altri sistemi scansionandoli con strumenti automatici o testandoli e valutandoli manualmente. Gli analisti si affidano anche a database di vulnerabilità, annunci di vulnerabilità di fornitori, sistemi di gestione delle risorse e feed di intelligence sulle minacce per identificare eventuali debolezze nella sicurezza.

- **2. Vulnerability analysis** In questa fase, lo scopo è quello di identificare l'origine e la causa principale delle vulnerabilità emerse nel passaggio uno.
  Implica l'identificazione dei componenti di sistema responsabili di ciascuna vulnerabilità e la principale causa della stessa. Ad esempio, la fonte di una vulnerabilità potrebbe essere una vecchia versione di una libreria open source. Ciò fornisce un percorso chiaro per la riparazione ovvero l'aggiornamento della libreria.

- **3. Risk assessment** L'obiettivo di questo passaggio è la definizione delle priorità delle vulnerabilità. Coinvolge gli analisti della sicurezza che assegnano un punteggio o un punteggio di gravità a ciascuna vulnerabilità, in base a fattori quali:
  - Quali sistemi sono interessati.
  - Quali dati sono a rischio.
  - Quali funzioni aziendali sono a rischio.
  - Facilità di attacco o compromissione.
  - Gravità di un attacco.
  - Potenziali danni a causa della vulnerabilità.

- **4. Remediation** L'obiettivo di questa fase è colmare le lacune di sicurezza. In genere si tratta di uno sforzo congiunto del personale addetto alla sicurezza, dei team di sviluppo e operativi, che determinano il percorso più efficace per la riparazione o la mitigazione di ciascuna vulnerabilità.
  Le fasi di riparazione specifiche potrebbero includere:
  - Introduzione di nuove procedure, misure o strumenti di sicurezza.
  - L'aggiornamento delle modifiche operative o di configurazione.

- Sviluppo e implementazione di una patch di vulnerabilità.
La valutazione delle vulnerabilità non può essere un'attività una tantum.
Per essere efficaci, le organizzazioni devono rendere operativo questo processo e ripeterlo a intervalli regolari. È inoltre fondamentale promuovere la cooperazione tra i team di sicurezza, operativi e di sviluppo, un processo noto come DevSecOps.

Prima di effettuare una scansione alla ricerca di possibili vulnerabilità, dobbiamo conoscere i livelli di gravità e l'intervallo di punteggio CVSS (Common Vulnerability Scoring System) corrispondente utilizzati in tutto il documento per valutare la vulnerabilità e l'impatto del rischio. Nella tabella sottostante 2.1, la classificazione dei rischi in base al punteggio.

Criticità	**CVSS**	**Definizione**
Critical	9.0-10.0	Lo sfruttamento è semplice e di solito comporta un compromesso a livello di sistema. Si consiglia di formare immediatamente un piano d'azione e di patch.
High	7.0-8.9	Lo sfruttamento è più difficile ma potrebbe causare privilegi elevati e potenzialmente una perdita di dati e tempi di inattività. Si consiglia di formare un piano d'azione e patch il più presto possibile.
Moderate	4.0-6.9	Esistono vulnerabilità ma non sono sfruttabili o richiedono passaggi aggiuntivi come il social engineering. Si consiglia di formare un piano di azione e patch dopo che i problemi ad alta priorità sono stati risolti.
Low	0.1-3.9	Le vulnerabilità non sono sfruttabili ma ridurrebbero la superficie di attacco di un'organizzazione. Si consiglia di formare un piano di azione e patch durante la successiva finestra di manutenzione.
Informational	N/A	Non esiste alcuna vulnerabilità. Ulteriori informazioni vengono fornite per quanto riguarda gli elementi rilevati durante i test, i controlli efficaci e la documentazione aggiuntiva.

Tabella 2.1: Tabella CVSS

2.3 Automated tools

In questa sezione andiamo a delineare le caratteristiche degli strumenti automatici per il *penetration testing*. Nello specifico vogliamo individuare come funzionano e, soprattutto, cosa non possono fare.

Prima di tutto, la "consegna" del pentest viene eseguita da un agente o da una *virtual*

machine (VM), che simula efficacemente il laptop del pentester e/o il proxy di attacco che si collegano alla rete. Il pentesting bot eseguirà quindi la ricognizione sul suo ambiente allo stesso modo in cui i pentesters umani eseguono una scansione delle vulnerabilità con il loro strumento preferito o solo una scansione delle porte e dei servizi con Nmap o Masscan. Una volta stabilito dove si trovano nell'ambiente, filtreranno ciò che hanno trovato, ed è qui che finiscono le loro somiglianze con gli scanner di vulnerabilità.

Gli scanner di vulnerabilità elencheranno semplicemente una serie di vulnerabilità e potenziali vulnerabilità che sono state trovate senza alcun contesto per quanto riguarda la loro sfruttabilità. Rigurgiteranno poi riferimenti CVE e punteggi CVSS. In determinati casi aggiungono le "prove" della vulnerabilità del sistema ma esse non si adattano bene ai falsi positivi.

Gli strumenti di penetration test automatizzati sceglieranno quindi da questo elenco di obiettivi il target "migliore", prendendo decisioni basate sulla facilità di exploit, rumore e altri fattori simili. Ad esempio, se l'obiettivo è stato presentato con una macchina Windows che era vulnerabile a EternalBlue, potrebbe favorire questo rispetto alla forzatura bruta di una porta SSH aperta che si autentica con una password poiché è una quantità nota e molto più veloce/facile da sfruttare.

Lo strumento, una volta che ha preso piede, si propagherà attraverso la rete, imitando il modo in cui lo farebbe un pentester o un attaccante. L'unica differenza è che in realtà installa una versione del proprio agente sulla macchina sfruttata e continua il suo perno da lì (c'è una varietà nei modi in cui i diversi providers lo fanno).

A questo punto ricomincia il processo da zero, ma questa volta si assicurerà anche di indagare in modo forense sulla macchina su cui è atterrato per fornirgli più "munizioni" per continuare il suo viaggio attraverso la rete. Eseguirà poi il dump degli hash delle password, se possibile, o cercherà le credenziali hardcoded o le chiavi SSH; le aggiungerà al suo repertorio per il prossimo round della sua espansione. Quindi, mentre in precedenza potrebbe aver semplicemente ripetuto la scansione/exploit/pivot, questa volta tenterà un attacco pass-the-hash o tenterà di connettersi a una porta SSH utilizzando la chiave che ha appena rubato. Ricomincia da qui il suo cammino ciclico.

Il processo sopra descritto presenta numerose somiglianze con il comportamento di un pentester umano, infatti si noti come per la gran parte è esattamente il modo in cui si comportano i pentester e, in misura minore, gli aggressori. I set di strumenti sono simili e le tecniche ed i vettori utilizzati per eseguire il pivot sono identici sotto diversi punti di vista.

2.4 Automated vs Manual

L'atto di automazione offre alcuni vantaggi rispetto alla vecchia metodologia pentesting e all'altrettanto caotica metodologia crowdsourcing. La velocità del test e dei rapporti è molto più veloce e i rapporti sono in realtà sorprendentemente leggibili (dopo la verifica con alcuni QSA, supereranno anche i vari requisiti di pentesting PCI DSS).

Niente più giorni o settimane di attesa per un rapporto redatto manualmente e sottoposto ad alcuni cicli di controllo qualità prima di essere consegnato. Uno dei principali

punti deboli dei test umani è l'obsolescenza quasi immediata poiché, non appena consegnati, molti rapporti di pentesting diventano obsoleti. Ciò accade in quanto l'ambiente dove il test è stato effettuato viene aggiornato più volte. A seguito dell'aggiornamento potrebbero insorgere potenziali vulnerabilità e configurazioni errate non presenti al momento del pentest. Questo è il motivo per il quale il pentesting tradizionale è più simile a un'istantanea della posizione di sicurezza in un determinato momento.

Gli strumenti di penetration test automatizzati aggirano questa limitazione essendo in grado di eseguire test ogni giorno o ad ogni modifica e fornire un report quasi istantaneo.

Un altro vantaggio è il punto di ingresso. A un pentester umano può essere assegnato un punto di ingresso specifico nella rete, mentre uno strumento di pentest automatico può eseguire lo stesso test più volte da diversi punti di ingresso per scoprire vettori vulnerabili all'interno della rete e monitorare vari scenari di impatto. Sebbene ciò sia teoricamente possibile per un essere umano, richiederebbe un enorme impiego di risorse a causa del dover pagare ogni volta per un test diverso.

Due sono i principali lati negativi:
1. Gli strumenti di penetration test automatizzati non comprendono affatto le applicazioni web. Sebbene rilevano qualcosa come un server web a livello di porte/servizi, non individuano quale sia la vulnerabilità IDOR (Insecure Direct Object Reference) nella tua API interna o un SSRF (Server Side Request Forgery) in una pagina web interna. Questo perché lo stack web moderno è complesso e, in virtù del vero, anche gli scanner specializzati (come gli scanner di applicazioni web) hanno difficoltà a rilevare le vulnerabilità di ampia portata (come XSS o SQLi).

2. È possibile utilizzare solo strumenti di pentest automatici "all'interno" della rete poiché l'infrastruttura aziendale più esposta è basata sul Web e gli strumenti di pentesting automatizzati non lo capiscono. Per quanto concerne il pentesting "dall'esterno" è comunque necessario attenersi a un buon pentester umano.

2.5 Cos'è meglio?

Il test di penetrazione automatizzato è molto più veloce, efficiente, facile, affidabile e verifica automaticamente la vulnerabilità e il rischio di una macchina. Questa tecnologia non richiede alcun ingegnere esperto, ma può essere gestita da chiunque abbia la minima conoscenza di questo campo.
Tuttavia, la tabella seguente (tabella 2.2) illustra la differenza fondamentale tra il test di penetrazione manuale e automatico:

Manual penetration testing	**Automatic penetration testing**
Richiede un esperto per eseguire il test	Automatico, anche uno studente della materia può eseguire il test
Richiede diversi strumenti per il test.	Ha strumenti integrati, nessuna dipendenza dall'esterno.
I risultati possono variare da test a test.	Ha risultato fisso.
È stancante e richiede tempo.	È più efficiente e veloce
Ha ulteriori vantaggi, ovvero se un esperto esegue un pentest, può analizzare meglio, può pensare come un hacker e capire dove può attaccare. Quindi, può adattare la sicurezza di conseguenza	Non può analizzare la situazione.
Un esperto può eseguire più tipologie di test.	I test che esegue sono sempre gli stessi.
Per condizioni critiche è più affidabile.	Meno affidabile in condizioni particolari o critiche

Tabella 2.2: Confronto tra pentesting

Possiamo quindi capire che non esiste, almeno per ora, una modalità di pentesting che sia migliore in assoluto.
Mentre gli strumenti automatici possono essere più affidabili in generale, in condizioni particolari il pensiero malleabile e saggio di un pentester esperto potrebbe essere più utile rispetto ad un software.
In compenso uno strumento così utile nelle mani di una persona esperta è sicuramente il modo più efficiente di scovare vulnerabilità e rendere software più sicuri.

3. Studio di tools automatici

In questo capitolo analizziamo alcuni tools per il pentesting presenti nel mercato della "*Cyber Security for industries*" [3.1] ed alcuni strumenti open-source che svolgono lo stesso compito [3.3].

Nel paragrafo 3.1 analizziamo solo superficialmente i tools a pagamento poiché, essendo pensati per le grandi realtà, i costi sono elevati ed i periodi di prova che forniscono sono di esclusivo accesso alle aziende interessate al loro acquisto. Cerchiamo dunque di fornire una panoramica delle principali funzionalità di cui dispongono senza poter dare una dimostrazione del loro utilizzo.

Il focus principale dell'elaborato è incentrato sull'analisi degli strumenti open-source e sul testing delle loro potenzialità.

Prima di addentrarci nell'analisi dei software di "libero accesso", nel paragrafo 3.2 descriviamo l'ambiente di testing utilizzato durante la sperimentazione dei programmi. Segue quindi una presentazione di *Kali Linux*, il sistema operativo specializzato nella cyber security ed utilizzato per tale scopo. Entriamo poi nel dettaglio di come il linguaggio di programmazione "Python" sia il più utilizzato in questo campo, soffermandoci sulle motivazioni del più elevato tasso di impiego, per poi fornirne una panoramica generale. Nel terzo sotto-paragrafo parliamo di Metasploitable, la macchina virtuale target dei penetration test eseguiti. Concludiamo il capitolo prendendo in considerazione tre tools open source: Sn1per, Legion e Xerror.

3.1 Cyber security for industries

Nel capitolo 1 abbiamo già trattato di come la sicurezza informatica abbia acquisito una grande fetta di mercato nella sfera IT.
Come illustrato nella figura 1.2, *Statista* prevede che nel 2023 raggiungerà circa 250 miliardi di dollari, una cifra difficile da trascurare.
Questo settore, che vede continui sviluppi, è caratterizzato da molti competitors ma non sembra essere dominato da imprese leader. Molte aziende che dichiarano di offrire applicazioni di pentesting automatico, in realtà si rivelano piattaforme per il vulnerability assessment con esperti in cyber security che forniscono servizi di pentesting manuale. Sembra infatti che le tecnologie che si celano dietro al penetration testing automatico non siano ancora abbastanza mature per essere messe in commercio.

