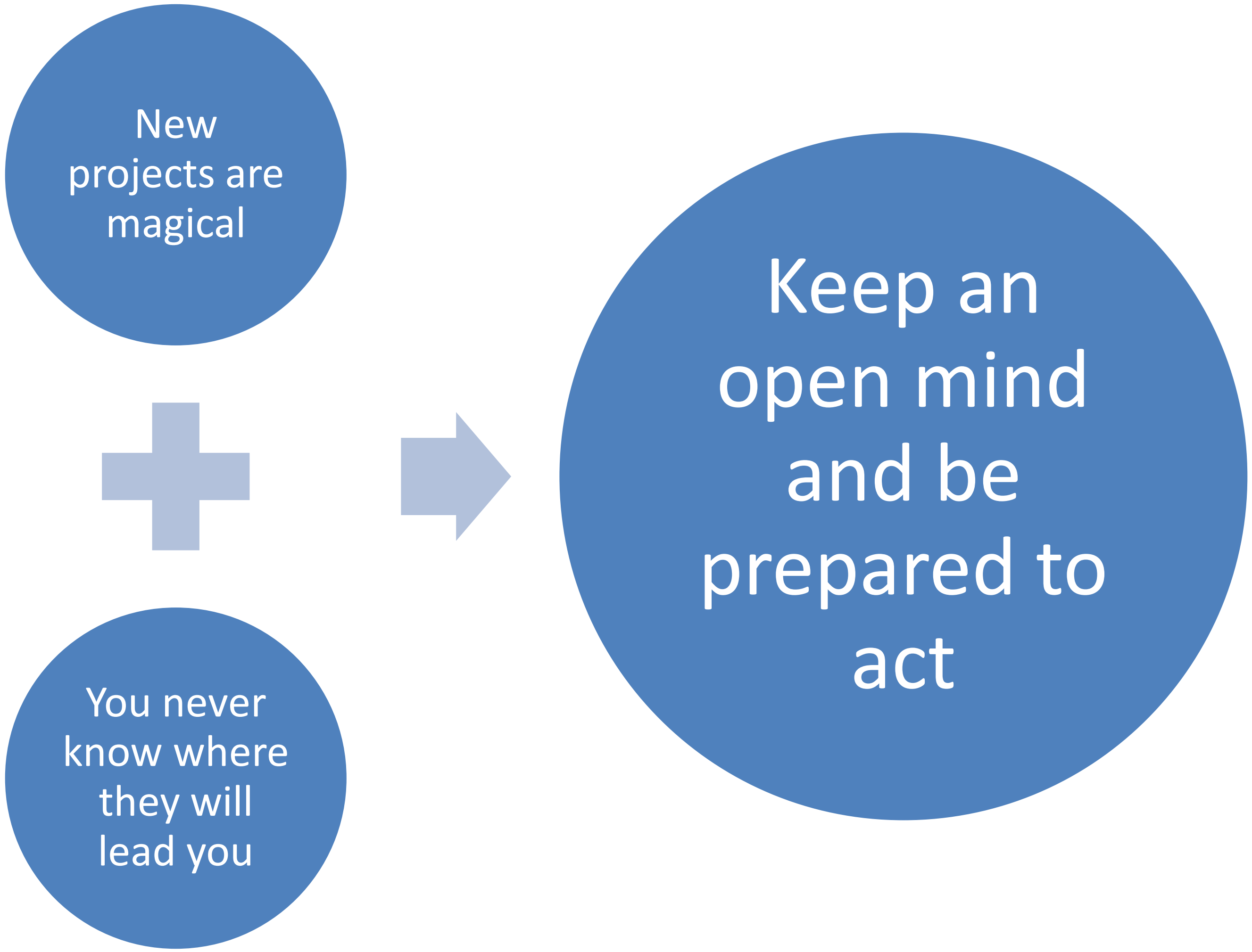
From Bandwidth to Beacon Detection, Prism and Touchpoints

George Jones
Paul Krystosek
Sid Faber
SEI CERT NetSA

Introduction

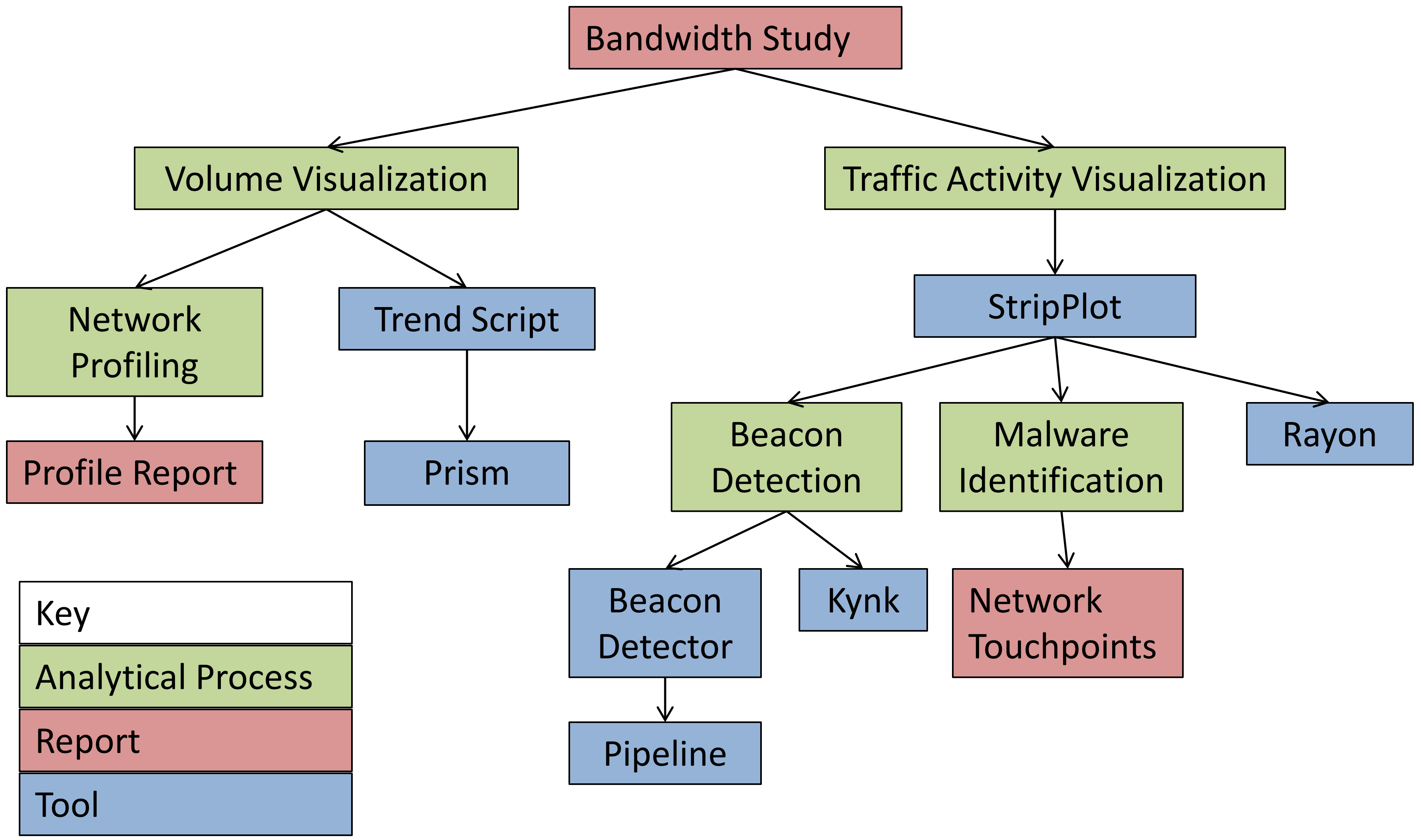


New projects are magical

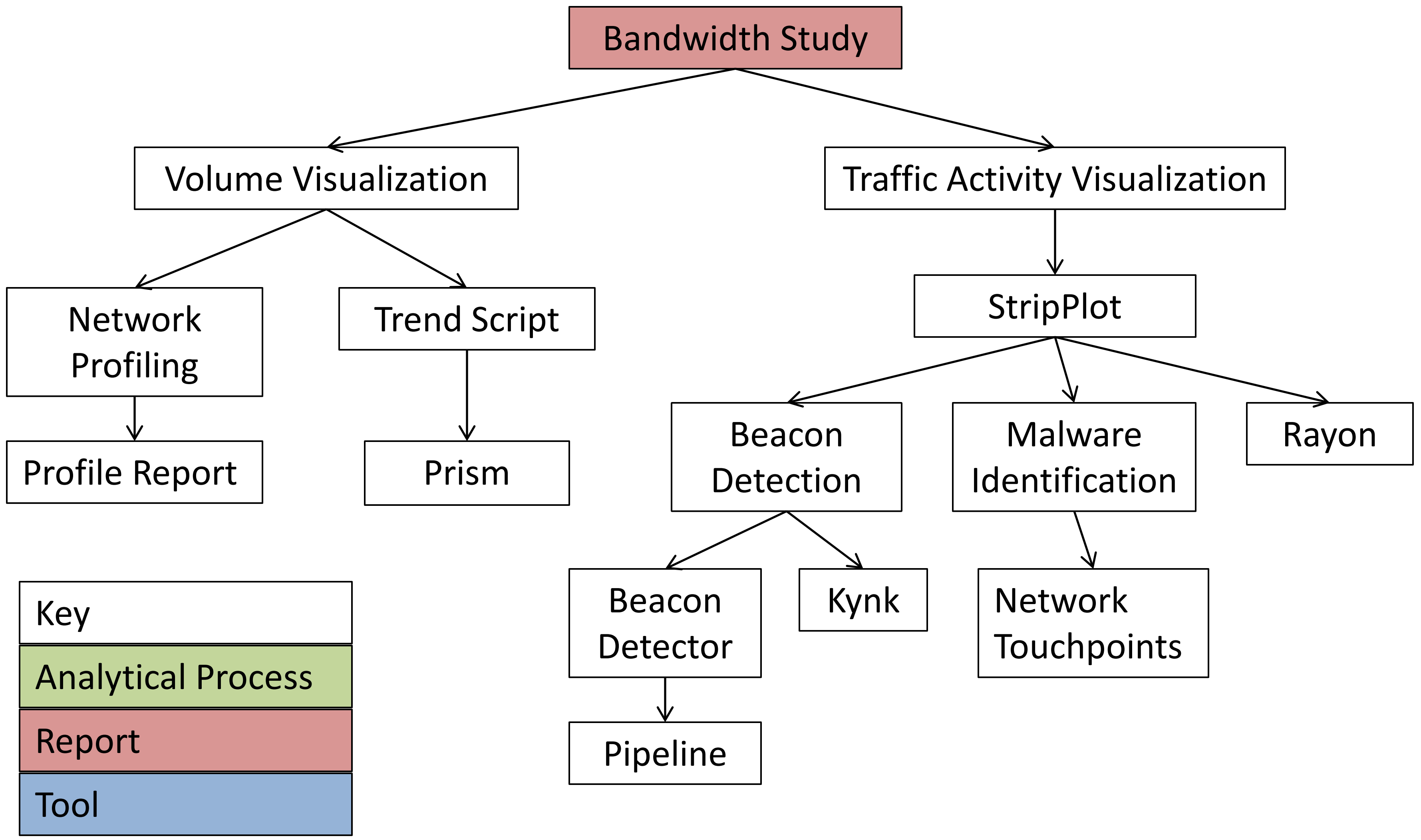
You never know where they will lead you

Keep an open mind and be prepared to act

Overview


Bandwidth Study
Volume Visualization
Traffic Activity Visualization
Network Profiling
Trend Script
StripPlot
Profile Report
Prism
Beacon Detection
Malware Identification
Rayon
Beacon Detector
Kynk
Network Touchpoints
Pipeline
Key
Analytical Process
Report
Tool

The starting point



The Bandwidth Study

Once upon a time…

There was a network that everyone thought was dirty.

