UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO
INE5632 – PROJETOS II

Avaliação dos Aspectos de Segurança em Um Cenário de Transição IPv4/IPv6

Eduardo de Mello Garcia

Florianópolis – SC

Orientadora:
Profa. Dra. Carla Merkle Westphall

Coorientador:

Guilherme Eliseu Rhoden, Msc.

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
SISTEMAS DE INFORMAÇÃO

AVALIAÇÃO DOS ASPECTOS DE SEGURANÇA EM UM CENÁRIO DE TRANSIÇÃO IPv4/IPv6

EDUARDO DE MELLO GARCIA

Orientadora:
Profª. Drª. Carla Merkle Westphall

Coorientador:
Guilherme Eliseu Rhoden, Msc.

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação na Universidade Federal de Santa Catarina

Florianópolis – SC
2014

AVALIAÇÃO DOS ASPECTOS DE SEGURANÇA EM UM CENÁRIO DE TRANSIÇÃO IPv4/IPv6

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de “Bacharel em Sistemas de Informação”, e aprovado em sua forma final pelo Curso de Bacharelado em Sistemas de Informação

Florianópolis - Santa Catarina, 03 de Julho 2014.

_____

Prof. Dr. Renato Cislaghi
Coordenador

Banca Examinadora:

_____

M.Sc. Profª Drª Carla Merkle Westphall
Orientadora

_____

M. Sc. Guilherme Eliseu Rhoden
Coorientador

_____

M. Sc. Guilherme Arthur Gerônimo

_____

M. Sc. Edison Tadeu Lopes Melo

_____

M. Sc. Jorge Werner

Agradecimentos

À minha família, a meus amigos, aos orientares Carla Merkle Westphall e Guilherme Eliseu Rhoden e ao apoio toda a equipe do PoP-SC/RNP.

Sumário

Lista de figuras

Lista de tabelas

Resumo

Com o passar dos anos, houve um grande aumento no número de dispositivos acessando a Internet. Isto ocorreu devido a uma série de fatores como o barateamento dos computadores pessoais e, mais recentemente, o surgimento de outros dispositivos pessoais capazes de acessar a Internet, como *smartphones* e *tablets*.

O aumento do número de dispositivos conectados fez crescer a demanda por endereços da versão 4 do Protocolo Internet (*Internet Protocol version 4* – IPv4) de forma que o esgotamento destes endereços foi se aproximando cada vez mais rápido.

Para substituir o IPv4, foi desenvolvida a versão 6 do Protocolo Internet (IPv6). Como não foi possível simplesmente desativar o IPv4 e implantar o IPv6, foram criadas técnicas de transição que visavam migrar toda a internet do IPv4 para o IPv6 mantendo a infraestrutura em produção. A princípio, a transição IPv4/IPv6 deveria ocorrer conforme um cronograma pré-estabelecido. No entanto, a transição não ocorreu conforme o previsto. Hoje, a utilização de endereços IPv4 segue aumentando e ainda não há IPV6 disponível em toda a Internet. Por este motivo, a transição IPv4/IPv6 se torna cada vez mais urgente e esta urgência traz à tona um fator crucial: a segurança. A urgência na transição, a criação de diversas técnicas de transição, o pouco tempo de depuração do protocolo IPv6 entre outros fatores fazem surgir novos riscos aos usuários, uma vez que há mais falhas de segurança para se explorar.

Este trabalho tem como objetivo avaliar os aspectos de segurança em técnicas de transição IPv4/IPv6, contribuindo para a identificação e minimização de ameaças e uma transição mais segura. Este trabalho inclui revisão bibliográfica acerca do IPv6, das técnicas de transição e de vulnerabilidades explorando o IPv6 e técnicas de transição. Estão inclusos também testes e experimentos envolvendos ataques a um conjunto de vulnerabilidades estudadas e levantamento de formas de detecção e defesa.

