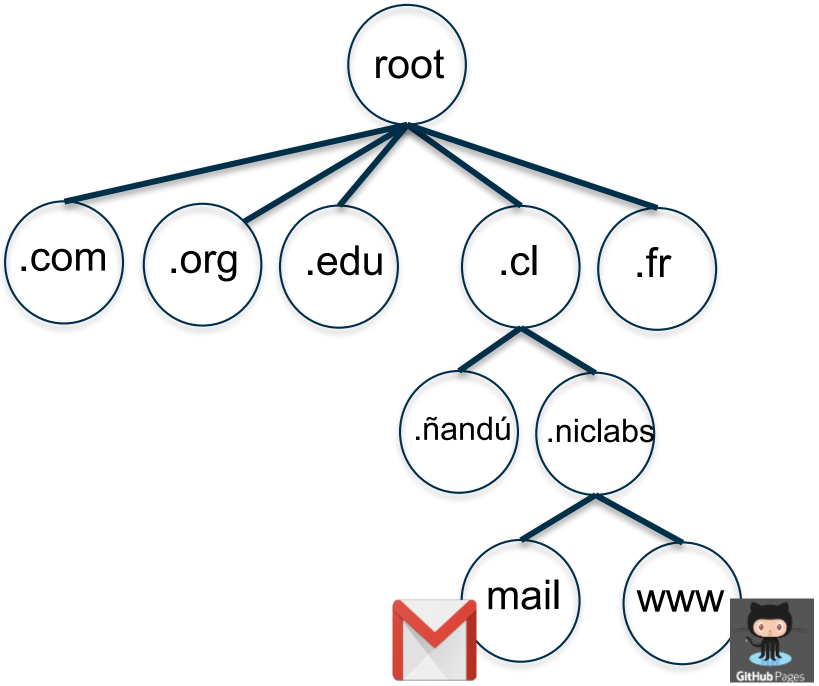
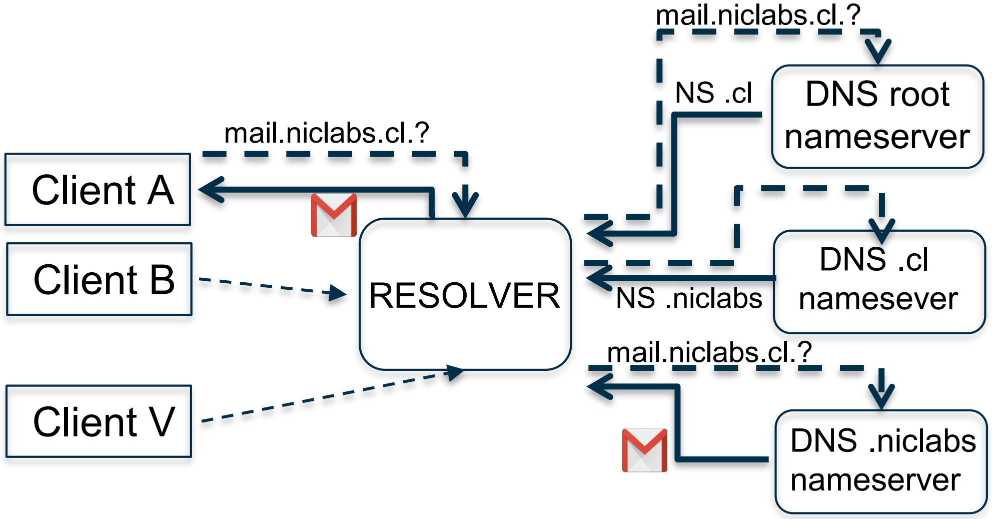
Monitoring DNS with Open-Source Solutions

Felipe Espinoza - Javier Bustos
NIC Chile Research Labs

Domain Name System (DNS)

- Decentralized naming system for resources.
- Hierachical.





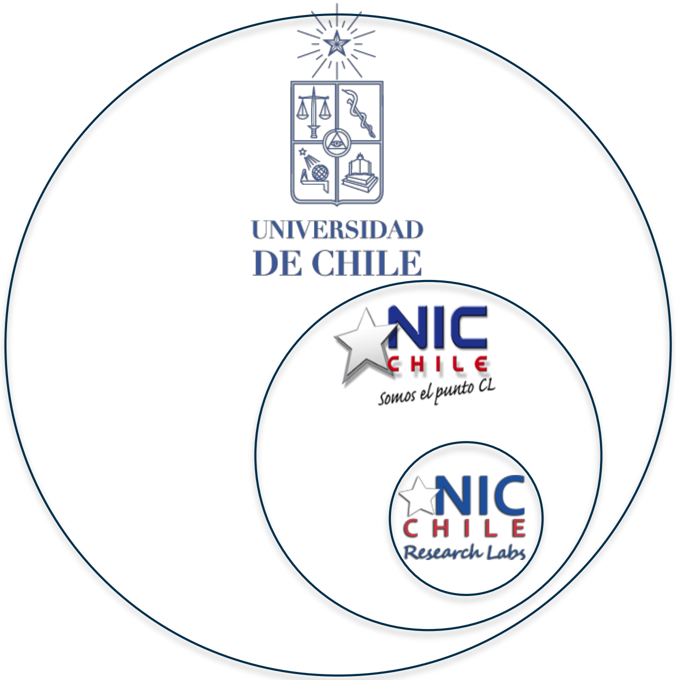
Some resource records (RR):

A: IPv4
AAAA: IPv6
NS: NameServer
CNAME: Alias
MX: Mail eXchange record

Some return codes:

NOERROR: all ok
SERVFAIL: server failed to complete the DNS request
NXDOMAIN: domain does not exists

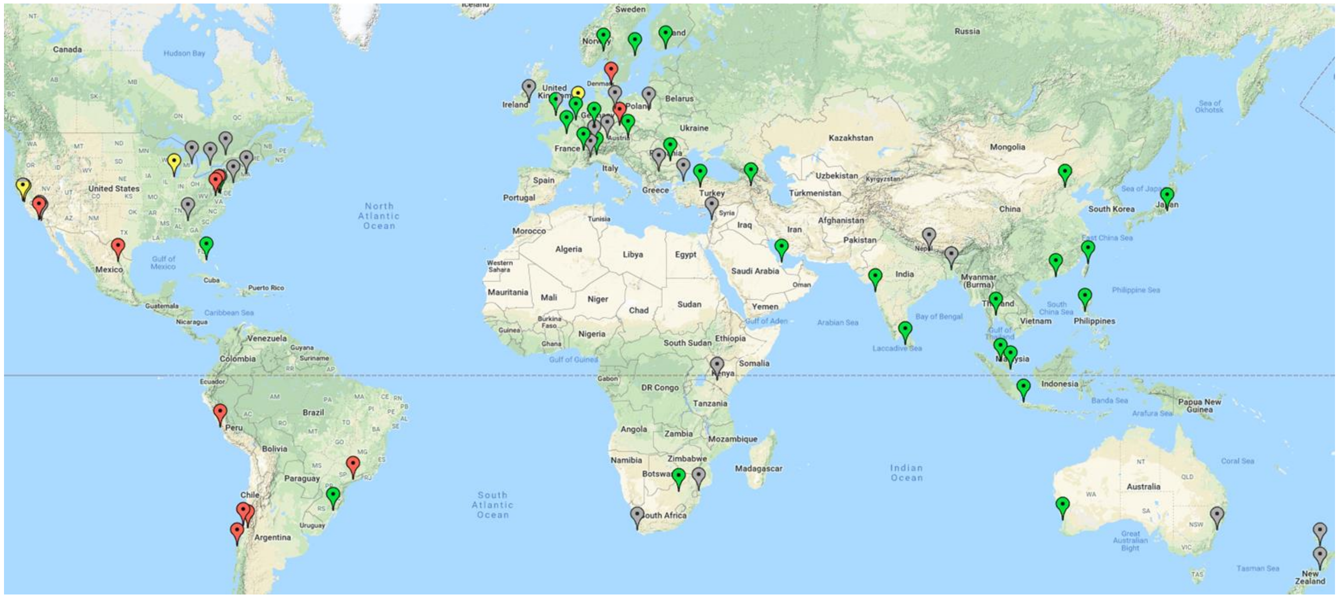
Context: .cl


UNIVERSIDAD
DE CHILE
NIC CHILE
Somos el punto CL
NIC CHILE
Research Labs

Context: NIC Chile operations

- Administrator of the “.cl” ccTLD.
- More than 550,000 registered domains.
- 26+ nodes directly managed on 10+ countries.

- Two external DNS clouds
  - Netnod
  - Packet Clearing House (PCH)



Context: why is DNS monitoring interesting?

