Università degli Studi di Camerino

Scuola di Scienze e Tecnologie

Corso di Laurea in Informatica (Classe L-31)

Wazuh

The Open Source Security Platform

Laureando

Mattia Incoronato

Matricola 100753

Relatore

Fausto Marcantoni

A.A. 2019/2020

Sommario

Il tema della sicurezza informatica sta diventando negli ultimi anni sempre più importante e fondamentale per tutte le aziende. Non solo per quelle del settore dell'informatica ma anche per tutte le altre che operano nei settori più disparati, vista la quantità di attacchi informatici in costante aumento. L'importanza di tenere al sicuro i computer dei dipendenti e i dati al loro interno è sotto gli occhi di tutti, visto che con degli attacchi mirati i pirati informatici sono in grado di bloccare interi processi produttivi. Proprio per questo motivo sono nate diverse piattaforme che hanno lo scopo di monitorare e controllare diversi aspetti di una macchina client. Queste piattaforme vengono denominate *host intrusion detection system* e l'obiettivo di questa tesi è di analizzare e testare le funzionalità di Wazuh, uno dei sistemi open-source più conosciuti della categoria.

Indice

1 Introduzione **9**

1.1 Obiettivi della tesi 9

1.2 Definizione di Intrusion Detection System 10

1.2.1 HIDS 10

1.2.2 NIDS 11

2 Presentazione di Wazuh **13**

2.1 Altre piattaforme HIDS 14

2.1.1 OSSEC 14

2.1.2 Tripwire 15

2.1.3 Samhain 15

2.1.4 SolarWinds Security Event Manager 15

2.2 Perché Wazuh 16

3 Componenti della piattaforma **17**

3.1 OSECC HIDS 17

3.2 OpenScap 17

3.3 Elastic Stack 18

4 Struttura di Wazuh **19**

4.1 Wazuh Agent 19

4.2 Wazuh Server 21

4.3 Interfaccia Grafica 22

5 Architettura della piattaforma **23**

5.1 Data Flow delle comunicazioni 23
5.1.1 La comunicazione fra Agent e Server 23
5.1.2 La comunicazione tra Server e Elastic Stack 23

6 Configurazione dell'ambiente di test 27
6.1 Installazione del Wazuh Server 27
6.1.1 Installazione modulo Server 28
6.1.2 Installazione Wazuh API 29
6.1.3 Installazione Filebeat 29
6.1.4 Installazione Elastic Stack 30
6.2 Rilascio degli Agent sulle macchine client 31
6.2.1 Rilascio della chiave univoca 31
6.3 Rendere sicuro l'accesso alle API di Wazuh 33
6.3.1 Abilitazione protocollo HTTPS 33
6.3.2 Configurazione credenziali sicure 34
6.3.3 Modifica della porta di default 34

7 La dashboard di Wazuh 37

8 Analisi delle funzionalità 39
8.1 Intrusion Detection - Attacco Brute Force in RDP 39
8.2 Vulnerabilty Detection - Ricerca delle vulnerabilità in ambiente Windows 42
8.3 File Integrity Monitoring - Monitorare directory e file 46
8.3.1 Integrazione di VirusTotal 48
8.4 System Inventory - Raccolta delle informazioni sugli Agent 50
8.5 Active Response 51
8.6 Security Configuration Assessment 53

9 Ulteriori funzionalità della piattaforma 55
9.1 Log Collection - Raccolta dei log di Windows 55
9.2 Command Monitoring 56

10 Management dei backup 57

10.1 Repository basata su File System 57

10.2 Salvataggio automatico degli snapshot 59

11 Rilascio centralizzato del file di configurazione per gli Agent 61

11.1 Creazione dei gruppi e aggiunta degli Agent 61

11.2 Modifica del file di configurazione 62

12 Conclusioni 65

12.1 Sviluppi futuri 66

1. Introduzione

Il crescente numero di attacchi informatici alle infrastrutture delle aziende, ha fatto nascere l'esigenza di avere a disposizione strumenti che possano aiutare a migliorare la sicurezza informatica degli ambienti di lavoro. Monitorare le macchine collegate ad una rete interna può essere di fondamentale importanza per tenere sotto controllo sia server che PC dedicati agli end-user.
Molto spesso nei software e/o nei sistemi operativi installati nelle macchine si possono nascondere enormi vulnerabilità o falle nella sicurezza che permettono ad un eventuale attacco di insediarsi. Inoltre molti malware pericolosi tendono a sostituirsi a file di sistema, rendendo in questo modo complicato scoprire dove si stiano nascondendo.

Negli ambienti di lavoro dove sono presenti decine o addirittura centinaia di macchine collegate alla rete, è impensabile effettuare controlli manualmente e periodicamente su tutti i computer. Per questo motivo vengono utilizzate delle piattaforme dedicate alla sicurezza denominate **Host Intrusion Detection System**.

Queste piattaforme permettono, grazie ad un sistema basato sulla comunicazione client-server, di monitorare le macchine collegate alla rete, rilevando vulnerabilità, tentativi di intrusione dall'esterno ed altro ancora.

1.1 Obiettivi della tesi

La tesi redatta ha lo scopo di analizzare e testare le funzionalità di Wazuh [Waz], un sistema di host intrusion detection system open-source disponibile gratuitamente.
Per studiare ed analizzare il sistema è stato preparato un ambiente di test plausibile e sono stati poi testati i vari moduli proposti dalla piattaforma, in modo tale da poter effettuare considerazioni sulle potenzialità e sui limiti di Wazuh.

1.2 Definizione di Intrusion Detection System

Nel mondo della sicurezza informatica un Intrusion Detection System rappresenta una piattaforma dedicata all'identificazione di accessi non autorizzati, attacchi informatici di varia natura e anomalie. Si tratta in sostanza di strumenti capaci di monitorare in tempo reale e con costanza i sistemi informatici.

Esistono diverse categorie di piattaforme IDS ma le principali sono i Network intrusion detection system **(NIDS)** e gli Host-based intrusion detection system **(HIDS)** (Figura 1.1).

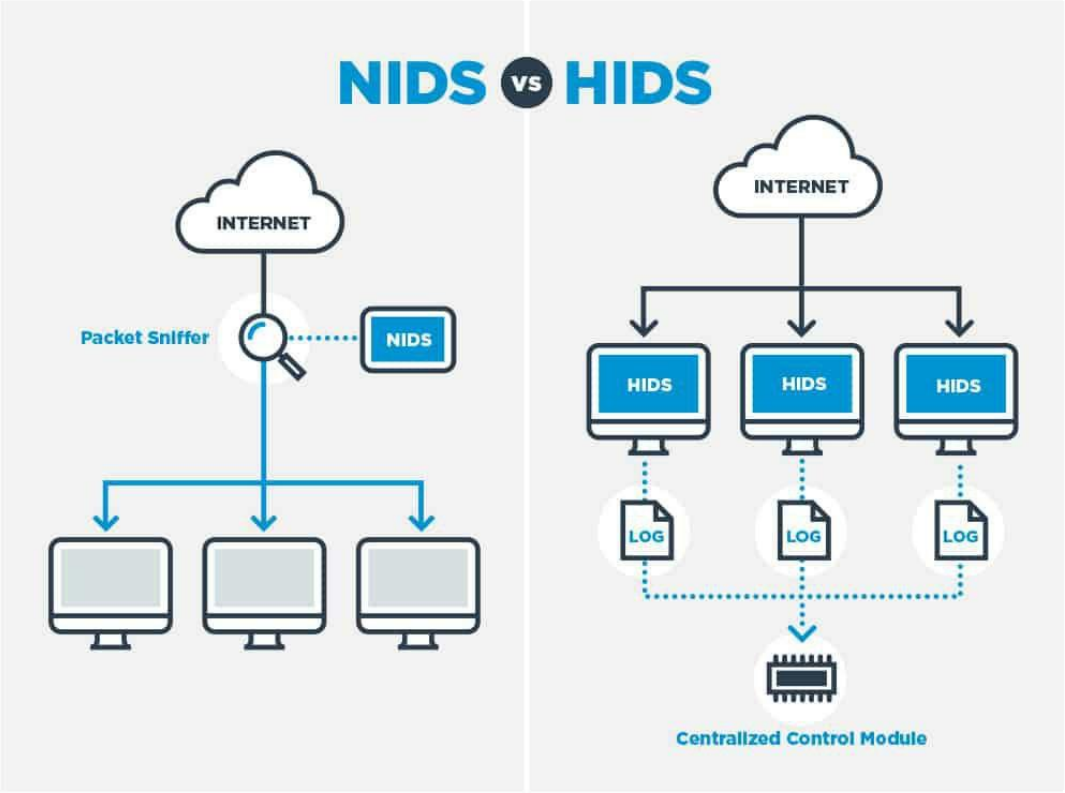

Figura 1.1: NIDS e HIDS a confronto

Fonte: https://www.comparitech.com/net-admin/network-intrusion-detection-tools/

1.2.1 HIDS

Un Host based intrusion detection system è una tipologia di intrusion detection system specializzato nell'analisi e nel monitoraggio dei computer. Queste piattaforme sono basate su una classica architettura client-server, dove il client comunica costantemente

informazioni sullo stato della macchina al server.

I client monitorati hanno il compito di inviare tutta una serie di dati al server, che potrà poi analizzarli, catalogarli e ordinarli. Grazie a questi dati è possibile ricostruire quello che accade sulla macchina monitorata.

I dati raccolti e le funzioni disponibili variano in base alla piattaforma in utilizzo. Grazie ai dati raccolti sui client si riescono a rilevare vulnerabilità, raccogliere log di sistema, controllare modifiche a file ed altro ancora. Wazuh rientra nella categoria dei sistemi HIDS.

1.2.2 NIDS

I NIDS, Network Intrusion Detection System, sono degli strumenti informatici, software o hardware, che hanno il compito di analizzare il traffico di uno o più segmenti di una LAN al fine di individuare anomalie nei flussi o probabili intrusioni informatiche.

I sistemi NIDS non interferiscono sul traffico di rete e si limitano a rimanere in ascolto pronti ad avviare notifiche di allerta nel caso di traffico sospetto. Queste piattaforme possono appoggiarsi ad una struttura hardware o software e solitamente utilizzano due interfacce di rete, una per l'ascolto passivo e l'altra per riportare gli eventi raccolti.

I due sistemi hanno gli stessi obiettivi ma operano in maniera totalmente differente. Mentre gli HIDS hanno una struttura che permette di andare ad analizzare in profondità tutto quello che succede sulla macchina client, i NIDS operano analizzando il traffico dei pacchetti sulla rete.

2. Presentazione di Wazuh

Come detto nell'introduzione, Wazuh è una piattaforma open-source che rientra nella categoria degli host intrusion detection system.
Il sistema si presenta come una soluzione flessibile e personalizzabile grazie alla sua natura open-source ed aperta alle integrazioni.

Inoltre grazie alla sua ampia documentazione ufficiale [Doc] è facilmente accessibile alla maggior parte degli utenti.

Le funzionalità principali delle piattaforma sono le seguenti:

Security Analytics
Wazuh è in grado di raccogliere, indicizzare ed ordinare dati relativi alla sicurezza che permettono la rilevazione di minacce e anomalie.

Intrusion Detection
Grazie al modulo dedicato all'Intrusion Detection, Wazuh è in grado di rilevare tentativi di intrusione, rootkit e malware. Inoltre è in grado di rilevare file nascosti o inconsistenze nelle risposte alle chiamate di sistema.

Log Data Analysis
Wazuh raccoglie e ordina i file di log dei sistemi monitorati, inviandoli ad un sistema centralizzato. Possono essere raccolti sia log di sistema che log delle applicazioni.

File Integrity Monitoring
La funzione di File Integrity Monitoring permette di monitorare costantemente delle directory definite (come ad esempio le cartelle dedicate ai file di sistema) in modo tale da segnalare qualsiasi cambiamento avvenuto.

Vulnerability Detection

Grazie al modulo dedicato al rilevamento delle vulnerabilità, Wazuh è in grado di scovare falle di sicurezza nei software e/o nei sistemi operativi installati sulle macchine monitorate.

Configuration Assessment

La Configuration Assessment permette di accertare che il sistema sia conforme a standard e criteri di sicurezza del settore.

Incident Response

Wazuh è in grado di effettuare Incident Response, ovvero rispondere in modo pro-attivo ed in tempo reale nel caso in cui vengano soddisfatti determinati criteri.

Regulatory Compliance

Questa funzionalità fornisce la possibilità di effettuare controlli sul sistema per assicurarsi che sia conforme a standard legislativi internazionali (Come ad esempio il GDPR).

2.1 Altre piattaforme HIDS

Oltre a Wazuh esistono altre piattaforme *Host Intrusion Detection System* utilizzate per effettuare il monitoraggio di macchine client, ed ognuno di questi sistemi presenta delle proprie particolarità.

2.1.1 OSSEC

OSSEC [Oss] è il progetto originario dal quale è nato Wazuh. Ancora oggi è un sistema molto diffuso ed utilizzato e condivide la gran parte delle funzioni e delle caratteristiche con Wazuh.

Anche OSSEC è un progetto open-source ma a differenza di Wazuh al momento non è molto aggiornato, non è presente una roadmap con degli update definiti e la maggior parte delle ultime versioni rilasciate ha implementato solo qualche correzione di bug.

Proprio questa mancanza di aggiornamenti ha portato gli sviluppatori di Wazuh a creare una nuova piattaforma che partisse dalla base di OSSEC. Grazie al progetto Wazuh sono state fatte molte migliorie e aggiunte alle funzionalità come ad esempio

una migliore scalabilità, miglioramenti nel modulo di Intrusion Detection, integrazione con i provider in cloud come Amazon AWS, Microsoft Azure ed altro ancora.

2.1.2 Tripwire

Tripwire [Tri] è un host-based detection system open-source sviluppato dall'azienda omonima. Oltre ad una versione gratuita, l'azienda offre anche delle soluzioni commerciali personalizzate per le aziende.

Il vantaggio principale di Tripwire è quello di essere un sistema molto leggero, ma il suo limite è quello di poter essere utilizzato esclusivamente su macchine con sistema operativo Linux.

Risulta una soluzione più che adeguata per piccoli ambienti Linux ma manca la possibilità di generare avvisi in tempo reale, di conseguenza è necessario analizzare manualmente tutti i log generati e raccolti.

