# Seguridad IPv6

Fernando Gont


Track "Protocolo IPv6", WALC 2011

Guayaquil, Ecuador. Octubre 10-14, 2011

# Acerca de...

- Consultor en el área ingeniería de Internet y seguridad informática para SI6 Networks (http://www.si6networks.com)
- Miembro del Centro de Estudios de Informatica (CEDI) de UTN/FRH
- Participo activamente en la Internet Engineering Task Force (IETF)
- Moderador del Foro de Seguridad de LACNIC, y chair del evento anual LACSEC
- Mas información disponible en: http://www.gont.com.ar

# Motivación de esta presentación

- Se han creado muchos mitos alrededor de la seguridad IPv6:
  - La seguridad fue considerada durante el desarrollo del mismo
  - El paradigma de seguridad cambiará de network-centric a host-centric
  - Se incrementará el uso de IPv6
  - etc.
- Estos mitos han llevado a concepciones erróneas sobre IPv6, con un consecuente impacto negativo en la seguridad de las redes emergentes (o ya existentes)
- A nivel conceptual, este tutorial intentará:
  - Separar “mito” de “realidad”, y ofrecer una visión realista sobre el tema
  - Influenciar la manera en al cual piensas sobre “seguridad IPv6”
- A nivel práctico, este tutorial intentará:
  - Mostrarte herramientas de utilidad en el área de seguridad IPv6
  - Analizar y explotar algunas vulnerabilidades concretas, y discutir posibles contramedidas

# Agenda

- Breve comparación de IPv6/IPv4
- Discusión de aspectos de seguridad de IPv6 (básicos)
- Seguridad de los mecanismos de transición/co-existencia
- Implicancias de seguridad de IPv6 en redes IPv4
- Áreas en las que se necesita progreso
- Conclusiones
- Preguntas y respuestas

# Consideraciones generales sobre seguridad IPv6

# Aspectos interesantes sobre seguridad IPv6

- Se cuenta con mucha menos experiencia que con IPv4
- Las implementaciones de IPv6 son menos maduras que las de IPv4
- Los productos de seguridad (firewalls, NIDS, etc.) tienen menos soporte para IPv4 que para IPv6
- La complejidad de las redes se incrementará durante el periodo de transición/co-existencia:
  - Dos protocolos de red (IPv4 e IPv6)
  - Mayor uso de NATs
  - Mayor uso de túneles
  - Uso de otras tecnologías de transición
- Pocos recursos humanos bien capacitados

…y así y todo IPv6 será en muchos casos la única opción disponible para continuar en el negocio de Internet

# Comparación entre IPv6 e IPv4

(qué cambia, y qué no)

# Breve comparación de IPv4 e IPv6

- IPv4 e IPv6 son muy similares en términos de *funcionalidad* (no así de *mecanismos*)

| | IPv4 | IPv6 |
|---|---|---|
| Direccionamiento | 32 bits | 128 bits |
| Resolución de direcciones | ARP | ICMPv6 NS/NA (+ MLD) |
| Auto-configuración | DHCP & ICMP RS/RA | ICMPv6 RS/RA & DHCPv6 (opcional) (+ MLD) |
| Aislamiento de fallas | ICMP | ICMPv6 |
| Soporte de IPsec | Opcional | Recomendado (no mandatorio) |
| Fragmentación | Tanto en hosts como routers | Sólo en hosts |

# Implicancias de Seguridad de IPv6

# Direccionamiento

# Direccionamiento

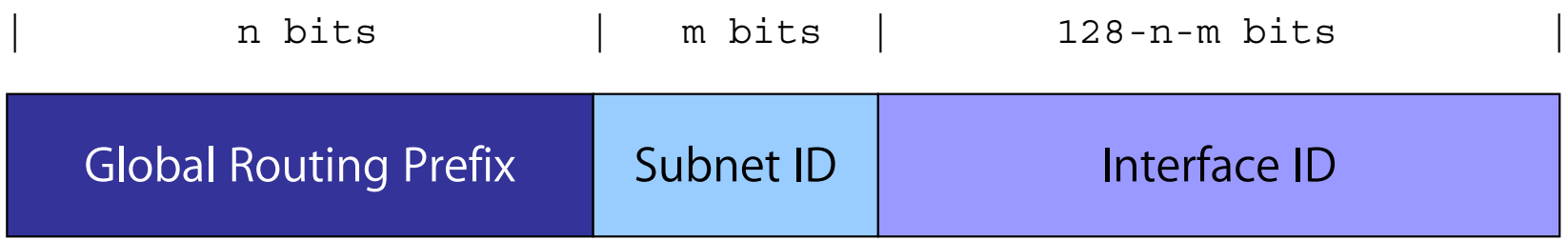
## Implicancias sobre “escaneo”

# Breve reseña

- El principal motivador de IPv6 es su mayor espacio de direcciones
- IPv6 utiliza direcciones de 128 bits
- De manera similar a IPv4,
  - Las direcciones se “agregan” en prefijos para su ruteo
  - Se definen distintos tipos de direcciones (unicast, anycast, y multicast)
  - Se definen distintos alcances para las direcciones (link-local, global, etc.)
- Lo usual es que en un determinado instante, un nodo utilice varias direcciones, de distintos tipos y alcances

# Breve reseña (II)

- Formato de las direcciones IPv6 unicast globales:




- El Interface ID es típicamente de 64 bits
- Las direcciones unicast globales pueden "generarse" con distintos criterios:
  - Formato EUI-64 modificado (embebiendo direcciones de capa de enlace)
  - Direcciones "temporales" (o sus variantes)
  - Patrones predeterminados por el administrador (por ej., PREFIJO::1)
  - De acuerdo a lo especificado por una tecnología de transición/co-existencia

# Implicancias en ***brute-force*** scanning

Mito: *"Es imposible realizar un ataque de escaneo de direcciones en IPv6, ya que el espacio de direcciones es muy grande. Hacerlo llevaría una eternidad!"*

- Esto asume que las direcciones de los hosts están uniformemente distribuídas en la subred
- Sin embargo, estudios realizados (*) indican que este no es necesariamente el caso: las direcciones suelen generarse con patrones predeterminados. Básicamente,
  - SLAAC (Interface-ID derivado de la MAC address)
  - Basadas en IPv4 (por ej., 2001:db8::192.168.10.1)
  - "Low byte" (por ej., 2001:db8::1, 2001:db8::2, etc.)
  - Privacy Addresses (Interface-ID aleatorio)
  - "Wordy" (por ej., 2001:db8::dead:beef)
  - Relacionadas con tecnologías de transición (por ej., Teredo)

(*) Malone, D. 2008. *Observations of IPv6 Addresses*. Passive and Active Measurement Conference (PAM 2008, LNCS 4979), 29–30 April 2008.

# Algunos datos reales….

- [Malone, 2008] (*) midió las direcciones asignadas a clientes y routers:

Clientes

| Tipo de dirección | Porcentaje |
|---|---|
| SLAAC | 50% |
| Basada en IPv4 | 20% |
| Teredo | 10% |
| Low-byte | 8% |
| Privacy | 6% |
| wordy | <1% |
| Otras | <1% |

Routers

| Tipo de dirección | Porcentaje |
|---|---|
| Low-byte | 70% |
| Basada en IPv4 | 5% |
| SLAAC | 1% |
| Wordy | <1% |
| Privacy | <1% |
| Teredo | <1% |
| Otras | <1% |

(*) Malone, D. 2008. *Observations of IPv6 Addresses*. Passive and Active Measurement Conference (PAM 2008, LNCS 4979), 29–30 April 2008.

