Progetto

Importate su Splunk i dati di esempio “tutorialdata.zip”:

● Crea una query Splunk per identificare tutti i tentativi di accesso falliti "Failed password". La query dovrebbe mostrare il timestamp, l'indirizzo IP di origine, il nome utente e il motivo del fallimento.

● 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.

● 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.

● 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.

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

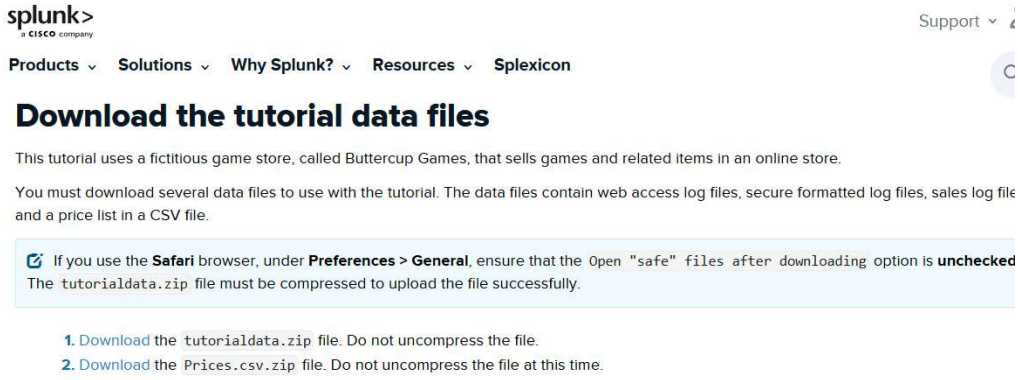
Trarre delle conclusioni sui log analizzati utilizzando AI.

Introduzione a Splunk

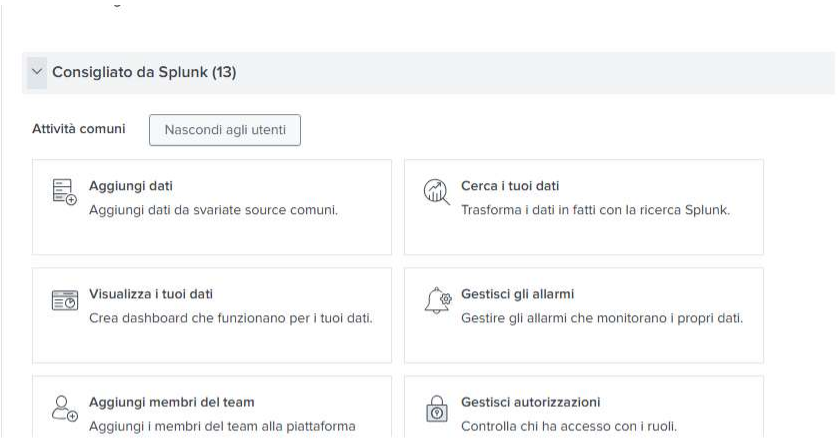
Splunk è una piattaforma software utilizzata per cercare, monitorare, analizzare e visualizzare i dati generati dalle macchine in tempo reale. Questi dati possono essere raccolti da diverse sorgenti come applicazioni, log di sistema, dati di rete, dispositivi IoT, file di registro e altro ancora. Una volta raccolti e analizzati i dati, gli utenti possono eseguire delle ricerche tramite query per estrarre informazioni utili e produrre dei report per visualizzare i risultati delle analisi svolte.

L'esercizio ci chiede di analizzare i dati del file “tutorialdata.zip” e creare delle query per trovare specifiche informazione da analizzare.

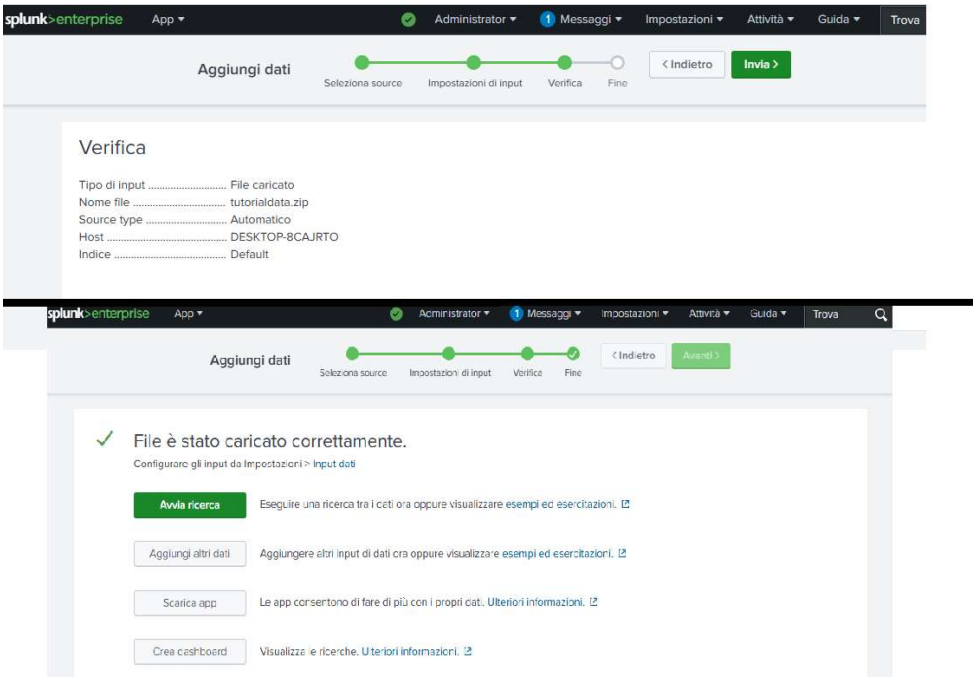
Come prima cosa andiamo sulla pagina web di Splunk dove troveremo il file “tutorialdata.zip” da scaricare.



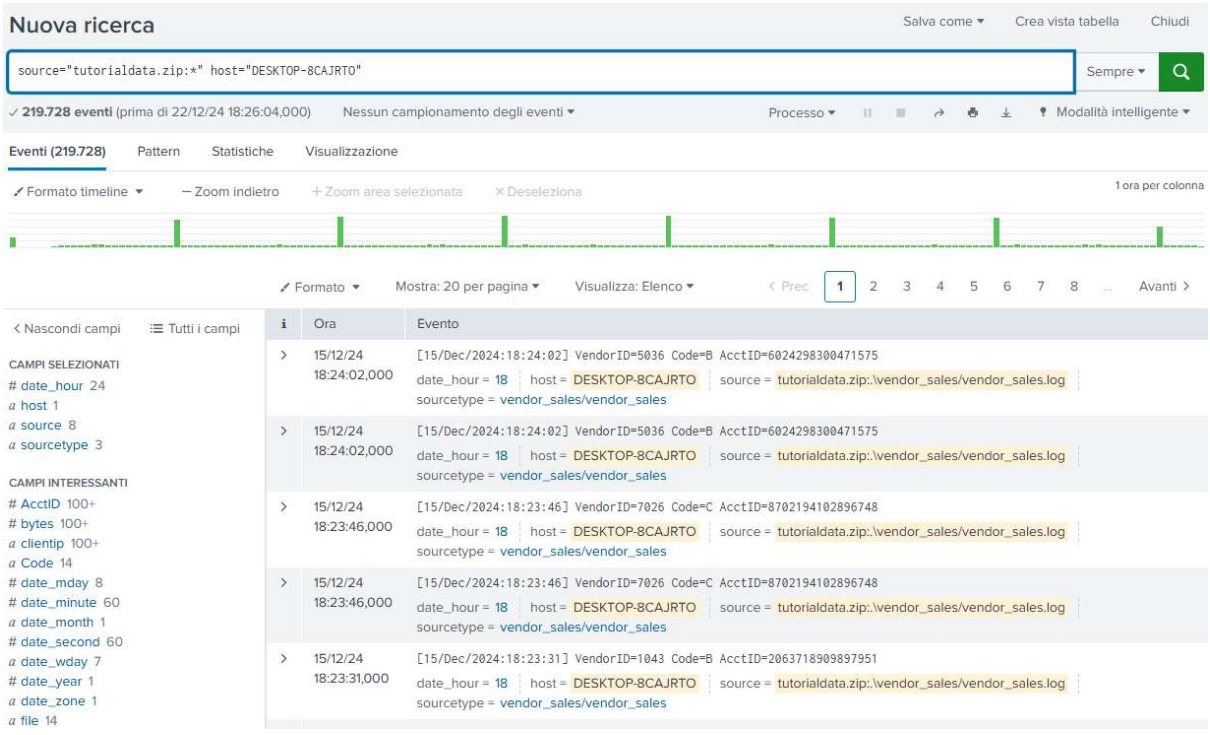
Una volta scaricato lo possiamo importare su Splunk per analizzarlo, per farlo basta cliccare su "**Aggiungi dati**", che si trova alla pagina iniziale, e poi "**Carica**".



