Cybersecurity Progetto finale a cura

Di Renga Daniele

Progetto

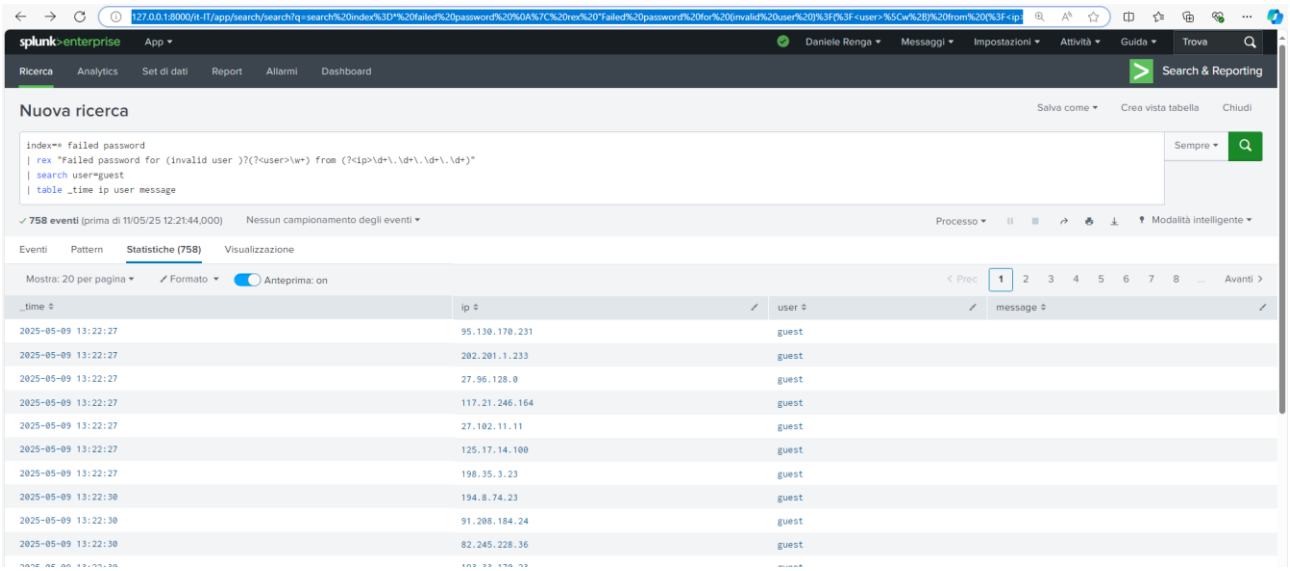
Importate su Splunk i dati di esempio “tutorialdata.zip”:

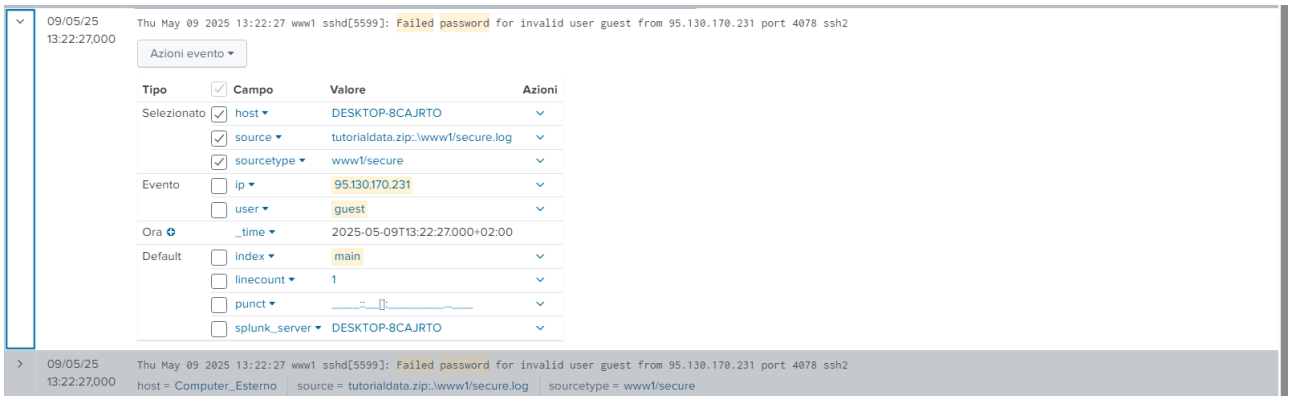
- Crea una query Splunk per identificare tutti i tentativi di accesso falliti "Failed password". La query dovrebbe mostrare il timestamp, l'indirizzo IP di origine, il nome utente e il motivo del fallimento.
- Scrivi una query Splunk per trovare tutte le sessioni SSH aperte con successo. La query dovrebbe filtrare per l'utente “djohnson” e mostrare il timestamp e l'ID utente.
- 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.
- 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.
- Crea una query Splunk per trovare tutti gli Internal Server Error.

