

09.08.2025



ANALISI MALWARE E SPLUNK

PROGETTO FINE MODULO M6

FABIO MUNGIOVÌ

ANALISI MALWARE E SPLUNK

PROGETTO FINE MODULO M6

INDICE

INTRODUZIONE.....	2
QUERY N. 1	3
QUERY N. 2.....	9
QUERY N. 3.....	11
QUERY N. 4	13
QUERY N. 5.....	14
CONCLUSIONI	15
INCIDENT RESPONSE	16

INTRODUZIONE

Il seguente report si concentra sull'analisi di dati di log utilizzando **Splunk**, una piattaforma per la ricerca, il monitoraggio e l'analisi dei dati generati da macchine.

I log sono una fonte cruciale di informazioni per la sicurezza informatica e l'amministrazione dei sistemi, ma la loro grande quantità e la loro natura non strutturata rendono necessaria l'automazione.

In questo contesto, l'esercitazione mira a mettere in pratica le funzionalità di Splunk per esaminare un set di dati di esempio, "tutorialdata.zip".

L'obiettivo è sviluppare e implementare una serie di **query** per identificare eventi specifici, come i tentativi di accesso falliti, le sessioni SSH aperte e gli errori del server.

Questo processo permette di estrarre informazioni cruciali dai dati grezzi, trasformandole in conoscenza utile per la gestione della sicurezza e l'osservabilità del sistema.

Nello specifico le query richieste sono:

1. Identificare tutti i tentativi di accesso falliti (Failed password).
Mostrare il timestamp, l'indirizzo IP di origine, il nome utente e il motivo del fallimento.
2. Trovare tutte le sessioni SSH aperte con successo.
Filtrare i risultati per l'utente *djohnson*, mostrare il timestamp e l'ID utente.
3. Trovare tutti i tentativi di accesso falliti provenienti dall'indirizzo IP "86.212.199.60".
Mostrare il timestamp, il nome utente e il numero di porta.
4. Identificare gli indirizzi IP che hanno tentato di accedere ("Failed password") al sistema più di 5 volte.
Mostrare l'indirizzo IP e il numero di tentativi.
5. Trovare tutti gli Internal Server Error.

In conclusione, analizzeremo i log degli eventi studiati per trarre delle conclusioni sugli eventi accaduti sul sistema in analisi.

QUERY N. 1

IDENTIFICARE TUTTI I TENTATIVI DI ACCESSO FALLITI (FAILED PASSWORD).
MOSTRARE IL TIMESTAMP, L'INDIRIZZO IP DI ORIGINE, IL NOME UTENTE E IL MOTIVO DEL FALLIMENTO.

Per la creazione di questa query prendiamo innanzitutto in analisi la composizione dei log dei tentativi di accesso falliti, con la seguente query:

```
source="tutorialdata.zip:*" "Failed password"
```

Nuova ricerca

```
source="tutorialdata.zip:*" "Failed password"
```

✓ **166.265 eventi** (prima di 09/08/25 12:09:20,000)

Salta all'occhio l'elevato numero di accessi falliti, ma ci concentreremo sull'analisi più avanti.

Qui una panoramica di alcuni dei risultati ottenuti, con a sinistra i campi di selezione disponibili.