Solo uno strumento, tra quelli presi in esame, è uno strumento completamente automatico per il pentesting, si tratta di *PenTera* dell'azienda *PcySys*.

Data la mancanza di prodotti necessari per poter fare un confronto tra i tools commerciali, esamineremo comunque altri due software utilizzati nell'ambito della cyber security: Faraday, un programma che fornisce un servizio di vulnerability management, ed Intruder, una piattaforma online "automatizzata" per il penetration testing.

Di seguito andiamo a fornire una visione d'insieme della differenza tra alcuni software per il vulnerability assessment ed uno specializzato in pentesting automatico.

3.1.1 Faraday

Faraday [Kalb] è un software dedicato al *Vulnerability management*. Questo framework è integrato in tutti i sistemi operativi Kali Linux dalla versione 2017.1 del sistema operativo e successive.
Progettato per la distribuzione, l'indicizzazione e l'analisi dei dati generati durante un audit di sicurezza, Faraday introduce un nuovo concetto nel campo della cyber security: IPE (Integrated Penetration-Test Environment) ovvero un IDE di Penetration test multiutente.

Il software è ben distante dagli strumenti di penetration testing automatici. Si tratta di un framework dove poter registrare le vulnerabilità relative ad un determinato workspace e ricevere una rappresentazione grafica intuitiva della situazione di uno specifico host o di una determinata rete. Faraday fornisce un ambiente di pentesting dove più utenti possono contribuire ad analizzare un determinato target e li assiste nell'organizzazione di attacchi di penetration testing.

La versione di Faraday installata in Kali è la distribuzione gratuita del software destinata ai singoli pentester ma, pagando un prezzo che varia in base alla versione, sono disponibili anche le licenze Professional o Corporate. I software premium sono destinati ad aziende che forniscono consulenza sulla sicurezza informatica e permettono una comunicazione immediata con i clienti per la sollecitazione al fixing di vulnerabilità.

Le funzionalità del software di "prova" installato in Kali sono molto ridotte rispetto alle altre versioni ma sono comunque lo stretto necessario per fornire un esempio di

quale sia il *workflow* al suo interno. Nell'immagine 3.1 il terminale all'avvio di Faraday.

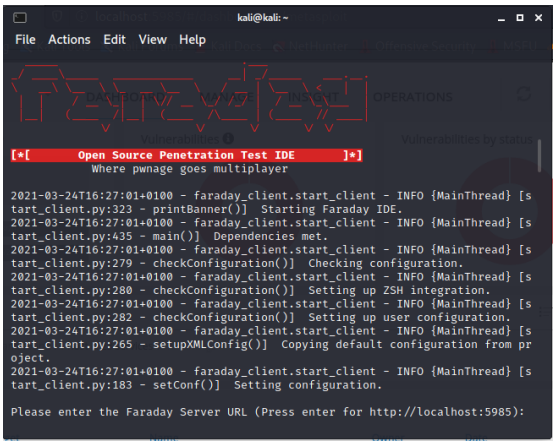

Figura 3.1: Terminale all'avvio di Faraday

Dopo aver impostato Faraday con username e password ed inserito l'URL del server avviato in precedenza, il software ci presenta due diverse schermate: un web client che ci mostra le situazioni di vulnerabilità nei vari workspaces e le attività svolte al loro interno ed un client dove possiamo aprire diversi terminali riferiti al workspace in cui ci troviamo.

Le immagini (3.2 e 3.3) mostrano il risultato dell'utilizzo di un popolare strumento per il mapping delle porte di un client: Nmap.

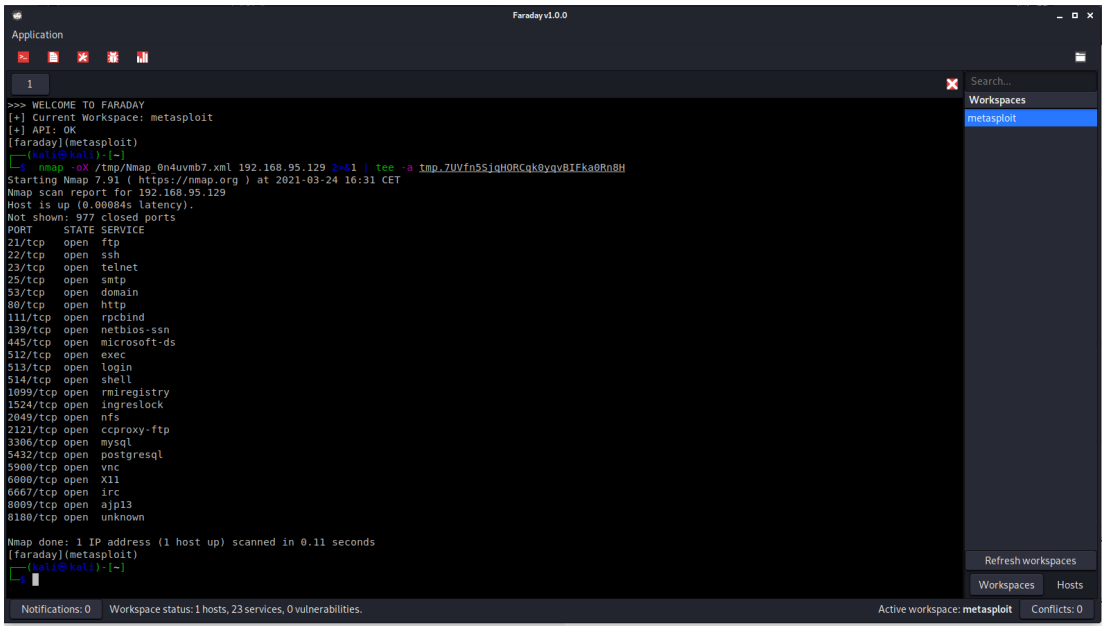

Figura 3.2: Client faraday al termine di Nmap

Nella figura 3.3 abbiamo inoltre associato una vulnerabilità di alta importanza al report che ci ha fornito Nmap riguardante la porta ":5432" associata al server PostgreSQL di Metasploitable.

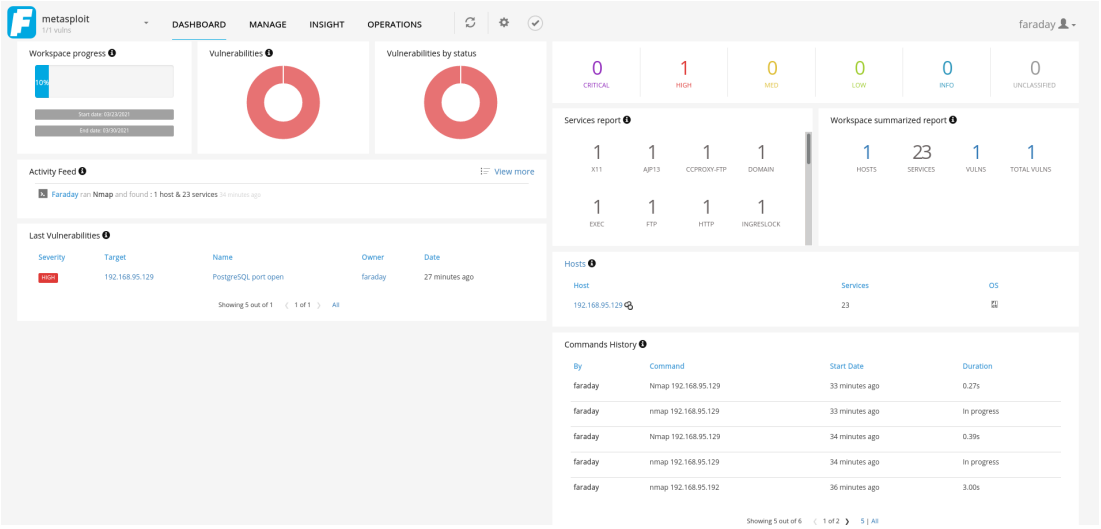

Figura 3.3: Web client di Faraday

Per il penetration testing Faraday offre veramente poco all'hacker, nella versione base il software permette solo di tener conto delle credenziali che sono state ottenute nel corso del processo. Il programma si conferma quindi essere una semplice IDE in cui il pentester deve esclusivamente affidarsi alle proprie capacità e risorse per ottenere informazioni e fare breccia nei suoi target.

3.1.2 Intruder

Intruder è un software per la network security basato su cloud. L'app utilizza lo stesso motore di scansione di cui si servono le grandi istituzioni finanziarie, pertanto è ideale per le grandi aziende che necessitano di una sicurezza più solida senza alcuna complessità aggiuntiva.

La differenza principale tra il resto dei tools per il vulnerability assessment ed Intruder è che al cliente viene fornita esclusivamente l'interfaccia che mostra le vulnerabilità dei propri sistemi. L'utente che utilizza questo servizio viene infatti connesso, 24 ore su 24, ad un server in remoto che periodicamente esegue uno scanner delle vulnerabilità. Un punto forte di Intruder è infatti l'essere sempre aggiornato con gli exploit più recenti (essendo *cloud based*). L'immagine 3.4 mostra la dashboard di Intruder che comunica al cliente la presenza di diverse vulnerabilità e l'andamento dei test periodici nel tempo.

Figura 3.4: L'interfaccia principale di Intruder

Di seguito un dettaglio su cosa questo strumento, classificabile come Automated Vulnerability Assessment tool, fornisce all'utente nell'ambito del penetration testing.
Il servizio offerto da questa azienda permette di organizzare degli interventi di pentesting su richiesta e mette a disposizione il "più alto standard di eccellenza dell'industria, con professionisti della sicurezza altamente qualificati". Di seguito (Figura 3.5) i vari tipi di pentesting offerti dal team di Intruder nella loro pagina Web [Int].

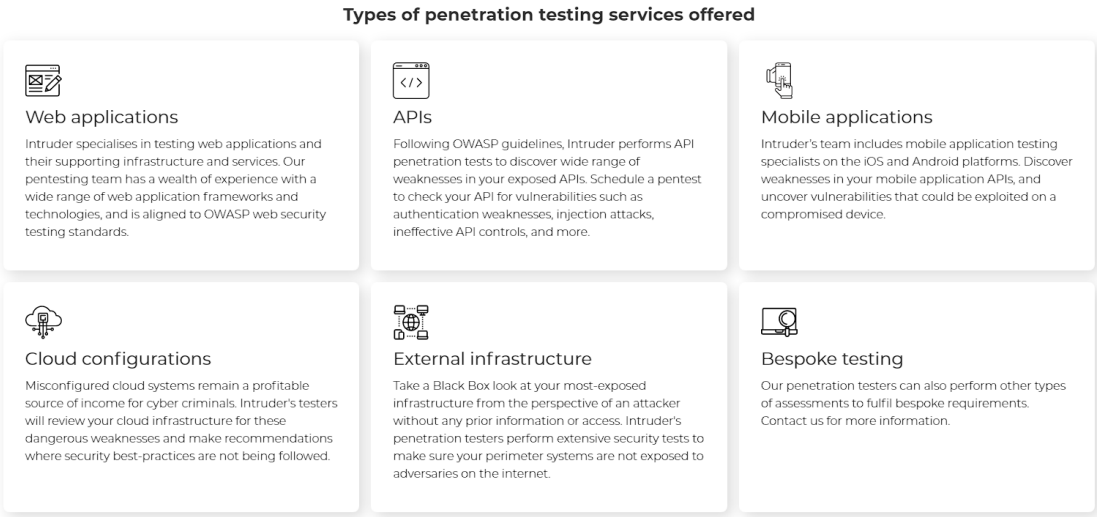

Figura 3.5: I tipi di pentesting offerti da Intruder

Notiamo quindi che anche una delle aziende più importanti nel panorama della cyber security (partner di Google), si affida al penetration testing eseguito manualmente.

3.1.3 PenTera

PenTera è il primo strumento automatico di penetration testing preso in considerazione in questo elaborato. Lo strumento sviluppato dall'azienda israeliana PcySys, vincitore di numerosi premi per l'innovazione tecnologica in ambito pentesting, ha riscosso grande successo nelll'ambito della sicurezza informatica.

Citando la pagina principale del loro prodotto: "La piattaforma automatizzata per il pentesting di Pcysys conduce continuamente exploits etici sulle vulnerabilità dell'infrastruttura, per fornire alle debolezze del sistema delle priorità basate sui risultati ottenuti nei test."

Il software non è disponibile in forma gratuita né in forma open source pertanto non è stato possibile analizzarne i meccanismi per comprendere quale sia il suo funzionamento. La società Deloitte ("azienda di consulenza e revisione, prima nel mondo in termini di ricavi e numero di professionisti" [Wik]), che utilizza la piattaforma di PcySys per i suoi servizi di consulenza, rende disponibile un breve video dove mostra il funzionamento del software. Servendoci di tale dimostrazione [Pcy], di seguito un'analisi sulle funzionalità del software.

