Data-Plane Accelerated Intrusion Detection Using P4

Ben Lewis
Lancaster University
Supervisors: Prof. Nicholas Race & Dr Matthew Broadbent

Lancaster University

Existing network security solutions

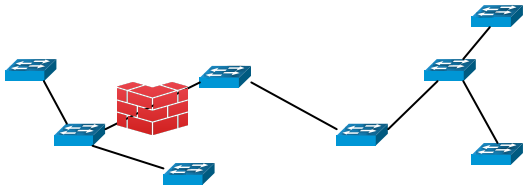
SNMP
- Network performance monitor
- Deals well with heterogeneity

IPS, IDS
- Firewalls
- Middle boxes
- Sampled and full

Host based intrusion detection
- Requires client access
- Good for servers and DevOps

Relies on an understanding of the underlying physical network

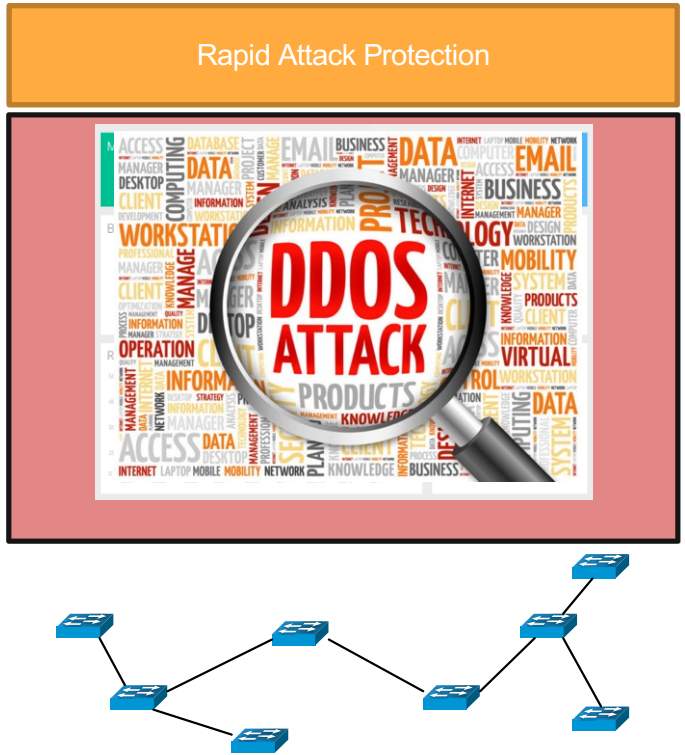
Focuses on static points in the network, with limited ability to remediate between these points



Software Defined Networks (SDN) for Monitoring

Holistic and detailed view to identify and stop wide range of attacks

- Intrusion
- DDoS, DoS
- Spoofing
- Reconnaissance
- Botnets
- New attacks



Intrusion Detection in a Software-Defined World

Typical challenges

- Placement
  - How do you link the underlying physical and logical networks
- Scale
  - How do you scale intrusion detection to bigger, faster, topologically diverse networks?
  - How do SDN controllers scale to manage larger networks?

P4 allows us to control forwarding logic

- How can we harness this?

Expressing IDS rules in P4

Can we take a published ruleset for Snort/Suricata and push that into a P4 table?

First task: write a parser and determine what can be represented in P4

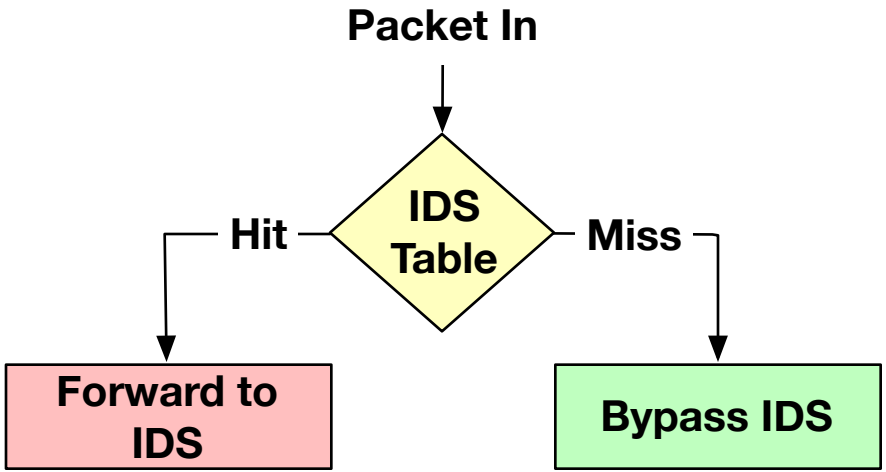
IDS Rule Parser Example

By example: how many rules can take from a published set and represent in P4?

