BLOG POSTS

	Content
Post Title	Exploring an ELK Stack: Part 5: Logstash and Beats
Post Date	
Attributed To	Peter Welcher
Written By	Peter Welcher
Reviewed By (Name & Date)	Dave Donati (11/19)
Reviewed By (Name & Date)	

Meta Title (55 characters including spaces)	Exploring an ELK Stack: Part 5: Logstash and Beats
Meta Description (156 characters including spaces)	
Target Keywords	
Categories	Technology
Tags	N/A
Call to Action	N/A
Image	Put a brief description of what the image should be or note that a file is attached. DO NOT paste the image into this Word doc; send it as a separate file.

Note: Naming convention for files as they go back and forth

- Original writer names file with "_V1" at the end (e.g., blogtitle_V1)
- First reviewer, makes edits and renames with initials at end (e.g., blogtitle_V1_af)
- If another reviewer, again add initials to end to keep the string of reviewers (e.g., blogtitle_V1_af_pw)
- When original writer gets it post back with edits, she makes revisions and saves the file as V2 (e.g. blogtitle_V2) – then reviewers continue as above with initials
- When post is complete, it is saved with "Final" and the post date at the end (e.g. blogtitle_FINAL_022012)

COPY FOR POST:

This is Part 5 of a blog series tutorial about the ELK stack. The prior blogs in the series are:

- Exploring an ELK Stack Part 1: Importing Data and Patterns
- Exploring an ELK Stack: Part 2: Kibana Visualizations
- Exploring an ELK Stack: Part 3: Dashboards
- Exploring an ELK Stack: Part 4: Elasticsearch and Pivot Tables

Commented [PW1]: Insert link

Commented [PW2]: Insert link

Commented [PW3]: Ditto: insert link

Commented [PW4]: Link needed!

Part 1 introduced the ELK stack and covered getting some data into Elasticsearch via a new experimental Kibana feature. It also covered Logstash grok patterns lightly, and some web tools for working with them (regular expressions made easier!).

Part 2 covered some basic things you can do with the Kibana visualization tool (the "K" in "ELK).

Part 3 walked through creating a Kibana dashboard.

Part 4 focused on a syslog reporting task, solving it via Kibana and Elasticsearch queries. It also covered Python scripts to post-process the query results and create an Excel pivot table.

This blog, Part 5, lightly covers the Logstash and Beats tools from Elastic (the company that developed the Elasticsearch suite of tools).

LOGSTASH

Logstash exists to ingest data and store it for use by the Elasticsearch search engine. It can process various inputs, primarily logging data or data in a similar format.

It can transform the data (clean up, reformat, etc.) and forward it to various outputs, one of which is Elasticsearch. It might also forward the data to other tools for analysis, archival, alerting, and / or monitoring. **Note**: Elasticsearch can also do alerting.

One option is to activate geo-location as part of the data transformation.

Logstash supports plug-ins for specialized tasks. For example, it can process Azure, ArcSight and NetFlow data via specialized modules.

Besides log data, Logstash can also collect metric data from various platforms, including IOT sensor data.

That's the short version of the product description, and about all I have to say about Logstash. In skimming the documentation, I found nothing I felt the need to try out within Logstash itself. It looks like there's a lot of useful functionality. However, building data pipelines to other tools gets outside the intended scope for this series of blogs.

See the References section for links to the Logstash documentation for more detail.

Beats

The Beats functionality does strike me as more immediately relevant and interesting.

Beats are agents that you can run on systems, to process and forward data to Logstash or directly to Elasticsearch or other tools. Their web page describes them as "single-purpose lightweight data shippers". I interpret that as getting data off a system and into the central Elasticsearch system. In addition, we should be thinking "distributed processing" — the Beat can pre-process data to a degree.

Sometimes you won't miss a Beat, so to speak; routers might forward syslog directly to Logstash or Elasticsearch. In other cases, though, you need a way to get the data off the system.

Here are some of the Beats that are available, with Elastic's descriptions.

