Analisi del malware e Splunk

Progetto - W24D4

Natalia Zanghi’

07/03/2025

Traccia:

Progetto 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.

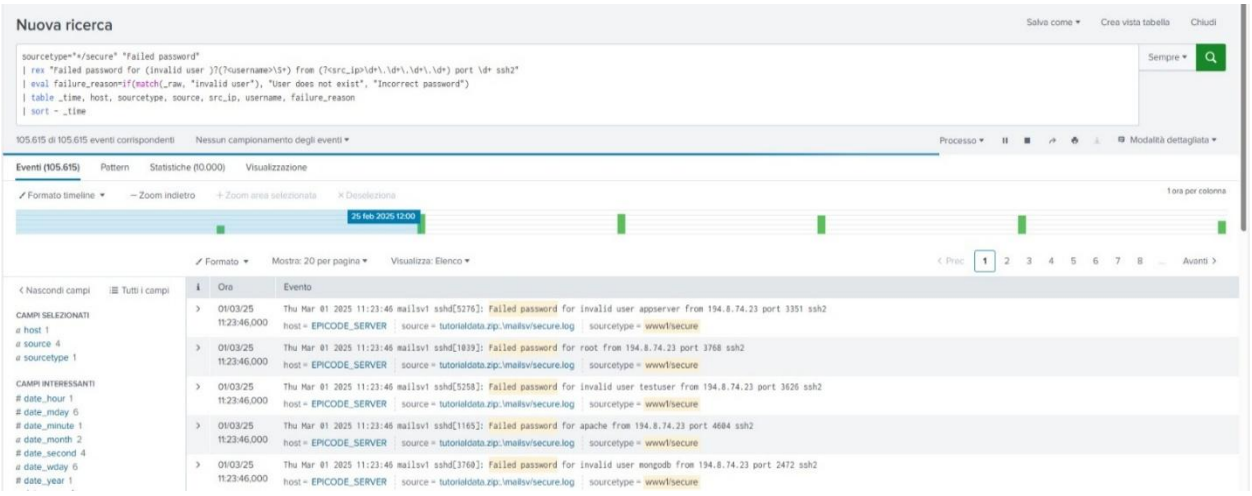
Trarre delle conclusioni sui log analizzati utilizzando AI.

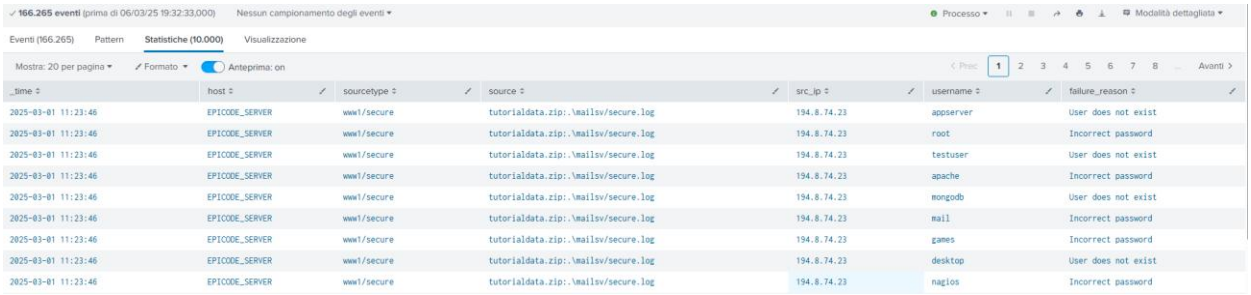
Disclaimer: Dato che il file "tutorialdata.zip" è già stato caricato e i log provengono dal server "EPICODE_SERVER", queste condizioni sono considerate implicitamente incluse. Questo ci permette di focalizzarci direttamente sugli obiettivi delle query, come tentativi di accesso falliti o connessioni autorizzate, rendendole più concise e chiare.

1. Questa query identifica i tentativi di accesso falliti nei log del server, isolando dettagli fondamentali come il timestamp, l'indirizzo IP di origine , il nome utente e la causa del fallimento . Utilizza la stringa "Failed password" per filtrare gli eventi e una regex per estrarre i dati chiave dai messaggi di log. Successivamente, crea una tabella con i campi essenziali (_time, host, source, src_ip, username, failure_reason) e ordina gli eventi in ordine cronologico decrescente con sort - _time, rendendo i risultati chiari, ordinati e pronti per l'analisi.

