Università degli Studi di Camerino

Scuola di Scienze e Tecnologie

Laurea in Informatica (L-31)

WebMVA

Collaborative Application for Vulnerability Assessment Analysis

Candidati

Margherita Renieri
Matricola n. 095100

Riccardo Cannella
Matricola n. 096001

Relatore

Prof. Fausto Marcantoni

ANNO ACCADEMICO 2017/2018

Indice

Abstract

La **vulnerabilità** è definita come "*l'esistenza di un punto debole di progettazione o di un errore di implementazione che può portare a un evento indesiderato che compromette la sicurezza del sistema, della rete, dell'applicazione o del protocollo in questione.*"
Nell'ambito della Sicurezza Informatica, una vulnerabilità può essere sfruttata da un utente malintenzionato per accedere a un sistema o a una rete per eseguire azioni che non è autorizzato a compiere. Per far ciò, egli deve disporre di uno strumento (tool) in grado di connetterlo a tale sistema.
Gestire le vulnerabilità di un sistema o di una rete rappresenta la capacità ciclica di riconoscere le criticità e correggerle con delle patches, porzioni software progettate per aggiornare e migliorare il sistema.
La ricerca effettuata ha portato alla realizzazione di uno strumento open source, *WebMVA*, che effettua differenti tipologie di scansione di sistema e/o di rete per determinare le vulnerabilità presenti.
WebMVA è stata organizzata in maniera modulare: l'utente è libero di gestire i propri progetti di scansione, creare moduli *ad hoc* secondo le sue necessità e definire i target che sono l'obiettivo delle sue operazioni.
È stata inclusa una sezione relativa al salvataggio dei resoconti di tali scansioni, utili per la visualizzazione delle falle e per la riduzione del tempo necessario di applicazione delle patches al sistema.

Introduzione

La ricerca svolta nel periodo di Stage++ ha prodotto un'applicazione web, **WebMVA**, caricata su GitHub (`https://github.com/unicam-webmva/webmva`), che permette la gestione di differenti tipologie di scansioni.
Alla base di essa si trova il concetto di *vulnerabilità*: presenza di una falla in un sistema che è il target della scansione.

Il *Capitolo 1* - ***Vulnerabilità e modalità di sicurezza*** - presenta nozioni teoriche relative all'ambito della sicurezza di sistema e di rete.
Si sviluppano i seguenti argomenti:

- definizione di *vulnerabilità* (dei software e dei protocolli) e panoramica generale nella *vulnerability assessment*;
- concetto di *penetration test*, varie metodologie di attacco (riconosciute a livello internazionale) e possibili bersagli;
- analisi delle modalità di sicurezza, un elenco di *liste* e di *database* contenenti le vulnerabilità aggiornate;
- procedure di aggiornamento del sistema, contromisura agli attacchi, nell'ambito della *gestione del rischio*;
- tipologie di *attacchi di rete*, *servizi di sicurezza* e *implementazioni*, nel settore della *sicurezza di rete*;
- servizi di *posta elettronica* garanti della riservatezza (PGP, S/MIME) e loro vantaggi;
- *architetture Web* e diversi *livelli di sicurezza*;
- modalità di *sicurezza nei CMS* e *attacchi a sistemi DNS*;
- *firewall*, differenti tipologie di filtraggio di rete;
- *file di log*, convenienza nell'utilizzo del *log management* nei tools;
- vantaggi nell'uso dei *database* e requisiti per la loro *protezione*.

Il *Capitolo 2* - ***WebMVA*** - contiene un breve *excursus* riguardante il periodo di Stage e la ricerca esposta al Project.
È stata presentata l'applicazione, *WebMVA*, definita la sua tecnologia e indicata la configurazione iniziale.

Sono stati esposti tutti i *Moduli* presenti nell'omonima sezione ed elencate tutte o parte delle funzionalità implementate.

Il *Capitolo 3* - ***Analisi dei Report*** - analizza i risultati dei test effettuati nell'ultimo periodo di ricerca, per verificare le funzionalità di *WebMVA*.
Sono stati creati *Progetti* per ogni tipologia di scansione presente nell'applicazione e inseriti *target comuni* per comparare i risultati ottenuti dai tools inseriti.

Il *Capitolo 4* - ***Evoluzione*** - esamina le modifiche e i miglioramenti apportati al progetto. L'applicazione risulta più efficiente e dà all'utente la possibilità di gestire in maniera autonoma i propri Progetti.

Il *Capitolo 5* - ***Sviluppi futuri*** - comunica ampliamenti futuri della ricerca.

Il *Capitolo 6* - ***Conclusioni*** - sintetizza gli obiettivi raggiunti dalla ricerca nel periodo di Stage++.

Per rendere completo l'elaborato sono presenti *Appendici* contenenti immagini e documenti "citati" nella relazione.

1. Vulnerabilità e modalità di sicurezza

Il Capitolo tratta le basi teoriche di due macro-temi: *vulnerabilità* e *modalità di sicurezza*.
La **prima sezione** fornisce la definizione di *vulnerability assessment*, le possibili *valutazioni* delle vulnerabilità e i *penetration tests*.
Nella **seconda sezione** si analizzano le principali *modalità di sicurezza*.

1.1 Vulnerabilità

La **vulnerabilità** è "*l'esistenza di un punto debole di progettazione o di un errore di implementazione che può portare a un evento indesiderato che compromette la sicurezza del sistema, della rete, dell'applicazione o del protocollo in questione*[1]".
Una vulnerabilità costituisce una falla di sistema che permette l'accesso a un utente malintenzionato. Essa si può presentare ad ogni livello di sistema informatico.

Esistono diversi livelli di vulnerabilità tra cui:

- **vulnerabilità hardware**: rappresenta la perdita di informazioni causate da eventi atmosferici e dalle condizioni precarie dei supporti di memorizzazione;
- **vulnerabilità di software - *bug software***: rappresenta un difetto di progettazione, codifica, installazione e configurazione del software causato dall'incompletezza dei dati forniti in input dal programmatore e dalla mancanza di sistemi di controllo;
- **vulnerabilità dei protocolli**: si manifesta un difetto che avviene quando i protocolli di comunicazione non prendono in considerazione un determinato problema legato alla sicurezza.

Da sole, le vulnerabilità, non compromettono il sistema, ma è il loro sfruttamento che diventa una minaccia e le trasforma in un evento indesiderato. La documentazione **RFC**, *Request for comments*, descrive pubblicamente come questi protocolli debbano essere implementati e permette ai potenziali attaccanti e alle vittime di conoscerne le debolezze.

1.1.1 Tools di vulnerability assessment

L'attività di *vulnerability assessment*, gestione delle vulnerabilità, inizia con l'identificazione dei sistemi e delle risorse a disposizione. Il sistema può essere esposto a diverse minacce come malwares, exploits o scripts che vogliono entrare nel sistema o nella rete per sottrarre/acquisire più informazioni possibili.
Un utente malintenzionato arriva ad impossessarsi di dati personali ed acquisisce privilegi amministrativi, fino ad ottenere tutti i possibili vantaggi. I *tools di vulnerability assessment* sono servizi con l'obiettivo di identificare le vulnerabilità già note, evidenziarne i malfunzionamenti, misurarne il livello di gravità e fornire informazioni riguardo la loro risoluzione. Per aggiornare e riconfigurare il sistema è necessario inserire delle patches.

1.1.2 Valutazione delle vulnerabilità

Vulnerability assessments e *penetration tests* sono strumenti necessari per una buona gestione del sistema e garanzia della sicurezza.
Per *vulnerability assessment* si intende una scansione sistematica e globale delle risorse di rete e dei sistemi di un ambiente informatico che genera un ***report*** dettagliato delle *security vulnerabilities*.
I *penetration tests* comprendono: la scansione, l'identificazione e lo sfruttamento delle vulnerabilità trovate.

Esistono due tipi di valutazione della vulnerabilità:

- **interna**: eseguita all'interno dell'Ente (banca, azienda..) che richiede il servizio. Si effettua un accesso remoto sicuro, simulando un attacco dall'interno verso l'esterno, oppure si ottiene un'autorizzazione e si effettua una vera e propria scansione dall'interno.

  Tale valutazione porta ad identificare:

  - i dispositivi interni,
  - i servizi e i dispositivi interni del footprint;
  - i punti deboli della sicurezza interna per:
    - gestione delle patches
    - controllo dell'accesso alla rete
    - definizione di sicurezza dei dati.

- **esterna**: eseguita effettuando la scansione e l'identificazione delle vulnerabilità e, in alcuni casi, il loro sfruttamento.

  Tale valutazione porta ad identificare:

  - i dispositivi esterni;
  - i servizi;
  - i punti deboli della sicurezza esterna tra cui: firewall, dispositivi di gestione del database, sistemi di sicurezza delle applicazioni Web, sistemi di rilevamento di intrusioni, servizi di prevenzione delle intrusion prevention services (IPS) testing.

1.1.3 Penetration test

Un **penetration test** è un metodo per la valutazione della sicurezza di un sistema informatico o di una rete che simula un attacco di un utente malintenzionato esterno e/o di attaccanti interni autorizzati all'accesso al sistema. Esso verifica se è possibile compromettere la sicurezza del sistema-bersaglio.

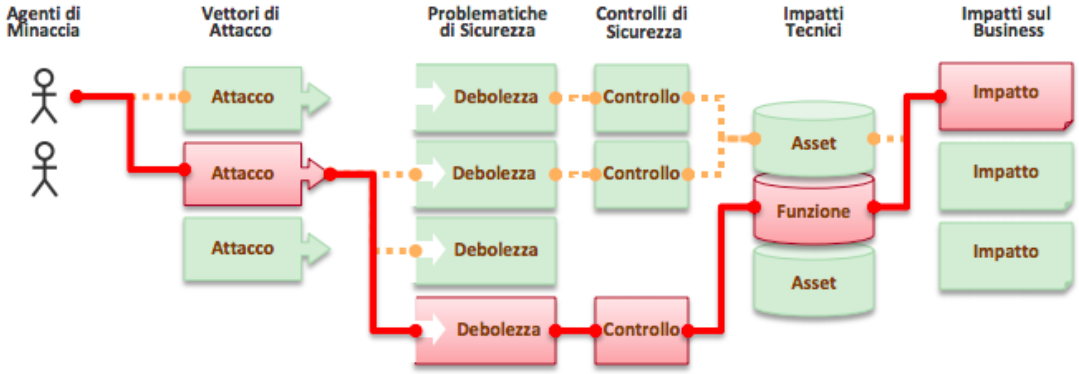


Figura 1.1: Percorso di attacco degli agenti di minaccia[2] .

Queste problematiche affiorano durante una simulazione di una scansione di rete.
Il penetration test è un *metodo* che si adatta all'attività specifica e al bersaglio da verificare. Utilizza diverse metodologie riconosciute a livello internazionale.
Tra le varie metodologie sono presenti:

- **SP-800-115**[3] del NIST[4], *National Institute of Standards and Technology*, agenzia del Governo Americano che si occupa della gestione delle tecnologie;

Figura 1.2: Logo NIST.

- **OSSTMM**[5], *Open Source Security Testing Methodology Manual*, dell'**ISECOM**[6], *Insitute for Security and Open Methodologies*, associazione no-profit internazionale di esperti del settore;

Figura 1.3: Loghi ISECOM e OSSTMM.

- **Testing Guide** di **OWASP**[7], *Open Web Application Security Project*, comunità online che produce articoli, metodologie, documentazione, strumenti e tecnologie disponibili nella sicurezza delle applicazioni web. L'OWASP divulga i rischi della sicurezza informatica legati al web per rendere aziende, enti, professionisti ed utenti più consapevoli. La pubblicazione della **Testing Guide 4.0**[8] risale al 17 settembre 2014.
  Il Project Leader della guida è l'italiano Matteo Meucci, Chief Executive Officer di Minded Security;

Figura 1.4: Logo OWASP.

- **PTES**[9], *Penetration Testing Executing Standards*, un nuovo standard, avviato nel 2009, progettato per fornire servizi di supporto nell'esecuzione di penetration tests e di tutte le sue fasi.

Figura 1.5: Logo PTES.

Il NIST definisce il penetration test come un *processo iterativo con andamento non lineare*. Esso ha un ciclo continuo di ricerca e di attacco: vengono acquisite e utilizzate grandi quantità di informazioni poi reiterate.
L'obiettivo dei penetration test è *valutare* la sicurezza e *verificare* la presenza di problemi nel sistema informatico prima che venga sfruttato da terzi.

Nel documento **SP 880-115**, la *Guida tecnica alla verifica e alla valutazione della sicurezza delle informazioni*, viene affrontato il concetto di penetration test inteso come:

- capacità del sistema di **tollerare** scenari di **attacchi reali**;
- livello di **dettaglio** che un attaccante deve utilizzare per **compromettere** il sistema;
- opportunità di trovare **nuove misure** di sicurezza per il sistema;
- possibilità dei difensori di **individuare** e di **reagire** all'attacco.

Il Testing Guide 4.0 della OWASP è un framework per i test di sicurezza delle applicazioni e dei servizi web basato sulla raccolta delle migliori "best practice" di penetration test. Esso comprende l'inserimento della *Guida degli sviluppatori* e della *Guida di revisione del codice*.

I test case sono aumentati, rispetto alla precedente versione, da 64 ad 87 e sono stati inseriti quattro nuovi capitoli e controlli:

- test identity management
- gestione degli errori
- crittografia
- test lato client.

1.1.3.1 I target

L'OSSTMM definisce una serie di canali e vettori sui quali un'attività può essere svolta:

- **COMMSEC**, *Sicurezza delle Comunicazioni*: test a livello di reti informatiche (Data Network) e telecomunicazioni (Telco);
- **SPECSEC**, *Sicurezza dello Spettro/Segnali*: test a livello wireless e sui segnali (Wireless);
- **PHYSEC**, *Sicurezza fisica*: test di sicurezza fisica (Physical)e test sugli aspetti psicologici degli utenti (Human).

1.1.3.2 Modalità di attacco

Il *penetration test* può essere eseguito all'interno del sistema in caso di simulazione di un attaccante che si trova nell'infrastruttura oggetto della verifica, o anche all'esterno per simulare un attacco al di fuori del perimetro del sistema.
Come definito dall'OSSTMM, il test è un vettore che contiene una quantità di informazioni condivise tra attaccanti e bersagli secondo la tipologia di test realizzata.
Le tipologie vengono classificate su una scala di grigi: *white*, *gray* e *black box*.

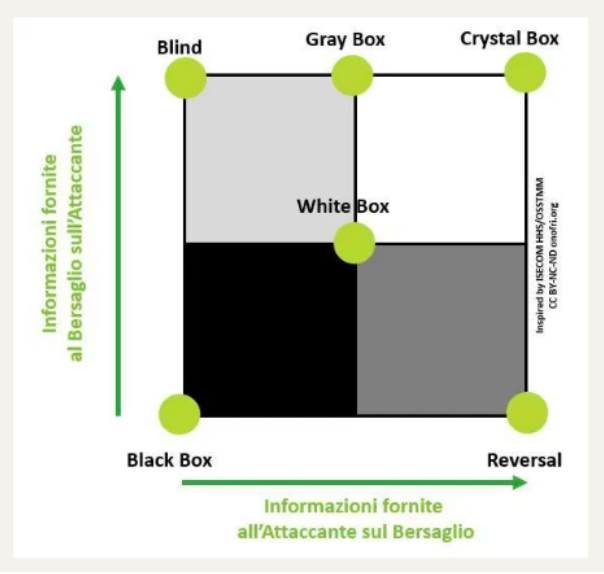


Figura 1.6: Descrizione di penetration test nell'OSSTMM.

1.2 Modalità di sicurezza

1.2.1 Liste e database di vulnerabilità

Ogni giorno vengono scoperte nuove vulnerabilità in ogni ambito informatico (protocolli, algoritmi o software).
Le vulnerabilità che non sono ancora di dominio pubblico sono dette **0-day** perché gli sviluppatori dell'applicazione vulnerabile la conoscono da zero giorni.
Esiste un vero e proprio mercato delle vulnerabilità e più tempo passa dalla scoperta, meno valgono in termini di denaro e di affidabilità.
Quando gli sviluppatori o degli *hacker* vengono a conoscenza di nuove vulnerabilità, in genere, pubblicano le patches da inserire nel sistema violato oppure dei metodi per aggirare temporaneamente il problema. La diffusione avviene nelle *liste di vulnerabilità note*, sfruttate dai software di vulnerability assessment per capire il livello di rischio di un sistema.
Esistono molte liste, di cui le più importanti sono:

- **CVE**: *Common Vulnerabilities and Exposures*, lista creata e mantenuta dalla **MITRE Corporation** a partire dal 1999. Contiene un ID univoco per ogni vulnerabilità, la data di creazione (che non sempre corrisponde alla data di *pubblicazione*), la descrizione della vulnerabilità e alcuni riferimenti esterni. Per inserire le vulnerabilità nella lista è necessario seguire una procedura:
  - gli enti autorizzati propongono una nuova vulnerabilità alla MITRE;
  - gli addetti della MITRE valutano la richiesta;
  - introduzione di un nuovo ID nella lista.
- **CVSS**: *Common Vulnerability Scoring System*, è un *open industry standard* che assegna un punteggio a ogni vulnerabilità in base alla gravità della stessa e alla categoria di appartenenza;
- **NVD**: *National Vulnerability Database*, database mantenuto dal Governo Statunitense, è strettamente legato al CVE, in quanto contiene ogni singolo ID presente nella lista CVE pubblica.
  L'ID viene associato a dati provenienti da altre liste, come CVSS, tramite **SCAP**, *Security Content Automatation Protocol*;
- **VulnDB**: servizio commerciale a cura della compagnia *Risk Based Security*, basato sul precedente **OSVBD**, *Open Source Vulnerability Database*, che ha smesso di funzionare nel 2016. Il suo contenuto è molto simile a quello del NVD.

1.2.1.1 Mailing-list

Molti distributori e progetti open source riportano advices sulla sicurezza e notifica di patches nelle **mailing list** di *Bugtraq*[10] di *Full-Disclosure*[11], di *National Cyber Awareness System*[12]
Bugtraq contiene una lista di vulnerabilità comunicate dai ricercatori di sicurezza ai distributori. Le vulnerabilità riportate sono associate alle relative patches o ai workaround disponibili al momento dell'annuncio.
Full-Disclosure elenca le vulnerabilità inserite dai ricercatori indipendenti che non hanno ottenuto la collaborazione necessaria per aggiustare i difetti riscontrati.
US CERT, *United States Computer Emergency Response Team*, offre le mailing list con le vulnerabilità più diffuse, tra cui il *National Cyber Awareness System* e gli aggiornamenti di *Current Activity*.
Il *National Cyber Awareness System*[13] garantisce l'accesso a informazioni riguardanti argomenti e minacce di sicurezza. Il sistema permette di consultare:

- **avvisi**: informazioni tempestive sugli attuali problemi di sicurezza, vulnerabilità ed exploits;
- **report di analisi**: approfondimenti su minacce informatica nuove o in evoluzione;
- **attività correnti**: aggiornamenti sulle tipologie di attività di sicurezza ad alto impatto per la comunità;
- **bollettini**: liste settimanali di nuove vulnerabilità e delle relative patches disponibili;
- **advices**: consigli su problemi di sicurezza più comuni.

Nella sezione relativa ai *Current Activity* sono indicate le più frequenti tipologie di incidenti di sicurezza ad alto impatto, segnalate al US-CERT.
US-CERT utilizza dei *feed RSS* per permettere agli utenti registrati di essere aggiornati sui nuovi articoli.
Oltre alle tre mailing-list sopra descritte, molti progetti open source presentano delle proprie mailing-list dove inserire problematiche relative all'ambito della sicurezza.

1.2.2 Aggiornamenti di sistema

Un aspetto fondamentale della *gestione del rischio* informatico, soprattutto in azienda, è la gestione delle patches di sicurezza, degli aggiornamenti dei sistemi e dei software utilizzati.
È frequente avere computer con sistemi operativi non più supportati (esempio: Windows XP) usati in maniera sprovveduta nell'ambito business. Il loro utilizzo può provocare problemi di sicurezza, perfino arrestare un'intera attività.
Una buona gestione del rischio permette un aggiornamento repentino del sistema, con procedure software automatizzate e/o lavoro fisico di consulenti informatici per rendere l'ambiente di lavoro più sicuro ed efficiente.

1.2.2.1 WSUS

Per i sistemi *Microsoft Windows* esiste un tool gratuito fornito dalla Microsoft, chiamato **WSUS**, *Windows Server Update Services*, per gestire gli aggiornamenti di sistema creando policy di mantenimento in base ai prodotti e ai target. È possibile centralizzare la distribuzione degli aggiornamenti all'interno della struttura aziendale, in modo da gestire al meglio la sicurezza delle workstation e dei server. Si può eventualmente fare in modo che tutti i PC Windows in rete disinstallino un aggiornamento se si dovessero verificare anomalie in seguito alla sua installazione.

1.2.2.2 Red Hat Satellite

In ambito *Linux enterprise*, è possibile gestire il software e gli aggiornamenti per sistemi Red Hat tramite **Red Hat Satellite**. Esso amministra vari aspetti di un'infrastruttura aziendale, tra cui gli aggiornamenti di sicurezza, le policy di gruppo, i software installati, la clusterizzazione e configurazione.

