SCALING NETWORK MONITORING IN A LARGE ENTERPRISE

BroCon 2016 – Austin, TX
I work for Amazon’s Worldwide Consumer Information Security group
What are we going to talk about?

How we scaled our network monitoring solution while the network is continuously growing.
Why do we even do this?

Understanding what is occurring on our corporate network is important to us
In the beginning...

We originally decided on using vendor network sensors to get visibility in to what was occurring on our network.
• Decided a vendor appliance was an effective way of gathering the data we needed

• We can buy network sensors, right?

• So we bought network sensors and plugged them into our network
Life was much simpler back then...

- 1Gb/s capable firewalls
- SPAN sessions from our routers to vendor network sensors
- Small number of firewalls to monitor
- We got layer 3 and layer 4 header information from this network sensor
It looked something like this

- Authorized users
- Vendor appliance
- SPAN session
- Corporate network
- Router
- Firewall
- Netflow collector
- Netflow export
- The Internet
What is a SPAN port?

Where do we go from here?

- Our network traffic volume kept growing
- Our sensor vendor stopped selling and supporting the platform we were using
- Increased internal maturity about using this data
- Vendor Management platform can’t scale
  - Driven by API usage by internal customers
  - Started getting close to the limit of network sensors the management platform could handle
Future proofing?

- We have a vendor’s system we’re starting to push the limits on

- What features do we need?

- Do we continue to buy or do we look at building instead?
<table>
<thead>
<tr>
<th></th>
<th>Build</th>
<th>Buy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed of execution</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Control</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Vendor support</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Logistics</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Performance</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>
My co-workers evaluated various options

- nProbe
- Snort
- Suricata
- Bro
• Ran on a single host

• Connected to our router via a 10G fiber link

• SPAN session from the router to our Bro host
Bro Generation One looks like...

Authorized users → Log store → SPAN session → Bro

Corporate network

Netflow export

Netflow collector

Router

Firewall → The Internet
The challenges of Generation One

• The Bro host was a single point of failure

• Individual host installs have high operational costs

• High traffic volumes on our SPAN sessions caused our router to reboot

• Will this continue to scale with the growth of our network?
## Scorecard

<table>
<thead>
<tr>
<th></th>
<th>Vendor solution</th>
<th>Generation One</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single point of failure?</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Data collected via</td>
<td>SPAN</td>
<td>SPAN</td>
</tr>
<tr>
<td>Control</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Scalability</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Logistics / Install effort</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Cost per Gb/s</td>
<td>$$$</td>
<td>$</td>
</tr>
</tbody>
</table>
And we are done!

Or so we thought....
• Seth spotted everything in the history field was in upper case
  – Turned out to be a trivial configuration change

• We started off with 32GB of RAM in our hosts and ended up upgrading to 128GB
Scaling to infinity and beyond!

- Capture loss levels (as reported by Bro) started rising beyond acceptable levels once we were past 3Gb/s of traffic on our existing hardware platform.

- We knew that traffic levels were going to continue to increase so our design needed to evolve as well.
• We migrated to optical taps over SPAN sessions
  – SPAN sessions were good for speed of deployment but not for long term use

• Introduced a method to allow us load balance traffic among physical hosts
  – Similar outcome to the work done by LBNL
  – Eliminated the SPOF with our Bro host
  – https://commons.lbl.gov/download/attachments/120063098/100GIntrusionDetection.pdf
While we do run Bro in a cluster, it is limited to a single physical host

We don’t want to share state across hosts

The Bro manager process being a single point of failure isn’t all that appealing to us

Keep the hosts simple and consistent
And here is how it looks

**Diagram:***

- **Bro host #1**
- **Bro host #2**
- **Bro host #3**

- **Load balancer**
- **Corporate network**
- **Optical tap**
- **Router**
- **Netflow export**
- **Firewall**
- **The Internet**
- **Netflow collector**
<table>
<thead>
<tr>
<th></th>
<th>Vendor solution</th>
<th>Generation One</th>
<th>Generation 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single point of failure?</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>Data collected via</td>
<td>SPAN</td>
<td>SPAN</td>
<td>Optical taps</td>
</tr>
<tr>
<td>Control</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Scalability</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>Logistics/Install effort</td>
<td>✗</td>
<td>✗</td>
<td>✗ ✗</td>
</tr>
<tr>
<td>Cost per Gb/s</td>
<td>$$$</td>
<td>$</td>
<td>$</td>
</tr>
</tbody>
</table>
Optical taps overview

- **Router**
  - TX
  - RX
  - TX 10Gb/s

- **Load balancer**
  - TX 10Gb/s

- **Optical tap**
  - RX
  - TX

- **Firewall**
  - TX
This was a great step forward, but it was only an incremental improvement.

We can now scale out but it is still time consuming to get individual hosts deployed.

Migrating to an integrated solution would help solve these challenges.
• Combined our hosts, load balancers and optical taps into a “cookie cutter” rack design

• We now just order a small, medium or large rack depending expected traffic volumes
Scaling Bro Generation 2.0 footprint

Network rack

Load balancer

Optical tap

Router

Firewall

Bro rack #1

Load balancer

Bro host #1 Bro host #2 Bro host #n

Bro rack #2

Load balancer

Bro host #1 Bro host #2 Bro host #n
<table>
<thead>
<tr>
<th></th>
<th>Vendor solution</th>
<th>Generation One</th>
<th>Generation 1.5</th>
<th>Generation 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single point of failure?</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Data collected via</td>
<td>SPAN</td>
<td>SPAN</td>
<td>Optical taps</td>
<td>Optical taps</td>
</tr>
<tr>
<td>Control</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Scalability</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Logistics/Install effort</td>
<td>✗</td>
<td>✗</td>
<td>✗ ✗</td>
<td>✓</td>
</tr>
<tr>
<td>Cost per Gb/s</td>
<td>$$$</td>
<td>$</td>
<td>$</td>
<td>$</td>
</tr>
</tbody>
</table>
What do we do with all this data?

We stream the logs to our central log store
Central log storage
Our original ETL jobs were based on the Bro 2.3 field order (output in TSV)

– Bro 2.4 changed the ordering of some of the fields

– Use JSON if you’re loading this data elsewhere
  • One line configuration change!
• Scale horizontally and not vertically
• Stateless sensors
• Decouple dependencies
• Plan up-front
• Lab testing is never overrated
• Get experts on-site to validate
• Document wins
• Know your customers
Industry peers

– Thanks to LBNL, Mozilla and the others who responded to our queries and everyone who has publicly spoke or documented their install
Thank you!