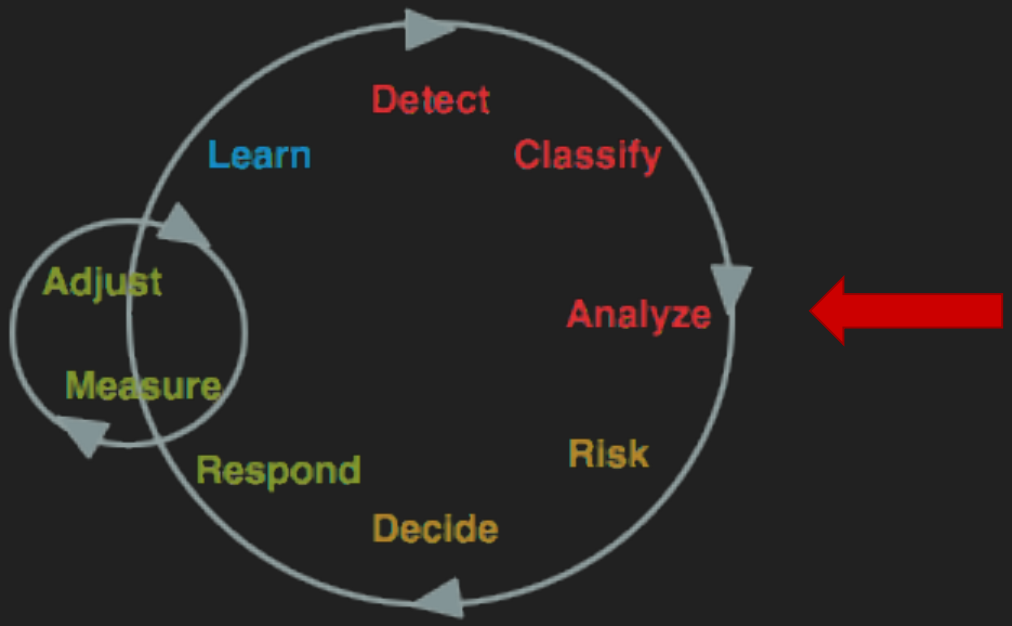
CoreFlow: Enriching Bro security events using network traffic monitoring data

Ralph Koning
Universiteit van Amsterdam, ESnet
Nick Buraglio (ESnet), Cees de Laat (UvA, ESnet) , Paola Grosso (UvA)

SARNET

Secure Autonomous Response Networks

- Goal is to build a networks that can autonomously defend against cyber attacks.
- Divided in three parts, Strategic, Tactical, **Operational**



The work at ESnet explores the analyze part.

CoreFlow motivation

To effectively block attacks, the information from an IDS is not always sufficient

When an event triggers, the security team has to **manually collect** additional **data from different sources** to enrich the event to **create context and understanding** of the event.

CoreFlow can auto this process by automatically correlating the security events to available data sources and provide this context.

In this prototype we focus on the following sources:

- **Bro** - Generates the events
- **NetFlow** - To add network traffic information
- **Route Explorer** - To assist in determining paths

Bro

Bro is an Intrusion Detection System that relies on **Deep Packet Inspection.**

- Scalable by clustering
- Bro outputs events in multiple log files:
  - Per protocol x509, ssh, ftp, http, sip,
  - **Connection** log, contains flow data that the detector sees
  - **Notice logs**, security alerts that require attention or processing
- Unique identifier per alert based on alert characteristics
  - Used to lookup more information from other files

Bro at ESnet:

Netflow

Netflow is a protocol to export statistical flow data from network devices to collectors.

Netflow contains information that is not available in Bro

A flow is a set of packets between one source and destination within a certain time
Flow = **(source IP, source port, destination IP, destination port, protocol)**

Netflow at ESnet:

- collected collected on routers in ESnet (53 sources)
- 10GB of data for all routers in ESnet per day
- Samplerate: 1:1000 packets
- Accessed using NFS

Correlation

Why?

- Bro provides information on the content of the event and basic traffic information from a fixed viewpoint in the network.
- Netflow is collected on all the routers in the network and includes more traffic specific information (which is not available in bro) such as: Router, interface, VLAN, MPLS label, TOS

How?

Correlate on common information that is available in both data sources:
(source IP, source port, destination IP, destination port, protocol)

When there is no matching data in the other data source the events are still sent out but are not enriched.

