

# **W24D4 – Pratica**

## **Epic Education Srl**

### **Analisi del malware e Splunk**

**Simone Giordano**

30/12/2025



#### **Contatti:**

Tel: 3280063044

Email: [mynameisimone@gmail.com](mailto:mynameisimone@gmail.com)

Linkedin: <https://www.linkedin.com/in/simone-giordano-91290652/>

## Sommario

|   |          |
|---|----------|
| <b>Query Splunk.....</b>                      | <b>3</b> |
| Query 1.....                                  | 3        |
| Query 2.....                                  | 4        |
| Query aggiuntiva .....                        | 5        |
| Query 3.....                                  | 6        |
| Query 4.....                                  | 7        |
| Query 5.....                                  | 8        |
| <b>Conclusioni AI sui log analizzati.....</b> | <b>8</b> |

# Query Splunk

## Query 1

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.

Ho importato i file di tutorialdata tramite **Settings > Add Data > Upload**.

Ho inserito una query che mostra tutti i risultati contenenti il testo "Failed password".

Nell'immagine in basso possiamo visualizzare il timestamp (1), l'indirizzo IP di origine (2), l'utente (3) e il motivo del fallimento (4).

The screenshot shows a Splunk search interface with the following details:

- Search Bar:** source="tutorialdata.zip:/\*" | search "failed password"
- Results Summary:** ✓ 33,253 events (before 12/18/25 12:31:09.000 PM) No Event Sampling
- Event List:** The main pane displays a table of events. The first event is highlighted:

| Time                    | Event  |
|-------------------------|--|
| 12/17/25 6:45:49.000 AM | Thu Dec 17 2025 06:45:49 mailsv1 sshd[5276]: Failed password for invalid user appserver from 194.8.74.23 port 3351 ssh2<br>host = mailsv1 source = tutorialdata.zip./mailsv/secure.log sourcetype = secure-2 |
- Annotations:** Red numbers (1, 2, 3, 4) are overlaid on the screenshot to identify specific fields:
  - (1) points to the timestamp in the event table.
  - (2) points to the IP address in the event table.
  - (3) points to the user name in the event table.
  - (4) points to the error message "Failed password" in the event table.
- Left Panel:** Shows selected fields (host, source, sourcetype) and interesting fields (date\_hour, date\_minute, date\_month, date\_second, date\_wday, date\_year, date\_zone, index, linecount, punct, splunk\_server, timeendpos, timestamppos). It also includes a "+ Extract New Fields" section.

## Query 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.

Source="tutorialdata.zip | search "accepted password" AND "ssh2" AND "djohnson"

**source="tutorialdata.zip"** limita la ricerca ai dati contenuti nel file **tutorialdata.zip**, **- search** applica un filtro che combina le tre stringhe: "**accepted password**", "**ssh2**" "**djohnson**", tramite l'operatore logico **AND**, restituendo solo gli eventi che contengono tutte e tre le stringhe.

Il risultato mostra quindi tutti i tentativi di accesso SSH riusciti (accepted password) tramite protocollo SSH2 effettuati dall'utente djohnson.

The screenshot shows the Splunk interface with the following search command in the top bar:

```
source="tutorialdata.zip:*" | search "accepted password" AND "ssh2" AND "djohnson"
```

Below the search bar, it says "955 events (before 12/18/25 12:55:05.000 PM)" and "No Event Sampling".