1.2.3 Logging

Un *log* è il risultato di una registrazione sequenziale e cronologica delle operazioni realizzate in un sistema informatico. Queste attività possono essere effettuate dall'utente oppure avvenire in modo automatizzato mediante procedure.
I file di log sono delle raccolte di eventi relativi ad un sistema operativo o un'applicazione dove sono registrati e memorizzati gli eventi. Il contenuto dei log non è standardizzato e, secondo la tipologia di informazioni, cambia il sistema di logging.
La sincronizzazione oraria è fondamentale, perché ogni aspetto della gestione, sicurezza, pianificazione, e debug di una rete ha necessità di determinare gli eventi accaduti perciò utilizza il protocollo NTP.
NTP, Network Time Protocol, è un protocollo client-server, appartenente al livello applicativo, in ascolto sulla porta UDP 123, utilizzato per sincronizzare gli orologi dei computer all'interno di una rete a commutazione di pacchetto, con tempi di latenza variabili ed inaffidabili.
SNTP, *Simple Network Time Protocol*, è una versione semplificata del protocollo NTP che non richiede la memorizzazione dei dati tra due successive comunicazioni (impiegato nei sistemi embedded dove non è richiesta un'estrema precisione). SNTP utilizza la stessa struttura dei pacchetti TCP/IP di NTP, ha una precisione di sincronizzazione molto ridotta, poiché usa algoritmi più semplici.
Il pacchetto NTP contiene un programma in background (daemon o servizio) che sincronizza il tempo di sistema del computer da uno o più orologi di riferimento esterni che possono essere sia altri dispositivi sulla rete o un orologio radio, collegato al computer.
È consigliabile sincronizzare i file di log con un sistema remoto.
Senza sincronizzazione oraria non è possibile correlare correttamente i files di log.

Esistono differenti log:

- **di sistema**: traccia gli eventi di funzionamento del sistema operativo del server, degli apparati di rete, dei dispositivi di sicurezza;
- **applicativo**: definisce gli eventi del software utili per visualizzare bug e fare numerose analisi;
- **di sicurezza**: individua le attività svolte dall'amministratore di sistema o dagli utenti finali che hanno un impatto sull'integrità, la disponibilità e la riservatezza dei dati.

Il log deve contenere le informazioni minime per *tracciare* un evento, *registrare* tutte le informazioni sul funzionamento del sistema e gli errori, problemi riscontrati.
Le registrazioni contengono messaggi al sistema come al kernel, ai servizi e alle applicazioni in funzione. I record indicano:

- momento di *registrazione*
- nome del sistema contenente il *programma che ha generato il log*
- *nome* del programma.

Un file di log è sequenziale e sempre aperto alla scrittura.
Una volta chiuso, viene memorizzato e conservato con periodicità.
I files di log sono disponibili per supportare attività di monitoraggio del sistema e di amministrazione. La conservazione di essi è importante per la sicurezza della rete, poiché contengono informazioni preziose per il supporto delle attività di monitoraggio, di diagnostica, di amministrazione funzionale e di compliance.

La tecnologia mobile e il cloud hanno aumentato in maniera esponenziale la quantità di dati in circolazione, la necessità di garantire sicurezza, protezione dei dati e continuità del servizio.
Il *Log Management*, la gestione dei log, è l'insieme delle attività legate alla raccolta e all'analisi dei dati tracciati dai messaggi di log. Si tratta di un'insieme di informazioni da vari sistemi-chiave con aggregazione centralizzata in un log server.

L'analisi di una grande quantità di dati di log comporta:

- sincronizzazione dei sistemi presi in considerazione, per poter controllare con **ordine temporale** esatto le informazioni dei record;
- **ampio volume** di dati presi in analisi;
- processo di **normalizzazione** per produrre un output comune dell'analisi da diverse fonti;
- **alta frequenza** di generazione dei log;
- **veridicità** e **accuratezza** degli eventi registrati;
- traccia i **movimenti** degli attaccanti e degli utenti e deve rispettare il *Regolamento generale sulla protezione dei dati (GDPR)*[14].

Figura 1.7: Vantaggi tools di log management.

L'attività di log genera informazioni utili per la *manutenzione di applicativi web*, per verificare se il problema riscontrato è una vulnerabilità di sistema, migliorare l'analisi, la risoluzione dei problemi e aiutare a testare nuove caratteristiche in fase di sviluppo. Le previsioni[15] affermano che il mercato dei Log Management aumenterà con un **tasso di crescita annuo del 12.1%**. Tale sviluppo si spiega perché i sistemi di Log Management rivestono un ruolo determinante nell'era digitale.
Un buon sistema di gestione dei log permette alle aziende di soddisfare le disposizioni normative, con la possibilità di usarlo per il monitoraggio e il controllo delle attività, sfruttandone i benefici della Business Intelligence e garantendo elevati standard di sicurezza.

1.2.3.1 Normativa

Secondo il decreto del *Garante per la protezione dei dati personali* pubblicato sulla Gazzetta Ufficiale n. 300 del 24 Dicembre 2008, ogni azienda, dopo l'individuazione dei sistemi di rete che contengono i dati più critici deve nominarne gli amministratori e dotarsi di un sistema di Log Management che riesca a tracciare gli accessi degli operatori ai dispositivi ed alle applicazioni che gestiscono.[16]

Il **Regolamento generale per la protezione dei dati personali** *n.2016/679*, pubblicato nella *Gazzetta Ufficiale Europea* il 4 maggio 2016, entrato in vigore il 24 maggio 2016, attuato dal *25 maggio 2018* si focalizza sulla *accountability*, *responsabilizzazione* del titolare e degli addetti del trattamento dei dati.

Evidenzia la necessità di attuare misure di tutela e garanzia dei dati trattati e di demandare ai titolari il compito di decidere autonomamente le modalità e i limiti del trattamento come indicato nel Regolamento:

- **principio "privacy by design"**, configurato prevedendo le garanzie per tutelare i diritti degli interessati;
- **rischio del trattamento**, valutazione dell'impatto negativo sulle libertà e i diritti degli interessati.

L'approccio del GDPR n.2016/679, a differenza della precedente normativa, n.2008/300, è centrato sulla *protezione dei dati*. Si tratta di un approccio *risk based*, che determina la misura di responsabilità del titolare o dell'addetto responsabile, tenendo conto della *natura, portata, contesto, finalità del trattamento, probabilità, gravità dei rischi* per i diritti e *libertà* degli utenti. Il criterio è *flessibile* e *adattabile* alle esigenze e agli strumenti tecnologici, ma *delega* all'azienda la valutazione del rischio, rendendo più difficili le contestazioni in caso di violazioni. Esso tiene in maggiore considerazione le esigenze delle aziende, rendendo meno burocratica la gestione dei dati. Le nuove norme prevedono, inoltre:

- per i cittadini, semplice accesso alle informazioni riguardanti i propri dati, le finalità e modalità di trattamento degli stessi;
- diritto alla portabilità, che consente il trasferimento dei dati personali tra i vari servizi online;
- istituzionalizzazione del *diritto all'oblio,* ***diritto alla cancellazione***, previsto dalla Corte di Giustizia Europea, che consente di chiedere ed ottenere la rimozione dei dati quando viene meno il loro interesse pubblico;
- obbligo di notifica delle aziende delle gravi violazione dei dati dei cittadini;
- obbligo, per le aziende, di rispondere alla sola autorità di vigilanza dello Stato dove hanno sede: *principio del "one stop shop"* o *sportello unico*;
- sanzioni amministrative fino al 4% del fatturato globale delle aziende in caso di violazioni delle norme.

1.2.4 Firewall

Nella progettazione di una rete è utile fissare delle regole per indicare come e dove si può accedere per usufruire di determinati servizi. Il *firewall* è la combinazione di hardware e software che separa una rete privata dal resto di Internet.
Esso permette all'amministratore di rete di controllare e gestire il flusso di traffico tra rete interna e risorse Internet, ispezionando ogni singolo pacchetto in entrata e in uscita. Vengono inoltrati i pacchetti che rispecchiano criteri definiti dalle regole formulate dall'amministratore, quelli che falliscono il test vengono respinti.

Un firewall ha tre funzionalità:

- **permette** il passaggio dall'esterno verso l'interno e viceversa;
- **consente** l'accesso al solo traffico autorizzato, secondo la politica di sicurezza locale. Il criterio di accesso è fornito da una tabella che elenca *sorgenti*, *destinazioni bloccate* e *regole di base* dove è definito cosa fare con i pacchetti in entrata o in uscita di altre macchine;
- **garantisce** l'immunità di penetrazione e la corretta progettazione, installazione e configurazione.

In un primo momento, i firewall applicavano regole indipendenti ad ogni pacchetto, ma si è costatato che scrivere regole che permettessero funzionalità utili e bloccassero tutto il traffico non desiderato risultava molto complicato.
Di conseguenza, sono state adottate tre categorie di firewall:

- **packet filter, filtri di pacchetti**: una rete privata è collegata al suo *ISP*, *Internet Service Provider*, mediante un router dove transita tutto il traffico e avviene il filtraggio dei pacchetti.
  Per prima cosa, vengono analizzate le intestazioni dei datagrammi e l'applicazione delle regole di filtraggio per stabilire quali pacchetti passano e quali no. Le decisioni sono basate su:
  - indirizzo IP sorgente o destinazione
  - tipo di protocollo nel campo del datagramma IP
  - porte sorgente e destinazione TCP o UDP
  - bit di flag TCP
  - regole diverse per i datagrammi che entrano ed escono dalla rete
  - regole diverse per le interfacce del router.

  La configurazione del firewall dipende dalle politiche dell'organizzazione dell'amministratore di rete. Le regole di controllo sono implementate nell'interfaccia dei router tramite liste di verifica degli accessi. *[CERT Filtering 2012]* fornisce una lista di regole di filtraggio di porta/pacchetto consigliati, per evitare lacune di sicurezza nelle applicazioni di reti esistenti.

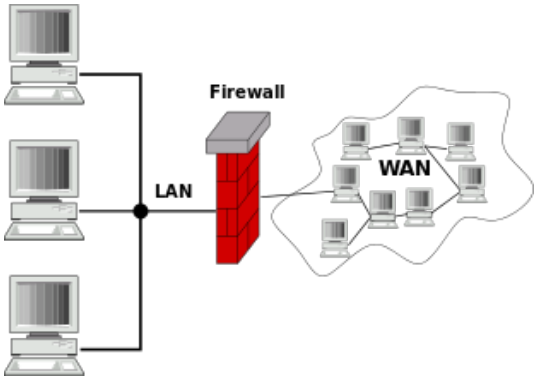


Figura 1.8: Generico firewall.

- **stateful filter, filtri con memoria di stato**: i firewall stateful associano i pacchetti alle connessioni e usano i campi dell'intestazione TCP/IP per tener traccia delle connessioni. Il Web server esterno, per spedire un pacchetto a un host interno, deve aspettare che quest'ultimo abbia stabilito una connessione col esso[17].
  Il firewall stateful visualizza l'inizio della nuova connessione e osserva dall'handshake a tre vie. Dopo aver visualizzato un pacchetto FIN, flag che determina la chiusura di una connessione TCP, per la connessione stessa, la considera terminata.

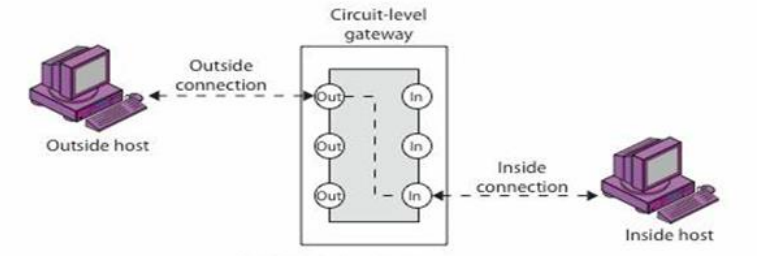


Figura 1.9: Application level gateway.

- **application level gateway, gateway di livello applicativo**: è uno specifico server dove passano tutti i dati delle applicazioni. Esso garantisce un elevato livello di sicurezza. Ogni richiesta viene interpretata, in modo da essere compresa dall'applicazione server e depurata, per prevenire il passaggio di contenuti maliziosi o malformati. Un gateway a livello applicativo visualizza le intestazioni IP/TCP/UDP e prende decisioni sulle politiche in base ai dati applicativi. Un host può avere differenti gateway ma ciascuno è considerato un server differente con processi a sé stanti.
  Il firewall limita il flusso di informazioni e ne impedisce l'uscita. Esso previene la propagazione di worm e frena la diffusione di informazioni confidenziali.
  I log dei firewall possono essere strumenti per comprendere da dove provengono le minacce della propria rete.

1.2.5 Sicurezza di rete

La sicurezza dei dati in rete è un tema molto complesso, trattato dallo **IAB**, *Internet Architecture Board*, con il documento RFC 1636.
La relazione, pubblicata nel 1994, esprime preoccupazioni per la sicurezza in Internet e identifica alcune delle aree-chiave su cui concentrarsi maggiormente.
La quantità di record presenti nelle liste *vulnerabilità note* fa capire quanto sia importante difendersi dagli attacchi. La figura *1.10* mostra le nuove vulnerabilità scoperte e catalogate anno per anno.

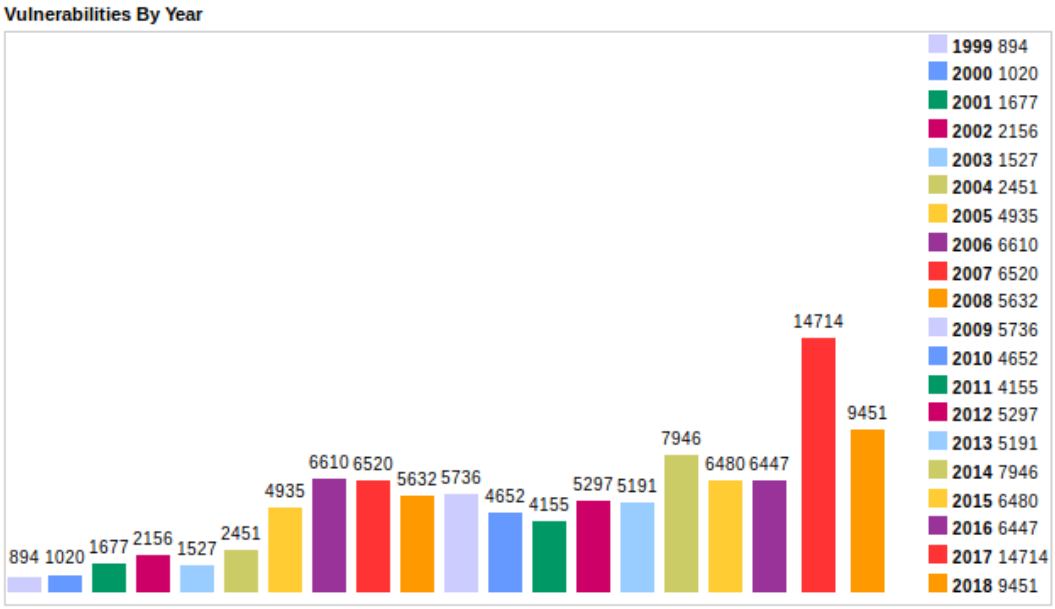


Figura 1.10: Nuove vulnerabilità per anno nella lista CVE.

Gli attacchi eseguiti attraverso una rete sono diventati sempre più sofisticati: le conoscenze dell'utente medio non sono più sufficienti per la sua difesa. Nella maggior parte dei casi, egli non è a conoscenza che il suo sistema è a rischio o che è già controllato.
L'**ITU**, *International Telecommunication Union*, è un'organizzazione internazionale che pubblica *raccomandazioni* per le telecomunicazioni. Una di queste è **X.800**, chiamata "Architettura per la sicurezza di OSI", *Open Systems Interconnection*, che identifica attacchi, meccanismi e servizi di sicurezza.

1.2.5.1 Tipi di attacchi in rete

X.800 descrive due grandi categorie di attacchi in rete:

- **attacchi passivi**: fanno uso di tecniche di monitoraggio del traffico per ottenere informazioni. Esistono due tipi di attacchi passivi:
  - *intercettazione non autorizzata* dei messaggi attraverso un canale di comunicazione, che permette all'attaccante di capire la frequenza dei messaggi, chi li invia, a chi sono indirizzati, in quale momento;
  - *rilascio delle informazioni intercettate* che possono compromettere l'entità di provenienza;

- **attacchi attivi**: il malintenzionato modifica in qualche modo il flusso dei dati o ne crea di falsi. Possono essere di quattro tipi:
  - *masquerade*: qualcuno finge di essere un'altra entità per ottenere privilegi o per creare false informazioni;
  - *replay*: dati intercettati e poi ritrasmessi per ottenere vantaggi ed informazioni (esempio: una falsa autenticazione);
  - *modifica dei messaggi*: cambiamento del flusso legittimo di dati per scopi malevoli;
  - *negazione del servizio*, *DoS*, *Denial of Service*: uso di diverse tecniche per rendere inutilizzabile e/o irraggiungibile un servizio.

1.2.5.2 Servizi di sicurezza

Un *servizio di sicurezza*, definito in X.800, è un mezzo per fornire determinate protezioni in ambito di rete. I servizi di sicurezza sono cinque:

- **autenticazione**: il ricevente ha la certezza che i dati provengano dall'entità dichiarata come mittente;
- **controllo degli accessi**: capacità del sistema di limitare gli accessi a risorse e ad applicazioni solo agli enti autorizzati;
- **riservatezza dei dati**: prevenzione degli attacchi passivi per rendere i dati, se intercettati, non interpretabili;
- **integrità dei dati**: il destinatario riceve gli stessi dati inviati dal mittente senza modifiche, ritardi o duplicati inattesi;
- **non-ripudio**: il peer che ha inviato/ricevuto un determinato dato non può negare di averlo fatto;
- **disponibilità del servizio**: dati e applicazioni raggiungibili in qualunque momento.

I servizi possono essere impiegati in un sistema di protezione sia singolarmente che in varie combinazioni, in base al livello di protezione richiesto.

1.2.5.3 Meccanismi di sicurezza

I meccanismi di sicurezza sono le implementazioni reali dei servizi di sicurezza descritti nella precedente sezione

Alcuni di questi meccanismi sono:

- **cifratura**: algoritmi di pubblico dominio che trasformano i dati in chiaro per non risalire ad essi in maniera semplice fornendo il servizio di riservatezza. Prevedono uno scambio sicuro di chiavi tra le parti attraverso un canale insicuro, ad esempio tramite l'algoritmo di scambio di **Diffie-Hellman**. Gli algoritmi di cifratura più comuni sono **3DES** e **AES**;
- **firma digitale**: fornisce a una delle due parti la certezza dell'identità del suo interlocutore e dell'integrità dei dati attraverso l'utilizzo di *chiavi pubbliche* e *chiavi private*. Non fornisce riservatezza: potenzialmente chiunque può avere la chiave pubblica per decifrare il messaggio. Un algoritmo per fornire firme digitali è **RSA**;
- **applicazioni di autenticazione**: meccanismi che permettono l'autenticazione di un ente per fornire un controllo degli accessi. L'applicazione di autenticazione più diffusa è **Kerberos**.

A dimostrazione della ciclicità del processo di analisi e verifica delle vulnerabilità, la maggior parte degli algoritmi usati per creare firme digitali sono stati dichiarati deprecati se usano **SHA-1** per l'hashing: un team di ricerca di Google, all'inizio del 2017, ha dimostrato di poter creare due file diversi con lo stesso hash SHA-1[18]. Il gruppo ha quindi raccomandato l'uso di algoritmi di hashing più sicuri, come **SHA-256**.

1.2.6 Sicurezza web

Il Web è in sostanza un'applicazione client/server, presente in Internet, che permette a chiunque di presentarsi al mondo in pochi e semplici passi. Uno dei modi di usare il Web è il commercio online, che ultimamente sta coinvolgendo anche aziende medio-piccole. Altri usi comprendono l'uso dei social network, la consultazione di riviste e periodici, fornire applicazioni complesse, come **WebMVA**, e molti altri.
Tutti questi *siti web* possono risultare vulnerabili a una miriade di attacchi se non vengono protetti e aggiornati costantemente.

1.2.6.1 HTTP

L'architettura Web è basata sul protocollo **HTTP**, *HyperText Transfer Protocol*, proposto nella sua versione 1.0 nell'RFC 1945. **HTTP** è un protocollo di scambio di testi dove il client invia una richiesta di dati al server, costantemente in ascolto.
HTTP non fornisce sufficiente sicurezza perché tutto lo scambio viaggia in chiaro.
Google, nella versione 68 di Chrome rilasciata a luglio 2018, segnala, nella barra degli indirizzi, i siti che fanno uso di HTTP come "non sicuri" .
L'alternativa sicura è **HTTPS**, che usa il protocollo SSL/TLS.

HTTPS e SSL/TLS
HTTPS fornisce connessioni private e autenticate proteggendo la privacy degli utilizzatori. In un primo momento era usato per i siti con servizi di pagamento online, oggi è più diffuso.
SSL (*Secure Socket Layer*) e la sua evoluzione **TLS** (*Transport Layer Security*) sono due suite di protocolli di cifratura usati nelle comunicazioni tramite HTTPS. Sfruttano vari algoritmi per ottenere un grado più che ottimale di sicurezza.
HTTPS cifra tutta la comunicazione HTTP tramite SSL/TLS, quindi protegge l'URL richiesta, i parametri, i cookie e gli header della richiesta.

L'architettura di SSL/TLS è basata su:

- **connessioni**: relazione *peer-to-peer* transitoria con scambio di dati privati;
- **sessioni**: legame associativo tra client e server, effettuato tramite un protocollo di *Handshake*.

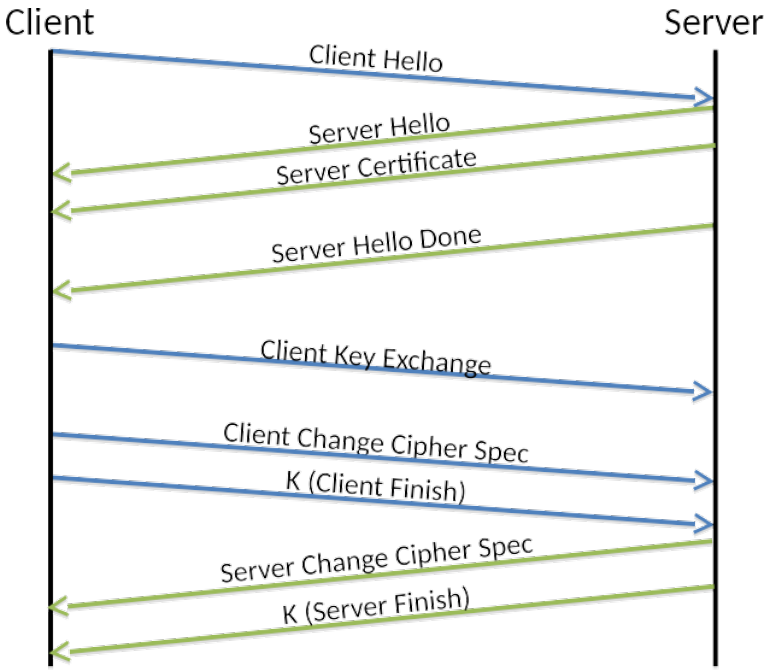


Figura 1.11: Three-way Handshake TLS.

