



**KSIT**  
K S INSTITUTE OF TECHNOLOGY

Kammavari Sangham (R) - 1952

**K.S.INSTITUTE OF TECHNOLOGY**

(Affiliated to VTU, Belgaum. Approved by AICTE, New Delhi)

No.14, Raghuvanahalli, Kanakapura Main Road, Bengaluru - 560109

Phone : 080-28435722 / 24, Fax : 080-28435723, Email : principal.ksit@gmail.com, Website : www.ksit.ac.in



# Detection of DDoS attack using Machine Learning Algorithms in SDN Environment

BATCH-05

## Guided By:

Mrs. Sougandhika Narayan

Assistant Professor

Dept of CSE, KSIT

## Presented by:

Dakaraju Viswateja 1KS16CS018

Keerthi M 1KS16CS032

Kiran Kumar M 1KS16CS034

Manipi Manoj 1KS16CS039

# Contents

- ▶ Abstract
- ▶ Introduction
- ▶ Literature Survey
- ▶ Problem Identification
- ▶ Methodology
- ▶ Project Goal
- ▶ Contribution to Society



## Abstract

- ▶ Computer intrusion and attack detection has always been a significant issue in networked environment. In most cases, there are two levels in which an intrusion may take place, namely the system level and the network level.
- ▶ This project discusses an algorithms to protect from a specific kind of network-level attack called Distributed Denial of Service attack.

# Introduction

- ▶ A distributed denial of service (DDoS) attack is when a hacker uses a botnet to send your web server an overwhelming number of HTTP requests in a very short period of time.
- ▶ DDoS attacks are the most common attacks in these technical era. So, that most of the important websites which are useful for finishing some of our daily tasks are DDoS attacked which is leading to unavailability of the web resources.
- ▶ By using wireshark a network monitor these bots are detected with the help of Machine Learning techniques which in turn prevents the attack of DDoS.

## Literature Survey

- ▶ A Saboor et al[3] proposed the detection of DDoS attack based on correlation algorithm and IAFV algorithm. They used different time series with sliding windows for improving the detection rate.
- ▶ Saurav Nanda et al. [5] used Bayesian Network and achieved an accuracy of 91.68 % which indicates that out of 278,598 attacks, their model was able to accurately predict 254,834 attacks.
- ▶ Gisung Kim, et.al[6] proposed a hybrid learning model to detect the DDoS attack and to protect the Open Flow switches. They found that their model work well for unknown attacks also.

## Literature Survey

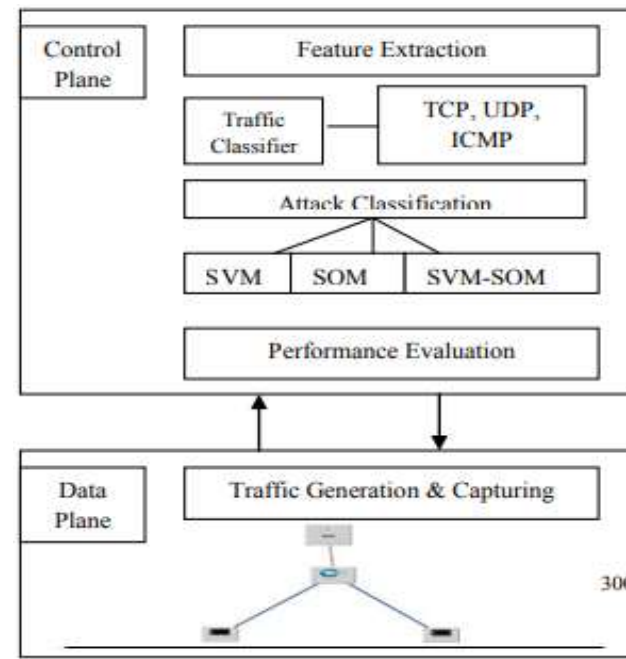
- ▶ Ahmad Y. Javaid et al. [7] used deep learning methods to detect the DDoS attack in SDN environment. They had collected the traffic from home wireless network (HWN) scenario. And they got 96.65% accuracy.
- ▶ Lohit Barki et al.[8] have used different machine learning techniques such as Naive Bayes, K-nearest neighbor, K-Means, K-medoids to detect the DDoS attack. They found that Naïve Bayes model work well compared to other considered algorithms with highest accuracy.
- ▶ The researchers have designed the system to detect DDoS attacks based on a decision-tree technique, and they traced back to the approximate locations of the attacker with a traffic flow pattern-matching technique .

# Problem Identification

- ▶ The main aim of DDoS attacks is to prevent the legitimate user to access the service for a long time. In this attack, attacker tries to compromise the multiple numbers of hosts to send a huge amount of traffic intentionally towards a legitimate user. This leads to unavailability of service for large amount of time.
- ▶ Two Algorithms are used to detect the DDoS attack one is SVM (Support vector Machine) another is SOM (Self Organized Map). Since SVM is a supervised machine learning model, it should be trained with labelled data only.
- ▶ In case of SOM, it is an unsupervised machine learning model, as they apply competitive learning as opposed to error-correction learning and in the sense that they use a neighborhood function to preserve the topological properties of the input space..
- ▶ In order to handle this DDoS attack, we have proposed a combination of two machine learning based model with Support Vector Machine (SVM) and Self Organized Map(SOM).

# Methodology

- Below Figure shows the architecture of our proposed method.

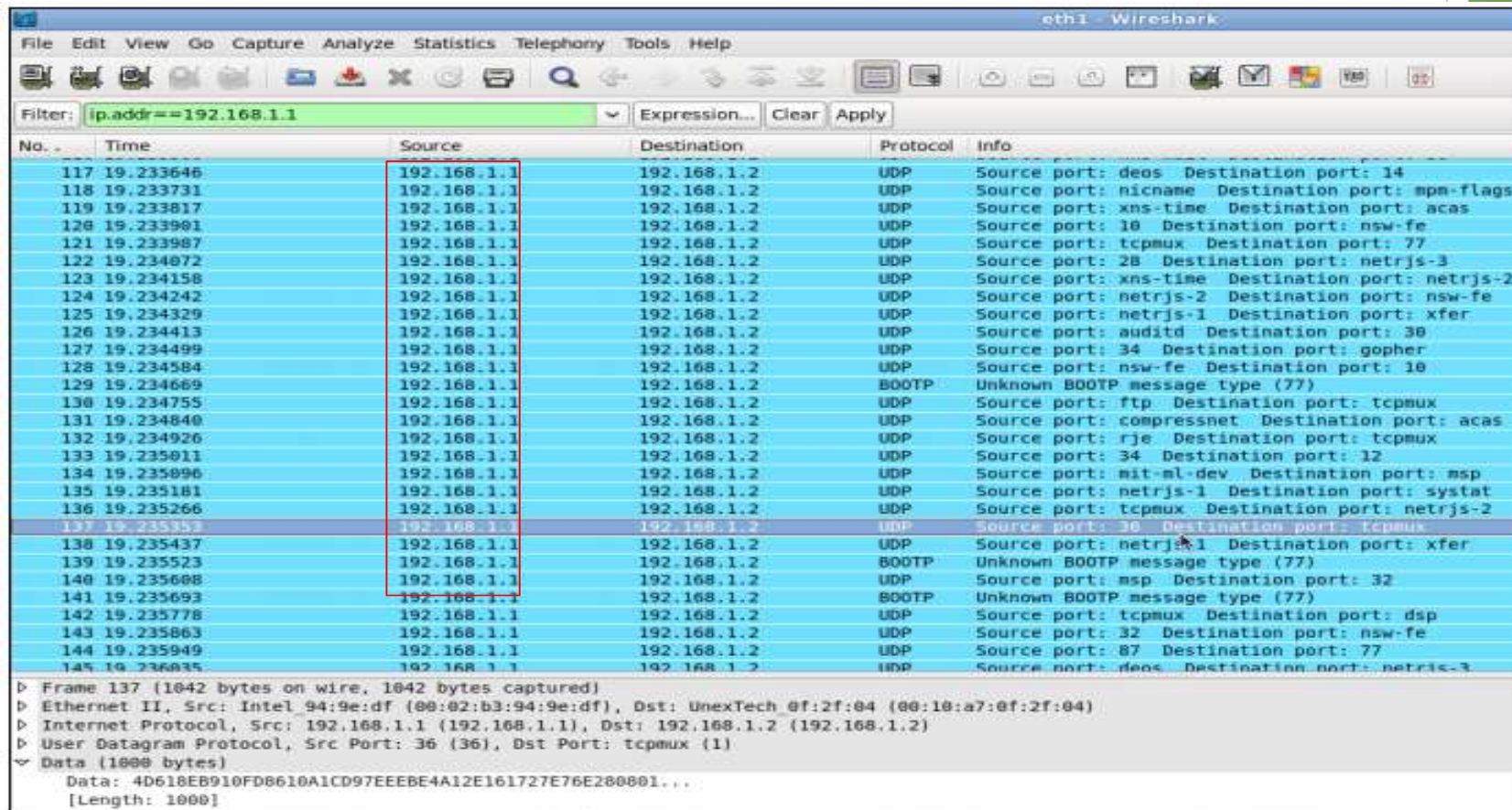


wireshark

Topology



# Methodology



The screenshot shows a Wireshark capture on interface eth1. A filter is applied: `ip.addr==192.168.1.1`. The packet list shows a series of UDP packets from 192.168.1.1 to 192.168.1.2. The packet details for packet 137 are expanded, showing the Ethernet II, Internet Protocol, User Datagram Protocol, and Data layers.