< Nascondi campi	Tutti i campi	i	Ora	Evento
CAMPI SELEZIONATI a host 1 a source 4 a sourcetype 1				
CAMPI INTERESSANTI # date_hour 1 # date_mday 8 # date_minute 1 a date_month 2 # date_second 4 a date_wday 7 # date_year 1 a date_zone 1 # index 1 # linecount 1 a punct 3 a splunk_server 1 # timeendpos 1 # timestartpos 1				
		>	04/08/25 13:06:40,000	Thu Aug 04 2025 13:06:40 mailsv1 sshd[5276]: Failed password for invalid user appserver from 194.8.74.23 port 3351 ssh2 host = Lenovo source = tutorialdata.zip:\mailsv\secure.log sourcetype = www1/secure
		>	04/08/25 13:06:40,000	Thu Aug 04 2025 13:06:40 mailsv1 sshd[1039]: Failed password for root from 194.8.74.23 port 3768 ssh2 host = Lenovo source = tutorialdata.zip:\mailsv\secure.log sourcetype = www1/secure
		>	04/08/25 13:06:40,000	Thu Aug 04 2025 13:06:40 mailsv1 sshd[5258]: Failed password for invalid user testuser from 194.8.74.23 port 3626 ssh2 host = Lenovo source = tutorialdata.zip:\mailsv\secure.log sourcetype = www1/secure
		>	04/08/25 13:06:40,000	Thu Aug 04 2025 13:06:40 mailsv1 sshd[1165]: Failed password for apache from 194.8.74.23 port 4604 ssh2 host = Lenovo source = tutorialdata.zip:\mailsv\secure.log sourcetype = www1/secure
		>	04/08/25 13:06:40,000	Thu Aug 04 2025 13:06:40 mailsv1 sshd[3760]: Failed password for invalid user mongodb from 194.8.74.23 port 2472 ssh2 host = Lenovo source = tutorialdata.zip:\mailsv\secure.log sourcetype = www1/secure
		>	04/08/25 13:06:40,000	Thu Aug 04 2025 13:06:40 mailsv1 sshd[4998]: Failed password for mail from 194.8.74.23 port 1552 ssh2 host = Lenovo source = tutorialdata.zip:\mailsv\secure.log sourcetype = www1/secure
				Thu Aug 04 2025 13:06:40 mailsv1 sshd[1930]: Failed password for games from 194.8.74.23 port 3007 ssh2

Analizzando i campi disponibili per ogni singolo log, i risultanti mostrano un parsing poco efficace, in quanto i dati che sono di nostro interesse non sono contenuti nei campi selezionabili, ma bensì sono nel `_raw` (la parte di testo grezza) del log.

</

SEARCH REFERENCE REX

L'obiettivo quindi, per ottenere risultati efficaci, sarà quello di estrarre i dati dal `_raw` del log, con l'utilizzo del comando di ricerca `rex`.

Rex è un comando specifico nel linguaggio di ricerca di Splunk.

Questo comando usa la sintassi delle regex per estrarre campi da un log grezzo (`_raw`), e trasformare dati non strutturati in dati strutturati, creando nuovi campi per l'analisi.

Analizziamo quindi il `_raw` dei log per identificare i campi da estrarre:

1 2 3 4
Thu Aug 04 2025 13:06:40 mailsrv1 sshd[5276] Failed password for invalid user appserver from 194.8.74.23 port 3351 ssh2


1. Timestamp dell'evento
2. Motivo del fallimento di accesso
3. Username
4. IP di origine

CREAZIONE NUOVI CAMPI

Riguardo il **timestamp** dell'evento, il valore è presente di default all'interno dei log, ma non come singolo dato, ma bensì suddiviso in ora, minuto, secondo ecc...

Questa configurazione rende il dato poco leggibile e gli eventi più difficili da identificare nel tempo.

Per questo motivo anche questo dato verrà estratto dal `_raw`.

Ora		<code>_time</code> ▼	2025-08-04T13:06:40.000+02:00
<input type="checkbox"/>	<code>date_hour</code> ▼	13	▼
<input type="checkbox"/>	<code>date_mday</code> ▼	4	▼
<input type="checkbox"/>	<code>date_minute</code> ▼	6	▼
<input type="checkbox"/>	<code>date_month</code> ▼	august	▼
<input type="checkbox"/>	<code>date_second</code> ▼	40	▼
<input type="checkbox"/>	<code>date_wday</code> ▼	monday	▼
<input type="checkbox"/>	<code>date_year</code> ▼	2025	▼
<input type="checkbox"/>	<code>date_zone</code> ▼	local	▼

Un'altra problematica che sorge con il dato del timestamp riguarda la sintassi di utilizzo del comando `rex`, che funziona estraendo stringhe caratteri e spazi dal `_raw`.

Essendo il dato del tempo, suddiviso in 5 stringhe [Thu Aug 04 2025 13:06:40], seguite da altri dati, che dovremmo ignorare, per evitare che la query diventi eccessivamente lunga e complessa, ho preferito estrarre singolarmente il dato del tempo dell'evento.

Dal menu di sinistra, sotto i campi disponibili, si accede alla sezione di creazione dei nuovi campi

[+ Estrai nuovi campi](#)

Una volta cliccato, troveremo una configurazione guidata, che ci farà scegliere un log di esempio e selezionare su di esso la posizione dei dati che ci interessano all'interno del `_raw`.

Seleziona campi

Evidenziare uno o più valori nell'evento di esempio per creare i campi. È possibile indicare un valore come obbligatorio, il che significa che i valori evidenziati nell'evento di esempio per modificarli. Per evidenziare il testo che fa già parte di un'estrazione esistente, disabilitare prima

Thu Aug 04 2025 13:06:40 mailsv1 sshd[5276]: Failed password for invalid user appserver from 194.8.74.23 port 3351 ssh2

Estrai

Richiedi

Nome campo

event_time

Valore di esempio

Thu Aug 04 2025 13:06:40

Add Extraction

Dopo aver impostato un nome al campo del dato [event_time], il tool intercetterà in automatico lo stesso dato in tutti i log dove è disponibile.

_raw
Thu Aug 04 2025 13:06:40 mailsv1 sshd[5276]: Failed password for invalid user appserver from 194.8.74.23 port 3351 ssh2
Thu Aug 04 2025 13:06:40 mailsv1 sshd[1039]: Failed password for root from 194.8.74.23 port 3768 ssh2
Thu Aug 04 2025 13:06:40 mailsv1 sshd[5258]: Failed password for invalid user testuser from 194.8.74.23 port 3626 ssh2
Thu Aug 04 2025 13:06:40 mailsv1 sshd[1165]: Failed password for apache from 194.8.74.23 port 4604 ssh2
Thu Aug 04 2025 13:06:40 mailsv1 sshd[3760]: Failed password for invalid user mongodb from 194.8.74.23 port 2472 ssh2
Thu Aug 04 2025 13:06:40 mailsv1 sshd[4998]: Failed password for mail from 194.8.74.23 port 1552 ssh2
Thu Aug 04 2025 13:06:40 mailsv1 sshd[1930]: Failed password for games from 194.8.74.23 port 3007 ssh2
Thu Aug 04 2025 13:06:40 mailsv1 sshd[5801]: Failed password for invalid user desktop from 194.8.74.23 port 2285 ssh2

Concludendo la configurazione guidata, troveremo il campo appena creato tra quelli disponibili.

i	Ora	Evento																																				
▼	04/08/25 13:06:40,000	Thu Aug 04 2025 13:06:40 mailsv1 sshd[5276]: Failed password for invalid user appserver from 194.8.74.23 port 3351 ssh2																																				
		Azioni evento ▼																																				
		<table><tr><th>Tipo</th><th>Campo</th><th>Valore</th><th>Azioni</th></tr><tr><td>Selezionato</td><td><input checked="" type="checkbox"/> host ▼</td><td>Lenovo</td><td>▼</td></tr><tr><td></td><td><input checked="" type="checkbox"/> source ▼</td><td>tutorialdata.zip:\mailsv\secure.log</td><td>▼</td></tr><tr><td></td><td><input checked="" type="checkbox"/> sourcetype ▼</td><td>www1/secure</td><td>▼</td></tr><tr><td>Evento</td><td><input type="checkbox"/> clientip ▼</td><td>194.8.74.23</td><td>▼</td></tr><tr><td></td><td><input type="checkbox"/> event_time ▼</td><td>Thu Aug 04 2025 13:06:40</td><td>▼</td></tr><tr><td></td><td><input type="checkbox"/> reason ▼</td><td>invalid user</td><td>▼</td></tr><tr><td></td><td><input type="checkbox"/> user ▼</td><td>appserver</td><td>▼</td></tr><tr><td></td><td>Ora ⚙</td><td>_time ▼</td><td>2025-08-04T13:06:40.000+02:00</td></tr></table>	Tipo	Campo	Valore	Azioni	Selezionato	<input checked="" type="checkbox"/> host ▼	Lenovo	▼		<input checked="" type="checkbox"/> source ▼	tutorialdata.zip:\mailsv\secure.log	▼		<input checked="" type="checkbox"/> sourcetype ▼	www1/secure	▼	Evento	<input type="checkbox"/> clientip ▼	194.8.74.23	▼		<input type="checkbox"/> event_time ▼	Thu Aug 04 2025 13:06:40	▼		<input type="checkbox"/> reason ▼	invalid user	▼		<input type="checkbox"/> user ▼	appserver	▼		Ora ⚙	_time ▼	2025-08-04T13:06:40.000+02:00
Tipo	Campo	Valore	Azioni																																			
Selezionato	<input checked="" type="checkbox"/> host ▼	Lenovo	▼																																			
	<input checked="" type="checkbox"/> source ▼	tutorialdata.zip:\mailsv\secure.log	▼																																			
	<input checked="" type="checkbox"/> sourcetype ▼	www1/secure	▼																																			
Evento	<input type="checkbox"/> clientip ▼	194.8.74.23	▼																																			
	<input type="checkbox"/> event_time ▼	Thu Aug 04 2025 13:06:40	▼																																			
	<input type="checkbox"/> reason ▼	invalid user	▼																																			
	<input type="checkbox"/> user ▼	appserver	▼																																			
	Ora ⚙	_time ▼	2025-08-04T13:06:40.000+02:00																																			

5

Ora siamo pronti per creare la query richiesta, che sarà:

```
source="tutorialdata.zip:*" "Failed password" | rex "Failed password for (?<reason>invalid user )?(?<user>\S+) from (?<clientip>\S+) port \S+ ssh2"
```

ANALISI:

```
source="tutorialdata.zip:*" "Failed password"
```

- `source="tutorialdata.zip:*"`
Questa parte restringe la ricerca ai file di log che provengono dall'archivio tutorialdata.zip. L'asterisco * funge da wildcard, includendo tutti i file all'interno dell'archivio.
- `"Failed password"`
Questo è un filtro di testo. Splunk cerca tutti gli eventi che contengono l'esatta frase "Failed password". Questo assicura che vengano considerati solo i tentativi di accesso non riusciti.

```
| rex "Failed password for (?<reason>invalid user )?(?<user>\S+) from (?<clientip>\S+) port \S+ ssh2"
```

- `| rex`
Questo è il comando di Splunk che usa le espressioni regolari (regex) per estrarre campi dal testo grezzo di ogni evento.
- `"Failed password for"`
L'espressione inizia con la stringa di testo fissa "Failed password for" che fa da punto di riferimento.
- `(?<reason>invalid user)?` Questo è un gruppo di cattura nominato.
 - `?<`: Indica che il testo catturato all'interno delle parentesi () sarà assegnato a un campo.
 - `reason`: È il nome che verrà dato al campo estratto.
 - `invalid user`: È il testo che si cerca di catturare.
 - `?`: Il punto interrogativo rende l'intero gruppo opzionale, il che significa che la query funzionerà anche se la stringa "invalid user" non è presente nel log.
- `(?<user>\S+)` Un altro gruppo di cattura nominato.
 - `user`: Il nome del campo.
 - `\S+`: Questo metacarattere cattura una o più caratteri non-spazio, che in questo caso corrispondono al nome utente.
- `from` Testo fisso che serve da delimitatore.
- `(?<clientip>\S+)` Un altro gruppo di cattura.
 - `clientip`: Il nome del campo, che identifica l'indirizzo IP di origine.
 - `\S+`: Cattura l'indirizzo IP.
- `port \S+ ssh2"`
Questa parte finale dell'espressione serve a ignorare il numero di porta (\S+ cattura il valore ma non lo assegna a un campo) e il testo "ssh2" che segue, assicurando che l'espressione regolare si concluda correttamente.