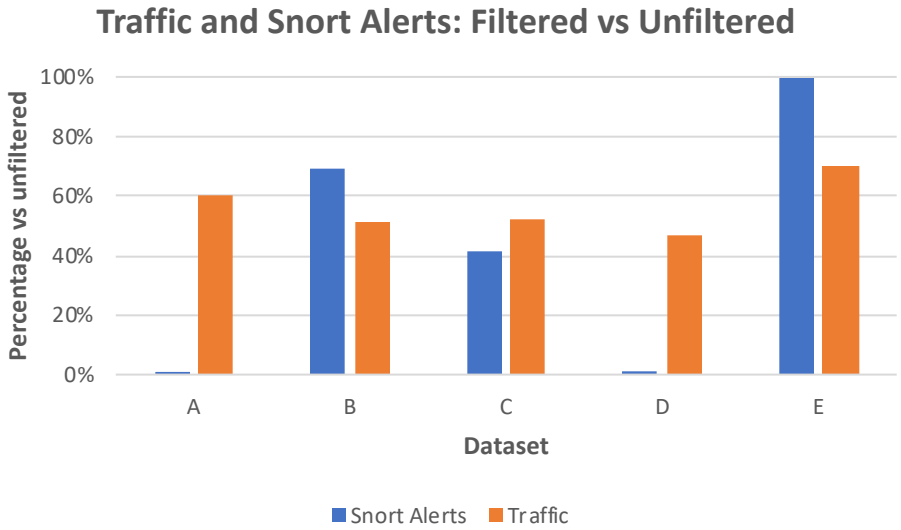
Set: Snort Community
Rules: 971
Un-usable: 63 (condition cannot be represented in P4)
In table entries after de-duplication: 337

P4 Forwarding Logic

- Integrated as part of a layer 2/3 switch
    - Bypassing the IDS forwards via normal forwarding tables
- Sending via IDS can either use a dedicated port or encapsulate traffic for transit then inspection



Initial Results with Proactive Filtering and BMV2



Dataset key:
A: Baseline/Normal Traffic
B: FTP & SSH Bruteforce
C: DoS/DDoS
D: Web Attacks
E: Infiltration & Botnet Traffic

