Revisiting SDN Resilience in Cloud and Enterprise Environments

Sana Habib
Arizona State University
Tempe, Arizona, USA
Washington and Lee University
Lexington, Virginia, USA
shabib3@asu.edu

Jedidiah R. Crandall
Arizona State University
Tempe, Arizona, USA
Breakpointing Bad
Tempe, Arizona, USA
jedimaestro@asu.edu

Adam Doupé
Arizona State University
Tempe, Arizona, USA
doupe@asu.edu

Abstract

As SDN becomes the backbone of today's cloud and enterprise infrastructure, its security evaluation remains scattered and inconsistent. A persistent gap exists in current frameworks regarding transparently and specifically assessing the impact of SDN attacks and the effectiveness of corresponding defenses. General-purpose tools like CVSS overlook critical SDN characteristics—scalability, vendor independence, resource constraints, and operational visibility—essential for measuring impact. Platform-specific systems (e.g., ONOS, OpenDaylight) suffer from inconsistent criteria, limited cross-vendor applicability, and opaque scoring processes. Moreover, many impactful SDN attacks remain untracked in CVE databases due to disclosure barriers and the opaque nature of the SDN ecosystem. To address these gaps, we present Odin, an open-source, lightweight framework for evaluating SDN security in cloud and enterprise environments. Odin scores attack impact and defense effectiveness across three dimensions: technical severity, resource feasibility, and operational visibility. Applied to 20 real-world attack–defense scenarios—including underreported cases—Odin offers more contextualized prioritization and actionable insights than CVSS. Our results show that Odin scores align with or exceed CVSS ratings for all evaluated attacks, while also capturing defenses' effectiveness and operational trade-offs. By providing transparency, tunability, and SDN-specific relevance, Odin helps researchers and practitioners assess risk and improve resilience across varied deployment environments.

CCS Concepts

• **Security and privacy → Network security**.

Keywords

Security Assessment Model; SDN Attack Impact; SDN Defense Effectiveness; Assessment Metrics.

ACM Reference Format:
Sana Habib, Jedidiah R. Crandall, and Adam Doupé. 2025. Revisiting SDN Resilience in Cloud and Enterprise Environments. In *Proceedings of the 2025 Cloud Computing Security Workshop (CCSW '25), October 13–17, 2025, Taipei, Taiwan.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3733812.3765534

CCSW '25, Taipei, Taiwan

ACM ISBN 979-8-4007-1901-1/25/10
https://doi.org/10.1145/3733812.3765534

1 Introduction

Software-Defined Networking (SDN) has changed how networks are designed and managed by providing centralized control and scalable automation. Because of this, SDN has become a key technology powering modern infrastructure, with widespread use across cloud services, enterprise networks, data centers, and telecom systems. With a projected global market exceeding 60 billion by 2028 [31], SDN now underpins the connectivity demands of large-scale, mission-critical environments. Architectural standards such as RFC 7426 [52] and RFC 8342 [43] further underscore its growing maturity and standardization. This rapid evolution has also expanded the SDN attack surface. Threat actors increasingly exploit SDN-specific components — controllers, APIs, data stores, and programmable logic—yet mainstream security scoring frameworks fail to capture these vulnerabilities' impact accurately. General-purpose systems like CVSS [91], EPSS [17], and SSVC [8] overlook key features—such as scalability and vendor independence—that are essential for accurately assessing impact in SDN environments. Meanwhile, controller-specific efforts (e.g., OpenDaylight's internal ratings [21]) remain opaque, inconsistent, and non-portable across SDN platforms.

To address these gaps, we introduce Odin, an open-source framework for evaluating SDN attack impact and defense effectiveness in cloud and enterprise environments . Odin defines a structured, SDN-specific scoring model incorporating technical severity, resource feasibility, and operational visibility. Unlike existing approaches, Odin supports the evaluation of both attacks and defenses, integrates temporal factors, and remains vendor-neutral and extensible. Grounded in an analysis of 20 real-world SDN attack and defense case studies—including scenarios overlooked in CVE databases—Odin highlights risk dimensions that conventional tools miss. The framework is a lightweight, modular web app to enable practical adoption by cloud operators, security analysts, researchers, and SDN vendors. Our contributions are as follows:

(1) We present Odin, the first open-source, SDN-specific framework for structured, comparative evaluation of attack impact and defense effectiveness across cloud and enterprise settings.
(2) We apply Odin to 20 real-world case studies, demonstrating how it complements existing systems like CVSS while offering richer, SDN-aware prioritization insights.

By addressing the limitations of traditional scoring systems, Odin provides a transparent, extensible, and SDN-specific approach

to security evaluation—supporting more informed, cloud-aware decision-making in an increasingly programmable network ecosystem.

2 Background

This section provides an overview of SDN architecture, typical evaluation environments, and representative real-world attacks and defenses—laying the foundation for understanding the motivation and scope of our framework in cloud and enterprise SDN deployments.

2.1 The SDN Stack

SDN's layered architecture—shown in Figure 1—underpins many cloud-scale and enterprise networks today.

2.1.1 Application Plane. At the right, SDN applications—such as load balancers and firewalls—interact with the SDN controller via northbound APIs to define network policies and configurations. Currently, no universal standard governs these northbound interfaces, leading to diversity in implementations.

2.1.2 Control Plane (CP). The control plane acts as the centralized "brain" of the network, managing flow rules, topology, and policy enforcement. Widely used open-source controllers include OpenDaylight [30], ONOS [28] (Java-based), Ryu [14], POX [34] (Python-based), and NOX [59] (C/C++-based). The controller maintains two key datastores [43]: the configuration datastore holding intended network state, and the operational datastore reflecting the live network state. Core services within controllers handle rule dissemination, event tracking, and policy enforcement—functions that become more complex and critical at enterprise or multi-tenant cloud scales.

2.1.3 Network/Data Plane (DP). The data plane comprises physical or virtual switches, routers, links, and hosts. The southbound interface—most commonly OpenFlow [82]—connects the control plane to the data plane, enabling controller-driven forwarding and packet processing. Test environments range from logical emulations (e.g., Mininet [23]) to physical testbeds with commercial switches (IBM RackSwitch [65], Juniper MX [87], Pica8) and real servers, enabling scalable experimentation across both cloud and enterprise network conditions.

We conceptualize SDN as comprising five key components: (i) SDN applications in the application plane, (ii) the SDN controller in the control plane, (iii) switches and routers in the data plane, (iv) the northbound communication channel, and (v) the southbound communication channel.

2.2 SDN Evaluation Environments

Evaluating SDN security relies on three broad environments, discussed below.

2.2.1 Theoretical Environment. Formal models—including logic verification [68, 69], complexity analysis [70], and probabilistic models (e.g., Monte Carlo [117], Markov chains [64])—are used to assess SDN behavior in idealized settings. These models help reason about systemic properties and are especially useful for verifying security at scale in enterprise or cloud deployments.

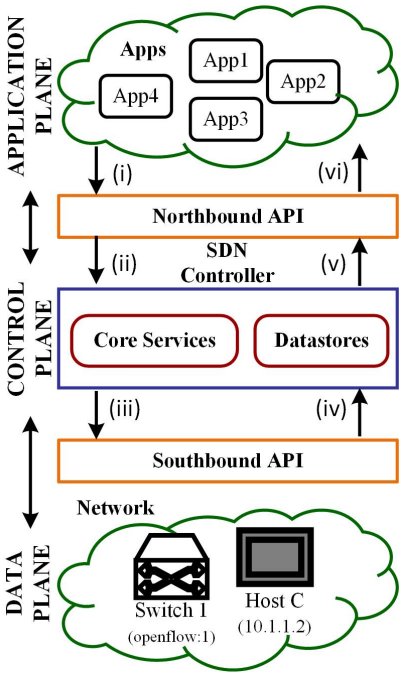

Figure 1: SDN architecture: Apps configure the network via the controller (northbound APIs); the controller enforces changes through southbound interfaces and maintains state in data stores.

2.2.2 Simulation Environment. For SDN simulation experiments, Mininet [23] is the most widely used tool, with others including Netty [26], Nmap [27], and Netcat [24]. Automated scripts [54, 62], cross-traffic emulators [41], and publicly available networking datasets [6, 10] can be used to generate simulation traffic. Virtual labs like Thoth Lab [32], GENI [19, 55], and OpenStack [92] are used to create simulated networks.

2.2.3 Real-World Experimentation. ESnet [15], Internet Topology Zoo [12], and .pcap traces [25] offer real traffic for evaluation in physical testbeds. Case studies span small labs [53, 90] to production-scale cloud networks [22, 51], demonstrating SDN's exposure in operational environments.

2.3 SDN Attacks-SDN Defenses

We reviewed 42 SDN attack and 45 defense strategies published between 2018–2024 (listed in Table 10, Appendix 8), covering all architectural layers. These span data plane reconnaissance [48], control-plane DoS [96], and northbound privilege escalation, as well as cloud-integrated attacks and defenses targeting scalable and dynamic controller operations. Defenses include formal verification [99], architectural techniques like Moving Target Defense [94], and platform-specific patches [66]. This body of work highlights recurring challenges—multi-layer vulnerabilities, slow defense deployment, and inconsistent impact reporting—which are particularly pressing in cloud and enterprise networks.

Our framework builds on these insights by enabling systematic, side-by-side comparisons of attack severity and defense robustness in realistic deployment contexts. It is designed to bridge the gap

Table 1: Assessment of Key Frameworks and Initiatives for SDN Security Assessment.

(Notations: D → Partially Covered, ✗ → Not Covered, N/A → Not Available.)

Vulnerability-Resilience Management Framework	SDN Specific	Assessment (Attacks)	Assessment (Defenses)	Primary Assessment	Resource Assessment	Visibility Assessment
Common Vulnerability Scoring System (CVSS) [91]	No	Yes	No	D	✗	✗
Exploit Prediction Scoring System (EPSS) [17]	No	Yes	No	D	✗	✗
Stakeholder-Specific Vulnerability Categorization (SSVC) [8]	No	Yes	No	D	✗	✗
Vulnerability Priority Rating (VPR) [33]	No	Yes	No	D	✗	✗
FloodLight Assessment System [18]	Yes	N/A	N/A	N/A	N/A	N/A
ODL Assessment System [21, 30]	Yes	Yes	Yes	D	N/A	N/A
ONOS Assessment System [21, 28]	Yes	Yes	Yes	D	N/A	N/A
Ericson Cloud SDN Assessment System [13]	Yes	N/A	N/A	N/A	N/A	N/A
Huawei Agile Assessment System [35]	Yes	N/A	N/A	N/A	N/A	N/A
Juniper Networks SDN [87]	Yes	N/A	N/A	N/A	N/A	N/A
Cisco SDN [56]	Yes	N/A	N/A	N/A	N/A	N/A
IBM SDN [65]	Yes	N/A	N/A	N/A	N/A	N/A
DELTA [78]	Yes	No	No	N/A	N/A	N/A
Sphinx [53]	Yes	No	No	N/A	N/A	N/A
Open Networking Foundation (ONF) [29]	Partly	N/A	N/A	N/A	N/A	N/A
European Telecommunications Standards Institute (ETSI) [16]	Partly	N/A	N/A	N/A	N/A	N/A
International Telecommunication Union (ITU-T) [20]	Partly	N/A	N/A	N/A	N/A	N/A
Odin	Yes	Yes	Yes	✓	✓	✓

between academic evaluations and practical, cross-platform SDN risk assessment in modern network environments.

3 The Evaluation Gap in SDN Security

As SDN continues to redefine how networks are architected and managed—especially in cloud and enterprise environments—its security landscape grows increasingly complex. Yet, there remains no principled, SDN-specific framework for evaluating both the impact of attacks (e.g., scalability, visibility) and the effectiveness of defenses. Existing approaches—from generic scoring systems to vendor-specific tools and academic prototypes—remain fragmented, opaque, or fundamentally misaligned with the architectural realities of SDN. This absence of a unified evaluation standard undermines risk prioritization, slows mitigation planning, and hinders the ability to compare defenses across platforms and deployments. In this paper, we revisit these gaps to motivate the design of Odin.

3.1 Generic Rating Systems

CVSS [91] remains the most widely adopted vulnerability scoring system, but it was never designed to assess the impact of attacks in programmable, modular environments like Software-Defined Networking (SDN). It lacks support for critical SDN-specific properties such as vendor neutrality and architectural scalability—all essential for accurately measuring attack impact. More importantly, CVSS is attack-focused and entirely omits defense evaluation, offering no mechanism to assess how effectively an attack can be mitigated or at what operational cost.

Other general-purpose frameworks—such as EPSS [17], SSVC [8], and VPR [33]—inherit these same limitations. While they offer useful heuristics for exploitability and prioritization, they fall short in capturing the dynamic, programmable, and defense-aware nature of SDN environments. As a result, these tools fail to support meaningful security assessment in cloud-scale SDN deployments.

3.2 SDN Platform-Specific Tools

Several SDN controller platforms have introduced their own vulnerability and resilience rating features. OpenDaylight (ODL) [30] maintains a proprietary scoring system, but it is closed-source and ecosystem-locked [21]. FloodLight [18] and ONOS [28] either provide no public evaluation data or lack structured scoring models.

Commercial solutions from Ericsson [13], Huawei [35], Cisco [56], Juniper [87], and IBM [65] similarly depend on internal, non-transparent methods. These fragmented efforts make cross-vendor evaluation impossible, stifle collaboration, and prevent reproducibility in security research.

3.3 Academic Security Tools for SDN

Academic work has produced essential tools for SDN threat detection—e.g., DELTA [78] and Sphinx [53]—but these focus on runtime analysis rather than structured evaluation. They do not assign severity scores, assess defense robustness, or support comparative benchmarking. Ivkić et al.'s security framework [67] makes strides toward SDN-specific modeling. However, it still lacks a quantitative, metric-driven evaluation of attack and defense pairs, especially regarding operational cost, technical feasibility, and real-world deployment tradeoffs.

3.4 Open Standards and Community Initiatives

Standardization bodies such as ONF [29], ETSI [16], and ITU-T [20] have laid foundational principles for SDN security but stop short of defining how to measure it. They identify security requirements

Table 2: The Formal Details of the Odin Framework.

Primary (Common) Assessment Metrics: Used to evaluate the severity-robustness of an SDN attack-defense (Tables 3 and 4).
Temporal Primary (TP) Assessment Metrics: Accounts for temporal changes in the values of primary assessment metrics (Tables 12 and 13, Appendix 10.1).

$$\text{Primary Attack-Impact Score} = \sum_{i=1}^{5} P_i + \sum_{j=6}^{9} PA_j \quad (1)$$

where P_i ($i = 1, 2, 3, 4, 5$) denotes the primary assessment metrics, and PA_j ($j = 6, 7, 8, 9$) denotes the primary assessment metrics specific to SDN attacks.

$$\text{Primary Defense-Effectiveness Score} = \sum_{i=1}^{5} P_i + \sum_{k=6}^{7} PD_k \quad (2)$$

where P_i ($i = 1, 2, 3, 4, 5$) denotes the primary assessment metrics, and PD_k ($k = 6, 7$) denotes the primary assessment metrics specific to SDN defenses.

Primary Attack-Impact Score Range: Low (0.0 - 1.0), Medium (> 1.0 - 3.25), High (> 3.25 - 6.5), Critical (> 6.5 - 9.0);
Primary Attack-Impact Score Interpretation: Least Severe (0.0) - Most Severe (9.0);
Primary Defense-Effectiveness Score Range: Low (0.0 - 0.75), Medium (> 0.75 - 2.5), High (> 2.5 - 5.0), Critical (> 5.0 - 7.0);
Primary Defense-Effectiveness Score Interpretation: Least Robust (0.0) - Most Robust (7.0);
Rating Levels for Primary (P) Assessment Metrics: PL0, PL1, PL2, PL3;
Rating Levels for Temporal Primary (TP) Assessment Metrics: TPL0, TPL1, TPL2, TPL3, TPL4;

Resource (R) Assessment Metrics: Used to generate a resource score to indicate resource requirement (Table 5).
Temporal Resource (TR) Assessment Metrics: Accounts for temporal changes in the values of resource assessment metrics (Table 14, Appendix 10.2).

$$\text{Resource Score} = \sum_{m=1}^{4} R_m \quad (3)$$

where R_m ($m = 1, 2, 3, 4$) denotes the resource assessment metrics.

