Università degli Studi di Camerino

SCUOLA DI SCIENZE E TECNOLOGIE

Corso di Laurea in Informatica (Classe L-31)

Analisi degli strumenti per la scansione di vulnerabilità
Case Study: Jok3r

Laureando
Brian Bernardini

Matricola 098694

Relatore
Prof. Fausto Marcantoni

A.A. 2019/2020

Prima di procedere con la trattazione, vorrei dedicare qualche riga alle persone che mi sono state vicine in questo percorso di crescita personale e professionale.

In primis, un sentito grazie al mio relatore Marcantoni Fausto per la sua disponibilità e tempestività. Grazie per le conoscenze trasmesse e per avermi dato le giuste indicazioni in ogni fase della realizzazione dell'elaborato.

Ringrazio infinitamente i miei genitori e mia sorella per avermi permesso di proseguire gli studi, perché nonostante il periodo difficile non hanno smesso di supportare la mia scelta e credere in me.

Un grandissimo ringraziamento va ai miei amici più cari, i "Baroni Birra", mi ritengo fortunato ad avere degli amici così perché ci sono sempre se ho bisogno e grazie a loro ho vissuto molte esperienze incredibili.

Grazie anche a Leonardo M. perché con la sua collaborazione ha reso meno stressanti e più divertenti gli ultimi lavori di gruppo.

Un ringraziamento speciale alla mia ragazza Manila, per essere sempre al mio fianco e con cui ho condiviso molte avventure. Ha sopportato ansie e paure e con la sua infinita pazienza mi ha sempre dato una mano ad affrontare le difficoltà incontrate. È anche grazie a lei che ho raggiunto questo traguardo.

Abstract

Negli ultimi dieci anni, le applicazioni Web sono diventate il modo più diffuso per fornire servizi su Internet. Integrandosi sempre di più nelle attività aziendali, richiedono di supportare funzionalità sofisticate rendendone la progettazione e l'implementazione sempre più complicate. La crescente popolarità e complessità rendono le applicazioni web un obiettivo primario per gli hacker.

Una vulnerabilità di un sito Web è una debolezza o un'errata configurazione nel codice del sito Web o di un'applicazione Web che consente a un utente malintenzionato di ottenere un certo livello di controllo del sito e possibilmente del server di hosting. Un'analisi di sicurezza (vulnerability assessment) consente l'identificazione e la classificazione di tutte le vulnerabilità potenziali dei sistemi e delle applicazioni valutando il danno potenziale che l'eventuale "attaccante" può infliggere.

Questo studio ha l'obiettivo principale di rilevare le vulnerabilità web più comuni attraverso strumenti di vulnerability assessment su diversi target (DVWA, Metasploitable 2, Windows Server 2012 R2 e Metasploitable 3). Il tool Jok3r essendo in grado di automatizzare la maggior parte delle attività durante l'identificazione/sfruttamento delle vulnerabilità "di base" dei più comuni servizi TCP/UDP è stato scelto come caso studio.

Dall'analisi emerge che sono state trovate delle vulnerabilità rilevanti nelle porte del servizio HTTP (porta 80/tcp) e FTP (porta 21/tcp) mentre nelle porte degli altri servizi sono state trovate vulnerabilità "trascurabili".

Sviluppi futuri del lavoro di tesi consentiranno di condurre un'analisi delle performance dello strumento prescelto su un target reale per la rilevazione delle vulnerabilità presenti fornendo consiglio per la loro risoluzione.

Keywords

Vulnerability, Vulnerability Assessment, Scanning Tools, Security, Web application.

SOMMARIO

INTRODUZIONE

Negli ultimi dieci anni, le applicazioni Web sono diventate il modo più diffuso per fornire servizi su Internet. Integrandosi sempre di più nelle attività aziendali, richiedono di supportare funzionalità sofisticate rendendone la progettazione e l'implementazione sempre più complicate. La crescente popolarità e complessità rendono le applicazioni web un obiettivo primario per gli hacker, infatti ogni giorno viene attaccata un'enorme quantità di siti Web, con un impatto sia diretto che significativo su un'enorme quantità di persone e dati.

Inoltre, a causa della pandemia COVID-19 è stata necessaria l'adozione forzata di nuove tecnologie e la corsa di emergenza al lavoro a distanza. Ciò ha costretto le aziende a ricontrollare ed evolvere le loro pratiche di sicurezza più velocemente che mai.

Le compagnie che lavoravano offline si son trovate a dover utilizzare Internet fornendo terreno ad hacker malintenzionati e sebbene molte abbiano investito in soluzioni efficaci per la sicurezza delle applicazioni, alcune sono ancora in ritardo perché ogni vulnerabilità non affrontata in un'applicazione aumenta l'esposizione al rischio dell'organizzazione.

Secondo il report *Web Application Vulnerabilities and Threats: Statistics for 2019*, pubblicato da Positive Technologies, gli hacker possono attaccare gli utenti di applicazioni web 9 volte su 10. Gli esperti hanno anche scoperto che il 16% delle applicazioni contiene vulnerabilità che consentono agli aggressori di assumere il pieno controllo del sistema e, sull'8% dei sistemi, e il pieno controllo del server di applicazioni web ha consentito di attaccare la rete locale. Con pieno accesso al server web, gli hacker possono anche inserire i propri contenuti sul sito attaccato (deturparlo) o persino attaccare gli utenti del sito, ad esempio infettando i loro computer con malware.

La percentuale di applicazioni web contenenti vulnerabilità ad alto rischio nel 2020 è diminuita in modo considerevole, del 17% rispetto all'anno precedente. Tuttavia, il livello di sicurezza generale delle applicazioni web rimane scarso.

Secondo gli esperti di Positive Technologies l'82% delle vulnerabilità si trova nel codice dell'applicazione, suggerendo che il codice sorgente non viene controllato per le vulnerabilità durante lo sviluppo; ciò lascia intendere che gli sviluppatori prediligono concentrarsi sullo sviluppo delle funzionalità dell'app piuttosto che sulla sua sicurezza.

OBIETTIVO DELLA TESI

L'obiettivo principale di questo lavoro di tesi è di fornire una conoscenza generale sul concetto di vulnerabilità web e riconoscere le più comuni. Inoltre verranno presi in esame diversi strumenti open source che hanno la funzione di rilevare le vulnerabilità web.

Tra gli strumenti presi in considerazione per questo studio, il focus principale verrà rivolto ad uno solo ovvero Jok3r. Verranno elencati i suoi punti di forza e spiegati i risultati ottenuti testandolo su diversi sistemi.

La tesi è strutturata come segue:

Il Capitolo 1 ha il compito di introdurre il concetto di vulnerabilità web ed elencare le categorie più comuni. Inoltre è presente un riferimento a come vengono classificate e riconosciute a livello mondiale e il capitolo si conclude con la spiegazione del vulnerability assessment.

Nel Capitolo 2 vengono elencati i tools che sono stati presi in esame per la scrittura della tesi, spiegandone brevemente l'installazione e le caratteristiche principali. Una tabella riassumerà i concetti più importanti. Inoltre è presente una parte dedicata ai sistemi che sono stati usati come target e oltre a riportare il modo in cui sono stati installati e configurati vengono descritte le loro peculiarità.

Il Capitolo 3 è dedicato al caso studio della tesi, ovvero Jok3r. Ci si sofferma su ciò che offre questo tool e come è stato installato e configurato per l'uso. Infine vengono raccolti i risultati ottenuti dalle scansioni sui sistemi descritti nel capitolo precedente facendo un confronto tra loro.

La tesi termina con il Capitolo 4 dove sono raccolte le conclusioni e discusso ciò che è stato appreso nel periodo di sviluppo e scrittura della tesi.

1 VULNERABILITÀ WEB

1.1 COS'È UNA VULNERABILITÀ

Quando gli errori del software mettono a rischio la sicurezza dei nostri dati allora si parla di vulnerabilità informatica e bisogna fare molta attenzione.

Una vulnerabilità informatica può essere intesa come una componente (esplicita o implicita) di un sistema informatico in corrispondenza della quale le misure di sicurezza sono assenti, ridotte o compromesse, il che rappresenta un punto debole del sistema e consente a un eventuale aggressore di compromettere il livello di sicurezza dell'intero sistema.

In particolare una vulnerabilità di un sito Web è una debolezza o un'errata configurazione nel codice del sito Web o di un'applicazione Web che consente a un utente malintenzionato di ottenere un certo livello di controllo del sito e possibilmente del server di hosting. La maggior parte delle vulnerabilità viene sfruttata tramite mezzi automatizzati, come scanner di vulnerabilità e botnet. I criminali informatici creano strumenti specializzati che setacciano Internet per determinate piattaforme, come WordPress o Joomla, alla ricerca di vulnerabilità comuni e pubblicizzate. Una volta trovate, queste vulnerabilità vengono quindi sfruttate per rubare dati, distribuire contenuti dannosi o iniettare contenuti deturpanti e spam nel sito vulnerabile.

1.1.1 Differenza tra vulnerabilità e minaccia

Le vulnerabilità quindi non compromettono un sistema, ma se utilizzate da quella che viene definita una minaccia (azione indesiderata) possono trasformarsi in un evento indesiderato.

Infatti, è la minaccia l'elemento attivo di un potenziale innesco di un rischio. La minaccia è cioè l'agente che, sfruttando una vulnerabilità, potrebbe arrecare un disturbo, un attacco, un danno al sistema.

In pratica, la minaccia è la causa scatenante (spesso non controllabile direttamente), la vulnerabilità è la con-causa (questa sì controllabile) che consente l'azione della minaccia e quindi il manifestarsi del rischio e del relativo effetto.

1.2 COMMON VULNERABILITIES AND EXPOSURES

Man mano che vengono rilevate delle vulnerabilità, queste vengono rese note in un database, raccogliendone le informazioni utili (versioni del software impattate, problemi potenziali, soluzioni temporanee ecc.) e associando ad ognuna di esse un identificatore univoco per evitare di duplicare le informazioni.

Il *Common Vulnerabilities and Exposures*, o CVE, è un dizionario di vulnerabilità e falle di sicurezza note pubblicamente. È mantenuto dalla MITRE Corporation ed è finanziato dalla National Cybersecurity FFRDC del Dipartimento della Sicurezza interna degli Stati Uniti. Il CVE è utilizzato dal Security Content Automation Protocol (SCAP) e le vulnerabilità, identificate da un identificatore univoco, sono elencate nel sistema MITRE e nel National Vulnerability Database americano. L'identificazione univoca delle CVE permette una maggiore comunicazione nel mondo della sicurezza e aiuta nella valutazione della diffusione di servizi e strumenti.

Figura 1 Logo ufficiale CVE

La documentazione di MITRE Corporation definisce gli identificatori CVE come identificatori univoci e comuni per vulnerabilità di sicurezza delle informazioni note pubblicamente in pacchetti software rilasciati pubblicamente.

Storicamente, gli identificatori CVE avevano uno stato di "candidato" ("CAN-") e potevano quindi essere promossi a voci ("CVE-"), tuttavia questa pratica è stata terminata qualche tempo fa e tutti gli identificatori sono ora assegnati come CVE.

L'assegnazione di un numero CVE non è una garanzia che diventerà una voce CVE ufficiale (ad esempio, un CVE potrebbe essere assegnato in modo improprio a un problema che non è una vulnerabilità di sicurezza o che duplica una voce esistente).

I CVE vengono assegnati da un'autorità di numerazione CVE (CNA ovvero **CVE** Numbering Authorities); ci sono tre tipi principali di assegnazione del numero CVE:

1. La Mitre Corporation funge da Editor e Primary CNA
2. Vari CNA assegnano numeri CVE ai propri prodotti (ad esempio Microsoft, Oracle, HP, Red Hat, ecc.)
3. Un coordinatore di terze parti come il centro di coordinamento CERT può assegnare numeri CVE per prodotti non coperti da altri CNA

I CVE sono per il software che è stato rilasciato pubblicamente; questo può includere beta e altre versioni pre-rilascio se sono ampiamente utilizzate. Il software commerciale è incluso nella categoria "rilasciato pubblicamente", tuttavia il software personalizzato che non è distribuito generalmente non riceve un CVE. Inoltre, ai servizi (ad esempio un provider di posta elettronica basato sul Web) non vengono assegnati CVE per le vulnerabilità rilevate nel servizio (ad esempio una vulnerabilità XSS) a meno che il problema non esista in un prodotto software sottostante che è distribuito pubblicamente.

1.2.1 Campi dati CVE

Il database CVE contiene diversi campi:

1. Descrizione: Questa è una descrizione testuale standardizzata dei problemi. Una voce comune è:

   ** RESERVED ** This candidate has been reserved by an organization or individual that will use it when announcing a new security problem. When the candidate has been publicized, the details for this candidate will be provided.

   Ciò significa che il numero di accesso è stato riservato da Mitre per un problema o che un CNA ha riservato il numero;

2. Riferimenti: Questo è un elenco di URL e altre informazioni;

3. Data di creazione: Questa è la data in cui è stata creata la voce. Per CVE assegnati direttamente da Mitre, questa è la data in cui Mitre ha creato la voce CVE. Per CVE assegnati da CNA (ad esempio Microsoft, Oracle, HP, Red Hat, ecc).

Campi obsoleti

I seguenti campi erano utilizzati in precedenza nei record CVE meno recenti, ma non vengono più utilizzati.

1. Fase: la fase in cui si trova CVE (ad esempio CAN, CVE);

2. Voti: in precedenza i membri del consiglio avrebbero votato sì o no se la CAN dovesse essere accettata o trasformata in CVE;

3. Commenti: commenti sul problema;

4. Proposta: quando il problema è stato proposto per la prima volta.

1.3 OWASP Top 10

Ogni tre o quattro anni, OWASP, Open Web Application Security Project, (https://owasp.org/) rivede e pubblica il proprio elenco delle 10 principali vulnerabilità delle applicazioni web.

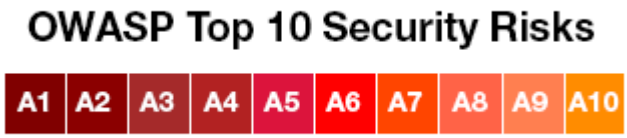

Figura 2 Scala di rischio secondo OWASP

L'OWASP è un'organizzazione di beneficenza senza scopo di lucro focalizzata sul miglioramento della sicurezza del software.

L'elenco include non solo le prime dieci minacce OWASP, ma anche il potenziale impatto di ciascuna vulnerabilità e informazioni su come evitarle. L'elenco completo è compilato da una varietà di fonti esperte come consulenti per la sicurezza, fornitori di sicurezza e team di sicurezza di aziende e organizzazioni di tutte le dimensioni. È riconosciuto come una guida essenziale alle migliori pratiche di sicurezza delle applicazioni web.

