Anomaly Detection in Network Traffic: A Machine Learning Approach with OMNeT++

EHB 415E

Network Design:

Detailed System Design and Implementation for Anomaly Detection

Prepared by:

Batu Burgu
Student ID: 040210098

Javad Ibrahimli
Student ID: 040210932

Kerem Karadeniz
Student ID: 040210049

2024-2025 Academic Year

Supervised by: İbrahim Hökelek PhD.

Contents

Contents

1 Introduction

Distributed Denial-of-Service (DDoS) attacks are a significant threat to network security, flooding systems with malicious traffic to disrupt legitimate services. To detect and mitigate such attacks, Intrusion Detection Systems (IDS) analyze network traffic for unusual patterns and alert administrators to potential threats.

This report outlines the design and implementation of a network simulation to study the effects of DDoS attacks. Using tools like OMNeT++ and the INET framework, the project models legitimate traffic alongside coordinated attacks from zombie nodes, providing a foundation for evaluating intrusion detection strategies.

2 System Design

2.1 Topology Overview

The network topology consists of an ISP router connecting multiple subnets, each hosting legitimate clients and zombie nodes to simulate a distributed denial-of-service (DDoS) attack. The server is positioned in a separate subnet to act as the primary target of both legitimate and malicious traffic.

The topology includes the following components:

- **ISP Router:** Central router connecting all subnets and facilitating inter-network communication.
- **Subnets:** Three subnets exist in the topology, two of which hosting legitimate clients and one hosting zombie nodes to generate attack traffic. A router and an ethernet switch is also located in all subnets.
- **Server:** The main target of legitimate client requests and DDoS traffic, hosted in a separate subnet.

This design models a realistic environment for evaluating the impact of DDoS attacks and testing the effectiveness of intrusion detection strategies.

2.2 Routing and Traffic Management

The Configurator module available in the INET library was utilized in order to configure the devices within the network. This module automates tasks such as creating the necessary subnets, assigning specific IP addresses to each device and setting up the default gateways to the each subnet and client.

For traffic routing, the Configurator's default option, the OSPF routing protocol, was used. OSPF ensures efficient and dynamic management of routing paths across the network, facilitating seamless communication between devices

3 Implementation

3.1 Tools and Configuration

The simulation is designed to model a wired network scenario, with plans to extend testing to wireless network based on the capabilities of the specific environments and attack scenarios that will be tested.In order to simulate a realistic traffic, the UDP protocol has been employed, modeling the channel load generated by real users on the channel. UDP protocol is used instead of TCP because of the eliminating of the TCP protocol's built in network traffic management properties.

3.1.1 Nodes:

All legitimate clients (named as source) are created and simulated by using the app "UdpBasicApp" provided by the INET [1]. They are programmed to send the ARP (Address Resolution Protocol) request when simulation time reaches two seconds. After getting ARP response, they are programmed to send 20 bytes of data every 0.0005 ms.

Zombies are created and simulated by using the app "UdpBasicBurst" provided by the INET [2]. They are programmed to send the ARP (Address Resolution Protocol) request when simulation time reaches 2.0001 seconds. The difference between the starting time of normal clients and zombies enables the system to be simulated in normal conditions for a while which will play a crucial role for training our AI model. After getting ARP response, they are programmed to send 500 bytes of data every 0.0001 ms.

Server is created and simulated by using the app "UdpSink" provided by the INET [3]. It is programmed to drop the received package after getting it. Hence, there is not any data sent-back.

3.1.2 Connections

For simplicity, all connections are established with ethernet cables at the moment we are writing this report. The connections between the ISP router and other routers will be established wirelessly in the upcoming weeks.

All ports except the port between the server and the switch connecting the server to its router are connected by an ethernet channel capable of handling data rate of 400 gigabytes/second. This ensures that the system works without any problems when there is not any anomaly in the network. (The maximum throughput generated by the legitimate clients is calculated to be 320 kilobytes/second.)

The server is connected to its switch with an ethernet cable capable of handling 10 gigabytes/second data flow. This ensures that the system works when there is not any anomaly but this channel bottlenecks the system when a DDoS attack is happening. (Zombies' throughput is 10 gigabytes/second.).

EthernetSwitch [4] and Router [5] classes are used for creating and simulating switches and routers. An Ipv4NetworkConfigurator object is used as the configurator of the network.

3.2 Network Simulation

Since the simulation starts from time zero, initial ARP requests that forms ARP tables are observed during the early stages of the simulation. This phase ensures that the network is properly established and functioning as intended before deploying zombie clients and simulating attacks.

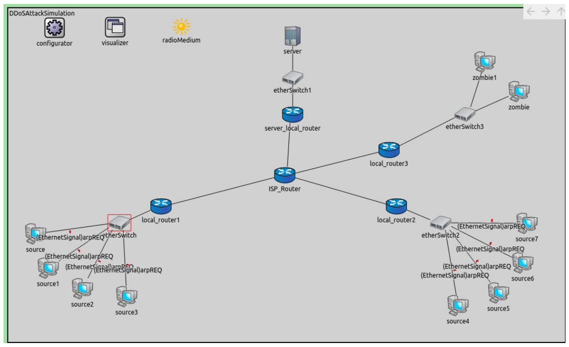

Figure 1: ARP Request

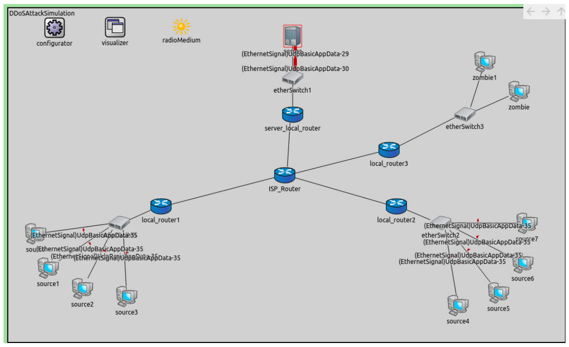

Figure 2: Healty Network

Once the healthy network operation started, zombie clients—responsible for generating DDoS attack traffic are activated. The effects of these clients on the network, such as increased queue length and packet drops, are closely monitored.

