



GitLab

The Tyranny of the Clock

Linux.conf.au 2020: Sysadmin Miniconf
Jan 13th

Craig Miskell - SRE, GitLab





```
ssh_exchange_identification: connection closed by  
remote host  
fatal: Could not read from remote repository
```



~~DNS~~

Firewall/proxy

~~DNS, it's always DNS~~

Problem at *The Other End*

Capturing some data



SRE team, inspecting ~26 million connections/day

<https://unsplash.com/photos/2Huoqyf8DDE> - @matthew_t_rader

Packet captures



1-1.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 48

No.	Time	Source	Destination	Protocol	Length	Info
3867	2019-07-04 21:53:02.458988	35.190.168.187	35.190.168.187	TCP	76	35062 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=56648773 TSecr=0 W
3868	2019-07-04 21:53:02.459038	35.190.168.187	35.190.168.187	TCP	76	443 → 35062 [SYN, ACK] Seq=0 Ack=1 Win=28160 Len=0 MSS=1420 SACK_PERM=1 TSval=2664408
3869	2019-07-04 21:53:02.557115	35.190.168.187	35.190.168.187	TCP	68	35062 → 443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=56648871 TSecr=2664408375
3870	2019-07-04 21:53:02.557239	35.190.168.187	35.190.168.187	TCP	89	35062 → 443 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=21 TSval=56648872 TSecr=2664408375
3871	2019-07-04 21:53:02.557256	35.190.168.187	35.190.168.187	TCP	68	443 → 35062 [ACK] Seq=1 Ack=22 Win=28160 Len=0 TSval=2664408400 TSecr=56648872
3872	2019-07-04 21:53:02.557956	35.190.168.187	35.190.168.187	TCP	68	443 → 35062 [FIN, ACK] Seq=1 Ack=22 Win=28160 Len=0 TSval=2664408400 TSecr=56648872
3873	2019-07-04 21:53:02.656097	35.190.168.187	35.190.168.187	TCP	68	35062 → 443 [FIN, ACK] Seq=22 Ack=2 Win=29312 Len=0 TSval=56648970 TSecr=2664408400
3874	2019-07-04 21:53:02.656138	35.190.168.187	35.190.168.187	TCP	68	443 → 35062 [ACK] Seq=2 Ack=23 Win=28160 Len=0 TSval=2664408425 TSecr=56648970

Frame 3867: 76 bytes on wire (608 bits), 76 bytes captured (608 bits)
Linux cooked capture
Internet Protocol Version 4, Src: ..., Dst: 35.190.168.187
Transmission Control Protocol, Src Port: 35062, Dst Port: 443, Seq: 0, Len: 0

0000 00 00 00 01 00 06 42 01 0a d8 03 01 64 32 08 00B.....d2..
0010 45 60 00 3c de d9 40 00 33 06 51 f9E<...3Q...
0020 23 be a8 bb 88 f6 01 bb 81 ec 1d 6f 00 00 00 00 #.....o...
0030 a0 02 72 10 2d b4 00 00 02 04 05 b4 04 02 08 0ar.....
0040 03 60 64 45 00 00 00 00 01 03 03 07dE.....

Wireshark · Follow TCP Stream (tcp.stream eq 48) · 1-1.pcap

SSH-2.0-OpenSSH_7.7

1 client pkt, 0 server pkts, 0 turns.

