International Journal of Information Technology & Management
Vol.20, Issue No. 1, February-2025, ISSN 2249-4510

Detection and Mitigation Techniques against Link-Flooding Attacks in Software-Defined Networking: Coremelt, Crossfire, and Cross-Path Attacks

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 flexible and programmable networking architecture that separates the control plane and data plane for centralised network administration. SDN is susceptible to sophisticated Distributed Denial-of-Service (DDoS) assaults, notably Link-Flooding assaults, despite its benefits in network administration, traffic control, and resource allocation. Because they interrupt network availability via network architecture, common communication connections, and control-plane dependencies, Coremelt, Crossfire, and Cross-Path attacks pose security risks. A multi-strategy defence framework integrating intelligent traffic monitoring, anomaly-based detection, attack classification, adaptive routing, selective traffic filtering, and control-channel protection is proposed to address these three attack types' security issues. The proposed framework detects suspicious traffic patterns, distinguishes link-flooding attack techniques, and initiates SDN controller mitigation measures. Moreover, network topology analysis and dynamic traffic control reduce congestion and retain lawful communication. Performance evaluation includes detection accuracy, precision, recall, F1-score, latency, network speed, packet loss, and controller overhead. The suggested strategy organises coordinated protection against several link-flooding risks to increase SDN infrastructure resilience, availability, and operational dependability.

Keywords: Detection, Mitigation techniques, Link-Flooding attacks, Software-Defined networking, Coremelt attacks, Crossfire attacks, Cross-Path attacks

INTRODUCTION

Flexible, scalable, and dependable networking designs are needed due to the fast expansion of communication technologies, cloud computing, IoT, and virtualised network infrastructures. Traditional networks use dispersed control systems, where routers and switches handle control and forwarding. Although this design supported conventional communication methods, its limited programmability, extensive setup requirements, and limited network-wide visibility make current network administration difficult. Software-Defined Networking (SDN) separates the control plane from the data plane to overcome these restrictions. Centralised network

administration, programmable forwarding, and dynamic resource setup are possible with this separation.

Flexible architecture and centralised administration present security weaknesses. Dependence on the SDN controller for network-wide decisions is a risk. Since the controller controls forwarding activities, its absence or resource depletion may impact many network devices. The restricted capacity of Ternary Content Addressable Memory (TCAM) in programmable switches allows attacks that generate many new traffic flows to deplete flow-table resources. When control and data traffic share network pathways, the controller-forwarding device communication channel may become congested, manipulated, or disrupted. Thus, although SDN enables improved security management, its architecture needs extra protection against network resource and communication channel threats.

Because malicious traffic may consist of low-rate flows from several origins and destinations, link-flooding assaults are hard to detect. These flows may seem like valid communication and not violate traffic-volume limits. Such attacks operate by locating network connections that carry a lot of valid traffic and sending enough data over them to deplete their capacity. Thus, standard detection systems that focus on source-level traffic volume or destination-oriented behaviour may not defend against these threats. Thus, network-wide traffic monitoring, flow correlation, and topology-based analysis are necessary to detect link-flooding.

Coremelt, Crossfire, and Cross-Path are three major link-flooding threats examined in this study. The Coremelt attack uses hijacked devices to coordinate data flows to congest critical network channels. Malicious traffic may not directly target the victim, unlike DDoS assaults. Instead, the attacker chooses communication channels across important connections, disrupting normal network traffic. Compromised devices provide low-rate data to several public decoy servers in the Crossfire attack. Attacker intentionally chooses destinations so traffic flows over network connections shared with victim's communication channels. Due to crucial connection congestion, the targeted network is isolated.

The Cross-Path attack exploits control plane-data plane communication lines in SDN settings to enhance security risk. In in-band control networks, control messages and data packets may share network routes. This setup allows attackers to clog shared connections and disrupt controller-switch communication. This may interrupt forwarding rule implementation, network monitoring, and SDN architecture.

Coremelt and Crossfire mostly impact data-plane connection, although Cross-Path assaults may compromise control communication. Thus, the three attack types need coordinated but distinct detection and mitigation.

