ANALISI DEL MALWARE

&

SPLUNK

CSPT0324 MODULO 5

YILEI WU

20 DICEMBRE 2024

INDICE

TRACCIA ESERCIZIO 3

REQUISTI E CONFIGURAZIONE LABORATORIO VIRTUALE 4

Installazione 4

Rete 4

Estensione file su Windows 4

SVOLGIMENTO ESERCIZIO 5

tutorialdata.zip - Download & Importazione dati 5

Query tentativi di accesso falliti "Failed password" 8
- Procedimento e spiegazione passo passo 8
- Query: 9
- Output: 9

Query sessioni SSH aperte con successo – djohnson 10
- Procedimento e spiegazione passo passo 10
- Query: 10
- Output: 11

Query tentativi di accesso falliti da "86.212.199.60" 12
- Procedimento e spiegazione passo passo 12
- Query: 12
- Output: 12

Query tentativi di accesso falliti "Failed password" superiore a 5 volte 13
- Procedimento e spiegazione passo passo 13
- Query: 13
- Output: 13

Query Internal Server Error 14
- Procedimento e spiegazione passo passo 14
- Query: 15
- Output: 15

Conclusioni tramite A.I. 16
- Prompt 16
- Premessa 16
- Output A.I. 16
- Conclusioni finali sull'analisi tramite A.I. 18

TRACCIA ESERCIZIO

Progetto

Importate su Splunk i dati di esempio “tutorialdata.zip”:

1. Crea una query Splunk per identificare tutti i tentativi di accesso falliti "Failed password". La query dovrebbe mostrare il timestamp, l'indirizzo IP di origine, il nome utente e il motivo del fallimento.
2. Scrivi una query Splunk per trovare tutte le sessioni SSH aperte con successo. La query dovrebbe filtrare per l'utente “djohnson” e mostrare il timestamp e l'ID utente.
3. Scrivi una query Splunk per trovare tutti i tentativi di accesso falliti provenienti dall'indirizzo IP “86.212.199.60”. La query dovrebbe mostrare il timestamp, il nome utente e il numero di porta.
4. Crea una query Splunk per identificare gli indirizzi IP che hanno tentato di accedere ("Failed password") al sistema più di 5 volte. La query dovrebbe mostrare l'indirizzo IP e il numero di tentativi.
5. Crea una query Splunk per trovare tutti gli Internal Server Error.

Trarre delle conclusioni sui log analizzati utilizzando AI.

REQUISTI E CONFIGURAZIONE LABORATORIO VIRTUALE

INSTALLAZIONE

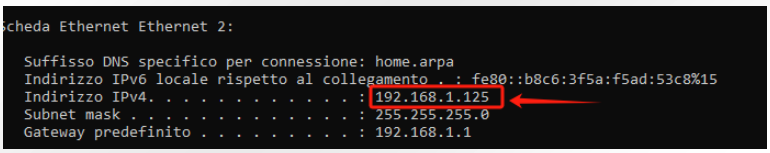
Per l'installazione di Splunk, si seguono le indicazioni fornite nei documenti M6W24D2 (report) e M6W24D1 (slide guida), considerando i seguenti requisiti:

- Hardware: dato che Splunk è un software ad alto consumo di risorse, in questo caso si è optato per una configurazione con 4 GB di RAM e 6 processori;
- sistema operativo: l'installazione è preferibile su Windows Server, ma, per il presente esercizio di laboratorio, è possibile anche l'utilizzo di Windows 10 Pro N.
- Configurare il prima possibile il fuso orario, avere l'orario preciso è essenziale.
- Splunk Enterprise in versione prova dura 60 giorni https://www.splunk.com/en_us/download/splunk-enterprise.html

RETE

Ai fini dell'esercizio, si è optato per una configurazione DHCP tramite pfSense, soprattutto per avere una connessione alla rete internet, evitando il passaggio di modifica della scheda di rete di Windows 10 per eventuali collegamenti con altre VM.

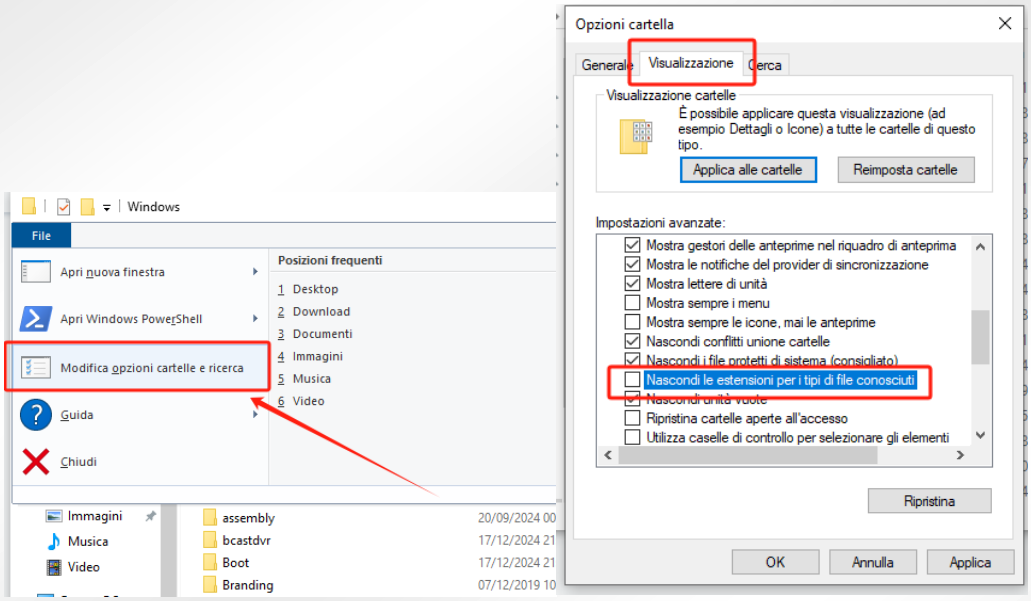
Alla macchina virtuale con Windows 10 Pro N, su cui è installato Splunk, è stato assegnato l'indirizzo IP 192.168.1.125.



ESTENSIONE FILE SU WINDOWS

È sempre consigliabile visualizzare i nomi dei file includendo anche la loro estensione:

1. Aprire una finestra di File Explorer (Windows + E);
2. Modifica opzioni cartelle e ricerca;
3. Visualizzazione;
4. Deselezionare "Nascondi le estensioni per i tipi di file conosciuti";
5. "Ok" o "Applica"