# Algunas recomendaciones

- Para servidores, la política de generación de direcciones es generalmente irrelevante
- Para clientes, en escenarios generales es deseable el uso de “direcciones temporales” (extensiones de privacidad)
- Para nodos que no necesiten ser “alcanzables”, es deseable la asignación de direcciones “no previsibles”
- En todos los casos anteriores, se debe considerar si es deseable implementar una política de filtrado de paquetes

# Conclusiones

- IPv6 incrementa la dificultad de realizar "brute force scanning"
- Sin embargo, es esperable que las herramientas utilizadas por atacantes evolucionen, permitiendo reducir el espacio de búsqueda.
- Asimismo, es probable que se exploren otros métodos de scanning:
  - Direcciones expuestas por protocolos de aplicación (P2P, e-mail, etc.)
  - Direcciones multicast (por ej., all-nodes multicast address)
  - Protocolos de descubrimiento de vecinos (por ej., mDNS)
- El port-scanning (en contraposición con el host scanning), permanece igual

# Ejemplo de scanning por multicast

- All-nodes link-local multicast address:
  - ping6 ff02::1%eth0
- All-routers link-local multicast address:
  - ping6 ff02::2%em0
- etc

# Ejemplo de leak en protocolo de aplicación

```
X-ClientAddr: 46.21.160.232
Received: from srv01.bbserve.nl (srv01.bbserve.nl [46.21.160.232])
        by venus.xmundo.net (8.13.8/8.13.8) with ESMTP id p93Ar0E4003196
        for <fernando@gont.com.ar>; Mon, 3 Oct 2011 07:53:01 -0300
Received: from [2001:5c0:1000:a::943]
        by srv01.bbserve.nl with esmtpsa (TLSv1:AES256-SHA:256)
        (Exim 4.76)
        (envelope-from <fgont@si6networks.com>)
        id 1RAg8k-0000Qf-Hu; Mon, 03 Oct 2011 12:52:55 +0200
Message-ID: <4E8993FC.30600@si6networks.com>
Date: Mon, 03 Oct 2011 07:52:44 -0300
From: Fernando Gont <fgont@si6networks.com>
Organization: SI6 Networks
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.23)
Gecko/20110922 Thunderbird/3.1.15
MIME-Version: 1.0
To: Fernando Gont <fernando@gont.com.ar>
Subject: Prueba
```

# IPv6 port-scanning

- Se puede realizar mediante nmap, del siguiente modo:

```
# nmap -6 -p1-10000 -n 2000:db8::1
80/tcp open http
135/tcp open msrpc
445/tcp open microsoft-ds
554/tcp open rtsp
1025/tcp open NFS-or-IIS
1026/tcp open LSA-or-nterm
1027/tcp open IIS
1030/tcp open iad1
1032/tcp open iad3
1034/tcp open unknown
1035/tcp open unknown
1036/tcp open unknown
1755/tcp open wms
9464/tcp open unknown
```

# Conectividad “extremo a extremo” (“end-to-end”)

# Breve reseña

- Dado que IPv6 posee un gran espacio de direcciones, se espera que cada dispositivo conectado a la red cuente con una dirección IPv6 global unica.
- Es usual asumir que esto “devolverá” a la Internet el principio conocido como “end-to-end”:
  - La comunicación entre sistemas es transparente (por ej., los nodos intermedios no modifican los paquetes)
  - Cualquier sistema de la red es capaz de establecer una comunicación con cualquier otro sistema de la red
  - Usualmente se argumenta que esto permitiría la “innovación” en la red

# Consideraciones varias

Mito: "*IPv6 devolverá a Internet la conectividad extremo a extremo*"

- El hecho de que cada sistema posea una dirección global única no garantiza la posibilidad de comunicación "extremo a extremo"
- La realidad es que la mayoría de las redes de hoy en día no tienen como fin la innovación, sino que son un medio para trabajar o recrearse
- Y los servicios esperados por los usuarios son aquellos mismos que hoy se brindan en IPv4 sin conectividad "end-to-end" (web, email, redes sociales, etc.)
- En conclusión,
  - La conectividad "extremo a extremo" no es necesariamente una propiedad "deseable" en una red de producción
  - La subred IPv6 típica (como ser una red hogareña) esté protegida por un firewall stateful que solo permita el tráfico "de retorno" (aquél en respuesta a comunicaciones iniciadas desde el interiore de la red)

# Resolución de Direcciones en IPv6

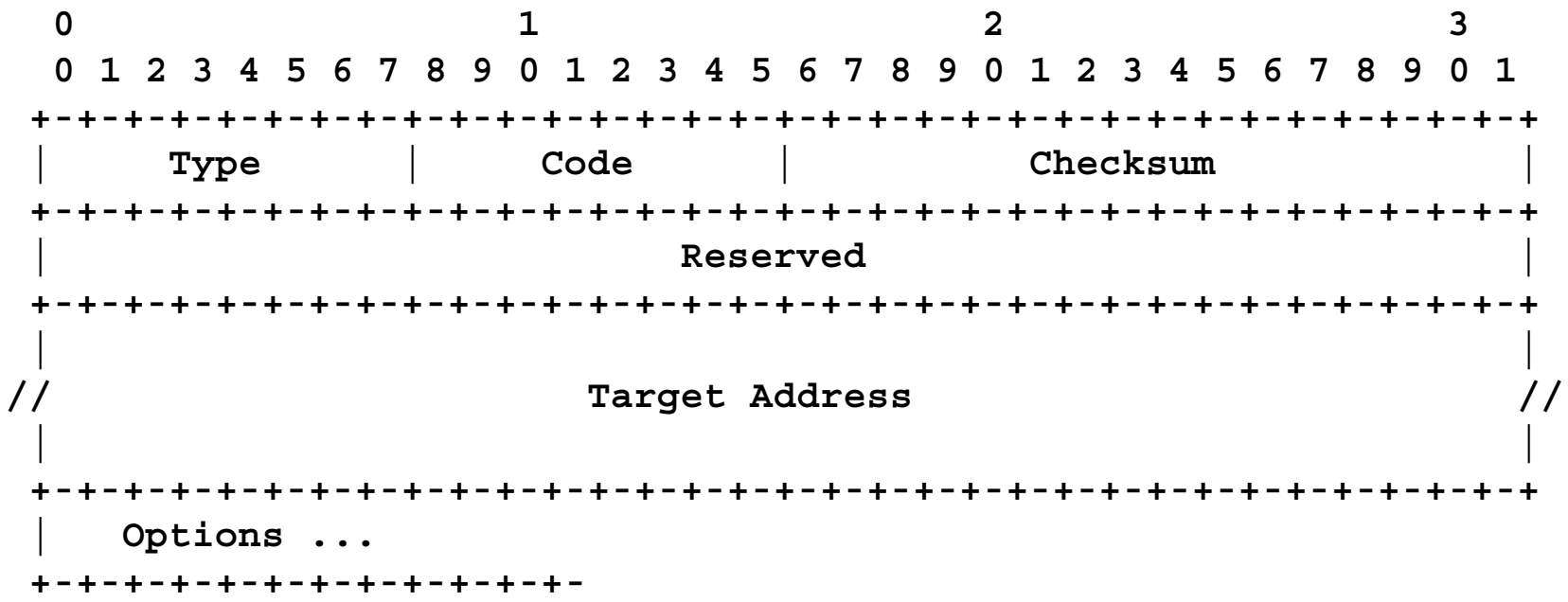
## Operación

# Operación de Neighbor Discovery para IPv6

- Utiliza mensajes ICMPv6 Neighbor Solicitation y Neighbor Advertisement
- El proceso es simple:
  1. El Host 1 envía un NS: "*Quién tiene la dirección IPv6 2001:db8::1?*"
  2. El Host 2 responde con una NA: "*Yo tengo la dirección 2001:db8::1, y la MAC address correspondiente es 06:09:12:cf:db:55*".
  3. El Host 1 "cachea" la informacíón recibida en el "Neighbor Cache" durante un tiempo (esto es una optimización similar al ARP cache)
  4. El Host 1 puede ahora enviarle paquetes al Host 2

