Log4j Sourcetype Investigation - Splunk Enterprise

Date: 2025-08-01 **Environment**: Splunk Enterprise **Index**: `main` **Sourcetype**: `log4j`

🔍 Objective

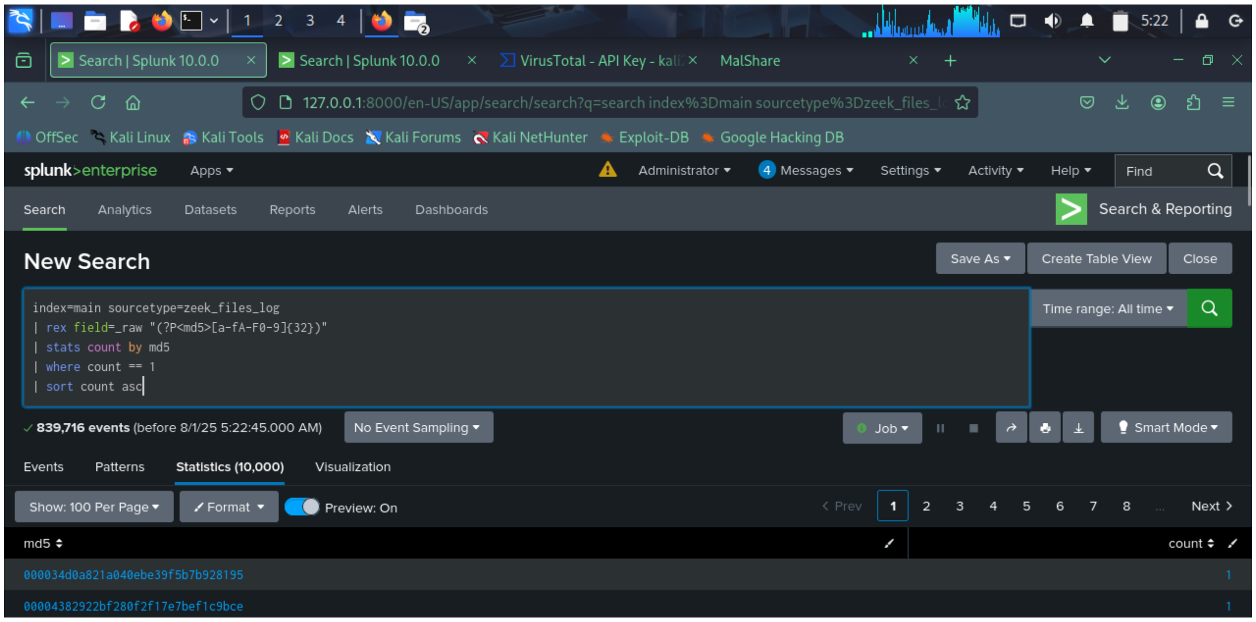
To investigate events under the `log4j` sourcetype in the `main` index, focusing on identifying the types of events recorded, extracting relevant fields, analyzing success vs failure results, and determining peak failure times for root cause analysis.

✅ Initial Exploration

- Identified `log4j` and `access_combined` as top sourcetypes in the `main` index.
- Chose to begin deep-dive investigation with `log4j`.
- Searched the log data with a base query:

  ```
  index=main sourcetype=log4j
  ```

- Observed that logs had entries with terms like `result="success"` or `result="failure"`.



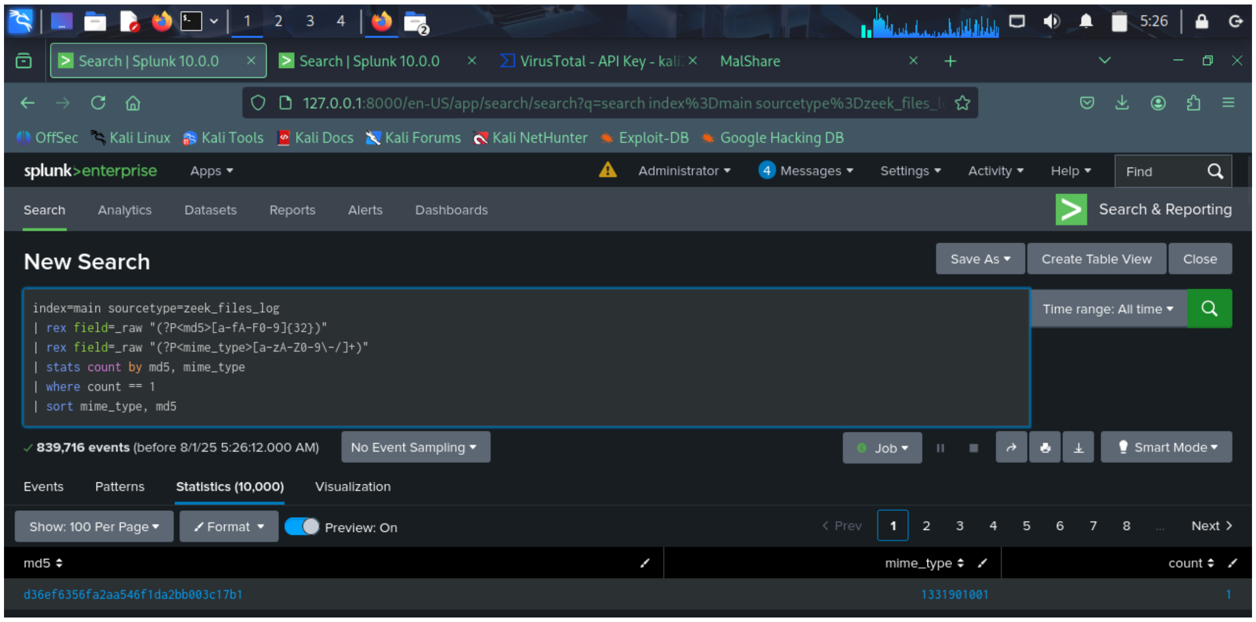
New Search

```
index=main sourcetype=zeek_files_log
| rex field=_raw "(?P<md5>[a-fA-F0-9]{32})"
| stats count by md5
| where count == 1
| sort count asc
```

✓ **839,716 events** (before 8/1/25 5:22:45.000 AM)

