Forward-looking statements

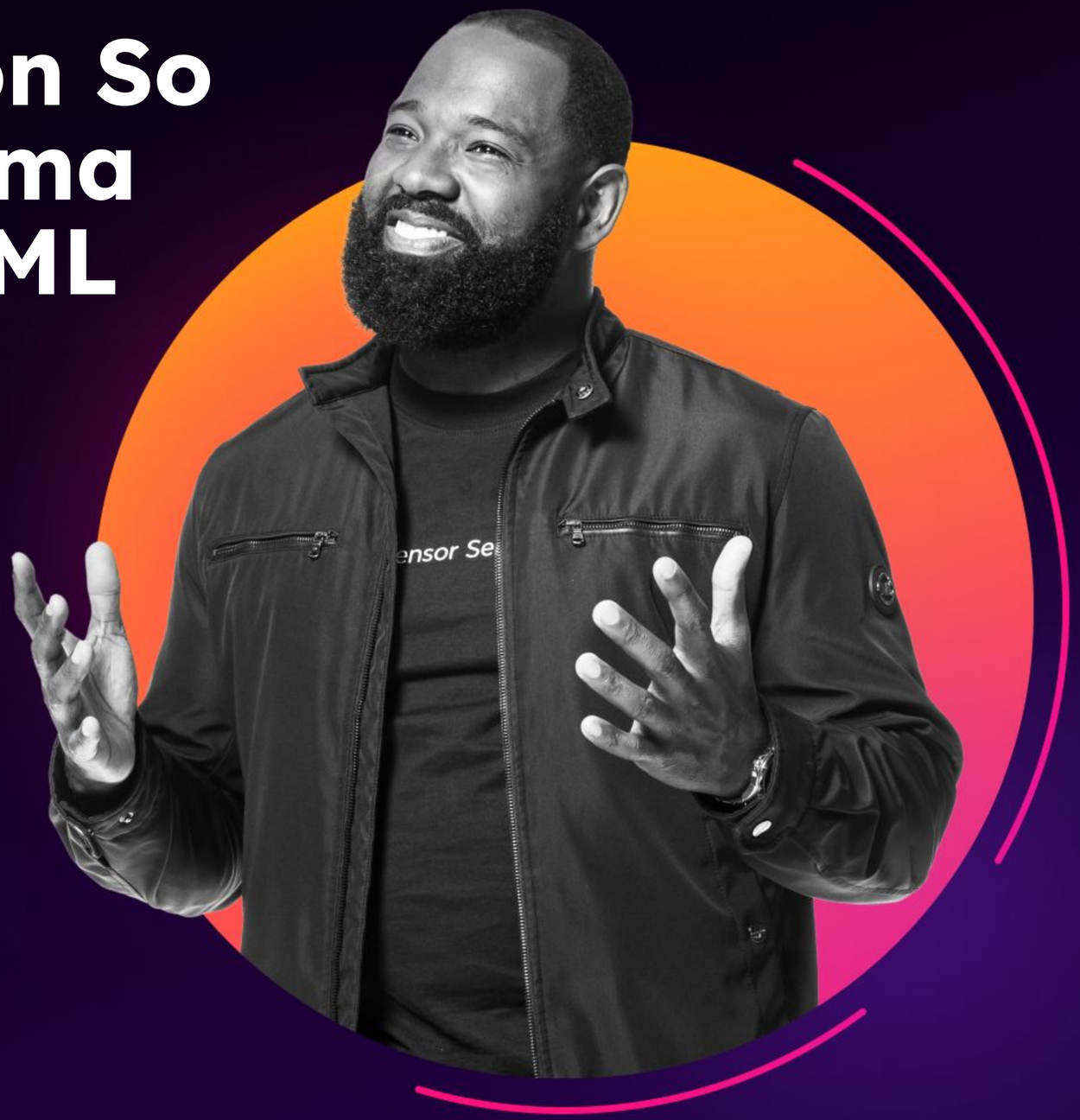
This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's 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.