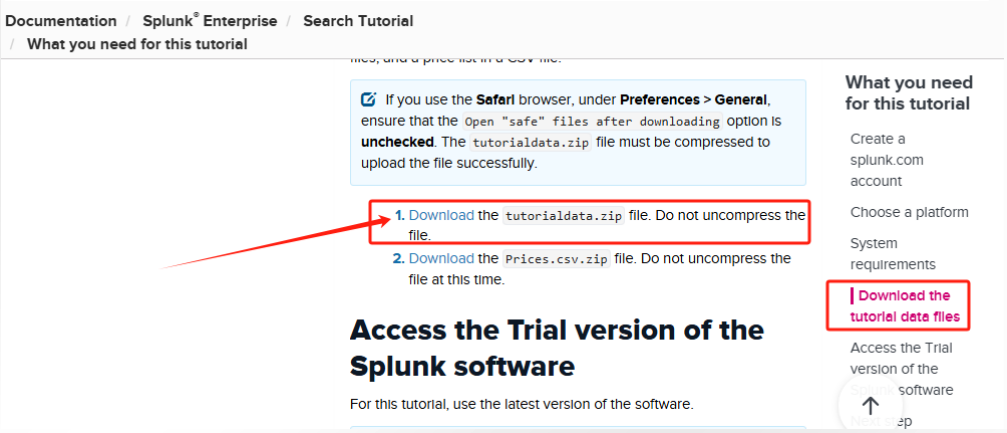


SVOLGIMENTO ESERCIZIO

TUTORIALDATA.ZIP - DOWNLOAD & IMPORTAZIONE DATI

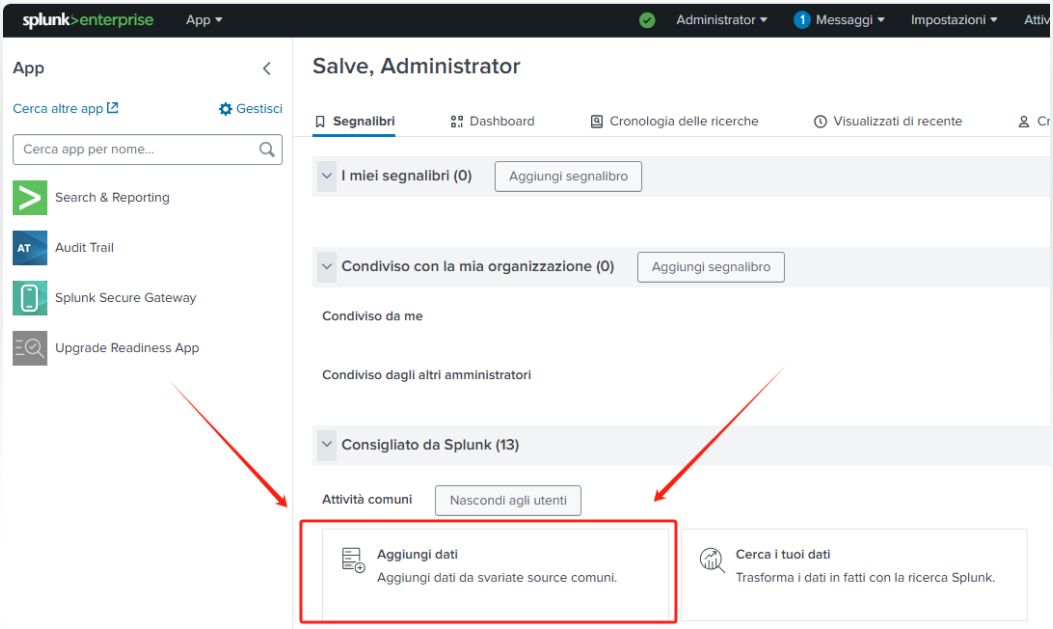
1. Aprire il link tramite il browser di della VM su cui è installata Splunk, Windows 10 Pro N, il seguente link: https://docs.splunk.com/Documentation/Splunk/9.2.1/SearchTutorial/Systemrequirements#Choose_a_platform

2. Scaricare il file "tutorialdata.zip", all'incirca a metà pagina.



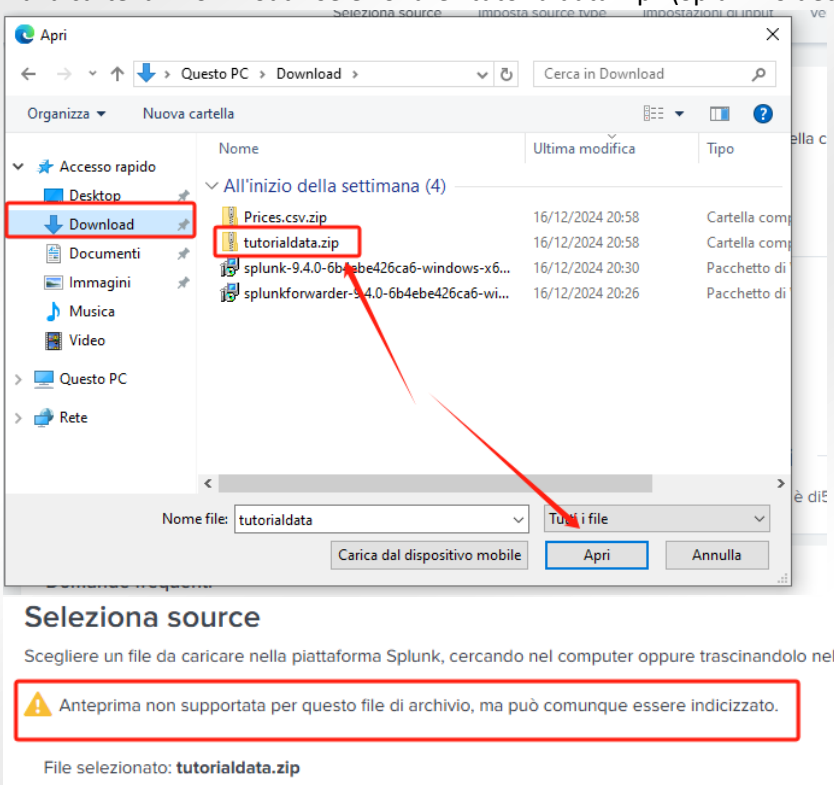
3. Aprire Splunk Enterprise tramite link http://127.0.0.1:8000/ (localhost) o tramite icona sul desktop.

4. Dalla home page recarsi su "Aggiungi dati"

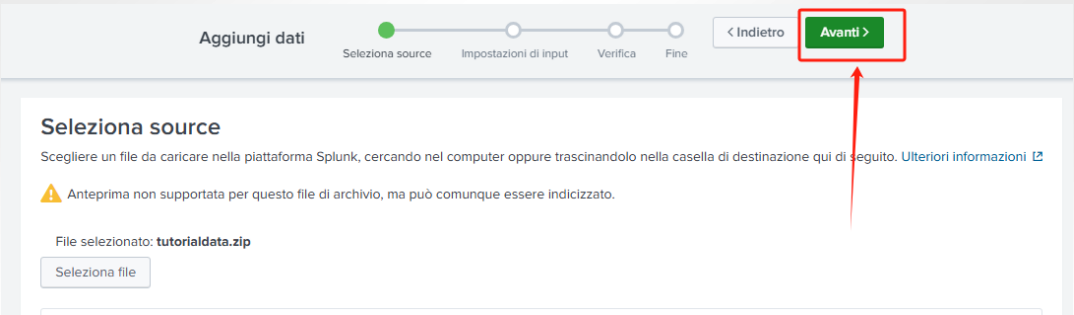


5. Selezionare “Carica” e caricare il file scaricato.

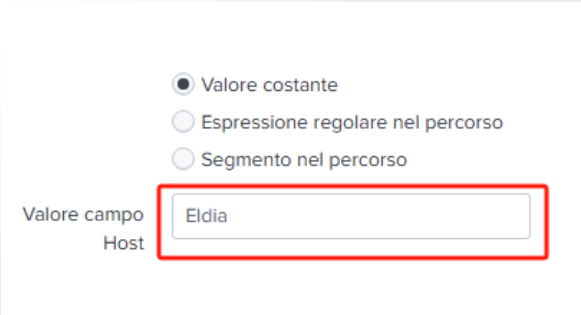
6. Dalla cartella “Download” selezionare “tutorialdata.zip” (Splunk lo decomprimerà in autonomia)



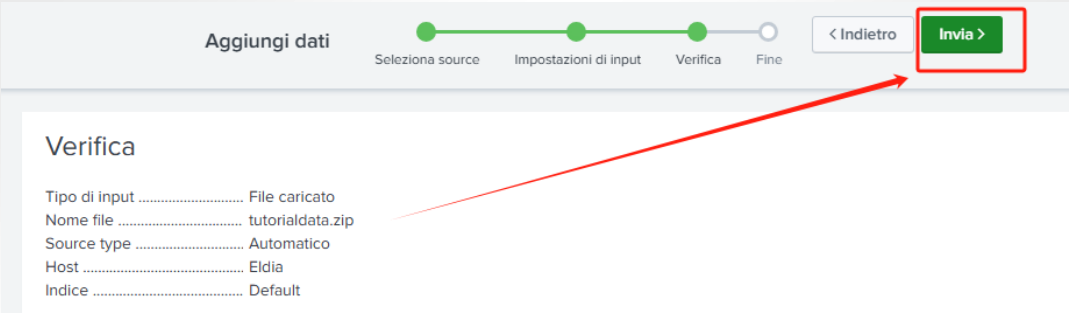
7. “Avanti”



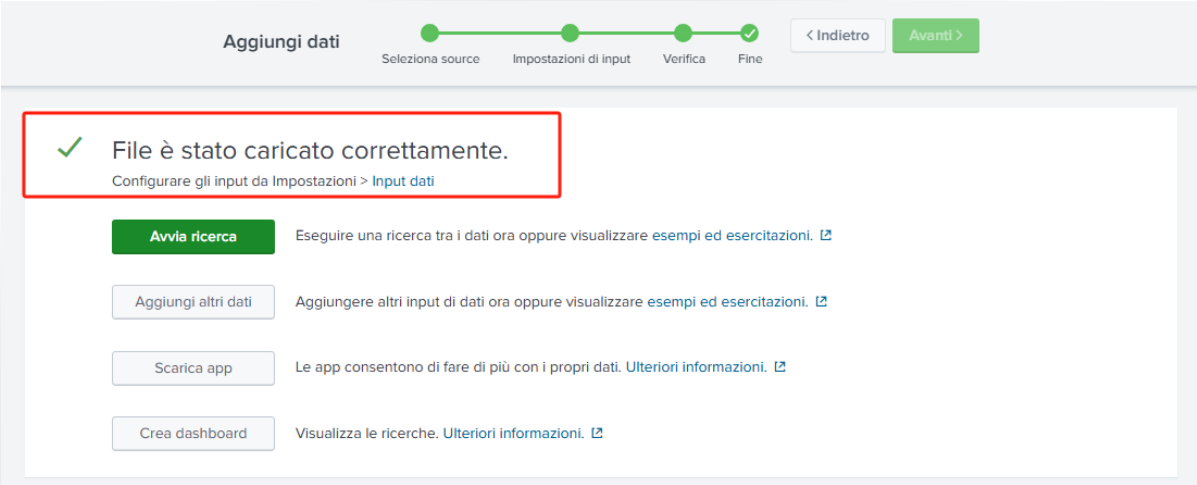
8. Personalizzare il “Valore campo Host”, in questo caso, si chiamerà “Eldia” e tenere invariato gli altri campi.



