

Impacts Evaluation of DoS Attacks Over IPv6 Neighbor Discovery Protocol

¹Amjed Sid Ahmed, ²Rosilah Hassan, ²Nor Effendy Othman, ³Nor Idayu Ahmad and ⁴Yassir Kenish

¹Computing Department, Engineering Faculty, Global College of Engineering and Technology (GCET), P.O. Box 2546 CPO Ruwi 112, Muscat, Sultanate of Oman

²Research Centre for Software Technology and Management (SOFTAM), Network and Communication Technology Laboratory (NCT LAB), Faculty of Information Science and Technology (FTSM),

Universiti Kebangsaan Malaysia (UKM), 43600 UKM, Bangi, Selangor, Malaysia.

³Network Research Group (NGR), Faculty of Information and Communication Technology (FICT), Limkokwing University of Creative Technology (LUCT), 63000 Cyberjaya, Selangore, Malaysia

⁴Information Technology Department, Global College of Engineering and Technology (GCET), P.O. Box 2546 CPO Ruwi 112, Muscat, Sultanate of Oman

Article history

Received: 19-07-2018

Revised: 20-02-2019

Accepted: 29-05-2019

Corresponding Author:

Amjed Sid Ahmed

Computing Department,
Engineering Faculty, Global
College of Engineering and
Technology (GCET), P.O. Box
2546 CPO Ruwi 112, Muscat,
Sultanate of Oman

Email: amjed@gcet.edu.my

Abstract: The Neighbor Discovery Protocol (NDP) is one of the main protocols in the Internet Protocol version 6 (IPv6) suite. It provides many basic functions for the normal operations of IPv6 in a Local Area Network (LAN), such as address auto-configuration and address resolution. However, NDP has several vulnerabilities that can be used by malicious nodes to launch attacks, because NDP messages are easily spoofed. Surrounding this problem many solutions have been proposed for securing NDP but these solutions either proposed new protocols that need to be supported by all nodes or built mechanisms that require the cooperation of all nodes. In this paper we overview NDP vulnerabilities and available solutions to overcome their impacts on IPv6 network. In addition a research test bed setup to implement these vulnerabilities was introduced. Moreover attacks that prove these vulnerabilities are implemented on different types of operating systems, Windows and Linux platforms. Three network metrics throughput, delay and resources consumption have been chosen to investigate, analyze and evaluate the impacts of NDP related attacks on IPv6 link-local communication. Overall, the results had shown that performance of Linux based operating system is better than Windows based operating system.

Keywords: IPv6, NDP, SLAAC, DoS

Introduction

IPv6 is a protocol designed as the successor to IPv4 protocol (Hakiem *et al.*, 2015). It is used to solve the problems faced by IPv4 in today's internet, such as IP address space limitation, security and scalability. Compared with the 32-bit length of the IP address in IPv4, the IPv6 address comprises 128 bits. This is absolutely enough in the foreseeable future as it supports an IP address for each single meter on the earth. The NDP is an auxiliary protocol for IPv6 and it comprises two Requests For Comments (RFC): Neighbor Discovery for IPv6 (Anbar *et al.*, 2016) and IPv6 Stateless Address Auto Configuration (SLAAC)

(Ahmed *et al.*, 2017). The former is used for discovery of the IPv6 nodes on the same link and the latter allows the hosts to automatically configure the IPv6 address without the outside help like Dynamic Host Configuration Protocol (DHCP) server.

As the IPv6 address is long and its address space is huge, SLAAC is a very convenient function and makes the IPv6 network become plug-and-play. For the normal operation of IPv6, NDP also provides other functions including router/prefix/parameter discovery, address resolution, next-hop determination, Neighbor Unreachability Detection (NUD), Duplicate Address Detection (DAD) and redirection. All of these functions are based on the transmission of NDP messages, which

are encapsulated in Internet Control Message Protocol Version 6 (ICMPv6) packets. NDP messages are confined to a link and only transmitted in the scope of a LAN. This means attached routers will not forward NDP messages from one network to another. According to Anbar *et al.* (2016), NDP uses five types of ICMPv6 messages as follows. Router Solicitation (RS), hosts send RS messages to find the default router and request for the network information from routers. Router Advertisement (RA), RA message is sent by routers periodically or in response to the RS message. Neighbor Solicitation (NS), nodes send NS message to resolve a neighbor node's IPv6 address to its Media Access Control (MAC) address or to detect the reachability of a neighbor. Neighbor Advertisement (NA), a node sends NA message to answer solicited NS message or sends unsolicited NA message to propagate its changed information, such as the MAC address variation. Redirect Message (RM), routers send redirect packets to inform a host of a better first-hop node on the path to a destination, a summary of NDP messages and functions presented in Table 1.

Here, we introduce two procedures of the functions to show how the NDP messages are used. The first is address resolution. When a node wants to communicate with another node using IPv6 address without knowing the corresponding MAC address, it will firstly send a multicast NS message to ask all nodes in the LAN who has this IPv6 address. Then, the node occupying this address will send back a unicast NA message to advise its MAC address. The second is DAD procedure. When a node auto-configures itself with an IPv6 address, it will firstly verify the uniqueness of this address. It orderly sends several NS messages with setting the destination as solicited-node multicast address.

Then, if it receives any NA message in response to this solicitation, this address is already used. Otherwise, this address could be issued on the network by this node. From these two examples, we could find that both procedures are vulnerable to be attacked through spoofing. A fake reply to address resolution may lead to Man-in-the-Middle (MITM) attack and forged NAs to DAD will result in Denial of Service (DoS) attack.

From all of the above, we discern that NDP is an essential component in an IPv6 network LAN. However, there are many security issues related to NDP that can be used by attackers to impact the legitimate communication of users. Although the NDP defined many rules for the nodes to send or receive NDP messages legitimately, there is no compulsive method to guarantee the node behaves normally. Therefore, malicious nodes can launch attacks through illegally using NDP messages. An effective authentication mechanism is very important for securing the NDP.

Table 1: NDP message and function

Message name	ICMPv6 type	Function
Router Solicitation	133	Router discovery
Router Advertisement	134	Router presence
Neighbor Solicitation	135	Neighbor discovery
Neighbor Advertisement	136	Neighbor presence
Redirect	137	Better next hop

The necessity to have a test bed along with its correspondent configurations, topologies, attacking tools and data gathering techniques to study NDP cannot be denied. Such setup will allow researchers to study the behavior of the real networks under different types of NDP attacks. Moreover, the test bed setup could help researchers in future with newly proposed solutions, against NDP attacks, to test the effectiveness and efficiency of these solutions. In this paper we provide a complete test bed setup for examining IPv6 NDP related attacks. The impacts of these attacks under different types of operating systems have been investigated, analyzed and evaluated. This paper is organized as follow; in part two we overviewed DoS attacks showing their types and classifications. Following by part three in which NDP vulnerabilities are well explained and categorized according to their relation to the routing process. The test bed setup along with corresponding configurations to implement NDP attacks was given in part four. Gained results of impacts evaluation for the attacks are presented in part five. Existing solutions in the era also covered in part six and we provide conclusion in part seven.

Denial of Service Attacks

Overview of Denial of Service Attacks

One of the major concerns in interconnected networks of the current era is the network security. Network traffic can be disrupted by attack on one node which could severely affect the other nodes in a network. A network server may encounter various kinds of attacks, time to time, which results in the degrading of the performance of server in the network. A DoS which is considered to be a really troublesome problem to handle is one example of these attacks. A DoS attack takes place by preventing the victim node, by a malicious node, from communicate with other nodes on the network, as per Fig. 1. Consequently the victim node won't be able to process requests received from all other nodes. And because of this, the services needed by the authentic users could not get provided to them. Due to this, the inspection of the network traffic is essential to find the malicious or infected packets. And it should be done in such a way that the malicious packets are isolated from the uninfected ones thereby delivering services to the authentic users or clients smoothly. A

small amount of resources and bandwidth are essential for the attackers to execute DoS attack. The attacks can take place in several ways, one way in which software vulnerabilities present in the victim node are exploited by an attacker and another way wherein an attacker produces a huge number of malicious packets (Rehman and Manickam, 2016). A web server can be crashed by these types of attacks no matter what hardware capabilities it possesses. The first major DoS attack, recognized as email worm, was executed in Europe in the year 1987 by an IBM employee. The attack gathered quite some attention because IBM's shared network became overloaded and crashed in both continents Europe and USA. As a result of system downtime and recovery (Rehman and Manickam, 2015a), a significant damage is still being caused to the productivity and revenues of corporates networks by these types of attacks. IPv6, which was created by the Internet Engineering Task Force (IETF) in order to address the limitations of IPv4, is exposed to DoS attacks. Legitimate nodes are prevented from acquiring access to network resources as a result of DoS attacks. Stealing of information is not included in a DoS attack instead the security of a network is violated and tends to discontinuing network connections. As these types of attacks are designed for the IP network, they can target any system regardless of its operating system. Therefore, any operating system using IPv4 or IPv6 can encounter these attacks (Rehman and Manickam, 2015b). Even though they are frequently aimed at IP network services, DoS attacks can also threatening VoIP and other real-time services. The source of the DoS attack can be hidden by the attackers by means of spoofing, i.e., IP address spoofing or MAC address spoofing.

Classification of Denial of Service Attacks

A single computer is needed in launching of a DoS attack, while Distributed Denial of Service (DDoS) attack is more complex than a DoS attack. A DDoS attack involves a number of compromised computers, known as zombies, which are all used at the same time (Baishya *et al.*, 2017). Accordingly, flooding-based attacks could be initiated from one source in case of DoS attack or multiple sources in case of DDoS attack. Below we will explain the differences between software and flooding types of DoS attacks.

Software Exploits

A low-rate DoS attack which, in order to remain hidden, keeps a low profile is referred to as software exploit. For the purpose of making use of the system vulnerabilities, to prevent authentic users from acquiring access to services and available resources, the attacker utilizes malicious nodes in a software exploits attacks (Kavitha and Padmavathi, 2017).

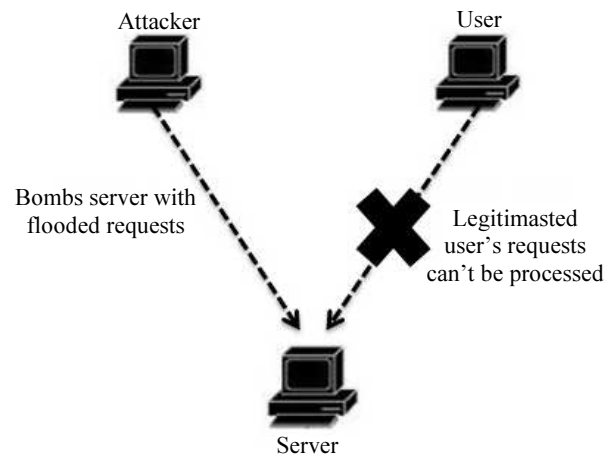


Fig. 1: Denial of service attack

Flooding

In this type of DoS attack, the attacker sends a non-stoppable massive amount of packets to the victim's node to dissipate resources that can be earned by legitimate users. Due to this, the victim node freezes as the processing of the flood of malicious packets consumed all available resources. Traffic may be transferred from other nodes to the victim mode by the attacker during flooding attack (Najjar *et al.*, 2015). Resulting in causing network congestion and consume the resources of the victim node like Central Processing Unit (CPU), memory or bandwidth. Consequently, network communication amongst the victim and other nodes is prevented by this type of attack (Rehman and Manickam, 2015c).

DoS Attacks on Internal Networks

Web servers, which do not have a direct link to the internal network of an organization, are not the only targets of DoS attacks. Internal networks are also susceptible to DoS attacks. In order to acquire access of the internal network, the attackers may utilize malware. Saad *et al.* (2015) mentioned that a research, which included respondents from 130 organizations, was held in 2012 for the purpose of recognizing the security concerns of organizations like those related to internal IPv6 networks. In accordance with 70% of the respondent, we came to know that DoS attacks were amongst their IPv6 security concerns. Compromised hosts on the internal networks were face by approximately 50% of the respondents. We can understand from this that the respondents were really concerned about the monitoring and guarding the availability of services on their internal IPv6 networks. Due to the need of the attacker to get access to the local IPv6 network in order to initiate attack, these types of attacks may be considered as minor. Nevertheless, access to the local network can be granted through a number of techniques and tools. For the purpose of stopping the

malicious packets from passing through a firewall or a Demilitarized Zone (DMZ) (Shrivastava *et al.*, 2010). Malware can be utilized by an attacker in order to avoid firewalls so that access can be granted to the internal LAN. A number of organizations have encountered network attacks initiated by malicious insiders; a trusted person from the organization is referred to as an insider. In case those malicious insiders are the ones who are initiating attacks, it will be troublesome to find them out because of the fact that insiders generally have knowledge about the security mechanisms of the organization's network (Kuldeep and Tyagi, 2014). A link-local DoS attack cannot be prevented with the utilization of encryption and integrity checks, which are commonly used to encounter attacks that take place outside the network. Packets of DoS attack may be signed by a server and they might contain real or fake IP address. With the usage of imaginary key the attack packets can also be encrypted. Accordingly, attacks cannot be stopped by use of encryption and integrity checks and the devices inside of an internal network can be flooded by an attacker causing them to stop working.

DoS Attacks via IPv6 Tunneling

We can suppose, provided that an enterprise is not using IPv6, that the IPv4 network is secured from IPv6 attacks. However, network administrators may not realize that cryptic IPv6 tunneling is taking place in the network, in order to deal with IPv4 only networks (Hassan *et al.*, 2014). On the network, a malicious IPv6 appliance might be there. A number of operating systems have IPv6 enabled by default including, Windows 7, Linux, MAC OS/X. With IPv4, IPv6 traffic can be tunneled therefore evading security controls which are meant for IPv4 only. Actually, IPv6 tunnel can serve as being a backdoor into the inside network. IPv4 to IPv6 transition mechanisms can be utilized by the attackers, which include Teredo, for initiating DoS attacks. IPv6 over IPv4 tunneling protocols can be detected by the edge devices, for instance routers or firewalls, even if encapsulated IPv6 packets cannot be secured by these devices. Conventional network security tools, for instance Intrusion Detection Systems (IDS), which perform in IPv4 environments only, are not useful for IPv6 transition mechanisms like tunneling. While an IPv6 flooding-based DoS attack is taking place, these tools may not be successful in detecting of anomalies.

