# IPv6 Security and Forensics

Vincent Nicolls[1,2], Nhien-An Le-Khac[2], Lei Chen[3], Mark Scanlon[2]

[1]Federal Computer Crime Unit,
Brussels,
Belgium.
*vincent.nicolls@fccu.be*

[2]School of Computer Science,
University College Dublin,
Dublin, Ireland.
*{an.lekhac, mark.scanlon}@ucd.ie*

[3]College of Engineering and
Information Technology,
Georgia Southern University,
USA.
*lchen@georgiasouthern.edu*

***Abstract*—IPv4 is the historical addressing protocol used for all devices connected worldwide. It has survived for over 30 years and has been an integral part of the Internet revolution. However, due to its limitation, IPv4 is being replacing by IPv6. Today, IPv6 is more and more widely used on the Internet. On the other hand, criminals are also well aware of the introduction of IPv6. They are continuously seeking new methods to make profit, hiding their activities, infiltrate or exfiltrate important data from companies. The introduction of this new protocol may provide savvy cybercriminals more opportunities to discover new system vulnerabilities and exploit them. To date, there is little research on IPv6 security and forensics in the literature. In this paper, we look at different types of IPv6 attacks and we present a new approach to investigate IPv6 network attack with case studies.**

***Index Terms*—IPv6, Forensic Analysis, Network Security, IPv6 Attacks.**

## I. INTRODUCTION

IPv4 is the historical addressing protocol used for all devices connected worldwide. It has survived for over 30 years and has been an integral part of the Internet revolution. IP addresses are fundamental to the operation of the cyberspace: billions of people utilize millions of them every day and yet they are invisible in day-to-day use. The increasing number of equipment, not just computers, laptops and tablets but also security cameras and mobile phones among others, has gradually and dramatically reduced the number of available IPv4 addresses, which led to the development of IPv6 [1].

Recently, the use of IPv6 is increasing significantly. For example, Google collects statistics about IPv6 adoption in the Internet on an ongoing basis. They are continuously measuring the availability of IPv6 connectivity among Google users. The graph in Figure **Error! Bookmark not defined.** shows the percentage of users that access Google over IPv6.

With new technology comes new challenges, new threats, new crimes and a need for new skill sets. The advent of a new IPv6 Internet is an important challenge for digital forensic investigators, security researchers and network administrators. In general, they have a good understanding of the IPv4 system and they know how to trace and manipulate IPv4 addresses: investigating in an IPv4 world is usual and comfortable. The arrival of IPv6 brings a series of new concepts and models such as IPv6 address formats and types, a new IPv6 header and extension headers, ICMPv6 Neighbour Discovery, stateless and stateful auto-configuration, transition mechanisms, a new DNS record, NAT64, DNS64, etc. IPv6 is significantly different from what we know nowadays and will require new efforts, studies and a new understanding of this system.

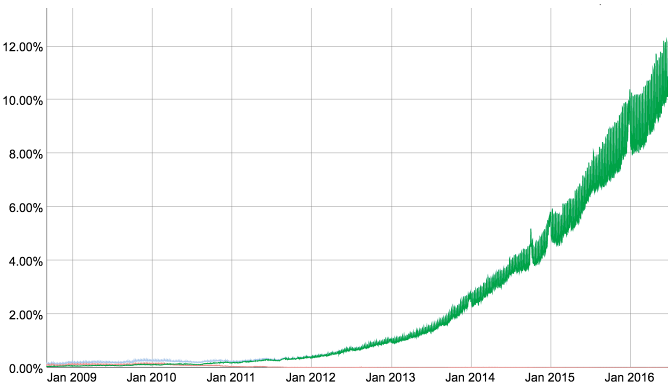


Fig. 1. Growth of IPv6 users accessing Google's services to 10.88% in June 2016 [7]

Criminals are also well aware of the introduction of IPv6. They are continuously seeking new methods to make profit, hiding their activities, infiltrate or ex-filtrate important data from companies. The introduction of this new protocol may give them new opportunities to find out new system vulnerabilities and exploit them. There must be also a number of important security problems and threats that we still ignore. It is crucial for investigators to understand the security issues that come with IPv6.

The main goal of this paper is to provide digital forensic investigators with an introduction to IPv6 forensics and its investigations. There are two main parts to this document. The first presents some basic concepts and key features of Ipv6 that are essential to navigate comfortably in IPv6. The second is more practically focused. A manner to get connected to the IPv6 Internet is presented, IPv6 address manipulation is presented and IPv6-ready tools are discussed. Regular expressions are explored for different types and formats of IPv6 addresses. Next we explore techniques that can be employed for reconnaissance of an IPv6 network or to compromise it. As Vyncke says "The earlier you think about IPv6, the better you are" [2].

The rest of this paper is organized as follows: Section II, we discuss on different ways to set up a forensic lab for analysing IPv6. We show different attack patterns and forensic analysis to IPv6 network in Section III. Finally, we conclude in Section IV.

