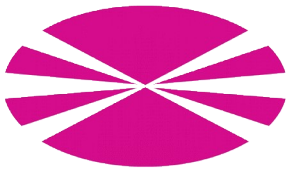
ATAQUES EN REDES IPV6

Mauro de los Santos Nodar mauro.delossantos@udc.gal

Javier Díaz Santiso javier.diazs@udc.gal

Lucas Rega Posada lucas.rega@udc.gal



UNIVERSIDADE DA CORUÑA

Obxectivos

1. Comprensión de conceptos básicos de redes IPv6.

2. Capacidade para descubrir equipos en redes IPv6.

3. Aprender funcionamento de ataques MITM en IPv6.

4. Coñecemento doutros ataques en IPv6.

5. Introdución a diferentes ferramentas de ataque en IPv6.

6. Aprendizaxe do proceso de montaxe dun servidor Rogue en IPv6.

Índice

1. Introdución.

Cada vez son máis as noticias de fallos de seguridade e ferramentas de hacking centradas en IPv6. Aínda que a implementación deste protocolo pareza moi recente, a definición do mesmo xa ten algún tempo, concretamente empezou a fraguarse no ano 1995 co RFC 1883. Tendo isto en conta e máis a velocidade coa que evolucionan as comunicacións e os entornos TI, estamos falando dun período de tempo considerable.

Ao longo destes anos vimos unha serie de *bugs* de grande seriedade no que respecta ao protocolo IPv6. Imos comentar algúns deles.

O primeiro grande escándalo sobre un *bug* en IPv6 dentro do mundo do hacking tivo como protagonista ás empresas *CISCO*, *ISS* e ao hacker *Mychael Lynn.* A historia comezou cando *CISCO* parchea en 2005 un *bug* na pila IPv6 dos seus routers, porén, a información que publica sobre ese bug, para *Mychael Lynn* (traballador de *ISS*) non era de todo completa e verídica e o risco era moito maior. E non se equivocaba, pouco tempo máis tarde, el mesmo conseguiu a forma de tomar o control total dos routers con soporte IPv6 de *CISCO*.

No hai que irse tan atrás para ver fallos só na cuestión da implementación da pila de IPv6. En 2013, a suite de seguridade *Kaspersky Internet Security 2013* sufriu un *bug* que permitía a un atacante remoto bloquear todo equipo que tivese instalado ese software por medio da ferramenta *firewall6* de *THC* (*The Hacker Choice*), da que falaremos máis adiante. Simplemente con enviar una serie de probas a calquera porto cos parámetros 18,19,20,21 o equipo quedaba totalmente bloqueado (*firewall6* <*interface*> <*target*> <*port*> *19*).

Outro proxecto a nomear dentro dos sistemas *__NIX__* é o proxecto *SUDO*, o cal se viu afectado por un fallo ao implementar o soporte sobre IPv6. Este problema viu motivado por como se configurara o tratamento dos hosts: A descrición dun ou varios hosts pódese facer en base ao nome do equipo, a súa IP ou a dirección de rede de varios equipos xunto á súa máscara. O *bug*, produciuse por un fallo tan simple como o esquecemento dunha instrución *break* dentro dun *switch*, cuxa consecuencia era que despois de executar o código para IPv4, se non había coincidencias, se seguise para IPv6 aínda que a dirección fora IPv4.

Estes, son algúns dos ataques, fallos ou *bugs* en IPv6 máis coñecidos ou salientables, pero coma eles, houbo centos de casos e un dos grandes problemas é que a grande maioría de sistemas non contemplan IPv6 dentro dos posibles vectores de ataque, polo que as medidas de seguridade seguen ancoradas en IPv4. Existen implementadas moitas solucións para *ARP-Spoofing*, pero non tantas para protexerse de *Neighbor Spoofing*, o seu equivalente en IPv6 e do que tamén falaremos máis adiante. Exactamente o mesmo pasa en ataques usando a electrónica de rede, como poden ser ataques *DoS* baseados en tormentas de mensaxes *Router Advertisement (RA)*, ou en ataques *MItM* baseados no protocolo *SLAAC,* do cal falaremos en apartados posteriores.

Facendo referencia a isto último, cabe destacar que tamén é moi importante o factor humano, xa que moitos administradores de redes aínda non pensan en que IPv6 está nas súas redes e ese é o gran problema, que si que o está, e que pode ser explotado como veremos neste documento.

2. Conceptos básicos de IPv6.

O seguinte punto vaise centrar nos conceptos básicos deste protocolo, que debera entrar fai tempo na nosa vida cotiá e porén segue sendo un gran descoñecido. Este apartado pretende ser breve e non entrar en detalle de conceptos como a análise das cabeceiras ou o direccionamento en IPv6, se non que a súa finalidade é refrescar conceptos básicos que xa deberían ser coñecidos por todos, e que son os estritamente necesarios para entender posteriormente as accións realizadas nos ataques a exemplificar. Para un estudo en máis profundidade sobre o que é IPv6, as súas diferenzas e melloras con IPv4, as súas limitacións e fallos, etc. existe moitísima información a consultar na rede.

O primeiro de todo que deberemos saber é que as direccións de IPv6 son de 128 bits e que se escriben en hexadecimal separadas en grupos de 16 bits, é dicir 4 valores hexadecimais. Isto fai que queden escritas da forma: **fe80:123:0000:0000:0000:0000:0000:1ab0**.

Destacar como detalle útil, que cando exista un grupo de catro ceros consecutivos podemos utilizar o acurtador reducindo a dirección anterior a **fe80:123::1ab0**. Esta redución pódese aplicar unha vez por dirección e só para grupos de catro ceros consecutivos.

En segundo lugar, aínda que non existe a máscara de rede ou *netmask* de IPv4, temos a súa equivalente chamada Prefixo de Subrede. Isto naceu pola gran cantidade de problemas de *subnetting*, *supernetting* e asignación de determinadas máscaras problemáticas en IPv4 (como por exemplo 255.0.124.255).

En IPv6 o prefixo ten a mesma función que a *netmask*, xestionar a visibilidade de rede e utilizarse para facer *subnetting* e *supernetting*, coa diferenza de que todos os uns van seguidos e ao principio, por definición do estándar.
Desta forma se tivésemos dúas direccións IPv6 (sen usar *default gateway*): **A**(fc00::2000:0001/96) e **B**(fc00::2001:0001/112), ao facer un ping de A a B obteríamos **timeout**, e ao facelo de B a A obteríase **host unreachable**, debido a que A non entra na mesma rede ca B, pero B si que está dentro da de A.

Para interconectar redes IPv6 ao igual que en IPv4 usaremos *default gateway*, así como tamén usaremos servidores IPv6 para a resolución de nomes. Apuntar que estes se usarán para resolver todo tipo de consultas cando se utilice o protocolo IPv6, é dicir, resolverá rexistros tanto tipo AAAA como tipo A cando na rede local se utilice IPv6. Estes dous rexistros mencionados, son aqueles encargados de conter a dirección, IPv6 ou IPv4 respectivamente, asociada á un nome de dominio dentro do protocolo DNS.

Unha vez que vimos as direccións, a máscara, a interconexión entre redes e algo de DNS imos comentar rapidamente unha serie de características ou novidades referentes a IPv6:

- Desaparecen as direccións *Broadcast* e introdúcense as *Anycast*.

- Simplifícase e mellórase o formato de cabeceira respecto a IPv4.

- O protocolo *ARP* de IPv4 é substituído.
- Mellora da autoconfiguración dos equipos axilización do enrutamento.

- Seguridade intrínseca e capacidade para etiquetado de fluxos (*QoS*).

- Referente ao *DNS*, incorpóranse os rexistros AAAA (equivalentes a rexistros A de IPv4).

2.1 Direccións link local

Unha vez sabidos estes coñecementos sobre IPv6, temos que entender que tanto se facemos unha configuración manual como automática das direccións IPv6, as tarxetas de rede con soporte IPv6 terán asociada unha dirección coñecida como de vínculo local ou *link local*.

Esta dirección é xerada de xeito automático polo propio equipo e é anunciada pola rede para evitar a duplicidade da mesma usando o protocolo *NDP*, do que falaremos máis adiante.

Esta dirección é do rango **fe80::/10** (cuxa equivalente en IPv4 é 169.254.1.X – 169.254.254.X).

```
inet6 fe80::a00:27ff:fe4c:b11a
```

1. Dirección Link Local

As direccións *link local* non son enrutables, pero si que son utilizadas para comunicarse co router ou con calquera servidor ou equipo que se atope na mesma rede local.

Se non se tocou nada da configuración terase unha destas direccións asociadas a todas e cada unha das tarxetas nas que se habilitara o protocolo de IPv6. A parte de *host* das mesmas será calculada por defecto a partir da *MAC* da tarxeta. Este tema abordarémolo máis adiante no apartado de descubrimento de rede.

2.2 Direccións *Well-Known* en IPv6.

Ademais das direccións de *link local* hai unha serie de direccións en IPv6 que deben ser coñecidas, así que imos describir as máis importantes:

- ::/128 → Dirección IPv6 indefinida.
- ::/0 → Dirección da ruta por defecto nunha táboa de enrutamento.
- ::1/128 → *Localhost*.
- fe80::/10 → Direccións *link local*.
- ff02::/16 → *Multicast*.
- fc00::/7 → Direccións para redes IPv6 privadas.
- ::ffff:0:0/96 → Direccións IPv4 mapeadas a IPv6. Utilizadas para interconexións de protocolos IPv4 e IPv6.

- 64:ff9b::/96 → Direccións IPv6 xeradas a partires de direccións IPv4.
- 2002::/16 -> Indica que é unha rede 6to4.