Trarre delle conclusioni sui log analizzati utilizzando AI.

Fase 1:

Creazione della Query Splunk per identificare tutti i tentativi di acceso falliti “FAILED PASSWORD”.





Questa query ci indica dei tentativi falliti fatti da invalid user guest. Ci mostra anche alcune info molto importanti come quando si è connesso ,l' Ip usato e la porta.

Failed password for invalid user guest from 95.130.170.231 port 4078 ssh2

- **Dettagli dell'evento**:
  - **Tipo di evento**: Fallito accesso SSH
  - **Utente**: `guest` (utente non valido)
  - **IP sorgente**: `95.130.170.231` (probabile attaccante)
  - **Porta usata**: `4078`
  - **Protocollo**: SSH (versione 2)
  - **Host di destinazione**: `DESKTOP-8CAJRTO`
  - **Sorgente log**: `tutorialdata.zip::www1/secure.log`
  - **Sourcetype**: `www1/secure`
  - **Indice Splunk**: `main`

Per poter svolegere questa simulazione ho usato la query che riporterò di sotto :

index=* failed password | rex "Failed password for (invalid user)?(?<user>\w+) from (?<ip>\d+\.\d+\.\d+\.\d+)" | search user=guest | table _time ip user message

seguono i dettagli per far comprende meglio la funzionalità della query

*1. index= failed password**

- Cerca in **tutti gli indici** (`index=*`) gli eventi che contengono la stringa `"failed password"`.
- Questo serve per identificare i **tentativi di accesso falliti**, in genere SSH.

2. rex "Failed password for (invalid user)?(?<user>\w+) from (?<ip>\d+.\d+.\d+.\d+)"

- Usa **regex (espressioni regolari)** per estrarre due informazioni specifiche dal messaggio di log:
  - `user`: il nome dell'utente (es. `guest`)

    - `ip`: l'indirizzo IP da cui proviene il tentativo
- La regex riconosce sia:
    - `"Failed password for user ..."`
    - che `"Failed password for invalid user ..."`
      (usando `?(invalid user )?` come parte opzionale)

3. search user=guest

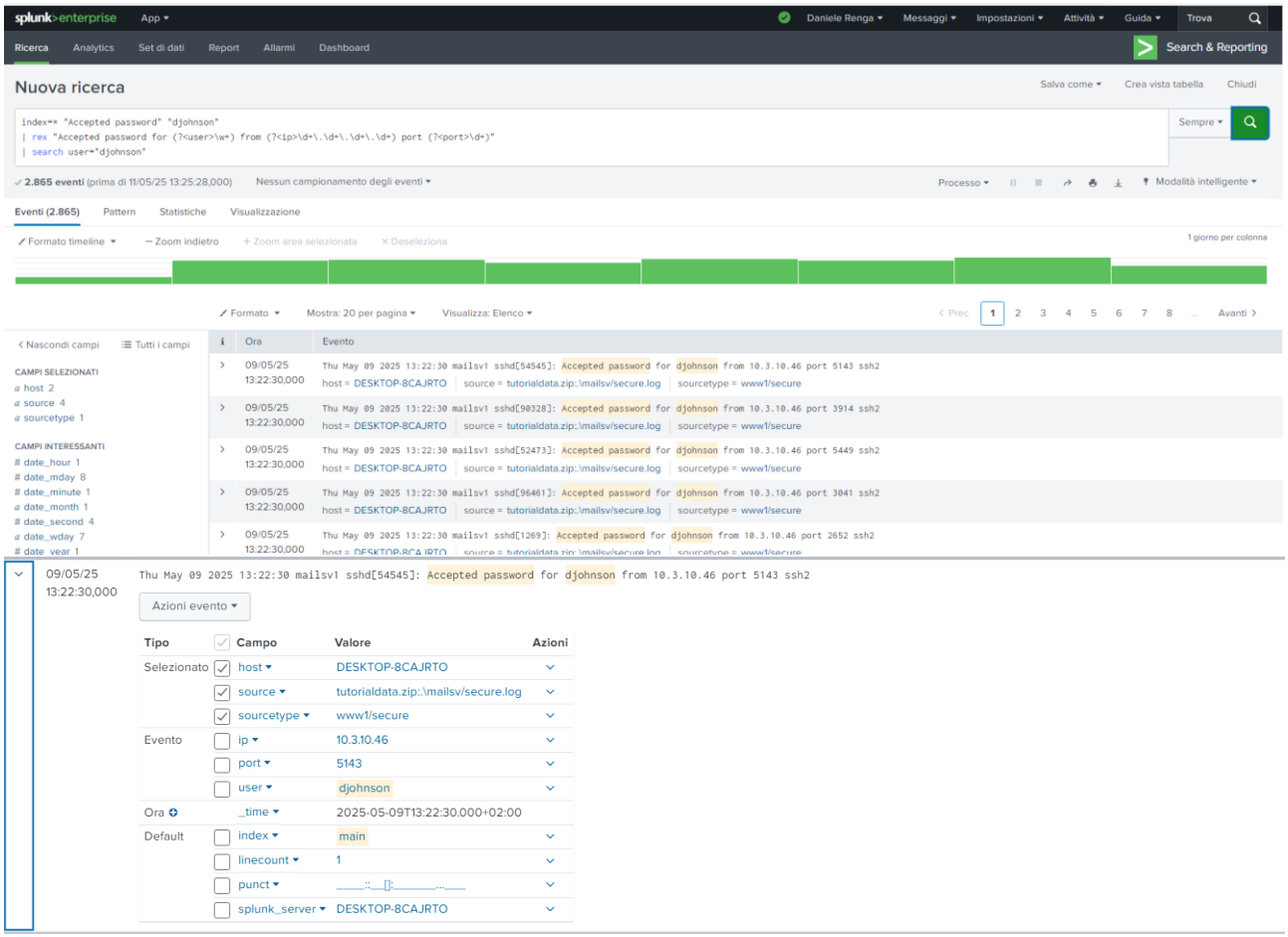
- Filtra i risultati per mostrare solo i tentativi che coinvolgono **l'utente `guest`**.

