

# Forward-Looking Statements



This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at [www.investors.splunk.com](http://www.investors.splunk.com) or the SEC's website at [www.sec.gov](http://www.sec.gov). The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. We undertake no obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

Splunk, Splunk> and Turn Data Into Doing are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names or trademarks belong to their respective owners. © 2023 Splunk Inc. All rights reserved.

# I Am Speed!

Searching on Your Own TERMS  
With Simple Techniques That  
99% Aren't Using!

PLA1258C

## Jerrod Anderson

Cyber Intel Analyst | Lockheed Martin

## Chad Anderson

Manager, Security Intelligence Center | Reddit





## Jerrod Anderson

Cyber Intel Analyst | LM CIRT Insider Threat



## Chad Anderson

Manager | Reddit Security Intelligence Center

# Example 1

Windows Security Event Logs

## Search 1

Slow  890 Seconds

```
| search index=windows_sec user="user1" EventCode=4624
```

```
scanning 240,233,347 events in 889.208 seconds
```

## Search 2

Fast  12 Seconds

```
| search index=windows_sec user="user1" "user1" EventCode=4624
```

```
scanning 1,043 events in 12.62 seconds
```



# Example 2

Proxy Logs

## Search 1

Slow  144 Seconds

```
| search index=proxy q=* www.website.com*
```

scanning **22,437,072** events in **144.139** seconds

## Search 2

Fast  37 Seconds

```
| search index=proxy TERM(q=*) TERM(www.website.com*)
```

scanning **2,896,022** events in **37.635** seconds



## Jerrod Anderson

Cyber Intel Analyst | LM CIRT Insider Threat



## Chad Anderson

Manager | Reddit Security Intelligence Center

# Example 3

## Data Ingest Metrics

### Search 1

Slow  21 Seconds

```
| search index=_internal source=*license_usage.log  
| timechart limit=0 sum(b) by st | transpose 0
```

returned **80** results by scanning **2,621,263** events in **21.113** seconds

### Search 2

Fast  1 Second

```
| tstats sum(PREFIX(b=)) as Bytes WHERE index=_internal  
source=*/license_usage.log by PREFIX(st=) _time span=1d  
| rename st= as SourceType | xyseries SourceType _time Bytes
```

returned **27** results by scanning **2,621,263** events in **1.299** seconds



# Culmination of 10 Years of .conf

Kellen Green, Clara Merriman, Richard Morgan, Martin Müller, David Veuve, Brian Wooden, Simeon Yep  
And many more!

**Clara-Fication: More Tstats for Your Buckets**  
 TRU1133B  
 Clara Merriman  
 Senior Splunk Engineer | Splunk  
 Martin Müller  
 Principal Consultant | Consultant  
 splunk> .conf21  
<https://conf.splunk.com/files/2021/slides/TRU1133B.pdf>

**Behind the Magnifying Glass: How Search Works**  
 Jeff Champagne  
 Staff Architect, Splunk  
 .conf2016  
<https://conf.splunk.com/files/2016/slides/behind-the-magnifying-glass-how-search-works.pdf>

Splunk® Enterprise  
**Knowledge Manager Manual**  
 Download manual as PDF  
<https://docs.splunk.com/Documentation/Splunk>

(12) **United States Patent**  
 Baum et al.  
 (54) **TIME SERIES SEARCH WITH INTERPOLATED TIME STAMP**  
<https://image-ppubs.uspto.gov/dirsearch-public/print/downloadPdf/9002854>

**inurl:conf.splunk.com TSTATS TERM**

**Optimizing Data Analysis with a Semi-structured Time Series Database**  
 Lediton Bitincka, Archana Ganapathi, Stephen Serkin and Steve Zhang  
 Splunk Inc.  
 Abstract  
 Most modern systems generate abundant and diverse log data. With increasing storage costs, there are fewer reasons to store or discard data. However, the lack of tools to efficiently store and cross-combine heterogeneous datasets makes it tedious to mine the data for analytic insights. In this paper, we present Splunk, a semi-structured time series database that can be used to index, search and analyze massive heterogeneous datasets. We share observations, lessons and case studies from real world datasets, and demonstrate Splunk's power and flexibility for enabling insightful data mining searches.  
 Figure 1: Overview of the Splunk platform  
[https://www.usenix.org/legacy/event/slam10/tech/full\\_papers/Bitincka.pdf](https://www.usenix.org/legacy/event/slam10/tech/full_papers/Bitincka.pdf)

**The Power Of Data Normalization: A Look At The Common Information Model**  
 Mark Sacks, CISSP  
 Staff Sales Engineer, Splunk  
 Vladimir Skoryk, CISSP, CCFE, CFI, CISA, CISM, RGTT  
 Senior Professional Services Consultant, Splunk  
 .conf2016  
<https://conf.splunk.com/files/2016/slides/the-power-of-data-normalization-a-look-at-cim-under-the-hood.pdf>

**Fields, Indexed Tokens, and You**  
 PLA1466B  
 Martin Müller  
 Principal Consultant | Consultant  
 .conf22  
<https://conf.splunk.com/files/2022/slides/PLA1466B.pdf>

splunk> .conf2017  
**Revealing the Magic**  
 The Lifecycle of a Splunk Search  
 Kellen Green | Senior Software Engineer  
 September 27th, 2017 | Washington, DC  
<https://conf.splunk.com/files/2017/slides/revealing-the-magic-the-life-cycle-of-a-splunk-search.pdf>

**INGEST\_EVAL and CLONE\_SOURCETYPE**  
 Advanced pipeline configurations  
 Richard Morgan  
 Vladimir Skoryk  
 Splunk  
<https://conf.splunk.com/files/2020/slides/PLA1154C.pdf>

splunk> .conf2017  
**Splunk Performance**  
 Observations and Recommendations  
 Simeon Yep | AVP, GSA  
 Brian Wooden | GSA Partner Integrations  
 2017-09-27 | Washington, DC  
<https://conf.splunk.com/files/2017/slides/observations-and-recommendations-on-splunk-performance.pdf>

splunk> .conf2017  
**Security Ninjutsu Part Four**  
 The SPLening  
 2.5 hours of EPIC SPL stuffed into 45 minutes  
 David Veuve | Principal Security Strategist  
 September 2017 | Washington, DC  
<https://www.davidveuve.com/presentations.html>

**Clara-fication: Job Inspector**  
 Clara-fy your jobs  
 Clara Merriman  
 Senior Splunk Engineer | Splunk  
 Martin Müller  
 Principal Consultant | Consultant Software Solutions  
<https://conf.splunk.com/files/2020/slides/TRU1143C.pdf>

splunk> .conf2017  
**How splunkd works**  
 splunkd: Pipelines, Processors, Queues  
 Inputs: File, Network, Script, HEC, S2S, ...  
 Debugging: Metrics, Monitoring Console  
 by Amrit Bath, Abhinav Nekkanti  
<https://conf.splunk.com/files/2017/slides/how-splunkd-works.pdf>

splunk> .conf2017  
**Searching FAST**  
 How to Start Using Istats and Other Acceleration Techniques  
 David Veuve | Principal Security Strategist  
 September 2017 | Washington, DC  
<http://conf.splunk.com/files/2017/slides/searching-fast-how-to-start-using-istats-and-other-acceleration-techniques.pdf>

**TSTATS and PREFIX**  
 How to get the most out of your lexicon, with walklex, tstats, indexed fields, PREFIX, TERM and CASE  
 Richard Morgan  
 Principal Architect | Splunk  
<https://conf.splunk.com/files/2020/slides/PLA1089C.pdf>





# Why is That Faster?

## Data In

## Data Out

## Even Faster

### Data In - Write the Events to Disk

- Inputs
- Parsing
- Merging
- Typing
- Indexing

... / index / db / db\_time\_time\_#

```

Props.conf:
- index_extractions = JSON|W3C|CVT|P|V|HEC
- trace() SEGMENTATION + msg_size
Transforms.conf (fieldname extraction):
- index_time, host, source, sourcetype
- ingest_eval [ regex | sed | cml | clone_sourcetype
- write_metadata ...meta fields are indexed]
Fields.conf
- [Field_Name] INDEXED=true
  
```

TSIDX index files are updated + host, source, sourcetype  
\_raw is compressed into Slices & saved to Journal

when Hot rolls to Warm, create a Bloom Filter

```

haah_1("foo=0") == 0
haah_2("foo=0") == 7
  
```

splunk> .conf23

### Data Out

- Index + Time
- Meta + Bloom
- LISPY / Index
- Raw Data
- Schema on Fly
- Process SPL
- Search Head

... / index / db / db\_time\_time\_#

I thought this session was about SPEED!

