



Analisi del malware e Splunk

Jimsop Dario Barcia

Alfredo Verduci

Federico Presti

Guyphard Ndombasi

CSPT0424 - M6

Mars 2025



Sommario

Sommario	1
Traccia	3
Svolgimento	4
0. Installazione e configurazione Splunk Enterprise	5
1. Prima Domanda	8
1.1. Query Utilizzata	8
1.2. Breve Spiegazione ed Analisi dettagliata della query	8
1.3. Output atteso della query	10
1.4. Analisi effettuata da ChatGPT caricando l'intero file dei log	10
1.5. Conclusione	11
2. Seconda domanda	12
2.1. Query utilizzata	12
2.2. Output atteso della query	13
2.3. Analisi effettuata da ChatGPT esportando l'intero file dei log	14
2.4. Conclusione	15
3. Terza Domanda:	16
3.1. Query utilizzata	16
3.2. Breve spiegazione ed Analisi Dettagliata della query	16
3.3. Output atteso	17
3.4. Analisi effettuata da ChatGPT caricando l'intero file dei log	18
3.5. Conclusione	20
4. Quarta Domanda:	21
4.1. Query utilizzata	21
4.2. Breve spiegazione ed Analisi dettagliata della query	21
4.3. Analisi effettuata da ChatGpt esportando l'intero file di log	22
4.4. Conclusione	22
5. Quinta Domanda:	23
5.1. Query utilizzata	23
5.2. Breve Spiegazione ed Analisi dettagliata della query	23
5.3. Output atteso	24
5.4. Analisi effettuata da ChatGpt esportando l'intero file di log	25
5.5. Conclusione	27
Considerazioni Finali di Prevenzione e Breve Conclusione	28

Figura 1 : Log in Splunk	5
Figura 2 : Upload file ZIP tutorialdata	6
Figura 3 : Select file nel percorso Windows.....	6
Figura 4 : Upload file ZIP completata.....	7
Figura 5 : Immagine dell'output atteso query 1	8
Figura 6 : Dettaglio motivo del fallimento	10
Figura 7 : Immagine dell'output atteso query 2	12
Figura 8 : Dettaglio su una sessione SSH aperta.....	14
Figura 9 : Immagine dell'output atteso query 3	16
Figura 10 : Immagine dell'output atteso query 3 con dettaglio di tentativo di un accesso fallito	18
Figura 11 : Immagine con Output su Splunk 4.....	21
Figura 12 : Immagine con Output su Splunk 5.....	23
Figura 13 : Immagine dell'output atteso query 5 con tutti gli Internal Server Error.....	25

Traccia

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.

Svolgimento

Breve Introduzione al Progetto, Splunk e strumenti I.A. come ChatGpt nell'analisi delle query

In questo progetto affronteremo l'analisi dei log di sistema per capire attività sospette, intrusioni e minacce malware.

In questo scenario, il nostro team SOC, **Allsave Cybersecurity**, andrà ad analizzare i file di log contenuti in *tutorialdata.zip*, come se stessimo simulando un'indagine sui sistemi di un'azienda fittizia che chiameremo **AvilCorp**.

L'obiettivo è identificare tentativi di accesso non autorizzati e altre anomalie nei log, utilizzando **Splunk Enterprise**, una piattaforma avanzata di analisi dei log che sfrutta il **Search Processing Language (SPL)** per interrogare grandi volumi di dati in tempo reale.

In **Splunk Search Processing Language (SPL)** infatti, ogni comando prende in input i risultati del comando precedente e li elabora ulteriormente in modo da costruire query complesse in modo modulare ed efficiente.

L'uso della **pipe** | in Splunk Enterprise è quindi essenziale per **raffinare progressivamente i risultati** e nel corso di questo Report ci faciliterà l'analisi dei dati e la rilevazione di minacce o anomalie nei log di sistema.

La traccia sviluppata da **Epicode school of technology** è creata per rispondere concretamente a specifici interrogativi sulla sicurezza dei sistemi informatici e che possiamo riepilogare come

1. **Individuazione di tentativi di accesso falliti**
2. **Identificazione delle sessioni SSH aperte con successo**
3. **Analisi dei tentativi di accesso falliti da un IP specifico**
4. **Rilevamento di attacchi a forza bruta**
5. **Monitoraggio degli errori critici del server**

L'uso di strumenti avanzati come Splunk in questo progetto M6 incentrato su "IA e Cybersecurity" è quindi fondamentale per il **threat detection** e il **monitoraggio proattivo** dei sistemi informatici:

Splunk con la sua capacità di raccogliere, indicizzare e correlare i dati dei log in tempo reale e che consente quindi ai team SOC di rilevare anomalie, rispondere rapidamente alle minacce e implementare misure di sicurezza preventive.

I.A. con l'utilizzo di ChatGPT PLUS, una L.L.M (Large Language Model) basata su una variante di G.P.T. (Generative Pre-Trained Transformer) addestrata su di un vasto dataset per la

modellazione del linguaggio naturale (NLP - Natural Language Processing) leader sul campo per la sua capacità di analisi avanzata, automazione ed assistenza strategica

La versione di Splunk utilizzata in questa esercitazione è la 9.4.0 e la potete scaricare da qui: https://www.splunk.com/en_us/download.html

Chat GPT è disponibile qui: <https://chatgpt.com/>

0. Installazione e configurazione Splunk Enterprise

Abbiamo effettuato la registrazione prima di scaricare il **pacchetto di installazione** via il sito ufficiale di Splunk: <https://www.splunk.com>.

Una volta scaricato il file, avviamo e completiamo l'installazione per avere la possibilità di accedere all'interfaccia web : <http://localhost:8000>.

