IPv6 Router Advertisement (RA) Flooding Attack Design Report

Group ID-05

Student ID - 2005102,2005118

July 11, 2025

Contents

Project Overview

This project focuses on designing and detailing an IPv6 Router Advertisement (RA) flooding attack tool. The tool will be developed from scratch using raw sockets, specifically avoiding any existing third-party packet crafting libraries or tools. The primary goal is to demonstrate a Denial-of-Service (DoS) attack by overwhelming target hosts and the local network segment with an excessive volume of crafted IPv6 RA messages.

1 Definition of the Attack with Topology Diagram

1.1 Definition of the Attack

IPv6 Router Advertisement (RA) Flooding is a Layer 2/Layer 3 denial-of-service (DoS) attack targeting the Neighbor Discovery Protocol (NDP). In this attack, a malicious actor rapidly sends an overwhelming number of ICMPv6 Router Advertisement messages on a local network segment. These messages are typically unsolicited and can contain legitimate-looking or outright malicious configuration information (e.g., rogue prefixes, default gateways, or DNS servers).

The primary objectives of RA flooding are:

1. **Resource Exhaustion:** Overload target host CPUs, memory, and network stacks as they attempt to process and store information from the flood of RAs, leading to system slowdowns or crashes.

2. **Network Congestion:** Consume significant network bandwidth with the sheer volume of RA traffic, degrading performance for legitimate communications.

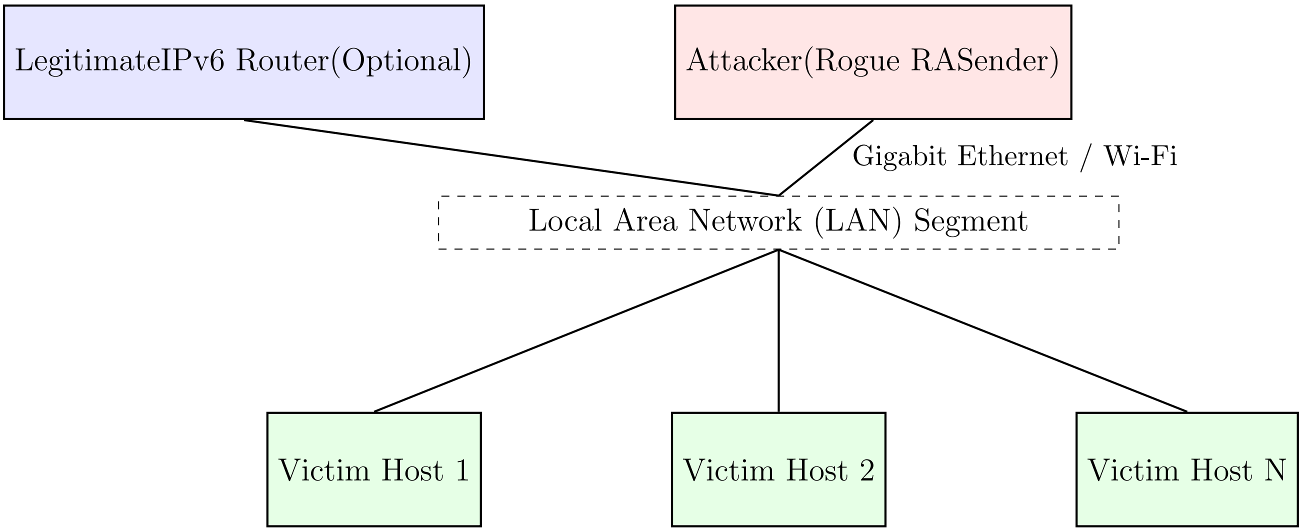
3. **Configuration Instability:** Cause hosts to constantly re-evaluate and reconfigure their IPv6 addresses, default gateways, and other network parameters, leading to intermittent connectivity or complete network disruption.

4. **Information Overload:** Potentially flood host Neighbor Caches (equivalent to ARP caches in IPv4) and routing tables with excessive or conflicting entries.

While the core focus here is pure DoS through flooding, the crafted RAs can optionally include malicious data to also facilitate Man-in-the-Middle (MITM) scenarios (e.g., advertising the attacker as the preferred default router or providing malicious DNS server addresses), further compounding the attack's impact.

1.2 Topology Diagram

The attack operates on a local broadcast domain (Layer 2 segment).