2.1.3 Samhain

Samhain [Sam] è un'ulteriore alternativa open-source che offre funzionalità di File Integrity Monitoring e analisi dei file di log. La piattaforma è stata strutturata in modo tale da poter monitorare host con sistemi operativi differenti (Unix, Linux, Cygwin/-Windows) ed anche in questo caso si tratta un sistema che può essere gestito con un monitoraggio centralizzato.

Il principale vantaggio rispetto ad altre piattaforme HIDS è la capacità del sistema di essere estremamente impercettibile, nascondendo molto bene la sua presenza ad eventuali tentativi di intrusione.

La sua installazione però è meno intuitiva e più complicata da eseguire ed i report generati non sono molto chiari da leggere in un primo momento. Anche se, sul sito web ufficiale è presente una grande quantità di documentazione che risulta di grande aiuto per gli utenti.

2.1.4 SolarWinds Security Event Manager

SolarWinds [Sol] propone un prodotto completo, intuitivo e ricco di funzionalità. Difatti è presente la possibilità effettuare una comoda archiviazione di log, effettuare File

Integrity Monitoring, creare notifiche di allerta in automatico ed altro ancora.

Inoltre la piattaforma è in grado di effettuare l'analisi del traffico di rete per rilevare eventuali anomalie e può raccogliere dati da quasi tutti i sistemi operativi disponibili.

La soluzione proposta da SolarWinds è però a pagamento, ed i prezzi non sono alla portata di chi cerca una soluzione per piccole infrastrutture.

2.2 Perché Wazuh

Fra tutti i sistemi HIDS disponibili, è stato scelto di studiare Wazuh peché propone una fra le soluzioni più complete sul mercato. Wazuh è infatti supportato dalla maggior parte dei sistemi operativi, è in continuo sviluppo ed ha alle spalle una community di utenti molto attiva.

Inoltre la sua capacità di integrazione con sistemi esterni gli permette di adeguarsi alle più svariate situazioni. Infine la sua natura gratuita e open-source si presta bene ad uno studio approfondito.

3. Componenti della piattaforma

Wazuh nasce come progetto branca di **OSECC HIDS** (un'ulteriore piattaforma per l'intrusion detection), ma è stato poi integrato nel tempo con i moduli open source **Elastic Stack** e **OpenSCAP**, divenendo così un sistema più completo e affidabile.

3.1 OSECC HIDS

OSECC è un sistema di Intrusion Detection utilizzato per monitorare le macchine host. È uno dei sistemi dedicati alla sicurezza informatica più diffusi in assoluto ed il progetto è in continua crescita grazie anche alla sua scalabilità e compatibilità con la maggior parte dei sistemi operativi. OSECC è basato su Agent multipiattaforma in grado di inoltrare tutti i dati utili ad un modulo centrale di controllo, il quale avrà il compito di processare ed analizzare tutte le informazioni rilevate. OSECC HIDS mette a disposizione tra le varie funzionalità la possibilità di effettuare Malware Detection, Active Response in tempo reale, monitoraggio dell'integrità dei file e System Inventory. Grazie al System Inventory è possibile infatti collezionare dati riguardanti informazioni su software installato, hardware, servizi di rete ed altro ancora delle macchine host monitorate.

3.2 OpenScap

OpenScap è uno strumento basato su standard riconosciuti realizzato con l'obiettivo di verificare conformità e sicurezza di sistemi informatici in ambiente aziendale. Viene utilizzato prettamente per attività di Security Compliance e Vulnerability Assessment.

3.3 Elastic Stack

Elastic Stack è una suite di software (Filebeat, Elasticsearch, Kibana) utilizzata per raccogliere, indicizzare, archiviare e presentare i dati di log. Grazie ad un intuitiva interfaccia grafica Elastic Stack permette infatti all'end-user di poter analizzare in maniera intuitiva le informazioni raccolte dal sistema. L'interfaccia permette di visualizzare i dati anche in maniera grafica, così da essere immediatamente visibili e riconoscibili anche a utenti e personale non tecnico.

4. Struttura di Wazuh

Il progetto Wazuh ha lavorato su tre moduli principali per creare una piattaforma unica in cui le varie funzioni e capacità sono raccolte al suo interno:

- Wazuh Agent
- Wazuh Server
- Interfaccia grafica

4.1 Wazuh Agent

L'Agent è un piccolo applicativo che viene installato sulle macchine da monitorare, il quale va a comunicare tutte le rilevazioni effettuate al Wazuh Server.

L'Agent può essere installato su quasi tutti i tipi di sistemi operativi (Windows, Linux, Solaris, Mac ecc.) ed una volta installato inizierà a raccogliere tutti i dati utili da inoltrare poi al server tramite un canale criptato.

Per stabilire la connessione a questo canale sicuro per la comunicazione tra Agent e server viene inizializzato un processo di registrazione che utilizza delle chiavi pre-condivise.

L'Agent utilizza diversi task e processi per monitorare la macchina host in diversi modi. I vari task possono essere infatti suddivisi fra lettura di log di sistema, monitoraggio dell'integrità dei file, scansione della configurazione di sistema ecc.

La documentazione ufficiale di Wazuh mette a disposizione anche dei comodi diagrammi (Fig. 4.1) che favoriscono la comprensione della struttura delle varie componenti.

Ogni processo dell'Agent ha un obiettivo ben definito e dei compiti specifici.

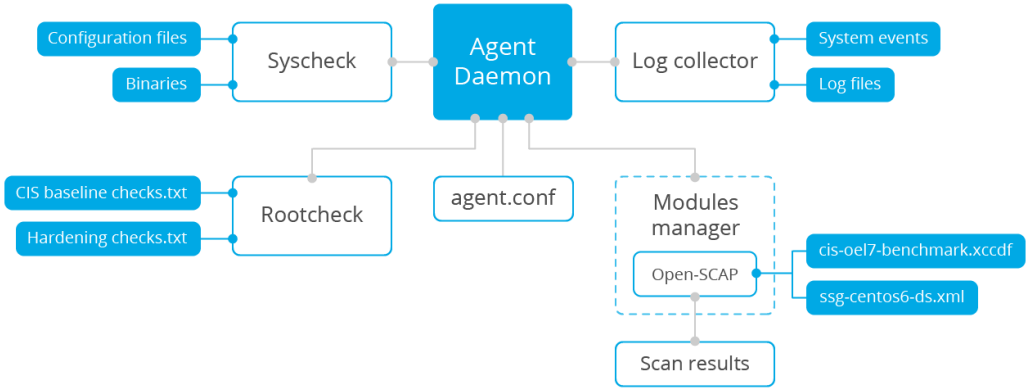

Figura 4.1: Struttura dell'Agent

Fonte: https://documentation.wazuh.com/3.12/getting-started/components.html

Rootcheck: Grazie a questo processo l'Agent è in grado di rilevare rootkits, malware e altre anomalie di sistema. Inoltre è in grado di fare dei basilari check di sicurezza sui file di configurazione di sistema.

Log Collector: Questo task si occupa di leggere i messaggi di log di applicazioni e del sistema operativo. Può leggere anche gli standard log del registro eventi di Windows.

Syscheck: Questo processo si occupa di monitorare l'integrità dei file ed è inoltre in grado di monitorare l'integrità delle chiavi di registro dei sistemi Windows. Grazie al Syscheck è possibile rilevare cambiamenti ai contenuti di un file, ai suoi permessi o altri attributi. Anche la creazione o l'eliminazione di un file può essere rilevata da questo task.

OpenScap: Il modulo di OpenScap utilizza profili di sicurezza basati su OVAL (Open Vulnerability Assessment Language) e XCCDF (Extensible Configuration Checklist Description Format).

Agent Daemon: è il processo che riceve tutti i dati generati e raccolti da tutti gli altri componenti dell'Agent. Il suo scopo è quello di comprimere, criptare ed inviare i dati al server tramite il canale sicuro autenticato. Questo processo ha un accesso limitato al sistema host monitorato, in modo tale che la sicurezza generale dell'Agent sia migliorata (considerando che è l'unico componente con accesso diretto alla rete).

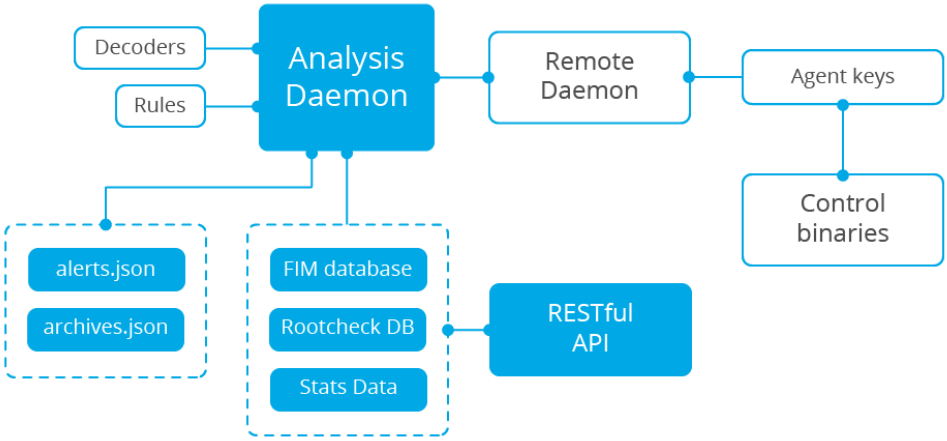

Figura 4.2: Struttura del Wazuh Server

Fonte: https://documentation.wazuh.com/3.12/getting-started/components.html

4.2 Wazuh Server

Il Wazuh Server (Fig. 4.2) è solitamente installato su una macchina fisica, su una virtual machine o su una macchina in cloud. Il server ha il compito di ricevere e raccogliere tutti i dati inviati dalle varie macchine host che ospitano l'Agent. Anche il server è composto da una serie di moduli con compiti specifici.

Registration Service: Questo processo ha il compito di gestire la registrazione di nuovi Agent, infatti si occupa di fornire e distribuire le chiavi necessarie all'autenticazione del canale sicuro di comunicazione. Questo processo supporta l'autenticazione tramite TLS/SSL.

Remote daemon service: è il servizio che riceve i dati dai vari Agent. Utilizza le chiavi per validare la connessione ed identificare l'Agent dal quale provengono i dati ricevuti.

Analysis Daemon: Si tratta del processo che effettua l'analisi dei dati ricevuti. Sfrutta i decoders per identificare il tipo di informazioni che devono essere trattate (Eventi di Windows, log di web server, ecc.), dopodiché estrapola le informazioni utili dai log (indirizzi IP, id dell'evento, account utente ecc.). Grazie a delle regole che è possibile impostare è poi in grado di attivare in automatico degli avvisi o delle contromisure (in base ad eventuali allarmi scattati dopo la lettura).

RESTful API: Fornisce l'interfaccia tramite la quale è possibile gestire la configurazione e il rilascio degli Agent. Questo modulo è sfruttato anche dall'interfaccia web di Wazuh.

4.3 Interfaccia Grafica

Elastic Stack rappresenta un insieme di progetti open source dedicati al log management, tra i quali troviamo Elasticsearch, Kibana, Filebeat ed altri, ma solo alcuni di questi sono rilevanti per la piattaforma Wazuh.

Grazie alle potenzialità di questi tool, Wazuh mette a disposizione un'interfaccia web basata su Kibana dove poter gestire l'infrastruttura di sistema e controllare in tempo reale tutti i dispositivi in cui è stato rilasciato l'Agent.

5. Architettura della piattaforma

Wazuh risulta essere uno strumento legato a questo specifico modello di architettura in cui gli host monitorati comunicano con il server grazie al lavoro degli Agent.

In ambienti di grandi dimensioni vengono sfruttati dei cluster dove più nodi del modulo di Elastic Stack comunicano fra di loro per catalogare e indicizzare le informazioni raccolte. Mentre in ambienti con un numero minore di 50 host monitorati è possibile utilizzare un singolo nodo Elastic Stack.

Come è possibile vedere dalla Figura 5.1, in architetture medio/grandi i nodi di Elastic Stack sono installati su macchine separate rispetto al Wazuh Server, mentre in contesti di piccole dimensioni (Fig. 5.2) è possibile installare entrambe le istanze sulla stessa macchina.

5.1 Data Flow delle comunicazioni

5.1.1 La comunicazione fra Agent e Server

L'Agent di Wazuh sfrutta il protocollo dei messaggi di OSSEC per inviare i dati raccolti al Wazuh Server sulla porta 1514. Il Wazuh Server riceve i dati, effettua la decodifica e controlla gli eventi grazie all'engine per l'analisi.

Tutti gli eventi che faranno scattare degli allarmi vengono segnalati e catalogati in maniera differente, così da poterli riconoscere in maniera semplice.

