DoS and DDoS Attacks: Impact, Analysis and Countermeasures

Nikhil Tripathi
Indian Institute of Technology Indore

Babu Mehtre
IDRBT - Institute for Development & Research in Banking Technology

DoS and DDoS Attacks: Impact, Analysis and Countermeasures

Nikhil Tripathi
School of Computer and Information Sciences
Hyderabad Central University
Institute for Development and Research in Banking Technology
Hyderabad, India
nikhiltripathi684@gmail.com

B.M. Mehtre
Center for Information Assurance and Management
Institute for Development and Research in Banking Technology
(Established by Reserve Bank of India)
Hyderabad, India
bmmehtre@idrbt.ac.in

Abstract— **Confidentiality, Integrity and Availability are the three major components of cyber security. Denial of Service (DoS) and its variant, Distributed Denial of Service (DDoS), are possible threats which exhaust the resources to make it unavailable for the legitimate users, thereby, violating one of the security components- *Availability*. DoS attacks to networks are numerous and potentially devastating. So far, many types of DoS attacks are identified and most of them are quite effective to stop the communication in the networks. IPv4 as well as IPv6 are quite vulnerable to these attacks. A number of countermeasures are developed to mitigate these attacks. This paper presents classification of DoS/DDoS attacks under IPv4 and IPv6. The impact of these attacks, analysis and their countermeasures are also discussed in this paper. The analysis of these attacks is performed using different utilities and traffic analyzer.**

Index Terms— **IPv4, IPv6, DoS/DDoS, Attack Experiments, Security Issues, Countermeasures, Cyber Defense.**

I. INTRODUCTION

Denial of Service (DoS) attacks to networks are numerous and potentially devastating. So far, many types of DoS attacks are identified and most of them are quite effective to stop the communication in the networks. These attacks involve either the use of single computer or multiple computers, called zombies. Former technique is known as simple DoS attack and latter as Distributed Denial of Service(DDoS) attacks. Not only IPv4, but also the new road map of internet, IPv6, is quite vulnerable to DoS attacks. A number of countermeasures are developed to mitigate these attacks.

This paper is organized as follows: Section II includes a brief understanding of DoS Attacks, to form the basis for subsequent sections. Section III describes the classification of attacks on the basis of IPv4 and IPv6. Section IV, V and VI describes the Layer-4 DDoS attacks, Layer-7 DoS attacks and IPv6 Neighbor Discovery Protocol [1] based attacks, respectively. Section VII includes an overview of countermeasures being used against these attacks. Finally, Section VIII concludes the paper.

II. BACKGROUND: DoS ATTACKS

This section covers a brief understanding of DoS attacks to form the basis for the subsequent sections.

A DoS is an attack which is launched to make networks' and systems' resources unavailable for the legitimate users so that no one else can access it. Hackers can create a situation in which the organizations come to a grinding halt. The main targets of these attacks are web servers, default gateways, personal computers, etc.

Most of the hackers keep three things in mind. One is to explore a way through which they can get the secret information. This is to compromise the *confidentiality*. Second is get access to the confidential information to change or modify it. This involves the compromising of *integrity*. Third is to compromise the *availability*. The first two options cannot be enjoyed by novice attackers because it is not easy to gain an unauthorized remote access to a system. Thus, they try to target the availability for which they do not need any administrative privilege on the target system.

Most DoS attacks depend upon the weaknesses in TCP/IP stack protocols. Some of the classical examples of DoS attacks are TCP Syn Flood, UDP Flood, ICMP flood, Smurf [2] and Incomplete HTTP Requests, etc.

Attackers either make use of single computer or multiple computers to launch these attacks. The usage of multiple computers to perform the attack is known as DDoS attack. The different systems are first compromised by using Trojans, worms, etc. and then used by the attackers. These compromised machines are named as *zombies* while the controller machine is considered as *master*. This master-zombie relationship works somewhat similar to client-server architecture. It can be very difficult to detect the DDoS attacks because the zombies may be situated across the globe. As a result, they cannot be differentiated from the legitimate traffic.

III. CLASSIFICATION

Figure 1 shows the classification of DoS/DDoS attacks. First, the attack is divided on the basis of IPv4 and IPv6. Then both are further divided into Layer-4 and Layer-7 attacks. Moreover, IPv6 attacks are also divided into Neighbor Discovery protocol based attacks [3].

The subsequent sections IV, V and VI explain these attacks along with their generated traffic captured on Wireshark [4] to show the attacks experimentally. We have performed all the experiments on the Debian-based KALI LINUX [5] operating system which is a well-known penetration testing tool.

