Implementation of an Algorithm to Find the Source of Attack

Software Engineering (CSE-381)

Semester-Long Assignment Report

Report submitted in fulfilment of the requirements
for the Software Engineering Course of

Third Year B. Tech.
in
Computer Science and Engineering

Submitted by
Pulkit Srivastava (Roll No: 18075045)

Under the mentorship of
Ms. Dipty Tripathi

and guidance of
Prof. A. K. Tripathi

Department of Computer Science and Engineering
Indian Institute of Technology (BHU) Varanasi
Varanasi, Uttar Pradesh, India – 221005

Semester VI - April, 2021

Contents

1 Problem Statement

1.1 Description

Cybersecurity is critical for the internet users' safety and for the nations' defence & security. One of the important aspects of cybersecurity is to "Traceback the Source of the Cyber Attack", which is challenging because of attribution, unlike conventional attacks. The attacker spoofs the source field of the packets to make it appear like it's coming from some other source. The attacker does through, by using a variety of techniques like VPNs, Tor or Jump Host.

1.2 Objective & Scope

The objective of this project is to implement a proposed algorithm for tracing/ localizing the source of the cyberattack. Finding the attack source helps in identifying who is responsible and thus, helps in retaliation by proper authorities. Without knowing the source, retaliation is not possible which makes the attacker more fearless and it may be encouraging for them to conduct more cyber attacks. On the other hand, retaliation deters the attackers and the attack-sponsoring state. Hence, tracing the attack source is very critical for reducing the cyberattacks and extensive research has been done on this field.

1.3 Abstract

We started the project by understanding the cybersecurity threats, various cyber-attacks and challenges involved in finding sources. Then, a literature survey was done on the present algorithms for finding the source of attack. Based on the scope, we selected and implemented the algorithm proposed by O. Demir and B. Khan[1]. The paper proposes SLANT algorithm (short for, 'Searchlight Localization Algorithm for Network Topography') for localizing the attack sources using information collected over sequences of DDoS attacks. After thoroughly understanding the concepts and mathematical model, we implemented the algorithm in Python. We conducted various experiments to check if the implementation is good enough. The results we found in the experiment were as expected based on the paper.

2 Introduction

2.1 Cyberattack

A cyberattack is a malevolent and intentional endeavour by an individual or association to breach another individual or organization's information system. For the most part, the attacker looks for some advantage from intruding on the victim’s network.

The common types of cyberattacks are:

1. **Malwares** are malicious softwares like, viruses, spyware, ransomware, and worms. Malware breaks into a system through vulnerabilities like when the user clicks on a malicious link or on an email attachment.

2. **Phishing** is the act of sending deceitful correspondences that seem to come from a legitimate source, normally through email. The objective is to steal confidential information like login credentials or credit card information or to introduce malware on the victim's device.

3. **Man-in-the-middle (MitM) attack**, also called eavesdropping attack, happens when attackers intrude in a two-party transaction and then the attacker can interfere with the traffic and can channel and steal the victim's information.

4. **Denial-of-service (DoS) attack: In Denial-of-service attack,** servers or networks are flooded with traffic to exhaust the system’s resources and bandwidth. This makes the system incapable of satisfying authentic requests. Attackers can also strike DoS attacks using a number of compromised computers (botnet). This is known as Distributed-Denial-of-Service (DDoS) attacks.

5. **SQL Injection:** A Structured Query Language (SQL) injection happens when an attacker embeds malignant code into a server (that uses SQL) and gets hold of the server’s confidential data using that code. SQL injection can be carried out by submitting malicious code through a vulnerable form, like a search box.

In this project, I will be working on DDos Attack. So, in the next subsection, I will describe DDoS attacks in detail.

2.2 DDoS Attack

DDoS (Distributed denial-of-service) attack is one of the most common cyber-attack. DDoS attack targets websites and online applications by flooding them with more traffic than the server or network can handle. The goal of the DDoS attack is to disable the server from rendering any more requests. The traffic with which the server is flooded consists of fake packets, connection requests or some messages.

