UNIVERSITÀ DEGLI STUDI DI MILANO

DIPARTIMENTO DI INFORMATICA

Corso di Laurea in Sicurezza dei Sistemi e delle Reti Informatiche

Threat Modeling Process: Strumenti per la gestione del processo di modellamento delle minacce

RELATORE

Dott.ssa Elvinia Riccobene

CORRELATORE

Dott. Marco Anisetti

TESI DI LAUREA DI

Vincenzo Paolo Di Perna

Matr. 872875

Anno Accademico 2017/2018

“ Don’t repeat the tactics which have gained you one victory,
but let your methods be regulated by the infinite variety
of circumstances.”
Sun Tzu

Prefazione

Spesso e volentieri, siamo abituati ad affrontare i problemi dopo che questi abbiano causato danni, presentandoci il conto. Eppure non è ancora abbastanza per farci reagire e prendere delle precauzioni.

“ Prevenire è meglio che curare. ” Ippocrate

Nella vita reale come in altri ambiti, ad esempio in medicina, la prevenzione è segno di forte consapevolezza dei rischi e un buon segnale di prudenza verso agenti esterni che possono indebolirci.

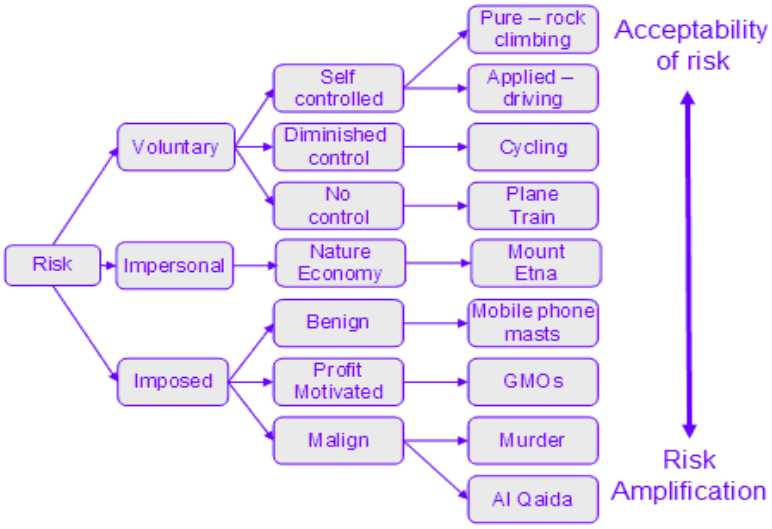


Figura X.1 – Accettabilità e ampliamento del rischio[1]

Sistemi complessi, con un elevato numero di elementi che interagiscono, spesso presentano comportamenti caotici e non prevedibili. Edward Lorenz nel 1960 ebbe modo di scoprire che questi sistemi erano influenzati dall’*effetto farfalla* [1] cioè la dipendenza da condizioni iniziali. E se consideriamo la *teoria del Caos*[2] , è proprio questo, la difficoltà di comprendere il non-determinismo rispetto all’imprevedibilità di questi sistemi. Allerta però: bisogna stare attenti a

[1] Definisce come variazioni sulle condizioni iniziali di un sistema portino a grandi variazioni sul suo comportamento, soprattutto nel lungo termine.
[2] http://leganerd.com/2016/08/01/la-teoria-del-caos/

non confondere il tutto con la casualità, poiché volendo potremmo descrivere questi sistemi in modo preciso, con equazioni e rappresentazioni.

Un sistema informatico rappresenta molto di quanto detto: diversi componenti elettronici, tecnologie software e hardware con funzioni diverse, interagiscono con lo scopo sia di far parte di una macchina - i personal computer - sia rendendo possibile l'implementazione di un sistema, definendone la sua struttura. La teoria del caos è un'ottima introduzione al concetto di modellamento delle minacce per un sistema: abbiamo detto che i comportamenti che osserviamo spesso non sono prevedibili, ed è proprio questa la caratteristica che bisogna contro-adottare per ottenere un adeguato e sufficiente livello di sicurezza, la prevenzione.

Più di una volta ho avuto modo di confrontarmi con un particolare proverbio giapponese accostato a scienziati, filosofi e altri importanti figure, e al quale viene fatto riferimento anche in pellicole cinematografiche. Eccone l'interpretazione migliore:

" Per un chiodo mancante si perse il ferro di cavallo,
per un ferro di cavallo mancante si perse il cavallo,
per un cavallo mancante il messaggio non fu consegnato,
per un messaggio non consegnato si perse la battaglia,
perdendo quella battaglia si perse la guerra,
perdendo la guerra si perse la libertà,
e tutto ciò per un chiodo. "

Riconducendoci alla complessità dei sistemi informatici, dobbiamo tenere conto di questa complessità, di questi componenti, di queste macchine come qualcosa a cui dedicare una cura maniacale e dettagliata, pur comportando uno sforzo considerevole, ma con profitti appaganti. Focalizzarsi solo su alcuni componenti o casistiche, non permette di avere un'ampia visione di quello che succede in un sistema informatico. Seppur il proverbio mostra una conclusione abbastanza tragica rispetto al pericolo che può accorrere ad un utente che fa un utilizzo basilare del PC, non bisogna sottovalutarne gli accostamenti toccanti aziende, organizzazioni, enti di sicurezza nazionale, sistemi economico-finanziari e bancari, ecc. Le conseguenze che questi sistemi dovrebbero affrontare si potrebbero trasformare in danni davvero rilevanti per l'ambiente di cui fanno parte e su cui poggiano, sia esso sociale, economico, politico, bellico, nucleare o di altro genere.

“ La sicurezza non è un prodotto, ma una catena forte quanto il suo anello più debole. ”

Bruce Schneier

Se paragonassimo quindi, un sistema informatico e la sua sicurezza ad una catena, notiamo che tutta la resistenza si focalizzerà sull’anello più debole e per questo, riallacciandomi al proverbio giapponese, c’è bisogno di lavorare e considerare ogni aspetto di questi sistemi per far sì che quella debolezza sia distribuita su tutta la catena poiché non sarà mai possibile garantire sicurezza al 100%.

Con queste poche righe è stata fatta una buona raccolta di informazioni che possono introdurre ad alcuni semplici concetti sull’argomento di tesi che ho deciso di trattare, il Threat Modeling. E’ importante fare attenzione sul fatto che non si tratta di un antivirus o di un firewall, ma è appunto un processo che il sistema acquisisce, e quindi integra al suo interno, rendendolo parte attiva dell’intero ecosistema informatico, al pari di concetti più ampi come protocollo di comunicazione o sistema operativo, macro-argomenti che inglobano diversi concetti, funzionalità, modalità operative, e non il singolo strumento, o nel caso della sicurezza, il singolo antivirus. In altre parole si tratta di aggiungere ad un sistema modulare, come quello informatico, un apposito modulo per l’ampliamento delle capacità protettive necessarie, non per la singola minaccia, ma per la definizione generale di minaccia, e quindi a salvaguardia di qualsiasi cosa possa considerarsi pericoloso.

Indice

Elenco delle figure

Elenco delle tabelle

Acronimi

SDLC Software Development Life Cycle. 1, 8, 14,

DFD Data Flow Diagram. 10, 17.

STRIDE Spoofing Tampering Repudiation InformationDisclosure DenialOfService ElevationOfPrivilege. 13, 17,

PASTA Process for Attack Simulation and Threat Analysis. 13.

OCTAVE Operational Critical Threat, Asset and Vulnerability Evualtion. 14.

DML Detection Maturity Level. 15.

TTP Tactics Techniques and Procedures. 16, 21.

DREAD Damage Reproducibility Exploitability AffectedUsers Discoberability. 16.

UML Unified Modeling Language. 17-18.

XML eXtensible Markup Language. 21, 22, 36, 49-50, 59.

IOC Indicator of Compromise. 20, 21, 29, 60.

IPS Intrusion Prevention System. 30, 41, 60.

IDS Intrusion Detection System. 30, 41, 60.

CERT Computer Emergency Response Team. 21.

CSIRT Computer Security Incident Response Team. 21.

MISP Malware Information Sharing Platform. 20.

STIX Structured Threat Information Expression. 20-21, 79.

TAXII Trusted Automated eXchange of Indicator Information. 21.

TLP Traffic Light Protocol. 21.

API Application Program Interface. 21, 35-36, 44, 49, 51, 55-56, 59, 62, 67-68, 72, 74, 79, 83, 86-87, 90.

SSH Secure SHell. 36, 41, 77, 92.

SIEM Security Information and Event Management. 21.

CVSS Common Vulnerability Scoring System. 11, 17.

CWE CommonWeakness Enumeration. 11.

JSON JavaScript Object Notation. 22, 33, 44, 49-52, 55, 57, 59, 89-90.

Capitolo I

1. INTRODUZIONE

1.1 Analisi dello scenario

Un software viene valutato in base a qualità esterne come correttezza, affidabilità, efficienza, usabilità, robustezza, ecc., tutte percepibili dall'esterno senza sapere come è strutturato, e qualità interne come riusabilità, verificabilità e manutenzione, tutte particolarmente legate alla fase di sviluppo del sistema. Tra le qualità interne ed esterne abbiamo anche la sicurezza, che deve comportare il rispetto di alcune caratteristiche come prevenzione, tracciabilità, anonimato, autenticazione, integrità, confidenzialità, privatezza, monitoraggio. E' difficile rispondere alla domanda "Quando e quanto un programma può ritenersi sicuro?". Inoltre, il significato della parola sicurezza è relativo al contesto in cui stiamo ragionando. Proteggere un sistema non vuol dire inserire tecnologie casualmente. L'analisi sulla sicurezza è necessaria per bilanciare sicurezza e funzionalità, e andrebbe fatta ad ogni fase dello sviluppo poiché si possono incontrare minacce ad ogni fase. Esiste quindi la necessità di integrare elementi di sicurezza in ogni step del ciclo di vita del sistema, costituendo un modello di sviluppo sicuro del software (SSDLC[3]). Di fatto, ogni azienda poi personalizza il proprio modello, potendo incontrare diverse versioni di sviluppo software sicuro come BSIMM[4], OWASP SAMM[5], Microsoft SDL[6][2].

[3] Secure Software Development Life Cycle, particolare modello di sviluppo software incentrato sul raggiungimento di qualità di sicurezza.
[4] Buiding Security in Maturity Model è un software che valuta le strategie di sicurezza esistenti per creare modelli di sicurezza innovativi.
[5] Software Assurance Maturity Model è un open-project dell'organizzazione OWASP per implementare strategie di sicurezza nel sistema. https://www.owasp.org/index.php/OWASP_SAMM_Project
[6] Security Development Lifecycle è un software di casa Microsoft che adotta un modello a spirale per ridurre i costi e le vulnerabilità all'interno delle fasi di sviluppo del software

Aggiornare un normale modello di sviluppo software per consentirgli di raggiungere obiettivi di sicurezza consiste nell'introdurre le seguenti fasi:

- Ottenere requisiti di sicurezza, categorizzabili in:
    - Funzionali, cosa vorremmo NON accadesse
    - Non-Funzionali, derivati da principi architetturali, come robustezza e scalabilità
    - Derivati, combinazione dei primi due
    - User stories
    - Casi d'abuso
- Progettare il livello di sicurezza

  Tutte le parti interessate, come sviluppatori e designer, vengono coinvolti cercando di produrre una rappresentazione del sistema, basandosi sui requisiti di sicurezza raccolti.
- Sviluppare la sicurezza

  Implementare codice
- Testare la sicurezza

  Il testing per un processo di messa in sicurezza è differente dal testing che accorre al normale ciclo di vita di un software, poiché qui dovremo essere in grado di assumere realmente le parti di un attaccante. Alcune metodologie di testing possono essere:
    - Penetration Testing[7]
    - Fuzz Testing[8]
    - Fault Injection[9]

Viene fatta questa introduzione per spiegare che l'argomento del threat modeling tocca ognuna di queste fasi, considerando questo processo non solo come parte del ciclo di vita di un sistema, ma diventando un sinonimo dello sviluppo di software sicuro.

Il processo di Threat model permette di andare ad analizzare le debolezze che un sistema informatico può presentare, cioè le vulnerabilità. Quest'analisi porta ad un argomentazione di tali minacce, permettendo di arginare le lacune adottando delle contromisure. La conoscenza

[7] Processo di simulazione di minacce ed attacchi per la valutazione della sicurezza del sistema e monitoraggio dei cambiamenti
[8] Tecnica di testing per individuare le falle e le vulnerabilità attraverso l'introduzione nel sistema di input casuali
[9] Spesso utilizzato come stress test, individua attraverso l'introduzione volontaria di errori nel codice

delle principali tecniche d'attacco, degli obiettivi più appetibili, dei diversi profili che un attaccante può avere, e rispondendo alle domande che il processo del Threat modeling porta a galla, favorisce l'analisi del sistema sotto l'aspetto della sicurezza:

Cos'è effettivamente un threat model?

In un momento storico e tecnologico in cui la sicurezza diventa sempre più un "must-have" si è capito il bisogno di correre ai ripari dalle conseguenze che può portare la presenza di debolezze in un sistema, cercando di adattare la struttura che lo regge a esigenze di carattere protettivo. Il processo con cui si ottiene questo risultato è anche chiamato threat model, ovvero modellamento delle minacce. In altre parole si tratta di aggiungere ad un sistema modulare, come quello informatico, un apposito modulo per l'ampliamento delle capacità protettive a salvaguardia da azioni pericolose.

Protezione da chi e perché?

Chiunque produce e utilizza informazioni, personali e non, per far parte di questo mondo virtuale che ha preso il sopravvento sulle vecchie abitudini. Spesso questi valori sono presi di mira da persone cui hanno lo scopo di sfruttare vulnerabilità per poter raggiungere un determinato obiettivo di carattere economico o spesso associato a valori morali o situazioni il cui quadro risulta molto più ampio di quello a cui danno ad apparire.

Da cosa è caratterizzato?

Un threat model non è altro che un semplice processo che prende in input il sistema e dopo un'attenta elaborazione ottiene la spunta a tutte le caratteristiche di sicurezza che richiede: confidenzialità, integrità, anonimato, ecc. Per rispondere bisogna prendere la fase centrale di questo processo, l'elaborazione dell'input, e descriverla: è caratterizzato da un forte lavoro di analisi e prevenzione.

Perché viene utilizzato?

E' fondamentale quantificare la portata di un attacco ad un sistema informatico affinché questo dia la migliore idea all'utente principale di cosa sta andando incontro! Furti di dati, informazioni personali, password mal gestite, quantificano danni per miliardi di dollari ogni anno, ed è stato verificato che la gran parte di queste minacce è portata a buon termine per il semplice motivo che non sono state prese delle misure di sicurezza nell'ordine di un semplice antivirus. Il threat modeling si pone l'obiettivo di prevenire

tutte queste problematiche andando ad inserirsi, possibilmente, in una fase di sviluppo del sistema, dove risulta essere più efficiente.

Che benefici porta?

"Prevenire è meglio che curare". La prevenzione come forma di contrasto per l'arginamento delle problematiche di sicurezza, permette soprattutto un risparmio sui costi. Avere un piano d'emergenza permette di ridurre drasticamente i danni che si potrebbero subire, cosi come avere un piano di ripristino permette di ridurre i tempi in cui il sistema non può lavorare.

Ha un costo?

Comporta del lavoro molto accurato, da cui deriverà la qualità della sicurezza nel sistema. Non si può garantire sicurezza in una forma perfetta ne concentrare tutti gli sforzi del sistema per garantirla. Per cui bisogna bilanciare le energie per ottenere il miglior rapporto d'efficienza computazionale.

Come si costruisce un processo di modellamento delle minacce?

Fondamentalmente si poggia su 3 fasi da cui otterrà materiale per impostare il miglior profilo di sicurezza possibile per le necessità dell'utente e del sistema: caratterizzare il sistema su cui si andrà a lavorare, individuare ciò che può renderlo vulnerabile e adottare le contromisure necessarie ad arginare queste debolezze.

Se e cosa può andare storto?

La sicurezza è un concetto "context-sensitive", quindi dipende molto dai casi, ed è difficile descriverla nella sua forma perfetta. Tenendo conto del periodo storico molto importante per la sicurezza informatica, dove le nuove minacce sono all'ordine del giorno, il bisogno di essere sempre aggiornati è un obbligo.

E' possibile migliorare ulteriormente questo processo?

L'ambito è ormai affrontato da molte aziende che conducono sperimentazioni cercando di introdurre migliorie. La tesi poggerà proprio su questo lavoro: cercare di condurre un'indagine per individuare il contesto storico, lo stato dell'arte, e le problematiche più diffuse, per cercare di avanzare delle argomentazioni.

Come valutiamo se stiamo proseguendo lungo la strada corretta?

Occorre considerare il bilanciamento tra risultato e lo sforzo richiesto. L'aggiunta giornaliera di nuove minacce all'arsenale di un attaccante permette poi di validare il lavoro fatto per una finestra temporale davvero breve. La conseguenza di tutto ciò è quella di rendere questo processo un vero e proprio componente del sistema, con meccanismi iterativi.

Quanto può essere efficace?

Secondo uno studio condotto da IBM, adottare un modello per le minacce già in fase di design del sistema può essere 7 volte più conveniente che farlo in fase di implementazione, e fino a 100 volte che farlo in fase di produzione del sistema[3].

1.2 Struttura dell'elaborato

La struttura dell'elaborato seguirà poi questo percorso:

- *Capitolo 2*: sviluppo dello stato dell'arte attraverso una sequenza di passi che mostrano l'evoluzione del threat model; panoramica sulla forma generale del processo di modeling precedentemente all'illustrazione delle varie metodologie e tecniche presenti negli ultimi 15 anni; focus sulle novità introdotte dagli ultimi studi tra cui l'applicazione del machine learning e approdo ad un approccio ibrido.
- *Capitolo 3:* considerazioni sugli strumenti incontrati nelle varie casistiche presenti nel materiale raccolto, e valutazione di questi ultimi per le varie fasi che compongono il processo di modellamento; introduzione alla tipologia di tool e alla loro categorizzazione con confronti e riscontri per la scelta di alcuni su cui testarne le potenzialità; scelta di diversi tool per le varie aree individuate con illustrazione dei meccanismi.
- *Capitolo 4:* considerazioni finali sul tema della sicurezza, del threat model e quindi del lavoro di tesi. Cenni sulle prospettive future riguardanti il processo di individuamento delle minacce e le fasi che lo compongono.

Capitolo II

2. STATO DELL'ARTE

Il capitolo individua lo sviluppo dello stato dell'arte seguendo un percorso che mostra come il threat model si sia evoluto negli ultimi 20-30 anni. Sarà illustrata una panoramica sulla forma generale del processo di modeling. Seguirà un percorso di individuamento delle varie metodologie e tecniche, cercando di introdurre un focus sulle novità degli ultimi studi tra cui l'applicazione del machine learning e approdo ad un approccio ibrido.

2.1 Primi passi ed evoluzione

La ricerca di vulnerabilità nei sistemi informatici, per carattere sperimentale, etico o per un ritorno economico, ha favorito sempre più l'analisi e quindi lo studio preventivo di Come tali minacce potessero manifestarsi, Dove, Quando e Perché, alcune delle principali domande del Threath Model Process. Da questa base di partenza è possibile proseguire sulla linea temporale identificando dei momenti in cui il Threat Modeling ha fatto passi avanti, incontrando soprattutto nuove tecniche da aggiungere al proprio "arsenale di conoscenze".

2.1.1 Threat and Attack Trees

Partendo dal concetto di Architectural pattern[10] introdotto nel 1977 da Christopher Alexander, e adattato per la prima volta ad un IT system da Robert Bernard nel 1988, si arriva a cavallo del 1994, dove viaggiano su due strade parallele, ma indipendenti, le rappresentazioni tramite grafico ad albero di due concetti relativamente collegati: da una parte Edward Amoroso[11],

[10] Concetto architettonico introdotto nel libro *A pattern language* e che ha permesso di indirizzare i paradigmi e i linguaggi di programmazione verso il concetto di programmazione ad oggetti.
[11] https://www.tag-cyber.com/people/eamoroso

all'epoca professore di informatica presso lo Stevens Institute of Technology, mette in mostra nel suo libro "Fundamentals of Computer Security Technology" come può avvenire la scoperta di una minaccia utilizzando un diagramma ad albero, un threat tree; dall'altra parte lo stesso lavoro condotto dall'NSA[12], DARPA[13], e presente nell'articolo del '98 di Bruce Schneier "Toward a Secure System Engineering"[4], mostrava il percorso d'esecuzione di un attacco ad un sistema informatico, utilizzando sì, la stessa soluzione grafica, ma proponendo un albero degli attacchi (anziché delle minacce) , posizionando alla radice l'obiettivo dell'attaccante.

Portando avanti questi controlli e analisi di sicurezza e dei rischi legati ad un'ambiente IT, si evidenzia come l'obiettivo di un'infrastruttura per le informazioni sia quello di essere preparata a proteggersi partendo dalla conoscenza di vulnerabilità, contromisure, e tutto ciò che riguarda l'individuazione di una minaccia. Bisogna individuare le debolezze nel sistema per permettere all'amministratore di mettere in atto processi di messa in sicurezza, progettando delle corrette strategie e tenendo soprattutto conto di come si muove l'avversario. La conoscenza del sistema dovrà essere mappata in modo da evidenziare i suoi punti focali, come il flusso dei dati, le componenti, le fasi di produzione, utilizzo, ecc., ed individuarne quali vulnerabilità potrebbero avere per poter adottare delle contromisure (se possibili in relazione ai costi, compatibilità e performance).

C'è bisogno innanzitutto di categorizzare gli attaccanti in base all'arsenale che li accompagna in termini di risorse, obiettivi, tolleranza sul rischio, andando ad individuare cosi alcune figure come hacker, crimine organizzato, competitori industriali, servizi di intelligence nazionali, che avranno tutti obiettivi diversi, ma solo uno in comune: scegliere il percorso con meno ostacoli e che permetta di ottenere il massimo risultato. Ogni attacco poi avrà bisogno di quantità diverse dell'arsenale dell'attaccante, ad esempio un attacco forza bruta richiederà molta potenza computazionale, ma accessi minimi e rischi nulli.

Il compito del difensore è quello di costruire un muro sul "Terreno delle Vulnerabilità" in cui una valle rappresenta una vulnerabilità, e una cima una contromisura. Non basta costruire una torre! Man mano che poi si aggiungono informazioni o si aggiorna il sistema non si deve ripartire da zero. Bloccare un attacco vuol dire bloccare almeno uno dei i 3 step che lo compone: analizzare il sistema, ottenere gli accessi necessari, sferrare l'attacco. Considerando l'SDLC delle componenti di un sistema e ponendosi domande a riguardo, è possibile trovare sulla mappa

[12] National Security Agency
[13] Dipartimento della Difesa degli Stati Uniti per lo sviluppo di nuove tecnologie per uso militare.

le opportunità d'attacco, come un bug in un sistema o un trojan distribuito in una rete o una mail spam. Si arriva cosi a caratterizzare un attacco e le relative contromisure adottando un diagramma ad albero in cui andiamo ad analizzare i pesi e i percorsi di ogni attacco.

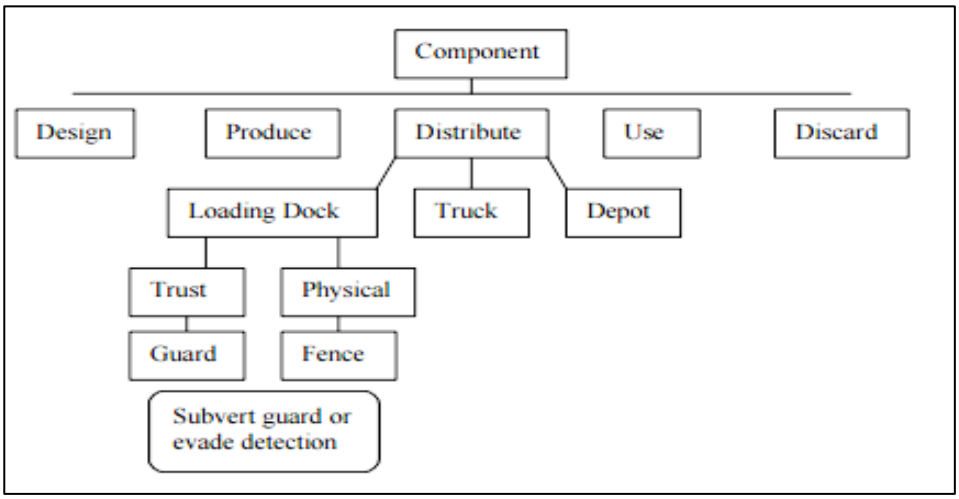


Figura 2.1 – Caratterizzazione d'attacchi

Utilizzare l'albero degli attacchi fornisce un set limitato di attacchi, quindi efficiente soprattutto per piccoli scenari. Inoltre partendo dall'idea di attack trees si sono poi sviluppate diverse evoluzioni[3]:

- Attack suits, miglioria apportata agli attack tree usando algebra semantica che rappresenta gli attacchi come set di componenti. Complesso, quindi non molto utilizzato;
- Attack nets, combina dei set di falle per ottenere attacchi sofisticati. Utile per la costruzione di scenari d'attacco;
- Mitigation tree, l'opposto dell'attack tree, costruito in modo costruttivo per estinguere le minacce, e non distruttivo.

2.2 Processo generale per il threat model

Sinteticamente un modello delle minacce non è altro che una rappresentazione strutturata delle informazioni di un'organizzazione sotto la lente della sicurezza.

Il processo generale di threat modeling si compone di 3 macro fasi, divisibili in sottocategorie:

1) Caratterizzazione del sistema

1.1 Valutazione del sistema e settaggio degli obiettivi chiave di sicurezza: confidenzialità, integrità, disponibilità, autenticazione, autorizzazione, ripudio. Considerazione di risorse come database, informazioni sensibili, funzionalità, analizzando le necessità dell'organizzazione, le politiche di sicurezza. E' d'aiuto se la valutazione avviene in termini di impedimenti, cioè chiedersi "Cosa vogliamo che NON accada"?. La fase 1 permette di razionalizzare gli sforzi e di capire gli obiettivi dell'attaccante.

1.2 Rappresentare il sistema, ottenendo (con dei diagrammi) le relative tecnologie utilizzate, meccanismi di sicurezza presenti, autenticazione, autorizzazione, scenari chiave, ricordando sempre che il fatto che non si trovi tutto immediatamente non deve essere d'intralcio. Si dovrà illustrare la topologia di rete, fisica e logica, i diversi livelli logici come il livello d'accesso ai dati, i processi importanti, i protocolli e le porte utilizzate, i collegamenti con risorse esterne, i privilegi che dovranno avere le tipologie d'utenti che fanno parte del sistema, e tutti i dettagli sui meccanismi di sicurezza e sulle tecnologie come SO, server, linguaggi utilizzati. Ogni processo, ogni funzione potrà presentare queste capacità: Creare, Leggere, Aggiornare, Cancellare. Questo aiuta nel definire sia lo scenario sia ad estrarne le caratteristiche principali.

I diversi approcci utilizzabili sono:

- tramite l'utilizzo dei DFD[14], considerando le minacce che possono essere associate ad ogni simbologia con cui si configura il sistema: flussi, processi, confini, utenti, database. Questa rappresentazione però non è adatta per quei sistemi complessi dove è impossibile riuscire a rappresentare tutto.
- tramite il diagramma del flusso dei processi, che facilitano il processo di identificazione della minaccia e di come affrontarla, scomponendo il sistema considerando i suoi casi d'uso e le caratteristiche, per poi connettere tutto con i protocolli di comunicazione utilizzati.

[14] Diagrammi di flusso dei dati

2) Identificazione delle minacce e delle vulnerabilità:

2.1 Identificare le minacce, occorre riunire team di sviluppo, professionisti della sicurezza, tester e amministratori di sistema. Fondamentalmente è possibile considerare come punto di partenza:

- Partire dalle minacce comuni[15] ed applicarle al sistema per vedere velocemente se ci sono delle corrispondenze
- Porsi delle domande in base alle categorie degli attacchi, agli obiettivi di un attaccante, ecc.:

  - Come può un attaccante ottenere privilegi più alti?
  - Come può un attaccante ottenere l'accesso ai file di configurazione o eseguire funzioni riservate all'amministratore di sistema?
  - Può un attaccante utilizzare un input insolito per causare danni al sistema o lanciare un attacco SQL o XSS[16]?
  - Come può un attaccante causare errori nel sistema e che dettagli può ottenere da questi?