The main area displays the search results in a table format:

| Events (955)            |   |  |
|-------------------------|---|--|
| Time                    | Event   |  |
| 12/17/25 6:45:49.000 AM | Thu Dec 17 2025 06:45:49 mailsv1 sshd[54545]: Accepted password for djohnson from 10.3.10.46 port 5143 ssh2 host = mailsv1 source = tutorialdata.zip:/mailsv/secure.log sourcetype = secure-2 |  |
| 12/17/25 6:45:49.000 AM | Thu Dec 17 2025 06:45:49 mailsv1 sshd[90328]: Accepted password for djohnson from 10.3.10.46 port 3914 ssh2 host = mailsv1 source = tutorialdata.zip:/mailsv/secure.log sourcetype = secure-2 |  |
| 12/17/25 6:45:49.000 AM | Thu Dec 17 2025 06:45:49 mailsv1 sshd[52473]: Accepted password for djohnson from 10.3.10.46 port 5449 ssh2 host = mailsv1 source = tutorialdata.zip:/mailsv/secure.log sourcetype = secure-2 |  |
| 12/17/25 6:45:49.000 AM | Thu Dec 17 2025 06:45:49 mailsv1 sshd[96461]: Accepted password for djohnson from 10.3.10.46 port 3041 ssh2 host = mailsv1 source = tutorialdata.zip:/mailsv/secure.log sourcetype = secure-2 |  |
| 12/17/25 6:45:49.000 AM | Thu Dec 17 2025 06:45:49 mailsv1 sshd[1269]: Accepted password for djohnson from 10.3.10.46 port 2652 ssh2 host = mailsv1 source = tutorialdata.zip:/mailsv/secure.log sourcetype = secure-2  |  |
| 12/17/25 6:45:49.000 AM | Thu Dec 17 2025 06:45:49 mailsv1 sshd[94708]: Accepted password for djohnson from 10.3.10.46 port 2408 ssh2 host = mailsv1 source = tutorialdata.zip:/mailsv/secure.log sourcetype = secure-2 |  |
| 12/17/25 6:45:49.000 AM | Thu Dec 17 2025 06:45:49 mailsv1 sshd[98104]: Accepted password for djohnson from 10.3.10.46 port 4577 ssh2 host = mailsv1 source = tutorialdata.zip:/mailsv/secure.log sourcetype = secure-2 |  |

## Query aggiuntiva

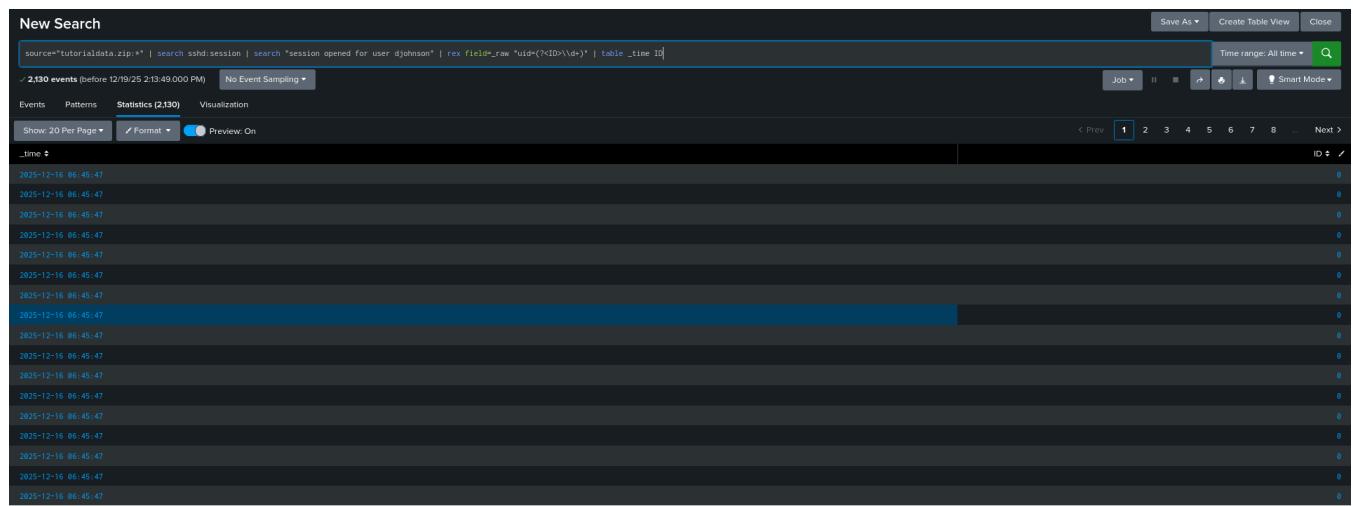
`source="tutorialdata.zip:*`" – Limita la ricerca ai log provenienti dal file sorgente **tutorialdata.zip**. Il simbolo \* indica **tutti i file contenuti** nell'archivio

