summarizing traffic for observability and DDoS mitigation

Shannon Weyrick • VP Research @ NS1 • Office of CTO
sweyrick@ns1.com
pktvisor.com • IDS 2021
1. pktvisor in 15 mins
2. Deeper Dive
3. The Future: Orb
pktvisor in 15 mins
What is pktvisor?

- Open Source observability Agent
- Taps into pcap and (soon) DNSTAP streams
- Summarizes critical data from streams
- Provides both Local and Global visibility
What is pktvisor not?

- A full packet capture system
- A query audit log system
- A database
- Resource heavy
Why pktvisor?

- Deep L7 analysis with streaming algorithms
- Not based on flow/sampling
- Small data, big information
pktvisor extracts signal and produces summaries

- “Signal” is critical Net and DNS info
- Summarizes into live + 1 minute buckets
- JSON output is ~4kb per bucket
- ...regardless of input throughput!
DNS signal extraction

L7 Dissection
- Top Queries
- Top Query Types
- Top Result Codes
- Rate Percentiles
- Top Sources
- Counters
...

1 minute summaries

raw DNS traffic

DNS packet on the wire

pktvisor.com ∙ IDS 2021

Shannon Weyrick ∙ sweyrick@ns1.com
DNS signal extraction

Low DNS Traffic

1 minute summaries

flow
DNS signal extraction

Information we may need to mitigate an attack!

Top Queries
Top Query Types
Top Result Codes
Rate Percentiles
Top Sources
Counters
...

1 minute summaries

flow

127x152 to 523x322

DDoS Traffic

 pktvisor.com ∙ IDS 2021

Shannon Weyrick ∙ sweyrick@ns1.com

pktvisor.com ∙ IDS 2021
Command Line UI (think “dns top”) Updates display once / sec
How many unique IPs have been seen in the time window?
How many unique Qnames have been seen in the time window?
<table>
<thead>
<tr>
<th>QName</th>
<th>Value</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>.com.google</td>
<td>48</td>
<td>11.5%</td>
</tr>
<tr>
<td>.com.apple</td>
<td>28</td>
<td>6.7%</td>
</tr>
<tr>
<td>.com.akadns</td>
<td>28</td>
<td>6.7%</td>
</tr>
<tr>
<td>.com.googleapis</td>
<td>24</td>
<td>5.9%</td>
</tr>
<tr>
<td>.in-addr.arpa</td>
<td>24</td>
<td>5.9%</td>
</tr>
<tr>
<td>.microsoft.com</td>
<td>12</td>
<td>2.9%</td>
</tr>
<tr>
<td>.office.com</td>
<td>12</td>
<td>2.9%</td>
</tr>
<tr>
<td>.akadns.net</td>
<td>20</td>
<td>4.9%</td>
</tr>
<tr>
<td>.in-addr.arpa</td>
<td>18</td>
<td>2.1%</td>
</tr>
<tr>
<td>.游戏.google.com</td>
<td>18</td>
<td>2.1%</td>
</tr>
<tr>
<td>.dns.google.com</td>
<td>18</td>
<td>2.1%</td>
</tr>
<tr>
<td>.googleapis.com</td>
<td>12</td>
<td>2.9%</td>
</tr>
<tr>
<td>.calendar.google.com</td>
<td>12</td>
<td>2.9%</td>
</tr>
<tr>
<td>.gapis.google.com</td>
<td>12</td>
<td>2.9%</td>
</tr>
<tr>
<td>.googleapis.com</td>
<td>12</td>
<td>2.9%</td>
</tr>
<tr>
<td>.calendar.google.com</td>
<td>12</td>
<td>2.9%</td>
</tr>
<tr>
<td>.gapis.google.com</td>
<td>12</td>
<td>2.9%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QTypes</th>
<th>Value</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>310</td>
<td>74.5%</td>
</tr>
<tr>
<td>HTTPS</td>
<td>52</td>
<td>12.5%</td>
</tr>
<tr>
<td>PTR</td>
<td>38</td>
<td>9.1%</td>
</tr>
<tr>
<td>AAAA</td>
<td>12</td>
<td>2.9%</td>
</tr>
<tr>
<td>SOA</td>
<td>4</td>
<td>1%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RCodes</th>
<th>Value</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOERROR</td>
<td>185</td>
<td>90.2%</td>
</tr>
<tr>
<td>NXDOMAIN</td>
<td>20</td>
<td>9.8%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REFUSED</th>
<th>Value</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 192.168.0.189</td>
<td>1175</td>
<td>67.9%</td>
</tr>
<tr>
<td>IPv4 192.168.0.114</td>
<td>118</td>
<td>6.8%</td>
</tr>
<tr>
<td>IPv4 35.190.20.61</td>
<td>108</td>
<td>6.2%</td>
</tr>
<tr>
<td>IPv4 91.188.88.185</td>
<td>20</td>
<td>1.1%</td>
</tr>
<tr>
<td>IPv4 239.255.255.250</td>
<td>16</td>
<td>0.9%</td>
</tr>
<tr>
<td>IPv4 216.239.32.10</td>
<td>12</td>
<td>0.7%</td>
</tr>
<tr>
<td>IPv4 35.224.179.84</td>
<td>10</td>
<td>0.6%</td>
</tr>
<tr>
<td>IPv6 ff02::1:2</td>
<td>3</td>
<td>0.2%</td>
</tr>
<tr>
<td>IPv6 ff02::f5</td>
<td>3</td>
<td>0.2%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GeoLoc</th>
<th>Value</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown</td>
<td>1318</td>
<td>76.2%</td>
</tr>
<tr>
<td>NA/United States</td>
<td>222</td>
<td>12.8%</td>
</tr>
<tr>
<td>NA/United States/CA/Mountain View</td>
<td>3</td>
<td>2%</td>
</tr>
<tr>
<td>EU/United Kingdom/ENG/London</td>
<td>26</td>
<td>1.4%</td>
</tr>
<tr>
<td>EU</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>NA/United States/VA</td>
<td>10</td>
<td>0.6%</td>
</tr>
<tr>
<td>NA/United States/NA/Redmond</td>
<td>6</td>
<td>0.3%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ASN</th>
<th>Value</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown</td>
<td>1320</td>
<td>76.3%</td>
</tr>
<tr>
<td>15169/GOOGLE</td>
<td>154</td>
<td>8.9%</td>
</tr>
<tr>
<td>21342/Akamai International B.V.</td>
<td>3</td>
<td>2%</td>
</tr>
<tr>
<td>8075/MICROSOFT-CORP-MSN-AS-BLOCK</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>41231/Canonical Group Limited</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>8068/MICROSOFT-CORP-MSN-AS-BLOCK</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>16595/AMAZON-02</td>
<td>28</td>
<td></td>
</tr>
</tbody>
</table>
Central Collection