- Usare attack trees, pattern d'attacchi, o altre metodologie per scendere più in profondità e andare a individuare minacce nascoste.

Seppur non rilevanti, è bene tener conto anche di quelle minacce "potenziali".

2.2 Trovare le vulnerabilità sulla base delle minacce.

Ecco alcuni esempi di accorgimenti da porsi per individuare vulnerabilità:

- Far passare dati importanti su un collegamento non-criptato
- Effettuare la validazione solo lato client
- Preferire negare l'accesso piuttosto che filtrare gli input
- Usare un'unica chiave per più tempo o utilizzare una crittografia personalizzata
- Includere dati sensibili in cookie, query o campi che non utilizzano crittografia

[15] Ottimi esempi di raccolta di minacce comuni e di metodi generali possono essere i Cheat Sheet disponibili sul sito della Microsoft, https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649461(v=pandp.10) , o anche considerando CWE, CVSS

[16] Tipologie di attacchi come l'SQL Injection e il Cross Site Scripting permettono di sfruttare minacce atte a penetrare un sistema attraverso debolezze nel controllo degli input

3) Risultato

3.1 Documentare e classificare le minacce.

Threat Description	**Attacker obtains authentication credentials by monitoring the network**
Threat target	Web application user authentication process
Risk	
Attack techniques	Use of network monitoring software
Countermeasures	Use SSL to provide encrypted channel

Figura 2.2 – Individuamento e documentazione di una minaccia

In base alla valutazione che gli sarà attribuita[17] si saprà quale minaccia andrà gestita prima perché più importante, lasciando in coda quelle "trascurabili" o che non hanno un'alta priorità, quindi sviluppare un piano di pronto intervento alle minacce per ridurre al minimo il rischio.

Fattori da considerare:

- *Riduzione del rischio*, una buona analisi della tipologia di sistema su cui lavorare e sulla metodologia di modellazione scelta permette di fare delle distinzioni sulle vulnerabilità che si potrebbero incontrare e avere maggiori informazioni sulle minacce identificate, sulla loro priorità, e su come inibirle indirizzando al meglio il lavoro. E' raro che un sistema abbia bisogno di provvedere a tutte le minacce.
- *Auto-apprendimento*, l'importanza del processo non è quella di soffermarsi ad analizzare "indeterminatamente" ogni fase, ma far si che ogni fase produca un risultato, per poi andare a iterare il processo. Continuare a chiedersi "Ma se…?" oppure "Però potrei…".
- *Precisione*, focalizzare attentamente lo scenario, il tipo d'applicazione, ruoli, firewall, politica d'azienda, protocolli, porte attive, permette di veicolare meglio le informazioni necessarie.
- *Perimetro*, segnalarli aiuta nella definizione di obiettivi, vincoli, quindi cosa può o non può accadere.

[17] La valutazione è permessa da metodologie che saranno presentate e affrontate nel corso dell'elaborato, come DREAD model o valutazioni basate su formati comuni di indicizzazione delle minacce come CWE

- *Flussi*, definire le regole per i punti d'ingresso e d'uscita.
- *Gathering*, usare strumenti per la raccolta di dati sugli allarmi e gli eventi è utile per trovare minacce ad alta probabilità o impatto.
- *Collaborazione*, il processo deve permettere anche una funzione di condivisione delle informazioni raccolte.
- *Bugfix*, uno dei risultati del modello dovrebbe essere quello di evidenziare i punti chiave dove dovrebbero andare ad intervenire gli amministratori di sistema.

2.3 Metodologie per il threat model

> " Una minaccia è un pericolo che sfruttando una vulnerabilità può invalidare le misure di sicurezza e fare danni. " [5]

Le metodologie moderne per il processo di individuazione delle minacce si caratterizzano da diversi punti di vista, e seguono 3 approcci[5]:

- Software centric threat modeling, qui i diagrammi sia di flusso sia dei casi d'uso descrivono la struttura del sistema;
- Asset centric, classificando gli asset in base alla sensibilità dei dati e al valore dell'asset per un attaccante, aiutandosi nell'individuazione con alberi degli attacchi o grafici. Un esempio sono i tool Trike, Amenaza SecurITree[18];
- Attacker centric, capire dagli obiettivi come sarà l'attacco e come fermarlo.

Le metodologie più conosciute sono:

- STRIDE

  La metodologia Microsoft. Con il giudizio di professionisti della sicurezza, permette di capire quali attacchi, presi da un determinato set, possono essere utilizzati su ogni foglia dell'attack tree. Vennero categorizzati una serie d'attacchi riconducibili alle varie vulnerabilità presenti nei sistemi informatici, dando vita al threat model STRIDE che indicava appunto, la tecnica dello spoofing, perdita d'integrità, ripudio, informazioni visibili, DoS e l'elevazione dei privilegi.

[18] https://www.amenaza.com/SS-what_is.php

- P.A.S.T.A. (Process for Attack Simulation and Threat Analysis)

  E' un percorso composto da 7 step che vuole raggiungere un'identificazione delle minacce in maniera dinamica, permettendone una conseguente valutazione. Si scompone il sistema con un diagramma. La strategia di "riparazione" si basa quindi sul punto di vista dell'attaccante. (asset centric)
- Trike

  Un tool che permette la gestione della sicurezza attraverso l'utilizzo di threat model basati su dei requisiti come asset, ruoli, rischi d'esposizione e che identificano il livello di rischio "accettabile". Un suo vantaggio è l'alto livello di automazione, ma è ancora in fase di sviluppo quindi non ancora pronto. Un esempio di utilizzo è spreadsheet v2[19].
- VAST (Visual Agile Simple Threat modeling)

  Porta a scalare l'intero processo di modellamento sull'intero SDLC per poi lavorare con il software di sviluppo Agile che mostra un unico schema, non richiedendo specifiche competenze di sicurezza.
- OCTAVE (Operational Critical Threat, Asset and Vulnerability Evaluation[6])

  Metodo che permette di prendere decisioni sulla protezione delle informazioni valutando il rischio per l'integrità, disponibilità e confidenzialità. Nello scenario di OCTAVE si valutano categorie di minacce e obiettivi comuni, ad esempio se si considerano attaccanti che usano un accesso da rete, si dovrà valutare in base alle vulnerabilità dei collegamenti. Il risultato sarà un diagramma ad albero costruito in base alle caratteristiche delle minacce prima elencate. Se lo scenario che si configura corrisponde alla categoria degli accessi abusivi dalla rete, l'albero avrà 2 tipologie dell'attaccante, interno o esterno all'organizzazione.

[19] http://www.octotrike.org/tools.shtml

Si possono individuare 3 fasi:

1) L'organizzazione identifica le risorse fondamentali, le vulnerabilità associate e quindi le misure di sicurezza necessarie, i requisiti di sicurezza, meccanismi di sicurezza attuali;

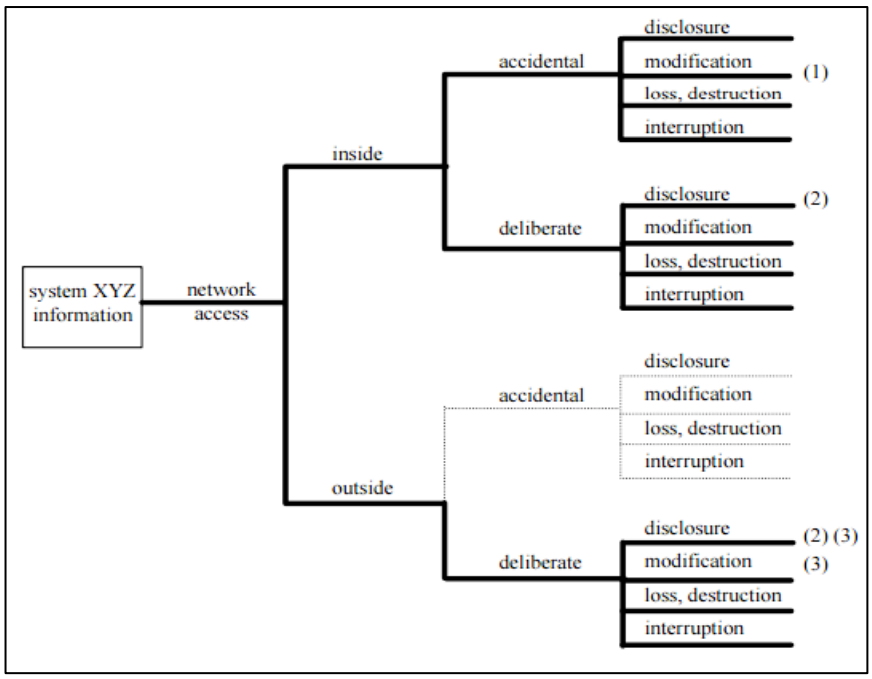


Figura 2.3 Albero delle minacce per un intervento d'accesso alla rete

Nell'esempio in figura, partendo dalla considerazione che una persona all'interno potrebbe anche distruggere e/o modificare le informazioni, è stato possibile intuire che anche per gli altri rami era lo stesso.

2) Dalla 1° fase si può dedurre i componenti chiave del sistema e valutare cosa può permettere azioni pericolose.

3) Vengono decise delle strategie per garantire un alto profilo di sicurezza andando prima a valutare il rischio di ogni ramo (di solito osservando cosa causano, perdita, danneggiamento, ecc.) e poi cercando di ridurre questi rischi con le dovute contromisure.

- DML

  Recenti studi condotti da Ryan Stillions, rappresentano la minaccia in base al livello di stratificazione che aveva raggiunto, presentando il modello DML cui fa riferimento a dei principi chiave:

  - un sistema è maturo quando ha ottime capacità nella detenzione e nella risposta alle minacce. Una buona detenzione deve sempre essere pronta anche quando la prevenzione fallisce, poiché tralasciare o ignorare è pericoloso;
  - senza un sistema di detenzione non è possibile reagire.

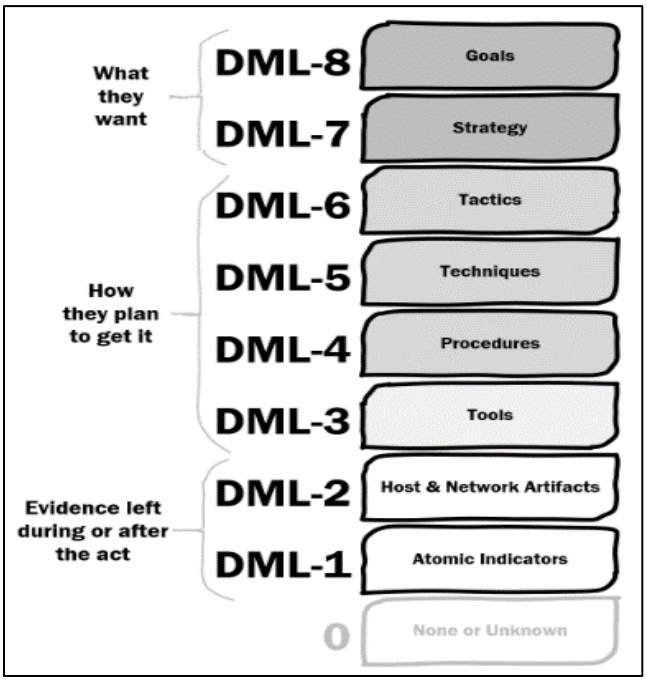


Figura 2.4 – Stratificazione del modello DML

Nella suddivisione in 9 livelli si nota nella parte più alta l'obiettivo e la strategia dell'attaccante, nel livello centrale le tecniche e le procedure, le TTP, mentre al livello più basso gli strumenti - i tools – utilizzati, insieme alle informazioni di rete come pacchetti e indirizzi IP.

- DREAD, si classificano i rischi in base a 2 metodi, "Low, Medium, High" oppure "Ordinary Ranking"; mentre le minacce hanno 5 categorie, da cui facendo la media ricaviamo la valutazione :
  - Danni potenziali, usato per valutare l'estensione dei danni se l'attacco ha successo
  - Riproducibilità, sforzo per riprodurre l'attacco

    - Sfruttamento, si valuta la facilità con cui si può riprodurre la minaccia
    - Utenti interessati, stimare i sistemi potenzialmente affetti nel caso di vulnerabilità vaste
    - Scoperta, valutazione per indicare quanto un attaccante abbia impiegato per individuare la vulnerabilità
- CVSS[20], usato per standardizzare e valutare, su un intervallo che va da 1 a 10, le vulnerabilità e i rischi. Composta da 3 metriche, Base – Temporale – Ambientale.
- T-MAP, calcola il peso dei vari percorsi dell'albero degli attacchi usando i diagrammi delle classi UML[21], degli accessi, delle vulnerabilità, asset. Un esempio è Tiramisu[22] che calcola un elenco di tutti i percorsi di attacco e genera minacce con i pesi dei percorsi di attacco.
- Fuzzy logica (grado di verità), MATLAB Fuzzy tool[23] dopo aver usato un modello STRIDE usa un fuzzy interference engine[24] per riportare una lista di minacce.
- LINDDUN, approccio basato sulla privacy, simile a STRIDE, ma comprendenti:
    - Collegabilità, se due elementi sono collegabili;
    - Identificabilità
    - Non-ripudio
    - Rilevabilità
    - Divulgazione di informazioni: con gli accessi non autorizzati alle informazioni personali
    - Inconsapevolezza, quando non si conoscono le informazioni condivise con il sistema.
    - Non conformità, alle leggi e alle politiche

  Composto da 3 fasi: modellare il sistema in base allo scenario come DFD, successivamente mappato nelle categorie prima elencate

[20] Common Vulnerability Scoring System, utilizzato per caratterizzare ogni vulnerabilità e la sua pericolosità
[21] Unified modeling language, linguaggio semi-formale per modellazione e specifica orientato agli oggetti. Lo si ritrova in molti contesti grazie alla semplicità di rappresentazione per i diversi approcci metodologici attraverso diagrammi.
[22] http://csse.usc.edu/csse/research/COTS_Security/index.html
[23] https://it.mathworks.com/products/fuzzy-logic.html?requestedDomain=
[24] Sistema software o hardware utilizzato per un controllo logico fuzzy (casuale), che produce un output o "conclusione" da un fatto (input) e da conoscenze (regole di controllo). Queste ultime devono includere termini linguistici fuzzy. Ci si riferisce spesso come *ragionamento approssimato*.

- Approcci per il cloud:
    - CORAS, metodologia basata su UML e composta da 7 fasi
    - Cloud privacy threat modeling, per la tutela della privacy nel cloud, uso principale nella diretta per Data Protection dell'UE.

- Per le Reti personali, principalmente per telecomunicazioni, transazioni finanziarie, utilizzano alberi d'attacco e diagrammi UML, composta da 7 step che comprendono la valutazione di minacce e vulnerabilità
- Analisi pratica delle minacce, ci sono 4 fasi in cui identifico gli asset e il loro valore, poi le minacce e le contromisure, e infine sulla base delle vulnerabilità e dei potenziali danni, preparerò un piano di intervento.

Queste sono solo le metodologie principali, ma in realtà le metodologie son molto più numerose, sia per i continui risultati ottenuti dall'innovazione nel campo - fornendo la facoltà di scelta sulla metodologia da adottare - sia per l'ampia scelta di possibilità messe a disposizione da un fattore come la personalizzazione, che ha fatto sì che le più grandi aziende dedicassero molte risorse affinché il modello sviluppato fosse quello con più benefici per l'organizzazione. Anche in ambito open-source, dove numerosi sviluppatori hanno testato soluzioni ibride e modulari.

2.4 Componenti esterne al processo

Al di là del processo generale e delle contromisure da adottare, il threat modeling ingloba anche componenti esterne che aggiungono efficienza e funzionalità al processo stesso. Fondamentalmente non si sta parlando di un metodo modulare, ma con il passare degli anni ci si indirizza verso questa strada. Gli articoli accademici e le riviste scientifiche presentano nuove problematiche che portano allo sviluppo di nuovi scenari in cui c'è bisogno di adottare un processo sperimentale, per arrivare a trarre delle conclusioni che possano migliorare il processo di messa in sicurezza di un sistema. La base di partenza è una scomposizione capillare del processo in modo tale da osservare minuziosamente i comportamenti che assume il sistema e il processo in ogni passo.

2.4.1 Logica-Apprendimento

Finora si è sempre parlato delle minacce senza focalizzarsi su un aspetto che inizialmente abbiamo introdotto, l'innovazione. Minaccia e sicurezza proseguono lungo un percorso, non inversamente proporzionale, ma parallelo. Come le tecnologie si adattano alle minacce che evolvono, cosi ci sono sempre più vulnerabilità che introducono nuove minacce. E' possibile subito distinguere come da quest'ultimo concetto ci si presentino 2 situazioni differenti: affrontare minacce comuni, che già conosciamo e sappiamo come trattare, e incontrare una minaccia non ancora diffusa, recente. Non avere già un metodo testato per arginarle porta a occupare una grande quantità di lavoro, sapendo che il nostro prodotto e le informazioni contenute al suo interno sono in pericolo. Inoltre non è detto che una minaccia presente nel nostro sistema voglia far danni in maniera plateale, ma restare ben nascosta e lavorare silenziosamente. Si prendi il caso delle minacce più recenti che distribuiscono il lavoro di crypto miner[25] su più pc proprio per non dare dell'occhio. Sarebbe proprio il caso di dire che non solo si potrebbero avere delle minacce sconosciute che influenzano il nostro sistema, ma che agiscono in incognito senza voler causare danni evidenti.

Riuscire a controllare ed estirpare anche quelle minacce che non sono comuni, aggiungerebbe un grado di sicurezza in più al nostro sistema. Questo è stato dimostrato possibile grazie all'introduzione di intelligenza nel processo di threat model che ci permette di adottare azioni mirate per questi tipi d'attacchi. La differenza rispetto al normale metodo d'individuazione di minacce già conosciute sta nell'obiettivo che si vuole raggiungere: usare un semplice threat model permette di agire attraverso strategie, quindi in maniera programmata, mentre un sistema intelligente permetterà di essere più veloce e reattivo. Una miglior rappresentazione del problema verrà data nel paragrafo sul *sistema multiagent*.

	Threat Modeling	**Threat Intelligence**
Intervento	Proattiva	Reattiva
Scopo	Ricerca Problemi	Ricerca l'attaccante

Tabella 2.1 – Caratteristiche per un modello intelligente

[25] Malware in grado di produrre criptovalute utilizzando una piccolissima parte della potenza computazionale dei dispositivi infettati. https://www.wired.it/economia/finanza/2018/03/09/miner-virus-bitcoin/?refresh_ce=

Oltre ad una questione d'intelligence, di Machine Learning e quindi di auto-apprendimento, ci si trova ad analizzare molte informazioni che si ricevono dal sistema durante il suo ciclo di vita. Diventa una grande difficoltà saper gestire grosse quantità d'informazioni raccolte per determinare se richiedono un intervento, questo non solo per un analista, ma anche per i sistemi tradizionali per l'individuazione di tali minacce. Uno studio in particolare mostra come si sia cercato di raggruppare gli allarmi in base a se avessero fattori in comune attraverso l'implementazione di un deep neural network classifier[7] per valutare se l'allarme andasse considerato direttamente oppure ignorato.

2.4.2 Pubblicazione – Canale per le minacce

In ambito di cyber threat intelligence la raccolta di informazioni circa minacce e attacchi cresce a intervalli giornalieri formando quei Big Data necessari per valutare soluzioni di sicurezza "istruite", ma di difficile carico per quelle organizzazioni che ancora tengono queste informazioni in ambienti non-condivisi. L'approfondimento sulla questione del condividere questi database di minacce cresce molto velocemente con nazioni a lavoro per standardizzare lo scambio tra governi e organizzazioni, ma non esistono molte soluzioni di scambio che permettano una lettura più semplice di questi dati.

La possibilità di avere disponibili informazioni di questo tipo prima ancora dell'attacco, è fondamentale! Attacchi, minacce, vulnerabilità diventano sempre più complesse e quindi è bene distinguere che l'organizzazione di questa tipologia di dati sia ben strutturata affinché sia possibile trarne un beneficio contro l'imminente minaccia.

L'obiettivo della modellazione delle minacce è proprio quello di anticipare la minaccia. Questi dati possono essere raccolti da fonti pubbliche o private, interne o esterne, ma la rilevanza sta nel beneficio della condivisione e quindi lavorare in gruppo.

"Collaborare anziché lavorarle indipendentemente."

Cosi gli approcci per lo scambio e la condivisione delle cyber threat information avvengono attraverso un mix di[8]:

- Piattaforme online, tra cui:

- MISP, focalizzata solo sulle informazioni di malware, permette di condividere le informazioni su attacchi e IOC[26], di tipo tecnico e non, salvandole in un formato strutturato che possa permettere l'esportazione anche in formato STIX;
- AbuseHelper, usata da CERT e ISP, quindi con una grande quantità di dati. Processa automaticamente una notifica d'incidente e le considera in maniera molto semplice, non adatta alla condivisione;
- IntelMQ, limitato a collezioni di dati quindi non adatto alla condivisione, utilizza una coda di messaggi per processare feeds, tweets, ecc;
- Cyber Threat XChange, componente di HITRUST C3[27], dedicato al settore della medicina;
- Open Threat Exchange, una rete creata per condividere informazioni anonimamente, analizzarle e condividerle con la community di AlienVault[28] e per questo non la miglior scelta per un modello di condivisione;
- Soltra
- Collaborative Research Into Threats, si appoggia a diversi open-tool per condividere gli incidenti. I dati vengono convertiti in oggetti CybOX[29], e poi inseriti in file STIX e condivisi con il metodo TAXII. È disponibile da remoto, localmente e attraverso librerie API.

- Utilizzo di protocolli di scambio:
  - STIX, linguaggio che permette la miglior rappresentazione delle informazioni e delle diverse tipologie: incidenti, vulnerabilità, IOC, TTP[30]. Estendibile, facile da capire per l'uomo. Personalizzabile per includere altre informazioni come TLP;
  - TAXII, protocollo per il trasporto di file STIX, con messaggi dedicati al servizio usando XML e HTTP;
  - IODEF, standard per CML, usato dai CSIRT[31];

[26] Indicator of Compromise, dal termine stesso è un indicatore che ci permette di segnalare una possibile variazione rispetto al normale funzionamento di un sistema di diversi fattori come indirizzi IP, URL o MD5 hash malevolo.
[27] Programma per la raccolta e condivisione di dati informatici indicanti problemi di sicurezza nell'ambito medico-ospedaliero.
[28] Software opensource utilizzato come SIEM per la gestione di attacchi informatici e threat modeling .
[29] Linguaggio strutturato per la raccolta e osservabilità di cyber-informazioni, eventi dinamici o misure statiche.
[30] Tactics, Techniques and Procedures è l'acronimo per l'identificazione di modelli di attività o metodi associati ad uno specifico attaccante o gruppo di attaccanti.
[31] Centro/Squadra di pronto intervento alle emergenze informatiche.

- CIF, permette di combinare informazioni per trovare minacce nei sistemi, contiene molti indirizzi IP e URL sospetti;
- TLP, trasporta informazioni solo al corretto destinatario, utilizza 4 colori per indicare il grado di sensibilità.

In assenza di alternative i dati possono essere gestiti attraverso formati come XML e JSON, linguaggi molto versatili ed efficienti per la gestione, la raccolta e il salvataggio di informazioni, incluse quelle appartenenti a Database.

2.5 Approccio ibrido

Seppur l'approccio al threat modeling per un'organizzazione permette di identificare debolezze preventivamente, questo avviene attraverso l'utilizzo di un singolo metodo, ricordando però che distintamente hanno diverse limitazioni. E' fondamentale non dimenticare che:

" Nel Processo di modellazione delle minacce individuiamo le minacce,
che possono essere causate dalla presenza di vulnerabilità,
che quindi possono essere realizzate con un attacco,
a cui possono essere adottate delle contromisure. " [9]

La realizzazione di un modello per le minacce efficiente deve seguire 3 aspetti:

- Approccio strutturato
- Dettagli ottimizzati
- Leggibilità

Negli ultimi anni il processo di modellamento si è evoluto ulteriormente arrivando allo stato attuale dove è preferito un approccio ibrido[9].

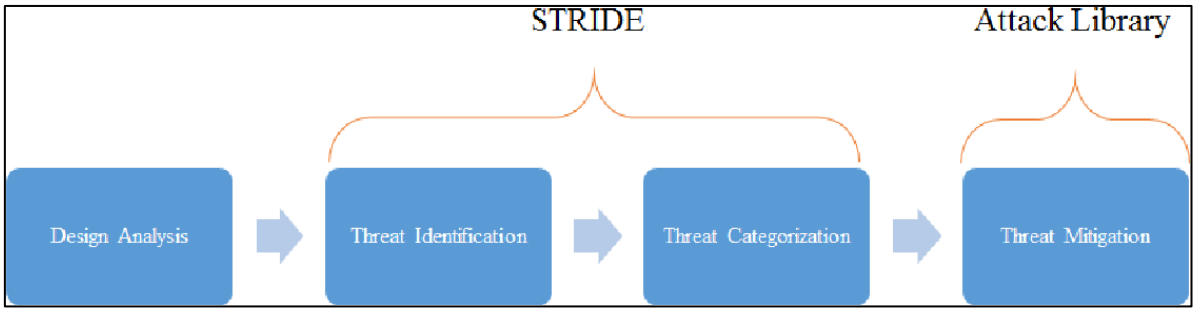


Figura 2.5 – Esempio di configurazione ibrida di un threat model

La figura 2.6 è solo un esempio di come l'unione di diversi approcci sia possibile per andare a perfezionare le diverse fasi del processo. D'altro canto la natura evolutiva e iterativa è già di per se un enorme passo in avanti verso una sicurezza maggiore, però come detto nell'introduzione di questo paragrafo, i diversi approcci presentano diverse lacune.

Per ogni Pro c'è un Contro, e per ogni Contro c'è un Pro. Parlare delle diverse lacune che affliggono i diversi metodi, non mette in evidenza i punti focali che possono essere utilizzati da ogni metodo per l'introduzione ad un approccio ibrido. Per fondere diversi meccanismi, in maniera molto semplice si va ad analizzare il processo stesso cercando di conciliare le caratteristiche delle varie fasi con le peculiarità dei diversi approcci.

Ognuno ha poi personalizzato la propria esperienza ibrida attraverso la valutazione personale assegnata ai diversi metodi e tool. Nella letteratura raccolta sono presenti diversi casi e articoli in cui i tool descritti vengono scelti per specifici motivi, atti ad effettuare un lavoro mirato per uno specifico campo.

2.6 Sistemi multiagent

La gestione dei rischi in un sistema si può dividere tra quelli di cui si ha conoscenza e quindi è possibile adottare efficacemente delle contromisure, e quelli di cui non si ha prevedibilità. Il modello proposto consisterà di 3 fasi[10]:

1) Identificare le minacce, quelle comuni e quelle non ancora individuabili attraverso un meccanismo che fonde modelli statistici e rilevamento di honeytokens, come password, record di database, ecc. che implicano un'attività sospetta. Gli honeytokens quindi si presentano come delle honeypot[32], componenti appetibili per un attaccante, ma con la differenza che sono entità digitali e che forniscono degli schemi d'attacco per le minacce non rilevate.

[32] Sistema "esca", ma ben isolato, situato in una rete allo scopo di essere esplicitamente attaccato per permettere di andare a raccogliere ed analizzare informazioni sugli eventi.

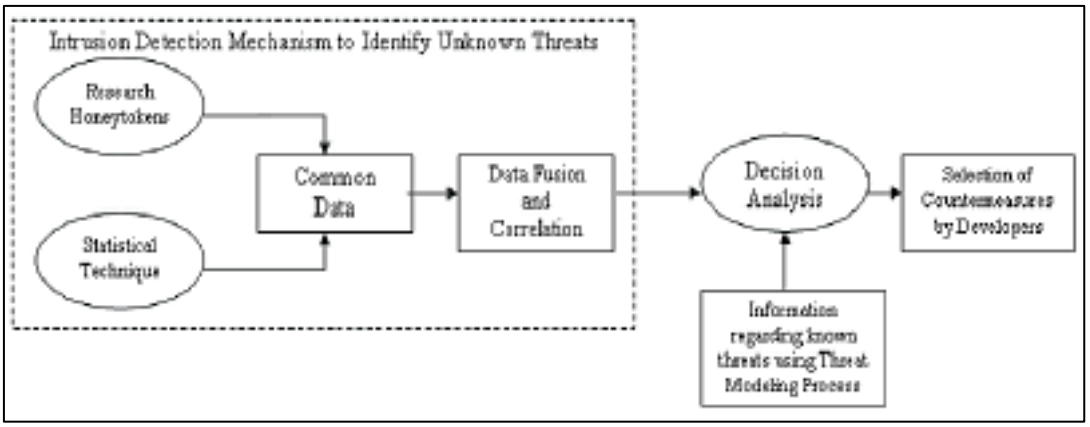


Figura 2.6 – Approccio proattivo per la gestione delle minacce

2) Neutralizzazione, le minacce trovate nella fase 1 vengono eliminate attraverso la suddivisione del lavoro tra più "agents" con MAPSTA. Qui viene creato un albero delle minacce per analizzare i percorsi con cui l'attacco si è manifestato.