They planned to get some sensors in place…

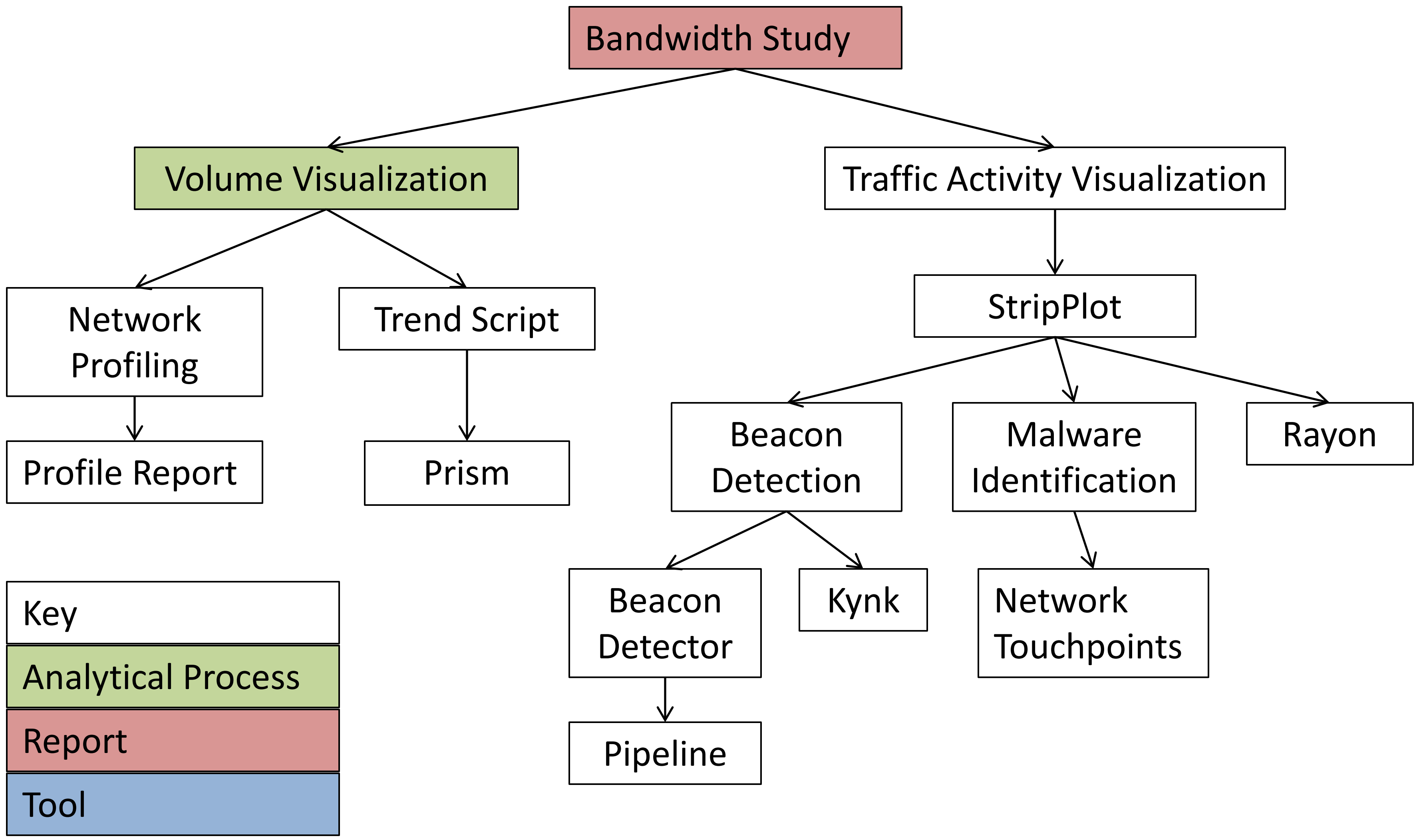
but all they had for now was flow.

What could be done to keep them safe until sensors were deployed?

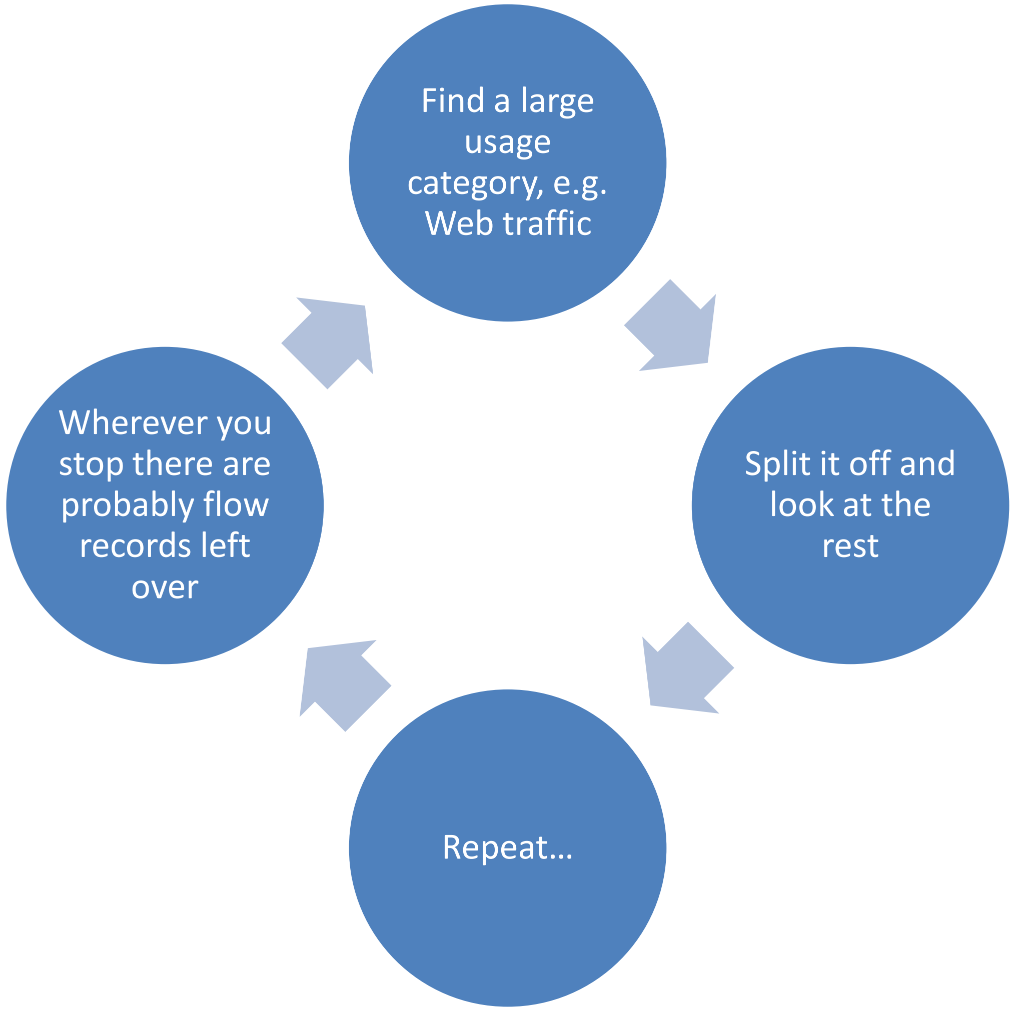
This is where our story starts

From there it meanders hither and yon

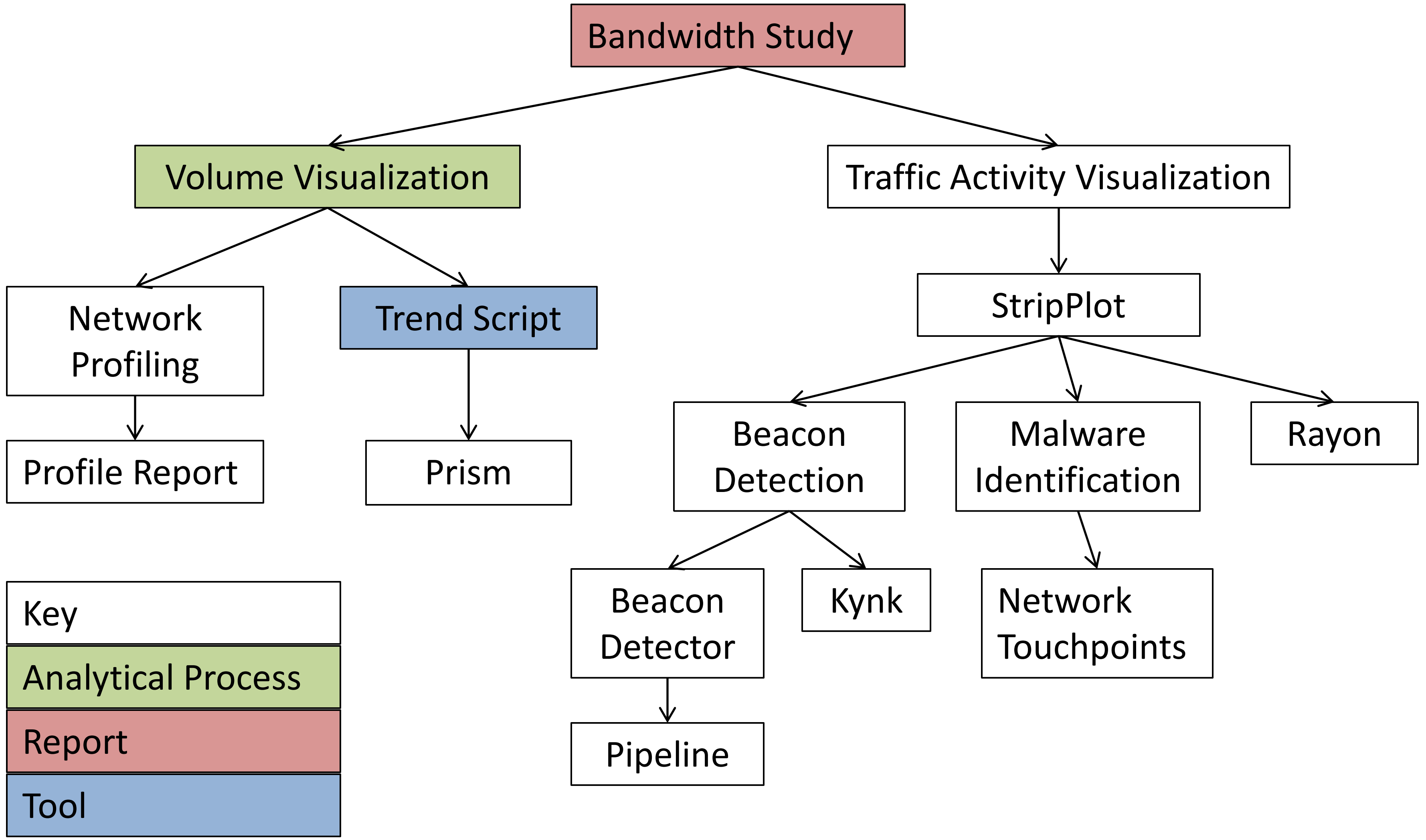
Overview


Bandwidth Study
Volume Visualization
Traffic Activity Visualization
Network Profiling
Trend Script
StripPlot
Profile Report
Prism
Beacon Detection
Malware Identification
Rayon
Beacon Detector
Kynk
Network Touchpoints
Pipeline
Key
Analytical Process
Report
Tool

An iterative Process

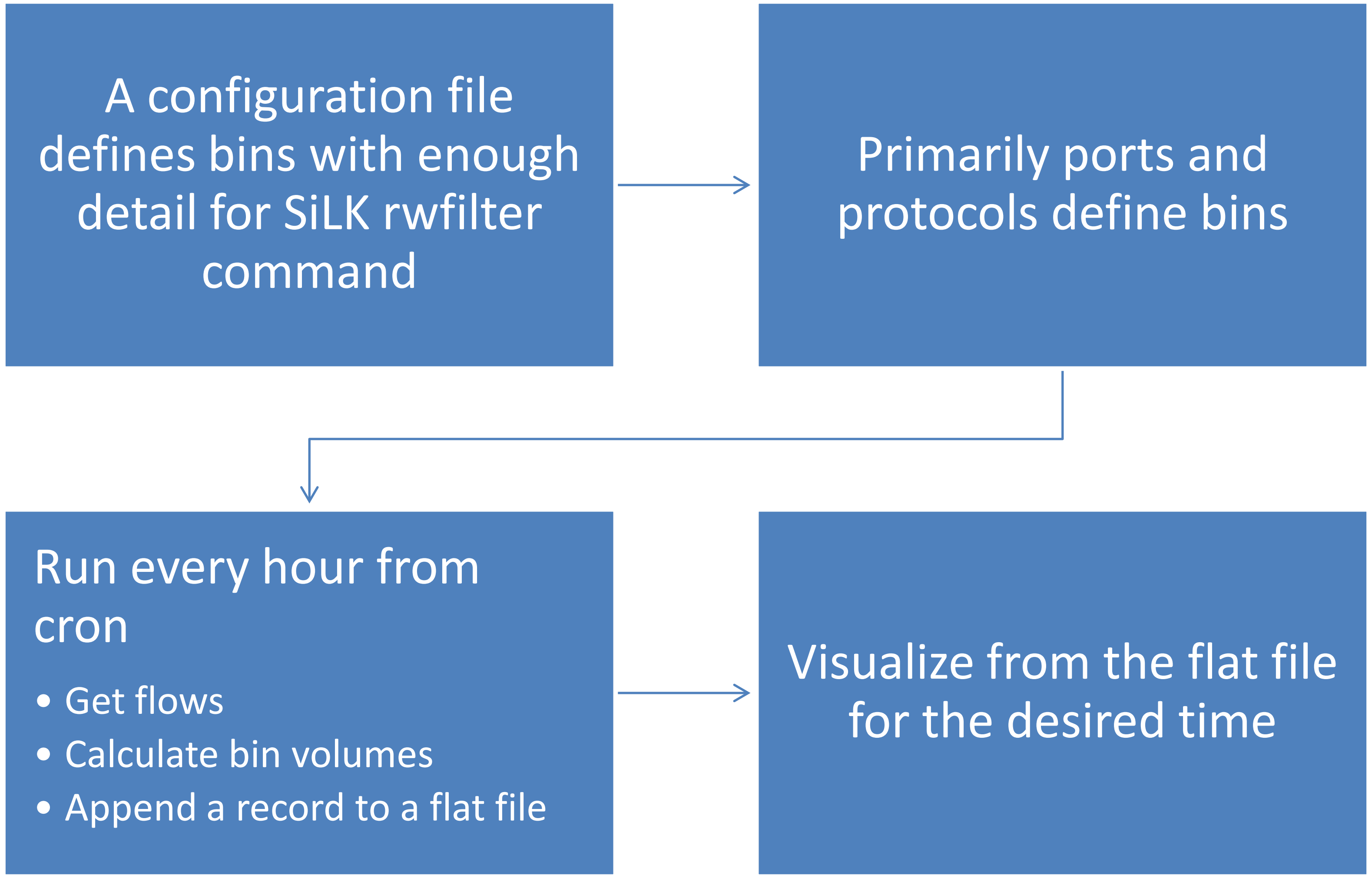

Find a large usage category, e.g. Web traffic
Split it off and look at the rest
Repeat…
Wherever you stop there are probably flow records left over