CoreFlow

CoreFLow correlates events from Bro with Netflow data (and in the future maybe other sources)**.**

Accepts input from:

- File (bro log files)
- Elasticsearch imported flow data
- STDIN bro log data
- Splunk

Enrichment with data from:

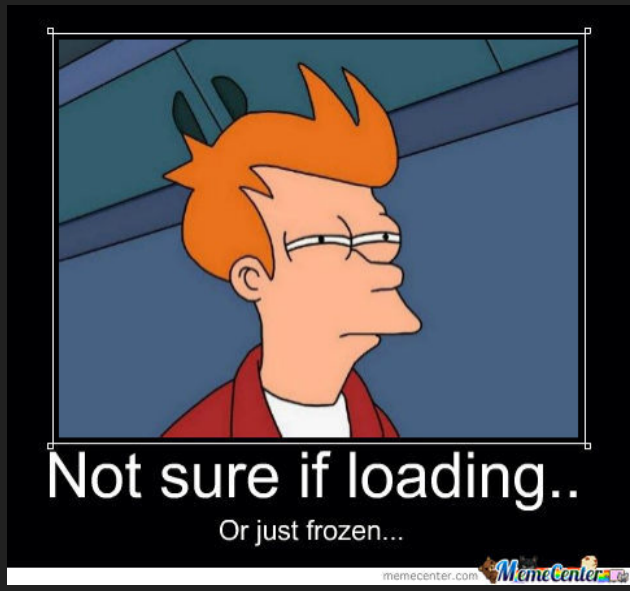
- netflow (elasticsearch or nfdump)
- packet designs route explorer (in progress)

Outputs to:

Written in:

- Python 3.5
- requests, elasticsearch

Approaches that did not work



Loading data into memory is a bad idea*

- The data sets are to big to load into memory
- Swapping and reading from disk renders system Unusable.
  Solution: filter searches for alerts and use iterators