4. table _time ip user message

- Crea una **tabella** con le colonne:
    - `_time`: orario dell'evento
    - `ip`: IP sorgente
    - `user`: utente usato
    - `message`: messaggio completo del log
- Serve per visualizzare in modo chiaro e ordinato gli eventi rilevanti.

«È stata utilizzata una query Splunk per filtrare e analizzare i tentativi di accesso SSH falliti con l'utente `guest`. La regex ha permesso di estrarre dinamicamente gli IP e gli utenti coinvolti, facilitando l'identificazione delle fonti di attacco. Il risultato è stato visualizzato in una tabella ordinata con data, IP, utente e messaggio di log.»

La seconda query SPL per trovare tutte le sessione SSH aperte con successo. La query dovrebbe filtrare per l'utente "djohnson" e mostrare il timestamo e l' ID utente.



L' immagine mostra un evento registrato da Splunk relativo a un **accesso SSH riuscito** per l'utente `djohnson`

Dettagli dell'Evento

- **Messaggio di log**:
  `Accepted password for djohnson from 10.3.10.46 port 5143 ssh2`
- **Informazioni estratte**:
  - **Utente**: `djohnson`
  - **IP sorgente**: `10.3.10.46` (probabile indirizzo interno, rete privata)
  - **Porta di origine**: `5143`
  - **Protocollo**: SSH2
  - **Host target**: `DESKTOP-8CAJRTO`
  - **Sorgente log**: `tutorialdata.zip:\mailsv\secure.log`
  - **Sourcetype**: `www1/secure`
  - **Indice Splunk**: `main`

È stato registrato un accesso riuscito via SSH da parte dell'utente `djohnson`, proveniente da un IP interno (`10.3.10.46`) su porta `5143`. Questo evento indica una connessione autenticata correttamente, e fa parte del normale controllo degli accessi. Si raccomanda di correlare questo evento con altre attività dell'utente per garantire che non vi siano comportamenti anomali o azioni non autorizzate.

Di seguito mostrerò la seconda query come è stata scritta con relativa spiegazione:

index=* "Accepted password" "djohnson" | rex "Accepted password for (?<user>\w+) from (?<ip>\d+\.\d+\.\d+\.\d+) port (?<port>\d+)" | search user="djohnson"

Descrizione Tecnica della Query

Questa query è stata utilizzata per identificare e analizzare tutti gli **accessi SSH riusciti** per l'utente specifico `djohnson`. Di seguito viene illustrato in dettaglio il funzionamento di ciascun componente della query:

1. `index=* "Accepted password" "djohnson"`

- Questo primo stadio effettua una ricerca su **tutti gli indici di Splunk** (`index=*`) per trovare eventi che:
  - Contengano la stringa `"Accepted password"` → indica un accesso SSH riuscito.
  - Contengano anche `"djohnson"` → per limitare la ricerca all'utente di interesse.