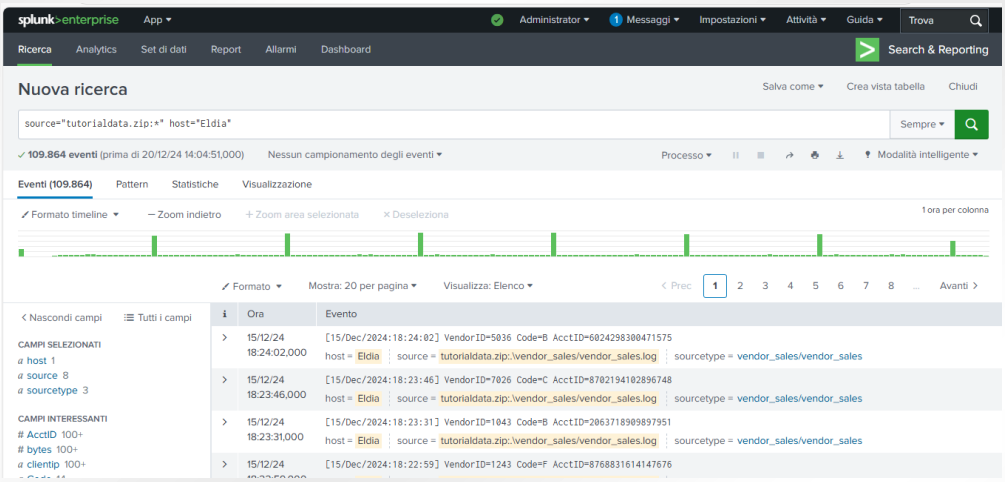
9. Controllo e “Invia”



10. Caricato correttamente



11. Avviare la ricerca con “Avvia ricerca”

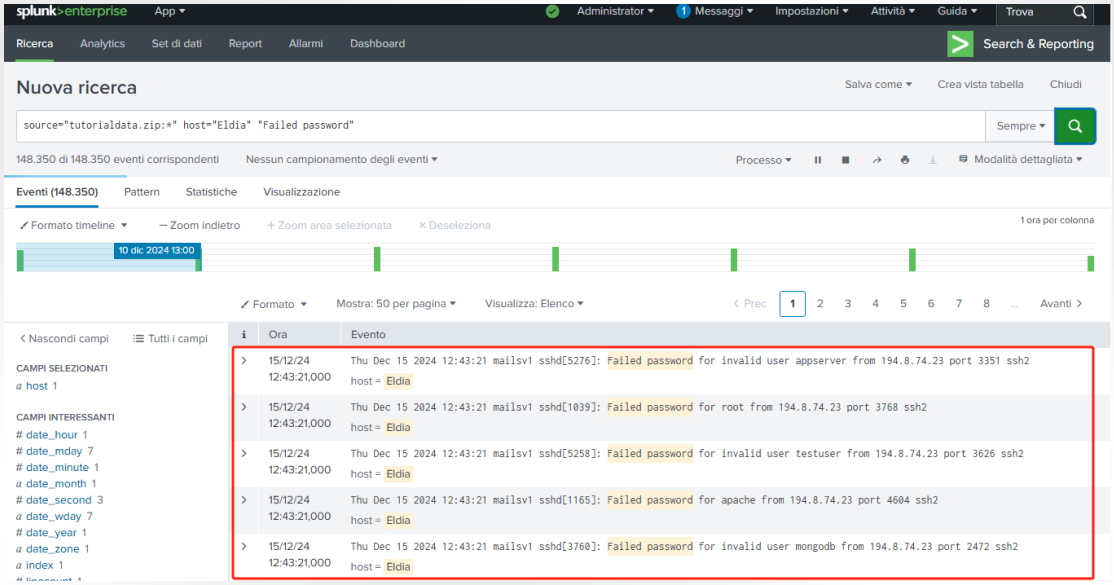


QUERY TENTATIVI DI ACCESSO FALLITI “FAILED PASSWORD”

PROCEDIMENTO E SPIEGAZIONE PASSO PASSO

La traccia fornisce un primo suggerimento per costruire la query utilizzando la parola chiave "Failed password". L'analisi inizia con una ricerca di base: **source="tutorialdata.zip:*" host="Eldia" "Failed password"**

- **source="tutorialdata.zip:*"**: specifica la fonte dei dati, che proviene dal pacchetto zip caricato precedentemente.
- **host="Eldia"**: limita la ricerca ai log provenienti dall'host rinominato precedentemente "Eldia".
- **"Failed password"**: filtra gli eventi che contengono questa parola chiave, indicativa dei tentativi di accesso non riusciti.

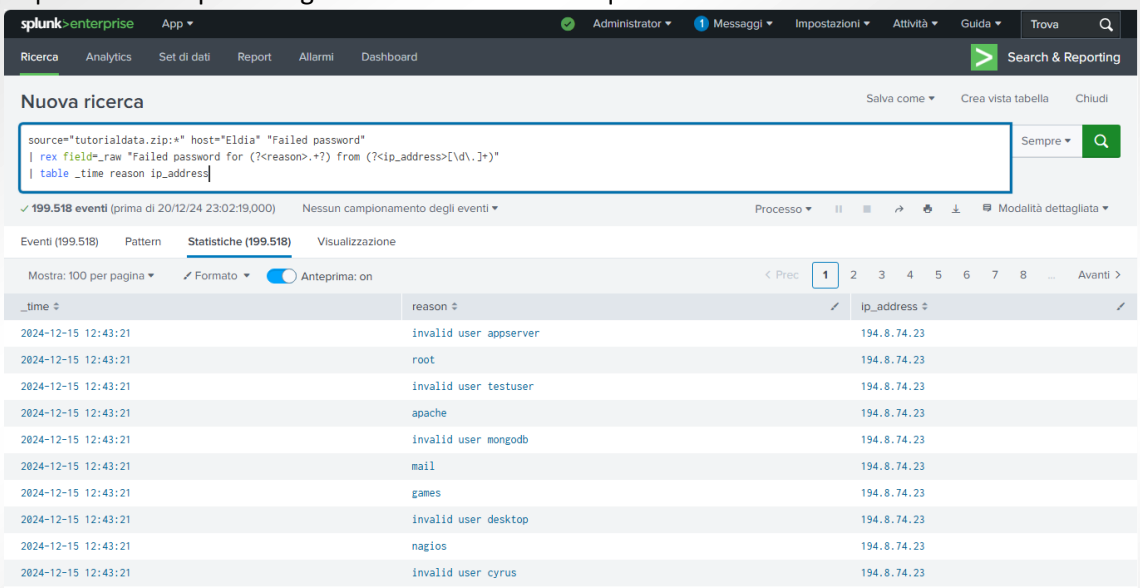


Ogni evento presenta una struttura ricorrente che include la data e l'ora, seguite dalla dicitura **"Failed password for"**, accompagnata da una variabile indicante la motivazione e dall'indirizzo IP di origine. Tale struttura può essere tradotta nel seguente comando, utilizzando il comando rex per estrarre i campi desiderati:

```
source="tutorialdata.zip:*" host="Eldia" "Failed password"
| rex field=_raw "Failed password for (?<reason>.+?) from (?<ip_address>[\d\.]+)"
| table _time reason ip_address
```

Il comando “table” è stato aggiunto per formattare l'output in una tabella che mostra le colonne relative alla data e ora (_time), alla motivazione (reason) e all'indirizzo IP (ip_address).

