

UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Valutazione tecnologie honeypot open-source ed implementazione in azienda

Relatore

Prof. Migliardi Mauro

Laureando

Bonotto Jacopo

ANNO ACCADEMICO 2024-2025

Data di laurea 22/07/2025

Sommario

In questa tesi si tratta il tema della sicurezza informatica attraverso gli strumenti afferenti alla deception technology. Un esempio di questa tecnologia sono gli honeypot, che sono dei dispositivi di difesa della rete che funzionano tramite esche posizionate in modo strategico. Il loro scopo è quello di attirare un attaccante informatico e, una volta che quest'ultimo ha interagito con le esche create dallo strumento stesso, mandare un avviso di sicurezza per avvertire della sua presenza e registrare ogni comando da lui eseguito. Si sono esposti i pregi e i difetti, e si sono implementati due esempi di honeypot open-source in una rete aziendale : Cowrie e T-pot. Si sono poi testati questi due strumenti e sono state riportate le rilevazioni svolte dai due honeypot. Infine si è trattato il tema della legalità e l'aspetto etico nell'uso di questi strumenti, argomenti delicati poiché non c'è una chiara linea guida sull'argomento, ma è soggettiva poiché non esiste una linea guida e dipende anche dal luogo geografico in cui si implementano queste esche. Si è concluso il discorso affrontando una possibile continuazione di questa tesi : gli honeypot LLM, che consistono in ambienti virtuali privi di server e risorse fisiche, le quali sono sostituite da un Large Language Model , il quale interagisce direttamente con i comandi inseriti dall'attaccante e genera risposte automatiche e istantanee. Queste tipologie di honeypot sono quelle più moderne e garantiscono elevata sicurezza poiché non sono presenti risorse fisiche da sfruttare.

Indice

1	Introduzione alla Cybersecurity	1
1.1	Introduzione	1
1.2	Definizione ed importanza della cybersecurity	2
1.3	Minacce e vulnerabilità comuni nelle reti aziendali	2
1.4	Principi per mettersi in sicurezza	4
1.5	Introduzione alla <i>Deception Technology</i>	6
2	Concetti di Honeypot	9
2.1	Definizione di honeypot e suo funzionamento	9
2.2	Storia ed evoluzione degli honeypot	10
2.3	Classificazione di honeypot	11
2.3.1	Per tecnologia	11
2.3.2	Per interazione	12
2.3.3	Per categoria	13
3	Uso di honeypot in rete aziendale	15
3.1	Scopo e benefici dell'uso di Honeypot in rete aziendale	15
3.1.1	Scopo degli Honeypot	15
3.1.2	Vantaggi degli honeypot	16
3.2	Architettura e design di un sistema honeypot	17
3.3	Strumenti e software open-source per la creazione di honeypot	19
3.3.1	Cowrie	19
3.3.2	T-Pot	20
4	Case study e Analisi dei dati	23
4.1	Esempi di implementazioni reali di honeypot in azienda	23
4.1.1	Installazione di Cowrie	23
4.1.2	Test di Cowrie	26
4.1.3	Installazione di T-Pot	27

4.2	Analisi dei dati raccolti : Dashboard di T-Pot	29
4.3	Valutazione dell'efficacia dell' honeypot per il miglioramento della sicurezza della rete aziendale	32
5	Sfide e Limitazioni	33
5.1	Problemi etici e legali nell'uso di honeypot	33
5.1.1	Problemi etici	33
5.1.2	Problemi legali	33
5.2	Limitazioni tecniche e rischi nell' implementazione di Honeypot	35
5.3	Soluzioni e best practices per superare queste sfide	35
6	Honeypot LLM	37
6.1	Introduzione e tipologie esistenti	37
6.2	Pregi e difetti	38
6.3	Struttura di un honeypot LLM	39
7	Conclusioni e prospettive future	41
	Bibliografia	43

Elenco delle figure

2.1	Posizionamento di honeypot di ricerca e di produzione	14
3.1	Architettura di Cowrie	19
3.2	Architettura di T-Pot	21
4.1	Esempio di accesso di Cowrie	26
4.2	Esempio di comandi eseguiti nell'interfaccia di Cowrie	26
4.3	Esempio di file contenuti in Cowrie	27
4.4	Scelta di installazione di T-Pot	28
4.5	Porte attive dopo l'installazione di T-Pot	28
4.6	Attacchi totali rilevati	29
4.7	Attacchi giornalieri rilevati	30
4.8	Mappa degli attacchi rilevati	30
4.9	Provenienza degli attacchi	31
4.10	Porte maggiormente attaccate	31
4.11	Porte maggiormente attaccate	32
6.1	Struttura di honeypot LLM	39

Elenco delle tabelle

2.1	Confronto tra honeypot ad alta e bassa interazione	12
3.1	Implementazioni possibili di honeypot[20]	17

Capitolo 1

Introduzione alla Cybersecurity

1.1 Introduzione

Al giorno d'oggi la sicurezza informatica è un aspetto fondamentale non solo per i cittadini ma soprattutto per le aziende : come si può leggere dal report *Cost of a data breach* [1] di IBM del 2024, in un gruppo di oltre 600 aziende coinvolte in un *data theft* nell'ultimo anno il costo medio causato da questa violazione informatica è stato pari a \$4.9 milioni, in aumento di circa il 10% dal 2023 e in costante crescita da molti anni. Con questi attacchi subiti, le aziende intervistate hanno perso principalmente dati riservati, hanno subito gravi danni economici e soprattutto danni d'immagine : quest'ultimo porta a una sfiducia da parte del cliente relativamente all'azienda e ai prodotti o servizi che fornisce, e rischia di mandare in bancarotta un'azienda anche affermata.

I 5 settori maggiormente colpiti sono stati, in ordine crescente di perdita pecuniaria, l'energetico, il tecnologico, l'industriale, il finanziario e il sanitario. La mancanza del personale lavorativo ha inficiato negativamente il controllo della sicurezza informatica, causando lacune in certi luoghi della rete produttiva ed esponendo così la rete aziendale ad attacchi esterni. Inoltre, altra causa che ha portato a un aumento di attacchi è stata la creazione di copie di file riservati, i cosiddetti "*shadow data*", cioè copie di file che contengono informazioni riservate e che devono essere tenute sicure. Una volta create queste copie se ne perde il controllo : nel report, il 93% delle aziende ha ammesso di non avere il pieno controllo dei loro dati salvati, e di non sapere se fossero esistite delle copie non ufficiali. Infine, l'ultima causa presente nel report, ma di gran lunga la più grave a livello di perdita di dati, è il phishing e, conseguentemente, il furto d'identità. Infatti dal report si può notare come quasi il 75% delle fughe di dati abbia una di queste due cause.

I dati di maggior allerta contenuti nel report sono il *mean time to identify* (MTTI), cioè il tempo di giorni in cui si rileva un breach informatico, con un valore medio di 200 giorni, ed il *mean time to contain* (MTTC), cioè il tempo medio necessario per contenere un breach nel siste-

ma ed una conseguente fuga di dati, pari a 70 giorni. In totale per rilevare e debellare il sistema da infiltrazioni, passano mediamente 270 giorni : è importante quindi limitare l'avvenimento di breach informatici, usando strumenti e dispositivi di difesa della rete tipici della cybersecurity, ma soprattutto educando il personale riguardo le minacce esistenti e, infine, usare sempre il buon senso.

1.2 Definizione ed importanza della cybersecurity

La cybersecurity[2] si riferisce a qualsiasi tecnologia, pratica e politica volta a prevenire gli attacchi informatici o a ridurne l'impatto. Ha lo scopo di proteggere i sistemi informatici, le applicazioni, i dispositivi, i dati, le risorse finanziarie e le persone da minacce informatiche. A livello aziendale, lo scopo della cybersecurity è la gestione del rischio dell'organizzazione, ricoprendo un ruolo fondamentale perché gli attacchi possono interrompere, danneggiare o distruggere l'operato delle aziende, oltre alla possibilità di furto d'identità, estorsione, perdita di informazioni sensibili, di clienti e di affari.

1.3 Minacce e vulnerabilità comuni nelle reti aziendali

Una minaccia informatica è un tentativo deliberato di ottenere l'accesso al sistema di un singolo individuo o di un'organizzazione. I malintenzionati perfezionano continuamente i loro metodi di attacco per eludere i sistemi di rilevamento e sfruttare nuove vulnerabilità, ma si affidano ad alcune metodologie di attacco comuni [3], dalle quali è possibile prepararsi e difendersi.

Malware[4] :

E' un termine generale che include qualsiasi software dannoso, tra cui *worm*, *ransomware*, *spyware* e *virus*. E' progettato per danneggiare computer o le reti cancellando o alterando file ed estraendo dati sensibili. Può essere installato da un utente malintenzionato che ottiene accesso alla rete, ma spesso viene installato da utenti ignari, scaricando qualche file o programma infetto.

Ransomware[5] :

Il *ransomware* è una forma di estorsione che consiste nell'utilizzo di malware per crittografare dei file, rendendoli inaccessibili al proprietario legittimo, con lo scopo di chiedere un compenso al fine di riconsegnare i file. Le vittime infatti, in cambio di una chiave di decrittografia, devono pagare il riscatto in genere attraverso criptovalute.

Social Engineering[6] :

Nell'ingegneria sociale, gli utenti malintenzionati approfittano della fiducia delle persone per ingannarle, facendole per esempio inserire informazioni private, cliccare link nelle mail o scaricare malware. In questo tipo di attacchi, gli hacker fingono di essere una persona conosciuta, un'azienda famosa o la propria banca, per contattare e creare un senso d'urgenza in modo da lasciare poco tempo per reagire alle vittime.

Phishing[7] :

Il *phishing* è un sottogruppo dell'ingegneria sociale, basato sull'utilizzo di mail o messaggi mandati da fonti attendibili con lo scopo di convincere le persone a cliccare su un collegamento o condividere informazioni private. Esistono due tipi di phishing :

1. Mass phishing

Consiste nel mandare a molte persone lo stesso messaggio o mail, nella speranza che una piccola parte dei ricevitori clicchi o invii le proprie informazioni.

2. Spear phishing

Consiste nel mandare un messaggio costruito appositamente con settimane di preparazione, e mirato ad una persona di solito di grande importanza come dirigenti d'azienda o governatori.

Minacce interne :

In una minaccia interna, le persone che già hanno accesso ad alcuni sistemi in funzione di lavoratori causano una violazione della sicurezza o una perdita finanziaria alla propria azienda. La maggior parte delle volte ciò non è volontario, può succedere per esempio che un dipendente carichi erroneamente le proprie credenziali lavorative in un cloud personale, e che quel cloud fosse vulnerabile. Tuttavia, alcune volte i dipendenti che causano minacce interne agiscono in malafede.

APT[8] :

L' *Advanced Persistent Threat*, o minaccia persistente avanzata, è una situazione in cui i malintenzionati ottengono l'accesso ai sistemi informatici aziendali, ma non si rivelano fino a che non abbiano collezionato molte informazioni e dati privati. Gli autori di queste minacce eseguono una ricerca nei sistemi aziendali da colpire, e rubano dati senza che le contromisure di sicurezza vengano attivate.

1.4 Principi per mettersi in sicurezza

I principi base per la sicurezza informatica possono essere riassunti con l'acronimo CIA[9] : *Confidentiality, Integrity e Availability*.

1. **Confidenzialità** :

La riservatezza delle informazioni non riguarda solo la segretezza del dato, ma anche la possibilità di essere accessibile solo da chi ne ha il diritto. Per tale ragione, gestire la riservatezza delle informazioni significa anche avere la capacità di gestire chi, quando e in che modo ha diritto di accesso ai dati.

2. **Integrità** :

Le azioni volte a preservare l'integrità delle informazioni servono ad evitare che il dato sia alterato, o cancellato indebitamente, cioè senza autorizzazione. Contestualmente, vengono stabilite delle regole che definiscono il diritto di agire sul dato, modificando l'informazione. All'interno del più ampio concetto di integrità rientrano anche quello di autenticità e non ripudiabilità: da un lato, infatti, la modifica non autorizzata comporta anche la non autenticità dell'informazione, dall'altro un dato integro è non ripudiabile, ovvero il proprietario non può negare di averlo prodotto.