NDP Vulnerabilities

According to RFC 3756, NDP vulnerabilities have three common types. The redirect attacks are the first vulnerability type whereby the malicious nodes are to direct away the packets from the legitimated nodes. Hence, we cannot trace the packets from the last hop router. It is important to mention that other genuine

receivers are directed to alternative nodes upon facing the redirect attacks. The DoS is believed to be the second category of NDP vulnerabilities. The preventions of information flow between the attacked nodes and all other nodes, performed by malicious nodes, are likely to describe this type of attack (Ahmed *et al.*, 2015a). The communication is also disallowed between the attacked nodes and specific intended addresses. Thirdly, the NDP is encountered by the attack of Flooding DoS (Ahmed *et al.*, 2015b). The malicious nodes direct the traffic of other hosts to the victim node in such attack. A scenario of flooded bogus traffic is created whereby the victim host is the target. Three sub sections are used to identify threats, of NDP, with regarding to routing process in the given below section. These are: Threats that are related to the routing data, router independent threats and threats that can be remotely manipulated. We used NDP trust models and threats in RFC 3756 to outline those categories of threats.

Non Routing Based Threats

Neighbor Solicitation/Advertisement Spoofing

In this type of attack, legitimated nodes will not receive their legitimated packets. Instead the attacker will divert it to other node either by sending NA message with incorrect target link layer address or NS message with incorrect source link layer address, as per Fig. 2.

Neighbor Unreachability Detection (NUD) Failure

This attack success because the attacker send a fabricated NA message in response to the victim NS message during NUD process (Praptodiyono *et al.*, 2015a). The victim will be cheated by receiving this fabricated NA message and thought the neighbor is still reachable, while it is not.

Duplicate Address Detection DoS Attack

When a new node join an IPv6 link, it will make DAD check for the address that it trying to use. This is the nature of SLAAC mechanism within IPv6 communication link. As a response the attacker will replay to every single check for an IPv6 address that victim trying to use, claiming that he (attacker) already using this address (Rehman and Manickam, 2015c). This will prevent the victim from gaining a valid address and consequently denied access to the communication link, as per Fig. 3.

Routing Based Threats

Malicious Last Hop Router

Attacker in this type of attack pretending to act as last hop router by sending spoofed RA messages either as a response to RS message or in a routine base. This spoofed RA message, with the last hop router source

address, has a short router life time. Followed by another RA message, has attacker source address, but with longer router life time (Song and Ji, 2016). Once the victim select attacker address as default router all traffic will be directed to the attacker’s host instead of the last hop router, as per Fig. 4.

Default Router is Killed

In this type of attack the victim assumes that all nodes are local. This is simply happened because

attacker killed the default router, either by launching a DoS attack against the router or sends a spoofed RA message with zero life time and make default router list empty (Praptodiyono *et al.*, 2015b). Consequently and according to RFC 2461 victim will never send packets to the default router, as per Fig. 5.

Good Router Goes Bad

A router that earlier was trusted is compromised in such attack. This is known as a redirect/DoS attack.

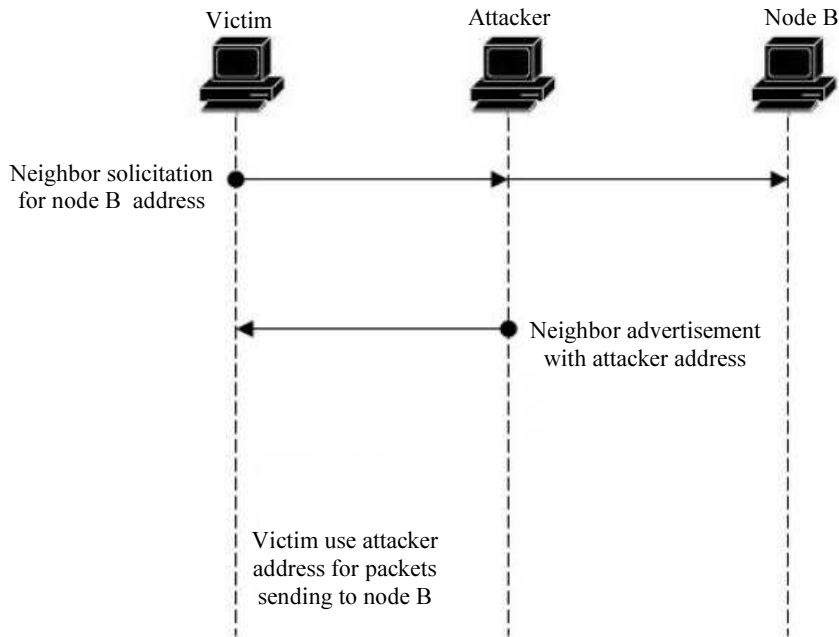


Fig. 2: Neighbor solicitation/advertisement spoofing attack

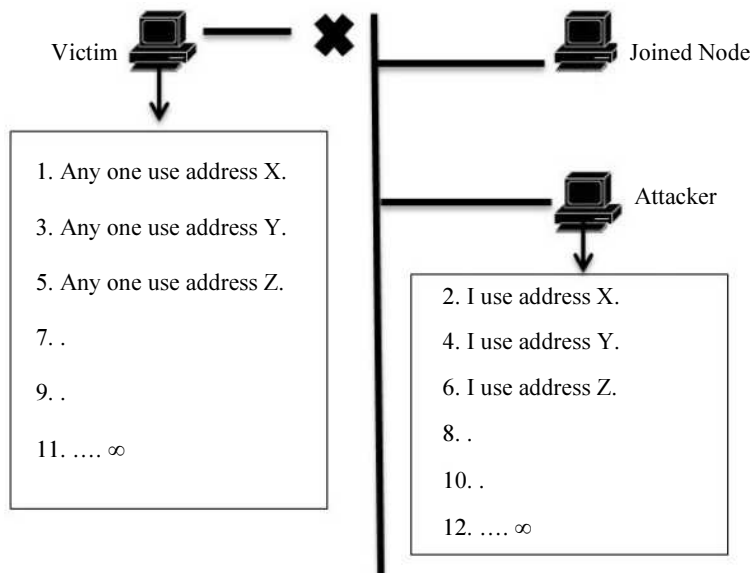


Fig. 3: Duplicate address detection DoS attack

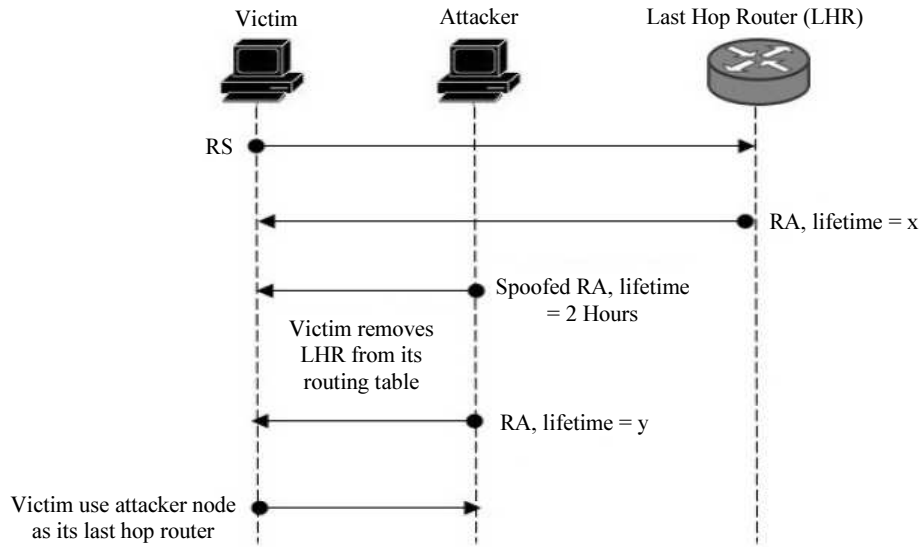


Fig. 4: Malicious last hop router DoS attack

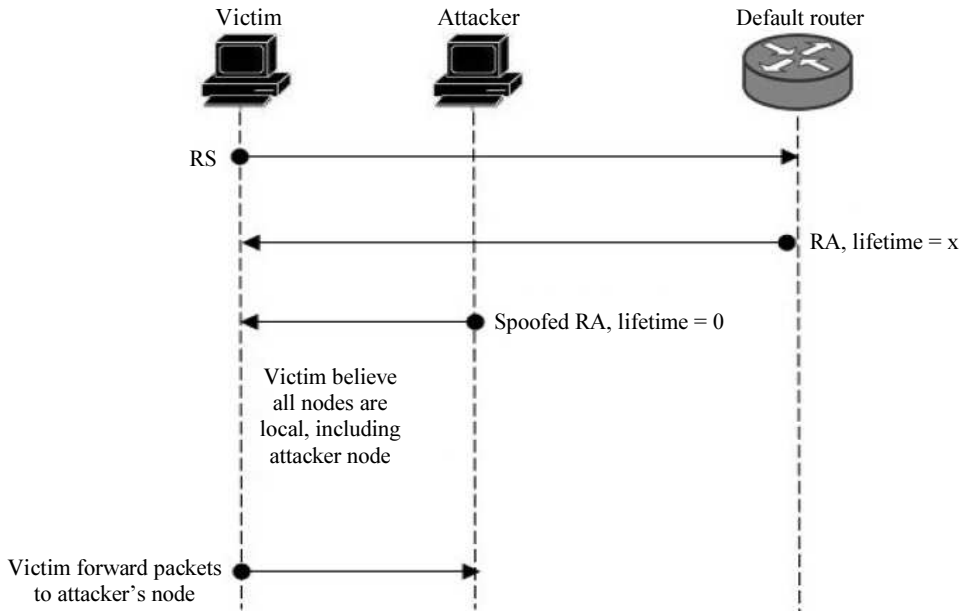


Fig. 5: Default router is killed DoS attack

Spoofed Redirect Message

This attack used to redirect packets for a specific destination to another node attached to the local link. The attacker uses the current first hop router’s link-local address to send spoofed redirect message (Perumal and Priya, 2016). Packets will continue to flow to that specific destination as long as attacker replays to NUD messages.

Bogus On -Link Prefix

The attacker cheats the victim that some prefix is on-link by sending fabricated RA message.

Accordingly the victim will assume the nodes are on-link and instead of send the packets to router it will send NS messages that will never be responded and lead to service denying to that node.

Bogus Address Configuration Prefix

In this type of attack the victim received a bogus RA message from attacker that identify wrong subnet prefix. Consequently and according to SLACC procedure the victim will use this invalid prefix and construct invalid address. The victim will denied service as a result because nodes will replay using

invalid source address of the victim when sending packets to victim's host (Shah, 2016).

Parameter Spoofing

As a part of SLAAC procedure the RA message contains some parameters that should be used by nodes in order to establish communication. The attacker executing this attack by sending RA messages that include incorrect parameters that may cause the communication between nodes to be interrupted (Shah and Parvez, 2015).

Replay Threats

Replay Attacks

The replay attacks are susceptible to all router discovery and neighbor discovery messages. The valid messages can also be captured by an attacker and he/she would replay them later, even if they were cryptographically secured so that one cannot falsify their contents. Hence, a secure mechanism must be established for protection against replay attacks.

Neighbor Discovery DoS Attack

The addresses are fabricated with the subnet prefix and packets are continuously being sent to the victims in such type of attack. After sending neighbor solicitation packets, these addresses are resolved by the last hop router (Najjar *et al.*, 2016). From the last hop router, the neighbor discovery service is not obtained by a legitimate host attempting to enter the network as it will be already busy with sending other solicitations. Since the attacker may be off-link, this DoS attack is different from the other attacks. In this attack, the conceptual neighbor cache is the resource being attacked, which will be occupied with attempts to resolve IPv6 addresses containing a valid prefix but invalid suffix (Mohamed *et al.*, 2017).

Experimental Work

Evaluation Methods

Simulation is commonly used for finding answers to network performance questions. However, simulation software cannot be used to produce experimental results that are as accurate as the results obtained using a real network such as a test bed. For example some devices, such as switches and routers, are only modeled at high levels in well-known simulators like Network Simulator 2 (NS-2). The ranges of latencies within devices and maximum rates at which packets are forwarded, in commercial forwarding devices, are not included in such simulators.

Experiments can be conducted in a mini-network, such as a test bed, which provides a more realistic evaluation environment compared to simulation. One of the reasons is that real operating systems, applications and real hardware are used to conduct experiments.

Both legitimate and DoS traffic can be generated and customized in a number of ways with such experiments. Even though it is time consuming compared to the simulation methods, test beds usually produce results that are more reliable.

In this paper a network test bed was deployed to report the impacts of DoS attacks over NDP. Figure 6 illustrates the test bed we used to collect data, from experiments conducted, before and during the different types of DoS attacks. For the cabling we did used the Category 5 Enhanced (Cat5e) cable type and default IPv6 subnet size/64 were used. Once RA DoS attack launched, all hosts configured with automatic IPv6 addresses, excluding the attacker, lost their connectivity to the communication link. Therefore we used static and dynamic IP addressing plans as automatic IP addressing is not suitable to study and evaluate some DoS attacks. The test bed consists of monitoring computer, one attacking computer and two victim's computers. As shown in Fig. 7a Windows-based computer with static IPv6 address FE80::1, to test the Transfer Control Protocol (TCP) throughput and Round Trip Time (RTT) before and during the attacks, was set up as monitoring computer. Two victims' computers Windows and Linux based, which had a statically configured IPv6 address FE80::2 and FE80::3 respectively, were used to test their behaviors and performance before and during attacks. Kali Linux was used to launch attacks with IPv6 address FE80::4. We need the automatics IP addressing configuration to test some routing related attacks. For the purpose of automatic assignment of IP addresses, using SLAAC, a D-Link router was used. Because NDP attacks are local-link scope, the router does not connect to an outside network.

Performance Metrics

Three performance metrics were used TCP Throughput, RTT delay and CPU utilization to evaluate the impacts of DoS attacks over NDP.

Network throughput defined as the average number of bytes received successfully by the intended receiver at a given time. Impacts of DoS over a network could be measured using a parameter such as TCP Throughput. Throughput is important for TCP based traffic, as it may lower the ratio at which it sends packets in case of network congestion occurred. TCP Throughput was measured on Windows 10 client using Iperf, it was measured in Mega Bytes per second (MBps).

RTT is calculated by subtracting the time at which a network packet were sent from the time at which acknowledge, for this packet, is received. RTT is significant because it used for measuring delay within computers networks. A packet considered lost if it is go beyond its predefined RTT, that's why during DoS attack retransmissions always occurred. RTT delay was measured on Windows 10 using Windows Ping utility, it was measured in milliseconds.

