PROPUESTA DE MÉTODOS PARA MINIMIZAR EL IMPACTO DE LAS VULNERABILIDADES DE SEGURIDAD ASOCIADAS AL PROTOCOLO *NEIGHBOR DISCOVERY* PARA IPV6

UNIVERSIDAD PONTIFICIA BOLIVARIANA

Maestría en TIC– Seguridad Informática

Medellín

2015

PROPUESTA DE MÉTODOS PARA MINIMIZAR EL IMPACTO DE LAS VULNERABILIDADES DE SEGURIDAD ASOCIADAS AL PROTOCOLO *NEIGHBOR DISCOVERY* PARA IPV6

JUAN CAMILO DANIEL ISAZA ZAPATA

UNIVERSIDAD PONTIFICIA BOLIVARIANA

ESCUELA DE INGENIERÍAS

FACULTAD DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

MAESTRÍA EN TIC

MEDELLÍN

2015

PROPUESTA DE MÉTODOS PARA MINIMIZAR EL IMPACTO DE LAS VULNERABILIDADES DE SEGURIDAD ASOCIADAS AL PROTOCOLO *NEIGHBOR DISCOVERY* PARA IPV6

JUAN CAMILO DANIEL ISAZA ZAPATA

Trabajo de grado para optar al título de Magíster en TIC línea de Seguridad informática

Tutor
Reinaldo Mayol Arnao
PhD (c) Ingeniería Electrónica

Director
Grupo de Investigación GIDATI

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
FACULTAD DE TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES
MAESTRÍA EN TIC
MEDELLÍN
2015

NOTA DE ACEPTACION

_____

_____
_____
_____

_____
Firma
Nombre
Presidente del jurado

_____
Firma
Nombre
Presidente del jurado

Medellín, Febrero de 2015

Agradecimientos

A mi madre, mi familia y todas las personas que me rodearon y otorgaron su apoyo en los momentos más difíciles.

TABLA DE CONTENIDO

Agradecimientos..... 5
IMAGENES 8
TABLAS 11
1 RESUMEN. 12
2 PLANTEAMIENTO DEL PROBLEMA. 13
3 ESTADO DEL ARTE. 14
3.1 Estructura dirección IP. 14
3.2 Tipos de direcciones IP. 15
3.3 Mecanismos para la asignación de direcciones IP. 18
3.4 Asignación de dirección IP mediante DHCP. 19
3.5 Configuración automática de direcciones sin estado (SLAAC), utilizando el protocolo Neighbor Discovery para IP versión 6. 19
3.6 Detección de Direcciones Duplicadas, DAD. 21
3.7 Asignación de dirección IP Global Unicast. 22
3.8 Posibles vulnerabilidades de seguridad en el protocolo Neighbor Discovery para IPv6 23
4 PERTINENCIA Y ANTECEDENTES. 24
5 PROTOCOLO NEIGHBOR DISCOVERY PARA IPV6 A TRAVÉS DE ICMPV6. 26
5.1 Mensajes del protocolo Neighbor Discovery para IPv6. 27
5.2 Formato Router Solicitation Message. 28
5.3 Formato Router Advertisement Message. 29
5.4 Formato Neighbor Solicitation Message. 31
5.5 Formato Neighbor Advertisement Message. 32
5.6 Formato Redirect Message. 33
6 TÉCNICAS DE ATAQUE CON GRAN IMPACTO EN EL PROTOCOLO NEIGHBOR DISCOVERY. 35
6.1 Falsificación mensajes Router Advertisement. 35
6.2 Falsificación mensaje Neighbor Advertisement durante DAD. 35
6.3 Inundación Neighbor Cache. 36
6.4 Modificación del MTU. 36
6.5 Falsificación del prefijo. 36
7 PRÁCTICA DE LAS TECNICAS DE ATAQUE EN EL PROTOCOLO NEIGHBOR DISCOVERY. 37
7.1 Falsificación mensajes Router Advertisement. 37

7.2 Falsificación mensaje Neighbor Advertisement durante DAD. 44
7.3 Inundación Neighbor Cache..... 50
7.4 Falsificación del prefijo y modificación del MTU..... 58
8 MÉTODOS PARA LA PROTECCIÓN DEL PROTOCOLO NEIGHBOR DISCOVERY. 63
8.1 SEcure Neighbor Discovery (SEND). 63
8.2 Funcionamiento del protocolo SEND. 63
8.3 Opción CGA. 64
8.4 Opción Firma RSA. 65
8.5 Opciones de sello de tiempo y *nonce.* 66
8.6 Certificados Digitales. 68
8.7 Controlando ataques mediante SEND. 70
8.8 Easy-SEND. 70
8.9 WinSEND. 71
8.10 ipv6-send-cga. 72
8.11 NDprotector. 73
8.12 Funcionamiento de NDprotector. 74
9 PROTOCOLO DE PRUEBAS. 76
10 OBSERVACIONES SOBRE EL PROTOCOLO SEND. 85
11 CONCLUSIONES Y TRABAJOS FUTUROS. 86
12 BIBLIOGRAFÍA 87

IMAGENES

Imagen 1. Estructura dirección Global Unicast..... 16
Imagen 2. Estructura dirección Link Local..... 16
Imagen 3. Estructura dirección Unique Local..... 17
Imagen 4. Estructura dirección Embedded IPv4 18
Imagen 5. Generación 64 bits a partir de la MAC con EUI-64..... 21
Imagen 6. Cabecera de un datagrama ICMPv6. 26
Imagen 7. Cabecera IPv6 indicando que la información siguiente corresponde a una cabecera ICMPv6. 27
Imagen 8. Cabecera del mensaje Router Solicitation en el protocolo Neighbor Discovery para IPv6. 28
Imagen 9. Cabecera del protocolo IP anterior a la cabecera ND RS. Los números mostrados son base decimal..... 29
Imagen 10. Cabecera del mensaje Router Advertisement en el protocolo Neighbor Discovery para IPv6. 30
Imagen 11. Encabezado IP para un mensaje ICMP tipo 134. 31
Imagen 12. Cabecera del mensaje Neighbor Solicitation en el protocolo Neighbor Discovery para IPv6. 31
Imagen 13. Cabecera del protocolo IP anterior a la cabecera ND NS. Los números mostrados son base decimal..... 32
Imagen 14. Cabecera del mensaje Neighbor Advertisement en el protocolo Neighbor Discovery para IPv6..... 32
Imagen 15. Cabecera del protocolo IP anterior a la cabecera ND NA. Los números mostrados son base decimal..... 33
Imagen 16. Cabecera del mensaje redirect en el protocolo Neighbor Discovery para IPv6. 34
Imagen 17. Cabecera del protocolo IP anterior a la cabecera Redirect del protocolo ND. Los números mostrados son base decimal. 34
Imagen 18. Formato para la opción del prefijo en el mensaje 133. 35
Imagen 19. Ejecución comando ping hacia el enrutador..... 37
Imagen 20. Se muestra IP Global Unicast y puerta de enlace de la víctima. 38
Imagen 21. Tabla de enrutamiento antes del ataque. 38
Imagen 22. Inicia el ataque de falsificación mensajes router advertisement. 39
Imagen 23. Nueva dirección IP Global Unicast y puerta de enlace en la víctima luego del ataque..... 40
Imagen 24. Ejecución comando route print. 41
Imagen 25. Dirección IP link-local del atacante..... 41
Imagen 26. Enrutador oficial..... 42
Imagen 27. Ejecución comando ping posterior al ataque..... 42
Imagen 28. Captura de datos con Wireshark. 43
Imagen 29. Análisis mensaje router advertisement..... 43
Imagen 30. Configuración de la tarjeta de red en modo promiscuo. 44
Imagen 31. Ejecución comandos dos-new-ip6 eth0 y detect-new-ip6 eth0..... 45
Imagen 32. Interfaz de red "LAN" desactivada..... 45
Imagen 33. Activación tarjeta de red para la ejecución de la prueba. 46
Imagen 34. Error generado por el sistema operativo Windows 7 luego de varios intentos de asignación de IP 46

Imagen 35. Interfaz de red sin asignación de IP 47
Imagen 36. Paquetes capturados y falsificados durante el ataque. 48
Imagen 37. Paquetes capturados con Wireshark..... 48
Imagen 38. Evidencia de uno de los paquetes falsificados para engañar a la víctima. 49
Imagen 39. Asignación correcta de la dirección IP en el equipo de la víctima. 50
Imagen 40. Uso estándar en el enrutador de la red. 51
Imagen 41. Ataque de inundación de paquetes router advertisement. 52
Imagen 42. Incremento del uso del procesador. 52
Imagen 43. Interfaz del equipo víctima sin asignación de IP. 53
Imagen 44. Incremento en el uso del procesador del enrutador. 54
Imagen 45. Reinicio del enrutador..... 54
Imagen 46. Normalidad en los procesos del router luego del reinicio. 55
Imagen 47. Dirección IP reconfigurada luego de terminado el ataque, reiniciado el enrutador y el huésped víctima. 56
Imagen 48. Ejecución del comando net statistics workstation..... 56
Imagen 49. Paquetes capturados con Wireshark y evidencia de la generación de paquetes aleatorios con IP y MAC diferentes..... 57
Imagen 50. Dirección IP del equipo de la víctima..... 58
Imagen 51. Dirección IP del equipo de la víctima en consola. 59
Imagen 52. Ejecución del comando netsh interface ipv6 show subinterfaces. 59
Imagen 53. Ejecución del comando sysctl -w net.ipv6.conf.all.forwarding=1 60
Imagen 54. Ejecución comando ifconfig eth0 promisc 60
Imagen 55. Ejecución comando fake_router6 eth0 ca::8/64 fe80::a00:27ff:feb5:4dd5 fe80::a00:27ff:feb5:4dd5 1400..... 60
Imagen 56. Dirección IP asignada con el prefijo falso. 61
Imagen 57. Se muestra como se asignó el MTU de 1400 en vez de 1500 que es el original. 61
Imagen 58. Información capturada con Wireshark. Prefijo y MTU enviados falsamente. 62
Imagen 59. Formato Opción CGA. 65
Imagen 60. Formato Opción Firma RSA. 66
Imagen 61. Formato Opción Sello de tiempo. 67
Imagen 62. Formato Opción Nonce. 67
Imagen 63. Formato mensaje CPS. 68
Imagen 64. Formato mensaje CPA. 69
Imagen 65. Arquitectura de Easy-SEND. [45] 71
Imagen 66. Arquitectura WinSEND [41] 72
Imagen 67. Arquitectectura Nucleo/Usuario [53]. 73
Imagen 68. Arquitectura NDprotector [59]. 75
Imagen 69. Inicio de la ejecución del comando apt-get. 77
Imagen 70. Fin de la ejecución del comando apt-get. 78
Imagen 71. Verificación Ipv6 en Python. 78
Imagen 72. Adquisición NDprotector. 79
Imagen 73. Instalación del NDprotector 79
Imagen 74. Generación CGA IP Global Unicast. 80
Imagen 75. Generación CGA IP Link-local. 81

Imagen 76.Archivo sendd.conf con las claves registradas. 82
Imagen 77. Error durante la ejecución del servicio NDprotector. 83
Imagen 78. Comunicación con Tony Cheneau de amnesiak.org. 84

TABLAS

Tabla 1. Diferentes métodos de escribir una dirección IPv6. 14
Tabla 2. Simplificación de ceros en direcciones IPv6. 15
Tabla 3 Dirección IPv6 con prefijo de red. 15
Tabla 4. Rango de direcciones IPv6 establecido por la IANA. Tomado de [14]. ... 18
Tabla 5. Tipos de mensajes del protocolo Neighbor Discovery 20
Tabla 6. Tipos de mensajes de error utilizados en el protocolo ICMPv6. Tomado de [21]. 26
Tabla 7. Tipos de mensajes informativos utilizados en el protocolo ICMPv6. Tomado de [21]. 27
Tabla 8. Tipos de mensajes del protocolo Neighbor Discovery. 28
Tabla 9. Matriz del protocolo de pruebas. 77

1 RESUMEN.

Durante este proyecto de grado se analizaron diferentes métodos tanto de ataque como de defensa en el protocolo *Neighbor Discovery* para IPv6. Para iniciar se procedió con la realización del estado del arte de las posibles soluciones que puede contener el protocolo *Neighbor Discovery* a sus respectivos ataques. Posteriormente se elaboró el estudio detallado de cada una de las vulnerabilidades factibles que se presentan, así mismo se especificó la metodología utilizada por el protocolo *Neighbor Discovery* para su funcionamiento. Después se estableció un ambiente controlado de pruebas para verificar la calidad y el impacto de los ataques que puede llegar a sufrir el protocolo *Neighbor Discover.* En este ambiente pruebas se ejecutaron diversos ataques con éxito comprobando, mediante la práctica, las vulnerabilidades estudiadas con anterioridad. Además se explicó a profundidad el funcionamiento del protocolo SEND con sus principales características para contrarrestar las vulnerabilidades sobre el protocolo *Neighbor Discovery*. Para continuar se estudiaron las posibles soluciones que permitan minimizar los ataques realizados al protocolo en cuestión. Y para finalizar, se propuso el diseño de un protocolo de pruebas como posible solución al problema planteado.

2 PLANTEAMIENTO DEL PROBLEMA.

En la definición del protocolo IPv6, como opción de reemplazo al protocolo IPv4, se planteó la alternativa de configurar la dirección IP sin necesidad de un servidor DHCPv6 ni la utilización del método manual, aclarando que el protocolo DHCPv6 sigue existiendo en redes IPv6. Este mecanismo brinda la opción de auto-configurar la IP a cualquier dispositivo que desee conectarse a la red. Incluso posee la capacidad de detectar direcciones IP duplicadas.

Haciendo uso del protocolo *Neighbor Discovery* para IPv6 los nodos pueden generar, validar duplicidad y asignar su propia dirección IP. Sin embargo, esta implementación carece de seguridad debido a que no se brinda autenticidad y confidencialidad en los mensajes que viajan por la red relacionados con el protocolo *Neighbor Discovery.* Esta debilidad se plantea como problema principal en este trabajo de grado y se busca investigar tanto posibles ataques como mecanismos de defensa para disminuir la brecha de seguridad.

3 ESTADO DEL ARTE.

Una de las principales razones por la cual se trabaja fuertemente en el protocolo IPv6 es para aumentar la cantidad de direcciones IP. Con IPv4 la dirección IP contiene una longitud de 32 bits lo que permite obtener cerca de $4.2\ x\ 10^9$, es decir, 2^{32} direcciones IP. Mientras que con IPv6 la longitud es de 128 bits que equivalen aproximadamente a $3.4\ x\ 10^{38}$, es decir 2^{128}. Adicionalmente tiene como objetivo simplificar los campos del encabezado, mejorar el soporte a las extensiones y opciones, agregar una capa para etiquetar el flujo de paquetes y extensiones adicional para el tema de autenticación, integridad de los datos y confidencialidad [1].

Cabe apreciar que en este documento, cada vez que se mencione la palabra IP se hace referencia a IPv6, la palabra ICMP hace referencia a ICMPv6 y la palabra DHCP hace referencia a DHCPv6 a excepción que se indique lo contrario.

3.1 Estructura dirección IP.

Una dirección IP se divide en 8 partes separadas por el carácter ":" x:x:x:x:x:x:x:x, cada "x" contiene 16 bits, generalmente divididos en una cadena de 4 caracteres en lenguaje hexadecimal. Por ejemplo: 2001:0db8:0000:0b3s:cccc:dddd:eeee:aaaa [2]. Una dirección IP puede ser representada de varias maneras [3] (ver Tabla 1) en la cual todos los métodos representan la misma dirección IP, se evidencia que no es necesario escribir los ceros a la izquierda en cada bloque de 16 bits y con la sintaxis especial "::" se pueden agrupar uno o más ceros (ver Tabla 2), esta sintaxis sólo debe ser utilizada una vez en la dirección IP y se recomienda utilizarla donde más cantidad de ceros existan.

Dirección IPv6
2001:db8:0:0:1:0:0:1
2001:0db8:0:0:1:0:0:1
2001:db8::1:0:0:1
2001:db8::0:1:0:0:1
2001:0db8::1:0:0:1
2001:db8:0:0:1::1
2001:db8:0000:0:1::1
2001:DB8:0:0:1::1

Tabla 1. Diferentes métodos de escribir una dirección IPv6.

Dirección IPv6
2001:db8:0:0:0::1

2001:db8:0:0::1
2001:db8:0::1
2001:db8::1

Tabla 2. Simplificación de ceros en direcciones IPv6.

De los 128 bits que componen una dirección IPv6 una parte, al inicio, hace referencia a un prefijo de red seguido del identificador de la interfaz y al final, con el carácter "/", se define el tamaño del mismo. Su nomenclatura se representa en decimal [4]. Como se observa en la Tabla 3 Dirección IPv6 con prefijo de red., los caracteres resaltados en negrilla corresponden al prefijo de red (60 bits).

Dirección IPv6 con prefijo
2001:0DB8:0000:CD30:0000:0000:0000:0000/60
2001:0DB8::CD30:0:0:0:0/60
2001:0DB8:0:CD30::/60

Tabla 3 Dirección IPv6 con prefijo de red.

3.2 Tipos de direcciones IP.

En el protocolo IPv6 existen tres tipos de direcciones IP: *Unicast, Anycast* y *Multicast* [4]. Las direcciones *Unicast* son un identificador para sólo una interfaz. Un paquete enviado a una dirección *Unicast* es entregado únicamente a la interface identificada con esa dirección IP. Una dirección *Multicast* es un identificador para un grupo de interfaces. Un paquete enviado a una dirección *Multicast* es entregado a todas las interfaces que tengan asignadas esta dirección de red [4]. Y por último, la dirección *Anycast* es un identificador para un grupo de interfaces y puede ser asignada a una o más interfaces, es decir, varios dispositivos pueden tener la misma dirección *Anycast*. Un paquete enviado a una dirección *Anycast* es entregado a la interfaz identificada con esa dirección IP más cercana, según lo establecido en el medición de las distancias en el protocolo de enrutamiento. La comunicación en las direcciones *Anycast* es de uno a muchos y solamente uno responde. Este tipo de direcciones no existen en IPv4. Durante el diseño y desarrollo del protocolo IPv6 se definió que las direcciones de tipo *broadcast* del protocolo IPv4 no serían implementadas en IPv6. Se llegó a esta conclusión debido a que un mensaje enviado al *broadcast* de la red le llega a todos los nodos de la misma y genera tráfico extra en la red. Para implementar algo similar sin generar tanto tráfico ni incomodar a todos los huéspedes de la red se implementó la dirección de tipo *Multicast* en el protocolo IPv6.

Existen varios tipos de direcciones *Unicast:* direcciones globales (*Global Unicast*), enlace local (*Link Local*), *Loopback, Unspecified, Unique Local, Embedded IPv4 y Site-Local*, esta última está obsoleta y no se recomienda su utilización. La dirección *Site-Local* muestra ambigüedad porque puede estar presente en múltiples sitios. Adicionalmente la dirección en sí no contiene ningún indicador del sitio al cual

pertenece, dificultando el uso de la misma a los desarrolladores de aplicaciones [5]. Lo anterior implica que una dirección *Site-Local* puede ser usada cuando un cliente o un servidor se encuentran en el mismo sitio, pero al intentar conectarlos en lugares geográficamente diferentes se presentan inconvenientes. Por este motivo, las aplicaciones necesitarían tener algún conocimiento sobre la topología de la red, esta ha sido una de las causas de su inutilización. La dirección *Global Unicast* es equivalente a una dirección pública IPv4. Es enrutable y alcanzable en la red de internet IPv6 y está diseñada para ser agregada o extraída para colaborar en una infraestructura de enrutamiento eficiente [6]. La estructura de la dirección *Global Unicast* se define en el RFC 3587 [7] y se muestra en la Imagen 1. Su segmento inicia en 2000::/3 y finaliza en 3FFF::/3.

n bits	**m bits**	**128-n-m bits**
global routing prefix	subnet ID	interface ID

Imagen 1. Estructura dirección Global Unicast

La dirección *Link Local* es usada por nodos para comunicarse con otros nodos de la misma red. Está diseñada con el propósito de auto-configurarse para los casos en los cuales la información de red no se encuentre disponible. No son enrutables o alcanzables desde la red externa IPv6. Adicionalmente, un enrutador no debe encaminar paquetes a otros enlaces cuando la dirección de destino u origen sea de tipo enlace local [4]. La dirección *Link Local* en IPv6 es similar a la dirección *Link Local* de IPv4 definida en el RFC 3927 y utiliza el segmento 169.254.0.0/16. En Ipv6 se tiene el rango desde FE80::/64 hasta FEBF::/64. La Imagen 2 enseña la estructura de la dirección *Link Local.*

10 bits	**54 bits**	**64 bits**
1111111010	0	interface ID

Imagen 2. Estructura dirección Link Local.

Otro tipo de dirección *Unicast* es la *Loopback* que se representa como 0:0:0:0:0:0:0:1 ó ::1 y puede ser utilizada por un nodo para enviarse paquetes IPv6 a sí mismo. No puede ser asignada a ninguna interfaz de red física. Además ningún paquete que se envíe a un nodo diferente del mismo que lo genera, puede contener como dirección IP de origen o destino una de tipo *Loopback* [4]. Esta dirección IP es equivalente a la dirección IPv4 127.0.0.1 que en realidad comprende el rango 127.0.0.0 hasta 127.255.255.255

También se tiene la dirección IP de tipo *Unspecified* e igual que la dirección *Loopback* no debe ser asignada a un nodo. Indica la ausencia de una dirección IP. Su representación es :: ó 0:0:0:0:0:0:0:0. Comúnmente se utiliza como dirección de origen (s*ource address*), cuando una única dirección no ha sido determinada, en el encabezado de un paquete IPv6 enviado por un nodo que desea establecer su propia dirección IP [8]. Esta dirección IP es similar en IPv4 a la dirección 0.0.0.0

Las direcciones *Unique Local* [2] se asemejan a las direcciones privadas de IPv4 [9] con una alta probabilidad de que sean únicas, no deben ser enrutables hacia la red de internet IPv6. Su estructura se define en el RFC 4193 [2] como se muestra en la Imagen 3. Su segmento inicia en FC00::/7 y termina en FDFF::/7

7 bits	**1 bit**	**40 bits**	**16 bits**	**64 bits**
Prefix	L	Global ID	Subnet ID	Interface ID

Imagen 3. Estructura dirección Unique Local

Y por último, las direcciones *Embedded IPv4* se utilizan para aquellas redes donde conviven tanto IPv6 como IPv4 y permite representar las direcciones de los nodos IPv4 como direcciones IPv6 [10]. La composición de la dirección IP *Embedded IPv4* se exhibe en la Imagen 4. Tres métodos comunes para la coexistencia entre IPv4 e IPv6 son: de doble pila por su nombre en inglés *dual-stack, tunneling* y traslación de dirección de red por su nombre en inglés *Network Address Translation IPv6 to IPv4* o también conocido como NAT64 [11]. A continuación se explicará brevemente cada uno de ellos. Un dispositivo *dual-stack* soporta tanto IPv4 como IPv6. Para transmitir un paquete se realiza mediante IPv4 o IPv6 y con el puerto TCP [12] o UDP [13] se determina la aplicación, dado el caso que la aplicación soporte IPv4 e IPv6. La selección no se realiza aleatoriamente, primero se averigua cuál de las dos opciones utiliza el receptor. Si son ambas, el sistema operativo puede definir, previa a una configuración, cuál utilizar por omisión. El esquema de *tunneling* es una solución temporal mientras se migra a IPv6 completamente. Básicamente consiste en encapsular un paquete IP en otro paquete. De esta manera se pueden encapsular paquetes IPv6 en paquetes IPv4, es muy útil para redes que solo soportan IPv4. Se pueden transmitir paquetes desde una red IPv6 hacia otra red IPv6 pasando por una red IPv4. Y por último, el NAT64, que se asimila con NAT en IPv4 que permite trasladar direcciones IPv4 de públicas a privadas y viceversa. Fundamentalmente permite comunicar redes de solo IPv6 con redes IPv4 haciendo traducciones y asignaciones de IP temporales.

10 bits	54 bits	64 bits
1111111011	0	interface ID

Imagen 4. Estructura dirección Embedded IPv4

La Tabla 4, tomada de [14], muestra como la IANA, por su nombre en inglés *Internet Assigned Numbers Authority,* ha definido los rangos para las direcciones IPv6.

IPv6 Prefijo	Inicia	Termina	Descripción
0000::/8	0	00FF	Unspecified, Loopback y Embedded IPv4
0100::/8	100	01FF	Reservado por la IETF - Internet Engineering Task Force
0200::/7	200	03FF	Reservado por la IETF
0400::/6	400	05FF	Reservado por la IETF
0800::/5	800	0FFF	Reservado por la IETF
1000::/4	1000	1FFF	Reservado por la IETF
2000::/3	2000	3FFF	Global Unicast
4000::/3	4000	5FFF	Reservado por la IETF
6000::/3	6000	7FFF	Reservado por la IETF
8000::/3	8000	9FFF	Reservado por la IETF
a000::/3	A000	BFFF	Reservado por la IETF
c000::/3	C000	DFFF	Reservado por la IETF
e000::/4	E000	EFFF	Reservado por la IETF
f000::/5	F000	F7FF	Reservado por la IETF
f800::/6	F800	FBFF	Reservado por la IETF
fc00::/7	FC00	FDFF	Unique Local Unicast
fe00::/9	FE00	FE74	Reservado por la IETF
fe80::/10	FE80	FEBF	Link-Scoped Unicast
fec0::/10	FEC0	FEFF	Reservado por la IETF
ff00::/8	FF00	FFFF	Multicast

Tabla 4. Rango de direcciones IPv6 establecido por la IANA. Tomado de [14]*.*

3.3 Mecanismos para la asignación de direcciones IP.

El protocolo IPv6 provee tres mecanismos para la configuración de direcciones IP. Uno de ellos es la asignación manual, también a través del protocolo DHCPv6 [15] y el último es por medio de la “Configuración automática de direcciones sin estado”

[16] por su nombre en inglés *"Stateless Address Autoconfiguration"* con sus siglas SLAAC. Unas de las grandes ventajas de este último son: no requiere de servidores adicionales y permite generar la dirección IP de cualquier nodo, que no sea un enrutador, sin intervención manual.

3.4 Asignación de dirección IP mediante DHCP.

Para obtener una dirección IPv4 en el protocolo DHCPv4 [17], un cliente envía un mensaje *Broadcast* preguntando si existe alguna servidor DHCPv4. El servidor DHCPv4 le suministra los datos al huésped y así se logra conectar a la red. La gran diferencia entre DHCPv4 con respecto a DHCPv6 es que en la versión 6 no existe dirección *Broadcast.* En vez de *Broadcast* se utiliza la dirección *Multicast*, para que los paquetes solo sean recibidos por dispositivos que tengan servicios que dependen de dirección *Multicast* como DHCP o NTP [18], y así no enviar mensajes a toda la red generando tráfico de más. Para obtener una dirección IPv6 con el procedimiento DHCPv6, el cliente primero consulta con el enrutador si se conecta a través de SLAAC, este método se detallará más adelante, o vía DHCP. Por DHCPv6, el cliente envía un mensaje *Multicast* preguntando si existe algún servidor DHCPv6. El servidor DHCPv6 le contesta afirmativamente. El cliente le solicita los parámetros de configuración, incluida la dirección IP. El servidor le envía los valores correspondientes. Y por último se valida que la dirección IP no se encuentre duplicada, esta práctica se explicará posteriormente.

3.5 Configuración automática de direcciones sin estado (SLAAC), utilizando el protocolo Neighbor Discovery para IP versión 6.

En primera instancia, cuando una interfaz inicia, SLAAC genera la dirección IP *Link Local*. Usando como prefijo un rango desde FE80::/10 hasta FEBF::/10 y se procede con el complemento de la dirección IP. Para añadir los bits posteriores al prefijo en la IP, se pueden utilizar números aleatorios o a través de EUI-64 [19], este último será explicado detalladamente más adelante. Seguidamente a la generación, se inicia el proceso para validar la unicidad de la dirección IP. Mediante el Protocolo de Descubrimiento de Vecinos, por su nombre en inglés *Neighbor Discovery for IP versión 6* [20], se envía un mensaje *Multicast* preguntándole a todos los nodos del enlace si alguien tiene asignada la dirección IP que recién se creó. En caso que alguna interfaz la tenga asignada, solamente esa interfaz deberá contestar con la alerta. Por el contrario, sino se no recibe mensaje, se asignará la dirección *Link Local.* Este proceso se conoce como Detección de Direcciones Duplicadas, DAD por su nombre en inglés *Duplicate Address Detection* [16] y se entrará en detalle posteriormente. Durante la validación DAD se pueden generar algunos problemas de seguridad, estos serán estudiados en este documento.

Luego de generada, validada la unicidad y asignada la dirección IP *Link Local,* se continua con la búsqueda de la dirección *Global Unicast.* Nuevamente utilizando el protocolo *Neighbor Discovery,* se envía un mensaje *Multicast* solamente a los

enrutadores de la red solicitando el prefijo y su tamaño para las direcciones *Global Unicast.* Con esta información el huésped termina de completar la dirección IP, igual que en la *Link Local*, aleatoriamente o con EUI-64. Y por último, una vez más se ejecuta el proceso DAD.

El método SLAAC, como mecanismo dinámico para configurar una dirección IP, se basa en el *Neighbor Discovery for IP version 6* [20]. Este protocolo utiliza los cinco mensajes del protocolo ICMPv6 [21] que se muestran en la Tabla 5.

Nombre en Español	**Nombre en inglés**	**Siglas**	**Tipo mensaje ICMPv6**
Mensaje de solicitud de enrutador	Router Solicitation message	RS	133
Mensaje de anuncio de enrutador	Router Advertisement message	RA	134
Mensaje de solicitud de vecino	Neighbor Solicitation message	NS	135
Mensaje de anuncio de vecino	Neighbor Advertisement message	NA	136
Mensaje para redirigir	Redirect message		137

Tabla 5. Tipos de mensajes del protocolo Neighbor Discovery

Como se mencionó anteriormente, para generar la dirección IP *Link Local* con el método SLAAC se utiliza un prefijo con tamaño 10. Tiene un rango desde FE80::/10 hasta FEBF::/10 que en binario corresponden a 1111111010000000 y 1111111010111111 respectivamente. Los siguientes 54 bits pueden contener cualquier valor y los últimos 64 bits pueden generarse aleatoriamente, manualmente o mediante el método EUI-64 [19]. EUI-64 consiste en tomar los 48 bits que abarcan la dirección MAC de la interfaz. Los primeros 24 bits corresponden al identificador único de la organización por su nombre en inglés *Organizational Unique Identifier* con sus siglas OUI. Los últimos 24 bits pertenecen al identificador del dispositivo. Estos valores se dividen y entre ellos se agregan los siguientes 16 bits 11111111-11111110 que equivalen a los caracteres FF-FE en hexadecimal. Adicionalmente, el séptimo bit permite determinar si la IP se administra de forma universal o local. En 1 indica que la administración corresponde a la IEEE. En 0 hace mención a una administración local. De esta manera se completan los 64 bits restantes para la dirección IP. La descripción grafica se observa en la Imagen 5.