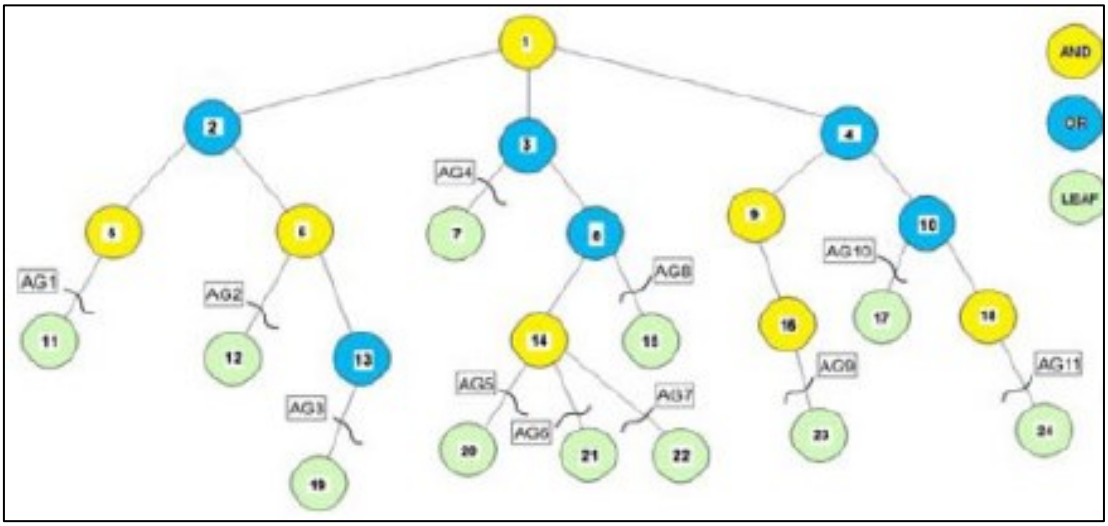


Figura 2.7 - Agents per il contrasto delle minacce

I nodi AND indicano che la minaccia attuale può venire da tutti i collegamenti del nodo, mentre i nodi OR che anche solo con un collegamento la minaccia è attiva. Gli agent allora vengono coinvolti ai nodi foglia non permettendo all'attacco di arrivare alla radice.

3) Verifica del rischio, qui viene valutato che le contromisure adottate abbiano effettivamente ridotto il rischio. Dei "meta-agents" controllano il risultato della fase 2,

altri valutano variazioni sul livello di sicurezza attraverso un "fuzzy interference system". Il loro vero compito sta nel passare dalla strategia alla decisione. Ad ogni modo il processo di valutazione passa da un ultimo meta agent di risposta che è colui che valuta il livello di sicurezza tramite i colori: verde se la sicurezza è OK; giallo se c'è il bisogno di aggiungere altre contromisure; rosso se il sistema non è sicuro.

Partendo dal modello a spirale di Boehm, si è passati ad una sua versione che basa il tutto sul rischio, quindi man mano che il processo viene iterato il rischio deve diminuire mentre l'affidabilità aumenta. Bisogna fare attenzione ad alcune limitazioni: il modello descritto è idoneo per lo sviluppo di sistemi su larga scala.

Capitolo III

3. VALUTAZIONE SUI TOOL

Nel capitolo verranno effettuate delle considerazioni e delle valutazioni sugli strumenti che gli articoli accademici e le riviste presentano come soluzione a diversi problemi e alle diverse fasi che compongono il processo di modellamento. La scelta dei tool si baserà sulla loro tipologia e sulla loro categorizzazione, andando ad individuare tramite confronti dei punti su cui poi si andrà a testarne le potenzialità. Queste scelte porteranno ad illustrare i meccanismi dei diversi tool per le diverse aree di utilizzo. Al termine del lavoro verranno raccolte le scelte che più potranno risultare efficienti.

3.1 Considerazioni

L'utilizzo di diversi strumenti permette di dare forma alla struttura del processo di modellamento delle minacce. La raccolta e lo studio di lavori affrontati sul tema e sulle problematiche che ha comportato e che comporta, hanno permesso di individuare un set di tool utilizzati per far fronte a diverse casistiche e a diversi approcci del threat model. In una fase successiva alla raccolta di questi tool è stato possibile valutarli e classificarli in base a diversi fattori, come l'adattamento a diverse funzioni, la categorizzazione circa i meccanismi adottati, la disponibilità di licenza open-source e di librerie dedicate.

3.1.1 Fasi del processo di modeling

L'elaborato ha focalizzato molto l'attenzione sulla divisione in fasi del processo di modellamento, utile per analizzare quali sono i passaggi che avvengono tra uno step e l'altro. Come visto nel paragrafo 2.5 e 2.6 per l'approccio ibrido e i sistemi multiagent, per

massimizzare l'output del threat modeling è fondamentale sfruttare l'efficienza di diversi strumenti individuabili per le diverse categorie di lavoro. Per tanto, un ulteriore processo d'individuazione e valutazione dei requisiti richiesti ad ogni step del threat model permetterà di associare dei tool che meglio si fondono con la categoria individuata.

Figura 3.1 - Schema divisione grafica threat model nelle diverse aree

3.2 Raccolta dati

Le valutazioni di sicurezza necessarie per le attività di individuamento minacce, necessitano di dati e informazioni su cui operare affinché l'analisi metta in risalto vulnerabilità nascoste e minacce emergenti. Si inquadra come input della fase di raccolta dati la necessità di dover raccogliere e gestire diverse sorgenti di dati provenienti dal proprio sistema, localizzate tramite un dettagliato lavoro di focus sui punti del sistema che producono e rendono disponibili tali informazioni. Non si sta parlando solo di file di log: bisogna affidarsi a diversi tipi di meccanismi che, oltre ad una raccolta di questi file già predisposti per rappresentare gli avvenimenti e lo stato del sistema o di un software, devono anche poter analizzare informazioni generali, ma consistenti, come indirizzi IP e valori di hash che introducono ad un ampliamento dei risultati sia sulla raccolta (più capillare) che sulla valutazione. In parole povere, secondo i primi passi del processo generale di modellamento delle minacce (paragrafo 2.2), la necessità di "mandare in input" l'intero sistema si traduce appunto nell'analisi di queste informazioni, siano esse prodotte dal sistema operativo, dai software o da altra sorgente. Questa categoria di lavoro introduce un concetto interessante: la caratteristica di focalizzare il sistema e permetterne un'analisi sia statica (file di log precedenti) che dinamica, trasforma un processo di modeling "basilare" aggiungendogli delle skin qualitative come la proattività alla caccia delle minacce, introducendo quello che in altri termini viene definita "intelligence", il non attendere il pericolo, ma avvertirlo in anticipo[11].

Bisogna fare un'attenta distinzione tra 2 tipologie di sorgenti di dati che vanno trattate diversamente: quelle con necessità di un processo di analisi, e quelle che svolgono un lavoro di analisi intrinsecamente al tool che le gestisce. Il primo caso appartiene alla particolare casistica dei file di log del sistema, i quali hanno necessità di essere analizzati per poter condurre indagini su pericoli emergenti. Per tali dati c'è la necessità di dover correlare diverse voci per tirar su un quadro più preciso di quello che è accaduto nel sistema. Il compito del sistema è quello di tenere traccia degli eventi in modo tale che ci possa essere una cronologia pronta allo studio di particolari situazioni insorte. Mentre, il particolare caso delle fonti di dati che non hanno bisogno di analisi si differenziano appunto perché trattano, internamente agli strumenti che le gestiscono, la parte di elaborazione ed analisi delle informazioni.

LOG, IOC, IDS, IPS

Conoscere la natura di un attacco o di una minaccia non basta. E' necessaria la capacità di conoscere le strutture e i fondamenti dei dati associati a questi attacchi. Tali informazioni sono individuabili come indicatori di compromissione (IOC). Il loro evidenziamento avviene nel momento in cui viene rilevata un'anomalia dal normale comportamento, risultato della domanda "Cosa NON sta facendo il suo lavoro?", "Cosa si è discostato dalla normalità?". Non bisogna confonderli con l'Intelligence, ma piuttosto come una sua componente.

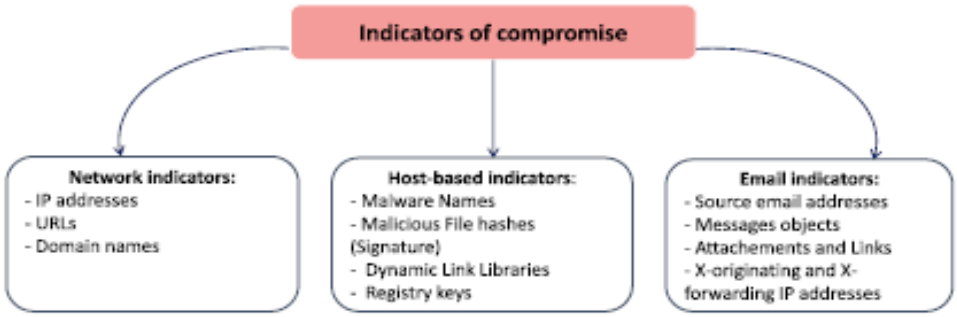


Figura 3.2 – Principali indicatori di compromissione [11]

L' immagine descrive i principali punti da monitorare e su cui percepire anomalie [12]:

- Quelli provenienti dalla rete e che mantengono una finestra temporale molto piccola dovuta alle diverse tattiche e strategie di rete adottate dall'attaccante per sferrare un attacco, indirizzi IP associati a botnet[33], server untrusted, URLs malevoli, ecc. .
- Quelli interni al sistema (infetto) come file di malware, hash d'attacchi, registri di sistema, tutte risorse prese di mira per la creazione di Trojan[34]
- Quelli individuati dal resto delle componenti di sistema che offrono servizi e funzionalità necessarie, siano essi strumenti di lavoro, mail, tool.

Un'altra tipologia di tool che utilizza i propri meccanismi interni per poter elaborare e analizzare direttamente i dati nel proprio ambiente di lavoro sono gli IDS[35]. Questi però hanno un comportamento passivo che gli permette di offrire solamente una post-notifica dovuta alla presenza della minaccia. Spesso si richiede l'intervento dell'operatore per il fixing. La conseguenza ovvia è stata quella di iniziare ad adottare meccanismi proattivi per la riduzione della finestra temporale che passava dalla notifica all'intervento. La soluzione passò per gli IPS[36], permettendo il blocco diretto al rilevamento di contenuti insicuri. La particolare configurazione adottata permette di poter attivare un'implementazione ai bordi della rete per il controllo del traffico in entrata (Network-basedIPS) e/o sulle postazioni interne la rete come ulteriore monitor per il traffico entrato (Host-basedIPS)[13]. La complementarità di IDS e IPS offre la possibilità di poter unire proattività e capacità difensive ed in tal senso sono stati sviluppate soluzioni che adottano un approccio ibrido, rendendone anche più semplice l'implementazione.

[33] Una rete composta da dispositivi infettati, zombie, a cui possono aggiungersi altri dispositivi infetti connessi ad Internet. Il botmaster può controllare il sistema tramite accesso remoto avviando attacchi.

[35] Intrusion Detection System, strumenti per il monitoraggio del traffico di rete con funzionalità di notifica dell'intrusione.

[36] Intrusione Prevention System, componenti attivi per individuare attività dannose, eliminare pacchetti malevoli o bloccare il traffico da un indirizzo IP

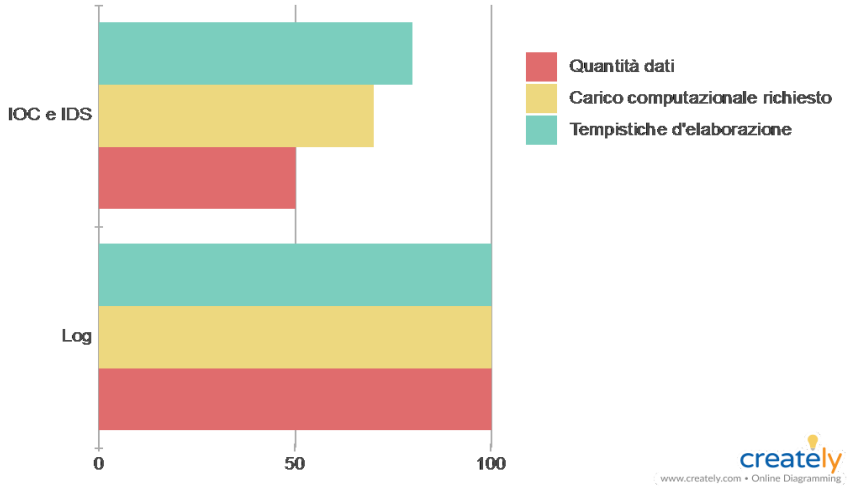


Figura 3.3 – Grafico per il confronto tra log e IDS

Da una parte si riduce la generalità degli eventi presenti nei file di log, dall'altra si massimizza il lavoro d'identificazione attraverso una vasta conoscenza dei comportamenti delle minacce.

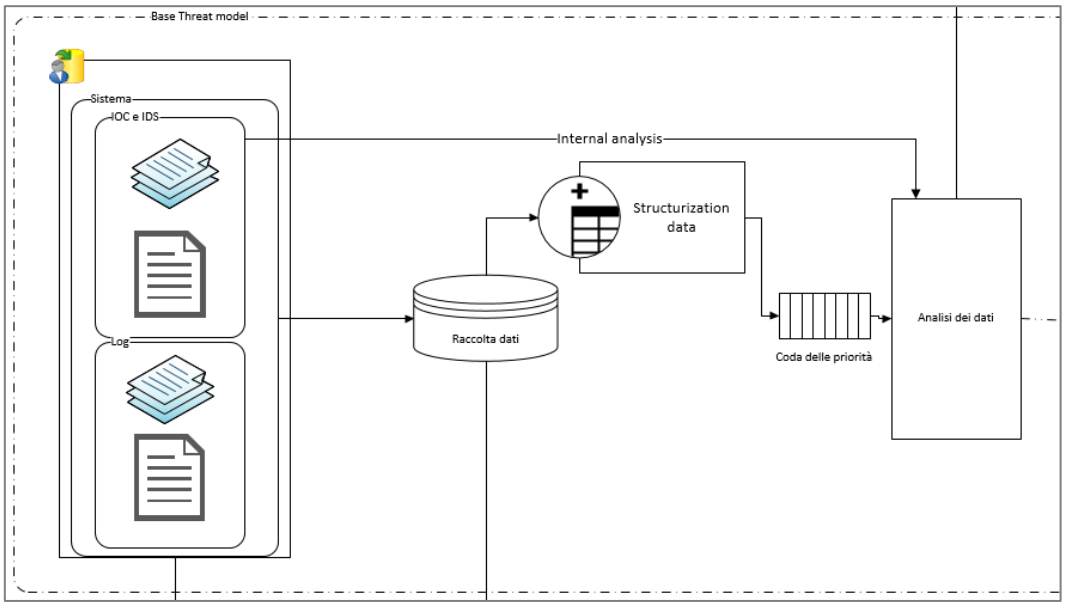


Figura 3.4 – Fase della raccolta dati da diverse sorgenti

3.2.1 Logstash e Filebeat

La Elastic è un azienda che cerca di proporre una soluzione all'implementazione di un ecosistema basato sull'individuamento delle minacce attraverso strumenti open source altamente complementari. Elastic è anche parte del nome del componente principale dell'ecosistema, Elasticsearch, motore d'indicizzazione particolarmente performante. Ma prima di passare da Elasticsearch è fondamentale riuscire ad ottimizzare la fase di raccolta dei dati da inviare. Esistono diverse soluzioni in commercio per lo scopo, ma la proposta di Elastic si basa molto sulla possibilità di combinare le caratteristiche di 2 tool particolarmente leggeri e performanti, Logstash e Beats, in modo particolare Filebeat.

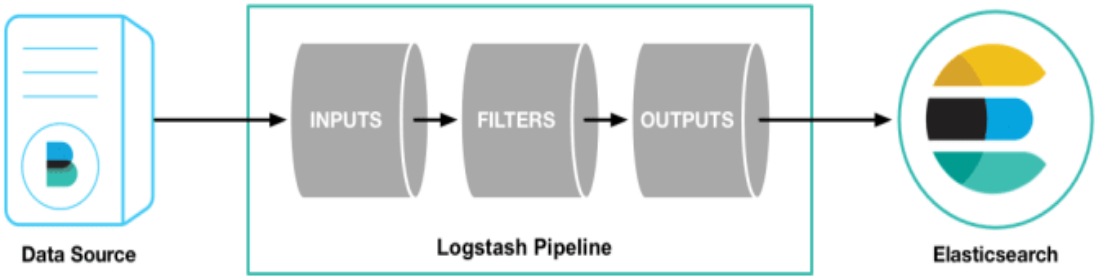


Figura 3.5 - Pipeline input e output per Logstash [37]

In realtà i 2 tool potrebbero tranquillamente esistere senza l'altro: sia Filebeat che Logstash permettono di inviare i dati raccolti al motore che li analizzerà, ma dato il carico di dover delegare tutto il lavoro a Logstash è necessario l'intervento di Filebeat per diminuire i consumi. Filebeat permette di raccogliere ed inviare diversi tipi di dati a Logstash che andrà ad elaborarli e ad inviarli ad Elasticsearch. I dati hanno bisogno di essere pre-processati prima di arrivare al motore d'indicizzazione. Logstash raccoglie dati di differente formato che deve andare a catalogare, arricchire e mappare. A tal proposito, è consigliato andare a settare sia diverse istanze di Logstash per gestire diversi tipi di dato, sia andare a migliorare la gestione dei dati in entrata per evitare perdite di dati tramite tool come Apache Kafka o Redis.

[37] https://www.peerlyst.com/posts/how-to-build-a-threat-hunting-platform-using-elk-stack-chiheb-chebbi

Versioni

Parlando di licenze open source, l'installazione è gratuita, ma contiene sezioni commerciali che è possibile provare per un periodo di prova limitato.

	FREE		GOLD	PLATINUM
	OPEN SOURCE	BASIC		
	Download		Request Info	Request Info
Beats				
Data Collection	✓	✓	✓	✓
Data Shipping	✓	✓	✓	✓
Modules	✓	✓	✓	✓
Monitoring and Management		✓	✓	✓
Logstash				
Data Collection	✓	✓	✓	✓
Data Enrichment	✓	✓	✓	✓
Data Shipping	✓	✓	✓	✓
Modules	✓	✓	✓	✓
Monitoring and Management		✓	✓	✓

Figura 3.6 – Versioni dei tool Logstash e Beats[38]

Installazione ed utilizzo

Bisogna partire dall'installazione di Logstash con i comandi:

```
$ echo 'deb http://packages.elastic.co/logstash/2.2/debian stable main' | sudo tee /etc/apt/sources.list.d/logstash-2.2.x.list

$ sudo apt-get update && sudo apt-get install logstash
```

Per la configurazione si andrà a considerare il file `logstash.yml` per settaggi generali come le fasi d'avvio e arresto, e il file `conf.d` nella directory `/etc/logstash/` per le istruzioni da eseguire per trattare i dati. Per quest'ultimo si tratta di un file in formato JSON composto da 3 sezioni:

1. Inputs, per indicare quali e dove saranno le sorgenti di dati
2. Filters, sezione dedicata al lavoro da compiere sui dati per poterli correlare, filtrare, ecc.

[38] https://www.elastic.co/subscriptions

3. Outputs, ovvero la destinazione dei file che può essere un salvataggio in un file oppure l'invio ad un ulteriore strumento, come Elasticsearch

Ci sarà bisogno di impostare la connessione in entrata proveniente da Filebeat, quindi il file dovrà comparire come in figura 3.7 .

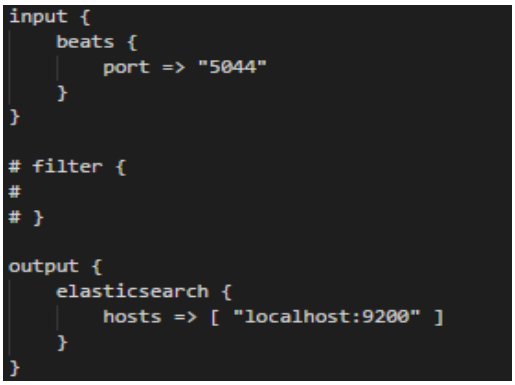


Figura 3.7 – Sample conf.d

Una volta completato si dovrà andare a riavviare Logstash per abilitare le modifiche di configurazione.

```
$     sudo service logstash restart
```

Ed ora è il momento di andare ad implementare Filebeat eseguendo i comandi d'installazione:

```
$     echo "deb https://packages.elastic.co/beats/apt stable main"
      |  sudo tee -a /etc/apt/sources.list.d/beats.list

$     wget -qO - https://packages.elastic.co/GPG-KEY-elasticsearch
      | sudo apt-key add -

$     sudo apt-get install filebeat
```

Completata l'installazione sarà necessario configurare Filebeat per la connessione a Logstash. Il file di configurazione `/etc/filebeat/filebeat.yml` (figura 3.8) permette di andare a settare parametri come le sorgenti da cui prendere i file di log. Per settare la connessione bisognerà andare a de-commentare la sezione output alla voce logstash e specificare come host l'indirizzo `localhost:5044` .

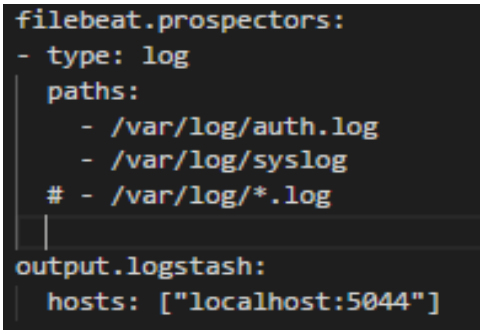


Figura 3.8 – Sample filebeat.yml

Come per Logstash, bisognerà riavviare il servizio per rendere effettive le modifiche.

```
$     sudo service filebeat restart
```

Il tempo impiegato per installare e settare Logstash e Filebeat sarà poi ampliamente ripagato dalla semplicità d'uso ed efficienza degli stessi. Non resta che attivare i servizi e valutarne il lavoro.

```
$     sudo systemctl start logstash.service

$     sudo systemctl start filebeat.service
```

Plugins

Logstash vanta un'ampia varietà di plugins - oltre 200 [39] - da poter implementare per personalizzare lo strumento. Con Filebeat è possibile integrare delle soluzioni pronte all'uso per diversi tipo di log come nginx o MySQL, soluzioni o moduli individuabili al percorso `/etc/filebeat/module.d` .

API

Logstash mette a disposizione delle API per la comunicazione con le informazioni che deve processare. Ci sono a disposizione diverse tipi di API per il monitoring:

- Node API, che permette di recuperare informazioni di diversi tipi come quelle per le pipelines o sul SO, es.

  ```
  $     curl –XGET 'localhost:9600/_node/os'
  ```

- Node Stats API
- Plugins API, per prendere tutte le informazioni sui plugins installati

[39] https://www.elastic.co/products/logstash

- Hot Threads API, per indicare quali processi stanno maggiormente incaricando le risorse

3.2.2 Nagios

Un'ottima applicazione per il monitoraggio dell'infrastruttura informatica è Nagios, tool open source multipiattaforma che verifica il corretto funzionamento dei diversi servizi del sistema come:

- monitoring sulla rete, SMTP, HTTP, ICMP, FTP, SSH;
- monitoring sul sistema, lavoro del processore, uso dell'hard disk, log di sistema;
- controllo da remoto;
- plugins per la personalizzazione dell'ambiente;
- alerting via mail e SMS;
- definizione di trigger per la gestione dinamica di situazioni anomale;
- interfaccia web-based.

E' possibile gestire la struttura del software tramite terminale o con l'interfaccia dedicata, permettendo la notifica di malfunzionamenti e problemi via mail o SMS.

Con l'implementazione di Nagios si includono diverse componenti per garantire il funzionamento:

- Nagios Remote Data Processor (NRDP), meccanismo per il trasporto dati che si appoggia a HTTP(S) e XML;
- Nagios Cross Platform Agent (NCPA), per il controllo attivo o passivo;
- Nagios Remote Plugin Executor (NRPE) , per il monitoring e la gestione da remoto sulla porta 5666.

Grazie alle molteplici versioni riesce ad adattarsi alle diverse realtà aziendali[14], piccole o grandi che siano, ad esempio: tramite l'acquisto di sensori monitorabili con Nagios, il Columbus Museum of Art è riuscito a gestire il problema dell'unità di raffreddamento, permettendo di ricevere segnalazione sui frequenti arresti e poter ovviare. Uno svantaggio di Nagios sta nella

configurazione, al quale bisogna dedicare particolare attenzione: conviene usufruire delle guide sul sito ufficiale[40] prima di andare ad implementarlo ufficialmente nel proprio sistema.

Versioni

In origine NetSaint, Nagios era implementabile esclusivamente su Linux, ma attualmente è uno strumento multipiattaforma per architetture a 64 bit. È possibile scegliere tra la versione gratuita o quella a pagamento in base alle necessità.

	Nagios Core	Nagios XI
Available as Source Install Script		√
Complete Infrastructure Monitoring	√	√
Hundreds of Free Addons	√	√
Open Source Monitoring Engine	√	√
Forum Support	√	√
Pre-Configured Virtual Machine		√
Web Configuration UI (CCM)		√
Mobile App (Nagios Mobile)		√
Business Process Monitoring		√
Custom Maps (Nagvis)		√
Database Backend		√
Integrated UI		√
Dashboards		√
Configuration Wizards		√
Scheduled Reporting		√
Configuration Rollback		√
Email and Phone Support		√

Tabella 3.1 – Versioni di Nagios

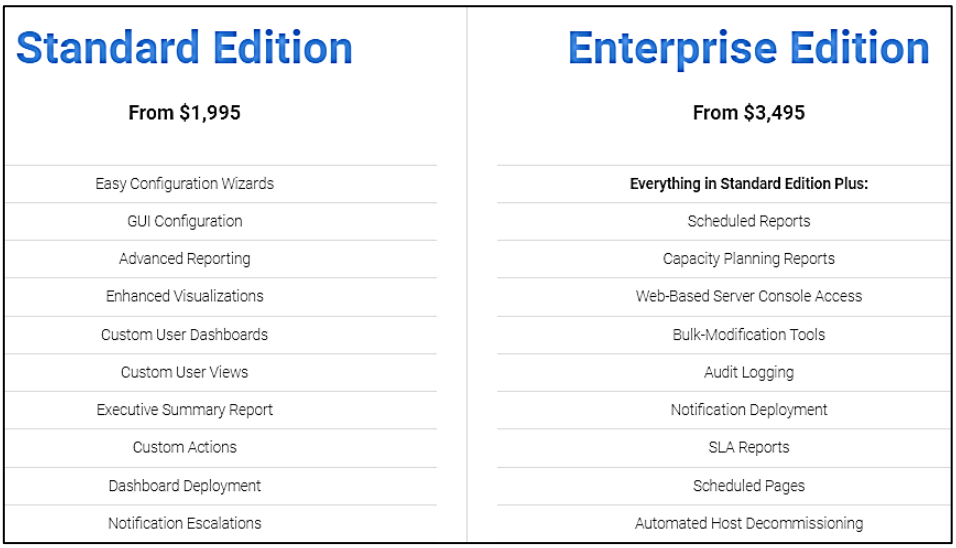
Nagios Core, la versione gratuita, permette un utilizzo di tutte le funzionalità di monitoring basilari come l'interfaccia web-based o alerting, ed offre l'opportunità di integrare funzionalità aggiuntive tramite plugins, gratuiti e a pagamento. Invece Nagios Enterprises, la soluzione commerciale, ha le fondamenta della versione Core, ma con la disponibilità di diverse versioni con servizi specifici, come Log Server, XI e Fusion, per le grandi compagnie:

- XI aggiunge VM preimpostate, disponibilità di app mobile, un supporto dedicato;

[40] https://library.nagios.com/training/

- Log Server, più specifica per la gestione di log con il settaggio di allarmi in base al rilevamento di anomalie;
- Fusion per il monitoring parallelo su più sistemi quindi poter specificare quale nodo XI o Core dovrà essere gestito.

