

Implemented Solution Developments and Improvements

My solution was hampered by the constraints of the SDNCockpit and time as such I created a rudimentary solution to the scenario developed to instantiate flow rules affecting the flow of traffic from one network to another across the controller. I will discuss areas and methods that if this project were to be developed further provide a more rounded security solution.

Flow entries may be developed further by pointing toward a group enabling the execution of complex packet operations undefinable by flow alone.

Use cases include:

Port mirroring / sniffing allows - traffic monitoring, traffic archive and failure resilience.

Load balancing - optimise traffic flow through a network based on the given path weighting.

Fast failover – Overcome a port failure when detected, in this group table there is a monitored watch port/group in each bucket, if it is deemed unresponsive the bucket will not be used. Limitations of this feature include only one bucket can be used at a time and the transition period to an active port/group.

Stateless Firewall Limitations

A firewall analyses a packet in isolation with no context of previous network traffic as such it is a simple safeguard that should not be relied upon as there are other methods that may be implemented to attack a vulnerable network such as IP or mac address spoofing, DoS and DDoS which this solution cannot protect against.

Scalability

As the topology becomes more complex with additional hosts, network interconnectivity, flow tables, group tables, redundancies, monitoring and other forms of security analysis are incorporated this can impact the overall network performance.

Multicontroller Benefits

With a single controller in the project scenario its failure affects the entire network topology as such Openflow supports the configuration of a multi-controller configuration. As such it is important that in a multicontroller environment data is synchronised between each controller.

These controllers may each be given one of three roles:

Equal Role - These controllers may update or modify flows, the switch sends Packet IN messages to each equal role controller and process the Packet OUT, FLOW UPDATES and other such messages from the controllers

Master Role - This controller is responsible for managing all control messages to and from the.

Slave Role - The slave controllers are backups to the Master controller; they cannot send or receive control messages it is only capable of receiving hello and keepalive messages.