5.1.2 La comunicazione tra Server e Elastic Stack

Il Wazuh Server utilizza Filebeat per inviare i dati riguardanti alert ed eventi ad Elastic Search tramite crittografia TLS. Filebeat formatta i dati e li arricchisce con alcune

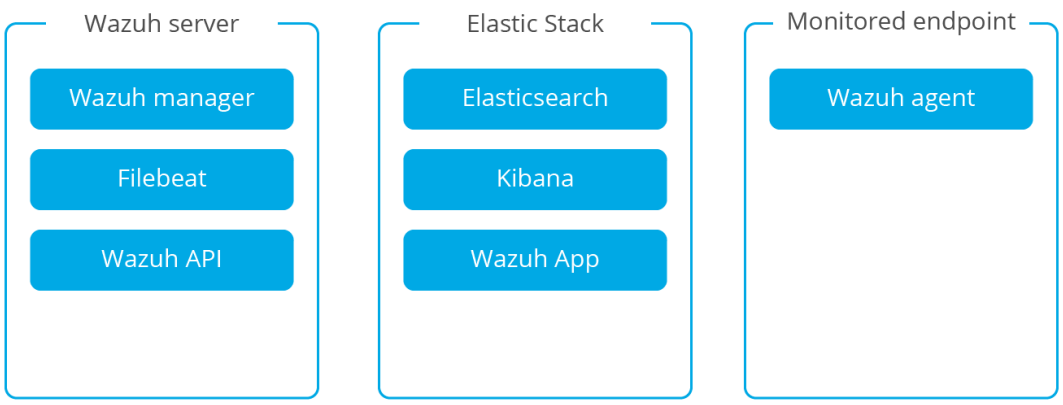

Figura 5.1: Architettura classica di Wazuh

Fonte: https://documentation.wazuh.com/3.12/getting-started/architecture.html

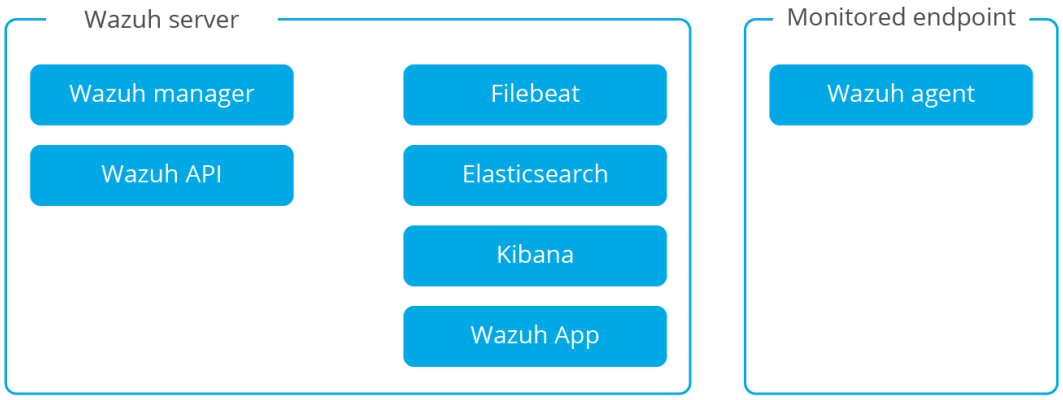

Figura 5.2: Architettura per infrastutture di piccole dimensioni

Fonte: https://documentation.wazuh.com/3.12/getting-started/architecture.html

informazioni prima di spedire i pacchetti verso Elastic Search.

Una volta che Elastic Search si è occupato di indicizzare tutti i dati ricevuti, potranno essere visualizzati attraverso l'interfaccia di Kibana.

L'app di Wazuh all'interno di Kibana effettua poi costantemente richieste di query alla RESTful API in modo tale da poter tenere sempre aggiornato in tempo reale lo status degli host monitorati.

6. Configurazione dell'ambiente di test

Per poter lavorare con Wazuh, in modo tale da analizzare le sue potenzialità, è stato preparato un ambiente di test con l'obiettivo di ricreare uno scenario realmente plausibile.

Per l'ambiente di test sono state preparate e configurate 3 macchine, di cui una facente da server e le altre due da macchine client (Fig. 6.1). Per il PC-Server è stato scelto di utilizzare come sistema operativo Ubuntu 20.04, vista la sua praticità e immediatezza nell'utilizzo.

Sulle due macchine client è stato invece installato Windows 10 (1909), visto che è il sistema operativo più sfruttato ed utilizzato dagli utenti impiegati negli uffici e in generale sul luogo di lavoro (Oltre 600 milioni di dispositivi utilizzano il sistema operativo di Microsoft).

Inoltre è risaputo che Windows non è nuovo a falle nella sicurezza e problemi di vulnerabilità di diversa entità.

6.1 Installazione del Wazuh Server

Considerato l'ambiente di test molto piccolo, si è optato per la scelta di un'architettura in cui i moduli del Wazuh Server e di Elastic Stack fossero installati sulla stessa macchina.

Come anticipato poco fa il sistema operativo scelto per la macchina PC-SERVER è Ubuntu (Wazuh Server non può essere installato su macchine Windows).
Il Wazuh Server necessita inoltre dell'installazione delle **Wazuh API**, di **Filebeat** e

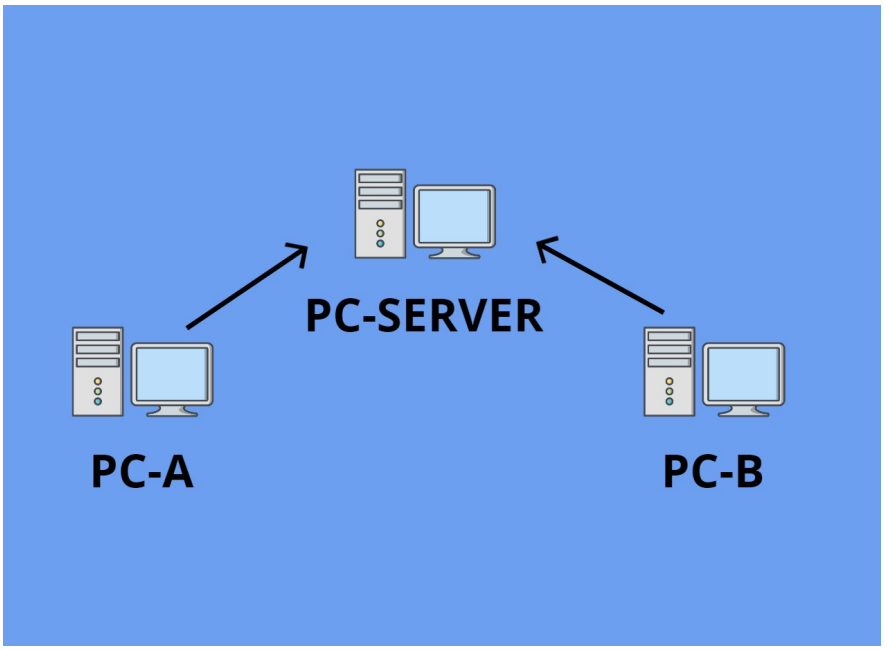

Figura 6.1: Ambiente di test

di **Elastic Stack**.

6.1.1 Installazione modulo Server

L'installazione del modulo server è molto semplice e può essere effettuate interamente da terminale.

Come primo passaggio è necessario importare la chiave GPG di Wazuh e poi aggiungere le repository associate. Dopodiché è possibile passare all'installazione vera e propria.

```
url -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | apt-key add

deb https://packages.wazuh.com/3.x/apt/ stable main | tee -a
/etc/apt/sources.list.d/wazuh.list

apt-get install wazuh-manager
```

Una volta che l'installazione è stata effettuata, è possibile verificare lo stato del servizio tramite il comando:

```
systemctl status wazuh-manager
```

6.1.2 Installazione Wazuh API

Per utilizzare la Wazuh API è necessario installare il framework Nodejs (in versione superiore o almeno uguale alla 4.6.1).

Una volta aggiunta la repository interessata è possibile effettuare anche qui l'installazione da terminale.

```
curl -sL https://deb.nodesource.com/setup_10.x | bash -

apt-get install nodejs
```

Dopo aver installato il framework è possibile passare all'installazione dell'API di Wazuh.

```
apt-get install wazuh-api
```

6.1.3 Installazione Filebeat

Per l'installazione del modulo Filebeat procediamo in maniera analoga all'installazione del Wazuh Server.

Come primo passaggio quindi importiamo la chiave GPG, poi la repository ed infine procediamo con l'installazione di Filebeat.

```
curl -s https://artifacts.elastic.co/GPG-KEY-elasticsearch |
apt-key add -

echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main"
| tee /etc/apt/sources.list.d/elastic-7.x.list

apt-get install filebeat=7.8.1
```

Dopo l'installazione è necessario scaricare il file di configurazione di Filebeat per Wazuh, il template per la visualizzazione degli alert ed il modulo per l'interazione con il server.

Come ulteriore passaggio è essenziale modificare il file **filebeat.yml** per inserire l'indirizzo IP della macchina Wazuh Server.

Infine può essere fatto partire il servizio di Filebeat

```
systemctl daemon-reload
systemctl enable filebeat.service
systemctl start filebeat.service
```

6.1.4 Installazione Elastic Stack

Il modulo di Elastic Stack si suddivide nell'installazione di Elastic Search e Kibana. In entrambi i casi le procedure sono molto simili a quelle delle precedenti installazioni. Anche qui è fondamentale aggiungere la chiave GPG e le repository necessarie.

Per installare Elastic Search basterà poi lanciare il comando:

```
apt-get install elasticsearch=7.8.1
```

Di default Elastic Search sarà in ascolto solo sull'interfaccia di loopback locale. Se si vuole lasciare Elastic Search in ascolto da un indirizzo esterno è quindi opportuno modificare il file **elasticsearch.yml** ed inserire l'indirizzo IP interessato alla voce *network.host*

Dopo l'installazione è necessario far partire il servizio di Elastic Search e caricare il template di Filebeat.

```
systemctl daemon-reload
systemctl enable elasticsearch.service
systemctl start elasticsearch.service
filebeat setup --index-management -E setup.template.json.enabled=false
```

Il prossimo passo da effettuare è l'installazione di Kibana

```
apt-get install kibana=7.8.1
```

Come per Elastic Search, anche Kibana è di deafult in ascolto sull'indirizzo di loopback, per poter adattare la configurazione basterà modificare il file *kibana.yml*.

Una volta fatto partire il servizio di Kibana è possibile passare al rilascio degli Agent sulle macchine client.

```
systemctl daemon-reload
systemctl enable kibana.service
systemctl start kibana.service
```

6.2 Rilascio degli Agent sulle macchine client

Ci sono diversi modi per effettuare il rilascio degli Agent sulle macchine client e in ambiente Windows il sistema più immediato è sicuramente quello del download del pacchetto .msi dal sito di Wazuh.

In alternativa è possibile anche effettuare il download da terminale con il seguente comando:

```
wazuh-Agent-3.13.1-1.msi /q
```

Una volta scaricato il pacchetto è possibile proseguire con l'installazione guidata.

Completata l'installazione, bisognerà configurare l'Agent tramite GUI (Fig. 6.2).

Sarà infatti necessario compilare i due campi con l'indirizzo IP della macchina che ospita il Wazuh Server e la chiave di autenticazione univoca che permetterà la comunicazione sicura con il server.

In un'infrastruttura di dominio basata su Windows Server è inoltre possibile installare l'Agent sui client tramite una GPO personalizzata.

6.2.1 Rilascio della chiave univoca

Per il rilascio della chiave da aggiungere all'Agent si può procedere con diverse modalità fra cui:

- Comando dal terminale
- Sfruttando la Wazuh API
- Servizio di registrazione tramite autorizzazione con password
- Servizio di registrazione tramite verifica della macchina client

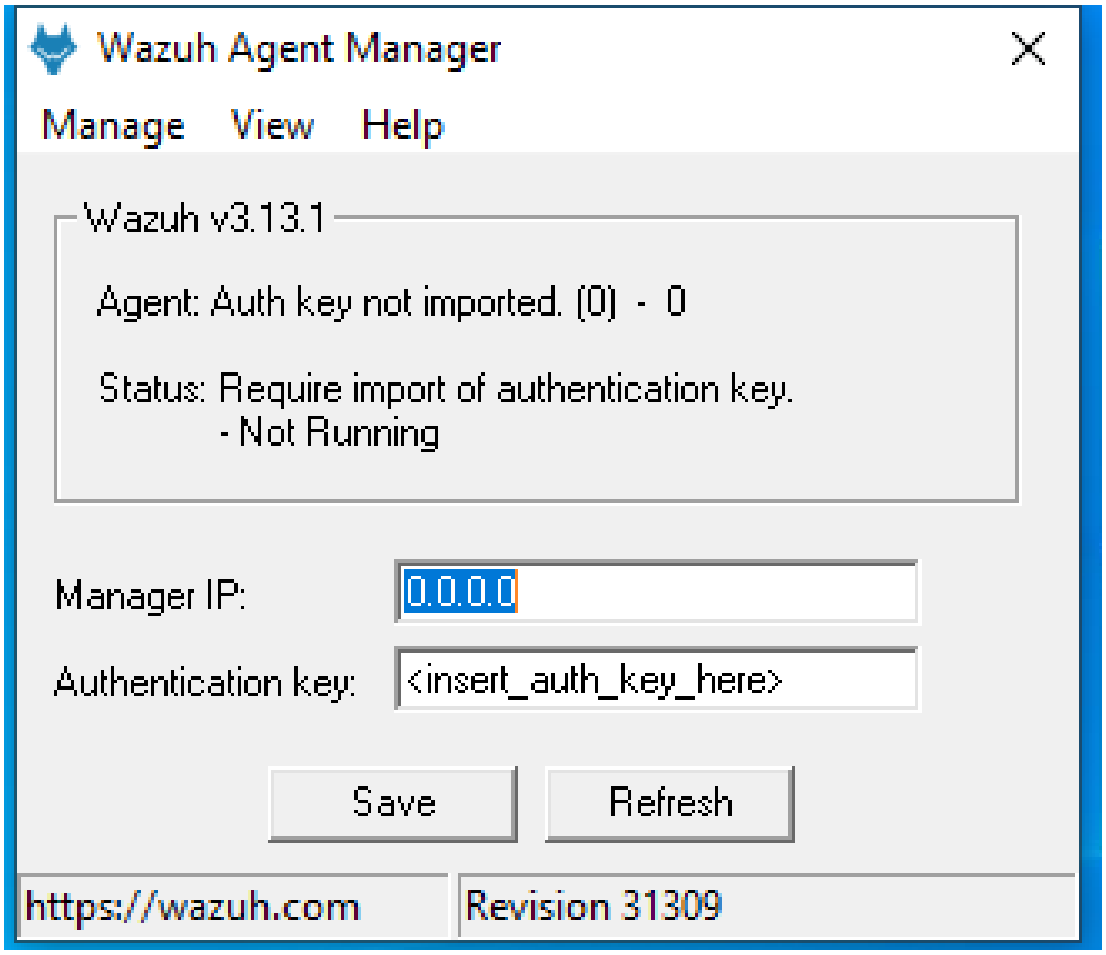

Figura 6.2: GUI per configurazione dell'Agent

Per semplicità di utilizzo è stato preferita la possibilità di utilizzare il terminale anche in questo caso.

Dalla CLI della macchina server basterà utilizzare il seguente comando per aggiungere un nuovo client:

```
/var/ossec/bin/manage_Agents -a <IPClient> -n <NomeClient>
```

A questo punto basterà sostituire ad *IPClient* l'indirizzo IP della macchina da monitorare e a *NomeClient* il nome da assegnare all'Agent (Nel caso dell'ambiente di test abbiamo quindi PC-A e PC-B).

