Università degli Studi di Camerino

Scuola di Scienze e Tecnologie

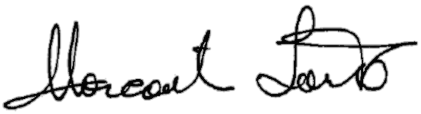
Corso di Laurea in Informatica (Classe L-31)

SECURITY OPERATION CENTER IN AMBITO DOMESTICO E NELLE PMI

Laureando
Twinkal Sikri

Matricola 119251

Relatore
Prof. Fausto Marcantoni



A.A. 2023/2024

Indice

Elenco dei codici

Elenco delle figure

Elenco delle tabelle

1. Introduzione

Nel seguente elaborato verrà illustrata la costruzione di un *Security Operation Center* (**SOC**).
Un framework molto diffuso nella sicurezza informatica, che verrà approfondito e spiegato, mediante esempi pratici, affiancati da un manuale dell'utente nella sua applicazione in contesti *non enterprise*.
Nel Capitolo 1 vengono illustrate le motivazioni e l'obiettivo della tesi.

1.1 Definizione di Security Operation Center

Un SOC è un'unità centrale, all'interno di un'organizzazione, dedicata alla sicurezza informatica, responsabile della sorveglianza, rilevamento e risposta a minacce e incidenti di sicurezza. Svolge un ruolo cruciale nella protezione delle risorse digitali, monitorando costantemente i sistemi e le reti per identificare attività sospette o dannose per poi decidere le corrispettive contromisure.

Per una definizione generale dei SOC, si veda il NIST Special Publication 800-53, che fornisce linee guida sulla gestione della sicurezza delle informazioni, inclusa la sorveglianza continua e il rilevamento delle minacce.[1]

1.2 Importanza di un SOC in ambito domestico e nelle PMI

Le funzioni principali di un SOC sono:

- Monitoraggio continuo: Il SOC opera 24 ore su 24, 7 giorni su 7, per garantire che tutte le attività di rete siano costantemente monitorate per rilevare anomalie.

- Rilevamento delle minacce: Utilizza strumenti avanzati di analisi dei dati e intelligenza artificiale per identificare potenziali attacchi informatici, come botnet e malware [2][3].

- Risposta agli incidenti: Quando viene rilevata una minaccia, il SOC è responsabile della gestione dell'incidente, che include l'analisi dell'impatto, la mitigazione della minaccia e la comunicazione con le parti interessate[4][5].

- Gestione dei log: Una parte fondamentale del lavoro del SOC è la raccolta e l'analisi dei log di sicurezza, che aiutano a identificare attività anomale e a fornire dati per audit e indagini forensi [6].

Avere uno strumento del genere, utilizzando poche risorse, è un considerevole vantaggio per chiunque abbia il desiderio di monitorare e tenere in sicurezza i propri sistemi.

Oggigiorno, possiamo notare come la digitalizzazione stia raggiungendo gli angoli più reconditi di ogni contesto e/o ambiente, a partire dalla natura, alle grandi industrie fino ad arrivare, addirittura, alle porte delle nostre case.

Con questo elaborato si vuole andare a dimostrare che non bisogna essere limitati dal concetto che alcuni strumenti siano circoscritti al contesto *enterprise*, bensì, oggi più che mai, fra le nostre mani abbiamo tutto ciò che ci serve per aggiungere dei layer di **sicurezza e robustezza** alle infrastrutture personali, senza ovviamente andare a spendere capitali interi per hardware e software.

1.3 Obiettivi della tesi

Quello che si vuole fare è sviluppare un framework operativo per un SOC basato su tecnologie open-source low-buget, capace di monitorare attraverso delle **dashboard** (cruscotti), l'intera infrastruttura di rete (una scuola, una casa, un'azienda, ecc..) sia dal punto di vista della sicurezza che dal punto di vista della manutenzione dei sistemi. (aggiornamenti, installazioni sia software che hardware)

Il fine di questo elaborato, quindi, si può articolare in quattro punti fondamentali, ciascuno mirato a esplorare aspetti critici relativi ai SOC e al loro impatto pratico.

- **Dimostrazione dell'efficacia operativa dei SOC:** Il primo obiettivo consiste nel validare empiricamente l'efficacia dei SOC nella mitigazione delle minacce informatiche. Attraverso l'analisi di casi studio, si intende evidenziare come l'adozione di un SOC riduca significativamente i rischi associati a violazioni dati o interruzioni operative. Saranno approfonditi meccanismi come il monitoraggio continuo delle reti e l'integrazione di threat intelligence, elementi chiave per contrastare attacchi avanzati come ransomware o compromissioni di botnet.

- **Analisi critica delle strategie di selezione dei SIEM:** Un secondo pilastro della ricerca riguarda la valutazione sistematica degli strumenti SIEM (Security Information and Event Management), componenti tecnologici centrali nei SOC. Sarà esaminato l'ecosistema dei vendor, da soluzioni enterprise come Splunk e IBM QRadar a piattaforme open-source come Wazuh, con focus critico su fattori decisionali: costi, scalabilità, interoperabilità con infrastrutture eterogenee, e capacità di automazione tramite machine learning. Particolare attenzione sarà dedicata alle sfide di normalizzazione dei log in ambienti multi-cloud.

- **Divulgazione del valore dei SOC per contesti non enterprise:** Il terzo obiettivo punta a sovvertire il paradigma che associa i SOC esclusivamente a grandi organizzazioni, dimostrandone l'applicabilità in scenari domestici e PMI. Attraverso l'elaborazione di framework semplificati, come l'implementazione di tool unified threat management (UTM)[7] abbinati a servizi SOC-as-a-Service, si illustrerà come anche realtà con budget limitati possano beneficiare di funzionalità avanzate: dal rilevamento di intrusioni alla gestione centralizzata delle patch.

- **Sperimentazione pratica di tool e metodologie:** L'ultimo asse della tesi prevede una campagna sperimentale su piattaforme *representative*, finalizzata a verificare prestazioni e limiti degli strumenti analizzati. In laboratorio saranno ricreate minacce simulate (es. TEST EICAR) per testare l'efficacia di configurazioni SIEM diversificate. I risultati, visualizzati tramite dashboard comparative, mirano a fornire indicazioni operative per:
  - Ottimizzazione delle regole di correlazione degli eventi;
  - Calibrazione dei sistemi di allerting per evitare fatigue da falsi positivi;
  - Integrazione di protocolli di risposta automatizzata (playbook SOAR[8]).

  Questo approccio ibrido, teorico e sperimentale, consentirà di convalidare le ipotesi di ricerca attraverso evidenze tangibili, offrendo al contempo spunti per futuri sviluppi nel campo della sicurezza proattiva in contesti domestici o nelle piccole-medie imprese.

2. Panoramica del Progetto

Il progetto nasce per dare una risposta alla domanda: *"Come posso proteggere la mia rete personale o aziendale e renderla robusta, ma con risorse limitate?"*[9][10]
In questo elaborato si esplora il mondo della cyber security, ma con una prospettiva completa e unica, ovvero combinando entrambi i lati della sicurezza: il **Red Team** e il **Blue Team** [10].
Questo tipo di prospettiva o meglio approccio, è noto come **"Purple Team"**, che sfrutta sia tecniche offensive che difensive, per fornire una comprensione olistica delle **minacce** e delle **soluzioni** di sicurezza, quindi delle contromisure o prevenzioni.
Facendo un breve excursus, possiamo definire:

- **Red Team**: il fronte dell'attacco simulato[11].
  Cercano di mettere alla prova le difese di un'organizzazione esattamente come farebbe un vero aggressore, ma con un obiettivo diverso: scoprire le **vulnerabilità** con il fine di correggerle prima che un malintenzionato riesca a sfruttarle. In questo contesto simulano i *cattivi*.

- **Blue Team**: la contrapposizione del Red Team[11].
  Spesso converge con il **SOC**, quindi lo scopo principale di questo lato è quello di gestire la fase di incidente di sicurezza, effettuando incident response (IR) e tentando di rilevare le azioni malevole del Red Team.

Quindi, visto che si vuole mostrare come ci si muove da ambo le parti, bisogna necessariamente mettere su una struttura di difesa, e poi un fronte di attacco, per andare a mettere alla prova i sistemi di contromisura adottati per le minacce rilevate.

Nei capitoli a venire viene mostrato tutto il processo di costruzione di un laboratorio dove poi sono stati testati i temi citati precedentemente e, quindi, viene dato un valore al concetto di Security Operation Center nei contesti domestici e/o nelle PMI.

Si è iniziato con il dichiarare cosa si vuole andare a simulare, quindi si è definita una *topologia* di rete, che poi è stata sottoposta a un'analisi di sicurezza e irrobustita il più possibile.

Successivamente è stato stabilito con *cosa* e *come* si è cercato di penetrare all'interno della rete monitorata e/o rilevare vulnerabilità.

Questo tipo di approccio ha creato una base empirica su cui, poi, si sono tratte delle conclusioni, in termini di utilità, per quanto concerne il framework operativo protagonista dell'esperimento.

2.1 Descrizione dell'ambiente di simulazione

L'ambiente di simulazione è stato sviluppato in una sub-net specifica dell'Ateneo che, per motivi di sicurezza, si raggiunge solo tramite autenticazione.
Quello che si è voluto andare a creare è una topologia di rete di questo tipo:

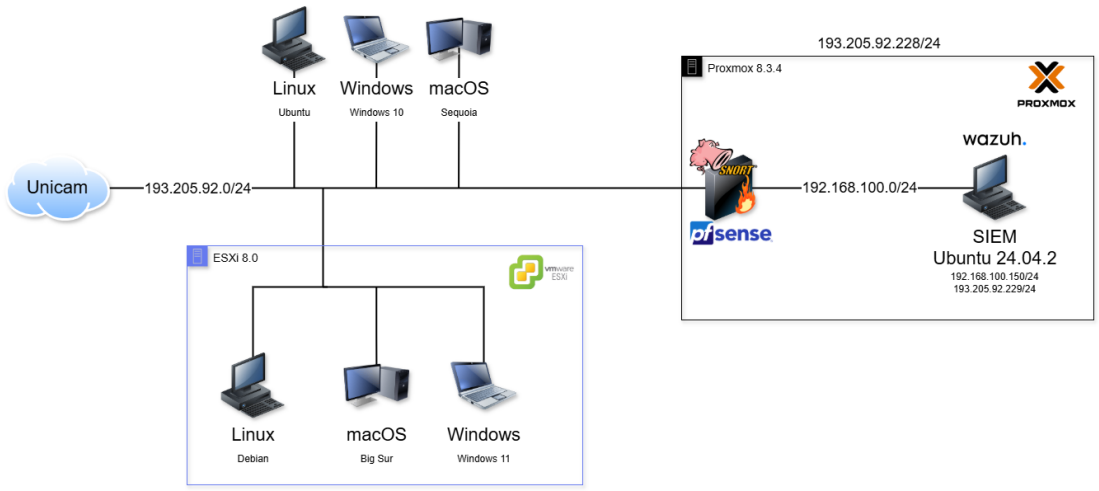


Figura 2.1: Topologia dell'ambiente di simulazione usato in laboratorio

Inizialmente, la topologia era ben diversa e addirittura coinvolgeva un'altra macchina server uguale, ma configurata diversamente in termini di RAM e storage.

La topologia è composta da:

- Ambiente virtuale **Proxmox VE** che gira su un server dedicato, con un indirizzo riservato sulla subnet 193.205.92.1/24 e contiene:
  - VM **pfSense**(vedi sez. 2.3.3): una VM su cui è installata la ISO di un firewall che regola il traffico per la macchina virtuale del SIEM.
    Ottiene un indirizzo IP passando per il server, visto che ha un'interfaccia di rete in bridge con la macchina fisica e un'altra interfaccia virtuale che andrà a costituire la LAN 192.168.100.0/24
  - VM del **SIEM**(vedi sez. 2.3.4): una VM su cui è installato Ubuntu 24.04.2 su cui gira il SIEM che sarà il cuore pulsante del SOC. Il traffico che esce o entra è regolato dal firewall pfSense.
- Ambiente virtuale **ESXi** che gira su un server dedicato riservato ai docenti, sempre con indirizzo riservato sulla subnet 193.205.92.0/24:
  - VM **Linux**: VM con Debian come sistema operativo
  - VM **MacOS**: VM con BigSur installato come sistema operativo per scopi di test come questo.
  - VM **Windows**: VM con installato Windows 11 come sistema operativo.
- Diverse **macchine fisiche** collegate alla subnet 193.205.92.0/24:
  - PC **Linux** (Ubuntu)
  - PC **Windows** (Windows 11)
  - PC **MacOS** (Sequoia)

2.2 Hardware utilizzato: HP ProLiant DL385 G7

Nella topologia di rete in figura 2.1, possiamo notare che è anche rappresentato un server che ospita ESXi 7.0 come ambiente virtuale. Questo server non verrà illustrato dettagliatamente, poichè non è stato toccato effettivamente per creare l'attuale Security Operation Center, ma è utilizzato solo per virtualizzare delle particolari macchine per vedere come si comportano in relazione al SIEM che li monitora.
Comunque basti sapere che si tratta di una macchina server Dell con processore Intel molto più recente (Dell Server rack PowerEdge R760 con IntelXeon, 383gb ram[12]) e quindi predisposta per virtualizzare dei sistemi operativi complessi come MacOS (nel nostro caso, Big Sur).

La **macchina** utilizzata, su cui si è lavorato effettivamente, è situata nella sala server del Polo A della Scuola di Scienze e Tecnologie Informatiche dell'Università degli Studi Camerino ed è collegata alla rete dell'Ateneo.

È stata scelta questa macchina poiché rappresenta l'esempio più vicino a un contesto domestico o di una PMI, in cui si potrebbe andare a utilizzare una macchina datata, o comunque, con performance e compatibilità diverse rispetto all'hardware odierno, per crearci sopra il proprio SOC personale.

In questo caso si tratta di un HP ProLiant DL385 G7, è un server rack 2U progettato per ambienti virtualizzati e carichi di lavoro enterprise, basato su architettura AMD Opteron 6100 "Magny-Cours". (risalente al 2012 ca.)
La configurazione attuale consiste in 2 processori AMD Opteron 6174 (12 core ciascuno a 2,2 GHz) e 64 GB di RAM DDR3 che offrono una piattaforma bilanciata per applicazioni ad alta densità computazionale.[13]

Figura 2.2: Server utilizzato - HP ProLiant DL385 G7 [14]

Progettato per VMware vSphere ESXi e Hyper-V, il DL385 G7 supporta fino a 128 vCPU e 1 TB di RAM virtuale grazie alla topologia NUMA a 4 domini (2 nodi per processore). La configurazione con 64 GB RAM è ideale per hosting di 15-20 VM medio-piccole con allocazione dinamica della memoria.

2.2.1 Architettura di processore e memoria

- **CPU**: 2 × AMD Opteron 6174 (12 core, 12 MB L3 cache, TDP 115W), compatibili con socket G34 e dotati di tecnologia AMD-V/RVI per l'accelerazione della virtualizzazione.
- **RAM**: 64 GB DDR3 ECC Registered (configurazione tipica: 16 × 4 GB DIMM) su 24 slot disponibili, con supporto fino a 256 GB utilizzando moduli RDIMM da 16 GB. La velocità operativa è regolata a 1333 MHz in configurazioni ottimizzate.

2.2.2 Storage e controller RAID

- **Controller integrato**: HP Smart Array P410i con 1 GB di cache FBWC (Flash-Backed Write Cache), supporto RAID 0/1/5 e interfaccia SAS 6 Gbps.
- **Supporto dischi**: Fino a 8 dischi SAS/SATA hot-swap da 2,5" o 6 LFF da 3,5", con capacità massima teorica di 16 TB (8×2 TB) nella configurazione SFF.

2.2.3 Rete e connettività

- **Schede di rete integrate**: $2 \times$ HP NC382i Dual Port (4 porte Gigabit totali) con offload TCP/IP e supporto a teaming/failover.
- **Opzioni di espansione**: 4 slot PCI-X 64-bit e 2 slot PCIe Gen2 x8, utilizzabili per schede Fibre Channel (es. 8 Gbps HBAs) o schede di rete aggiuntive.

2.3 Software e strumenti selezionati

Ritornando al discorso SOC, selezionare i software da utilizzare era un compito abbastanza semplice da fare, visto che si era preso come riferimento il magazine open-news Red Hot Cyber che aveva pubblicato un articolo dedicato[10] all'argomento e, quindi, la struttura prevista per il SOC e la rete di testing inizialmente era questa:

- Ambiente virtuale **Proxmox VE** (vedi sez. 2.3.2) *pve1*: in questo ambiente avrebbero dovuto girare le macchine che avrebbero costituito la rete da proteggere con una macchina firewall a monte.
  Il *pve1* è stato costruito su una macchina server dedicata, della tipologia indicata precedentemente (2.2).

- Ambiente virtuale **Proxmox VE** (vedi sez. 2.3.2) *pve2*: in questo ambiente avrebbe dovuto girare semplicemente il SIEM[9] (Security Information and Event Management) installato su una macchina virtuale, con le relative accortezze. Anche il *pve2* era costruito su una macchina fisica individuale (2.2). *Questa macchina, nella configurazione finale, non è stata più utilizzata.*

Quindi, una volta completato il setup di tutta la rete di laboratorio, si sarebbe potuto passare alla fase di testing e trarre delle conclusioni, ma la configurazione ha rivelato complessità superiori alle stime preliminari.

Nella sezione 2.3.1, la questione viene illustrata nel dettaglio.

2.3.1 Troubleshooting durante il setup

Una volta definita e creata una topologia all'apparenza ideale (una versione precedente alla topologia finale mostrata nella sezione 2.1), sono emersi diversi problemi relativi sia alla struttura, sia all'hardware che si stava utilizzando.

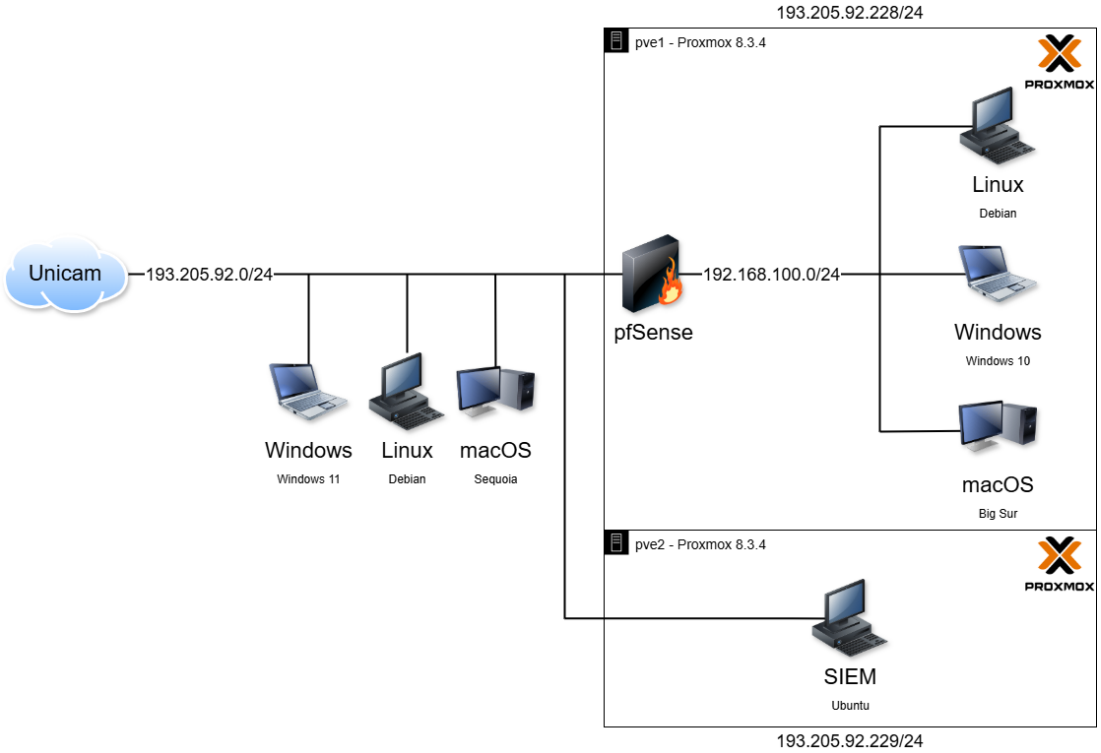


Figura 2.3: Topologia iniziale, successivamente modificata

Inizialmente, avere le VM distribuite in questo modo, ovvero con il *pve1* e il *pve2* installati su due macchine server distinte, messe poi in cluster, sembrava una buona soluzione, però poi sono venute a galla diverse questioni:

- **Storage**: lo storage del pve1 non riusciva a contenere tutte le VM che dovevano comporre la rete di test e il cluster non permetteva la condivisione dei dischi poiché non è stato possibile implementare una vSAN[15] (Virtual Storage Area Network) fra i due nodi.
- **Limiti Hardware**: anche se abbastanza performante, pur essendo datata, l'architettura e i componenti dei server utilizzati in laboratorio limita la virtualizzazione di certi sistemi operativi (come macOS o Android) e in certi casi è praticamente impossibile per via dell'architettura per cui è nativamente predisposto.
- **Connessione fra SIEM e VM**: le VM nel pve1, per comunicare con il SIEM instanziato sul pve2, avrebbero dovuto passare attraverso il firewall, il che avrebbe condotto a delle soluzioni poco pratiche o che comunque avrebbero coinvolto troppo la subnet 193.205.92.0/24:
  - **Regole NAT** sul firewall (Outbound 1:1): richiede la possibilità di poter fissare gli indirizzi IP delle VM, ma visto che si era deciso che l'assegnazione degli indirizzi delle macchine sotto il firewall dovesse essere automatica,

tramite DHCP, quest'opzione è stata scartata.
Tralasciando il fatto che poi si sarebbero dovuti fissare anche degli indirizzi IP nella rete locale 193.205.92.0/24, che sarebbero stati i Virtual IP delle VM utilizzati per comunicare con la macchina fuori dal firewall. Questo dettaglio va a rimarcare ancora di più il fatto che fosse un approccio poco pratico.