Nella fase di *Handshake* vengono concordati gli algoritmi da usare per gli scambi e per la cifratura delle chiavi di sessione. Una volta scambiate le chiavi, client e server hanno una *sessione* aperta che può supportare molteplici *connessioni*.
In HTTPS solo il server si autentica tramite un certificato fornito da una CA pubblica. Le identità delle CA fidate a livello mondiale sono integrate nei browser e nei sistemi operativi, ma è possibile crearne di proprie per utilizzarle in ambiti più ristretti, per esempio una intranet.

1.2.7 Sicurezza dei CMS

Un **CMS**, *Content Management System*, è un software che permette la gestione dei contenuti di un sito web, in maniera semplice e veloce, anche a chi non conosce i linguaggi di programmazione web.
Esistono vari tipi di CMS, programmati in diversi linguaggi e con molteplici obiettivi: quelli legati alla gestione classica dei contenuti di un sito, quelli più specializzati in blogging, fino a CMS dedicati all'e-commerce.
Un CMS, in quanto software, è soggetto a bug e alle vulnerabilità sfruttabili dai malintenzionati. I rischi sono perlopiù legati al database usato e al linguaggio di programmazione web lato server.

1.2.8 Sicurezza dei DNS

Il **DNS**, *Domain Name Service*, è un protocollo, concepito nel 1982, che permette di tradurre i nomi di host forniti dall'utente in indirizzi IP.
Esso non prevede alcuna procedura di autenticazione all'invio delle richieste del servizio e nessun requisito di confidenzialità delle informazioni scambiate nelle domande e nelle risposte del server. L'*assenza* di questa verifica, permette agli attaccanti di:

- **interrompere** il servizio;
- **indirizzare** a un determinato server DNS pubblico numerose domande;
- **sovraccaricare** il sistema, non rendendolo disponibile per la risoluzione delle query degli utenti legittimi.

L'attacco è propagato, chiedendo a un server DNS risoluzioni relative a $FQDN$[19] appartenenti a domini diversi. Il server DNS locale invia le richieste ai server gerarchicamente superiori e contribuisce alla propagazione dell'attacco, distributed DoS.
La *contromisura* per questo attacco è configurare ogni client con una **lista di server DNS alternativi**. Così è possibile effettuare un attacco solamente se tutti i server DNS della lista sono bloccati.
Per evitare problemi di interpretazione, l'identificativo presente nell'intestazione delle *query DNS*, è ripetuto nelle *DNS reply*.
L'utente malintenzionato, senza questa ripetizione, può inserire informazioni false in un client o in un server DNS, inviando DNS reply relative a query mai effettuate o contenenti informazioni fraudolente. Le informazioni presenti nel database del server DNS sono alterate e compromettono la raggiungibilità di una singola destinazione finale.
Effettuare attacchi *man-in-the-middle*, determina l'alterazione delle *cache DNS* e la modifica del contenuto nel campo *response* delle reply legittime.
Il *DNS spoofing* altera il database di un server DNS e produce il risultato di un attacco DoS o può essere il presupposto per un attacco di re-direzione del traffico applicativo (dipende dalle informazioni relative a indirizzi IP inesistenti o associati a stazioni finali controllate dall'attaccante).
L'inserimento dell'agente di minaccia nella *zone transfer*, può modificare tutte le informazioni relative a una zona e diffonderle nella gerarchia di server DNS, compromettendo le richieste di risoluzione del dominio della zona.
In mancanza di sistemi di autenticazione più evoluti, un server fidato può essere riconosciuto sulla base dell'indirizzo IP, incluso in una lista di *sistemi trusted*.
Le informazioni restituite da una DNS reply devono essere verificate. Procedure di autenticazione basate sull'uso dei nomi di dominio sono pericolose, perché i meccanismi di riconoscimento basati sull'indirizzo IP possono essere aggirate tramite *IP Spoofing*.
Si possono implementare meccanismi di verifica dell'affidabilità dell'informazione ricevuta con quanto contenuto nel database distribuito del DNS.

Lo schema è conosciuto come procedura di *double inverse lookup*:

- *prendere* l'indirizzo *IP*, ottenuto dalla reply di una DNS o estratto da un pacchetto IP ricevuto dalla stazione client di cui si cerca di stabilire il dominio di appartenenza, ed effettuare una inverse lookup;
- *tradurre* l'hostname ricavato in indirizzo IP;
- *confrontare* i due indirizzi IP.

Se gli indirizzi IP coincidono, il sistema DNS non ha subito manomissioni. Altrimenti il database è stato manomesso.
Col meccanismo di double inverse lookup, è possibile verificare l'alterazione di un server DNS, poiché le operazioni di traduzione diretta e inversa coinvolgono componenti del database DNS mantenuti su server DNS fisicamente diversi. Entrambi i database sui server remoti devono essere manomessi per poter superare il doppio controllo e effettuare un attacco.
Quando il server DNS di default, accetta qualsiasi informazione contenuta nel campo *additional* del messaggio di reply ricevuto, l'utente malintenzionato inietta un record contraffatto nella sua cache locale. In caso di verifica di attacco, il record contiene l'associazione richiesta nella seconda DNS query dal client, il server DNS la restituisce senza contattare server DNS superiori.

1.2.9 Sicurezza della posta elettronica

La posta elettronica è l'applicazione di rete più usata in assoluto, il cui scopo è quello di fornire un meccanismo asincrono di scambio di messaggi tra utenti, indipendentemente dal computer o dal sistema operativo utilizzati. Data la diffusione di quest'applicazione per gli usi più disparati, è ovvio cercare di renderla più sicura.
I due metodi che possono fornire garanzie di riservatezza sono **PGP**, *Pretty Good Privacy* e **S/MIME**, *Secure/Multipurpose Internet Mail Extensions*.

1.2.9.1 PGP

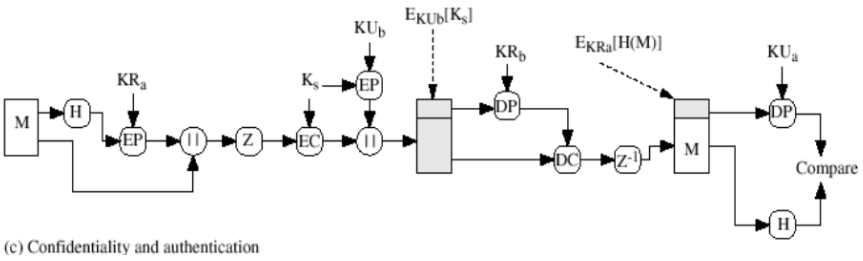


Figura 1.12: Funzionalità crittografiche di PGP.

PGP, sviluppato da Phil Zimmermann nel 1991, è uno standard dello IETF descritto tramite gli RFC 2440, 3156 e 4880. Fornisce servizi di autenticazione, integrità, riservatezza, compressione e segmentazione dei messaggi. Comprende molti algoritmi di cifratura, sia simmetrici (**CAST-128**, **IDEA** e **3DES**) che asimmetrici (**DSS** e **RSA**) e di autenticazione (**SHA-1**). Per lo scambio delle chiavi di sessione usa l'algoritmo **Diffie-Hellman**.
Una delle applicazioni che usano PGP è *GNU Privacy Guard*.

Le proprietà del PGP sono:

- **confidenzialità**: si genera un numero casuale di 128 bit, usato come chiave di sessione *one-time* per cifrare il messaggio in modo che non viaggi in chiaro. Viene poi cifrato tramite la chiave pubblica del ricevente che lo dovrà successivamente decifrare con la sua chiave privata. Al destinatario arrivano quindi la chiave di sessione e il messaggio cifrati.

- **autenticazione**: viene generato un hash del messaggio con SHA-1, lo si cifra con la chiave privata RSA del mittente e si inserisce il risultato di queste operazioni alla fine del messaggio. Il destinatario è certo che il messaggio sia stato inviato effettivamente da chi dice di averlo spedito.

- **compressione**: l'algoritmo usato per comprimere i messaggi è *ZIP*.
  Le firme vengono generate prima di comprimere il messaggio, perché è possibile avere diverse modalità di compressione.
  Il messaggio viene compresso e poi cifrato per aumentare l'efficacia degli algoritmi di cifratura.

È possibile usare queste procedure insieme, per avere servizi sia di autenticazione che di riservatezza. La figura *1.12* mostra tutto il processo crittografico quando vengono usati sia i servizi di confidenzialità che di autenticazione.
Per la distribuzione e le revoche delle chiavi pubbliche PGP utilizza la **web of trust**, meccanismo di fiducia alternativo alla **public key infrastructure** basata sulle *certificate authority*: una chiave può essere firmata da più di un utente per creare una *catena* di fiducia tra loro. I parametri per decidere *quanto* andare in profondità con la rete di fiducia sono liberamente impostati da ogni utente nel proprio client di posta.

1.2.9.2 S/MIME

MIME, descritto nell'RFC 1341, consente l'inserimento di elementi multimediali, l'inclusione di più contenuti e la scrittura in formati diversi da ASCII in qualsiasi messaggio email. Ogni formato multimediale compatibile ha un suo *MIME type*.
Sviluppato da RSA, **S/MIME** amplia i MIME type disponibli inserendo servizi crittografici. È equivalente a PGP ma usa le CA invece della web of trust.
È identificato con il MIME type *application/pkcs7-mime*.
Offre gli stessi servizi di PGP usando **SHA-1** o **MD5** per l'autenticazione del messaggio, **DSS** o **RSA** insieme a un digest per la firma, **3DES** a tre chiavi o **RC2**.

1.2.9.3 E-mail e privacy

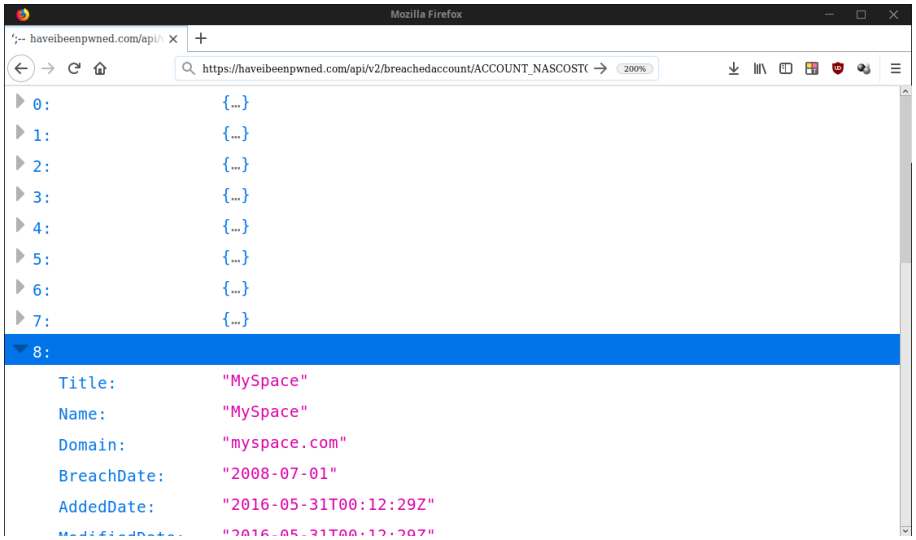


Figura 1.13: Un indirizzo e-mail di uno dei due autori è risultato pubblicato in 9 liste.

L'indirizzo e-mail viene molto spesso usato come mezzo identificativo per vari servizi su Internet. Questa comune pratica può, in alcuni casi, esporre informazioni personali al pubblico: ogni volta che un servizio è vittima di un attacco di diffusione l'indirizzo e-mail, spesso in associazione con una password o un hash della password, nome e cognome, data di nascita e altri dati personali, può essere pubblicato nelle liste di *indirizzi e-mail vulnerabili*, usabili da malintenzionati per produrre nuovi attacchi. Si può, per esempio, sfruttare l'abitudine diffusa fra gli utenti meno esperti di riutilizzare la stessa password in più servizi per mettere in pratica un furto d'identità nei social network.
Un sito che raccoglie dati di centinaia di *data breach* è `https://haveibeenpwned.com/`.

1.2.10 Sicurezza dei sistemi di gestione dei database

La sicurezza dei DBMS, *DataBase Management System*, è estremamente importante perché i database sono fondamentali per imprese e per enti statali, perchè contengono dati vitali. I database non sono solamente dei depositi di dati: anche la loro organizzazione e il contenuto sono informazioni preziose che devono essere protette.
La prevenzione dei sistemi di gestione dei database porta risultati di vario tipo.
Con il tempo, la comprensione dei problemi di protezione dei database è migliorata e sono stati sviluppati diversi controlli.

1.2.10.1 Concetti base

Un **database** è una raccolta di dati e un insieme di regole che organizzano i dati mediante determinate relazioni tra essi.
L'utente, attraverso queste regole, descrive il formato logico per i dati. I dati vengono memorizzati in file, il cui formato fisico non ha alcun interesse per l'utente.
L'**amministratore di sistema** è colui che definisce le regole di organizzazione dei dati e controlla gli utenti che possono accedere ed interagire con il database mediante un programma definito **gestore del database** o **sistema di gestione del database** (DBMS), detto *front end*.
Il file del database è costituito da **record**, ognuno dei quali contiene un gruppo di dati correlato: campo ed elemento.
La struttura logica dei database è chiamata **schema**, un utente può avere accesso solamente a un **sottoschema**, una parte del database.
Le regole del database prevedono che le colonne siano contrassegnate da nomi che sono chiamati **attributi** del database.
Una **relazione** è un insieme di colonne che specifica gruppi di valori correlati.
Gli utenti interagiscono con i programmi di gestione dei database dando comandi al DBMS, che recupera, modifica, aggiunge o elimina campi e record del database. Un comando viene definito **query**.
I sistemi di gestione dei database devono osservare determinate regole di sintassi per le query: la maggior parte dei linguaggi di interrogazione utilizza SQL.

Una volta selezionati i record, è possibile proiettare questi record su uno o più attributi tramite due tipologie di operazioni:

- *selezione*: estrazione di determinate righe dal DBMS;
- *proiezione*: definizione dei valori di alcuni campi e le colonne, di questi record.

Il risultato di un'operazione di selezione/proiezione è l'insieme dei valori di attributi specifici per i record selezionati.

1.2.10.2 Vantaggi dell'uso di database

Un database è una raccolta di dati che sono archiviati e mantenuti in una posizione centrale e a cui molti utenti hanno accesso. L'utente che accede al database non deve essere a conoscenza dell'organizzazione fisica di quest'ultimo, ma vede la disposizione logica unificata.

Un database offre molti vantaggi rispetto a un file system semplificato:

- *accesso condiviso*: insieme centralizzato di dati comuni a cui possono accede molti utenti;
- *ridondanza minima*: gli utenti non devono raccogliere e mantenere insieme i dati personali;
- *coerenza dei dati*: ogni modifica a un valore influisce su tutti gli utenti che possono accedervi;
- *integrità dei dati*: valori sono protetti da modifiche indesiderate accidentali:
- *accesso controllato*: gli utenti autorizzati possono accedere, visualizzare o modificare i valori dei dati.

Come evidente, gli interessi della protezione possono entrare in conflitto con le prestazioni: le misure per rafforzare la protezione aumentano la complessità e la dimensione del sistema informatico.
Gli interessi di protezione vanno, invece, a ridurre la capacità del sistema di fornire dati agli utenti, limitando le query.

1.2.10.3 Requisiti di protezione

I requisiti per la protezione dei sistemi di database sono:

- *integrità del database*: i database devono essere immuni dai problemi:
  - fisici: devono esistere delle protezioni contro cali di tensione, incendi, disastri ambientali e quant'altro;
  - logici: la struttura delle tabelle deve mantenere funzionalità e semantica;
  - dei singoli elementi: i dati presenti devono essere coerenti e veritieri.

  Solo gli utenti autorizzati sono responsabili dell'inserimento dei dati nel database; tuttavia, questi possono compiere degli errori, di conseguenza i DBMS compiono azioni speciali per aiutare ad intercettare gli errori nell'istante in cui vengono compiuti per poi correggerli. Il gestore del database, del sistema operativo e del sistema informatico si occupa di rendere il sistema integro. I dati presenti nel DBMS non devono essere danneggiati né illeggibili perché influirebbe sull'integrità del database.
  L'amministratore del database deve assicurarsi degli aggiornamenti di sistema per far sì che i dati siano protetti da qualsiasi tipo di danneggiamento.

Esso effettua differenti tipi di *backup*:

- **backup completo**: trasferimento di tutti i dati che occupa una grande quantità di spazio in memoria.
- **backup differenziale**: backup completo con salvataggio dei soli file modificati o aggiunti dall'ultimo backup completo. (maggior risparmio di memoria);
  I file modificati e i nuovi (fino al successivo backup completo) vengono copiati ad ogni backup differenziale.
  Il recupero dei dati avviene soltanto con l'ultimo backup completo inclusivo del backup differenziale;
- **backup incrementale**: copia dei dati, nuovi o modificati, creati e rinnovati dall'ultima esecuzione di un backup, sia incrementale che completo.
  Ogni file viene salvato una volta sola e la memoria viene risparmiata.
  Per ripristinare i file desiderati sono necessarie tutte le funzioni di sicurezza dal backup completo fino allo stato desiderato.

L'*integrità* degli elementi del database corrisponde alla loro correttezza o accuratezza. In un database può essere compiuta un'azione correttiva in tre modalità:

- applicazione *campi di controllo* con la verifica dei valori appropriati in una determinata posizione;
- verifica e *controllo degli accessi* dei dati che si trovano in una fonte centrale e delle eventuali modifiche di un valore;
- realizzazione del *registro delle modifiche* con i cambiamenti riportati nel DBMS. Il database mantiene i valori originali e quelli modificati.
  L'amministratore è in grado di annullare qualsiasi tipo di modifica effettuata per errore.

- *verificabilità*: inserimento di record per la verifica degli accessi al DBMS.
  Il campo mantiene il database integro e permette di scoprire chi e quando modifica i valori.
  Gli utenti possono entrare in maniera incrementale nel database: la query rileva un insieme di accessi sequenziali che, visualizzati completamente, mostrano i dati.
  Gli itinerari di controllo, per essere efficaci, devono includere gli accessi a livello di record, campo ed elemento.
  Se il sistema non tiene traccia dell'accesso dell'utente, si verificano problemi.
  Si possono conoscere i valori di determinati elementi senza accedervi direttamente e, di conseguenza, il registro dei record può sottostimare o sovrastimare le conoscenze effettive dell'utente.

- *controllo degli accessi*: l'amministratore del database specifica a chi deve essere consentito l'accesso a determinati dati e il DBMS impone tale criterio garantendo la query ai soli dati specificati.
  L'utente può determinare il valore di un dato leggendone altri, questo è chiamato *inferenza*. Si può accedere ai dati con l'inferenza senza dover accedere all'oggetto che li contiene.
  Limitare gli accessi per controllare l'inferenza vincola le interrogazioni degli utenti che non hanno alcuna interazione con i valori a cui non sono autorizzati. Controllare gli accessi, ricercando le possibili inferenze inaccettabili, degrada le prestazioni del DBMS.

- *autenticazione dell'utente*: un DBMS potrebbe richiedere un'autenticazione rigorosa dell'utente (autenticazione che si aggiunge a quella eseguita dal sistema operativo).
  Non esistono percorsi *trusted* (di fiducia) dal DBMS al sistema operativo, quindi il DBMS esegue un'autenticazione propria.

- *disponibilità*: il DBMS è un programma che utilizza risorse hardware e software. Tali risorse devono risultare disponibili, soprattutto in ambiti di lavoro con basi di dati centralizzate e condivise. Per esempio, se un server deve essere spento o riavviato per manutenzioni straordinarie è bene averne un altro di backup per permettere agli utenti di continuare ad usufruire dei dati.

2. WebMVA

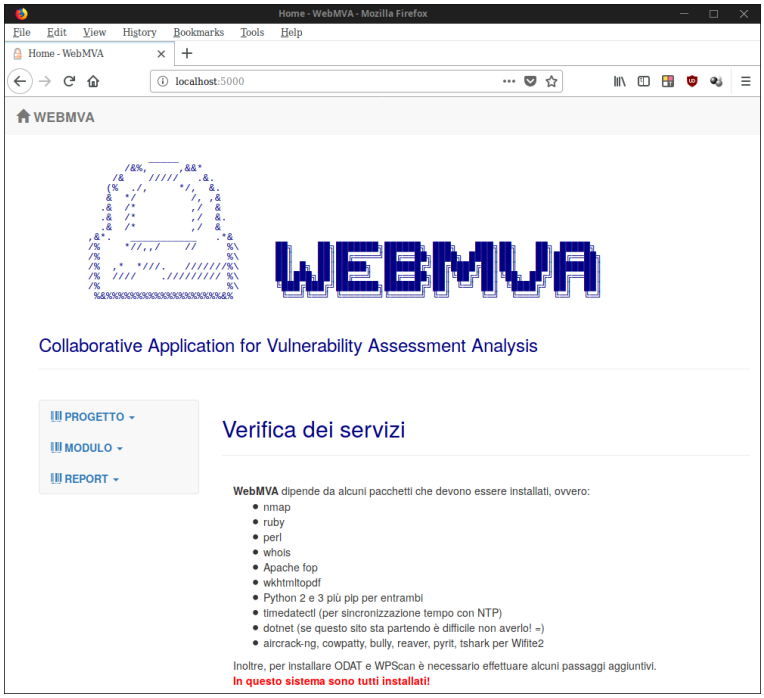


Figura 2.1: Homepage di WebMVA.

