Splunk
Epicode - Progetto Finale
Analisi dei Log

Giuseppe Gigliotti - gius199874@gmail.com

4 gennaio 2026

Indice

Riassunto Esecutivo:

Dopo l'installazione di splunk, importare su esso i dati di esempio "**tutorialdata.zip**". Fatto ciò vediamo le richieste sulla traccia:

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.

Scopo del test:

1. Tentativi di accesso falliti:

```
source="tutorialdata.zip:*" | search "Failed password"
| rex field=_raw "from (?<ip>\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3})"
| rex field=_raw "(?<user>[\w@-]+) from" |rex field=_raw "(?<motivo>invalid user)"
| eval fallimento = if(isnull(motivo), "Failed password for valid user", "Failed password
↪  for invalid user")
| sort - _time
| table _time ip user fallimento
```

Spiegazione:

Questa query Splunk ha l'obiettivo di analizzare i log di autenticazione fallita per distinguere se il tentativo di accesso è avvenuto con un nome utente esistente (valido) o non esistente (non valido), estraendo contestualmente l'IP e il nome utente. Inoltre, i risultati vengono ordinati dal più recente al più vecchio.

Costruzione della query: 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.

b. `search "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.

c. `rex field=_raw "from (?<ip>\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3})"`

"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 cerca la parola "from" e cattura l'indirizzo IP che segue, salvandolo nel campo

d. `rex field=_raw "(?<user>[\w@-]+) from"`

Questo pezzo di query cattura il nome utente (caratteri alfanumerici, @, trattini) che precede la parola "from", salvandolo nel campo user

e. `rex field=_raw "(?<motivo>invalid user)"`

Questo pezzo di query dice che se nel log è presente la stringa "invalid user", la salva nel campo motivo. Se non c'è, il campo rimane vuoto (null).

f. `eval fallimento = if(isnull(motivo), "Failed password for valid user", "Failed password for invalid user")`

“eval” crea un campo fallimento per rendere più chiaro il motivo dell’errore.

- Se il campo “motivo” è vuoto (isnull), significa che il log non diceva “invalid user”, quindi deduce che l’utente esiste ma la password era errata (“Failed password for valid user”).
- Se il campo “motivo” contiene dati, assegna il valore “Failed password for invalid user”.

e. `sort - _time`

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

f. `table _time ip user fallimento`

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

- _time → Il timestamp dell’evento.
- ip → L’indirizzo IP di origine dell’utente che ha tentato l’accesso.
- user → Il nome utente utilizzato nel tentativo fallito.
- fallimento → Il motivo dell’errore.

Qui si possono vedere i risultati delle query:

- Risultato della query

2. Sessioni SSH aperte con successo per l'utente:

```
source="tutorialdata.zip:*"
| search sshd:session
| search "session opened for user djohnson"
| table _time uid
```

Spiegazione

Questa query è progettata per filtrare eventi relativi a sessioni SSH (sshd:session). Cerca specificamente le sessioni aperte per djohnson, visualizza timestamp e UID. Il campo uid viene estratto automaticamente da Splunk dai log

Costruzione della query: 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.

b. `search sshd:session`

Restringe i risultati solo agli eventi che contengono il termine "sshd:session", ovvero i log relativi alla gestione delle sessioni SSH di Linux.

c. `search "session opened for user djohnson"`

Applica un ulteriore filtro per isolare solo i messaggi che confermano l'avvenuta apertura di una sessione per l'utente specifico djohnson.

d. `table _time uid`

Formatta i risultati in una tabella pulita, mostrando solo due colonne:

- _time → Il timestamp dell'evento.
- uid → L'ID dell'utente

Qui si possono vedere i risultati delle query:

- Risultato della query

3. Tentativi di accesso falliti provenienti dall'indirizzo IP:

```
source="tutorialdata.zip:*" | search "failed password" "86.212.199.60"
| rex field=_raw "(?<user>[\w@-]+)\s+from\s+(.*?)"
| rex field=_raw "port (?<port>\d{1,5})"
| sort - _time
| table _time user port
```

Spiegazione:

Questa query in Splunk cerca tutti i tentativi di accesso falliti registrati nei log di autenticazione del server, 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.