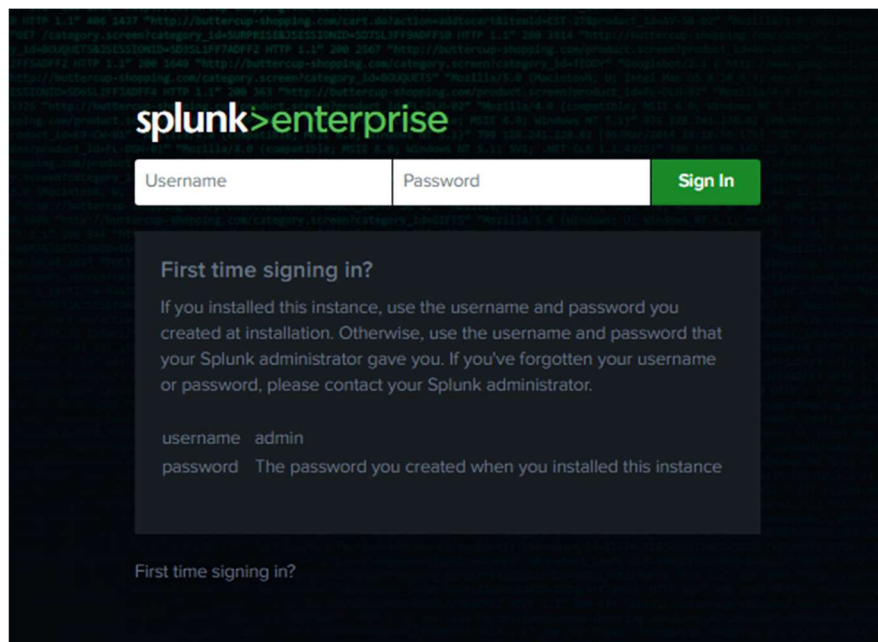


Figura 1 : Log in Splunk

Facciamo upload di **tutorialdata.zip**

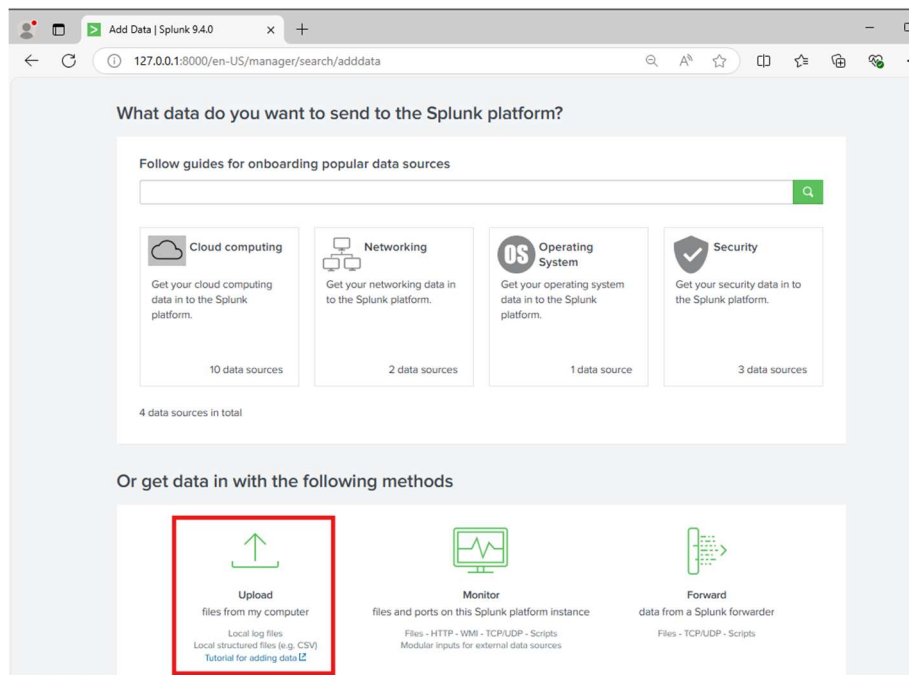


Figura 2 : Upload file ZIP tutorialdata

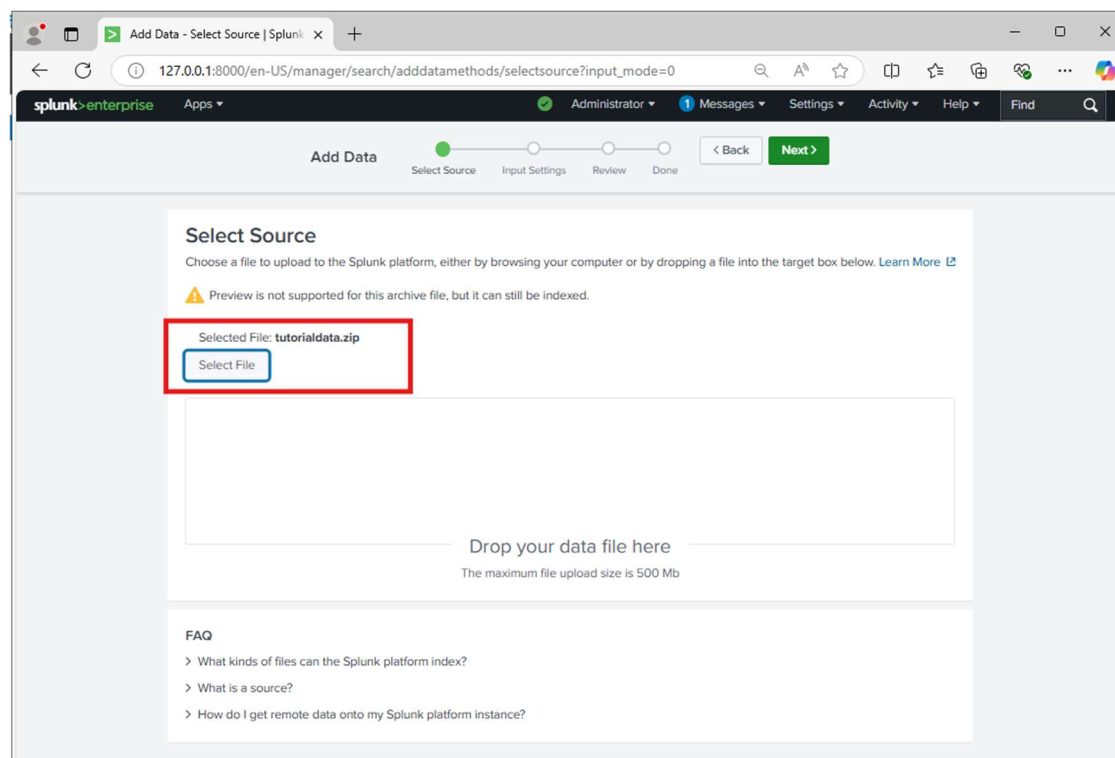


Figura 3 : Select file nel percorso Windows

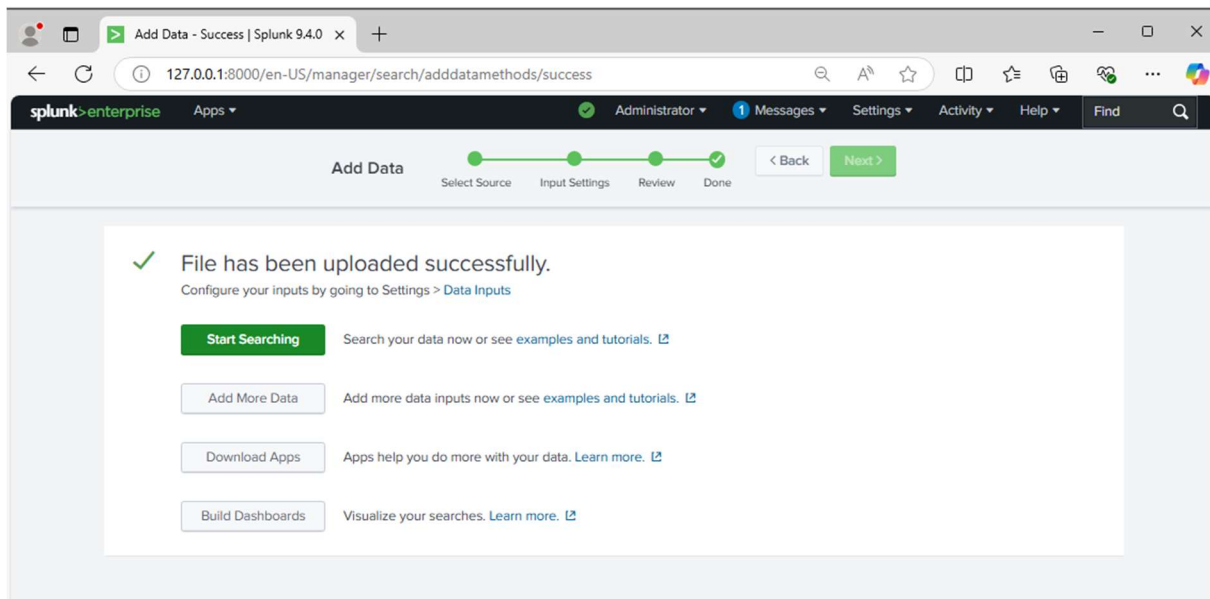


Figura 4 : Upload file ZIP completata

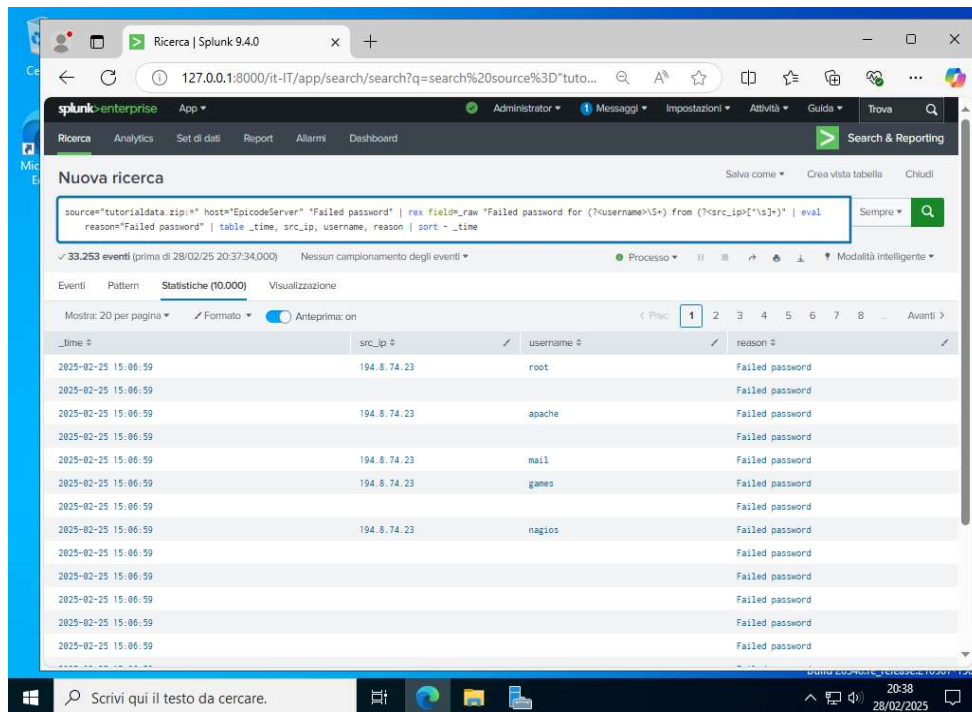
1. Prima Domanda

Traccia:

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.

1.1. Query Utilizzata

source="tutorialdata.zip:" host="EpicodeServer" "Failed password" | rex field=_raw "Failed password for (?<username>\S+) from (?<src_ip>[\^\\s]+)" | eval reason="Failed password" | table _time, src_ip, username, reason | sort - _time*



_time	src_ip	username	reason
2025-02-25 15:06:59	194.8.74.23	root	Failed password
2025-02-25 15:06:59	194.8.74.23		Failed password
2025-02-25 15:06:59	194.8.74.23	apache	Failed password
2025-02-25 15:06:59			Failed password
2025-02-25 15:06:59	194.8.74.23	mail	Failed password
2025-02-25 15:06:59	194.8.74.23	games	Failed password
2025-02-25 15:06:59			Failed password
2025-02-25 15:06:59	194.8.74.23	nagios	Failed password
2025-02-25 15:06:59			Failed password
2025-02-25 15:06:59			Failed password
2025-02-25 15:06:59			Failed password
2025-02-25 15:06:59			Failed password
2025-02-25 15:06:59			Failed password

Figura 5 : Immagine dell'output atteso query 1

1.2. Breve Spiegazione ed Analisi dettagliata della query

Questa query è progettata per identificare i tentativi di accesso falliti su il nostro server EpicodeServer, estrarre l'IP di origine, il nome utente coinvolto e mostrare il motivo del fallimento. Inoltre, i risultati vengono ordinati dal più recente al più vecchio.