Anomaly Detection So Easy, Your Grandma Could Do It - No ML Degree Required

PLA1149C



.conf24 splunk>

Bring on the future.

Troy Moore

Log Analysis Made Easy (L.A.M.E.) Creations

lamecreations@gmail.com

@lamecreations_guides - Youtube

discord.gg/k5M6eme2CK

Table of Contents

Baselining

a baseline is the expected values or conditions against which all performances are compared.

Common Baselines

Hardware baselines

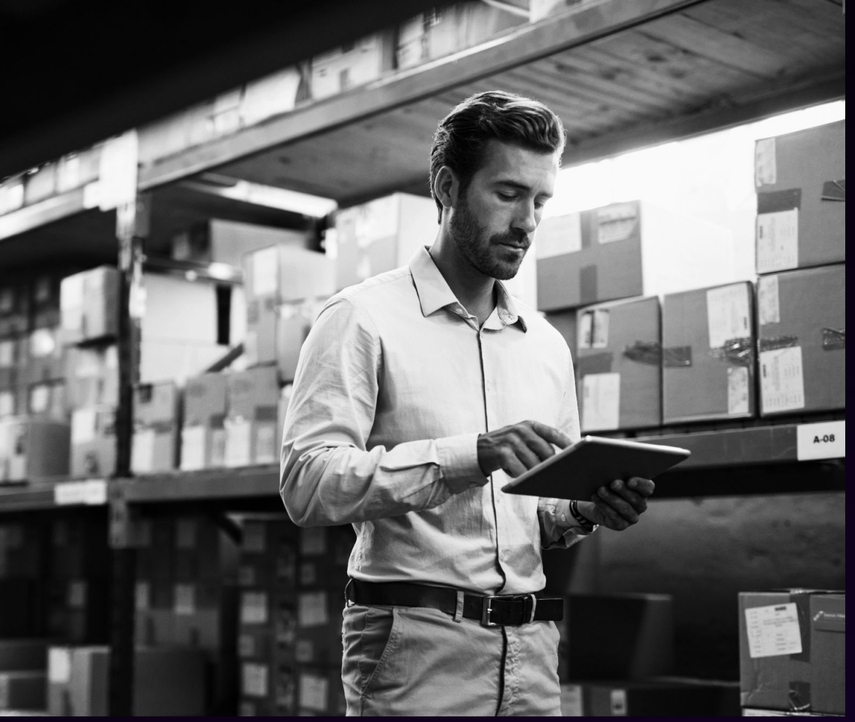
Software baselines

Network ports and protocol baselines

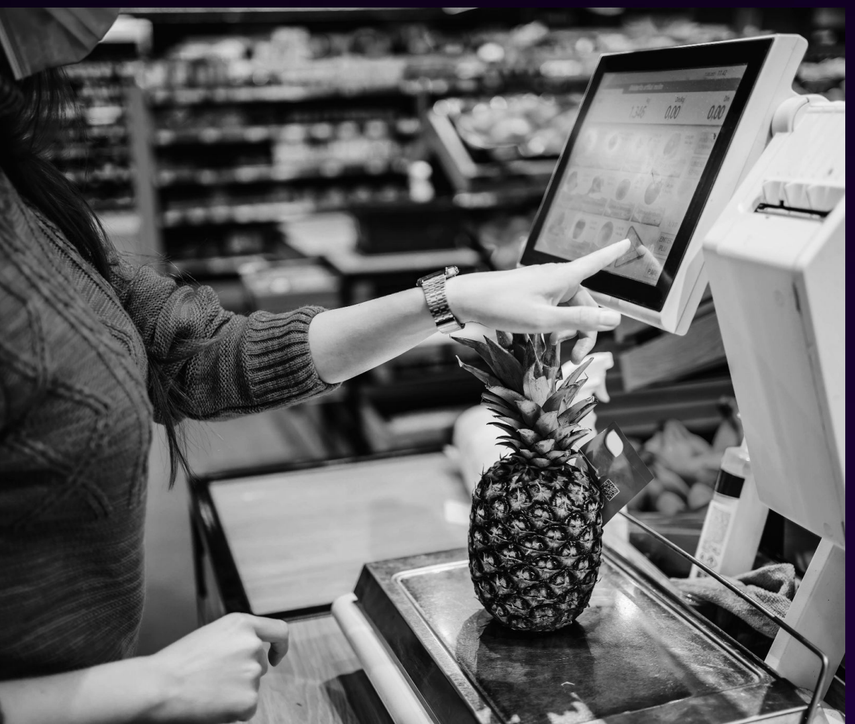
User baselines

Who do I ask for these baselines?

- Ever asked anyone for their current network inventory?
- Or any of the example baselines?



Splunk & Statistical Models Make You That Hero



Look to the Past

Use historical IP connections

Track historical processes

Track the ports used by systems historically

Track historical log-in events

Two Simple Methods For Baselining

Rolling Window

Query a period of time
X is the baseline
Y is a period we are assessing for anomalies

If value appears in Y but not in X, this is new and flagged as anomalous

Allow Listing

Create an allow list CSV for the baseline
outputlookup

Use **Lookup** command on all new logs to check if value is not in lookup. Values not in lookup are flagged as anomalous

Demo Time

Let’s see this anomaly detection in Splunk
Queries are at end of presentation

Demo Video Summary

Previous Ten Minute Video Demonstrated the following:

Using Stats Command to Baseline Normal Behavior from Historical Data.

Use that baseline to determine new events that are anomalous.

- detected anomalous network connections
- detected anomalous processes on host
- detected anomalous open ports on host

Created CSV Lookup for Baselining Normal Behavior from Historical Data

Use lookup command to detect new events that are anomalous

- detected anomalous network connections
- detected anomalous processes on host
- detected anomalous open ports on host

All queries used in video demo are at the end of the slide deck