The proposed framework will be tested in an SDN experimental environment utilising Mininet, an OpenFlow-compatible controller, Python-based traffic analysis, and machine learning. The performance will be evaluated under regular traffic and individual and combination assault scenarios. Accuracy, precision, recall, F1-score, detection delay, throughput, packet loss, and controller resource utilisation will be assessed. This review examines the efficacy and practical appropriateness of an integrated defensive method for SDN infrastructure security and resilience.

SDN ARCHITECTURE

Supporting software-defined networking (SDN) has changed the game for network administration because it makes networks more flexible and programmable by isolating control activities from data-forwarding hardware [1]. There are three levels in a typical SDN design: the data plane, the application plane, and the control plane. The SDN controllers are located on the control plane, which also communicates routing choices and instructions to the data plane in order to centrally supervise network operations. Fast and flexible network administration is made possible by this one-of-a-kind department, which enables administrators to set up the network and respond swiftly to new needs and situations.

When the control plane issues a command, the packet routing responsibility falls on the data plane switches [2]. The ability to execute OpenFlow means that many of these switches can carry out complex network operations like routing and security enforcement [3, 4]. Nevertheless, the northbound interface connects the controllers to the application plane, which is also called the management plane. This plane can do a lot of different things. This architecture makes the network more responsive and adaptable to meet increasing operational needs by allowing configuration and management updates in real-time.

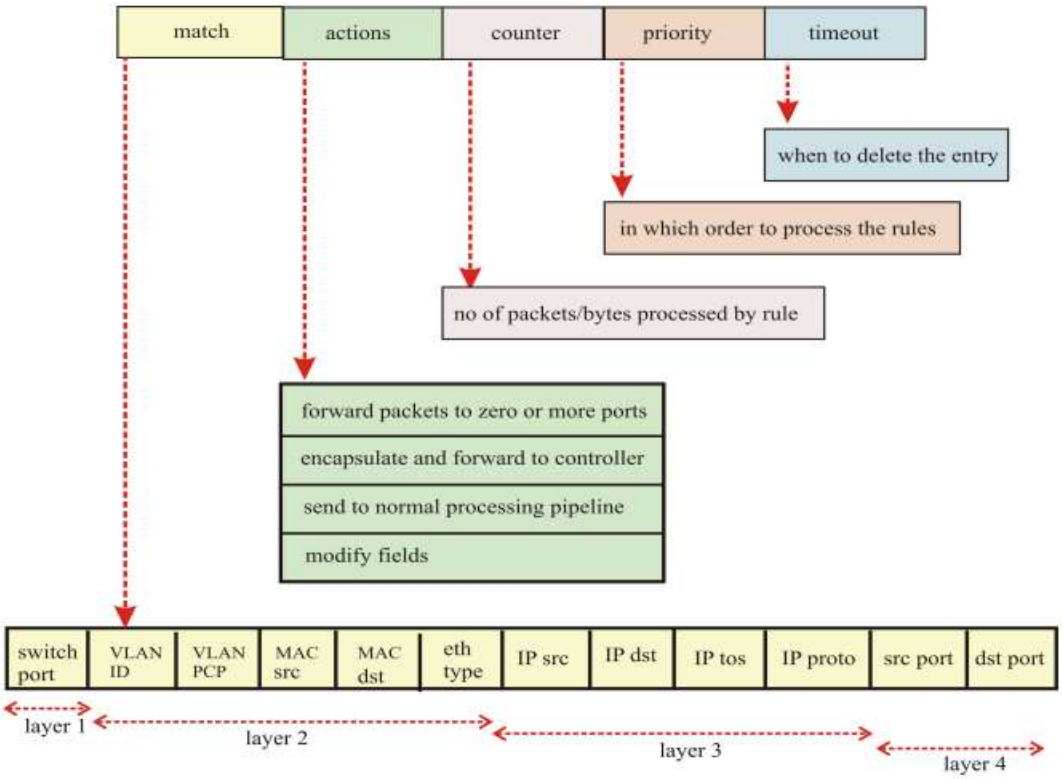


Figure 1: Flow table structure in SDN switches

SDN SECURITY FEATURES

When compared to conventional networks, SDN provides clear benefits, especially in the area of security management, and completely changes the way networks are managed and designed. But because of its design, it also has certain special weaknesses [5]. There are two primary facets to think about when talking about the security of SDN: first, the features that make the network more resilient and protect it from threats like DDoS attacks; and second, the features that could make the network vulnerable to security risks because of flaws in its design. In this article, we discussed the advantages and disadvantages of SDN with regard to security.