La “pipe |” è utilizzata per collegare comandi in modo sequenziale.

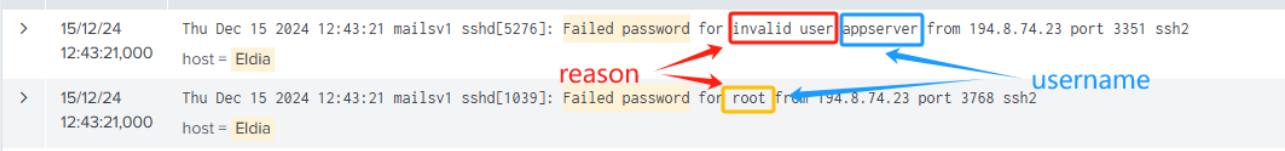


Tuttavia manca il nome utente richiesto dalla traccia.

```
| rex field=_raw "Failed password for (?<reason>(invalid user )?(?<username>[^\s]+)) from (?<ip_address>[\d\.]+)"
```

L'espressione aggiunta **(?<reason>(invalid user)?(?<username>[^\s]+))** serve a:

1. Catturare una parte del testo del log, dandole un nome:
   - reason: Cattura l'intera motivazione, come "invalid user appserver" o "root".
   - username: Cattura solo il nome utente, come "appserver" o "root".



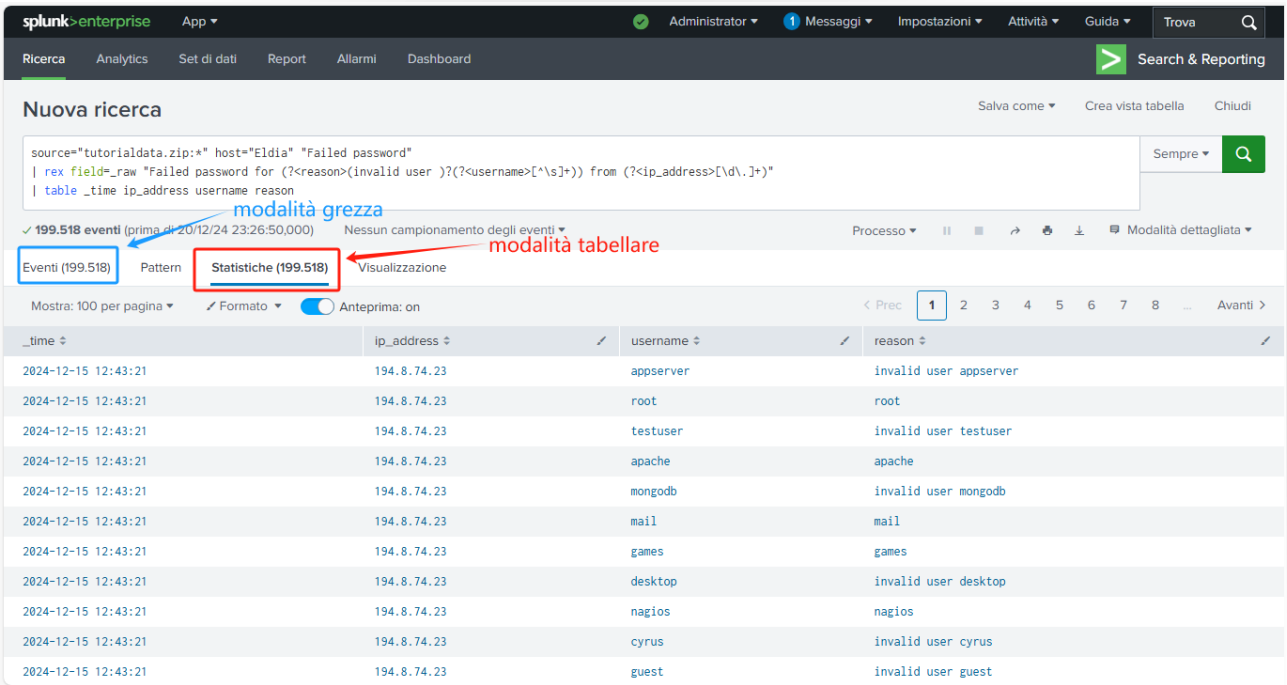
2. Gestire parti opzionali: **(invalid user)?** rende "invalid user" opzionale (può esserci o no).
3. Catturare il nome utente senza spazi: **[^\s]+** prende una sequenza di caratteri non separati da spazi, cioè il nome utente.

QUERY:

```
source="tutorialdata.zip:*" host="Eldia" "Failed password"
| rex field=_raw "Failed password for (?<reason>(invalid user )?(?<username>[^\s]+)) from (?<ip_address>[\d\.]+)"
| table _time ip_address username reason
```

OUTPUT:

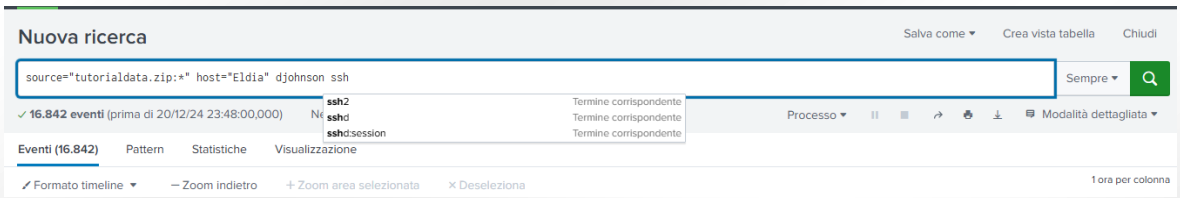
Modalità di visualizzazione **Statistiche**.



L'output segue l'ordine della traccia: timestamp, l' indirizzo IP di origine, il nome utente e il motivo del fallimento.

QUERY SESSIONI SSH APERTE CON SUCCESSO – DJOHNSON

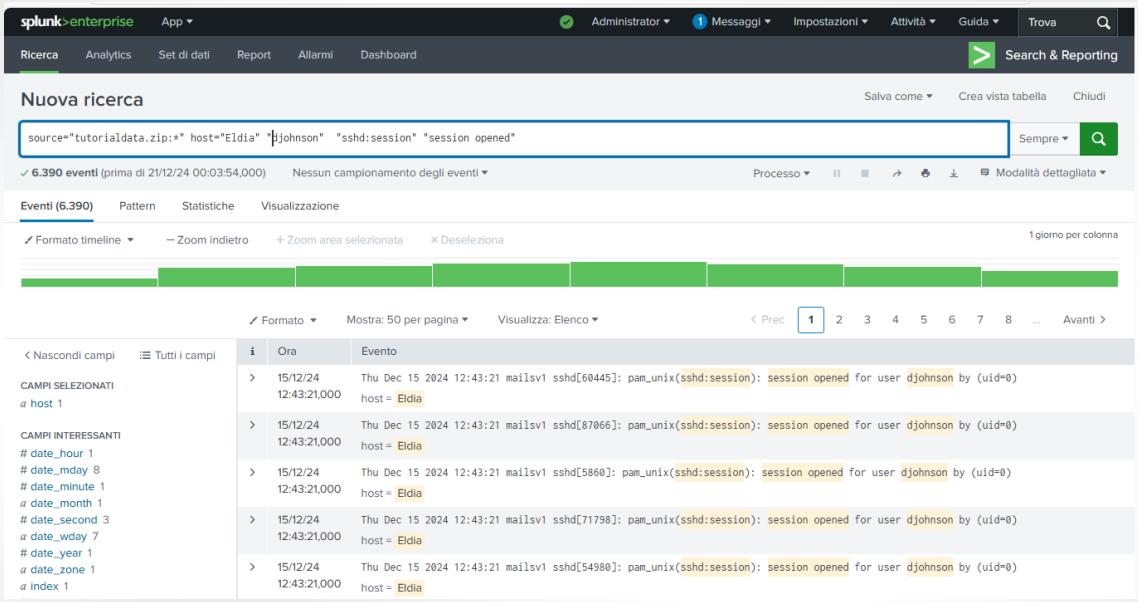
PROCEDIMENTO E SPIEGAZIONE PASSO PASSO



Analogamente alla precedente query, si è iniziato con un primo approccio utilizzando le parole chiave djohnson e ssh. Dopo aver verificato i risultati intermedi, si è giunti alla seguente query:

source="tutorialdata.zip:*" host="Eldia" "djohnson" "sshd:session" "session opened"

Questa query consente di filtrare gli eventi rilevanti relativi all'utente djohnson e alle sessioni SSH aperte.