3. **Disponibilità** :

Il parametro di *availability* nel campo della cybersecurity governa il modo e le tempistiche con cui i dati sono resi disponibili all'utente che li richiede e che ne ha diritto. Il ripristino di emergenza del dato e la continuità di utilizzo sono due sottocategorie della disponibilità. Tali fattori delineano in che modo il sistema risponde a un qualsiasi evento che causi la perdita dei dati o di operatività per tornare allo stato esistente prima dell'incidente informatico e stabiliscono le tecniche di Data Loss Prevention.

Altri principi di sicurezza degni di nota sono i seguenti[10] :

No single point of failure[11] :

Il termine *single point of failure* indica un'assenza di *backup* o di ridondanza di un determinato dato che non si vuole perdere. Per garantire la conservazione dei dati, è necessario dunque implementare un luogo secondario dove copiare e tenere sicuro il file.

Least privilege^[12] :

Il principio di *least privilege* funziona limitando l'accesso a dati, risorse e applicazioni attraverso un processo di autorizzazione. Ciò permette una maggiore sicurezza al dato, ed offre la possibilità di conoscere chi ha accesso ai dati e quando ne ha usufruito. É uno strumento cardine per limitare i *breach* di dati e gli attacchi informatici, riuscendo anche a ridurre gli attacchi provenienti dall'interno della rete aziendale.

Separation of duties^[13] :

Il principio di separazione dei compiti consiste nel non rilasciare a nessun lavoratore privilegi tali da poter compromettere la rete aziendale. Spesso viene usato la regola *two-person*, che consiste nel dividere i privilegi di rete ad entrambe le persone, senza però che una sola persona abbia potere tale da provocare danni ingenti.

Complexity is not security^[14] :

Secondo questo principio la complessità non porta a sicurezza, ma crea una fragilità. Infatti, utilizzando un sistema complesso, un attaccante avrà più area di attacco. Inoltre, sarà necessaria una maggiore manutenzione, e ciò può portare ad un errore umano, introducendo in questo modo nel sistema vulnerabilità.

Infine, l'ultimo principio da seguire per ottenere un sistema sicuro è il non utilizzare il metodo di *security by obscurity*, cioè non tenere segreto un funzionamento interno di un sistema pensando che ciò garantisca maggiore sicurezza. Al contrario, nascondere un processo interno porterebbe a una maggiore difficoltà nell'accorgersi di un attacco esterno e a un successivo ostacolo nel riparare il danno provocato.

Un' implementazione di questi principi, seguita dall'impiego di sistemi di rilevazione delle intrusioni e di strumenti di *deception technology*, aumenta la robustezza della rete ad attacchi informatici e garantisce una sicurezza dei dati e delle informazioni.

1.5 Introduzione alla *Deception Technology*

Le "tecnologie di inganno" [15] accennate nel capitolo precedente appartengono alla branca di cybersecurity che si occupa di rilevare minacce in una fase iniziale con una bassa percentuale di errore. Queste tecnologie usano esche che simulano delle vere istanze tecnologiche, come siti, database, server e cartelle di file, posizionate nelle vicinanze delle componenti della rete produttiva.

Il loro scopo è quello di attirare gli attaccanti all'interno di queste esche contenenti informazioni e dati fittizi, e distogliere la loro attenzione dai file e dai prodotti veri. Consentono anche di rilevare ogni accesso alle esche, e avvertono tramite opportuna notifica l'effettivo accesso in tempo reale, per permettere di contenere l'attacco nei primi momenti. Inoltre, questi strumenti permettono di intercettare e bloccare le minacce, permettendone anche lo studio con l'obiettivo di migliorare le difese della rete esistente, evidenziando le parti dell'infrastruttura di rete maggiormente deboli.

L'utilizzo di queste tecnologie in una rete aziendale porta molti benefici, tra i quali :

Copertura difensiva :

Migliora infatti il rilevamento di minacce nella rete, e permette di aumentarne la copertura difensiva tramite l'intercettazione. Inoltre garantiscono il contenimento della minaccia in sicurezza, dando la possibilità anche di studiarla e capire com'è stata progettata e strutturata.

Basso numero di falsi positivi :

Infatti tutte le interazioni con le esche monitorate da queste tecnologie sono da ritenersi minacce : nessuno che conosce il posizionamento di queste esche ha il bisogno di accedervi una volta posizionate, poiché non hanno bisogno di manutenzione.

Difesa predicibile e preimpostata :

Grazie alla possibilità di studio nell'intercettazione e contenimento delle minacce, si ha anche la possibilità di migliorare le difese della rete con le conoscenze create nel tempo, creando così un sistema sempre più efficiente nella difesa e nella protezione dei dati.

Ci sono vari tipi di deception technology, in base al loro posizionamento nella rete [15] :

1. *Perimeter deception defense*

Esche poste sul perimetro della rete, per impedire a minacce esterne di entrare nella rete aziendale. Permette di avere molti dati riguardo gli attaccanti, perchè sono aperte a tutto il traffico di rete. Tuttavia questo porta anche la possibilità di scan di rete globali, i quali non sono definibili minacce, ma che comunque verranno rilevati.

2. *Network deception defense*

Esche poste all'interno della rete, dove un attaccante potrebbe andare a guardare : vicino a database, cartelle o applicazioni e software. Grazie alle notifiche mandate da queste esche, si può capire se dei malintenzionati sono entrati nella rete e dove sono esattamente. Questo posizionamento è quello standard, perché permette di avere una buona copertura delle risorse, con un rischio accettabile.

3. *Endpoint deception defense*

Esche posizionate nei terminali di lavoro, come pc aziendali o server. Lo scopo di questo posizionamento è un sorta di "ultima difesa", per distrarre l'attaccante dalle informazioni contenute nelle vere cartelle. Un esempio potrebbe essere un file chiamato "passwords.txt", contenente password inventate e diverse da quelle realmente utilizzate, ma che comunque potrebbero sembrare autentiche. Questo posizionamento è utile anche a rilevare dei "lavoratori infedeli", i quali potrebbero accedere a risorse non spettanti : nel momento in cui aprissero una risorsa creata appositamente, come nel caso di "passwords.txt", sarebbero scoperti.

Capitolo 2

Concetti di Honeypot

2.1 Definizione di honeypot e suo funzionamento

Un honeypot[16] in cybersecurity è un particolare dispositivo di difesa afferente all'insieme di deception technology, il cui scopo è fare da esca per attirare gli aggressori informatici in un ambiente isolato dall'ambiente di produzione principale con l'obiettivo di studiare ed osservare in sicurezza il modo in cui operano [17]. Infatti, un honeypot è un ambiente sicuro e diviso dal sistema di rete principale, creato appositamente per apparire vulnerabile agli occhi di trasgressori emulando uno o più applicativi informatici del mondo reale, come per esempio una pagina di accesso, un servizio, un'applicazione aziendale interessante per i trasgressori oppure una cartella con informazioni importanti e private.

Il suo funzionamento è semplice : l'obiettivo di un honeypot è attirare i trasgressori e intrappolarli al loro interno, inviando poi un avviso al team di sicurezza non appena viene rilevato un tentativo di accesso, fornendo visibilità e tracciamento dell'attività dei malintenzionati. Il punto focale del loro utilizzo è il creare falsi dati che siano “*too attractive to resist*” per un assalitore, come per esempio un file denominato “passwords.txt” contenente nomi utente che sembrano veri (ma che non lo sono) con password casuali ma credibili. Non appena il malintenzionato prova ad aprire il file o ad accedere a una pagina di login, viene mandata la notifica di tentativo di accesso in modo che si possa controllare e bloccare la minaccia in corso, e in seguito migliorare i propri meccanismi di difesa studiando gli attacchi subiti.

L'aspetto di maggior rilievo del suo funzionamento è il bassissimo tasso di errore nell'avviso di minaccia, poiché ogni tentativo di accesso a questi strumenti è da considerarsi sospetto. Infatti non esiste un motivo valido per cui un utente debba interagire con un honeypot, e la maggior parte delle volte che vengono rilevati accessi sono da persone con intenzioni criminali, sia che siano all'esterno della rete aziendale, sia al proprio interno.

Un'organizzazione che adotta l'uso di questi dispositivi di sicurezza può migliorare la pro-

pria strategia di sicurezza informatica, identificando la fonte degli attacchi, i metodi in cui si svolgono, e gli strumenti che vengono utilizzati. Identificando infatti i propri punti deboli, l'azienda che adotta l'uso di honeypot può concentrare i suoi sforzi informatici nei punti più carenti della propria architettura di rete esistente, o può prioritizzare la difesa delle risorse maggiormente prese di mira.

In sostanza, gli honeypots agiscono come vere e proprie trappole e distrazioni informatiche. Distolgono i malintenzionati dalle risorse e dalle applicazioni reali, fornendo al contempo preziose informazioni sulle vulnerabilità potenziali e sulle minacce emergenti. Comprendendo e analizzando le interazioni con gli honeypots, le organizzazioni possono rafforzare le loro difese informatiche.

2.2 Storia ed evoluzione degli honeypot

Il concetto di honeypot esiste dall'inizio degli anni Novanta, quando l'idea è stata documentata in due pubblicazioni : “The Cuckoo’s Egg” di C. Stoll e “An Evening with Berferd” di B. Cheswick[18], entrambe del 1991, in cui si descriveva un esperimento per attirare i pirati informatici in una trappola (al tempo chiamata “roach motel”) appositamente creata dagli autori, in modo da osservare e studiare i loro metodi di attacco.

Tuttavia la prima implementazione, solo per la comunità di sicurezza, avvenne nel 1997 con la pubblicazione del “Deception Toolkit” di F. Cohen, dove era possibile emulare un sistema UNIX con le principali vulnerabilità conosciute. Lo stesso autore nel 1998 sviluppò CyberCop Sting[18], uno dei primi honeypot commerciali disponibili al pubblico.

Nel 1999 L. Spitzner fondò la Honeynet Project, una comunità non-profit che si impegna a studiare e sviluppare tecniche di infiltrazione informatica. Questa comunità è ancora attiva al giorno d'oggi ed è visitabile da <http://www.honeynet.org> .

Con il passare del tempo questi strumenti si sono evoluti e gli esemplari più moderni prevedono l'abbandono di risorse fisiche, utilizzando al loro posto un linguaggio LLM che risponda ad ogni interazione e comando dell'attaccante.

2.3 Classificazione di honeypot

Ogni tipologia di honeypot presenta casi d'uso specifici e conseguenti punti di forza e di debolezza. Pertanto, si devono valutare attentamente il proprio obiettivo e le proprie risorse prima di progettare una strategia di honeypotting. Possiamo distinguere i vari tipi di honeypot per tecnologia, per interazione e per categoria di appartenenza.

2.3.1 Per tecnologia

La prima grande distinzione che si può fare è per tecnologia, infatti esistono più tecnologie di honeypotting :

Honeypot Standard :

Si tratta di un honeypot completo, composto da un sistema hardware e da uno software, come un computer o un server fisico.

Honeytoken [19] :

Sono singoli pezzi di dati creati appositamente per emulare delle credenziali, delle risorse o dei file. Vengono poi appositamente posti in modo da segnalare eventuali intrusioni da parte di attaccanti nelle reti da proteggere. Hanno bisogno di meno manutenzione rispetto agli honeypot standard, ed offrono una implementazione semplice e veloce. Il loro obiettivo è quello di agire come trappole, che una volta innescate permettono di rilevare la presenza di un attaccante, e di capire la sua posizione nella rete.

Honeynet [1] :

È una rete formata da più honeypot standard.

¹ Spiegate in dettaglio nel capitolo 3.2

2.3.2 Per interazione

Si possono poi classificare gli honeypot standard in base al loro grado di interazione :

Bassa interazione :

Consentono un'interazione parziale con i sistemi, perché contengono pochi servizi e hanno ristrette funzionalità d'uso. Si ottengono di solito da emulatori di software e di sistemi operativi virtuali, con vulnerabilità note e non nascoste. Sono facili da rilevare, e non permettono una completa copertura di ogni comportamento possibile dell'attaccante, ma hanno implementate solo le funzioni basiche. Questo implica un rischio ridotto in quanto i malware non vengono realmente eseguiti poiché bloccati dal servizio di emulazione stessa. Sono facili da schierare ed utilizzare, poiché sono ambienti prettamente statici, ma non sono però così efficaci, vengono infatti usati prettamente per l'individuazione di bot.

