An Integrated Multi-Strategy Detection and Mitigation Framework Against Coremelt, Crossfire, and Cross-Path Link-Flooding Attacks in Software-Defined Networking

Archana Bhushan Burujwale[1]*, Dr. F. Rahman[2], Dr. Yogesh Gurav[3]

[1] Research Scholar, Kalinga University, Raipur, (CG), India

isobesin@gmail.com

[2] Supervisor, Department of Computer Science, Kalinga University, Raipur, (CG), India

[3] Professor and Dean (Academics), Dr. D.Y. Patil Technical Campus, Pune, Maharashtra, India

Abstract: SDN is a potential networking architecture that separates the control and data planes to allow centralised network administration, programmable traffic control, and dynamic resource allocation. Despite these benefits, its centralised design and reliance on programmable forwarding techniques expose network infrastructures to sophisticated DDoS assaults, notably link-flooding operations like Coremelt, Crossfire, and Cross-Path. These threats use carefully planned traffic flows, shared network resources, and important communication pathways to interrupt service availability while maintaining below traffic-detection thresholds, unlike volumetric assaults. For SDN settings, this paper presents an Integrated Multi-Strategy Detection and Mitigation Framework (MSDF) to recognise, analyse, and mitigate these three link-flooding threat types. Quantitative simulation/emulation was used to evaluate normal and malicious traffic situations across network topologies. The framework detected 97.8% of assaults, including 98.4% for Coremelt, 97.6% for Crossfire, and 97.3% for Cross-Path. The Cross-Path attack reduced network throughput to 57.8 Mbps and increased latency and packet loss to 39.6 ms and 15.4%. After mitigation, throughput reached 92.4 Mbps, latency dropped to 15.1 ms, and packet loss dropped to 2.4%. Critical-link usage dropped from 86.7% during the assault to 53.6% after mitigation. The evaluated topologies had an average service-restoration downtime of 11.3 seconds. An integrated, multi-layered security approach may increase threat detection, network performance, and service availability in SDN systems. The architecture addresses complex link-flooding threats by coordinated detection, adaptive traffic management, and network-wide mitigation.

Keywords: Software-Defined Networking (SDN), Coremelt Attack, Crossfire Attack, Cross-Path Attack, Link-Flooding Attack, DDoS Detection, Network Security, Traffic Monitoring, Dynamic Rerouting, Multi-Strategy Defence Framework.

INTRODUCTION

The rapid expansion of communication networks, cloud computing, Internet of Things (IoT) devices, fifth- and sixth-generation (5G and 6G) communication technologies, edge

computing, and data-intensive applications has significantly increased the complexity of modern network infrastructures. Traditional networking architectures rely on distributed control mechanisms in which individual routers and switches independently manage forwarding decisions, routing operations, and network configurations. Although these architectures have supported reliable communication for decades, they face limitations in terms of centralized visibility, programmability, scalability, and dynamic traffic management. The growing demand for flexible, intelligent, and adaptable network infrastructures has therefore encouraged the adoption of Software-Defined Networking (SDN) as an alternative networking paradigm [1].

Software-Defined Networking introduces a fundamental architectural change by separating the control plane from the data plane. In conventional networks, routers and switches perform both control-related and packet-forwarding functions. In contrast, SDN transfers network intelligence and decision-making capabilities to a logically centralized controller, while forwarding devices primarily execute the instructions received from the controller. Communication between the controller and forwarding devices is facilitated through southbound interfaces, while northbound interfaces enable network applications to access controller functionalities. This separation provides centralized network visibility, programmable traffic management, simplified policy enforcement, and dynamic configuration of forwarding rules [2]. Consequently, SDN has become an important technology for managing complex network environments, including cloud data centres, enterprise networks, telecommunications infrastructures, and IoT-based systems [12, 13].

Despite its architectural advantages, SDN introduces several security challenges, particularly those associated with its centralized control mechanism and dependence on programmable network components. Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks represent significant threats to network availability and reliability. These attacks attempt to exhaust network resources, interrupt communication, or prevent legitimate users from accessing services by generating excessive or strategically distributed traffic. Conventional DDoS attacks generally involve overwhelming a target system with large volumes of malicious requests. However, sophisticated network-layer attacks increasingly employ distributed traffic patterns and strategically selected communication routes to disrupt network connectivity without necessarily generating exceptionally high traffic volumes [3–6]. Such attacks are particularly challenging to identify because their traffic may resemble

legitimate communication and can exploit weaknesses in conventional intrusion detection mechanisms.