Query:

```
sourcetype="*/secure" "Failed password"
| rex "Failed password for (invalid user )?(?<username>\S+) from
(?<src_ip>\d+\.\d+\.\d+\.\d+) port \d+ ssh2"
| eval failure_reason=if(match(_raw, "invalid user"), "User does not exist", "Incorrect
password")
| table _time, host, sourcetype, source, src_ip, username, failure_reason
| sort - _time
```





✓ **166.265 eventi** (prima di 06/03/25 19:32:33,000) Nessun campionamento degli eventi ▾ Processo ▾ Modalità dettagliata ▾

Eventi (166.265) Pattern **Statistiche (10.000)** Visualizzazione

Mostra: 20 per pagina ▾ Formato ▾ Anteprima: on < Prec 1 2 3 4 5 6 7 8 … Avanti >

_time ≎	host ≎	sourcetype ≎	source ≎	src_ip ≎	username ≎	failure_reason ≎
2025-03-01 11:23:46	EPICODE_SERVER	www1/secure	tutorialdata.zip:.\mailsv/secure.log	194.8.74.23	appserver	User does not exist
2025-03-01 11:23:46	EPICODE_SERVER	www1/secure	tutorialdata.zip:.\mailsv/secure.log	194.8.74.23	root	Incorrect password
2025-03-01 11:23:46	EPICODE_SERVER	www1/secure	tutorialdata.zip:.\mailsv/secure.log	194.8.74.23	testuser	User does not exist
2025-03-01 11:23:46	EPICODE_SERVER	www1/secure	tutorialdata.zip:.\mailsv/secure.log	194.8.74.23	apache	Incorrect password
2025-03-01 11:23:46	EPICODE_SERVER	www1/secure	tutorialdata.zip:.\mailsv/secure.log	194.8.74.23	mongodb	User does not exist
2025-03-01 11:23:46	EPICODE_SERVER	www1/secure	tutorialdata.zip:.\mailsv/secure.log	194.8.74.23	mail	Incorrect password
2025-03-01 11:23:46	EPICODE_SERVER	www1/secure	tutorialdata.zip:.\mailsv/secure.log	194.8.74.23	games	Incorrect password
2025-03-01 11:23:46	EPICODE_SERVER	www1/secure	tutorialdata.zip:.\mailsv/secure.log	194.8.74.23	desktop	User does not exist
2025-03-01 11:23:46	EPICODE_SERVER	www1/secure	tutorialdata.zip:.\mailsv/secure.log	194.8.74.23	nagios	Incorrect password

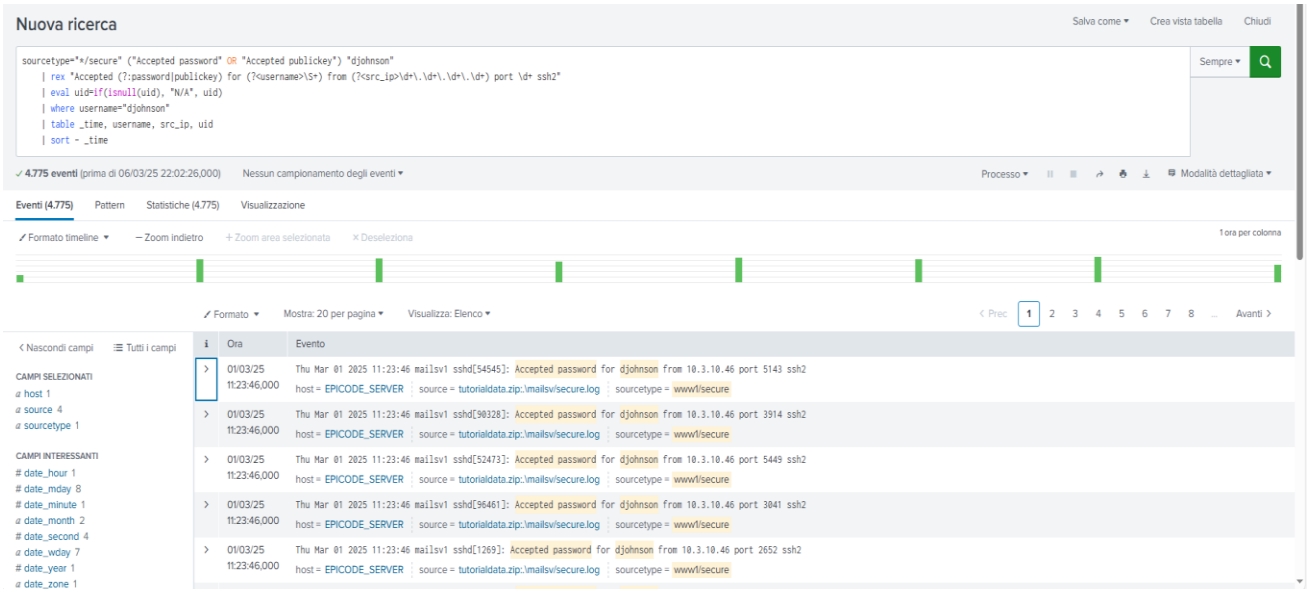
Il risultato di questa query documenta ogni tentativo di accesso fallito, indicando il momento preciso dell'evento, il server o dispositivo coinvolto, l'indirizzo IP della macchina di origine, il nome utente utilizzato (anche se non valido) e il motivo del fallimento. Se il log contiene la stringa "invalid user", il motivo sarà "User does not exist"; altrimenti, sarà "Incorrect password".