- **Proxy**: una soluzione per creare un "ponte" fra le VM e il SIEM, ma comunque è un approccio non troppo robusto e andrebbe ad aggiungere un ulteriore elemento da gestire in termini di sicurezza e manutenzione.
- **Tunnel VPN**: si sarebbe potuto fare, ma a questo punto avrebbe avuto più senso spostare la macchina SIEM direttamente sotto la LAN del firewall, visto che la logistica non era un problema, ma ovviamente le macchine test si sarebbero dovute istanziare altrove.

Per sopperire questi problemi, si è cercato di rimodellare il tutto mirando a fare un *cambiamento drastico*: cambiare la **topologia** e rinnovare una macchina server installandoci **ESXi**.

Quindi arrivati a questo punto, si è optato per ricominciare praticamente da capo, tuttavia, anche qui, un nuovo grande ostacolo si è anteposto al completamento del setup del laboratorio, dopo l'installazione di ESXi 6.5 (è stata scelta questa versione visto che il server è datato e bisognava installarne una pressoché compatibile).

Bisogna fare una premessa, l'installazione di ESXi è stata possibile solo disabilitando il controllo della CPU dal kernel, quindi bypassando lo step che effettivamente verifica la possibilità di virtualizzare delle macchine tramite questo software su una determinata architettura. Come ci si poteva aspettare, infatti, VMware ESXi non ne voleva proprio sapere di virtualizzare le macchine desiderate, un' incognita dettata dall'incompatibilità HW/SW.

Detto questo, si può evincere chiaramente che la macchina server abbia posto innumerevoli limiti hardware, per quanto concerne la compatibilità con i vari software, ma questo è proprio uno dei **principi** su cui verte questo elaborato: cercare la soluzione più adatta, ma sempre usando le minor risorse possibili e sfruttando al meglio ciò che si ha.
Dopo l'ennesimo fallimento, il focus è diventato quello di far diventare l'ambiente di virtualizzazione il più solido possibile e proprio per questo non si sono più usate due macchine server separate, bensì sono stati spostati i dischi rigidi dal server dove c'era il pve2, al server "principale", ovvero quello dove girava il pve1, poiché provvisto di più memoria RAM (ben 64GB contro i 16GB dell'altra), ma necessitava più spazio di archiviazione. Quindi si è ricominciato con una configurazione ex-novo di Proxmox, con molte meno macchine, ma principalmente:

- Una VM con installato un firewall
- Una VM Linux che ospita il SIEM

Quindi si è ottenuto un singolo *pve* che ospita le VM necessarie. In questo modo, il firewall e il cuore del SOC possono lavorare molto più liberamente e sfruttare le risorse della macchina fisica al massimo, mentre le VMs che faranno da cavie, e che quindi il SIEM dovrà monitorare, sono state create su una macchina fisica diversa, che ha ESXi come ambiente di virtualizzazione.
Quest'ultima macchina è stata già citata in precedenza, si tratta di un server Dell di ultima generazione con processore Intel che permette la virtualizzazione di praticamente tutti i sistemi operativi.

Dopo che l'ambiente di virtualizzazione era pronto e la topologia della rete di laboratorio era stata ultimata, finalmente si è passati alla configurazione del firewall e dei componenti del SOC.

2.3.2 La scelta del Virtual Environment

Proxmox VE (Virtual Environment) è una piattaforma open-source per la virtualizzazione e la gestione di infrastrutture IT. Offre una soluzione completa per la creazione e la gestione di macchine virtuali (VM) e container, supportando sia KVM che LXC. Proxmox è noto per la sua facilità d'uso, scalabilità e flessibilità, rendendolo popolare tra le piccole e medie imprese, nonché tra gli utenti domestici.

Ha diversi vantaggi:

- **Costo**: Essendo open-source, Proxmox è gratuito, il che lo rende una scelta economica rispetto alle soluzioni proprietarie.
- **Facilità d'uso**: L'interfaccia web di Proxmox è intuitiva e facile da gestire, anche per gli utenti meno esperti.
- **Scalabilità**: Supporta cluster di server per garantire alta disponibilità e scalabilità.
- **Supporto hardware**: Ampio supporto per hardware diversificato, inclusi server e storage. Utilizzando Proxmox, possiamo gestire efficientemente diverse macchine virtuali e container con risorse hardware limitate, ottenendo prestazioni elevate grazie alla virtualizzazione di livello uno. (Bare-Metal Hypervisor)

Per avere un'idea ancora più chiara del perché è stato scelto Proxmox, possiamo andarlo a confrontare con **VMware vSphere ESXi**. Si tratta di una soluzione di virtualizzazione enterprise leader nel mercato, nota per la sua stabilità e le funzionalità avanzate. ESXi è una scelta comune per le grandi organizzazioni che richiedono elevate prestazioni e sicurezza.

Caratteristica	**Proxmox VE**	**VMware ESXi Enterprise**
Costo	Gratuito	Costoso (licenze necessarie)
Facilità d'uso	Interfaccia web intuitiva	Interfaccia avanzata, richiede competenze specifiche
Scalabilità	Supporto cluster integrato	Alta scalabilità con vCenter
Supporto hardware	Compatibile con hardware generico, funziona anche con risorse limitate	Supporto hardware esteso, ottimizzato per dispositivi certificati VMware
Sicurezza	Funzionalità di base affidabili	Soluzioni di sicurezza avanzate (es. NSX)
Assistenza	Community e supporto opzionale	Supporto professionale VMware
Funzionalità	Essenziali ma complete	Funzionalità enterprise come vMotion, HA, DRS

Tabella 2.1: Tabella di confronto fra Proxmox ed ESXi

2.3.3 La scelta del Firewall

Nel contesto di questo progetto, **pfSense** è stato selezionato come firewall principale per la sua maturità, scalabilità, facilità di personalizzazione e integrazione con altri strumenti di sicurezza. La sua capacità di supportare configurazioni avanzate, come VPN e IDS/IPS, lo rende ideale per creare un'architettura di sicurezza robusta e adattabile. Inoltre, la sua interfaccia web intuitiva facilita la gestione anche per gli utenti meno esperti, mentre la comunità attiva garantisce supporto continuo e aggiornamenti regolari. Sebbene, ad esempio, OPNsense offra una semplicità maggiore, la maturità e la versatilità di pfSense lo rendono la scelta ottimale per ambienti che richiedono personalizzazioni avanzate. Di seguito, una classifica e un confronto tra i principali firewall open-source, con particolare attenzione a pfSense e OPNsense visto che possono essere i contendenti principali in una configurazione come quella del progetto.

Panoramica su alcuni Firewall Open-Source

pfSense: Un firewall open-source basato su FreeBSD, noto per la sua stabilità e facilità d'uso.

OPNsense: Un altro firewall open-source basato su FreeBSD, derivato da pfSense, con un focus maggiore sulla semplicità e l'usabilità.

Iptables: Un firewall open-source per Linux, molto versatile ma richiede competenze tecniche avanzate.

UFW (Uncomplicated Firewall): Un frontend semplice per iptables, facile da usare ma meno potente rispetto a pfSense.

Firewall	Licenza	Costo	Integrabilità	Funzionalità Principali
pfSense	BSD	Gratuito	API, plugin per HAProxy, Squid, VPN, integrabile con Wazuh	Firewall stateful, VPN, routing avanzato, IDS/IPS, load balancing
OPNsense	BSD	Gratuito	API, plugin per HAProxy, Squid, VPN, integrabile con Wazuh	Firewall stateful, VPN, routing avanzato, IDS/IPS, load balancing, interfaccia utente migliorata
Iptables	GPL	Gratuito	Configurazione manuale, integrabile con Linux	Regole firewall personalizzabili, supporto per NAT e routing
UFW	GPL	Gratuito	Facile integrazione con Ubuntu, semplice da configurare	Regole firewall semplici, facile da usare, supporto per IPv6

Tabella 2.2: Tabella di confronto fra vari firewall open-source

pfSense vs. OPNsense

Entrambi sono basati su FreeBSD e offrono funzionalità simili, ma OPNsense è derivato da pfSense e si concentra maggiormente sulla semplicità e l'usabilità. pfSense è più maturo e offre un supporto più ampio per le configurazioni avanzate. Inoltre, pfSense consente uno spazio di manovra molto più grande, mentre OPNsense è limitante per quanto concerne, ad esempio, le regole di controllo del traffico in entrata ed uscita: OPNsense ha delle regole auto-generate che non si possono modificare che, però, a volte forzano l'utente a dover rimodellare tutto, mentre pfSense non ha questo tipo di problematica e il configuratore può muoversi liberamente.
Nel contesto del progetto, si sono verificate varie incompatibilità con il SIEM che hanno fatto cambiare il firewall diverse volte, passando fra diversi software e versioni degli stessi, ma alla fine **pfSense-CE-2.7.2** ha consentito la realizzazione della configurazione desiderata.

2.3.4 La scelta del SIEM

Cominciamo dal ricordare il paradigma fondamentale di questo progetto: creare un SOC che sia funzionale, ma con risorse limitate. Per rispettare questa esigenza, sono state selezionate le opzioni più accessibili che garantiscono al contempo efficacia operativa, senza ricorrere a costosi applicativi commerciali o a significativi upgrade dell'infrastruttura tecnologica esistente.

Un **SIEM** rappresenta il *"cuore pulsante"* di un **SOC**, poiché svolge un ruolo cruciale nella raccolta, analisi e gestione degli eventi di sicurezza. Grazie alla sua capacità di aggregare e correlare log provenienti da diverse fonti all'interno dell'infrastruttura IT (firewall, server, dispositivi di rete), il SIEM offre una visibilità completa sulle attività di rete, consentendo di identificare potenziali minacce o anomalie che potrebbero passare inosservate se analizzate singolarmente. Inoltre, il SIEM genera alert e notifiche che facilitano una risposta rapida e coordinata agli incidenti di sicurezza, integrandosi con altri strumenti di sicurezza per creare un'architettura difensiva completa. Un aspetto non trascurabile è anche il supporto offerto per soddisfare requisiti normativi e di conformità attraverso report dettagliati sugli eventi di sicurezza e accessi autorizzati.

Di seguito si presenta una tabella comparativa dei principali SIEM disponibili sul mercato, con un'analisi dettagliata delle loro caratteristiche e funzionalità.

SIEM	Tipo	Sistema Operativo	Categoria	Funzionalità principali	Plugin/Estensioni	Assistenza e Integrazione
AlienVault OSSIM[16][17]	On-premises	Linux	Open-source	Asset discovery, correlazione eventi, intrusion detection	Open Threat Exchange (OTX), Suricata, Nagios, OSSEC, OpenVAS	Community attiva, documentazione dettagliata
Elastic SIEM[18][17]	On-premises/ Cloud	Windows/Linux/ macOS	Open-source	Scalabilità elevata, analisi log avanzata, visualizzazione dati	Beats (Filebeat, Metricbeat), Logstash, Kibana, Elastic Agent	Supporto Elastic (community e commerciale), ampia documentazione
Wazuh[19][17]	On-premises/ Cloud	Linux (server), Multi-OS (agenti)	Open-source	Rilevamento intrusioni, gestione log, compliance GDPR/NIST/PCI-DSS	TheHive, Cortex, VirusTotal, MITRE ATT&CK, Suricata integration	Community attiva, supporto collaborativo e commerciale opzionale
SIEMonster[20][17]	On-premises/ Cloud	Linux/Amazon Linux	Open-source	Personalizzazione avanzata, dashboard configurabili, rilevamento minacce avanzate	ELK Stack (ElasticSearch/Kibana), API REST personalizzabili	Supporto tramite community e forum dedicati
Security Onion[21][17]	On-premises/ Cloud	Linux	Open-source	Monitoraggio rete, rilevamento intrusioni (IDS/IPS), analisi traffico di rete	Suricata, Zeek (Bro IDS), Wazuh integration, NetworkMiner	Documentazione dettagliata, community specializzata
ELK Stack[22][17]	On-premises/ Cloud-native	Vari (configurabile)	Open-source	Analisi log, visualizzazione dati personalizzata, scalabilità elevata	Kibana, Logstash, Beats (Filebeat/-Metricbeat), Elastic Agent plugins personalizzati	Supporto Elastic (community e premium)
ManageEngine Log360[23][17]	On-premises/ Cloud-native	Windows/ Linux/ macOS	Enterprise	Feed di intelligence preconfigurati	Active Directory integration, SYSLOG connectors avanzati	Assistenza 24/7 commerciale premium
Exabeam[24][17]	Cloud-native	Vari (configurabile)	Enterprise	UEBA, automazione TDIR	Estensioni per analisi comportamentale	Supporto premium, training e consulenza
Splunk[25][26]	On-premises/ Cloud	Windows/ Linux/ macOS	Enterprise	Machine learning, dashboard personalizzabili, analisi predittiva	Splunk ES, Splunk UBA, Splunk Phantom (SOAR)	Supporto commerciale, community e training estesi
Microsoft Sentinel[27][26]	Azure-integrated	Cloud (Azure)	Enterprise	Intelligenza artificiale per le minacce, analisi predittiva	Connettori Azure, Microsoft 365 Defender, Azure AD	Supporto Microsoft, aggiornamenti continui, integrazione nativa con ecosistema Azure
LogRhythm[28][26][17]	Cloud/ On-premises	Windows/ Linux	Enterprise	Analisi avanzata delle minacce, UEBA integrato	LogRhythm NDR, LogRhythm UEBA, SmartResponse automation	Supporto tecnico dedicato, formazione e assistenza 24/7

Tabella 2.3: Tabella comparativa dettagliata dei principali SIEM

In questo panorama tecnologico, **Wazuh** emerge come soluzione ideale per il contesto specifico dell'elaborato. Si distingue come piattaforma SIEM open-source particolarmente adatta per ambienti con vincoli di budget come piccole-medie imprese o un SOC domestico con risorse limitate.

La scelta di Wazuh come **nucleo** del SOC deriva da un'attenta valutazione del rapporto costo-efficacia e dalla sua eccellente capacità di integrazione. A differenza delle soluzioni proprietarie che richiedono investimenti considerevoli, Wazuh consente di implementare un sistema di difesa robusto mantenendo i costi operativi al minimo, senza sacrificare funzionalità essenziali. La piattaforma si distingue per la sua architettura modulare che permette di personalizzare regole di correlazione e flussi di lavoro in base alle specificità dell'infrastruttura target.

Analisi critica tramite confronto con alcuni SIEM

- **Wazuh vs. ELK Stack**[29]:
  - **Licenza e Costi**: Entrambi sono classificati come open-source, ma con differenze significative. Wazuh è rilasciato completamente sotto licenza GPLv2, garantendo libertà totale di utilizzo e modifica. ELK Stack, d'altra parte, ha adottato una licenza Elastic più restrittiva dal 2021, con funzionalità premium riservate alle versioni a pagamento.
  - **Integrazione SOAR**: Wazuh offre integrazioni native con piattaforme SOAR come TheHive e Cortex, essenziali per l'orchestrazione automatizzata della risposta agli incidenti. ELK richiede configurazioni complesse e personalizzazioni per ottenere funzionalità equivalenti.
  - **Orientamento funzionale**: Wazuh è progettato specificamente per la sicurezza, con supporto integrato per framework come MITRE ATT&CK. ELK è una piattaforma generalista di analisi dati che richiede configurazioni aggiuntive per specializzarla in ambito security.
  - **Risorse computazionali**: Wazuh richiede significativamente meno risorse hardware rispetto alla stack ELK completa, un fattore critico in ambienti con risorse limitate.
- **Wazuh vs. Soluzioni Enterprise (Splunk/QRadar/Sentinel)**[29]:
  - **Modello economico**: Le soluzioni enterprise adottano modelli di pricing basati su volume o abbonamento che possono raggiungere rapidamente migliaia di euro annui. Wazuh elimina completamente questi costi, rendendo sostenibile il SOC anche per budget limitati.
  - **Personalizzazione**: La natura open-source di Wazuh permette modifiche a livello di codice per integrazioni specifiche (come l'integrazione con pfSense implementata in questo progetto), mentre le soluzioni proprietarie limitano le personalizzazioni alle API disponibili.
  - **Curva di apprendimento**: Splunk, QRadar e Sentinel offrono funzionalità avanzate di analytics predittivo e UEBA (User and Entity Behavior Analytics), ma richiedono competenze specialistiche più costose e difficili da acquisire. Wazuh ha una curva di apprendimento più accessibile.
  - **Lock-in tecnologico**: Le soluzioni enterprise spesso creano dipendenza dall'ecosistema del vendor, mentre Wazuh garantisce flessibilità e indipendenza tecnologica.

- **Wazuh vs. Security Onion**[29]:
  - **Focus operativo**: Security Onion eccelle nel monitoraggio di rete e nell'analisi del traffico (IDS/packet capture), mentre Wazuh privilegia il monitoraggio degli endpoint e l'integrità dei file. La combinazione complementare dei due crea una difesa stratificata ideale.
  - **Architettura**: Wazuh adotta un'architettura client-server più leggera e scalabile, mentre Security Onion è più monolitico e richiede maggiori risorse per un deployment completo.
  - **Integrazioni**: Recenti versioni di Security Onion incorporano Wazuh come componente, suggerendo la possibilità di una futura evoluzione verso un deployment ibrido per massimizzare i vantaggi di entrambi.

Motivazioni della Scelta di Wazuh nel Progetto

Wazuh è stato selezionato come nucleo del SOC per diverse ragioni tecniche e pratiche che lo rendono particolarmente adatto al nostro contesto:

- **Integrazione con infrastruttura esistente**: I log del firewall perimetrale (formato RFC 3164) vengono ingestati direttamente in Wazuh tramite agente dedicato, permettendo correlazione in tempo reale tra alert di rete e attività sugli endpoint. Questa integrazione è stata implementata senza costi aggiuntivi grazie alle API native.
- **Automazione SOAR efficiente**: L'ecosistema Wazuh-TheHive-Cortex abilita workflow automatizzati di risposta agli incidenti, come l'analisi dinamica di file sospetti tramite VirusTotal e la quarantena automatizzata di endpoint compromessi, riducendo drasticamente i tempi di risposta.
- **Supporto nativo per conformità**: Wazuh include moduli preconfigurali per standard normativi come GDPR, NIST 800-53 e PCI-DSS, facilitando il processo di audit e compliance senza costi aggiuntivi per consulenze specialistiche.
- **Community attiva**: A differenza di soluzioni proprietarie che richiedono costosi contratti di supporto, Wazuh beneficia di una community globale attiva, con documentazione dettagliata e forum di supporto che facilitano la risoluzione autonoma di problemi complessi.
- **Requisiti hardware accessibili**: La configurazione base di Wazuh può operare efficacemente su hardware modesto (4 CPU, 8GB RAM), rendendo possibile l'implementazione anche su infrastrutture limitate o virtualizzate, con possibilità di scalare verticalmente in futuro se necessario.
- **Maturità della soluzione**: Con oltre 10 milioni di download, Wazuh ha raggiunto un livello di maturità e stabilità paragonabile a soluzioni commerciali, ma senza i relativi costi, rappresentando il compromesso ideale tra affidabilità ed economicità per il nostro progetto.

3. Installazione e configurazione dell'infrastruttura virtuale

In questo capitolo verranno illustrati tutti i passaggi fatti per andare a creare l'infrastruttura virtuale. Nel nostro caso si tratta del setup di Proxmox VE e la creazione delle VM utilizzate per formare il SOC.

3.1 Installazione di Proxmox VE

L'installazione e la configurazione di Proxmox è stata la prima cosa fatta durante la costruzione del SOC e si è evoluta in diverse fasi, descritte nei paragrafi successivi.

3.1.1 Preparazione del supporto di installazione

La prima cosa da fare è stata preparare un flash drive USB-A configurata come USB avviabile con dentro l'immagine ISO di Proxmox 8.3.4.
L'ISO si può reperire direttamente dalla pagina ufficiale di Proxmox[30] e, una volta scaricata, si può prendere una pen drive qualsiasi (si consiglia una chiavetta con una capcacità di lettura/scrittura abbastanza rapida per permette un caricamento della ISO più veloce nella fase di avvio dell'interfaccia di installazione) e successivamente, mediante tool come Rufus o Balena Etcher, si va a scrivere l'immagine desiderata sul supporto removibile, così da renderlo avviabile.

3.1.2 Configurazione del BIOS e del kernel

Nel caso specifico dell'architettura utilizzata in questo progetto, descritta nella sezione 2.2, è stato necessario modificare alcuni parametri nel GRUB. Una volta che si fa il boot tramite la chiavetta usb, si aspetta il caricamento dell'interfaccia di setup, dove si può selezionare la modalità con cui si vuole fare l'installazione (Tramite GUI, tramite terminal UI, Debug GUI ecc..). Per l'incompatibilità dei setting dell'installazione di default e la macchina su cui si sta facendo l'installazione (2.2, sono stati modificate le stringhe di avvio nel GRUB, a cui si può accedere nella schermata di installazione. È stato un rompicapo difficile da risolvere poiché non si riusciva a capire da cosa derivasse l'errore, ma alla fine il problme era il seguente: la scheda grafica del server con uscita VGA non permetteva la visualizzazione della schermata del setup dell'installazione (errore di "Frequenza non supportata" in modalità installazione tramite GUI oppure schermo nero nel caso dell'instllazione tramite Terminal UI).

Ecco come si è risolta la questione: dopo aver spostato la selezione su "installazione tramite Terminal UI", si è fatto l'accesso alla directory '/etc/default/grub' (con Ctrl+X o F10) e si modificata la stringa:

```
linux /boot/vmlinux... root=... quiet splash=silent ..
```

eliminando 'quiet' e 'splash=silent' e scrivendo:

```
linux /boot/vmlinux... root=... nomodeset proxtui vga=791
```

Per arrivare a questa soluzione sono state fatte svariate prove ed è stata consultata la wiki di Proxmox[31].