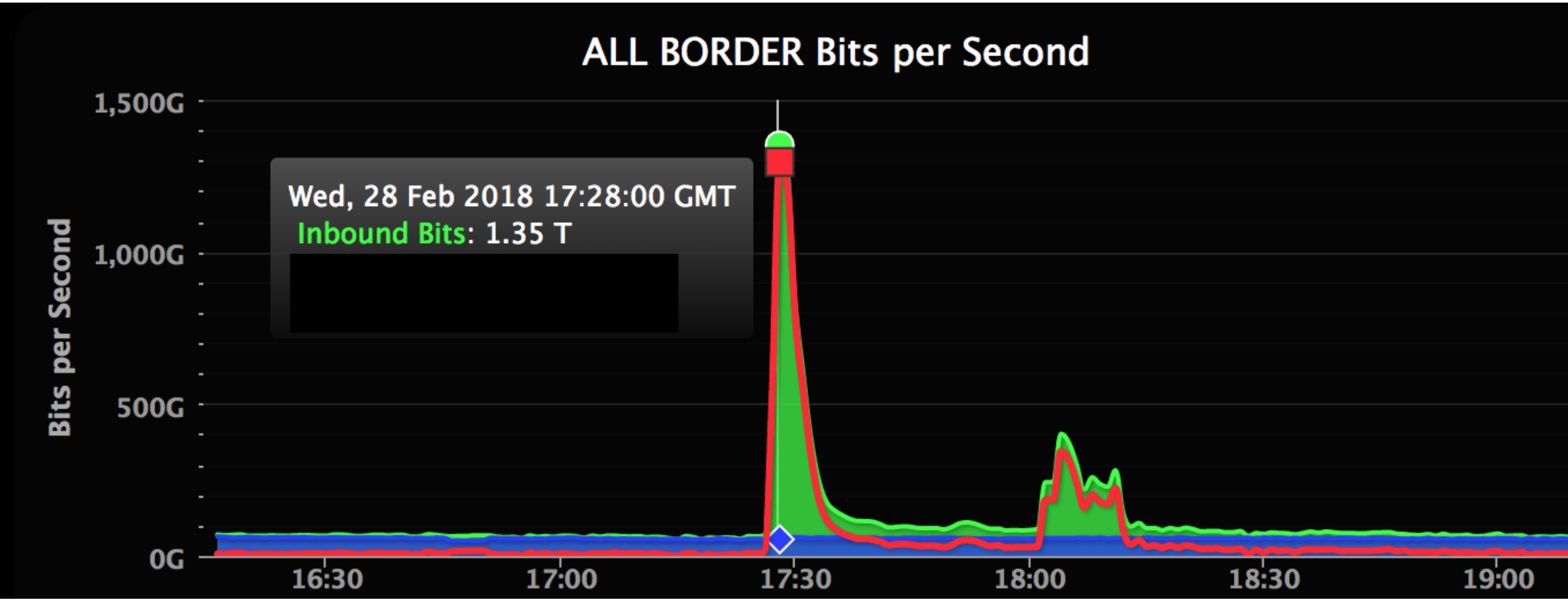
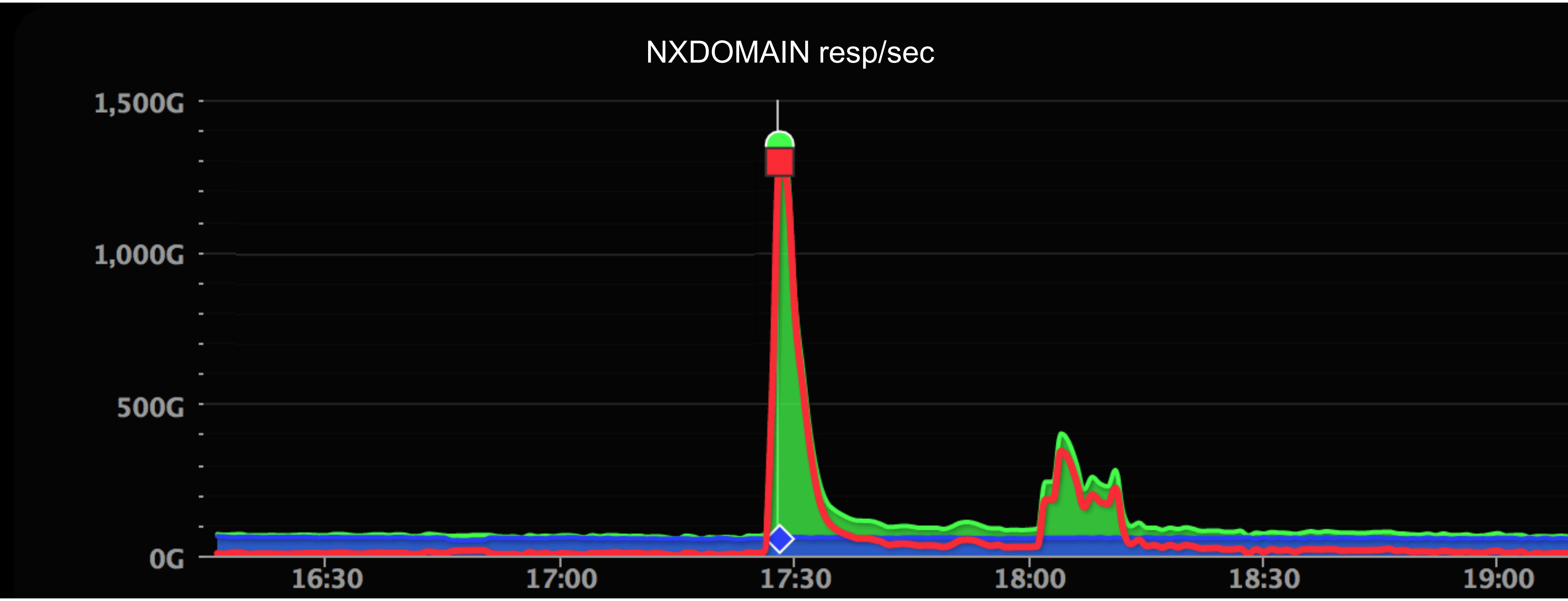


Image from Github

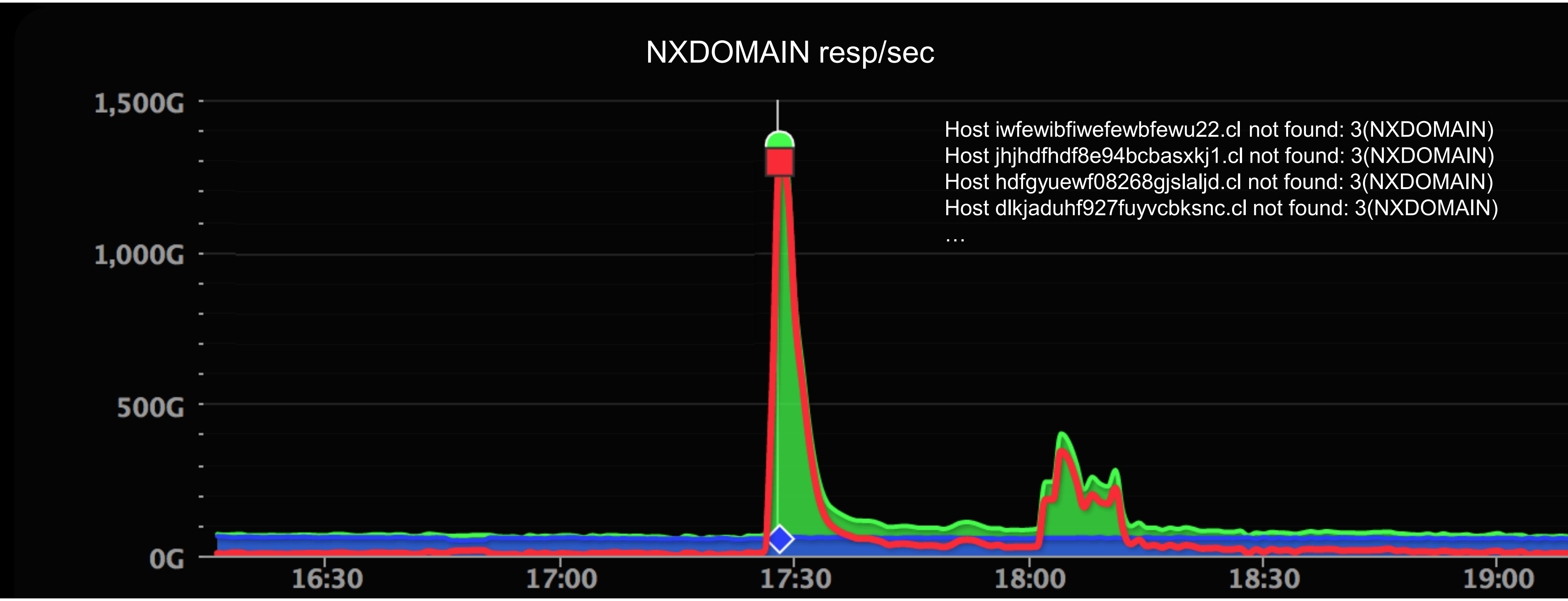
Context: why is DNS monitoring interesting?



NXDOMAIN resp/sec

Image from Github

Context: why is DNS monitoring interesting?



```
Host iwfewibfiwefewbfewu22.cl not found: 3(NXDOMAIN)
Host jhjhdfhdf8e94bcbasxkj1.cl not found: 3(NXDOMAIN)
Host hdfgyuewf08268gjslaljd.cl not found: 3(NXDOMAIN)
Host dlkjaduhf927fuyvcbksnc.cl not found: 3(NXDOMAIN)
…
```

Image from Github

Context: why is DNS monitoring interesting?

- 2016: Dyn DNS attack.
  - More than 1,200 affected domains.
  - Peak of 1.2 Tb/s.
  - 2 hours between detection and resolution for every event.