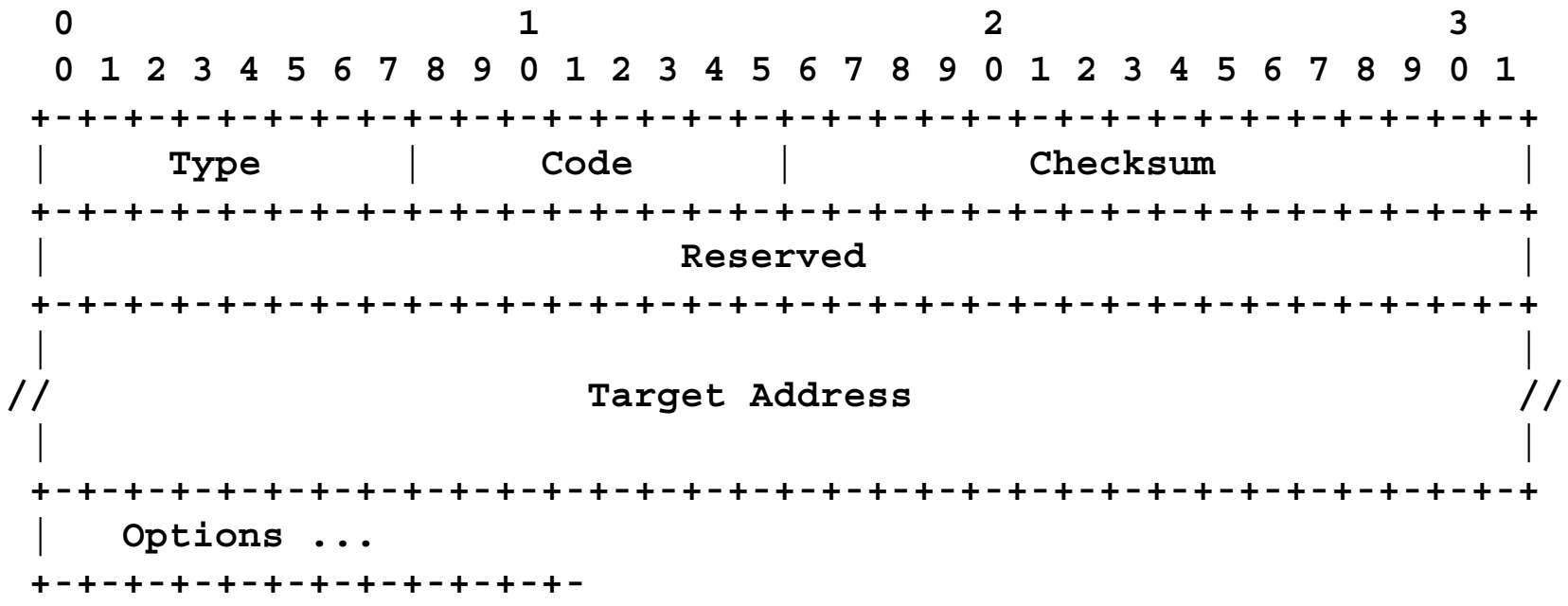
# Mensajes Neighbor Solicitation

- Son mensajes ICMPv6 de Tipo 135, Código 0
- Utilizados para solicitar la dirección de capa de enlace correspondiente a una dirección IPv6
- La única opción permitida en ellos es la “Source Link-layer address”



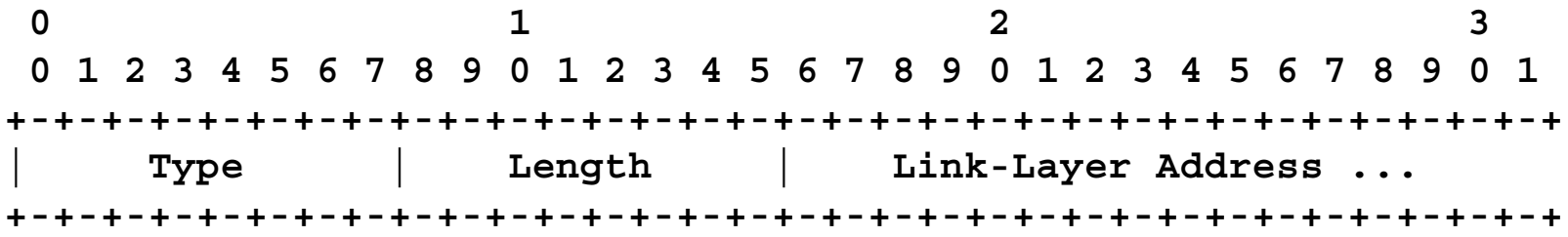
# Neighbor Advertisement messages

- Son mensajes ICMPv6 de Tipo 136, Código 0
- Utilizados para informar la dirección de capa de enlace correspondiente a una dirección IPv6
- La única opción permitida en ellos es la "Target Link-layer address"



# Opción Source/Target Link-layer address

- La opción Source Link-layer Address contiene la dirección de capa de enlace correspondiente a la dirección origen del paquete IPv6
- La opción Target Link-layer address contiene la dirección de capa de enlace correspondiente a la "Target Address" del mensaje Neighbor Solicitation




Type: 1 para Source Link-layer Address
2 para Target Link-layer Address

# Neighbor Cache

- Almacena la información correspondiente a mapeos IPv6 -> link-layer
- Cada entrada (IPv6 address, link-layer address) puede estar en alguno de los siguientes estados:

| NC state | Semantics |
|---|---|
| INCOMPLETE | Res. de Dir. en curso (todavía no determinada) |
| REACHABLE | Vecino alcanzable |
| STALE | Se desconoce si es alcanzable |
| DELAY | Se desconoce si es alcanzable (esperar indicación) |
| PROBE | Se desconoce si es alcanzable (enviando pruebas) |

# Ejemplo de tráfico Neighbor Discovery

```
% ping6 2004::1

12:12:42.086657 2004::20c:29ff:fe49:ebdd > ff02::1:ff00:1: icmp6: neighbor
sol: who has 2004::1(src lladdr: 00:0c:29:49:eb:dd) (len 32, hlim 255)
12:12:42.087654 2004::1 > 2004::20c:29ff:fe49:ebdd: icmp6: neighbor adv:
tgt is 2004::1(RSO)(tgt lladdr: 00:0c:29:c0:97:ae) (len 32, hlim 255)
12:12:42.089147 2004::20c:29ff:fe49:ebdd > 2004::1: icmp6: echo request
(len 16, hlim 64)
12:12:42.089415 2004::1 > 2004::20c:29ff:fe49:ebdd: icmp6: echo reply (len
16, hlim 64)
```

# ndisc6: Herramienta de diagnóstico

- Puede utilizarse para enviar NS para alguna dirección particular
- Ejemplo:

```
fgont@fernando-HP-530-Notebook:~$ ndisc6 2000:1::1 vboxnet0
Soliciting 2000:1::1 (2000:1::1) on vboxnet0...
Target link-layer address: 08:00:27:F9:73:04
 from fe80::a00:27ff:fef9:7304
```

# Neighbor Cache (contenido en *BSD)

- Ejemplo de salida de "`ndp -a`" (BSDs):

```
% ndp -a
Neighbor                        Linklayer Address  Netif Expire    S Flags
2004:1::f8dd:347d:8fd8:1d2c     0:c:29:49:eb:e7      em1 permanent R
fe80::20c:29ff:fec0:97b8%em1    0:c:29:c0:97:b8      em1 23h48m16s S R
2004:1::20c:29ff:fe49:ebe7      0:c:29:49:eb:e7      em1 permanent R
fe80::20c:29ff:fe49:ebe7%em1    0:c:29:49:eb:e7      em1 permanent R
2004::1                         0:c:29:c0:97:ae      em0 23h49m27s S R
2004::20c:29ff:fe49:ebdd        0:c:29:49:eb:dd      em0 permanent R
fe80::20c:29ff:fe49:ebdd%em0    0:c:29:49:eb:dd      em0 permanent R
fe80::20c:29ff:fec0:97ae%em0    0:c:29:c0:97:ae      em0 23h48m16s S R
2004::d13e:2428:bae7:5605       0:c:29:49:eb:dd      em0 permanent R
```

# Neighbor Cache (contenido en Linux)

- **Ejemplo de salida de "ip -6 neigh show" (Linux):**

```
$ ip -6 neigh show
fe80::a00:27ff:fef9:7304 dev vboxnet0 lladdr 08:00:27:f9:73:04 router STALE
2000::4000 dev vboxnet0 lladdr 11:22:33:44:55:66 PERMANENT
2000:1::1 dev vboxnet0 lladdr 08:00:27:f9:73:04 router REACHABLE
fe80::fc8d:15ed:7f43:68ea dev wlan0 lladdr 00:21:5c:0b:5d:61 router STALE
```

# Resolución de Direcciones en IPv6

## Algunos ataques...

# “Man in the Middle” o DoS

- Son en equivalente IPv6 del “Envenenamiento de ARP cache” en IPv4
- Ataque:
  - Esperar mensajes Neighbor Solicitation que contengan la dirección IPv6 de la víctima en el campo “Target Address”
  - Al recibir un NS, responder con un Neighbor Advertisement falsificado
- Si la “Target Link-layer address” corresponde aun nodo no existente, se descarta el tráfico, y resulta un DoS.
- Si la “Target Link-layer address” es la del atacante, entonces resulta un ataque “man in the middle” (MITM).

