The story of a little Splunk engine that could

Lessons learned from resolving the issues caused by 8 years of organic growth of a Splunk landscape

Pieter Bovy | Solution Architect
Lex Crielaars | CTO

4th of Oct, 2018 | 12:15:00 PM - 01:00:00 PM

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.

Pieter Bovy

Solution Architect, ASML

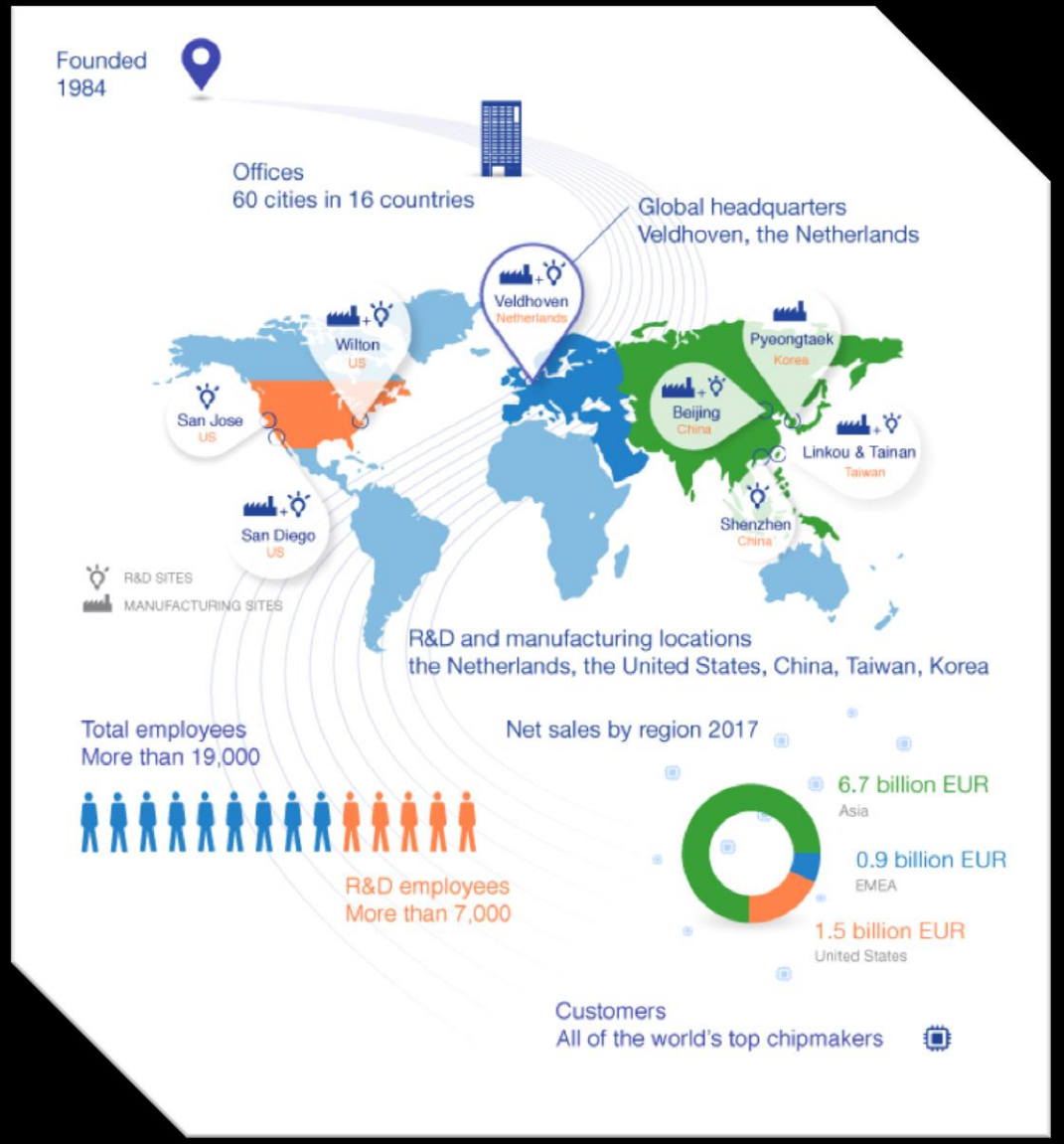
Lex Crielaars

CTO, SMT Netherlands

splunk> .conf18

ASML

ASML is the world's leading provider of lithography systems for the semiconductor industry, manufacturing complex machines that are critical to the production of integrated circuits or chips



Data Driven Decision Making

We provide autonomous advice about the possibilities, application and strategic use of data. We support your strategic decisions and lay the groundwork for the right choices.

OPERATIONAL — STRATEGIC

- Use cases
- Software licenses
- Implementation

- Intensification
- Broadening
- Integration

- Build on results
- Proactive advice
- Platform maintenance

- Partner
- Trusted advisor

How did we get here?

“The Journey, Not the Destination Matters”

T.S. Elliot

The Journey






Customer Satisfaction

Size & Complexity

2010

2014

2018

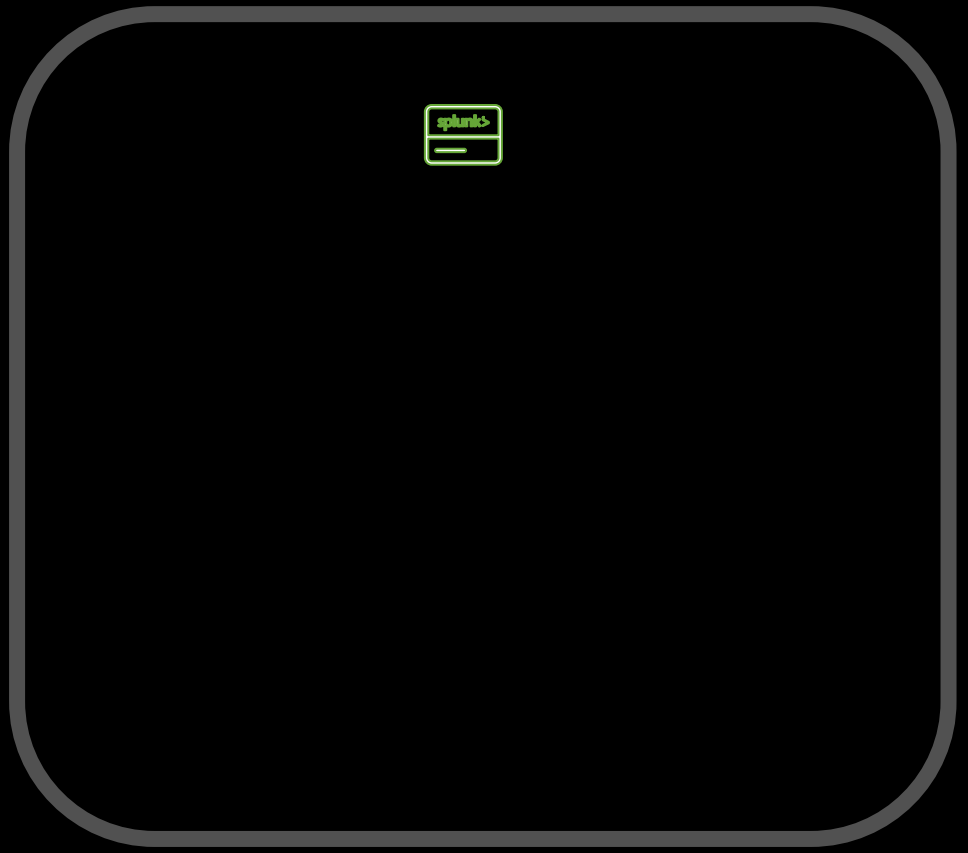
Conception

2010

IT has to begin somewhere

- a Single Splunk server

- License: 20GB/Day
- Splunk 4.x
- Splunk Core



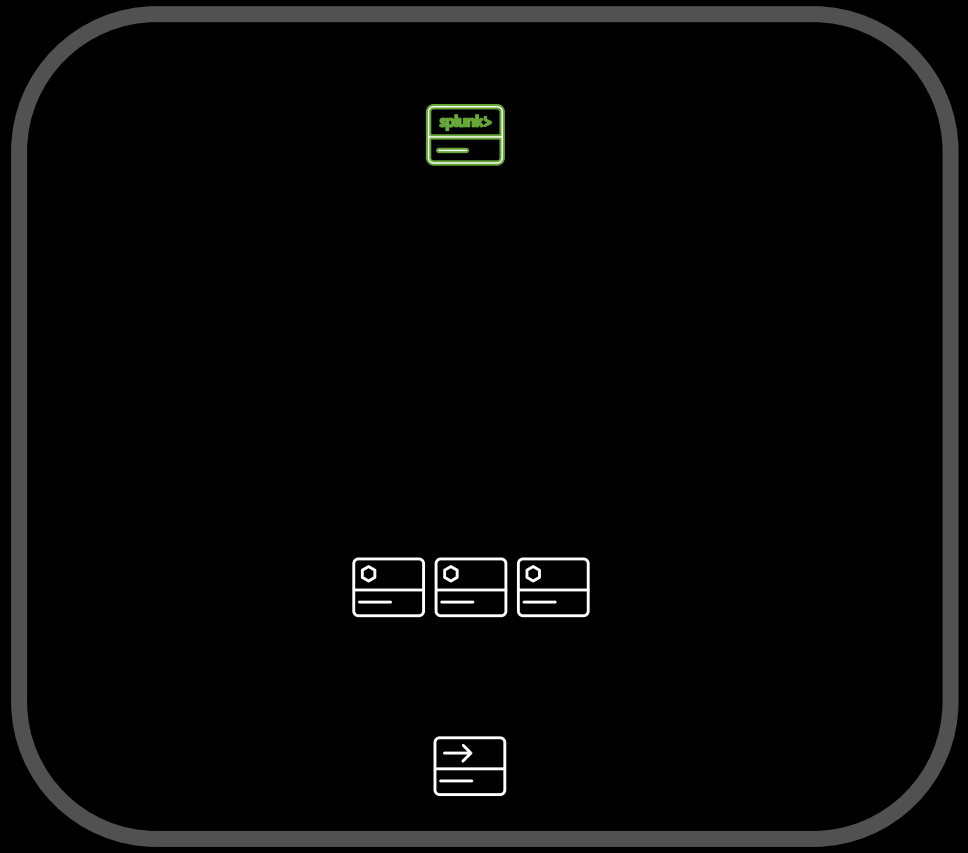
Security use case

2014

First use case based onboarding

- 1 Search head
- 3 Indexers
- 1 Heavy Forwarder

- License: 250GB/Day
- Splunk 5.x
- Splunk Core



Security Incident 2015

VELDHOVEN, Netherlands, 1 March 2015 - ASML Holding N.V. (ASML) recently discovered unauthorized access to a limited portion of its IT systems. ASML took immediate steps to contain the breach and is conducting an ongoing investigation. The time between the break-in and the discovery by ASML IT staff was short. At this time it appears that only a limited amount of data has been accessed. ASML has not found any evidence that valuable files, either from ASML or our customers and suppliers, have been compromised. We cannot be certain about the identity of the hackers