Gotchas for Baselining

Rolling Windows and Allowlists have Pros and Cons in their Creation

Rolling Window

- You will be alerted once on the anomalous connection and then the anomaly becomes part of the baseline
- The frequency of times this alert is run in a day can have an impact on services. If you use 90 days of data and a 1 hour window or a 1 day window, the query takes ~ same time to run

Allowlist

- Will notify you each time an anomalous event is seen until you update the CSV
- Only query time since last time the query ran, which means it will run faster the more often you run it.
- Baseline might become very large, which may mean you want to use a compressed zip or KV store

Hybrid Approach

A combination of Rolling and Allowlisting

- **Use Collect Command to Write Results to Index**

  [Your Query Goes Here]
  | collect index=summary source=[your_custom_name]

- **Update Lookup File**

  index=summary source=[your_custom_name| append [inputlookup [allow_list.csv] | table [matching fields]
  | stats count by [matching fields]
  | outputlookup [allow_list.csv]

- **New Hybrid Query**

  index=botsv3 sourcetype=PerfmonMk:Process

  | stats min(_time) as earliestTime max(_time) as latestTime count by instance, host

  | eval nowTime=1534774680

  | lookup [allow_list..csv] instance as instance host as host output instance as recurring

  | where earliestTime > nowTime OR (recurring="*" AND latestTime > nowTime)

Hybrid Approach Demo Time

Let’s see the hybrid approach

Queries are at end of presentation

Hybrid Approach Video Summary

Previous Video Demonstrated the following:

Combine the two approaches

Create Lookup of “Anomalous Behaviors” so they don’t get excluded

Use the results from one method against to other to validate how much “change” exists in your environment.

All queries used in video demo are at the end of the slide deck

What’s next?

What logs do you have in Splunk that you could baseline?

Try out the rolling window and lookup approach on that data.