Il primo passo riguarda la creazione di un nuovo template, ovvero la configurazione del range di indirizzi IP che si vuole esaminare, vedi Figura 3.6.

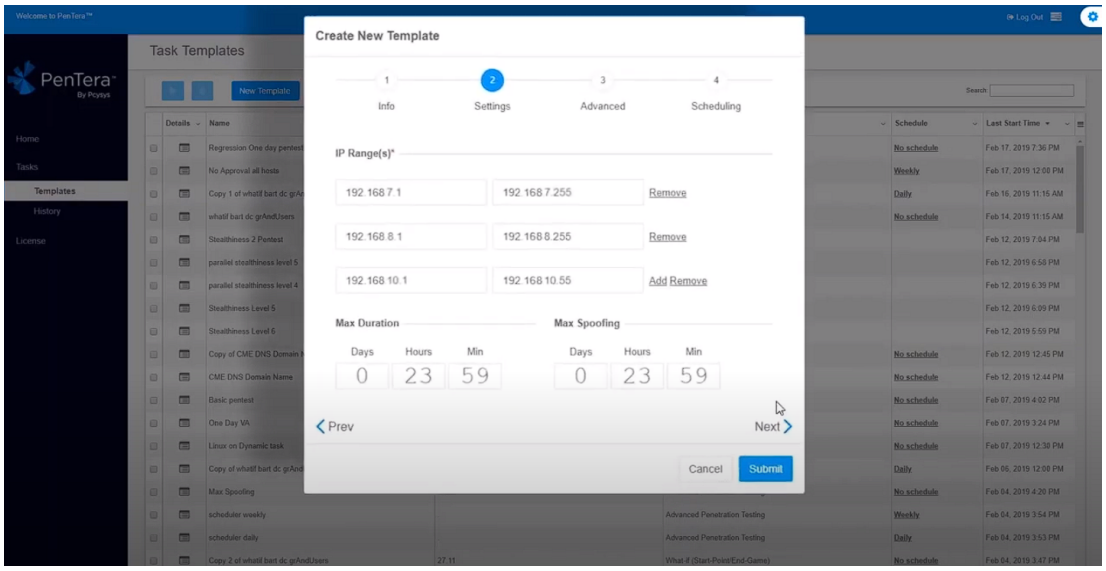

Figura 3.6: Configurazione range IP

Successivamente il software permette di scegliere il livello di furtività dell'attacco: si può scegliere un approccio più silenzioso rispetto ad uno più facilmente rintracciabile. La scelta andrà ad influire sulle metodologie di pentesting che il software adotterà negli attacchi verso i vari target. Il suddetto step in figura 3.7.

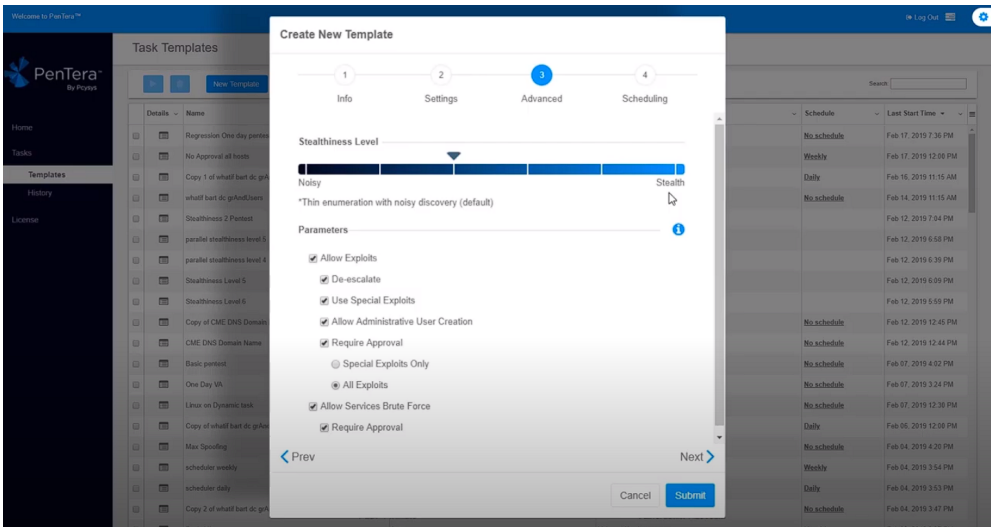

Figura 3.7: Configurazione dell'attacco

PenTera permette poi di schedulare i vari attacchi nel tempo, decidendo gli orari e le giornate in cui effettuarli. Questa funzione permette di stabilire se le misure di sicurezza adottate siano durevoli nel tempo. La facilità e l'efficienza con la quale un software di pentesting automatico esegue le proprie pratiche rende la schedulazione un punto di forza rispetto al penetration testing manuale.

Dopo aver configurato il nostro attacco, PenTera inizia subito una ricognizione di tutti i dispositivi che si trovano all'interno del range precedentemente impostato ed effettua uno scan delle vulnerabilità nella rete stessa. In riferimento la figura 3.8

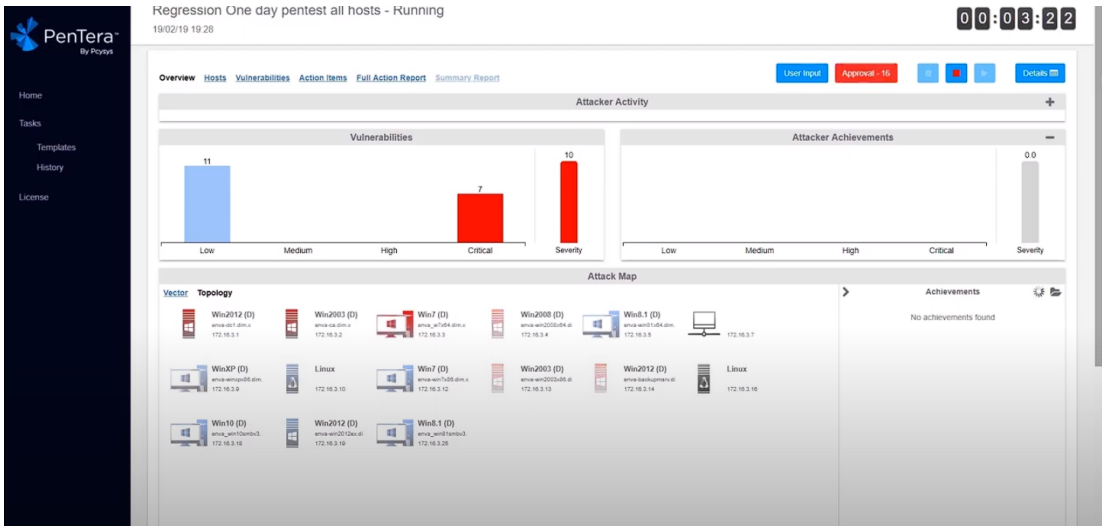

Figura 3.8: Ricognizione e vulnerability assessment della rete

A questo punto terminano le similarità con gli strumenti che abbiamo precedentemente considerato. La piattaforma incomincia con il vero e proprio *ethical hacking*. Le tecniche per il pentesting sviluppate dal team di ricerca PcySys, unite a quelle tradizionali, servono per ottenere più informazioni possibili dai target. Il software mostra quindi nella dashboard, in tempo reale, tutti gli "achievements" ottenuti tramite i diversi attacchi ed assegna un voto da 1 a 10 in base a quanto esso sia critico per il

cliente. Ad esempio, nella figura 3.9, in basso a destra, si può osservare come il test che ha permesso di ottenere la password leggibile dell'amministratore sia valutato 10 per il rischio di sicurezza della rete.

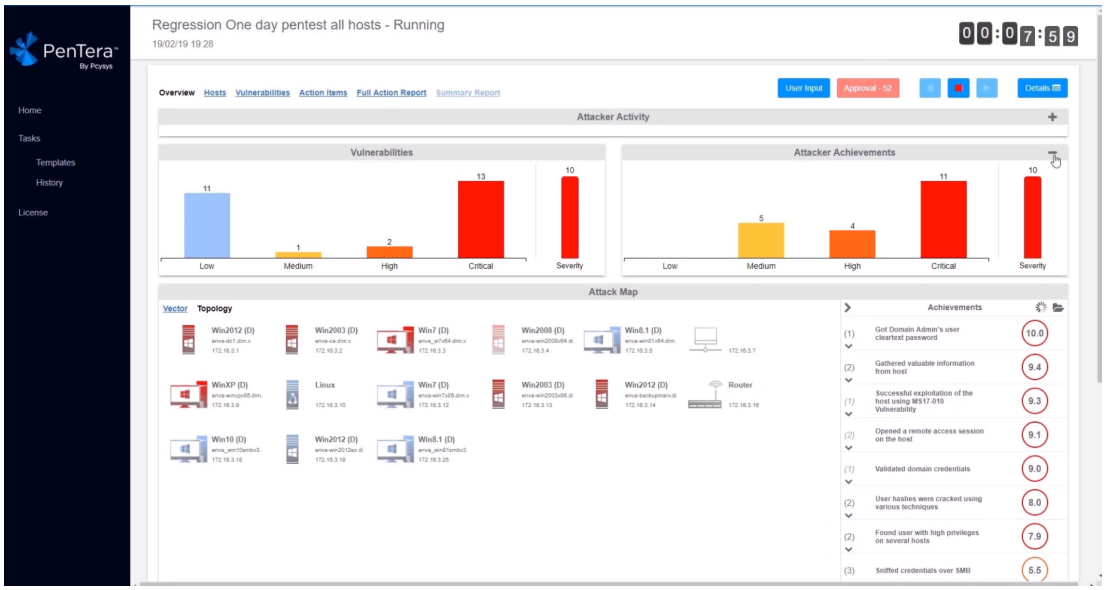

Figura 3.9: Risultati degli attacchi alla rete

Al termine della sequenza di attacco, PenTera fornisce diversi tipi di visualizzazione delle vulnerabilità scoperte e dei dati ottenuti.
Il programma mostra, passo per passo, tramite un grafico vettoriale, come si siano raggiunti determinati achievements e quali vulnerabilità si siano sfruttate per raggiungerli. Genera un dettagliato report dei suddetti achievements suggerendo le azioni necessarie per rimediare alle criticità che li hanno causati.
Nelle figure 3.10 e 3.11 gli esempi forniti dall'azienda Deloitte.

