Securing the Software-Defined Network Control Layer

Phillip Porras, Steven Cheung, Martin Fong, Keith Skinner and Vinod Yegneswaran
SRI International

NDSS’15

Background

Software defined networking can roughly be split into three layers: application, control and data plane.

This work focuses on securing the *control layer*.

The control layer represents the flow rule decisions made by ***multiple*** *sdn-applications* operating within a network slice, a logical network "group" (controller,switch,host tuple).

Problem

Provide effective and comprehensive *multi sdn-app* arbitration, securing the control layer.

Several challenges to consider:

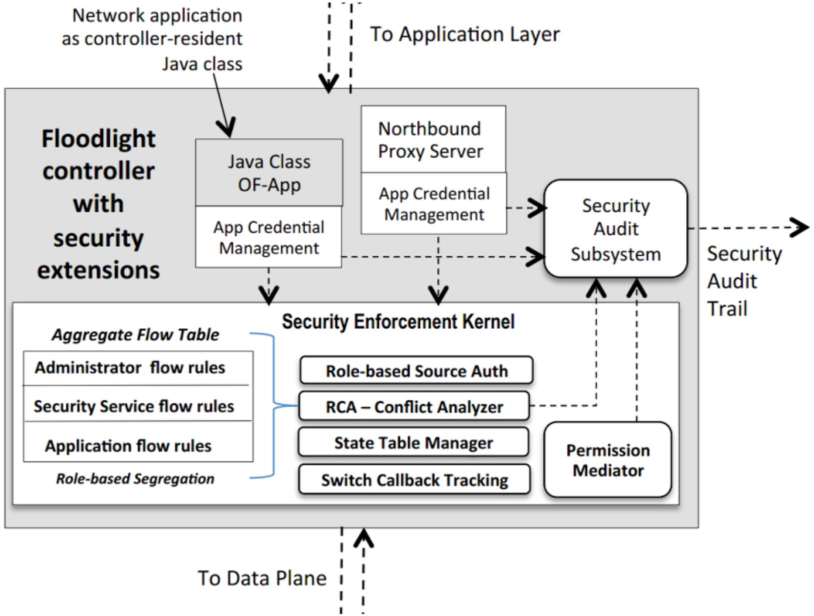
- *coexistence*
- *flow constraints vs circuits*
- *permission model*
- *accountability*
- *privilege separation*

Proposed Solution

Security Enforcement Kernel (*SEK*)

A monolithic controller module, vetting *all data* exchange operations between the application and data plane layers.

Provide a hierarchical authorization scheme… a permissions table with policies.

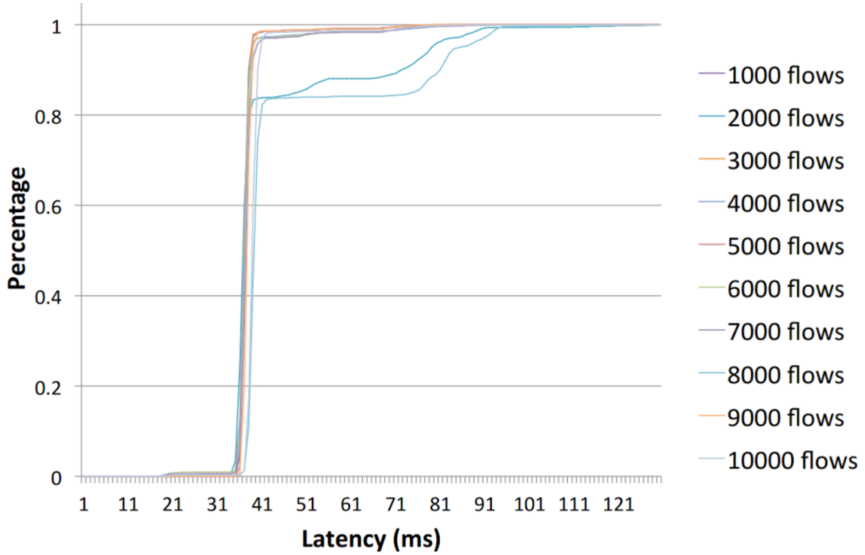


Evalution

Measures *latency*, time SEK takes to perform flow table check and conflict resolution to process and insert rule.

Basically, measure how the system scales with many new flows by generating random traffic.

On average addition of a large number of flows incurs 35ms overhead.



Conclusion

This work presents SEK, a fine grain sdn-app policy control system.

It is partly an extension of fortNOX, but provides extended control options and covers a larger problem/challenge set.

One major limitation we can point out is *one way security enforcement*, the application is not notified its traffic is blocked.

This means this implementation can easily, albeit unintentionally, cripple snd-apps which utilize state control, even those with robust error handling.

Thank you.

Questions?