# Realizando el ataque con na6

- Ejecutar la herramienta como:

  # ./na6 –i IFACE –W VICTIMADDR –L –E MACADDR –c –o

- Lo que ahora se envíe a la víctima será enviado a MACADDR
- Verifícalo con tcpdump como:

  # tcpdump –i em0 –e –vv ip6

# Sniffeando en una red “conmutada”

- En las redes conmutadas se dificulta el “sniffing”
- Un truco “elegante” en IPv6 consiste en mapear la dirección de la víctima a:
  - La dirección broadcast Ethernet (ff:ff:ff:ff:ff:ff), o,
  - Direcciones Ethernet multicast (e.g., 33:33:00:00:01)
- Así, los paquetes serán enviados a todos (o varios) los sistemas de la red
- Tanto la víctima como el atacante recibirán el tráfico correspondiente

# Realizando el ataque con na6

- Ejecuta la herramienta así:

  # ./na6 –i IFACE –W VICTIMADDR –L –E ff:ff:ff:ff:ff:ff –c –o

- Los paquetes enviados a la víctima utilizaran como dirección MAC la dirección broadcast (ff:ff:ff:ff:ff:ff)
- Verifícalo con tcpdump:

  # tcpdump –i IFACE –e –vv ip6

- Incluso mejor:
  - Ejecutar tcpdump en el nodo atacante, previo al ataque (no verás el tráfico deseado)
  - Realiza el ataque, y repite lo anterior – ahora verás el tráfico en cuestión

# Introduciendo un bucle de ruteo en routers

- Cuando un router envía un NS para un host local, responder con un NA, indicando la MAC address del router
- El router enviará el paquete a su propia MAC address, por lo que recibirá una copia del mismo
- Decrementará el Hop Limit del paquete, y lo reenviará (a si mismo!)
- El proceso se repetirá hasta que el Hop Limit se haga 0

# Realizando el ataque con na6

- Ejecuta la herramienta como:

  # ./na6 –i IFACE –W 2000:1::80 –L –E ROUTERMAC –c –o

- En el router, ejecuta:

  % ping6 –i 20 –w 40 2000:1::80 &

- En el mismo router, ejecuta tcpdump como:

  sudo tcpdump –i em0 –e –vv ip6

- Prestar atención al Hop Limit de cada paquete!

# Resolución de direcciones

## Contramedidas

# Secure Neighbor Discovery (SeND)

- Utiliza criptografía para mitigar vulnerabilidades en Neighbor Discovery (incluyendo NS falsificados):
  - Se utilizan Direcciones Criptográficamente Generadas (CGAs) para relacionar direcciones IPv6 con un par de llaves asimétricas
  - Se utilizan firmas RSA para autentificar todos los mensajes de Neighbor Discovery
  - Se utilizan "caminos certificados" para verificar la autoridad de routers
- SeND es difícil de desplegar:
  - No tiene amplio soporte (por ej., no es soportado por Windows)
  - El requisito de una PKI hace que sea difícil de desplegar para escenarios generales

# Monitoreo de tráfico Neighbor Discovery

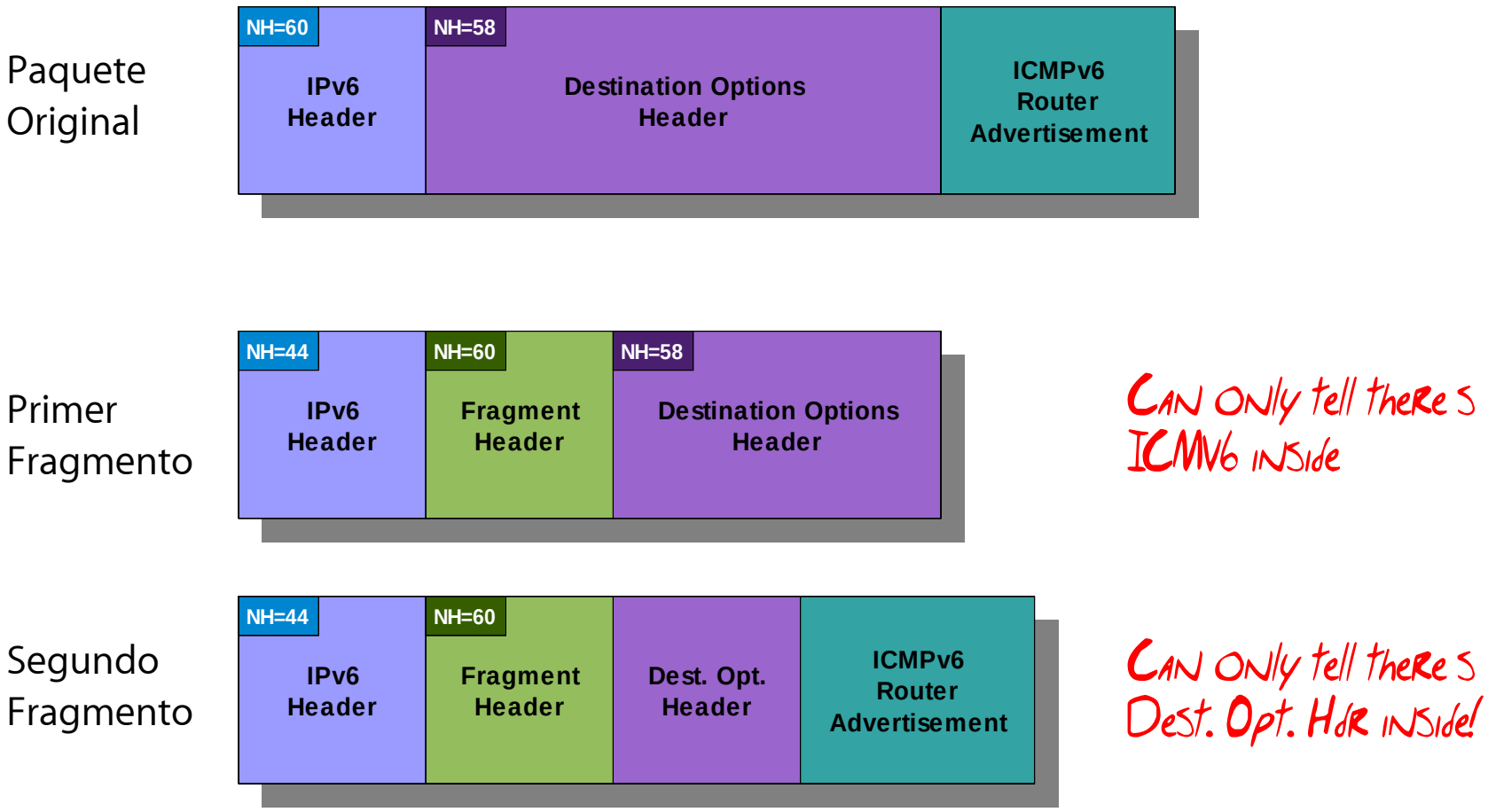
- Algunas herramientas mantienen un registro de los mapeos (IPv6 -> Ethernet) válidos, y suenan una alarma cuando cambian
- Similar a arpwatch en IPv4
- Sin embargo, típicamente son triviales de evadir:
  - ND “corre encima” de IPv6
  - Los paquetes pueden contener Encabezados de Extensión IPv6
  - Los paquetes pueden fragmentarse
  - Y como es tráfico local, no es posible introducir un MITM para que los “normalice”

# Monitoreo de tráfico Neighbor Discovery (II)

- Se puede insertar una cantidad arbitraria de Encabezados de Extensión
- La herramienta de monitoreo debe poder procesar la cadena de encabezados completa para “detectar” los mensajes ND

# Monitoreo de tráfico Neighbor Discovery (III)

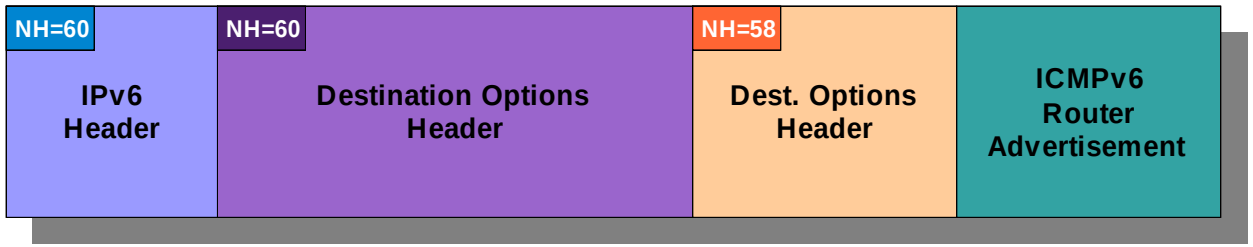
- **Combinación de Destination Options Header y fragmentación:**