Overview



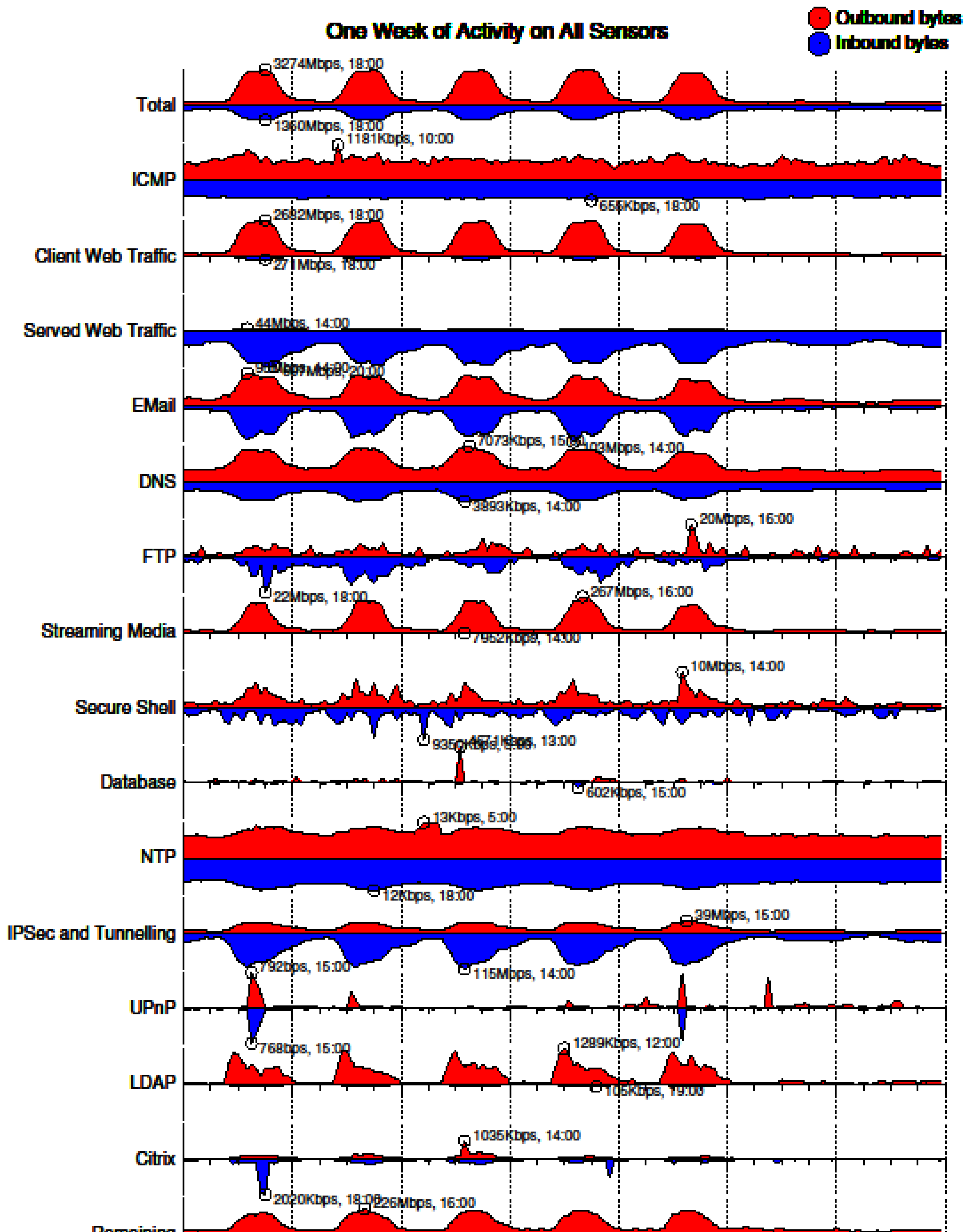
Bandwidth Study
Volume Visualization
Traffic Activity Visualization
Network Profiling
Trend Script
StripPlot
Profile Report
Prism
Beacon Detection
Malware Identification
Rayon
Beacon Detector
Kynk
Network Touchpoints
Pipeline
Key
Analytical Process
Report
Tool

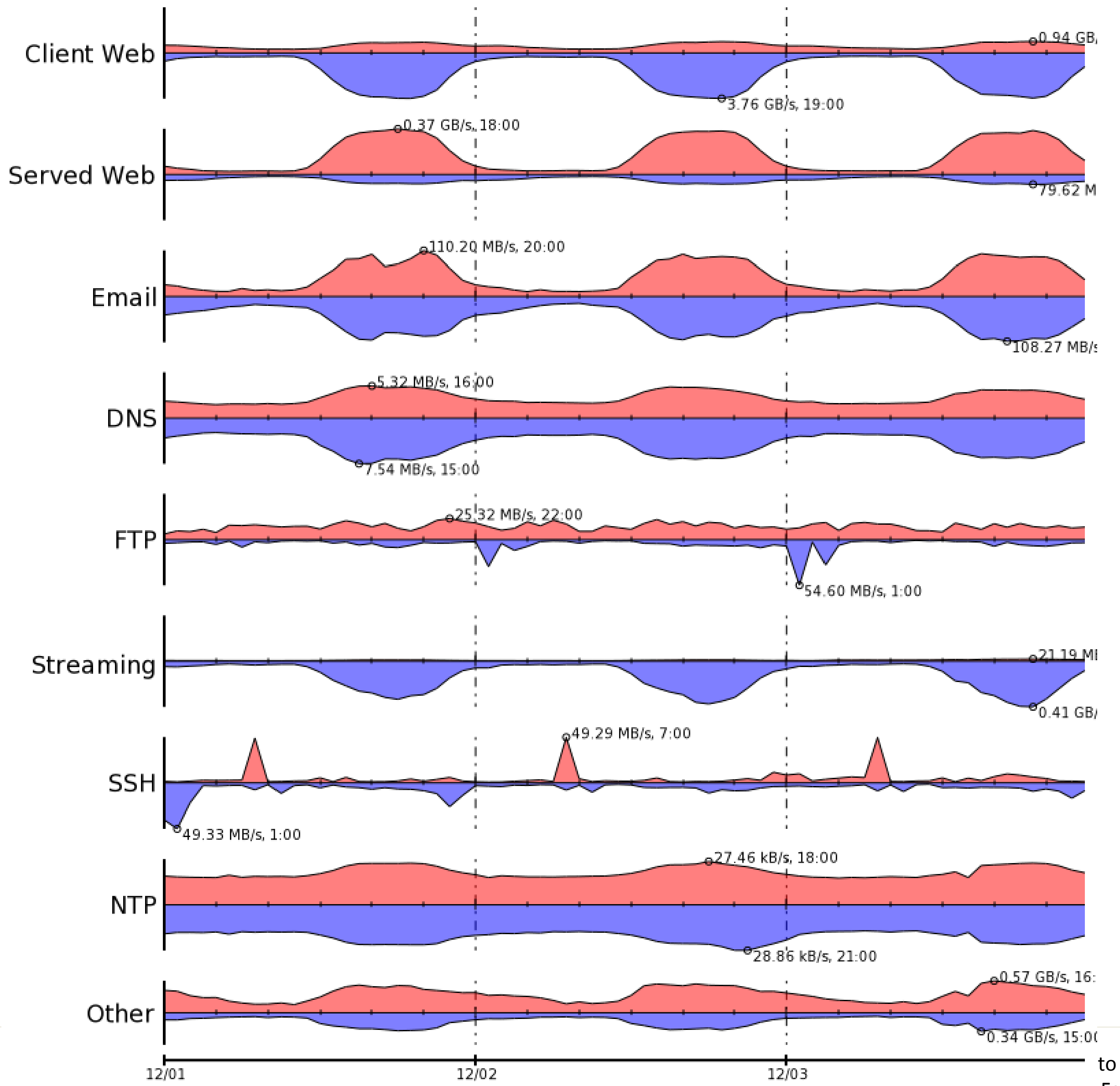
The Trend Script is born


A configuration file defines bins with enough detail for SiLK rwfilter command
Primarily ports and protocols define bins
Run every hour from cron
• Get flows
• Calculate bin volumes
• Append a record to a flat file
Visualize from the flat file for the desired time

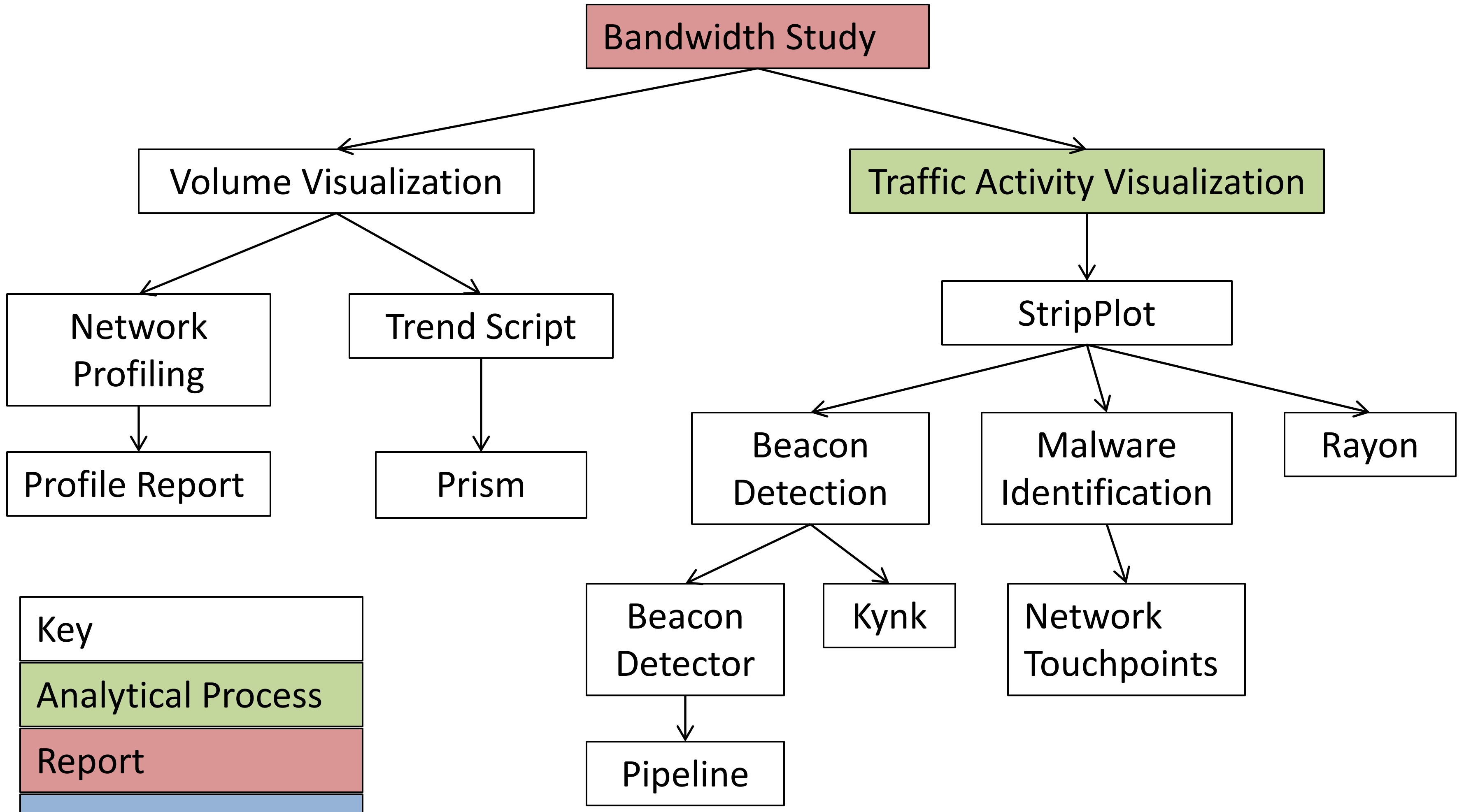
Sample Trend Script Configuration

```
[bin]
name: http-client
title: Client Web
filter: --protocol=6
out-filter: --dport=80,443,8080
in-filter: --sport=80,443,8080
```


One Week of Activity on All Sensors
Outbound bytes
Inbound bytes
Total
3274Mbps, 18:00
1360Mbps, 18:00
ICMP
1181Kbps, 10:00
656Kbps, 18:00
Client Web Traffic
2682Mbps, 18:00
271Mbps, 18:00
Served Web Traffic
44Mbps, 14:00
EMail
DNS
7073Kbps, 15:00
103Mbps, 14:00
3893Kbps, 14:00
FTP
20Mbps, 16:00
22Mbps, 18:00
Streaming Media
267Mbps, 16:00
7952Kbps, 14:00
Secure Shell
10Mbps, 14:00
Database
602Kbps, 15:00
NTP
13Kbps, 5:00
12Kbps, 18:00
IPSec and Tunnelling
39Mbps, 15:00
115Mbps, 14:00
UPnP
792bps, 15:00
768bps, 15:00
LDAP
1289Kbps, 12:00
105Kbps, 19:00
Citrix
1035Kbps, 14:00
2020Kbps, 18:00
226Mbps, 16:00
Remaining


Client Web
0.94 GB/
3.76 GB/s, 19:00
Served Web
0.37 GB/s, 18:00
79.62 M
Email
110.20 MB/s, 20:00
108.27 MB/s
DNS
5.32 MB/s, 16:00
7.54 MB/s, 15:00
FTP
25.32 MB/s, 22:00
54.60 MB/s, 1:00
Streaming
21.19 MI
0.41 GB/
SSH
49.29 MB/s, 7:00
49.33 MB/s, 1:00
NTP
27.46 kB/s, 18:00
28.86 kB/s, 21:00
Other
0.57 GB/s, 16:
0.34 GB/s, 15:00
12/01
12/02
12/03

Overview



Key
🟩 Analytical Process
🟥 Report
🟦 Tool

Traffic Activity

The activity in the bins is fairly well known

The “left over” flows, less so

What is happening “at the edge”?