SDN Security Advantages

There are a number of characteristics offered by SDN that improve network security:

1. Monitoring in One Place:

Centralised monitoring of traffic and activity is made possible by SDN controllers, which provide a comprehensive perspective of the network [5]. Network managers may improve the

network's efficiency and security with this centralised knowledge since it enables them to quickly discover and prevent abnormalities and possible attacks.

2. Configuration that can be programmed:

One of the key characteristics of software-defined networking (SDN) is programmability, which allows for dynamic configuration changes throughout the network [6]. Upon discovery of a security vulnerability, the network configuration may be immediately adjusted to mitigate exposure, address vulnerabilities, or isolate affected areas. A significant improvement over earlier networks, which sometimes need human intervention at different points, is this responsiveness.

Security Vulnerabilities of SDN

Even as SDN has a lot of good features for security, several parts of its architecture make it vulnerable:

- **Capacity Limitation in Ternary Content Addressable Memory (TCAM):**

The rules-based handling of packet flows is controlled by flow tables in TCAMs of SDN switches. One of the main issues of interest is the limited capacity of TCAM when a network is overloaded with requests as a result of distributed denial of service (DDoS) attacks. Take a look at the Pica8 3290 switch. It has a maximum rule size of 2000 which is clearly not enough to withstand high intensity DDoS attacks [7].

- **Failed Steps and Their Consequences:**

Centralisation is the company's strength and its weakness.Centralisation is the backbone and the weakness of SDN controller. While centralisation of control has numerous benefits, it also creates an opportunity to attack easily, as just one weak point is enough [8]. The whole network might be put at risk if the controller is compromised. A multi-controller network can suffer from the problems of synchronisation and consistency, which can lead to cascade failures, where the occurrence of a fault in one controller can have a detrimental effect on the other controllers.

- **Separating the Data Plane from the Control Plane:**

Because control logic and the network's actual hardware are separated in software-defined networking (SDN), security is a problem, even if it enhances efficiency and flexibility. It is common practice to use protocols like OpenFlow to enable communication between the forwarding and control levels. Attacks like man-in-the-middle and saturation attempt to overrun networks and corrupt data in transit, respectively, in an effort to interrupt this communication [9].

- **Decision-Making Reliance on the Controller:**

Designed with simplicity and reliance on the controller for operational instructions, SDN switches prioritize ease of use. Because of this, with very heavy traffic or an unresponsive controller, the network performance as a whole could take a major hit [10]. The inability of the switches to independently make decisions is a big problem in scenarios with heavy demand.

Finally, SDN's architecture allows centralized control and rapid configuration changes, which substantially enhance security. Nevertheless, it does bring about challenges that must be handled with caution. To safeguard the network from potential dangers like distributed denial of service (DDoS) attacks and over-reliance on one controller, robust security policies and backup plans are essential.

DDoS Attacks in SDN

The goal of a distributed denial of service attack is to interrupt service availability by sending a deluge of requests to the system from several compromised devices at once [11]. The fundamental objective of these attacks is to reduce system resources like CPU power, network bandwidth, and server storage to an unusable state for authorized users. Distributed denial of service attacks are becoming increasingly common as more and more business services go online. These attacks may cause substantial financial harm. There are a lot of potential motivations for such attacks, including financial extortion and competitive sabotage. It is critical to study the various distributed denial of service (DDoS) assaults that can affect SDN networks down the road. It is crucial to be aware of the vulnerabilities at each level of SDN infrastructure because, as shown in Figure 2, distributed denial of service attacks may affect any level of SDN.