RISULTATI

Avviata la query, nella lista dei campi disponibili, troveremo gli argomenti estratti dalla query disponibili per essere analizzati

event_time

32 Valori, 100% di eventi

Selezionato

Sì

No

Report

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Primi 10 valori	Conteggio	%
Sun Jul 31 2025 13:06:36	8.820	4,421%
Mon Aug 01 2025 13:06:36	8.328	4,174%
Tue Aug 02 2025 13:06:39	8.028	4,024%
Wed Aug 03 2025 13:06:40	7.968	3,994%
Sat Jul 30 2025 13:06:37	7.908	3,964%
Sun Jul 31 2025 13:06:39	7.734	3,876%
Sat Jul 30 2025 13:06:36	7.554	3,786%
Tue Aug 02 2025 13:06:37	7.374	3,696%
Mon Aug 01 2025 13:06:40	7.344	3,681%
Mon Aug 01 2025 13:06:39	7.320	3,669%

clientip

>100 Valori, 99,447% di eventi

Selezionato

Sì

No

Report

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Primi 10 valori	Conteggio	%
87.194.216.51	5.688	2,867%
211.166.11.101	4.434	2,235%
128.241.220.82	3.732	1,881%
109.169.32.135	3.090	1,557%
194.215.205.19	3.084	1,554%
216.221.226.11	2.544	1,282%
65.19.167.94	1.716	0,865%
188.138.40.166	1.698	0,856%
107.3.146.207	1.692	0,853%
95.130.170.231	1.674	0,844%

user

>100 Valori, 99,447% di eventi

Selezionato

Sì

No

Report

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Primi 10 valori	Conteggio	%
root	8.958	4,515%
administrator	6.120	3,084%
admin	5.628	2,836%
operator	5.538	2,791%
mail	4.518	2,277%
mailman	4.512	2,274%
irc	3.864	1,947%
email	3.756	1,893%
games	3.606	1,817%
sys	3.516	1,772%

reason

1 Valore, 72,207% di eventi

Selezionato

Sì

No

Report

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Valori	Conteggio	%
invalid user	144.066	100%

QUERY N. 2

TROVARE TUTTE LE SESSIONI SSH APERTE CON SUCCESSO.
FILTRARE I RISULTATI PER L'UTENTE *djohnson*, MOSTRARE IL TIMESTAMP E L'ID UTENTE.

Per intercettare i log delle sessioni SSH aperte con successo dallo user *djohnson*, creiamo questa query iniziale:

```
source="tutorialdata.zip:*" "Accepted password for djohnson"
```

Questa parte cerca nel file di log *tutorialdata.zip* gli eventi che contengono la stringa "Accepted password for djohnson".

Questa stringa è l'indicatore di una sessione SSH aperta con successo per quell'utente.

Di seguito un log estratto dai risultati, con i campi disponibili.

04/08/25

13:06:40,000

Thu Aug 04 2025 13:06:40 mailsv1 sshd[54545]: Accepted password for djohnson from 10.3.10.46 port 5143 ssh2

Azioni evento ▼

Tipo	Campo	Valore	Azioni
Selezionato	host	Lenovo	▼
	source	tutorialdata.zip:./mailsv/secure.log	▼
	sourcetype	www1/secure	▼
Evento	event_time	Thu Aug 04 2025 13:06:40	▼
Ora	_time	2025-08-04T13:06:40.000+02:00	
Default	index	main	▼
	linecount	1	▼
	punct	_____::_____:_____..._____	▼
	splunk_server	Lenovo	▼

Analogamente alla query precedente, anche in questo caso i dati utili sono contenuti nel *_raw* del log, tranne *event_time*, che è stato creato in precedenza.