Resource Score Range: Low (0.0 - 0.5), Medium (> 0.5 - 1.5), High (> 1.5 - 3.0), Critical (> 3.0 - 4.0);
Resource Score Interpretation: Least Expensive (0.0) - Most Expensive (4.0);
Rating Levels for Resource (R) Assessment Metrics: RL0, RL1, RL2, RL3;
Rating Levels for Temporal Resource (TR) Assessment Metrics: TRL0, TRL1, TRL2, TRL3, TRL4;

Visibility (V) Assessment Metrics: Used to generate a visibility score to indicate prominence of an SDN attack-defense (Table 6).
Temporal Visibility (TV) Assessment Metrics: Accounts for temporal changes in the values of visibility assessment metrics (Table 15, Appendix 10.3).

$$\text{Visibility Score} = \sum_{n=1}^{2} V_n \quad (4)$$

where V_n ($n = 1, 2$) denotes the visibility assessment metrics.

Visibility Score Range: Low (0.0 - 0.25), Medium (> 0.25 - 0.75), High (> 0.75 - 1.5), Critical (> 1.5 - 2.0);
Visibility Score Interpretation: Least Prominent (0.0) - Most Prominent (2.0);
Rating Levels for Visibility (V) Assessment Metrics: VL0, VL1, VL2, VL3;
Rating Levels for Temporal Visibility (TV) Assessment Metrics: TVL0, TVL1, TVL2, TVL3, TVL4;

but do not provide operational frameworks for evaluating threat impact or mitigation strategies in evolving SDN infrastructures.

3.5 Towards SDN-Aware Security Assessment

To close these gaps, the SDN community requires a rigorous framework that integrates impact assessment, defense effectiveness, and operational cost. We introduce Odin, a domain-specific scoring system that evaluates both SDN attacks and defenses in SDN deployments across cloud and enterprise environments. Odin evaluates security events across four core dimensions: *impact* (disruption potential of attacks), *robustness* (effectiveness of defenses), *resource requirements* (cost, time, and expertise), and *visibility*[1] (strategic or operational prominence). Unlike CVSS, Odin captures programmable modular networks' architectural and operational nuances. Unlike vendor-specific tools, it is open-source, vendor-neutral, and built for reproducibility. And unlike prior academic approaches, it supports structured, side-by-side comparisons to inform real-world security decisions.

This paper presents the Odin scoring methodology, its web-based implementation, and an evaluation of 20 real-world SDN attack and defense scenarios. This work aims to advance rigorous, interoperable, and reproducible tooling for SDN security assessment—tailored to the unique demands of modern cloud and enterprise networks.

4 System Design

Practical SDN security evaluation requires more than just identifying threats—it demands a structured, repeatable method for assessing the impact of attacks, the robustness of defenses, the feasibility of execution (i.e., resource requirements), and the visibility or operational relevance of each scenario.

4.1 Design Philosophy

The Odin framework addresses this need through a metric-based scoring model structured around three key dimensions: technical

[1]This term is discussed in greater detail in Section 4.5.

impact or robustness, operational feasibility, and strategic significance. These dimensions are grounded in a review of over 40 SDN security papers (listed in Table 10, Appendix 8) and a critical analysis of widely adopted evaluation frameworks (summarized in Table 1). While tools like CVSS [91] and STRIDE [72] offer general-purpose guidance, they fall short of capturing the modular, programmable, and evolving nature of SDN deployments—especially in cloud-integrated environments. Our design philosophy is rooted in real-world applicability: we aim to support scoring that reflects technical correctness and coverage, deployment feasibility, and broader operational impact. To this end, Odin defines the following metric categories:

(1) **Primary Assessment Metrics** capture core technical properties—such as correctness, scalability, vendor independence, reproducibility, and complexity—that shape the severity, generalizability, and mitigability of SDN attacks and defenses. This category comprises eleven metrics: five common to both attacks and defenses, four specific to attacks, and two exclusive to defenses.
(2) **Resource Assessment Metrics** evaluate the practical feasibility of executing or deploying an attack or defense, covering dimensions such as operation time, equipment cost, workforce size, and expertise level. These four metrics reflect the operational burden associated with real-world exploitation or mitigation.
(3) **Visibility Assessment Metrics** assess the symbolic or strategic prominence of a given attack or defense, incorporating factors such as public exposure, research venue, and broader societal or policy impact. This category includes two metrics: visibility via venue and social impact.

To capture how feasibility and credibility evolve over time, *each metric is paired with a corresponding temporal variant*, allowing dynamic reassessment as new information emerges (e.g., defenses become outdated, or attacks become easier to replicate). The only exception is the *Venue* subscore under Visibility, which is fixed at the time of publication and does not change temporally.

This clear partitioning allows consistent, transparent, and SDN-aware evaluation across a wide variety of attack and defense scenarios. Practitioners can use these metrics to prioritize mitigations based on both technical risk and operational feasibility. Researchers and vendors can use the same scoring framework to benchmark innovations, track performance over time, and compare results against a unified, reproducible metric space.

4.2 Algorithm 1 Rationale

To enable fair, interpretable, and repeatable evaluation across diverse SDN attack and defense scenarios, the Odin framework employs a normalized scoring algorithm that maps raw metric values into structured severity or effectiveness bands.

Metrics are organized into four main categories—*primary*, *resource*, *visibility*, and their *temporal* variants—and rated using discrete ordinal levels. The base scale has four levels: `Level 0 (0.0)`, `Level 1 (0.25)`, `Level 2 (0.5)`, and `Level 3 (1.0)`. Temporal metrics refine this to five levels by adjusting `Level 3` to 0.75 and adding `Level 4 (1.0)`, enabling finer tracking of changes over time.

Algorithm 1 Score Range Computation for Odin Framework

(A step-by-step walkthrough of the Visibility score range computation is provided in Appendix 9.)

Input:
Define metric counts:
  γ = number of primary attack-impact metrics
  ζ = number of primary defense-effectiveness metrics
  δ = number of resource metrics
  λ = number of visibility metrics

Define ordered list *rating_levels* = [0.0, 0.25, 0.5, 1.0]
Define ordered list *temporal_levels* = [0.0, 0.25, 0.5, 0.75, 1.0]

Define *categories* = {primary (common), primary (attack-impact), primary (defense-effectiveness), resource, visibility}
Define *temporal_categories* = {temporal primary (common), temporal primary (attack-impact), temporal (defense-effectiveness), temporal resource, temporal visibility}
for *cat* in *categories* **do**
  Determine metric count α for category:
  if cat == primary attack-impact **then**
    $\alpha = \gamma$
  else if cat == primary defense-effectiveness **then**
    $\alpha = \zeta$
  else if cat == resource **then**
    $\alpha = \delta$
  else
    $\alpha = \lambda$
  end if
  Initialize index $i = 1$ and $lower\_limit = 0.0$
  for *severity* in {low, medium, high, critical} **do**
    if $i \leq 3$ **then**
      $\beta = \lfloor \alpha/2 \rfloor$
      Set $score[severity].lower = lower\_limit$
      Set $score[severity].upper =$
        $rating\_levels[i-1] \cdot (\alpha-\beta) + rating\_levels[i] \cdot \beta$
      $lower\_limit = score[severity].upper$
      $i \leftarrow i + 1$
    else
      $score[severity].lower = lower\_limit$
      $score[severity].upper = rating\_levels[i-1] \cdot \alpha$
    end if
  end for
end for
for *cat* in *temporal_categories* **do**
  Repeat same process using *temporal_levels* list
end for

To avoid arbitrary thresholds, the algorithm dynamically partitions each category's total score range into four bands—*Low*, *Medium*, *High*, and *Critical*. This ensures proportional scaling, preventing larger categories (e.g., attack impact) from overshadowing smaller ones (e.g., visibility). Band boundaries are computed through weighted interpolation of adjacent rating levels, using a running variable, `lower_limit`, to update boundaries iteratively. Temporal metrics use the same approach with finer granularity to capture evolving urgency, feasibility, or prominence—such as exploit disclosure or mitigation deployment—without changing

Table 3: Primary Assessment Metrics (P-series) in the Odin Framework for Evaluating SDN Security.

(*Notation:* CP → Control Plane, DP → Data Plane, (↓) → Reduced by, (|) → OR, (&) → AND, PLX → Rating Level X where X = 0, 1, 2, 3).

	Metric	Metric Description	Value Level	Score	Value Explanation
P_1.	Correctness	Measures degree of verified correctness of an SDN attack or defense.	PL0	0.0	Unknown or missing details.
			PL1	0.25	Verified only in theoretical models or simulation.
			PL2	0.5	Verified in both theoretical models and simulation.
			PL3	1.0	Verified in real SDN deployments.
P_2.	Scalability	Measures evaluation scale.	PL0	0.0	Unknown or untested scalability.
			PL1	0.25	Tested on small topologies (e.g., ≤ 20 nodes).
			PL2	0.5	Tested on moderate topologies (e.g., 10–50 nodes) with latency and flow table stress representative of enterprise-scale networks.
			PL3	1.0	Tested on large-scale SDN deployments (i.e., 50+ nodes, multi-domain or datacenter-scale) representative of cloud-scale operations.
P_3.	Vendor Independence	Measures generalizability across SDN platforms.	PL0	0.0	Verified for only one minor SDN platform or vendor.
			PL1	0.25	Verified for one major vendor's controller or switch.
			PL2	0.5	Verified across at least two vendor-specific SDN controller or switch stacks.
			PL3	1.0	Verified across three or more major controllers or switch combinations (e.g., ODL+OpenFlow+FloodLight).
P_4.	Reproducibility	Measures how easily the attack or defense can be reproduced.	PL0	0.0	No code or configuration access (e.g., proprietary, undocumented platform).
			PL1	0.25	Approximately 25% reproducible (partial scripts and code availability).
			PL2	0.5	Approximately 50% reproducible.
			PL3	1.0	Fully reproducible attack or defense stack (e.g., open-source scripts).
P_5.	Complexity	Measures complexity of integrating solution into real SDN control or data planes.	PL0	0.0	Impractical for production systems (e.g., requires controller redesign). Verified only for minor SDN vendors.
			PL1	0.25	High complexity (e.g., requires vendor-specific APIs). Verified for one major vendor.
			PL2	0.5	Moderate complexity—deployable with limited changes. Verified for two major vendors.
			PL3	1.0	Low complexity; verified for three or more major vendors.

intrinsic severity. Inspired by standards like CVSS, the ordinal levels and dynamic bands balance usability and analytical rigor. Appendix 9 provides a step-by-step walkthrough of the score range computation for the Visibility category.

The framework is explicitly designed for tunability and extensibility: rating levels, category weights, and metric definitions can be adjusted based on expert feedback, empirical data, or emerging SDN challenges. This flexibility supports ongoing recalibration, ensuring relevance amid new attack vectors and defenses. In summary, this scoring model provides a structured yet adaptable method for aggregating technical, resource, and visibility metrics, enabling meaningful comparison, prioritization, and strategic planning for SDN security.

4.3 Primary Assessment Metrics

The primary assessment metrics calculate an SDN attack and defense score using Equation 1 and Equation 2 in Table 2, reflecting attack impact and defense effectiveness, respectively. As mentioned earlier, we propose eleven primary metrics, five common to attack and defense, detailed in Table 3 and Table 4.

4.3.1 Correctness (P_1.). This metric assesses the degree to which the correctness of an SDN attack or defense has been validated. Evaluation methods may include theoretical analysis, simulation-based testing, or deployment in real-world SDN environments. Higher scores reflect stronger empirical or formal support for the behavior and effectiveness of the attack or defense mechanism.

4.3.2 Scalability (P_2.). This metric assesses the scalability of an SDN attack or defense based on the network size (small, medium, or large) used for testing. A small network has fewer than 20 nodes, a medium network contains 20–50 nodes, and a large network has more than 50 nodes, including multi-domain or datacenter-scale topologies representative of cloud-scale operations. Intuitively, an attack or defense tested on a large network has higher impact or effectiveness.

4.3.3 Vendor Independence (P_3.). This metric evaluates the vendor independence of an SDN attack or defense scheme. An attack or defense that impacts major SDN platforms (e.g., FloodLight [18], OpenDayLight [30], ONOS [28]) is considered more significant than one affecting less widely used platforms. Similarly, a defense scheme applicable across multiple SDN platforms offers excellent utility.

4.3.4 Reproducibility (P_4.). This parameter assesses the reproducibility of an SDN attack or defense scheme, considering factors such as the availability of source code, detailed white papers, or openly accessible methodologies. Schemes that provide sufficient implementation details enable independent validation and foster broader adoption by researchers and practitioners. In contrast, attacks or defenses that lack transparency or rely on proprietary resources are more difficult to reproduce, limiting their credibility and long-term impact.

4.3.5 Complexity (P_5.). This metric evaluates the practicality of executing an SDN attack or deploying a defense by assessing its complexity (such as ease of deployment). Attacks that can be launched

Table 4: Primary Attack-Impact (PA-series)-Defense-Effectiveness (PD-series.) Assessment Metrics in the Odin Framework.

(*Notation*: C → Confidentiality, I → Integrity, A → Availability, CP → Control Plane, DP → Data Plane, SDN Components → Section 2.1.)

	Metric	Metric Description	Value Level	Score	Value Explanation
PA_6.	Confidentiality Impact (C)	Measures confidentiality impact—CP/DP data exposure.	C: PL0	0.0	No or unknown confidentiality compromise.
			C: PL1	0.25	Compromise of one SDN component.
			C: PL2	0.5	Compromise of two SDN components.
			C: PL3	1.0	Widespread exposure across CP and DP of three or more components.
PA_7.	Integrity Impact (I)	Measures integrity impact—tampering in CP or DP.	I: PL0	0.0	No or unknown integrity violation.
			I: PL1	0.25	CP policy injection or DP rule override on one SDN component.
			I: PL2	0.5	Compromise of two components across CP/DP layers.
			I: PL3	1.0	Multi-component, multi-plane policy/rule corruption.
PA_8.	Availability Impact (A)	Measures availability disruption—CP and/or DP failure.	A: PL0	0.0	No or unknown availability loss.
			A: PL1	0.25	Localized CP crash or DP flow timeout affecting one SDN component.
			A: PL2	0.5	Partial CP failure or flow loss across two SDN components.
			A: PL3	1.0	Broad CP unavailability or data path blackout across three SDN components.
PA_9.	Severity Level	Severity score based on standardized metrics (CVSS or equivalent).	PL0	0.0	Low severity (e.g., CVSS 0.1–3.9).
			PL1	0.25	Medium severity (CVSS 4.0–6.9).
			PL2	0.5	High severity (CVSS 7.0–8.9).
			PL3	1.0	Critical severity (CVSS 9.0–10.0).
PD_6.	Performance Cost	Measures performance cost of migration on CP/DP.	PL0	0.0	Unknown or very high overhead: CP latency > s or DP throughput loss > 10%.
			PL1	0.25	High CP reaction delay (ms–s) or DP flow miss overhead (5–10%).
			PL2	0.5	Moderate latency (us–ms) or minor DP performance impact (1–5%).
			PL3	1.0	Low impact on both planes (CP latency < us, DP loss ≤ 1%).
PD_7.	Resilience to Disruption	Defense robustness across CP/DP disruption vectors.	PL0	0.0	No known mitigation, or easily bypassed workaround (either plane).
			PL1	0.25	Mild mitigation on one plane; attack remains partially effective.
			PL2	0.5	Moderate cross-plane hardening (e.g., controller patch + switch config).
			PL3	1.0	Strong mitigation across planes, including proactive detection and recovery.

with minimal technical expertise or limited resources are considered more severe, as they are accessible to a wider range of adversaries. Conversely, defenses that can be integrated with existing systems without extensive customization are regarded as more practical. High implementation barriers — such as requiring specialized hardware, deep protocol modifications, or significant performance trade-offs — reduce the feasibility of both attacks and defenses in real-world environments.