Costruzione della query:

a. `source="*tutorialdata.zip*"`

Cerchiamo i dati all'interno dell'archivio ZIP "tutorialdata.zip". Il :* indica che il file può avere più versioni o estensioni.

b. `search "failed password" "86.212.199.60"`

Questo è un filtro molto specifico. Recupera solo gli eventi che contengono contemporaneamente la frase "failed password"e l'indirizzo IP "86.212.199.60". In pratica, si sta isolando l'attività di un sospetto attaccante da quell'IP.

c. `rex field=_raw "(?<user>[\w@-]+)\s+from\s+(.*?)"`

- Cerca una parola (composta da lettere, numeri, @ o -) che si trova prima della parola "from".
- Salva questa parola nel campo user

d. `rex field=_raw "port (?<port>\d{1,5})"`

- Cerca la parola "port" seguita da una sequenza di numeri (da 1 a 5 cifre).
- Salva questo numero nel campo port.

Questo serve per identificare quale porta di rete (es. 22 per SSH) l'attaccante stava cercando di colpire.

e. `sort - _time`

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

f. `table _time user port`

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

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

Qui si possono vedere i risultati delle query:

- Risultato della query

4. Tentativo di accedere al sistema più di 5 volte:

```
source="tutorialdata.zip:*" | search  "Failed password"
| rex field=_raw "from (?<ip>\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3})"
| stats count as failed_attempts by ip
| where failed_attempts > 5
| sort - failed_attempts
```

Spiegazione:

Questa query ha l'obiettivo di identificare i potenziali attacchi brute-force, filtrando gli indirizzi IP che hanno accumulato più di 5 tentativi di accesso falliti e ordinandoli dal più pericoloso al meno pericoloso.

Costruzione della 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.

b. `search  "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.

c. `rex field=_raw "from (?<ip>\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3})"`

"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 cerca la parola "from" e cattura l'indirizzo IP che segue, salvandolo nel campo

d. `stats count as failed_attempts by ip`

Conta quante volte compare ogni IP e salva il totale in un campo chiamato failed_attempts.

e. `where failed_attempts > 5`

Applica un controllo. Mostra solo gli IP che hanno fallito il login più di 5 volte.

f. `sort - failed_attempts`

Ordina i risultati in modo decrescente (i “peggiori” in alto)

Qui si possono vedere i risultati delle query:

- Risultato della query

N.B. In questa query non serve il comando “table” in quanto, usare table alla fine dopo che i campi sono già stati aggregati da stats è ridondante. “stats” crea già una tabella con i campi definiti (ip e failed_attempts).

5. Internal Server Error (Errore 500):

```
source="tutorialdata.zip:*"
| search status=500
```

Spiegazione:

Questa query è molto semplice ma fondamentale per il monitoraggio della stabilità di un sistema. Serve a isolare tutti i gravi errori lato server registrati nei tuoi log (codice 500 Internal Server Error). Splunk ti restituirà l’intero contenuto dei log (_raw) per ogni richiesta fallita.

Costruzione:

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.

b. `search status=500`

Questo comando filtra i risultati per mostrare solo gli eventi che hanno il campo status esattamente uguale a 500.

- A differenza degli errori 404 (pagina non trovata),i codici 500 indicano quasi sempre un bug nel codice, un crash di un’applicazione o un problema di database che deve essere risolto immediatamente dal team tecnico.

Qui si possono vedere i risultati delle query:

- Risultato della query

N.B. Se si vuole scoprire quali sono le pagine o i servizi che stanno generando più errori 500, si può aggiungere questo comando alla fine della query:

```
| stats count by uri_path
```

Questo ti mostrerà una tabella con l'elenco degli indirizzi (URL) del tuo sito e quante volte ciascuno ha generato un errore interno.

Conclusioni:

Tutte e 5 le query richieste dal progetto sono state implementate con successo e testate sui dati reali di tutorialdata.zip.

Le query forniscono insight importanti su:

- Tentativi di accesso non autorizzati
- Attività degli utenti legittimi
- Pattern di attacco
- Problemi di stabilità applicativa

Il progetto dimostra competenza nell'utilizzo di Splunk per:

- Ricerca e filtraggio di log
- Estrazione di campi tramite regex
- Aggregazione e analisi statistica
- Identificazione di minacce alla sicurezza

Documentazione:

- Manuale di Riferimento SPL (Search Processing Language)
- Ricerche di base e risultati di ricerca