Statistics (10,000)

md5	count
000034d0a821a040ebe39f5b7b928195	1
00004382922bf280f2f17e7bef1c9bce	1



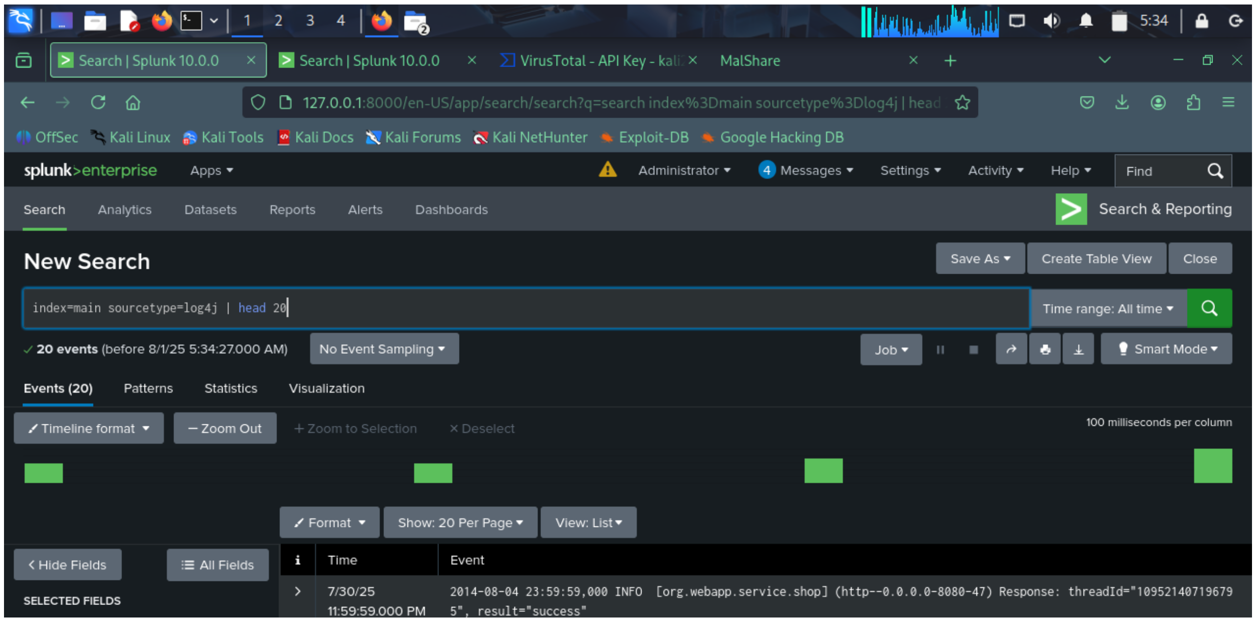
New Search

```
index=main sourcetype=zeek_files_log
| rex field=_raw "(?P<md5>[a-fA-F0-9]{32})"
| rex field=_raw "(?P<mime_type>[a-zA-Z0-9\-/]+)"
| stats count by md5, mime_type
| where count == 1
| sort mime_type, md5
```

✓ **839,716 events** (before 8/1/25 5:26:12.000 AM)

Statistics (10,000)

md5	mime_type	count
d36ef6356fa2aa546f1da2bb003c17b1	1331901001	1



127.0.0.1:8000/en-US/app/search/search?q=search index%3Dmain sourcetype%3Dlog4j | head

New Search

```
index=main sourcetype=log4j | head 20
```

Time range: All time

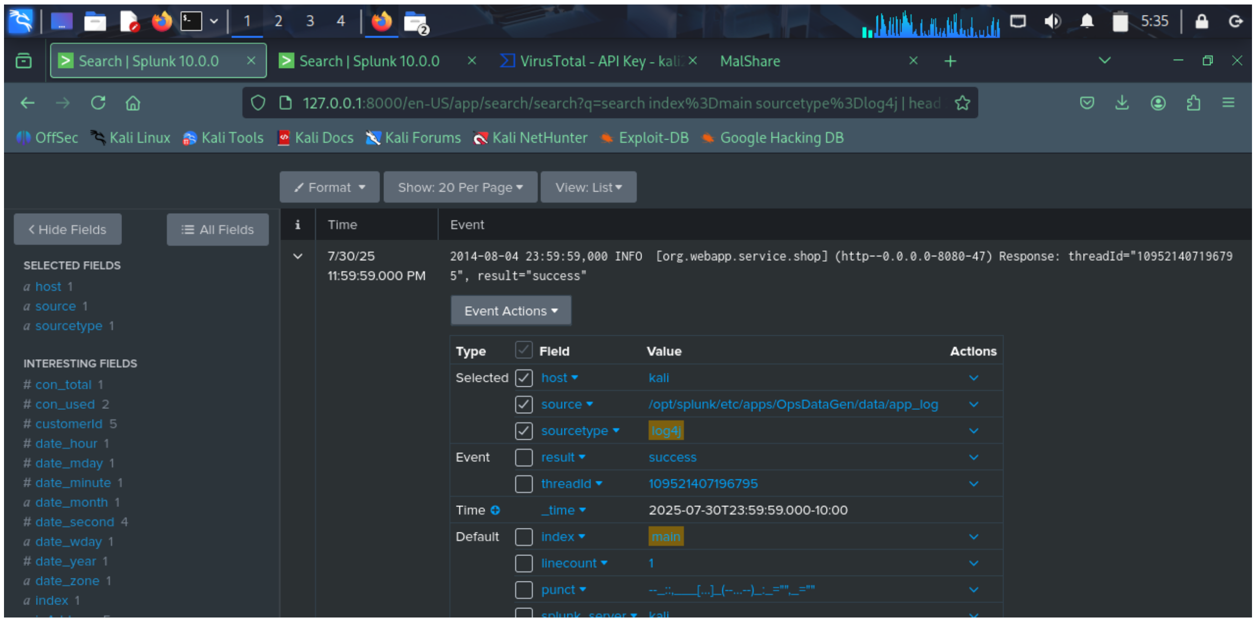
✓ **20 events** (before 8/1/25 5:34:27.000 AM) No Event Sampling

Events (20) | Patterns | Statistics | Visualization

100 milliseconds per column

Format | Show: 20 Per Page | View: List

i	Time	Event
>	7/30/25 11:59:59.000 PM	2014-08-04 23:59:59,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="109521407196795", result="success"



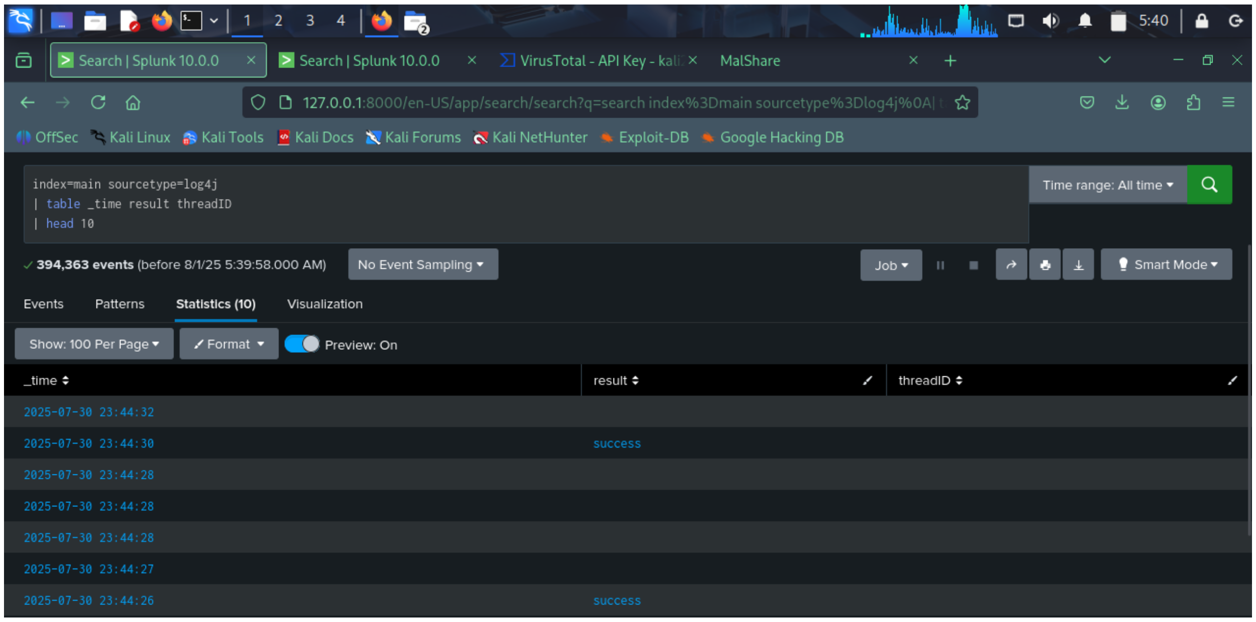
SELECTED FIELDS
- *a* host 1
- *a* source 1
- *a* sourcetype 1

