.conf18

splunk>

Turning Security Use Cases into SPL

Marquis Montgomery | Sr. Staff Security Consultant

Anthony Tellez | Staff Data Scientist, Machine Learning & AI

October 2018 | Orlando, FL

Forward-Looking Statements

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 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.

Introductions and Agenda

| rest /services/speakerinfo

Marquis Montgomery, CISSP, SSCP, GSEC

marquis@splunk.com / @trademarq

Sr. Staff Security Consultant, Splunk

- **| where _time@Splunk > 5y**
- Previous: Former Splunk Customer, Manager of Corporate Secuirity at MSSP, Security Architect
- Leads Global Enterprise Security Field Enablement @ Splunk

Fact: No longer spends 80% of the year on a plane traveling globally

Anthony Tellez, CISSP, CEH, CDNA, Sec+

atellez@splunk.com / @anthonygtellez

Staff Data Scientist, Machine Learning & AI, Splunk

- **| where _time@Splunk > 4y**
- Previous: U.S. Gov Contractor, Geospatial Analyst
- Specializations
  - Cryptography
  - Information Security – Red Team

Fact: Spends 80% of the year on a plane traveling globally.

splunk> .conf18

Agenda

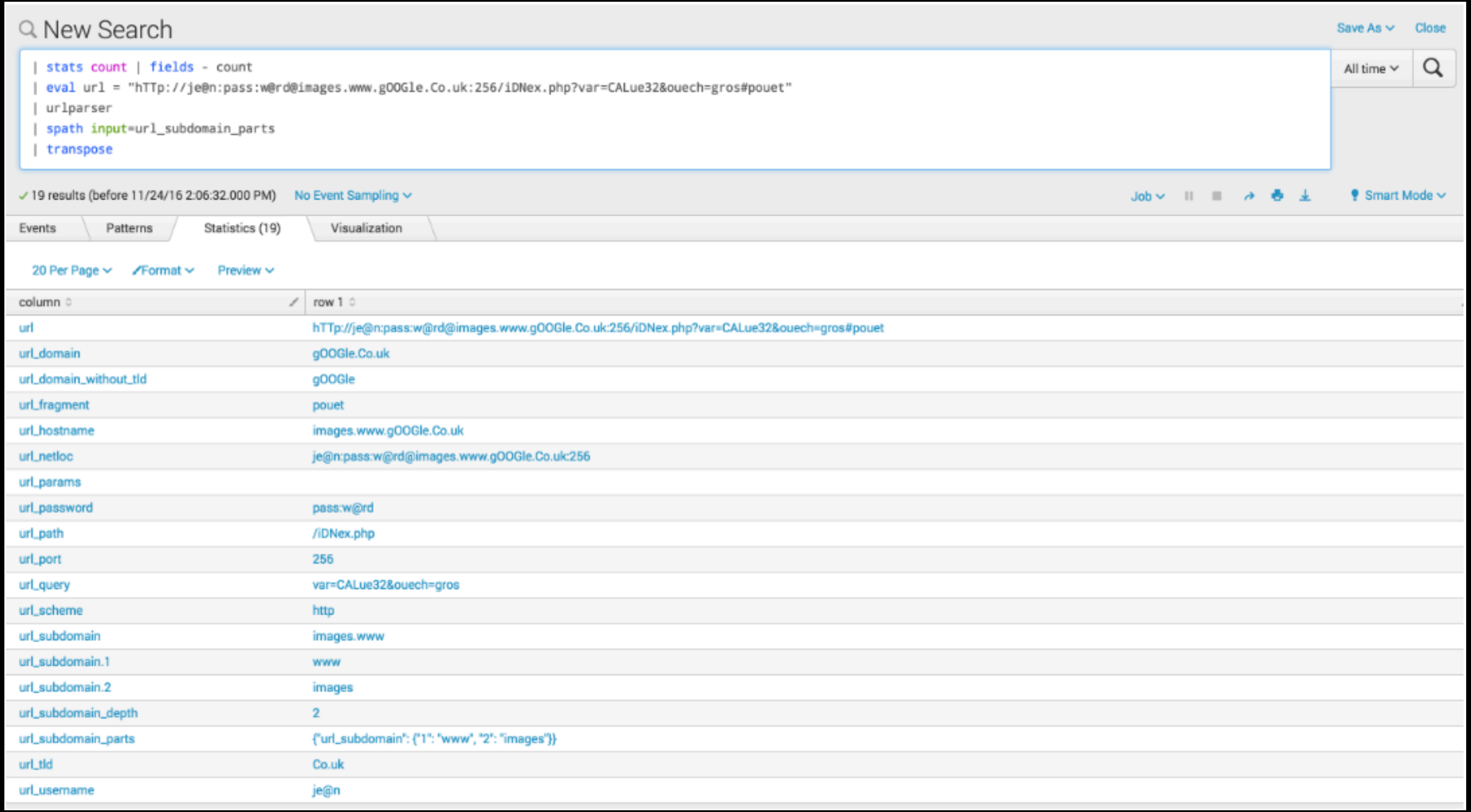
- Building Your Toolbox
- Understanding Data Models, and the tools to access them
- Explaining Security Use Case Patterns
- Useful Patterns and the SPL that drive them
- Key Takeaways and Q&A

Building your Toolbox

“Useful Weapons for your Arsenal of Security Analytics”

▶ URLparser

- Useful for deconstructing URLs in your data – giving you a wealth of useful fields to interrogate



```
| stats count | fields - count
| eval url = "hTTp://je@n:pass:w@rd@images.www.gOOGle.Co.uk:256/iDNex.php?var=CALue32&ouech=gros#pouet"
| urlparser
| spath input=url_subdomain_parts
| transpose
```

✓ 19 results (before 11/24/16 2:06:32.000 PM) No Event Sampling

column	row 1
url	hTTp://je@n:pass:w@rd@images.www.gOOGle.Co.uk:256/iDNex.php?var=CALue32&ouech=gros#pouet
url_domain	gOOGle.Co.uk
url_domain_without_tld	gOOGle
url_fragment	pouet
url_hostname	images.www.gOOGle.Co.uk
url_netloc	je@n:pass:w@rd@images.www.gOOGle.Co.uk:256
url_params	
url_password	pass:w@rd
url_path	/iDNex.php
url_port	256
url_query	var=CALue32&ouech=gros
url_scheme	http
url_subdomain	images.www
url_subdomain.1	www
url_subdomain.2	images
url_subdomain_depth	2
url_subdomain_parts	{"url_subdomain": {"1": "www", "2": "images"}}
url_tld	Co.uk
url_username	je@n

https://splunkbase.splunk.com/app/3396/

“Useful Weapons for your Arsenal of Security Analytics”

Getwatchlist

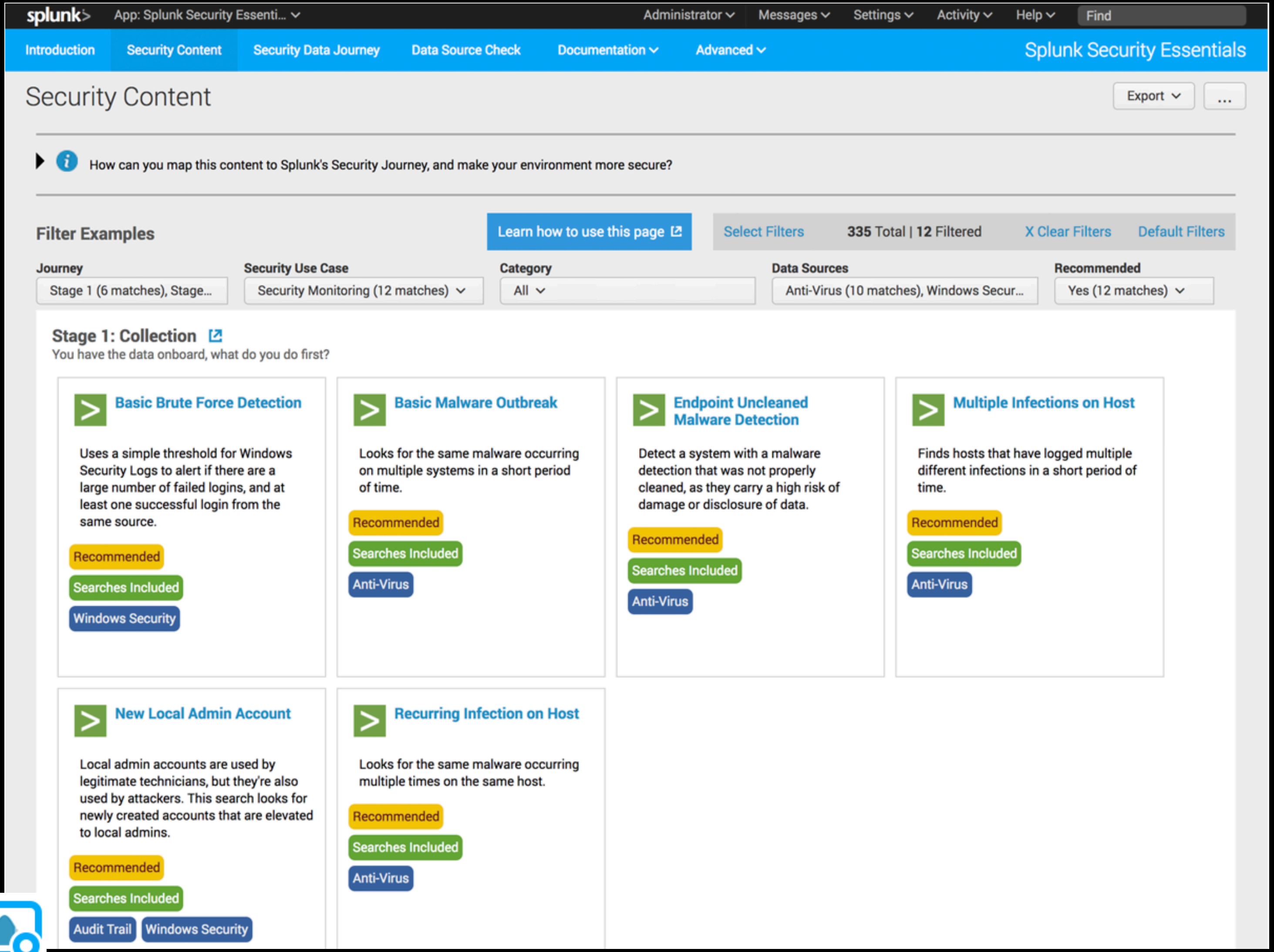
- Useful for retrieving CSV data from any URL.
- The common use for this is retrieving domains or IP addresses to be stored in a lookup table and then using them in searches in events (watchlists/threatlists).
- You can leverage this capability for all sorts of things, not just threat lists – what information is available that will make your use cases smarter? Ideas:
  - Centrally managed lookup tables for multiple search heads in your environment.

https://splunkbase.splunk.com/app/635/

“Useful weapons for your arsenal of security analytics”

Splunk Security Essentials

- Incredible collection of Security Content from Splunk’s top Security SMEs
- Includes easy to approach descriptions of each use case, and of course, example SPL!