| `search sshd:session` - Filtra gli eventi che contengono la stringa `sshd:session`

| `search "session opened for user djohnson"` - Seleziona solo gli eventi in cui la sessione SSH è stata aperta con successo e l'utente coinvolto è `djohnson`

| `rex field=_raw "uid=(?<ID>\d+)"` - Usa una espressione regolare (regex) sul campo `_raw` (log grezzo), estrae il valore numerico dopo `uid=` e salva il risultato in un nuovo campo chiamato `ID`

| `table _time ID` - Mostra il risultato in formato tabellare, visualizza solo: `_time` (timestamp dell'evento) e `ID` (user ID) estratti dal log.



The screenshot shows the Kibana interface with a search bar containing the query: `source="tutorialdata.zip:/*" | search sshd:session | search "session opened for user djohnson" | rex field=_raw "uid=(?<ID>\d+)" | table _time ID`. Below the search bar, it says `2,130 events (before 12/19/25 2:13:49.000 PM)` and `No Event Sampling`. The Statistics tab is selected, showing 2,130 events. The visualization is set to a table view. The table has two columns: `_time` and `ID`. The `_time` column lists dates from 2025-12-16 06:45:47 to 2025-12-16 06:45:47. The `ID` column lists values from 0 to 0. A single row in the table is highlighted with a blue background.

## Query 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.

**source="tutorialdata.zip"** limita la ricerca ai dati contenuti nel file **tutorialdata.zip**,  
- **search** applica un filtro che combina le tre stringhe: "**failed password**" e "**86.212.199.60**", tramite l'operatore logico **AND**, restituendo solo gli eventi che contengono tutte e due le stringhe.

Il risultato mostra quindi tutti i tentativi di accesso falliti (failed password) dall'IP 86.212.199.60, mostrando il time stamp (1), l'utente (2) e il numero di porta (3).

The screenshot shows a Splunk search interface with the following details:

- Search bar: `source="tutorialdata.zip:/*" | search "failed password" AND "86.212.199.60"`
- Event count: 158 events (before 12/18/25 12:59:46.000 PM)
- Sampling: No Event Sampling
- Panel tabs: Events (158), Patterns, Statistics, Visualization
- Timeline format: Selected
- Event list:
  - Time: 12/17/25 6:45:49.000 AM, Event: Thu Dec 17 2025 06:45:49 mailsv1 sshd[5728]: Failed password for invalid user agushto from 86.212.199.60 port 3692 ssh2. Fields: host = mailsv1; source = tutorialdata.zip:/mailsv/secure.log; sourcetype = secure-2. (1) (2) (3)
  - Time: 12/17/25 6:45:49.000 AM, Event: Thu Dec 17 2025 06:45:49 mailsv1 sshd[4843]: Failed password for invalid user tomcat from 86.212.199.60 port 1464 ssh2. Fields: host = mailsv1; source = tutorialdata.zip:/mailsv/secure.log; sourcetype = secure-2.
  - Time: 12/17/25 6:45:49.000 AM, Event: Thu Dec 17 2025 06:45:49 mailsv1 sshd[5718]: Failed password for invalid user desktop from 86.212.199.60 port 3518 ssh2. Fields: host = mailsv1; source = tutorialdata.zip:/mailsv/secure.log; sourcetype = secure-2.
  - Time: 12/17/25 6:45:49.000 AM, Event: Thu Dec 17 2025 06:45:49 mailsv1 sshd[1008]: Failed password for invalid user yp from 86.212.199.60 port 2856 ssh2. Fields: host = mailsv1; source = tutorialdata.zip:/mailsv/secure.log; sourcetype = secure-2.
  - Time: 12/17/25 6:45:49.000 AM, Event: Thu Dec 17 2025 06:45:49 mailsv1 sshd[5878]: Failed password for mail from 86.212.199.60 port 1054 ssh2. Fields: host = mailsv1; source = tutorialdata.zip:/mailsv/secure.log; sourcetype = secure-2.
  - Time: 12/17/25 6:45:49.000 AM, Event: Thu Dec 17 2025 06:45:49 mailsv1 sshd[2649]: Failed password for apache from 86.212.199.60 port 2630 ssh2. Fields: host = mailsv1; source = tutorialdata.zip:/mailsv/secure.log; sourcetype = secure-2.
  - Time: 12/17/25 6:45:49.000 AM, Event: Thu Dec 17 2025 06:45:49 mailsv1 sshd[2079]: Failed password for invalid user services from 86.212.199.60 port 4740 ssh2. Fields: host = mailsv1; source = tutorialdata.zip:/mailsv/secure.log; sourcetype = secure-2.
- Selected Fields:
  - host 4
  - source 4
  - sourcetype 1
- Interesting Fields:
  - # date\_hour 1
  - # date\_mday 7
  - # date\_minute 1
  - # date\_month 1
  - # date\_second 4
  - # date\_wday 6
  - # date\_year 1
  - # date\_zone 1
  - index 1
  - linecount 1
  - punct 2
  - splunk\_server 1
  - timeendpos 1