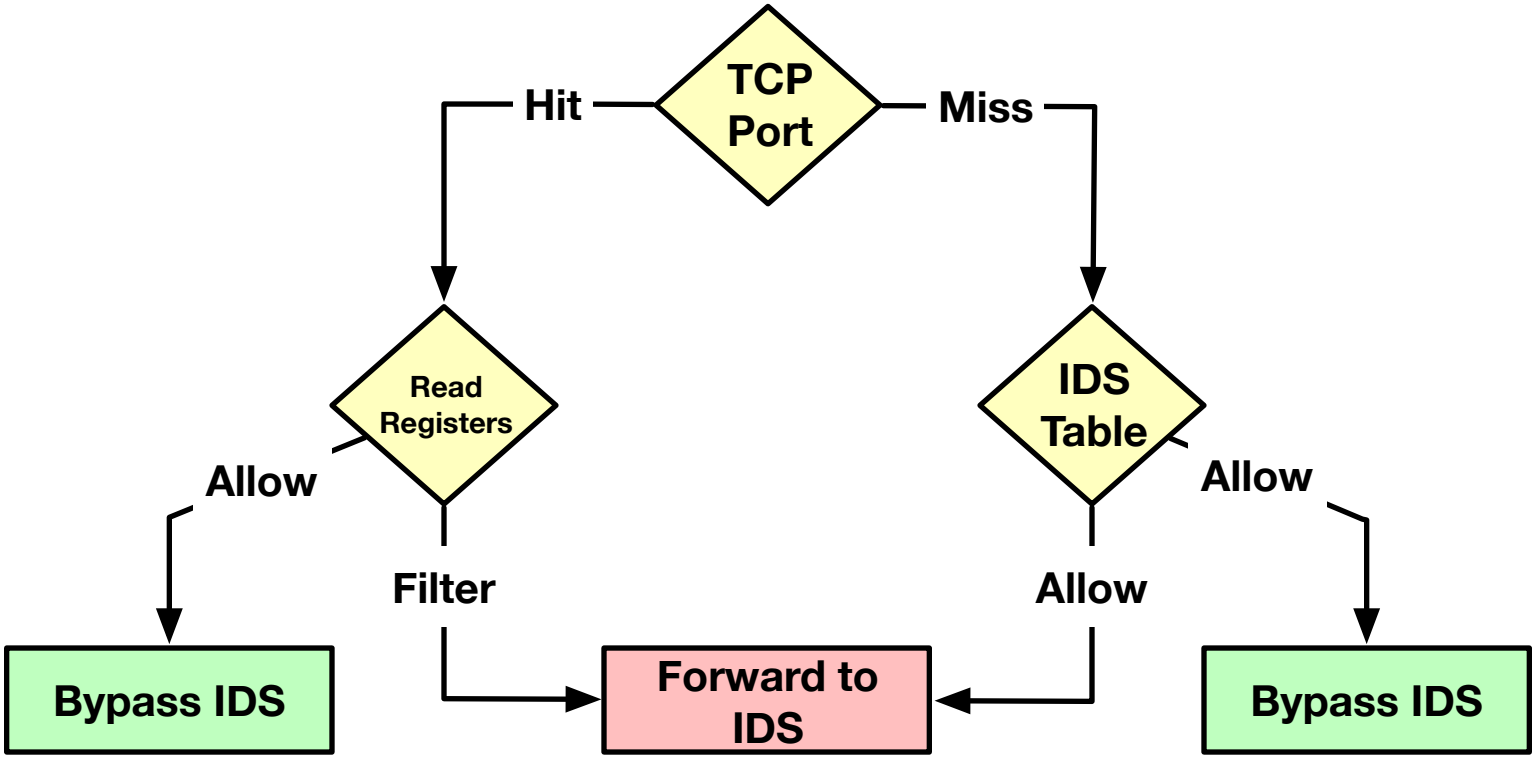
Making Packet Filtering Stateful

P4 registers offer read/write access to state during packet processing

We use 2 sets of registers:

- to track packet counts for a given flow
- to track packet timestamps, the flow counters have adjustable timeouts

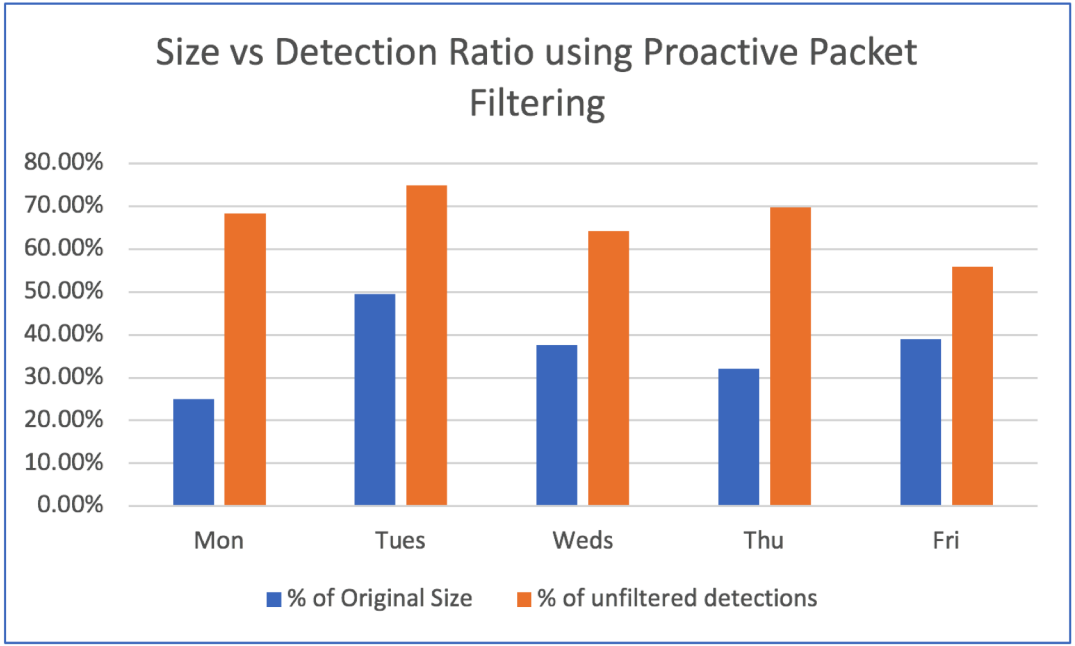
Making Packet Filtering Stateful 2



Results with stateful processing

We show up to 75% of signatures detected, with only 50% of traffic

In the worst case, we get ~55% of detected signatures, with only ~38% of the traffic. We want to improve on this.



