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 or the SEC's website at 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> .conf23

Splunk® Ingest Actions and Rulesets

Advanced Pipeline Configurations
PLA1641B

Luke Netto

Chief Technical Advisor | Splunk

Lane Netto

Senior Professional Services Consultant | Splunk

splunk> .conf23

Luke Netto

Chief Technical Advisor
Splunk

Lane Netto

Senior Professional Services Consultant
Splunk

splunk> .conf23

Who Are We?

- Over 15 years of combined Splunk experience
- Experienced with many diverse customer environments
- Subject to customer requirements and policies

Who Are You?

- Migrating environments
- Consolidating environments
- Forwarding into a centralized "corporate" instance
- Correcting “bad data” with limited access at the source

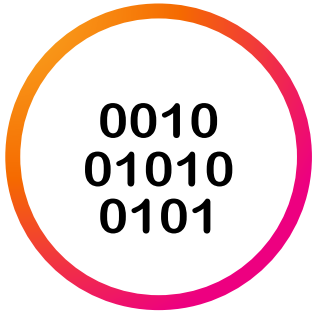
Why You Are Here?

- Overview of Ingest Actions
- Destinations
- Rulesets
- Four Real-World Scenarios

What Your Executives Want

Value from your data

Data



Getting Data In

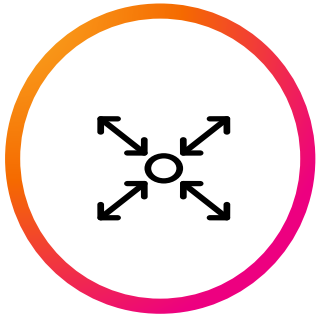
Reporting

Searching

Analysis

Dashboarding

Action



Reports/Alerts

Value

Use Case Completion

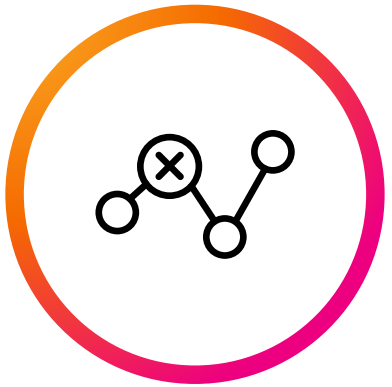
Ingest Actions Overview

Innovation in data preprocessing at Splunk

Filter

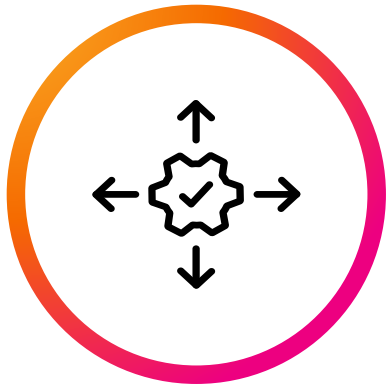
Discard unwanted events

Mask



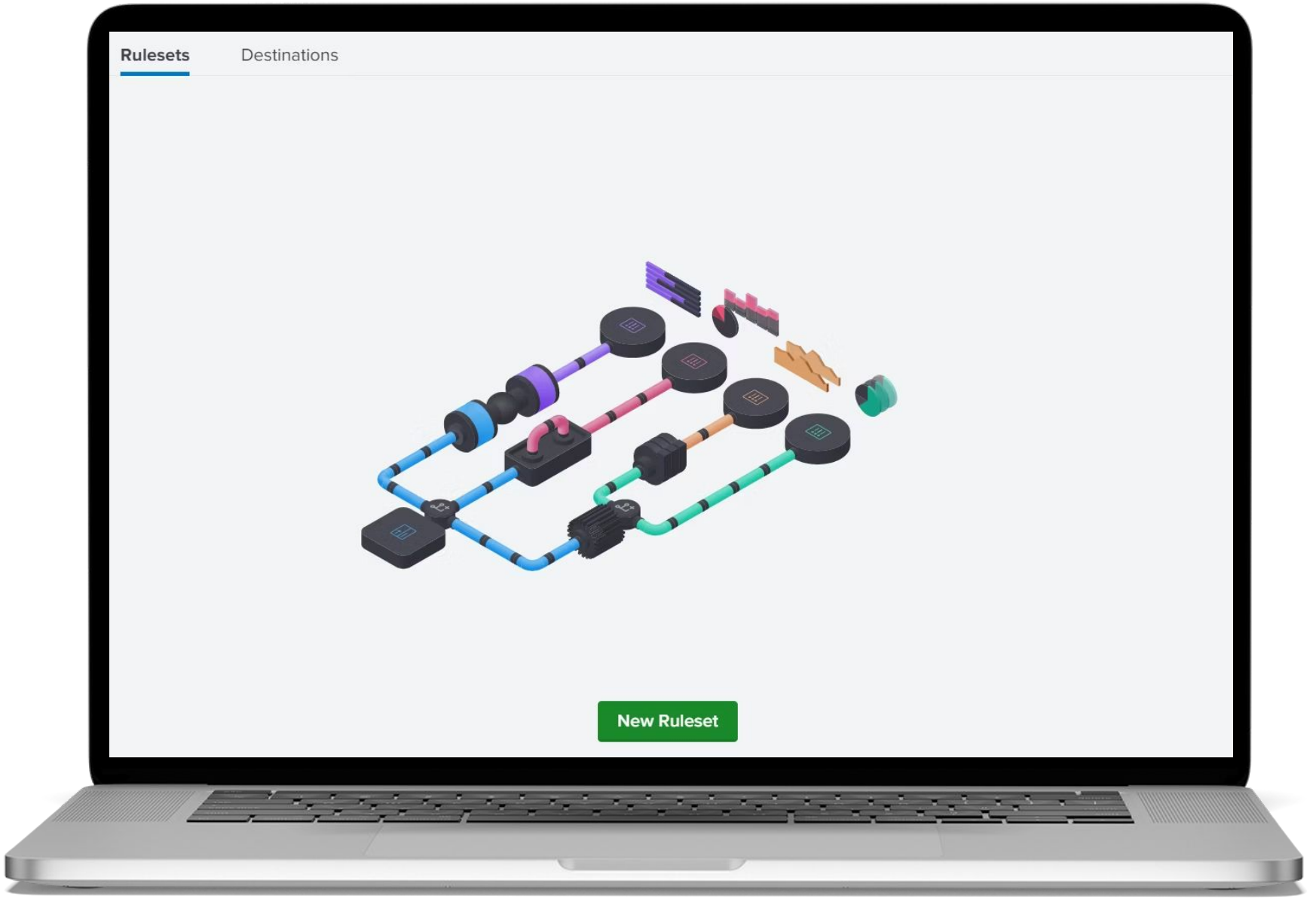
