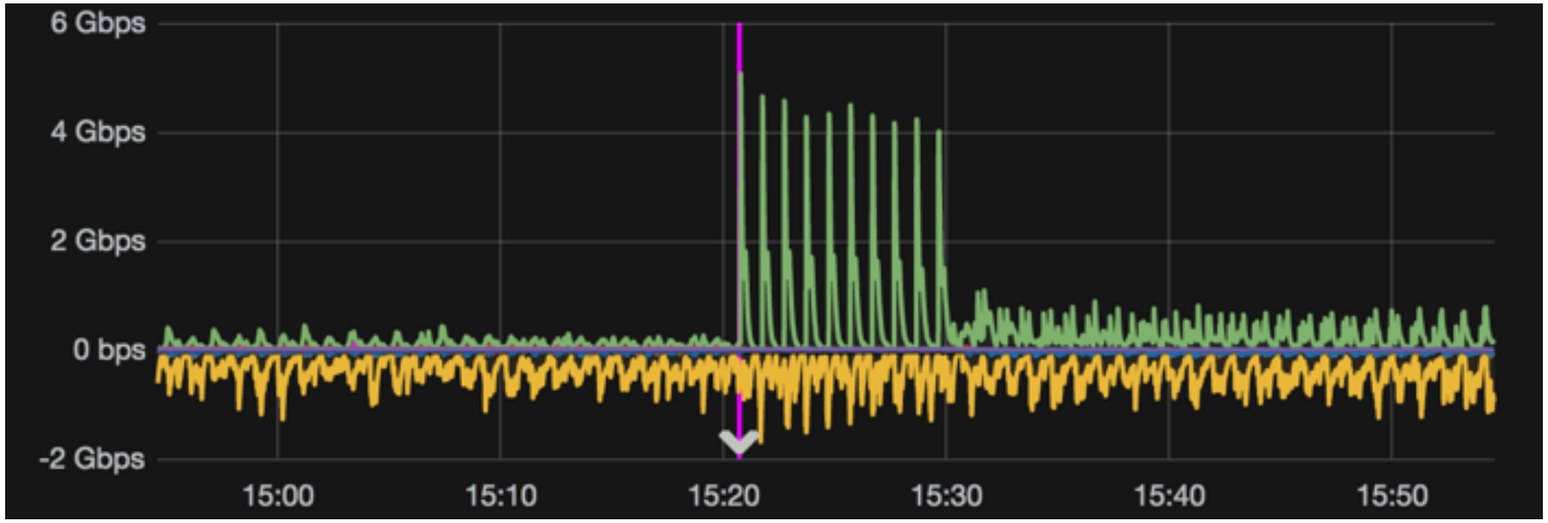
Detecting and Mitigating DDoS

A FastNetMon (+more) use case



AS61186 / AS21880

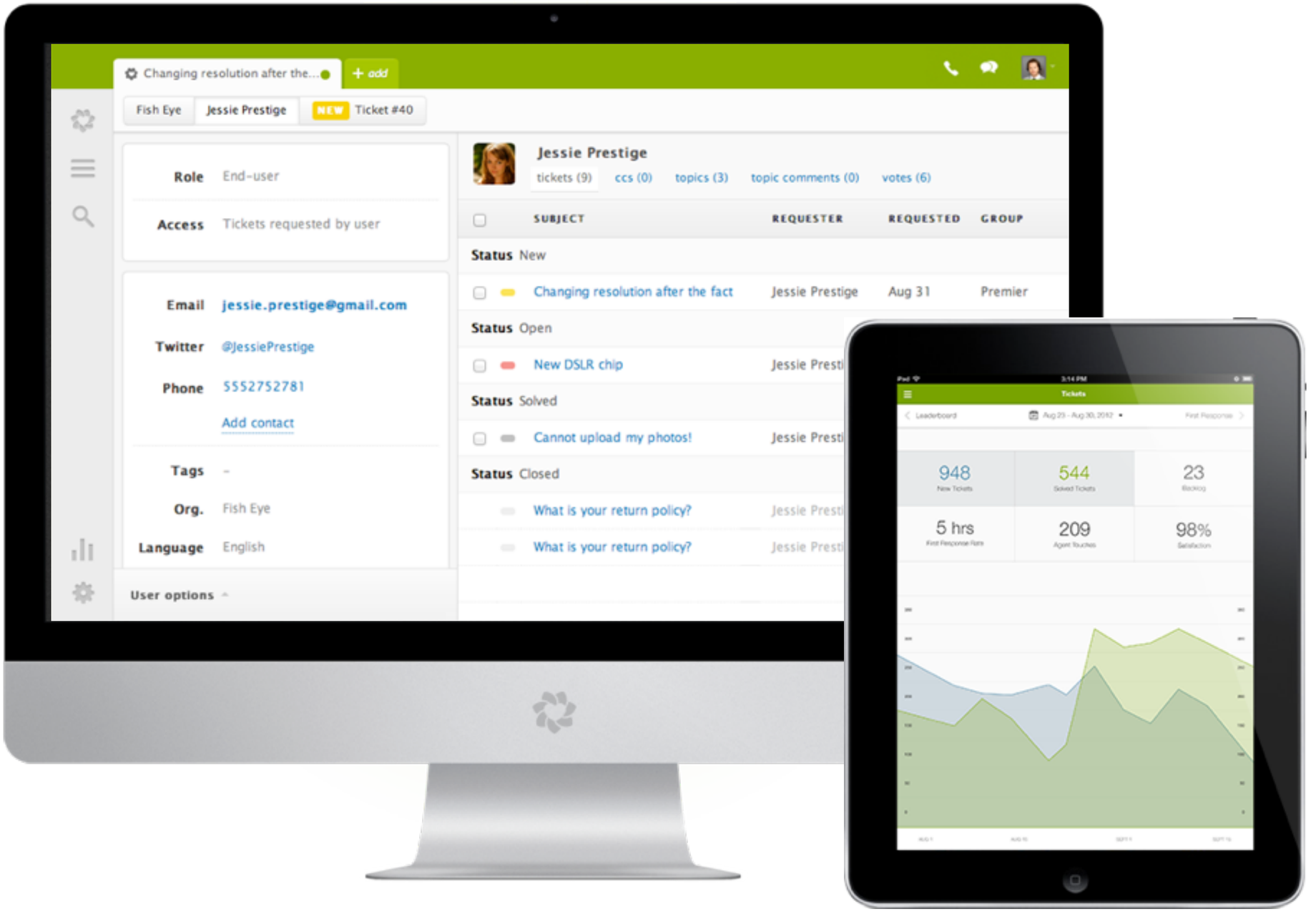
zendesk

Vicente De Luca

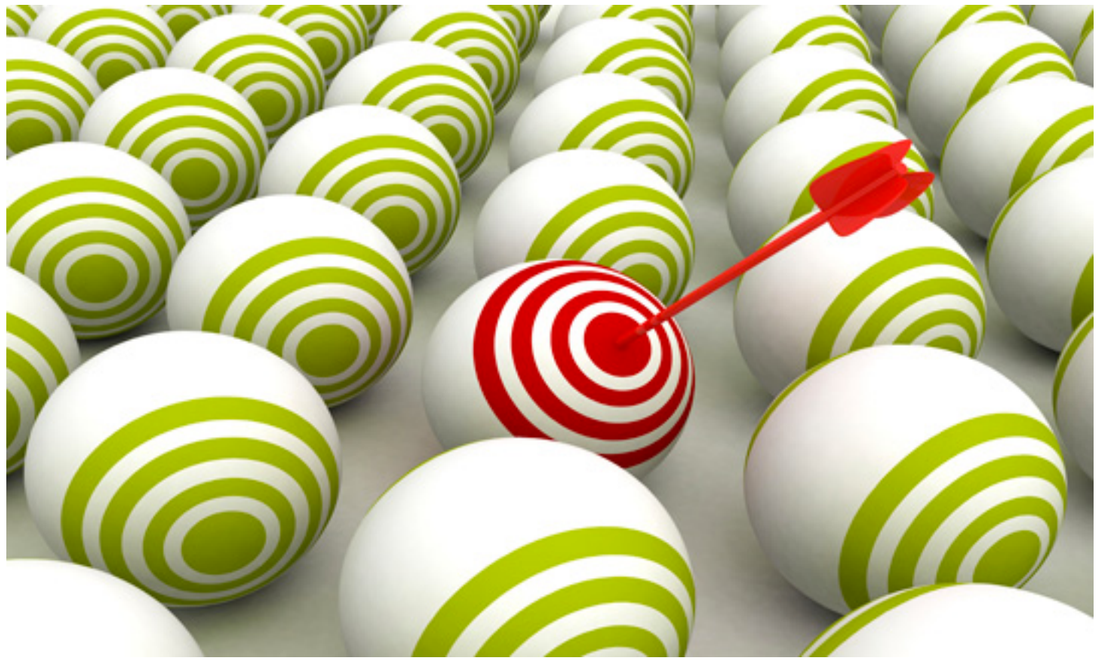
Sr. Network Engineer, ZENDESK

Follow me on Twitter: @zenvdeluca
Follow me on Github: zenvdeluca
Pay me a visit a http://nethero.org

vdeluca@zendesk.com


Changing resolution after the...
+ add
Fish Eye
Jessie Prestige
NEW
Ticket #40
Role
End-user
Access
Tickets requested by user
Email
jessie.prestige@gmail.com
Twitter
@JessiePrestige
Phone
5552752781
Add contact
Tags
-
Org.
Fish Eye
Language
English
User options
Jessie Prestige
tickets (9)
ccs (0)
topics (3)
topic comments (0)
votes (6)
SUBJECT
REQUESTER
REQUESTED
GROUP
Status New
Changing resolution after the fact
Jessie Prestige
Aug 31
Premier
Status Open
New DSLR chip
Status Solved
Cannot upload my photos!
Status Closed
What is your return policy?
What is your return policy?
Tickets
948
544
23
5 hrs
209
98%

Why are we targets?



zendesk

support.acme.com

CNAME

acme.zendesk.com

zendesk

The good, the bad and the ugly

The good

cloud mitigation provider

- six scrubbing centers across the globe
- 3 Tbps of mitigation bandwidth
- BGP: advertise the target /24, receive legit traffic via clean pipe

cons:

- Reaction time: Internet route convergence (BGP)
- mitigation occurs on incoming only
- always on = very pricy

The bad

NOC find out about an ongoing DDoS attack with site-down alert

The ugly

detecting takes too long
all dependent on humans :(

zendesk

Improving detection
Automating mitigation

zendesk

Why not simply use a DDoS mitigation equipment?

- $$$$ + $$$$$$$$$!!!
- Need dedicated qualified network engineers
- Demand significant changes in network architecture
- Not simple to automate with modern APIs out in the wild
- Mitigation is useless in case of channel overflow

Open-source recipe

- FastNetMon: main core of our solution. DDoS analyzer with sflow/netflow/mirror support
- InfluxDB: Scalable data store for metrics, events, and real-time analytics.
- Grafana: Gorgeous metric viz, dashboards & editors
- Redis: An in-memory database that persists on disk.
- Morgoth: Metric anomaly detection for Influx databases
- BIRD: a fully functional dynamic IP routing daemon
- Net Healer: experimental code to "glue" all moving parts, trigger actions and provide API

zendesk

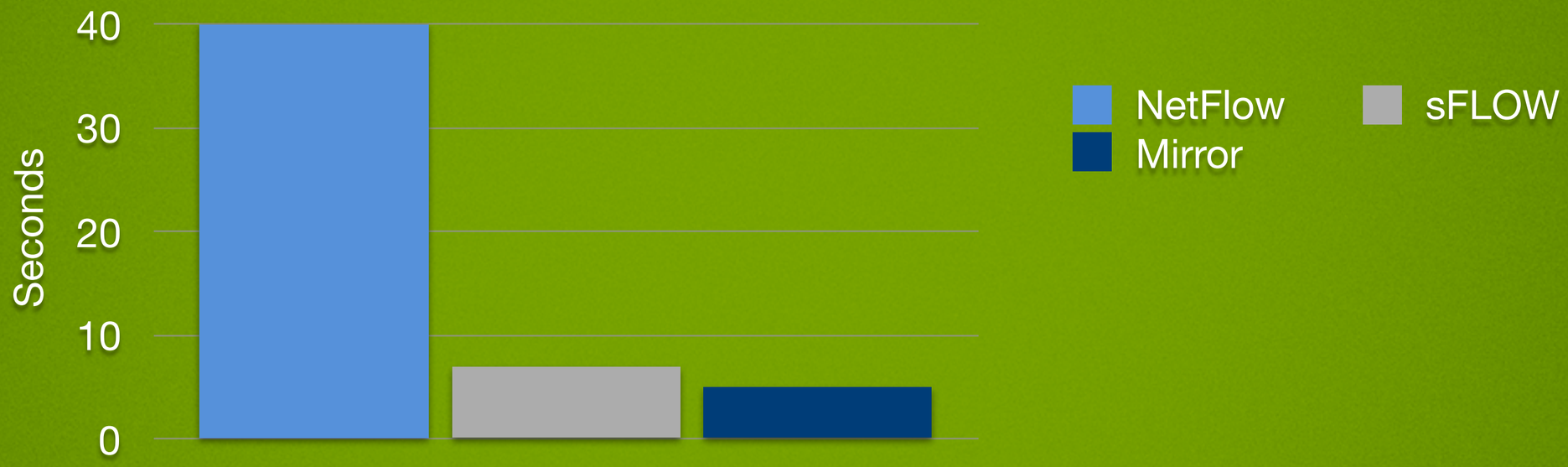
How does it work?

FastNetMon - the main core

- supports sFlow (v4/v5), NetFlow (v5/v9/v10), IPFIX and SPAN/mirror
- officially supported in CentOS / Ubuntu / Debian / Vyatta / FreeBSD
- source code available for compiling on your platform
- fast detect IPv4 targets and notify by email, script or populating RedisDB
- BGP support - can advertise routes with communities (for blackhole), or BGP FlowSpec (RFC 5575)
- Tested with Juniper, Cisco, Extreme, Huawei and Linux (ipt_NETFLOW)

zendesk

FastNetMon: Detection time for capture backends



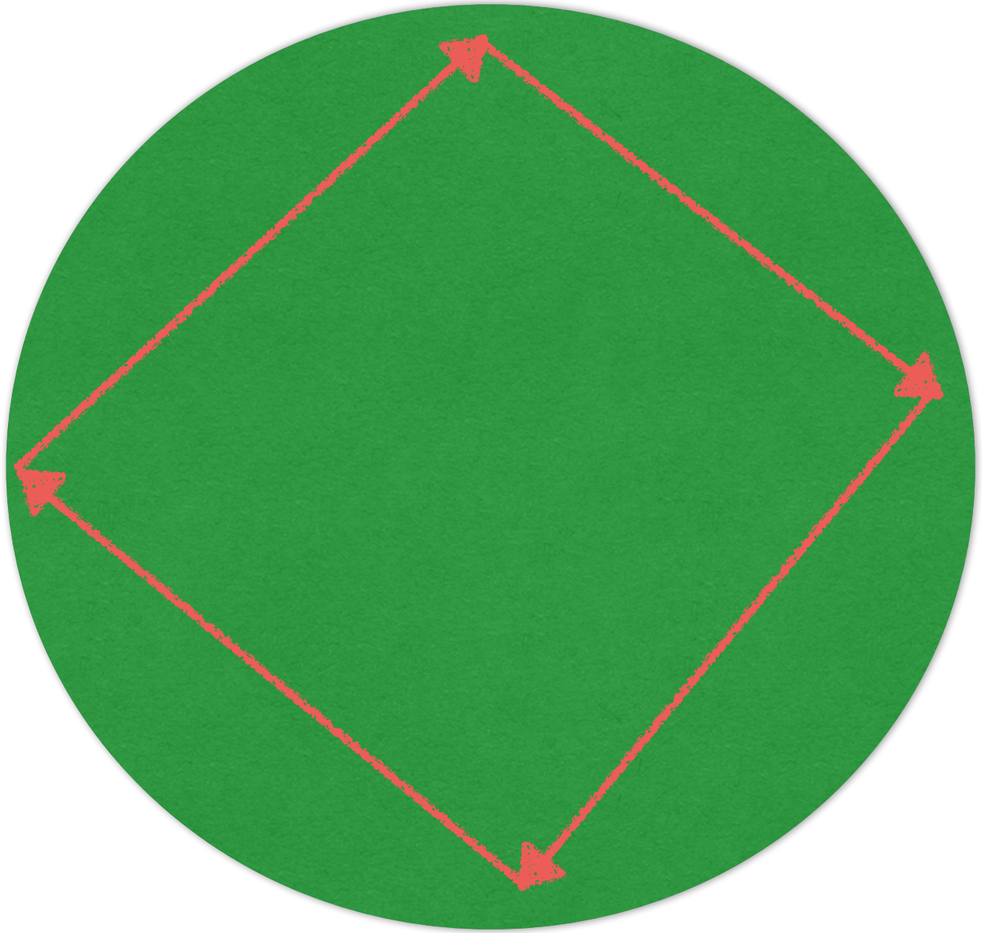
zendesk

FastNetMon - the main core

- Works by configured thresholds for mbps, pps or flows:
  - global
  - host (/32)
  - host group (CIDR)
  - per protocol (tcp/udp/icmp)

our in-house tests

Cycle
DDoS Attack started



FNM quiescence:
15s per /32

FastNetMon detect the attack
ban = add /32 attack report
on Redis DB

Net Healer watches Redis DB
if the attack report(s) match a policy, trigger an action

zendesk

Net Healer Policies example:

(in a time period of 5 min)

if attack reports = 2 then trigger on call
if attack reports >=4 then inject /24 route

if attack report = 1 + anomaly detected (morgoth)
then trigger on call + inject /24 route

time window / policies can be customized

zendesk

Why Net Healer ?

- FNM relies on pre-configured thresholds

- Hard to predict realistic thresholds since our traffic is influenced by our customers activity (out of our control)

- To avoid false positives we prefer to trigger different actions based on each attack cycle.

- Allows quick integrations like Morgoth x FNM consensus, or API calls such as Pagerduty, etc.

Why InfluxDB ?

- Speaks graphite protocol (compatible with FastNetMon)
- Drop in binary - simple install
- Supports cluster mode - easy to scale

zendesk

Why Morgoth ?

- Implements non-gaussian algorithm (MGOF) to detect anomaly on data stream metrics

- Takes InfluxDB (bps/pps) fingerprints every chunk of 10s

- Compares the actual fingerprint with the past learned traffic

- Anomaly detected? create an alert entry and a graph vertical markdown

Why BIRD ?

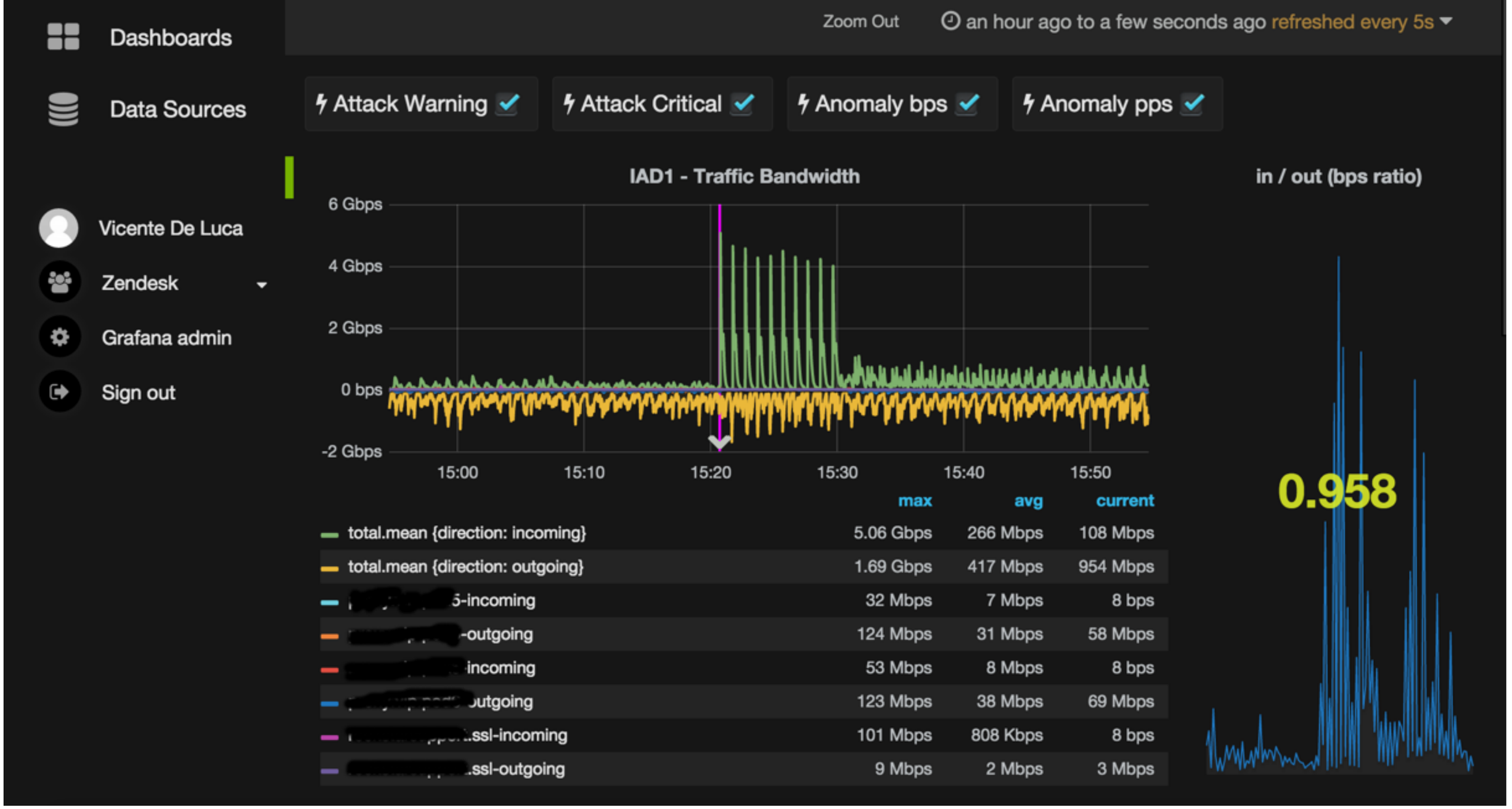
- synced with linux kernel routing table

- if a route is added on a separate linux routing table, BIRD will learn this route and advertise it to my edge routers.
RPL announces the prefix to DDoS scrubbing center

- friendly to Network Engineers (birdc)

How does it look ?

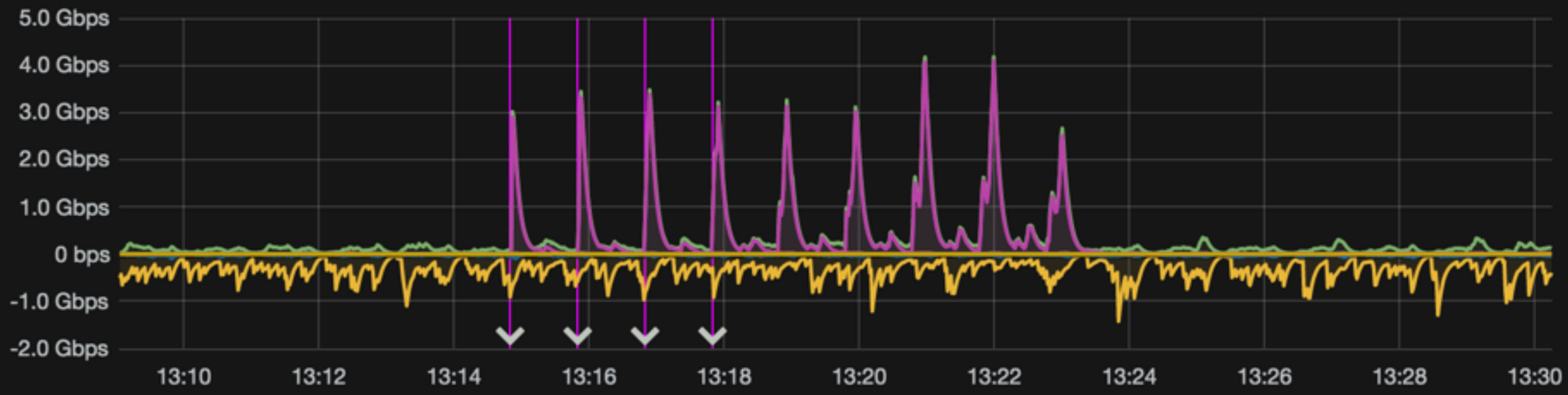
zendesk


Dashboards
Data Sources
Vicente De Luca
Zendesk
Grafana admin
Sign out
Zoom Out
an hour ago to a few seconds ago refreshed every 5s
Attack Warning
Attack Critical
Anomaly bps
Anomaly pps
IAD1 - Traffic Bandwidth
6 Gbps
4 Gbps
2 Gbps
0 bps
-2 Gbps
15:00
15:10
15:20
15:30
15:40
15:50
max
avg
current
total.mean {direction: incoming} 5.06 Gbps 266 Mbps 108 Mbps
total.mean {direction: outgoing} 1.69 Gbps 417 Mbps 954 Mbps
[redacted]-incoming 32 Mbps 7 Mbps 8 bps
[redacted]-outgoing 124 Mbps 31 Mbps 58 Mbps
[redacted]incoming 53 Mbps 8 Mbps 8 bps
[redacted]utgoing 123 Mbps 38 Mbps 69 Mbps
[redacted].ssl-incoming 101 Mbps 808 Kbps 8 bps
[redacted].ssl-outgoing 9 Mbps 2 Mbps 3 Mbps
in / out (bps ratio)
0.958

zendesk

Attack Warning ✓ Attack Critical ✓ Anomaly bps ✓ Anomaly pps ✓

IAD1 - Traffic Bandwidth



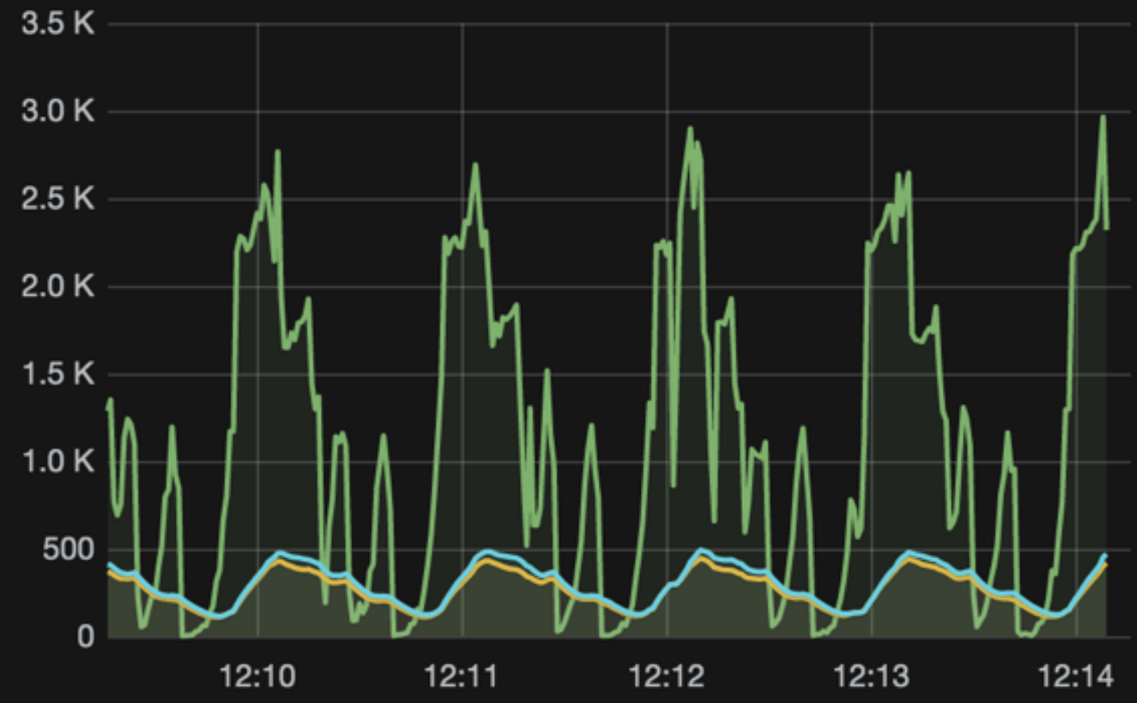
	max	avg	current
total.mean {direction: incoming}	4.177 Gbps	328 Mbps	121 Mbps
total.mean {direction: outgoing}	1.425 Gbps	342 Mbps	397 Mbps
[redacted]-incoming	22 Mbps	7 Mbps	14 Mbps
[redacted]-outgoing	59 Mbps	22 Mbps	36 Mbps
[redacted]-incoming	53 Mbps	9 Mbps	14 Mbps
[redacted]-outgoing	89 Mbps	31 Mbps	49 Mbps
[redacted]-incoming	4.089 Gbps	323 Mbps	8 bps
[redacted]outgoing	7 Mbps	2 Mbps	5 Mbps
[redacted]incoming	4 Mbps	48 Kbps	8 bps
[redacted]outgoing	13 Mbps	144 Kbps	83 Kbps

in / out (bps ratio)

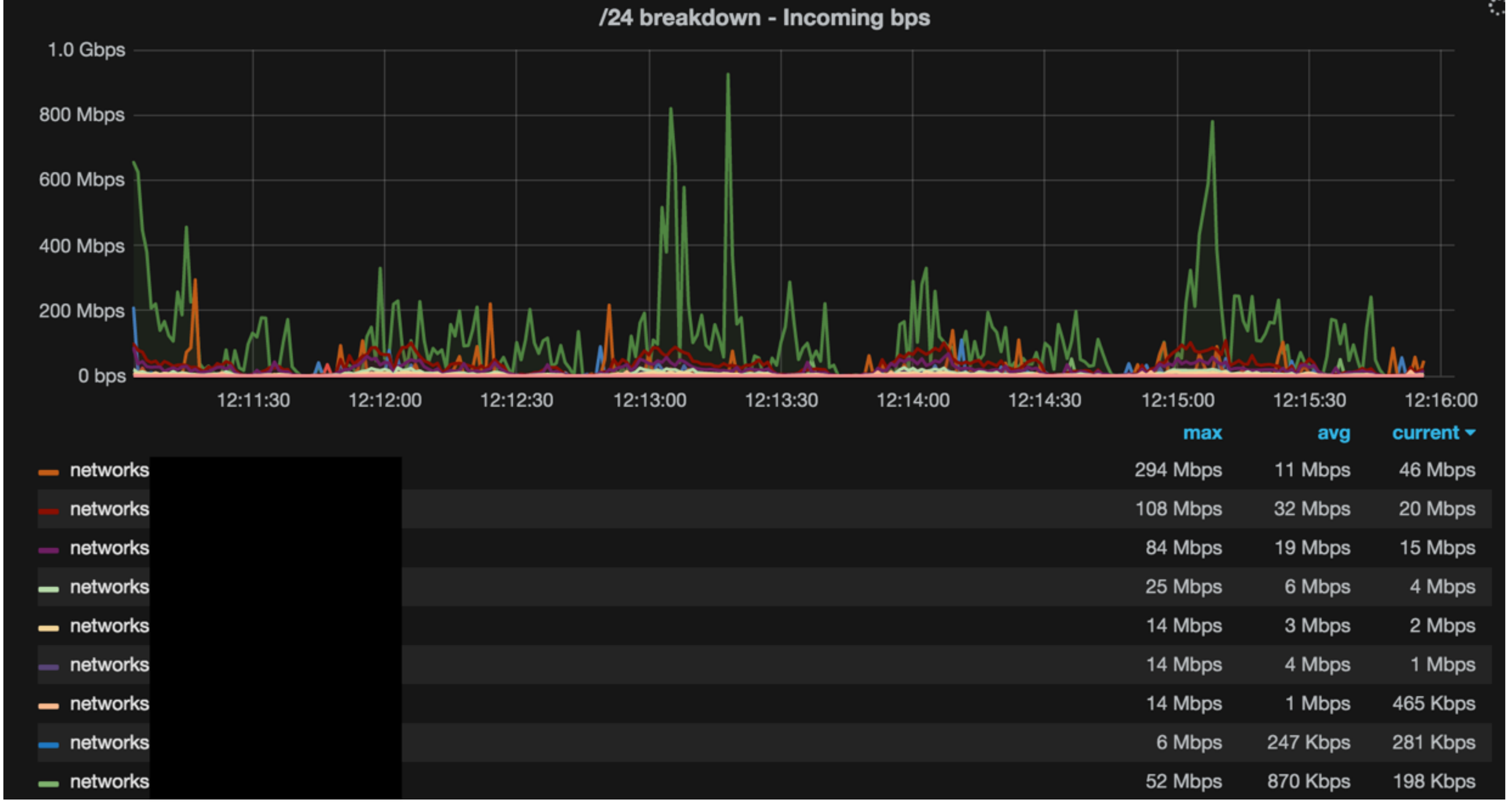
[redacted]1 - Packets per second

	max	avg	current
total-incoming	130 Kpps	79 Kpps	113 Kpps
total-outgoing	194 Kpps	129 Kpps	166 Kpps
[redacted]-incoming	9 Kpps	5 Kpps	8 Kpps
[redacted]outgoing	14 Kpps	6 Kpps	8 Kpps
[redacted]-incoming	9 Kpps	5 Kpps	9 Kpps
[redacted]-outgoing	11 Kpps	7 Kpps	11 Kpps
[redacted]ssl-incoming	337 pps	107 pps	48 pps
[redacted]ssl-outgoing	252 pps	112 pps	70 pps
[redacted]ncoming	15 pps	4 pps	1 pps
[redacted]l-outgoing	17 pps	5 pps	1 pps

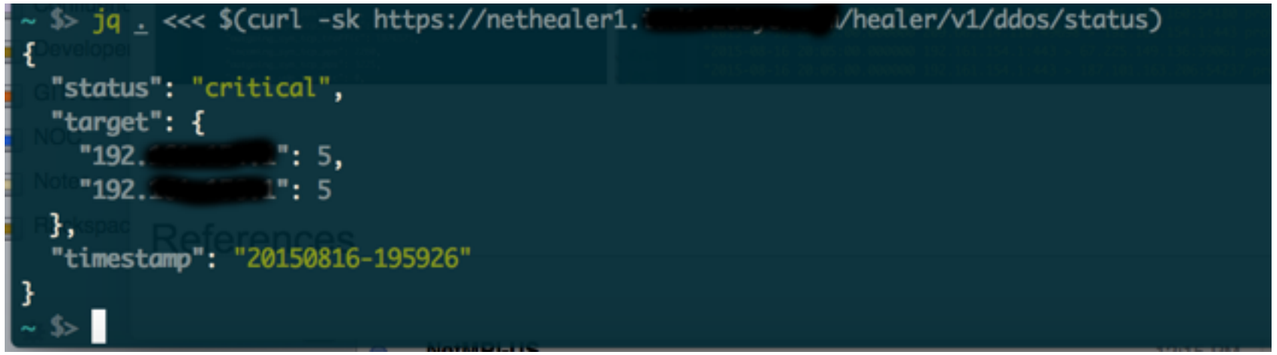
[redacted]1 - Flow amount



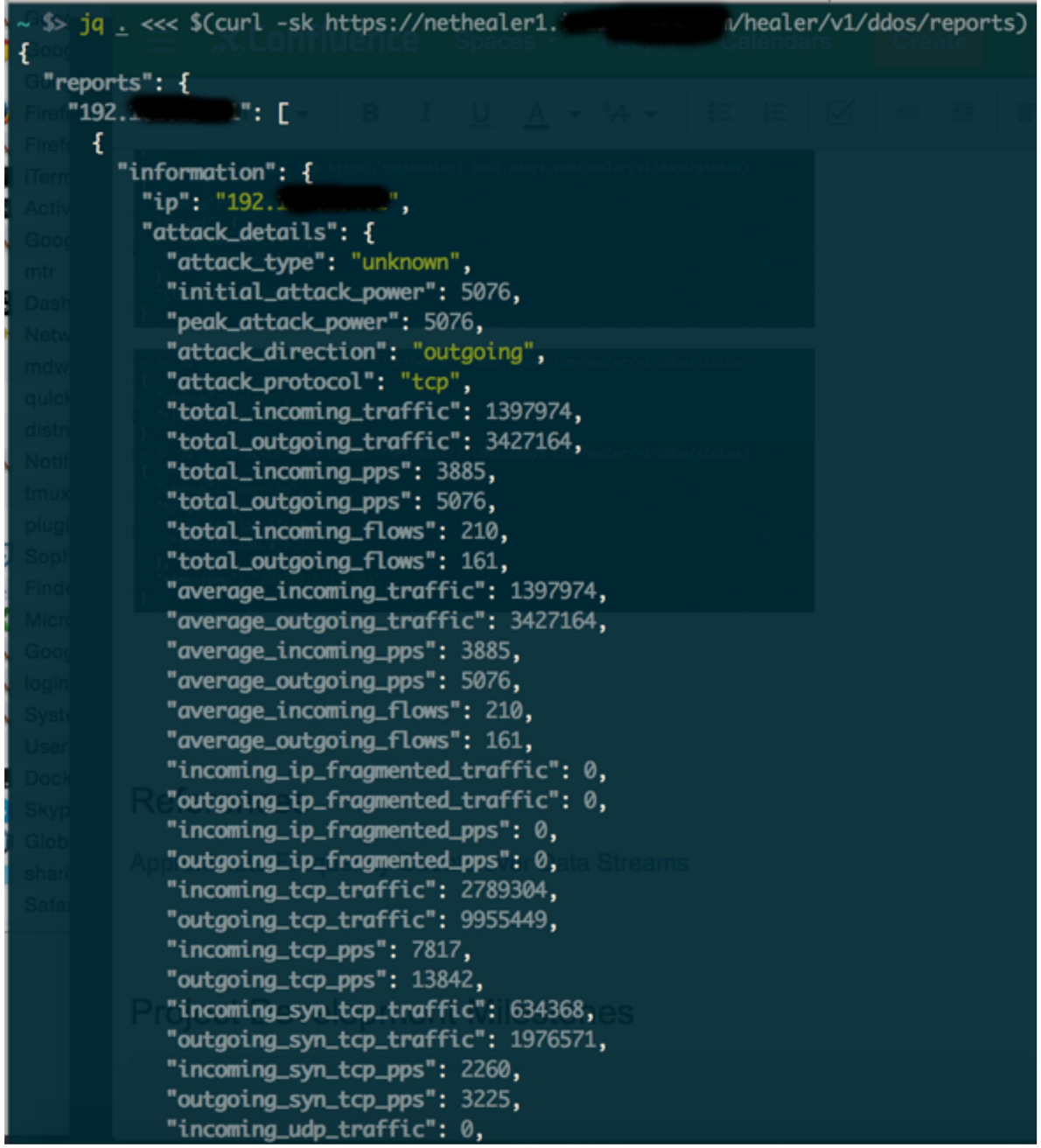
	max	avg	current
total-incoming	2.963 K	1.111 K	2.318 K
[redacted]incoming	445	268	417
[redacted]ncoming	492	297	469
[redacted]ssl-incoming			
[redacted]ncoming			


/24 breakdown - Incoming bps
1.0 Gbps
800 Mbps
600 Mbps
400 Mbps
200 Mbps
0 bps
12:11:30
12:12:00
12:12:30
12:13:00
12:13:30
12:14:00
12:14:30
12:15:00
12:15:30
12:16:00
max
avg
current
networks [redacted] 294 Mbps 11 Mbps 46 Mbps
networks [redacted] 108 Mbps 32 Mbps 20 Mbps
networks [redacted] 84 Mbps 19 Mbps 15 Mbps
networks [redacted] 25 Mbps 6 Mbps 4 Mbps
networks [redacted] 14 Mbps 3 Mbps 2 Mbps
networks [redacted] 14 Mbps 4 Mbps 1 Mbps
networks [redacted] 14 Mbps 1 Mbps 465 Kbps
networks [redacted] 6 Mbps 247 Kbps 281 Kbps
networks [redacted] 52 Mbps 870 Kbps 198 Kbps