Looking at flows by hand is tedious

It’s hard to program looking for the unknown

That means, it’s time for…

Flow Activity Visualization

We want to find "interesting activity"

But interesting means different things to different people

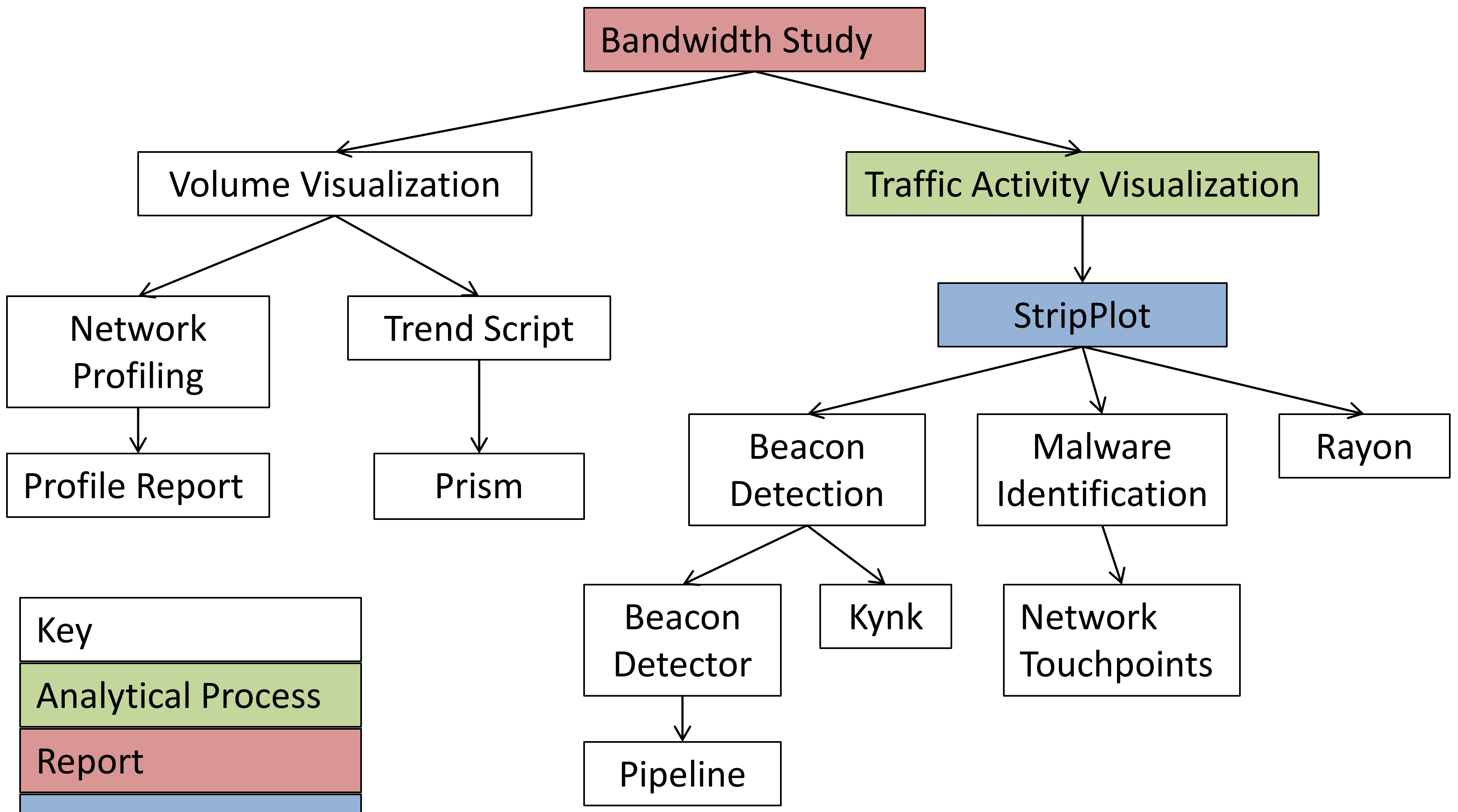
- "May you live in interesting times."
  Chinese Curse
- "Only accurate rifles are interesting."
  Colonel Townsend Whelen
- "The only interesting answers are those that destroy the questions."
  Susan Sontag

Flow Activity Visualization

Goal: produce a self-maintaining network profile

- Categorize and display activity
  - Stuff we know about: Email, Web, DNS…
  - And everything else
- Need a mechanism to permit the analyst to examine "everything else" aka leftovers
- Too bad about the "self-maintaining" part

Overview



Key
🟩 Analytical Process
🟥 Report
🟦 Tool

StripPlot “enables the eyeball”

Get a good idea of what a particular IP addressing is doing

See how a port is used

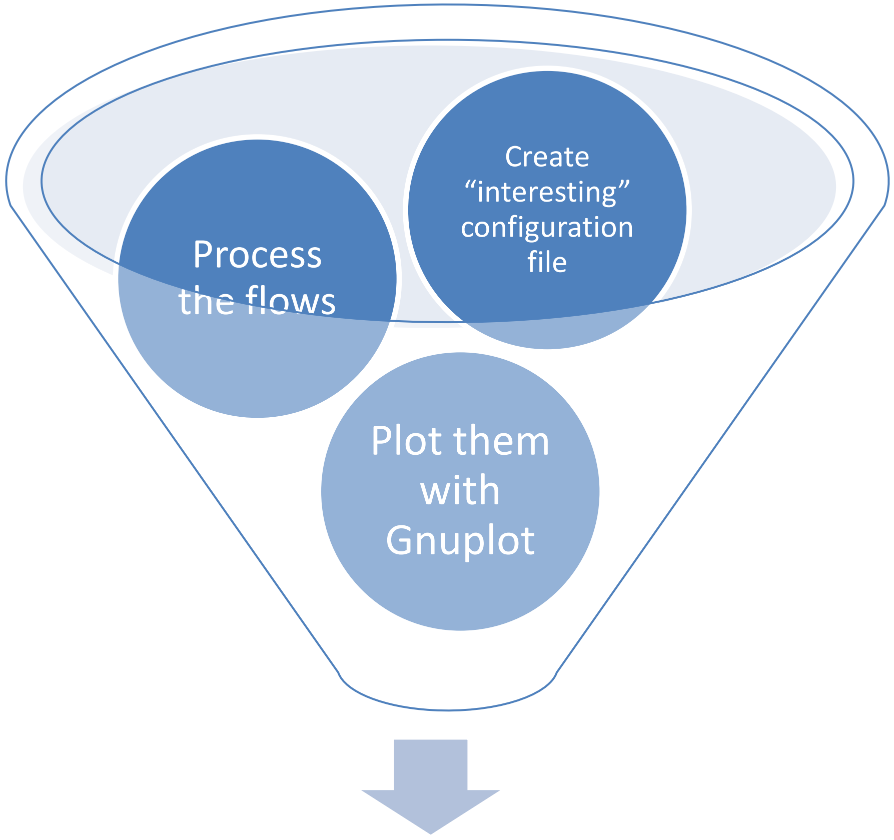
Streaming video and audio are immediately apparent

Make Beacons stand out

For more information on StripPlot see:

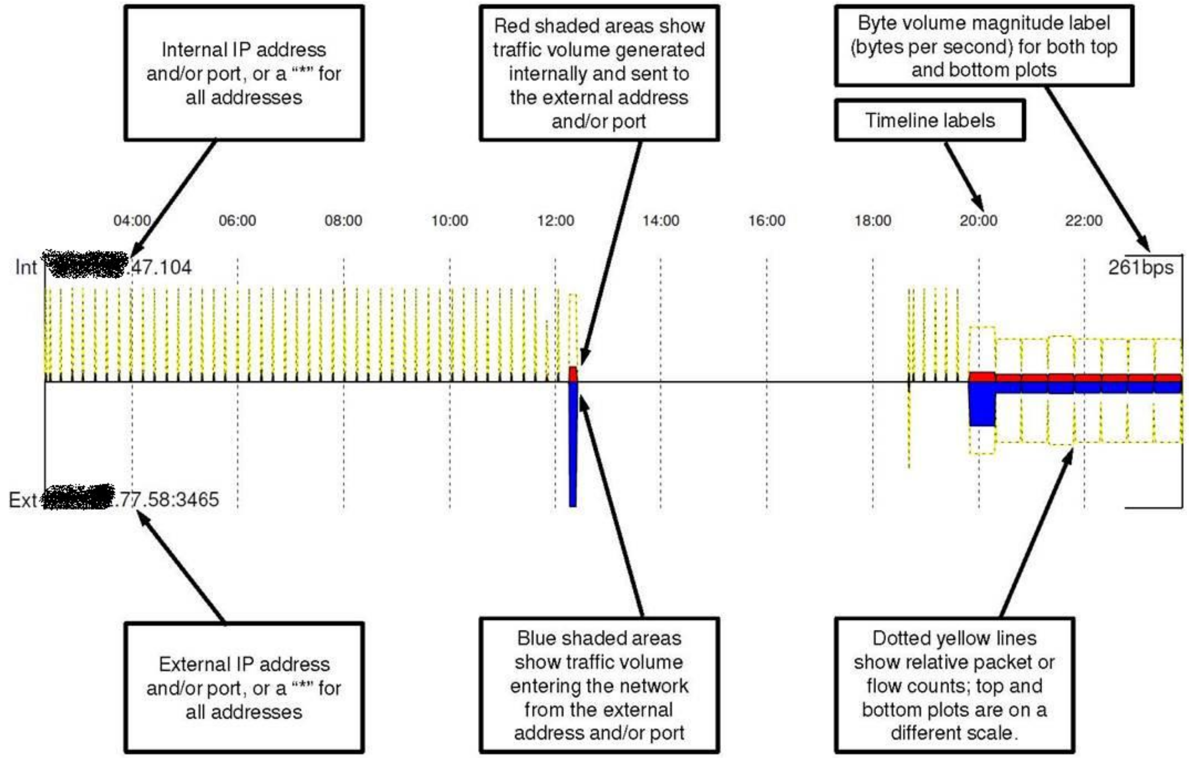
- http://www.cert.org/flocon/2010/presentations/Faber_StripPlots.pdf