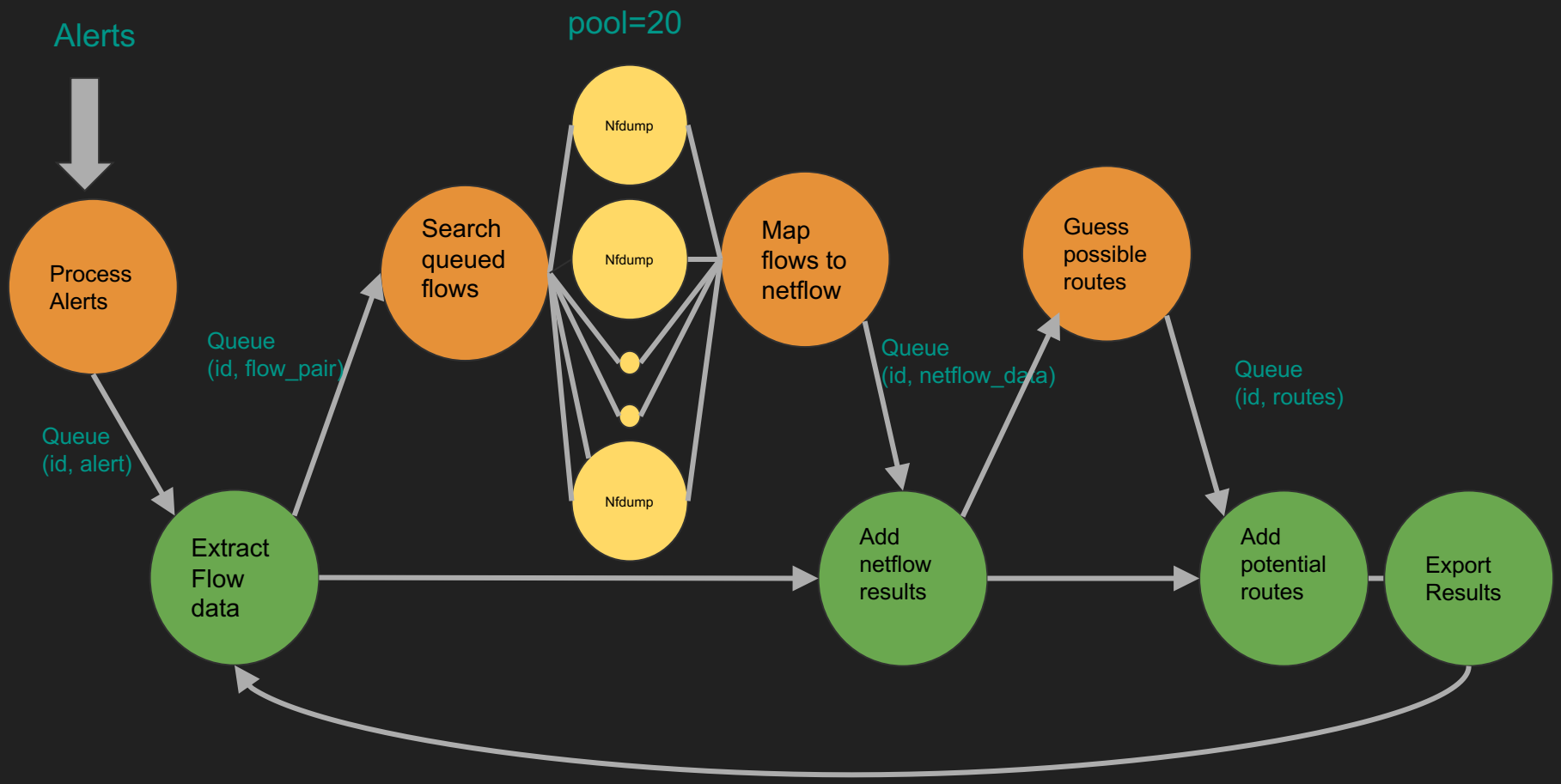
Searching flow by flow is slow*

- Batching flows and querying for the whole batch give acceptable speeds
- This requires an extra step to map the results back to the originating flow

Importing and indexing bro conn/netflow data into elasticsearch*

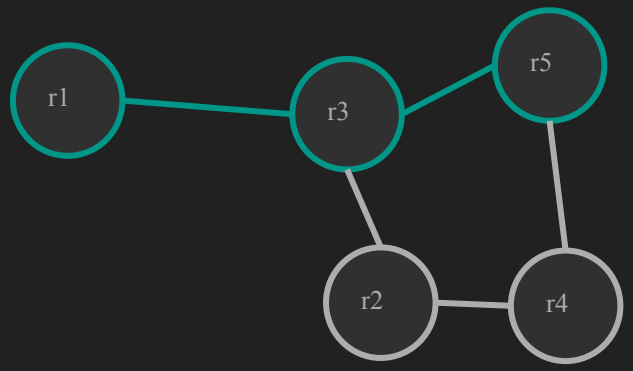
- Took me a few hours to import bro and netflow data for one hour
- Streaming information directly into elasticsearch may be better

Current workflow



CoreFlow Route estimation algorithm

- It's able to fill in missing routers
- Flow traverse a router multiple times (loops)
- Finds potential 'shortest paths'



- Topology information from OSCARS
- Based on latest topology
- Does not account for policies or metrics

Unordered route:	Get possible routes from r3:	Reverse	Concat	Shortest
r3, r1, r5	r3, r1	r1, r3	r1, r3, r1	r1, r3, r5
	r3, r5	r5, r3	r1, r3, r5	r5, r3, r1
	r3, r2	r2, r3	r1, r3, r2	
	r3, r5, r4	r4, r2, r3	r1, r3, r2, r4	
	r3, r2, r4	r4, r5, r3	r1, r3, r5, r4	
			…	

Route estimation with Route Explorer

- Appliance sold by Packet Design
- Route Explorer peers with the routers in a network and stores routing information
- It also records routing changes and historical data
- Accounts for metrics and routing policies
- It provides an API that can be used to calculate paths at a point in time

The required information comes from the following sources:

	Date	Source router	Destination Prefix
Bro	X		X
Netflow	X	X	X

Conclusions

- Increasing the sample rate increases the chance of finding an event in the flow data.
- When flows show up we can, in some cases, estimate the path the malicious flow took through the network.
  - This allows for filtering traffic at the network entry point
- Some analysis tools require data that is not available in just one data source; Enrichment can provide the required information for these tools to operate.
  - E.g. Route Explorer

Future work

- Modularize core
- Add more information sources:
  - PerfSonar, syslog, etc
- More advanced alerts
  - Lower threshold for alerts from bro
  - New critical alerts based on enriched Information
- Experiment with different sample rates: 1:1
  - At the edge?
  - In the core?

Thank you!

COMMIT/

Code available at (private repository):

https://github.com/esnet/CoreFlow

Contact:

Ralph Koning (UvA / ESnet)

rkoning _at_ es.net | r.koning _at_ uva.nl

https://staff.fnwi.uva.nl/r.koning/

Nick Buraglio (ESnet)

buraglio _at_ es.net

https://www.es.net/about/esnet-staff/network-planning/nick-buraglio/

SNE

UNIVERSITY OF AMSTERDAM

ESnet

ENERGY SCIENCES NETWORK