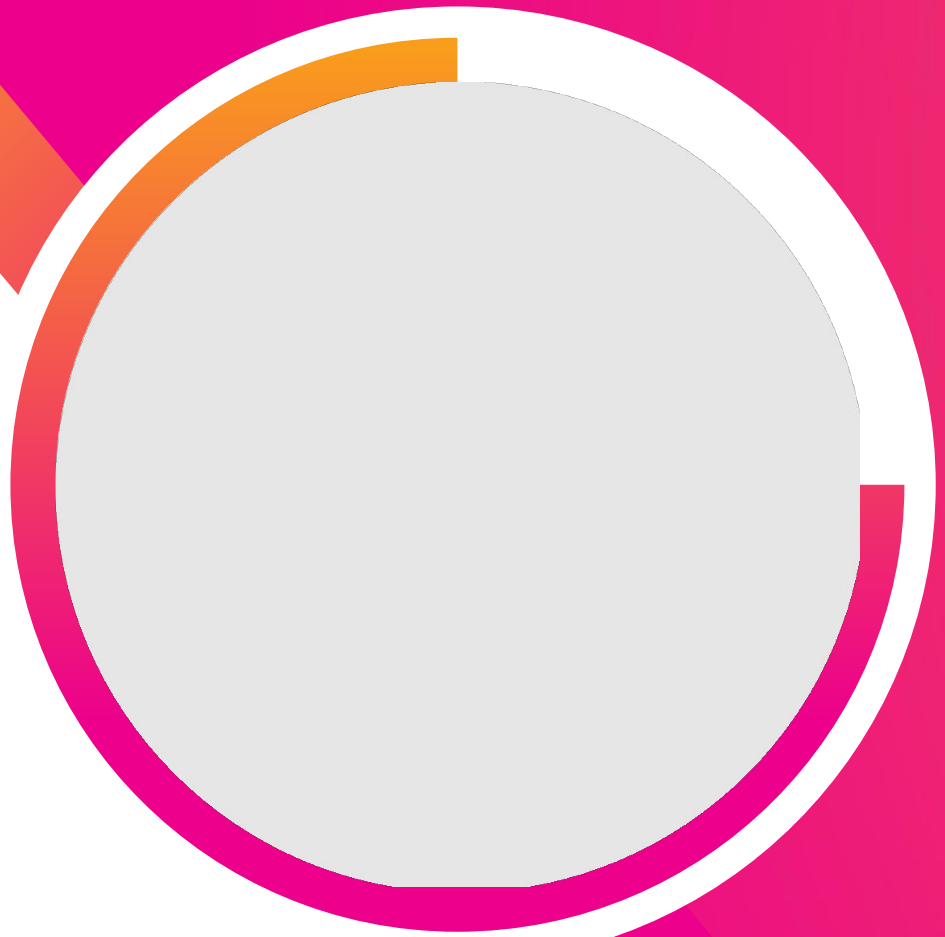
A Prescriptive Design for Enterprise-Wide Alerts in IT Service Intelligence