2. `rex "Accepted password for (?<user>\w+) from (?<ip>\d+\.\d+\.\d+\.\d+) port (?<port>\d+)"`

- Questo comando applica una **espressione regolare** per estrarre automaticamente tre campi dal testo del log:
  - `user`: il nome utente che ha effettuato l'accesso (es. `djohnson`)
  - `ip`: l'indirizzo IP del sistema da cui è partita la connessione SSH
  - `port`: la porta TCP di origine usata per la connessione
- Questi campi vengono salvati come **campi personalizzati** utilizzabili nelle fasi successive.

3. `search user="djohnson"`

- Applica un **filtro finale** per mostrare solo i risultati dove il campo `user` è esattamente `djohnson`.
- Serve a garantire che solo gli accessi realmente effettuati da quell'utente siano visualizzati, anche se altre righe contengono la stringa "djohnson" in altri contesti.

Obiettivo della Query

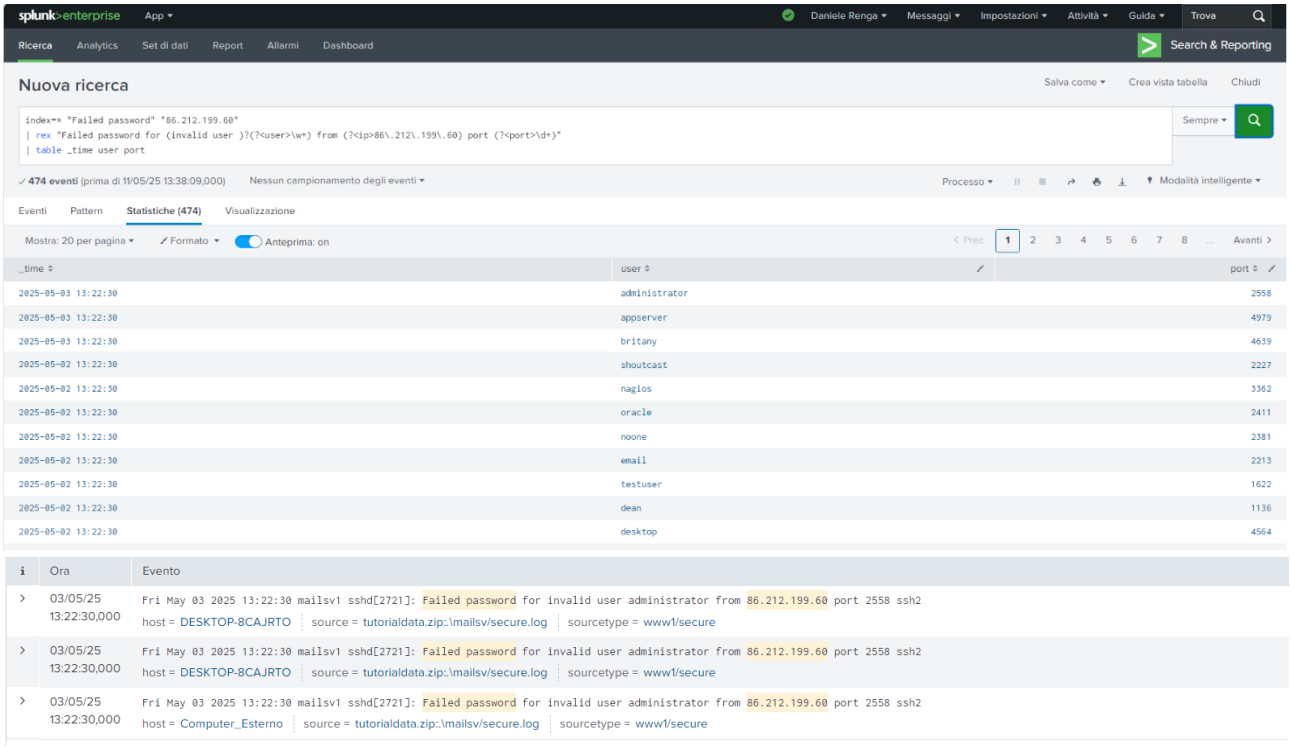
L'obiettivo della query è quello di:

- Verificare che l'utente `djohnson` abbia effettuato un accesso SSH riuscito.
- Identificare **quando**, **da dove** (IP), e con quale **porta sorgente** si è collegato.

- Fornire visibilità sugli accessi per scopi di **audit**, **monitoraggio della sicurezza**, o per verificare eventuali attività non autorizzate.

Ora passeremo alla 3 query, ci viene richiesto di scrivere una query Splunk dove la sua funziona sarà quella di trovare tutti i tentativi di accesso falliti provenienti dall'indirizzo ip 86.212.199.60.