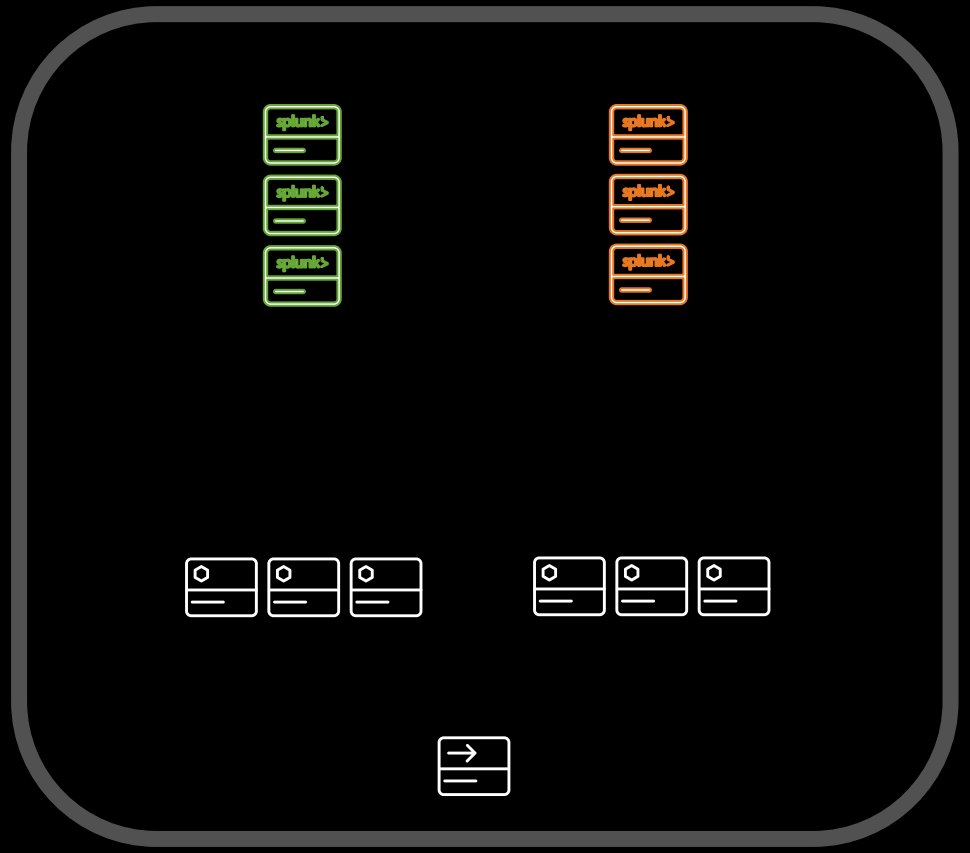
Enterprise Security & disaster recovery

2016

Use case based SIEM rollout

- 2 SHC’s
- 2-Site Indexing cluster
- 1 Heavy Forwarder

- License: 1,2TB/Day
- Splunk 6.x
- Splunk Core + ES



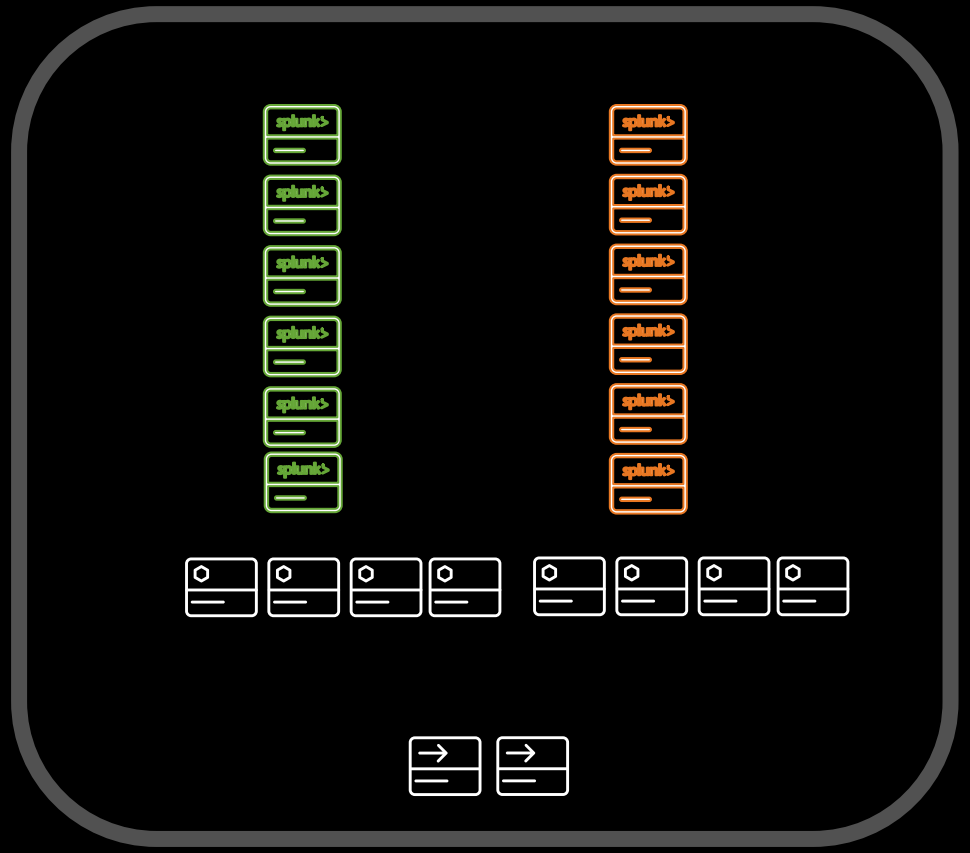
Technical debt catchup

2016

Get back in control

- 2 SHC’s
- 2-Site Indexing cluster
- 2 Heavy Forwarders

- License: 1,2TB/Day
- Splunk 6.x
- Splunk Core + ES



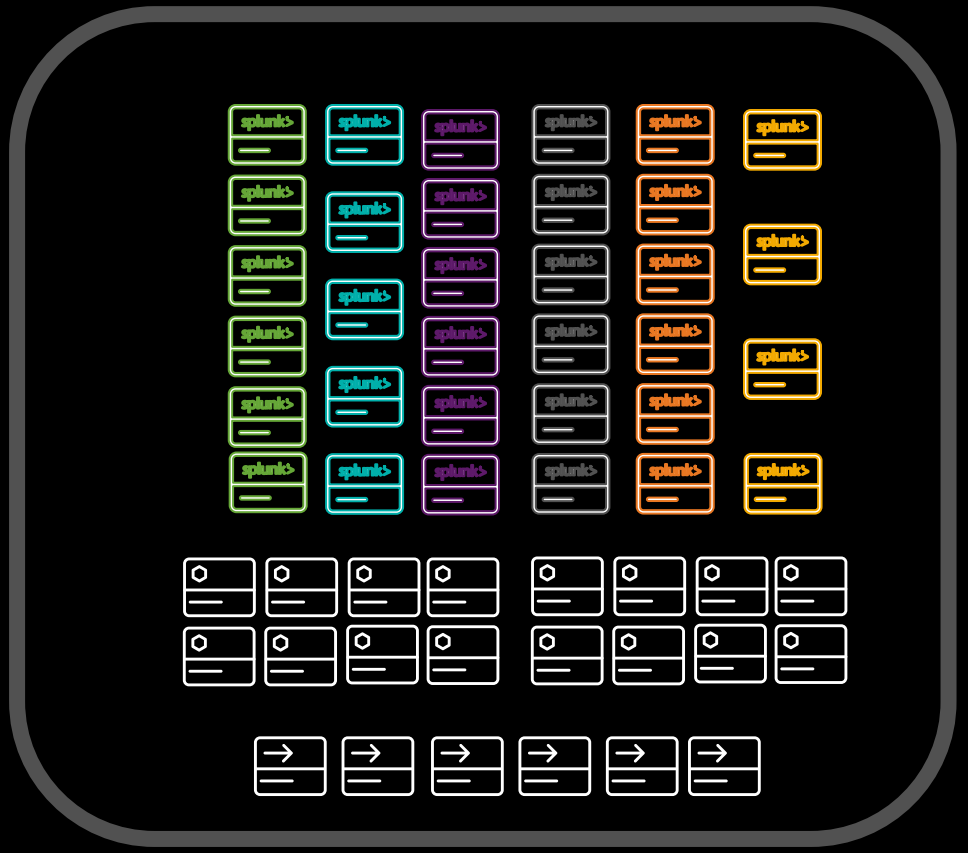
Current Situation

2018

Keep on growing

- 43 (PRD) Search Heads
- 6 SHC’s
- 2-Site 16-Node Indexing cluster
- 6 Heavy Forwarders