In a lake of events, how quickly can you eliminate Buckets?

splunk> .conf23

### Field Storage Structure Trade-Offs

| Indexing Fields   | Summary/Metrics Index  | Data Models  |
|---|--|--|
| <b>Pros</b> <ul style="list-style-type: none"> <li>Smallest Space Option*</li> </ul>  | <b>Pros</b> <ul style="list-style-type: none"> <li>Familiar</li> <li>Long Summarization Range</li> <li>Easy to Add Fields or Sources</li> <li>Good for Aggregation/Archival</li> </ul> | <b>Pros</b> <ul style="list-style-type: none"> <li>Helps enforce the CIM</li> <li>Great for Acceleration over Short Time Periods</li> <li>Pointers to Original Events</li> </ul> |
| <b>Considerations</b> <ul style="list-style-type: none"> <li>Done at index time</li> <li>Could cause index bloat</li> </ul> | <b>Considerations</b> <ul style="list-style-type: none"> <li>Need Indexed Fields?</li> </ul>   | <b>Considerations</b> <ul style="list-style-type: none"> <li>Additional Learning Curve</li> </ul>  |

```

eval IndexedField="IndexedField:",".value
collect index=MySummary output_format=hec
collect index=MySummary output_format=raw
  
```

<https://docs.splunk.com/Documentation/SplunkCloud/latest/SearchReference/collect>

splunk> .conf23

### Indexing Fields

```

05-25-2023 21:44:26.100 INFO
st="auth:sudo" idx="ec2" b=6516
  
```

meta: index = \_internal

source = license\_usage.log

sourcetype = splunkd host = idx1.cl

CustomField = "Case Matters"

### Walklex

to Investigate the Index

```

| walklex index=aws_prod type=field
| stats av distinct_values as NumValues by
  
```

✓ 1,117 events (4/27/23 7:00:00.000 PM to 5/27/23 1

No Event Sampling

100 Per Page | Format | Preview

field ↓ NumV

```

arn:aws:s3
arn:aws:sts
arn:aws:iam
  
```

### TERM( )

It's not magic - Searching Breaks

IPs: 192.0.0.1

Emails: email@domain.com

Domains: host.domain.tld

Usernames: domain\user

Field??: b=6516

### LISPY

LISPY IS KEWL, Where do I See That?

Expanded index

```

05-29-2023 04:11:54.059 INFO
UnifiedSearch [32054 searchOrchestrator]
base lispy: [ AND [ OR 6516 b:=6516 ] ]
05-29-2023 04:11:54.060 INFO
  
```

### Understanding I

What is the LISPY Doing?

```

| search user=user1
[ OR user1 sourcetype::crowdstrike:events:ser
sourcetype::gws:reports:admin sourcetype::gws:
sourcetype::gws:reports:context_aware_access
sourcetype::gws:reports:drive sourcetype::gws:
sourcetype::gws:reports:groups_enterprise
sourcetype::gws:reports:login sourcetype::gws:
sourcetype::gws:reports:token sourcetype::okta
user::user1 [ AND sourcetype::ipassword:insigh
user:uid:user1 ] [ AND
sourcetype::ipassword:insights:signin_attempts
target_user:uid:user1 ] [ AND profile.name::
sourcetype::okta:group ] [ AND profile.logi
sourcetype::okta:group ] [ AND
sourcetype::duo:authentication user.name:user
message_info.source.from_header.displayName:cu
sourcetype::gws:gmail ] [ AND entities.users()
sourcetype::uba_threat_json ] [ AND sourcetype
OR uid:user1 user_id:user1 ] ]
  
```

### Tstats

Use the Index, Just the Index

Search 1  
Slow ⌚ 120 Seconds

```

| search index=window
returned 44,832 results by sca
  
```

Search 2  
Fast ⌚ 5 Seconds

```

| tstats count where
returned 44,800 results
  
```

### Data Format Matter

Data Models don't store raw data! Index Only

Indexes are a data, but fields added

Adding Many Fields (Difference between)

# Data In

- 1. Inputs
- 2. Parsing
- 3. Merging
- 4. Typing
- 5. Indexing

... / index / db /  
db\_time\_time\_#



## Props.conf:

- index\_extractions = JSON|W3C|CSV|TSV|PSV|HEC
- [<spec>] SEGMENTATION = <seg\_rule>

## Transforms.conf (field/term extraction):

- index, \_time, host, source, sourcetype
- ingest\_eval | regex | sed\_cmd | clone\_sourcetype
- write\_meta=true    \_meta fields are indexed!

## Fields.conf

- [Field\_Name] INDEXED=true

**Time Series Index Files (TSIDX) are being created**

```
05-25-2023 21:44:26.100 INFO
st="auth:sudo" idx="ec2" b=6516
```



# Indexing Fields

```
05-25-2023 21:44:26.100 INFO
```

```
st="auth:sudo" idx="ec2" b=6516
```

```
meta: index = _internal
```

```
source = license_usage.log
```

```
sourcetype = splunkd host = idx1.cloud.com
```

```
CustomField = "Case Matters"
```

```
EarliestEventTime_LatestEventTime.tsidx
```

```
CustomField::Case Matters  
customfield::case matters  
host::idx1.cloud.com  
source::license_usage.log  
sourcetype::splunkd
```

# Major Breakers

05-25-2023 21:44:26.100 INFO

st="auth:sudo" idx="ec2" b=6516

*meta*: index = \_internal

source = license\_usage.log

sourcetype = splunkd host = idx1.cloud.com

CustomField = "Case Matters"

EarliestEventTime\_LatestEventTime.tsidx

CustomField::Case Matters

customfield::case matters

host::idx1.cloud.com

source::license\_usage.log

sourcetype::splunkd

05-25-2023

21:44:26.100

auth:sudo

b=6516

ec2

idx=

info

st=



# Minor Breakers

05-25-2023 21:44:26.100 INFO

st="auth:sudo" idx="ec2" b=6516

meta: index = \_internal

source = license\_usage.log

sourcetype = splunkd host = idx1.cloud.com

CustomField = "Case Matters"

EarliestEventTime\_LatestEventTime.tsidx

CustomField::Case Matters  
 customfield::case matters  
 host::idx1.cloud.com  
 source::license\_usage.log  
 sourcetype::splunkd

05

05-25-2023

100

2023

21

21:44:26.100

25

26

44

6516

auth

auth:sudo

b

b=6516

ec2

idx

idx=

info

st

st=

sudo

# Back to Example 3

```
05-25-2023 21:44:26.100 INFO
```

```
st="auth:sudo" idx="ec2" b=6516
```

```
EarliestEventTime_LatestEventTime.tsidx
```

```
CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
source::license_usage.log
sourcetype::splunkd
```

```
05
05-25-2023
100
2023
21
```

```
| search index=_internal source=license_usage.log
| timechart limit=0 sum(b) by st
```

```
6516
auth
auth:sudo
b
b=6516
ec2
```

```
| tstats sum(PREFIX(b=)) as Bytes by PREFIX(st=)
```

```
st
st=
sudo
```

# Back to Example 3

05-25-2023 21:44:26.100 INFO

st="auth:sudo" idx="ec2" b=6516

EarliestEventTime\_LatestEventTime.tsidx

CustomField::Case Matters  
customfield::case matters  
host::idx1.cloud.com  
source::license\_usage.log  
sourcetype::splunkd

05  
05-25-2023  
100  
2023  
21

Major Breakers impact  
how you use the Index!

auth:sudo  
b  
b=6516  
ec2

| tstats sum(PREFIX(b=)) as Bytes by PREFIX(st=)

st  
st=  
sudo

# Walklex

to Investigate the Index

Splunk Enterprise  
**Knowledge Manager Manual**  
[Download manual as PDF](#)  
<https://docs.splunk.com/Documentation/SplunkCloud/latest/SearchReference/walklex>

```
| walklex index=aws_prod type=field  
| stats avg(distinct_values) as NumValues by field
```

✓ 1,117 events (5/27/23 7:00:00.000 PM to 5/27/23 11:30:07.000 PM)  
No Event Sampling ▾

Events Patterns **Statistics (22)** Visualization

100 Per Page ▾ Format Preview ▾

| field       | NumValues |
|-------------|-----------|
| arn:aws:s3  | 187,264   |
| arn:aws:sts | 3,285     |
| arn:aws:iam | 142       |

- Type:
- field
  - fieldvalue
  - term
  - all

Are these fields intentional?