3.1.3 Procedura di installazione e setup passo-passo

In questa sezione verranno illustrati tutti i passaggi per arrivare alla configurazione finale utilizzata nel progetto.

1. **Scelta storage**: il primo step è scegliere il target dell'installazione e quindi si sceglie il disco desiderato. Nel nostro caso è lo storage della macchina server configurato RAID 5[32]

2. **Scelta area geografica, fascia oraria e layout tastiera**: impostare correttamente l'area geografica e la fascia oraria è uno step molto importante per quanto riguarda la continuità dei dati nell'infrastruttura (e.g. logging).

3. **Scelta password e username (admin)**: impostare password e email per il successivo login tramite web UI.

4. **Configurazione della rete**: impostare interfaccia di rete preferita, successivamente dare un nome all'host e infine definire:

   - Indirizzo IP e subnet mask: nel nostro caso è stato riservato un indirizzo IP nella subnet dell'Ateneo (vedi figura 2.1) per poter avere un indirizzo IP fisso.
   - Gateway
   - Server DNS

5. Revisione finale delle impostazini e successiva conferma per l'installazione

6. Arrivati a questo punto si può accedere all'interfaccia web dell'ambiente virtuale. Si trova all'indirizzo: https://"IP-SERVER":8006.

7. Una volta fatto l'accesso con le credenziali scelte prima, si può notare un avviso popup: "*You do not have a valid subscription for this server*". Per poter togliere questo pop-up, bisogna accedere alla shell della macchina, anche tramite la Web UI e bisogna usare questo script[33]:

```
sed -Ezi.bak "s/(function\(orig_cmd\) \{)/\1\n\torig_cmd\();\n\treturn;/g"
/usr/share/javascript/proxmox-widget-toolkit/proxmoxlib.js && systemctl
restart pveproxy.service
```

Codice 3.1: Script disattivazione popup

Una volta eseguito lo script, svuotare la cache del browser se si sta usando la Web UI. (quest'operazione è eventualmente reversibile)

8. **Update di apt**: per poter aggiornare i pacchetti, visto che si vuole utilizzare la versione gratuita di Proxmox e quindi non si dispone di un abbonamento, bisogna disabilitare i repository enterprise sulla macchina:

   - Se non hai una licenza enterprise, devi rimuovere il repository enterprise e aggiungere quello no-subscription.
   - **Rimuovere il repository enterprise**:
     (a) **Aprire il file di configurazione del repository di Proxmox**:
     ```
     nano /etc/apt/sources.list.d/pve-enterprise.list
     ```
     (b) **Commentare la riga (aggiungendo # all'inizio)**:
     ```
     deb https://enterprise.proxmox.com/debian/pve bookworm pve-enterprise
     ```
     (c) **Aprire il file dei repository enterprise di Ceph**:
     ```
     nano /etc/apt/sources.list.d/ceph.list
     ```
     (d) **Commentare la riga enterprise mettendo # all'inizio**:
     ```
     deb https://enterprise.proxmox.com/debian/ceph-quincy bookworm enterprise
     ```
   - **Inserire il Repository No-Subscription**:
     – **Stringa da utilizzare**:
     ```
     echo deb http://download.proxmox.com/debian/pve bookworm
     pve-no-subscription > /etc/apt/sources.list.d/pve-no-subscription.list
     ```
   - **Aggiornare i pacchetti**:
   ```
   apt-get update && apt-get dist-upgrade -y
   ```
   - Dopo l'aggiornamento, a volte, potrebbe esserci l'eventualità in cui bisogna eseguire nuovamente lo script per disabilitare il pop-up mostrato precedentemente.

Dopo questa configurazione, si passa alla creazione delle macchine virtuali.

3.2 Gestione delle macchine virtuali

La creazione delle VM è stata una delle fasi cruciali della struttura, visto che gli endpoint da monitorare avrebbero dovuto avere diversi sistemi operatvi, ognuno di loro ha le sue caratteristiche specifiche e in alcuni casi è risultato un processo complesso e delicato.

3.2.1 Definizione di QEMU in Proxmox VE

QEMU (Quick Emulator)[34] è un emulatore di macchine virtuali open-source che permette la virtualizzazione completa (full virtualization) di hardware, consentendo l'esecuzione di sistemi operativi non modificati su diverse architetture hardware. In Proxmox VE, QEMU è integrato con **KVM (Kernel-based Virtual Machine)**[35] per fornire virtualizzazione ad alte prestazioni, sfruttando le estensioni di virtualizzazione hardware (Intel VT/AMD-V) dei processori moderni.

Caratteristiche chiave di QEMU in Proxmox

- Supporta oltre 100 architetture guest (x86, ARM, RISC-V, ecc.).
- Emula dispositivi hardware virtuali (schede di rete, GPU, storage).
- Abilita funzionalità avanzate come live migration, snapshot e backup.

Proxmox VE utilizza QEMU/KVM per la gestione delle macchine virtuali (VM).

Nel *pve*, per creare una VM, innanzitutto bisogna scegliere la modalità di caricamento delle ISO desiderate:

- Upload manuale della ISO scelta nello storage.
- Download della ISO tramite Web GUI della piattaforma.
- Selezione di uno dei template disponibili nella lista.

Di seguito le fasi principali.

3.2.2 Creazione e configurazione delle VM

Web UI

- Selezionare "Create VM" dalla dashboard.
- Configurare parametri:

Esempio CLI

Dalla shell del pve:

```
1   qm create 100 --name "VM_Test" --memory 2048 --cores 2 --net0 virtio,bridge=vmbr0
```

Parametro	Descrizione
Nome e ID	Identificativo unico per la VM (es. vm-100)
Sistema Operativo	Selezionare ISO o template (da storage configurato)
CPU e memoria	Assegnare core virtuali (vCPU) e RAM
Disco	Scegliere storage locale o condiviso (Ceph, NFS, ZFS)
Rete	Configurare bridge virtuale o VLAN

Tabella 3.1: Configurazione VM

3.2.3 Gestione delle risorse e storage

La gestione delle risorse e dello storage in Proxmox VE è strettamente legata alle caratteristiche hardware del server fisico e alla tipologia di macchina virtuale (VM) che si intende creare, in funzione dell'utilizzo previsto. Proxmox VE, grazie all'integrazione di QEMU/KVM, consente di ottimizzare l'allocazione delle risorse hardware (CPU, RAM, storage) per garantire prestazioni elevate e un utilizzo efficiente dell'infrastruttura.

Gestione delle risorse

La creazione di una VM richiede una pianificazione attenta delle risorse disponibili[36]:

- **CPU**: La quantità di core virtuali assegnati dipende dal carico computazionale previsto. Per applicazioni leggere (es. server web), è sufficiente un numero ridotto di vCPU, mentre per carichi intensivi (es. database), è necessario un numero maggiore.
- **RAM**: La memoria allocata deve essere proporzionale ai requisiti del sistema operativo guest e delle applicazioni. Proxmox supporta la funzionalità di ballooning, che consente di ridistribuire dinamicamente la RAM tra le VM per ottimizzare l'utilizzo.
- **Storage**: Proxmox permette di configurare diversi tipi di storage (locale, Ceph, NFS, ZFS), scegliendo il più adatto in base alla necessità di prestazioni o ridondanza. Ad esempio, per VM critiche è consigliabile uno storage distribuito come Ceph per garantire alta disponibilità.

Tipologia di VM e utilizzo

La configurazione delle risorse varia a seconda dell'uso della VM:

- **Server applicativo**: Richiede CPU e RAM bilanciate con storage affidabile per garantire tempi di risposta rapidi.
- **Database**: Necessita di storage ad alte prestazioni (es. SSD) e maggiore RAM per gestire query intensive.
- **Test/Dev Environment**: Può utilizzare risorse condivise con configurazioni meno stringenti grazie alla flessibilità della virtualizzazione.

Ottimizzazione dello storage

Proxmox VE supporta funzionalità avanzate per ottimizzare lo storage[34]:

- **Thin Provisioning**: Consente di allocare solo lo spazio effettivamente utilizzato dalla VM, riducendo gli sprechi.
- **Snapshot e Backup**: Gli snapshot permettono di salvare lo stato della VM in modo rapido, mentre i backup incrementali minimizzano l'uso dello spazio su disco.
- **Live Migration**: Le VM possono essere migrate tra nodi senza downtime, garantendo continuità operativa anche durante interventi di manutenzione.

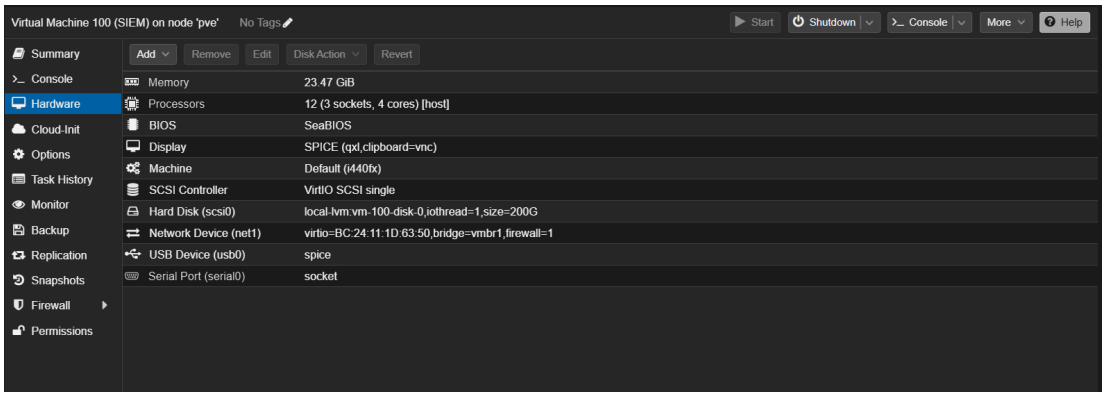


Figura 3.1: Esempio di configurazione di una VM su Proxmox

3.3 Setup del firewall

Il firewall utilizzato in questo progetto è pfSense, come spiegato nella sezione 2.3.3 (vedi fig. 2.1)

3.3.1 Configurazione di un firewall

Dopo l'installazione, si è passati alla configurazione. Fase in cui, principalmente, si sono configurate le interfacce di rete dato che il firewall è il ponte verso l'esterno di tutti i dispositivi che sono nella sua subnet.
Le interfacce sono state assegnate in questo modo:

- **WAN**: interfaccia virtuale (vtnet0), creata su Proxmox, che mette in collegamento la VM con l'ambiente virtuale e la sua scheda di rete fisica (macchina server), nella subnet 193.205.92.0/24
- **LAN**: interfaccia virtuale (vtnet1), creata su Proxmox, che mette in collegamento tutti i dispositivi che devono essere monitorati dal firewall, creando la subnet 192.168.100.0/24

Dopodichè, questo tipo di configurazione, consente di accedere alla Web UI di pfSense tramite l'inidirizzo 192.168.100.1. Ovviamente, senza l'utilizzo di Virtual IP e regole specifiche, l'interfaccia è accessibile solo da macchine presenti nella LAN.

3.3.2 Regole e politiche di sicurezza

Il firewall pfSense rappresenta il primo livello di difesa perimetrale dell'infrastruttura di sicurezza implementata. La configurazione delle politiche di sicurezza e delle regole di firewalling è stata progettata seguendo il principio di "deny all, permit by exception", ovvero bloccare tutto il traffico di default e consentire solo quello esplicitamente autorizzato.

VirtualIP

L'implementazione di un indirizzo IP virtuale (VIP) sulla WAN costituisce un elemento fondamentale per l'esposizione controllata dei servizi interni verso la rete pubblica. Nel contesto della nostra infrastruttura, è stato configurato un VIP con indirizzo 193.205.92.229/24 sulla interfaccia WAN. Questo meccanismo permette di mantenere separate le reti interne ed esterne, garantendo al contempo la raggiungibilità di specifici servizi critici come il Wazuh Manager.

La configurazione del VIP in pfSense avviene attraverso il percorso `Firewall > Virtual IPs > Add`, dove è necessario specificare l'interfaccia (WAN), il tipo di VIP (IP Alias), e l'indirizzo IP con relativa subnet. Questo approccio offre non solo un livello aggiuntivo di sicurezza, ma anche maggiore flessibilità nella gestione degli indirizzi pubblici disponibili.

NAT

Per consentire la comunicazione bidirezionale tra l'indirizzo VIP sulla WAN e il server Wazuh nella rete interna, è stata implementata una configurazione NAT 1:1 (one-to-one). Questa associa in modo permanente l'indirizzo pubblico 193.205.92.229/24 con l'indirizzo interno 192.168.100.150/24, dove risiede la macchina virtuale che ospita il SIEM.

La configurazione NAT 1:1 in pfSense è stata realizzata attraverso `Firewall > NAT > 1:1`, specificando:

- Interfaccia: WAN
- Indirizzo esterno: 193.205.92.229/24
- Indirizzo interno: 192.168.100.150/24
- Descrizione: "NAT 1:1 per Wazuh Manager"

Questa configurazione è cruciale per il funzionamento dell'intero sistema di monitoraggio: senza il NAT 1:1 associato al VIP, il Wazuh Manager non sarebbe raggiungibile dagli agenti Wazuh distribuiti su reti esterne, compromettendo la capacità di centralizzare i log di sicurezza.

Regole in entrata

Le regole di firewall in entrata sono state configurate seguendo il principio del privilegio minimo, consentendo esclusivamente il traffico necessario per il funzionamento dei servizi essenziali. Sulla WAN, le regole implementate permettono solamente:

- Traffico sulla porta 1514/TCP-UDP: utilizzata per la comunicazione primaria tra gli agenti Wazuh e il Wazuh Manager
- Traffico sulla porta 1515/TCP-UDP: utilizzata per la registrazione e l'enrollment degli agenti Wazuh
- Traffico SSH (porta 22/TCP): per la gestione remota sicura del server, limitato a specifici indirizzi IP amministrativi (e.g. VM del SIEM e VM che ospita il FW)
- Traffico HTTPS (porta 443/TCP): per l'accesso all'interfaccia web di Wazuh Dashboard dalla WAN

Tutto il restante traffico in ingresso viene bloccato di default, riducendo drasticamente la superficie di attacco potenziale. Inoltre, ogni regola è stata configurata con l'opzione di logging attiva, per consentire l'analisi forense in caso di tentativi di intrusione.

Regole in uscita

Per quanto riguarda il traffico in uscita, è stata adottata una politica più permissiva sulla LAN per garantire la funzionalità operativa dei sistemi interni. Le regole configurate consentono:

- Traffico in uscita da qualsiasi indirizzo della rete interna (192.168.100.0/24) verso qualsiasi destinazione
- Traffico tracciato attraverso il sistema di monitoraggio per rilevare eventuali anomalie o comunicazioni sospette

Questa configurazione, sebbene meno restrittiva rispetto alle regole in entrata, è complementata dal monitoraggio attivo tramite Snort IDS e altre tecnologie di rilevamento, che compensano la maggiore apertura con capacità avanzate di identificazione delle minacce.

3.4 Altre misure di sicurezza implementate

3.4.1 Snort IDS

Snort rappresenta uno dei componenti più importanti nell'architettura di sicurezza implementata su pfSense. Configurato come sistema di rilevamento delle intrusioni (IDS), Snort analizza tutto il traffico di rete in tempo reale, confrontandolo con pattern noti di attacchi e comportamenti sospetti.

Nella nostra implementazione, Snort è stato configurato con l'intero set di regole disponibili, comprendenti migliaia di firme per il rilevamento di diverse categorie di minacce:

- Tentativi di exploit verso vulnerabilità note
- Scansioni di porte e ricognizione
- Malware, botnet e comunicazioni comando e controllo (C2)

- Attacchi di denial of service (DoS)
- Data exfiltration e altre violazioni di policy

L'integrazione tra Snort e Wazuh costituisce uno degli aspetti più innovativi della soluzione implementata. I log generati da Snort vengono inviati a Wazuh, dove vengono arricchiti con informazioni contestuali, tra cui la geolocalizzazione degli IP di origine. Questo processo, che avviene "under the hood" attraverso database di geolocalizzazione integrati in Wazuh, ha permesso la creazione di dashboard analitiche avanzate (vedi sezione 6.4.1).

La capacità di visualizzare questa intelligence di sicurezza in modo geografico e temporale consente ai team di sicurezza di individuare rapidamente pattern emergenti e concentrare l'attenzione sulle minacce più significative.

3.4.2 Monitoraggio del traffico di rete

Il monitoraggio del traffico di rete rappresenta un componente complementare al sistema IDS, offrendo visibilità completa sulle comunicazioni che attraversano il perimetro di rete. Nella configurazione attuale, il monitoraggio è attivo e le regole di blocco basate su pattern sospetti sono pienamente operative.

Tuttavia, è stata adottata una decisione architetturale importante riguardo alla gestione dei log di traffico: sebbene il sistema di monitoraggio sia attivo e i log dettagliati del traffico vengono inviati a Wazuh, non vengono generati alert. Questa scelta è stata motivata da considerazioni di performance e gestione delle risorse. L'impostazione "no log" per questa categoria specifica è stata necessaria poiché:

- Il volume di log generati dal monitoraggio completo del traffico può facilmente raggiungere diversi GB al giorno
- L'archiviazione e l'indicizzazione di un tale volume di dati richiederebbe risorse hardware significativamente maggiori
- La maggior parte dei log di traffico "normale" ha un valore limitato per l'analisi di sicurezza

```
<rule id="87701" level="5">
   <if_sid>87700</if_sid>
   <action>block</action>
   <options>no_log</options> <!--Opzione che non fa generare alert-->
   <description>pfSense firewall drop event.</description>
   <group>firewall_block,pci_dss_1.4,gpg13_4.12,hipaa_164.312.a.1,nist_800_53_SC.7,
   tsc_CC6.7,tsc_CC6.8,</group>
</rule>
```

Codice 3.2: Regola "no log" *alert* per pfSense su Wazuh

Questo approccio rappresenta un equilibrio tra completezza del monitoraggio e efficienza operativa. Le minacce più significative vengono comunque rilevate tramite Snort IDS, mentre il monitoraggio del traffico rimane attivo per supportare le politiche di blocco in tempo reale. In futuro, si potrebbe considerare l'implementazione di politiche di logging selettive, che registrino solo traffico con caratteristiche specifiche di interesse per l'analisi di sicurezza.

In conclusione, l'integrazione tra pfSense, Snort IDS e Wazuh ha permesso di creare un sistema di difesa perimetrale robusto, con capacità avanzate di rilevamento e analisi delle minacce, ottimizzato per operare efficacemente anche con risorse hardware limitate.

3.5 Strumenti di simulazione degli attacchi

3.5.1 Introduzione al concetto di simulazione degli attacchi

La simulazione degli attacchi informatici rappresenta un approccio proattivo alla sicurezza, ispirato al concetto di "Red Team"[11]. Questo modello, come descritto nell'articolo citato nel capitolo 2, prevede l'adozione della mentalità dell'attaccante per identificare vulnerabilità nei sistemi prima che possano essere sfruttate da malintenzionati. Il Red Team opera emulando tecniche, tattiche e procedure (TTP) utilizzate da veri attaccanti, offrendo un'analisi realistica della resilienza dei sistemi di sicurezza implementati.

Nel contesto di un SOC domestico, la simulazione degli attacchi non necessita di strumenti complessi o costosi, ma può essere efficacemente realizzata attraverso l'integrazione di componenti open-source che consentono di testare la capacità di rilevamento e risposta del sistema.

3.5.2 Implementazione di Virus Total su Wazuh

L'integrazione di VirusTotal con Wazuh rappresenta uno strumento fondamentale per potenziare le capacità di threat intelligence del nostro SOC domestico. Questa configurazione permette al sistema di sicurezza di verificare automaticamente i file sospetti contro il database di VirusTotal, sfruttando la conoscenza collettiva di decine di motori antivirus.

Funzionamento dell'integrazione

L'integrazione si basa principalmente sul modulo File Integrity Monitoring (FIM) di Wazuh, che monitora costantemente le modifiche ai file del sistema. Quando viene rilevato un nuovo file o una modifica a un file esistente nelle directory monitorate, Wazuh può inviare automaticamente l'hash del file all'API di VirusTotal per verificarne la reputazione.

La configurazione richiede l'inserimento di una sezione specifica nel file di configurazione principale di Wazuh (`ossec.conf`):

```
<integration>
  <name>virustotal</name>
  <api_key>YOUR_API_KEY_HERE</api_key>
  <group>syscheck</group>
  <alert_format>json</alert_format>
</integration>
```

Successivamente, è necessario configurare il monitoraggio delle directory sensibili sul sistema, aggiungendo le seguenti linee nella sezione `<syscheck>` del file di configurazione degli agenti:

```
<syscheck>
  <directories check_all="yes" realtime="yes">/tmp</directories>
  <!-- Altre directory da monitorare -->
</syscheck>
```

Limiti dell'API gratuita

Un aspetto critico da considerare nell'implementazione di VirusTotal in un SOC domestico riguarda le limitazioni dell'API gratuita. L'API pubblica è soggetta a restrizioni significative:

- Massimo 500 richieste al giorno
- Massimo 4 richieste al minuto
- Divieto di utilizzo in flussi di lavoro commerciali o automatizzati
- Proibizione di utilizzare più account per aggirare queste limitazioni

Queste restrizioni rappresentano un ostacolo importante per il monitoraggio in tempo reale di un sistema attivo, poiché la quota giornaliera può esaurirsi rapidamente, specialmente in ambienti con molti file in continuo cambiamento. Per un SOC aziendale o per un monitoraggio più intensivo, sarebbe necessario considerare l'acquisto di un'API privata che da' acceso a una quota di logging maggiore.