INTERESTING FIELDS
- # con_total 1
- # con_used 2
- # customerId 5
- # date_hour 1
- # date_mday 1
- # date_minute 1
- *a* date_month 1
- # date_second 4
- *a* date_wday 1
- # date_year 1
- *a* date_zone 1
- *a* index 1

i	Time	Event
⌄	7/30/25 11:59:59.000 PM	2014-08-04 23:59:59,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="109521407196795", result="success"

Event Actions

Type		Field	Value	Actions
Selected	✓	host	kali	⌄
	✓	source	/opt/splunk/etc/apps/OpsDataGen/data/app_log	⌄
	✓	sourcetype	log4j	⌄
Event	☐	result	success	⌄
	☐	threadId	109521407196795	⌄
Time		_time	2025-07-30T23:59:59.000-10:00	
Default	☐	index	main	⌄
	☐	linecount	1	⌄
	☐	punct	--_::,____[...]_(--...--)_:_="",_=""	⌄
	☐	splunk_server	kali	⌄



```
index=main sourcetype=log4j
| table _time result threadID
| head 10
```

✓ 394,363 events (before 8/1/25 5:39:58.000 AM)

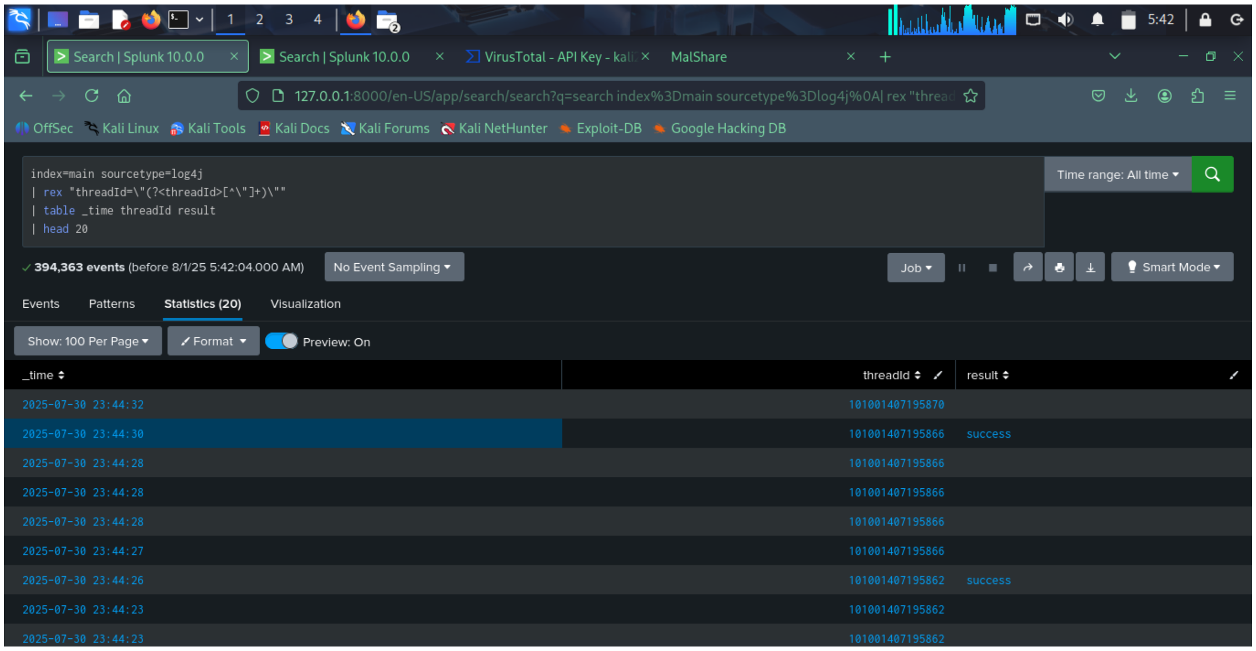
_time	result	threadID
2025-07-30 23:44:32		
2025-07-30 23:44:30	success	
2025-07-30 23:44:28		
2025-07-30 23:44:28		
2025-07-30 23:44:28		
2025-07-30 23:44:27		
2025-07-30 23:44:26	success	

🔢 Field Extraction

- Used `rex` to extract structured fields from the raw log events:

```
| rex "threadId=\"(?<threadId>[^\"]+)\".*?result=\"(?<result>[^\"]+)\""
| table _time, threadId, result
```

