It’s 10:00 PM.
Do You Know Where Your Data Is?

Jason

Engagement Manager, Splunk Professional Services

Disclaimer

During the course of this presentation, we may make forward looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. 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. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

Splunk or its representatives can not be responsible for the time lost while viewing this presentation.

About Jason

Edit | More Info

Years working with Splunk

6 ↗ 1

Customers Helped

100+

Continents Worked

5 → 0

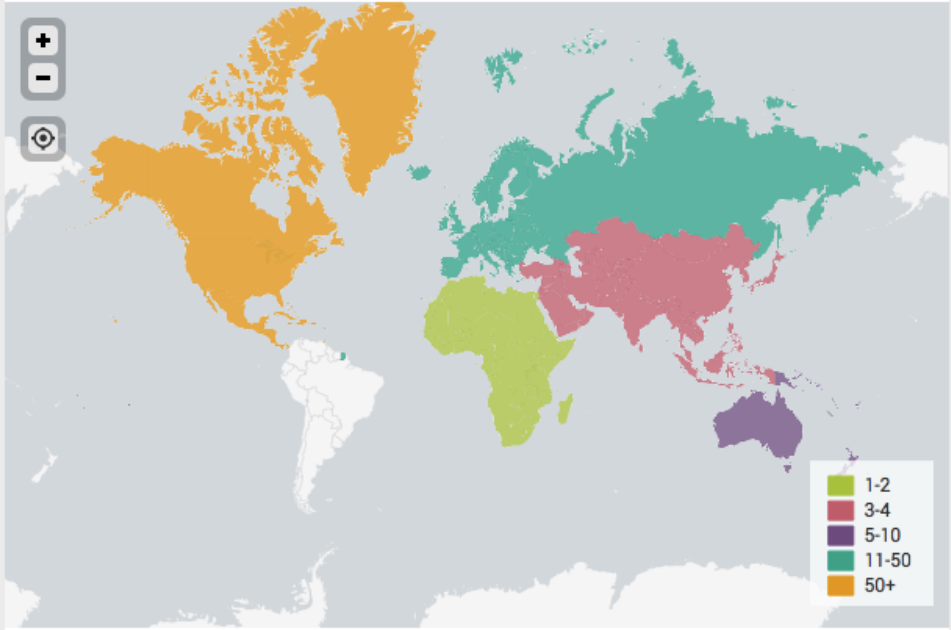
Shirts Acquired

28 ↗ 5

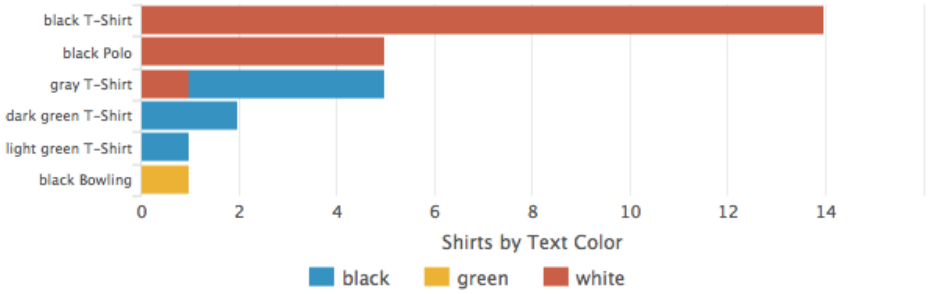
Favorite T-Shirt Slogan

Cool story, bro.