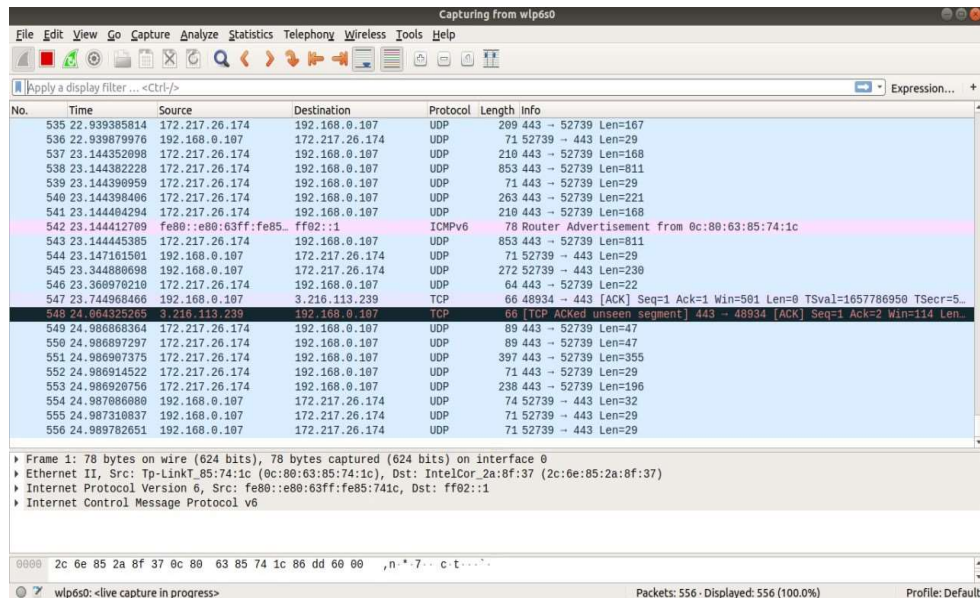
No.	Time	Source	Destination	Protocol	Info
117	19.233646	192.168.1.1	192.168.1.2	UDP	Source port: deos Destination port: 14
118	19.233731	192.168.1.1	192.168.1.2	UDP	Source port: nicname Destination port: mpm-flags
119	19.233817	192.168.1.1	192.168.1.2	UDP	Source port: xns-time Destination port: acas
120	19.233901	192.168.1.1	192.168.1.2	UDP	Source port: 10 Destination port: nsw-fe
121	19.233987	192.168.1.1	192.168.1.2	UDP	Source port: tcpmux Destination port: 77
122	19.234072	192.168.1.1	192.168.1.2	UDP	Source port: 28 Destination port: netrjs-3
123	19.234158	192.168.1.1	192.168.1.2	UDP	Source port: xns-time Destination port: netrjs-2
124	19.234242	192.168.1.1	192.168.1.2	UDP	Source port: netrjs-2 Destination port: nsw-fe
125	19.234329	192.168.1.1	192.168.1.2	UDP	Source port: netrjs-1 Destination port: xfer
126	19.234413	192.168.1.1	192.168.1.2	UDP	Source port: auditd Destination port: 30
127	19.234499	192.168.1.1	192.168.1.2	UDP	Source port: 34 Destination port: gopher
128	19.234584	192.168.1.1	192.168.1.2	UDP	Source port: nsw-fe Destination port: 10
129	19.234669	192.168.1.1	192.168.1.2	BOOTP	Unknown BOOTP message type (77)
130	19.234755	192.168.1.1	192.168.1.2	UDP	Source port: ftp Destination port: tcpmux
131	19.234840	192.168.1.1	192.168.1.2	UDP	Source port: compressnet Destination port: acas
132	19.234926	192.168.1.1	192.168.1.2	UDP	Source port: rje Destination port: tcpmux
133	19.235011	192.168.1.1	192.168.1.2	UDP	Source port: 34 Destination port: 12
134	19.235096	192.168.1.1	192.168.1.2	UDP	Source port: mit-ml-dev Destination port: msp
135	19.235181	192.168.1.1	192.168.1.2	UDP	Source port: netrjs-1 Destination port: systat
136	19.235266	192.168.1.1	192.168.1.2	UDP	Source port: tcpmux Destination port: netrjs-2
137	19.235353	192.168.1.1	192.168.1.2	UDP	Source port: 30 Destination port: tcpmux
138	19.235437	192.168.1.1	192.168.1.2	UDP	Source port: netrjs-1 Destination port: xfer
139	19.235523	192.168.1.1	192.168.1.2	BOOTP	Unknown BOOTP message type (77)
140	19.235608	192.168.1.1	192.168.1.2	UDP	Source port: msp Destination port: 32
141	19.235693	192.168.1.1	192.168.1.2	BOOTP	Unknown BOOTP message type (77)
142	19.235778	192.168.1.1	192.168.1.2	UDP	Source port: tcpmux Destination port: dsp
143	19.235863	192.168.1.1	192.168.1.2	UDP	Source port: 32 Destination port: nsw-fe
144	19.235949	192.168.1.1	192.168.1.2	UDP	Source port: 87 Destination port: 77
145	19.236035	192.168.1.1	192.168.1.2	UDP	Source port: deos Destination port: netrjs-3