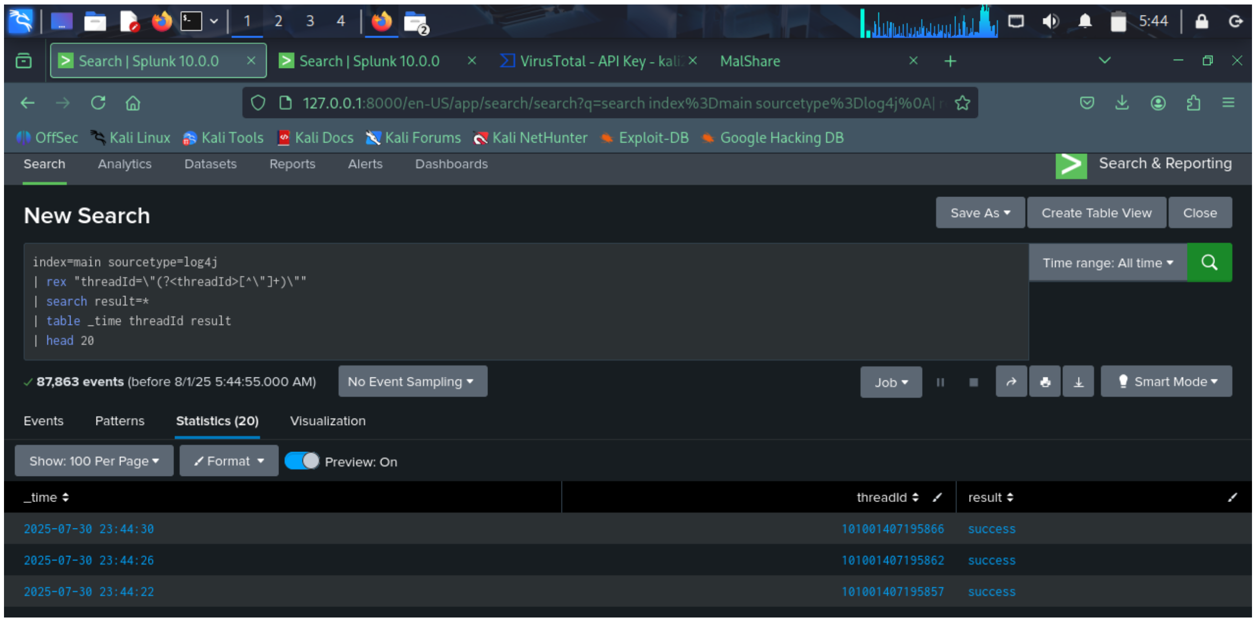
- Initially limited events to 20 to verify field extraction logic.
- Confirmed the appearance of both `success` and `failure` values in the `result` field.



```
index=main sourcetype=log4j
| rex "threadId=\"(?<threadId>[^\"]+)\""
| table _time threadId result
| head 20
```

✓ 394,363 events (before 8/1/25 5:42:04.000 AM)

_time	threadId	result
2025-07-30 23:44:32	101001407195870	
2025-07-30 23:44:30	101001407195866	success
2025-07-30 23:44:28	101001407195866	
2025-07-30 23:44:28	101001407195866	
2025-07-30 23:44:28	101001407195866	
2025-07-30 23:44:27	101001407195866	
2025-07-30 23:44:26	101001407195862	success
2025-07-30 23:44:23	101001407195862	
2025-07-30 23:44:23	101001407195862	

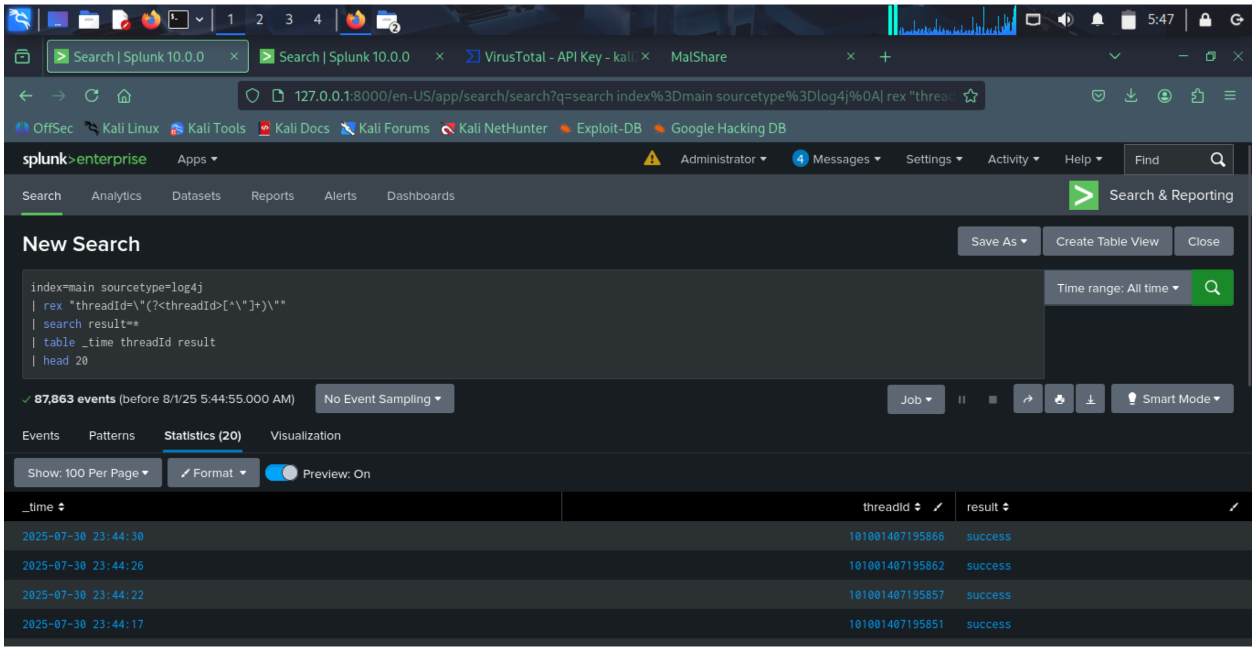


New Search

```
index=main sourcetype=log4j
| rex "threadId=\"(?<threadId>[^\"]+)\""
| search result=*
| table _time threadId result
| head 20
```

✓ 87,863 events (before 8/1/25 5:44:55.000 AM)

_time	threadId	result
2025-07-30 23:44:30	101001407195866	success
2025-07-30 23:44:26	101001407195862	success
2025-07-30 23:44:22	101001407195857	success

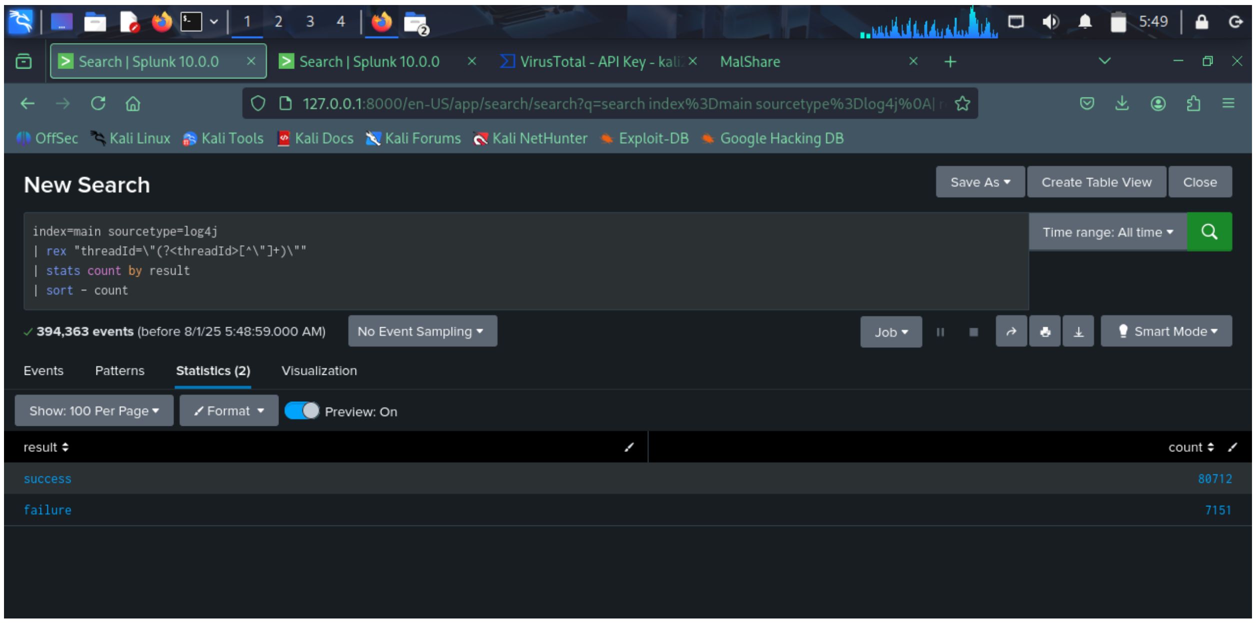


New Search

```
index=main sourcetype=log4j
| rex "threadId=\"(?<threadId>[^\"]+)\""
| search result=*
| table _time threadId result
| head 20
```