IV. LAYER-4 ATTACKS

In this section, we have explained different Layer-4 based DoS/DDoS attacks which are possible under IPv4 and IPv6. We have shown the experimental traffic under IPv4, considering the steps involved in these attacks under IPv6 are the same. Layer-4 attacks include the following:

A. TCP-flood Attack

Xinyu Yang et al. [3] proved that this attack is effective under both *IPv4* and *IPv6*. This attack exploits the Three Way Handshake mechanism of TCP protocol [6].

When a client machine wants to establish a connection with a server, both machines exchange a set of messages sequentially. This is known as Three Way Handshaking mechanism. First, the client sends a SYN (synchronization) message to the server. The server then sends back a SYN-ACK (acknowledgment) message to acknowledge the SYN message sent by the client. The client then responds with an ACK message to finish the establishment of the connection. The connection is then opened and as a result, data exchange between the server and the client start taking place.

To exploit this mechanism, an attacker sends a series of SYN messages to the victim with spoofed source IP addresses. As a result, the victim responds back with a SYN-ACK message and then wait for a fixed amount of time for an ACK to come back so as to finish the connection establishment. Since the attacker has sent the SYN messages with spoofed source addresses, there will not be any ACK return due to absence of such addresses. As a result, these several *partially-opened* connections fill the connection queue and memory buffers. Thus, the legitimate users will not be able to get services anymore.

The experimental traffic captured for this attack using Wireshark is shown in Fig.2. The part of the image above the bold red line represents the SYN packet sent by the attacker. The SYN is shown inside the red box. The part of the image below the red line represents the (SYN, ACK) packet sent by the victim to the attacker. The SYN-ACK is shown inside the red box. In response to this packet, the attacker will not send any ACK so that it results in partially-opened connections.

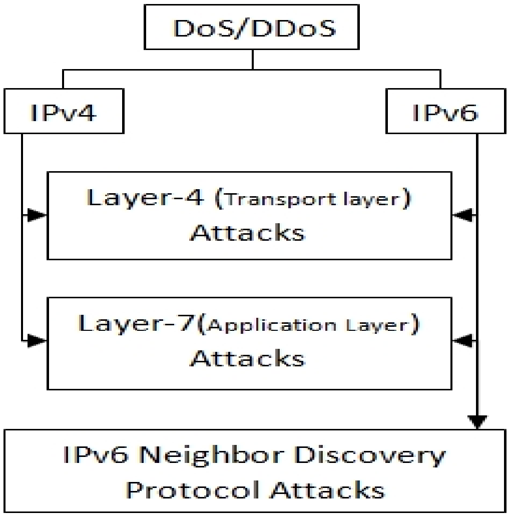

DoS/DDoS
IPv4
IPv6
Layer-4 (Transport layer) Attacks
Layer-7(Application Layer) Attacks
IPv6 Neighbor Discovery Protocol Attacks

Fig. 1. Classification of DoS/DDoS Attacks

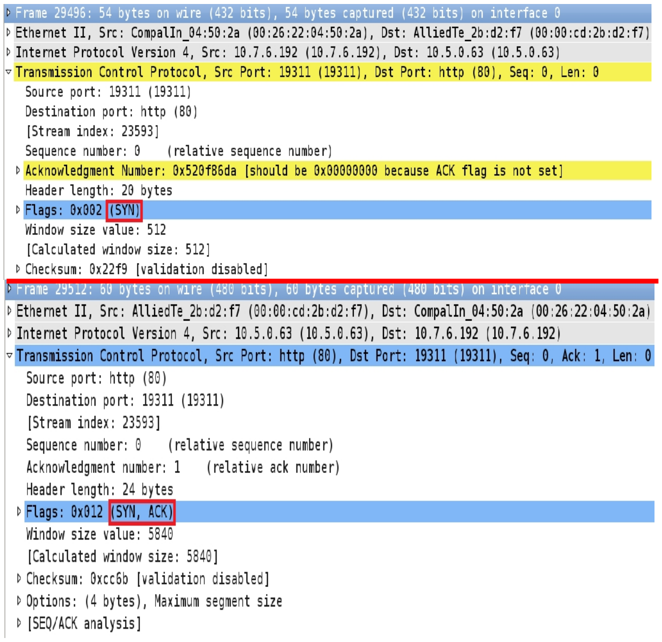
B. UDP-flood Attack

There is no such Three Way Handshaking (TWH) mechanism required to establish a UDP connection. As a result, it is secure against exploiting TWH vulnerabilities. Two services CHARGEN (character generator) and ECHO [3] of UDP protocol provides a technique through which UDP-flood can be launched. CHARGEN service is supported at 19th port whereas ECHO is supported at 7th port of UDP. CHARGEN service returns a random string when someone tries to connect to it whereas ECHO service returns what we have sent to the entering node. If an attacker connects the victim's 19th port to the 7th port of some other node, there will be a huge amount of traffic flow between the victim and the other node. This can be achieved by sending a packet to the victim's 19th port with a spoofed source address which is targeted towards the other node's 7th port.