Matt Hasty, Sr. Engineer, GEHA
Jeff Wiedemann, Sales Engineer, Splunk

Matthew Hasty
Sr. Engineer | GEHA

Jeff Wiedemann
Sales Engineer | Splunk

splunk> .conf19

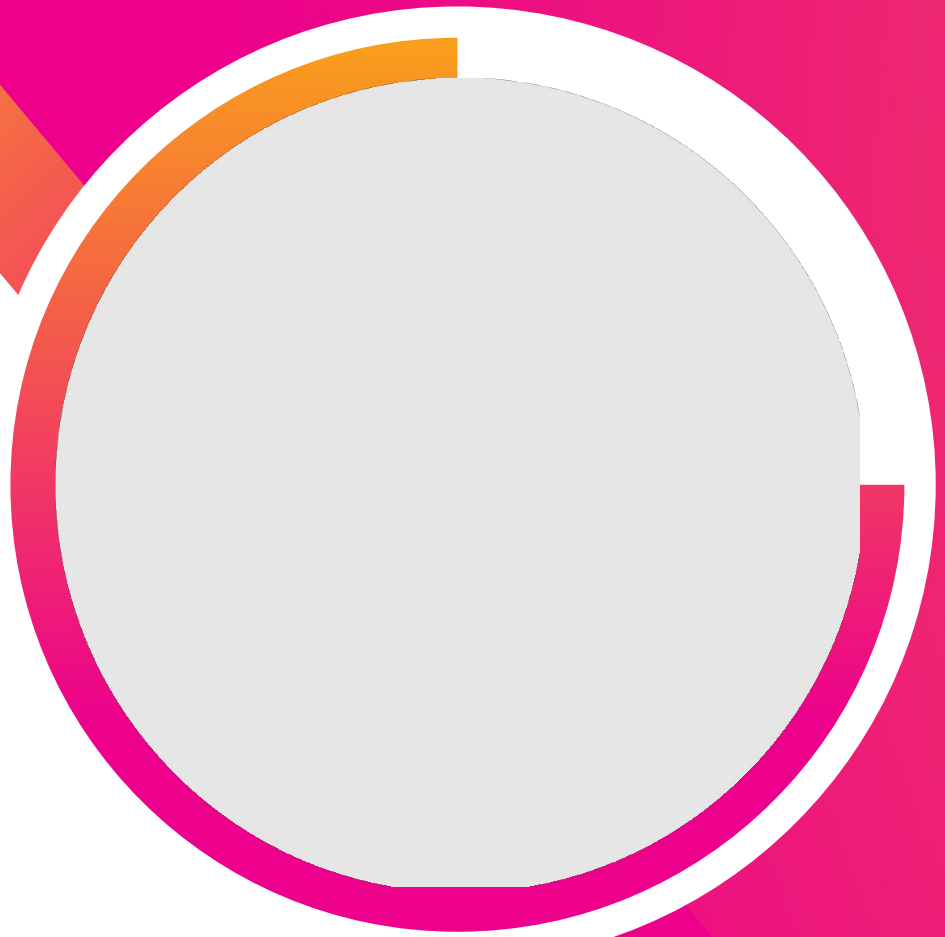


Matthew Hasty

- Sr. Engineer at GEHA
- Started working with Splunk 3 years ago
- Was placed into project to rebuild instance
- Responsible for ITSI/Splunk Enterprise
- Been working in ITSI since 2017
- New Father, baby born August 24th!

GEHA

- 81 Years old
- One of the largest providers of medical/dental plans for Federal Employees
- Not for Profit
- Based in Lee's Summit MO
- Over 2 million members
- 1500 employees



Jeff Wiedemann

- Recovering Software Developer
- IT Service Intelligence Expert
- Blog Series
  - Ensuring Success with ITSI
  - A Blueprint for Splunk ITSI Alerting
- Ask me about thresholding KPIs

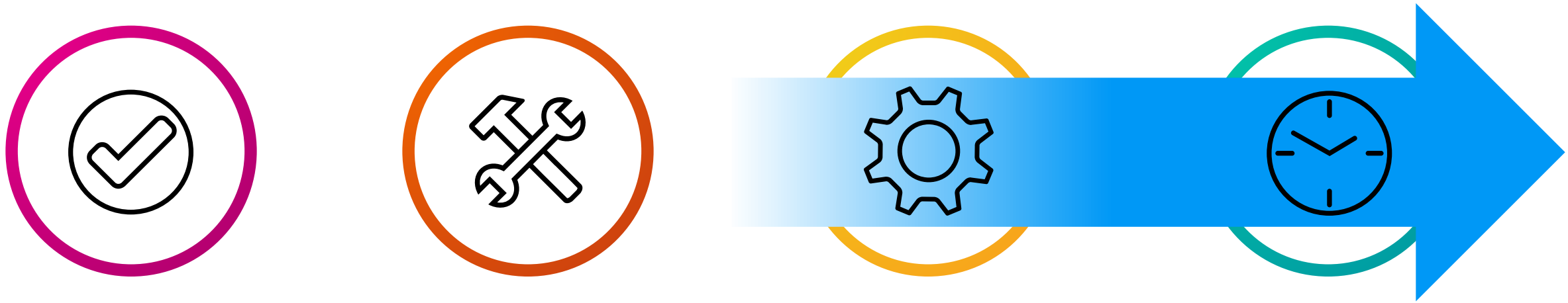
GEHA and SE “Partnership”