The StripPlot Process



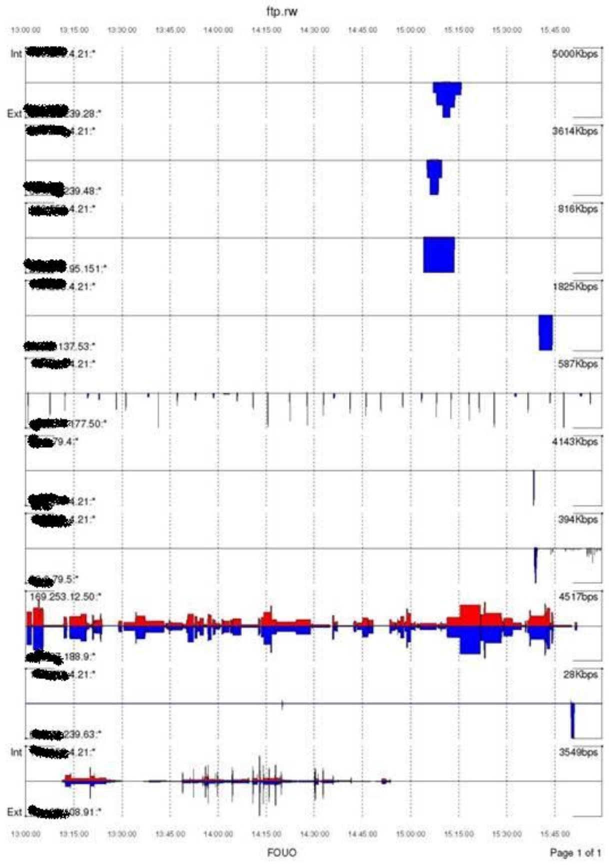
StripPlot Graphic

How to Interpret StripPlot

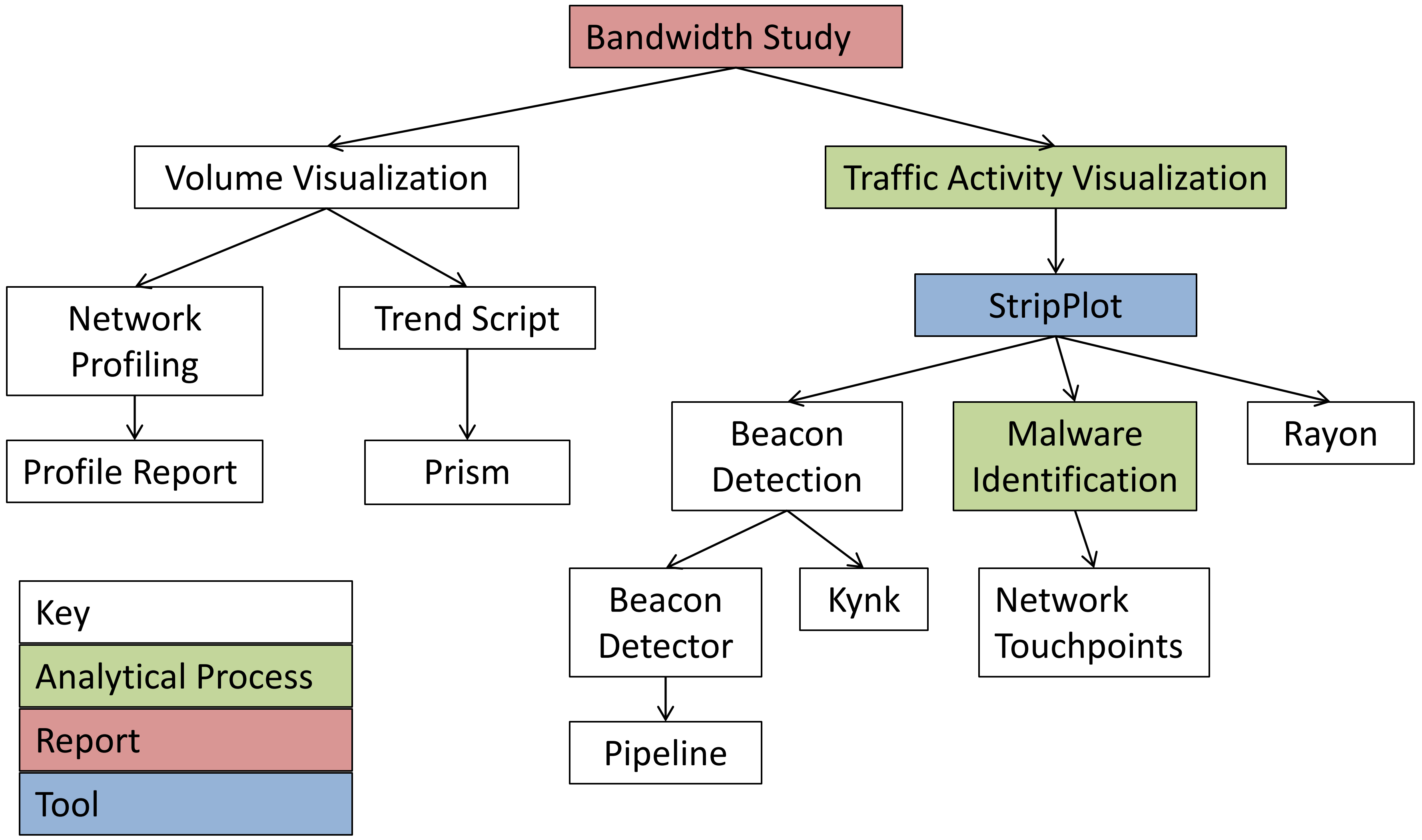


Internal IP address and/or port, or a "*" for all addresses
Red shaded areas show traffic volume generated internally and sent to the external address and/or port
Byte volume magnitude label (bytes per second) for both top and bottom plots
Timeline labels
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
20:00
22:00
Int [redacted].47.104
261bps
Ext [redacted].77.58:3465
External IP address and/or port, or a "*" for all addresses
Blue shaded areas show traffic volume entering the network from the external address and/or port
Dotted yellow lines show relative packet or flow counts; top and bottom plots are on a different scale.

Sample StripPlot



Overview


Bandwidth Study
Volume Visualization
Traffic Activity Visualization
Network Profiling
Trend Script
StripPlot
Profile Report
Prism
Beacon Detection
Malware Identification
Rayon
Beacon Detector
Kynk
Network Touchpoints
Pipeline
Key
Analytical Process
Report
Tool

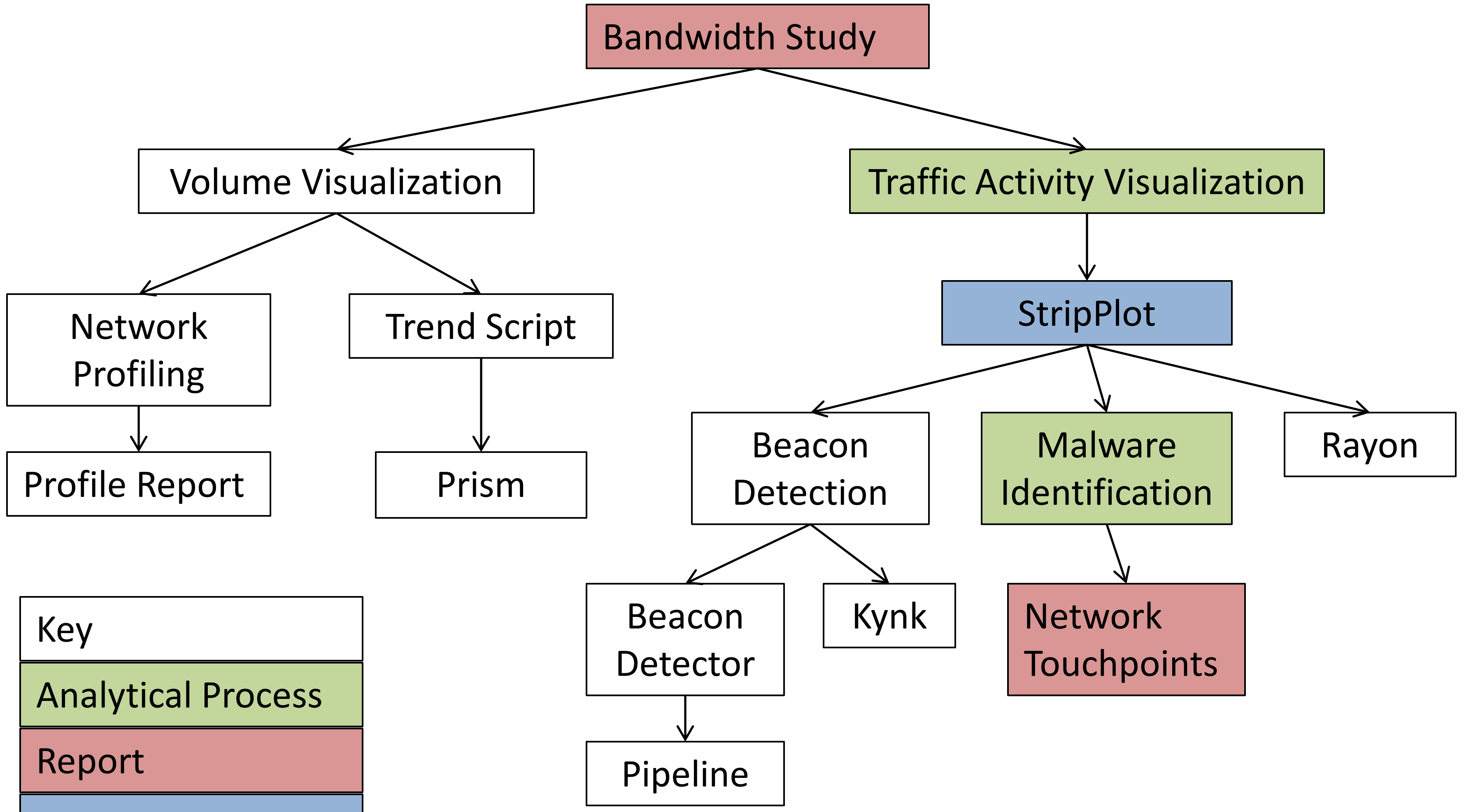
Finding Malicious Activity

Malware Team to NetSA Analysis Team:

- “You might find this interesting”

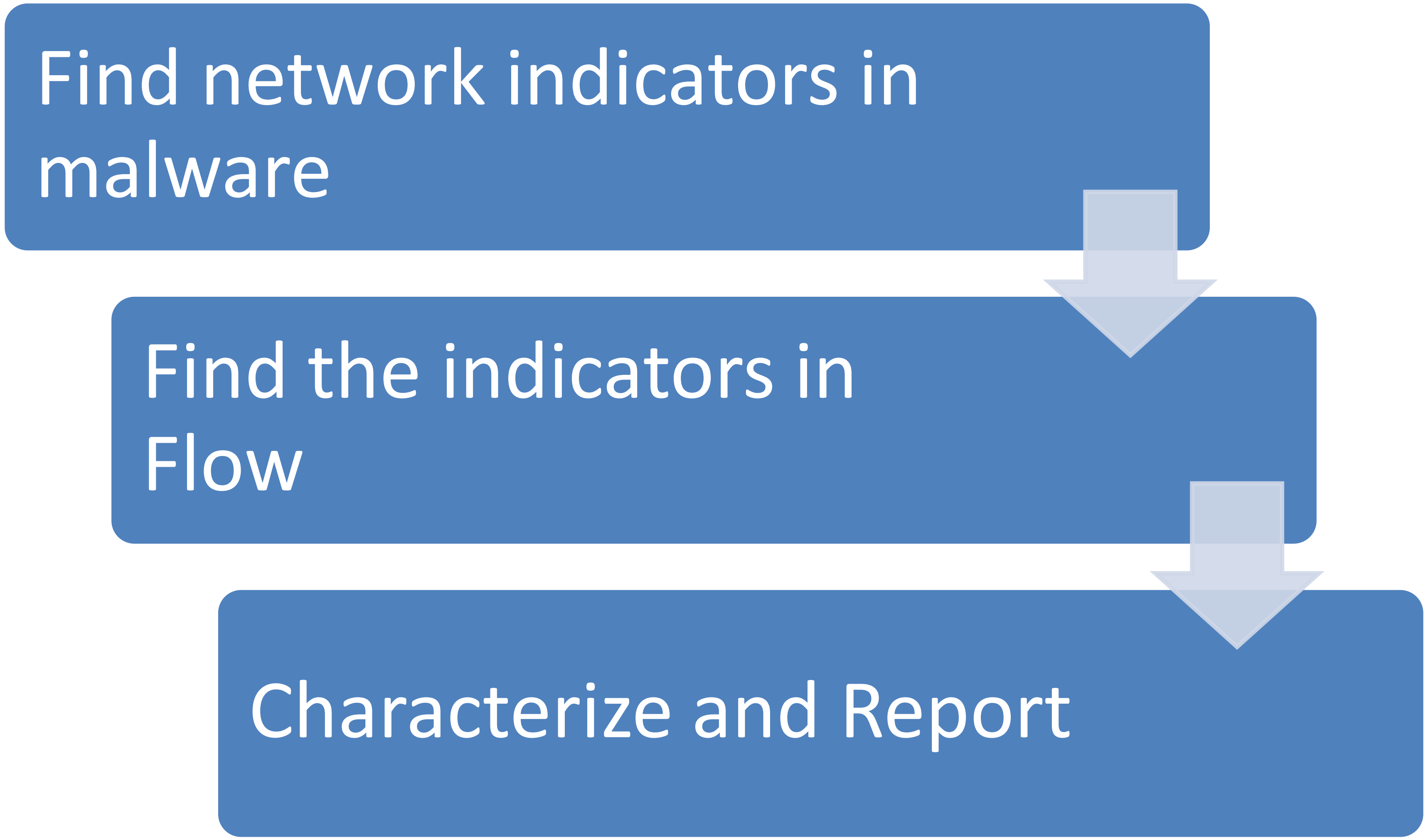
The visualization in StripPlot made it easy to spot the interesting behavior