Queste informazioni permettono di ricostruire la sequenza degli eventi e individuare da dove provengono i tentativi e chi ha provato ad accedere. Ogni riga rappresenta un singolo tentativo, e analizzando i dati si possono identificare schemi ripetitivi o attività sospette, segnali di possibili attacchi in corso. La query si rivela così uno strumento essenziale per monitorare e mitigare rischi di sicurezza sul server.

2. Questa query identifica tutte le sessioni SSH aperte con successo dall'utente djohnson, includendo il timestamp e l'ID utente. Poiché SSH è un protocollo sicuro per l'accesso remoto, la query analizza messaggi specifici di autenticazione, come "Accepted password" e "Accepted publickey". Inoltre, con la funzione eval, verifica se il campo uid è nullo; in tal caso, assegna il valore N/A (non disponibile). Questo garantisce una visualizzazione chiara e completa anche quando l'UID non è presente nei log.

Query:

```
sourcetype="*/secure" ("Accepted password" OR "Accepted publickey") "djohnson"
| rex "Accepted (?:password|publickey) for (?<username>\S+) from
(?<src_ip>\d+\.\d+\.\d+\.\d+) port \d+ ssh2"
| eval uid=if(isnull(uid), "N/A", uid)
| where username="djohnson"
| table _time, username, src_ip, uid
| sort - _time
```



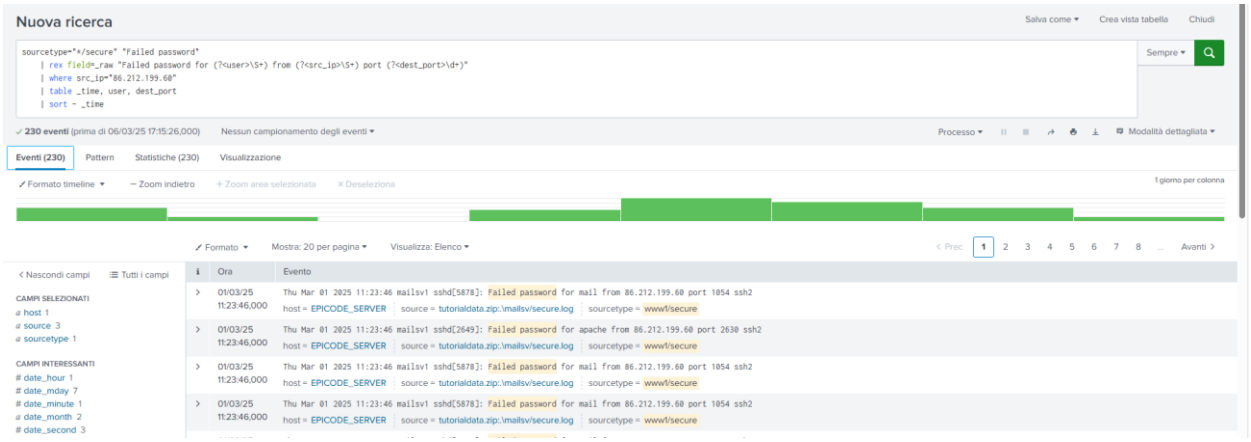
_time	username	src_ip	uid
2025-03-01 11:23:46	djohnson	10.3.10.46	N/A
2025-03-01 11:23:46	djohnson	10.3.10.46	N/A
2025-03-01 11:23:46	djohnson	10.3.10.46	N/A
2025-03-01 11:23:46	djohnson	10.3.10.46	N/A
2025-03-01 11:23:46	djohnson	10.3.10.46	N/A
2025-03-01 11:23:46	djohnson	10.3.10.46	N/A
2025-03-01 11:23:46	djohnson	10.3.10.46	N/A
2025-03-01 11:23:46	djohnson	10.3.10.46	N/A

Notiamo che l'IP è costante, mentre le porte variano, il che è normale nei protocolli SSH dove ad ogni sessione viene assegnata una porta temporanea per identificare in modo univoco le connessioni. Tuttavia, l'elevato numero di connessioni in un breve lasso di tempo potrebbe indicare l'utilizzo di uno script automatico o un'attività sospetta.