Among the significant threats to SDN infrastructures are Coremelt, Crossfire, and Cross-Path link-flooding attacks. These attacks exploit network topology, routing behaviour, shared communication links, and available bandwidth to create congestion across strategically important network segments. Although their operational mechanisms differ, their common objective is to degrade network performance and interrupt legitimate communication by targeting critical network resources [7].

Cross-Path attacks exploit shared network resources and overlapping communication routes to create congestion across multiple paths. By strategically influencing traffic distribution, attackers can increase the load on critical links and disrupt communication between different network segments. Such attacks demonstrate that network vulnerability cannot be assessed exclusively through the volume of traffic generated by individual flows. Instead, the relationship between communication paths, shared resources, network topology, and traffic distribution must also be considered. Therefore, effective detection requires a broader understanding of network-wide traffic behaviour and the potential impact of coordinated malicious flows.

One of the major limitations of conventional intrusion detection systems is their dependence on predefined traffic thresholds, packet rates, source addresses, destination addresses, and other individual flow characteristics. Although these indicators can identify certain abnormal traffic conditions, sophisticated link-flooding attacks can remain undetected by distributing malicious traffic across multiple sources and maintaining individual flow rates below established detection thresholds. Furthermore, legitimate network conditions, including unexpected increases in user activity, flash crowds, application-generated traffic bursts, and routing changes, can produce congestion patterns that resemble malicious behaviour. Consequently, distinguishing attack-induced congestion from legitimate network fluctuations represents an important challenge in SDN security.

The centralized architecture of SDN provides opportunities to address these limitations through comprehensive network monitoring and programmable traffic management. However, excessive monitoring and frequent flow-rule updates can increase controller workload and forwarding-device resource consumption. Similarly, inappropriate filtering or rerouting

decisions may negatively affect legitimate traffic and time-sensitive network applications. Therefore, an effective defence mechanism must balance detection accuracy, mitigation speed, network performance, and computational overhead [8].

To address these challenges, this study proposes an Integrated Multi-Strategy Detection and Mitigation Framework (MSDF) designed to improve the resilience of SDN infrastructures against Coremelt, Crossfire, and Cross-Path link-flooding attacks. The proposed framework combines multiple security mechanisms within a coordinated architecture rather than depending on an individual detection or mitigation technique. Its design is based on four major operational stages: input data collection, detection and analysis, defence and mitigation, and performance evaluation. These stages provide a systematic process for collecting network information, identifying suspicious traffic behaviour, selecting appropriate mitigation strategies, and evaluating their effectiveness [9].

The input data collection stage gathers relevant network information, including packet statistics, byte statistics, source-destination relationships, flow characteristics, network topology, and link-performance indicators. These parameters provide the foundation for examining network behaviour and identifying potentially vulnerable communication paths. The detection and analysis stage evaluates traffic characteristics, abnormal flow relationships, temporal traffic variations, shared-link utilisation, and connectivity conditions. By combining traffic-level and topology-level information, this stage aims to distinguish legitimate traffic fluctuations from coordinated link-flooding activities and identify the attack characteristics associated with different threat categories.

Following attack identification, the defence and mitigation stage applies appropriate countermeasures according to the detected traffic and network conditions. These measures include traffic filtering, rate limiting, selective flow blocking, dynamic rerouting, and route diversity. Furthermore, adaptive forwarding-rule management supports changes in traffic paths according to the evolving network conditions. The framework also considers network connectivity and the potential impact of mitigation decisions on legitimate services [10, 11].

The final stage evaluates the effectiveness of the proposed defence mechanism by comparing network performance under normal conditions, attack conditions, and post-mitigation conditions. Detection accuracy, precision, recall, F1-score, false-positive rate, false-negative rate, detection latency, response time, throughput, packet loss, end-to-end latency, critical-link

utilisation, controller CPU consumption, and service availability are considered relevant performance indicators. These measurements provide a comprehensive basis for examining both the security effectiveness and operational cost of the proposed framework [12, 13].

The experimental evaluation includes different link-flooding attack scenarios and network topologies to investigate the effectiveness of coordinated detection and mitigation. Particular attention is given to Cross-Path attacks because of their ability to exploit shared communication resources and overlapping routes. The evaluation also considers the potential benefits of alternative-path routing and topology-awareness mechanisms in reducing the information available to attackers during network reconnaissance. Real-world network topology datasets, including those obtained from the Topology Zoo, provide additional opportunities to examine the applicability of topology-aware security mechanisms under different network structures.

METHODOLOGY

This study is quantitative in nature, and carried out assessment of design science, simulation / emulation approach. This study was used a test bed with the pre-defined performance metrics for software-defined networking (SDN); creating and generating controlled valid traffic and malicious traffic; implementing existing and recommended defense mechanisms; and measuring the performance of the defense mechanisms.