Overview

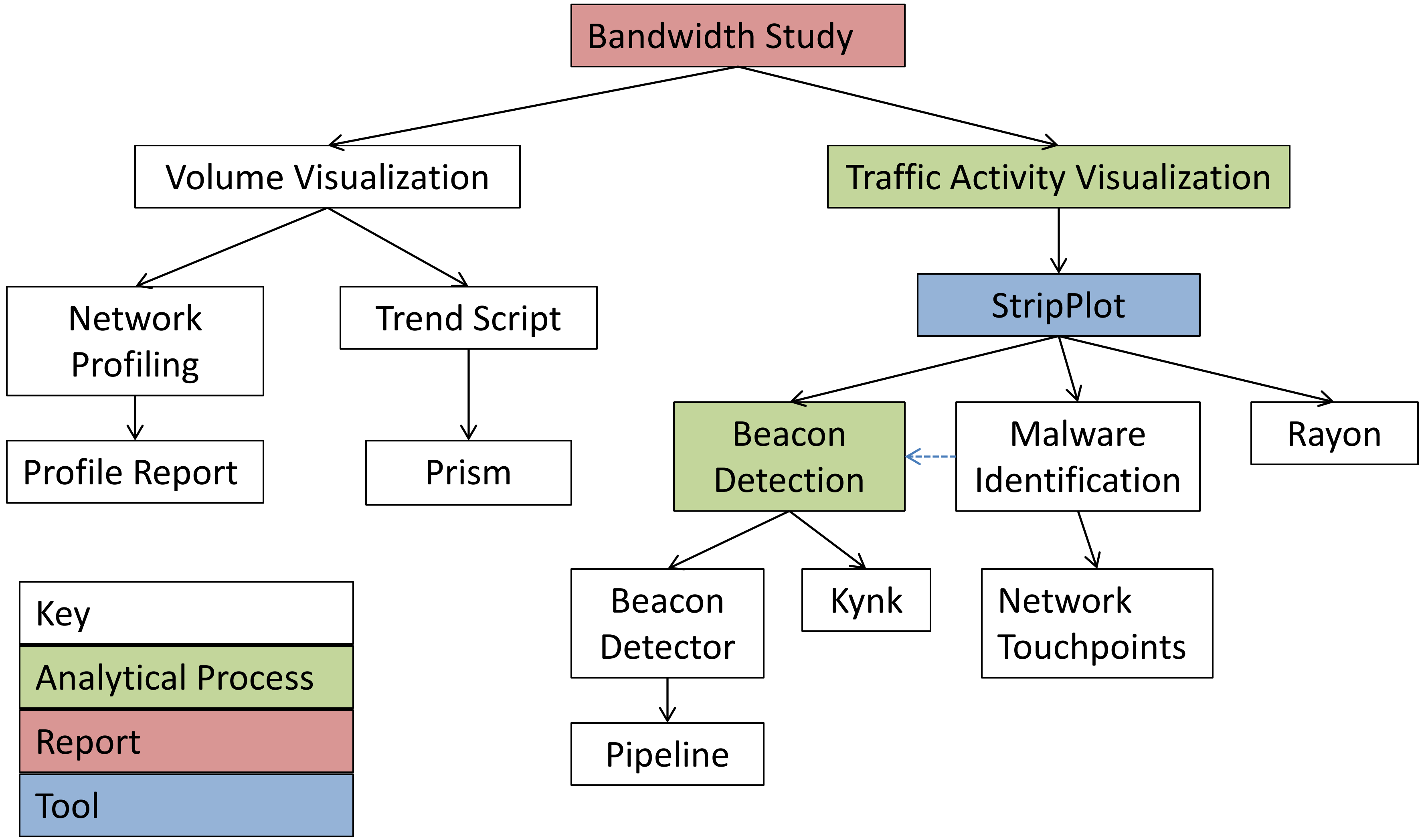


Key
🟩 Analytical Process
🟥 Report
🟦 Tool

Spin off the Network Touchpoints Project


Find network indicators in malware
Find the indicators in Flow
Characterize and Report

Overview


Bandwidth Study
Volume Visualization
Traffic Activity Visualization
Network Profiling
Trend Script
StripPlot
Profile Report
Prism
Beacon Detection
Malware Identification
Rayon
Beacon Detector
Kynk
Network Touchpoints
Pipeline
Key
Analytical Process
Report
Tool

Beacon Detection

StripPlot “enabled the eyeball” to see botnet nodes phoning home

We even saw a handoff from one C2 host to another

Beacon detection attempts to “replace the eyeball”

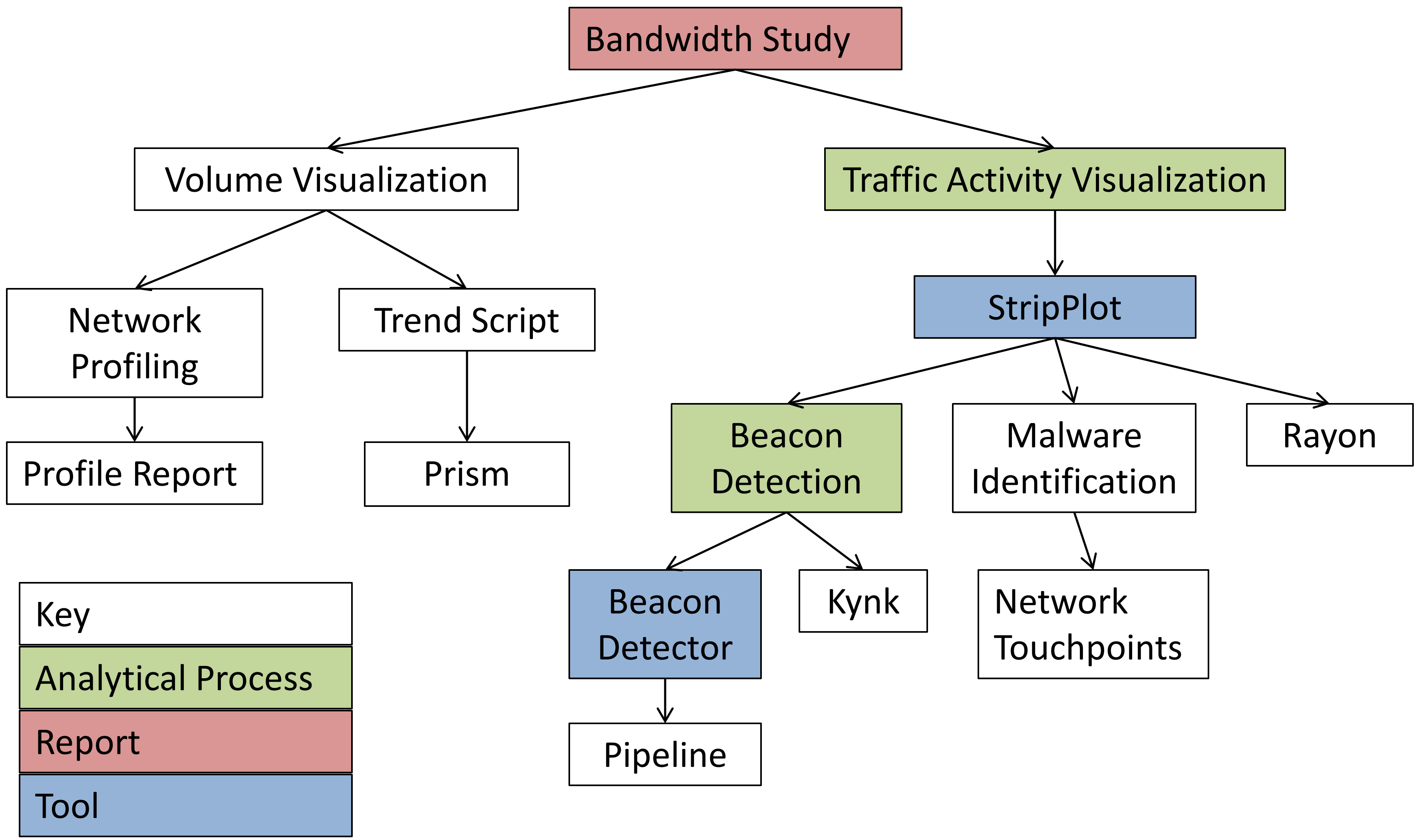
Beacon Detection

So… if we can find beacons we can find botnets, Right?

Yes, if you can distinguish a beacon from other regular behavior

Which is hard

Overview


Bandwidth Study
Volume Visualization
Traffic Activity Visualization
Network Profiling
Trend Script
StripPlot
Profile Report
Prism
Beacon Detection
Malware Identification
Rayon
Beacon Detector
Kynk
Network Touchpoints
Pipeline
Key
Analytical Process
Report
Tool

Paul's Beacon Detector

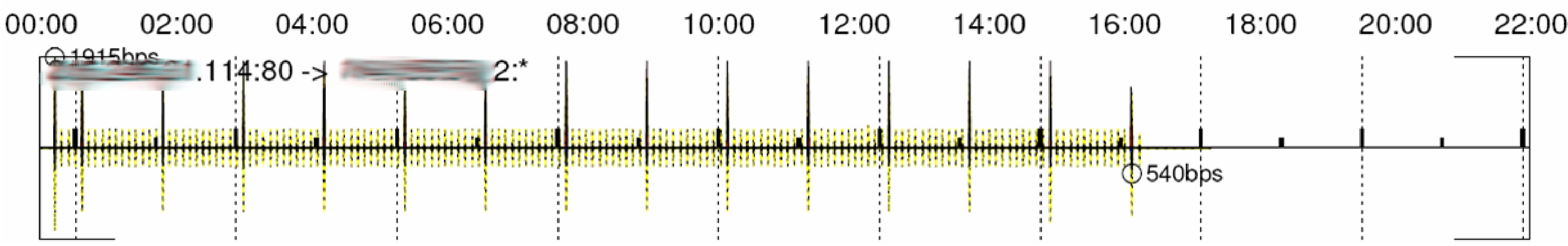
Beacons exhibit regular behavior

- A series of connections or connection attempts
- Between the same two IP addresses
- At regular time intervals

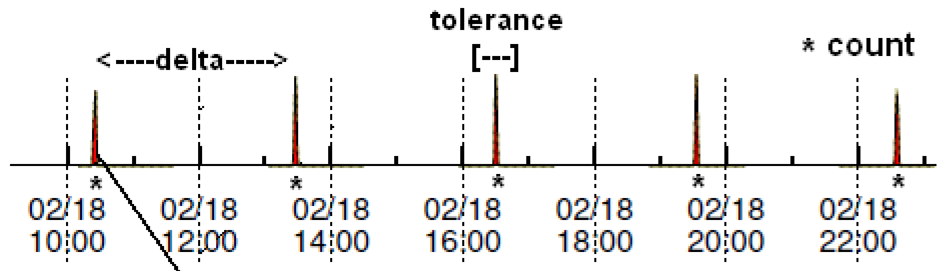
Implemented a Finite State Machine to find

- X or more flows (5 flows)
- At regular interval of Y (Y>= 5 minutes)
- With a tolerance of Z percent (5%)

Beacon Detection



Characterizing Beaconing Activity



Traffic characteristics
-- protocol
-- TCP flags
-- bytes
-- duration

Did it work?

Did it find regular behavior?

- Yes, rather a lot of it

Did it find botnet beacons?

- Probably but hard to distinguish from all the other stuff

What other stuff?

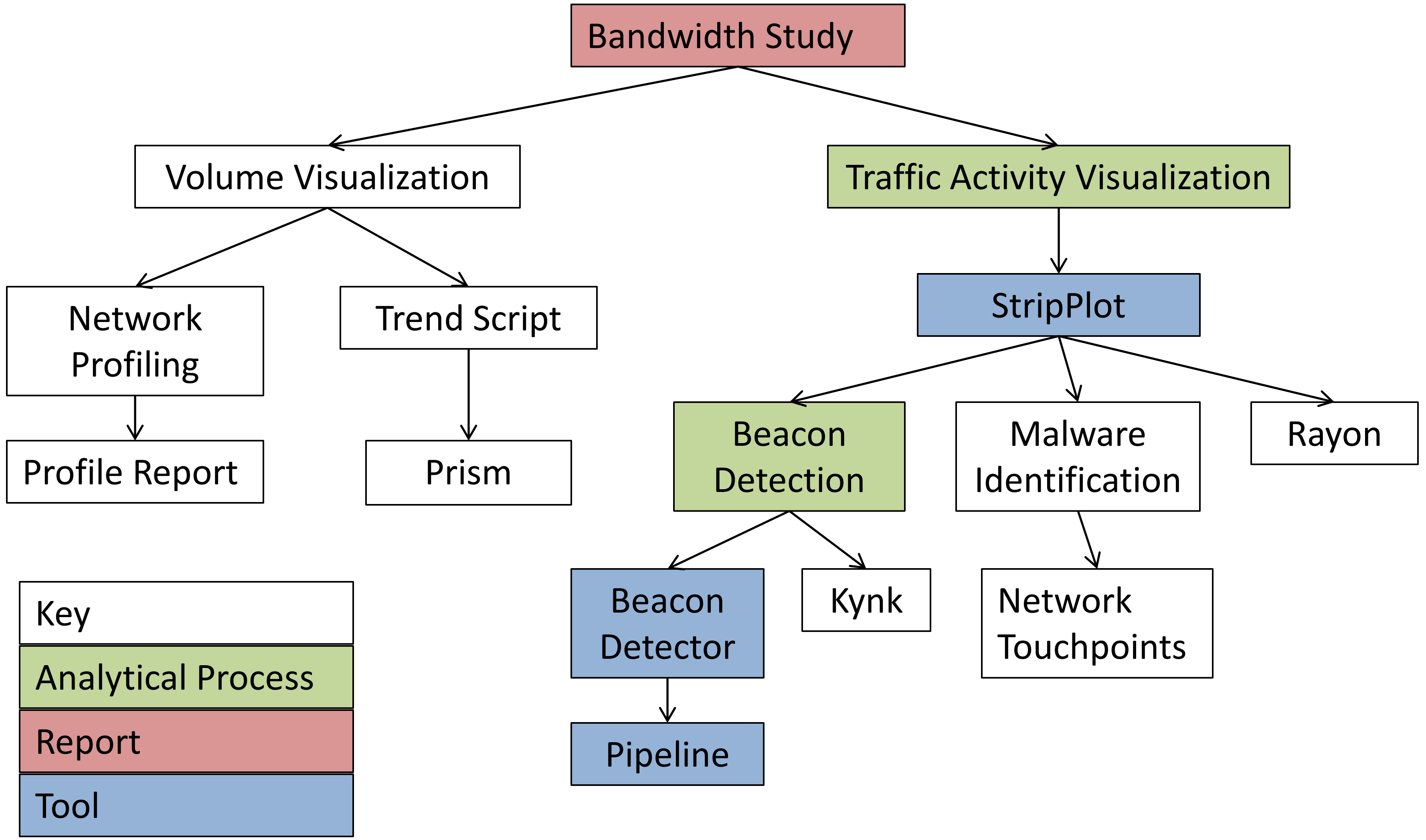
- NTP, News updates, email updates, DNS…

Can it be made better?

