.conf18

splunk>

Using Splunk ML for Threat Hunting

Joe Partlow, ReliaQuest

>whoami

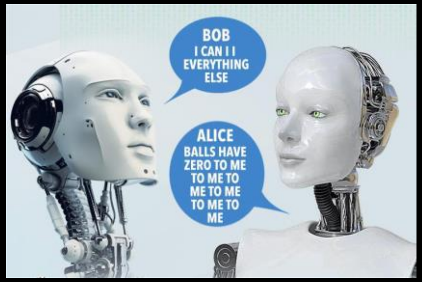
Joe Partlow – CTO, ReliaQuest

Joe has been in the IT and informaxtion Security industry for 20+ years and most recently been working with simulated attack & defense networks, security analytics, building big data platforms and machine learning. Reliaquest partners with the worlds largest enterprise splunk customers performing analyst, engineering/architecture and content development functions.

Inspired by gibson

What ml/ai is not

- Algos will not replace good analysts, just another resource
  - But when can I have wintermute in my soc???
- Many times, successful hunt campaigns achievable with just effective searches
- Many good products/models fail because of poor and incomplete data
- Haven’t even scratched the surface of basic ML, let alone “ai” …. yet
- Gone too soon:


BOB
I CAN I I EVERYTHING ELSE
ALICE
BALLS HAVE ZERO TO ME TO ME TO ME TO ME TO ME

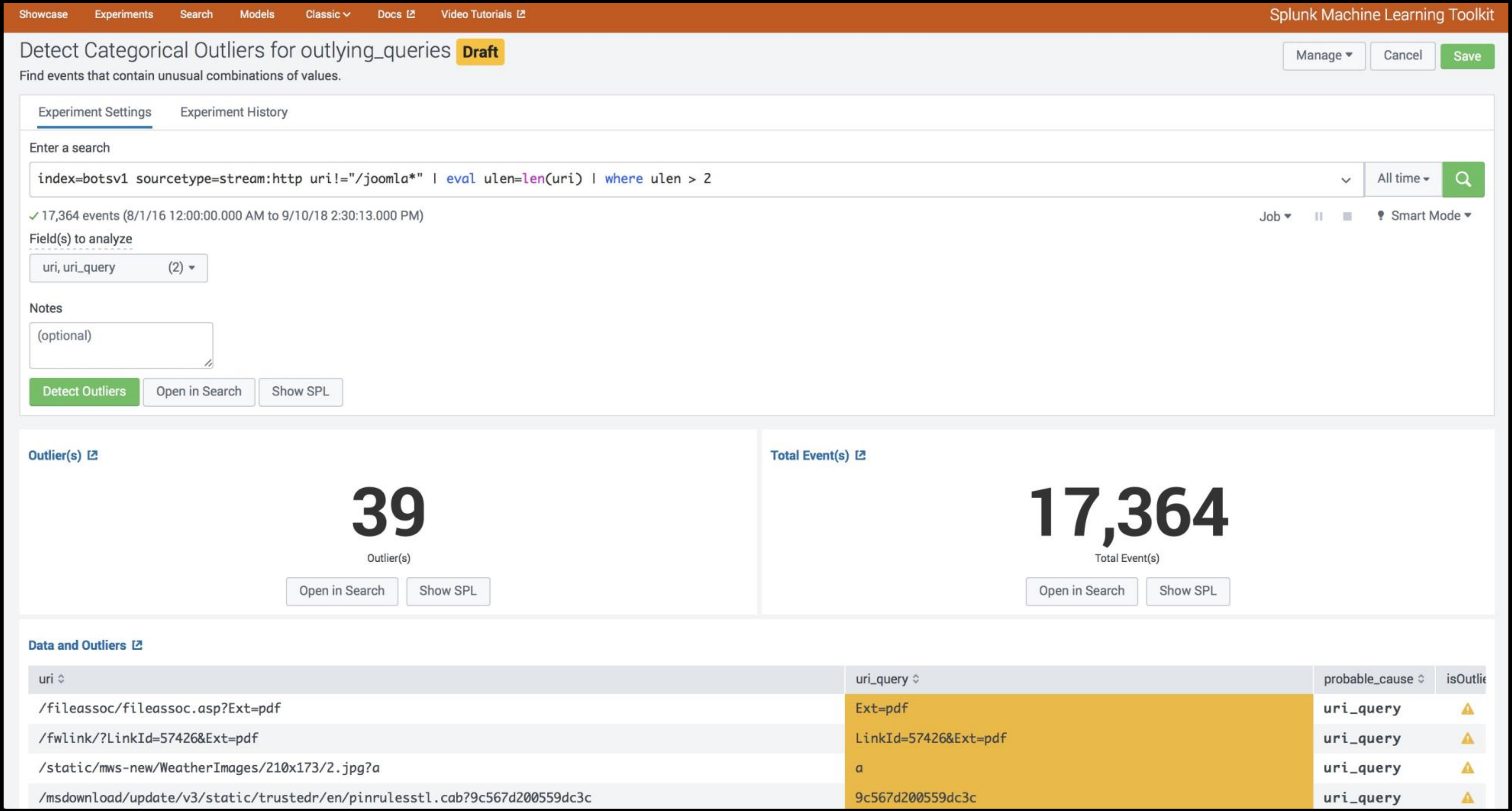
Ensuring success

- Clean data will make or break your training models
  - Include filtering, black/white lists to remove large datasets that could skew results
- Data normalization across the various sources
- Know your data! (supervised and unsupervised learning)
- Numeric data shouldn’t always be treated as such (ie. port numbers)
- Incorporate red teamers with your data science team for better domain knowledge
- Continuously retrain your models as the environment changes

(most) Relevant algorithms

- YMMV but below are some algorithms well-suited for common hunt campaigns:
  - Detect Numeric Outliers – Useful for determining weird status codes or abnormal event IDs
  - Detect Categorical Outliers – Useful for finding weird DNS queries or abnormal user-agents/page requests
  - Cluster Numeric Events – Helpful for finding outlying host login counts or application usage counts
  - Prediction and Forecasting algorithms might be better suited until the environment is stable and baselined

Web attack use case



Detect Categorical Outliers for outlying_queries **Draft**

Find events that contain unusual combinations of values.

Enter a search

```
index=botsv1 sourcetype=stream:http uri!="/joomla*" | eval ulen=len(uri) | where ulen > 2
```

✓ 17,364 events (8/1/16 12:00:00.000 AM to 9/10/18 2:30:13.000 PM)

Field(s) to analyze: uri, uri_query (2)

Notes: (optional)

Outlier(s)

39

Outlier(s)

Total Event(s)