Dal 2003, OWASP pubblica la Top 10 di OWASP ogni tre / quattro anni, di seguito è riportato l'elenco del 2017 dato che la realizzazione della lista del 2020 è ancora in corso.

1. **Injection:** I difetti di injection, come SQL, NoSQL, OS e LDAP injection, si verificano quando dati non attendibili vengono inviati a un interprete come parte di un comando o di una query. I dati ostili dell'aggressore possono indurre l'interprete a eseguire comandi involontari o ad accedere ai dati senza un'adeguata autorizzazione. Le vulnerabilità di injection possono comparire in tutti i posti all'interno dell'applicazione web che consentono all'utente di fornire input dannosi. Questo è uno dei più vecchi attacchi contro le applicazioni web, ma è ancora il re delle vulnerabilità perché è ancora diffuso e molto dannoso;

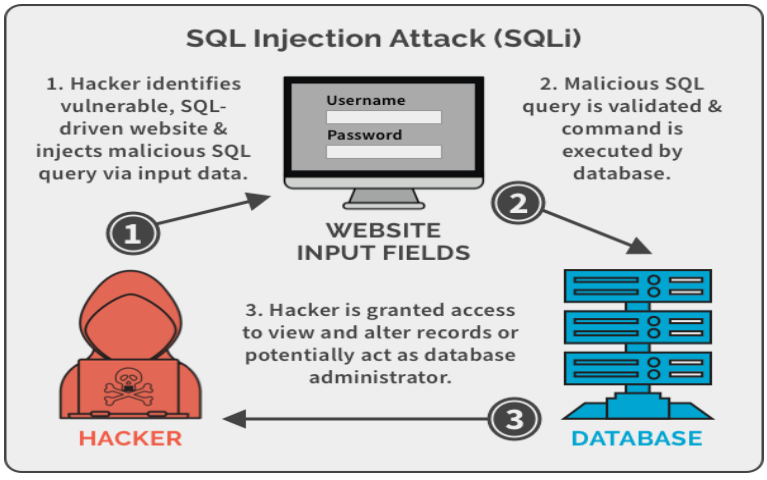

Figura 3 Schema dell'injection, in particolare dell'SQL Injection

2. **Broken Authentication:** Le funzioni dell'applicazione relative all'autenticazione (login, reimpostazione della password, la modifica della password, il ripristino dell'account, ecc.) e alla gestione delle sessioni sono spesso implementate in modo errato, consentendo agli aggressori di compromettere password, chiavi o token di sessione o di sfruttare altri difetti di implementazione per assumere l'identità di altri utenti temporaneamente o permanentemente. Le sessioni sono gli identificatori univoci assegnati agli utenti dopo l'autenticazione e hanno molte vulnerabilità o attacchi associati al modo in cui questi identificatori vengono utilizzati dall'applicazione web. Le sessioni quindi servono per tenere traccia delle richieste di ogni utente e quindi sono una componente chiave dell'hacking dell'utente web;

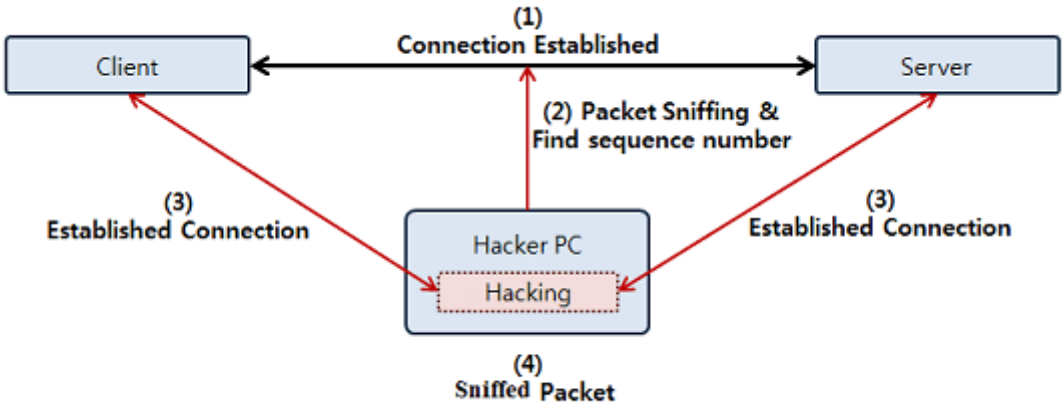

Figura 4 Schema del Broken Authentication

3. **Sensitive Data Exposure:** Si verifica quando un'applicazione, un'azienda o un'altra entità espone inavvertitamente dati personali. L'esposizione ai dati sensibili è diversa da una violazione dei dati, in cui un utente malintenzionato accede e ruba informazioni. L'esposizione ai dati sensibili si verifica a causa della non adeguata protezione di un database in cui sono archiviate le informazioni. Questo potrebbe essere il risultato di una moltitudine di cose come crittografia debole, nessuna crittografia, difetti del software o quando qualcuno carica per errore i dati in un database errato. Diversi tipi di dati possono essere esposti come numeri di conto bancario, numeri di carta di credito, dati sanitari, token di sessione, numero di previdenza sociale, indirizzo di casa, numeri di telefono, date di nascita e informazioni sull'account utente come nomi utente e password;

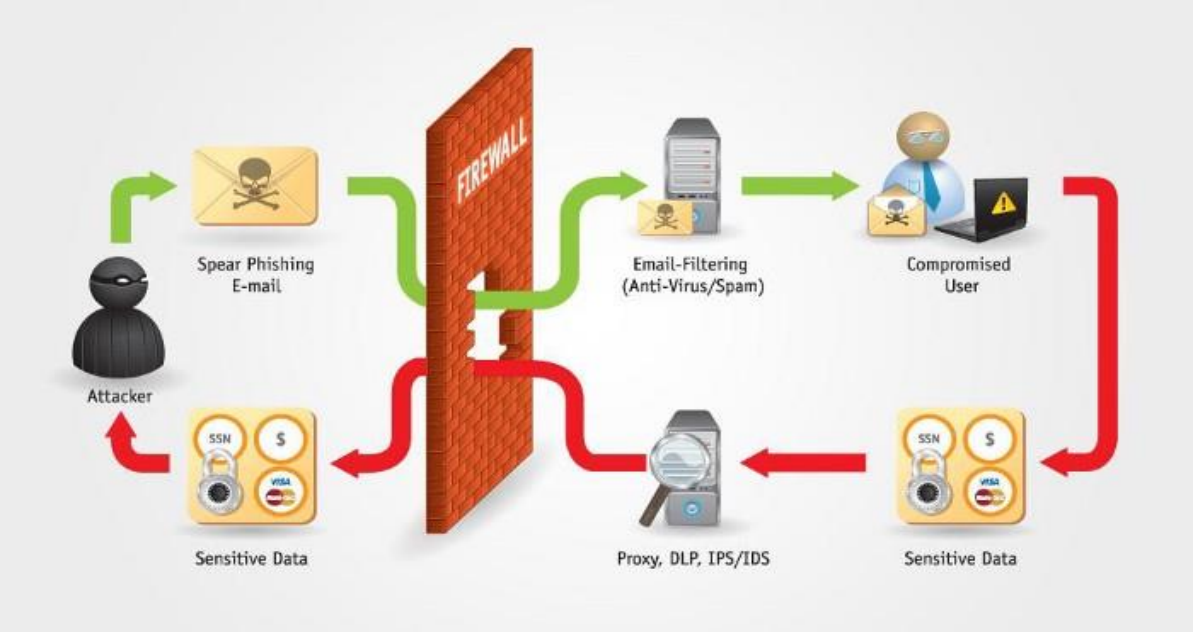

Figura 5 Schema del Sensitive Data Exposure

4. **XML External Entities (XXE):** È una vulnerabilità di sicurezza Web che consente a un utente malintenzionato di interferire con l'elaborazione dei dati XML da parte di un'applicazione. Un attacco volto a sfruttare questa vulnerabilità si verifica quando l'input XML contenente un riferimento a un'entità esterna viene elaborato da un parser XML configurato in modo debole. Questo attacco può portare alla divulgazione di dati riservati, denial of service, falsificazione di richieste lato server, port scanning dal punto di vista della macchina in cui si trova il parser e altri impatti sul sistema;

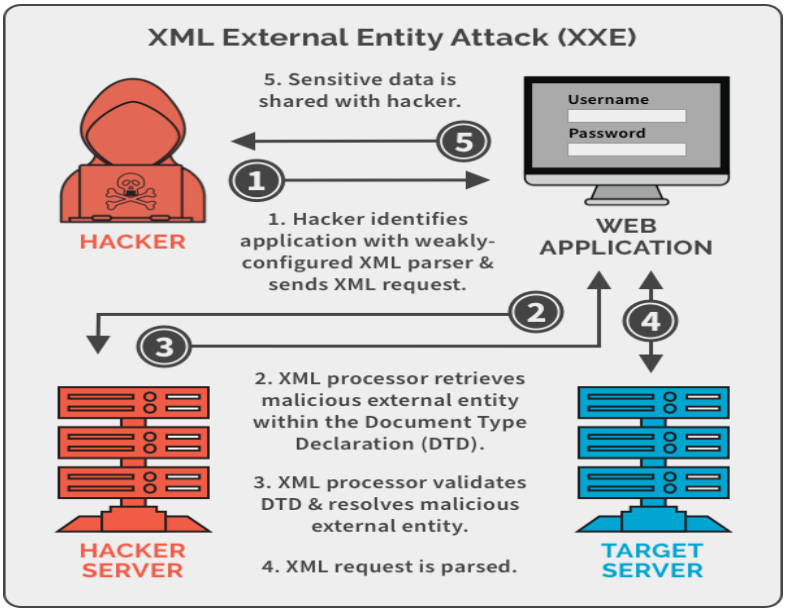

Figura 6 Schema del XML External Entities

5. **Broken Access Control:** Le restrizioni su ciò che gli utenti autenticati possono fare spesso non vengono applicate correttamente. Gli aggressori possono sfruttare questi difetti per accedere a funzionalità e / o dati non autorizzati, come accedere agli account di altri utenti, visualizzare file sensibili, modificare i dati di altri utenti, modificare i diritti di accesso, ecc;

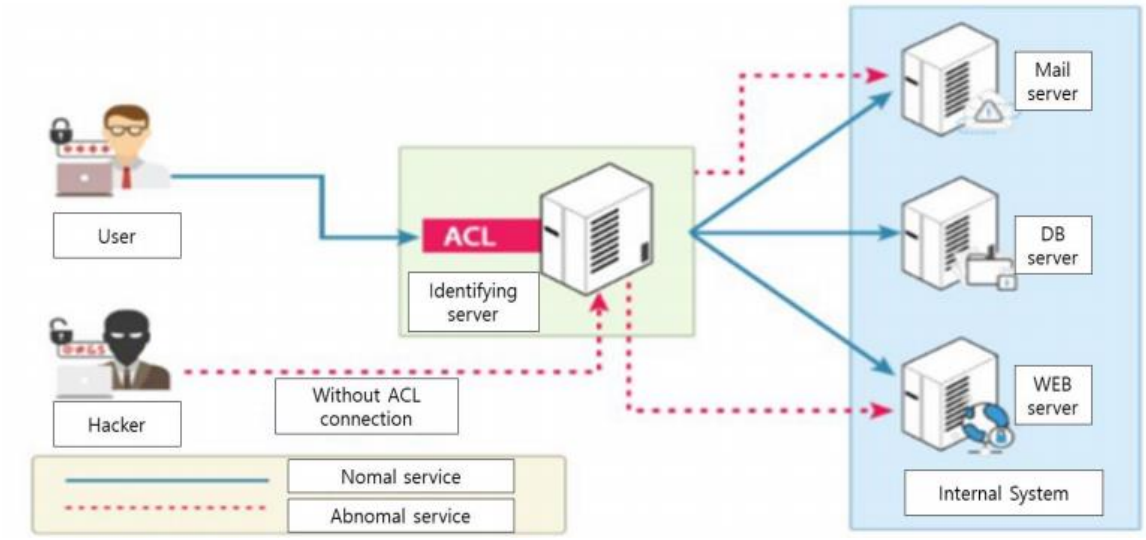

Figura 7 Schema del Broken Access Control

6. **Security Misconfiguration:** Si verifica quando un componente dell'applicazione Web è suscettibile di attacchi a causa di un'opzione di configurazione errata o non sicura. Sono debolezze di configurazione che possono esistere nei componenti software o nei sottosistemi. Ad esempio, il software del server Web può essere fornito con account utente predefiniti che un utente malintenzionato può utilizzare per accedere al sistema oppure il software potrebbe avere servizi non necessari abilitati, come la funzionalità di amministrazione remota. Quindi l'applicazione risulta debole contro attacchi come Brute force/credential stuffing, Code injection, Buffer overflow, Command injection;

Figura 8 Immagine rappresentativa del Security Misconfiguration

7. **Cross-site Scripting (XSS):** è un tipo di vulnerabilità dei siti web che consente a chi lancia l'attacco di piazzare script dannosi in pagine web e app altrimenti affidabili, per poi installare malware sui browser web degli utenti. Mediante il cross-site scripting, gli hacker non attaccano né dirottano direttamente gli utenti, ma si limitano di disseminare liberamente il proprio malware a un numero incalcolabile di persone. Gli attacchi XSS sono rivolti al codice (detto anche script) di una pagina web in esecuzione nel browser dell'utente, senza attaccare direttamente il server del sito web. Una volta attaccato, il browser viene infettato con degli script dannosi che cercheranno di danneggiare il computer. Esiste una varietà quasi infinita di attacchi XSS, tuttavia la maggioranza cerca di impadronirsi di dati personali, reindirizzare le vittime su siti web controllati da un hacker o far eseguire al PC le operazioni scelte dall'hacker;

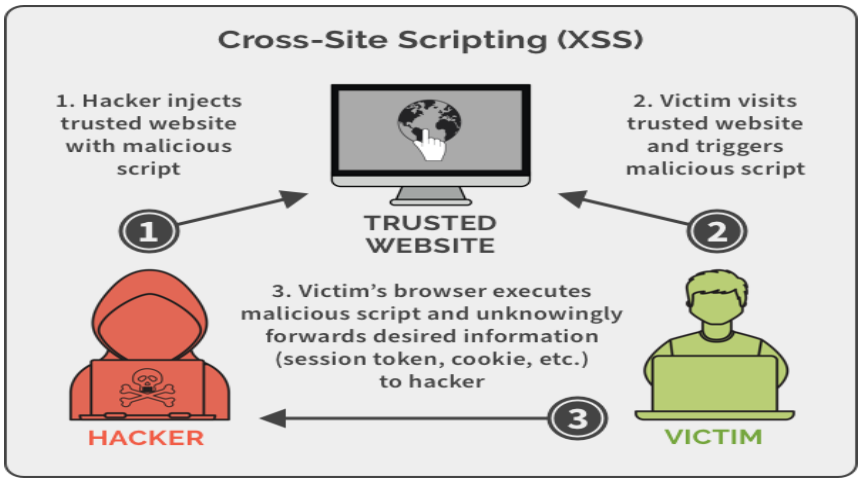

Figura 9 Schema del Cross-site Scripting

8. **Insecure Deserialization:** La maggior parte dei linguaggi di programmazione offre la possibilità di personalizzare i processi di deserializzazione ovvero trasformare i dati serializzati provenienti da un file, flusso o socket di rete in un oggetto. Sfortunatamente, è spesso possibile che un utente malintenzionato abusi di queste funzionalità di deserializzazione quando l'applicazione deserializza dati non attendibili controllati dall'autore dell'attacco. Il successo di questi attacchi si verifica quando dati non attendibili vengono utilizzati per compromettere la logica di un'applicazione, infliggere un attacco DoS (Denial of Service) o persino eseguire codice arbitrario dopo la deserializzazione;