4.3.6 Confidentiality (C), Integrity (I), Availability (A) Impact. (PA_6., PA_7., PA_8.) As the names suggest, these three metrics assess the impact of an SDN attack on (i) Confidentiality (C), (ii) Integrity (I), and (iii) Availability (A) of SDN components, including the SDN controller, data plane (hosts, switches, and links), and communication interfaces (northbound and southbound). Each metric is assigned one of four values based on the number of compromised components, as detailed in Table 4.

4.3.7 Severity Level (PA_9.). This metric quantifies the severity of an SDN attack by referencing its corresponding CVSS score, providing a standardized baseline for impact comparison. The scoring scale and interpretation are detailed in row PA_9. of Table 4.

4.3.8 Performance Cost (PD_6.). This metric captures the performance overhead introduced by an SDN defense mechanism, considering increased latency (measured in seconds, milliseconds, microseconds, or nanoseconds) and the percentage reduction in network throughput. Higher penalties reflect greater performance trade-offs, which may affect deployment feasibility in latency-sensitive environments.

4.3.9 Resilience to Disruption (PD_7.). This metric evaluates the robustness of an SDN defense in mitigating not only the original attack but also more advanced or variant attacks within the same category. A higher score reflects greater adaptability and long-term effectiveness of the defense. This metric takes one of four predefined values, as detailed in row PD_7. of Table 4.

Temporal Primary Assessment Metrics. Temporal primary assessment metrics do not reflect changes in the intrinsic severity of a vulnerability but rather capture how the risk posed by an attack or defense scenario evolves over time. For example, once a fix or mitigation is introduced, security teams must re-evaluate the system's residual risk through CVSS, Odin, or other frameworks to inform prioritization and resource allocation. These temporal metrics enable dynamic risk assessment by capturing how changes—such as patch deployment, exploit evolution, or defense rollout—affect the real-world urgency and operational significance of a vulnerability or mitigation. Definitions are provided in rows TP_1.–TP_5. of Table 12 and in rows TPA_6., TPA_7., TPD_6., and TPD_7. of Table 13. Table 12 and Table 13 are included in Appendix 10.1 to conserve space.

4.4 Resource Assessment Metrics

Technical analysis alone is not enough to evaluate the impact of an SDN attack and the effectiveness of an SDN defense. It also demands an understanding of the practical resources involved. The resource assessment metrics—quantify the real-world effort required to launch attacks or deploy defenses (in terms of operation time, equipment cost, workforce required, and expertise). By capturing these operational costs, the framework supports a balanced view of

Table 5: Resource Assessment Metrics (R-series) in the Odin Framework for Evaluating Resource Burden of SDN Attacks and Defenses.

(*Notation*: RLX → Level X for R, where X = 0, 1, 2, 3, AT → Attack, DF → Defense).

	Metric	Metric Description	Value Level	Score	Value Explanation
R_1.	Operation Time	Time required to execute or mitigate attack/defense in SDN CP or DP.	AT: RL0	0.0	Unknown or missing details.
			AT: RL1	0.25	Execution time ranges from microseconds to under 1 second.
			AT: RL2	0.5	Execution time ranges from 1 second to under 1 hour.
			AT: RL3	1.0	Execution time exceeds 1 hour.
			DF: RL0	0.0	Unknown or missing details.
			DF: RL1	0.25	Mitigation time ranges from hours to under 1 week.
			DF: RL2	0.5	Mitigation time ranges from 1 week to under 1 year.
			DF: RL3	1.0	Mitigation time exceeds 1 year.
R_2.	Equipment Cost	Hardware/software cost for executing or mitigating SDN attacks/defenses.	RL0	0.0	Unknown or missing cost data.
			RL1	0.25	Cost between USD 100 and less than USD 1,000.
			RL2	0.5	Cost between USD 1,000 and less than USD 10,000.
			RL3	1.0	Cost USD 10,000 or more.
R_3.	Workforce Requirement	Number of skilled personnel needed for SDN CP/DP attack or defense.	AT: RL0	0.0	Unknown or missing data.
			AT: RL1	0.25	Five or fewer personnel.
			AT: RL2	0.5	Six to ten personnel.
			AT: RL3	1.0	More than ten personnel.
			DF: RL0	0.0	Unknown or missing data.
			DF: RL1	0.25	Twenty or fewer personnel.
			DF: RL2	0.5	Twenty-one to fifty personnel.
			DF: RL3	1.0	More than fifty personnel.
R_4.	Expertise	Skill level required to perform or mitigate SDN CP/DP attack or defense.	AT: RL0	0.0	Unknown or missing data.
			AT: RL1	0.25	Low skill (e.g., script kiddie level).
			AT: RL2	0.5	Moderate skill (e.g., experienced attacker).
			AT: RL3	1.0	High expertise (e.g., elite SDN security researcher).
			DF: RL0	0.0	Unknown or beginner-level defense implementation.
			DF: RL1	0.25	Intermediate-level developer.
			DF: RL2	0.5	Advanced-level developer.
			DF: RL3	1.0	Expert-level developer.

feasibility and sustainability, which enables security practitioners to make informed, resource-aware decisions—prioritizing vulnerabilities not only by impact but also by the practical demands of mitigation or exploitation. The resource score is formally represented by Equation 3 in Table 2.

4.4.1 Operation Time (R_1.). This metric estimates the time required for an SDN attack to cause impact and for a defense to be developed and deployed. Instead of statistical aggregates, we assign categorical scores based on time ranges (e.g., seconds, minutes, hours, days, weeks), as reported in the original study or derived from replication efforts. The aim is not precise measurement but consistent differentiation between rapid threats and slower mitigation efforts. Attacks and defenses are scored separately to reflect their distinct timelines. Definitions appear in row R_1. of Table 5.

4.4.2 Equipment Cost (R_2.). This metric evaluates the equipment cost required to carry out an SDN attack or defense scheme—the framework rates attacks and defenses with higher equipment costs as more expensive. We have based the four rating levels on SDN literature, blogs, and industry reports.[2]

4.4.3 Work Force Requirement (R_3.). This metric evaluates the workforce required to execute an SDN attack and develop its defense. Attacks usually demand fewer resources, as breaking a system is more straightforward than defending it. This metric has eight levels, four for attacks and four for defenses.

4.4.4 Expertise (R_4.). This metric evaluates the technical skill level required to develop and execute an SDN attack or to design and implement its corresponding defense. Higher expertise generally correlates with increased development cost and complexity. Given the differing demands of offensive and defensive efforts, we define eight discrete rating levels—four tailored to attacks and four to defenses—each reflecting the depth of domain knowledge, tooling proficiency, and operational familiarity needed.

Temporal Resource Assessment Metrics. Temporal resource metrics do not alter the inherent severity of a vulnerability or defense but instead, track how the practical cost and feasibility of exploiting or mitigating it evolve over time. For example, the operation time, expertise, or equipment cost required to launch an attack may decrease as tools become public, while defense costs may rise due to patch deployment at scale. Definitions for these metrics are detailed in Table 14 in Appendix 10.2.

4.5 Visibility Assessment Metrics

While traditional vulnerability assessments emphasize technical severity, security incidents and defenses in programmable environments like SDN often carry broader implications. To capture these dimensions, we introduce the Visibility (V) metrics, which

[2]These definitions serve as a starting point for evaluating SDN attack or defense costs, with future refinements possible.

Table 6: Visibility Assessment Metrics (V-series) in the Odin Framework for Evaluating Strategic & Operational Prominence of SDN Attacks and Defenses.

(*Notation*: VLX → Level X for V where X = 0, 1, 2, 3).

	Metric	Metric Description	Value Level	Score	Value Explanation
V_1.	Venue	Evaluates the visibility of the SDN vulnerability or defense based on publishing venue and practical deployment.	VL0	0.0	Unknown or missing details.
			VL1	0.25	Published in lesser-known venues or reported only in academic labs, no practical SDN deployment.
			VL2	0.5	Published at reputable mid-tier venues or demonstrated in SDN testbeds with limited real-world adoption.
			VL3	1.0	Published in a top-tier venue or assigned a CVE, with confirmed deployment or exploitation in production SDN systems (e.g., cloud networks, enterprise data centers).
V_2.	Social Impact	Measures the societal and operational impact of the SDN security issue or defense. Citation counts, vendor recognition, and public awareness specific to SDN.	VL0	0.0	Unknown or missing impact data.
			VL1	0.25	Low impact: fewer than 20 citations, no vendor advisories or industry alerts.
			VL2	0.5	Moderate impact: 20–50 citations, some vendor advisories or niche community recognition.
			VL3	1.0	High impact: over 50 citations, widespread vendor advisories, inclusion in security standards, or international media coverage.

evaluate the strategic and operational prominence, influence, and societal impact of a given SDN attack or defense. These metrics reflect a technique's practical reach—whether it has been widely cited, adopted, or deployed—and its perceived influence within research and operational communities (as formalized in Equation 4 of Table 2). The Visibility category consists of two metrics, described in detail in Table 6.

4.5.1 Venue (V_1.) This metric evaluates the prominence of an SDN attack or defense based on the venue or platform where it was reported, such as open-source platforms and security conferences. We use conference rankings from [9] to define four scoring ranges for this metric (details are in Table 6).

4.5.2 Social Impact (V_2.) This metric assesses the societal impact of an SDN attack or defense, considering factors like business effects (e.g., revenue change) and media coverage. The attack or defense is scored based on its impact level, as outlined in row V_2. of Table 6.

Temporal Visibility Assessment Metrics. Rather than indicating technical impact, temporal visibility metrics monitor how community recognition and societal significance of an SDN vulnerability or defense evolve with time. For instance, while the venue subscore is static—anchored to the publication outlet—the social impact subscore may change as the community engages with the work (e.g., through citations, deployments, or policy discussions). These temporal shifts reflect external recognition or operational significance changes rather than technical properties. Definitions for the temporal social impact subscore are provided in Table 15 in Appendix 10.3.

4.6 Implementation

We have developed the Odin framework as a lightweight web-based app that enables users to input values for various assessment metrics and obtain corresponding composite scores. These scores reflect key evaluation dimensions: technical impact or effectiveness (primary score), practical feasibility (resource score), and societal or research prominence (visibility score).

5 Functionality Demonstration

In this section, we demonstrate the applicability of the Odin framework by evaluating 20 SDN attack–defense pairs selected from Table 10, with results summarized in Table 7.[3] These pairs span all SDN architectural layers and include a mix of well-cited, underexplored, and recent studies to reflect a broad spectrum of technical impact and design diversity.

To conserve space, detailed score breakdowns for the first three pairs appear in Table 16 (Appendix 11), and their temporal score variations are shown in Table 8. We also compare Odin against CVSS and the ODL categorization system for this subset in Table 9. While some attacks in Table 7 lack CVE identifiers—often due to non-disclosure or reporting gaps—they have been peer-reviewed and published in reputable venues. To ensure a fair comparison, we derive unofficial CVSS estimates for these entries based on publicly available technical details, denoted as "U" in Table 7.

Odin's primary scores match or exceed CVSS ratings across all evaluated cases, demonstrating that the framework aligns with widely recognized standards while also capturing dimensions that CVSS does not fully consider. Its multi-dimensional lens—including resource requirements, operational prominence, visibility, scalability, and vendor independence—provides additional insight into the real-world impact of SDN attacks and the effectiveness of SDN defenses. By incorporating dimensions such as vendor independence and scalability, Odin assigns higher severity to attacks that CVSS underrates. For example, CVSS categorizes the Cross-App Poisoning attack [98] (AT. 4 in Table 7) as medium severity, whereas Odin more accurately reflects its cascading and systemic risks. In contrast, for the Controller Information Flood attack [54] (AT. 1), Odin aligns with CVSS on severity but extends the evaluation by explicitly considering the attack's resource requirements, visibility within the network, and the robustness of available defense strategies [4]. These additional dimensions provide a richer operational perspective that CVSS alone cannot capture.

[3] Any pair listed in Table 10 can be evaluated using the Odin framework.

Table 7: Odin's Application to SDN Attacks-SDN Defenses.

(*Notation*: U → Unofficial Score, O → Official Score, AT. → Attack, DF. → Defense, N/A → Not Available, N/A* → Not Applicable.).
(The Odin score breakdown is provided in Tables 17, 18, and 19, in Appendix 11.)

ID	SDN Attack-Defense Title	Year	Odin Primary Score	Odin Resource Score	Odin Visibility Score	CVSS (Base) Score
AT. 1	Controller Information Flood [54]	2018	High (4.5)	High (2.0)	Critical (2.0)	O: High (7.5) [3]
DF. 1A	OpenFlow Plugin-962 [4]	2018	Low (0.75)	Medium (1.0)	Medium (0.5)	N/A*
DF. 1B	Heap Utilization Limit [54]	2018	Low (0.0)	Low (0.0)	High (1.0)	N/A*
DF. 1C	Eirene [60]	2022	High (5.0)	High (2.25)	Medium (0.75)	–
AT. 2	Blurred Responsibilities [54]	2018	Critical (7.0)	High (2.0)	Critical (2.0)	O: Critical (9.8) [7]
DF. 2A	OpenFlow Plugin-971 [5]	2018	Low (0.75)	Medium (1.25)	Medium (0.5)	N/A*
DF. 2B	Node Reconciliation [54]	2018	Low (0.5)	Low (0.0)	High (1.0)	N/A*
AT. 3	Cache Invalidation [54]	2018	High (5.25)	Medium (1.0)	Critical (2.0)	O: High (7.5) [2]
DF. 3	AAA-151 [1]	2018	Medium (2.25)	Medium (1.0)	Medium (0.5)	N/A*
AT. 4	Cross-App Poisoning [98]	2018	High (5.0)	High (3.0)	Critical (2.0)	U: Medium (5.4)
DF. 4	Prov-SDN [98]	2018	High (5.0)	High (2.25)	Critical (2.0)	N/A*
AT. 5	Control Plane Reflection Attack [113]	2018	Medium (3.25)	High (2.25)	High (1.0)	U: Medium (4.3)
DF. 5	SWGuard [113]	2018	High (3.5)	High (2.25)	High (1.0)	N/A*
AT. 6	Port Amnesia & Port Probing [93]	2018	Medium (2.75)	High (1.75)	High (1.0)	U: Medium (4.3)
DF. 6	TopoGuard+ [93]	2018	High (3.0)	High (2.0)	High (1.0)	N/A*
AT. 7	Covert Channel Attacks [79]	2018	Medium (2.5)	High (2.25)	High (1.0)	U: Medium (4.3)
DF. 7	Covert Channel Defender [79]	2018	High (3.0)	High (1.75)	High (1.0)	N/A*
AT. 8	Link Flooding Attack [101]	2018	Medium (2.5)	High (1.75)	High (1.0)	U: Medium (4.3)
DF. 8	Link Flooding Defender [101]	2018	High (3.0)	High (1.75)	High (1.0)	N/A*
AT. 9	Crossfire Table-Overflow [108]	2019	Medium (2.5)	Medium (1.5)	Medium (0.75)	U: Medium (4.3)
DF. 9	Fire Guard [108]	2019	High (3.0)	High (1.75)	Medium (0.75)	N/A*
AT. 10	Topology Freezing & Reverse Loop [81]	2019	High (3.5)	High (2.75)	High (1.5)	U: Medium (6.3)
DF. 10	Cryptographic Key for MAC tag over DPID [81]	2019	Medium (1.75)	High (2.25)	High (1.5)	N/A*
AT. 11	Cross-path Attack [45]	2019	High (5.5)	High (3.0)	Critical (2.0)	U: Medium (5.4)
DF. 11	Reserving Bandwidth, Prioritizing Control Traffic [45]	2019	Critical (5.5)	High (2.25)	Critical (2.0)	N/A*
AT. 12	Fingerprinting Match Fields of Flow Rules [61]	2020	Medium (3.25)	High (2.5)	Medium (0.5)	U: Medium (4.3)
DF. 12	Postponing Flow Installation [61]	2020	High (3.25)	High (1.75)	Medium (0.5)	N/A*
AT. 13	Buffered Packet Hijacking [47]	2020	High (5.25)	High (3.0)	High (1.25)	U: Medium (5.4)
DF. 13	ConCheck [47]	2020	Critical (5.0)	High (2.25)	High (1.25)	N/A*
AT. 14	SYN Flood [85]	2020	High (3.75)	High (2.25)	Medium (0.75)	U: Medium (5.4)
DF. 14A	AEGIS [85]	2020	High (3.25)	High (1.75)	Medium (0.75)	N/A*
DF. 14B	SYNGuard [83]	2021	Medium (2.5)	High (1.75)	Medium (0.5)	N/A*
AT. 15A	Fingerprinting Critical Flow Rules [103]	2021	Medium (2.5)	High (2.5)	Medium (0.5)	U: Medium (4.3)
AT. 15B	Fingerprinting Network and Controller Type [103]	2021	Medium (2.5)	High (2.5)	Medium (0.5)	U: Medium (4.3)
DF. 15	Probabilistic Scrambling-Controller Dynamic Scheduling [103]	2021	High (2.75)	High (2.0)	Medium (0.5)	N/A*
AT. 16	Cross Path Attack [107]	2022	High (4.25)	High (3.0)	Medium (0.75)	U: Medium (5.4)
DF. 16	Cross Guard [107]	2022	High (4.5)	High (2.25)	Medium (0.75)	N/A*
AT. 17	Invisible Assailant Attack (IAA) [75]	2022	Medium (2.25)	High (2.5)	Medium (0.75)	U: Medium (4.3)
DF. 17	Route Path Verification (RPV) [75]	2022	High (3.0)	High (1.75)	Medium (0.75)	N/A*
AT. 18	DHCP DoS and Starvation [66]	2023	High (3.75)	Medium (1.25)	Low (0.25)	Medium (6.5)
DF. 18	DHCP DoS and Starvation Mitigation [66]	2023	High (4.5)	Medium (1.5)	Low (0.25)	N/A*
AT. 19	Flow Table Overflow [97]	2023	Medium (2.0)	Medium (1.25)	Low (0.25)	U: Medium (6.5)
DF. 19	FTODefender [97]	2023	Medium (2.5)	Medium (1.25)	Low (0.25)	N/A*
AT. 20	Marionette Attacks [49]	2024	Critical (7.0)	High (2.0)	Critical (2.0)	O: Critical (9.1) [11]
DF. 20	Not Released yet.	2024	N/A	N/A	N/A	N/A