Generalmente, tutte queste versioni permettono 60 giorni di prova prima di procedere all'acquisto di una licenza, a loro volta disponibili in diverse varianti (figura 3.9).



Standard Edition	Enterprise Edition
From $1,995	From $3,495
Easy Configuration Wizards	**Everything in Standard Edition Plus:**
GUI Configuration	Scheduled Reports
Advanced Reporting	Capacity Planning Reports
Enhanced Visualizations	Web-Based Server Console Access
Custom User Dashboards	Bulk-Modification Tools
Custom User Views	Audit Logging
Executive Summary Report	Notification Deployment
Custom Actions	SLA Reports
Dashboard Deployment	Scheduled Pages
Notification Escalations	Automated Host Decommissioning

Figura 3.9 – Versioni di Nagios XI

Per Nagios XI esiste anche una versione gratuita che permette di controllare solo un numero ristretto (7) di host.

Installazione ed utilizzo

Prima dell'installazione c'è bisogno di parlare di alcuni pre-requisiti necessari al funzionamento di Nagios, tra cui LAMP per l'interfaccia web-based e Sendmail per le notifiche d'allerta, ma anche la creazione di un utente e gruppo dedicato.

```
$     sudo apt-get install wget build-essential apache2 php php-gd
libgd-dev sendmail unzip autoconf gcc libc6 make libapache2-mod-
php7.2

$     sudo make install-groups-users

$     sudo usermod -a -G nagios www-data
```

Dopodiché sarà possibile procedere con la vera installazione andando a scaricare i pacchetti sorgente e per i plugins (al momento della scrittura 4.4.3 e 2.2.1):

1. Pacchetto installazione,

```
$ wget -O nagioscore.tar.gz
https://github.com/NagiosEnterprises/nagioscore/archive/
nagios-4.4.3.tar.gz

$ tar -xzf nagioscore.tar.gz

$ cd nagioscore-nagios-4.4.3

$ sudo ./configure --with-httpd-conf=/etc/apache2/sites-
enabled

$ sudo make all

$ sudo make install

$ sudo make install-commandmode

$ sudo make install-config

$ sudo make install-webconf
```

   Adesso è possibile completare le operazioni d'implementazione con il comando

```
$ sudo make install-daemoninit
```

   che andrà ad impostare il servizio al boot del sistema o procedere con i soliti comandi di `systemctl start` e `stop`.

2. Pacchetto plugins,

```
$ wget --no-check-certificate -O nagios-plugins.tar.gz
https://github.com/nagios-plugins/nagios-
plugins/archive/release-2.2.1.tar.gz
$ tar zxf nagios-plugins.tar.gz sudo ./tools/setup
$ sudo ./configure
$ sudo make
$ sudo make install
```

Per confermare il completamento dell'installazione basterà connettersi all'interfaccia tramite l'URL `nagios_server_public_ip/nagios` ed effettuare il login con le credenziali, ma prima bisognerà definire queste credenziali e abilitare la connessione all'interfaccia tramite Apache e firewall con i comandi:

```
$     htpasswd -c /usr/local/nagios/etc/htpasswd.users admin

$     sudo a2enmod rewrite

$     sudo a2enmod cgi

$     sudo ufw allow Apache

$     sudo ufw reload .
```

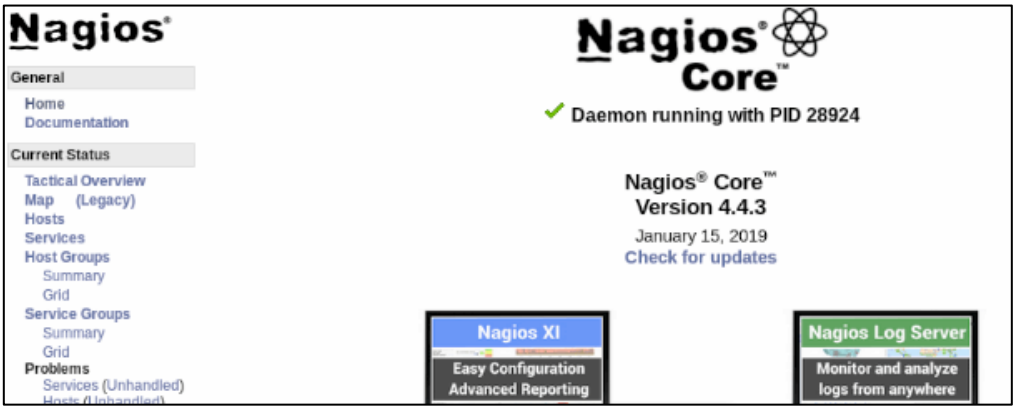


Figura 3.10 – Web-based interface Nagios[41]

La directory principale per la configurazione di Nagios è `/usr/local/nagios/etc` principalmente tramite il file `nagios.cfg` . Sarà necessario eseguire una verifica prima di riavviare il servizio usando l'opzione –v con Nagios.

```
$ /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Plugins

C'è una disponibilità abbastanza vasta circa i plugins, con numeri che vanno dai 50 per i plugins gestiti direttamente da Nagios, ai 3000 provenienti dalle community. Si dividono per categorie

[41] https://www.drivemeca.com/nagios-core-tutorial/

tutte disponibili attraverso la piattaforma dedicate Exchange[42]: hardware, cloud, security, network, ecc. .

3.2.3 Zabbix

Nato da un'idea di Alexei Vladishev e inizialmente utilizzato all'interno di un sistema bancario, Zabbix fa parte sempre di quella categoria di free software per il monitoraggio di sistemi IT, e dedicato ai diversi aspetti di controllo come il supporto a SNMP, TCP, ICMP, SSH, VM, Database o la notifica con SMS. Il suo funzionamento si fonda su una struttura composta da:

1. Agent, opzionalmente installabili e adatti al monitoraggio efficiente di molti host;
2. Item, singola risorsa monitorata;
3. Trigger, espressione di controllo con valore OK o PROBLEM su un item.

Per la raccolta dei dati si interfaccia a vari database quali MySQL, PostgreSQL, ecc. Questi dati poi, potranno essere gestiti graficamente attraverso la web-based interface. Inoltre, per semplici controlli sul funzionamento di servizi come HTTP non occorre installare il software sul sistema da monitorare. Permette 3 modalità d'intervento:

1. Network, scansione periodica su dati come IP, device, timer, ecc.;
2. Low-level, grafici, trigger, analisi di servizi, protocolli e linguaggi per database, rilevamento di macchine virtuali;
3. Auto-discovery, agent-based.

Come intuito non c'è l'obbligo di usare agents ed è possibile interagire anche con l'interfaccia.
In poche parole, più che un semplice IPS/IDS, è una vera è propria piattaforma per il controllo di sistemi informatici, ricca di funzionalità come:

- Raccolta dati;
- Analisi;
- Alerting;
- Dashboard;
- Distributed Monitoring;
- Proactive.

[42] https://exchange.nagios.org/#_ga=2.124491656.347565800.1552290437-1504939175.1549530292

Versioni

Appartenente alla società Zabbix LLC, si dispone di un'unica versione open-source indipendentemente dall'uso commerciale o non. Grazie alle qualità di scalabilità, monitoraggio distribuito, disponibilità dei servizi, facile integrazione, garantisce l'utilizzo per piccole e grandi imprese. Nonostante tutto viene messo a disposizione servizi tecnici e per il supporto a pagamento su abbonamento.

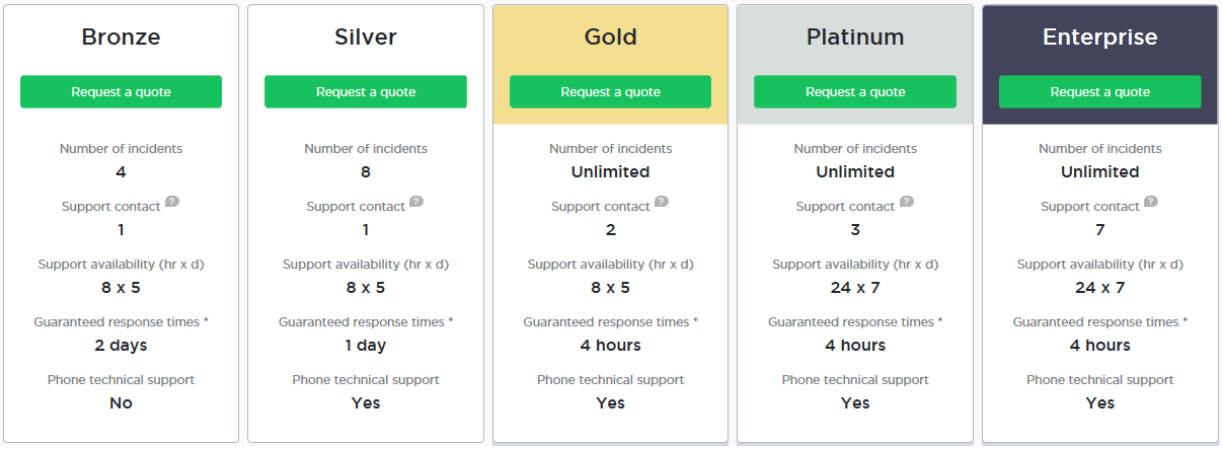


Figura 3.11 – Piani supporto tecnico per Zabbix

Installazione ed utilizzo

Tramite la pagina ufficiale[43] è possibile scaricare i sorgenti di Zabbix e poi andare a scompattare i pacchetti per l'installazione.

```
$     wget
https://repo.zabbix.com/zabbix/4.0/ubuntu/pool/main/z/zabbix-
release/zabbix-release_4.0-2+bionic_all.deb
$     dpkg          -i          zabbix-release_4.0-2+bionic_all.deb
$     apt update
```

Ora, inizieranno una serie di fasi in cui si cercherà di creare le fondamenta di Zabbix partendo dai requisiti, creando il database per le informazioni e arrivando all'esecuzione.

```
$     sudo apt-get install zabbix-server-mysql zabbix-frontend-php
zabbix-agent

$     mysql –u root –p
```

[43] https://www.zabbix.com/download_sources

```
$ mysql> create database zabbix character set utf8 collate utf8_bin;

$ mysql> grant all privileges on zabbix.* to zabbix@localhost identified by 'password';

$ zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
```

Arrivando alla configurazione, attraverso il file `/etc/zabbix/zabbix_server.conf` è possibile settare la password per il database zabbix e settare il giusto fuso orario con `/etc/zabbix/apache.conf` .

Non resta che impostare i servizi all'avvio con i comandi

```
$ systemctl enable zabbix-server zabbix-agent apache2
```

Conclusa questa prima fase d'installazione, basterà connettersi all'interfaccia web all'URL `server_ip_zabbix/zabbix` per avviare la seconda parte.

Figura 3.12 – Procedura installazione da interfaccia Zabbix

Procedendo lungo gli step si andranno a fare delle verifiche sulla presenza dei software nel sistema necessari per il funzionamento, impostare una connessione col database MySQL creato in precedenza, dopodiché ci si potrà avviare all'utilizzo vero e proprio di Zabbix inserendo username(admin) e password(zabbix) di default per il log-in.

Attraverso l'interfaccia sarà possibile settare gli utenti da monitorare dal pannello Host, impostando tutti i parametri necessari. Allo stesso modo sarà possibile sia settare trigger per il controllo su determinati eventi, sia impostare risorse e servizi specifici il quale si vuole monitorare. Attraverso i *Media types* si andranno ad impostare le notifiche di allerta.

Plugins

Parallelamente ai plugins, sono disponibili anche dei template che permettono di raggruppare più host e applicare configurazioni a tutti efficientemente. Cambiare qualcosa sul template permette di propagare la modifica a tutti, riducendo il carico di lavoro.

API

Tramite le web-based API è possibile integrare altre componenti e funzionalità come go-zabbix, o host.create per creare un nuovo host, o ancora user.login per l'autenticazione, dialogando tramite JSON. Permettono di dialogare con tutte le funzioni di Zabbix.

3.2.5 Confronto

I 3 software descritti in questo paragrafo, Logstash-Filebeat, Nagios e Zabbix si rendono subito partecipi nel processo di modeling delle minacce grazie alle loro funzioni e caratteristiche[15][16]. La realtà dei fatti però è un'altra: anziché un vero confronto, non è insolito veder collaborare questi tool, ma se considerati singolarmente è bene andare a distinguere i punti chiavi di ognuno.

	Logstash - Filebeat	**Nagios**	**Zabbix**
Funzione principale	Raccolta e pre-elaborazione log	Network Monitoring	Network Monitoring
Linguaggio	Java	C	PHP, C, Java, JavaScript
Multipiattaforma	√	√	√
Installazione	Facile	Media	Media

Configurazione	text-files, .yml	templates e text-files	web-based interface
Plugins	1000+	1000+	100+
Interazione	da terminale	da terminale	da terminale o interfaccia
Community	ampia	ampia	in crescita
Soluzione per la visualizzazione	Integrazione con Kibana	Raw web interface	Include grafici e dashboard
Licenza	Totalmente opensource	Gratuito e Enterprise edition	Completamente gratuito
Carico computazionale	Basso	Medio-Alto	Medio-Alto
Database	non-relazionali	SQL	SQL
Dimensioni	150+ MB	10+ MB	15+ MB

Tabella 3.2 – Confronto Logstash vs Nagios vs Zabbix

Generali

Innanzitutto i tool seppur open-source, si differenziano per le versioni che distribuiscono:

- Logstash e Filebeat in versione gratuita o soluzione a pagamento con l'aggiunta di più servizi e funzionalità;
- Nagios oltre alla versione gratuita Core, include anche una soluzione enterprise come Nagios XI;
- Zabbix rende disponibile un'unica versione gratuita, però offre soluzioni tecniche e di supporto a pagamento.

Installazione e Configurazione

Per i 3 software l'installazione, come la configurazione, richiede delle tempistiche quantomeno adeguate, vedendo spiccare come semplicità Zabbix grazie anche al wizard d'installazione tramite l'interfaccia. Si tratta comunque di tool con il bisogno di dover impostare al meglio la struttura in modo da essere efficienti per il loro compito:

- Logstash e Filebeat richiedono diversi passi per la configurazione come settare le fonti da cui ricevere dati, o impostare l'output dell'elaborazione verso altri tool, o permettendo un pre-filtraggio delle informazioni.

- Nagios, anche grazie all'uso dei template, garantisce una facile configurazione individuata in un unico file.
- Zabbix cerca invece di garantire una facile configurazione soprattutto tramite l'interfaccia grafica.

Monitoraggio

La funzionalità di Logstash e Filebeat sta soprattutto in un ottimo lavoro di raccolta file log, quindi con monitoraggio si intende prettamente questo compito. Nagios e Zabbix invece adottano approcci agent-based o agentless per permettere di ottenere migliori risultati nel monitoring. Come detto in precedenza, per fare questo con Nagios ci sarà bisogno di implementare Nagios Remote Plugin Executor e dopodiché andare a strutturare il file di configurazione apposito, mentre con Zabbix il tutto è permesso via interfaccia.

Interfaccia

Logstash è completamente gestito tramite terminale, quindi non utilizza un'interfaccia grafica, mentre Zabbix e Nagios permettono di avere una web-based interface come punto aggiuntivo (seppur quest'ultimo permetta di poter solo visualizzare i risultati del suo funzionamento, e non la configurazione).

Community

La recente ascesa di Zabbix lo mette sicuramente in svantaggio rispetto a Logstash e Nagios per l'ampiezza della community, ma è la crescita della popolarità a far riflettere.

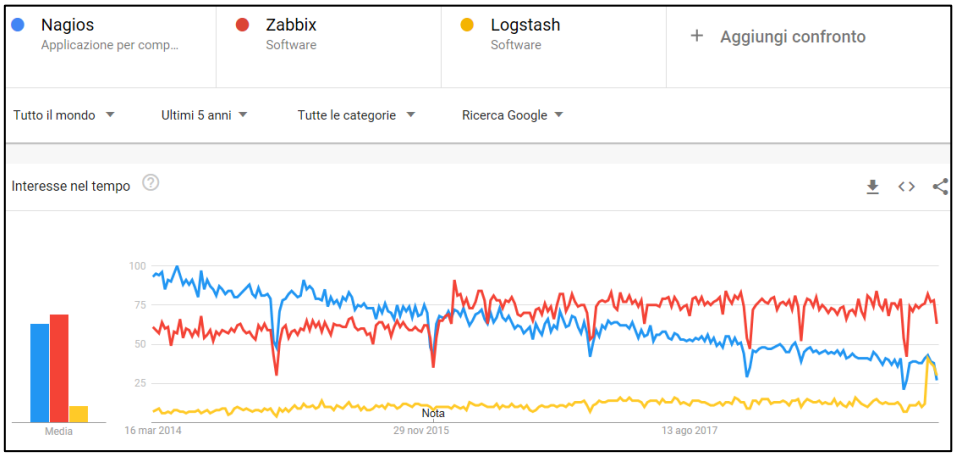


Figura 3.13 – Interesse ultimi 5 anni per Logstash, Nagios e Zabbix[44]

[44] https://trends.google.it/trends/explore?q=%2Fm%2F03hm4g&geo=IT

E’ evidente come la crescita di Zabbix abbia contrastato Nagios, mettendo in evidenza come l’interesse per Logstash sia valutato soprattutto in particolari ambiti d’analisi di log e nell’implementazione con tutto l’ELK Stack, senza di cui non esprime tutto il suo potenziale.

Conclusione

In base all’analisi svolta nel paragrafo si è giunti alla conclusione che i 3 software differenziano per diversi fattori chiave, ma offrendo diverse soluzioni adatte per l’integrazione in un threat model[17][45]. Dovendo sceglierne uno la scelta ricadrebbe senza dubbio sulla versatilità e la completezza di Zabbix, ma nonostante ciò è possibile ritrovare anche implementazioni che uniscono i diversi tool. Anche Logstash e Filebeat, ottimi strumenti per la gestione di log, restano “isolati” se implementati singolarmente, però facenti parte di una struttura più completa come ELK Stack che permette loro di essere ancora più efficienti.

3.3 Analisi dei dati

E’ stata fatta distinzione tra dati da analizzare e dati già individuanti possibili minacce che non comportano un ulteriore lavoro di analisi e per cui verranno trattati ulteriormente nelle fasi successive. Il lavoro di analisi consiste nel prendere in input il materiale individuato nella fase di raccolta dei dati - file di log, eventi di sistema, ecc. - e cercare di gestirlo e correlarlo a possibili situazioni di pericolosità. Questo lavoro spesso inizia con una fase d’indicizzazione per trattare le sorgenti di informazioni in modo da rendere più efficiente sia l’analisi sia la correlazione a comuni minacce. Gran parte dei tool presenti in commercio basano molte delle loro peculiarità sul fatto di lavorare con i database relazionali e non, permettendo di gestire con efficienza diversa le informazioni in entrata. Quindi, il lavoro di analisi è composto da 2 parti:

- L’indicizzazione, l’elaborazione, la gestione, e la correlazione dei dati;
- L’eventuale possibilità di un intervento umano per un analisi diretta o per una specifica richiesta da sottoporre al risultato dell’indicizzazione, come filtrare per una categoria di dato, ad es. “solo protocollo UDP e ARP” o per un certo intervallo temporale, es. “04-12-2018 , 08-02-2019”.

[45] https://github.com/vulnersCom/zabbix-threat-control

La struttura dei file che vengono mandati in pasto all'analisi tiene fede al concetto di database e per tanto le informazioni sono modellate tenendo conto proprio di questo fattore. Si potrebbe dire che c'è una gestione sotto forma di struttura tabellare.

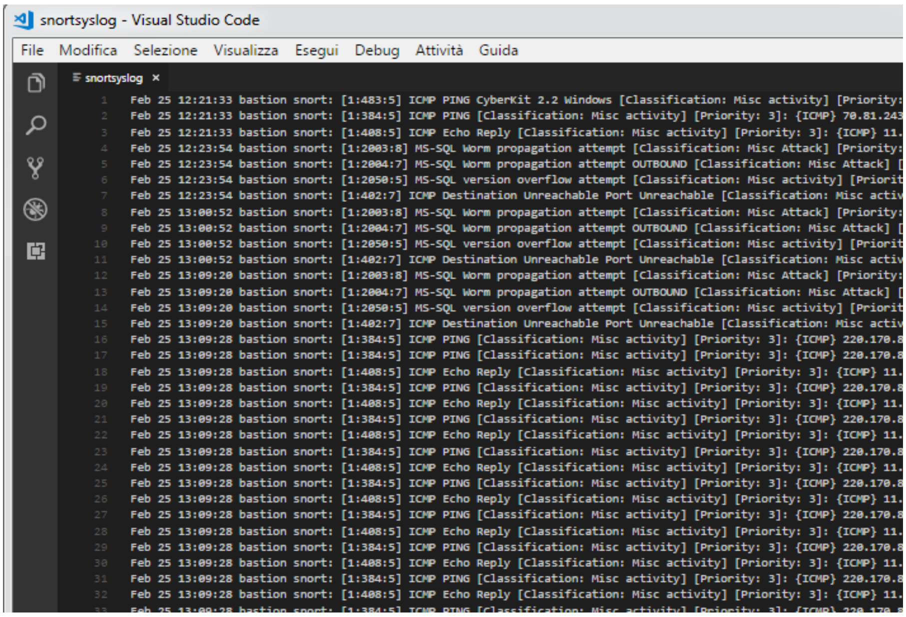


Figura 3.14 – Sample file .log di Snort [46]

Questo permette di individuare tool che lavorano meglio con certi tipi di formato di dati o con una diversa gestione del dato e del database utilizzato. Questa situazione è ancora parte del lavoro di analisi in quanto si avrà sia l'operatore che potrà consultare e monitorare attraverso query, sia una gestione di trigger che andranno a stabilire una condizione di dinamicità rispetto all'individuamento di determinate voci e categorie di informazioni caratterizzanti minacce e attacchi. Qui appunto rientrano parallelamente l'altra categoria di dati individuata nella fase di raccolta, cioè quei dati che non necessitano di un'analisi o confronto per segnalare una minaccia. L'attuale situazione si verifica poiché questi dati, per la particolarità dovuta ai tool che li lavorano, non richiedono analisi grazie al fatto che il loro reperimento avviene sull'individuazione diretta di informazioni che discostano dal normale ("Cosa vogliamo che NON accada?") e quindi di facile riscontro. Per tanto il loro processo di raccolta e di analisi è molto più compatto e racchiuso nello specifico tool trattato, quindi alla fase di analisi descritta qui resta solo il compito del permettere l'intervento umano per un'analisi più diretta e specifica, per poi passare direttamente alla fase d'individuazione.

[46] http://log-sharing.dreamhosters.com/

3.3.1 Apache Solr

Appartenente alla numerosa famiglia di prodotti open-source di Apache, Solr è un progetto che permette l'analisi di grandi moli di dati, e quindi l'indicizzazione e interrogazione di grandi database, attraverso ricerca full-text[47] [18]. La parte centrale è l'implementazione di Lucene, altro progetto Apache, che contribuisce ad una prestazione performante[19]. Le sue caratteristiche principali sono:

- Utilizzo di standard come XML e JSON, poter inserire diverse tipologie di file permette di non perdere tempo per andare ad effettuare conversioni (fastidiose) di formato, ma poter inviare direttamente a Solr i file
- Possibilità di poter usare più linguaggi per dialogare, grazie alle REST APIs
- Un'ottima scalabilità
- Capacità di backup e recovery
- La possibilità di indicizzare documenti complessi e ricchi come PDF

Esattamente come anche altri strumenti del settore, lavora su database non relazionali. E' fondamentale notificare come in passato (e anche oggi) è stato considerato un ottimo mezzo per migliorare le prestazioni in ambienti consideranti grandi quantità di dati come Apache Hadoop. Lo strumento permette di interrogare i dati tramite diverse soluzioni:

1. Piattaforma
2. APIs
3. CURL

La presenza e il potere dell'interfaccia web-based, per questo particolare tipo di strumento è molto fondamentale, anche per avvicinare diversi tipi d'utenze. Non è pratica diffusa trovare motori di ricerca che implementano un'interfaccia oltre il normale approccio con riga di comando. C'è un termine particolarmente associato poi a Solr, cioè *core*. Ogni core corrisponde ad un singolo indice, ha una sua propria directory di gestione ed un proprio file di configurazione.

Installazione ed utilizzo

La sua struttura si basa su Java ed è necessaria una buona dimestichezza per poter ampliare l'ambiente. Bisogna partire andando a scaricare l'ultimo pacchetto d'installazione disponibile

[47] Ricerche attraverso frasi e token, cosi come la possibilità di auto-completamento.

sul sito ufficiale https://lucene.apache.org/solr/mirrors-solr-latest-redir.html (al momento della scrittura l'ultima release è la 7.7.0) e andare a scompattare il pacchetto.