- Provides a *Global* view of distributed Agents
- Metric database agnostic
- Tools for Prometheus and Elasticsearch
- Grafana Dashboard
Central Collection

- Database agnostic
- Scrape or Push
- Small Data
Grafana Dashboard: Prometheus

- Deep Inspection
- Avg Rate 99.5

- Q/Types
- Result Codes

- DNS Packets (in+out)
- DNS Protocols

- DNS Transactions
- DNS Errors

- Top DNS Names

<table>
<thead>
<tr>
<th>Name</th>
<th>Requests (sum)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.google.com</td>
<td>956</td>
</tr>
<tr>
<td>googleapis.com</td>
<td>352</td>
</tr>
<tr>
<td>gstatic.com</td>
<td>288</td>
</tr>
<tr>
<td>apple.com</td>
<td>284</td>
</tr>
</tbody>
</table>
### pktvisor

pktvisor summarizes network data streams in real time, enabling on-node and centralized data visibility and analysis.

**pktvisor.com**
- agent
- monitoring
- grafana
- prometheus
- observability
- packet-capture
- aot-first
- data-streams
- collector-agent
- datasketches
- stream-processors
- stream-summaries

#### Releases
- 3.2.0 (Latest)
  - 26 days ago

#### Contributors
- [GitHub user icon](https://github.com)
- [GitHub user icon](https://github.com)
- [GitHub user icon](https://github.com)
- [GitHub user icon](https://github.com)
- [GitHub user icon](https://github.com)
- [GitHub user icon](https://github.com)
- [GitHub user icon](https://github.com)
- [GitHub user icon](https://github.com)
- [GitHub user icon](https://github.com)
- [GitHub user icon](https://github.com)

#### Languages
- C++ 85.1%
- C 5.6%
- Python 5.6%
3.2.0
weyrick released this on Apr 16

New Features

- Introduce native Prometheus support into pktvisor with `--prometheus` flag, which will expose Prometheus compatible metrics at `/metrics` endpoint. Also see `--prom-instance`
- Add a new docker container for easily collecting and sending Prometheus compatible metrics, see docker hub
- Add a new Grafana dashboard for Prometheus, both to the repo and to Grafana dashboard community
- Begin building and distributing an AppImage (static Linux binary) which includes pktvisor, pktvisor-cli, and pktvisor-pcap
- Ability to daemonize pktvisor with the `-d` flag
- Ability to send pktvisor logs to either an output file `(--log-file)`, or to syslog `(--syslog)

Other Improvements

- CI and build improvements including better use of Conan and automatic dependency installation
- Improved documentation and READMEs