## II. IPv6 Analysis

Network forensic activities will still require Internet IPv6 connectivity (collecting evidence from remote IPv6 services, `traceroute`, `tracepath`, etc.). There are two options for getting it: (i) get a dual stack or native IPv6 connection to the Internet; (ii) get tunnelled IPv6 connectivity.

### *A. Dual stack or native IPv6:*

Few ISPs have started providing dual stack or native IPv6 connections to end users but if you are lucky you have already one.

1) *Dual stack:* end customers will either receive both an IPv4 and an IPv6 address (or they will receive an IPv6 prefix). No tunnel mechanism is required, as can be seen in Figure 2.

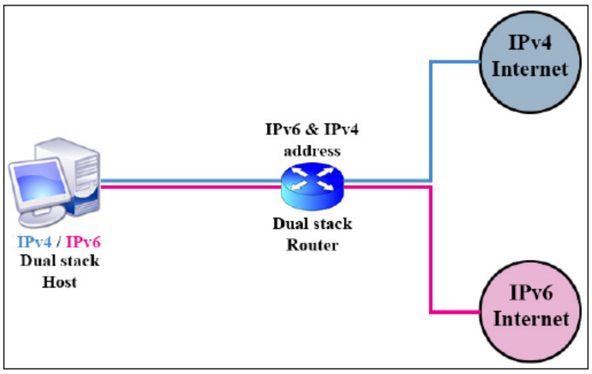


Fig. 2. Dual stack

2) *Native IPv6/DS-Lite:* end customers will receive IPv6 addresses only. For example, access to the Internet can be realized via Dual Stack Lite (DS-Lite). In this scenario, IPv4 packets are tunnelled via IPv6 to a Carrier Grade NAT (CGN) gateway and are then sent to IPv4 addresses in the Internet. This is necessary as long as there are end devices which solely use IPv4, as can be seen in Figure 3.

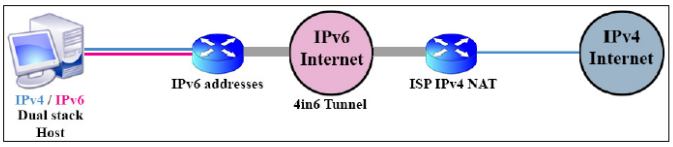


Fig. 3. Native IPv6/DS-Lite

### *B. Tunnelled IPv6 Connection:*

There are many tunnel brokers59, which provide encapsulated IPv6 connectivity over existing IPv4 network, to end users. Tunnels brokers enable us to reach the IPv6 Internet one by tunnelling over existing IPv4 connections from IPv6 enabled host or router to one of tunnel brokers' IPv6 routers. Some of the most famous are: sixXs, gogo6, Hurricane Electric. Since end customers only receive IPv4 addresses from their providers, they require a tunnel broker to access the IPv6 Internet, as can be seen in Figure 4. The IPv4 packets are transported to the tunnel broker by a tunnel mechanism such as 6in4 or 6to4. The customer's router packages the packets for the IPv6 Internet into IPv4 packets, and these are unpacked again by the tunnel broker. In the case of a 6in4 tunnel the client is given an IPv6 address by the tunnel broker. In the case of a 6to4 tunnel, no tunnel broker is required. Obtain a tunnelled IPv6 connection using a tunnel broker service is very straightforward.

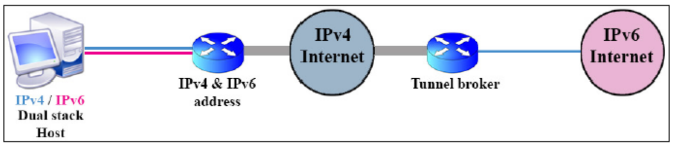


Fig. 4. Tunnelled IPv6 connection

### *C. Setting up an IPv6 tunnel using sixXs Tunnel Broker service (6in4 tunnel):*

*IPv6-ready kernel:* The following steps are for Debian Linux setups. Most modern distribution should support IPv6 out of the box. To determine if your system has an IPv6 address configured for a particular interface, use the `ifconfig` or `ip` command. If the system output includes statements containing the string "inet6", then IPv6 support is currently enabled. If the output does not contain this string, you must enable IPv6 support.

## III. Attacks Against IPv6 Networks

In this section, we discuss about different types of attacks and how hackers could exploit IPv6 networks to gather sensible information. Our goal is to show the main principles of these attacks. It is crucial for investigators to understand the security issues that come with IPv6. The test laboratory was mainly constituted of a Kali Linux (representing the attacker), a Windows 7 SP1 and an Ubuntu 14.04 LTS (the latter two representing the victims).

