open access
Published online on: https://journal.uii.ac.id/jurnalsnati/
Check for updates
DOI: 10.20885/snati.v5.i1.45526
JURNAL SNATI Sains, Nalar, dan Aplikasi Teknologi Informasi
Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi
Vol. 5 No. 1 (2026) 45 - 53 ISSN Media Electronic: 2807-5935

Real-time Forensic Reconstruction of IPv6 NA Flood Attacks: A D4I Approach

Frendi Yusroni Romadhona[1], Ahmad Luthfi[2]
[1,2] Department of Informatics, Faculty of Industrial Technology, Universitas Islam Indonesia, Sleman, Indonesia
[1]frendi.romadhona@students.uii.ac.id, [2]ahmad.luthfi@uii.ac.id

Abstract

The global transition to IPv6 has introduced new attack surfaces within core network protocols, particularly the Neighbor Discovery Protocol (NDP). One of the most critical yet often overlooked threats is the Neighbor Advertisement (NA) Flood attack. Unlike conventional volumetric DDoS attacks aimed at saturating network bandwidth, NA Flood exploits the Stateless Address Autoconfiguration (SLAAC) mechanism to trigger resource exhaustion on target devices. Investigating such incidents presents unique forensic challenges, as attack traces in volatile memory are often lost when using traditional dead forensics methods. This study implements a real-time forensic investigation approach by integrating Live Forensics methods with the Digital Forensic Framework for Reviewing and Investigating Cyber Attack (D4I). This method is applied to acquire crucial volatile artifacts during the attack and reconstruct the modus operandi through Cyber Kill Chain (CKC) mapping and Chain of Artifacts (CoA) construction. Experimental results demonstrate that NA Flood attacks possess dangerous asymmetric characteristics: generating low network traffic (4.71 Mbps) while causing a CPU surge of up to 50% and a memory increase of 89.5 MB on the target server. The novelty of this study lies in the integration of Live Forensics with the D4I framework to acquire volatile data in real-time and systematically transform raw artifacts into a comprehensive forensic conclusion. This approach successfully reconstructs the 5W1H (Who, What, Where, When, Why, How) elements of the incident and visualizes the shift of the point of failure from the network infrastructure to the endpoint, offering a robust model for investigating protocol-based resource exhaustion attacks.

Keywords: IPv6; Neighbor Advertisement Flood; Live Forensics; D4I Framework; Forensic Investigation; Network Forensics; IPv6 security

1. Introduction

The global transition from Internet Protocol version 4 (IPv4) to Internet Protocol version 6 (IPv6) has accelerated significantly in recent years, driven by the exhaustion of the IPv4 address space and the exponential growth of the Internet of Things (IoT)[1]. Recent statistics indicate a substantial increase in IPv6 adoption, with Google reporting a global user adoption rate exceeding 48% as of late 2025 [2]. Similarly, infrastructure readiness in several regions, particularly in Asia and the Americas, has surpassed 50% [3]. This consistent upward trend, as illustrated in Figure 1, highlights the expanding attack surface within modern network infrastructures.

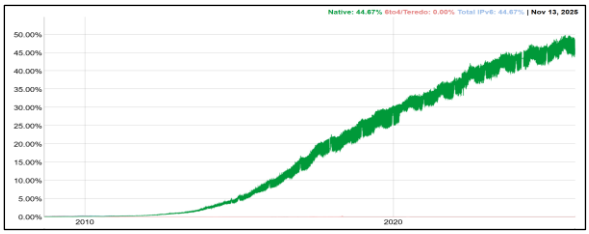

Figure 1. Global IPv6 Adoption Trend (2015–2025). Source: Google, 2025 [2].

Central to the operation of IPv6 is the Neighbor Discovery Protocol (NDP), defined in RFC 4861 [4], [5], which replaces the Address Resolution Protocol (ARP) of IPv4. NDP is critical for essential network functions, including router discovery, address resolution, and the Stateless Address Autoconfiguration (SLAAC) mechanism [6], [7].

Despite its enhanced features, the inherent design of NDP introduces new attack surfaces. A critical vulnerability lies in its lack of built-in authentication, where nodes implicitly trust Neighbor Advertisement (NA) messages received from the local link [8]. This trust model can be exploited to launch Neighbor Advertisement (NA) Flood attacks. Unlike conventional volumetric Distributed Denial of Service (DDoS) attacks that aim to saturate network bandwidth, NA Flood attacks are characteristically asymmetric; they generate relatively low traffic volumes but can trigger catastrophic resource exhaustion on target endpoints [9]. By flooding a

Received: 12-12-2025 | Revised: 07-01-2026 | Accepted: 08-01-2026 | Published: 31-01-2026

target with spoofed NA messages, an attacker forces the victim's operating system to continuously process and update its neighbor cache, leading to rapid CPU and memory depletion rather than network congestion [10].

Investigating such protocol-based attacks presents unique forensic challenges. Traditional "dead forensics" methodologies, which rely on post-mortem analysis of persistent storage, are largely ineffective for these incidents. The critical evidence of an NA Flood—such as the state of the neighbor cache, active network connections, and real-time memory spikes—resides in volatile memory. This evidence is inevitably lost when the system is powered down for imaging, a limitation exacerbated by the ephemeral nature of IPv6 temporary addresses [11]. Furthermore, without a structured analysis framework, correlating fragmented volatile artifacts to reconstruct the causality of the attack remains a complex task for forensic investigators [12].