GEHA®

splunk>®

ITSI Adoption & Maturity Curve

A 4-Stage Model for Maturity



Getting Started

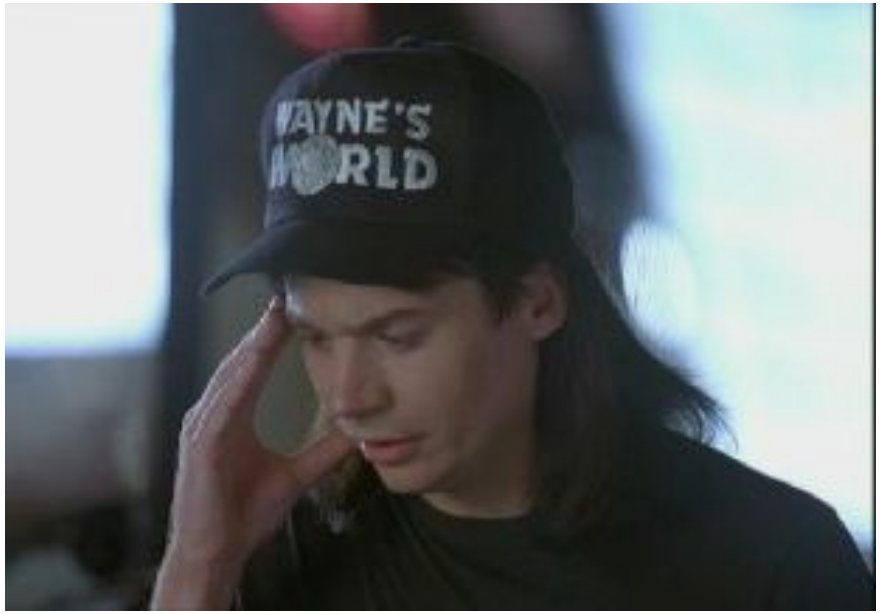
Monitoring the "Important Stuff" Meaningful Services Key or Critical KPIs

KPI Thresholds "Well-Tuned"

Meaningful & Actionable Alerts

Producing Meaningful Alerts Can Be a Headache

- Thresholds must be well tuned
- Service owners don't always know when to alert
- Scaling up is a challenge
- Notables should be grouped to reduce noise
- Best practice guidance is lacking
- Activating an alert is a visceral decision



Our Solution Aims to Solve Much of the Headache

- An alerting design you can copy, customize and enhance
- A scalable, performant, and maintainable strategy
- Enterprise-wide consistency (no snowflakes)
- Next-gen alerting rules

The Two Cornerstone Concepts

- Create notable events for any noteworthy situations
- Apply attributes to your notable events to drive actions

The Five Step Process

- Step 1: Create Initial Notables
- Step 2: Group Related Notables
- Step 3: Create Additional Notables
- Step 4: Add Alerting
- Step 5: Throttle Alerts

Step One: Create Initial Notables

splunk> .conf19

Problem:

- Lots of Services: 150+
- Need to be easily able to ID issues
- Need to easily bring on new Services
- Need to be able to easily maintain
- Multi-KPI alerting: Can get out of hand fast