# Walklex

to Investigate the Index Bloat

Pivot from **field** to:  
**type=fieldvalue**

Splunk® Enterprise  
**Knowledge Manager Manual**  
[Download manual as PDF](#)  
<https://docs.splunk.com/Documentation/SplunkCloud/latest/SearchReference/walklex>

```
| walklex index=aws_prod type=fieldvalue pattern=arn:aws:s3*
| stats avg(count) as NumEvents by term
```

**Pattern:**

- Filter the search
- Accepts Wildcards

✓ 100,528 events (5/27/23 11:34:00.000 PM to 5/27/23 11:54:47.000 PM)

No Events

**Multi-Colon Data can Bloat the Index!**

“url”=“www.web.com?  
 Ad:1234::5678::abcd”

Events

100 Per Page ▾ Format Preview ▾

| term   | NumEvents |
|--|-----------|
| arn:aws:s3:::reddit-uploaded-media           | 17,984    |
| arn:aws:s3:::reddit-hosted-video             | 13,823    |
| arn:aws:s3:::reddit-hosted-media             | 11,650    |
| arn:aws:s3:::reddit-subreddit-uploaded-media | 2,884     |

**Multi-Colon Data?**

- CloudTrail
- Proxy Logs
- etc



# Data In - Write the Events to Disk

1. Inputs

2. Parsing

3. Merging

4. Typing

5. Indexing

## Props.conf:

```
- index_extractions = JSON|W3C|CSV|TSV|PSV|HEC  
- [<spec>] SEGMENTATION = <seg_rule>
```

## Transforms.conf (field/term extraction):

```
- index, _time, host, source, sourcetype  
- ingest_eval | regex | sed_cmd | clone_sourcetype  
- write_meta=true  _meta fields are indexed!
```

## Fields.conf

```
- [Field_Name] INDEXED=true
```

**TSIDX index files are updated + host, source, sourcetype  
\_raw is compressed into Slices  within the Journal** 

# Data In - Create the Bloom Filter

- 1. Inputs
- 2. Parsing
- 3. Merging
- 4. Typing
- 5. Indexing

### Props.conf:

```
- index_extractions = JSON|W3C|CSV|TSV|PSV|HEC
- [<spec>] SEGMENTATION = <seg_rule>
```

### Transforms.conf (field/term extraction):

```
- index, _time, host, source, sourcetype
- ingest_eval | regex | sed_cmd | clone_sourcetype
- write_meta=true  _meta fields are indexed!
```

### Fields.conf

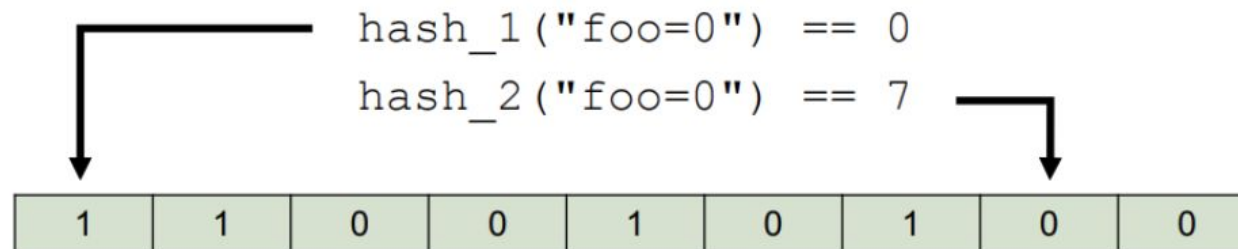
```
- [Field_Name] INDEXED=true
```

**TSIDX index files are updated + host, source, sourcetype**

**\_raw is compressed into Slices  within the Journal**



**when Bucket is made Warm, create the Bloom Filter**



**AUTOMATICALLY GENERATED**

splunk> .conf23



# Data In X 10000s of Buckets

- 1. Inputs
- 2. Parsing
- 3. Merging
- 4. Typing
- 5. Indexing

... / index / db / db\_time\_time\_#



### Props.conf:

- index\_extractions = JSON|W3C|CSV|TSV|PSV|HEC
- [<spec>] SEGMENTATION = <seg\_rule>

### Transforms.conf (field/term extraction):

- index, \_time, host, source, sourcetype
- ingest\_eval | regex | sed\_cmd | clone\_sourcetype
- write\_meta=true \_meta fields are indexed!

### Fields.conf

- [Field\_Name] INDEXED=true

**TSIDX index files are updated + host, source, sourcetype**

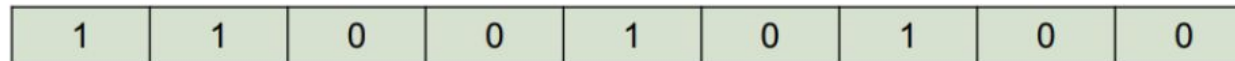
**\_raw is compressed into Slices**  **within the Journal**



**when Bucket is made Warm, create the Bloom Filter**

hash\_1 ("foo=0") == 0

hash\_2 ("foo=0") == 7



**AUTOMATICALLY GENERATED**

splunk> .conf23

# Why is That Faster?

## Data In

## Data Out

**Data In - Write the Events to Disk**

1. Inputs
2. Parsing
3. Merging
4. Typing
5. Indexing

... / index / db / db\_time\_time\_#

Write a Bloom Filter

1 1 0 0 0 1 0 0 0

**Data Out**

1. Index + Time
2. Meta + Bloom
3. LISPY / Index
4. Raw Data
5. Schema on Fly
6. Process SPL
7. Search Head

... / index / db / db\_time\_time\_#

I thought this session was about SPEED!

In a lake of events, how quickly can you eliminate Buckets?

### Indexing Fields

05-25-2023 21:44:26.100 INFO

st="auth:s... b=6516

meta

source =

sourcetype = splunkd host = idx1.cl

CustomField = "Case Matters"

### Walklex

to Investigate the Index

walklex index=aws prod type=field

stats as NumValues by

arn:aws:s3

arn:aws:sts

arn:aws:iam

### TERM()

It's not magic - Searching Breaker

User Search

192.0.0.1

Emails

email@domain.com

Domains

host.domain.tld

Usernames

domain\user

Field??

b=6516

### LISPY

LISPY is KEWL, Where do I See That?

Job

Clara-fication: Job Inspector

Learn more about troubleshooting empty search results at Splunk

(SID: 1685326569159:\*) search.log Job Details Dashboard

splunkcloud.com lispy

Expanded index

05-29-2023 04:11:54.059 INFO

UnifiedSearch [32054 searchOrchestrator]

base lispy: [ AND [ OR 6516 b:6516 ] ]

05-29-2023 04:11:54.060 INFO

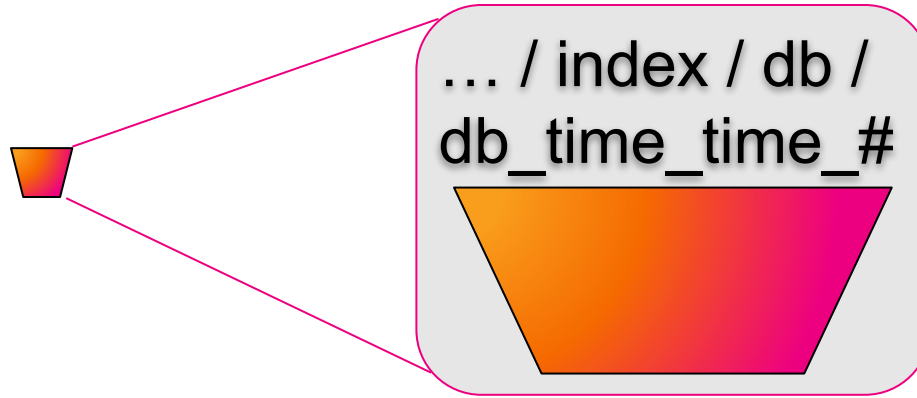
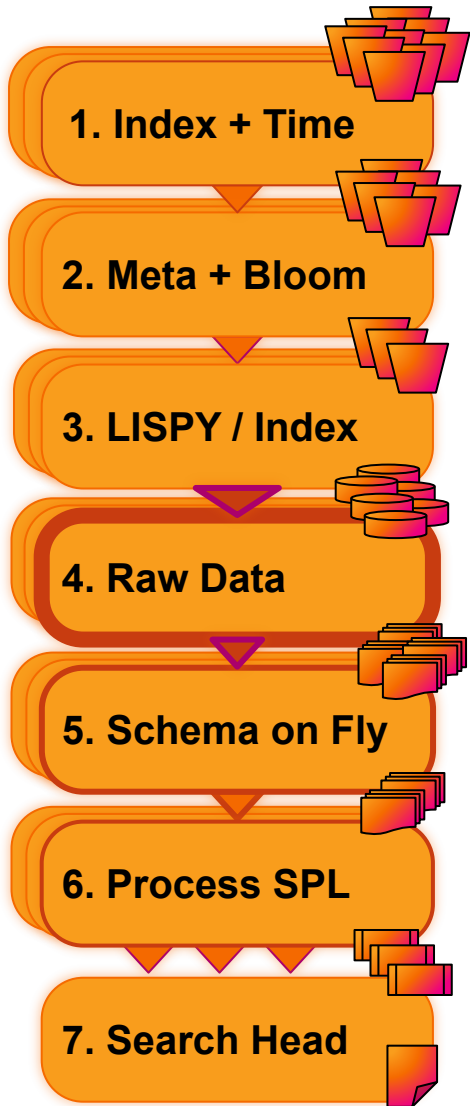
### Understanding LISPY

What is the LISPY Doing?

search user=user1

```
[ OR user1 sourcetype::crowdstrike:events:ser
sourcetype::crowdstrike:events:streams
sourcetype::gws:reports:admin sourcetype::gws:
sourcetype::gws:reports:context_aware_access
sourcetype::gws:reports:drive sourcetype::gws:
sourcetype::gws:reports:groups_enterprise
sourcetype::gws:reports:login sourcetype::gws:
sourcetype::gws:reports:token sourcetype::okta
user::user1 [ AND sourcetype::ipassword:insigh
user:uid:user1 ] [ AND
sourcetype::ipassword:insights:signin_attempts
target_user:uid:user1 ] [ AND profile.name::
sourcetype::okta:group ] [ AND profile.logi
sourcetype::okta:group ] [ AND
sourcetype::duo:authentication user.name:user
message_info.source.from_header.displayname:cu
sourcetype::gws:gmail ] [ AND entities.users()
sourcetype::uba_threat_json ] [ AND sourcetype
OR uid:user1 user_id:user1 ] ] ]
```

# Data Out



*I thought this session was about SPEED!*

In a lake of events,  
how quickly can you  
eliminate Buckets?