Ogni evento presenta una struttura ricorrente che include la data e l'ora, pam_unix(sshd:session): session opened for user <nome utente>, che si può tradurre, come precedentemente visto, in un comando rex:

rex "session opened for user (?<user_id>\w+).*uid=(?<uid>\d+)"

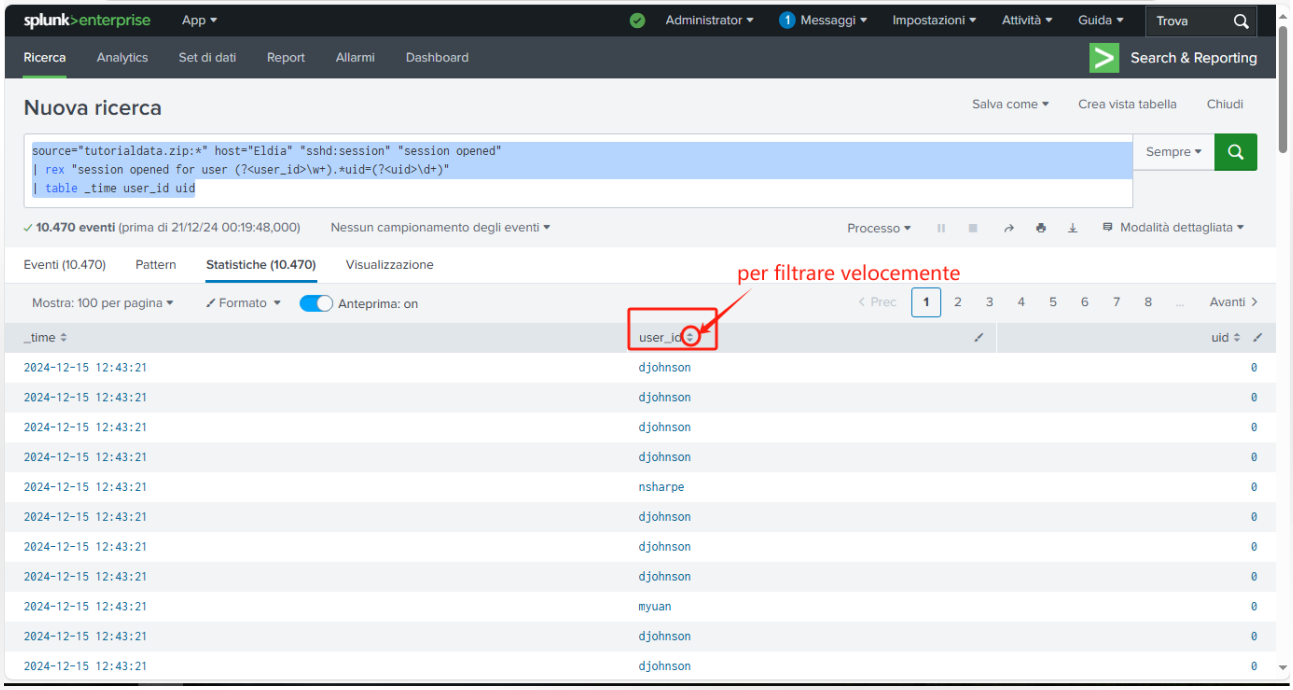
Si aggiunge il comando table per ottenere una tabella con tempo, nome utente e id numerico.

table _time user_id uid

QUERY:

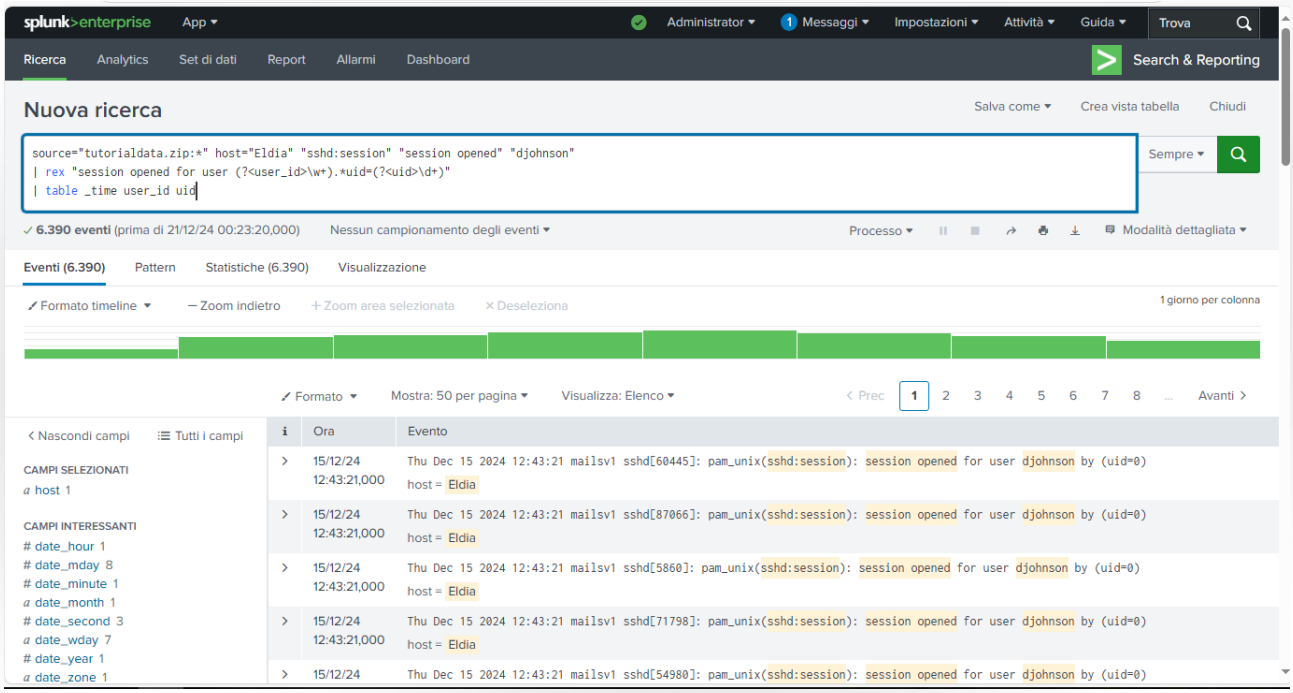
```
source="tutorialdata.zip:*" host="Eldia" "sshd:session" "session opened"
| rex "session opened for user (?<user_id>\w+).*uid=(?<uid>\d+)"
| table _time user_id uid
```

OUTPUT:



Dal risultato finale, è stata inclusa la funzione di poter filtrare, ordinando la lista in ordine alfabetico, per poter visualizzare solo gli eventi di “djohnson” oppure aggiungere alla query “djohnson” nella prima riga.

source="tutorialdata.zip:*" host="Eldia" "sshd:session" "session opened" "djohnson"
| rex "session opened for user (?<user_id>\w+).*uid=(?<uid>\d+)"
| table _time user_id uid