The DDoS attack is a distributed attack, that is, it is accomplished using a number of internet-connected devices. The devices which are used in the attack are infected by some malware, which allows the attacker to control these devices remotely. These compromised machines that the attacker uses are called ‘bots’ and the group of these infected devices is together called ‘botnet’.

After establishing the botnet, the attacker sends remote instructions to the botnet. During an attack, each bot sends request to the targeted server/ network, which may overwhelm the server’s capacity. Since each botnet is a legitimate device, it is difficult to distinguish between legitimate tariff and attack tariff. Also, since the attacker uses a large number of devices to conduct this attack, it is very difficult to find the source of attack.

3 Papers Surveyed

For this project, we surveyed 3 papers:

1. Spinelli et al.[2] proposes a general framework that uses static and dynamic sensors (sensors are nodes that report its state and time of infection) and models the epidemic using a mathematical graphical model to perform source localization. The approach used by the author is generalized and can be applied to find the source of computer worm/malware.

2. Omer Demir and Bilal Khan[1] proposes SLANT algorithm (short for, 'Searchlight Localization Algorithm for Network Topography') for localizing the attack sources using information collected over sequences of DDoS attacks. The proposed method is based on the assumption that the same set of attackers (botnets) execute various attacks over time. This assumption is based on the argument that a Botnet requires significant investment and exposure to risk, and hence, is likely to be used multiple times.

3. Dong et al.[3] proposes a deep learning-based model, GCNSI (Graph Convolutional Networks based Source Identification), to locate multiple rumour sources, without prior knowledge of the underlying propagation model. The method uses the traditional Graph Convolutional Networks technique but modifies the input and loss function of the traditional network. The GCNSI model outperforms the state-of-the-art model LPSI, NetSleuth and Zang on five real-world network datasets.

Out of these 3 papers, we chose to implement the 2nd paper, because of the following reasons:

- SLANT algorithm has high accuracy and is scalable to large networks.
- We were unable to find a proper dataset that could be used to train GCNSI kind of network for attack source localization (since, malware spreads in the same pattern like an epidemic). We even submitted a report on the datasets we have surveyed.

4 Finding DDoS Attack Sources

O. Demir and B. Khan proposes a novel technique that makes use of information obtained over sequences of DDoS attacks to determine the sources of attack. According to the author, Botnet is a capital investment that requires a considerable amount of time & resources, hence it is very unlikely that the attacker would use a botnet merely for a single attack.
The author proposes the system based on the following assumptions:

- the same Botnet carries out a sequence of attacks over time, on different victims
- Network topology, Routing table and the attacker nodes remain constant over the set of attacks.

This idealized setting will allow us to evaluate the effectiveness of proposed schemes in aggregating attacker location information from multiple attacks over long time scales

Mathematical Model -

Let G = (V, E) be a network with nodes V and bidirectional links, $E \subset V \times V$.
The routing tables of the network is represented by a function, $R: V \times V \rightarrow V$ which is in agreement with the set of links, E.
Flow from d to v, represented by F(d,v), with respect to the routing table R is defined as:

- F(d,v) = (f(d, v, i) ; i = 0, 1, ...)
- f(d, v, 0) = d, and
- f(d, v, n + 1) = R(f(d, v, n), v)

4.1 Flow Reconstruction Problem

In this mathematical model, each DDoS attack is modelled using a flow reconstruction problem. The maximal solution to these flow reconstruction problems, along with the problems, is sent to the SLANT algorithm (defined in the next subsection), which locates the attacking nodes.

A flow reconstruction problem (FRP) is a tuple (A, D, v) where $D \subseteq V$ is a collection of attacking nodes, $v \in V$ is the victim node, and $A \subseteq E$ is some set of DDoS sensor agents. A solution to an FRP instance is a logical network L = (S, E_S) on a subset of agents $S \subset A$ and $E_S \subset S \times S$.