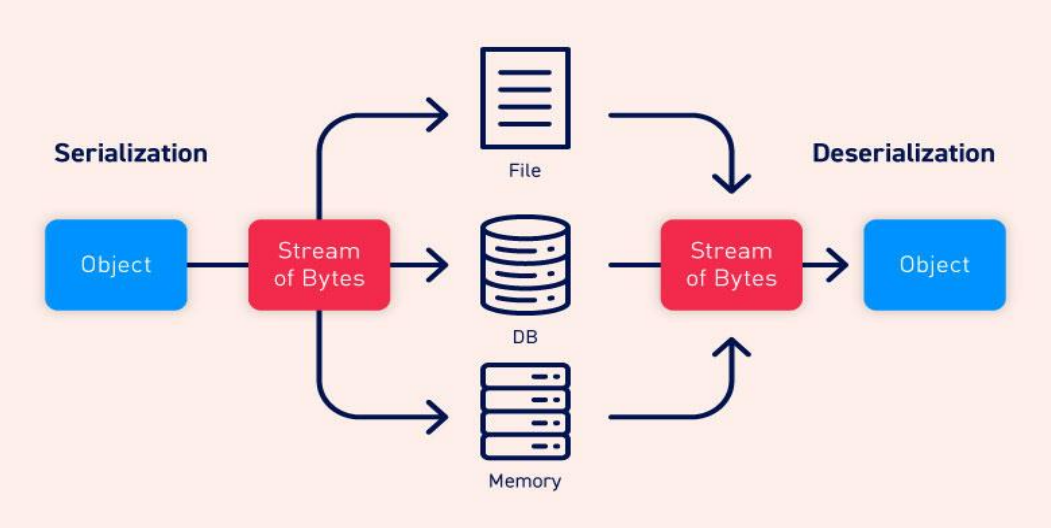

Figura 10 Schema dell'Insicure deserialization

9. **Use of Components with Known Vulnerabilities:** Questo tipo di minaccia si verifica quando i componenti come le librerie e i framework utilizzati all'interno dell'app vengono quasi sempre eseguiti con privilegi completi. Se un componente con delle vulnerabilità note viene implementato nell'app, rende più facile il lavoro dell'hacker nel provocare una grave perdita di dati o perdere il controllo del server;

Figura 11 Immagine rappresentativa dell'Use of Components with Know Vulnerabilities

10. **Insufficient Logging and Monitoring:** Piuttosto che essere una vulnerabilità di per sé è la mancanza di varie best practice sulla sicurezza che potrebbero prevenire o ridurre le violazioni della sicurezza. È il fondamento di quasi tutti i principali incidenti perché gli aggressori fanno affidamento sulla mancanza di controllo e sulla lentezza di una risposta tempestiva di una vulnerabilità nota per raggiungere i loro obiettivi.

Figura 12 Immagine rappresentativa dell'Insufficient Logging and Monitoring

1.4 VULNERABILITY ASSESSMENT

Il vulnerability assessment è un'analisi di sicurezza che ha come obiettivo l'identificazione e la classificazione di tutte le vulnerabilità potenziali dei sistemi e delle applicazioni valutando il danno potenziale che l'eventuale "attaccante" può infliggere.

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.

Ecco perché è utile eseguirlo. Si tratta di un vero e proprio check-up, "un'analisi del sangue del sistema informatico", indispensabile per stabilire se si è a rischio di un attacco informatico.

Il vulnerability assessment è composto da delle scansioni che vengono effettuate sulle Web app o sulle reti aziendali mediante strumenti professionali, i cosiddetti vulnerability scanner, che setacciano i target aziendali e che possono eseguire:

- scansioni relative al networking e quindi ai dispositivi di rete;
- scansioni relative agli host/server;
- scansioni specifiche per le reti wireless;
- scansioni relative alle Web application;
- scansioni relative ai database.

Sul mercato esistono diversi vulnerability scanner, strumenti sia open source che commerciali che possono essere utilizzati.

Nessun vulnerability assessment aziendale potrà dirsi completo senza una verifica del fatto che una vulnerabilità informatica sia effettivamente sfruttabile dall'attaccante (**exploitation**) e senza una verifica di cosa si trovi davanti l'attaccante dopo aver sfruttato la vulnerabilità (difficilmente riscontrabile con strumenti standard che non tengono conto delle diverse implementazioni aziendali).

1.4.1 Vulnerability Assessment vs Penetration Test

La maggior parte dei professionisti della sicurezza informatica ha familiarità con i termini “vulnerability assessment” e “penetration test”. Sfortunatamente, in molti casi, i due termini sono sostituiti in modo erroneo. Entrambe le attività sono componenti integrate di un programma più completo di gestione delle vulnerabilità.

Il penetration test, è un’attività orientata al raggiungimento di un risultato. È un esercizio che simula un attacco reale, come elusione delle difese, mappatura dei vettori d’attacco, sfruttamento delle vulnerabilità e utilizzo di exploit. In altre parole, dimostra come un attaccante malintenzionato potrebbe eludere le difese dell’obbiettivo e sfruttare le vulnerabilità per accedere ai dati o prendere il controllo del sistema. Come nel caso dei vulnerability assessment anche nell’attività di penetration test, spesso, vengono utilizzati strumenti per la scansione delle vulnerabilità o altri strumenti specifici per analizzare applicazioni web e infrastrutture di rete. Nello specifico, nelle attività di penetration test vengono sfruttate le vulnerabilità trovate in modo da raggiungere l’obiettivo finale. Ovvero la compromissione del sistema.

La differenza fondamentale tra un vulnerability assessment ed un penetration test è che il primo è list-oriented invece il secondo è goal-oriented. Quindi, dato che di base per entrambi vengono utilizzare le medesime tecniche e strumenti, quale metodologia è più adatta? Come sceglierle e perché?

Tramite un penetration test si cerca di aprire una breccia nelle difese del target allo scopo di raggiungere un obiettivo definito a priori. Questo implica un sufficiente livello di maturità nella gestione della sicurezza. Per tanto è più adatto in situazione dove si preferisce la profondità piuttosto che l’ampiezza. Un’attività di vulnerability assessment, invece, è indicata in situazioni dove c’è la presunzione della presenza di vulnerabilità o come punto di inizio per migliorare la sicurezza. In conclusione è un approccio utile per fornire all’organizzazione un elenco completo delle vulnerabilità presenti nel sistema che devono essere risolte a breve termine, senza valutare specifici scenari di attacco. Questo rende il l’attività più adatta in contesti in cui è preferibile l’ampiezza piuttosto che profondità.

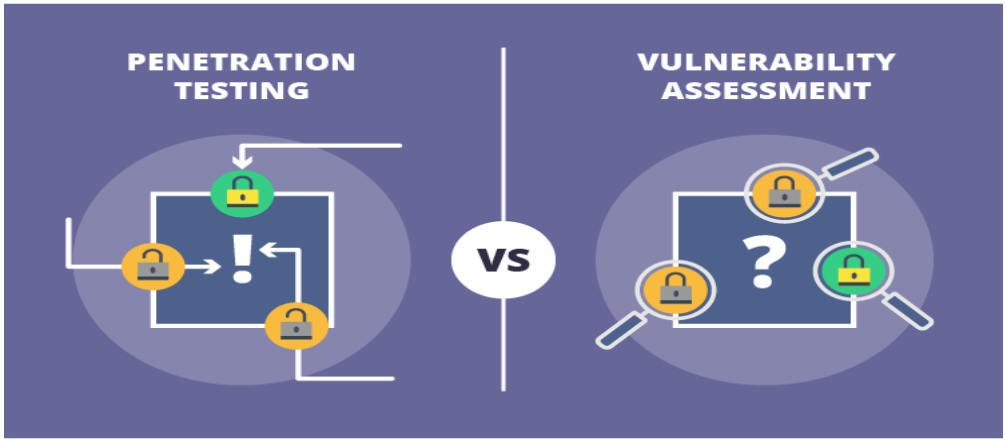

Figura 13 Differenza tra Penetration Test e Vulnerability Assessment

2 TOOLS E TARGET

Nelle pagine seguenti vengo descritti i tools che sono stati presi in considerazione per effettuare delle scansioni a scopo educativo su dei target ad hoc. Per ogni strumento vengono elencate le features principali e come svolgono le loro funzioni. La selezione di questi strumenti è il frutto di diverse ricerche su siti competenti che trattano l'ambito della sicurezza informatica.

Tutti i tools, e anche tutti i sistemi sottoposti a scansioni, sono stati installati in una virtual machine con sistema operativo Debian (64-bit) con una distribuzione Kali Linux, la scelta di usare questa distribuzione perché ogni strumento che è stato studiato è compatibile con l'ambiente Linux e inoltre Kali Linux è pensata per l'informatica forense e la sicurezza informatica.

Tutte le virtual machine utilizzate sono state create usando VirtualBox. Rilasciato come progetto open source, adesso è supportato da Oracle. VirtualBox crea su un computer con un sistema operativo (definito 'host') una macchina virtuale (VM, virtual machine) su cui può essere eseguito un sistema operativo differente (definito 'guest').

In fase di configurazione, si può scegliere quanti core della CPU, quanta RAM e quanto spazio su disco devono essere dedicati alla VM. Come sistemi operativi host, VirtualBox supporta SO Windows, Mac, Linux e Solaris; come sistemi operativi guest sono supportate varie versioni di Windows, Linux, Solaris, OpenBSD e altri.

Mentre per testare e valutare gli strumenti presi in esame, come target sono stati scelti dei sistemi con delle vulnerabilità già note. Ogni sistema, come già detto in precedenza, viene hostato in una macchina virtuale usando un software per la virtualizzazione di sistemi, VirtualBox.

I sistemi scelti sono DVWA, Metasploitable 2, Windows Server 2012 R2 e Metasploitable 3.

Di seguito una tabella riassuntiva di comparazione tra gli strumenti analizzati (**Tabella 1**). Per quanto riguarda lo strumento Jok3r, in quanto protagonista del caso studio, verrà analizzato nel dettaglio nel capitolo successivo.

Figura 14 Logo VirtualBox

Tabella 1 Caratteristiche principali degli strumenti analizzati durante lo svolgimento della tesi

	XAttacker	Red Hawk	Osmedeus	OpenVas	Raccoon	Jok3r
Tipologia	Website Vulnerability Scanner & Auto Exploiter	Information Gatherer and Detecting Vulnerabilities.	Fully Automated Vulnerability Scanner	Vulnerability Scanner	Offensive Security Tool for Reconnaissance and Information Gathering	Network and Web Pentest Framework
Linguaggio di programmazione	Perl	PHP	Python	C	Python	Python
Licenza	Open Source, Premium	Open Source	Open Source	Open Source	Open Source	Open Source
Sistema Operativo	Windows Linux Android	Linux	Kali Linux *nx OS Mac OS	Linux	Linux	Linux Mac OS X
Interfaccia	CLI	CLI	CLI	CLI	CLI	CLI
Report	Al termine della scansione	Al termine della scansione	Salvato nel database	Salvato nel database	Al termine della scansione	Salvato in locale
Formato del Report	Non scaricabile	Non scaricabile	HTML	TXT, XML, PDF	Non scaricabile	HTML

2.1 XATTACKER

XAttacker è uno scanner di vulnerabilità Web e un auto exploiter che analizza i siti Web per trovare diverse vulnerabilità a seconda del CMS (content management systems) che utilizzano. Dopo aver individuato le vulnerabilità, lo strumento genererà un exploit per il sito Web e invierà all'utente il collegamento dell'exploit.

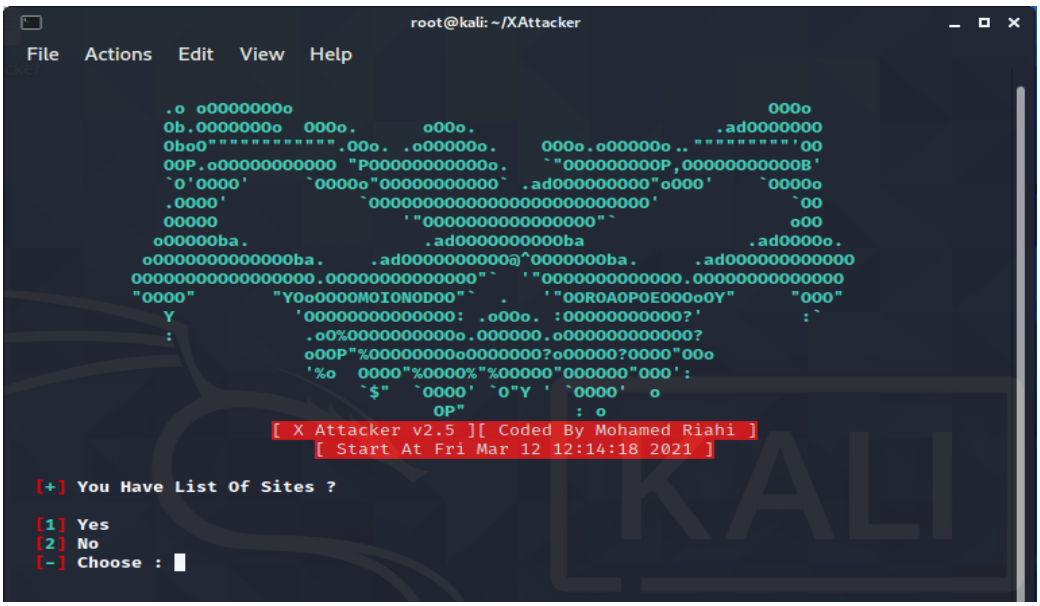

Figura 15 Home e logo XAttacker

XAttacker copre cinque diversi CMS che hanno tutti i propri exploit unici:

- WordPress (Revslider Upload Shell, WP All Import, Gravity Forms, WP Job Manager, Cherry Plugin, ecc.);
- Joomla (Com Media, Fabrik, Users, Blog, Weblinks, EXplorer, Jce, ecc.);
- DruPal (Add Admin, Drupalgeddon);
- PrestaShop (columnadverts, homepageadvertise, simpleslideshow, videostab, ecc.);
- Lokomedia (SQL injection).