~ $> jq . <<< $(curl -sk https://nethealer1.i[redacted]/healer/v1/ddos/status)
{
"status": "clear",
"timestamp": "20150816-195527"
}
~ $> jq . <<< $(curl -sk https://nethealer1.i[redacted]/healer/v1/ddos/status)
{
"status": "warning",
"target": {
"192[redacted]": 3,
"192[redacted]": 3
},
"timestamp": "20150816-195703"
}


~ $> jq . <<< $(curl -sk https://nethealer1.[redacted]/healer/v1/ddos/status)
{
"status": "critical",
"target": {
"192.[redacted]": 5,
"192.[redacted]": 5
},
"timestamp": "20150816-195926"
}
~ $>

zendesk


~ $> jq . <<< $(curl -sk https://nethealer1.[redacted]/healer/v1/ddos/reports)
{
"reports": {
"192.[redacted]": [
{
"information": {
"ip": "192.[redacted]",
"attack_details": {
"attack_type": "unknown",
"initial_attack_power": 5076,
"peak_attack_power": 5076,
"attack_direction": "outgoing",
"attack_protocol": "tcp",
"total_incoming_traffic": 1397974,
"total_outgoing_traffic": 3427164,
"total_incoming_pps": 3885,
"total_outgoing_pps": 5076,
"total_incoming_flows": 210,
"total_outgoing_flows": 161,
"average_incoming_traffic": 1397974,
"average_outgoing_traffic": 3427164,
"average_incoming_pps": 3885,
"average_outgoing_pps": 5076,
"average_incoming_flows": 210,
"average_outgoing_flows": 161,
"incoming_ip_fragmented_traffic": 0,
"outgoing_ip_fragmented_traffic": 0,
"incoming_ip_fragmented_pps": 0,
"outgoing_ip_fragmented_pps": 0,
"incoming_tcp_traffic": 2789304,
"outgoing_tcp_traffic": 9955449,
"incoming_tcp_pps": 7817,
"outgoing_tcp_pps": 13842,
"incoming_syn_tcp_traffic": 634368,
"outgoing_syn_tcp_traffic": 1976571,
"incoming_syn_tcp_pps": 2260,
"outgoing_syn_tcp_pps": 3225,
"incoming_udp_traffic": 0,

zendesk

From: **DDoS Detection**
To: Network Operations

Healer Dashboard: https://netmonitor.

Attack info:
---
192. .1:
fqdn:
site: IAD1
alerts: 2
protocol:
- udp
incoming:
total:
mbps: 263.59
pps: 628056
flows: 523
tcp:
mbps: 0.9
pps: 2790
syn:
mbps: 0.14
pps: 808
udp:
mbps: 646.55
pps: 9622050
icmp:
mbps: 1.98
pps: 36764
attack_type: udp_flood

- 'UDP flows: 76'
- .1:29126 < 193.15.55.5:123 256392 bytes 2374 packets
- .1:29126 < 83.234.89.5:123 3975192 bytes 9998 packets
- .1:29126 < 37.59.44.6:123 5301036 bytes 11327 packets
- .1:29126 < 46.105.191.10:123 3513420 bytes 7591 packets
- .1:29126 < 217.198.8.13:123 3734928 bytes 9786 packets
- .1:29126 < 109.201.64.18:123 116312976 bytes 248532 packets
- .1:29126 < 5.135.125.18:123 147996 bytes 329 packets
- .1:29126 < 94.190.64.21:123 445680 bytes 2476 packets
- .1:29126 < 185.70.135.26:123 116703756 bytes 249367 packets
- .1:29126 < 95.188.64.32:123 4507272 bytes 10014 packets
- .1:29126 < 198.27.65.33:123 149256 bytes 336 packets
- .1:29126 < 46.105.49.36:123 1235196 bytes 2685 packets
- .1:29126 < 198.91.51.37:123 5482692 bytes 12473 packets
- .1:29126 < 37.187.90.41:123 308880 bytes 660 packets
- .1:29126 < 77.37.160.44:123 452520 bytes 2514 packets
- .1:29126 < 94.23.217.46:123 2492100 bytes 5325 packets
- .1:29126 < 176.31.34.50:123 64620 bytes 199 packets
- .1:29126 < 86.59.35.50:123 7455996 bytes 17435 packets
- .1:29126 < 46.105.74.56:123 235512 bytes 512 packets
- .1:29126 < 85.118.60.57:123 76747788 bytes 163991 packets
- .1:29126 < 5.39.65.67:123 202032 bytes 508 packets
- .1:29126 < 92.63.149.74:123 267840 bytes 2480 packets
- .1:29126 < 149.202.30.78:123 410472 bytes 1084 packets
- .1:29126 < 194.177.31.81:123 619668 bytes 2459 packets
- .1:29126 < 178.47.130.82:123 4165956 bytes 10055 packets
- .1:29126 < 89.203.137.86:123 7477560 bytes 16216 packets
- .1:29126 < 185.50.214.100:123 265896 bytes 2462 packets
- .1:29126 < 208.92.219.100:123 999504 bytes 2524 packets
- .1:29126 < 208.92.219.103:123 3187368 bytes 7582 packets
- .1:29126 < 212.232.61.120:123 2802780 bytes 7523 packets
- .1:29126 < 74.118.17.124:123 112431852 bytes 240239 packets
- .1:29126 < 213.155.226.132:123 1996740 bytes 5037 packets
- .1:29126 < 142.54.178.138:123 1797876 bytes 4983 packets
- .1:29126 < 176.31.104.142:123 190296 bytes 502 packets
- .1:29126 < 5.39.70.150:123 4489992 bytes 9594 packets

Work in progress

all the ingredients on this recipe are open source

About Net Healer: experimental Ruby code
ideas, issues and pull requests are **more** than **welcome**

Join FastNetMon mail list
- https://groups.google.com/forum/#!forum/fastnetmon

Contribute at Net Healer github
- https://github.com/zenvdeluca/net_healer

zendesk

zendesk

Thank you !

vdeluca@zendesk.com