Number of Customers Helped by Continent



Shirt Metrics



Acquisition over Time

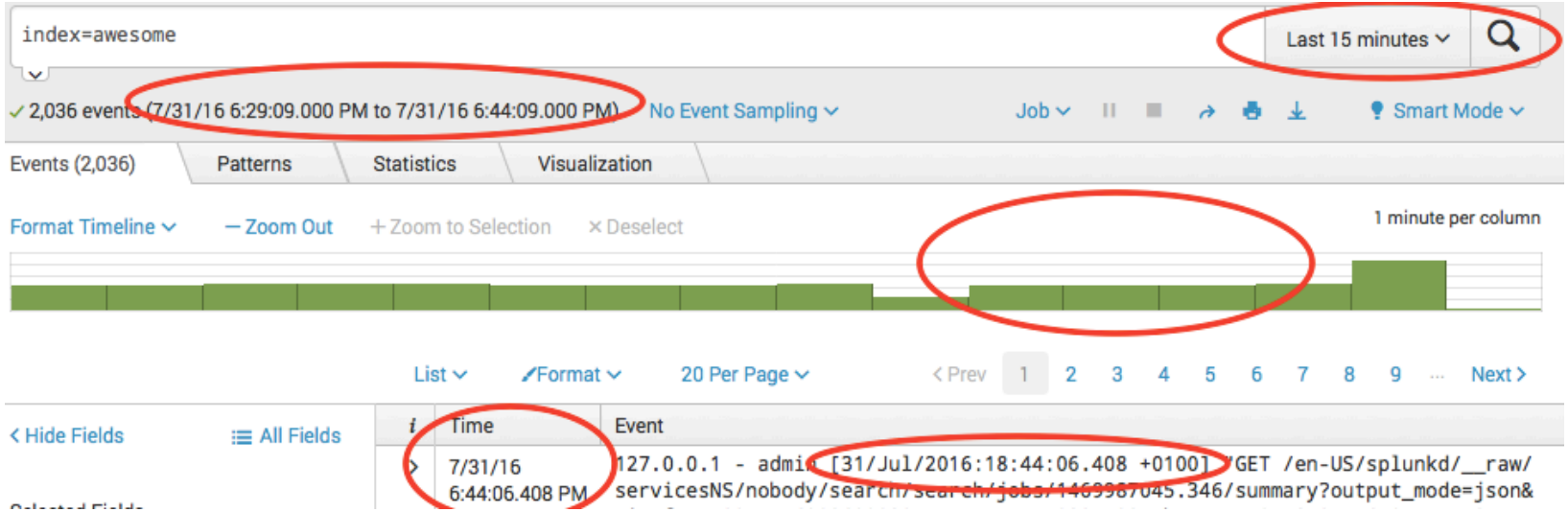
What We Will Cover

- Why is Time a Problem in Splunk?
  - Splunk searches by time, and if it's wrong, Splunk won't find your data
- How do I Identify a Time Problem in Splunk?
  - Get to know _time and _indextime
- How do I Fix a Time Problem in Splunk?
  - Learn the configurations and where to put them

Why is Time a Problem?

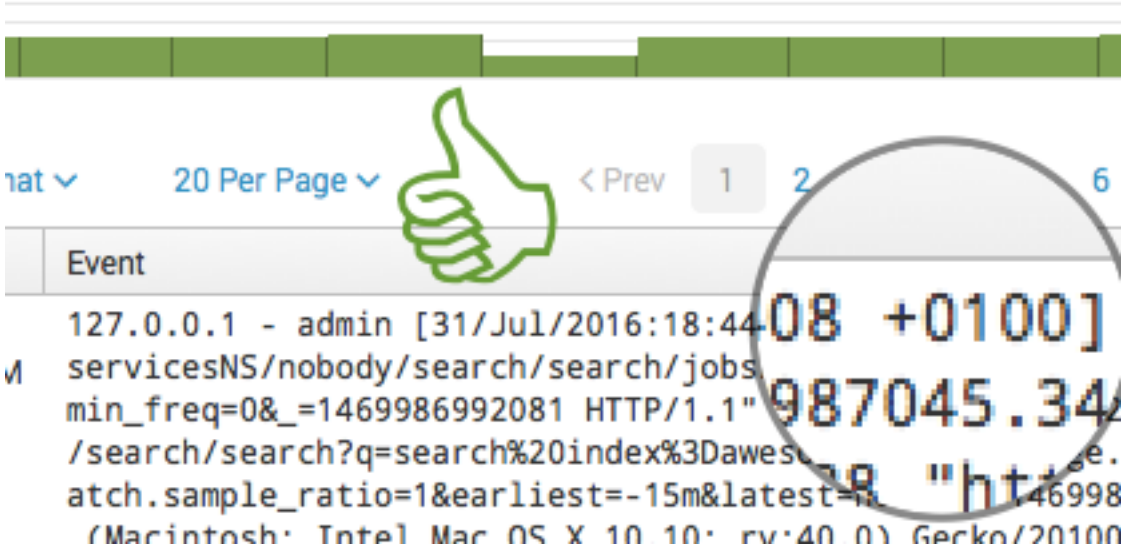
Splunk is Time-based

- The timestamp extracted from the event... is used to store it
- Searches execute based on that event time