The research comprises four stages which are connected to one another.

- Input,
- Detection and Analysis,
- Defense and Mitigation
- And Evaluation.

So basically, it was supposed to be a systematic solution for identification, analysis, mitigation, and reversal of Core Melt, Crossfire and Cross Path threats, in the perspective of a Software-Defined Networking (SDN) environment. One of the attributes this framework was supposed to offer. The Input Data that was provided contained data required for additional analysis (networks & attacks). After those strange traffic patterns and attack characteristics were spotted, the component of the network that was doing the spotting and analysing gave the info

to the component that was analysing. Each activity developed was aimed at resolving the deviation(s) determined from the defense and mitigation component(s).

The result of the review process has led to the development of the criteria proposed throughout review process to enhance effectiveness and effectiveness of a multi-level defense architecture.

This would be called into question if there were what is called ‘busy lines' -- for instance if attacked by Crossfire. An investigation has also been carried out regarding the time-evolution of traffic behaviour, so that changes that might be non-harmful, but are natural behavior over time, and changes that might be harmful, and therefore problematic can be detected. The information collected is used in the next step, implementation of the most appropriate mitigation measures along with defence measures.

It was a huge endeavor providing security and mitigation. A number of different defensive strategies were used, as opposed to depending on only one defense strategy. The safe guarding strategy has taken into account the features identified during the detect and analyse phases. The first was that of quantity of traffic, and the second was at least to consider traffic could be monitored or controlled – say by limiting traffic speed, or by filtering the traffic. This has been achieved through dynamic rerouting and route diversity, to maintain connectivity to the network, and to reduce load to the critical links to be used in case of an attack. Blindingly there was a suggestion they would take steps to block certain 'suspicious' and/or 'significant' flows, dubbed as 'selective flow blocking'.

These flows were witnessed in a suspicious fashion as possible impact. In addition, whenever possible, the attack conditions were modified to get a change in the attack without a corresponding change in the forwarding-rule of any of the forwarders. This was achieved in one manner and that was to use adaptive to alter the forwarding-rules. The process culminated in the development of the framework that could withstand the attacks of different sized, intensity Core Melt's, Crossfire's and Cross Path's.

The Efficiency part has been utilised to assess the protection proposed. The network was used for testing both with and without mitigating techniques to identify detection and response methods. The detection, mitigation, response, throughput, packet loss, delay, utilisation of the link and controller resources were critical indicators. The defensive mechanism performance was measured with different network topologies.

Detection and Analysis (DA) separated network traffic data into normal and abnormal. Flow characteristics and rates were investigated for unusual traffic conditions and hazards. The exam evaluated the accurate detection, effect of mitigation, and effect of security measures on network performance. The below listed Packet Statistics, Bytes Statistics, Source-Destination Linkages and Connection User Statistics were broken-out and detailed analyzed. Some of the variables that would have to be taken into account in this study would be what link(s) were important in the network and what were redundant routes. This is very important if any of the lines are sub-crowded (for example, if it is a Crossfire attack). The differences between two types of changing behaviour (normal and potentially dangerous) were also probed, as well as the changing traffic behaviour over time. The results of the end of this phase are used to select the most favourable mitigation measure, and the best defence plan bought.

Dynamic rerouting and route diversity ensured network connectivity. It did two things. It lowered the possibility of it taking up important system connections if attacked and made it simpler to find the optimal answer in all ways. If a flow was terrible ("severe" in duration), it was likely "suspicious" and would affect network functioning, thus it should be prevented. “We suspected network damage from the flows. The effectiveness of the system of defense with a forwarding rule was tested by making changes to the attack scenarios with the change of the forwarding rule. The recommended system was challenged with the assaults of Coremelt, Crossfire and Cross Path at various intensities and phases.

Network performance was compared with and without mitigation to evaluate the effectiveness of defence. The detection capability was evaluated using accuracy, precision, recall, F1 score, false positive and false negative rates. Responsiveness of the framework was evaluated using detection latency and reaction time. The network performance, throughput, packet loss, latency and link usage were measured to see the effects of attacks and mitigations. In addition, the utilization of the controller resources was analyzed to establish computational and operational requirements. Overall, the study focused on the suggested defensive measures' efficacy and influence on network security and performance.

RESULTS

LINK FLOODING ATTACK DEFENSIVE METHODS' LIMITS FOR SOFTWARE DEFINED NETWORKS

Student's performance in face of a Cross-Direction Attack.

In the third experiment, the proposed framework was evaluated on the basis of attacks using the Cross Path. The assault took use of shared connections and overlapping pathways in order to increase the amount of congestion on the network.