Description:

- **Attacker:** Machine running the custom RA flooding tool.
- **Victim Hosts:** Target machines on the same LAN segment, running IPv6.
- **Legitimate IPv6 Router (Optional):** A standard IPv6 router that might be providing legitimate RAs.
- **LAN Segment:** Shared physical or logical network (e.g., Ethernet Switch or Wi-Fi AP) where all devices communicate.

Figure 1: IPv6 RA Flooding Attack Topology

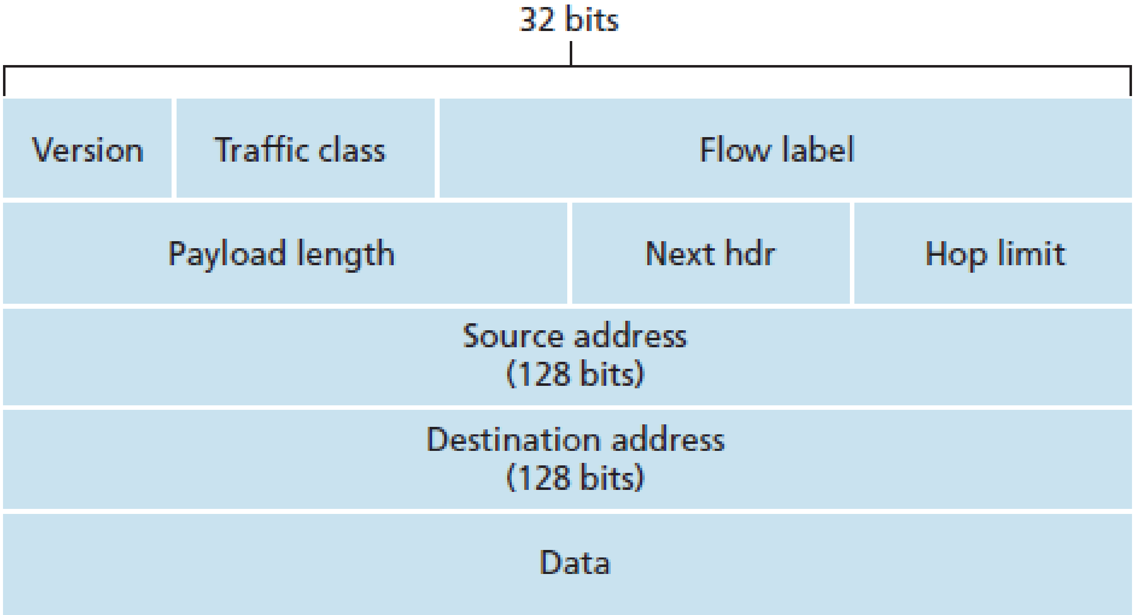


Figure 2: IPv6 Header

2 Timing Diagram of the Original Protocol and Your Attack Timing Diagram

2.1 Original Protocol Timing Diagram (Normal IPv6 RA Operation)

In a typical IPv6 network, Router Advertisements are part of the Neighbor Discovery Protocol (NDP).

```
Time ------------------------------------------------------------------->

Legitimate Router:   RA (Unsolicited)  --------------> RA (Unsolicited) ...
                      (Multicast)  (every 200-600 seconds)  (Multicast)

Host A:              RS (initial startup) -->
                      (Multicast)       |
                                        |
Host A Processing:   Receive RA, configure SLAAC address, default route,
                      DNS (if RDNSS)
                                        |
Host B:              RS (initial startup) -->
                      (Multicast)       |
                                        |
Host B Processing:   Receive RA, configure SLAAC address, default route,
                      DNS (if RDNSS)
```

Description of Normal Operation:

- **Unsolicited RAs:** Legitimate IPv6 routers periodically multicast Router Advertisement messages to the `ff02::1` (all-nodes) address. This occurs every 200-600 seconds by default.
- **Solicited RAs:** When a host first connects to a link or needs updated information, it can send a Router Solicitation (RS) message (ICMPv6 Type 133). Legitimate routers respond immediately with an RA.
- **Host Configuration:** Hosts use the information in these RAs for Stateless Address Autoconfiguration (SLAAC), discovering the default gateway, and potentially other configuration like DNS servers (via RDNSS option). They expect a relatively low, predictable rate of these messages.

2.2 Attack Timing Diagram (IPv6 RA Flooding)

The attack introduces an abnormal, extremely high frequency of RA messages.

```
Time -------------------------------------------------------------------------->

Legitimate Router:   RA (Unsolicited)  ----------------> RA (Unsolicited) ...
                      (Multicast)   (normal rate, if present)
```

```
Attacker Tool:      RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,
                     RA,RA,RA,RA,RA,RA,RA,...
                     (Rapid Multicast - e.g., 100s-1000s packets/second)

Host A:             Receiving RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,RA,
                     RA,RA,RA,RA,RA,RA,RA,...
                     (Continuous processing/reprocessing)

Host A Processing:  High CPU/Memory utilization, constant cache updates, frequent
                       address/route recalculations,
                     potential instability.
```

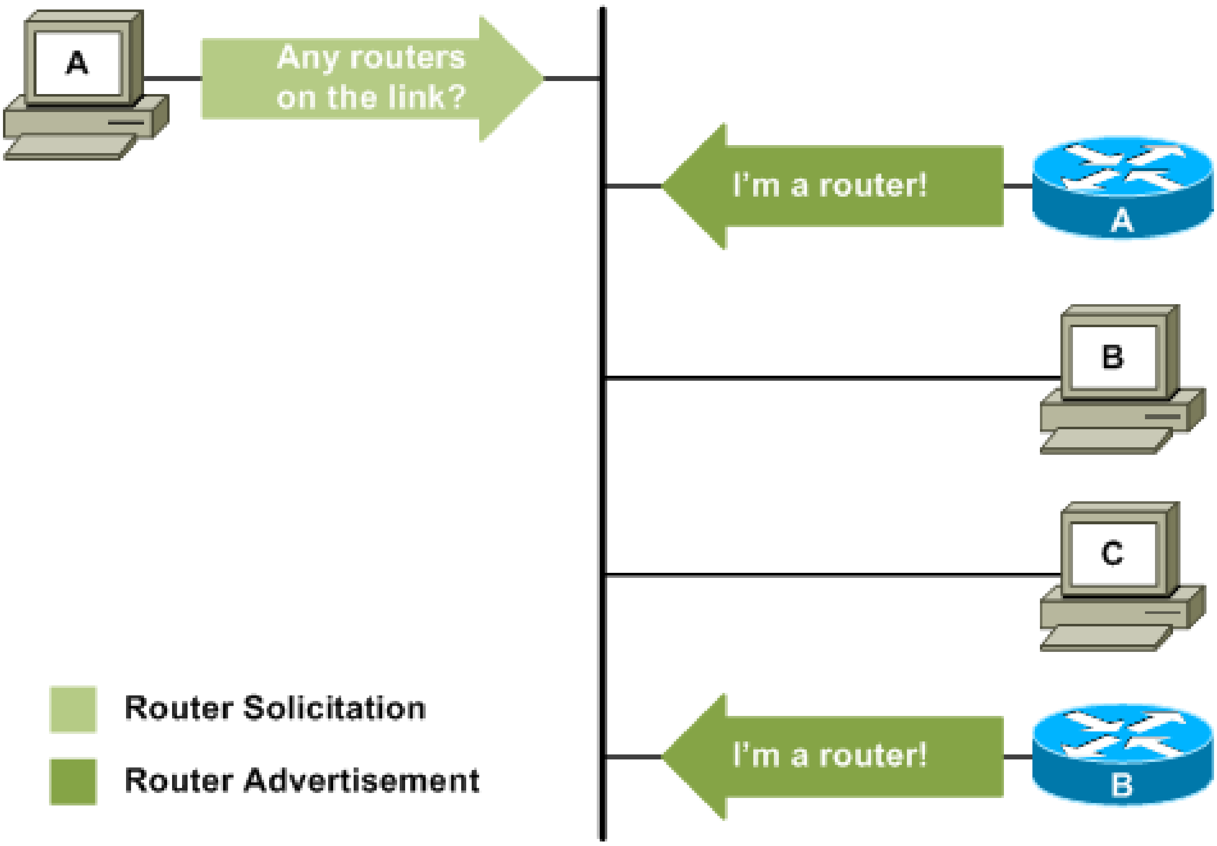


Figure 3: Router Discovery using ICMPv6 Router Solicitation and Advertisement

Attack Strategies:

1. **Pure Flooding (DoS):**
   - **Strategy:** Maximize the sending rate of RA messages, with less emphasis on the specific content of each RA (though valid headers are critical).
   - **Timing:** Send RAs with minimal or no delay between packets (e.g., `interval = 0` or `0.001` seconds). The goal is to saturate the network and overwhelm host processing.
   - **Impact:** Primarily resource exhaustion and network congestion.

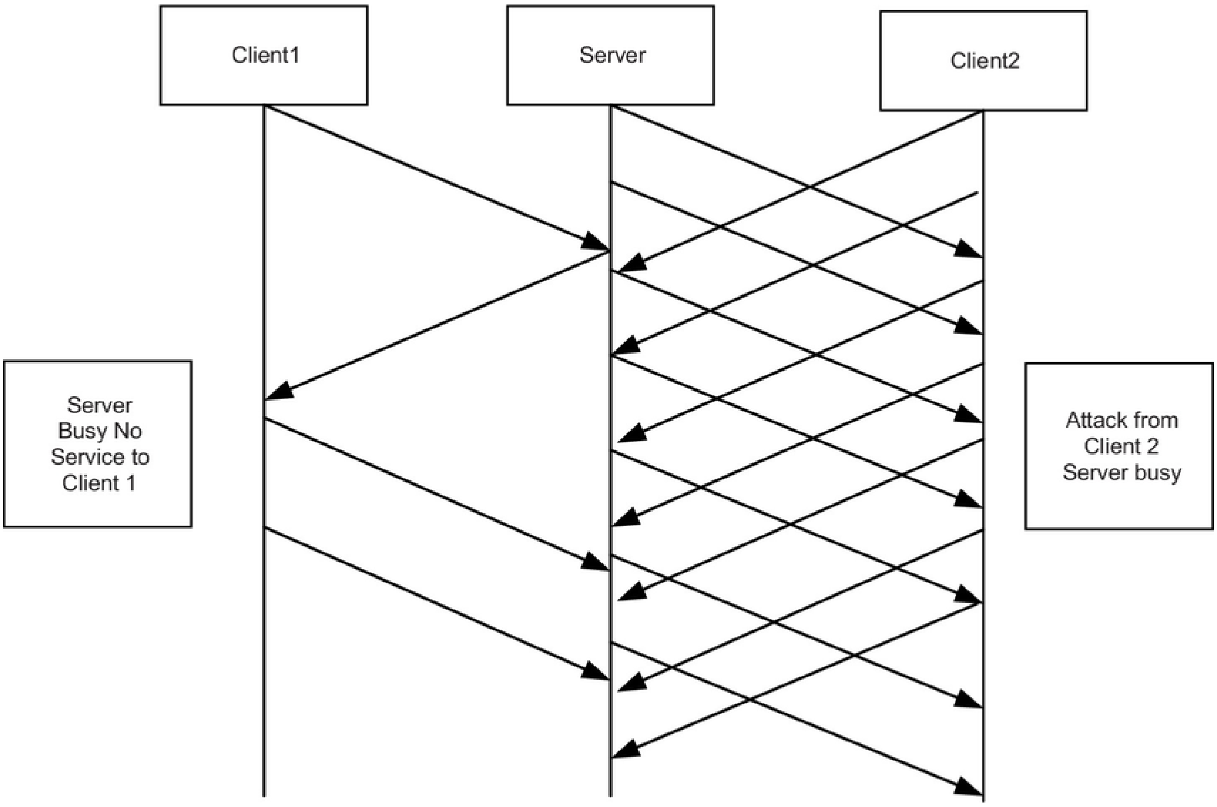

Client1
Server
Client2
Server Busy No Service to Client 1
Attack from Client 2 Server busy

Figure 4: DoS Attack

2. **Malicious Configuration Flooding (DoS + Potential MITM):**
    - **Strategy:** In addition to high volume, craft RAs with malicious or conflicting information.
    - **Timing:** Same high rate as pure flooding.
    - **Payload Modifications:**
        - **Rogue Prefix:** Advertise a new or overlapping prefix (e.g., `2001:db8:beef::/64`) with the Autonomous (A) flag set, causing hosts to autoconfigure incorrect addresses.
        - **Rogue Default Gateway:** Set the `Source Link-Layer Address` option to the attacker's MAC address, making hosts perceive the attacker as a valid router. Set `Router Lifetime` to a very high value to prolong the effect.
        - **Malicious DNS Server:** Include the `Recursive DNS Server (RDNSS)` option pointing to an attacker-controlled DNS server, enabling DNS hijacking.
        - **High Router Preference:** Set the `Prf` (Router Preference) flag to a high value (e.g., `1` for High) to make the rogue router preferred over legitimate ones.
    - **Impact:** Combines resource exhaustion with active misconfiguration, leading to potential MITM attacks, incorrect routing, and connectivity loss.

Our custom tool will focus on the **Malicious Configuration Flooding** strategy, as it offers a more complete demonstration of attack capabilities beyond mere resource consumption.

3 Packet / Frame Details for Your Attack and Any Modification in the Header

Our custom tool will construct Ethernet frames encapsulating IPv6 packets, which in turn encapsulate ICMPv6 Router Advertisement messages along with various options.

Full Packet Structure (from bottom up):

1. **Ethernet Header (Layer 2):** • **Purpose:** Encapsulates the IPv6 packet for transmission on the local link.
   - **Size:** 14 bytes
   - **Fields:**
     - `Destination MAC Address`: `33:33:00:00:00:01` (6 bytes) - Standard multicast MAC address for IPv6 all-nodes.
     - `Source MAC Address`: Attacker's physical interface MAC address (6 bytes) - E.g., `00:11:22:33:44:55`.
     - `EtherType`: `0x86DD` (2 bytes) - Indicates that the payload is an IPv6 packet.
2. **IPv6 Header (Layer 3):** • **Purpose:** Delivers the ICMPv6 message to the target.
   - **Size:** 40 bytes
   - **Fields:**
     - `Version`: `6` (4 bits)
     - `Traffic Class`: `0` (8 bits)
     - `Flow Label`: `0` (20 bits)
     - `Payload Length`: Length of the ICMPv6 message (including options) in bytes (16 bits) - *Calculated dynamically.*
     - `Next Header`: `58` (8 bits) - Indicates the next header is ICMPv6.
     - `Hop Limit`: `255` (8 bits) - Standard value for Neighbor Discovery messages.
     - `Source IP Address`: Attacker's Link-Local Address (LLA), e.g.,`fe80::dead:beef:cafe:1` (16 bytes).
     - `Destination IP Address`: `ff02::1` (16 bytes) - IPv6 all-nodes multicast address.
3. **ICMPv6 Header (Router Advertisement - Type 134):** • **Purpose:** Contains the core RA information.
   - **Size:** 16 bytes (excluding options)
   - **Fields:**
     - `Type`: `134` (8 bits)
     - `Code`: `0` (8 bits)
     - `Checksum`: (16 bits) - *Crucial! Calculated over ICMPv6 message + IPv6 pseudo-header.*

- `Current Hop Limit`: `64` (8 bits)
- `M (Managed Flag)`: `0` (1 bit) - For SLAAC.
- `O (Other Config Flag)`: `1` (1 bit) - To enable RDNSS option.
- `Router Lifetime`: `9000` (16 bits) - In seconds.
- `Reserved`: `0` (6 bits)
- `Prf (Router Preference)`: `1` (2 bits) - *High Preference.*
- `Reachable Time`: `0` (32 bits)
- `Retrans Timer`: `0` (32 bits)

4. **ICMPv6 Options (Variable Size): • a) Source Link-Layer Address Option (Type 1):**
   - `Type`: `1` (8 bits)
   - `Length`: `1` (8 bits) - In units of 8 bytes.
   - `Link-Layer Address`: Attacker's physical interface MAC address (6 bytes).

   - **b) Prefix Information Option (PIO) (Type 3):**
     - `Type`: `3` (8 bits)
     - `Length`: `4` (8 bits) - 32 bytes (4 * 8 bytes).
     - `Prefix Length`: `64` (8 bits)
     - `L (On-Link Flag)`: `1` (1 bit)
     - `A (Autonomous Flag)`: `1` (1 bit)
     - `Reserved2`: `0` (6 bits)
     - `Valid Lifetime`: `0xFFFFFFFF` (32 bits) - Very high.
     - `Preferred Lifetime`: `0xFFFFFFFF` (32 bits) - Very high.
     - `Reserved3`: `0` (32 bits)
     - `Prefix`: `2001:db8:dead:beef::` (16 bytes) - *Malicious/Arbitrary prefix.*
   - **c) Recursive DNS Server (RDNSS) Option (Type 25) - Optional for MITM:**
     - `Type`: `25` (8 bits)
     - `Length`: `3` (8 bits) - For one DNS server.
     - `Reserved`: `0` (16 bits)
     - `Lifetime`: `9000` (32 bits) - In seconds.
     - `DNS Servers`: `2001:db8:evil::1` (16 bytes) - *Malicious DNS server address.*

Modifications in Header/Payload for Attack:

- **Source MAC/IP (for the "router"):** While we use the attacker's real MAC/LLA for simplicity, these fields effectively identify the *rogue router*. The attacker's machine is *not* a legitimate router.

- **Router Lifetime:** Set to a very high value (e.g., 9000 seconds) to ensure hosts consider the rogue router valid for an extended period.

- **Router Preference (Prf):** Set to 'High' (`1`) to make the rogue router more attractive than legitimate ones.

- **Prefix Information Option (PIO):**
  - **Prefix:** Advertise a malicious or arbitrary prefix (e.g., `2001:db8:dead:beef::/64`) that conflicts with or is irrelevant to the actual network.
  - **Lifetimes:** `Valid Lifetime` and `Preferred Lifetime` set to extremely high values (e.g., `0xFFFFFFFF`) to ensure hosts retain this erroneous prefix indefinitely.
- **RDNSS Option (Optional):** Include a `Recursive DNS Server (RDNSS)` option pointing to an attacker-controlled or non-existent DNS server to hijack DNS queries.
- **Sending Rate:** This is the most critical modification. The tool will send these carefully crafted RA packets at an extremely high frequency (e.g., thousands per second), far exceeding the legitimate RA rate (one every 200-600 seconds). This high volume is the core of the flooding aspect.

4 Justification

This design leverages fundamental aspects of IPv6 networking and the raw socket programming model to achieve the RA flooding attack:

1. **Raw Socket Capability:**
   - **Full Packet Control:** By using raw sockets, our tool bypasses the operating system's normal network stack for packet construction. This allows us to craft *every single byte* of the Ethernet, IPv6, and ICMPv6 headers, including normally restricted fields like the source IP (for LLAs) and destination MAC (for multicast), and crucial elements like the ICMPv6 checksum.
   - **Layer 2 Injection:** Sending at the Ethernet layer ensures the packets are injected directly onto the wire with our specified MAC addresses, rather than relying on the kernel's ARP/NDP resolution.
2. **Exploiting IPv6 Multicast and NDP Trust Model:**
   - **All-Nodes Multicast:** Router Advertisements are always sent to the IPv6 all-nodes multicast address (`ff02::1`) and the corresponding multicast MAC (`33:33:00:00:00:01`). Any device configured for IPv6 on the local segment will receive and process these packets, regardless of their source. Our tool leverages this by sending to these specific multicast addresses.
   - **Lack of Native Authentication:** IPv6 Neighbor Discovery Protocol (NDP) was designed with an assumption of a trusted local link. Without advanced security measures like SEND (Secure Neighbor Discovery), hosts on a segment simply trust any RA message they receive. Our attack exploits this fundamental trust model; the victim hosts have no built-in mechanism to verify if the sending "router" is legitimate.
3. **Overwhelming Processing Demands on Victims:**

- **Kernel Processing:** Every incoming network packet, especially those that trigger protocol-level processing (like ICMPv6), demands CPU cycles and memory from the operating system kernel.
- **NDP Cache Updates:** Hosts maintain a Neighbor Cache (like an ARP table) and a Destination Cache (routing entries). Each RA, particularly those with new or conflicting information (different router lifetime, preferred prefixes, RDNSS options), can trigger updates, lookups, and re-evaluations in these caches. An overwhelming rate of RAs forces constant, rapid updates, leading to:
  - **High CPU Utilization:** Constantly processing new packets, recalculating checksums, and updating kernel data structures.
  - **Memory Exhaustion:** Allocating memory for incoming packet buffers, and potentially for rapidly changing or growing NDP cache entries.
  - **Context Switching:** Frequent interrupts and context switches as the kernel handles the packet flood.
- **SLAAC Reconfiguration:** If the RAs advertise different prefixes with the Autonomous (A) flag, hosts will attempt to autoconfigure new IPv6 addresses, further consuming resources and potentially causing address instability.
- **Default Route Churn:** If multiple rogue RAs with varying router lifetimes or preferences are sent, or if they conflict with legitimate RAs, the host's default route may constantly change, leading to intermittent connectivity.