A valid solution L = (S, E_S) is said to be the maximal valid solution to a flow reconstruction problem if:

1. Any agent on the path from an attacker to the victim is included in the solution set of agents, S. Except these agents, no other agent is a member of S.
2. If two agents are linked by a logical connection in E_S, then they must appear sequentially in the flow from some attacker to the victim with no other agent in-between them; and if two agents are sequentially in the flow from some attacker to the victim and there is no agent between them, then there should exist a logical link between them.

4.2 Attacker Localization Problem

The Attacker Localization Problem is used to simulate a sequence of DDoS attacks, represented by FRPs, together with their maximal solutions.

Formally, Attacker Localization Problem (ALP) is a collection of FRPs, $P = (P_1, P_2, ..., P_n)$ where each $P_i = (A, D, v_i)$ differs only in the target identity v_i; along with a collection of corresponding maximal viable solutions $L = \{L_i = (S_i, E_{S_i}) \mid i = 1, \ldots, n\}$.

A solution to an instance of ALP is a suspicion map, s: $V[G] \rightarrow R$ which assigns a suspicion degree for each vertex, v ∈ V. Using a fixed threshold τ ∈ R, a suspicious set, SS(τ) is calculated as SS(τ) = {v ∈ V |s(v) > τ}.

To evaluate this solution, two measures are defined:

1) False Positive Rate (FPR): It represents the fraction of suspects that are not attackers.

$$FPR(\tau) = \frac{|\{v \in V \backslash D | s(v) \geqslant \tau\}|}{|\{v \in V | s(v) \geqslant \tau\}| + \epsilon},$$

where ∈> 0 is a very small real number to prevent the denominator from being zero.

2) False Negative Rate(FNR): It represents the fraction of attacking nodes that are not in suspicion set

$$FNR(\tau) = \frac{|\{v \in D | s(v) < \tau\}|}{|D|}$$

4.3 SLANT Algorithm

The author describes an algorithm that can be applied to solve ALP instances - the Searchlight Localization Algorithm for Network Tomography (SLANT) algorithm.

Given an ALP instance (P, L) the SLANT algorithm outputs a suspicion map, s.
The algorithm operates as following:

- Defining s(v) := 0 for all v ∈ V .
- For each attack i, where i = 1, . . . , n:
  – For each DDoS sensor agent $a \in S_i$, compute the cone $C(a, v_i)$.
  – Then,for all u ∈ V , compute
    $\Delta\tau\ (u,\ i)\ = \Sigma_{a \in S_i}\ \sigma(u,\ C(a,\ v_i),\ S_i)$
  – Update s(v) := s(v) + Δτ (u, i)/n.
- Output the suspicion map s.

where, cone C(a,v) for agent a and node v is defined as: C(a, v) = {u ∈ V |a ∈ F (u, v)}
and σ is the multi-agent information combining function, denied as:

$$\sigma(u, C, S) = \begin{cases} 1/|S| & \text{if } u \in C \\ 0 & \text{otherwise.} \end{cases}$$

5 Implementation & Experiments

We have implemented the source code of the project using the Python language and Spyder IDE.The project can be found on the following link on github:
https://github.com/pulkitt15/ddos-attack-detection
We have followed the Object-Oriented Programming approach for this project. The project consists of following classes:

- Class routing_table: Constructs the routing table for a given graph
- Class FRP_solver: Takes input an FRP instance and outputs it maximal solution
- Class ALP_solver: Class that implements SLANT algorithm and its accompanying functions, cone and σ
- Class network: class that simulates an artificial network for running and testing of the proposed algorithm

To show that the implementation of the project is good enough and works well, we have repeated the experiments that have been conducted by the author.