A parte destas direccións *Well-Known*, temos tamén unha serie de direccións reservadas como poden ser:

- 2001::/32 -> Usado polo protocolo de túneles Teredo.
- 2001:2::/48 -> *Benchmarking* en IPv6.
- 2001:10::/28 -> Identificadores criptográficos *Hash*.
- 2001:db8::/32 -> Documentación ou exemplos IPv6.

2.3 Precedencia de protocolos.

Unha das preguntas máis frecuentes e que debemos entender é que cando temos un equipo con IPv4 e IPv6, que protocolo vai utilizar o SO? Nos sistemas actuais o máis probable é que ambas versións convivan así que o SO deberá elixir en cada comunicación cal usar.

Isto farao seguindo unhas normas de precedencia definidas no RFC 3484 e no máis recente RFC 6724.

Estas normas son establecidas por diferentes algoritmos que teñen múltiples parámetros en consideración, como pode ser detalles da configuración, a existencia ou non de *default gateway*... e moitos máis.

Para descubrilo, podemos usar o comando en Windows, *Netsh interface ipv6 show prefix*, e en Linux, *ip -f inet6 addrlabel show*, onde teremos unha táboa de prioridades. A saída será do estilo de:

```
C:\Users\      >netsh interface ipv6 show prefixpolicies
Consultando el estado activo...

Precedencia Etiq.  Prefijo
----------- -----  ------------------------------
         50     0  ::1/128
         40     1  ::/0
         35     4  ::ffff:0:0/96
         30     2  2002::/16
          5     5  2001::/32
          3    13  fc00::/7
          1    11  fec0::/10
          1    12  3ffe::/16
          1     3  ::/96
```

2. Táboa de precedencia en Windows

```
root@kali:~# ip -f inet6 addrlabel show
prefix ::1/128 label 56
prefix ::/96 label 56
prefix ::ffff:0.0.0.0/96 label 56
prefix 2001::/32 label 56
prefix 2001:10::/28 label 56
prefix 3ffe::/16 label 56
prefix 2002::/16 label 56
prefix fec0::/10 label 56
prefix fc00::/7 label 56
prefix ::/0 label 0
```

3. Táboa de precedencia en Linux

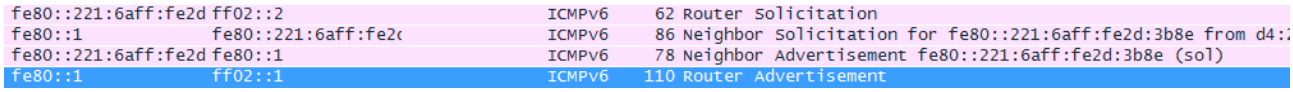
Onde, tendo en conta o antes explicado nas direccións *Well-Known*, vemos como IPv4 ten preferencia 35 e a ruta por defecto e o *Localhost* en IPv6 teñen preferencia 40 e 50.
O algoritmo de precedencia da prioridade a IPv6 sobre IPv4, se é posible establecer unha comunicación con este protocolo, pero isto sempre se poderá cambiar polo medio de comandos *Netsh* ou *Ip* (dependendo do noso SO). Ademais, estas regras só actúan cando non se estableceu explicitamente ningunha restrición previa.

Á hora de escoller entre v4 e v6 terá moito peso tamén a resolución de nomes, xa que se o servidor *DNS* só responde con IPv4, só se utilizará esta versión do protocolo, por exemplo.

2.4 NDP.

Como xa mencionamos anteriormente, para descubrir veciños nunha rede IPv6 non existe o protocolo *ARP* ou *RARP*, se non que todo se basea en *ICMPv6*. O protocolo para facer este descubrimento chamase *NDP*, *Neighbor Discovery Protocol*, e implementa 5 tipos de mensaxes diferentes, algúns deles equivalentes a *ARP*, concretamente o **NS** (*Neighbor Solicitation*) onde se pide a resolución dunha *MAC* asociada a unha dirección IPv6, e o **NA** (*Neighbor Advertisement*), onde se contesta coa *MAC* da dirección IPv6 buscada.

O normal é que estas mensaxes sexan enviadas a unha dirección *Multicast* á que só contesta o veciño con esa dirección configurada, pero tamén poden ser mensaxes *Unicast* enviados para interrogar para saber se coñece ou non coñece a dito veciño.



4. Descubrimento de veciños con mensaxes NS e NA.

Todas as direccións *MAC* asociadas a direccións IPv6 quedarán almacenadas no que se chama **Táboa de veciños** ou ***Neighbor Table*** e pode ser consultada en calquera momento usando o comando, *netsh interface ipv6 show neighbor* en Windows ou *ip neighbour show* en Linux.

```
C:\Users\mauro>netsh interface ipv6 show Neighbor

Interfaz 1: Loopback Pseudo-Interface 1


Dirección de Internet                          Dirección física   Tipo
---------------------------------------------  -----------------  -----------
ff02::c                                                           Permanente
ff02::16                                                          Permanente
ff02::1:2                                                         Permanente
ff02::1:3                                                         Permanente

Interfaz 3: Wi-Fi


Dirección de Internet                          Dirección física   Tipo
---------------------------------------------  -----------------  -----------
ff02::1                                        33-33-00-00-00-01  Permanente
ff02::2                                        33-33-00-00-00-02  Permanente
ff02::16                                       33-33-00-00-00-16  Permanente
ff02::fb                                       33-33-00-00-00-fb  Permanente
ff02::1:2                                      33-33-00-01-00-02  Permanente
ff02::1:3                                      33-33-00-01-00-03  Permanente
ff02::1:ff1a:eb64                              33-33-ff-1a-eb-64  Permanente

Interfaz 20: Ethernet
```
5. Táboa de veciños en IPv6

2.5 LLMNR

Existe un protocolo chamado *LLMNR*, que utilizando *Multicast* permite resolver as direccións IPv4 e/ou IPv6 asociadas a un nome de dominio. Este sistema permite realizar buscas locais ou mediante o uso de resolución de rexistros A e/ou AAAA cun servidor *DNS*. Utilizando estas últimas tres cousas vistas, é dicir: este tipo de resolución de nomes,

a busca de direccións *MAC* de veciños con NDP e a táboa de precedencia, os equipos constrúen a comunicación entre equipos IPv6.

2.6 Configuración de equipos IPv6 na rede.

Para configurar o protocolo IPv6 dos equipos dunha rede existen diferentes alternativas:

1. Configuración estática ou manual, especificando dirección IPv6, *default gateway* e servidores *DNS* de forma individual e manual, ou mediante un *script*.
2. Utilizando un servidor ***DHCPv6*** como veremos no apartado final do traballo.
3. Mediante *NDP* e as mensaxes *RS*, *RA* e *Redirect*, xunto co funcionamento *SLAAC* dos equipos.

   A idea é que un equipo pode conectarse automaticamente nunha rede IPv6 se coñece algún router de conexión. Para iso, o equipo realiza unha petición **RS** (*Router Solicitation*) en busca de una porta de enlace, todos os routers contestaranlle cun **RA** (*Router Advertisement*) dándolle a *SLAAC* a información necesaria para que o equipo se autoconfigure unha dirección IPv6 que lle permita conectividade a través do router. Se hai máis dun router e o equipo elixe un, como primeiro salto, erróneo, este contestaralle cunha mensaxe *NDP* de tipo **Redirect** informándolle de cal é a mellor ruta para que actualice a súa táboa de enrutamento.

Por suposto, estes dos últimos métodos van poder ser explotados para realizar ataques *DoS* e *MitM* en redes IPv6 como veremos máis adiante.

2.7 DNS Autodiscovery

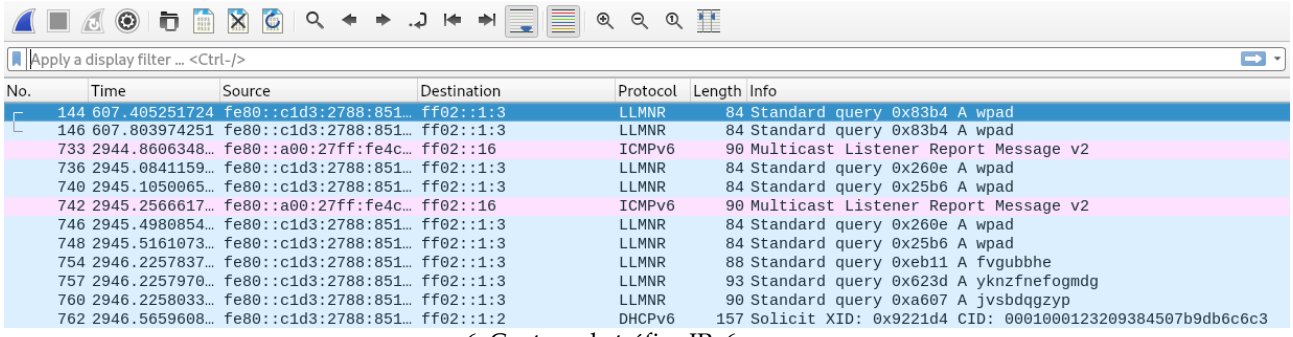
Cando un equipo se conecta á rede a través de configuración *SLAAC*, existe o problema de que non se poden configurar os servidores *DNS* e todas as peticións se reducen a *LLMNR* de tipo difusión en busca de posibles servidores na rede de vínculo local. Porén, se o servidor fose externo sería necesario contar cun servizo de resolución de nomes na rede IPv6. Para iso, cando non se configura ningún servidor, búscase automaticamente en 3 direccións establecidas polo estándar *IPv6 DNS Autodiscovery*, e estas son **fec0:0:0:ffff:1,2,3**.

Se unha empresa non quere usar *DHCPv6*, pode configurar un *DNS* nunha desas direccións IPv6 e xunto cun router IPv6, enviando mensaxes *RA* para que os clientes se autoconfiguren como vimos anteriormente, podería ter a rede funcionando.