Alta interazione :

Questi sistemi sono composti da componenti sia software che hardware, con installati veri sistemi operativi. Sono più complessi e consentono agli aggressori di interagire in modo totale con l'honeypot. É meno facile da rilevare, ed ha integrate molte funzioni, una tra le quali è la possibilità di ripristinare il sistema in caso di attacco malware avanzato. Richiedono più risorse e più manutenzione rispetto agli honeypot a bassa interazione, ed hanno un rischio specifico più alto, in quanto l'attaccante potrebbe prendere il possesso del sistema operativo ed usarlo come origine di molteplici attacchi rivolti ad altre macchine o reti. É dunque opportuno monitorare costantemente questa tipologia di honeypot, e limitare il traffico in uscita, oltre che avere un team dedicato di esperti per controllare e mettere in sicurezza il sistema.

Di seguito una tabella per elencare le differenze sostanziali tra queste due tipologie.

	Alta Interazione	Bassa Interazione
Tipo di interazione	Reale/Virtuale	Emulazione
Installazione	Complessa	Semplice
Manutenzione	Frequente	Non frequente
Costi	Elevati	Bassi
Monitoraggio	Costante	Non necessario
Acquisizione dati	Elevata	Limitata
Rischio	Alto	Basso

Tabella 2.1: Confronto tra honeypot ad alta e bassa interazione

2.3.3 Per categoria

Infine, gli honeypot standard si possono distinguere in due categorie :

Produzione :

Di solito appositamente posizionati vicino ai server e alle reti di produzione, con l'obiettivo di identificare delle falle di sicurezza o compromissioni nella rete aziendale ed aumentarne la sicurezza. Sono una vera e propria esca, posizionata vicino o all'interno delle reti aziendali. Generalmente si usano honeypot a bassa interazione, poiché semplici da posizionare ed economici da mantenere.

Ricerca :

Di solito posizionati al di fuori delle reti produttive e aziendali, all'esterno del firewall. Il loro obiettivo è quello di raccogliere dati riguardo gli attaccanti e le loro strategie, con lo scopo di migliorare i meccanismi di difesa esistenti. Forniscono infatti preziose informazioni riguardo nuove tipologie, tecniche o strumenti usati da attacchi informatici. Sono usati principalmente da organizzazioni di ricerca, da entità militari o governative. Tipicamente vengono usati honeypot ad alta interazione, perché garantiscono l'ottenimento di più informazioni riguardo la minaccia, anche se hanno un costo notevolmente più alto di implementazione e di mantenimento.

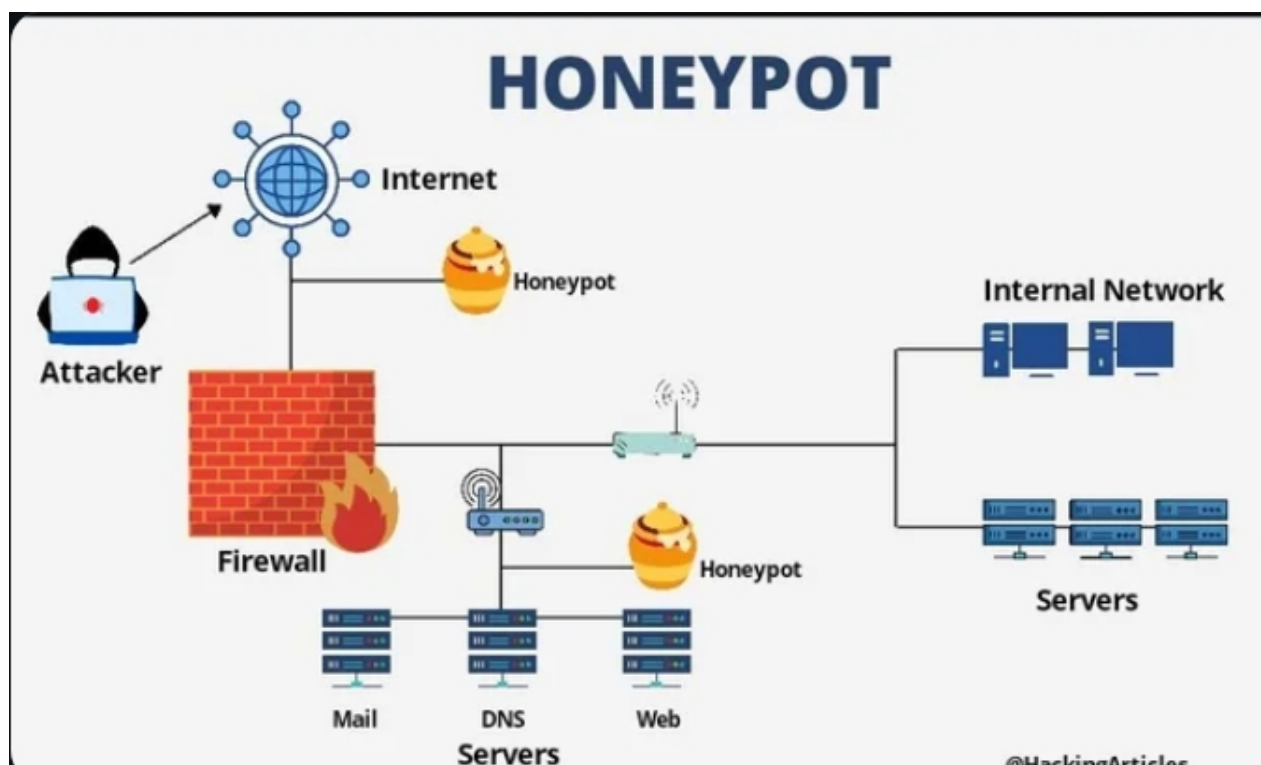


Figura 2.1: Posizionamento di honeypot di ricerca e di produzione

Come si può vedere in questa immagine, nella rete sono implementate entrambe le tipologie appena spiegate : l'honey-pot esterno al firewall e a diretto contatto con il traffico di rete appartiene alla categoria degli honeypot di ricerca, mentre quello più interno, dentro il firewall e posizionato vicino alle componenti della rete aziendale, appartiene alla categoria degli honeypot di produzione.

Capitolo 3

Uso di honeypot in rete aziendale

3.1 Scopo e benefici dell'uso di Honeypot in rete aziendale

3.1.1 Scopo degli Honeypot

Gli honeypot sono diversi dai comuni strumenti di sicurezza, come il firewall, che vigila sul perimetro della rete e monitora le connessioni, e le IDS, *Intrusion Detection Systems*, che generano allarmi quando rilevano una probabile intrusione dopo aver esaminato i pacchetti di rete. Gli honeypot, inoltre, non forniscono servizi di produzione, cioè non creano, trasformano o modificano informazioni e file, ma li monitorano e li proteggono da attacchi esterni.

Questo li differenzia dalle informazioni che devono monitorare; sono uno strumento adattabile a più situazioni e di facile utilizzo, ma vanno visti come completamento dei sistemi di sicurezza. Infatti è importante ricordare che gli honeypot monitorano solo le risorse che gli sono state assegnate, ed emanano allarmi solo nel caso in cui gli attaccanti vengano a contatto con l'honeybot stesso; mentre gli IDS garantiscono una protezione completa, scannerizzando ogni singolo pacchetto di rete mandato e ricevuto, pur avendo un numero di falsi positivi non indifferente.

La scelta del tipo di honeypot da implementare dipende da vari fattori, tra cui gli obiettivi di sicurezza voluti, le risorse a disposizione per gestire l'infrastruttura di rete e il livello di rischio che l'azienda è disposta ad accettare. Infatti, scegliendo un honeypot a bassa interazione, il livello di rischio scelto sarà basso, ma saranno basse anche l'utilità e la sicurezza fornite da quest'ultimo. Quello ad alta interazione fornirà più dettagli legati agli attacchi e alle tecniche usate dagli attaccanti, avendo però un rischio intrinseco più alto.

Indipendentemente dalla scelta di tipologia, l'implementazione di un honeypot nella strategia di sicurezza informatica rappresenta un passo importante verso una difesa della rete più proattiva, contro le minacce in continua evoluzione.

Il suo impiego permette all'utilizzatore di capire quali sono le parti vulnerabili della propria rete, avendo così la possibilità di migliorare la sicurezza di aree specifiche. Inoltre, permette anche di comprendere e studiare le tecniche di attacco subite, poiché ogni interazione con questi sistemi di difesa viene registrata. Le azioni registrate dagli honeypot riguardano i tentativi di accesso, i movimenti all'interno del sistema, i comandi eseguiti dagli attaccanti e i tentativi di copiatura di file o cartelle.

3.1.2 Vantaggi degli honeypot

I vantaggi[19] di un honeypot sono molteplici:

Strumenti semplici e a bassa manutenzione :

Sono infatti strumenti facili da implementare ed efficaci per fornire avvisi ed informazioni sul comportamento degli aggressori delle reti. Inoltre non è necessario monitorarli costantemente, ma sarà l'honeytrap stesso ad avvisarci quando qualcuno interagisce con l'esca.

Forniscono un preavviso per potenziali attacchi sulla rete :

Infatti, non appena qualcuno entra in contatto con l'esca posizionata dall'honeytrap, questo manderà un avviso. In questo modo un potenziale attaccante che era libero per la rete aziendale non ha più l'effetto sorpresa, bensì è stato scoperto e possibilmente fermato prima che faccia ingenti danni.

Testare i processi di risposta automatici :

Sono infatti un modo efficiente per studiare i tentativi di accesso e mettere in pratica delle difese migliori, anche automatiche, in risposta agli attacchi studiati. Inoltre, con lo studio degli attacchi si possono capire i punti vulnerabili della rete, e difenderla di conseguenza in un modo migliore.

Bassa probabilità di falso positivo :

Infatti la presenza di avvisi provenienti da un honeypot comporta sicuramente un'intrusione nel suo sistema : bisogna quindi intervenire per controllarlo ogni qual volta produca un avviso.

Diminuisce il MTTD :

Infatti il *Mean Time To Detect* diminuisce notevolmente quando in una rete è presente un'architettura honeypot, migliorando così il tempo di risposta agli attacchi.

Fornisce i dettagli dell'attività dell'attaccante attraverso logs :

Infatti, tramite i logs dell'honeypot si possono sapere che comandi ha eseguito il criminale, e che cartelle ha visitato, oltre a che username e password ha inserito per accedere. In questo modo, si possono studiare i metodi di attacco subiti, potendo così migliorare le proprie difese per gli attacchi futuri.

Strumenti che rallentano un attacco in corso :

Infatti, un attaccante che è entrato in una rete ha poco tempo a disposizione prima che venga intercettato e bloccato dagli strumenti di sicurezza. Se questo attaccante entra in contatto con un honeypot, di fatto perde tempo e non provoca danni reali al sistema.

3.2 Architettura e design di un sistema honeypot

Un aspetto importante nell'implementazione degli honeypot è il loro posizionamento nella rete: è possibile posizionarli sia internamente al firewall, sia esternamente. Nel primo caso gli honeypot non sono esposti alla rete esterna, quindi svolgono una funzione prevalentemente di “ultima difesa”, perché verranno prodotti avvisi solo quando un attaccante ha già oltrepassato il firewall. Un avviso proveniente da questo tipo di honeypot deve mettere in guardia gli addetti alla sicurezza, perché un attaccante è a piede libero nel sistema. Invece, nel caso in cui l'honeypot venga messo all'esterno del firewall, sarà esposto a tutti gli attacchi esterni, garantendo così un grande flusso di informazioni, che permette uno studio completo sulle metodologie di attacco usate. Questo tipo di implementazione è usato principalmente per gli honeypot di ricerca, in modo da avere molti dati ed attacchi su cui effettuare ricerca e migliorare le proprie difese.

Tipo	Descrizione
Sacrificial Lamb	Un honeypot isolato, senza ingressi o collegamenti al sistema produttivo.
Hacker Zoo	Un intero sistema isolato di honeypots, con differenti servizi emulati, vulnerabilità e configurazioni.
Minefield	Un gran numero di honeypot posizionati davanti alle risorse maggiormente attaccate.
Proximity Decoys	Honeypots posizionati in prossimità del sistema di produzione principale.
Redirection Shield	Honeypots esterni che sembrano essere posizionati all'interno del sistema di produzione usando il reindirizzamento delle porte.
Deception Ports	Servizi emulati, come DNS e FTP, nel sistema di produzione.