- Filebeat: Log Files
- Metricbeat: Metrics
- Packetbeat: Network data
- Winlogbeat: Windows event logs
- Auditbeat: Audit data
- Heartbeat: Uptime monitoring data
- Functionbeat: Serverless data

My intent here is to drill down on the two or three of those that are most relevant for network people. I'll dismiss the rest with "there's a ton of support for server / application data capture and forwarding".

Filebeat

Filebeat essentially does tail -f on specified log files and forwards them to the Logstash or another target port. Tail -f is a Linux command that follows a log file and displays new lines as they come in. In this context, as new log lines get appended to a local log file, Filebeat ships them off to Elasticsearch as well.

If you want, you can then watch the log file data from all the sources as it arrives centrally via Kibana discovery. The Filebeat page (see References below) has an animation showing what that looks like.

Filebeat remembers what it has shipped, so if interrupted, it will pick up where it left off. I've experienced this myself, while experimenting with using Filebeat as a way to load syslog data into the system. (Drawback: by default, Filebeat adds its own current timestamps, so you'd need a custom pattern to process such data I ELK.)

Filebeat has modules for common cases: auditd, Apache, NGINX, system, MySQL and other logs. For such cases, Filebeat can set up Elasticsearch patterns and Kibana dashboards automatically for you. Apparently, some also now do machine learning.

Filebeat also supports backpressure, so as to not overrun the central processing pipeline. That is, it slows down the sending of log messages if the central processing needs it to.

There are now Palo Alto and Cisco ASA modules for firewall logs. This is probably because processing firewall logs as a SIEM is a big market. I'll also note router / switch syslog is pretty much covered by what we did earlier in this blog series!

And yes, there's an Elastic SIEM application in Kibana.

There is now also a NetFlow module that will monitor and forward NetFlow and IPFIX flow records. See the links in the References section.

Heartbeat

From the product page: "Heartbeat pings all things, via ICMP, TCP, HTTP, TLS, with support for authentication and proxies". It also allows monitoring the servers behind a load balancer via DNS resolution. And it supports dynamic automation of monitoring targets. Heartbeat also holds data when it can't be delivered, until connectivity is available again.

More specifically, heartbeat generates uptime and response time data.

Packetbeat

Packetbeat provides packet flow data. It includes a library that supports some application layer protocols. It can work with the Geo Location Logstash function to provide geographic context.

Logstash has a DNS plugin to convert IP addresses to DNS names, but there is a caution that it will considerably slow down processing in a pipeline, doing the lookups. I didn't see mention of whether it does local caching or not.

Here's where we get to the hands-on fun part of this blog!

Working with Packetbeat

Packetbeat works with libpcap, which I already have on my Mac. So, I downloaded and installed Packetbeat (see the Downloads link in the References section).

Part of the setup is to edit the configuration file packetbeat.yml in the folder /usr/local/etc/packetbeat.

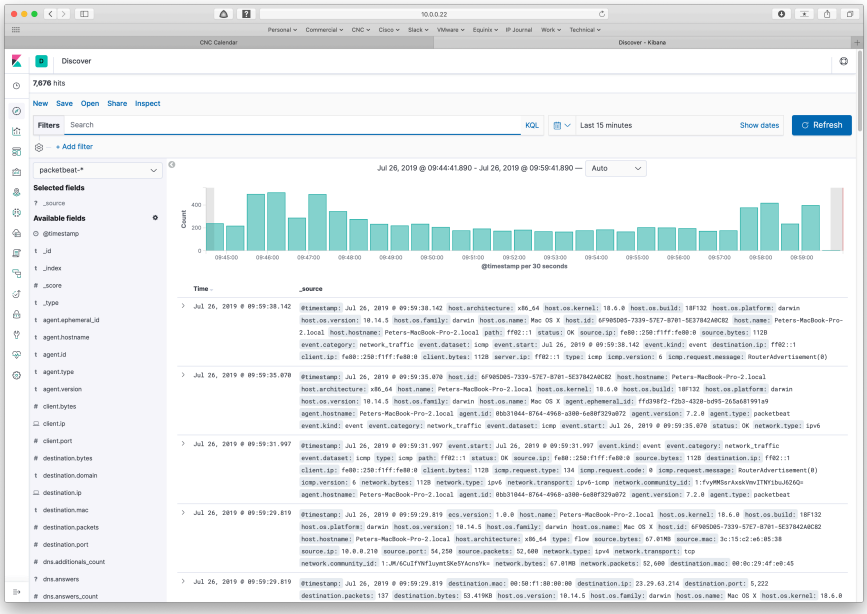
Things to set and / or check:

- Do you have the active interface for packetbeat.interfaces.device? (Yes, en0 in my case).

- Set setup.dashboards.enabled to true, to use the built-in dashboards.
- I set Kibana host to “10.0.0.22:5601” (the IP of my VM running the ELK container, which is different than the IP of the Mac on the local network).
- Similarly for Elasticsearch, except the port is 9200.
- I left the Logstash part as-is, commented out.
- I changed nothing else in the configuration file.

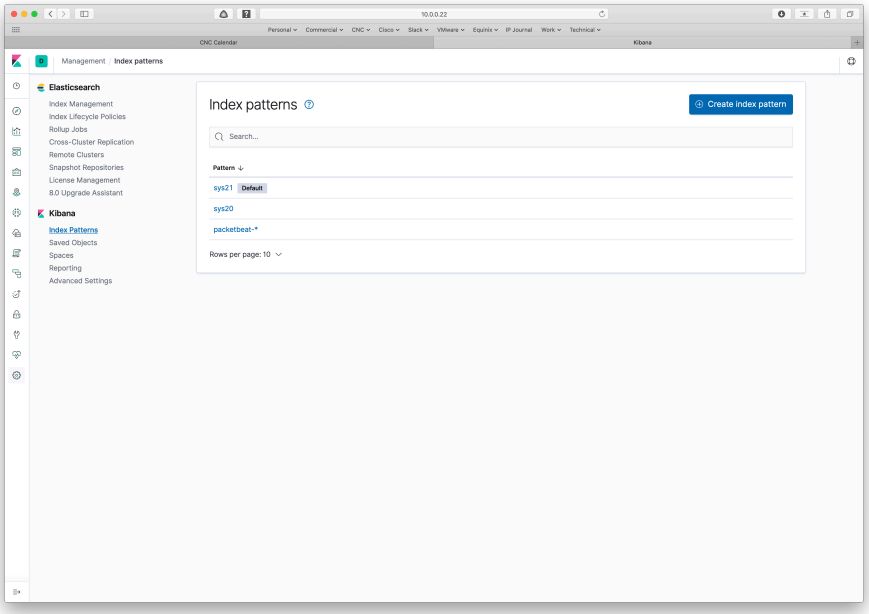
As documented, I ran the daemon via “sudo packetbeat -e -c packetbeat.yml”. You do need to first make sure your ELK stack is up and listening.

Once things were working correctly, I went to Discover in Kibana and “packetbeat-*” showed up in the Index list.



If you scroll the “Available Fields” area on the left, you can see all the information that Packetbeat is sending to Elasticsearch.

If you click on the Settings icon (the bottom icon on the left, the gear), then on Index Patterns, you can see that a pattern was also created for you:



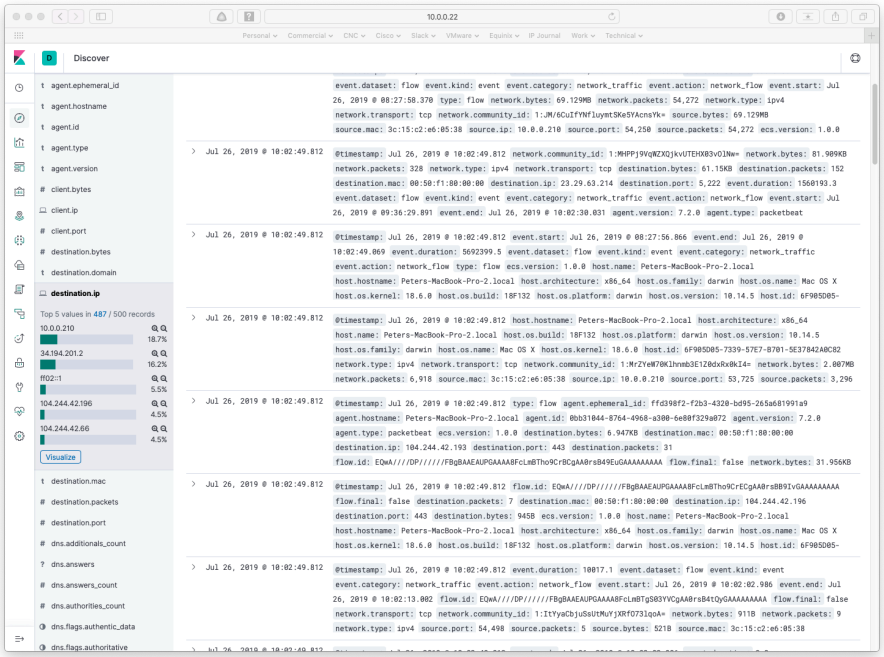
Clicking on Index Management will also confirm the existence of the Index we saw in Discover.

I'm mentioning this because either the setup.dashboard flag being "false" or firing up Packetbeat too soon after starting the ELK stack caused the index and pattern to not get created in Kibana. Troubleshooting ensued. The above helped.

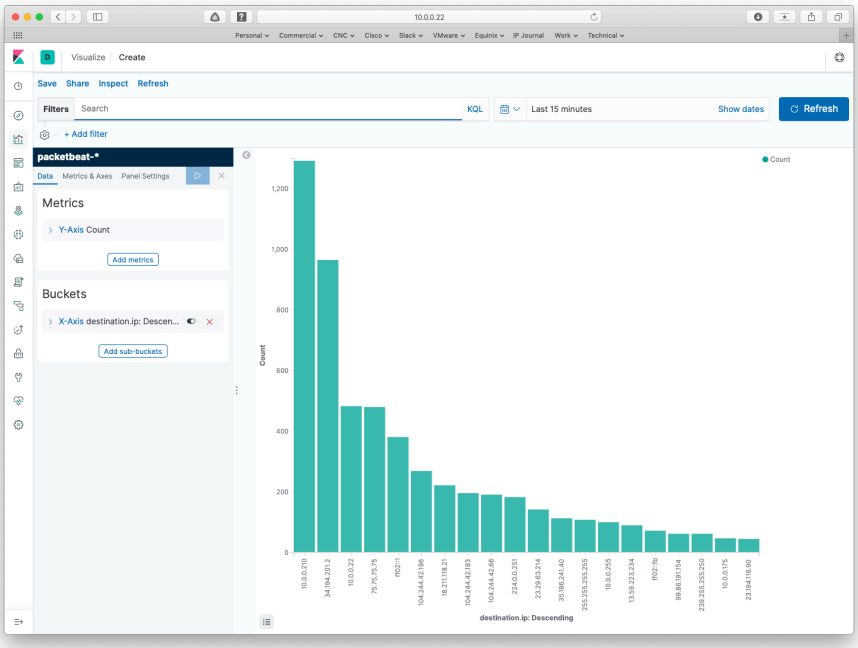
PACKETBEAT VISUALIZATION

Before we sample the supplied dashboards, let's create a basic visualization from the Discover screen.