Lasciamo le impostazioni di input così come sono, clicchiamo su verifica e andiamo avanti, una volta che il file è stato caricato correttamente possiamo avviare la ricerca.



Avviata la ricerca ci apparirà questa schermata principale, da qui possiamo eseguire le query e visualizzare i risultati. Nella barra di ricerca in alto ci viene mostrata la query di base, mentre subito in basso a sinistra, il numero di eventi (log) che la ricerca ci ha restituito, in questo caso 219.728. Con il proseguire dell'esercizio vedremo come si andranno a costruire su essa le nostre query e come si andrà ad abbassare il numero di eventi che ci viene restituito.



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**

La query creata è la seguente:

```
source="tutorialdata.zip:*" host="DESKTOP-8CAJRTO" "failed password"
| rex "Failed password for (?<username>\S+) from (?<src_ip>\S+)"
| rex "reason: (?<failure_reason>.+)"
| fields _time, src_ip, username, failure_reason
| sort - _time
```

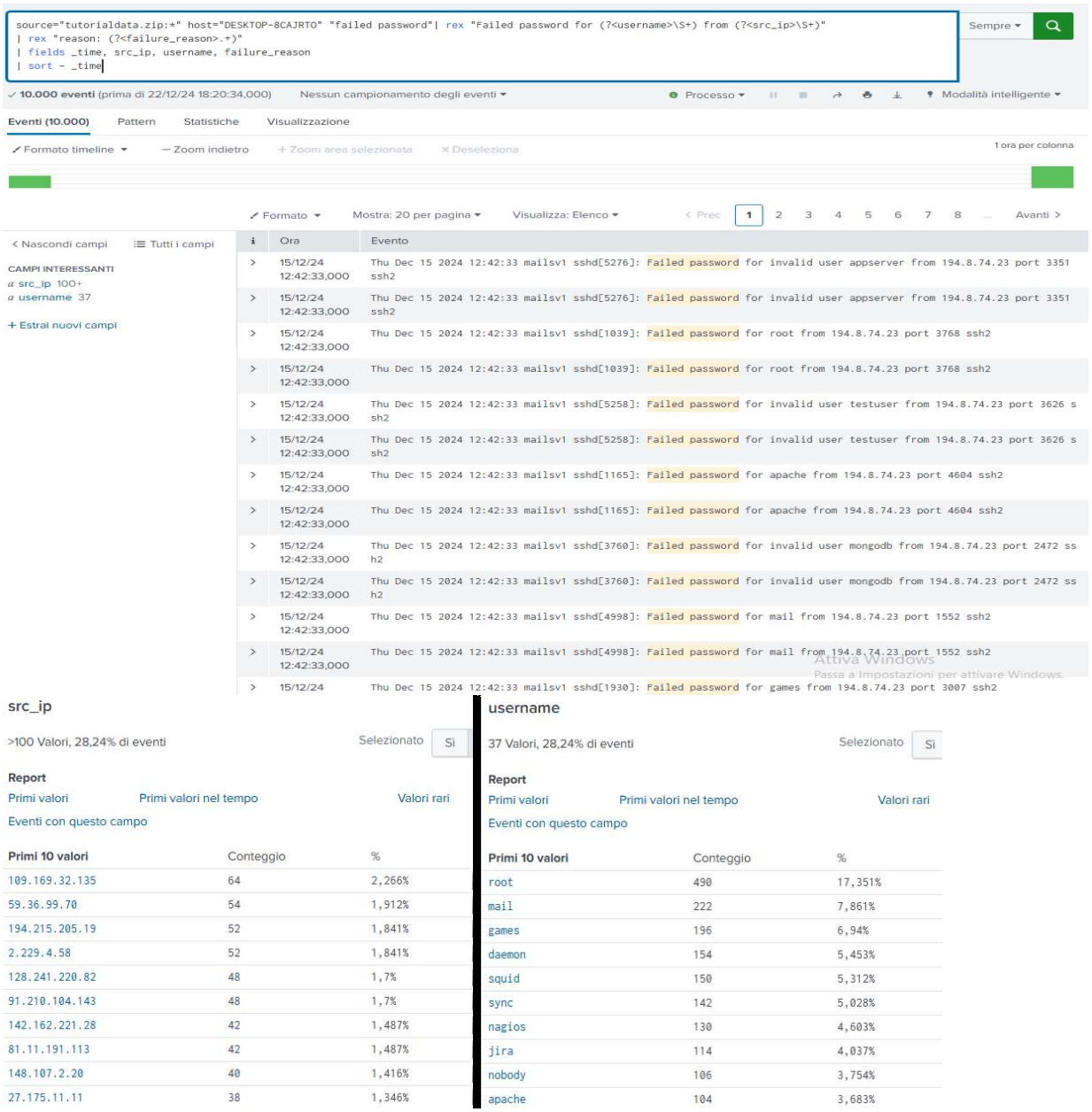
Esaminandola possiamo vedere che essa cerca tutti gli eventi nel file sorgente **tutorialdata.zip** provenienti dall'host **DESKTOP-8CAJRTO** e tramite il filtro **failed password** stiamo specificando che ci vengano mostrati tutti i tentativi di accesso falliti.

Il comando **rex** invece viene utilizzato per estrarre informazioni dai log. In questo caso, è usato per:

- Catturare il nome utente associato al tentativo di accesso fallito "**(?<username>\S+)**" .
- Catturare l'indirizzo IP di origine "**(?<src_ip>\S+)".**
- Estrarre il campo "**(?<failure_reason>.+)**", che rappresenta il motivo del fallimento del tentativo di accesso.

Il comando **fields** serve a selezionare solo i campi che ci interessano. In questo caso, stiamo selezionando: Il timestamp (**_time**) , quindi data e ora dell'evento, l'indirizzo IP da cui è stato effettuato il tentativo di login, il nome utente usato nel tentativo di login fallito e il motivo del fallimento (ad esempio "Password incorrect").
Questa parte del comando serve a semplificare l'output, mostrando solo le informazioni rilevanti.

Il comando **sort - _time** infine, ordina i risultati in base al tempo di origine dell'evento dove l'operatore "**-**" indica un ordinamento decrescente (dal più recente al più vecchio). Se avessimo scritto **sort _time** senza il meno, l'ordinamento sarebbe stato crescente.