We next demonstrate the Odin score calculation for the Controller Information Flood attack and its mitigation (Table 7). The remaining evaluations in Table 7 follow the same process. Readers are encouraged to reverse-engineer the scores in Table 7 to gain a deeper understanding of the framework's evaluation logic and cross-check their reasoning against the detailed breakdowns in Appendix 11 (Tables 17, 18, and 19).

5.0.1 Controller Information Flood (AT. 1) [54]. Dixit et al. [54] demonstrated an SDN controller attack that sends mutated configurations to the controller's datastore, rendering it unavailable to legitimate applications. Evaluated for 2018 in Table 7, this attack's Odin scores are detailed in Table 16 (Appendix 11), using primary metrics P_1.–PA_9., resource metrics R_1.–R_4., and visibility metrics V_1.–V_2.. The attack was successfully replicated on real-world SDN controllers (ODL and ONOS), yielding `PL3 (1.0)` for correctness

Table 8: Temporal Odin Score Changes Computation.

(*Notation*: (↓) → Reduced, (-) → No change, (↑) → Increased, N/A* → Not Applicable).

ID	SDN Attack-Defense Title	Year	Odin Primary Score	Odin Resource Score	Odin Visibility Score	CVSS (Base) Score
AT. 1	Controller Information Flood [54]	2025	Medium (2.75) (↓)	High (2.0) (-)	Critical (2.0) (-)	O : Medium (5.0) (↓) [3]
DF. 1A	OpenFlow Plugin-962 [4]	2025	Low (0.75) (-)	Medium (1.0) (-)	Medium (0.5) (-)	N/A*
DF. 1B	Heap Utilization Limit [54]	2025	Low (0.0) (-)	Low (0.0) (-)	High (1.0) (-)	N/A*
DF. 1C	Eirene [60]	2025	High (5.0)	High (2.25)	Medium (0.75)	N/A*
AT. 2	Blurred Responsibilities [54]	2025	High (4.0) (↓)	High (2.0) (-)	Critical (2.0) (-)	O : High (7.5) (↓) [7]
DF. 2A	OpenFlow Plugin-971 [5]	2025	Low (0.75) (-)	Medium (1.25) (-)	Medium (0.5) (-)	N/A*
DF. 2B	Node Reconciliation [54]	2025	Low (0.5) (-)	Low (0.0) (-)	High (1.0) (-)	N/A*
AT. 3	Cache Invalidation [54]	2025	High (3.5) (↓)	Medium (1.0) (-)	Critical (2.0) (-)	O : Medium (5.0) (↓) [2]
DF. 3	AAA-151 [1]	2025	High (3.0) (-)	Medium (1.0) (-)	Medium (0.5) (-)	N/A*

Table 9: Comparative Breakdown: Odin Base Score vs. CVSS and ODL—highlighting the added SDN-specific granularity.

	SDN Attack Title	Year	CVSS Vector	CVSS (Base) Score [91]	ODL Cat. [21]	Odin Primary Score
AT. 1	Controller Information Flood [54]	2018	AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H [3]	O: High (7.5) [3]	High [4]	High (4.5)
AT. 2	Blurred Responsibilities [54]	2018	AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H [7]	O : Critical (9.8) [7]	Medium [5]	Critical (7.0)
AT. 3	Cache Invalidation [54]	2018	AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N [2]	O: High (7.5) [2]	High [1]	High (5.25)
AT. 1	Controller Information Flood [54]	2025	AV:N/AC:L/Au:N/C:N/I:N/A:P [3]	O: Medium (5.0) [3]	Resolved [4]	Medium (2.75)
AT. 2	Blurred Responsibilities [54]	2025	AV:N/AC:L/Au:N/C:P/I:P/A:P [7]	O: High (7.5) [7]	Medium [5]	High (4.0)
AT. 3	Cache Invalidation [54]	2025	AV:N/AC:L/Au:N/C:N/I:P/A:N [2]	O: Medium (5.0) [2]	Resolved [1]	High (3.5)

(P_1.). It was tested on a small campus network but can scale by increasing the volume of mutated rules, resulting in PL2 (0.5) for scalability (P_2.). The attack was carried out on two major SDN controllers, thereby vendor independence (P_3.) is PL2 (0.5). Due to the availability of open attack code, reproducibility (P_4.) is PL3 (1.0), while performance complexity (P_5.) is PL2 (0.5). The impact on confidentiality (PA_6.) is negligible (PL0 (0.0)), but integrity (PA_7.) and availability (PA_8.) are both rated PL1 (0.25) (due to SDN controller compromise). With a CVSS base score of high (7.5) [3], the severity level (PA_9.) is PL2 (0.5) (based on CVSS 2018 rating). These values result in a primary score of high (4.5).

For resource costs, the Controller Information Flood attack [54] was executed by five security experts using a standard laptop to inject mutated configurations into the SDN datastore. We confirmed the attack could be reproduced on a machine with average computing power within five hours. Accordingly, the operation time requirement (R_1.) is RL3 (1.0), while equipment cost (R_2.) and workforce (R_3.) are both RL1 (0.25). The required expertise (R_4.) is rated RL2 (0.5), yielding a total resource score of high (2.0) using Equation 3. This attack appeared in a tier-1 security venue and was assigned a CVE (CVE-2017-1000411), earning VL3 (1.0) for both venue (V_1.) and social impact (V_2.). The visibility score, computed via Equation 4, is high (2.0).

5.0.2 OpenFlow Plugin-962 [4] (DF. 1A). DF. 1A OpenFlow Plugin-962 [4], developed in 2018 to mitigate the Controller Information Flood attack is evaluated in Table 7 with a detailed breakdown in Table 16 (Appendix 11).

Due to lack of documentation on correctness (P_1.), scalability (P_2.), reproducibility (P_4.), and performance impact (PD_6.), these metrics are scored PL0 (0.0). Since the patch is specific to OpenDaylight (ODL), vendor independence (P_3.) and complexity (P_5.) are both rated PL1 (0.25). The defense is a basic rate limiter that only partially mitigates the issue; thus, resilience (PD_7.) is also PL1 (0.25). Using Equation 2, the resulting primary score is low (0.75).

Resource-wise, time and workforce (R_1., R_3.) are unknown and scored RL0 (0.0). Estimated development costs for similar patches fall within the USD 1,000–10,000 range, giving equipment cost (R_2.) a RL2 (0.5) rating. The patch was authored by ODL maintainers with moderate experience, so expertise level (R_4.) is also RL2 (0.5), resulting in a total resource score of medium (1.0). This patch was not published in an academic or industry venue (V_1. = VL0 (0.0)), but it was integrated into ODL releases, contributing to moderate community recognition (V_2. = VL2 (0.5)). Thus, the overall visibility score is medium (0.5).

Temporal Changes. Table 8 illustrates the dynamic adjustment capabilities of the Odin framework by re-evaluating the first three SDN attack–defense pairs from Table 7 for the year 2025—approximately six years after their initial publication, except for DF. 1C *Eirene* [60], which was introduced in 2022. As before, we highlight a representative pair for detailed discussion: AT. 1 *Controller Information Flood* [54] and its corresponding defense, DF. 1A *OpenFlow Plugin-962* [4].

The temporal metrics capture how an attack's practical impact, defense effectiveness, and deployment cost evolve over time. For the *Controller Information Flood* attack, the temporal integrity and availability impact (TPA_7. and TPA_8.) each decrease by TPL1 (−0.25) following the release of *OpenFlow Plugin-962* (2018) and the more comprehensive *Eirene* framework (2022), both of which mitigate the underlying controller datastore overflow issue. This mitigation also reduces the temporal severity level (TPA_9.) by TPL1 (−0.25), consistent with the CVSS base score dropping from High (7.5)

to Medium (5.0) [3]. Furthermore, as mitigation frameworks mature and exploit code becomes unavailable or obsolete, the attack's reproducibility (TP_4.) decreases by TPL4 (−1.0).

As a result, the Odin primary score for this attack adjusts from High (4.0) to Medium (2.75), reflecting improved defense efficacy. Notably, Resource and Visibility scores remain unchanged. Since the *OpenFlow Plugin-962* patch has not received any further updates or adoption traction, its Odin scores remain static. In contrast, *Eirene*, with a significantly higher primary score, offers a more robust defense than both *OpenFlow Plugin-962* and the simpler heap utilization patch proposed in [54].

This case study demonstrates how Odin's temporal metrics dynamically reflect changes in defense effectiveness, attack relevance, and operational manageability over time—complementing the static primary metrics that characterize the inherent properties of each attack or defense at a fixed point in time.

Comparison with CVSS and ODL Categorization Systems. Table 9 compares the severity ratings of three SDN attacks using CVSS [91], the ODL categorization system [21], and Odin. Unlike CVSS, which provides generic scores, and ODL, which relies on opaque internal categories, Odin offers more SDN-specific and transparent evaluations. It assigns equal or higher severity to impactful attacks, reflecting their real-world feasibility and architectural implications. Moreover, Odin uniquely supports scoring of both attacks and defenses—something neither CVSS nor ODL can do—enabling more comprehensive SDN risk assessment.

6 Discussion

This section reflects on the broader value, limitations, and practical implications of the Odin framework—particularly in contrast with existing tools and the context of increasingly complex SDN deployments across cloud and enterprise environments.

6.1 Why a New Framework?

While CVSS remains the de facto standard for vulnerability scoring, it was not designed with SDN in mind. Its lack of SDN-specific context—such as scalability across topologies and vendor independence—limits its applicability in programmable networks. Similarly, OpenDaylight's internal categorization labels threats through opaque, non-reproducible logic tied to specific platforms. Odin fills this gap by offering the first open, lightweight, and extensible scoring framework that assesses both SDN attack impact and defense effectiveness. By evaluating technical impact, deployment feasibility, and operational visibility—alongside their temporal evolution— Odin enables consistent, SDN-aware assessments that support practical risk decisions in both cloud and enterprise deployments. Importantly, Odin's qualitative metrics are grounded in clearly defined scoring levels. These levels combine structured criteria with expert judgment, reflecting practical SDN realities while maintaining transparency and consistency. This structured approach enables meaningful comparisons across attacks and defenses—even without proprietary datasets (e.g., closed enterprise logs) or exhaustive telemetry (e.g., fine-grained packet traces).

6.2 Insights from Comparative Evaluation

Table 9 provides comparative insights across CVSS, ODL, and Odin, demonstrating how Odin enhances traditional evaluations through SDN-specific granularity. Key takeaways include:

- *Alignment with existing standards:* In all three attack scenarios, Odin's primary scores match or exceed CVSS ratings, validating its risk sensitivity while surfacing additional context (Table 9).
- *Architectural fidelity:* ODL's medium rating of *Blurred Responsibilities* conflicts with its critical CVSS score and broader architectural implications (Table 9). Odin more accurately reflects its impact by incorporating scalability and cross-vendor relevance.
- *Defense evaluation:* Unlike CVSS and the ODL categorization system [21], Odin scores defenses explicitly. For example, it captures the performance cost and resilience of mitigations like *OpenFlow Plugin-962* [4]. In turn, Odin supports a deeper analysis of how severe attacks are counterbalanced by the effectiveness of deployed defenses.

These comparisons suggest that Odin offers a more complete, context-sensitive foundation for prioritizing threats and validating defenses in programmable, multi-tenant SDN deployments.

6.3 Practical Use and Future Directions

Odin provides a structured, SDN-aware scoring framework that enables transparent risk assessment across attacks and defenses—even when CVSS or platform-specific scores are unavailable. While its thresholds and weights are currently based on heuristics and SDN literature, future refinement using operational feedback, real-world data, and comparative studies (e.g., against CVSS or ODL) can improve accuracy and generalizability. By extending traditional scoring to capture vendor independence, scalability, resource constraints, and operational visibility, Odin addresses gaps in systemic and cascading risk coverage. Although some defense descriptions remain high-level and formal equations may limit theoretical depth, the framework offers an immediate, practical foundation for reproducible and consistent evaluation in programmable, multi-tenant SDN environments, with clear opportunities for further enhancement.

7 Conclusion

SDN's growing role in cloud and enterprise environments calls for security evaluation frameworks tailored to its unique architectural and operational challenges. Traditional models often fall short—overlooking critical aspects such as vendor independence, scalability, and cross-layer interactions—while proprietary solutions limit transparency and interoperability. The Odin framework addresses this gap as the first open-source, extensible system specifically designed to assess both the impact of SDN attacks and the effectiveness of defenses. By complementing established tools like CVSS with operationally grounded, SDN-specific metrics, Odin enables consistent and reproducible assessments across diverse deployments. This supports security strategies that scale alongside SDN growth, while remaining attuned to platform-specific risks.

Acknowledgments

This work relates to the Department of Navy award N00014-24-1-2193 issued by the Office of Naval Research. This material is based upon work supported by the Advanced Research Projects Agency for Health (ARPA-H) under Contract No. SP4701-23-C-0074 and the National Science Foundation (NSF) under Grant No. CNS-2141547. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research, ARPA-H, or NSF. Sana Habib gratefully acknowledges support from the Fulbright Fellowship and the HIVE Fellowship at the Center for Digital Resilience. We also thank the anonymous reviewers of CCSW'25 for their thoughtful and constructive feedback.