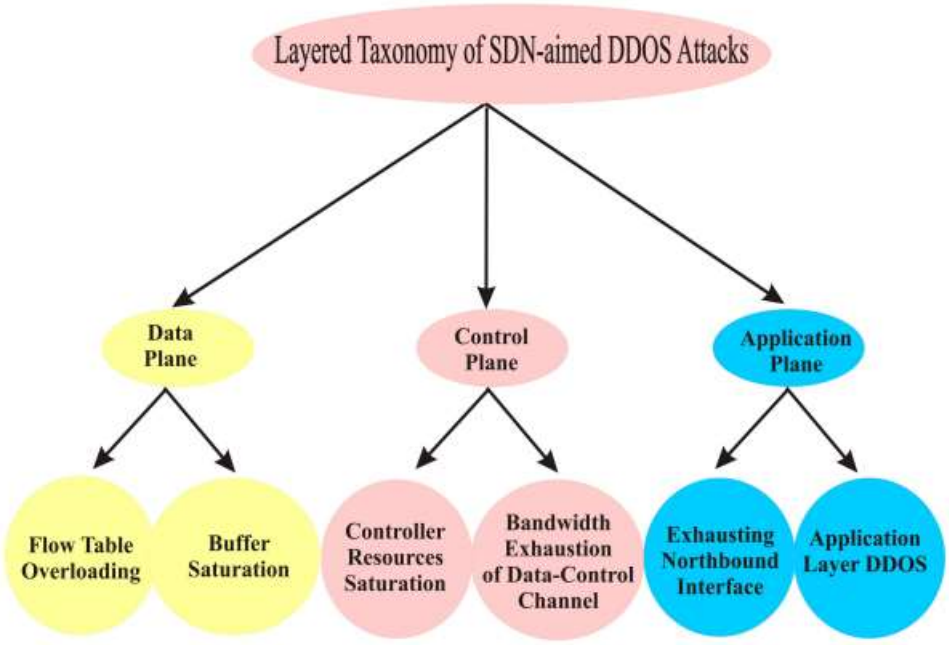


Figure 2: Layered taxonomy of DDoS attacks targeting SDN

DDOS ATTACK ON DATA PLANE

Distributed denial of service attacks aim to disrupt service availability by simultaneously flooding the system with requests from several infected devices [12]. At its core, this kind of attack seeks to render authorized users unable to use system resources such as CPU power, network bandwidth, and server storage. With more and more commercial services moving online, DDoS attacks are on the rise. Significant monetary damage could result from these assaults. Financial extortion and competitive sabotage are two of many possible reasons for such assaults. Research on the many distributed denial of service (DDoS) attacks that can impact SDN networks in the future is essential. Given that distributed denial of service attacks may impact any level of SDN (as seen in Figure 2), it is critical to be knowledgeable of the vulnerabilities at each level of SDN infrastructure.

SDN (SOFTWARE NETWORKS DEFINED)

The longest amount of time it will take to get flow data from a database. On timeout, the item is torn apart and on idle timeout if there are no corresponding packets, then the record is torn apart. Perhaps limit the size of flowtable, and utilize their help to add the rules.

Each entry consists of some criteria and actions associated with those criteria it will carry out if any of the criteria are met with the packet. If the destination port is found and matched, the switch will switch, drop or pass packets to a port. In the event of a error, the switch will switch between the two. It can be deleted or passed on to an SDN controller via OpenFlow.

In SDN packets are sent to SDN controller and controller decides on the packets based on the information contained in the flow packet. It is the controller's decision to use, drop or forward the packet, based on the rules that are given in the network. The controller can also automatically detect which packets to switch by comparing similar packets, and add entries into the controller flow table. However until now no one was able to achieve such scalability, efficiency and flexibility as Centralia. The old, decentralised, networks are deployed by installing individual routers/switches and individually configuring them. Its inflexible systems which lack responsiveness to users, centralisation and are traffic-centric.

The big difference between the two is the perceived changes in network lifespans brought about by software defined network (SDN), where data planes forward packets while control planes make the decisions, the data plane switches and routers are no longer in control of what to forward and instead are converted into "forwarding devices," and one central controller is the “brain” of the network. In software defined networking (SDN), the network administrators can now have a birds' eye view of the network, and effectively manage rules, resources and traffic.

One word of advice for SDN: Flexibility! The controller is able to communicate with other nodes in the network, and can be programmed remotely using OpenFlow. Since software-defined networking (SDN) is capable of automating network administration, providing policy-based control, and responding rapidly to application demands, it is well-suited for use in corporate networks, backbone networks, and data center applications. Because of its abstraction and scalability, it is able to handle different workloads and allow network growth. Additionally, it makes the design, development, and migration of Virtual Network Functions (VNFs) much simpler.

