Motivation

- Two ways to filter traffic
  - In software – high latency or after-the-fact but expressive
  - In hardware – Line rate but limited to basic network headers
- Want filters that are both expressive and run at line-rate
- Want to filter only what is needed
- Want to capture in real-time
- Examples
  - Billing based on traffic to specific domains
  - Redirecting traffic based on network path
  - Monitoring a particular user
- Network Principals
  - What something is
  - What something is called
  - What something does

Design

- NetAssay has two parts with different responsibilities
  - Main Control Module (MCM)
    - Runtime handling dynamic updates
    - Interfaces with controller and MEs
  - Metadata Engines (ME)
    - Creates equivalent rule sets
    - Uses external data sources for mapping
- External data sources
  - Active Sniffing
  - Instrumented servers and engines
- MEs can use one or more data sources
  - For mapping users – LDAP, 802.1X, and RADIUS
- Dynamic updates
  - Feedback loop based on new information
  - Driven by MEs processing updates

Challenges

- Need a way to describe a network principal in terms of what switches understand
  - Create equivalent sets of rules based on network headers that are equivalent to the network principals
- Systems already do this!
  - BGP, DNS, LDAP, Active Directory, ARP, RADIUS, 802.1X
  - Use the external data sources for mapping
- Network principals are static
  - Mappings are dynamic
  - Dynamism to be handled automatically

Examples

- DNS Update of example.com
- John’s mobile joining network

Done: 1

References

- NetAssay prototype
  https://github.com/sdonovan1985/netassay-proto
- NetAssay development
  https://github.com/sdonovan1985/netassay