Since this flood is also based on IP spoofing technique, we need not explain this attack in detail.

C. ICMP(v6)-flood Attack

ICMP(v6)-based utility *ping(Packet Internet Groper)* uses echo response mechanism. In this attack, attacker sends large amount of packets to the victim with different spoofed invalid source IP addresses. This results in the victim's resource wastage and makes the network bandwidth drain, causing legitimate packets unable to get the services [6].



```
▷ Frame 29496: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
▷ Ethernet II, Src: CompalIn_04:50:2a (00:26:22:04:50:2a), Dst: AlliedTe_2b:d2:f7 (00:00:cd:2b:d2:f7)
▷ Internet Protocol Version 4, Src: 10.7.6.192 (10.7.6.192), Dst: 10.5.0.63 (10.5.0.63)
▽ Transmission Control Protocol, Src Port: 19311 (19311), Dst Port: http (80), Seq: 0, Len: 0
    Source port: 19311 (19311)
    Destination port: http (80)
    [Stream index: 23593]
    Sequence number: 0    (relative sequence number)
  ▷ Acknowledgment Number: 0x520f86da [should be 0x00000000 because ACK flag is not set]
    Header length: 20 bytes
  ▷ Flags: 0x002 (SYN)
    Window size value: 512
    [Calculated window size: 512]
  ▷ Checksum: 0x22f9 [validation disabled]
▷ Frame 29512: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▷ Ethernet II, Src: AlliedTe_2b:d2:f7 (00:00:cd:2b:d2:f7), Dst: CompalIn_04:50:2a (00:26:22:04:50:2a)
▷ Internet Protocol Version 4, Src: 10.5.0.63 (10.5.0.63), Dst: 10.7.6.192 (10.7.6.192)
▽ Transmission Control Protocol, Src Port: http (80), Dst Port: 19311 (19311), Seq: 0, Ack: 1, Len: 0
    Source port: http (80)
    Destination port: 19311 (19311)
    [Stream index: 23593]
    Sequence number: 0    (relative sequence number)
    Acknowledgment number: 1    (relative ack number)
    Header length: 24 bytes
  ▷ Flags: 0x012 (SYN, ACK)
    Window size value: 5840
    [Calculated window size: 5840]
  ▷ Checksum: 0xcc6b [validation disabled]
  ▷ Options: (4 bytes), Maximum segment size
  ▷ [SEQ/ACK analysis]
```

Fig. 2. Traffic Analysis for TCP-flood

It is somewhat similar to the Smurf attack so we need not show traffic analysis for this attack.

D. Smurf Attack

This attack is somewhat similar to the ICMP (v6)-flood attack in the sense that it also uses the echo response mechanism of ICMP (v6). In this attack, the attacker broadcasts packets with the spoofed source IP address targeted to the victim. Since the packets are sent at broadcast address, it is received by all the nodes within the network. As a result, each and every node responds back to the victim machine since the source IP address is spoofed as that of the victim's address. Since it causes a large amount of echo response packets, the victim's resources can easily be exhausted.

According to the RFC2463 [7], if a node receives a packet with an IPv6 multicast destination address, a link-layer multicast address, or a link-layer broadcast address, it should not generate a response. As a result, Smurf Attacks are not effective under IPv6.

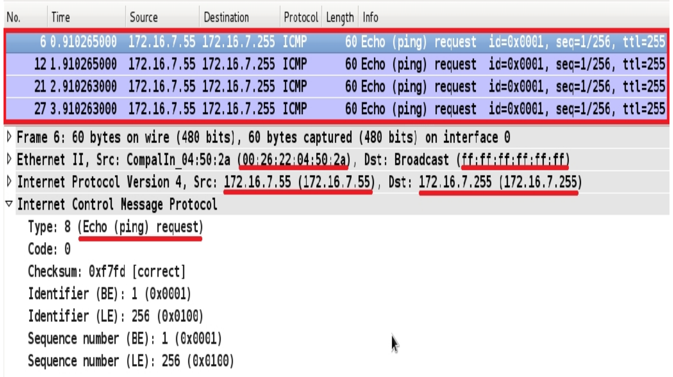
The experimental traffic captured for this attack under IPv4 using Wireshark is shown in Fig.3. The victim's MAC address and IP address is set as the source MAC address and source IP address for sending the requests. The destination MAC and IP addresses are set to the broadcast address so that the request is received is received by every host within the subnet. As a result, every host will respond back to victim which causes the unusual consumption of the victim's resources.