Analizzando questi dati possiamo notare che:

Nella colonna "**src_ip**" ci vengono mostrati gli indirizzi IP di origine più comuni che hanno generato tentativi di accesso falliti, ad esempio:

109.169.23.135 ha generato 64 tentativi falliti (probabile attività sospetta o brute force).

Nella colonna "**username**" ci vengono evidenziati quali nomi utente sono stati presi di mira, ad esempio:

root è stato tentato 490 volte.

Dal timestamp possiamo notare quando gli eventi si sono verificati. I tentativi provengono dallo stesso giorno e ora, è possibile che un attacco potrebbe essere ancora in corso oppure che si tratti di un'attività mirata a una finestra di tempo specifica.

Per quanto riguarda il motivo del fallimento, nella prima stringa notiamo scritto:

"**Failed password for invalid user appserver**"

Ciò indica che il nome utente utilizzato nel tentativo di login non esiste o non è autorizzato, di conseguenza non è valido. Caso contrario invece quando notiamo scritto:

"Failed password for root"

Significa che il nome utente è valido ma la password è errata.

In conclusione abbiamo 10.000 eventi visualizzati, il sistema è stato sottoposto a un carico elevato di tentativi di accesso falliti, si tratta quindi di un attacco su larga scala, probabilmente un attacco brute force dove gli attaccanti stanno cercando di accedere usando account comuni o standard in un sistema.

Azioni raccomandate:

1. **Blocco degli IP sospetti**:
   - Implementare il blocco degli IP più ricorrenti nella lista (ad esempio: 109.169.23.135, 59.36.96.56).
   - Usare strumenti come **fail2ban** per rilevare e bloccare automaticamente attività sospette.
2. **Rafforzamento della sicurezza**:
   - Garantire che tutte le password siano sicure e non banali.
   - Disabilitare gli account non necessari o predefiniti (ad esempio: games, mail, daemon).
3. **Monitoraggio continuo**:
   - Continuare a monitorare i log per eventuali nuovi tentativi sospetti.
   - Configurare alert automatici per eventi con un volume elevato.
4. **Abilitazione di strumenti di protezione**:
   - Considerare di abilitare autenticazioni multi-fattore (MFA) e limitare gli accessi alle risorse tramite VPN.
5. **Indagine degli attacchi**:
   - Verificare se gli indirizzi IP sospetti sono associati a campagne note di attacchi brute force (ad esempio, tramite servizi di blacklist).

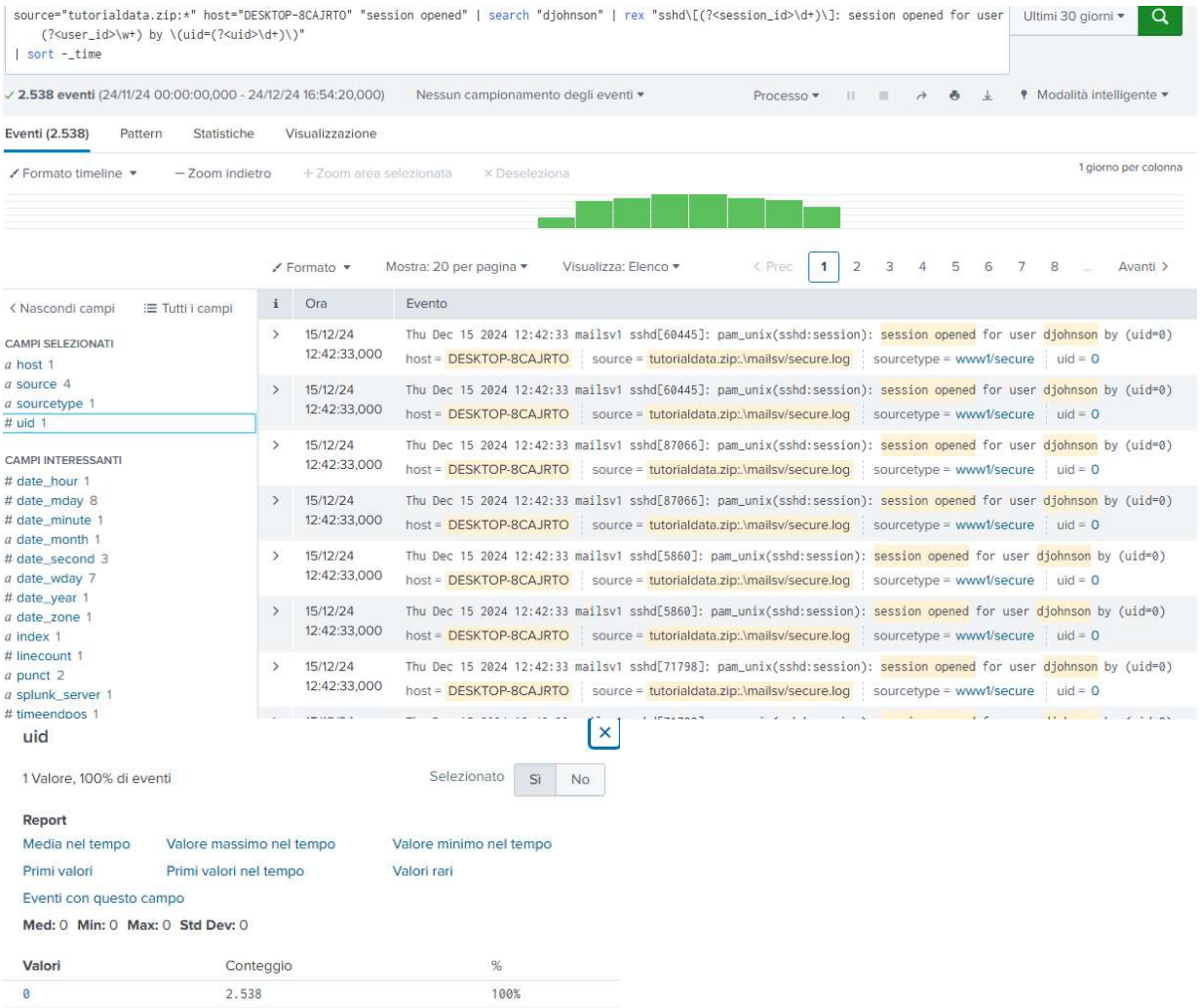
2. **Scrivi una query Splunk per trovare tutte le sessioni SSH aperte con successo. La query dovrebbe filtrare per l'utente "djohnson" e mostrare il timestamp e l'ID utente.**

   Per questa query ci basterà filtrare per "**session opened**" per trovare tutte le sessioni SSH aperte e fare una ricerca per l'utente "**djohnson**".