The author has simulated DDoS attacks in a grid network with one attacker |D| = 1.
If number of nodes are |V|, a $\sqrt{V} \, x \, \sqrt{V}$ grid is constructed. Each lattice point represents a vertex of the network. All horizontal and vertical links between consecutive pairs of nodes constitute the edge set, E. The routing table is constructed using Dijkstra's algorithm from each node. If agent density is p, k is defined as 1/p and an agent is placed along every kth horizontal and along every kth vertical link.

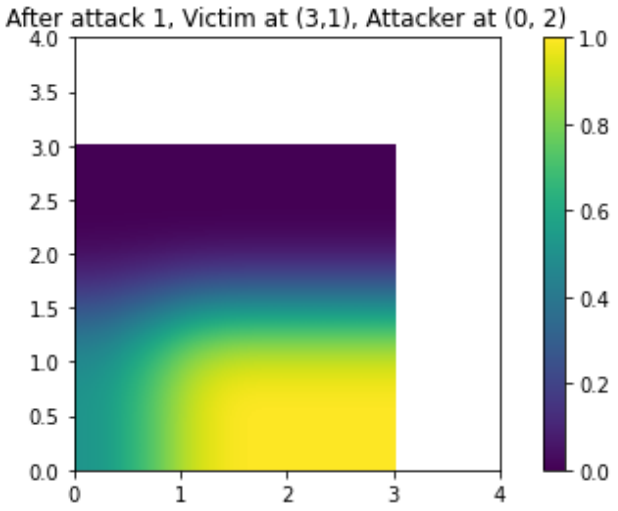
After establishing the network topology and routing table, the attackers are selected randomly from all vertices to make set D. The number of attacks is taken as input, n. For each attack, a victim is chosen from the set of non-attacking nodes, with replacement.
Hence, the grid experiment is characterized by |V|, |k|, |D| and n.

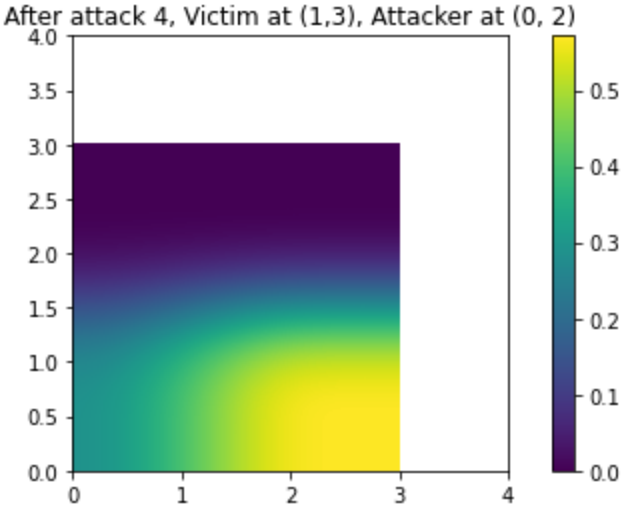
We have also developed a basic interface for this simulation. The following figure shows our code interface for a grid network along with its output:

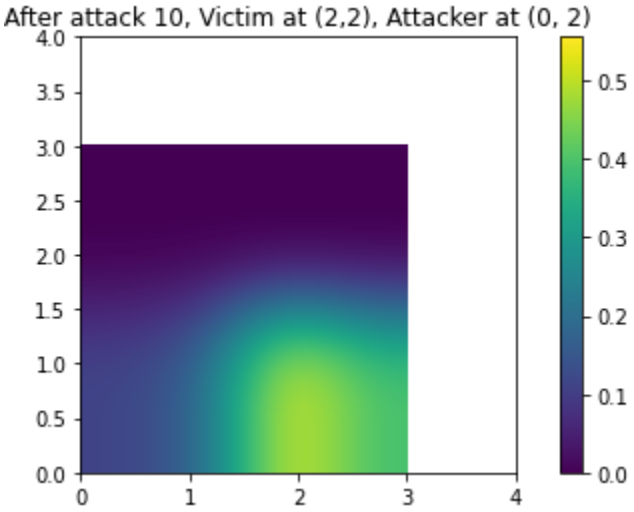
```
Enter number of nodes: 10

Enter agent density (in percentage): 50

Enter number of attacker nodes: 4

Enter number of attacks: 12

Enter threshold: 0.6
The attacker nodes are:  [(1, 1), (1, 0), (0, 0), (0, 1)]
Suspicion set size is 4
FPR score is 0.0

Do you want to print whole suspicion map(Y/N)? Y
Suspicion map:
{(0, 0): 0.8333333333333333, (0, 1): 0.8333333333333333, (0, 2): 0.16666666666666666, (1, 0): 0.8333333333333333, (1, 1): 0.8333333333333333, (1, 2):
0.16666666666666666, (2, 0): 0.5833333333333334, (2, 1): 0.5833333333333334, (2, 2): 0.0}

FNR score is 0.0

Do you want to run again with different threshold value(Y/N)? N
```

5.1 Visualizing SLANT localization

The objective of this experiment is to investigate how color map varies as the number of attacks increases. For this experiment, a 16 node grid topology is used. The experiment has only one attacker at (0,2) and the agent density has been set to 50%. The below figure shows suspicion map s, after n=1,4 and 10 attacks have been done.