Thank you

Troy Moore

Log Analysis Made Easy (L.A.M.E.) Creations

lamecreations@gmail.com

@lamecreations_guides - Youtube

discord.gg/k5M6eme2CK

Appendix

Botsv3 dataset

https://github.com/splunk/botsv3

- Allows everyone to use same source to repeat my queries
- Date is set as all time with an epoch time for NowTime field. This is to account for a historical set of data.
- Not all examples are perfect and usually you want a larger rolling window, but it’s good enough

Example Rolling Window Network Baseline

```
index=botsv3 sourcetype=stream:tcp
| stats min(_time) as earliestTime, count by src_ip, dest_ip
| eval nowTime=1534774680
| where earliestTime > nowTime
```

Example for running this alert daily

```
index=botsv3 sourcetype=stream:tcp
| stats earliest(_time) as earliestTime count by src_ip, dest_ip
| where earliestTime > now() - 86400
```

Example Rolling Window Process Baseline

```
index=botsv3 sourcetype=PerfmonMk:Process
| stats min(_time) as earliestTime count by instance, host
| eval nowTime=1534774680
| where earliestTime > nowTime
```

Example for running this alert daily

```
index=botsv3 sourcetype= PerfmonMk :Process
| stats earliest(_time) as earliestTime count by instance, host
| where earliestTime > now() - 86400
```

Example Rolling Window Port Baseline

```
index=botsv3 sourcetype=Script:ListeningPorts
| stats min(_time) as earliestTime count by host, dest_port
| eval nowTime=1534774680
| where earliestTime > nowTime
```

Example for running this alert daily

```
index=botsv3 sourcetype= Script:ListeningPorts
| stats earliest(_time) as earliestTime count by host, dest_port
| where earliestTime > now() - 86400
```

Allowlist Network Baseline

```
index=botsv3 sourcetype=stream:tcp
| stats count by src_ip, dest_ip
| outputlookup network_baseline_allowlist.csv
```

Example for running this alert daily

```
index=botsv3 sourcetype=stream:tcp
| stats count by src_ip, dest_ip
| lookup network_baseline_allowlist.csv src_ip as src_ip, dest_ip as dest_ip output count as matched
| where isnull(matched)
```

Allowlist Process Baseline

```
index=botsv3 sourcetype=PerfmonMk:Process
| stats earliest(_time) as earliestTime count by instance, host
| outputlookup process_baseline_allowlist.csv
```

Example for running this alert daily

```
index=botsv3 sourcetype=PerfmonMk:Process
| stats count by instance, host
| lookup process_baseline_allowlist.csv instance as instance, host as host output count as matched
| where isnull(matched)
```

Allowlist Port Baseline

```
index=botsv3 sourcetype=Script:ListeningPorts
| stats count by host, dest_port
| outputlookup port_baseline_allowlist.csv
```

Example for running this alert daily

```
index=botsv3 sourcetype=Script:ListeningPorts
| stats count by host, dest_port
| lookup port_baseline_allowlist.csv dest_port as dest_port, host as host output count as matched
| where isnull(matched)
```

Hybrid Approach

Example for running this alert daily

```
index=botsv3 sourcetype=PerfmonMk:Process
| stats earliest(_time) as earliestTime count by instance, host
| where earliestTime > 1534774680
| table instance, host, earliestTime
| collect index=summary source=”new_process”

index=summary source=”new_process”
| append
[inputlookup new_process.csv | table instance, host ]
| stats count by instance, host
| outputlookup new_process.csv
```

Hybrid Approach Part 2

Example for running this alert daily

```
index=botsv3 sourcetype=PerfmonMk:Process
| stats min(_time) as earliestTime max(_time) as latestTime count by instance, host
| eval nowTime=1534774680
|  lookup new_process.csv instance as instance host as host output instance as recurring
| where earliestTime > nowTime OR (recurring="*" AND latestTime > nowTime)
```