Security Content

How can you map this content to Splunk's Security Journey, and make your environment more secure?

Filter Examples

335 Total | 12 Filtered

Journey: Stage 1 (6 matches), Stage...

Security Use Case: Security Monitoring (12 matches)

Category: All

Data Sources: Anti-Virus (10 matches), Windows Secur...

Recommended: Yes (12 matches)

Stage 1: Collection

You have the data onboard, what do you do first?

Basic Brute Force Detection

Uses a simple threshold for Windows Security Logs to alert if there are a large number of failed logins, and at least one successful login from the same source.

Recommended | Searches Included | Windows Security

Basic Malware Outbreak

Looks for the same malware occurring on multiple systems in a short period of time.

Recommended | Searches Included | Anti-Virus

Endpoint Uncleaned Malware Detection

Detect a system with a malware detection that was not properly cleaned, as they carry a high risk of damage or disclosure of data.

Recommended | Searches Included | Anti-Virus

Multiple Infections on Host

Finds hosts that have logged multiple different infections in a short period of time.

Recommended | Searches Included | Anti-Virus

New Local Admin Account

Local admin accounts are used by legitimate technicians, but they're also used by attackers. This search looks for newly created accounts that are elevated to local admins.

Recommended | Searches Included | Audit Trail | Windows Security

Recurring Infection on Host

Looks for the same malware occurring multiple times on the same host.

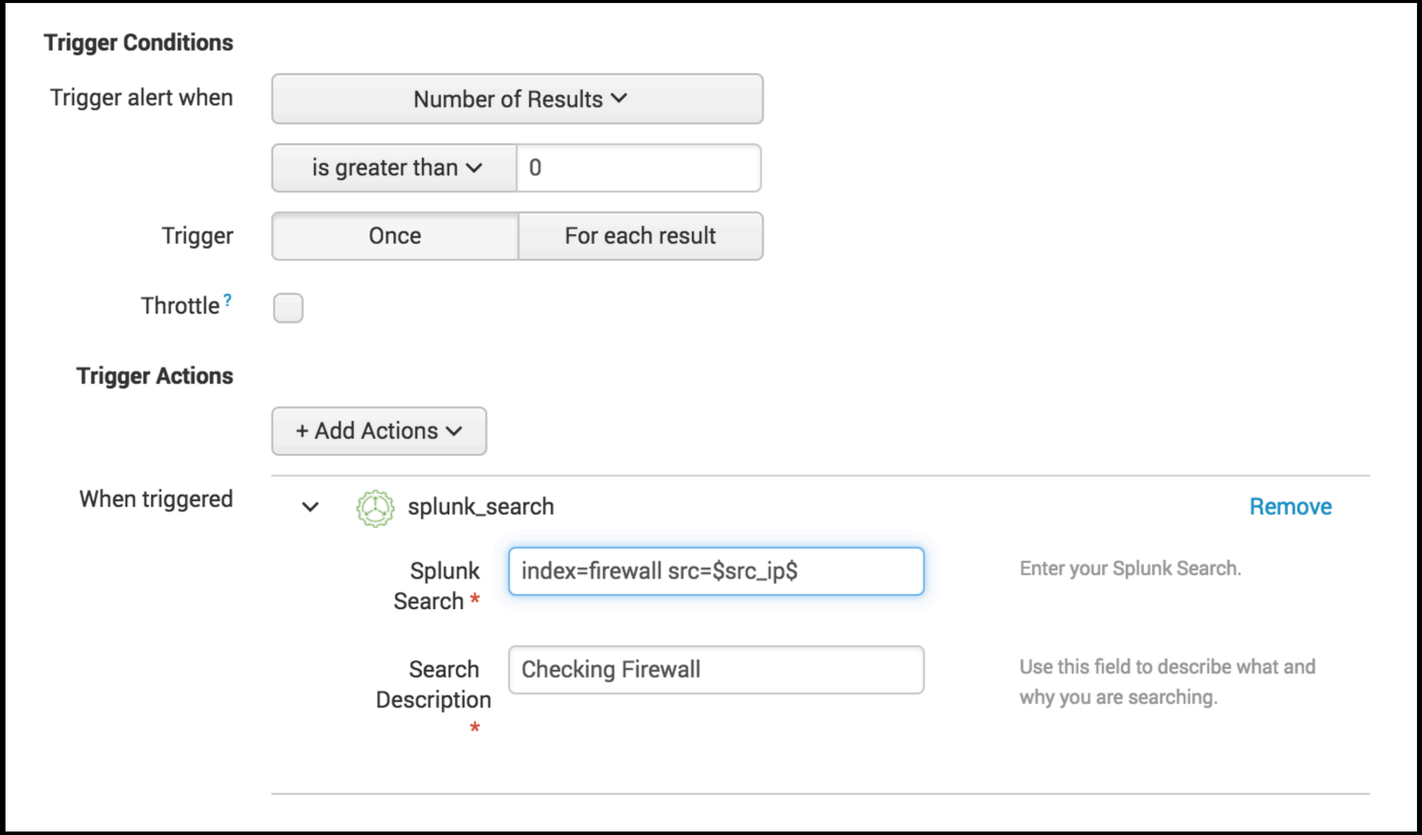
Recommended | Searches Included | Anti-Virus

https://splunkbase.splunk.com/app/3435/

“Useful Weapons for your Arsenal of Security Analytics”

Automatic Search Add-on for Splunk

- Fantastic Add-on with Adaptive Response Framework capabilities that will automatically run searches and index their results, as an alert action of the original search
- Ideal for streamlining additional investigation immediately following a detection event



Trigger Conditions

Trigger alert when: Number of Results

is greater than: 0

Trigger: Once | For each result

Throttle ? □

Trigger Actions

+ Add Actions

When triggered: splunk_search — Remove

Splunk Search *: index=firewall src=src_ip — Enter your Splunk Search.

Search Description *: Checking Firewall — Use this field to describe what and why you are searching.

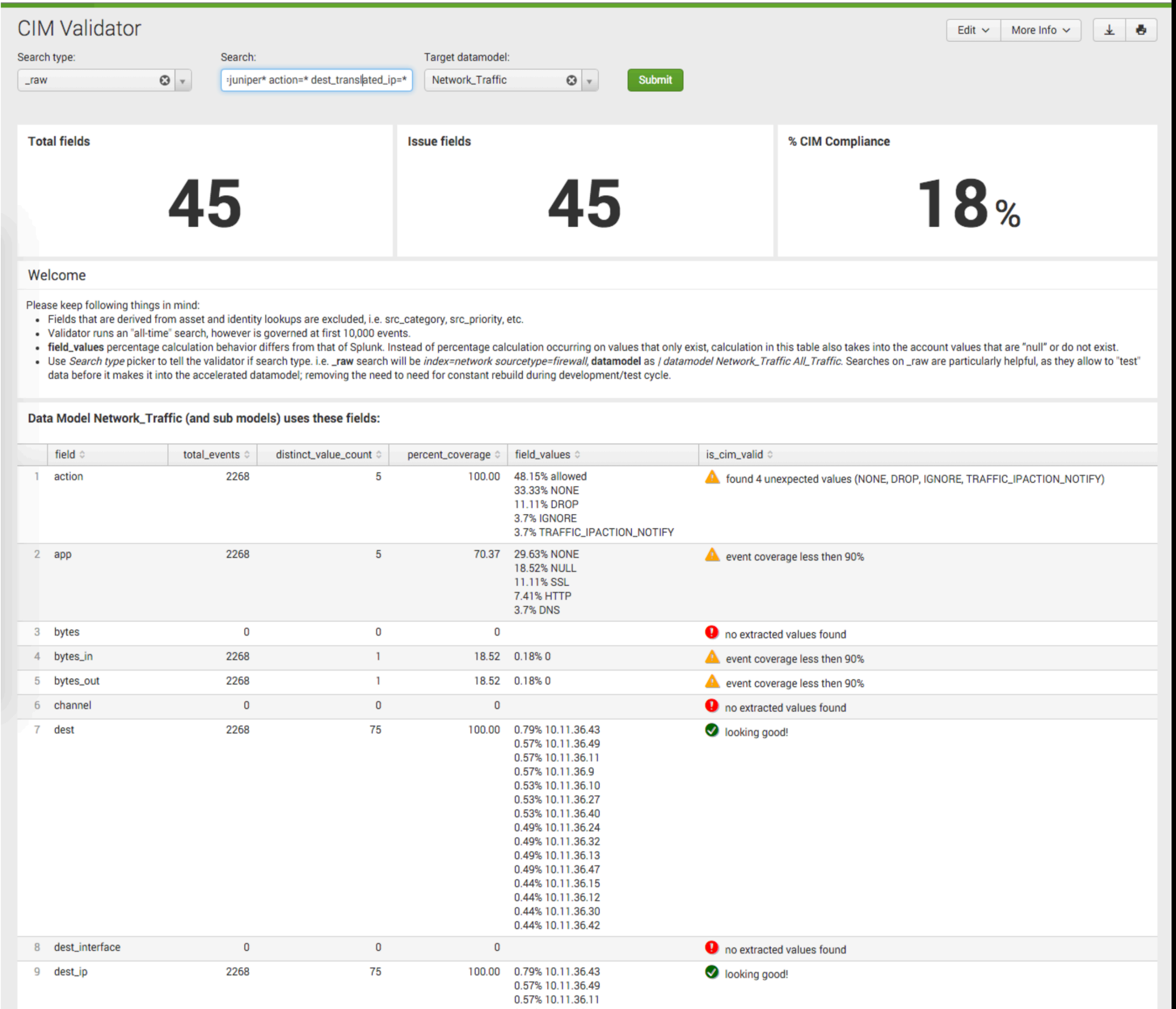
https://splunkbase.splunk.com/app/3837/

splunk> .conf18

"Useful Weapons for your Arsenal of Security Analytics"

▶ cim_validator

- Useful for validating all of your data is Common Information Model compliant
- This is super important!