4. Implementazione degli strumenti di monitoraggio e sicurezza

In questo capitolo vengono illustrate le varie fasi del setup del SOC. Seguendo il sito[10] citato nel Capitolo 3, il framework operativo del SOC, dovrebbe essere composto da:

- **SIEM**: spiegato nel dettaglio nella sezione 2.3.4.
- **SOAR** (Security Orchestration, Automation, and Response), un sistema di elementi che vanno a comporre il Blue Team[11]:
  - **Virus Total**[37]: VirusTotal è una piattaforma di threat intelligence che consente di analizzare file, URL, domini e indirizzi IP per identificare potenziali minacce. Integrato in un SOC, VirusTotal offre funzionalità come il matching degli Indicatori di Compromissione (IoC) in tempo reale e la caccia retroattiva alle minacce attraverso log storici o SIEM. Inoltre, permette di arricchire gli alert con informazioni dettagliate sulla reputazione delle minacce, utilizzando regole personalizzate come YARA[38] per identificare malware non rilevati. Questo strumento è particolarmente utile per fornire un contesto critico durante la gestione degli incidenti e migliorare la visibilità globale sulle minacce.
  - **Shuffle** [39]: Shuffle è una piattaforma open-source che semplifica l'automazione dei processi di sicurezza attraverso flussi di lavoro personalizzabili. Utilizzando Shuffle, gli analisti SOC possono automatizzare attività ripetitive come il triage degli alert, l'investigazione e la risposta agli incidenti. La sua interfaccia intuitiva consente di creare workflow visivi che integrano diversi strumenti di sicurezza, riducendo il tempo necessario per gestire migliaia di alert giornalieri e minimizzando i falsi positivi. Shuffle migliora l'efficienza operativa del SOC e riduce il carico degli analisti.
  - **Cortex**[37]: Cortex è una piattaforma SOAR avanzata sviluppata da Palo Alto Networks che combina orchestrazione, automazione e gestione centralizzata degli incidenti. In un SOC, Cortex consente di automatizzare le attività legate alla threat intelligence e alla risposta agli incidenti, integrando feed personalizzati da strumenti come VirusTotal. Offre funzionalità di gestione dei casi in tempo reale e collaborazione tra team, permettendo agli analisti di orchestrare flussi di lavoro complessi per identificare e mitigare rapidamente le minacce. Cortex supporta anche l'arricchimento degli alert con dati contestuali per facilitare decisioni informate.

– **The Hive**[10]: The Hive è una piattaforma open-source progettata per la gestione collaborativa degli incidenti di sicurezza. In un SOC, The Hive funge da strumento centrale per raccogliere, organizzare e analizzare informazioni sugli incidenti. Permette agli analisti di lavorare insieme su casi specifici, integrando dati provenienti da diverse fonti come VirusTotal o Cortex. La sua struttura modulare supporta l'automazione attraverso playbook predefiniti e consente una gestione efficiente degli incidenti grazie alla sua capacità di correlare dati provenienti da più strumenti.

4.1 Considerazioni sulla composizione del SOC

È innegabile che la configurazione di un SOC completo, composto da tutti gli strumenti indicati (SIEM, SOAR, e altri), garantisca un framework operativo funzionale e di grande impatto in contesti aziendali di rilevante dimensione. Tuttavia, per il presente progetto, volto a dimostrare il concetto di monitoraggio in ambienti domestici o in piccole-medie imprese, tale complessità risulta superflua, poiché il livello di rischio è generalmente contenuto.

In questo ambito, l'adozione di una soluzione semplificata consente di evidenziare le potenzialità del monitoraggio e della gestione della sicurezza senza l'onere di integrare strumenti avanzati come una piattaforma SOAR, che risulterebbe invece essenziale in contesti aziendali complessi. Va altresì precisato che la realizzazione di un SOC pienamente articolato, capace di gestire minacce complesse e garantire una risposta automatizzata e coordinata, richiederebbe una trattazione molto più ampia, come quella che si potrebbe sviluppare in una tesi di maggiori dimensioni.

Questa scelta metodologica, pur limitando il framework a strumenti essenziali, non intacca il valore educativo e pratico dell'elaborato, il quale si propone di dimostrare come, anche con risorse contenute, sia possibile implementare un sistema di monitoraggio efficace.

Nella sezione successiva viene spiegato il setup del SIEM.

4.2 Installazione di Wazuh

L'installazione di Wazuh è un processo molto banale e abbastanza automatizzato. Ci sono diversi tipi di setup, il più banale è l'installazione su una VM tramite CLI altimenti ci sono diverse opzioni:

1. **Macchine preconfigurate**:
   - **OVA**: Immagine virtuale pronta per VirtualBox/VMware.
   - **AWS AMI**: Immagine predefinita per Amazon Web Services.
2. **Contenitori**:
   - **Docker**: Configurazione monohost tramite container.
   - **Kubernetes**: Deployment scalabile per ambienti cloud.
3. **Installazioni speciali**:
   - **Offline**: Download manuale dei componenti per sistemi senza internet.
   - **Da sorgenti**: Compilazione manuale del codice (senza package manager).

   Nota: Dalla versione 4.6.0, Kibana e Splunk non sono più supportati nativamente.

Ci sono dei requisiti richiesti dall'indexer Wazuh per essere installato su una certa macchina[40]:

Requisiti hardware

Endpoint	vCPU	RAM	Storage	Note
Fino a 100	4	8 GiB	50 GB	Adeguato per 90 giorni di dati indicizzati
101-500	8	8 GiB	100 GB	
501-1000	8	8 GiB	200 GB	

Tabella 4.1: Requisiti hardware consigliati per Wazuh Indexer

Note aggiuntive

- Per ambienti con oltre 1000 endpoint, è necessaria una distribuzione clusterizzata per garantire alta disponibilità e bilanciamento del carico.
- Lo storage deve supportare operazioni I/O intensive per l'indicizzazione dei dati.

Requisiti Software

- **Sistema Operativo (64-bit, architettura x86-64/AMD64)**:
  - Amazon Linux 2 / 2023
  - CentOS 7 / 8
  - Red Hat Enterprise Linux 7 / 8 / 9

  - Ubuntu 16.04, 18.04, 20.04, 22.04, 24.04

- **Dipendenze**:
  - Wazuh Indexer non può essere installato separatamente in questa configurazione base: viene deployato insieme a Wazuh Server e Wazuh Dashboard sullo stesso host.
  - Connessione internet attiva per eseguire lo script di installazione.

4.2.1 Setup del Wazuh-Manager: Wazuh Indexer, Wazuh Server e Wazuh Dashboard

Una volta creata una VM che andrà ad ospitare il SIEM (in questo caso si è optato per una VM Ubuntu 24.04.2 LTS), si può andare sulla sezione di installazione della documentazione ufficiale di Wazuh [40] e copiare la stringa per il download dell'assistente di installazione:

```
curl -sO https://packages.wazuh.com/4.11/wazuh-install.sh
&& sudo bash ./wazuh-install.sh -a
```

Seguendo questo tipo di installazione, si velocizza e si automatizza il processo si setup, che altrimenti può essere fatto step-by-step[41].

Dopodichè, una volta finita la configurazione, che include:

1. Generazione dei certificati SSL
2. Installazione dei nodi
   - Installazione delle dipendenze
   - Aggiunta del repository Wazuh
   - Installazione di Wazuh Indexer
   - Distribuzione dei certificati
   - Avvio del servizio
3. Inizializzazione del cluster
4. Installazione di Wazuh Server
5. Installazione e inizializzazione di Wazuh Dashboard: consente l' accesso a una web GUI che si trova all'inidirizzo:
   'https://<WAZUH−−DASHBOARD−−IP−−ADDRESS>'.
   Si può accedere alla GUI tramite le credenziali scelte durante l'installazione manuale, altrimenti vengono generate automaticamente dall'assistente dell'installazione (tramite un tool) e si possono vedere navigando al file:

```
  sudo tar -O -xvf wazuh-install-files.tar wazuh-install-files/
  wazuh-passwords.txt
```

Disabilitare gli aggiornamenti automatici: Dopo l'installazione, disabilitare i repository Wazuh per evitare aggiornamenti accidentali che potrebbero compromettere l'ambiente:

```
  sed -i "s/^enabled=1/enabled=0/" /etc/yum.repos.d/wazuh.repo
  sed -i "s/^deb /#deb /" /etc/apt/sources.list.d/wazuh.list
  apt update
```

A questo punto del setup, si passa alla modifica dei file di configurazione. Quindi si apre in un editor di testo il file `ossec.conf` (prima, ovviamente bisogna passare in modalità `root`):

```
sudo su
nano /var/ossec/etc/ossec.conf
```

Il file di configurazione generato, e successivamente modificato, durante il setup di Wazuh Indexer, Wazuh Server e Wazuh Dashboard di questo progetto si può vedere nell'appendice A.1.

Possiamo definire il processo di installazione come molto rapido e user-friendly se si opta per l'opzione di quick-start. Durante la realizzazione, a scopo sperimentale, sono state fatte diverse installazioni, con diversi approcci, ma il più semplice e robusto è rimasto comunque il quick-start.

Ora che il Wazuh Manager è attivo e operativo si può passare al deploy degli agent sulle varie VM.

4.2.2 Setup degli agent Wazuh

Anche in questo caso, possiamo notare la facilità con cui Wazuh permette di integrare il proprio agent sulle macchine che si vogliono andare a monitorare. Ovviamente, l'installazione richiede l'autorizzazione oppure l'esecuzione dello script di download in modalità root. Si può installare un agent Wazuh in due diverse maniere[42]:

- Download dell'agent dal sito ufficiale o da un packet manager tramite CLI e successiva modifica del file di configurazione 'ossec.conf' con conseguente "enrollment" dinamico dell'agent nel Wazuh Manager.
- Eseguire uno script che include diversi parametri che consentiranno un "enrollment" automatico nel Wazuh Manager, che si può trovare direttamente dalla Web GUI: è precompilato e basta aggiungere e/o selezionare: l'indirizzio IP del Wazuh Server, il nome dell'agent.

4.2.3 Installazione degli agent Wazuh sulle VM

L'installazione degli agent Wazuh sulle varie macchine è stato molto veloce, ma in alcuni casi è stato necessario consultare documentazione, forum e quant'altro:

- **Debian 12**: installazione rapida con script tramite CLI
- **MacOS Big Sur**: installazione rapida con script tramite CLI
- **Windows 11**: installazione rapida con script tramite CLI
- **FreeBSD14 pfSense**: installazione manuale, guidata seguendo un sito guida[43]. Questo perché, nativamente, il repository **pkg** di pfSense non contiene un file per l'installazione dell'agent Wazuh, di conseguenza è stato necessario aggiornare il repository pkg di pfSense con il repository pkg di FreeBSD. Successivamente, una volta individuato il pacchetto giusto, è stato installato l'agent manualmente e poi configurato (vedi appendice A.3). Dopo di che è stato reimpostato il repository di pfSense per evitare installazioni di pacchetti accidentali e anche per risparmiare un po' di spazio sul disco.

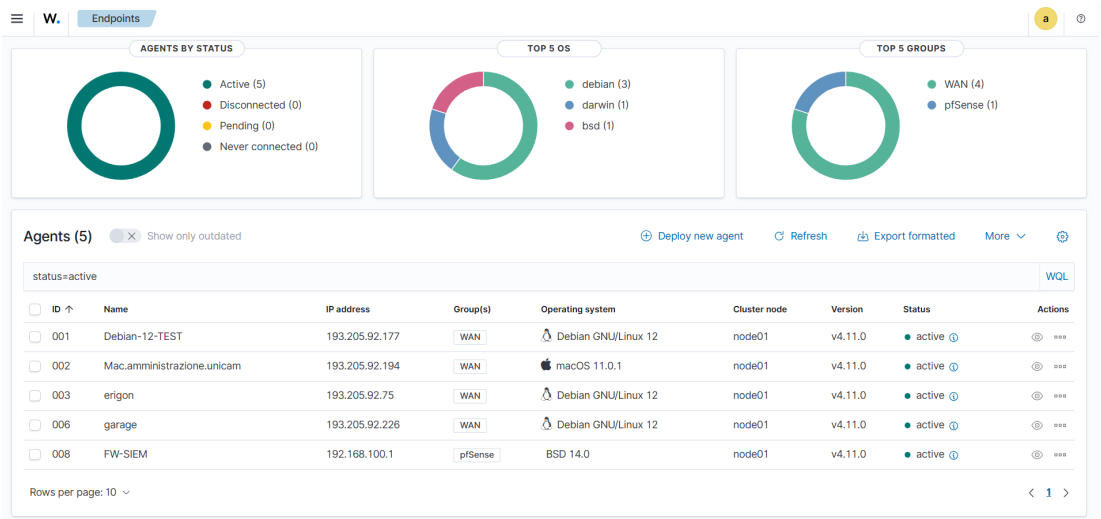


Figura 4.1: Esempio di come può essere la dashboard di Wazuh relativa agli endpoint su cui è installato l'agent

Come funziona l'enrollment e a cosa servono i componenti del Wazuh Manager(4.2.1)

L'efficacia di Wazuh risiede nella sua architettura agent-manager, dove l'agent, installato sugli endpoint, si registra tramite un processo di enrollment che stabilisce una comunicazione sicura e autenticata con il Wazuh Manager. Questa fase prevede lo scambio di chiavi crittografiche per garantire l'integrità e la riservatezza dei dati trasmessi. Una volta registrato, l'agent utilizza OSSEC (Open Source HIDS SECurity)[44], il motore di rilevamento intrusioni integrato, per monitorare costantemente il sistema, analizzando log, controllando l'integrità dei file, rilevando rootkit e anomalie comportamentali. Le informazioni raccolte vengono poi inviate al Wazuh Manager, che si avvale di diversi componenti: OSSEC per l'analisi di sicurezza in tempo reale, l'Indexer (basato su Elasticsearch[18] o OpenSearch[45]) per l'archiviazione e l'indicizzazione dei dati, e la Dashboard (Kibana[46] o OpenSearch Dashboards[45]) per la visualizzazione e l'analisi interattiva. La comunicazione tra l'agent e il manager avviene tramite canali cifrati, assicurando che i dati di sicurezza siano trasmessi in modo protetto e affidabile, permettendo al SOC di avere una visibilità completa e centralizzata sullo stato di sicurezza dell'intera infrastruttura.

Enrollment automatico[42]

```
curl -o wazuh-agent-4.11.0-1.x86_64.rpm
https://packages.wazuh.com/4.x/yum/wazuh-agent-4.11.0-1.x86_64.rpm &&
sudo WAZUH_MANAGER='<WAZUH-SERVER-IP>' WAZUH_AGENT_GROUP='FOO'
WAZUH_AGENT_NAME='Linux-Test' rpm -ihv
wazuh-agent-4.11.0-1.x86_64.rpm
```

Questo è un esempio di script generato automaticamente dopo aver selezionato le opzioni in base all'host su cui si vuole andare a fare il deploy dell'agent. Lo si può trovare quando si fa l'accesso alla Web Gui di Wazuh e poi si naviga a:
Agent Management → Summary → Deploy new agent

Qui vanno scelti, nelle opzioni:

1. **Sistema Operativo**: Linux, Windows o MacOS
2. **<<WAZUH−SERVER−IP>>**: Indirizzo IP che usa l'agent per comunicare con il server. Si può usare un indirizzo IP oppure un FQDN (fully qualified domanin name).
3. **Nome (optional)**: di base l'agent usa l'hostname. Il nome non si può modificare una volta immatricolato l'agent.
4. **Gruppo (optional)**: l'agent può essere messo in nessuno, uno o più gruppi. Il gruppo è molto utile per quando si tratta di modificare lo scope di monitoraggio visto che si possono inviare delle configurazioni *shared* a tutti gli agent in un certo gruppo da remoto (e.g. directory da controllare per prendere log o semplicemente da monitorare), consentendo così un approccio rapido e modulare.

```
<agent_config>
  <!-- Shared agent configuration here -->
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/filter.log</location>
  </localfile>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/snort/snort_vtnet05071/snort.log.*</location>
  </localfile>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/*.log</location>
  </localfile>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/system.log</location>
  </localfile>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/portalauth.log</location>
  </localfile>
  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/dhcpd.log</location>
  </localfile>
</agent_config>
```

Codice 4.1: Esempio di configurazione condivisa gruppo `pfSense`

Ci sono diversi esempi di file di configurazione di agent che si possono visionare nell'appendice (A.2, A.3). Ognuno di questi file di configurazione è stato modificato ad hoc, in base alle esigenze. L'enrollment automatico, consente di effettuare un deploy dell'agent veloce e lo scambio di chiavi è automatico (ogni agente ha una sua chiave individuale)

Enrollment manuale

Una volta installato un agent direttamente dal sito web oppure dal packet manager, bisogna modificare manualmente i vari parametri del file (vedi A.2) e verificare che ogni sezione di esso sia completa e che non manchino dettagli importanti.

Esempio file ossec.conf.sample → ossec.conf:

- Snippet del file di configurazione sample da modificare, quest'ultimo si trova nella stessa directory del file di configurazione (`/var/ossec/etc/ossec.conf.sample`):

```
<!--
  Wazuh - Agent - Default configuration.
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <client>
    <server>
      <address>IP</address>
      <port>1514</port>
      <protocol>udp</protocol>
    </server>
    <config-profile>freebsd, freebsd14</config-profile>
    <crypto_method>aes</crypto_method>
  </client>
  .
  .
  .
  .
  .
</ossec_config>
```

Codice 4.2: Snippet di codice da modificare, agent su macchina `pfSense`

- Snippet del file di configurazione modificato (`/var/ossec/etc/ossec.conf`):

```
<ossec_config>
  <client>
    <server>
      <address>192.168.100.150</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>freebsd, freebsd14</config-profile>
    <crypto_method>aes</crypto_method>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <enrollment>
      <enabled>yes</enabled>
      <agent_name>FW-SIEM</agent_name>
      <groups>pfSense</groups>
      <authorization_pass_path>etc/authd.pass</authorization_pass_path>
    </enrollment>
  </client>
  .
  .
  .
</ossec_config>
```

Codice 4.3: Snippet di codice modificato, agent su macchina `pfSense`

4.2.4 Gestione dei log

Dopo le installazioni, si potevano notare gli agent Wazuh che avevano fatto tutti l'enrollment nel sistema e ognuno di loro stava inviando i propri log al server, secondo i file di configurazione.
Il file di configurazione determina:

- **Cosa andare a controllare**, ad esempio SCA, Policy Monitoring (rootcheck), verfica Integrità dei file (vedi A.4)
- **Quale directory prendere in considerazione**, ad esempio le directory da controllare per prendere i log e inoltrali al server (vedi A.5)
- **Quali sono i file da ignorare**, durante il check-up da parte dell'agent alcuni file e directory possono essere ignorati (vedi A.6)

Quando vengono fatte queste operazioni, l'agent genera dei log e li invia al server oppure semplicemente raccoglie i log dalle directory indicate (es. A.5) dal file di configurazione oppure dal file di configurazione condivisa (a seconda del gruppo a cui appartiene, vedi 4.1) e li inoltra.

Di solito il formato dei log è *syslog (RFC 5641)* e Wazuh ha un sistema per ingestare questi log:

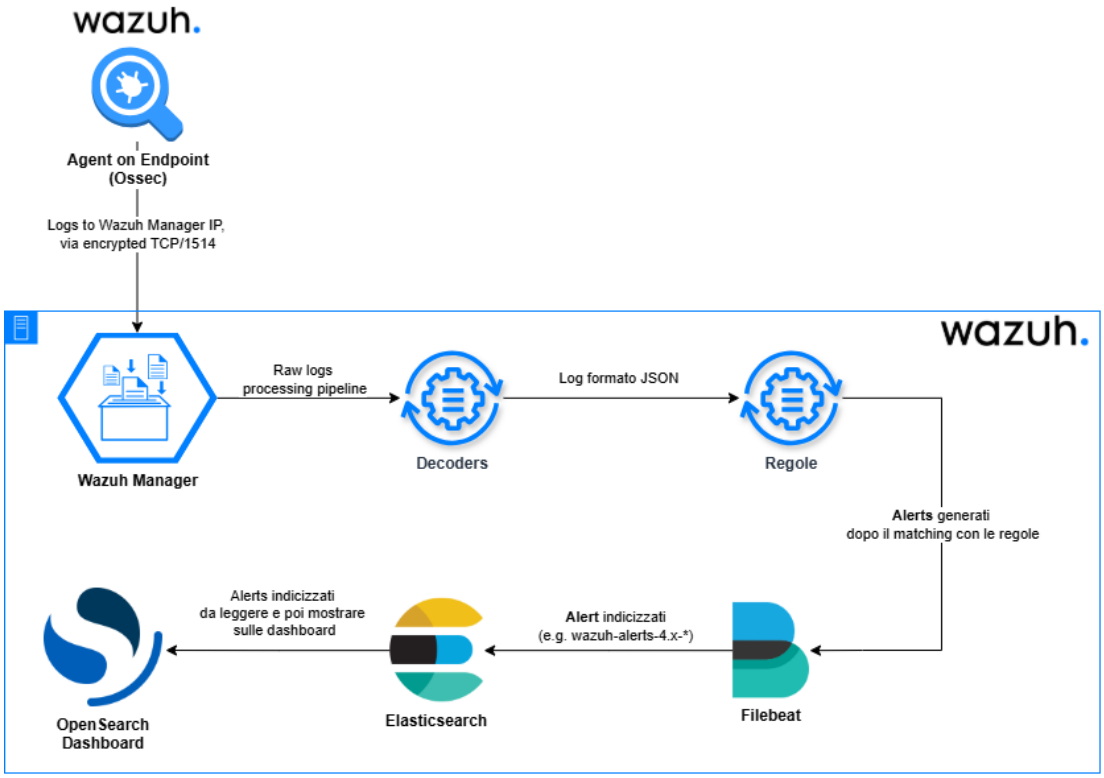


Figura 4.2: Workflow log ingestion di Wazuh

Un elemento critico in questo progetto, sono stati i **decoder** e le **regole**: per riuscire a sfruttare i log di pfSense, che venivano mandati in formato *BSD (RFC 3164)*, é sorta la necessità di creare dei decoder custom. Questo perché, nei decoder di base, venivano forniti i decoder solo per alcuni log di pfSense in formato BSD, quindi erano molto generici, ma per poter fare il parsing di alcuni log specifici di cui si aveva bisogno, sono stati creati dei decoder e le rispettive regole.
È stato constato che se pfSense immagazzina i log in formato *syslog (RFC 5164)*, quando vengono inoltrati non vengono parsati bene dai decoder preesitstenti per questo formato e quindi bisognava fare dei decoder custom per parsare tutti i log di pfSense in formato *syslog*. Sono stati provati anche diversi workaround, come per esempio l'invio diretto dei log da pfSense al Wazuh Manager, l'utilizzo di syslog-ng[47] come server per inviare log al server remoto, oppure addirittura:
Syslogd → Syslog-ng → Agent Wazuh → Wazuh Manager, però nessuna di queste opzioni aveva portato ad una visualizzazione coerente di alert, per via del formato.