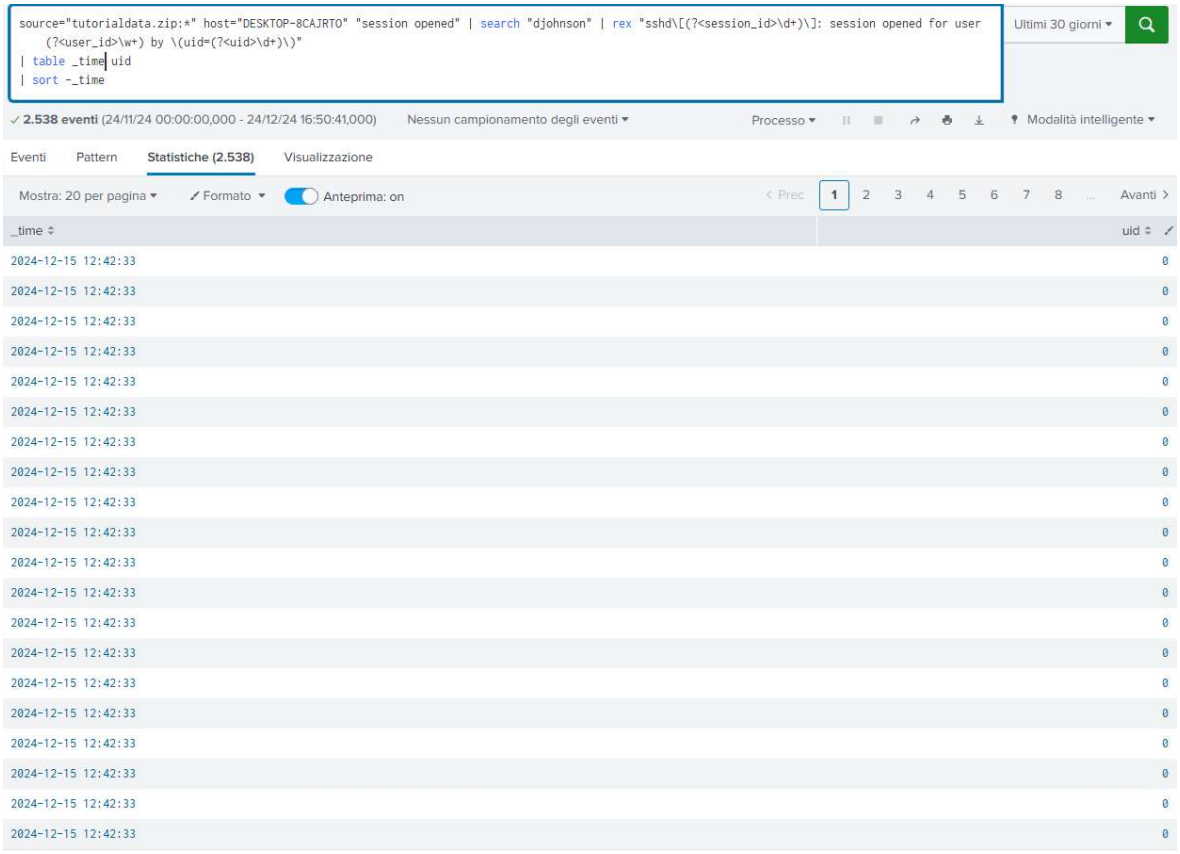
   Sempre tramite il comando **rex** estraiamo l'ID della sessione SSH rappresentato dal numero nella parentesi quadra dopo **sshd,** il nome utente (**djohnson**) e l'ID utente (**uid= 0**)

   La query utilizzata sarà la seguente:

   source="tutorialdata.zip:*" host="DESKTOP-8CAJRTO" "session opened" | search "djohnson" | rex "sshd\[(?<session_id>\d+)\]: session opened for user (?<user_id>\w+) by \(uid=(?<uid>\d+)\)"
   | sort -_time



Per avere una rappresentazione più ordinata e schematica che ci mostri il Timestamp e l'ID utente in modo chiaro possiamo utilizzare il comando **table** seguito da ciò che vogliamo ci mostri, in questo caso **_time** e **uid.**



In conclusione, dai log generati possiamo dedurre che l'utente djohnoson ha aperto molteplici sessioni SSH contemporaneamente con ID utente = 0, che rappresenta l'utente root, (possibile escalation dei privilegi o di un accesso non autorizzato).
Questo comportamento potrebbe indicare un'attività automatizzata come un tentativo di exploit o un attacco di brute force per mantenere una presenza persistente nel sistema.

Azioni raccomandate:

1. **Verifica delle sessioni SSH**:
   Controlla le sessioni attive e verifica se ci sono sessioni duplicate o sospette per l'utente **djohnson**.

2. **Rafforzamento della sicurezza SSH**:
   Disabilita l'accesso root via SSH e usa l'autenticazione a chiave SSH invece delle password.

3. **Controlla i privilegi di djohnson**:
   Assicurati che l'utente non abbia privilegi root non giustificati e verifica l'uso di comandi come sudo o su.

4. **Esamina i log**:
   Controlla il contenuto del file zip tutorialdata.zip e cerca attività sospette nel log **mailsv/secure.log**.

5. **Aggiornamenti di sicurezza**:
   Mantieni il sistema e le applicazioni aggiornati con le ultime patch di sicurezza.

6. **Monitoraggio e allarmi**:

Usa strumenti come **Fail2ban** per monitorare tentativi di accesso non autorizzati e implementa monitoraggio in tempo reale.

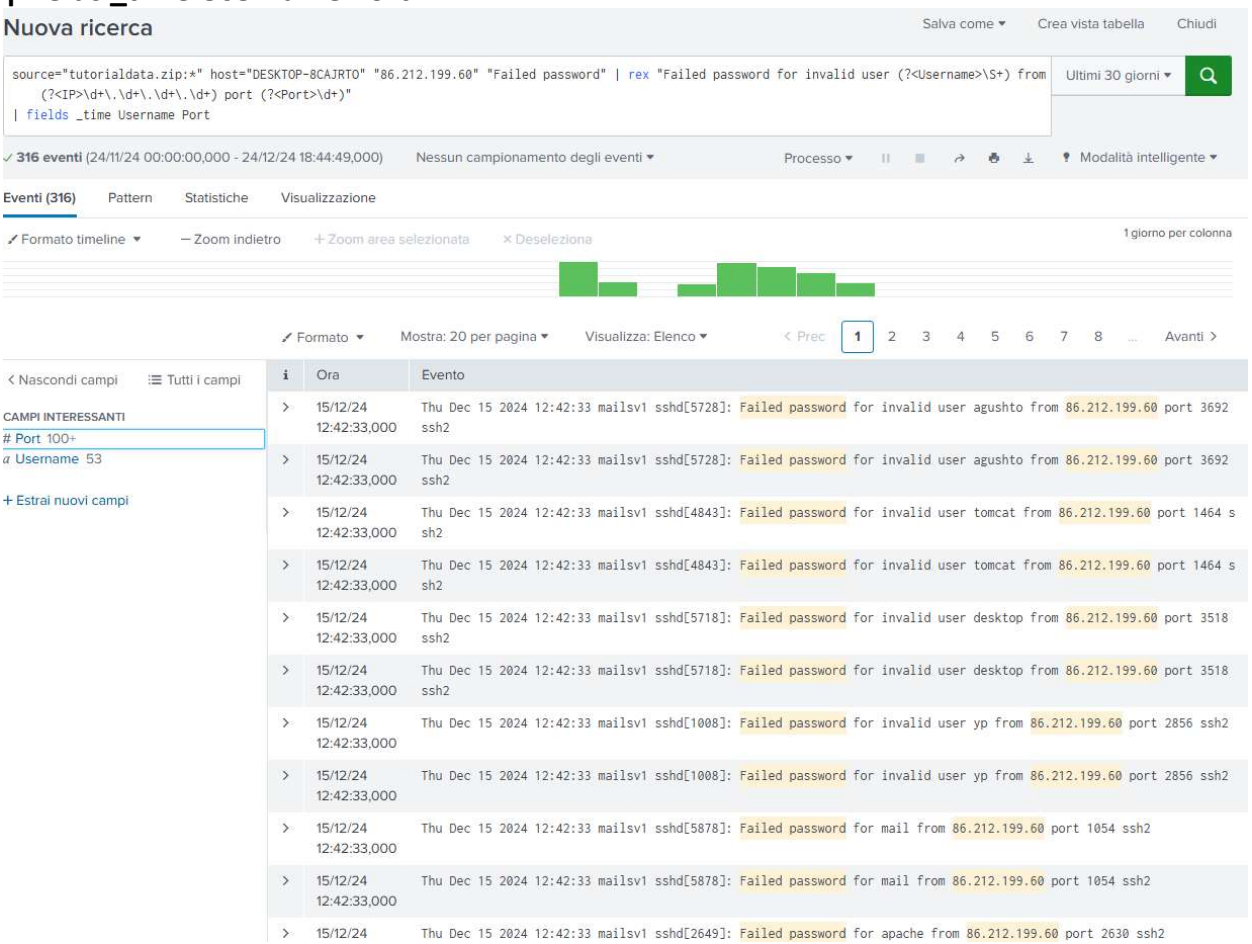
7. **Verifica dell'integrità del sistema**:
   Usa strumenti come **AIDE** per rilevare modifiche non autorizzate ai file di sistema.