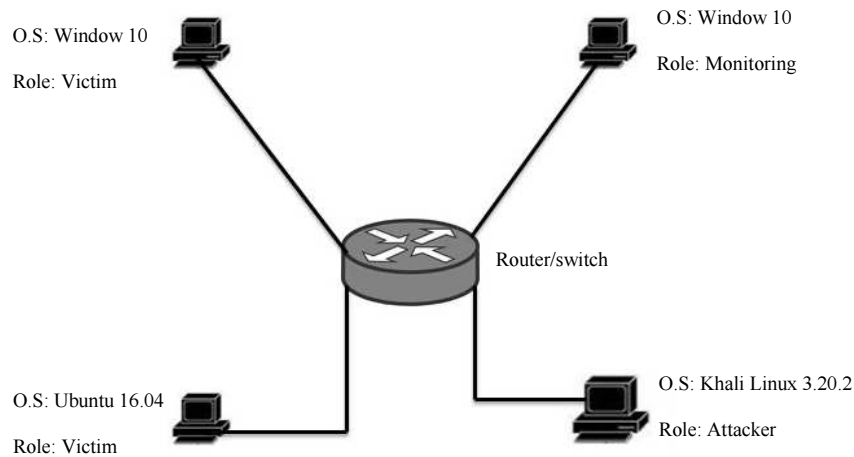


Fig. 6: Test bed topology

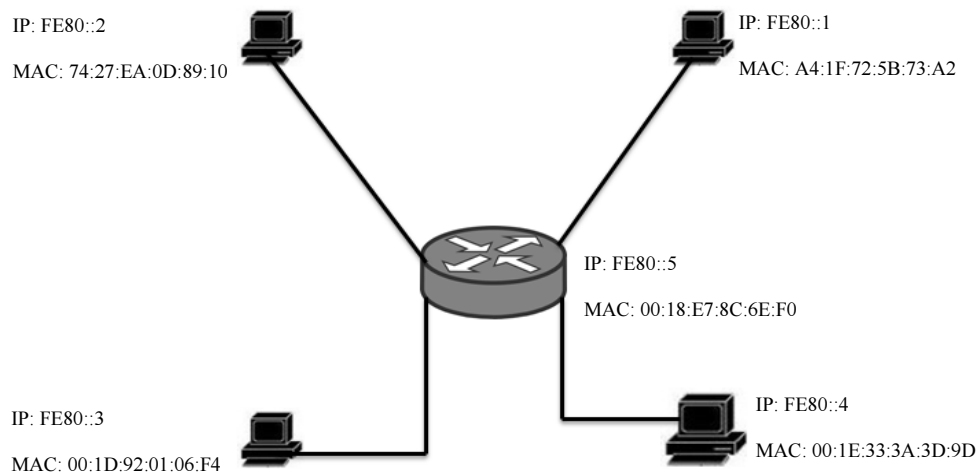


Fig. 7: Test Bed addressing scheme

During DoS based attacks packet transmission exhaust the processor, which in turn reduce the host's performance. CPU utilization was measured as percentage using resource monitor and system monitor on Windows 8 and Ubuntu 16.04 respectively.

Data Collection Tools

A test bed environment was deployed to carry out experiments and collecting data for analysis as mentioned earlier. After then, gained results were entered to Microsoft Excel spreadsheet to generate graphs.

Iperf is a network tool that measures TCP or Unit Datagram Protocol (UDP) bandwidth. By default, Iperf uses port 5001 and 10 sec tests time periods. In our experiment we used 20 sec test time periods for more consistency. Iperf can measure the maximum amount of data transmitted between any two hosts at any given time. For Iperf to work correctly it needs to be installed on two hosts one act as Iperf client and the other act as Iperf server.

In this study, Iperf was installed on Windows 10, Windows 8 and Ubuntu 16.04. Windows 8 and Ubuntu 16.04 are defined as Iperf servers and Windows 10 is defined as Iperf client. Thus, TCP Throughput was measured between Windows 10 and Windows 8 and then it was measured between Windows 10 and Ubuntu 16.04.

Ping is a network utility used to test the reachability of a node within IP networks. It measures the RTT for packets sent from a source node to destination node. The name of Ping comes from active sonar terminology that sends a pulse of sound and listens for the echo to detect objects under water.

Ping operates by sending ICMP/ICMPv6 echo request messages to the target node and waiting for an ICMP/ICMPv6 echo reply messages. The Ping utility program reports errors, packet loss and a statistical summary of the packets journey. Typically including the minimum, maximum, the mean round-trip times and standard deviation of the mean for the packets sent.

Table 2: Attacks commands

Attack name	Command
RS Flooding	atk6-flood_rs6 [-sS] interface [target]
RA Flooding	atk6-flood_router6 <interface>
NS Flooding	atk6-flood_solicit6 <interface> [target-ip]
NA Flooding	atk6-flood_advertise6 <interface> [target-ip]
NS/NA Spoofing	atk6-parasite6 <interface> [fake-mac]
DAD DoS	atk6-dos-new-ip6 <interface>
Malicious Last Hop Router DoS	atk6-fake_router6
Default Router is Killed DoS	atk6-kill_router6 <interface> <target-ip>
Good Router Goes Bad DoS	atk6-dump_router6 <interface>
Spoofed Redirect Message DoS	atk6-redirsniff6 <interface> <victim-ip> <destination -ip> <original-router> [<new-router> [new-router-mac]]

Table 3: Experiment legends

Legend	Description
WBA	Windows Before Attack.
LBA	Linux Before Attack.
LDA	Linux During Attack.
WDA	Windows During Attack.

Table 4: Computers roles, software and hardware specifications

Node role	Operating system	IP address	MAC address	Software	Hardware
Monitoring	Windows 10	FE80::1	A4:1F:72:5B:73:A2	Iperf. Wireshark.	Intel Pentium G645 2.90GHz processor. 2.00 GB RAM Memory.
Attacker	Kali Linux 3.20.2	FE80::4	00:1E:33:3A:D3:9D	THC-IPV6.	Intel Pentium Dual T2390 1.86GHz processor. 2.00 GB RAM Memory.
Victim	Ubuntu 16.04	FE80::3	00:1D:92:01:06:F4	Iperf.	Intel Core 2 Duo E4500 2.20GHz. 2.00 GB RAM Memory.
Victim	Windows 8	FE80::2	74:27:EA:0D:89:10	Iperf.	Intel Core i5 3.00GHz Processor. 4.00 GB RAM Memory.

In our experiment, Ping measured RTT between monitoring computer and victims' computers. Ping was installed by default on Windows 10, which connected to the Windows 8 and Ubuntu 16.04 victims' computers to measure the RTT. We test the RTT 30 times between the monitor computer and victims' computers for more consistency.

A built in tool, resource monitor, came bundled with Windows operating systems families. It allows the users to see processor utilization, hard disk, network and memory usage. For Linux based systems the same tool did exist under the name system monitor.

In our experiment, resource monitor and system monitor are used to monitor the computer's processor usage on Windows 8 and Ubuntu 16.04 respectively for a period of 60 sec. Table 2 shows the NDP attacks and corresponding commands to execute it. In Table 3 we show the legends used for generating the graphs. Table 4 illustrates the software and hardware specifications of the joint nodes and the role of each node as well.

Results

TCP Throughput

RA flood attack and NS/NA spoofing attack caused the Windows 8 and Ubuntu 16.04 TCP Throughput to drop from around 1400 and 1700 MBps respectively to almost 0 MBps. Legitimate packets could not be transmitted during these two attacks. During NA, NS and RS flooding attacks throughput dropped from 1400 MBps to just few MBps for Windows 8 while for Ubuntu 16.04 the throughput were dropped slightly compared to Windows 8. Thus, legitimate packets could be transmitted at lower rates on both operating systems during NA, NS and RS flooding attacks. The detailed rates of the TCP Throughput before and during NS, NA, RS, RA flooding attacks and NS/NA spoofing attack are shown in Fig. 8 to Fig. 12.

CPU Utilization

CPU utilization was expected to increase on both Windows 8 and Ubuntu 16.04. However, the CPU

utilization on Ubuntu 16.04 did not show any significant changes before and during the NS, NA, RS and RA flooding attacks. On the contrary, during RA flooding Windows 8 CPU utilization reached 100% rapidly and dropped to around 37%. During RS and NA flooding attacks CPU utilization only increased from almost 3% to 28%, while for NS flooding attack it reach to 40% and then dropped to 30%. The detailed rates of the CPU utilization percentage before and during NS, NA, RS and RA flooding attacks are shown in Fig. 13 to 16.

Round Trip Time

NS flooding attack and NS/NA spoofing attack make Windows 8 and Ubuntu 16.04 operating systems RTT to increase significantly, the packets during these two attacks are totally lost. During NA and RS flooding

attacks RTT results for Ubuntu 16.04 were even not changed from normal. While for Windows 8 RTT results were considerably high during the attacks compared to the RTT during the normal operations. For RA flooding attack both operating systems were considerably has higher RTT results compare to normal status. The detailed rates of the RTT before and during NS, NA, RS, RA flooding attacks and NS/NA spoofing attack are shown in Fig. 17 to 21.

Packets Drop Ratio

Figure 22 to 26, a the packets drop ratio is presented.

In most of the attacks results shown better resistance for Linux operating systems when compare to Windows operating system. Only in RA flooding attack both operating systems have almost the same influence ratio.