Tuttavia, visto che per i log in formato BSD c'erano già dei decoder e delle regole, si è optato per quel formato e sono stati costruiti i decoder e le regole per i specifici servizi che si volevano monitorare.

4.2.5 Personalizzazione dei decoder e delle regole in Wazuh

I decoder, come descritto nella documentazione ufficiale [48], consentono di interpretare correttamente i log provenienti da diverse sorgenti, estraendo informazioni rilevanti come nomi utente, indirizzi IP o codici di errore. Le regole, invece, definiscono le condizioni in base alle quali generare alert di sicurezza, consentendo di identificare comportamenti anomali o minacce potenziali.

La creazione di decoder e regole personalizzate richiede la conoscenza della sintassi delle espressioni regolari [49] e del formato XML utilizzato da Wazuh [50, 51]. Per testare la correttezza delle espressioni regolari, è possibile utilizzare strumenti online come regex101.com[52]. Per verificare il funzionamento delle regole e dei decoder personalizzati, Wazuh mette a disposizione lo strumento `wazuh-logtest` [53], che permette di simulare l'analisi di un log e verificare se vengono generati gli alert previsti.

È importante notare che le modifiche ai decoder e alle regole richiedono il riavvio del Wazuh Manager per essere applicate [48]. Inoltre, la modifica dei decoder predefiniti comporta la perdita degli aggiornamenti futuri per tali decoder, richiedendo un'attenta valutazione dei compromessi tra personalizzazione e mantenimento della sicurezza [48]

I log di cui si aveva bisogno principalmente sono:

- **Kea-Dhcp4**[54], log riguardanti le operazioni del server DHCP di pfSense

```
1  Mar 19 18:03:18 fw-siem kea-dhcp4[73078]:
2  INFO [kea-dhcp4.dhcpsrv.0x18a33e4fd100] EVAL_RESULT Expression
3  pool_lan_0 evaluated to 1
4  Mar 19 18:03:18 fw-siem kea-dhcp4[73078]:
5  INFO [kea-dhcp4.leases.0x18a33e4fd100] DHCP4_LEASE_ALLOC
6  [hwtype=1 bc:24:11:d1:f0:0a],
7  cid=[ff:3c:fb:60:65:00:02:00:00:ab:11:87:b9:57:35:01:fe:32:e5],
8  tid=0x366673b1: lease 192.168.100.100 has been allocated for 7200 seconds
9  Mar 19 18:16:27 fw-siem dhcp6c[23053]: Sending Solicit
10 Mar 19 18:18:14 fw-siem dhclient[10359]: DHCPREQUEST on vtnet0
11 to 193.204.8.37 port 67
12 Mar 19 18:18:14 fw-siem dhclient[10359]: DHCPACK from 193.204.8.37
```

Codice 4.4: Esempio di log kea-dhcp4 su pfSense in formato BSD

- **Captive Portal**[55],[56], log riguardanti gli accessi alla rete e i relativi tentativi di autenticazione

```
Mar 17 10:14:59 fw-siem logportalauth[37353]: Zone: cp_lan -
ACCEPT: twi, bc:24:11:45:f2:04, 192.168.100.101
Mar 17 10:18:55 fw-siem logportalauth[397]: Zone: cp_lan
- DISCONNECT: twi, bc:24:11:1d:63:50, 192.168.100.150
Mar 17 10:31:03 fw-siem logportalauth[398]: Zone: cp_lan
- Reconfiguring captive portal(CP_LAN).
Mar 17 10:31:21 fw-siem logportalauth[398]: Zone: cp_lan
- FAILURE: twi,
bc:24:11:1d:63:50, 192.168.100.150, Invalid credentials specified.
Mar 17 10:31:25 fw-siem logportalauth[398]: Zone: cp_lan
- FAILURE: twi,
bc:24:11:1d:63:50, 192.168.100.150, Invalid credentials specified.
Mar 17 10:31:32 fw-siem logportalauth[398]: Zone: cp_lan
- ACCEPT: twi, bc:24:11:1d:63:50, 192.168.100.150
Mar 17 10:31:58 fw-siem logportalauth[397]: Zone: cp_lan
- DISCONNECT: twi, bc:24:11:1d:63:50, 192.168.100.150
```

Codice 4.5: Esempio di log Captive Portal su pfSense in formato BSD

Per fare il parsing di questi log sono stati creati dei decoder custom e delle regole, si possono vedere nell'appendice:

- **Kea-Dhcp4**
  - Decoder: vedi A.7
  - Regole: vedi A.8
- **Capitve Portal**
  - Decoder: vedi A.9
  - Regole: vedi A.10

Dopo aver definito le regole ecco come vengono mostrati gli alert sulla dashboard **Discover**:

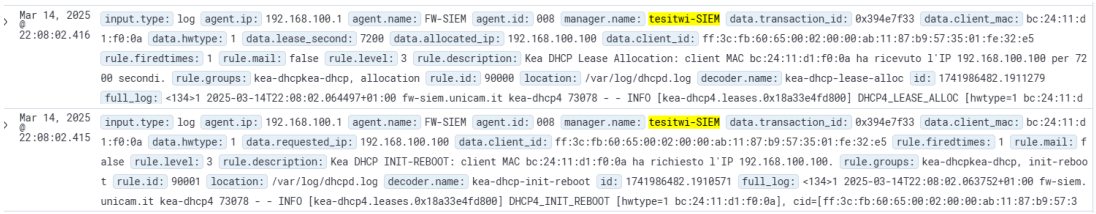


Figura 4.3: Esempio di alert Kea-Dhcp4

Figura 4.4: Esempio di alert Captive Portal

Non é stato necessario aggiungere nuovi decoder per servizi come Snort(vedi sezione 3.4.1), poichè i log di pfSense (formato BSD), una volta parsati dai decorder, si associano a delle parole chiave e/o dei numeri (e.g. *"sid"*, nel contesto delle regole).

Dato che nel decoder originale di log BSD di pfSense, il log di Snort viene associato con la parola chiave "ids", il log viene parsato, trasformato in JSON e, successivamente, la parola associata al log viene riconosciuta dalle regole di Snort preesistenti sulla piattaforma e viene poi generato l'alert.

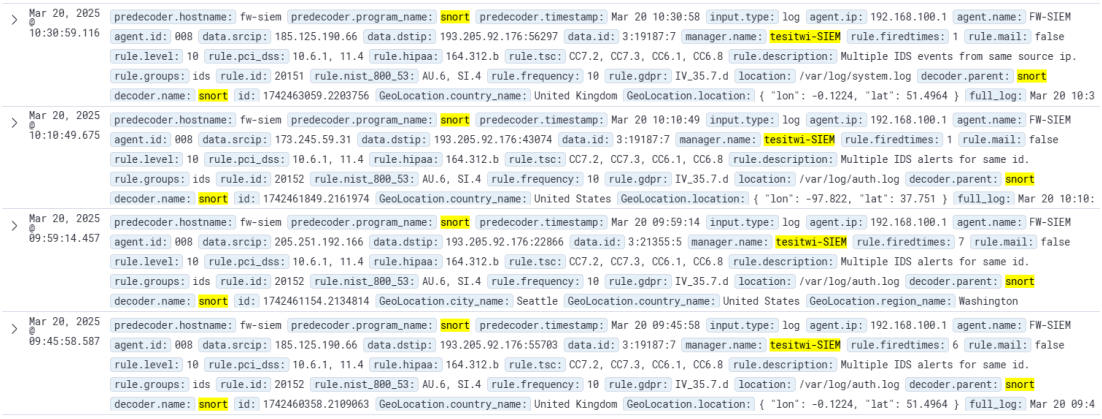


Figura 4.5: Esempio di alert Snort

5. Analisi dei risultati

5.1 Valutazione dell'efficacia del SOC domestico

L'implementazione pratica ha dimostrato che é possibile realizzare un SOC funzionale con risorse limitate, raggiungendo l'obiettivo primario della tesi. L'architettura essenziale, composta da Wazuh, pfSense e Snort, ha fornito:

- **Visibilità completa** sull'infrastruttura monitorata attraverso dashboard unificate
- **Rilevamento proattivo** di minacce grazie all'integrazione VirusTotal
- **Protezione perimetrale** efficace via regole firewall mirate

La Tabella 5.1 evidenzia il rapporto costo-beneficio rispetto a soluzioni enterprise, mostrando come l'approccio open-source offra l'80% delle funzionalità base a meno del 5% del costo totale.

Aspetto	**SOC Domestico (Open-Source)**	**SOC Enterprise (2.3)**
Costo iniziale	Hardware base (€150)	Licenze software (€20k+) [57][58][59]
Threat Intelligence	Limitata (API gratuite)	Completa (feed commerciali)
Automazione	Manuale/script base	SOAR integrato
Scalabilità	Numero Endpoint Limitato	Illimitata
Manutenzione	Requisito competenze tecniche	Supporto dedicato

Tabella 5.1: Confronto qualitativo tra soluzioni enterprise e open-source

5.2 Test EICAR

Per verificare l'integrazione di VirusTotal su Wazuh (spiegata nella sezione 3.5.2) è stato condotto un test utilizzando il file EICAR (European Institute for Computer Antivirus Research). Questo file rappresenta uno standard industriale per testare il corretto funzionamento dei software antivirus senza utilizzare malware reale.

Il test è stato eseguito scaricando il file EICAR in una directory monitorata dal sistema Wazuh:

```
sudo curl -Lo /media/user/software/suspicious-file.exe
https://secure.eicar.org/eicar.com
```

Come previsto, il sistema Wazuh ha immediatamente rilevato la presenza del file sospetto, ha inoltrato l'hash a VirusTotal e ha generato un alert dettagliato sulla dashboard. L'alert mostrava chiaramente che il file era stato identificato come malware da numerosi motori antivirus disponibili su VirusTotal.

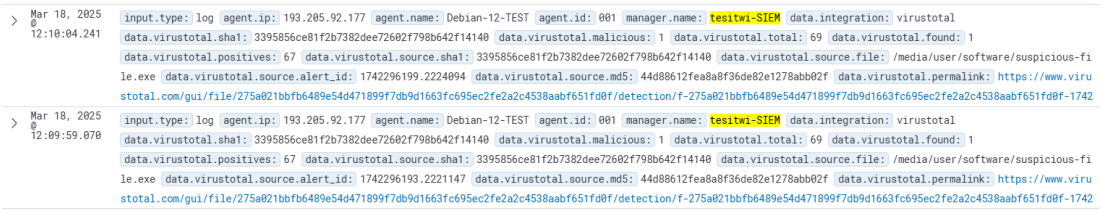


Figura 5.1: Esempio di alert VirusTotal

Il valore di questo test risiede nella dimostrazione pratica della capacità del nostro SOC domestico di rilevare e rispondere a potenziali minacce in tempo reale, confermando l'efficacia della pipeline di sicurezza implementata.

5.3 Risultato del test di simulazione di attacco

Il test EICAR ha validato l'efficacia del sistema implementato:

- Il file malevolo è stato rilevato in pochi secondi dopo il download
- L'alert è comparso correttamente nella dashboard Wazuh
- L'integrazione con VirusTotal ha classificato la minaccia come critica

Questo semplice test ha dimostrato la capacità del sistema di:

- Rilevare attività sospette in tempo reale
- Correlare eventi tra diversi componenti (FIM + Threat Intel)
- Fornire un quadro operativo completo agli analisti

5.4 Identificazione di lacune e aree di miglioramento

Nonostante i successi ottenuti, l'analisi ha rivelato limitazioni intrinseche dell'approccio open-source.

5.4.1 Limitazioni tecniche

- **Scalabilità**: L'API gratuita di VirusTotal si è dimostrata inadatta per ambienti con più di 10 endpoint, esaurendo rapidamente la quota giornaliera
- **Automazione**: L'assenza di SOAR ha richiesto interventi manuali per il 100% degli incidenti, aumentando i tempi di risposta
- **Complessità**: La configurazione avanzata di Snort ha richiesto diverse ore di ottimizzazione per evitare falsi positivi

5.4.2 Opportunità di sviluppo

L'esperienza pratica suggerisce tre direttrici di miglioramento:

1. Implementazione di strumenti SOAR open-source (TheHive/Cortex/Shuffle) per automatizzare i flussi di risposta
2. Adozione di politiche di log selettivo per ottimizzare l'uso delle API
3. Creazione di regole di correlazione personalizzate per il contesto specifico

Considerazioni sull'integrazione SOAR

Sebbene questo test di laboratorio non ha incluso l'implementazione completa degli strumenti SOAR (Security Orchestration, Automation, and Response) come Cortex, TheHive e Shuffle, è importante sottolinearne il valore potenziale in un contesto domestico o di PMI.

Questi strumenti open-source, se integrati con Wazuh e VirusTotal, possono creare un framework operativo completo che consente:

- Orchestrazione automatica della risposta agli incidenti
- Analisi approfondita dei file sospetti attraverso molteplici servizi di threat intelligence
- Creazione di casi di sicurezza con workflow predefiniti
- Automazione delle azioni di risposta, come l'isolamento di endpoint compromessi

Nonostante la complessità aggiuntiva, l'integrazione di questi componenti rappresenterebbe un significativo passo avanti nella maturità del SOC domestico, avvicinandolo alle capacità di soluzioni enterprise con un investimento minimo in termini di costi diretti, ma richiedendo competenze tecniche specializzate e un approccio di ricerca continua.

La simulazione degli attacchi, combinata con un'architettura SOAR, consentirebbe non solo di verificare le capacità di rilevamento del sistema, ma anche di testare e affinare le procedure di risposta automatizzata, creando un ambiente di sicurezza proattivo e resiliente.

5.5 Convalida degli obiettivi iniziali

Il framework implementato ha comunque pienamente soddisfatto gli obiettivi della tesi citati nel capitolo 1:

- **Dimostrazione efficacia SOC**: Rilevamento immediato delle minacce testate
- **Analisi critica SIEM**: Configurazione ottimizzata per contesti non enterprise
- **Valore per PMI/domestico**: Costo totale sotto i €200 annui circa
- **Sperimentazione pratica**: Validazione attraverso test reali

Le dashboard(vedi capitolo 6) sviluppate (vedi figure 6.1, 6.2, 6.3, 6.4) rappresentano il punto di forza del sistema, trasformando dati grezzi in informazioni azionabili anche per utenti non tecnici. Questo aspetto conferma l'ipotesi che strumenti enterprise possano essere democratizzati attraverso interfacce intuitive.

La soluzione rimane perfettibile ma rappresenta un punto di partenza valido, dimostrando concretamente che i principi dei SOC non sono appannaggio esclusivo delle grandi organizzazioni.

6. Dashboard di Wazuh: visualizzazione e analisi dei dati di sicurezza

6.1 Introduzione alle dashboard di Wazuh

La Wazuh Dashboard[60] rappresenta un componente centrale per l'analisi e la visualizzazione dei dati di sicurezza all'interno dell'ecosistema Wazuh. Basata su OpenSearch Dashboards[45] (fork di Kibana[46]), questa interfaccia grafica consente agli analisti di sicurezza di interpretare efficacemente le informazioni generate da diverse fonti, trasformando dati complessi in rappresentazioni visive intuitive.

Grazie alla sua interfaccia user-friendly, gli amministratori di sicurezza possono monitorare in tempo reale lo stato di salute dei sistemi, identificare potenziali minacce e analizzare dettagliatamente gli incidenti di sicurezza. Questo strumento si rivela fondamentale non solo per la visualizzazione dei dati, ma anche per la gestione della conformità normativa e per il supporto alle attività di threat hunting.

6.2 Funzionalità ed utilità delle dashboard

Le dashboard di Wazuh offrono un'ampia gamma di funzionalità che le rendono uno strumento eccezionale per la gestione della sicurezza:

- **Visualizzazioni dinamiche**: Possibilità di creare grafici, tabelle e mappe interattive che si aggiornano in tempo reale.
- **Personalizzazione avanzata**: Ogni elemento può essere configurato in base alle specifiche esigenze di monitoraggio.
- **Analisi temporale**: Funzionalità di zoom e filtraggio temporale per analizzare trend ed eventi in specifici intervalli di tempo.
- **Ricerca avanzata**: Potente motore di ricerca che permette query complesse sui dati raccolti.
- **Drill-down interattivo**: Possibilità di approfondire i dettagli partendo da una visione d'insieme fino ai singoli eventi.
- **Esportazione dei dati**: Capacità di esportare report e visualizzazioni in formati standard.

La loro utilità si manifesta in diversi contesti operativi, dal monitoraggio quotidiano della sicurezza all'analisi forense post-incidente, fornendo un supporto visivo che facilita l'identificazione di pattern complessi difficilmente rilevabili attraverso l'analisi manuale dei log.

6.3 Dashboard preconfigurate

Wazuh offre un set completo di dashboard preconfigurate che coprono diversi ambiti di monitoraggio della sicurezza. Queste dashboard "pronte all'uso" rappresentano un notevole vantaggio, consentendo un'operatività immediata senza necessità di configurazioni complesse. Tra le principali dashboard preconfigurate troviamo:

- **Security events**: Panoramica generale degli eventi di sicurezza con classificazione per gravità, origine e tipo.
- **Integrity monitoring**: Monitoraggio delle modifiche a file e configurazioni critiche del sistema.
- **Vulnerability detection**: Visualizzazione delle vulnerabilità rilevate sui sistemi monitorati.
- **PCI DSS**: Dashboard specifiche per il monitoraggio della conformità agli standard PCI DSS.
- **NIST 800-53**: Visualizzazioni orientate alla conformità con lo standard NIST.
- **MITRE ATT&CK**: Rappresentazione degli alert correlati alle tecniche MITRE.
- **Agent Status**: Monitoraggio dello stato degli agenti Wazuh distribuiti nell'infrastruttura.

Queste dashboard preconfigurate offrono un punto di partenza efficace che può essere ulteriormente personalizzato in base alle esigenze specifiche dell'organizzazione.

6.4 Creazione di dashboard personalizzate per il monitoraggio avanzato

Nel contesto del SOC di laboratorio, sono state sviluppate dashboard personalizzate per rispondere a esigenze specifiche di monitoraggio, con particolare attenzione all'integrazione con Snort IDS e VirusTotal e agli alert generati dalle attività Kea-Dhcp4 e Captive Portal.

6.4.1 Dashboard Snort IDS

La dashboard dedicata a Snort IDS rappresenta uno strumento fondamentale per il monitoraggio del traffico di rete e l'identificazione di potenziali intrusioni.

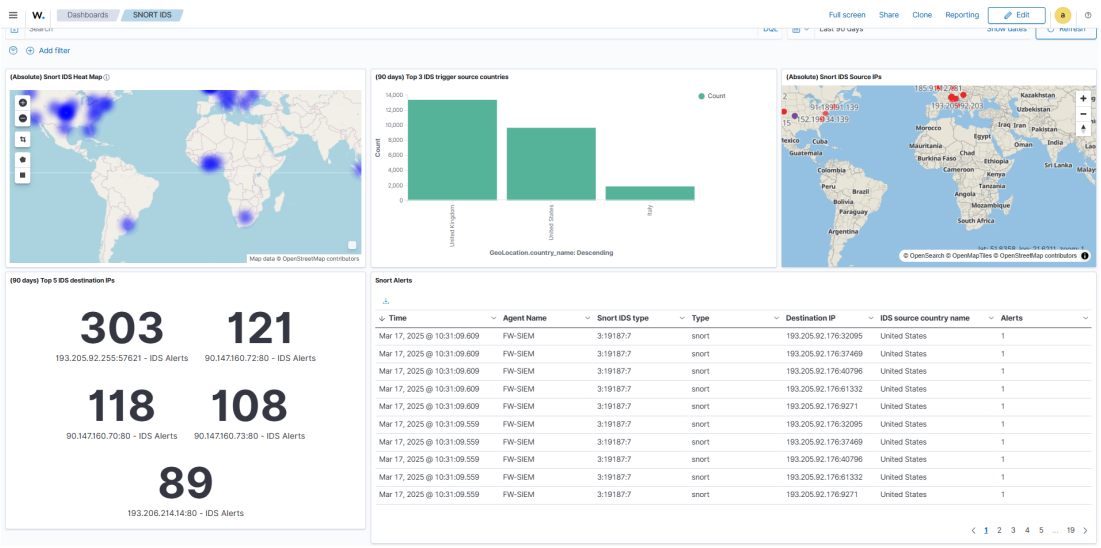


Figura 6.1: Dashboard Snort con geolocalizzazione degli attacchi e statistiche sulle intrusioni rilevate

Come visibile nella figura 6.1, questa dashboard include:

- Mappa di calore (heatmap) con geolocalizzazione degli IP sorgente degli attacchi
- Mappa con la geolocalizzazione e la posizione sulla mappa degli IP sorgente degli attacchi
- Istogramma relativo alle 3 nazioni più attive nella generazione di IDS alerts
- Elenco dei 5 indirizzi IP più frequentemente coinvolti in attività sospette
- Tabella di visualizzazione degli alert inerenti a Snort

La geolocalizzazione degli indirizzi IP, implementata nativamente in Wazuh, offre un'immediata visualizzazione della distribuzione geografica delle minacce, consentendo di identificare rapidamente pattern di attacco provenienti da specifiche regioni.