Change the content of events

Route



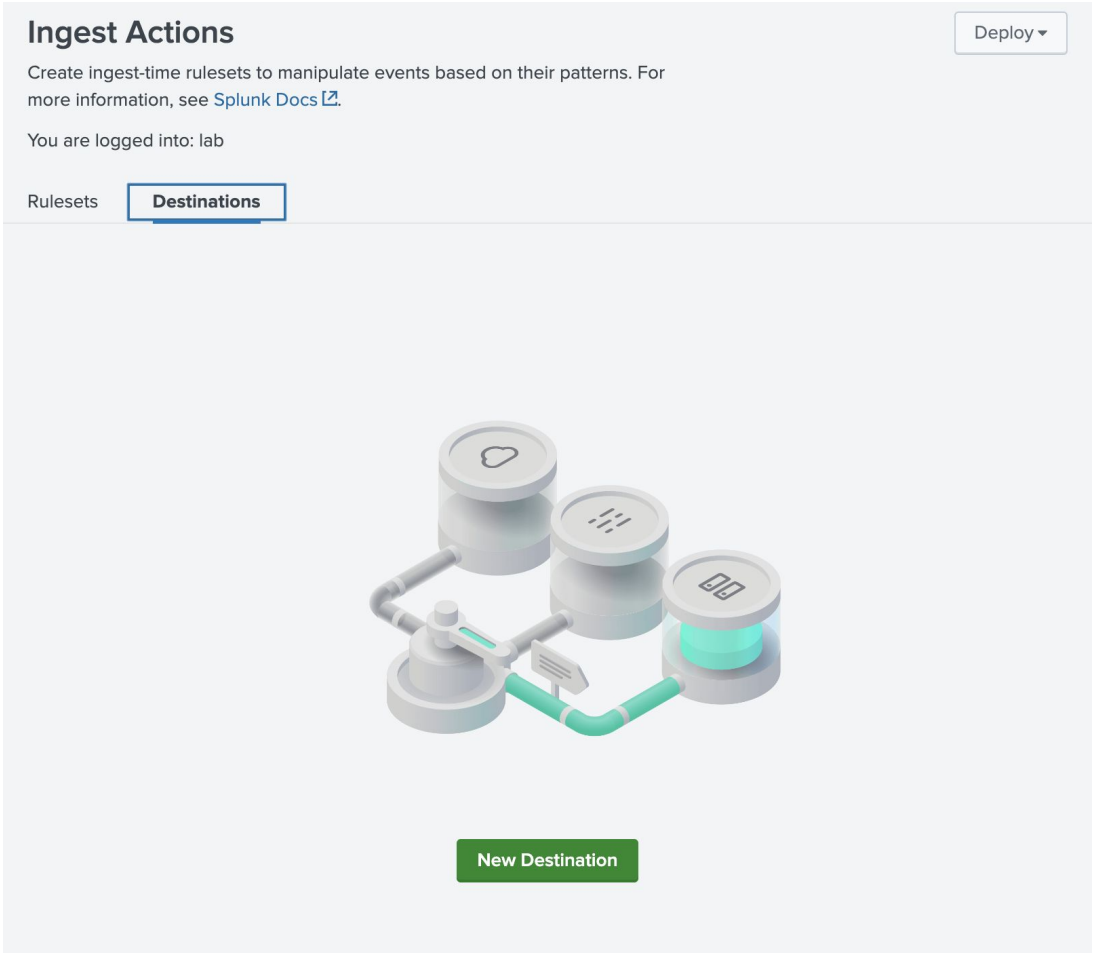
Determine the destination of events

Here we go!



Ingest Actions: Destinations

Route



- Write to s3 bucket(s)
- And more…

Destination Configs

outputs.conf - s3

[rfs:my_s3_location]

path = s3://<bucket>/<folder>

remote.s3.endpoint = <url>

remote.s3.access_key = <key>

remote.s3.secret_key = <secret>

outputs.conf - file

[rfs:my_file_location]

path = file:///opt/data/ingest-actions/

compression = none

100% Unsupported!

What Does It Look Like?

```
lab:/data/01/2023 # tree | head
.
└── 04
    └── 28
        ├── events_1682697160_1682697119_1682697163_000000_273EFB7C-9C64-4F57-90B8-73ABBECB02FC.json
        ├── events_1682697184_1682697163_1682697193_000002_273EFB7C-9C64-4F57-90B8-73ABBECB02FC.json
        ├── events_1682697214_1682697191_1682697224_000005_273EFB7C-9C64-4F57-90B8-73ABBECB02FC.json
        ├── events_1682697244_1682697224_1682697255_000006_273EFB7C-9C64-4F57-90B8-73ABBECB02FC.json
        ├── events_1682697283_1682697255_1682697285_000008_273EFB7C-9C64-4F57-90B8-73ABBECB02FC.json
        ├── events_1682697313_1682697285_1682697316_000010_273EFB7C-9C64-4F57-90B8-73ABBECB02FC.json
        ├── events_1682697346_1682697316_1682697349_000013_273EFB7C-9C64-4F57-90B8-73ABBECB02FC.json
```

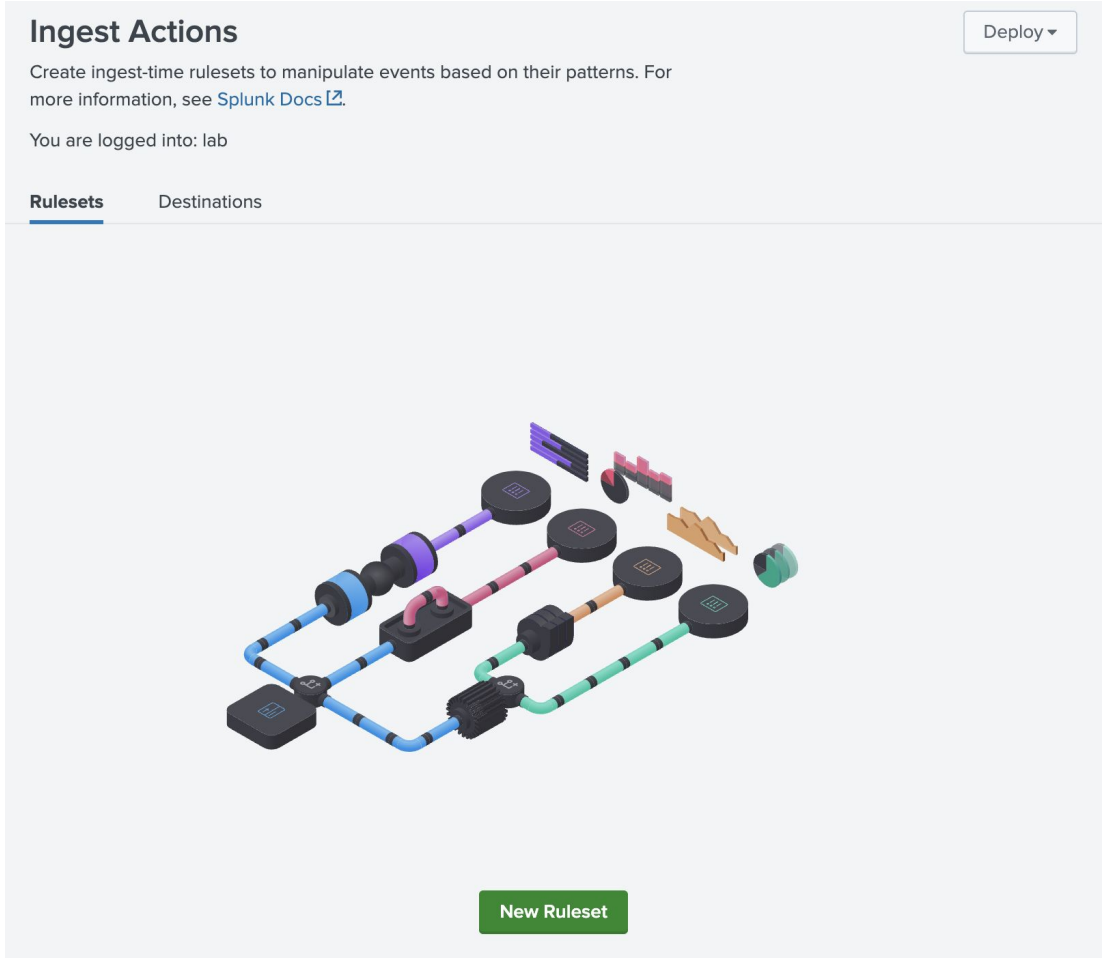
Ready to use .json files ready for your imagination!