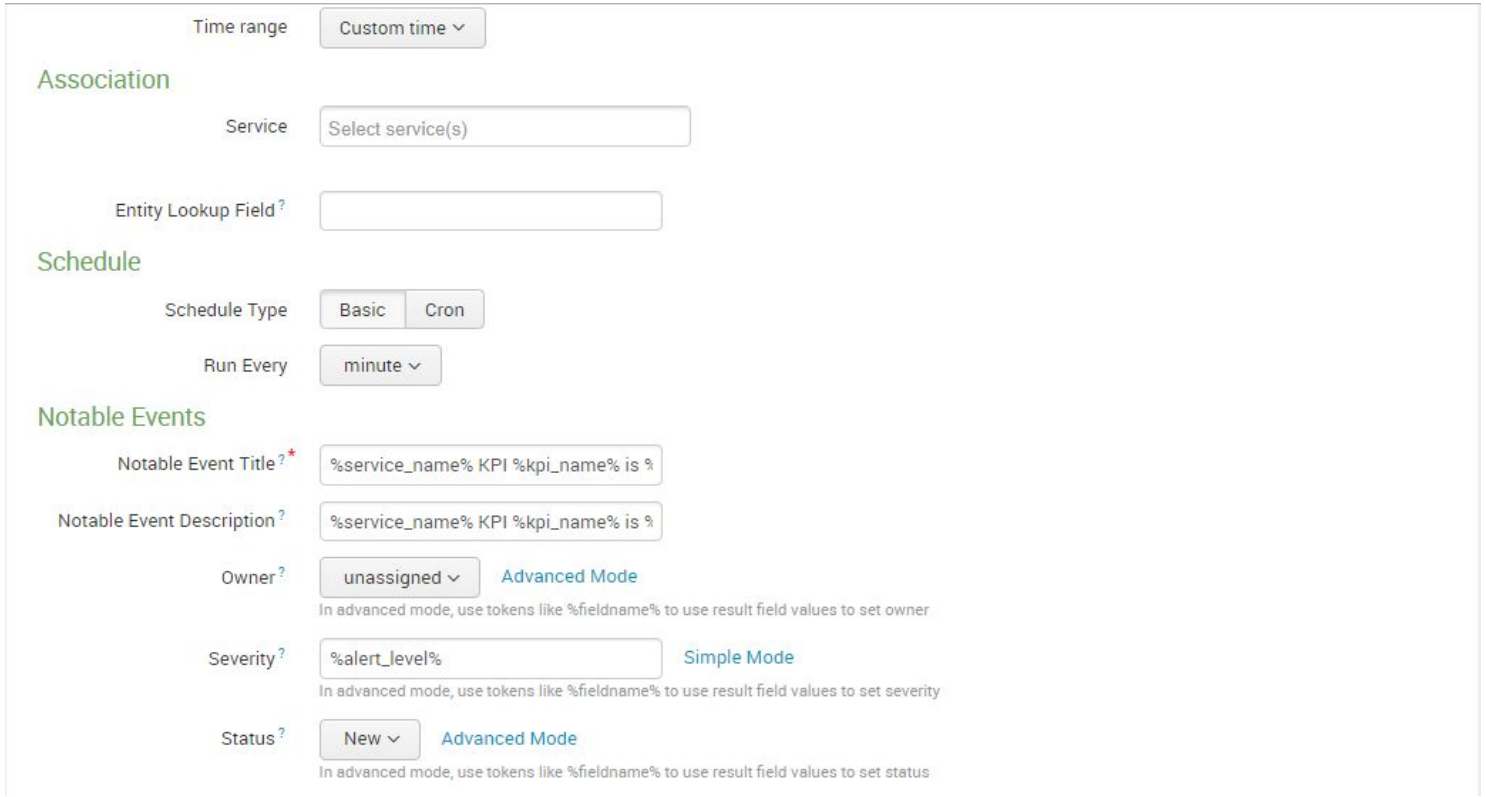
Create the Search

- Copied default search as base
- Wanted to grab KPIs only
- Added in custom fields
- Enriches the rules you use for alert generation
- Custom fields will be written to itsi_summary
- Written fields necessary for NEAPs
- Deduped to prevent excess event generation

```
index="itsi_summary" kpi=* (kpi!="ServiceHealthScore") indexed_is_service_max_severity_event=* alert_level>2
| rename kpiid as itsi_kpi_id
| rename kpi as kpi_name
| eval actual_time=_time
| eval wday=date_wday
| eval hour=date_hour
| convert ctime(actual_time) as actual_time
| eval Orion_Alertable=lower(Orion_Alertable)
| eval geha_alertable=lower(geha_alertable)
| eval alertable = if((alert_level > 5 AND geha_alertable="yes" AND Orion_Alertable="no"),"yes","no")
| dedup itsi_kpi_id, entity_title
```

Create in ITSI

- Set up custom time
  - -10m to -1m
  - Reason: Monitoring Lag
  - Problem: Extra Notables
    - Dedup in correlation will help some with this
    - Have to decide: Extra notables vs missing events
- Set up custom title: Descriptive title and description
  - Makes it easier for your NOC



We Have Descriptive Events!