Splunk Needs a Time Zone

- Indexers* use the timezone to normalize event times into epoch
- They get it from one of these places:
  - The event text
  - TZ in props.conf
  - The UF's† timezone
  - Its own timezone



- * First Indexer or HF
- † If UF is 6.x+

Splunk is Very Tolerant of Future/Past Times

- Splunk trusts the event to report its actual time
- It is designed to bring in archive data and remembers the previous event

`MAX_DAYS_AGO = 2000`
Highest number of days in the past a timestamp can be valid

`MAX_DAYS_HENCE = 2`
Highest number of days in the future a timestamp can be valid

`MAX_DIFF_SECS_AGO = 3600`
Highest number of seconds in the past a timestamp can be valid *compared to the previous event*

`MAX_DIFF_SECS_HENCE = 604800`
Highest number of seconds in the future a timestamp can be valid *compared to the previous event*

- These props.conf configurations manage keeping or throwing away timestamps

Demo

- Three live data sources generating

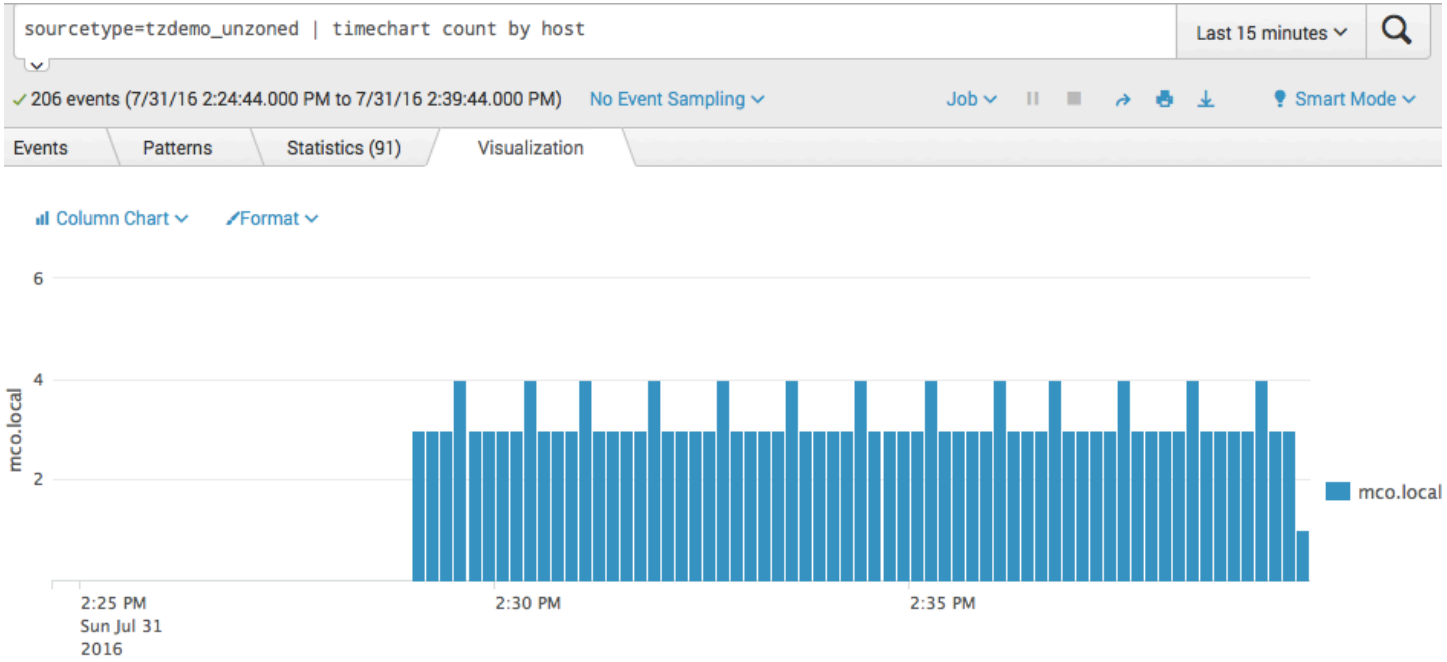
```
j@s data$ date;tail -n 4 *unzoned
Sun Jul 31 14:31:24 EDT 2016
==> LHRunzoned <==
2016-07-31 19:31:12 lhr.local This is a message from London with no timezone.
2016-07-31 19:31:16 lhr.local This is a message from London with no timezone.
2016-07-31 19:31:19 lhr.local This is a message from London with no timezone.
2016-07-31 19:31:22 lhr.local This is a message from London with no timezone.

==> MCOunzoned <==
2016-07-31 14:31:12 mco.local This is a message from Orlando with no timezone.
2016-07-31 14:31:16 mco.local This is a message from Orlando with no timezone.
2016-07-31 14:31:19 mco.local This is a message from Orlando with no timezone.
2016-07-31 14:31:22 mco.local This is a message from Orlando with no timezone.

==> SFOunzoned <==
2016-07-31 11:31:12 sfo.local This is a message from San Francisco with no timezone.
2016-07-31 11:31:16 sfo.local This is a message from San Francisco with no timezone.
2016-07-31 11:31:19 sfo.local This is a message from San Francisco with no timezone.
2016-07-31 11:31:22 sfo.local This is a message from San Francisco with no timezone.
j@s data$
```