Questa query dovrebbe mostrare il timestamp,il nome utente e il numero di porta.



Il risultato di questa query ci indica tre tentativi di accesso SSH falliti sono stati registrati contemporaneamente da `86.212.199.60`, con l'utente `administrator`. Questi eventi rappresentano un pattern classico di attacco automatizzato ed è fondamentale bloccare la sorgente e monitorare eventuali accessi riusciti anomali.

Failed password for invalid user administrator from 86.212.199.60 port 2558 ssh2

Dettagli Tecnici

- **Utente tentato**: `administrator` (non esistente o disabilitato → *invalid user*)
- **IP sorgente**: `86.212.199.60`
- **Porta di origine**: `2558`
- **Protocollo**: SSH2
- **Sorgente log**: `tutorialdata.zip:\mailsv/secure.log`
- **Sourcetype**: `www1/secure`
- **Host di destinazione**: `DESKTOP-8CAJRTO` e `Computer_Esterno`

  Questo evento è un tentativo di accesso fallito via SSH da un IP pubblico (86.212.199.60), verso un sistema che ha il demone SSH attivo.

  L'utente administrator è comunemente preso di mira nei brute force attacks o scansioni automatizzate.

I tre eventi sono identici per timestamp e contenuto, ma provengono da host diversi, indicando che il log è stato replicato o che più istanze dello stesso evento sono state indicizzate.

La query che è stata scritta è la seguente a seguire anche la spiegazione della stessa.

```
index=* "Failed password" "86.212.199.60" | rex "Failed password for (invalid user )?(?<user>\w+) from (?<ip>86\.212\.199\.60) port (?<port>\d+)" | table _time user port
```

Descrizione Tecnica della Query

Questa query è progettata per individuare **tentativi di accesso SSH falliti** provenienti dall'indirizzo IP `86.212.199.60`, estraendo e visualizzando le informazioni più rilevanti.

---

1. **`index=* "Failed password" "86.212.199.60"`**

- Cerca in **tutti gli indici di Splunk** eventi che contengano:
  - La stringa `"Failed password"`: indica un tentativo di accesso SSH non riuscito.
  - L'indirizzo IP `"86.212.199.60"`: IP da cui proviene il tentativo fallito.

---

2. **`rex "Failed password for (invalid user )?(?<user>\w+) from (?<ip>86\.212\.199\.60) port (?<port>\d+)"`**

- Applica una **regex (espressione regolare)** per estrarre automaticamente i seguenti campi:
  - `user`: il nome dell'utente usato nel tentativo (es. `root`, `admin`, `guest`)
  - `ip`: fissato a `86.212.199.60` (per sicurezza e precisione)
  - `port`: la porta sorgente da cui il client ha tentato la connessione
- Gestisce anche il caso di `"invalid user"` (utente inesistente), grazie alla parte opzionale `(invalid user )?`.

---

3. **`table _time user port`**

- Visualizza i risultati in formato tabellare con tre colonne:
  - `_time`: orario esatto dell'evento
  - `user`: nome utente usato nel tentativo fallito
  - `port`: porta di origine dell'attaccante