Questa la costruzione della nostra query:

- a. [source="tutorialdata.zip:*"](#)

Questa parte della query dice a Splunk di cercare i dati all'interno di un file ZIP, chiamato "tutorialdata.zip". Il `:*` permette di cercare in qualsiasi file contenuto all'interno dell'archivio ZIP, indipendentemente dall'estensione.

Assicura che la ricerca avvenga all'interno dell'archivio ZIP.

- b. [host="EpicodeServer"](#)

Restringe la ricerca, filtra i log e mostra solo quelli provenienti dall'host **"EpicodeServer"** ignorando gli altri.

- c. ["Failed password"](#)

Cerca nei log solo le righe che contengono **"Failed password"**. Questo è un messaggio tipico nei log di autenticazione SSH quando un utente inserisce una password errata. Filtra i log per mostrare solo i tentativi di accesso falliti.

- d. [rex field= raw "Failed password for \(?<username>\S+\) from \(?<src_ip>\[^\s\]+\)"](#)

“rex” viene usato per applicare un'espressione regolare (regex) e estrarre informazioni dal campo `_raw` (che contiene il testo completo del log). Questo pezzo di query estrae e salva il nome utente e l'IP di origine dai log di autenticazione falliti.

- e. [eval reason="Failed password"](#)

“eval” crea un campo `reason` per rendere più chiaro il motivo dell'errore.

- f. [table time, src_ip, username, reason](#)

Mostra i dati più importanti in una tabella leggibile. Seleziona solo i campi essenziali per una visualizzazione pulita:

- o `_time` → Il timestamp dell'evento.
- o `src_ip` → L'indirizzo IP di origine dell'utente che ha tentato l'accesso.
- o `username` → Il nome utente utilizzato nel tentativo fallito.
- o `reason` → Il motivo dell'errore, impostato su "Failed password".

- g. [sort - time](#)

Assicura che gli accessi falliti più recenti siano in cima alla lista.

1.3 Output atteso della query

Timestamp	IP Origine	Nome Utente	Motivo del Fallimento
2025-02-28 10:30	192.168.1.100	admin	Failed password
2025-02-28 10:20	203.0.113.50	root	Failed password
2025-02-28 10:15	198.51.100.75	user1	Failed password

Cliccando una riga, si possono visualizzare i dettagli:

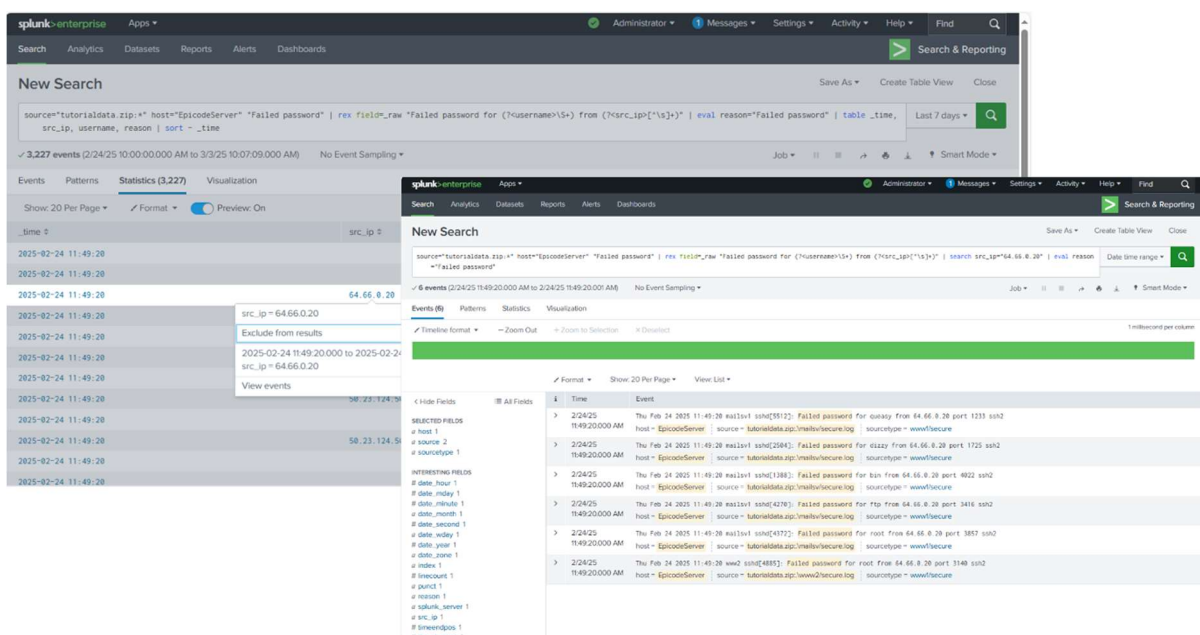


Figura 6 : Dettaglio motivo del fallimento

1.4 Analisi effettuata da ChatGPT caricando l'intero file dei log

Dall'analisi dei dati ottenuti, emergono le seguenti osservazioni:

- **Volume degli Accessi Falliti:** Sono stati registrati numerosi tentativi di accesso falliti in un breve intervallo di tempo, indicando la possibilità di un attacco brute force.
- **Indirizzi IP Coinvolti:** Gli accessi provengono da IP sospetti, tra cui:
 - 194.8.74.23
 - 203.45.206.135
 - 89.106.20.218
 - 69.175.97.11
 - 95.130.170.231

- **Account Presi di Mira:** Gli utenti più frequentemente coinvolti nei tentativi di accesso falliti includono:
 - root (account privilegiato, target comune negli attacchi)
 - apache
 - mail
 - nagios
 - games
- **Tipologia dell'Attacco:** Gli attacchi sembrano essere sistematici e mirati a conti con permessi amministrativi o con accesso a servizi sensibili.

Implicazioni e Rischi

- **Attacchi Brute Force:** L'alto numero di tentativi di accesso falliti in un breve periodo suggerisce un possibile attacco brute force.
- **Compromissione degli Account:** Se l'attacco avesse successo, gli attaccanti potrebbero ottenere accesso non autorizzato ai sistemi critici.
- **Utilizzo di IP Distribuiti:** Gli indirizzi IP provengono da diverse aree geografiche, suggerendo l'impiego di botnet o proxy anonimi per mascherare l'origine dell'attacco.

L'analisi dei log di autenticazione ha evidenziato un numero significativo di tentativi di accesso falliti, probabilmente legati a un attacco brute force. Implementare misure di sicurezza adeguate, come il blocco degli IP malevoli e l'uso di MFA, può ridurre il rischio di compromissione.

1.5 Conclusione

Abbiamo visto che la query ci ha permesso di:

- Identificare tutti i tentativi di accesso SSH falliti.
- Estrarre automaticamente il nome utente e l'IP di origine.
- Organizzare i dati in una tabella chiara con timestamp e motivo del fallimento.
- Ordinare i tentativi dal più recente al più vecchio.

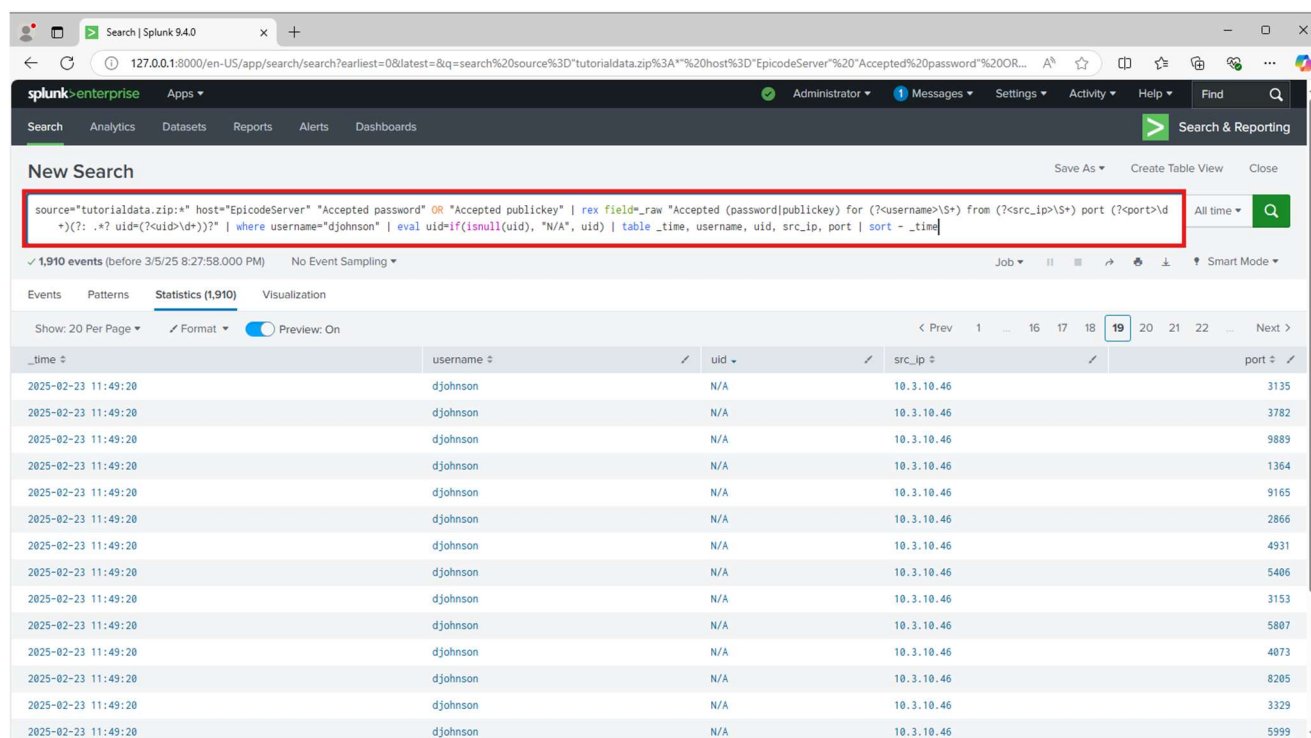
2 Seconda domanda

Traccia:

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.