What Does the Data Look Like?


```
 {
   "time": 1682697938.024,
   "event": "04-28-2023 12:05:38.024 -0400 INFO  Metrics - group=pipeline, name=typing, processor=sendout,
 cpu_seconds=0.000, executes=452, cumulative_hits=15276",
   "host": "lab",
   "source": "/opt/splunk/var/log/splunk/metrics.log",
   "sourcetype": "splunkd"
 },
 {
   "time": 1682697938.024,
   "event": "04-28-2023 12:05:38.024 -0400 INFO  Metrics - group=pipeline, name=typing, processor=tee, cpu
_seconds=0.000, executes=263, cumulative_hits=10632",
   "host": "lab",
   "source": "/opt/splunk/var/log/splunk/metrics.log",
   "sourcetype": "splunkd"
 },
```

Ingest Actions: Rulesets

Filter/Mask



- Mask with Regular Expressions
- Filter using Regular Expressions
- Filter using Eval Expressions
- Set a Field (Index)
- And **more…**

Where Are IA Configurations Written?

When using the Web UI

Cluster Manager

$SPLUNK_HOME/etc/manager-apps/splunk_ingest_actions

Standalone (incl. HF)

$SPLUNK_HOME/etc/apps/splunk_ingest_actions

Deployment Server

$SPLUNK_HOME/etc/deployment-apps/splunk_ingest_actions

Ruleset Configs

props.conf

- **RULESET-**
  Works the same as TRANSFORMS-* class, but will run transforms on parsed data
- **RULESET_DESC-**
  Description of ruleset

transforms.conf

- **INGEST_EVAL =**
  Very similar to “| eval”
- **STOP_PROCESSING_IF =**
  Comparable to a “break” if the RULESET was a “loop”

Destination + Ruleset Configs

Writing splunkd logs to “rfs:my_file_location”

props.conf

[splunkd]

RULESET-route_to_file_example = _rule:route_to_file_example:route:eval:r5ba3e1g

RULESET_DESC-route_to_file_example =

transforms.conf

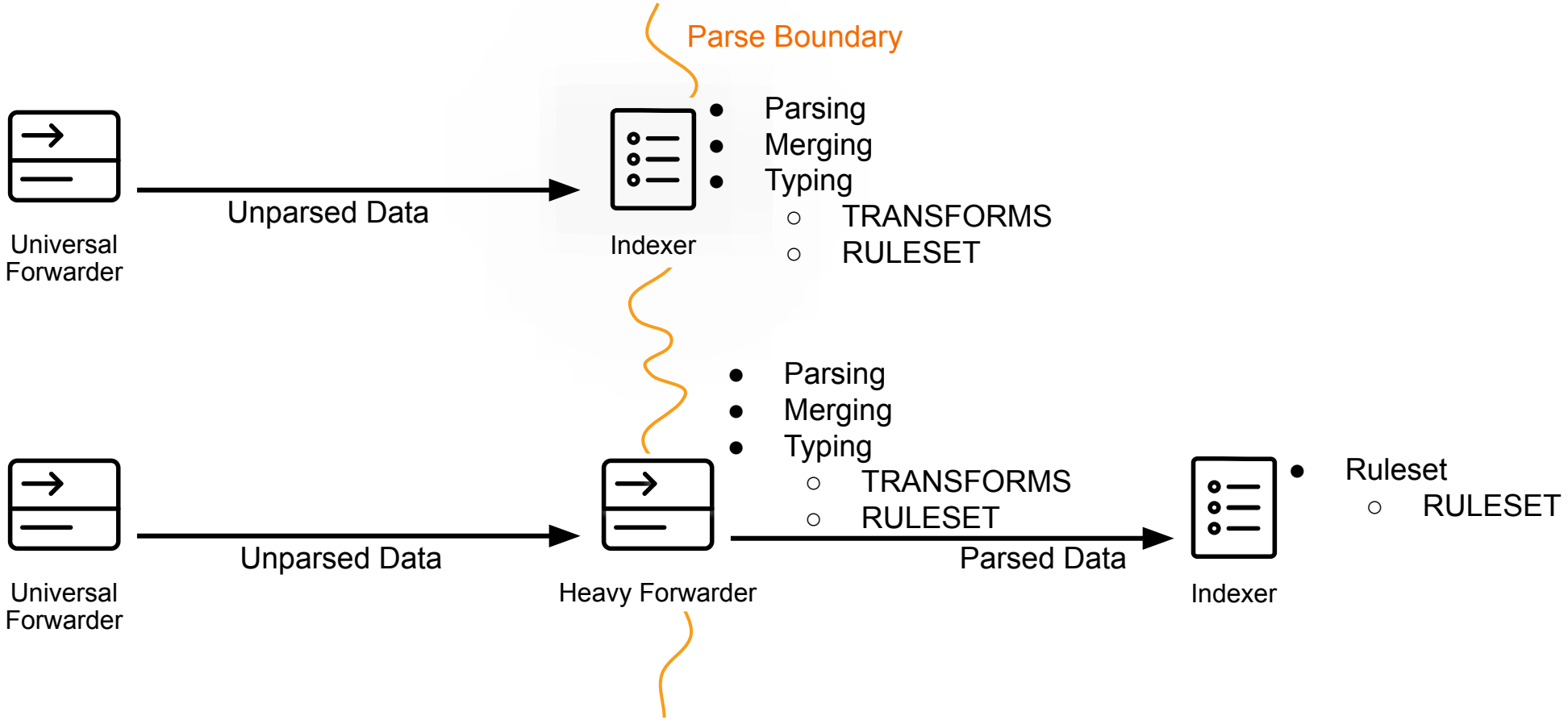
[_rule:route_to_file_example:route:eval:r5ba3e1g]

INGEST_EVAL = 'pd:_destinationKey'=if((true()), "_splunk_,rfs:my_file_location", 'pd:_destinationKey')

STOP_PROCESSING_IF = NOT isnull('pd:_destinationKey') AND 'pd:_destinationKey' != "" AND (isnull('pd:_doRouteClone') OR 'pd:_doRouteClone' == "")

Where Do Rulesets Execute?

Ingest Action Rulesets are executed after existing transforms, e.g. TAs



Order of Operations

123abc

Numbers are sorted before letters.
Numbers are sorted based on the first digit.
Uppercase letters are sorted before lowercase letters.

1. Location in “Parse Boundary”
2. Props stanza precedence
   a. [<sourcetype>], [host::<host>], [source::<source>]
3. All **TRANSFORMS**, alphabetically
   a. Within a single class set, list order
4. All **RULESETS**, alphabetically
   a. Within a single class set, list order

[<sourcetype>]

TRANSFORMS-colors = yellow, blue, red

TRANSFORMS-pets = cat, dog, fish

RULESET-colors = yellow, blue, red

RULESET-pets = cat, dog, fish

STOP_PROCESSING_IF

while true, do, break

props.conf

[<stanza>]

RULESET-ruleset1 = rule1, rule2, …

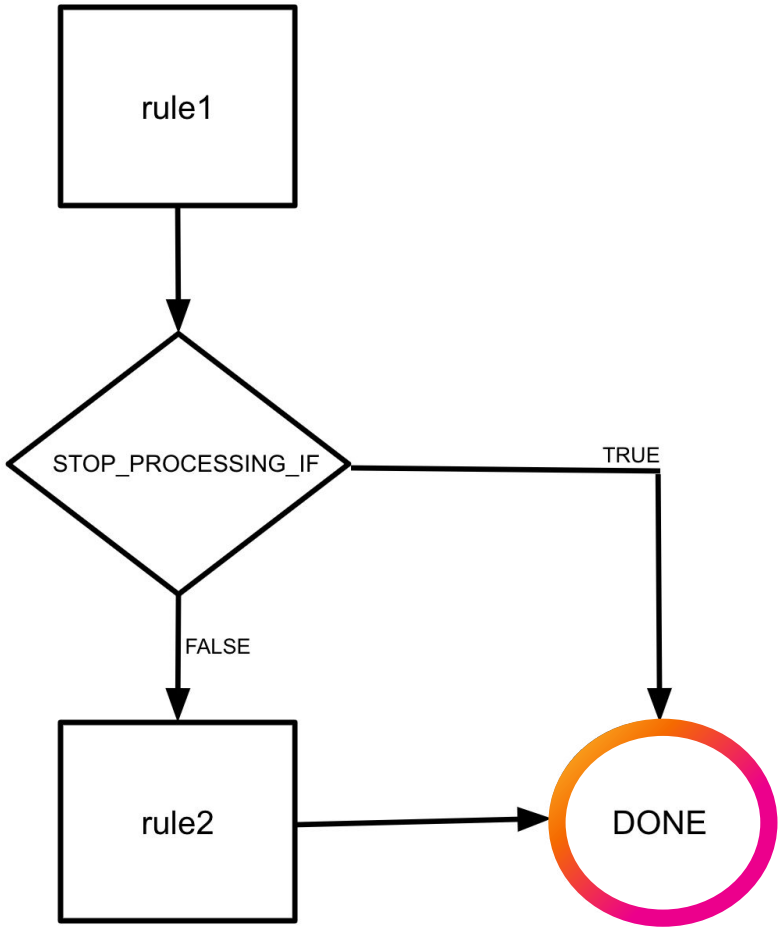
transforms.conf

[rule1]

STOP_PROCESSING_IF = <expression1>

[rule2]

STOP_PROCESSING_IF = <expression2>



Warranty Void

Consult with Professional Services

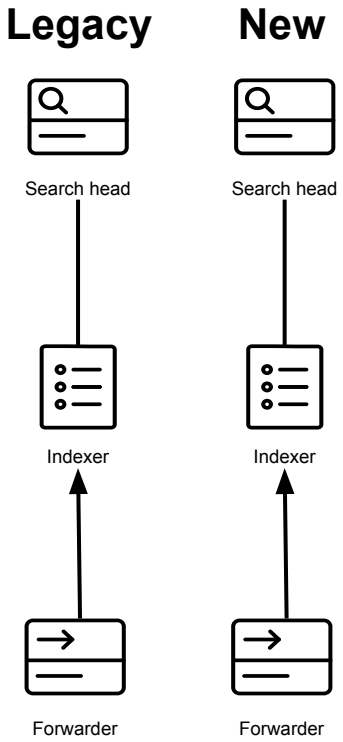
splunk> .conf23

Customer Story 1

The Scenario

One Splunk Environment to Another

- Customer is migrating a **legacy** environment **into** the **new** “corporate” environment
- Legacy environment is the **wild-wild-west** with no data quality standards
- New “corporate” environment has **very strict** data quality standards
- Can NOT break existing dashboards/monitoring until data and content is migrated
- What do we do?

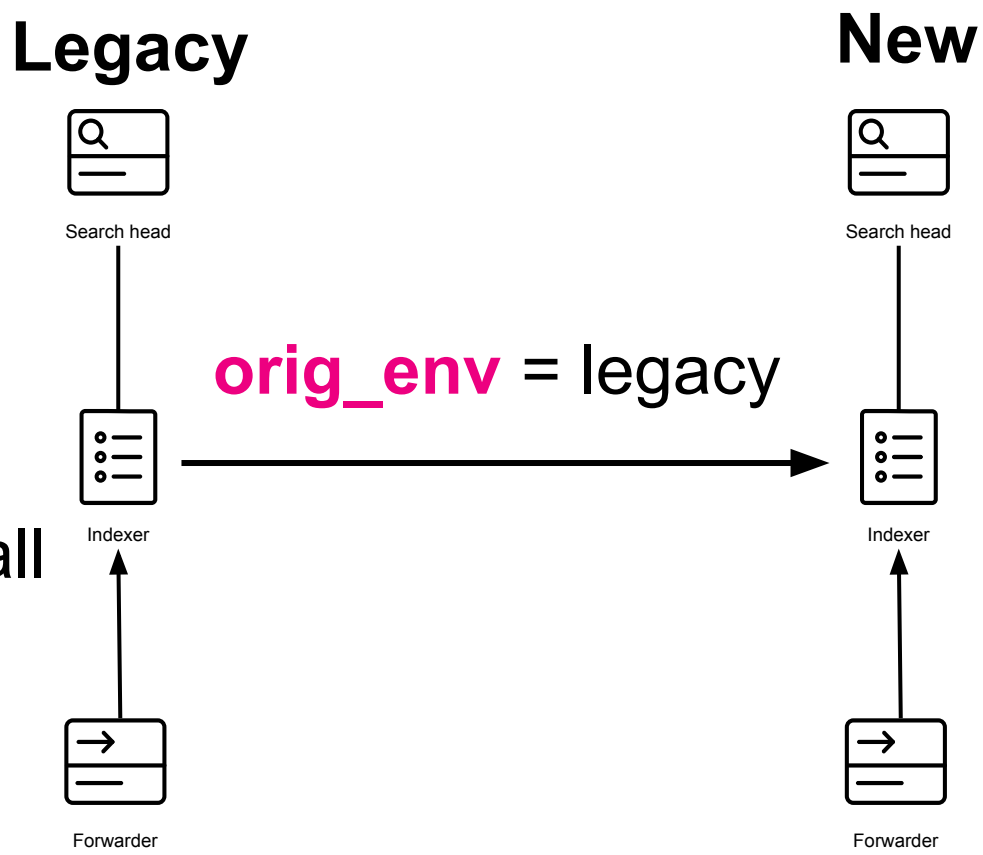


Before Ingest Actions

- Change sourcetypes/indexes in the legacy environment and break existing content?
- Forklift bad data and content into the new environment?
- Do it all after hours/weekend?

Using Ingest Actions

index = network
sourcetype = firewall



orig_env = legacy

orig_index = network
orig_sourcetype = firewall

index = firewall
sourcetype = cisco:asa

The Configs

Legacy Environment

props.conf

[host::*]

RULESET-z = _rule:set_orig_env

transforms.conf

[_rule:set_orig_env]

INGEST_EVAL = orig_env="legacy"

outputs.conf

[tcpout]

indexAndForward = true

defaultGroup = new

New Environment

props.conf

[firewall]

RULESET-1 = _rule:set_orig_index_st

RULESET-2 = _rule:set_fixed_firewall_st

transforms.conf

[_rule:set_orig_index_st]

INGEST_EVAL = orig_index=index, orig_sourcetype=sourcetype

[_rule:set_fixed_firewall_st]

INGEST_EVAL = index="firewall", sourcetype="cisco:asa"

What You Get

i	Time	Event
⌄	6/2/23 8:56:28.000 AM	June 02 2023 08:56:28 localhost %ASA-4-106023: Allow TCP src inbound:100.64.58.17/8890 dst restrict:192.168.50.59/8080 by access-group "inbound" [0x0, 0x0]

Event Actions ▾

Type	Field	Value	Actions
Selected	☑ host ▾	localhost	⌄
	☑ source ▾	firewall.txt	⌄
	☑ sourcetype ▾	cisco:asa	⌄
Event	☐ orig_env ▾	legacy	⌄
	☐ orig_index ▾	network	⌄
	☐ orig_sourcetype ▾	firewall	⌄
Time ⊕	_time ▾	2023-06-02T08:56:28.000-03:00	
Default	☐ index ▾	firewall	⌄

Customer Story 2

splunk> .conf23

The Scenario

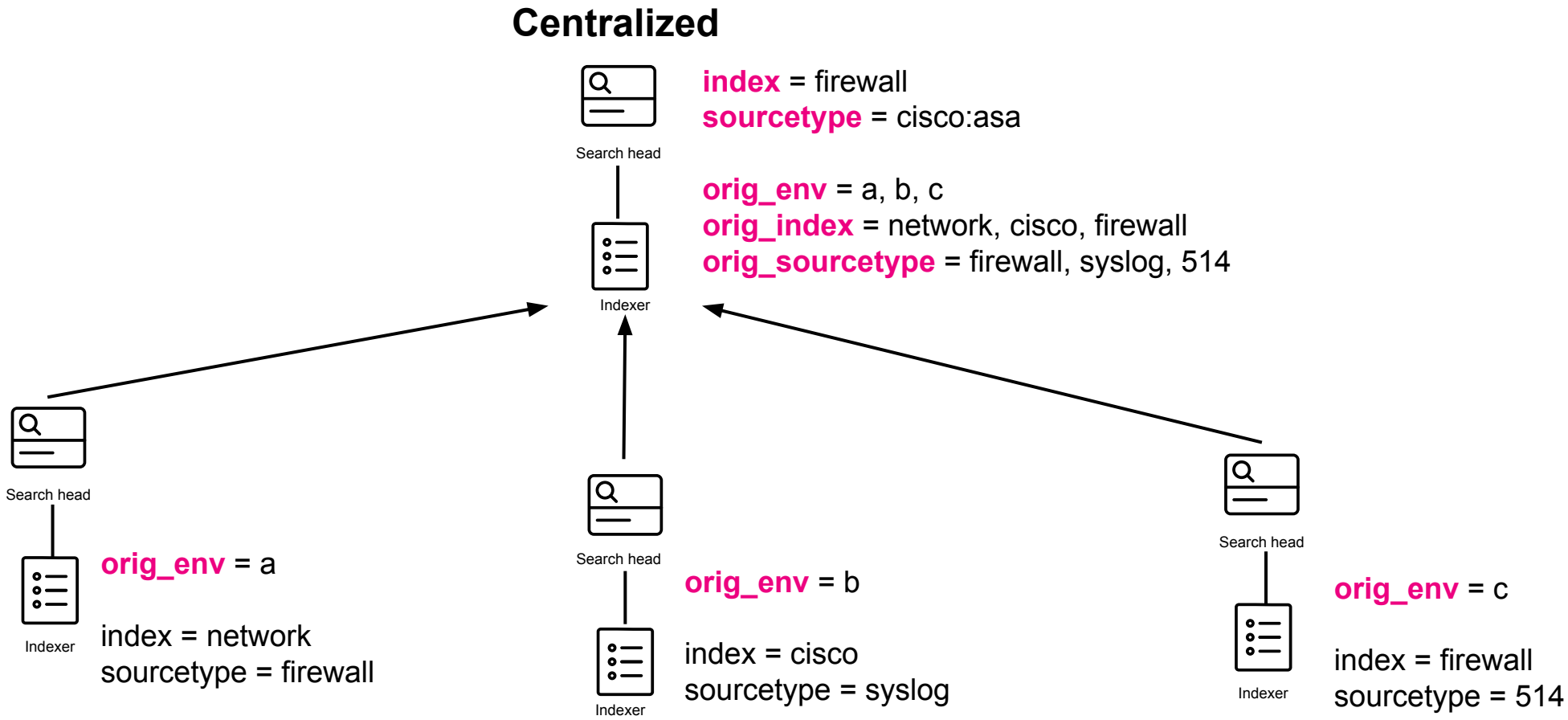
Many Splunks to One

- You have multiple Splunk silos
- "Centralized" site needs to be able to "search" these silos
- Silos have no governance from retention, data quality, change control
- The wild wild west

Before Ingest Actions

- Cry & run away
- Brew a gallon of espresso
- Blame the Intern

Using Ingest Actions



"Silo" Configs

Silo "A"

props.conf

[host::*]

RULESET-z = _rule:set_orig_env

transforms.conf

[_rule:set_orig_env]

INGEST_EVAL = orig_env="a"

outputs.conf

[tcpout]

indexAndForward = true

defaultGroup = consolidated

Silo "B"

props.conf

[host::*]

RULESET-z = _rule:set_orig_env

transforms.conf

[_rule:set_orig_env]

INGEST_EVAL = orig_env="b"

outputs.conf

[tcpout]

indexAndForward = true

defaultGroup = consolidated

Silo "C"

props.conf

[host::*]

RULESET-z = _rule:set_orig_env

transforms.conf

[_rule:set_orig_env]

INGEST_EVAL = orig_env="c"

outputs.conf

[tcpout]

indexAndForward = true

defaultGroup = consolidated

"Centralized" Configs - orig_*

props.conf

[firewall]

RULESET-1 = _rule:set_orig_index, _rule:set_orig_sourcetype

[syslog]

RULESET-1 = _rule:set_orig_index, _rule:set_orig_sourcetype

[514]

RULESET-1 = _rule:set_orig_index, _rule:set_orig_sourcetype

transforms.conf

[_rule:set_orig_index]

INGEST_EVAL = orig_index=if(isnull(orig_env), null()), index)

[_rule:set_orig_sourcetype]

INGEST_EVAL = orig_sourcetype=if(isnull(orig_env), null()), sourcetype)

"Centralized" Configs - Fixing Silo A

props.conf

[firewall]

RULESET-2 = _rule:fix_silo_a_index,
_rule:fix_silo_a_sourcetype

transforms.conf

[_rule:fix_silo_a_index]

INGEST_EVAL = index=if(orig_env="a" AND orig_index="network"), "firewall", index)

[_rule:fix_silo_a_sourcetype]

INGEST_EVAL = sourcetype=if(orig_env="a" AND orig_index="network"), "cisco:asa", sourcetype)

"Centralized" Configs - Fixing Silo B

props.conf

```
[syslog]

RULESET-2 = _rule:fix_silo_b_index,
_rule:fix_silo_b_sourcetype
```

transforms.conf

```
[_rule:fix_silo_b_index]

INGEST_EVAL = index=if(orig_env="b" AND
orig_index="cisco"), "firewall", index)

[_rule:fix_silo_b_sourcetype]

INGEST_EVAL = sourcetype=if(orig_env="b"
AND orig_index="cisco"), "cisco:asa", sourcetype)
```

"Centralized" Configs - Fixing Silo C

props.conf

[514]

RULESET-2 = _rule:fix_silo_c_index, _rule:fix_silo_c_sourcetype

transforms.conf

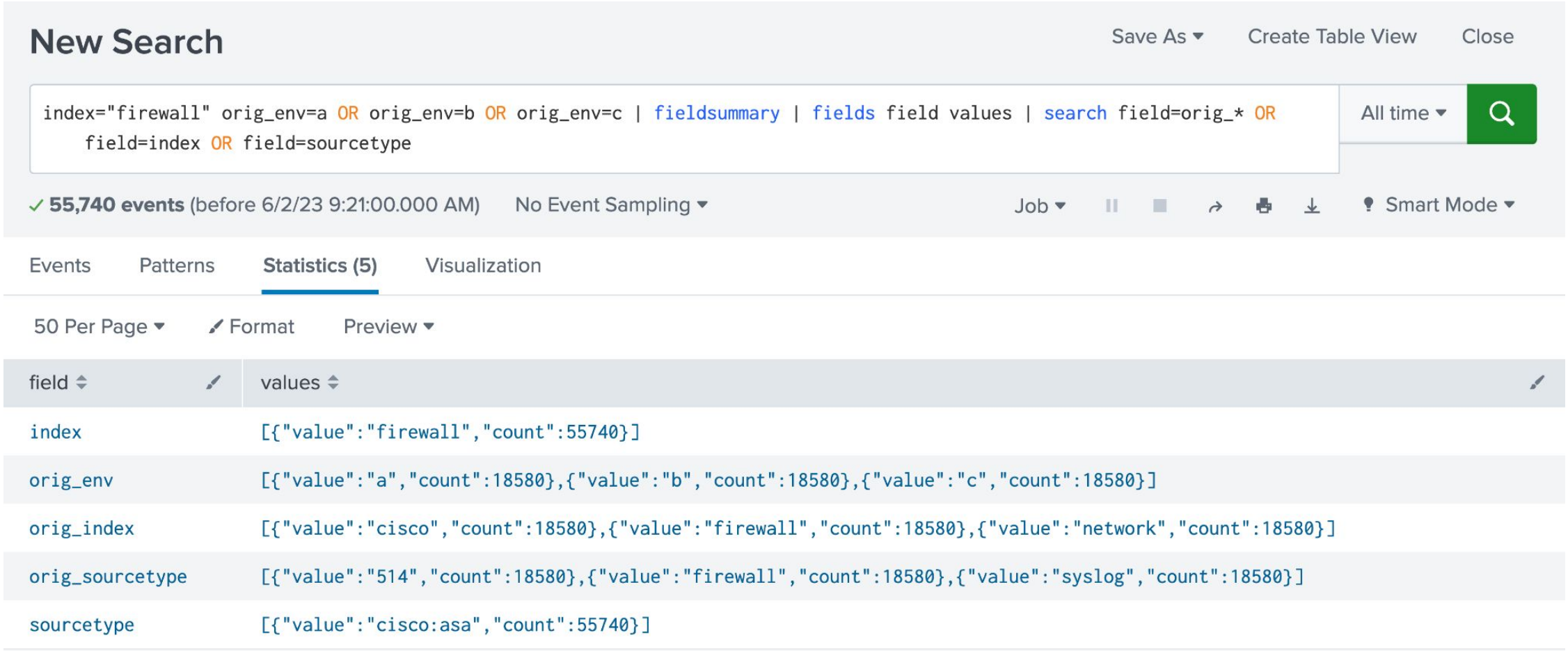
[_rule:fix_silo_c_index]

INGEST_EVAL = index=if(orig_env="c" AND orig_index="firewall", "firewall", index)

[_rule:fix_silo_c_sourcetype]

INGEST_EVAL = sourcetype=if(orig_env="c" AND orig_index="firewall", "cisco:asa", sourcetype)

What You Get



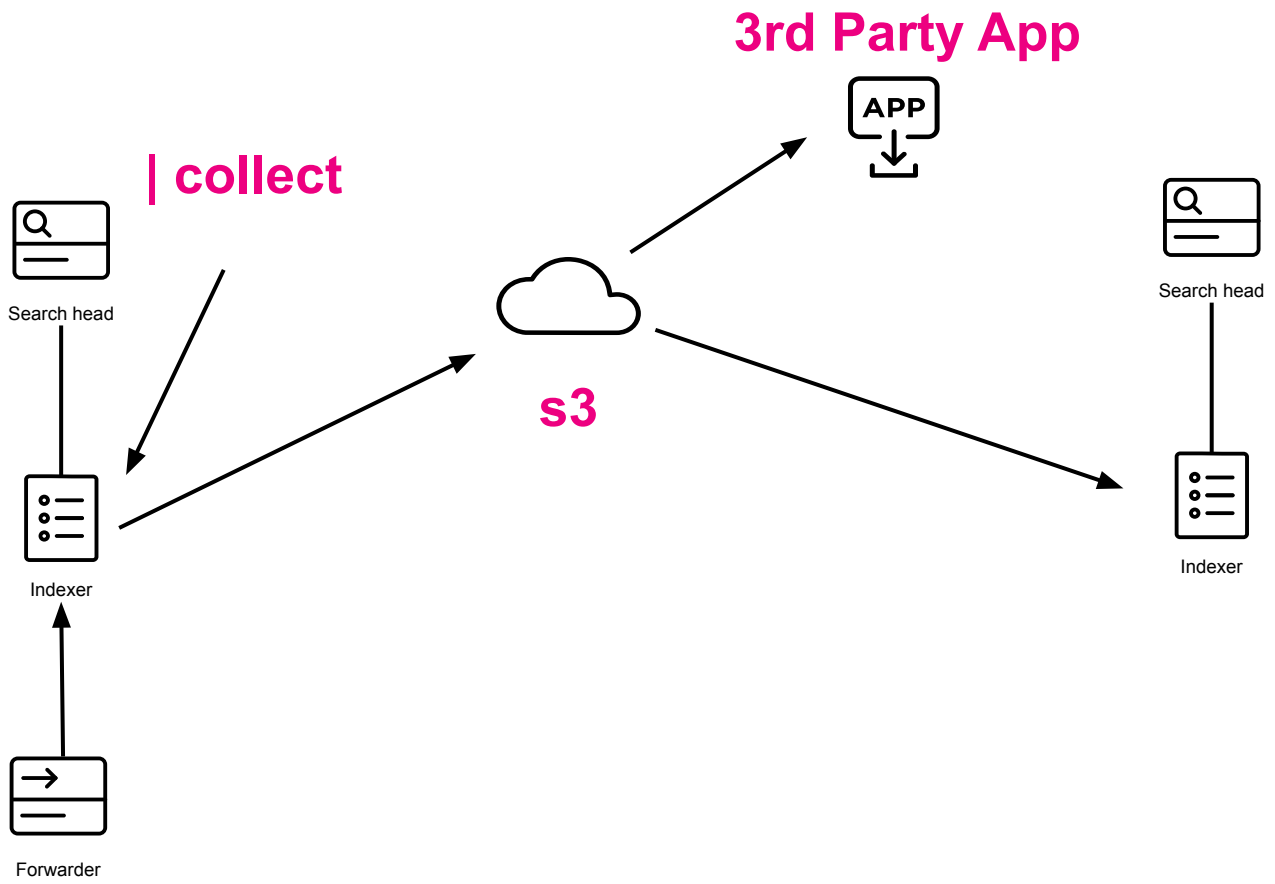
Customer Story 3

The Scenario

Airgap/Aggregation

- Environments can not communicate with each other
- Customer would like “summary” data sent to s3 bucket
- Use IA to push “**| collect**” summaries to s3 bucket
- Reindexed summaries using AWS S3 Modular Input
- Data is also available for third party applications

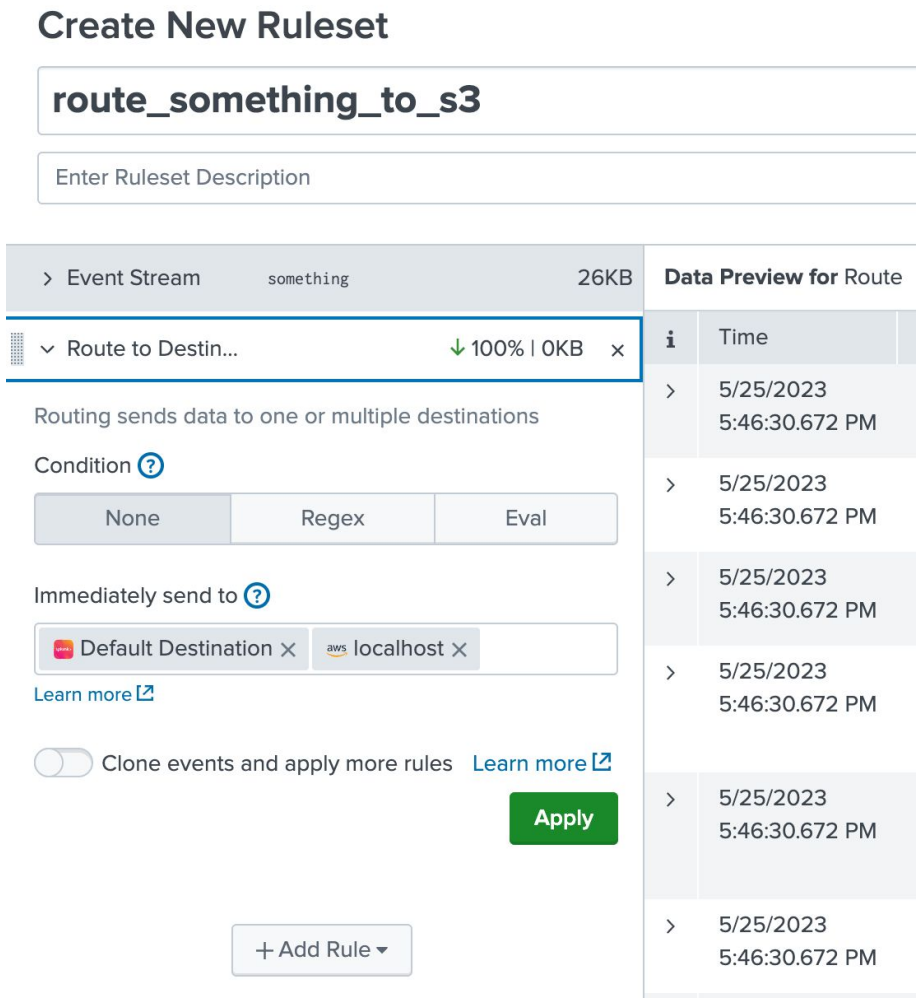
Architecture



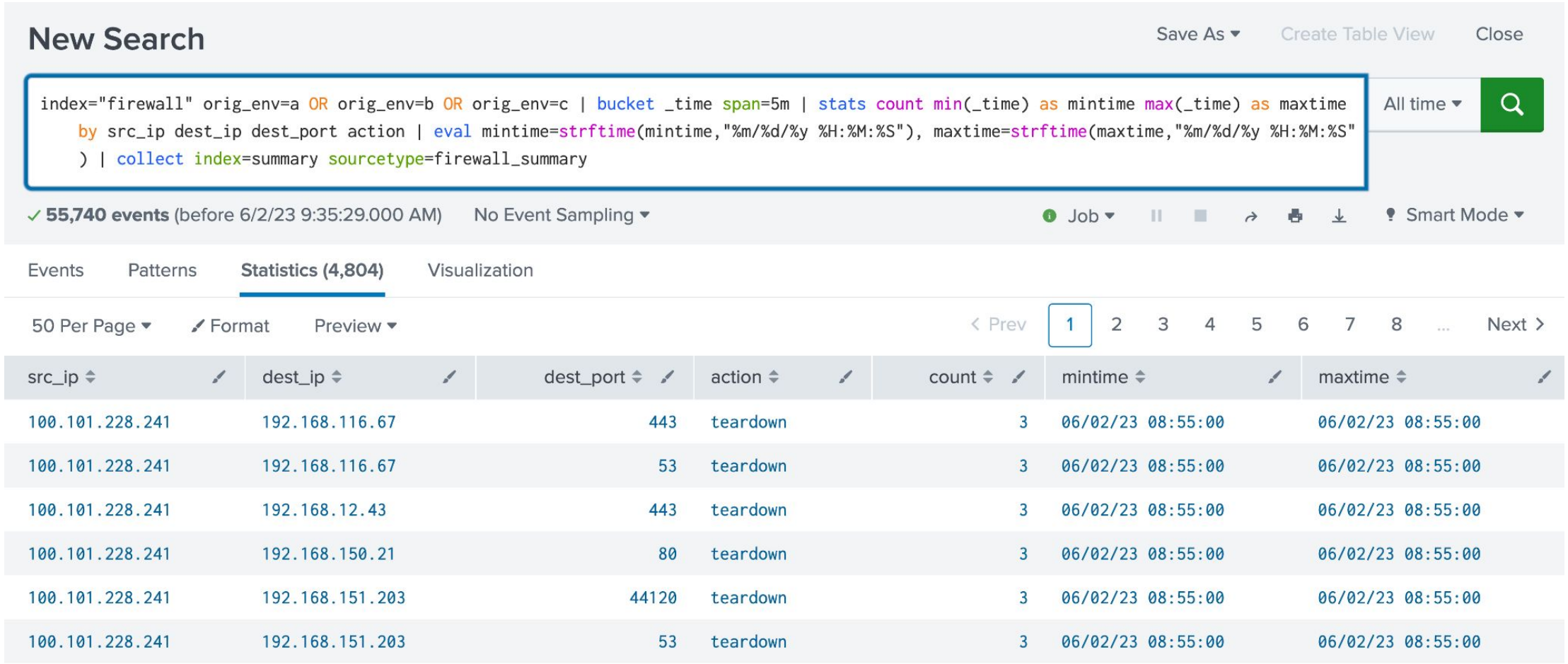
The Configs

Using the WebUI

- Use Summary Index or the "**| collect**" command to summarize
- index=network | stats count by src dest **| collect** index=summary **sourcetype=something**
- Use IA WebUI to create a "Route to Destination" rule for the output sourcetype



Summarizing



src_ip	dest_ip	dest_port	action	count	mintime	maxtime
100.101.228.241	192.168.116.67	443	teardown	3	06/02/23 08:55:00	06/02/23 08:55:00
100.101.228.241	192.168.116.67	53	teardown	3	06/02/23 08:55:00	06/02/23 08:55:00
100.101.228.241	192.168.12.43	443	teardown	3	06/02/23 08:55:00	06/02/23 08:55:00
100.101.228.241	192.168.150.21	80	teardown	3	06/02/23 08:55:00	06/02/23 08:55:00
100.101.228.241	192.168.151.203	44120	teardown	3	06/02/23 08:55:00	06/02/23 08:55:00
100.101.228.241	192.168.151.203	53	teardown	3	06/02/23 08:55:00	06/02/23 08:55:00

The Summarized Data

Now residing on s3


```
{
  "time": 1685709369,
  "event": "06/02/2023 09:36:09 -0300, info_search_time=1685709369.495,
count=3, action=teardown, src_ip=\"100.98.91.102\", dest_ip=\"192.168.36.143\",
maxtime=\"06/02/23 08:55:00\", mintime=\"06/02/23 08:55:00\", dest_port=443",
  "host": "lab",
  "source": "...events.stash_new",
  "sourcetype": "firewall_summary"
}
```

Reusing the Summarized Data

With Splunk!