3. Descubrimento de equipos na rede.

Un dos grandes problemas á hora de atacar unha rede é ver que equipos están dispoñibles nela, polo que antes de entrar en materia explicando diferentes tipos de ataques, imos ver como podemos conseguir esta información.

Hai múltiples xeitos, un deles, quizais o máis sinxelo é escoitar o tráfico da rede en modo pasivo e ir recoñecendo todas as direccións IPv6 que circulan. O problema con este método é que non sempre podemos acceder a todas as direccións debido a estar en redes con estruturas de *switching*.



6. Captura de tráfico IPv6

Outra aproximación sería escanear o esquema de direccións IPv4 no que estea configurado o equipo atacante e obter a resolución de nomes. A partir dese momento, pódese facer uso do protocolo *LLMNR* para obter as direccións IPv6 e ver cales están en vínculo local.

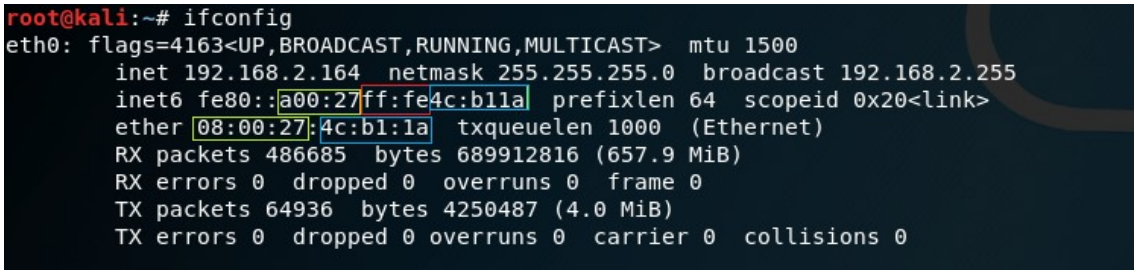
Estes dous esquemas son os que utiliza *Evil FOCA*, ferramenta para facer ataques que utilizaremos posteriormente e veremos como funcionan estes casos, aínda que podemos aplicar moitos máis mecanismos para facer máis efectivo este proceso de descubrimento de rede en IPv6.

En busca dun descubrimento exhaustivo poderíase facer un ***Nmap*** e un descubrimento masivo de todo o segmento fe80::/64, pero sería un consumo alto en tempo e recursos.

Outra posibilidade, sería intentar descubrir as direccións IPv6 de *link local* asignadas a unha dirección *MAC* tendo en conta o estándar orixinal das direccións.
Nese esquema, dos 64 bits asignados a host, os 24 primeiros corresponden cos 24 bits de maior peso da dirección *MAC*, despois énchense coas constantes **FF FE** e logo póñense os 24 bits de menos peso da dirección *MAC*. Polo tanto se algún equipo na rede utiliza este método de xeración de direccións de vínculo local podería ser descuberto facilmente.

Comentar que en algúns equipos, o segundo bit menos significativo do byte máis significativo da dirección MAC é cambiado por un complemento a un, é dicir, se é 1 pasase a 0 e se é 0 pasase a 1.

Esta conversión, podemos vela exemplificada na seguinte imaxe:



7. Descubrimento de dirección link local a través da MAC e do estándar

Vemos como ese **08** (1000), cambiando o seu penúltimo bit por un complemento a 1 quedaría (1010) que é en hexadecimal a letra **A**, como ben vemos na dirección de vínculo local.

Debido a esta curiosidade podemos utilizar o protocolo IPv4 para facer un escaneo por *ARP* e obter as direccións *MAC*, para despois xerar as direccións IPv6 con este método, e posteriormente probalas.

Esta debilidade, presente cando as direccións son xeradas con *SLAAC* e cando nos queremos conectar cun router coñecido por un paquete *RA*, pode evitarse xerando aleatoriamente a parte de host da dirección, como ben se di no RFC 4941.

Evil FOCA tamén incorpora unha opción de descubrimento de rede para localizar equipos que están facendo *routing*, configurando o tráfico a Internet a través dun destes equipos e vendo a ver se funciona.

Outra alternativa sería usar a suite **THC** (*The Hacker Choice*) e a súa ferramenta ***alive6*** que nos respondería coas direccións IPv6 do noso segmento (*alive6 eth0*).

Obviamente hai máis formas para descubrir equipos na nosa rede IPv6 pero non podemos mencionalas todas. Fixemos unha recompilación tanto das máis comúns e usadas como das utilizadas polas ferramentas que usaremos posteriormente neste traballo.

4. Ataques *MitM*

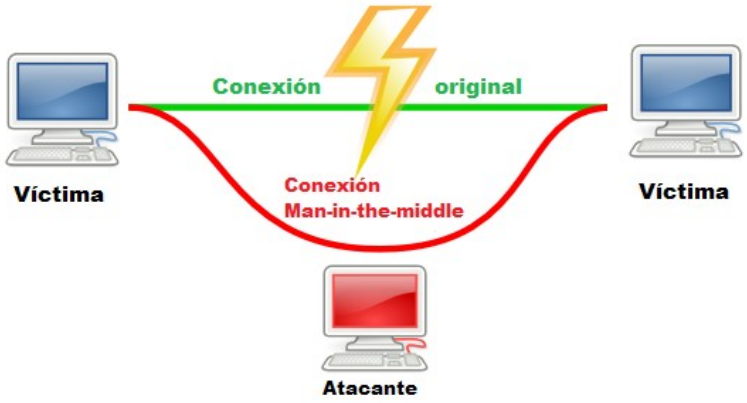
Unha vez visto todo o anterior, podemos empezar coa parte puramente de ataques.

O ataque máis presente neste documento será o *MITM*, ou o ataque *Man In The Middle*. Imos explicar brevemente o seu funcionamento:

Como xa se viu, para descubrir a todos os veciños dunha rede de equipos con IPv6 úsase o protocolo *NDP*. Concretamente, utilízanse dúas mensaxes *ICMPv6*. Un equipo envía unha mensaxe **NS** a unha dirección *Multicast* cando se vai a comunicar cun equipo e o que teña esa dirección IPv6 responde á mensaxe *Multicast* cunha mensaxe *Unicast* **NA** coa súa *MAC*. O receptor da mensaxe NA almacena esta MAC xunto a dirección IPv6 na **táboa de veciños**.

Ata aquí xa o viramos, é a teoría, pero o problema ven que ao igual que en *ARP* en IPv4, un atacante pode enviar unha mensaxe *NA* sen recibir unha mensaxe *NS* e facer que na caché da táboa de veciños se almacene o rexistro. Un ataque *Neighbor Spoofing* para facer *MITM* basearase polo tanto en enviar unha mensaxe *NA* aos dous equipos aos que se quere facer o ataque, poñendo en ambos a dirección IPv6 do outro, e a dirección MAC do atacante.

O ataque polo tanto realizase *spoofeando* a dirección IPv6 de orixe do paquete, para simular ser unha mensaxe que ven do outro equipo vítima pero en ambos casos ponse a dirección *MAC* do atacante, para conseguir que o *switch* de comunicacións faga chegar todas as mensaxes á máquina do home no medio.



8. Exemplo teórico do ataque de tipo MitM

4.1. *Parasite6*

Unha das ferramentas que implementan estes ataques é Parasite6, de *The Hacker's Choice*. Por defecto realiza *man in the middle* entre todos os equipos IPv6 que descobre pola rede.

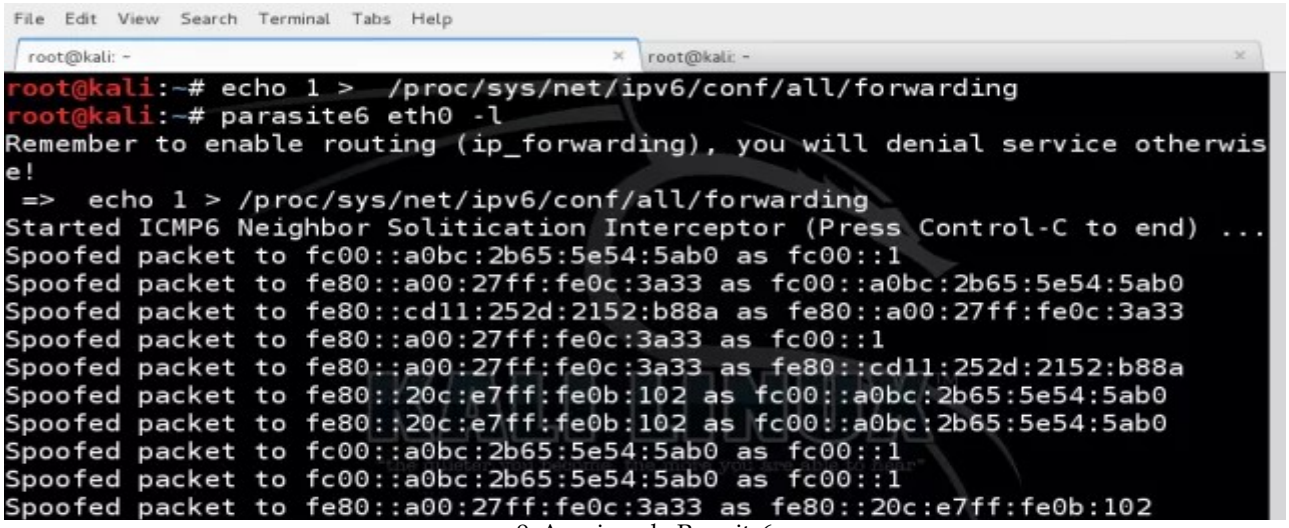
Os pasos para que funcione a configuración por defecto son:

1. Poñer interface ipv6:

   # *ifconfig eth0 inet6 add [ipv6]*

2. Arrancar Parasite6

   # *Parasite6 eth0*



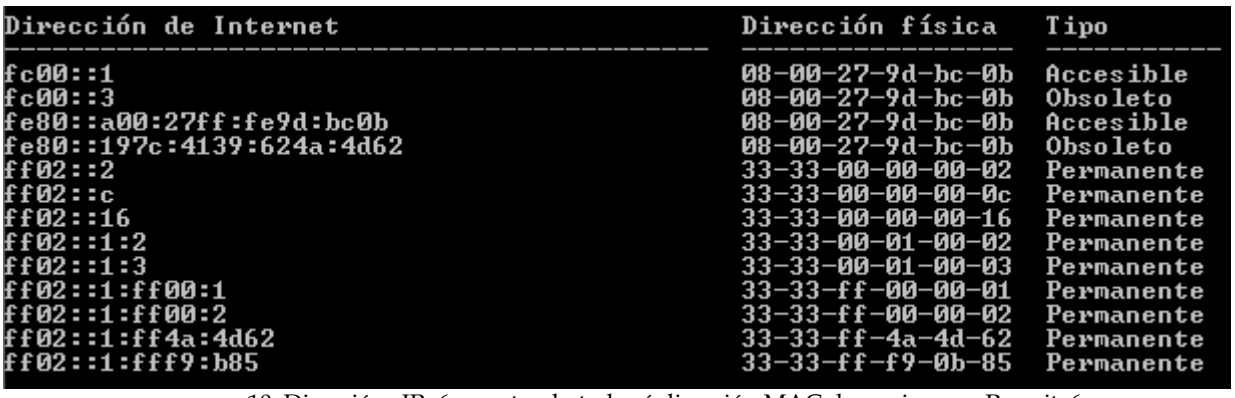
9. Arquivando Parasite6

3. Configurar o enrutamento:

   # *sysctl -w net.ipv6.conf.all.forwarding=1*

4. Activar un *sniffer* (p.e *Wireshark*) para analizar os paquetes

A partires deste momento empezaranse a enviar mensaxes *NA* para facer *MITM* nas direccións ip detectadas e envelenaranse as táboas dos veciños.



Dirección de Internet	Dirección física	Tipo
fc00::1	08-00-27-9d-bc-0b	Accesible
fc00::3	08-00-27-9d-bc-0b	Obsoleto
fe80::a00:27ff:fe9d:bc0b	08-00-27-9d-bc-0b	Accesible
fe80::197c:4139:624a:4d62	08-00-27-9d-bc-0b	Obsoleto
ff02::2	33-33-00-00-00-02	Permanente
ff02::c	33-33-00-00-00-0c	Permanente
ff02::16	33-33-00-00-00-16	Permanente
ff02::1:2	33-33-00-01-00-02	Permanente
ff02::1:3	33-33-00-01-00-03	Permanente
ff02::1:ff00:1	33-33-ff-00-00-01	Permanente
ff02::1:ff00:2	33-33-ff-00-00-02	Permanente
ff02::1:ff4a:4d62	33-33-ff-4a-4d-62	Permanente
ff02::1:fff9:b85	33-33-ff-f9-0b-85	Permanente

10. Direccións IPv6 apuntando todas á dirección MAC do equipo con Parasite6

4.2. Scapy Project

Esta ferramenta é unha potente interface interactiva cuxos obxectivos fundamentais son a manipulación de paquetes.

Un exemplo do seu uso sería crear paquetes falseados de tipo ICMPv6 para que a vítima adquira información de dirección física asociado a unha dirección IPv6 falseada.

Para iso, a través de *Scapy*:

- Para os parámetros de capa 2 introducimos a dirección física da vítima no destino e na orixe a dirección *MAC* do atacante.
- Para os de capa 3 introducimos como dirección de orixe a correspondente a unha das vítimas e a de destino á outra vítima, que deberá coincidir coa dirección física especificada en capa 2 como destino.

Tras configurar o paquete, só queda envialo pola rede co seguinte comando:

 # Sendp(ether/ipv6/na/lla, iface = 'eth0',loop=1, inter=5)

Para completar o ataque só quedaría analizar os paquetes con algún *sniffer*.

4.3. MITM con SLAAC

Dentro das funcionalidades que se aportan na implementación de IPv6, unha delas consiste na posibilidade de realizar unha configuración rápida dun adaptador de rede.

SLAAC *(Stateless Address Autoconfiguration)* é un mecanismo moi cómodo e potente que non ten un equivalente en IPv4, que permite a autoconfiguración dos nodos. Ao levantar a interface de rede, automaticamente configurase unha dirección dun enlace local.

A capacidade de **SLAAC** (*StateLess Address Auto Configuration*) pode ser utilizada por un atacante para configurarse como porta de enlace na rede IPv6 e acceder a todas as comunicacións.

O obxectivo do ataque é facer *MITM* cando un usuario se conecte a Internet a un servidor que non ten soporte IPv6. O atacante configura correctamente o soporte IPv6 para a vítima e busca un entorno no que IPv4 deixe de funcionar. Unha vez desconfigurado IPv4 e configurado IPv6, será necesario facer o cambio da rede IPv6 á rede IPv4 configurando os servizos NAT64 e DNS64 para que o equipo non perda conectividade.

4.4. MITM SLAAC con Evil FOCA

O obxectivo deste ataque é que o cliente navegue por unha web sen direccións IPv6 asociadas a el usando IPv6 na súa rede local, a través dun esquema man in the middle.

A forma máis sinxela de que este ataque funcione é buscar equipos que só teñan soporte IPv4 con servizo *DHCP* para asignar as direccións IPv4.

Primeiro, débese conseguir que o servidor *DHCPv4* non dea ningunha dirección ip nin porta de enlace á vítima facendo un ataque *Rogue DHCPv4* ou *DHCP ACK Injector* que configure ao equipo vítima cunha dirección IPv4 de vínculo local en IPv6 (169.254.X.X) e sen porta de enlace ou facendo *DdoS* ao servidor *DHCPv4*.
En ***Metasploit*** ven o módulo *DNS Exhaustion Attack*, que realiza peticións *DHCP* con distintas direccións *MAC* até que o servidor deixa de responder.

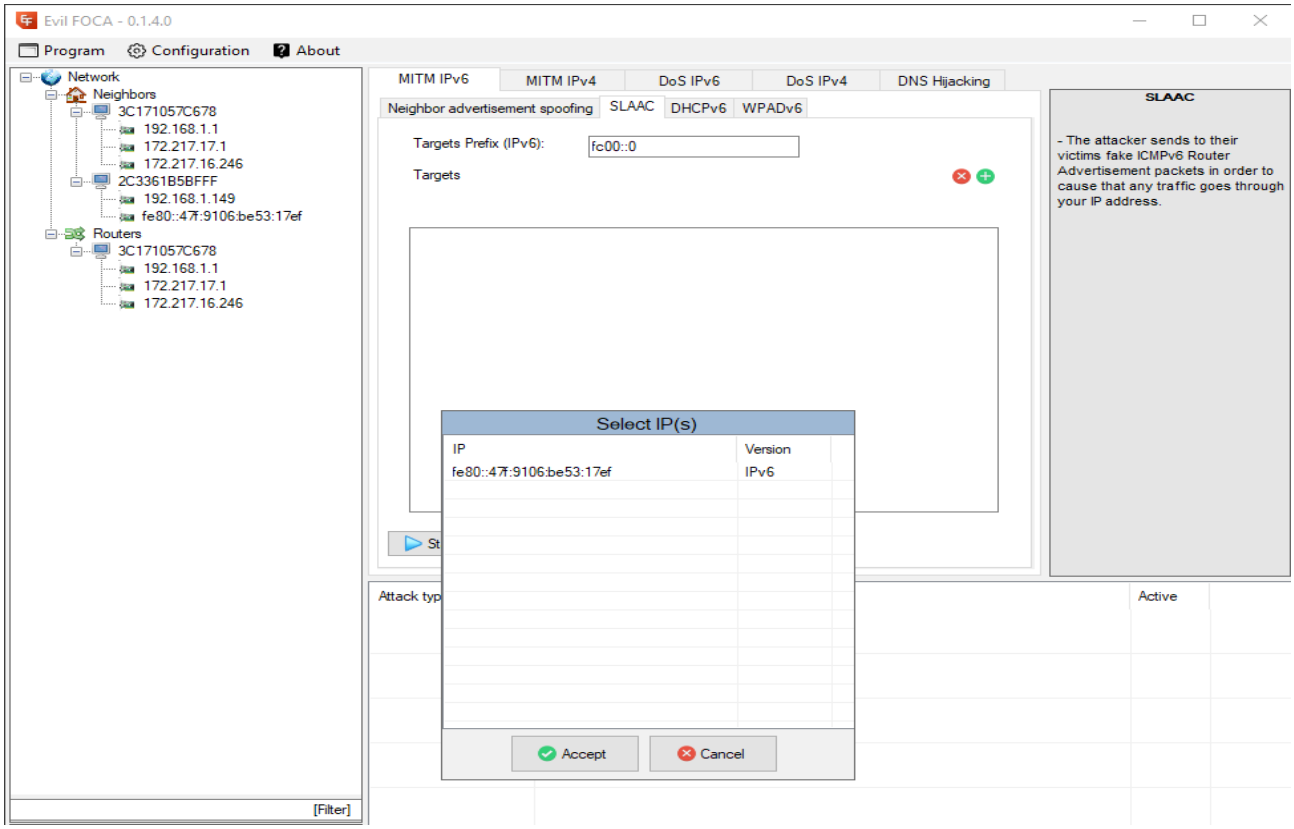
Unha vez realizado, a vítima terá tanto en IPv4 como en IPv6 unha dirección de vínculo, polo que os dous protocolos estarán en igualdade de condicións. Para conseguir que o equipo teña IPv6 funcionando é necesario que teña unha porta de enlace que apunte á dirección IPv6 do atacante. Para iso, envíase un paquete SLAAC que provoca que a vítima poña dirección IPv6 con conectividade con *Evil FOCA* e a porta de enlace apuntando á máquina do atacante.

