Big Data Analytics for Real-time DDoS Detection and Mitigation in SDN

{ Rahul Prabhu, Umar Majeed, Sanil Sinai Borkar, Sile Hu }
University of Florida, Department of CISE

Introduction

- Distributed Denial of Service (DDoS) is a major security threat on the Internet and the lack of real-time DDoS detection and mitigation tools can render servers vulnerable.
- In this paper, we propose a system to detect and counter DDoS for Software Defined Networks (SDN) that learns from historical DDoS attack data and deploys counter measures in real-time.
- The availability of centralized control and decoupling of the control and the data plane in SDN which allow for reprogramming routing decisions on the go make this possible.
- The accuracy of the proposed system is 97% using the CIADA DDoS dataset and the latency added by the extra computation was found to be very minimal.

Architecture

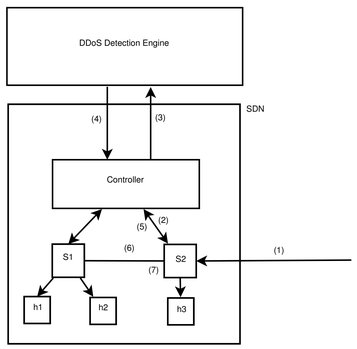


Figure 1: Architecture Overview

The main components of our system are:

- DDE - DDoS Detection Engine running on a Apache Spark cluster.
- SDN Network.
- Communication Modules.

Design and Implementation

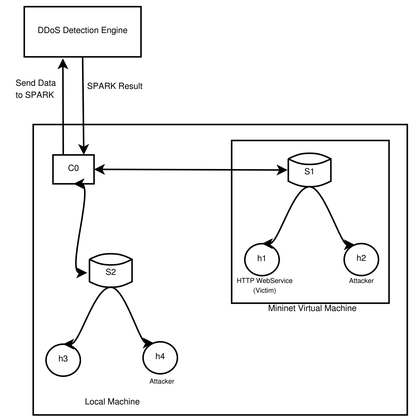


Figure 2: Setup

Our experimental Setup consisted of:

- A local Machine hosts the controller c0, two hosts h3 and h1 and a switch s2.
- A guest mininet VM containing two hosts h1 and h2 and a switch s1 (connected to remote controller c0).
- Attackers h2 and h4 and a victim h1.
- DDoS Detection Engine running on a SPARK Cluster

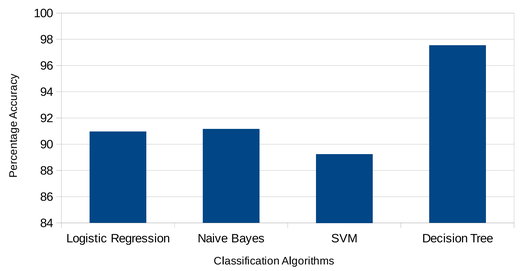
The setup works as follows:

- Host *h1* was setup to run a python HTTP web server which acted as a victim.
- The hosts *h2* and *h4* acted as the attackers which carried out TCP SYN flooding attack on *h1*.
- The switch *s1* communicated with the remote controller *c0* to get the flow-rules at the switch.
- The data received by the controller from the switch was then sent to the DDE running on the AWS EC2 cluster.
- The DDE is the heart of our project and makes a decision from the input it receives from the controller making use of Apache Spark and the MLlib present within it.
- On receiving the decision from EC2 the controller sends information to the switch in the form of add/modify flow rules.
- The switch then took the necessary action(s) as per the instructions given by the controller.

Experiments and Results

TABLE II: Performance of Various MLlib Machine Learning Algorithms

Algorithm	Feature Set Size	Dataset Records	Time for Training (s)	Time for Prediction (s)	Error
Decision Tree	43	36,646	0.0002520084	0.0040941238	2.46%
SVM	43	36,646	0.0002040863	0.0044519901	10.75%
Naive Bayes	43	36,646	0.0005002022	0.0016739368	8.84%
Logistic Regression	43	36,646	0.0003650188	0.0011081696	9.04%



The effectiveness of our system was measured in two folds.

- The accuracy with which the DDE can predict DDoS attacks.
  - As seen from the graph, Decision Tree gives the best accuracy of over 97%.
- The speed with which it can predict any incoming packet.
  - The time required to classify the packets is very small ($0.114\mu s$) and hence does not adversely impact the network performance.

Assumptions

- Can not simulate an actual DDoS attack due to university's network security policies.
- Controller is assumed to be secure.
- Network Latency
- We consider only TCP SYN flooding attack.

Conclusions

- We develop a prototype to detect and counter DDoS in SDN at real-time using SPARK with MLlib.
- Empirical analyses show that Decision Tree is the best algorithm for packet classification giving a classification accuracy of over 97% using CAIDA dataset.
- Latency can be improved by reducing the network bottleneck between the SDN controller and the DDE.

Future Work

- System could further be enhanced to detect and counter other types of DDoS attacks.
- Robustness and security can be further improved by making the controller resistant to DDoS attacks.
- Latency can be improved by reducing the network bottleneck between the SDN controller and the DDE.

Acknowledgement

- Dr. Andy Li
- Department of CISE and ECE, University of Florida.
- Center for Applied Internet Data Analysis.