Tabella 3.1: Implementazioni possibili di honeypot[20]

Una architettura usata frequentemente in azienda è la *honeynet*, che consiste in due o più honeypot standard interconnessi, in modo da emulare una rete completa o un server, riuscendo così a distogliere l'attenzione dell'attaccante dalla rete primaria.

Questa honeynet è posta in una rete controllata e sicura, in cui vengono usati dei “layer 2 bridge”, meglio conosciuti come *honey walls*, con l'obiettivo di filtrare i dati in entrata e in uscita. La prima implementazione di queste tecnologie è attribuibile a Honeynet Project[21].

Gli aggressori all'interno di queste *honeynets* non troveranno solo applicazioni o server vulnerabili, ma anche router, firewall e altri dispositivi di rete.

Di seguito troviamo delle caratteristiche tipiche delle honeynet :

- I dispositivi di rete vengono lasciati configurati ai valori di default, in modo da renderli suscettibili ad attacchi.
- Le esche usate non contengono informazioni sensibili, quindi questi file possono essere modificati o eliminati dagli aggressori senza conseguenze.
- Le esche contengono anche degli strumenti di monitoraggio, in modo da controllare l'operato degli attaccanti ed essere avvisati non appena questi file risultano sotto attacco.

Inoltre, le honeynet si possono differenziare in 3 sottocategorie, a seconda della generazione dei propri *honey walls* :

1. Gen I

Sono la prima implementazione di honeywalls in una honeynet, ed hanno la caratteristica di essere stati implementati manualmente. Non essendoci uno standard nel progettarli, gli honeywall di prima generazione variano considerevolmente riguardo le loro specifiche, ma la maggior parte di questi veniva dotata di un *host* con due o più interfacce di rete, per effettuare un collegamento tra le loro partizioni. Il loro scopo era quello di controllare, analizzare e collezionare i dati e catturare le minacce rintracciabili.

2. Gen II

Nella seconda generazione si è riusciti ad avere un'implementazione standard usufruibile a tutti grazie ad un CD ROM creato da Honeynet Project[21]. La particolarità di questa generazione era la possibilità di controllare sia il rate che il tipo di dati in entrata. Era inoltre in grado di intercettare attacchi tramite *Snort Inline*[22], un potente sistema open-source di rilevamento delle intrusioni.

3. Gen III

La loro architettura è uguale a quella della Gen II, ma sono dotati di una interfaccia web più moderna e di sistemi migliorati di allarme e di management degli eventi.

3.3 Strumenti e software open-source per la creazione di honeypot

Ci sono molti strumenti open-source per l'implementazione di honeypot, nelle prossime sottosezioni ci saranno delle descrizioni di due progetti che saranno implementati nel Cap. 4.

3.3.1 Cowrie

Cowrie^{[23]¹} è un honeypot ssh a media interazione open-source, usato per ottenere informazioni dettagliate sugli attacchi *brute force* e sui comandi eseguiti nella *shell* dagli attaccanti. Infatti, questo honeypot emula una shell in Linux, in cui sono già presenti dei dati e delle cartelle, con la possibilità di cambiarli o aggiungerne di nuovi. È il successore di Kippo, un altro honeypot open-source che emula shell in Linux. Entrambi gli honeypot sono stati sviluppati da M.Oosterhof. Come detto in precedenza, i punti di forza di questo honeypot sono la possibilità di avere i *logs* dettagliati delle interazioni tra trasgressori e shell, e la sicurezza dell'ambiente virtuale, poiché anche se i trasgressori riuscissero ad entrare nel sistema di Cowrie, non troverebbero nulla di compromettente in quanto l'ambiente virtuale è diviso dall'ambiente fisico aziendale.

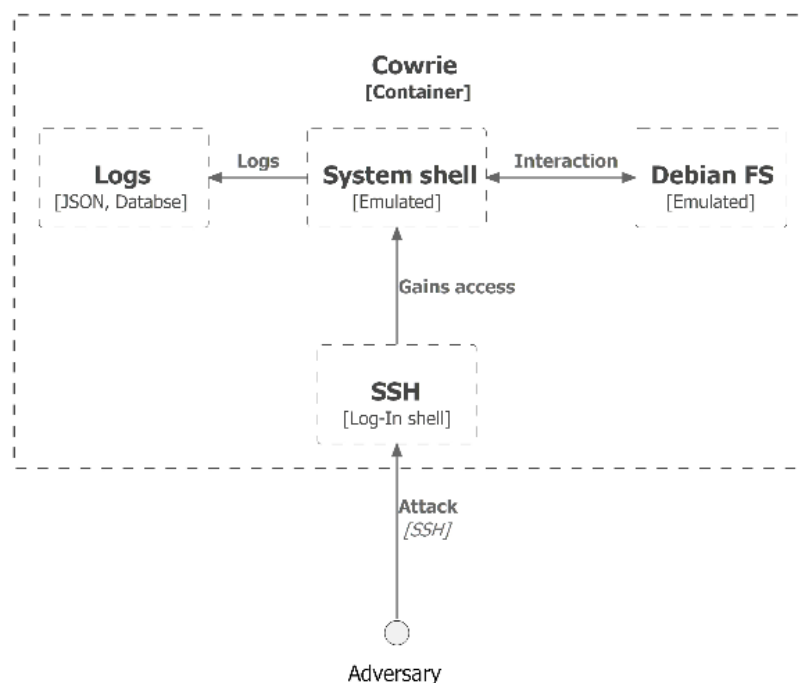


Figura 3.1: Architettura di Cowrie

¹L'implementazione è presente al capitolo 4.1.1

3.3.2 T-Pot

T-Pot^[24]² è una piattaforma di honeypot *all-in-one*, supporta più di 20 tipi di honeypot e ha al suo interno molte opzioni di visualizzazione per gli attacchi subiti, tramite Elastic Stack^[3]. Permette infatti di visualizzare una mappa in cui si può vedere la provenienza degli attacchi subiti, e di creare una dashboard in cui si possono vedere le credenziali più usate, la lista di indirizzi IP da dove provengono gli attacchi e la quantità di minacce intercettate.

Come detto in precedenza, al suo interno contiene più di 20 honeypot, di seguito una breve descrizione dei più importanti :

ADBHoney^[25] :

É un honeypot a bassa interazione che ha lo scopo di raccogliere malware caricati in dispositivi android su protocollo TCP.

Cisco ASA^[26] :

É un honeypot a bassa interazione che ha lo scopo di intercettare CVE-2018-0101^[27], una vulnerabilità che consiste nella possibilità di un attaccante remoto di connettersi ed eseguire codice dopo un *reboot* del sistema.

Conpot^[28] :

É un honeypot a bassa interazione per sistemi industriali ICS e SCADA. Offre un alta gamma di protocolli industriali, e c'è la possibilità di includere un interfaccia supervisionata da un umano, per aumentare la superficie d'attacco. Inoltre, utilizzando un *delay* artificiale, Conpot emula un sistema industriale che gestisce un certo carico di informazioni.

DDoSPot^[29] :

É un honeypot a bassa interazione con lo scopo di intercettare e loggare attacchi di *Distributed denial of service*, *DDoS*^[30]. Questo attacco consiste nel mandare un grande traffico di dati verso la vittima provenienti da fonti diverse, con lo scopo di "inondare" il servizio attaccato e renderlo inutilizzabile per un certo periodo.

Dicompot^[31] :

É un honeypot a bassa interazione per il *Digital imaging and Communications in Medicine*, *DICOM*. Emula un server DICOM in Go, per collezionare logs e intercettare attacchi.

Dionaea^[32] :

É un honeypot a media interazione che ha lo scopo di intercettare e catturare copie di malware emulando servizi e protocolli FTP e HTTP.

²L'implementazione è presente al capitolo 4.1.3

³Anche conosciuto come ELK : Elasticsearch, Logstash e Kibana

Elasticpot^[33] :

È un honeypot a bassa interazione per Elasticsearch, il motore di ricerca di ELK.

Glutton^[34] :

È un honeypot a bassa interazione che lavora come *Man in the middle*, *MITM* per protocolli SSH e TCP.

Heralding^[35] :

È un honeypot che ha lo scopo di catturare credenziali per protocolli FTP, Telnet, SSH e HTTP.

Honeytrap^[36] :

È un honeypot a bassa interazione con lo scopo di garantire sicurezza di rete, e di proteggere servizi basati su UDP o TCP.

Inoltre, permette l'utilizzo di molti strumenti di supporto, tra i quali **Autoheal**, che permette di effettuare un *restart* automatico nel caso in cui gli ambienti virtuali usati si trovino in condizioni critiche; **Elastic Stack**, già citato in precedenza, che permette di visualizzare tutti gli eventi intercettati dagli honeypots; **T-Pot Attack Map**, una mappa animata che visualizza in tempo reale la provenienza degli attacchi; **Suricata**, un motore di monitoraggio della sicurezza di rete.

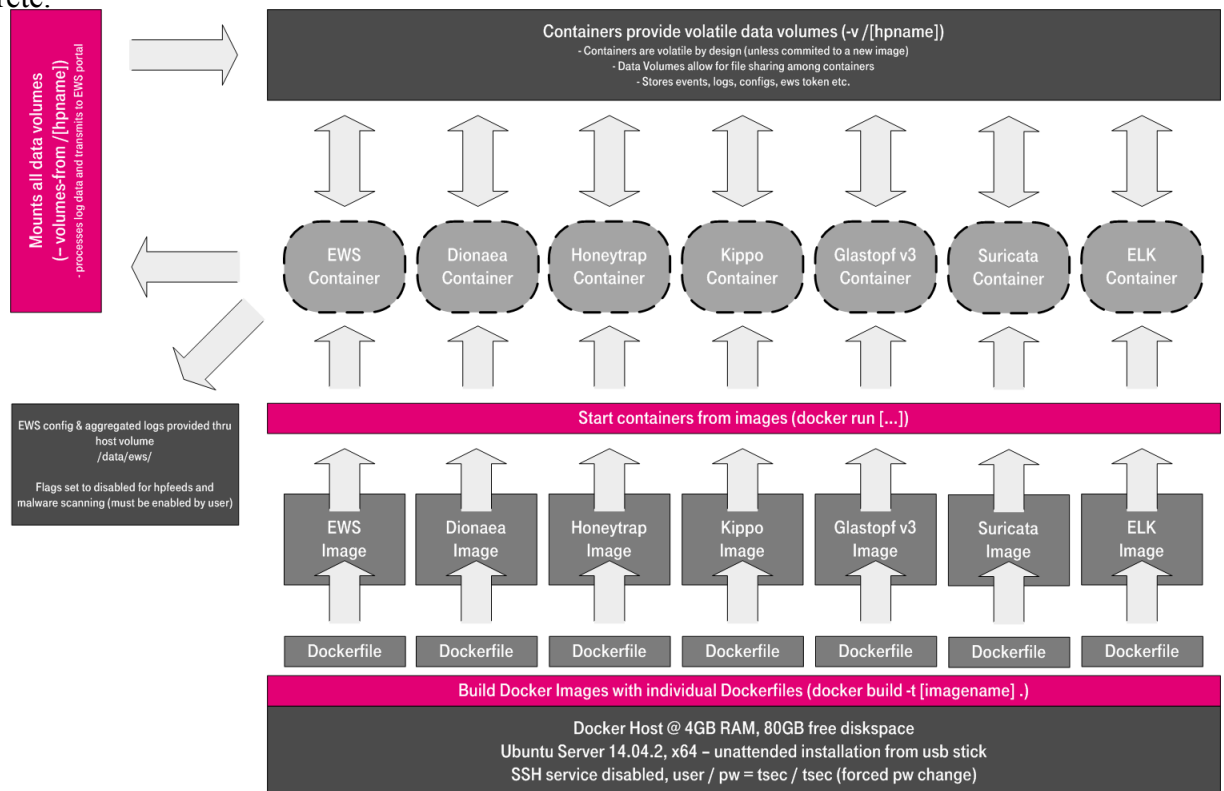


Figura 3.2: Architettura di T-Pot

Capitolo 4

Case study e Analisi dei dati

In questo capitolo sono descritte le implementazioni dei due honeypot open-source scelti : Cowrie[23] e T-Pot[24]. Inoltre, è presente l'analisi dei dati raccolti tramite questi strumenti, ed infine è presente anche una valutazione riguardo alla propria efficacia nel migliorare la sicurezza della rete aziendale.