L'applicazione **WebMVA** è il risultato della ricerca svolta nel periodo di Stage++, iniziato a Febbraio 2018 sotto la supervisione del professor Fausto Marcantoni.
La proposta è quella di semplificare l'analisi delle vulnerabilità, assemblando una suite di strumenti che sia di facile utilizzo per gli ambiti più comuni e che produca dei report di analisi chiari e completi. All'inizio è stata svolta un'approfondita ricerca sugli strumenti di valutazione delle vulnerabilità presenti in rete, per poi catalogarli e stimarli secondo le loro caratteristiche. Per comprendere al meglio le loro funzionalità i tools sono stati testati in diversi ambienti di virtualizzazione. Al termine delle prove, è stato stilato un elenco con le loro funzioni, i vantaggi e gli svantaggi riscontrati nell'utilizzo.
Lo studio di un numero sempre maggiore di tools ha reso necessario porre dei criteri oggettivi di valutazione utilizzando una tabella per classificarli con valori decimali.

Per organizzare la ricerca, sono stati identificati criteri di valutazione oggettivi:

- **Facilità di installazione**
- **Modalità di servizio**
- **Livello di documentazione**
- **Database aggiornati**
- **Usabilità**
- **Tempo di elaborazione dati**
- **Chiarezza dei risultati**
- **Esportabilità dei risultati**
- **Portabilità**
- **Personalizzazione dello scan**
- **Modalità di scansione**.

La tabella, successivamente, è stata suddivisa in varie sezioni secondo le **modalità di scansione**.
Sono state realizzate sette sotto-tabelle valutative per ambito di scansione[1]:

- Network
- Web
- CMS
- DNS
- Database
- Information Gathering
- Wifi

Alcuni strumenti trascendono questa suddivisione ma sono stati "forzatamente" inseriti in una delle sottotabelle.
OpenDoor e *Wapiti*, ad esempio, sono in grado di eseguire attacchi SQL ma il loro scopo principale è quello di analizzare le vulnerabilità di un servizio web, perciò sono stati inseriti nella categoria Web.

Le sottotabelle valutative hanno permesso di scegliere i tools gratuiti con valutazione sufficiente su tutti i criteri e di inserirli in WebMVA, il tool collaborativo per l'analisi della ricerca delle vulnerabilità. Gli strumenti più efficienti sono diventati parte della sezione *Moduli* presenti nell'applicazione.

2.1 Applicazione

Figura 2.2: Logo di WebMVA.

WebMVA è un'applicazione open source pubblicata su GitHub (all'indirizzo `https://github.com/unicam-webmva/webmva`) sotto licenza MIT.
Combina i risultati ottenuti durante lo Stage in un framework di analisi delle vulnerabilità semplice ed efficace. L'interfaccia utente è *web-based* per permettere a più collaboratori di usufruire in contemporanea del servizio fornito.
Il programma è installabile su distribuzioni *Linux Debian* e sue derivate (come Ubuntu, Mint, Kali...). Questa scelta è legata alla facilità con cui è possibile installare nuovi software tramite *apt*, il sistema di gestione dei pacchetti di Debian.

Licenza MIT

La licenza **MIT**, chiamata anche *licenza X* o *licenza X11*, è stata realizzata dal *MIT*, *Massachusetts Institute of Technology*.
Essa deriva dalla licenza BSD, ma è priva della clausola di non responsabilità.
La licenza MIT garantisce le quattro libertà del software libero:

- **libertà di eseguire il programma**, utilizzarlo su qualsiasi tipo di sistema informatico, per qualsiasi tipo di attività e senza dover comunicare con lo sviluppatore o con qualche altra entità specifica. *(libertà 0)*;
- **libertà di studiare come funziona il programma e adattarlo alle proprie necessità** (L'accesso al codice sorgente è un prerequisito.)*(libertà 1).*;
- **libertà di ridistribuire copie** in modo da aiutare il prossimo *(libertà 2)*;
- **libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti**, cosicché tutta la comunità possa trarne vantaggio (L'accesso al codice sorgente è un prerequisito.) *(libertà 3).*

La licenza permette che il codice sorgente derivato e non derivato possa essere incluso all'interno di software proprietario.
La condizione basilare di tale licenza è il mantenimento dei *credits*: viene mantenuto un riferimento allo sviluppatore originale. è *permissiva*, consente il riutilizzo nel software proprietario sotto la condizione che la licenza sia distribuita con tale software.

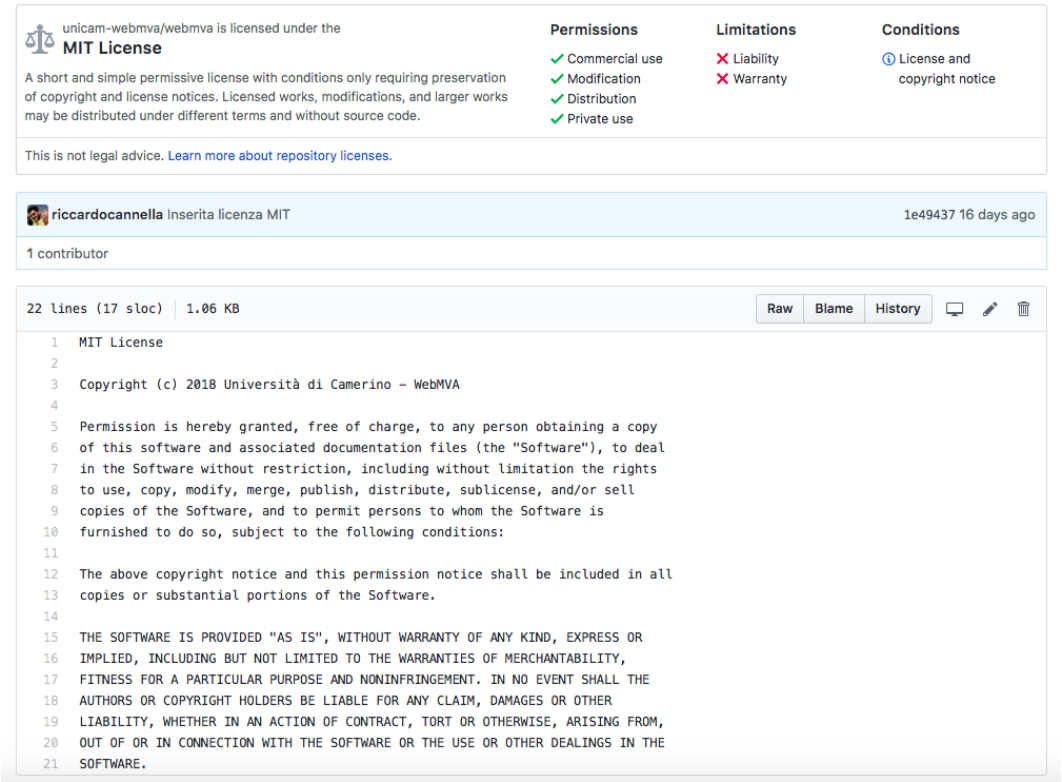


Figura 2.3: Licenza MIT del repository WebMVA.

Non pone vincoli all'utilizzo del software modificato, infatti, a differenza della GPL, con il software licenziato si può creare software proprietario. Nonostante ciò è anche GPL-compatibile: è possibile combinare e ridistribuire software GPL con codice.
È stata scelta questa licenza per rendere il codice accessibile dal più grande numero di sviluppatori possibile.

Configurazione iniziale

```
[10-07-2018_11:23:39] WebMVA - avviato su: rick-Inspiron-5559, Linux 4.15.0-24-
generic #26-Ubuntu SMP Wed Jun 13 08:44:47 UTC 2018
[10-07-2018_11:23:39] Caricato config file
[10-07-2018_11:23:40] Applicazione avviata
[10-07-2018_11:23:55] ProgettoController:Details - Richiesta GET con id 6
[10-07-2018_11:23:59] ProgettoController:Run - Richiesta POST con id 6
[10-07-2018_11:23:59] ProgettoController:Run -  Modulo NMAP - Ping Scan,
esecuzione comando: nmap  -sn -oX 20180710_112359_nmap_pingScan.xml --webxml
90.147.42.0/24 ...
[10-07-2018_11:24:01] ProgettoController:Run -  Fine esecuzione comando
[10-07-2018_11:24:01] ProgettoController:Run -  Fine esecuzione scan. Report con
id 9 salvato.
[10-07-2018_11:24:02] ReportController:Details - Richiesta GET con id 9
```

Figura 2.4: File di log di WebMVA.

La configurazione dei parametri di base di **WebMVA** avviene leggendo un file `json` all'avvio. Questo file si deve trovare nella cartella radice dell'applicazione e deve chiamarsi `webmvaSettings.json`. Se non viene trovato il file, l'applicazione verrà lanciata con i settaggi di default, che equivalgono a un file così formato:

```
{
"Porta": 5000,
"CartellaReport": "wwwroot/Report",
"Log": false,
"CartellaLog": "wwwroot/Log",
"TipoDB": "sqlite",
"ConnectionString": "Data Source=webmva.db"
}
```

I log generati dall'applicazione sono scritti su file e in contemporanea inviati allo standard output della console che ha avviato **WebMVA**. Il formato delle righe dei log è: `[dd-MM-yyyy_HH:mm:ss] Messaggio`. Il messaggio opzionalmente può contenere `NomeController:NomeMetodo` all'inizio della stringa.

2.1.1 Tecnologie

L'implementazione dell'applicazione ha portato ad utilizzare diverse tecnologie.

Figura 2.5: Tecnologie

Asp Net Core

ASP.NET Core è un framework web cross-platform, modulare, funziona sia su .NET Framework completo, su Windows, sia su .NET Core multipiattaforma .
DotNET Core è un framework open source, distribuito dalla Microsoft, che permette di scrivere applicazioni *cross-platform* sfruttando un sottoinsieme delle API di DotNET Framework.

È stata scelta questa tecnologia perché:

- il linguaggio di programmazione è facile e ben documentato,
- permette di effettuare operazioni fisiche sul filesystem,
- consente di avviare applicazioni esterne, controllarne lo stato ed reindirizzare l'output verso il back-end e usarlo nell'applicazione,
- esistono librerie di vario genere, sia sviluppate da Microsoft che da terzi, che permettono di ampliare le funzionalità fornite dal framework.

Entity Framework

Entity Framework è un insieme di tecnologie *ADO.NET* che permette di creare applicazioni data-oriented.
Esso astrae le funzioni del database per consentire di sviluppare gli oggetti da memorizzare senza doversi preoccupare della struttura e delle query al database.
Entity Framework supporta nativamente Sqlite, SQL Server, database in-memory, PostgreSQL, MySQL e Firebird. Esistono anche dei provider di terzi per gestire altri tipi di database.
È stato usato un database Sqlite per la sua semplicità e portabilità.

SqLite

SQLite è una libreria software scritta in C che implementa un DBMS SQL di tipo ACID incorporabile all'interno di applicazioni. Il creatore, D. R. Hipp, ha reso la libreria di pubblico dominio. SQLite permette di creare una base di dati incorporata in un unico file.

Git

Git è un *DVCS, Distributed Version Control System*, sistema software per il controllo di versione distribuito, realizzato nel 2005 da Linus Torvalds.
Git è uno strumento fondamentale per lo sviluppo, in ambito collaborativo e per l'implementazione di progetti open source. Supporta la realizzazione di ricerche non lineari: è possibile effettuare operazioni di branching and merging, rapide e comode.
Git permette di copiare in locale l'intera cronologia di sviluppo, e le modifiche vengono copiate da un repository a un altro.
Questi moglioramenti vengono importati come diramazioni aggiuntive di sviluppo, e possono essere fuse allo stesso modo di una diramazione sviluppata localmente.

Visual Studio Code

Visual Studio Code è un editor avanzato scritto in Electron, multipiattaforma, sviluppato da Microsoft, che permette di sviluppare applicazioni in molti linguaggi (per esempio C# e HTML).

Html-Css-Javascript

Html, Css e Javascript sono tre tecnologie sulle quali si fonda lo sviluppo front-end di un'applicazione o di un sito web.
HTML, *HyperText Markup Language*, è un linguaggio di markup; la sua rappresentazione viene gestita tramite gli stili CSS per adattarsi alle nuove esigenze di comunicazione e pubblicazione all'interno di Internet.
L'utilizzo del linguaggio **CSS**, *Cascading Style Sheets* separa i contenuti delle pagine HTML dalla loro formattazione, permette una programmazione più chiara e facile, garantisce il riutilizzo di codice ed una sua semplice manutenzione.
Javascript è un linguaggio di scripting orientato agli oggetti e agli eventi, utilizzato nella programmazione Web lato client per la creazione di effetti dinamici interattivi tramite funzioni di script invocate da eventi innescati a loro volta in vari modi dall'utente sulla pagina web in uso (mouse, tastiera, caricamento della pagina ecc...).

Bootstrap

Bootstrap è una raccolta di strumenti liberi per la creazione di applicazioni web.
Essa contiene modelli di progettazione basati su HTML e CSS, sia per la tipografia che per le varie componenti dell'interfaccia, come moduli, pulsanti e navigazione, così come alcune estensioni opzionali di JavaScript.

Bash

Bash è uno degli interpreti di comandi presenti nell'ambiente Unix-like. In ambito Linux e OSX, è probabilmente la shell più usata. Oltre all'inserimento manuale dei comandi tramite un terminale, reale o emulato, è possibile scrivere degli *script* contenenti una serie di comandi da eseguire in sequenza.
In **WebMVA** sono stati scritti degli script Bash per l'installazione iniziale delle dipendenze e il controllo delle stesse a runtime.

XML-XSL

Alcuni degli strumenti usati all'interno di **WebMVA** esportano i risultati ottenuti in formato **XML**, *eXtensible Markup Language*, metalinguaggio che permette di definire nodi, attributi e la loro semantica. Tramite XML è semplice scrivere e leggere dati complessi in maniera strutturata.
XSL, *eXtensible Stylesheet Language*, serve per definire *come* visualizzare un XML. Ogni browser è in grado di leggere un foglio di stile XSL ed applicarlo all'XML da renderizzare, proprio come accade per HTML e CSS.
In **WebMVA** è stato necessario scrivere da zero alcuni fogli di stile XSL per creare PDF a partire dai risultati XML degli strumenti.

2.1.2 Architettura

WebMVA è un'applicazione web locale che effettua la scansione di sistema e/o di rete utilizzando determinati tools.
È possibile avviare più scansioni in contemporanea e rivedere i loro risultati una volta ottenuti, anche da remoto.
Gli strumenti, estrapolati dalle sotto-tabelle valutative, rappresentano i Moduli, suddivisi per *modalità di scansione*.
L'utente crea un progetto, inserisce i Moduli desiderati e realizza la scansione su almeno un target.

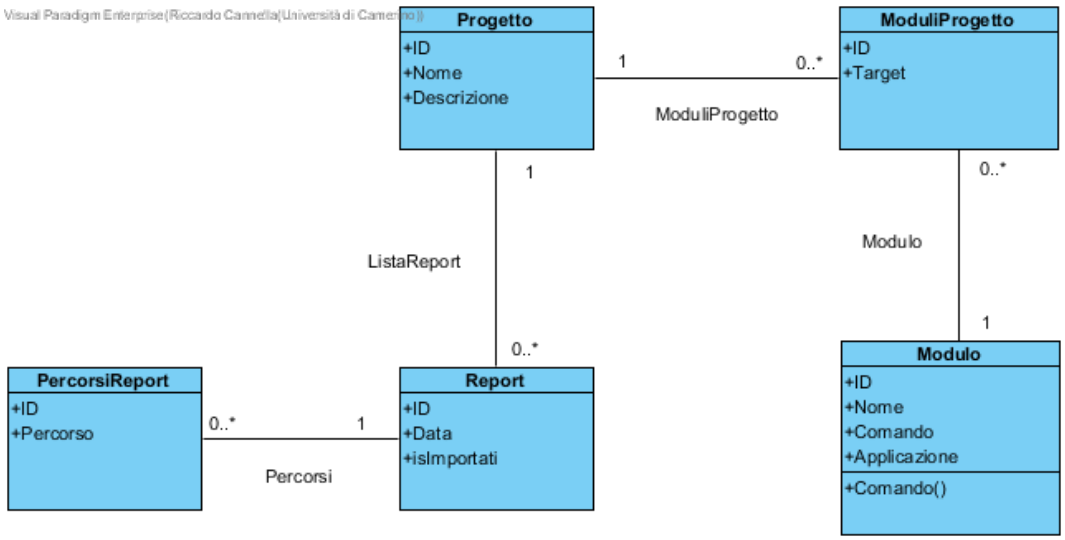


Figura 2.6: Un semplice diagramma delle classi di Webmva.

2.1.2.1 Progetto

L'utente crea un Progetto caratterizzato da un nome e una descrizione di cosa andrà ad analizzare.
In un progetto è possibile selezionare n moduli che rappresenteranno i veri e propri strumenti di scansione. Terminata la selezione, è possibile avviare la scansione tramite i moduli selezionati.
Al termine dell'operazione, l'utente ha la possibilità di visualizzare i risultati ottenuti da ciascun modulo e scaricarli come documentazione.
Nella sezione relativa ai Report è possibile scegliere in quale formato (nativo o pdf) e in che modalità scaricare il resoconto (pdf unico o cartella zip).

2.1.2.2 Modulo

I **Moduli**[2] sono i migliori scanner di vulnerabilità selezionati in ciascuna delle sette sotto-tabelle valutative.
L'utente è libero di crearne, in base alle personali necessità e agli obiettivi della propria ricerca.
Nella pagina di creazione del Modulo sono presenti i nomi delle sotto-tabelle per facilitare il fruitore del servizio. Il Modulo è un elemento generico, così da poter essere usato quante più volte possibili. Infatti, durante la creazione e modifica dei Moduli,

non viene mai richiesto un target ma solo le opzioni che si desidera attivare: in questo modo si possono creare più Moduli relativi allo stesso strumento con comportamenti diversi. L'obiettivo della scansione deve essere specificato, se necessario, nella pagina di modifica di un Progetto, dove l'utente selezionerà i Moduli e assocerà ad ognuno di essi il target desiderato.
Dalla pagina dei dettagli di un Progetto è possibile visualizzare un rapido resoconto dei Moduli scelti con i relativi target e lanciare la scansione automatizzata.

Nei Moduli di *WebMVA* è presente una generalizzazione per facilitare il lavoro di scrittura del codice tramite ereditarietà. Per quanto riguarda i database, nel modello relazionale non esiste una definizione per "sotto-tabelle", quindi ASP.NET, tramite Entity Framework, fornisce dei modelli per aggirare il problema. È stato scelto il modello di default TPH, *Table Per Hierarchy*. Questa metodologia fa in modo che Entity Framework crei nel database un'unica tabella Modulo con tutti i campi delle sotto-classi ed una colonna *Discriminator* per capire a quale classe appartiene un determinato record.

2.1.2.3 Report

Il **Report** è un resoconto di ogni scansione, lanciata all'interno di un Progetto, in cui vengono espressi tutti i risultati ottenuti dalla ricerca.
Non tutti i moduli inseriti generano un report: *Wifite* e *NoSqlmap* sono tools interattivi e non realizzano alcun resoconto.
I moduli *Nessus* e *OpenVAS* non sono stati gestiti, quindi non è possibile realizzare alcun report, ma è possibile importare report creati in precedenza dall'utente.

2.2 Moduli

2.2.1 Network

Nmap, **Nessus** e **OpenVAS** si focalizzano nella ricerca di vulnerabilità di rete.

2.2.1.1 Nmap

Figura 2.7: Logo di Nmap.

Nmap è uno strumento di scoperta ed analisi di reti TCP/IP scritto in C. Una parte del suo sorgente è stata riutilizzata nella fase iniziale dello sviluppo di Nessus. Il software, distribuito sotto la licenza GNU GPL, è in grado di individuare le porte aperte su un host e di ascoltare il traffico su una determinata porta o su un range di indirizzi IP, in modo da determinare quali servizi siano disponibili in una rete.
Sito web: `https://nmap.org`

Funzionalità implementate

In **WebMVA** sono state gestite diverse funzionalità di Nmap, tra cui:

Scan TCP/non TCP:

- `-sS` : TCP SYN Port Scan (Default);
- `-sT` : TCP Connect Port Scan;
- `-sU` : UDP Port Scan;
- `-sL` : produce solo una lista di target senza effettuare scan;
- `-sN` : solo host discovery, senza port scan;

Discovery:

- `-Pn`: solo port scan, senza host discovery;
- `-PR`: ARP Discovery;
- `-n` : disabilita la risoluzione DNS;

Altri:

- `-p PORTE` : scan sulle porte specificate;
- `-O` : scoperta del sistema operativo;
- `-sV` : scoperta della versione del servizio attivo;
- `-T` : tempo (può essere scelto su una scala da 0 a 5);
- `-f` : frammenta i pacchetti inviati.

2.2.1.2 Nessus

Figura 2.8: Logo di Nessus.