Lanciando questo comando Wazuh assegnerà all'Agent un ID, ad esempio 001. Con il successivo comando viene richiesto il rilascio della chiave per la macchina scelta:

```
/var/ossec/bin/manage_Agents -e <IDAgent>
```

L'output generato sarà una stringa di questo tipo:

```
Agent key information for '001' is:
MDAxIDE4NWVlNjE1Y2YzYi3ODhkNGQzMjM5ZTdlNGVmNzQzMGFjMDA4Nw==
```

Ora non resta altro che copiare la chiave, inserirla nella GUI di configurazione dell'Agent ed avviare la comunicazione con il Wazuh Server.

6.3 Rendere sicuro l'accesso alle API di Wazuh

Una volta effettuata l'installazione del Wazuh Manager, è importante rendere sicuro l'accesso alle API in modo tale da garantire l'accesso alla piattaforma solo agli utenti autorizzati.

Di default infatti, la comunicazione fra Kibana e le Wazuh API non è criptata.

6.3.1 Abilitazione protocollo HTTPS

Per abilitare l'HTTPS è possibile generare un proprio certificato personale, oppure generarne uno grazie allo script presente al percorso */var/ossec/api/scripts/configure_api.sh.*

Per generare il certificato tramite lo script è sufficiente eseguirlo ed inserire i parametri richiesti, verrà così creato il certificato e le Wazuh API verranno riavviate automaticamente.

Per utilizzare il proprio certificato è necessario andare a personalizzare il file di configurazione presente al path */var/ossec/api/configuration/config.js*

All'interno è presente la seguente sezione:

```
//config.https\_key = "configuration/ssl/server.key"
//config.https\_cert = "configuration/ssl/server.crt"
//config.https\_use\_ca = "no"
//config.https\_ca = "configuration/ssl/ca.crt"
```

In questa sezione è poi necessario eliminare i caratteri per il commento ed andare a sostituire i path di default con quelli necessari per il certificato personalizzato.

Dopo aver configurato il file di configurazione è necessario riavviare le Wazuh API con il comando:

```
systemctl restart wazuh-api
```

6.3.2 Configurazione credenziali sicure

Il secondo passo da seguire per rendere sicuro l'accesso è la configurazione delle credenziali.

Anche in questo caso è possibile effettuare questa operazione tramite script. Basterà eseguire lo script presente al path */var/ossec/api/scripts/configure_api.sh* e seguire le indicazioni. Da qui è possibile creare e modificare username e password per l'accesso a Wazuh. In alternativa è possibile configurare le credenziali tramite il seguente comando:

```
cd /var/ossec/api/configuration/auth
node Password -Bc -C 10 user NomeUtente
```

6.3.3 Modifica della porta di default

Come terzo punto da seguire, è possibile modificare la porta utilizzata di default. Anche qui è possibile utilizzare lo stesso script nominato in precedenza oppure effettuare una

modifica manuale dal file **config.js**

Basterà modificare il parametro alla voce:

```
config.port = "55000";
```

Ed inserire la porta che si vuole andare ad utilizzare, dopodiché è necessario effettuare un restart delle Wazuh API.

7. La dashboard di Wazuh

Possiamo ora effettuare l'accesso alla dashboard principale di Wazuh, tramite la quale potremo tenere sotto controllo tutti gli Agent che si sono registrati, consultare eventi, report, grafici e accedere ai vari moduli della piattaforma (Fig. 7.1). La dashboard sarà il punto di partenza dal quale andare a effettuare i controlli necessari per garantire la sicurezza delle macchine client.

Da qui possiamo consultare la lista degli Agent che sono agganciati al server, infatti possiamo notare le due macchine di test **PC-A** e **PC-B**, oltre ad alcune informazioni di base sui due sistemi (come ad esempio il sistema operativo e la versione dello stesso).

Selezionando il logo di Wazuh in alto a sinistra è invece possibile accedere ad una finestra che ci permetterà di navigare verso tutti gli altri moduli del sistema. Grazie alle varie sezioni della dashboard potremmo infatti spostarci tra le visualizzazioni relative agli eventi di sicurezza, all'integrità dei file, alla gestione degli Agent ed altro ancora. Inoltre dalla scheda Management è possibile andare a gestire le regole e le configurazioni che il Wazuh Manager sta utilizzando

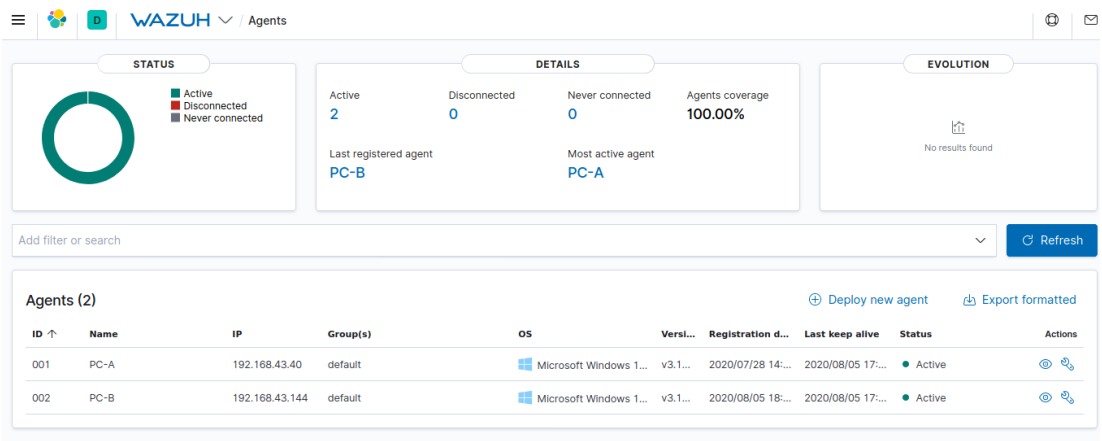

Figura 7.1: La dashboard di Wazuh

8. Analisi delle funzionalità

Per verificare nel dettaglio le funzionalità di Wazuh e le sue potenzialità, è stato eseguito un approfondimento dei vari moduli del sistema.

Inoltre sono stati attuati dei test mirati volti a verificare le capacità della piattaforma.

8.1 Intrusion Detection - Attacco Brute Force in RDP

Per testare le capacità di rilevazione di Intrusion Detection di Wazuh è stato simulato un attacco di forza bruta per tentare di accedere in desktop remoto al PC-A.

L'attacco in RDP è stato effettuato dalla stessa rete della macchina vittima del brute force. Nel PC-A è stato abilitato l'utente amministratore locale presente di default nei sistemi Windows e gli è stata assegnata una password personalizzata.

Per effettuare l'attacco è stato utilizzato un tool specifico per il brute force, ovvero **thc-hydra** [Thc].

Thc-hydra può essere richiamato da terminale e possono essere passati al tool diversi comandi specifici per personalizzare l'attacco in base alle esigenze e alle condizioni.

Nel caso specifico di questo test, il comando lanciato è stato del tipo:

```
hydra -t 10 -V -l administrator -P dizionario.txt rdp://192.68.43.40
```

Con il parametro **-l** andiamo a specificare il nome dell'user con il quale si vuole tentare il login, mentre con il parametro **-P** è possibile specificare un file con estensione .txt contenente le password da dare "in pasto" ad hydra. Dopodiché bisogna specificare il protocollo che si vuole utilizzare e l'indirizzo IP del bersaglio.

In questo test è stato chiesto ad hydra di utilizzare il file *dizionario.txt*, il quale contiene

```
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccardi" - 64713 of 385750 [child 1] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccardica" - 64714 of 385750 [child 2] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccardiche" - 64715 of 385750 [child 3] (0/0
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccardici" - 64716 of 385750 [child 0] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccardico" - 64717 of 385750 [child 1] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccardo" - 64718 of 385750 [child 2] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccare" - 64719 of 385750 [child 3] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccarono" - 64720 of 385750 [child 0] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccata" - 64721 of 385750 [child 0] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccate" - 64722 of 385750 [child 1] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccatelli" - 64723 of 385750 [child 2] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccatello" - 64724 of 385750 [child 3] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccati" - 64725 of 385750 [child 2] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccato" - 64726 of 385750 [child 0] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccava" - 64727 of 385750 [child 1] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccavano" - 64728 of 385750 [child 3] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocce" - 64729 of 385750 [child 0] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocche" - 64730 of 385750 [child 2] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocchesca" - 64731 of 385750 [child 0] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocchesche" - 64732 of 385750 [child 1] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocheschi" - 64733 of 385750 [child 3] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocchesco" - 64734 of 385750 [child 2] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocchi" - 64735 of 385750 [child 0] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocchiere" - 64736 of 385750 [child 0] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocchieri" - 64737 of 385750 [child 1] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocchini" - 64738 of 385750 [child 2] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocchino" - 64739 of 385750 [child 3] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocci" - 64740 of 385750 [child 3] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccia" - 64741 of 385750 [child 0] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocciatrice" - 64742 of 385750 [child 2] (0/0
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocciatrici" - 64743 of 385750 [child 0] (0/0
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocciatura" - 64744 of 385750 [child 1] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocciature" - 64745 of 385750 [child 2] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccio" - 64746 of 385750 [child 3] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "brocco" - 64747 of 385750 [child 0] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccoletti" - 64748 of 385750 [child 1] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccoletto" - 64749 of 385750 [child 2] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccoli" - 64750 of 385750 [child 3] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broccolo" - 64751 of 385750 [child 0] (0/0)
[ATTEMPT] target 192.168.43.40 - login "administrator" - pass "broch" - 64752 of 385750 [child 1] (0/0)
```

Figura 8.1: Output del terminale durante attacco brute force

una serie di parole del dizionario italiano. Il tool in questo caso tenterà di indovinare la password dell'administrator andando a provare una ad una tutte le parole inserite nel file di testo, in pieno stile brute force.

Nella porzione di output (Fig. 8.1) mostrata è possibile notare un elevatissimo numero di tentativi di accesso. Entrando successivamente sulla dashboard di Wazuh e selezionando dalla lista degli Agent il PC-A, risalta subito all'occhio il numero enorme di alert che sono stati generati dal sistema (Fig. 8.2).

Sono stati infatti rilevati più di 50 mila tentativi falliti di accesso. Grazie ai grafici che Wazuh propone è possibile anche farsi un'idea più chiara di quello che può essere successo.

Spostandosi sulla finestra degli eventi di sicurezza possiamo andare a verificare nel dettaglio le informazioni riguardo gli alert generati (Fig. 8.3).

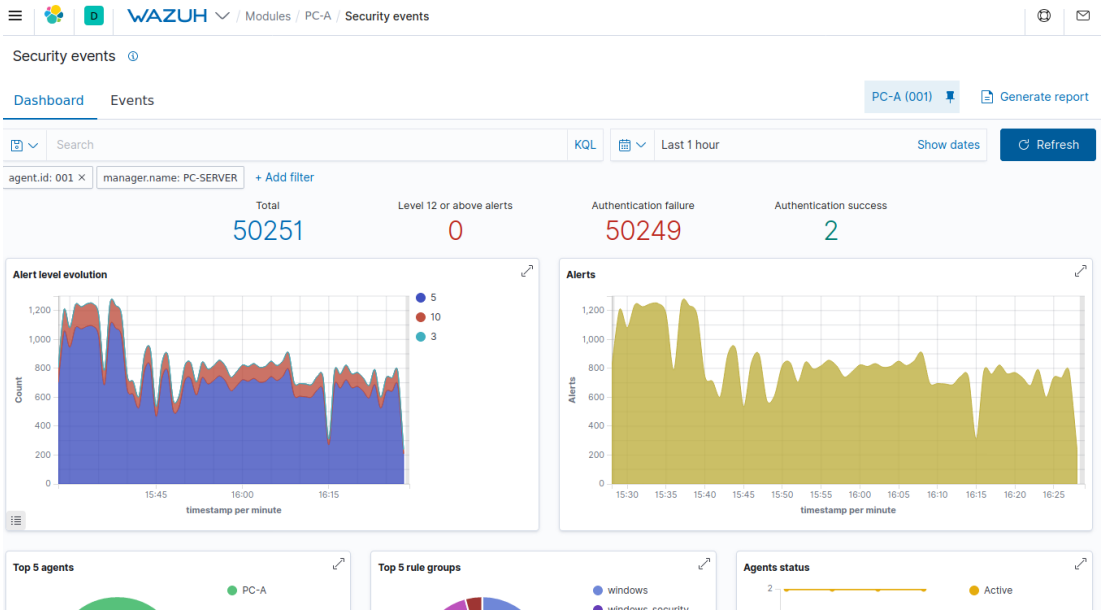

Figura 8.2: Grafici degli alert generati dall'attacco brute force

Come è possibile evincere, vengono segnalati tentativi falliti di accesso al sistema operativo Windows. Wazuh permette di verificare e controllare tutta una serie di informazioni raccolte che ci possono aiutare a mettere in sicurezza il sistema. Aprendo nel dettaglio una delle segnalazioni è infatti possibile ricavare ad esempio:

- Data e ora del tentativo di accesso
- Descrizione della regola che ha attivato l'alert
- Nome utente con il quale si tentava di accedere
- Indirizzo IP sorgente dell'attacco
- Il nome assegnato alla macchina sorgente dell'attacco

Grazie a queste info è piuttosto immediato riuscire a mettere in sicurezza il sistema da un attacco di questo tipo. Come soluzione potrebbe essere presa in considerazione l'idea di impostare una regola su Wazuh che blocca in automatico i tentativi di accesso da parte di quel determinato indirizzo IP.