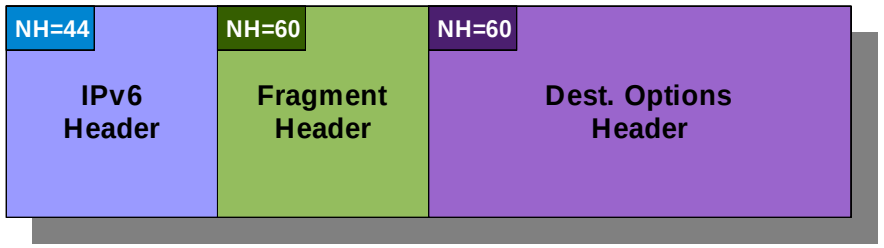
# Monitoreo de tráfico Neighbor Discovery (IV)

- **Dos encabezados Destination Options, y fragmentación:**

Paquete Original

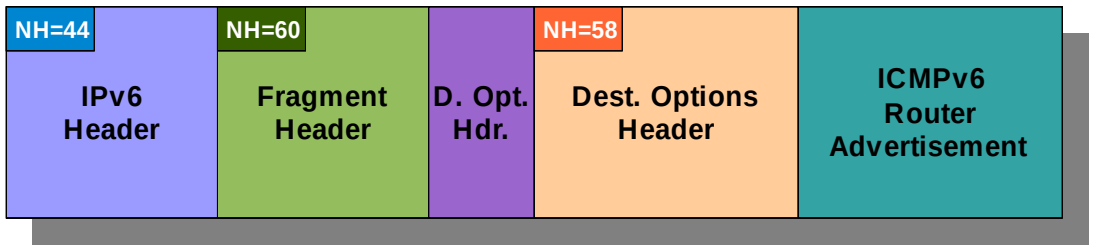



Primer Fragmento




Can only tell there's Dest. Opt. Hdr inside!

Segundo Fragmento




Can only tell there's Dest. Opt. Hdr inside!

# Restringir el acceso a la red local

- El tráfico Neighbor Discovery es local a la subred
- Separar a los nodos en distintas subredes limita el daño que el atacante puede causar
- No siempre es posible, pero usualmente es deseable

# Entradas estáticas en el Neighbor Cache

- Las entradas estáticas en el Neighbor Cache evitan el intercambio de NS/NA (al menos en teoría)
- Es algo análogo a las entradas estáticas en el ARP Cache de IPv4
- Advertencia: Algunas implementaciones seguían utilizando NS/NA pese a las entradas estáticas!

# Entradas estáticas en el Neighbor Cache (II)

- En los *BSD, se crean mediante el comando "ndp"
- Se pueden agregar entradas estáticas así:

  # ndp –s IPV6ADDR MACADDR
- Si IPV6ADDR es una dirección link-local, se debe especificar un “índice de interfaz”:

  # ndp –s IPV6ADDR%IFACE MACADDR

# Entradas estáticas en el Neighbor Cache (III)

- En Linux, se puede manipular el Neighbor Cache con el comando “ip”
- Las entradas estáticas pueden agregarse así:

sudo ip neigh add to IPV6ADDR lladdr MACADDR dev IFACE nud permanent

- El resultado se puede verificar con:

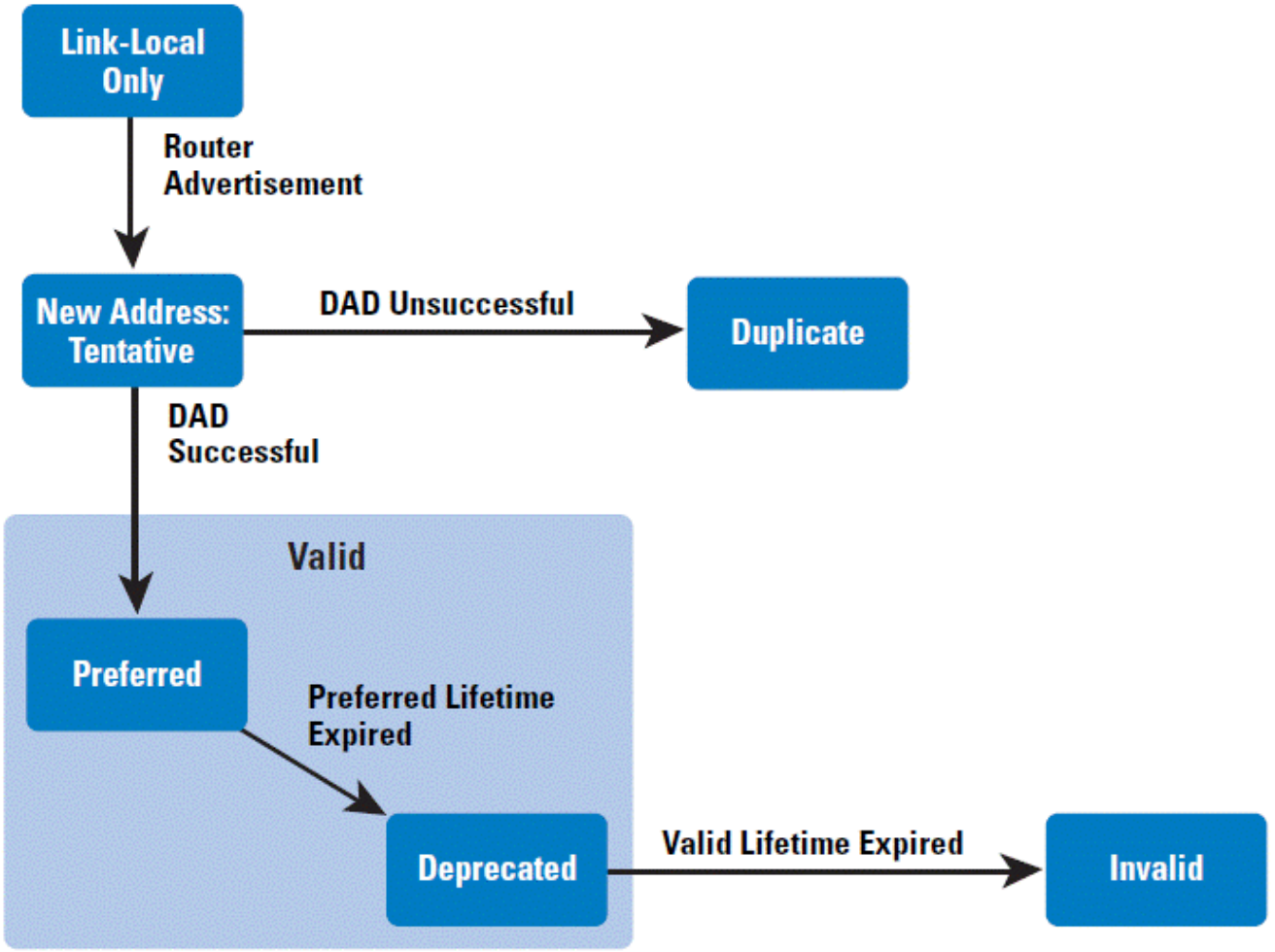
ip -6 neigh show

# IPv6 Stateless Address Autoconfiguration (SLAAC)

# Stateless Address Autoconfiguration

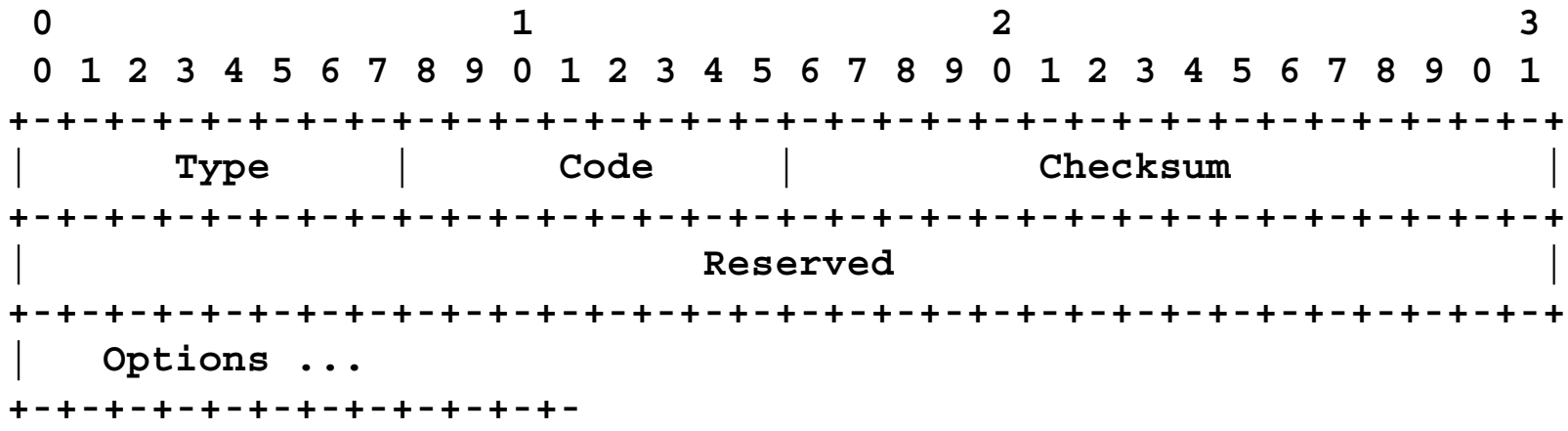
- A grandes rasgos, funciona así:
  1. El host configura una dirección link-local
  2. Chequea que la dirección sea única – es decir, realiza el procedimiento de Detección de Dirección Duplicada (DAD)
     - Enviar un NS, y ver si se obtiene respuesta
  3. El host envía un mensaje Router Solicitation
  4. Al recibir una respuesta, se configura una dirección IPv6 “tentativa”
  5. Chequea que la dirección sea única – es decir, realiza el procedimiento de Detección de Dirección Duplicada (DAD)
     - Enviar un NS, y ver si se obtiene respuesta
  6. Si es única, la dirección “tentativa” se convierte en una dirección valida