Bug Fixes

- #47 Fix live rates in pktvisor-cli

Assets 3

- pktvisor-x86_64-3.2.0.AppImage 8.96 MB
- Source code (zip)
Easy Install

pull the image

```
root@dnshost:~$ docker pull ns1labs/pktvisor
```

start the agent

```
root@dnshost:~$ docker run --net=host -d ns1labs/pktvisor pktvisord eth0
```

run the command line UI

```
root@dnshost:~$ docker run -it --rm --net=host ns1labs/pktvisor pktvisor-cli
```
download the binary, make executable

```
root@dnshost:~$ curl -L http://pktvisor.com/download -o pktvisor-x86_64.AppImage
root@dnshost:~$ chmod +x pktvisor-x86_64.AppImage
```

start the agent

```
root@dnshost:~$ sudo ./pktvisor-x86_64.AppImage pktvisord eth0
```

run the command line UI

```
root@dnshost:~$ ./pktvisor-x86_64.AppImage pktvisor-cli
```
Easily Plugin To Prometheus

pktvisor + centralized Prometheus collection

This container combines pktvisor with the Grafana Agent for collecting and sending metrics to Prometheus through remote write, including to cloud providers like Grafana Cloud.

There is a sample Grafana dashboard which provides a good starting point for visualizing pktvisor metrics. You can also find it online via the Grafana community dashboards, allowing you to import easily into any Grafana installation (ID 14221).

Example:

```bash
docker pull nilslabs/pktvisor-prom-write
docker run -d --net=host --env PKTVISOR_ARGS="--prom-instance <INSTANCE> <INTERFACE>" \ 
--env REMOTE_URL="https://<REMOTEHOST>/api/prom/push" --env USERNAME="<USERNAME>" \ 
--env PASSWORD="<PASSWORD>" nilslabs/pktvisor-prom-write
```

Example with Geo enabled (assuming files are located in /usr/local/geo):

```bash
docker pull nilslabs/pktvisor-prom-write
docker run -d --net=host --mount type=bind,source=/usr/local/geo,target=/geo --name=host --env \ 
PKTVISOR_ARGS="--prom-instance <INSTANCE> <INTERFACE> <geo-city>/geo GeoPZ2-City.mdb <geo-sas> /geo GeoPZ2-ISIP.mdb <INTERFACE>" \ 
--env REMOTE_URL="https://<REMOTEHOST>/api/prom/push" --env USERNAME="<USERNAME>" --env PASSWORD="<PASSWORD>" nilslabs/pktvisor-prom-write
```

There are several pieces of information you need to substitute above:

- `<INSTANCE>`: The Prometheus "instance" label for all metrics, e.g. "myhost"
- `<INTERFACE>`: The ethernet interface to capture on, e.g. "eth0"
- `<REMOTEHOST>`: The remote host to remote write the prometheus metric to
- `<USERNAME>`: If required by your prometheus setup, the user name to connect. If not required, leave off this environment variable.
- `<PASSWORD>`: If required by your prometheus setup, the password to connect. If not required, leave off this environment variable.

Other pktvisor arguments may be passed in the PKTVISOR_ARGS environment variable.
Easily Plugin To Elasticsearch

Metrics Collection

Metrics from the REST API

The metrics are available from the agent in JSON format via the REST API.

For most use cases, you will want to collect the most recent full 1-minute bucket, once per minute:

curl localhost:10853/api/v1/metrics/bucket/1

This can be done with tools like telegraf and the standard HTTP plugin. Example telegraf config snippet:

```
[inputs]
[inputs.http]
urls = [ "http://127.0.0.1:10853/api/v1/metrics/bucket/1", ]
interval = "5s"
data_format = "json"
json_query = "*m"
json_time_key = "period_start_ts"
json_time_format = "unix"
json_string_fields = [ "dns_\_\_\_",
    "packets_\_\_\_",
]

[inputs.http.tags]
t = "pktvisor"
interval = "50"
```
**Install Grafana Dashboard**

All dashboards » **pktvisor - prometheus**

**pktvisor - prometheus** by ns1labs

A dashboard for pktvisor observability tool (https://github.com/ns1labs/pktvisor), showcasing Network and DNS metrics.

Last updated: a month ago

Start with Grafana Cloud and the new FREE tier. Includes 10K series Prometheus or Graphite Metrics and 50gb Loki Logs

pktvisor summarizes network data streams in real time. It can capture Network, DNS, and other metrics via packet capture, dnstat, sflow, and other input methods.