2.1 Query utilizzata

```
source="tutorialdata.zip:*" host="EpicodeServer" "Accepted password" OR "Accepted publickey" | rex field=_raw "Accepted (password|publickey) for (?<username>\S+) from (?<src_ip>\S+) port (?<port>\d+)(?: .*? uid=(?<uid>\d+))?" | where username="djohnson" | eval uid=if(isnull(uid), "N/A", uid) | table _time, username, uid, src_ip, port | sort - _time
```



The screenshot shows the Splunk Search interface. The search bar contains the query: `source="tutorialdata.zip:*" host="EpicodeServer" "Accepted password" OR "Accepted publickey" | rex field=_raw "Accepted (password|publickey) for (?<username>\S+) from (?<src_ip>\S+) port (?<port>\d+)(?: .*? uid=(?<uid>\d+))?" | where username="djohnson" | eval uid=if(isnull(uid), "N/A", uid) | table _time, username, uid, src_ip, port | sort - _time`. The results table shows 19 events, all with the same timestamp (2025-02-23 11:49:20) and username (djohnson). The uid column contains "N/A" for all events. The src_ip column shows "10.3.10.46" for all events. The port column shows various values: 3135, 3782, 9889, 1364, 9165, 2866, 4931, 5406, 3153, 5807, 4073, 8205, 3329, and 5999.

_time	username	uid	src_ip	port
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	3135
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	3782
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	9889
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	1364
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	9165
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	2866
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	4931
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	5406
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	3153
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	5807
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	4073
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	8205
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	3329
2025-02-23 11:49:20	djohnson	N/A	10.3.10.46	5999

Figura 7 : Immagine dell'output atteso query 2

Breve Spiegazione ed Analisi dettagliata della query:

La query così strutturata ci mostra in dettaglio il numero di sessioni SSH aperte in un intervallo di tempo mirato in questo modo:

- `source="tutorialdata.zip:*" e host="EpicodeServer"`

Si assicura di cercare nei dati provenienti da un archivio ZIP specifico. E filtra solo gli eventi provenienti dall'host EpicodeServer, evitando dati irrilevanti da altri server.

- `"Accepted password" OR "Accepted publickey"`

Trova tutti gli eventi di autenticazione SSH avvenuta con successo. Cerca nel log tentativi di accesso riusciti:

- "Accepted password" Accesso riuscito con password.
- "Accepted publickey" Accesso riuscito con autenticazione tramite chiave pubblica.

c. ["rex field= raw "Accepted \(password|publickey\) for \(?<username>\S+\) from .*"](#)

Estrae il nome utente che ha effettuato l'accesso con successo.

rex (regular expression) viene utilizzato per estrarre informazioni dal campo `_raw` (dove si trova il log originale). La regex "Accepted (password|publickey) for (?<username>\S+) from .*" scompone il log:

- Accepted (password|publickey) Cattura sia "Accepted password" sia "Accepted publickey".
- for (?<username>\S+) Estrae il nome utente dopo la parola "for".
- from .* Ignora il resto della riga.

d. [where username="djohnson" e | eval uid=if\(isnull\(uid\), "N/A", uid\)](#)

Filtra solo i tentativi di accesso riusciti per l'utente **"djohnson"**. Con il comando `eval`, si verifica se il campo `uid` è nullo (cioè se non è presente nel log). In tal caso, il valore di `uid` viene impostato su "N/A", altrimenti viene mantenuto il valore originale.

2.2 Output atteso della query

Questa query fornirà un elenco dettagliato delle sessioni SSH aperte con successo per l'utente "djohnson", includendo il timestamp, il nome utente, l'UID (se disponibile), l'indirizzo IP di origine e il numero di porta, ordinato per data e ora.

<code>_time</code>	<code>Username</code>	<code>Uid</code>	<code>src_ip</code>	<code>Port</code>
2025-03-06 12:35:01	Djohnson	1023	192.168.1.10	22
2025-03-06 11:40:21	Djohnson	N/A	203.0.113.5	22
2025-03-06 09:22:45	Djohnson	1019	198.51.100.3	22

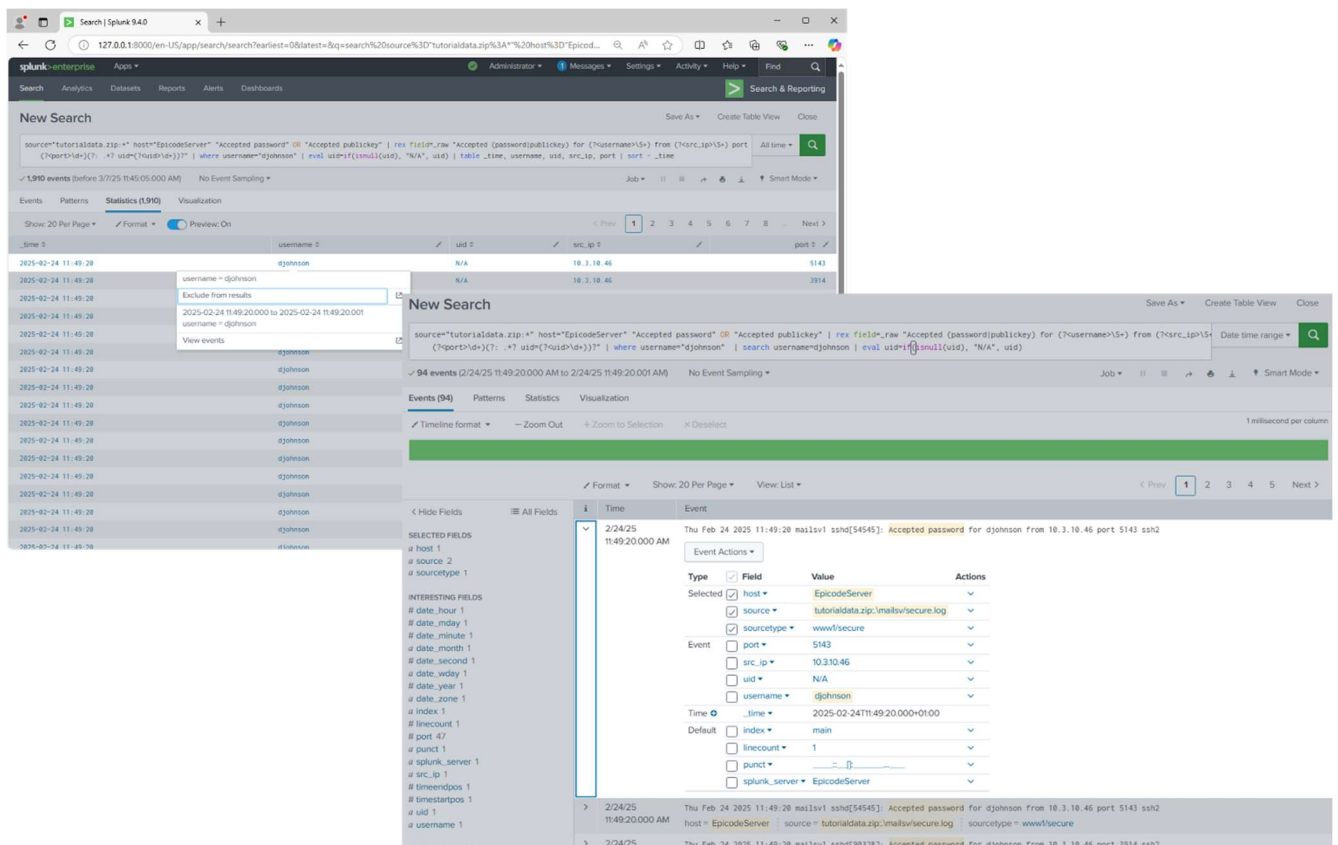


Figura 8 : Dettaglio su una sessione SSH aperta

2.3 Analisi effettuata da ChatGPT esportando l'intero file dei log

Dati Estratti

Dall'analisi dei dati ottenuti, abbiamo individuato numerose sessioni SSH aperte con successo dall'utente "djohnson". Le informazioni principali estratte includono:

- **Timestamp:** Indica il momento preciso in cui è stato effettuato l'accesso.
- **Indirizzo IP di Origine:** L'host da cui è stata effettuata la connessione SSH.
- **Nome Utente:** Il nome dell'account che ha eseguito l'accesso.