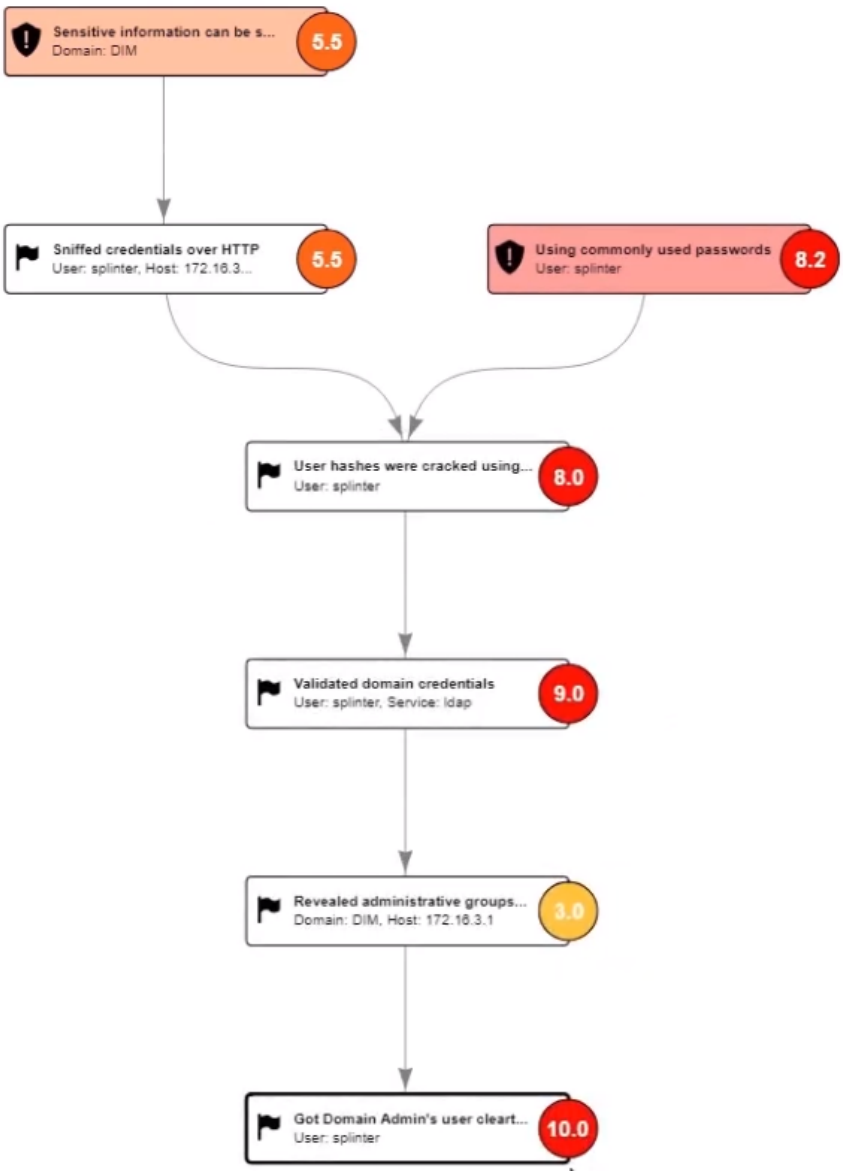

Figura 3.10: Grafico di un achievement

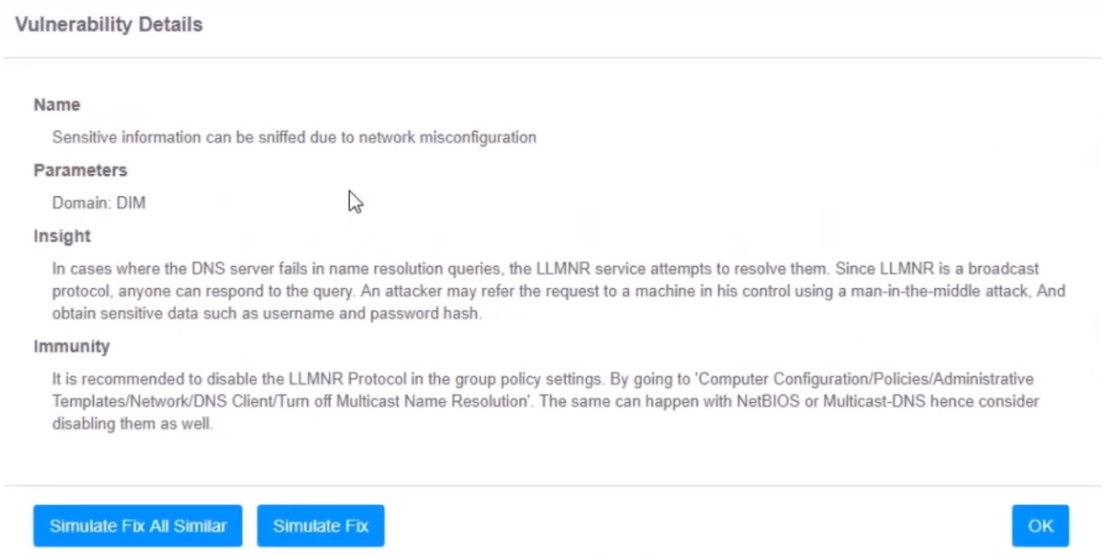

Figura 3.11: Report fornito per un achievement

Possiamo concludere che PenTera sia un valido strumento automatico per il penetration testing. Possiede infatti tutte le funzionalità che un tool automatico, per essere efficiente ed efficace, dovrebbe avere. Esso si interfaccia con l'utente in maniera impeccabile con grafici e report molto intuitivi.

Le differenze tra i software da noi presi in esame (Faraday, Intruder e PenTera) sono sostanziali. Nei primi due casi, utilizzando Faraday ed Intruder, possiamo solamente eseguire una ricognizione dei dispositivi che abbiamo come target mentre il pentesting è lasciato in mano a persone esperte del settore. PenTera invece fornisce all'utente non solo una visione completa di quanto i sistemi siano vulnerabili ma produce un output di facile interpretazione anche per i meno qualificati.
L'enorme complessità che risiede all'interno di un software così elaborato genera uno strumento che ha un importante potenziale commerciale e che migliora, sotto diversi punti di vista, ciò che un approccio manuale potrebbe fornire.

Da sottolineare che PenTera è il primo software per il pentesting automatico a riscuotere un successo di vasta portata in virtù dei molteplici vantaggi. Ciò lascia immaginare che strumenti ad esso assimilabili saranno oggetto di un interesse crescente in un futuro prossimo.

D'altro canto, a fronte dei vantaggi forniti da uno strumento così all'avanguardia e completo, supponiamo un prezzo più elevato per la prestazione offerta. Si tratta di una supposizione in quanto i prezzi effettivi dei software dipendono strettamente dalla grandezza della rete da analizzare. Pertanto è in mano al cliente un'analisi in termini di costi-benefici sulla base delle proprie disponibilità.

3.2 Ambiente del testing

Nel paragrafo precedente abbiamo solamente descritto le funzionalità dei programmi presi in esame utilizzando esempi trovati nel web. Nel paragrafo 3.3 invece, andremo ad analizzare dei software sviluppati in repositories open source dalla comunità della sicurezza informatica.

Prima di iniziare ad esporre il processo di analisi degli strumenti presi in esame è utile descrivere l'ambiente in cui sono stati utilizzati.

3.2.1 Kali Linux

L'intera comunità della sicurezza informatica citata precedentemente, è una parte integrante e fondamentale del settore.
La cyber security è un'attività che proviene dal "basso" poiché nasce per contrapporsi all'hacking malevolo. A dimostrazione di ciò, nella sicurezza informatica viene fatto un utilizzo degli stessi mezzi di hacking per raggiungere risultati diversi.
Basti pensare che, chi lavora per aiutare gli altri (pentester) viene chiamato White Hat Hacker, mentre chi utilizza le stesse tecniche in modo illegale è chiamato Black Hat Hacker.
La community della sicurezza informatica raccoglie i contributi di entrambe le "fazioni" che partecipano a questa battaglia. Infatti, grazie a questo contrasto, sono nati numerosi strumenti dedicati al settore della sicurezza. Uno dei progetti più ambiziosi in questo ambito è Kali Linux [Kala], un sistema operativo dedicato alla cyber security. In figura 3.12 il logo del software.

Figura 3.12: Logo di Kali Linux

Kali è una distribuzione Linux open source basata su Debian ed orientata a varie attività di sicurezza delle informazioni come Penetration Testing, Security Research, Computer Forensics e Reverse Engineering. Questa "distro" è ottimizzata per l'installazione di software dedicato al pentesting e, per questo motivo, è stata scelta per comporre l'ambiente di testing dell'elaborato.

Come possiamo vedere in figura 3.13, Kali contiene già all'installazione diversi software, come lo stesso Faraday, e Legion, uno dei tools che analizzeremo in seguito.

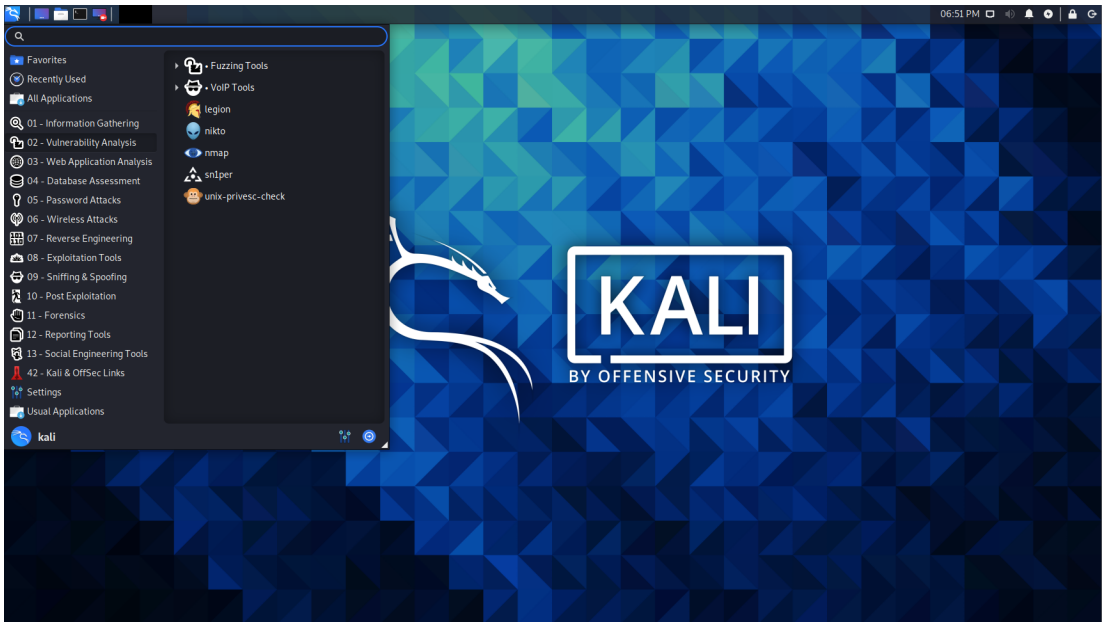

Figura 3.13: Barra delle applicazioni in Kali Linux

Il sistema operativo è stato installato in una macchina virtuale VMWare eseguita da un sistema Windows. La virtualizzazione di Kali ci ha permesso di creare una rete locale all'interno della macchina sulla quale abbiamo effettuato il testing dei diversi strumenti.

Il SO si distingue tra le diverse distro di linux per numerose caratteristiche:

- **Conforme a FHS:** Kali aderisce al Filesystem Hierarchy Standard, consentendo agli utenti Linux di individuare facilmente file binari, file di supporto, librerie, ecc.
- **Supporto di dispositivi wireless ad ampio raggio:** Può supportare il maggior numero possibile di dispositivi wireless, consentendogli di funzionare correttamente su un'ampia varietà di hardware e rendendolo compatibile con numerosi dispositivi USB e altri dispositivi wireless.
- **Kernel personalizzato con patch per l'iniezione:** Il kernel è sempre aggiornato con le patch per l'injection più recenti.
- **Sviluppato in un ambiente sicuro:** Vengono utilizzati i protocolli più sicuri per lo sviluppo del sistema.
- **Completamente personalizzabile:** Kali è personalizzabile in ogni suo singolo componente, anche il kernel può essere cambiato in base alle preferenze dell'utente.
- **Supporto ARMEL e ARMHF:** Poiché i sistemi a scheda singola basati su ARM come Raspberry Pi e BeagleBone Black stanno diventando sempre più diffusi ed economici, Kali Linux è disponibile su un'ampia gamma dei suddetti dispositivi. Kali dispone di repository ARM integrate nella distribuzione principale, pertanto gli strumenti per ARM vengono aggiornati insieme al resto della distribuzione.

Kali è un valido strumento per tutto ciò che concerne la sicurezza informatica, dal social engineering fino al pentesting più tecnico. Il sistema operativo nasconde numerose funzionalità utili in qualsiasi procedura di pentesting, la più curiosa e degna di menzione è quella di kali-undercover, che fornisce un aspetto "Windowsiano" (Figura 3.14) al desktop per non destare sospetti nella fase di "hacking".

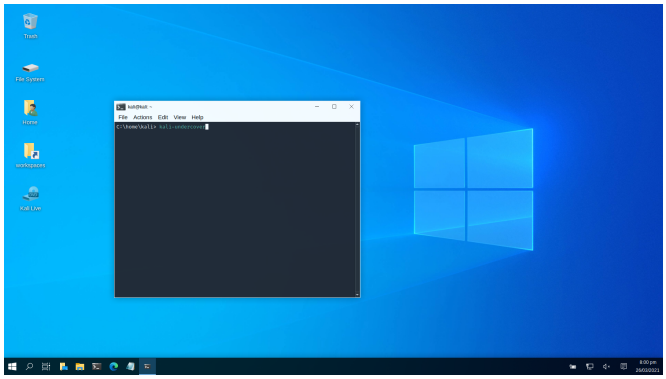

Figura 3.14: Il desktop di Kali "camuffato" in Windows

3.2.2 Python

Diversi software open source sono stati presi in considerazione nella stesura dell'elaborato ed una caratteristica che li accomuna è l'utilizzo del linguaggio Python. Quest'ultimo è un linguaggio di programmazione di più "alto livello" rispetto alla gran parte degli altri linguaggi. È orientato a oggetti ed adatto, tra gli altri usi, a sviluppare applicazioni distribuite, scripting, computazione numerica e system testing.

La forte flessibilità e la semplicità della sintassi che lo caratterizzano lo hanno reso il linguaggio più "popolare" al mondo, sorpassando Java nell'Aprile del 2018. Secondo l'indice PYPL (PopularitY of Programming Language) fornito da GitHub [Pyp] e basato sui dati provvisti da Google Trends, Python è protagonista del 30.17% delle ricerche mondiali, come riportato in figura 3.15.

Worldwide, Mar 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	30.17 %	-0.2 %
2		Java	17.18 %	-1.2 %
3		JavaScript	8.21 %	+0.2 %
4		C#	6.76 %	-0.6 %
5	↑	C/C++	6.71 %	+0.8 %
6	↓	PHP	6.13 %	+0.0 %
7		R	3.81 %	+0.0 %
8		Objective-C	3.56 %	+1.1 %
9		Swift	1.82 %	-0.4 %
10	↑	Matlab	1.8 %	-0.0 %
11	↑	Kotlin	1.76 %	+0.2 %
12	↓↓	TypeScript	1.74 %	-0.1 %
13	↑	Go	1.34 %	+0.0 %
14	↓	VBA	1.22 %	-0.1 %
15		Ruby	1.13 %	-0.1 %
16	↑↑	Rust	1.13 %	+0.5 %
17	↑↑↑↑↑↑	Ada	0.68 %	+0.4 %
18	↓	Visual Basic	0.67 %	-0.3 %
19	↓↓↓	Scala	0.66 %	-0.4 %
20	↑↑↑↑	Lua	0.55 %	+0.2 %

Figura 3.15: Indice di popolarità dei linguaggi

Questo idioma condivide la sua popolarità anche nel settore della sicurezza informatica. Nonostante per eseguire un Penetration Test manuale la conoscenza di un linguaggio di programmazione non sia necessaria, l'automatizzazione sta permettendo al Python di spopolare nella comunità della cyber security.