V. LAYER-7 ATTACKS

In this section, we have explained different Layer-7 DoS Attacks which are possible under IPv4 and IPv6. We have shown the experimental traffic under IPv4, considering the steps involved in these attacks under IPv6 are the same unless specified. They are:

A. Incomplete HTTP requests using GET method

This attack is based on how client sends the data to the web server while communication goes on between them. In this attack, client sends HTTP requests to the web server but in a different way. Client sends just a part of the HTTP header [8] and never sends the complete header. Client continues to send subsequent headers at regular intervals to keep socket alive. Client sends multiple incomplete requests to exhaust the server's resources. As a result, these requests consume all the available resources on the server, thereby denying the legitimate users' requests.



```
No.  Time        Source      Destination  Protocol Length Info
   6 0.910265000 172.16.7.55 172.16.7.255 ICMP        60 Echo (ping) request  id=0x0001, seq=1/256, ttl=255
  12 1.910265000 172.16.7.55 172.16.7.255 ICMP        60 Echo (ping) request  id=0x0001, seq=1/256, ttl=255
  21 2.910263000 172.16.7.55 172.16.7.255 ICMP        60 Echo (ping) request  id=0x0001, seq=1/256, ttl=255
  27 3.910263000 172.16.7.55 172.16.7.255 ICMP        60 Echo (ping) request  id=0x0001, seq=1/256, ttl=255
▷ Frame 6: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▷ Ethernet II, Src: CompalIn_04:50:2a (00:26:22:04:50:2a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Internet Protocol Version 4, Src: 172.16.7.55 (172.16.7.55), Dst: 172.16.7.255 (172.16.7.255)
▽ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xf7fd [correct]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 1 (0x0001)
    Sequence number (LE): 256 (0x0100)
```

Fig. 3. Generated traffic in case of smurf attack

This attack is quite dangerous because it can be launched with minimal bandwidth. Moreover, it does not require multiple number of computers to launch the attack. Also, when the attack is stopped, the server restores back within few seconds.

We used a tool *Slowloris [9]* to perform this attack. We performed this attack on the Apache 2.2 [10] web server. It is vulnerable to this attack and the results are shown in the Fig.4.

The '*R*' code in Fig. 4 shows that the server is reading requests while '*W*' code shows that the server is sending reply.

We highlighted the incoming server requests with red color rectangles. It is showing that no further connections will be entertained.

B. Incomplete http requests using post method

This attack is similar to the Incomplete HTTP requests using GET method. The only difference is in this case, client sends incomplete HTTP requests with the help of POST method instead of GET method.

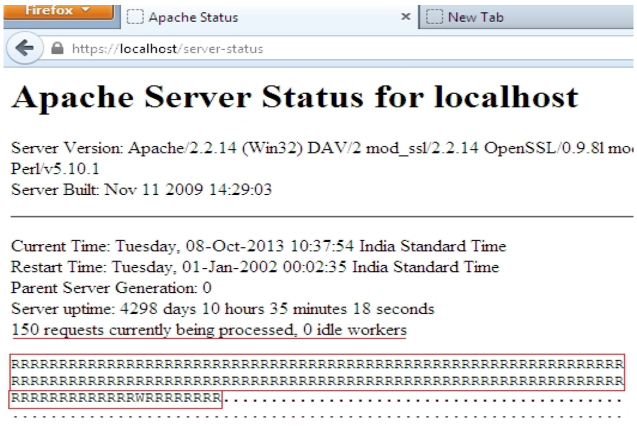
Since this attack is quite familiar to the GET method based attack, we do not present the experimented results for this attack.

C. HTTP requests using HEAD method

HEAD method differs from the GET method only in the sense that using HEAD method, the server must not return a message body in the response [8]. This saves resources on the attacker's side. With the help of HEAD method, the attacker targets a page that is expensive for the server to create, e.g. a search.

VI. IPv6 NEIGHBOR DISCOVERY PROTOCOL ATTACKS

In this section, we have explained different Neighbor Discovery Protocol based DoS Attacks which are possible



Apache Server Status for localhost

Server Version: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l mo
Perl/v5.10.1
Server Built: Nov 11 2009 14:29:03

Current Time: Tuesday, 08-Oct-2013 10:37:54 India Standard Time
Restart Time: Tuesday, 01-Jan-2002 00:02:35 India Standard Time
Parent Server Generation: 0
Server uptime: 4298 days 10 hours 35 minutes 18 seconds
150 requests currently being processed, 0 idle workers

```
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR
RRRRRRRRRRRRRWRRRRRRRR..........................................
................................................................
```

Fig. 4. Server Status after Incomplete HTTP requests using GET method

under IPv6. First, we have discussed two Neighbor Discovery based attacks. They are:

A. Duplicate Address Detection Attack

IPv6 provides ability to the nodes to configure themselves with global IP addresses automatically even without using DHCP server. This is known as Stateless Address Configuration [11]. In this process, one node tries to allot itself an IPv6 address but before allotting it permanently, it ensures that the address it wants is not already being used other node. The node ensures it by multicasting Neighbor Solicitation Messages with an unspecified source address *(::)* targeting on the address which is to be checked. If the node receives a Neighbor Advertisement Message for that address then it means that the address is already being used. Otherwise the checking node considers that the address is available and it can use that address for further communications. This process is known as *Duplicate Address Detection (DAD)*.