References

[1] 2017. AAA-151: Previous password continues to work after password change. *https://jira.opendaylight.org/browse/AAA-151* (2017).
[2] 2017. CVE-2017-1000406. *https://www.cvedetails.com/cve/CVE-2017-1000406/* (2017).
[3] 2017. CVE-2017-1000411. *https://www.cvedetails.com/cve/CVE-2017-1000411/* (2017).
[4] 2017. OPNFLWPLUG-962. Multiple "expired" flows take up the memory resource of CONFIG DS which leads to Controller shutdown. https://jira.opendaylight.org/browse/OPNFLWPLUG-962. (2017).
[5] 2017. OPNFLWPLUG-971. Node reconciliation installs old, expired flows: switch conquers flows for infinite time. https://jira.opendaylight.org/browse/OPNFLWPLUG-971. (2017).
[6] 2018. Center for Complex Networks and Systems Research (CNetS), Indiana University Bloomington. *https://cnets.indiana.edu/data-repository-for-nan-group/* (2018).
[7] 2018. CVE-2018-1078. *https://www.cvedetails.com/cve/CVE-2018-1078/* (2018).
[8] 2019. Prioritizing Vulnerability Response: A Stakeholder-Specific Vulnerability Categorization. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=636379. (2019).
[9] 2022. Computer Security Conference Ranking and Statistic. https://people.engr.tamu.edu/guofei/sec_conf_stat.htm. (2022).
[10] 2023. Stanford Large Network Dataset Collection. *http://snap.stanford.edu/data/index.html* (2023).
[11] 2024. CVE-2024-37018. *https://www.cvedetails.com/cve/CVE-2024-37018/* (2024).
[12] 2025. The 3D Internet Topology Zoo. (2025). https://github.com/afourmy/3D-internet-zoo
[13] 2025. Cloud SDN - Ericsson. https://www.ericsson.com/en/portfolio/cloud-software-and-services/cloud-core/cloud-infrastructure/nfvi/cloud-sdn. (2025).
[14] 2025. Component-based Software Defined Networking Framework. Build SDN Agilely. (2025). https://ryu-sdn.org/
[15] 2025. Energy Sciences Network (ESnet): 100G SDN Test-bed. (2025). https://www.es.net/network-r-and-d/experimental-network-testbeds/100g-sdn-testbed/
[16] 2025. European Telecommunications Standards Institute (ETSI). (2025). https://www.etsi.org/
[17] 2025. Exploit Prediction Scoring System. https://www.first.org/epss/. (2025).
[18] 2025. Floodlight. (2025). https://github.com/floodlight/floodlight
[19] 2025. Global Environment for Network Innovations (GENI). (2025). https://www.geni.net/about-geni/what-is-geni/
[20] 2025. International Telecommunication Union (ITU-T) Recommendations. *https://www.itu.int/en/ITU-T/publications/pages/recs.aspx* (2025).
[21] 2025. Jira Software. (2025). https://jira.opendaylight.org
[22] 2025. Lumen Enterprise SDN - Software-Defined Networking. (2025). https://www.lumen.com/
[23] 2025. Mininet. (2025). http://mininet.org/
[24] 2025. Netcat. (2025). https://nmap.org/ncat/
[25] 2025. Netresec, Publicly available PCAP files. *https://www.netresec.com/?page=PcapFiles* (2025).
[26] 2025. Netty. (2025). https://netty.io/
[27] 2025. Nmap. (2025). https://nmap.org/
[28] 2025. Open Network Operating System (ONOS). (2025). https://onosproject.org
[29] 2025. Open Networking Foundation Security Working Group. [online]. *https://opennetworking.org/tag/onf-security-working-group/* (2025).
[30] 2025. OpenDaylight. *https://www.opendaylight.org* (2025).
[31] 2025. Software Defined Networking (Sdn) Market Size, Share, Growth and Industry Trends. (2025). https://www.marketsandmarkets.com/Market-Reports/software-defined-networking-sdn-market-655.html
[32] 2025. Thoth Lab. (2025). https://pypi.org/project/thoth-lab/
[33] 2025. Vulnerability Priority Rating (VPR). https://www.tenable.com/sc-dashboards/vulnerability-priority-rating-vpr-summary. (2025).
[34] POX. 2020. 2020. https://github.com/noxrepo/pox. (2020).
[35] ACTFORNET. 2025. Huawei Software Defined Network (SDN) Solution. https://actfornet.com/huawei-cloud/sdn.html. (2025).
[36] Bilal Ahmed, Nadeem Ahmed, Asad Waqar Malik, Mohsin Jafri, and Taimur Hafeez. 2020. Fingerprinting SDN Policy Parameters: An Empirical Study. *IEEE Access* 8 (2020), 142379–142392.
[37] Abdullah Al-Alaj, Ram Krishnan, and Ravi Sandhu. 2019. Sdn-rbac: An access control model for sdn controller applications. In *2019 4th International Conference on Computing, Communications and Security (ICCCS)*. IEEE, 1–8.
[38] Sarwan Ali, Maria Khalid Alvi, Safi Faizullah, Muhammad Asad Khan, Abdullah Alshanqiti, and Imdadullah Khan. 2020. Detecting DDoS attack on SDN due to vulnerabilities in OpenFlow. In *2019 International Conference on Advances in the Emerging Computing Technologies (AECT)*. IEEE, 1–6.
[39] Amir Alimohammadifar, Suryadipta Majumdar, Taous Madi, Yosr Jarraya, Makan Pourzandi, Lingyu Wang, and Mourad Debbabi. 2018. Stealthy probing-based verification (spv): An active approach to defending software defined networks against topology poisoning attacks. In *European Symposium on Research in Computer Security*. Springer, 463–484.
[40] Sonali Sen Baidya and Rattikorn Hewett. 2020. Link Discovery Attacks in Software-Defined Networks: Topology Poisoning and Impact Analysis. *Journal of Communications* 15, 8 (2020).
[41] Roberto Bifulco, Heng Cui, Ghassan O Karame, and Felix Klaedtke. 2015. Fingerprinting software-defined networks. In *2015 IEEE 23rd International Conference on Network Protocols (ICNP)*. IEEE, 453–459.
[42] Celyn Birkinshaw, Elpida Rouka, and Vassilios G Vassilakis. 2019. Implementing an intrusion detection and prevention system using software-defined networking: Defending against port-scanning and denial-of-service attacks. *Journal of Network and Computer Applications* 136 (2019), 71–85.
[43] M Bjorklund, J Schoenwaelder, P Shafer, K Watsen, and R Wilton. 2018. *Network Management Datastore Architecture (NMDA)*. Technical Report. RFC 8342. RFC Editor.
[44] Thanh Bui, Markku Antikainen, and Tuomas Aura. 2019. Analysis of topology poisoning attacks in software-defined networking. In *Nordic Conference on Secure IT Systems*. Springer, 87–102.
[45] Jiahao Cao, Qi Li, Renjie Xie, Kun Sun, Guofei Gu, Mingwei Xu, and Yuan Yang. 2019. The CrossPath Attack: Disrupting the SDN Control Channel via Shared Links. In *28th USENIX Security Symposium (USENIX Security 19)*. 19–36.
[46] Jiahao Cao, Kun Sun, Qi Li, Mingwei Xu, Zijie Yang, Kyung Joon Kwak, and Jason Li. 2019. Covert Channels in SDN: Leaking Out Information from Controllers to End Hosts. In *International Conference on Security and Privacy in Communication Systems*. Springer, 429–449.
[47] Jiahao Cao, Renjie Xie, Kun Sun, Qi Li, Guofei Gu, and Mingwei Xu. 2020. When match fields do not need to match: Buffered packets hijacking in SDN. In *Proc. of the Network and Distributed System Security Symposium (NDSS'20)*.
[48] Jiahao Cao, Zijie Yang, Kun Sun, Qi Li, Mingwei Xu, and Peiyi Han. 2019. Fingerprinting SDN Applications via Encrypted Control Traffic. In *22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)*. 501–515.
[49] Mingming Chen, Thomas La Porta, Teryl Taylor, Frederico Araujo, and Trent Jaeger. 2024. Manipulating OpenFlow Link Discovery Packet Forwarding for Topology Poisoning. *arXiv preprint arXiv:2408.16940* (2024).
[50] Ankur Chowdhary, Dijiang Huang, Gail-Joon Ahn, Myong Kang, Anya Kim, and Alexander Velazquez. 2019. SDNSOC: Object oriented SDN framework. In *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*. 7–12.
[51] Laizhong Cui, F Richard Yu, and Qiao Yan. 2016. When big data meets software-defined networking: SDN for big data and big data for SDN. *IEEE network* 30, 1 (2016), 58–65.
[52] Spyros Denazis, Evangelos Haleplidis, Jamal Hadi Salim, Odysseas Koufopavlou, David Meyer, and Kostas Pentikousis. 2015. Software-defined networking (SDN): Layers and architecture terminology. (2015).
[53] Mohan Dhawan, Rishabh Poddar, Kshiteej Mahajan, and Vijay Mann. 2015. SPHINX: Detecting Security Attacks in Software-Defined Networks.. In *NDSS*, Vol. 15. 8–11.
[54] Vaibhav Hemant Dixit, Adam Doupé, Yan Shoshitaishvili, Ziming Zhao, and Gail-Joon Ahn. 2018. AIM-SDN: Attacking Information Mismanagement in SDN-datastores. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 664–676.
[55] Chip Elliott. 2008. GENI-global environment for network innovations. In *2008 33rd IEEE Conference on Local Computer Networks (LCN)*. IEEE, 8–8.
[56] Cisco SDN – Software Defined Networking Explained. 2025. Cisco SDN. (2025). https://study-ccna.com/cisco-sdn-software-defined-networking/
[57] Thomas Girdler and Vassilios G Vassilakis. 2021. Implementing an intrusion detection and prevention system using Software-Defined Networking: Defending against ARP spoofing attacks and Blacklisted MAC Addresses. *Computers & Electrical Engineering* 90 (2021), 106990.

[58] Steven R Gomez, Samuel Jero, Richard Skowyra, Jason Martin, Patrick Sullivan, David Bigelow, Zachary Ellenbogen, Bryan C Ward, Hamed Okhravi, and James W Landry. 2019. Controller-Oblivious Dynamic Access Control in Software-Defined Networks. In *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 447–459.

[59] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown, and Scott Shenker. 2008. NOX: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review* 38, 3 (2008), 105–110.

[60] Sana Habib, Tiffany Bao, Yan Shoshitaishvili, and Adam Doupé. 2022. Mitigating Threats Emerging from the Interaction between SDN Apps and SDN (Configuration) Datastore. In *Proceedings of the 2022 on Cloud Computing Security Workshop*. 23–39.

[61] Jianwei Hou, Minjian Zhang, Ziqi Zhang, Wenchang Shi, Bo Qin, and Bin Liang. 2020. On the fine-grained fingerprinting threat to software-defined networks. *Future Generation Computer Systems* 107 (2020), 485–497.

[62] Hongxin Hu, Wonkyu Han, Gail-Joon Ahn, and Ziming Zhao. 2014. FLOWGUARD: building robust firewalls for software-defined networks. In *Proceedings of the third workshop on Hot topics in software defined networking*. ACM, 97–102.

[63] Jingyu Hua, Zidong Zhou, and Sheng Zhong. 2020. Flow Misleading: Worm-Hole Attack in Software-Defined Networking via Building In-Band Covert Channel. *IEEE Transactions on Information Forensics and Security* 16 (2020), 1029–1043.

[64] Gan Huang and Hee Yong Youn. 2020. Proactive eviction of flow entry for SDN based on hidden Markov model. *Frontiers of Computer Science* 14, 4 (2020), 1–10.

[65] IBM. 2025. IBM. Software Defined Networking (SDN) Services. (2025). https://www.ibm.com/ae-en/services/network/software-defined

[66] Hafiz Usama Ishtiaq, Areeb Ahmed Bhutta, and Adnan Noor Mian. 2023. DHCP DoS and starvation attacks on SDN controllers and their mitigation. *Journal of Computer Virology and Hacking Techniques* (2023), 1–11.

[67] Igor Ivkić, Dominik Thiede, Nicholas Race, Matthew Broadbent, and Antonios Gouglidis. 2022. Security Evaluation in Software-Defined Networks. In *International Conference on Cloud Computing and Services Science*. Springer, 66–91.

[68] Miyoung Kang, Jin Young Choi, Hee Hwan Kwak, Inhye Kang, Myung Ki Shin, and Jong Hwa Yi. 2015. Formal modeling and verification for SDN firewall application using pACSR. In *Electronics, Communications and Networks IV*. CRC Press, 155.

[69] Miyoung Kang, Eun-Young Kang, Dae-Yon Hwang, Beom-Jin Kim, Ki-Hyuk Nam, Myung-Ki Shin, and Jin-Young Choi. 2013. Formal modeling and verification of SDN-OpenFlow. In *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*. IEEE, 481–482.

[70] Peyman Kazemian, Michael Chang, Hongyi Zeng, George Varghese, Nick McKeown, and Scott Whyte. 2013. Real time network policy checking using header space analysis. In *10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*. 99–111.

[71] HARMAN YI KHALID, PARISHAN M ISMAEL, and AHMAD BAHEEJ ALKHALIL. 2019. Efficient Mechanism for Securing Software Defined Network Against ARP Spoofing Attack. *Journal of Duhok University* 22, 1 (2019), 124–131.

[72] Rafiullah Khan, Kieran McLaughlin, David Laverty, and Sakir Sezer. 2017. STRIDE-based threat modeling for cyber-physical systems. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE, 1–6.

[73] Sajad Khorsandroo and Ali Saman Tosun. 2018. Time inference attacks on software defined networks: Challenges and countermeasures. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*. IEEE, 342–349.

[74] Jinwoo Kim, Minjae Seo, Seungsoo Lee, Jaehyun Nam, Vinod Yegneswaran, Phillip Porras, Guofei Gu, and Seungwon Shin. 2024. Enhancing security in SDN: Systematizing attacks and defenses from a penetration perspective. *Computer Networks* (2024), 110203.

[75] Dezhang Kong, Yi Shen, Xiang Chen, Qiumei Cheng, Hongyan Liu, Dong Zhang, Xuan Liu, Shuangxi Chen, and Chunming Wu. 2022. Combination Attacks and Defenses on SDN Topology Discovery. *IEEE/ACM Transactions on Networking* (2022).

[76] Prashant Kumar, Meenakshi Tripathi, Ajay Nehra, Mauro Conti, and Chhagan Lal. 2018. SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN. *IEEE Transactions on Network and Service Management* 15, 4 (2018), 1545–1559.

[77] Seungsoo Lee, Seungwon Woo, Jinwoo Kim, Vinod Yegneswaran, Phillip Porras, and Seungwon Shin. 2020. AudiSDN: Automated detection of network policy inconsistencies in software-defined networks. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 1788–1797.

[78] Seungsoo Lee, Changhoon Yoon, Chanhee Lee, Seungwon Shin, Vinod Yegneswaran, and Phillip A Porras. 2017. DELTA: A Security Assessment Framework for Software-Defined Networks.. In *NDSS*.

[79] Qi Li, Yanyu Chen, Patrick PC Lee, Mingwei Xu, and Kui Ren. 2018. Security policy violations in SDN data plane. *IEEE/ACM Transactions on Networking* 26, 4 (2018), 1715–1727.

[80] Qian Lv, Jing Zhu, Fen Zhou, and Zuqing Zhu. 2020. Network planning with bilevel optimization to address attacks to physical infrastructure of SDN. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[81] Eduard Marin, Nicola Bucciol, and Mauro Conti. 2019. An In-depth Look Into SDN Topology Discovery Mechanisms: Novel Attacks and Practical Countermeasures. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1101–1114.

[82] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review* 38, 2 (2008), 69–74.

[83] Mohamed Rahouti, Kaiqi Xiong, Nasir Ghani, and Farooq Shaikh. 2021. SYNGuard: Dynamic threshold-based SYN flood attack detection and mitigation in software-defined networks. *IET Networks* 10, 2 (2021), 76–87.

[84] Longyan Ran, Yunhe Cui, Chun Guo, Qing Qian, Guowei Shen, and Huanlai Xing. 2022. Defending saturation attacks on SDN controller: A confusable instance analysis-based algorithm. *Computer Networks* 213 (2022), 109098.

[85] Nagarathna Ravi, S Mercy Shalinie, Mauro Conti, et al. 2020. AEGIS: Detection and Mitigation of TCP SYN Flood on SDN Controller. *IEEE Transactions on Network and Service Management* (2020).

[86] Nicolas Schnepf, Rémi Badonnel, Abdelkader Lahmadi, and Stephan Merz. 2018. Synaptic: A formal checker for SDN-based security policies. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–2.

[87] Network Screen. 2025. Juniper Networks SDN. (2025). https://www.networkscreen.com/SDN-Series.asp

[88] PV Shalini, V Radha, and Sriram G Sanjeevi. 2023. Early detection and mitigation of TCP SYN flood attacks in SDN using chi-square test. *The Journal of Supercomputing* 79, 9 (2023), 10353–10385.