### *A. Reconnaissance:*

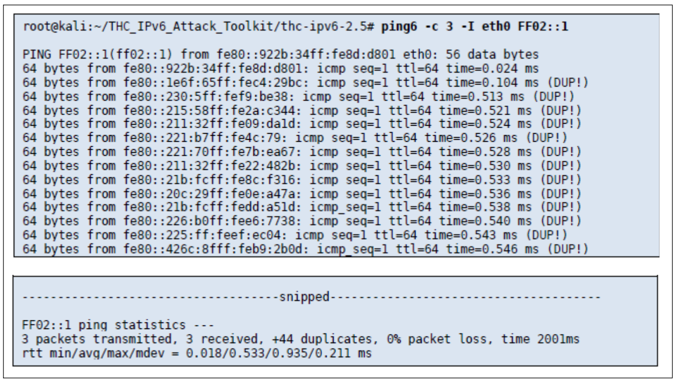
We describe how traditional address scanning techniques apply to IPv6 networks and illustrate how IPv6-enabled systems can be compromised and then we explore some additional techniques that can be employed for IPv6 network reconnaissance. So, we deal with finding active hosts on the IPv6 network. Apparently, IPv6 offers a much larger address space than IPv4. The standard /64 IPv6 subnets can accommodate approximately 18,446,744,073,709,551,616 hosts, thus it would require at least 18,446,744,073,709,551,616 packets to try to discover live systems with sequential probes. As a result, it is it is widely assumed that it is not feasible to sequentially probe IPv6 addresses. New techniques are being developed to gather information about IPv6 networks. Let's consider two techniques: `ping6` and `nmap`.

*1)* `ping6`*:* IPv6 Neighbour Discovery makes use of a number of multicast addresses in order to reach all local nodes of a given type. Table 1 enumerates the most useful of these addresses. We can perform local IPv6 node discovery by using multicast addresses and the Linux commands `ip` in conjunction with `ping6`.

TABLE I. MULTICAST IPV6 ADDRESSES

| Table Head | Table Column Head |
|---|---|
| FF01::1 | This address reaches all node-local IPv6 nodes |
| FF01::2 | This address reaches all local node-IPv6 routers |
| FF02::1 | This address reaches all link-local IPv6 nodes |
| FF02::2 | This address reaches all link-local IPv6 routers |
| FF05::2 | This address reaches all site-local routers |

The routers do not forward any packets with Link-Local source or destination addresses to other links. Therefore, all attacks that send messages to multicast addresses within the link-local scope are only applicable if the hacker or investigator resides on the local area network. To detect all link-local IPv6-active hosts on a link we use `ping6` to the link-local all-node multicast address, as shown in Figure 5.

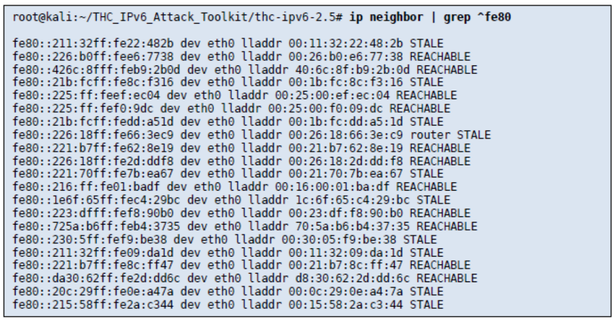


```
root@kali:~/THC_IPv6_Attack_Toolkit/thc-ipv6-2.5# ping6 -c 3 -I eth0 FF02::1

PING FF02::1(ff02::1) from fe80::922b:34ff:fe8d:d801 eth0: 56 data bytes
64 bytes from fe80::922b:34ff:fe8d:d801: icmp seq=1 ttl=64 time=0.024 ms
64 bytes from fe80::1e6f:65ff:fec4:29bc: icmp seq=1 ttl=64 time=0.104 ms (DUP!)
64 bytes from fe80::230:5ff:fef9:be38: icmp seq=1 ttl=64 time=0.513 ms (DUP!)
64 bytes from fe80::215:58ff:fe2a:c344: icmp seq=1 ttl=64 time=0.521 ms (DUP!)
64 bytes from fe80::211:32ff:fe09:da1d: icmp seq=1 ttl=64 time=0.524 ms (DUP!)
64 bytes from fe80::221:b7ff:fe4c:79: icmp seq=1 ttl=64 time=0.526 ms (DUP!)
64 bytes from fe80::221:70ff:fe7b:ea67: icmp seq=1 ttl=64 time=0.528 ms (DUP!)
64 bytes from fe80::211:32ff:fe22:482b: icmp seq=1 ttl=64 time=0.530 ms (DUP!)
64 bytes from fe80::21b:fcff:fe8c:f316: icmp seq=1 ttl=64 time=0.533 ms (DUP!)
64 bytes from fe80::20c:29ff:fe0e:a47a: icmp seq=1 ttl=64 time=0.536 ms (DUP!)
64 bytes from fe80::21b:fcff:fedd:a51d: icmp_seq=1 ttl=64 time=0.538 ms (DUP!)
64 bytes from fe80::226:b0ff:fee6:7738: icmp seq=1 ttl=64 time=0.540 ms (DUP!)
64 bytes from fe80::225:ff:feef:ec04: icmp seq=1 ttl=64 time=0.543 ms (DUP!)
64 bytes from fe80::426c:8fff:feb9:2b0d: icmp_seq=1 ttl=64 time=0.546 ms (DUP!)
```

```
-----------------------------------snipped-------------------------------------

FF02::1 ping statistics ---
3 packets transmitted, 3 received, +44 duplicates, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.018/0.533/0.935/0.211 ms
```

Fig. 5. ping6 to the link-local all-node multicast address.