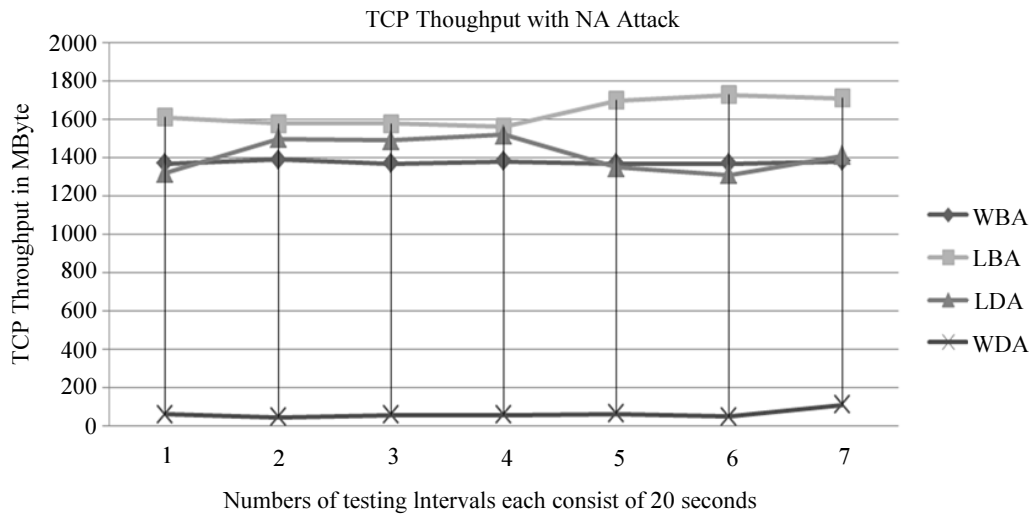


Fig. 8: TCP Throughput before and during NA flooding attack

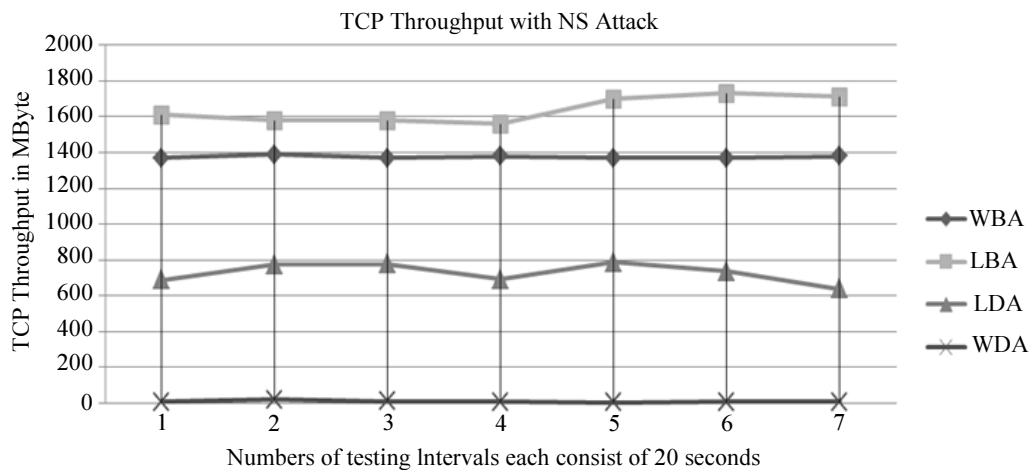


Fig. 9: TCP throughput before and during NS flooding attack

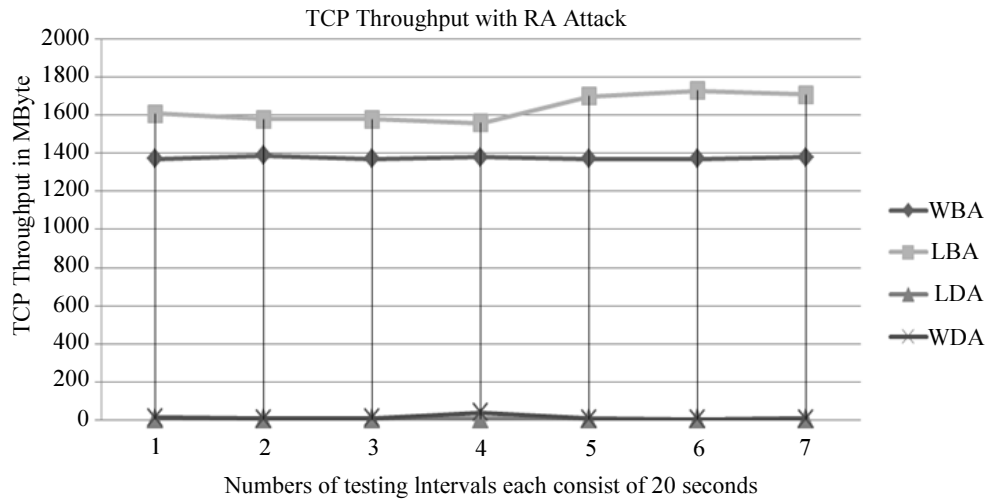


Fig. 10; TCP throughput before and during RA flooding attack

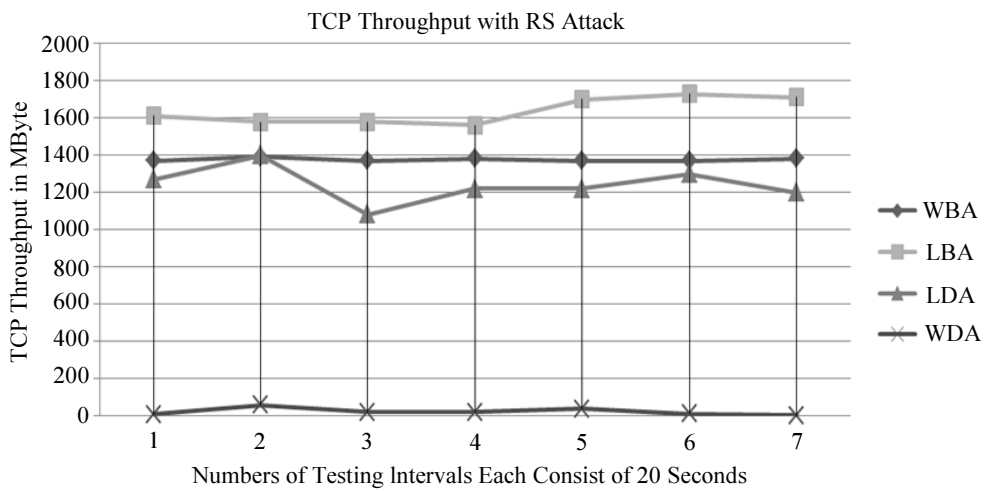


Fig. 11: TCP throughput before and during RS flooding attack

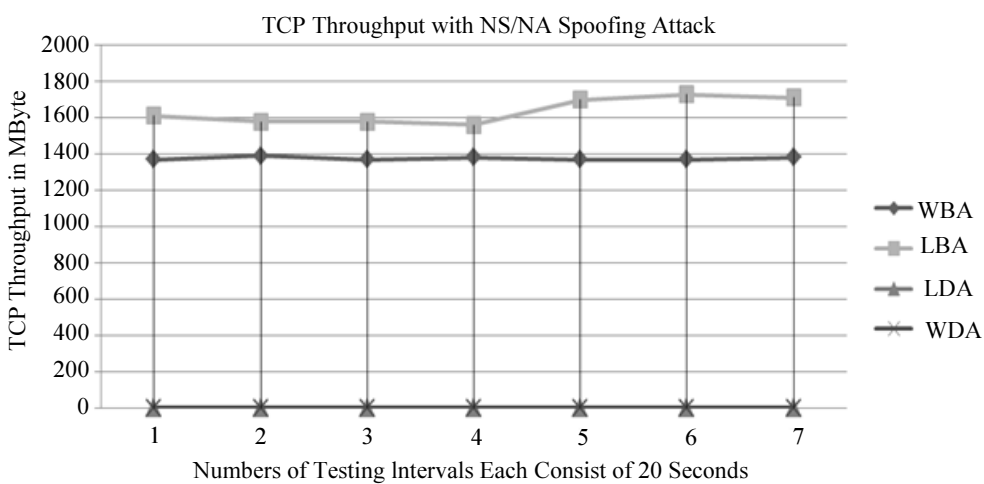


Fig. 12: TCP throughput before and during NS/NA spoofing attack

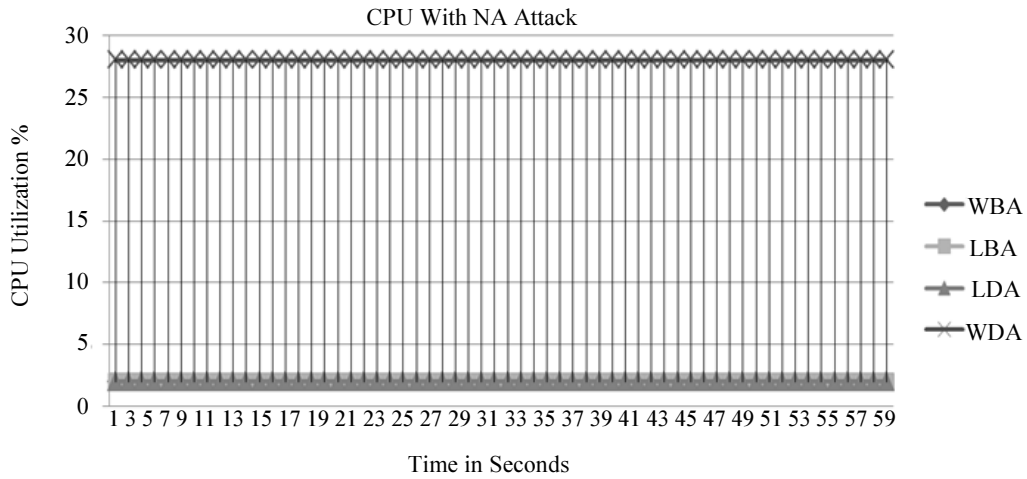


Fig. 13: CPU utilizations before and during NA flooding attack

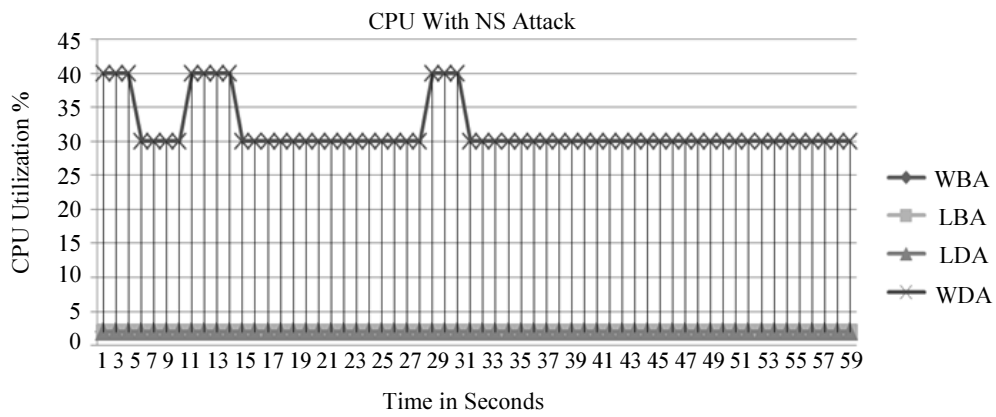


Fig. 14: CPU utilizations before and during NS flooding attack

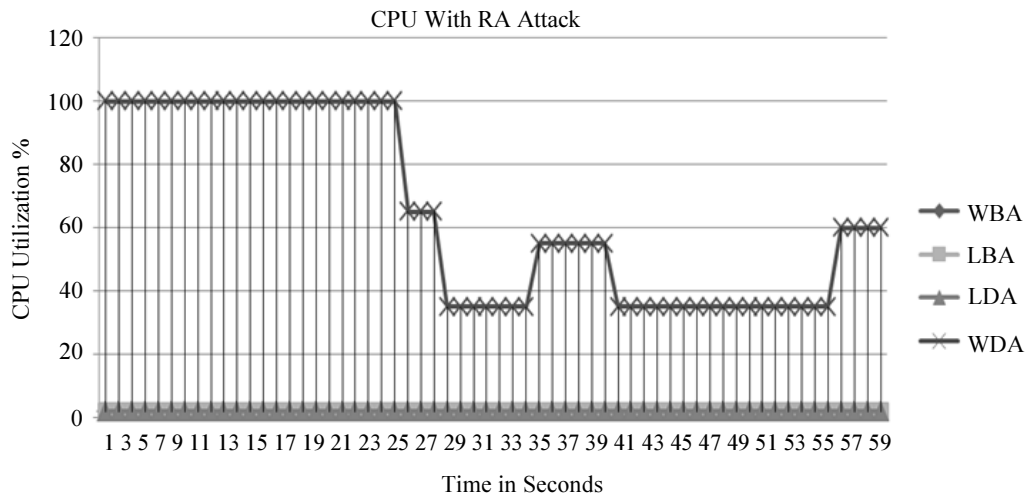


Fig. 15: CPU utilizations before and during RA flooding attack

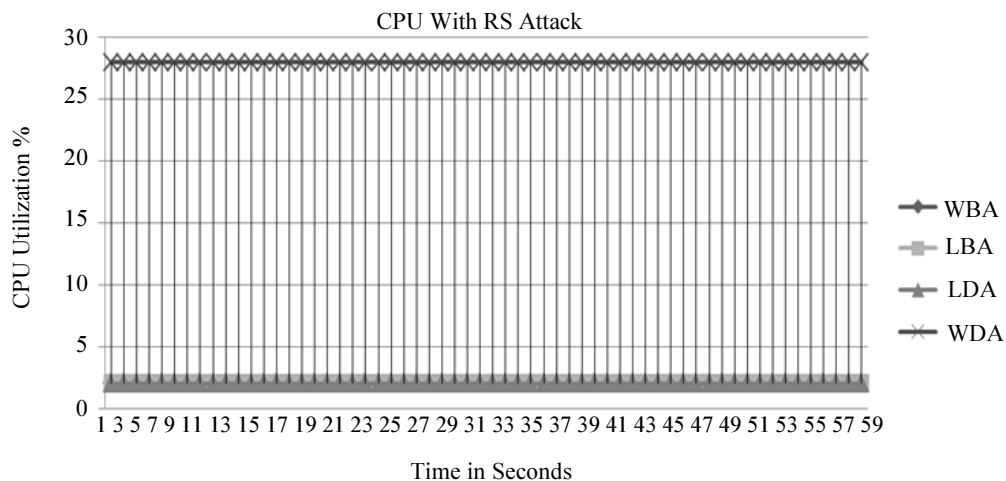


Fig. 16: CPU utilizations before and during RS flooding attack

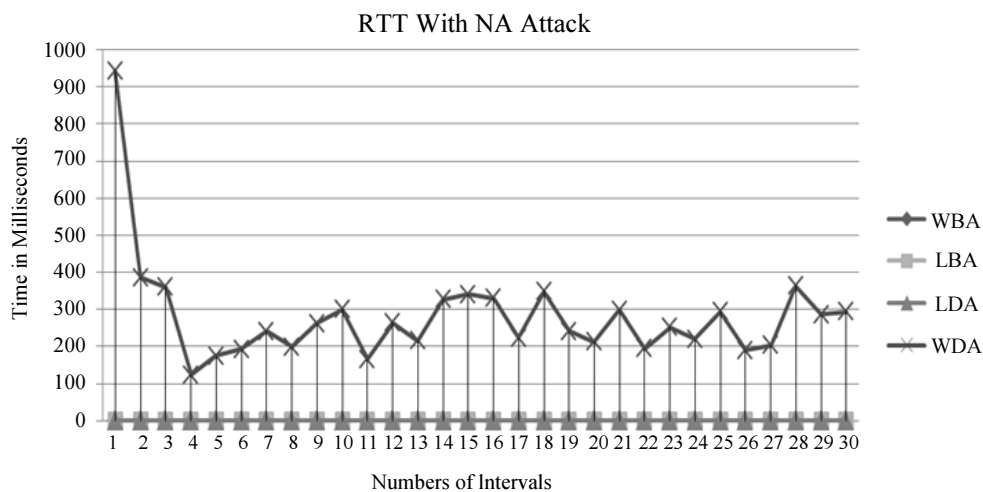


Fig. 17: Packet delay before and during NA flooding attack

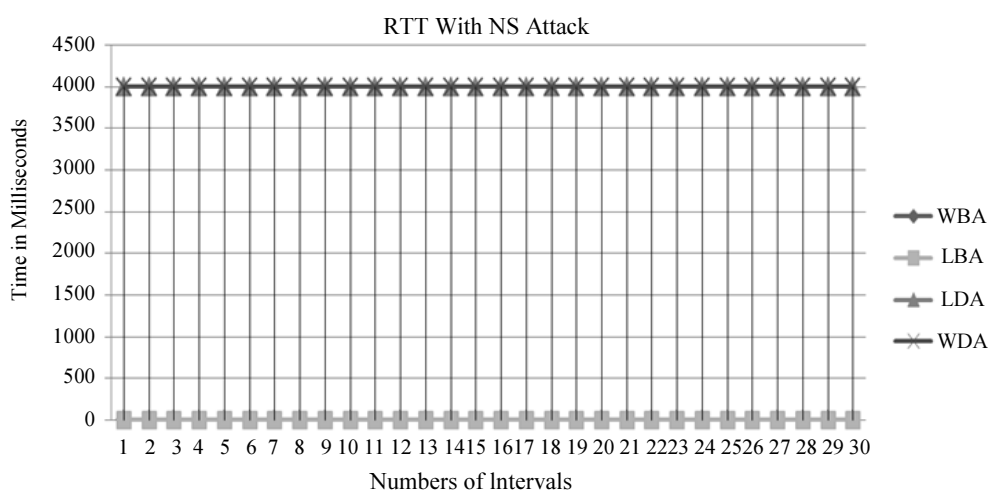


Fig. 18: Packet delay before and during NS flooding attack

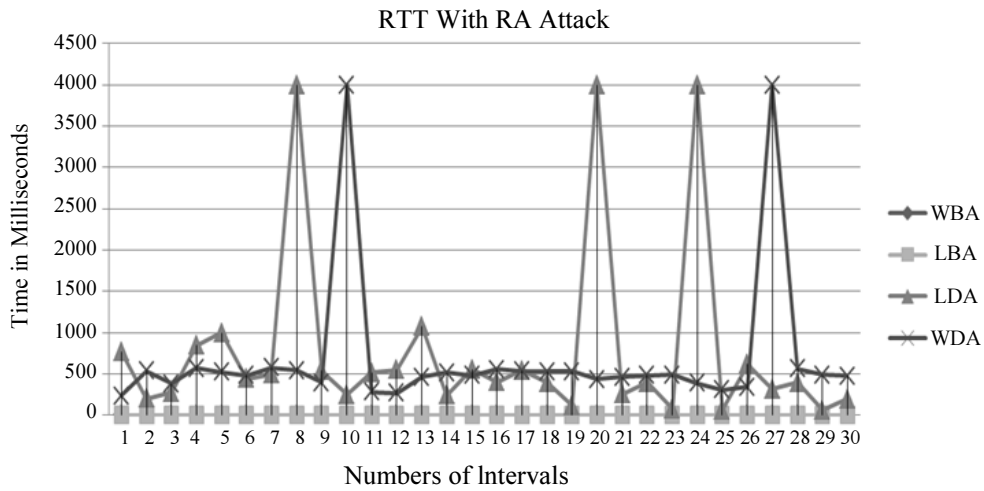


Fig. 19: Packet delay before and during RA flooding attack

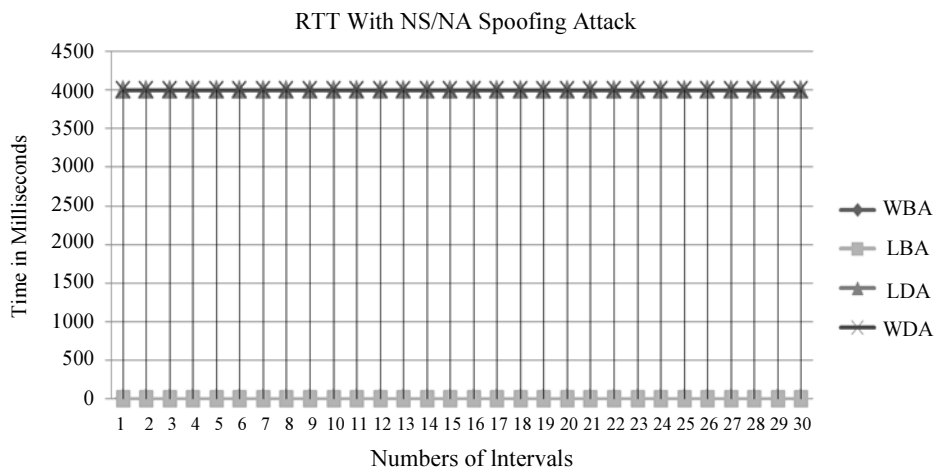


Fig. 20: Packet delay before and during NS/NA spoofing attack

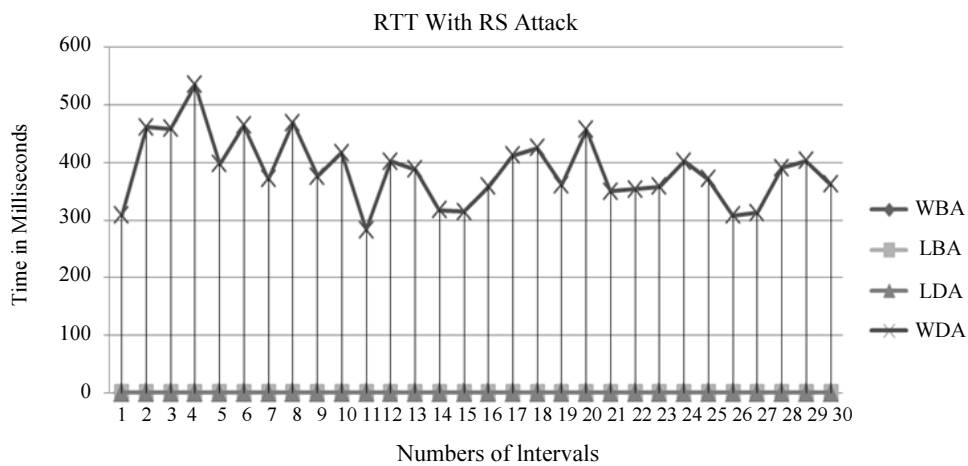


Fig. 21: Packet delay before and during RS flooding attack

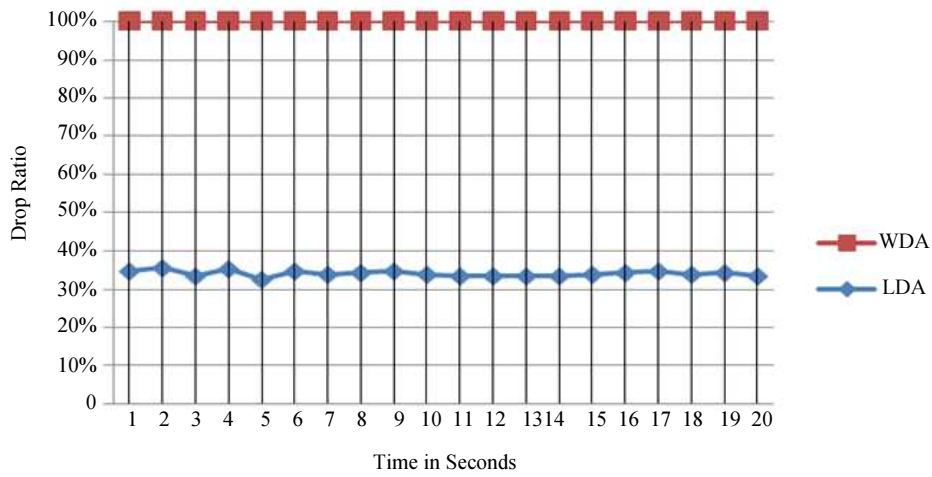


Fig. 22: Packet drop ratio during NS flooding attack

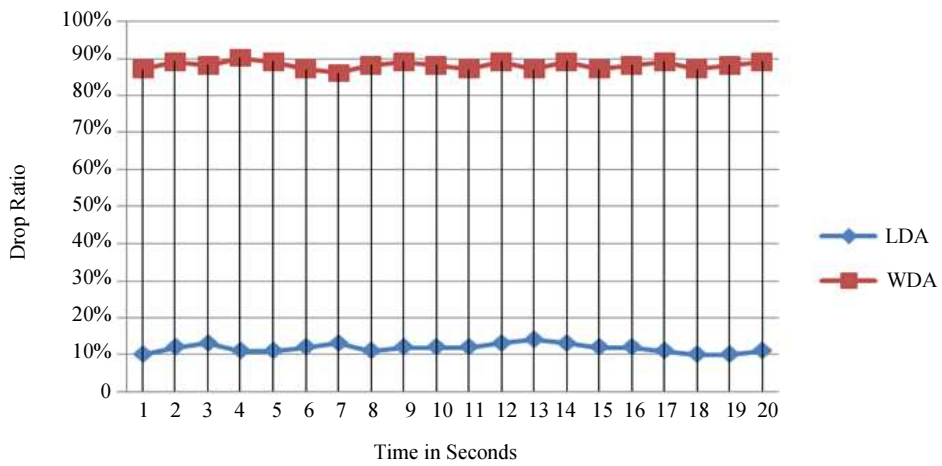


Fig. 23: Packet drop ratio during NA flooding attack

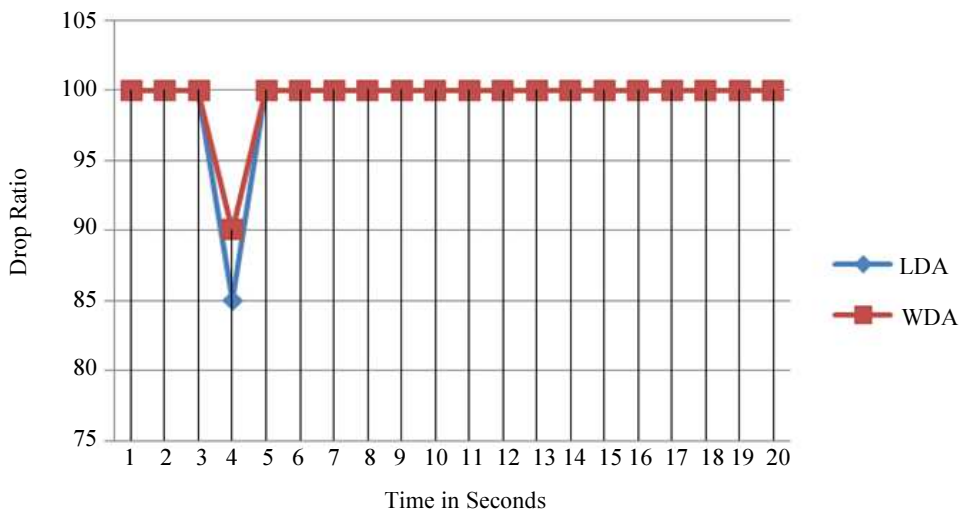


Fig. 24: Packet drop ratio during RA flooding attack

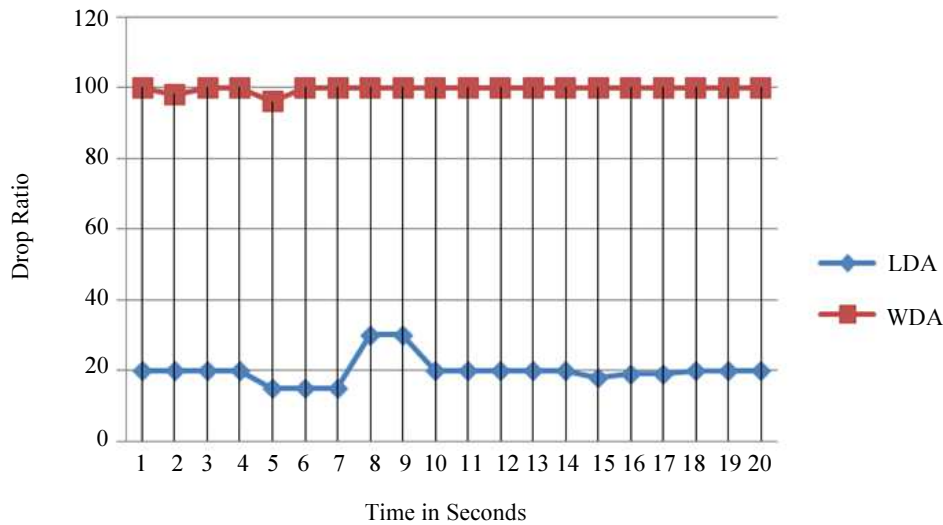


Fig. 25: Packet drop ratio during RS flooding attack

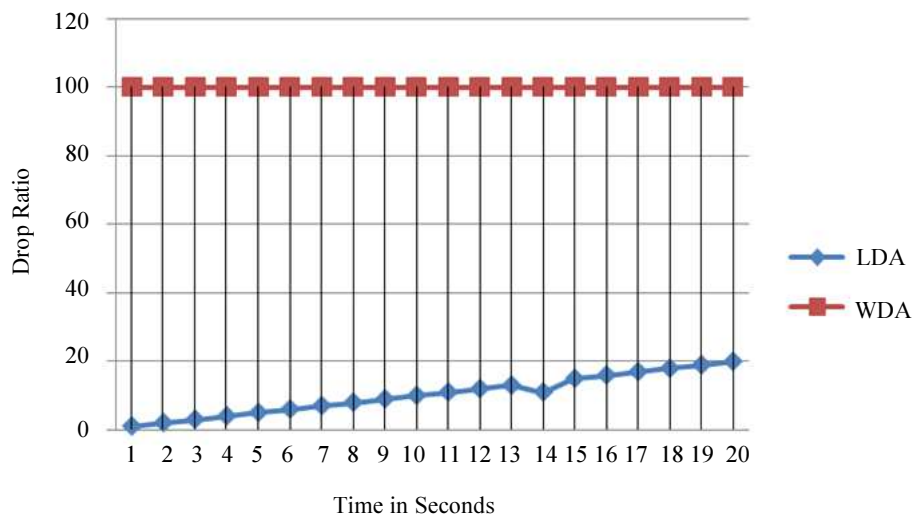


Fig. 26: Packet drop ratio during NS/NA spoofing attack

Note that for some types of NDP attacks, such as DAD DoS attack, there is no tangible performance metric that we can use to evaluate the impacts of the attacks. For such type of attacks, non-tangible, we used Wireshark to capture the frames and analyze it according to the contents it has.

Figure 27 shown a normal ICMPv6 echo request packet from FE80::1 to FE80::3. As we seen in the frame details the echo request has been responded by echo replay message in the following frame, number 366. We then run the NS/NA spoofing attack on the attacker machine, as per Fig. 28, which start listening to the communication link and waiting for the victim to send NS messages in order to spoof IP addresses. Again we sent ICMPv6 echo request packet from FE80::1 to

FE80::3. As per Fig. 29 illustrates, the echo request packet between the monitoring computer and Ubuntu 16.04 never been respondent. The attacker send a spoofed ICMPv6 echo replay packet in response to ICMPv6 echo request packet sent by the victim, using victim's IP address FE80::3 but with attacker's own MAC address 00:1E:33:3A:D3:9D. As a result the victim will never get replied because all the packets will be diverted to the attacker's machine. For spoofed redirect message DoS attack the scenario will be the same, as both attacks based on spoofing victim's IP address.