Nessus è un software proprietario di tipo client-server che, con lo scan e l'abilitazione di plugin configurabili, analizza e rileva le vulnerabilità presenti, creando report di facile analisi.
Sito web: `https://www.tenable.com/products/nessus-home`

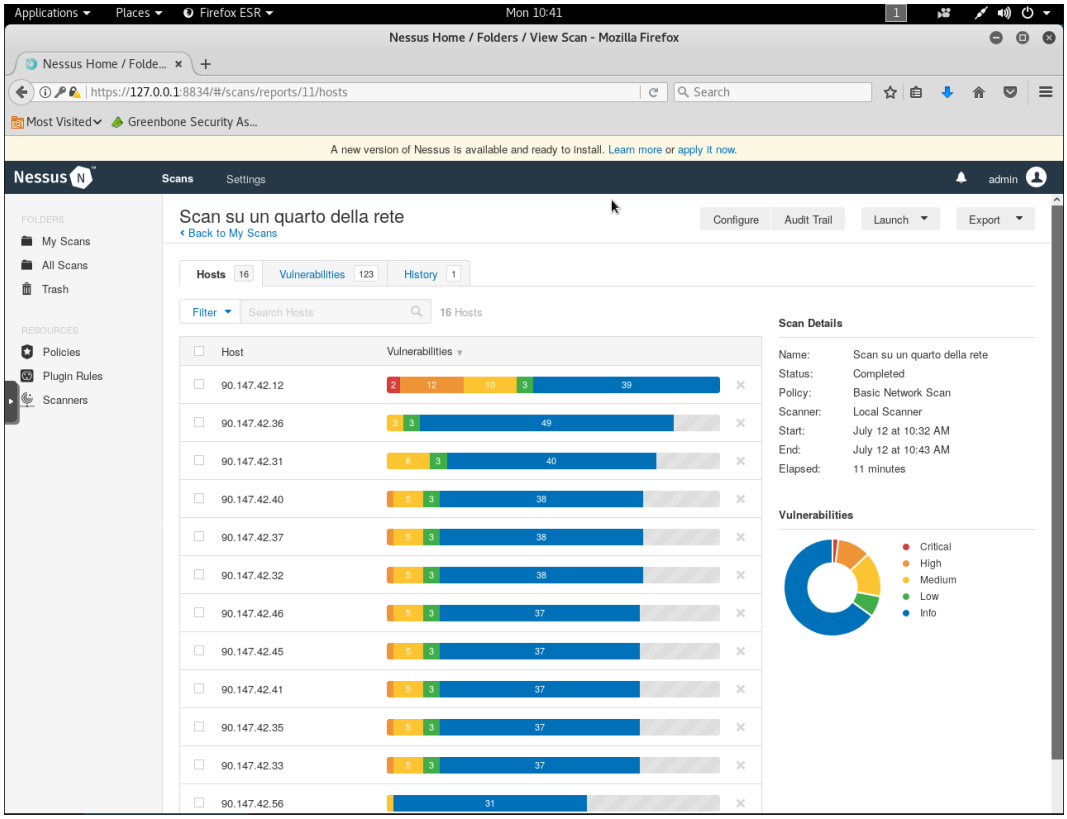


Figura 2.9: Screeen Nessus.

Funzionalità implementate

Fino alla versione 6 di *Nessus Home* era possibile usare delle Web API per creare nuovi scan, avviarli e generare dei report. A partire dalla versione 7, questa funzionalità è stata tolta[3].
In **WebMVA** Nessus non è gestito: il modulo reindirizza all'interfaccia web dove l'utente imposta i settaggi desiderati, avvia lo scan e genera un resoconto nel formato desiderato, importabile successivamente tra i Report di un Progetto.

2.2.1.3 OpenVAS

Figura 2.10: Logo di OpenVAS.

OpenVAS è un framework software di numerosi servizi e strumenti per gestire le scansioni di vulnerabilità. È un tool open source, distribuito sotto licenza GNU GPL. Lo scanner OpenVAS è un sistema completo che rileva i problemi di sicurezza in tutti i tipi di server e dispositivi di rete.
Sito web: `http://www.openvas.org/`

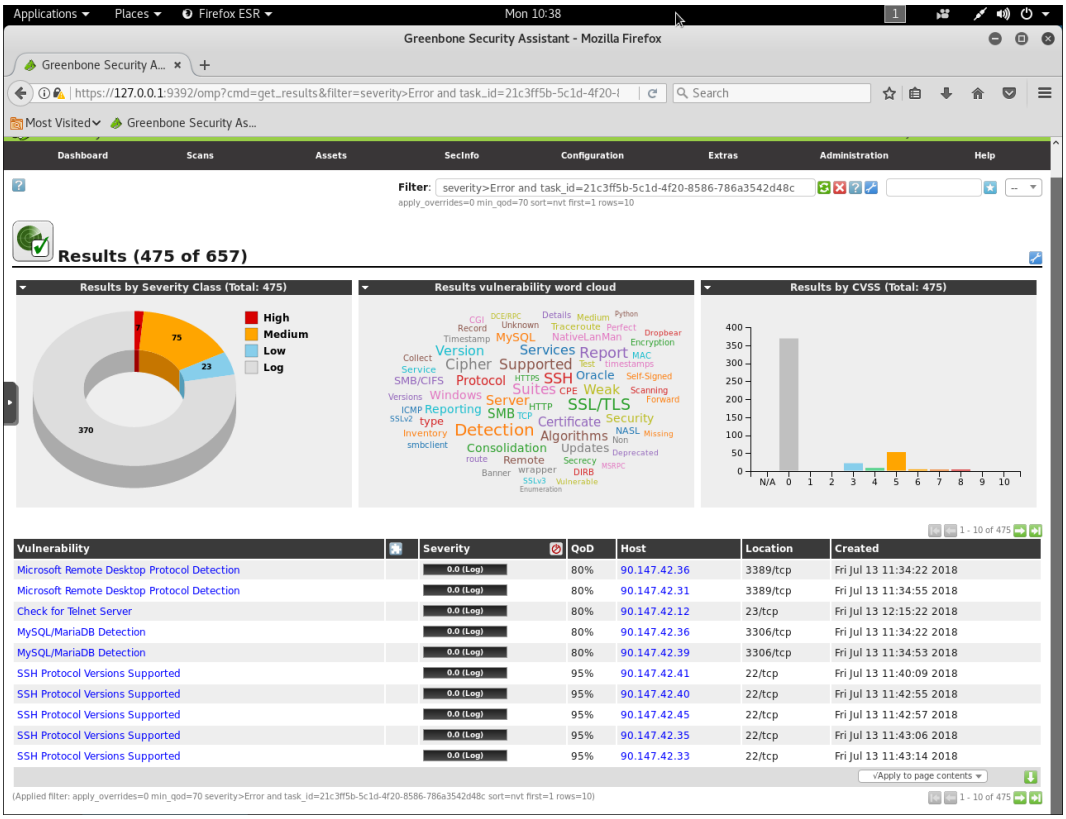


Figura 2.11: Screeen OpenVAS.

Funzionalità implementate

In **WebMVA** OpenVAS non è gestito: il modulo reindirizza all'interfaccia web dove l'utente imposta i settaggi desiderati, avvia lo scan e genera un resoconto nel formato desiderato, importabile successivamente tra i Report di un Progetto.

2.2.2 Web

Wapiti, **Opendoor** e **WAScan** effettuano scansioni per visualizzare vulnerabilità web.

2.2.2.1 Wapiti

Figura 2.12: Logo di Wapiti.

Wapiti è un tool di verifica della sicurezza delle applicazioni web, open source, scritto in Python (versione 3) e distribuito sotto la licenza GNU GPL versione 2.
Esegue scansioni "black-box", cioè non controlla il codice sorgente dell'applicazione, ma funziona come un fuzzer: scansiona le pagine web dell'applicazione, estrae collegamenti alla ricerca di script e di form in cui può inserire dei dati (invia payload e cerca stringhe speciali o comportamenti anomali).
Sito web: `https://sourceforge.net/projects/wapiti/`

Funzionalità implementate

In **WebMVA** sono state gestite diverse funzionalità di Wapiti, tra cui:

`-s`: definisce l'obiettivo dello scan e dell'attacco.
Si può scegliere tra:

- `Url`: effettua lo scan e l'attacco solo sul target definito dall'utente;
- `Page` : a partire dall'URL fornita dall'utente, effettua scansioni basate su query string;
- `Folder` : cerca di scansionare ed attaccare altre cartelle a partire dall'URL specificata dall'utente;
- `Domain` : effettua lo scan e attacca ogni URL con lo stesso dominio di quello inserito.

`--max-scan-time MINUTI` : il comando termina la scansione dopo gli X minuti messi in input;

`-v LEVEL` : setta il livello di verbosità dell'output.
È possibile scegliere tra : `quiet`, `normal` (di default), `verbose`;

`-m LISTADEIMODULI` : setta la lista degli attacchi dei moduli da lanciare contro il determinato target. Di default vengono utilizzati i moduli comuni. Si possono scegliere gli specifici moduli che si desiderano utilizzare.

2.2.2.2 Opendoor

Figura 2.13: Logo di Opendoor.

Opendoor è uno scanner di applicativi web multifunzionali, scritto in Python (versione 3), open source, distrubuito sotto licenza GNU GPL versione 3. Lo stumento utilizza un proxy server, accede alla web shell, ai sottodomini e ai dati nascosti. La scansione viene eseguita dal dizionario integrato e dai dizionari esterni.
Sito web: `https://github.com/stanislav-web/OpenDoor`

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

`-p PORTA` : setta la porta del client;

`-m METODO` : setta il metodo richiesto;

`-d DELAY`: imposta il ritardo tra le richieste;

`-r RETRIES` : imposta il numero maggior numero di volte che si riprova a connettersi dopo aver perso la connessione;

`--timeout TIMEOUT` : imposta il timeout della richiesta;

`--accept-cookies` : accetta e instrada i cookies dalle risposte.

2.2.2.3 WAScan

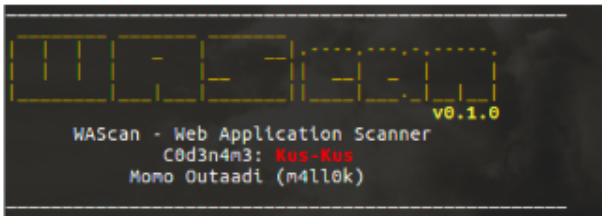


Figura 2.14: Logo di WAScan.

WAScan, *Web Application Scanner*, è un tool scritto in Python (versione 2.7) e distrubuito sotto licenza GNU GPL versione 3. WAScan trova le vulnerabilità col metodo "black-box": scannerizza le pagine di un sito per estrarre link e form, attacca gli script, invia payload, ricerca di messaggi di errore.
Sito web: `https://github.com/m4ll0k/WAScan`

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

`-a (user:pass)`: permette di definire l'autenticazione HTTP base;

`-A USER-AGENT` : imposta il valore dell'header dello user-agent HTTP;

`-c COOKIE` : imposta il valore dell'header del cookie HTTP ;

`-r RANDOM-USER` : setta il valore dell'haeder del random user agent;

`-t TEMPO` : definisce il tempo di attesa prima della connessione;

`-v` : aumenta verbosità.

2.2.3 CMS

Droopescan, **Joomscan** e **Wpscan** si incentrano nella ricerca di vulnerabilità dei CMS.

2.2.3.1 Droopscan

Figura 2.15: Logo di Droopscan.

Droopescan è uno scanner di sicurezza CMS, open source, scritto in Python (versione 3), distribuito sotto licenza GNU Affero GPL versione 3.
Esso esegue quattro tipi di test:

- **Controllo plug-in**: esegue richieste HTTP e restituisce un elenco di tutti i plug-in trovati nell'host di destinazione;
- **Controllo dei temi**: esegue richieste HTTP e restituisce un elenco di tutti i temi trovati nell'host di destinazione;
- **Controllo della versione**: scarica diversi file e, in base ai loro checksum, restituisce un elenco di tutte le possibili versioni;
- **Controllo url interessanti**: controlla la presenza di URL interessanti.

Sito web: `https://github.com/droope/droopescan`

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

`scan nome_del_CMS`: permette di scegliere il CMS che si vuole testare;

`-e` : permette di scegliere l'obiettivo dello scan e dell'attacco:

- `p` - plugins ;
- `t` - themes ;
- `v` - version ;
- `i` - url interessanti;
- `a` - tutti.

2.2.3.2 Joomscan

Figura 2.16: Logo di Joomscan.

Joomscan è uno strumento sviluppato in Perl, multi-piattaforma, open source, distribuito sotto licenza GNU GPL versione 3.
Joomscan è un tool che verifica l'installazione di Joomla per noti plug-in vulnerabili ed errori di configurazione di sicurezza di base.
Esso ha le seguenti caratteristiche:

- **Rilevamento della versione esatta**: lo scanner individua la versione con maggiore precisione rispetto al tag del generatore di meta;
- **Rilevamento del plugin presenti nel server**;
- **Rilevamento dei problemi di sicurezza più noti di Joomla Core**.

Sito web: `https://github.com/rezasp/joomscan`

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

`-ec` : cerca di elencare i componenti;

`-a USER-AGENT` : Permette di usare uno specifico user agent.

2.2.3.3 WPScan

Figura 2.17: Logo di WPscan.

WPScan è un web scanner creato per analizzare e cercare falle di sicurezza all'interno della piattaforma di blogging WordPress. Esso è sviluppato in Ruby, ha doppia licenza: commerciale e gratuita (open-source). Utilizza la modalità di scansione black-box.
Sito web: `https://github.com/wpscanteam/wpscan`

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

`--batch` : serve a rendere più automatica la scansione. Comportamento predefinito in WebMVA;

`-f` : comando che non verifica se sul target è in esecuzione WordPress;

`-r USER-AGENT` : usa uno user agent random;

`-v` : aumenta la verbosità;

`-a` : usa uno specifico user agent;

`-cookie` : legge da dove vengono i cookie;

`--disable-tls-checks` : disabilita la verifica dei certificati SSL/TLS.

2.2.4 DNS

Dnsrecon, **DnsEnum** e **Fierce** individuano vulnerabilità dei DNS.

2.2.4.1 DnsRecon

DNSRecon è uno scanner scritto in Python (versione 3), distribuito soot licenza GNU GPL versione 2, offre la possibilità di eseguire:

- Controllo di tutti i record DNS per i trasferimenti di zona;
- Enumerazione dei record DNS generali per un determinato dominio (MX, SOA, NS, A, AAAA, SPF e TXT);
- Esegue l'enumerazione dei record SRV comune;
- Controlla i record memorizzati nella cache di un server DNS per ottenere i record A, AAAA e CNAME relativi agli host forniti in un file di testo.

Sito web : `https://github.com/darkoperator/dnsrecon`.

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

`-a` : esegue AXFR con l'enumerazione standard;

`-n NAMESERVER` : imposta il domain server da utilizzare;

`-s` : esegue la ricerca inversa degli intervalli IP nel record SPF di dominio mirato con l'enumerazione standard;

`-g`: esegue l'enumerazione Google con l'enumerazione standard;

`-k` : esegue l'enumerazione crt.sh con l'enumerazione standard;

`-b` : usa uno specifico user agent;

`-w` : imposta un'analisi whois quando si esegue una query standard;

`-z` : esegue una DNSSEC Zone Walk con l'enumerazione standard.

2.2.4.2 DnsEnum

DnsEnum è uno strumento sviluppato in Perl per l'enumerazione DNS, open source. Esso offre la possibilità di:

- ottenere informazioni su un determinato host;
- prendere i name server infiltrati;
- ottenere i record MX;
- ottenere nomi e sottodomini extra tramite google scraping;
- calcolare intervalli di rete del dominio di classe C ed eseguire query whois su questi.

Sito web : `https://github.com/fwaeytens/dnsenum`.

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

`-t TIMEOUT` : setta il valore del timeout in secondi;

`-d DELAY` : imposta il massimo valore in secondi del tempo di attesa tra le domande whois;

`-p PAGE` : con questo valore si può impostare il numero di pagine di ricerca di google da elaborare durante la scelta (scrematura dei nomi);

`-s NUMERO` : imposta il massimo numero di sottodomini da ricercare;

`--threads NUMERO`: specifica il numero di threads da usare;

`-v` : aumenta la verbosità;

`-r` : imposta la ricorsione su sottodomini;

`-w` : si eseguono le query whois su intervalli di rete di classe c;

`--noreverse` : con questo comando, si saltano le operazioni di ricerca inversa.

2.2.4.3 Fierce

Fierce è uno strumento di ricognizione per localizzare lo spazio IP non contiguo, scritto in Python, open soure, distribuito sotto licenza GNU GPL versione 3.
Esso è uno scanner che aiuta a localizzare spazi IP e nomi host non contigui rispetto a domini specifici. Individua i probabili obiettivi sia all'interno che all'esterno della rete.
Sito web : `https://github.com/mschwager/fierce`.

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

`--dns-server NAME-SERVER` : setta il valore del DNS server;

`--subdomain SUBDOMAIN` : permette di selezionare quali sottodomini da analizzare;

`--connect` : si cerca di effettuare connessioni HTTP su domini rilevati con tale flag;

`--wide` : cerca domini su tutti gli IP del / 24 di un dominio scoperto.

2.2.5 Information Gathering

OSINT

Nel metodo **OSINT**, *Open Source Intelligence*, le informazioni sono trovate pubblicamente ed usate per ulteriori analisi.
La relazione tra le varie forme di informazioni raccolte può essere estremamente vantaggioso per l'attaccante.

2.2.5.1 Maltego

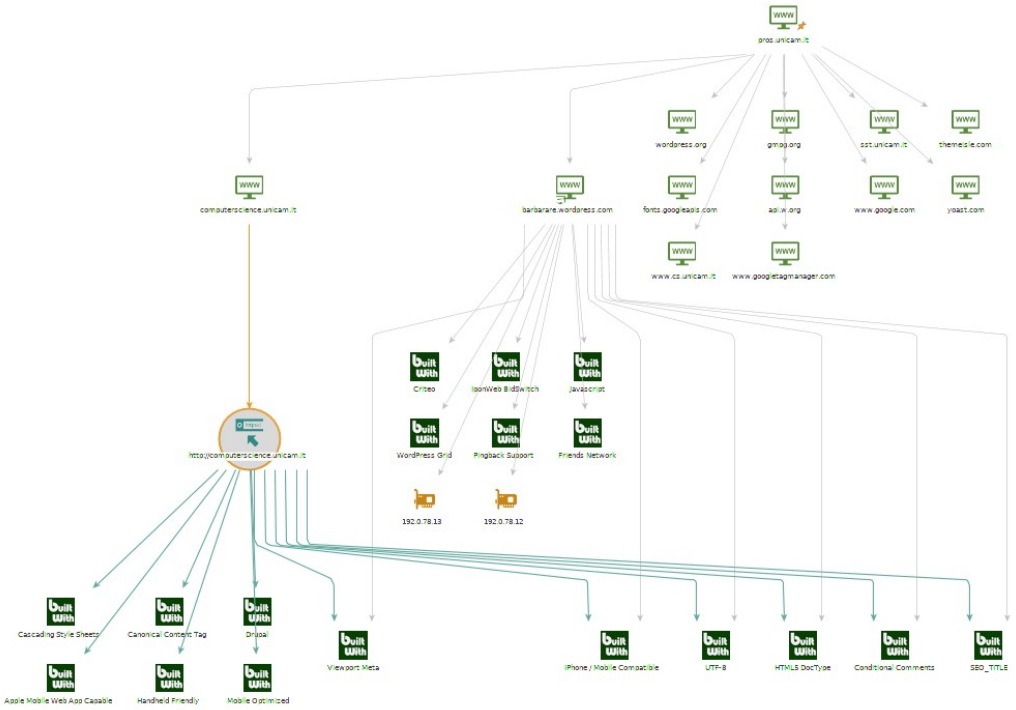


Figura 2.18: Esempio esecuzione Maltego.

Maltego[4], distribuito dalla software house *Paterva*, è un tool che utilizza *OSINT* mostrando l'interconnessione tra i vari tipologie di informazione e individua alcune relazioni sconosciute tra loro. Si trovano le relazioni dell'utente in considerazione: il suo profilo sociale, gli amici, le società correlate, i siti web visualizzati.
Si ricavano informazioni relative a qualsiasi infrastruttura: relazioni tra domini, nomi DNS, blocchi di rete.
Il client Maltego invia la richiesta ai seed server su HTTPS.
La richiesta dal seed server viene fornita ai server TAS, passati ai provider di servizi. I risultati della richiesta sono restituiti al client Maltego[5].
Maltego ha due versioni: gratuita (community) e a pagamento.
La versione *community* testata, presenta dei limiti: si ottengono al massimo 12 risultati per trasformazione, è possibile effettuare query solamente ai server Paterva e la comunicazione tra client e server non è né cifrata né compressa.

Infoga, **Infoga-email** e **Sublist3r** ricercano il maggior numero di informazioni relative ad un determinato target.

2.2.5.2 Infoga

Infoga è uno strumento, open source, scritto in Python (versione 3). Raccoglie informazioni sugli account e-mail (ip, hostname, paese, ...) da diverse fonti pubbliche (motori di ricerca, key server PGP e shodan) e controlla se le e-mail sono trapelate utilizzando l'API delle e-mail compromesse[6].
Sito web : `https://github.com/m4ll0k/Infoga`.

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

`-s` : è possibile scegliere tra le seguenti fonti:

- `google`;
- `bing`;
- `yahoo`;
- `ask`;
- `baidu`;
- `dogpile`;
- `exalead`;
- `pgp`.

`-v` : aumenta la verbosità;

`-b` : verifica se l'e-mail è stata violata (Infoga Email)

2.2.5.3 Sublist3r

Figura 2.19: Logo di Sublist3r.

Sublist3r è uno strumento scritto in Python, open source, progettato per enumerare sottodomini di siti Web che utilizzano OSINT. Permette di raccogliere i sottodomini a cascata del dominio ricercato usando i più noti motori di ricerca (Google, Yahoo, Bing, Baidu e Ask). Con Sublist3r è possibile elencare i sottodomini usando Netcraft, Virustotal, ThreatCrowd, DNSdumpster e ReverseDNS.
Sito web: `https://github.com/aboul3la/Sublist3r`

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

`-e` : è possibile definire quali i motori di ricerca per lo scan :

- `google`;
- `baidu`;
- `yahoo`;
- `bing`;
- `ask`;
- `netcraft`;
- `dnsdumpster`;
- `virustotal`;
- `threatcrowd`;
- `sslcertificates`;
- `passivedns`.

`-t THREADS` : imposta il numero di threads da utilizzare per la scansione.

2.2.6 Database

Sqlmap, **Odat** e **NoSqlmap** si concentrano nella ricerca di vulnerabilità dei database relazionali, Oracle e non relazionali.