```
$     wget    http://it.apache.contactlab.it/lucene/solr/7.7.0/solr-
      7.7.0.tgz
$     tar zxf solr-7.7.0.tgz
$     cd ~/solr—7.7.0
$     bin/solr start
$     bin/solr create –c collection
```

E' molto interessante la possibilità resa disponibile da parte del progetto Solr di diversi tutorial su come la piattaforma possa essere implementata ed utilizzata. Particolare interesse lo si mostra soprattutto per la possibilità di andare ad istanziare su una stessa macchina diversi nodi grazie alla modalità SolrCloud Mode[48] disponibile proprio attraverso procedura guidata con l'aggiunta al comando `bin/solr start` dell'opzione `-e cloud` . In alternativa si può passare direttamente alla creazione di core `bin/solr create –c name` . Ora il servizio sarà in ascolto sulla porta 8983. A questo punto non resta che interagire direttamente con lo strumento per inquadrarne le qualità descritte in precedenza.

Naturalmente bisogna partire dall'inserimento di voci al nostro motore di ricerca che può avvenire spostandoci nella directory interessata e importando file con estensione XML, CSV, JSON o presi direttamente da database con il comando `java –jar post.jar sample.xml` . Può essere utile andare a definire, attraverso gli schemi, la struttura dei campi e il loro tipo, di solito con un file `schema.xml` . Dall'altra parte, è possibile interrogare direttamente dall'interfaccia di Solr o utilizzando direttamente la barra per inserire la query di ricerca, come `http://localhost:8983/solr/select/?q=sample` .

La configurazione principale di Solr è caricata nel file `solrconfig.xml` e riesce a gestire i diversi aspetti caratterizzanti lo strumento come:

- Gestire le richieste in entrata, come aggiungere un documento
- Gestire l'interfaccia web
- Gestire i particolari processi in ascolto per le query e i trigger

[48] http://lucene.apache.org/solr/guide/7_6/solr-tutorial.html

Plugins

Solr permette di lavorare con plugin opzionali per andare ad inserire anche la possibilità di indicizzare documenti complessi e ricchi come PDF[49][50].

API

E' particolare come sia permessa la coesistenza e la complementarietà di due versioni di API, ovvero quelle pre-esistenti, sviluppate e consolidate insieme al tool, V1, e le V2 che vanno ad aggiungere caratteristiche innovative. Inoltre è interessante la possibilità di poter configurare direttamente Apache Solr attraverso le APIs e il dialogo che permettono verso un file alternativo, `configoverlay.json`, rispetto a quello di base.

3.3.2 Elasticsearch

Elasticsearch è un componente dello stack ELK (E->Elasticsearch) e rappresenta un motore di ricerca e indicizzazione dei dati valutato come "il più utilizzato", mostrando una crescita esponenziale. Elastic risulta una delle più recenti aziende nel settore. Lo scopo dello sviluppo era quello di poter lavorare comodamente offrendo una soluzione ottima anche per la distribuzione, ed in tal senso si è concentrati nel gestire file JSON. Sempre con l'intento di una facile interazione, Elasticsearch offre l'opportunità di poter dialogare anche con diversi linguaggi come Java, Python, Curl, ecc. .

[49] https://wiki.apache.org/solr/IntegratingSolr
[50] http://wiki.apache.org/solr/SolrPlugins

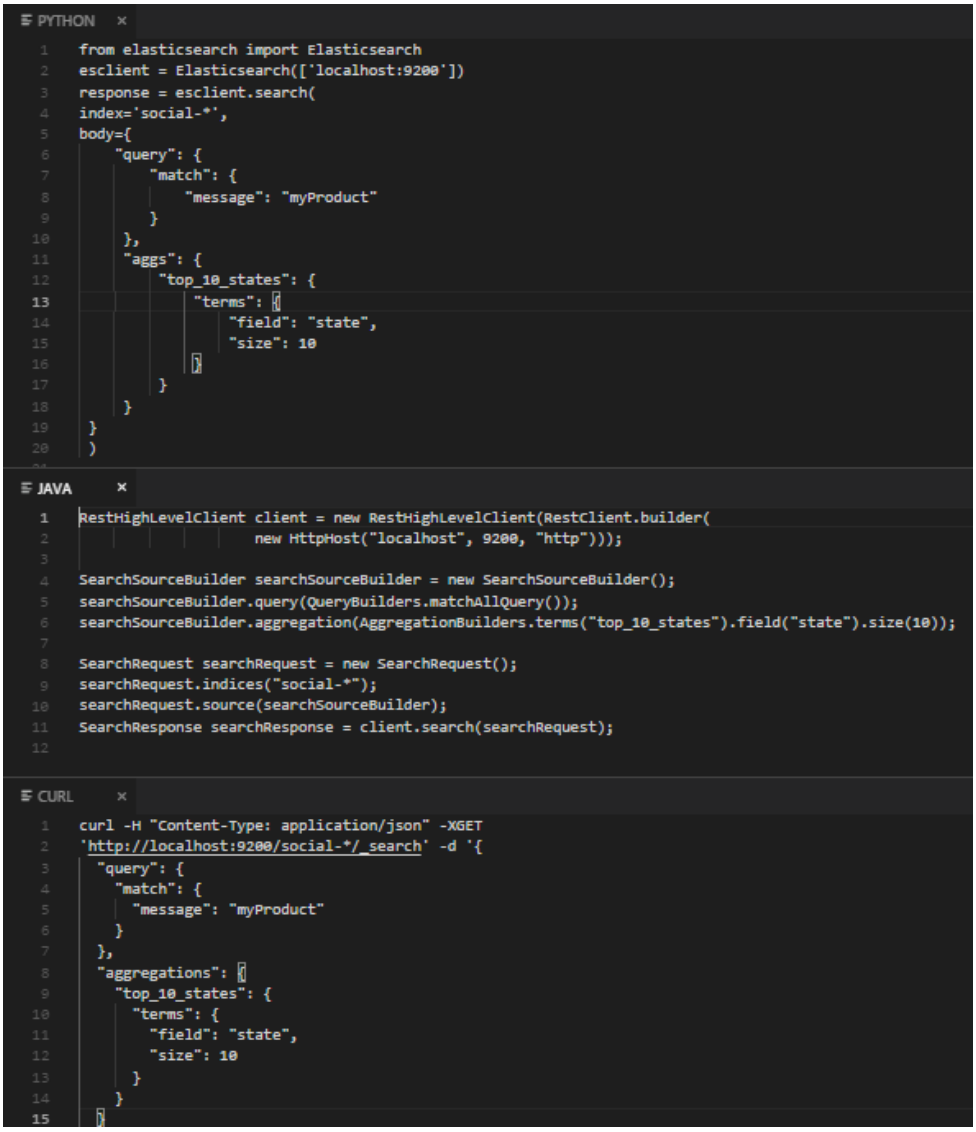


Figura 3.15 – Diversi tipi di linguaggio per la comunicazione con Elasticsearch

Basti considerare che con il crescente problema della gestione dei BigData e per poter sfruttare le sue potenzialità di ricerca real-time, Elasticsearch ha introdotto una collaborazione con Apache Hadoop. Grazie anche allo sviluppo delle community, alle molteplici funzionalità richieste nell'ambito IT ai giorni d'oggi e all'appartenenza ad una architettura emergente come lo stack ELK, introduce la possibilità di integrare soluzioni per il monitoring, machine learning e security. Il rapporto tra Elasticsearch e le altre soluzioni è da definire come un rapporto 1:10000 circa la velocità dei risultati, questo anche perché la ricerca nei normali database non permette ricerche full-text che riduce di molto il numero dei dati letti. Il compito d'indicizzazione parte dalla raccolta di dati per arrivare alla trasformazione di quest'ultimi in JSON, ottenendo semplici correlazioni chiavi-valori.

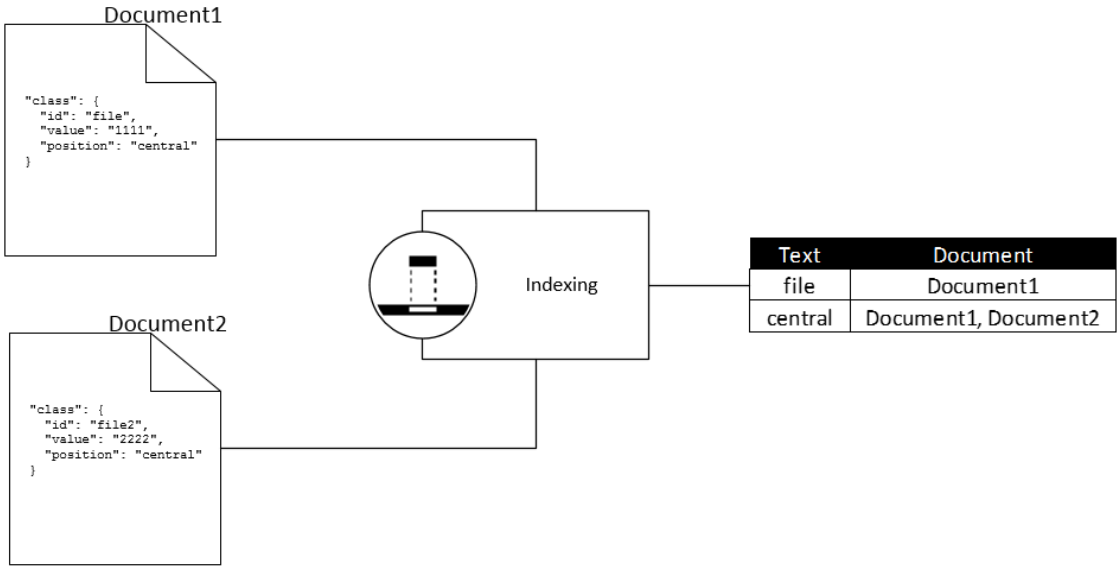


Figura 3.16 – Indicizzazione con Elasticsearch di diversi documenti

Inoltre non si tratta di database relazionali, e per tanto ci sarà bisogno di una normalizzazione dei dati. I documenti sono partizionati su shards, ovvero piccoli pezzi condivisi con qualsiasi nodo che permettono di ottenere scalabilità e distribuzione. Questo perché la capacità di archiviazione disponibile potrebbe non essere sufficiente, ed inoltre potrebbe portare a eccessivi deficit prestazionali. Poi grazie ad un meccanismo di replica è sempre possibile ricostruire le informazioni.

Versioni

Il tool può essere eseguito fisicamente sulla propria postazione attraverso la versione gratuita, oppure ospitato in cloud direttamente da Elastic, che potrà gestirne diversi aspetti come, ad esempio, mantenere la struttura aggiornata. Esiste una versione, Cloud Enterprise, per le aziende che hanno bisogno di gestire molte risorse, molti utenti e quindi con bisogni di scalabilità, di backup, con possibilità di lavorare da un'unica postazione.

	FREE		GOLD	PLATINUM
	OPEN SOURCE	BASIC		
	Download		Request Info	Request Info
ELASTIC STACK				
Elasticsearch				
Scalability & Resiliency	✓	✓	✓	✓
Query & Analytics	✓	✓	✓	✓
Data Enrichment	✓	✓	✓	✓
Management & Tooling	✓	✓	✓	✓
Security			✓	✓
Alerting			✓	✓
Machine Learning				✓

Figura 3.17 – Diverse versioni del tool Elasticsearch

Installazione ed utilizzo

Elasticsearch è sviluppato con Java ed ha quindi bisogno di JVM per poter essere eseguito. Come al solito il procedimento indicato coinvolge macchine Linux, ma lo strumento è disponibile anche per gli altri sistemi operativi:

- Scaricare e installare le chiavi pubbliche

  ```
  wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
  ```

- Aggiungere le repository
- ```
  echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list
  ```

- Installare Elasticsearch

  ```
  sudo apt-get install elasticsearch
  ```

- Avviare il servizio

  ```
  sudo systemctl start elasticsearch.service
  ```
  ```

La verifica di funzionamento la si fa collegandosi (in locale) a `localhost:9200` con il comando `curl -XGET localhost:9200` o direttamente con il browser controllando che il risultato sia come in figura 3.18 .

Figura 3.18 – JSON iniziale Elasticsearch

Per la configurazione ci si serve della directory `/etc/elasticsearch` e del file `elasticsearch.yml` . Si potrà inviare direttamente documenti in format JSON, e ricevere come risposte alle GET un corpo JSON, compito adatto alle APIs disponibili. E' d'aiuto se si usa l'opzione *?pretty* per migliorare il risultato grafico di una richiesta. Per eseguire una ricerca invece, utilizzo l'opzione `_search?q=` con la q che indica la query da inserire. Parallelamente agli schemi nei DB relazionali, in Elasticsearch c'è il mapping. Non adottare questo passaggio non blocca il lavoro di ricerca, ma crea un deficit prestazionale dovuto al salvataggio come stringhe dei valori dei file da indicizzare. Per tanto il mapping permette di identificare i campi, associarli a dei tipi, .. e migliorare le prestazioni:

```
$ curl -XPUT http:localhost:9200/classes/class/_mapping -d @mapping.json
```

Plugins

Il modo migliore per aggiungere funzionalità al nostro motore di ricerca. Il comando per le installazioni è il seguente:

```
sudo bin/elasticsearch-plugin install [plugin_name]
```

Ci sono diverse possibilità di scelta, tutte che permettono di andare a migliorare l'utilizzo, ricostruire l'ambiente che più si desidera o per cui faremo utilizzo, e cosi via. Ecco alcune delle più fondamentali[51]:

- Alerting Plugins
- Discovery Plugins
- Analysis Plugins
- Security Plugins
- Store Plugins

E' addirittura possibile poter aggiungere nuove APIs e permettere di migliorare la ricerca e il mapping[52]:

- SQL language Plugin
- Elasticsearch Taste Plugin
- Elasticsearch Experimental Highlighter
- Elasticsearch Trigram Accelerated Regular Expression Filter

API

Costituito da HTTP API e Java API. Utili per comunicare con Elasticsearch e l'ambiente d'analisi potendo gestire e amministrare gli indici, i nodi, eseguire operazioni CRUD[53] o di filtraggio.

Elasticsearch	DB-relazionale
GET	Select
PUT	Update
POST	Insert
DELETE	Delete

Tabella 3.3 – Confronto richieste ai database

Ecco un esempio dei diversi comando per il dialogo con Elasticsearch:

- Select * from class where id = 1

```
$     curl -XGET localhost:9200/classes/class/1
```

- Create index, type, id document from file

[51] https://www.elastic.co/guide/en/elasticsearch/plugins/6.6/index.html
[52] https://www.elastic.co/guide/en/elasticsearch/plugins/6.6/api.html
[53] Create, Read, Update, Delete

```
$ curl -XPOST localhost:9200/classes/class/1 -d @sample.json
```

- Create index name classes;

```
$ curl -XPUT localhost:9200/classes
```

- Delete from class where id = 1;

```
$ curl -XDELETE localhost:9200/classes/class/1
```

3.3.3 Confronto

I due software permettono un'attenta ricerca all'interno dei dati raccolti e mandati impasto ad ognuno. Riuscire a fare un paragone non è quindi lavoro facile. Il risultato del confronto tra Solr ed Elasticsearch (tabella 3.4) mostra come ci siano molte caratteristiche in comune. Cosa è meglio tra i due tool confrontati? Apache Solr o Elasticsearch? Molto probabilmente il modo migliore per riuscire a stabilire un rapporto di comparazione sarebbe quello di rispondere con questa domanda: Cosa è meglio *per me* tra i due tool confrontati? Cosa fa meglio per il *mio* caso?

Tool	Apache Solr	Elasticsearch
Funzione principale	search and index engine ideale per grandi quantità di dati	search and index engine ideale per l'analisi di file di log
Multipiattaforma	√	√
Installazione	Media	Easy
Configurazione	.xml , .json	.yml
Global Environment	√	√
Data Source	JSON, XML, CSV	JSON
Interrogazione	da terminale e interfaccia web-based	da terminale
Community	√ - in crescita	√
Soluzione per la visualizzazione	Banana	Kibana

Licenza	totalmente opensource	parzialmente opensource
Carico computazionale	alto	Basso
Motore	Lucene	Lucene
Database	non-relazionali	non-relazionali
Nascita	2004	2010
Contribution	libero	decisione finale dell'azienda
Dimensioni	100+ MB	20+ MB

Tabella 3.4 – Confronto Apache Solr vs Elasticsearch

Generali

Solr nasce nel 2004, meno recente rispetto ad Elasticsearch, 2010. Si consideri che Elasticsearch è parte integrante di un'ambiente particolarmente studiato per la coesione tra diversi tool, ELK Stack, cosa che con Apache Solr non è ancora ben definita. Secondo il sistema di valutazione DB-Engines[54] risulta che i 2 strumenti analizzati siano tra i più considerati e scelti per il campo dei motori di ricerca (figura 3.19), in modo particolare Elasticsearch con i più grandi valori di crescita.

Rank			DBMS	Database Model	Score		
Feb 2019	Jan 2019	Feb 2018			Feb 2019	Jan 2019	Feb 2018
1.	1.	1.	Elasticsearch	Search engine, Multi-model	145.25	+1.81	+19.93
2.	2.	2.	Splunk	Search engine	82.81	+1.39	+15.55
3.	3.	3.	Solr	Search engine	60.96	-0.52	-2.91
4.	4.	4.	MarkLogic	Multi-model	14.99	+0.73	+3.97
5.	5.	5.	Sphinx	Search engine	7.57	-0.12	+1.50

Figura 3.19 – Classifica motori d'indicizzazione

Configurazione

Si differenziano in base alle loro dimensioni: Solr richiede un download di più di 100MB, Elasticsearch poco più di 20MB. I due strumenti sono entrambi basati sulle librerie di ricerca Lucene, quello che però li differenzia è come sono state effettuate le implementazioni per ognuno. Solr è fortemente personalizzabile per renderlo adeguato alle nostre esigenze, ma

[54] https://db-engines.com/en/ranking/search+engine

richiede un consumo di risorse maggiore rispetto ad Elasticsearch. D'altra parte Elasticsearch, seppur con l'aiuto di Kibana risulti ideale per il monitoring, riesce ad essere una soluzione per chi ha bisogno di non faticare con la gestione dello strumento. Ciò non toglie che l'integrazione di plugins è limitata ad un'attenta configurazione e soprattutto alla scelta di un piano a pagamento.

Data source

Elasticsearch gestisce l'interazione con file JSON, mentre Solr include anche file XML, CSV, ecc. . La gestione dei documenti e dell'indicizzazione viene gestita da entrambe le parti con le API, ma lavorando in modo differente e appoggiandosi a soluzioni "esterne": Solr con Apache Tika riesce ad elaborare file come PDF o Word, invece Elasticsearch adotta Logstash per la raccolta dalle sorgenti.

Query request

Solr offre una soluzione più accessibile rispetto ad Elasticsearch, in quanto oltre alla gestione di documenti e di richieste per la ricerca o la gestione da terminale, permette una gestione di questi aspetti anche attraverso interfaccia web-based.

Community

Le community sono entrambe molto attive, seppur con rilievo quella di Elasticsearch, il che permette di poter avere sempre un feedback sui bug o le funzionalità. Una questione che rileva forti discussioni però, è quella riguardante la gestione dei contributi resi dalle community: in Solr c'è sempre posto per contribuire allo sviluppo dello strumento, mentre con Elasticsearch c'è bisogno di attendere per l'accettazione del contributo.

Conclusione

In conclusione si potrebbe dire che si debba scegliere Solr per il processo di grandi quantità di dati, possibilmente avendo in collaborazione l'aiuto e la dimestichezza con altri ambienti come Apache Hadoop o Hortonworks, o per un'analisi circa contenuti testuali, mentre Elasticsearch risulta la soluzione più adatta se c'è bisogno soprattutto di prestazioni, come anche la possibilità di visualizzare (con Kibana) i risultati dell'analisi. Tutte queste informazioni portano al matching con casistiche dove sono richieste funzionalità avanzate di ricerca come per la raccolta e l'analisi di log per individuare minacce, quindi adatto per il threat model, il che porta alla proposta di utilizzo di Elasticsearch per questa particolare fase d'analisi dati. La scelta ricade

sulla caratteristica di utilizzo prestante soprattutto per ambienti in crescita e la possibilità di avere a disposizione un ecosistema robustamente consolidato come ELK Stack.

3.4 Individuare, Condividere, Pubblicare

Conseguentemente al lavoro d'analisi si passa a quella che viene considerata la fase più ardua del processo di threat model: l'individuamento della minaccia. Come accennato nel paragrafo precedente, il lavoro d'analisi si conclude con report contenenti i correlamenti tra le informazioni. Per tanto l'input della fase d'individuazione sarà l'unione delle informazioni provenienti dai report sui file di log e sui report degli indicatori di anomalie come IDS, IPS, IOC. Anche qui inizierà un lavoro misto:

- Da una parte avremo la necessità di correlare queste informazioni a piattaforme online per le minacce che, appunto, collezionano ampi database di report condivisi tra organizzazioni, enti di sicurezza e chiunque possa inviare feed su minacce sempre in evoluzione;
- Parallelamente, è sempre consigliato l'intervento di esperti del settore, frequentemente presenti come figure professionali all'interno dell'azienda o assunti da terze parti per poter effettuare una correlazione diretta dei dati ricevuti. Spesso diventa un lavoro di collaborazione tra team di professionisti, tester e amministratori.

E' bene ricordare ancora una volta che bisogna tener conto anche di quelle minacce ritenute "potenziali" e cercare di concentrare la parte iniziale del lavoro su minacce comuni in modo da verificare velocemente corrispondenze e fixing delle vulnerabilità, non bloccandosi sull'analisi di una minaccia che avrà bisogno di un lavoro d'interpretazione più accurato.

Come detto nel paragrafo 2.4.2, standardizzare la condivisione delle minacce è diventato un punto fondamentale sulle agende di tutte le nazioni, cercando di ampliare lo scambio e la collaborazione tra governi e organizzazioni. Se l'obiettivo dell'implementare un processo di threat model è quello di anticipare la minaccia, è ancora più fondamentale avere a disposizione tutte le informazioni necessarie che il sistema può renderci. Indipendentemente se provenienti da fonti pubbliche o private, la rilevanza sta nel beneficio della condivisione e del lavoro di gruppo. Il lavoro individuazione-correlazione è a sua volta composto da 2 componenti tanto evidenti, quanto specifiche e dedicate per il settore trattato poiché cercano di trattare le informazioni con particolare precisione e dettaglio per la sicurezza:

- Piattaforme di condivisione
- Protocolli di scambio

Una delle soluzioni più adottate è quella di permettere un costante contatto con i database delle minacce. Questo può avvenire andando a consentire un output che va dalla fase d'analisi, verso le piattaforme con cui si cercherà di ottenere un riscontro. A tal punto è possibile distribuire queste informazioni in diversi file da poter confrontare direttamente con le piattaforme per le minacce attraverso un protocollo di scambio specifico per il formato delle informazioni che si trasmettono.

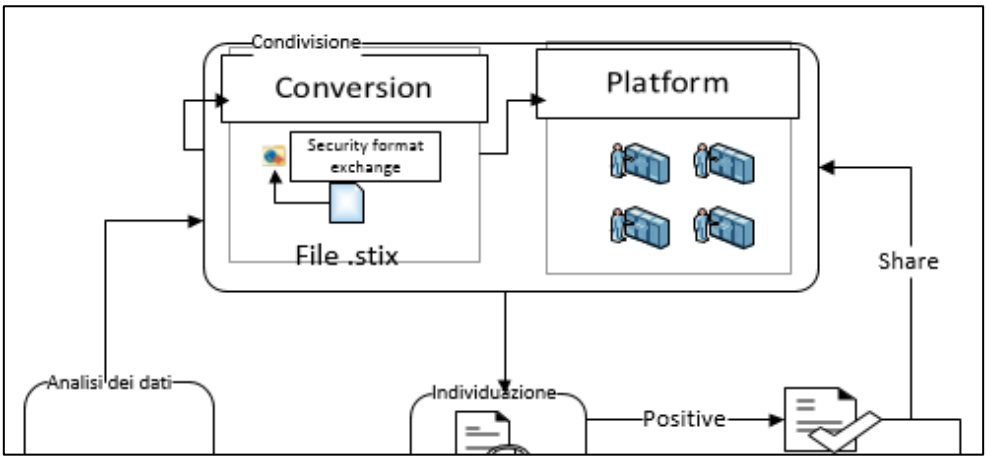


Figura 3.20 – Dialogo tra fase d'analisi, individuazione e piattaforme per le minacce

Sarà importante permettere l'elaborazione interna per la trasformazione e gestione dei dati per prepararli all'invio e alla comunicazione verso la piattaforma stabilita, in un unico formato.

Minaccia (s)conosciuta

Quando il risultato di un'analisi produce un esito negativo sia dall'intervento umano, sia dal confronto con le piattaforme per le minacce ci si ritrova in una situazione tanto ambigua quanto pericolosa:

- Presenza di un falso positivo o falso allarme
- Presenza di una minaccia non ancora identificata

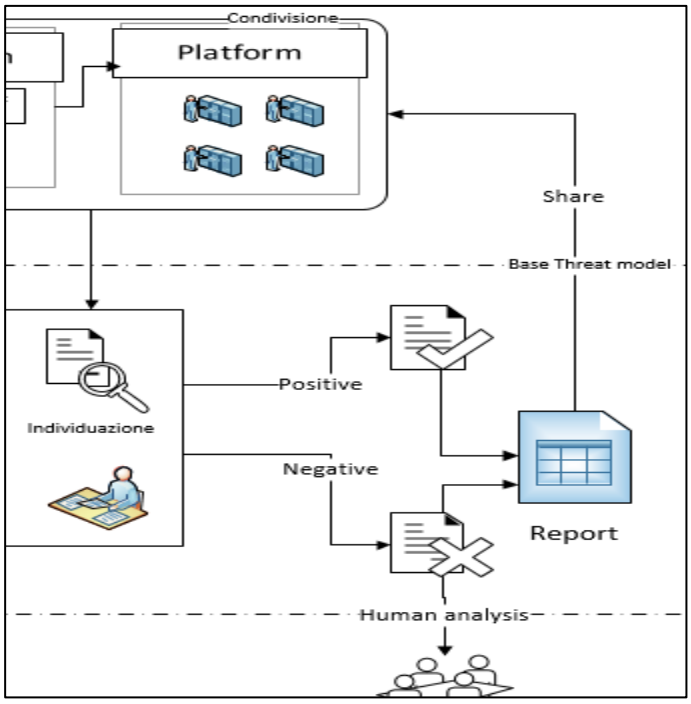


Figura 3.21 – Ambiguità sul risultato dell'analisi e individuazione

La prima situazione può essere sintomo di un eccessivo controllo sulle informazioni o, ancor prima, un'eccessiva raccolta di dati, anche da fonti non necessarie. Troppa precauzione può portare ad un alto tassi di falsi positivi ed è quindi necessario, come detto nel capitolo II, che ci sia un bilanciamento tra utilizzo di risorse e funzionalità. Garantire troppa sicurezza può dimostrarsi nocivo per il sistema. Mentre, la seconda situazione non permette di stabilire se e cosa sia andato storto nel lavoro d'identificazione. Trovarsi in presenza di un risultato negativo sia da parte del lavoro di controllo con le piattaforme, sia da parte della fase d'individuazione può far subito pensare che ci si ritrovi nella situazione di un falso allarme, ma non è permesso fare assunzioni di questo genere data l'importanza di ciò che si sta proteggendo. E' fondamentale analizzare ulteriormente i dati relativi all'allarme che ha dato esito negativo. C'è possibilità sia di "restare in attesa", andando a condividere con la rete ciò che preoccupa, sperando che la pubblicazione permetta di dare esito migliore all'identificazione, sia di utilizzare ulteriori strumenti, come il machine learning, per capire di cosa si tratta. Quindi esiste la possibilità di incontrare 2 situazioni differenti: affrontare minacce comuni, già conosciute e trattate, o incontrare una minaccia non ancora diffusa, recente. Riuscire a controllare ed estirpare anche quelle non comuni, aggiungerebbe un grado di sicurezza in più al proprio sistema.

3.5 Rappresentazione

L'approccio alla globalizzazione delle conoscenze è alla portata (quasi) di tutti, però dover accontentare le esigenze di più utenti comporta l'obbligo di sviluppare e pubblicare con uno sguardo all'usabilità. Non è da meno nell'ambito della sicurezza, ancor meno se si cerca di istruire ed abituare ad un concetto non ancora del tutto "compreso". Il livello richiesto per avere una corretta e fluida comprensione dei termini, dei meccanismi, dei processi attinenti il threat model e l'informatica più avanzata è alto! Ruoli specifici vengono riservati nelle organizzazioni per svolgere operazioni di manutenzione e monitoraggio sulle tecnologie. Gli amministratori di sistema e gli ingegneri di sicurezza ne hanno la qualifica, ma l'utente "base" avrà bisogno di più sostegno nelle operazioni. Per tanto è necessario aggiungere un particolare lavoro di adattamento, di astrazione delle informazioni elaborate nel processo di modeling. La cosiddetta dashboard ed una sua ottima progettazione è la caratteristica principale di ogni strumento per poter permettere all'utente di orchestrare le operazioni. E' importante come grafici, diagrammi, mappe e una generale correlazione delle informazioni permetta all'utente di poter comprendere cosa sta realmente accadendo nel suo sistema.

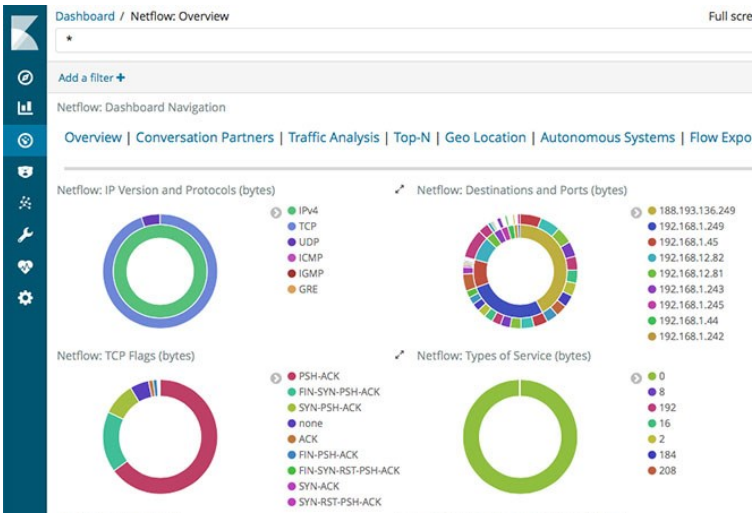


Figura 3.22 – Modello di dashboard e grafici d'analisi

Riuscire ad ottenere un risultato finale in termini di accessibilità e facilità d'utilizzo permetterà di garantire una proprietà di monitoring in tempo reale, esclusiva e focalizzata sugli aspetti toccanti la sicurezza nel processo di threat model.

La situazione del tool KAMAS (Knowledge-Assisted Visual Malware Analysis System)[20] mostra l'effettivo aiuto che comporta l'utilizzo di una dashboard per una correlazione delle

informazioni derivanti dalle analisi di minacce. Il contesto human-sensitive è particolarmente adatto per tentare anche di ridurre il carico computazionale legato alle decisioni d'intervento per arginare una minaccia. E' stato dimostrato come questo porti benefici anche in un'analisi del comportamento della minaccia e dell'attaccante. L'utilizzo di diversi colori, tipologie di grafici, permette di osservare meglio il comportamento dei dati, delle regole analizzate, delle anomalie riscontrate. La possibilità di attivare dei filtri, con funzioni di highlighting, è un fattore chiave nel focalizzare l'attenzione su un gruppo ristretto di dati.