For Default Router is Killed DoS attack, the attacker killing the router by setting router life time to zero. It sends a spoofed RA message to all nodes multicast address, pretending to be the router,

FE80::5, as per Fig. 30 and 31. In Fig. 32 we shown a packet of RS message during executing good router goes bad DoS attack, the attacker solliciting about existing router addresses in order to compromise

them. For malicious last hop router and good router goes bad DoS attacks the scenario will be the same, because three attacks are based on reducing the router life time using spoofed RA message.

```

> Frame 365: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0
> Ethernet II, Src: Dell_5b:73:a2 (a4:1f:72:5b:73:a2), Dst: Micro-St_01:06:f4 (00:1d:92:01:06:f4)
v Internet Protocol Version 6, Src: fe80::1, Dst: fe80::3
    0110 .... = Version: 6
    > .... 0000 0000 .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... .... 0000 0000 0000 0000 = Flow label: 0x000000
    Payload length: 40
    Next header: ICMPv6 (58)
    Hop limit: 128
    Source: fe80::1
    Destination: fe80::3
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
v Internet Control Message Protocol v6
    Type: Echo (ping) request (128)
    Code: 0
    Checksum: 0xc921 [correct]
    [Checksum Status: Good]
    Identifier: 0x0001
    Sequence: 3793
    [Response In: 366]
> Data (32 bytes)
    
```

Fig. 27: Normal ICMPv6 echo request packet from FE80::1 to FE80::3

```

^Croot@kali:~# atk6-parasite6 eth0
Remember to enable routing, you will denial service otherwise:
=> echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
Remember to prevent sending out ICMPv6 Redirect packets:
=> iptables -I OUTPUT -p icmpv6 --icmpv6-type redirect -j DROP
Started ICMP6 Neighbor Sollicitation Interceptor (Press Control-C to end) ...
Spoofed packet to fe80::1 as fe80::3
Spoofed packet to fe80::1 as fe80::3
Spoofed packet to fe80::1 as fe80::3
Spoofed packet to fe80::1 as fe80::3
Spoofed packet to fe80::1 as fe80::3
Spoofed packet to fe80::1 as fe80::3
Spoofed packet to fe80::1 as fe80::3
    
```

Fig. 28: NS/NA spoofing attack replies to FE80::1 on attacker machine

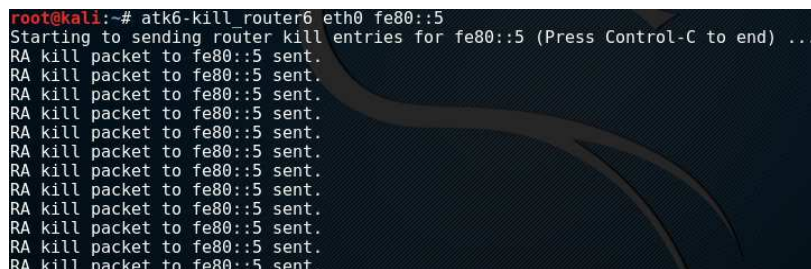
```

> Frame 1810: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0
> Ethernet II, Src: Dell_5b:73:a2 (a4:1f:72:5b:73:a2), Dst: Inventec_3a:d3:9d (00:1e:33:3a:d3:9d)
v Internet Protocol Version 6, Src: fe80::1, Dst: fe80::3
    0110 .... = Version: 6
    > .... 0000 0000 .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... .... 0000 0000 0000 0000 = Flow label: 0x000000
    Payload length: 40
    Next header: ICMPv6 (58)
    Hop limit: 128
    Source: fe80::1
    Destination: fe80::3
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
v Internet Control Message Protocol v6
    Type: Echo (ping) request (128)
    Code: 0
    Checksum: 0xd7b0 [correct]
    [Checksum Status: Good]
    Identifier: 0x0001
    Sequence: 66
    > [No response seen]
> Data (32 bytes)
    
```