These capabilities enhance the resilience and agility of the network while also offering solutions to newly developing difficulties. SDN, on the other hand, has the potential to become a target for assaults, which is why centralised threat detection is very necessary. Further strengthening of security may be accomplished by the use of Zero Trust concepts, traffic

anomaly analysis, and granular access control. As a whole, software-defined networking (SDN) revolutionises conventional networking by virtue of virtualisation, centralised control, and programmability. It provides more flexibility while simultaneously preserving network performance and security without relying on hardware-level abstraction. [13]

SDN: COMMON ISSUES

The centralized control, unparalleled programmability and dynamic network administration offered by SDN customers is what attracts customers. These properties are usually supposed to be signs of serious issues with the type of concepts being used to make these properties and that can affect whether students internalize the concepts, and whether they can make use of them. It might be “growing pains” or just a case of getting ready to make changes to the SDN paradigm due to technical, operational and security issues. These issues of performance and reliability in the control plane should be suitably characterised, aiming to further develop the technology and offer good solutions to the market [14].

Clinging to software defined networking (SDN) backbone's side, there is at least one major single point of failure: The logically centralized controllers. In the case where this “brain” component is not operational then nothing can be done from the data plane and it will not know what to do in case someone wants to direct some network traffic there. This decreases the scalability and reliability of the system. The solution to this is a combination of the use of dispersed network controllers. This solution is more complex as each controller instance needs to be synced and communicate through complex consensus protocols. The dependency can result in high latency, network resource consumption and increased complexity for high availability. Hence, the network architect should thoroughly analyse the distributed architecture according to the requirement of the network, its expected performance and limitations of the resources before deciding on a deployment technique.

TYPES OF DENIAL-OF-SERVICE ATTACKS

The bandwidth and resources of servers are overwhelmed by traditional volumetric attacks. A common target for these types of attacks in SDN is the central controller or the data plane switches. The primary target of the first kind of attack is the controller. The processor of the controller has been overloaded with millions of unknown packet-in requests. It can't handle legitimate control actions. The control plane becomes unresponsive. Attacks on the data switches included in the confined memory constitute the second kind. Already at capacity, the

attack traffic is putting a strain on the flow tables. The controller's visibility and management capabilities are severely limited due to improper communication with switches caused by congested connections. There is also a more involved volumetric method that takes use of the interaction between the data and control planes. An attacker in the SDN sets up a flood of slow TCP connections with a specific server. Every time a new IP address sends a TCP SYN packet, the switch does not apply any flow rules [15].

By continually initiating new, slow connections from a multitude of IPs, the attacker overwhelms the switch's flow table with "incomplete" entries. Once the database is filled, the switch cannot install rules for genuine traffic and must query it for every incoming packet, which overwhelms the controller. Protocol attacks are more sophisticated and aim at stateful system resources like memory, connection tables, or the central processing unit (CPU). The controller is also in charge of protocols and services that might be susceptible. Most controllers come with an inbuilt ARP handler. A deluge of ARP requests for seemingly random IPs will cause the controller's CPU to become overburdened. It is possible to transmit LLDP-like packets or modify the ports of the switches to cause the controller's topology discovery function to continually rebuild the network map, which consumes CPU cycles.

Vulnerabilities in switch migrations and poisoning flow tables are two further examples of such attacks. On the application plane of the SDN, denial-of-service attacks happen. The SDN applications are particularly at risk. There is a chance that programs will influence the intrusion detection systems, firewall, router, and load balancer. The Northbound API, the route via which SDN applications (Network Apps) communicate with the SDN controller, is another target of threat actors. Using this API, these apps can receive statistics, request network services, and make policy decisions.

CLASSIFICATION AND WORKING MECHANISMS OF LINK-FLOODING ATTACKS

Coremelt Attack

The Coremelt attack is a distributed link-flooding technique in which compromised devices generate traffic flows between themselves to congest selected network links. Unlike conventional DDoS attacks, the traffic is not necessarily directed towards the victim. Instead, the attacker selects pairs of compromised hosts whose communication paths pass through the targeted network links.