This dashboard can be used as a starting point to visualize pktvisor metrics. See the Github page for information on how to deploy and collect these metrics.
Deeper Dive
History

- pktvisor v1 2014 (forked netsniff-ng, remains open source)
- operations, debugging, DDoS visibility
- essentially simple DNS “top”
- deficiencies
  - central collection was a hack
  - resource usage
  - missing IPv6 and TCP support
  - did not track transactions (query/reply pair)
Rewrite

- move to Agent paradigm
- fix deficiencies
- modularize: inputs, dissectors, analyzers, sinks
- parallelize
- summarize with stream processing techniques (DataSketches)
- API first: built-in HTTP control plane
Sliding time window, JSON interface

- maintain mergeable 1m buckets of metrics to provide summary across full window
- always-on Agent supplies information to CLI UI and central collection via HTTP
- both merged and individual buckets are available for collection in REST API
  - CLI UI uses the merged window
  - Central collector gathers a single minute, once a minute
Under The Hood

- agent written in modern C++
- CLI UI is written in Go
- PcapPlusPlus abstraction for pcap input + custom AF_PACKET
- Apache Data Sketches
- optional MaxMind support for GeoIP and ASN
- HTTP(S) API, JSON + native Prometheus output
- Linux, OSX. Windows?
Data Sketches

- fast, probabilistic data structures designed for streaming
- results are approximate but within well defined error bounds
- provide cardinality, heavy hitters (frequent items), quantiles
- designed to be merged, which is how we support time window
- possible to expose raw binary sketch data via API so that it can be merged across hosts and data centers
The Future: Orb

IoT Inspired Cloud Control Plane for Fleet of pktvisor Agents
Orb

1. Agent fleet
2. IoT control plane
3. Data sinks
Orb

1. Agent fleet
2. IoT control plane
3. Data sinks
IoT
Control Plane

API based configuration management

Agents connect via MQTT
Exploring Edge Data with Dynamic Datasets

- apply multiple layered policies per Agent to extract different dimensions of Signal
- separate datasets for each policy
- filter out unwanted upstream data
- choose which summary data to collect
- choose where to send the data (built-in TSDB, S3 bucket, etc)
Orb Project Goals

- open source, vendor neutral, cloud native (microservices, k8s)
- orchestrate fleet of pktvisor Agents
- single pane of glass dashboarding
- create and explore Signal data sets in real time
- central analysis and alerting
<table>
<thead>
<tr>
<th>Directory</th>
<th>File Name</th>
<th>Description</th>
<th>Date</th>
<th>Commits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>wsyick</code></td>
<td>license notice (#12)</td>
<td>e1eb64</td>
<td>4 days ago</td>
</tr>
<tr>
<td></td>
<td><code>cmd</code></td>
<td>license notice (#12)</td>
<td></td>
<td>4 days ago</td>
</tr>
<tr>
<td></td>
<td><code>docker</code></td>
<td>license notice (#12)</td>
<td></td>
<td>4 days ago</td>
</tr>
<tr>
<td></td>
<td><code>docs/images</code></td>
<td>add header</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td></td>
<td><code>pkg</code></td>
<td>license notice (#12)</td>
<td></td>
<td>4 days ago</td>
</tr>
<tr>
<td></td>
<td><code>.dockerignore</code></td>
<td>initial sketch of project</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td></td>
<td><code>.gitignore</code></td>
<td>feature/mainflux bootstrap (#11)</td>
<td></td>
<td>5 days ago</td>
</tr>
<tr>
<td></td>
<td>LICENSE</td>
<td>Initial commit</td>
<td></td>
<td>3 months ago</td>
</tr>
<tr>
<td></td>
<td>Makefile</td>
<td>license notice (#12)</td>
<td></td>
<td>4 days ago</td>
</tr>
<tr>
<td></td>
<td>README.md</td>
<td>Update README.md</td>
<td></td>
<td>2 months ago</td>
</tr>
<tr>
<td></td>
<td><code>go.mod</code></td>
<td>feature/mainflux bootstrap (#11)</td>
<td></td>
<td>5 days ago</td>
</tr>
<tr>
<td></td>
<td><code>go.sum</code></td>
<td>feature/mainflux bootstrap (#11)</td>
<td></td>
<td>5 days ago</td>
</tr>
<tr>
<td></td>
<td><code>version.go</code></td>
<td>license notice (#12)</td>
<td></td>
<td>4 days ago</td>
</tr>
</tbody>
</table>

**About**

Network observability platform, based on [http://pktvisor.com](http://pktvisor.com)

**Contributors**

- weyrick - Shannon Weyrick
- jabyrd3 - Jordan Byrd
- ChemRuislBesares - Che Ruisl...

**Languages**

- Go 94.4%
- JavaScript 0.1%

**Releases**

No releases published

Create a new release
Thank You!

Questions?

Shannon Weyrick  •  VP Research @ NS1  •  Office of CTO

sweyrick@ns1.com