OPEN NETWORKING
SUMMIT 2016
MARCH 14-17, 2016 | SANTA CLARA, CA

Controller Shield

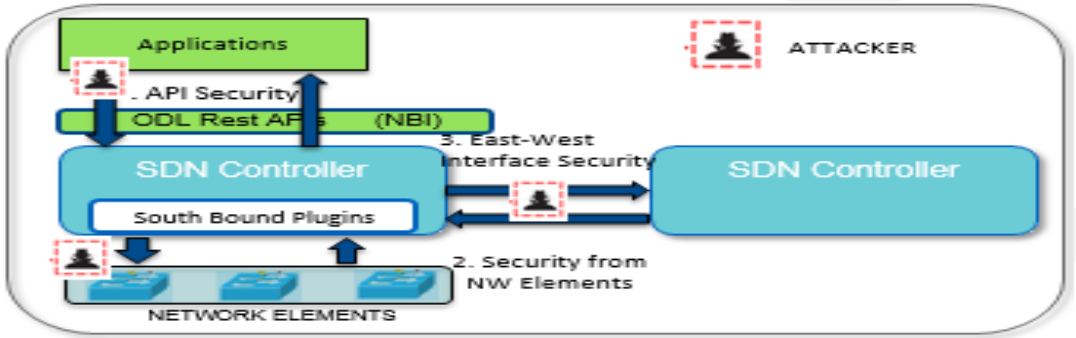
Rafat Jahan, Tata Consultancy Services

Agenda

- Introduction
- Business Needs
- SDN Controller vulnerabilities
- Solution Overview
- Controller Shield Algorithm
- Demonstration
- Releases & References

Introduction

- The SDN controllers provides a bird's eye view of the network and therefore acts as an enabler to program the underlying network and also enhance its security, but, at the same time it opens up a host of security issues due to its own existence.
- The SDN controller acts as a new node which network security has to secure.
- The controller is susceptible to attacks from the users and applications, the peer controller and also from the underlying network elements that it controls
- There are other application layer security issues that have to be handled at the controller.



Diagrammatic representation of the attack scenario

Business Need

Business Need:

- Primary requirement for successful adoption of SDN networks is the security of the controller from variety of network attacks where controller can be the attack surface.
- While the controller may be protected from the network by adding additional layers of security before the packets can reach the controller, the rationale behind a unified controller security plugin is to devise a security information repository for the applications that may be designed to protect the controller from external security threats.
- The present activity of controller shield involves detecting, reporting and storing any anomalous for the incoming packets. The Controller Shield will also send notification to the third party application on detection of possible breach.

Business Benefit:

- Reduction of network outage time and thence loss of revenue can be controlled.
- Loss of valid information that can expose the controller internals to miscreants will be checked upon.
- Solution extendable to any existing SDN controller.

SDN Controller Vulnerabilities

- Denial-of-service (DoS) when deserializing malformed packets
  - packet deserializers in controllers would throw exceptions when handling malformed, truncated, or maliciously-crafted packets.
  - If exceptions are not caught and handled, relevant switch being disconnected
  - DoS attack can be initiated to disconnect switches from controllers
- Topology spoofing via host tracking
  - Most SDN controllers include host tracking
  - Uses Packet-In messages without any validation, authentication, or authorization.
  - An attacker to impersonate a host and bluff the SDN controller
- Same , but different
  - XXE is a well known vulnerability in Java-based projects
  - Dos attacks based on unhandled exceptions are common and well established
  - Topology spoofing is related to MAC spoofing
- SDN is a new and novel technology, it is still just software, and a relatively small number of fundamental issues represent the vast majority of security flaws.

Solution Overview

Controller Shield attempts to identify, thwart and mitigate security attacks t
can occur through various interfaces of controller (here, ODL) :

North Bound Interface :

- Verifying integrity of messages sent from Applications.

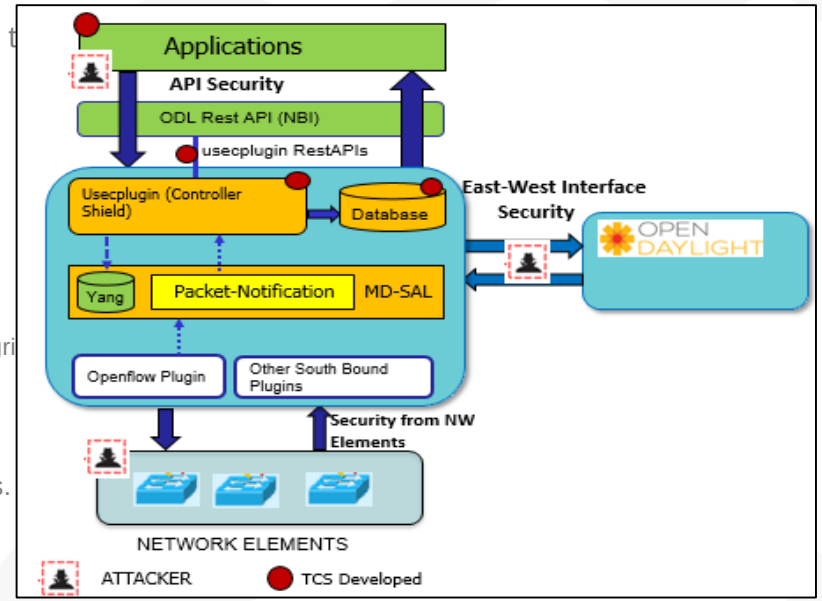
South Bound Interface :

- Securing the controller from DDoS attacks originating from switches/hosts

East-West Interface :

- Authenticating peer controllers before session establishment and checking integri
  protection for the messages exchanged

The present activity on Controller Shield involves detecting, reporting and storing any anomalous behavior at the OpenFlow SBI for the Packet_In messages. The Controller Shield will also send notification to the third party application on detection of possible breach.

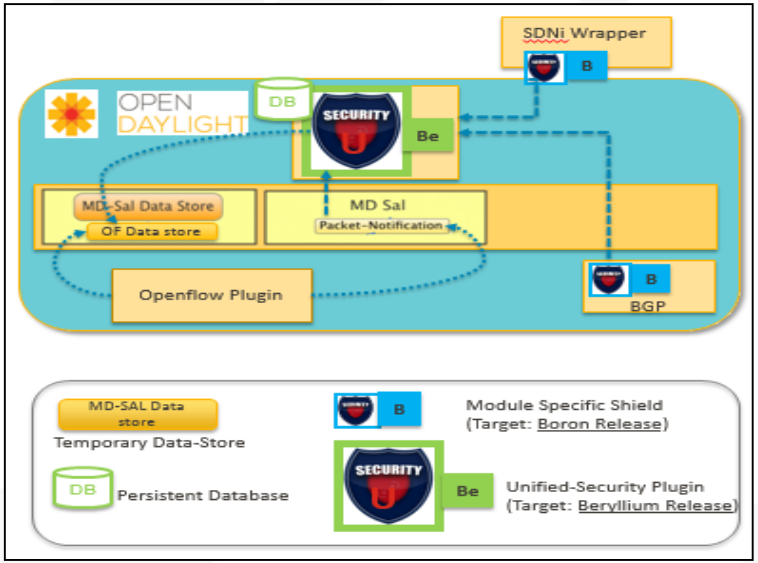


Architecture

Attack Mitigation from Packet-In Messages:

In order to have a mitigation system in place, four key factors need to be kept in mind:

- A fair estimate of the rate of packet influx to be able to figure out an impending and/or actual attack on the system.
- The source of the packet-in messages need to be identified.
- Develop a system to send notification to interested parties when an attack is impending or detected.
- Have a persistent database to record the source of high Packet-In messages for current or any future reference.



Controller Shield Algorithm

Controller Shield is doing Anomaly detection by Reactive defense mechanism:

Detection: Anomaly Detection with Threshold Setting.

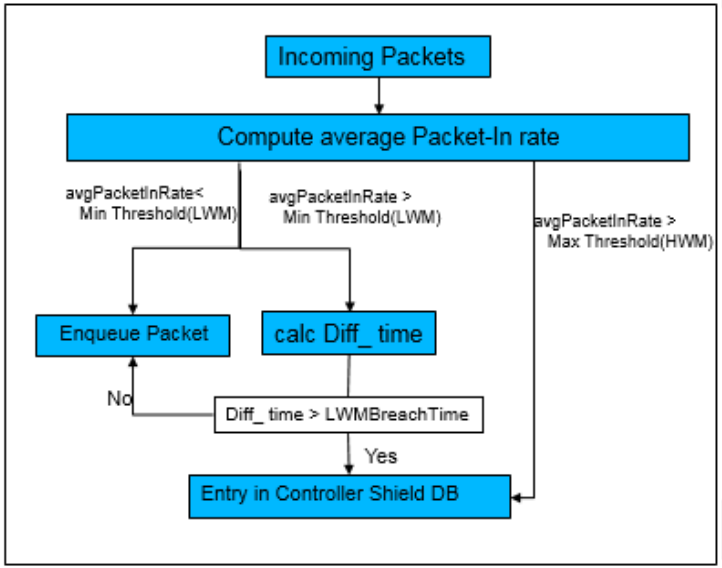
Threshold Setting

- High Threshold (HWM): Reduces the sensitivity of the detection.
- Low Threshold (LWM): Leads to many false positives.