Os pasos a realizar serían os seguintes:

1. Descubrimento de equipos na rede.

2. No panel de ataque *SLAAC* configurar prefixo de rede necesario para que a vítima se configure a dirección IPv6, que supostamente lle dará saída a Internet por IPv6. Este prefixo é o que supostamente ten o router de conexión configurado, e que a vítima encontrará cando queira saír cara outra rede fóra do segmento local.

3. Lanzar o ataque. A vítima recibirá un paquete *ICMPv6* tipo **RA** (*Router Advertisment)* para que o protocolo *SLAAC* do equipo decida configurarse unha dirección IPv6 dentro dese prefixo de rede para ter conectividade co router.



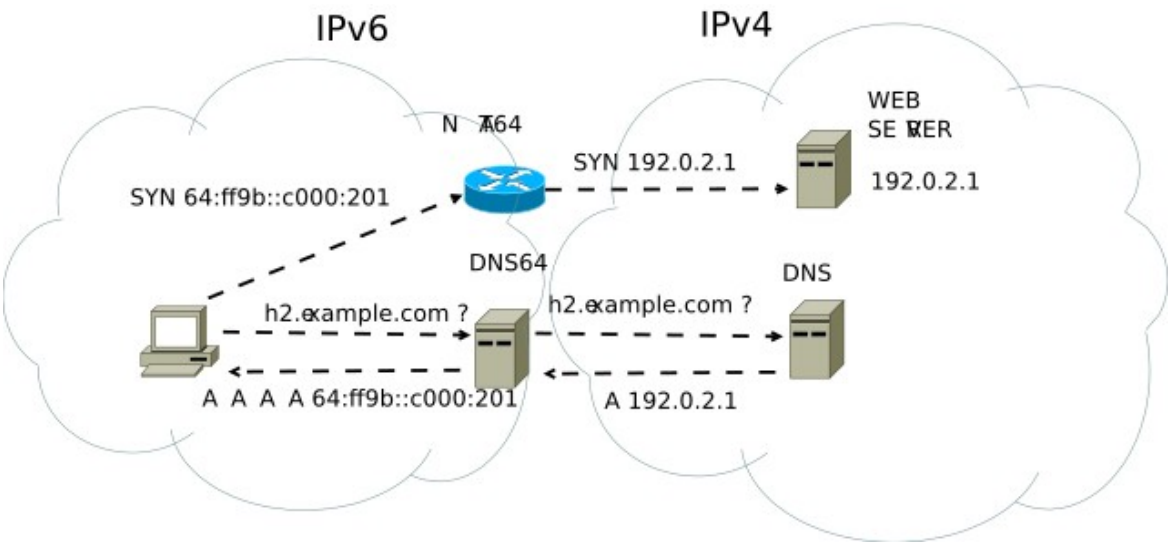
11. Selección de vítimas do ataque SLAAC

Non é necesario configurar o servidor *DNS*, xa que cando unha máquina ten porta de enlace para saír da rede local buscará servidores *DNSv6* nas direccións establecidas polo estándar para os servidores *DNS Autodiscovery*.

As peticións aos servidores *DNS Autodiscovery* pasarán polo atacante que se ocupará de que todo o tráfico de rede sexa procesado para que teña conectividade.

4.4.1 NAT64(Network Address Transalation 6 to 4)

NAT64 é un mecanismo que permite a *hosts* IPv6 comunicarse con servidores IPv4. O servidor NAT64 dispón de polo menos unha dirección IPv4 e un segmento de rede IPv6 de 32-bits.



12. Tanto o servizo DNS64 como NAT64 estarán correndo no man in the middle

1. A vítima envía a unha dirección *DNS Autodiscovery* a petición de resolver un nome de dominio.

2. *Evil FOCA* fai a petición de resolver ese dominio usando IPv4.

3. O servidor *DNSv4* responde coa dirección IPv4.

4. *Evil FOCA* xera unha dirección IPv6 a partires da IPv4 proporcionada e entrégaa á vítima para que faga o resto de peticións.

Unha vez que a vítima ten a dirección IPv6 asociada ao dominio que quería, a petición *HTTP* que fará o navegador será por IPv6. Pode haber certos problemas debido a que algúns routers non soportan IPv6 ou que os propios servidores web non o soportan. Para probar se o soporte para conexións a sitios usando IPv6 no navegador funciona invocar á seguinte URL:

- *http://[2001:db8:81a8:bd1:1111:145a:14a:1001]*

Evil FOCA fará unha implementación do servizo *NAT64* para converterse nun enrutador de tráfico IPv6 cara unha rede IPv4. Ademais, para os equipos con Windows Vista ou superior *Evil FOCA* responde tal e como espera o servizo que detecta a conexión a Internet, para que na máquina da vítima non poda detectar ningunha mensaxe de alerta.

4.5. MITM SLAAC con Radvd & NATPD

Para construír este ataque será necesario inicialmente montar un sistema que se encargará dos encamiñamentos falsos. Dito sistema contará con dous adaptadores de rede, un interno con direccionamento para conectarse coa vítima IPv6 e outro con soporte IPv4 para a conexión externa cara Internet.

A dirección de vínculo local será utilizada para atender solicitudes de descubrimento de router na rede. Xa que o sistema íase encargar dos encamiñamentos será necesario habilitar o *forwarding* de paquetes.

A funcionalidade de atender as solicitudes e proporcionar os valores de configuración axeitados pode establecerse a través do paquete *Radvd*. A continuación amosase a configuración do ficheiro Radvd.conf:

```
interface eth1
{
        AdvSendAlert on;
        AdvOtherConfigFlag on;
        MinRtrAdvInterval 3;
        MaxRtrAdvInterval 10;
        prefix 2001:/64
        {
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr on;
        };
```

12.Configuración Radvd.conf

Ao igual que nos anteriores ataques, debemos configurar o prefixo da rede na que se vai configurar o falso router e dar servizo de encamiñamento cara Internet, para que a vítima se configure de forma automática facendo uso de *SLAAC* unha dirección de ese segmento.

Tamén será necesario interceptar as peticións que se fagan aos servidores do estándar *DNS Autodiscovery* xa que desde un paquete *SLAAC* non se pode configurar as direccións dos servidores *DNS*. Tamén se pode crear un servidor *Rogue DHCPv6* para poñer os valores axeitados.

Unha vez activado o servizo *Radvd*, os clientes con configuración predeterminada do protocolo IPv6, recibirán unha resposta as súas peticións de enrutamentos IPv6 que realicen mediante o protocolo *ICMPv6* usando paquetes **RS** (*Router Solicitation*). Periodicamente tamén xera mensaxes *ICMPv6 RA*.

Para que o ataque sexa efectivo será necesario que o router falso teña configurado *NAT64*. Pode ser realizado a través do servizo *naptd* de distribucións Linux.

O seguinte paso a realizar é configurar o *firewall* con *iptables* e *ip6tables* coas seguintes regras:

- ip6tables -A OUTPUT -p icmpv6 –icmpv6-type 1 -j DROP

-ip6tables -A FORWARD -d 2001:ffff:: -j DROP

-iptables -A INPUT -i lo -j ACCEPT

-iptables -A INPUT -m state –-state ESTABLISHED -j ACCEPT

-iptables -A INPUT -m state –state NEW -p tcp -m tcp –dport 22 -j ACCEPT

-iptables -A INPUT -j DROP

O último paso consistirá en implementar un *proxy DNS*, por exemplo con *totd* (servidor proxy *DNS* pequeno que admite *hosts*/redes IPv6 unicamente que se comunican co mundo IPv4), para a resolución dos rexistros *host* solicitados polas vítimas.

4.6. MITM SLAAC con SuddenSix

SuddenSix é un script de configuración de todo o ataque (*MITM SLAAC*) que está dispoñible no repositorio *https://github.com/Neohapsis/SuddenSix*

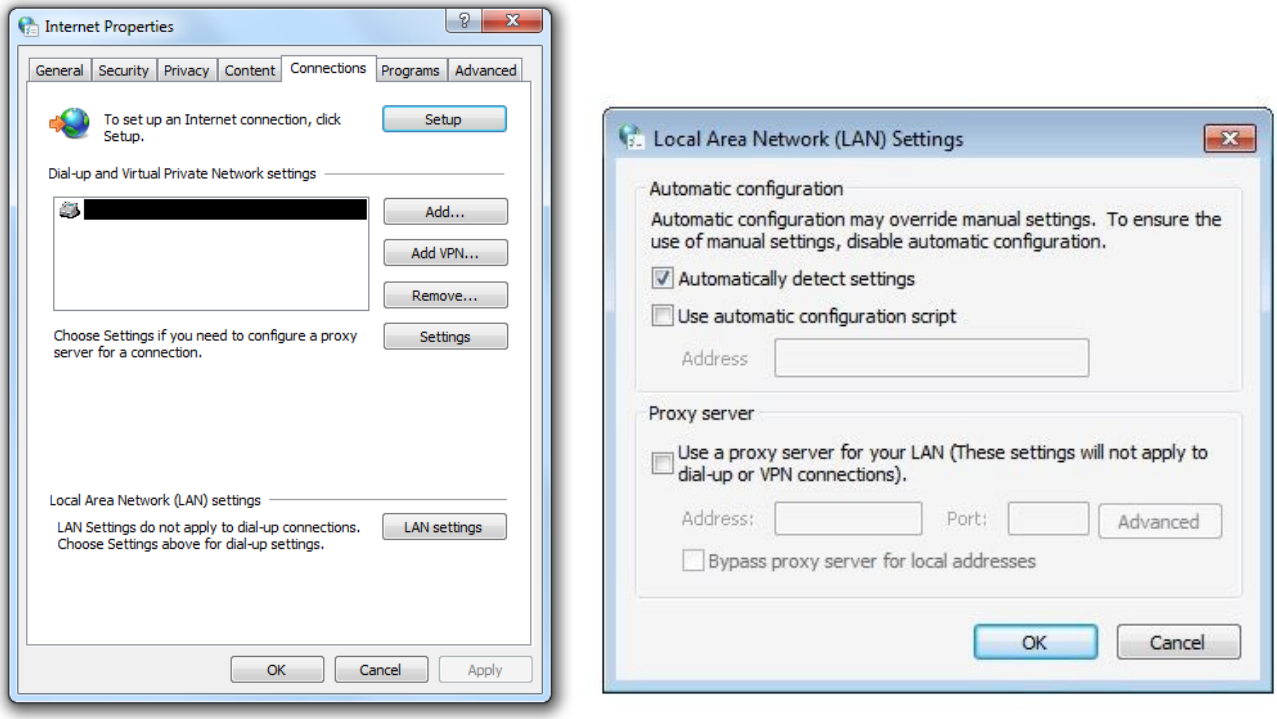
Este script configura todo o ataque *SLAAC* descrito anteriormente e os paquetes software necesarios para o proceso, tales como *sipcalc, tayga, Radvd, wideDHCPv6-server e bind9.*