# Data Out - 1. Index + Time

## First Bucket Reduction

1. Index + Time

FREE

The screenshot shows the Splunk Search interface. At the top, there are navigation tabs: Search, Analytics, Datasets, Reports, Alerts, and Dashboards. On the right, there is a 'Search & Reporting' button with a green arrow icon. Below the navigation, the 'New Search' page is displayed. The search query 'index=MyData' is entered in the search bar. To the right of the search bar, there are buttons for 'Save', 'Create Table View', and 'Close'. Below the search bar, there is a 'Date time range' dropdown menu with a search icon. The dropdown menu is open, showing options: Presets, Relative, Real-time, Date Range, and Date & Time Range. The 'Date & Time Range' option is selected and expanded, showing a configuration for a date range: 'Between' with a dropdown arrow, followed by two date-time fields: '04/27/2023 16:55:03.000' and '04/27/2023 16:55:13.001', separated by 'and'. Below these fields are the labels 'HH:MM:SS.SSS'. The main search results area shows '1 event (4/27/23 4:55:03.000 PM to 4/27/23 4:55:13.001 PM)' and 'No Event Sampling'. Below the search results, there are tabs for 'Events (1)', 'Patterns', and 'Statistics'. There are also buttons for 'Format Timeline' and 'Zoom Out'.

# Data Out - 2. Meta (H, S, ST) + Bloom Filters

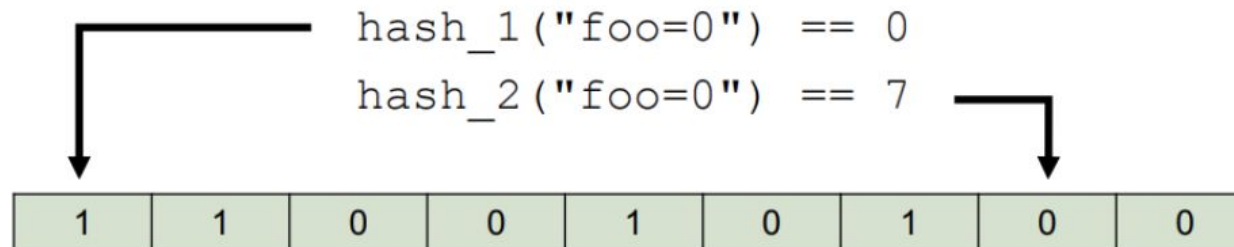
1. Index + Time

2. Meta + Bloom

FREE

Eliminate More Buckets!

Bloom Filters answer: does term "foo=0" exit?  
Can guarantee that term does NOT exist in any events = SKIP



But cannot guarantee term does exist ~98% accurate

Bloom Filters **DO NOT** use Wildcards\*!

# Data Out - 3. Search the Index

EarliestEventTime\_LatestEventTime.tsidx

```
CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
source::license_usage.log
sourcetype::splunkd
```

```
05
05-25-2023
100
2023
21
21:44:26.100
```

```
25
26
44
6516
auth
auth:sudo
b
b=6516
ec2
idx
idx=
info
st
st=
sudo
```

1. Index + Time

2. Meta + Bloom

3. LISPY / Index

**FAST**

TSIDX files are FAST! But not Free.

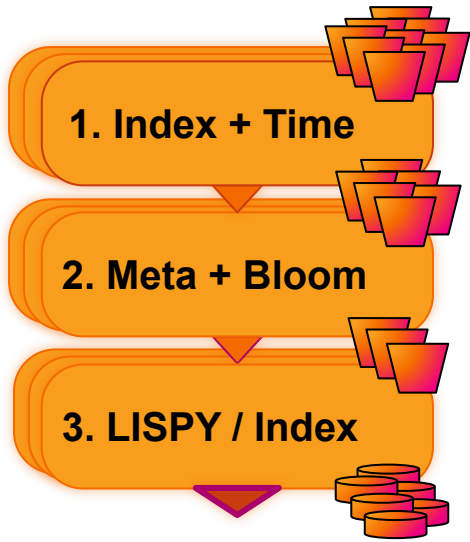
TSIDX points to specific event(s)

An event is stored in a Slice





# Data Out - 3. Search the Index



EarliestEventTime\_LatestEventTime.tsidx

```
CustomField::Case Matters
customfield::case matters
host::idx1.cloud.com
source::license_usage.log
sourcetype::splunkd
```

```
05
05-25-2023
100
2023
21
21:44:26.100
25
26
44
```

TSIDX files are FAST! But not Free.

TSIDX points to specific event(s)

An event is stored in a Slice



```
| search index=_internal b=6516
```

```
b
b=6516
ec2
```

```
[ AND index::_internal [ OR 6516 b::6516 ] ]
```

```
st
st=
sudo
```

# Data Out - 3. Search the Index with TERM

EarliestEventTime\_LatestEventTime.tsidx

CustomField::Case Matters  
 customfield::case matters  
 host::idx1.cloud.com  
 source::license\_usage.log  
 sourcetype::splunkd

05  
 05-25-2023  
 100  
 2023  
 21  
 21:44:26.100  
 25  
 26  
 44

TSIDX files are FAST! But not Free.

TSIDX points to specific event(s)

An event is stored in a Slice 

1. Index + Time

2. Meta + Bloom

3. LISPY / Index

```
| search index=_internal TERM(b=6516)
```

b=6516

ec2

```
[ AND index::_internal b=6156 ]
```

st=  
 sudo

# TERM()

It's not magic!

## Search 1

Slow ⌚ 144 Seconds

```
| search index=proxy q=* www.website.com*
```

scanning 22,437,072 events in 144.139 seconds

Wildcards

Minor Breakers

## Search 2

Fast 🕒 37 Seconds

```
| search index=proxy TERM(q=*) TERM(www.website.com*)
```

scanning 2,896,022 events in 37.635 seconds

# TERM( )

It's not magic - Searching Breakers

## Minor Breakers

/ \ : = @ . - \_ \$ # %

**IPs**

### User Search

192.0.0.1

### Base Lispy

[AND 0 1 192]

### TERM( ) Lispy

192.0.0.1

**Emails**

email@domain.com

[AND com domain email]

email@domain.com

**Domains**

host.domain.tld

[AND domain host tld]

host.domain.tld

**Usernames**

domain\user

[AND domain user]

domain\user

**Field??**

b=6516

[AND 6156 b::6516]

b=6516

# TERM( )

It's not magic - Searching Wildcards



Trailing\*

## User Search

SPLUN\*

\*Leading

\*PLUNK

\*Cont-ains\*

\*OMAIN\USE\*

Field = \*

B=\*

Field\* =

B\*=6516

## Base Lispy

splun\*

*ALL THE EVENTS*

[AND use\*]

*ALL THE EVENTS*

[OR 6156 b\*::6156]

## TERM( ) Lispy

splun\*

\*plunk

\*omain\use\*

b=\*

b\*=6516

Do Bloom Filters Support Wildcards?

**NO**



# LISPY

## LISPY is KEWL, Where Do I See That?

**Clara-fication: Job Inspector**  
Clara-fy your jobs

Clara Merriman  
Senior Splunk Engineer | Splunk

Martin Müller  
Principal Consultant | Consiast Software Solutions

<https://conf.splunk.com/files/2020/slides/TRU1143C.pdf>

Learn more about troubleshooting empty search results at [Splunk](#)

(SID: 1685326569.159351) [search.log](#) [Job Details Dashboard](#)

```
splunkcloud.com | lispy | 2/2 | ^ | v | x
Expanded index
05-29-2023 04:11:54.059 INFO
UnifiedSearch [32054 searchOrchestrator] -
base lispy: [ AND [ OR 6516 b::6516 ] ]
05-29-2023 04:11:54.060 INFO
```

## Search Performance Evaluator Dashboard

**Search specification**

| Name          | Description                    | Value                              |
|---------------|--------------------------------|------------------------------------|
| SPL           | The search entered by the user | search b=6516                      |
| Optimized SPL | The search post optimization   | search b=6516                      |
| Keywords      | The keywords found in SPL      | b::6516                            |
| <b>LISPY</b>  | The query on TSIDX             | <b>[ AND [ OR 6516 b::6516 ] ]</b> |



**TSTATS and PREFIX**

How to get the most out of your lexicon, with walklex, tstats, indexed fields, PREFIX, TERM and CASE

Richard Morgan  
Principal Architect | Splunk

<https://conf.splunk.com/files/2020/slides/PLA1089C.pdf>

[https://github.com/silkyrich/cluster\\_health\\_tools/blob/master/default/data/ui/views/search\\_performance\\_evaluator.xml](https://github.com/silkyrich/cluster_health_tools/blob/master/default/data/ui/views/search_performance_evaluator.xml)