✓ **87,863 events** (before 8/1/25 5:44:55.000 AM)

_time	threadId	result
2025-07-30 23:44:30	101001407195866	success
2025-07-30 23:44:26	101001407195862	success
2025-07-30 23:44:22	101001407195857	success
2025-07-30 23:44:17	101001407195851	success

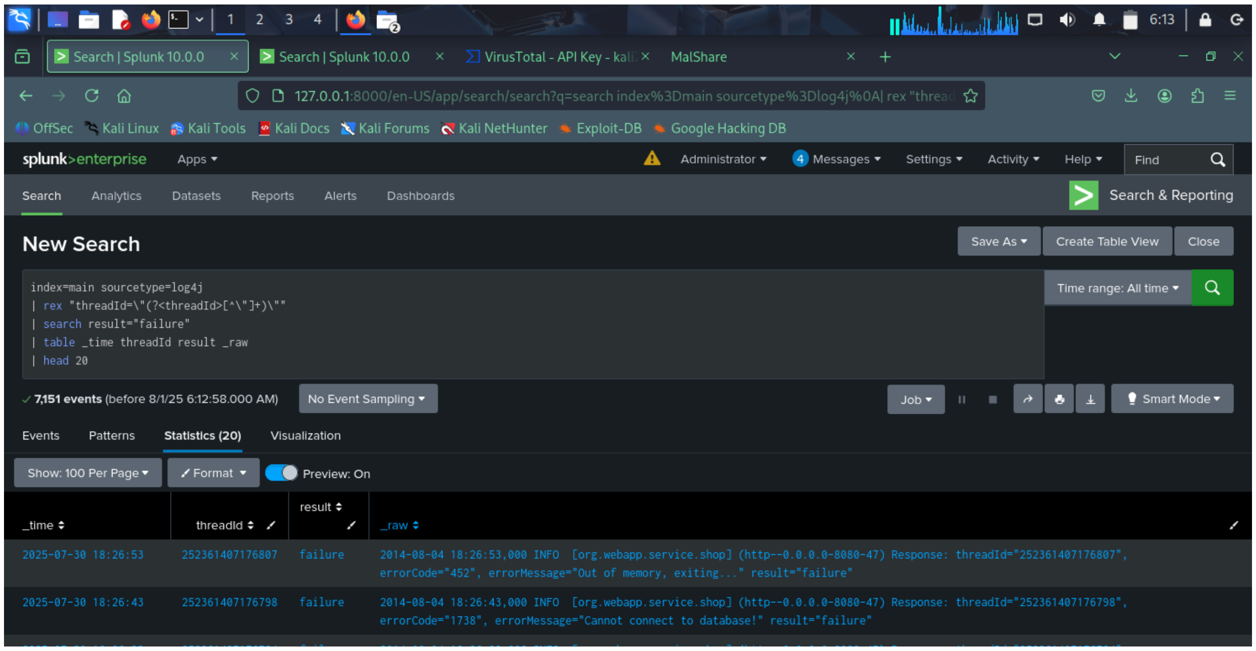


New Search

```
index=main sourcetype=log4j
| rex "threadId=\"(?<threadId>[^\"]+)\""
| stats count by result
| sort - count
```

✓ **394,363 events** (before 8/1/25 5:48:59.000 AM)

result	count
success	80712
failure	7151



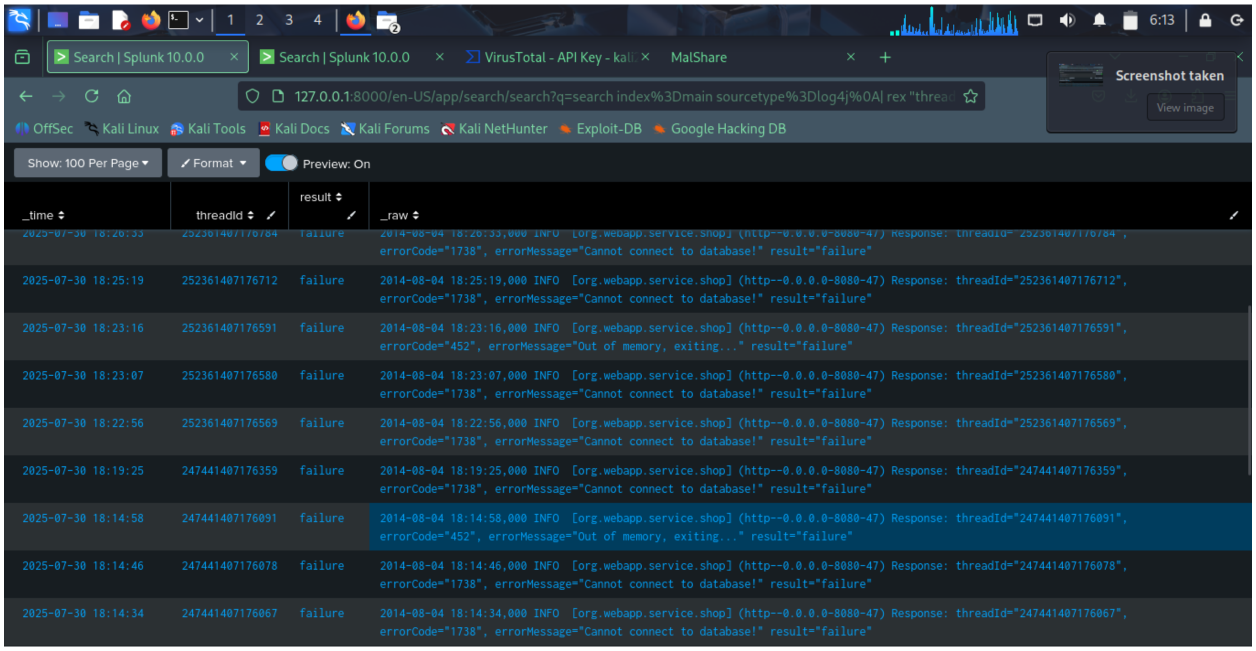
New Search

```
index=main sourcetype=log4j
| rex "threadId=\"(?<threadId>[^\"]+)\""
| search result="failure"
| table _time threadId result _raw
| head 20
```