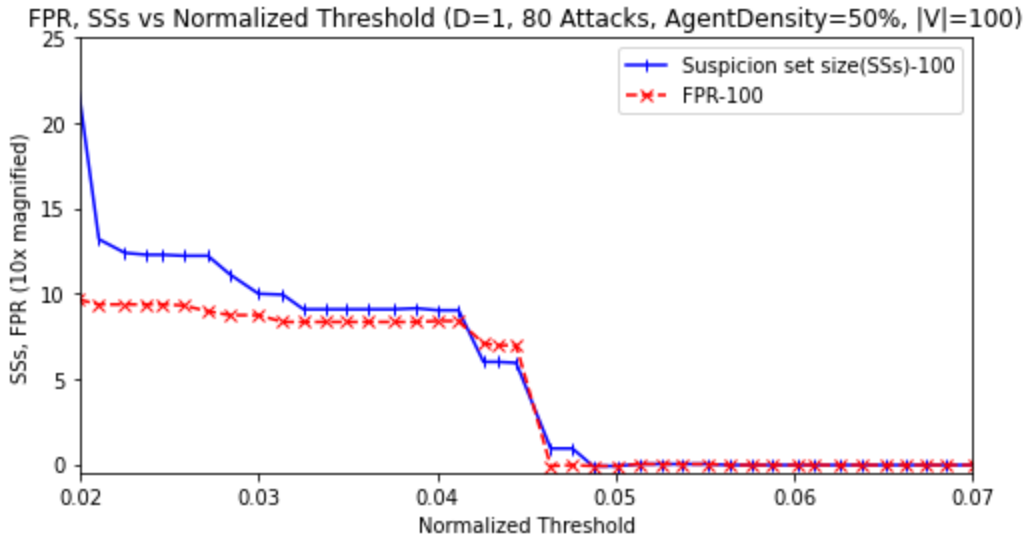


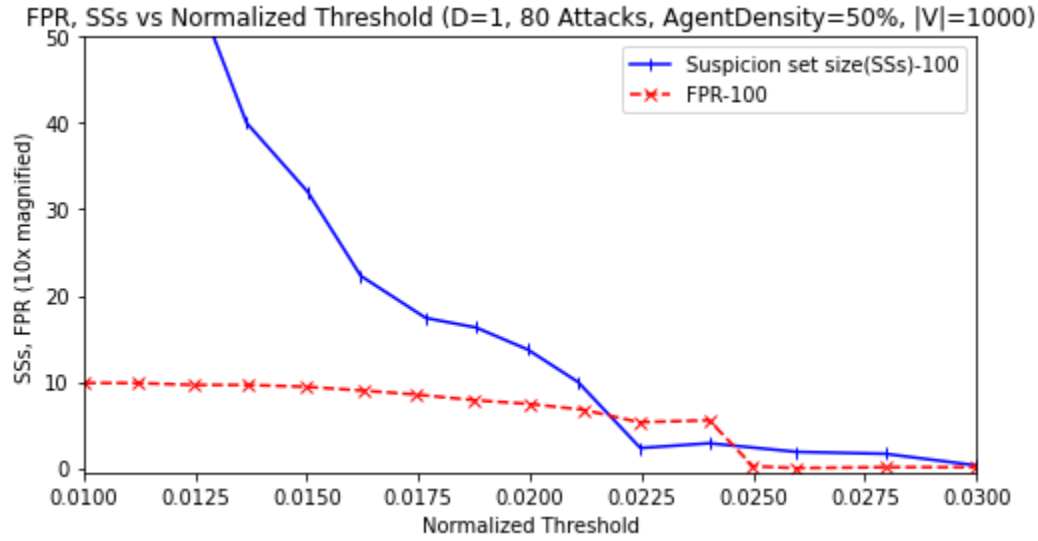


The colour distribution depicts the suspicion scores, which have been normalised against the highest value of suspicion map,s. The series of figures depicts the functioning of the SLANT algorithm and provides an understanding of its objective.

5.2 Varying network size

We now change the network size to 100 and 1000 and see how |SS| and FPR varies as the normalised threshold, τ varies.





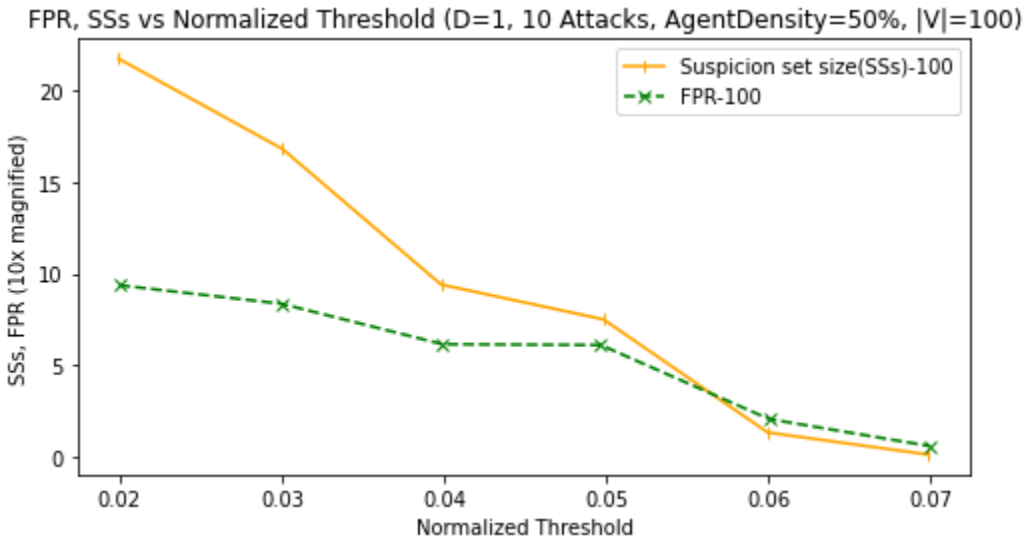
Since the suspicion level assigned to each vertex is constant, as we increase the threshold, suspicion set size decreases, and since FPR is an indication of innocent nodes that are in suspicion set, FPR also decreases.
For first graph with |V| = 100, |SS| = 1 and FPR = 0 for τ = 0.046-0.048. So, the one vertex in SS for which FPR = 0 is the true sole attacker