L'utilità del linguaggio è data da diverse caratteristiche:

- **I professionisti della sicurezza informatica possono mettersi al passo rapidamente.** Python ha generalmente una bassa curva di apprendimento, è diventato il linguaggio di programmazione preferito per coloro che operano nel campo della sicurezza informatica poiché molti di loro hanno conoscenze di programmazione limitate. La facilità d'uso di Python fa sì che un professionista esperto di sicurezza informatica con un forte background tecnico possa iniziare a programmare ed implementare rapidamente il proprio codice.

- **I team di sicurezza informatica possono formarsi rapidamente.** La flessibilità e la facilità d'uso di Python forniscono un grande vantaggio per i responsabili della sicurezza informatica che devono guidare i team e implementare rapidamente i progetti.

- **La vasta libreria di Python comprende già numerosi strumenti di sicurezza informatica.** La vasta libreria dei moduli di Python è un fattore chiave. Python è diventato ben noto e ampiamente utilizzato e, grazie alla sua vasta libreria, professionisti della sicurezza informatica non hanno bisogno di reinventare la ruota con attività comuni. Nella maggior parte dei casi possono trovare rapidamente analisi della sicurezza informatica o strumenti di penetration testing già disponibili.

- **Python può essere utilizzato per quasi tutti gli ambiti della sicurezza informatica.** Con una profonda conoscenza di Python e dei concetti di programmazione in generale, i professionisti possono svolgere praticamente qualsiasi attività di cui hanno bisogno utilizzando il codice Python. Ad esempio, Python è ampiamente utilizzato nell'analisi dei malware, nella scoperta di host, nell'invio e nella decodifica di pacchetti, nell'accesso ai server, nella scansione delle porte e nella scansione della rete, solo per citarne alcuni. Considerando anche l'efficacia di Python nello scripting, per l'automazione delle attività e per l'analisi dei dati, è comprensibile che esso sia sempre più popolare nell'ambito della sicurezza informatica.

- **Gli script in Python possono essere sviluppati rapidamente** Un altro vantaggio di Python che aiuta i professionisti della sicurezza informatica è il poter sviluppare soluzioni di cui hanno bisogno con un tempo minimo e con un codice piuttosto semplicistico. Gli errori nel codice sono più facili da trovare e tutto risulta più reattivo ed immediato.

Le capacità di automazione fornite da questo linguaggio sono quindi essenziali per uno strumento che vuole rendere meccanico il penetration testing.

3.2.3 Metasploitable 2/3

Per concludere la sezione sull'ambiente di testing riportiamo l'analisi delle altre macchine virtuali nella nostra rete locale. Il penetration testing ha bisogno di due attori per essere eseguito: l'attaccante, nel nostro caso la macchina Kali ed il target, una macchina attaccabile in modo sicuro e che ci consente di verificare se il testing ha avuto successo.

I target delle nostre pratiche di penetration testing in questo elaborato sono delle macchine Metasploitable [Ms2]. Metasploitable è un sistema operativo Linux costruito volontariamente per avere numerose vulnerabilità da sfruttare. Lo scopo di questa macchina è quello di fornire una versione di un sistema operativo popolare ed utilizzato sul quale fare pratica senza dover agire su macchine reali.

Le vulnerabilità presenti in Metasploitable sono numerose e di grande varietà. Nel sito di Rapid7 [Ms2], il team di sviluppo che ha creato il sistema, sono elencate le categorie più importanti:

- **Services** Molti network services possiedono porte TCP aperte facilmente accessibili dall'esterno.
- **Backdoors** Alcuni dei server FTP presenti all'interno delle macchine contengono delle popolari backdoor.
- **Unintentional Backdoors** Oltre alle backdoor dannose nella sezione precedente, alcuni servizi sono quasi backdoor per loro stessa natura.
- **Weak Passwords** Metasploitable presenta delle terribili scelte di password degli utenti e dei database. Attacchi brute force potrebbero facilmente ottenere tutte le credenziali dei servizi.
- **Vulnerable Web Services** I servizi web presenti nel sistema sono volontariamente vulnerabili e exploitabili. Alcuni esempi ne sono Multillidae e DVWA, applicazioni web che permettono diversi casi di SQL Injection e cross-site scripting.

Per analizzare il comportamento dei tools, essi vengono testati sulle tre tipologie di sistema che il team Rapid7 fornisce nel proprio sito. La rete locale installata contiene sia Metasplotable 2 che entrambe le versioni di Metasploitable 3.

Metasploitable 2 è il sistema precursore sviluppato in ambiente Linux Ubuntu. Questa versione è di comune utilizzo nell'addestramento sul penetration testing poiché richiede pochissime risorse ed è integrata nella suite di Metasploit, il framework per il pentesting più popolare al mondo. Metasploitable 3 invece si presenta in due tipologie differenti: Ubuntu 14.0.1 e quella Windows 2008, entrambi in versione server.

Il terzo modello della VM, essendo ancora in aggiornamento, contiene degli exploit più recenti ed alcuni più difficili da sfruttare. L'utilizzo di entrambi i software ci permetterà di analizzare come lo strumento preso in questione si comporta di fronte ad exploit più o meno attuali.

Per la virtualizzazione delle macchine "target" abbiamo utilizzato VirtualBox, un software gratuito sviluppato dalla Oracle mentre, per la macchina Kali da cui faremo partire l'attacco, il servizio di virtualizzazione è fornito da VMware Workstation. La scelta di utilizzare il programma firmato Oracle per le macchine Metasploit è stata forzata poiché il building delle stesse viene eseguito tramite Vagrant, un gestore di macchine virtuali open source, che riporta problemi di compatibilità nel trasferimento delle macchine virtuali in VMware.

Per la creazione della rete locale è stato virtualizzato un adattatore Ethernet Host-Only fornendo a VirtualBox l'indirizzo della scheda di rete, l'indirizzo del server dell'host ed il range di indirizzi allocabili ai dispositivi. Nella figura 3.16 la configurazione dell'adattatore.

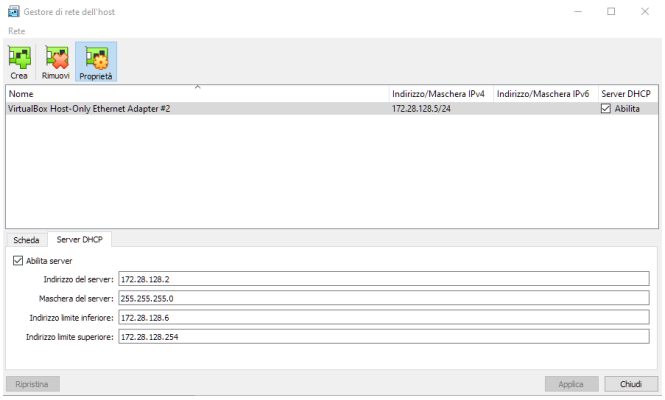

Figura 3.16: Configurazione dell'adattatore di rete.

Successivamente lo stesso adattatore viene collegato alle schede di rete delle macchine virtuali come in figura 3.17. Inoltre è consigliato, per evitare problemi di conflitto, il refreshing degli indirizzi MAC dei diversi host poichè il protocollo DHCP della subnet potrebbe assegnare lo stesso indirizzo a più macchine diverse.

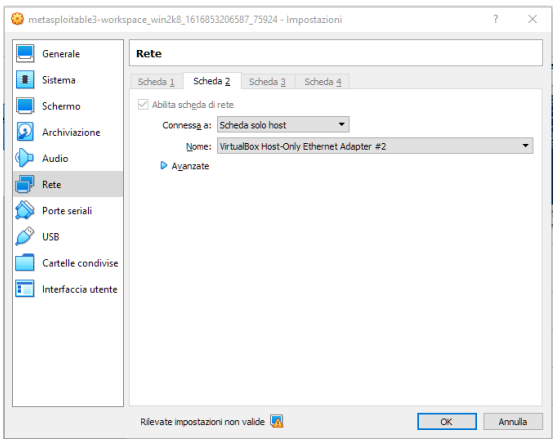

Figura 3.17: Assegnazione della scheda di rete.

A questo punto, non resta altro che collegare la nostra macchina virtuale d'attacco alla rete per terminare la costruzione della subnet locale. Nelle impostazioni del sistema Kali, dobbiamo impostare una connessione bridged tra la rete VMware e quella dell'adattatore Ethernet virtualizzato in precedenza. In figura 3.18 viene mostrato co-

me cambiare la connessione della VM ed in seguito nella figura 3.19 come far si che la connessione bridged standard di VMware si colleghi all'adattatore di VirtualBox.

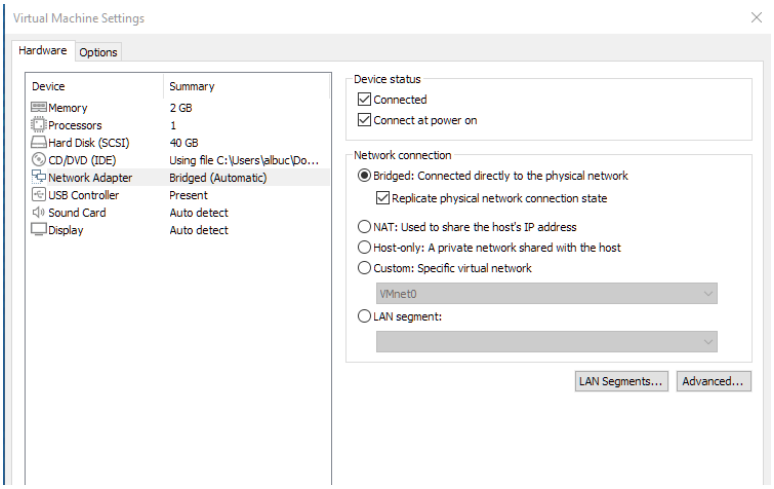

Figura 3.18: Impostazioni della VM

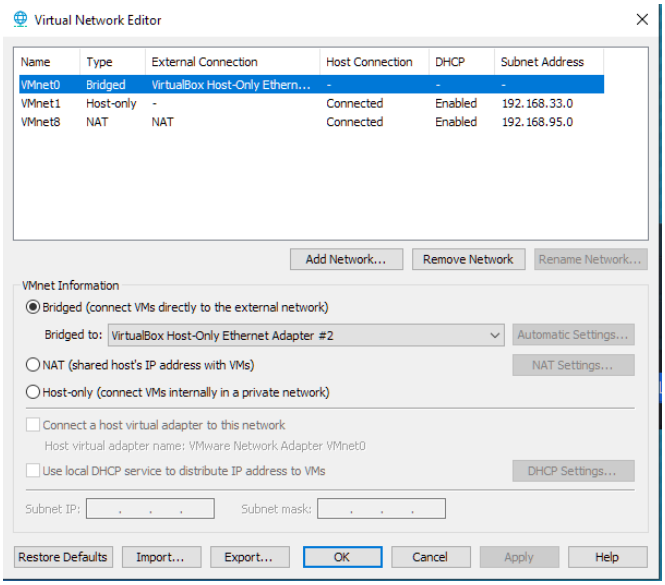

Figura 3.19: Network Editor di VMware.

Utilizzando i comandi "ipconfig" per Windows ed "ifconfig" per Linux, come nelle figure 3.20, 3.21, 3.22 e 3.23, possiamo notare che gli indirizzi IP sono stati assegnati alle macchine virtuali. L'impostazione della rete locale è quindi conclusa con successo.

```
PS C:\Users\vagrant> ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::284a:7994:176f:1c50
   IPv4 Address. . . . . . . . . . . : 172.28.128.6
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
PS C:\Users\vagrant> _
```

Figura 3.20: IP Metasploitable 3 Windows

```
Link encap:Ethernet  HWaddr 08:00:27:3e:15:33
inet addr:172.28.128.8  Bcast:172.28.128.255  Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe3e:1533/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:7397 errors:0 dropped:0 overruns:0 frame:0
TX packets:350 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2045512 (2.0 MB)  TX bytes:61099 (61.0 KB)
```

Figura 3.21: IP Metasploitable 3 Ubuntu

```
sfadmin@metasploitable:~$ ifconfig
th0       Link encap:Ethernet  HWaddr 08:00:27:70:98:91
          inet addr:172.28.128.7  Bcast:172.28.128.255  Mask:255.255.25
          inet6 addr: fe80::a00:27ff:fe70:9891/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6870 errors:0 dropped:0 overruns:0 frame:0
          TX packets:265 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2032628 (1.9 MB)  TX bytes:44660 (43.6 KB)
          Base address:0xd240 Memory:f0420000-f0440000
```

Figura 3.22: IP Metasploitable 2

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ifconfig
th0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 172.28.128.4  netmask 255.255.255.0  broadcast 172.28.128.255
      inet6 fe80::90ca:9dff:fec6:594a  prefixlen 64  scopeid 0x20<link>
      ether 92:ca:9d:c6:59:4a  txqueuelen 1000  (Ethernet)
      RX packets 676829  bytes 171647001 (163.6 MiB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 813884  bytes 96268579 (91.8 MiB)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figura 3.23: IP Kali Linux

3.3 Open source tools

Abbiamo già espresso quanto sia difficile trovare degli strumenti in commercio con lo scopo di automatizzare le tecniche di Penetration Testing. Tale difficoltà risulta ancora più marcata nel mondo open source. Gran parte dei progetti pubblici su piattaforme quali GitHub o GitLab sono, parallelamente agli strumenti commerciali, solamente tools di Vulnerability Assessment. Altri software che condividono lo stesso obiettivo sono ancora in via di sviluppo e quindi inutilizzabili. Un caso tra questi è quello di Xerror, il software di cui parleremo nella sezione 3.3.3. Tra i software open source esaminiamo Sn1per e Legion, due strumenti che, in modo differente, tentano di avvicinarsi ai software professionali come PenTera.