Ogni CMS ha il proprio set unico di exploit che includono la possibilità di inviare shell, deturpare il sito Web, accedere al cPanel o accedere al database. In totale, ci sono più di 40 diversi exploit disponibili. Il vantaggio principale di questo strumento è che controlla automaticamente CMS sul sito Web, esegue ogni exploit applicabile sul sito Web e quindi restituisce il collegamento per ogni exploit riuscito. Per ulteriori opzioni, lo sviluppatore fornisce anche una versione premium a un prezzo basso.

2.2 RED HAWK

Red Hawk è un information gatherer tutto in uno che consente all'utente di trovare informazioni critiche su qualsiasi sito Web come indirizzo IP, CMS, DNS e presenza di Cloudflare. Red Hawk è composto da 13 moduli che spaziano dalla raccolta di informazioni di base al rilevamento delle vulnerabilità. Pertanto, Red Hawk è uno strumento essenziale per qualsiasi penetration tester che si rispetti.

Red Hawk richiede solo due input da parte dell'utente, l'URL del target e se il target ha il protocollo HTTP o HTTPS. Successivamente, l'utente può selezionare tra 13 diverse opzioni che restituiranno le informazioni desiderate. Queste informazioni possono variare dalle più elementari come l'indirizzo IP e il titolo del sito a informazioni più avanzate come il numero e lo stato delle porte.

Oltre a queste funzionalità di raccolta delle informazioni, Red Hawk dispone di uno scanner SQLi che recupera collegamenti e parametri che potrebbero essere vulnerabili alle iniezioni SQL basate su errori.

Figura 16 Home e logo Red Hawk

2.3 OSMEDEUS

Osmedeus è uno scanner di vulnerabilità completamente automatizzato che analizza il sistema, il sottodominio e il sito Web per identificare le falle di sicurezza. È un utile strumento di sicurezza in grado di scansionare e acquisire screenshot del bersaglio.

Osmedeus è uno scanner di vulnerabilità open source sviluppato per proteggere la tua organizzazione da imminenti minacce alla sicurezza informatica. Combina il meglio della sorveglianza intranet ed extranet.

Presenta un'impressionante raccolta di strumenti come il rilevamento della tecnologia Web, il rilevamento IP e il rilevamento della macchina. Può separare lo spazio di lavoro per memorizzare tutto l'output della scansione ei dettagli di registrazione. Infine, supporta una scansione continua e consente di visualizzare il report della scansione dalla riga di comando.

Se usato bene, Osmedeus può essere un potente ed efficace scanner di sorveglianza e vulnerabilità. È possibile utilizzarlo per raccogliere informazioni sui server e identificare i modelli di comportamento del target.

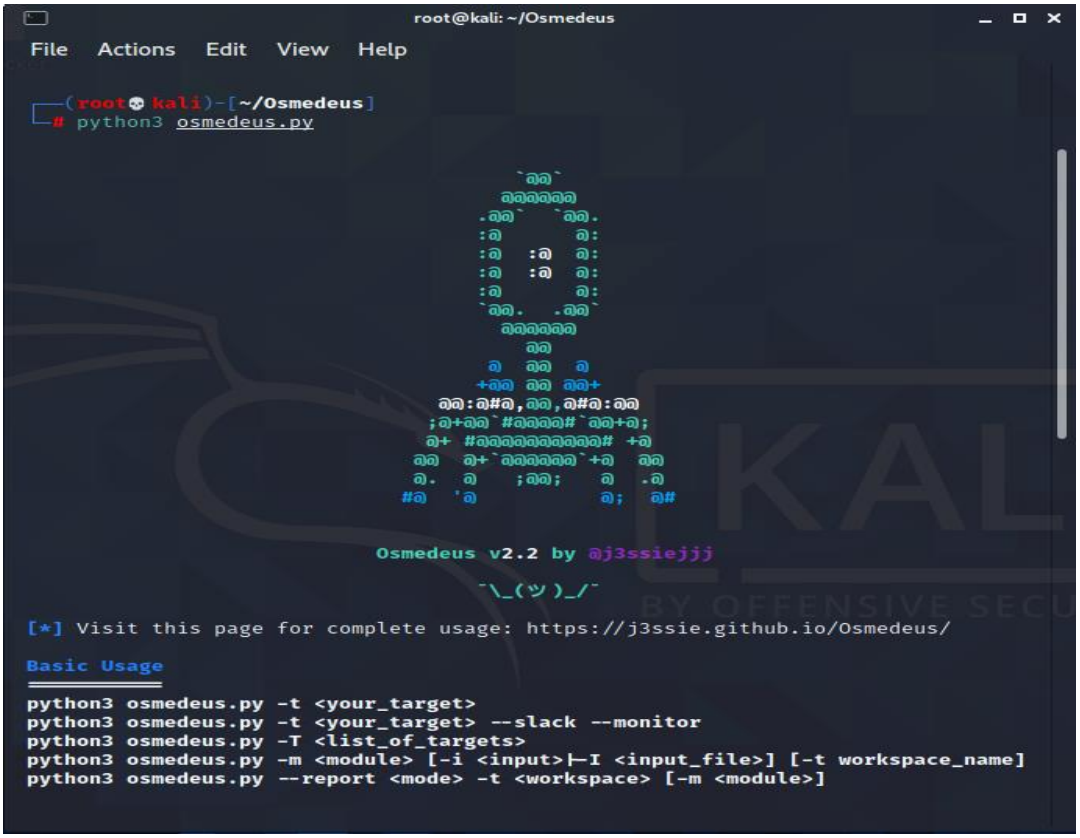

Figura 17 Home e logo Osmedeus

2.4 OpenVas

OpenVAS è uno scanner di vulnerabilità completo. È un framework di diversi servizi e strumenti che offre una soluzione di scansione / gestione delle vulnerabilità completa e potente. Le sue funzioni includono test non autenticati, test autenticati, vari protocolli Internet e industriali di alto e basso livello, ottimizzazione delle prestazioni per scansioni su larga scala e un potente linguaggio di programmazione interno per implementare qualsiasi tipo di test di vulnerabilità.

Lo scanner è accompagnato da un feed di test di vulnerabilità con una lunga cronologia e aggiornamenti quotidiani. Questo feed della community di Greenbone include più di 80.000 test di vulnerabilità.

Il progetto è nato nel 2005 come fork del celebre Nessus (commerciale), ed è rilasciato come Open Source alla comunità sotto la licenza GNU General Public License (GNU GPL). Di fatto è completamente libero e gratuito.

Lo scanner è sviluppato e mantenuto da Greenbone Networks dal 2009. Greenbone sviluppa OpenVAS come parte della famiglia di prodotti per la gestione delle vulnerabilità commerciali "Greenbone Security Manager" (GSM). OpenVAS è un elemento in un'architettura più ampia. In combinazione con ulteriori moduli Open Source, costituisce la soluzione Greenbone Vulnerability Management.

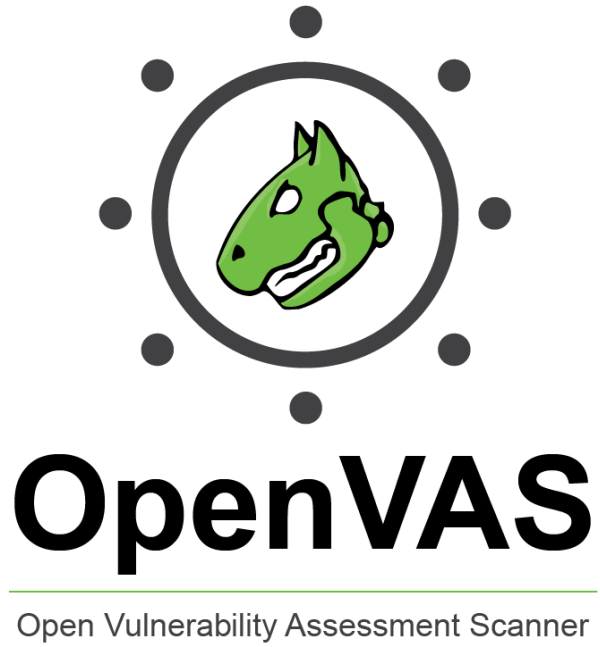

Figura 18 Logo OpenVas

2.5 RACCOON

Raccoon è uno strumento di sicurezza offensivo per la ricognizione e la raccolta di informazioni. Questo strumento di scansione delle vulnerabilità può estrarre informazioni utili sull'host di destinazione, come:

- Record DNS (dettagli / mappatura), record WHOIS, scansione delle porte, fuzzing URL, dati TLS (versione, cifrature, certificati), enumerazione dei sottodomini, informazioni WAF (Web Application Firewall);
- Informazioni sul server Web, informazioni CMS, enumerazione di file sensibili, indirizzi e-mail, robots.txt, moduli HTML disponibili, cookie, informazioni sulla mappa del sito, ecc.

Le scansioni di Raccoon sono per lo più indipendenti, quindi non si basano sui risultati l'una dell'altra.

Grazie al Python's asyncio, le scansioni vengono eseguite in modo asincrono. Questo strumento supporta anche Tor/proxy per il routing anonimo. Controlla il repository SecLists per vedere gli elenchi di parole predefiniti per il fuzzing degli URL e la scoperta di sottodomini.

Figura 19 Logo Raccoon

2.6 DVWA

DVWA (Damn Vulnerable Web Application) è un'applicazione web PHP/MySQL. Con questa applicazione i professionisti della sicurezza e gli hacker etici mettono alla prova le loro abilità ed eseguono test in un ambiente legale. Aiuta anche gli sviluppatori web a comprendere meglio i processi di sicurezza delle applicazioni web e gli insegnanti/studenti per insegnare/apprendere la sicurezza delle applicazioni web in un ambiente sicuro.

L'obiettivo del DVWA è quello di permettere lo studio di alcune delle più comuni vulnerabilità web con vari livelli di difficoltà in maniera sicura e legale.

In altre parole, si ha la possibilità di creare in poco tempo un computer Target dove poter ricercare, vedere e sfruttare alcune delle vulnerabilità insite in applicazioni web-based, in maniera del tutto sicura.

Come suggerisce il nome, DVWA ha molte vulnerabilità web. Ogni vulnerabilità ha quattro diversi livelli di sicurezza, basso, medio, alto e impossibile. I livelli di sicurezza rappresentano una sfida per l'"attaccante" e mostrano anche come ogni vulnerabilità può essere contrastata attraverso la codifica software sicura.

- **Impossibile:** Questo livello è il più difficile ed offre sfide che si dovranno affrontare nel mondo reale.

- **Alto:** Questo livello di vulnerabilità fornisce all'utente un esempio di come proteggere la vulnerabilità attraverso metodi di codifica sicura. Permette all'utente di capire come si può misurare la vulnerabilità. Questo livello di sicurezza dovrebbe essere inattaccabile, ma, come tutti sappiamo, non è sempre così.

- **Medio:** Lo scopo di questo livello di sicurezza è quello di dare all'"attaccante" una sfida nello sfruttamento delle vulnerabilità e servire anche come esempio di cattive pratiche di codifica e di sicurezza.

- **Basso:** Questo livello di sicurezza ha lo scopo di simulare un sito web senza alcuna sicurezza implementata nella loro codifica. Dà agli "aggressori" la possibilità di perfezionare le loro capacità.

Figura 20 Logo DVWA

2.6.1 Installazione e configurazione di DVWA

1. Download del file ISO di DVWA dal link https://dvwa.co.uk/
2. Creare una macchina virtuale con sistema operativo Linux (in questo studio è stato scelto Linux 2.6 64 bit;
3. Nelle impostazioni della VM appena creata, su archiviazione selezionare il controller IDE e montare la ISO di DVWA;
4. Nelle impostazioni di rete della VM selezionare “Scheda solo host”;
5. Avviare la macchina e selezionare “Live CD Option”;
6. Ora il server è pronto per essere utilizzato, digitando “ifconfig” verrà mostrato a schermo l’indirizzo IP;
7. Copiato l’IP in un browser, si accederà alla homepage di DVWA così da esser sicuri che il server è live;
    a. Le credenziali di default: **Username:** “*admin*”, **Password:** “*password*”.

2.7 METASPLOITABLE 2

La macchina virtuale Metasploitable è una versione intenzionalmente vulnerabile di Linux progettata per testare strumenti di sicurezza e dimostrare vulnerabilità comuni. La versione 2 di questa macchina virtuale viene fornita con ancora più vulnerabilità rispetto all'immagine originale.

Metasploitable Project è creato e mantenuto dalla community rapid7 (Metasploit-FrameWork Community). Metasploitable è originariamente progettato per Metasploit Framework Testing. In parole semplici, Metasploitable è un sistema operativo basato su Linux, progettato appositamente per esercitarsi con abilità di test di penetrazione, abilità di sicurezza di rete, abilità di Metasploit-Framework e molti altri.

L'obiettivo principale di Metasploitable è fornire un sistema operativo vulnerabile, che possa essere utilizzato da nuovi studenti di networking, nuovi penetration tester, hacker, ricercatori di rete per esercitare le loro abilità in un ambiente sicuro. Con Metasploitable, chiunque può facilmente configurare il proprio laboratorio personale di sicurezza per testare le proprie abilità o apprendere cose nuove in un ambiente sicuro.

2.7.1 Installazione e configurazione di Metasploitable 2

1. Scaricare il file Metasploitable.vmdk al link https://sourceforge.net/projects/metasploitable/files/latest/download

2. Creare una macchina virtuale con sistema operativo Linux;

3. Al momento della scelta dell'hard disk selezionare "Usa un file di disco fisso virtuale esistente", selezionare il file .vmdk precedentemente scaricato e completare la creazione della VM;

4. Avviare la macchia virtuale e attendere la fase di boot;

    a. Le credenziali di default sono: **Username:** "*msfadmin*", **Password:** "*msfadmin*".

Figura 21 Home e logo di Metasploitable 2

2.8 WINDOWS SERVER 2012 R2

In informatica Windows Server è il marchio di una famiglia di sistemi operativi per sistemi server realizzati da Microsoft Corporation. Il suo utilizzo è tipico di reti informatiche di qualsiasi numero di host per la centralizzazione di diverse funzioni di networking ad uso Internet ed intranet, come web server, posta elettronica, DHCP, DNS, centralizzazione di risorse di memorizzazione, servizio di directory, file sharing, backup automatici, stampa, accesso remoto, ecc.… dette Reti Windows.

Windows Server 2012 (precedentemente noto con il nome in codice Windows Server 8) è una versione server del sistema operativo Microsoft Windows, reso disponibile al pubblico il 4 settembre 2012, mentre il 18 ottobre 2013 ne è stato rilasciato l'aggiornamento Windows Server 2012 R2. Windows Server 2012 R2 è la versione server di Windows 8.1. Introduce una nuova interfaccia grafica nel linguaggio di design Moderni UI, ed è la prima versione di Windows Server a non avere il supporto per i computer basati su Itanium a partire da Windows NT 4.0.

