EDINBURGH
JUNE 16-21
2019

Optimized Playbook, Roll out!

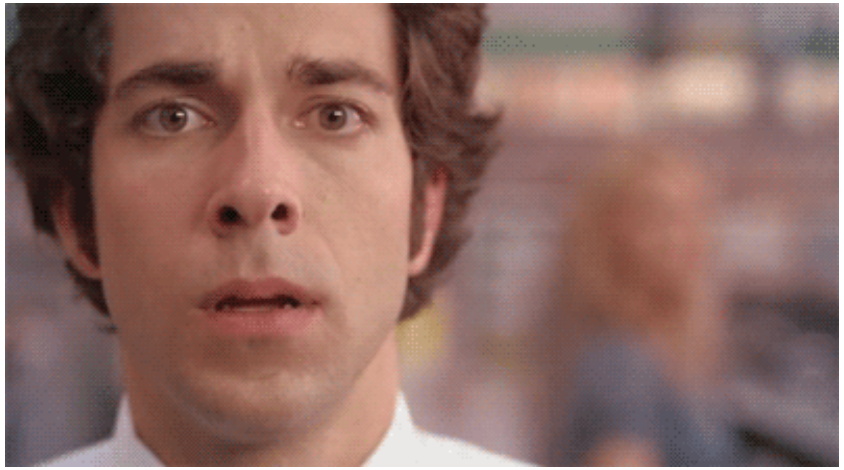
How an optimized playbook can reduce time-to-detect

Chris Merida

Jason Kmack

Who are these guys?

- Chris Merida
- InfoSec Engineer for Cisco CSIRT
- Focus on operations for SIEM, web, and special projects
- Likes angry music that you only find on Myspace
- Usually wearing a hat
- Enjoys coffee and fine liquor (occasionally at the same time)



Who are these guys?

- Jason Kmack
- InfoSec Engineer for Cisco CSIRT
- Application developer focusing on security monitoring tools
- Often found near empty pints of micro-brews or bottles of bourbon

First things first…

- Play
  - A query to look for specific indicators of compromise
- Playbook
  - A collection of plays
- Time-to-detect (TTD)
  - Amount of time between a compromise occurring and it being discovered
- SIEM
  - Security and information event management
- “Indexes”
  - Indices

Why do my queries take so long?!

- Large datasets
  - ~5TB a day across all
- Long retrospective searches
  - Last 90 days or longer
- Poorly written queries
  - “index=* badguys.com earliest=-500y”
- A LOT of queries running simultaneously
  - 970 play runs per day

How can we solve this?

Not going to work

- ROI on hardware is less than on process improvement
- Not enough time to rewrite manually
  - Over 360 plays to go back and fix
- Best practice guide won't be leveraged

Might work!

- Interactive script to suggest changes to the play owner
- Integration with playbook tool to run optimized query automatically
- Reusable query templates

Don’t overthink this…

- What if we
  - Run query through Splunk’s built-in optimizer multiple times to get to its final form
  - Calculate tons of statistics for query runtimes to determine what is within acceptable bounds
  - Compile results to make the query better based off of the above outputs
    - Then we can suggest the best possible query to the user
      - Then we can ask them change it based off our suggestion!
        - Then we wait for them to change it
          - But they don’t have time to change it
            - This is not a good idea…