3.5.1 Grafana

Grafana è una piattaforma che permette la creazione di una dashboard per la visualizzazione e la gestione, attraverso un'interfaccia web, dei dati scelti per l'analisi. Si tratta di un tool open-source facilmente installabile sui diversi sistemi operativi e architetture o piattaforme come Docker. Permette inoltre una perfetta integrazione con diversi plug-in come Graphite, Prometheus, ecc. a seconda delle esigenze, e grazie a cui si sta sviluppando fortemente l'appartenenza ad una community che è in costante crescita. La ricerca di una facilità d'uso e di poter introdurre nuove forme per le diverse metriche rende l'azienda costantemente competitiva, infatti sono numerosi i nomi importanti che ne hanno acquisito i benefici: PayPal, Ebay, Intel, Booking.com, ecc. sono solo alcune delle più importanti. Spesso sono comprese anche funzioni di alerting.

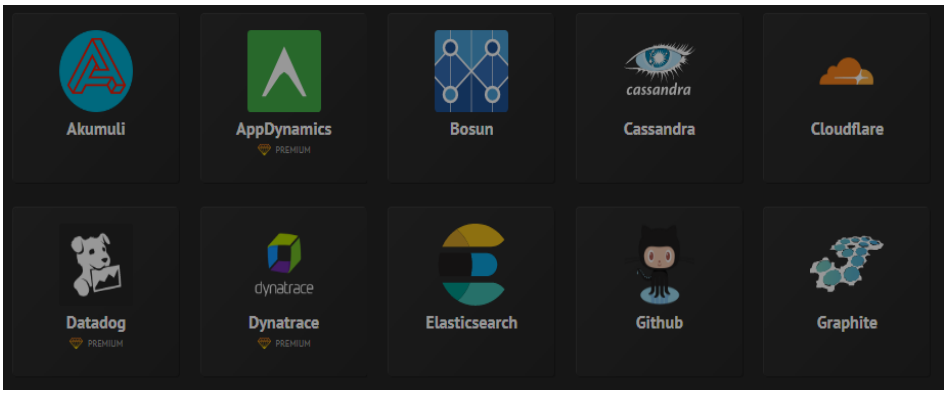


Figura 3.23 – Plugins per Grafana

Versioni

Il sito[55] permette una dimostrazione di utilizzo tramite un LiveDemo, inoltre rende chiare le versioni attraverso delle pagine dedicate:

- Grafana, la soluzione software di base, completamente open-source.
- Grafana Hosted and Cloud, ha le stesse funzioni della soluzione in locale. Per la modalità Hosted si trova una soluzione parzialmente free che non necessità di nessuna installazione, quindi si permette l'utilizzo di un'istanza ospitata direttamente da Grafana Labs, gratuita per 1 utente con l'opportunità di gestione di 5 dashboard, previa iscrizione al sito. Per la modalità Cloud l'istanza dispone da subito, dell'utilizzo di plug-in come Graphite e Prometheus, inoltre si basa sul concetto di "Pay only for what you use"
    - Questa versione è disponibile sia in formato Standard sia in formato Pro con tutte le agevolazioni e la possibilità di gestire team che amministrano moltissimi dati, con assistenza h24.

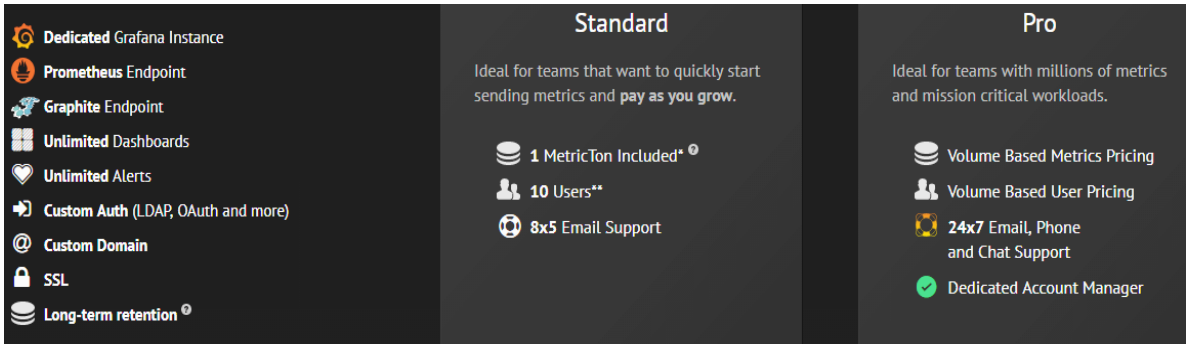


Figura 3.24 – Versioni del tool Grafana

- Grafana Enterprise, soluzione adatta ad aziende e anche Startup, con piani speciali e dedicati per il tipo di compagnia esistente. Qui le funzionalità vengono estese aggiungendo supporto immediato, training, protection, autenticazione, ecc. Il costo viene calcolato in base agli utenti attivi e alla loro categorizzazione in Editor e Viewers.

Installazione ed utilizzo

E' possibile ottenere la repository ufficiale da diverse fonti, (GitHub, Linux, ecc), ma per garantire costante aggiornamento si preferisce PackageCloud. Quindi si va ad eseguire la procedura che aggiunge la chiave GPG e la repository al sistema APT per l'installazione e aggiornamento software, e l'aggiunta del servizio come avvio automatico al boot.

[55] https://play.grafana.org/d/000000012/grafana-play-home?orgId=1

```
$     sudo add-apt-repository "deb
      https://packagecloud.io/grafana/stable/debian/ stretch main"
$     curl https://packagecloud.io/gpg.key | sudo apt-key add -
$     sudo apt-get update
$     sudo apt-get install grafana
$     sudo systemctl enable grafana-server
$     sudo systemctl start grafana-server
```

Ad eventuali problemi si ricorre con la verifica dello stato del servizio.

```
$     sudo systemctl status grafana-server
```

Ora che Grafana è stato installato basterà connettersi all'indirizzo (in locale) http://localhost:3000 . La configurazione di base di prevede che la connessione avvenga sulla porta 3000, ed imposta le credenziali d'accesso con `admin/admin`.

Figura 3.25 – Interfaccia log-in Grafana

Il primo passo verso il monitoring è quello di andare a selezionare la sorgente dei dati che può essere di 2 tipologie:

1. Supportata, ovvero le fonti da cui è possibile prendere dati, es. Graphite, Elasticsearch, CloudWatch, InfluDB, MySQL, Postgres, ecc. .
2. Da plugins, funzione disponibile dalla versione 3.0 e che quindi permette un accesso ad un ampia community[56]

[56] https://grafana.com/plugins

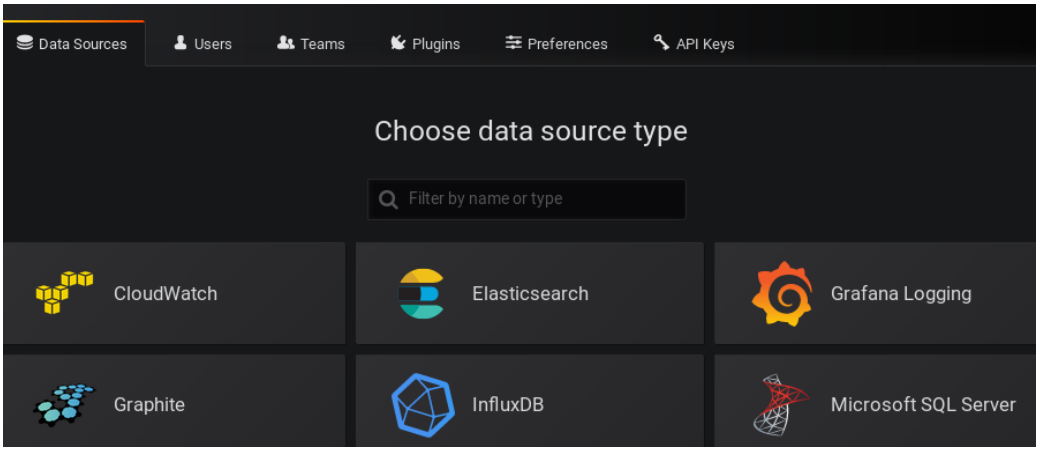


Figura 3.26 – Scelta delle sorgenti

Attraverso la barra laterale si ha acceso alle diverse funzioni:

- *Configurazione*: pannello che permette la gestione del tool attraverso la scelta delle sorgenti, delle preferenze, dei plugins e le sezioni per l'invito ad utenti esterni con la possibilità di indicare la tipologia d'utente (Editor, Viewer, Admin).
- *Dashboard*: una scorciatoia permette di raggiungere velocemente la pagina principale di monitoring, inoltre è possibile la gestione dell'interfaccia ed è permessa la funzione di snapshot.
- *Alert*: sezione dedicata alla creazione degli allarmi per la notifica di cambiamenti, anche sensibili, alle informazioni considerate nella dashboard.
- *Configurazione*: pannello che permette la gestione del tool attraverso la scelta delle sorgenti, delle preferenze, dei plugins e le sezioni per l'invito ad utenti esterni con la possibilità di indicare la tipologia d'utente (Editor, Viewer, Admin).

Plugins

Dalla versione 3.0 i plugins non sono solo strumenti per sorgenti di dati, ma anche apps e panel che permettono di creare e gestire direttamente diversi tipi di dati sulla dashboard. Dopo ogni installazione è necessario il riavvio del servizio. Attraverso il comando per terminale `grafana-cli plugins` è possibile installare, rimuovere e gestire i plugins.

HTTP API

La possibilità d'integrare le caratteristiche e i mezzi di Grafana nel nostro ambiente di lavoro è permesso attraverso l'utilizzo delle API, disponibili nella sezione Documentazione.

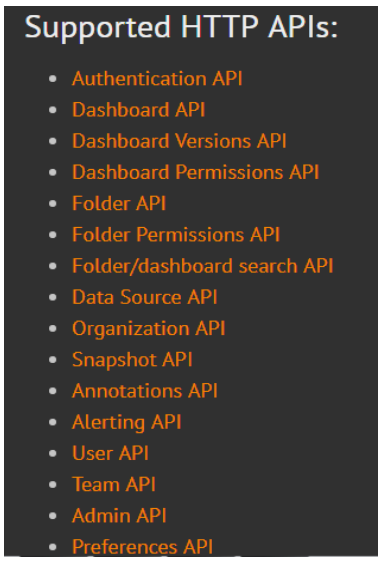


Figura 3.27 – Documentazione delle API

3.5.2 Kibana

Altro componente della famiglia di prodotti Elasticsearch e dell'ecosistema ELK (K->Kibana) che interagisce con il motore d'indicizzazione Elasticsearch per permettere la miglior rappresentazione dei dati attraverso grafici, tabelle, ecc. [21]. Anch'esso completamente open-source, scritto in JavaScript e disponibile anche come soluzione As a Service in collaborazione con AWS[57], delegando il compito della gestione delle risorse e lasciando all'utente la possibilità della costruzione della dashboard. Permette la gestione di grandi moli di dati come file di log e la creazione di script per una loro gestione dinamica, poter correlare le informazioni in entrata andando a visualizzare come coordinate su mappe[22], oppure tramite un'analisi temporale. Sarà sempre compito dell'amministratore andare a scegliere come, quando e a chi condividere i risultati ottenuti.

[57] Amazon Web Services, piattaforma per servizi di cloud……

Versioni

Come detto, il tool è disponibile gratuitamente, permettendo il funzionamento con la versione base di Kibana, ed insieme alle altre piattaforme emergenti permette di poter gestire per un periodo di prova lo strumento, ospitato in cloud direttamente dall'azienda che potrà gestirne diversi aspetti come, ad esempio, mantenere la struttura aggiornata. Esiste anche una versione per le aziende, Cloud Enterprise, che hanno bisogno di gestire molte risorse, molti utenti e quindi con bisogni di scalabilità, di backup, con possibilità di lavorare da un'unica postazione. Kibana ha anche un'ottima community che fornisce supporto agli utenti.

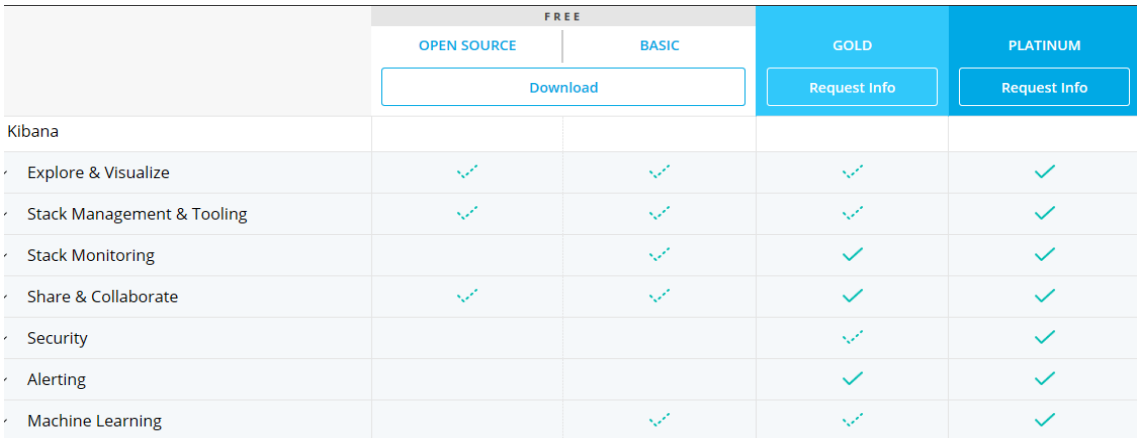


Figura 3.28 – Diverse versioni del tool Kibana

Installazione ed utilizzo

Il requisito per l'installazione di Kibana è includere Node.js, base di Kibana. Ai fini del funzionamento è consigliabile configurare il tool per adattarsi a Elasticsearch, tool della stessa famiglia. Come al solito il procedimento indicato coinvolge macchine Linux, ma lo strumento è disponibile anche per gli altri sistemi operativi:

- Scarico e installo le chiavi pubbliche

  ```
  wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add –
  ```

- Aggiungo le repository

  ```
  echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list
  ```

- Installo Kibana

  ```
  sudo apt-get install kibana
  ```

- Avvio il servizio

  ```
  sudo systemctl start kibana.service
  ```

Ora è possibile avviare l'interfaccia collegandosi via browser al link (in locale) `localhost:5601` , e da qui poi andare a verificare anche lo stato dello strumento collegandosi a `localhost:5601/api/status` .

La configurazione di tutti gli aspetti del tool avviene tramite il file `kibana.yml` andando a valutare e modificare i parametri. Alcune delle voci principali[58]:

```
#server.port: 5601
#server.host: "localhost"
#server.maxPayloadBytes: 1048576
#server.name: "your-hostname"
#elasticsearch.hosts: ["http://localhost:9200"]
#elasticsearch.username: "user"
#elasticsearch.password: "pass"
#server.ssl.enabled: false
```

Per iniziare a lavorare con Kibana bisogna permettere a quest'ultimo di poter ricevere un set di dati indicizzati e su cui poter visualizzare i dati nel modo che più si adatta alla situazione andando a comporre poi, la dashboard principale. Attraverso il menù Discover sarà possibile andare a visualizzare le rappresentazioni inserite, con la possibilità di fare analisi temporali e interrogare i dati con delle query.

[58] https://github.com/elastic/kibana/blob/master/config/kibana.yml

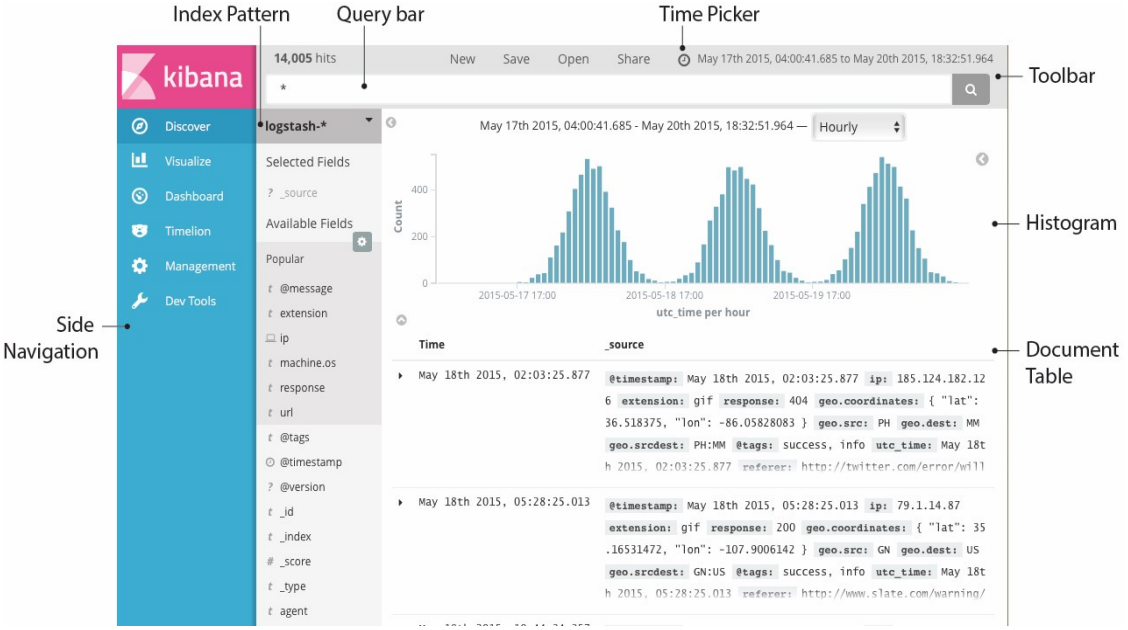


Figura 3.29 – Menù Discover, index and query bar[59]

Dalla versione 6.4 Kibana permette di attuare, dall'interfaccia principale, una breve dimostrazione di come la dashboard potrà essere popolata, ma senza caricare dati effettivi, dando la possibilità di un'interazione con il cliente davvero efficiente. Questo si integra con la possibilità di seguire tutorial, con l'aggiunta di scorciatoie per permettere l'ispezioni dei dati tramite pop-up e senza dover abbandonare la finestra.

Plugins

Attivando l'opzione a pagamento Platinum si avrà accesso alle caratteristiche più emergenti come machine learning, alerting e altro. Ognuna di queste integrazioni agisce come plugin e permettono un periodo di prova che va dalle 2 alle 4 settimane. Ecco alcune delle più fondamentali:

- *Canvas*: oltre a quanto può Kibana rappresentare in molteplici modi i dati, c'è la possibilità anche di poter personalizzare le rappresentazioni lavorando con Canvas[60] che permette di creare figure, tabelle, grafici personalizzati, oltre a poter creare soluzioni per diversi settori come l'e-commerce, analisi finanziarie, monitoring;
- *Graph*: come indicato anche sul sito del produttore, può essere molto utile andare a creare ed individuare dei collegamenti tra le informazioni, ed a tal proposito Graph[61] offre un approccio relationship-oriented. Questo può essere utile per andare ad ottenere

[59] https://www.elastic.co/guide/en/kibana/current/discover.html
[60] https://www.elastic.co/products/stack/canvas
[61] https://www.elastic.co/products/stack/graph

nuovi tipi d'informazioni dai dati, come il collegare spostamenti o particolari azioni ad un determinato utente o azienda;

- *Machine-learning*[62]: grazie a questa integrazione sarà possibile effettuare un'ulteriore correlazione tra i dati andando ad individuare, tra enormi cambiamenti e veloci scambi d'informazioni, anche le più improbabili frequenze di parole chiavi, in tempo reale.

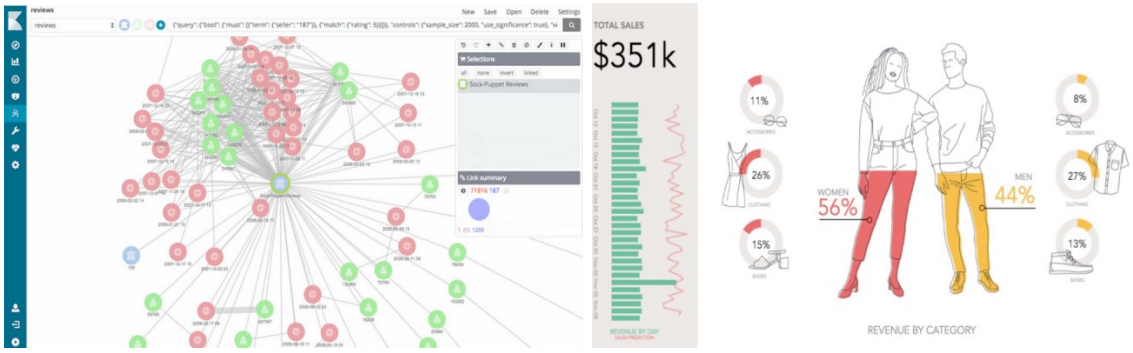


Figura 3.30 – Arricchire la dashboard con plugin come Graph o Canvas

Inoltre Kibana offre anche un set di plugins che è possibile installare tranquillamente da interfaccia di comando:

```
$     sudo –u kibana bin/kibana-plugin install x-pack
```

REST API

Kibana permette un uso delle sue funzionalità anche attraverso l'uso di REST API per andare ad implementarle nel nostro ambiente di lavoro. La disponibilità si divide in 3 categorie:

- Stable APIs, certificate e verificate per un uso sicuro esternamente dal tool Kibana;
- Beta APIs, passo precedente al diventare Stable, quindi con bisogni di diverse fasi di testing;
- Experimental APIs, ovvero tutto le altre e che possono ritrovare conflitti con ogni versione installata.

[62] https://www.elastic.co/products/stack/machine-learning

3.5.3 Confronto

Ricapitolando, le considerazioni finali mostrano le peculiarità di questi software in grado di permettere un attento monitoraggio delle risorse attraverso delle dashboard contenenti pannelli, grafici, diagrammi ed altro. Il confronto tra i 2 tool (tabella 3.x) mostra come sia opportuno il paragone con strumenti dello stesso genere per permettere una scelta più appropriata, andando ad evidenziare cosa può essere considerato "migliore" per un'implementazione ed un lavoro più efficiente. Secondo un analisi più attenta dei tool, per fare un confronto su quale si adatti meglio alle caratteristiche del threat model bisogna considerare diversi aspetti di ognuno.

Tool	Grafana	Kibana
Funzione principale	proposta generale per la visualizzazione dati	Log analysis
Multipiattaforma	√	√
Installazione	Easy	Easy
Configurazione	.ini	.yml
Global Environment	√	
Data Source	30+ tipi	Elasticsearch
Accessi e permessi	comandi built-in	X-Pack
Interrogazione	query editor con linguaggi diversi per le diverse fonti di dati	Lucene Query Parser & elasticsearch query DSL
Visualizzazione	dashboard con panel	dashboard dinamica
Alert	comandi built-in	X-Pack
Community	√ - in crescita	√
Partner e aziende	500+	1000+

Tabella 3.5 – Confronto Grafana vs Kibana

Generali

Kibana è più predisposto all'analisi di log, lavora come parte del motore Elasticsearch ed ha quindi bisogno di stabilire una connessione con quest'ultimo. Grafana invece è una soluzione più adatta ad una analisi generica dei dati e permetterne la visualizzazione in dashboard. Entrambi i tool sono multipiattaforma, Linux, Windows, Mac, Docker.

Configurazione

I tool sono di semplice facilità d'installazione. Riguardo la loro configurazione invece, Grafana permetto l'utilizzo di variabili globali e file di configurazione .ini , più semplici da gestire rispetto ai file .yml di Kibana che permettono, sì di gestire molti parametri, ma con tempistiche non proprio adeguate.

Data source

Grafana, grazie alla sua flessibilità, permette di ricevere dati da diverse fonti (più di 30), mentre Kibana è relegato ad Elasticsearch e pertanto dovrà fare affidamento alla struttura sottostante per la raccolta dati da diverse fonti.

Accessi e permessi

Grafana comprende comandi built-in per la gestione degli accessi e l'autenticazione, incluse APIs, avendo la possibilità di assegnare un ruolo diverso per ogni utente, Kibana invece ha bisogno del plug-in a pagamento X-Pack (sempre di Elasticsearch) o di soluzioni open source riadattate, altrimenti resta di pubblico accesso.

Query editor

Kibana permette di interrogare le informazioni in maniera molto efficiente, ma dopo aver indicizzato i dati. Questo può voler dire più velocità nel reperire i dati, attraverso Lucene query syntax[63], utilizzato anche da altre realtà, o elasticsearch query DSL[64]. D'altra parte Grafana usa un query editor diverso per ogni sorgente di dati, ma con altrettanto linguaggio diverso per ogni fonte di dati.

[63] Rende disponibile un linguaggio di query avanzato tramite query parser che interpreta una stringa utilizzando JavaCC, https://lucene.apache.org/core/2_9_4/queryparsersyntax.html .
[64] Basato sull'interrogazione di file JSON

Visualizzazione

In Grafana la dashboard è composta da più panel mentre Kibana adotta soluzioni più dinamiche permettendo di filtrare velocemente i dati e poter interagire dinamicamente con i grafici.

Alert

Kibana senza X-Pack non permette l'alerting, a differenza di Grafana che dalla versione 4 ne adotta le funzionalità attraverso dei comandi built-in.

Community

In termini di community si potrebbe definire che quella di Grafana è ancora in una fase di crescita – si consideri il minor numero di domande presenti su StackOverFlow[65] - mentre Kibana è in particolare vantaggio soprattutto per l'offerta di conferenze e training.

Compagnie

Kibana coinvolge più di 1000 compagnie, mentre Grafana la metà [66].

Conclusione

Concludendo, per la categoria di lavoro della rappresentazione del lavoro sull'identificazione delle minacce per la sicurezza, sembra prevalere Kibana grazie alle sue caratteristiche di log analysis. Nonostante Grafana mostri evidenti segni di crescita, sembri restare la scelta migliore per aspetti più generici l'analisi.

3.6 Big data, cloud e scalabilità

Relativamente alla fase di raccolta dati e all'associamento di caratteristiche di intelligence e machine learning al processo di threat model, viene affiancata la possibilità di un over-compensamento delle informazioni. Se poi ci si sposta da sistemi relativamente semplici come in ambito locale o di piccole imprese, verso realtà più grandi come organizzazioni, grandi aziende, servizi nazionali o banche ci sarà bisogno di necessità nettamente superiori.

[65] https://stackoverflow.com/
[66] https://stackshare.io/stackups/grafana-vs-kibana

Le differenze sono evidenti: i "macro-ambienti" devono gestire situazioni particolari dovute al particolare tipo di servizio che offrono, ma alla base si trovano sempre rappresentazioni del sistema che evidenziano la presenza di un elevato numero di macchine, server, firewall, ecc. . Tutto ciò si traduce in un'elevata quantità d'informazioni prodotte, scambiate, gestite ogni giorno, facendo intuire molto del peso computazionale che ricade sul sistema stesso e sul processo di threat model per gestire tali informazioni e poterle analizzare. Questo, senza considerare il fattore "falso allarme" che incide molto in termini di efficienza del threat model.

La formazione dei BigData nasce proprio da queste situazioni, ovvero dall'eccessiva produzione di informazioni dovute a sistemi sempre più grandi con necessità di gestire più utenti.

3.6.1 Apache HADOOP

Gli Apache Project sono davvero numerosi e toccano diverse aree dell'IT. Quando c'è bisogno di dover gestire elevate operazioni su grandi moli di dati, in particolare su sistemi distribuiti composti da molti nodi, non si può non nominare Hadoop, soluzione open source scritta in Java ed elaborata per lo scopo. Anche grazie a Java è presente una community molto espansa che ha portato ad uno sviluppo efficiente del progetto, permettendo di essere adottato da importanti nomi ne settore come Facebook, IBM, Linkedin, Spotify, Yahoo.

La suddivisione dell'elaborazione dei dati su più nodi riduce i tempi d'accesso e rende subito disponibili i dati. Il principale componente è Hadoop Common, che permette l'avvio del tool e fornisce accesso al filesystem supportato da Hadoop. Un cluster Hadoop è composto da:

- un NameNode, su cui risiedono i metadati dei file
- dei DataNode, su cui risiedono i file dell'HDFS in blocchi
- un CheckPointNode, che sincronizza i cambiamenti aggiungendoli all'ultima cattura del sistema
- un BackupNode, sempre sincronizzato col NameNode.