SITES-[redacted] KPI 4xx Errors Count is medium 4	7/18/2019 4:05:14 PM CDT	Unassigned	Medium	New	SITES-[redacted] KPI 4xx Errors Count is medium 4
SITES-[redacted] KPI 4xx Errors Count is medium 6	7/18/2019 4:05:14 PM CDT	Unassigned	Medium	New	SITES-[redacted] KPI 4xx Errors Count is medium 6
[redacted] KPI Logoff Duration - Profile is medium 65.53611111111111	7/18/2019 4:04:16 PM CDT	Unassigned	Medium	New	[redacted] KPI Logoff Duration - Profile is medium 65.53611111111111
AZURE-EVENT HUB KPI Quota Exceeded Errors is low 93	7/18/2019 4:04:16 PM CDT	Unassigned	Low	New	AZURE-EVENT HUB KPI Quota Exceeded Errors is low 93
SITES-[redacted] KPI Successful Provider Eligibility Check is medium 64	7/18/2019 4:04:16 PM CDT	Unassigned	Medium	New	SITES-[redacted] KPI Successful [redacted] Check is medium 64
AZURE-WEB 2.0 STORAGE KPI Server_Latency is high 703	7/18/2019 4:04:16 PM CDT	Unassigned	High	New	AZURE-WEB 2.0 STORAGE KPI Server_Latency is high 703
AZURE-WEB 2.0 STORAGE KPI Server_Latency is low 415.9464285714286	7/18/2019 4:04:16 PM CDT	Unassigned	Low	New	AZURE-WEB 2.0 STORAGE KPI Server_Latency is low 415.9464285714286
AZURE-WEB 2.0 STORAGE KPI Server_Latency is high 931.875	7/18/2019 4:04:16 PM CDT	Unassigned	High	New	AZURE-WEB 2.0 STORAGE KPI Server_Latency is high 931.875

Step Two:
Group Related Notables

splunk> .conf19

Problem:

- Lots of Events
- Need to be easily able to ID issues
- Need to be able to tie everything together
  - Want to correlate to help make issues easier to see
  - Make life easier on Operations folks

Determining Alert Groups

- Planned out what groups would be
  - What is service watching
  - Who has responsibility for each piece?
    - Each service has KPIs that could be owned by different folks
    - A lot of this depends on how services are set up
    - Some of our services have multiple Departments
  - What is the architecture?
    - Front End vs Backend?
    - Cloud or On Prem?
    - Mixture of both?

Adding in the Alert Group

- Lookup tables!
  - Look up editor App
  - Set alert groups tied to kpiid
  - Set up auto lookup for stash sourcetype
  - Can set any custom properties with this table as well

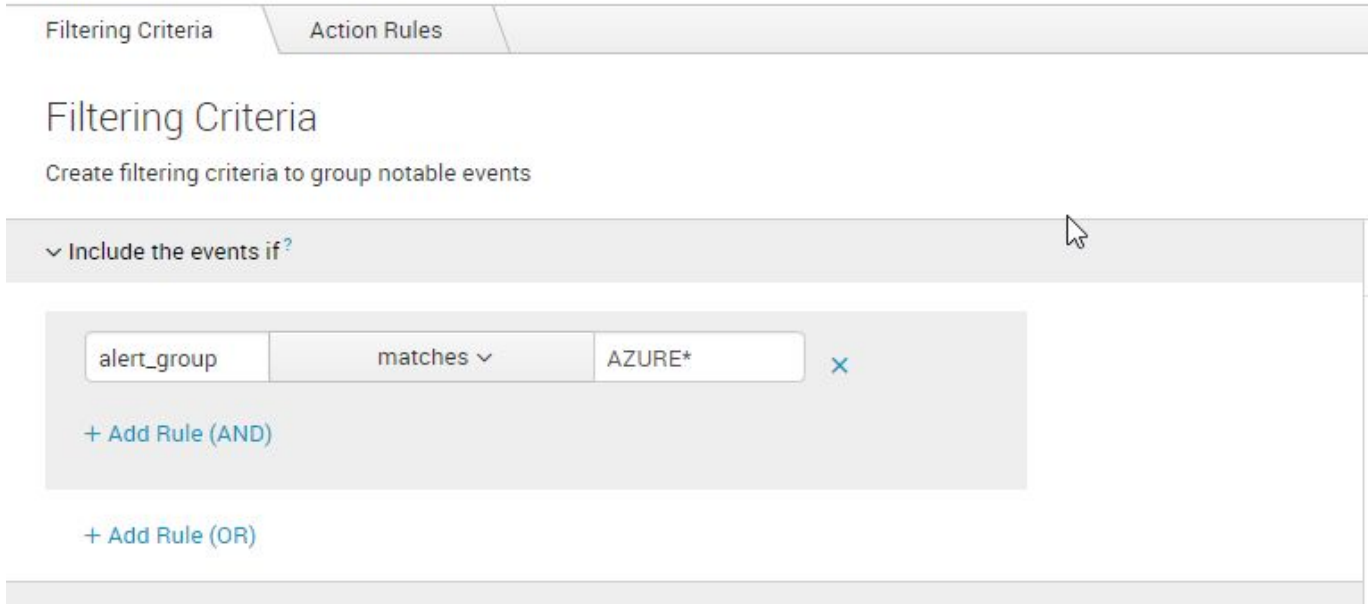
1	service_name	serviceid	kpi_name	kpiid	Department	Orion_Alertable	Sev_Level	geha_alertable	alert_group
94	AZURE [redacted] EXPRESS ROUTE	[redacted]	KB in/sec	[redacted]	COMMS	no	1	yes	AZURE-NETWORK
95	AZURE [redacted] EXPRESS ROUTE		KB out/sec		COMMS	no	1	yes	AZURE-NETWORK
96	AZURE [redacted] EXPRESS ROUTE		ServiceHealthScore		COMMS	no	1	yes	AZURE-NETWORK