# Example 1

Windows Security Event Logs

## Search 1

Slow  890 Seconds

```
| search index=windows_sec user="user1" EventCode=4624
```

scanning **240,233,347** events in **889.208** seconds

## Search 2

Fast  12 Seconds

```
| search index=windows_sec user="user1" "user1" EventCode=4624
```

scanning **1,043** events in **12.62** seconds

# Understanding Example 1

What is the LISPY Doing?

```
| search user=user1
```

```
[ OR user1 sourcetype::crowdstrike:events:sensor
sourcetype::crowdstrike:events:streams
sourcetype::gws:reports:admin sourcetype::gws:reports:calendar
sourcetype::gws:reports:context_aware_access
sourcetype::gws:reports:drive sourcetype::gws:reports:gcp
sourcetype::gws:reports:groups_enterprise
sourcetype::gws:reports:login sourcetype::gws:reports:saml
sourcetype::gws:reports:token sourcetype::oktaim2:log user1
user::user1 [ AND sourcetype::1password:insights:item_usages
user.uuid::user1 ] [ AND
sourcetype::1password:insights:signin_attempts
target_user.uuid::user1 ] [ AND profile.name::user1
sourcetype::oktaim2:group ] [ AND profile.login::user1
sourcetype::oktaim2:user ] [ AND sourcetype::duo:authentication
user.name::user1 ] [ AND
message_info.source.from_header_displayname::user1
sourcetype::gws:gmail ] [ AND entities.users{}.name::user1
sourcetype::uba_threat_json ] [ AND sourcetype::audittrail [ OR
uid::user1 user_id::user1 ] ] ]
```

LISPY

## 3 Possible Solutions:

```
| search user=user1 user1
```

```
[ AND user1 ]
```

LISPY

```
| search TERM(user=user1)
```

```
[ AND user=user1 ]
```

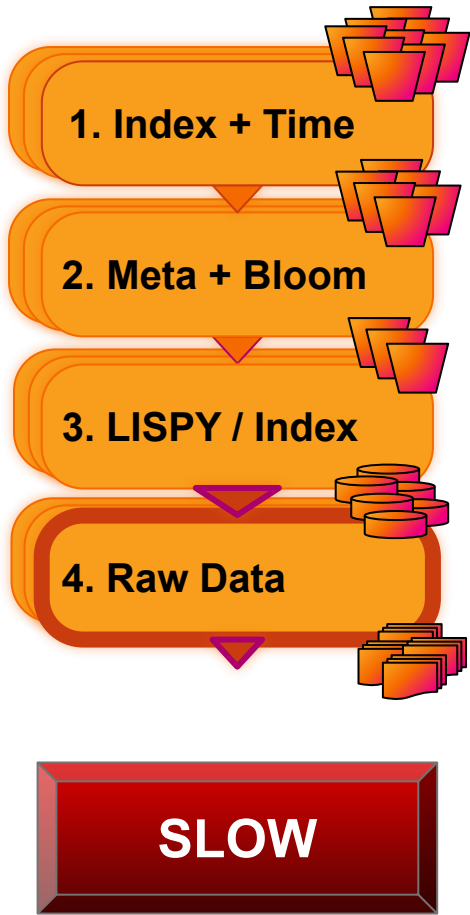
LISPY

```
| search user::user1
```

```
[ AND user::user1 ]
```

LISPY

# Data Out - 4. Read Compressed Data



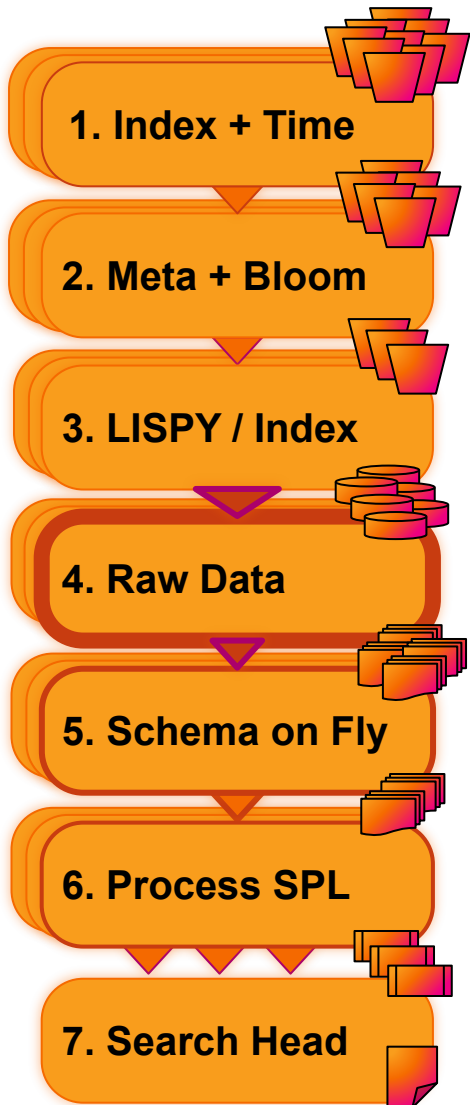
Extract Slice of Compressed Raw Data

Decompress Events (CPU Expensive)

Extract Specific Events (too many is Memory Expensive)



# Data Out - 5.Parse, 6.SPL, 7.Search Head



5. Schema on the Fly - parse fields

6. Streaming SPL™ Commands

7. Return required data to the Search Head  
**Verbose or Fast?**

# Why is That Faster?

## Data In

## Data Out

## Even Faster

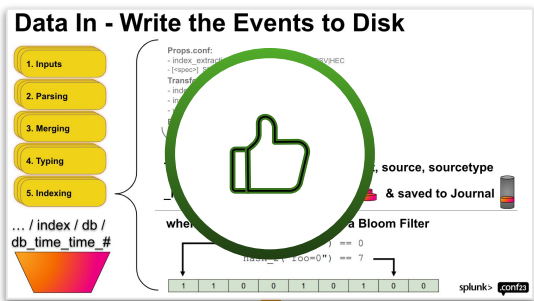
### Data In - Write the Events to Disk

1. Inputs
2. Parsing
3. Merging
4. Typing
5. Indexing

... / index / db / db\_time\_time\_#

*... source, sourcetype & saved to Journal*

when ... Bloom Filter



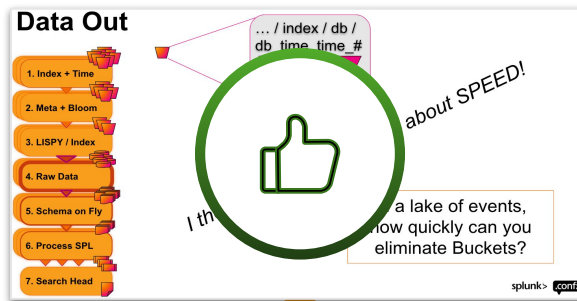
### Data Out

1. Index + Time
2. Meta + Bloom
3. LISPY / Index
4. Raw Data
5. Schema on Fly
6. Process SPL
7. Search Head

... / index / db / db\_time\_time\_#

*I thought about SPEED!*

*a lake of events, how quickly can you eliminate Buckets?*

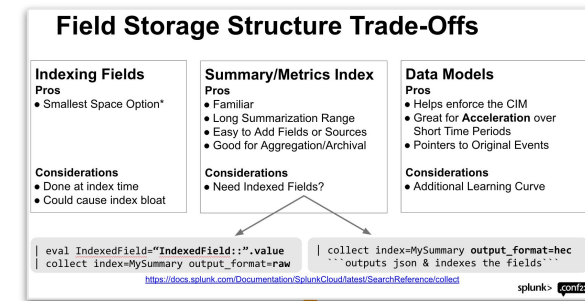


### Field Storage Structure Trade-Offs

| Indexing Fields  | Summary/Metrics Index   | Data Models   |
|--|---|---|
| <b>Pros</b><br>• Smallest Space Option*                                    | <b>Pros</b><br>• Familiar<br>• Long Summarization Range<br>• Easy to Add Fields or Sources<br>• Good for Aggregation/Archival | <b>Pros</b><br>• Helps enforce the CIM<br>• Great for Acceleration over Short Time Periods<br>• Pointers to Original Events |
| <b>Considerations</b><br>• Done at index time<br>• Could cause index bloat | <b>Considerations</b><br>• Need Indexed Fields?   | <b>Considerations</b><br>• Additional Learning Curve  |

```
eval IndexedField="IndexedField:":value | collect index=MySummary output_format=hec
collect index=MySummary output_format=raw outputs json & indexes the Fields...
```

<https://docs.splunk.com/Documentation/SplunkCloud/latest/SearchReference/collect>



### Indexing Fields

05-25-2023 21:44:26.100 INFO

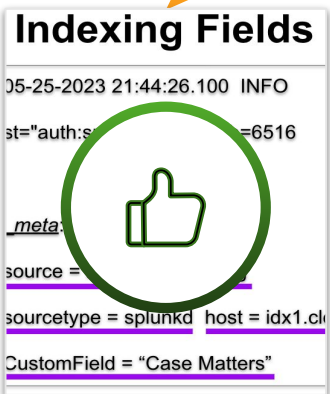
st="auth:s...=6516

*meta:*

*source =*

*sourcetype = splunkd host = idx1.cl*

*CustomField = "Case Matters"*



### Walklex

to Investigate the Index

```
| walklex index=main sourcetype=field
| stats values as NumValues by
```

✓ 1/1

Visualiz

100

field ↓

NumV

arn:aws:s3

arn:aws:sts

arn:aws:iam



### TERM()

It's not magic - Searching Breake

IPs

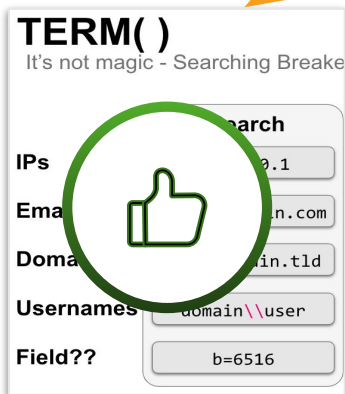
Email

Domain

Username

Field??

b=6516



### LISPY

LISPY IS KEWL, Where do I See That?