# SLAAC: Diagrama de estados



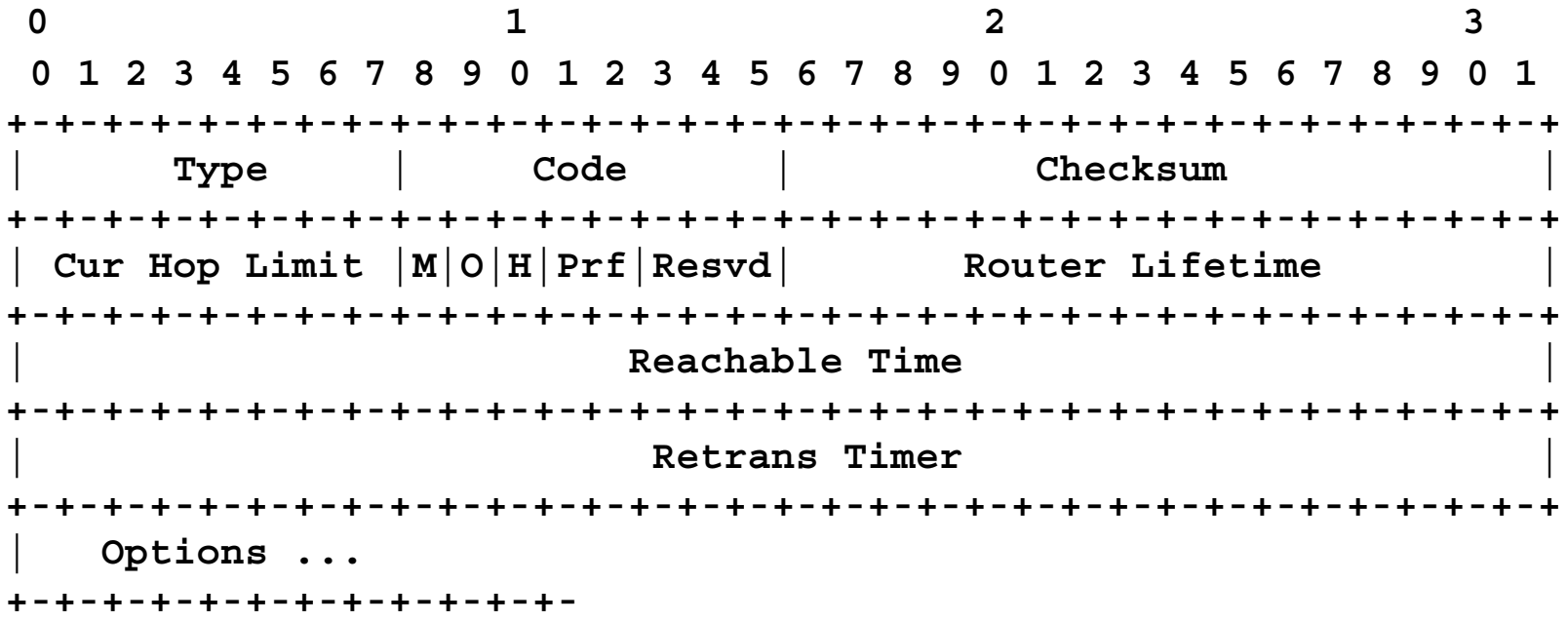
# Mensajes Router Solicitation

- Son mensajes ICMPv6 con Tipo 133, Código 0
- Utilizados para solicitar a un router local información de configuración de red
- La única opción actualmente permitida en estos mensajes es la Source Link-layer Address



# Mensajes Router Advertisement

- Son mensajes ICMPv6 de Tipo 134, código 0
- Utilizados para anunciar información de configuración de red

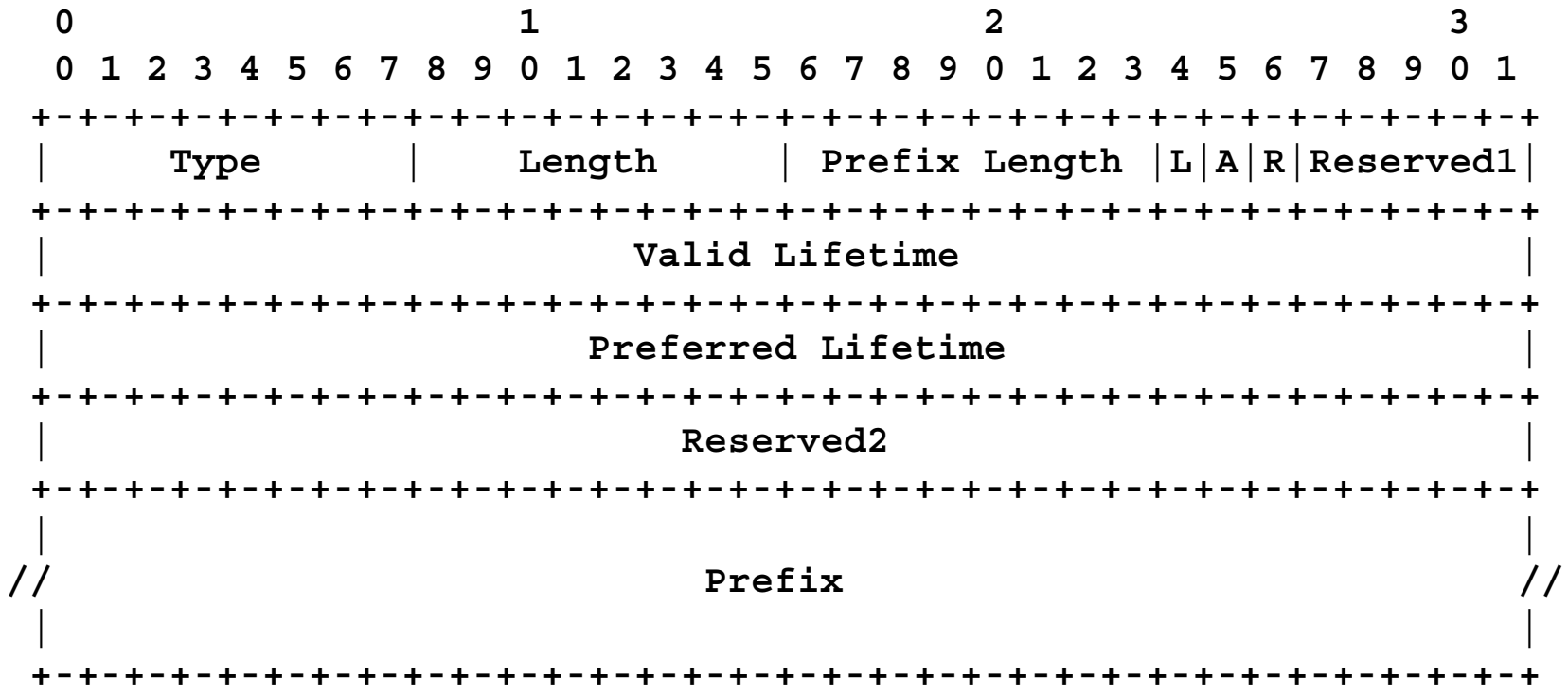


# Opciones permitidas en los mensajes RA

- Los mensajes RA pueden contener cualquiera de las siguientes opciones:
  - Source Link-layer address
  - Prefix Information
  - MTU
  - Route Information
  - Recursive DNS Server
- Usualmente, incluyen varias de ellas

# Opción Prefix Information

- Utilizada para especificar prefijos “on-link” y prefijos para auto-configuración



# Ejemplo de tráfico SLAAC

- Capturado de un host OpenBSD

```
17:28:50 :: > ff02::1:ffaf:1958: icmp6: neighbor sol: who has
fe80::20c:29ff:feaf:1958 (len 24, hlim 255)
17:28:52 fe80::20c:29ff:feaf:1958 > ff02::2: icmp6: router solicitation (src
lladdr: 00:0c:29:af:19:58) (len 16, hlim 255)
17:28:52 fe80::20c:29ff:fec0:97b8 > ff02::1: icmp6: router
advertisement(chlim=64, router_ltime=1800, reachable_time=0,
retrans_time=0)(src lladdr: 00:0c:29:c0:97:b8)(prefix info: LA
valid_ltime=2592000, preferred_ltime=604800, prefix=2004:1::/64) (len 56,
hlim 255)
17:28:52 :: > ff02::1:ffaf:1958: icmp6: neighbor sol: who has
2004:1::20c:29ff:feaf:1958 (len 24, hlim 255)
```

# rdisc6: Herramienta de diagnóstico

- Envia mensajes RS, y decodifica las respuestas
- Ejemplo:

```
# rdisc6 -v eth0
Soliciting ff02::2 (ff02::2) on eth0...

Hop limit                 :           64 (      0x40)
Stateful address conf.    :           No
Stateful other conf.      :           No
Router preference         :       medium
Router lifetime           :           30 (0x0000001e) seconds
Reachable time            :  unspecified (0x00000000)
Retransmit time           :  unspecified (0x00000000)
 Prefix                   : fc00:1::/64
  Valid time              :      2592000 (0x00278d00) seconds
  Pref. time              :       604800 (0x00093a80) seconds
 Source link-layer address: 00:4F:4E:12:88:0F
 from fe80::24f:4eff:fe12:880f
```

# Tabla de “default routers” en *BSD

- Ejemplo de salida de “**`ndp -r`**” (BSDs):

```
% ndp -r
fe80::20c:29ff:fec0:97b8%em1 if=em1, flags=, pref=medium, expire=20m23s
fe80::20c:29ff:fec0:97ae%em0 if=em0, flags=, pref=medium, expire=26m53s
```

# Prefjios en *BSD

- Ejemplo de salida de "`ndp -p`" (BSDs):

```
% ndp -p
2004::/64 if=em0
flags=LAO vltime=2592000, pltime=604800, expire=29d23h57m4s, ref=2
  advertised by
    fe80::20c:29ff:fec0:97ae%em0 (reachable)
2004:1::/64 if=em1
flags=LAO vltime=2592000, pltime=604800, expire=29d23h50m34s, ref=2
  advertised by
    fe80::20c:29ff:fec0:97b8%em1 (reachable)
fe80::%em1/64 if=em1
flags=LAO vltime=infinity, pltime=infinity, expire=Never, ref=0
  No advertising router
fe80::%em0/64 if=em0
flags=LAO vltime=infinity, pltime=infinity, expire=Never, ref=0
  No advertising router
fe80::%lo0/64 if=lo0
flags=LAO vltime=infinity, pltime=infinity, expire=Never, ref=0
  No advertising router
```

# SLAAC

## Implicancias sobre privacidad

# Super “cookies”

- Cuando se utiliza SLAAC, el Interface ID se setea de acuerdo a la dirección MAC de la interfaz de red
- Dado que las direcciones MAC son globalmente únicas, esto equivale a una “super cookie”
- Se hace posible “tracear” la actividad de un host, por mas que el mismo se mueva en la red
  - en cuyo caso cambiaría el prefijo, pero se mantendría el mismo Interface ID

# Extensiones de Privacidad

- Para combatir este problema, se estandarizaron las “extensiones de privacidad para SLAAC”
- Básicamente, se reemplaza el I-ID por uno aleatorio, que varia en el tiempo
  - en ciertos casos esto se vuelve indeseable, ya que hace más problematica la administración de la red (loggeo, etc.)
- Algunos OSes (e.g., Windows 7) utilizan un esquema alternativo:
  - El I-ID se selecciona a partir de un hash computado sobre el prefijo de red
  - Para un prefijo dado, las direcciones IPv6 son constantes
  - Si cambia el prefijo, cambia el I-ID
  - Esta alternativa contiene “lo mejor de los dos mundos”

# Algunas sysctl's para ***Privacy Addresses***

- No todos los sistemas implementan extensiones (por ej., OpenBSD no lo hace)
- Algunos las implementan, pero no las habilitan por defecto (e.g., FreeBSD)
- Sysctl's para controlar la operación de las Extensiones de Privacidad en FreeBSD:
  - `net.inet6.ip6.use_tempaddr` (defaults to 0)
    - Controls whether Privacy addresses are configured
  - `net.inet6.ip6.temppltime` (defaults to 86400)
    - Specifies the "preferred lifetime" for privacy addresses
  - `net.inet6.ip6.tempvltime` (defaults to 604800)
    - Specifies the "valid lifetime" for privacy addresses
  - `net.inet6.ip6.prefer_tempaddr` (defaults to 0)
    - Controls whether privacy addresses are "preferred" (i.e., whether outgoing "conections" should use privacy addresses)

# IPv6 SLAAC

## Algunos ataques…

# Deshabilitar un router

- Falsificar un Router Advertisement “impersonando” al router local
- Setear el “Router Lifetime” en 0 (u otro valor pequeño)
- Como resultado, la victima eliminará al router en cuestión de la lista de la “default routers list”

# Realizando el ataque con ra6

- Ejecutar ra6 como:

  # ./ra6 –i IFACE –s ROUTERADDR –d TARGETADDR –t 0 –l 1 –v

- Por ejemplo:

  # ./ra6 –i em0 –s fe80::a00:27ff:fef9:7304 -d ff02::1 –t 0 –l 1 –v

- Chequar los resultados en Linux

# Explotar DAD para DoS

- Esperar NS que utilicen como Source Address la dirección IPv6 “unspecified” address (::).
- Al recibir uno, responder con un mensaje Neighbor Advertisement
- La dirección se considerará “duplicada”, y DAD “fallará”
- El host no podrá usar la dirección “tentativa” en cuestión

# DoS con RAs con parámetros incorrectos

- El atacante puede falsificar RAs con parámetros incorrectos
- Por ejemplo, anunciar un Hop Limit (Cur Hop Limit) pequeño, de modo que los paquetes sean descartados por un router próximo
- Tal vez sea un DoS mas dificil de detectar...

# Realizando el ataque con ra6

- Ejecutar la herramienta así:

  # ./ra6 –i IFACE –s ROUTERADDR –d TARGETADDR –c HOPS –v

- Por ejemplo:

  # ./ra6 –i em0 –s fe80::a00:27ff:fef9:7304 -d ff02::1 –c 1 –v

- Linux, por ejemplo, es vulnerable a este ataque

# Posibles contramedidas

- Desplegar SeND (Secure Neighbor Discovery)
- Monitorear el tráfico Neighbor Discovery traffic (e.g., con NDPMon)
- Restringir el acceso a la red local
- Utilizar configuración manual (y deshabilitar RAs)
- Deplegar Router Advertisement Guard (RA-Guard)

# Router Advertisement Guard

- Muchas organizaciones utilizan “RA-Guard” como “primer línea de defensa” contra RAs falsificados
- Se trata de una política de filtrado aplicable en switches
- RA-Guard funciona (a grandes rasgos) así:
  - Se configura el switch de modo que acepte RAs solamente en un puerto especificado
  - Los RAs recibidos en otros puertos son descartados
- RA-Guard asume que el switch puede identificar los RA

# Problemas con RA-Guard

- RA-Guard debe identificar en layer-2, paquetes de layer-3 (los RAs)
- El uso (teorico!) de encabezados de extensión y fragmentación dificultan esta tarea
- El filtrado de paquetes se vuelve básicamente imposible:
  - Se debe poder reensamblar paquetes IPv6 en capa 2 (!)
  - No es posible normalizar el tráfico
  - en fin...

# Mejoras a RA-Guard

- Es posible lograr el monitoreo de tráfico Neighbor Discovery
- RA-Guard debe procesar el “IPv6 header chain” completo (obvio!)
- Se debe prohibir el uso de (a menos que se utilice SeND)
- Ver trabajo actual en el ámbito de la IETF:
  - http://tools.ietf.org/id/draft-gont-v6ops-ra-guard-evasion-01.txt
  - http://tools.ietf.org/id/draft-gont-6man-nd-extension-headers-01.txt
  - Or http://tools.ietf.org/id/gont

# Soporte de IPsec

# Breve reseña y consideraciones…

Mito: *"IPv6 es mas seguro que IPv4 porque tiene soporte mandatorio de IPsec"*

- Actualmente, se el soporte de IPsec es mandatorio en toda implementación de IPv6 (y opcional en IPv4) – aunque la IETF está en proceso de cambiar este requerimiento
- Sin embargo, a los fines prácticos, esto es completamente irrelevante:
  - Es/era mandatorío el *soporte* de IPv6 – no así su *utilización*
  - Así y todo, existen muchas implementaciones IPv4 con soporte IPsec, como también implementaciones IPv6 sin soporte IPsec
- Existen en IPv6 básicamente los mismos problemas para el despliegue de IPsec que en IPv4
- Por tal motivo, no existen motivos para esperar más uso de IPsec con IPv6 que el que se tiene con IPv4

# Seguridad de los Mecanismos de Transición/Co-existencia

# Breve reseña

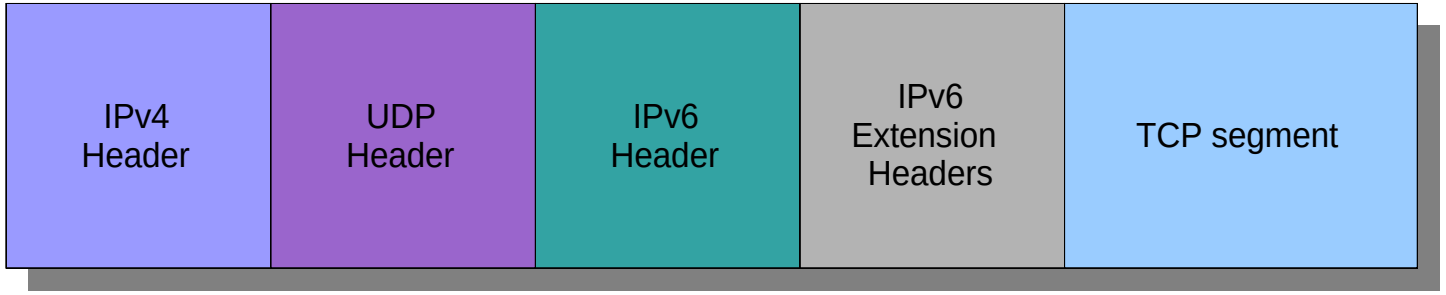
- El plan original de transición era el uso de dual-stack (*si, este plan falló*)
- La estrategia actual es un plan de transición/co-existencia basado en un grupo de herramientas:
  - Dual Stack
  - Túneles “configurados”
  - Túneles automáticos (ISATAP, 6to4, Teredo, etc.)
  - Traducción (por ej., NAT64)
- Algunas variantes de túneles automáticos (como Teredo e ISATAP) están habilitados por defecto en Windows Vista y Windows 7

# Consideraciones de seguridad

- La mayoría de estas tecnologías incrementan la complejidad de la red, y así las potenciales vulnerabilidades
- Muchas de estas tecnologías introducen Puntos Únicos de Falla ("Single Point of Failure") en la red.
- Algunos de estos mecanismos merecen consideraciones de privacidad:
  - ¿Por dónde circula su tráfico Teredo y 6to4?
  - Esto puede (o no) ser importante para su red
- Algunas de ellas han sido explotadas para violar políticas de seguridad, ya que en ocasiones no son tenidas en cuenta por firewalls y NIDS

# Consideraciones de seguridad (II)

- Los paquetes resultantes de la utilización te tecnologías de transición co-existencia pueden tener varias capas de encapsulamiento
- Esto dificulta notablemente la aplicación de políticas de filtrado, a menos que el firewall tenga soporte de dicha tecnología de transición.
- Ejemplo de tráfico Teredo:




- Ejercicio ilustrativo: escribir un filtro para libpcap que “detecte” paquetes TCP/IPv6 transportados sobre Teredo, destinados al host 2001:db8::1, puerto TCP 25.

# Implicancias de seguridad de IPv6 en redes IPv4

# Breve reseña

- Muchos sistemas tienen algun tipo de soporte IPv6 habilitado “por defecto” – soporte IPv6 nativo, y usualmente suporte de algún mecanismo de transición/co-existencia
- Por ejemplo, Linux, *BSD, y Windows Vista/7 tienen soporte IPv6 nativo habilitado “por defecto”
- Windows Vista/7 tienen, adicionalmente, soporte Teredo e ISATAP habilitado “por defecto”
- Es importante destacar que algunas tecnologías de transición, como Teredo, fueron diseñadas para funcionar incluso a través de NATs

# Consideraciones de seguridad

- Un atacante con acceso a una red local podría realizar ataques contra SLAAC (falsificando RAs), haciendo que los hosts locales configuren direcciones IPv6
- Esto podría permitir que se evadan controles de filtrado de tráfico y/o NIDS
- El uso de tecnologías como Teredo podría resultar en que incluso hosts que están detrás de NATs quedaran expuestos a la red pública (Internet)
- Por tales motivos,
  - Incluso si una red no espera utilizar IPv6, debe tener en cuenta las implicancias de seguridad de este protocolo (por ej. en lo que respecta a filtrado y monitoreo)
  - Si se espera que en una red IPv4 no se utilicen mecanismos de transición/co-existencia, se deberían aplicar las políticas de filtrado correspondientes

# Filtrado de Tecnologías de Transición

| Tecnología | Regla de filtrado |
|---|---|
| Dual-Stack | Automático |
| IPv6-in-IPv4 tunnels | IPv4 Protocol == 41 |
| 6to4 | IPv4.Protocol == 41<br>IPv4.{src,dst} == 192.88.99.0/24 |
| ISATAP | IPv4 Protocol == 41 |
| Teredo | IPv4.dst == known_teredo_servers &&<br>UDP.DstPort == 3544 |
| TSP | IPv4.dst == known_teredo_servers &&<br>{TCP,UDP}.dst == 3653 |

# Trabajo a futuro

# Algúnas áreas clave en las que se necesita progreso

- Mejora de implementaciones IPv6
  - Las implementaciones de IPv6 todavía no han estado en el foco de los atacantes. Es muy probable que se descubran muchas vulnerabilidades y bugs en las implementaciones IPv6 actuales.
  - Existen muy pocas herramientas de ataque disponibles públicamente
- Soporte de IPv6 en dispositivos de seguridad
  - IPv6 no tiene el mismo nivel de soporte que IPv5 en dispositivos tales como firewalls, IDS/IPS, etc.
  - Esto es clave para poder aplicar en IPv6 políticas de seguridad comparables con las aplicadas en IPv4.
- Educación/Entrenamiento
  - Desplegar IPv6 sin un conocimiento aceptable del mismo podría llevar a resultados muy desfavorables
  - Se necesita entranamiento para ingenieros, técnicos, personal de seguridad, etc., previo al diseño y puesta en funcionamiento de una red IPv6.

**20 million engineers need IPv6 training, says IPv6 Forum**
The IPv6 Forum - a global consortium of vendors, ISPs and national research & Education networks - has launched an IPv6 education certification programme in a bid to address what it says is an IPv6 training infrastructure that is "way too embryonic to have any critical impact.“ (**http://www.itwire.com**)

# Algunas conclusiones

# Algunas conclusiones

- Pese a que IPv6 provee una funcionalidad similar a la de IPv4, muchos de los mecanismos utilizados son diferentes. Por tal motivo, requiere de un análisis cuidadoso.
- Las implicancias de seguridad de IPv6 deben ser consideradas previo a su despliegue, para evitar un impacto negativo en las redes correspodientes
- Dado que la mayoría de los sistemas de uso general cuenta con soporte IPv6, incluso los administradores de redes IPv4 deberían conocer las implicancias de seguridad de IPv6
- Incluso si todavía no lo ha planificado, es probable que necesite desplegar IPv6 en el corto plazo.
- Es hora de capacitarse, entrenarse, y experimentar con IPv6!

Preguntas?

# Para más información

- Súmate a la lista de correo IPv6 Hackers en:

  http://www.si6networks.com/community/

- Súmate a la lista de correo del Foro de Seguridad de LACNIC:

  http://seguridad.lacnic.net

# Gracias!

Fernando Gont

fgont@si6networks.com


www.si6networks.com