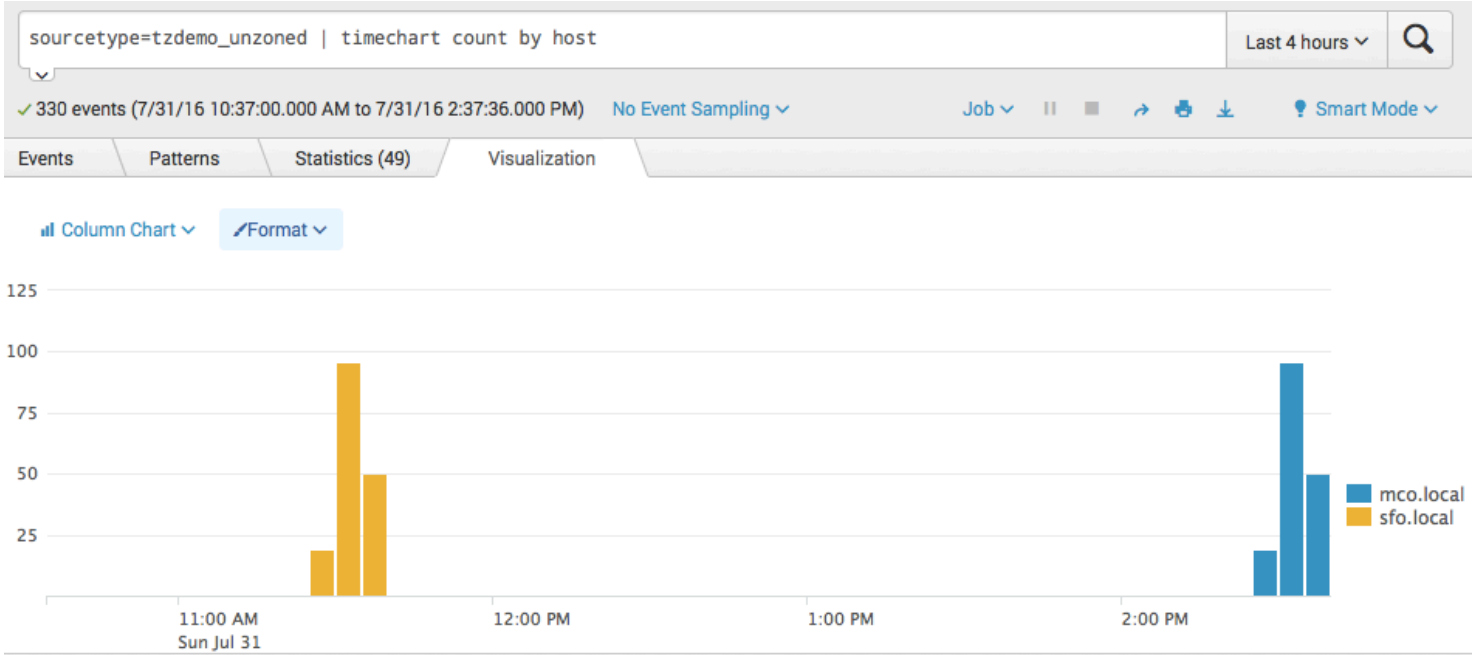
Demo

- Only one host (Orlando) shown in `Last 15 minutes` search



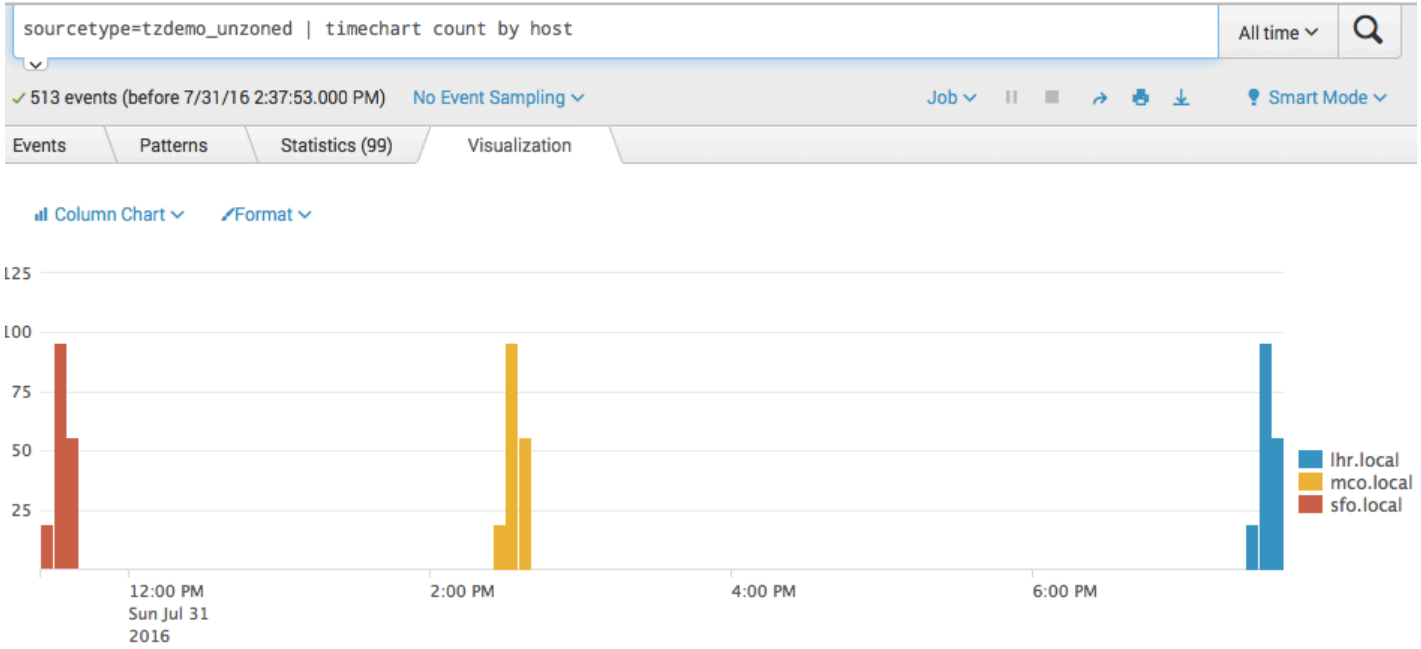
Demo

- San Francisco shown in `Last 4 hours` search, but no London.



Demo

- Must select `All Time` to find London data – in the "future"



Why is this Happening?

- San Francisco is 3 hours earlier than Orlando
- San Francisco data arrives at Orlando at 11am, appearing as 8am
- If the indexer can't get a time zone from the event or the UF (v6.0+ only) it uses its own
- Stores the event as 8am Orlando time (Noon UTC) in the index
- Search displays the Noon UTC event as 8am Orlando time to the user

Why is this a Problem?

- Data is searched on extracted time
- Searching `Last 60 Minutes` won't include California data
- Even `Last 24 Hours` won't include London data
  - Why? Because it's 5 hours in the "future"
  - `Last 24 Hours` searches `-24h` to `now`
- This affects all searches, including metrics, correlations, and alerts!

Key Takeaways

- If data does not have the correct time, and time zone, it will not be found in a recent-time search, such as `Last 15 Minutes`
- Critical data and correlations will be lost
- Calculations and stats may be inaccurate
- If data appears too far in the past, or at all in the future, it may **never** be accurately counted in statistics done in small chunks

How Do I Identify A Time Problem?