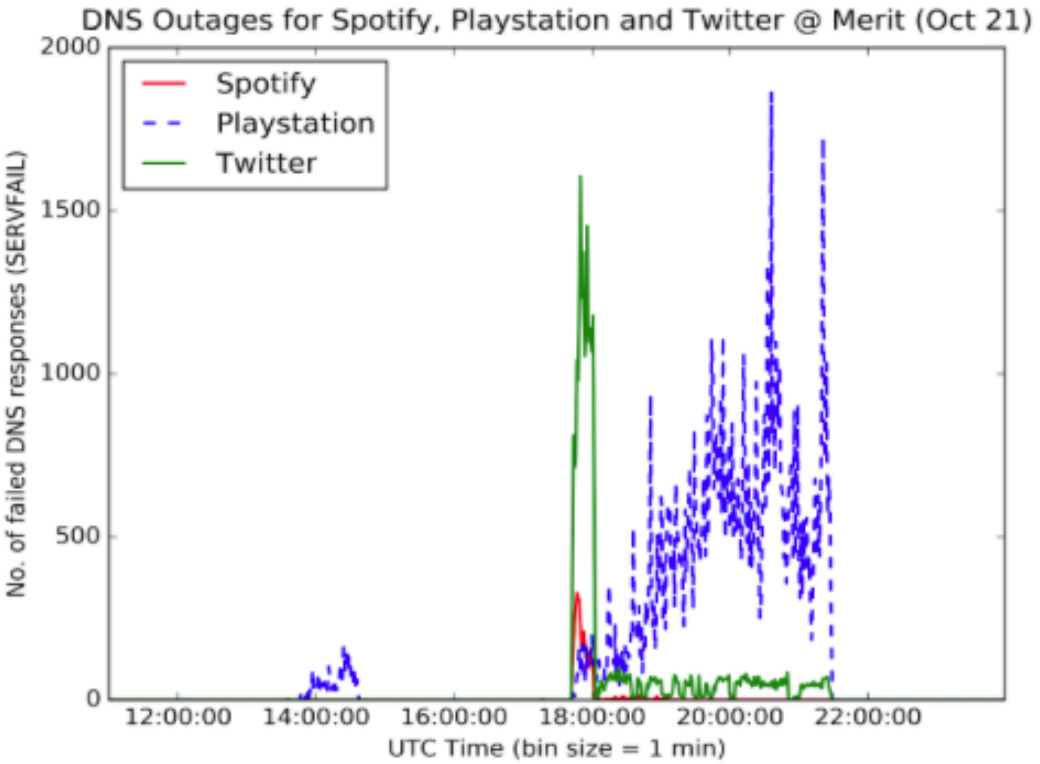
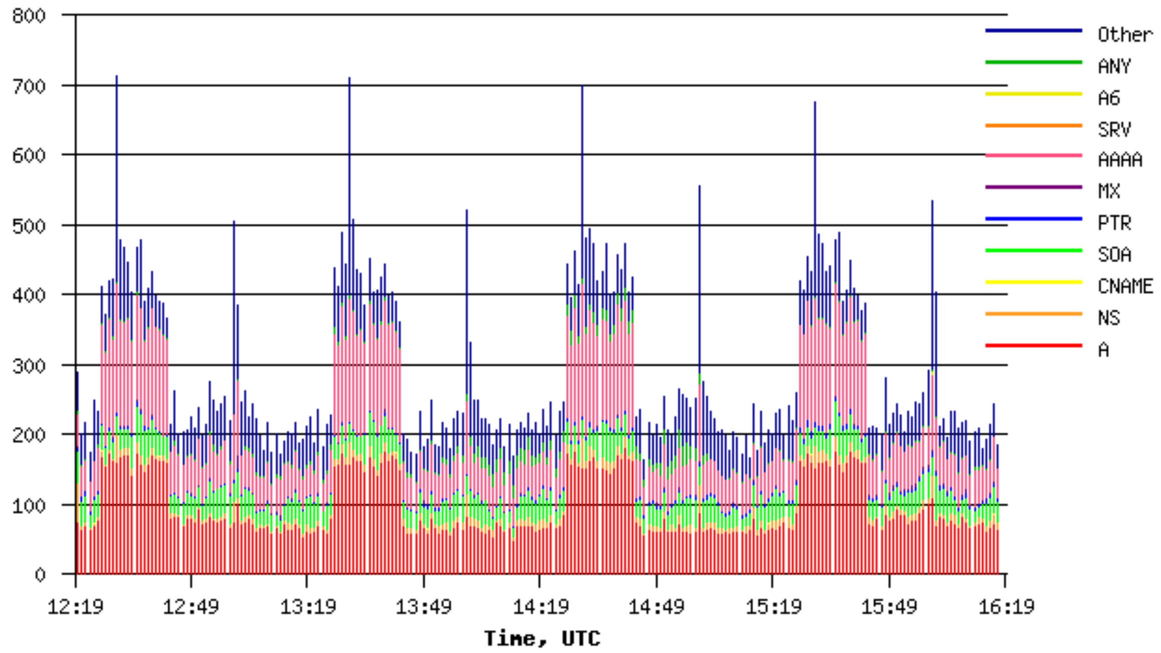


Image from Merit.edu

How is DNS Monitored?

- DNS Statistics Collector (DSC)
  - Pre-Aggregated Data
    - QTYPE
    - OPCODE
    - RCODE
    - …
  - Pos-Aggregation
    - Stats by server
- DNS-STATS
- ENTRADA
  - Transfer pcap files
  - Hadoop Cluster for processing

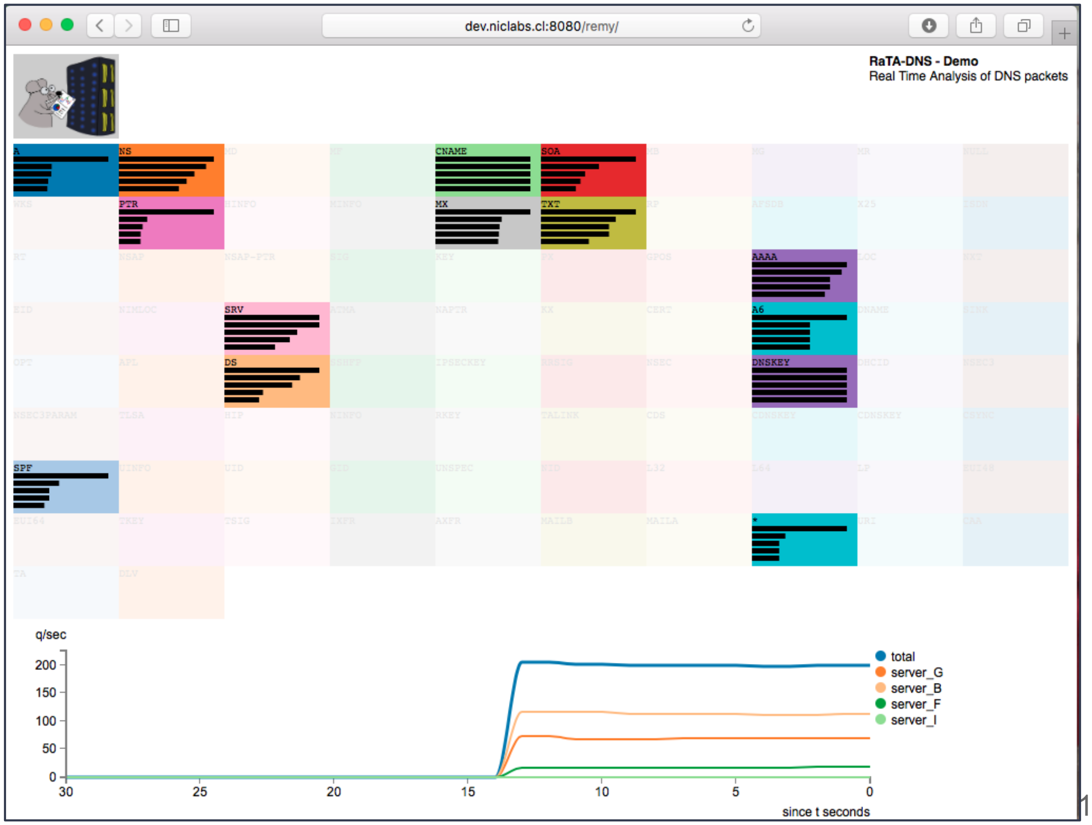


First Try: Develop our own solution

We developed RaTA DNS (Real Time Analysis of DNS packets)

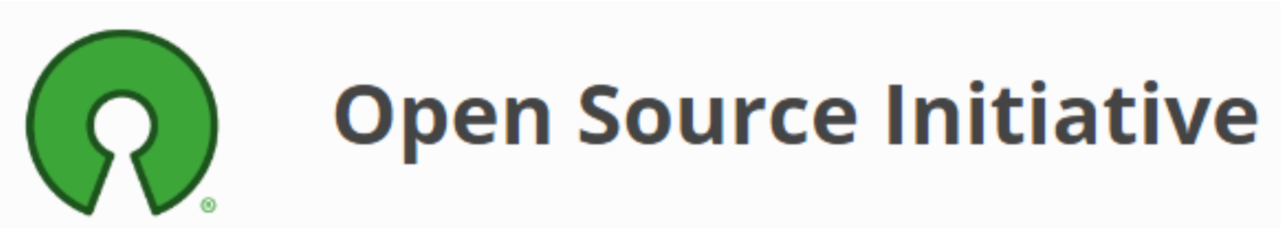
- Capture and reduce information.
- Transfer results over REDIS Queue.
- Show the information on our own presenter.

Were we reinventing the wheel?



Second Try: Use Open Source Software

- Instead of developing everything, integrate different open source software.
- Many parts of a monitoring system have already been developed.
- Many of them are used in production.