[89] Yi Shen, Chunming Wu, Dezhang Kong, and Qiumei Cheng. 2023. Flow Table Saturation Attack against Dynamic Timeout Mechanisms in SDN. *Applied Sciences* 13, 12 (2023), 7210.

[90] Seungwon Shin, Vinod Yegneswaran, Phillip Porras, and Guofei Gu. 2013. Avantguard: Scalable and vigilant switch flow management in software-defined networks. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 413–424.

[91] Common Vulnerability Scoring System SIG. 2025. Common Vulnerability Scoring System. https://www.first.org/cvss/. (2025).

[92] Parminder Singh and Selvakumar Manickam. 2015. Design and deployment of OpenStack-SDN based test-bed for EDoS. In *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)*. IEEE, 1–5.

[93] Richard Skowyra, Lei Xu, Guofei Gu, Veer Dedhia, Thomas Hobson, Hamed Okhravi, and James Landry. 2018. Effective topology tampering attacks and defenses in software-defined networks. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 374–385.

[94] Rochak Swami, Mayank Dave, and Virender Ranga. 2023. Mitigation of DDoS Attack Using Moving Target Defense in SDN. *Wireless Personal Communications* (2023), 1–15.

[95] Dan Tang, Siyuan Wang, Boru Liu, Wenqiang Jin, and Jiliang Zhang. 2023. GASF-IPP: Detection and Mitigation of LDoS Attack in SDN. *IEEE Transactions on Services Computing* (2023).

[96] Dan Tang, Xiyin Wang, Yudong Yan, Dongshuo Zhang, and Huan Zhao. 2022. ADMS: An online attack detection and mitigation system for LDoS attacks via SDN. *Computer Communications* 181 (2022), 454–471.

[97] Dan Tang, Zhiqing Zheng, Chao Yin, Bing Xiong, Zheng Qin, and Qiuwei Yang. 2023. Ftodefender: An Efficient Flow Table Overflow Attacks Defending System in SDN. *Available at SSRN 4438152* (2023).

[98] Benjamin E Ujcich, Samuel Jero, Anne Edmundson, Qi Wang, Richard Skowyra, James Landry, Adam Bates, William H Sanders, Cristina Nita-Rotaru, and Hamed Okhravi. 2018. Cross-app poisoning in software-defined networking. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 648–663.

[99] Evgenii Vinarskii, Jorge Lopez, Natalia Kushik, Nina Yevtushenko, and Djamal Zeghlache. 2019. A model checking based approach for detecting SDN races. In *IFIP International Conference on Testing Software and Systems*. Springer, 194–211.

[100] Haopei Wang, Guangliang Yang, Phakpoom Chinprutthiwong, Lei Xu, Yangyong Zhang, and Guofei Gu. 2018. Towards fine-grained network security forensics and diagnosis in the sdn era. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 3–16.

[101] Juan Wang, Ru Wen, Jiangqi Li, Fei Yan, Bo Zhao, and Fajiang Yu. 2018. Detecting and mitigating target link-flooding attacks using sdn. *IEEE Transactions on Dependable and Secure Computing* 16, 6 (2018), 944–956.

[102] Lei Wang, Qing Li, Yong Jiang, Xuya Jia, and Jianping Wu. 2018. Woodpecker: Detecting and mitigating link-flooding attacks via SDN. *Computer Networks* 147 (2018), 1–13.

[103] Tao Wang and Hongchang Chen. 2021. A Lightweight SDN Fingerprint Attack Defense Mechanism Based on Probabilistic Scrambling and Controller Dynamic Scheduling Strategies. *Security and Communication Networks* 2021 (2021), 1–23.

[104] Pengpeng Wu, Lin Yao, Chi Lin, Guowei Wu, and Mohammad S Obaidat. 2018. FMD: A DoS mitigation scheme based on flow migration in software-defined networking. *International Journal of Communication Systems* 31, 9 (2018), e3543.

[105] Jing Xia, Zhiping Cai, Gang Hu, and Ming Xu. 2019. An active defense solution for ARP spoofing in OpenFlow network. *Chinese Journal of Electronics* 28, 1 (2019), 172–178.
[106] Feng Xiao, Jinquan Zhang, Jianwei Huang, Guofei Gu, Dinghao Wu, and Peng Liu. 2020. Unexpected Data Dependency Creation and Chaining: A New Attack to SDN. In *Proceedings of the 41st IEEE Symposium on Security and Privacy (S&P'20)*.
[107] Renjie Xie, Jiahao Cao, Qi Li, Kun Sun, Guofei Gu, Mingwei Xu, and Yuan Yang. 2022. Disrupting the SDN Control Channel via Shared Links: Attacks and Countermeasures. *IEEE/ACM Transactions on Networking* (2022).
[108] Jianfeng Xu, Liming Wang, Chen Song, and Zhen Xu. 2019. An Effective Table-Overflow Attack and Defense in Software-Defined Networking. In *2019 IEEE 44th LCN Symposium on Emerging Topics in Networking (LCN Symposium)*. IEEE, 10–17.
[109] Meng Yue, Qingxin Yan, Han Zheng, Zhijun Wu, et al. 2022. Cross-Plane DDoS Attack Defense Architecture Based on Flow Table Features in SDN. *Security and Communication Networks* 2022 (2022).
[110] Noe M Yungaicela-Naula, Cesar Vargas-Rosales, Jesús Arturo Pérez-Díaz, and Diego Fernando Carrera. 2022. A flexible SDN-based framework for slow-rate DDoS attack mitigation by using deep reinforcement learning. *Journal of Network and Computer Applications* 205 (2022), 103444.
[111] Menghao Zhang, Jun Bi, Jiasong Bai, and Guanyu Li. 2018. Floodshield: Securing the sdn infrastructure against denial-of-service attacks. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 687–698.
[112] Minjian Zhang, Jianwei Hou, Ziqi Zhang, Wenchang Shi, Bo Qin, and Bin Liang. 2017. Fine-Grained Fingerprinting Threats to Software-Defined Networks. In *2017 IEEE Trustcom/BigDataSE/ICESS*. IEEE, 128–135.
[113] Menghao Zhang, Guanyu Li, Lei Xu, Jun Bi, Guofei Gu, and Jiasong Bai. 2018. Control plane reflection attacks in SDNs: new attacks and countermeasures. In *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 161–183.
[114] Jing Zheng, Qi Li, Guofei Gu, Jiahao Cao, David K.Y. Yau, and Jianping Wu. 2018. Realtime DDoS Defense Using COTS SDN Switches via Adaptive Correlation Analysis. *IEEE Transactions on Information Forensics and Security (TIFS)* (2018).
[115] Jing Zheng, Qi Li, Guofei Gu, Jiahao Cao, David KY Yau, and Jianping Wu. 2018. Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis. *IEEE Transactions on Information Forensics and Security* 13, 7 (2018), 1838–1853.
[116] Jing Zhu, Marija Furdek, Carlos Natalino, Lena Wosinska, and Zuqing Zhu. 2019. How to survive targeted fiber cuts: A game theoretic approach for resilient SDON control plane design. In *International IFIP Conference on Optical Network Design and Modeling*. Springer, 168–180.
[117] Enrico Zio. 2013. Monte carlo simulation: The method. In *The Monte Carlo simulation method for system reliability and risk analysis*. Springer, 19–58.

8 SDN Attacks-SDN Defenses

Table 10 presents a survey of 42 SDN attacks and 45 defenses published between 2018 and 2024, spanning vulnerabilities across all layers of the SDN architecture. Notable trends include multi-vector threats, low-rate DoS techniques [95, 96], and covert reconnaissance [48], while defenses range from protocol-level patches [66] to Moving Target Defense [94] and formal verification approaches [98, 99]. This classification highlights persistent challenges—such as complex attack surfaces, varying defense granularity, and the gap between academic proposals and real-world deployment—reinforcing the need for a unified, SDN-aware scoring framework. For deeper technical analysis, see the original works and the review in [74]. Our framework supports consistent and reproducible comparison across this diverse threat landscape.

9 Step-by-Step Walkthrough of Score Range Algorithm

This section provides a detailed step-by-step computation of the score range algorithm (Algorithm 1) for computing visibility score range (i.e., the number of metrics α is set to 2). This example illustrates how lower and upper score bounds are computed for each severity level.

9.1 Initialization

- Metric count: $\alpha = 2$
- Rating levels: `[0.0, 0.25, 0.5, 1.0]`
- Compute $\beta = \lfloor \alpha/2 \rfloor = \lfloor 2/2 \rfloor = 1$
- Initialize `lower_limit` = 0.0
- Initialize severity index $i = 1$

9.2 Step-by-Step Computation

(1) **Severity = Low ($i = 1$):**

$$\text{upper} = (rating\_levels[0] \cdot (\alpha - \beta)) + (rating\_levels[1] \cdot \beta)$$
$$= (0.0 \cdot 1) + (0.25 \cdot 1) = 0.25$$
$$\text{lower} = 0.0$$

Update:
- `lower_limit` = 0.25
- $i = 2$

(2) **Severity = Medium ($i = 2$):**

$$\text{upper} = (rating\_levels[1] \cdot (\alpha - \beta)) + (rating\_levels[2] \cdot \beta)$$
$$= (0.25 \cdot 1) + (0.5 \cdot 1) = 0.75$$
$$\text{lower} = 0.25$$

Update:
- `lower_limit` = 0.75
- $i = 3$

(3) **Severity = High ($i = 3$):**

$$\text{upper} = (rating\_levels[2] \cdot (\alpha - \beta)) + (rating\_levels[3] \cdot \beta)$$
$$= (0.5 \cdot 1) + (1.0 \cdot 1) = 1.5$$
$$\text{lower} = 0.75$$

Update:
- `lower_limit` = 1.5
- $i = 4$

(4) **Severity = Critical ($i = 4$):**

$$\text{upper} = rating\_levels[3] \cdot \alpha = 1.0 \cdot 2 = 2.0$$
$$\text{lower} = 1.5$$

The final score ranges are summarized in Table 11.

9.3 Temporal Categories

The same computation process applies to temporal categories by substituting the `rating_levels` with `temporal_levels` = `[0.0, 0.25, 0.5, 0.75, 1.0]`. For $\alpha = 2$, the resulting ranges are numerically identical.

10 Temporal Assessment Metrics

As previously noted, all assessment metrics have a corresponding temporal component to account for temporal changes, with the exception of the Venue metric under the Visibility assessment category.

10.1 Temporal Primary Assessment Metrics

Temporal primary assessment metrics do not change a vulnerability's intrinsic severity but capture how the associated risk evolves over time. For example, after a fix is deployed, security teams reassess residual risk using CVSS, Odin, or similar frameworks to

Table 10: A List of Notable SDN Attacks and SDN Defenses.

Year	SDN Attack Title (Component/s Affected)
2024	Manipulating OpenFlow Link Discovery Packet Forwarding for Topology Poisoning (Marionette Attacks) [49]
2023	(i) Low-rate Denial of Service (LDoS) (Data Plane) [95]; (ii) Flow Table Saturation Attack (Flow Table) [89]; (iii) DHCP DoS and Starvation [66] (Control Plane and Data Plane); (iv) TCP SYN Flood (Control Plane and Data Plane) [88]; (v) Flow Table Overflow [97] (Data Plane);
2022	(i) Cross Path (Communication Interfaces) [107]; (ii) Invisible Assailant Attack (IAA) (Links in Data plane) [75]; (iii) Low-rate Denial of Service (LDoS) (Control Plane and Data Plane) [96]; (iv) Slow-rate DDoS Attack (Control Plane and Data Plane) [110];
2021	(i) ARP Request, ARP Reply, ARP Reply Destination Attack (Hosts in Data plane) [57]; (ii) Fingerprinting Network and Controller Type (SDN Controller and Data Plane) [103]; (iii) Fingerprinting Critical Flow Rules (Switches in Data Plane) [103];
2020	(i) SYN Flood (Communication Interfaces) [85]; (ii) Disrupting SDN switches (Switches in Data Plane) [38]; (iii) Zombie Host (Communication Interfaces and Hosts in Data Plane) [38]; (iv) Buffered Packet Hijacking (SDN Controller and Data Plane) [47]; (v) Network Policy Inconsistencies (SDN Controller and Data Plane) [77]; (vi) Fingerprinting Match Fields of Flow Rules (SDN Controller and Data Plane) [61, 112]; (vii) Inter-channel Cross Talk (SDN Controller and Data Plane) [80]; (viii) Link Discovery Attacks (Links in Data Plane) [40]; (ix) Worm-Hole Attack (SDN Controller and Data Plane) [63]; (x) Data Dependency Creation & Hijacking (SDN Controller and Data Plane) [106]; (xi) Flow Table Entry Attack (Switches in Data Plane) [36];
2019	(i) Cross-path Attack (Communication Interfaces) [45]; (ii) Crossfire Table-Overflow Attack (Switches in Data Plane) [108]; (iii) Fingerprinting SDN Applications Using Encrypted Control Traffic (SDN Controller and SDN Apps) [48]; (iv) Exploiting Covert Channels (SDN Controller, Data Plane, and SDN Apps) [46]; (v) Targeted Fiber Cuts (Data Plane) [116]; (vi) Single-switch, Two-switch, Extended Two-switch Tunnel Attack (Data Plane) [44]; (vii) Topology Freezing and Reverse Loop (Data Plane) [81];
2018	(i) Configuration Information Flood (SDN Controller) [54]; (ii) Link Flood (Links in Data Plane) [101, 102]; (iv) Control-Data plane Saturation Attack (Communication Interfaces) [111]; (v) Cross-App Poisoning (SDN Controller and SDN Apps) [98]; (vi) Covert Channel Attacks (Data Plane) [79]; (vii) Cache Invalidation (SDN Controller) [54]; (viii) Control Plane Reflection Attack (SDN Controller and Data Plane) [113]; (ix) Estimating Flow Table Size & Flow State Reconnaissance (SDN Controller and Data Plane) [73]; (x) Port Amnesia & Port Probing (Data Plane) [93]; (xi) Blurred Responsibilities (SDN Controller and Data Plane) [54];
	SDN Defense Title (Component/s Secured)
2023	(i) GASF-IPP (Data Plane) [95]; (ii) Moving Target Defence (MTD) based DDoS Mitigation (Control Plane and Data Plane) [94]; (iii) DHCP DoS and Starvation Mitigation (Control Plane and Data Plane) [66]; (iv) TCP SYN Flood Mitigation (Control Plane and Data Plane) [88]; (v) FtoDefender: Flow Table Overflow Mitigation [97] (Data Plane);
2022	(i) Cross Guard (Communication Interfaces) [107]; (ii) LICENSE (SDN Controller) [84]; (iii) Route Path Verification (RPV) (Data Plane) [75]; (iv) Cross-Plane DDoS Attack Defense (SDN Controller and Data Plane) [109]; (v) Eirene (SDN controller) [60]; (vi) Attack Detection and Mitigation System (ADMS) [96]; (vii) Slow-rate DDoS Attack Mitigation (Control Plane and Data Plane) [110];
2021	(i) SYNGuard [83] (Control Plane and Data Plane); (ii) Probabilistic Scrambling and Controller Dynamic Scheduling (SDN Controller and Data Plane) [103]; (iii) Intrusion Detection and Prevention System (IDPS) (Hosts in Data Plane) [57];
2020	(i) AEGIS (Communication Interfaces) [85]; (ii) Zombie Host Detection (Communication Interfaces and Hosts in Data Plane) [38]; (iii) ConCheck (SDN Controller and Data Plane) [47]; (iv) Postponing Flow Installation (Control Plane and Data Plane) [61]; (v) Bilevel Optimization (SDN Controller and Data Plane) [80]; (vi) SVHunter (SDN Controller and Data Plane) [106];
2019	(i) Reserving Bandwidth & Prioritizing Control Traffic (Communication Interfaces) [45]; (ii) FireGuard (Switches in Data Plane) [108]; (iii) SDN-RBAC (SDN Controller) [37]; (iii) Controller-Oblivious **D**ynamic Acesss Control using **F**low **I**solation (DFI) (SDN Controller) [58]; (iv) Credit-Based Threshold Random Walk (CB-TRW) and Rate Limiting (RL) (Control Plane and Data Plane) [42]; (v) SDNSOC (Control Plane) [50]; (vi) Game Theoretic Approach for Resilient Control Plane Design (Control Plane) [116]; (vii) Cryptographic Key to Compute MAC tag over DPID (Control Plane and Data Plane) [81]; (viii) Model Checking based Approach for Identifying SDN Races (Control Plane) [99]; (ix) ARP Poisoning Detection & Prevention by Checking Every ARP Packet (Data Plane) [71]; (x) Active ARP Inspection (AAI) (Data Plane) [105];
2018	(i) Adaptive Correlation Analysis (Data Plane) [114]; (ii) FMD (Control Plane and Data Plane) [104]; (iii) Woodpecker (Links in Data Plane) [102]; (iv) RADAR (Switches in Data Plane) [115]; (v) LFA (Link Flooding Attack) Defender (Links in Data Plane) [101]; (vi) FloodShield (Communication Interfaces) [111]; (vii) SAFETY (Control Plane and Data Plane) [76]; (viii) PROVSDN (SDN Controller and SDN Apps) [98]; (ix) Covert Channel Defender (Data Plane) [79]; (x) Synaptic (SDN Controller) [86]; (xi) ForenGuard (Control Plane and Data Plane) [100]; (xii) TopoGuard+ (Data Plane) [93]; (xiii) SWGuard (SDN Controller and Data Plane) [113]; (xiv) Stealthy probing-based verification (SPV) (Control Plane and Data Plane) [39];

Table 11: Score Ranges for Visibility (i.e., $\alpha = 2$)

Severity	*Lower Bound*	*Upper Bound*
Low	0.00	0.25
Medium	0.25	0.75
High	0.75	1.50
Critical	1.50	2.00

guide prioritization. These metrics support dynamic risk management by reflecting how factors like patching, exploit development, or defense deployment influence real-world urgency and impact. Definitions are provided in rows TP_1. to TP_5. of Table 12 and TPA_6., TPA_7., TPD_6., and TPD_7. of Table 13.