FF02::1 (ip6-allnodes) is short for FF02:0:0:0:0:0:0:1, which is a special link-local multicast address for discovering all link-local hosts (DUP! means duplicate packet, the ping program expects only one answer at a time then subsequent replies are treated as duplicates). There are only Link-Local addresses because the ping program used the Link-Local address. All nodes replied with messages sent from their Link-Local addresses. Unlike in IPv4, where replies to a ping on the broadcast address can be disabled, in IPv6 this behaviour cannot be disabled except by local firewalling. Now we have connected to the LAN IPv6 hosts they are in the IPv6 neighbour table (the nodes have been contacted and cached in the neighbour table). We can read this table with the `ip` command, as can be seen in Figure 6:



```
root@kali:~/THC_IPv6_Attack_Toolkit/thc-ipv6-2.5# ip neighbor | grep ^fe80

fe80::211:32ff:fe22:482b dev eth0 lladdr 00:11:32:22:48:2b STALE
fe80::226:b0ff:fee6:7738 dev eth0 lladdr 00:26:b0:e6:77:38 REACHABLE
fe80::426c:8fff:feb9:2b0d dev eth0 lladdr 40:6c:8f:b9:2b:0d REACHABLE
fe80::21b:fcff:fe8c:f316 dev eth0 lladdr 00:1b:fc:8c:f3:16 STALE
fe80::225:ff:feef:ec04 dev eth0 lladdr 00:25:00:ef:ec:04 REACHABLE
fe80::225:ff:fef0:9dc dev eth0 lladdr 00:25:00:f0:09:dc REACHABLE
fe80::21b:fcff:fedd:a51d dev eth0 lladdr 00:1b:fc:dd:a5:1d STALE
fe80::226:18ff:fe66:3ec9 dev eth0 lladdr 00:26:18:66:3e:c9 router STALE
fe80::221:b7ff:fe62:8e19 dev eth0 lladdr 00:21:b7:62:8e:19 REACHABLE
fe80::226:18ff:fe2d:ddf8 dev eth0 lladdr 00:26:18:2d:dd:f8 REACHABLE
fe80::221:70ff:fe7b:ea67 dev eth0 lladdr 00:21:70:7b:ea:67 STALE
fe80::216:ff:fe01:badf dev eth0 lladdr 00:16:00:01:ba:df REACHABLE
fe80::1e6f:65ff:fec4:29bc dev eth0 lladdr 1c:6f:65:c4:29:bc STALE
fe80::223:dfff:fef8:90b0 dev eth0 lladdr 00:23:df:f8:90:b0 REACHABLE
fe80::725a:b6ff:feb4:3735 dev eth0 lladdr 70:5a:b6:b4:37:35 REACHABLE
fe80::230:5ff:fef9:be38 dev eth0 lladdr 00:30:05:f9:be:38 STALE
fe80::211:32ff:fe09:da1d dev eth0 lladdr 00:11:32:09:da:1d STALE
fe80::221:b7ff:fe8c:ff47 dev eth0 lladdr 00:21:b7:8c:ff:47 REACHABLE
fe80::da30:62ff:fe2d:dd6c dev eth0 lladdr d8:30:62:2d:dd:6c REACHABLE
fe80::20c:29ff:fe0e:a47a dev eth0 lladdr 00:0c:29:0e:a4:7a STALE
fe80::215:58ff:fe2a:c344 dev eth0 lladdr 00:15:58:2a:c3:44 STALE
```

Fig. 6. Result from the `ip` command.

The neighbour table is temporary and entries disappear in a few minutes when there is no traffic to them. By using `ping6` and `ip` commands, we discovered 22 unique active hosts. We can ping each of these neighbours. Not all IPv6 nodes reply to an echo-request message sent to a multicast address. An echo reply to a multicast echo-request is not mandatory. For example, Windows 7 does not reply to a ping command.