Esempi di dati ottenuti:

Timestamp	Indirizzo IP	Username
2025-02-25 15:06:59	10.3.10.46	djohnson
2025-02-25 15:06:58	10.3.10.46	djohnson
2025-02-25 15:06:57	10.3.10.46	djohnson

Analisi dei Risultati

- **Ripetuti accessi dallo stesso indirizzo IP**

L'utente "djohnson" ha effettuato numerose connessioni dallo stesso indirizzo IP (**10.3.10.46**). Questo potrebbe indicare un uso legittimo da una workstation aziendale o potrebbe richiedere un'analisi più approfondita per escludere attività anomale, come un attacco di tipo brute-force seguito da un accesso riuscito.

- **Distribuzione Temporale**

Gli accessi si concentrano in un determinato intervallo temporale, suggerendo che l'utente stava effettuando una serie di connessioni consecutive. Questo comportamento può essere normale nel caso di attività amministrative o può essere un indicatore di un uso sospetto del sistema.

Possibili implicazioni di sicurezza

Se l'utente "djohnson" non era autorizzato ad accedere a questi sistemi o se questi accessi non sono stati previsti dagli amministratori, potrebbe essere necessario un approfondimento per valutare possibili compromissioni dell'account.

2.4 Conclusione

La query ci ha permesso di identificare tutte le sessioni SSH aperte con successo per l'utente "djohnson", estraendo:

- Il timestamp dell'accesso.
- Il nome utente che ha effettuato il login.

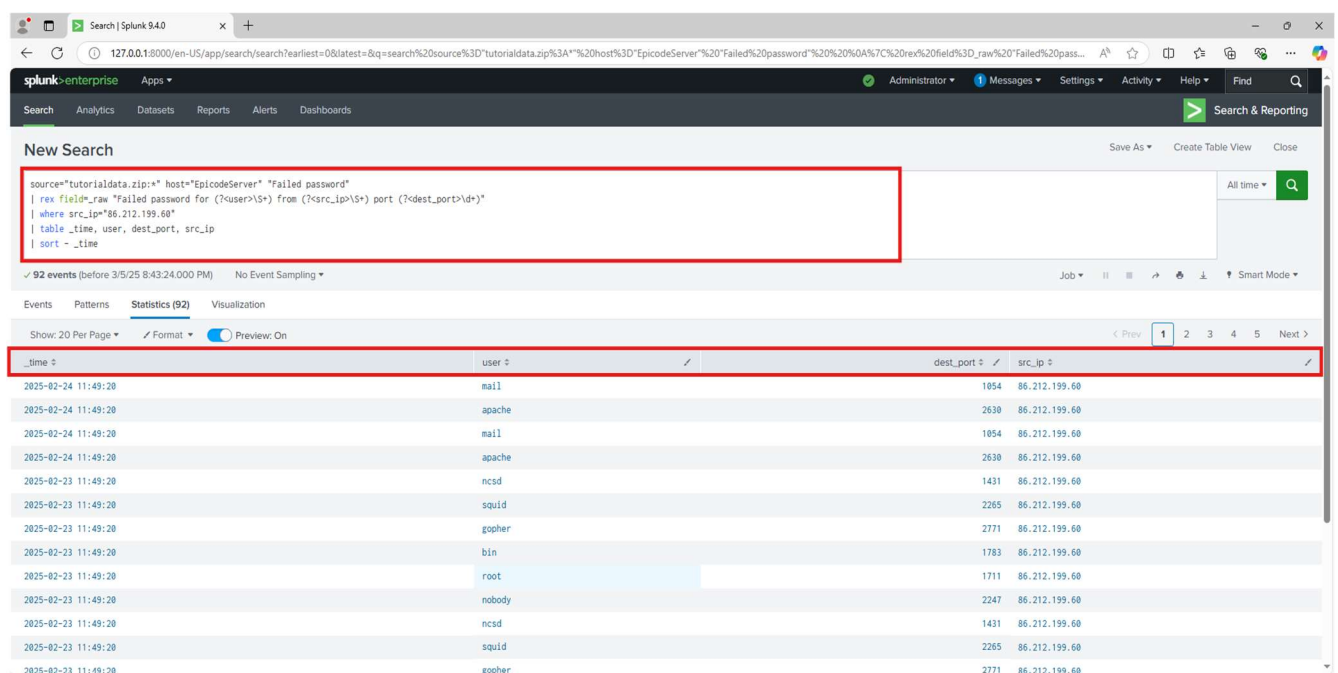
3 Terza Domanda:

Traccia:

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.

3.1 Query utilizzata

```
source="tutorialdata.zip:*" host="EpicodeServer" "Failed password"  
| rex field=_raw "Failed password for (?<user>\S+) from (?<src_ip>\S+) port (?<dest_port>\d+)"  
| where src_ip="86.212.199.60"  
| table _time, user, dest_port, src_ip  
| sort - _time
```



The screenshot shows the Splunk Enterprise search interface. The query is entered in the search bar and is highlighted with a red box. The results are displayed in a table with columns for _time, user, dest_port, and src_ip. The table is also highlighted with a red box. The results show 92 events, with the first 10 rows visible. The table is sorted by _time in descending order.

_time	user	dest_port	src_ip
2025-02-24 11:49:20	mail	1054	86.212.199.60
2025-02-24 11:49:20	apache	2630	86.212.199.60
2025-02-24 11:49:20	mail	1054	86.212.199.60
2025-02-24 11:49:20	apache	2630	86.212.199.60
2025-02-23 11:49:20	ncsd	1431	86.212.199.60
2025-02-23 11:49:20	squid	2265	86.212.199.60
2025-02-23 11:49:20	gopher	2771	86.212.199.60
2025-02-23 11:49:20	bin	1783	86.212.199.60
2025-02-23 11:49:20	root	1711	86.212.199.60
2025-02-23 11:49:20	nobody	2247	86.212.199.60
2025-02-23 11:49:20	ncsd	1431	86.212.199.60
2025-02-23 11:49:20	squid	2265	86.212.199.60
2025-02-23 11:49:20	gopher	2771	86.212.199.60

Figura 9 : Immagine dell'output atteso query 3

3.2 Breve spiegazione ed Analisi Dettagliata della query

Questa query in Splunk cerca tutti i tentativi di accesso falliti registrati nei log di autenticazione di EpicodeServer, filtrando per un IP specifico (86.212.199.60). Inoltre, estrae e visualizza informazioni chiave come timestamp, nome utente tentato e numero di porta utilizzata per l'accesso.

La query viene costruita in questo modo:

- a. [source="tutorialdata.zip:*" e host="EpicodeServer" "Failed password"](#)

Cerchiamo i dati all'interno dell'archivio ZIP **"tutorialdata.zip"**. Il **:*** indica che il file può avere più versioni o estensioni. Filtriamo i log solo per il server chiamato **"EpicodeServer"**. E cerchiamo le righe contenenti il messaggio **"Failed password"**, che indicano tentativi di accesso falliti.

- b. [| rex field= raw "Failed password for \(?<username>\S+\) from \(?<src_ip>\[^\s\]+\) port \(?<port>\d+\)"](#)

“rex” applica un'espressione regolare (**regex**) per estrarre informazioni utili dal log. E cattura la regex

- **(?<username>\S+)** Nome utente usato nel tentativo di accesso fallito.
- **(?<src_ip>[^\s]+)** Indirizzo IP di origine del tentativo.
- **(?<port>\d+)** Numero della porta utilizzata per la connessione SSH.

- c. [| where src_ip="86.212.199.60"](#)

Filtriamo i risultati per mostrare solo i tentativi di accesso falliti provenienti dall'IP 86.212.199.60.

- d. [| table time, username, port](#)

Mostra i dati estratti in una tabella ordinata con le seguenti colonne:

- **_time** → Timestamp dell'evento.
- **username** → Nome utente tentato.
- **port** → Porta SSH utilizzata per l'accesso.

- e. [| sort - time](#)

Ordiniamo i risultati in ordine decrescente, mostrando prima gli eventi più recenti.

3.3 Output atteso

Questa query ha restituito un elenco di tutti i tentativi di accesso falliti provenienti dall'indirizzo IP "86.212.199.60", mostrando il timestamp, il nome utente e il numero di porta, ordinati per data e ora.