The primary objective is to maximise congestion on critical links by coordinating traffic between multiple compromised devices. Since the communication occurs between distributed hosts, the victim's server may not receive the attack traffic directly. Nevertheless, legitimate traffic passing through the congested links experiences degradation or complete disruption.

The Coremelt attack is particularly challenging because individual traffic flows can appear legitimate. Network monitoring systems that primarily examine destination-oriented traffic may fail to identify the attack because the victim is not necessarily the endpoint of the malicious communication.

The Coremelt attack and examined its ability to exhaust the bandwidth of core network links through coordinated traffic between compromised machines. Subsequent work has investigated modelling and mitigation mechanisms to understand the impact of coordinated flows on network congestion.

Working mechanism

The Coremelt attack generally involves the following stages:

1. The attacker compromises multiple devices distributed across different network locations.
2. The attacker obtains information about the network topology and identifies strategically important communication links.
3. Pairs of compromised devices are selected according to whether their communication paths traverse the targeted links.
4. The compromised devices generate coordinated traffic flows between one another.
5. The combined traffic consumes the available bandwidth of selected links.
6. Legitimate communication passing through the congested links experiences packet loss, delay, and service disruption.

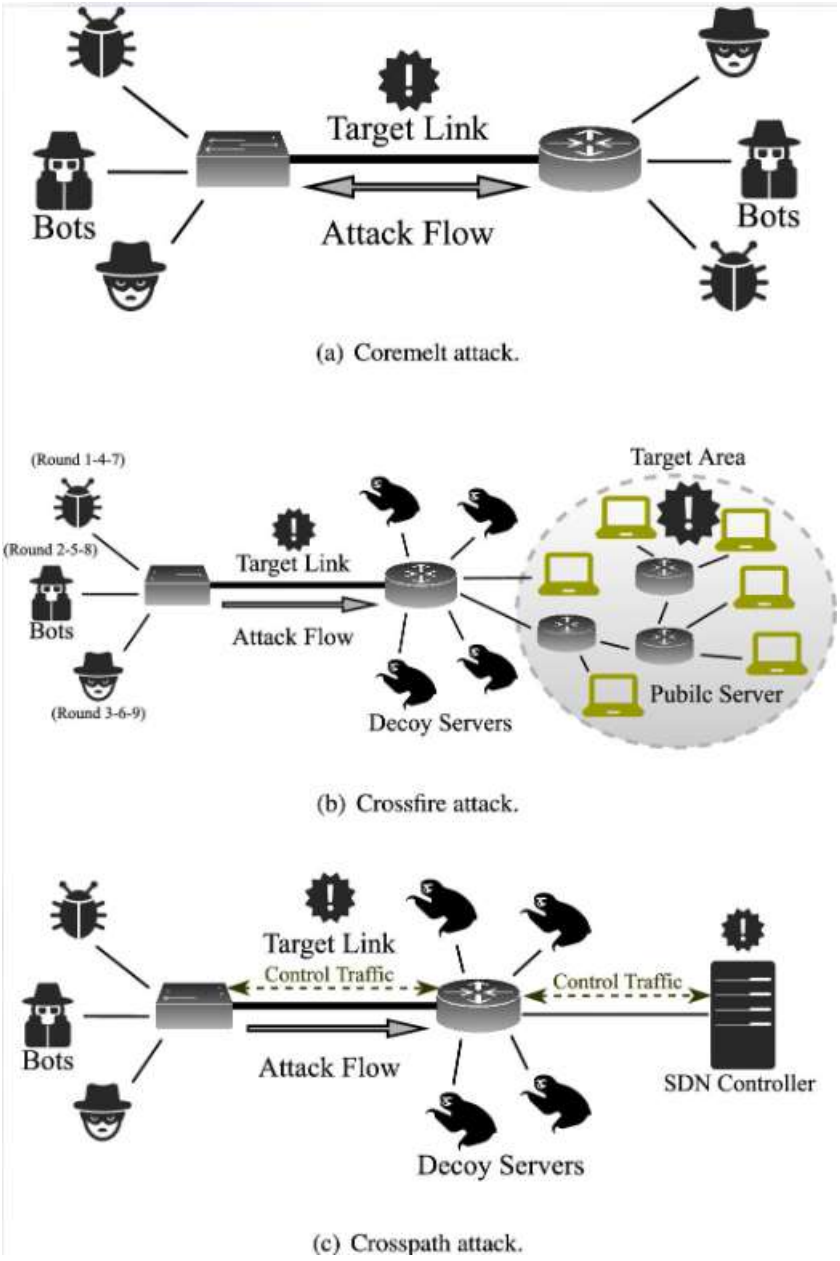


Figure 3: Conceptual representation of the Coremelt attack.