	Time	agent.id	rule.description	rule.level	rule.id	data.win.eventdata.targetUserName	@src_ip
›	Aug 6, 2020 @ 16:31:21.028	001	Logon Failure - Unknown user or bad password	5	60122	administrator	192.168.43.144
›	Aug 6, 2020 @ 16:31:21.024	001	Logon Failure - Unknown user or bad password	5	60122	administrator	192.168.43.144
›	Aug 6, 2020 @ 16:31:21.020	001	Multiple Windows Logon Failures	10	60204	administrator	192.168.43.144
›	Aug 6, 2020 @ 16:31:21.018	001	Logon Failure - Unknown user or bad password	5	60122	administrator	192.168.43.144
›	Aug 6, 2020 @ 16:31:21.016	001	Logon Failure - Unknown user or bad password	5	60122	administrator	192.168.43.144
›	Aug 6, 2020 @ 16:31:21.016	001	Logon Failure - Unknown user or bad password	5	60122	administrator	192.168.43.144
›	Aug 6, 2020 @ 16:31:21.012	001	Logon Failure - Unknown user or bad password	5	60122	administrator	192.168.43.144
›	Aug 6, 2020 @ 16:31:21.008	001	Logon Failure - Unknown user or bad password	5	60122	administrator	192.168.43.144
›	Aug 6, 2020 @ 16:31:19.796	001	Logon Failure - Unknown user or bad password	5	60122	administrator	192.168.43.144
›	Aug 6, 2020 @ 16:31:19.793	001	Logon Failure - Unknown user or bad password	5	60122	administrator	192.168.43.144

Figura 8.3: Eventi raccolti da Wazuh durante l'attacco brute force

8.2 Vulnerabilty Detection - Ricerca delle vulnerabilità in ambiente Windows

Come accennato in precedenza, Windows è un sistema soggetto a parecchie falle di sicurezza, infatti solo nel 2019 ne sono state scoperte più di 700. Per questo motivo è necessario e importante (soprattutto in ambienti di lavoro) rendersi conto di quali possono essere le vulnerabilità principali, in modo tale da poter intervenire per mettere in sicurezza l'ambiente.

Wazuh ha introdotto la funzionalità del Vulnerability Detection a partire dalla versione 3.11, grazie alla quale è possibile andare ad analizzare le falle nella sicurezza degli Agent monitorati. Il Wazuh Manager sfrutta il **National Vulnerability Database** [Nvd] del governo degli Stati Uniti per rimanere sempre aggiornato in termini di vulnerabilità scoperte.

Prima di iniziare il test è necessario configurare il Wazuh Manager per la rilevazione delle vulnerabilità, per fare ciò è necessario andare a modificare il file di configurazione posizionato in:

```
/var/ossec/etc/ossec.conf
```

All'interno del file va individuata la seguente porzione:

```
<vulnerability-detector>
<enabled>no</enabled>
<interval>5m</interval>
<ignore_time>6h</ignore_time>
<run_on_start>yes</run_on_start>
```

Da qui è possibile abilitare il modulo modificando il valore della tag $<$ *enabled* $>$ in *yes*.

Modificando il valore compreso fra la tag $<$ *interval* $>$ è possibile invece selezionare l'intervallo di tempo fra una scansione e quella successiva.

La tag $<$ *ignore_time* $>$ ci permette di selezionare il valore temporale entro il quale una vulnerabilità già scoperta, non generi un'ulteriore alert.

Se la tag $<$ *run_on_start* $>$ ha come valore *yes*, la scansione delle vulnerabilità e l'aggiornamento del database avverranno all'avvio del servizio.

La porzione successiva da configurare è la seguente:

```
<provider name="nvd">
<enabled>no</enabled>
<update_from_year>2010</update_from_year>
<update_interval>1h</update_interval>
</provider>
```

Scegliamo come provider **nvd**, che è necessario per la rilevazione delle vulnerabilità negli ambienti Windows (nel file di configurazione sono presenti anche i provider per i sistemi basati su Red Hat e Debian che possono essere attivati alla bisogna).

Modifichiamo anche qui il valore della tag $<$ *enabled* $>$ in *yes* per abilitare il provider.

Modificando la tag $<$ *update_from_year* $>$ possiamo selezionare l'anno da cui vogliamo partire per l'indicizzazione del database delle vulnerabilità. Selezionando un anno più lontano cronologicamente, il database di Wazuh sarà sicuramente più completo, ma la mole di informazioni da gestire sarà maggiore.

La tag $<$ *update_interval* $>$ ci permette invece di configurare la frequenza con la quale il Wazuh Manager controlla se il database delle vulnerabilità è stato aggiornato. In caso di esito positivo vengono scaricati gli aggiornamenti.

Una volta effettuate e salvate tutte le modifiche necessarie bisognerà riavviare il servizio

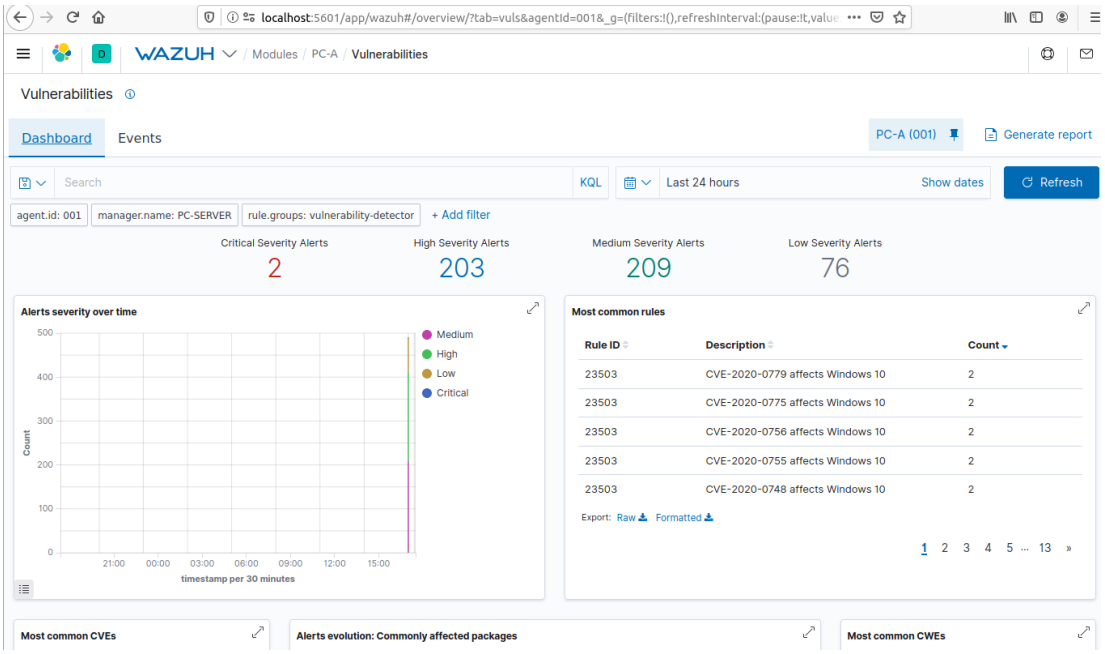

Figura 8.4: Interfaccia delle vulnerabilità riscontrate sul PC-A

del Wazuh Manager per poterle rendere effettive.

```
systemctl restart wazuh-manager
```

Dopo il riavvio, il Manager inizierà a scaricare tutto ciò che è necessario per far funzionare correttamente questo modulo, compreso il database delle vulnerabilità.
Per verificare se il servizio è correttamente partito è possibile leggere i log dal file *ossec.log*.

Collegandosi all'interfaccia web di Kibana, è ora possibile andare a leggere i report e a visualizzare i risultati delle scansioni selezionando la funzione **Vulnerabilities**.

Selezionando come filtro la visualizzazione della scansione delle vulnerabilità sull'Agente **PC-A**, Wazuh ci presenta il risultato in Figura 8.4.

Sono diverse le vulnerabilità riscontrate, di cui 2 di livello critico. Selezionando la finestra **Events** è possibile andare ad analizzare nel dettaglio tutti gli eventi generati da Wazuh, il livello di allerta ed altre informazioni utili che aiutano l'utente a capire l'origine della vulnerabilità riscontrata.

Come è possibile notare dalla Figura 8.5 , sono state riscontrate molte vulnerabilità nel software per la riproduzione di file audio e video VLC Media Player. La versione

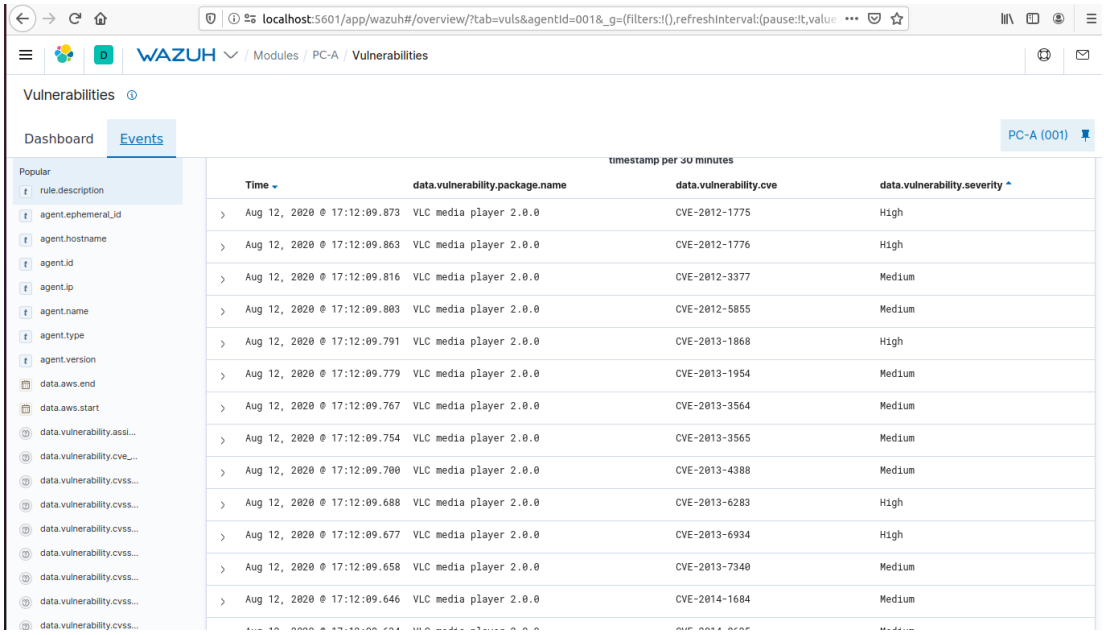

Figura 8.5: Eventi generati dal Wazuh Manager riguardo le vulnerabilità del PC-A

installata sulla macchina client è infatti una versione obsoleta e non aggiornata. Per capire nel dettaglio quali sono i rischi collegati a queste vulnerabilità possiamo andare ad eseguire una ricerca grazie al codice CVE riportato da Wazuh.

Prendiamo come esempio il **CVE-2012-1775** visto che si tratta di una vulnerabilità segnalata ad alto rischio. Colleghiamoci quindi al sito web https://cve.mitre.org/ ed inseriamo il codice CVE di cui vogliamo conoscere i dettagli.

La ricerca effettuata ci ha permesso di visualizzare una descrizione della vulnerabilità riscontrata, il link alla pagina di riferimento NVD ed un'ulteriore serie di riferimenti utili (Fig. 8.6). Grazie ai riferimenti proposti è possibile scoprire che questa vulnerabilità è stata risolta nelle versioni successive dell'applicazione, quindi basterà aggiornare il software all'ultima release distribuita dagli sviluppatori per migliorare la sicurezza del client monitorato.

Un'ulteriore vulnerabilità ad alto rischio che è stata riscontrata da Wazuh è la **CVE-2019-1468**. Si tratta di una vulnerabilità del sistema operativo Windows 10 che permette all'attaccante di eseguire codice malevolo sul client e di prendere il controllo del sistema

L'attacco può partire nel momento in cui l'utente va ad avviare o ad aprire un file

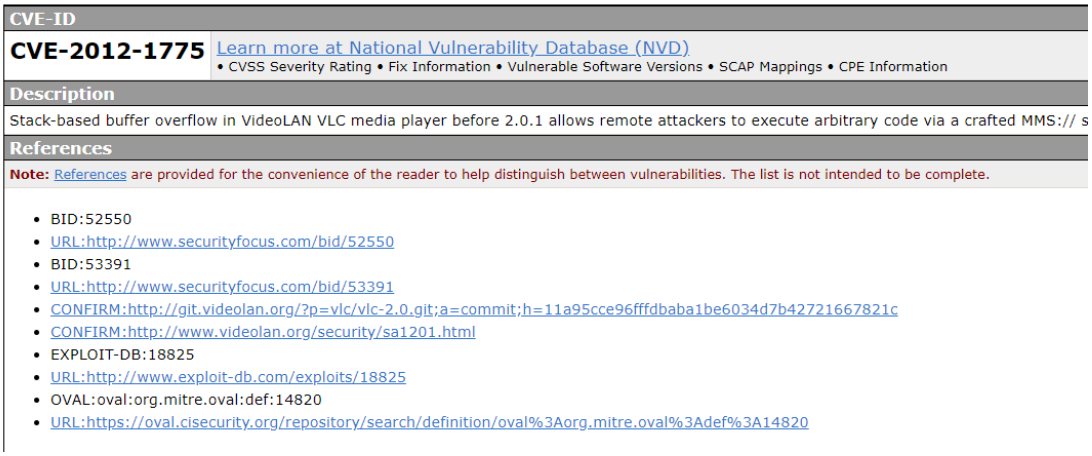

Figura 8.6: Risultato della ricerca tramite CVE

dannoso appositamente camuffato per sembrare sicuro, o attraverso una pagina web creata per questo scopo. Anche in questo caso la soluzione proposta da Microsoft è quella di aggiornare il sistema operativo e scaricare gli ultimi update della sicurezza che vanno a risolvere questa vulnerabilità

8.3 File Integrity Monitoring - Monitorare directory e file

Fra le varie capacità della piattaforma troviamo la possibilità di effettuare File Integrity Monitoring, ovvero il monitoraggio dell'integrità strutturale dei file. Questa funzione di Wazuh si rivela particolarmente utile nel caso in cui venga sfruttata per monitorare directory di sistema che contengono file importanti e/o necessari al corretto funzionamento del client.

Spesso e volentieri, malware e root kit effettuano attacchi andando a sostituire alcuni file del sistema operativo con altri solo all'apparenza uguali, ma che in realtà possono essere molto pericolosi.

Grazie al FIM è possibile tenere sotto controllo tutte le modifiche non autorizzate, ed in particolare nei client in cui è presente un sistema operativo Windows è possibile controllare anche il registro di sistema. Per questo test, è stata creata un'apposita cartella di esempio al path *C:\Software*

Per abilitare il FIM sulla macchina client è necessario modificare il modulo associato aggiungendo le directory che vogliamo andare a monitorare.