3.3.1 Sn1per

Sn1per [Xer] è uno scanner automatico in grado di automatizzare il processo di raccolta dei dati per l'esplorazione ed i test di penetrazione. Utilizza strumenti noti come amap, arachni, amap, cisco-torch, dnsenum, enum4linux, golismero, hydra, metasploit-framework, nbtscan, nmap smtp-user-enum, sqlmap, sslscan, theharvester, w3af, wapiti, whatweb , whois, nikto e wpscan.d durante un penetration test per enumerare ed analizzare le vulnerabilità.

Sn1per fornisce alcune funzionalità di utility come la suddivisione della azioni in workspace e la possibilità di schedulare i penetration tests. Il programma è interamente da linea di comando e non offre nessun tipo di interfaccia grafica all'utente nella versione gratuita. XeroSecurity, il team di sviluppo di Sn1per, fornisce infatti un modello a pagamento del software chiamato "Sn1per Professional". La versione gratuita del programma è scaricabile dal GitHub del team e possiede molti degli attacchi disponibili in Sn1per premium:

- **REPORT** Restituisce tutti i risultati in formato testo nella directory del "loot" per riferimento futuro.
- **STEALTH** Enumera rapidamente i singoli target utilizzando scansioni per lo più non intrusive per evitare il blocco WAF/IPS.
- **DISCOVER** Analizza tutti gli host su una sottorete/CIDR e avvia una scansione Sn1per su ogni host. Risulta utile per le scansioni delle reti interne.
- **PORT** Esegue la scansione di una porta specifica per le vulnerabilità.
- **FULLPORTONLY** Esegue una scansione dettagliata completa delle porte e salva i risultati in XML.
- **WEB** Aggiunge scansioni di applicazioni Web completamente automatiche ai risultati (solo porta 80/tcp e 443/tcp). È ideale per le applicazioni Web, ma può aumentare notevolmente il tempo di scansione.
- **NOBRUTE** Avvia una scansione completa su un host/dominio di destinazione senza servizi di forzatura bruta.
- **AIRSTRIKE** Enumera rapidamente porte/servizi aperti su più host ed esegue il fingerprinting di base.

- **NUKE** Avvia il controllo completo di più host specificati nel file di testo scelto.
- **LOOT** Organizza e visualizza automaticamente la cartella del "loot" nel browser e apre la GUI di Metasploit Pro e Zenmap con tutti i risultati della scansione delle porte.

Il software fornisce, come sopra dettagliato, diverse modalità di attacco, dalla più furtiva alla più silenziosa.
Durante la fase di analisi, le azioni verso i target sono strutturate in modo da testare i limiti dell'applicazione. Inizialmente abbiamo avviato Sn1per in AIRSTRIKE mode provando le sue funzionalità verso ogni istanza di Metasploit. In seguito abbiamo messo alla prova le macchine virtuali avviando la modalità NUKE, la più "irruente" tra quelle sopra elencate. Nella seguente immagine (3.24) viene mostrato Sn1per appena inizializzato il primo attacco AIRSTRIKE.

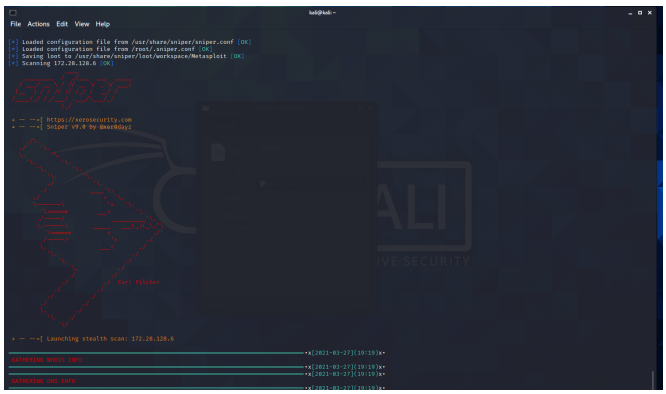

Figura 3.24: Attacco AIRSTRIKE di sn1per

AIRSTRIKE alla fine della sua attività ha svolto un lavoro completo di ricognizione nelle macchine virtuali, riportando tutte le porte aperte, eseguendo screenshot delle schermate dei vari server negli host e riportando il tutto in una cartella appartenente al workspace che abbiamo assegnato all'attacco. Dopo aver osservato i risultati della ricognizione fornita in precedenza, con le stesse impostazioni è stato inizializzato l'attacco NUKE, come si evince in figura 3.25.

Figura 3.25: Attacco NUKE di sn1per

Il nome affidato dal team di XeroSecurity a questo attacco rispecchia l'azione di pentesting eseguita sulle macchine Metasploit. Sn1per utilizza numerose tecniche per tentare di scovare tutte le debolezze del sistema, alcune delle quali molto dispendiose

in termini di tempo. Notevole è stato l'utilizzo di Brutex, il modulo per scovare combinazioni di username e password utilizzando la "forza bruta". Il software ha tentato più di 4000 combinazioni in più di 30 minuti su 4 diverse porte dopo averle trovate libere. In un altro tentativo di exploit, lo strumento è riuscito ad aprire una command shell tramite una backdoor nella porta 21 dedicata al protocollo FTP, rendendo possibile ottenere il completo comando sul sistema. Nella figura 3.26 è mostrato come, eseguendo il comando "ifconfig", il terminale ritorni l'indirizzo IP della macchina metasploitable di cui abbiamo ottenuto il controllo.

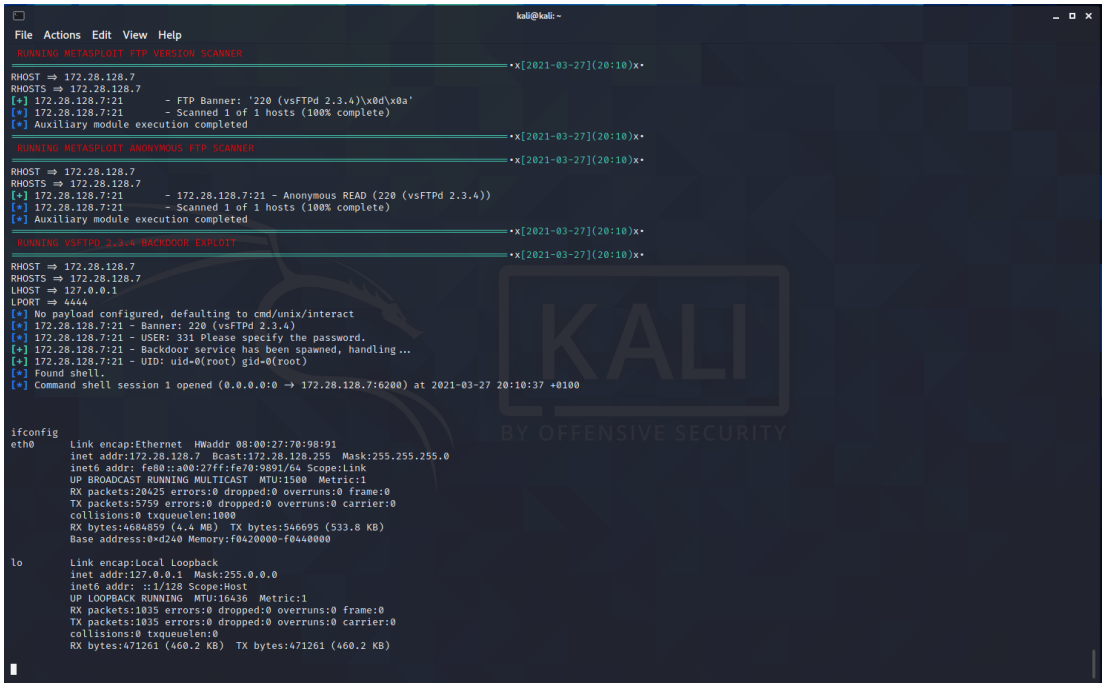

Figura 3.26: Backdoor exploit ed opening della shell.

Al termine del pentest, Sn1per fornisce una cartella all'interno della quale inserisce tutti i report generati dall'attacco. I risultati prodotti dall'applicazione, al termine di un'operazione durata più di due ore, sono in parte sorprendenti ma allo stesso tempo un po' deludenti. Nonostante lo strumento utilizzato abbia fornito diverse informazioni e sia riuscito ad ottenere alcune credenziali (come mostrato in figura 3.27), la maggior parte dei traguardi sono quelli raggiunti attaccando Metasploitable 2, la macchina virtuale con gli exploit più datati. Durante uno dei test eseguiti sulla rete, il programma si è perfino bloccato nel tentativo di stabilire il contatto con una porta chiusa di Metasploitable 3 Ubuntu.

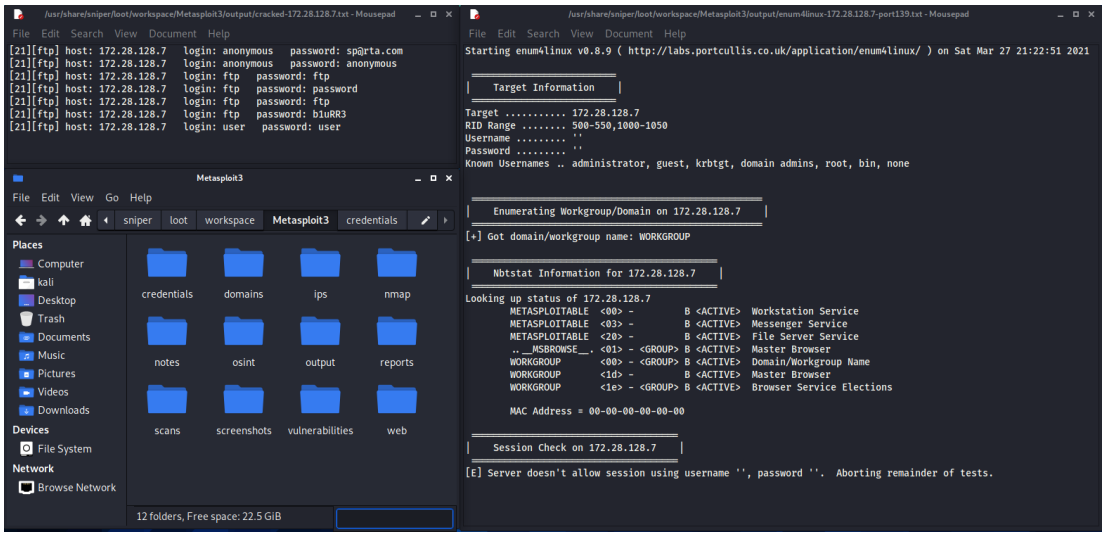

Figura 3.27: Alcune delle informazioni ottenute dall'attacco.

Il software si è dunque rivelato debole di fronte a vulnerabilità più moderne e recenti ma efficace con quelle più riconosciute ed affermate. Questo lato negativo può essere trascurato se pensiamo che quella utilizzata è la versione gratuita ed open source del programma e che quindi potrebbe non essere aggiornata tanto quanto la versione a pagamento. Potrebbe infatti essere concepita come un semplice "banco di prova" ed un incentivo a passare alla versione professionale del software.

3.3.2 Legion

Legion [GoV] è una fork di Sparta, un software sviluppato da SECFORCE che semplifica i network pentetration test assistendo l'utente nella fase di scansione ed enumerazione. Consente al tester di risparmiare tempo avendo accesso point-and-click al suo toolkit e visualizzando tutti gli output degli strumenti in modo conveniente.

Il software da cui Legion si è distaccato è una IDE per il penetration testing che fornisce degli strumenti di scanning ma il programma su cui condurremo i nostri test aggiunge numerose funzionalità al suo toolkit. Esso fornisce un'interfaccia grafica a differenza di Sniper, lo strumento analizzato precedentemente. Legion un framework modulare, che consente di aggiungere o personalizzare funzionalità. È un altro strumento di pentesting scritto in Python, il che significa che può essere eseguito su qualsiasi sistema Windows, MacOS e Linux.

Come già detto in precedenza, Legion è uno dei programmi presenti nella lista dei tools di Kali. L'assenza di meccanismi di installazione ne facilità notevolmente l'utilizzo per chi non eccelle in Linux.