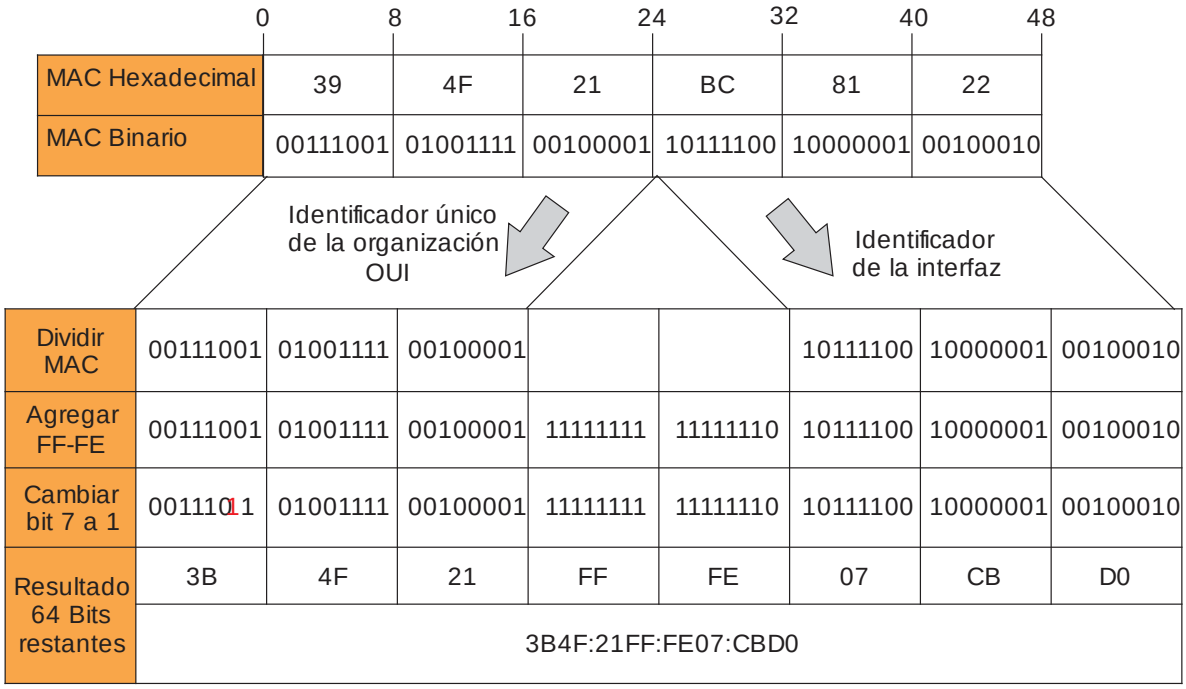


Imagen 5. Generación 64 bits a partir de la MAC con EUI-64.

3.6 Detección de Direcciones Duplicadas, DAD.

Luego de generada la dirección IP *Link Local*, se procede con la Detección de Direcciones Duplicadas. Este método funciona para cualquier dirección *Unicast, Link Local* o *Global-unicast*, independiente de si se genera mediante SLAAC, DHCPv6 o manualmente. Mientras se ejecuta el proceso DAD la dirección IP generada lleva el nombre de dirección IP tentativa que no es considerada como una dirección IP asignada a una interfaz. El proceso DAD hace uso del protocolo ND con los mensajes *Neighbor Solicitation* y *Neighbor Advertisement*. De este modo se permite identificar si una IP generada, con los métodos anteriormente explicados, se encuentra siendo utilizada por otro nodo. En caso de encontrar alguna dirección IP como duplicada se debe cambiar el identificador de la interfaz o configurar la IP manualmente antes de ser asignada a la interfaz. Durante la ejecución del DAD la interfaz debe aceptar tráfico de mensajes NS y NA, que en el campo *target address* contengan una IP tentativa. Antes de proceder con el envío de estos mensajes el nodo debe estar conectado a la red *Multicast*. A continuación se detalla el proceso.

Teniendo la dirección IP tentativa, el nodo procede a enviar un mensaje *Neighbor Solicitation* ICMPv6 de tipo 135 a todos los nodos de la red con el objetivo de preguntar si la IP tentativa se encuentra siendo utilizada por alguna interfaz. Este mensaje contiene en el campo dirección de origen una IP de tipo *Unspecified*, en el campo dirección de destino la dirección IP *Multicast* y por último el campo *target address* la dirección IP tentativa. El receptor logra reconocer que el transmisor se encuentra ejecutando un DAD de acuerdo a la identificación de los tipos de IP que

contienen los campos en el mensaje. Si la dirección IP del receptor es diferente a la dirección IP tentativa de emisor significa que la dirección IP no se encuentra duplicada y el receptor no deberá responder al mensaje. En cambio, si la dirección IP tentativa del emisor es igual a la dirección IP del receptor denota que se encuentra duplicada. Como consecuencia, el receptor deberá contestar con un mensaje *Neighbor Advertisement* ICMPv6 de tipo 136 agregando en el campo *target address* la dirección IP tentativa, en el campo dirección de origen la IP del nodo que responde al mensaje NS y, por último, con la IP *Multicast* en el campo dirección de destino. De esta manera el emisor del mensaje NS recibirá el mensaje NA e identificará en el campo *target address* la dirección IP tentativa. De tal modo podrá deducir que la IP que generó se encuentra duplicada, viéndose obligado a crear una nueva dirección IP para poder conectarse a la red. El emisor tomará un tiempo prudente para esperar por una respuesta a su mensaje NA. Luego de pasado este tiempo se procederá con la asignación de la dirección IP generada en primera instancia.

3.7 Asignación de dirección IP Global Unicast.

Después de la asignación de la dirección IP *Link Local* se continúa con la generación de la IP *Global Unicast.* Nuevamente se emplea el método SLAAC con el protocolo ND pero ahora aprovechando los mensajes RS y RA. Los enrutadores de la red periódicamente envían mensajes *Router Advertisement* ICMPv6 de tipo 134*,* permitiéndole a una interfaz conocer el método de generación de la IP, tales como SLAAC o DHCPv6. Los mensajes RA también contienen el prefijo de la dirección de red que debe utilizar el nodo, al tamaño del prefijo, la cantidad máxima de saltos (*Hop Limit*) del encabezado IPv6 y opcionalmente el tamaño de la unidad máxima de transmisión por su nombre en inglés *Maximum Transmission Unit* con su sigla MTU. Una interfaz espera por el mensaje RA por parte del enrutador. Si en determinado lapso de tiempo no se ha recibido el mensaje el nodo procederá a enviar un mensaje *Router Solicitation* ICMPv6 de tipo 133 de modo *Multicast* a todos los enrutadores de la red. El enrutador le responderá con un mensaje RA. Con esta información el nodo iniciará su proceso de generación de la IP definitiva. Partiendo del prefijo de red recibido, compuesto del prefijo de enrutamiento global y sumándole la identificación de la subred como se muestra en la Imagen 1, y completando los bits restantes con la identificación de la interfaz aleatoriamente o utilizando el método EUI-64. Finalmente se revisa si la dirección IP generada se encuentra duplicada con el mecanismo de detección de direcciones duplicadas.

Otra utilidad del protocolo Neighbor Discovery es el almacenamiento de la dirección MAC asociada a una IP en una tabla. Esto ocurre cada que se recibe un mensaje RS, RA, NS o NA teniendo en cuenta que no se debe incluir cuando la dirección de origen es de tipo *Unspecified*. Adicionalmente, en caso de necesitar una dirección MAC que no se encuentre en la tabla, el nodo puede averiguar la MAC a través del protocolo ND. Este proceso tiene similitud con el protocolo ARP [22] en IPv4. Cuando un nodo requiere conocer la dirección MAC de un IP a la cual le transmitirá un mensaje, primero revisa su tabla interna, llamada “Cache de vecinos” por su nombre en inglés *Neighbor Cache*. Allí se almacenan los pares IP-MAC. Dado el

caso de no encontrar una MAC asociada a la IP de destino se utilizan los menajes *Neighbor Solicitation* y *Neighbor Advertisement*. Se envía un mensaje NS a una IP *Multicast* preguntando la MAC de determinada dirección IP. Solamente la interfaz con la IP en cuestión asignada deberá responder con un mensaje NA para que el emisor actualice su tabla *Neighbor Cache.*

3.8 Posibles vulnerabilidades de seguridad en el protocolo Neighbor Discovery para IPv6

Durante la asignación de direcciones IP de tipo *Link Local* y *Global Unicast,* en el protocolo IPv6, se debe maximizar la probabilidad de que la dirección IP sea única en la red. Debido a que el protocolo ND carece de autenticidad e integridad en los mensajes, durante el procedimiento de detección de direcciones duplicadas se evidencias grandes falencias en seguridad. Estas debilidades pueden ser aprovechadas por un atacante para generar una denegación de servicio con sus siglas DOS por su nombre en inglés *Denial of Service* [23], a un nodo que intenta asignar su dirección IP. Una de estas debilidades aparece cuando el nodo pregunta si la dirección IP se encuentra duplicada. Un atacante puede contestar todos los mensajes NS con un mensaje NA falsificado, indicando en el campo *target address* una dirección IP tentativa. De tal manera que el emisor crea que la IP se encuentra en uso, viéndose obligado a generar una nueva dirección IP. Igualmente el atacante puede continuar en un ciclo infinito generando un ataque de denegación de servicio. A consecuencia, el nodo no podrá conectarse a la red.

Por otra parte, el protocolo ND no limita la cantidad de mensajes NA que pueden ser enviados. Un atacante puede explotar esta falencia y enviar mensajes ilimitadamente. Con el objetivo de inundar la tabla *Neighbor Cache* hasta lograr desbordarla. Lo anterior provoca que se consuma la memoria del núcleo del sistema generando otro ataque de DOS.

Teniendo en cuenta que los mensajes RA contienen información importante, un atacante puede modificar estos mensajes con datos falsificados. Así se podría cambiar el MTU, tamaño del prefijo para una dirección *Global Unicast* o la cantidad máxima de saltos. Con estas modificaciones se puede alterar notablemente la autoconfiguración de una IP, a tales puntos como: colocar un MTU pequeño para generar fragmentación y por ende mayor tráfico en la red, reducir la cantidad de saltos de un paquete para que nunca lleguen a su destino o incluso brindar un prefijo falso para causar conflicto en el enrutamiento de paquetes.

Estos ataques se especifican y detallan a profundidad en el numeral 6 - Técnicas de ataque con gran impacto en el protocolo Neighbor Discovery.

4 PERTINENCIA Y ANTECEDENTES.

En [24] Bansal *et al* indican que el protocolo ND es un protocolo que carece de autenticidad en los mensajes lo que pueden conllevar a enfrentar ataques como: falsificación de mensajes de NS, NA, falsificación de DAD e inundación de la tabla *Neighbor Cache*. Con la explotación del este protocolo pueden también generarse ataques de denegación de servicio. Allí mismo los autores proponen como solución un esquema con un sistema de detección de intrusos, por su nombre en inglés *Intrusion System Detection* con sus siglas IDS, para prevenir falsificación de paquetes NA. El planteamiento tiene como requerimiento que las direcciones IP se configuren a través de SLAAC. Los dispositivos deben responder a los mensajes NA en un intervalo de tiempo establecido. Adicionalmente la máquina IDS debe ser confiable con dos interfaces de red, una para recolectar información de la red y la otra exclusivamente para manejar los mensajes NS/NA. El método consiste en crear tablas que permitan almacenar los datos de los mensajes NS/NA. Almacenar parte de la información de cada mensaje como: IP origen, MAC, si es DAD, entre otros. Con estos datos se verifica si la información corresponde al huésped que los envía. También permite detectar mensajes NA que viajan por la red que no corresponden a una respuesta de un mensaje NS y no un mensaje generado sin antes haber recibido un NS. De tal modo de disminuir la probabilidad de un ataque de DOS.

En [25] Arkko *et al* escriben que los nodos de un enlace local no son confiables. En la práctica, cualquier nodo puede ejecutar un ataque de DOS falsificando los mensajes del protocolo ND. Por ejemplo, sobre las redes que utilizan y posteriormente ejecutan el método de DAD se ven expuestas a un ataque de DOS, debido a que un atacante puede responder a las solicitudes NS de DAD con respuesta de IP duplicada. También con al redirección de mensajes, un atacante puede capturar mensajes con objetivos tales como: redirigir paquetes hacia un nodo inexistente para impedir la correcta comunicación o tomar gran cantidad de paquetes y enviarla a un cliente para inundar las bitácoras, el procesador o la interfaz de red. Para mitigar estos problemas, se proponer crear las direcciones IP utilizando criptografía asimétrica. Empleando una infraestructura de criptografía basa en identidad para el manejo de las claves públicas y privadas. De este modo se podrán autenticar los mensajes y disminuir la probabilidad de éxito en un ataque.

En [26] Gont *et al* manifiestan que un atacante puede escuchar tráfico del esquema DAD y responder con una NA falsificado indicando que la dirección IP generada ya se encuentra en uso. De esta manera le impide a su víctima realizar la configuración dinámica de la dirección IP generando un ataque de denegación de servicio. En el mismo proceso DAD, un atacante puede responder a los mensajes NS con otro mensaje NS con la misma dirección IP. El nodo atacado creerá que hay una colisión y cambiara la dirección IP. También se menciona la falsificación de parámetros en los mensajes RA. Cambiando el MTU, la cantidad de saltos de cada paquete, prefijo de red o incluso ordenando la configuración a través de DHCP. En caso de buscar un servidor DHCP y no encontrar un servicio se provocará un fallo. Con lo anterior se pueden producir ataques de denegación de servicio o pérdida en el desempeño. Para aminorar estos problemas, en el mismo documento, se estimula la utilización

de herramientas de monitoreo como NDPMon tal cual se menciona en [27] o con la utilización de la herramienta *SI6 Networks' IPv6 Toolkit.*

En [28] Nikander *et al* especifican un ataque de denegación de servicios utilizando el mecanismo DAD, en el cual un atacante responde a todos los mensajes NS que pertenezcan al DAD con IP duplicada evitando que un host logre obtener dirección IP. También se menciona la modificación de parámetros para causar ataques de DOS. Como procedimiento de defensa, recomiendan la configuración manual del protocolo IPsec [29] con el objetivo de proteger el protocolo *Neighbor Discovery* para IPv6. Para mayor información sobre la implementación se puede consultar el documento [30].

En [31] Barbhuiya *et al* mencionan que para un par comunicarse con otro nodo necesita conocer su MAC. Para ello utiliza mensajes NS y NA preguntando la MAC correspondiente a la IP que desea enviarle. La respuesta con la MAC llega en el mensaje NA. La ventaja es que todos los nodos de la red obtienen este mensaje y pueden actualizar su tabla *Neighbor Cache* sin necesidad de enviar un mensaje NS nuevamente, aumentando el desempeño de la red. Esto puede conllevar a un problema de seguridad debido a que la tabla es actualizada sin ninguna verificación que compruebe datos no falsificados. A consecuencia, se mencionan como ataques comunes la falsificación de mensajes *Neighbor Advertisement* y *Neighbor Solicitation.* Este tipo de ataques pueden derivar en una denegación de servicio o, incluso, de un hombre en el medio, por su nombre en inglés *Man in the middle* con sus siglas *MitM* [32]*.* El esquema propuesto de solución es la implementación de un IDS. Este IDS debe filtrar los mensajes NS y NA e identificar paquetes falsificados. Basado en los 5 algoritmos que allí se expresan (manejador de paquetes NS, manejador de paquetes NA, verificación IP-MAC, analizador de respuesta y manejador de anuncios no solicitados), se busca mermar la probabilidad de éxito de un ataque de falsificación de mensajes NS y NA.

En [33] Arkko *et al* muestran diferentes mecanismos de seguridad para el protocolo *Neighbor Discovery* sin la utilización de IPsec. Para ello propone generar las direcciones IP criptográficamente, este procedimiento se conoce en inglés como *Cryptographically Generated Addresses* con sus siglas en inglés *CGA* [34]*.* Por consiguiente, se utilizan claves públicas y privadas, funciones *hash* de una sola vía y parámetros auxiliares. También la inclusión de firmas RSA [35] en los paquetes pueden ayudar a proteger los mensajes. Y como punto interesante es la no utilización de una autoridad de certificación.

5 PROTOCOLO NEIGHBOR DISCOVERY PARA IPV6 A TRAVÉS DE ICMPV6.

El protocolo *Neighbor Discovery* para IPv6 es una herramienta de gran valor agregado para el protocolo IPv6. A través de este protocolo es posible auto-configurar las direcciones IP *unicast*: *link-local* y *global-unicast* en los huéspedes. Adicionalmente reemplaza al protocolo ARP IPv4 permitiéndole a los nodos conocer determinada dirección de MAC. También le facilita a los dispositivos de la red determinar si un nodo es alcanzable o debe enviar la información a través del enrutador de red. El protocolo Neighbor Discovery utiliza como base el protocolo ICMPv6 [21] para transportar los datagramas mediante la capa de internet del modelo jerárquico TCP/IP.

El protocolo ICMPv6 contiene la cabecera mostrada en la Imagen 6.

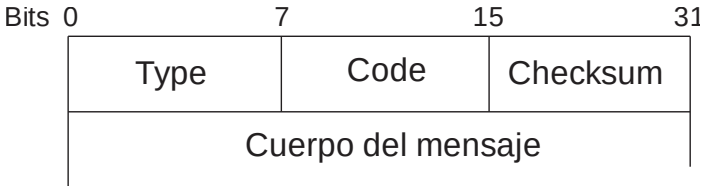


Imagen 6. Cabecera de un datagrama ICMPv6.

El campo ***Type*** permite determinar los tipos de mensajes, el protocolo ICMP maneja dos clases de mensajes: de errores y de información. El campo tiene una longitud máxima de 8 bits ($2^8 = 256$) a lo que equivale un número entre 0 y 255 para un total de 256 tipos de mensajes. El protocolo define que la primera mitad (0 a 127) corresponden a mensajes de error mientras que los restantes (128 a 255) a mensajes de clase informativa.

Tipo	Mensajes Error
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem
100	Reservados para experimentos
101	Reservados para experimentos
127	Reservados para experimentos

Tabla 6. Tipos de mensajes de error utilizados en el protocolo ICMPv6. Tomado de [21].

Tipo	Mensajes informativos
128	Echo Request
129	Echo Reply
200	Reservados para experimentos
201	Reservados para experimentos
255	Reservados para experimentos

Tabla 7. Tipos de mensajes informativos utilizados en el protocolo ICMPv6. Tomado de [21]*.*

En las tablas 6 y 7 no se muestran los mensajes de tipo 133, 134, 135, 136 y 137 que son utilizados por el protocolo *Neighbor Discovery,* estos mensajes se reflejan en la Tabla 8, los cuales serán estudiados más adelante. El campo ***Code*** es utilizado para determinar un subnivel en el tipo de mensajes y dependen del campo ***Type***. Y por último el campo ***Checksum*** que permite identificar si la información se encuentra corrupta.

Cabe apreciar que en este documento, cada vez que se mencione la palabra ICMP se hace referencia a ICMPv6 a excepción que se indique lo contrario.

5.1 Mensajes del protocolo Neighbor Discovery para IPv6.

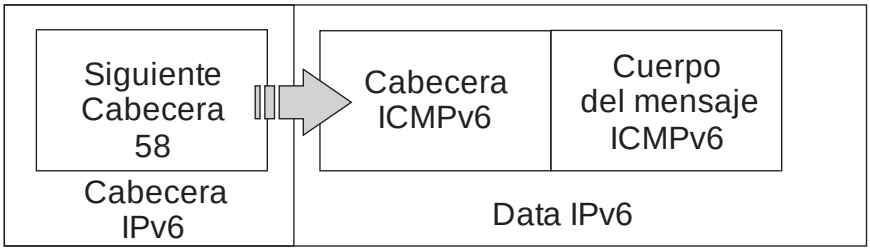


Imagen 7. Cabecera IPv6 indicando que la información siguiente corresponde a una cabecera ICMPv6.

Como se mencionó con anterioridad, el protocolo *Neighbor Discovery* utiliza 5 tipos de mensajes (ver Tabla 8) y se explicarán el formato de cada uno de los mensajes.

Nombre en Español	Nombre en inglés	Siglas	Tipo mensaje ICMPv6
Mensaje de solicitud de enrutador	Router Solicitation message	RS	133
Mensaje de anuncio de enrutador	Router Advertisement message	RA	134
Mensaje de solicitud de vecino	Neighbor Solicitation message	NS	135
Mensaje de anuncio de vecino	Neighbor Advertisement message	NA	136
Mensaje para redirigir	Redirect message		137

Tabla 8. Tipos de mensajes del protocolo Neighbor Discovery.

5.2 Formato Router Solicitation Message.

El mensaje de solicitud de enrutador tiene la información mostrada en la Imagen 8.

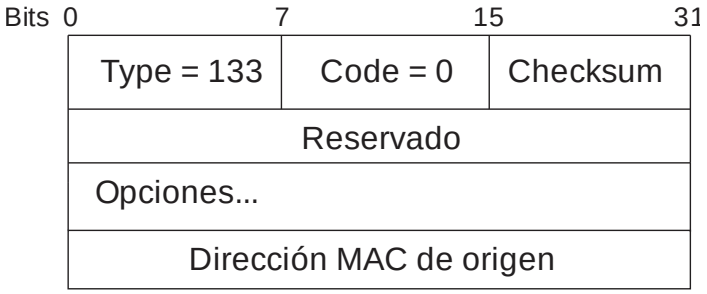


Imagen 8. Cabecera del mensaje Router Solicitation en el protocolo Neighbor Discovery para IPv6.

La cabecera ICMP muestra el campo ***Type*** con valor de 133 para hacer referencia a un mensaje *Router Solicitation*. El campo ***Code*** está en cero por diseño del protocolo y se le agrega el ***Checksum***. Posteriormente se reservan 32 bits. Por último está la opción para agregar la **Dirección MAC de origen**.

Luego de asignada la dirección *link-local* a través del método SLAAC se procede a enviar el mensaje *Router Solicitation* con IP destino *multicast*. En la Imagen 9 se observa un ejemplo de la cabecera del protocolo IP para indicar que contiene una cabecera ICMP. Se observa el campo **versión** (4 bits) en estado 6 indicando que es protocolo IPv6 y no IPv4 (En el protocolo IPv4 se denota con el número 4). Los siguientes dos campos **clase de tráfico** (8 bits) y **secuencia** (20 bits) se encuentran

en ceros. El campo **tamaño *payload*** muestra el tamaño del datagrama, para el ejemplo se utilizó el valor 72. El campo **siguiente cabecera** indica que protocolo corresponde a la información siguiente de la cabecera IP, para el caso de ICMP se utiliza el valor 58 definido en [21]. Para los mensajes *Router Solicitation* el campo **límite de saltos** se define en 255. La dirección **IP fuente** es la generada a través de SLAAC, aleatoriamente, DHCPv6 o manualmente y la **IP de destino** es *multicast* (FF00::/8 - desde FF00:: hasta FFFF).

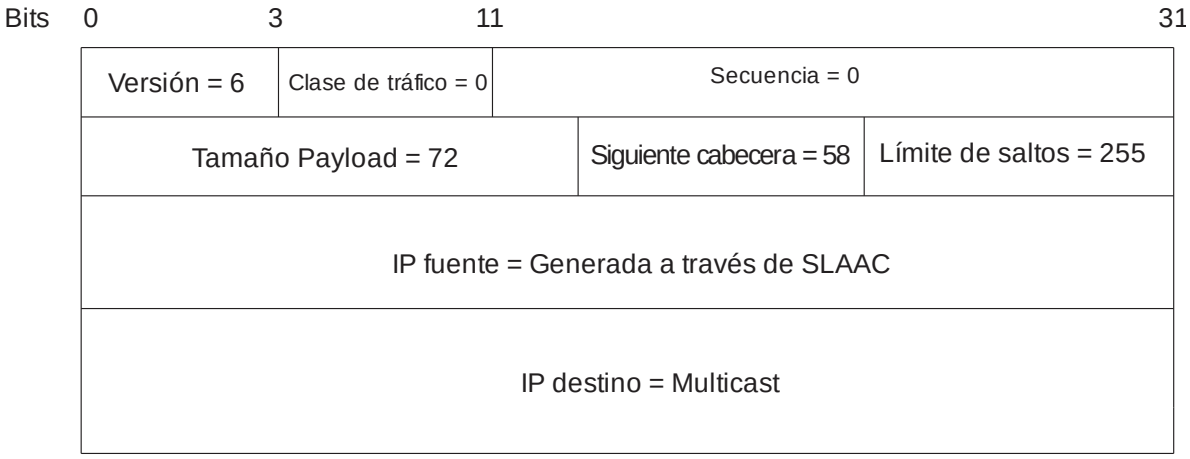


Imagen 9. Cabecera del protocolo IP anterior a la cabecera ND RS. Los números mostrados son base decimal.

Una vez generada la información se procede a enviar el mensaje *multicast* que solamente deberá responder el enrutador con un mensaje *Router Advertisement*.

5.3 Formato Router Advertisement Message.

El mensaje de anuncio de enrutador tiene la información mostrada en la Imagen 10.

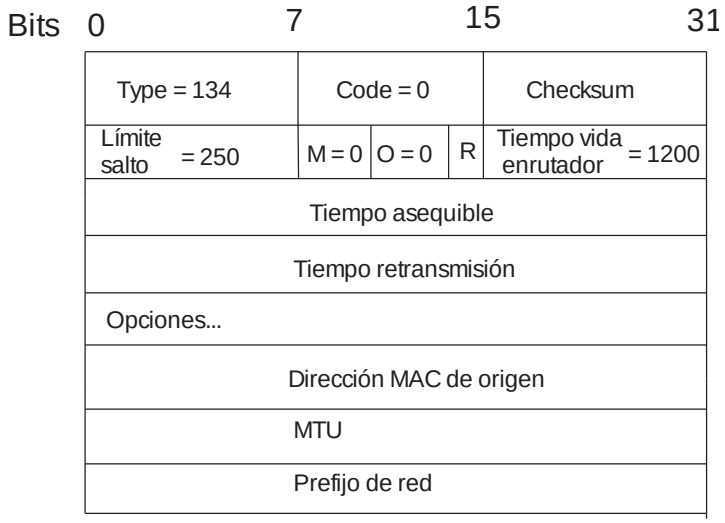


Imagen 10. Cabecera del mensaje Router Advertisement en el protocolo Neighbor Discovery para IPv6.

La Imagen 10 exhibe el mensaje *Router Advertisement* que contiene en el campo ***Type*** el valor 134 indicando que es una cabecera ICMP de *Router Advertisement*. El ***code*** en 0 y el ***checksum*** correspondiente. El protocolo define la **cantidad de saltos** en 255. El campo **M** (1 bit), cuando se activa permite identificar si la configuración de la dirección de red se hacen a través de DHCPv6. Cuando el bit está en cero indica que la configuración de la dirección IP se hace mediante SLAAC. El campo **O** (1 bit), cuando se activa indica que hay más información disponible en el servicio DHCPv6. En caso que ambos campos lleguen en 0 significa que no existe ningún servicio en la red que proporcione la información del direccionamiento y deberá realizarse SLAAC. El campo **R** es un valor reservado. El valor de **tiempo de vida del enrutador** indica por cuanto tiempo el enrutador estará disponible como primario, el valor mostrado es en segundos. Para este caso se utilizó como ejemplo 1200 pero puede variar según la red. El valor **tiempo asequible** muestra un tiempo en milisegundos para determinar el tiempo que se encuentra disponible el huésped que envió la información, es utilizado por el "algoritmo de detección de vecinos no alcanzables" por su nombre en inglés *Neighbor Unreachability Detection algorithm* [20]. El campo **tiempo de transmisión** muestra el valor en milisegundos entre los mensajes *Neighbor Solicitation* retransmitidos. Por último están las **opciones** que muestran información importante para los nodos de la red, se optimiza la red permitiendo la actualización de información sin haberla requerido y a su vez evita congestión de en la red. En las opciones está el campo **dirección MAC de origen** que permite a todos los nodos de la red llenar la tabla "Cache de vecinos" por su nombre en inglés *Neighbor Cache.* Con este método se reemplaza el protocolo ARP en IPv4. También se encuentra el campo **MTU** que permite determinar la unidad de máxima de transmisión. Y el **prefijo de red** le muestra al nodo como debe iniciar la dirección *global-unicast* si desea configurarla, a la vez viaja el tamaño del prefijo de la red.

El encabezado IP se muestra en la Imagen 11.

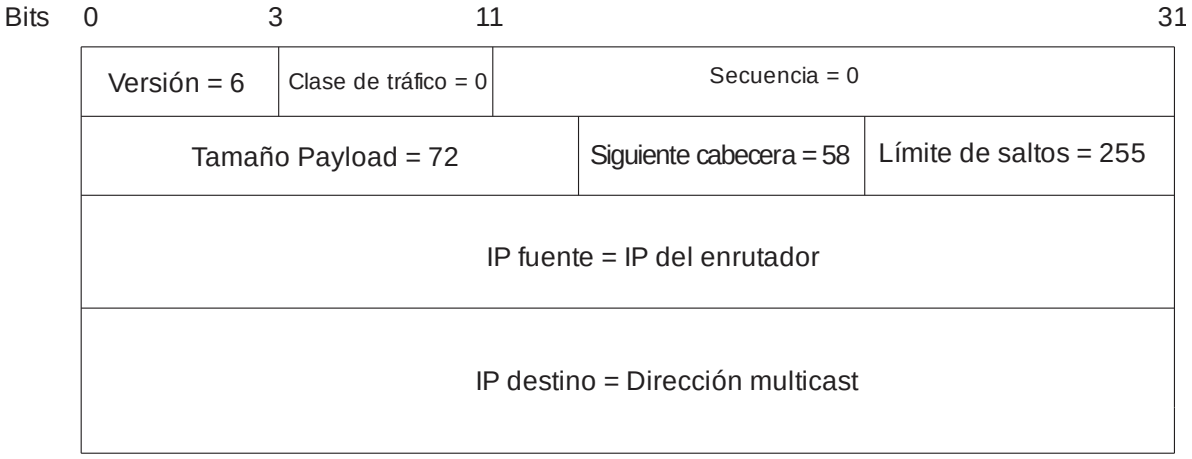


Imagen 11. Encabezado IP para un mensaje ICMP tipo 134.

Los datos son muy similares a los generados en el mensaje *Router Solicitation*. El cambio radica en el campo **IP fuente** donde se asigna la IP del enrutador que envía el mensaje. Y para tomar ventaja del campo **opciones** de la cabecera ICMP, se asigna la **IP destino** una dirección *multicast*.

5.4 Formato Neighbor Solicitation Message.

El mensaje de solicitud de vecino tiene la información mostrada en la Imagen 12.

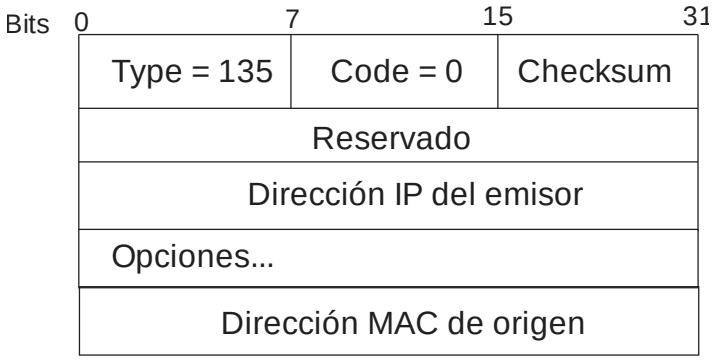


Imagen 12. Cabecera del mensaje Neighbor Solicitation en el protocolo Neighbor Discovery para IPv6.

La cabecera muestra el campo ***type*** con valor de 135 para hacer referencia a un mensaje NS. El campo ***code*** está en cero por diseño del protocolo y se le agrega el ***checksum***. Posteriormente se reservan 32 bits. Continúa el campo **dirección IP del emisor** y no puede ser una dirección *multicast*. Para finalizar esta la opción de agregar la **dirección MAC de origen**.