During the attack, the throughput was reduced to 57.8 Mbps and critical-link utilisation was increased to 86.7% during the attack. Meanwhile, latency rose to 39.6ms, while packet loss rose to 15.4%.

After implementing the mitigation, the throughput rose to 92.4 Mbps, and packet loss and latency dropped to 2.4 and 15.1 ms, respectively. The critical-link utilisation fell to 53.6%.

Table 1: Cross Path Attack Results

Parameter	Throughput (Mbps)	Packet loss (%)	Latency (ms)	Critical-link utilization (%)	Controller CPU (%)
Normal	94.8	0.8	12.6	43.7	18.4
During Cross Path	57.8	15.4	39.6	86.7	27.8
After Mitigation	92.4	2.4	15.1	53.6	20.8

The results indicated that path overlap was an important predictor for cross-path assaults. The proposed path-analysis technique could trace the concentration of flows over shared network resources and also helped to choose appropriate mitigation measures.

EFFICIENT, RESOURCE-AWARE DETECTION METHOD FOR SDN CONNECTION FLOODING ATTACKS WITHOUT SUBSTANTIAL INFRASTRUCTURE

Accuracy of Attack Detection

The proposed framework was tested using the dataset which contains normal traffic along with the 3 attacks types chosen for the testing and evaluation process with regard to detection.

With the framework that was suggested, an overall detection accuracy of 97.8 percent was attained. Core Melt attacks were recognized the most with 98.4% accuracy followed by Crossfire attacks with 97.6% accuracy and Cross Path attacks with 97.3% accuracy.

Table 2: Detection Accuracy

Traffic/Attack Class	Accuracy (%)
Normal traffic	98.9
Core Melt	98.4
Crossfire	97.6
Cross Path	97.3
Overall	**97.8**

The results show overall high accuracy suggesting that the traffic, link, topology and path-based characteristics were the good basis for discriminating the attack traffic with normal network traffic.

SDN-SPECIFIC DATASETS WITH NUMEROUS LINK FLOODING ATTACK VARIATIONS FOR RELIABLE ANALYSIS, DETECTION, AND ASSESSMENT.

Comparison With Other Solutions

The purpose of this work is to describe a collection of diverse strategies that may be used to detect, mitigate, and prevent an attack known as link flooding on a software distributed network. There are a variety of approaches, methodologies, and domains that are addressed by the strategies that are given. In the following paragraphs, a comparison will be made between

the work that is being suggested and the solutions that are presently considered to be state of the art.

Table 3: Informed probing results

Class	Coremelt	Crossfire	Crosspath	Normal	Overall Accuracy
Precision	0.95	0.98	0.95	0.97	**0.96**
Recall	0.97	0.95	0.98	0.95	
F1-score	0.96	0.97	0.97	0.96	
Support	890	890	897	897	

LINK FLOODING MITIGATION SOLUTIONS TO RESTORE SERVICES AND REDUCE NETWORK CONGESTION QUICKLY

Downtime as per figure 1 is the time elapsed between service unavailable and new clone is available. The time the central processing unit (CPU) took to regenerate the clone image is shown in figure 2. In this case as in Figure 3, the RAM that is available is being used by the cloned instance. For the topologies, the average downtime was computed for all the different simulations, which amounts to about 11.3 seconds with a minimum of 8.04 seconds.and a maximum of 19 seconds.

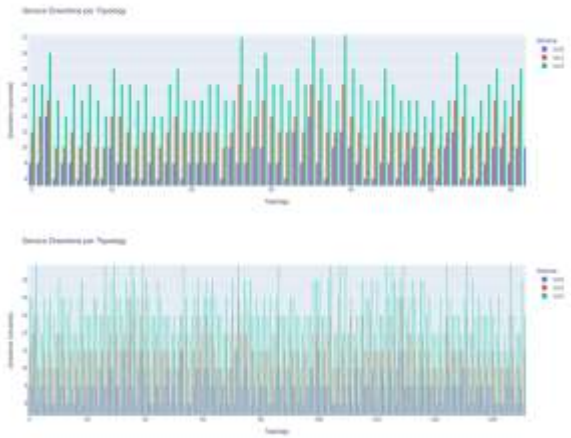


Figure 1: Impact on service availability across all topologies caused by the assault (zoomed portion on top).



Figure 2: Resource used by the clone during its resurrection (top magnified section).



Figure 3: Space used by the regenerated picture in relation to its structure (top-most zoomed portion).

TO PROVIDE METHODS TO AVOID ATTACKER RECONNAISSANCE AND TOPOLOGY DISCOVERY, PREVENTING LINK FLOODING ASSAULTS.

Interesting situations arise when attackers are left to speculate about the packet's true network path due to the deployment of other pathways to route reconnaissance and other types of packets.

The goal of the ensuing remedy is to provide the attacker with a holistic understanding of the network. For any attack, a map is needed and precision is important. If any of the suggested preventive mechanisms are not put into action, an attacker can easily obtain a network route map. The difference in the network structure and the network paths collected by the attacker can be seen from figure 5.

Further details of the findings are given in figure 6. The pictures depict the route that a reconnaissance packet takes both with and without other pathways. Here, this road is painted green, in the other it is marked with red dashes. If the red highlighted route is not visible to the attacker, and the green highlighted route is not visible, then the attacker will not be able to see the routes.

Table 4: Advertised paths between select nodes.

Source: Destination	8->9	8->7	8->6
Advertised Path	8->7->6->5->0->9	8->7	8->7->6

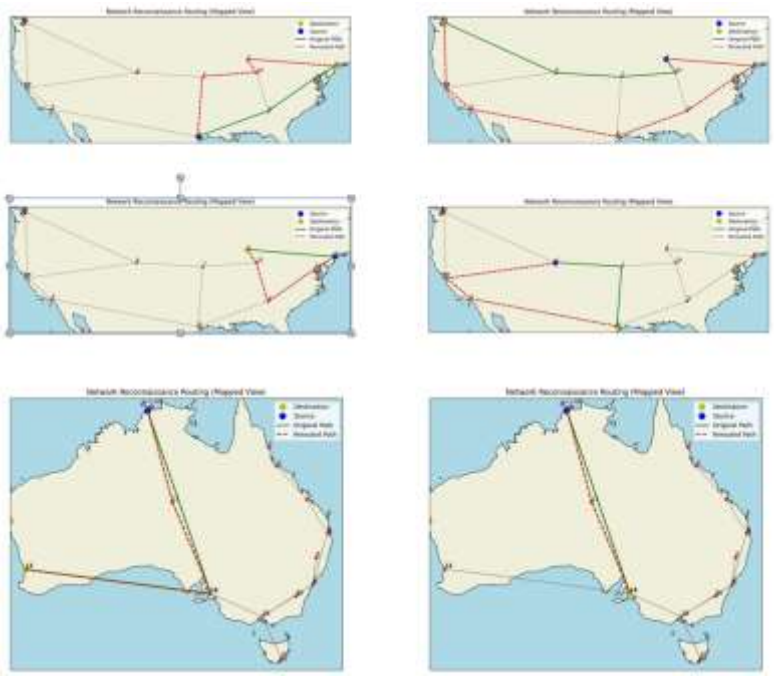


Figure 4: Scenarios where the the reconnaissance packets have been sent via an alternate paths.