## Query 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.

```
source="tutorialdata.zip:/*" | search "failed password" | rex field=_raw "from (?<ip>[\d\.]*)" | stats count AS failed_attempts by ip | where failed_attempts > 5 | sort - failed_attempts | table ip failed_attempts
```

source="tutorialdata.zip:/\*" - Dice a Splunk di cercare dentro il file tutorialdata.zip, in tutti i file contenuti  
search "failed password" - Mostra solo i log che contengono la stringa "failed password" (tentativi di login falliti)

rex field=\_raw "from (?<ip>[\d\.]\*)" – Regex che cerca nel testo del log (\_raw) un IP dopo la parola from e lo salva nel campo ip

stats count AS failed\_attempts by ip - Conta quanti tentativi falliti ci sono per ogni indirizzo IP

where failed\_attempts > 5 - Mostra solo gli IP che hanno più di 5 tentativi falliti

sort - failed\_attempts - Ordina gli IP dal numero più alto al più basso di tentativi falliti

table ip failed\_attempts - Mostra solo le colonne IP e numero di tentativi falliti

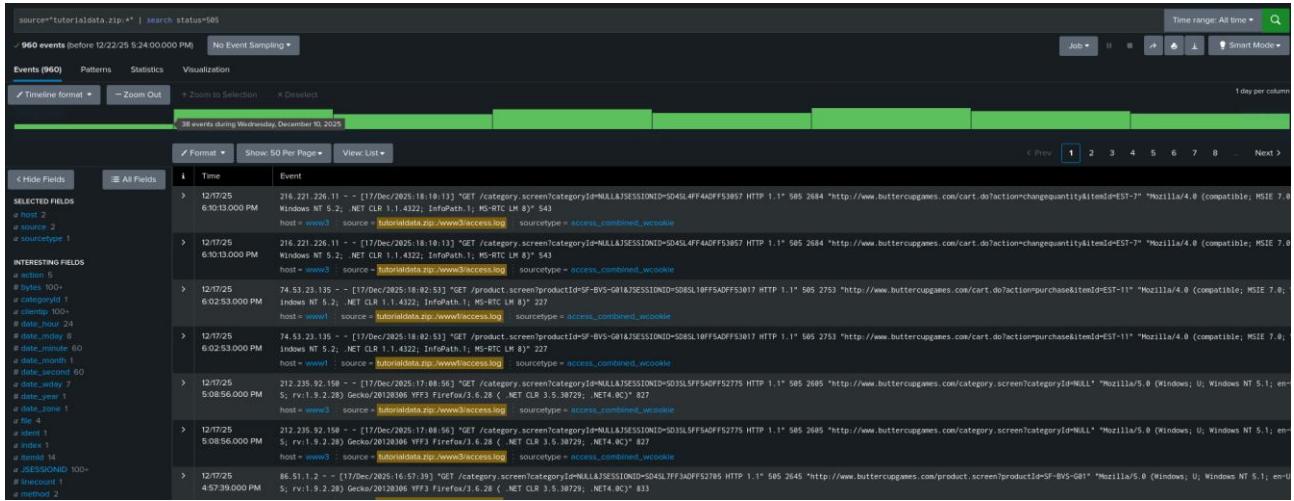
| New Search  |                                   |
|---|-----------------------------------|
| <pre>source="tutorialdata.zip:/*"   search "failed password"   rex field=_raw "from (?&lt;ip&gt;[\d\.]*)"   stats count AS failed_attempts by ip   where failed_attempts &gt; 5   sort - failed_attempts   table ip failed_attempts</pre> |                                   |
| Events  | Save As ▾ Create Table View Close |
| Patterns  | Time range: All time ▾            |
| Statistics (1B9)  | Job ▾                             |
| Visualization   | Smart Mode                        |
| Show: 100 Per Page ▾  | < Prev 1 2 Next >                 |
| Format ▾  | failed_attempts ▾                 |
| Preview: On   |                                   |
| ip  | ↳                                 |
| 87.194.216.51   | 1895                              |
| 211.166.11.181  | 1486                              |
| 128.241.228.82  | 1244                              |
| 189.169.32.135  | 1038                              |
| 194.215.205.19  | 1028                              |
| 216.221.226.11  | 866                               |
| 188.138.40.166  | 594                               |