5. Web Proxy Autodiscovery en IPv6

Este ataque de rede está baseado en facer un man in the middle a través do servizo *Web Proxy Auto-Discovery*.

Os navegadores de Internet veñen configurados para buscar ao servidor *Proxy* (servidor intermediario nas peticións de recursos que realiza un cliente a outro servidor) que lles de acceso a Internet por medio dun servizo que se chama *Web Proxy Auto-Discovery*.

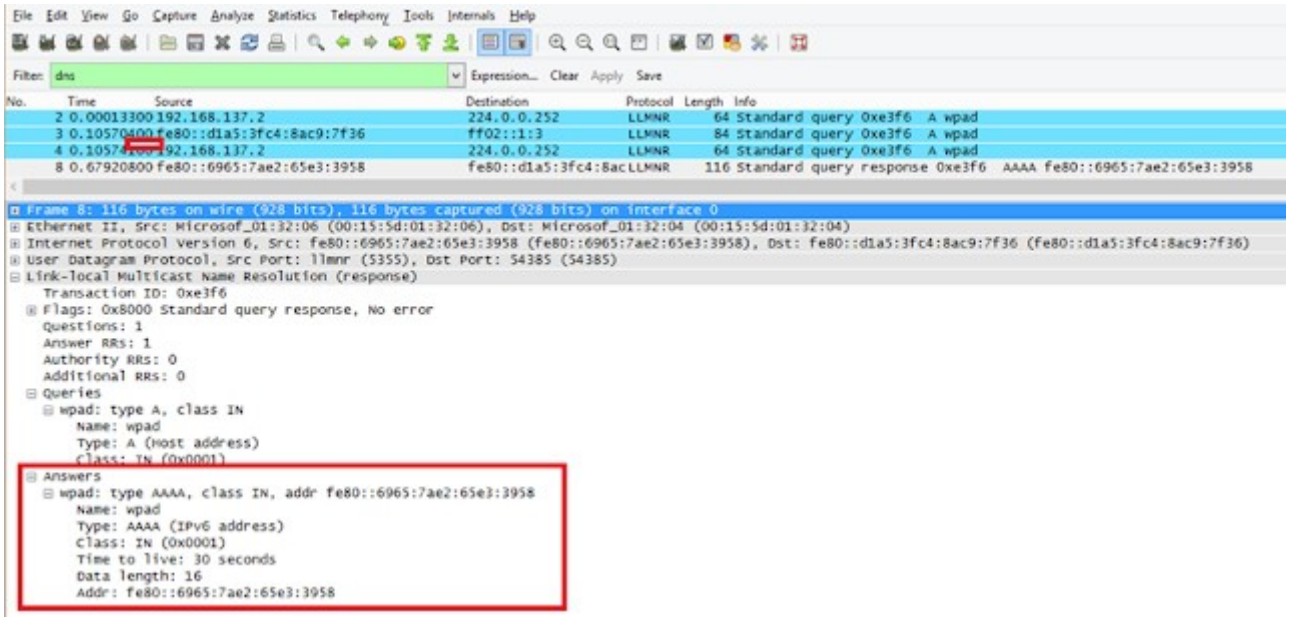
Este servizo de descubrimento basease no uso dun rexistro no servidor *well-known* (explicado no apartado 2.2) creado no servidor *DNS*, e que é chamado **WPAD** (nome que vén de *Web Proxy Auto-Discovery*). Esta configuración está activada por defecto en todos os navegadores e a recomendación é desactivar esta característica se estades nunha rede que non é xestionada por vos, e se o é previr a activación deste servizo nos navegadores se non se vai usar.



13. Propiedades de Internet Explorer onde a opción de buscar automaticamente a configuración de rede está activada

O rexistro **WPAD** é buscado por toda a rede mediante o protocolo **LLMNR** (explicado anteriormente) tanto por IPv4 como por IPv6 pero buscando un rexistro tipo A, é dicir a dirección IPv4 onde se debe conectar o navegador para descubrir ao servidor *Web Proxy*.

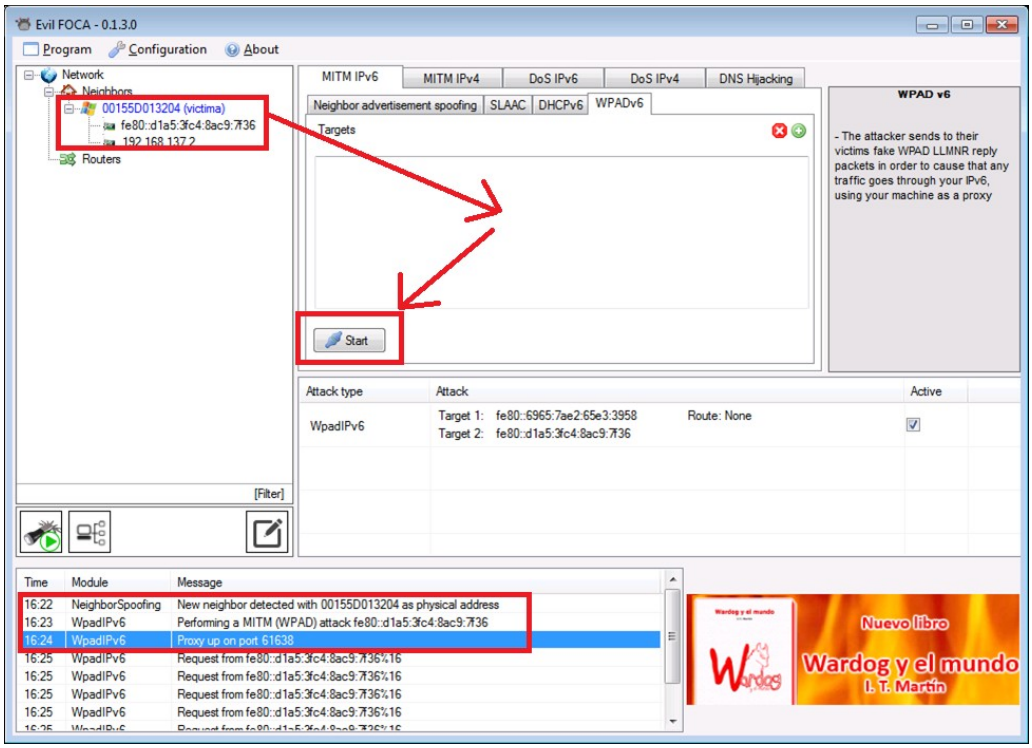
A busca do rexistro WPAD faise por *multicast*, en *Evil FOCA* capturase a petición que vai co destinatario a *Multicast* IPv6 cando o ataque sexa a unha dirección IPv6 ou a IPv4 cando o ataque sexa a unha dirección IPv4 (*Evil FOCA* pode facer ese ataque en ambas), logo de capturar, contestará coa dirección IPv6 (o ataque estámolo facendo en IPv6) asociada a un rexistro de tipo AAAA, pasando dunha petición A a resposta AAAA, por iso é posible que o cliente pida realizar unha confirmación preguntando esta vez polo rexistro AAAA para asegurar (non pasa nada, *Evil FOCA* confirma).



14. Evil Foca contesta cun rexistro AAAA

Agora o cliente sabe a dirección IPv6 do rexistro **WPAD** e xa sabe onde esta o servidor *Web Proxy Auto-Discovery*, conéctase a el para conseguir información do servidor *Web Proxy* (pode ser distinto) solicitando un ficheiro de configuración (*WPAD.PAC*) co que obtén onde está o servidor *Web Proxy* (ficheiro con formato texto plano).

Realizar en *Evil FOCA* este ataque é tan sinxelo coma arrastrar o equipo ao panel de ataque **WPAD** e facer clic en *start*. *Evil FOCA* encárgase de interceptar a petición do rexistro **WPAD** do cliente e de proporcionarlle o *WPAD.PAC* e facer de servidor *Web Proxy HTTP*.



15. Ataque WPAD en IPv6 con Evil FOCA.

A vítima navegará sen notar nada raro pero facéndoo a través do *Proxy* configurado que estará facendo un *man in the middle* e interceptando todo o tráfico de rede.

Recoméndase utilizar **IDS** (*Sistema de Detección de Intrusos*) para descubrir quen realiza peticións/respostas ao protocolo de *Web Proxy Auto-Discovery*, así como cambiar a configuración dos navegadores web como xa comentamos.