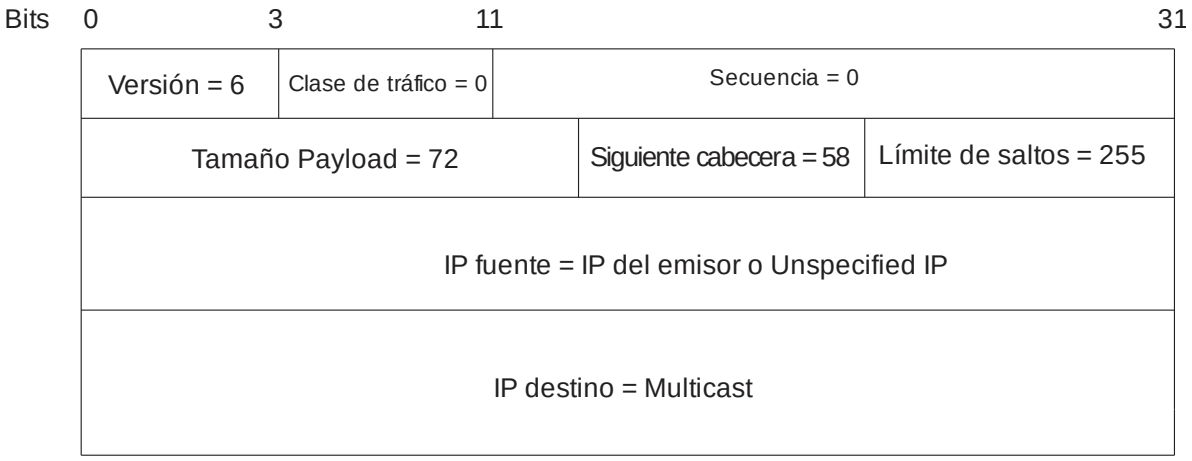


Imagen 13. Cabecera del protocolo IP anterior a la cabecera ND NS. Los números mostrados son base decimal.

La cabecera IP de los mensajes *Neighbor Solicitation* es similar a la cabecera IP de los mensajes *Router Advertisement* (ver Imagen 13). El cambio significativo radica en el campo IP fuente, si la dirección IP del nodo emisor ya fue asignada se colocara esta, si el mensaje es para ejecutar DAD se coloca un a IP *unspecified*.

5.5 Formato Neighbor Advertisement Message.

El mensaje de anuncio de vecino tiene la información mostrada en la Imagen 14.

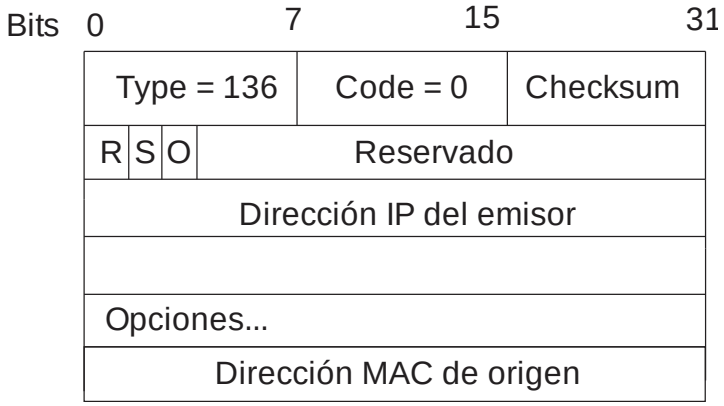


Imagen 14. Cabecera del mensaje Neighbor Advertisement en el protocolo Neighbor Discovery para IPv6.

La cabecera ICMP tipo 136 es similar a la cabecera del mensaje *Neighbor Solicitation*. La diferencia se nota en 3 campos. El campo **R** (1bit) cuando se activa indica que el emisor es un enrutador. El campo **S** (1bit) cuando se activa indica que

el mensaje es una respuesta a un mensaje *Neighbor Solicitation*. Y por último el campo **O** (1bit) cuando se activa indica que se debe reescribir el registro de la dirección MAC en la tabla cache de vecinos.

La Imagen 15 muestra la cabecera del protocolo IP la cual es similar al mensaje *Neighbor Solicitation*. La variación se encuentra en dos campos. El campo **IP fuente** contienen la IP del emisor. El campo **IP destino** puede variar, si el datagrama corresponde a una respuesta de un mensaje *Neighbor Solicitation* se coloca la IP de quién solicitó el mensaje en primera instancia, de lo contrario se coloca una IP *multicast*.

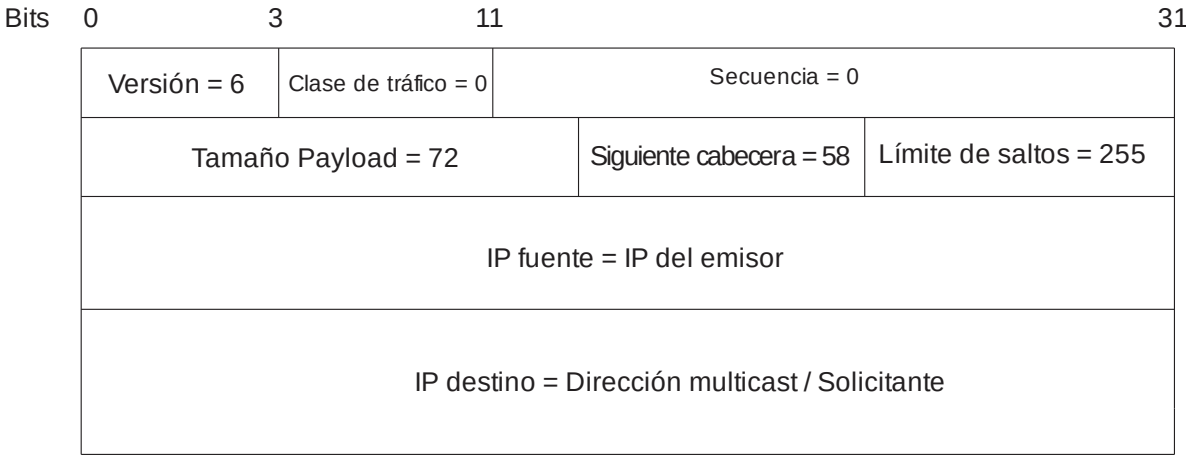


Imagen 15. Cabecera del protocolo IP anterior a la cabecera ND NA. Los números mostrados son base decimal.

5.6 Formato Redirect Message.

El mensaje para redirigir tiene la información mostrada en la Imagen 16.

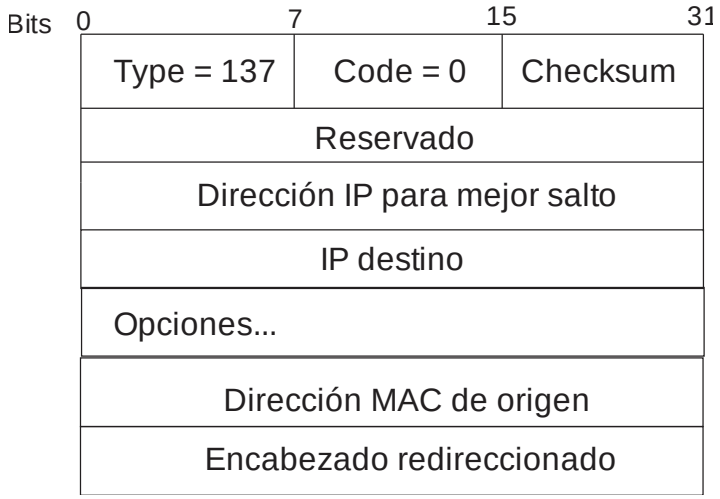


Imagen 16. Cabecera del mensaje redirect en el protocolo Neighbor Discovery para IPv6.

La *Imagen 16* muestra que el datagrama es de tipo 137 haciendo alusión a un mensaje *redirect.* En el campo **Dirección IP para mejor salto** se indica la IP con el mejor camino hacia el destino descrito en la cabecera IP como se muestra en la *Imagen 17*. La **IP de destino** es la IP del huésped que desea ser alcanzado. Y en las opciones está la dirección MAC de origen y el encabezado original del paquete que se redireccionó sin exceder el MTU de la red.

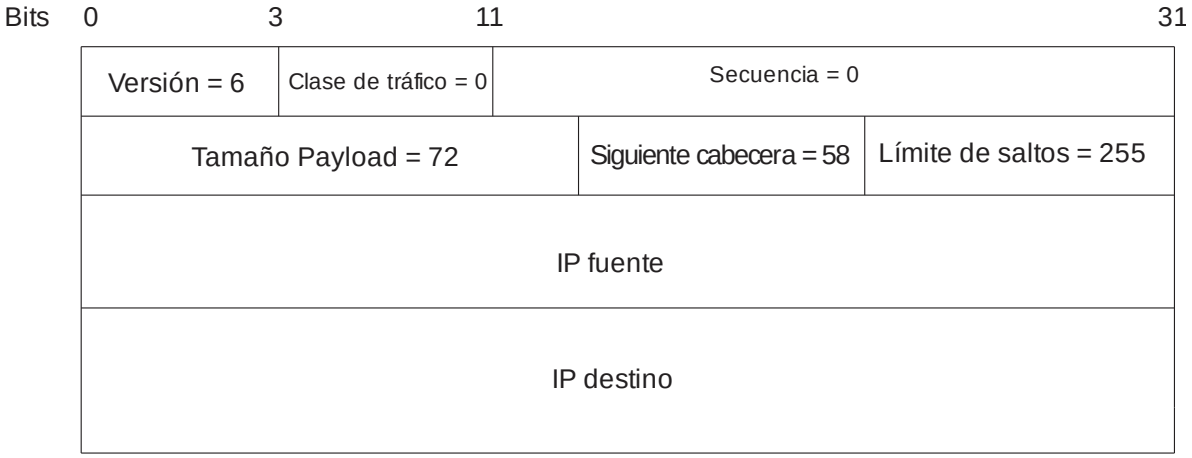


Imagen 17. Cabecera del protocolo IP anterior a la cabecera Redirect del protocolo ND. Los números mostrados son base decimal.

6 TÉCNICAS DE ATAQUE CON GRAN IMPACTO EN EL PROTOCOLO NEIGHBOR DISCOVERY.

6.1 Falsificación mensajes Router Advertisement.

Un huésped envía un mensaje *Router Solicitation* a los enrutadores de la red, con el objetivo de conocer el prefijo de red y su longitud para las direcciones *global-unicast*. Generalmente se usa una IP de destino *multicast* de sólo enrutadores, comúnmente es FF02::2 [25]. El enrutador contesta con un mensaje *Router Advertisement* y en las opciones del encabezado del mensaje tipo 133 le indica al emisor como debe configurar su prefijo de red. La Imagen 18 muestra el formato para la opción de **prefijo de red.** Un atacante puede falsificar estos mensajes y cambiar la información que en las opciones viajan. El nodo utilizará datos erróneos para generar una IP que no tendrá conectividad con la red. De esta manera el atacante dejará por fuera de conexión al huésped generándole un ataque de DOS.

Imagen 18. Formato para la opción del prefijo en el mensaje 133.

6.2 Falsificación mensaje Neighbor Advertisement durante DAD.

Luego de generada la dirección IP tentativa *link-local* o *local-unicast* el nodo procede con la verificación de la unicidad de la IP en la red. Para este trabajo envía un mensaje *Neighbor Solicitation* indicando en el encabezado IP una IP de destino *multicast* y la IP de origen como *Unspecified*. En el encabezado ICMP, indica el campo ***target IP*** la IP tentativa generada aleatoriamente, a través de SLAAC, DHCPv6, manualmente o aleatoriamente. Con esta información los nodos receptores reconocen que el mensaje corresponde a la detección de IP duplicada. Si la IP se encuentra duplicada, el receptor deberá contestar con un mensaje *Neighbor Advertisement* señalando en el campo *target address* la dirección IP tentativa con destino *multicast,* de esta manera el emisor podrá saber que su IP tentativa está duplicada. Un atacante podría tomar ventaja de esta cualidad, contestando todos los mensajes *Neighbor Solicitation* con un *Neighbor Advertisement* advirtiendo que la IP generada ya se encuentra en uso. Repitiendo este proceso con cada mensaje *Neighbor Solicitation* que llegue podría generar un ataque de DOS en el huésped generador.

6.3 Inundación Neighbor Cache.

La cabecera del mensaje *Neighbor Advertisement* en el protocolo *Neighbor Discovery* para IPv6 tiene el campo **O** que permite indicarle al receptor actualizar la dirección MAC. Esta dirección se encuentra en uno de los campos opcionales. El protocolo *Neighbor Discovery* no limita la cantidad de mensajes *Neighbor Advertisement,* dándole la alternativa a un atacante para enviar n cantidad de datagramas con el campo **O** activado. En cada uno de estos datagramas puede incluir una dirección MAC diferente, haciendo que la tabla *Neighbor Cache* se inunde de registros falsos generando un bloqueo en el núcleo del sistema derivando en un ataque DOS.

6.4 Modificación del MTU.

Como se explicó anteriormente, los mensajes *Router Advertisement* del protocolo *Neighbor Discovery* para IPv6 tiene en su encabezado la opción de incluir la unidad máxima de transmisión (MTU). Explotando la carencia de integridad de los mensajes, un atacante puede aprovechar este campo para indicarle a un nodo que el MTU es de tamaño menor al real asignado por los enrutadores. Por ejemplo, una red con MTU de 1500 bytes, un nodo puede recibir un mensaje *Router Advertisement* indicándole que el MTU es de sólo 100 bytes. Lo anterior obligará al huésped a fragmentar casi todos sus paquetes causando mayor tráfico en la red, aumentando el consumo de procesador para generar la cantidad de paquetes. De igual manera un atacante puede incrementar el MTU de la red para generar paquetes más grandes, aprovechando los 32 bits del campo, forzando al enrutador a desfragmentarlos para hacerlos llegar a su destino, igualmente el procesador del emisor se verá exigido para generar los paquetes con mayor tamaño. Ambos ataques pueden derivar en un ataque DOS afectando la red, el enrutador y/o el nodo emisor.

6.5 Falsificación del prefijo.

Al igual que en la modificación del MTU, también se puede enviar un prefijo falsificado en los mensajes *Router Advertisement.* Para la generación de la IP g*lobal-unicast* el nodo requiere conocer el prefijo de la red. Un atacante puede enviar un prefijo de red falsificado para hacer que el nodo genere una IP no alcanzable en la red, causando conflictos en el enrutamiento de sus mensajes, dejando el huésped con una IP asignada pero sin conexión a la red provocado por la modificación del prefijo. De la misma manera que en el ataque al MTU, los atacantes aprovechan la falta de integridad de los mensajes.

7 PRÁCTICA DE LAS TECNICAS DE ATAQUE EN EL PROTOCOLO NEIGHBOR DISCOVERY.

Para realizar los ataques anteriormente descritos, se utilizaron las herramientas que pertenecen a al conjunto THC-IPV6 [36]. Esta amplia gama de aplicaciones son utilizadas para realizar todos los ataques de este proyecto. Gracias a su extensa serie de utilidades que contiene, permite realizar pruebas de vulnerabilidad en el protocolo tanto IPv6 como ICMPv6. Los ataques realizados a continuación son ejecutados en ambientes controlados y de propiedad exclusiva del tesista. Todos los ataques acá descritos son aplicados con fines académicos. En el ambiente destinado para las pruebas, fue necesario realizar la adaptación de un sistema operativo Ubuntu 12.04.3 LTS con capacidades de enrutador. El equipo tiene como características 78 GB de disco duro, 758 MB de memoria RAM y un procesador Intel Celeron a 2.40 GHz.

7.1 Falsificación mensajes Router Advertisement.

Para el ataque de falsificación de mensajes *router advertisement* se hizo uso de la herramienta fake_router6 [37]. La aplicación permite generar paquetes *router advertisement* indicando en las opciones del paquete que el atacante es el enrutador, adicionalmente se le envía un prefijo de red falso y de esta manera engañar a la victimas dejándola por fuera de la red sin redireccionar los paquetes. Este es considerado un ataque de denegación de servicio debido a que inhabilita el uso de la red a la víctima y a la vez ningún huésped podrá comunicarse con la víctima.

Antes de realizar el ataque se verifica que la víctima tiene conexión con el enrutador oficial de la red. Ver Imagen 19

```
ping 2001:db8:1::2 -t
```

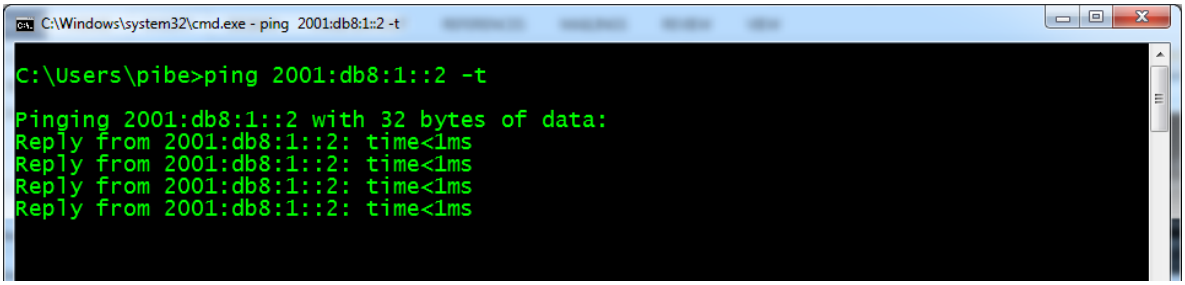


Imagen 19. Ejecución comando ping hacia el enrutador

Adicionalmente se verifica la puerta de enlace (*getaway*) de la víctima y su dirección *Global Unicast* antes de proceder con el envío de los mensajes *router advertisement* falsificados. Ver Imagen 20

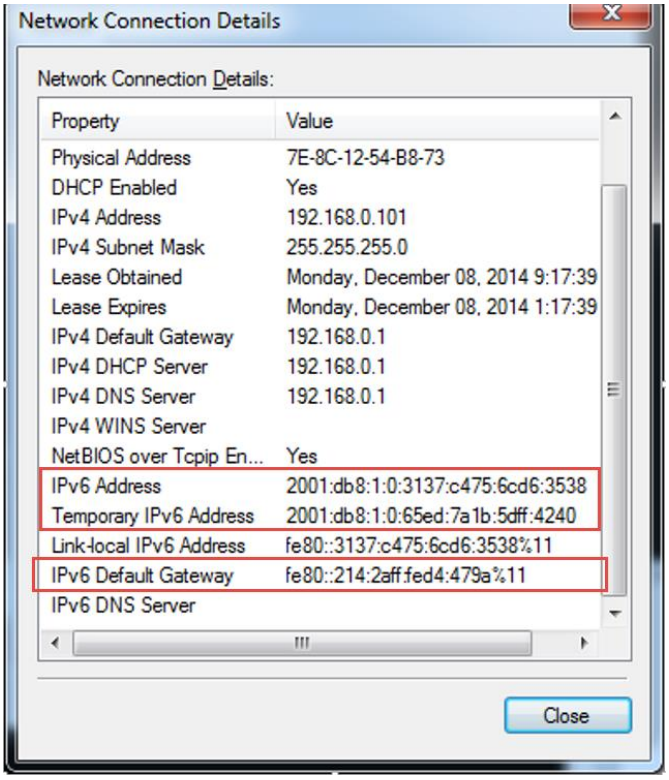


Imagen 20. Se muestra IP Global Unicast y puerta de enlace de la víctima.

Igualmente, se verifica la tabla de enrutamiento con el comando `route print`. Ver Imagen 21

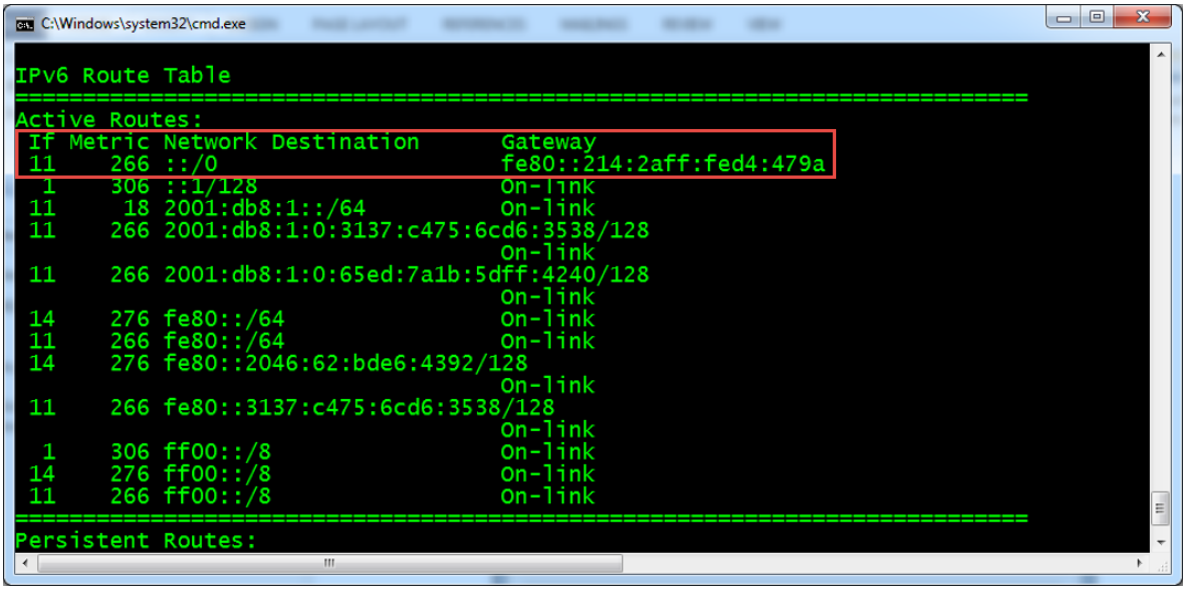


Imagen 21. Tabla de enrutamiento antes del ataque.

Luego de realizadas todas las validaciones se procede a implementar el ataque. Con la herramienta fake_router6 se envía un mensaje falsificado *router advertisement* a la víctima indicándole que el atacante es el nuevo enrutador, su prefijo de red con el que debe generar la dirección IP *Global Unicast* y una dirección MAC falsa para que al buscar la MAC no pueda realizar ninguna acción.

Con el comando `fake_router6 eth0 1::/64 1:1:1:1:1:1` se inicia el ataque. Ver Imagen 22

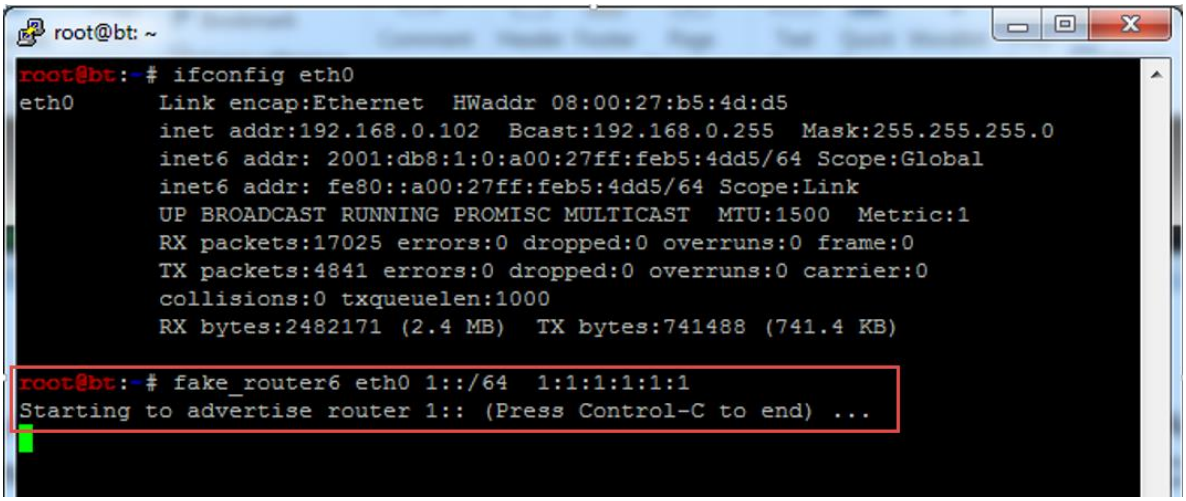


Imagen 22. Inicia el ataque de falsificación mensajes router advertisement.

Se verifica en el equipo de la víctima que la puerta de enlace (*getaway*) se ha modificado a la dirección *link-local* del atacante, y su dirección IP *Global Unicast* ha cambiado con el prefijo indicado `1::`. Ver Imagen 23

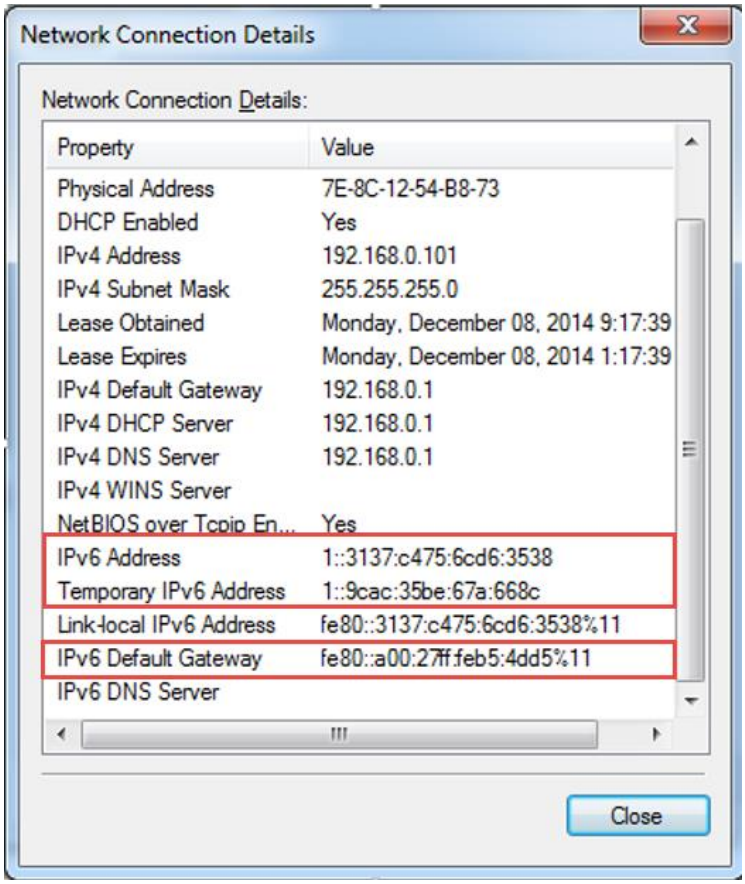


Imagen 23. Nueva dirección IP Global Unicast y puerta de enlace en la víctima luego del ataque.

Nuevamente, se verifica la tabla de enrutamiento con comando `route print` (Ver Imagen 24) y se evidencia que ha cambiado su valor comparado con el anterior, y ahora contiene la dirección *link-local* del atacante (ver Imagen 25).

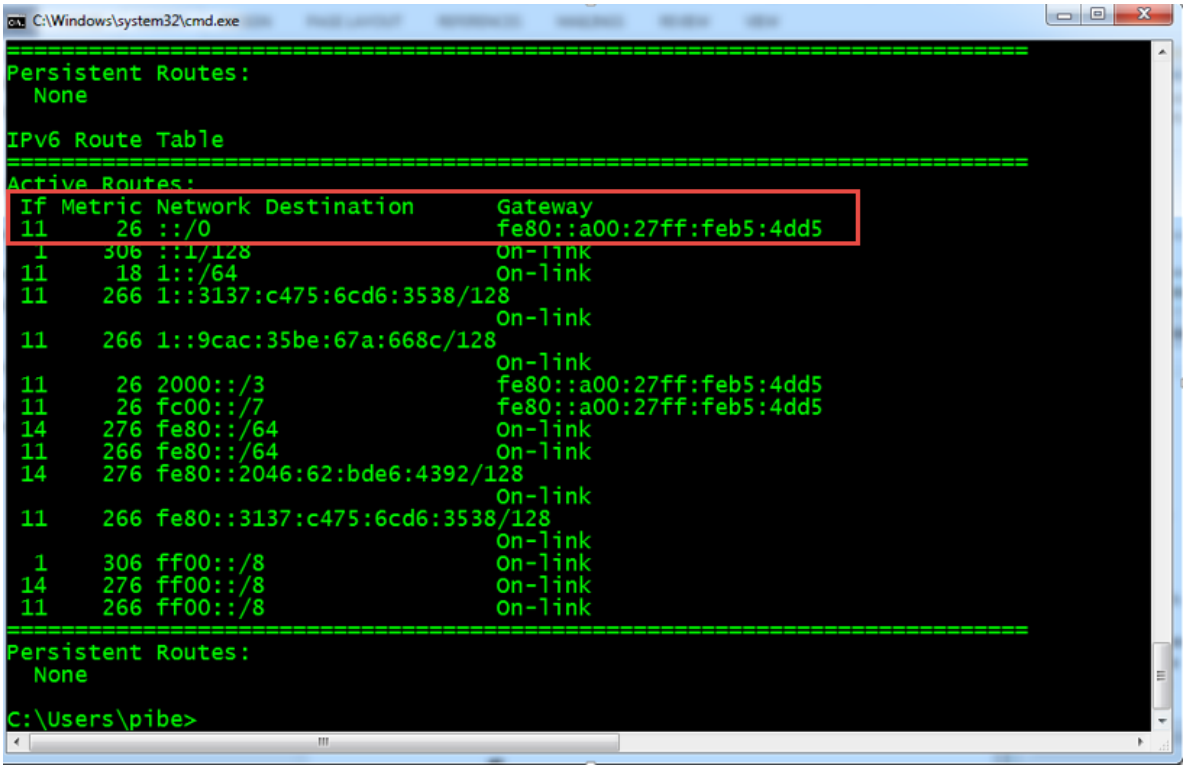


Imagen 24. Ejecución comando route print.

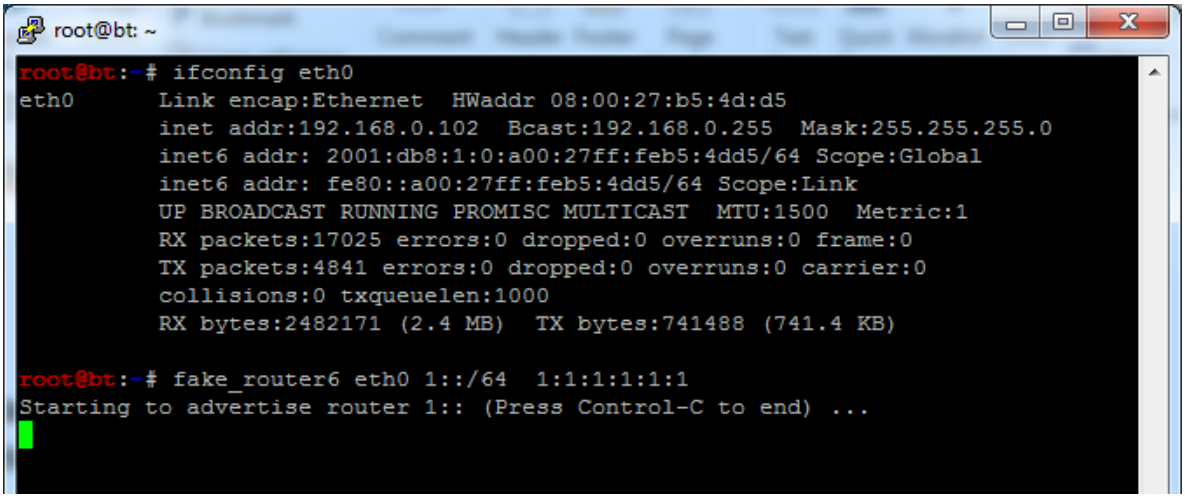


Imagen 25. Dirección IP link-local del atacante.