Fig. 29: ICMPv6 echo request packet from FE80::1 to FE80::3 during NS/NA spoofing attack

```
> Frame 394: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: Inventec_3a:d3:9d (00:1e:33:3a:d3:9d), Dst: IPv6mcast_01 (33:33:00:00:00:01)
v Internet Protocol Version 6, Src: fe80::5, Dst: ff02::1
  0110 .... = Version: 6
  > .... 1110 0000 .... .... .... = Traffic class: 0xe0 (DSCP: CS7, ECN: Not-ECT)
  .... .... 0000 0000 0000 0000 0000 = Flow label: 0x000000
  Payload length: 16
  Next header: ICMPv6 (58)
  Hop limit: 255
  Source: fe80::5
  Destination: ff02::1
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
v Internet Control Message Protocol v6
  Type: Router Advertisement (134)
  Code: 0
  Checksum: 0x3c23 [correct]
  [Checksum Status: Good]
  Cur hop limit: 64
  > Flags: 0x08
  Router lifetime (s): 0
  Reachable time (ms): 0
  Retrans timer (ms): 0
```

Fig. 30: Default router is killed DoS packet from FE80::4 to all nodes multicast address



```
root@kali:~# atk6-kill router6 eth0 fe80::5
Starting to sending router kill entries for fe80::5 (Press Control-C to end) ...
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
RA kill packet to fe80::5 sent.
```

Fig. 31: Default router is killed DoS attack on attacker node

```
> Frame 71: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: Inventec_3a:d3:9d (00:1e:33:3a:d3:9d), Dst: IPv6mcast_02 (33:33:00:00:00:02)
v Internet Protocol Version 6, Src: fe80::4, Dst: ff02::2
  0110 .... = Version: 6
  > .... 1110 0000 .... .... .... = Traffic class: 0xe0 (DSCP: CS7, ECN: Not-ECT)
  .... .... 0000 0000 0000 0000 0000 = Flow label: 0x000000
  Payload length: 16
  Next header: ICMPv6 (58)
  Hop limit: 255
  Source: fe80::4
  Destination: ff02::2
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
v Internet Control Message Protocol v6
  Type: Router Solicitation (133)
  Code: 0
  Checksum: 0x7534 [correct]
  [Checksum Status: Good]
  Reserved: 00000000
v ICMPv6 Option (Source link-layer address : 00:1e:33:3a:d3:9d)
  Type: Source link-layer address (1)
  Length: 1 (8 bytes)
  Link-layer address: Inventec_3a:d3:9d (00:1e:33:3a:d3:9d)
```

Fig. 32: Dump router packet from FE80::4 to all routers multicast address

For DAD DoS attack we run the attack in attacker's machine and then we try to connect new nodes to the link, Windows 8 and Ubuntu 16.04 respectively. As we seen in Fig. 33 to 37, Windows 8 try ten times to gain an IP address before it quit DAD operations while for

Ubuntu 16.04 it try three times. This is because of they have different DAD procedures programming within their kernel's IP stack. We arrange frames based on protocol type, which is ICMPv6, to easily trace NS and NA messages during the attack.

Table 5: Dos attacks impacts ranks

Attack Type	Windows	Linux
RS Flooding	Extremely High	Moderate
RA Flooding	Catastrophic	Catastrophic
NS Flooding	Extremely High	High
NA Flooding	Extremely High	Low
NS/NA Spoofing	Catastrophic	Catastrophic
DAD DoS	Catastrophic	Catastrophic
Malicious Last Hop Router DoS	Catastrophic	Catastrophic
Default Router is Killed DoS	Catastrophic	Catastrophic
Good Router Goes Bad DoS	Catastrophic	Catastrophic
Spoofed Redirect Message DoS	Catastrophic	Catastrophic

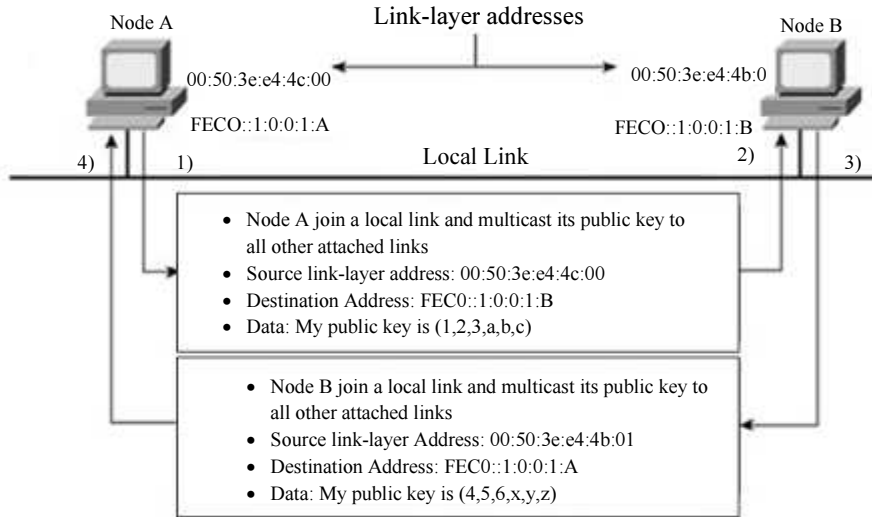


Fig. 33: Packets Flow between FE80::2 to FE80::4 During DAD DoS Attack

No.	Time	Source	Destination	Protocol	Length	Info
14	9.787136	fe80::d598:d598:94ff:961e	ff02::2	ICMPv6	62	Router Solicitation
172	13.786863	fe80::2558:1c0a:696a:fa52	ff02::2	ICMPv6	62	Router Solicitation
13	9.787135	::	ff02::1:ffff:961e	ICMPv6	78	Neighbor Solicitation for fe80::d598:d598:94ff:961e
37	10.286941	::	ff02::1:fff9:c85b	ICMPv6	78	Neighbor Solicitation for fe80::50ce:b6b7:df9c:b85b
64	10.787120	::	ff02::1:fff7:c87f	ICMPv6	78	Neighbor Solicitation for fe80::1d00:ea6:2f7c:87bf
87	11.286917	::	ff02::1:fff9:5102	ICMPv6	78	Neighbor Solicitation for fe80::5803:c4d7:af39:5102
96	11.787055	::	ff02::1:fff7:29f0	ICMPv6	78	Neighbor Solicitation for fe80::4048:c4d2:4879:29f0
104	12.286886	::	ff02::1:ffbd:19a1	ICMPv6	78	Neighbor Solicitation for fe80::9d9f:4208:b1bd:19a1
115	12.787096	::	ff02::1:fff8:2414	ICMPv6	78	Neighbor Solicitation for fe80::304d:21c0:8b8f:2414
156	13.287158	::	ff02::1:ffbb:f142	ICMPv6	78	Neighbor Solicitation for fe80::196e:3156:32bb:f142
171	13.786861	::	ff02::1:ff6a:fa52	ICMPv6	78	Neighbor Solicitation for fe80::2558:1c0a:696a:fa52
195	14.287055	::	ff02::1:fff5:f12b	ICMPv6	78	Neighbor Solicitation for fe80::2833:7b7:99f5:f12b
16	9.787409	fe80::d598:d598:94ff:961e	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::d598:d598:94ff:961e (ovr) is at 00:1e:62:0e:e9:04
18	9.788358	fe80::d598:d598:94ff:961e	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::d598:d598:94ff:961e (ovr) is at 00:1e:62:0e:e9:04
39	10.287441	fe80::50ce:b6b7:df9c:b85b	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::50ce:b6b7:df9c:b85b (ovr) is at 00:1e:14:02:bc:c2
41	10.288149	fe80::50ce:b6b7:df9c:b85b	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::50ce:b6b7:df9c:b85b (ovr) is at 00:1e:14:02:bc:c2
66	10.787408	fe80::1d00:ea6:2f7c:87bf	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::1d00:ea6:2f7c:87bf (ovr) is at 00:1e:28:32:98:33
68	10.788145	fe80::1d00:ea6:2f7c:87bf	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::1d00:ea6:2f7c:87bf (ovr) is at 00:1e:28:32:98:33
89	11.287190	fe80::5803:c4d7:af39:5102	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::5803:c4d7:af39:5102 (ovr) is at 00:1e:87:33:ff:14
91	11.288109	fe80::5803:c4d7:af39:5102	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::5803:c4d7:af39:5102 (ovr) is at 00:1e:87:33:ff:14
98	11.787316	fe80::4048:c4d2:4879:29f0	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::4048:c4d2:4879:29f0 (ovr) is at 00:1e:00:cf:5e:c8
100	11.788211	fe80::4048:c4d2:4879:29f0	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::4048:c4d2:4879:29f0 (ovr) is at 00:1e:00:cf:5e:c8
106	12.287149	fe80::9d9f:4208:b1bd:19a1	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::9d9f:4208:b1bd:19a1 (ovr) is at 00:1e:6d:a6:9c:0e
108	12.287984	fe80::9d9f:4208:b1bd:19a1	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::9d9f:4208:b1bd:19a1 (ovr) is at 00:1e:6d:a6:9c:0e
117	12.787378	fe80::304d:21c0:8b8f:2414	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::304d:21c0:8b8f:2414 (ovr) is at 00:1e:03:e0:52:c4
123	12.788359	fe80::304d:21c0:8b8f:2414	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::304d:21c0:8b8f:2414 (ovr) is at 00:1e:03:e0:52:c4
158	13.287442	fe80::196e:3156:32bb:f142	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::196e:3156:32bb:f142 (ovr) is at 00:1e:4c:7f:40:af
160	13.288145	fe80::196e:3156:32bb:f142	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::196e:3156:32bb:f142 (ovr) is at 00:1e:4c:7f:40:af
174	13.787167	fe80::2558:1c0a:696a:fa52	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::2558:1c0a:696a:fa52 (ovr) is at 00:1e:8d:29:b3:a1
176	13.788056	fe80::2558:1c0a:696a:fa52	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::2558:1c0a:696a:fa52 (ovr) is at 00:1e:8d:29:b3:a1
197	14.289877	fe80::2833:7b7:99f5:f12b	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::2833:7b7:99f5:f12b (ovr) is at 00:1e:2b:6f:64:54
198	14.289877	fe80::2833:7b7:99f5:f12b	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::2833:7b7:99f5:f12b (ovr) is at 00:1e:2b:6f:64:54

Fig. 34: Packets Flow between FE80::2 to FE80::4 During DAD DoS Attack

No.	Time	Source	Destination	Protocol	Length	Info
86	10.657807	::	ff02::1:ff58:e5e9	ICMPv6	78	Neighbor Solicitation for fe80::c5a1:37bc:4d58:e5e9
101	11.245197	::	ff02::1:ff51:eece	ICMPv6	78	Neighbor Solicitation for fe80::9ae4:2c91:8051:eece
113	11.685261	::	ff02::1:ffa2:4ffd	ICMPv6	78	Neighbor Solicitation for fe80::3ba6:551e:18a2:4ffd
89	10.659054	fe80::c5a1:37bc:4d58:e5e9	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::c5a1:37bc:4d58:e5e9 (ovr) is at 00:1e:27:21:67:e2
88	10.658449	fe80::c5a1:37bc:4d58:e5e9	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::c5a1:37bc:4d58:e5e9 (ovr) is at 00:1e:27:21:67:e2
105	11.246604	fe80::9ae4:2c91:8051:eece	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::9ae4:2c91:8051:eece (ovr) is at 00:1e:33:f2:59:5a
102	11.245510	fe80::9ae4:2c91:8051:eece	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::9ae4:2c91:8051:eece (ovr) is at 00:1e:33:f2:59:5a
115	11.686371	fe80::3ba6:551e:18a2:4ffd	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::3ba6:551e:18a2:4ffd (ovr) is at 00:1e:7e:bf:7d:55
114	11.685533	fe80::3ba6:551e:18a2:4ffd	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::3ba6:551e:18a2:4ffd (ovr) is at 00:1e:7e:bf:7d:55

Fig. 35: Packets flow between FE80::2 to FE80::4 during DAD DoS attack

```
root@kali:~# atk6-dos-new-ip6 eth0
Started ICMP6 DAD Denial-of-Service (Press Control-C to end) ...
Spoofed packet for existing ip6 as fe80::dcc3:d598:94ff:961e
Spoofed packet for existing ip6 as fe80::50ce:b6b7:df9c:b85b
Spoofed packet for existing ip6 as fe80::1d00:ea6:2f7c:87bf
Spoofed packet for existing ip6 as fe80::5803:c4d7:af39:5102
Spoofed packet for existing ip6 as fe80::4048:c4d2:4879:29f0
Spoofed packet for existing ip6 as fe80::9df8:4208:b1bd:19a1
Spoofed packet for existing ip6 as fe80::304d:21c0:8b8f:2414
Spoofed packet for existing ip6 as fe80::196e:3156:32bb:f142
Spoofed packet for existing ip6 as fe80::2558:1c0a:696a:fa52
Spoofed packet for existing ip6 as fe80::2833:7b7:93f5:f12b
```

Fig. 36: DAD DoS attack replies to FE80::2 on attacker's machine

```
root@kali:~# atk6-dos-new-ip6 eth0
Started ICMP6 DAD Denial-of-Service (Press Control-C to end) ...
Spoofed packet for existing ip6 as fe80::c5a1:37bc:4d58:e5e9
Spoofed packet for existing ip6 as fe80::9ae4:2c91:8051:eece
Spoofed packet for existing ip6 as fe80::3ba6:551e:18a2:4ffd
```

Fig. 37: DAD DoS attack replies to FE80::3 on attacker's machine

A summary of the DoS attacks impacts for both operating systems, Windows and Linux, is presented in Table 5.