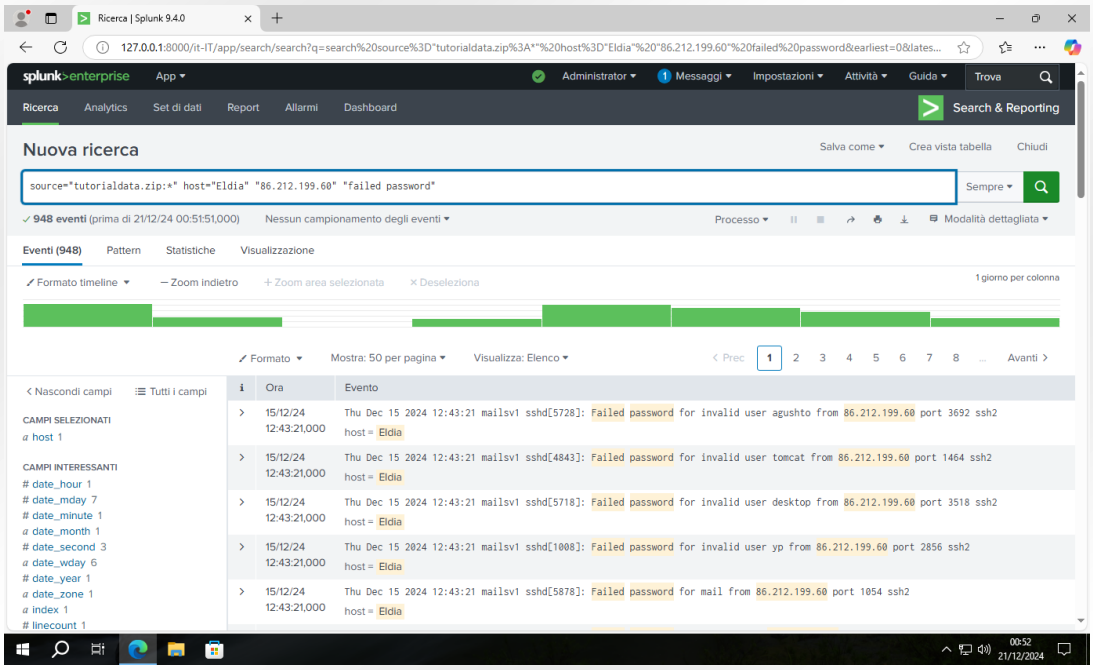


Infatti gli eventi sono ridotti a 6390 dopo il filtro.

QUERY TENTATIVI DI ACCESSO FALLITI DA “86.212.199.60”

PROCEDIMENTO E SPIEGAZIONE PASSO PASSO

Ricerca preliminare attraverso le parole chiavi della traccia: **source="tutorialdata.zip:*" host="Eldia" "86.212.199.60" "failed password"**

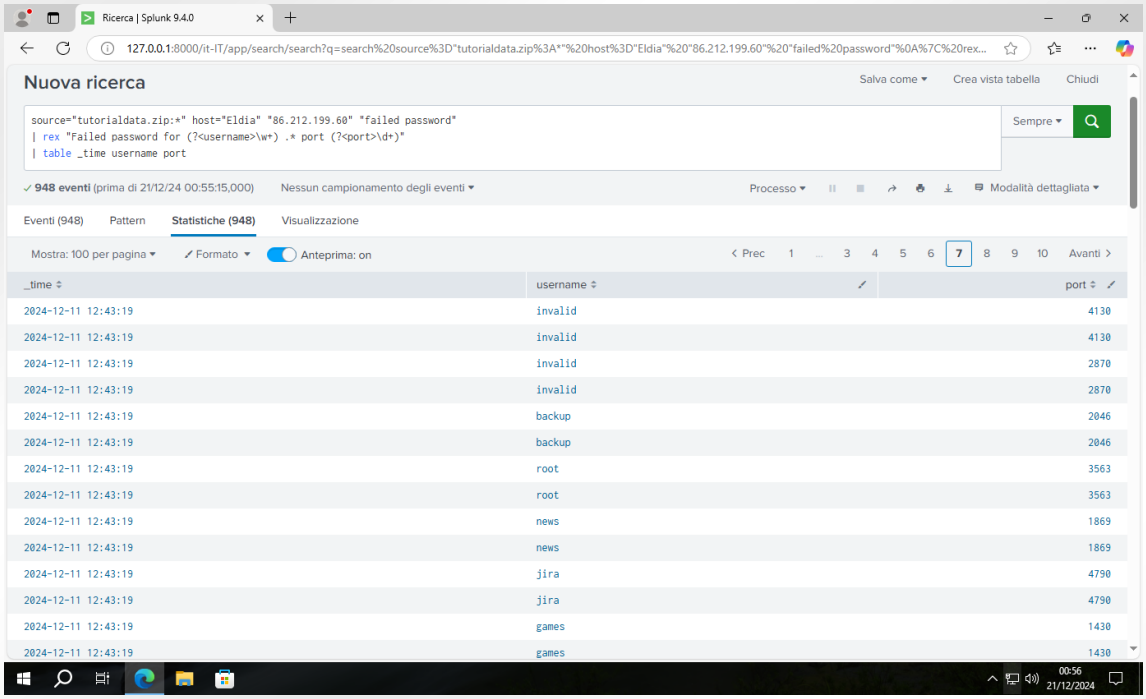


Data la struttura ripetitiva degli eventi si crea un rex per la variabile username e numero di porta, riportandolo in tabella.

QUERY:

```
source="tutorialdata.zip:*" host="Eldia" "86.212.199.60" "failed password"
| rex "Failed password for (?<username>\w+) .* port (?<port>\d+)"
| table _time username port
```

OUTPUT:



QUERY TENTATIVI DI ACCESSO FALLITI "FAILED PASSWORD" SUPERIORE A 5 VOLTE

PROCEDIMENTO E SPIEGAZIONE PASSO PASSO

Analogamente alla precedente, si parte togliendo l'indirizzo IP dalla ricerca: **source="tutorialdata.zip:*" host="Eldia" "failed password"**

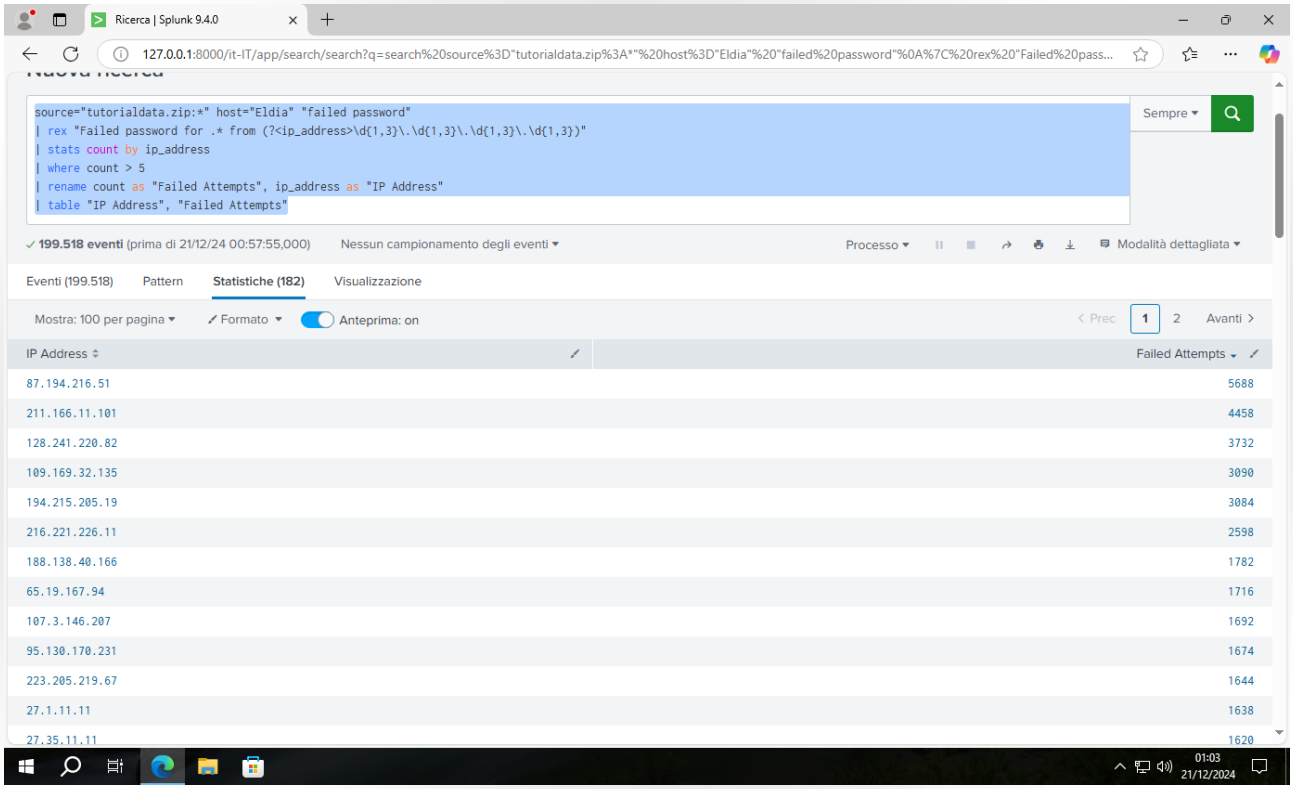
Si aggiungono i seguenti comandi e operazioni per elaborare e filtrare i dati desiderati:

- **rex**: estrae l'indirizzo IP dai messaggi di log che contengono "Failed password". Il campo estratto è ip_address. La regex **\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}** cattura un indirizzo IPv4.
- **stats count by ip_address**: conta il numero di eventi per ciascun indirizzo IP.
- **where count > 5**: Filtra i risultati per includere solo gli indirizzi IP con più di 5 tentativi falliti.
- **rename**: Rinomina i campi per una presentazione più chiara:
    a. count → "Failed Attempts"
    b. ip_address → "IP Address"
- **table**: Mostra solo i campi "IP Address" e "Failed Attempts" nella tabella finale.