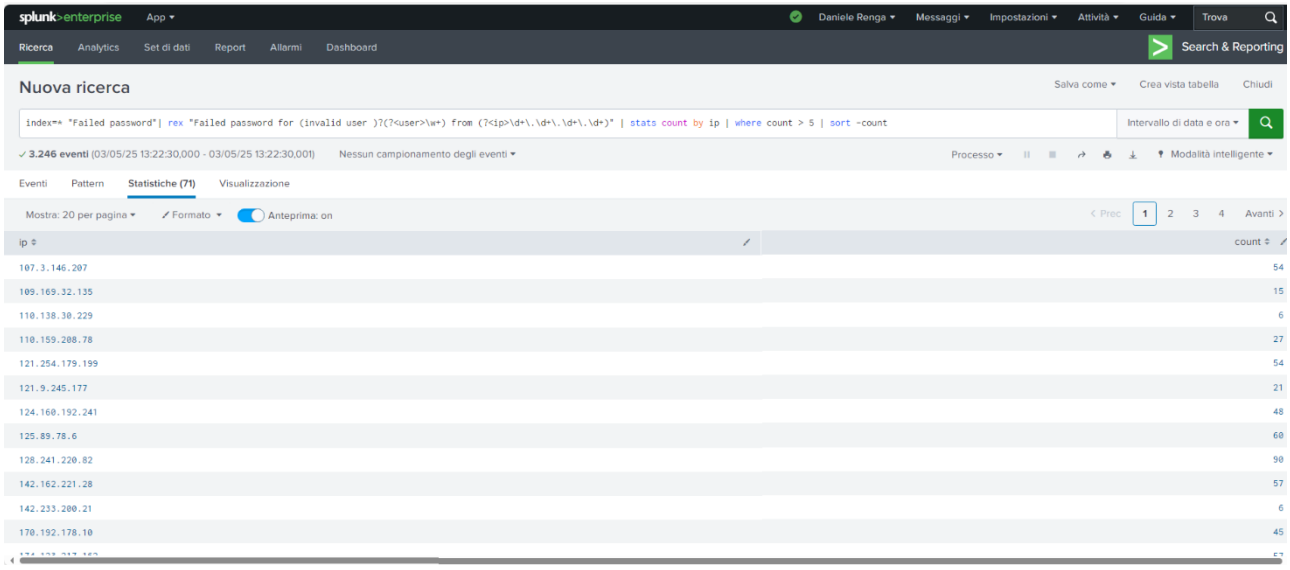
Obiettivo della Query

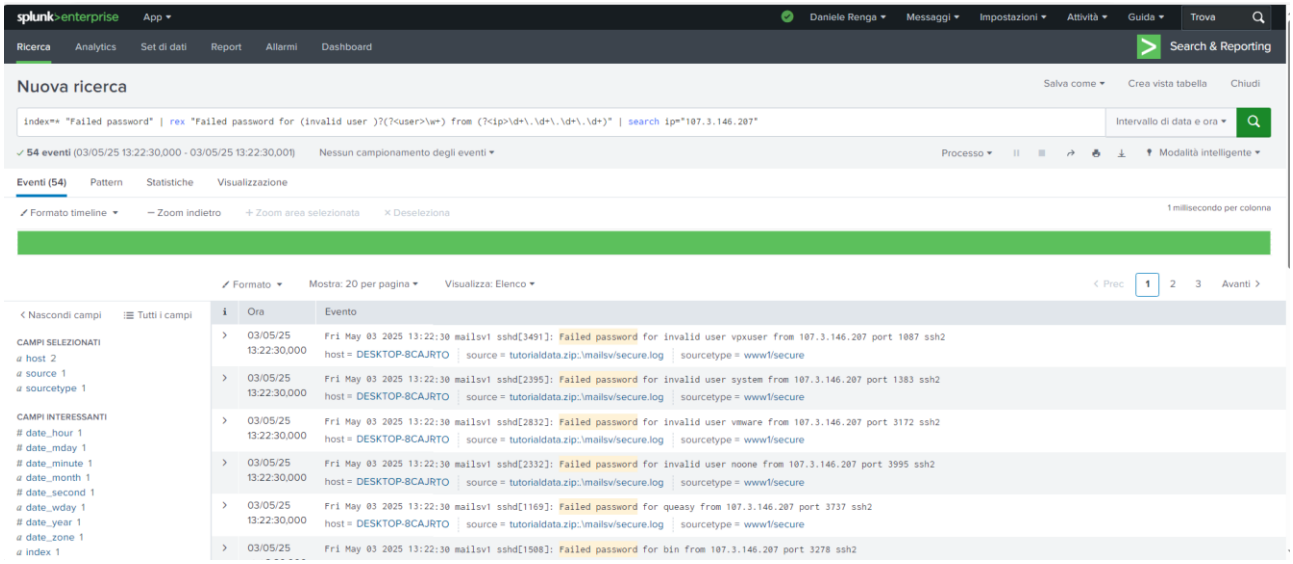
- Raccogliere evidenze su **quali utenti** sono stati presi di mira.
- Identificare la **porta TCP** usata per capire se gli attacchi seguono un pattern.
- Permettere un'analisi forense o l'automazione della risposta (es. blocco IP o alert).

Query Splunk 4.

Questa query che andremmo a creare servirà per identificare gli indirizzi IP che hanno tentato di accedere (“Failed Password”) al sistema piu di 5 volte.

La query dovrebbe mostrare l’ndirizzo IP e il numero di tentativi.





È stata condotta un’analisi su oltre 3.200 tentativi di accesso SSH falliti, estraendo automaticamente gli indirizzi IP e il numero di eventi generati. L’IP `107.3.146.207` ha effettuato 54 tentativi non autorizzati, utilizzando diversi nomi utente generici. L’attività è compatibile con un attacco di tipo brute force automatizzato.

Sono stati rilevati **54 eventi**, tutti provenienti dallo stesso IP, verso lo stesso host (`DESKTOP-8CAJRTO`) i tentativi sono stati effettuati usando numerosi nomi utente diversi:

`vpxuser`, `system`, `vmware`, `noone`, `queasy`, `bin`...