Neste trabalho, foram elaborados dois cenários de testes para experimentos, explorando duas das principais técnicas de transição IPv4/IPv6 utilizadas: pilha dupla e NAT64. No cenário pilha dupla foram realizados ataques de Negação de Serviço (*Denial of Service - DoS)* para novos endereços IPv6, ataques de *DoS* e *man-in-the-middle* a partir do anúncio de um roteador falso e ataques de *flooding* de mensagens *Neighbor Advertisement* e *Router Advertisement*. No cenário NAT64, foram repetidos estes ataques e, adicionalmente, foi realizado um ataque ao mecanismo DNS64 através do anúncio de um servidor DNS falso. Para cada um dos ataques realizados nos cenários, foram testadas formas e ferramentas para a detecção e foram estudadas estratégias de

defesa.

Palavras-chave: transição IPv4/IPv6, IPv6, segurança em redes

Abstract

Over the years, there was a large increase on the number of devices accessing the Internet. This ocurred due to a number of factors like the lower cost of personal computers and, more recently, the emergence other devices capable of accessing the Internet, like smartphones e tablets.

The increase of connected devices also increased de demand for Internet Protocol version 4 (IPv4) addresses so that the depletion of these addresses is approaching fast.

To replace IPv4, the version 6 of the Internet Protocol (IPv6) was designed. As is was not possible to just disable IPv4 and deploy IPv6, transition techniques were created to migrate the entire Internet from IPv4 to IPv6 keeping the whole infrastructure functional. In principle, the IPv4/IPv6 should take place following a predefined schedule. However, the transiction did not occur as expected. Nowadays, IPv4 utilization keeps increasing and there is no IPv6 available in all the Internet. For that reason IPv4/IPv6 transition becomes increasingly urgent and this urgency brings up a crucial factor: security. The transition urgency, the creation of various transition techniques, the little debugging time of IPv6 among other factors bring new risks to users as there are more security flaws to exploit.

The objective of this work is to evaluate IPv4/IPv6 transition security aspects, ccontributing to the identification and mitigation of threats and to a safer transition. This work includes bibliographic research about IPv6, the transition techniques and vulnerabilities exploiting IPv6 and the transition techniques. This work also includes experiments with attacks to a set of studied vulnerabilities research of forms of detection and deffense.

In this work, two test scenarios were prepared for experimentation, exploiting two of the

main IPv4/IPv6 transition techniques: dual stack and NAT64. In the dual stack scenario, Denial of Service (DoS) to new IPv6 addresses, DoS and man-in-the-middle through a fake router advertisement and Neighbor Advertisement and Router Advertisement flooding attacks were performed. In the NAT64 scenario, these attacks were repeated and, additionally, an attack to the DNS64 mechanism through the advertisement of a fake DNS server was performed. For each attack performed on the scenarios, detection forms and tools were tested and deffense strategies were studied.

Keywords: IPv4/IPv6 transition, IPv6, networking security

Introdução

O IPv6 é a nova versão do Protocolo Internet (*Internet Protocol* – IP) e foi projetado para substituir a versão 4 deste protocolo (IPv4) tendo em vista o esgotamento do espaço de endereçamento do mesmo. Entre as principais mudanças do Ipv4 para o Ipv6, destaca-se o maior espaço de endereçamento, simplificação do cabeçalho, maior suporte a extensões e controle de fluxo, incluindo suporte a extensões de segurança (DEERING; HINDEN, 2013).

Com o esgotamento de endereços IPv4 cada vez mais próximo, cresce a urgência da transição para o IPv6. Para este fim, foram criadas diversas técnicas de transição baseadas em pilha dupla, tunelamento e tradução. Com a implantação do IPv6, surgem novas demandas de segurança. Estas demandas também estão relacionadas às técnicas de transição, uma vez que podem ser exploradas vulnerabilidades inerentes às mesmas. Um problema enfrentado nesta área é como implantar uma solução para a transição IPv4/IPv6 considerando as demandas de segurança do protocolo IPv6 e da(s) técnica(s) de transição envolvida(s).

A transição do protocolo IPv4 para o IPv6 é inevitável e vem avançando gradativamente. Com o passar do tempo esta transição vai se tornando cada vez mais urgente, o que levou ao desenvolvimento de diversas técnicas de transição (MOREIRAS et al., 2013). No entanto, a urgência na transição para o IPv6, causada pelo eminente esgotamento de endereços IPv4, não é o único problema envolvendo a transição IPv4/IPv6.