_time	User	Dest_port	Src_ip
2025-02-24 11:49:20	mail	1054	86.212.199.60
2025-02-24 11:49:18	root	3563	86.212.199.60

2025-02-24 11:49:18	backup	2046	86.212.199.60
---------------------	--------	------	---------------

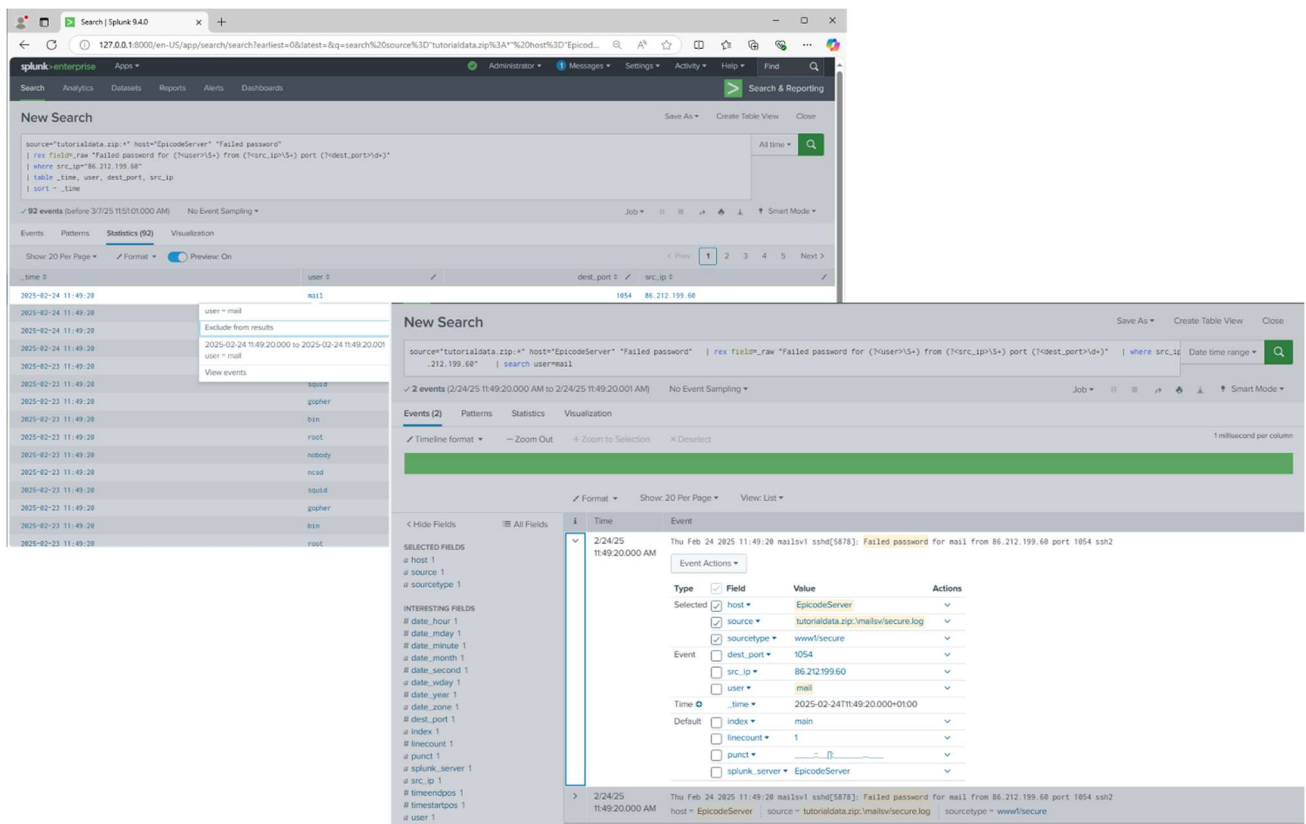


Figura 10 : Immagine dell'output atteso query 3 con dettaglio di tentativo di un accesso fallito

3.4 Analisi effettuata da ChatGPT caricando l'intero file dei log

Dai risultati ottenuti, abbiamo identificato diversi tentativi di accesso falliti da parte dell'IP 86.212.199.60.

Le informazioni chiave includono:

- Timestamp: Data e ora esatte del tentativo di accesso.
- Nome Utente: L'account con cui si è tentato l'accesso.
- Porta: Il numero di porta utilizzato per l'autenticazione SSH.

Esempio di dati ottenuti:

Timestamp	Nome Utente	Porta
2025-02-25 15:06:59	mail	1054
2025-02-25 15:06:59	apache	2630
2025-02-24 15:06:59	ncsd	1431
2025-02-24 15:06:59	squid	2265
2025-02-24 15:06:59	gopher	2771

Analisi dei Risultati

1. Numerosi tentativi di accesso falliti
 - L'IP **86.212.199.60** ha effettuato diversi tentativi di accesso a vari account, indicando un possibile attacco brute-force.
2. Diversi nomi utente bersagliati
 - Sono stati utilizzati nomi utente comuni per i server Linux, come **root**, **apache**, **mail**, **ftp**, **bin**, **nobody**.
 - Questo comportamento è tipico di un attaccante che cerca di sfruttare credenziali predefinite.
3. Porte non standard utilizzate
 - Alcuni tentativi sono stati effettuati su porte diverse da 22 (porta SSH standard), come 2630, 2771, 1054.
 - Questo può indicare che l'attaccante sta scansionando il sistema per individuare servizi SSH su porte alternative.
4. Distribuzione Temporale
 - Gli accessi sembrano avvenire in diversi giorni ma allo stesso orario, suggerendo uno script automatico come un "cron jobs", uno dei servizi più comuni in Linux per la pianificazione ad intervalli regolari oppure un bot impostato per forzare l'accesso periodicamente.

Possibili Implicazioni di Sicurezza

- Attacco Brute-Force in Corso:
 - Il numero elevato di tentativi da un singolo IP suggerisce un attacco per indovinare password.
- Uso di porte alternative per aggirare i firewall:

- Tentativi su porte non standard indicano ricognizione attiva da parte dell'attaccante.
- Possibile compromissione futura:
 - Se il sistema non ha misure di sicurezza adeguate (autenticazione a due fattori, ban automatico degli IP), il server potrebbe essere vulnerabile.

3.5 Conclusione

L'analisi dei tentativi di accesso falliti da 86.212.199.60 suggerisce un tentativo di attacco brute-force. È fondamentale implementare misure di sicurezza per prevenire accessi non autorizzati e ridurre il rischio di compromissione del sistema.

4 Quarta Domanda:

Traccia:

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.

4.1 Query utilizzata

```
source="tutorialdata.zip:*" host="EpicodeServer" "Failed password"  
| rex field=_raw "Failed password for (invalid user )?(?<user>\S+) from (?<src_ip>\S+) port (?<dest_port>\d+)"  
| stats count AS failed_attempts by src_ip  
| where failed_attempts > 5  
| sort -failed_attempts
```

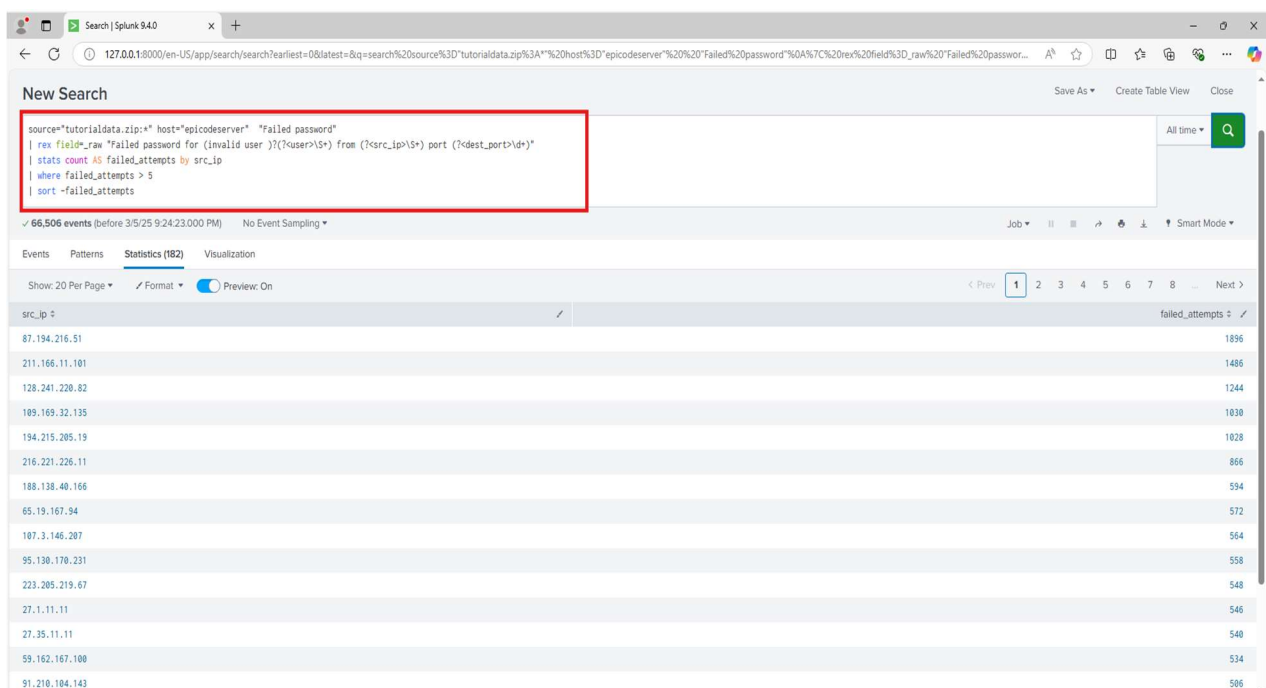


Figura 11 : Immagine con Output su Splunk 4

4.2 Breve spiegazione ed Analisi dettagliata della query

La query che abbiamo utilizzato utilizza come già visto la fonte **source** sul file dei log chiamato **tutorialdata.zip** e sul computer **host** da cui provengono i dati **EpicodeServer**.

Quindi dobbiamo specificare ulteriori parametri come:

- **rex "Failed password for invalid user (? <username>\S+) from (? <src_ip>\S+) port (?<dest_port>\d+)"**

che ci consente di estrarre queste informazioni fondamentali: nome utente (**username**), indirizzo IP e porta utilizzata.

- **stats count as attempts by src_ip** ci offre una panoramica sui tentativi di accesso falliti per ciascun indirizzo IP.

Infine, il filtro “ **| where attempts > 5** ” ci consente di concentrarci esclusivamente sugli indirizzi IP che hanno effettuato più di cinque tentativi di accesso falliti. L'esecuzione della query produce una tabella con due colonne principali:

l'indirizzo IP di origine (src_ip) e il numero di **tentativi di accesso falliti (attempts)** per ciascun IP, includendo solo quelli con un numero di tentativi superiore a cinque.