6.4.2 Dashboard di riepilogo e integrazione con VirusTotal

La seconda dashboard personalizzata fornisce una panoramica generale dello stato di sicurezza del sistema, con focus particolare sull'integrazione con VirusTota per il rilevamento di malware.

Gli elementi principali di questa dashboard, illustrati nella figura 6.2, includono:

- Grafico a torta (piechart) delle top 10 tipologie di alert rilevati
- Contatore del numero totale di alert nell'intervallo di tempo selezionato
- Sezione dedicata agli "hits" di VirusTotal, evidenziando quando il servizio ha rilevato minacce serie

Particolarmente significativa è la visualizzazione degli "hits" di VirusTotal, che evidenzia i file potenzialmente dannosi identificati attraverso l'integrazione con questo servizio di threat intelligence. Questa funzionalità consente di prioritizzare efficacemente

la risposta agli incidenti, concentrando l'attenzione sulle minacce concrete validate da molteplici motori antivirus.

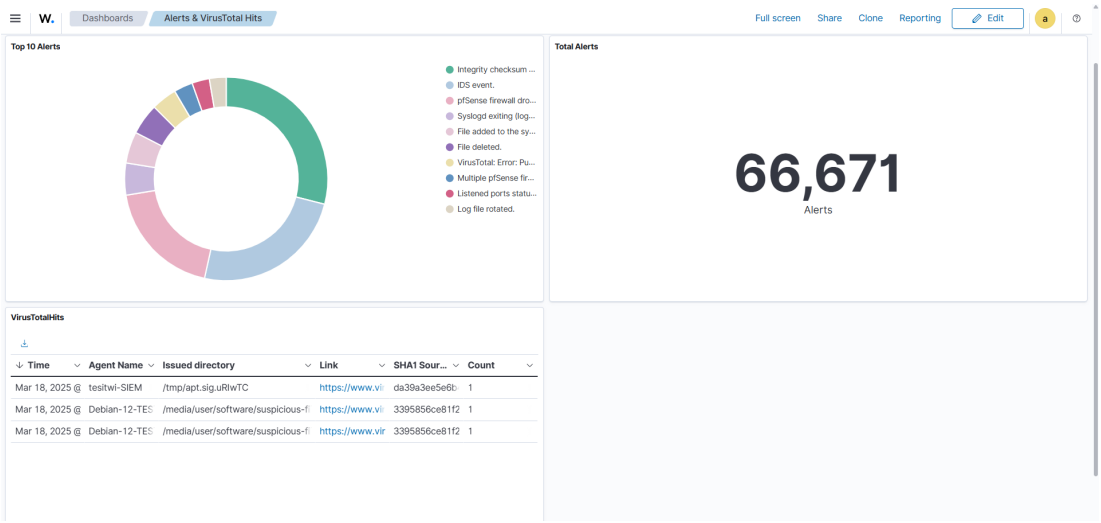


Figura 6.2: Dashboard di riepilogo con statistiche sugli alert e integrazione VirusTotal

6.4.3 Dashboard di monitoraggio Kea-DHCP4

Un'altra dashboard personalizzata è stata sviluppata per monitorare l'attività del servizio Kea-DHCP4, fornendo una visione chiara delle operazioni DHCP all'interno della rete.

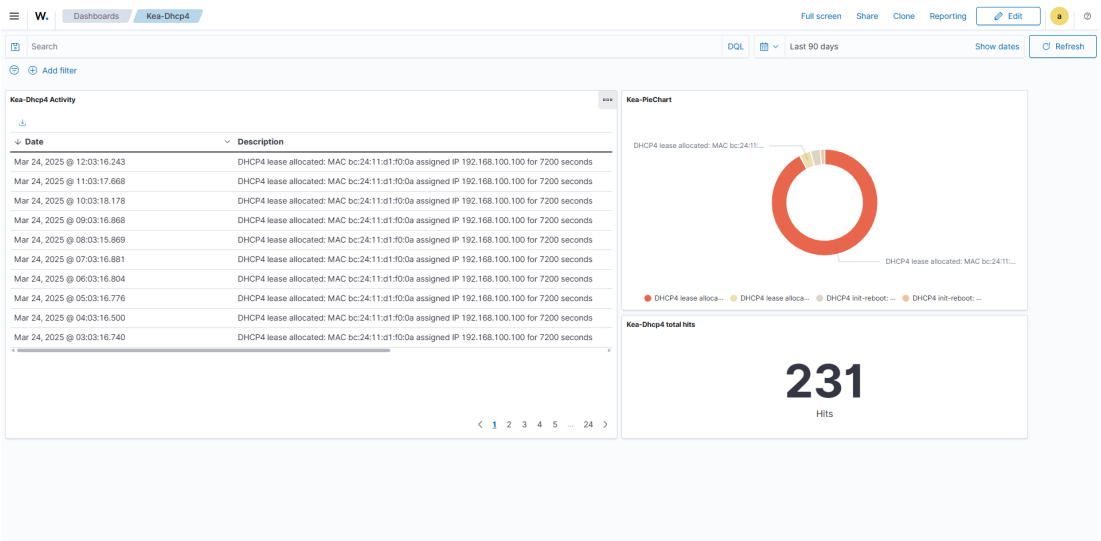


Figura 6.3: Dashboard di monitoraggio delle attività Kea-DHCP4

Come illustrato nella figura 6.3, la dashboard si compone di tre elementi principali:

- Un grafico a torta (piechart) che mostra la distribuzione delle diverse tipologie di attività DHCP rilevate
- Un contatore che indica il numero totale di eventi registrati nell'intervallo di tempo selezionato

- Un log dettagliato degli alert generati dalle attività DHCP monitorate, utile per individuare anomalie nel rilascio e rinnovo degli indirizzi IP

Questa dashboard è essenziale per rilevare eventuali comportamenti anomali nel servizio DHCP, come assegnazioni non autorizzate o fallimenti nella gestione degli indirizzi, contribuendo a migliorare la sicurezza e la stabilità della rete.

6.4.4 Dashboard di monitoraggio Captive Portal

Per il monitoraggio degli accessi alla rete tramite il Captive Portal, è stata realizzata una dashboard dedicata, volta a evidenziare le attività di autenticazione e le eventuali anomalie nei tentativi di accesso.

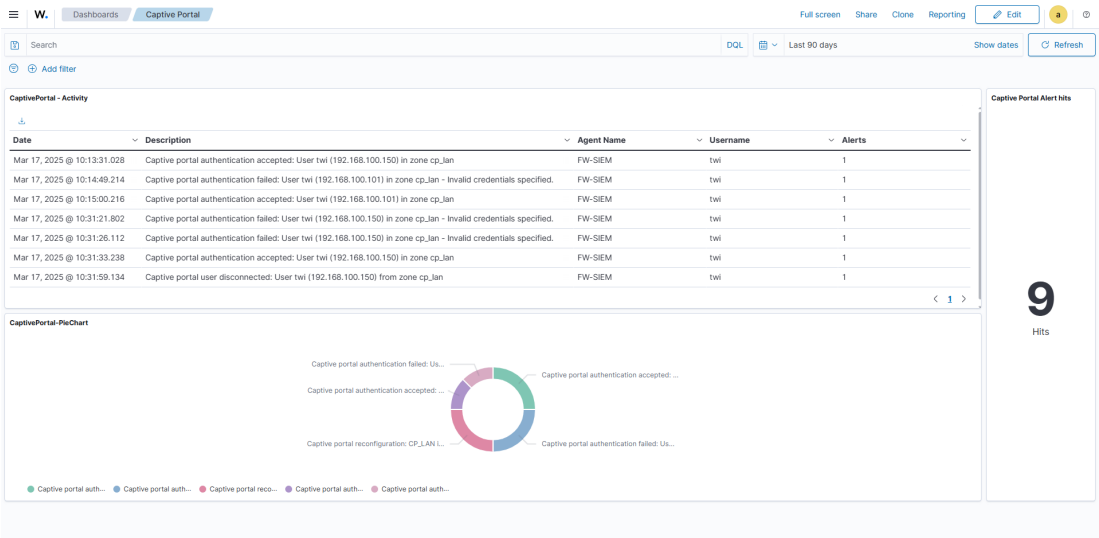


Figura 6.4: Dashboard di monitoraggio delle attività del Captive Portal

Come mostrato nella figura 6.4, la dashboard è composta da:

- Un grafico a torta (piechart) che rappresenta le diverse tipologie di attività registrate dal Captive Portal
- Un contatore che riporta il numero totale di accessi e tentativi di autenticazione registrati nel periodo selezionato
- Un log dettagliato degli alert generati, utile per identificare accessi non autorizzati o tentativi di autenticazione falliti

Questa dashboard è particolarmente utile per analizzare le modalità di accesso alla rete e identificare potenziali attività sospette, garantendo un controllo più efficace sugli utenti che tentano di connettersi tramite il Captive Portal.

6.5 Esempio di configurazione di una dashboard.

La configurazione di una dashboard personalizzata in Wazuh rappresenta un processo strutturato che consente di adattare la visualizzazione alle specifiche esigenze di monitoraggio. Di seguito viene illustrato il procedimento passo-passo per la creazione di una dashboard simile a quella utilizzata per il monitoraggio del servizio Kea-DHCP4 (figura 6.3).

6.5.1 Prerequisiti e accesso

Prima di iniziare, è necessario:

- Accedere all'interfaccia web di Wazuh con credenziali amministrative
- Verificare che gli agenti stiano correttamente inviando i log desiderati
- Assicurarsi che Wazuh stia indicizzando correttamente i dati attraverso l'Indexer
- Studiarsi bene la struttura degli alert generati dai log, in modo da poter sfruttare dei parametri comuni fra le varie istanze, riconducibili a uno stesso soggetto (e.g. id delle regole, indirizzi IP sorgente o di destinazione, decoder, ecc..)

6.5.2 Procedimento di configurazione

1. **Accesso alla sezione Dashboard**: Dalla barra laterale dell'interfaccia di Wazuh, selezionare "`Dashboard`".
2. **Creazione di una nuova dashboard**: Cliccare sul pulsante "`Create dashboard`" nell'angolo superiore destro dell'interfaccia.
3. **Aggiunta di visualizzazioni**:
   - Per aggiungere il grafico a torta (piechart), selezionare "`Create new`" e quindi `Pie`.
   - Configurare la metrica selezionando il campo "`rule.description`" come dimensione e applicare un filtro per mostrare solo i log relativi a "`kea-dhcp4`". (es. `decoder.parent`)
   - Utilizzare l'opzione "`Split slices`" per dividere il grafico in base ai tipi di attività (e.g. `dhcp4 lease allocated`, `dhcp4 init-reboot`).
4. **Creazione del contatore di eventi**:
   - Aggiungere un elemento "`Metric`" tramite l'opzione "`Create new`".
   - Configurare la metrica come "`Count of hits`" e applicare il filtro "`decoder.parent: kea-dhcp4`".
   - Personalizzare l'etichetta come "Kea-Dhcp4 total hits".
5. **Configurazione della tabella di log**:
   - Aggiungere un elemento "`Data table`" scegliendo "`Create new`".
   - Selezionare le colonne da visualizzare: "`timestamp`", "`data.allocated.ip`", "`rule.description`".

- Applicare il filtro "`decoder.parent: kea-dhcp4`" per mostrare solo i log pertinenti.
- Ordinare la tabella in ordine cronologico inverso per visualizzare gli eventi più recenti in cima.

6. **Disposizione degli elementi**: Trascinare e ridimensionare i vari componenti per ottenere un layout ottimale simile a quello mostrato nella figura 6.3.
7. **Configurazione delle opzioni temporali**:
   - Impostare l'intervallo di tempo predefinito per la dashboard (es. `Last 90 days`).
   - Attivare l'aggiornamento automatico selezionando un intervallo appropriato (es. `5 minutes`).
8. **Salvataggio della dashboard**: Cliccare su "`Save`" nell'angolo superiore destro, assegnare un nome descrittivo come "Kea-Dhcp4 Activity Monitoring" e aggiungere una breve descrizione.

6.5.3 Ottimizzazione e manutenzione

Una volta creata la dashboard, è consigliabile:

- Perfezionare i filtri per ridurre i falsi positivi
- Aggiungere annotazioni per eventi significativi
- Configurare alert basati su soglie specifiche (e.g. numero anomalo di richieste DHCP in un breve periodo)
- Eseguire regolarmente backup delle configurazioni delle dashboard attraverso l'opzione di esportazione

Questa procedura di configurazione può essere adattata per creare dashboard personalizzate per qualsiasi tipo di log o evento monitorato da Wazuh, seguendo lo stesso approccio metodico ma modificando i filtri e le visualizzazioni in base alle specifiche esigenze di monitoraggio.

6.6 Valore operativo delle dashboard

L'implementazione delle dashboard di Wazuh ha dimostrato un notevole valore operativo nel contesto del nostro SOC domestico. La trasformazione di dati grezzi in visualizzazioni intuitive ha notevolmente ridotto il tempo necessario per l'identificazione e l'analisi delle minacce, consentendo:

- Riduzione dei tempi di risposta agli incidenti grazie all'immediata visibilità sugli eventi critici
- Miglioramento della consapevolezza situazionale attraverso la correlazione visuale di eventi apparentemente sconnessi
- Ottimizzazione dell'allocazione delle risorse di sicurezza, concentrandosi sulle minacce più significative

- Semplificazione della reportistica e della documentazione degli incidenti di sicurezza

In conclusione, le dashboard di Wazuh rappresentano non solo uno strumento di visualizzazione, ma un vero e proprio moltiplicatore di efficacia per le operazioni di sicurezza, trasformando dati complessi in informazioni azionabili e supportando efficacemente il processo decisionale nella gestione degli incidenti di sicurezza.

7. Conclusioni e sviluppi futuri

7.1 Sintesi dei risultati ottenuti

La realizzazione di questo progetto ha dimostrato la fattibilità e l'efficacia di un SOC, in ambiente domestico e/o nelle pmi, basato su tecnologie open-source, come Wazuh, pfSense e Snort. L'implementazione ha raggiunto gli obiettivi principali di:

- Creare un sistema di monitoraggio della sicurezza in grado di rilevare minacce in tempo reale
- Introdurre un framework operativo per la gestione degli eventi di sicurezza a basso costo
- Monitoraggio dell'integrità dei file system
- Dimostrare l'applicabilità di strumenti enterprise in contesti non aziendali

I risultati ottenuti mostrano che è possibile ottenere una buona copertura delle funzionalità di sicurezza con un investimento minimo, sfruttando le potenzialità delle tecnologie open-source. La dashboard di Wazuh ha fornito una visibilità completa sull'infrastruttura monitorata, consentendo di identificare e analizzare rapidamente le minacce rilevate.

7.2 Discussione sulle sfide incontrate

Durante la realizzazione del progetto, sono state incontrate diverse sfide che hanno richiesto soluzioni creative e un approccio pragmatico:

- **Limiti hardware**: L'utilizzo di hardware datato ha introdotto considerevoli impedimenti nel setup del laboratorio. Per sopperire a queste problematiche è stato necessario consultare numerosi documenti, forum e blog che alla fine hanno fatto sì che l'ambiente virtuale prendesse forma.
- **Assenza di automazione SOAR**: La mancanza di un sistema di orchestrazione e automazione della risposta agli incidenti (SOAR) ha richiesto interventi manuali per la gestione degli alert, aumentando i tempi di risposta.
- **Complessità della configurazione avanzata**: La personalizzazione delle regole di Snort e l'ottimizzazione delle impostazioni di Wazuh hanno richiesto un notevole impegno in termini di tempo e competenze tecniche.

Queste sfide hanno evidenziato l'importanza di una pianificazione accurata e di una continua formazione per gestire al meglio le tecnologie di sicurezza open-source.

7.3 Proposte per sviluppi futuri

Per migliorare ulteriormente il sistema e renderlo più efficace, si propongono i seguenti sviluppi futuri:

1. **Implementazione di un sistema SOAR open-source**: L'integrazione di strumenti come TheHive e Cortex potrebbe automatizzare significativamente le procedure di risposta agli incidenti, riducendo i tempi di intervento e migliorando l'efficienza operativa.

2. **Ottimizzazione delle regole di correlazione degli eventi**: Un ulteriore lavoro di ottimizzazione delle regole di correlazione potrebbe ridurre i falsi positivi e migliorare la precisione del sistema di rilevamento delle minacce.

3. **Estensione della copertura dei log**: Implementare una politica di log più completa, inclusa la gestione del traffico di rete, potrebbe fornire una visibilità più approfondita sugli eventi di sicurezza.

4. **Sviluppo di dashboard personalizzate per specifiche esigenze**: Creare dashboard focalizzate su particolari tipologie di minacce (e.g. IP spoofing[61], attacchi DoS[62] ecc..) su specifiche aree di interesse potrebbe migliorare ulteriormente la capacità di analisi e risposta.

Questi sviluppi futuri rappresentano un'opportunità per migliorare ulteriormente l'efficacia del sistema, aumentando la sua capacità di adattamento a nuove minacce e migliorando la gestione degli incidenti di sicurezza.