CIM Validator

Search type: _raw

Search: :juniper* action=* dest_translated_ip=*

Target datamodel: Network_Traffic

Total fields	Issue fields	% CIM Compliance
45	45	18%

Welcome

Please keep following things in mind:

- Fields that are derived from asset and identity lookups are excluded, i.e. src_category, src_priority, etc.
- Validator runs an "all-time" search, however is governed at first 10,000 events.
- **field_values** percentage calculation behavior differs from that of Splunk. Instead of percentage calculation occurring on values that only exist, calculation in this table also takes into the account values that are "null" or do not exist.
- Use *Search type* picker to tell the validator if search type. i.e. **_raw** search will be *index=network sourcetype=firewall*, **datamodel** as *| datamodel Network_Traffic All_Traffic*. Searches on _raw are particularly helpful, as they allow to "test" data before it makes it into the accelerated datamodel; removing the need to need for constant rebuild during development/test cycle.

Data Model Network_Traffic (and sub models) uses these fields:

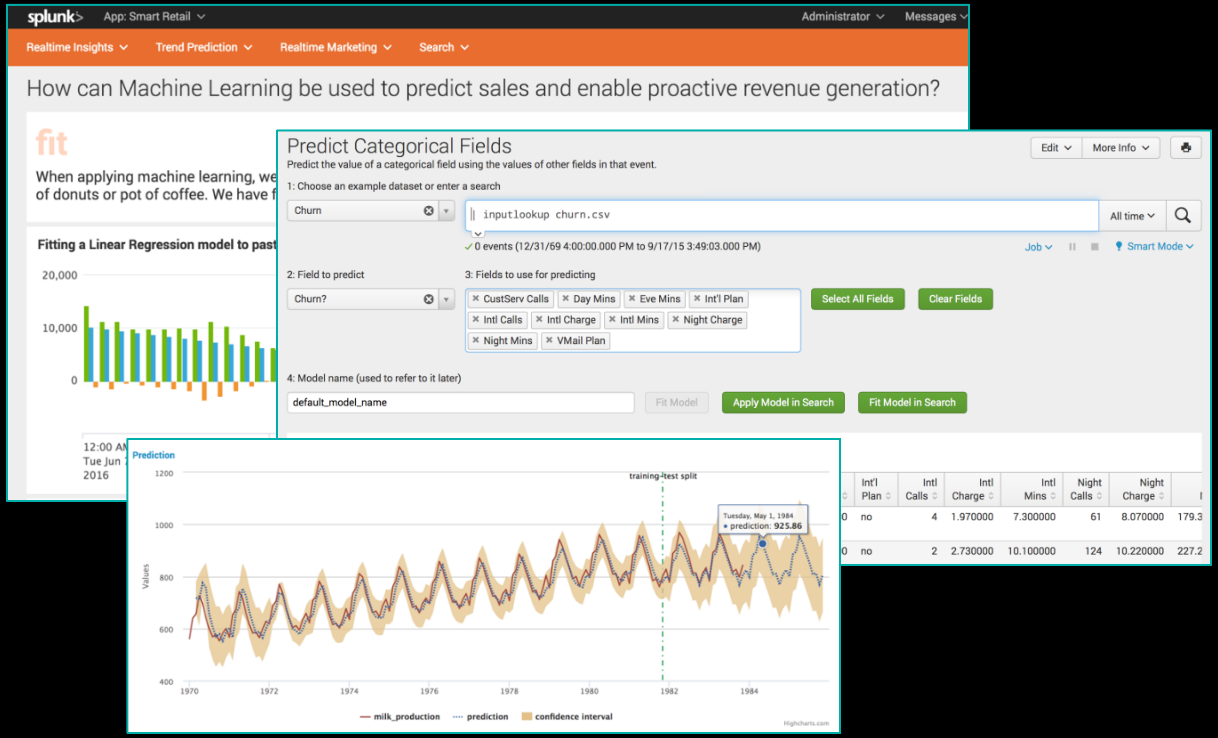
	field	total_events	distinct_value_count	percent_coverage	field_values	is_cim_valid
1	action	2268	5	100.00	48.15% allowed 33.33% NONE 11.11% DROP 3.7% IGNORE 3.7% TRAFFIC_IPACTION_NOTIFY	⚠ found 4 unexpected values (NONE, DROP, IGNORE, TRAFFIC_IPACTION_NOTIFY)
2	app	2268	5	70.37	29.63% NONE 18.52% NULL 11.11% SSL 7.41% HTTP 3.7% DNS	⚠ event coverage less then 90%
3	bytes	0	0	0		❗ no extracted values found
4	bytes_in	2268	1	18.52	0.18% 0	⚠ event coverage less then 90%
5	bytes_out	2268	1	18.52	0.18% 0	⚠ event coverage less then 90%
6	channel	0	0	0		❗ no extracted values found
7	dest	2268	75	100.00	0.79% 10.11.36.43 0.57% 10.11.36.49 0.57% 10.11.36.11 0.57% 10.11.36.9 0.53% 10.11.36.10 0.53% 10.11.36.27 0.53% 10.11.36.40 0.49% 10.11.36.24 0.49% 10.11.36.32 0.49% 10.11.36.13 0.49% 10.11.36.47 0.44% 10.11.36.15 0.44% 10.11.36.12 0.44% 10.11.36.30 0.44% 10.11.36.42	✓ looking good!
8	dest_interface	0	0	0		❗ no extracted values found
9	dest_ip	2268	75	100.00	0.79% 10.11.36.43 0.57% 10.11.36.49 0.57% 10.11.36.11	✓ looking good!

https://splunkbase.splunk.com/app/2968/

“Useful Weapons for your Arsenal of Security Analytics”

▶ Splunk Machine Learning Toolkit

- Macros
- Showcase Examples
- 300+ open source algorithms
  - Classification
  - Pre-Processing
  - Anomaly Detection
  - Etc.



How can Machine Learning be used to predict sales and enable proactive revenue generation?

When applying machine learning, we
of donuts or pot of coffee. We have f

Predict Categorical Fields

Predict the value of a categorical field using the values of other fields in that event.

1: Choose an example dataset or enter a search

Churn

```
| inputlookup churn.csv
```

✓ 0 events (12/31/69 4:00:00.000 PM to 9/17/15 3:49:03.000 PM)

2: Field to predict

Churn?

3: Fields to use for predicting

× CustServ Calls × Day Mins × Eve Mins × Int'l Plan × Intl Calls × Intl Charge × Intl Mins × Night Charge × Night Mins × VMail Plan

4: Model name (used to refer to it later)

default_model_name

Tuesday, May 1, 1984
• prediction: 925.86

Int'l Plan	Intl Calls	Intl Charge	Intl Mins	Night Calls	Night Charge
no	4	1.970000	7.300000	61	8.070000
no	2	2.730000	10.100000	124	10.220000

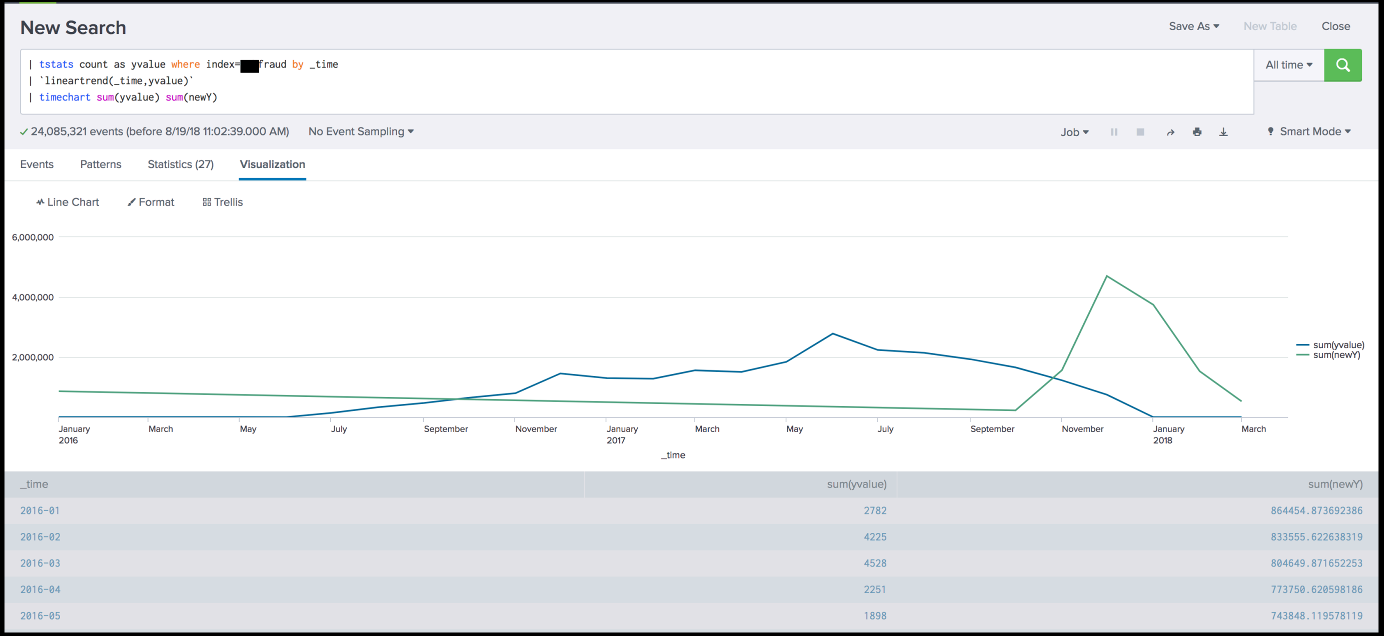
```
… | fit <ALGORITHM> <TARGET> from <VARIABLES …>
        <PARAMETERS> into <MODEL>
```

https://splunkbase.splunk.com/app/2890/

“Useful Weapons for your Arsenal of Security Analytics”

Analytics Toolkit

- Statistical Macros
- Threat based lookups
- Custom algorithms using ML-SPL API



```
| tstats count as yvalue where index=[redacted]fraud by _time
| `lineartrend(_time,yvalue)`
| timechart sum(yvalue) sum(newY)
```

✓ 24,085,321 events (before 8/19/18 11:02:39.000 AM) No Event Sampling

_time	sum(yvalue)	sum(newY)
2016-01	2782	864454.873692386
2016-02	4225	833555.622638319
2016-03	4528	804649.871652253
2016-04	2251	773750.620598186
2016-05	1898	743848.119578119

```
… | fit <ALGORITHM> <TARGET> from <VARIABLES …>
        <PARAMETERS> into <MODEL>
```

https://github.com/anthonygtellez/analytics_toolkit

Understanding Data Models

What is a Data Model?