- License: 1,3TB/Day
- Splunk 7.1.2
- Splunk Core + ES + ITSI



Splunk @ASML IT

2018

The Current State

600+ Users

100+ Use Cases, including ITSI and ES

~200 servers

D-A-P environment

1.3 TB / day Data

10k+ sources

DevOps Style Support

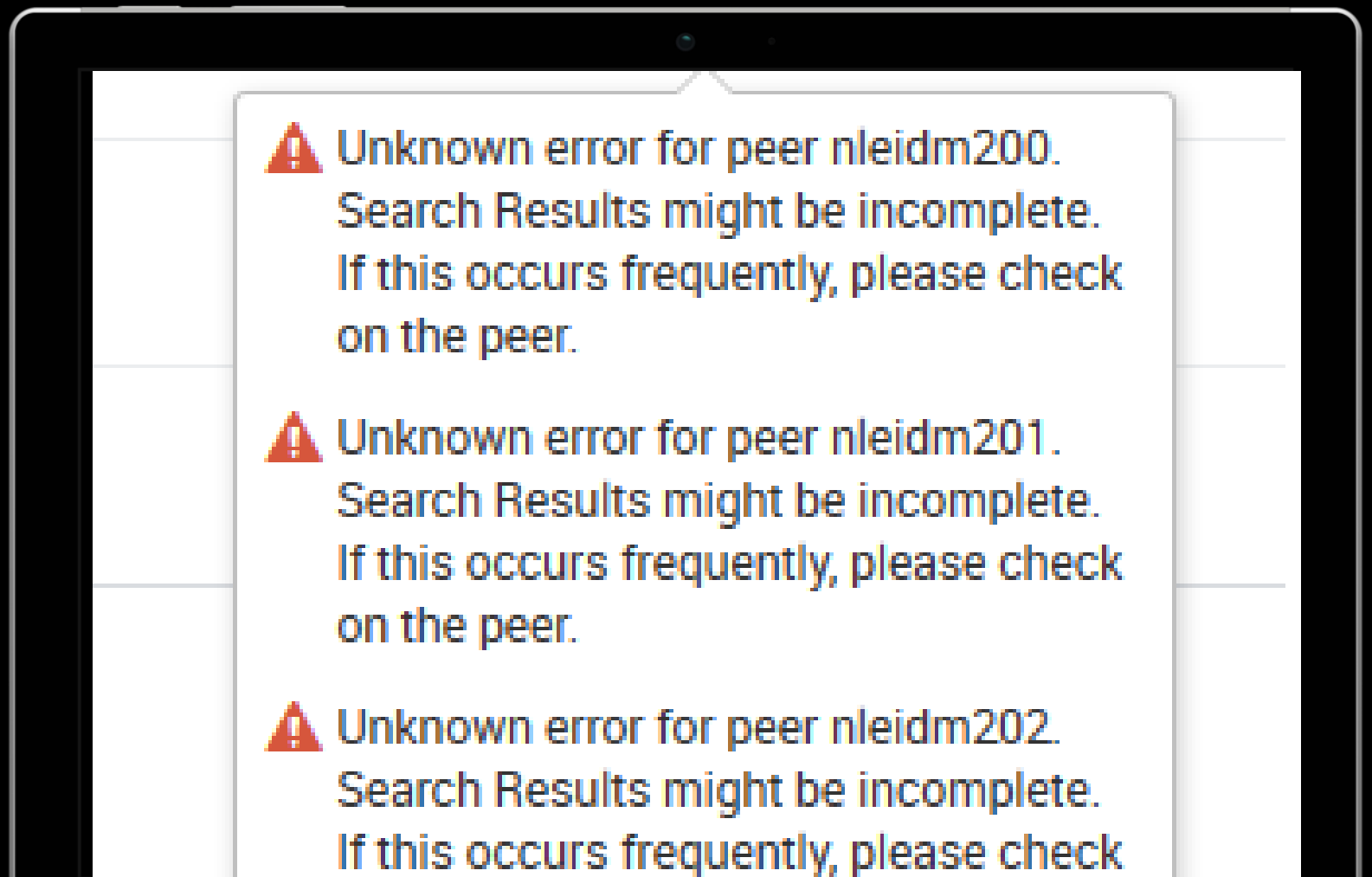
The Unknown Error

"There are no mistakes, save one: the failure to learn from a mistake"

Robert Fripp

The Unknown Error

Our Arch Enemy



- Visible to all users
- Catch all for Spunk saying ‘I don’t know’
- May be many small problems combined

The investigation

Project Heracles: Fighting the Hydra

What did we do?

- Initial investigation & trouble shooting
- Raised Splunk support cases
- Raised Red Hat support cases
- Pressure Cooker with:
  - OS / Hardware management team
  - Splunk application support team
  - Network Team
  - Splunk Engineering

Step 1: Things to check by yourself

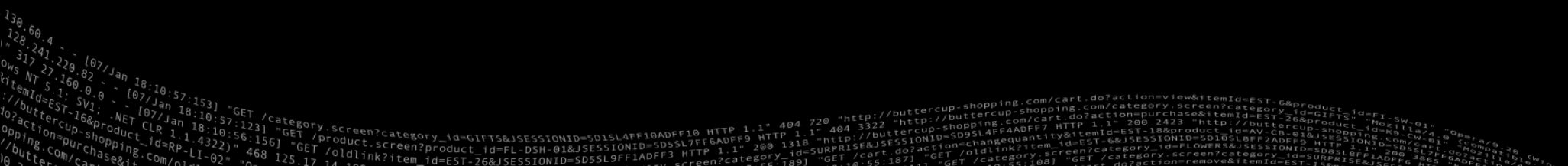
What did we check ourselves?

WHAT

- Are we running the latest Splunk version?
- Is NTP setup correctly?
- Is THP disabled?
- File limits
- Kernel Tunables
- Inter Splunk Communication
- Saved Search Scheduler
- Knowledge Bundle Replication

HOW

- Are we running the latest Splunk version?
- ntpq –p is all you need.
- Check it. The (D)MC might be lying to you!
- Linux is a fickle mistress. Check it yourself.
- Have you tuned your kernel today?
- TCP Dumps / Wireshark
- Congestion is bad.
- Blacklisting is your friend.



splunk> .conf18

Step 1: Things to check by yourself

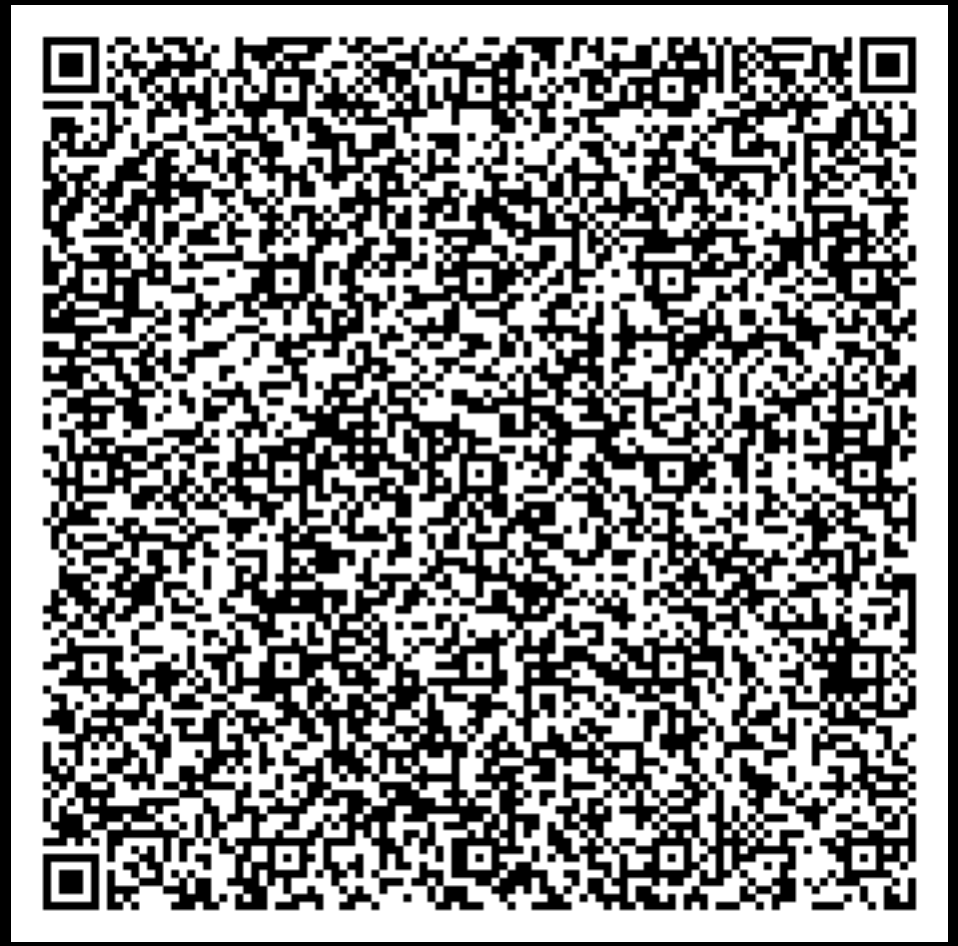
THP? Where we're going, we don't need THP.

WHAT

THP (Transparent Huge Pages) is a method of combining five hundred 4KB memory pages into one 2MB memory page.

- Can be enabled and disabled on the fly
- Trade-off between memory and CPU usage
- Performance loss in short-lived processes
- Around 30% performance hit for Splunk

QR Code is a RHEL6-compliant SysVinit script to enable, disable and check the status of THP.