Know Your _times

_time

The extracted event time

- Time Picker
- Histogram
- earliest=/latest= on search bar

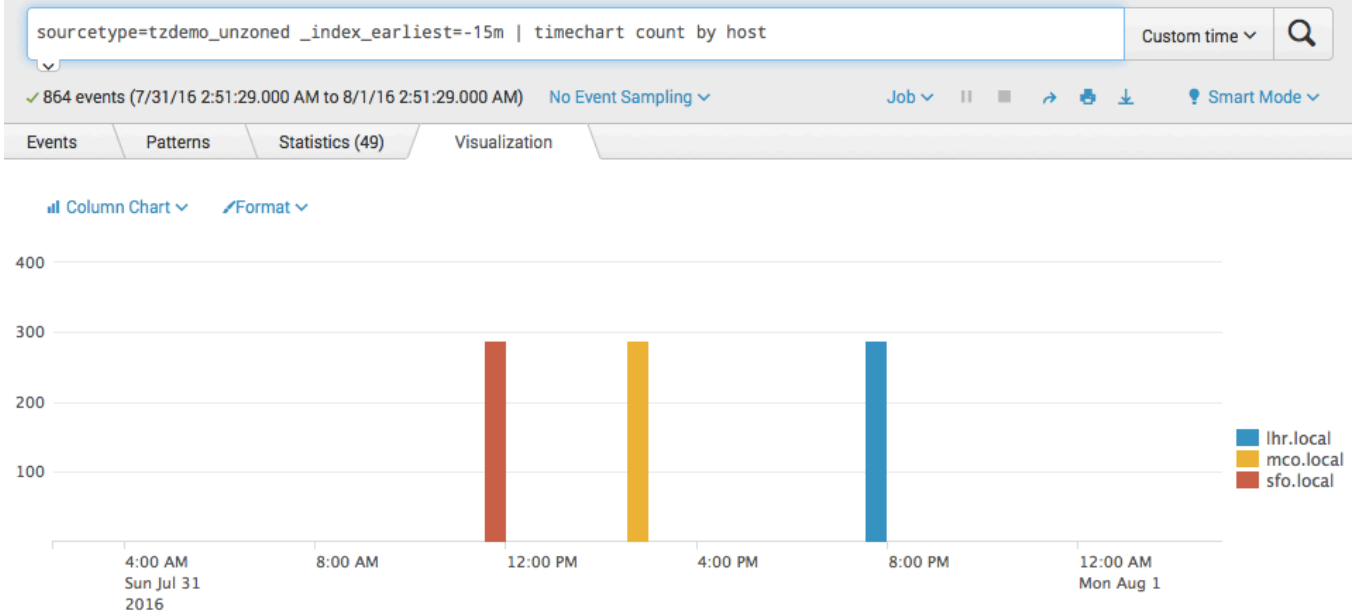
_indextime

The time Splunk received it

- Hidden field
- Can be used with eval
- _index_earliest=/_index_latest= on search bar