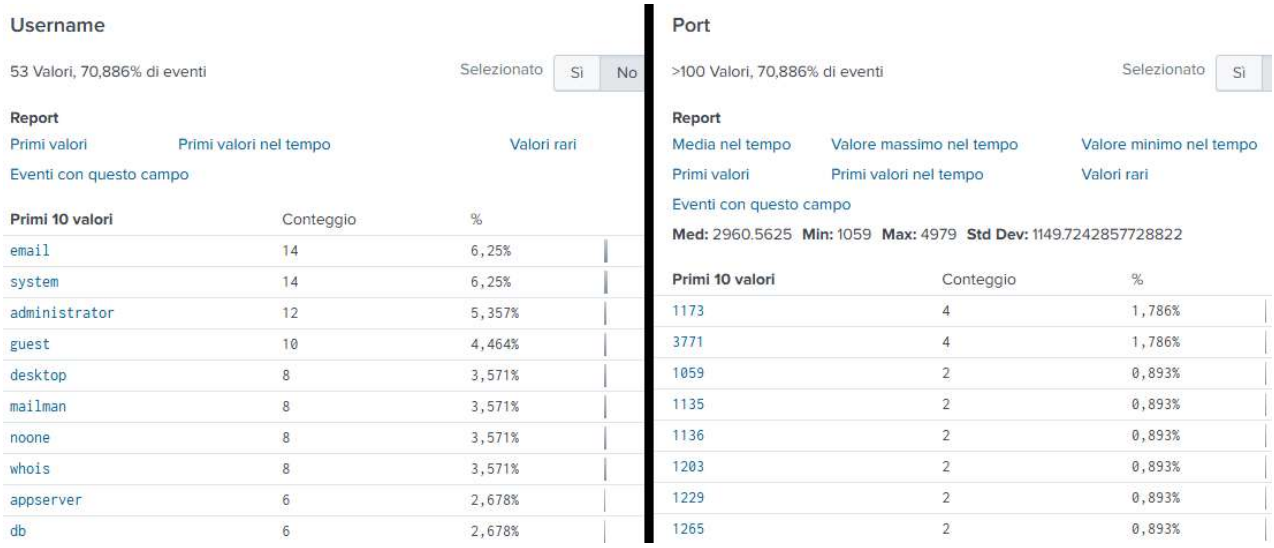
3. **Scrivi una query Splunk per trovare tutti i tentativi di accesso falliti provenienti dall'indirizzo IP "86.212.199.60". La query dovrebbe mostrare il timestamp, il nome utente e il numero di porta.**

La sintassi della query è simile alle due viste in precedenza, filtriamo la ricerca per l'indirizzo IP che ci serve (**86.212.199.60**) e per i tentativi di accesso falliti (**Failed password**) e con **rex** andiamo a estrarre le informazioni richieste in questo caso:

- Tentativi di accesso falliti (Failed password for invalid user)
- Il nome utente ((?<Username>\S+))
- L'indirizzo IP in IPv4 ((?<IP>\d+\.\d+\.\d+\.\d+))
- Il numero di porta ((?<Port>\d+))

source="tutorialdata.zip:*" host="DESKTOP-8CAJRTO" "86.212.199.60" "Failed password" | rex "Failed password for invalid user (?<Username>\S+) from (?<IP>\d+\.\d+\.\d+\.\d+) port (?<Port>\d+)"
| fields _time Username Port





Username

53 Valori, 70,886% di eventi — Selezionato: Sì / No

Report: Primi valori | Primi valori nel tempo | Valori rari | Eventi con questo campo

Primi 10 valori	Conteggio	%
email	14	6,25%
system	14	6,25%
administrator	12	5,357%
guest	10	4,464%
desktop	8	3,571%
mailman	8	3,571%
noone	8	3,571%
whois	8	3,571%
appserver	6	2,678%
db	6	2,678%

Port

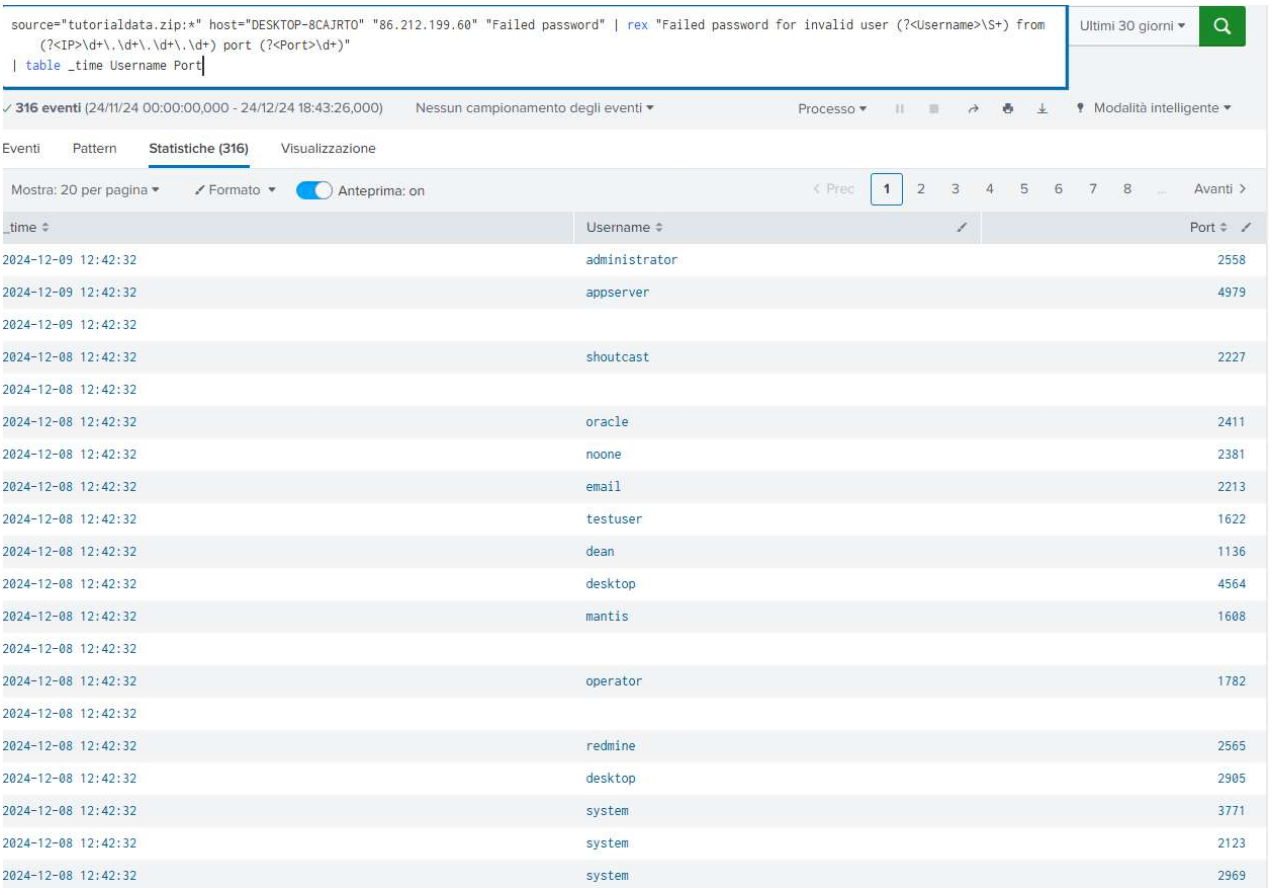
>100 Valori, 70,886% di eventi — Selezionato: Sì