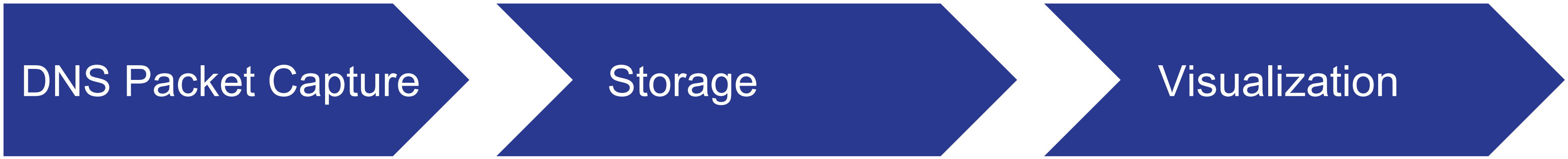


What we wanted to measure?

- Packet Metadata
  - Datetime
  - Server Name
  - IP Version
  - IP Prefix
  - Network Protocol
  - Size

- DNS Query/Responce
  - QR
  - OpCode
  - Class
  - Type
  - Edns0
  - DoBit
  - ResponceCode
  - Question

Requirements



DNS Packet Capture

Storage

Visualization

Requirements

DNS Packet Capture

- Secure
- Fast
- Low Cost

Storage

- Unitary
- Compressed
- Fast to process
- Big Volume of Information
- Scalable

Visualization

- Fast Access
- Relevant Information
- Alert Abnormalities

Software to analyze

Capture

- PacketBeat
- Collectd
- Fievel
- DSC
- gopassivedns

Storage

- Prometheus
- Druid
- ClickHouse
- InfluxDB
- ElasticSearch
- OpenTSDB

Visualization

- Kibana
- Grafana
- Graphite

Packet Capture

	IPv4	IPv4 Fragmented	IPv6	IPv6 Fragmented	UDP	TCP	Disaggregated Information
Fievel	✓		✓		✓		✓
Packetbeat	✓		✓		✓	✓	✓
collectd	✓		✓		✓		
dsc	✓	✓	✓	✓	✓	✓	
gopassivedns	✓				✓	✓	✓

Packet Capture

- DnsZeppelin: DNS Packet capturer.
  - Based on PacketBeat and gopassivedns.
  - Fragmented IP Assembly.
  - TCP Assembly.
  - Direct connection to database system.

- Source code: https://github.com/niclabs/dnszeppelin

Software to analyze

Capture

- PacketBeat
- Collectd
- Fievel
- DSC
- gopassivedns
- DnsZeppelin ✓

Storage

- Prometheus
- Druid
- ClickHouse
- InfluxDB
- ElasticSearch
- OpenTSDB

Visualization

- Kibana
- Grafana
- Graphite

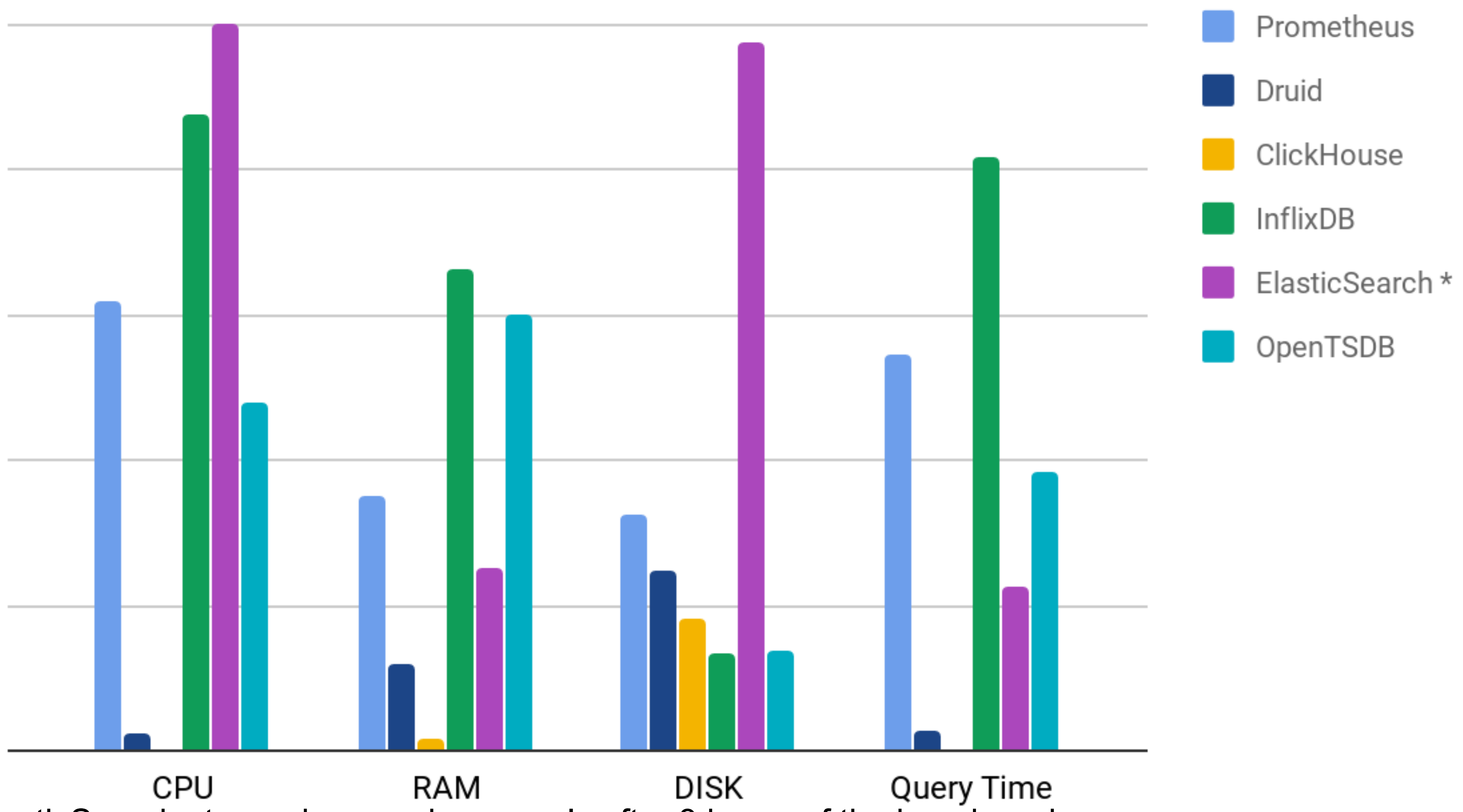
Benchmark

- CPU Usage
- Primary Memory
- Secondary Memory
- Query Time

- CPU: Intel(R) Core(TM) i5-4200U.
- Cores: 2.
- Threads: 2.
- Primary Memory: 8GiB DDR3 1600.
- Operating System: Ubuntu 14.04 LTS.
- Architecture: x64

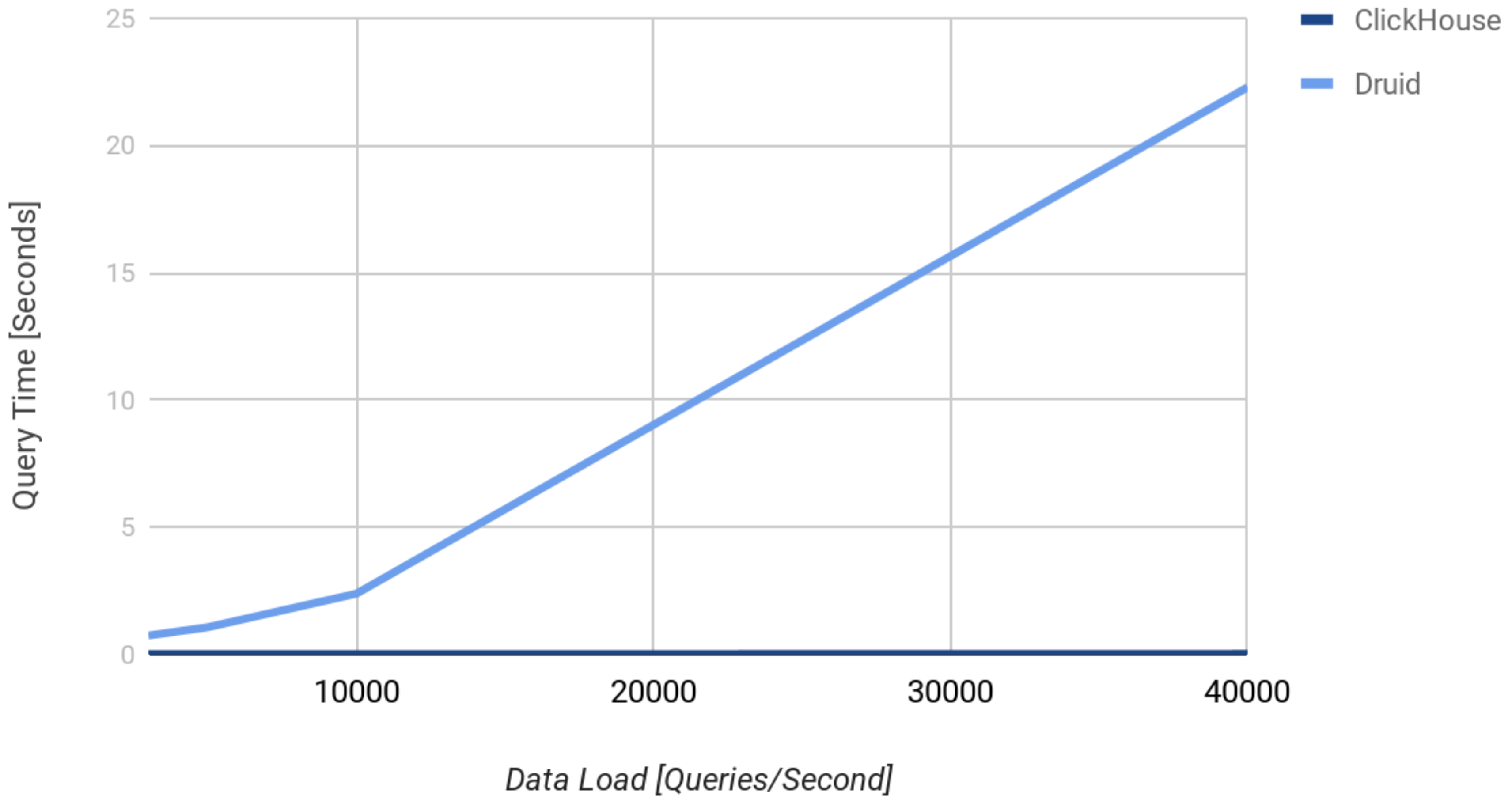
- Testing rate: 3,000 Packets/Second.