Creiamo quindi, sempre con l'utilizzo del comando di ricerca rex, la query per estrarre i dati richiesti

```
source="tutorialdata.zip:*" "Accepted password for djohnson" | rex "Accepted password for (?<user>\S+) from (?<clientip>\S+) port \S+ ssh2"
```

RISULTATI

event_time

32 Valori, 100% di eventi

Selezionato

Sì

No

Report

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Primi 10 valori	Conteggio	%	
Wed Aug 03 2025 13:06:40	276	4,817%	
Sat Jul 30 2025 13:06:36	270	4,712%	
Mon Aug 01 2025 13:06:40	252	4,398%	
Tue Aug 02 2025 13:06:39	252	4,398%	
Wed Aug 03 2025 13:06:39	246	4,293%	
Mon Aug 01 2025 13:06:36	240	4,188%	
Fri Jul 29 2025 13:06:39	222	3,874%	
Sat Jul 30 2025 13:06:37	222	3,874%	
Fri Jul 29 2025 13:06:36	216	3,77%	
Mon Aug 01 2025 13:06:39	216	3,77%	

user

1 Valore, 100% di eventi

Selezionato

Sì

No

Report

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Valori	Conteggio	%
djohnson	5.730	100%

QUERY N. 3

TROVARE TUTTI I TENTATIVI DI ACCESSO FALLITI PROVENIENTI DALL'INDIRIZZO IP "86.212.199.60".
MOSTRARE IL TIMESTAMP, IL NOME UTENTE E IL NUMERO DI PORTA.

```
source="tutorialdata.zip:*" "Failed password" | rex "Failed password for (?<user>\S+)  
from (?<clientip>\S+) port (?<port>\S+) ssh2" | search clientip="86.212.199.60"
```

La query, analoga alle precedenti, estrae dal `_raw` del log i dati richiesti.

La differenza nella richiesta sta che i dati da intercettare sono solo quelli indirizzabili all'indirizzo IP 86.212.199.60.

Per questa richiesta è stata aggiunta alla query una parte finale `| search clientip="86.212.199.60"`

Questo comando filtra i risultati della ricerca precedente, mostrando solo gli eventi in cui il campo `clientip` corrisponde esattamente all'indirizzo IP "86.212.199.60".

RISULTATI

clientip

1 Valore, 100% di eventi

Selezionato

Si

No

Report

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Valori	Conteggio	%
86.212.199.60	276	100%

event_time

11 Valori, 100% di eventi

Selezionato

Si

No

Report

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Primi 10 valori	Conteggio	%
Mon Aug 01 2025 13:06:40	72	26,087%
Tue Aug 02 2025 13:06:37	42	15,217%
Sun Jul 31 2025 13:06:37	36	13,043%
Wed Aug 03 2025 13:06:40	36	13,043%
Thu Jul 28 2025 13:06:40	24	8,696%
Thu Jul 28 2025 13:06:39	18	6,522%
Fri Jul 29 2025 13:06:40	12	4,348%
Thu Aug 04 2025 13:06:40	12	4,348%
Tue Aug 02 2025 13:06:39	12	4,348%
Tue Aug 02 2025 13:06:40	6	2,174%

user

23 Valori, 100% di eventi

Selezionato

Sì

No

Report

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Primi 10 valori	Conteggio	%	
root	54	19,565%	
jira	24	8,696%	
mail	24	8,696%	
nagios	18	6,522%	
nobody	18	6,522%	
britany	12	4,348%	
ftp	12	4,348%	
games	12	4,348%	
hammer	12	4,348%	
ncsd	12	4,348%	

port

45 Valori, 100% di eventi

Selezionato

Sì

No

Report

Media nel tempo

Valore massimo nel tempo

Valore minimo nel tempo

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Med: 2543.1304347826085 **Min:** 1054 **Max:** 4938 **Std Dev:** 1118.0394582471715

Primi 10 valori	Conteggio	%	
1783	12	4,348%	
1054	6	2,174%	
1090	6	2,174%	
1097	6	2,174%	
1301	6	2,174%	
1309	6	2,174%	
1323	6	2,174%	
1430	6	2,174%	
1431	6	2,174%	
1465	6	2,174%	

QUERY N. 4

IDENTIFICARE GLI INDIRIZZI IP CHE HANNO TENTATO DI ACCEDERE ("FAILED PASSWORD") AL SISTEMA PIÙ DI 5 VOLTE.
MOSTRARE L'INDIRIZZO IP E IL NUMERO DI TENTATIVI.