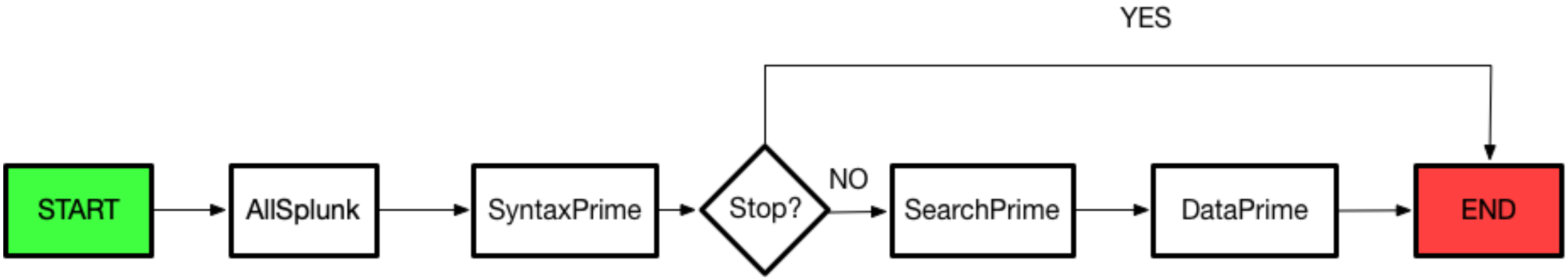
Defining the requirements…

- Reduce performance impact on Splunk
- Make plays run faster
  - Plays can run faster and more frequently
  - Therefore reduces time-to-detect
- Help analysts/investigator write better queries
- Integrate with playbook automation tool
- Give it a cool name
  - Like OptimizePrime!

Which tools do we use?

- Splunk API
- Python/Django/MySQL
- Gunicorn/Nginx
- Docker
- Splunk query best practices
- Recommendations from personal experience

Overview


YES
START
AllSplunk
SyntaxPrime
Stop?
NO
SearchPrime
DataPrime
END

AllSplunk

- Like the AllSpark!
  - Doesn’t require star sacrifices to run
- Coordinates flow of modules
- Handles collection of results
- Delivers results to UI/DB

SyntaxPrime

- Uses admin page to define custom regular expressions
- Admin can determine whether OptimizePrime stops on a failed check or alerts the user
  - If stop condition is met, no other modules are run
- List of warnings collected and sent back to AllSplunk

Expression	Expression type	Output message	Stop	Active	Created date	Modified date
index\s*=\s**	regex	Use of "index=*" is restricted!	✓	✓	Dec. 13, 2018, 9:19 p.m.	Dec. 13, 2018, 9:19 p.m.
latest\s*=\s*	regex	latest= was not found	⛔	✓	Dec. 13, 2018, 8:13 p.m.	Dec. 13, 2018, 9 p.m.
|\s*vt\s*	regex	Warning: This search has an external lookup (|vt)	⛔	✓	Dec. 12, 2018, 7:40 p.m.	Dec. 12, 2018, 10:48 p.m.
|\s*dce\s*	regex	Warning: This search has an external lookup (dce)	⛔	✓	Dec. 12, 2018, 7:39 p.m.	Dec. 12, 2018, 10:48 p.m.
earliest\s*=\s*	regex	earliest= was not found	✓	✓	Dec. 12, 2018, 7:37 p.m.	Dec. 13, 2018, 8:12 p.m.
index\s*=\s*	regex	No index specified. #FAILED	✓	✓	Dec. 12, 2018, 7:35 p.m.	Dec. 13, 2018, 1:17 a.m.