A. Configurazioni

A.1 XML di configurazione del Wazuh-Manager

```
<!--
  Wazuh - Manager - Default configuration for ubuntu 24.04
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <global>
    <jsonout_output>yes</jsonout_output>
    <alerts_log>yes</alerts_log>
    <logall>no</logall>
    <logall_json>no</logall_json>
    <email_notification>no</email_notification>
    <smtp_server>smtp.example.wazuh.com</smtp_server>
    <email_from>wazuh@example.wazuh.com</email_from>
    <email_to>recipient@example.wazuh.com</email_to>
    <email_maxperhour>12</email_maxperhour>
    <email_log_source>alerts.log</email_log_source>
    <agents_disconnection_time>10m</agents_disconnection_time>
    <agents_disconnection_alert_time>0</agents_disconnection_alert_time>
    <update_check>yes</update_check>
  </global>

  <alerts>
    <log_alert_level>3</log_alert_level>
    <email_alert_level>12</email_alert_level>
  </alerts>

  <!-- Choose between "plain", "json", or "plain,json" for the format of internal logs
       -->
  <logging>
    <log_format>plain</log_format>
  </logging>

  <remote>
    <connection>secure</connection>
    <port>1514</port>
    <protocol>tcp</protocol>
    <queue_size>131072</queue_size>
  </remote>

  <remote>
    <connection>syslog</connection>
    <port>5514</port>
    <protocol>tcp</protocol>
    <allowed-ips>192.168.100.1</allowed-ips>
    <local_ip>192.168.100.150</local_ip>
  </remote>

  <remote>
    <connection>syslog</connection>
    <port>5514</port>
    <protocol>udp</protocol>
```

```
    <allowed-ips>192.168.100.1</allowed-ips>
    <local_ip>192.168.100.150</local_ip>
  </remote>

  <!-- Policy monitoring -->
  <rootcheck>
    <disabled>no</disabled>
    <check_files>yes</check_files>
    <check_trojans>yes</check_trojans>
    <check_dev>yes</check_dev>
    <check_sys>yes</check_sys>
    <check_pids>yes</check_pids>
    <check_ports>yes</check_ports>
    <check_if>yes</check_if>

    <!-- Frequency that rootcheck is executed - every 12 hours -->
    <frequency>43200</frequency>

    <rootkit_files>etc/rootcheck/rootkit_files.txt</rootkit_files>
    <rootkit_trojans>etc/rootcheck/rootkit_trojans.txt</rootkit_trojans>

    <skip_nfs>yes</skip_nfs>

    <ignore>/var/lib/containerd</ignore>
    <ignore>/var/lib/docker/overlay2</ignore>
  </rootcheck>

   <wodle name="cis-cat">
    <disabled>yes</disabled>
    <timeout>1800</timeout>
    <interval>1d</interval>
    <scan-on-start>yes</scan-on-start>

    <java_path>wodles/java</java_path>
    <ciscat_path>wodles/ciscat</ciscat_path>
  </wodle>

  <!-- Osquery integration -->
  <wodle name="osquery">
    <disabled>yes</disabled>
    <run_daemon>yes</run_daemon>
    <log_path>/var/log/osquery/osqueryd.results.log</log_path>
    <config_path>/etc/osquery/osquery.conf</config_path>
    <add_labels>yes</add_labels>
  </wodle>

  <!-- System inventory -->
  <wodle name="syscollector">
    <disabled>no</disabled>
    <interval>1h</interval>
    <scan_on_start>yes</scan_on_start>
    <hardware>yes</hardware>
    <os>yes</os>
    <network>yes</network>
    <packages>yes</packages>
    <ports all="no">yes</ports>
    <processes>yes</processes>

    <!-- Database synchronization settings -->
    <synchronization>
      <max_eps>10</max_eps>
    </synchronization>
  </wodle>

  <sca>
    <enabled>yes</enabled>
    <scan_on_start>yes</scan_on_start>
    <interval>12h</interval>
    <skip_nfs>yes</skip_nfs>
  </sca>
```

```
123
124   <vulnerability-detection>
125     <enabled>yes</enabled>
126     <index-status>yes</index-status>
127     <feed-update-interval>60m</feed-update-interval>
128   </vulnerability-detection>
129
130   <indexer>
131     <enabled>yes</enabled>
132     <hosts>
133       <host>https://127.0.0.1:9200</host>
134     </hosts>
135     <ssl>
136       <certificate_authorities>
137         <ca>/etc/filebeat/certs/root-ca.pem</ca>
138       </certificate_authorities>
139       <certificate>/etc/filebeat/certs/wazuh-server.pem</certificate>
140       <key>/etc/filebeat/certs/wazuh-server-key.pem</key>
141     </ssl>
142   </indexer>
143
144   <!-- File integrity monitoring -->
145   <syscheck>
146     <disabled>no</disabled>
147
148     <!-- Frequency that syscheck is executed default every 12 hours -->
149     <frequency>43200</frequency>
150
151     <scan_on_start>yes</scan_on_start>
152
153     <!-- Generate alert when new file detected -->
154     <alert_new_files>yes</alert_new_files>
155
156     <!-- Don't ignore files that change more than 'frequency' times -->
157     <auto_ignore frequency="10" timeframe="3600">no</auto_ignore>
158
159     <!-- Directories to check (perform all possible verifications) -->
160     <directories>/etc,/usr/bin,/usr/sbin</directories>
161     <directories>/bin,/sbin,/boot</directories>
162
163     <!-- Files/directories to ignore -->
164     <ignore>/etc/mtab</ignore>
165     <ignore>/etc/hosts.deny</ignore>
166     <ignore>/etc/mail/statistics</ignore>
167     <ignore>/etc/random-seed</ignore>
168     <ignore>/etc/random.seed</ignore>
169     <ignore>/etc/adjtime</ignore>
170     <ignore>/etc/httpd/logs</ignore>
171     <ignore>/etc/utmpx</ignore>
172     <ignore>/etc/wtmpx</ignore>
173     <ignore>/etc/cups/certs</ignore>
174     <ignore>/etc/dumpdates</ignore>
175     <ignore>/etc/svc/volatile</ignore>
176
177     <!-- File types to ignore -->
178     <ignore type="sregex">.log$|.swp$</ignore>
179
180     <!-- Check the file, but never compute the diff -->
181      <nodiff>/etc/ssl/private.key</nodiff>
182
183     <skip_nfs>yes</skip_nfs>
184     <skip_dev>yes</skip_dev>
185     <skip_proc>yes</skip_proc>
186     <skip_sys>yes</skip_sys>
187
188     <!-- Nice value for Syscheck process -->
189     <process_priority>10</process_priority>
190
191     <!-- Maximum output throughput -->
192     <max_eps>50</max_eps>
```

```
193
194     <!-- Database synchronization settings -->
195     <synchronization>
196       <enabled>yes</enabled>
197       <interval>5m</interval>
198       <max_eps>10</max_eps>
199     </synchronization>
200   </syscheck>
201
202   <!-- Active response -->
203   <global>
204     <white_list>127.0.0.1</white_list>
205     <white_list>^localhost.localdomain$</white_list>
206     <white_list>127.0.0.53</white_list>
207   </global>
208
209   <command>
210     <name>disable-account</name>
211     <executable>disable-account</executable>
212     <timeout_allowed>yes</timeout_allowed>
213   </command>
214
215   <command>
216     <name>restart-wazuh</name>
217     <executable>restart-wazuh</executable>
218   </command>
219
220   <command>
221     <name>firewall-drop</name>
222     <executable>firewall-drop</executable>
223     <timeout_allowed>yes</timeout_allowed>
224   </command>
225
226   <command>
227     <name>host-deny</name>
228     <executable>host-deny</executable>
229     <timeout_allowed>yes</timeout_allowed>
230   </command>
231
232   <command>
233    <name>route-null</name>
234     <executable>route-null</executable>
235     <timeout_allowed>yes</timeout_allowed>
236   </command>
237
238   <command>
239     <name>win_route-null</name>
240     <executable>route-null.exe</executable>
241     <timeout_allowed>yes</timeout_allowed>
242   </command>
243
244   <command>
245     <name>netsh</name>
246     <executable>netsh.exe</executable>
247     <timeout_allowed>yes</timeout_allowed>
248   </command>
249
250   <!--
251   <active-response>
252     active-response options here
253   </active-response>
254   -->
255
256   <!-- Log analysis -->
257   <localfile>
258     <log_format>command</log_format>
259     <command>df -P</command>
260     <frequency>360</frequency>
261   </localfile>
262
```

```
263   <localfile>
264     <log_format>full_command</log_format>
265     <command>netstat -tulpn | sed 's/\([[:alnum:]]\+\)\ \+[[:digit:]]
266     \+\ \+[[:digit:]]\+\ \+\(.*\):\([[:digit:]]*\)\ \+>
267     <alias>netstat listening ports</alias>
268     <frequency>360</frequency>
269   </localfile>
270
271   <localfile>
272     <log_format>full_command</log_format>
273     <command>last -n 20</command>
274     <frequency>360</frequency>
275   </localfile>
276
277   <ruleset>
278     <!-- Default ruleset -->
279     <decoder_dir>ruleset/decoders</decoder_dir>
280     <rule_dir>ruleset/rules</rule_dir>
281     <rule_exclude>0215-policy_rules.xml</rule_exclude>
282     <list>etc/lists/audit-keys</list>
283     <list>etc/lists/amazon/aws-eventnames</list>
284     <list>etc/lists/security-eventchannel</list>
285     <!-- User-defined ruleset -->
286     <decoder_dir>etc/decoders</decoder_dir>
287     <rule_dir>etc/rules</rule_dir>
288   </ruleset>
289
290   <rule_test>
291     <enabled>yes</enabled>
292     <threads>1</threads>
293     <max_sessions>64</max_sessions>
294     <session_timeout>15m</session_timeout>
295   </rule_test>
296
297   <!-- Configuration for wazuh-authd -->
298   <auth>
299     <disabled>no</disabled>
300     <port>1515</port>
301     <use_source_ip>no</use_source_ip>
302     <purge>yes</purge>
303     <use_password>no</use_password>
304     <ciphers>HIGH:!ADH:!EXP:!MD5:!RC4:!3DES:!CAMELLIA:@STRENGTH</ciphers>
305     <!-- <ssl_agent_ca></ssl_agent_ca> -->
306     <ssl_verify_host>no</ssl_verify_host>
307     <ssl_manager_cert>etc/sslmanager.cert</ssl_manager_cert>
308     <ssl_manager_key>etc/sslmanager.key</ssl_manager_key>
309     <ssl_auto_negotiate>no</ssl_auto_negotiate>
310   </auth>
311   <cluster>
312     <name>wazuh</name>
313     <node_name>node01</node_name>
314     <node_type>master</node_type>
315     <key></key>
316     <port>1516</port>
317     <bind_addr>0.0.0.0</bind_addr>
318     <nodes>
319        <node>NODE_IP</node>
320     </nodes>
321     <hidden>no</hidden>
322     <disabled>yes</disabled>
323   </cluster>
324
325 </ossec_config>
326
327 <ossec_config>
328   <localfile>
329     <log_format>journald</log_format>
330     <location>journald</location>
331   </localfile>
332
```

```
333   <localfile>
334     <log_format>syslog</log_format>
335     <location>/var/ossec/logs/active-responses.log</location>
336   </localfile>
337
338   <localfile>
339     <log_format>syslog</log_format>
340     <location>/var/ossec/logs/active-responses.log</location>
341   </localfile>
342
343   <localfile>
344     <log_format>syslog</log_format>
345     <location>/var/log/dpkg.log</location>
346   </localfile>
347
348 </ossec_config>
```

Codice A.1: File di configurazione del Wazuh-Manager, ossec.conf, su VM Ubuntu 24.04.2 LTS

A.2 XML di configurazione dell'agent Wazuh su pfSense

```
<!--
  Wazuh - Agent - Default configuration for bsd 14
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->


<ossec_config>
  <client>
    <server>
      <address>192.168.100.150</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>freebsd, freebsd14</config-profile>
    <crypto_method>aes</crypto_method>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <enrollment>
      <enabled>yes</enabled>
      <agent_name>FW-SIEM</agent_name>
      <groups>pfSense</groups>
      <authorization_pass_path>etc/authd.pass</authorization_pass_path>
    </enrollment>
  </client>

  <client_buffer>
    <!-- Agent buffer options -->
    <disabled>no</disabled>
    <queue_size>5000</queue_size>
    <events_per_second>500</events_per_second>
  </client_buffer>

  <sca>
    <enabled>yes</enabled>
    <scan_on_start>yes</scan_on_start>
    <interval>12h</interval>
    <skip_nfs>yes</skip_nfs>
  </sca>

  <!-- Policy monitoring -->
  <rootcheck>
    <disabled>no</disabled>

    <!-- Frequency that rootcheck is executed - every 12 hours -->
    <frequency>43200</frequency>

    <rootkit_files>/var/ossec/etc/shared/rootkit_files.txt</rootkit_files>
    <rootkit_trojans>/var/ossec/etc/shared/rootkit_trojans.txt</rootkit_trojans>

    <system_audit>/var/ossec/etc/shared/system_audit_rcl.txt</system_audit>
    <system_audit>/var/ossec/etc/shared/system_audit_ssh.txt</system_audit>
    <system_audit>/var/ossec/etc/shared/cis_freebsd14.yml</system_audit>

    <skip_nfs>yes</skip_nfs>
  </rootcheck>

  <wodle name="open-scap">
    <disabled>yes</disabled>
    <timeout>1800</timeout>
    <interval>1d</interval>
    <scan-on-start>yes</scan-on-start>

    <content type="xccdf" path="ssg-debian-8-ds.xml">
      <profile>xccdf_org.ssgproject.content_profile_common</profile>
    </content>
    <content type="oval" path="cve-debian-oval.xml"/>
```

```
  </wodle>

  <wodle name="syscollector">
    <disabled>no</disabled>
    <interval>1h</interval>
    <scan_on_start>yes</scan_on_start>
    <hardware>yes</hardware>
    <os>yes</os>
    <network>yes</network>

    <!-- Database synchronization settings -->
    <synchronization>
      <max_eps>10</max_eps>
    </synchronization>
  </wodle>

  <!-- File integrity monitoring -->
  <syscheck>
    <disabled>no</disabled>

    <!-- Frequency that syscheck is executed default every 12 hours -->
    <frequency>43200</frequency>

    <scan_on_start>yes</scan_on_start>

    <!-- Directories to check (perform all possible verifications) -->
    <directories>/etc,/usr/bin,/usr/sbin</directories>
    <directories>/bin,/sbin,/boot</directories>

    <!-- Files/directories to ignore -->
    <ignore>/etc/mtab</ignore>
    <ignore>/etc/hosts.deny</ignore>
    <ignore>/etc/mail/statistics</ignore>
    <ignore>/etc/random-seed</ignore>
    <ignore>/etc/random.seed</ignore>
    <ignore>/etc/adjtime</ignore>
    <ignore>/etc/httpd/logs</ignore>
    <ignore>/etc/utmpx</ignore>
    <ignore>/etc/wtmpx</ignore>
    <ignore>/etc/cups/certs</ignore>
    <ignore>/etc/dumpdates</ignore>
    <ignore>/etc/svc/volatile</ignore>
    <ignore>/sys/kernel/security</ignore>
    <ignore>/sys/kernel/debug</ignore>

    <!-- File types to ignore -->
    <ignore type="sregex">.log$|.swp$</ignore>

    <!-- Check the file, but never compute the diff -->
    <nodiff>/etc/ssl/private.key</nodiff>

    <skip_nfs>yes</skip_nfs>
    <skip_dev>yes</skip_dev>
    <skip_proc>yes</skip_proc>
    <skip_sys>yes</skip_sys>

    <!-- Nice value for Syscheck process -->
    <process_priority>10</process_priority>

    <!-- Maximum output throughput -->
    <max_eps>50</max_eps>

    <!-- Database synchronization settings -->
    <synchronization>
      <enabled>yes</enabled>
      <interval>5m</interval>
      <max_eps>10</max_eps>
    </synchronization>
  </syscheck>

```

```
139   <!-- Log analysis -->
140   <localfile>
141     <log_format>syslog</log_format>
142     <location>/var/ossec/logs/active-responses.log</location>
143   </localfile>
144
145   <localfile>
146     <log_format>syslog</log_format>
147     <location>/var/log/messages</location>
148   </localfile>
149
150   <localfile>
151     <log_format>syslog</log_format>
152     <location>/var/log/auth.log</location>
153   </localfile>
154
155   <localfile>
156     <log_format>syslog</log_format>
157     <location>/var/log/cron</location>
158   </localfile>
159
160   <localfile>
161     <log_format>syslog</log_format>
162     <location>/var/log/daemon.log</location>
163   </localfile>
164
165   <localfile>
166     <log_format>syslog</log_format>
167     <location>/var/log/debug.log</location>
168   </localfile>
169
170   <localfile>
171     <log_format>syslog</log_format>
172     <location>/var/log/userlog</location>
173   </localfile>
174
175   <localfile>
176     <log_format>command</log_format>
177     <command>df -P</command>
178     <frequency>360</frequency>
179   </localfile>
180
181   <localfile>
182     <log_format>full_command</log_format>
183     <command>(netstat -n -f inet && netstat -n -f inet) | grep -e "udp" -e "tcp" | sed
          's/\([[:alnum:]]*\)\ *[[:digit:]]*\ *[[:digit:]]*\ *\([[:digit:]\.]*\)\.\([[:
        digit:]]*\)\ *\([[:digit:]\.]*\).*/\1 \2 == \3 == \4/' | sort -k4 -g | sed 's/
          == \(.*\) ==/.\1/'</command>
184     <frequency>360</frequency>
185   </localfile>
186
187   <localfile>
188     <log_format>full_command</log_format>
189     <command>last -n 5</command>
190     <frequency>360</frequency>
191   </localfile>
192
193   <!-- Active response -->
194   <active-response>
195     <disabled>no</disabled>
196   </active-response>
197
198 </ossec_config>
```

Codice A.2: File di configurazione dell'agent Wazuh, ossec.conf, su VM pfSense v2.7.2-RELEASE

A.3 XML di configurazione dell'agent Wazuh su Debian

```
<!--
  Wazuh - Agent - Default configuration for debian 12
  More info at: https://documentation.wazuh.com
  Mailing list: https://groups.google.com/forum/#!forum/wazuh
-->

<ossec_config>
  <client>
    <server>
      <address>193.205.92.229</address>
      <port>1514</port>
      <protocol>tcp</protocol>
    </server>
    <config-profile>debian, debian12</config-profile>
    <notify_time>10</notify_time>
    <time-reconnect>60</time-reconnect>
    <auto_restart>yes</auto_restart>
    <crypto_method>aes</crypto_method>
    <enrollment>
      <enabled>yes</enabled>
      <agent_name>Debian-12-TEST</agent_name>
      <groups>WAN</groups>
      <authorization_pass_path>etc/authd.pass</authorization_pass_path>
    </enrollment>
  </client>

  <client_buffer>
    <!-- Agent buffer options -->
    <disabled>no</disabled>
    <queue_size>5000</queue_size>
    <events_per_second>500</events_per_second>
  </client_buffer>

  <!-- Policy monitoring -->
  <rootcheck>
    <disabled>no</disabled>
    <check_files>yes</check_files>
    <check_trojans>yes</check_trojans>
    <check_dev>yes</check_dev>
    <check_sys>yes</check_sys>
    <check_pids>yes</check_pids>
    <check_ports>yes</check_ports>
    <check_if>yes</check_if>

    <!-- Frequency that rootcheck is executed - every 12 hours -->
    <frequency>43200</frequency>

    <rootkit_files>etc/shared/rootkit_files.txt</rootkit_files>
    <rootkit_trojans>etc/shared/rootkit_trojans.txt</rootkit_trojans>

    <skip_nfs>yes</skip_nfs>

    <ignore>/var/lib/containerd</ignore>
    <ignore>/var/lib/docker/overlay2</ignore>
  </rootcheck>

  <wodle name="cis-cat">
    <disabled>yes</disabled>
    <timeout>1800</timeout>
    <interval>1d</interval>
    <scan-on-start>yes</scan-on-start>
    <java_path>wodles/java</java_path>
    <ciscat_path>wodles/ciscat</ciscat_path>
  </wodle>

  <!-- Osquery integration -->
  <wodle name="osquery">
    <disabled>yes</disabled>
```

```
    <run_daemon>yes</run_daemon>
    <log_path>/var/log/osquery/osqueryd.results.log</log_path>
    <config_path>/etc/osquery/osquery.conf</config_path>
    <add_labels>yes</add_labels>
  </wodle>

  <!-- System inventory -->
  <wodle name="syscollector">
    <disabled>no</disabled>
    <interval>1h</interval>
    <scan_on_start>yes</scan_on_start>
    <hardware>yes</hardware>
    <os>yes</os>
    <network>yes</network>
    <packages>yes</packages>
    <ports all="no">yes</ports>
    <processes>yes</processes>
    <!-- Database synchronization settings -->
    <synchronization>
      <max_eps>10</max_eps>
    </synchronization>
  </wodle>

  <sca>
    <enabled>yes</enabled>
    <scan_on_start>yes</scan_on_start>
    <interval>12h</interval>
    <skip_nfs>yes</skip_nfs>
  </sca>

  <!-- File integrity monitoring -->
  <syscheck>
    <disabled>no</disabled>
    <!-- Frequency that syscheck is executed default every 12 hours -->
    <frequency>43200</frequency>
    <scan_on_start>yes</scan_on_start>
    <!-- Directories to check (perform all possible verifications) -->
    <directories>/etc,/usr/bin,/usr/sbin</directories>
    <directories>/bin,/sbin,/boot</directories>
    <!-- Files/directories to ignore -->
    <ignore>/etc/mtab</ignore>
    <ignore>/etc/hosts.deny</ignore>
    <ignore>/etc/mail/statistics</ignore>
    <ignore>/etc/random-seed</ignore>
    <ignore>/etc/random.seed</ignore>
    <ignore>/etc/adjtime</ignore>
    <ignore>/etc/httpd/logs</ignore>
    <ignore>/etc/utmpx</ignore>
    <ignore>/etc/wtmpx</ignore>
    <ignore>/etc/cups/certs</ignore>
    <ignore>/etc/dumpdates</ignore>
    <ignore>/etc/svc/volatile</ignore>
    <!-- File types to ignore -->
    <ignore type="sregex">.log$|.swp$</ignore>
    <!-- Check the file, but never compute the diff -->
    <nodiff>/etc/ssl/private.key</nodiff>
    <skip_nfs>yes</skip_nfs>
    <skip_dev>yes</skip_dev>
    <skip_proc>yes</skip_proc>
    <skip_sys>yes</skip_sys>

    <!-- Nice value for Syscheck process -->
    <process_priority>10</process_priority>

    <!-- Maximum output throughput -->
    <max_eps>50</max_eps>

    <!-- Database synchronization settings -->
    <synchronization>
      <enabled>yes</enabled>
```

```
    <interval>5m</interval>
    <max_eps>10</max_eps>
  </synchronization>
</syscheck>

<!-- Log analysis -->
<localfile>
  <log_format>command</log_format>
  <command>df -P</command>
  <frequency>360</frequency>
</localfile>

<localfile>
  <log_format>full_command</log_format>
  <command>netstat -tulpn | sed 's/\([[:alnum:]]\+\)\ \+[[:digit:]]\+\ \+[[:digit
      :]]\+\ \+\(.*\):\([[:digit:]]*\)\ \+\([0-9\.\:\*]\+\).\+\ \([[:digit:]]*\/[[:
      alnum:]\-]*\).*/\1 \2 == \3 == \4 \5/' | sort -k 4 -g | sed 's/ == \(.*\)
      ==/:\1/' | sed 1,2d</command>
  <alias>netstat listening ports</alias>
  <frequency>360</frequency>
</localfile>

<localfile>
  <log_format>full_command</log_format>
  <command>last -n 20</command>
  <frequency>360</frequency>
</localfile>

<!-- Active response -->
<active-response>
  <disabled>no</disabled>
  <ca_store>etc/wpk_root.pem</ca_store>
  <ca_verification>yes</ca_verification>
</active-response>

<!-- Choose between "plain", "json", or "plain,json" for the format of internal logs
    -->
<logging>
  <log_format>plain</log_format>
</logging>

</ossec_config>

<ossec_config>
  <localfile>
    <log_format>journald</log_format>
    <location>journald</location>
  </localfile>

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/ossec/logs/active-responses.log</location>
  </localfile>

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/dpkg.log</location>
  </localfile>

</ossec_config>
```

Codice A.3: File di configurazione dell'agent Wazuh, ossec.conf, su VM Debian 12

A.4 Esempio di configurazione delle azioni di monitoraggio

```
<sca>
   <enabled>yes</enabled>
   <scan_on_start>yes</scan_on_start>
   <interval>12h</interval>
   <skip_nfs>yes</skip_nfs>
 </sca>

 <!-- Policy monitoring -->
 <rootcheck>
   <disabled>no</disabled>

   <!-- Frequency that rootcheck is executed - every 12 hours -->
   <frequency>43200</frequency>

   <rootkit_files>/var/ossec/etc/shared/rootkit_files.txt</rootkit_files>
   <rootkit_trojans>/var/ossec/etc/shared/rootkit_trojans.txt</rootkit_trojans>

   <system_audit>/var/ossec/etc/shared/system_audit_rcl.txt</system_audit>
   <system_audit>/var/ossec/etc/shared/system_audit_ssh.txt</system_audit>
   <system_audit>/var/ossec/etc/shared/cis_freebsd14.yml</system_audit>

   <skip_nfs>yes</skip_nfs>
 </rootcheck>

 <wodle name="open-scap">
   <disabled>yes</disabled>
   <timeout>1800</timeout>
   <interval>1d</interval>
   <scan-on-start>yes</scan-on-start>

   <content type="xccdf" path="ssg-debian-8-ds.xml">
     <profile>xccdf_org.ssgproject.content_profile_common</profile>
   </content>
   <content type="oval" path="cve-debian-oval.xml"/>
 </wodle>

 <wodle name="syscollector">
   <disabled>no</disabled>
   <interval>1h</interval>
   <scan_on_start>yes</scan_on_start>
   <hardware>yes</hardware>
   <os>yes</os>
   <network>yes</network>

   <!-- Database synchronization settings -->
   <synchronization>
     <max_eps>10</max_eps>
   </synchronization>
 </wodle>

 <!-- File integrity monitoring -->
 <syscheck>
   <disabled>no</disabled>

   <!-- Frequency that syscheck is executed default every 12 hours -->
   <frequency>43200</frequency>

   <scan_on_start>yes</scan_on_start>

   <!-- Directories to check (perform all possible verifications) -->
   <directories>/etc,/usr/bin,/usr/sbin</directories>
   <directories>/bin,/sbin,/boot</directories>


   <!-- Check the file, but never compute the diff -->
   <nodiff>/etc/ssl/private.key</nodiff>
```

```
67
68      <skip_nfs>yes</skip_nfs>
69      <skip_dev>yes</skip_dev>
70      <skip_proc>yes</skip_proc>
71      <skip_sys>yes</skip_sys>
72
73      <!-- Nice value for Syscheck process -->
74      <process_priority>10</process_priority>
75
76      <!-- Maximum output throughput -->
77      <max_eps>50</max_eps>
78
79      <!-- Database synchronization settings -->
80      <synchronization>
81        <enabled>yes</enabled>
82        <interval>5m</interval>
83        <max_eps>10</max_eps>
84      </synchronization>
85    </syscheck>
```