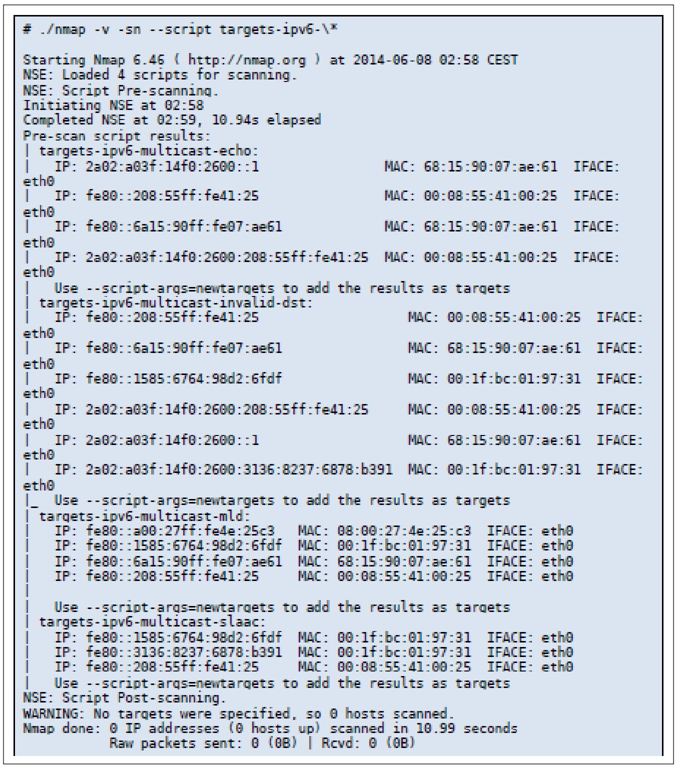
*2)* `nmap`*:* Nmap version 6.0 includes full support for IPv6 networks including (i) raw IPv6 port scanning (IPv6 echo requests, TCP/UDP discovery packets, SYN scan, UDP scan, ACK scan, etc.); (ii) IPv6 protocol scan; (iii) IPv6 `traceroute` implementation; (iv) OS detection.

Since the IPv6 address space is too large to brute scan in general, the developers of nmap researched IPv6 host discovery techniques for finding all the machines on a local network. They ended up implementing four techniques they found the most effective. They are all implemented as NSE92 (Nmap Scripting Engine) scripts which can simply print out discovered addresses or (if requested) add them to Nmap's target queue. Four techniques were developed:

- ***targets-ipv6-multicast-echo***: This technique sends an ICMPv6 echo request packet to the all-nodes link-local multicast address (FF02::1) to discover responsive hosts on a LAN without an individual ping to each IPv6 address. When ICMPV6 echo response packets are received, it collects the IPv6 addresses that they come from and mark those hosts as potential scan targets (if new hosts are discovered, the script add them to the scan queue). This is a technique which uses the protocols as designed (just like using ICMPv4 echo request packets for host discover) and it is quite effective.
- ***targets-ipv6-multicast-invalid-dst***: This technique sends an ICMPv6 packet with an invalid extension header to the all-nodes link-local multicast address to discover (some) available hosts on the LAN. Any hosts replying with an ICMPv6 parameter problem packet can be marked as up and available for potential scanning.
- ***targets-ipv6-multicast-mld***: This technique attempts to discover available IPv6 hosts on the LAN by sending an MLD (Multicast Listener Discovery) query to the link-local multicast address and listening for any responses (the query's maximum response delay is set to 0 to provoke hosts to respond immediately rather

than waiting for other responses from their multicast group). The default timeout to wait for responses is 10 seconds.

- ***targets-ipv6-multicast-slaac:*** This technique sends an ICMPv6 Router Advertisement packet with a random address prefix, causing hosts to begin stateless address auto-configuration (SLAAC) and sends a solicitation for their newly configured address97 (as part of duplicate address detection). The remote address is guessing by combining the link-local prefix of the interface with the interface identifier in each of the received solicitations. An ICMPv6 neighbour discovery probe can then be used to verify that the guessed addresses are correct.



```
# ./nmap -v -sn --script targets-ipv6-\*

Starting Nmap 6.46 ( http://nmap.org ) at 2014-06-08 02:58 CEST
NSE: Loaded 4 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 02:58
Completed NSE at 02:59, 10.94s elapsed
Pre-scan script results:
| targets-ipv6-multicast-echo:
|   IP: 2a02:a03f:14f0:2600::1                  MAC: 68:15:90:07:ae:61  IFACE:
eth0
|   IP: fe80::208:55ff:fe41:25                  MAC: 00:08:55:41:00:25  IFACE:
eth0
|   IP: fe80::6a15:90ff:fe07:ae61               MAC: 68:15:90:07:ae:61  IFACE:
eth0
|   IP: 2a02:a03f:14f0:2600:208:55ff:fe41:25  MAC: 00:08:55:41:00:25  IFACE:
eth0
|   Use --script-args=newtargets to add the results as targets
| targets-ipv6-multicast-invalid-dst:
|   IP: fe80::208:55ff:fe41:25                     MAC: 00:08:55:41:00:25  IFACE:
eth0
|   IP: fe80::6a15:90ff:fe07:ae61                  MAC: 68:15:90:07:ae:61  IFACE:
eth0
|   IP: fe80::1585:6764:98d2:6fdf                  MAC: 00:1f:bc:01:97:31  IFACE:
eth0
|   IP: 2a02:a03f:14f0:2600:208:55ff:fe41:25     MAC: 00:08:55:41:00:25  IFACE:
eth0
|   IP: 2a02:a03f:14f0:2600::1                     MAC: 68:15:90:07:ae:61  IFACE:
eth0
|   IP: 2a02:a03f:14f0:2600:3136:8237:6878:b391  MAC: 00:1f:bc:01:97:31  IFACE:
eth0
|_  Use --script-args=newtargets to add the results as targets
| targets-ipv6-multicast-mld:
|   IP: fe80::a00:27ff:fe4e:25c3   MAC: 08:00:27:4e:25:c3  IFACE: eth0
|   IP: fe80::1585:6764:98d2:6fdf  MAC: 00:1f:bc:01:97:31  IFACE: eth0
|   IP: fe80::6a15:90ff:fe07:ae61  MAC: 68:15:90:07:ae:61  IFACE: eth0
|   IP: fe80::208:55ff:fe41:25     MAC: 00:08:55:41:00:25  IFACE: eth0
|
|   Use --script-args=newtargets to add the results as targets
| targets-ipv6-multicast-slaac:
|   IP: fe80::1585:6764:98d2:6fdf  MAC: 00:1f:bc:01:97:31  IFACE: eth0
|   IP: fe80::3136:8237:6878:b391  MAC: 00:1f:bc:01:97:31  IFACE: eth0
|   IP: fe80::208:55ff:fe41:25     MAC: 00:08:55:41:00:25  IFACE: eth0
|   Use --script-args=newtargets to add the results as targets
NSE: Script Post-scanning.
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 10.99 seconds
           Raw packets sent: 0 (0B) | Rcvd: 0 (0B)
```