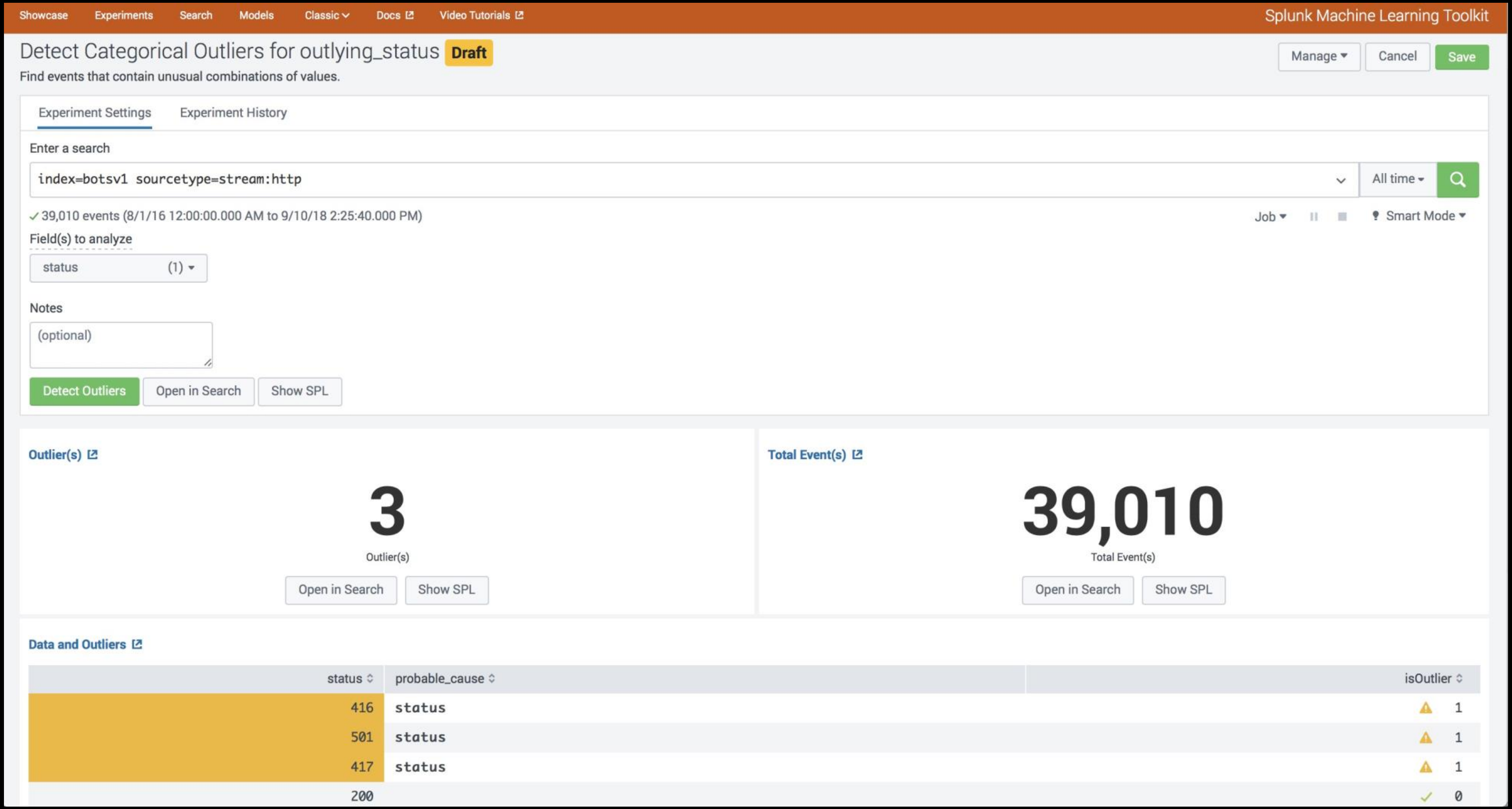
17,364

Total Event(s)

Data and Outliers

uri	uri_query	probable_cause	isOutlie
/fileassoc/fileassoc.asp?Ext=pdf	Ext=pdf	uri_query	⚠
/fwlink/?LinkId=57426&Ext=pdf	LinkId=57426&Ext=pdf	uri_query	⚠
/static/mws-new/WeatherImages/210x173/2.jpg?a	a	uri_query	⚠
/msdownload/update/v3/static/trustedr/en/pinrulesstl.cab?9c567d200559dc3c	9c567d200559dc3c	uri_query	⚠

Web attack use case



Detect Categorical Outliers for outlying_status **Draft**

Find events that contain unusual combinations of values.

Enter a search

```
index=botsv1 sourcetype=stream:http
```

✓ 39,010 events (8/1/16 12:00:00.000 AM to 9/10/18 2:25:40.000 PM)

Field(s) to analyze: status (1)

Notes: (optional)

Outlier(s)

3

Outlier(s)

Total Event(s)