Per identificare gli IP che hanno fallito più di 5 tentativi di accesso, è necessario contare i fallimenti per ogni indirizzo IP e poi filtrare i risultati.

```
source="tutorialdata.zip:*" "Failed password" | rex "Failed password for (?<user>\S+) from (?<clientip>\S+)" | stats count by clientip | where count > 5
```

ANALISI

```
source="tutorialdata.zip:*" "Failed password" | rex "Failed password for (?<user>\S+) from (?<clientip>\S+)"
```

La ricerca iniziale filtra gli eventi che contengono *"Failed password"*.

Il comando rex estrae l'indirizzo IP di origine e lo assegna al campo `clientip`.

Aggregazione: `| stats count by clientip`

Questo comando aggrega gli eventi.

`stats` è un comando statistico che esegue calcoli sui risultati della ricerca.

`count` conta il numero di eventi per ogni gruppo.

`by clientip` raggruppa i risultati in base all'indirizzo IP, creando una riga separata per ogni IP e contando i tentativi di accesso falliti.

Filtro: `| where count > 5`

Questo comando filtra i risultati del comando `stats`.

`where` scarta le righe in cui il conteggio (count) dei tentativi non è superiore a 5.

RISULTATI

clientip

>100 Valori, 27,24% di eventi

Selezionato

Si

No

Report

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Primi 10 valori	Conteggio	%
87.194.216.51	1.542	2,837%
211.166.11.101	1.164	2,142%
128.241.220.82	978	1,8%
109.169.32.135	852	1,568%
194.215.205.19	834	1,534%
216.221.226.11	618	1,137%
188.138.40.166	522	0,96%
107.3.146.207	504	0,927%
59.162.167.100	504	0,927%
108.65.113.83	450	0,828%

QUERY N. 5

TROVARE TUTTI GLI INTERNAL SERVER ERROR.

Per identificare gli "Internal Server Error", abbiamo utilizzato una ricerca basata sul codice di stato HTTP standard **500** invece che sulla stringa di testo.

Questo approccio è più affidabile, perché i messaggi di errore nei log possono variare, ma il codice di stato 500 rimane un identificatore univoco di un problema lato server.

Query utilizzata: `source="tutorialdata.zip:*" status=500`

Questa query filtra gli eventi per il campo **status** uguale a **500**, consentendo di individuare in modo preciso e veloce tutti gli "Internal Server Error" presenti nei log.

RISULTATI

status

×

1 Valore, 100% di eventi

Selezionato

Sì

No

Report

Media nel tempo

Valore massimo nel tempo

Valore minimo nel tempo

Primi valori

Primi valori nel tempo

Valori rari

Eventi con questo campo

Med: 500 Min: 500 Max: 500 Std Dev: 0

Valori	Conteggio	%
500	4.398	100%

CONCLUSIONI

ANALISI DEI TENTATIVI DI ACCESSO FALLITI

L'analisi dei log ha rivelato una situazione di sicurezza critica per il sistema. Invece di trovare evidenze di un'infezione da malware, come suggerito dal tema del progetto, i dati puntano chiaramente a un **attacco brute-force** massiccio e mirato.

Un totale di **199.518** eventi di *"Failed password"* è stato registrato.

Gli eventi sono avvenuti con una cadenza giornaliera sistematica nel periodo analizzato, con migliaia di tentativi di accesso registrati nel giro di pochi istanti.

Questo volume massiccio è un chiaro indicatore di un attacco di tipo **brute-force** mirato al sistema.

I dati mostrano che gli attaccanti si sono concentrati su account con privilegi elevati, con **"root"** come l'utente più bersagliato, registrando **8.958** tentativi di accesso falliti.

Altri utenti frequentemente presi di mira includono *"administrator"* e *"admin"*.