Fig. 7.  Nmap experience.

Nmap's developers recommend specifying all four (since each techniques may discover a different set of hosts). Running all scripts at once reveals the most different IPv6 addresses, as can be seen in Figure 7.

### *B. The Hackers Choice IPv6 Attack Toolkit:*

We THC-IPv6 is a toolkit that attacks the inherent protocol weaknesses of IPv6 and ICMPv6 and it includes an easy to use packet factory library. This toolkit is maintained by Van Hause of *The Hacker's Choice* and its primary purpose is to test network devices with active IPv6 interfaces. With over 50 separate tools98 the toolkit allows you to perform such tasks like detecting new IPv6 devices which join the network, you can run a script to automatically scan these systems (detct-new-ip6), finding all local IPv6 systems (alive6), announcing yourself as a router on the network, with the highest priority (fake_router6), test against a target known IPv6 vulnerabilities (exploit6), etc. Running the tools without options will give us help and command line options.

Parasite6 is part of THC IPv6 Attack Toolkit. This program is an ICMPv6 Neighbour Solicitation/Advertisement spoofer103 (redirecting all local traffic to your own system by answering falsely to Neighbour Solicitation requests). It puts you as man-in-the-middle, same as "ARP spoofer" for IPv4. IPv6 is running as a valid protocol stack on most computers that come to companies today. So unless they have taken specific steps to disable it, we can leverage IPv6 and how it operates to go ahead and compromise a network. Let's assume a Windows 7 machine boots up on the network. If it is running an IPv6 protocol stack, it is going to be looking for a router.

There is a router in the network and it gets a Router Advertisement, it figures out what network it is on, run automatic configuration to get its host ID of its IPv6 address via SLAAC (or it may have learned that it should be using DHCPv6 stateful services through the Router Advertisement messages or maybe just DHCP stateless services looking for a DNS server from DHCP). At the end, it has got an IPv6 address. The machine wants to send out a packet to the rest of the Internet. This device learns the MAC address of the router by the use of the Neighbour Discovery Protocol. It sends a Neighbour Solicitation to the solicited node multicast group address and the router responds back with a Neighbour Advertisement (solicitations are asking and advertisements are giving the information that was asked for). Parasite6 is going to be running on our interception device and after we enable it, it will listen for any solicitations and answer them. But instead of answering with the correct layer two information, it is going to answer with its own MAC address. It is also possible to specify a fake MAC address to use, instead of using our own on the interception machine. The "-l" option is going to loop and resent every five seconds the poisoned information for every target that it is spoofing. The "-R" is going to try to poison the destination of the solicitation. We are going to use both of these options. If we launch the attack at this moment, we have not a man in the middle attack but a denial of service attack because the frames at layer two will be forward to my interception machine but we are not going to forward them by default on to the correct destination. So we also need to enable IPv6 forwarding on my Interception machine. Figure 8 describes the topology we set up for an attack.

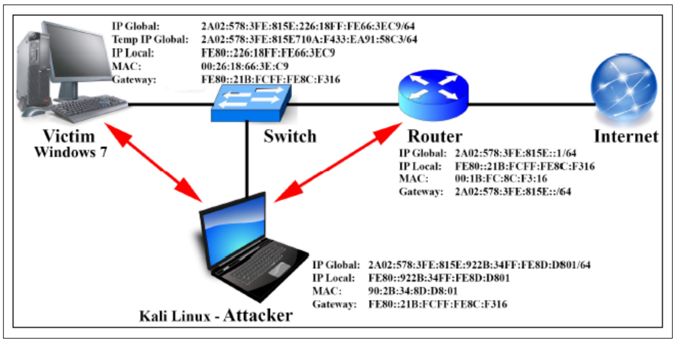
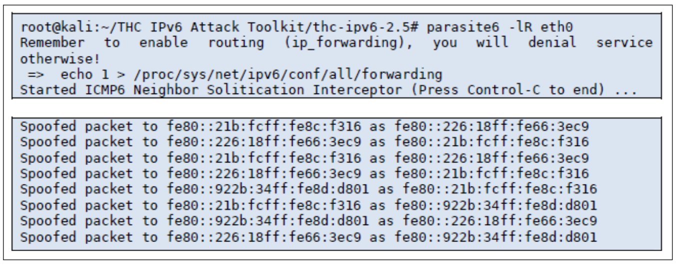