Three ways that we know of

- Find more regular behavior
  - Missing flows
- Additional information
  - Actual botnet beacon characteristics
  - Any other information that can be used with flow analysis
- Extreme whitelisting
  - Keep track of <u>everything</u> that beacons, and ignore it
  - Only look for new stuff
    - Keep track of the beaconing addresses for the last 30 days
    - Whitelist them

Overview



Bandwidth Study
Volume Visualization
Traffic Activity Visualization
Network Profiling
Trend Script
StripPlot
Profile Report
Prism
Beacon Detection
Malware Identification
Rayon
Beacon Detector
Kynk
Network Touchpoints
Pipeline
Key
Analytical Process
Report
Tool

Get results sooner

Traditional SiLK commands find flows in the repository

To get the most recent, set the search time and run it in cron, but how often

- Run cron too often and one doesn’t finish before the next one starts
- Run it less often and you wait longer than necessary

We want to look at flows as soon as they are available

Pipeline fills that role

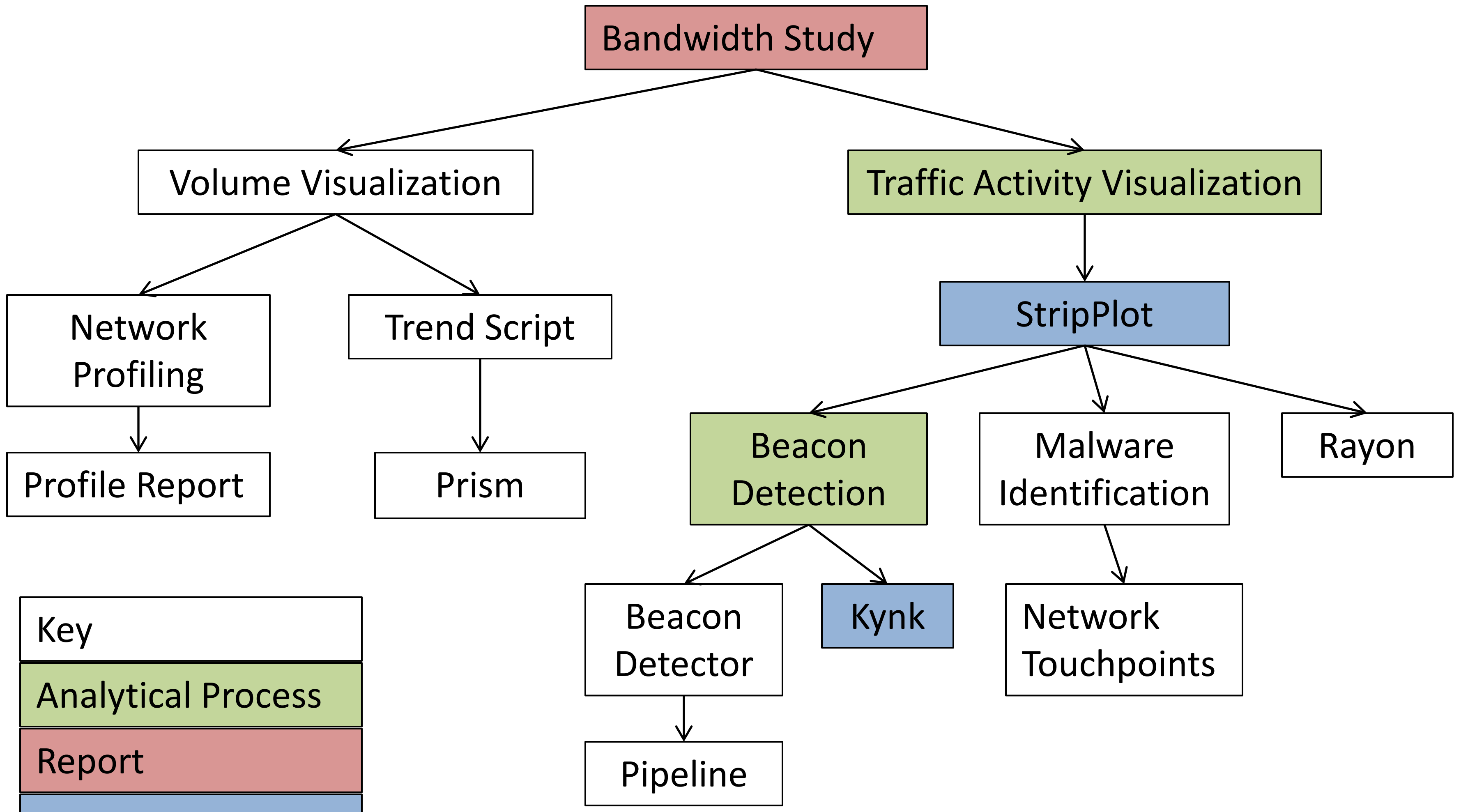
Pipeline runs continuously and processes SiLK files as they are written

Pipeline has its own unique filtering strategy

Paul’s Finite State Machine was implemented in Pipeline

It will alert as beacons (instances of regular behavior) are found

Overview



Key
🟩 Analytical Process
🟥 Report
🟦 Tool

Eight Different Beacon Detectors?

Motivation

- Beacon detection is either very useful or a very shiny object: I know of at least 8 implementations, 9 if you count stripplot.
- Saw beaconing in strip plots of RAT
- Recognized utility of finding beacons to detect certain RATs
- Concluded that "eye charts don't scale”
- Determined to explore algorithmic approaches

YABD[1] – Yet Another Beacon Detector?

Activities

- Explored different algorithms, implemented several
- Performed analysis of running time
- Identified common sources of false positives
- Generated RAT traffic in lab for testing
- Explored live data

[1] Biologists use YABD as an index of the health of deer in relation to carrying capacity.

From Eye Charts to…

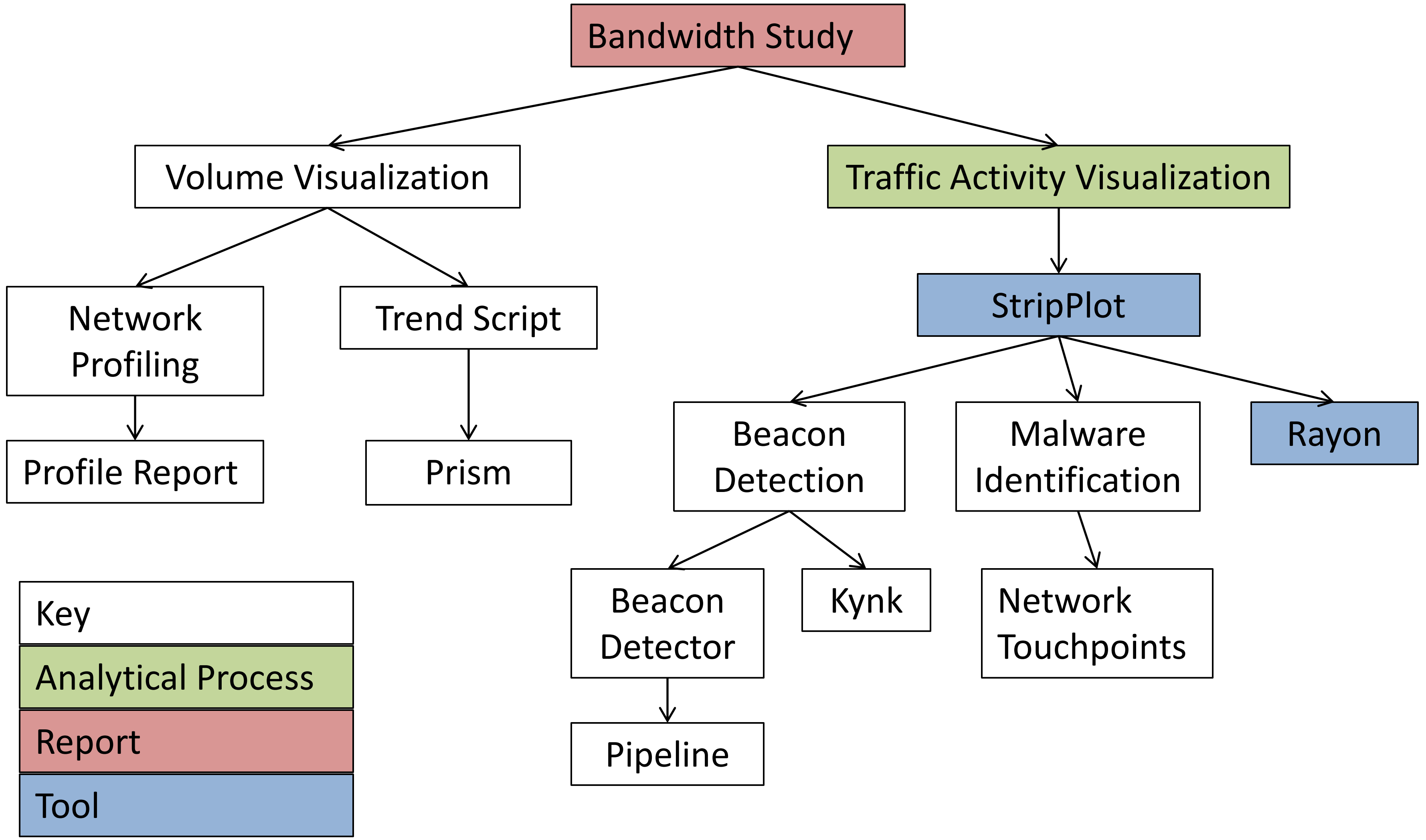
Outcomes

- Two first generation beacon detectors
- One second generation detector in pipeline
- Tools delivered to different analyst communities with mixed levels of adoption.

Lessons Learned

- Your first thought on algorithms may not be right
- You need need a large sample of ground truth to test against
- Algorithms that work on a few samples may not work in the wild.
- It's hard to generate realistic background data.
- False positives are common.
- Need to socialize more with analyst community.
- Adoption is tied to perceived utility of the tool, ownership the analysts feel of it (homegrown tools win), and their trust in the person/organization providing the tool to meet their specific needs.

Overview



Bandwidth Study
Volume Visualization
Traffic Activity Visualization
Network Profiling
Trend Script
StripPlot
Profile Report
Prism
Beacon Detection
Malware Identification
Rayon
Beacon Detector
Kynk
Network Touchpoints
Pipeline
Key
Analytical Process
Report
Tool

The Rayon Viz Library

Several analytics had visualization requirements in common

StripPlot pushed Gnuplot to its limits

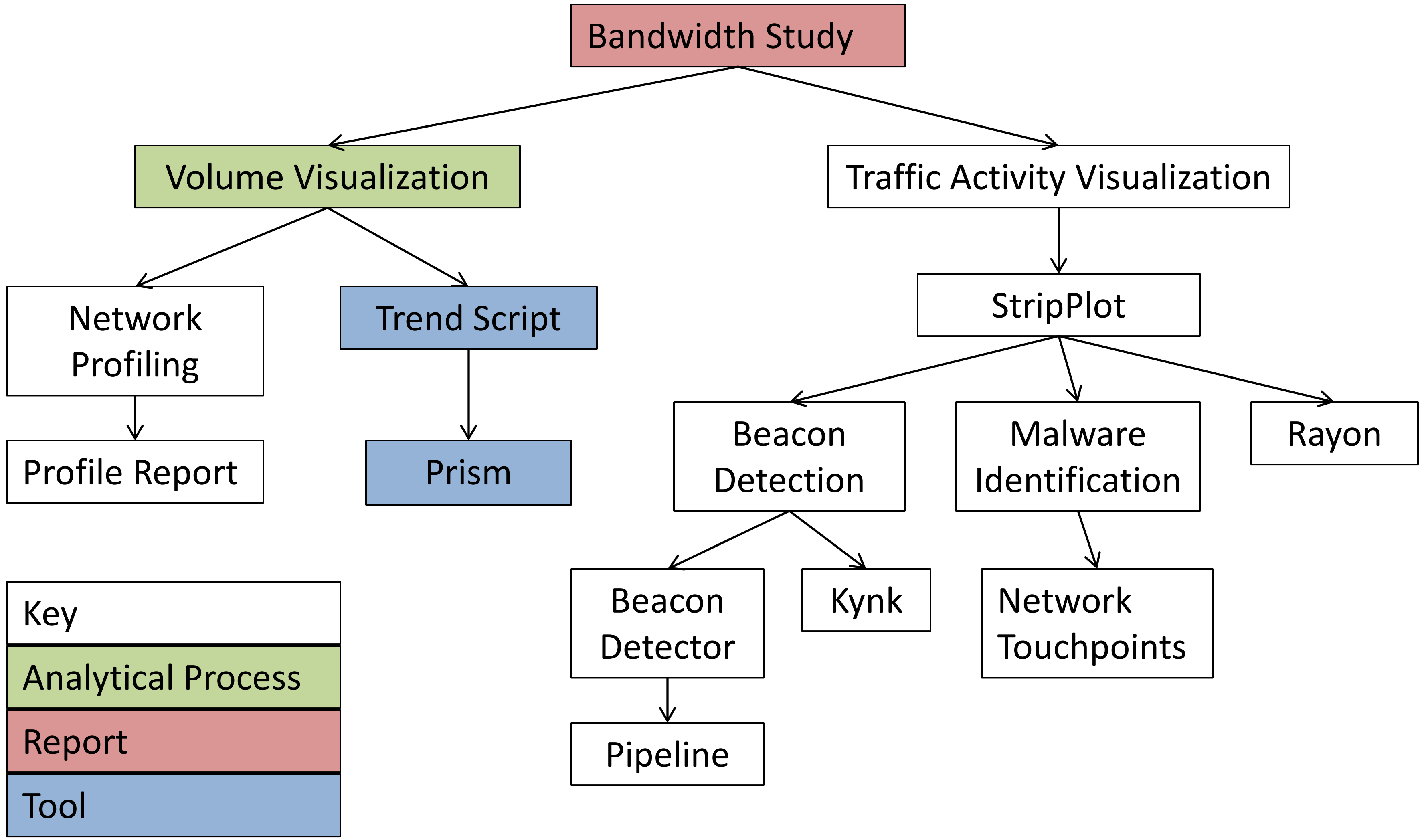
It was time to move away from “Analyst Code”