Dataset key:
A: Baseline/Normal Traffic
B: FTP & SSH Bruteforce
C: DoS/DDoS
D: Web Attacks
E: Infiltration & Botnet Traffic

Feedback from the IDS

IDS identifies and tracks flows of interest

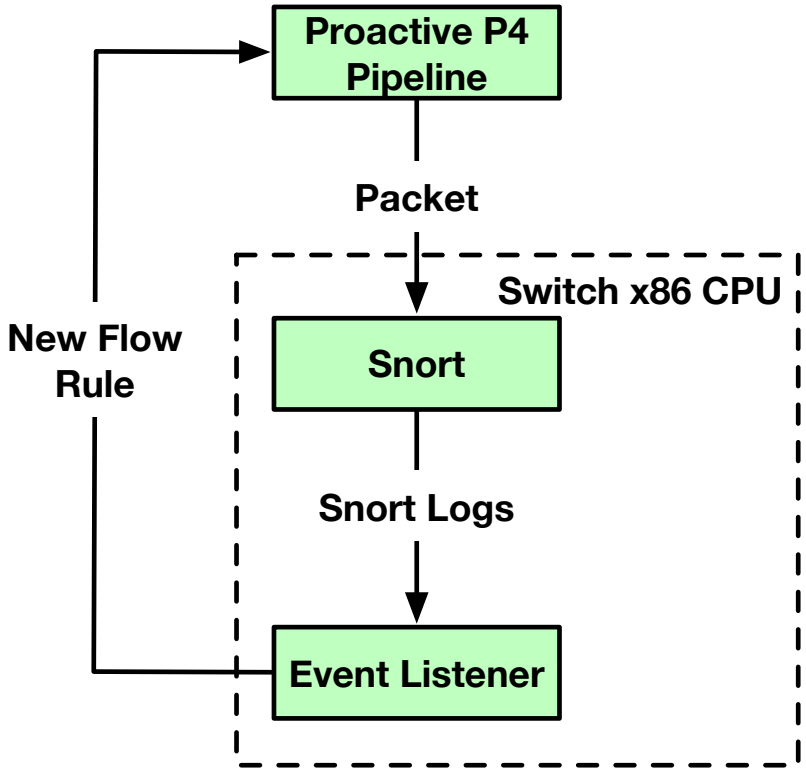
Can we use this tracking and push it back into the network?

How would we get Snort to start generating flow rules?

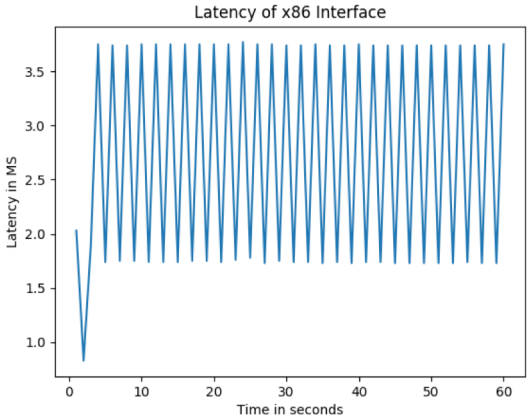
Having Snort feedback to the IDS

Reactive response to alerts generated by security applications

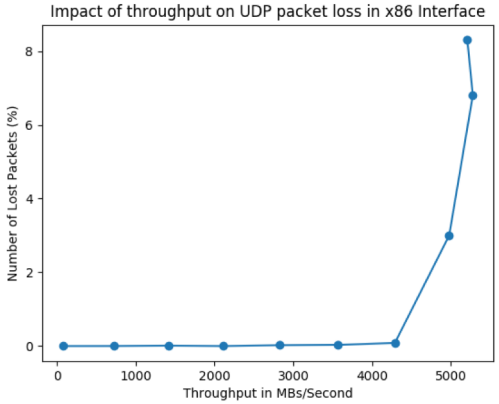
- Used to feedback to the proactive P4 pipeline
  - Update pipeline packet filter



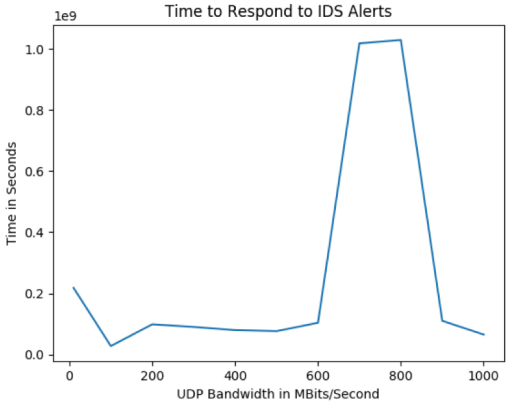
Preliminary Reactive Results



Low Latency to CPU between 0.5 and 3.8 MS



Low Packet Loss Up to 4GBs/Second



Quick Response to Security Alerts < 0.3ms at low-mid bandwidth

*Based on our implementation on a pre-production Barefoot Tofino Switch

Future Work

SDN controller integration

Application Layer Detection

Combination with an anomaly-based IDS such as Zeek (formerly Bro)

Summary

Proactive stateful filtering for an IDS with integrated feedback loop for reactive control

Prototype shows viability of proposed approach