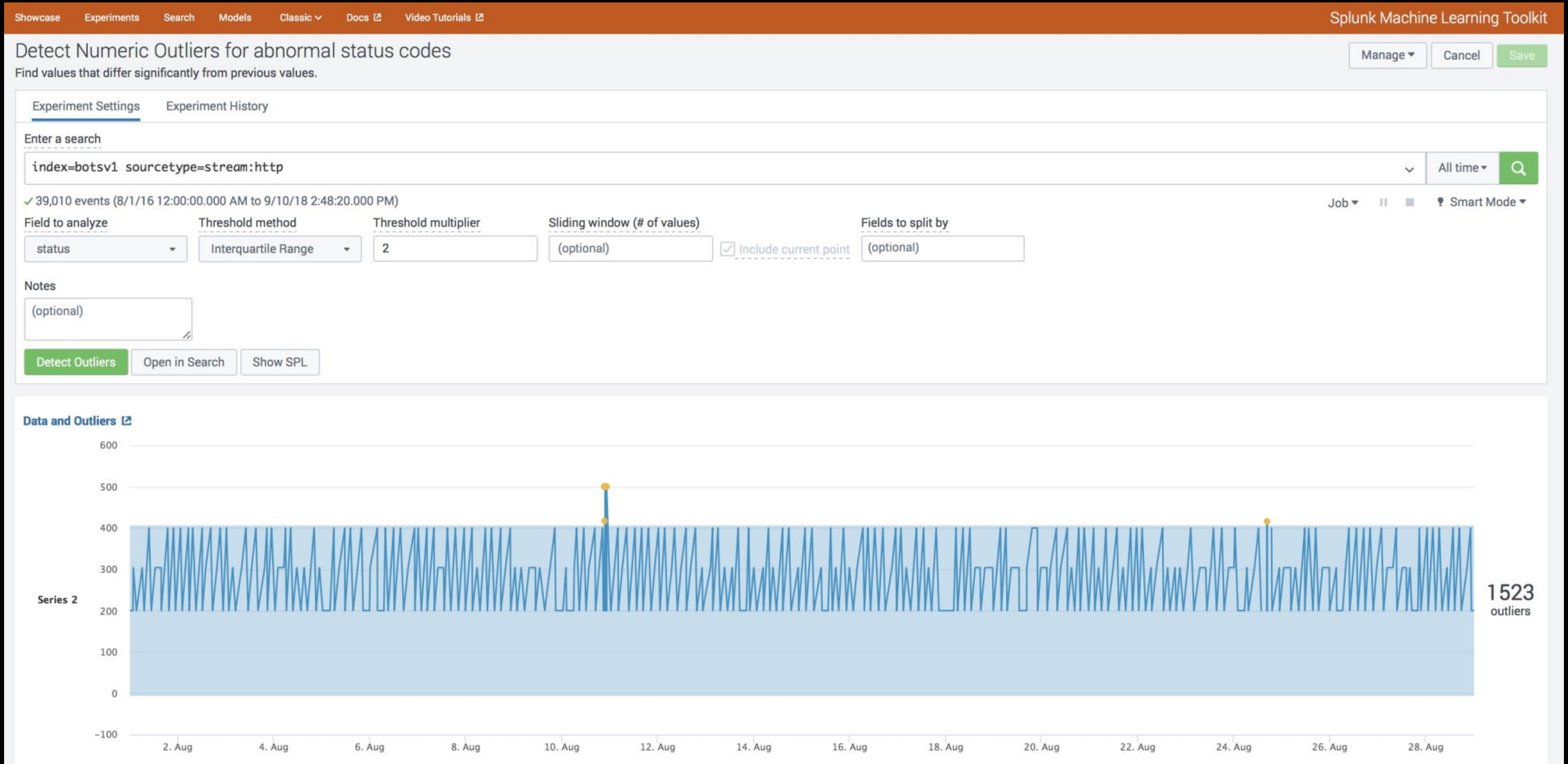
39,010

Total Event(s)

Data and Outliers

status	probable_cause	isOutlier
416	status	⚠ 1
501	status	⚠ 1
417	status	⚠ 1
200		✓ 0

Web attack use case



Detect Numeric Outliers for abnormal status codes

Find values that differ significantly from previous values.