Why didn't he call it Yet Another Graphics Package

Phil Groce of the NetSA Development team

- gathered requirements
- wrote a set of "flow aware" graphics primitives
- wrote several applications using the primitives
- released it to the world as Rayon
  - http://tools.netsa.cert.org/rayon/index.html
- ask us later if you don't get the play on words

Overview



Bandwidth Study
Volume Visualization
Traffic Activity Visualization
Network Profiling
Trend Script
StripPlot
Profile Report
Prism
Beacon Detection
Malware Identification
Rayon
Beacon Detector
Kynk
Network Touchpoints
Pipeline
Key
Analytical Process
Report
Tool

Prism

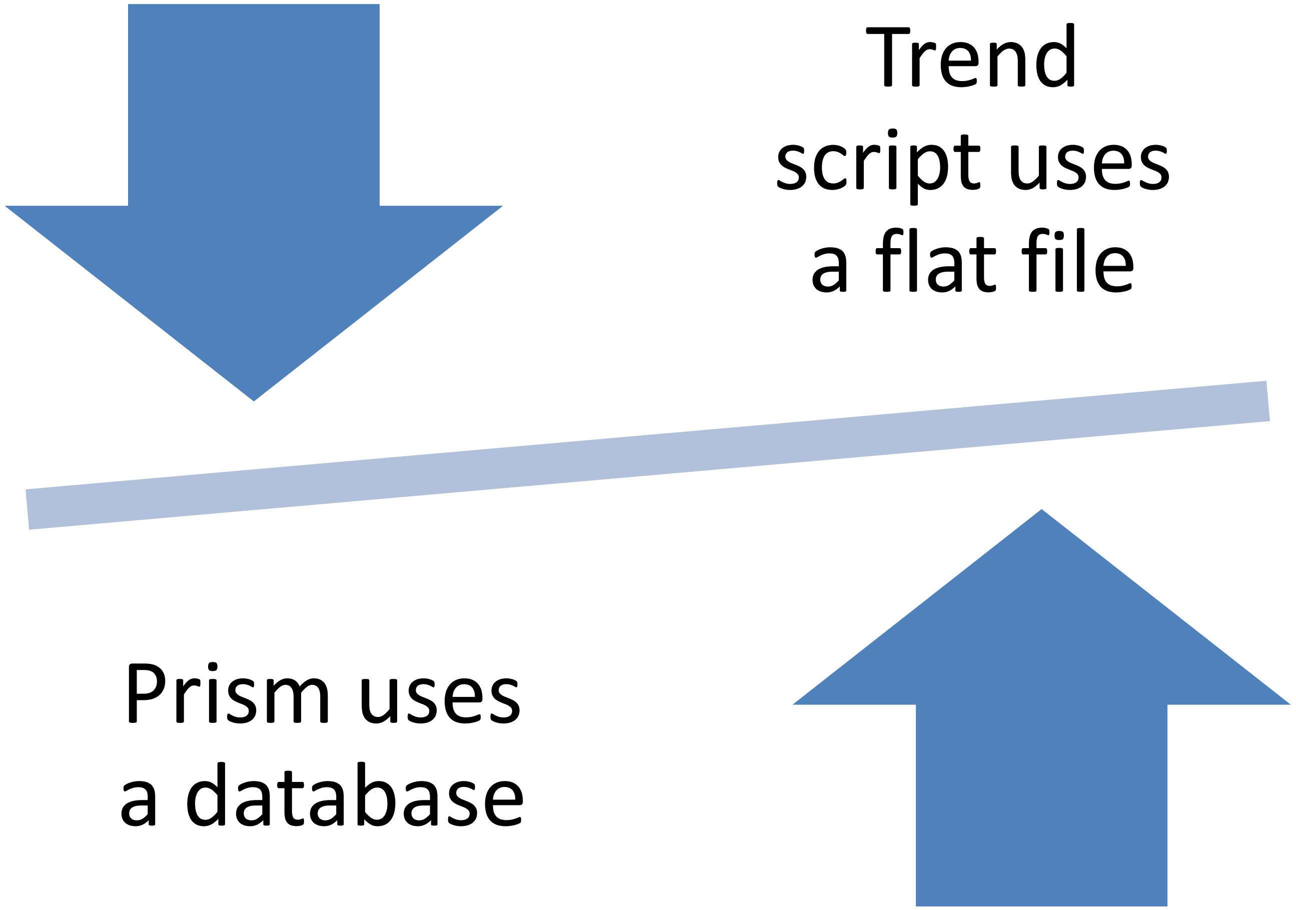
There was a renewed interest in Trend Script

But it is an analyst’s tool for specific tasks

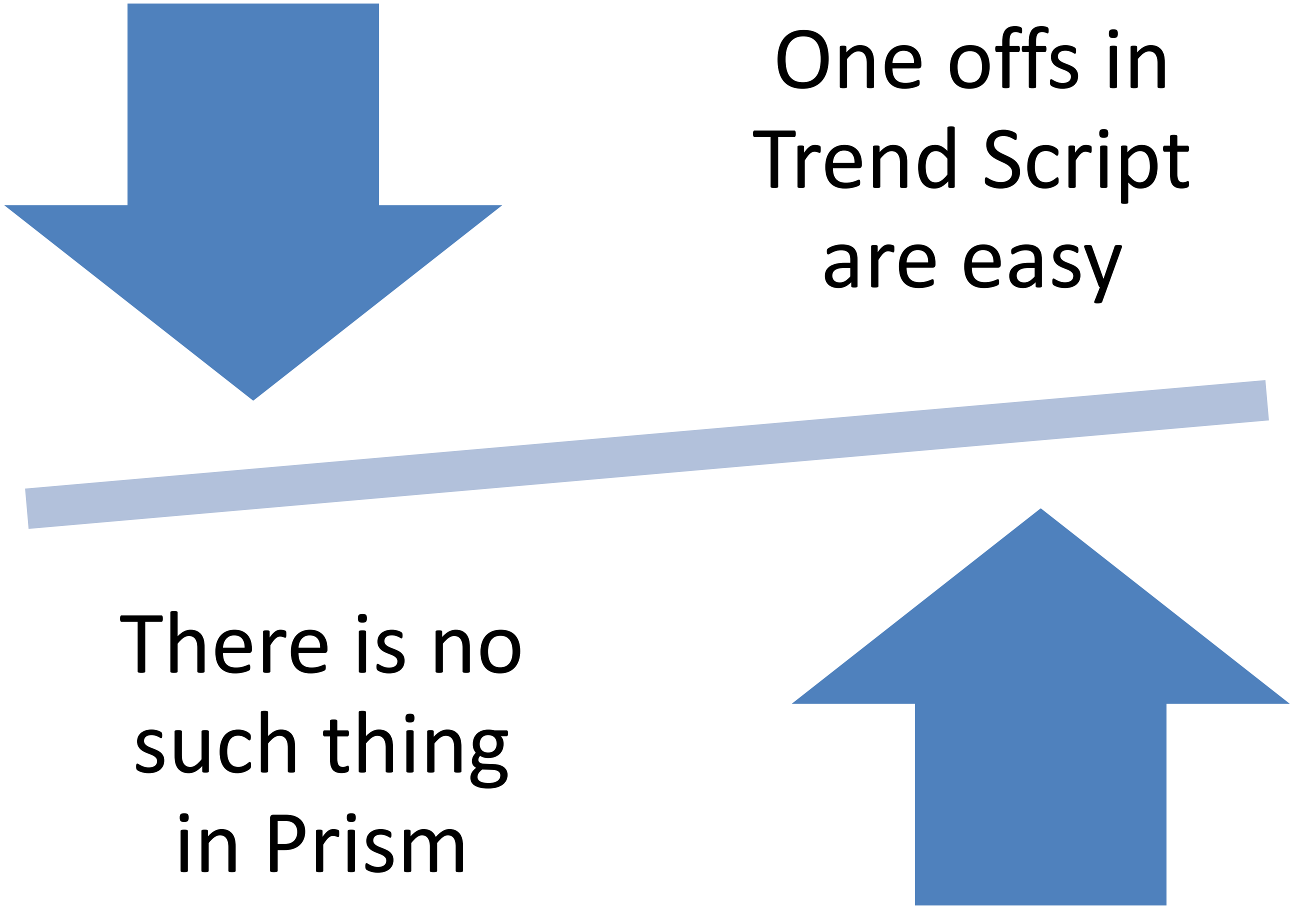
A continuous volume display has other requirements

Prism is a re-write of the Trend Script by NetSA Development Team member John Prevost

Prism vs Trend Script



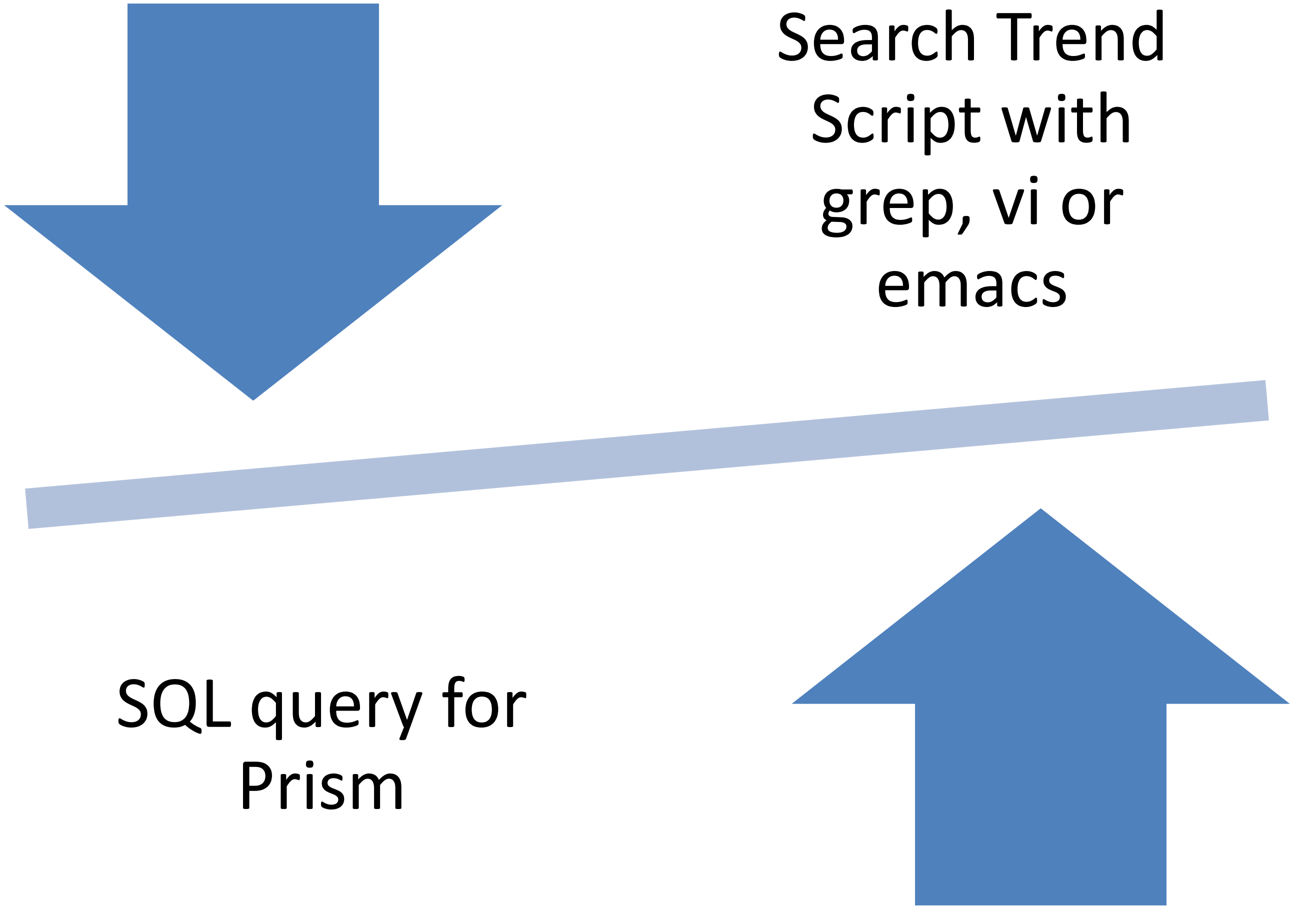
Prism vs Trend Script



One offs in Trend Script are easy

There is no such thing in Prism

Prism vs Trend Script



Search Trend Script with grep, vi or emacs

SQL query for Prism

Conclusions

One thing leads to another

“If we knew what we were doing, it wouldn’t be called research, would it?” A. Einstein

Don’t be afraid to scrap something and start over

Paul Krystosek `pnk@cert.org`
George Jones `gmj@cert.org`
Sid Faber `sfaber@cert.org`
Network Situational Awareness Group
CERT Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA