OpenFlow based Network Intrusion Detection System

Student: Sujayyendhiren Ramarao Srinivasamurthi

E-mail: sxr1043@rit.edu

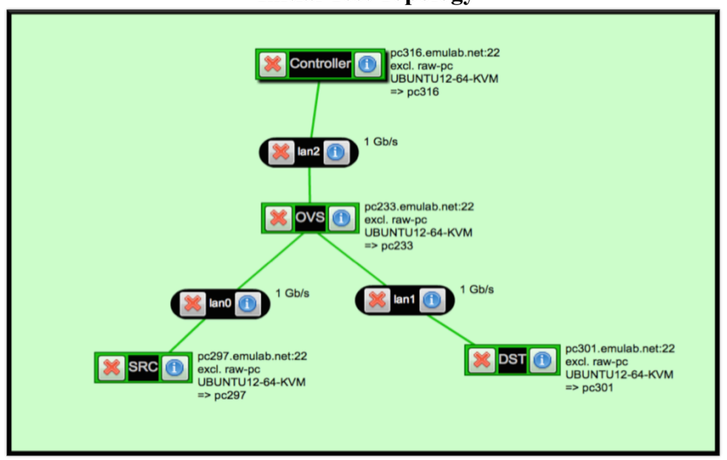
Advisors: Prof. Kaiqi Xiong and Prof. Minseok Kwon

Email: kxxics@rit.edu and jmk@cs.rit.edu

Abstract

OpenFlow based Network Intrusion Detection System (NIDS) is designed to offer an efficient security mechanism via a Software Defined Network (SDN). The network traffic received in the dataplane of OpenVSwitch is processed by matching traffic patterns against the signatures stored in the database based on Bloom filter (in approach-1) or Aho-corasick algorithm (in approach-2). NIDS provides a robust, efficient and modular framework for filtering and configuration in the data plane.

Initial Test Topology

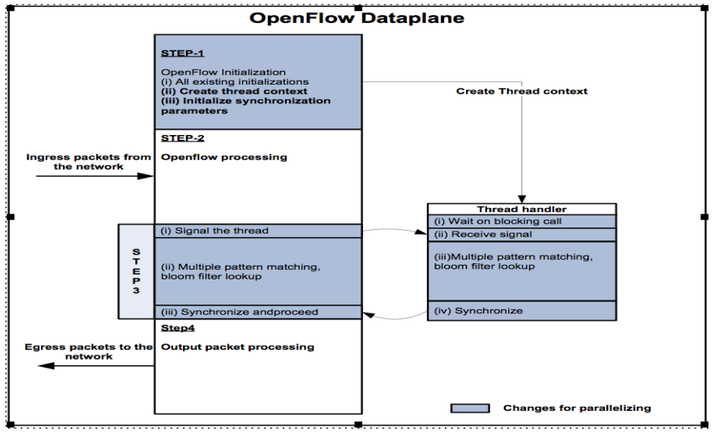


The experimental setup comprises the minimal number of four machines. OpenFlow controller acts as an administrator; it is named as 'Controller' that comprises the configurations of NIDS signatures.

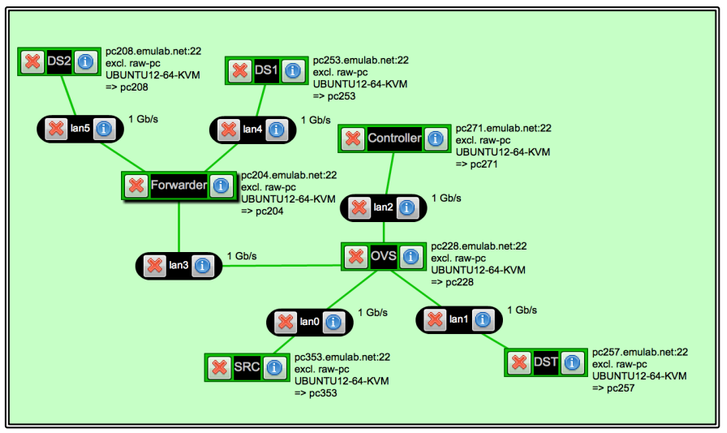
The machine placed at the center contains the OVS kernel module (openvswitch.ko). Network traffic is captured via a hook that adds additional processing. The NIDS control plane resides at the userspace on this machine.

The PC marked as 'Source' generates data plane packets for testing. All the data plane packets are destined to the PC marked as 'Destination.' Packet generator client transmits packets from 'Source' and is destined to 'Destination.'

OVS dataplane parallelized

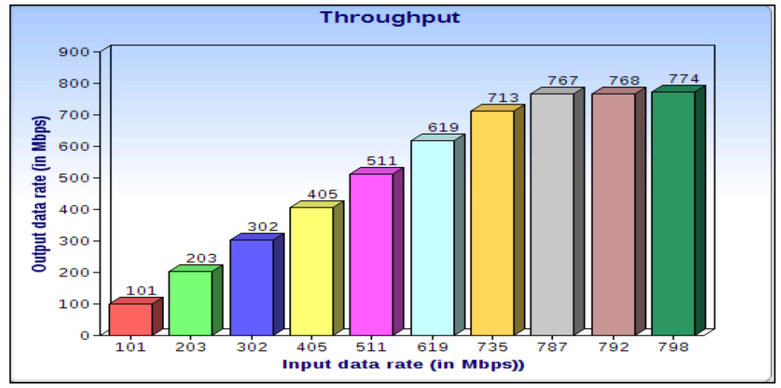


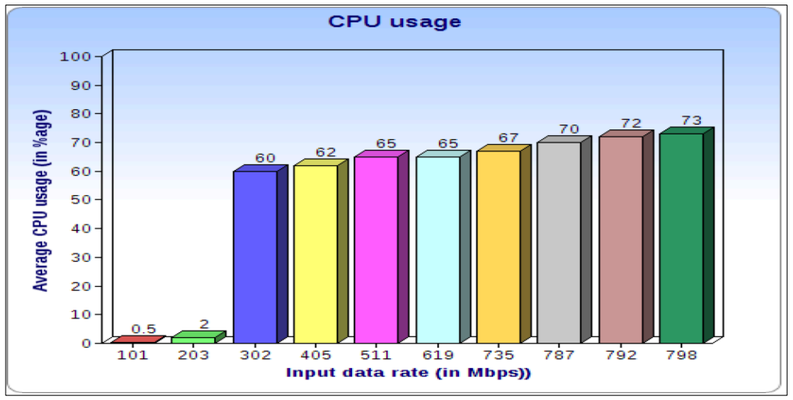
Current Test Topology



The experimental setup comprises previously described four machines along with multiple machines labeled a prefix 'DS'. This architecture offloads the signature pattern matching overhead from the OVS dataplane to the distributed machines. The NIDS control plane more or less remains the same.

The PCs marked as 'DS1' and 'DS2' are distributed processing machines. Malicious data traffic is still blocked by using OpenFlow's flow table.





Conclusion

- NIDS is modular. It consumes negligible data plane memory. It can be scaled into a number of hardware configurations and different kinds of networks.
- Our design eliminates all possibile false positives.

Future Work

- Dynamic threat identification – threats are currently identified based on known signatures. In dynamic threat identification, we may analyze network traffic based on network protocols and traffic activity patterns
- Our implementation will be ported to hardware and introduced to real time traffic.

geni Exploring Networks of the Future

NETWORK SCIENCE AND ENGINEERING

National Science Foundation WHERE DISCOVERIES BEGIN