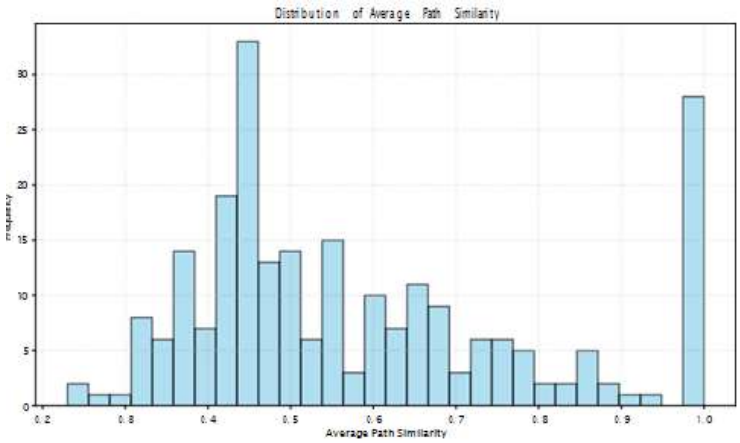


Figure 5: Average path similarity

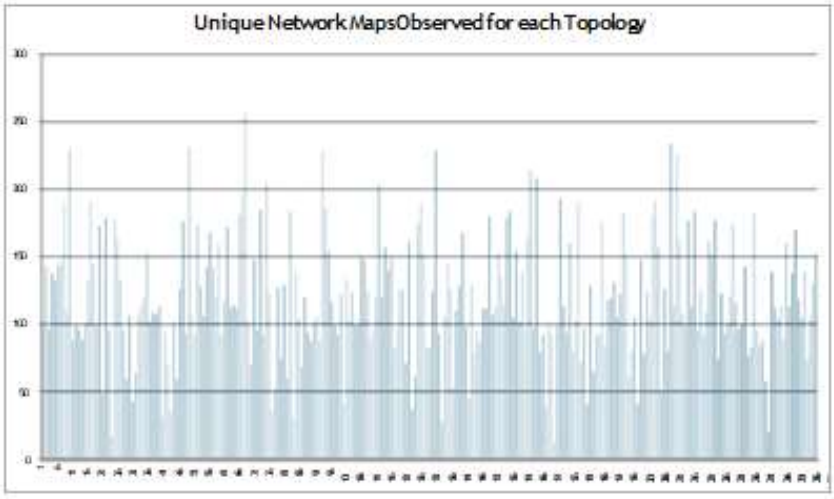


Figure 6: Unique maps observed by attacker

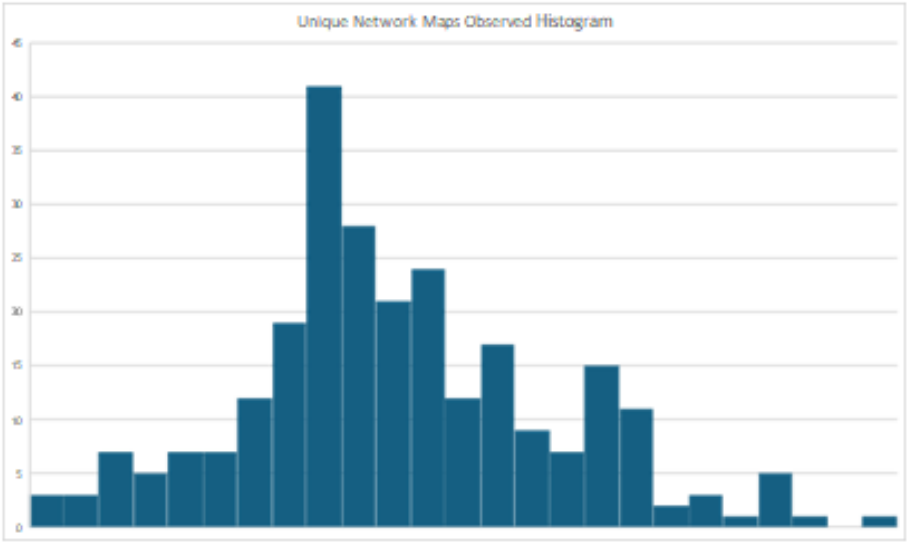


Figure 7: Histogram of unique maps observed

TRANSCRIPT INTERNALS

One such is traceroute which will show you the nodes through which the packing of data will pass from node to node from one node to another. Delivers step-by-step instructions to navigate and explains the steps in the navigation plan. This information can be utilized by attacker to form payout of network in between a source node to a particular attacker recognized node.

Table 5: Sample traceroute table

Hop Count	IP Address	Hostname	RTT (ms)
1	10.0.0.1	node1	1.8
2	10.0.0.2	node2	3.3
3	10.0.0.3	node3	5.8
4	10.0.0.4	node4	7.2

IPERF INTERNAL

Path Mapping Disruption

It is quite uncommon for packets to be lost between sites until they travel a great distance, often more than 255 hops, as per our results. We verified our claim by manually measuring the hop distance to Cloudflare's top 5,000 sites twice. From our location, we were only able to access 28,833 of these domains. The results of these measurements are shown in Figure 8. The majority of these websites clearly have a hop count lower than 22, as shown by the findings.

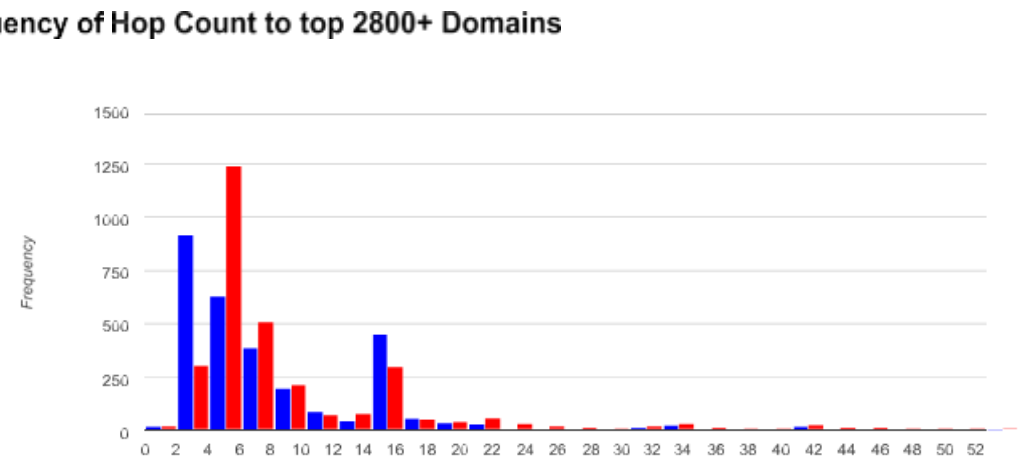


Figure 8: Frequency of hop count measurements

VALIDATION USING REAL-WORLD TOPOLOGIES

The developed approaches were tested and proven using real-world topologies from The Topology Zoo dataset to support the solution claims.