Step 1: Things to check by yourself

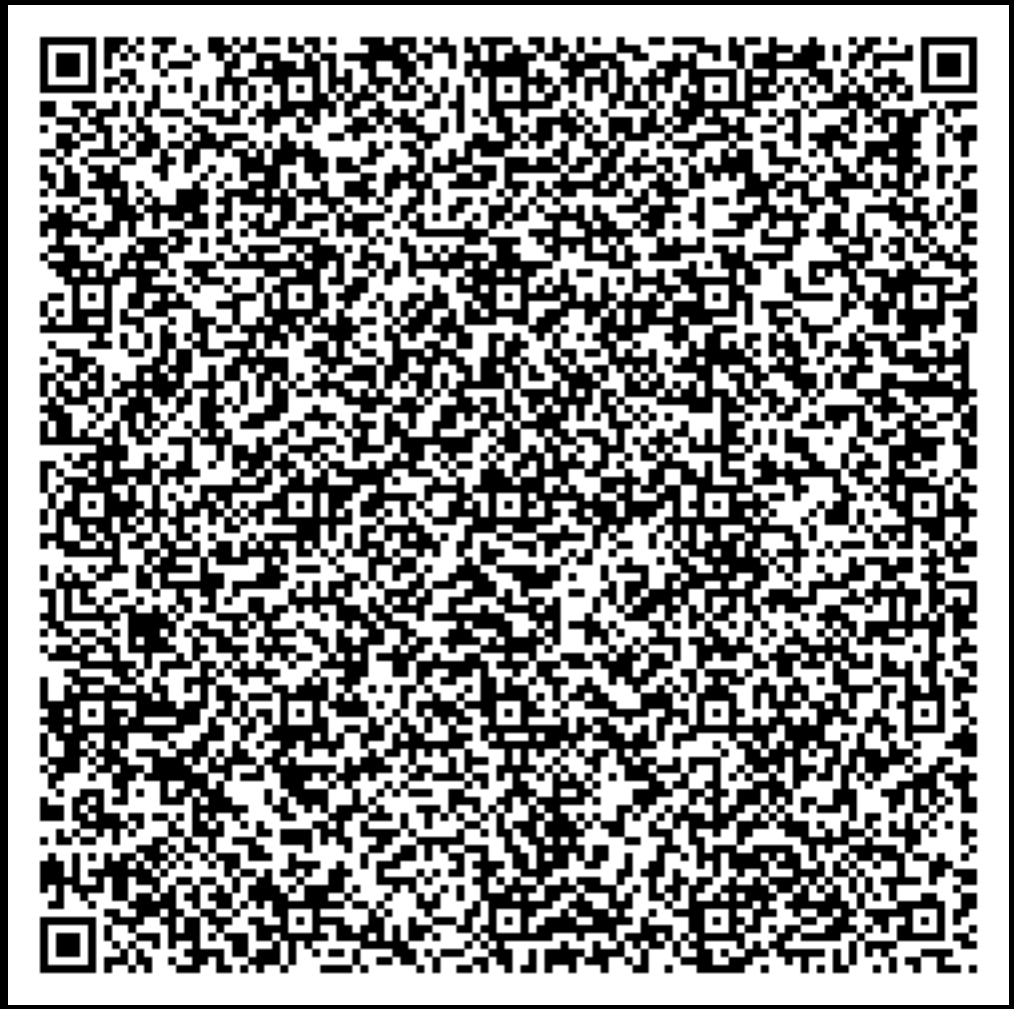
Raise the limits on Splunk

WHAT

Linux imposes strict limits on processes about how many files they can open and the number of simultaneous processes a user can have.

- Default limits can cause performance issues
- Indexers are especially affected
- best practices don’t always work
- Difference in system bootup and Splunk restart

QR Code is a modified Splunk SysVinit script that sets appropriate limits just before (re)starting Splunk.



Step 1: Things to check by yourself

Tune all the kernels!

WHAT

- net.core.somaxconn = 2048

Determines the maximum of backlog connections an application can request.

- net.ipv4.tcp_max_syn_backlog = 4096

Dictates the maximum amount of outstanding syn requests.

- net.ipv4.ip_local_port_range = 1024 65535

Expands the ephemeral port range on Linux.

- net.ipv4.tcp_fin_timeout = 20

This specifies how many seconds to wait for a final FIN packet before the socket is forcibly closed.

- net.ipv4.tcp_tw_reuse = 1

Allows for the re-use of connections in TIME_WAIT status. Relatively safe to use.

- net.ipv4.tcp_tw_recycle = 1

Enables fast recycling of TIME_WAIT sockets but causes problems when your clients are behind a NAT (such a load balancers).

- These go in /etc/sysctl.conf to make them permanent

splunk> .conf18

Step 1: Things to check by yourself

Sniff packets. All the cool ninja's are doing it.

WHAT

tcpdump –i *interface* –w *file* –s *snaplen* [src|dst] [net] … [and|or] [port] …

- tcpdump –i eth0 –w *splunk-dump.pcap* –s 0 'dst net *192.168.0.0/24* or dst 192.168.5.10' and port *8089*
- tcpdump -I eth0 –w splunk-dump.pcap 'tcp[13] & 4!=0'
- Booktip! Practical Packet Analysis (ISBN-13: 978-1-59327-802-1)

Load the PCAP file in Wireshark and keep an eye out for red, black and blue items

- TCP resets
- TCP Duplicate ACK's
- TCP ZeroWindow

No.	Time	Source	Destination	Protocol	Length	Info
92	1.431993	146.106.61.26	146.106.69.30	TCP	66	35289 → 8089 [ACK] Seq=1 Ack=326 Win=501 Len=0 TSval=286987800 TSecr=1256509331
93	1.432215	146.106.69.30	146.106.61.26	TLSv1.2	343	Application Data
94	1.432218	146.106.61.26	146.106.69.30	TCP	66	35289 → 8089 [ACK] Seq=1 Ack=603 Win=501 Len=0 TSval=286987800 TSecr=1256509331
95	1.544137	146.106.17.168	146.106.61.26	TCP	66	[TCP Dup ACK 14#2] 8089 → 40293 [ACK] Seq=1 Ack=1 Win=67 Len=0 TSval=2029962897 TSecr=2869869
96	1.544152	146.106.61.26	146.106.17.168	TCP	66	[TCP ZeroWindow] [TCP ACKed unseen segment] 40293 → 8089 [ACK] Seq=1 Ack=2 Win=0 Len=0 TSval=
97	1.734756	146.106.61.26	146.106.69.30	TCP	66	34494 → 8089 [FIN, ACK] Seq=1 Ack=1 Win=352 Len=0 TSval=286988103 TSecr=1256297843
98	1.735339	146.106.69.30	146.106.61.26	TLSv1.2	1514	Application Data, Application Data
99	1.735362	146.106.61.26	146.106.69.30	TCP	54	34494 → 8089 [RST] Seq=2 Win=0 Len=0
100	1.735366	146.106.69.30	146.106.61.26	TLSv1.2	1514	Application Data, Application Data
101	1.735368	146.106.61.26	146.106.69.30	TCP	54	34494 → 8089 [RST] Seq=2 Win=0 Len=0
102	1.930500	146.106.17.251	146.106.61.26	TLSv1.2	391	Application Data

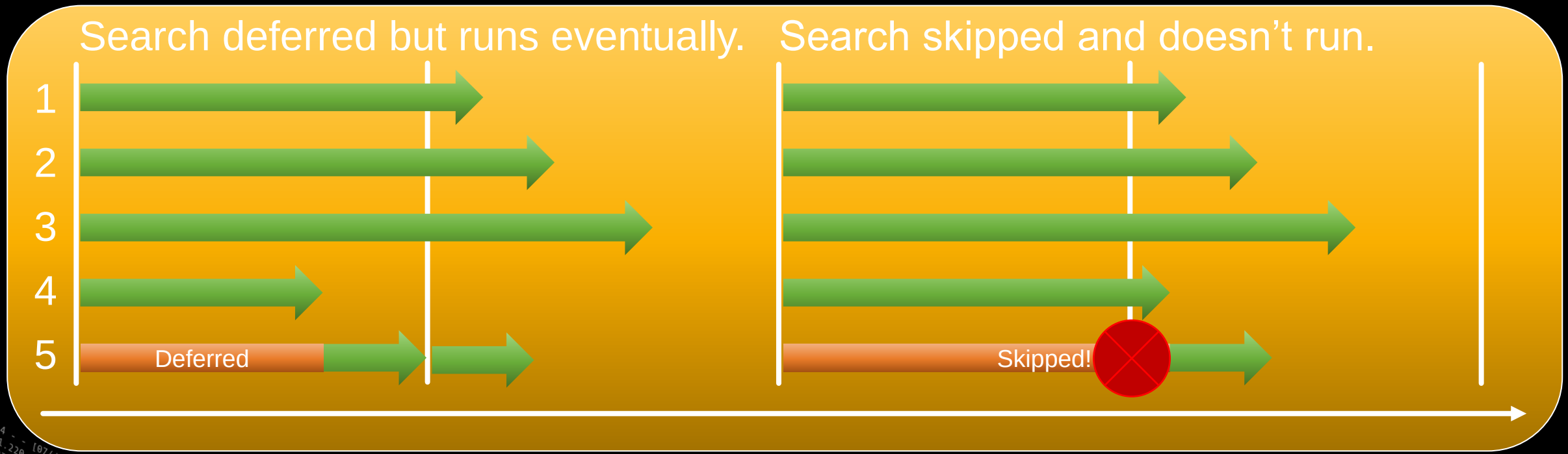
Step 1: Things to check by yourself

Don’t skip searches. Don’t skip leg day.

WHAT

Splunk runs as many concurrent searches as it can. The limit is based on the number of CPU cores.

- limits.conf : max_hist_searches = max_searches_per_cpu (1) x number_of_cpus + base_max_searches (6)
- limits.conf : max_rt_searches = max_rt_search_multiplier (1) x max_hist_searches



Step 1: Things to check by yourself

Bundle your knowledge but only what is needed

WHAT

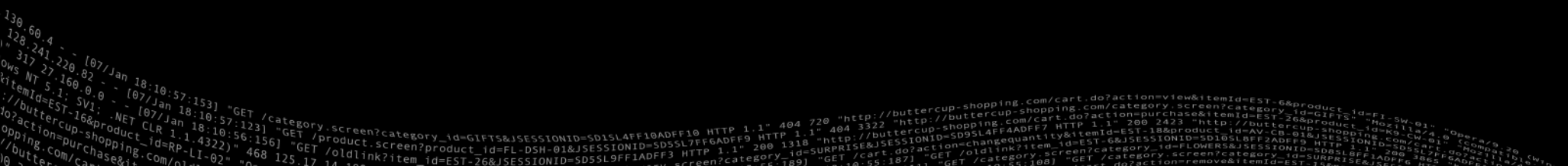
Search heads push out knowledge objects (event types, lookups, saved searches, etc.) to the indexers

- The bundle is compressed
- Nearly every knowledge object is bundled by default
- Large knowledge bundles hurt your performance
- Splunk tries to push delta bundles instead of full bundles
- In Search Head Clusters only the captain pushes out bundles
- You don’t always need every knowledge object (such as lookups) on your indexers

distsearch.conf:

[replicationBlacklist]

NO_CSV_APPNAME = apps/<appname>/lookups/*.csv

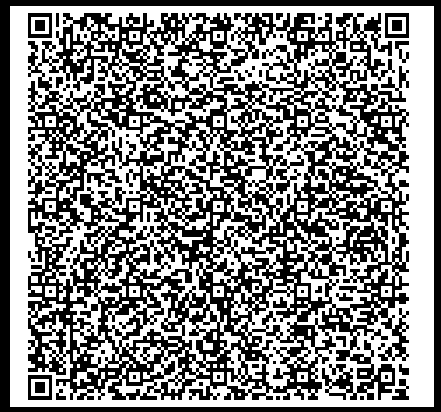


Step 1: Things to check by yourself

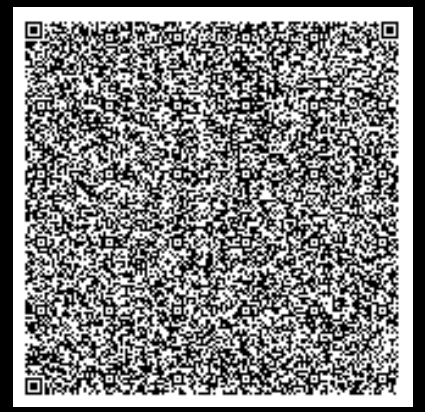
The power of the Dispatch Folder

- Will show Unknown Error with Indexer and Peer involved
- Can help predict UE's for timing of PStacks, Diags, Etcetera
- Onboard it into _internal index (no license impact)
- Huge volume (!!!)