i	Time	Event
>	6/2/23 9:36:09.000 AM	06/02/2023 09:36:09 -0300, info_search_time=1685709369.495, count=3, action=teardown, src_ip="100.98.91.102", dest_ip="192.168.50.59", maxtime="06/02/23 08:55:00", mintime="06/02/23 08:55:00", dest_port=9200 host = summarized source = events_1685709369_1685709369_1685709403_000000_62C59AB2-C78B-4... sourcetype = _json
>	6/2/23 9:36:09.000 AM	06/02/2023 09:36:09 -0300, info_search_time=1685709369.495, count=3, action=teardown, src_ip="100.98.91.102", dest_ip="192.168.36.143", maxtime="06/02/23 08:55:00", mintime="06/02/23 08:55:00", dest_port=443 host = summarized source = events_1685709369_1685709369_1685709403_000000_62C59AB2-C78B-4... sourcetype = _json
>	6/2/23 9:36:09.000 AM	06/02/2023 09:36:09 -0300, info_search_time=1685709369.495, count=3, action=teardown, src_ip="100.98.91.102", dest_ip="192.168.35.22", maxtime="06/02/23 08:55:00", mintime="06/02/23 08:55:00", dest_port=443 host = summarized source = events_1685709369_1685709369_1685709403_000000_62C59AB2-C78B-4... sourcetype = _json
>	6/2/23 9:36:09.000 AM	06/02/2023 09:36:09 -0300, info_search_time=1685709369.495, count=3, action=teardown, src_ip="100.98.91.102", dest_ip="192.168.35.205", maxtime="06/02/23 08:55:00", mintime="06/02/23 08:55:00", dest_port=22 host = summarized source = events_1685709369_1685709369_1685709403_000000_62C59AB2-C78B-4... sourcetype = _json
>	6/2/23 9:36:09.000 AM	06/02/2023 09:36:09 -0300, info_search_time=1685709369.495, count=3, action=teardown, src_ip="100.98.91.102", dest_ip="192.168.26.86", maxtime="06/02/23 08:55:00", mintime="06/02/23 08:55:00", dest_port=443 host = summarized source = events_1685709369_1685709369_1685709403_000000_62C59AB2-C78B-4... sourcetype = _json

splunk> .conf23

Reusing the Summarized Data

With anything!

```
17
18  def main():
19
20      for i in glob("/opt/splunk/ingest-actions/**/*.json"):
21          with open(i, "r") as fin:
22              data = fin.read()
23              for row in json.loads(data):
24                  print(row)
25
26  if __name__ == "__main__":
27      main()
28
```

Filter | All Output

```
{'time': 1685709369, 'event': '06/02/2023 09:36:09 -0300, info_search_time=1685709369.495, count=3,
    action=teardown, src_ip="100.101.228.241", dest_ip="192.168.116.67", maxtime="06/02/23 08:55:00",
    mintime="06/02/23 08:55:00", dest_port=443', 'host': 'lab', 'source': '/opt/splunk/var/spool/splunk/
    10fe3cb05ee081a0_eda28350538dce88_events.stash_new', 'sourcetype': 'firewall_summary'}
{'time': 1685709369, 'event': '06/02/2023 09:36:09 -0300, info_search_time=1685709369.495, count=3,
    action=teardown, src_ip="100.101.228.241", dest_ip="192.168.116.67", maxtime="06/02/23 08:55:00",
    mintime="06/02/23 08:55:00", dest_port=53', 'host': 'lab', 'source': '/opt/splunk/var/spool/splunk/
    10fe3cb05ee081a0_eda28350538dce88_events.stash_new', 'sourcetype': 'firewall_summary'}
{'time': 1685709369, 'event': '06/02/2023 09:36:09 -0300, info_search_time=1685709369.495, count=3,
    action=teardown, src_ip="100.101.228.241", dest_ip="192.168.12.43", maxtime="06/02/23 08:55:00",
    mintime="06/02/23 08:55:00", dest_port=443', 'host': 'lab', 'source': '/opt/splunk/var/spool/splunk/
    10fe3cb05ee081a0_eda28350538dce88_events.stash_new', 'sourcetype': 'firewall_summary'}
```

Customer Story 4

splunk> .conf23

The Scenario

_time > now

- Events are being forwarded from one environment to another using a Heavy Forwarder or Indexer
- You notice _time is incorrect, some events are from the future
- Original environment refuses to change anything

The Configs

props.conf

[<sourcetype>]

RULESET-1 = fix_future_time

transforms.conf

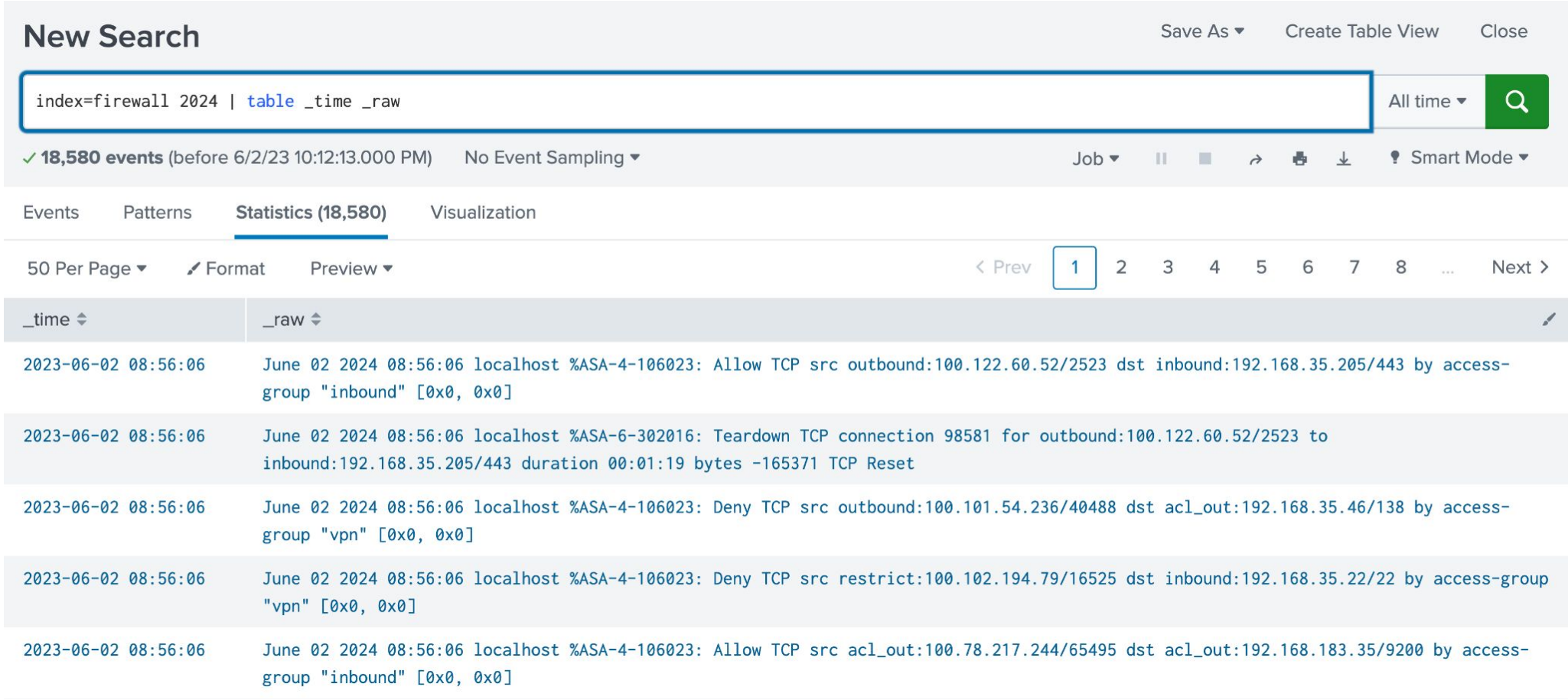
[_rule:fix_future_time]

INGEST_EVAL = _time=if(_time > time(),time(),_time)

Could even use strptime for more accuracy!

What You Get

Without going 88 miles per hour



What's Next?

Can Ingest Actions make your data more usable?

- You understand that **Ingest Actions** consist of **Destinations** and **Rulesets**
- You have seen **Ingest Actions** used with the **WebUI** and **config** files
- You understand **Ingest Actions** use **INGEST_EVALs**
- Do you have an upcoming migration that **Ingest Actions** can make easier?

Thank You

splunk> .conf23