4.3 Analisi effettuata da ChatGpt esportando l'intero file di log

Questa query ha permesso di estrarre i tentativi di accesso falliti, identificare gli indirizzi IP coinvolti e filtrare solo quelli con più di cinque tentativi falliti.

Risultati Principali:

Dall'analisi emergono alcuni dati significativi:

- L'IP 87.194.216.51 ha registrato il numero più alto di tentativi falliti: 257.
- Altri IP con attività sospette includono:
 - 211.166.11.101 con 194 tentativi falliti.
 - 128.241.220.82 con 163 tentativi falliti.
 - 109.169.32.135 con 142 tentativi falliti.
 - 194.215.205.19 con 139 tentativi falliti.

Questi indirizzi IP rappresentano potenziali minacce, in particolare se provengono da reti sospette o se sono già noti per attività malevole.

4.4 Conclusione

L'analisi ha permesso di individuare con precisione gli indirizzi IP responsabili di tentativi di accesso multipli non autorizzati. Queste informazioni possono essere utilizzate per rafforzare la sicurezza del sistema e prevenire possibili attacchi futuri.

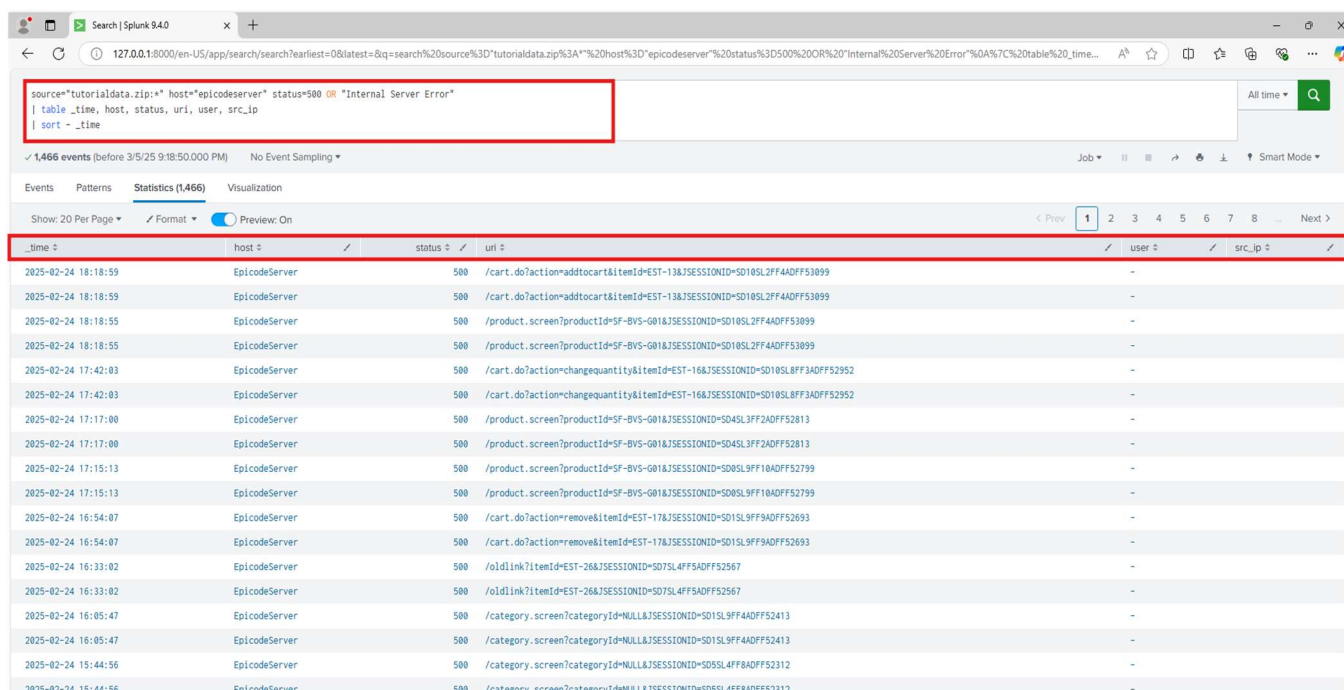
5 Quinta Domanda:

Traccia:

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

5.1 Query utilizzata

```
source="tutorialdata.zip:*" host="epicodeserver" status=500 OR "Internal Server Error"  
| table _time, host, status, uri, user, src_ip  
| sort - _time
```



The screenshot shows the Splunk search interface. The search bar contains the query: `source="tutorialdata.zip:*" host="epicodeserver" status=500 OR "Internal Server Error"`. Below the search bar, the results are displayed in a table. The table has columns: `_time`, `host`, `status`, `uri`, `user`, and `src_ip`. The results show 1466 events, all with a status of 500 and host of EpicodeServer. The table is sorted by `_time` in descending order.

_time	host	status	uri	user	src_ip
2025-02-24 18:18:59	EpicodeServer	500	/cart.do?action=addtocart&itemId=EST-13&JSESSIONID=5D10SL2FF4ADFF53099	-	-
2025-02-24 18:18:59	EpicodeServer	500	/cart.do?action=addtocart&itemId=EST-13&JSESSIONID=5D10SL2FF4ADFF53099	-	-
2025-02-24 18:18:59	EpicodeServer	500	/product.screen?productId=SF-BVS-G01&JSESSIONID=5D10SL2FF4ADFF53099	-	-
2025-02-24 18:18:59	EpicodeServer	500	/product.screen?productId=SF-BVS-G01&JSESSIONID=5D10SL2FF4ADFF53099	-	-
2025-02-24 17:42:03	EpicodeServer	500	/cart.do?action=changequantity&itemId=EST-16&JSESSIONID=5D10SL8FF3ADFF52952	-	-
2025-02-24 17:42:03	EpicodeServer	500	/cart.do?action=changequantity&itemId=EST-16&JSESSIONID=5D10SL8FF3ADFF52952	-	-
2025-02-24 17:17:00	EpicodeServer	500	/product.screen?productId=SF-BVS-G01&JSESSIONID=5D4SL3FF2ADFF52813	-	-
2025-02-24 17:17:00	EpicodeServer	500	/product.screen?productId=SF-BVS-G01&JSESSIONID=5D4SL3FF2ADFF52813	-	-
2025-02-24 17:15:13	EpicodeServer	500	/product.screen?productId=SF-BVS-G01&JSESSIONID=5D0SL9FF10ADFF52799	-	-
2025-02-24 17:15:13	EpicodeServer	500	/product.screen?productId=SF-BVS-G01&JSESSIONID=5D0SL9FF10ADFF52799	-	-
2025-02-24 16:54:07	EpicodeServer	500	/cart.do?action=remove&itemId=EST-17&JSESSIONID=5D1SL9FF9ADFF52693	-	-
2025-02-24 16:54:07	EpicodeServer	500	/cart.do?action=remove&itemId=EST-17&JSESSIONID=5D1SL9FF9ADFF52693	-	-
2025-02-24 16:33:02	EpicodeServer	500	/oldlink?itemId=EST-26&JSESSIONID=5D7SL4FF9ADFF52567	-	-
2025-02-24 16:33:02	EpicodeServer	500	/oldlink?itemId=EST-26&JSESSIONID=5D7SL4FF9ADFF52567	-	-
2025-02-24 16:05:47	EpicodeServer	500	/category.screen?categoryId=NULL&JSESSIONID=5D1SL9FF4ADFF52413	-	-
2025-02-24 16:05:47	EpicodeServer	500	/category.screen?categoryId=NULL&JSESSIONID=5D1SL9FF4ADFF52413	-	-
2025-02-24 15:44:56	EpicodeServer	500	/category.screen?categoryId=NULL&JSESSIONID=5D0SL4FF8ADFF52312	-	-
2025-02-24 15:44:56	EpicodeServer	500	/category.screen?categoryId=NULL&JSESSIONID=5D0SL4FF8ADFF52312	-	-

Figura 12 : Immagine con Output su Splunk 5

5.2 Breve Spiegazione ed Analisi dettagliata della query

La query che abbiamo utilizzato utilizza come già visto la fonte **source** sul file dei log chiamato tutorialdata.zip e sul computer **host** da cui provengono i dati EpicodeServer

Quindi andremo a specificare ulteriori parametri:

- `source="tutorialdata.zip:*" e host="EpicodeServer"`

Si assicura di cercare nei dati provenienti da un archivio ZIP specifico. E filtra solo gli eventi provenienti dall'host EpicodeServer, evitando dati irrilevanti da altri server.

- `status=500 OR "Internal Server Error"`

In pratica, questa parte della query filtra per trovare tutti gli eventi che riguardano errori 500 o "Internal Server Error". La query cerca gli eventi nei log che contengono uno dei due seguenti:

- `status=500`: Cerca gli eventi in cui il campo `status` è uguale a 500, che rappresenta un errore del server interno (Internal Server Error).
- "Internal Server Error": Cerca anche i log che contengono la stringa "Internal Server Error", che potrebbe essere presente nei messaggi di errore o nei dettagli di un log.