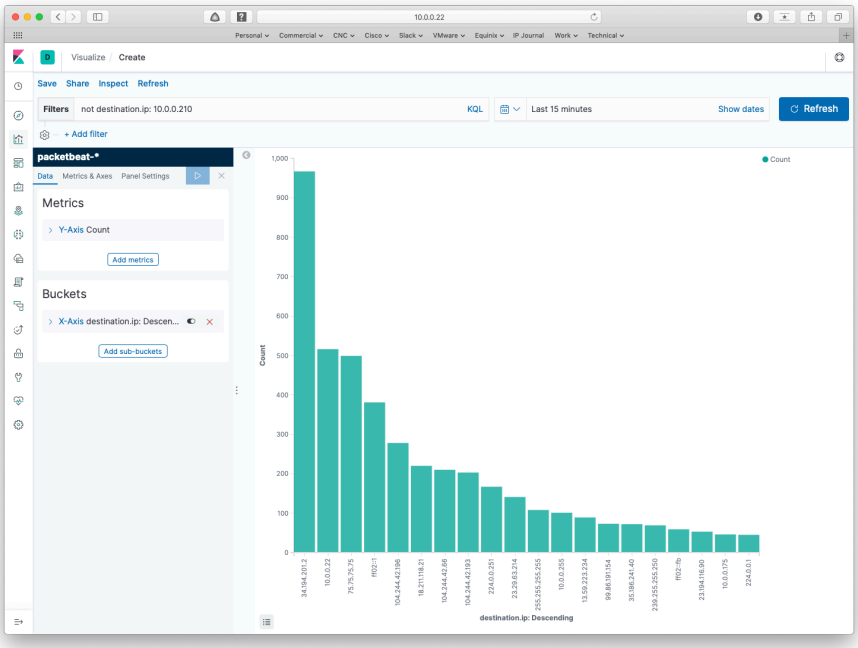
I found "destination.ip" field on the left, and clicked on Visualize.



That gets you this:



10.0.0.210 is the address DHCP gave my Mac. Let's add a filter to exclude that destination, so we see only the sites my Mac is talking to. To do that, click in the Filters search space, and enter "not destination.ip: 10.0.0.210". Then click the Refresh button.



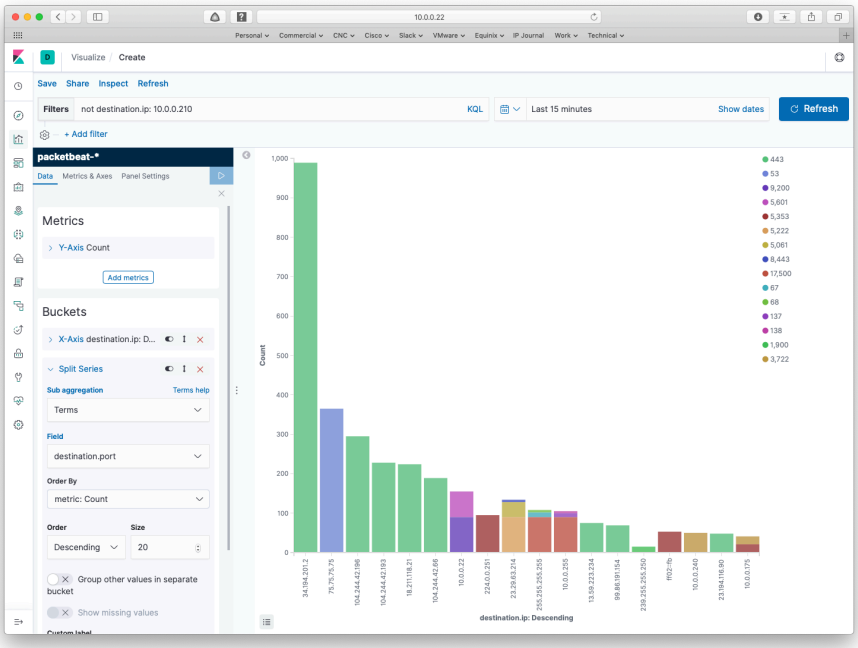
Using dig -x 34.194.201.2, I see my top destination is in EC2 at AWS. That's presumably a service that's hosted in AWS.

Exercise for the reader: If you're following along and trying this at home, this is where you might want to try enabling DNS lookup and Geo Location in Logstash. That was a little more than I wanted to spend time on for this blog.

I suggest adding sub-buckets by clicking on the button, clicking on Split Series, picking the aggregation terms, and the field destination.port. Change size to 20 and click on the blue right arrow to make the change take effect.

This should all be sounding familiar by now!

That makes the graph more interesting:



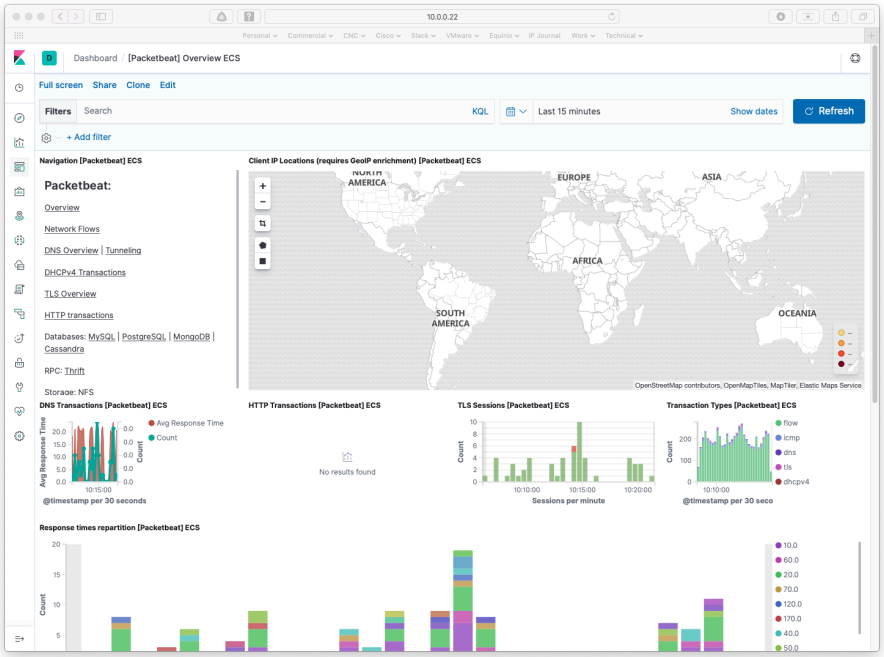
Does that suggest that everything is becoming TLS hence indistinguishable?

PACKETBEAT DASHBOARDS

That concludes the DIY part of this blog. The rest is exploring a couple of highlights among the wealth of canned dashboards.

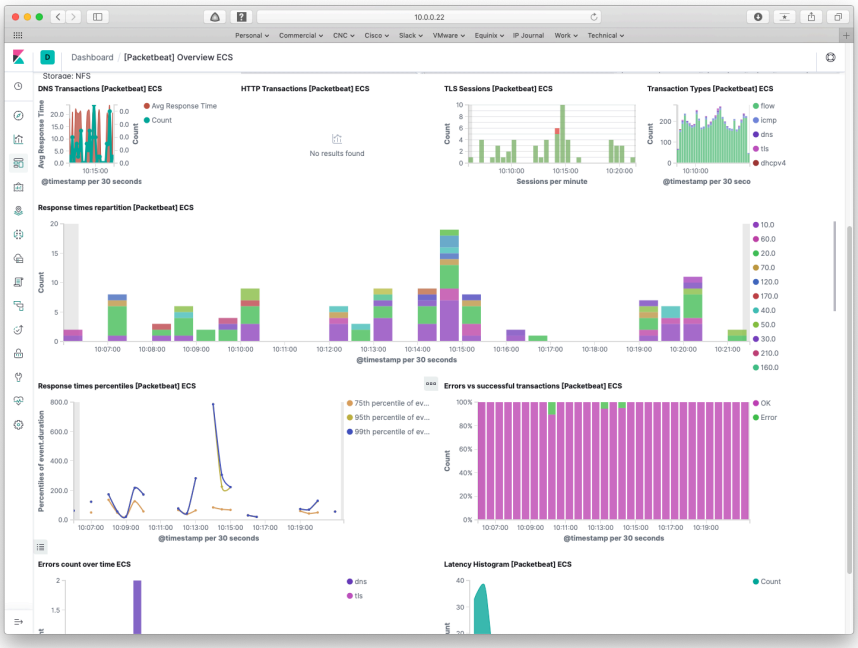
Go to the Dashboard icon and click it. That brings up a long list of all the dashboards Packetbeat added to Kibana. Find “[Packetbeat] Overview ECS” and click on it.

Here’s the top of that dashboard:



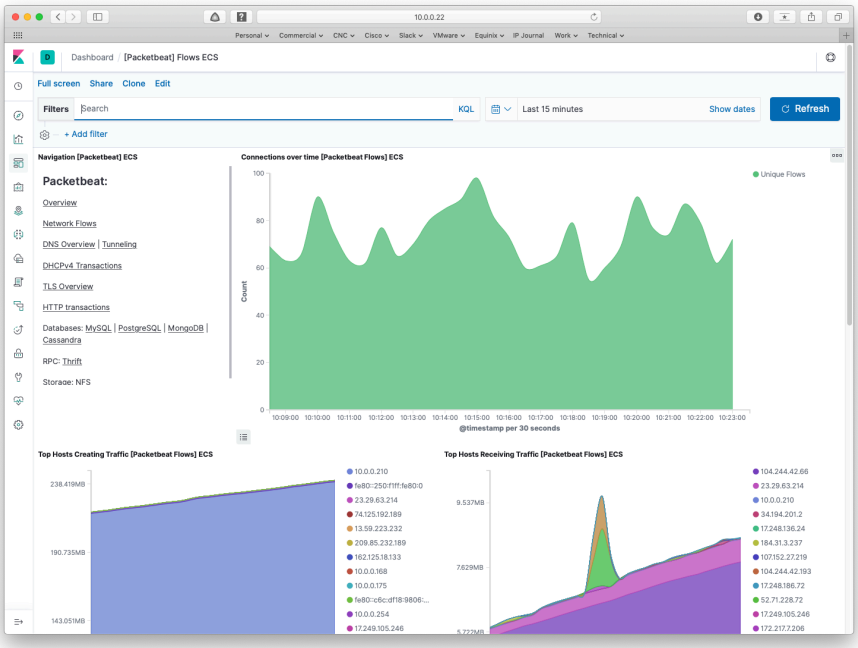
I don't have the Geo tagging turned on, so the world map is blank. Do note the links on the left, which appear to be common across many of the supplied dashboards.

Here's further down on that dashboard:

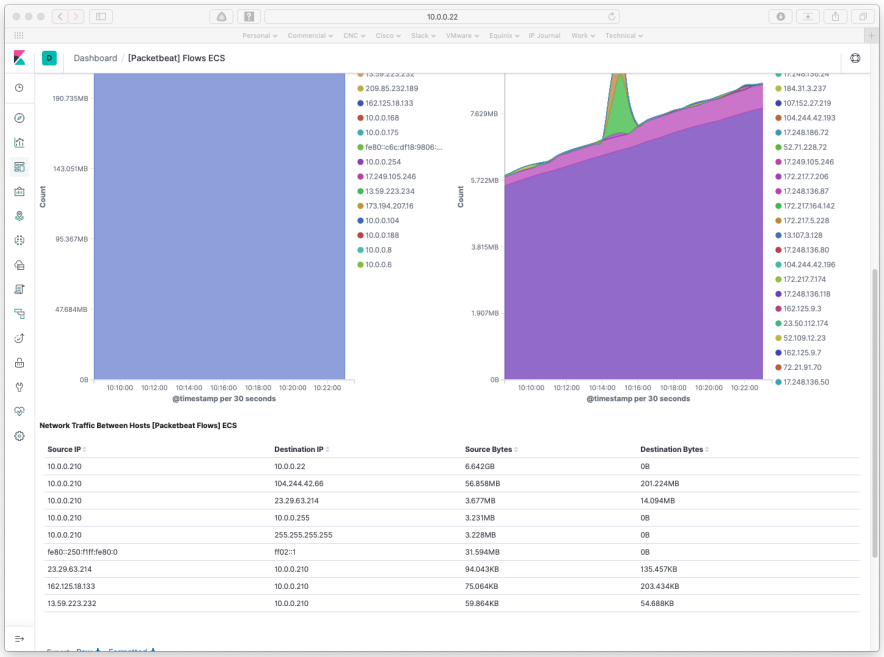


I'll leave interpretation to you.