Existing Solutions

Some other sophisticated attacks, that are the combination of one or two of the mentioned earlier attacks, could be used to exploit NDP vulnerabilities during SLAAC or ND procedures. The name of the attack usually is given based on the type of the NDP messages utilized for executing that attack. From the above, we conclude that all attacks rely on the spoofing or abusing of the NDP message. If there is a perfect authentication mechanism to verify the NDP messages, this protocol can be protected comprehensively and have strong resistibility to various attacks. Many works to secure NDP are making efforts toward this direction and several related works will be talked about next.

According to IETF two types of solutions have been introduced to protect NDP, which are Internet Protocol Security (IPSec) and Secure Neighbor Discovery (SEND), as we will explain in the sections below.

Internet Protocol Security (IPSec)

IPSec is used to ensure that the IP packets between the IP layer and the transport layer remain confidential

and accurate. This protocol comprises the Authentication Header (AH) protocol, the Encapsulation Security Payload (ESP) protocol and the Internet Key Exchange (IKE) protocol. The AH protocol mainly keeps transmitted packets private and accurate. The ESP protocol ensures the authenticity of the origin in the encryption process. The IKE protocol uses a Diffie-Hellman key exchange mechanism to prepare Security Associations (SA) for IPSec communication. The two modes of IPSec, namely the transport mode and the tunnel mode, enables users to implement IPSec even under various network environments. IPSec under the transport mode protects the information being delivered from the transport layer to the network layer. On the contrary, IPSec under the tunnel mode protects entire IP packets. The original specifications of NDP recommend using IPSec in ensuring the protection of NDP messages even while the details and associated limitations have yet to be explained (Nikander, 2001). NDP intended to use IPSec to protect itself through IP layer authentication, but IPSec is not suited for the auto-configuration in SLAAC as there is a bootstrapping problem existed. There have been several proposals for IPSec protocol, regarding keys distribution and amongst them the IKE is considered the standard (Aiello *et al.*, 2002; Blaze, 2001). For this application, IPSec is compatible with manual keying whereas the currently standardized IKE

key management protocols may not be deployed considering NDP using multicasts which are not supported through IKE. Consequently, a chicken-and-egg problem (Arkko, 2002) is raised in using IKE (Harkins and Carrel, 1998) prior to ND being considered operational. Regardless of manual keying could be utilized for neighbor discovery, the number of SAs required will be truly extensive (Arkko *et al.*, 2002) (Chiu and Gamess, 2010). More importantly, the utilization of symmetric security doesn't prevent verified nodes from start masquerading as routers for different hosts, provided that multicast is utilized. Finally, the absence of nitty gritty information, to RFC 2461, around how should set up the fundamental SAs makes a trouble to administrators and might be a breaking point of interoperability. Proposals to utilize IPSec and make it workable for securing NDP are introduced in (Liu and Dai, 2013; Kim *et al.*, 2008).

Secure Neighbor Discovery (SEND)

SEND is developed by the IETF to specify security mechanisms for NDP. SEND proposed three mechanisms to protect NDP messages. The first is router authorization, SEND uses Authorization Delegation Discovery (ADD) procedure to validate and authorize the IPv6 routers.

Proposed Solution

A proposed solution have been introduced in (Ahmed *et al.*, 2014). In IPv6 Neighbor Discovery Protocol, an attacking node can cause packets for legitimate nodes, both hosts and routers, to be sent to some other link-layer address. This can be done by either sending a Neighbor Solicitation (NS) with a spoofed source link-layer address, or sending a Neighbor Advertisement (NA) with a spoofed target link-layer address. If the spoofed link-layer address is a valid one, packets will continue to be redirected, this is also lead to Man-in-The-Middle attack. The other part of the attack is Neighbor Discovery DoS attack in this attack; the attacking node fabricates addresses with the subnet prefix of the target network and continuously sends packets to them. The last hop router is obligated to resolve the addresses with the Neighbor Discovery protocol. A legitimate host attempting to enter the network may be unable to obtain Neighbor Discovery service from the last hop router as the router is already busy with resolving the bogus addresses. The proposed mechanism is a cryptographic based solution. It is working according to the digital signature procedure. The nodes (Router/Hosts) will advertise their public keys once they are joined a local link to all other attached link in the network in a form of multicast message. Nodes will update their cash values with the new entries, now the nodes have each other public keys. In future any

nodes receiving a message from another node will decrypt it with the sender public key they already have. If the message is spoofed one the nodes will detect this because the accompanied private key of the sender inside the message will mismatch with the sender public key that the receiver already have, the receiver will drop the message. **Algorithm 1** shows the steps for the proposed mechanism and Fig. 33 representing the logical diagram of the proposed mechanism.

Algorithm 1

```
A, B network nodes;  
A: Join a local link;  
A: Multicast its public key;  
B: Join a local link;  
A: Multicast its public key;  
A, B Update their cache with public keys new entries;  
A, B Exchange messages according to their private keys  
and new entries;  
IF  
A send B and the keys are not matched;  
THEN  
Drop the packets;  
Else  
IF  
B send A and the keys are not matched;  
Then  
Drop the packets;  
Else  
Receive the packets;
```

This is based on a trusted third party, called trust anchor, to issue the certifications. Only after the router is authorized it can act as a router and every node must certify the router via the trust anchor before setting the router as a default router. The second mechanism provided by SEND is Cryptographically Generated Addresses (CGA). A node cryptographically generates IPv6 address by using a one-way hash function from the node's public key and some other parameters. CGA is used to make sure that the sender of NDP packets is the owner of the claimed address. The third mechanism used by SEND aims to protect the integrity of the messages and authenticate the identity of their sender.

In order to activate these three mechanisms SEND introduces four NDP options which are CGA option to prevent IPv6 address stealing, nonce and timestamp option to protect NDP from replay attack and RSA signature option to do authentication. The main problem on SEND is the complexity on the address generation, CGA option generation and the signing of the RSA signature option (AlSa'deh and Meinel, 2012; An *et al.*, 2007). Moreover SEND was only implement by a very few number of operation systems and network devices. In addition, it is also vulnerable to DoS attack that could

exploit the SEND messages. Attacker may send more packets with the four NDP options to force the victim to

process it. Moreover, the new options add more than one Kbyte to each NDP packet().

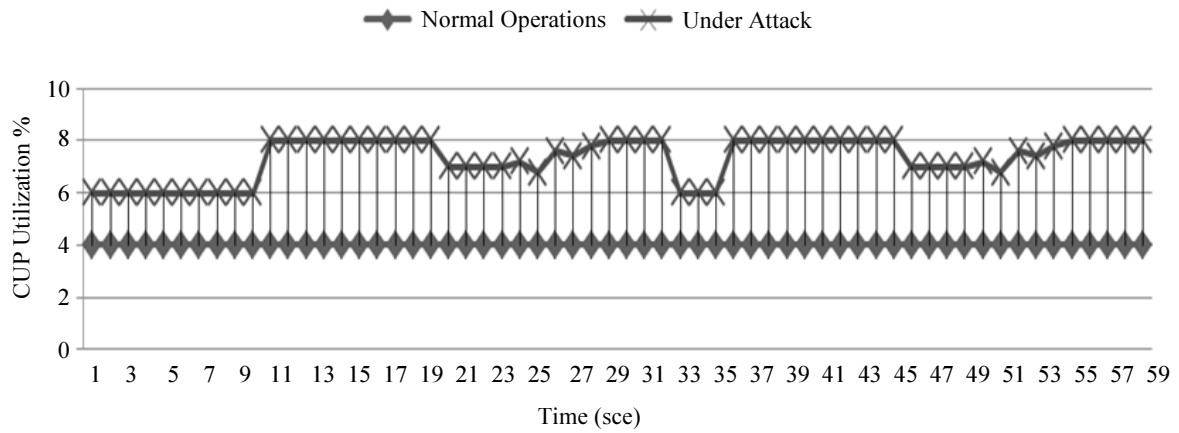


Fig. 38: Processor consumption for windows 10 home before and during SEND DoS attack

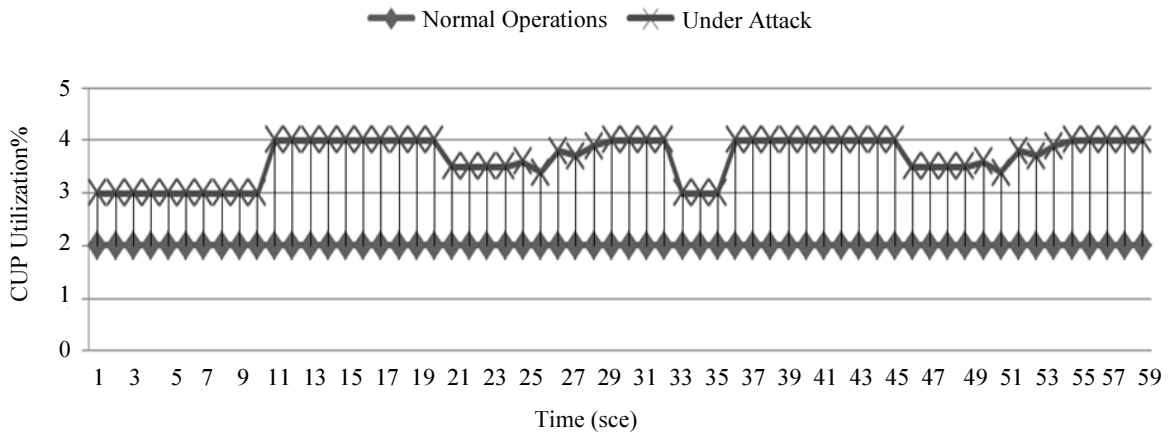


Fig. 39: Processor consumption for ubuntu 16.04 before and during SEND DoS attack

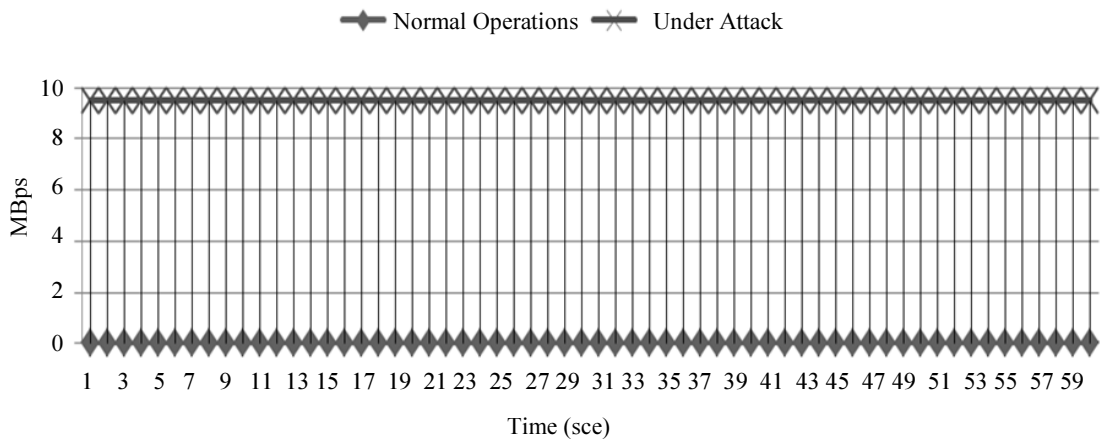


Fig. 40: Bandwidth consumption for windows 10 home before and during SEND DoS attack

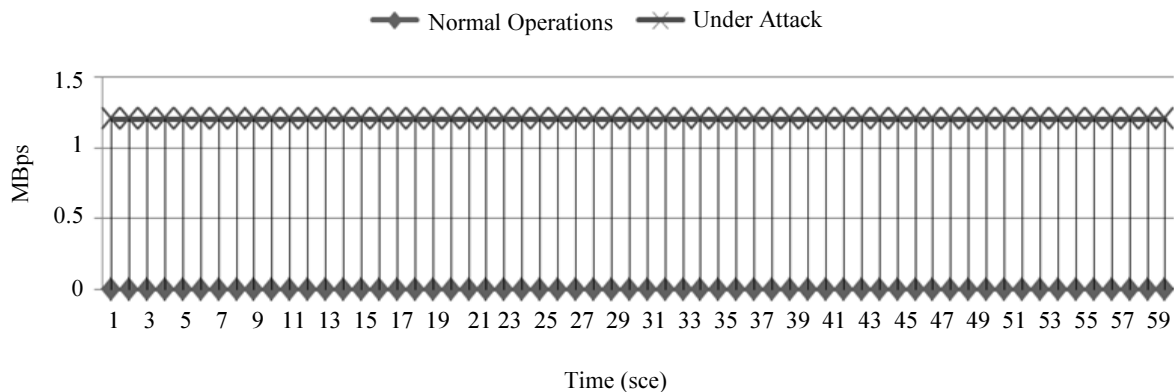


Fig. 41: Bandwidth consumption for ubuntu 16.04 before and during SEND DoS attack

A preliminary experimentation on flooding attack targeting a SEND machine showed that the SEND machine could only process up to 442 NS messages within 1.43 seconds before the machine getting crash (Praptodiyono *et al.*, 2015c). In a word, SEND has many limitations including computation, deployment and security (Ahmed *et al.*, 2017; Gelogo *et al.*, 2011). Proposals to enhance SEND and make it applicable were introduced in (Sarma, 2014; Rafiee *et al.*, 2011; Doja and Sagar, 2012; Kempf *et al.*, 2006; Park *et al.*, 2007; Cheneau and Laurent, 2011; Huang *et al.*, 2009; Oh and Chae, 2007; Vasić *et al.*, 2011; Lu *et al.*, 2017). A small test bed consists of three computers; switch and router were used to implement DoS attacks against SeND. The computers consists of one attacking node (Kali Linux 3.20.2) and two victims nodes (Windows 10 Home and Ubuntu 16.04 respectively). THC-IPv6 attacking tools were used to implement DoS attack using sendpees6. Two performance metrics, processor utilization and network bandwidth consumption, were used to evaluate the impacts of the DoS attack. The attack successfully consumes available resources because it keeps sending incorrect parameters that made CGA verification process to fail (Qadir *et al.*, 2015a; 2015b). Experiment have proved that both Windows and Linux, SeND implementations, are still suffers from DoS attack. Results shown that Linux is more resistible to DoS attack compares to Windows as per Fig. 38 to 41.

Conclusion

NDP is the core protocol of IPv6 suite. When NDP was developed there is an assumption that mutual hosts within a subnet will trust each other. This assumption was proved wrong when it turn into implementation especially in a wireless environment such as airports, cafes and public restaurants. NDP lack security and vulnerable to several DoS attacks that may lead to a total system crash. A test bed setup and corresponding configurations to evaluate the impacts of

NDP attacks on Windows and Linux based operating systems were provided in this study. The impacts of each DoS attack were evaluated using TCP Throughput, RTT and CPU utilization metrics between monitoring and victims computers before and during attacks. Overall, the results have shown that the performance of Linux based operating system was better than Windows based operating system. It was mainly because Linux accept a few number of prefixes while Windows do accept a big number of prefixes during these attacks. We summarized the industry available solutions, describing their technical specifications and components, in addition to highlighting pros and cons of each solution. We also presented available proposals and researches in the era that aim to protect NDP messages and enhance its overall security.