Figura 22 Logo Windows Server 2012 R2

2.8.1 Installazione e configurazione di Windows Server 2012 R2

1. Nel sito ufficiale di Microsoft (https://www.microsoft.com/en-gb/evalcenter/evaluate-windows-server-2012-r2) è possibile scaricare la versione di prova per 180 giorni del sistema operativo, quindi selezionare il formato VHD e aspettare il termine del download;
2. Successivamente con VirtualBox creare una nuova VM con Windows 8.1 (64-bit) o Windows 2012 (64-bit);
3. Al momento della scelta dell'hard disk, invece di crearne uno nuovo, selezionare il file VHD scaricato precedentemente;
4. Terminata la creazione della VM, ora è possibile avviarla e dopo una breve configurazione verrà chiesto di scegliere una password da amministratore;
5. Quando richiesto, andare nella voce Inserimento -> Tastiera -> Inserisci CTRL-ALT-DEL per visualizzare la pagina del login;
6. Effettuato il login, si aprirà automaticamente il Server Manager e da qui si può configurare il server;
7. Nella voce Manage, in alto a destra, selezionare "Add Roles and Features" e installare tutti i servizi Web necessari.

2.9 METASPLOITABLE 3

Metasploitable3 è una VM costruita da zero con una grande quantità di vulnerabilità di sicurezza. È concepito per essere utilizzato come target per testare gli exploit.

Le versioni precedenti di Metasploitable venivano distribuite come snapshot della VM in cui tutto era impostato e salvato in quello stato. Metasploitable 3 introduce un nuovo approccio: la creazione dinamica dell'immagine della VM. Utilizza Packer, Vagrant e molti script per ottenere una VM sfruttabile completamente funzionante in pochi minuti.

Ci sono molti vantaggi in questo nuovo metodo di costruzione. Si possono facilmente applicare gli stessi exploit su più sistemi operativi, compatibile per più piattaforme di virtualizzazione e, soprattutto, accettare contributi dalla comunità.

Tuttavia nonostante sia possibile creare da soli il file ISO di Metasploitable 3 ci sono numerosi problemi riguardanti la versione dei software da usare durante il processo perciò in questo studio è stato preso un file ISO già creato dalla community rapid7. È stato reso disponibile questo file proprio per venire incontro a chi ha problemi durante il processo di creazione.

2.9.1 Installazione e configurazione di Metasploitable 3

1. Scaricare il file OVA usando il link https://github.com/brimstone/metasploitable3/releases/tag/0.1.4

2. Aprire VirtualBox e fare click su Importa e scegliere il file scaricato in precedenza;

3. Al termine del percorso guidato della creazione della VM sarà possibile avviarla;

4. Al momento dell'avvio attendere diversi minuti per completare la configurazione, dopodiché la VM è pronta per l'utilizzo.

Figura 23 Logo Metasploitable 3

3 JOK3R

3.1 COS'È JOK3R

Jok3r è un'applicazione Python3 CLI (Command Line Interface) progettato per aiutare i penetration tester a indagare sui difetti nell'infrastruttura di rete o nei siti web/applicazioni web.

L'obiettivo principale di Jok3r è quello di automatizzare la maggior parte delle attività durante l'identificazione/sfruttamento delle vulnerabilità "di base" trovate sui più comuni servizi TCP/UDP così come sulle più comuni applicazioni web (come Apache, Nginx, WordPress, Joomla, Node.JS, Ruby on Rails e altri) in modo da risparmiare tempo su tutto ciò che può essere automatizzato durante il processo di pentesting di rete /web, per concentrarsi su cose più importanti e interessanti.

Per ottenere ciò, combina 50+ strumenti di hacking open source per eseguire vari controlli di sicurezza su tutti i servizi di rete comuni come per esempio Port scanning, Fingerprinting, scansione delle vulnerabilità, sfruttamento delle vulnerabilità rilevate, Bruteforce attack se necessario, Post-exploitation, e molto altro.

Figura 24 Logo di Jok3r

3.2 FEATURES

Gestione della Toolbox

- **Selezione di strumenti**: presenza di oltre 50 strumenti e script open source, da varie fonti.
- **Basato su Docker**: applicazione confezionata in un'immagine Docker che esegue il sistema operativo Kali, disponibile su Docker Hub.
- **Pronto per l'uso**: tutti gli strumenti e le dipendenze installati, basta estrarre l'immagine Docker ed eseguire un nuovo container.
- **Aggiornamenti semplificati**: mantieni facilmente aggiornata l'intera casella degli strumenti eseguendo un solo comando.
- **Facile personalizzazione**: aggiungi / rimuovi facilmente strumenti da un semplice file di configurazione.

Valutazione della sicurezza dell'infrastruttura di rete

- **Molti servizi supportati**: indirizza i servizi TCP / UDP più comuni (HTTP, FTP, SSH, SMB, Oracle, MS-SQL, MySQL, PostgreSQL, VNC, ecc.).
- **Combina la potenza degli strumenti**: ogni controllo di sicurezza viene eseguito da uno strumento della toolbox. Gli attacchi vengono eseguiti concatenando i controlli di sicurezza.
- **Consapevolezza del contesto**: i controlli di sicurezza da eseguire vengono selezionati e adattati in base al contesto del target (ovvero tecnologie rilevate, credenziali, vulnerabilità, ecc.).
- **Ricognizione**: viene eseguito un fingerprint automatico (rilevamento del prodotto) dei servizi targhettati.
- **Ricerca CVE**: quando vengono rilevati i nomi dei prodotti e le relative versioni, viene eseguita una ricerca delle vulnerabilità sui database CVE in linea (utilizzando Vulner e CVE Details).
- **Scansione delle vulnerabilità**: controlla automaticamente le vulnerabilità comuni e tenta di eseguire alcuni exploit (auto-pwn).
- **Brute-force**: controlla automaticamente le credenziali predefinite / comuni sul servizio ed esegue un dictionary attack (attacco a dizionario), se necessario. Le wordlists sono ottimizzate in base ai servizi targhettati.
- **Test post-autenticazione**: esegue automaticamente alcuni controlli post-exploitation quando sono state trovate credenziali valide.

Valutazione della sicurezza web

- **Grande attenzione all'HTTP**: sono supportati più di 60 diversi controlli di sicurezza mirati all'HTTP.
- **Rilevamento tecnologia Web**: il fingerprinter basato su Wappalyzer viene eseguito prima dei controlli di sicurezza, consentendo di rilevare: linguaggio di programmazione, Framework, libreria JS, CMS, Web e Application Server.
- **Server exploitation**: scansiona e/o sfrutta automaticamente le vulnerabilità più critiche (es. RCE) sui server web e applicativi (es. JBoss, Tomcat, Weblogic, Websphere, Jenkins, ecc.).
- **Scansione delle vulnerabilità CMS**: esegue automaticamente le scan delle vulnerabilità sui CMS più comuni (Wordpress, Drupal, Joomla, ecc.).

Database locale e reportistica

- **Database locale**: i dati relativi ai target sono organizzati per missioni (workspaces) in un database SQLite locale che viene mantenuto aggiornato durante i test di sicurezza.
- **Shell interattiva simile a Metasploit**: accedi al database tramite una shell interattiva con diversi comandi incorporati.
- **Importa obiettivi da Nmap**: aggiungi obiettivi a una missione manualmente o caricando i risultati di Nmap.
- **Importa obiettivi da Shodan**: aggiungi manualmente obiettivi a una missione da Shodan (è necessaria la key API Shodan).
- **Accesso a tutti i risultati**: tutti gli output dei controlli di sicurezza, le credenziali rilevate e le vulnerabilità vengono archiviati nel database e sono facilmente accessibili.
- **Report**: genera report HTML completi con riepilogo degli obiettivi, screenshot web e tutti i risultati dei test di sicurezza.

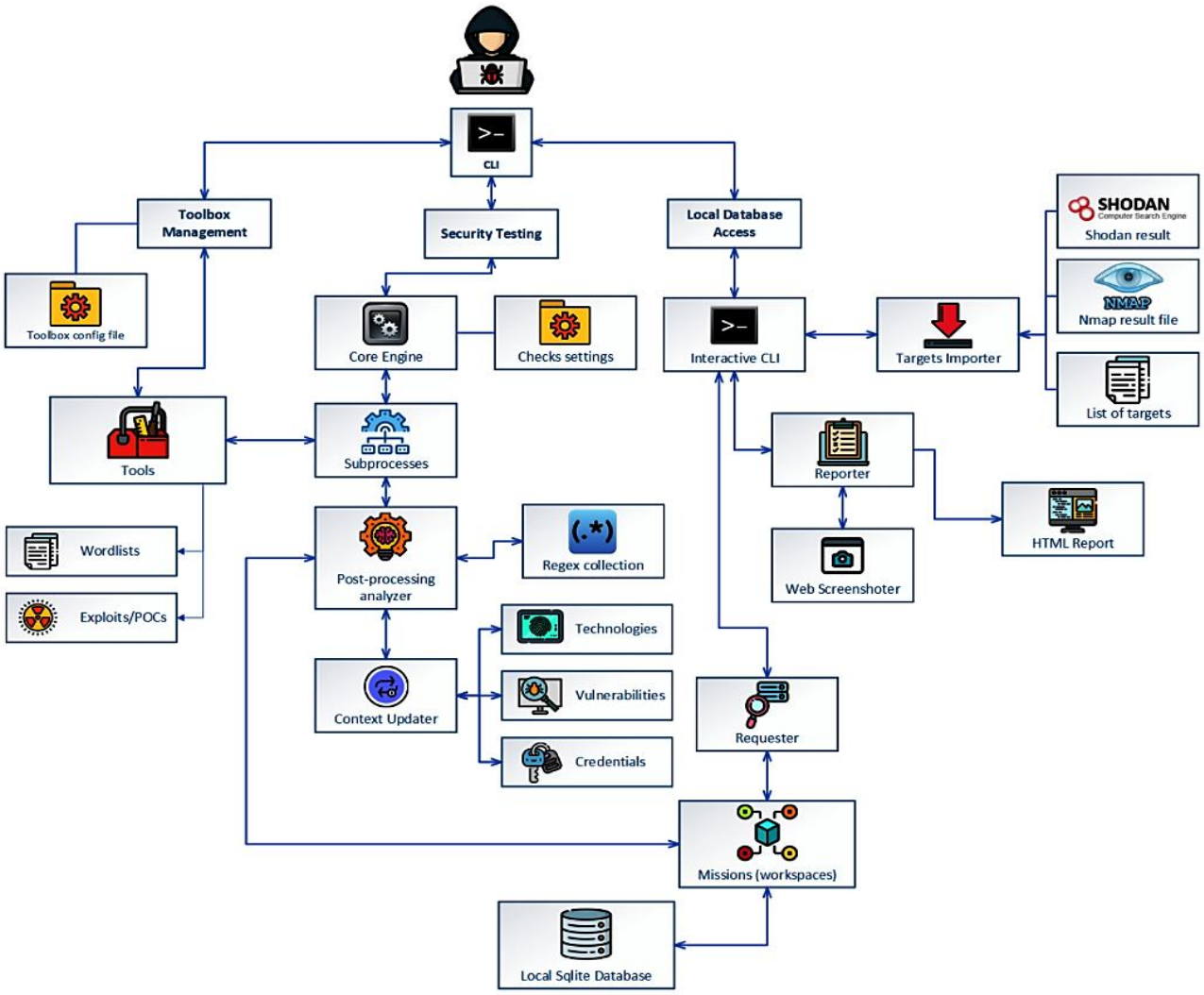

Figura 25 Schema dell'architettura di Jok3r

Servizi supportati e controlli di sicurezza

- AJP (predefinito 8009 / tcp)
- FTP (predefinito 21 / tcp)
- HTTP (predefinito 80 / tcp)
- Java-RMI (predefinito 1099 / tcp)
- JDWP (predefinito 9000 / tcp)
- MSSQL (predefinito 1433 / tcp)
- MySQL (predefinito 3306 / tcp)
- Oracle (predefinito 1521 / tcp)
- PostgreSQL (predefinito 5432 / tcp)
- RDP (predefinito 3389 / tcp)
- SMB (predefinito 445 / tcp)
- SMTP (predefinito 25 / tcp)
- SNMP (predefinito 161 / udp)
- SSH (predefinito 22 / tcp)
- Telnet (predefinito 21 / tcp)
- VNC (predefinito 5900 / tcp)

3.3 INSTALLAZIONE

Il modo consigliato per utilizzare Jok3r è fare il pull dell'immagine Docker in modo da non preoccuparsi dei problemi per installare singolarmente i vari strumenti di hacking con i requisiti necessari.

Pull dell'immagine Docker Jok3r:

```
sudo docker pull koutto/jok3r
```

Eseguire un nuovo Docker container:

```
sudo docker run -i -t --name jok3r-container -w/root/
jok3r -e DISPLAY = $ DISPLAY -v/tmp/.X11-unix:/tmp/.X11-
unix --shm-size 2g - net = host koutto/jok3r
```

Info:

-e DISPLAY = $ DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix è richiesto per poter avviare l'applicazione GUI dal Docker container (ad es. aprire il browser web per leggere i report). Richiede l'esecuzione di xhost + local: root sull'host.

--shm-size 2g viene utilizzato per aumentare le dimensioni della memoria condivisa, è necessario per evitare arresti anomali del browser Web durante la lettura dei report dal contenitore Docker.

--net = host è necessario per condividere l'interfaccia dell'host.