Alcune delle applicazioni usate nel cluster sono:

- Motori di esecuzione
    - Apache Spark, più veloce di MapReduce;
    - Pig, una piattaforma di alto livello per creare programmi MapReduce
- Storage

- Apache Hive, per interrogare il file system con Hive Query Language, simile ad SQL;
- Apache Impala, più veloce di Hive, ma con Impala Query Language;
- HBase, una base dati distribuita;
- Sqoop, importa dati da database relazionali verso HDFS e vice versa tramite riga comando;

- Console web
  - Apache Ambari, semplice gestione del cluster;
  - Hue, interfaccia per gli strumenti di Hadoop;

L'architettura di Hadoop permette quindi l'integrazione di molti strumenti come anche Apache ZooKeeper, servizio per la configurazione e sincronizzazione, o Apache Flume, maggiormente indicato per trattare dati di log, e molti altri!

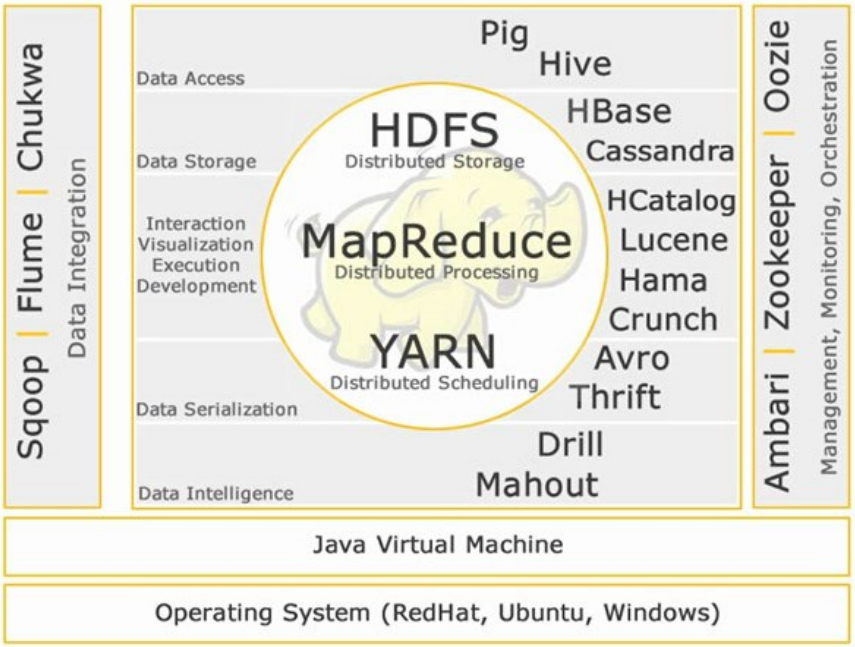


Figura 3.31 – Ecosistema Hadoop[67]

Versioni

In base alle esigenze di prestazioni e dell'avanzare delle tecnologie, si sono sviluppate diverse versioni di Hadoop, con caratteristiche quindi incrementali:

[67] https://www.edupristine.com/blog/hadoop-ecosystem-and-components

1. La prima utilizza l'algoritmo Map Reduce e Hadoop Distributed File System (HDFS), scritto in Java e adatto a garantire portabilità e scalabilità;
2. La seconda versione implementa l'architettura YARN;
3. Le successive variano in base alle necessità d'efficienza.

Creare il proprio ambiente per Hadoop, il proprio ecosistema per la gestione dei BigData, è permesso in primis grazie alla vastità di strumenti open source integrabili con Hadoop, anche se sono disponibili diverse distribuzioni (sempre open source!) in cui è presente già tutto l'ecosistema configurato per lo scopo della distribuzione. Le più importanti sono:

- Cloudera, che include i principali pacchetti per Hadoop senza Ambari;
- Hortonworks, con Ambari.

Installazione ed utilizzo

Come consigliato da diverse fonti[68], conviene andare a separare l'installazione di Hadoop dagli altri software andando a permettere al nuovo utente di connettersi ad Hadoop con *ssh*.

```
$     sudo useradd –s /bin/bash –m –p hadoop hadoop
$     su – hadoop

$     ssh-keygen –t rsa –P “”
$     cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
$     ssh localhost
```

Una volta completato sarà possibile scaricare[69] Hadoop ed installarlo.

```
$     sudo tar xzf hadoop-3.1.2.tar.gz
```

Successivamente bisognerà andare a settare il file `~/.bashrc` per le variabili d'ambiente, andando ad aggiungere le seguenti righe

```
#Set HADOOP_HOME
export HADOOP_HOME=$HOME/hadoop-3.1.2
# Add bin/ directory of Hadoop to PATH
export PATH=$PATH:$HADOOP_HOME/bin
```

e concludendo con il commando `source ~/.bashrc .`

[68] https://www.html.it/pag/52466/installazione-di-hadoop-su-linux/
[69] https://hadoop.apache.org/releases.html

Per la configurazione bisognerebbe andare nella directory principale di Hadoop `$HADOOP_HOME/etc/hadoop` , cambiare i parametri nei file:

- `core-site.xml` , con informazioni su checkpoint e filesystem
- `mapred-site.xml` , per la configurazione di MapReduce
- `hdfs-site.xml` , con informazioni sull'uso di security, salvataggio della tabella dei nomi

Dopodiché è necessario andare a formattare il filesystem con

```
$ hadoop namenode -format .
```

Finalmente sarà possibile avviare il sistema tramite terminale con `$ $HADOOP_HOME/sbin/start-all.sh` oppure da interfaccia-web http://localhost:50070 .

Per un controllo sullo stato d'attività dei servizi si utilizza il comando `jps` . Ora non resta che andare a dialogare con il filesystem attraverso terminale o con Web API. La cartella principale è disponibile con il comando `hdfs dfs -ls /` poi, con i comandi `-put -get` si può andare ad inserire o richiedere dalla directory indicata.

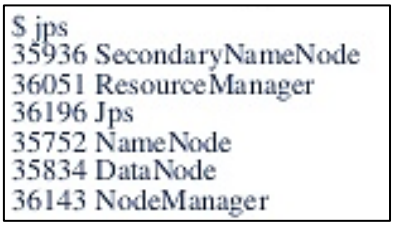


Figura 3.32 – Comando jps

3.6.2 Analisi per il threat model process

Incorrere in problemi e anomalie in questa fase, rischia di causare perdite di dati o ridurre le velocità di trasferimento. Dalla lettura di diversi articoli accademici viene messo in mostra come si possa ricorrere a particolari caratteristiche di Hadoop per andare a garantire un livello maggiore di sicurezza. Ad esempio, nella lettura di [23][24][25] viene dimostrato come si possa andare ad ostacolare attacchi di tipo DDoS con MapReduce. Diverse tipologie di algoritmi e

soluzioni mostrano come si sia già entrati abbondantemente in un'ottica di prevenzione e di come questa caratteristica non manchi neanche nell'ambito della gestione dei dati. Inoltre, è stato detto in precedenza come Hadoop permetta, tramite un lavoro d'integrazione, di aggiungere diversi componenti all'ecosistema o, in alternativa, come sia possibile reperire distribuzioni con l'ecosistema già "arricchito". Un esempio è Hortonworks: si introduce ad un controllo real-time sulle minacce tramite un approccio basato su analisi dei comportamenti delle minacce e l'utilizzo di machine learning. Altra versione particolarmente adatta per il modeling di minacce in ambito di BigData è Apache Metron[26], soluzione focalizzata sull'efficienza nell'elaborare dati e correlarli con strumenti per la threat intelligence, il tutto gestito in formato STIX. Inoltre permette di poter variare la configurazione senza dover riavviare per applicare le modifiche. Tramite un attento insieme di passi e l'implementazione della piattaforma Hortonworks Cybersecurity, permette di poter gestire con precisione l'avanzamento della minaccia.

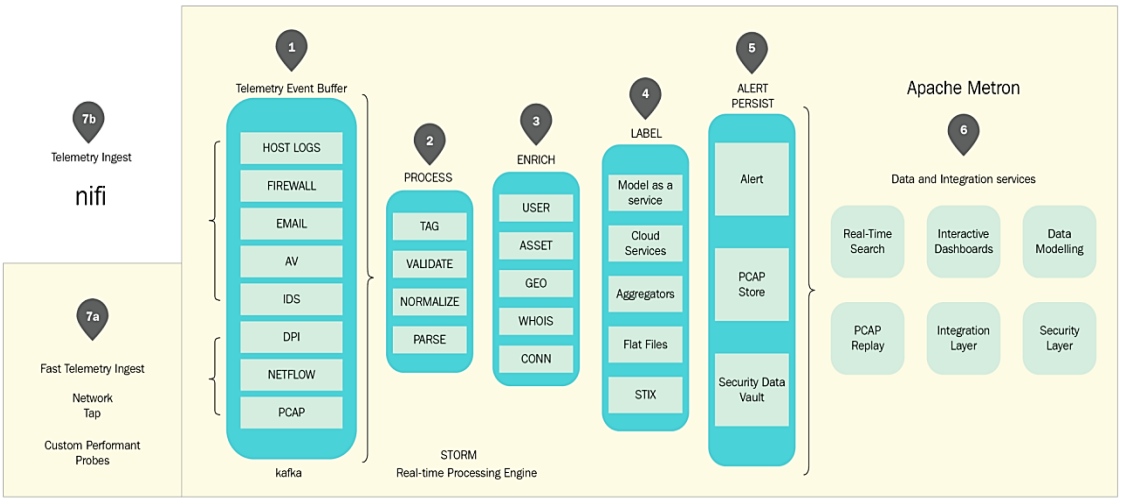


Figura 3.33 – Apache Metron e HCP[26]

3.7 Container e isolamento

Principalmente collegato al concetto di Cloud, As a Service[70] e BigData, la scalabilità diventa un principio fondamentale quando si sta parlando di grosse realtà (si vedi la struttura di grandi piattaforme come Google). Si cerca di migliorare l'efficienza attraverso una distribuzione

[70] Per il cloud computing, ovvero la gestione di una o più risorse informatiche erogate via Web permettendo di evitare l'acquisto software, hardware, ecc..

uniforme sia delle risorse sia dei servizi per adattare e bilanciare le necessità dovute all'incremento di utenti.

Di recente, il concetto di virtualizzazione presente nel sistema Linux, LXC, è diventato di spunto per il rivoluzionario avanzamento dei Container (e di Docker). Il concetto di container si basa molto sulla metafora presente nel nome stesso: permette un particolare tipo di virtualizzazione che figurativamente è molto simile alle navi cargo per il trasporto di container.

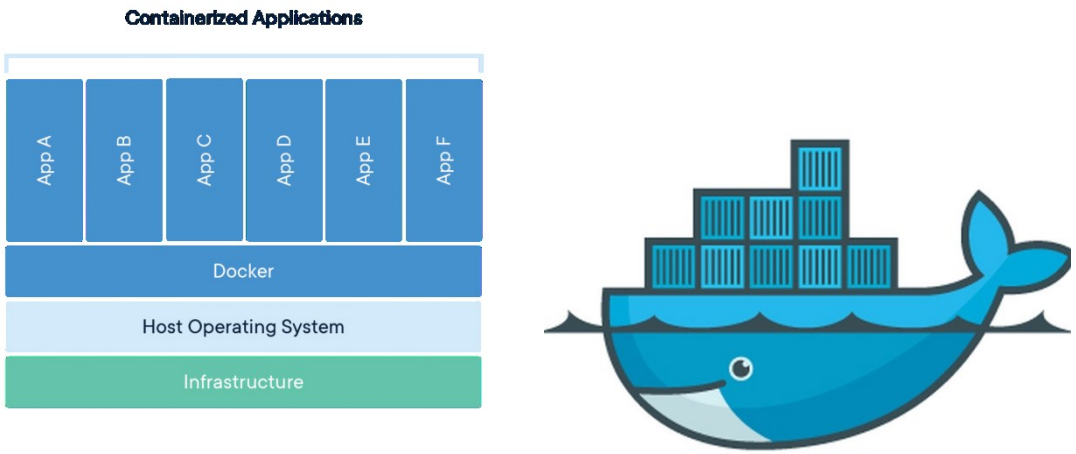


Figura 3.34 – Struttura della containerizzazione[71]

L'infrastruttura alla base e il sistema operativo sono unici (la nave), mentre la variabile è l'ambiente di lavoro (i container), molto simili ai processi attivi nei sistemi operativi e al multithreading. La scelta tra VM e container comporta delle differenze sostanziali[27]. Adottare una VM richiede dover riservare spazio sul disco, quantità di banda e risorse, e se si pensi alla necessità di avere più di una VM, il risultato è davvero svantaggioso e complicato. L'approccio ai container, invece, permette un livello di astrazione differente dalle VM: c'è la possibilità di avere alla base sempre il kernel del S.O. andando a costruire ogni container sopra di esso. Questa situazione permette di non andare a dedicare risorse per avere un sistema operativo su ogni virtualizzazione, ma andare a lavorare come processi stessi del S.O., permettendo un migliore avvio dovuto al non dover ricreare ogni volta l'intero ambiente operativo. Inoltre, per la specifica situazione del monitoraggio dei comportamenti degli ambienti creati, c'è la possibilità di archiviare le informazioni in file diversi[28].

[71] https://www.docker.com/resources/what-container

Ragionando molto su questo scenario, si provi a pensare a quelli che potrebbero essere i benefici in una situazione come il modellamento delle minacce, attraverso un approccio più:

1. Isolante
2. Prestante
3. Efficiente
4. Personalizzabile

L'idea, il tormento, nasce da come ottenere proprio Queste migliorie. L'analisi ha portato ad individuare la containerizzazione come soluzione più adeguata. La possibilità di poter diminuire il carico computazionale del processo di modeling dovuto all'ambiente "container" si può tradurre nella possibilità di adottare diversi tool, diversi strumenti da poter sfruttare per ottenere maggiore sicurezza dal processo, evitando conflitti dovuti all'installazione diretta in un unico ambiente operativo, mentre qui i container fanno da isolanti. Proprio l'isolamento avrebbe permesso di poter sia testare un nuovo tool, sia osservare la minaccia stessa, anche dopo un eventuale successo, situazione molto simile all'utilizzo di un honeypot.

3.7.1 Docker

Docker è uno strumento open-source per lo sviluppo di soluzioni applicative indipendenti e la creazione di ambienti distribuiti, attraverso il concetto di container e della coesistenza che possono avere sulla stessa istanza del SO. Questo avviene isolando le risorse del kernel, ad esempio come in Linux con cgroups e namespaces: i namespaces isolano ciò che si può vedere dell'ambiente operativo, mentre i cgroups forniscono l'isolamento delle risorse. Questa separazione dal SO garantisce tempistiche decisamente più brevi, non dovendo riavviare un sistema operativo ogni volta che si aggiunge un container.

I container lavorano con le immagini, paragonabili a dei pacchetti contenenti il codice necessario ad eseguire determinati applicativi, SO, ecc. . Ogni file immagine in Docker è composto da più strati, creati ad ogni modifica dell'immagine, es. ogni volta in cui un utente specifica un comando. Si riutilizzano questi strati per velocizzare il processo di creazione dei container. Le immagini e gli strati possono essere condivisi tra container (figura 3.35), migliorando ulteriormente l'efficienza. In poche parole si crea un registro delle modifiche dando la possibilità di tornare ad una versione precedente.

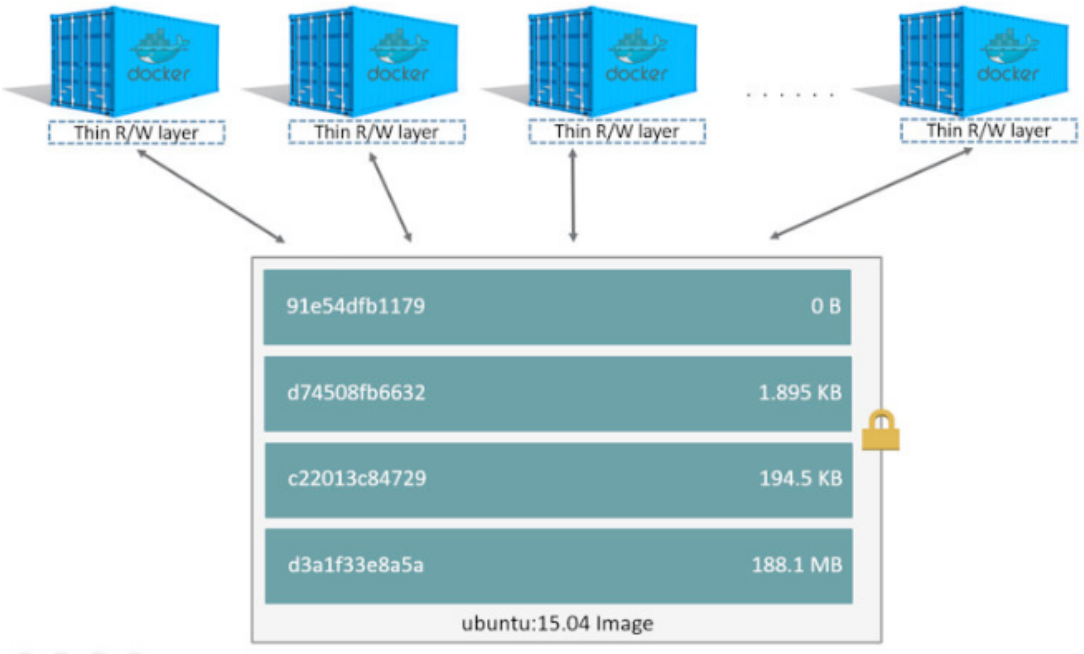


Figura 3.35 – Stratificazione di un'immagine Docker

Purtroppo nel momento in cui il kernel dell'host viene condiviso con i container viene a galla una vulnerabilità che comporta uno svantaggio in termini di sicurezza in questi sottosistemi leggeri. Ciò non si verifica nel caso delle macchine virtuali che sono molto più isolate dal sistema host.

L'architettura di Docker è composta da un client che comunica attraverso REST API con un demone, motore della creazione e gestione dei container. La fonte pubblica principale da cui vengono scaricate le immagini è Docker Hub.

Quando con il comando `docker run` o `docker pull` si va a richiedere l'utilizzo di una immagine, poi queste vengono salvate in un registro della macchina locale.

Versioni

La versione principale è quella open-source, Docker CE, accessibile ad ogni tipologia d'utente ed ideale per un approccio iniziale alla tecnologia. A sua volta ogni versione è suddivisa in base alla release richiesta:

- Stable, versione rilasciata e già adottata dalla community
- Test, versione ancora in fase di testing, quindi pre-rilascio

- Nightly, versione con i lavoro in corso per la prossima release

Docker Enterprise, invece, non è altro che una rivisitazione (in meglio) della versione CE, mantenendo il codice privato. Sviluppato principalmente per la gestione di processi critici e la scalabilità all'interno delle aziende.

Capabilities	Docker Engine - Community	Docker Engine - Enterprise	Docker Enterprise
Container engine and built in orchestration, networking, security	✓	✓	✓
Certified infrastructure, plugins and ISV containers		✓	✓
Image management			✓
Container app management			✓
Image security scanning			✓

Figura 3.36 – Confronto versioni Docker[72]

A partire dalla versione 0.9, Docker include la libreria libcontainer per poter utilizzare direttamente le funzionalità di virtualizzazione del kernel Linux, in aggiunta alle interfacce di virtualizzazione come libvirt, LXC e systemd-nspawn. La community open source cerca di migliorare tutto l'ambiente dei container. Successivamente Docker Inc., l'azienda, sviluppa i progetti della community offrendo soluzioni migliorate, anche di livello enterprise.

Installazione ed utilizzo

Bisogna seguire molto attentamente diversi passi per completare l'installazione di Docker. Partendo dalla routine per l'aggiunta di Docker repository con i comandi,

```
$ sudo apt-get install docker-ce docker-ce-cli containered.io
```

si conclude col settaggio di privilegi per non richiamare sempre il comando `sudo` .

```
$ sudo groupadd docker
$ sudo usermod –aG docker $USER
```

72

Dopo l'installazione base è consigliato andare ad aggiungere *docker-compose* e *docker-machine*, 2 strumenti che permettono di creare diversi container sia in locale che su cloud, o andare ad installare su altri nodi l'ambiente Docker.

```
$ base=https://github.com/docker/machine/releases/download/v0.16.0 && curl -L $base/docker-machine-$(uname -s)-$(uname -m) >/tmp/docker-machine && sudo install /tmp/docker-machine /usr/local/bin/docker-machine

$ sudo curl -L "https://github.com/docker/compose/releases/download/1.23.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

$ sudo chmod +x /usr/local/bin/docker-compose
```

Ora è possibile avviare la coppia container-immagini.

```
$ docker container run --publish 80:80 nginx
```

Quello che avviene a basso livello è la ricerca dell'immagine nel proprio registro, oppure prendendola da Docker Hub, e iniziare ad eseguirla in un container con una propria interfaccia di rete. In questo caso il container creato sarà in ascolto sulla porta 80 e potrà ricevere dalla porta 80 della nostra macchina, con l'assegnamento di un indirizzo IP per la rete creata da Docker.

E' possibile andare a definire delle impostazioni, delle configurazioni per i container attraverso *Dockerfile,* gestito con un proprio linguaggio. Quelle in figura 3.37 sono solo alcune opzione da poter inserire per comporre e gestire la propria versione dell'immagine.

```
FROM ubuntu:15.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

Figura 3.37 – Dockerfile sample[73]

Con il file docker-compose.yml invece, si definiscono i servizi per quando si compila un'applicazione su più container. Permette di settare porte, reti, volumi, variabili, ecc. .

[73] https://docs.docker.com/develop/develop-images/dockerfile_best-practices/

Figura 3.38 – Esempio configurazione docker-compose.yml[74]

Docker non dispone nativamente delle capacità di usare processi come cron o syslog all'interno del container, ma possono essere attivate modificando il file di configurazione per poterle usare sin dagli inizi, cosa non immediatamente ovvia.

Plugins

C'è l'opzione di poter estendere le funzionalità di Docker con plugins, attraverso il semplice comando

```
$     docker plugin install
```

Questi si presentano come delle immagini. E' possibile trovare plugin per la gestione dei volumi, andando a permettere la condivisione tra più host. Attualmente sono disponibili 3 tipi di plugin:

1. Volume
2. Network
3. Authorization

API

Sono le API che permettono l'interazione tra Docker e il Docker Engine, e sono sempre le API coloro che configurano comandi, ad eccezione dell'avviamento dei container per cui sono disponibili diverse soluzioni. Ad ogni release sono rilasciate diverse API.

[74] https://docs.docker.com/compose/overview/

3.7.2 Kubernetes

Per far interagire più container Docker tra di loro, spesso si utilizzano dei software di orchestrazione, come ad esempio Docker Swarm o Kubernetes. Esistono limitazioni all'uso di Docker poiché un numero sempre maggiore di container comporta un aumento della complessità nella gestione. Si cerca di raggruppare i container per fornire servizi (tra cui i servizi di rete, sicurezza e telemetria) a tutti i container utilizzati, ma non solo: quando il livello di affidabilità e disponibilità ricercato per il servizio deve essere elevato (soluzioni ad altà disponibilità), è qui che entra in gioco Kubernetes. La sua struttura si articola in:

- Pods, elemento che raggruppa container che condividono risorse ad esso legato, come definire un volume e metterlo a disposizione a tutti nel pod, gestibile con API
- Label, per assegnare coppie di valori agli elementi
- Services, insieme di pod
- Kubectl, comando da terminale per configurare Kubernetes

Il nodo master riceve e trasmette le istruzioni ai nodi, che possono essere macchine fisiche o virtuali, permettendo di stabilire automaticamente quale nodo è più adatto. A quel punto Kubernetes, che interagisce con i pod, assegna loro le risorse per l'attività che svolgono. Il controllo sui container avviene a un livello superiore, quindi strutturalmente la differenza dall'utilizzo esclusivo di Docker è minima.

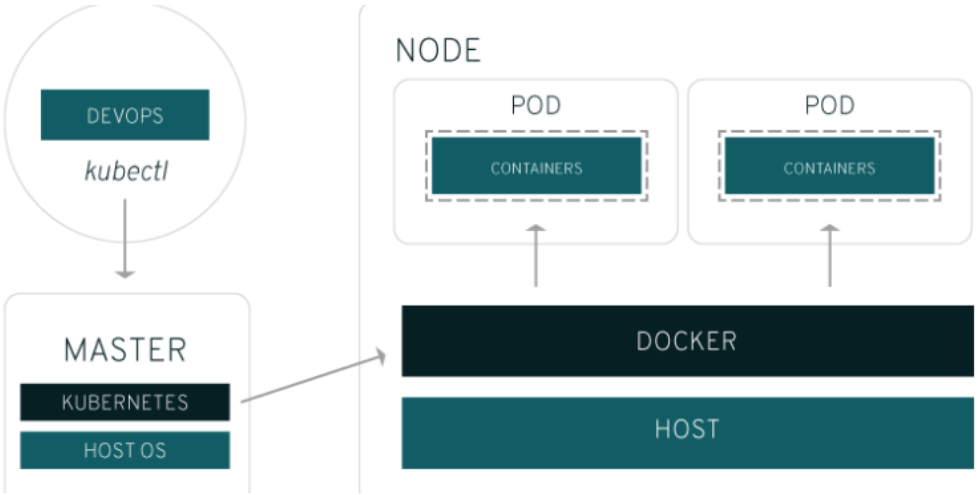


Figura 3.39 – Struttura Kubernetes e integrazione con Docker

Kubelet, che monitora continuamente lo stato dei container, chiede a Docker di lanciare i container indicati, altrimenti sarebbe un amministratore a eseguire manualmente le operazioni

per i container. Proprio questo alleggerimento è stato davvero apprezzato dalle diverse aziende che hanno fatto di Kubernetes e la containerizzazione un passo necessario: Ebay, Huawei, ING, IBM, BlaBlaCar, Philips, Zalando e molti altri ancora.

Versioni

Inizialmente definito come progetto "Borg" di Google, nel 2015 viene rilasciata la prima versione. Dopodiché verranno stabilite delle collaborazioni che andranno a formare la Cloud Native Computing Foundation, per dare vita ad uno standard aperto. Esiste anche una versione "ridotta" che permette di andare a sperimentare l'utilizzo di Kubernetes, ovvero *minikube*, adatta per la gestione (in locale) di un singolo cluster. Naturalmente sia Google, che Amazon, che altri servizi cloud rendono disponibile questa tecnologia all'interno delle loro istanze per una migliore gestione.

Installazione ed utilizzo

Prima di eseguire l'installazione conviene sempre considerare ciò di cui si ha bisogno per capire se andare a sviluppare in locale, in remoto, sul cloud, ecc. La documentazione ufficiale sul sito offre a tal proposito, una sezione[75] dedicata alla guida per la scelta della soluzione ideale. In questo caso la scelta, per semplicità, ricadrà sulla versione *minikube*. Di base bisognerà:

1. Utilizzare la virtualizzazione con VirtualBox o simili, o in alternativa è possibile evitare la virtualizzazione con l'opzione `--vm-driver=none`, ma solo utilizzando Linux e Docker
2. Installare kubectl, per il controllo del cluster Kubernetes

```
$ sudo apt-get update && sudo apt-get install -y apt-transport-https
$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
$ echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a /etc/apt/sources.list.d/kubernetes.list
$ sudo apt-get update
$ sudo apt-get install -y kubectl
$ chmod +x ./kubectl
$ sudo mv ./kubectl /usr/local/bin/kubectl
```

[75] https://kubernetes.io/docs/setup/pick-right-solution/

3. Installare minikube, includendo la creazione di un file di configurazione per kubectl

   ```
   $ curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 \ && chmod +x minikube && sudo mv minikube /usr/local/bin
   ```

Completate queste operazioni si avrà a disposizione l'intero ambiente semplicemente con il comando `minikube start [--vm-driver=none]`.

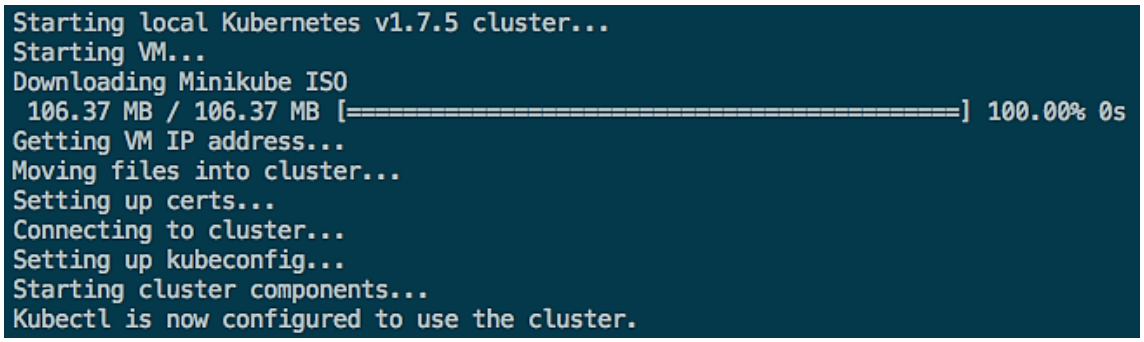


Figura 3.40 – Avvio di minikube e Kubernetes

Attraverso diversi comandi è possibile andare ad impostare immagini, esporre porte, settare parametri, esattamente come con Docker.