✓ 7,151 events (before 8/1/25 6:12:58.000 AM)

Statistics (20)

_time	threadId	result	_raw
2025-07-30 18:26:53	252361407176807	failure	2014-08-04 18:26:53,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="252361407176807", errorCode="452", errorMessage="Out of memory, exiting..." result="failure"
2025-07-30 18:26:43	252361407176798	failure	2014-08-04 18:26:43,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="252361407176798", errorCode="1738", errorMessage="Cannot connect to database!" result="failure"



_time	threadId	result	_raw
2025-07-30 18:26:33	252361407176784	failure	2014-08-04 18:26:33,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="252361407176784", errorCode="1738", errorMessage="Cannot connect to database!" result="failure"
2025-07-30 18:25:19	252361407176712	failure	2014-08-04 18:25:19,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="252361407176712", errorCode="1738", errorMessage="Cannot connect to database!" result="failure"
2025-07-30 18:23:16	252361407176591	failure	2014-08-04 18:23:16,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="252361407176591", errorCode="452", errorMessage="Out of memory, exiting..." result="failure"
2025-07-30 18:23:07	252361407176580	failure	2014-08-04 18:23:07,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="252361407176580", errorCode="1738", errorMessage="Cannot connect to database!" result="failure"
2025-07-30 18:22:56	252361407176569	failure	2014-08-04 18:22:56,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="252361407176569", errorCode="1738", errorMessage="Cannot connect to database!" result="failure"
2025-07-30 18:19:25	247441407176359	failure	2014-08-04 18:19:25,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="247441407176359", errorCode="1738", errorMessage="Cannot connect to database!" result="failure"
2025-07-30 18:14:58	247441407176091	failure	2014-08-04 18:14:58,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="247441407176091", errorCode="452", errorMessage="Out of memory, exiting..." result="failure"
2025-07-30 18:14:46	247441407176078	failure	2014-08-04 18:14:46,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="247441407176078", errorCode="1738", errorMessage="Cannot connect to database!" result="failure"
2025-07-30 18:14:34	247441407176067	failure	2014-08-04 18:14:34,000 INFO [org.webapp.service.shop] (http--0.0.0.0-8080-47) Response: threadId="247441407176067", errorCode="1738", errorMessage="Cannot connect to database!" result="failure"

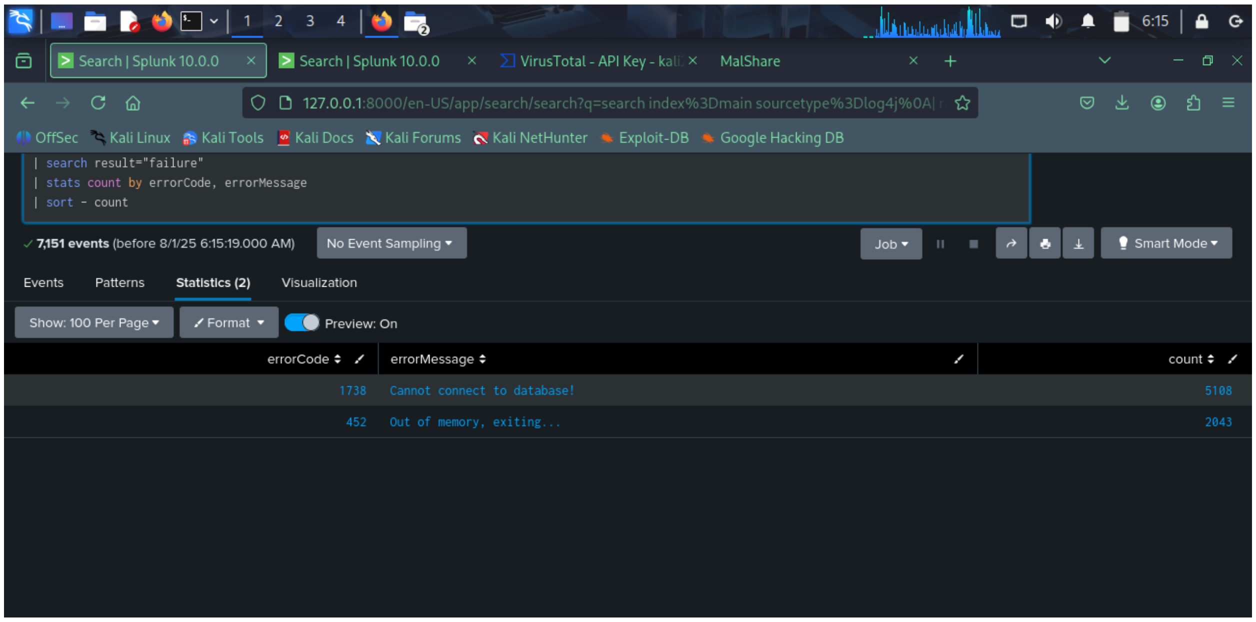
🔍 Filtering Failures

- Modified the query to only include failed events:

  ```
  index=main sourcetype=log4j result="failure"
  | rex "threadId=\"(?<threadId>[^\"]+)\".*?errorCode=\"(?<errorCode>[^\"
  ]+)\".*?errorMessage=\"(?<errorMessage>[^\"]+)\".*?result=\"(?<result>[
  ^\"]+)\""
  | table _time, threadId, errorCode, errorMessage, result
  ```

- Counted total failures:

- Error Code 1738 (Cannot connect to database): **5108 events**
- Error Code 452 (Out of memory, exiting): **2043 events**



```
| search result="failure"
| stats count by errorCode, errorMessage
| sort - count
```

7,151 events (before 8/1/25 6:15:19.000 AM)