Acknowledgment

Authors would like to acknowledge Universiti Kebangsaan Malaysia (UKM), Limkokwing University and Global College of Engineering and Technology (GCET) for providing supporting tools, research labs and fund to carry out this research.

Author's Contributions

Amjed Sid Ahmed, Rosilah Hassan and Nor Effendy Othman: Authors contribute to this paper by formulating problem statement, plan the methodology and set up experimental test-bed to carry out the results.

Nor Idayu Ahmad and Yassir Kenish: Authors were responsible for systematic literature review, paper writing and formatting.

Ethics

Authors ensure novelty and ownership of the work and results carried out on this paper. Although we did understand that Journal of Computer Science is an open

access based journal, we have no objections to refer, reuse or re-cite any part of the work that have been done on this paper.

References

- Ahmed, A.S., N.H.A. Ismail, R. Hassan and N.E. Othman, 2015a. Balancing performance and security for IPv6 neighbor discovery. *Int. J. Applied Eng. Res.*, 10: 40191-40196.
- Ahmed, A.S., R. Hassan and N.E. Othman, 2015b. Improving security for IPv6 neighbor discovery. *Proceedings of the International Conference on Electrical Engineering and Informatics*, Aug. 10-11, IEEE Xplore Press, Denpasar, Indonesia, pp: 271-274. DOI: 10.1109/ICEEI.2015.7352509
- Ahmed, A.S., R. Hassan and N.E. Othman, 2017. Secure Neighbor Discovery (SeND): Attacks and challenges. *Proceedings of the 6th International Conference on Electrical Engineering and Informatics*, Nov. 25-27, IEEE Xplore Press, Langkawi, Malaysia, pp: 1-6. DOI: 10.1109/ICEEI.2017.8312422
- Ahmed, A.S., R. Hassan and N.E. Othman, 2014. Security threats for IPv6 transition strategies: A review. *Proceedings of the 4th International Conference on Engineering Technology and Technopreneuship*, Aug. 27-29, IEEE Xplore Press, Kuala Lumpur, Malaysia, pp: 83-88. DOI: 10.1109/ICEET.2014.7006224
- Aiello, W., S.M. Bellovin, M. Blaze, J. Ioannidis and A.D. Keromytis *et al.*, 2002. Efficient, DoS-resistant, secure key exchange for internet protocols. *Proceedings of the 9th ACM Conference on Computer and Communications security*, Nov. 18-22, ACM, Washington, DC, USA, pp: 48-58. DOI: 10.1145/586110.586118
- AlSa'deh, A. and C. Meinel, 2012. Secure neighbor discovery: Review, challenges, perspectives and recommendations. *IEEE Security Privacy*, 10: 26-34. DOI: 10.1109/MSP.2012.27
- An, G., K. Kim, J. Jang and Y. Jeon, 2007. Analysis of SEND protocol through implementation and simulation. *Proceedings of the International Conference on Convergence Information Technology*, Nov. 21-23, IEEE Xplore Press, Gyeongju, South Korea, pp: 670-676. DOI: 10.1109/ICCIT.2007.403
- Anbar, M., R. Abdullah, R.M.A. Saad, E. Alomari and S. Alsaleem, 2016. Review of Security Vulnerabilities in the IPv6 Neighbor Discovery Protocol. In: *Information Science and Applications*, Kim, K. and N. Joukov (Eds.), Springer, Singapore, ISBN-10: 978-981-10-0556-5, pp: 603-612.
- Arkko, J., 2002. Effects of ICMPv6 on IKE and IPsec Policies. Internet Draft Draftarkko-icmpv6-ike-ects-01.txt Work In Progress, IETF.
- Arkko, J., P. Nikander, T. Kivinen and M. Rossi, 2002. Manual SA configuration for IPv6 link local messages. Work in Progress.
- Baishya, R.C., N. Hoque and D.K. Bhattacharyya, 2017. DDoS attack detection using unique source IP deviation. *Int. J. Netw. Security*, 6: 929-939. DOI: 10.6633/IJNS.201711.19(6).09
- Blaze, M., 2001. Efficient, DoS-resistant, secure key exchange for internet protocols. *Proceedings of the International Workshop Security Protocols*, (WSP'01), Springer, Berlin Heidelberg, pp: 40-48. DOI: 10.1007/3-540-45807-7_6
- Cheneau, T. and M. Laurent, 2011. Using SEND signature algorithm agility and multiple-key CGA to secure proxy neighbor discovery and anycast addressing. *Proceedings of the Conference on Network and Information Systems Security*, May, 18-21, IEEE Xplore Press, La Rochelle, France, pp: 1-7. DOI: 10.1109/SAR-SSI.2011.5931376
- Chiu, S. and E. Gamess, 2010. A free and didactic implementation of the SEND protocol for IPv6. In: *Machine Learning and Systems Engineering*, Ao, S.I., B. Rieger, M. Amouzegar (Eds.), Springer Netherlands, ISBN-13: 978-90-481-9418-6, pp: 451-463.
- Doja, M.N. and R. Saggat, 2012. Securing IPv6's Neighbour discovery, using locally authentication process. *Int. J. Comput. Eng. Res.*, 2: 1234-1242.
- Gelogo, Y.E., R.D. Caytiles and B. Park, 2011. Threats and security analysis for enhanced secure neighbor discovery protocol (send) of ipv6 ndp security. *Int. J. Control Automat.*, 4: 179-184.
- Hakim, N., M.U. Siddiqi and H.M. Rafiq, 2015. Simulation study of a many-to-one mapping for IPv6 address owner identification in an enterprise local area network. *Int. J. Netw. Security*, 17: 106-113.
- Harkins, D. and D. Carrel, 1998. The Internet Key Exchange (IKE). No. RFC 2409.
- Hassan, R., A.S. Ahmed, N.E. Othman and S. Sami, 2014. Enhanced encapsulated security payload a new mechanism to secure internet protocol version 6 over internet protocol version 4. *J. Comput. Sci.*, 10: 1344-1354. DOI: 10.3844/jcssp.2014.1344.1354
- Huang, M., J. Liu and Y. Zhou, 2009. An improved SEND protocol against DoS attacks in Mobile IPv6 environment. *Proceedings of the IEEE International Conference on Network Infrastructure and Digital Content*, Nov. 6-8, IEEE Xplore Press, Beijing, China, pp: 232-235. DOI: 10.1109/ICNIDC.2009.5360962
- Kavitha, R. and G. Padmavathi, 2017. Advanced random time queue blocking for effective protection of application servers against low-rate DoS attacks. *Int. J. Netw. Security*, 19: 1024-1035.

- Kempf, J., J. Wood, Z. Ramzan and C. Gentry, 2006. Ip address authorization for secure address proxying using multi-key cgas and ring signatures. Proceedings of the International Workshop on Security, (IWS' 06), Springer, Berlin Heidelberg, pp: 196-211.
- Kim, T., I. Kim, Z. Zhen, J.H. Kim and G. Gyeong *et al.*, 2008. A cooperative authentication of ipsec and send mechanisms in ipv6 environments. Proceedings of the International Conference on Advanced Language Processing and Web Information, Jul. 23-25, IEEE Xplore Press, Dalian Liaoning, China, pp: 418-423. DOI: 10.1109/ALPIT.2008.43
- Kuldeep, T. and S.S. Tyagi, 2014. Enhancing Network Security by implementing preventive mechanism using GNS3. Proceedings of the International Conference on Reliability Optimization and Information Technology, Feb. 6-8, IEEE Xplore Press, Faridabad, India, pp: 300-305. DOI: 10.1109/ICROIT.2014.6798342
- Liu, H.C. and Q.G. Dai, 2013. Design of security neighbor discovery protocol. Proceedings of the International Conference on Communication Systems and Network Technologies, Apr. 6-8, IEEE Xplore Press, Gwalior, India, pp: 538-541. DOI: 10.1109/CSNT.2013.195
- Lu, Y., M. Wang and P. Huang, 2017. An SDN-based authentication mechanism for securing neighbor discovery protocol in IPv6. Security Commun. Netw.
- Mohamed, S.A., R. Hassan and N.E. Othman, 2017. IPv6 neighbor discovery protocol specifications, threats and countermeasures: A Survey. IEEE Access, 5: 18187-18210.
- Najjar, F., M. Kadhum and H. El-Taj, 2015. Neighbor discovery protocol anomaly detection using finite state machine and strict anomaly detection. Proceedings of the 4th International Conference Internet Application, Protocols Series, (APS' 15).
- Najjar, F., M.M. Kadhum and H. El-Taj, 2016. Detecting Neighbor Discovery Protocol-Based Flooding Attack using Machine Learning Techniques. In: Advances in Machine Learning and Signal Processing, Soh, P., W. Woo, H. Sulaiman, M. Othman and M. Saat (Eds.), Cham, Switzerland, Springer, ISBN-13: 978-3-319-32212-4, pp: 129-139.
- Nikander, P., 2001. Denial-of-service, address ownership and early authentication in the IPv6 world. Proceedings of the International Workshop on Security Protocols, (WSP' 01), Springer, Berlin Heidelberg, pp: 12-21.
- Oh, H. and K. Chae, 2007. An efficient security management in IPv6 network via MCGA. Proceedings of the the 9th International Conference on Advanced Communication Technology, Feb. 12-14, IEEE Xplore Press, Okamoto, Kobe, Japan, pp: 1179-1181. DOI: 10.1109/ICACT.2007.358569
- Park, J.H., K.H. Choi, J.S. Kim, C.I. Cho and E.G. Im, 2007. A survey of the Secure Neighbor Discovery (SEND) and Multi-key Cryptographically Generated Addresses (MCGAS). Proceedings of the 9th International Conference on Advanced Communication Technology, Feb. 12-14, IEEE Xplore Press, Okamoto, Kobe, Japan, pp: 2124-2127. DOI: 10.1109/ICACT.2007.358791
- Perumal, K. and M.J.P.J. Priya, 2016. Trust based security enhancement mechanism for neighbor discovery protocol in IPV6. Int. J. Applied Eng. Res., 7: 4787-4796.
- Praptodiyono, S., I.H. Hasbullah, M. Anbar, R.K. Murugesan and A. Osman, 2015a. Improvement of address resolution security in IPv6 local network using trust-ND. TELKOMNIKA Indonesian J. Electr. Eng., 1: 195-202.
- Praptodiyono, S., I.H. Hasbullah, M.M. Kadhum, R.K. Murugesan and A. Osman *et al.*, 2015b. Improving security of duplicate address detection on IPv6 local network in public area. Proceedings of the 9th Asia Modelling Symposium, Sept. 7-9, IEEE Xplore Press, Kuala Lumpur, Malaysia, pp: 123-128. DOI: 10.1109/AMS.2015.28
- Praptodiyono, S., R.K. Murugesan, I.H. Hasbullah, C.Y. Wey and A. Osman *et al.*, 2015c. Security mechanism for IPv6 stateless address autoconfiguration. Proceeding of the International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System and Information Technology, Oct. 29-30, IEEE Xplore Press, Bandung, Indonesia, pp: 31-36. DOI: 10.1109/ICACOMIT.2015.7440150
- Qadir, S., M.U. Siddiqi and W.F. Al-Khateeb, 2015a. Analysing and improving performance and security of cryptographically generated address algorithm for mobile IPv6 networks. Int. J. Netw. Security, 17: 535-547.
- Qadir, S., M.U. Siddiqi and W.F.M. Al-Khateeb, 2015b. An investigation of the Merkle signature scheme for cryptographically generated address signatures in mobile IPv6. Int. J. Netw. Security, 17: 311-321.
- Rafiee, H., A. Alsa'deh and C. Meinel, 2011. Winsend: Windows secure neighbor discovery. Proceedings of the 4th International Conference on Security of Information and Networks, Nov. 14-19, ACM, Sydney, Australia, pp: 243-246. DOI: 10.1145/2070425.2070469
- Rehman, S.U. and S. Manickam, 2015a. Integrated framework to detect and mitigate Denial of Service (DoS) attacks on duplicate address detection process in IPv6 link local communication. Int. J. Security Appl., 11: 77-86.

- Rehman, S.U. and S. Manickam, 2015b. Rule-based mechanism to detect Denial of Service (DoS) attacks on duplicate address detection process in IPv6 link local communication. Proceeding of the 4th International Conference on Reliability, Infocom Technologies and Optimization, Sept. 2-4, IEEE Xplore Press, Noida, India, pp: 1-6.
DOI: 10.1109/ICRITO.2015.7359243
- Rehman, S.U. and S. Manickam, 2015c. Significance of duplicate address detection mechanism in Ipv6 and its security issues: A survey. *Ind. J. Sci. Technol.*, 8: 1-8. DOI: 10.17485/ijst/2015/v8i30/85940
- Rehman, S.U. and S. Manickam, 2016. Novel mechanism to prevent Denial Of Service (DoS) attacks in IPv6 duplicate address detection process. *Int. J. Security Applic.*, 4: 143-154.
- Saad, R.M.A., M. Anbar, S. Manickam and E. Alomari, 2015. An intelligent ICMPv6 DDoS flooding-attack detection framework (v6IIDS) using back-propagation neural network. *IETE Tech. Rev.*, 3: 244-255.
- Sarma, S., 2014. Securing IPv6's NEIGHBOUR and router discovery, using locally authentication process. *IOSR J. Comput. Eng.*, 1: 22-31.
DOI: 10.9790/0661-16352231
- Shah, J.L., 2016. A novel approach for securing IPv6 link local communication. *Inform. Security J.*, 25: 136-150.
- Shah, J.L. and J. Parvez, 2015. Optimizing security and address configuration in IPv6 SLAAC. *Proc. Comput. Sci.*, 54: 177-185.
DOI: 10.1016/j.procs.2015.06.020
- Shrivastava, G., K. Sharma and S. Rai, 2010. The detection and defense of DoS and DDos attack: A technical overview. *Proceedings of the ICC, (ICC' 10)*, pp: 28.
- Song, G. and Z. Ji, 2016. Novel duplicate address detection with hash function. *PLoS ONE*, 3: e0151612.
- Vasić, V., A. Kukec and M. Mikuc, 2011. Deploying new hash algorithms in secure neighbor discovery. *Proceedings of the 19th International Conference on Software, Telecommunications and Computer Networks*, Sept. 5-17, IEEE Xplore Press, Split, Croatia, pp: 1-5.