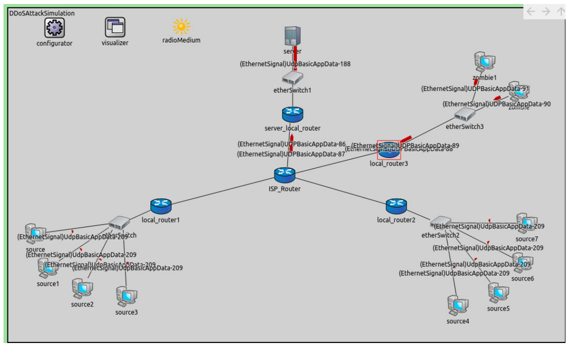
DDoSAttackSimulation
configurator
visualizer
radioMedium
server
(EthernetSignal)UdpBasicAppData-188
etherSwitch1
server_local_router
(EthernetSignal)UDPBasicAppData-86
(EthernetSignal)UDPBasicAppData-87
(EthernetSignal)UDPBasicAppData-89
(EthernetSignal)UDPBasicAppData-88
local_router3
zombie1
(EthernetSignal)UDPBasicAppData-91
(EthernetSignal)UDPBasicAppData-90
zombie
etherSwitch3
ISP_Router
local_router1
local_router2
etherSwitch
etherSwitch2
(EthernetSignal)UdpBasicAppData-209
source
source1
source2
source3
source4
source5
source6
source7

Figure 3: Attack Activated

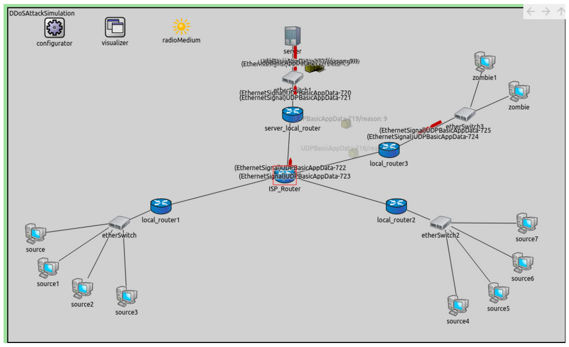
DDoSAttackSimulation
configurator
visualizer
radioMedium
server
UDPBasicAppData-272/reason: 9
etherSwitch1
(EthernetSignal)UDPBasicAppData-720
(EthernetSignal)UDPBasicAppData-721
server_local_router
UDPBasicAppData-719/reason: 9
UDPBasicAppData-718/rea
(EthernetSignal)UDPBasicAppData-722
(EthernetSignal)UDPBasicAppData-723
ISP_Router
local_router3
(EthernetSignal)UDPBasicAppData-725
(EthernetSignal)UDPBasicAppData-724
etherSwitch3
zombie1
zombie
local_router1
local_router2
etherSwitch
etherSwitch2
source
source1
source2
source3
source4
source5
source6
source7

Figure 4: Package Drops

4 Testing and Evaluation

4.1 Performance Metrics

The analysis of the simulation results focuses on several key performance metrics, including packet send frequency, queue length, source and destination ports, packet size, and flow duration. These metrics were extracted from the simulation results to evaluate the network's behavior under normal conditions and during DDoS attacks.

However, it was observed that some data might be lost during OMNeT++'s data exports. To address this potential limitation, the simulation environment will be enhanced by integrating tools like Wireshark or other PCAP managers. These tools will enable more detailed packet-level analysis and provide a complementary perspective on the collected metrics, ensuring the accuracy and completeness of the performance evaluations.

5 Conclusion and Future Work

The next steps for this project include refining the data export process to address any potential loss of metrics during simulation analysis. Additionally, the simulation will be extended to wireless network scenarios, enabling the DDoS attacks for various environments. Finally, these efforts will also focus on gathering a large dataset and detailed performance metrics necessary for developing an artificial intelligence model aimed at improving attack detection and preventing strategies.

6 References

1. Legitimate Client App: `https://doc.omnetpp.org/inet/api-current/neddoc/inet.applications.udpapp.UdpBasicApp.html`

2. Zombie Client App: `https://doc.omnetpp.org/inet/api-current/neddoc/inet.applications.udpapp.UdpBasicBurst.html`

3. Server App: `https://doc.omnetpp.org/inet/api-current/neddoc/inet.applications.udpapp.UdpSink.html`

4. Ethernet Switch: `https://doc.omnetpp.org/inet/api-current/neddoc/inet.node.ethernet.EthernetSwitch.html`

5. Router: `https://doc.omnetpp.org/inet/api-4.4.0/neddoc/inet.node.inet.Router.html`