Normalised Benchmark Results



* ElasticSearch stopped answering query's after 3 hours of the benchmark.

Average Query Time



Software to analyze

Capture

- PacketBeat
- Collectd
- Fievel
- DSC
- gopassivedns
- DnsZeppelin ✓

Storage

- Prometheus
- Druid
- ClickHouse ✓
- InfluxDB
- ElasticSearch
- OpenTSDB

Visualization

- Kibana
- Grafana
- Graphite

Visualization

	Prometheus	Druid	ClickHouse	InfluxDB	ElasticSearch	OpenTSDB
Kibana	✓				✓	
Grafana	✓	✓	✓	✓	✓	✓
Graphite				✓		✓

Software to analyze

Capture

- PacketBeat
- Collectd
- Fievel
- DSC
- gopassivedns
- DnsZeppelin ✓

Storage

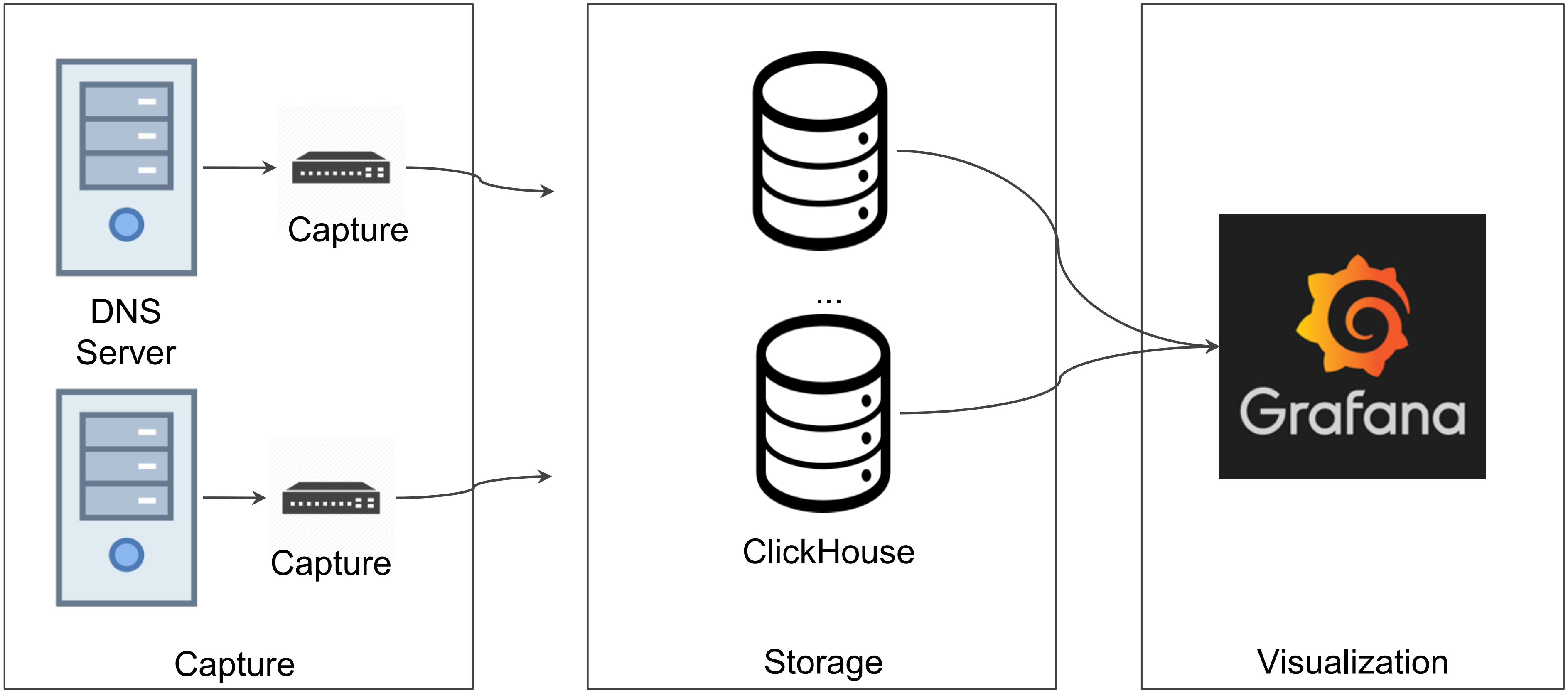
- Prometheus
- Druid
- ClickHouse ✓
- InfluxDB
- ElasticSearch
- OpenTSDB

Visualization

- Kibana
- Grafana ✓
- Graphite

Resulted System

Architecture



Load Simulation

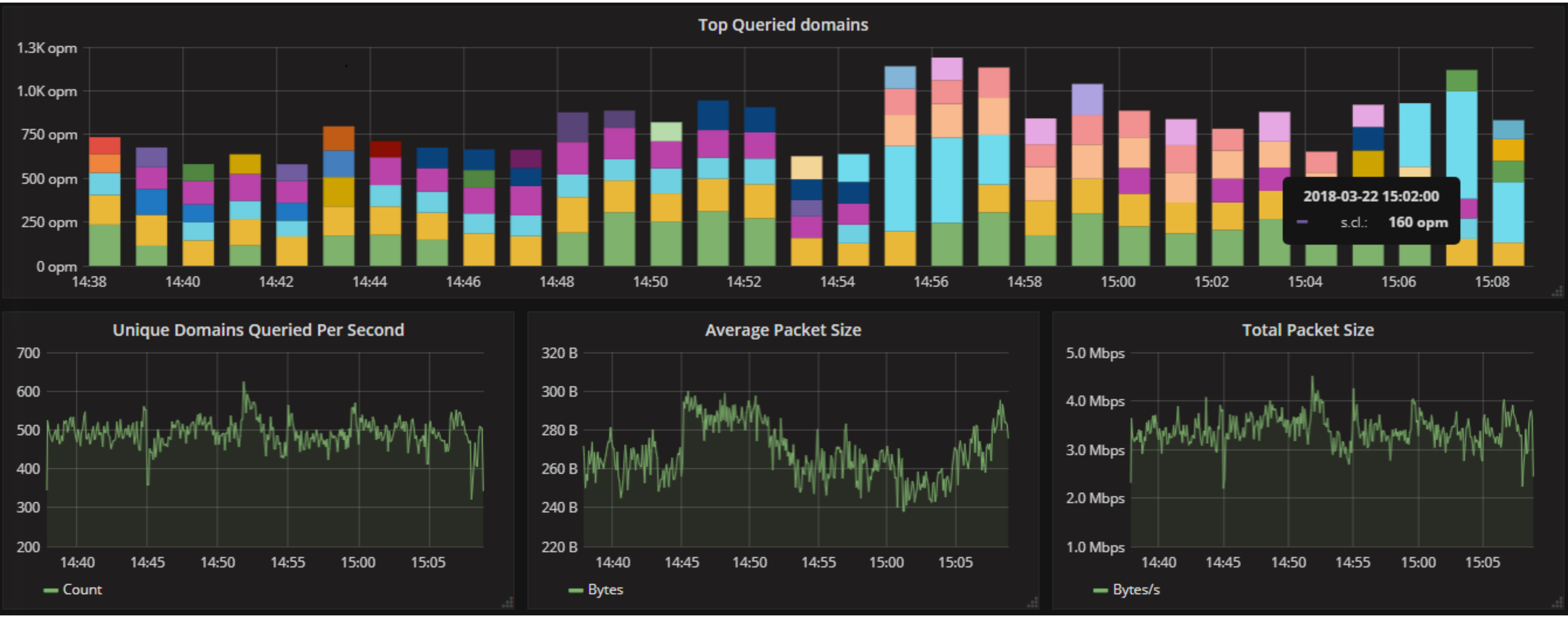
- Normal Simulation:
    - Packets/Second: ~7,000 pps
    - Time running: 36 Hours
    - Total packet count: ~927,000,000
    - Total uncompressed data: 52 GB
    - Total compressed data: 7.1 GB
    - Compressed packet size: ~8.3 Bytes