Entire conversation (21 bytes) Show and save data as ASCII Stream 48

Find: Find Next

Help Filter Out This Stream Print Save as... Back Close



~~DNS~~

Firewall/proxy

~~DNS, it's always DNS~~

Problem at *The Other End*

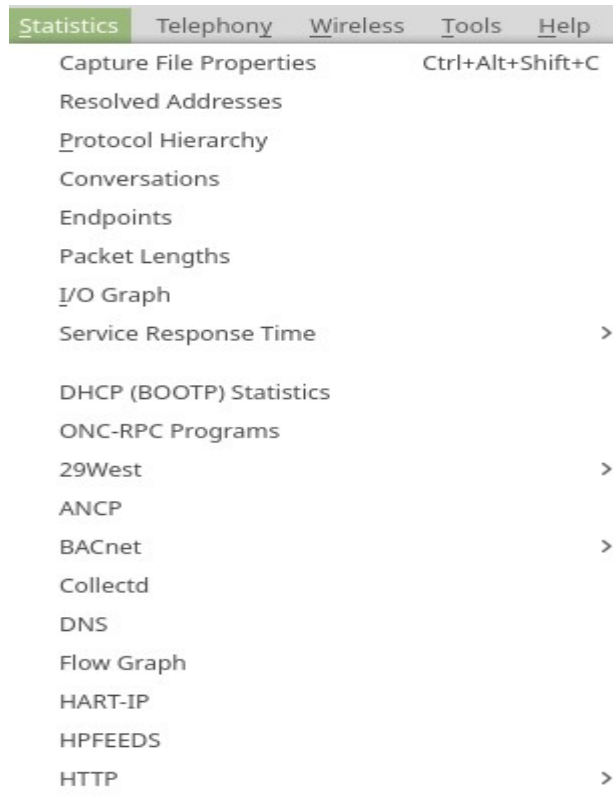


Lesson #1:

Wireshark has lots of analysis tools

You should know:

1. They exist
2. Vaguely what they might be able to do



Conversations



Wireshark · Conversations · 1-1.pcap

Ethernet IPv4 · 1 IPv6 TCP · 893 UDP

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
165.227.164.44	35062	35.190.168.187	443	8	581	4	301	4	280	355.364513	0.1972	12 k	11 k
165.227.164.44	51812	35.190.168.187	443	8	581	4	301	4	280	1676.526316	0.1973	12 k	11 k
165.227.164.44	50524	35.190.168.187	443	8	581	4	301	4	280	2156.500751	0.1993	12 k	11 k
165.227.164.44	45492	35.190.168.187	443	8	581	4	301	4	280	2577.330465	0.1969	12 k	11 k
165.227.164.44	40574	35.190.168.187	443	8	581	4	301	4	280	2695.559340	0.1980	12 k	11 k
165.227.164.44	42950	35.190.168.187	443	8	581	4	301	4	280	3535.547009	0.1976	12 k	11 k
165.227.164.44	48286	35.190.168.187	443	8	581	4	301	4	280	3955.558162	0.1972	12 k	11 k
165.227.164.44	57782	35.190.168.187	443	8	581	4	301	4	280	7077.904880	0.2061	11 k	10 k
165.227.164.44	35240	35.190.168.187	443	8	581	4	301	4	280	7135.422444	0.1969	12 k	11 k
165.227.164.44	53934	35.190.168.187	443	45	10 k	22	4,589	23	5,441	29.785366	1.4654	25 k	29 k
165.227.164.44	54634	35.190.168.187	443	45	10 k	22	4,589	23	5,441	91.808260	1.4643	25 k	29 k
165.227.164.44	34298	35.190.168.187	443	45	10 k	22	4,589	23	5,441	186.662641	1.5414	23 k	28 k
165.227.164.44	38570	35.190.168.187	443	45	10 k	22	4,589	23	5,441	285.415039	1.4911	24 k	29 k
165.227.164.44	58172	35.190.168.187	443	45	10 k	22	4,589	23	5,441	402.069511	1.4854	24 k	29 k
165.227.164.44	40686	35.190.168.187	443	45	10 k	22	4,589	23	5,441	471.626509	1.4664	25 k	29 k
165.227.164.44	37824	35.190.168.187	443	45	10 k	23	4,657	22	5,373	496.982369	1.5788	23 k	27 k
165.227.164.44	40062	35.190.168.187	443	45	10 k	22	4,589	23	5,441	683.197344	1.4651	25 k	29 k
165.227.164.44	40798	35.190.168.187	443	45	10 k	22	4,589	23	5,441	745.243296	1.4707	24 k	29 k
165.227.164.44	41538	35.190.168.187	443	45	10 k	22	4,589	23	5,485	807.263141	1.5462	23 k	28 k
165.227.164.44	38620	35.190.168.187	443	45	10 k	22	4,589	23	5,441	1150.002860	1.5046	24 k	28 k
165.227.164.44	39330	35.190.168.187	443	45	10 k	22	4,589	23	5,441	1212.053023	1.5110	24 k	28 k
165.227.164.44	40770	35.190.168.187	443	45	10 k	22	4,589	23	5,441	1336.182252	1.5037	24 k	28 k
165.227.164.44	50874	35.190.168.187	443	45	10 k	23	4,657	22	5,373	1342.224522	1.5173	24 k	28 k
165.227.164.44	48134	35.190.168.187	443	45	10 k	22	4,589	23	5,485	1365.681355	1.4626	25 k	30 k
165.227.164.44	41504	35.190.168.187	443	45	10 k	23	4,657	22	5,453	1398.242332	1.5035	24 k	29 k

☐ Name resolution ☐ Limit to display filter ☐ Absolute start time

Conversation Types ▾

Help Copy ▾ Follow Stream... Graph... Close

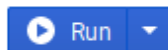
Logs logs and more logs - BigQuery to the rescue



Query editor

[HIDE EDITOR](#)[FULL SCREEN](#)

```
1 SELECT * FROM `gitlab-production.haproxy_logs.haproxy_20191208`  
2 WHERE jsonPayload.frontend_name like 'altssh'  
3 AND jsonPayload.c_port = '49898'  
4 AND jsonPayload.c_ip = '  
5 order by jsonPayload.t
```

[Save query](#)[Save view](#)[Schedule query](#)[More](#)

This query will process 183.6 GB when run.



Query results

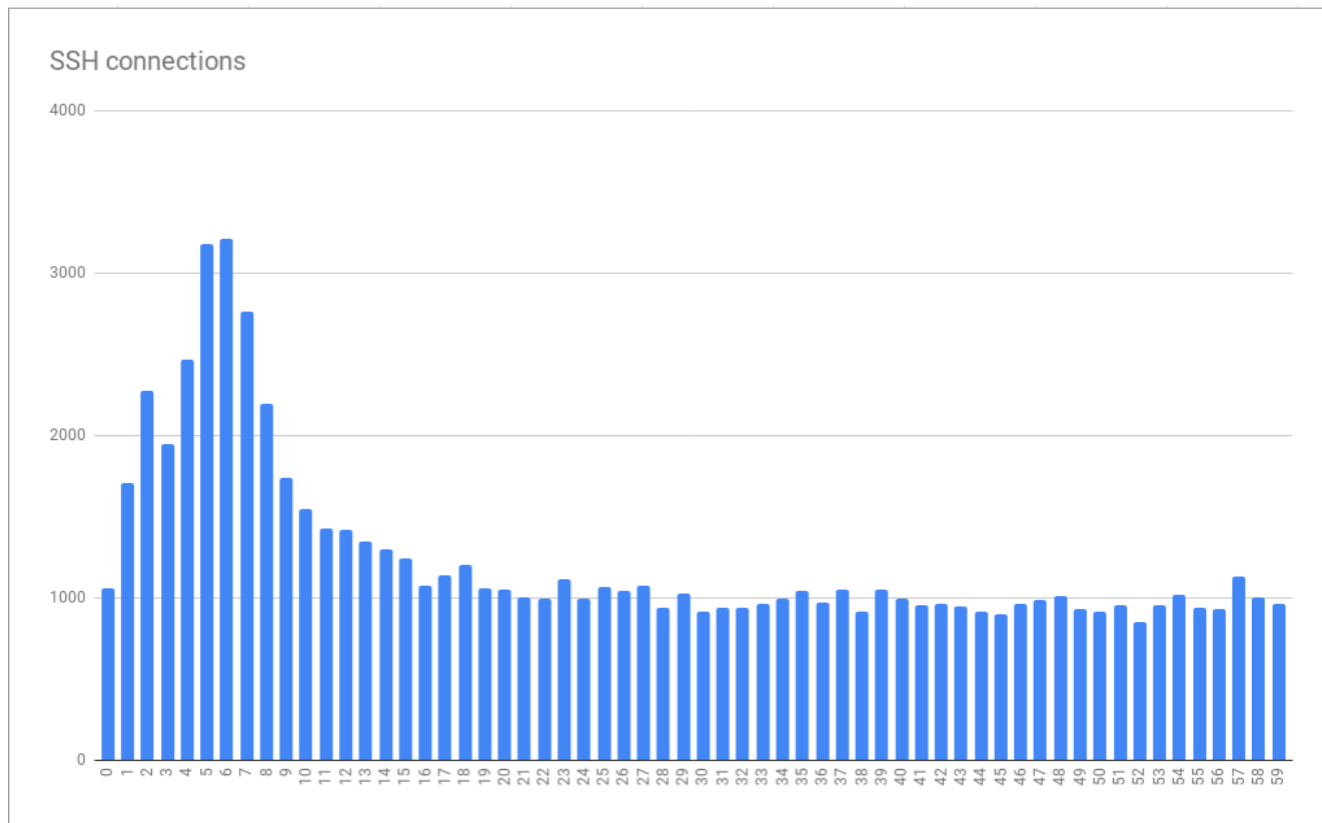
[SAVE RESULTS](#)[EXPLORE WITH DATA STUDIO](#)

oad.hostname	jsonPayload.environment	jsonPayload.verb	jsonPayload.t	jsonPayload.ssl_version	jsonPayload.tw	jsonPayload.statu
102-lb-gprd	gprd	null	08/Dec/2019:18:25:09.507	-	1	null



S: aborted by the server, or the server explicitly refused it
D: the session was in the DATA phase.

An illuminating graph



Connection errors, grouped by second-of-the-minute



**Lesson #2: Apparently a lot of people have time synchronization set up
properly.**

Yay?



MaxStartups

Specifies the maximum number of concurrent **unauthenticated** connections to the SSH daemon. Additional connections will be dropped until authentication succeeds or the LoginGraceTime expires for a connection. The default is **10:30:100**.

Alternatively, random early drop can be enabled by specifying the three colon separated values start:rate:full (e.g. "10:30:60"). sshd(8) will refuse connection attempts with a probability of rate/100 (30%) if there are currently start (10) unauthenticated connections. The probability increases linearly and all connection attempts are refused if the number of unauthenticated connections reaches full (60).



~~DNS~~

~~Firewall/proxy~~

~~DNS, it's always DNS~~

~~Problem at *The Other End*~~

SSH Configuration Issue



Lesson #3:

It is polite to log interesting information at default levels

Deliberately dropping a connection for any reason is definitely interesting to system administrators.

Bump it



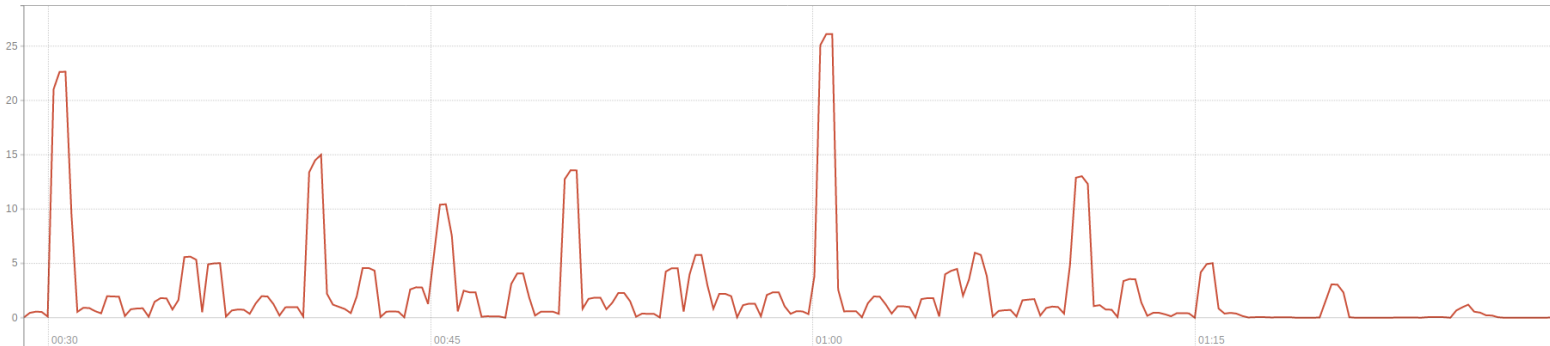
```
sum(rate(haproxy_server_response_errors_total{server=~"git.*"}[1m]))
```

Load time: 430ms
Resolution: 14s
Total time series: 1

Execute - insert metric at cursor -

Graph Console

- 1h + << Until >> Res. (s) stacked



0

How high could we go?



By Leandro Inocencio - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=22166102>

Some extremely dodgy math



$$ratelimit = \frac{B * S}{F * T}$$

F => Front-end (haproxy) server count (18)

B => Back-end server count (27)

S => Session allowed in startup (first value in MaxStartups)

T => How long a session spends in unauthenticated state (in seconds)

ssh						
	Queue			Session rate		
	Cur	Max	Limit	Cur	Max	Limit
Frontend				27	110	110
sock-1						

Surprising data



```
sum(instance_cpu:node_cpu_not_idle:rate5m{tier="sv", environment="gprd", type="git"}) without (cpu)
/
instance_cpu:count{environment="gprd", tier="sv", type="git"}
```

```
sum(instance_cpu:node_cpu_not_idle:rate5m{tier="sv", environment="gprd", type="git"}) without (cpu)
/
instance_cpu:count{environment="gprd", tier="sv", type="git"}
```

Load time: 715ms
Resolution: 14s
Total time series: 27

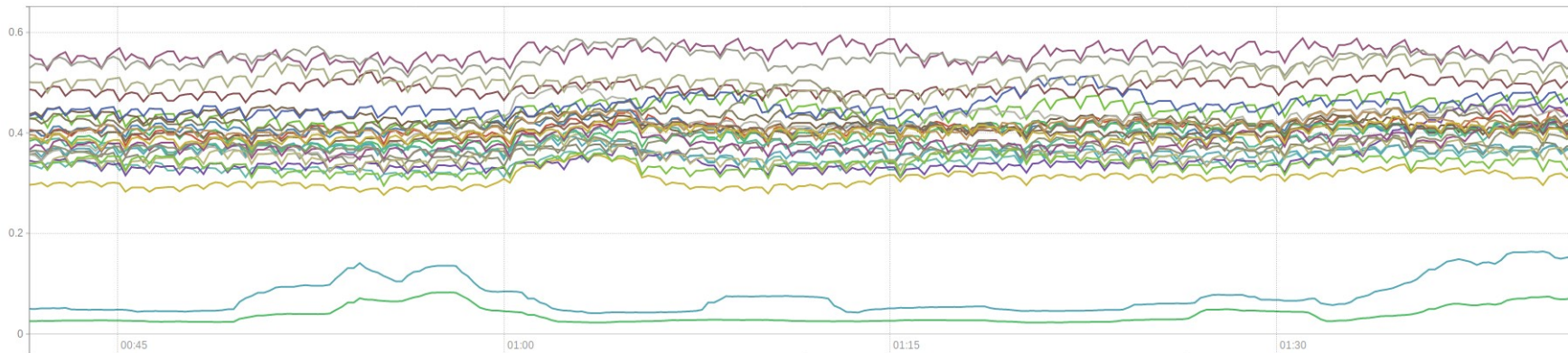
Execute

- insert metric at cursor -

Graph

Console

- 1h + << Until >> Res. (s) stacked





Lesson #4: When you choose specific non-default settings, leave a comment or link to documentation/issues as to why, future people will thank you.

A delightful graph



Load time: 494ms
Resolution: 28s
Total time series: 1

```
sum(rate(haproxy_server_response_errors_total{server=~"git.*"}[1m]))
```

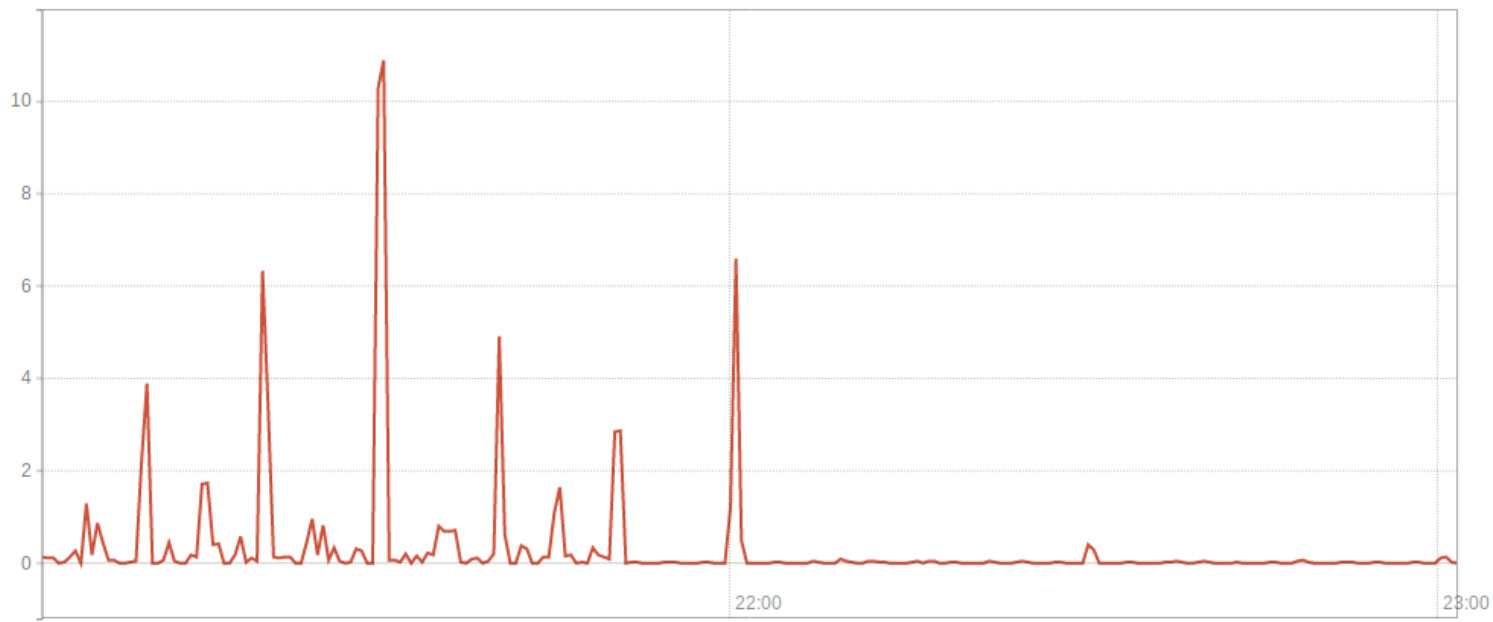
Execute

- insert metric at cursor - ▾

Graph

Console

- 2h + << Until >> Res. (s) ☐ stacked





Lesson #5: As scary as it looks, MaxStartups appears to have very little performance impact even if it's raised much higher than the default.



$$ratelimit = \frac{B * S}{F * T} \quad \text{or} \quad T = \frac{B * S}{F * ratelimit}$$

$$\frac{27 * 200}{18 * 110} = 2.\overline{72}$$

$$\frac{27 * 250}{18 * 110} = 3.4\overline{09}$$



1.5%





Lesson #6: Measure early, measure often

Followup: Why u no alert?



```
rate(haproxy_server_response_errors_total[1m]) > .5  
for: 2m
```

1. Arbitrary limit (0.5), not a ratio
2. Per front-end (18) and back-end (27) server combo
3. Average over 1 minute
4. Had to be elevated for 2 minutes

Also not everything is HTTP (yet)...



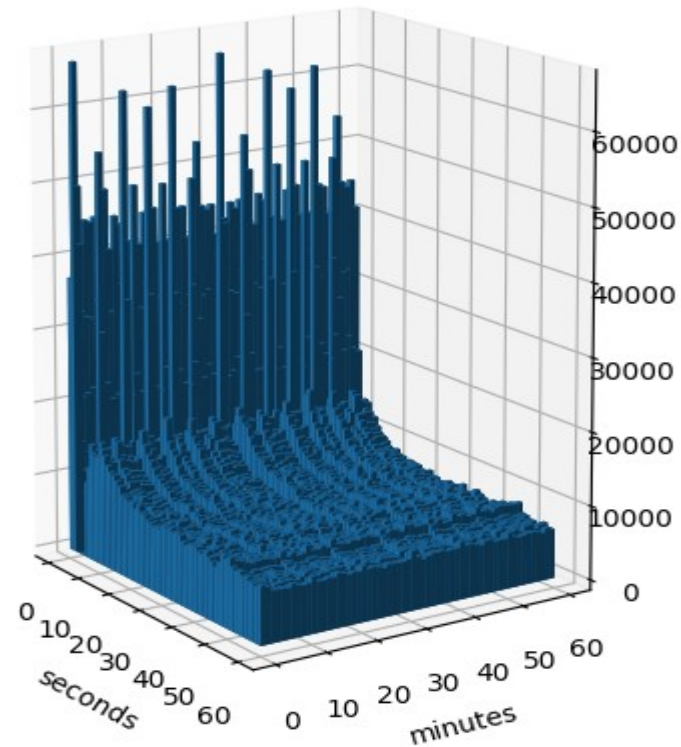


```
$termination_cause == "S" && $termination_state == "D" &&  
$bytes_read == 0 {  
    haproxy_ssh_max_startups_breached_total[$server_name]++  
}
```

Visualize the Tyranny



2019-12-11: grouped by minute/second



Make people like me happy or sad with your scheduling



```
X * * * * sleep $((RANDOM/653 + 5)) && /path/to/script.sh  
    I love you.
```

Or, in a systemd.timer:

RandomizedDelaySec=50

AccuracySec=0

OnCalendar=*-*-* *:X:05



1. The little details matter
2. Assumptions and estimates can be risky
3. Stampeding herds bound by the clock can ruin all your best laid plans