Existing research in this domain has predominantly focused on two separate areas: intrusion detection and general live forensics. Several studies have developed flow-based Intrusion Detection Systems (IDS) to identify ICMPv6 anomalies [9], [13], while others have explored live forensic acquisition on IPv4 routers [11],[14]. However, there is a notable gap in the literature regarding the forensic reconstruction of IPv6-specific protocol attacks. Current methodologies often lack a structured framework to correlate volatile artifacts—such as neighbor cache states and CPU interrupts—into a coherent attack narrative.

Therefore, this study aims to bridge this gap by making the following main contributions: 1. It proposes an integrated forensic workflow combining Live Forensics with the D4I Framework specifically for IPv6 environments. 2. It provides empirical evidence of the asymmetric nature of NA Flood attacks, distinguishing them from volumetric DDoS. 3. It demonstrates the utility of the Chain of Artifacts (CoA) in visualizing the shift of failure from network infrastructure to endpoint resources.

To address these limitations, this research proposes a real-time forensic investigation approach that integrates Live Forensics with the Digital Forensic Framework for Reviewing and Investigating Cyber Attack (D4I). Live Forensics is employed to acquire volatile artifacts from the router and server memory during the ongoing attack, ensuring the preservation of transient evidence in accordance with the Order of Volatility [15]. Subsequently, the D4I framework is utilized to systematically map these artifacts to the Cyber Kill Chain (CKC) and construct a Chain of Artifacts (CoA). This integrated approach allows for the visualization of the attack's modus operandi and provides empirical evidence of the shift in the point of failure from the network infrastructure to the endpoint [12].

2. Related Work

The domain of IPv6 security and digital forensics has garnered significant academic attention in recent years. This section reviews existing literature categorized into three primary domains: IPv6 intrusion detection, live forensic methodologies for DDoS, and digital forensic frameworks. A summary comparison of these works with the proposed approach is presented in Table 1.

Table 1. Comparison of Existing Works vs. Proposed Approach

Reference	Focus Area	Method	Key Limitation (Gap)
[8]	DDoS Detection on IPv6	Review	Focuses on theoretical review; No specific forensic reconstruction.
[9]	Intrusion Detection on IPv6	Flow-based IDS	Automated detection only; lacks post-incident forensic analysis.
[11]	Live Forensics on IPv4	Live Acquisition	Tested on IPv4; lacks structured analysis.
[14]	Router Forensics on IPv4	Network Forensics	General framework; does not visualize attack causality (CoA).
Proposed Method	IPv6 Protocol Forensics	Live Forensics	Real-time acquisition & visualization of NA Flood impact.

2.1. IPv6 Security and Intrusion Detection

The inherent vulnerabilities of the Neighbor Discovery Protocol (NDP) have been extensively studied. Research by [8] provided a comprehensive review of ICMPv6-based DoS attacks, highlighting that the lack of authentication in NDP messages (RS, RA, NS, NA) remains a critical security flaw. To address this, [9] proposed a flow-based Intrusion Detection System (IDS) that enriches flow features to detect ICMPv6 flooding attacks with high accuracy. Similarly, [13] developed a high-performance IDS specifically for NDP attacks using machine learning techniques. While these studies contribute significantly to detection and prevention, they primarily focus on automated alerting systems. They do not address the post-detection phase, specifically the forensic reconstruction of the attack's modus operandi or the detailed analysis of its impact on endpoint resources.

2.2. Live Forensics in DDoS Investigations

Given the volatile nature of DDoS attacks, traditional dead forensics has proven insufficient. [11] demonstrated the efficacy of Live Forensics in acquiring evidence from a MikroTik Routerboard during a DoS attack. Their research confirmed that acquiring volatile data—such as CPU load and active connections—in real-time is essential for proving the occurrence of an attack. [14] further expanded this by conducting router forensic analysis against Distributed Denial of Service attacks, emphasizing the need for

capturing network traffic logs during the incident. However, these studies were largely conducted within IPv4 environments. Furthermore, while they successfully acquired evidence, they lacked a structured analysis framework to systematically map the evidence to the attacker's lifecycle, often resulting in fragmented findings.

2.3. Digital Forensic Frameworks (D4I)

To bridge the gap between data acquisition and analysis, various frameworks have been proposed. [12] introduced the D4I Framework (Digital Forensic Framework for Reviewing and Investigating Cyber Attack), which integrates the Cyber Kill Chain (CKC) into the forensic examination phase. D4I was designed to be attack-agnostic and has been successfully applied to investigate application-layer attacks, such as social engineering and phishing scams, as demonstrated by [16]. Nevertheless, the application of D4I in investigating low-level network protocol attacks, specifically IPv6 Neighbor Advertisement Flooding, remains unexplored. Consequently, its effectiveness in visualizing the causal relationship between protocol exploitation and resource exhaustion has not yet been empirically validated.

2.4. Research Gap and Contribution

The review of related work indicates a clear research gap. While methods for detecting IPv6 attacks exist, and Live Forensics has been applied to IPv4 DDoS, there is a lack of research that:

1. Conducts a forensic investigation specifically targeting IPv6 Neighbor Advertisement (NA) Flood attacks.
2. Integrates Live Forensics with the D4I Framework to reconstruct the attack lifecycle.
3. Empirically proves the asymmetric impact (resource exhaustion vs. bandwidth saturation) of such attacks on server endpoints.

This paper addresses these gaps by proposing an integrated forensic approach to visualize and analyze the shift in the point of failure from the network infrastructure to the endpoint during an NA Flood attack.

3. Research Methods

This study adopts an experimental research design combining Live Forensics for real-time evidence acquisition and the D4I Framework for structured analysis. The methodology is divided into four stages: experimental environment setup, attack simulation, forensic data acquisition, and D4I-based analysis. The comprehensive research framework illustrating the workflow and logical progression of these phases is presented in Figure 2.

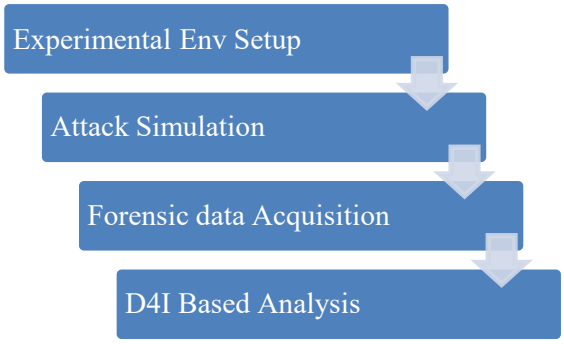

Figure 2. Research Framework

3.1. Experimental Environment and Topology

To simulate a realistic IPv6 network environment, a virtualized testbed was deployed using PNETLab running on a VMware ESXi host. The specifications of the devices used are detailed in Table 2.

Table 2. Experimental Device Specifications

Device Role	OS / Firmware	Tools Installed	Function
Attacker	Kali Linux 2024.04	thc-ipv6, fping, nping	Packet injection & Traffic generation
Router	Mikrotik CHR 7.18.2	SNMP, Traffic Monitor	IPv6 Routing & Infrastructure target
Victim Server	Ubuntu Server 22.04	tcpdump, mpstat, vmstat	Endpoint target
Investigator	Ubuntu Desktop 21.04	Wireshark, Browser, SFTP Client	Forensic data collection & analysis

The network topology, as illustrated in Figure 3, was designed to replicate a realistic production environment comprising three distinct zones: the External Attack Zone, the Network Infrastructure and the Internal Victim Zone.

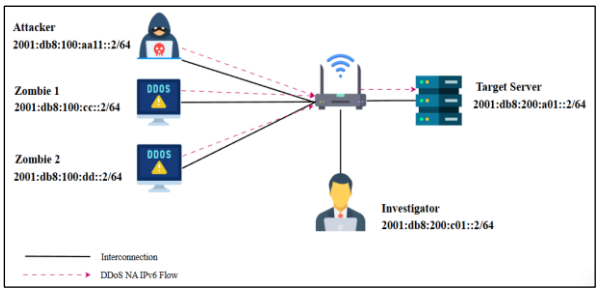

Figure 3. Experimental Network Topology.

The setup includes an external attacker coordinating distributed zombies to flood the target server, while the investigator node performs real-time acquisition via a mirror port. The topology consists of the following components:

1. Attacker Node (2001:db8:100:aa11::2): Acts as the Command and Control (C2) center, initiating the attack commands.
2. Zombie Nodes: Two distributed nodes (Zombie 1 and 2) are utilized to amplify the attack traffic, simulating a Distributed Denial of Service (DDoS) scenario.
3. Infrastructure (Router): A Mikrotik CHR router serves as the IPv6 gateway (2001:db8:200::48), handling routing and advertisement messages.
4. Victim Server (2001:db8:200:a01::2): The target endpoint hosting web services, monitored for resource exhaustion.
5. Investigator Node: A dedicated workstation connected to the core switch to capture traffic and retrieve volatile logs from the victim server securely.

3.2. Attack Scenario: Neighbor Advertisement Flood

The study focuses specifically on the Neighbor Advertisement (NA) Flood attack. This vector exploits the trust model inherent in the IPv6 Neighbor Discovery Protocol (NDP), specifically within the Stateless Address Autoconfiguration (SLAAC) mechanism described in RFC 4861 [4].

In a standard operation, a node updates its Neighbor Cache upon receiving a legitimate Neighbor Advertisement. However, as illustrated in Figure 4, in an NA Flood attack, the perpetrator injects a high volume of spoofed NA packets. Each packet carries the Override flag and a randomized Link-Layer Address (MAC), claiming to be the legitimate gateway or neighbor.

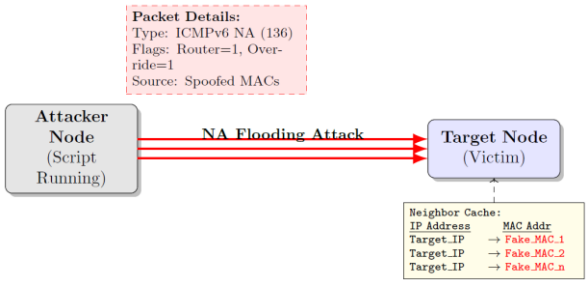

Figure 4. Logic of Neighbor Advertisement Flood Attack.

Algorithm 1 Neighbor Advertisement Flooding Logic

Input: Target_IPv6 (V), Network_Interface (I)
Parameter: Flood_Rate (Max), Randomize_Source ($True$)
Initialize: Raw_Socket on I
$Target\_MAC \leftarrow Resolve(V)$
while $attack\_active$ **do**
  $Rand\_IPv6 \leftarrow GenerateRandomIPv6()$
  $Rand\_MAC \leftarrow GenerateRandomMAC()$
  Construct ICMPv6_Packet:
  $Type \leftarrow 136$ (Neighbor Advertisement)
  $Code \leftarrow 0$
  $Flags \leftarrow (Router = 1, Solicited = 1, Override = 1)$
  $Target\_Address \leftarrow V$
  $Option \leftarrow Target\_Link\_Layer\_Address(Rand\_MAC)$
  Send Packet to V
end while

The attack execution logic is formalized in Algorithm 1. The tool generates randomized source addresses and floods the target with NA messages containing the Override flag set to 1. This forces the victim's kernel to perpetually update its neighbor cache entries, consuming significant CPU cycles for state management rather than data processing.

To ensure the reproducibility of this experiment, the specific parameters configured for the NA Flood attack using the the-ipv6 toolkit are detailed in Table 3. These parameters were chosen to simulate a high-intensity protocol attack aimed at exhausting the target's neighbor cache resources

Table 3. Experimental Attack Parameters

Parameter	Configuration / Value	Description
Attack Tool	atk6-flood_advertise6	Part of *the-ipv6* security toolkit
Protocol Vector	ICMPv6 Type 136	Neighbor Advertisement (NA)
Injection Flags	Override (O)=1, Router (R)=1	Forces cache entry updates on the victim
Source Address	Randomized	Random IPv6 and Link-Layer (MAC) addresses
Attack Duration	480 Seconds	00:02 – 00:10 WIB (8 Minutes)
Packet Volume	127,048 Packets	Total malicious packets injected
Avg. Packet Rate	~265 pps	Calculated based on total volume/duration
Target	2001:db8:200:a01::2	Direct injection to the network server

The attack was executed using the thc-ipv6 toolkit on 11 May 2025, 00:02 - 00:10 WIB from the Attacker node. The specific command utilized to initiate the flood is:

```
atk6-flood_advertise6 eth0 2001:db8:200:a01::2
```

This command generates a high volume of fake NA packets claiming ownership of random IPv6 addresses within the subnet, forcing the victim server (2001:db8:200:a01::2) to continuously update its neighbor cache table, thereby consuming system resources.

3.3. Live Forensic Acquisition

Given the volatile nature of the attack impact — specifically the rapid spikes in CPU load and memory consumption targeted by the Neighbor Advertisement flooding— a Live Forensics approach was mandated. Data acquisition was performed in real-time during the attack execution, strictly adhering to the Order of Volatility as outlined in RFC 3227.

A. Volatile Data Dumping

To capture the transient state of the victim server, a custom acquisition script (start_monitoring.sh) was executed to dump critical volatile artifacts into a secure directory. This automated process generated

time-stamped log files for CPU usage (mpstat), memory paging (vmstat), bandwidth usage (iftop) and a full packet capture (tcpdump) of the network interface. The acquisition script is presented in Listing 1.

Listing 1. Automated Live Forensic Acquisition Script (start_monitoring.sh)

```
#!/bin/bash
# Filename: start_monitoring.sh

# Start packet capture
tcpdump -i ens3 -w "start_traffic_$(date
+%Y%m%d_%H%M%S).pcap" &
 # Start CPU monitoring
mpstat 1 300 > "start_cpu_usage_$(date
+%Y%m%d_%H%M%S).log" 2>&1 &
 # Start memory monitoring
vmstat 1 300 > "start_memory_usage_$(date
+%Y%m%d_%H%M%S).log" 2>&1 &
 # Start bandwidth monitoring
sudo iftop -i ens3 -t -s 300 >
"start_bandwidth_iftop_$(date
+%Y%m%d_%H%M%S).log" 2>&1 &

echo "Monitoring started. Use 'pkill tcpdump
mpstat vmstat iftop' to stop."
```

These scripts were configured to capture critical artifacts at 1-second intervals. Figure. 5 displays the successful generation of these raw artifact files immediately following the attack simulation.

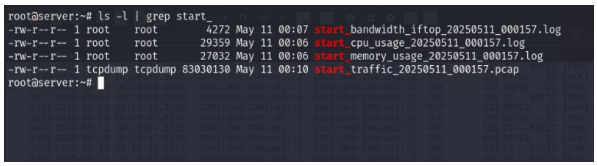

Figure 5. Directory listing showing the acquired volatile artifacts (CPU logs, Memory logs, and PCAP) generated by the live acquisition script.

The acquisition parameters and tools utilized are detailed in Table 4.

Table 4. Summary of Acquired Artifacts and Tools

Artifact Category	Target Metric	Tool / Command	Objective
Network Traffic	Full Packet Data	tcpdump	To analyze ICMPv6 headers, payload integrity and attack volume.
CPU Resources	Processor Load	mpstat	To detect process exhaustion caused by interrupt handling.
Memory State	RAM Usage	vmstat	To observe anomalies indicative of Neighbor Cache table exhaustion.
Throughput	Bandwidth (bps)	iftop / RouterOS	To measure the saturation levels of the network interface.

B. Data Preservation and Hashing

To ensure the integrity and admissibility of the acquired evidence, a cryptographic hashing process was performed immediately after acquisition. A custom script (forensic-prop.bash) was utilized to calculate the MD5 and SHA-256 hash values for each artifact file. This step is critical to guarantee that the evidence has not been altered during the transfer or analysis phases. . The hashing script is presented in Listing 2.

Listing 2. Cryptographic Hashing Script (forensic-prop.bash)

```
#!/bin/bash
# ------
        if [ "$#" -ne 1 ]; then
            echo "Usage: $0 <file_path>"
            exit 1
        fi
        FILE="$1"
        OUTPUT="${FILE}_forensic_properties.txt"

# ------
        if [ ! -f "$FILE" ]; then
            echo "Error: File $FILE not found!"
            exit 1
        fi
# ------
        FILE_NAME=$(basename "$FILE")
        DATE_CREATED=$(stat   -c   %w   "$FILE"
        2>/dev/null || date -r "$FILE" "+%Y-%m-
        %d %H:%M:%S")
# ------
        MD5=$(md5sum "$FILE" | awk '{print $1}')
        SHA1=$(sha1sum  "$FILE"  |  awk  '{print
        $1}')
        SHA256=$(sha256sum "$FILE" | awk '{print
        $1}')
        SHA512=$(sha512sum "$FILE" | awk '{print
        $1}')
 # ------
        echo  "===  FORENSIC  PROPERTIES  ==="  >
        "$OUTPUT"
        echo   "File   Name:   $FILE_NAME"   >>
        "$OUTPUT"
        echo  "Created  Date:  $DATE_CREATED"  >>
        "$OUTPUT"
        echo "MD5: $MD5" >> "$OUTPUT"
        echo "SHA-1: $SHA1" >> "$OUTPUT"
        echo "SHA-256: $SHA256" >> "$OUTPUT"
        echo "SHA-512: $SHA512" >> "$OUTPUT"

        echo  "Forensic  properties  saved  to:
```

Figure. 6 illustrates the preservation output, documenting the unique hash signature for the bandwidth log file.

Figure 6. Forensic properties file displaying the generated MD5 and SHA-256 hash values, ensuring the chain of custody for the acquired evidence.

Following the hashing process, the artifacts were securely transferred from the victim server to the investigator's workstation using SFTP (Secure File

Transfer Protocol) to maintain data confidentiality and integrity during transit.

3.4. D4I Analysis Framework

An analysis of the acquired raw data was conducted using the D4I Framework [12] chosen for its robust capability in reconstructing attack scenarios from heterogeneous log sources. The analytical procedure was executed in two sequential phases:

1. Phase 1: Cyber Kill Chain (CKC) Mapping In this initial phase, identified artifacts were categorized and aligned with the seven stages of the Cyber Kill Chain adapted for DDoS scenarios. This mapping provided a structured taxonomy of the attack lifecycle, covering Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control (C2), and Actions on Objectives.
2. Phase 2: Chain of Artifacts (CoA) Construction The second phase involved the recursive application of the D4I six-step workflow: Choose, Identify, Correlate, Construct CoA, Repeat, and Analyze. This iterative process aimed to visually reconstruct the attack path and pinpoint the root cause of service failure.

By synchronizing the timestamps of network packet influx with system resource spikes, we established a definitive causality between the attack traffic and the subsequent exhaustion of router resources. The comprehensive investigation workflow of the D4I framework, is illustrated in Fig. 7.

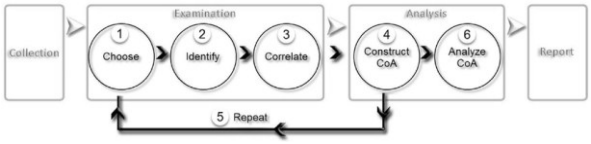

Figure 7. The D4I Investigation Process [12].

4. Results and Discussions

This section presents the empirical findings from the real-time forensic investigation of the IPv6 Neighbor Advertisement (NA) Flood attack. The analysis focuses on two key dimensions: (1) the quantitative impact on system resources to demonstrate the attack's asymmetric nature, and (2) the qualitative reconstruction of the attack lifecycle using the D4I framework.

4.1. Asymmetric Resource Exhaustion Analysis

The experimental results revealed a significant anomaly in the correlation between network traffic volume and endpoint resource consumption. During the attack execution window (00:02 – 00:10 WIB), the monitoring systems captured distinct behavioral patterns that differentiate this protocol-based attack from traditional volumetric DDoS.

4.1.1. Network Traffic vs System Impact

Contrary to typical flooding attacks designed to saturate link bandwidth, the NA Flood generated a relatively low average traffic rate. As recorded in the forensic logs, the throughput remained stable with a peak download rate of approximately 4.71 Mbps and negligible upload traffic. Under normal circumstances, such traffic volume is well within the capacity of modern network infrastructure and would likely evade threshold-based detection systems configured for volumetric attacks.

However, despite the minimal bandwidth footprint, the impact on the Victim Server's computational resources was disproportionately severe. As detailed in the acquired forensic artifacts:

- CPU Utilization: The server's CPU usage surged dramatically from a baseline of ~2% to a peak of 50.24% during the attack. This indicates that the server's processing power was diverted to handling the influx of protocol packets rather than serving legitimate requests.
- Memory Consumption: A sharp, continuous increase in memory usage was observed. The server's free memory decreased by 89.5 MB within the 8-minute attack duration, signaling rapid resource exhaustion due to neighbor cache table instability.

Fig. 8 highlights that while the network throughput remained negligible (< 5 Mbps), the server's CPU load surged to over 50%, confirming the resource exhaustion characteristic of the attack.

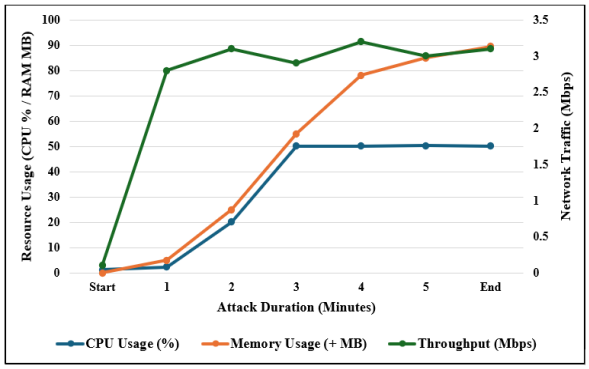

Figure 8. Temporal Analysis of Victim Server Performance Metrics under NA Flood Attack Conditions.

4.1.2. Root Cause of Exhaustion

This resource exhaustion is attributed to the operating system's handling of the Neighbor Discovery Protocol (NDP). The forensic analysis of the PCAP files on 11 May 2025, 00:02 - 00:10 WIB revealed an influx of 127,048 spoofed Neighbor Advertisement (NA)

packets. Unlike standard traffic, these packets carried the Override flag set to 1, combined with randomized Link-Layer addresses.

To visualize this interaction, Figure 9 illustrates the mechanism of the attack impact on the victim's internal state. The diagram depicts how the stream of spoofed NA packets forces the victim's kernel to perpetually overwrite existing entries in its Neighbor Cache table. This continuous validation and state-update process prevents the CPU from entering an idle state, converting a low-bandwidth network stream into a high-load computational task. This finding validates that the attack vector successfully shifted the point of failure from the network infrastructure to the endpoint.

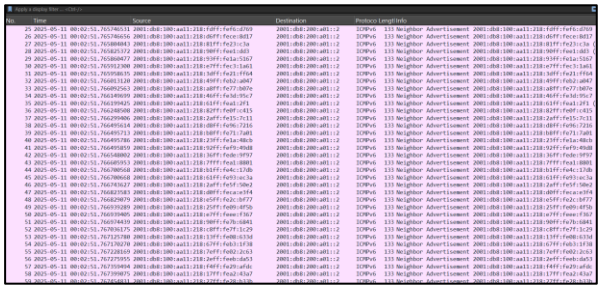

Figure 9. Mechanism of the Neighbor Advertisement (NA) Flood Attack.

4.2. Forensic Analysis & Reconstruction using D4I

To provide a structured interpretation of the acquired artifacts and attribute the root cause of the incident, the D4I framework was applied. The investigation followed the framework's two-pillared approach: categorizing artifacts into the *Cyber Kill Chain* (CKC) and reconstructing the attack lifecycle through the *Chain of Artifacts* (CoA).

4.2.1. Cyber Kill Chain (CKC) Mapping

The raw artifacts acquired from the Live Forensics phase were filtered and mapped to the seven phases of the Cyber Kill Chain, adapted for IPv6 protocol attacks. This mapping transforms isolated log entries into a coherent attack narrative. Table 5 summarizes the CKC mapping specific to the Neighbor Advertisement (NA) Flood scenario.

Table 5. Cyber Kill Chain (CKC) Mapping for Neighbor Advertisement (NA) Flood Attack

CKC Phase	Artifact Category	Identified Evidence
Reconnaissance (R)	Network Log	Scanning activity detected targeting the victim's IPv6 address (2001:db8:200:a01::2) prior to the traffic spike.
Weaponization (W)	Tool Analysis	Identification of the thc-ipv6 attack suite execution, specifically the fake_advertise6 tool capable of generating spoofed NA packets.
Delivery (D)	Packet Capture	On 11 May 2025, 00:02 - 00:10 WIB Influx of 127,048 ICMPv6 Type 136 (Neighbor Advertisement) packets originating from the attacker's subnet (2001:db8:100:aa11::/64).
Exploitation (E)	Resource Metric	Critical Spike: Server CPU usage reached 50% and Memory consumption increased by 89.5 MB, indicating active processing of the spoofed packets by the OS kernel.
Installation (I)	-	No persistent malware installation artifacts were found, confirming the attack nature as a protocol flood rather than an intrusion.
Command & Control (C2)	-	No external Command and Control communication was established post-exploitation.
Actions (A)	Service Impact	Service unresponsiveness verified by a drop in valid HTTP traffic throughput and connection timeouts.

4.2.2. Chain of Artifacts (CoA) Analysis

The second phase involved constructing the Chain of Artifacts (CoA) to visualize the causal relationship between the vector (Delivery) and the impact (Exploitation). By correlating the timestamps of the packet capture (.pcap) with the system performance logs (mpstat, vmstat), the attack path was reconstructed as illustrated in Figure 10.

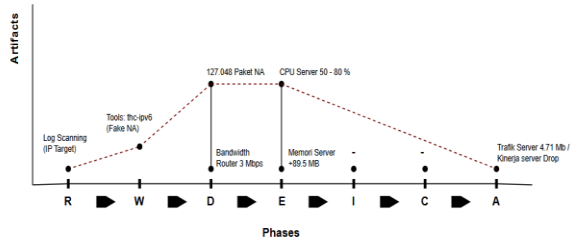

Figure 10. Chain of Artifacts (CoA) Visualization for NA Flood Attack

The CoA visualization explicitly demonstrates a shift in the point of failure, distinguishing this attack from standard volumetric DDoS:

a. Delivery Phase: The chain initiates with the delivery of spoofed NA packets on 11 May 2025, 00:02 - 00:10 WIB. Unlike volumetric attacks where the bottleneck typically appears at the router (network bandwidth saturation), the bandwidth usage at this stage remained low (~3 Mbps).
b. Exploitation Phase (The Shift): The attack signature line bypasses the infrastructure

bottleneck and connects directly to the Server's CPU and Memory artifacts. The vertical correlation in this phase proves that the 127,048 NA packets triggered an intensive Neighbor Cache update process in the victim's operating system, leading to Resource Exhaustion.

c. Actions Phase: The final node in the chain confirms that the denial of service was a direct result of the server's inability to allocate computational resources for legitimate requests, not due to a clogged network pipe.

4.2.3. Investigative Report Summary

To synthesize the forensic findings into actionable intelligence, the investigation results are consolidated using the standard 5W1H (Who, What, Where, When, Why, How) investigative model. The inquiry confirmed that the service disruption was caused by a Neighbor Advertisement (NA) Flood (What) executed during the specific window of 11 May 2025, 00:02–00:10 WIB (When). Forensic analysis attributed the source to the host 2001:db8:100:aa11::2, ..cc::2, ..dd::2 (Who), which utilized the atk6-flood_advertise6 tool to inject spoofed packets with the 'Override' flag (How). This exploitation of the SLAAC mechanism (Why) resulted in a universal resource exhaustion affecting both the network gateway and the target endpoint (Where). A definitive and structured reconstruction of these findings is presented in Table 6

Table 6. Forensic Investigation (5W1H)

Element	Forensic Definition	Investigation Findings
WHAT	Incident Description	A Neighbor Advertisement (NA) Flood attack was confirmed. The incident involved the injection of spoofed ICMPv6 packets, leading to critical resource exhaustion.
WHO	Technical Attribution	The attack originated from the host 2001:db8:100:aa11::2 (Attacker/C2) coordinating with Zombies (...cc::2, ...dd::2) targeting the victim server 2001:db8:200:a01::2.
WHERE	Locus of Impact	Universal Resource Exhaustion affecting both the Router (High IRQ/~100% CPU) and the Victim Server (~50% CPU).
WHEN	Timeline	The incident occurred on 11 May 2025, with the active attack window recorded between 00:02 and 00:10 WIB.
WHY	Root Cause	SLAAC vulnerability exploitation. The flood of 'Override' flags triggered a CPU Interrupt Storm (IRQ) and excessive Context Switching, overwhelming single-core processing capabilities.
HOW	Modus Operandi	The attacker utilized the atk6-flood_advertise6 tool from the the-ipv6 suite to flood 127,048 spoofed NA packets with the Override flag, forcing the victim's kernel to perpetually overwrite cache entries.

This reconstruction validates the efficacy of the proposed framework in converting raw volatile data into a complete forensic narrative, satisfying the reporting requirements of the D4I model

4.3. Discussion

The experimental results demonstrate that IPv6 Neighbor Advertisement (NA) Flooding functions primarily as a Protocol Exploitation attack rather than a volumetric denial of service. The vulnerability lies within the Stateless Address Autoconfiguration (SLAAC) mechanism, which inherently trusts NA messages to map IPv6 addresses to Link-Layer addresses (RFC 4861). By flooding the network with spoofed claims carrying the Override flag, the attacker forces the victim node to perform computationally expensive verification and cache update operations. This confirms that the service failure was caused by endpoint resource exhaustion (CPU and Memory) rather than network bandwidth saturation.

Critically, the investigation highlights the necessity of Live Forensics in IPv6 incident response. Traditional dead forensics would have failed to capture the momentary memory consumption spikes and the active poisoning of the neighbor cache, leading to an inconclusive investigation. The D4I Framework successfully contextualized these volatile artifacts, structuring the raw data into a legally admissible narrative. This distinction is vital for network administrators, as it shifts the required mitigation strategy from upstream traffic filtering to endpoint-level protocol hardening (e.g., enabling SEND or RA Guard).

The completeness of this reconstruction highlights the critical synergy between the acquisition and analysis methodologies employed. The Live Forensics approach was instrumental in capturing volatile evidence —specifically the active neighbor cache states and real-time memory spikes— that would have been irretrievable via post-mortem analysis. Concurrently, the D4I Framework provided the necessary structural rigor to correlate these artifacts. By bridging the gap between real-time data acquisition and structured analysis, this approach facilitates a comprehensive conclusion capable of answering the 5W1H questions definitively.

5. Conclusion

This study presented a comprehensive forensic reconstruction of IPv6 Neighbor Advertisement (NA) Flood attacks by integrating Live Forensics techniques with the D4I framework. Based on experimental analysis, three primary conclusions are drawn. First, the NA Flood operates as an asymmetric protocol attack. Unlike traditional volumetric DDoS, this attack generates negligible traffic (approximately 4.7 Mbps) yet induces critical resource exhaustion on the target

endpoint. The vulnerability lies within the Stateless Address Autoconfiguration (SLAAC) mechanism, where the victim's CPU and memory are consumed by processing spoofed cache entries rather than by network bandwidth saturation.

Second, the investigation confirms that Live Forensics is indispensable for IPv6 incidents. Traditional post-mortem (dead) forensics failed to detect the attack artifacts because the primary evidence—active neighbor cache poisoning and transient memory spikes—is volatile and vanishes upon system reboot. Third, the systematic application of the D4I Framework proved critical in structuring the forensic analysis. By correlating disparate artifacts into a cohesive narrative, this framework enabled the reconstruction of a comprehensive case conclusion that satisfies the 5W1H (Who, What, Where, When, Why, How) investigative standard.

Looking forward, several opportunities for future development remain. Subsequent research should focus on developing Machine Learning (ML) models capable of automating the detection of NA Flooding patterns by training on the artifact features identified in this study, such as CPU spikes correlated with ICMPv6 Override flags. Additionally, active mitigation strategies, including the implementation of RA-Guard and SEND (Secure Neighbor Discovery), should be evaluated in diverse network environments. Finally, the scope of investigation should be expanded to include other IPv6 Extension Headers, such as Routing Headers and Hop-by-Hop Options, which may present similar asymmetric attack surfaces.

Reference

[1] K. Nikolina, "Overview of the progress of IPv6 adoption in Croatia," in 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia: IEEE, May 2022, pp. 405–408. doi: 10.23919/MIPRO55190.2022.9803479.

[2] Google, "IPv6 Adoption Statistics," May 2025, [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html

[3] G. Huston, "IPv6 Capability Metrics: World," May 2025, [Online]. Available: https://stats.labs.apnic.net/ipv6/XA

[4] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC Editor, RFC4861, Sept. 2007. doi: 10.17487/rfc4861.

[5] S. Praptodiyono, Moh. Jauhari, R. Fahrizal, I. H. Hasbullah, A. Osman, and S. Ul Rehman, "Integration of Firewall and IDS on Securing Mobile IPv6," in 2020 2nd International Conference on Industrial Electrical and Electronics (ICIEE), Lombok, Indonesia: IEEE, Oct. 2020, pp. 163–168. doi: 10.1109/ICIEE49813.2020.9277354.

[6] W. Hui, Y. Sun, J. Liu, and K. Lu, "DDoS/DoS Attacks and Safety Analysis of IPv6 Campus Network: Security Research under IPv6 Campus Network," in 2011 International Conference on Internet Technology and Applications, Wuhan, China: IEEE, Aug. 2011, pp. 1–4. doi: 10.1109/ITAP.2011.6006421.

[7] S. Manickam et al., "Labelled Dataset on Distributed Denial-of-Service (DDoS) Attacks Based on Internet Control Message Protocol Version 6 (ICMPv6)," Wirel. Commun. Mob. Comput., vol. 2022, pp. 1–13, Apr. 2022, doi: 10.1155/2022/8060333.

[8] M. Tayyab, B. Belaton, and M. Anbar, "ICMPv6-Based DoS and DDoS Attacks Detection Using Machine Learning Techniques, Open Challenges, and Blockchain Applicability: A Review," IEEE Access, vol. 8, pp. 170529–170547, 2020, doi: 10.1109/ACCESS.2020.3022963.

[9] O. E. Elejla, M. Anbar, S. Hamouda, B. Belaton, T. A. Al-Amiedy, and I. H. Hasbullah, "Flow-Based IDS Features Enrichment for ICMPv6-DDoS Attacks Detection," Symmetry, vol. 14, no. 12, p. 2556, 2022, doi: 10.3390/sym14122556.

[10] R. Sood and P. Lim, "CYBER FORENSIC MANUAL FOR DENIAL-OF- SERVICE ATTACK." ResearchGate, Dec. 11, 2023. [Online]. Available: https://www.researchgate.net/publication/376692478

[11] M. Alim, I. Riadi, and Y. Prayudi, "Live Forensics Method for Analysis Denial of Service (DOS) Attack on Routerboard," Int. J. Comput. Appl., vol. 180, no. 35, pp. 23–30, Apr. 2018, doi: 10.5120/ijca2018916879.

[12] A. Dimitriadis, N. Ivezic, B. Kulvatunyou, and I. Mavridis, "D4I - Digital forensics framework for reviewing and investigating cyber attacks," Array, vol. 5, p. 100015, Mar. 2020, doi: 10.1016/j.array.2019.100015.

[13] A. Alzaqebah, I. Aljarah, and O. Al-Kadi, "A hierarchical intrusion detection system based on extreme learning machine and nature-inspired optimization," Comput. Secur., vol. 124, p. 102957, Jan. 2023, doi: 10.1016/j.cose.2022.102957.

[14] O. R. Prayogo and I. Riadi, "Router Forensic Analysis against Distributed Denial of Service (DDoS) Attacks," Int. J. Comput. Appl., vol. 175, no. 39, pp. 19–25, 2020, doi: 10.5120/ijca2020920944.

[15] D. Brezinski and T. Killalea, "Guidelines for Evidence Collection and Archiving," RFC Editor, RFC3227, Feb. 2002. doi: 10.17487/rfc3227.

[16] R. Nurdin, "Investigasi Forensika Digital WhatsApp Scam Dengan Menggunakan Framework D4I," JATISI J. Tek. Inform. Dan Sist. Inf., vol. 11, no. 1, pp. 158–166, 2024, doi: 10.35957/jatisi.v11i1.6616.