4.1 Esempi di implementazioni reali di honeypot in azienda

I due honeypot sono stati installati ed eseguiti in due macchine virtuali create con VMWare[37], su cui si è scaricato il sistema operativo Ubuntu Server 24.04.2[38] .

4.1.1 Installazione di Cowrie

Per l'installazione di Cowrie abbiamo usato Docker[39], seguendo la guida ufficiale[40]. I passaggi sono i seguenti :

1. Cambiare porta SSH :

Abbiamo scelto di spostare la porta SSH dalla standard 22 alla 4222, poiché Cowrie ha bisogno di connettersi alla porta 22 ed effettuare un bind con la 2222, la quale è monitorata dall' honeypot.

Si può cambiare porta con il seguente codice :

```
$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.backup
$ sudo nano /etc/ssh/sshd_config
Port 4222
$ sudo ufw allow 4222/tcp
$ sudo systemctl restart sshd
```

Bisogna copiare il file di configurazione standard, e modificarlo mettendo la porta desiderata. Dopo un restart, la porta sarà effettivamente cambiata.

2. Scaricare Docker :

Abbiamo scaricato Docker seguendo la guida ufficiale [\[41\]](#)

3. Creare il *compose* :

Creare una cartella `docker/cowrie`, in cui creare il `compose.yaml` :

```
services:

  cowrie:

    image: cowrie/cowrie:latest

    hostname: cowrie-hostname

    ports:

      - "2222:2222"

    networks:

      - cowrie_network

networks:

  cowrie_network:

    name: cowrie_network

    driver: bridge
```

4. Eseguire Cowrie :

Tramite il comando

```
$ docker compose up -d
```

si esegue l'honeypot.

5. Vedere i logs :

Tramite il comando

```
$ docker logs cowrie -f
```

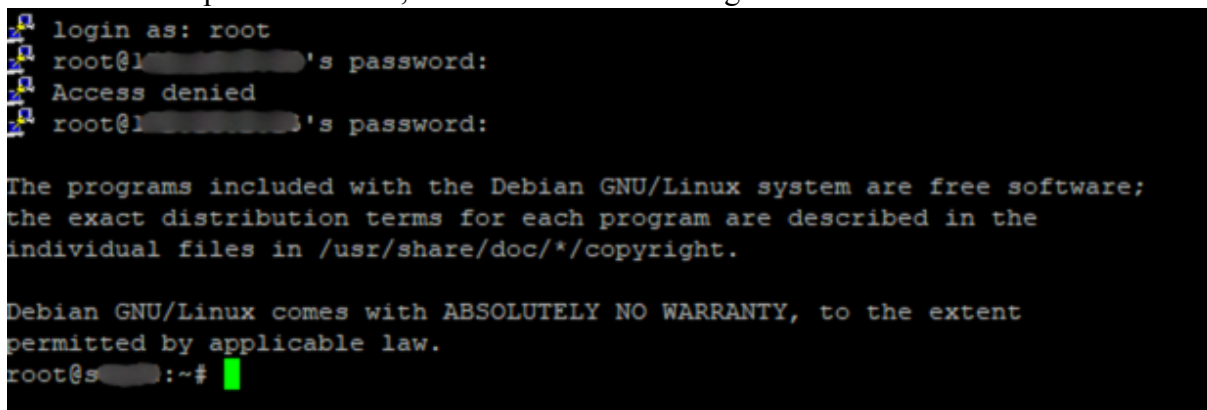
si possono vedere in tempo reale le connessioni effettuate alle porte ssh scelte, oltre alle credenziali di accesso usate e i comandi eseguiti dagli attaccanti.

6. Bloccare esecuzione di Cowrie

```
$ docker compose down
```

4.1.2 Test di Cowrie

Per testare l'operato di Cowrie abbiamo utilizzato l'applicazione *mRemote NG* [42], che ci permette di connetterci da remoto alle macchine virtuali create. Non appena effettuiamo il collegamento ci vengono chiesti user e password per effettuare l'accesso : di default Cowrie accetta qualsiasi combinazione di *username* e *password* dopo un numero di tentativi determinato da un intero *random* compreso tra 2 e 10, che viene cambiato ad ogni interazione.



```
login as: root
root@1[REDACTED]'s password:
Access denied
root@1[REDACTED]'s password:

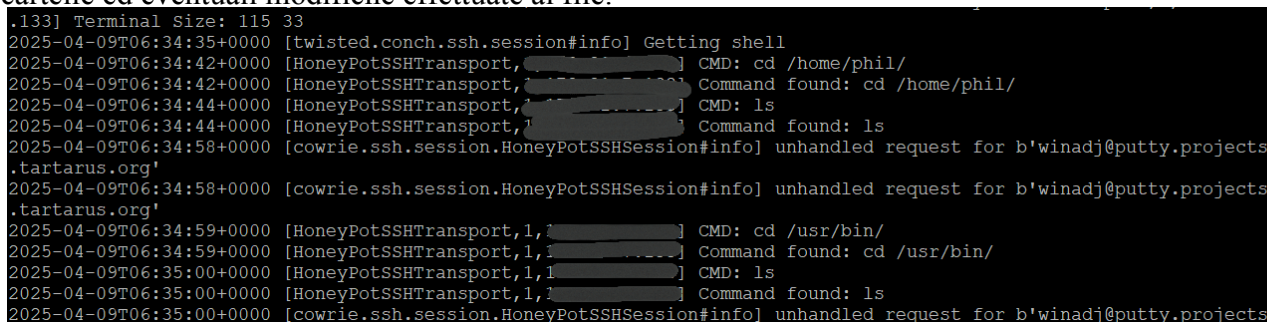
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@s[REDACTED]:~#
```

Figura 4.1: Esempio di accesso di Cowrie

In questo caso, abbiamo utilizzato come user : *root* e come password prima *root* e poi *admin*. Al secondo tentativo siamo riusciti a effettuare l'accesso nell'ambiente virtuale creato appositamente con sistema operativo Debian Linux.

Come detto in precedenza, c'è la possibilità di consultare i logs, in cui si possono vedere l'indirizzo IP dell'attaccante, lo username utilizzato e la password inserita, oltre a ogni movimento tra cartelle ed eventuali modifiche effettuate ai file.



```
.133] Terminal Size: 115 33
2025-04-09T06:34:35+0000 [twisted.conch.ssh.session#info] Getting shell
2025-04-09T06:34:42+0000 [HoneyPotSSHTransport,1,[REDACTED]] CMD: cd /home/phil/
2025-04-09T06:34:42+0000 [HoneyPotSSHTransport,1,[REDACTED]] Command found: cd /home/phil/
2025-04-09T06:34:44+0000 [HoneyPotSSHTransport,1,[REDACTED]] CMD: ls
2025-04-09T06:34:44+0000 [HoneyPotSSHTransport,1,[REDACTED]] Command found: ls
2025-04-09T06:34:58+0000 [cowrie.ssh.session.HoneyPotSSHSession#info] unhandled request for b'winadj@putty.projects
.tartarus.org'
2025-04-09T06:34:58+0000 [cowrie.ssh.session.HoneyPotSSHSession#info] unhandled request for b'winadj@putty.projects
.tartarus.org'
2025-04-09T06:34:59+0000 [HoneyPotSSHTransport,1,[REDACTED]] CMD: cd /usr/bin/
2025-04-09T06:34:59+0000 [HoneyPotSSHTransport,1,[REDACTED]] Command found: cd /usr/bin/
2025-04-09T06:35:00+0000 [HoneyPotSSHTransport,1,[REDACTED]] CMD: ls
2025-04-09T06:35:00+0000 [HoneyPotSSHTransport,1,[REDACTED]] Command found: ls
2025-04-09T06:35:00+0000 [cowrie.ssh.session.HoneyPotSSHSession#info] unhandled request for b'winadj@putty.projects
```

Figura 4.2: Esempio di comandi eseguiti nell'interfaccia di Cowrie

Una volta effettuato il login, possiamo spostarci di *directory*, tramite il comando

```
$ cd /home/phil
```

perché di default viene creata una cartella sotto l'utente phil contenente varie tipologie di file.

```
root@svr04:~# ls
root@svr04:~# pwd
/root
root@svr04:~# cd /home/phil
root@svr04:/home/phil# ls
root
bin      boot      dev        etc        home       initrd.img lib        lost+found media  mnt        opt        proc
root     run       sbin       selinux    srv        sys        tmp        usr        var        vmlinuz   test2
root@svr04:/home/phil# cd /etc
root@svr04:/etc# ls
X11
bash.bashrc      acpi              adduser.conf      alternatives      apt
calendar          bash_completion.d bindresvport.blacklist blkid.tab         blkid.tab.old
cron.monthly      console-setup     cron.d            cron.daily        cron.hourly
cron.weekly       cron.weekly      crontab           debconf.conf      debian_version
default           deluser.conf     dhcp             dictionaries-common discover-modprobe.conf
discover.conf.d   dkms             dpkg             drirc             emacs
environment       fstab            fstab.d          gal.conf          groff
group            group-          grub.d           gshadow          gshadow-
host.conf         hostname        hosts            hosts.allow       hosts.deny
init             init.d          initramfs-tools  inittab           inputrc
insserv          insserv.conf    inserv.conf.d    iproute2          iscsi
issue            issue.net       kbd              kernel            kernel-img.conf
ld.so.cache      ld.so.conf      ld.so.conf.d     libaudit.conf     locale.alias
locale.gen       localtime      logcheck         login.defs        logrotate.conf
logrotate.d      magic          magic.mime       mailcap           mailcap.order
manpath.config   menu           menu-methods     mime.types        mke2fs.conf
modprobe.d       modules        motd             mnt              nanorc
network         networks       nologin          nsswitch.conf     opt
ps-release      pam.conf       pam.d            passwd            passwd-
profile         profile.d      protocols        python            python2.7
rc.local        rc0.d          rc1.d            rc2.d             rc3.d
rc4.d           rc5.d          rc6.d            rcS.d             resolv.conf
rmt             rpc            rsyslog.conf     rsyslog.d         securetty
security        selinux       services         shadow            shadow-
```

Figura 4.3: Esempio di file contenuti in Cowrie

4.1.3 Installazione di T-Pot

Per l'installazione di T-Pot abbiamo seguito la guida ufficiale[24]. Il processo è veloce e semplice, divisibile in due parti :

1. Installare curl :

```
$ sudo [apt, dnf, zypper] install curl
```

2. Lanciare l'installazione :

```
$ env bash -c "$(curl -sL https://github.com/telekom-
security/tpotce/raw/master/install.sh)"
```

Una volta avviato il processo, il programma in autonomia scaricherà le immagini docker degli honeypot e tutte le dipendenze che servono per utilizzare T-Pot. Si potrà poi scegliere il tipo di installazione dell'honeyot voluta, tra Hive, Sensor, LLM, Mini, Mobile o Tarpit. Abbiamo scelto Hive, poiché è l'installazione standard che permette di avere la maggior parte degli honeypot disponibili.

```

### Playbook was successful.

### Choose your T-Pot type:
### (H)ive - T-Pot Standard / HIVE installation.
### Includes also everything you need for a distributed setup with sensors.
### (S)ensor - T-Pot Sensor installation.
### Optimized for a distributed installation, without WebUI, Elasticsearch and Kibana.
### (L)LM - T-Pot LLM installation.
### Uses LLM based honeypots Beelzebub & Galah.
### Requires Ollama (recommended) or ChatGPT subscription.
### M(i)ni - T-Pot Mini installation.
### Run 30+ honeypots with just a couple of honeypot daemons.
### (M)obile - T-Pot Mobile installation.
### Includes everything to run T-Pot Mobile (available separately).
### (T)arpit - T-Pot Tarpit installation.
### Feed data endlessly to attackers, bots and scanners.
### Also runs a Denial of Service Honeypot (ddospot).