A questo punto è tutto pronto per l'avvio e quindi basta eseguire il comando

```
sudo docker start -i jok3r-container
```

Infine è presente un comando per aprire più shell all'interno del container, molto importante per poter gestire contemporaneamente sia il database che la scansione:

```
sudo docker exec -it jok3r-container bash
```

3.3.1 Comandi utili

Gestione della Toolbox:

- Mostra tutti gli strumenti nella toolbox:

```
python3 jok3r.py toolbox –show-all
```

- Installa tutti gli strumenti nella toolbox (già fatto nel container Docker):

```
python3 jok3r.py toolbox --install-all --auto
```

- Aggiorna tutti gli strumenti toolbox e chiede di controllare l'aggiornamento ogni volta che termina l'aggiornamento di uno strumento:

```
python3 jok3r.py toolbox --update-all
```

- Aggiorna tutti gli strumenti toolbox senza attendere l'input di conferma dell'user dopo ogni aggiornamento installato:

```
python3 jok3r.py toolbox --update-all –auto
```

Informazioni utili sui servizi:

- Elenca i servizi supportati:

```
python3 jok3r.py info --services
```

- Mostra i controlli di sicurezza per un determinato servizio:

```
python3 jok3r.py info --checks <service>
```

- Mostra i profili di attacco supportati per un dato servizio:

```
python3 jok3r.py info --attack-profiles <service>
```

- Mostra i prodotti supportati per tutti i servizi:

```
python3 jok3r.py informazioni –prodocts
```

Accesso al database e reportistica:

- Seleziona una missione:

```
python3 jok3r.py db

jok3rdb [default]> mission prova

[*] Selected mission is now prova
```

- Importa host/servizi dai risultati di Nmap (XML) nell'ambito della missione:

```
jok3rdb [prova]> nmap results.xml
```

- Importa host/servizi da Shodan results (IP) nell'ambito della missione:

```
jok3rdb [prova]> ip shodan
```

- Visualizza servizi, host, prodotti rilevati e credenziali registrate nella missione selezionata:

```
jok3rdb [prova]> services

jok3rdb [prova]> hosts

jok3rdb [prova]> products

jok3rdb [prova]> creds
```

- Cerca una stringa nei risultati della scansione nella missione selezionata:

```
jok3rdb [prova]> results --search "<search_string>"
```

- Visualizza le vulnerabilità rilevate automaticamente dagli output delle scansioni nella missione selezionata:

```
jok3rdb [prova]> vulns
```

- Genera report HTML per la missione selezionata:

```
jok3rdb [prova]> report
```

Comandi per scansione:

- Esegui tutti i controlli di sicurezza su un URL in modalità interattiva e aggiungi risultati alla missione "prova" senza interazione da parte dell'utente:

```
python3 jok3r.py attack -t https://www.example.com/ --
                    add2db prova --fast
```

- Esegui solo i controlli di sicurezza "recon" e "vulnscan" su un servizio FTP e aggiungi risultati alla missione:

```
python3 jok3r.py attack -t 192.168.1.142:21 -s ftp --cat-only
                recon, vulnscan --add2db prova
```

Target multipli:

- Esegui tutti i controlli di sicurezza su tutti i servizi nella missione data e archivia i risultati nel database:

```
            python3 jok3r.py attack -m prova --fast
```

- Esegui controlli di sicurezza solo sui servizi FTP in esecuzione sulle porte 21 / tcp e 2121 / tcp dalla missione:

```
python3 jok3r.py attack -m prova -f "port = 21,2121; ser-
                    vice = ftp" --fast
```

- Esegui controlli di sicurezza solo sui servizi FTP in esecuzione sulle porte 2121 / tcp e su tutti i servizi HTTP su 192.168.1.42 dalla missione:

```
python3 jok3r.py attack -m mayhem -f "port = 2121; service
      = ftp" -f "ip = 192.168.1.42; service = http"
```

3.4 UTILIZZO

Quando si utilizza Jok3r è consigliabile aprire due shell di questo tool. Questo perché è possibile che siano necessari diversi minuti prima che la scansione venga completata e dato che il database viene aggiornato in modo simultaneo con il procedere della scansione, risulta pratico avere una shell per creare, gestire e visualizzare il database mentre la seconda per avviare la scansione verso il target.

Ecco un tipico esempio di utilizzo di *JoK3r*:

1. Si può usa re Nmap per fare una scansione sui server del target (opzionale)

2. Creare una nuova missione nel database locale:

```
python3 jok3r.py db

jok3rdb [default]> mission prova

jok3rdb[prova]>
```

3. Se è stata effettuata la scansione con Nmap, importare i risultati della scan nel database:

```
jok3rdb[prova]> nmap results.xml
```

4. Ora si può eseguire la scansione verso l'obiettivo eseguendo il comando:

```
python3 jok3r.py attack -t 192.168.1.42:1433 --add2db
prova --fast
```

5. È possibile visualizzare i risultati completi della scansione in tempo reale quando gli strumenti vengono eseguiti o successivamente dal database utilizzando il seguente comando:

```
jok3rdb [prova]> results
```

6. Inoltre in qualsiasi momento è possibile visualizzare i dati estratti automaticamente dagli output della scansione, ovvero prodotti rilevati, credenziali e vulnerabilità.

```
jok3rdb [prova]> products

[...]

jok3rdb [prova]> creds

[...]

jok3rdb [prova]> vulns

[...]
```

7. Una volta che la scansione è terminata si può generare un report HTML con un riepilogo di tutti i target della missione, screenshot delle pagine web per i servizi HTTP e output completi di tutti i controlli sulla sicurezza che sono stati eseguiti:

```
jok3rdb [prova]> report
```

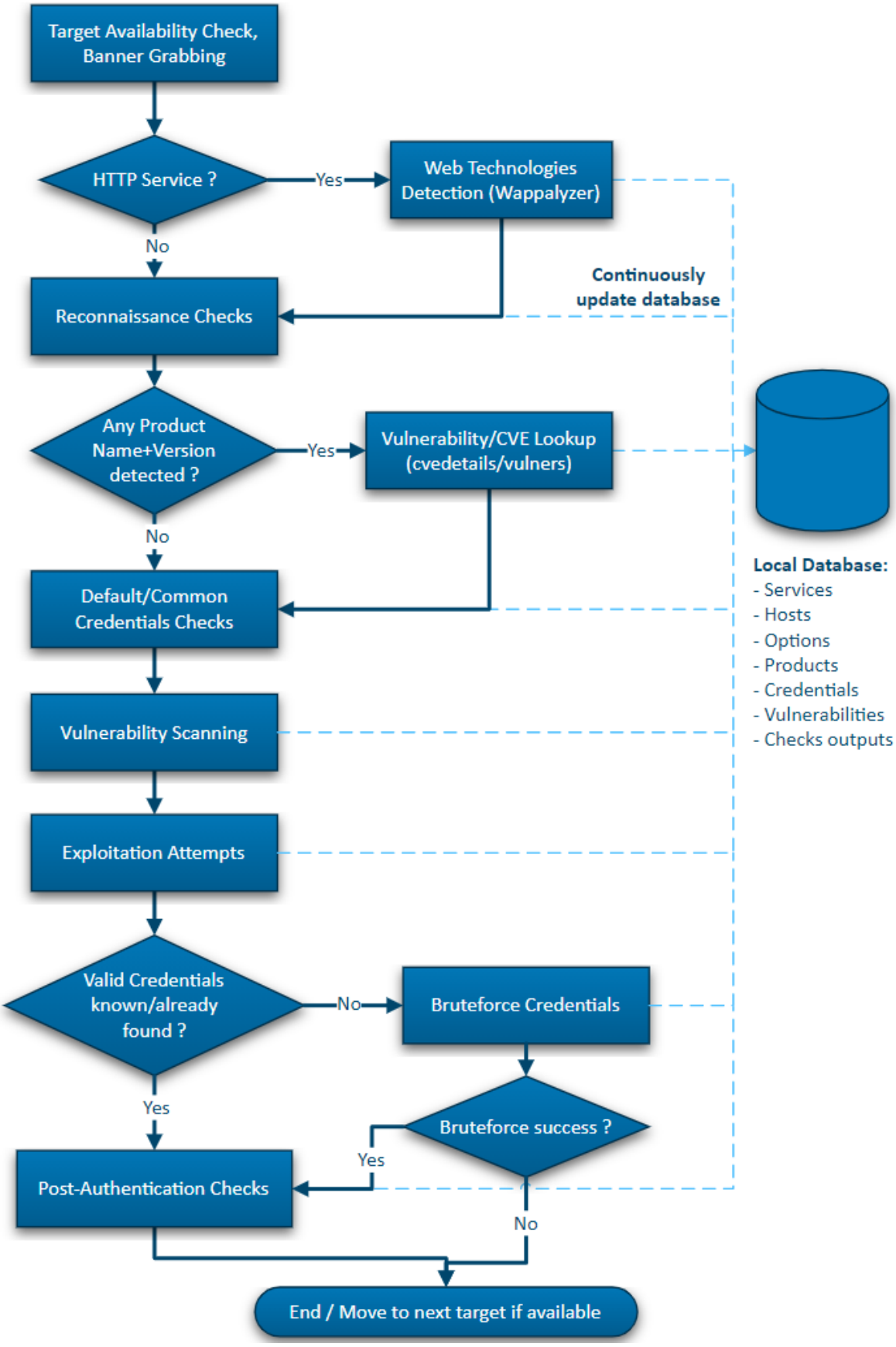

Figura 26 Workflow di Jok3r

3.5 REPORT E RISULTATI

Quando la scansione verso l'obbiettivo è conclusa, per avere una visione più dettagliata e analitica dei risultati ottenuti si deve usare il comando per la creazione di un file in formato HTML.

```
jok3rdb [prova]> report
```

Quando la pagina HTML verrà visualizzata nel browser si avrà un riepilogo di tutti i servizi Web scansionati.

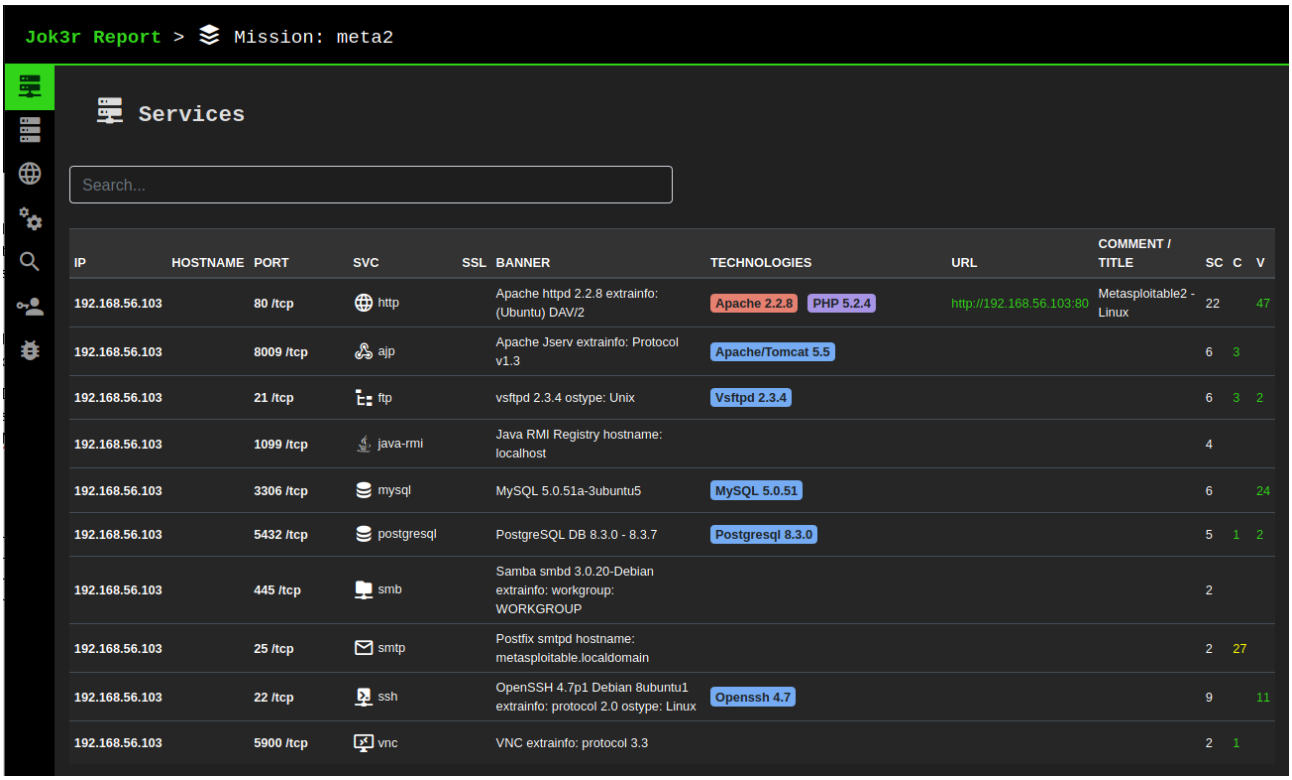

Figura 27 Elenco dei servizi ottenuti dalla scansione

Analizzando questa prima schermata (**Figura 27**), viene mostrata una tabella con tutti i servizi web rilevati dalla scansione. Nella tabella oltre all'indirizzo IP del bersaglio e le porte analizzate, viene riportato anche il tipo di servizio ad essa associato. Inoltre viene mostrato il tipo di tecnologia utilizzata da ogni porta e infine la parte più importante, cioè il numero di "Security check" eseguiti (SC), il numero delle eventuali credenziali trovate (C) e per finire il totale delle vulnerabilità scovate.

Per proseguire con l'analisi più dettagliata delle vulnerabilità di ogni servizio basta selezionare con un click una riga della tabella (**Figura 27**).

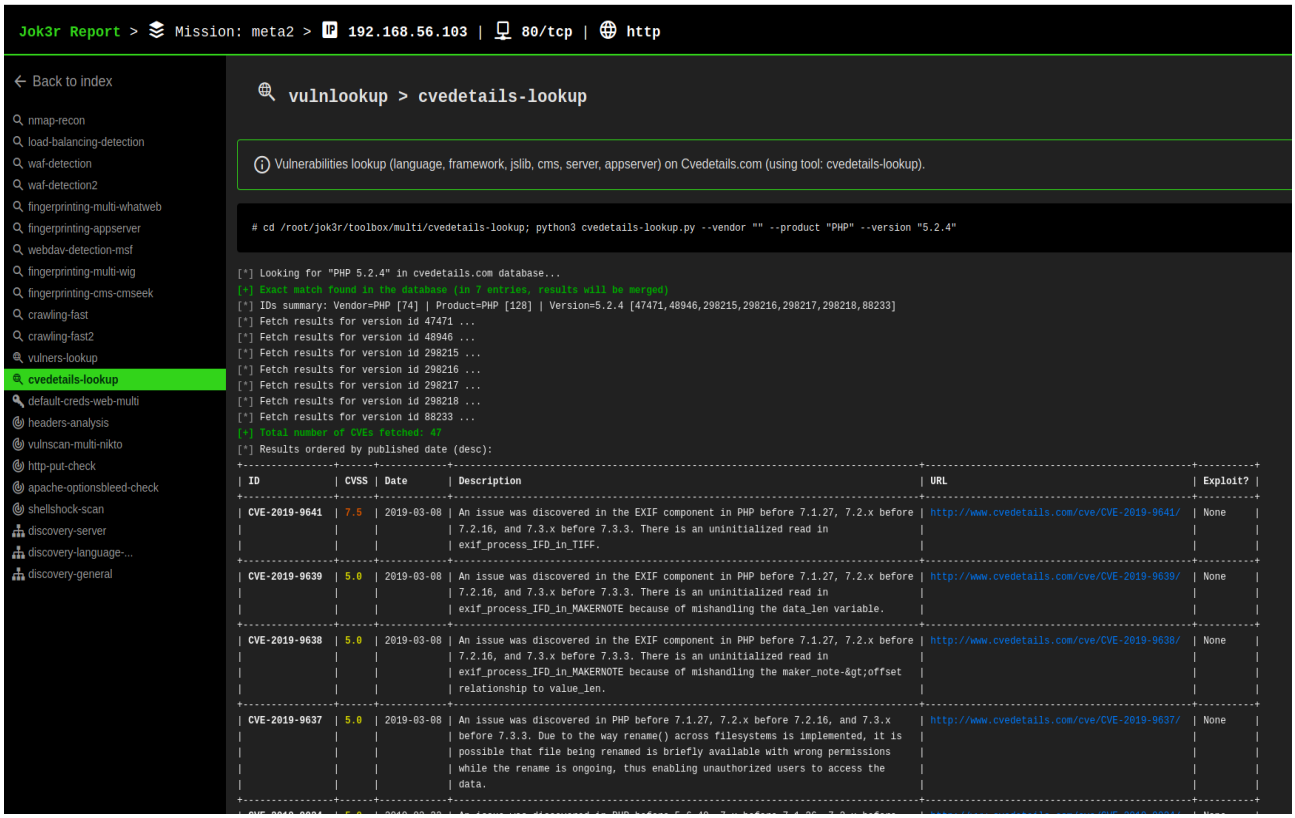

Figura 28 Elenco dettagliato delle vulnerabilità trovate del servizio selezionato

Si verrà reindirizzati in una nuova pagina (**Figura 28**) dove a destra vengono riportati tutti gli strumenti utilizzati da jok3r durante la scansione per quel servizio e selezionandone uno nella lista, ad esempio “cvedetails-lookup”, viene mostrato l’elenco delle vulnerabilità trovate con il codice CVE ad essa associato, il CVSS (Common Vulnerability Scoring System) ovvero il sistema comune di valutazione della gravità delle vulnerabilità, una breve descrizione della vulnerabilità e un link che rimanda a dati più approfonditi per studiarla più nel dettaglio.