QR Code is SPL on dispatch data for UE



QR Code is SPL on dispatch and internal data for searches during UE



QR Code is SPL on dispatch to predict an UE

Step 1: Things to check by yourself

Any other business we found

- Pruning / Collapsed Packages
- netstat -s | egrep 'prune|collapse’ Showed increasing numbers in pruned and collapsed packets.
- NIC buffering
- Increase NIC buffers according to https://access.redhat.com/solutions/369563
- Jumbo Frame Settings
- Jumbo frames were set on the indexers but is only effective inside the same subnet/VLAN.

Step 2: Time to call in the Cavalry

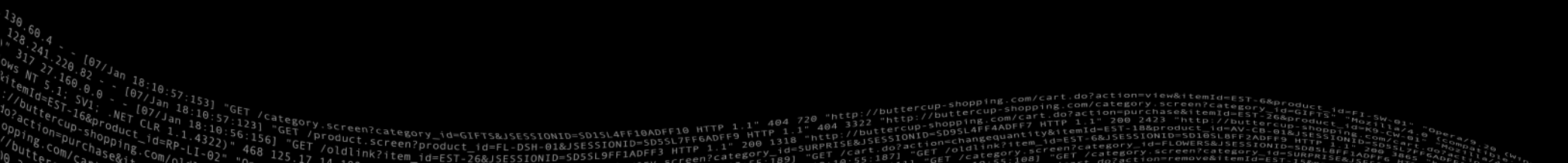
What did we check?

WHAT WE CHECKED

- Help from:
  - Splunk Support & Engineering
  - Red Hat Support & TAM
- Diags, diags and more diags
- Gigabytes of packet captures
- Pstacks of Splunk processes
- GDB'ing splunkd
- Splunk Config files
- OS config

WHAT WE FOUND

- A way to force the Unknown Error
- Pstacks indicated CPU Starvation of splunkd
- Disk schedulers
- Sourcetype definitions re-write



splunk> .conf18

Step 2: Time to call in the Cavalry

Force the Unknown Error

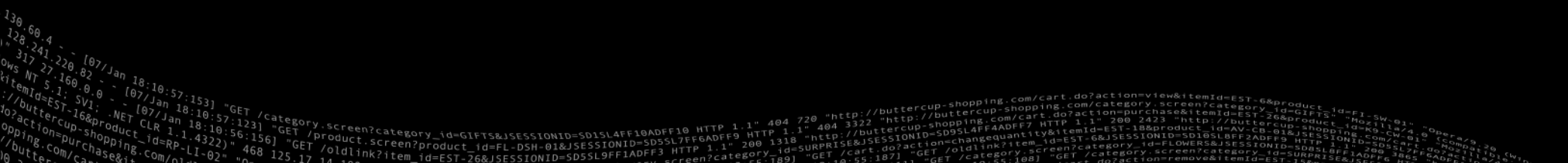
WHAT

We discovered that the Unknown Error was more likely to occur during high CPU load.

- It would occur more often at the 0, 15, 30 and 45 minute mark of each hour
- Errors would go up during the day and down towards the end of the day
- Taking down “dashboard TV’s” lowered the amount of Unknown Errors

Forcing the Unknown Error to appear was done by doing stupid searches

- Index=*
- All time
- 5 to 10 times



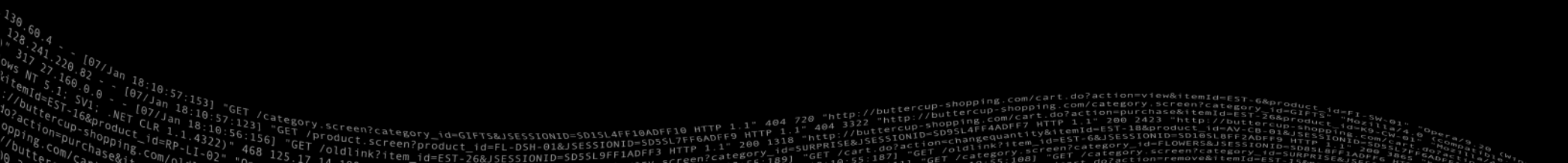
Step 2: Time to call in the Cavalry

Pstack investigation

WHAT

- Pstack attaches to the active process named by the pid on the command line, and prints out an execution stack trace.
- By running multiple Pstacks we could investigate the splunkd process
- Analysis told us that splunkd was occasionally being CPU-starved for more 10s
- Anything with a timeout of 10 seconds or less would die at this point
- Lowering the CPU load helped resolve this problem
- Booktip! Art of debugging (ISBN-13: 978-1-59327-174-9)

*i=0; while [$i -lt 100] ; do date > /tmp/pstack$i.out; pstack **$splunkd_pid** >> /tmp/pstack$i.out; let "i+=1"; sleep 1; done*



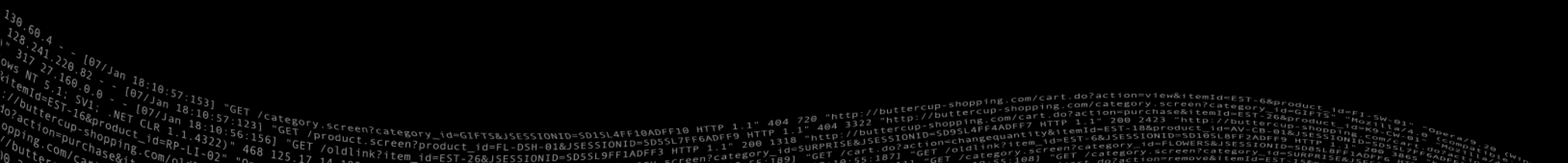
Step 2: Time to call in the Cavalry

Who schedules the schedulers?

WHAT

The I/O scheduler determines in what order I/O operations are sent to the storage

- Multiple schedulers: Anticipatory, Deadline, CFQ and Noop
- Applies to block devices only (SSD's, HDD's and iSCSI LUN's)
- Does not apply to network mounts such as NFS
- RHEL6: CFQ scheduler is the default
- RHEL7: Deadline scheduler is the default except for SATA disks



Step 2: Time to call in the Cavalry

Who schedules the schedulers?

WHAT

Splunk is a *write once/read many* application

- the deadline scheduler will give better performance
  - cat /sys/block/*xxx*/queue/scheduler → noop anticipatory [deadline] cfq
  - echo 'deadline' > /sys/block/*xxx*/queue/scheduler
- Must be set after each reboot or add *elevator=deadline* to the kernel parameters in grub

Hypervisors have their own IO scheduler

- Makes the internal IO scheduler of the VM obsolete
- Set the scheduler to *noop* (NO OPeration)

Step 2: Time to call in the Cavalry

Minimum sourcetype definitions

WHAT

- Splunk's flexibility to perform automatic sourcetype recognition, timestamp recognition, etc. come at the expense of performance
- To maximize CPU efficiency on indexers, always configure:
  - LINE_BREAKER
  - SHOULD_LINEMERGE
  - MAX_TIMESTAMP_LOOKAHEAD
  - TIME_PREFIX
  - TIME_FORMAT

How to handle an investigation into Complex Splunk Issues

- Start with the easy stuff
- Don’t be too proud: Call in the Cavalry
- Pressure Cooker with SME’s at the same table
- The Power of the Dispatch folder

Regaining Control

“Control your own destiny or someone else will”

Jack Welch

Focus on what?

Veni Vidi Vici

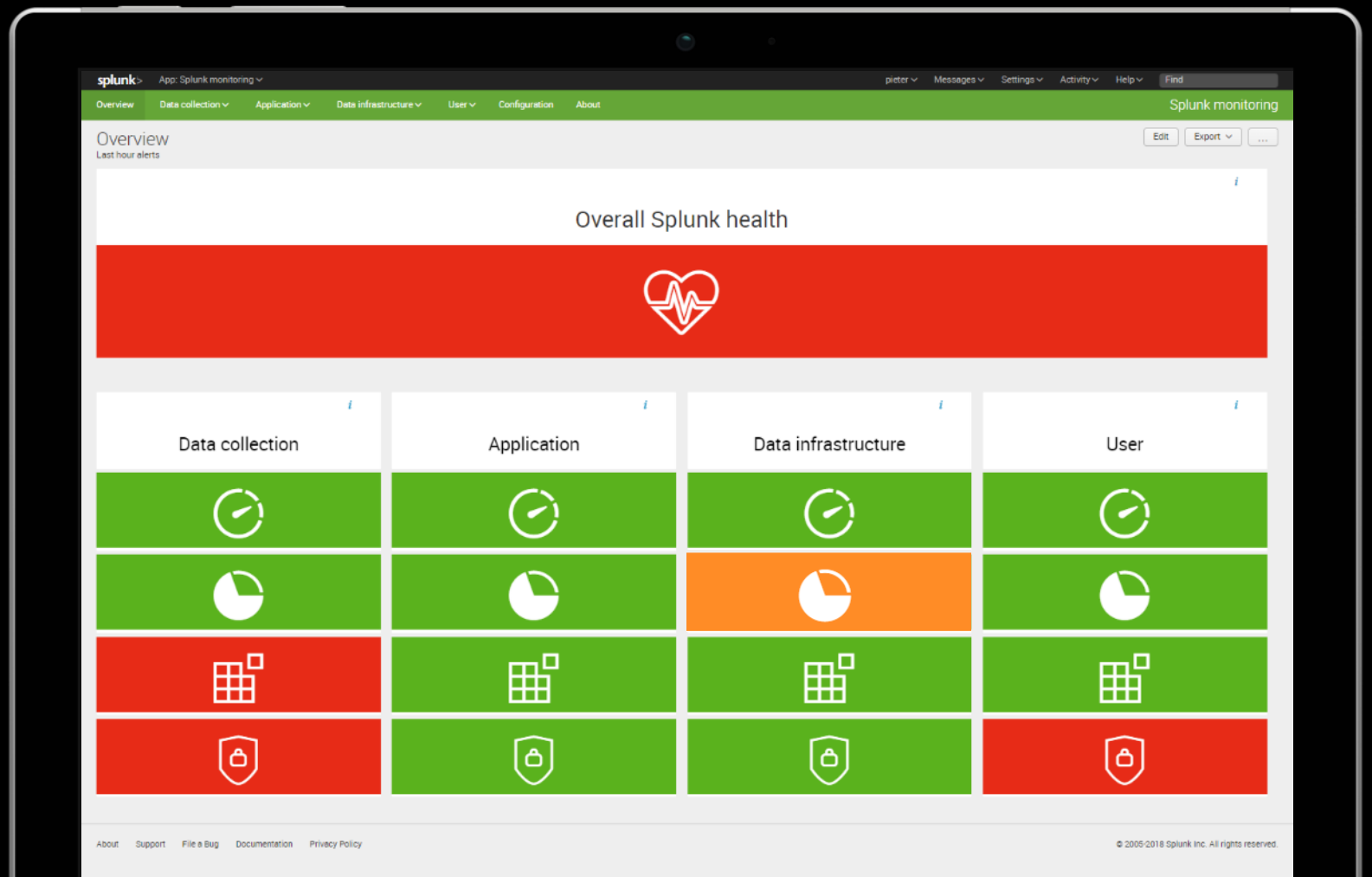
Monitor

Balance

Onboarding

REST Based Monitoring

REST assured all is well…

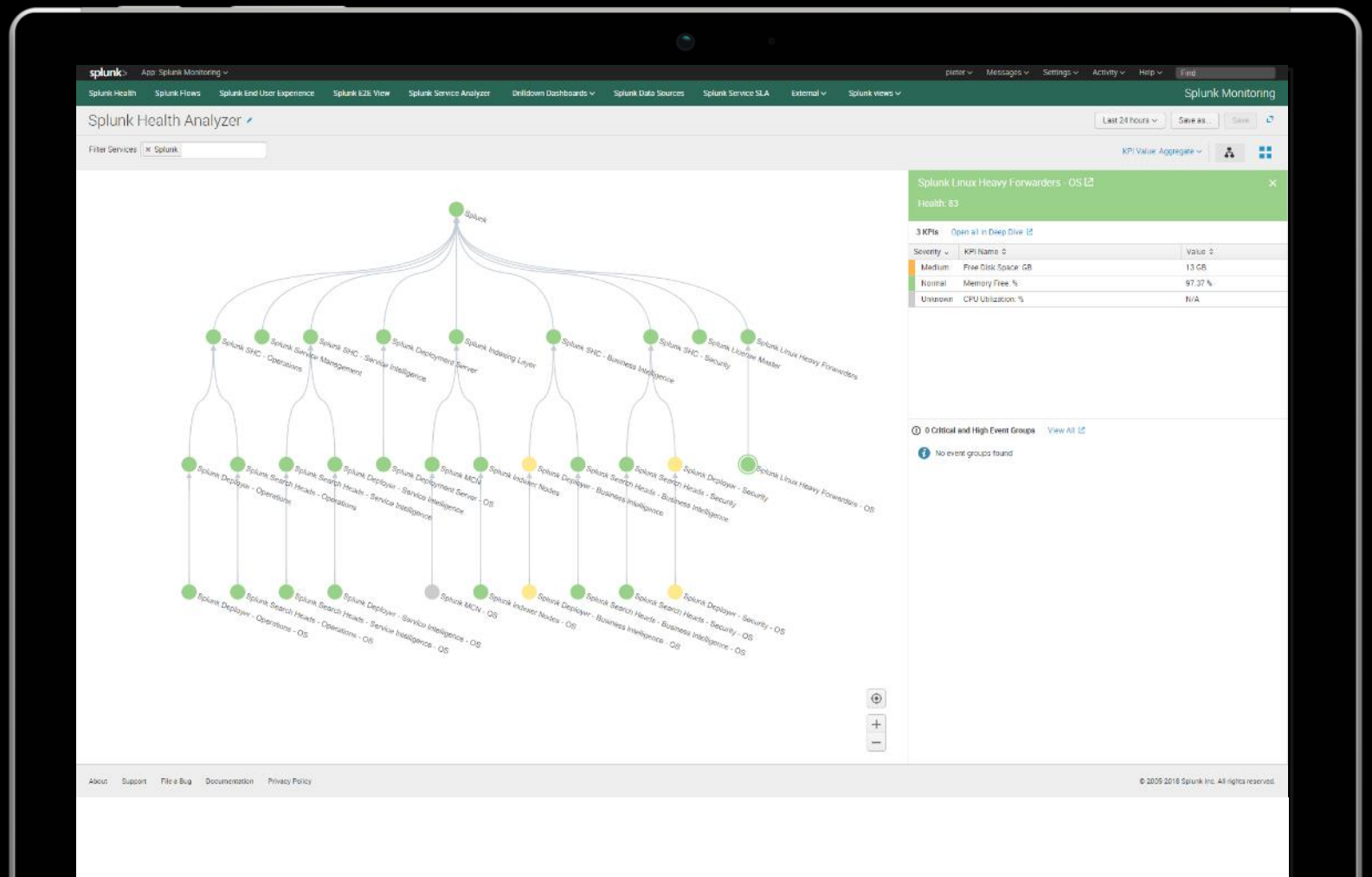


- Dedicated Search Head
- Dedicated Index
- Alerting through VictorOps
- External monitoring on availability of instance
- Contact analytics@itility.nl for information

splunk> .conf18

ITSI Based Splunk Monitoring

Quis custodiet ipsos custodies?

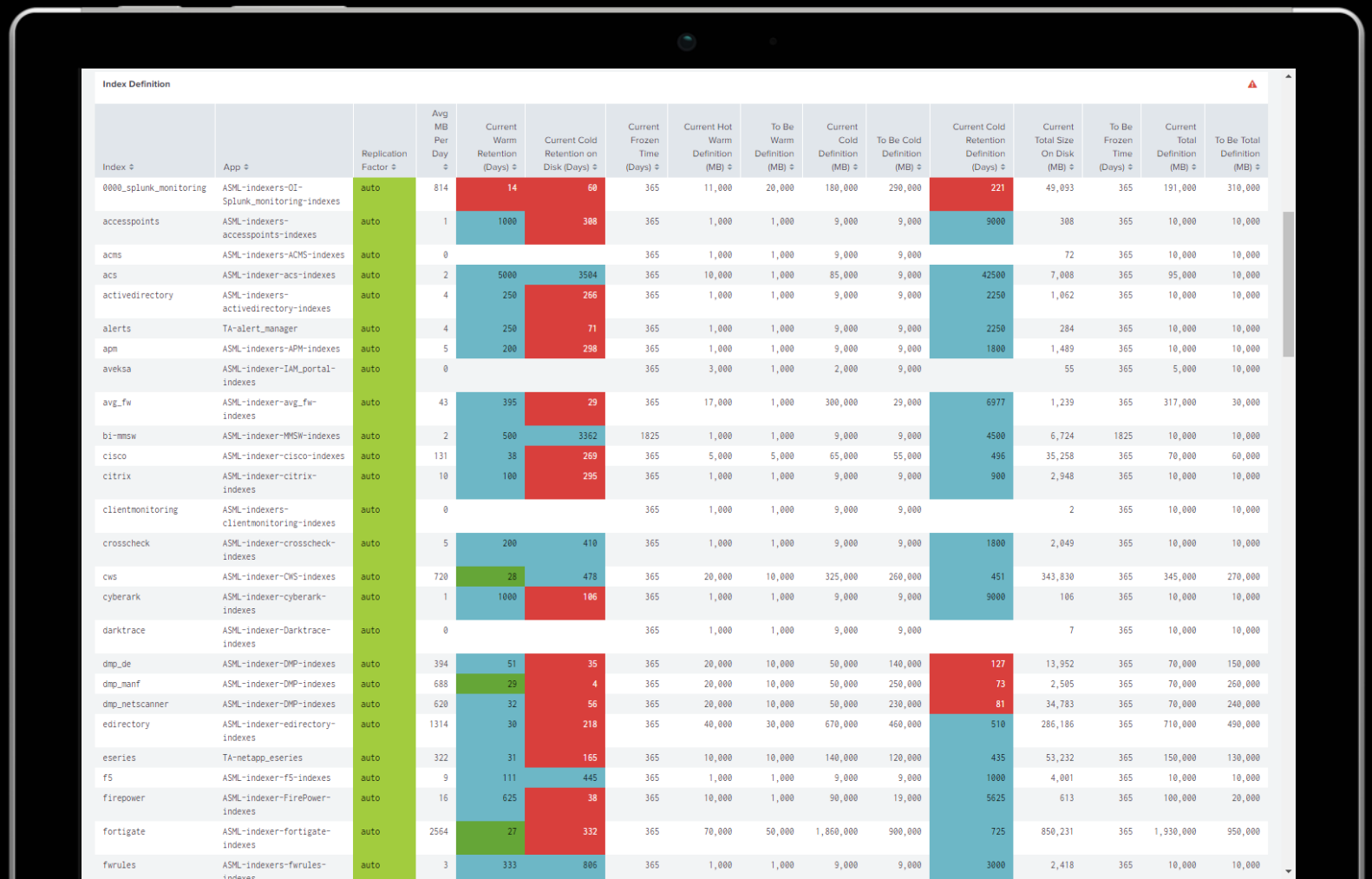


- Based on the REST sources, _internal data and OS data
- 100+ KPI's based on lessons Learned
- Service model based on CMDB, so it's dynamic

splunk> .conf18

Index retention reporting

How to monitor index definitions without volumes



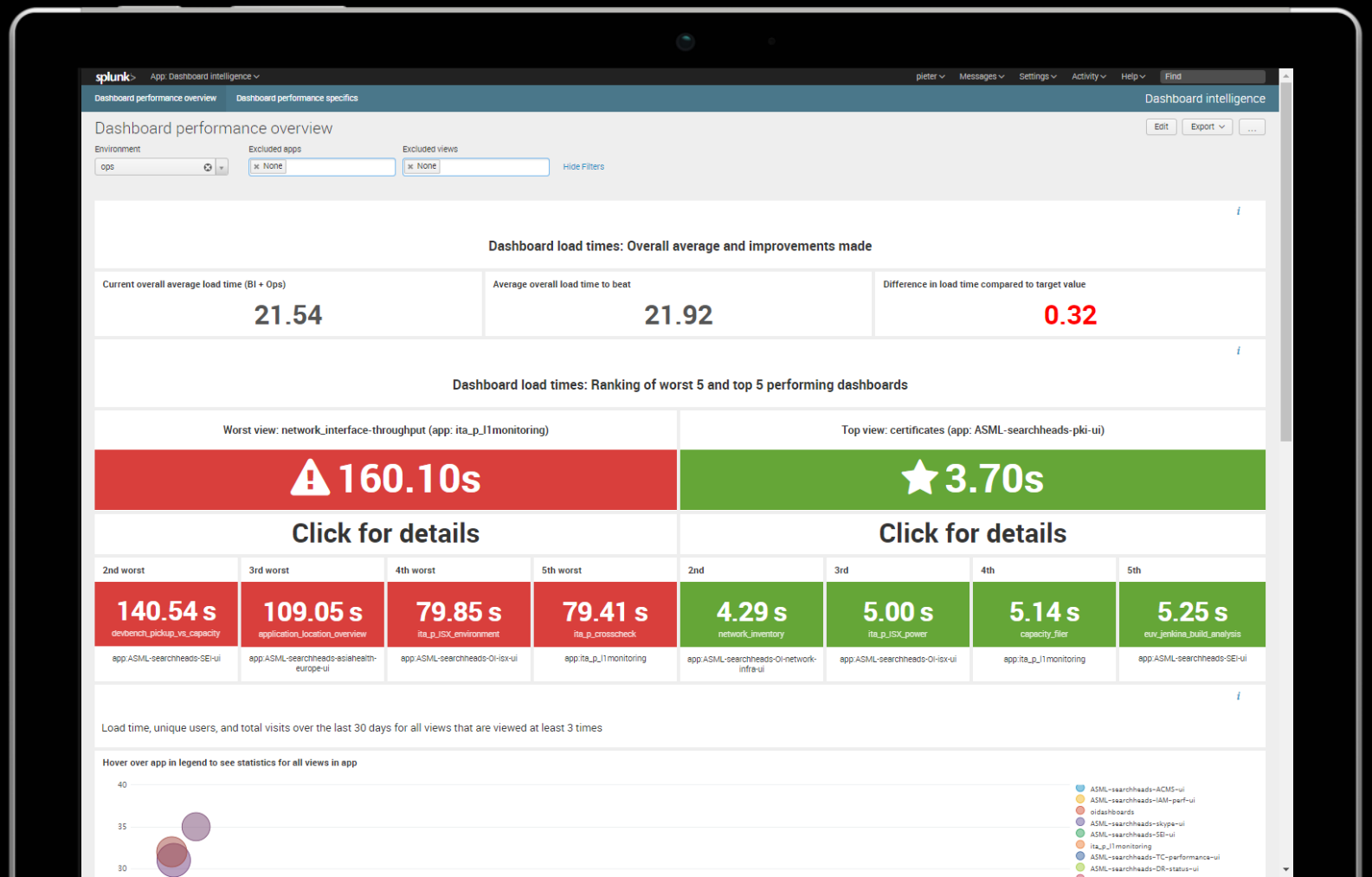
Index Definition

Index	App	Replication Factor	Avg MB Per Day	Current Warm Retention (Days)	Current Cold Retention on Disk (Days)	Current Frozen Time (Days)	Current Hot Warm Definition (MB)	To Be Warm Definition (MB)	Current Cold Definition (MB)	To Be Cold Definition (MB)	Current Cold Retention Definition (Days)	Current Total Size On Disk (MB)	To Be Frozen Time (Days)	Current Total Definition (MB)	To Be Total Definition (MB)
0000_splunk_monitoring	ASML-indexers-OI-Splunk_monitoring-indexes	auto	814	14	60	365	11,000	20,000	180,000	290,000	221	49,093	365	191,000	310,000
accesspoints	ASML-indexers-accesspoints-indexes	auto	1	1000	308	365	1,000	1,000	9,000	9,000	9000	308	365	10,000	10,000
acms	ASML-indexers-ACMS-indexes	auto	0			365	1,000	1,000	9,000	9,000		72	365	10,000	10,000
acs	ASML-indexer-acs-indexes	auto	2	5000	3504	365	10,000	1,000	85,000	9,000	42500	7,008	365	95,000	10,000
activedirectory	ASML-indexers-activedirectory-indexes	auto	4	250	266	365	1,000	1,000	9,000	9,000	2250	1,062	365	10,000	10,000
alerts	TA-alert_manager	auto	4	250	71	365	1,000	1,000	9,000	9,000	2250	284	365	10,000	10,000
apm	ASML-indexers-APM-indexes	auto	5	200	298	365	1,000	1,000	9,000	9,000	1800	1,489	365	10,000	10,000
aveksa	ASML-indexer-IAM_portal-indexes	auto	0			365	3,000	1,000	2,000	9,000		55	365	5,000	10,000
avg_fw	ASML-indexer-avg_fw-indexes	auto	43	395	29	365	17,000	1,000	300,000	29,000	6977	1,239	365	317,000	30,000
bi-mmsw	ASML-indexer-MMSW-indexes	auto	2	500	3362	1825	1,000	1,000	9,000	9,000	4500	6,724	1825	10,000	10,000
cisco	ASML-indexer-cisco-indexes	auto	131	38	269	365	5,000	5,000	65,000	55,000	496	35,258	365	70,000	60,000
citrix	ASML-indexer-citrix-indexes	auto	10	100	295	365	1,000	1,000	9,000	9,000	900	2,948	365	10,000	10,000
clientmonitoring	ASML-indexers-clientmonitoring-indexes	auto	0			365	1,000	1,000	9,000	9,000		2	365	10,000	10,000
crosscheck	ASML-indexer-crosscheck-indexes	auto	5	200	410	365	1,000	1,000	9,000	9,000	1800	2,049	365	10,000	10,000
cws	ASML-indexer-CWS-indexes	auto	720	28	478	365	20,000	10,000	325,000	260,000	451	343,830	365	345,000	270,000
cyberark	ASML-indexer-cyberark-indexes	auto	1	1000	106	365	1,000	1,000	9,000	9,000	9000	106	365	10,000	10,000
darktrace	ASML-indexer-Darktrace-indexes	auto	0			365	1,000	1,000	9,000	9,000		7	365	10,000	10,000
dmp_de	ASML-indexer-DMP-indexes	auto	394	51	35	365	20,000	10,000	50,000	140,000	127	13,952	365	70,000	150,000
dmp_manf	ASML-indexer-DMP-indexes	auto	688	29	4	365	20,000	10,000	50,000	250,000	73	2,505	365	70,000	260,000
dmp_netscanner	ASML-indexer-DMP-indexes	auto	620	32	56	365	20,000	10,000	50,000	230,000	81	34,783	365	70,000	240,000
edirectory	ASML-indexer-edirectory-indexes	auto	1314	30	218	365	40,000	30,000	670,000	460,000	510	286,186	365	710,000	490,000
eseries	TA-netapp_eseries	auto	322	31	165	365	10,000	10,000	140,000	120,000	435	53,232	365	150,000	130,000
f5	ASML-indexer-f5-indexes	auto	9	111	445	365	1,000	1,000	9,000	9,000	1000	4,001	365	10,000	10,000
firepower	ASML-indexer-FirePower-indexes	auto	16	625	38	365	10,000	1,000	90,000	19,000	5625	613	365	100,000	20,000
fortigate	ASML-indexer-fortigate-indexes	auto	2564	27	332	365	70,000	50,000	1,860,000	900,000	725	850,231	365	1,930,000	950,000
fwrules	ASML-indexers-fwrules-indexes	auto	3	333	806	365	1,000	1,000	9,000	9,000	3000	2,418	365	10,000	10,000

- Based on Firebrigade App
- Guarantee SLA retention
- Predict growth

splunk> .conf18

AVAILABLE ON GITHUB HERE: https://github.com/pbovy/SplunkDashboardLoadTimeApp/



- Dashboard load time monitoring
- Based on Job Manager REST data
- Identify improvement opportunities with most positive experience impact

splunk> .conf18

RECAP

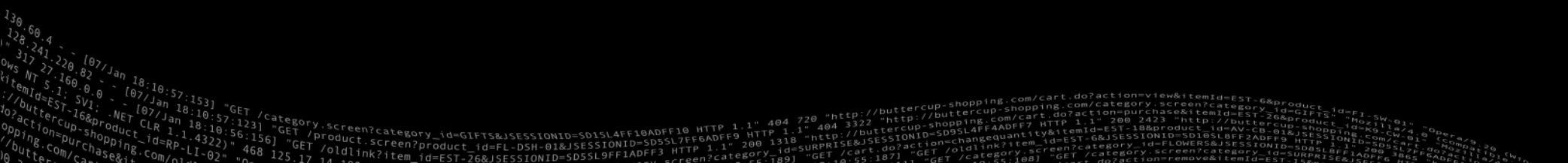
Key Takeaways

1. Check the easy stuff first, tune everything, then check it again
2. Monitor everything
3. I've learned so much from my mistakes, I think I'll make another

Other Great Presentations

Standing on the Shoulders of Giants

- The Critical Syslog Tricks That No One Seems to Know - .Conf 2017
- Architecting Splunk for Epic Performance at Blizzard Entertainment - .Conf 2016
- Quis Custodiet Ipsos Custodes? (Who watches the watchmen?) - .Conf 2016
- How did you get so big? - .Conf 2017
- Best practices and better practices for admins - .Conf 2017
- Worst practices and how to fix them - .Conf 2017



splunk> .conf18

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

Backup Slides

Helpful Monitoring Apps

Imitation is the sincerest form of flattery

All on Splunkbase under respective number

- (D)MC
- Splunk Health Check Overview (#1919)
- Sysadmin (#3760)
- SplunkAdmins (#3796)
- FireBrigade (#1581)
- Broken Hosts app (#3247)