Enter a search

```
index=botsv1 sourcetype=stream:http
```

✓ 39,010 events (8/1/16 12:00:00.000 AM to 9/10/18 2:48:20.000 PM)

Field to analyze: status

Threshold method: Interquartile Range

Threshold multiplier: 2

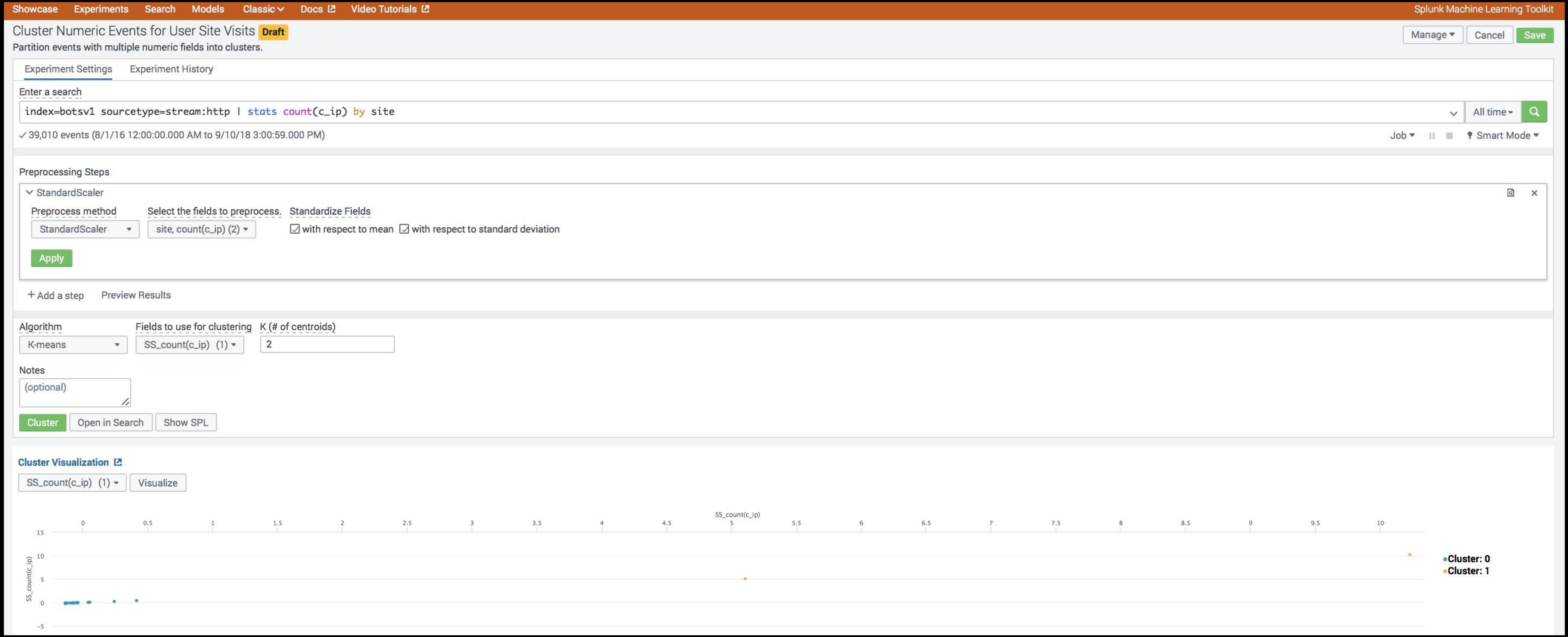
Sliding window (# of values): (optional)