Se verifica el enrutador oficial de la red y mantiene su misma dirección IP enrutando los paquetes de la red sin inconveniente. Ver Imagen 26.

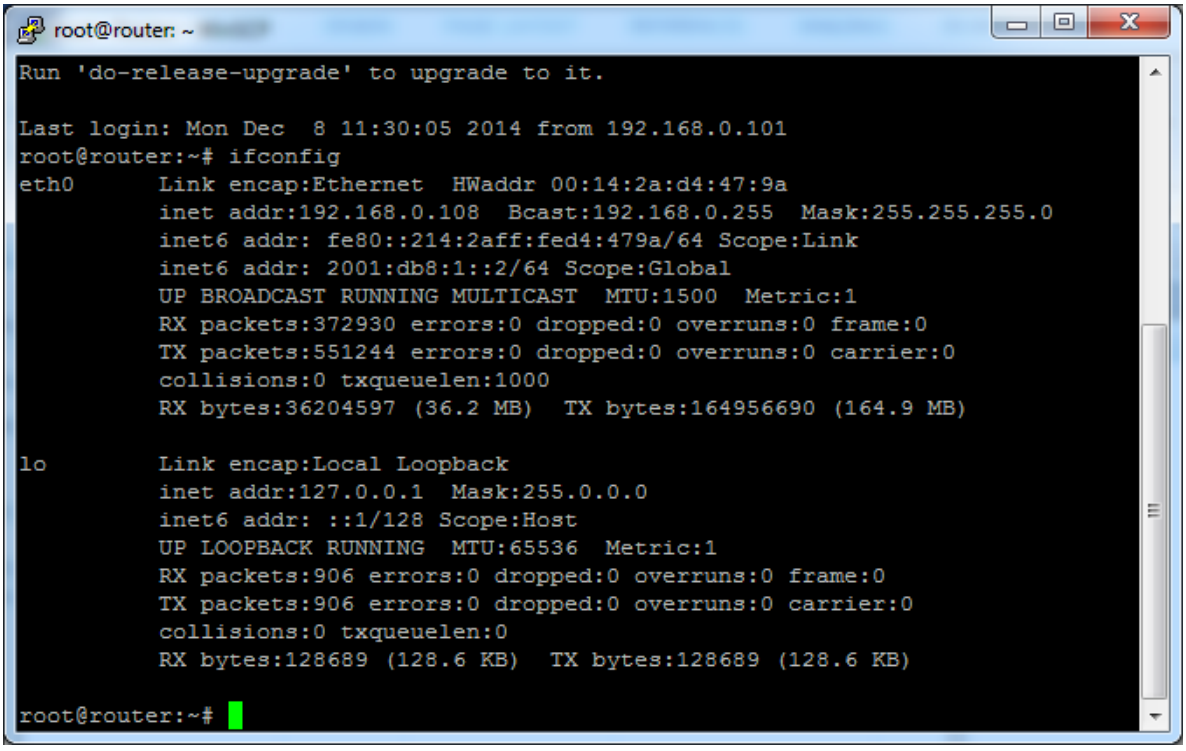


Imagen 26. Enrutador oficial.

Posterior al ataque, desde la máquina de la víctima se ejecuta el comando `ping 2001:db8:1::2 -t` al enrutador original en la dirección *global link* 2001:db8:1::2 y se observa que no hay conectividad. Ver Imagen 27.

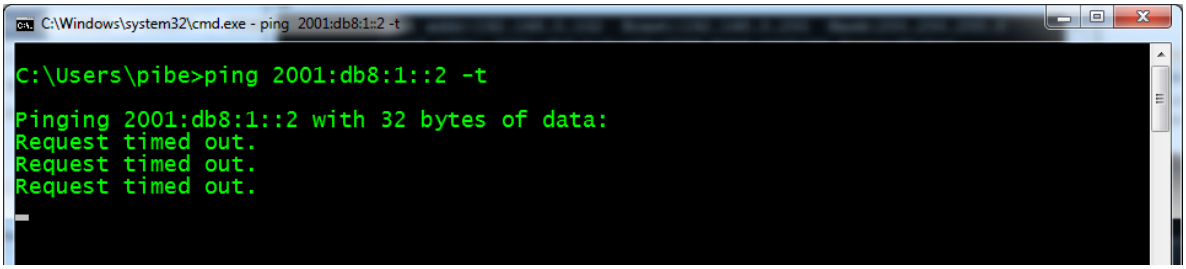


Imagen 27. Ejecución comando ping posterior al ataque.

Durante el ataque se realizó la captura de datos con la herramienta Wireshark (ver Imagen 28. Analizando el mensaje *router advertisement (*ver Imagen 29) se logra observar en las opciones del mensaje que la dirección del enrutador falso es la misma IP del atacante, en las opciones del mensaje contiene el prefijo `1::` para generar la IP *Global Unicast* y por último que le mensaje es de tipo 134 en el protocolo ICMPv6.

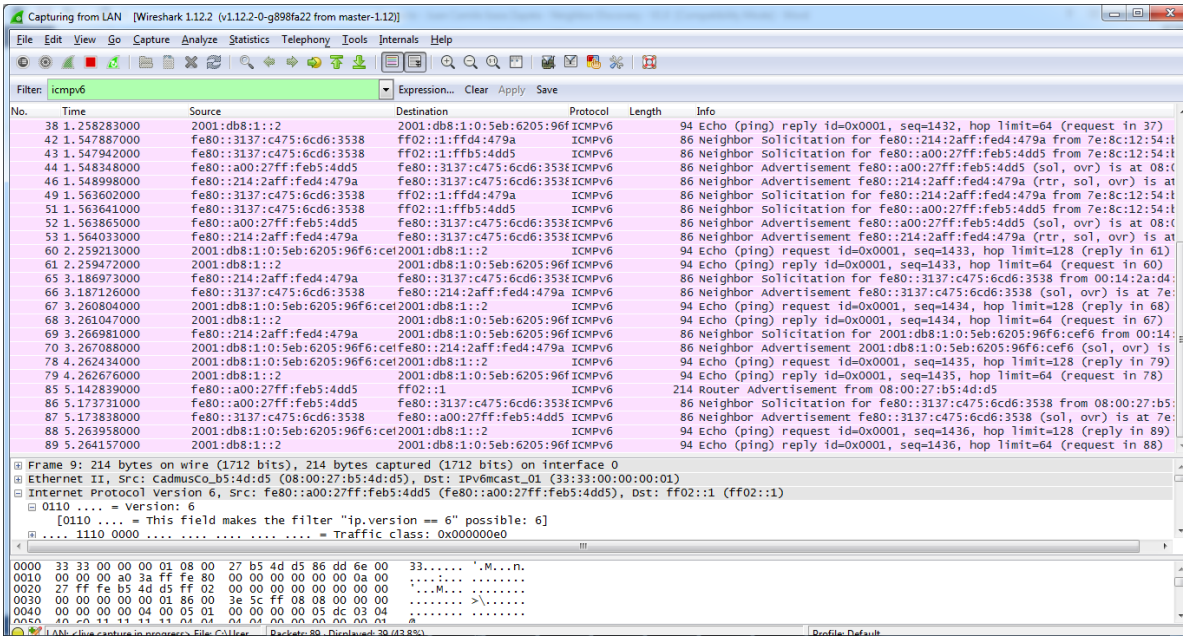


Imagen 28. Captura de datos con Wireshark.

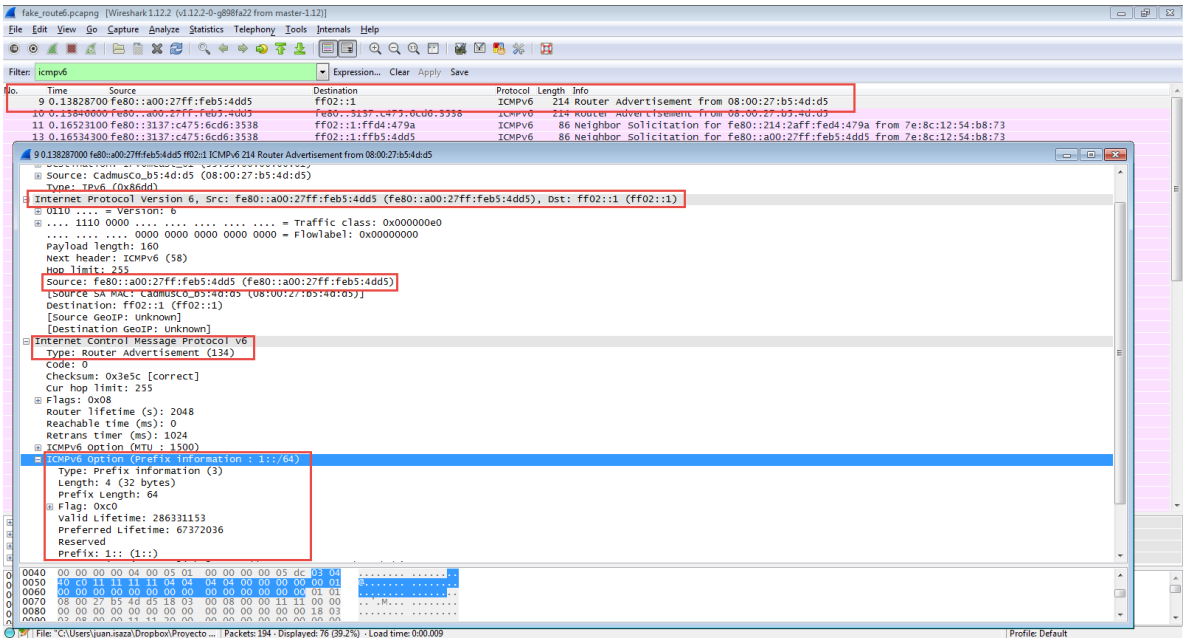


Imagen 29. Análisis mensaje router advertisement.

7.2 Falsificación mensaje Neighbor Advertisement durante DAD.

Para el ataque anteriormente descrito de falsificación mensaje *neighbor advertisement* durante DAD se utilizaron dos herramientas de THC-IPV6. La aplicación detect-new-ip6 [38] y dos-new-ip6 [39]. La primera utilidad permite detectar huéspedes que se encuentren realizando el proceso DAD, mientras que la segunda herramienta envía mensajes falsificados de duplicidad en IP para evitar que un huésped obtenga dirección IP y generarle un ataque de denegación de servicio.

Para iniciar el ataque, se configura la tarjeta de red en modo promiscuo para escuchar todo el tráfico de la red con el comando `ifcofing eth0 promisc` Ver Imagen 30.

Imagen 30. Configuración de la tarjeta de red en modo promiscuo.

En la máquina del atacante se utilizan las herramientas en paralelo para iniciar el ataque indicando la tarjeta de red que se utilizará. Los comandos ejecutados son `dos-new-ip6 eth0` y `detect-new-ip6 eth0` Ver Imagen 31

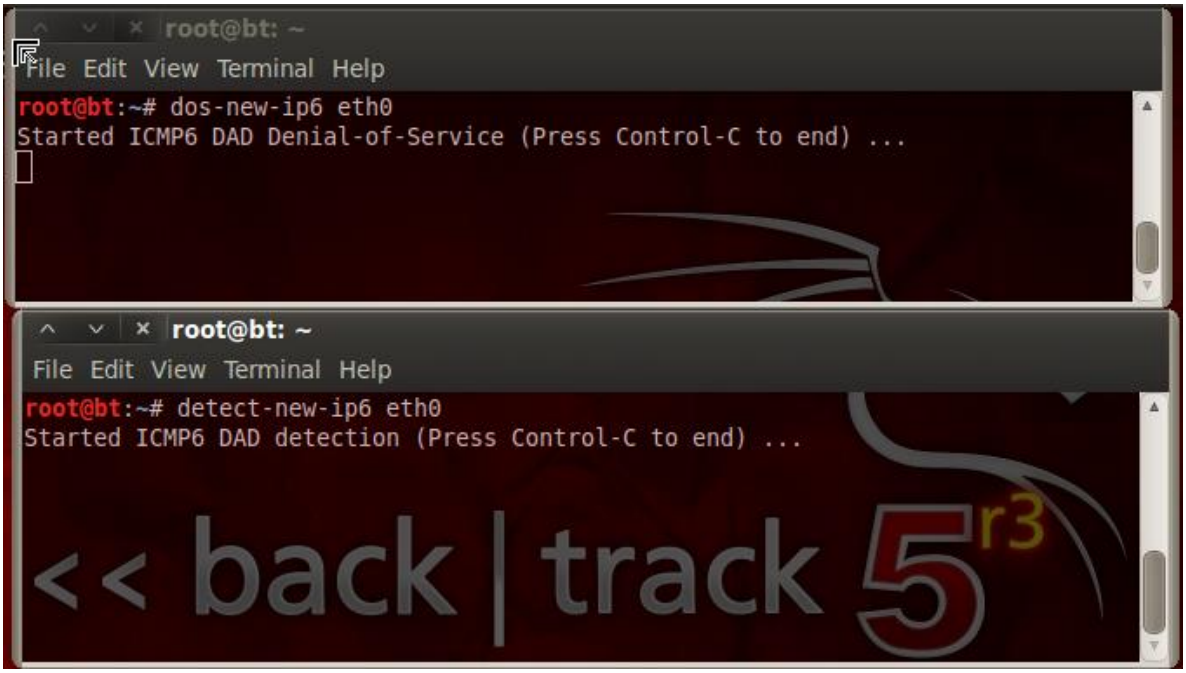


Imagen 31. Ejecución comandos dos-new-ip6 eth0 y detect-new-ip6 eth0

Se verifica que la interfaz de red de la víctima se encuentre desactivada. En el ambiente controlado se utiliza la NIC con nombre LAN como se observa en la Imagen 32

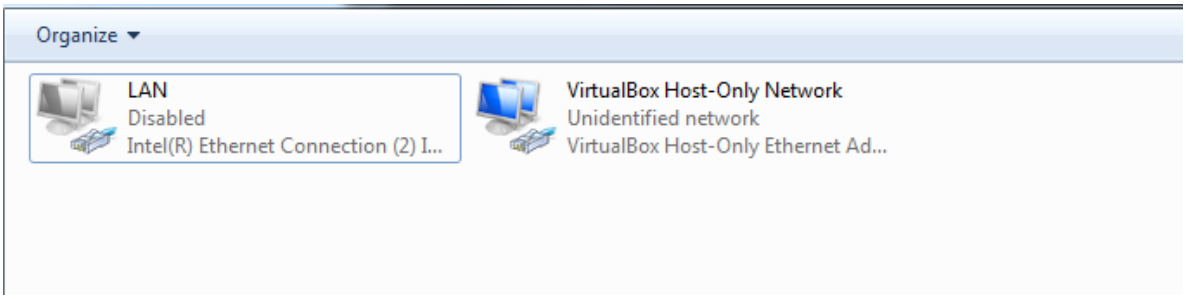


Imagen 32. Interfaz de red "LAN" desactivada.

Se enciende la tarjeta de red para que inicie el proceso de asignación de IP y DAD. Ver Imagen 33

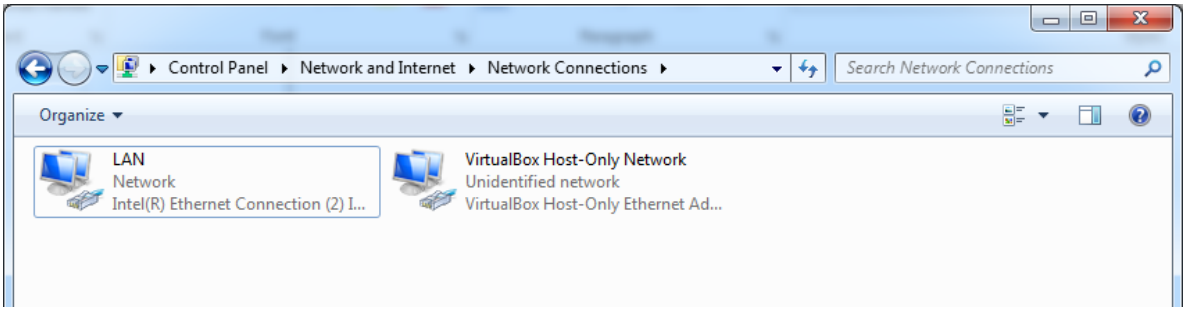


Imagen 33. Activación tarjeta de red para la ejecución de la prueba.

Debido a que el atacante se encuentra respondiendo a todos los intentos de asignación IP con mensajes de IP duplicada el proceso en el host de la víctima falla y genera el mensaje de error que se observa en la Imagen 34

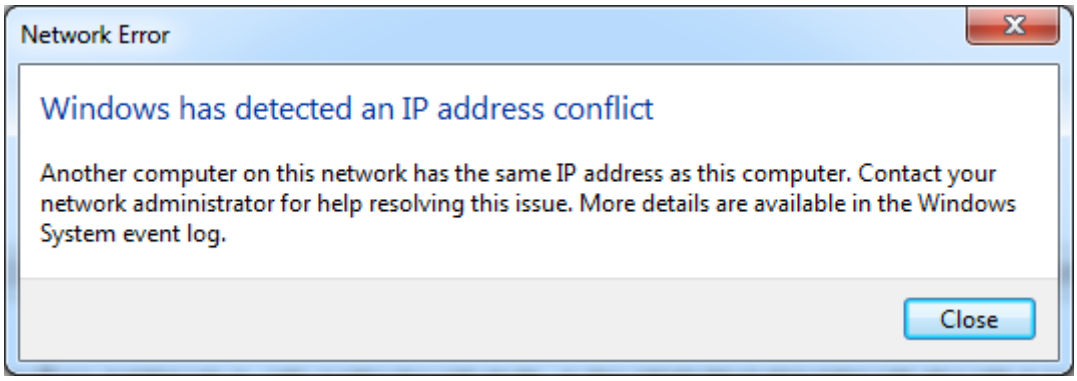


Imagen 34. Error generado por el sistema operativo Windows 7 luego de varios intentos de asignación de IP

Se revisa la configuración de la tarjeta de red y se evidencia que no se asigna dirección IP ni *link-local* ni *Global Unicast.* Ver Imagen 35

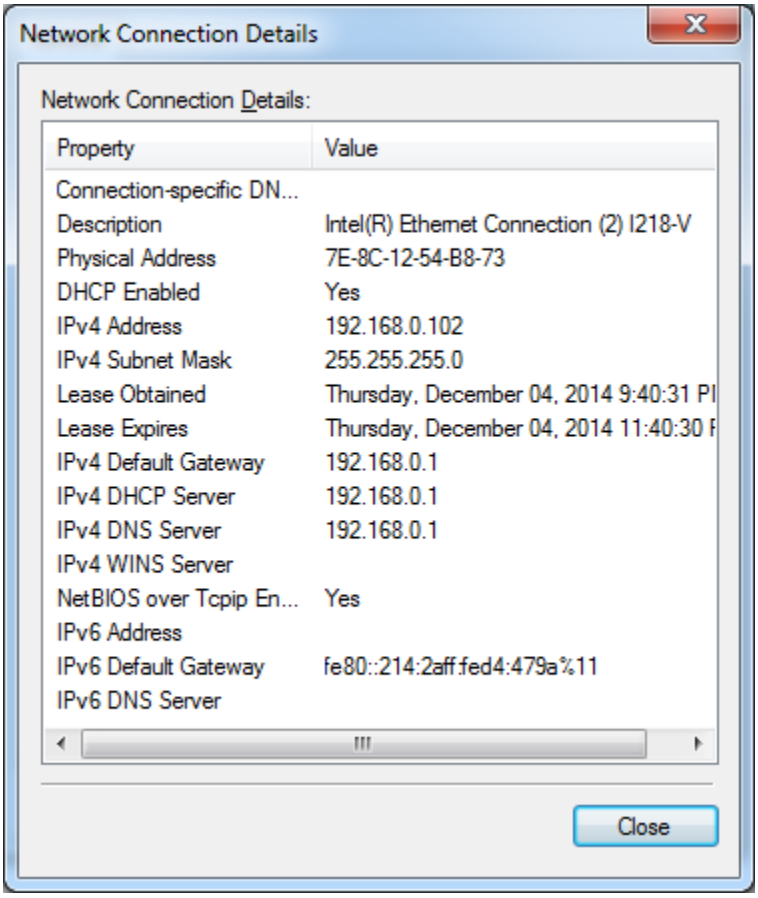


Imagen 35. Interfaz de red sin asignación de IP

Revisando la configuración del atacante, se tienen los paquetes detectados y falsificados cada que se recibe un mensaje con intención de DAD. Ver Imagen 36

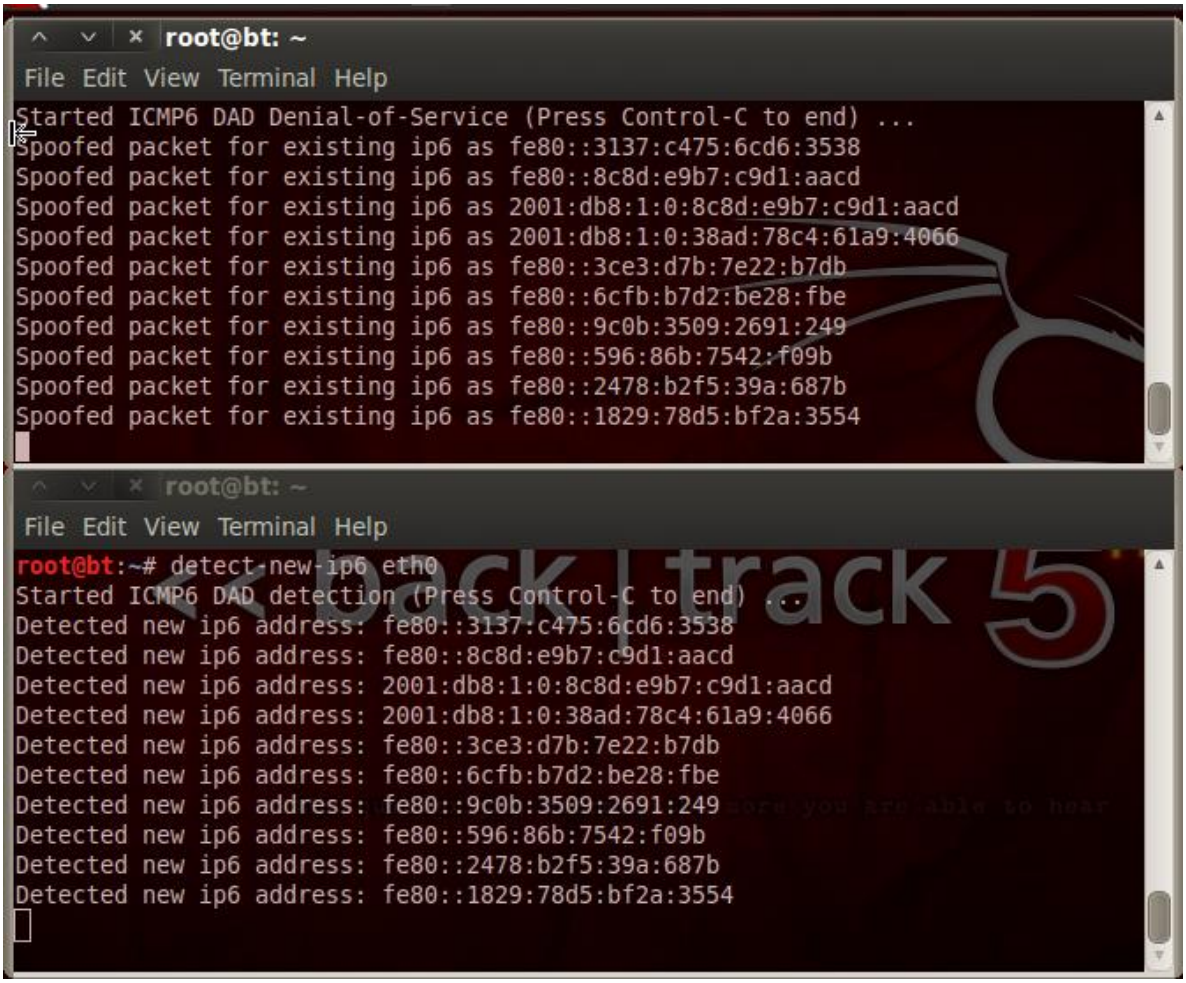


Imagen 36. Paquetes capturados y falsificados durante el ataque.

Durante el ataque se realizó la captura de datos con la herramienta Wireshark y se identificaron los mensajes *neighbor advertisement* enviados por el atacante a cada solicitud *neighbor solicitation* enviada por la víctima. Ver Imagen 37

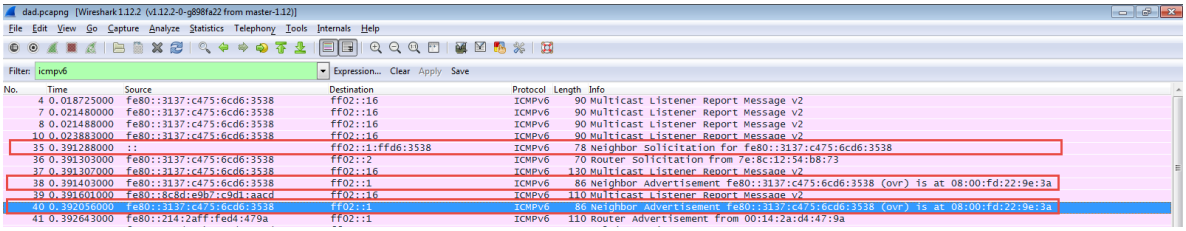


Imagen 37. Paquetes capturados con Wireshark

Analizando los paquetes *neighbor advertisement* falsificados se observa como en las opciones *target address* la dirección IP tentativa con destino *multicast* para que toda la red crea que la dirección IP está asignada. Adicionalmente se muestra la respuesta indicando el tipo de mensaje 136.

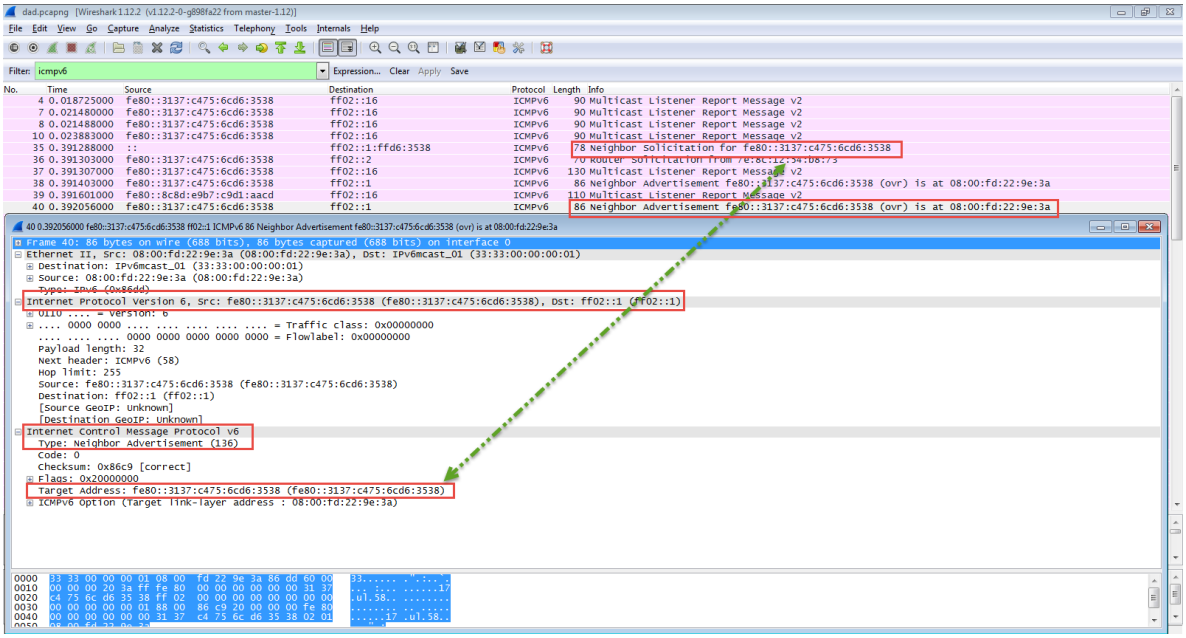


Imagen 38. Evidencia de uno de los paquetes falsificados para engañar a la víctima.

7.3 Inundación Neighbor Cache.

Para realizar el ataque anteriormente descrito de inundación a la tabla *Neighbor Cache* se utilizó la herramienta flood_router6. Esta utilidad permite la generación de paquetes aleatorios *router advertisement* con direcciones IP y Mac diferentes en cada paquete, lo cual causa una denegación de servicio a la red.

Antes de iniciar el ataque se verifica la dirección IP de la víctima como se observa en la Imagen 39 y el correcto funcionamiento del enrutador en la Imagen 40.

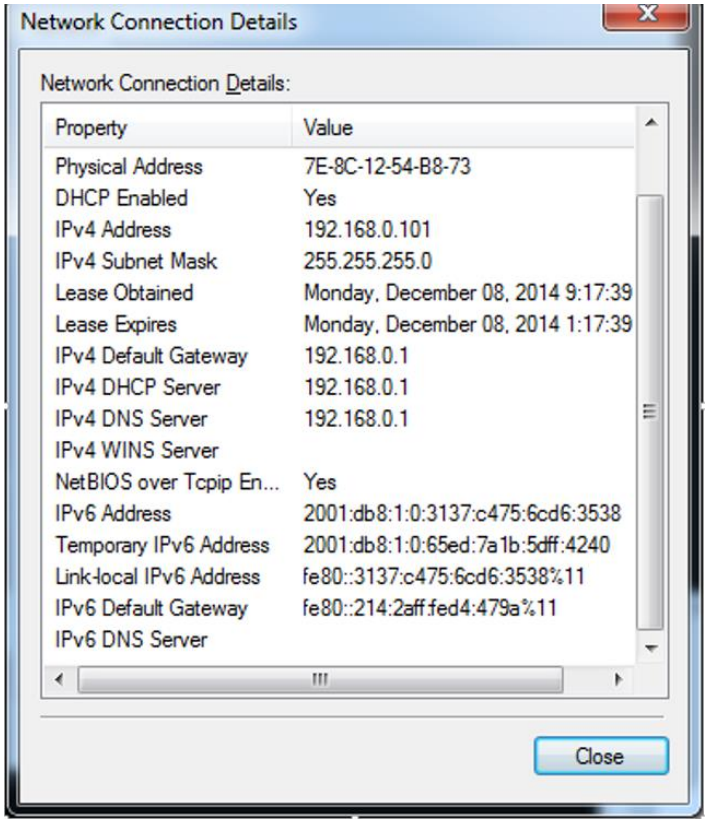


Imagen 39. Asignación correcta de la dirección IP en el equipo de la víctima.

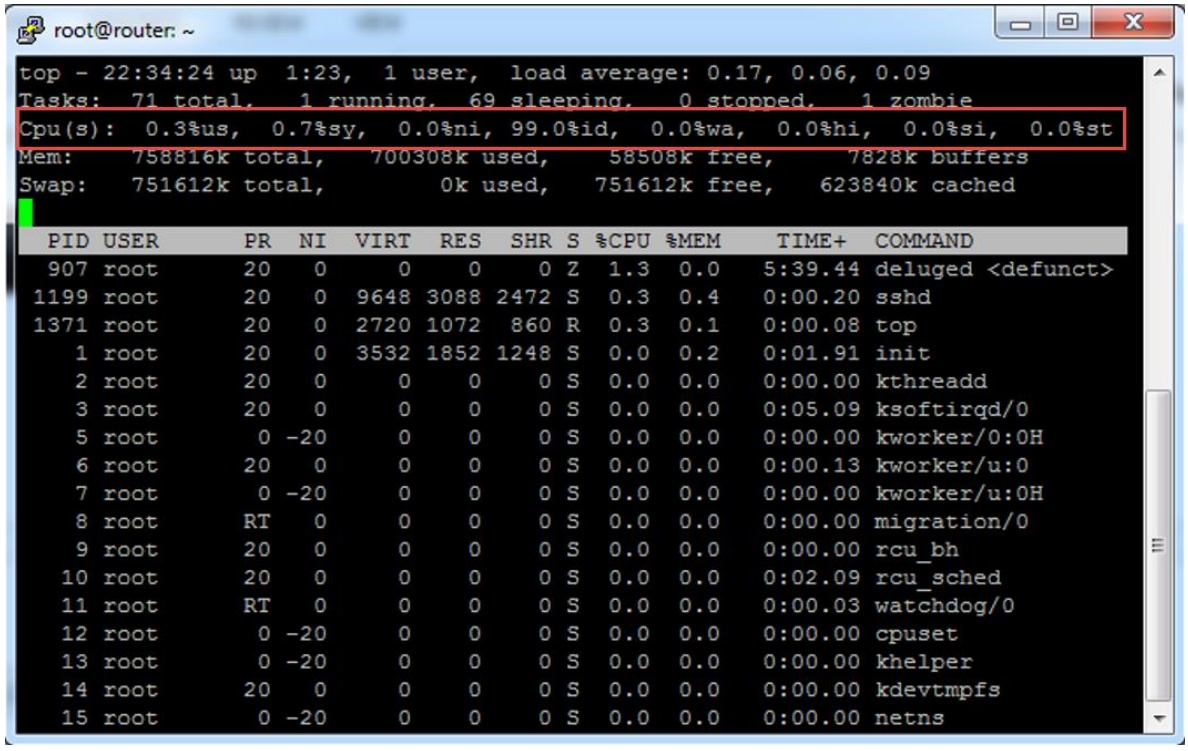


Imagen 40. Uso estándar en el enrutador de la red.

Para iniciar el ataque, desde el huésped atacante se ejecuta el comando `flood_router6 eth0` donde `eth0` es la interfaz de red que se utilizará para efectuar la práctica. Este herramienta envía cientos de paquetes por segundo a toda la red, por cada punto que se observa en la Imagen 41 se han enviado 100 paquetes. Este ataque no solo afecta a un anfitrión sino también al enrutador.

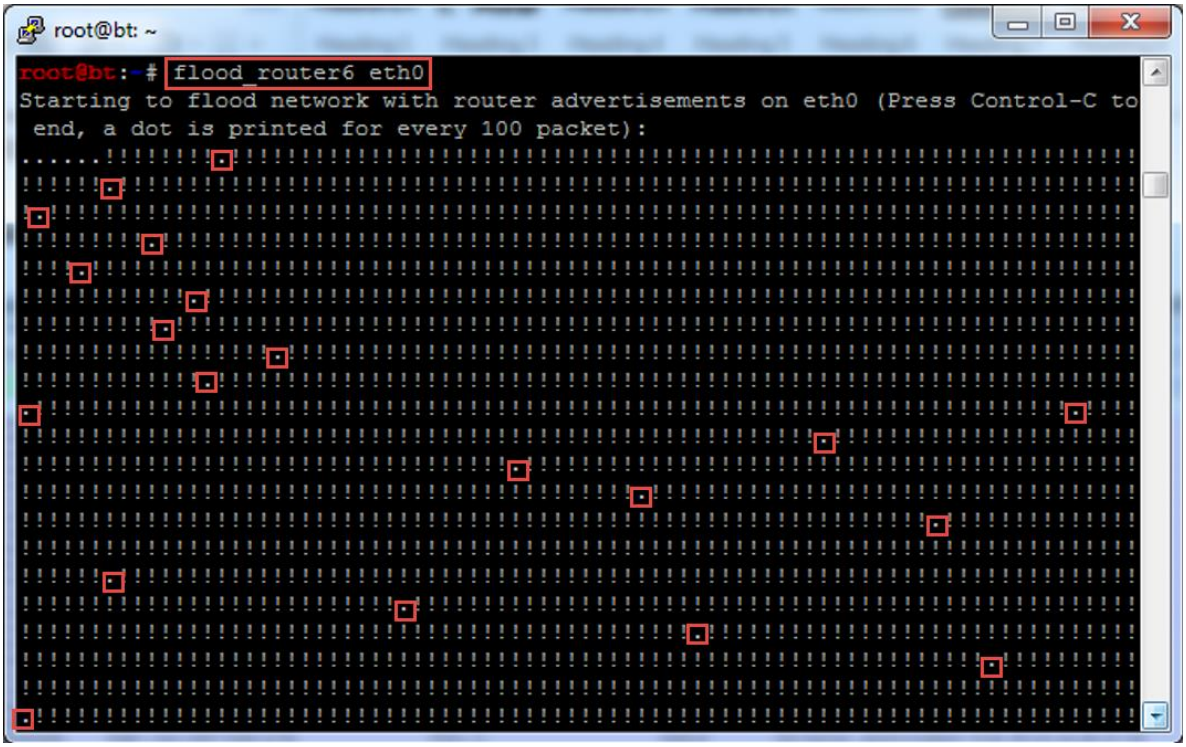


Imagen 41. Ataque de inundación de paquetes router advertisement.

Durante la ejecución del ataque se monitorea el equipo de la víctima y se evidencia como el procesador incrementa su uso de manera significativa. Ver Imagen 42.

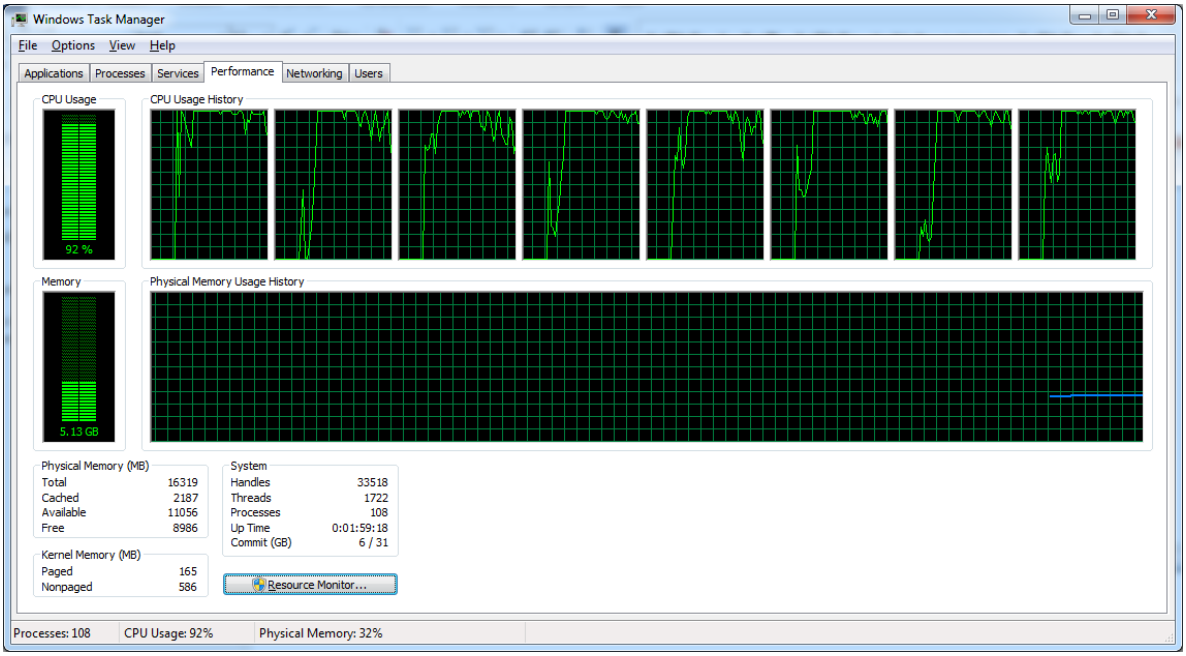


Imagen 42. Incremento del uso del procesador.

Se revisa la asignación de su IP *Global Unicast* y se encuentra nula como se observa en la Imagen 43. Luego de un par de minutos el huésped víctima se bloquea completamente y es necesario reiniciar la máquina para restablecer el servicio.

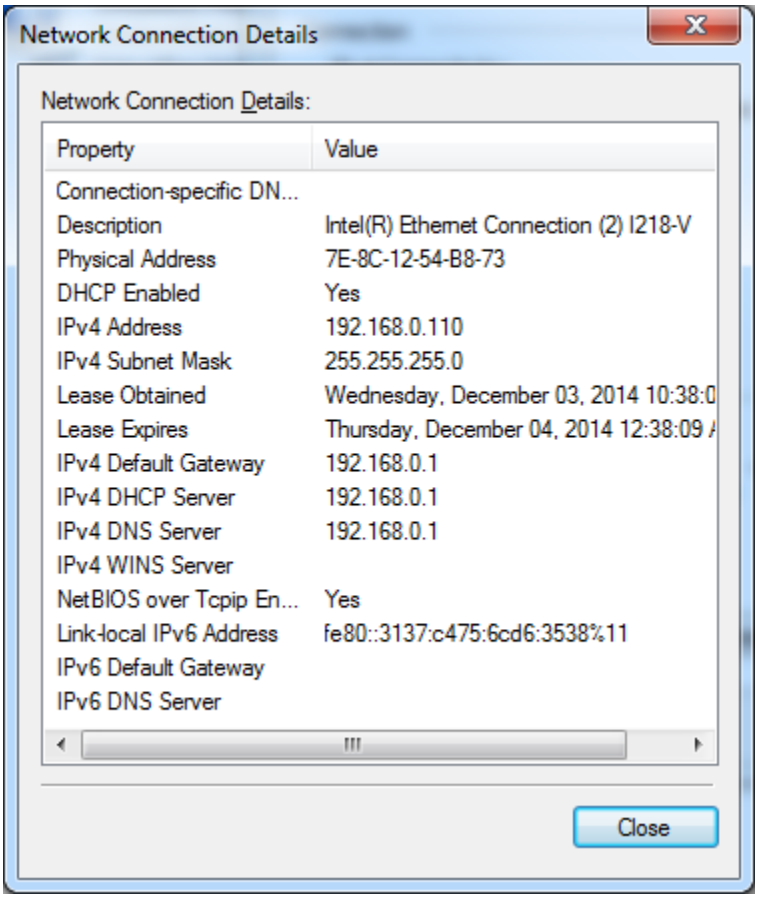


Imagen 43. Interfaz del equipo víctima sin asignación de IP.

Adicionalmente se realiza un monitoreo al enrutador de la red y se evidencia un incremento en el uso de su procesador del 95% que se puede observar en la Imagen 44. Luego de algunos minutos el enrutador no responde y es necesario reiniciarlo para restablecer el servicio al igual que el equipo de la víctima. Asimismo en la Imagen 44se observa que el servicio `radvd` está consumiendo 61% de la CPU, este servicio es el encargado de realizar las funciones de enrutamiento en la máquina adaptada para esta función, mientras que el `rsyslogd` el 33.9% de la CPU

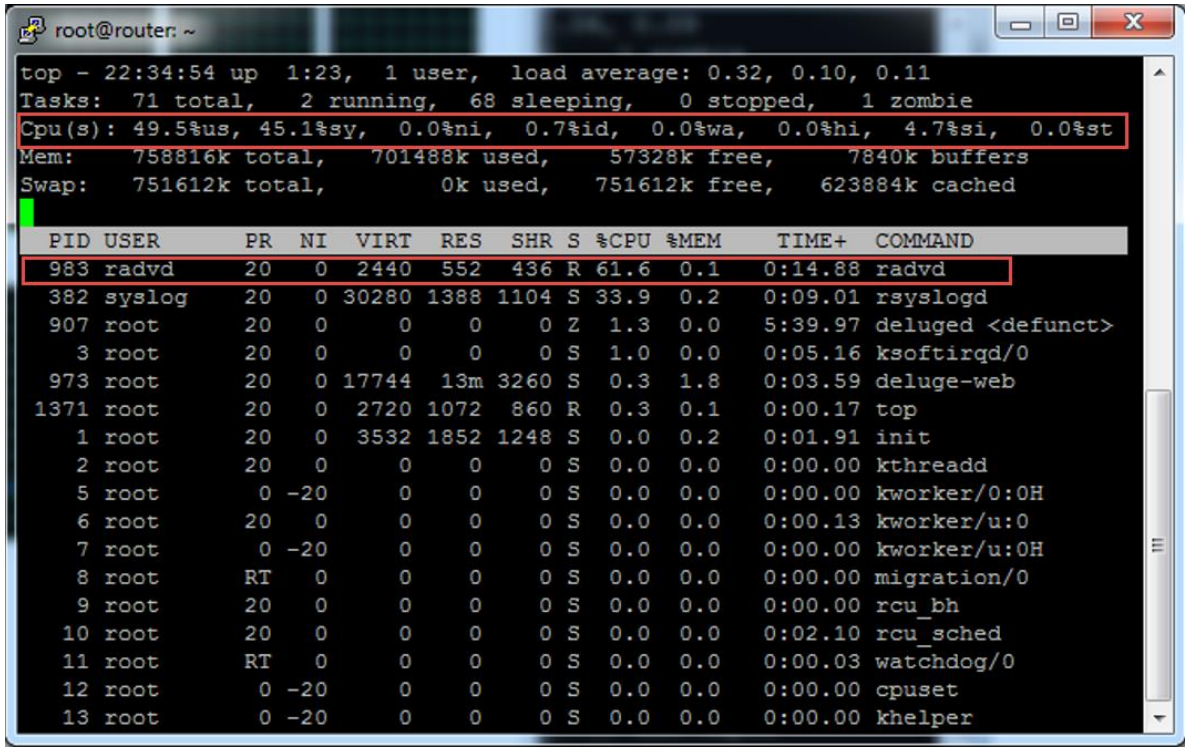


Imagen 44. Incremento en el uso del procesador del enrutador.

Luego de detenido el ataque y reiniciado el enrutador se observa la normalidad en el servicio ver Imagen 45 e Imagen 46

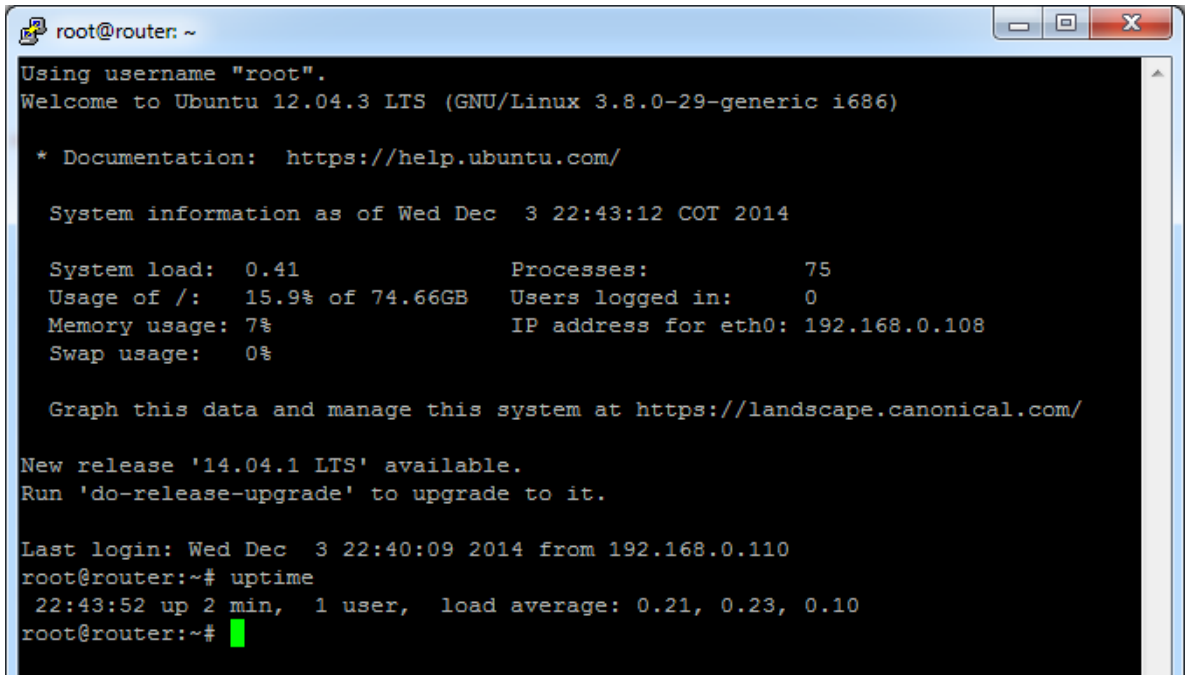


Imagen 45. Reinicio del enrutador.

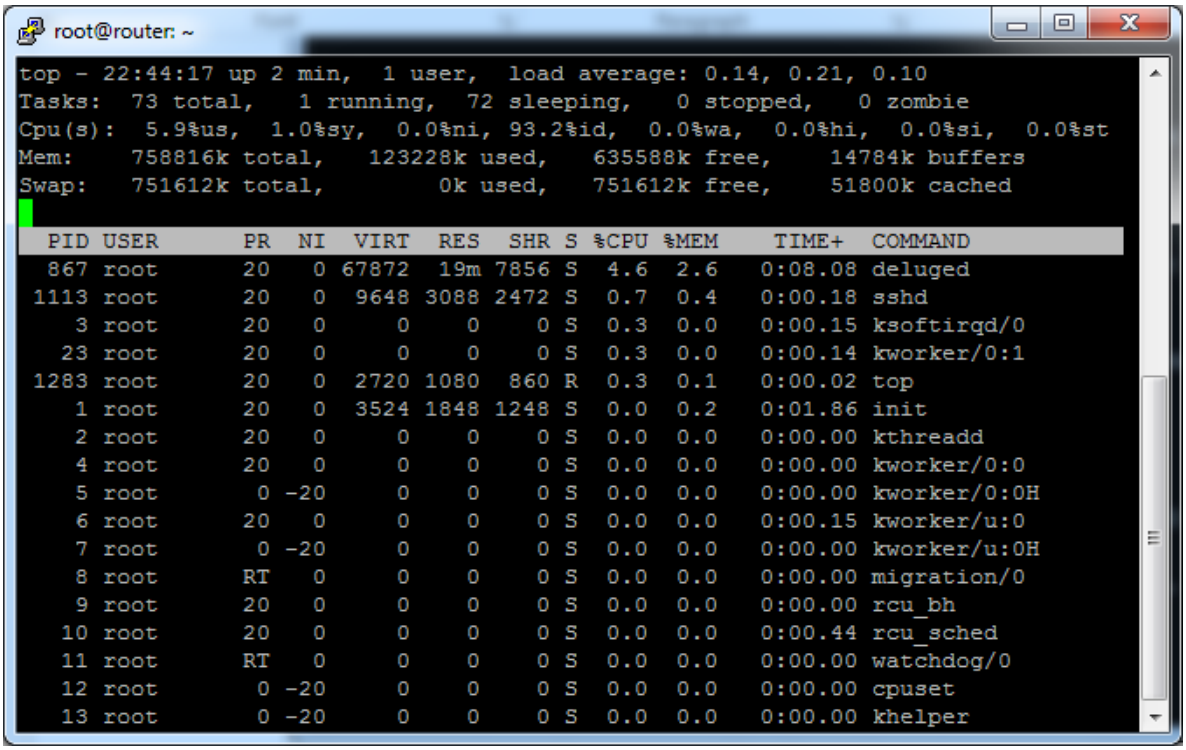


Imagen 46. Normalidad en los procesos del router luego del reinicio.

Igualmente con el huésped víctima, luego de reiniciado se observa como obtiene dirección IP sin inconveniente. Ver Imagen 47

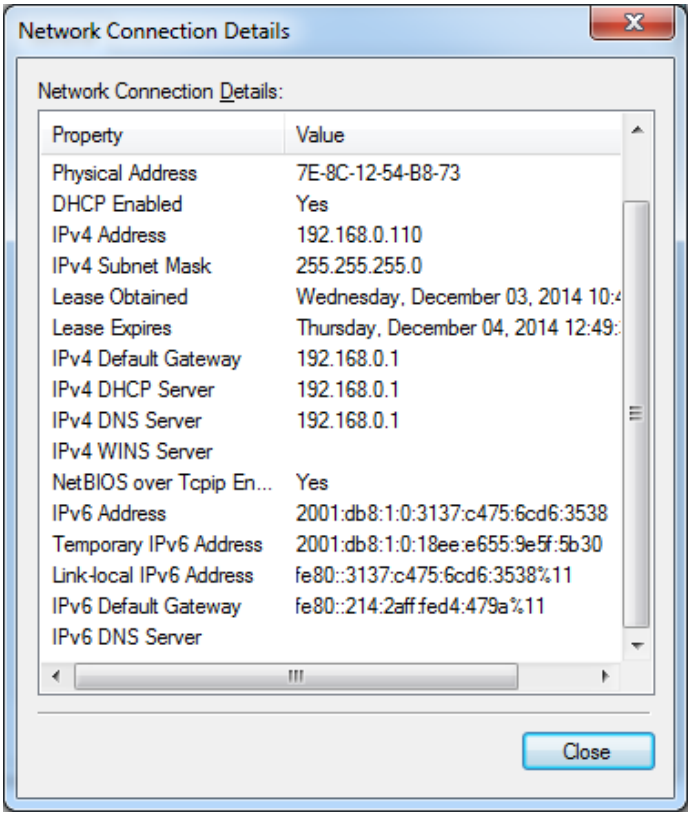


Imagen 47. Dirección IP reconfigurada luego de terminado el ataque, reiniciado el enrutador y el huésped víctima.

Con el comando `net statistics workstation` se observan las estadísticas de red una vez iniciada la interfaz. Ver Imagen 48.

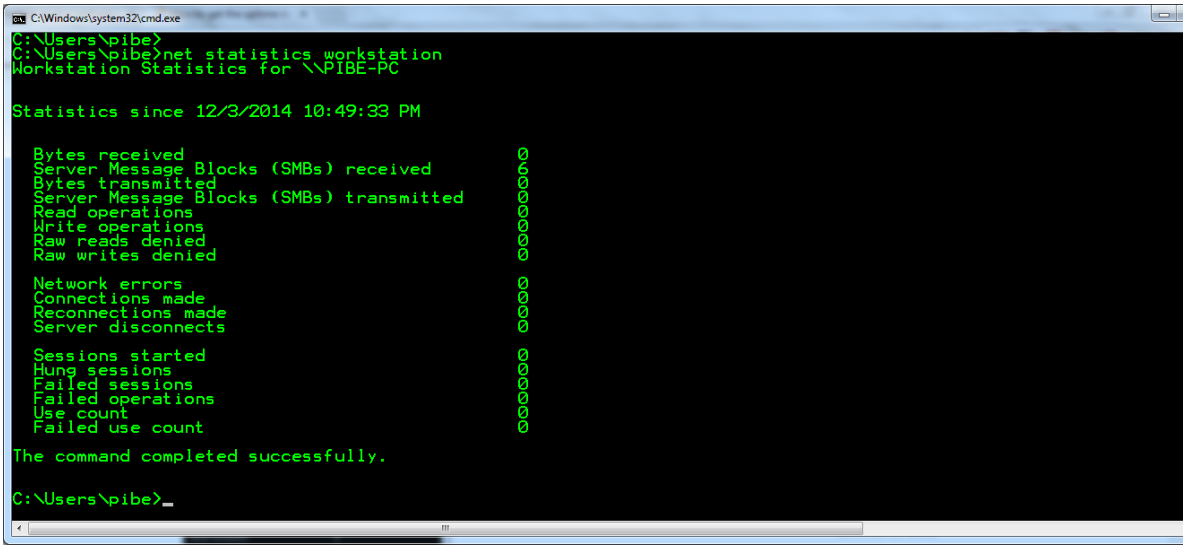


Imagen 48. Ejecución del comando net statistics workstation

Igual que en los otros ataques, se realizó la captura de los datos mediante el Wireshark (ver Imagen 49). En la herramienta de monitoreo y captura de paquetes se realiza el filtro para que solo muestre datos del protocolo ICMPv6. Durante el tiempo del ataque se lograron capturar 257008 paquetes, y se observa como cada uno de ellos contiene una IP fuente y MAC diferentes. El procesamiento de estos paquetes tanto para la víctima como el enrutador resulto en un ataque de denegación de servicio en toda la red.

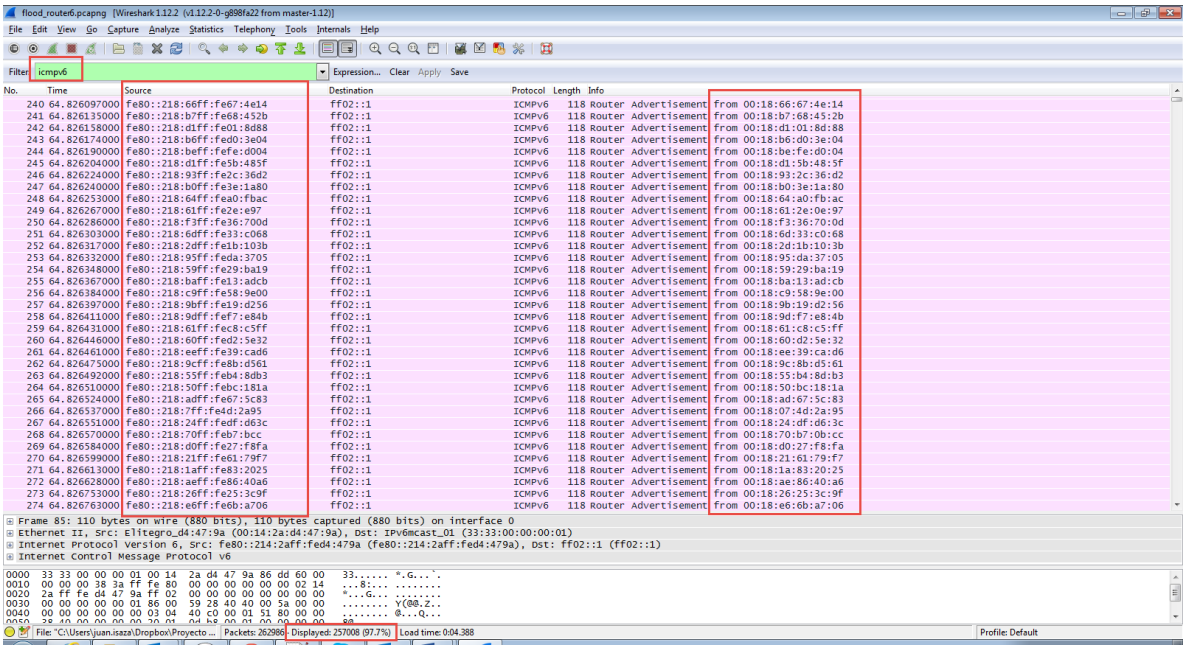


Imagen 49. Paquetes capturados con Wireshark y evidencia de la generación de paquetes aleatorios con IP y MAC diferentes

7.4 Falsificación del prefijo y modificación del MTU

Para los ataques de falsificación del prefijo y modificación del MTU se utilizará la herramienta fake_router6 [37]. Esta utilidad permite enviar simultáneamente un prefijo falso y a su vez un MTU indicado.

Antes de realizar el ataque se verifica la dirección IP de la víctima tanto gráficamente (ver Imagen 50) como en la línea de comandos (ver Imagen 51).

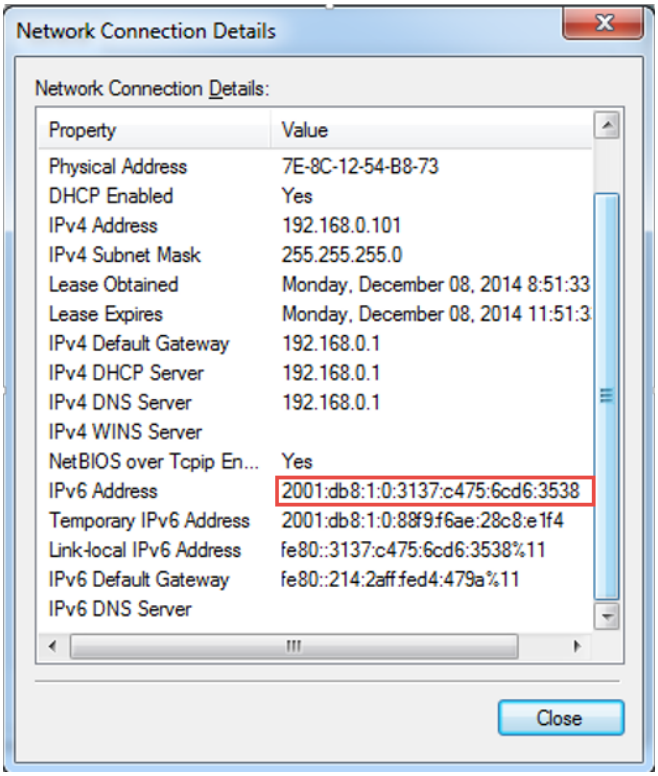


Imagen 50. Dirección IP del equipo de la víctima.

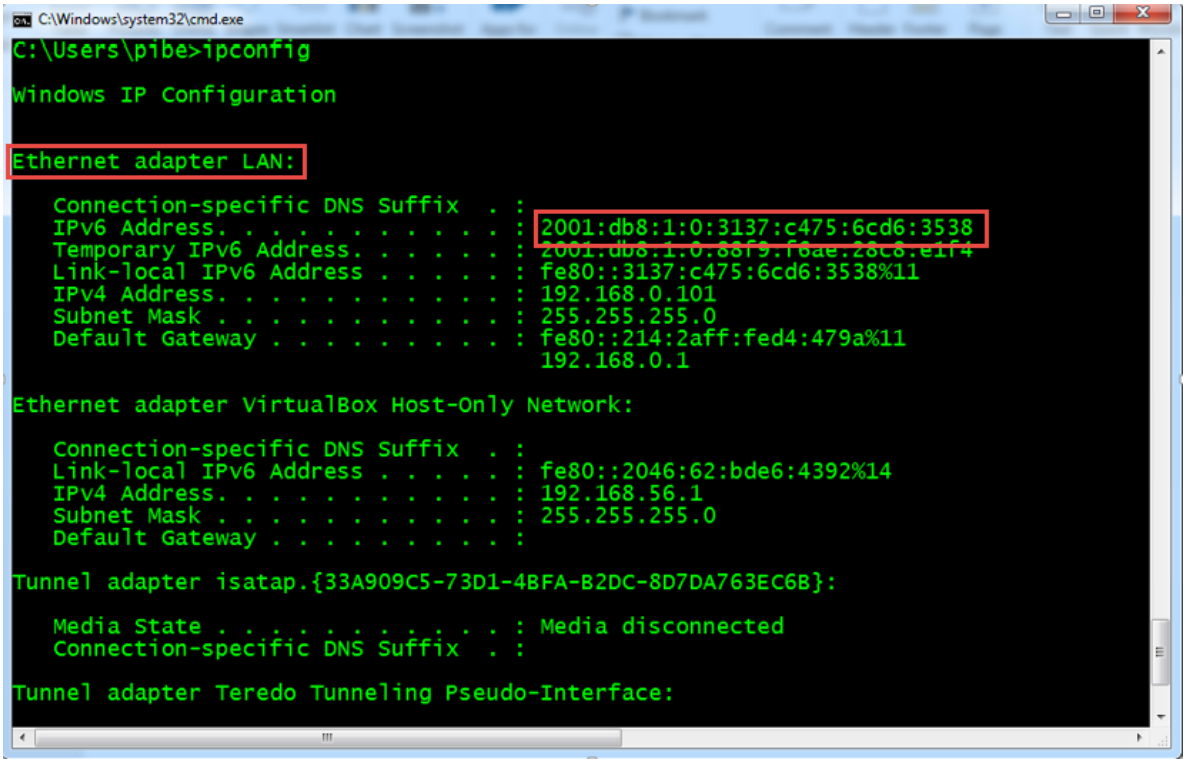


Imagen 51. Dirección IP del equipo de la víctima en consola.

Igualmente se revisa el MTU que generalmente contiene un valor de 1500. Para realizar esta verificación se utiliza el comando `netsh interface ipv6 show subinterfaces` Imagen 52

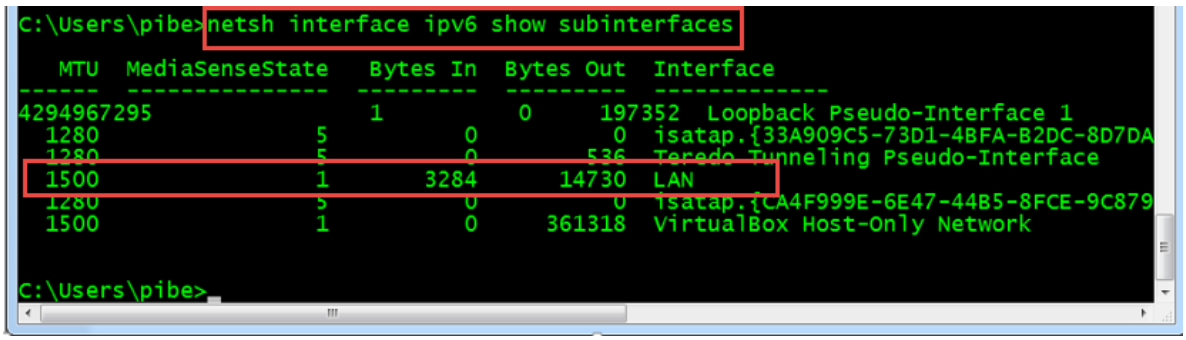


Imagen 52. Ejecución del comando netsh interface ipv6 show subinterfaces.

Para realizar el ataque se configura el redireccionamiento de los paquetes en el cliente del atacante con el comando `sysctl -w net.ipv6.conf.all.forwarding=1` Ver Imagen 53

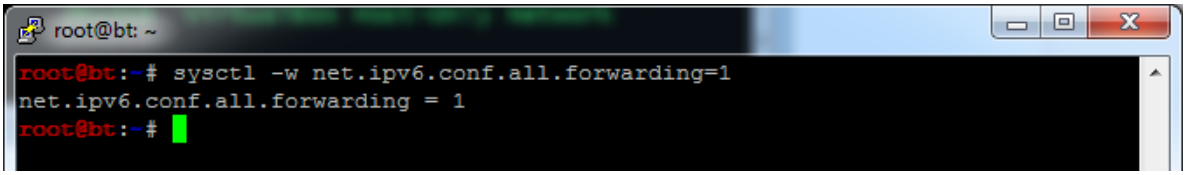


Imagen 53. Ejecución del comando sysctl -w net.ipv6.conf.all.forwarding=1

Asimismo se configura la interfaz en modo promiscuo para escuchar todo el tráfico de la red con el comando `ifconfig eth0 promisc` ver Imagen 54

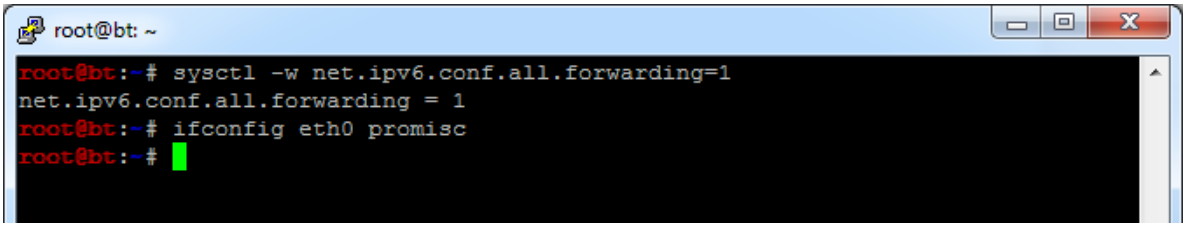


Imagen 54. Ejecución comando ifconfig eth0 promisc

Para iniciar el ataque se utiliza el siguiente comando (ver Imagen 55):

```
fake_router6     eth0     ca::8/64     fe80::a00:27ff:feb5:4dd5
fe80::a00:27ff:feb5:4dd5 1400
```

Donde:

eth0 = La interfaz de red que se utilizará para la comunicación

ca::8/64 = El prefijo de red falsificado

fe80::a00:27ff:feb5:4dd5 = Dirección IP origen

fe80::a00:27ff:feb5:4dd5 = Dirección IP del router

1400 = Valor del MTU

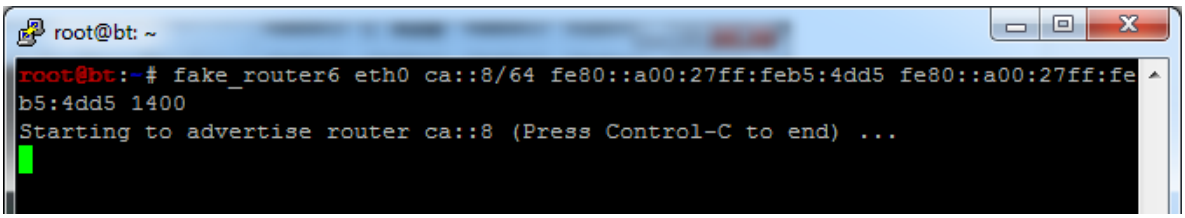


Imagen 55. Ejecución comando fake_router6 eth0 ca::8/64 fe80::a00:27ff:feb5:4dd5 fe80::a00:27ff:feb5:4dd5 1400

Durante la ejecución del ataque se revisa la IP de la víctima y se observa la dirección IP con el prefijo enviado ca::8/64 (ver Imagen 56)

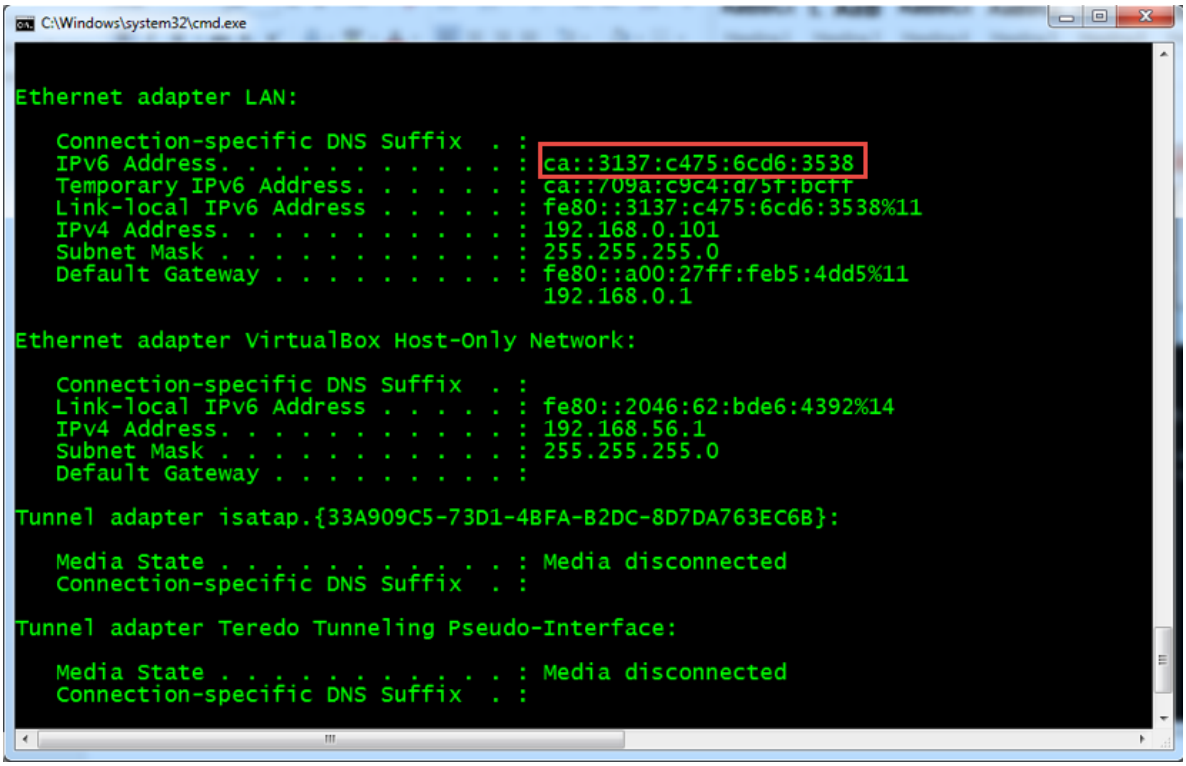


Imagen 56. Dirección IP asignada con el prefijo falso.

Adicionalmente se ejecuta el comando para revisar el MTU y se evidencia que cambio por 1400 como se envió en el mensaje falsificado. Ver Imagen 57

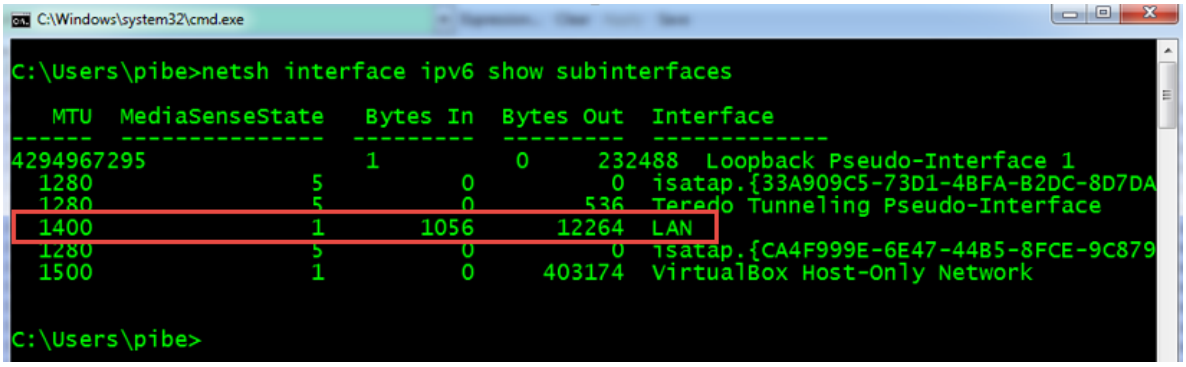


Imagen 57. Se muestra como se asignó el MTU de 1400 en vez de 1500 que es el original.

Durante la ejecución del ataque se capturaron los paquetes con la herramienta Wireshark. Se observa que es un mensaje de tipo 134, de igual modo el MTU corresponde al valor de 1400 y por último el prefijo de red es ca:: (ver Imagen 58).

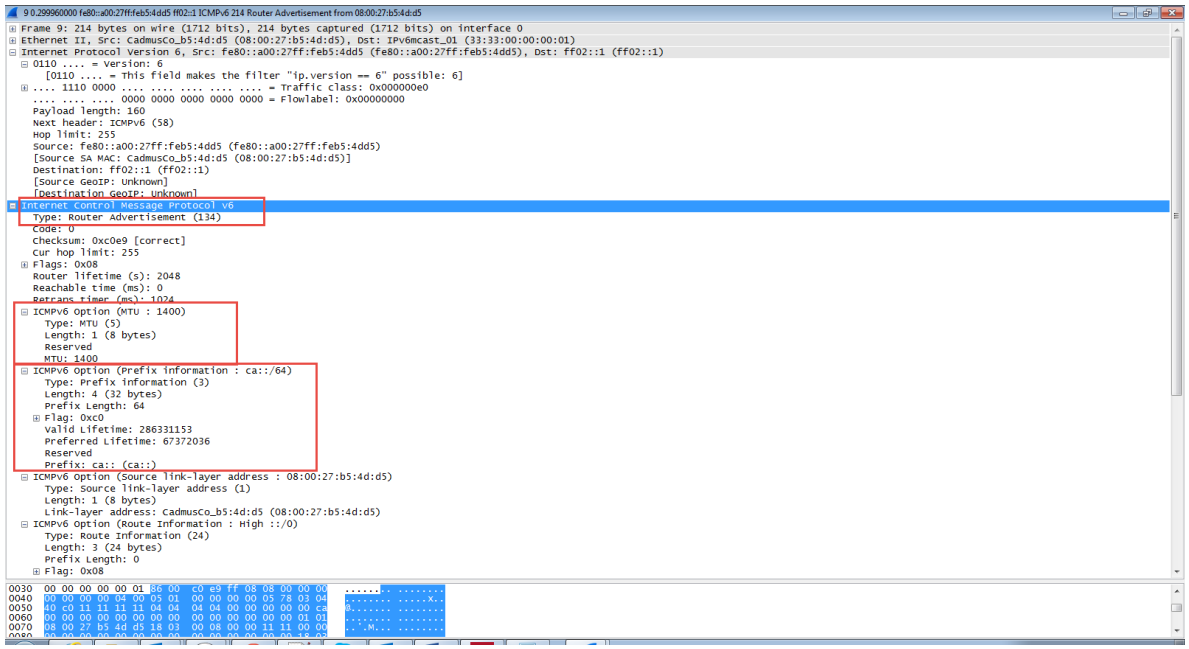


Imagen 58. Información capturada con Wireshark. Prefijo y MTU enviados falsamente.

8 MÉTODOS PARA LA PROTECCIÓN DEL PROTOCOLO NEIGHBOR DISCOVERY.

8.1 SEcure Neighbor Discovery (SEND).

En el RFC 4861 del protocolo Neighbor Discovery [20] hace referencia al método IPSec [29] para implementar una grado de seguridad a los mensajes del protocolo. Para los investigadores de esta materia, el documento carece de especificaciones de cómo implementarlo. Para ello crearon el protocolo SEND por sus siglas en inglés *SEcure Neighbor Discovery* [33].

Los desarrolladores del protocolo decidieron implementar diversos componentes para obtener seguridad en el protocolo Neighbor Discovery. Componentes como "rutas certificadas", nodos o contrapartes certificados o anclas hacen parte de esta arquitectura. Todos los nodos deben tener su ancla la cual le permite al enrutador de la red definir el camino certificado por el cual viajara el mensaje. Adicionalmente se agregan las direcciones generadas criptográficamente por su nombre en inglés *Cryptographically Generated Addresses* con sus siglas CGA, este campo será explicado más adelante.

Para el modelo de despliegue, el protocolo SEND puede manejarse mediante una infraestructura centralizada de clave privada por su nombre en inglés *public key infrastructure* con sus siglas PKI [40]. Consiste básicamente en un nodo raíz que autoriza a los enrutadores a enviar mensajes en determinado segmento de red. Y adicionalmente los huéspedes deben ser configurados con el certificado del nodo raíz. Otro mecanismo que también soporta SEND es la descentralización, para ello requiere que cada enrutador y dispositivo que se conecte a la red genere tanto su clave privada como pública y distribuir la clave pública en todos los nodos y enrutadores de la red.

El protocolo SEND solo funciona a partir del momento en que se tenga una IP asignada o preparada para ser asignada. Por lo tanto, no permite desarticular el ataque de inundación de *neighbor cache*.

El protocolo SEND fue soportado desde el inicio por la empresa DoCoMo USA Labs, pero desafortunadamente dejaron de brindarle su apoyo y el código fuente no se encuentra disponible para ser obtenido desde su sitio web [41]

8.2 Funcionamiento del protocolo SEND.

Para entender el funcionamiento del protocolo SEND es necesario aclarar las nuevas opciones que se agregan a los mensajes utilizados por el protocolo Neighbor Discovery, y los nuevos mensajes que contiene el protocolo SEND.

La primera opción agregada a los mensajes son las *rutas certificadas*, anclas o nodos de confianza. Todos los nodos de la red deben tener su ancla configurada la

cual le permita a un enrutador tener su ruta certificada para que el nodo pueda determinar el enrutador por omisión. Para poder tener una ruta certificada se agregan dos nuevos mensajes del protocolo ICMP al protocolo SEND. El primer mensaje es de solicitud de ruta certificada por su nombre en inglés *Certification Path Solicitation* con sus siglas CPS. El otro mensaje corresponde al anuncio de ruta certificada por su nombre en inglés *Certification Path Advertisement* con siglas CPA. Estos mensajes son de tipo 148 para CPS y 149 para CPA, ambos en el protocolo ICMP. Estos mensajes serán explicados más adelante.

Otra de las opciones agregadas en los mensajes del protocolo Neighbor Discovery es el campo de Direcciones Generadas Criptográficamente por su nombre en inglés *Cryptographically Generated Addresses* con sus siglas CGA [34]. Este componente es utilizado para asegurar que el emisor de un mensaje realmente es el propietario de la dirección que dice tener. Para poder realizar uso del CGA cada nodo debe generar su par de llaves privada/publica.

Otro de los elementos pertenecientes al protocolo SEND es la opción de la firma RSA [42]. Este componente permite proteger la integridad de los mensajes relacionados al protocolo Neighbor Discovery así como la autenticidad del emisor. Los creadores del protocolo decidieron adoptar el algoritmo RSA por su grado de confianza y adicionalmente no implementaron otros algoritmos para no aumentar la complejidad del funcionamiento de SEND. Por último agregaron dos nuevas opciones de sello de tiempo y *nonce* que hace referencia a un número generado aleatoriamente para evitar determinados ataques como el de falsificación de *router advertisement.*

La construcción y verificación de la firma mediante RSA es computacionalmente costosa lo que impacta el desempeño de la red. Los enrutadores se ven impactados debido a que requieren hacer gran cantidad de operaciones, la mayoría en los mensajes *router advertisement*. La recomendación brindada en la documentación es realizar las pruebas de desempeño que se consideren necesaria para no afectar el funcionamiento de red.

8.3 Opción CGA.

La opción CGA contiene el formato que muestra la Imagen 59.

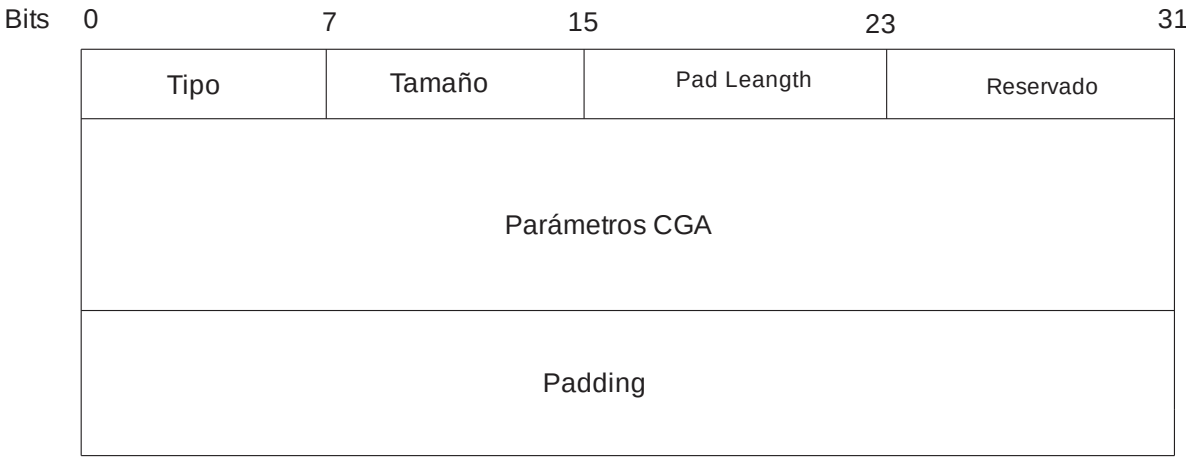


Imagen 59. Formato Opción CGA.

El campo tipo siempre tiene el valor 11 en decimal. Su valor más significativo corresponde al campo parámetros CGA y contiene el algoritmo de cifrado, el prefijo de la subred de 64 bits, la calve pública del dueño de la dirección y un modificador. La generación de la CGA puede ser consultada en [34].

Para el envío de los mensajes del protocolo ND, siempre debe contener la opción CGA desde que se utilice el protocolo SEND. En los mensajes RS no se incluye opción CGA cuando el campo dirección de origen es una dirección de tipo *unspecified*.

Todos los mensajes que no contengan la opción CGA, con excepción del caso mencionado anteriormente, deben tratarse como mensajes inseguros. El tamaño mínimo de una clave pública por omisión es de 1024 bits y debe tener un límite máximo de 2048 bits. Como buena práctica, cualquier implementación de SEND debe ser prudente con el tamaño de la calve pública para no afectar el desempeño de la red.

8.4 Opción Firma RSA.

La opción firma RSA contiene el formato que muestra la Imagen 60.

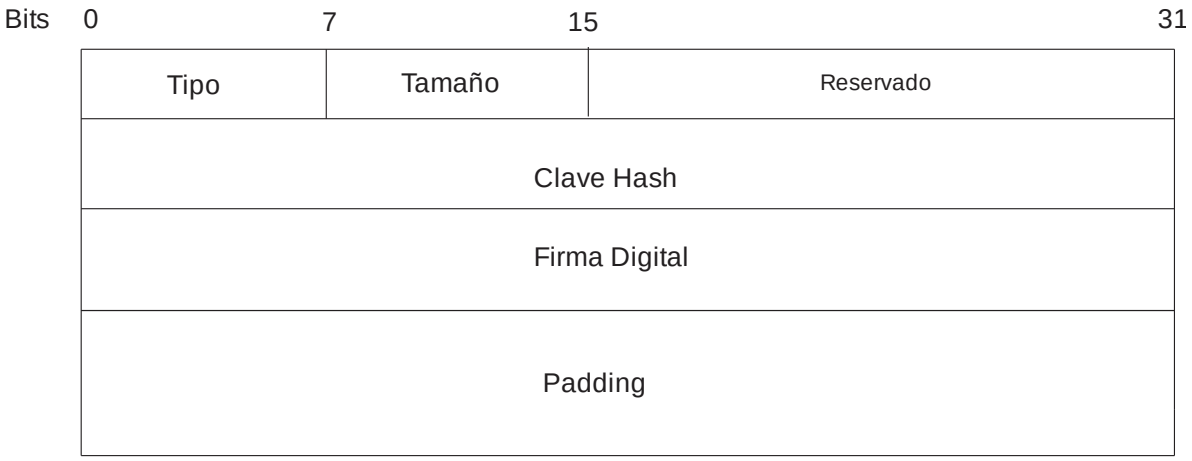


Imagen 60. Formato Opción Firma RSA.

El campo tipo siempre tiene el valor 12 en decimal. Su valor más significativo corresponde al campo *llave hash*. Este campo contiene los 128 bits más significativos del *hash* realizado a la clave pública con el algoritmo SHA1 [43] para construcción de la firma. Con este valor se puede verificar la clave pública de un emisor, la clave pública puede ser almacenada en una cache u obtenida del campo CGA. El campo firma digital contiene la firma digital del emisor.

Para el envío de los mensajes del protocolo ND, siempre debe contener la opción firma RSA desde que se utilice el protocolo SEND. En los mensajes RS no se incluye opción firma RSA cuando el campo dirección de origen es una dirección de tipo *unspecified*. La opción firma RSA siempre debe ir al final de cada mensaje ND.

Este es uno de los campos más complejos de validar durante el uso del protocolo SEND. Todos los mensajes que no contengan la opción firma RSA, con excepción del caso mencionado anteriormente, deben tratarse como mensajes inseguros. Todas las opciones que lleguen luego de la opción firma RSA deben ser ignoradas por los receptores. El *hash* de la llave pública debe realizarse a una llave pública conocida.

8.5 Opciones de sello de tiempo y *nonce.*

La opción de sello de tiempo contiene el formato que muestra la Imagen 61.

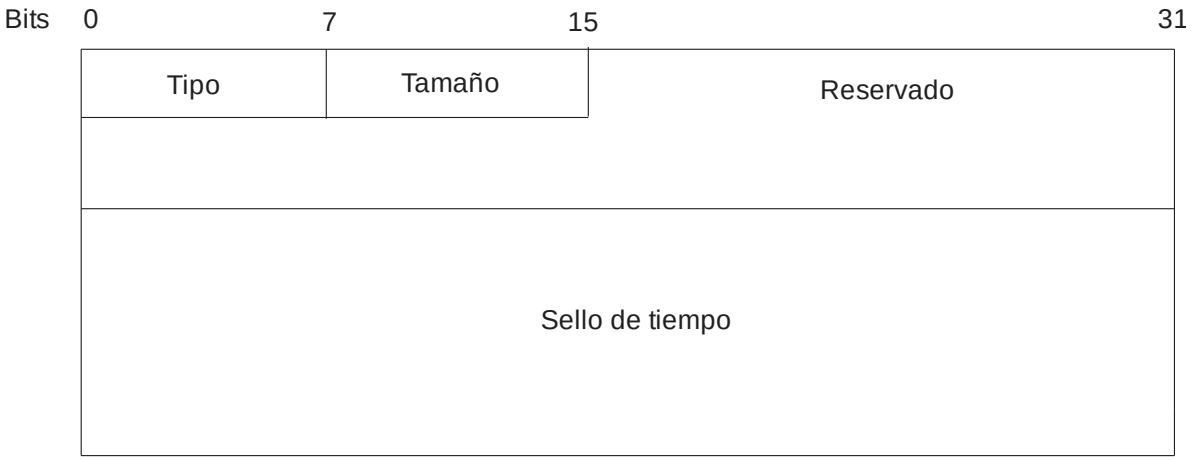


Imagen 61. Formato Opción Sello de tiempo.

El campo tipo siempre tiene el valor 13 en decimal. El campo de sello de tiempo es de 64 bits y contiene la cantidad de segundo desde enero 1 de 1970 00:00 UTC. Utiliza un formato estándar, los primeros 48 bits corresponden a un número entero y los últimos 16 bits indican $\frac{1}{64k}$ fracciones de un segundo. Este campo es compatible con la forma usual de representar el tiempo en sistemas Unix. El objetivo de este campo es que mensajes RS, NS y de redirección no sean generados sin antes haber sido solicitados.

Para el envío de los mensajes del protocolo ND, siempre debe contener la opción de sello de tiempo, este campo debe ser llenado de acuerdo a la hora de cada uno de los nodos.

La opción de *nonce* contiene el formato que muestra la Imagen 62. El objetivo de este campo es que los mensajes de NA y RA correspondan a respuestas de NS y RS respectivamente enviadas por un nodo originalmente.

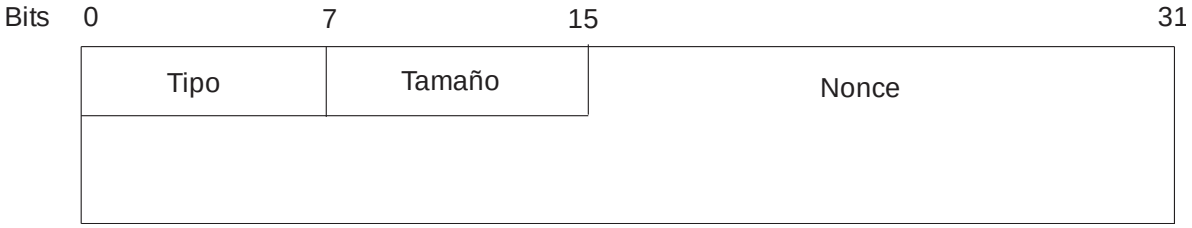


Imagen 62. Formato Opción Nonce.

El campo tipo siempre tiene el valor 14 en decimal. El campo *nonce* contiene un número aleatorio de 6 bits impuesto por el emisor del mensaje. Para el envío de los mensajes NS y RS del protocolo ND, siempre debe contener la opción CGA desde que se utilice el protocolo SEND. El nodo fuente debe almacenar el número

generado para que posteriormente él mismo pueda reconocer una respuesta al paquete enviado. Asimismo, todos los receptores que generen una respuesta deben incluir el mismo valor *nonce* en su respuesta con el fin de facilitar el reconocimiento del mensaje al emisor.

Todos los mensajes que no contengan la opción *nonce* y sello de tiempo deben tratarse como mensajes inseguros. Los mensajes que contengan la firma RSA y no el sello de tiempo o la opción *nonce* deben ser descartados por los huéspedes. El protocolo SEND brinda un mecanismo para manejar la cache del sello de tiempo lo suficientemente fuerte para prevenir ataques a esta opción. El receptor debe estar preparado para recibir las opciones de sello de tiempo y *nonce* en cualquier orden.

8.6 Certificados Digitales.

Un huésped debe tener un enrutador de confianza, para ellos es necesario que tenga un certificado digital, el cual debe estar salvaguardado por una autoridad certificadora por su nombre en inglés Certificate Authority con las siglas CA. En la CA cada nodo de la red puede consultar la veracidad del enrutador que tomará por omisión para el envío de sus mensajes. El estándar adoptado por el protocolo SEND para los certificados es X.509 [44].

Un nodo solicita el certificado digital de un enrutador mediante el mensaje CPA, la CA responde con un mensaje CPS indicándole cual es la ruta certificada o el enrutador que debe utilizar. La Imagen 63 muestra el formato del mensaje CPS.

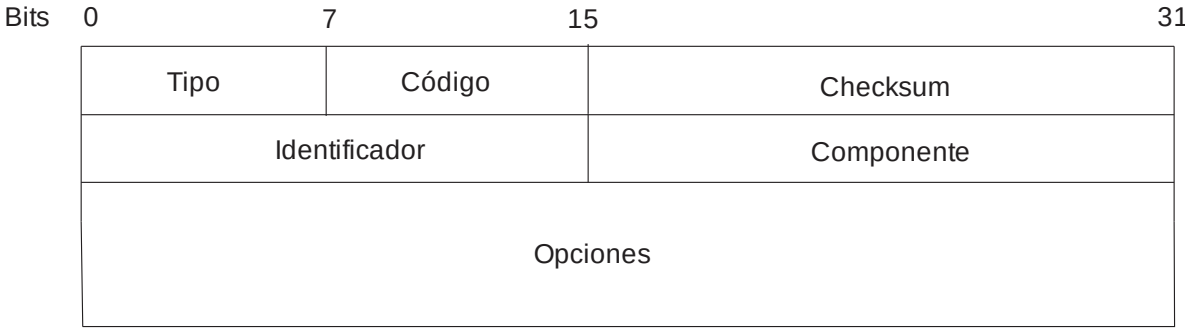


Imagen 63. Formato mensaje CPS.

El campo tipo siempre tiene el valor 148 en decimal. El campo identificador permite posteriormente realizar el emparejamiento del mensaje CPS que se envió con el CPA que se reciba. Para el mensaje CPS este valor debe ser diferente de cero (0) y generado aleatoriamente. El campo componente indica el certificado que desea obtener, en caso de completar los 16 bits con unos (1) indica que desea traer todos los certificados. Por último se tiene la opción de ancla del nodo emisor.