2.2.6.1 SqlMap

Figura 2.20: Logo di Sqlmap.

SqlMap è uno strumento open source, scritto in Python, distribuito sotto licenza GNU GPL. Esso verifica e testa le vulnerabilità di tipo SQL Injection.
Il software consente l'accesso al database, permette la sua modifica e l'eliminazione dei dati.
Tra i vari tipi di SQL Injection, SQLMAP supporta:

- **B** : Boolean-based blind SQl Injection;
- **E** : Error-based SQL Injection;
- **U** : UNION query SQL Injection;
- **S** : Stacked queries SQL Injection;
- **T** : Time-based blin SQL Injection.

Sito web : `https://github.com/sqlmapproject/sqlmap`.

Funzionalità implementate

In **WebMVA** sono state gestite diverse funzionalità di SqlMap, tra cui:

`--auth-cred CREDENZIALIAUTENTICAZIONE` : le credenziali HTTP da usare;

`-o` : attiva tutte le ottimizzazioni;

`--dbms-cred CREDENZIALIDBMS` : definisce le credenziali di autenticazione per il DBMS;

`-a` : ricava tutte le informazioni possibili da back end;

`--users`/`--passwords`/`--privileges`/`--roles` : si deducono la lista di utenti/gli hash delle password/privilegi/ruoli degli utenti del DBMS;

`--dbs` : ricava i nomi dei database presenti nel DBMS;

`--tables`/`--columns` : si traggono i nomi delle tabelle/colonne presenti nel DBMS;

`--dump-all` : si acquisiscono tutti i record presenti nel DBMS;

`--common-tables` /`--common-columns` : controlla l'esistenza di tabelle/ colonne comuni;

2.2.6.2 Odat

Figura 2.21: Logo di Odat.

Odat, *Oracle Database Attacking Tool*, è uno strumento di penetration testing, open source, scritto in Python, distribuito sotto licenza GNU GPL versione 3, compatibile solo su Linux. Verifica da remoto la sicurezza di database Oracle.
Odat utilizza un database Oracle in ascolto remoto, trova SID e credenziali validi per la connessione. Può essere usato, qualora si abbia un account Oracle, per incrementare i privilegi e diventare DBA o SYSDBA o eseguire comandi di sistema per andare sul sistema operativo che ospita il database.
Sito web : `https://github.com/quentinhardy/odat`.

Funzionalità implementate

In **WebMVA** sono state gestite diverse funzionalità di Odat, tra cui:

`-p` : definisce la porta del DB;

`-d` : imposta il System ID (SID) Oracle;

`-U` : determina l'username Oracle;

`-P` : ottiene la password Oracle;

`-v` : aumenta la verbosità;

`--test-module` : forza solo l'assessment, senza attaccare;

`all` : usa tutti i moduli, utile per iniziare;

`tnspoison` : cerca di sfruttare l'exploit TNS poison (CVE-2012-1675);

`tnscmd` : cerca di ottenere informazioni dal listener TNS senza autenticazione;

`smb` : se il DB è installato su Windows come servizio utente (non di sistema) tenta di catturare un'autenticazione SMB;

`passwordstealer` : ottiene gli hash delle password Oracle;

`passwordguesser` : ricerca account Oracle validi da remoto.

2.2.6.3 NoSqlMap

Figura 2.22: Logo di NoSqlmap.

NoSqlmap è uno strumento di sviluppo automatizzato, basato su Python, open source, progettato per controllare e automatizzare gli attacchi di iniezione e sfruttare le debolezze di configurazione predefinite nei database NoSQL. è destinato ad attaccare le applicazioni web usando NoSQL per divulgare i dati dal database. Le funzioni di NoSqlmap permettono di impostare:

- gli hosts di destinazione: il server Web di destinazione o il server MongoDB che si vuole attaccare;
- la porta TCP per l'applicazione Web qualora sia l'obiettivo;
- il percorso URI;
- il metodo di richiesta HTTP (le richieste POST sono in fase di implementazione);
- l'IP Mongo e la Shell locale: qualora si attacchi un'istanza MongoDB direttamente all'IP per clonare i database delle vittime o aprire le shell Meterpreter;
- la porta listener di shell;
- caricare e salvare i files delle opzioni, della richiesta Burp.

Sito web : `https://github.com/sqlmapproject/sqlmap`.

Funzionalità implementate

NoSqlmap è un tool interattivo e non è stata implementata nessuna funzionalità.
Il sistema di ripulitura dello schermo per la navigazione nei menu rende impossibile catturare l'output prodotto, perciò non viene generato nessun Report.

2.2.7 WiFi

Wifite si sofferma nella nell'indagine delle vulnerabilità legate alla rete wifi.

2.2.7.1 Wifite

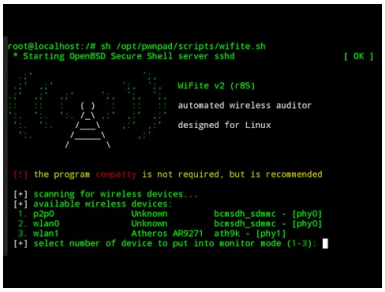


Figura 2.23: screeen di avvio Wifite.

Wifite è uno strumento di attacco wireless automatico, distrubuito sotto licenza GNU GPL versione 2, scritto in Python, compatibile solo con Linux.
Esso ha le seguenti caratteristiche:

- ordina gli obiettivi in base alla potenza del segnale;
- intacca i punti di accesso più vicini;
- de-autentica automaticamente i client delle reti nascoste per rivelare SSID numerosi filtri specificando cosa attaccare;
- cambia MAC a un indirizzo casuale prima di attaccare.

Sito web: `https://github.com/derv82/wifite2`.

Funzionalità implementate

Non è possibile catturare l'output prodotto da questo software e non viene generato nessun Report.
È possibile specificare alcuni parametri per pre-impostare il comportamento del software:

- `-c CANALE` : seleziona il canale da scansionare;
- `-co` : visualizza solo i target che hanno dei client associati;
- `--keep-ivs` : salva gli IV per tentare un attacco successivamente;
- `--new-hs` : cattura nuovi handshake e ignora quelli eventualmente già salvati;
- `-wep`/ `-wpa` : visualizza solo i target con autenticazione WEP/WPA;
- `-wps` : visualizza solo i target in cui è abilitato WPS.

3. Analisi dei Report

Il Capitolo presenta vari documenti contenenti i report delle scansioni di differenti modalità di scansione. Nell'ultimo periodo della ricerca sono stati effettuati numerosi test per verifica l'efficienza dell'applicazione e per effettuare eventuali modifiche. La realizzazione di diversi report ha permesso di analizzare nel dettaglio le funzionalità di ciascun tool inserito.
A termine di ogni scansione, sono stati comparati i risultati ottenuti dagli strumenti appartenenti alle stesse categorie di modalità di scansione e valutate le differenze.

3.1 Comparazione dei resoconti

Per ogni categoria di modalità di scansione è stato scelto un target comune e sono state effettuate diverse scansioni.
Una volta terminate le scansioni, è stato realizzato un resoconto con risultati ottenuti e sono state tratte delle conclusioni riguardo l'efficacia degli strumenti.

3.1.1 Tools di scansione della rete

Prima di avviare la scansione, è stata verificata la raggiungibilità dei moduli che presentano un'interfaccia web esterna.
Nmap è gestito direttamente dall'applicazione. I server Nessus e OpenVAS sono situati su una macchina virtuale di Proxmox, raggiungibile all'indirizzo 90.147.42.43.
Il test è stato effettuato su un quarto della rete pubblica `90.142.47.0`.

- *Nmap*:
  target `90.147.42.0/26`,
  comando eseguito `nmap -A -T4 -vv`.
  La scansione ha generato un report dopo pochi minuti. Sono stati identificati diversi hosts e definiti servizi, porte e versioni attivi.

- *Nessus*:
  target `90.147.42.0/26`,
  tipo di scansione eseguita: `Basic network scan`.
  La scansione è stata completata in breve tempo. Il report generato elenca le vulnerabilità per ogni host senza fornire ulteriori dettagli. Le vulnerabilità sono catalogate in base alla gravità, in 4 sezioni: low, medium, high, critical Sono state evidenziate 2 vulnerabilità critical, 20 high, 69 medium e 37 low.

- *OpenVAS*:
  target `90.147.42.0/26`,
  tipo di scansione eseguita: `Full and fast ultimate`.
  La scansione ha impiegato quasi una giornata, nonostante ci fossero 62 hosts da scansionare.
  Il report è molto dettagliato, ben strutturato, ha evidenziato diverse vulnerabilità con riferimento alle liste CVSS. Ogni vulnerabilità viene descritto descritta, riportato il metodo di scoperta, fornita una soluzione.
  Sono state evidenziate 7 vulnerabilità gravi, 75 medie e 23 basse.

I tre tools combinati offrono una solida base per effettuare vulnerability assessment. Nmap è uno software che permette la semplice visualizzazione dei sistemi presenti nella rete e non ricerca le vulnerabilità. L'interfaccia web di Nessus fornisce opzioni per comprendere le falle dei sistemi-target. I reports generati sono troppo sintetici. OpenVAS, pur essendo lo strumento più lento, produce resoconti molto efficienti contenenti sia i workarounds sia patches esisitenti.

3.1.2 Tools di scansione Web

Sono state effettuate due serie di scansioni

- con target `http://quasylab.unicam.it/`;
- con target `http://pros.unicam.it/`.

Risultati della scansione con target `http://quasylab.unicam.it/`:

- *Wapiti*:
  target `http://quasylab.unicam.it`,
  comando `wapiti -m ,crlf,exec,file,htaccess, permanentxss,shellshock -S aggressive -v 2`.
  La scansione non ha prodotto alcun risultato.

- *Opendoor*:
  target `http://quasylab.unicam.it`,
  comando `opendoor --accept-cookies`.
  Sono stati trovati diversi file a cui è stato negato l'accesso e due risorse disponibili: `index.html` e la cartella `images`.

- *WAScan*:
  target `http://quasylab.unicam.it`,
  comando `wascan.py -s [0,2,4] -m GET -r -n -v`.
  Identifica il server, *Apache 2.4.18*, e il sistema operativo del server, *Ubuntu*.

Il risultato della scansione non è stato significativo. Il sito è ancora in costruzione, per questo i risultati ottenuti sono corretti.

Risultati della scansione con target `http://pros.unicam.it`:

- *Wapiti*:
  target `http://pros.unicam.it`,
  comando `wapiti -m ,crlf,exec,file,htaccess,`
  `permanentxss,shellshock -S aggressive -v 2`.
  La scansione ha impiegato diverso tempo ma è stata efficace. È stata realizzata con successo una Blind SQL Injection sul parametro `wpuf_login`: è stato eseguito il comando `sleep`.
  Altri tipi di anomalie riguardano i timeout causati dall'inserimento di un payload maligno in alcuni campi delle form presenti nel sito.

- *Opendoor*:
  target `http://pros.unicam.it`,
  comando `opendoor --accept-cookies`.
  Sono stati trovati diversi file a cui è stato negato l'accesso e alcune pagine che reindirizzano ad altre.

- *WAScan*:
  target `http://pros.unicam.it`,
  comando `wascan.py -s [0,2,4] -m GET -r -n -v`.
  Ha rilevato il server *Apache 2.4.18*, il CMS *Wordpress*, il linguaggio installato nel webserver *ASP* e il sistema operativo *Ubuntu*.

Wapiti è l'unico tool che ha trovato delle vulnerabilità nel target. I risultati ottenuti dagli altri tool non sono soddisfacenti.

3.1.3 Tools di scansione dei CMS

Il test è effettuato su macchine virtuali che hanno installato al loro interno un solo CMS. Per *Wordpress* è stato testato `90.147.42.38`, per *Drupal* `90.147.42.103`, per *Joomla* `90.147.42.39`. È stato verificato da browser che le tre macchine target rispondessero correttamente.
Avendo già accesso alle macchine e agli account administrator di ogni CMS è stata verificata concretamente l'accuratezza degli strumenti.
I risultati otteniti sono:

- *Wordpress*:
  - **Droopescan**:
    target `90.147.42.38`,
    comando `droopescan scan`.
    Ha identificato correttamente il plug-in *Akismet*, una lista di possibili versioni e i temi installati nel gestionale senza riuscire a determinarli con precisione;
  - **WPScan**:
    target `90.147.42.38`,
    comando `wpscan.rb --batch -f -r`.
    Ha riportato la versione corretta del gestionale *4.7.4*, il tema in uso *twentyseventeen* versione *1.2* e le rispettive vulnerabilità. Non stati espressi gli altri due temi installati *twentyfifteen, twentysixteen.* I plug-in trovati non rispecchiano quelli installati: *wp-super-cache* non è installato.

  La combinazione di entrambi i tools permette di comprendere molte vulnerabilità. Droopescan è uno strumento affidabile ma fornisce meno informazioni di WPScan, che mostra dettagli sulle vulnerabilità mostrando i rispettivi link al CVE. WPScan non rileva correttamente i plug-in installati.

- *Drupal*:
  - **Droopescan**:
    target `90.147.42.103`,
    comando `droopescan scan`.
    Ha identificato la versione corretta del CMS *7.54*, i due temi installati *Garland* e *Seven* ma non altri due temi presenti: *Bartik* e *Stak.* Ha notificato che l'amministratore del CMS è quello di default, *admin.*
    I plug-in trovati sono stati molteplici, tutti corretti tranne *profile*, pagina esistente ma non presente nella lista dei plug-in del CMS.
  - **Drupwn**:
    target `90.147.42.103`,
    comando `drupwn enum --users --modules --dfiles --themes --thread 6`.
    Ha individuato la versione corretta del CMS *7.54* e il nome utente dell'amministratore *admin.* Non ha trovato alcun riferimento dei moduli e dei temi.

  Il modulo Drupwn si è rivelato fallimentare. I risultati raggiunti da Droopescan permettono di avere un quadro generale abbastanza completo del gestionale-target della scansione.

- *Joomla*:
  - **Droopescan**:
    target `90.147.42.39`,
    comando `droopescan scan`.
    Ha definito la lista di possibili versioni, non ha segnalato la versione specifica. Ha trovato la pagina di amministrazione predefinita.
  - **Joomscan**:
    target `90.147.42.39`,
    comando `joomscan.pl -ec`.
    Ha identificato la versione esatta del CMS *3.7.0* ed informa che questa non è vulnerabile. Ha trovato la pagina di amministrazione di default.

  I risultati della scansione non sono sufficienti per un paragone delle funzionalità degli strumenti, per questo, è stata effettuata una scansione su un target reale, `www.iononcrollo.org`.

  - **Droopescan**:
    target `http://www.iononcrollo.org/`,
    comando `droopescan scan`.
    Ha individuato la lista di possibili versioni del CMS, non segnalando la versione specifica. Ha trovato la pagina di amministrazione predefinita.
  - **Joomscan**:
    target `www.iononcrollo.org`,
    comando `joomscan.pl -ec`.
    Ha riconosciuto la versione esatta del CMS *3.6.5* e informa che questa non è vulnerabile. Ha trovato la pagina di amministrazione di default e il file `robots.txt`, da cui ha ricavato alcuni URL nel sito.

  I risultati ottenuti sono deludenti.

3.1.4 Tools di scansione dei DNS

Il test è effettuato sul dominio di Unicam: `unicam.it` per ricercare tutti i record DNS relativi.

- *DnsRecon*:
  target `unicam.it`,
  comando `dnsrecon.py -a -s -g -k -w -z`.
  Ha trovato correttamente il *SoA* - *camcic.unicam.it*, e due *NS* - *camcic.unicam.it* e *dns.cineca.it*.
  Ha identificato le versioni dei sistemi operativi presenti nei due NS: per *camcic.unicam.it RHEL6* e per *dns.cineca.it Debian 8*.
  Sono stati enumerati 5 mailserver di Google poichè l'Università di appoggia al servizio Gmail per la posta universitaria. Di tutti i record trovati è stato fornito anche un IPv4 e un IPv6 (se disponibile).

- *Fierce*:
  target `unicam.it`,
  comando `fierce.py --connect --wide`.
  Ha trovato correttamente il *SoA* - *camcic.unicam.it*, di cui fornisce l'IPv4, e due *NS* - *camcic.unicam.it* e *dns.cineca.it*, senza i rispettivi IP. Ha trovato diversi nomi di dominio interni all'Università di Camerino.
  Ha evidenziato alcuni siti in base al dominio come *blog.unicam.it*, *cms.unicam.it*, *career.unicam.it* e altri.
  Ha trovato l'indirizzo di un Proxy, *193.204.8.28*.

- *DnsEnum*:
  target `unicam.it`,
  comando `dnsenum.pl -p10 -s100 --threads6 -r -w`.
  Ha trovato i record *camcic.unicam.it*, *dns.cineca.it* e i mailserver di Google senza però identificare di che tipo sono.

- *TheHarvester*:
  target `unicam.it`,
  comando `theHarvester.py -b bing -l 100 -v -n -t`.
  Ha trovato alcuni hosts fornendo nome e IP e ha provato di identificare i sever, alcuni sono DNS, altri applicazioni web e virtual hosts.

I moduli DnsRecon e Fierce svolgono una scannerizzazione abbastanza completa e profonda. DnsEnum e TheHarvester non producono report soddisfacenti.

3.1.5 Tools di Information Gathering

Il test prevede la ricerca del maggior numero di informazioni possibile (persone, indirizzi e-mail, sottodomini) relative al dominio Unicam.

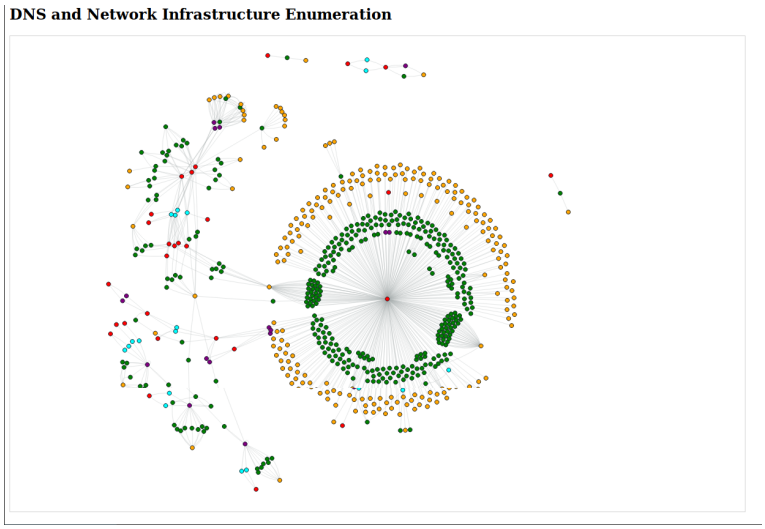


Figura 3.1: Report visuale di Amass.

- *Amass*:
  target `unicam.it`,
  comando `amass -ip -noalts -brute -whois -v`.
  Ha prodotto la mappa in figura *3.1* incorporata in un file HTML (generato di default in **WebMVA**), ma il report testuale è risultato vuoto. È impossibile convertire il report da HTML a PDF data la sua struttura dinamica ottenuta con Javascript;

- *Infoga Email*:
  target `rick90@email.it`, `margherita.renieri@studenti.unicam.it`, `fausto.marcantoni@unicam.it`, `barbara.re@unicam.it`,
  comando `infoga.py -v 2 -b`.
  Pur avendo inserito tra i target email vulnerabili, come quello citato nella figura *1.13*, non ha prodotto alcun risultato.

- *Infoga*:
  target `unicam.it`,
  comando `infoga.py -v 2`.
  Ha trovato 12 indirizzi email personali di alcuni docenti Unicam.

- *Sublist3r*:
  target `unicam.it`,
  comando `sublist3r.py -v -b -t 8`.
  Ha trovato 327 sottodomini, dopo circa 15 minuti di scansione.

Questa sezione del software è la più critica. I tool trovati non risultano soddisfacenti in nessun caso, fatta eccezione per Sublist3r che ha però il solo scopo di trovare sottodomini. La pagina generata da Amass è molto bella esteticamente ma poco usabile senza dati. Infoga è il tool più fallimentare.

3.1.6 Tools di scansione dei Database

- *SqlMap*:
  target `http://testphp.vulnweb.com`[1],
  comando `sqlmap.py --batch --technique=BEUSTQ`.
  La scansione, molto veloce, ha riconosciuto la possibile versione del DBMS, *maggiore della 5.0*, la versione di PHP, *5.3.10*, il server su cui si appoggia, *Nginx* e il parametro vulnerabile, *cat*.

Per effettuare scansione approfondita delle vulnerabilità di un database, è necessario conoscere l'architettura del DBMS e dell'applicazione associata.
Non è stato possibile testare i software Odat e NoSqlMap perché nell'ambiente di testing non sono disponibili tecnologie compatibili.

3.1.7 Tools di scansione dei WiFi

È stato creato un *hotspot* con una password molto debole, *password1234* ed è stato avviato dal WebMVA, Wifite ed è stato effettuato un accesso alla Wifi.
Il tool ha riscontrato un *handshake* senza restituire ulteriori informazioni. Il file dell'handshake catturato è inserito nella cartella dei report per possibili approfondimenti.

4. Evoluzione

L'applicazione è stata migliorata e ampliata in termini di efficienza e specificità.
È stato effettuato uno studio sulle tecnologie usabili per visualizzare il terminale nella pagina delle scansioni.
È stata migliorata la pagina di creazione di un Progetto.
È stata aggiunta la possibilità di inserire qualunque interfaccia web esterna, effettuati diversi controlli sui campi "Nome" delle sezioni Moduli e Progetti e sui percorsi dei file.
È stata inserita la possibilità di decidere l'ordine di salvataggio dei Report.
Sono stati aggiunti Moduli di differenti modalità di scansione per rendere l'applicazione più funzionale.