Response :Throttling technique.

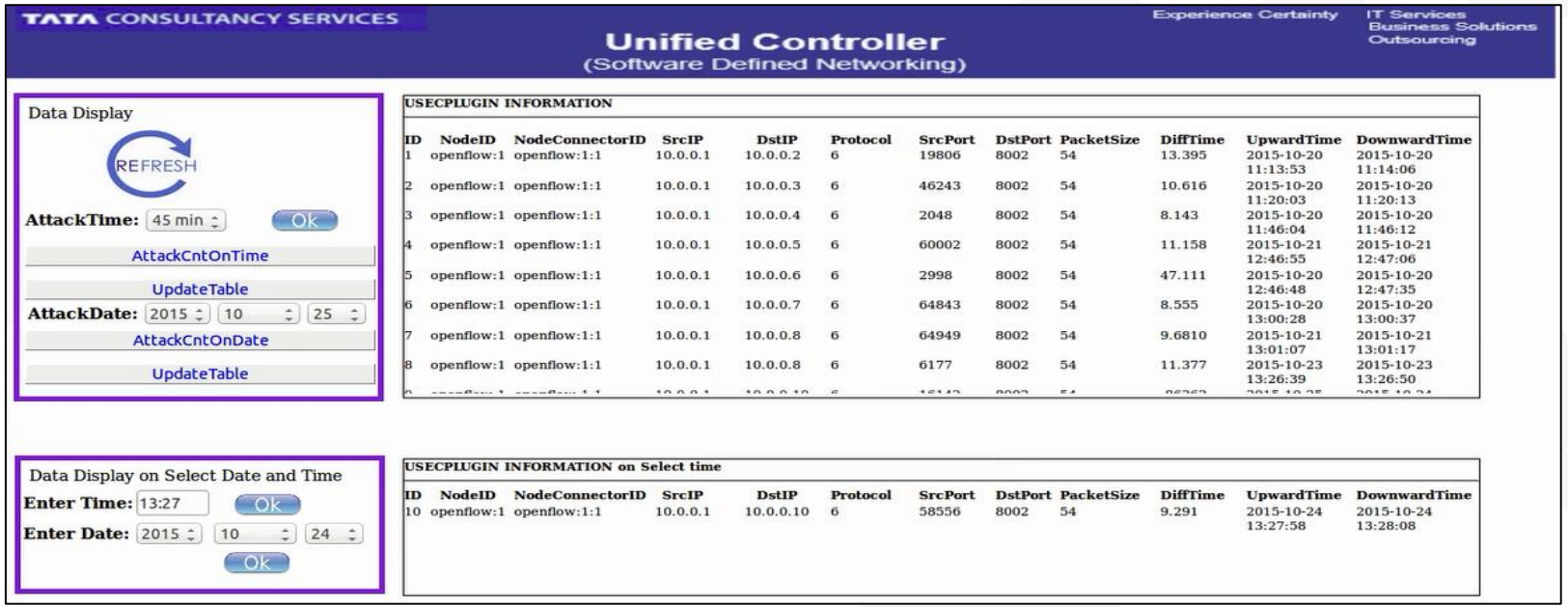
- This technique uses Max-Min (HWM and LWM)throttles.
- Values are adjusted with the logic to adjust the incoming traffic to levels that will be safe for process.
- This data will be provided by the application deploying Controller Shield to set the water marks.
- RestConf developed to set these values.

Defense Mechanism: Reactive Mechanisms where the third party to be deployed is Controller Shield.



Algorithm used – RED

Demo-Controller Shield Dash Board



Releases

ODL Beryllium Release:

- Security from Network Elements - This covers security for the controller from DDoS attacks originating from Switches/Hosts in the network.
- Incorporating RED algorithm.
- Implement database for Dash Board.
- Develop RestConf

Road Map for further Release:

- Northbound interface for AAA communication.
- Enhance security plugin to collect data from TSDR.
- East-West Interface Security.
- Database upgrade

References

Documents to refer for Controller Shield (usecplugin)

Project Proposal: https://wiki.opendaylight.org/view/Project_Proposals:Controller_Shield

User Guide: https://wiki.opendaylight.org/view/Controller_Shield:_Beryllium_User_Guide

Developer Guide: https://wiki.opendaylight.org/view/Controller_Shield:_Beryllium_Developer_Guide

THANK YOU