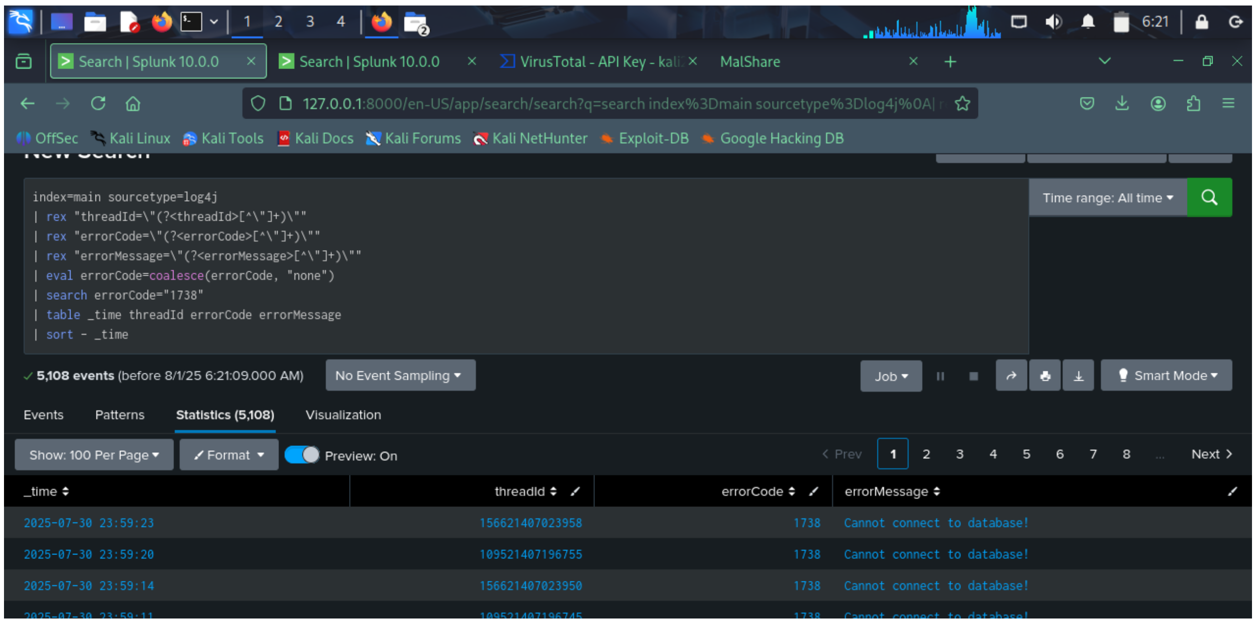
errorCode	errorMessage	count
1738	Cannot connect to database!	5108
452	Out of memory, exiting...	2043

Field Normalization using `coalesce`

- Used `eval` with `coalesce` to safely extract errorCode in case of field presence inconsistency:

  ```
  | eval itercode=coalesce(errorCode, error_code)
  ```

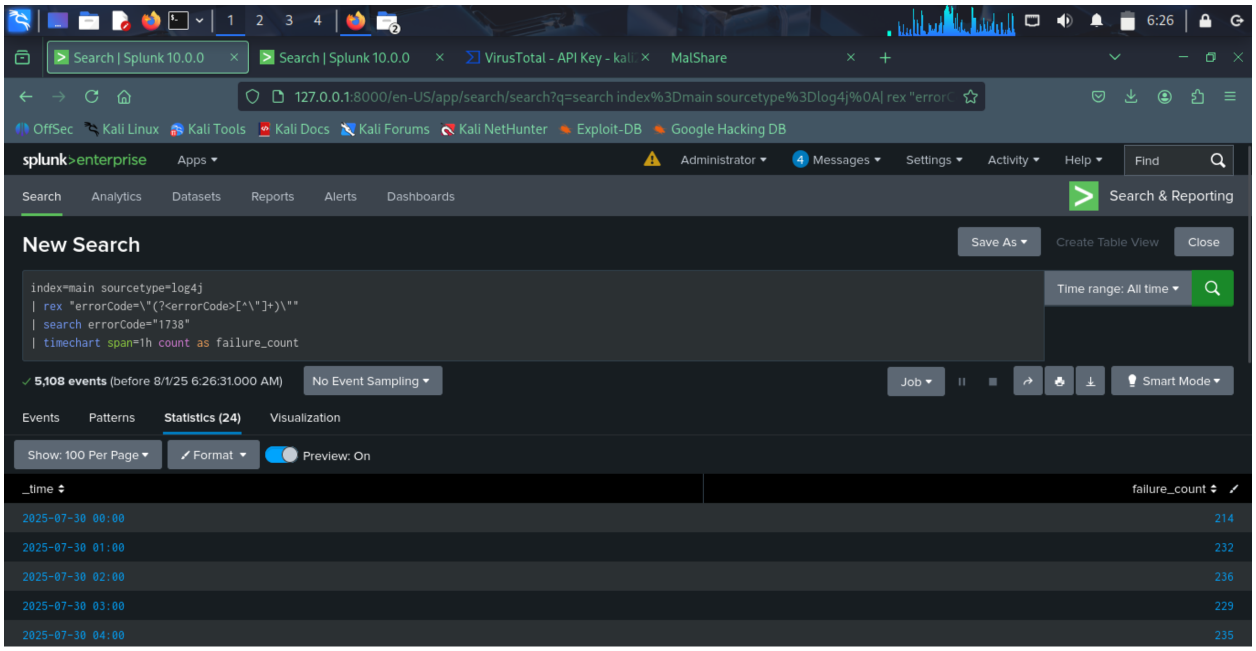
- Ensured that error code statistics aggregated reliably regardless of field name variation.



```
index=main sourcetype=log4j
| rex "threadId=\"(?<threadId>[^\"]+)\""
| rex "errorCode=\"(?<errorCode>[^\"]+)\""
| rex "errorMessage=\"(?<errorMessage>[^\"]+)\""
| eval errorCode=coalesce(errorCode, "none")
| search errorCode="1738"
| table _time threadId errorCode errorMessage
| sort - _time
```

✓ **5,108 events** (before 8/1/25 6:21:09.000 AM)

_time	threadId	errorCode	errorMessage
2025-07-30 23:59:23	156621407023958	1738	Cannot connect to database!
2025-07-30 23:59:20	109521407196755	1738	Cannot connect to database!
2025-07-30 23:59:14	156621407023950	1738	Cannot connect to database!
2025-07-30 23:59:11	109521407196745	1738	Cannot connect to database!

