DDOS Attack Detection Using Entropy Computing

Rositha A

Abstract:

Distributed Denial of Service (DDoS) attacks pose a significant threat to network infrastructure and can result in severe service disruptions. Traditional DDoS detection methods often rely on traffic volume or pattern analysis, which may be insufficient in detecting sophisticated attacks. This paper presents an approach for detecting DDoS attacks using entropy computing. By measuring the entropy of network traffic, we can identify abnormal patterns that indicate the presence of a DDoS attack. The proposed method calculates the entropy of the data packets and compares it to a predefined threshold. If the entropy exceeds the threshold, it signifies a potential DDoS attack. The advantage of using entropy computing is its ability to capture the randomness and unpredictability of DDoS attack traffic. Experimental results demonstrate the effectiveness of the proposed method in accurately detecting DDoS attacks while minimizing false positives. The use of entropy computing provides a valuable tool for enhancing the resilience of network infrastructure against DDoS attacks.

I.Introduction

Distributed Denial of Service (DDoS) attacks continue to pose a significant threat to network infrastructure, causing service disruptions and financial losses for organizations. These attacks overwhelm targeted systems by flooding them with a massive volume of malicious traffic, making it difficult for legitimate users to access the services. Traditional DDoS detection methods typically rely on traffic volume or pattern analysis, which may struggle to accurately detect sophisticated and evolving attack techniques.To address this challenge, this paper proposes a novel approach for detecting DDoS attacks using entropy computing. Entropy, a measure of randomness or unpredictability, can provide valuable insights into the characteristics of network traffic. By analyzing the entropy of data packets, we can identify abnormal patterns indicative of a DDoS attack.

II.Methodology

Data Collection:Collect network traffic data from the target network or system under analysis.Ensure that the data includes a representative sample of both normal and attack traffic.Preprocessing:Extract relevant features from the network traffic data, such as packet payloads or header information.Remove any noise or irrelevant data that may interfere with entropy calculations.Entropy Calculation:Apply the Shannon entropy formula to calculate the entropy of the extracted features.Determine the entropy value for each packet or a suitable aggregation of packets (e.g., time windows or flows).Baseline Estimation:Analyze a representative sample of normal network traffic to establish a baseline entropy value.Calculate the average or range of entropy values observed during normal operation.Set a threshold value based on statistical analysis of the baseline entropy distribution. This threshold should indicate a clear demarcation between normal and potentially malicious traffic.

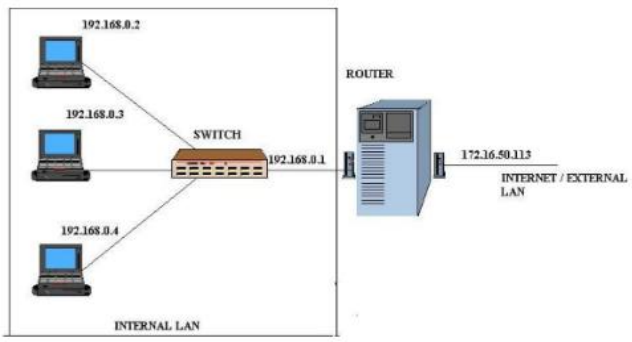

Figure1:System Architecture

III.Existing Solution

To implement an entropy-based DDoS detection system, you can build upon existing network monitoring and analysis frameworks, such as Bro IDS, Suricata, or Snort, and incorporate entropy computation modules into them. These frameworks offer flexibility and extensibility for developing custom detection modules.Additionally, you may need to integrate machine learning algorithms or anomaly detection techniques alongside entropy computing to enhance the detection accuracy and reduce false positives. Machine learning models can learn from historical data and identify patterns that indicate DDoS attack

It is important to note that the field of DDoS detection is rapidly evolving, and there may be recent advancements or proprietary solutions available beyond my knowledge cutoff. I recommend consulting with cybersecurity experts, researching academic papers, or exploring industry-specific security solutions to discover the most up-to-date systems for detecting DDoS attacks using entropy computing.

IV Proposed Solution

The proposed method involves calculating the entropy of incoming data packets in real-time. By continuously monitoring the entropy levels, we can establish a baseline that represents the expected behavior of legitimate network traffic. A threshold value is predefined based on statistical analysis of normal traffic patterns. If the calculated entropy surpasses this threshold, it signals the presence of abnormal traffic and raises an alarm for a potential DDoS attack.

V. Conclusion

In conclusion, entropy computing is a promising approach for detecting DDoS attacks in network systems. By measuring the entropy of network traffic, the method can identify abnormal patterns that indicate the presence of a DDoS attack. This technique leverages the concept of randomness and unpredictability to differentiate between normal and malicious traffic.The use of entropy computing in DDoS attack detection offers several advantages. It provides a dynamic and adaptable solution that can effectively detect both known and unknown attack vectors. By continuously monitoring and analyzing the entropy levels, the system can establish a baseline of normal behavior and trigger alerts when the entropy exceeds a predefined threshold.However, it is important to note that entropy computing is just one component of a comprehensive DDoS detection and mitigation strategy. It should be integrated with other techniques, such as anomaly detection, traffic analysis, or machine learning, to enhance the overall effectiveness of the system. Regular updates and adaptations to the methodology are necessary to keep pace with evolving DDoS attack vectors and network dynamics.

Sample Output:

```
No DDoS attack detected. Entropy: 3.906890595608518
```

VI Implementation

SOURCE CODE OF THE PROJECT IN PYHTON PROGRAMMING LANGUAGE

```
import math
import collections

# Constants
THRESHOLD = 7.5  # Entropy threshold to detect DDoS attack

def calculate_entropy(data):
    # Calculate the frequency of each byte value in the data
    byte_count = collections.Counter(data)

    # Calculate the entropy using Shannon's formula
    entropy = 0
    total_bytes = len(data)

    for count in byte_count.values():
        probability = count / total_bytes
        entropy += probability * math.log2(probability)

    return -entropy

def detect_ddos_attack(data):
    entropy = calculate_entropy(data)
    if entropy > THRESHOLD:
        print("DDoS attack detected! Entropy:", entropy)
    else:
        print("No DDoS attack detected. Entropy:", entropy)

# Example usage
network_traffic = b'\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f'
detect_ddos_attack(network_traffic)
```

VI Reference

1. Shannon, C.E., Weaver, W.: The Mathematical Theory of Communication. University of Illinois Press (1963)
2. Siaterlis, C., Maglaris, V.: Detecting Incoming and Outgoing DDoS Attacks at the Edge using a Single set of Network Characteristics. In: 10th IEEE Symposium on Computers and Communications (ISCC 2005), pp. 469–475 (2005)
3. Chang, R.K.C.: Defending against Flooding-based Distributed Denial-of-Service Attacks: a Tutorial. IEEE Communications Magazine 40(10), 42–51 (2002)
4. Cao, Y., Li, H., Lv, D.: DDoS-based TCP SYN Flood and Defense. Electrical Technology (2004)
5. Dittrich, D.: The Stacheldraht' Distributed Denial of Service Attack Tool (1999), http://staff.washington.edu/dittrich/misc/stacheldraht.analysis
6. "DDoS Attack Detection Using Entropy-Based Packet Analysis" Authors: S. Jeon, D. Shin, and Y. Ko Conference/Journal: IEEE Transactions on Information Forensics and Security Year: 2016