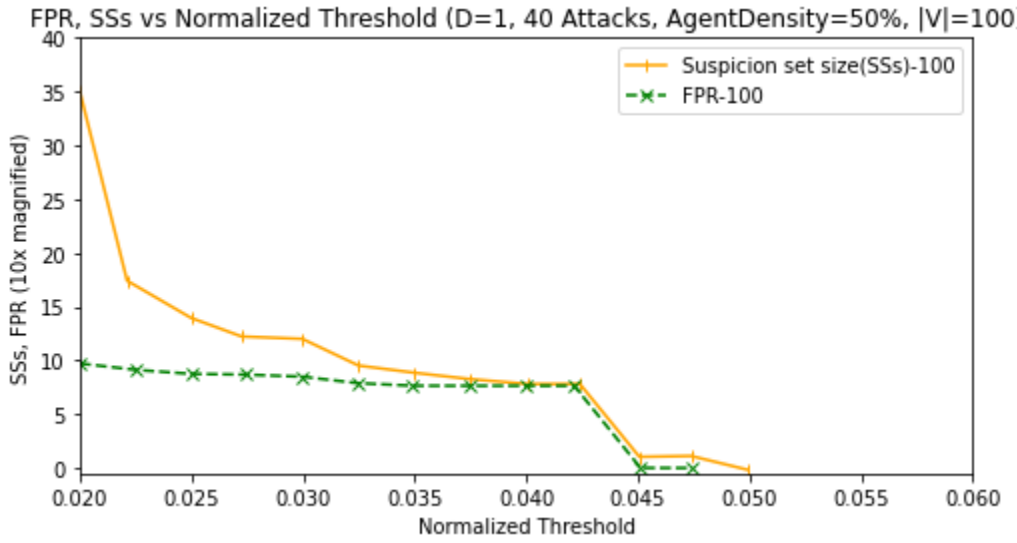
For the second graph with |V| = 1000, for τ = 0.0263-0.0279, |SS| = 1 and FPR = 0. So, for threshold in this range, the suspicion set contains only one attacker, which is the real attacker.

Hence in both the cases, SLANT algorithm showed promising results by identifying the real attacker(s).

5.3 Varying the number of attacks

The objective of this experiment is to illustrate the effect of number of attacks,n on |SS| and FPR. For this, a 100 node grid topology is used, with only one attacker and 50%. agent density. The number of attacks are set to 10 and 40.





When n=10, |SS| and FPR both drops to zero simultaneously, in which case the attacker cannot be identified. But when n=40, the attacker can be specified as at

τ = 0.045-0.0475, |SS|=1 ,but FPR = 0, so the attacker can be determined. Hence, number of attacks,n are critical for the success of the SLANT algorithm.

6 Conclusion

The experiments conducted show that the implemented algorithm works as expected and the results obtained are also similar to the results obtained in the paper. Hence, we can conclude that our implementation of the system is good enough and with this code, it is possible to localize the attack source using the information obtained over a series of DDoS attacks.
In terms of scalability to vast networks and the number of attacking nodes, the algorithm's efficiency is encouraging. As a result, the SLANT algorithm solves instances of the recently defined attacker localization problem. Under an appropriate threshold, SLANT's solutions have a low FPR and modest suspicious set sizes.

7 References

[1] O. Demir and B. Khan, "Finding DDoS attack sources: Searchlight localization algorithm for network tomography," 2011 7th International Wireless Communications and Mobile Computing Conference, Istanbul, Turkey, 2011, pp. 418-423, doi: 10.1109/IWCMC.2011.5982570.

[2] B. Spinelli, L. E. Celis and P. Thiran, "A General Framework for Sensor Placement in Source Localization," in IEEE Transactions on Network Science and Engineering, vol. 6, no. 2, pp. 86-102, 1 April-June 2019, doi: 10.1109/TNSE.2017.2787551.

[3] Dong, Ming, Bolong Zheng, Nguyen Quoc Viet Hung, Han Su, and Guohui Li. "Multiple rumor source detection with graph convolutional networks." In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 569-578. 2019.