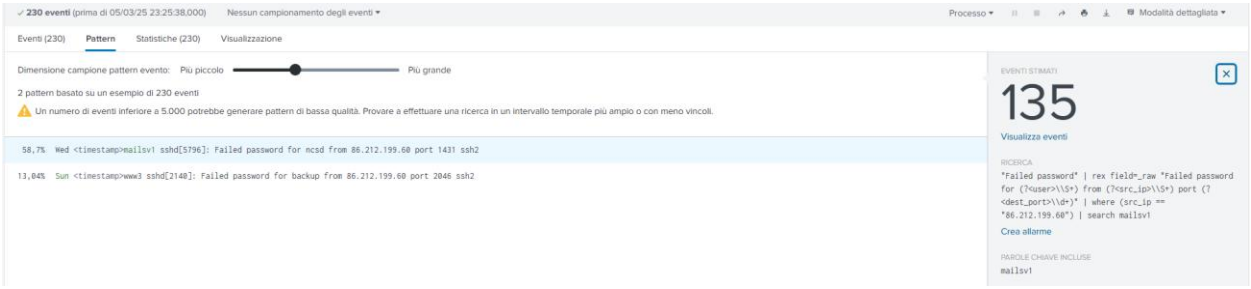
3. Questa query cerca nei log tutti i tentativi di accesso falliti dall'IP 86.212.199.60, estrae il nome utente, l'indirizzo IP e la porta usata, e organizza i risultati in ordine cronologico. Questo tipo di analisi è utile per capire come il sistema sta operando e per identificare eventuali anomalie o comportamenti insoliti

Query:

```
sourcetype="*/secure" "Failed password"
| rex field=_raw "Failed password for (?\S+) from (?\S+) port (?\d+)"
| where src_ip="86.212.199.60"
| table _time, user, dest_port
| sort - _time
```



Questi log mostrano più tentativi di accesso falliti al server dall'indirizzo IP 86.212.199.60 nello stesso momento temporale: 01/03/25 11:23:46. Notiamo che stanno provando a connettersi da "apache" e "mail" che sono dei server comuni che possono fungere da bersaglio e che i tentativi sono effettuati da porte diverse la 1054 e 2630 ma utilizzando i protocollo sicuro SSH2 che è specifico per accessi remoti sicuri. Questo comportamento suggeriscono un possibile attacco brute-force o una scansione mirata per individuare vulnerabilità.



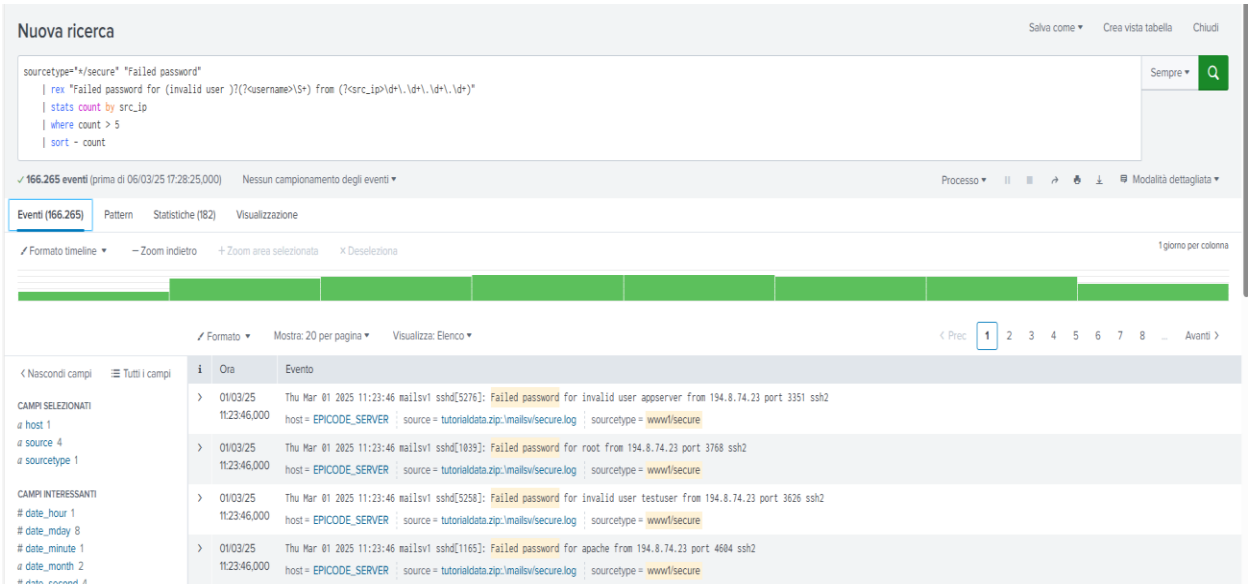
I log mostrano che l'IP 86.212.199.60 ha effettuato tentativi di accesso falliti usando l'utente "ncsd" sulla porta 1431 (135 eventi) e "backup" sulla porta 2046 (80 eventi) fino al massimo di 230 eventi log in un giorno. Questo comportamento è sospetto e potrebbe indicare un attacco brute-force. Si consiglia di bloccare l'IP e rafforzare la configurazione del server SSH.