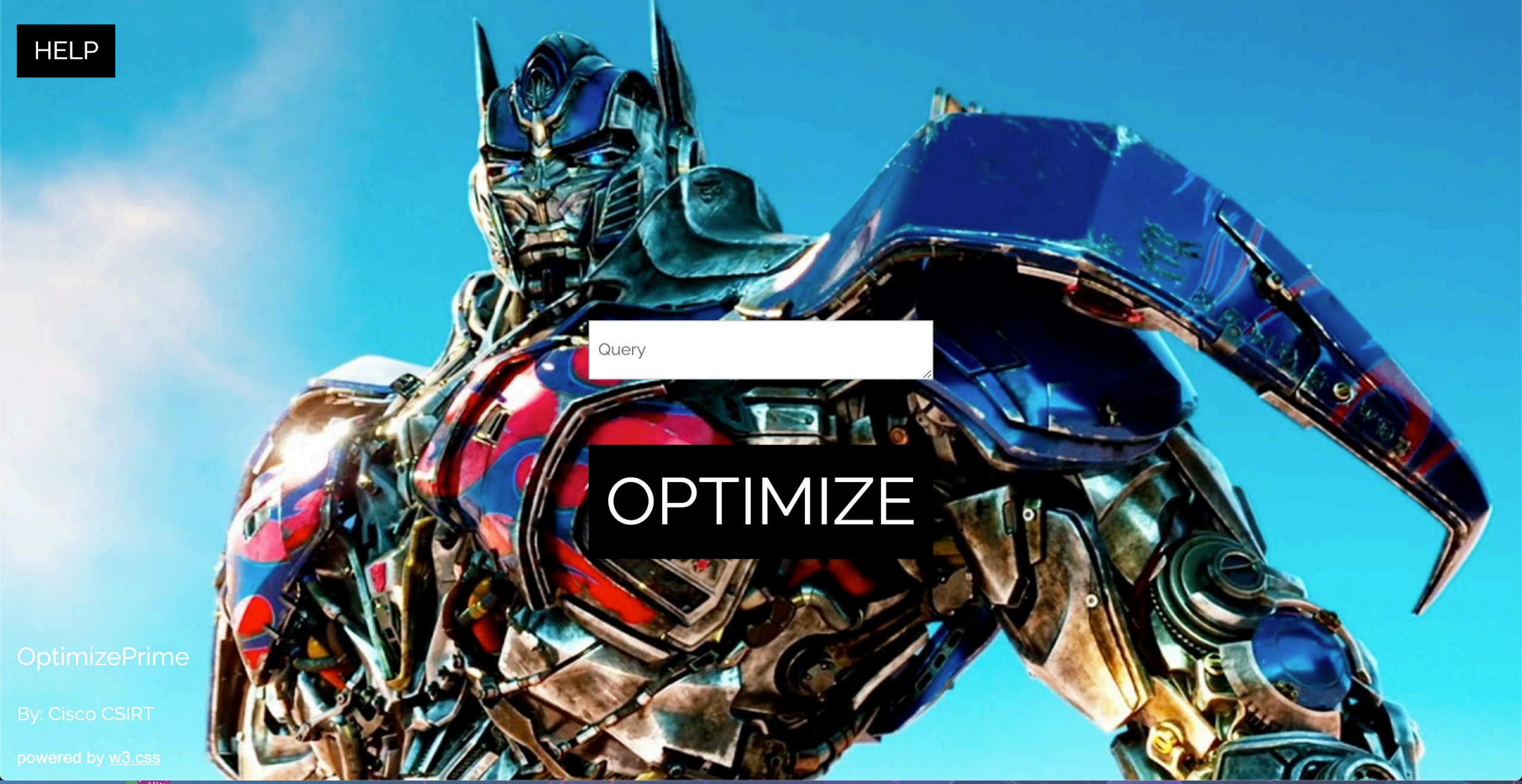
SearchPrime

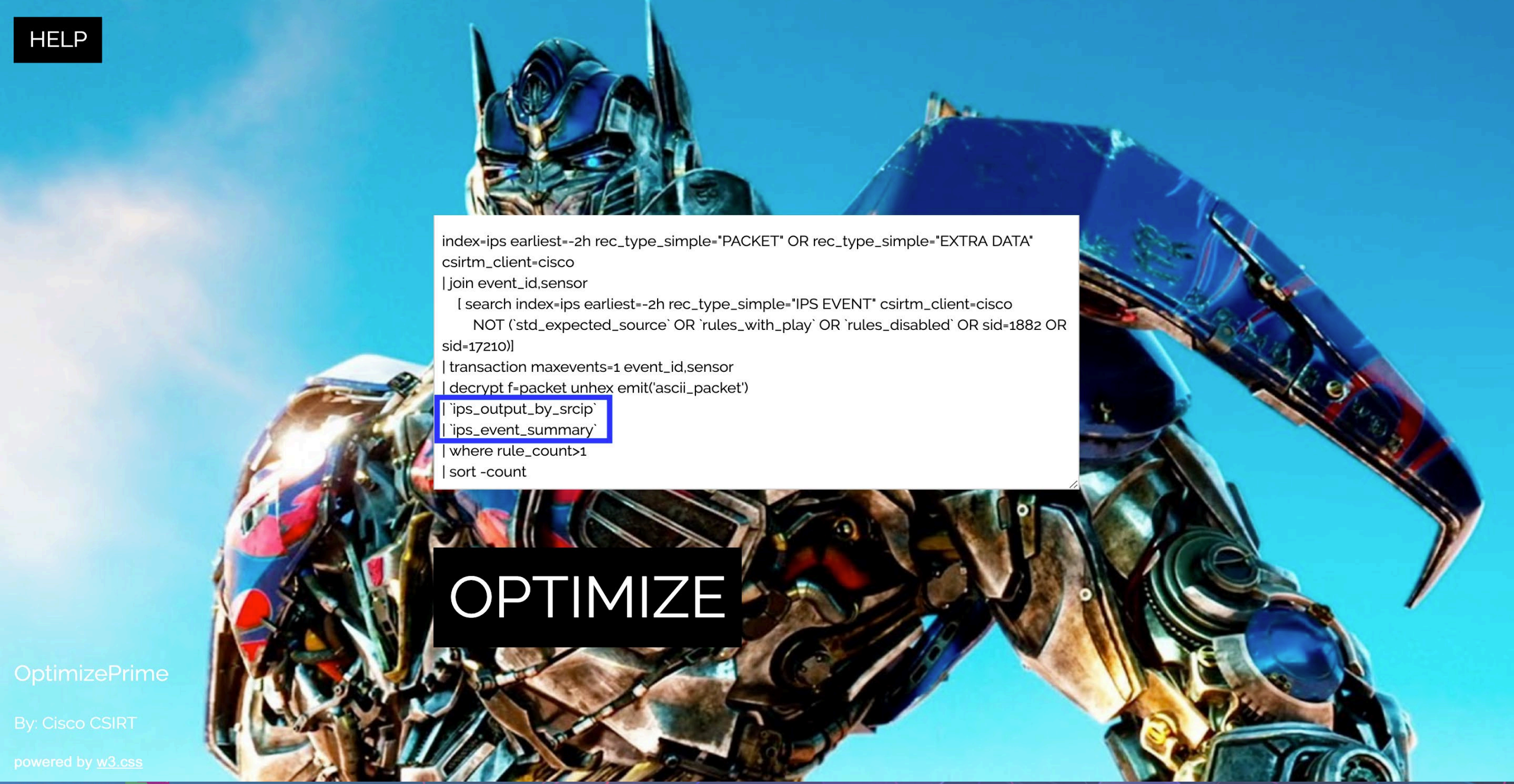
- Receives the candidate search string and runs through normal search
- Collect optimized query and begin running optimized queries in parallel with candidate query
- Receive result sets, calculate runtimes, select fastest query
- Configurable
  - Number of search heads
  - Number of optimized runs (ex. 2 optimized query runs per search head)
  - Enable/Disable multithreading (running optimized queries in parallel with candidate)
  - Timeout for query to be submitted (in case Splunk is super busy)
  - Timeout for query to finish running (we don’t want to wait for hours)
- Added a bonus “Stats for nerds” section