i	Time	Event
>	6/18/19 11:28:14.360 AM	{ [-]

actual_time: 06/18/2019 11:19:56
alert_color: #FCB64E
alert_group: DOTCOM
alert_level: 4
alert_period: 5
alert_severity: medium
alert_value: 5
alertable: no
description: SITES-[redacted] KPI 5xx Errors Count is medium 5
drilldown_search_earliest_offset: -300
drilldown_search_latest_offset: 300
drilldown_search_search: null
drilldown_search_title: null
drilldown_title: null
drilldown_uri: null
entity_key: [redacted]
entity_title: [redacted]
event_id: [redacted]
event_identifier_fields: source
event_identifier_hash: [redacted]
gs_kpi_id: [redacted]
gs_service_id: [redacted]
hour: 11
hour_alert: no
indexed_is_service_aggregate: 0
indexed_is_service_max_severity_event: 1
indexed_itsi_kpi_id: [redacted]
indexed_itsi_service_id: [redacted]
is_entity_defined: 1

NEAP Creation

- Set up a Policy to group by alert_group
  - Include events that have alert_group as a field
  - We used AZURE* because we were using our cloud services as our POC
  - Use * if you want everything to flow into this policy
- Broad is good here
  - We want to filter the majority of events though here
  - The fewer the NEAPs the easier to manage



Grouped Events!

- Notable Events grouping to alert group
- Notable events review correlating between services/KPIs
- Ability to take actions based on these correlated event groups
- Automated Closing: Separates new occurrences

7	AZURE-STORAGE	7/21/2019 7:08:15 PM CDT - 7/21/2019 7:14:16 PM CDT
7	AZURE-STORAGE	7/21/2019 7:08:15 PM CDT - 7/21/2019 7:14:16 PM CDT
17	AZURE-WEB	7/21/2019 6:47:12 PM CDT - 7/21/2019 7:03:18 PM CDT
17	AZURE-WEB	7/21/2019 6:47:12 PM CDT - 7/21/2019 7:03:18 PM CDT
75	AZURE-PAAS	7/21/2019 5:49:16 PM CDT - 7/21/2019 6:32:12 PM CDT
75	AZURE-PAAS	7/21/2019 5:49:16 PM CDT - 7/21/2019 6:32:12 PM CDT
7	AZURE-STORAGE	7/21/2019 6:18:16 PM CDT - 7/21/2019 6:24:17 PM CDT

Step Three: Create Additional Notables

splunk> .conf19

Problem:

- Some KPIs Flap
- Produce a significant amount of noise/events
- Want a way to be able to filter out this noise
  - Engineer getting up for an alert that self heals in 5 min will promote alert fatigue

Set Up Correlation search

```
index=itsi_summary kpiid="SHKPI-*"
| eventstats count(eval(alert_level>2)) as unhealthy_count count as total_count by serviceid
| eval perc_unhealthy = unhealthy_count / total_count
| dedup serviceid
| search perc_unhealthy > 0.8
| ` acme_itsi_summary_to_itsi_tracked_alerts_field_mapping`
```

Known Flappers Now Much Quieter!