Continuando ad analizzare il risultato vediamo che tutti gli eventi hanno lo stesso timestamp, ma variano per nomi utente e porte di destinazione. La presenza dello stesso orario con combinazioni diverse di utenti e porte potrebbe indicare tentativi simultanei o automatizzati di accesso fallito. Questo rende l'attività particolarmente sospetta e merita ulteriori indagini.

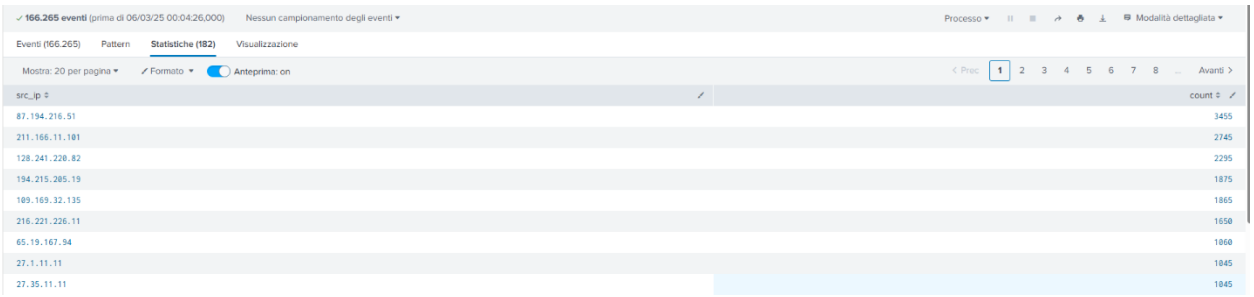
4. Questa query identifica gli indirizzi IP con più di 5 tentativi di autenticazione falliti. Filtra i log con "Failed password", estrae username e src_ip tramite regex. Successivamente aggrega i tentativi con stats count by src_ip, applica il filtro where count > 5 e ordina i risultati con sort – count.

Query:

sourcetype="*/secure" "Failed password"
| rex "Failed password for (invalid user)?(?<username>\S+) from
(?<src_ip>\d+\.\d+\.\d+\.\d+)"
| stats count by src_ip
| where count > 5
| sort - count