Job

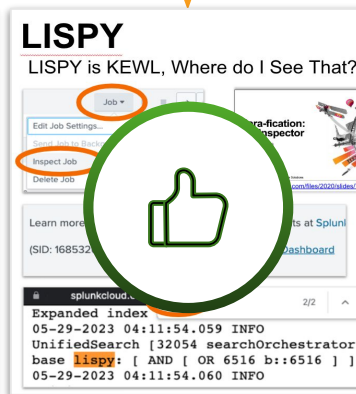
Inspect Job

Learn more

(SID: 168532)

Expanded index

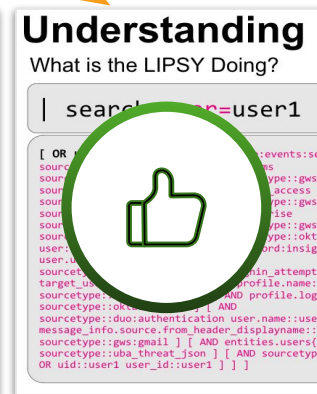
```
05-29-2023 04:11:54.059 INFO
UnifiedSearch [32054 searchOrchestrator]
base lispys: [ AND [ OR 6516 b::6516 ] ]
05-29-2023 04:11:54.060 INFO
```



### Understanding LISPY

What is the LISPY Doing?

```
[ OR
sourcetype:duo:authentication user.name:user
message_info.source.from.header.displayName:cu
sourcetype:aws:mail [ AND entities.users()
sourcetype:uba_threat_json [ AND sourcetype
OR uid:user1 user_id:user1 ] ] ]
```



### Tstats

Use the Index, Just the Index

Search 1

Slow 120 Seconds

```
| search index=window
```

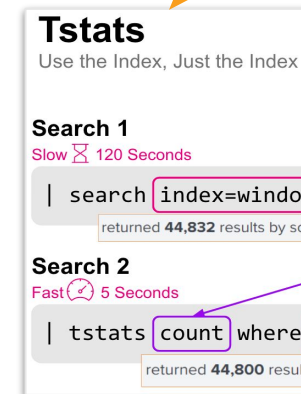
returned 44,832 results by sca

Search 2

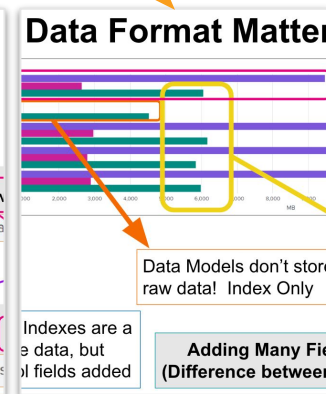
Fast 5 Seconds

```
| tstats count where
```

returned 44,800 results



### Data Format Matter



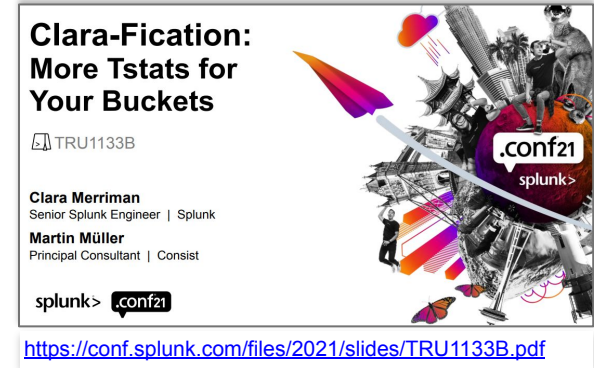
Data Models don't store raw data! Index Only

Indexes are a e data, but l fields added

Adding Many Fie (Difference between

# Tstats

Use the Index, Just the Index



## Search 1

Slow ⌚ 120 Seconds

```
| search index=windows_security | stats count by source
```

returned 44,832 results by scanning 196,540,296 events in 120.761 seconds

## Search 2

Fast 🕒 5 Seconds

```
| tstats count where index=windows_security by source
```

returned 44,800 results by scanning 196,407,587 events in 4.557 seconds

# Tstats Prefix( )

Use the Index, Just the Index for any TERM

```
| tstats sum(PREFIX(b=)) as Bytes WHERE index=_internal
source=*/license_usage.log by PREFIX(05-) _time span=1d
```

CustomField::Case Matters  
 customfield::case matters  
 host::idx1.cloud.com

05-25-2023

auth:sudo

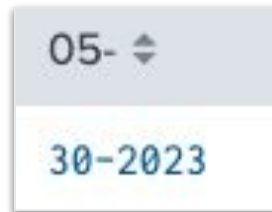
b=6516

ec2

idx=

st=

- PREFIX( ) can use any TERM in the index
- Want to count bytes by terms that start with 05?



- Major Breakers are not indexed: {"json":"value"}
  - That is why PREFIX(st=) didn't work

- | tstats has SPAN built in!

# Field Storage Structure Trade-Offs

## Add Fields to Index

### Pros

- Smallest Space Option\*

### Considerations

- Done at index time
- Could cause index bloat

## Summary/Metrics Index

### Pros

- Familiar
- Long Summarization Range
- Easy to Add Fields or Sources
- Good for Aggregation/Archival

### Considerations

- Only Copy Required Data
- Need Indexed Fields?

## Data Models

### Pros

- Helps enforce the CIM
- Great for **Acceleration** over Short Time Periods
- Pointers to Original Events

### Considerations

- Additional Learning Curve

```
| eval IndexedField="IndexedField:".value
| collect index=MySummary output_format=raw
```