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: 2960.5625 **Min:** 1059 **Max:** 4979 **Std Dev:** 1149.7242857728822

Primi 10 valori	Conteggio	%
1173	4	1,786%
3771	4	1,786%
1059	2	0,893%
1135	2	0,893%
1136	2	0,893%
1203	2	0,893%
1229	2	0,893%
1265	2	0,893%

In base alle preferenze al posto di **fields** possiamo inserire **table**



```
source="tutorialdata.zip:*" host="DESKTOP-8CAJRTO" "86.212.199.60" "Failed password" | rex "Failed password for invalid user (?<Username>\S+) from
    (?<IP>\d+\.\d+\.\d+\.\d+) port (?<Port>\d+)"
| table _time Username Port
```

Ultimi 30 giorni

✓ **316 eventi** (24/11/24 00:00:00,000 - 24/12/24 18:43:26,000) — Nessun campionamento degli eventi — Processo — Modalità intelligente

Eventi | Pattern | **Statistiche (316)** | Visualizzazione

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

_time	Username	Port
2024-12-09 12:42:32	administrator	2558
2024-12-09 12:42:32	appserver	4979
2024-12-09 12:42:32		
2024-12-08 12:42:32	shoutcast	2227
2024-12-08 12:42:32		
2024-12-08 12:42:32	oracle	2411
2024-12-08 12:42:32	noone	2381
2024-12-08 12:42:32	email	2213
2024-12-08 12:42:32	testuser	1622
2024-12-08 12:42:32	dean	1136
2024-12-08 12:42:32	desktop	4564
2024-12-08 12:42:32	mantis	1608
2024-12-08 12:42:32		
2024-12-08 12:42:32	operator	1782
2024-12-08 12:42:32		
2024-12-08 12:42:32	redmine	2565
2024-12-08 12:42:32	desktop	2905
2024-12-08 12:42:32	system	3771
2024-12-08 12:42:32	system	2123
2024-12-08 12:42:32	system	2969

Analizzando i log notiamo subito che sono **316** i tentativi di accesso falliti da parte dell'indirizzo IP **86.212.199.60**. Sono stati presi di mira 53 nomi utente diversi i cui più frequenti sono:

- **email** (14 tentativi, 6.25%)
- **system** (14 tentativi, 6.25%)
- **administrator** (12 tentativi, 5.357%)

È evidente che vengono provati sia nomi utente generici che legati a configurazioni di

sistema comuni, suggerendo un attacco di tipo brute-force.

Le porte utilizzate nei tentativi variano, con più di 100 valori. Le porte più comuni sono:
- **1173** (4 tentativi, 1.786%)
- **3771** (4 tentativi, 1.786%)

La varietà delle porte suggerisce che l'attaccante potrebbe cercare di accedere tramite servizi diversi o potrebbe usare tecniche per confondere i controlli di sicurezza.

In conclusione i dati indicano la presenza di un attacco brute-force sull'SSH, proveniente dall'IP 86.212.199.60, che potrebbe essere stato compromesso o potrebbe trattarsi di un server controllato dall'attaccante. Questi attacchi mirano a compromettere l'accesso tramite SSH sfruttando una combinazione di:

- Nomi utente generici o di default (es. guest, administrator).

- Varie porte di comunicazione, che potrebbero includere configurazioni non standard di SSH.

Azioni Raccomandate

1. **Bloccare l'IP:**
   Implementare una regola nel firewall per bloccare l'indirizzo 86.212.199.60.

2. **Rafforzare la configurazione SSH:**
   - Disabilitare l'accesso SSH per utenti sconosciuti o generici.
   - Cambiare la porta SSH predefinita (22) in una porta meno comune.
   - Abilitare l'autenticazione tramite chiave pubblica e disabilitare quella basata su password.

3. **Implementare un IDS/IPS:**
   Utilizzare un sistema di rilevamento/prevenzione delle intrusioni per identificare e bloccare attacchi simili.

4. **Monitorare e analizzare:**
   Continuare a monitorare i log per verificare se l'attaccante cambia tattica o se compaiono nuovi IP.

5. **Aggiungere limitazioni:**
   Utilizzare uno strumento come Fail2Ban per bloccare automaticamente gli IP che effettuano tentativi di accesso falliti multipli.

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.**

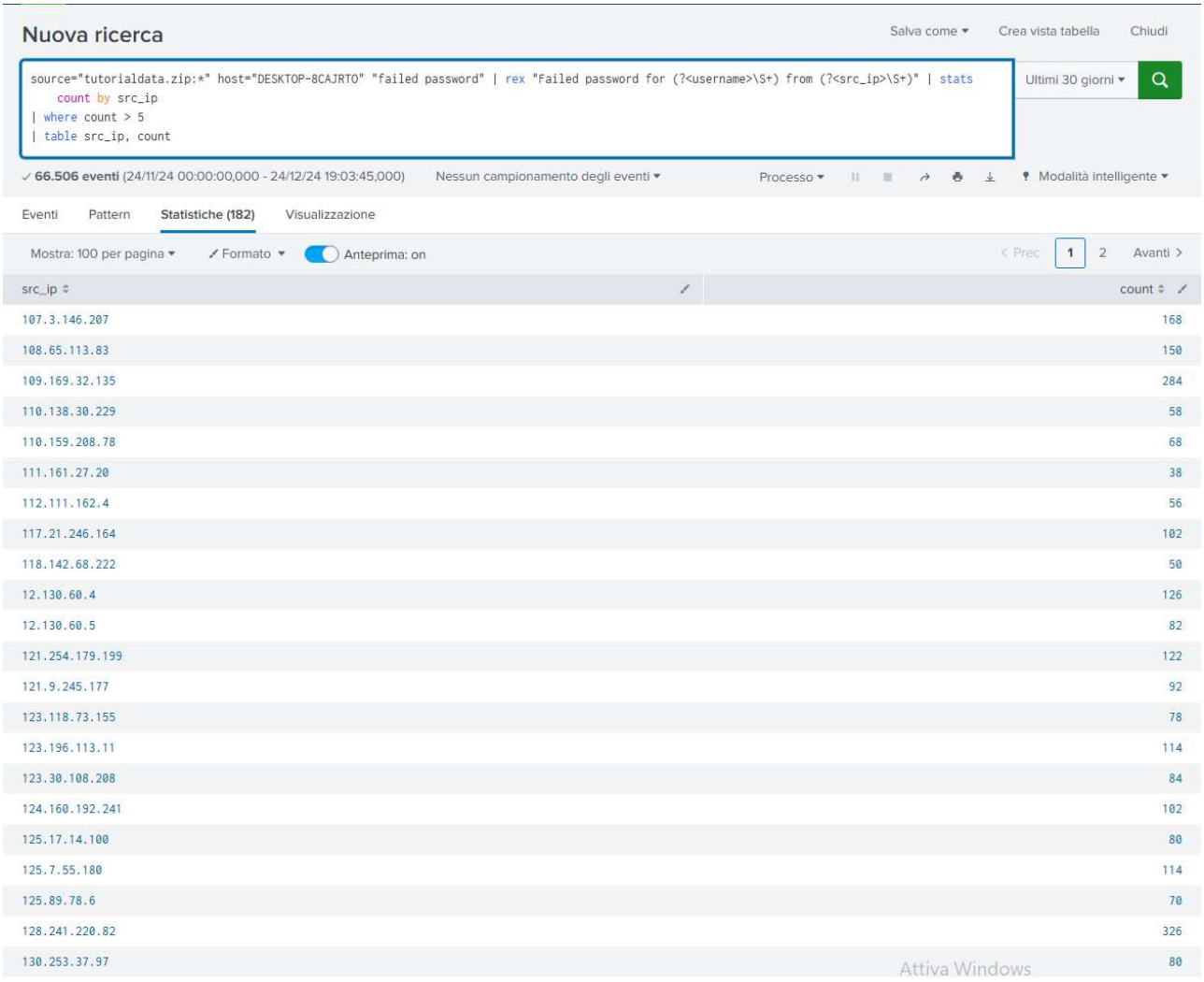
Query creata:

source="tutorialdata.zip:*" host="DESKTOP-8CAJRTO" "failed password" | rex "Failed password for (?<username>\S+) from (?<src_ip>\S+)" | stats count by src_ip
| where count > 5
| table src_ip, count