Andiamo quindi ad aprire il file di configurazione (C:\Program Files (x86)\ossec-Agent\ossec.conf) e inseriamo la directory nella sezione $< syscheck >$.

```
<syscheck>
  <disabled>no</disabled>
  <frequency>300</frequency>
  <directories check_all="yes" realtime="yes" report_changes="yes">
  C:/Software</directories>
</syscheck>
```

Personalizzando il tag $< frequency >$ è possibile selezionare l'intervallo di tempo fra una scansione ed un'altra. Di norma la scansione viene effettuata ogni 12 ore, ma per effettuare il test abbiamo abilitato il parametro realtime che permette la verifica in tempo reale delle eventuali modifiche.

Dalla GUI per la gestione dell'Agent effettuiamo un riavvio del servizio così da applicare la modifica appena effettuata. Dopodiché nella cartella C:\Software sono stati fatti diversi test, tra cui creazione di file, eliminazione, rinominazione e modifica dei permessi.

Rientrando sull'interfaccia web del Wazuh Manager, selezioniamo l'Agent che ci interessa monitorare e tramite la sezione Integrity Monitor verifichiamo gli alert e gli eventi che sono stati generati dalla piattaforma.

Wazuh ha registrato tutte le modifiche che sono state effettuate (Fig. 8.7), generando un evento per ognuna di esse. Tra le varie info che il sistema mette a disposizione troviamo l'orario in cui è stata fatta la modifica, il tipo di azione effettuata ed una serie di attributi associati al file tra cui:

- Modifica del checksum
- Peso del file
- Modifica dei permessi
- L'utente che ha effettuato la modifica

Grazie a queste info è possibile ricostruire in maniera estremamente precisa tutto quello che è avvenuto all'interno della directory.

›	Aug 17, 2020 @ 16:35:16.031	c:\software\test immagine bitmap.bmp	added	File added to the system.	5	554
›	Aug 17, 2020 @ 16:35:15.980	c:\software\new bitmap image.bmp	deleted	File deleted.	7	553
›	Aug 17, 2020 @ 16:35:08.689	c:\software\new bitmap image.bmp	added	File added to the system.	5	554
›	Aug 17, 2020 @ 16:35:02.202	c:\software\file di testo di esempio 2.txt	modified	Integrity checksum changed.	7	550
›	Aug 17, 2020 @ 16:34:36.786	c:\software\file di testo di esempio 2.txt	modified	Integrity checksum changed.	7	550
›	Aug 17, 2020 @ 16:34:26.821	c:\software\file di testo di esempio 2.txt	modified	Integrity checksum changed.	7	550
›	Aug 17, 2020 @ 16:34:11.867	c:\software\file di testo di esempio 2.txt	modified	Integrity checksum changed.	7	550
›	Aug 17, 2020 @ 16:33:59.712	c:\software\file di testo di esempio 2.txt	added	File added to the system.	5	554
›	Aug 17, 2020 @ 16:33:59.694	c:\software\file di testo di esempio ◆◆.txt	deleted	File deleted.	7	553
›	Aug 17, 2020 @ 16:33:51.822	c:\software\file di testo di esempio ◆◆.txt	added	File added to the system.	5	554
›	Aug 17, 2020 @ 16:33:51.704	c:\software\new text document.txt	deleted	File deleted.	7	553

Figura 8.7: Eventi generati dal modulo FIM

8.3.1 Integrazione di VirusTotal

VirusTotal [Vir] è un servizio online che permette di analizzare gratuitamente file e indirizzi web per rilevare malware ed altro codice malevolo al loro interno. Questo strumento utilizza le risorse di oltre 70 antivirus, ottenendo così un enorme potenziale. Wazuh permette di integrare il modulo di File Integration Monitor con le API di VirusTotal.

Grazie alla collaborazione con questo servizio, Wazuh può fare automaticamente una richiesta alle API di VirusTotal inviando gli hash dei file (Fig. 8.8). Se VirusTotal dovesse segnalare i file come pericolosi, Wazuh genererà di risposta un alert.

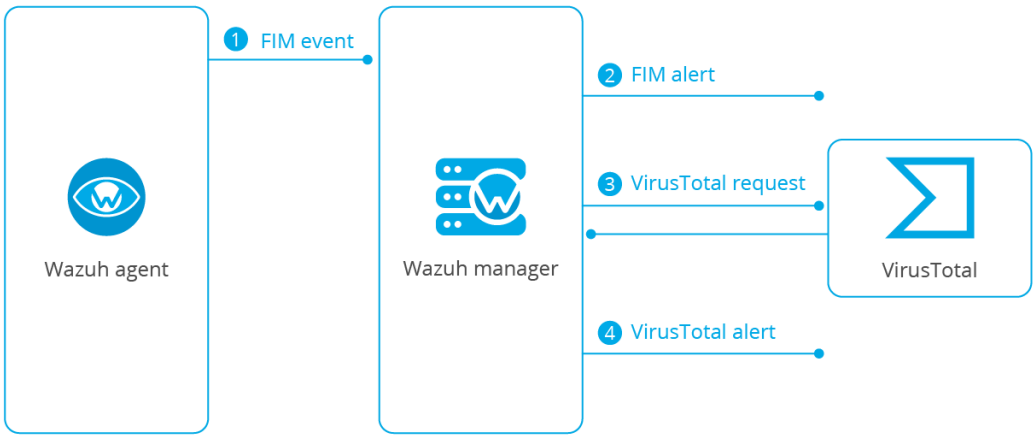

Figura 8.8: Data flow fra Agent, Wazuh Manager e VirusTotal

Fonte: https://wazuh.com/blog/emotet-malware-detection/

Per abilitare l'integrazione con VirusTotal è necessario andare a modificare il file di configurazione del Wazuh Server ed aggiungere i seguenti parametri:

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

Nella tag $< api\_key >$ è necessario inserire la chiave API personale del servizio VirusTotal.

L'integrazione del servizio VirusTotal può rivelarsi estremamente utile contro tutti quei malware che tentano di sostituire file di sistema. Inoltre potrebbe essere sfruttato per controllare la cartella dei download effettuati dall'utente finale della macchina client, visto che il pericolo di scaricare file infetti e pericolosi è spesso dietro l'angolo.

Molti dei malware più pericolosi, come ad esempio Emotet, (un pericoloso virus nato con l'intento di rubare informazioni bancarie sensibili agli utenti) si nascondono all'interno di file pressoché innocui e un utente poco esperto potrebbe essere facilmente tratto in inganno.

Grazie all'integrazione con VirusTotal, la piattaforma Wazuh riesce ad essere un ottimo alleato nella difesa dei sistemi.

8.4 System Inventory - Raccolta delle informazioni sugli Agent

Grazie al modulo *Syscollector*, gli Agent sono in grado di comunicare al Wazuh Manager moltissime informazioni riguardo il sistema. È infatti possibile creare un vero e proprio inventario capace di fornire info riguardo l'hardware della macchina, i software installati e molto altro.

All'avvio dell'Agent, il *Syscollector* eseguirà periodicamente delle scansioni in modo tale da raccogliere tutte le info richieste, le quali saranno poi inoltrate al Wazuh Manager. Il manager aggiornerà di conseguenza i record del database associato al suddetto Agent.

Alcune di queste informazioni (come ad esempio la lista dei software installati sulla macchina) saranno poi riutilizzate dal modulo per la ricerca delle vulnerabilità. L'inventario può essere poi consultato dalla voce Inventory dell'interfaccia web di Wazuh.

Il Syscollector è in grado di raccogliere molte informazioni riguardo l'hardware dell'Agent e dati riguardo il sistema operativo (Fig. 8.9). Fra le altre possibilità troviamo poi la capacità di riportare l'intera lista dei software installati sulla macchina, informazioni riguardo le interfacce di rete, la lista delle porte aperte ed i processi attivi al momento.

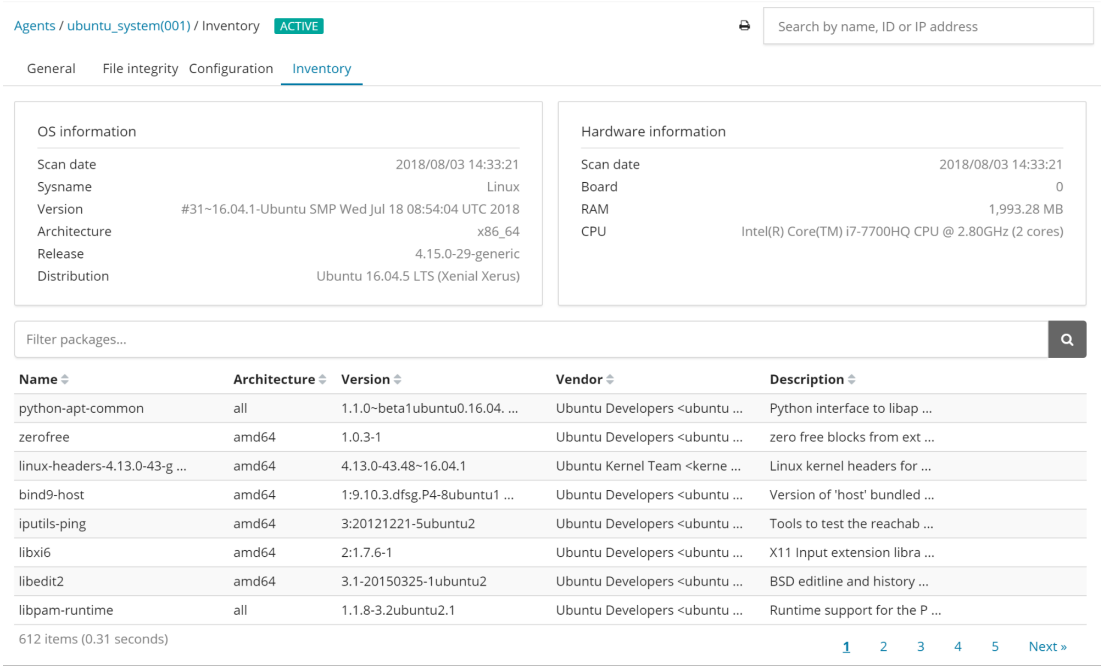

Figura 8.9: Schermata del System Inventory

Anche nel caso del Syscollector è possibile andare a specificare delle regole personalizzate per far scattare degli alert in base alle informazioni raccolte da questo modulo.

Nell'esempio proposto dalla documentazione ufficiale della piattaforma viene mostrato come creare una regola che permette la generazione di un evento nel caso in cui l'interfaccia di rete eth0 venga abilitata.

```
<rule id="100001" level="5">
<if_sid>221</if_sid>
<decoded_as>syscollector</decoded_as>
<field name="netinfo.iface.name">eth0</field>
<description>eth0 interface enabled. IP:
$(netinfo.iface.ipv4.address)</description>
</rule>
```

L'evento generato riporterà l'avvenuta abilitazione della scheda di rete e l'indirizzo IPv4 assegnato ad essa.

8.5 Active Response

La capacità di eseguire Active Response rappresenta sicuramente una delle funzionalità più importanti del Wazuh Manager. Per Active Response si intende infatti la capacità

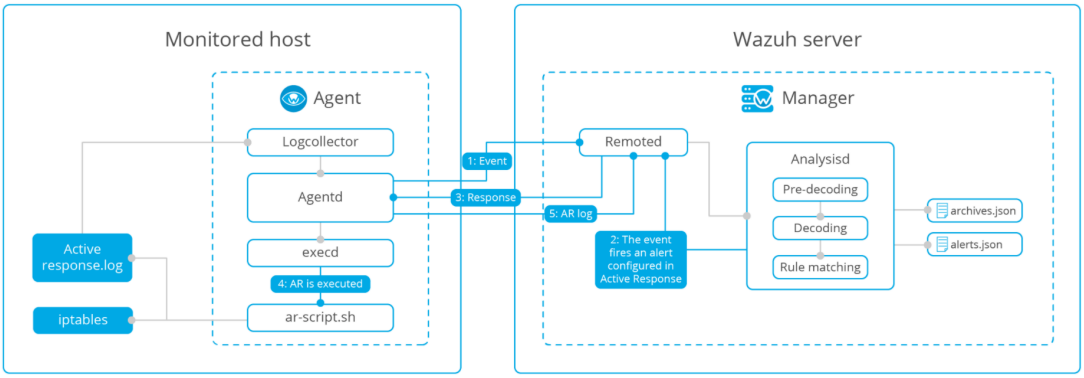

Figura 8.10: Data Flow dell'Active Response

Fonte: https://documentation.wazuh.com/3.13/user-manual/capabilities/active-response/

di intervenire in maniera pro-attiva nel caso in cui vengano soddisfatti determinati criteri.

Un esempio immediato potrebbe essere rappresentato dal blocco di un certo indirizzo IP dopo diversi tentativi falliti di accesso.

L'Active Response può essere configurato per intervenire in due modi:

Stateful: In questo caso, le azioni che vengono effettuate in risposta ad un evento generato modificano l'ambiente per una quantità di tempo limitata, dopodiché viene ripristinata la situazione originale
Stateless: Nel caso di azioni di questo tipo, le modifiche effettuate sono permanenti

Ad ogni risposta generata da questo modulo è associato uno specifico comando che sarà poi eseguito sull'Agent coinvolto, sul Wazuh Manager o su tutti gli Agent (Fig. 8.10).

L'Active Response viene configurata nel file *ossec.conf* del Wazuh Manager tramite diversi passaggi.

Come primo passaggio bisogna definire un comando che eseguirà uno script nel caso in cui vengano soddisfatti i criteri stabiliti:

```
<command>
<name>host-deny</name>
<executable>host-deny.sh</executable>
<expect>srcip</expect>
<timeout_allowed>yes</timeout_allowed>
```

```
</command>
```

Dopo aver definito il Command, si passa alla configurazione della regola di active response.
La regola di Active Response permette di definire quando e dove il comando (ed il relativo script) sarà eseguito. Inoltre è possibile definire il dominio nel quale il comando deve essere eseguito (solo sull'Agent, sul manager, su tutti gli Agent ecc.)

```
<active-response>
<command>host-deny</command>
<location>local</location>
<level>7</level>
<timeout>600</timeout>
</active-response>
```

In questo esempio, la regola viene definita per intervenire sull'Agent locale nel momento in cui viene rilevato un evento con livello di sicurezza 7 o superiore. Nel momento in cui la regola viene attivata, viene eseguito il comando **host-deny** definito poco prima.