La Imagen 64 muestra el formato del mensaje CPA.

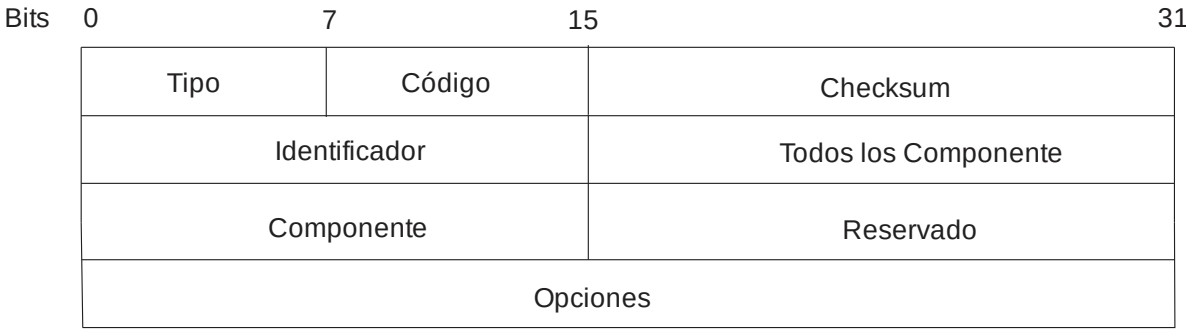


Imagen 64. Formato mensaje CPA.

El campo tipo siempre tiene el valor 149 en decimal. El campo identificador debe ser cero (0) para poder realizar el emparejamiento con el mensaje CPS. El campo “todos los componentes” permite identificar el número de certificados que lleva el mensaje. El campo componente permite identificar el cual es el certificado que transporta el mensaje. El campo certificado contiene la ruta certificada que debe tomar el nodo solicitante. Por último se tiene la opción de ancla del nodo emisor.

8.7 Controlando ataques mediante SEND.

Para prevenir los ataques de falsificación de mensajes Neighbor Solicitation y Neighbor Advertisement SEND realiza la verificación de la opción de firma RSA y revisa que la opción de dirección generada criptográficamente este presente. Con el *hash* de la clave púbica puede determinar si los mensajes sí corresponden al remitente que dice ser. Adicionalmente, en los paquetes RA y NA se verifica que contenga la copia de la opción *nonce*. En caso de encontrar la opción *nonce* en el mensaje pero el huésped no la reconozca en su cache el mensaje será descartado evitando un ataque de falsificación.

Asimismo, para neutralizar los ataques de detección de direcciones duplicadas revisa que el mensaje Neighbor Advertisement como respuesta al DAD incluya la opción de firma RSA, CGA, sello de tiempo y *nonce*. Con estas opciones se puede validar que el mensaje corresponda a quien dice el generador y no se encuentra falsificado. En caso de que alguna de las validaciones en los mensajes Neighbor Solicitation o Neighbor Advertisement no se encuentre presente el nodo descarta el mensaje.

Para controlar la falsificación a mensajes Router Solicitation y Router Advertisement se realiza el cálculo de la firma RCA mediante la clave pública. Adicionalmente debe realizar la verificación del certificado que contenga la información del prefijo de red correcta y no sea una falsificación. En caso de que alguna de las validaciones en los mensajes Router Solicitation y Router Advertisement no se encuentre presente el nodo descarta el mensaje.

Como se mencionó anteriormente, para contrarrestar un ataque masivo de gran cantidad de registros en poco tiempo, el protocolo SEND utiliza la opción sello de tiempo o en inglés *timestamp* y a su vez la opción *nonce*.

8.8 Easy-SEND.

En [45] Chiu *et al* muestran un documento para implementar el protocolo SEND una manera diferente a la presentada en el RFC oficial de SEND. En el paper indican que el protocolo IPsec puede ser utilizado para asegurar los mensajes Neighbor Discovery sólo si se tiene una dirección IP asignada. Siendo la autoconfiguración una gran apuesta del protocolo ND en IPv6 para no asignar direcciones IP mediante servicio DHCP o manualmente, el uso de IPsec no es viable.

En esta presentación de SEND se incluyen dos herramientas útiles para generar las calves tanto privado como pública y la dirección IP cifrada (CGA). A través de *cgatool* [46] y *ipexttool* [47] una implementación puede simplificar el proceso de configuración. Otra de las ventajas del uso de Easy-SEND es que no requiere modificación del núcleo del sistema operativo GNU/Linux. Adicionalmente es necesario el uso de Ip6tables [48] con el fin de filtrar los mensajes y asegurar el protocolo. En la Imagen 65 se muestra la arquitectura de Easy-SEND y es requerida la librería *libipq.*

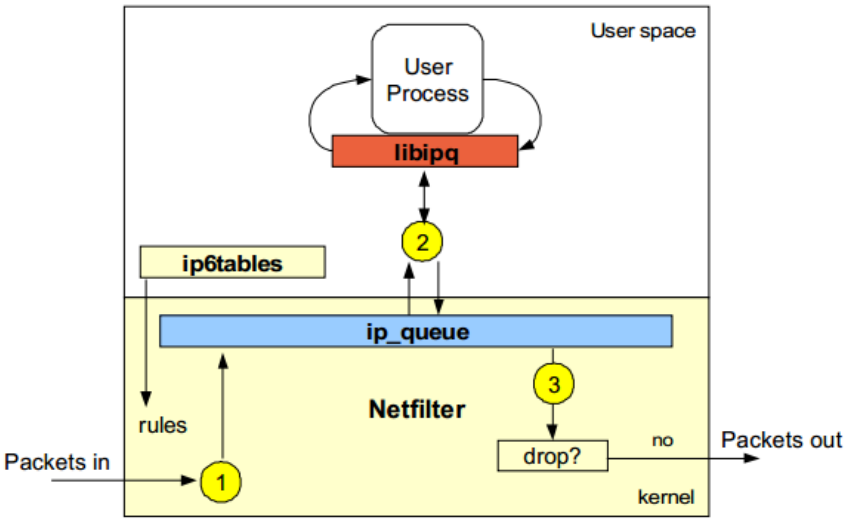


Imagen 65. Arquitectura de Easy-SEND. [45]

Para iniciar los mensajes del protocolo Neighbor Discovery se filtran en ip6tables y son enviados a la cola *ip_queue*. Los mensajes 133, 134, 135, 136 y 137 del protocolo ICMP son filtrados y la librería *libipq* permite agregarles las diferentes opciones del protocolo SEND. Los mensajes que con cumplan con las características serán descartados por el módulo.

Por último Easy-SEND utiliza su propio manejador de registros (*logs*) basado en *Apache log4j.* Con tres tipos de mensajes (INFO, DEBUG y ERROR) se puede determinar y realizar trazabilidad a los eventos ocurridos.

La herramienta Easy-SEND carece de soporte actualmente, pero aún su código se encuentra disponible para quien desee continuar con el proyecto. [45]

8.9 WinSEND.

En [41] Rafiee *et al* diseñan un mecanismo para implementar el protocolo SEND en plataformas Windows. Para esta implementación se utiliza la librería llamada *Winpcap* [49] que permite analizar el tráfico de red en esos sistemas operativos. Esta librería no sólo captura información del protocolo ICMP sino también de diferentes protocolos presentes en la red, especialmente TCP/IP. Para la implementación de WinSEND se utiliza *Winpcap* sólo para obtener los paquetes 133, 134, 135, 136 y 137 que hacen referencia al protocolo Neighbor Discovery. Durante algunas pruebas de WinSEND se determinó que la librería *Winpcap* al tener activo la adquisición de todos los paquetes TCP/IP puede presentar grandes problemas de desempeño en la máquina.

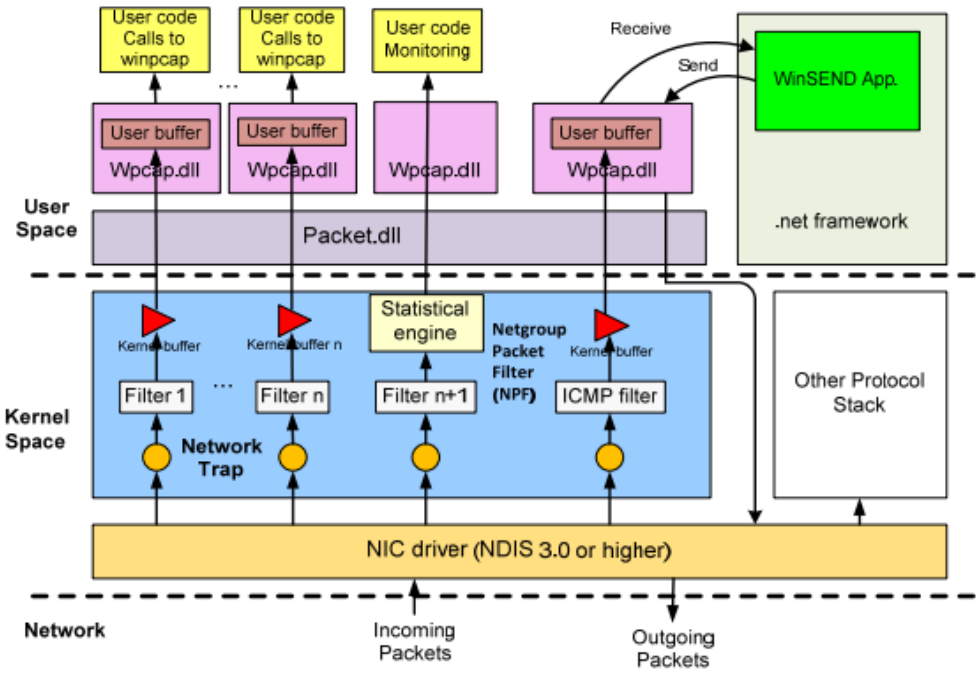


Imagen 66. Arquitectura WinSEND [41]

La Imagen 66 muestra el detalle de la arquitectura de WinSEND desde la entrada de los paquetes hasta la salida de los mismos. AL ingresar un paquete WinSEND utiliza el filtro de paquetes de red de la librería *Winpcap* para solo tomar los paquetes ICMP. Posteriormente el componente *Wpcap.dll* permite hacer el envío del paquete a núcleo de WinSEND, si el paquete cumple con las características correspondientes se procede a ser enviado por la red, en caso contrario se rechaza el paquete. Todas la parametrización para generar la clave pública y privada y la firma RSA se almacena en un archivo de tipo XML con el objetivo de mejorar el desempeño en el nodo que procesa el mensaje.

La herramienta WinSEND carece de soporte actualmente pero aún su código se encuentra disponible para quien desee continuar con el proyecto. [50]

8.10 ipv6-send-cga.

Actualmente se tiene el proyecto ipv6-send-cga [51] que permite implementar el protocolo SEND en un núcleo Linux. Para su implementación es necesario compilar el núcleo Linux aplicando SEND en IPv6. Adicionalmente en los nodos debe instalarse y configurarse el demonio SEND junto a sus archivos de configuración y módulos de parametrización. Para los certificados se hace uso de la extensión X.509 [52] mediante la herramienta *addIpExt* [53]. La Imagen 67 muestra la estructura en relación al núcleo el demonio en cada usuario.

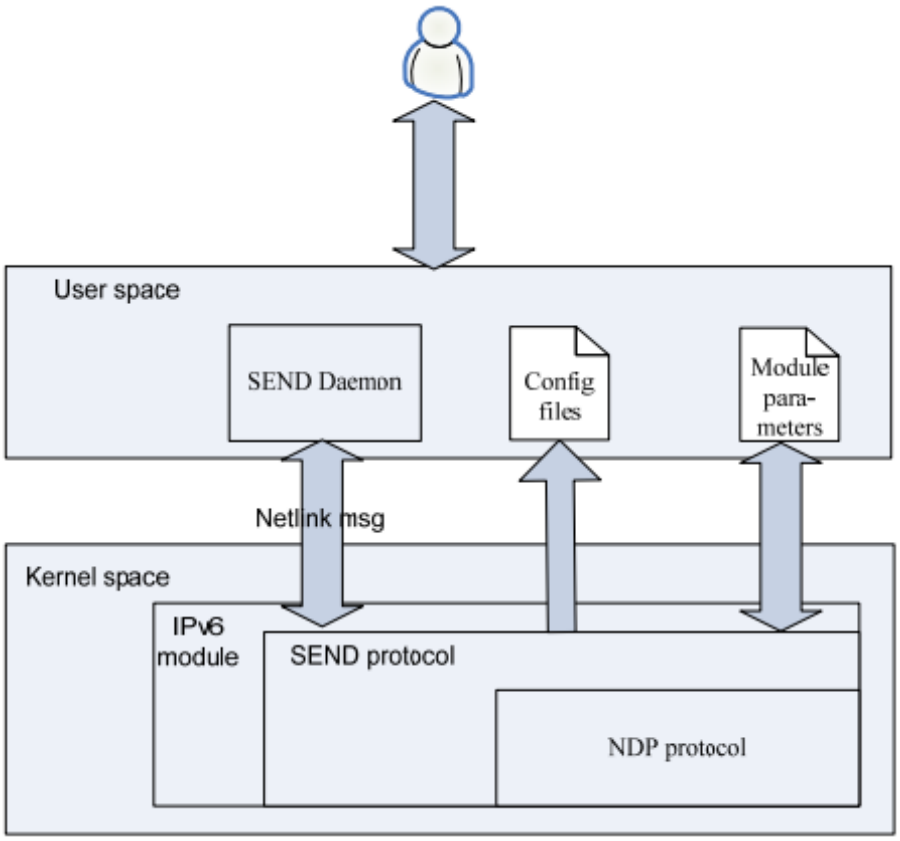


Imagen 67. Arquitectectura Nucleo/Usuario [53].

La configuración en el sistema operativo debe realizarse en dos partes. La primera parte corresponde a los parámetros que recibe el módulo. A esta primera fase le corresponde manejar el comportamiento del protocolo SEND en tiempo real. La segunda parte concierne a los archivos de configuración. En esta sección se le envían la información esencial o necesaria al protocolo SEND para su inicialización.

8.11 NDprotector.

NDprotector [54] es una aplicación libre que permite integrar el protocolo SEND y CGA en ambientes con sistema operativo GNU/Linux. Es soportado por la Agencia Nacional de Investigación por su nombre en francés L'Agence Nationale de la recherche [55]. Principalmente depende de las aplicaciones iproute2 [56] e ip6tables [48], adicionalmente de la librería libnetfiter_queue [57]. Gracias a estas aplicaciones y librería usadas en el desarrollo, es posible realizar la instalación de

NDprotector en sistemas BSD. Dichas instalaciones no hacen parte del alcance de este documento.

8.12 Funcionamiento de NDprotector.

Con el fin de minimizar el impacto de los ataques al protocolo Neighbor Discovery, cuando se recibe un mensaje del protocolo ICMP de tipo 133, 134, 135, 136 o 137 en primera instancia se realiza un filtro. El primer esquema de seguridad actúa mediante ip6tables el cual revisa el mensaje y lo redirecciona hacia el núcleo o un interfaz según sea el caso. De acuerdo la ocurrencia, si el mensaje no cumple con los estándares procede a desechar el paquete. La extracción de los paquetes se hace a través de la librería libnetfiter_queue. Adicionalmente NDprotector hace uso de scapy [58] para estudiar a fondo cada paquete y revisar si es necesario agregar nuevos campos al mismo mediante las opciones. Estos campos pueden ser la firma RSA, nonce, un certificado, entre otros. Si scapy6 considera que no es necesario agregar ninguna opción al paquete simplemente permite seguir su rumbo. La Imagen 68 muestra la arquitectura de NDprotector.

Una ventaja de NDprotector es que no requiere interacción directamente con el núcleo del sistema operativo lo que evita una colisión a nivel de núcleo. Pero a su vez se convierte en un obstáculo debido a que algunos casos es necesario emular comportamientos de la estructura del núcleo.

Como protocolo de pruebas en este documento se hará uso del NDprotector, debido a que incorpora CGA y SEND. Se realizará su implementación y posteriormente se ejecutarán los ataques anteriormente lanzados teniendo en cuenta la matriz del protocolo de pruebas que se muestra en la Tabla 9. Por último se probará la efectividad de la herramienta.

Para la instalación del NDprotector es necesario de las siguientes dependencias:

- Python en su versión 2.5 o mayor.
- OpenSSL en su versión 2.5 o mayor 1.0.
- Ip6tables.
- Librería libnetfilter_queue.

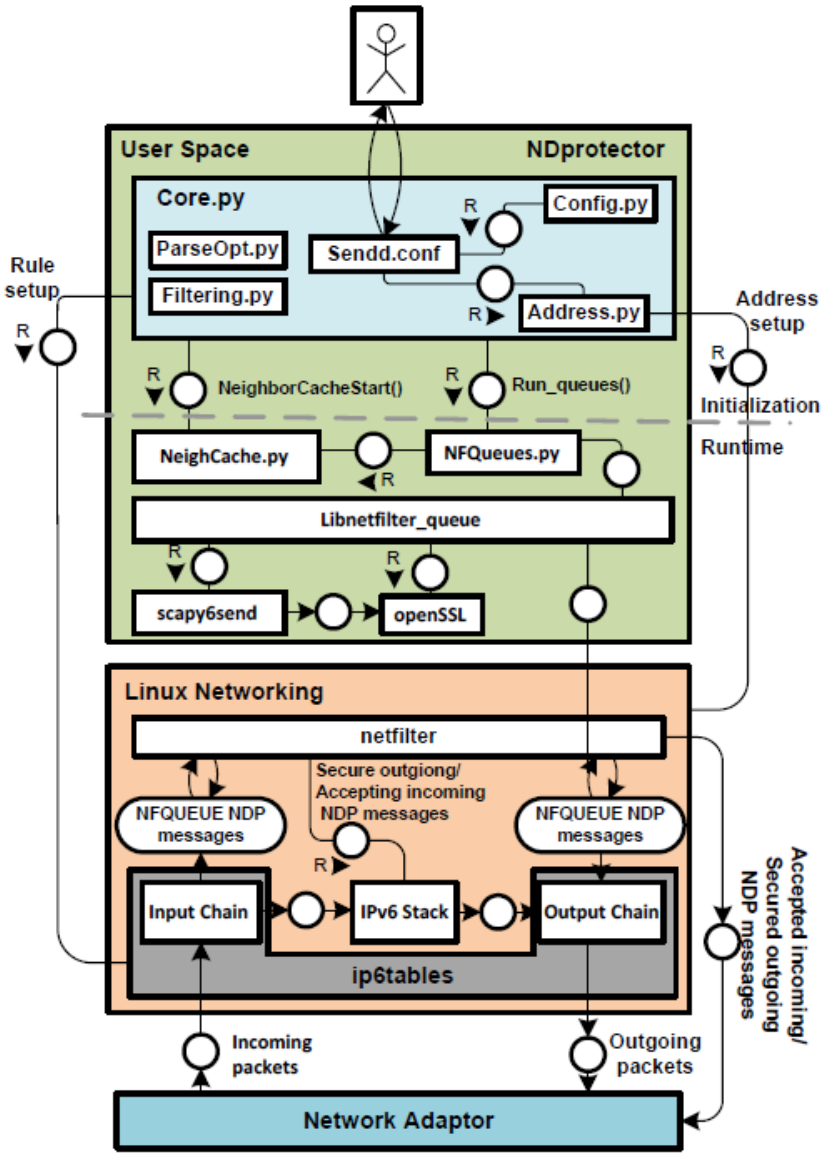


Imagen 68. Arquitectura NDprotector [59].

9 PROTOCOLO DE PRUEBAS.

La Tabla 9 muestra los ataques que se probarán durante el protocolo de pruebas con el resultado esperado.

Ataque a validar	Procedimiento	Resultado esperado
Falsificación mensajes Router Advertisement	Comparar los valores hash. Revisar la dirección IP generada criptográficamente en el campo IP origen, se deberá utilizar la clave pública del enrutador para descifrar la IP. Luego de aplicar hash al mensaje, se debe comparar con el hash que llegó en el mensaje.	Si los dos hash son iguales se procesará el mensaje, de lo contrario se rechazará el mensaje
Falsificación mensaje Neighbor Advertisement durante DAD.	Verificar de la firma RSA. Revisar que el mensaje NA enviado como respuesta a DAD se incluya la opción firma RSA. Verificar la autorización para utilizar el identificador de la interfaz	Si la firma es verificada con éxito se procesará el mensaje, de lo contrario se rechazará el mensaje
Inundación Neighbor Cache.	Verificar tiempos en el sello de tiempo. Revisar que el mensaje contenga el campo de sello de tiempo y un valor asignado, y compararlo con el lapso de tiempo configurado.	Si el sello de tiempo es mayor al lapso configurado se procesará el mensaje, de lo contrario se rechazará el mensaje
Modificación del MTU.	Comparar los valores hash. Revisar la dirección IP generada criptográficamente en el campo IP origen, se deberá utilizar la clave pública del enrutador para descifrar la IP. Luego de aplicar hash al mensaje, se debe comparar con el hash que llegó en el mensaje.	Si los dos hash son iguales se procesará el mensaje, de lo contrario se rechazará el mensaje

Falsificación del prefijo.	Comparar los valores hash. Revisar la dirección IP generada criptográficamente en el campo IP origen, se deberá utilizar la clave pública del enrutador para descifrar la IP. Luego de aplicar hash al mensaje, se debe comparar con el hash que llegó en el mensaje.	Si los dos hash son iguales se procesará el mensaje, de lo contrario se rechazará el mensaje

Tabla 9. Matriz del protocolo de pruebas.

Para proceder con el protocolo de pruebas se utilizará la herramienta NDprotector como se mencionó anteriormente.

Antes de iniciar la instalación se deben verificar o instalar las dependencias con el administrador de paquetes de Ubuntu. Para ello se ejecuta el siguiente comando:

```
apt-get install openssl iptables libnetfilter-queue-dev
nfqueue-bindings-python
```

Los resultados del comando se muestran en la Imagen 69 e Imagen 70

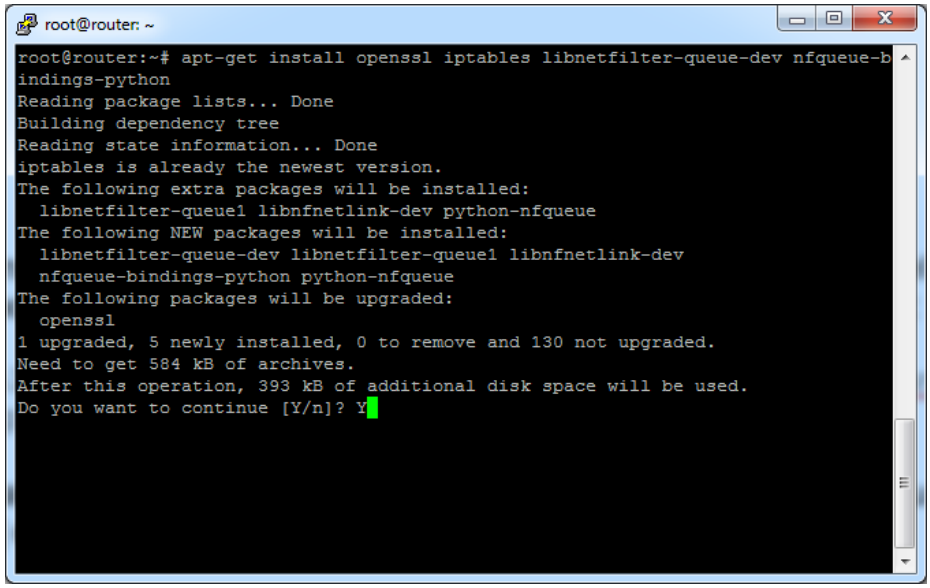


Imagen 69. Inicio de la ejecución del comando apt-get.

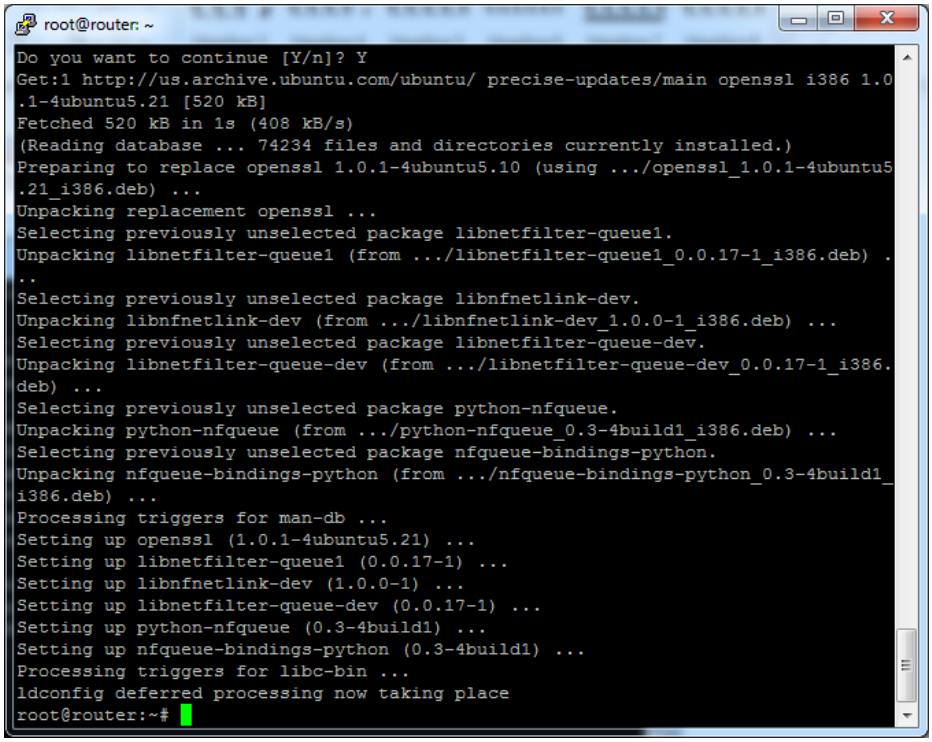


Imagen 70. Fin de la ejecución del comando apt-get.

Posteriormente, es necesario activar y verificar la opción IPv6 en el aplicativo Python la cual es fundamental para la instalación y el funcionamiento de NDprotector. Para realizar la activación, desde el terminal de Python, se ejecuta el comando `import socket,` posteriormente el comando `socket.has_ipv6=True` y por último se verifica que quedó activada la opción con el comando `socket.has_ipv6` como se muestra en la Imagen 71. El resultado debe arrojar la palabra `True` lo cual indica que efectivamente la opción IPv6 se encuentra activa en Python.

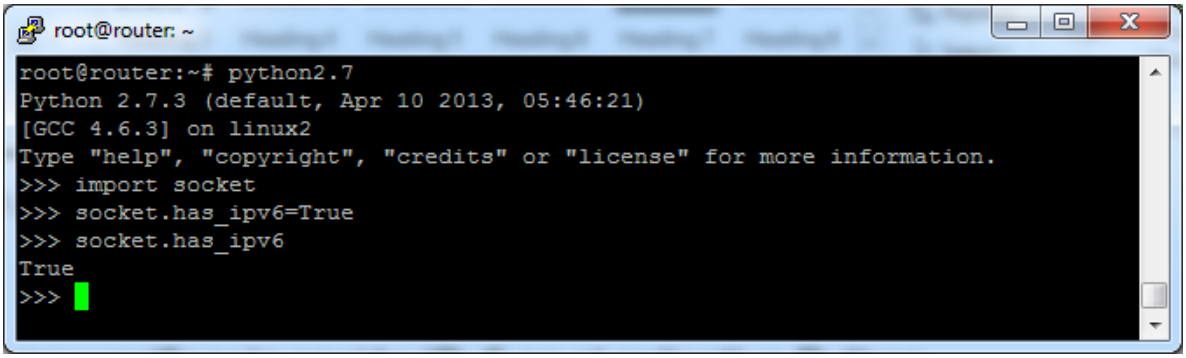


Imagen 71. Verificación Ipv6 en Python.

Para realizar la instalación del NDprotector se descarga su última versión desde su sitio web oficial [54] el paquete con nombre ndprotector_0.5_all.deb. La Imagen 72 muestra la ejecución del comando:

```
wget
http://amnesiak.org/media/software/ndprotector_0.5_all.deb
```

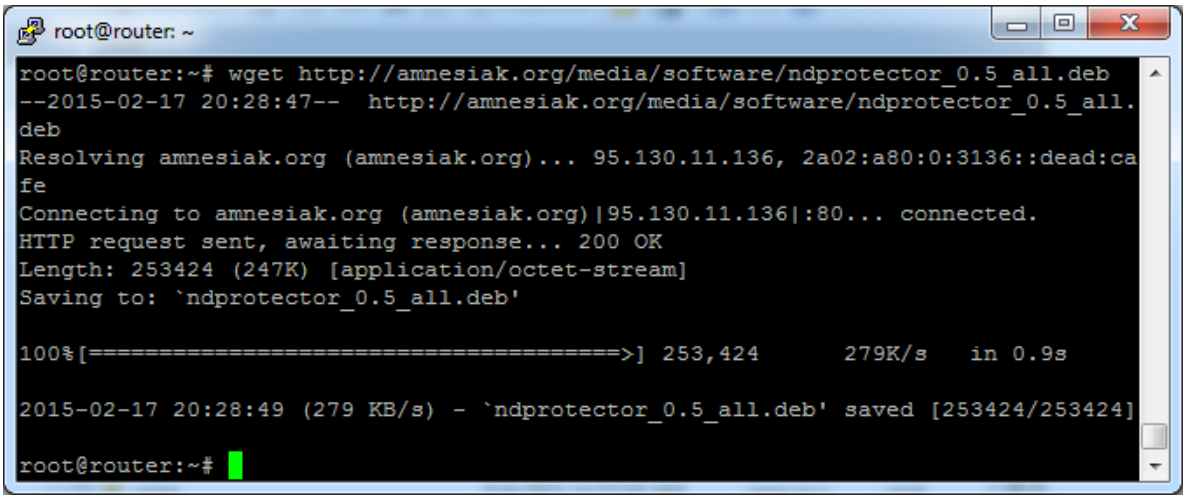


Imagen 72. Adquisición NDprotector.

Se realiza la instalación mediante el paquete .deb debido a que ya ha sido probado en sistemas Ubuntu anteriormente por los desarrolladores. Si se desea realizar la instalación en un sistema operativo diferente se puede descargar el paquete ndprotector-0.5.tar.gz que contiene el código fuente para ser compilado desde el enlace http://amnesiak.org/media/software/ndprotector-0.5.tar.gz.