10.2 Temporal Resource Assessment Metrics

These metrics do not reflect changes in a vulnerability's inherent severity but rather track how the practical effort—such as operation time, equipmemnt cost, or expertise—required to exploit or mitigate it shifts over time. For example, public exploits may lower attack

Table 12: Temporal Primary Assessment Metrics (TP-series) in the Odin Framework for Evaluating SDN Security.

	Metric	Metric Description	Value Level	Score	Value Explanation
TP_1.	Temporal Correctness	Accounts for how temporal correctess verification changes over time.	TPL0	0.0	No change, unknown, or missing details.
			TPL1	0.25	Correctness improvement: PL0 ⇒ PL1 or PL1 ⇒ PL2.
			TPL2	0.5	Correctness improvement: PL0 ⇒ PL2 or PL2 ⇒ PL3.
			TPL3	0.75	Correctness improvement: PL1 ⇒ PL3.
			TPL4	1.0	Correctness improvement: PL0 ⇒ PL3.
TP_2.	Temporal Scalability	Accounts for how scalability changes over time.	TPL0	0.0	No change, unknown, or missing details.
			TPL1	0.25	Scalability improvement: PL0 ⇒ PL1 or PL1 ⇒ PL2.
			TPL2	0.5	Scalability improvement: PL0 ⇒ PL2 or PL2 ⇒ PL3.
			TPL3	0.75	Scalability improvement: PL1 ⇒ PL3.
			TPL4	1.0	Scalability improvement: PL0 ⇒ PL3.
TP_3.	Temporal Vendor Independence	Accounts for how vendor independence changes over time.	TPL0	0.0	No change, unknown, or missing details.
			TPL1	±0.25	Level transitions: PL0 ⇔ PL1, PL1 ⇔ PL2.
			TPL2	±0.5	Level transitions: PL0 ⇔ PL2, PL2 ⇔ BL3.
			TPL3	±0.75	Level transition: PL1 ⇔ PL3.
			TPL4	±1.0	Level transition: PL0 ⇔ PL3.
TP_4.	Temporal Reproducibility	Accounts for how reproducibility of an SDN attack or defense changes over time.	TPL0	0.0	No change, unknown, or missing details.
			TPL1	±0.25	Level transitions: PL0 ⇔ PL1, PL1 ⇔ PL2.
			TPL2	±0.5	Level transitions: PL0 ⇔ PL2, PL2 ⇔ PL3.
			TPL3	±0.75	Level transition: PL1 ⇔ PL3.
			TPL4	±1.0	Level transition: PL0 ⇔ PL3.
TP_5.	Temporal Complexity	Accounts for how complexity of an SDN attack or defense changes over time.	TPL0	0.0	No change, unknown, or missing details.
			TPL1	±0.25	Level transitions: PL0 ⇔ PL1, PL1 ⇔ PL2.
			TPL2	±0.5	Level transitions: PL0 ⇔ PL2, PL2 ⇔ PL3.
			TPL3	±0.75	Level transition: PL1 ⇔ PL3.
			TPL4	±1.0	Level transition: PL0 ⇔ PL3.

Table 13: Temporal Primary Attack-Impact (TPA-series)-Defense-Effectiveness (TPD-series) Assessment Metrics in the Odin Framework for Evaluating SDN Security.

	Metric	Metric Description	Value Level	Score	Value Explanation
TPA_6.	Temporal Confidentiality, Integrity, and Availability Impact (C , I, A)	Accounts for temporal changes in the impact of an SDN attack.	C, I, A : TPL0	0.0	No change or unknown or missing details.
			C, I, A : TPL1	± 0.25	Transition b/w levels: (C: PL0 ⇔ C: PL1) | (C: PL1 ⇔ C: PL2) | (I: PL0 ⇔ I: PL1) | (I: PL1 ⇔ I: PL2) | (A: PL0 ⇔ A: PL1) | (A: PL1 ⇔ A: PL2)
			C, I, A : TPL2	± 0.5	Transition b/w levels: (C: PL0 ⇔ C: PL2) | (C: PL1 ⇔ C: PL3) | (I: PL0 ⇔ I: PL2) | (I: PL1 ⇔ I: PL3) | (A: PL0 ⇔ A: PL2) | (A: PL1 ⇔ A: PL3).
			C , I, A : TPL3	± 0.75	Transition b/w levels: (C: PL1 ⇔ C: PL3) | (I: PL1 ⇔ I: PL3) | (A: PL1 ⇔ A: PL3).
			C, I, A : TPL4	± 1.0	Transition b/w levels: (C: PL0 ⇔ C: PL3) | (I: PL0 ⇔ I: PL3) | (A : PL0 ⇔ A : PL3).
TPA_7.	Temporal Severity Level	Accounts for temporal changes in the severity level of an SDN attack.	TPL0	0.0	No change or unknown or missing details.
			TPL1	± 0.25	Transition b/w levels: (PL0 ⇔ PL1) | (PL1 ⇔ PL2).
			TPL2	± 0.5	Transition b/w levels: (PL0 ⇔ PL2) | (PL2 ⇔ PL3).
			TPL3	± 0.75	Transition b/w levels: (PL1 ⇔ PL3).
			TPL4	± 1.0	Transition b/w levels: (PL0 ⇔ PL3).
TPD_6.	Temporal Performance Cost	Accounts for temporal changes in the performance cost of an SDN defense.	TPL0	0.0	No change or unknown or missing details.
			TPL1	± 0.25	Transition b/w levels: (PL0 ⇔ PL1) | (PL1 ⇔ PL2).
			TPL2	± 0.5	Transition b/w levels: (PL0 ⇔ PL2) | (PL2 ⇔ PL3).
			TPL3	± 0.75	Transition b/w levels: PL1 ⇔ PL3.
			TPL4	± 1.0	Transition b/w levels: PL0 ⇔ PL3.
TPD_7.	Temporal Resilience to Disruption	Accounts for temporal changes in the resilience of an SDN defense.	TPL0	0.0	No change or unknown or missing details.
			TPL1	± 0.25	Transition b/w levels: (PL0 ⇔ PL1) | (PL1 ⇔ PL2).
			TPL2	± 0.5	Transition b/w levels: (PL0 ⇔ PL2) | (PL2 ⇔ PL3).
			TPL3	± 0.75	Transition b/w levels: PL1 ⇔ PL3.
			TPL4	± 1.0	Transition b/w levels: PL0 ⇔ PL3.

Table 14: Temporal Resource Assessment Metrics (TR-series) in the Odin Framework for Evaluating Resource Burden of SDN Attacks and Defenses.

(*Notation*: TRLX → Level X for TR, where X = 0, 1, 2, 3, 4.)

	Metric	Metric Description	Value Level	Score	Value Explanation
TR_1.	Temporal Operation Time	Accounts for temporal changes in operation time to execute or mitigate SDN attacks or defenses.	TRL0	0.0	No change or unknown.
			TRL1	± 0.25	Small increase or decrease between adjacent resource levels (e.g., RL0 ⇔ RL1).
			TRL2	± 0.5	Moderate shift in time (e.g., RL0 ⇔ RL2 or RL2 ⇔ RL3).
			TRL3	± 0.75	Large jump in required time (e.g., RL1 ⇔ RL3).
			TRL4	± 1.0	Complete shift in time requirement (e.g., RL0 ⇔ RL3).
TR_2.	Temporal Equipment Cost	Accounts for temporal changes in hardware or software cost for executing or mitigating SDN attacks or defenses.	TRL0	0.0	No change or unknown.
			TRL1	± 0.25	Small cost shift between adjacent resource levels (RL0 ⇔ RL1 or RL1 ⇔ RL2).
			TRL2	± 0.5	Moderate cost shift (RL0 ⇔ RL2 or RL2 ⇔ RL3).
			TRL3	± 0.75	Large change in cost (RL1 ⇔ RL3).
TR_3.	Temporal Workforce Requirement	Accounts for changes in personnel requirements for SDN CP/DP attack or defense over time.	TRL0	0.0	No change or unknown.
			TRL1	± 0.25	Minor personnel change (RL0 ⇔ RL1 or RL1 ⇔ RL2).
			TRL2	± 0.5	Moderate shift in workforce size (RL0 ⇔ RL2 or RL2 ⇔ RL3).
			TRL3	± 0.75	Large workforce shift (RL1 ⇔ RL3).
			TRL4	± 1.0	Full personnel change (RL0 ⇔ RL3).
TR_4.	Temporal Expertise	Accounts for changes in required skill level for SDN CP/DP attack or defense over time.	TRL0	0.0	No change or unknown.
			TRL1	± 0.25	Minor skill level change (RL0 ⇔ RL1 or RL1 ⇔ RL2).
			TRL2	± 0.5	Moderate skill level shift (RL0 ⇔ RL2 or RL2 ⇔ RL3).
			TRL3	± 0.75	Significant jump in expertise required (RL1 ⇔ RL3).
			TRL4	± 1.0	Maximum skill level change (RL0 ⇔ RL3).

Table 15: Temporal Visibility Assessment Metrics (TV-series) in the Odin Framework for Evaluating Strategic Operational Prominence of SDN Attacks and Defenses.

(*Notation*: TVLX → Level X for TV, where X = 0, 1, 2, 3).

	Metric	Metric Description	Value Level	Score	Value Explanation
TV.	Temporal Social Impact	Measures the magnitude of change in social impact over time (i.e., changes in citations, deployment status.	TVL0	0.0	No change or missing details.
			TVL1	± 0.25	Small change between adjacent levels (VL0 ⇔ VL1 or VL1 ⇔ VL2).
			TVL2	± 0.5	Moderate change (VL0 ⇔ VL2 or VL2 ⇔ VL3).
			TVL3	± 0.75	Significant jump in visibility (VL1 ⇔ VL3).
			TVL4	± 1.0	Complete shift from no visibility to top-tier recognition (VL0 ⇔ VL3).

costs, while widespread patching may increase defense overhead. Definitions are provided in Table 14.

10.3 Temporal Visibility Assessment Metrics

Temporal visibility metrics are designed to capture the evolving community engagement and societal relevance of an SDN vulnerability or defense. Unlike the static venue subscore—anchored to the original publication outlet—the social impact subscore reflects dynamic factors such as citations, real-world deployments, and policy influence. These changes indicate shifts in external recognition rather than alterations in the underlying technical severity. Formal definitions are provided in Table 15.

11 Odin Score Breakdown

To illustrate the practical application of the Odin framework, Table 16 presents a detailed breakdown of scores for representative SDN attack–defense pairs. The complete set of scores for all SDN attacks and defenses is provided in Tables 17, 18, and 19.

Table 16: Odin Breakdown of Primary, Resource, and Visibility Scores (for 2018 SDN Attacks and Defenses).

	SDN Attack- SDN Defense Title	P_1.	P_2.	P_3.	P_4.	P_5.	PA_6.	PA_7.	PA_8.
AT. 1	Controller Information Flood [54]	PL3	PL2	PL2	PL3	PL2	PL0	PL1	PL1
DF. 1A	OpenFlow Plugin-962 [4]	PL0	PL0	PL1	PL0	PL1	N/A*	N/A*	N/A*
DF. 1B	Heap Utilization Limit [54]	PL0	PL0	PL0	PL0	PL0	N/A*	N/A*	N/A*
DF. 1C	Eirene [60]	PL3	PL3	PL1	PL3	PL1	N/A*	N/A*	N/A*
AT. 2	Blurred Responsibilities [54]	PL3	PL3	PL2	PL3	PL3	PL0	PL3	PL2
DF. 2A	OpenFlow Plugin-971 [5]	PL0	PL0	PL1	PL0	PL1	N/A*	N/A*	N/A*
DF. 2B	Node Reconciliation [54]	PL0	PL0	PL0	PL0	PL0	N/A*	N/A*	N/A*
AT. 3	Cache Invalidation [54]	PL3	PL3	PL1	PL3	PL3	PL1	PL1	PL0
DF. 3	AAA-151 [1]	PL3	PL2	PL1	PL0	PL1	N/A*	N/A*	N/A*
	SDN Attack-SDN Defense Title	PA_9./PD_6.	PD_7.	R_1.	R_2.	R_3.	R_4.	V_1.	V_2.
AT. 1	Controller Information Flood [54]	PL2/N/A*	N/A*	RL3	RL1	RL1	RL2	VL3	VL3
DF. 1A	OpenFlow Plugin-962 [4]	N/A*/PL0	PL1	RL0	RL2	RL0	RL2	VL0	VL2
DF. 1B	Heap Utilization Limit [54]	N/A*/PL0	PL0	RL0	RL0	RL0	RL0	VL3	VL0
DF. 1C	Eirene [60]	N/A*/PL2	PL3	RL3	RL2	RL1	RL2	VL1	VL2
AT. 2	Blurred Responsibilities [54]	PL3/N/A*	N/A*	RL1	RL2	RL1	RL3	VL3	VL3
DF. 2A	OpenFlow Plugin-971 [5]	N/A*/PL0	PL1	RL0	RL2	RL1	RL2	VL0	VL2
DF. 2B	Node Reconciliation [54]	N/A*/PL0	PL2	RL0	RL0	RL0	RL0	VL3	VL0
AT. 3	Cache Invalidation [54]	PL2/N/A*	N/A*	RL1	RL1	RL1	RL1	VL3	VL3
DF. 3	AAA-151 [1]	N/A*/PL0	PL1	RL0	RL2	RL1	RL1	VL0	VL2

Note: PL*, RL*, and VL* levels 0–3 correspond to scores 0.0, 0.25, 0.5, and 1.0, respectively. N/A* → Not Applicable.

Table 17: Primary Score Breakdown for Table 7 (P_1.–P_5.).