Load Simulation

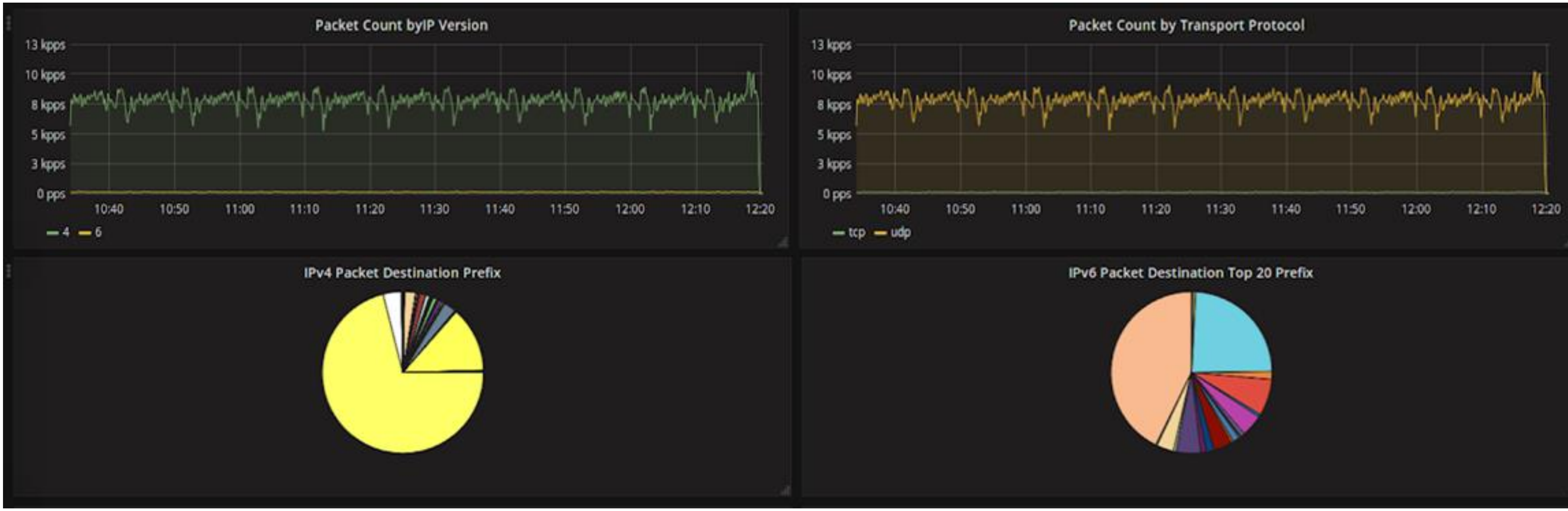
- Normal Simulation:
  - Packets/Second: ~7,000 qps
  - Time running: 36 Hours
  - Total packet count: ~927,000,000
  - Total uncompressed data: 52 GB
  - Total compressed data: 7.1 GB
  - Compressed packet size: ~8.3 Bytes

- Flood Simulation:
  - Packets/Second: 120,000 qps
  - Average CPU Usage: 30%

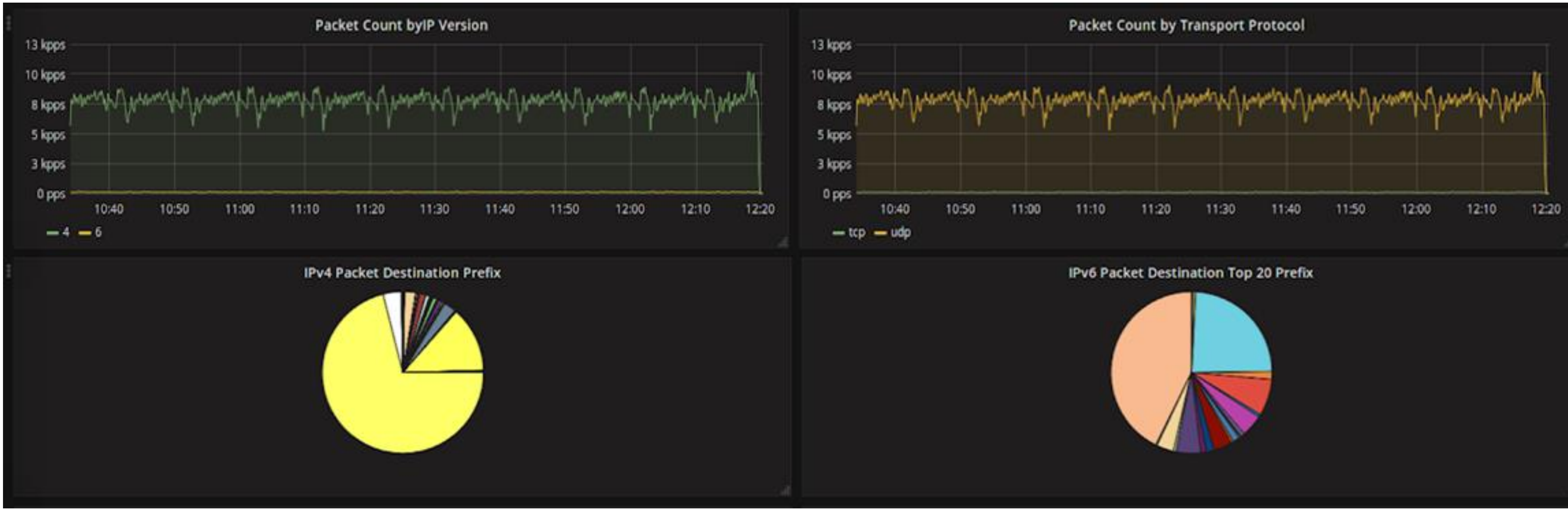
Grafana Panel



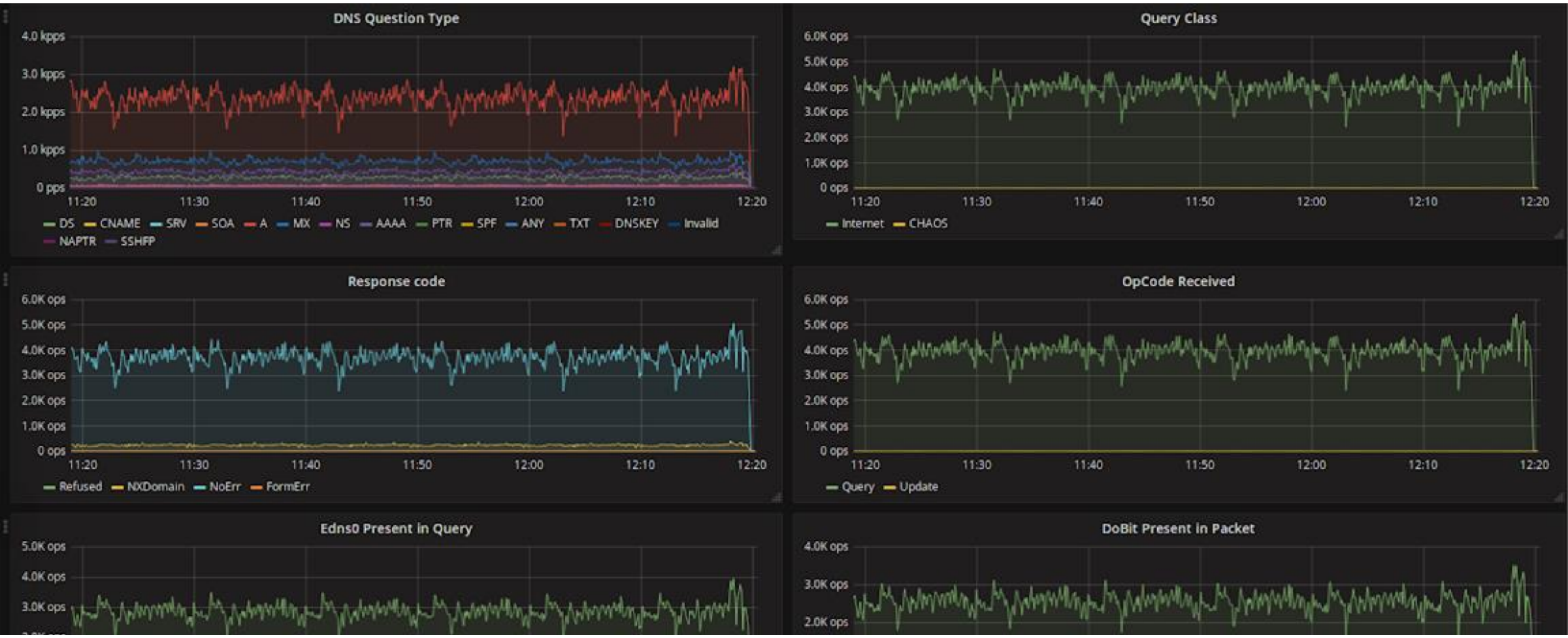
Grafana Panel



Grafana Panel

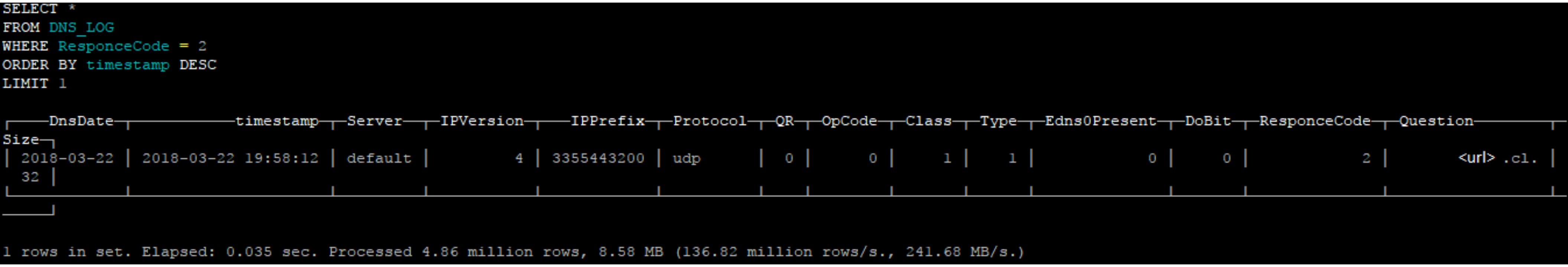


Grafana Panel



SQL Interface

- Query individual DNS packet.