6. Conexións HTTPs en ataques MITM na rede IPv6

Unha vez no medio e tendo acceso ao tráfico do cliente, podemos interceptalo con diferentes ferramentas, tendo un comportamento diferente en cada unha. No caso de usar ***Cain***, esta ferramenta fai unha copia do certificado dixital para enviarlle ao cliente un Falso-*CA*.

Algunhas ferramentas non validan que o certificado estea emitido para este sitio en concreto, que estea caducado ou emitido por unha entidade certificadora válida. Se non é así ou ben non funciona a conexión ou ben se obtén unha alerta de seguridade. Se o usuario acepta o certificado estará enviando datos cifrados ao atacante que os descifrará, lerá e manipulará e volverá enviar ao servidor cifrados nunha conexión *HTTPs* co certificado orixinal do servidor *Web*. *Evil FOCA* non fai uso de esta técnica que temos que facer nós de xeito manual.

SSLSniff, fai algo parecido pero aproveitándose dun bug nos clientes que non verifican as *BasicConstrains* do certificado que reciben. Utilízase un certificado válido pero que non ten validez para crear novos certificados,é dicir, o atacante saca un certificado dixital para un servidor A e despois fai uso del para crear certificados falsos para outros sitios pero asinados por A, se temos o *bug* que comentabamos (non se comproba que o certificado non ten autoridade para crear certificados) podería tomar o certificado como bo. Isto é un fallo que tiveron todos os navegadores nalgún momento.

6.1. Stripping de HTTPs: Bridging HTTPs(IPv4)-HTTP(IPv6)

No caso de ferramentas como *SSL Sniff* ou *Evil FOCA* o ataque basease en facer que a vítima navegue a través do atacante só con *HTTP* e será o atacante o que navegue en *HTTPs* co servidor real. No caso de que o servidor intente facer unha redirección a *HTTPs* cando se faga unha petición *HTTP* será o atacante o que manterá ao cliente sempre en *HTTP*.

Unha vez que se obteña a páxina de resultados é importante que as *cookies* de sesión, que virán marcadas co *flag secure* para que non funcionen sobre *HTTP* sexan xestionadas polo atacante e xerar unha *cookie* falsa para enviar ao cliente (sen ter o *flag* mencionado) permitíndolle a navegación e sen notar tampouco o servidor unha manipulación da *cookie*.

Isto non é algo que sexa necesario sempre, xa que aínda que moitos servidores que envían a mensaxe de redirección a *HTTPs* seguen escoitando por *HTTP*, polo que permiten autenticación por *HTTP*.

Para conseguir máis peticións de inicio desde o cliente con *HTTP*, *Evil FOCA* filtra tamén os resultados da busca que ofrece *google*, é dicir, cando a vítima busque ***HTTP://www.google.es*** (a través dunha conexión atacada) e o servidor lle faga a redirección a *HTTPs*, interceptarase igualmente e entregaralle a conexión en *HTTP* (***Bridging HTTPs-HTTP***), para así garantir que se poderá seguir accedendo ao tráfico intermedio.

Despois cando o usuario busque algo, todos os resultados que veñan como *HTTPs* serán substituídos por *HTTP* e os *redirect JavaScript* serán eliminados tamén para ter un engano completo.

Se a vítima envía o contrasinal para facer *login* nunha conexión controlada por *Evil FOCA*, está quedará comprometida ao ser enviada en texto plano.

Cando se está sendo afectado por un *man in the middle* pouco se pode facer. Unha opción sería cifrar o tráfico de rede contra un punto de conexión seguro facendo uso por exemplo dun servizo de ***VPN*** (*Virtual Private Network*), facendo uso de protocolos de cifrado coma ***L2TP*** (*Layer 2 Tunneling Protocol*) ou ***SSTP*** (*Secure Socket Tunneling Protocol*).

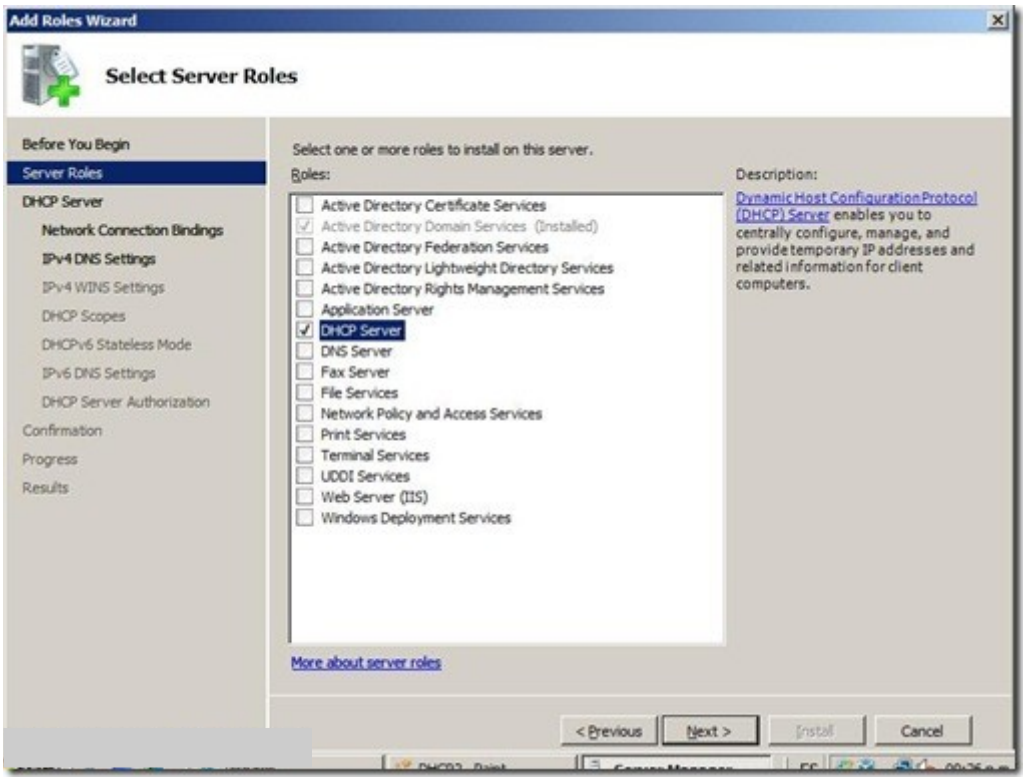
Outra opción, facer uso dun *plug-in* que obrigue a navegar sempre vía *HTTPs*, usar *Certificate Pinning* en aplicacións instaladas no equipo ou dos sitios web visitados habitualmente para evitar así sufrir ese *bug* de *Basic Constrains* ou o ataque coa axuda dunha entidade certificadora intermedia maliciosa. Tamén é recomendable non entrar aos sitios facendo *clic* (nin nos resultados de *google*), senón intentar buscar coa sua dirección ***URL*** (*https://www...*) e non confiar nos *redirect* de páxinas *HTTPs*.

7. Montaxe Servidor Rogue DHCPv6

Outro xeito de atacar unha rede IPv6 que hai que ter en conta consiste en montar un servidor ***DHCPv6*** (*Dynamic Host Configuration Protocol*) nunha rede na que non haxa control da aparición de servidores *Rogue DHCP* para IPv6 que é algo moito máis común do que debería. Para facelo é tan doado como utilizar calquera servidor *DHCP* en Linux ou Windows e configuralo en función do entorno.

7.1. Montaxe servidor DHCPv6 en Windows Server

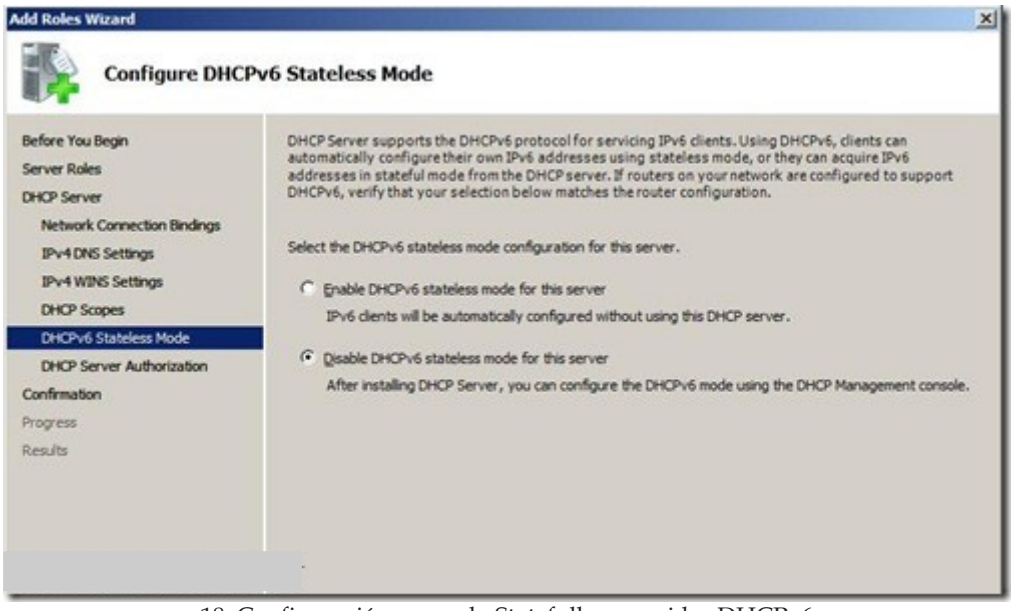
Para instalar un servidor *DHCPv6* en Windows Server é tan sinxelo como irse ás opcións de Roles de Servidor e engadir o rol de servidor *DHCP:*



17. Engadir Rol de Servidor DHCP en Windows Server.

Unha vez escollido haberá que configurar as opcións básicas do servizo e neste caso temos que configurar que se vai facer uso de configuración de direccións IP en modo *Statefull*. O modo *Stateless* permite que sexan os routers de conexión os que configuren aos equipos mentres no modo *Statefull* os servidores *DHCPv6* configurarán de xeito máis permanente todas as direccións que se asignen aos clientes, establecendo períodos de vida para cada unha.