This DAD mechanism can be exploited very easily. An attacker can send spoofed Neighbor Advertisement Messages in response to the DAD Neighbor Solicitation Messages with the source IP address as the address which is being checked. As a result, the legitimate node will not be able to acquire the IPv6 address.

We used the '*dos-new-ip6'* utility of Kali Linux to perform this attack. Fig. 5 shows that the attack sends spoofed Neighbor Advertisement packets for each DAD Neighbor Solicitation messages. As a result, the victim will not be able to configure itself with an address automatically.

B. Neighbor Solicitation/Advertisement Spoofing

This attack is similar to the ARP poisoning in the IPv4 networks. In this attack, attacker sends spoofed Neighbor Advertisement messages in response to the neighbor solicitation messages. As a result, the legitimate traffic is redirected to an illegitimate node. This is generally performed to get a Man-In-The-Middle(MITM) position.

Since this attack is also based on spoofed Neighbor Advertisement messages, it is similar to the Neighbor Unreachability Detection Failure attack. So we need not to show the experimental results for this attack.

The attacks based on Router Discovery mechanism are:

A. Killing the Default Router

Nodes maintain a Default router list [1] of IP addresses corresponding to on-link routers that send Router Advertisement messages and are eligible to be default routers. If the list is empty then the sender assumes that there is no default router and all the nodes are on-link. To launch this attack, the attacker sends spoofed Router Advertisement Messages with a zero Router lifetime[3]. As a result, the victim node tries to send all the packets directly to the nodes rather than by router. Since the destination is not an on-link node, the packet will never reach the real destination.

The experimental traffic captured for this attack with the help of Wireshark is shown in Fig.6.

B. Bogus Address Configuration Prefix attack

In absence of DHCPv6, on-link nodes configure IPv6 addresses [12] with the help of on-link prefixes advertised by router in the router advertisement messages. Generally, these messages are advertised periodically. To launch this attack, attacker can easily send spoofed router advertisement messages having invalid prefixes. With the help of these prefixes, the nodes configure addresses automatically. As a result, they configure themselves with an invalid address which causes the loss of communication between the nodes.

The most serious problem this attack causes is it results in complete consumption of CPU resource on the nodes using Windows Platform operating system. We tested it even on the latest versions of Microsoft Windows based operating systems and found out that this attack is effective. However, this attack is ineffective to consume the CPU resources on the nodes using Linux operating systems.

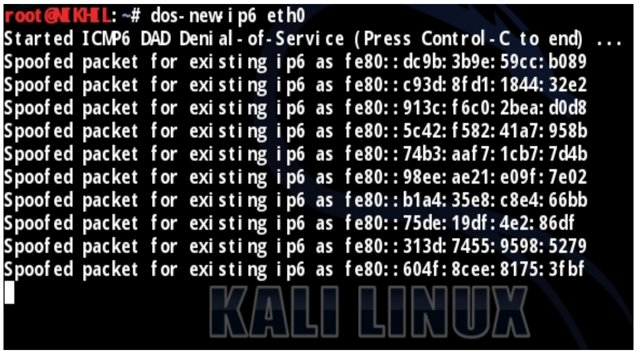
The experimental traffic captured for this attack with the help of Wireshark is shown in Fig.7. The bigger red colored rectangle shows the continuous flood of Router advertisement messages. The smaller rectangle highlights the prefix being advertised within a particular message. As a result, a flood of different prefixes swims within the whole network. It causes the hosts to configure themselves with different IPv6 addresses which makes Windows platform vulnerable. The result of multiple address configurations on Windows host is shown in Fig. 8.

There are some other attacks based on router advertisement messages like *Malicious Last Hop Router* attack, *parameter spoofing* attack, etc. These attacks can be launched in the same way which we discussed in various attacks previously so we need not explain these attacks.

VII. COUNTERMEASURES

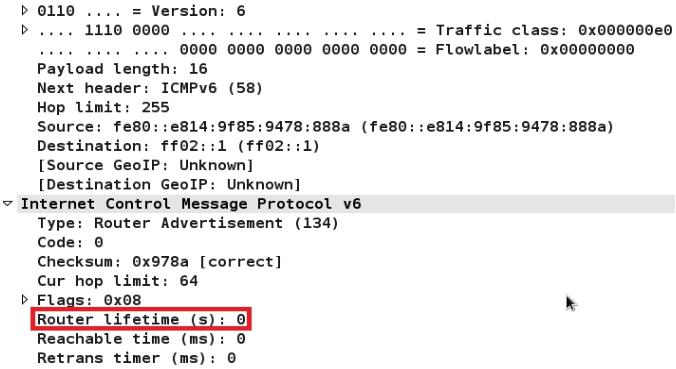
DoS/DDoS attacks are not stand-alone attacks. These attacks require other attacks first which have to be performed. Some examples of these attacks are IP Spoofing, ARP spoofing/poisoning, MITM attacks, MAC spoofing, etc. If these attacks are prevented, it will be quite difficult for the attackers to launch the DoS/DDoS attacks.