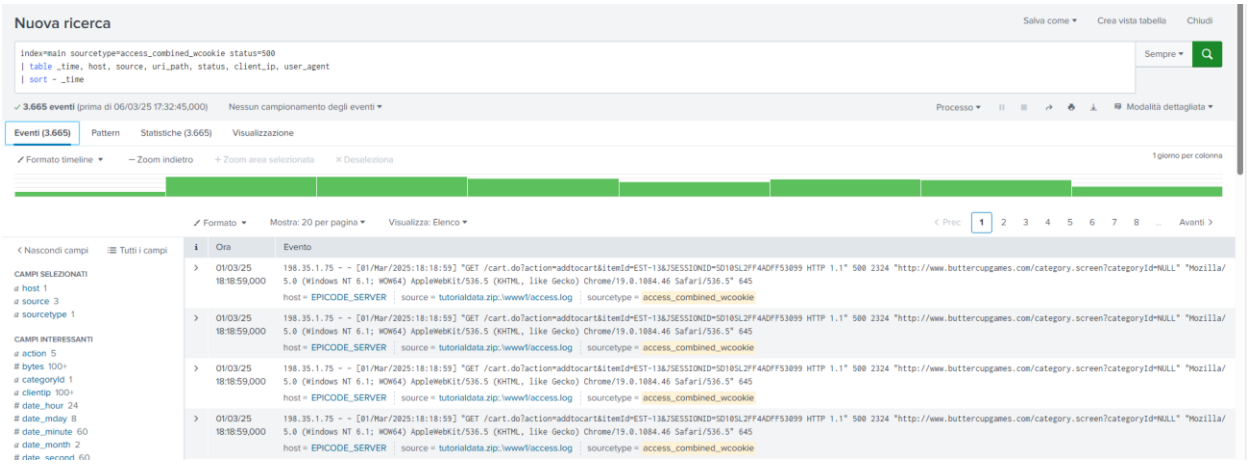
Possiamo notare che gli eventi evidenziano una serie di tentativi di accesso falliti al sistema, caratterizzati dall'utilizzo di nomi utente diversi e l'uso di porte variabili proveniente da sempre lo stesso 86.212.199.60.
Possono essere tentativi automatizzati di un malintenzionato poiché gli attacchi manuali avrebbero intervalli di tempo irregolari.



Osserviamo che alcuni indirizzi IP registrano un numero elevato di tentativi di accesso falliti, come l'IP 87.194.216.51 con 3.455 tentativi e l'IP 211.166.11.101 con 2.745 tentativi. Questa alta frequenza di eventi suggerisce un'attività sospetta. La query risulta particolarmente utile per individuare comportamenti anomali o malevoli, come attacchi di tipo brute-force, permettendo di individuare rapidamente le sorgenti di tali azioni.

5.La query identifica gli errori interni del server (HTTP 500) cercando nei log eventi con status=500, limitandosi all'indice specificato (index=main) e al tipo di sorgente definito access_combined_wcookie

index=main sourcetype=access_combined_wcookie status=500
| table _time, host, source, uri_path, status, client_ip, user_agent
| sort - _time



La query ha identificato 3.665 eventi in totale con il codice HTTP 500, un numero significativo, che potrebbe indicare un problema ricorrente o sistemico.
Possiamo vedere che l'IP 198.35.1.75 ha fatto una richiesta al server per aggiungere un articolo al carrello usando un percorso URI specifico con parametri. Tuttavia, il server ha generato un errore 500 perché potrebbe esserci un problema tecnico con l'endpoint oppure l'IP potrebbe aver inviato input malevoli o manipolati (itemId, JSESSIONID)
L'errore è sospetto ma non ne indica la colpevolezza ma sarebbe un segnale da approfondire .Può essere dall'incapacità del server di processare correttamente la richiesta ma tuttavia c'è bisogno di approfondire per distinguere tra un problema tecnico e un'attività malevola.

Conclusione:

Dall'analisi delle query e dei risultati emersi, si evince quanto sia importante un monitoraggio accurato e strutturato dei log di sistema per garantire la sicurezza e l'efficienza operativa. Ecco alcune deduzioni principali:

1. **Importanza dell'automazione e dei filtri avanzati**: Utilizzare query ben costruite con filtri, regex e aggregazioni consente di individuare rapidamente potenziali minacce o anomalie, migliorando la capacità di risposta alle emergenze.
2. **Identificazione e mitigazione delle minacce**: Indirizzi IP con tentativi di accesso ripetuti o errori sistematici devono essere immediatamente investigati. Blocchi automatici, analisi temporale e strumenti di autenticazione avanzata, come l'MFA, possono mitigare i rischi di attacchi brute-force e attività malevole.
3. **Analisi degli errori per migliorare la stabilità**: Gli errori HTTP (500) evidenziano vulnerabilità tecniche o input sospetti. È fondamentale distinguere tra problemi sistemici e potenziali exploit da parte di utenti malintenzionati, adottando misure correttive proattive.
4. **Valore di un approccio proattivo**: L'uso di strumenti come Splunk per monitorare accessi, errori e connessioni è essenziale per rilevare schemi sospetti e anticipare possibili attacchi, garantendo la continuità operativa e la sicurezza.

In sintesi, il progetto ha dimostrato quanto un'analisi mirata e iterativa dei log sia fondamentale per la cybersecurity e il miglioramento dei sistemi informatici. Integrare queste metodologie con l'uso di AI e sistemi di alert in tempo reale potrebbe ulteriormente rafforzare le difese e ottimizzare le prestazioni.