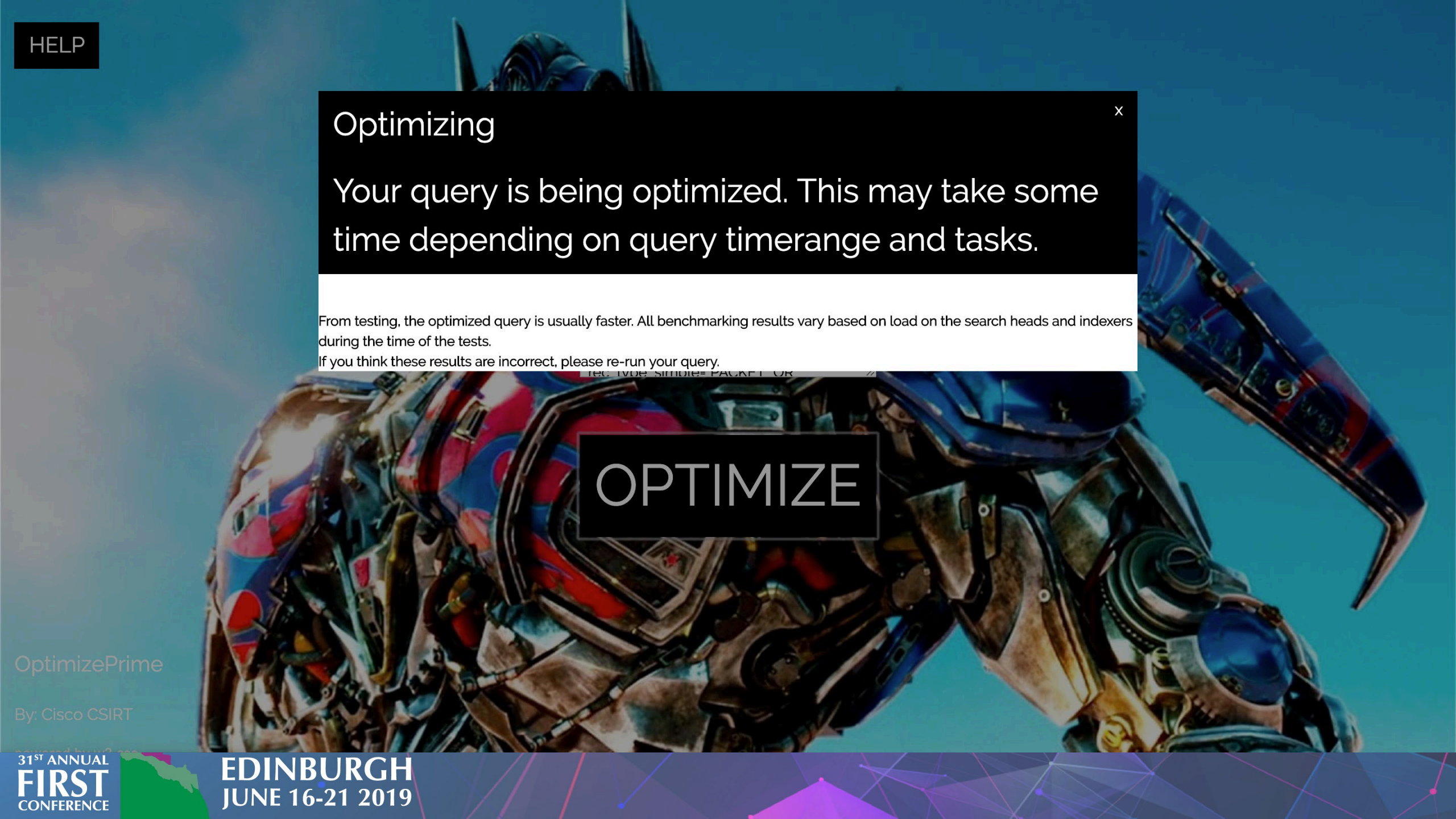
DataPrime

- Uses admin page to define custom thresholds
- Thresholds can be defined by:
  - Size on disk
  - Events
- No stop condition; only warnings
- List of warnings collected and sent back to AllSplunk

Current size	Size threshold	Size active	Count threshold	Count active	Message
27208402	1024000	⛔	5000000	✅	Search is looking at greater than 50M events data.
10059911	(None)	⛔	(None)	⛔	(None)
10005201	1	⛔	1	⛔	
1405101	(None)	⛔	(None)	⛔	(None)
1186320	(None)	⛔	(None)	⛔	(None)
1142917	(None)	⛔	(None)	⛔	(None)
905769	(None)	⛔	(None)	⛔	(None)
554704	(None)	⛔	(None)	⛔	(None)
439591	1	⛔	1000000	✅	Searching greater than 1M events.


HELP
Query
OPTIMIZE
OptimizePrime
By: Cisco CSIRT
powered by w3.css


HELP
index=ips earliest=-2h rec_type_simple="PACKET" OR rec_type_simple="EXTRA DATA"
csirtm_client=cisco
| join event_id,sensor
[search index=ips earliest=-2h rec_type_simple="IPS EVENT" csirtm_client=cisco
NOT (`std_expected_source` OR `rules_with_play` OR `rules_disabled` OR sid=1882 OR
sid=17210)]
| transaction maxevents=1 event_id,sensor
| decrypt f=packet unhex emit('ascii_packet')
| `ips_output_by_srcip`
| `ips_event_summary`
| where rule_count>1
| sort -count
OPTIMIZE
OptimizePrime
By: Cisco CSIRT
powered by w3.css


HELP
Optimizing
x
Your query is being optimized. This may take some time depending on query timerange and tasks.
From testing, the optimized query is usually faster. All benchmarking results vary based on load on the search heads and indexers during the time of the tests.
If you think these results are incorrect, please re-run your query.
OPTIMIZE
OptimizePrime
By: Cisco CSIRT