> "A data model is a hierarchically structured search-time mapping of semantic knowledge about one or more datasets."

Splunk Docs

Level Set: What is a Data Model?

- A useful feature of Splunk Enterprise to help normalize and accelerate Splunk Data. Data Model objects allow you to shape and query information without the need for old school searches that require indexes and source types.
- Old School SPL to get successful authentication events across data sources

  -
    ```
    (index=windows_servers OR index=unix_servers) ((EventCode=4768 OR
    EventCode=672 OR EventCode=676) OR (sshd Accepted OR "Authorized to”)
    ```
- New School SPL to get successful authentication events across data sources

  -
    ```
    | from datamodel:"Authentication.Successful_Authentication”
    ```
- New school SPL to get successful authentication results across data sources, *quickly at scale*

  -
    ```
    | tstats summariesonly=t count from datamodel=Authentication where
    (nodename = Authentication.Successful_Authentication)
    ```

Level Set: What is Common Information Model (CIM)?

- The Splunk **Common Information Model (CIM)** is a shared semantic model focused on extracting value from data. The CIM is implemented as an add-on that contains a collection of data models, documentation, and tools that support the consistent, normalized treatment of data for maximum efficiency at search time.
- The built in Data Models from the CIM Add-On are leveraged by all Splunk Premium solutions – you should use them if you are building Security Use Cases with Splunk. All of the common data models you need to get started are included.
- CIM Data Models leverage common field names, allowing for easy correlation across data sources and data types.
- See Splunk Docs for details and more information: http://docs.splunk.com/Documentation/CIM/latest/User/Overview

Deep Dive: | tstats

- Allows you to access summarized(accelerated) data via the lens of a Data Model. Try to use this as much as possible – this is how you will scale performance.
- Supports all of the stats functions to help you report on data (no raw data supported). Leverage these for statistics and retrieving values:
  - `| tstats count values(Authentication.src_ip) from datamodel=Authentication by Authentication.user, Authentication.dest`
- Has a few useful attributes you really want to understand and use:
  - `summariesonly=t`
    - Allows you to control if we work in mixed mode or not (summariesonly gets you the perf boost)
  - `append=t`
    - Allows you to control if this generating command can actually be used to add data to the search pipeline. Essential for correlating multiple data sources.
  - `where = index=“windows”`
    - Allows you to filter to a specific subset of data eg: data associated with a specific index, source, sourcetype.

DEMO

splunk> .conf18

Ever Deeper Dive: | tstats

- See these slides from .conf 2017 from David Veuve
- https://conf.splunk.com/files/2017/slides/searching-fast-how-to-start-using-tstats-and-other-acceleration-techniques.pdf
- Much more detail on how data is indexed, how raw data is stored, how indexed fields are stored in TSIDX, and how somebody had the idea to use the lexicon as a field value store.
- Just remember, you need Data Models first, then summary acceleration turned on.
- When you build security use cases, leverage | tstats and data model summaries as much as possible – this is how you scale up.

Understanding Security Use Case Patterns

Understanding Security Use Case Patterns

- Essentially Security Use Case Patterns are my phrase for **a way of doing something via SPL in a useful way.**
- Requirements:
  - Highly Performant SPL
  - Leverages analytics that are accurate
  - Simplifies correlation across data sources
  - Can be used interchangeably across use cases.
- Examples:
  - Detecting when a user visits a website for the first time == **use case**
  - Detecting when a new remote host scans your website == **use case**
  - SPL for detecting anything we've never seen before == **use case pattern**

Useful Patterns and the SPL that Drive Them

Pattern #1: Basic Correlation

How do I correlate a field across two different data types / data models?

- Example: **Identify a server who has been subject to an IDS attack, and retrieve any outbound communications from that server**
- **SPL**

```
| tstats summariesonly=t count from datamodel=Network_Traffic where [ |
tstats summariesonly=t count from datamodel=IDS_Attacks by IDS_Attacks.dest |
rename IDS_Attack.dest as All_Traffic.src ]  by All_Traffic.src,
All_Traffic.dest
```

1. **(subsearch) Find IDS DM Attack Targets (on our network) and rename them as All_Traffic.src**
2. **Using the hostnames/ips we have seen attacked, ask Network Traffic DM for counts of traffic events grouped by src (One of our machines recently attacked) and dest (some other host)**

Pattern #2: Count Against Static Threshold

How do I compare against a Static Threshold?

- Example: **Identify a server who has been subject to an IDS attack, and retrieve any outbound communications from that server, filter to static requirement for minimum number of connections**
- SPL

```
| tstats summariesonly=t count from datamodel=Network_Traffic where [ |
tstats summariesonly=t count from datamodel=IDS_Attacks by IDS_Attacks.dest |
rename IDS_Attack.dest as All_Traffic.src ]  by All_Traffic.src,
All_Traffic.dest
| where count > 10
```

- **Using example from last slide – but this time only alert me when the number of events is greater than 10.**

Pattern #3: Count Against Dynamic Threshold

How do I compare against a Dynamic Threshold?

- Example: **I need to determine when something happens more often than normal, and I don’t know what normal is.**
- SPL

```
tag=email
| search src_user=*@mycompany.com
| bucket _time span=1d
| stats count by src_user, _time
| stats count as num_data_samples
        max(eval(if(_time >= relative_time(now(), "-1d@d"), 'count',null))) as recent_count
        avg(eval(if(_time<relative_time(now(),"-1d@d"),'count',null))) as avg
        stdev(eval(if(_time<relative_time(now(),"-1d@d"),'count',null))) as stdev by src_user
| where recent_count > (avg+stdev*2) AND num_data_samples >=7
```

- Notes
  - **Basically, we are finding email events with a particular filter, and then bucketing those events together to find recent_count, avg count, and std_dev for that filter, and then only alerting on event that break two standard deviatations and have a full weeks worth of data.**

Pattern #4: First Seen Detection

How do I know if I have seen this <ip/host/signature/user/url> before?

- Example: **I want to know the first time someone connects a USB drive**
- **SPL**

```
sourcetype=win*security EventCode=<some event code>
| stats earliest(_time) as earliest latest(_time) as latest by user, dest
| eval isOutlier=if(earliest >= relative_time(now(), "-1d@d"), 1, 0)
| where isOutlier=1
```

- **Notes**
  - **Here, we are determining the earliest and latest time we have seen this user, dest combination, then we find out if the earliest time is within the last day (adjust as needed)**

Pattern #5: Quickly Search A Lookup Table

How do I retrieve a specific record from a lookup table?

- Example: **I need to identify who my approved account managers are so I can use that information in other searches.**
- **SPL**

```
| inputlookup big_lookup_table.csv WHERE ACCOUNT_TYPE=“Privileged”
```

- **Notes**
  - **This prevents you from needing to create a separate lookup table for every thing you need enrichment for. With the WHERE attribute, you can easily and quickly filter your lookup table request for what you are looking for.**
  - **Combine this with other patterns to efficiently optimize your use case.**

Pattern #6: Compare Known List to Current Activity

How do I compare a list of known <ips/hosts/signatures/users/urls> to activity happening now?

- Example: **I have a list of critical business application URIs I need to watch proxy logs for, what the most efficient way to do that?**
- **SPL**

```
|  tstats summariesonly=t count from datamodel=Web where (
     [|  inputlookup test_watchlist.csv
     |  table watchlist
     | rename watchlist as Web.uri_path]) by Web.src, Web.dest, Web.user,
Web.uri_path
```

- **Notes**
  - **First we need a lookup table containing the uris to watch for – you can update this without messing around with searches, AND its reusable!**
  - **Then we reference that lookup table using a subsearch and format it to match the relevant field name of the data model we are searching against.**

Pattern #6: Compare Known List to Current Activity

How do I compare a list of known <ips/hosts/signatures/users/urls> to activity happening now?

- Example: **I want to know when a service account is used in a Windows batch logon.**
- SPL

```
index=domain_controller sourcetype=WinEventLog (EventCode=4624 OR
EventCode=4625) LogonType=4 [inputlookup service_accounts.csv]
```

- Notes
  - **The lookup table needs to have a column with a field name that corresponds to the data found in Splunk. In this example, the column containing the service accounts should be called "user" for this search to work. Splunk will append each user with an OR automatically and then run the resulting search against the raw data for you.**

Pattern #7: Tune Out False Positives

- **What is the *best* way to tune out false positives?**
- Well, it depends.
  - Verify the accuracy of your search
  - Use Confidence Checking to ensure your data meets your expectations (i.e. ensuring # of data samples in pattern #3)
  - Leverage a summary index where other things must happen in concert there first and then your actual alert is based off those things happening in concert
  - Finally - Leverage NOT in SPL or a NOT subsearch lookup table to exclude one-off bad values, but USE SPARINGLY. NOT is expensive!