```
& kubectl run tomcat-deployment --image=tomcat-deployment --port=8080
& kubectl expose deployment tomcat-deployment --type=NodePort
```

Questi ultimi 2 comandi permettono di andare a istanziare l'immagine sample hello-minikube e ad esporla alla rete esterna. Con il comando

```
& curl $(minikube service tomcat-deployment --url)
```

vado poi ad accedere al cluster. Da qui è possibile utilizzare altri comandi:

- Per individuare i pod attivi

  ```
  & kubectl get pods
  ```

- Per mostrare tutte le informazioni su pod

  ```
  & kubectl describe pod
  ```

- Per eseguire comandi all'interno di un container

  ```
  & kubectl exec –it <nome completo pod> bash
  ```

API

Esistono diverse versioni di API con diversi livelli di stabilità:

- Stable
- Alpha
- Beta, da non utilizzare in ambiti aziendali e critici

Fondamentalmente si parla di RESTful API adatte per la gestione di oggetti. Accettando JSON si permette anche di rilevare la particolare versione delle API adottata. I termini che si ritrovano sono questi:

- Kinds, top-level API, presenti 3 categorie:
    - Objects, per indicare l'intenzione di creare un'entità nel sistema e mettere a disposizione dell'utente specifiche risorse, es. *Pod, Service, Namespace*
    - Lists, collezioni di Kinds presenti nel campo *items* , con il termine list alla fine del nome, es. *PodLists, NodeLists*
    - Simple, kinds non-persistenti e per specifiche azioni, es. *Status*
- Resources, per elaborare un oggetto, i risultati JSON devono avere un campo *kind* per indicare lo schema dell'oggetto, e un campo *apiVersion* per la versione dello schema
- Objects, elementi con specifici metadati associati
    - MUST HAVE
        - Name, stringa per identificare l'oggetto nel namespace
        - Namespace, livello in cui si suddividono gli oggetti
        - Uid, identificativo basato su tempo e spazio, per differenziare con oggetti con nome uguale

3.7.3 Analisi per il threat model process

Docker vs Kubernetes, più che di un confronto, qui si sta parlando di 2 strumenti complementari. Uno può funzionare senza l'altro, ma è insieme che acquisiscono un'elevata efficienza. Scendendo in profondità si potrebbe confrontare la controparte di Kubernetes in Docker, ovvero Docker Swarm, ma in realtà anche l'azienda stessa ha intuito come Kubernetes permetta una gestione efficiente dei container, e si sta muovendo proprio in tal senso per rendere Swarm più efficiente[29]. Sotto l'aspetto della sicurezza venga da pensare come i container possano davvero essere una luce emergente per dare ancora più resistenza ed efficienza ad un processo di modellamento delle minacce. Basti pensare a come i container costituiscano una forma più evoluta di sand-box, e quindi di isolamento, dove sia possibile condurre test controllati sulle

vulnerabilità incontrate nel processo per le minacce. D'altra parte, seppur in letteratura esistano diverse soluzioni all-in-one (paragrafo 3.10) per andare ad implementare un threat model process, almeno basilare, esiste anche la possibilità di andare a creare e personalizzare l'ambiente secondo le proprie esigenze. Ebbene, la containerizzazione potrebbe fare proprio a questo scopo in quanto garantire isolamento non vuol dire che non sia offerta anche la possibilità di comunicazione, infatti fa pensare alla possibilità di andare a implementare diversi tool, con diverse caratteristiche di analisi che mi permettano di vedere da diversi punti di vista le vulnerabilità e le minacce individuate. Questa è stata un'idea cresciuta durante il lavoro di stesura dell'elaborato che ha ritrovato particolare interesse nei professori. Seppur in letteratura non sia ancora presente questa tipologia di struttura, è comunque possibile ritrovare diverse iniziative che coinvolgono i container nel threat model. Il particolare caso di adattamento di Chromium alla tecnologia Docker [30] mostra come la struttura di base utilizzante sandbox, mostri diverse falle derivanti proprio dagli svantaggi delle di queste ultime, come il fatto che essendo eseguita senza privilegi è possibile renderla vulnerabile se il controllo del processo viene attaccato. O anche come sia possibile ridurre le tempistiche di servizi DNS in server, con l'utilizzo di una virtualizzazione ibrida che utilizzi i container come parte per aumentare la velocità di servizio del server e per la richiesta di un minor numero di risorse[31].

3.8 Sistemi all-in-one

Finora la direzione indicata ha comportato l'analisi e il confronto di diversi tool per ognuna delle fasi del threat modeling. Scegliere ed implementare un tool per ognuna delle fasi permette di creare un ecosistema che più si adatta alle esigenze richieste per il processo per le minacce. La possibilità di andare ad implementare il processo in maniera modulare, permette un intercambiabilità degli strumenti in base all'avanzare delle tecnologie o ad una diversa scelta in fase di progettazione. Questo però richiede un lavoro molto esaustivo da diversi punti di vista:

- La scelta del tool
- L'installazione e configurazione del tool
- Implementazione, collaborazione e compatibilità con gli altri tool scelti per le altri fasi

L'alternativa alla scelta di singoli tool, è quella di andare ad implementare soluzioni già pre-impostate, complete, ecosistemi già collaudati. Le soluzioni commerciali non mancano, così come quelle open source, ed è proprio queste ultime che ritrovano ampio svago nel caso di sistemi all-in-one, soprattutto dovuto alla pubblicazione di diverse personalizzazioni studiate

dagli sviluppatori[76][77][78]. Purtroppo è risaputo come la scelta di soluzioni open source comporti svantaggi da parte di grandi aziende che si ritrovano con una tecnologia dotata di un supporto strutturato in maniera formale a cui affidare il suo business: eventuali problemi metterebbero in difficoltà l'azienda. Queste tipologie di soluzioni all-in-one spesso vengono accostate al termine Network Security Manager (NSM), ovvero piattaforme che devono fornire diverse funzionalità per il controllo.

3.8.1 Security Onion

Un esempio di piattaforma che ingloba caratteristiche di HIDS, NIDS, analisi, ecc. è Security Onion. La gestione della piattaforma viene orchestrata con diversi strumenti integrati, mentre attraverso l'uso di sensori è possibile poi andare a raccogliere diverse tipologie di dati per inviarli successivamente ai tool d'analisi. Quindi, un insieme di strumenti che favoriscono lo sviluppo del concetto di modeling delle minacce, ma con soluzioni open source o low cost. Riepilogando, Security Onion riesce ad implementare le necessità di un processo per il modellamento di minacce[32]:

- Raccolta dati, la possibilità di andare ad utilizzare soluzioni ibride per NIDS e HIDS permette di coprire un ampio raggio di dati e di salvarli in file di log
- Analisi dei dati, gestita dall'integrazione di diverse soluzioni software per lo scopo o attraverso la possibilità di effettuare un'analisi ancora più dettagliata attraverso tool forensi
- Gestione della struttura, attraverso comandi da terminale o, più comodamente, con interfacce utente
- Alerting
- Scalabilità, garantita dall'eventuale configurazione attraverso sensori multipli e un master server
- Accessibilità, come detto precedentemente, la presenza di interfacce grafiche fornite da tool permette l'interrogazione e il filtraggio dei dati raccolti
- Sicurezza, gestione delle comunicazioni con sensori, server, interfacce, tool, ecc. attraverso SSH

[76] https://github.com/Cyb3rWard0g/HELK
[77] https://github.com/silascutler/MalPipe

[78] https://github.com/RookLabs/milano

Un dato importante riguarda i requisiti che tale strumento richiede per poter funzionare attivamente, es.: su un rete da 50MB/s si può stimare un traffico di 16 Terabyte in 30 giorni derivante dall'utilizzo della piattaforma.

Versioni

Per garantire queste caratteristiche, Security Onion ingloba diversi tool in una versione live della distribuzione Xubuntu, scaricabile dal sito [79], disponibili solo per processori con architettura a 64-bit.

Installazione ed utilizzo

Dovendo scaricare un immagine ISO del sistema si procede al download con il comando

```
$ wget https://github.com/Security-Onion-Solutions/security-onion/releases/download/v16.04.5.6_20190110/securityonion-16.04.5.6.iso
```

per poi andare a montare l'immagine, volendo sul sistema operativo, su USB o con VM. Completata la procedura, come in tutte le distribuzioni live, ci sarà modo di poter scegliere tra diverse opzioni dal menù d'avvio o attendere l'avvio automatico del sistema live da cui andremo a preparare il sistema all'installazione di Security Onion. Purtroppo è una fase che richiede diversi passaggi e riavvii di sistema. Partita la procedura d'installazione tramite l'icona sul Desktop, bisognerà seguire diversi step, tra cui scegliere la lingua, username, password e altre informazioni basilari prima di completare riavviando la macchina. Tornati sulla home, sempre attraverso l'icona principale, si andrà ad eseguire una seconda procedura che permetterà di configurare i diversi tool, scegliere l'interfaccia di rete del sistema ospitante (o virtualizzato), scegliere l'interfaccia dell'host per lo sniffing, ed effettuare un secondo riavvio. Una volta sulla home, andare di nuovo ad usare l'icona principale per completare l'installazione: è possibile scegliere una configurazione locale e adatta per il testing con la scelta della modalità Evaluation, e distribuita con la scelta di Production Mode; indicare dell'interfaccia di rete da valutare (quindi l'host), e di credenziali per l'accesso a specifici tool come Kibana, per arrivare al completamento. Terminata la procedura sarà necessario aggiornare le componenti essenziali con il comando `$ sudo soup` .

[79] https://github.com/Security-Onion-Solutions/security-onion/blob/master/Verify_ISO.md

Per controllare lo stato dei tool e monitorare che non ci siano problemi con il sistema è possibile utilizzare il comando `$ sudo sostat` . In caso di errori o fail da parte di diverse componenti si può far ricorso al comando `$ sudo nsm_sensor_ps-restart` per andare a riavviare tutti i servizi di Security Onion.

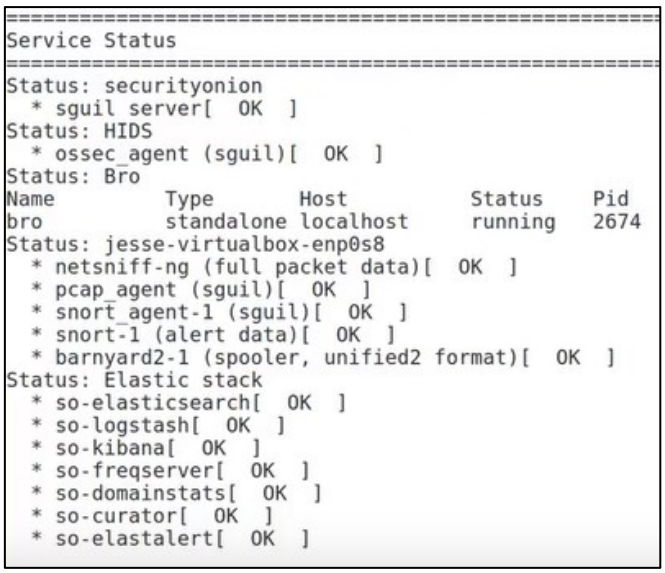


Figura 3.41 – Comando sudo sostat

Ecco un elenco dei principali tool[80] presenti nell'ecosistema:
Cattura: wireshark, netsniffing
NIDS: Bro, Snort, Suricata
HIDS: Wazuh
Analisi e gestione degli allarmi: Sguil, Squert, CapMe, Kibana

Nelle versioni pre-2018, durante l'installazione era possibile scegliere se procedere all'installazione e configurazione in modalità veloce o andando ad indicare diversi parametri con le scelte avanzate. Con le ultime versioni invece, completata l'installazione, si ha subito disponibile la piattaforma e i tool che la compongono per andare ad arginare le minacce. Migliori adattamenti possono essere attuati andando a configurare ulteriormente l'ecosistema, es [33]:

- Al percorso `/etc/nsm/name_of_sensor/Snort.conf` è disponibile il file per la configurazione di Snort, il quale attraverso ritocchi sulle variabili di rete, preprocessori, detection engine, ecc. permette di personalizzarne il funzionamento;

[80] https://securityonion.readthedocs.io/en/latest/Tools.html

- Gestire le regole per individuare minacce nella directory `/etc/nsm/rules/`
- Consigliabile impostare, per la configurazione di Suricata tramite il file `/etc/nsm/name_of_sensor/Suricata.yaml` , la topologia e le variabili di rete attraverso HOME_NET per la rete interna e !HOME_NET per tutto il resto.

Plugins

E' bene concordare come il supporto venga provvisto solo per la versione ufficiale di Security Onion e i suoi componenti, quindi implementazioni esterne come plugin non sono garantite. Ad ogni modo non mancano le situazioni di aziende che preferiscono servirsi di Security Onion solo come organismo di raccolta dati, per poi lasciare gli altri compiti ad altri strumenti come FIR, GRR, NtopNG, RITA, Etherpad.

3.8.2 Analisi per il threat model process

Appunto che l'elaborato tratta il threat model process, con l'utilizzo di Security Onion è possibile andare a sviluppare diverse soluzioni di sicurezza. Senza dubbio la necessità di testing e pratica del tool sono la caratteristica fondamentale, per tanto è interessante apprendere come si possa, attraverso la virtualizzazione, andare ad effettuare veri è propri training. Gran parte degli strumenti sono costosi, ed inoltre andare ad utilizzare veri malware per constatare l'efficienza, può comunque mettere a rischio i sistemi. Seguire, in sequenza, analisi di log, monitoraggio, correlamento e alerting, passi della Kill Chain[34], permette di individuare Security Onion come strumento adatto a garantire sicurezza sui servizi e protezione verso attacchi sconosciuti. E' possibile incrementare ulteriormente le prestazioni con l'integrazione di ELK Stack, ma richiede una macchina con buone prestazioni. Questo perché, rispetto alla normale struttura client(sensori)-server(Security Onion), viene aggiunto un livello(ELK) per il salvataggio dei dati.

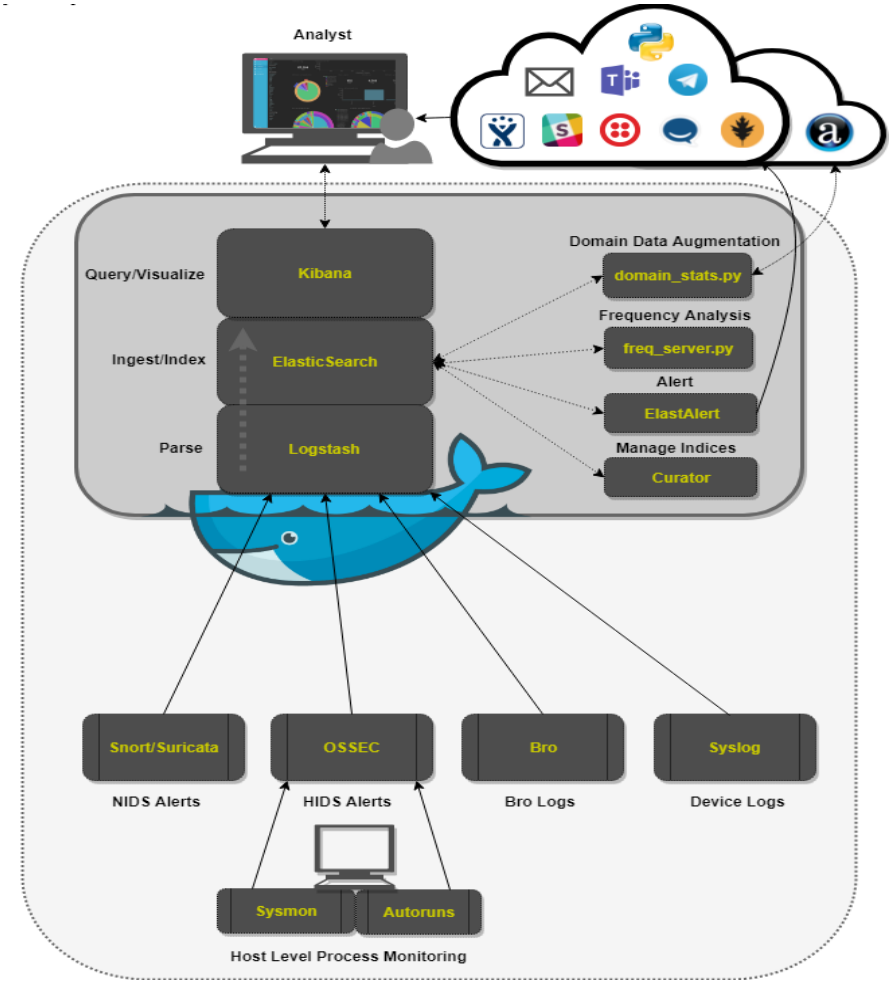


Figura 3.42 – Architettura Security Onion con ELK Stack[81]

[81] https://raw.githubusercontent.com/Security-Onion-Solutions/securityonion-docs/master/images/elastic-architecture/elastic-architecture.png

Capitolo IV

4. SVILUPPI FUTURI E CONCLUSIONI

4.1 Considerazioni

Ai giorni d'oggi credere che la sicurezza sia un'opzione, una facoltà, è come credere di vivere nella scatola di Schrodinger: non pensiamo a se la condizione di protezione è soddisfatta oppure no. Questo non ci preoccupa, eppure si continua ad essere vulnerabili verso ogni tipo di minaccia. Pensare poi di adottare una contromisura adeguata al livello di sicurezza che vorremmo raggiungere, spesso si trasforma nell'installazione di un antivirus, di un anti-malware o piattaforme di questa categoria, con l'illusione di aver eretto un muro invalicabile da chiunque: Sbagliatissimo! E' pur sempre un passo verso la giusta direzione, ma non basta. La considerazione di implementare nel nostro sistema una caratteristica costante, robusta, definitiva, ampliabile come il threat modeling, permette di adeguare il tutto al concetto di protezione e cambiare radicalmente il nostro ecosistema informatico permettendo di aggiungere qualità determinanti e obbligatorie per far fronte al dovere di proteggerci. Il bottino dell'attaccante, quindi dati, password o interruzioni di sistema, assumono un rilievo sempre maggiore sia nella nostra percezione della tecnologia, sia nel più ampio quadro di disponibilità indefinite di informazioni personali nella rete, o meglio nell'oblio.

La presentazione del lavoro di tesi ha cercato di chiarire il concetto di una forte impregnatura di sicurezza a tal punto da non adottare diverse soluzioni tra quelle disponibili in commercio, e quindi andando ad installare migliaia di tool per la protezione, ma andando ad assumere un atteggiamento positivo e più ampio già in fase di studio del sistema. Il concetto di threat modeling affrontato mira a chiarire le fasi e le qualità di un processo che viene inglobato nel sistema e che si rigenera, itera, andando ad apprendere sia dal sistema stesso, sia dal lavoro che ha compiuto fino all'ultima iterazione. Nello specifico l'obiettivo della tesi è quello di rappresentare in maniera completa le diverse fasi che colpiscono il threat modeling,

generalmente ed individualmente secondo le analisi effettuate, e come queste vengano gestite opportunamente andando ad ampliare e rinnovare l'ecosistema informatico vigente nell'organizzazione o sulla tipologia e topologia di cluster che stiamo considerando.

4.2 Prospettive

Lo spazio per ampliamenti e lavori futuri parte dall'apprendimento del fatto che stiamo considerando non un singolo tool o area di protezione o argomento di sicurezza, ma l'acquisizione di un intero comparto di regole, meccanismi, funzioni, ecc. . Partendo dalla considerazione appunto di "acquisizione" o "implementazione", la direzione più consona sarà sicuramente quella di sviluppare e migliorare piattaforme complete di tutto ciò che deve essere gestito per il processo di threat modeling, e quindi l'individuazione dei punti focali del sistema, l'individuamento delle minacce, la raccolta d'informazioni, la condivisione di tali, l'alerting, ecc. . A sua volta ognuna di queste fasi può essere migliorata andando a confrontare le diverse soluzioni software disponibili e lo sviluppo, anche parziale, di approcci personali, passando ad una prototipazione verso l'obiettivo di implementare definitivamente una miglioria. Nel paragrafo 3.9 poi, si accenna all'idea di poter sfruttare i vantaggi della containerizzazione per raggiungere risultati migliori con il processo di modellamento delle minacce. E' stata mostrata l'idea che ha portato al ragionamento condotto su diversi spunti da valutare per riuscire ad implementare e tastare direttamente. Tale valutazione, condotta in condivisione con i professori è stata vista di buon occhio anche per le emergenti realtà costituenti già un approccio ai container.

Riferimenti Bibliografici

[1] T. W. Federation, A. Kamal, and W. Federation, “Challenges to the Quantification of the Risks of Terrorism Three kinds of risk,” in *Risk Management*, 2004, no. May, pp. 1–11.

[2] Y. H. Tung, S. C. Lo, J. F. Shih, and H. F. Lin, “An integrated security testing framework for Secure Software Development Life Cycle,” in *18th Asia-Pacific Network Operations and Management Symposium, APNOMS 2016: Management of Softwarized Infrastructure - Proceedings*, 2016.

[3] I. No and S. S. Priya, “Threat Modeling for a Secured Software Development,” vol. 7, no. 1, pp. 40–48, 2016.

[4] C. Salter, O. S. Saydjari, B. Schneier, and J. Wallner, “Toward a secure system engineering methodolgy,” in *Proceedings of the 1998 workshop on New security paradigms - NSPW ’98*, 1998, pp. 2–10.

[5] U. B and K. Kandasamy, “Profiling Threat Modeling Approaches and Methodologies for It and Cloud Computing,” *Int. J. Pure Appl. Math.*, vol. 115, no. 8, pp. 121–126, 2017.

[6] C. Alberts and A. Dorofee, “OCTAVE SM* Threat Profiles,” *Network*, pp. 1–14, 2001.

[7] S. McElwee, J. Heaton, J. Fraley, and J. Cannady, “Deep learning for prioritizing and responding to intrusion detection alerts,” in *Proceedings - IEEE Military Communications Conference MILCOM*, 2017.

[8] J. Mtsweni, N. A. Shozi, K. Matenche, and M. Mutemwa, “Development of a Semantic-Enabled Cybersecurity Threat Intelligence Sharing Model,” in *Proceeedings of the International Conference on Cyber Warfare and Security*, 2016, pp. 244–252.

[9] S. Krishnan, “A Hybrid Approach to Threat Modelling,” 2017.

[10] A. Pardo, J. A. Fisteus, and C. D. Kloos, “A distributed collaborative system for flexible learning content production and management,” *J. Res. Pract. Inf. Technol.*, vol. 44, no. 2, pp. 203–221, 2012.

[11] D. Anstee, “The great threat intelligence debate,” *Comput. Fraud Secur.*, vol. 2017, no. 9, pp. 14–16, 2017.

[12] W. Tounsi and H. Rais, “A survey on technical threat intelligence in the age of sophisticated cyber attacks,” *Comput. Secur.*, vol. 72, pp. 212–233, 2018.

[13] Z. Wu, D. Xiao, H. Xu, X. Peng, and X. Zhuang, “Virtual inline: A technique of

combining IDS and IPS together in response intrusion," in *Proceedings of the 1st International Workshop on Education Technology and Computer Science, ETCS 2009*, 2009, vol. 1, pp. 1118–1121.

[14] G. Katsaros, R. Kübert, and G. Gallizo, "Building a service-oriented monitoring framework with REST and nagios," *Proc. - 2011 IEEE Int. Conf. Serv. Comput. SCC 2011*, pp. 426–431, 2011.

[15] J. Hernantes, G. Gallardo, and N. Serrano, "IT Infrastructure-Monitoring Tools," *IEEE Softw.*, vol. 32, no. 4, pp. 88–93, 2015.

[16] S. Taherizadeh, A. C. Jones, I. Taylor, Z. Zhao, and V. Stankovski, "Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review," *J. Syst. Softw.*, vol. 136, pp. 19–38, 2018.

[17] C. Toland, C. Meenan, M. Warnock, and P. Nagy, "Proactively monitoring departmental clinical IT systems with an open source availability system," *J. Digit. Imaging*, vol. 20, no. SUPPL. 1, pp. 119–124, 2007.

[18] A. Bardi, C. Atzori, M. Artini, M. Mikulicic, P. Manghi, and S. La Bruzzo, "High-Performance Annotation Tagging over Solr Full-text Indexes," *Inf. Technol. Libr.*, vol. 33, no. 3, p. 22, 2014.

[19] D. Sonntag and H. J. Profitlich, "An architecture of open-source tools to combine textual information extraction, faceted search and information visualisation," *Artif. Intell. Med.*, vol. 93, pp. 13–28, 2019.

[20] M. Wagner, A. Rind, N. Thür, and W. Aigner, "A knowledge-assisted visual malware analysis system: Design, validation, and reflection of KAMAS," *Comput. Secur.*, vol. 67, pp. 1–15, 2017.

[21] D. Barberis *et al.*, "ATLAS Eventlndex monitoring system using the Kibana analytics and visualization platform," *J. Phys. Conf. Ser.*, vol. 762, no. 1, 2016.

[22] T. Prakash, M. Kakkar, and K. Patel, "Geo-identification of web users through logs using ELK stack," *Proc. 2016 6th Int. Conf. - Cloud Syst. Big Data Eng. Conflu. 2016*, pp. 606–610, 2016.

[23] Y. Lee, W. Kang, and Y. Lee, "A Hadoop-Based Packet Trace Processing Tool," in *Lecture Notes in Computer Science*, 2011, vol. 9, no. 3, pp. 51–63.

[24] S. Tripathi, A. Mishra, A. Almomani, B. Gupta, and S. Veluru, "Hadoop Based Defense Solution to Handle Distributed Denial of Service (DDoS) Attacks," *J. Inf. Secur.*, vol. 04, no. 03, pp. 150–164, 2013.

[25] A. Hameed, S. Hameed, and U. Ali, "EURASIP Journal on Information Security HADEC: Hadoop-based live DDoS detection framework," *EURASIP J. Inf. Secur.*, vol.

2018, pp. 1–19, 2018.

[26] T. Hsu, *Hands-on security in DevOps: ensure continuous security, deployment, and delivery with DevSecOps*. 2018.

[27] S. K. Garg, J. Lakshmi, and J. Johny, “Migrating VM Workloads to Containers: Issues and Challenges,” in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, 2018, pp. 778–785.

[28] M. F. Thompson and C. E. Irvine, “Individualizing Cybersecurity Lab Exercises with Labtainers,” *IEEE Secur. Priv.*, vol. 16, no. 2, pp. 91–95, 2018.

[29] I. No, V. Ojha, and K. Parihar, “Available Online at www.ijarcs.info International Journal of Advanced Research in Computer Science COMPARISON OF THE TWO GIANTS IN CONTAINER PLATFORMS : KUBERNETES WITH OPENSTACK VS . DOCKER,” vol. 9, no. 3, pp. 229–232, 2018.

[30] X. Geng, X. Zeng, L. Hu, and Z. Guo, “An Novel Architecture and Inter-process Communication Scheme to Adapt Chromium Based on Docker Container,” *Procedia Comput. Sci.*, vol. 107, no. Icict, pp. 691–696, 2017.

[31] F. Sano, T. Okamoto, I. Winarno, Y. Hata, and Y. Ishida, “A Cyber Attack-Resilient Server Using Hybrid Virtualization,” *Procedia Comput. Sci.*, vol. 96, pp. 1627–1636, 2016.

[32] R. Brislin, “Introduction to Security,” in *The Effective Security Officer’s Training Manual*, 2014, no. May, pp. 1–7.

[33] A. Deuble, “Information Security Reading Room Using and Configuring Security Onion to detect and prevent Web Application In s t i t u t e u t h o r r e t a i n s f u l l r i g h t s,” 2019.

[34] B. S. J and S. Prabhakaran, “Intrusion Detection using Security Onion Based on Kill Chain Approach,” vol. 4, no. 3, pp. 2013–2016, 2015.