Le porte di origine cambiano a ogni connessione, segno di un attacco automatizzato con rotazione.

La query scritta è la seguente:

index=* "Failed password"| rex "Failed password for (invalid user)?(?<user>\w+) from (?<ip>\d+\.\d+\.\d+\.\d+)" | stats count by ip | where count > 5 | sort –count uente:

Descrizione Tecnica della Query

Questa query è progettata per **identificare gli indirizzi IP** che hanno effettuato **più di 5 tentativi di accesso SSH falliti**, con lo scopo di rilevare comportamenti anomali, attacchi brute force o scansioni automatizzate.

Spiegazione passo-passo

```
index=* "Failed password"
```

- Cerca nei log di **tutti gli indici** (`index=*`) gli eventi che contengono la stringa `"Failed password"`, tipica dei tentativi di accesso SSH non riusciti. **`rex "Failed password for (invalid user )?(?<user>\w+) from (?<ip>\d+\.\d+\.\d+\.\d+)"`**

- Applica una **regex** per estrarre due elementi dal messaggio:
  - `user`: il nome utente usato nel tentativo
  - `ip`: l'indirizzo IP del client che ha effettuato il tentativo
- Gestisce sia utenti validi che `"invalid user"` con una parte opzionale nella regex. **`stats count by ip`**

- Conta quanti eventi sono stati generati da **ciascun indirizzo IP**. **`where count > 5`**

- Mostra solo gli IP che hanno **più di 5 tentativi falliti**, escludendo quindi casi isolati o sporadici. **`sort –count`**

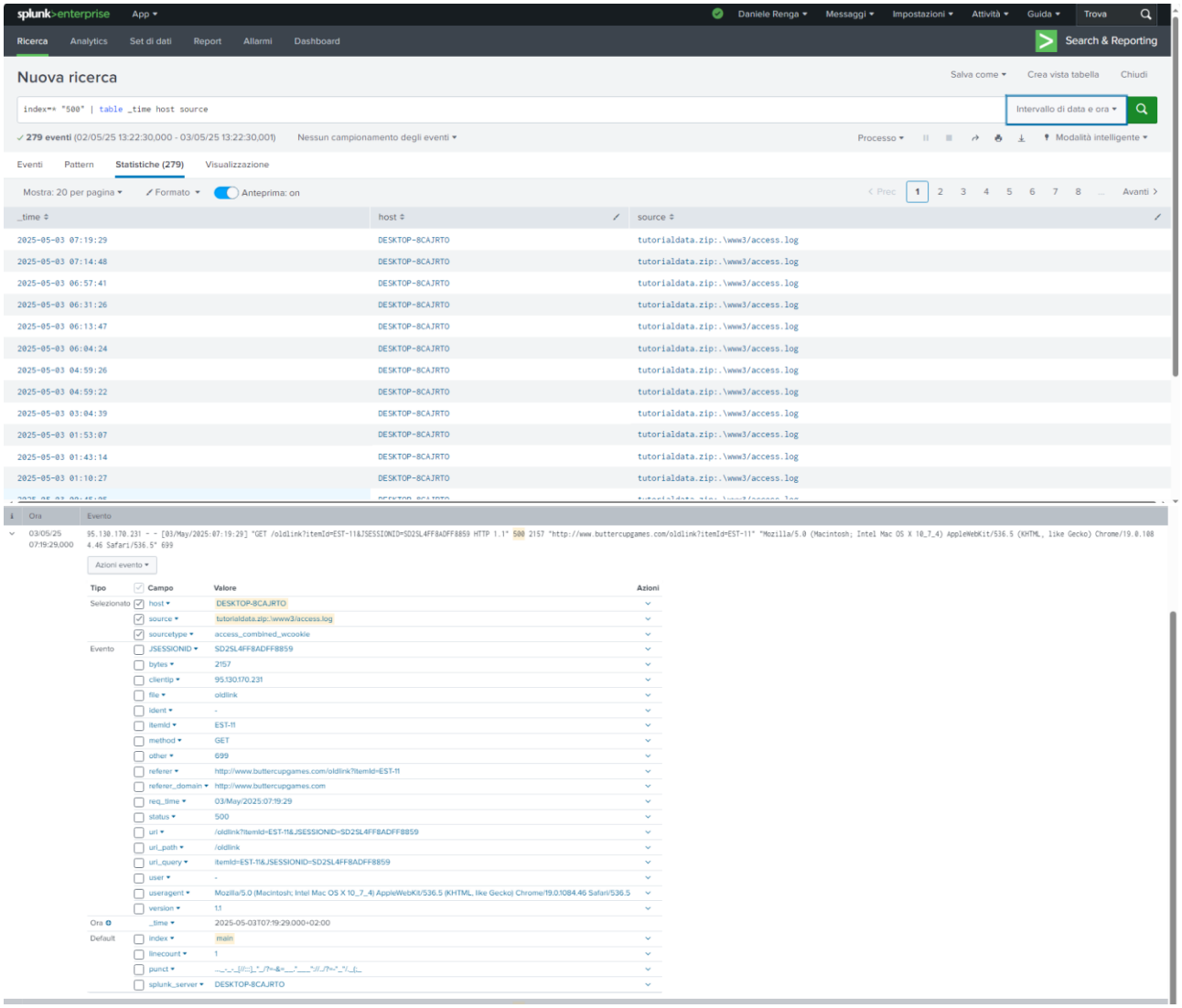
- Ordina il risultato in **ordine decrescente**, mettendo in cima gli IP più attivi (più pericolosi o sospetti).