QR Code

Slide 20

```
#!/bin/sh
#
# /etc/init.d/splunk
# init script for Splunk.
# generated by 'splunk enable boot-start'.
#
# chkconfig: 2345 90 60
# description: Splunk indexer service
#
RETVAL=0

. /etc/init.d/functions

# change ulimits
change_ulimit() {
  ulimit -Hn 65535
  ulimit -Sn 65535
  ulimit -Hu 20480
  ulimit -Su 20480
  ulimit -Hf unlimited
  ulimit -Sf unlimited
}

splunk_start() {
  echo Starting Splunk...
  "/opt/splunk/bin/splunk" start --no-prompt --answer-yes
  RETVAL=$?
  [ $RETVAL -eq 0 ] && touch /var/lock/subsys/splunk
}

splunk_stop() {
  echo Stopping Splunk...
  "/opt/splunk/bin/splunk" stop
  RETVAL=$?
  [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/splunk
}

splunk_restart() {
  echo Restarting Splunk...
  "/opt/splunk/bin/splunk" restart
  RETVAL=$?
  [ $RETVAL -eq 0 ] && touch /var/lock/subsys/splunk
}

splunk_status() {
  echo Splunk status:
  "/opt/splunk/bin/splunk" status
  RETVAL=$?
}

case "$1" in
  start)
    change_ulimit
    splunk_start
  ;;
  stop)
    splunk_stop
  ;;
  restart)
    change_ulimit
    splunk_restart
  ;;
  status)
    splunk_status
  ;;
  *)
    echo "Usage: $0 {start|stop|restart|status}"
    exit 1
;;
esac

exit $RETVAL
```

QR Code

Slide 21

```
#!/bin/sh
### BEGIN INIT INFO
# Provides: disable-transparent-hugepages
# Required-Start: $local_fs
# Required-Stop:
# X-Start-Before: splunk
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: Disable Linux transparent huge pages
# Description: Disable Linux transparent huge pages, to improve
# splunk performance.
### END INIT INFO

# Check if we're on Red Hat or CentOS
if [ -d /sys/kernel/mm/transparent_hugepage ]; then
  thp_path=/sys/kernel/mm/transparent_hugepage
elif [ -d /sys/kernel/mm/redhat_transparent_hugepage ]; then
  thp_path=/sys/kernel/mm/redhat_transparent_hugepage
else
  echo "Could not detect THP paths."
  exit 0
fi

case ${1} in
  start)
    echo 'never' > ${thp_path}/enabled
    echo 'never' > ${thp_path}/defrag
  ;;
  stop)
    echo 'always' > ${thp_path}/enabled
    echo 'always' > ${thp_path}/defrag
  ;;
  status)
    echo -n "THP enabled state: "
    cat ${thp_path}/enabled | sed -E "s/.*\[(\w+)\].*/\1/"
    echo -n "THP defrag state: "
    cat ${thp_path}/defrag | sed -E "s/.*\[(\w+)\].*/\1/"
  ;;
esac

unset thp_path
```

QR Code

Slide 26-1

```
index=_internal sourcetype=dispatch ERROR "unknown error" | regex _raw="^[^\s]+\s[^\s]+\sERROR"
| regex _raw="please\s\S{5}\s"
| rex field=_raw "Unknown error for peer (?<peer>[^\.]+)"
| eval clientAndErrorSource=host."-".peer
| timechart span=5m limit=0 count by clientAndErrorSource
```

QR Code

Slide 26-2

```
index=_internal sourcetype=dispatch ERROR "unknown error"
| regex _raw="^[^\s]+\s[^\s]+\sERROR"
| regex _raw="please\s\S{5}\s"
| rex field=_raw "Unknown error for peer (?<peer>[^\.]+)"
| table _time peer _raw
| eval bucketTime=_time
| bucket bucketTime span=1m
| eval windowEarliest=_time-(3600*5), windowLatest=_time+(3600*5), peer=peer." ", raw=_raw." "
| map maxsearches=500 search="search index=_internal sourcetype=scheduler source=/opt/splunk/var/log/splunk/scheduler.log status=* savedsearch_name=* app=* run_time=*
    earliest=$windowEarliest$ latest=$windowLatest$
| fields _time host user app savedsearch_name status dispatch_time run_time
| eval run_time=round(run_time,0)
| eval times=mvrange(dispatch_time,dispatch_time+run_time,60)
| mvexpand times
| rename times as _time, host as searchHead
| bucket _time span=1m
| search _time=$bucketTime$
| stats values(savedsearch_name) as savedsearch_name by _time
| eval details=$raw$"
| table _time details savedsearch_name
| rename savedsearch_name as "runningSavedSearch"
```

QR Code

Slide 26-3

```
search index=_internal sourcetype=dispatch ERROR unknown error earliest=@d-1d latest=now
    | dedup _raw host
    | regex _raw="^[^\s]+\s[^\s]+\sERROR"
    | regex _raw="please\s\S{5}\s"
    | rex field=_raw "Unknown error for peer (?<peer>[^\.]+)"
    | fields _time _raw host peer]
| sort 50000 - _time
| dedup _raw host
| table _time _raw host peer
| | rename comment as "We want to answer the question: at a given moment, what is the chance we get an unknown error for a host-peer combination in the next 15 minutes?"
| rename comment as "Base the data on the last month of unknown errors"
| eval earliest=relative_time(now(),"@d-14d"), latest=relative_time(now(),"@d")
| where _time>earliest AND _time<latest


| rename comment as "Generic fields"
| eval originalTime=strftime(_time,"%c"), hostPeer=host."-".peer, monthDay=strftime(_time,"%m-%d")


| rename comment as "Translate the _time timestamp to today, This alows to chart the errors in a single timechart."
| eval _time=_time+( relative_time(now(),"@d")-relative_time(_time,"@d") )


| rename comment as "Bucket the time for whole minutes, make sure that 13:15:23 => 13:15:23 by adding 60 seconds to the _time"
| eval _time=_time+60
| bucket _time span=1m
| eval bucketTime=strftime(_time,"%c")


| rename comment as "Expand every unknown error into 3 events, this is to solve issues on the left and right of the timerange"
| eval times=mvrange(_time-86400,_time+86401,86400)
| mvexpand times
| eval _time=times
| fields - times


| rename comment as "The events were duplicated in three, now, delete a big part of them by only selecting 'today' +/- 1 hour"
| eval earliest=relative_time(now(),"@d-1h"), latest=relative_time(now(),"@d+1d+1h")
| where _time>earliest AND _time<latest


| rename comment as "Expand every unknown error into 15 events, starting 15 minutes before the unknown error. "
| eval times=mvrange(_time-900-1,_time+1,60)
| mvexpand times
| eval _time=times
| fields - times


| eventstats dc(monthDay) as monthDayCountByTime by hostPeer _time
| eventstats dc(monthDay) as numberOfMonthDays


| rename comment as "The events were duplicated in three, now, delete the left and right one by only selecting 'today'"
| eval earliest=relative_time(now(),"@d"), latest=relative_time(now(),"@d+1d")
| where _time>earliest AND _time<latest


| rename comment as "translate the monthDayCountByTime into a chance"
| eval errorChanceWithin15minutes=round(monthDayCountByTime/numberOfMonthDays*100,1)


| rename comment as "Select only hostPeers whose max chance on unknown errors is higher than 10%"
| eventstats max(errorChanceWithin15minutes) as maxErrorChanceWithin15minutes by hostPeer
| search maxErrorChanceWithin15minutes>10


| table _time hostPeer errorChanceWithin15minutes monthDayCountByTime numberOfMonthDays maxErrorChanceWithin15minutes
| timechart limit=0 span=5m max(errorChanceWithin15minutes) by hostPeer
```

QR Code

Slide 39

```
| rest /services/data/indexes count=0

| table title, eai:acl.app, repFactor, currentDBSizeMB, frozenTimePeriodInSecs, homePath.maxDataSizeMB, coldPath.maxDataSizeMB, maxTotalDataSizeMB,splunk_server

| rename title as Index, eai:acl.app as App, currentDBSizeMB as CurrentTotalSizeOnDiskMB, homePath.maxDataSizeMB as CurrentHotWarmDefMB, coldPath.maxDataSizeMB as CurrentColdDefMB, maxTotalDataSizeMB as CurrentTotalDefMB

| eval CurrentfrozenTime=round(frozenTimePeriodInSecs/86400,0)

| eval 2BFrozenTime=if(CurrentfrozenTime<181,365,CurrentfrozenTime)

| search splunk_server=ics106061038

| join type=inner Index

    [ search index=index_utilization_summary

    | stats avg(total_volume) AS AvgGB by idx

    | rename idx as Index

    | eval AvgMB=round(1000*AvgGB/16,0)

    | table Index, AvgMB]

| eval 2BWarmDef=if(round(20*AvgMB,0)<1000,1000,if(round(20*AvgMB,0)<5000,5000,round(20*AvgMB/10000,0)*10000))

| eval 2BTotalDef=(round(if(CurrentfrozenTime<181,365,CurrentfrozenTime)*AvgMB/10000,0)+1)*10000

| eval 2BColdDef=((round(if(CurrentfrozenTime<181,365,CurrentfrozenTime)*AvgMB/10000,0)+1)*10000)-(if(round(20*AvgMB,0)<1000,1000,if(round(20*AvgMB,0)<5000,5000,round(20*AvgMB/10000,0)*10000)))

| eval CurrentWarmRetention=round(CurrentHotWarmDefMB/AvgMB,0)

| eval CurrentColdRetentionDef=round(CurrentColdDefMB/AvgMB,0)

| eval CurrentColdRetentionOnDisk=round(CurrentTotalSizeOnDiskMB/AvgMB,0)

| rename repFactor AS "Replication Factor", CurrentfrozenTime AS "Current Frozen Time (Days)", AvgMB AS "Avg MB Per Day", CurrentHotWarmDefMB AS "Current Hot Warm Definition (MB)", 2BWarmDef AS "To Be Warm Definition (MB)", CurrentWarmRetention AS "Current Warm Retention (Days)", CurrentColdDefMB AS "Current Cold Definition (MB)", 2BColdDef AS "To Be Cold Definition (MB)", CurrentColdRetentionOnDisk AS "Current Cold Retention on Disk (Days)",CurrentColdRetentionDef AS "Current Cold Retention Definition (Days)", 2BFrozenTime AS "To Be Frozen Time (Days)", CurrentTotalDefMB AS "Current Total Definition (MB)", 2BTotalDef AS "To Be Total Definition (MB)", CurrentTotalSizeOnDiskMB AS "Current Total Size On Disk (MB)"

| table Index, App, "Replication Factor", "Avg MB Per Day", "Current Warm Retention (Days)", "Current Cold Retention on Disk (Days)", "Current Frozen Time (Days)", "Current Hot Warm Definition (MB)", "To Be Warm Definition (MB)", "Current Cold Definition (MB)", "To Be Cold Definition (MB)", "Current Cold Retention Definition (Days)", "Current Total Size On Disk (MB)", "To Be Frozen Time (Days)", "Current Total Definition (MB)", "To Be Total Definition (MB)"
```