Escollemos modo *Statefull* para configurar direccións IPv6 dos clientes á vez que outras propiedades que se poden asignar ás conexións de rede.



18. Configuración en modo Statefull no servidor DHCPv6

Para continuar temos que configurar un ámbito de asignación de direccións IPv6 no que se definirán as distintas propiedades de rede a establecer nos clientes. Abrimos a ferramenta de administración do servidor *DHCP* e seleccionamos a opción de Novo Ámbito (*New Scope*) no nodo IPv6 do servidor (*clic* dereito en IPv6)

Nas opcións de creación de ámbito IPv6 só hai que configurar as direccións que van ser asignadas, excluídas e o tempo que se vai asignar cada dirección.

Desde un punto de vista do atacante a asignación de tempos debería realizarse igual sen ter en conta se a solicitude do cliente é temporal ou non, tendo en conta que o tempo sexa necesario para realizar o ataque con éxito. Se o tempo remata só se producira un proceso de renovación polo que non sería un problema crítico.

Unha vez acabemos de configurar o ámbito estará listo para empezar a asignar direccións IPv6 pola rede e poderemos configurar o volume de opcións que se queiran configurar (Dentro do ámbito creado – *Scope Options*).

Estas opcións ao igual que no servizo *DHCP* IPv4, permiten configurar servidores ***NISv6***, ***SIPv6***, ***DNSv6***, etc. co que o atacante podería realizar calquera ***MitM*** o ***Phishing*** a calquera vítima configurada por este servidor *Rogue*.

Estas opcións tamén poden ser configuradas a nivel de servidor afectando a todos os ámbitos configurados no equipo co rol de *DHCP*.

Se existe un servidor asignando direccións IPv4 aos clientes por medio do protocolo *DHCPv4*, existe a ferramenta que ven dentro do *framework* de explotación de vulnerabilidades ***Metasploit***, chamada ***DHCP Exhaustion*** que, realizando un ataque de forza bruta con peticións falsas, fai por acabar con todo o ámbito de direccións que tivese configurado.

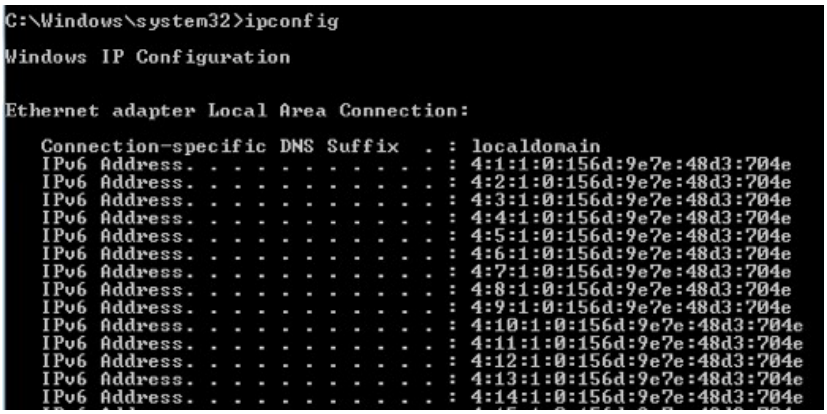
7.2. Montaxe do servidor Rogue DHCPv6 con Evil FOCA

En *Evil FOCA* tamén se pode configurar os clientes da rede cun *fake DHCPv6*, o cal ten moitas menos posibilidades ca ter un servidor montado pero pode ser válido para algúns casos.
Temos que configurar o valor que se lle quere asignar ao servidor *DNS* da configuración IPv6 para os equipos seleccionados dunha rede ademais dunha dirección IPv6 cun rango establecido (similar a un ámbito).

8. Outros ataques e ferramentas para IPv6

8.1. DOS RA Storm

Probablemente un dos ataques máis coñecidos pola súa aparición nos medios de comunicación é o ataque de denegación de servizo *DoS* feito a maquinas Windows mediante o envío de múltiples mensaxes **RA** (*Router Advertisement*) do protocolo *SLAAC* no que se configura un grande número de direccións IPv6 que fai que as máquinas Windows se bloqueen.
Este é un ataque coñecido dende o 2010 afectando a outros dispositivos IPv6 ademais das maquinas Windows por ese *flood* de mensaxes **RA**. Este é un ataque tamén implementado dentro das opcións de *Evil FOCA*.



19. Flood de paquetes RA a unha máquina Windows

8.2. The Hacker Choice: The IPv6 Attack Toolkit

É unha ferramenta centrada en ataques e vulnerabilidades a redes IPv6, incorporado en ***BackTrack, Kali***... Dispón das seguintes funcionalidades ou ataques:

- ***Parasite6:*** Realiza ataques MITM con listas de veciños spoofeado.

- ***alive6:*** Escáner que detecta equipos con IPv6.

- ***dnsdict:*** Realiza ataques de dicionario a *DNSv6.*

- ***fake_router6:*** Anúnciase un equipo como router IPv6 coa maior prioridade.

- ***redir6:*** Redirixe o tráfico con *ICMPv6 spoofing* (*man in the middle*).

- ***toobig6:*** Permite reducir os valores *MTU.*

- ***detect-new-ip6:*** Detecta novos equipos IPv6 que se unen á rede e lle pasa un script ao novo equipo.

- ***dos-new-ip6:*** detecta novos equipos e indícalle que a súa IPv6 está repetida na rede xerando un ataque DoS.

- ***trace6:*** Rápido *traceroute6* con soporte para *ICMP6 echo request* e *TCP-SYN*.

- ***flood_router6**:* ataque *flood RA (Router Advertisement).*

- ***flood_advertise6**:* Ataque *flood NA (Neighbor Advertisement).*

- ***fuzz_ip6**: Fuzzer* para IPv6.

- ***implementation6**:* comproba a implementación de IPv6.

- ***implementation6d**:* demo que escoita *implementation6* detrás dun *FW.*

- ***fake_mld6**:* Anúnciase como un grupo *Multicast* na rede.

- ***fake_mld26**:* Igual para *MLDv2*.

- ***fake_mldrouter6**:* Envía mensaxes falsos de routers *MLD*.

- ***fake_m IPv6**:* Rouba unha dirección IP móbil se *IPSEC* non pide autenticación.

- ***fake_advertiser6**:* Anuncia ao usuario na rede.

- ***smurf6**:* local *smurfer.*

- ***rsmurf6:*** *remote smurfer* (só para Linux).

- ***exploit6**:* lanza *exploits* IPv6 coñecidos contra un equipo.

- ***thcping6**:* envía un paquete *ping6* personalizado.

- ***sendpees6**:* Envía un paquete especial de ***NS*** (*Neighbor Solicitation*) que fai entrar a CPU en *thrashing* realizando cálculos criptográficos.

Pódese atopar información de todos estes ataques en
https://github.com/vanhauser-thc/thc-ipv6

8.3. Topera 2

Topera é un escáner de portos para ser utilizado co protocolo con IPv6 ao estilo do popular ***nmap***, moi sinxelo de utilizar e con funcionalidade limitada. É capaz de realizar un escaneo de rede eludindo os sistemas de detección de intrusos baseados en ***Snort*** que si detectan os realizados por ***nmap***.

O truco detrás de *Topera 2* é que para facer o escaneo utiliza *extensión headers* que engade nas cabeceiras IPv6 para que cando se realice un escaneo de portos *SYN* este non sexa detectado por software de seguridade de rede.

Ademais desta funcionalidade, posúe unha implementación do ataque *Apache Slowloris* que permite facer ataques de *DoS* a servidores *Apache* non fortificados, xerando conexións con duración infinita e que producen un consumo descomunal de recursos dentro do servidor web.

8.4. IPv6 Toolkit

IPv6 Toolkit é un dos *frameworks* máis populares na auditoría de IPv6, nel inclúense as seguintes ferramentas:

- ***addr6**:* Ferramenta para a análise e manipulación de direccións IPv6.
- ***flow6**:* Ferramenta para facer análises de seguridade IPv6 *Flow Label*.
- ***frag6**:* Ferramenta para facer probas de fragmentación en IPv6.
- ***icmp6**:* Ferramenta para facer ataques baseados en mensaxes de erro *ICMPv6.*
- ***jumbo6**:* Realiza probas en paquetes de tipo *IPv6 Jumbograms.*
- ***na6**:* Ferramentas para enviar mensaxes *Neighbor Advertisement*.
- ***ni6**:* Ferramenta para facer probas con mensaxes *ICMPv6 Node Information.*
- ***ns6**:* Ferramenta para enviar mensaxes *Neighbor Solicitation.*
- ***ra6**:* Ferramenta para enviar mensaxes *Router Advertisement.*
- ***rd6**:* Ferramenta para enviar mensaxes *ICMPv6 Redirect.*
- ***scan6**:* Scanner de direccións IPv6.
- ***tcp6**:* Ataques con fragmentos TCP en IPv6.

Bibliografía

-*Ataques en redes de datos IPv4 e IPv6.* — {Juan Luís García Rambla}

-*IPv6: Principios e implementación.* — {Archier, Jean-Paul}

-*Tema IPV6.* — {Apuntes materia de Redes (614111502)}

-*Ataques a IPv6: THC-IPv6* — {Juan Rodrigo Cantorna}

- https://openwebinars.net/blog/ipv6-autoconfiguracion-slaac/

- http://www.dillema.net/software/totd.html

- http://www.elladodelmal.com/2013/08/man-in-middle-con-web-proxy-auto.html

- *https://github.com/vanhauser-thc/thc-ipv6*