```

Figura 4.4: Scelta di installazione di T-Pot

In seguito bisogna creare username e password per l'interfaccia web, contenente una mappa degli attacchi ed una dashboard con i dati più importanti riguardanti gli attacchi, come indirizzi IP, username e password usati nel login. È possibile connettersi alla dashboard tramite browser all'indirizzo :

`https://<indirizzo IP> :64297`

L'installazione termina con una panoramica delle porte attive.

```

### Please review for possible honeypot port conflicts.
### While SSH is taken care of, other services such as
### SMTP, HTTP, etc. might prevent T-Pot from starting.

Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       User        Inode      PID/Program name
tcp        0      0 0.0.0.0:64295          0.0.0.0:*               LISTEN      0           45514       8559/sshd: /usr/sbi
tcp        0      0 0.0.0.0:4222          0.0.0.0:*               LISTEN      0           45527       8559/sshd: /usr/sbi
tcp6       0      0 :::64295              :::*                    LISTEN      0           45525       8559/sshd: /usr/sbi
tcp6       0      0 :::4222               :::*                    LISTEN      0           45529       8559/sshd: /usr/sbi

### Done. Please reboot and re-connect via SSH on tcp/64295.

```

Figura 4.5: Porte attive dopo l'installazione di T-Pot

4.2 Analisi dei dati raccolti : Dashboard di T-Pot

In questa sezione si analizzano i dati raccolti da T-Pot, con annessa dashboard web dove è possibile vedere tutte le informazioni più importanti riguardo le connessioni intercettate. L'honeypot implementato in rete aziendale è stato configurato come honeypot di produzione, e quindi non è stato aperto completamente al traffico di rete, bensì è stato posto dietro al firewall, bloccando così la maggior parte degli attacchi provenienti dall'esterno. La maggioranza delle connessioni che hanno oltrepassato il firewall sono *bot* o *crawler* [43], delle tipologie di bot dotati di uno script che permette di eseguire automaticamente scan globali a una serie di indirizzi IP, per poi stilare una lista di IP con porte aperte per un successivo tentativo di infiltrazione da parte di un umano. Lo scopo di questo determinato posizionamento era, infatti, quello di proteggere le risorse di produzione e rilevare connessioni o scan provenienti direttamente dall'interno della rete, dai cosiddetti "lavoratori infedeli", oltre a rilevare eventuali attaccanti provenienti dall'esterno e che sono riusciti a oltrepassare gli altri meccanismi di difesa. Durante il periodo di servizio dell'honeypot non sono state intercettate connessioni provenienti dalla rete interna, quindi non sono stati rilevati casi di lavoratori infedeli nell'azienda.

Per quanto riguarda il caso di scan provenienti dall'esterno della rete, non sono stati rilevati accessi eseguiti, bensì solo scan di porte lasciate aperte da T-Pot stesso.

Di seguito sono riportati i dati più importanti che ha catturato l'honeypot implementato in 30 giorni di attività :

Scan totali e per giorno :

In totale, T-Pot ha rilevato quasi 1.1 milioni di connessioni e scan, con una media di circa 36500 rilevazioni al giorno.



Figura 4.6: Attacchi totali rilevati

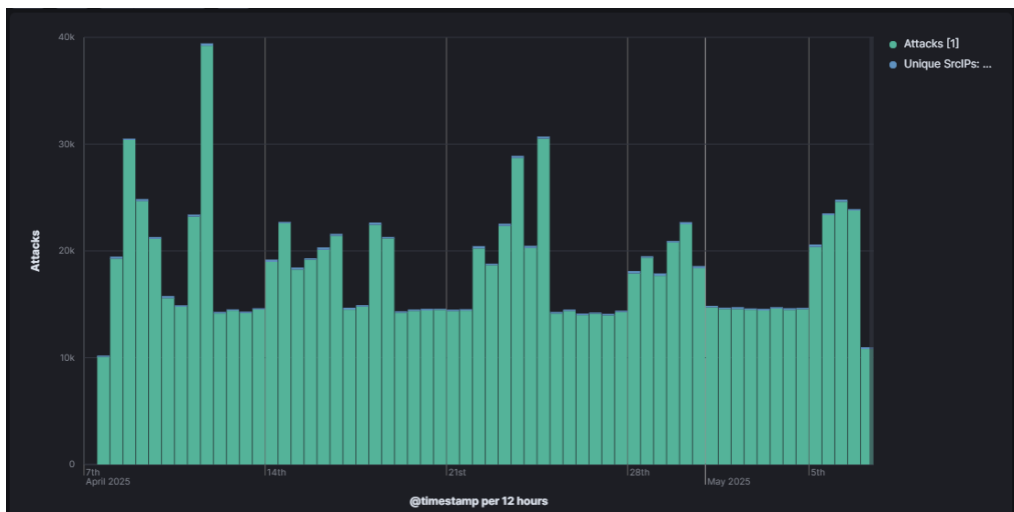


Figura 4.7: Attacchi giornalieri rilevati

Mappa degli attacchi :

Dalla seguente mappa si possono vedere le origini delle connessioni rilevate. La maggior parte proviene dagli USA, con quasi il 98% del totale.

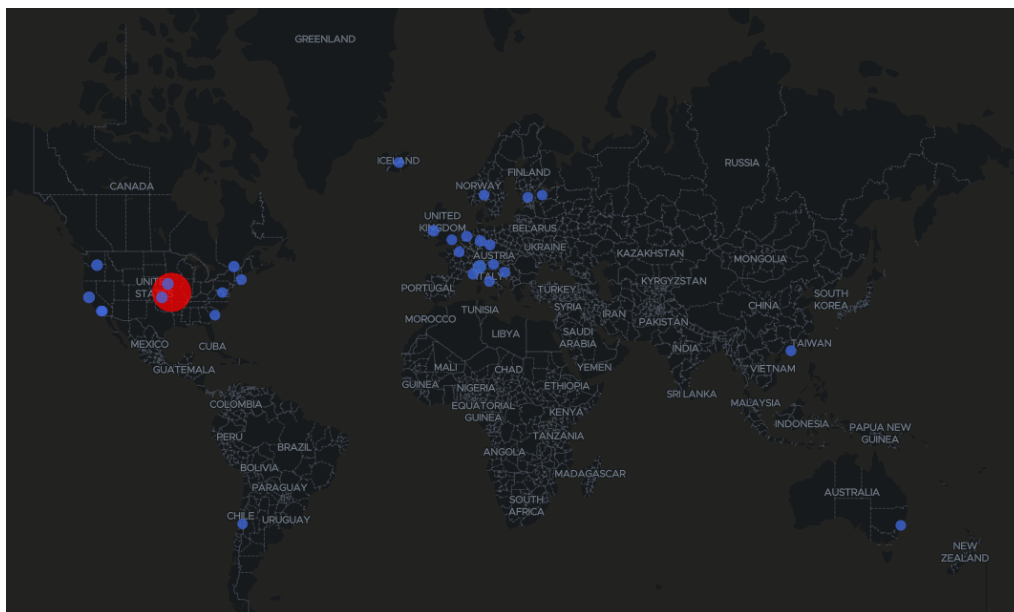


Figura 4.8: Mappa degli attacchi rilevati

Provenienza degli attacchi rilevati :

Di seguito possiamo vedere un grafico della provenienza delle connessioni rilevate. Come già detto in precedenza, la quasi totalità degli scan ha come provenienza gli USA.

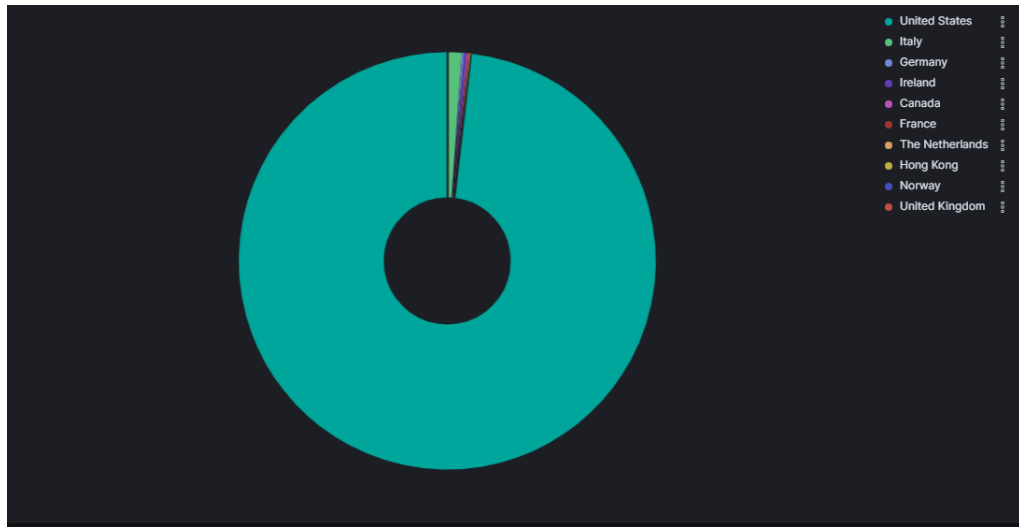


Figura 4.9: Provenienza degli attacchi

Porte maggiormente attaccate :

Infine possiamo soffermarci su una panoramica delle porte attaccate : quelle maggiormente colpite sono state la 137 [44], e la 443 [45]. La prima ha lo scopo di far comunicare più computer, utilizzando al posto degli indirizzi IP dei nomi. Inoltre serve anche per il traffico di *Server message block*, *SMB*, che viene utilizzato per il *file sharing*, per stampare in modo wireless e per altri tipi di comunicazione tra dispositivi Windows.

La seconda porta, la 443, viene utilizzata per il protocollo *Hypertext Transfer Protocol Secure*, *HTTPS*, che permette una comunicazione sicura verso l'internet. Funziona criptando i dati scambiati tra *client* e *server*, impedendo a trasgressori di intercettare e rubare informazioni importanti.

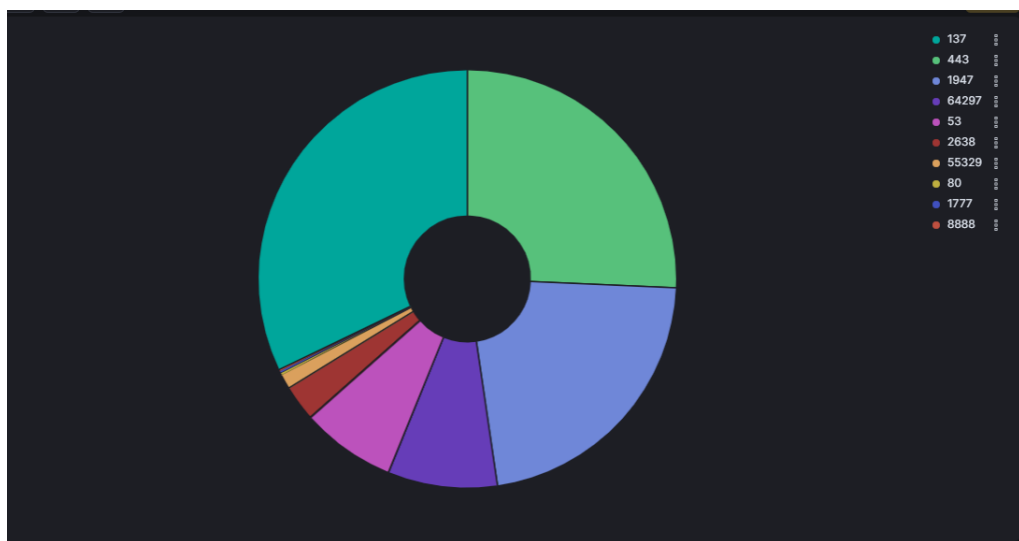


Figura 4.10: Porte maggiormente attaccate

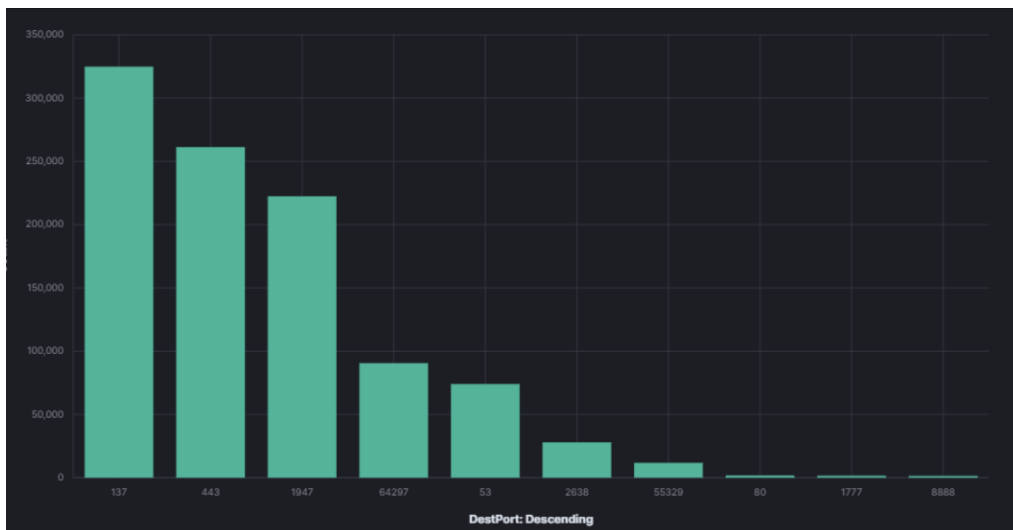


Figura 4.11: Porte maggiormente attaccate

4.3 Valutazione dell'efficacia dell' honeypot per il miglioramento della sicurezza della rete aziendale

Si può concludere che l'honeypot implementato in questo modo ha migliorato la difesa della rete aziendale, essendo di fatto un ulteriore filtro dalle connessioni in aggiunta al firewall. Lo scopo era di vedere se la rete fosse sotto attacco dall'esterno, ma soprattutto dall'interno, e si può affermare che, per ora, la rete è sicura.