CONCLUSION

Despite its flexibility in network management and centralized control, Software-Defined Networking (SDN) is still susceptible to attacks such as Coremelt, Crossfire and Cross-Path attacks. These threats take advantage of network connections, communication paths, and dependencies on control channels to interrupt the network or prevent the availability of services. In this work, a multi-strategy defence approach has been analysed that combines traffic monitoring, anomaly detection, attack classification, network topology analysis,

adaptive routing, selective traffic filtering and protection of control channel. The proposed approach emphasizes the need to use multiple detection and mitigation approaches instead of using one security approach. Dynamic TMC traffic management and alternate path selection may be applied to minimize congestion and legitimate communication. Moreover, the evaluation of performance with the detection accuracy, precision, recall, F1-score, latency, throughput, packet loss, and controller overhead can be used to evaluate the effectiveness of defence. There is a need for future research in the fields of machine learning for detection, large-scale experimental validation, and efficient mitigation mechanisms in complex SDN environments.

References

1. Agborubere, B., & Sanchez-Velazquez, E. (2017). OpenFlow communications and TLS security in software-defined networks. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 560–566). IEEE.

2. Mahdy, M. (2018). *SDN southbound threats*. Global Information Assurance Certification, SANS Institute.

3. Thangasamy, P. (2017). Early detection of DDoS attacks in a multi-controller based SDN. *International Journal of Engineering Science and Computing, 7*, 13422–13429.

4. Viet, A. N., Van, L. P., Minh, H.-A. N., Xuan, H. D., Ngoc, N. P., & Huu, T. N. (2017). Mitigating HTTP GET flooding attacks in SDN using NetFPGA-based OpenFlow switch. In *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* (pp. 660–663). IEEE.

5. Shang, G., Zhe, P., Bin, X., Aiqun, H., & Kui, R. (2017). FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*. IEEE.

6. Dong, S., & Sarem, M. (2019). DDoS attack detection method based on improved KNN with the degree of DDoS attack in software defined networks. *IEEE Access*.

7. Krishnan, P., Duttagupta, S., & Achuthan, K. (2019). Varman: Multi-plane security framework for software defined networks. *Computer Communications, 148*, 215–239.

8. Musumeci, F., Ionata, V., Paolucci, F., Cugini, F., & Tornatore, M. (2020). Machine-learning-assisted DDoS attack detection with P4 language. In *Proceedings of the 2020 IEEE International Conference on Communications (ICC)* (pp. 1–6). IEEE.

9. Studer, A. and Perrig, A. (2009). The Coremelt attack. In: Proceedings of the European Symposium on Research in Computer Security (ESORICS 2009), Lecture Notes in Computer Science, vol. 5789, pp. 37–52. Springer.

10. Kang, M.S., Lee, S.B. and Gligor, V.D. (2013). The Crossfire attack. Proceedings of the IEEE Symposium on Security and Privacy, pp. 127–141.

11. Cao, J., Li, Q., Xie, R., Sun, K., Gu, G., Xu, M. and Yang, Y. (2019). The CrossPath attack: Disrupting the SDN control channel via shared links. Proceedings of the 28th USENIX Security Symposium, pp. 19–36.

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

13. Feamster, N.G., Rexford, J. and Zegura, E. (2014). The road to SDN: An intellectual history of programmable networks. ACM SIGCOMM Computer Communication Review, 44(2), pp. 87–98.

14. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. and Turner, J. (2008). OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38(2), pp. 69–74.

15. Kreutz, D., Ramos, F.M.V., Veríssimo, P., Rothenberg, C.E., Azodolmolky, S. and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), pp. 14–76.