	SDN Attacks-SDN Defenses	Year	P_1.	P_2.	P_3.	P_4.	P_5.
AT. 1	Controller Information Flood [54]	2018	PL3	PL2	PL2	PL3	PL2
DF. 1A	OpenFlow Plugin-962 [4]	2018	PL0	PL0	PL1	PL0	PL1
DF. 1B	Heap Utilization Limit [54]	2018	PL0	PL0	PL0	PL0	PL0
DF. 1C	Eirene [60]	2022	PL3	PL3	PL1	PL3	PL1
AT. 2	Blurred Responsibilities [54]	2018	PL3	PL3	PL2	PL3	PL3
DF. 2A	OpenFlow Plugin-971 [5]	2018	PL0	PL0	PL1	PL0	PL1
DF. 2B	Node Reconciliation [54]	2018	PL0	PL0	PL0	PL0	PL0
AT. 3	Cache Invalidation [54]	2018	PL3	PL3	PL1	PL3	PL3
DF. 3	AAA-151 [1]	2018	PL3	PL2	PL1	PL0	PL1
AT. 4	Cross-App Poisoning [98]	2018	PL2	PL3	PL3	PL2	PL3
DF. 4	Prov-SDN [98]	2018	PL2	PL3	PL3	PL2	PL3
AT. 5	Control Plane Reflection Attack [113]	2018	PL3	PL2	PL1	PL2	PL1
DF. 5	SWGuard [113]	2018	PL3	PL2	PL1	PL2	PL1
AT. 6	Port Amnesia & Port Probing [93]	2018	PL2	PL2	PL1	PL2	PL1
DF. 6	TopoGuard+ [93]	2018	PL2	PL2	PL1	PL2	PL1
AT. 7	Covert Channel Attacks [79]	2018	PL2	PL2	PL1	PL2	PL1
DF. 7	Covert Channel Defender [79]	2018	PL2	PL2	PL1	PL2	PL1
AT. 8	Link Flooding Attack [101]	2018	PL2	PL2	PL1	PL2	PL1
DF. 8	Link Flooding Defender [101]	2018	PL2	PL2	PL1	PL2	PL1
AT. 9	Crossfire Table-Overflow [108]	2019	PL2	PL2	PL1	PL2	PL1
DF. 9	Fire Guard [108]	2019	PL2	PL2	PL1	PL2	PL1
AT. 10	Topology Freezing & Reverse Loop [81]	2019	PL1	PL2	PL1	PL2	PL1
DF. 10	Cryptographic Key for MAC tag over DPID [81]	2019	PL2	PL2	PL1	PL1	PL1
AT. 11	Cross-path Attack [45]	2019	PL3	PL3	PL3	PL2	PL3
DF. 11	Reserving Bandwidth & Prioritizing Control Traffic [45]	2019	PL3	PL3	PL3	PL2	PL3
AT. 12	Fingerprinting Match Fields of Flow Rules [61]	2020	PL2	PL2	PL2	PL2	PL2
DF. 12	Postponing Flow Installation [61]	2020	PL2	PL2	PL2	PL2	PL2
AT. 13	Buffered Packet Hijacking [47]	2020	PL3	PL2	PL3	PL3	PL2
DF. 13	ConCheck [47]	2020	PL3	PL2	PL3	PL3	PL2
AT. 14	SYN Flood [85]	2020	PL3	PL2	PL1	PL2	PL1
DF. 14A	AEGIS [85]	2020	PL3	PL2	PL1	PL2	PL1
DF. 14B	SYNGuard [83]	2021	PL2	PL2	PL1	PL2	PL1
AT. 15A	Fingerprinting Critical Flow Rules [103]	2021	PL2	PL1	PL2	PL1	PL2
AT. 15B	Fingerprinting Network and Controller Type [103]	2021	PL2	PL1	PL2	PL1	PL2
DF. 15	Probabilistic Scrambling and Controller Dynamic Scheduling [103]	2021	PL2	PL1	PL2	PL1	PL2
AT. 16	Cross Path Attack [107]	2022	PL3	PL3	PL2	PL2	PL2
DF. 16	Cross Guard [107]	2022	PL3	PL3	PL2	PL2	PL2
AT. 17	Invisible Assailant Attack (IAA) [75]	2022	PL2	PL2	PL1	PL2	PL1
DF. 17	Route Path Verification (RPV) [75]	2022	PL2	PL2	PL1	PL2	PL1
AT. 18	DHCP DoS and Starvation [66]	2023	PL2	PL2	PL3	PL1	PL3
DF. 18	DHCP DoS and Starvation Mitigation [66]	2023	PL2	PL2	PL3	PL1	PL3
AT. 19	Flow Table Overflow [97]	2023	PL2	PL2	PL1	PL1	PL1
DF. 19	FTODefender [97]	2023	PL2	PL2	PL1	PL1	PL1
AT. 20	Marionette Attacks [49]	2024	PL3	PL3	PL1	PL3	PL3
DF. 20	Not Released yet.	2024	N/A	N/A	N/A	N/A	N/A

Note: PL* levels 0–3 correspond to scores 0.0, 0.25, 0.5, and 1.0, respectively. N/A → Not Available. N/A* → Not Applicable.

Table 18: Primary Score Breakdown for Table 7 (PA_6., PA_7., PA_8., PA_9., PD_6., PD_7.).

ID	SDN Attack-SDN Defense Title	Year	PA_6.	PA_7.	PA_8.	PA_9.	PD_6.	PD_7.
AT. 1	Controller Information Flood [54]	2018	PL0	PL1	PL1	PL2	N/A*	N/A*
DF. 1A	OpenFlow Plugin-962 [4]	2018	N/A*	N/A*	N/A*	N/A*	PL0	PL1
DF. 1B	Heap Utilization Limit [54]	2018	N/A*	N/A*	N/A*	N/A*	PL0	PL0
DF. 1C	Eirene [60]	2022	N/A*	N/A*	N/A*	N/A*	PL2	PL3
AT. 2	Blurred Responsibilities [54]	2018	PL0	PL3	PL2	PL3	N/A*	N/A*
DF. 2A	OpenFlow Plugin-971 [5]	2018	N/A*	N/A*	N/A*	N/A*	PL0	PL1
DF. 2B	Node Reconciliation [54]	2018	N/A*	N/A*	N/A*	N/A*	PL0	PL2
AT. 3	Cache Invalidation [54]	2018	PL1	PL1	PL0	PL2	N/A*	N/A*
DF. 3	AAA-151 [1]	2018	N/A*	N/A*	N/A*	N/A*	PL0	PL1
AT. 4	Cross-App Poisoning [98]	2018	PL0	PL2	PL1	PL1	N/A*	N/A*
DF. 4	Prov-SDN [98]	2018	N/A*	N/A*	N/A*	N/A*	PL2	PL2
AT. 5	Control Plane Reflection Attack [113]	2018	PL0	PL2	PL0	PL1	N/A*	N/A*
DF. 5	SWGuard [113]	2018	N/A*	N/A*	N/A*	N/A*	PL2	PL2
AT. 6	Port Amnesia & Port Probing [93]	2018	PL0	PL2	PL0	PL1	N/A*	N/A*
DF. 6	TopoGuard+ [93]	2018	N/A*	N/A*	N/A*	N/A*	PL2	PL2
AT. 7	Covert Channel Attacks [79]	2018	PL0	PL1	PL0	PL1	N/A*	N/A*
DF. 7	Covert Channel Defender [79]	2018	N/A*	N/A*	N/A*	N/A*	PL2	PL2
AT. 8	Link Flooding Attack [101]	2018	PL0	PL0	PL1	PL1	N/A*	N/A*
DF. 8	Link Flooding Defender [101]	2018	N/A*	N/A*	N/A*	N/A*	PL2	PL2
AT. 9	Crossfire Table-Overflow [108]	2019	PL0	PL0	PL1	PL1	N/A*	N/A*
DF. 9	Fire Guard [108]	2019	N/A*	N/A*	N/A*	N/A*	PL2	PL2
AT. 10	Topology Freezing & Reverse Loop [81]	2019	PL1	PL2	PL0	PL1	N/A*	N/A*
DF. 10	Cryptographic Key for MAC tag over DPID [81]	2019	N/A*	N/A*	N/A*	N/A*	PL0	PL0
AT. 11	Cross-path Attack [45]	2019	PL0	PL2	PL1	PL1	N/A*	N/A*
DF. 11	Reserving Bandwidth & Prioritizing Control Traffic [45]	2019	N/A*	N/A*	N/A*	N/A*	PL2	PL2
AT. 12	Fingerprinting Match Fields of Flow Rules [61]	2020	PL2	PL0	PL0	PL1	N/A*	N/A*
DF. 12	Postponing Flow Installation [61]	2020	N/A*	N/A*	N/A*	N/A*	PL1	PL2
AT. 13	Buffered Packet Hijacking [47]	2020	PL0	PL2	PL2	PL1	N/A*	N/A*
DF. 13	ConCheck [47]	2020	N/A*	N/A*	N/A*	N/A*	PL2	PL2
AT. 14	SYN Flood [85]	2020	PL0	PL2	PL2	PL1	N/A*	N/A*
DF. 14A	AEGIS [85]	2020	N/A*	N/A*	N/A*	N/A*	PL1	PL2
DF. 14B	SYNGuard [83]	2021	N/A*	N/A*	N/A*	N/A*	PL1	PL1
AT. 15A	Fingerprinting Critical Flow Rules [103]	2021	PL1	PL0	PL0	PL1	N/A*	N/A*
AT. 15B	Fingerprinting Network and Controller Type [103]	2021	PL1	PL0	PL0	PL1	N/A*	N/A*
DF. 15	Probabilistic Scrambling and Controller Dynamic Scheduling [103]	2021	N/A*	N/A*	N/A*	N/A*	PL2	PL1
AT. 16	Cross Path Attack [107]	2022	PL0	PL1	PL1	PL1	N/A*	N/A*
DF. 16	Cross Guard [107]	2022	N/A*	N/A*	N/A*	N/A*	PL2	PL2
AT. 17	Invisible Assailant Attack (IAA) [75]	2022	PL0	PL0	PL0	PL1	N/A*	N/A*
DF. 17	Route Path Verification (RPV) [75]	2022	N/A*	N/A*	N/A*	N/A*	PL2	PL2
AT. 18	DHCP DoS and Starvation [66]	2023	PL0	PL0	PL1	PL1	N/A*	N/A*
DF. 18	DHCP DoS and Starvation Mitigation [66]	2023	N/A*	N/A*	N/A*	N/A*	PL1	PL3
AT. 19	Flow Table Overflow [97]	2023	PL0	PL0	PL0	PL1	N/A*	N/A*
DF. 19	FTODefender [97]	2023	N/A*	N/A*	N/A	N/A*	PL1	PL2
AT. 20	Marionette Attacks [49]	2024	PL1	PL3	PL2	PL3	N/A*	N/A*
DF. 20	Not Released yet.	2024	N/A*	N/A*	N/A*	N/A*	N/A	N/A

Note: PL* levels 0–3 correspond to scores 0.0, 0.25, 0.5, and 1.0, respectively. N/A → Not Available. N/A* → Not Applicable.

Table 19: Resource and Visibility Score Breakdown for Table 7.

ID	SDN Attack-SDN Defense Title	Year	R_1.	R_2.	R_3.	R_4.	V_1.	V_2.
AT. 1	Controller Information Flood [54]	2018	RL3	RL1	RL1	RL2	VL3	VL3
DF. 1A	OpenFlow Plugin-962 [4]	2018	RL0	RL2	RL0	RL2	VL0	VL2
DF. 1B	Heap Utilization Limit [54]	2018	RL0	RL0	RL0	RL0	VL3	VL0
DF. 1C	Eirene [60]	2022	RL3	RL2	RL1	RL2	VL1	VL2
AT. 2	Blurred Responsibilities [54]	2018	RL1	RL2	RL1	RL3	VL3	VL3
DF. 2A	OpenFlow Plugin-971 [5]	2018	RL0	RL2	RL1	RL2	VL0	VL2
DF. 2B	Node Reconciliation [54]	2018	RL0	RL0	RL0	RL0	VL3	VL0
AT. 3	Cache Invalidation [54]	2018	RL1	RL1	RL1	RL1	VL3	VL3
DF. 3	AAA-151 [1]	2018	RL0	RL2	RL1	RL1	VL0	VL2
AT. 4	Cross-App Poisoning [98]	2018	RL3	RL2	RL2	RL3	VL3	VL3
DF. 4	Prov-SDN [98]	2018	RL2	RL2	RL1	RL3	VL3	VL3
AT. 5	Control Plane Reflection Attack [113]	2018	RL1	RL2	RL2	RL3	VL2	VL2
DF. 5	SWGuard [113]	2018	RL2	RL2	RL1	RL3	VL2	VL2
AT. 6	Port Amnesia & Port Probing [93]	2018	RL1	RL2	RL2	RL2	VL2	VL2
DF. 6	TopoGuard+ [93]	2018	RL2	RL2	RL2	RL2	VL2	VL2
AT. 7	Covert Channel Attacks [79]	2018	RL3	RL2	RL1	RL2	VL2	VL2
DF. 7	Covert Channel Defender [79]	2018	RL2	RL2	RL1	RL2	VL2	VL2
AT. 8	Link Flooding Attack [101]	2018	RL1	RL2	RL2	RL2	PL2	PL2
DF. 8	Link Flooding Defender [101]	2018	RL2	RL2	RL1	RL2	VL2	VL2
AT. 9	Crossfire Table-Overflow [108]	2019	RL1	RL2	RL1	RL2	VL2	VL1
DF. 9	Fire Guard [108]	2019	RL2	RL2	RL1	RL2	VL2	VL1
AT. 10	Topology Freezing & Reverse Loop [81]	2019	RL3	RL2	RL1	RL3	VL3	VL2
DF. 10	Cryptographic Key for MAC tag over DPID [81]	2019	RL2	RL2	RL1	RL3	VL3	VL2
AT. 11	Cross-path Attack [45]	2019	RL3	RL2	RL2	RL3	VL3	VL3
DF. 11	Reserving Bandwidth & Prioritizing Control Traffic [45]	2019	RL2	RL2	RL1	RL3	VL3	VL3
AT. 12	Fingerprinting Match Fields of Flow Rules [61]	2020	RL3	RL2	RL2	RL2	VL1	VL1
DF. 12	Postponing Flow Installation [61]	2020	RL2	RL2	RL1	RL2	VL1	VL1
AT. 13	Buffered Packet Hijacking [47]	2020	RL3	RL2	RL2	RL3	VL3	VL1
DF. 13	ConCheck [47]	2020	RL2	RL2	RL1	RL3	VL3	VL1
AT. 14	SYN Flood [85]	2020	RL3	RL2	RL1	RL2	VL2	VL1
DF. 14A	AEGIS [85]	2020	RL2	RL2	RL1	RL2	VL2	VL1
DF. 14B	SYNGuard [83]	2021	RL2	RL2	RL1	RL2	VL1	VL1
AT. 15A	Fingerprinting Critical Flow Rules [103]	2021	RL3	RL2	RL2	RL2	VL1	VL1
AT. 15B	Fingerprinting Network and Controller Type [103]	2021	RL3	RL2	RL2	RL2	VL1	VL1
DF. 15	Probabilistic Scrambling and Controller Dynamic Scheduling [103]	2021	RL2	RL2	RL2	RL2	VL1	VL1
AT. 16	Cross Path Attack [107]	2022	RL3	RL2	RL2	RL3	VL2	VL1
DF. 16	Cross Guard [107]	2022	RL2	RL2	RL1	RL3	VL2	VL1
AT. 17	Invisible Assailant Attack (IAA) [75]	2022	RL3	RL2	RL1	RL2	VL2	VL1
DF. 17	Route Path Verification (RPV) [75]	2022	RL2	RL2	RL1	RL1	VL2	VL1
AT. 18	DHCP DoS and Starvation [66]	2023	RL1	RL2	RL1	RL1	VL1	VL0
DF. 18	DHCP DoS and Starvation Mitigation [66]	2023	RL2	RL2	RL1	RL1	VL1	VL0
AT. 19	Flow Table Overflow [97]	2023	RL1	RL2	RL1	RL1	VL1	VL0
DF. 19	FTODefender [97]	2023	RL1	RL2	RL1	RL1	VL1	VL0
AT. 20	Marionette Attacks [49]	2024	RL1	RL2	RL1	RL3	VL3	VL3
DF. 20	Not Released yet.	2024	N/A	N/A	N/A	N/A	N/A	N/A

Note: RL* and VL* levels 0–3 correspond to scores 0.0, 0.25, 0.5, and 1.0, respectively. N/A → Not Available. N/A* → Not Applicable.