- Reduction in Flapping
- Can set events simply as degraded vs instantly alerting
- Engineers Sleep More
- Noise Reduced
- More meaningful

Step Four: Add Alerting

splunk> .conf19

Problem:

- No alerting
- No way to page out
- No Way to communicate issues to engineers
- Analysts are not sent any notification: have to notice themselves

Add Alerting to NEAP

- Evaluate action on alertable
  - Allows us to only alert on items we deem important
  - Set up e-mail
  - Use tokens
  - Descriptive alert based on any field from itsi_summary
  - Set condition as “if the following event occurs”
    - Alertable matches yes

Action Rules

Create action rules upon this episode

Alert!

SPLUNK | SITES-[redacted] | critical | Service Alert | KPI Successful Provider Eligibility Check is critical 48

To Hasty, Matthew; [redacted]

Department: APPS

Alert Group: DOTCOM

Entity: service_aggregate

Sev Level=[redacted]

Policy:GEHA_KPI_ALERT_TRIGGER

View in Splunk ITSI: [redacted]

If you believe you've received this email in error, please see your Splunk administrator.

splunk > the engine for machine data

Step Five: Throttle Alerts

splunk> .conf19

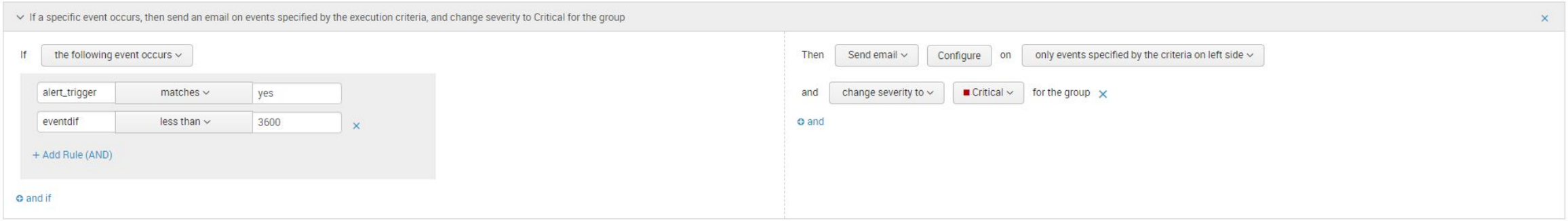
Create Second Correlation Search

- What does this search do?
  - Looks at all notable events created by our Azure NEAP
  - Looks for the alertable tag
  - Once an alertable event has been found, this search writes alert_trigger to the event
  - Search will only write a new event if a group had 0 alert_trigger events
  - Eventdif: use this to limit alerts to events from the last hour

```
((index=itsi_grouped_alerts itsi_policy_id=[redacted] OR (index=itsi_tracked_alerts))
| eventstats first(itsi_group_id) as itsi_group_id by event_id
| search index=itsi_tracked_alerts
| lookup itsi_notable_event_group_lookup _key as group_id OUTPUT status as itsi_group_status
| eventstats count(eval(alertable="yes")) as alertable_count count(eval(alert_trigger="yes")) as alert_trigger_count by itsi_group_id
| where alertable_count>0 and alert_trigger_count=0 and alertable="yes"
| dedup itsi_group_id
| eval alert_trigger="yes"
| eval eventdif= abs(_indextime-now())
| fields - alert_trigger_count alertable_count itsi_group_id itsi_group_status
```

Modify the NEAP

- Change alertable to alert_trigger
- Add in eventdif value as a condition



Simplified Actionable Alerting

- We had no good alerting through ITSI, it was complicated
- Developed this method for easy onboarding of new ITSI alerting
- Used Azure Services as our first Use Case
  - When Azure was starting to be used, used outsourcing for alerting on Azure issues
  - During this time, I began setting up ITSI services for us
  - Working with Jeff, we set up this method and used it for all our Azure Services
  - Was able to bring this back in house using this method + ITSI, save $$$ on contract