4.1 Gestione desktop remoto

Uno dei problemi relativi alla collaborazione nell'uso di **WebMVA** è legato alla visualizzazione in tempo reale delle scansioni in corso: alcuni tool sono interattivi (Wifite e NoSQLMap) e altri richiedono l'intervento dell'utente in alcuni casi (dnsrecon, WPScan). L'implementazione iniziale ha previsto la visualizzazione di un terminale *XTERM* con i programmi in esecuzione. Questo limita la collaborazione perché il terminale viene aperto solo sul PC che esegue **WebMVA** ed eventualmente un collaboratore collegato tramite un altro PC non può vedere cosa sta succedendo e se c'è bisogno del suo intervento.

4.1.1 SPICE

SPICE è un protocollo per avere un desktop remoto, simile a **VNC**, testato in ambito virtuale tramite un hypervisor Proxmox. La visualizzazione e la velocità sono risultati più che soddisfacenti, quindi è stato studiato il protocollo per capire se potesse essere integrato nell'architettura che esegue **WebMVA**.
Purtroppo non è stato possibile: **SPICE**, a livello server, ha bisogno di una scheda grafica virtuale di QEMU, ma l'applicazione non è stata pensata per essere lanciata in ambito virtuale perché sorgerebbero alcune limitazioni per la connettività e le scansioni WiFi.

4.1.2 SSH.NET e SignalR

SSH è un protocollo per stabilire una sessione remota cifrata e sicura. Può anche essere usato in ambito locale: basta collegarsi all'indirizzo `127.0.0.1` per aprire una sessione SSH nello stesso PC su cui è stato lanciato il comando.
SSH.NET[1] è una libreria, scritta originariamente per il .NET Framework ma portata anche su .NET Core, che permette di creare e gestire sessioni SSH da codice in maniera semplice ed efficace.
La libreria fornisce tutti gli stream necessari per leggere e scrivere nella sessione creata. Funziona perfettamente se si reindirizza tutto l'output di un comando una volta che l'esecuzione è terminata, ma nell'applicazione è necessario visualizzare cosa accade *in tempo reale*. È stata provata a questo scopo la libreria **SignalR Core**: scritta e mantenuta da Microsoft, permette di avere una connessione **WebSocket** costantemente aperta tra una particolare pagina fra quelle presenti e il server. Tramite un *Hub* di SignalR è possibile scambiare messaggi in tempo reale tra il server e uno o più client collegati alla pagina, eventualmente raggruppando questi ultimi o selezionandoli in base alle necessità.
Purtroppo SignalR ha una limitazione che per l'ambito di **WebMVA** è risultata critica: un'istanza dell'Hub di SignalR lato server viene creata per inviare o ricevere un messaggio e subito dopo eliminata, per ottimizzare l'uso della memoria. È dunque impossibile associare una classe complessa come quella di SSH.NET ad un Hub SignalR, perché la prima ha bisogno di rimanere in memoria fin tanto che la connessione remota non è terminata; è necessario inoltre associare un'istanza separata del client SSH per ogni client collegato.

4.1.3 Shell In A Box

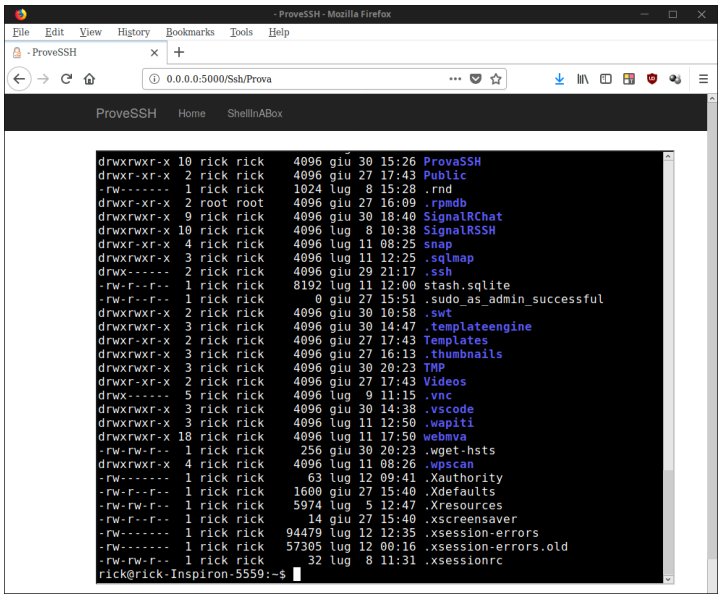


Figura 4.1: Shell In A Box in funzione su un server di prova.

Delle ricerche in rete hanno portato alla scoperta di un tool già esistente, **Shell In A Box**[2], che permette di creare un *demone* in grado di aprire un server Apache e reindirizzare l'output di una shell, creata ad ogni connessione, direttamente nel browser.

È possibile impostare questo demone per far eseguire un comando o uno script appositamente configurati.
In questo caso, pur sembrando promettente ed usabile, la sua implementazione nell'applicazione è stata lasciata in secondo piano dato che richiederebbe una ridefinizione abbastanza importante della logica di esecuzione degli scan.

4.2 Copia dei moduli da un altro Progetto

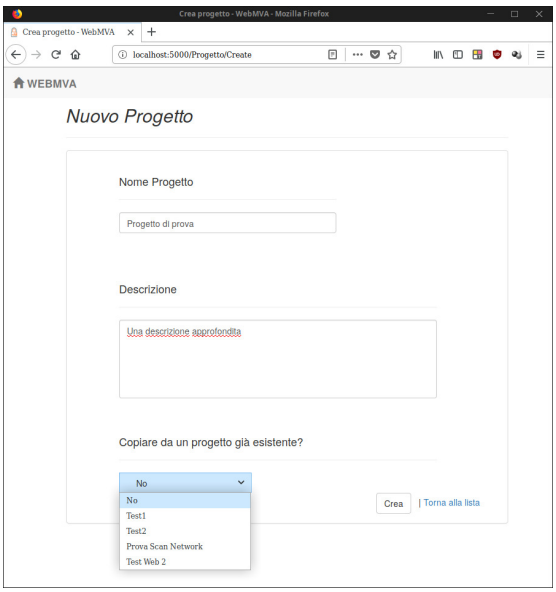


Figura 4.2: Screen funzionalità inserita nella pagina di creazione del nuovo Progetto.

Nella pagina di creazione di un nuovo Progetto è stata inserita una tendina, popolata con quelli già presenti nel sistema, per permettere all'utente di selezionarne uno da cui copiare la lista dei moduli inseriti. Questa funzionalità permette di facilitare l'organizzazione del lavoro in caso di scannerizzazioni molto simili tra loro ma appartenenti a diversi ambiti.

4.3 Modulo Server esterno

È stato rinominato il modulo di *Nessus* in **ModuloSERVER** dato che Nessus non viene gestito. In questo modo, è possibile aggiungere altre applicazioni con interfacce web. Ai campi già presenti è stato aggiunto un campo boolean, per specificare se il server risponde a richeste HTTPS.
Il modulo di OpenVAS non è stato modificato, poiché esistono dei metodi per interfacciarsi con esso.

4.4 Controlli Apportati

4.4.1 Controllo dei percorsi

Alcune delle applicazioni all'interno di **WebMVA** hanno bisogno di percorsi assoluti nei comandi (OpenDoor, wkhtmltopdf tra gli altri). Questi percorsi potrebbero contenere spazi che, se non vengono gestiti, fanno sì che l'interprete dei comandi veda un parametro aggiuntivo invece che la continuazione di un percorso.
La soluzione adottata sostituisce ogni spazio con "\ " nelle stringhe lato server, dove necessario, perché racchiudere tutto tra virgolette era impraticabile per il modo in cui sono gestiti i comandi: già vengono usate sia le virgolette che gli apici singoli.

4.4.2 Controllo del campo "Nome"

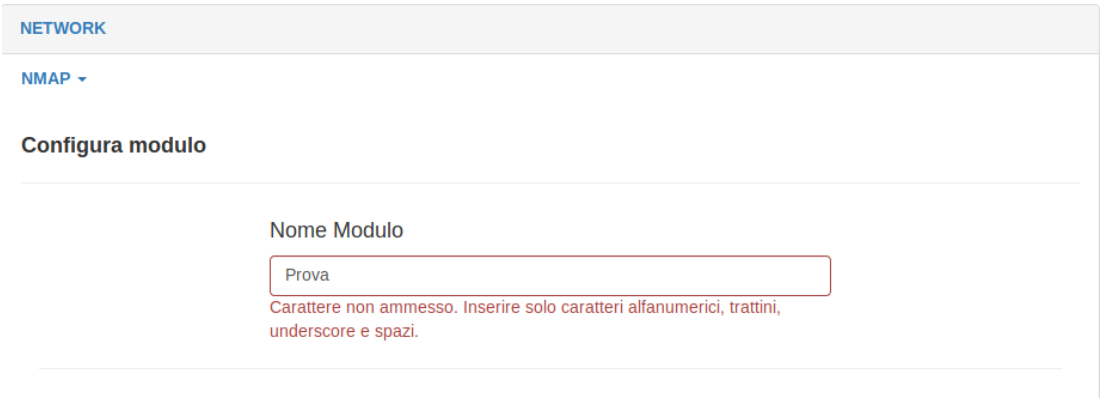


Figura 4.3: Screen compilazione campo "Nome".

I campi "Nome" sono soggetti al controllo dei caratteri inseriti. Con funzioni Javascript, sono stati definiti i caratteri ammissibili ed un feedback visivo immediato in caso di errore. La modifica è stata apportata perché il file system non permette l'uso dei caratteri speciali che sono stati esclusi per la creazione di file e cartelle.

4.5 Ordinamento dei Report

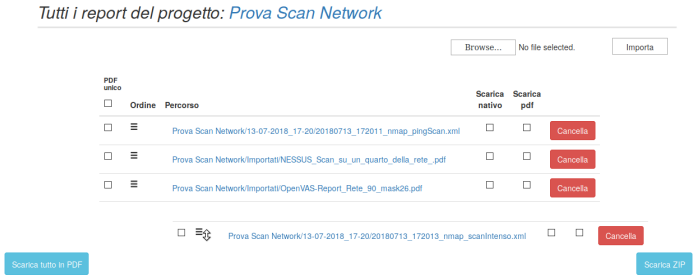


Figura 4.4: Screen ordinamento file.

Nella sezione Report, è possibile scaricare i resoconti della scansione in un *ordine scelto* dalla necessità dell'*utente*, per rendere lo strumento ancora più usabile.

4.6 Inserimento Moduli

I Report ottenuti da *Infoga*, ricavati dopo una prima serie di test, sono poco specifici qualora il *target* sia un *indirizzo e-mail*.
per potenziare la categoria *Information gathering*, sono stati inseriti:

- **TheHarvester**
- **Amass**

Nella sezione della scansione *CMS*, è stato introdotto il software **Drupwn** perché i report prodotti dallo strumento Droopescan sono risultati poco dettagliati.
Tra i Moduli *DNS*, è stato inserito **Whois**, software integrato in Linux.

4.6.0.1 Whois

Whois è un protocollo di rete, descritto nell'RFC 3912, che permette di eseguire un particolare tipo di query per ottenere i dettagli di un dominio registrato su Internet, tra cui:

- il *registrar*, ovvero l'entità che ha effettivamente registrato il dominio;
- la data di scadenza della registrazione;
- l'entità che ha richiesto la registrazione;
- eventuali contatti per abusi e assistenza.

Un client whois chiede queste informazioni a un server in grado di rispondere a queste richieste sulla porta 43. Ne esistono di vari tipi, come client web[3] o librerie per linguaggi di programmazione.
Il client inserito nell'applicazione è l'eseguibile **whois**, un'interfaccia a riga di comando installabile nei sistemi Linux.
Sito web: `https://github.com/rfc1036/whois` e per la documentazione `https://www.mankier.com/1/whois.md`.

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

- `-h` : per identificare il server a cui chiedere informazioni;
- `-p` : per definire quale porta usare per le richieste;
- `--verbose` : per aumentare i dettagli delle informazioni ottenute.
- `-H` : per togliere eventuali disclaimer legali.
- `-a` : per ricercare anche in eventuali mirror del server.

4.6.0.2 TheHarvester

Figura 4.5: Logo di TheHarvester.

TheHarvester è un software scritto in Python, distribuito sotto licenza GNU GPL v2 e raccoglie account di posta elettronica, nomi di sottodomini, host virtuali, porte aperte, banner da diverse fonti pubbliche (motori di ricerca, key server PGP).
Il tool ottiene informazioni su un determinato target e della sua presenza su Internet.
Sito web: `https://tools.kali.org/information-gathering/theharvester`.

Funzionalità implementate

Nell'applicazione sono stare implementate le seguenti funzionalità:

`-b` : permette la scelta del motore di ricerca dove effettuare la scansione.
è possibile scegliere tra:

- google
- bing (microsoft search engine)
- bingapi (motore di ricerca Microsoft, tramite l'API)
- pgp (server chiave pgp)
- linkedin (motore di ricerca di google, genera una ricerca spacifica per gli utenti di Linkedin)
- goolgle-profiles (motore di ricerca di google che effettua una specifica ricerca per profili Google)
- people123
- jigsaw;

`-v` : verifica il nome dell'host tramite la risoluzione DNS e cerca gli host virtuali;

`-n` : esegue una query reverse DNS su tutti gli intervalli rilevati;

`-c` : effettua ua ricerca DNS brute force del nome di dominio inserito in input;

`-t` : permette il rilevamento dell'espansione dei DNS TLD;

`-l` : limita il numero di risultati d'indagine.

4.6.0.3 Amass

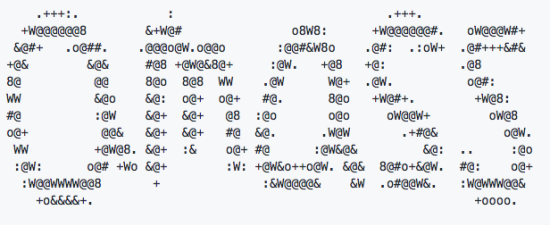


Figura 4.6: Logo di Amass.

Amass, scritto in Go, distribuito sotto licenza Apache versione 2, ottiene i nomi dei sottodomini, scansionando gli archivi web, alterando i nomi, abilitando e/o disabilitando diverse modalità di ricognizione.
Amass utilizza gli indirizzi IP ottenuti durante la risoluzione per scoprire i netblock. Tutte le informazioni vengono poi utilizzate per costruire mappe delle reti di destinazione.
Sito web: `https://github.com/caffix/amass`.

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità:

`-whois` : ricerca utilizzando informazioni whois

`-nodns` : effettua una modalità di scansione passiva;

`-ip` : stampa gli indirizzi IP con i nomi scoperti;

`-bl` : genera una blacklist dei sottodomini indesiderati;

`-blf` :identifica i sottodomini della blacklist provenienti da un file;

`-active` : abilita i metodi di ricognizione attivi;

`-brute` : esegue enumerazioni brute force dei sottodomini;

`-noalts` : disabilita le alterazioni dei nomi scoperti;

`-v` : aumenta la verbosità;

`-visjs` : permette la creazione di grafi relativi ai risultati effettuati utilizzando la libreria VisJS (opzione sempre abilitata);

`-freq` : limita la velocità delle query DNS per numero al minuto.

4.6.0.4 Drupwn

Drupwn scritto in Python (versione 3), open-source, distribuito sotto licenza GNU GPL v.3, è un tool di enumeration ed exploitation per analizzare, cercare e sfruttare le falle di sicurezza all'interno della piattaforma di blogging Drupal.
Una volta eseguito il programma, è possibilie eseguire due differenti funzioni:

- il comando `enum`, esegue diverse modalità di enumerazioni. tra cui quelle degli utenti, dei moduli, dei temi, dei cookies
- il comando `exploit` verifica le vulnerabilità e le sfutta.

Sito web: `https://github.com/immunIT/drupwn`.

Funzionalità implementate

Nell'applicazione sono state implementate le seguenti funzionalità

- `enum`:

  `--users`: esegue l'enumerazioni degli utenti;

  `--nodes`: realizza l'enumerazioni dei nodi;

  `--modules`: compie l'enumerazioni degli dei moduli;

  `--dfiles`: effettua l'enumerazioni dei file;

  `--themes`: svolge l'enumerazioni dei temi;

  `--theads`: permette di settare il numero dei threads per la scansione;

  `--ua`: ricerca un determinato user agent.

5. Sviluppi futuri

5.1 Refactor del codice

Alcune parti del codice sorgente di **WebMVA** potrebbero essere riscritte e/o strutturate in maniera diversa, per esempio:

- lancio degli scan in ProgettoController: il codice in questione è troppo confuso e poco ordinato, andrebbe migliorato;
- organizzazione delle Views: le pagine sono state scritte in maniera piuttosto rozza e statica. Si potrebbe implementare l'uso delle *partial views* in modo da riutilizzare parti di codice comuni, come le sidebar e le form dei moduli;
- struttura dei moduli: in alternativa all'attuale struttura statica data ai Moduli, si potrebbe implementare un comportamento più dinamico tramite la definizione di un'interfaccia "SwitchComando". Con questa strategia un Modulo conterrebbe la sua tipologia (network, web, DNS...), l'eseguibile da lanciare e una lista di "SwitchComando". Sarebbe così possibile gestire anche nuovi software direttamente dall'interfaccia web;
- gestione dipendenze: gli script di installazione e check delle dipendenze eseguono ogni volta *tutti* i controlli. È possibile riscrivere gli script in modo che vengano effettuate solo le azioni effettivamente necessarie.

5.2 Rimozione e rimodellazione dei moduli

I moduli inseriti dovrebbero essere scremati ed i i meno efficienti, in base ai risultati ottenuti dalle analisi mostrate nel Capitolo 5, eliminati.
Ai moduli produttivi andrebbero aggiunte ulteriori funzionalità.

OpenVAS

OpenVAS è gestito come un `ModuloServer` preimpostato per l'uso di HTTPS. Gli scan e i report OpenVAS possono essere automatizzati, perciò il modulo non è stato accorpato in `ModuloServer`, in previsione di una futura gestione tramite **OpenVAS CLI**[1].

5.3 Integrazione Shell In A Box

Come espresso nella sezione *4.1.3*, integrare *Shell In A Box* nel workflow gestito da **WebMVA** renderebbe l'applicazione più collaborativa e funzionale.
Al lancio dei moduli di un progetto si dovrebbe quindi:

- creare un file bash con tutti i comandi degli scan da effettuare;
- lanciare `shellinabox` su una porta particolare (diversa ogni volta che viene eseguito un Progetto) per fargli eseguire questo script quando ci si connette e tenere traccia del suo **PID**, *Process Identificator*;
- chiudere il demone tramite il PID ed eliminare il file dello script quando l'utente esce dalla pagina.

5.4 Ottimizzazione gestione file

File insesistenti

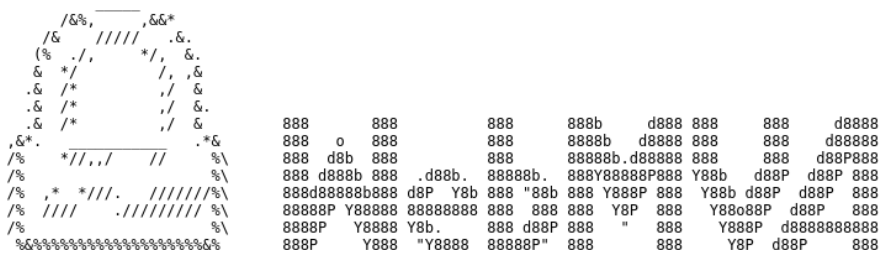
I software possono fallire inaspettatamente o venire interrotti dall'utente prima che abbiano finito la scansione. Nel database dell'applicazione, in questi casi, possono trovarsi dei riferimenti a file non esistenti se l'applicazione non aveva ancora creato il suo report.
Allo stato attuale l'applicazione è quindi troppo "ottimista" e si aspetta sempre che vada tutto bene. Un possibile miglioramento sarebbe la gestione dei record dei Percorsi per inserire nel database solo i riferimenti ai file effettivamente esistenti.

PDF

La generazione dei file PDF prevede una conversione dei file generati dalle scansioni eseguita nella loro stessa cartella, creando così un nuovo file.
I percorsi di questi nuovi documenti non vengono gestiti nel database e quando si vuole eliminare un Report non vengono cancellati.
È possibile integrare questi file nei Percorsi di un Report e gestirli in modo più efficace, evitando anche la creazione di file PDF già esistenti.