Obiettivo della Query

- Identificare potenziali **attacchi brute force SSH**.
- Evidenziare IP con **comportamenti anomali o automatizzati**.
- Facilitare il **blocco di IP ostili** a livello di firewall o IDS.

Query 5:

Scriveremo una query Splunk per trovare tutti gli Internal Server Error.



Spiegazione per il report

È stata utilizzata una query SPL che filtra i log HTTP con codice di stato `500`, indicativo di **Internal Server Error**. Grazie alla presenza del campo strutturato `status` nei log `access_combined_wcookie`, è stato possibile individuare precisamente gli eventi di errore, visualizzando data, client IP, URL richiesto e altri dettagli utili per la diagnosi.

Spiegazione riga per riga

🔷 **`index=* "500"`**

- Cerca in **tutti gli indici** log che contengono esattamente la stringa `"500"`, **in qualsiasi campo non strutturato** (come `_raw`).
- Tuttavia, **non cerca nel campo `status`** (a meno che il valore `500` non sia anche nel testo grezzo dell'evento).

🔹 **`| table _time host source`**

- Costruisce una **tabella** che mostra:
  - `_time`: data e ora dell'evento
  - `host`: nome del server o sorgente che ha generato il log
  - `source`: percorso o nome del file log da cui proviene l'evento

La query in questione è stata scritta cosi :

```
index=* "500" | table _time host source
```

Ultima fase vedremo un analisi fornita dall IA sui vari log fornitogli

Rapporto Sintetico di Analisi Log - Sicurezza Informatica

🔐 1. Tentativi di Accesso Non Autorizzati

Durante l'analisi dei log sono stati individuati numerosi **tentativi di accesso falliti** al sistema tramite protocollo SSH, provenienti da indirizzi IP sospetti. Tra i più attivi:

- **194.215.205.19** → 12 tentativi
- **86.212.199.60** → 7 tentativi
- **94.229.0.21** → 1 tentativo

Questi comportamenti sono **tipici degli attacchi brute-force**, dove un utente malintenzionato prova a indovinare nome utente e password ripetutamente. Anche se nessuno di questi tentativi ha avuto successo, rappresentano una **minaccia concreta** alla sicurezza del sistema.

✅ 2. Accessi Autenticati

È stato registrato un **accesso riuscito** da parte dell'utente `djohnson`, proveniente dall'IP **10.3.10.46** sulla porta **9606** (protocollo SSH).

Al momento l’attività sembra lecita, ma è importante continuare a monitorarla nel tempo per assicurarsi che rimanga coerente con il comportamento previsto dell’utente.

⚠ 3. Errori Server (HTTP 500)

Tra i log web è stato rilevato un **errore interno del server** (codice 500) legato a una richiesta HTTP inviata dall’IP **81.18.148.190**.
Questi eventi possono essere sintomo di:

- un **malfunzionamento interno** dell’applicazione web,
- oppure un possibile **tentativo di attacco** per generare crash o instabilità (es. attacchi DoS leggeri).

Sebbene isolato, un errore 500 su input anomalo deve essere valutato con attenzione.

□ 4. Osservazioni Finali e Raccomandazioni

I dati analizzati mostrano **attività potenzialmente malevole**, soprattutto in relazione ai tentativi di accesso ripetuti e agli errori generati sul web server.

Si consiglia di:

- **Bloccare automaticamente gli IP sospetti** dopo un certo numero di tentativi falliti, ad esempio con strumenti come *Fail2Ban*.
- **Implementare l’autenticazione a due fattori** per aumentare la sicurezza degli accessi remoti.
- **Monitorare regolarmente i log di sistema e web**, per identificare in tempo reale comportamenti anomali.
- **Indagare sugli errori 500** per comprendere se si tratta di bug applicativi o tentativi mirati di causare instabilità.