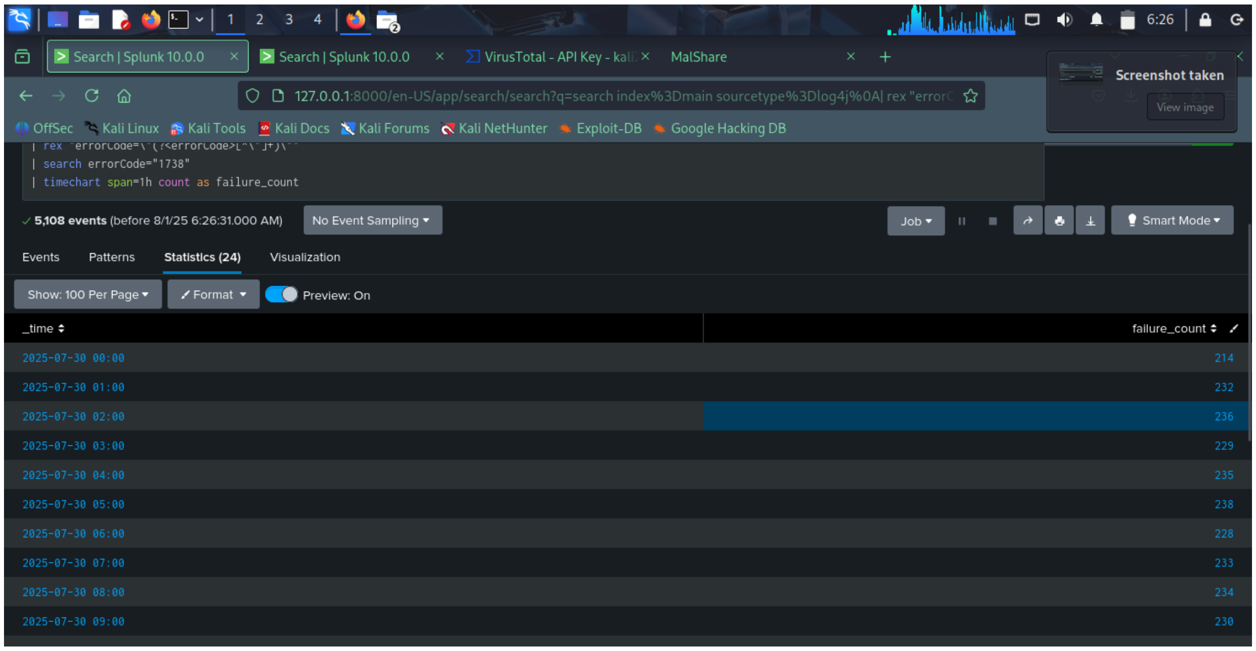


New Search

```
index=main sourcetype=log4j
| rex "errorCode=\"(?<errorCode>[^\"]+)\""
| search errorCode="1738"
| timechart span=1h count as failure_count
```

✓ **5,108 events** (before 8/1/25 6:26:31.000 AM)

_time	failure_count
2025-07-30 00:00	214
2025-07-30 01:00	232
2025-07-30 02:00	236
2025-07-30 03:00	229
2025-07-30 04:00	235



```
| rex "errorCode=\"(?<errorCode>[^\"]+)\""
| search errorCode="1738"
| timechart span=1h count as failure_count
```

✓ **5,108 events** (before 8/1/25 6:26:31.000 AM)

_time	failure_count
2025-07-30 00:00	214
2025-07-30 01:00	232
2025-07-30 02:00	236
2025-07-30 03:00	229
2025-07-30 04:00	235
2025-07-30 05:00	238
2025-07-30 06:00	228
2025-07-30 07:00	233
2025-07-30 08:00	234
2025-07-30 09:00	230

📊 Time-Based Analysis

- Grouped errorCode=1738 failures by hour:

```
index=main sourcetype=log4j errorCode="1738"
| bucket span=1h _time
| stats count as failure_count by _time
```

- Identified **05:00–05:59** as the peak failure hour (highest failure count: **238**).

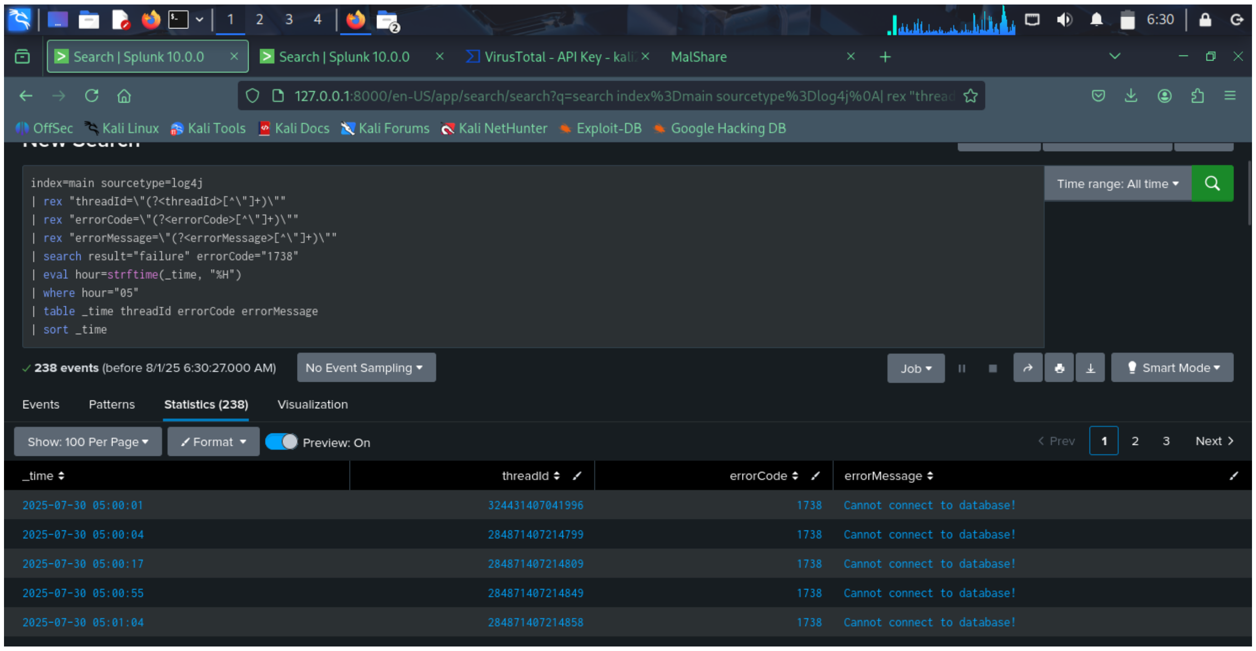
📈 Deep Dive into Peak Hour

- Queried full events for 05:00–05:59 with errorCode 1738:

```
index=main sourcetype=log4j result="failure" errorCode="1738" earliest=
"2025-07-30T05:00:00" latest="2025-07-30T05:59:59"
| rex "threadId=\"(?<threadId>[^\"]+)\".*?errorCode=\"(?<errorCode>[^\"
]+)\".*?errorMessage=\"(?<errorMessage>[^\"]+)\".*?result=\"(?<result>[
^\"]+)\""
| table _time, threadId, errorCode, errorMessage, result
```

- Verified error messages matched expected: `Cannot connect to database!`

- Analyzed frequency and thread diversity.



```
index=main sourcetype=log4j
| rex "threadId=\"(?<threadId>[^\"]+)\""
| rex "errorCode=\"(?<errorCode>[^\"]+)\""
| rex "errorMessage=\"(?<errorMessage>[^\"]+)\""
| search result="failure" errorCode="1738"
| eval hour=strftime(_time, "%H")
| where hour="05"
| table _time threadId errorCode errorMessage
| sort _time
```

✓ 238 events (before 8/1/25 6:30:27.000 AM)

_time	threadId	errorCode	errorMessage
2025-07-30 05:00:01	324431407041996	1738	Cannot connect to database!
2025-07-30 05:00:04	284871407214799	1738	Cannot connect to database!
2025-07-30 05:00:17	284871407214809	1738	Cannot connect to database!
2025-07-30 05:00:55	284871407214849	1738	Cannot connect to database!
2025-07-30 05:01:04	284871407214858	1738	Cannot connect to database!

✉️ Summary

A complete investigation was carried out to identify and analyze `failure` events in the `log4j` sourcetype. I successfully isolated the most common failure condition (`errorCode=1738`) and determined its peak hour for occurrence. This prepared me for a more targeted root cause analysis moving forward.