IP Global: 2A02:578:3FE:815E:226:18FF:FE66:3EC9/64
Temp IP Global: 2A02:578:3FE:815E710A:F433:EA91:58C3/64
IP Local: FE80::226:18FF:FE66:3EC9
MAC: 00:26:18:66:3E:C9
Gateway: FE80::21B:FCFF:FE8C:F316
Victim
Windows 7
Switch
Router
Internet
IP Global: 2A02:578:3FE:815E::1/64
IP Local: FE80::21B:FCFF:FE8C:F316
MAC: 00:1B:FC:8C:F3:16
Gateway: 2A02:578:3FE:815E::/64
IP Global: 2A02:578:3FE:815E:922B:34FF:FE8D:D801/64
IP Local: FE80::922B:34FF:FE8D:D801
MAC: 90:2B:34:8D:D8:01
Gateway: FE80::21B:FCFF:FE8C:F316
Kali Linux - Attacker

Fig. 8.  Attack topology. *(figure caption)*

In fact, the command `netsh` interface ipv6 show neighbours "Local Area Connection 2" show us the IPv6 Neighbour Cache104 of our Victim. It shows us the Layer 3 to Layer 2 mapping.

The attack: Parasite 6 is now going to listen for any Neighbour Solicitation messages that it sees and responds to them with its own MAC address, as can be seen in Figure 9. The Attacker's computer has a MAC address of 90:2B:34:8D:D8:01 and the legitimate Router has a MAC address of 00:1B:FC:8C:F3:16. If we inspect again the Neighbour Cache of the victim's computer we can observe that the Physical Address of its gateway is the MAC address belonging to the Attacker's computer and not the one of the legitimate Router, as can be seen in Figure 10.



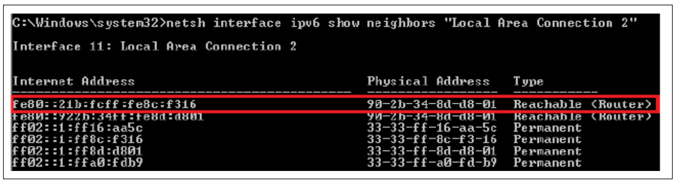
```
root@kali:~/THC IPv6 Attack Toolkit/thc-ipv6-2.5# parasite6 -lR eth0
Remember  to  enable  routing  (ip_forwarding),  you  will  denial  service
otherwise!
 =>  echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
Started ICMP6 Neighbor Solitication Interceptor (Press Control-C to end) ...
```

```
Spoofed packet to fe80::21b:fcff:fe8c:f316 as fe80::226:18ff:fe66:3ec9
Spoofed packet to fe80::226:18ff:fe66:3ec9 as fe80::21b:fcff:fe8c:f316
Spoofed packet to fe80::21b:fcff:fe8c:f316 as fe80::226:18ff:fe66:3ec9
Spoofed packet to fe80::226:18ff:fe66:3ec9 as fe80::21b:fcff:fe8c:f316
Spoofed packet to fe80::922b:34ff:fe8d:d801 as fe80::21b:fcff:fe8c:f316
Spoofed packet to fe80::21b:fcff:fe8c:f316 as fe80::922b:34ff:fe8d:d801
Spoofed packet to fe80::922b:34ff:fe8d:d801 as fe80::226:18ff:fe66:3ec9
Spoofed packet to fe80::226:18ff:fe66:3ec9 as fe80::922b:34ff:fe8d:d801
```

Fig. 9. Neighbour Solicitation message.

We could look at the data by using Wireshark. For example, the filter "ipv6" display the IPv6 based traffic and "ipv6.addr eq 2A02:578:3FE:815E710A:F433:EA91:58C3" display specific IPv6 address. Let's suppose that the victim has just logged in a non-secured website (like a simple HTTP website) when parasite6 was running into the network. At the same time, Wireshark was running on the Attacker's computer and we could look for IPv6 packets containing private or confidential information (such as emails, passwords, Internet traffic, etc.).