Posterior a la descarga se utiliza la herramienta dpkg para dar inicio a la instalación del NDprotector. En la Imagen 73 se evidencia la instalación con el comando:

```
dpkg -i ndprotector_0.5_all.deb
```

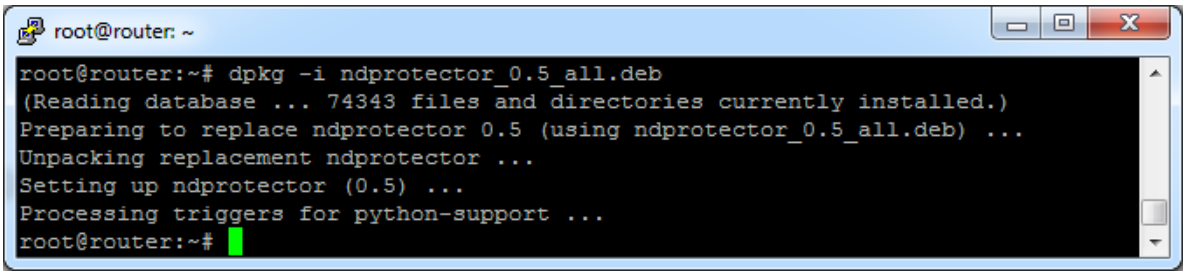


Imagen 73. Instalación del NDprotector

Se realiza la generación de la CGA para los prefijos `2001:db8:1::` y `fe80::` como se muestra en la Imagen 74 e Imagen 75 respectivamente, con los siguientes comandos:

```
python2.7 genCGA.py -p 2001:db8:1::
```

```
python2.7 genCGA.py -p fe80::
```

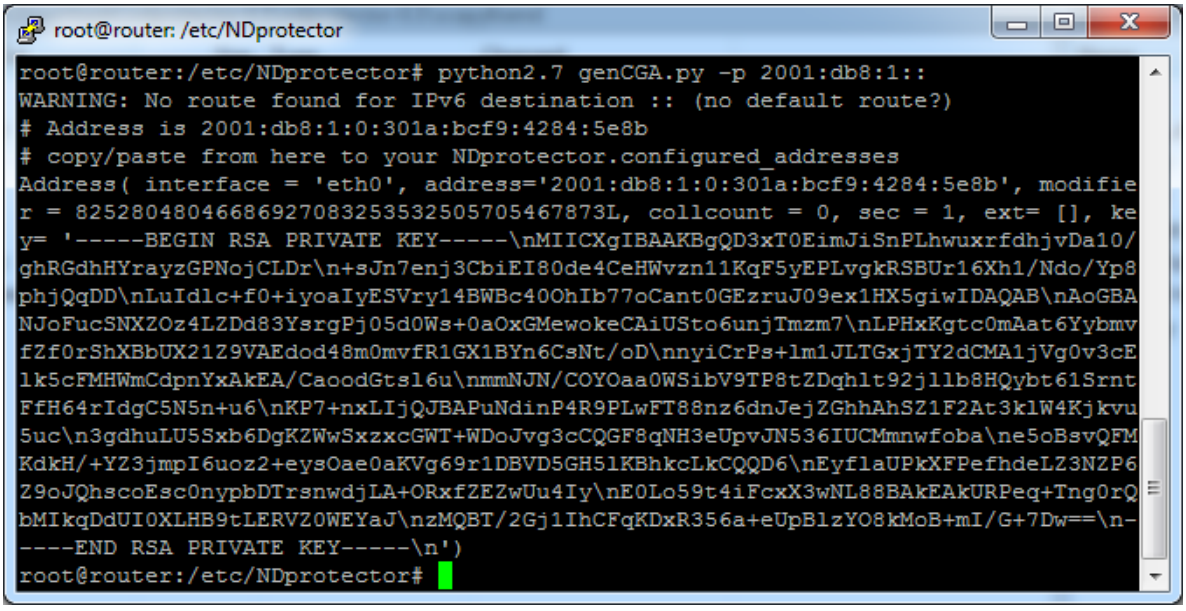


Imagen 74. Generación CGA IP Global Unicast.

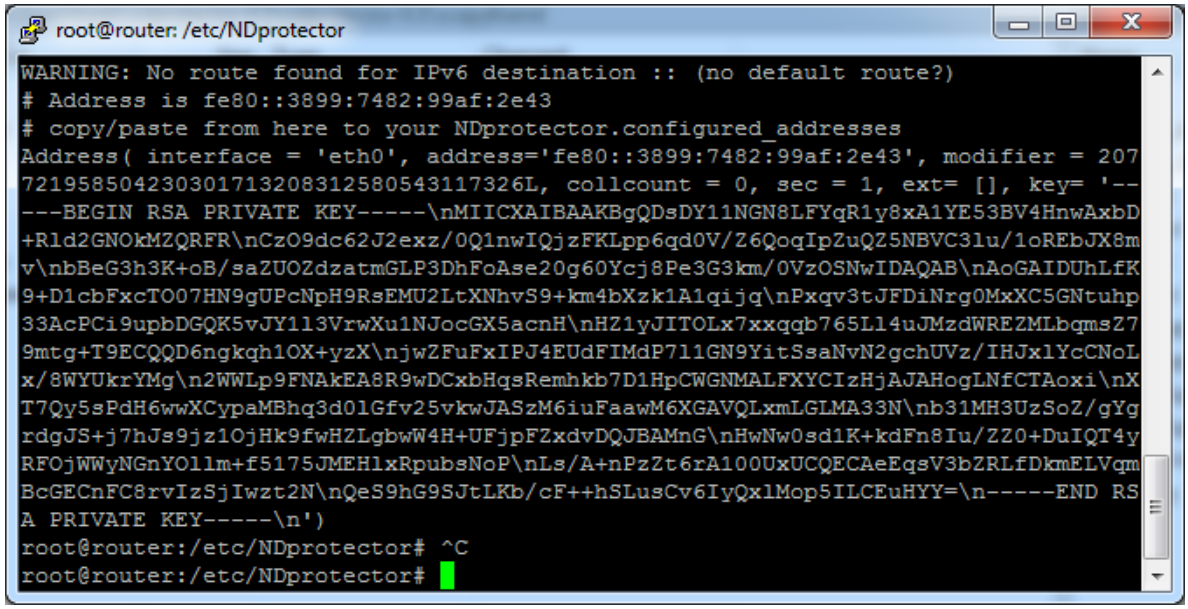


Imagen 75. Generación CGA IP Link-local.

A continuación se procede con la configuración del archivo que contiene los parámetros principales de la aplicación. Allí se registra la asignación de las claves anteriormente generadas. El archivo con la configuración se encuentra en la ruta `/etc/NDprotector/sendd.conf`. En la Imagen 76 se aprecian las modificaciones realizadas.

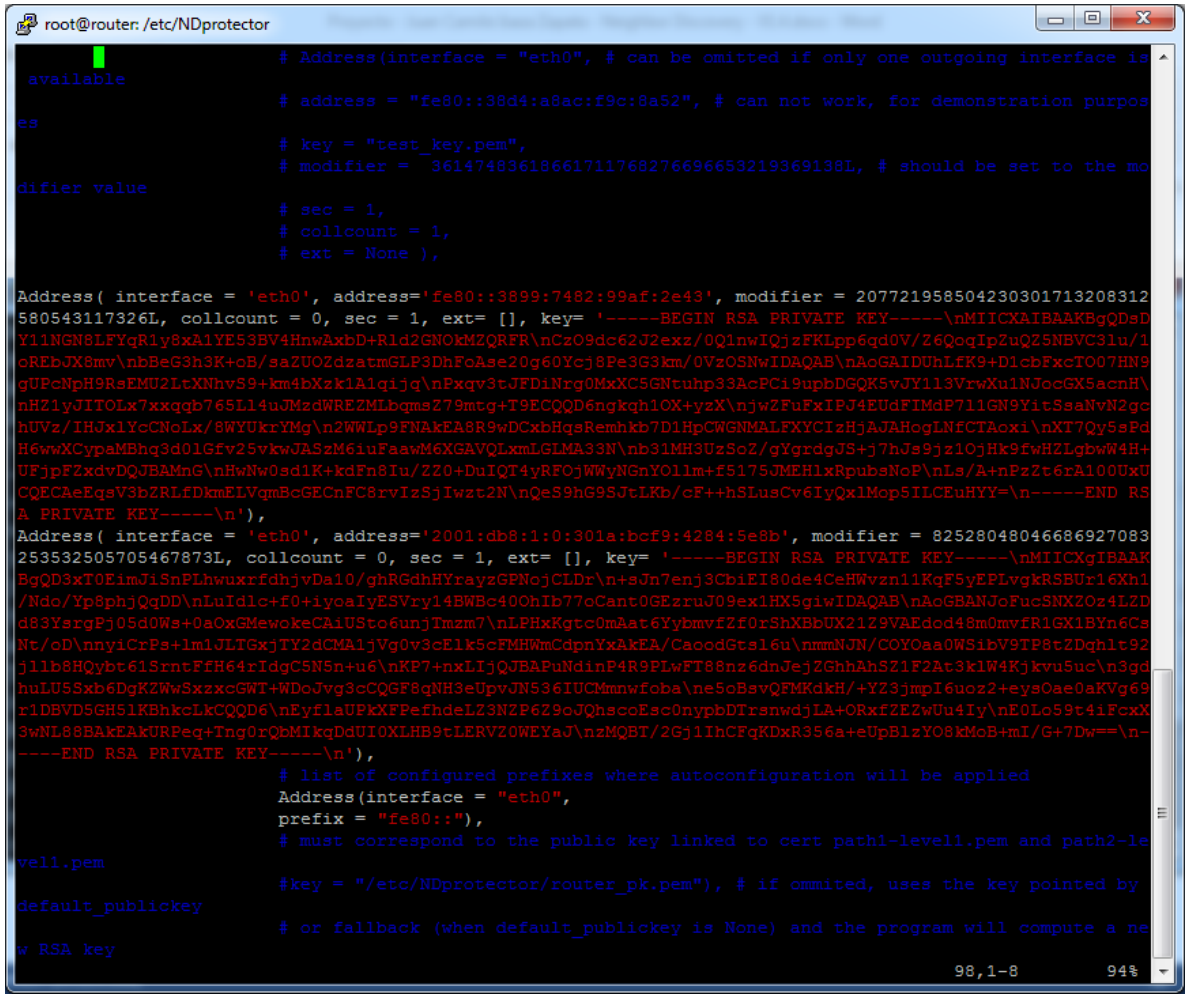


Imagen 76.Archivo sendd.conf con las claves registradas.

Por último se realiza la iniciación del servicio NDprotector con el siguiente comando: `ndprotector.py`

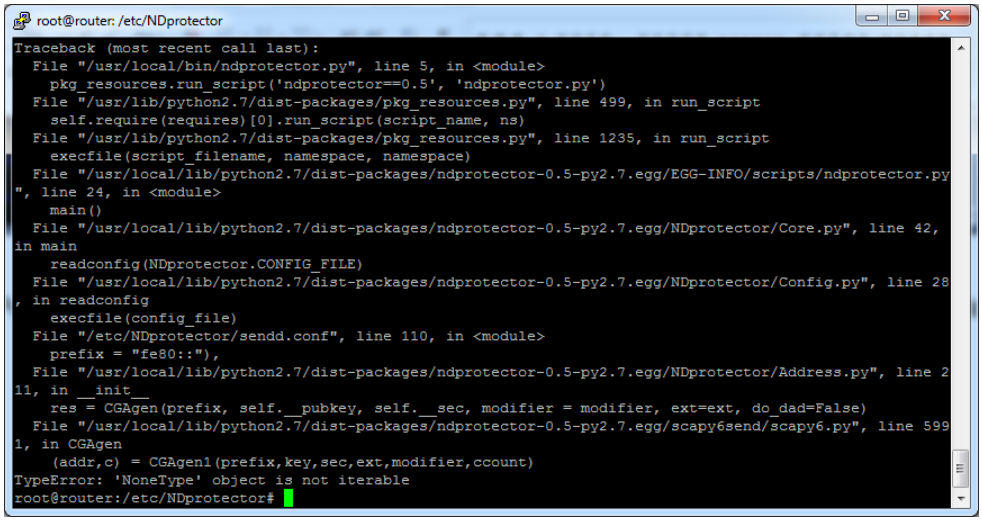
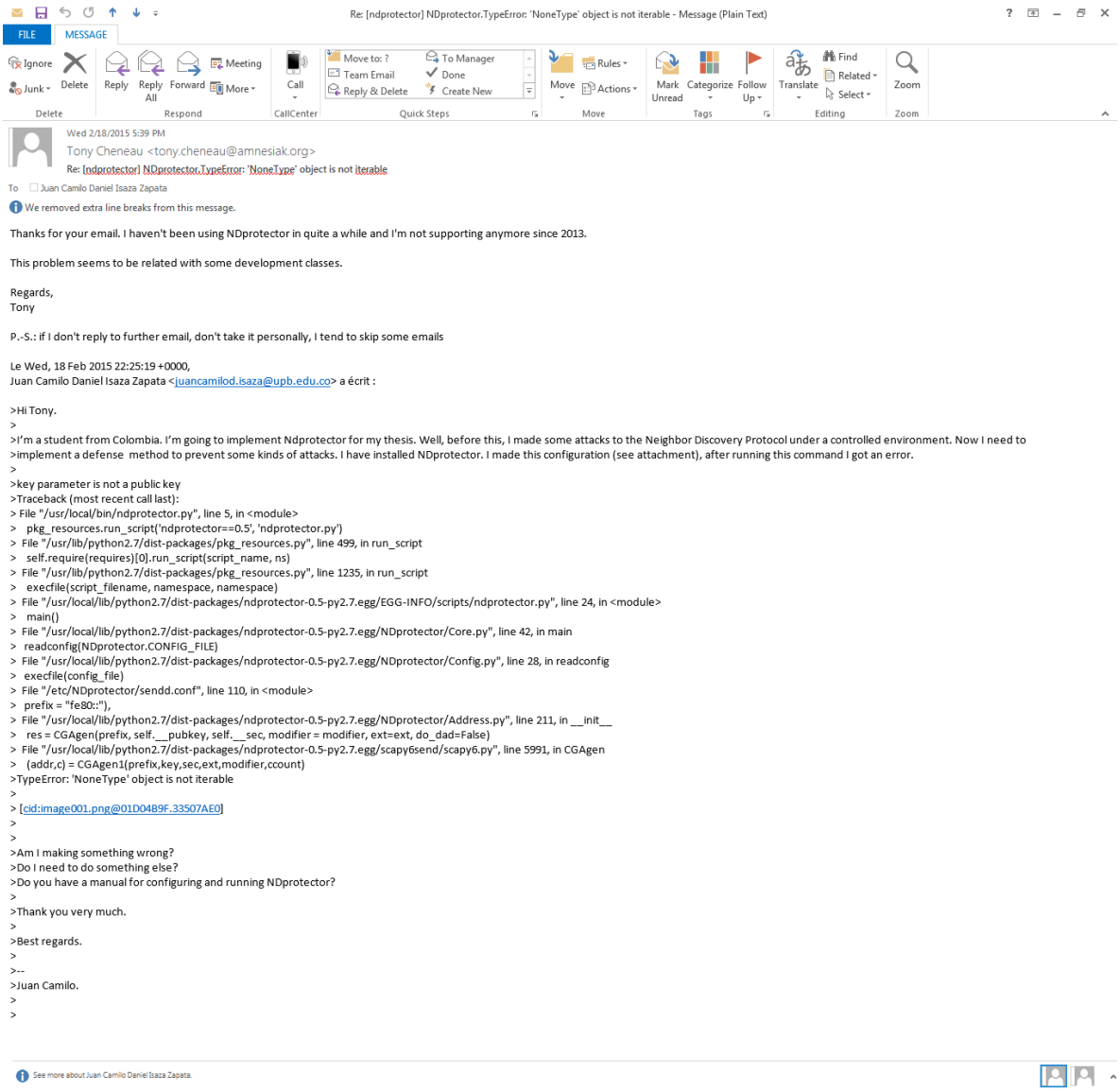


Imagen 77. Error durante la ejecución del servicio NDprotector.

Luego de su ejecución la aplicación genera un error que se visualiza en la Imagen 77.

Se procede a contactar a los desarrolladores para obtener alguna ayuda con este error pero indican que la herramienta contiene este fallo, además mencionan que desde el 2013 no le brindan soporte a este desarrollo como se muestra en la Imagen 78



Re: [ndprotector] NDprotector.TypeError: 'NoneType' object is not iterable - Message (Plain Text)

FILE MESSAGE

Ignore Junk Delete | Reply Reply All Forward Meeting More | Call | Move to: ? Team Email Reply & Delete To Manager Done Create New | Move Rules Actions | Mark Unread Categorize Follow Up | Translate Find Related Select | Zoom

Delete Respond CallCenter Quick Steps Move Tags Editing Zoom

Wed 2/18/2015 5:39 PM

Tony Cheneau <tony.cheneau@amnesiak.org>

Re: [ndprotector] NDprotector.TypeError: 'NoneType' object is not iterable

To Juan Camilo Daniel Isaza Zapata

We removed extra line breaks from this message.

Thanks for your email. I haven't been using NDprotector in quite a while and I'm not supporting anymore since 2013.

This problem seems to be related with some development classes.

Regards,
Tony

P.-S.: if I don't reply to further email, don't take it personally, I tend to skip some emails

Le Wed, 18 Feb 2015 22:25:19 +0000,
Juan Camilo Daniel Isaza Zapata <juancamilod.isaza@upb.edu.co> a écrit :

>Hi Tony.
>
>I'm a student from Colombia. I'm going to implement Ndprotector for my thesis. Well, before this, I made some attacks to the Neighbor Discovery Protocol under a controlled environment. Now I need to
>implement a defense method to prevent some kinds of attacks. I have installed NDprotector. I made this configuration (see attachment), after running this command I got an error.
>
>key parameter is not a public key
>Traceback (most recent call last):
> File "/usr/local/bin/ndprotector.py", line 5, in <module>
> pkg_resources.run_script('ndprotector==0.5', 'ndprotector.py')
> File "/usr/lib/python2.7/dist-packages/pkg_resources.py", line 499, in run_script
> self.require(requires)[0].run_script(script_name, ns)
> File "/usr/lib/python2.7/dist-packages/pkg_resources.py", line 1235, in run_script
> execfile(script_filename, namespace, namespace)
> File "/usr/local/lib/python2.7/dist-packages/ndprotector-0.5-py2.7.egg/EGG-INFO/scripts/ndprotector.py", line 24, in <module>
> main()
> File "/usr/local/lib/python2.7/dist-packages/ndprotector-0.5-py2.7.egg/NDprotector/Core.py", line 42, in main
> readconfig(NDprotector.CONFIG_FILE)
> File "/usr/local/lib/python2.7/dist-packages/ndprotector-0.5-py2.7.egg/NDprotector/Config.py", line 28, in readconfig
> execfile(config_file)
> File "/etc/NDprotector/sendd.conf", line 110, in <module>
> prefix = "fe80::"),
> File "/usr/local/lib/python2.7/dist-packages/ndprotector-0.5-py2.7.egg/NDprotector/Address.py", line 211, in __init__
> res = CGAgen(prefix, self.__pubkey, self.__sec, modifier = modifier, ext=ext, do_dad=False)
> File "/usr/local/lib/python2.7/dist-packages/ndprotector-0.5-py2.7.egg/scapy6send/scapy6.py", line 5991, in CGAgen
> (addr,c) = CGAgen1(prefix,key,sec,ext,modifier,ccount)
>TypeError: 'NoneType' object is not iterable
>
> [cid:image001.png@01D04B9F.33507AE0]
>
>
>Am I making something wrong?
>Do I need to do something else?
>Do you have a manual for configuring and running NDprotector?
>
>Thank you very much.
>
>Best regards.
>
>--
>Juan Camilo.
>
>

See more about Juan Camilo Daniel Isaza Zapata.

Imagen 78. Comunicación con Tony Cheneau de amnesiak.org.

10 OBSERVACIONES SOBRE EL PROTOCOLO SEND.

Las observaciones encontradas durante la elaboración de este trabajo de grado con respecto al protocolo SEND en IPv6 son:

Las soluciones planteadas para el protocolo SEND son de complejidad alta. El uso de criptografía asimétrica en el protocolo SEND presupone de implementación de certificados digitales lo que genera un gran impacto operativo y funcional alto. Para ello se requiere poner en funcionamiento una entidad certificadora que pueda generar los certificados, adicionalmente por cada nuevo huésped que ingresé en la red es necesario que se le genere su respectivo certificado para poder conectarse.

Actualmente la penetración de implementaciones con IPv6 aún es baja a nivel mundial [60], lo cual hace que la probabilidad de presentarse algún ataque es baja también.

Por último, no es el objetivo de esta tesis proponer una alternativa al protocolo SEND. Una posible solución sería cambiar en el protocolo SEND la criptografía basada en certificados por criptografía basada en identidad estableciendo el generador de clave privada, por su nombre en inglés *Private Key Generator* con sus siglas PKG, en los enrutadores de la red.

11 CONCLUSIONES Y TRABAJOS FUTUROS.

Los ataques realizados al protocolo *Neighbor Discovery* para IPv6 tienen un grado de complejidad medio, sin embargo las industrias que han implementado IPv6 podrían verse afectadas mediante algún tipo de ataque expuesto durante este documento.

Los métodos consultados durante esta tesis para la defensa cuentan con una dificultad alta para su implementación, adicionalmente la generación de claves públicas y privadas, las direcciones CGA, las firmas RSA y demás componentes del protocolo SEND son computacionalmente costos. Asimismo se deben agregar paquetes del protocolo ICMPv6 que generarían más tráfico en la red y mayor carga para el enrutador. El comportamiento y el desempeño de la red podrían verse afectado en caso de su implementación.

La implementación del protocolo SEND en una compañía puede ser riesgosa debido a que no cuenta con el soporte necesario para su mantenimiento y actualización. Adicionalmente su implementación no detendría los ataques a neighbor cache, dejando expuesta a la compañía ante una inminente denegación de servicio.

Como futuro trabajo está la opción de crear una protección sobre el protocolo *Neighbor Discovery* para IPv6. Una de las opciones brindadas es filtrar los mensajes e ir almacenando la información en una base de datos para posteriormente detectar posibles ataques como se expresa en [31]. El esquema es muy similar a un IDS y en varios casos podría servir de monitoreo de posibles ataques.

12 BIBLIOGRAFÍA

[1] R. Graziani, *IPv6 Fundamentals: A Straightforward Approach to Understanding IPv6*. Cisco Press, 2013, p. 456.

[2] R. M. Hinden and B. Haberman, “RFC 4193: Unique Local IPv6 Unicast Addresses.” IETF, p. 16, 2005.

[3] S. Kawamura and M. Kawashima, “RFC 5952: A Recommendation for IPv6 Address Text Representation,” no. 5952. IETF, p. 14, Aug-2010.

[4] R. Hinden and S. Deering, “RFC 4291: IP Version 6 Addressing Architecture,” *RFC 4291*. IETF, pp. 1–26, 2006.

[5] C. Huitema and B. Carpenter, “RFC 3879: Deprecating Site Local Addresses,” no. 3879. IETF, Sep-2004.

[6] J. Davies, *Understanding IPv6: Covers Windows 8 and Windows Server 2012*. Microsoft Press, 2012, p. 716.

[7] R. M. Hinden and S. E. Deering, “RFC 3587: IPv6 Global Unicast Address Format.” IETF, p. 5, 2003.

[8] M. Blanchet, *Migrating to IPv6: A Practical Guide to Implementing IPv6 in Mobile and Fixed Networks*. John Wiley & Sons, 2009, p. 450.

[9] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, “RFC 1918: Address Allocation for Private Internets,” no. 1918. IETF, Feb-1996.

[10] M.-K. Shin, Y.-G. Hong, J. Hagino, P. Savola, and E. M. Castro, “RFC 4038: Application Aspects of IPv6 Transition,” no. 4038. IETF, Mar-2005.

[11] M. Bagnulo, P. Matthews, and I. van Beijnum, “RFC 6146: Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers,” no. 6146. IETF, Apr-2011.

[12] J. Postel, “RFC 793: Transmission Control Protocol,” no. 793. IETF, p. 85, Sep-1981.

[13] J. Postel, “RFC 768: User Datagram Protocol,” no. 768. IETF, p. 3, Aug-1980.

[14] IANA, “Internet Protocol Version 6 Address Space,” 2013. [Online]. Available: http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xhtml.

[15] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, “RFC 3315: Dynamic Host Configuration Protocol for IPv6 (DHCPv6),” no. 3315. IETF, p. 101, Jul-2003.

[16] S. Thomson, T. Narten, and T. Jinmei, “RFC 4862: IPv6 Stateless Address Autoconfiguration,” no. 4862. IETF, p. 30, Sep-2007.

[17] R. Droms, “RFC 2131: Dynamic Host Configuration Protocol.” IETF, p. 45, 1997.

[18] D. L. Mills, “RFC 958: Network Time Protocol (NTP),” *IETF*, p. 14, 1985.

[19] IEEE, “Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority,” *http://standards.ieee.org/regauth/oui/tutorials/EUI64.html*, 1997. .

[20] T. Narten, W. Simpson, E. Nordmark, and H. Soliman, “RFC 4861: Neighbor discovery for IP version 6 (IPv6),” *Request for Comments*. pp. 1–97, 2007.

[21] A. Conta, S. Deering, and E. M. Gupta, “Internet control message protocol (icmpv6) for the internet protocol version 6 (ipv6) specification,” *RFC 4443*, vol. 6, pp. 1–24, 2006.

[22] D. Plummer, “RFC 826: Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware,” no. 826. IETF, Nov-1982.

[23] M. Handley, E. Rescorla, and IAB, “RFC 4732: Internet Denial-of-Service Considerations,” no. 4732. IETF, p. 38, Dec-2006.

[24] G. Bansal, N. Kumar, S. Nandi, and S. Biswas, “Detection of NDP based attacks using MLD,” in *Proceedings of the Fifth International Conference on Security of Information and Networks - SIN ’12*, 2012, pp. 163–167.

[25] J. Arkko, T. Aura, J. Kempf, V.-M. Mäntylä, P. Nikander, and M. Roe, “Securing IPv6 neighbor and router discovery,” in *Proceedings of the ACM workshop on Wireless security - WiSE ’02*, 2002, pp. 77–86.

[26] R. Bonica, W. Liu, and F. Gont, “Security Assessment of Neighbor Discovery (ND) for IPv6.” IETF, p. 62, 2013.

[27] F. Beck, T. Cholez, O. Festor, and I. Chrisment, “Monitoring the Neighbor Discovery Protocol,” *2007 Int. Multi-Conference Comput. Glob. Inf. Technol.*, 2007.

[28] P. Nikander, J. Kempf, and E. Nordmark, “RFC 3756 - IPv6 Neighbor Discovery (ND) Trust Models and Threats,” *RFC 3756, May 2004*, pp. 1–23, 2004.

[29] S. Kent and K. Seo, “RFC 4301 Security Architecture for IP,” *IETF*, pp. 1–101, 2005.

[30] J. Arkko, P. Nikander, T. Kivinen, and M. Rossi, “Manual Configuration of Security Associations for IPv6 Neighbor Discovery.” IETF, p. 27, 2003.

[31] F. A. Barbhuiya, S. Biswas, and S. Nandi, “Detection of neighbor solicitation and advertisement spoofing in IPv6 neighbor discovery protocol,” in *Proceedings of the 4th international conference on Security of information and networks - SIN ’11*, 2011, p. 111.

[32] E. Rescorla and B. Korver, “Guidelines for Writing RFC Text on Security Considerations,” no. 3552. IETF, p. 44, Jul-2003.

[33] J. Arkko, J. Kempf, B. Zill, and P. Nikander, “RFC 3971: SEcure Neighbor Discovery (SEND),” no. 3971. IETF, p. 56, Mar-2005.

[34] T. Aura, “RFC 3972: Cryptographically Generated Addresses (CGA),” no. 3972. IETF, p. 22, Mar-2005.

[35] J. Jonsson and B. Kaliski, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1,” no. 3447. IETF, p. 72, Feb-2003.

[36] THC.org, “THC-IPV6,” 2014. [Online]. Available: https://www.thc.org/thc-ipv6/.

[37] THC.org, “fake_router6.c,” 2014. [Online]. Available: https://github.com/mmoya/pkg-thc-ipv6/blob/master/fake_router6.c.

[38] THC.org, “detect-new-ip6,” 2014. [Online]. Available: https://github.com/mmoya/pkg-thc-ipv6/blob/master/detect-new-ip6.c.

[39] THC.org, “dos-new-ip6,” 2014. [Online]. Available: https://github.com/mmoya/pkg-thc-ipv6/blob/master/dos-new-ip6.c.

[40] P. Yee, “RFC 6818: Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.” p. 8, 2013.

[41] H. Rafiee, A. Alsa’deh, and C. Meinel, “WinSEND,” in *Proceedings of the 4th international conference on Security of information and networks - SIN ’11*, 2011, p. 243.

[42] B. Jonsson, J. Kaliski, “RFC 3587: Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1.” p. 72, 2003.

[43] D. E. E. 3rd and P. E. Jones, “RFC 3174: US Secure Hash Algorithm 1 (SHA1).” p. 22, 2001.

[44] D. Cooper, “RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.” p. 151, 2008.

[45] S. Chiu and E. Gamess, “Easy-SEND: A Didactic Implementation of the Secure Neighbor Discovery Protocol for IPv6,” in *WCECS 2009: WORLD CONGRESS ON ENGINEERING AND COMPUTER SCIENCE, VOLS I AND II*, 2009, pp. 260–265.

[46] I. Genomics, “cgatools documentation,” 2013. [Online]. Available: http://cgatools.sourceforge.net/. [Accessed: 12-Feb-2015].

[47] FreeBSD, “Ipexttool,” 2013. [Online]. Available: http://people.freebsd.org/~anchie/send-net-mgmt/send03_ports/ports/net-mgmt/send/work/send_0.3/docs/UserGuide.pdf. [Accessed: 12-Feb-2015].

[48] J. Davies, E. Mohacsi, “RFC 4980: Recommendations for Filtering ICMPv6 Messages in Firewalls.” p. 38, 2007.

[49] Riverbed Technology, “WinPcap - Home.” [Online]. Available: http://www.winpcap.org/. [Accessed: 13-Feb-2015].

[50] tclsoap, “WinSEND,” 2004. [Online]. Available: http://sourceforge.net/projects/tclsoap/files/. [Accessed: 25-Feb-2015].

[51] Beijing University of Post and Telecommunications, “ipv6-send-cga - an implementation of SEND protocol in LINUX kernel,” 2013. [Online]. Available: https://code.google.com/p/ipv6-send-cga/. [Accessed: 13-Feb-2015].

[52] K. Lynn, C. Kent, S. Seo, “RFC 3779: X.509 Extensions for IP Addresses and AS Identifiers.” 2004.

[53] B. U. of P. and Telecommunications, “ipv6-send-cga,” 2013. [Online]. Available: https://void.gr/kargig/ipv6/IPV6 SEND UserGuide.pdf. [Accessed: 13-Feb-2015].

[54] T. Cheneau, “NDprotector: an implementation of CGA & SEND for GNU/Linux based on Scapy6,” 2012. [Online]. Available: http://amnesiak.org/NDprotector/. [Accessed: 16-Feb-2015].

[55] ANR, “Le financement sur projets au service de la recherche | ANR - Agence Nationale de la Recherche,” 2014. [Online]. Available: http://www.agence-nationale-recherche.fr/. [Accessed: 16-Feb-2015].

[56] A. Kuznetsov, “iproute2 | The Linux Foundation,” 2009. [Online]. Available: http://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2. [Accessed: 16-Feb-2015].

[57] H. Welte, “‘libnetfilter_queue’ project,” 2009. [Online]. Available: http://www.netfilter.org/projects/libnetfilter_queue/. [Accessed: 16-Feb-2015].

[58] Secdev-foundation.org, “SecDev / Scapy,” 2008. [Online]. Available: http://bb.secdev.org/scapy/wiki/doc/index. [Accessed: 16-Feb-2015].

[59] A. Alsa’deh, F. Cheng, and C. Meinel, “CS-CGA: Compact and more secure CGA,” in *ICON 2011 - 17th IEEE International Conference on Networks*, 2011, pp. 299–304.

[60] J. Villa, “IPv6 - LACNIC,” 2015. [Online]. Available: http://portalipv6.lacnic.net/lento-pero-seguro-el-despliegue-de-ipv6-en-al-y-caribe/. [Accessed: 13-Mar-2015].