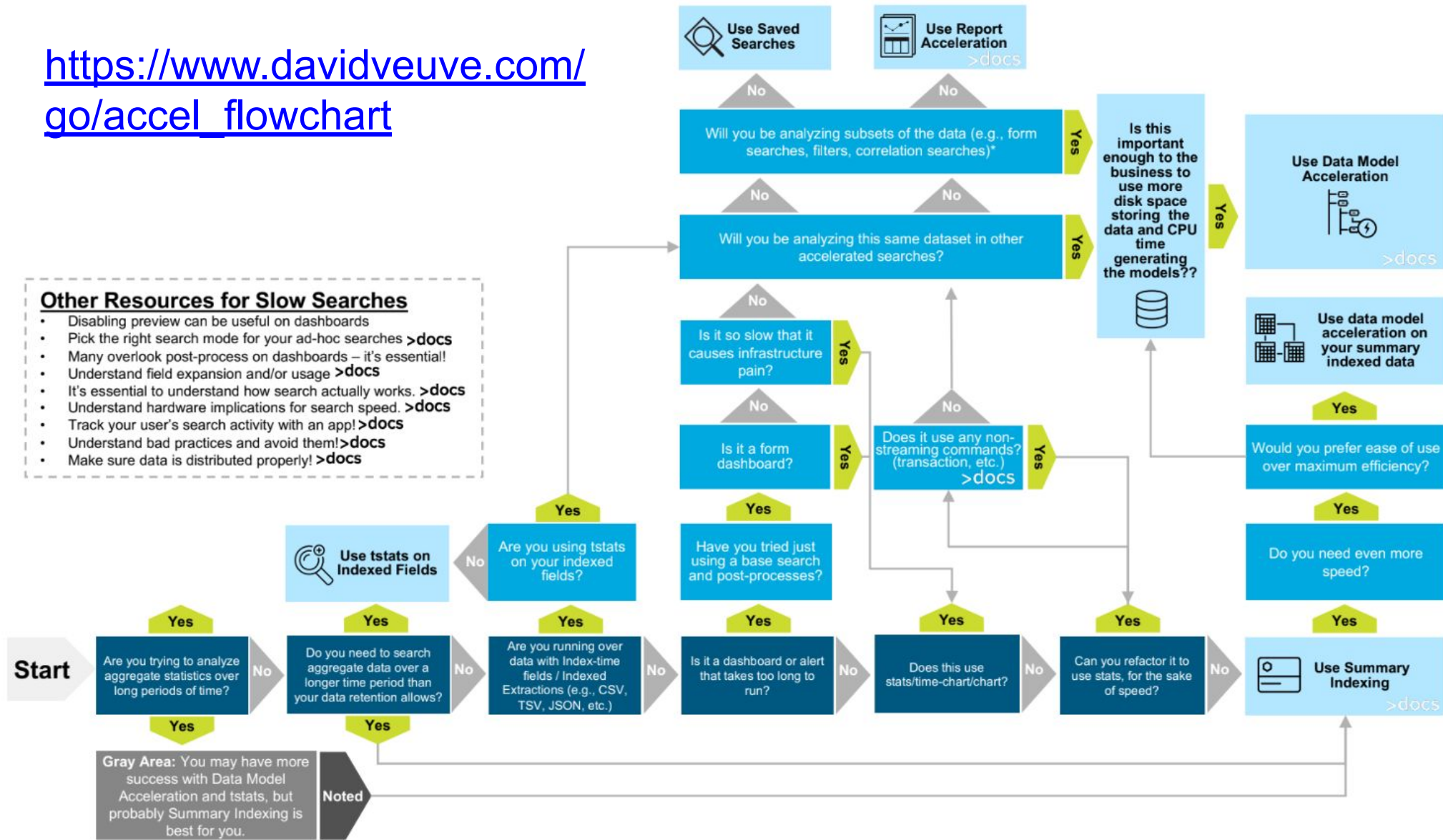
```
| collect index=MySummary output_format=hec
````outputs json & indexes the fields````
```

<https://docs.splunk.com/Documentation/SplunkCloud/latest/SearchReference/collect>



[https://www.davidveuve.com/go/accel\\_flowchart](https://www.davidveuve.com/go/accel_flowchart)

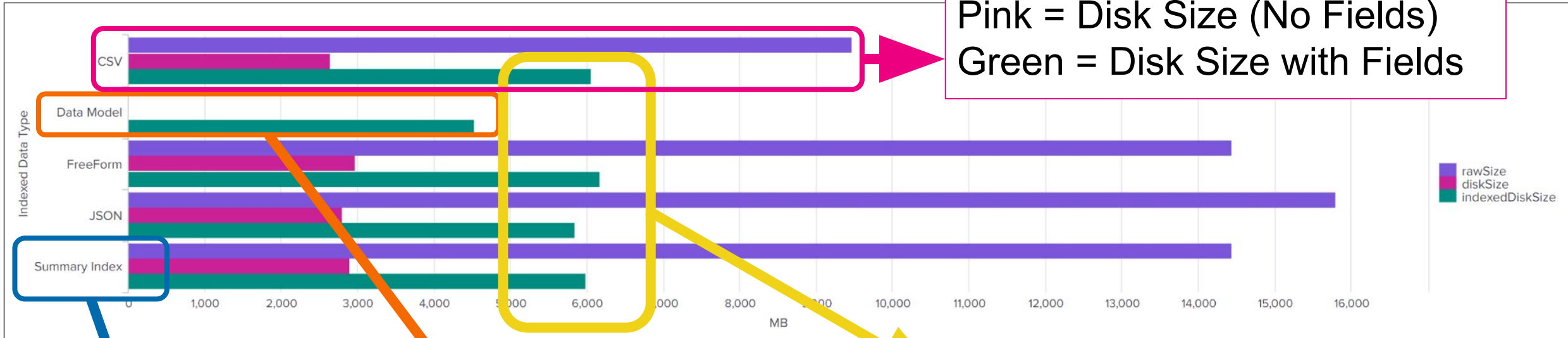
- Other Resources for Slow Searches**
- Disabling preview can be useful on dashboards
  - Pick the right search mode for your ad-hoc searches >docs
  - Many overlook post-process on dashboards – it's essential!
  - Understand field expansion and/or usage >docs
  - It's essential to understand how search actually works. >docs
  - Understand hardware implications for search speed. >docs
  - Track your user's search activity with an app! >docs
  - Understand bad practices and avoid them! >docs
  - Make sure data is distributed properly! >docs



\* Technically you can do with with Report Acceleration or Saved Searches, but it gets tricky and confusing pretty quickly.

7:10:57:153] "GET /category.screen?category\_id=GLFTS&JSESSIONID=SD15L4FF19ADFF10 HTTP 1.1" 404 3322 "http://buttercup-shopping.com/cart.do?action=view&itemId=EST-6&product\_id=FI-5W-83" "Opera/9.80...  
/Jan 18:10:57:123] "GET /product.screen?product\_id=FL-D5H-01&JSESSIONID=SD55L7FF6ADFF9 HTTP 1.1" 404 1318 "http://buttercup-shopping.com/cart.do?action=purchase&itemId=EST-6&product\_id=FI-5W-83" "Opera/9.80...  
CLR 1.1.4322)" "GET /oldlink?item\_id=EST-26&JSESSIONID=SD55L9FF1ADFF3 HTTP 1.1" 200 1318 "http://buttercup-shopping.com/cart.do?action=changequantity&itemId=EST-18&product\_id=AV-LS-03&JSESSIONID=SD55L9FF1ADFF3" "Opera/9.80...  
com/oldlink?item\_id=EST-26&JSESSIONID=SD55L9FF1ADFF3 HTTP 1.1" 200 1318 "http://buttercup-shopping.com/cart.do?action=changequantity&itemId=EST-18&product\_id=AV-LS-03&JSESSIONID=SD55L9FF1ADFF3" "Opera/9.80...

# Does Data Format Matter?



Purple = Raw Event Size  
Pink = Disk Size (No Fields)  
Green = Disk Size with Fields

Data Models don't store raw data! Index Only

Total disk is about the same no matter the data format

Summary Indexes are a copy of the data, but you control fields added

**Adding Many Fields Causes Bloat (Difference between Pink & Green Bars)**

# Why is That Faster?

## Data In

## Data Out

## Even Faster

**Data In - Write the Events to Disk**

- Inputs
- Parsing
- Merging
- Typing
- Indexing

... / index / db / db\_time\_time\_#

source, sourcetype

& saved to Journal

Bloom Filter

1 1 0 0 1 0 1 0 0 0

**Data Out**

- Index + Time
- Meta + Bloom
- LISPY / Index
- Raw Data
- Schema on Fly
- Process SPL
- Search Head

... / index / db / db\_time\_time\_#

about SPEED!

I think you can eliminate Buckets?

a lake of events, how quickly can you eliminate Buckets?

**Field Storage Structure Trade-Offs**

**Indexing Fields**

Pros

- Smallest Space Option\*

Considerations

- Done at index time
- Could cause index bloat

**Summaries**

Pros

- Helps enforce the CIM
- Great for Acceleration over Short Time Periods
- Pointers to Original Events

Considerations

- Additional Learning Curve

**Data Models**

Pros

- Helps enforce the CIM
- Great for Acceleration over Short Time Periods
- Pointers to Original Events

Considerations

- Additional Learning Curve

eval IndexedField="IndexedField:" collect index=MySummary output\_format=raw outputs json & indexes the Fields...

<https://docs.splunk.com/Documentation/SplunkCloud/latest/SearchReference/collect>

**Indexing Fields**

05-25-2023 21:44:26.100 INFO

st="auth:s=6516

meta:

source =

sourcetype = splunkd host = idx1.cl

CustomField = "Case Matters"

**Walklex**

to Investigate the Index

walklex index=prod type=field

stats as NumValues by

Visualiz

100

field

arn:aws:s3

arn:aws:sts

arn:aws:iam

**TERM()**

It's not magic - Searching Breake

IPs

Email

Domain

Username

Field??

b=6516

**LISPY**

LISPY IS KEWL, Where do I See That?

Expanded index

05-29-2023 04:11:54.059 INFO

UnifiedSearch [32054 searchOrchestrator]

base lisp: [ AND [ OR 6516 b::6516 ] ]

05-29-2023 04:11:54.060 INFO

**Understanding LISPY**

What is the LISPY Doing?

search user=user1

[ OR

source:gs:gmail ] [ AND entities.users(

source:gs:gmail ] [ AND sourcetype

OR uid:user1 user\_id:user1 ] ]

**Tstats**

Use the Index, Just the Index

Search Slow

Search Fast

tstats count where

returned 44,800 results

**Data Format Matter**

Indexes are a data, but fields added

Adding Many Fields (Difference between



# Culmination of 10 years of .conf

Kellen Green, Clara Merriman, Richard Morgan, Martin Müller, David Veuve, Brian Wooden, Simeon Yep  
And many more!


**Clara-Fication: More Tstats for Your Buckets**  
TRU1133B



Clara Merriman  
Senior Splunk Engineer | Splunk  
Martin Müller  
Principal Consultant | Conisat

<https://conf.splunk.com/files/2021/slides/TRU1133B.pdf>

**Behind the Magnifying Glass: How Search Works**



Jeff Champagne  
Staff Architect, Splunk

<https://conf.splunk.com/files/2016/slides/behind-the-magnifying-glass-how-search-works.pdf>

Splunk® Enterprise  
**Knowledge Manager Manual**

[Download manual as PDF](#)

<https://docs.splunk.com/Documentation/Splunk>

(12) **United States Patent**  
Baum et al.

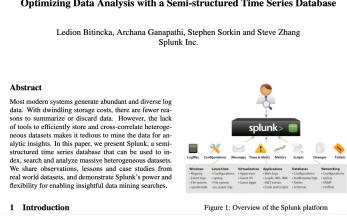
---

(54) **TIME SERIES SEARCH WITH INTERPOLATED TIME STAMP**

<https://image-ppubs.uspto.gov/dirsearch-public/print/downloadPdf/9002854>

**inurl:conf.splunk.com TSTATS TERM**

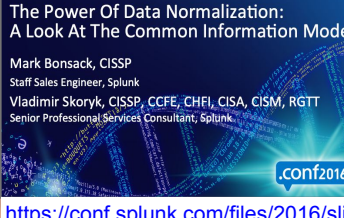
**Optimizing Data Analysis with a Semi-structured Time Series Database**



Ledion Bitincka, Archana Ganapathi, Stephen Serkin and Steve Zhang  
Splunk Inc.

[https://www.usenix.org/legacy/event/slaml10/tech/full\\_papers/Bitincka.pdf](https://www.usenix.org/legacy/event/slaml10/tech/full_papers/Bitincka.pdf)

**The Power Of Data Normalization: A Look At The Common Information Model**



Mark Bonsack, CISSP  
Staff Sales Engineer, Splunk  
Vladimir Skoryk, CISSP, CCFE, CHFI, CISA, CISM, RGTT  
Senior Professional Services Consultant, Splunk

<https://conf.splunk.com/files/2016/slides/the-power-of-data-normalization-a-look-at-cim-under-the-hood.pdf>

**Fields, Indexed Tokens, and You**  
PLA1466B



Martin Müller  
Principal Consultant | Conisat

<https://conf.splunk.com/files/2022/slides/PLA1466B.pdf>

**Revealing the Magic**  
The Lifecycle of a Splunk Search



Kellen Green | Senior Software Engineer  
September 27th, 2017 | Washington, DC

<https://conf.splunk.com/files/2017/slides/revealing-the-magic-the-life-cycle-of-a-splunk-search.pdf>

**INGEST\_EVAL and CLONE\_SOURCETYPE**  
Advanced pipeline configurations



Richard Morgan  
Vladimir Skoryk  
Splunk

<https://conf.splunk.com/files/2020/slides/PLA1154C.pdf>

**Splunk Performance**  
Observations and Recommendations



Simeon Yep | AVP, GSA  
Brian Wooden | GSA Partner Integrations  
2017-09-27 | Washington, DC

<https://conf.splunk.com/files/2017/slides/observations-and-recommendations-on-splunk-performance.pdf>

**Security Ninjutsu Part Four**  
The SPLening  
2.5 hours of EPIC SPL stuffed into 45 minutes



David Veuve | Principal Security Strategist  
September 2017 | Washington, DC

<https://www.davidveuve.com/presentations.html>

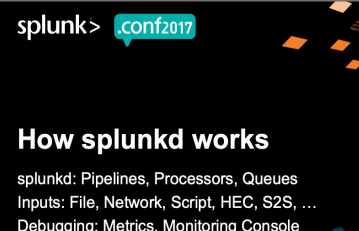
**Clara-fication: Job Inspector**  
Clara-fy your jobs



Clara Merriman  
Senior Splunk Engineer | Splunk  
Martin Müller  
Principal Consultant | Conisat Software Solutions

<https://conf.splunk.com/files/2020/slides/TRU1143C.pdf>

**How splunkd works**




splunkd: Pipelines, Processors, Queues  
Inputs: File, Network, Script, HEC, S2S, ...  
Debugging: Metrics, Monitoring Console

by Amrit Bath, Abhinav Nekkanti

<https://conf.splunk.com/files/2017/slides/how-splunkd-works.pdf>

**Searching FAST**  
How to Start Using Istats and Other Acceleration Techniques



David Veuve | Principal Security Strategist  
September 2017 | Washington, DC

<http://conf.splunk.com/files/2017/slides/searching-fast-how-to-start-using-tstats-and-other-acceleration-techniques.pdf>

**TSTATS and PREFIX**  
How to get the most out of your lexicon, with walklex, tstats, indexed fields, PREFIX, TERM and CASE



Richard Morgan  
Principal Architect | Splunk

<https://conf.splunk.com/files/2020/slides/PLA1089C.pdf>

# Thank You





# Bugs in Walklex

Do you Trust that Pattern?


**New Search**