Wazuh presenta una serie di script preconfigurati che possono essere impostati per fare active response sia su sistemi operativi Linux che Windows. Basterà quindi consultare la lista degli script disponibili dalla documentazione ufficiale per poter andare a personalizzare il nostro ambiente nella maniera che riteniamo più efficiente.

8.6 Security Configuration Assessment

Tra i modi migliori per rafforzare la sicurezza degli host è presente sicuramente la pratica dell' "hardening" ovvero la riduzione della "superficie esposta" (e quindi potenzialmente vulnerabile).

Il Security Configuration Assessment può aiutare l'utente ad effettuare l'hardening e a scoprire configurazioni errate sugli host monitorati. Grazie agli scan effettuati da questo modulo è infatti possibile scoprire e valutare se sono necessari cambi di password, se è necessario rimuovere software superfluo, se è necessario disabilitare dei servizi ed altro ancora.

Le policy per le scansioni SCA sono state create in YAML. Questo formato è stato scelto per la sua leggibilità da parte degli umani, in modo tale che gli utenti possano

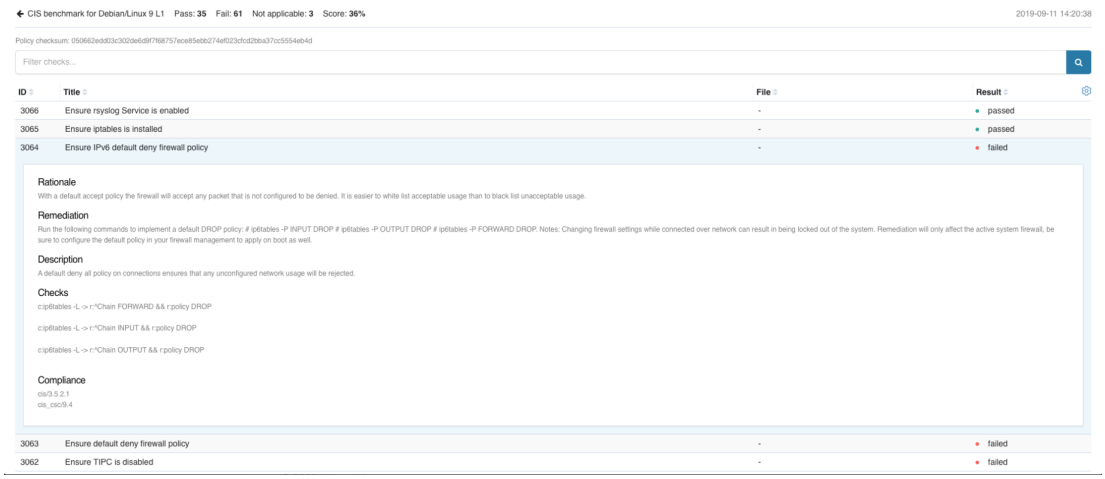

Figura 8.11: Risultati della scansione SCA

capire da subito e senza difficoltà quali policy andrebbero riviste e/o corrette.

Wazuh è distribuito con un set di policy standard basato sui benchmark CIS [Cis] ma è possibile scrivere nuove policy o modificare le esistenti per adattarle alle possibili esigenze.

Nel test di scansione SCA effettuato nell'ambiente di lavoro (Fig. 8.11), è possibile notare come venga spiegato e documentato il motivo per cui il test di riferimento per le policy dedicate agli IPv6 non sia passato.

Inoltre è presente anche il rimedio suggerito dalla piattaforma per eliminare la falla in questione.

9. Ulteriori funzionalità della piattaforma

9.1 Log Collection - Raccolta dei log di Windows

Gli eventi generati dai sistemi operativi Windows possono essere raccolti ed analizzati dal Wazuh Manager, che farà scattare degli alert nel caso in cui siano delle corrispondenze con le regole stabilite.

Esistono principalmente due formati nel quale è possibile salvare e raccogliere gli eventi di Windows:

Eventlog - (Supportati da qualsiasi versione di Windows)
Eventchannel - (Supportati a partire da Windows Vista in poi)

Gli eventi di Windows sono messaggi descrittivi che contengono informazioni utili riguardo la maggior parte di ciò che avviene all'interno del sistema. Sono catalogati in base al tipo e all'origine all'interno dell'Event Viewer. Wazuh è in grado di analizzare entrambi i formati (Sia Eventlog che Eventchannel) grazie alle API di Windows e riesce ad importare tutte le informazioni necessari per l'analisi.

Gli eventi di tipo Eventlog supportano qualsiasi sistema operativo basato su Windows e possono monitorare qualsiasi tipo di log eccetto gli eventi generati da particolari applicazioni e dai servizi.

Gli eventi di tipo Eventchannel invece possono monitorare anche gli eventi delle applicazioni e dei servizi, risultando così un soluzione più completa ed integra.

9.2 Command Monitoring

Nel caso in cui si volessero monitorare alcune info non presenti nei classici log di sistema, Wazuh permette di monitorare l'output di alcuni specifici comandi di sistema. Per implementare questa funzione, è necessario configurare l'Agent remoto per permettergli di accettare comandi dal Wazuh Manager.

Per fare ciò, è necessario andare a modificare il file *local_internal_options.conf* dell'Agent interessato ed inserire la seguente stringa:

```
logcollector.remote_commands=1
```

Successivamente è possibile ritoccare il file *ossec.conf* per andare ad inserire i comandi da monitorare.

```
<localfile>
 <log_format>full_command</log_format>
 <command>....</command>
 <frequency>120</frequency>
</localfile>
```

Si potrebbe ad esempio monitorare l'utilizzo del disco di un Agent e grazie ad una regola personalizzata generare un alert se questo valore dovesse superare una certa soglia.

È facile quindi immaginare come sia possibile costruire un insieme di comandi e regole che facciano al nostro caso.

10. Management dei backup

Per essere sempre sicuri di non perdere i dati raccolti e indicizzati, è buona norma eseguire periodicamente dei backup. Wazuh permette di gestire i backup in maniera automatizzata e di salvare i file in locale o in cloud.

Per salvare i dati raccolti da Elasticsearch vengono utilizzati degli snapshot incrementali, quindi ogni snapshot conterrà solo le informazioni non contenute nello snapshot precedente.

Per automatizzare il processo di backup è necessario prima di tutto creare una repository dove poter salvare gli snapshot generati. Esistono diversi tipi di repository e come anticipato prima è possibile salvare i file sia in locale che in cloud.

10.1 Repository basata su File System

Nell'esempio andremo a salvare gli snapshot in locale usando il file system della macchina in cui è in esecuzione il servizio di Elasticsearch.

Come primo punto da seguire è necessario specificare un punto per il mount con abbastanza spazio su disco disposizione:

```
chown elasticsearch: /mount/elasticsearch_backup/
```

Dove *elasticsearch_backup* è il nome della nostra cartella.

Dopodichè è necessario aggiungere il path della repository appena creata nel file di configurazione di Elasticsearch (il file si trova al percorso */etc/elasticsearch/elasticsearch.yml*).

```
path.repo: ["/mount/elasticsearch_backup"]
```

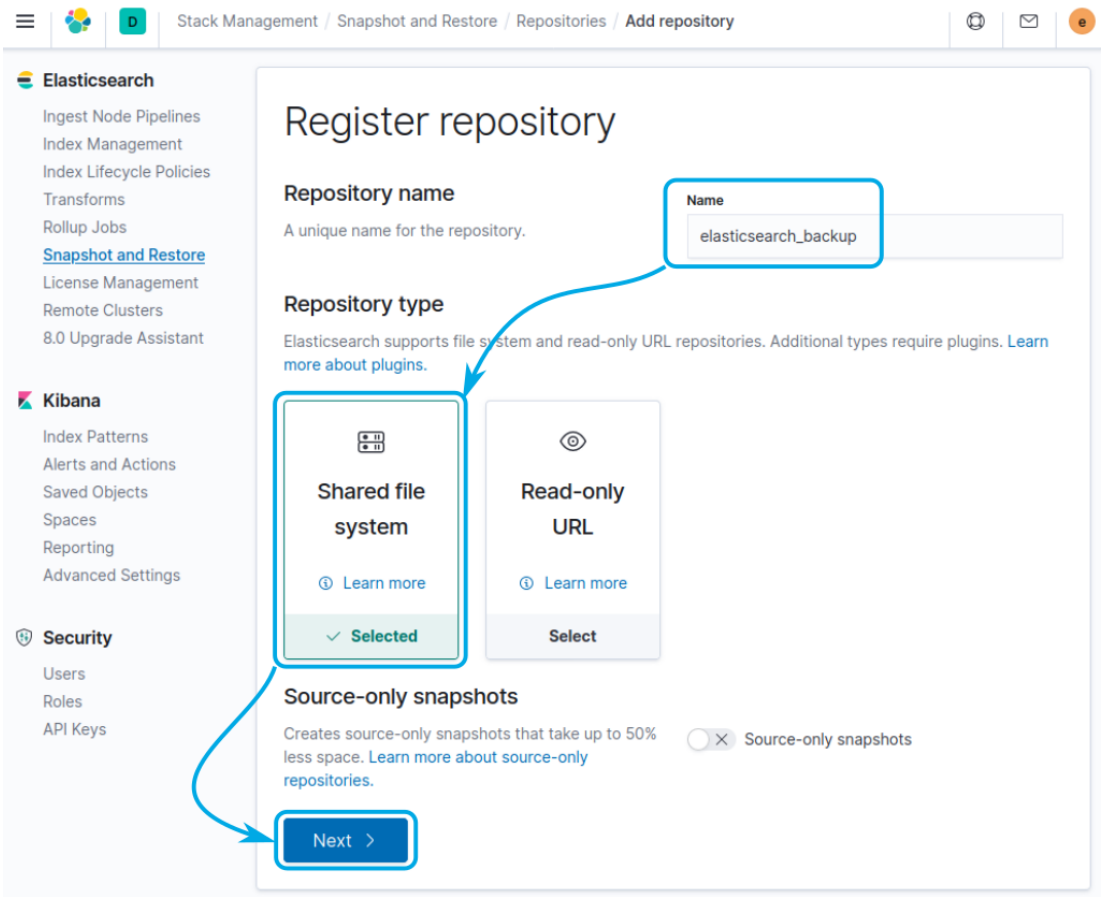

Figura 10.1: Interfaccia per la registrazione della Repository

Il servizio di Elasticsearch andrà poi riavviato:

```
systemctl restart elasticsearch
```

Ora è possibile configurare la repository dall'interfaccia web di Kibana.
Per fare ciò basterà collegarsi all'applicazione e cliccare su:
Stack Management → Snapshot and Restore → Repositories → Register a Repository.

Da qui (Fig. 10.1) è possibile inserire il nome della repository (nel nostro caso *elasticsearch_backup*) e selezionare la tipologia **Shared File System**.

Come ultima informazione da inserire, è necessario specificare il path della repository dove verranno salvate le informazioni, dopodiché la registrazione è stata completata.

In alternativa è possibile salvare i backup tramite repository in cloud su Amazon Web Services, Microsoft Azure o Google Cloud.

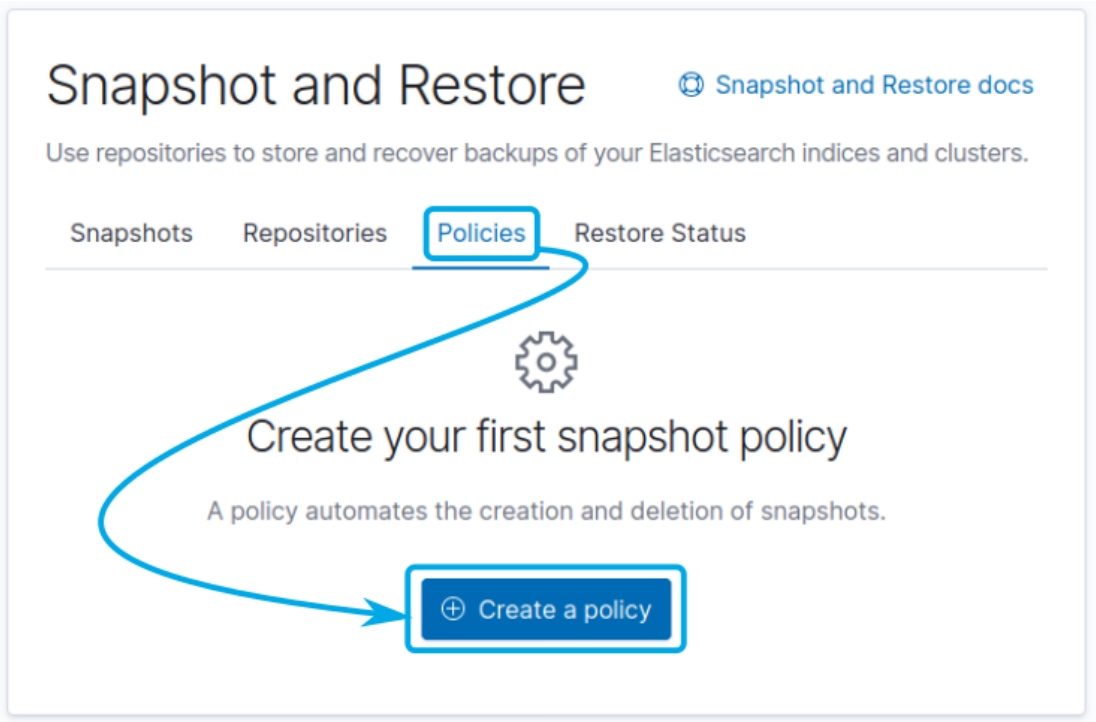

Figura 10.2: Interfaccia per la creazione di policy di backup personalizzate

Per i servizi in cloud è necessario installare dei plugin tramite i seguenti comandi:

```
./elasticsearch-plugin install repository-s3
./elasticsearch-plugin install repository-azure
./elasticsearch-plugin install repository-gcs
```

Dopo aver effettuato l'installazione dei plugin e aver riavviato il Wazuh Manager è possibile impostare la repository in cloud seguendo la stessa procedura ma selezionando il tipo di repository in cloud interessato.