Per identificare gli indirizzi IP che hanno tentato di accedere al sistema più di 5 volte inseriamo:

- **| stats count by src_ip :** Conta il numero totale di tentativi falliti per ogni indirizzo IP sorgente

- **| where count > 5 :** Filtra i risultati per mostrare solo gli indirizzi IP che hanno effettuato più di 5 tentativi di accesso falliti.



Come possiamo vedere dall'immagine nella tabella ci vengono mostrati 182 indirizzi IP diversi con un numero variabile di tentativi falliti, che vanno da un minimo di circa 38 a un massimo di 326.
L'indirizzo IP 128.241.202.82 ha registrato il massimo numero di tentativi falliti (326), seguito da 109.169.32.135(284). Un numero così alto di tentativi falliti da un singolo indirizzo IP ci fa pensare che ci sia un attacco di tipo brute force, in più data la vasta gamma di questi indirizzi possiamo dedurre che gli attacchi potrebbero essere distribuiti globalmente.
È quindi possibile che gli indirizzi IP facciano parte di una botnet che tenta di accedere al sistema in modo automatizzato.

La mancanza di limitazioni sui tentativi di login (es. account lockout policies) permette a questi IP di eseguire numerosi tentativi falliti.

Azioni consigliate

1. **Bloccare gli IP sospetti**:
    - Configurare il firewall per bloccare immediatamente gli indirizzi IP che hanno effettuato più di 5 tentativi falliti (o secondo una soglia definita).
2. **Implementare meccanismi di sicurezza**:
    - **Account lockout**: Bloccare temporaneamente gli account dopo un certo numero di tentativi falliti.
    - **CAPTCHA**: Aggiungere un sistema CAPTCHA per distinguere utenti reali da bot.
    - **Autenticazione a due fattori (2FA)**: Migliora significativamente la sicurezza, anche in caso di brute force.
3. **Monitoraggio continuo**:
    - Configurare alert su Splunk per identificare in tempo reale IP con alto numero di tentativi falliti.
    - Verificare se questi IP sono noti per attività malevole, utilizzando servizi come AbuseIPDB o altre blacklist pubbliche.
4. **Indagine sugli IP**:
    - Effettuare una geolocalizzazione degli indirizzi IP per identificare possibili pattern (es. provenienza geografica simile).
    - Verificare se ci sono altre attività sospette associate a questi IP (es. scansioni di porte).

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

Per poter trovare tutti gli Internal Server Error dobbiamo filtrare per **status=500**, esso ci andrà a restituire tutti gli eventi con codice di stato HTTP 500 che indica che c'è un errore nel server.

Utilizziamo anche **table** per avere una visualizzazione dei log più chiara inserendo di seguito:

_time: Il timestamp dell'evento.

host: DESKTOP-8CAJRTO.

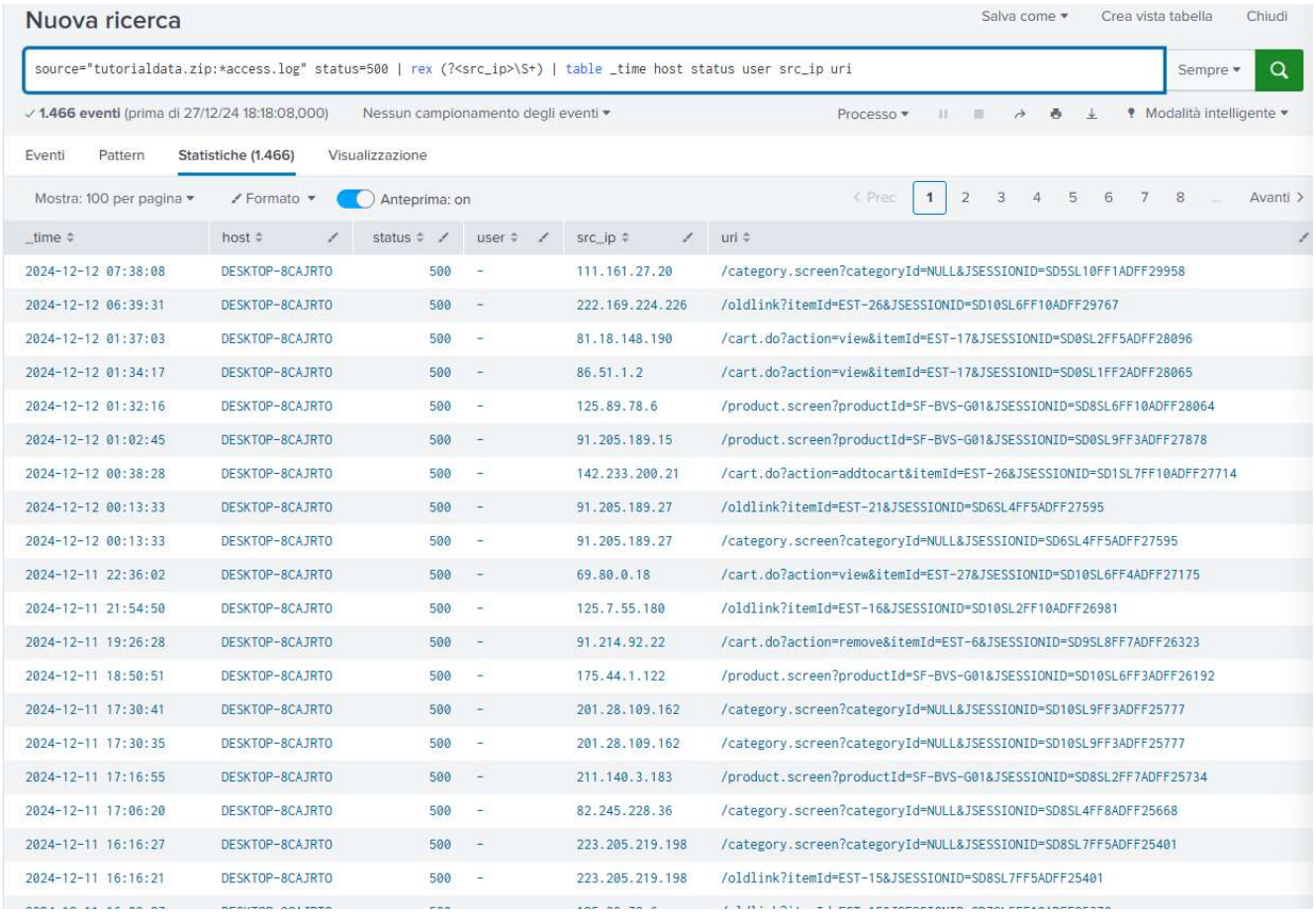
status: Il codice di stato HTTP cioè 500.

user: L'utente coinvolto, - .

src_ip: L'indirizzo IP sorgente della richiesta.

uri: L'URI richiesto, utile per identificare quale risorsa ha generato l'errore.