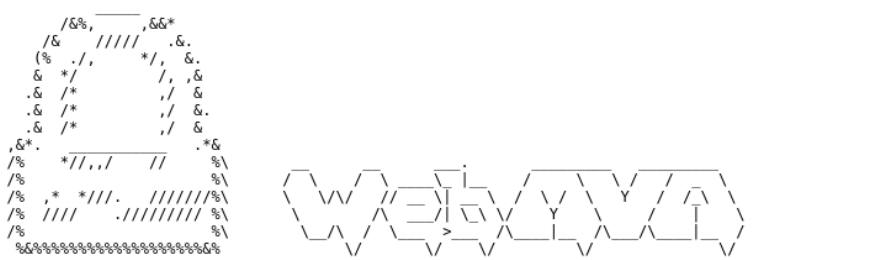
6. Conclusioni

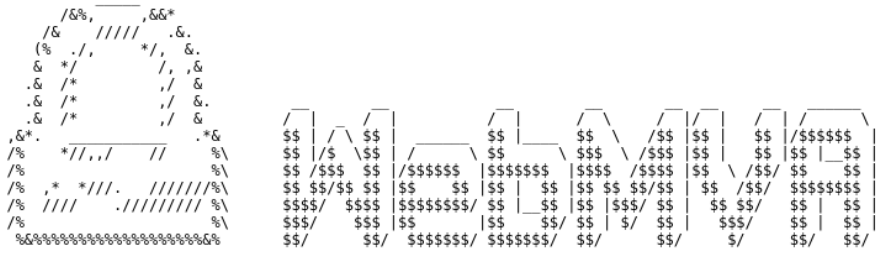
La ricerca effettuata ha portato alla stesura della tesi "***WebMVA - Collaborative application for Vulnerability Assessment Analysis***", conclusione del percorso triennale universitario.
L'argomento e il suo ambito, proposto del Relatore, Professore Fausto Marcantoni, ha portato alla realizzazione dell'applicazione, *WebMVA*, per la ricerca della vulnerabilità attraverso diverse tipologie di scansione.
La tesi è la relazione finale del percorso di *Stage++* iniziato il secondo semestre del terzo anno accademico. Esso è costituito da tre tappe: *Stage*, svolto presso il laboratorio Cisco presso il corso di laurea di Unicam, *Project*, esposizione dell'elaborato sviluppato e *stesura finale della Tesi*.
L'argomento studiato è attuale, impegnativo e particolarmente sentito in ogni ambito collettivo, vista la diffusione delle nuove tecnologie nella società contemporanea.
Il progetto ha permesso di approfondire problematiche legate a diversi aspetti del vulnerability assessment e la necessità di trovare strumenti affidabili i cui report siano dettagliati con opzioni di collaborazione. Il lavoro eseguito ha permesso un approccio alla programmazione di uno strumento particolare e dettagliato, di cui non si trovano molte alternative.
L'idea di contribuire, in piccola parte con l'utilizzo del tool implementato e con le competenze acquisite, alla messa in sicurezza degli apparati informatici di una società superficialmente informatizzata, ha motivato i due laureandi.
Le difficoltà incontrate sono state molte, grazie alla supervisione del Relatore sono state superate e hanno portato a maturare la capacità critica e professionale del team di lavoro di fronte a dubbi implementativi e ad ampliare le competenze professionali: dalla realizzazione e le relative scelte delle tecnologie sopportanti l'applicazione, all'assimilazione di nozioni tecniche nell'ambito della sicurezza informatica e alla scoperta di un mondo altamente tecnologico ma estremamente vulnerabile.
La materia presa in esame è molto versatile ed in continua trasformazione, tanto che, durante il periodo di Stage++, più volte si sono dovute apportare modifiche per rendere l'applicazione sempre più aggiornata.
Oltre alla mera indagine tecnica che ha prodotto WebMVA, il team di ricerca ha preso coscienza che dietro questo lavoro apparentemente tecnico, si nasconde una sfida etica per rendere sicuri gli apparati elettronico-digitali che non sono solo un semplice accessorio dell'uomo tecnologico del terzo millennio, ma hanno una valenza ben più profonda racchiudendo tutti i propri dati sensibili.
È evidente come l'evoluzione tecnologica corrisponda ad un'insicurezza dei sistemi informatici utilizzati ma "*...solo facendosi delle domande*" - afferma Marco Raimilli di Yoroi - "*è possibile capire quali sono gli scopi di un attacco e come debellarlo completamente. Per questo l'essere umano sarà sempre l'elemento indispensabile della sicurezza...*[1]".

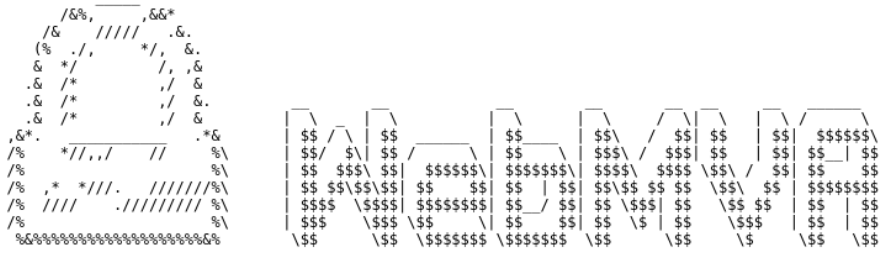

888 888 888 888b d888 888 888 d8888
888 o 888 888 888 8888b d8888 888 888 d88888
888 d8b 888 888 88888b.d88888 888 888 d88P888
888 d888b 888 .d88b. 88888b. 888Y88888P888 Y88b d88P d88P 888
888d88888b888 d8P Y8b 888 "88b 888 Y888P 888 Y88b d88P d88P 888
88888P Y88888 88888888 888 888 888 Y8P 888 Y88o88P d88P 888
8888P Y8888 Y8b. 888 d88P 888 " 888 Y888P d8888888888
888P Y888 "Y8888 88888P" 888 888 Y8P d88P 888

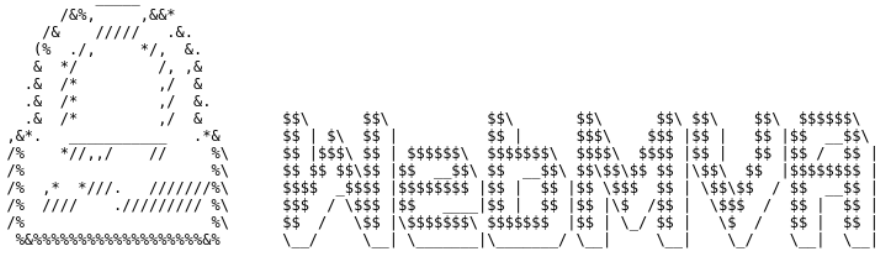

WEBMVA

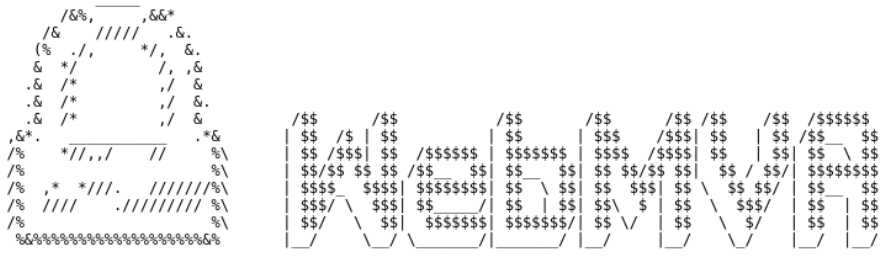

WEBMVA











7. Ringraziamenti

Un ringraziamento speciale va al professor Fausto Marcantoni, Relatore della nostra Tesi di Laurea.
"Magister et Mentor", ha guidato la nostra ricerca con grande professionalità ed autorevolezza. Disponibile oltre misura, è Professore di grande umanità: ha condiviso le sue competenze per accrescere le nostre. Ci ha spronato nel realizzare al meglio la ricerca, ponendoci obiettivi ogni volta più alti ed incoraggiandoci nel superare le difficoltà incontrate. Con la sua costante e determinate presenza, il professor Marcantoni ci ha insegnato, soprattutto, che " *...l'unico modo per fare un buon lavoro, è amare quello che fate...*".

Appendice A - Sotto-tabelle

	Facilità di installazione	Licenza	Modalità di servizio	Livelli di documentazione	Database aggiornati	Difficoltà di utilizzo	Usabilità	Tempo di elaborazione dei dati	Chiarezza dei risultati	Esportabilità dei documenti	Portabilità	Personalizzazione di scansione
Nmap	8	GPL2	free	8	2018	2	10	10	10	10	10	8
Nessus	7	GPL	free + comm.	8	2018	2	10	9	10	10	10	8
OpenVAS	7	LGPL3	free	7	2017	3	10	9	10	10	10	8

Figura 1: Sotto-tabella Network.

	Facilità di installazione	Licenza	Modalità di servizio	Livelli di documentazione	Database aggiornati	Difficoltà di utilizzo	Usabilità	Tempo di elaborazione dei dati	Chiarezza dei risultati	Esportabilità dei documenti	Portabilità	Personalizzazione di scansione
Wapiti	7	GPL2	free	7	2018	3	8	7	8	8	10	7
OpenDoor	6	GPL3	free	7	2017	4	6	7	7	8	8	7
WAScan	8	GPL3	free	6	2018	2	8	8	6	-	10	7
Vega	7	proprietaria	free + comm.	7	2014	2	10	0	8	8	10	4
netsparker	7	proprietaria	commerciale	7	-	3	7	-	8	8	10	8
W3af	7	copyright	free	8	2018	2	10	7	8	10	10	8

Figura 2: Sotto-tabella Web.

	Facilità di installazione	Licenza	Modalità di servizio	Livelli di documentazione	Database aggiornati	Difficoltà di utilizzo	Usabilità	Tempo di elaborazione dei dati	Chiarezza dei risultati	Esportabilità dei documenti	Portabilità	Personalizzazione di scansione
CMSScanner	5	MIT	free	5	2018	4	4	7	-	-	7	4
CMSmap	7	GPL2	free	7	2015	2	5	7	8	8	10	7
JoomScan	7	GPL3	free	7	2018	1	5	7	6	6	6	5
WPScan	7	Public Source	free	7	2018	4	6	8	7	7	6	6
Sharepoint	8	proprietaria	free	8	2018	5	7	7	7	5	6	10
Droopescan	8	Affero GPL3	free	8	2018	4	7	7	7	7	6	6

Figura 3: Sotto-tabella CMS.

	Facilità di installazione	Licenza	Modalità di servizio	Livelli di documentazione	Database aggiornati	Difficoltà di utilizzo	Usabilità	Tempo di elaborazione dei dati	Chiarezza dei risultati	Esportabilità dei documenti	Portabilità	Personalizzazione di scansione
DnsEnum	7	GPL2	free	7	2017	6	6	10	8	-	-	6
DnsRecon	7	proprietaria	free	10	2018	3	8	10	8	8	10	8
Dnsmap	6	-	free	4	2015	4	6	7	7	8	6	6

Figura 4: Sotto-tabella DNS.

	Facilità di installazione	Licenza	Modalità di servizio	Livelli di documentazione	Database aggiornati	Difficoltà di utilizzo	Usabilità	Tempo di elaborazione dei dati	Chiarezza dei risultati	Esportabilità dei documenti	Portabilità	Personalizzazione di scansione
Infoga	7	GPL3	free	7	2018	3	9	9	7	7	10	8
Maltego	7	proprietaria	free (+ p)	3	2018	6	7	8	8	5	10	9
Sublist3r	7	GPL2	free	8	2017	4	8	9	8	8	10	8

Figura 5: Sotto-tabella Information Gathering.

	Facilità di installazione	Licenza	Modalità di servizio	Livelli di documentazione	Database aggiornati	Difficoltà di utilizzo	Usabilità	Tempo di elaborazione dei dati	Chiarezza dei risultati	Esportabilità dei documenti	Portabilità	Personalizzazione di scansione
Sqlmap	6	GPL2	free	7	2018	6	7	7	7	7	6	7
Arachni	7	LLC	try (+ p)	8	2017	3	8	8	7	6	7	7
Shadow Db S	6	proprietaria	free	7	2016	5	7	7	8	7	8	7
NoSqlmap	7	Copyright	free	6	2018	6	6	7	7	6	7	6
Odat	5	GPL3	free	7	2018	7	7	7	6	6	7	6

Figura 6: Sotto-tabella Database.

	Facilità di installazione	Licenza	Modalità di servizio	Livelli di documentazione	Database aggiornati	Difficoltà di utilizzo	Usabilità	Tempo di elaborazione dei dati	Chiarezza dei risultati	Esportabilità dei documenti	Portabilità	Personalizzazione di scansione
Wifite	7	GPL2	free	8	2018	3	8	6	8	-	4	8
FruityWifi	5	GPL3	free	8	2018	6	6	5	6	5	4	8
Fern	6	GPL2	free	7	2017	3	8	5	6	-	4	8

Figura 7: Sotto-tabella Wifi.

Appendice B - Class Diagram

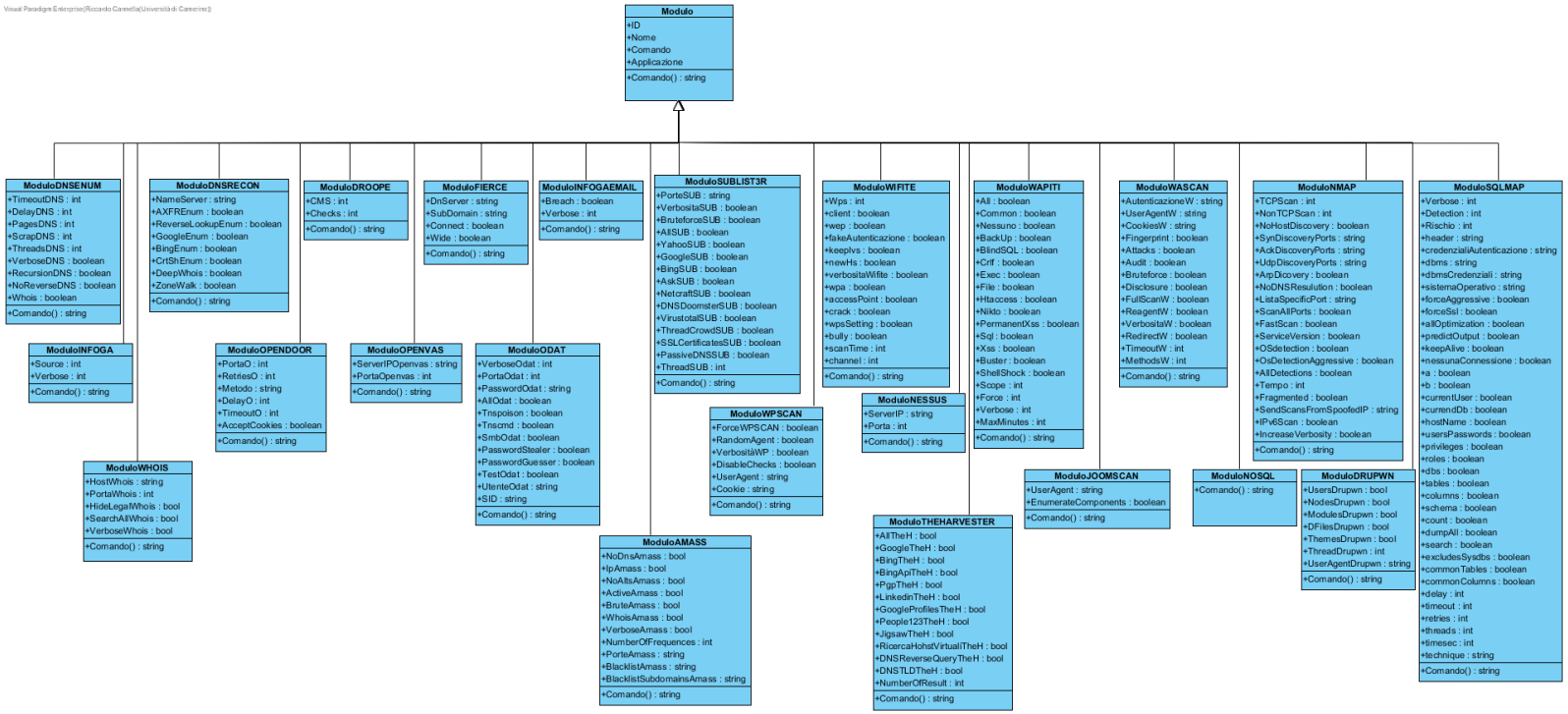


Figura 8: Generalizzazione classe Modulo.

Note

Capitolo 1

1. (Pag. 8) Definizione dell'ENISA, *Agenzia dell'Unione europea per la sicurezza delle reti e dell'informazione*, riportate nel Glossario della gestione dei rischi.

2. (Pag. 10) `https://www.owasp.org/images/c/c9/OWASP_Top_10_-_2013_-_Italiano.pdf`

3. (Pag. 10) Documento, Settembre del 2008, realizzato da K. Scarfone, M. Souppaya, A. Cody, A. Orebaugh, che guida le organizzazioni nella *pianificazione* e nella *conduzione dei test* per la verifica della sicurezza delle informazioni tecniche, analizzando i risultati e sviluppando nuove strategie. Propone degli advices relativi alla progettazione, l'implementazione e il mantenimento delle procedure di sicurezza, `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf`

4. (Pag. 10) `https://www.nist.gov/`

5. (Pag. 10) `http://www.isecom.org/mirror/OSSTMM.3.pdf`

6. (Pag. 10) `http://www.isecom.org/home.html`

7. (Pag. 11) `https://www.owasp.org/index.php/Main_Page`

8. (Pag. 11) `https://www.owasp.org/images/1/19/OTGv4.pdf`

9. (Pag. 11) `http://www.pentest-standard.org/index.php/Main_Page`

10. (Pag. 14) `http://seclists.org/bugtraq/`

11. (Pag. 14) `http://seclists.org/fulldisclosure/`

12. (Pag. 14) `https://www.us-cert.gov/ncas`

13. (Pag. 14) `https://www.us-cert.gov/ncas`

14. (Pag. 17) `https://www.garanteprivacy.it/regolamentoue`

15. (Pag. 18) MarketsandMarkets 2017

16. (Pag. 18) Prov. del 27 novembre 2008, Misure e accorgimenti prescritti ai titolari dei trattamenti effettuati con strumenti elettronici relativamente alle attribuzioni delle funzioni di amministratore del sistema-domanda 10:
"Qualora il sistema di log adottato generi una raccolta dati più ampia, comunque non in contrasto con le disposizioni del Codice e con i principi della protezione dei dati personali, il requisito del provvedimento è certamente soddisfatto. Comunque è sempre possibile effettuare un'estrazione o un filtraggio dei logfiles al fine di selezionare i soli dati pertinenti agli AdS."

17. (Pag. 21) In un'architettura stateless, tutti i pacchetti che provengono dal Web server esterno passano il firewall.

18. (Pag. 24) `https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html`

19. (Pag. 27) Nome di dominio non ambiguo che specifica la posizione assoluta di un

nodo all'interno della gerarchia dell'albero DNS.

Capitolo 2

1. (Pag. 36) Nell'*Appendice A - Sotto-tabelle* sono presenti le sette sotto-tabelle valutative realizzate durante il periodo di Stage.
2. (Pag. 42) Nell'*Appendice B - Class Diagram* è presente il class diagram generalizzazione del Modulo.
3. (Pag. 45) `https://www.tenable.com/products/nessus/nessus-answers`
4. (Pag. 56) `https://www.paterva.com/web7/`
5. (Pag. 56) Maltego non è inserita tra gli strumenti usati dall'applicazione perchè si tratta di uno tool collaborativo.
6. (Pag. 57) Nell'applicazione si trovano due sezioni nel modulo di Infoga per distinguere i due possibili target del modulo: dominio ed email.

Capitolo 3

1. (Pag. 70) `http://www.vulnweb.com/`

Capitolo 4

1. (Pag. 72) `https://github.com/sshnet/SSH.NET`
2. (Pag. 72) `https://code.google.com/archive/p/shellinabox/`
3. (Pag. 75) Un esempio di client web fornito da Registro.it: `https://web-whois.nic.it`

Capitolo 5

1. (Pag. 79) Manuale su `http://www.openvas.org/src-doc/openvas-cli/index.html`

Capitolo 6

1. (Pag. 81) "*Sicurezza, i limiti dell'uso dell'intelligenza artificiale*", Giancarlo Calzetta, Nòva24, n.653, 3 Giugno 2018

Elenco delle figure

Bibliografia

[1] James Kurose, Keith W. Ross, *Reti di calcolatori e internet, Un approccio top-down*, VI Edizione, Pearson 2013.

[2] Paolo Atzeni, Stefano Ceri, Piero Fraternali, Stefano Paraboschi, Riccardo Torlone, *Basi di dati (Modelli e linguaggi di interrogazione)*, IV Edizione, McGraw-Hill 2014.

[3] Charles P.Pfleeger, Shari Lawrence Pfleeger *Sicurezza informatica*, I edizione Italiana, Pearson, 2004.

[4] Andrew Lockhart, *Reti e sicurezza (Trucchi e segreti)*, II edizione, Hops, 2007.

[5] Andrew Tanenbaum, David J. Wetherall, *Reti di calcolatori*, V edizione, Pearson, 2011.

[6] William Stallings, *Sicurezza delle reti (Applicazioni e standard)*, III Edizione, Pearson 2007.

Sitografia

[1] Serkan Özkan, *Browse Vulnerabilities By Date*, https://www.cvedetails.com/browse-by-date.php, 2018.

[2] *Open Systems Interconnection*, https://it.wikipedia.org/wiki/Open_Systems_Interconnection, 2018

[3] *Threat (computer)*, https://en.wikipedia.org/wiki/Threat_(computer), 2018

[4] *Information Security*, https://en.wikipedia.org/wiki/Information_security, 2018

[5] *Firma digitale*, https://it.wikipedia.org/wiki/Firma_digitale, 2018

[6] *Sicurezza informatica*, https://it.wikipedia.org/wiki/Sicurezza_informatica, 2018

[7] *Kerberos: The Network Authentication Protocol*, http://web.mit.edu/kerberos/, 2018

[8] *Posta elettronica*, https://it.wikipedia.org/wiki/Posta_elettronica, 2018

[9] *Diffie-Hellman and PGP*, https://www.rootnetsec.com/diffie-hellman-and-pgp/, 2016

[10] Nino Grasso, *Chrome 68, siti HTTP segnalati come Non sicuri*, https://www.hwupgrade.it/news/programmi/chrome-68-siti-http-segnalati-come-non-sicuri_74108.html, 2018

[11] *Transport Layer Security*, https://en.wikipedia.org/wiki/Transport_Layer_Security, 2018

[12] *Penetration Testing – Cos'è un Penetration Test*, https://onofri.org/security/cose-un-penetration-test/, 2017

[13] *Guida all'applicazione del Regolamento Europeo in materia di protezione dei dati personali*, https://www.privacyitalia.eu/wp-content/uploads/2018/03/Guida-al-Gdpr-2018.pdf, 02/2018

[14] *Che cosa sono i file log e perché non c'è sicurezza senza log management*, https://www.zerounoweb.it/techtarget/searchsecurity/che-cosa-sono-i-file-log-e-perche-non-c-e-sicurezza-senza-log-management/, 2017

[15] *Le vulnerabilità dei sistemi informatici informatici*, https://www.roccobalzama.it/le-vulnerabilita-dei-sistemi-informatici-e-della-rete-storia-ed-analisi/, 2018

[16] *OWASP Testing Guide 4.0*, https://www.owasp.org/images/1/19/OTGv4.pdf, 2018