Frame 137 (1042 bytes on wire (8336 bytes captured) on interface eth1):

- Ethernet II, Src: Intel 94:9e:df (08:02:b3:94:9e:df), Dst: UnexTech 0f:2f:04 (08:10:a7:0f:2f:04)
- Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)
- User Datagram Protocol, Src Port: 36 (36), Dst Port: tcpmux (1)
- Data (1000 bytes)
  - Data: 4D618EB910FD8610A1CD97EEBE4A12E161727E76E280801...
  - [Length: 1000]

Flooding a Particular Network

# Methodology



The image shows the 'Wireshark · Destinations and Ports · wlp6s0' window. It displays a table with columns: Topic / Item, Count, Average, Min val, Max val, Rate (ms), Percent, Burst rate, and Burst start. The table lists various destinations and ports, with a red box highlighting the 'Count' column.

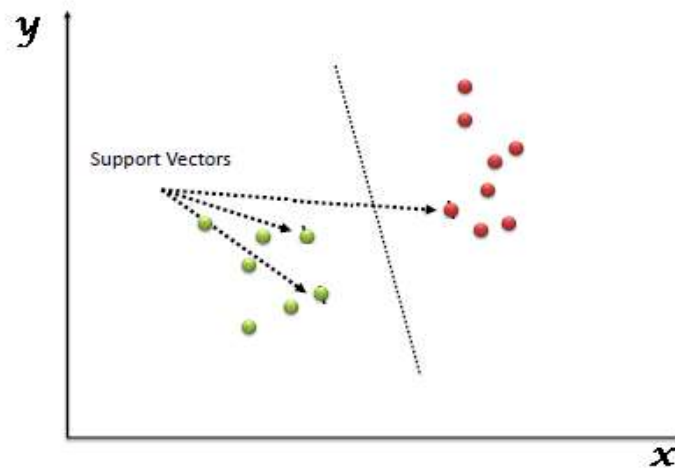
Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Destinations and Ports	574				0.0141	100%	0.9300	16.710
91.189.91.157	1				0.0000	0.17%	0.0100	26.090
74.125.130.188	1				0.0000	0.17%	0.0100	33.985
3.216.113.239	5				0.0001	0.87%	0.0200	14.591
216.58.200.138	1				0.0000	0.17%	0.0100	14.359
216.58.196.163	6				0.0001	1.05%	0.0200	5.093
192.168.0.107	283				0.0070	49.30%	0.5300	16.719
192.168.0.1	2				0.0000	0.35%	0.0100	14.280
172.217.31.196	18				0.0004	3.14%	0.0200	14.359
172.217.26.174	138				0.0034	24.04%	0.4500	22.456
172.217.167.142	15				0.0004	2.61%	0.0400	2.459
172.217.166.99	1				0.0000	0.17%	0.0100	14.359
172.217.166.110	32				0.0008	5.57%	0.1500	16.578
172.217.163.35	1				0.0000	0.17%	0.0100	17.601
172.217.163.163	1				0.0000	0.17%	0.0100	17.601
172.217.163.138	12				0.0003	2.09%	0.0700	1.628
172.217.160.142	57				0.0014	9.93%	0.1400	14.358

Monitoring particular network using Wireshark

Analyzing the network and generation of DataSet

# Methodology

- “Support Vector Machine” (SVM) is a supervised machine learning which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).



SVM Algorithm Classification

# Project Goal

- ▶ The main goal of our project is to build a model which must be able to detect a DDoS attack.

## Contribution to Society

- ▶ Prevention against DDoS attacks is the most desirable defence technique to fight against the DDoS attacks. DDoS put an immense threat to the resources of the victim as well as to the network bandwidth and Infrastructure.
- ▶ Therefore, if an attack has been already launched and become successful, it may cause significant compromise to the Victim's system.
- ▶ Thus, protection against DDoS attacks is more effective against DDoS attack traffic as well as manages large attack load before it may cause the attack to be successful. This ensures normal operation of the victim.

## References:

- ▶ <https://github.com/jivoi/awesome-ml-for-cybersecurity>
- ▶ <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>
- ▶ <https://helpdeskgeek.com/how-to/how-to-identify-a-ddos-attack-on-your-server-stop-it/>
- ▶ <https://www.svm-tutorial.com/2017/02/svms-overview-support-vector-machines/>