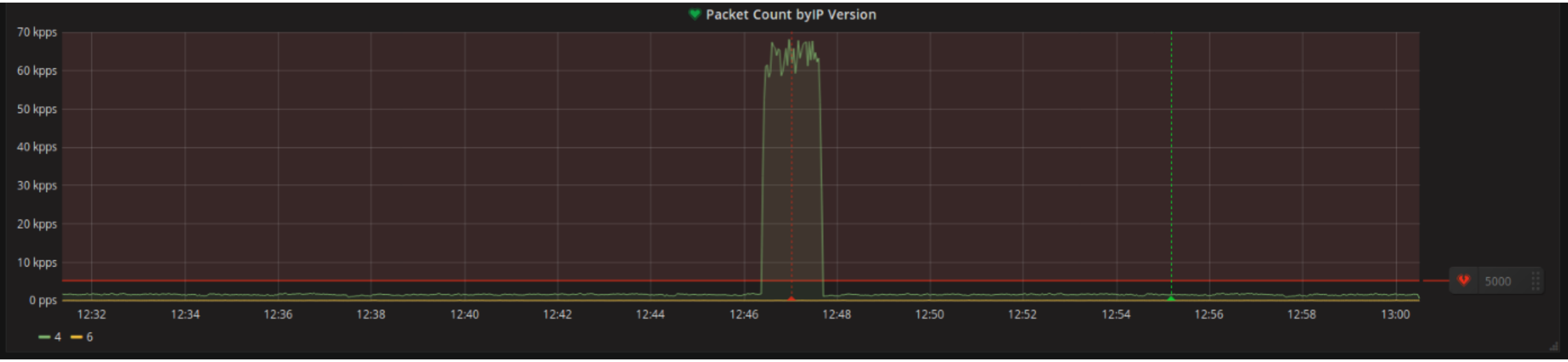


```
SELECT *
FROM DNS_LOG
WHERE ResponceCode = 2
ORDER BY timestamp DESC
LIMIT 1

┌────DnsDate─┬───────────timestamp─┬─Server──┬─IPVersion─┬────IPPrefix─┬─Protocol─┬─QR─┬─OpCode─┬─Class─┬─Type─┬─Edns0Present─┬─DoBit─┬─ResponceCode─┬─Question──────┬
Size─┐
│ 2018-03-22 │ 2018-03-22 19:58:12 │ default │         4 │ 3355443200 │ udp      │  0 │      0 │     1 │    1 │            0 │     0 │            2 │     <url> .cl. │
  32 │
└────────────┴─────────────────────┴─────────┴───────────┴────────────┴──────────┴────┴────────┴───────┴──────┴──────────────┴───────┴──────────────┴───────────────┴
─────┘

1 rows in set. Elapsed: 0.035 sec. Processed 4.86 million rows, 8.58 MB (136.82 million rows/s., 241.68 MB/s.)
```

- Show last ServFail

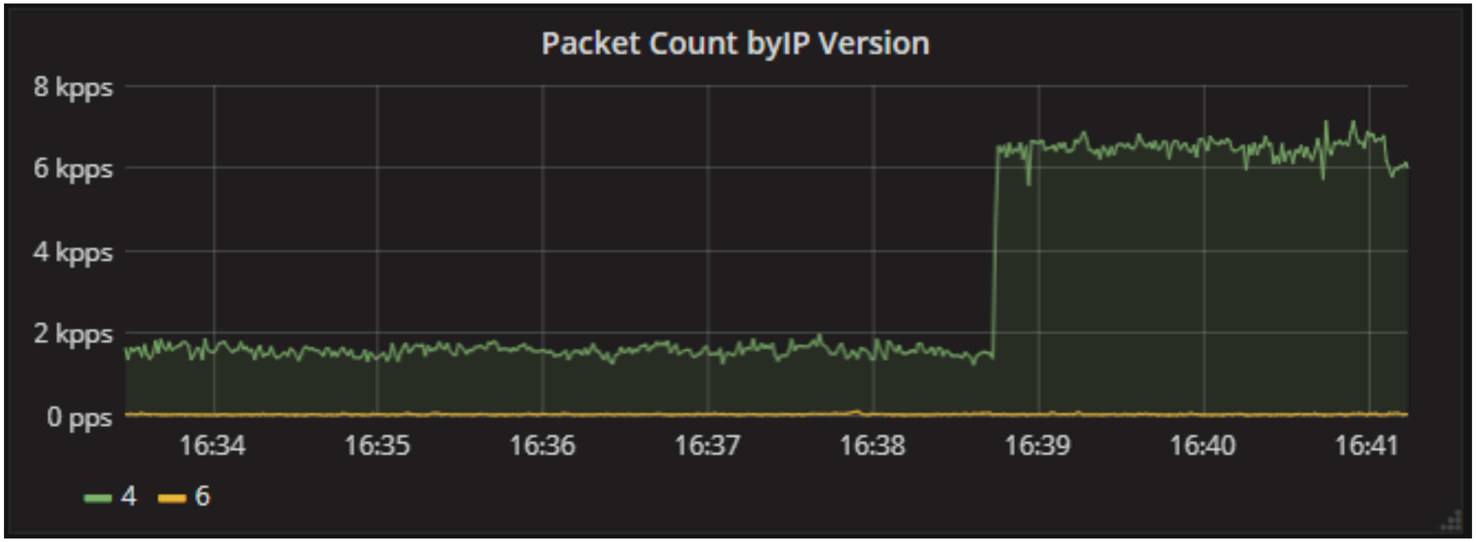
Alerting

- Grafana Alerting
  - Define thresholds.
  - Send messages on start/end of events.