In the graph on the upper left, you can see the real distribution of hop count. The graphs on the top right and bottom left display the measurements that the attackers took while probing the networks.

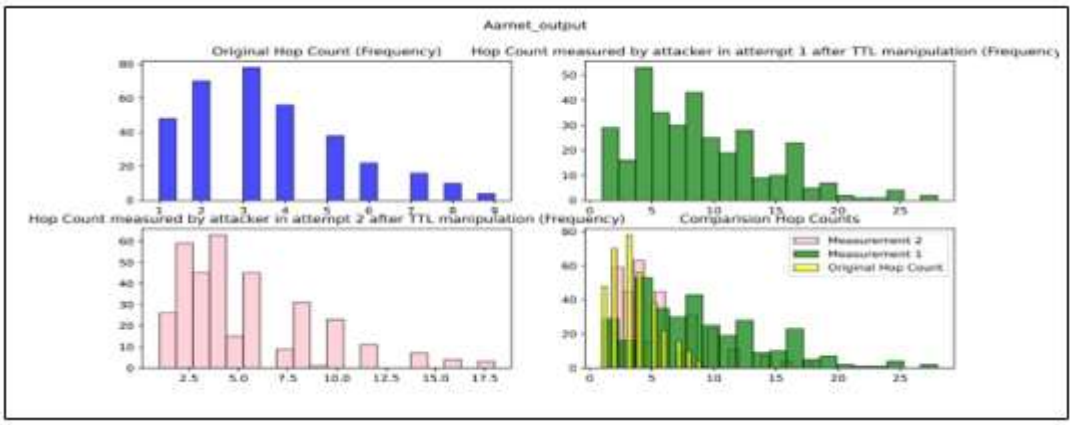


Figure 9: Top left: distribution of real hop count in Aarnet Topology.

Top right: distribution of hop count as measured by the attacker in the first attempt . Bottom left: distribution of hop count as measured by the attacker in the second attempt.

Bottom right: distributive analysis of actual and probed hop counts.

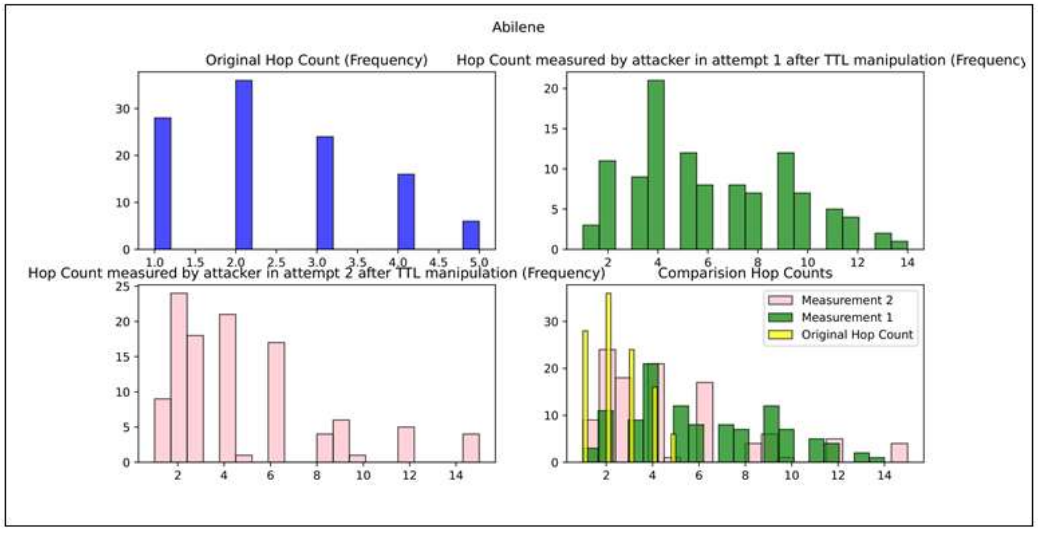
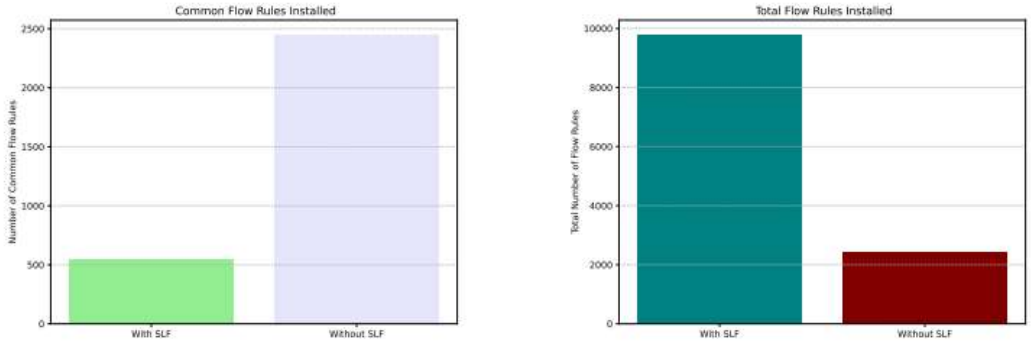