Results

x

Syntax Prime

Errors	Warnings
No Errors	latest= was not found

Search Prime

Candidate Query:	| search index=ips earliest=-2h rec_type_simple="PACKET" OR rec_type_simple="EXTRA DATA" csirtm_client=cisco | join event_id,sensor [search index=ips earliest=-2h rec_type_simple="IPS EVENT" csirtm_client=cisco NOT (`std_expected_source` OR `rules_with_play` OR `rules_disabled` OR sid=1882 OR sid=17210)] | transaction maxevents=1 event_id,sensor | decrypt f=packet unhex emit('ascii_packet') | `ips_output_by_srcip` | `ips_event_summary` | where rule_count>1 | sort -count
Candidate Runtime (seconds):	33.169
Optimized Query:	| search (index=ips earliest=-2h (rec_type_simple="PACKET" OR rec_type_simple="EXTRA DATA") csirtm_client=cisco) | join max=1 overwrite=1 type=inner usetime=0 event_id,sensor [| search (index=ips earliest=-2h rec_type_simple="IPS EVENT" csirtm_client=cisco NOT src_description=IN_VULNERABILITY_SCANNER NOT src_description=OUT NOT src_description=OUT_QUALYS_SCANNER NOT "Malware" NOT "outbound" NOT msg="INDICATOR-COMPROMISE*" NOT msg="*sql*" NOT dest_port=80 NOT dest_port=445 NOT "libssh" NOT sid=45549 NOT msg="ET POLICY Proxy Connection detected" NOT "STOR" NOT sid=34061 NOT sid=29639 NOT sid=1325 NOT "External Hosts sending SSH key to internal Cisco hosts" NOT sid=1882 NOT sid=17210)] | transaction maxevents=1 event_id,sensor | decrypt f=packet unhex emit('ascii_packet') | fillnull value=Undefined csirtm_client | stats earliest(_time) AS first_event, latest(_time) AS last_event, values(src_description) AS src_description, values(msg) AS rule_description, values(sid) as snortid, distinct_count(msg) AS rule_count, values(dest_ip) AS dest_ip, values(dest_description) as dest_description, dc(dest_ip) as dest_count, values(src_port) as src_port, values(dest_port) AS dest_port, values(ascii_packet) AS payload, values(data) AS xdata, values(sensor) AS sensors, values(csirtm_client) as client, count as event_count by src_ip | convert timeformat="%m/%d/%Y %H:%M:%S %Z" ctime(first_event), ctime(last_event) | sort -event_count | eval src_port=if((mvcount(src_port) > 10),mvappend(mvindex(src_port,0,10),"[...]"),src_port) | eval src_ip=if((mvcount(src_ip) > 10),mvappend(mvindex(src_ip,0,10),"[...]"),src_ip) | eval dest_ip=if((mvcount(dest_ip) > 10),mvappend(mvindex(dest_ip,0,10),"[...]"),dest_ip) | eval dest_port=if((mvcount(dest_port) > 10),mvappend(mvindex(dest_port,0,10),"[...]"),dest_port) | eval payload=if((mvcount(payload) > 10),mvappend(mvindex(payload,0,10),"[...]"),payload) | eval xdata=if((mvcount(xdata) > 10),mvappend(mvindex(xdata,0,10),"[...]"),xdata) | where (rule_count > 1) | sort -count | noop search_optimization=false

Optimized Query Runtime (seconds):	31.353
Fastest Query:	Optimized Query
Difference (seconds):	1.816
Percent Faster:	5.475%
Splunk Search Head Version:	7.1.4