Query:
source="tutorialdata.zip:*access.log" status=500 | rex (?<src_ip>\S+) | table _time host status user src_ip uri



_time	host	status	user	src_ip	uri
2024-12-12 07:38:08	DESKTOP-8CAJRTO	500	-	111.161.27.20	/category.screen?categoryId=NULL&JSESSIONID=SD5SL10FF1ADFF29958
2024-12-12 06:39:31	DESKTOP-8CAJRTO	500	-	222.169.224.226	/oldlink?itemId=EST-26&JSESSIONID=SD10SL6FF10ADFF29767
2024-12-12 01:37:03	DESKTOP-8CAJRTO	500	-	81.18.148.190	/cart.do?action=view&itemId=EST-17&JSESSIONID=SD0SL2FF5ADFF28096
2024-12-12 01:34:17	DESKTOP-8CAJRTO	500	-	86.51.1.2	/cart.do?action=view&itemId=EST-17&JSESSIONID=SD0SL1FF2ADFF28065
2024-12-12 01:32:16	DESKTOP-8CAJRTO	500	-	125.89.78.6	/product.screen?productId=SF-BVS-G01&JSESSIONID=SD8SL6FF10ADFF28064
2024-12-12 01:02:45	DESKTOP-8CAJRTO	500	-	91.205.189.15	/product.screen?productId=SF-BVS-G01&JSESSIONID=SD0SL9FF3ADFF27878
2024-12-12 00:38:28	DESKTOP-8CAJRTO	500	-	142.233.200.21	/cart.do?action=addtocart&itemId=EST-26&JSESSIONID=SD1SL7FF10ADFF27714
2024-12-12 00:13:33	DESKTOP-8CAJRTO	500	-	91.205.189.27	/oldlink?itemId=EST-21&JSESSIONID=SD6SL4FF5ADFF27595
2024-12-12 00:13:33	DESKTOP-8CAJRTO	500	-	91.205.189.27	/category.screen?categoryId=NULL&JSESSIONID=SD6SL4FF5ADFF27595
2024-12-11 22:36:02	DESKTOP-8CAJRTO	500	-	69.80.0.18	/cart.do?action=view&itemId=EST-27&JSESSIONID=SD10SL6FF4ADFF27175
2024-12-11 21:54:50	DESKTOP-8CAJRTO	500	-	125.7.55.180	/oldlink?itemId=EST-16&JSESSIONID=SD10SL2FF10ADFF26981
2024-12-11 19:26:28	DESKTOP-8CAJRTO	500	-	91.214.92.22	/cart.do?action=remove&itemId=EST-6&JSESSIONID=SD9SL8FF7ADFF26323
2024-12-11 18:50:51	DESKTOP-8CAJRTO	500	-	175.44.1.122	/product.screen?productId=SF-BVS-G01&JSESSIONID=SD10SL6FF3ADFF26192
2024-12-11 17:30:41	DESKTOP-8CAJRTO	500	-	201.28.109.162	/category.screen?categoryId=NULL&JSESSIONID=SD10SL9FF3ADFF25777
2024-12-11 17:30:35	DESKTOP-8CAJRTO	500	-	201.28.109.162	/category.screen?categoryId=NULL&JSESSIONID=SD10SL9FF3ADFF25777
2024-12-11 17:16:55	DESKTOP-8CAJRTO	500	-	211.140.3.183	/product.screen?productId=SF-BVS-G01&JSESSIONID=SD8SL2FF7ADFF25734
2024-12-11 17:06:20	DESKTOP-8CAJRTO	500	-	82.245.228.36	/category.screen?categoryId=NULL&JSESSIONID=SD8SL4FF8ADFF25668
2024-12-11 16:16:27	DESKTOP-8CAJRTO	500	-	223.205.219.198	/category.screen?categoryId=NULL&JSESSIONID=SD8SL7FF5ADFF25401
2024-12-11 16:16:21	DESKTOP-8CAJRTO	500	-	223.205.219.198	/oldlink?itemId=EST-15&JSESSIONID=SD8SL7FF5ADFF25401

Un **Internal Server Error** (errore 500) è un codice di stato HTTP che indica che si è verificato un errore imprevisto sul server, impedendo il completamento della richiesta. Questo errore è generico e non fornisce dettagli specifici sul problema, ma indica che il server ha incontrato una condizione che non è riuscito a gestire.
Quando un utente riceve un errore 500, di solito non c'è molto che possa fare per risolvere il problema, poiché la causa è lato server. È compito dell'amministratore del server o del team di sviluppo diagnosticare e risolvere l'errore.

La nostra query ha trovato **1.466 eventi** con codice di stato HTTP 500, indicando una problematica frequente nei sistemi coinvolti, questi errori sembrano riguardare principalmente gli URI:

- **/cart.do?action=addtocart**
- **/product.screen?productId**

Questi URI sono legati a un sistema di gestione di carrelli e prodotti, probabilmente parte di un'applicazione di e-commerce. Come possiamo vedere dallo screen gli eventi mostrano richieste da indirizzi IP diversi, questo suggerisce che gli errori potrebbero essere scatenati da diversi utenti/clienti, indicando che il problema non è legato a un solo indirizzo IP.

Ipotesi sulle cause

Le cause di questi errori potrebbero derivare sia dall'interno che da attacchi esterni.

Cause interne:

1. **Problemi applicativi**:
   Gli errori HTTP 500 indicano problemi lato server. I log suggeriscono che il problema potrebbe riguardare:
   - **Operazioni sul carrello**: /cart.do?action=addtocart
   - **Richieste di visualizzazione prodotto**: /product.screen?productId

   Questi endpoint potrebbero avere bug nel codice o dipendenze non funzionanti correttamente.

2. **Carico del sistema**:
   Se il numero di richieste è molto alto in alcuni momenti, potrebbe esserci un problema di sovraccarico del server o della base dati.

3. **Malfunzionamento di componenti di backend**:
   Gli errori 500 potrebbero essere legati a problemi di connessione con il database, malfunzionamento di un'API esterna usata dal server o errori di configurazione.

Cause legate a un attacco:

Attacco di tipo DoS/DDoS:

Possiamo notare un grande numero di richieste concentrate in un breve periodo di tempo, si potrebbe trattare di un attacco di Denial of Service. Questo può sovraccaricare il server e causare errori 500.

Tentativi di exploit:

Gli attaccanti potrebbero inviare richieste malformate o payload specifici agli endpoint **/cart.do** e **/product.screen** per sfruttare vulnerabilità nel codice (ad esempio, **SQL injection**).

Attacco di enumerazione:

Gli IP nel log potrebbero indicare un tentativo di enumerazione degli ID prodotto (es. productId=...) o delle funzionalità del carrello, cercando di trovare falle nel sistema.

Credential stuffing o brute force:

Gli errori potrebbero derivare da tentativi ripetuti di autenticazione con credenziali non valide o manipolate.

Azioni preventive e correttive:

1. **Configurazione di rate-limiting**:
   - Blocca o limita le richieste frequenti dallo stesso IP.
2. **Validazione degli input**:
   - Assicurati che i parametri degli endpoint siano sempre controllati e filtrati.
3. **Monitoraggio attivo**:
   - Usa strumenti di monitoraggio per identificare picchi anomali di traffico.
4. **Firewall applicativo (WAF)**:
   - Implementa un firewall per bloccare richieste sospette o malformate.
5. **Analisi del codice degli endpoint**:
   - Rivedi il codice di /cart.do e /product.screen per identificare vulnerabilità o mancanze nella gestione degli errori.