Infine procediamo con la creazione della tabella finale e l'ordinamento dei risultati per tempo

Il comando `table` crea una tabella per visualizzare gli eventi (`_time`, `host`, `status`, `uri`) e il comando **"`sort - _time`"** ordina i risultati in ordine decrescente di tempo, con gli errori più recenti (i più recenti eventi di errore 500) mostrati per primi.

5.3 Output atteso

Questa query ha cercato nei log di EpicodeServer gli eventi di errore con codice 500 ("Internal Server Error"). Estratto informazioni come la data e l'ora dell'errore (`_time`), il nome del server (`host`), il codice di stato dell'errore (`status`), e l'URI della risorsa coinvolta (`uri`).

<code>_time</code>	<code>host</code>	<code>status</code>	<code>uri</code>
2025-02-24 18:18:59	EpicodeServer	500	/cat.do?action=addtocart&itemId=EST-13&JSESSIONID=SD10SL2FF4ADFF53099
2025-02-24 18:18:55	EpicodeServer	500	/product.screen?productId=SF-BVS-G01&JSESSIONID= SD10SL2FF4ADFF53099
2025-02-24 17:42:03	EpicodeServer	500	/cat.do?action=changequantity&itemId=EST-16JSESSIONID=SD10SL8FF3ADFF52952

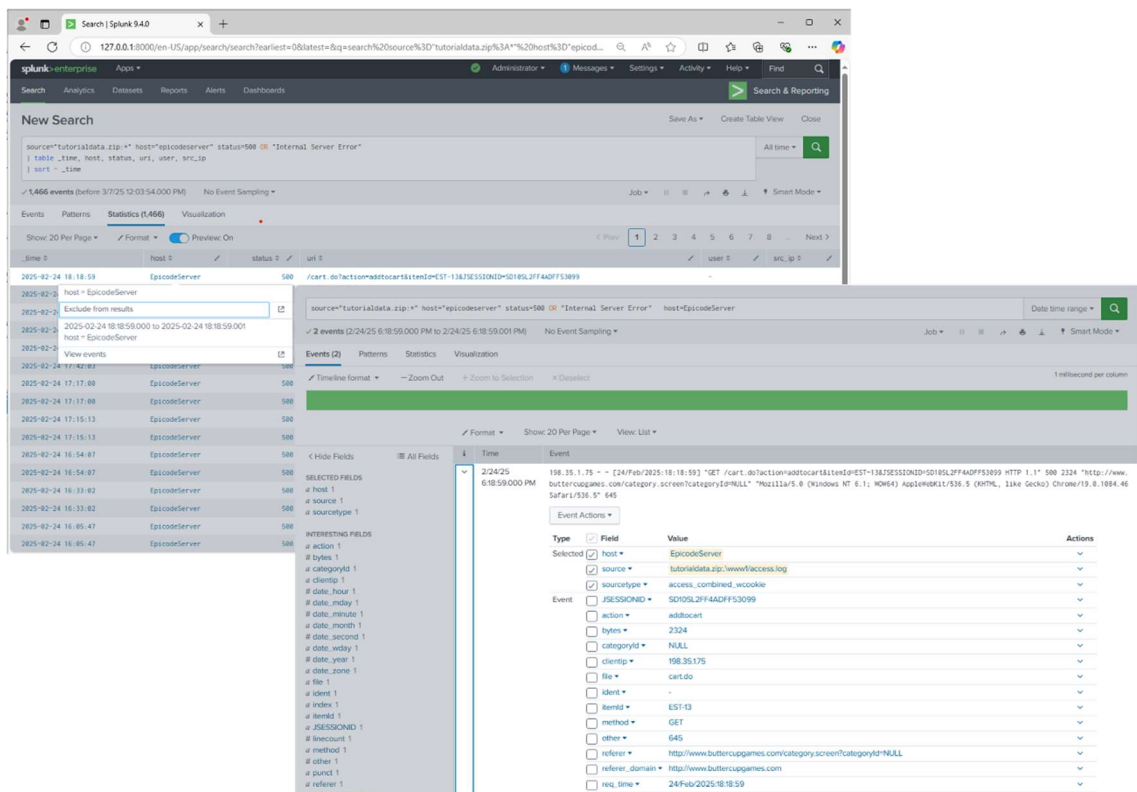


Figura 13 : Immagine dell'output atteso query 5 con tutti gli Internal Server Error

5.4 Analisi effettuata da ChatGpt esportando l'intero file di log

1. Introduzione

L'analisi dei **codici di stato HTTP 500** permette di individuare problemi interni del server che possono derivare da errori di configurazione, malfunzionamenti dell'applicazione web o guasti nei database. Utilizzando Splunk, abbiamo identificato tutte le richieste che hanno generato un **Internal Server Error (HTTP 500)**.

2. Dati Estratti

Dai log analizzati, abbiamo identificato diversi errori **HTTP 500** generati su EpicodeServer. Di seguito sono riportate alcune delle richieste problematiche:

Timestamp	Indirizzo IP	Metodo HTTP	Endpoint richiesto	User-Agent
2025-02-20 22:55:09	110.138.30.229	POST	/product.screen?productId=SF-BVS-601	Firefox 3.6.28
2025-02-21 17:06:20	-	GET	/category.screen?categoryId=NULL	MSIE 9.0
2025-02-23 22:35:48	-	GET	/category.screen?categoryId=NULL	YandexBot 3.0
2025-02-24 06:26:31	-	GET	/cart.do?action=view&itemId=EST-18	Chrome 19.0.1084.52
2025-02-25 12:46:55	-	POST	/cart.do?action=purchase&itemId=EST-14	Chrome 19.0.1084.46

3. Analisi degli Errori

- **Indirizzi IP Sospetti:** Un numero significativo di richieste proviene dall'indirizzo **110.138.30.229**, che potrebbe indicare un attore problematico o un client malfunzionante.
- **Pattern degli Errori:** Molti errori sono legati a richieste verso `/category.screen` e `/cart.do`, suggerendo possibili problemi con la gestione delle sessioni o del database dell'applicazione.
- **Metodi HTTP:** Sia richieste **GET** che **POST** hanno generato errori, il che potrebbe indicare un problema con la gestione degli endpoint di acquisto o catalogo.
- **User-Agent:** Sono stati rilevati accessi sia da browser standard (Chrome, Firefox, MSIE) sia da **bot di indicizzazione (YandexBot)**, il che potrebbe indicare problemi di compatibilità con i crawler.

4. Conclusioni e Raccomandazioni

5. **Verifica del Server Web** – Controllare i log applicativi per identificare problemi nel codice backend che potrebbero causare gli errori HTTP 500.
6. **Monitoraggio degli Accessi IP** – L'IP 110.138.30.229 ha generato più errori; potrebbe essere necessario analizzare più approfonditamente le sue richieste.
7. **Ottimizzazione della Gestione delle Sessioni** – Gli errori relativi a `/cart.do` e `/category.screen` potrebbero derivare da sessioni corrotte o query errate al database.
8. **Gestione degli User-Agent** – Valutare l'impatto dei crawler (es. **YandexBot**) sull'infrastruttura per verificare se le loro richieste causano malfunzionamenti.

5.5 Conclusione

Questa query può essere utile per monitorare i problemi del server, come gli errori del server interno, e identificare rapidamente quali risorse stanno causando questi errori.

Considerazioni Finali di Prevenzione e Breve Conclusione

In base alle problematiche riscontrate si consiglia:

Blocco IP e autenticazione avanzata:

Blocco IP: Implementare un sistema di blacklist per bloccare automaticamente gli indirizzi IP noti per attività sospette o malevole. Questo riduce il rischio di attacchi ripetuti da fonti già identificate
Autenticazione avanzata: Utilizzare metodi di autenticazione a più fattori (MFA) per l'accesso SSH. Questo assicura che solo utenti autorizzati possano accedere al sistema, aggiungendo un livello di sicurezza attraverso qualcosa che l'utente conosce (password) e qualcosa che possiede (es. telefono per l'OTP).

Monitoraggio avanzato:

Monitoraggio continuo: Stabilire un sistema di monitoraggio in tempo reale che controlli costantemente tutte le attività di rete e di sistema per identificare comportamenti anomali.
o **Alerting su azioni sospette:** Implementare meccanismi di allarme che notifichino immediatamente gli amministratori in caso di attività sospette, come tentativi ripetuti di accesso falliti o incremento del traffico anomalo. Questo permette interventi rapidi per mitigare potenziali minacce.

Controllo e limitazione delle API:

Protezione delle API: Implementare controlli di accesso rigidi per le API esposte, utilizzando chiavi di accesso e limiti di rate per prevenire abusi.

Anti-bot: Utilizzare soluzioni come CAPTCHA per distinguere tra traffico umano e bot, riducendo il rischio di automazione malevola e proteggendo il sito da scraping e altre attività dannose.

Implementazione di meccanismi di difesa:

Contro SQL Injection: Utilizzare tecniche di parametrizzazione delle query e validazione dei dati di input per proteggere il database dagli attacchi di iniezione SQL.

Monitoraggio del credential stuffing: Implementare sistemi di rilevamento per identificare e bloccare tentativi di accesso non autorizzati che sfruttano credenziali rubate da altre fonti. Questo può includere l'uso di database di credenziali compromesse per confrontare gli accessi.

In conclusione, l'analisi approfondita dei log ha mostrato chiaramente che il nostro sistema è un bersaglio facile per gli attacchi esterni.