Clicking on the Network Flows link brings up the following:



And the bottom part of that window:



Again, I'll leave interpretation up to you. Note that you can clone and edit the dashboards if you don't like them, so they provide a quick way for you to get up and running with Packetbeats data.

Exercise for the reader: Tour the other Packetbeats dashboards.

More Beats

There is also a Beats community for contributed open source Beats. It can be found at https://www.elastic.co/guide/en/beats/libbeat/current/community-beats.html.

Conclusion

I hope you've enjoyed this series of blogs and working with ELK + Beats.

My perspective is that the path to value (information out!) is fairly short, leveraging the power of these tools to get data about your network.

How much you want to invest, both money and time, in servers, storage, building clusters for robustness, installing the applications rather than running in a container, is something to consider before going too far down the ELK path.

I've had another reason for working with ELK, which is likely to surface in another blog. **Hint**: telemetry, Nexus 9K.

REFERENCES

See also the Part 1 to 4 References.

Logstash:

- Product page: https://www.elastic.co/products/logstash
- Documentation: https://www.elastic.co/guide/en/logstash/current/index.html

Beats:

- Product page: https://www.elastic.co/products/beats
- Documentation: https://www.elastic.co/guide/en/beats/libbeat/current/index.html
- Links to documentation for specific major Beats: https://www.elastic.co/guide/index.html (scroll down to the Beats section)

Filebeat:

- Product page: https://www.elastic.co/products/beats/filebeat
- Latest (7.2) Release Announcement: https://www.elastic.co/blog/beats-7-2-0-released
- Filebeat NetFlow:
  - https://www.elastic.co/guide/en/beats/filebeat/master/filebeat-input-netflow.html
  - https://blogs.cisco.com/security/step-by-step-setup-of-elk-for-netflow-analytics
  - https://www.elastic.co/guide/en/beats/filebeat/7.5/filebeat-module-netflow.html
  - https://www.gns3.com/news/article/monitoring-network-infratsructur

Heartbeat:

- Product page: https://www.elastic.co/products/beats/heartbeat
- Documentation: https://www.elastic.co/guide/en/beats/heartbeat/current/index.html

Packetbeat:

- Product page: https://www.elastic.co/products/beats/packetbeat
- Documentation: https://www.elastic.co/guide/en/beats/packetbeat/current/index.html
- Flows page:

https://www.elastic.co/guide/en/beats/packetbeat/master/configuration-flows.html

- Download and install page: https://www.elastic.co/downloads/beats/packetbeat

Google or other search engines will also lead you to more resources, including courses and canned visualizations others have developed.

COMMENTS

Comments are welcome, both in agreement or constructive disagreement about the above. I enjoy hearing from readers and carrying on deeper discussion via comments. Thanks in advance!

Hashtags: #CiscoChampion #TechFieldDay #TheNetCraftsmenWay

Twitter: @pjwelcher

Disclosure Statement

[INSERT the usual IMAGES HERE: 20 Year CCIE and Cisco Champions **2019** as per recent blogs]

NETCRAFTSMEN SERVICES

Did you know that NetCraftsmen does network /datacenter / security / collaboration design / design review? Or that we have deep UC&C experts on staff, including @ucguerilla? For more information, contact us at <<insert suitable link here>>.

SOCIAL MEDIA:

Facebook: Like, comment or share our status using this link.

Twitter: Like and RT our tweet using this link.

LinkedIn: Like, comment or share our status using this link.