Future Hopes and Dreams

Time permitting…

splunk> .conf19

Integrating Alerts from Other Systems

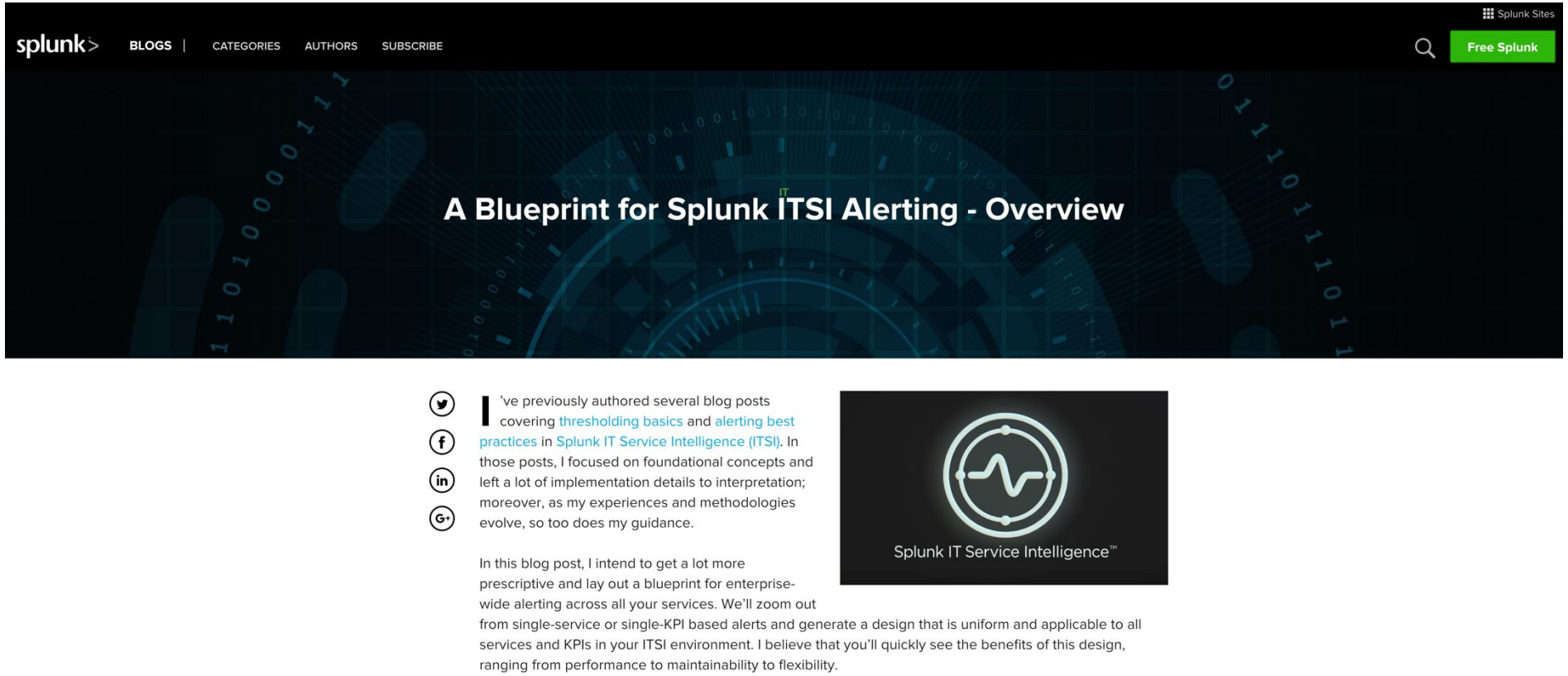
- Orion events coming in through add on
- Can add alert_groups to these
- Think about other platforms (ie through snmp, supported APMs,other add-ons,etc)

Move Toward Risk-based Alerting

- NEs get risk scores when created
- Alert based on services quickly rising in risk
- Apply modifiers to services, kpis, and entities to affect risk levels

splunk> .conf19

Get more information



TBD - Meet Matt and Jeff at this location at this time

Q&A

splunk> .conf19

.conf19

splunk>

Thank You !

Go to the .conf19 mobile app to

RATE THIS SESSION