| 65.19.167.54  | 572                               |
| 107.3.146.287   | 564                               |
| 95.138.170.231  | 558                               |
| 223.295.219.67  | 548                               |
| 27.1.11.11  | 546                               |
| 27.38.31.11   | 540                               |
| 59.162.167.190  | 534                               |
| 91.218.104.143  | 506                               |
| 27.181.11.11  | 502                               |

## Query 5

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

Gli Internal Server Error hanno come codice di stato 505, quindi per filtrare i risultati ho utilizzato la seguente query.

Source="tutorialdata.zip:/\*" | search status=505



## Conclusioni AI sui log analizzati

Trarre delle conclusioni sui log analizzati utilizzando AI.

Dalle conclusioni tratte dai log forniti, sembra esserci un problema diffuso con richieste API non valide che generano errori HTTP 400.

### Analisi dei Log

- Errori HTTP 400 Diffusi:** I log mostrano un numero significativo di risposte con codice di stato **400 Bad Request**. Questo indica che le richieste inviate dal client al server non sono state comprese o non erano valide a causa di sintassi errata, parametri mancanti o dati non corretti.
- Servizio AWS Coinvolto:** L'intestazione x-amzn-errortype suggerisce che il servizio in questione sia gestito su **Amazon Web Services (AWS)**, probabilmente tramite Amazon API Gateway o servizi simili.
- Tipi di Richieste (Metodi HTTP):** Vengono registrate richieste di tipo POST, GET, PUT e DELETE. Le richieste POST e PUT sono spesso utilizzate per inviare o aggiornare dati, mentre GET e DELETE per recuperare o rimuovere risorse.
- Attività Utente Sospetta/Ripetitiva:** La ripetizione frequente degli errori 400 potrebbe indicare:
  - Un client mal configurato che invia costantemente richieste errate.
  - Un tentativo di accesso fallito ripetuto troppe volte.
  - Un'attività anomala o un attacco informatico (come un tentativo di brute force o fuzzing) dato il contesto di log di sicurezza/monitoraggio visualizzato.

I log documentano principalmente una serie di interazioni fallite con un'API AWS, probabilmente a causa di richieste formate male da parte del client.