Il primo passo da eseguire in Legion è quello di aggiungere degli host nello scope dell'applicazione ovvero inserire gli indirizzi dei target che si vogliono analizzare/attaccare. In figura 3.28 notiamo inoltre che il software ci consente di scegliere il livello di performance delle azioni che andrà ad eseguire in una scala da "Paranoid" ad "Insane". Di seguito andremo ad analizzare i limiti del programma realizzando test con il più alto livello di performance.

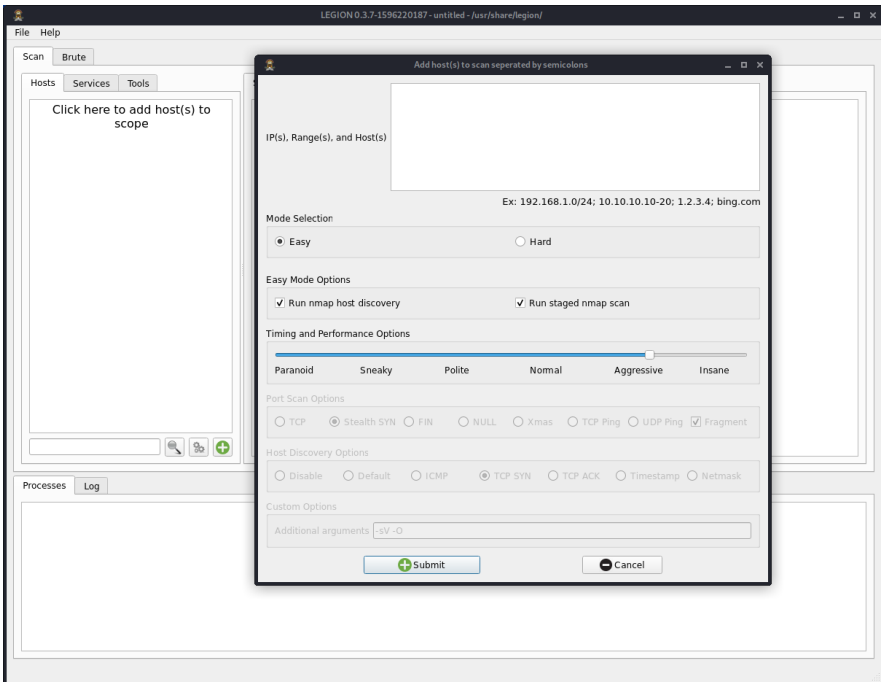

Figura 3.28: L'interfaccia di Legion.

Inseriti gli indirizzi IP dei target, Legion inizia subito con la ricognizione mappando tutte le porte aperte e mostrandole nell'interfaccia mano a mano che vengono scoperte. L'approccio di Legion all'automatizzazione è ben diverso da quello di Sn1per. In quest'ultimo, dopo aver inserito i target dell'attacco, l'utente non deve far altro che aspettare che il software esaurisca tutti i suoi script e strumenti esterni. Nel caso di Legion invece, lo strumento guida chi utilizza questo software a procedere nell'attacco fornendogli tutti i mezzi di cui ha bisogno.

Nonostante questa applicazione non svolga il processo di pentesting in maniera completamente automatica, ciò non la rende priva di automatismi al suo interno. Legion infatti esegue un lavoro di riconoscimento delle vulnerabilità conosciute ed exploitabili ed, in tal caso, fornisce una guida all'utente per sfruttarle.

Prendendo come esempio il caso Metasploitable 3 Windows2k8, alla conclusione dei cicli di analisi e dei pentest basilari, nella categoria Scripts vengono elencate diverse vulnerabilità rilevate nel sistema. Selezionando una di esse possiamo notare, come in figura 3.29, che sulla destra ci sono una serie di codici, alcuni etichettati con il termine "*EXPLOIT*" sopra di essi e seguiti da un link.

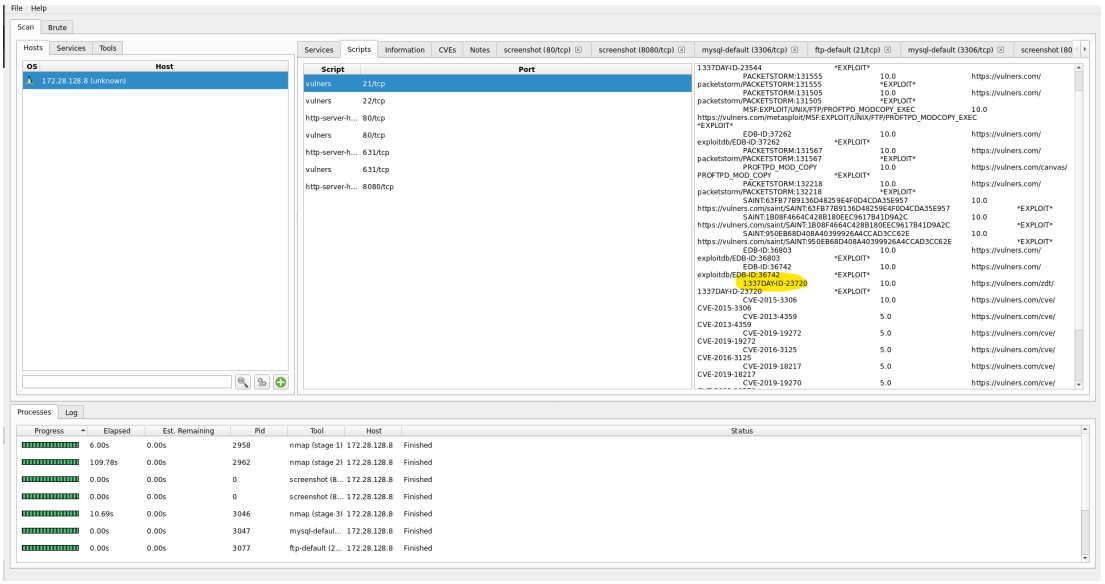

Figura 3.29: Elenco degli scripts riguardanti diversi exploits.

Copiando il link in un browser veniamo indirizzati a "Vulners", un sito contenente tutti gli exploit scoperti fino ad ora e contenente gli script utilizzabili per sfruttare la vulnerabilità in questione. In figura 3.30 l'exploit della vulnerabilità "1337DAY-ID-23720" con CVSS score 10.

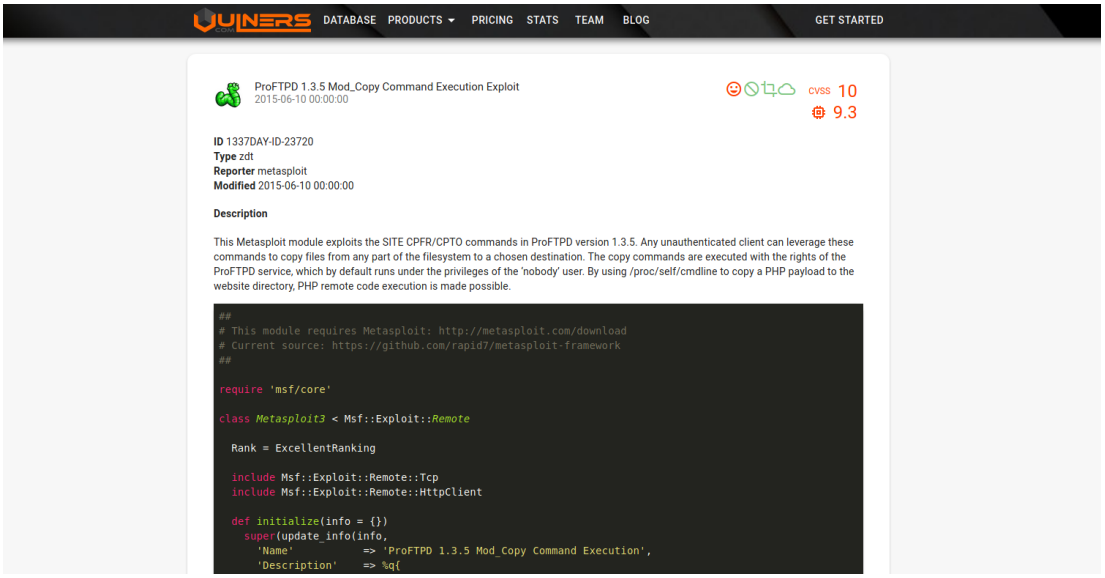

Figura 3.30: Exploit per la vulnerabilità 1337DAY-ID-23720

Il meccanismo di riconoscimento di Legion è uno strumento utile per chi vuole eseguire diverse tecniche di penetration testing ma non è uno strumento automatico. Nonostante non appartenga a questa categoria però, il suo sistema di riconoscimento è un componente utilizzabile in strumenti di più alto livello per automatizzarli ed in futuro potrebbe essere implementato come un automated tool per il penetration testing.

3.3.3 Xerror

L'applicazione Xerror [Chub] è un progetto ambizioso di Chudry, il nickname di un ragazzo Pakistano nella piattaforma GitHub. Il software utilizza Python2 come linguaggio principale per lo scripting e Django per interfacciarsi con l'utente tramite una web GUI. Per ottenere l'asincronia nell'applicazione, Xerror sfrutta i servizi di un server Redis ed un server Celery.

La peculiarità che contraddistingue questo software è l'utilizzo automatico di alcuni framework di pentesting come Metasploit e OpenVAS. Quest'ultimi aumentano le potenzialità del software poiché sono strumenti free che vengono aggiornati periodicamente con gli exploit più recenti ma che possono causare, come in questo caso studio, degli enormi problemi di compatibilità.

Le funzionalità che Xerror mostra nella pagina GitHub dedicata al progetto sono numerose:

- **Scanner di rete e host**
- **Scanner di vulnerabilità**
- **Exploit automatico di vulnerabilità**
- **Possibilità di aprire diverse sessioni di exploit contemporaneamente**
- **Interfaccia per interagire con shell aperte nei target host**
- **Possibilità di ottenere tutte le informazioni riguardanti il pentesting in formato PDF**

L'installazione del programma è decisamente ostica per l'utente. L'assenza di specifiche precise che guidano il setup di Xerror mettono in risalto la sua provenienza amatoriale. Lo sviluppo dell'applicazione è in un linguaggio oramai deprecato poiché il supporto degli sviluppatori per Python2 è terminato nel 2020. Alcuni dei package sono stati inseriti con una versione sbagliata e l'unico modo per rimediare a questi errori è stato andare a tentativi per verificarne il funzionamento.

Al termine della lunga fase di installazione il programma si presenta con un'interfaccia piuttosto intuitiva come mostrato in figura 3.31.

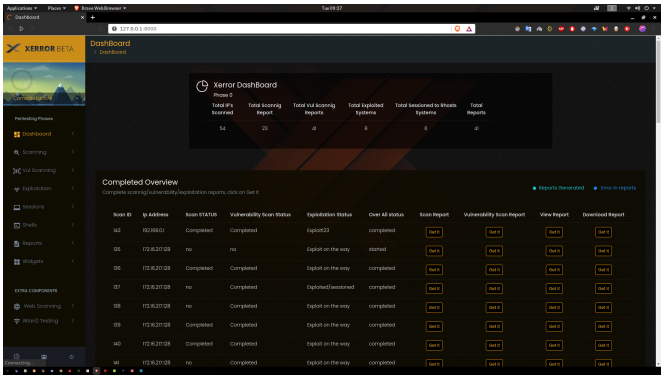

Figura 3.31: L'interfaccia di Xerror.

Diverse sono le funzionalità che Xerror propone nella barra di navigazione ma purtroppo poche sono quelle operative. Il software fornisce dei servizi di ricognizione e vulnerability scanning ma ciò che lo avrebbe reso più interessante, ovvero l'automazione del pentesting, presenta dei problemi di compatibilità con l'applicazione che fornisce questo servizio: OpenVAS. Quest'ultimo infatti ha subito un rebranding dal team di sviluppo che lo ha creato. Il cambio del nome del software, che ora si chiama Greenbone Vulnerability Management, ha reso inutilizzabili i comandi eseguiti da Xerror per comunicare con esso.

Il team Greenbone ha dimostrato quanto possa essere dannoso integrare software di terze parti. Data l'impossibilità di utilizzare Xerror per gli scopi che ci interessano, l'analisi del programma è da ritenersi un fallimento. La web GUI fornisce solamente comuni funzionalità di scanning che, come abbiamo visto nel corso dell'elaborato, non sono una novità nel mondo della sicurezza informatica. In figura 3.32 viene mostrata l'implementazione dello scanning nell'applicazione.