☑ Include current point

Fields to split by: (optional)

Notes: (optional)

Web attack use case



Cluster Numeric Events for User Site Visits **Draft**

Partition events with multiple numeric fields into clusters.

Enter a search

```
index=botsv1 sourcetype=stream:http | stats count(c_ip) by site
```

✓ 39,010 events (8/1/16 12:00:00.000 AM to 9/10/18 3:00:59.000 PM)

Preprocessing Steps

StandardScaler

Preprocess method: StandardScaler

Select the fields to preprocess.: site, count(c_ip) (2)

Standardize Fields: ☑ with respect to mean ☑ with respect to standard deviation

Algorithm: K-means

Fields to use for clustering: SS_count(c_ip) (1)

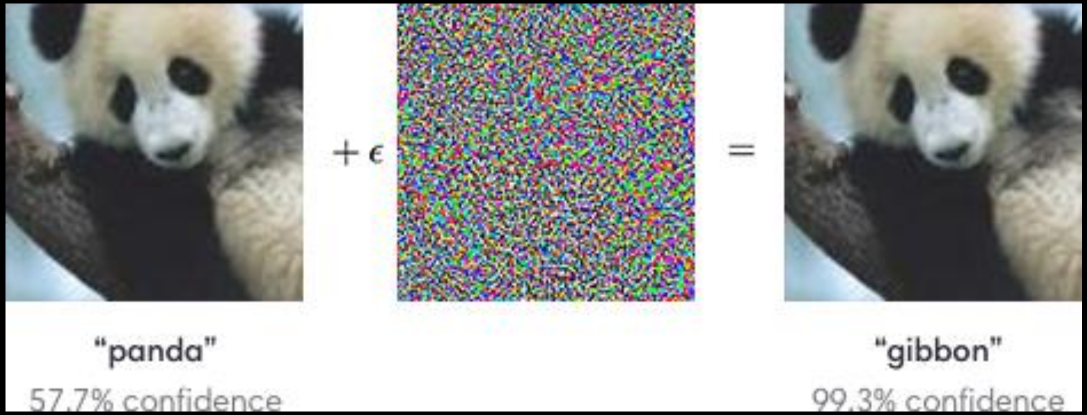
K (# of centroids): 2

Notes: (optional)

SS_count(c_ip) (1)

Breaking the models

- Already proven for image classification deep learning*
- Similar to SIEM issues, overload the data ingestion with enough noise that the “abnormal becomes normal”
- Attackers are already good at blending in (living off the land, pivoting, etc.)



* *https://blog.openai.com/adversarial-example-research/*

Future enhancements

- Field is progressing amazingly fast. Just because something isn’t possible now, give it 6 months!
- Move towards stacked/ensemble learning to avoid “jack of all trades, master of none” algos
- Build up and better utilize belief networks
  - Attempts to increase accuracy by adding conditional dependencies
  - Excellent blackhat talk by raffael marty - https://www.slideshare.net/zrlram

THANK YOU

Questions? jpartlow@reliaquest.com

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