10.2 Salvataggio automatico degli snapshot

Per automatizzare la creazione di snapshot è necessario impostare una policy personalizzata andando su:
Stack Management → Snapshot and Restore → Repositories → Policies
Dopodiché basterà cliccare sul pulsante "Create a Policy" (Fig. 10.2).

Dalla schermata successiva è poi possibile impostare la configurazione della policy nella maniera più adatta alle nostre esigenze. È infatti possibile selezionare la schedulazione

degli snapshot scegliendo frequenza e orario di creazione. Inoltre è possibile automatizzare la cancellazione dei vecchi snapshot per liberare spazio su disco.

Dal percorso:
Stack Management → Snapshot and Restore → Snapshots
è poi possibile eseguire il restore degli snapshot salvati.

11. Rilascio centralizzato del file di configurazione per gli Agent

In un ambiente di grandi dimensioni in cui il numero di Agent da monitorare è molto elevato, è impossibile pensare di poter gestire le configurazioni modificando singolarmente i file di configurazione.

Proprio per questo motivo, Wazuh permette di effettuare un rilascio centralizzato del file di configurazione attraverso i gruppi. Di norma, ogni Agent fa parte del gruppo di default e di conseguenza è soggetto alle regole standard di Wazuh.

Grazie ai gruppi è possibile creare degli insiemi di Agent, divisi secondo tutti i criteri che riteniamo possano esserci utili. Ad ogni gruppo può essere poi assegnato un file di configurazione differente, in modo tale da assecondare i bisogni di quelle specifiche macchine.

La gestione dei gruppi è altamente intuitiva e può essere eseguita attraverso l'interfaccia di Kibana.

11.1 Creazione dei gruppi e aggiunta degli Agent

Per creare un gruppo personalizzato, basta accedere all'applicazione di Wazuh dall'interfaccia di Kibana e selezionare Management → Groups. Da qui (Fig. 11.1) è possibile creare, modificare ed eliminare i vari gruppi di Agent. I gruppi creati saranno poi aggiunti alla lista dei gruppi disponibili.

Per aggiungere degli Agent al gruppo appena creato, bisognerà entrare all'interno del gruppo e selezionare l'icona "*Add or remove agent*". Tramite un'interfaccia è possibile aggiungere o rimuovere in maniera intuitiva gli Agent interessati.

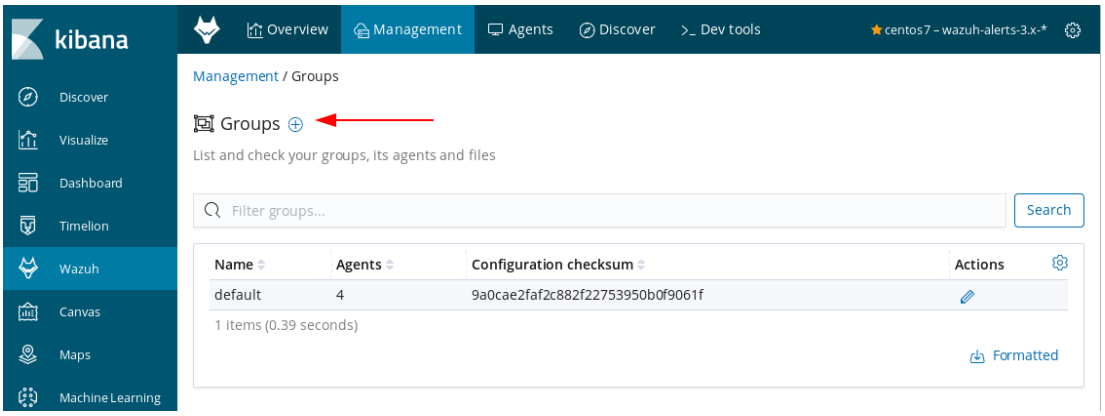

Figura 11.1: Interfaccia per la creazione di gruppi

Di norma, gli Agent aggiunti al gruppo appena creato saranno ancora all'interno del gruppo di default, è necessario quindi rimuoverli manualmente dal gruppo default in caso fosse necessario.

11.2 Modifica del file di configurazione

Una volta che gli Agent interessati sono stati aggiunti al gruppo, è poi possibile modificare il file di configurazione selezionando Content → agent.conf. Nella schermata che si presenta (Fig. 11.2) è possibile modificare secondo necessità il file di configurazione, dopodiché basterà confermare le modifiche effettuate ed il Wazuh Server inoltrerà il file agli Agent coinvolti.

Dopo un riavvio del servizio degli Agent (che avviene in automatico) il file di configurazione sarà importato ed utilizzato.

In questo modo è possibile adattare facilmente i moduli da attivare e le scansioni da effettuare in base alle specifiche dei vari Agent (ad esempio differenziando i gruppi in base al sistema operativo delle macchine monitorate).

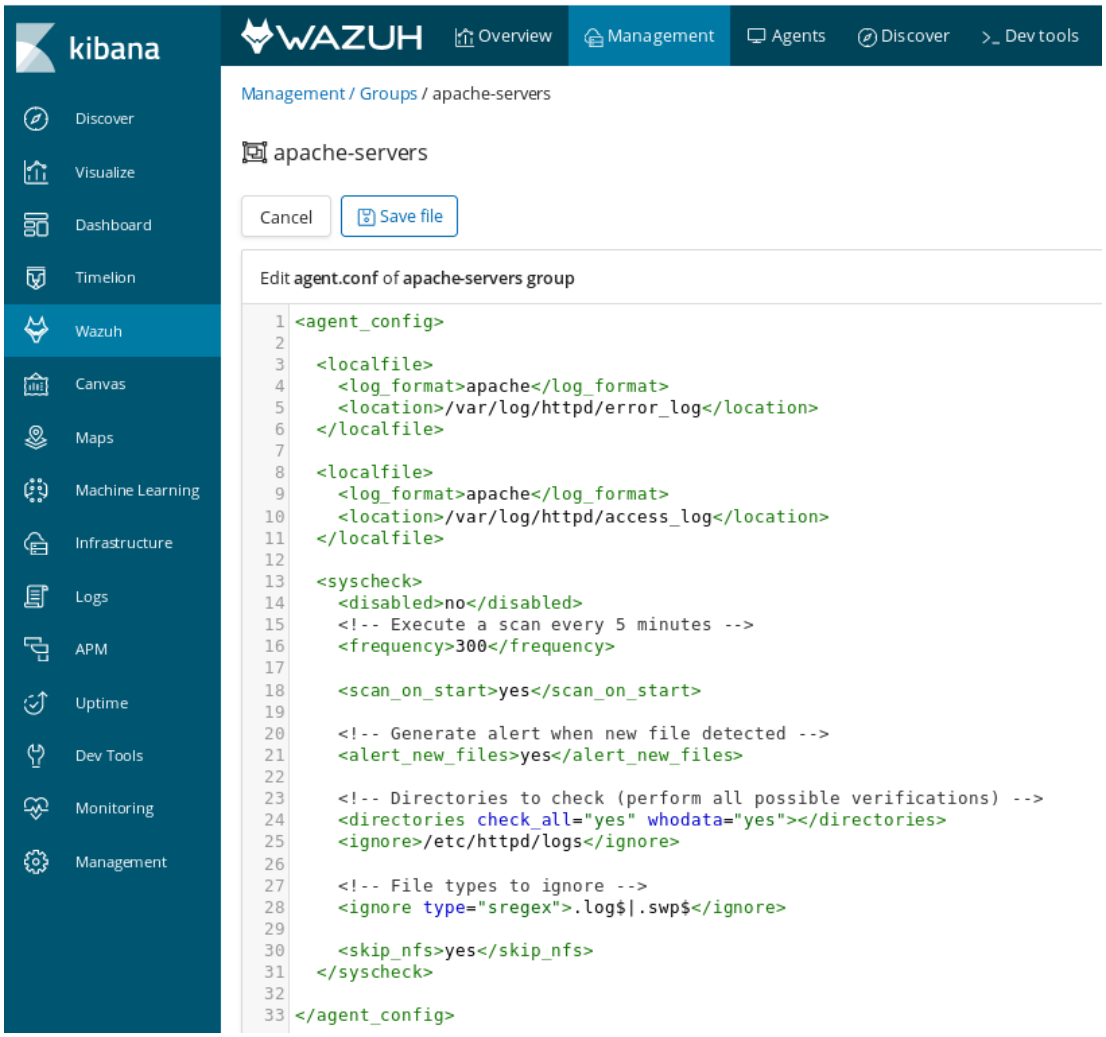

Figura 11.2: Interfaccia per la modifica del file di configurazione degli Agent

12. Conclusioni

Wazuh ha dimostrato di essere una piattaforma capace di soddisfare molte esigenze, grazie alle varie funzionalità proposte e alla sua natura versatile. Il più grande punto di forza di questo sistema è sicuramente la possibilità di essere adattato e personalizzato in base alla situazione e all'architettura in cui deve essere rilasciato.

Grazie infatti alla capacità di Wazuh di essere uno strumento compatibile con diversi sistemi operativi e con macchine client diverse fra loro è semplice immaginare come possa essere sfruttato in contesti molto differenti.

Wazuh risulta essere uno strumento che può fornire un grande aiuto per migliorare la sicurezza dei client sia in ambienti piccoli, sia in ambienti aziendali con decine di macchine collegate alla rete.

Per poter essere utilizzata al meglio, è però necessaria della formazione di utenti specializzati nella gestione dell'intera piattaforma. L'utilizzo di Wazuh in modo superficiale è infatti accessibile a qualunque utente con un minimo di dimestichezza nel settore informatico, ma la preparazione di un ambiente per un rilascio in ambito aziendale presuppone una conoscenza profonda e concreta della piattaforma.

Una preparazione frettolosa dell'ambiente potrebbe infatti portare a configurare regole di Active Response errate, le quali potrebbero portare ad un risultato contrario a quello sperato, ovvero la creazione di vulnerabilità o falle

Per questo motivo è altamente consigliato effettuare modifiche e valutazioni in un ambiente dedicato esclusivamente ai test. Grazie ad uno studio approfondito delle capacità di Wazuh è inoltre possibile adeguare la piattaforma alle esigenze aziendali, che possono variare notevolmente in base a molti fattori, fra cui policy interne, tipologia di architettura già in utilizzo e così via.

In conclusione, Wazuh è uno strumento dalle grandi capacità che può essere utilizzato per migliorare notevolmente la sicurezza informatica delle macchine monitorate. Grazie ai report e agli eventi generati è possibile individuare in maniera immediata dove c'è bisogno di intervenire per andare a eliminare le vulnerabilità dei vari client, creando così un ambiente più sicuro dalle minacce informatiche più comuni.

Inoltre il team di sviluppo dietro al progetto Wazuh è costantemente attivo per il rilascio di nuove funzionalità e migliorie, grazie anche al supporto degli stessi utenti sui canali ufficiali della piattaforma.

Il progetto è quindi sicuramente destinato ad essere ancora ampliato, rendendo così Wazuh un alleato invidiabile nel campo della sicurezza informatica.

12.1 Sviluppi futuri

La flessibilità della piattaforma Wazuh permette di adeguare il sistema in base a disparate esigenze aziendali. Con una formazione adeguata sarà possibile implementare la piattaforma all'interno di infrastrutture aziendali anche già ben definite.

Inoltre grazie alle soluzioni in Cloud proposte dal team di sviluppo è possiible integrare la piattaforma senza installare il Wazuh Server in una macchina in locale.

L'obiettivo futuro sarà quello di sfruttare le conoscenze apprese durante la scrittura di questa tesi per applicarle nel mondo del lavoro.

Bibliografia

[Ant] *Anthesia.net*. URL: https://www.anthesia.net/.

[Cis] *CIS Benchmarks*. URL: https://www.cisecurity.org/cis-benchmarks/.

[Doc] *Documentazione ufficiale di Wazuh*. URL: https://documentation.wazuh.com/3.13/index.html.

[Nvd] *National Vulnerability Database*. URL: https://nvd.nist.gov/.

[Oss] *Ossec*. URL: https://www.ossec.net/.

[Sam] *Samhain*. URL: https://www.la-samhna.de/samhain/.

[Sol] *SolarWinds Security Event Manager*. URL: https://www.solarwinds.com/.

[Thc] *Repository GitHub Thc-Hydra*. URL: https://github.com/vanhauser-thc/thc-hydra.

[Tri] *Tripwire*. URL: https://www.tripwire.com/.

[Upg] *Upguard*. URL: https://www.upguard.com/.

[Vir] *VirusTotal*. URL: https://www.virustotal.com/.

[Waz] *Sito ufficiale del progetto Wazuh*. URL: https://wazuh.com/.

[You] *Canale YouTube ufficiale di Wazuh*. URL: https://www.youtube.com/channel/UC3Kr7V99AXOOOuPy4bLhS8w.

Elenco delle figure

1.1 NIDS e HIDS a confronto 10

4.1 Struttura dell'Agent 20
4.2 Struttura del Wazuh Server 21

5.1 Architettura classica di Wazuh 24
5.2 Architettura per infrastutture di piccole dimensioni 24

6.1 Ambiente di test 28
6.2 GUI per configurazione dell'Agent 32

7.1 La dashboard di Wazuh 37

8.1 Output del terminale durante attacco brute force 40
8.2 Grafici degli alert generati dall'attacco brute force 41
8.3 Eventi raccolti da Wazuh durante l'attacco brute force 42
8.4 Interfaccia delle vulnerabilità riscontrate sul PC-A 44
8.5 Eventi generati dal Wazuh Manager riguardo le vulnerabilità del PC-A . 45
8.6 Risultato della ricerca tramite CVE 46
8.7 Eventi generati dal modulo FIM 48
8.8 Data flow fra Agent, Wazuh Manager e VirusTotal 49
8.9 Schermata del System Inventory 51
8.10 Data Flow dell'Active Response 52
8.11 Risultati della scansione SCA 54

10.1 Interfaccia per la registrazione della Repository 58

10.2 Interfaccia per la creazione di policy di backup personalizzate 59

11.1 Interfaccia per la creazione di gruppi 62

11.2 Interfaccia per la modifica del file di configurazione degli Agent 63

Ringraziamenti

Ringrazio la mia famiglia per il supporto che mi ha dato durante il mio percorso universitario.
Ringrazio la mia compagna Beatrice per aver sempre creduto in me.
Ringrazio inoltre il Prof. Marcantoni per avermi fatto appassionare al tema della sicurezza informatica.