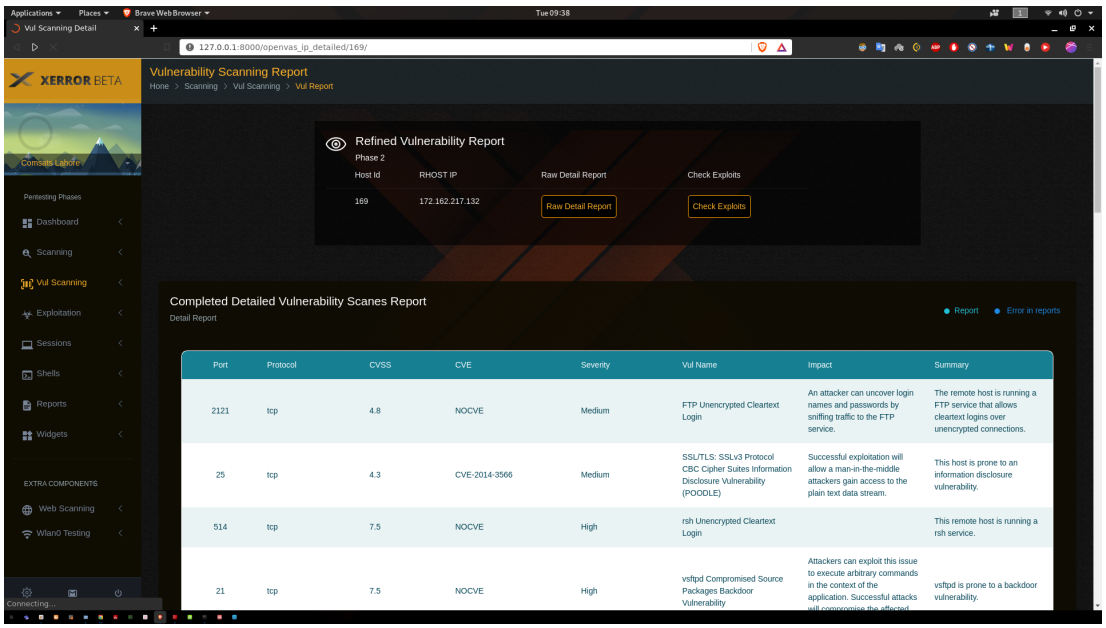

Figura 3.32: Vulnerability scanning in Xerror

Nonostante la delusione derivante dall'insuccesso di quest'ultimo caso studio, la decisione di inserire comunque questo software nell'elaborato nasce dall'ultimo aggiornamento fornito da Chudry sul proprio progetto:
"Mi scuso per chi sta riscontrando dei problemi relativi all'installazione e alla descrizione incompleta ma in realtà sto lavorando ad una versione successiva del progetto. La versione Pro non sarà dedicata solo al pentesting offensivo ma anche per il lato difensivo poiché supporta la soluzione SIEM completa basata su ELK (Wazuh), la risposta agli incidenti e le funzionalità di caccia alle minacce. Potete immaginarlo come una soluzione DevSecOps completamente automatizzata che, non solo manterrà la pipeline DevOps, ma la testerà/proteggerà anche dal codice sorgente. La distribuzione sarà gestita da un docker che, scaricando esclusivamente il codice sorgente, gestirà tutto il processo dell'installazione." [Chua].

4. Conclusioni e sviluppi futuri

Per trarre le conclusioni in merito a quanto analizzato nel corso dell'elaborato, nel capitolo passiamo in rassegna diversi punti. Nel primo paragrafo confrontiamo gli strumenti commerciali con quelli open source, soffermandoci sulle differenze dedotte dai tools presi in considerazione.
Nel paragrafo successivo poniamo in contrapposizione gli stessi strumenti dedicati al Vulnerability Assessment con quelli per il Penetration Testing automatico.

Il paragrafo 4.3 è dedicato alle difficoltà affrontate nel corso della fase di testing. In alcuni casi questi problemi hanno compromesso l'utilizzo di intere funzionalità dei software e pertanto costituiscono una parte importante dell'analisi degli strumenti.

Nelle ultime due sezioni un breve riepilogo su quanto emerso dalla nostra analisi ed una panoramica circa le tecnologie che potrebbero essere implementate in futuro.

4.1 Confronto tra commercial ed open-source tools

È bene precisare che, paragonare gli strumenti commerciali con quelli open source, è una forzatura sotto molteplici punti di vista.
La motivazione principale risiede nell'ampia differenza tra i target di utilizzo dei tools. Gli strumenti a pagamento, come PenTera ed Intruder, sono studiati per aziende di consulenza e forniscono pentesters esperti con i quali i clienti possono interfacciarsi.
I software open source invece rivestono uno scopo informativo ed istruttivo, ne sono una prova Legion, che mostra come agire in fase di pentesting, e Sn1per, utilizzato come dimostrazione della propria versione professionale.

Risulta quindi poco significativo sottolineare le differenze di funzionalità ed affidabilità dei tools presi in considerazione. Da un lato abbiamo grandi aziende che si servono di teams di esperti per la ricerca e lo sviluppo di nuove tecnologie come PcySys, dall'altro una comunità non retribuita che lavora su progetti per puro spirito di innovazione.

Ciò premesso, possiamo affermare che le funzionalità degli strumenti a pagamento sono più consistenti e varie rispetto agli altri software e l'assistenza che le aziende forniscono è sicuramente un punto a favore per il cliente.
D'altro canto le meccaniche di questi programmi, non essendo di pubblico dominio, sono nascoste a chi, invece di utilizzarle per motivi pratici, le studierebbe per scopi di ricerca come nel nostro caso.

4.2 Vulnerability Assessment e Penetration Testing tools

Nel capitolo 2 abbiamo discusso delle differenze, a livello teorico, tra analisi delle vulnerabilità e test di penetrazione. Per riassumerle brevemente, nel Vulnerability Assessment si esegue una ricognizione delle vulnerabilità presenti nel sistema mentre nel Penetration Testing vengono sfruttati gli exploit per ottenere ulteriori informazioni sull'obiettivo.

Ciò che si percepisce dall'analisi nel corso dell'elaborato è che, in ambito commerciale, il Vulnerability Assessment è la tecnologia più comune e utilizzata.
Questa preferenza del mercato potrebbe essere frutto di diverse motivazioni:
- Le tecnologie per il penetration testing automatico non sono ancora affidabili e competitive, in termini di prezzo ed efficienza, quanto strumenti di VA supportati da pentesters esperti.
- Per aziende che non necessitano di alti livelli di sicurezza informatica, rimediare alle vulnerabilità più esterne dei propri sistemi potrebbe essere sufficiente per evitare attacchi dall'esterno.
- I pentesters preferiscono utilizzare strumenti automatici per il Vulnerability Assessment e valutare personalmente quali sono gli attacchi più comuni e pericolosi.

Nonostante la tendenza a limitarsi all'analisi delle vulnerabilità sia predominante, il successo che ha riscosso un tool come PenTera ci fa pensare che gli strumenti automatici per il Penetration Testing guadagneranno velocemente importanza nel mercato della sicurezza informatica.

4.3 Problemi incontrati

Nell'analisi di strumenti che utilizzano tecnologie in via di sviluppo è comune imbattersi in problemi riguardanti la stabilità dei software. I tre diversi software open source utilizzati non si sono rivelati molto affidabili.
Sn1per e Legion hanno subito diversi crash durante la fase di utilizzo, interrompendo l'attività di analisi della rete e forzando la ripetizione del penetration test da zero ogni volta.
Nel caso di Xerror invece alcune funzionalità del software sono inutilizzabili a causa dell'incompatibilità con la versione aggiornata di OpenVAS, il framework per il pentesting utilizzato dal programma.

Un problema che può essere generalizzato a tutti gli strumenti di pentesting è quello dell'aggiornamento degli exploit. Mentre gli strumenti commerciali rimediano a questo problema utilizzando una struttura cloud based, gli altri software, come dimostrato da Sn1per, calano di effettività se utilizzati su target più recenti.

4.4 Conclusioni

Giunti al termine dell'elaborato riproponiamo le osservazioni emerse durante lo studio degli strumenti automatici per il penetration testing. Attualmente, nel settore della sicurezza informatica, c'è una grande confusione sulla differenza tra penetration testing e vulnerability assessment.

Molti degli strumenti che vengono pubblicizzati come pentesters automatici in realtà svolgono solamente l'analisi delle vulnerabilità a livello software.
Il pentesting automatico si comporta in modo molto simile a quello manuale, facendo delle scelte basate sulla vulnerabilità del sistema.
Entrambi i metodi di pentesting sopra citati possiedono sia lati positivi che negativi, attualmente il modo più efficace per rendere i software più sicuri è adottare un compromesso tra le due tipologie.

L'analisi degli strumenti ha messo in risalto la forte differenza tra strumenti di vulnerability assessment e quelli di penetration testing. Gli strumenti più efficaci nella parte del pentesting sono PenTera e Sn1per. Quest'ultimi infatti forniscono tecniche completamente automatiche e risultano i migliori software dei rispettivi settori, commerciale e open source.

Gli "automated tools for penetration testing" sono una tecnologia ancora in via di sviluppo ma il supporto ed i buoni risultati ottenuti da strumenti come PenTera creano alte aspettative circa la possibilità di affermarsi come standard.

4.5 Sviluppi futuri

Il futuro degli strumenti automatici per il pentesting sta prendendo la direzione che ha già rivoluzionato altri settori dell'informatica: lo sviluppo di Intelligenze Artificiali.
Ciò che differenzia il pentester da una macchina che utilizza un software come Sn1per è la capacità di poter capire quale sia la vulnerabilità più utile da sfruttare in base al sistema preso come target.
Sn1per esegue degli script preimpostati senza tener conto di quanto siano utili in determinate situazioni.
L'apporto che l'IA offrirebbe a questi strumenti è quello di adattare gli attacchi eseguiti in base alle informazioni ottenute. Ad esempio nel caso di un attacco ad un sistema Linux è inutile, ed alcune volte anche dannoso, che il software esegua tutti gli script per le vulnerabilità relative ad un sistema Windows.

Nell'elaborato redatto da Jonathon Schwartz [Sch] per l'Università del Queensland, egli mostra come l'utilizzo delle tecniche di Reinforcement learning siano ottimali nell'ambito di automazione del penetration testing.
Schwartz sostiene che l'utilizzo di questo tipo di apprendimento permetta un approccio generale con tutti i sistemi presi in considerazione poiché non richiede alcun tipo di "prior knowledge", ovvero nessuna informazione a priori.
L'utilizzo di tale approccio risolverebbe anche il problema della durabilità del software causato dal continuo aggiornamento dei sistemi informatici.

L'idea è che gli strumenti automatici di penetration testing, grazie al sostanziale apporto di intelligenze artificiali, possano in un futuro prossimo colmare la scarsa funzionalità e le lacune che emergono attualmente dalla gran parte dei software.
Per quanto concerne invece i software all'avanguardia ma ancora rilegati a nicchie di mercato, ulteriori progressi in questo campo abbasserebbero la frontiera costo/efficienza e li renderebbero più accessibili e di uso comune.

Bibliografia

[Cer] *CERT-US*. URL: https://us-cert.cisa.gov.

[Chua] Chudry. *From Xerror support*. URL: https://github.com/Chudry/Xerror/issues/9.

[Chub] Chudry. *Xerror*. URL: https://github.com/Chudry/Xerror.

[Cis] Cisco. *Mitigating the Cybersecurity Skills Shortage*. URL: https://www.cisco.com/c/dam/en/us/products/collateral/security/cybersecurity-talent.pdf.

[Dig19] AgID Team Digitale. ≪Sicurezza Informatica≫. In: (2019). URL: https://docs.italia.it/italia/piano-triennale-ict/pianotriennale-ict-doc/it/2019-2021/08_sicurezza-informatica.html.

[GoV] GoVanguard. *Legion*. URL: https://github.com/GoVanguard/legion.

[Impa] *Imperva*. URL: https://www.imperva.com.

[Impb] Imperva. *Vulnerability Assessment*. URL: https://www.imperva.com/learn/application-security/vulnerability-assessment/.

[Int] Intruder. *Penetration Testing*. URL: https://www.intruder.io/penetration-testing.

[Kala] *Kali Linux*. URL: https://www.kali.org.

[Kalb] KaliTools. *Faraday in Kali Linux*. URL: https://tools.kali.org/information-gathering/faraday.

[Lau] Minister Craig Laundy. *50 million investment into cyber security research and industry solutions*. URL: https://www.minister.industry.gov.au/ministers/craiglaundy/media-releases/50-million-investment-cyber-security-research-and-industry.

[Ms2] *Metasploitable 2*. URL: https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide/.

[Nay20] Dr Anand Nayyar. *The Best Open Source Automated Penetration Testing Tools*. 2020. URL: https://www.opensourceforu.com/2020/04/the-best-open-source-automated-penetration-testing-tools/.

[Pcy] PcySys. *Deloitte Cyber Security Service - Automated Penetration Testing Powered By PenTera*. URL: https://www.youtube.com/watch?v=bgl9NXTSRrM.

[Pyp] *PopularitY of Programming Language*. URL: https://pypl.github.io/PYPL.html.

[Sch] Jonathon Schwarz. *Autonomous Penetration Testing using Reinforcement Learning*. URL: https://arxiv.org/ftp/arxiv/papers/1905/1905.05965.pdf.

[Sta] *Statista*. URL: https://www.statista.com.

[Wik] Wikipedia. *Deloitte*. URL: https://it.wikipedia.org/wiki/Deloitte.

[Xer] XeroSecurity. *Sn1per*. URL: https://github.com/1N3/Sn1per.

Elenco delle figure