Anche se non viene riportato il link per l’exploit della vulnerabilità, bisogna tenere in conto che facendo una ricerca altrove è possibile trovare un modo per sfruttarla, ma per ora lo strumento non offre questa funzione.

Ovviamente selezionando un altro strumento dalla lista verranno riportate informazioni aggiuntive come per esempio fingerprint, la presenza di un Web application firewall (WAF), eventuali credenziali salvate, tecnologie del database, ecc.

Tornando nell’indice mostrato nella **Figura 27** si possono controllare ulteriori risultati come l’host del target (**Figura 29**), uno screen della sua interfaccia Web (**Figura 30**), la lista dei prodotti in uso con le relative versioni (**Figura 31**).

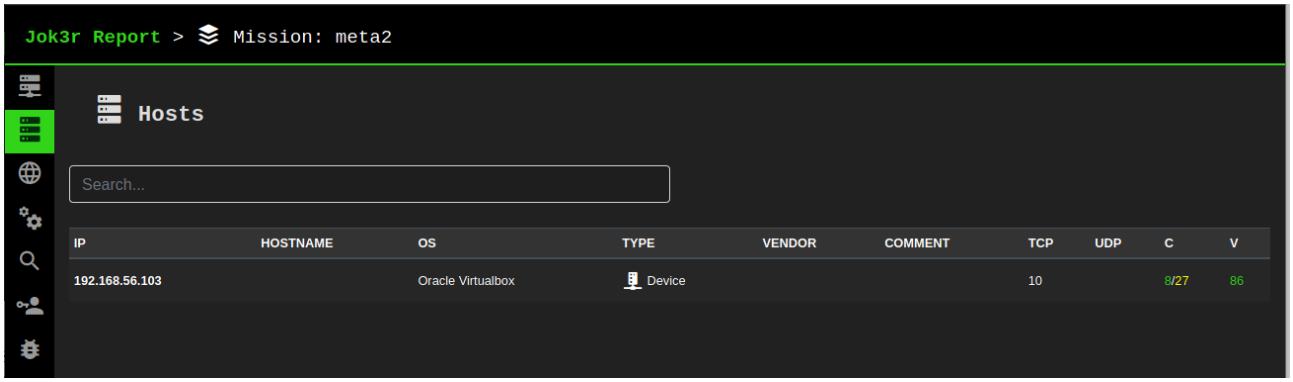

Figura 29 Elenco degli host del target

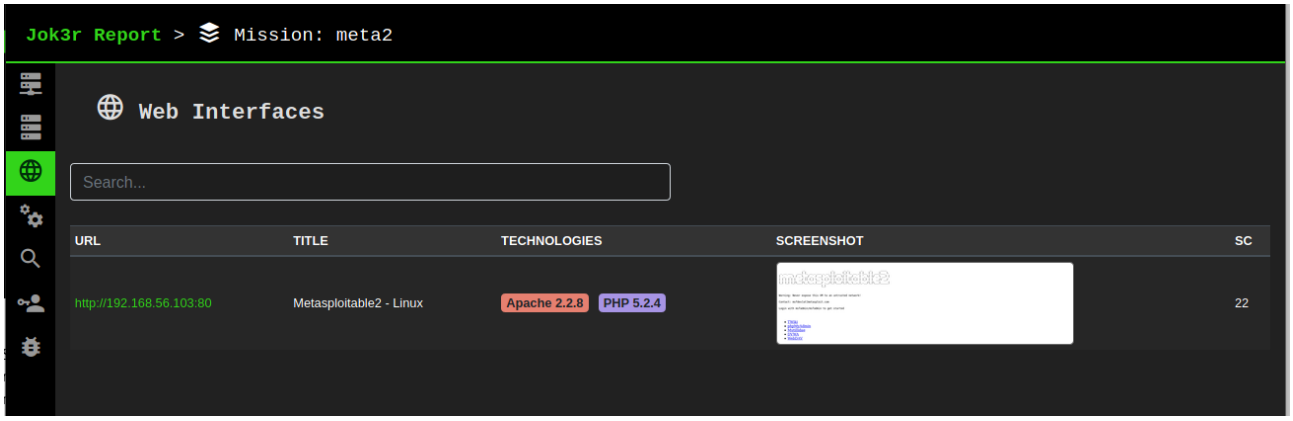

Figura 30 Screen dell'interfaccia Web del target

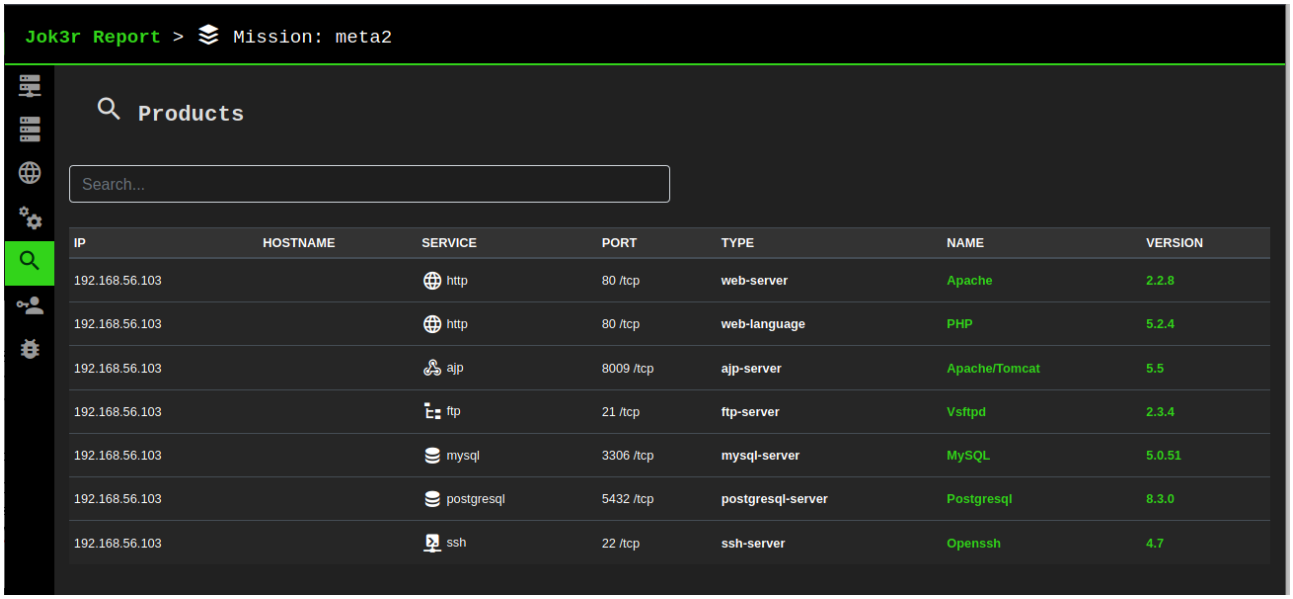

Figura 31 Elenco dei prodotti utilizzati dal target

Proseguendo con l'analisi del report, si arriva ad una pagina dove vengono raccolte, in maniera ordinata, tutte le credenziali scovate. Le credenziali scritte in verde rappresentano quelle "certe" che possono essere ritenute valide e funzionanti, mentre per quanto riguarda quelle scritte in giallo sono incomplete e perciò necessitano di ulteriori test o strumenti mirati. (**Figura 32**)

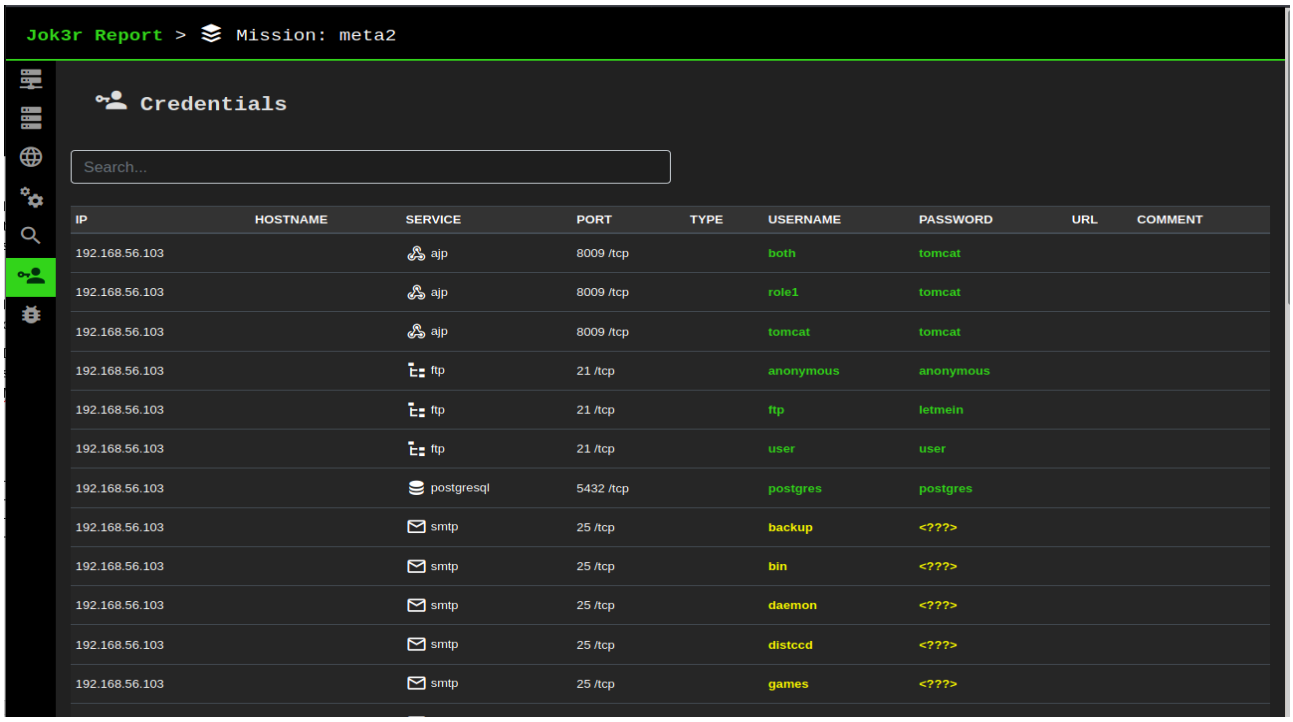

Figura 32 Elenco delle credenziali trovate

Concludendo l'analisi del report, viene messa a disposizione una pagina dove una tabella raccoglie tutte le vulnerabilità rilevate sul target scansionato. (**Figura 33**)

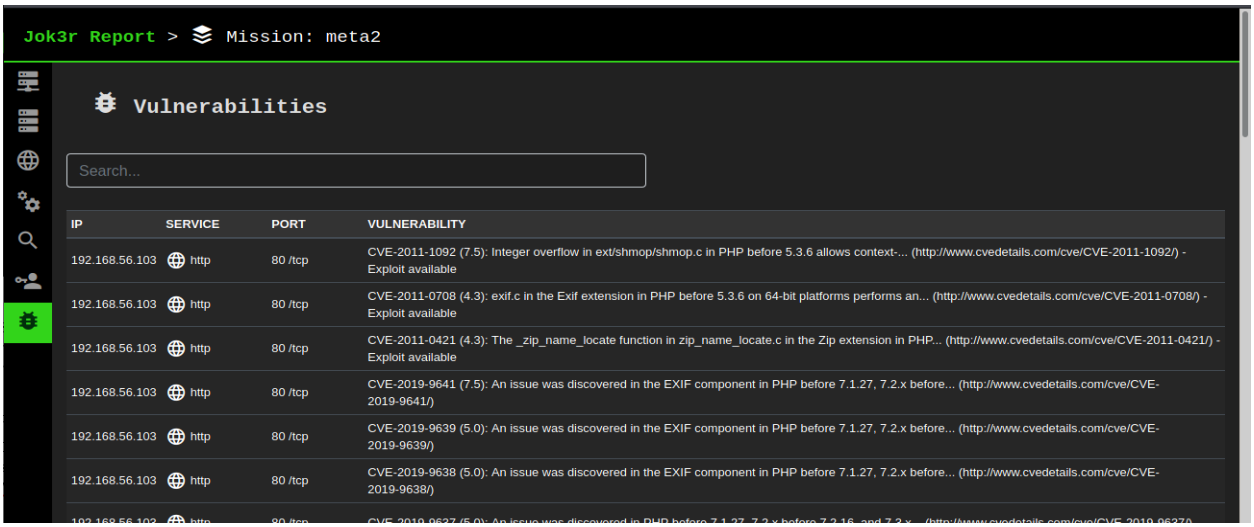

Figura 33 Elenco di tutte le vulnerabilità trovate di ogni servizio del target

Di seguito viene riportata una tabella (**Tabella 2**) riassuntiva che raccoglie in modo generale i risultati ottenuti scansionando i sistemi descritti nel Capitolo 3 (DVWA, Metasploitable 2, Windows Server 2012 R2, Metasploitable 3) così da poter fare un rapido confronto tra di loro.