```
|  tstats summariesonly=t count from datamodel=Web where NOT (
     [|  inputlookup test_watchlist.csv
      |  table watchlist
      | rename watchlist as Web.uri_path]) by Web.src, Web.dest, Web.user,
Web.uri_path
```

Pattern #8: How Can I Avoid Using | Join for Correlation

- What is the *best* way to not use join for correlation?
- Well here's one way:
- **SPL**

```
| tstats prestats=t summariesonly=t allow_old_summaries=t count(Malware_Attacks.src) from datamodel=Malware where Malware_Attacks.action=allowed groupby Malware_Attacks.dest

| tstats prestats=t append=t summariesonly=t allow_old_summaries=t count(IDS_Attacks.src) from datamodel=Intrusion_Detection groupby IDS_Attacks.dest

| tstats prestats=t append=t summariesonly=t allow_old_summaries=t dc(Authentication.dest) from datamodel=Authentication groupby Authentication.src

| rename Malware_Attacks.dest as dest IDS_Attacks.dest as dest Authentication.src as dest

| stats count(Malware_Attacks.src) as malware_sources count(IDS_Attacks.src) as ids_sources dc(Authentication.dest) as authentication_destinations by dest

| where malware_sources > 1 OR ids_sources > 1 AND authentication_destinations > 3
```

- **Notes**
  - **This is a bonus example, that is also an advanced lateral movement detection use case ☺**

Pattern #9: Detecting Randomness

- How do I detect any randomness in <insert field here>?
  - Examples of use cases could be detecting Dynamically Generated Domains or randomly generated process names.
- URLToolbox ut_shannon Shannon Entropy capability
- Random Process Name Example

```
sourcetype=win*security EventCode=4688 Users New_Process_Name=*\Users\* |
stats count by New_Process_Name,host | lookup ut_shannon_lookup word as
New_Process_Name | rename ut_shannon as "Shannon Entropy Score"
New_Process_Name as Process,host as Endpoint
```

- Splunk Security Essentials has examples!

Pattern #9: Detecting Randomness #2

- How do I detect any randomness in <insert field here>?
  - Examples of use cases could be detecting Dynamically Generated Domains or randomly generated process names.
- URLToolbox ut_shannon Shannon Entropy capability
- Dynamically Generated Domain Example

```
| tstats `summariesonly` count from datamodel=Network_Resolution where (DNS.record_type=A
OR DNS.record_type=AAAA) NOT (`cim_corporate_web_domain_search(DNS.query)`) AND
source!="stream:Splunk_*"  by _time span=1s DNS.src DNS.query
| `drop_dm_object_name(DNS)`
| rename query as dns_query
| rex field=dns_query ".*?(?=[^\.]+\.\w+)(?<domain>[^\.]+\.\w+)$"
| lookup alexa_lookup_by_str domain OUTPUTNEW rank as alexa_rank
| where isnull(alexa_rank)
| `ut_shannon(dns_query)`
| where ut_shannon>3.9
| stats count latest(_time) as _time values(dns_query) values(ut_shannon) dc(src) as
src_count values(src) by domain
```

- Another bonus use case, just for you ☺

Shannon Entropy for DGA Hunting

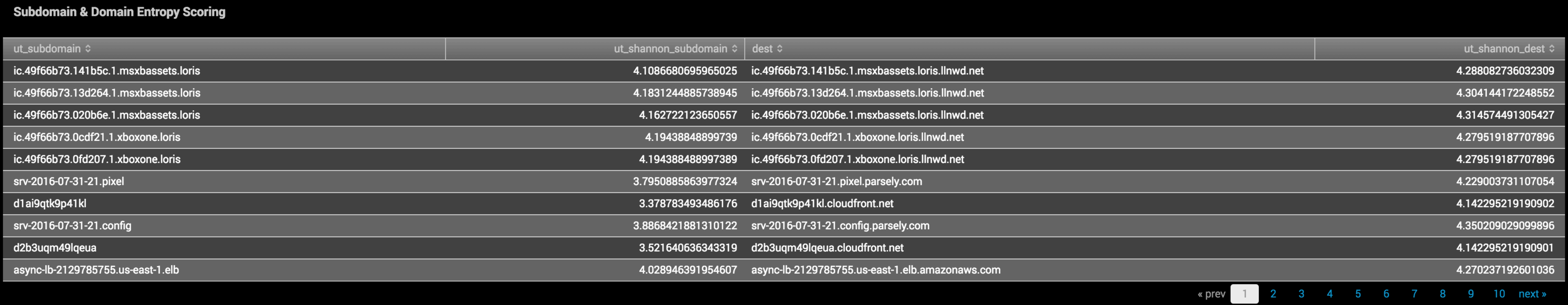
- **What is Shannon Entropy?**
  - "… a measure of uncertainty in a random variable"
- **How does it help us find malware and anomalous activity?**
  - The more random a string is, the higher its calculation of randomness.
    - *aaaaa.com (Score 1.8)*
    - *Google.com (Score 2.6)*
    - Ic49f66b73141b5c1.com (Score 4.1)
  - Domains and subdomains with high entropy are good indicators of malicious behavior.
  - **We can filter to domains or subdomains with a score above 3 or 4.**

$$H = -\sum p(x) \log p(x)$$

Shannon Entropy for DGA Hunting

Results of HTTP/DNS Entropy Scoring

- **Cons:**
  - **False positives**
    - CDNs like Amazon, Akamai, and others use pseudorandom generated subdomains
    - Requires to you to keep a blacklist or whitelist of domains to reduce noise when hunting (but, relatively easy to do in Splunk)
  - **Malware evolves**
    - Locky & others using shorter subdomains or domains to reduce randomness, reducing entropy score

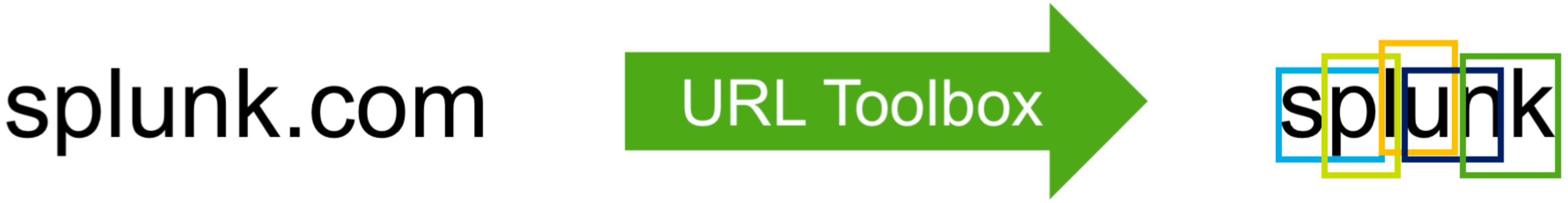


Subdomain & Domain Entropy Scoring

ut_subdomain ⇕	ut_shannon_subdomain ⇕	dest ⇕	ut_shannon_dest ⇕
ic.49f66b73.141b5c.1.msxbassets.loris	4.1086680695965025	ic.49f66b73.141b5c.1.msxbassets.loris.llnwd.net	4.288082736032309
ic.49f66b73.13d264.1.msxbassets.loris	4.1831244885738945	ic.49f66b73.13d264.1.msxbassets.loris.llnwd.net	4.304144172248552
ic.49f66b73.020b6e.1.msxbassets.loris	4.162722123650557	ic.49f66b73.020b6e.1.msxbassets.loris.llnwd.net	4.314574491305427
ic.49f66b73.0cdf21.1.xboxone.loris	4.19438848899739	ic.49f66b73.0cdf21.1.xboxone.loris.llnwd.net	4.279519187707896
ic.49f66b73.0fd207.1.xboxone.loris	4.194388488997389	ic.49f66b73.0fd207.1.xboxone.loris.llnwd.net	4.279519187707896
srv-2016-07-31-21.pixel	3.7950885863977324	srv-2016-07-31-21.pixel.parsely.com	4.229003731107054
d1ai9qtk9p41kl	3.378783493486176	d1ai9qtk9p41kl.cloudfront.net	4.142295219190902
srv-2016-07-31-21.config	3.8868421881310122	srv-2016-07-31-21.config.parsely.com	4.350209029099896
d2b3uqm49lqeua	3.521640636343319	d2b3uqm49lqeua.cloudfront.net	4.142295219190901
async-lb-2129785755.us-east-1.elb	4.028946391954607	async-lb-2129785755.us-east-1.elb.amazonaws.com	4.270237192601036

Randomness Algo

Basic Idea


splunk.com

URL Toolbox

splunk

What is the probability of each pair of letters?

Letter Pair (bi-gram)	Probability Score
sp	0.24
pl	0.18
lu	0.10
un	0.22
nk	0.09

Average Score: 0.169 ✓

Where do you Build Baselines From?

Domain	Rank
netflix	1
google	2
microsoft	3
facebook	4
hola	5
doubleclick	6
google-analytics	7
youtube	8
fbcdn	9
apple	10

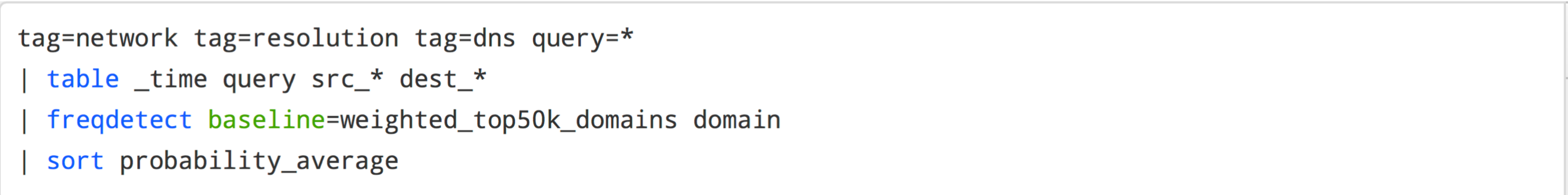
Letter Pair (bi-gram)	# Occurrences in Baseline	Probability Score
le	4	1.0
oo	3	0.75
ic	3	0.75
og	2	0.50
ou	2	0.50
… and 49 more bi-grams …		

Probability for each bigram = # occurrences / max(# occurrences)

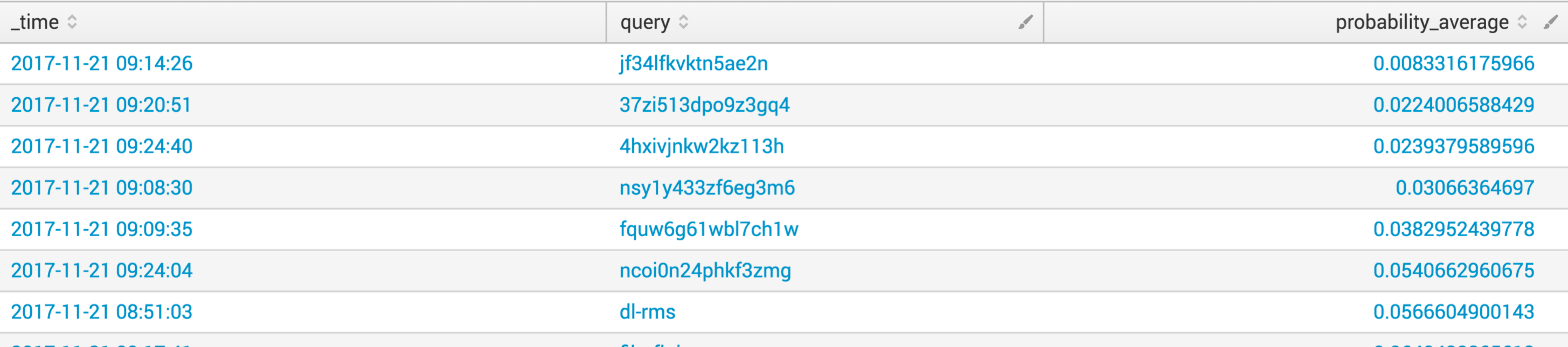
Probability for each word = avg(found bigrams)

Using the Frequency Baselines

- Simplest Form:



```
tag=network tag=resolution tag=dns query=*
| table _time query src_* dest_*
| freqdetect baseline=weighted_top50k_domains domain
| sort probability_average
```

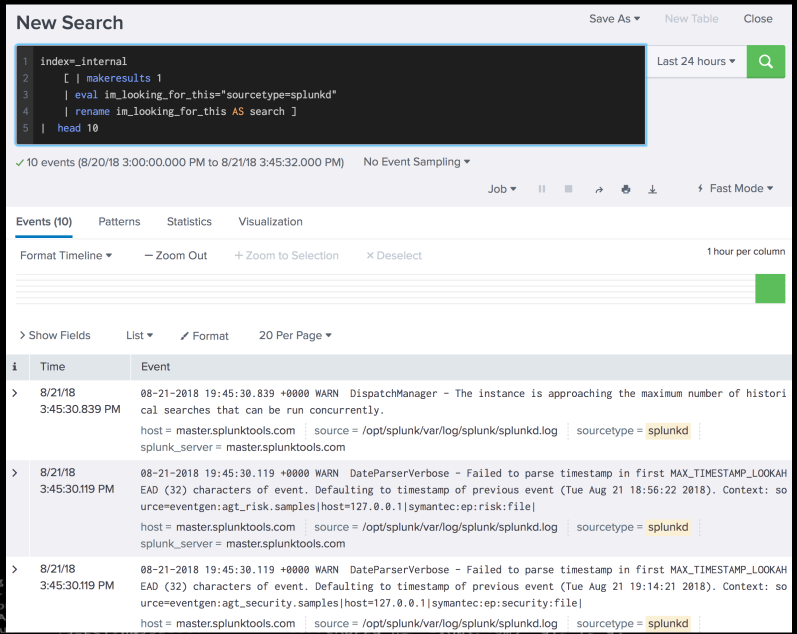


_time	query	probability_average
2017-11-21 09:14:26	jf34lfkvktn5ae2n	0.0083316175966
2017-11-21 09:20:51	37zi513dpo9z3gq4	0.0224006588429
2017-11-21 09:24:40	4hxivjnkw2kz113h	0.0239379589596
2017-11-21 09:08:30	nsy1y433zf6eg3m6	0.03066364697
2017-11-21 09:09:35	fquw6g61wbl7ch1w	0.0382952439778
2017-11-21 09:24:04	ncoi0n24phkf3zmg	0.0540662960675
2017-11-21 08:51:03	dl-rms	0.0566604900143

Pattern #10: Searching on the Fly

- I need to build a search on the fly and then search my data with it to find the things.. How can I do that?
- Explanation
  - You can use Splunk to generate a Splunk search, and then run it, in one search.
  - Yes you heard me right. This is different from finding key/value pairs to look for – we are actually passing SPL.
  - Example: I need to generate the parameters of multiple Splunk commands, so a subsearch won't work alone.
- SPL

```
index=_internal
    [ | makeresults 1
    | eval im_looking_for_this="sourcetype=splunkd"
    | rename im_looking_for_this AS search ]
|  head 10
```



```
index=_internal
    [ | makeresults 1
    | eval im_looking_for_this="sourcetype=splunkd"
    | rename im_looking_for_this AS search ]
|  head 10
```

✓ 10 events (8/20/18 3:00:00.000 PM to 8/21/18 3:45:32.000 PM) No Event Sampling ▾

i	Time	Event

>	8/21/18 3:45:30.839 PM	08-21-2018 19:45:30.839 +0000 WARN DispatchManager - The instance is approaching the maximum number of historical searches that can be run concurrently.
		host = master.splunktools.com source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd
		splunk_server = master.splunktools.com

>	8/21/18 3:45:30.119 PM	08-21-2018 19:45:30.119 +0000 WARN DateParserVerbose - Failed to parse timestamp in first MAX_TIMESTAMP_LOOKAHEAD (32) characters of event. Defaulting to timestamp of previous event (Tue Aug 21 18:56:22 2018). Context: source=eventgen:agt_risk.samples|host=127.0.0.1|symantec:ep:risk:file|
		host = master.splunktools.com source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd
		splunk_server = master.splunktools.com

>	8/21/18 3:45:30.119 PM	08-21-2018 19:45:30.119 +0000 WARN DateParserVerbose - Failed to parse timestamp in first MAX_TIMESTAMP_LOOKAHEAD (32) characters of event. Defaulting to timestamp of previous event (Tue Aug 21 19:14:21 2018). Context: source=eventgen:agt_security.samples|host=127.0.0.1|symantec:ep:security:file| host = master.splunktools.com source = /opt/splunk/var/log/splunk/splunkd.log sourcetype = splunkd

Pattern #11: Lookup Caching

- I need to automatically generate a list of things that I will use in searches later.
- Examples:
  - Lists of approved administrative accounts
  - Accounts that will be leaving the organization soon
  - Third Party Contractors
  - Hosts that have crossed a High Risk threshold
- SPL

```
index=my_special_data sourcetype=important_machines | stats count by
machine_name | table machine_name | outputlookup create_empty=f
override_if_empty=f my_special_machines.csv
```

  - Protip: make this a scheduled search!

Pattern #12: Time Series Analysis

- I want to detect someone <doing something more than they normally have in the past>.
- Examples:
  - Detect someone who is
    - Logging in to more servers than normal
    - Printing more files than normal
    - Uploading more data than normal
- SPL

```
index=windows OR index=login user=*
| bin span=1d _time | stats dc(host) as count by user _time
| stats max(eval(if(_time >= relative_time(now(), "1d"), count, null))) as latest avg(eval(if(_time < relative_time(now(), "-
1d"),count,null))) as average, stdev(eval(if(_time < relative_time(now(), "-1d"),count,null))) as stdev
by user
| where latest>stdev+average
```

- Notes
  - Bin time, relative time are totally adjustable, as are your search parameters and your key field.
  - This method utilizes the simple but powerful stddev approach, which is most useful for events that happen frequently. If your use case applies to actions that are not very frequent, *a more advanced machine learning approach is better.*

Chebyshev's Inequality

Assumes your data is not normal

- "In probability theory, **Chebyshev's inequality** (also called the **Bienaymé-Chebyshev inequality**) guarantees that, for a wide class of probability distributions, no more than a certain fraction of values can be more than a certain distance from the mean. Specifically, no more than $1/k^2$ of the distribution's values can be more than k standard deviations away from the mean (or equivalently, at least $1-1/k^2$ of the distribution's values are within k standard deviations of the mean)." – Wikipedia
- "In practical usage, in contrast to the 68–95–99.7 rule, which applies to normal distributions, Chebyshev's inequality is weaker, stating that a minimum of just 75% of values must lie within two standard deviations of the mean and 89% within three standard deviations.[1][2]"

$$
\begin{aligned}
\Pr\left(\|X-\mu\|_\alpha \geq k\sigma_\alpha\right) &= \int_\Omega \mathbf{1}_{\|X-\mu\|_\alpha \geq k\sigma_\alpha} \,\mathrm{d}\Pr \\
&= \int_\Omega \left(\frac{\|X-\mu\|_\alpha^2}{\|X-\mu\|_\alpha^2}\right) \cdot \mathbf{1}_{\|X-\mu\|_\alpha \geq k\sigma_\alpha} \,\mathrm{d}\Pr \\
&\leq \int_\Omega \left(\frac{\|X-\mu\|_\alpha^2}{(k\sigma_\alpha)^2}\right) \cdot \mathbf{1}_{\|X-\mu\|_\alpha \geq k\sigma_\alpha} \,\mathrm{d}\Pr \\
&\leq \frac{1}{k^2\sigma_\alpha^2} \int_\Omega \|X-\mu\|_\alpha^2 \,\mathrm{d}\Pr \qquad \mathbf{1}_{\|X-\mu\|_\alpha \geq k\sigma_\alpha} \leq 1 \\
&= \frac{1}{k^2\sigma_\alpha^2} \left(\mathrm{E}\,\|X-\mu\|_\alpha^2\right) \\
&= \frac{1}{k^2\sigma_\alpha^2} \left(\sigma_\alpha^2\right) \\
&= \frac{1}{k^2}
\end{aligned}
$$

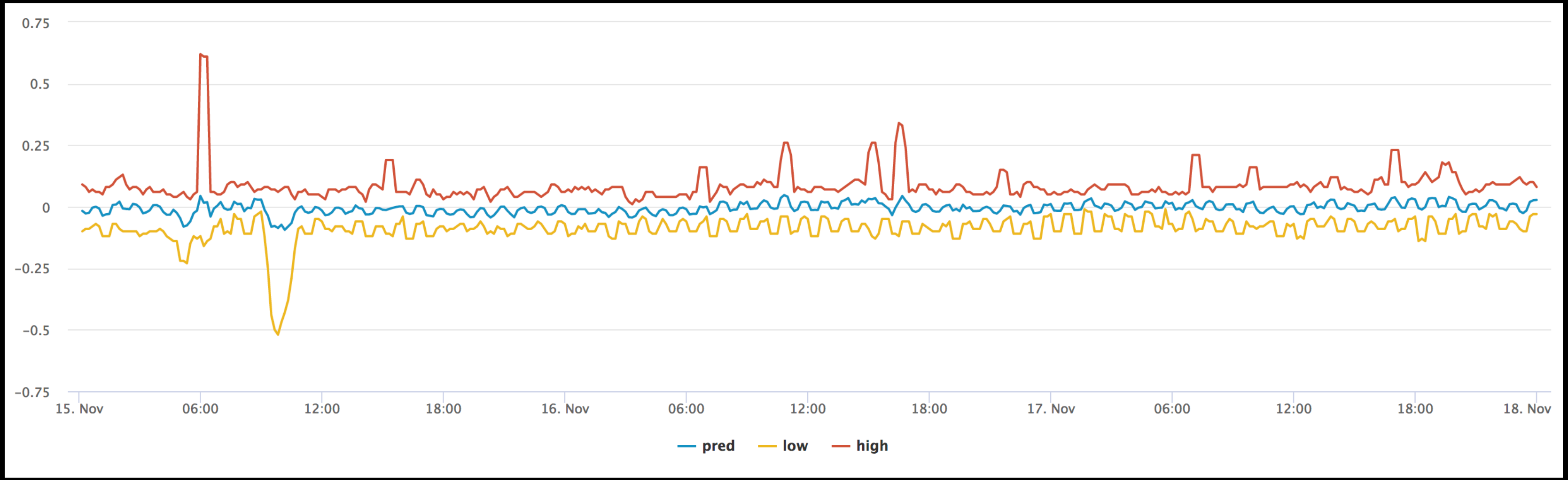
https://www.splunk.com/blog/2018/01/19/cyclical-statistical-forecasts-and-anomalies-part-1.html

Forecasting PCR

Crystal Ball forecasting for entity analysis

- Predict the next 3 days of traffic flow based on 5 weeks data with a 90% confidence interval, normalized around the average PCR ratio:

```
index=suricata event_type=flow
| `pcr(bytes_in,bytes_out)`
| timechart span=10m avg(pcr_ratio) as avg_pcr_ratio
| `forecast5w(avg_pcr_ratio,90.0,+1d,3)`
```

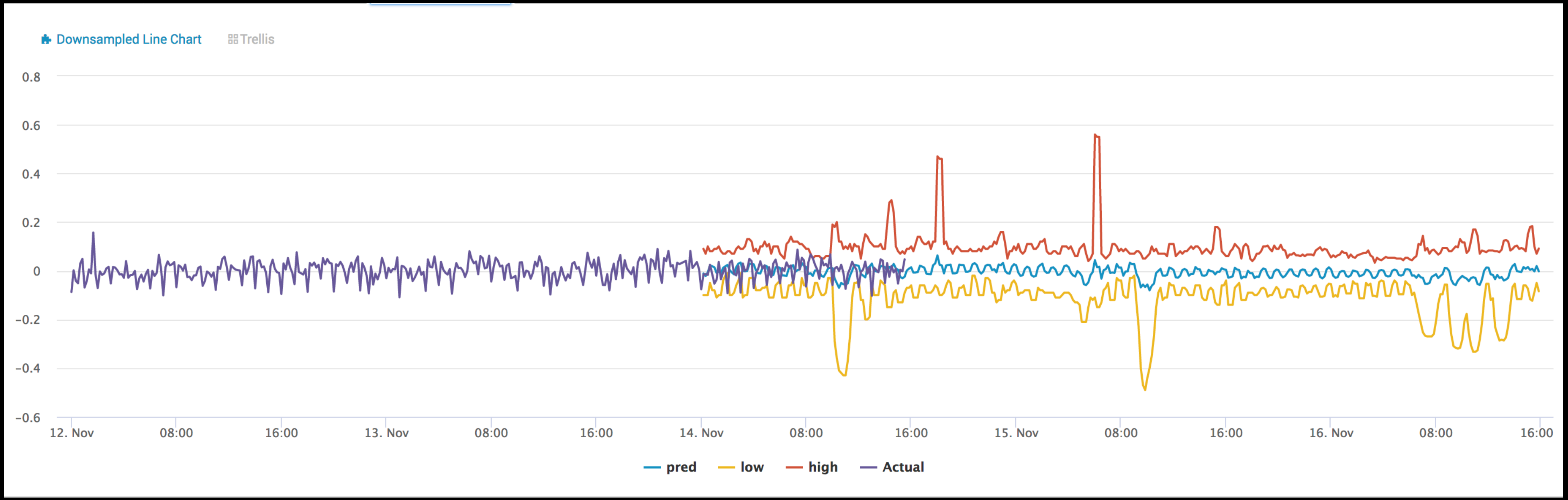


https://www.splunk.com/blog/2018/02/05/cyclical-statistical-forecasts-and-anomalies-part-2.html

Forecasting PCR

Compare prediction to observation for anomaly detection

- Store prediction into a summary index
- Store actual metric in to the same summary index or compare in real time
- Compare observation to prediction
  - Alert when observation is outside the upper or lower bounds of the prediction



https://www.splunk.com/blog/2018/03/20/cyclical-statistical-forecasts-and-anomalies-part-3.html

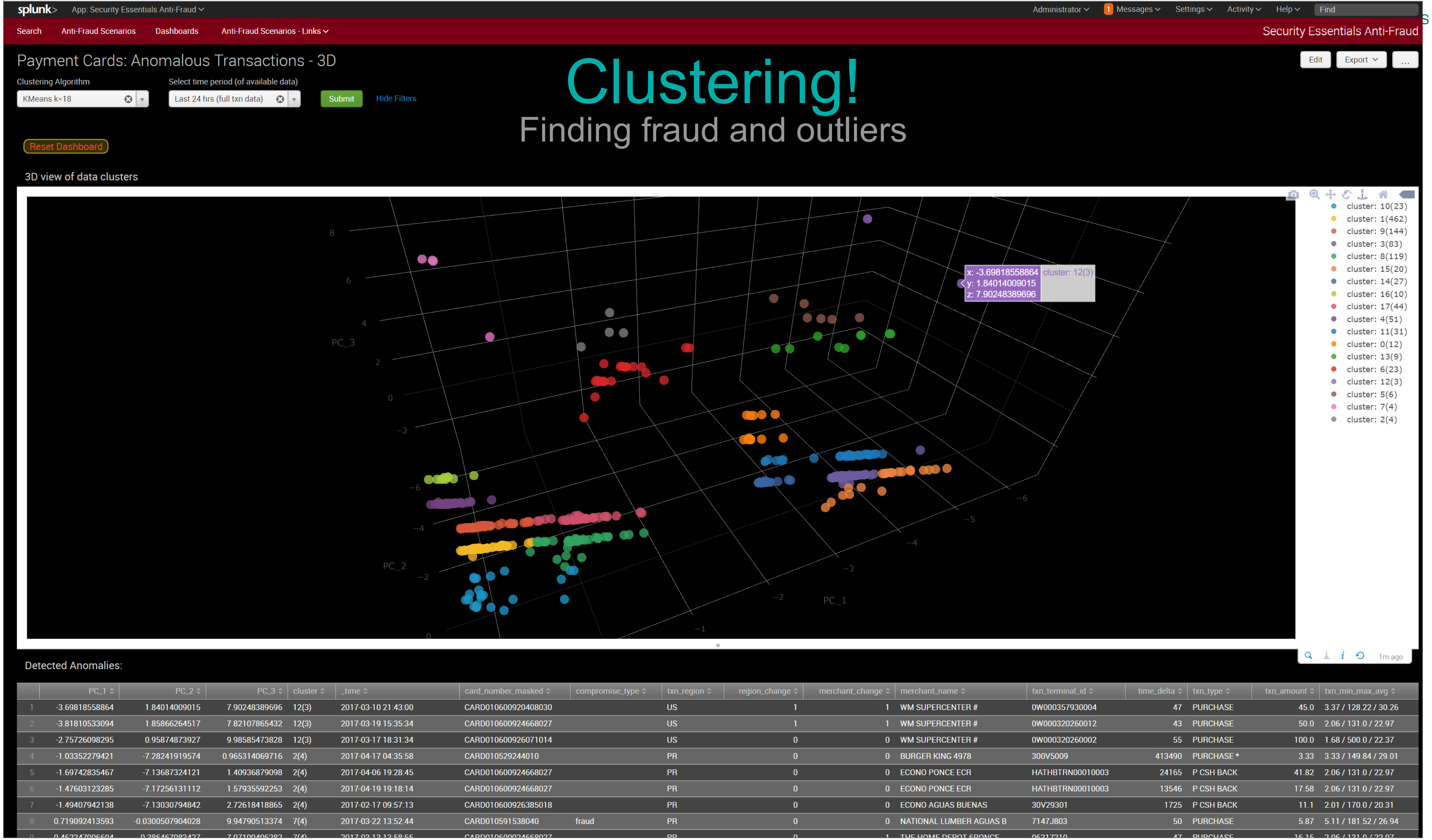
Pattern #13: Peer Group Analysis

- I want to detect someone <doing something that their peers don't do>
- Examples:
  - Detect someone who is
    - Logging in to more servers than others
    - Printing more files than others
    - Uploading more data than others
    - **Logged in to a server for the first time that others also haven't logged in to before**
- SPL