The various countermeasures/defenses against different types of DoS attacks are described in this section.



```
root@NIKHIL:~# dos-new-ip6 eth0
Started ICMP6 DAD Denial-of-Service (Press Control-C to end) ...
Spoofed packet for existing ip6 as fe80::dc9b:3b9e:59cc:b089
Spoofed packet for existing ip6 as fe80::c93d:8fd1:1844:32e2
Spoofed packet for existing ip6 as fe80::913c:f6c0:2bea:d0d8
Spoofed packet for existing ip6 as fe80::5c42:f582:41a7:958b
Spoofed packet for existing ip6 as fe80::74b3:aaf7:1cb7:7d4b
Spoofed packet for existing ip6 as fe80::98ee:ae21:e09f:7e02
Spoofed packet for existing ip6 as fe80::b1a4:35e8:c8e4:66bb
Spoofed packet for existing ip6 as fe80::75de:19df:4e2:86df
Spoofed packet for existing ip6 as fe80::313d:7455:9598:5279
Spoofed packet for existing ip6 as fe80::604f:8cee:8175:3fbf
```

Fig. 5. Spoofed Neighbor Advertisements for DAD messages



```
▷ 0110 .... = Version: 6
▷ .... 1110 0000 .... .... .... .... .... = Traffic class: 0x000000e0
  .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
  Payload length: 16
  Next header: ICMPv6 (58)
  Hop limit: 255
  Source: fe80::e814:9f85:9478:888a (fe80::e814:9f85:9478:888a)
  Destination: ff02::1 (ff02::1)
  [Source GeoIP: Unknown]
  [Destination GeoIP: Unknown]
▽ Internet Control Message Protocol v6
  Type: Router Advertisement (134)
  Code: 0
  Checksum: 0x978a [correct]
  Cur hop limit: 64
▷ Flags: 0x08
  Router lifetime (s): 0
  Reachable time (ms): 0
  Retrans timer (ms): 0
```

Fig. 6. Spoofed Router Advertisements claiming Default Router as killed

A. Defenses against Layer-4 DDoS Attacks

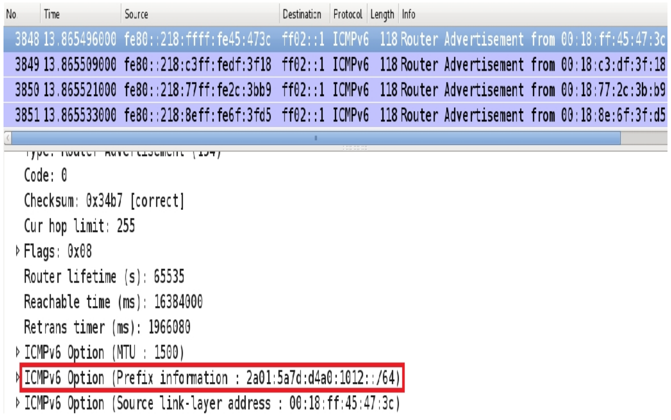
Since all the Transport Layer DDoS attacks are based on IP spoofing, simply defending against IP spoofing will be suffice to stop the DDoS attacks. Generally, network administrators use IPSec [13] to prevent the IP Spoofing. However, if the attacker is using large number of nodes as *zombies,* this defense will not work. Because the zombies may be distributed in different time zones due to which their IP addresses will vary widely. As a result, the traffic generated by them cannot be differentiated from the legitimate traffic.

Some other types of defenses involve the usage of IDS based software tools like Snort [14] which is an open source network intrusion detection system (NIDS) for networks.

Ahmad Sanmorino et al. [15] presented an idea of detecting and mitigating DDoS attacks using pattern of the flow entries and handling mechanism using layered firewall. They proved that the use of patterns of flow entries provides more accuracy for the detection of the source IP address of the attacker. With the help of this detection technique, IP spoofing method used by hacker becomes useless.