La stragrande maggioranza dei fallimenti (**144.066**) è dovuta all'uso di utenti non validi (*"invalid user"*), suggerendo che gli attaccanti stanno cercando di indovinare sia il nome utente che la password.

L'attacco sembra essere distribuito ma con una concentrazione su alcuni indirizzi IP.

Il conteggio dei tentativi per singolo IP mostra che alcuni sono particolarmente attivi, come **87.194.216.51** con **5.688** tentativi.

L'analisi di un IP specifico, **86.212.199.60**, ha rivelato **276** tentativi falliti, di cui la maggior parte era diretta all'utente *"root"*.

È evidente che il sistema è sotto un attacco costante e mirato.

ANALISI DELLE SESSIONI SSH

L'utente **"djohnson"** ha aperto un totale di **5.730** sessioni SSH con successo.

Nell'ambito di questo attacco, questi dati indicano che il sistema è stato violato tramite questo username.

ANALISI DEGLI "INTERNAL SERVER ERROR"

Dall'analisi dei log sono stati rilevati **4.398** eventi con status=500, che indicano *"Internal Server Error"*. Questo numero significativo di errori suggerisce che il server sta incontrando problemi di stabilità o configurazione.

Questi errori potrebbero essere sfruttati da un attaccante o potrebbero essere il risultato di un carico eccessivo causato proprio dagli attacchi.

CONCLUSIONI FINALI

Nel complesso, l'analisi dei log rivela una situazione di sicurezza critica per il sistema.

L'elevato volume di tentativi di accesso falliti indica un attacco brute-force sistematico e costante.

La concentrazione su utenti con privilegi elevati come *"root"* e *"admin"* suggerisce un obiettivo specifico da parte degli attaccanti.

Inoltre, la presenza di migliaia di *"Internal Server Error"* indica una debolezza aggiuntiva del sistema, rendendolo potenzialmente vulnerabile a ulteriori attacchi.

Le attività dell'utente *"djohnson"*, indicano chiaramente che l'attacco ha avuto successo e il sistema è stato violato.

In sintesi, il sistema è sotto attacco, è stato violato e ha evidenti problemi di stabilità e necessita di un'immediata implementazione di misure di sicurezza.

INCIDENT RESPONSE

CONTENIMENTO DELL'ATTACCO

Il primo passo è isolare gli indirizzi IP che hanno eseguito l'attacco, in particolare quelli più attivi, queste connessioni vanno bloccate tramite il firewall per fermare immediatamente i tentativi di accesso.

Successivamente, si esegue un'ispezione degli account più bersagliati. In caso di sospetto, si disabilitano temporaneamente gli account o si reimpostano le password.

INDAGINE FORENSE

Si analizzano i log, la memoria e i file di sistema per raccogliere prove.

In particolare, si cerca di capire quali tentativi di accesso abbia alla fine avuto successo.

Si analizzano anche i **4.398 Internal Server Error** (codice 500) per determinare se sono stati causati dagli attacchi stessi o se hanno contribuito a rendere il sistema più vulnerabile.

ELIMINAZIONE DELLA MINACCIA

Una volta identificata la causa principale della violazione, si procede con l'eliminazione. Si rimuovono eventuali backdoor o script lasciati dall'attaccante.

Tutte le password degli account compromessi o a rischio vengono immediatamente reimpostate.

Le vulnerabilità del servizio SSH che sono state sfruttate per l'attacco vengono patchate.

Si esegue una scansione completa del sistema per verificare la presenza di malware, anche se i log non ne hanno mostrato evidenze dirette.

RIPRISTINO E RAFFORZAMENTO

Dopo aver eliminato la minaccia, si ripristinano i servizi compromessi.

Si implementano nuove misure di sicurezza per prevenire futuri attacchi brute-force, ad esempio, si applicano limiti ai tentativi di accesso falliti da un singolo IP e si impone una politica di password più robusta.

Infine, si attiva un monitoraggio continuo per rilevare anomalie e tentativi di accesso sospetti, assicurando una maggiore resilienza del sistema in futuro.