```
| walklex index=crowdstrike type=field pattern=*volumeuuid*  
| stats avg(distinct_values) dc(source) as Buckets by field
```

✓ 91 events (Partial results for 5/28/23 12:00:00.000 AM to 5/31/23 1:53:49.000 AM)

No Event Sampling ▾

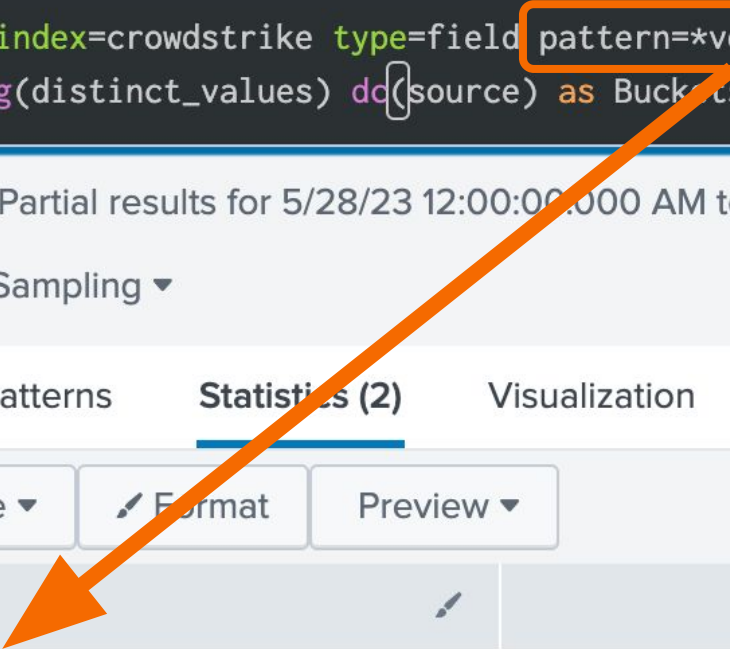
Events   Patterns   **Statistics (2)**   Visualization

100 Per Page ▾    Format   Preview ▾

field ▾ 

commandline

volumeuuid





# Lispy Wrangling

## Walklex in Searches

✓ 53 events

state  
hot

The screenshot shows a Splunk search interface. The search bar contains the query: `1 | dbinspect index=demonstratinghotbuckets` and `2 | table index state`. The search results show 1 result. The 'Statistics (1)' tab is active, displaying a table with columns 'index' and 'state'. The 'index' column contains the value 'demonstratinghotbuckets' and the 'state' column contains the value 'hot'. A callout box from the '53 events' text points to the search bar, and another callout box from the 'state' dropdown points to the 'hot' value in the table.

**Important:** The `walklex` command does not work on hot buckets.

The screenshot shows a Splunk search interface. The search bar contains the query: `1 | walklex index=demonstratinghotbuckets`. The search results show 53 events. The 'Events (53)' tab is active, displaying a table with columns 'index' and 'state'. The 'index' column contains the value 'demonstratinghotbuckets' and the 'state' column contains the value 'hot'. A callout box from the '53 events' text points to the search bar.

# Lispy Wrangling

## Walklex in Searches

The screenshot shows the Splunk Enterprise interface. At the top, the navigation bar includes 'splunk>enterprise', 'Apps', 'Administrator', '3 Messages', 'Settings', 'Activity', and 'Help'. A search bar on the right contains the text 'Find'. Below this is a secondary navigation bar with 'Search', 'Analytics', 'Datasets', 'Reports', 'Alerts', 'Dashboards', and 'Search & Reporting' (highlighted with a green arrow icon).

The main content area is titled 'New Search'. It features a search bar with the query: `1 | walklex index=summaryindexedfields type=term`. To the right of the search bar is a 'Date time range' dropdown and a green search icon. Below the search bar, the results are summarized as: **44,134 events** (2/10/20 12:56:01.000 PM to 3/18/20 4:09:21.000 PM). Additional controls include 'No Event Sampling', 'Job', a pause button, a refresh button, a download button, and 'Smart Mode'.

Below the summary, there are tabs for 'Events (44,134)', 'Patterns', 'Statistics', and 'Visualization'. The 'Events' tab is active. Below the tabs, there are controls for 'Format Timeline', 'Zoom Out', 'Zoom to Selection', and 'Deselect'. On the right, it says '1 day per column'. The bottom of the interface shows a timeline view with two columns for 'Dec 6, 2019' and 'Dec 7, 2019'. The 'Dec 6, 2019' column has a small green bar at the top, and the 'Dec 7, 2019' column has a small blue bar at the top.

# Lispy Wrangling

Walklex in Searches

```
1584572961-1581368161-292888130366260598.tsidx  
1587566595-1577378311-246491582942466064.tsidx  
1596745603-1596741219-461917483128354659.tsidx  
1599805871-1577305669-431433424464043900.tsidx  
1599809803-1576373397-435127942447157859.tsidx  
1600695635-1577668933-422669603826096087.tsidx  
1605806101-1575693933-423399937244955979.tsidx  
1606406553-1578937623-417986994256767567.tsidx  
1606419685-1576555609-425504948282739942.tsidx  
1606837757-1575692577-474445073883345343.tsidx  
1606858987-1575691189-412316302012581430.tsidx
```