B. Defenses against Layer-7 DDoS Attacks

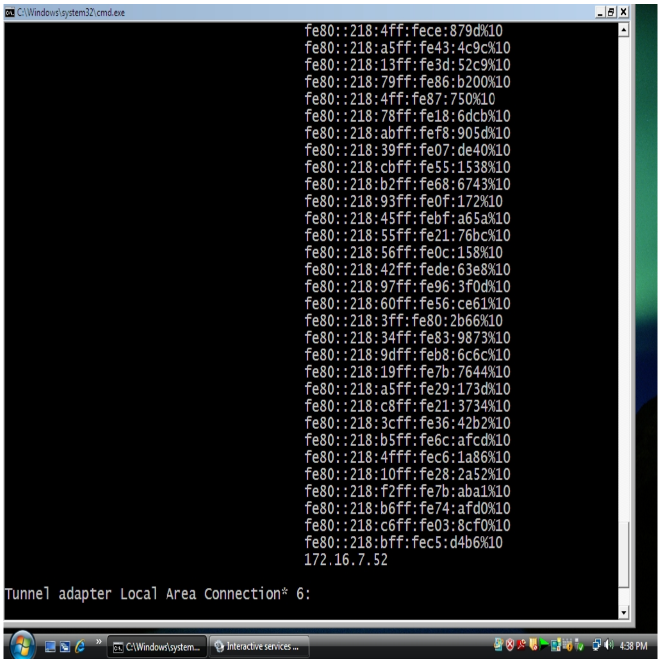
These attacks are quite dangerous because they are not limited to the usage of IP spoofing. These attacks can be launched using minimal bandwidth also.



```
No.  Time         Source                     Destination  Protocol  Length  Info
3848 13.865496000 fe80::218:ffff:fe45:473c   ff02::1      ICMPv6    118     Router Advertisement from 00:18:ff:45:47:3c
3849 13.865509000 fe80::218:c3ff:fedf:3f18   ff02::1      ICMPv6    118     Router Advertisement from 00:18:c3:df:3f:18
3850 13.865521000 fe80::218:77ff:fe2c:3bb9   ff02::1      ICMPv6    118     Router Advertisement from 00:18:77:2c:3b:b9
3851 13.865533000 fe80::218:8eff:fe6f:3fd5   ff02::1      ICMPv6    118     Router Advertisement from 00:18:8e:6f:3f:d5

  Code: 0
  Checksum: 0x34b7 [correct]
  Cur hop limit: 255
▷ Flags: 0x08
  Router lifetime (s): 65535
  Reachable time (ms): 16384000
  Retrans timer (ms): 1966080
▷ ICMPv6 Option (MTU : 1500)
▷ ICMPv6 Option (Prefix information : 2a01:5a7d:d4a0:1012::/64)
▷ ICMPv6 Option (Source link-layer address : 00:18:ff:45:47:3c)
```

Fig. 7. Spoofed Router Advertisements containing bogus prefix information



C:\Windows\system32\cmd.exe

```
                    fe80::218:4ff:fece:879d%10
                    fe80::218:a5ff:fe43:4c9c%10
                    fe80::218:13ff:fe3d:52c9%10
                    fe80::218:79ff:fe86:b200%10
                    fe80::218:4ff:fe87:750%10
                    fe80::218:78ff:fe18:6dcb%10
                    fe80::218:abff:fef8:905d%10
                    fe80::218:39ff:fe07:de40%10
                    fe80::218:cbff:fe55:1538%10
                    fe80::218:b2ff:fe68:6743%10
                    fe80::218:93ff:fe0f:172%10
                    fe80::218:45ff:febf:a65a%10
                    fe80::218:55ff:fe21:76bc%10
                    fe80::218:56ff:fe0c:158%10
                    fe80::218:42ff:fede:63e8%10
                    fe80::218:97ff:fe96:3f0d%10
                    fe80::218:60ff:fe56:ce61%10
                    fe80::218:3ff:fe80:2b66%10
                    fe80::218:34ff:fe83:9873%10
                    fe80::218:9dff:feb8:6c6c%10
                    fe80::218:19ff:fe7b:7644%10
                    fe80::218:a5ff:fe29:173d%10
                    fe80::218:c8ff:fe21:3734%10
                    fe80::218:3cff:fe36:42b2%10
                    fe80::218:b5ff:fe6c:afcd%10
                    fe80::218:4fff:fec6:1a86%10
                    fe80::218:10ff:fe28:2a52%10
                    fe80::218:f2ff:fe7b:aba1%10
                    fe80::218:b6ff:fe74:afd0%10
                    fe80::218:c6ff:fe03:8cf0%10
                    fe80::218:bff:fec5:d4b6%10
                    172.16.7.52

Tunnel adapter Local Area Connection* 6:
```

Fig. 8. Multiple addresses configuration in case of bogus prefixes

One possible solution is the web server does not accept a request until and unless it is a complete HTTP request. As a result, the partial requests having only a fragment of HTTP header is rejected by the server.

Another solution which can be implemented is to involve some sort of mathematical puzzle computation on the client side before establishing a connection. This will result in a computation expense at client side because of multiple connections due to which it cannot send numerous requests.