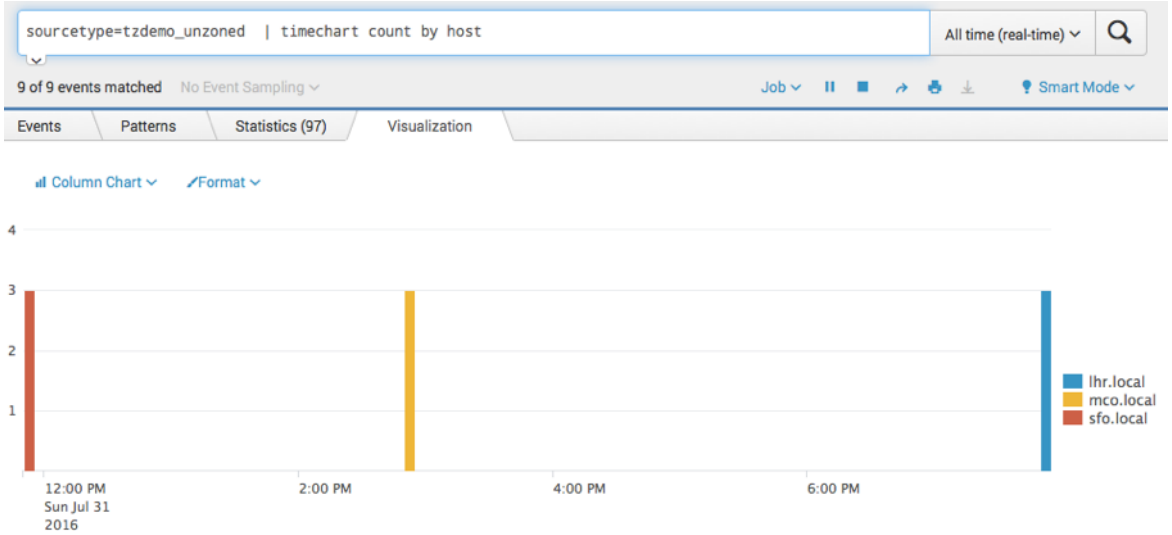
Quick Demo

- `_index_earliest=-15m` – and setting the time picker wide enough – finds all three data streams



All-time Real-time

- Useful for on the spot debugging – shows what is coming in now
- Performance hit – don't leave running



Use Searches to Calculate Hours of Difference

- Useful to find all hosts affected
- Periodically check data and alert
- Example:
- `... | eval diff=round((_time-_indextime)/3600, 1)`
  - Get the difference between event time and index time
- `| where abs(diff)>=0.5`
  - (Optional) Show only those events there there is a >30 minute difference
- `| stats values(diff) by host`
  - Show the hosts and how many kinds of time difference there were

Time Zone differences

- Single values of diff across all data from a host can indicate time zone problems

host	values(diff)
lhr.local	5.0
mco.local	0.0
sfo.local	-3.0

Time Zone Differences

- Odd values could mean multiple things
- Dumpbox probably receives hourly files
- Host2 should have its clock checked
- Web30 likely has some files in one time zone, and some in another
  - Logging in UTC is common for IIS logs

host	diff
dumpbox	-0.5
	-0.6
	-0.7
	-0.8
	-0.9
	-1.0
	-1.1
	-1.2
	-1.3
	-1.4
	-1.5
host2	2.3
	2.4
web30	0.0
	4.0

Key Takeaways

- Know how to use _time and _indextime
- Remember that _time has to match if searching with _index_earliest
- An all-time real-time search is a quick way to find out what is going on now

How Can I Fix A Time Problem?

.conf2016

splunk>

Remember the Basics

- Customize props.conf for each data source

- `TIME_PREFIX = ^`
  - This is the regular expression that the timestamp will immediately follow
- `TIME_FORMAT = %Y-%m-%d %H:%M:%S,%3N %Z`
  - This is the strptime-style format the timestamp will be in
- `TZ = America/New_York`
  - This is a timezone identifier from the tz database
  - See https://en.wikipedia.org/wiki/List_of_tz_database_time_zones

Ensure Correct Hosts Get the configs

- Timestamp and time zone is handled during the parsing phase
- This happens on the first Indexer or HF the data hits
- Settings will only affect new data coming in

Always Custom configs?

- Splunk will perform better on tasks, including timestamp extraction, if it has specific configurations and doesn't have to guess as much
- The easy way is to just make sure there is an easily-readable timestamp with a time zone in every line of your data

http://dev.splunk.com/view/logging-best-practices/SP-CAAADP6

Changing the Time of Events?

- _time is an indexed field
  - Malleable at search time (through eval)
  - Unable to be changed in the index
- If _time is wrong, data will need to be re-indexed

Be Aware When it Happens in the Future

- Fix the current issues
- Set up an alert based on difference between _time and _indextime!

It’s Wrap-up _time

Key Takeaways

- Splunk searches by the time it extracts from the event
- Time zones can cause data to consistently appear in the future or past, and never in a `last 15 minutes` search
- Simple configurations can set things right

THANK YOU

splunk>

Notes

- For making demo: local machine, python to make up demo files
- import pytz, datetime
- lhrtz = pytz.timezone("Europe/London") - mcotz, sfotz etc
- datetime.datetime.strftime(lhrtz.localize(datetime.datetime.now()).astimezone(sfotz), "%Y-%m-%d %H:%M:%S")
- Translates from one tz to another, then prints it out with no marker
  - Make files with TZ extensions, and without
  - Ingest into splunk