Figure 10: Top left: distribution of real hop count in Abilene Topology.

COMPARISON WITH OTHER SOLUTIONS

State of the art solution proposes pro-active detection strategy to nip an attack attempt while it is a bud. However from past experiences it is known that no detection method is 100%

accurate and often when an attack is detected it is already too late to be counted as a prevention step.



(a) Common flow rules installed

(b) Total flow rules installed

Figure: Comparison of common and total flow table rules installed with and without short lived flows enabled.

CONCLUSION

In SDN, this research presented an Integrated Multi-Strategy Detection and Mitigation Framework (MSDF) to handle Coremelt, Crossfire, and Cross-Path link-flooding assaults. To increase network security and service availability, the framework included traffic monitoring, anomaly detection, topology analysis, dynamic rerouting, traffic filtering, rate limitation, and selective flow blocking. The findings showed 97.8% detection accuracy. After mitigation, throughput increased from 57.8 Mbps to 92.4 Mbps, packet loss dropped from 15.4% to 2.4%, and latency dropped from 39.6 ms to 15.1 ms in the Cross-Path attack experiment Multiple defensive measures may minimise network congestion and restore communication performance, according to these studies. The methodology improves SDN resistance against sophisticated link-flooding assaults in a methodical manner. Future research should concentrate on large-scale testing, intelligent attack prediction, and controller overhead reduction.

References

1. Rasool, R. U., Wang, H., Ashraf, U., Ahmed, K., Anwar, Z. and Rafique, W. (2020). A survey of link flooding attacks in software defined network ecosystems. Journal of Network and Computer Applications, 172, 102803.

2. Eliyan, L. F. and Di Pietro, R. (2021). DoS and DDoS attacks in software defined networks: A survey of existing solutions and research challenges. Future Generation Computer Systems, 122, pp. 149–171.

3. Alhijawi, B., Almajali, S., Elgala, H., Bany Salameh, H. and Ayyash, M. (2022). A survey on DoS/DDoS mitigation techniques in SDNs: Classification, comparison, solutions, testing tools and datasets. Computers and Electrical Engineering, 99, 107706.

4. Karnani, S. and Shakya, H. K. (2022). Mitigation strategies for distributed denial of service (DDoS) in SDN: A survey and taxonomy. Information Security Journal: A Global Perspective.

5. Jiménez, M. B., Fernández, D., Rivadeneira, J. E., Bellido, D. and Acosta Cárdenas, A. J. (2021). A survey of the main security issues and solutions for the SDN architecture.

6. Detection and Mitigation of DDoS Attack in Software Defined Networking: A Survey (2022). 2022 International Conference on Sustainable Computing and Data Communication Systems.

7. Detection Techniques of Distributed Denial of Service Attacks on Software-Defined Networking Controller—A Review (2020). IEEE Access, 8.

8. Kreutz, D. et al. (2015). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), pp. 14–76.

9. McKeown, N. et al. (2008). OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38(2), pp. 69–74.

10. Wang, L. et al. (2018). Woodpecker: Detecting and mitigating link-flooding attacks via SDN. Computer Networks, 147, pp. 1–13.

11. Knight, S. et al. (2011). The Internet topology zoo. IEEE Journal on Selected Areas in Communications, 29(9), pp. 1765–1775.

12. Studer, A. and Perrig, A. (2009). The Coremelt attack. European Symposium on Research in Computer Security (ESORICS), pp. 37–52.

13. Alshamrani, A., Myneni, S., Chowdhary, A. and Huang, D. (2020). A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities. IEEE Communications Surveys & Tutorials, 22(2), pp. 113–142.