C. Defenses against IPv6 Neighbor Discovery Protocol Attacks

Xinyu Yang et al.[3] presented that IPSec can be implemented to prevent these attacks. They showed that IPSec is capable of detecting IP Spoofing attack due to which all the attacks using IP spoofing technique can easily be prevented. However, they concluded that attacks based on the attacker's true identity cannot be defended using IPSec.

Rohan D. Doshi et al.[16] proposed a technique to prevent bad prefixes attack in IPv6 stateless address auto-configuration Protocol. First, it is checked in the Neighbor Cache that a node, *Y* is reachable or not. If it is not reachable, *X* sends the packets via default gateway destined to *Y* irrespective of *Y* being on-link or off-link. Unreachability of Y may be there because of either the destination host is down or the network is under attack. If unreachability is because *Y* is down, gateway will return ICMP neighbor unreachable error message. Otherwise if alternate route to *Y* exists, gateway will send packets to *Y* and thus communication will be restored.

IPv6 Router Advertisement Guard (RFC6105) [17] stated another way to prevent entry of rogue Router Advertisments within a network. It describes a framework where network segments are designed around a single L2-switching device or a set of L2-switching devices capable of identifying rouge RAs and blocking them. The framework can span the spectrum from basic to advance. The framework suggests that the port of the L2 device is statically instructed to forward or not to forward RAs received from the connected device. Also, a criterion is used by the L2 device to dynamically validate or invalidate a received RA.

The table I shows a summary of different countermeasures.

TABLE I. COUNTERMEASURES

Attacks	Possible Countermeasures
Layer-4 Attacks	• Preventing IP Spoofing Attacks • IPSec • Usage of IDS and NIDS • Analyzing pattern of the flow Entries
Layer-7 Attacks	• Only Complete HTTP Requests Acceptance. • Mathematical computations on client-side before connection establishment.
IPv6 N.D. Protocol Attacks	• IPSec • Usage of Default Gateway to route traffic for on-link or off-link devices • Application of RFC6105 framework.

VIII. CONCLUSION

In this paper, we provide classification of DoS and DDoS attacks which are possible under IPv4 and IPv6. Along with this, we analyzed experimental results of attacks and their traffic generation with the help of network analyzer's and other utilities' screenshots. We named some of the tools which can be used for these attacks. Also, we gave an overview of the countermeasures used for different types of attacks. For our experiment, we used different tools and utilities to analyze the traffic and understand the complete practical scenario. This paper provides a foundation for DoS attacks possible in different scenarios along with their countermeasures.

We are sure that this study is useful for others who want to investigate zero-day DoS attacks, improved techniques to make these attacks more effective and new techniques and efficient algorithms to develop mitigation strategies against DoS/DDoS attacks.

REFERENCES

[1] T. Narten, E. Nordmark and W. Simpson, Neighbor Discovery for IP Version 6 (IPv6), RFC2461, Internet Engineering Task Force, December 1998.

[2] Long Zeng , Analyse of DDoS attacks , http://www.icsc.ncku.edu.tw/_doc/DDOS.pdf .

[3] Xinyu Yang, Ting Ma and Yi Shi, "Typical DoS/DDoS Threats under IPv6", ICCGI '07, Proceedings of the International Multi-Conference on Computing in the Global Information Technology.

[4] Wireshark: http://www.wireshark.org/

[5] Kali Linux: http://www.kali.org/

[6] Jon Postel,Transmission Control Protocol, RFC793, Internet Engineering Task Force ,1981.

[7] A.Conta and S.Deering, Internet Control Message Protocol(ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, RFC2463, Internet Engineering Task Force, December 1998.

[8] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC2616, Internet Engineering Task Force, June 1999.

[9] Slowloris: http://ckers.org/slowloris/

[10] Apache: httpd.apache.org/

[11] S. Thomson and T. Narten, IPv6 Stateless Address Autoconfiguration, RFC2462, Internet Engineering Task Force, December 1998.

[12] Pekka Nikander and James Kempf, IPv6 Neighbor Discovery (ND) Trust Models and Threats, RFC3756, Internet Engineering Task Force, May 2004.

[13] Stephen Kent and Randall Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, Internet Engineering Task Force, November 1998.

[14] Snort: http://www.snort.org/

[15] Ahmad Sanmorino and Setiadi Yazid, "DDoS Attack Detection Method and Mitigation Using Pattern of the Flow", ICoICT'13, International Conference of Information and Communication Technology.

[16] Rohan D. Doshi and B. R. Chandavarkar, "Preventing Bad Prefixes Attack in IPv6 Stateless Address Auto-configuration Protocol", ICIIS'12, International Conference on Industrial and Information Systems

[17] Levy-Abegnoli, G. Van de Velde, C. Popoviciu and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC6105, Internet Engineering Task Force, February 2011.