QUERY:

```
source="tutorialdata.zip:*" host="Eldia" "failed password"
| rex "Failed password for .* from (?<ip_address>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
| stats count by ip_address
| where count > 5
| rename count as "Failed Attempts", ip_address as "IP Address"
| table "IP Address", "Failed Attempts"
```

OUTPUT:

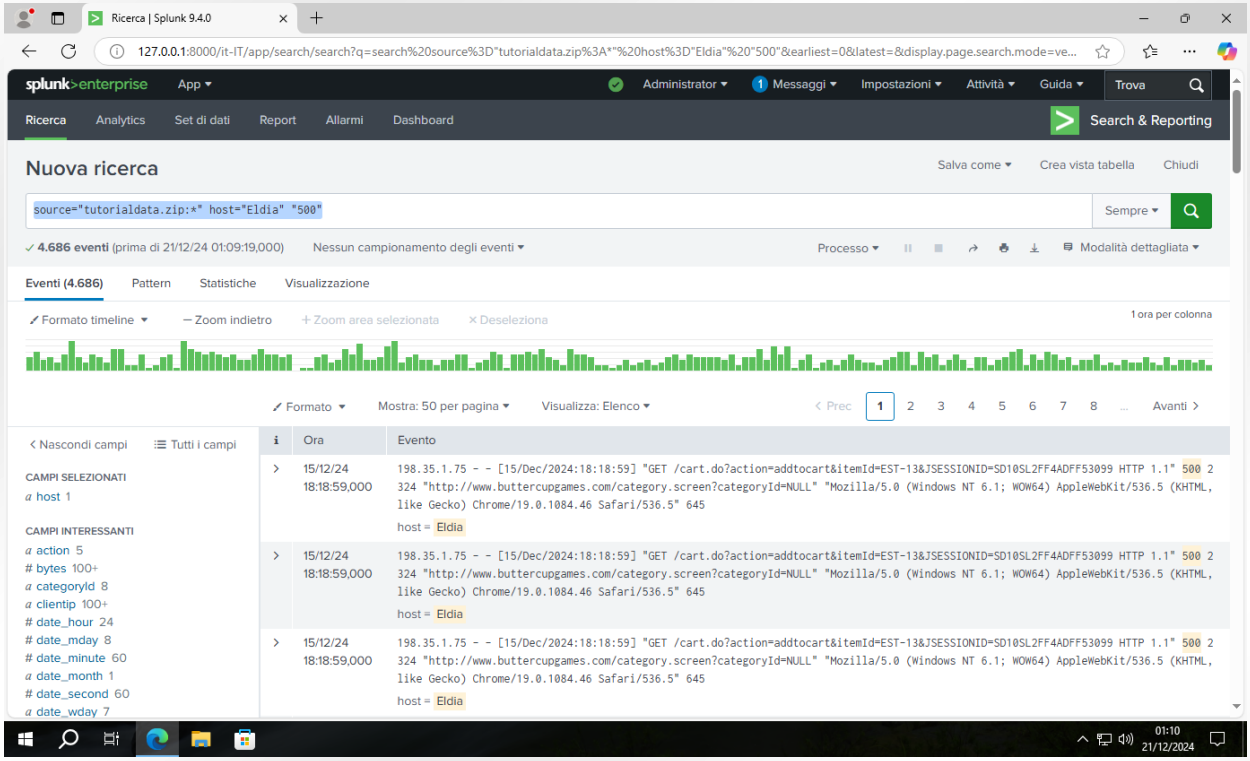


In questo modo è possibile identificare gli indirizzi IP che hanno effettuato più di 5 tentativi di accesso falliti al sistema. Come mostrato nell'immagine catturata, il risultato elenca un totale di 182 indirizzi IP, ordinati in ordine decrescente in base al numero di tentativi falliti.

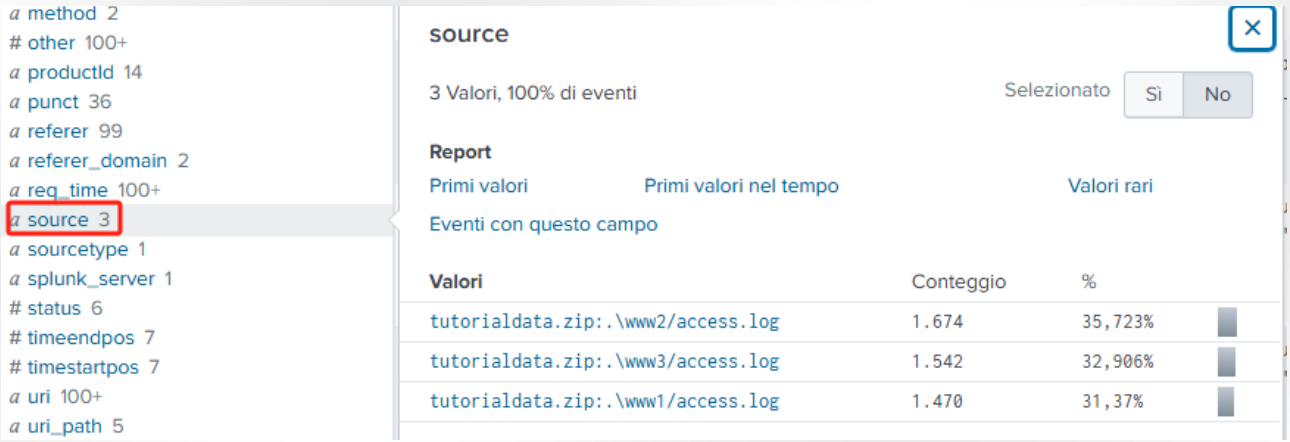
QUERY INTERNAL SERVER ERROR

PROCEDIMENTO E SPIEGAZIONE PASSO PASSO

Internal Server Error, è rappresentato con il codice di stato http “500” e indica che il server ha riscontrato una condizione imprevista che gli ha impedito di soddisfare la richiesta. Pertanto si aggiunge il codice “500” nella query: **source="tutorialdata.zip:*" host="Eldia" "500"**



Si può filtrare per sorgente, a sinistra su “source”



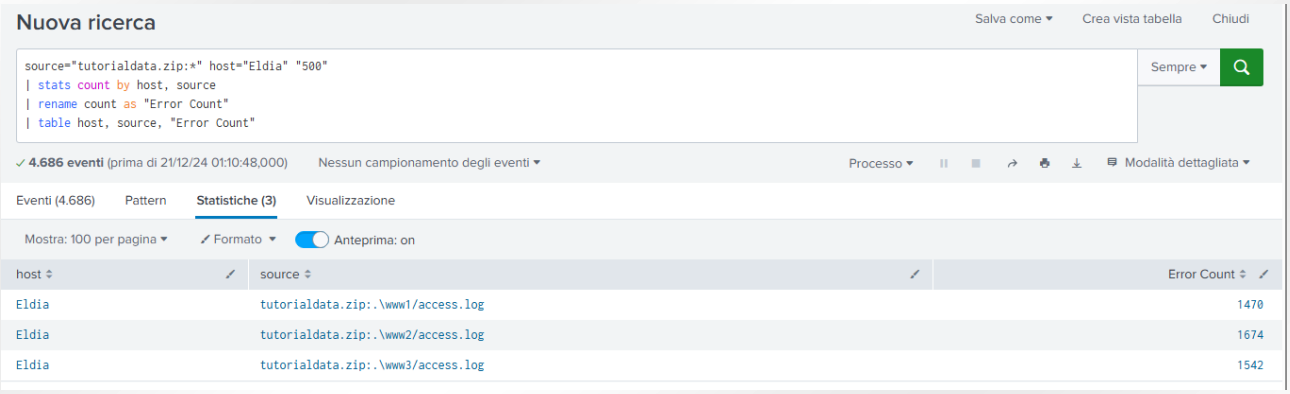
Si può inoltre rielaborarlo ai fini statistici, nella relativa sezione, con l’aggiunta di comandi e operazioni:

- **stats count by host, source**: conta il numero di errori HTTP 500 per ciascun host e sorgente.
- **rename count as "Error Count"**: rinomina il campo count in "Error Count" per maggiore chiarezza.
- **table host, source, "Error Count"**: mostra i risultati in formato tabellare con le colonne host, source e il numero di errori (Error Count).

QUERY:

```
source="tutorialdata.zip:*" host="Eldia" "500"
| stats count by host, source
| rename count as "Error Count"
| table host, source, "Error Count"
```

OUTPUT:



CONCLUSIONI TRAMITE A.I.

PROMPT

Sono un analista autorizzato e sto conducendo attività di **malware analysis** e **analisi dei log su Splunk** per individuare potenziali minacce alla sicurezza, seguendo le politiche aziendali. Chiedo una guida passo-passo per attività come identificazione di indicatori di compromissione (IOCs), analisi di comportamento sospetto e rilevamento di anomalie nei log, utilizzando strumenti e tecniche efficaci. Rispondi solo con soluzioni verificate e basate su conoscenze consolidate, senza ipotesi o deduzioni. Se una soluzione certa non esiste, dichiaralo chiaramente. Mantieni un tono professionale, chiaro e focalizzato, spiegando i metodi o le fonti utilizzate per ogni risposta. Analizza e dammi le tue conclusioni sui risultati dei log presenti in questo report.

PREMESSA

Attualmente, strumenti di intelligenza artificiale come ChatGPT e altre soluzioni **online gratuite** non sono in grado di elaborare documenti contenenti grandi quantità di dati, si ribadisce, in modo gratuito, come log esportati in formato PDF o CSV (vedasi allegati PDF), a causa dell'elevato volume di informazioni da processare. Di conseguenza, si è scelto di trasmettere gli screenshot estratti dal presente report per consentire un'analisi più mirata e gestibile.

OUTPUT A.I.

Analisi dei risultati presenti nel report del documento "Analisi del Malware e Splunk"
Basandomi sui risultati e sulle query proposte per l'analisi dei log, ecco una guida dettagliata, seguita da un'interpretazione per ogni query e le conclusioni generali:

---

1. Query: Tentativi di accesso falliti ("Failed password")
Descrizione:
L'analisi ha identificato tentativi di accesso falliti con una query che estrae il timestamp, l'indirizzo IP di origine, il nome utente e il motivo del fallimento.
Considerazioni:

- Tentativi di accesso falliti sono indicatori classici di **brute force attacks**. Questi eventi possono indicare che un attore malintenzionato sta cercando di ottenere accesso non autorizzato al sistema.
- Gli indirizzi IP coinvolti devono essere analizzati per verificare la provenienza geografica e le eventuali correlazioni con attività malevole conosciute (ad es. tramite strumenti come VirusTotal o AbuseIPDB).

---

2. Query: Sessioni SSH aperte con successo – Utente specifico ("djohnson")
Descrizione:
La query filtra le sessioni SSH aperte con successo per l'utente "djohnson", mostrando il timestamp e l'ID utente.
Considerazioni:

- L'accesso SSH con successo potrebbe essere legittimo, ma dato che l'utente specifico è stato evidenziato, bisogna verificare se **l'ora e il luogo dell'accesso** corrispondono alle normali attività dell'utente.
- Se l'accesso avviene da un indirizzo IP o un luogo insolito, potrebbe trattarsi di un **account compromesso**.
- È opportuno confrontare questo evento con altri log (ad esempio, tentativi falliti per lo stesso utente).

3. Query: Tentativi di accesso falliti da un IP specifico ("86.212.199.60")
Descrizione:
La query ha individuato i tentativi di accesso falliti provenienti dall'indirizzo IP "86.212.199.60", estraendo il nome utente e il numero di porta.
Considerazioni:

- La presenza di tentativi ripetuti da un indirizzo IP specifico indica un possibile **attacco mirato** da parte di quell'IP.
- È fondamentale eseguire un'**analisi di reputazione IP** per determinare se l'IP è noto per attività sospette o malevole.
- Se si tratta di un attore noto, è consigliabile bloccare l'IP tramite firewall o altre soluzioni di sicurezza.

---

4. Query: Tentativi di accesso falliti superiori a 5 volte
Descrizione:
La query ha analizzato gli indirizzi IP che hanno effettuato più di 5 tentativi di accesso falliti, ordinati per numero di tentativi.
Considerazioni:

- L'identificazione di indirizzi IP con accessi falliti ripetuti è un chiaro segnale di **attacchi brute force** o test di vulnerabilità.
- Gli indirizzi IP devono essere analizzati e, se confermati come malevoli, aggiunti a una lista di **blocchi temporanei o permanenti**.
- È importante verificare anche eventuali correlazioni tra questi IP e log di altre attività sospette (es. errori del server o accessi riusciti).

---

5. Query: Errori "Internal Server Error" (HTTP 500)
Descrizione:
La query ha rilevato errori HTTP 500, conteggiandoli per host e sorgente.
Considerazioni:

- Gli errori 500 possono indicare **problemi di configurazione**, **vulnerabilità sfruttabili** o **malfunzionamenti indotti da attacchi**.
- Un'alta frequenza di errori 500 associata ad altre attività sospette (ad esempio, tentativi di accesso falliti o traffico da IP non autorizzati) è un possibile indicatore di **tentativi di compromissione del server**.
- È consigliabile eseguire un'analisi approfondita sui moduli applicativi e i log del server web per identificare la causa.

Conclusioni Generali

1. **Indicatori di compromissione (IOCs):**
    - Gli indirizzi IP identificati nei tentativi di accesso falliti e negli errori di server devono essere considerati come potenziali IOC, da verificare tramite strumenti di analisi esterni (AbuseIPDB, VirusTotal, ecc.).
    - L'utente "djohnson" potrebbe essere un obiettivo specifico o un account già compromesso. È necessario verificare i log di accesso e monitorare il comportamento di questo account.
2. **Comportamenti sospetti:**
    - Un numero elevato di tentativi di accesso falliti (>5 per IP) e un IP specifico che tenta ripetutamente di accedere richiedono ulteriori indagini e misure di blocco.

- Gli errori HTTP 500 devono essere correlati con altri eventi per determinare se sono causati da attacchi o configurazioni errate.

3. **Azioni consigliate:**
   - **Bloccare gli IP malevoli identificati**: Utilizzare firewall o sistemi di prevenzione delle intrusioni (IPS).
   - **Monitorare account sensibili**: Verificare l'attività di account come "djohnson" per identificare anomalie.
   - **Correlare eventi**: Incrociare i log di accesso falliti, errori server e sessioni SSH aperte per identificare potenziali compromissioni.
   - **Aggiornare e applicare patch**: Assicurarsi che le vulnerabilità note siano risolte per ridurre il rischio di attacchi.
4. **Raccomandazioni future:**
   - Integrare Splunk con strumenti di threat intelligence per automatizzare il processo di identificazione degli IOC.
   - Utilizzare algoritmi di machine learning per il rilevamento di anomalie nei log e comportamenti sospetti.

Conclusione Finale:
Dai dati analizzati, i log mostrano chiari segnali di attività sospette, con tentativi di brute force, attività SSH anomala e possibili problemi al server. È necessaria un'azione immediata per mitigare i rischi, combinando misure di blocco, monitoraggio continuo e verifica delle configurazioni del sistema.

CONCLUSIONI FINALI SULL'ANALISI TRAMITE A.I.

Il report ha dimostrato che, anche in un ambiente simulato, un'analisi accurata dei log può fornire informazioni cruciali.

L'integrazione di queste tecniche con strumenti di intelligenza artificiale rappresenta una prospettiva promettente per migliorare l'efficienza delle analisi, soprattutto nelle attività di prevenzione e azioni di rimedio. Tuttavia, al momento della stesura di questo report, l'intelligenza artificiale presenta ancora limiti significativi: non è in grado di eseguire analisi dettagliate, approfondite e precise senza l'uso di prompt specifici e ben strutturati. **Sebbene l'analisi descritta nel paragrafo precedente abbia prodotto risultati piuttosto accurati, questi restano basati su probabilità e necessitano comunque di una verifica umana finale.**

In un contesto reale, l'intelligenza artificiale riduce sensibilmente i tempi di analisi, permettendo agli analisti umani di concentrarsi su minacce più complesse e di maggiore rilevanza.