```
sourcetype=win*security
| stats earliest(_time) as earliest latest(_time) as latest by user, dest
| inputlookup append=t sample_cache_group.csv
| stats min(earliest) as earliest max(latest) as latest by user, dest
| outputlookup sample_cache_group.csv
| lookup peer_group.csv user OUTPUT peergroup
| makemv peergroup delim=","
| multireport
    [| stats values(*) as * by user dest ]
    [| stats values(eval(if(earliest>=relative_time(now(),"-1d@d"),dest ,null))) as peertoday
values(eval(if(earliest<relative_time(now(),"-1d@d"),dest ,null))) as peerpast by peergroup dest ]
| eval user=coalsce(user, peergroup)
| fields - peergroup
| stats values(*) as * by user dest
| where isnotnull(earliest)
| isOutlier= if(isnotnull(earliest) AND earliest>=relative_time(now(),"-1d@d") AND isnull(peerpast),1,0)
```

Clustering!

Finding fraud and outliers



Payment Cards: Anomalous Transactions - 3D

Clustering Algorithm: KMeans k=18

Select time period (of available data): Last 24 hrs (full txn data)

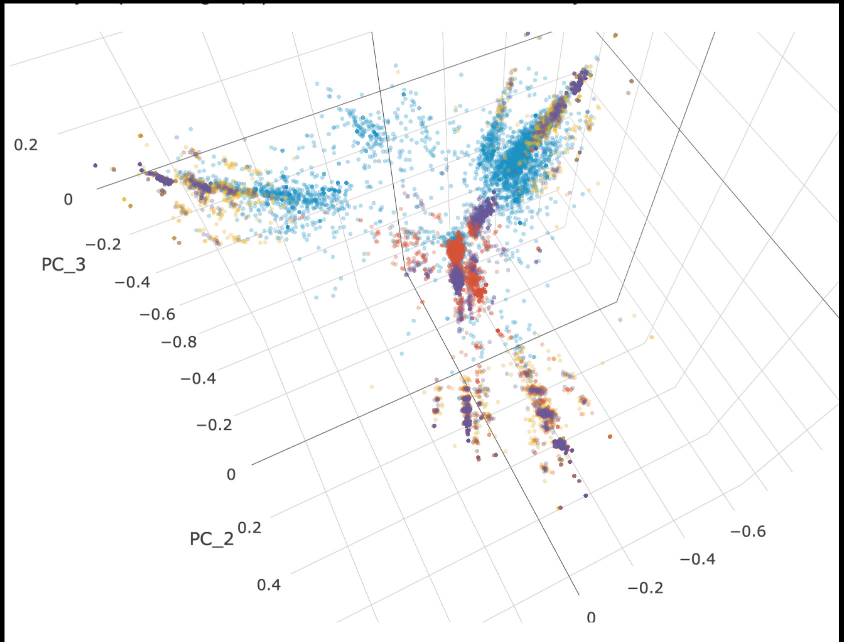
3D view of data clusters

Detected Anomalies:

	PC_1	PC_2	PC_3	cluster	_time	card_number_masked	compromise_type	txn_region	region_change	merchant_change	merchant_name	txn_terminal_id	time_delta	txn_type	txn_amount	txn_min_max_avg
1	-3.69818558864	1.84014009015	7.90248389696	12(3)	2017-03-10 21:43:00	CARD010600920408030		US	1	1	WM SUPERCENTER #	0W000357930004	47	PURCHASE	45.0	3.37 / 128.22 / 30.26
2	-3.81810533094	1.85866264517	7.82107865432	12(3)	2017-03-19 15:35:34	CARD010600924668027		US	1	1	WM SUPERCENTER #	0W000320260012	43	PURCHASE	50.0	2.06 / 131.0 / 22.97
3	-2.75726098295	0.95874873927	9.98585473828	12(3)	2017-03-17 18:31:34	CARD010600926071014		US	0	0	WM SUPERCENTER #	0W000320260002	55	PURCHASE	100.0	1.68 / 500.0 / 22.37
4	-1.03352279421	-7.28241919574	0.965314069716	2(4)	2017-04-17 04:35:58	CARD010529244010		PR	0	0	BURGER KING 4978	300V5009	413490	PURCHASE *	3.33	3.33 / 149.84 / 29.01
5	-1.69742835467	-7.13687324121	1.40936879098	2(4)	2017-04-06 19:28:45	CARD010600924668027		PR	0	0	ECONO PONCE ECR	HATHBTRN00010003	24165	P CSH BACK	41.82	2.06 / 131.0 / 22.97
6	-1.47603123285	-7.17256131112	1.57935592253	2(4)	2017-04-19 19:18:14	CARD010600924668027		PR	0	0	ECONO PONCE ECR	HATHBTRN00010003	13546	P CSH BACK	17.58	2.06 / 131.0 / 22.97
7	-1.49407942138	-7.13030794842	2.72618418865	2(4)	2017-02-17 09:57:13	CARD010600926385018		PR	0	0	ECONO AGUAS BUENAS	30V29301	1725	P CSH BACK	11.1	2.01 / 170.0 / 20.31
8	0.719092413593	-0.0300507904028	9.94790513374	7(4)	2017-03-22 13:52:44	CARD010591538040	fraud	PR	0	0	NATIONAL LUMBER AGUAS B	7147J803	50	PURCHASE	5.87	5.11 / 181.52 / 26.94
9	0.462247006604	0.386467083427	7.07199406282	7(4)	2017-03-13 13:58:56	CARD010600924668027		PR	0	1	THE HOME DEPOT 6PONCE	06317210	47	PURCHASE	16.15	2.06 / 131.0 / 22.97

Pattern #14: Numeric Clustering with MLTK

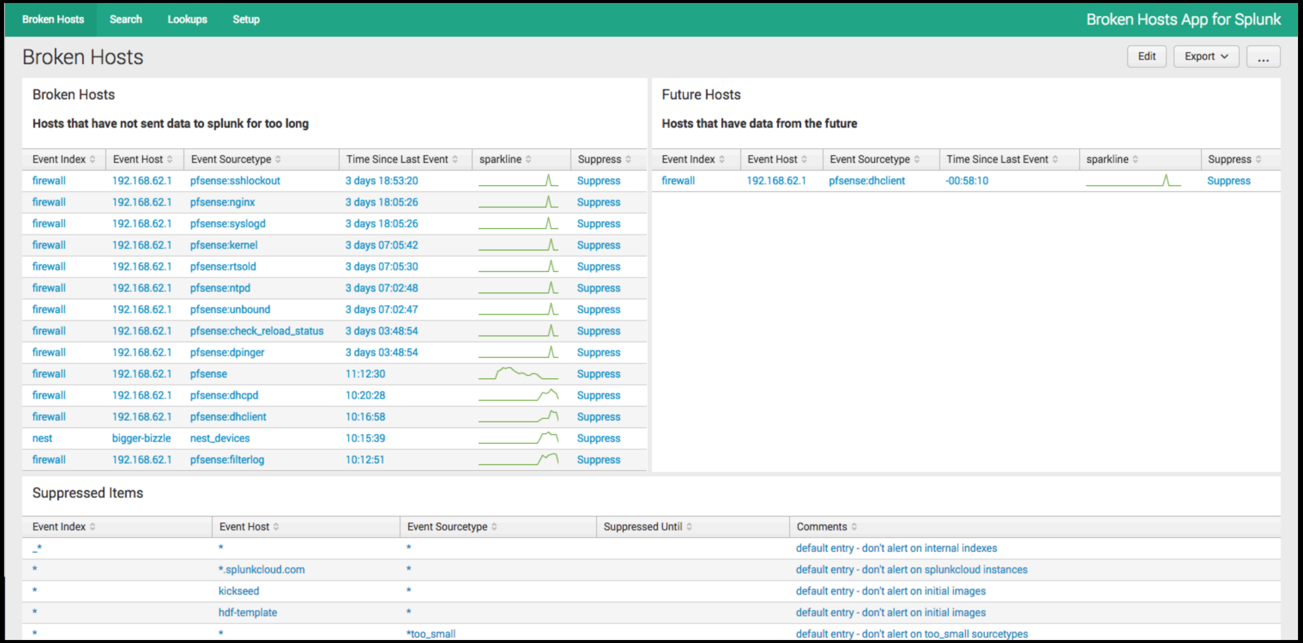
- I don’t know what is normal and what isn’t normal, as my user’s behaviors vary wildly – but I need smarter anomaly detection than previous patterns provide
- Examples:
  - Modeling acceptable behavior within a major business application, like Salesforce
  - Entity group analysis & market segmentation
- SPL



```
| inputlookup sfdc_aggregated_data.csv
| eventstats avg(*) as AVG_* stdev(*) as STDEV_* by USER_ID
| foreach *
    [ eval "Z_<>" = ('<>' - 'AVG_<>' ) / 'STDEV_<>']
| fields - AVG_* STDEV_*
| fillnull
| fit PCA k=5 Z_*
| fit KMeans k=5 PC_*
| eventstats max(clusterDist) as maxdistance p25(clusterDist) as p25_clusterDist p50(clusterDist) as p50_clusterDist p75(clusterDist) as p75_clusterDist dc(USER_ID) as NumIDs count as NumEntries by cluster
| eval MaxDistance_For_IQR= (p75_clusterDist + 12 * (p75_clusterDist - p25_clusterDist))
| where NumEntries < 5 OR clusterDist > MaxDistance_For_IQR
```

Pattern #15: Missing Data

- I want to know when one of my security data sources go quiet
- Examples:
  - A network partition has separated one of my data centers from my Splunk environment
- Broken Hosts App for Splunk
  - https://splunkbase.splunk.com/app/3247/



Broken Hosts

Broken Hosts

Hosts that have not sent data to splunk for too long

Event Index	Event Host	Event Sourcetype	Time Since Last Event	sparkline	Suppress
firewall	192.168.62.1	pfsense:sshlockout	3 days 18:53:20		Suppress
firewall	192.168.62.1	pfsense:nginx	3 days 18:05:26		Suppress
firewall	192.168.62.1	pfsense:syslogd	3 days 18:05:26		Suppress
firewall	192.168.62.1	pfsense:kernel	3 days 07:05:42		Suppress
firewall	192.168.62.1	pfsense:rtsold	3 days 07:05:30		Suppress
firewall	192.168.62.1	pfsense:ntpd	3 days 07:02:48		Suppress
firewall	192.168.62.1	pfsense:unbound	3 days 07:02:47		Suppress
firewall	192.168.62.1	pfsense:check_reload_status	3 days 03:48:54		Suppress
firewall	192.168.62.1	pfsense:dpinger	3 days 03:48:54		Suppress
firewall	192.168.62.1	pfsense	11:12:30		Suppress
firewall	192.168.62.1	pfsense:dhcpd	10:20:28		Suppress
firewall	192.168.62.1	pfsense:dhclient	10:16:58		Suppress
nest	bigger-bizzle	nest_devices	10:15:39		Suppress
firewall	192.168.62.1	pfsense:filterlog	10:12:51		Suppress

Future Hosts

Hosts that have data from the future

Event Index	Event Host	Event Sourcetype	Time Since Last Event	sparkline	Suppress
firewall	192.168.62.1	pfsense:dhclient	-00:58:10		Suppress

Suppressed Items

Event Index	Event Host	Event Sourcetype	Suppressed Until	Comments
_*	*	*		default entry - don't alert on internal indexes
*	*.splunkcloud.com	*		default entry - don't alert on splunkcloud instances
*	kickseed	*		default entry - don't alert on initial images
*	hdf-template	*		default entry - don't alert on initial images
*	*	*too_small		default entry - don't alert on too_small sourcetypes

Key Takeaways

Key Takeaways

- Focus on understanding the patterns that you find most useful, and **then reuse them!**
- Spend time learning the Splunk platform and adding useful things to your toolbox, and then turning them into more use case patterns to leverage in the future
- Take a look at the excellent Splunk Security Essentials app on Splunkbase that has examples of many of the patterns discussed, and a whole lot more!

Q&A

Thank You

Don't forget to rate this session in the .conf18 mobile app

.conf18

splunk>