In aggiunta a questo strumento, per solidificare la difesa della rete aziendale è opportuno seguire delle regole e dei piani specifici : in primo luogo è necessario creare gruppi di sicurezza per limitare l'accesso degli utenti ai servizi di cui hanno effettivo bisogno nel lavoro. In alternativa, è necessario monitorare costantemente gli accessi che vengono effettuati ai servizi aziendali. Successivamente, è opportuno avere una forte *password policy*: usare password complesse, cambiarle spesso e utilizzare il *multy factor authentication* sono delle misure di sicurezza che, se implementate, possono migliorare di molto la sicurezza della rete. Inoltre, la chiusura di porte di rete non utilizzate o non essenziali permette di ridurre l'area di attacco degli aggressori. Infine, attraverso il monitoraggio e il blocco di indirizzi IP già considerati ad alto rischio ed effettuando regolarmente aggiornamenti al nostro honeypot è possibile eliminare le vulnerabilità di attacchi conosciuti e frequenti.

Capitolo 5

Sfide e Limitazioni

5.1 Problemi etici e legali nell'uso di honeypot

5.1.1 Problemi etici

Riguardo l'etica nell'implementazione ed utilizzo di honeypot, bisogna ricordare che gli aggressori non seguono un codice etico. Tuttavia, i lavoratori nella sicurezza dovranno in molti casi aderire al codice etico della propria azienda, avendo così le mani legate.

Gli honeypot rimangono un aspetto controverso nella cybersecurity, sebbene siano visti da molti come soluzioni legali per contrastare gli attacchi informatici ai propri sistemi.

L'aspetto di incertezza per la questione etica dell'utilizzatore di honeypot è il fatto che l'uso di questi strumenti faccia in modo che gli attaccanti riescano a capire i metodi per sfruttare le vulnerabilità presenti nell'honey-pot e possano riutilizzare gli stessi metodi di attacco in reti aziendali strutturate con le stesse vulnerabilità. In un certo senso, l'uso di honeypot può far migliorare le doti di attacco degli aggressori, permettendo di attaccare altre reti meno difese che presentano le stesse vulnerabilità.

5.1.2 Problemi legali

1. *Entrapment* :

Il concetto di entrapment^[46] "prevede la concezione e la pianificazione di un reato da parte di ufficiale giudiziario, e il suo procurare la commissione del reato da parte di qualcuno che non lo avrebbe commesso se non fosse stato per l'inganno, la persuasione o la frode degli ufficiali"^[1]

¹ Wikipedia, <https://en.wikipedia.org/wiki/Entrapment>

Nell'ambito di honeypot, sarebbe reato di entrapment nel caso in cui si inviti l'attaccante ed entrare nel proprio sistema tramite messaggi o mail. Questo non è il caso, poiché l'honeytrap viene creato senza essere pubblicizzato ed è l'attaccante stesso a cercare di entrare nel sistema.

2. *Privacy* :

Per quanto riguarda il problema della privacy, ci sono due soggetti su cui concentrarci: l'attaccante e l'attaccato. Infatti l'aggressore, nel momento in cui interagisce con un honeypot, si trova il suo traffico completamente intercettato, monitorato e salvato nei file di log dell'honeytrap stesso, senza il suo consenso. Tuttavia, se l'attaccante ha diritto alla privacy, allora anche il difensore ne ha diritto; si potrebbe però pensare che l'implementazione di un sistema appositamente vulnerabile in termini di sicurezza con la speranza che venga attaccato, possa nullificare il diritto alla privacy del difensore. In realtà ciò è contestabile, poiché prima che un honeypot sia attaccato, l'aggressore deve eseguire numerosi scan sulla rete completa per trovare le componenti vulnerabili, violando a sua volta la privacy della rete del difensore.

Un altro aspetto sempre afferente al tema della privacy è la *data retention*. Quando consideriamo la conservazione dei dati ottenuti tramite honeypot, mantenerli per lunghi periodi aumenta notevolmente l'esposizione al rischio. L'eventualità di una fuga di questi dati rappresenta una seria minaccia per la sicurezza interna e per la privacy, e con il passare del tempo questa minaccia è sempre più pericolosa avendo sempre più dati salvati. Nel tentativo di aumentare lo stato di sicurezza di un sistema, in realtà lo stiamo esponendo ad una insicurezza implicita.

3. *Liability* :

Il nocciolo della questione per quanto riguarda la *liability*, cioè responsabilità legale, è la colpa: nel contesto di sicurezza, l'utilizzatore di honeypot potrebbe essere tenuto responsabile delle azioni e delle minacce lanciate dal sistema compromesso, se si è utilizzato la variante di honeypot ad alta interazione. Di fatto, in questo tipo di sistemi non è chiaro chi sarebbe responsabile dei danni subiti da una terza parte a causa degli attacchi lanciati dall'honeytrap compromesso.

5.2 Limitazioni tecniche e rischi nell' implementazione di Honeypot

Sebbene gli honeypot conferiscano un vantaggio per la sicurezza della rete, il loro utilizzo porta anche un rischio intrinseco nel caso in cui non siano stati configurati e mantenuti con rigore.

Il più grande limite di un honeypot è rappresentato dalla sua limitatezza difensiva. Infatti riesce a rilevare solo minacce e attacchi ad esso rivolti. Si capisce dunque che risulta inutile nel caso in cui l'attaccante scopra la vera natura dell'honey-pot e lo eviti. È importante quindi evitare di basare tutta l'infrastruttura di sicurezza della rete su un honeypot, ma è opportuno affiancarlo ad altri strumenti di difesa della rete, in modo da difenderla completamente.

Un altro aspetto su cui ragionare è il rischio creato dall'honey-pot per la sicurezza della rete. Da un lato infatti permette un controllo e fornisce un ulteriore livello di sicurezza alle minacce con cui interagisce, ma dall'altro con un'implementazione errata si espone una parte di rete che prima era protetta e nascosta, aumentando così il rischio di sabotaggio. L'honey-pot in questa situazione non garantirebbe la sicurezza di isolamento della minaccia e potrebbe addirittura rivelare la posizione delle risorse meno sicure. A maggior ragione se a essere stato implementato è un honeypot ad alta interazione, perché in quel caso, come abbiamo già detto, l'attaccante può compromettere il sistema e convertirlo in una macchina di attacco [19].

5.3 Soluzioni e best practices per superare queste sfide

Come detto nel capitolo precedente, ogni honeypot è in grado di rilevare solamente le minacce provenienti dagli attacchi ad esso rivolti, e quindi le informazioni raccolte da ogni singolo honeypot non possono essere utilizzate globalmente nella rete per renderla maggiormente sicura. Un modo per poter ottenere delle informazioni che abbiano valenza globale è la realizzazione di un'architettura honeypot distribuita, caratterizzata cioè dalla presenza di diversi sistemi situati fisicamente in spazi di indirizzi diversi. È però necessario sottolineare che la realizzazione di una simile soluzione deve avvenire seguendo un criterio: non basta infatti prendere degli honeypot e semplicemente distribuirli in locazioni distinte, ma è necessario seguire un certo rigore. È opportuno che essi siano divisi equamente nell'intero spazio di rete, che siano configurati in maniera omogenea e soprattutto, che vengano gestiti seguendo le stesse regole. Tutto ciò è complesso sia sotto un punto di vista prettamente tecnico che sotto quello amministrativo e di controllo. Innanzitutto, la creazione di una simile architettura è costosa e, in secondo luogo, la gestione e la raccolta dei dati vengono sensibilmente complicate. Per poter essere utilizzate, infatti, le informazioni ottenute devono essere raccolte in un luogo centrale, introducendo così nuove problematiche tecniche e di gestione.

Si possono poi seguire delle *best practises* [47] riguardanti l'implementazione e l'utilizzo di honeypot in rete aziendale :

1. Definire chiaramente l'obiettivo :

Prima di implementare e utilizzare un honeypot, è importante chiedersi in primis lo scopo dell'utilizzo di questo strumento e successivamente quali tipi di minaccia si vogliono intercettare e che livello di interazione dell'honey-pot si vuole utilizzare.

2. Scegliere il giusto livello di interazione :

Come detto nel primo punto, è necessario definire in modo opportuno l'obiettivo per cui si implementa l'honey-pot : per esempio, si sceglierà un'implementazione a bassa interazione qualora si volessero intercettare attacchi automatici provenienti da bot con un dispendio di risorse basso; invece, un'implementazione ad alta interazione è da preferirsi qualora l'obiettivo fosse la protezione da minacce più serie e il conseguente studio degli attacchi, aumentando però le risorse necessarie per l'implementazione.

3. Isolare l'honey-pot e metterlo in sicurezza :

Per ridurre il rischio di perdita del controllo dell'honey-pot, che poi comporta un susseguirsi di attacchi provenienti dall'honey-pot stesso e rivolti ad altri sistemi, è opportuno isolare il dispositivo in una sottorete creata appositamente, e utilizzare, dove possibile, la virtualizzazione, in modo da poter essere spenti e cancellati se si perdesse il controllo. Inoltre, implementare controlli di accesso in modo da far accedere solo il personale autorizzato è un buon metodo per migliorare ulteriormente la sicurezza dell'honey-pot.

4. Monitorare e analizzare le attività dell'honey-pot

La caratteristica più importante di un honeypot è la collezione di dati attraverso logs. Monitorandoli si può capire da dove provengono gli attacchi e quali sono le risorse maggiormente attaccate. Si può dunque migliorare la difesa specifica delle parti di rete più vulnerabili, e mettere in atto misure di difesa preimpostate per gli attacchi più frequenti.

Come detto in precedenza, l'implementazione e l'utilizzo di honeypot in maniera controllata darà alla rete aziendale una solidità aggiuntiva, tenendo sempre presente che questi ultimi devono essere affiancati ad altri strumenti di difesa della rete interna, in modo da ottenere una protezione completa.

Capitolo 6

Honeypot LLM

6.1 Introduzione e tipologie esistenti

Gli honeypot LLM sono una particolare categoria di honeypot evoluti che utilizzano modelli di linguaggio LLM[48] per simulare dei comportamenti e delle risposte realistiche di server o applicazioni web reali. Infatti, al loro interno non contengono delle implementazioni fisiche di quest'ultimi, bensì è presente solo il modello di linguaggio scelto che si interfacerà con l'attaccante e muterà la sua risposta in base alle richieste poste. Questa caratteristica li rende più convincenti rispetto ai tradizionali honeypot e garantisce una maggiore sicurezza non essendoci nessuna implementazione fisica, azzerando dunque la probabilità di manomissione.

Attualmente esistono diverse tipologie già implementate e testate, le seguenti sono quelle più interessanti :

ShellLM[49] :

Permette la creazione di honeypot interattivi, dinamici e realistici utilizzando gli LLM. Questo strumento fu creato per mostrare l'utilità di un file system falso e delle risposte dinamiche a seguito di comandi immessi dall'aggressore, permettendo così di ottenere una permanenza di maggior durata e di conseguenza raccogliere più dati riguardanti l'attacco in corso.