4. **Network Congestion:**

   - While individual RA packets are small (typically ∼70-100 bytes including headers), sending thousands per second (e.g., 1000 packets/sec × 100 bytes/packet = 100 KB/sec raw data) can consume significant bandwidth on a local segment, especially on slower links or Wi-Fi. This can lead to legitimate traffic being dropped or significantly delayed, causing a network-level DoS.

5. **Unthrottled Packet Generation:**

   - Our custom tool allows for precise control over the sending interval. By setting a minimal or zero delay between packets, we can achieve the maximum possible packet sending rate supported by the attacker's hardware and operating system, far exceeding any legitimate network traffic or protocol timing expectations.

5 Existing Tools and Frameworks

While this project focuses on designing and implementing an RA flooding tool from scratch using raw sockets, it is important to acknowledge the existing, publicly available tools and frameworks that can perform similar attacks. These tools serve as a benchmark, demonstrate the accessibility of such attacks, and highlight the need for robust network defenses.

THC-IPv6 Toolkit: A comprehensive suite of tools designed specifically for attacking IPv6 and its related protocols. It includes the `flood_router6` utility, which is purpose-built for RA flooding. This command-line tool automates the process of sending a high volume of RA packets, allowing an attacker to launch a powerful DoS attack with a single command. It offers various options to customize the source MAC/IP addresses and other packet parameters.

Nmap Scripting Engine (NSE): Nmap, a ubiquitous network scanner, can be extended with Lua scripts to perform a wide variety of tasks, including vulnerability discovery and network attacks. The `ipv6-ra-flood.nse` script leverages Nmap's powerful packet-sending capabilities to execute an RA flooding attack against all hosts on the local network segment, effectively demonstrating the DoS potential.

Scapy: A powerful interactive packet manipulation library for Python. While not a pre-packaged "tool" in the same way as THC-IPv6, Scapy is a very common framework for prototyping and executing custom network attacks. An attacker can use Scapy to build and send a flood of crafted RA packets with just a few lines of Python code. It offers a balance between the ease of use of pre-built tools and the full granular control of raw socket programming

6 Defenses and Mitigation Strategies

Effective mitigation against RA flooding attacks requires a multi-layered security posture, as simple, single-point defenses are often insufficient. A robust defense combines network-level filtering with host-based controls.

6.1 Network-Level Defenses and Bypasses

The primary network-level defense is **RA-Guard** (RFC 6105), a switch feature that operates by classifying ports as trusted (for routers) or untrusted (for users) and blocking RA messages on untrusted ports.

However, naive implementations of RA-Guard are vulnerable to a well-known bypass technique documented in RFC 7113. An attacker can encapsulate the malicious RA message behind an **IPv6 Extension Header** (e.g., a Fragmentation Header). The simple RA-Guard inspects only the first header, sees a non-RA packet, and forwards it. The target host, however, processes the full header chain and is successfully attacked.

6.2 Recommended Defense-in-Depth Strategy

To properly secure a network, a comprehensive strategy is necessary:

1. **Deploy Advanced RA-Guard:** Use switch hardware with an RFC 7113-compliant version of RA-Guard. This advanced implementation performs deep packet inspection, analyzing the entire extension header chain to find and block malicious RAs.

2. **Enable Port Security:** Implement switch-level Port Security to limit the number of MAC addresses that can be learned on a single port. This directly mitigates attempts to flood the network with spoofed source MAC addresses.

3. **Utilize Host-Based Filtering:** As a final layer of protection, configure host-based firewalls on endpoints. These firewalls should be set with rules to accept ICMPv6 RA packets *only* from the known, legitimate MAC addresses of the network's trusted routers.

, making it an excellent choice for creating proof-of-concept attack scripts.

7 Conclusion

In conclusion, the design directly exploits the trust model and processing requirements of IPv6's Neighbor Discovery Protocol, combined with the low-level control afforded by raw sockets. By manually crafting and rapidly injecting a high volume of ICMPv6 Router Advertisement messages onto the local network segment, this tool is designed to effectively achieve resource exhaustion, configuration instability, and network congestion on target hosts, thus demonstrating a successful RA flooding attack.