Com a implantação do IPv6, surgem novos problemas de segurança que podem representar novos riscos para o usuário. Alguns destes problemas não estão relacionados apenas ao IPv6, mas à própria transição, como é o caso da técnica de pilha-dupla. Nesta técnica de transição, uma configuração mal feita ou que não dê a devida atenção à coexistência de ambos os protocolos traz riscos para a rede. Em técnicas de tunelamento como o 6to4, o encapsulamento de um protocolo em outro poder ser explorado para mascaramento de pacotes (*spoofing*) (SANKARAN, 2013).

Motivação

A implantação do IPv6 é essencial para o funcionamento da Internet e, especialmente, para a sua expansão no futuro. No entanto, IPv4 e IPv6 são imcompatíveis entre si, logo, não é possível simplesmente desativar o IPv4 e implantar o IPv6. Será necessário que ambas as versões do protocolo IP coexistam até que o uso do IPv6 na Internet seja pleno. Em função da necessidade da

coexistência entre Ipv4 e Ipv6, foram criadas técnicas de transição.

No entanto, a transição do IPv4 para o IPv6 acontece lentamente e, com a implantação do IPv6 e das técnicas de transição, surgem novas vulnerabiliades que podem ser exploradas por atacantes. Além disto, em função da lenta transição do IPv4 para o IPv6, é menor a preocupação com a segurança em cenários de transição IPv4/IPv6, assim como em cenários puramente Ipv6. Portanto, tem-se como motivação deste trabalho, a realização de uma transição IPv4/IPv6 segura e visando, futuramente, medidas eficientes e eficazes de mitigação das vulnerabilidades específicas do IPv6 uma vez que a transição esteja completa.

Objetivo geral

O objetivo geral deste trabalho é realizar um estudo aprofundado das técnicas de transição IPv4/IPv6 tendo em vista os problemas de segurança das mesmas. A partir deste estudo, pretende-se elaborar e implantar um cenário de testes e avaliar a segurança nas técnicas de transição a partir da realização de testes no cenário elaborado. Para a montagem deste cenário, pretende-se utilizar uma infraestrutura de máquinas virtuais e blocos de endereçamento IPv4 e IPv6 disponibilizada pelo PoP-SC/RNP.

Objetivos específicos

São objetivos específicos deste trabalho:

- Revisão bibliográfica do protocolo IPv6, focando na relação entre transição IPv4/IPv6 e segurança;
- Realização de testes por meio da infraestrutura implantada e avaliação dos resultados;
- Avaliação da possibilidade de encontrar possíveis soluções de segurança tendo em vista os testes realizados.

Organização do texto

No Capítulo 1 são apresentados os conceitos básicos do Protocolo IPv6, abrangendo o funcionamento do protocolo, o cabeçalho IPv6, destacando as mudanças em relação ao cabeçalho IPv4, endereçamento e tipos de endereços IPv6, bem como funcionalidades básicas do protocolo IPv6.

O Capítulo 2 foca na transição do IPv4 para o IPv6. Neste capítulo são abordadas as técnicas de transição de pilha dupla, tunelamento 6over4, GRE e *tunnel broker*, tradução NAT64 e, brevemente, a técnica DS-Lite.

O Capítulo 3 trata da parte de segurança. Neste capítulo serão abordados aspectos de segurança específicos do IPv6 e ICMPv6. Serão também apresentados problemas de segurança comuns entre IPv4 e IPv6, além de aspectos de segurança de cada uma das técnicas de transição abordadas no Capítulo 2.

O Capítulo 4 apresenta os cenários de testes elaborados e os experimentos realizados. Neste capítulo, é descrita a topologia da rede de cada cenário de testes, bem como as configurações e a infraestrutura utilizadas para os mesmos, além das principais ferramentas utilizadas neste trabalho. Neste capítulo, são também detalhadas a forma como os experimentos foram realizados, as ferramentas utilizadas, as máquinas envolvidas e os objetivos de cada experimento

O Capítulo 5 apresenta os resultados obtidos. Neste capítulo, são mostradas a execução dos experimentos na prática bem como os resultados dos mesmos nos cenários de testes. Além disto, são apresentadas formas de detecção dos ataques e possíveis formas de defesa.

O capítulo 6 apresenta as conclusões obtidas a partir deste trabalho, bem como propostas de trabalhos futuros.