Stats for Nerds +

	Candidate	Optimized
command.decrypt	6.121	5.593
command.eval	0.001	0.001
command.join	1.408	1.495
command.search	89.499	85.306
command.search.calcfields	0.476	0.474
command.search.expand_search	1.456	1.415
command.search.fieldalias	0.696	0.68
command.search.filter	0.161	0.155
command.search.index	0.196	0.196
command.search.kv	3.872	3.658
command.search.lookups	2.169	2.08
command.search.rawdata	1.931	1.926
command.search.tags	0.479	0.456
command.search.typer	79.374	75.545
command.simpleresultcombiner	9.007	9.2
command.stats	0.027	0.031
command.stats.execute_input	0.026	0.029
command.stats.execute_output	0.001	0.002
command.transaction	0.1	0.106
dispatch.createdSearchResultInfrastructure	0.033	0.035
dispatch.evaluate.decrypt	0.001	N/A
dispatch.evaluate.join	65.044	68.712
dispatch.evaluate.search	2.24	2.171
dispatch.evaluate.transaction	0.001	0.001
dispatch.fetch.rcp.phase_0	0.062	0.048
dispatch.finalWriteToDisk	0.001	0.001
dispatch.parserThread	5.248	4.937
dispatch.stream.remote	89.501	85.308

Data Prime

Index:	ips
Size (MB):	814,015
Events Within Timeframe:	2,574,690
Event Count Warnings:	Searching greater than 1M events.
Data Size Warnings:	None

Total Events Within Timeframe:	2,574,690
Total Size of Indices (MB):	814,015
Total Event Count Warnings	None
Total Data Size Warnings	None

Initial Results – Time Saved

Seconds Per Day	Minutes Per Day	Hours Per Day
4289.01	71.48	1.19

Time saved after a week…

Seconds	Minutes	Hours
30,023.07	500.38	8.34

Time saved after a month…

Seconds	Minutes	Hours	Days
120,092.28	2,001.54	33.36	1.39

Time saved after a year…

Seconds	Minutes	Hours	Days
1,566,560.90	26,109.35	435.16	18.13

Additional stats from testing...

TIME SAVED	Minimum per year	Average per year	Maximum per year
Hours	292.06	408.96	555.66
Days	12.17	17.04	23.15

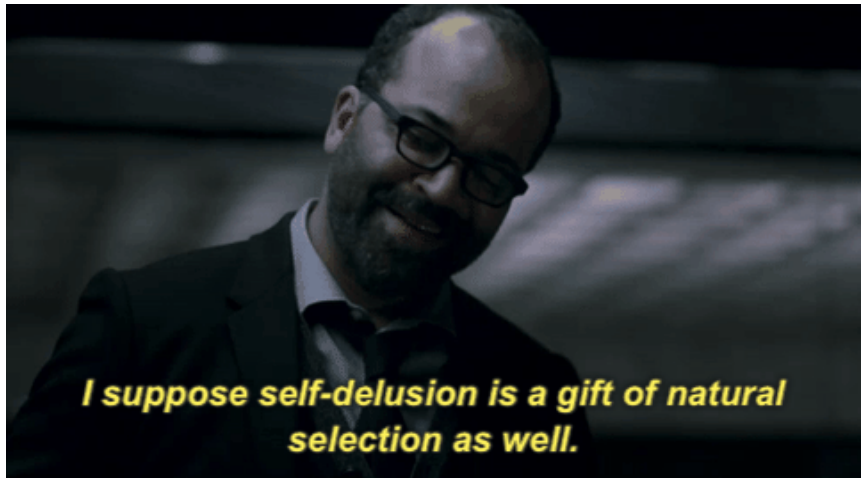
- Load varies during testing
- Intersection of data being searched in prod while trying to optimize
- Unoptimized query can have a significant runtime variance
  - Running an already optimized query can allow us to skip the built-in Splunk optimization process

Playbook Integration

- Ensures analyst always runs optimized play
- Automatic re-optimizations based on:
  - Splunk version upgrade
  - Play update
  - OptimizePrime code changes
- Can manually force a full playbook optimization
- Analyst can compare original query with optimized query side-by-side

Future Plans

- Add analysis for subsearches
- Add analysis for lookup tables
- Create benchmark criteria
  - ex. using transform commands versus specifying fields
  - Order of operations
- Build into play creation process
- Add streamed logging to web interface
- Open-source the code



What if I don’t use Splunk?

- Leverage built-in optimization engine
- Check query syntax for best practices or recommendations
- Set limits on how much data can be searched based off of your environment’s performance
- Compare similar statistic or transformative statements for performance gains
- Record your results
  - Justify your effort and show value to the mission

Questions