Parasite6 can quickly and easily implement an IPv6 man in the middle attack on a network (make sure that you enable the IPv6 forwarding function so you do not turn this on a Denial of Service Attack against everybody on the network as opposed to a man in the middle attack).



```
C:\Windows\system32>netsh interface ipv6 show neighbors "Local Area Connection 2"

Interface 11: Local Area Connection 2


Internet Address                              Physical Address   Type
--------------------------------------------  -----------------  -----------
fe80::21b:fcff:fe8c:f316                      90-2b-34-8d-d8-01  Reachable (Router)
fe80::922b:34ff:fe8d:d801                     90-2b-34-8d-d8-01  Reachable (Router)
ff02::1:ff16:aa5c                             33-33-ff-16-aa-5c  Permanent
ff02::1:ff8c:f316                             33-33-ff-8c-f3-16  Permanent
ff02::1:ff8d:d801                             33-33-ff-8d-d8-01  Permanent
ff02::1:ffa0:fdb9                             33-33-ff-a0-fd-b9  Permanent
```

Fig. 10. MAC address belonging to the Attacker's computer.

### *C. Other IPv6 security considerations:*

Attackers are already using IPv6 networks to attack users on IPv4 networks. According to an article written by James Lyne110, head of Global Security Research at Sophos, many cyber-criminals have already made the switch from IPv4 to IPv6. Malwares with IPv6-based command-and-control capabilities are growing in number. IPv6 brings no change, virus and worms will adapt to it. Many current firewalls are not configured to filter IPv6 packets. They focus uniquely on IPv4 packets. As a result, the system is completely exposed to IPv6 attacks and the attackers can bypass securities without detection of any sort at any location. As an example, Unix-based operating systems usually have two different firewalls: `iptables` for IPv4 and `ip6tables` for IPv6. If you have not correctly configured ip6tables and if an attacker gives you unexpected connectivity, then you become wide open to the Internet. We have also heard that IPv6 is more secure than IPv4 because IPsec was made mandatory for IPv6. IPsec provides channel security at the Internet layer, making it possible to provide secure communication for all communication flows at the IP layer between pairs of internet nodes. Actually, IPsec support is mandatory in IPv6, IPsec use is not. IPsec was a mandatory specification of the base IPv6 protocol suite, but has since been made optional (see RFC 6434 "IPv6 Node Requirements", Section 11). The IPv6 stack requirement for IPsec is passed from "MUST" to "SHOULD". At this time, IPv6 is mostly provided thanks to transition methods:

- Tunnelling technologies to transport IPv6 over IPv4 is a potential source of confusion, misconfiguration and security breaches. As an example, the IPv6 interim mechanism 6to4 uses automatic IPv6-over-IPv4 tunnelling to interconnect IPv6 networks. The 6to4 relays receive IPv6 traffic encapsulated in IPv4. They accept IPv4 packets from any node, decapsulate the IPv6 packets and then forward them to their IPv6 destination. There is absolutely no control on the IPv4 source address of the packet received by the relay because it could come from any remote site. This characteristic enables Denial of Service (DoS) and for nodes to spoof IPv6 addresses (for more details about 6to4 security, see RFC 3964: "Security considerations for 6to4). Traffic tunnelling will make network security systems less likely to identify attack.
- The NAT64 device is a potential victim of different types of attack. The NAT64 has a limited number of resources and can be a victim of DoS attacks. For example, it has a limited number of IPv4 addresses that it uses to create the bindings. By sending IPv6 packets with different source IPv6 transport addresses, it is possible for an attacker to consume all the IPv4 transport addresses (more details on attacks on NAT64 in RFC 6146: "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", Section 5.3).

## IV. CONCLUSION

In this paper, we outlined a study of IPv6 network forensics. The size and the various types of IPv6 addresses are impressive, but this could be a benefit and an opportunity for digital forensic investigators. Investigating, manipulating and filtrating server logs to find out valuable information could be more complicated. Although traditional IPv4 address scanning techniques are not efficient or impossible to apply in an IPv6 network, new ways are being developed to discover live systems such as ping to multicast addresses, reduce the IPv6 search space by using specific Interface IDs, or to detect if hosts are numbered sequentially or using any regular scheme. IPv6 is not more secure than IPv4. Modern operating systems come out of the box ready and willing to use IPv6. Most of networks still have only IPv4 network and only have IPv4 monitoring and defences in place. Since the victims believe that they are not using IPv6 they do not expect IPv6 activity and attacks that make use of it.

## REFERENCES

[1] Rick Grazian. IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6, First Edition. Cisco Press, 2012.

[2] Silvia Hagen. IPv6 Essentials, Second Edition. O'Reilly Media, Inc., 2006.

[3] Scott Hogg and Eric Vyncke. IPv6 Security, First Edition. Cisco Press, 2008

[4] Testing the security of IPv6 implementations, March 2014. The report is the result of a Dutch research.

[5] Johannes Weber. Master Thesis: IPv6 Security Test Laboratory. Ruhr-University Bochum, Germany, February 25, 2013.

[6] Roman Ammann. Network Forensic Readiness: A Bottom-up Approach for IPv6 Network. Auckland University of Technology, School of Computing and Mathematical Sciences, 2012.

[7] Google IPv6 Adoption Statistics, https://www.google.com/intl/en/ipv6/statistics.html, accessed June 2016.