HoneyGPT[50] :

HoneyGPT è un honeypot che emula una shell utilizzando ChatGPT[51], ed è caratterizzato da una notevole efficienza di calcolo e un' interazione proattiva, utilizzando tattiche di *chain of thought* volte a migliorare la memoria a lungo termine e incrementare il grado di sicurezza. Il progetto dimostra l'abilità dell' honeypot nel bilanciare la profondità dei contenuti prodotti e la capacità di inganno dell'interlocutore. Inoltre, HoneyGPT è anche in grado di interloquire in modo coinvolgente e catturare un ampio spettro di attacchi.

6.2 Pregi e difetti

I principali vantaggi di un honeypot LLM sono i seguenti :

1. Elevato realismo nell'interazione

Infatti, un honeypot LLM è in grado di interagire con un interlocutore simulando un sistema fisico, come un server. Nell'utilizzare gli LLM, l'honey-pot adotta una grande varianza nelle sue risposte, rispondendo in modo dinamico e meno prevedibile rispetto ad un honeypot classico, rendendolo anche più difficile da smascherare.

2. Riduzione del rischio di manomissione

Come visto in precedenza, non essendoci alcun sistema fisico all'interno dell'honey-pot, non viene esposto un sistema attaccabile, bensì una virtualizzazione di tale sistema. Infatti, nel momento in cui si perdesse il controllo dell'honey-pot virtuale, basterebbe resettarlo e riportarlo ad una condizione di stabilità, riducendo il pericolo di compromissione praticamente a zero.

3. Analisi di attacchi da parte di IA

Un ultimo aspetto, da non sottovalutare, è la possibilità di analizzare gli attacchi e di distinguere quelli umani da quelli di un'intelligenza artificiale. Infatti, questo honeypot è dotato di una elevata capacità di analisi riguardo gli attacchi da parte di AI, potendo rilevare ed analizzare le richieste di agenti LLM ed identificando le varie metodologie di attacco maggiormente utilizzate, come prompt-injection[52]. Questa tipologia di attacco sfrutta l'incapacità del modello nel distinguere l'origine degli input, di fatto permettendo ad un malintenzionato di influenzare il modello LLM stesso utilizzando dei pezzi di codice o richieste all'interno dello user input.

Inoltre, è possibile capire se un attacco è *AI driven* misurando il tempo di risposta medio : una risposta in un tempo inferiore ad 1,5 secondi è da considerarsi proveniente da un agente basato su LLM.

Per quanto riguarda gli svantaggi, questi sono quelli maggiormente significativi :

1. Rischio di abuso

Infatti, se viene utilizzato un LLM troppo permissivo si rischia di dare un'arma pericolosa agli attaccanti, in quanto possono essere in grado di generare disinformazione o codici malware. È quindi molto importante supervisionare questa tipologia di honeypot ed essere pronti a resettarli, per evitare la perdita di possesso di quest'ultimo e il conseguente problema legale e giuridico.

2. Complessità tecnica

A differenza di honeypot classici, quelli caratterizzati dalla presenza di LLM sono maggiormente ardui da implementare e mantenere. Infatti, per creare quest'ultimi bisogna avere una vasta conoscenza in ambito AI, creare un sistema di logging delle varie connessioni sicuro e controllato e sorvegliare costantemente il sistema, per evitare manomissioni o *exploits*.

3. Copertura limitata dagli attacchi

Caratteristica comune tra tutti i tipi di honeypot è la limitatezza nei risultati, poiché un honeypot cattura le interazioni di ciò che gli si interfaccia. Quindi se una minaccia riesce a rilevare la posizione dell'honeypot e di conseguenza evitarlo, non si avranno dati né si saprà della minaccia presente.

6.3 Struttura di un honeypot LLM

La struttura di un honeypot LLM è visibile nell'immagine che segue.

Si può notare che l'attaccante si interfaccia con l'honeypot tramite una shell in cui digita il comando, il quale poi verrà trasmesso al LLM per costruire la risposta, che poi verrà mandata all'attaccante. L'honeypot dunque fa da ponte tra l'attaccante e il linguaggio LLM.

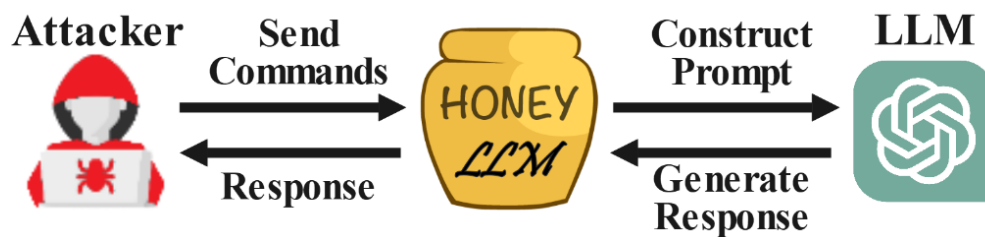


Figura 6.1: Struttura di honeypot LLM

Capitolo 7

Conclusioni e prospettive future

L'obiettivo di questa tesi è in primo luogo di introdurre la disciplina della cybersecurity e le principali tecniche e modalità di difesa della rete dalle minacce più conosciute. In seguito si è introdotta la *Deception Technology*, la quale ha lo scopo di difendere le risorse di rete attraverso l'utilizzo di esche posizionate in modo strategico nella totalità della rete. Un'applicazione pratica di questa tecnologia sono gli honeypot, spiegati nel capitolo 2 e 3, che sono degli specifici strumenti che permettono di creare un ambiente virtuale in cui gli aggressori vengono attirati e poi bloccati, mandando avvisi e fornendo i *logs* delle loro attività. Nel capitolo 3 si sono trattati i benefici ed i rischi nell'implementazione di honeypot nella rete aziendale, sottolineando come un utilizzo di honeypot in aggiunta ad altre difese della rete possa essere un ulteriore livello di difesa. Nel capitolo 4 si sono trattate le implementazioni ed il testing di due honeypot open-source, Cowrie e T-Pot. Si è potuto vedere come il primo abbia delle funzioni basiche e facili da implementare, mentre il secondo è più complesso e offre diversi honeypot, oltre che diverse possibilità di visualizzazione tramite i software installati. Si è poi affrontato il tema delle sfide legali e dei limiti di implementazione. Questi ultimi sono superabili utilizzando le *best practises* citate nel testo, mentre per quanto riguarda le sfide legali è stato visto come esse siano soggettive e come dipendano dalle leggi di ogni paese nel mondo.

In questa tesi è stato trattato l'argomento di honeypot standard e della loro implementazione, ma ci sono altre tipologie di honeypot utilizzabili. Un tipo di honeypot innovativo che sta guadagnando terreno nella sua implementazione è l'honeypot LLM[53], la cui caratteristica è l'assenza di un ambiente virtuale predefinito, sostituito da un modello di IA che risponde ad ogni richiesta inviata dall'aggressore in modo istantaneo, imitando un vero server linux e seguendo le regole imposte al momento dell'implementazione. Questo argomento merita un approfondimento che consisterebbe nella naturale continuazione di questa tesi.

Bibliografia

- [1] IBM, *Cost of a data breach*, <https://www.ibm.com/it-it/reports/data-breach>, 2024.
- [2] *Cybersecurity*, https://en.wikipedia.org/wiki/Sicurezza_informatica.
- [3] Microsoft, *Cos'è la cybersecurity?* <https://www.microsoft.com/it-it/security/business/security-101/what-is-cybersecurity>.
- [4] *Malware*, <https://en.wikipedia.org/wiki/Malware>.
- [5] *Ransomware*, <https://en.wikipedia.org/wiki/Ransomware>.
- [6] *Social Engineering*, https://en.wikipedia.org/wiki/Ingegneria_sociale.
- [7] *Phishing*, <https://en.wikipedia.org/wiki/Phishing>.
- [8] *Advanced Persistent Threats*, https://en.wikipedia.org/wiki/Advanced_Persistent_Threat.
- [9] F. digitale, *Quali sono le tre caratteristiche della sicurezza informatica?* <https://futurodigitale.infocert.it/sicurezza-in-rete/quali-sono-le-tre-caratteristiche-della-sicurezza-informatica/>.
- [10] IBM, *5 principles for cybersecurity*, https://www.youtube.com/watch?v=jq_LZ1RFPfU.
- [11] Wikipedia, *Single Point of Failure*, https://en.wikipedia.org/wiki/Single_point_of_failure.
- [12] Wikipedia, *Least Privilege*, https://en.wikipedia.org/wiki/Principle_of_least_privilege.
- [13] *Separation of duties*, https://csrc.nist.gov/glossary/term/separation_of_duty.
- [14] Z. Amos, *Is Complexity the Enemy of Security?* <https://hackernoon.com/is-complexity-the-enemy-of-security>.

- [15] ZScaler, *What is deception technologies*, <https://www.zscaler.com/resources/security-terms-glossary/what-is-deception-technology>.
- [16] *What is a honeypot?* <https://www.knowledgehut.com/blog/security/honeypot>.
- [17] L. Spitzner, *Honeypots: tracking hackers*. Addison-Wesley Longman Publishing Co., Inc., 2002.
- [18] O'Reilly, *History of Honeypots*, https://www.oreilly.com/library/view/honeypots-tracking-hackers/0321108957/0321108957_ch03lev1sec1.html?ref=blog.gitguardian.com.
- [19] D. Kelley, *The Significance of Honeypots in Cybersecurity and the Rise of Honeytokens*, <https://blog.gitguardian.com/the-significance-of-honeypots-and-the-rise-of-honeytokens/>.
- [20] S. Direct, *High Interaction Honeypots*, <https://www.sciencedirect.com/topics/computer-science/high-interaction-honeypot>.
- [21] H. Project, <https://honeynet.org>.
- [22] Snort, <https://www.snort.org>.
- [23] Cowrie, <https://github.com/cowrie/cowrie>.
- [24] T-Pot, <https://github.com/telekom-security/tpotce>.
- [25] ADBHoney, <https://github.com/huuck/ADBHoney>.
- [26] CiscoASA, https://github.com/Cymmetria/ciscoasa_honeypot.
- [27] CVE-2018-0101, <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-0101>.
- [28] Conpot, <https://github.com/mushorg/conpot>.
- [29] DDosPot, <https://github.com/aeth/ddospot>.
- [30] DDoS, https://it.wikipedia.org/wiki/Denial_of_service.
- [31] Dicompot, <https://github.com/nsmfoo/dicompot>.
- [32] Dionaea, <https://github.com/DinoTools/dionaea>.
- [33] Elasticpot, <https://gitlab.com/bontchev/elasticpot>.
- [34] Glutton, <https://github.com/mushorg/glutton>.
- [35] Heralding, <https://github.com/johnnykv/heralding>.
- [36] Honeytrap, <https://github.com/armedpot/honeytrap/>.
- [37] VMWare, www.vmware.com.

- [38] Ubuntu, <https://ubuntu.com/download/server>.
- [39] Docker, <https://www.docker.com/>.
- [40] Cowrie, *Install guide*, <https://docs.cowrie.org/en/latest/docker/README.html>.
- [41] Docker, <https://docs.docker.com/engine/install/ubuntu/>.
- [42] mRemote NG, <https://mremoteng.org/>.
- [43] *Crawler*, <https://it.wikipedia.org/wiki/Crawler>.
- [44] *Port 137*, <https://tcp-udp-ports.com/port-137.htm>.
- [45] *Port 443*, <https://tcp-udp-ports.com/port-443.htm>.
- [46] Wikipedia, *Entrapment*, <https://en.wikipedia.org/wiki/Entrapment>.
- [47] C. Nwezeaku, *Best practises for deploying honeypots*, <https://securemyorg.com/best-practices-for-deploying-honeypots-in-2025/>.
- [48] *LLM*, https://en.wikipedia.org/wiki/Large_language_model.
- [49] *ShellLM*, <https://github.com/stratosphereips/shellLM>.
- [50] *HoneyGPT*, <https://arxiv.org/abs/2406.01882>.
- [51] *ChatGPT*, <https://en.wikipedia.org/wiki/ChatGPT>.
- [52] *Prompt Injection*, https://wikipedia.org/wiki/Prompt_injection.
- [53] *Honeypot LLM*, <https://arxiv.org/html/2409.08234v1>.