Nella tabella vengono elencati i servizi più importanti e quelli supportati da Jok3r come ad esempio HTTP, FTP e così via, e per ogni target scansionato vengono riportati tre dati fondamentali:

- **Security Checks (SC),** ovvero il numero degli strumenti utilizzati da Jok3r durante la scansione per quel tipo di servizio;
- **Credentials (C),** il numero delle credenziali che sono state trovate;
- **Vulnerabilities (V),** il numero delle vulnerabilità che sono state scoperte in quel servizio.

Tabella 2 Risultati ottenuti dalle scansioni con Jok3r sui target presi in esame

	DVWA			**Metasploitable 2**			**Windows Server 2012 R2**			**Metasploitable 3**		
	SC	**C**	**V**	**SC**	**C**	**V**	**SC**	**C**	**V**	**SC**	**C**	**V**
HTTP (80 / tcp)	24	0	34	22	0	47	24	0	1	24	0	2
AJP (8009 / tcp)	/	/	/	6	3	0	/	/	/	/	/	/
FTP (21 / tcp)	6	0	1060	6	3	2	4	0	14	4	0	14
Java-RMI (1099 / tcp)	/	/	/	4	0	0	/	/	/	/	/	/
MySQL (3306 / tcp)	4	0	0	6	0	24	/	/	/	/	/	/
PostgreSQL (5432 / tcp)	/	/	/	5	1	2	/	/	/	/	/	/
SMB (445 / tcp)	/	/	/	2	0	0	2	0	0	2	0	1
SMTP (25 / tcp)	/	/	/	2	27	0	/	/	/	/	/	/
SSH (22 / tcp)	9	0	10	9	0	11	/	/	/	/	/	/
VNC (5900/ tcp)	/	/	/	2	1	0	/	/	/	/	/	/
RDP (3389 / tcp)	/	/	/	/	/	/	/	/	/	2	0	1

Dall'analisi della **Tabella 2**, si nota subito che il target su cui sono state trovate più vulnerabilità con la scansione fatta da Jok3r è Metasploitable 2. Questo perché il tipo di sistema è abbastanza datato, infatti Metasploitable 2 è stato rilasciato nel giugno 2012 ma si basa su un sistema operativo del 2008 (Linux 2.6.24).

Prendendo il servizio HTTP, Hypertext Transfer Protocol, (un protocollo a livello applicativo usato come principale sistema per la trasmissione d'informazioni sul web ovvero in un'architettura tipica Client-Server) si può fare un confronto con i risultati ottenuti. Nei sistemi basati su Linux, ovvero DVWA e Metasploitable 2, sono state trovate più vulnerabilità rispetto a Windows Server 2012 R2 e Metasploitable 3 che sono basati su Windows. Questo può dipendere innanzitutto dai prodotti usati e le relative versioni, infatti se la versione è troppo vecchia può essere un problema perché potrebbe mancare una patch correttiva e che porta ad un software più sicuro e stabile.

Analizzando le vulnerabilità sul lato tecnico, molte sono state riscontrate nei sistemi che fanno un largo utilizzo del linguaggio PHP, come appunto DVWA e Metasploitable 2.

Conoscere il numero esatto di vulnerabilità non è sufficiente, infatti è molto utile avere un'idea di quanto queste vulnerabilità possano avere un impatto sul sistema e ai danni che potrebbero susseguire. In questo Jok3r facilita il compito perché oltre al CVE ID viene mostrato anche il CVSS ovvero il Common Vulnerability Scoring System.

Rating	CVSS Score
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Figura 34 Metro di valutazione CVSS

Sempre prendendo come riferimento il servizio HTTP è possibile fare un valutazione del grado del rischio delle vulnerabilità riscontrate in ogni sistema secondo il criterio CVSS mostrato in **Figura 34**. Nel grafico (**Figura 35**), il 68% delle vulnerabilità hanno un rischio medio perché hanno un punteggio tra il 4.0 e il 6.9, mentre il 6% è di basso grado e il restante 26% è di alto grado. Quindi immaginando di avere un caso reale con risultati simili, il consiglio è di cominciare ad adottare una nuova politica della gestione del rischio e risolvere le problematiche più gravi fino a rendere il sistema più sicuro e stabile.

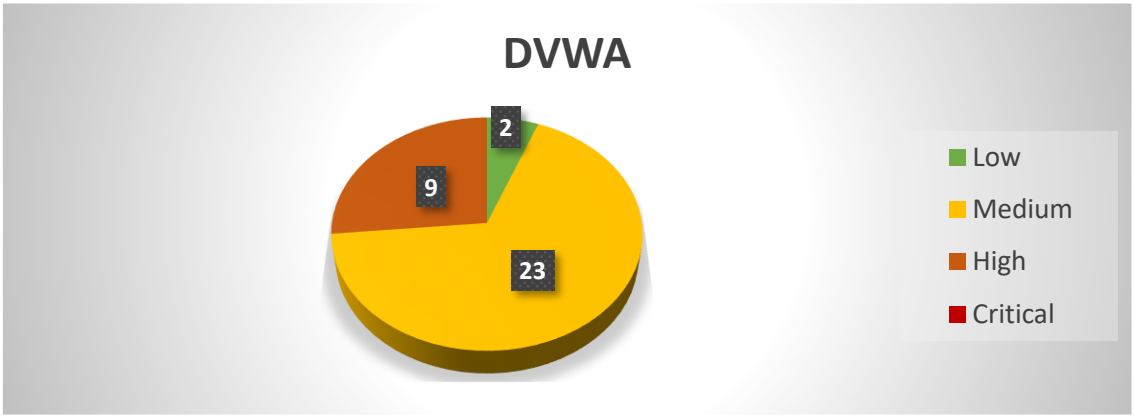

Figura 35 Grafico rischio delle vulnerabilità porta 80 DVWA

Per quanto riguarda Metasploitable 2 il 78% delle vulnerabilità è di livello medio, il 4% di livello basso e il 15% risultano di livello alto. L'unica di livello critico che fa riferimento al codice CVE-2007-5653 (**Figura 36**).

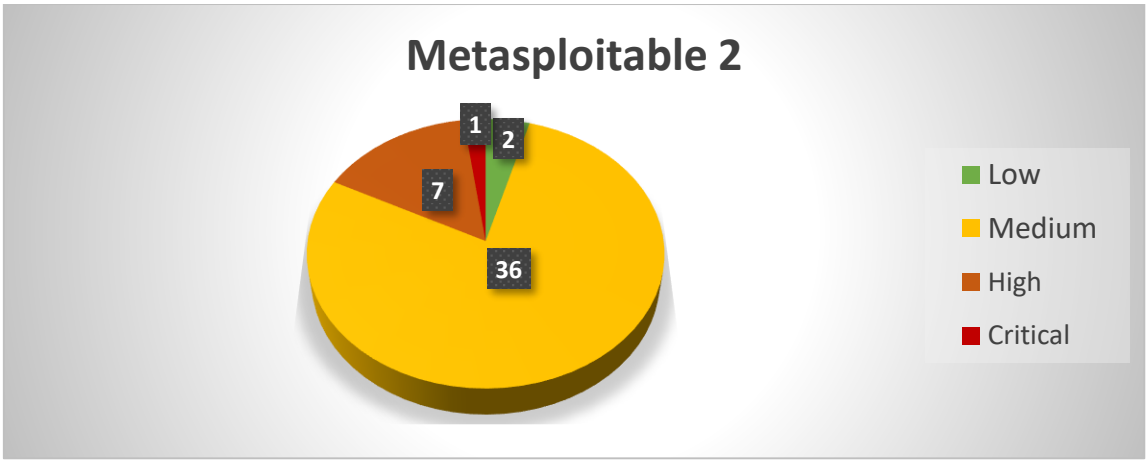

Figura 36 Grafico rischio delle vulnerabilità porta 80 Metasploitable 2

Il discorso cambia per gli altri due sistemi infatti nelle scansioni della porta 80 di Windows Server 2012 R2 è stata scovata solo una vulnerabilità critica, CVE-2015-1635 (**Figura 37)**. Mentre per quanto riguarda Metasploitable 3 sono state trovate solo due vulnerabilità di rischio medio e fanno riferimento alla categoria del Cross Site Scripting (**Figura 38**).

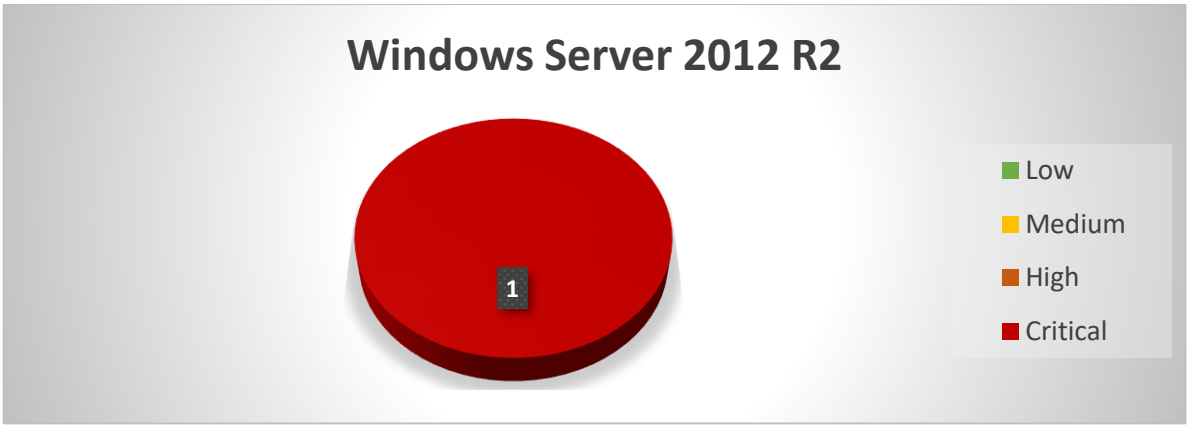

Figura 37 Grafico rischio delle vulnerabilità porta 80 Windows Server 2012 R2

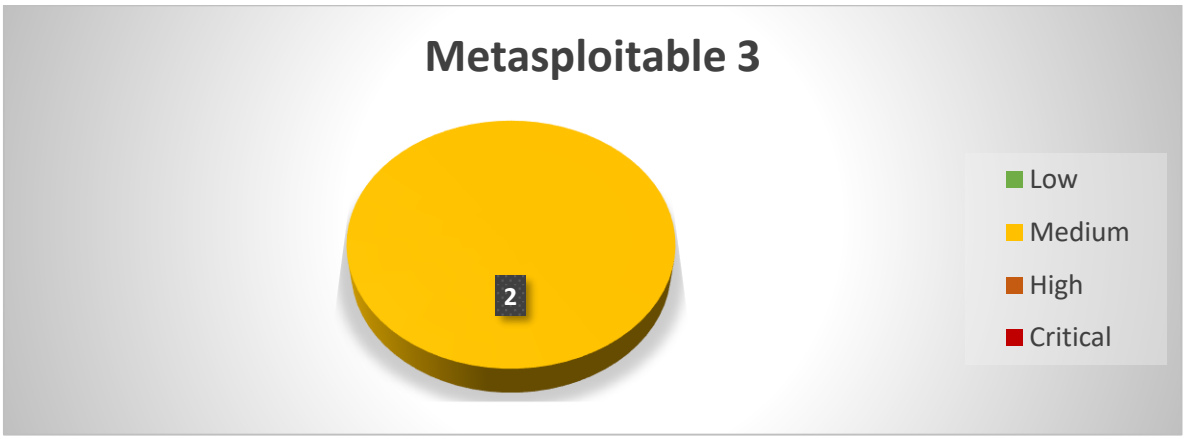

Figura 38 Grafico rischio delle vulnerabilità porta 80 Metasploitable 3

4 CONCLUSIONI

Il tema della sicurezza web acquista sempre più rilevanza in un contesto come quello attuale in cui la tecnologia diventa sempre più pervasiva e fondamentale per l'erogazione di alcuni servizi, soprattutto durante la pandemia da COVID-19.

Ogni giorno vengono rilevate centinaia di vulnerabilità che se sfruttate da utenti malintenzionati possono danneggiare gravemente i soggetti coinvolti. Perciò è importante agire preventivamente, adottando una migliore gestione del rischio e della sicurezza, effettuare controlli frequenti e approfonditi per rilevare le vulnerabilità della propria applicazione web e risolverle prima che queste vengano sfruttate impropriamente.

Esistono moltissimi strumenti per la rilevazione delle vulnerabilità web che differiscono secondo molti aspetti, dal sistema operativo al modo in cui funzionano. In questa tesi tra i vari strumenti raccolti e analizzati è stato scelto un tool, Jok3r per approfondire la valutazione delle performance di analisi confrontando i risultati ottenuti su quattro diversi target (DVWA, Metasploitable 2, Windows Server 2012 R2 e Metasploitable3).

Dall'analisi emerge che come ci si poteva aspettare sono state trovate delle vulnerabilità rilevanti nelle porte del servizio HTTP (porta 80/tcp) e FTP (porta 21/tcp) mentre nelle porte degli altri servizi sono state trovate vulnerabilità "trascurabili". Dato che Windows Server 2012 R2 e Metasploitable 3 sono basati su sistemi Windows e dato che è il sistema operativo più comune tra gli utenti, sono risultati più sicuri in quanto hanno già ricevuto delle eventuali patch che hanno risolto la maggior parte dei problemi.

Infine per migliorare l'attività di scansione di Jok3r, sarebbe opportuno un lavoro costante di manutenzione e aggiornamento sia degli strumenti di cui è provvisto sia del suo database per il riconoscimento delle vulnerabilità così da ridurre casi di falsi negativi.

Sviluppi futuri del lavoro di tesi consentiranno di condurre un'analisi delle performance dello strumento prescelto su un target reale per la rilevazione delle vulnerabilità presenti fornendo consiglio per la loro risoluzione. Infatti, sebbene sia importante rilevare le vulnerabilità, è altresì fondamentale saperle gestire!

5 ELENCO DELLE FIGURE

6 ELENCO DELLE TABELLE

7 SITOGRAFIA

https://www.ptsecurity.com/ww-en/analytics/web-vulnerabilities-2020/

https://cve.mitre.org/

https://owasp.org/www-project-top-ten/

https://www.virtualbox.org/

https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/

https://github.com/Moham3dRiahi/XAttacker/blob/master/README.md

https://github.com/Tuhinshubhra/RED_HAWK/blob/master/README.md

https://github.com/j3ssie/Osmedeus

https://www.openvas.org/

https://github.com/evyatarmeged/Raccoon

https://dvwa.co.uk/

https://sourceforge.net/projects/metasploitable/files/latest/download

https://www.microsoft.com/en-gb/evalcenter/evaluate-windows-server-2012-r2

https://github.com/brimstone/metasploitable3/releases

https://github.com/koutto/jok3r#full-documentation