Monitoring DNS with Open-Source Solutions

Felipe Espinoza - Javier Bustos-Jiménez
NIC Chile Research Labs
Context: NIC Chile (.cl ccTLD) 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?

- 2016: Dyn DNS attack.
  - More than 1,200 affected domains.
  - Peak of 1.2 Tbps.
  - 2 hours between detection and resolution.

Image from Merit.edu
Context: why is DNS monitoring interesting?

Talks Track 1

Grand Ballroom ABC

**SparkPost: The Day the DNS Died**
Jeremy Blosser, SparkPost
Show details

**Stable and Accurate Health-Checking of Horizontally-Scaled Services**
Lorenzo Saino, Fastly
Show details

Wednesday, 10:55 am–11:35 am

Wednesday, 11:40 am–12:20 pm
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?

Fun fact: dnsadmins didn’t liked it because the visual interface was too much white and clean.
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/Response
  ○ QR
  ○ OpCode
  ○ Class
  ○ Type
  ○ Edns0
  ○ DoBit
  ○ ResponseCode
  ○ Question
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

<table>
<thead>
<tr>
<th></th>
<th>IPv4</th>
<th>IPv4 Fragmented</th>
<th>IPv6</th>
<th>IPv6 Fragmented</th>
<th>UDP</th>
<th>TCP</th>
<th>Disaggregated Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fievel</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>Packetbeat</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>collectd</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsc</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>gopassivedns</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
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](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.
ElasticSearch stopped answering query's after 3 hours of the benchmark.
Software to analyze

Capture
- PacketBeat
- Collectd
- Fievel
- DSC
- gopassivedns
- DnsZeppelin ✔

Storage
- Prometheus
- Druid
- ClickHouse ✔
- InfluxDB
- ElasticSearch
- OpenTSDB

Visualization
- Kibana
- Grafana
- Graphite
## Visualization

<table>
<thead>
<tr>
<th></th>
<th>Prometheus</th>
<th>Druid</th>
<th>ClickHouse</th>
<th>InfluxDB</th>
<th>ElasticSearch</th>
<th>OpenTSDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kibana</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
</tr>
<tr>
<td>Grafana</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Graphite</td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td></td>
<td>✔️</td>
</tr>
</tbody>
</table>
Software to analyze

Capture
- PacketBeat
- Collectd
- Fievel
- DSC
- gopassivedns
- DnsZeppelin ✔

Storage
- Prometheus
- Druid
- ClickHouse ✔
- InfluxDB
- ElasticSearch
- OpenTSDB

Visualization
- Kibana
- Grafana ✔
- Graphite
Resulted System
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
SQL Interface

- Query individual DNS packet.

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

- 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.
tl;dr

- 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