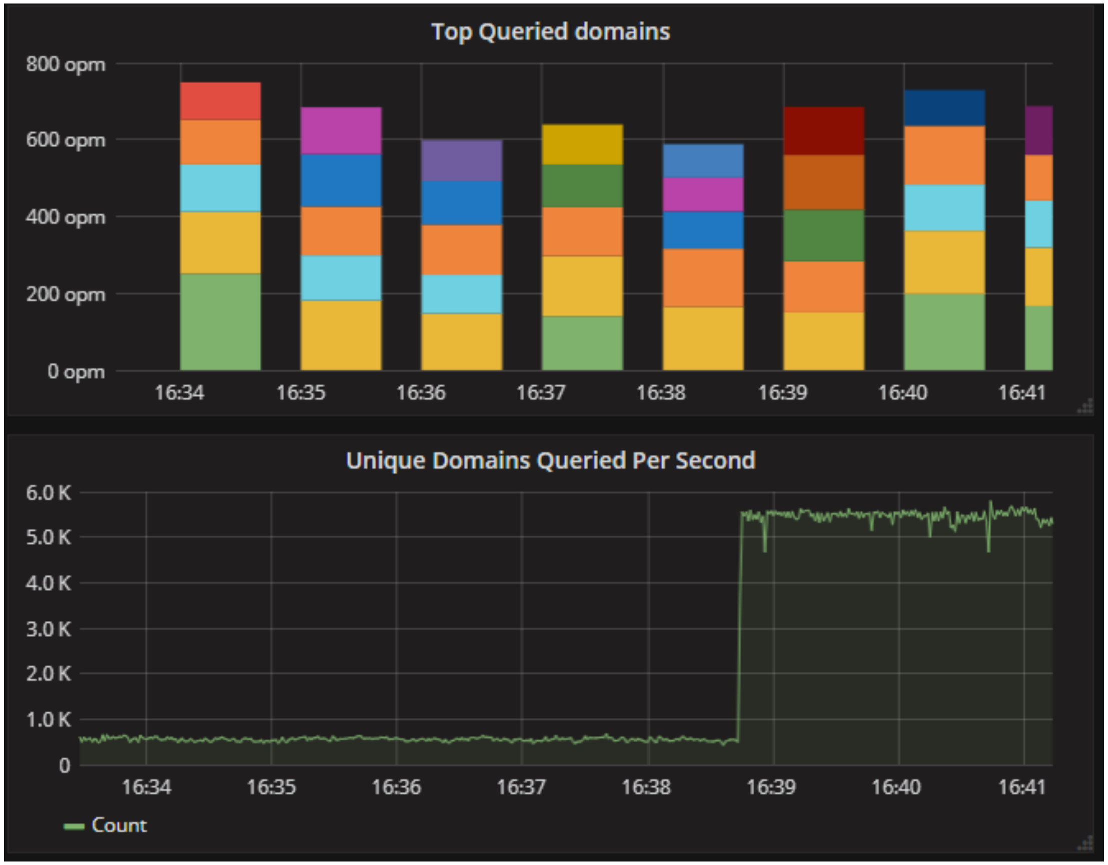


Attack Example

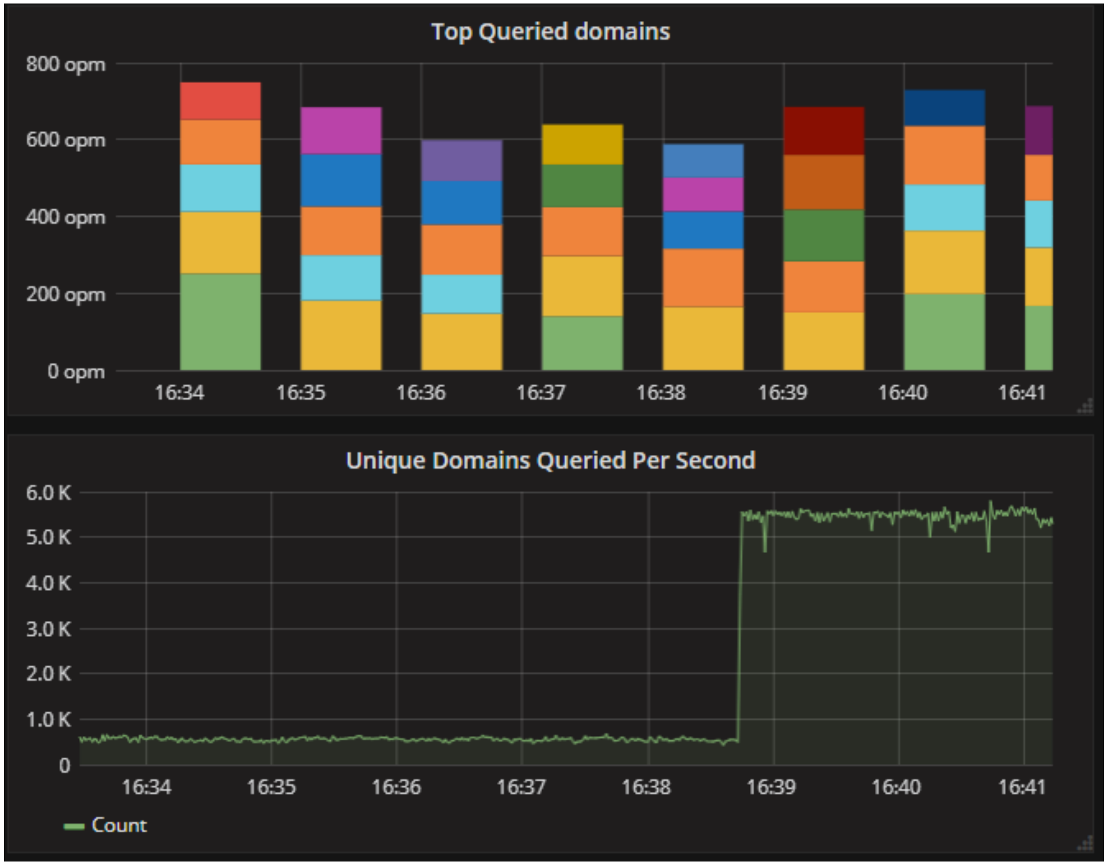
- Typical DNS packet flood.
- What type of attack is it?



Attack Example

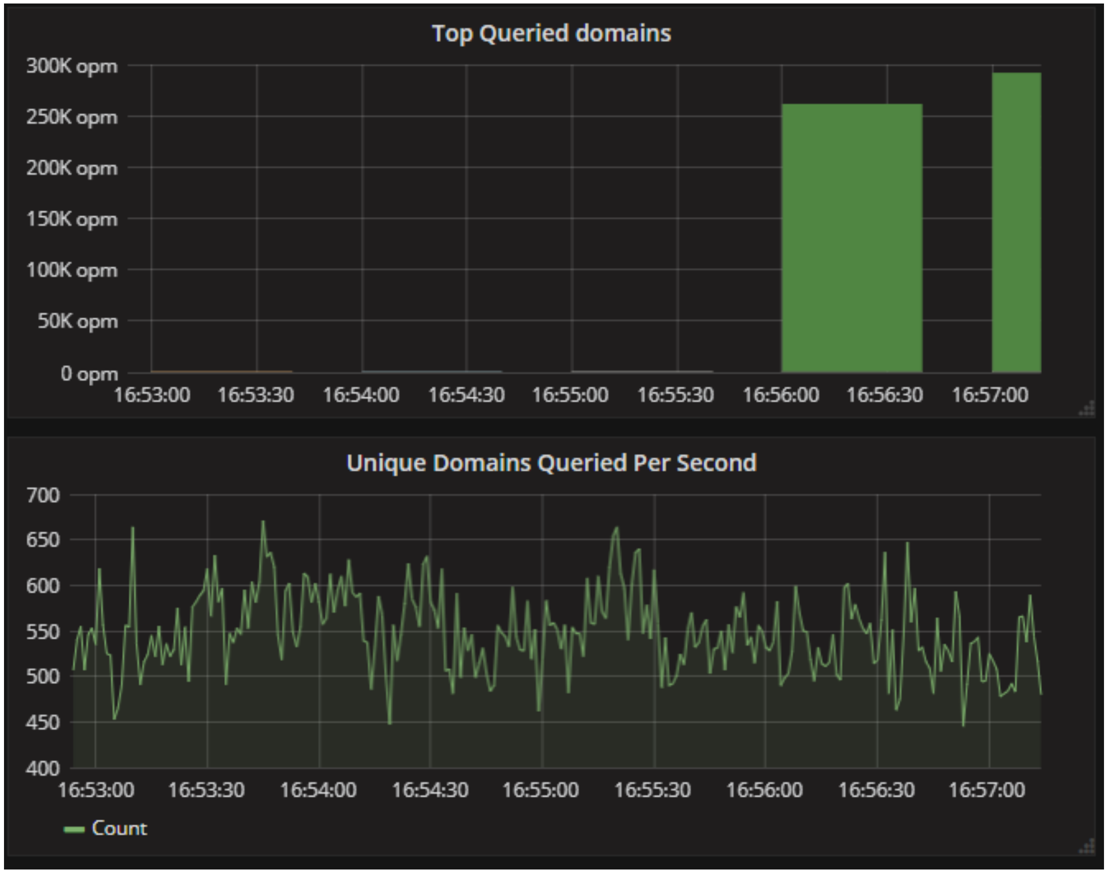


Attack Example

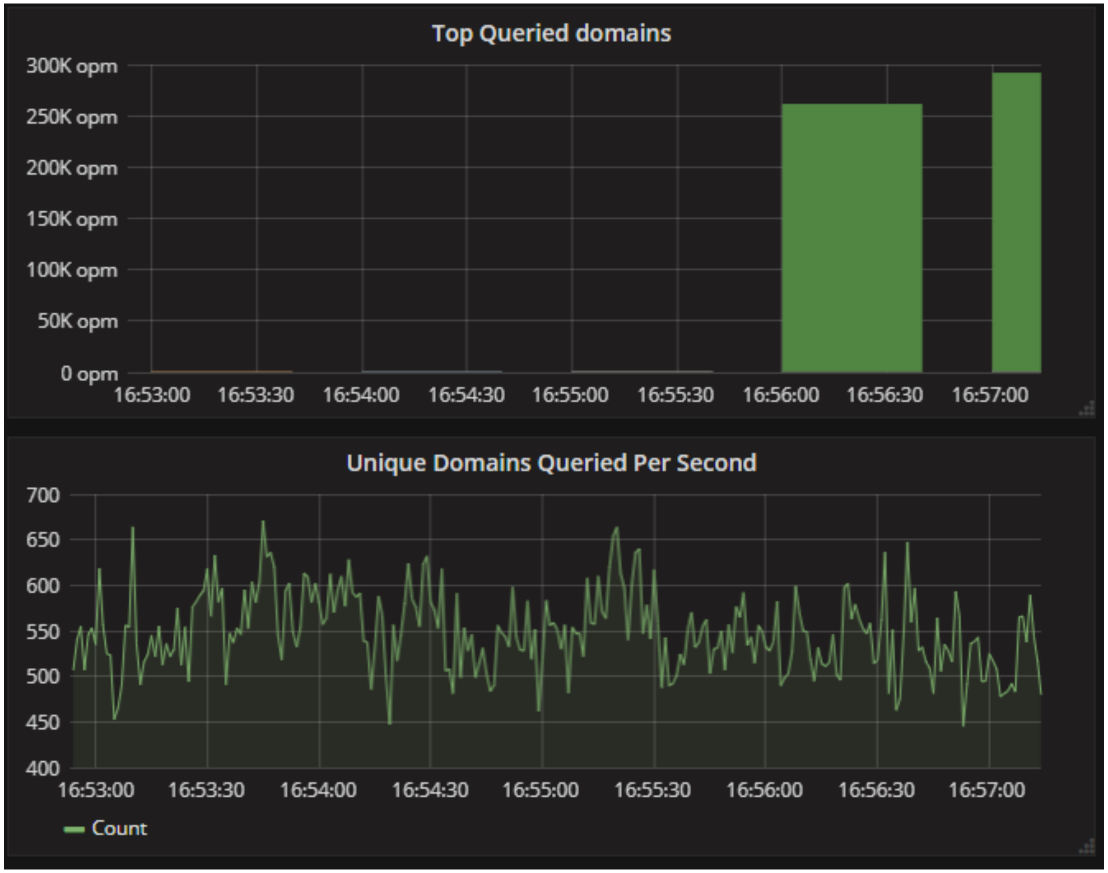


- <randomstring>.cl
- ISP don’t have query cached.
- Random DNS Query Attack.

Attack Example



Attack Example



- example.cl
- ISP have query cached.
- Packets are easier to craft.

Limitations

- Currently it’s not handling all the data in the DNS packet.
- Require small modifications to use the distributed capabilities of ClickHouse.
- The alert system is too simple.

Conclusion

- Working DNS Monitoring Solution
  - DnsZeppelin
  - ClickHouse
  - Grafana
- Make our monitoring more intelligent.
- Use open source software.

Questions?

Source code:
https://github.com/niclabs/dnszeppelin-clickhouse

Felipe Espinoza - fdns@niclabs.cl
Javier Bustos - jbustos@niclabs.cl