Codice A.4: Esempio di configurazione delle azioni di monitoraggio

A.5 Esempio di configurazione delle directory di log considerate

```
<!-- Log analysis -->
  <localfile>
    <log_format>command</log_format>
    <command>df -P</command>
    <frequency>360</frequency>
  </localfile>

  <localfile>
    <log_format>journald</log_format>
    <location>journald</location>
  </localfile>

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/ossec/logs/active-responses.log</location>
  </localfile>

  <localfile>
    <log_format>syslog</log_format>
    <location>/var/log/dpkg.log</location>
  </localfile>
```

Codice A.5: Esempio di configurazione delle directory di log considerate

A.6 Esempio di configurazione dei file e directory da ignorare

```
<!-- Files/directories to ignore -->
   <ignore>/etc/mtab</ignore>
   <ignore>/etc/hosts.deny</ignore>
   <ignore>/etc/mail/statistics</ignore>
   <ignore>/etc/random-seed</ignore>
   <ignore>/etc/random.seed</ignore>
   <ignore>/etc/adjtime</ignore>
   <ignore>/etc/httpd/logs</ignore>
   <ignore>/etc/utmpx</ignore>
   <ignore>/etc/wtmpx</ignore>
   <ignore>/etc/cups/certs</ignore>
   <ignore>/etc/dumpdates</ignore>
   <ignore>/etc/svc/volatile</ignore>

  <!-- File types to ignore -->
    <ignore type="sregex">.log$|.swp$</ignore>

  <!-- Check the file, but never compute the diff -->
    <nodiff>/etc/ssl/private.key</nodiff>
    <skip_nfs>yes</skip_nfs>
    <skip_dev>yes</skip_dev>
    <skip_proc>yes</skip_proc>
    <skip_sys>yes</skip_sys>
```

Codice A.6: Esempio di configurazione dei file e directory da ignorare

A.7 Kea-Dhcp4 Log Decoder

```
<!-- Decoder padre per tutti i log Kea DHCP -->
<decoder name="kea-dhcp4">
   <program_name>^kea-dhcp4</program_name>
</decoder>

<!-- Decoder per DHCP4_LEASE_ALLOC -->
<decoder name="kea-dhcp4-lease-alloc">
   <parent>kea-dhcp4</parent>
   <prematch>DHCP4_LEASE_ALLOC</prematch>
   <regex type="pcre2">DHCP4_LEASE_ALLOC \[hwtype=(\d+)\s+([\da-fA-F:]+)\],\s+cid
       =\[([^\]]+)\],\s+tid=([^:]+):\s+lease\s+([\d\.]+)\s+has\s+been\s+allocated\s+
       for\s+(\d+)\s+seconds</regex>
   <order>hwtype,client_mac,client_id,transaction_id,allocated_ip,lease_second</order
       >
</decoder>

<!-- Decoder per DHCP4_INIT_REBOOT -->
<decoder name="kea-dhcp4-init-reboot">
   <parent>kea-dhcp4</parent>
   <prematch>DHCP4_INIT_REBOOT</prematch>
   <regex type="pcre2">DHCP4_INIT_REBOOT \[hwtype=(\d+)\s+([\da-fA-F:]+)\],\s+cid
       =\[([^\]]+)\],\s+tid=([^:]+):\s+client\s+is\s+in\s+INIT-REBOOT\s+state\s+and\s
       +requests\s+address\s+([\d\.]+)</regex>
   <order>hwtype,client_mac,client_id,transaction_id,requested_ip</order>
</decoder>

<!-- Decoder per messaggi generici di valutazione -->
<decoder name="kea-dhcp4-eval">
   <parent>kea-dhcp4</parent>
   <prematch>EVAL_RESULT</prematch>
   <regex type="pcre2">EVAL_RESULT Expression (\S+) evaluated to (\d+)</regex>
   <order>expression,result</order>
</decoder>
```

Codice A.7: Decoder Kea-Dhcp4

A.8 Regole Kea-Dhcp4

```
<!-- Regole per log kea-dhcp4 -->
<group name="dhcp,kea,">
  <!-- Regola base per kea-dhcp4 -->
  <rule id="100500" level="0">
    <decoded_as>kea-dhcp4</decoded_as>
    <description>kea-dhcp4 message detected.</description>
  </rule>

  <!-- Rilevamento valutazione espressione -->
  <rule id="100510" level="2">
    <if_sid>100500</if_sid>
    <field name="expression">\.+</field>
    <description>DHCP4 evaluation: Expression $(expression) evaluated to $(result)</
        description>
  </rule>

  <!-- Rilevamento richiesta INIT-REBOOT -->
  <rule id="100520" level="3">
    <if_sid>100500</if_sid>
    <field name="requested_ip">\.+</field>
    <description>DHCP4 init-reboot: Device MAC $(client_mac) requested IP $(
        requested_ip)</description>
  </rule>

  <!-- Allocazione lease DHCP -->
  <rule id="100530" level="3">
    <if_sid>100500</if_sid>
    <field name="allocated_ip">\.+</field>
    <description>DHCP4 lease allocated: MAC $(client_mac) assigned IP $(allocated_ip)
        for $(lease_second) seconds</description>
  </rule>

  <!-- Alert per richieste multiple in breve tempo (possibile attacco DoS) -->
  <rule id="100540" level="10" frequency="8" timeframe="60">
    <if_matched_sid>100520</if_matched_sid>
    <same_field>client_mac</same_field>
    <description>DHCP4 possible DoS attack: Multiple requests from MAC $(client_mac)
        in short timeframe</description>
    <mitre>
      <id>T1498</id> <!-- Network Denial of Service -->
    </mitre>
  </rule>

  <!-- Alert per richieste da MAC address sospetti -->
  <rule id="100550" level="8">
    <if_sid>100520</if_sid>
    <field name="client_mac">^00:00:00|^ff:ff:ff</field>
    <description>DHCP4 suspicious MAC address $(client_mac) detected</description>
    <mitre>
      <id>T1040</id> <!-- Network Sniffing -->
    </mitre>
  </rule>

  <!-- Alert per tentativi di IP spoofing (richiesta di IP non valido) -->
  <rule id="100560" level="7">
    <if_sid>100520</if_sid>
    <field name="requested_ip">^127\.|^10\.|^172\.16\.|^192\.168\.</field>
    <field name="requested_ip" negate="yes">^192\.168\.100\.</field>
    <description>DHCP4 possible IP spoofing: MAC $(client_mac) requested suspicious IP
        $(requested_ip)</description>
    <mitre>
      <id>T1557</id> <!-- Man-in-the-Middle -->
    </mitre>
  </rule>
</group>
```

Codice A.8: Regole Kea-Dhcp4

A.9 Captive Portal Log Decoder

```
<!-- Decoder principale per logportalauth -->
<decoder name="logportalauth">
   <program_name>^logportalauth</program_name>
</decoder>

<!-- Decoder per riconfigurazioni captive portal -->
<decoder name="logportalauth-reconfig">
   <parent>logportalauth</parent>
   <prematch>Zone: \S+ - Reconfiguring captive portal</prematch>
   <regex type="pcre2">Zone: (\S+) - Reconfiguring captive portal\((\S+)\)</regex>
   <order>zone,portal_name</order>
</decoder>

<!-- Decoder per accettazione connessioni captive portal -->
<decoder name="logportalauth-accept">
   <parent>logportalauth</parent>
   <prematch>Zone: \S+ - ACCEPT:</prematch>
   <regex type="pcre2">Zone: (\S+) - ACCEPT: (\S+), ([\da-fA-F:]+), ([\d\.]+)</regex>
   <order>zone,username,client_mac,client_ip</order>
</decoder>

<!-- Decoder per fallimenti di autenticazione del captive portal -->
<decoder name="logportalauth-failure">
   <parent>logportalauth</parent>
   <prematch>Zone: \S+ - FAILURE:</prematch>
   <regex type="pcre2">Zone: (\S+) - FAILURE: (\S+), ([\da-fA-F:]+), ([\d\.]+), (.+)</
       regex>
   <order>zone,username,client_mac,client_ip,failure_reason</order>
</decoder>

<!-- Decoder per disconnessioni captive portal -->
<decoder name="logportalauth-disconnect">
   <parent>logportalauth</parent>
   <prematch>Zone: \S+ - DISCONNECT:</prematch>
   <regex type="pcre2">Zone: (\S+) - DISCONNECT: (\S+), ([\da-fA-F:]+), ([\d\.]+)</
       regex>
   <order>zone,username,client_mac,client_ip</order>
</decoder>
```

Codice A.9: Decoder Captive Portal

A.10 Regole Captive Portal

```
<!-- Regole per log captive portal -->
<group name="captive_portal,">
  <!-- Regola base per logportalauth -->
  <rule id="100100" level="0">
    <decoded_as>logportalauth</decoded_as>
    <description>Captive portal events</description>
  </rule>

  <!-- Regola per riconfigurazioni -->
  <rule id="100101" level="3">
    <if_sid>100100</if_sid>
    <field name="portal_name">\.+</field>
    <description>Captive portal reconfiguration: $(portal_name) in zone $(zone)</
        description>
  </rule>

  <!-- Regola per accettazione connessioni -->
  <rule id="100102" level="7">
    <if_sid>100100</if_sid>
    <field name="username">\.+</field>
    <field name="client_ip">\.+</field>
    <regex type="pcre2">Zone: \S+ - ACCEPT:</regex>
    <description>Captive portal authentication accepted: User $(username) ($(client_ip
        )) in zone $(zone)</description>
  </rule>

  <!-- Regola per fallimenti di autenticazione -->
  <rule id="100105" level="7">
    <if_sid>100100</if_sid>
    <field name="failure_reason">\.+</field>
    <description>Captive portal authentication failed: User $(username) ($(client_ip))
         in zone $(zone) - $(failure_reason)</description>
  </rule>

  <!-- Regola per disconnessioni -->
  <rule id="100109" level="4">
    <if_sid>100100</if_sid>
    <field name="username">\.+</field>
    <field name="client_ip">\.+</field>
    <regex type="pcre2">Zone: \S+ - DISCONNECT:</regex>
    <description>Captive portal user disconnected: User $(username) ($(client_ip))
        from zone $(zone)</description>
  </rule>
</group>
```

Codice A.10: Regole Captive Portal

Bibliografia

[1] National Institute of Standards e Technology. *Special Publication 800-53: Security and Privacy Controls for Federal Information Systems and Organizations. [DOI: 10.6028/NIST.SP.800-53r5].* URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf.

[2] Roslaily Muhammad, Saiful Adli Ismail e Noor Hafizah Hassan. «Botnet Detection and Incident Response in Security Operation Center (SOC): A Proposed Framework». In: *International Journal of Advanced Computer Science and Applications* (2024). URL: https://api.semanticscholar.org/CorpusID:268858695.

[3] Mohanur Jagadeesan e Pradeep Raj. «A Framework Design to Improve and Evaluate the Performance of Security Operation Center (SOC)». In: 2020. URL: https://api.semanticscholar.org/CorpusID:226189938.

[4] Muhammad Irfan Luthfi, Muharman Lubis e Rd. Rohmat Saedudin. «Development of Security Operation Center (SOC) Governance Blueprint Based on Consideration of Process Maturity Level Parameters». In: *2023 8th International Conference on Information Technology and Digital Applications (ICITDA)* (2023), pp. 1–8. URL: https://api.semanticscholar.org/CorpusID:267702236.

[5] Masomi Majid e K. A. Zainol Ariffi. «Success Factors for Cyber Security Operation Center (SOC) Establishment». In: *Proceedings of the Proceedings of the 1st International Conference on Informatics, Engineering, Science and Technology, INCITEST 2019, 18 July 2019, Bandung, Indonesia* (2019). URL: https://api.semanticscholar.org/CorpusID:207812597.

[6] Afsaneh Madani, Saed Rezayi e Hossein Gharaee. «Log management comprehensive architecture in Security Operation Center (SOC)». In: *2011 International Conference on Computational Aspects of Social Networks (CASoN)* (2011), pp. 284–289. URL: https://api.semanticscholar.org/CorpusID:18327555.

[7] Asif Siddiqui et al. «Survey on Unified Threat Management (UTM) Systems for Home Networks». In: *IEEE Communications Surveys & Tutorials* 26 (2024), pp. 2459–2509. URL: https://api.semanticscholar.org/CorpusID:268823678.

[8] Anish Sridharan e V. Kanchana. «SIEM integration with SOAR». In: *2022 International Conference on Futuristic Technologies (INCOFT)* (2022), pp. 1–6. URL: https://api.semanticscholar.org/CorpusID:258136783.

[9] Jawad Manzoor et al. «Cybersecurity on a budget: Evaluating security and performance of open-source SIEM solutions for SMEs». In: *PLOS ONE* 19 (2024). URL: https://api.semanticscholar.org/CorpusID:268738306.

[10] Alessandro Molinari. «Costruiamo un "SOC domestico" con soli 100 Euro – Prima Parte». In: *Red Hot Cyber* (2024). URL: https://www.redhotcyber.com/post/costruire-un-soc-in-casa-con-100-euro/.

[11] Massimiliano Brolli. «Che cos'è il Red Team? Un viaggio all'interno dell'universo degli esperti di sicurezza offensiva». In: *Red Hot Cyber* (2023). URL: https://www.redhotcyber.com/post/che-cose-il-red-team-un-viaggio-allinterno-delluniverso-degli-esperti-di-sicurezza-offensiva/.

[12] Dell. *Server rack PowerEdge R760*. URL: https://www.dell.com/it-it/shop/ipovw/poweredge-r760.

[13] HP. *QuickSpecs, HP ProLiant DL385 Generation 7 (G7)*. URL: https://www.powerserver.pl/UserFiles/data_sheets/servery/hp/dokumentacja%20Serwer%20HP%20ProLiant%20DL385G7.pdf.

[14] Ebay. *HP PROLIANT DL385 G7 24 CORE SERVER AMD OPTERON 6172 64GB RAM 2X 146GB SAS HDD*. URL: https://www.ebay.it/itm/233144525835.

[15] *What is a vSAN?* Online guide. URL: https://www.purestorage.com/it/knowledge/what-is-vsan.html.

[16] *AlienVault OSSIM*. Sito ufficiale di AlienVault OSSIM. URL: https://www.alienvault.com/products/ossim.

[17] Stephen Cooper. *The Best Next-Gen SIEM*. URL: https://www.comparitech.com/net-admin/best-next-gen-siem/.

[18] *Elastic SIEM*. Sito ufficiale di Elastic SIEM. URL: https://www.elastic.co/solutions/siem.

[19] *Wazuh - Open Source Security Platform*. Sito ufficiale di Wazuh. URL: https://wazuh.com/.

[20] *SIEMonster*. Sito ufficiale di SIEMonster. URL: https://siemonster.com/.

[21] *Security Onion Solutions*. Sito ufficiale di Security Onion. URL: https://securityonion.net/.

[22] *The ELK Stack - Elastic*. Descrizione di ELK Stack sul sito di Elastic. URL: https://www.elastic.co/what-is/elk-stack.

[23] Manage Engine. *Log 360*. URL: https://www.manageengine.com/cloud-siem/features/cloud-based-log-management.html.

[24] New-Scale SIEM Google. *Exabeam Fusion*. URL: https://cloud.google.com/exabeam.

[25] *Splunk*. Sito ufficiale di Splunk. URL: https://www.splunk.com/.

[26] Loadfocus. *10 Best SIEM Tools Of 2025*. URL: https://loadfocus.com/blog/comparisons/siem-tools/.

[27] *Microsoft Sentinel*. Sito ufficiale di Microsoft Sentinel. URL: https://azure.microsoft.com/en-us/products/microsoft-sentinel.

[28] *LogRhythm*. Sito ufficiale di LogRhythm. URL: https://logrhythm.com/.

[29] Peerspot. *Siem Comparison*. URL: https://www.peerspot.com/products/comparisons/.

[30] Proxmox. *Proxmox ISO download*. URL: https://www.proxmox.com/en/downloads.

[31] Proxmox. *Proxmox Installation WIKI*. URL: https://pve.proxmox.com/wiki/Installation#nomodeset_kernel_param.

[32] Raid5. *Raid5 features*. URL: https://www.ontrack.com/it-it/blog/raid5.

[33] McLaren Data Systems. *Remove Proxmox Subscription Notice (Tested to 8.3)*. URL: https://mclarendatasystems.com/remove-proxmox51-subscription-notice/.

[34] Proxmox Server Solutions GmbH. *Proxmox VE Administration Guide*. URL: https://pve.proxmox.com/pve-docs/pve-admin-guide.html.

[35] Amazon. *Che cos'è la KVM (Kernel-Based Virtual Machine)?* URL: https://aws.amazon.com/it/what-is/kvm/.

[36] Maya Sari, Azriel Christian Nurcahyo e Noviyanti. P. «Network Server Management Based on Virtualization Technology using Proxmox at Diskominfo Bengkayang Regency». In: *REKA ELKOMIKA: Jurnal Pengabdian kepada Masyarakat* (2024). URL: https://api.semanticscholar.org/CorpusID:275553104.

[37] Vicente Díaz. *Build a Champion SOC with VirusTotal and Palo Alto Networks Cortex XSOAR*. 2022. URL: https://blog.virustotal.com/2022/02/build-champion-soc-with-virustotal-and.html.

[38] Security Architect. *YARA rules: uno strumento contro il malware*. URL: https://www.securityarchitect.eu/2023/05/08/yara-rules-uno-strumento-contro-il-malware/.

[39] Shuffle. *Shuffle Official Website*. URL: https://shuffler.io/.

[40] Wazuh. *Wazuh Quickstart*. URL: https://documentation.wazuh.com/current/quickstart.html.

[41] Wazuh. *Wazuh Step by step installation*. URL: https://documentation.wazuh.com/current/installation-guide/wazuh-indexer/step-by-step.html.

[42] Wazuh. *Wazuh Agent deploy guide*. URL: https://documentation.wazuh.com/current/installation-guide/wazuh-agent/index.html.

[43] Marco Valle. *Installazione di Wazuh agent su pfSense per il monitoraggio dei log del firewall*. URL: https://www.raffaelechiatto.com/installazione-di-wazuh-agent-su-pfsense-per-il-monitoraggio-dei-log-del-firewall/.

[44] Ossec. *Ossec official website*. URL: https://www.ossec.net/.

[45] Opensearch. *Opensearch official website*. URL: https://opensearch.org/.

[46] Kibana. *Elastic-Kibana official website*. URL: https://www.elastic.co/kibana.

[47] *Send Pfsense logs to Wazuh*. Online guide. URL: https://devopstales.github.io/linux/wazuh-pfsense-syslog/.

[48] *Custom decoders*. Wazuh documentation. URL: https://documentation.wazuh.com/current/user-manual/ruleset/decoders/custom.html.

[49] *Wazuh Regex Syntax*. Wazuh documentation. URL: https://documentation.wazuh.com/current/user-manual/ruleset/ruleset-xml-syntax/regex.html.

[50] *Wazuh Decoder Syntax*. Wazuh documentation. URL: https://documentation.wazuh.com/current/user-manual/ruleset/ruleset-xml-syntax/decoders.html.

[51] *Wazuh Rule Syntax*. Wazuh documentation. URL: https://documentation.wazuh.com/current/user-manual/ruleset/ruleset-xml-syntax/rules.html.

[52] *RegexTestTool*. Online tool. URL: http://regex101.com/.

[53] *Wazuh-logtest tool*. Wazuh documentation. URL: https://documentation.wazuh.com/current/user-manual/reference/tools/wazuh-logtest.html.

[54] *Kea DHCP*. URL: https://www.isc.org/kea/.

[55] *Captive Portal Conf*. URL: https://blog.miniserver.it/pfsense/capitve-portal/.

[56] *Captive Portal*. URL: https://it.wikipedia.org/wiki/Captive_portal.

[57] *Splunk Pricing Calculator*. 2025. URL: https://www.splunk.com/pricing.

[58] Flexera. *State of the Cloud Report*. Rapp. tecn. Flexera, 2025.

[59] Gartner. *Market Guide for SIEM Technologies*. Rapp. tecn. Gartner, 2025.

[60] *Wazuh dashboard*. URL: https://documentation.wazuh.com/current/user-manual/wazuh-dashboard/index.html (visitato il giorno 20/03/2025).

[61] Salvatore Lombardo. *Spoofing: cos'è, tipologie di attacco e soluzioni di difesa*. URL: https://www.cybersecurity360.it/nuove-minacce/spoofing-cose-tipologie-di-attacco-e-soluzioni-di-difesa/.

[62] Alberto Stefani. *Attacchi DoS e DDoS: modalità di difesa e contromisure*. URL: https://www.cybersecurity360.it/nuove-minacce/attacchi-dos-e-ddos-modalita-di-difesa-e-contromisure/.

Ringraziamenti

Ringrazio l'Università degli Studi di Camerino per avermi consentito di affrontare questo percorso serenamente, in un ambiente accogliente e coinvolgente.

Ringrazio tutto il corpo docente della Scuola di Scienze e Tecnologie Informatiche, per avermi fatto appassionare a questo settore e avermi fatto scoprire delle capacità che non sapevo di avere.

Un ringraziamento speciale va al Prof. Fausto Marcantoni che, con la sua saggezza, gentilezza e pazienza, mi ha accompagnato nella stesura di questa tesi e mi ha fatto rendere conto, ancora di più, di quanto sia intrigante il mondo della Cyber Security.

Sono estremamente grato di aver potuto fare questa esperienza meravigliosa, piena di nuove opportunità e persone fantastiche.