1 O Protocolo Ipv6

No início da década de 1990, já visualizando o crescente uso da Internet especialmente por dispositivos particulares e a escassez de endereços IP que esta crescente demanda acarretaria, a *Internet Engineering Task Force* (IETF) começou a trabalhar em um novo protocolo da Internet. Diversas propostas foram elaboradas visando resolver o problema da escassez de endereços, bem como outros problemas técnicos do IPv4. A partir de uma combinação e de ajustes em algumas propostas, foi definida a versão 6 do Protocolo Internet – IPv6 (TANEMBAUM, 2003). Neste capítulo, será apresentada a fundamentação teórica sobre o protocolo IPv6.

1.1 Conceitos básicos

Do ponto de vista da arquitetura TCP/IP, o IPv6 é, conceitualmente, bastante semelhante ao IPv4. No entanto, foram realizadas mudanças em sua estrutura para proporcinar aprimoramentos no protocolo. Primeiramente, o IPv6 possui um espaço de endereçamento de 128 bits, contra 32 bits do IPv4. Este aumento no espaço de endereçamento, além de possibilitar a conexão de um número muito maior de dispositivos na rede, permite níveis mais específicos de agregação de endereços e a implementação de mecanismos de autoconfiguração, que possibilitam a obtenção de endereços IPv6 globais automaticamente sem o uso de *Dynamic Host Configuration Protocol* (DHCP) (MOREIRAS et al., 2013).

A simplificação no formato do cabeçalho é outro aprimoramento do IPv6 em relação ao IPv4. Alguns campos do cabeçalho IPv4 foram removidos ou tornados opcionais. Desta forma, é possível reduzir o custo de processamento dos pacotes pelos roteadores (MOREIRAS et al., 2013) e, portanto, melhorar a vazão em suas interfaces de rede (TANEMBAUM, 2003).

Além da simplificação do cabeçalho, o IPv6 suporta cabeçalhos de extensão. Os cabeçalhos de extensão comportam as opções, que não fazem mais parte do cabeçalho base. Com cabeçalhos de extensão, o roteamento se torna mais eficaz, há limitações menos rigorosas para o tamanho e a quantidade de opções e há também uma maior flexibilidade para a introdução de novas opções no futuro (MOREIRAS et al., 2013).

Dando maior atenção à qualidade de serviço, o IPv6 permite a identificação de um fluxo de dados. Desta forma, é possível determinar se pacotes pertencentes a um determinado fluxo devem ser tratados de forma diferenciada (MOREIRAS et al., 2013). Como exemplo de tráfego diferenciado, pode-se citar aplicações de tempo real (DEERING; HINDEN, 2013).

Com relação à segurança, foram especificados cabeçalhos de extensão capazes de fornecer

mecanismos de autenticação, além de garantir a integridade e confidencialidade dos dados transmitidos (MOREIRAS et al.2013).

1.2 O cabeçalho IPv6

Conforme mencionado na seção 1.1, o cabeçalho IPv6 sofreu algumas modificações de forma a simplificá-lo. O número de campos foi reduzido de 12 para 8 e o tamanho foi fixado em 40 *bytes*. Além disto, a adição de cabeçalhos de extensão tornou mais eficiente o cabeçalho base do IPv6, uma vez que os cabeçalhos de extensão não precisam ser processados por roteadores intermediários (MOREIRAS et al., 2013).

Dentre as mudanças do cabeçalho IPv6 em relação ao IPv4, pode-se destacar a remoção de 6 campos do cabeçalho IPv4, devido à inutilização ou reimplementação dos mesmos. A Figura 1 indica os campos removidos do cabeçalho IPv4. O primeiro campo removido foi o campo *Header Length* (HL). Este campo deixou de ser necessário uma vez que o tamanho do cabeçalho foi fixado. Em seguida, foram removidos os campos *Identification*, *Flags*, *Fragment Offset* e *Options + Padding*. Estes campos passaram a ter suas informações indicadas em cabeçalhos de extensão apropriados. Por fim, o campo *checksum* foi descartado para deixar o protocolo mais eficiente, uma vez que outras validações são realizadas por protocolos das camadas superiores (MOREIRAS et al., 2013).

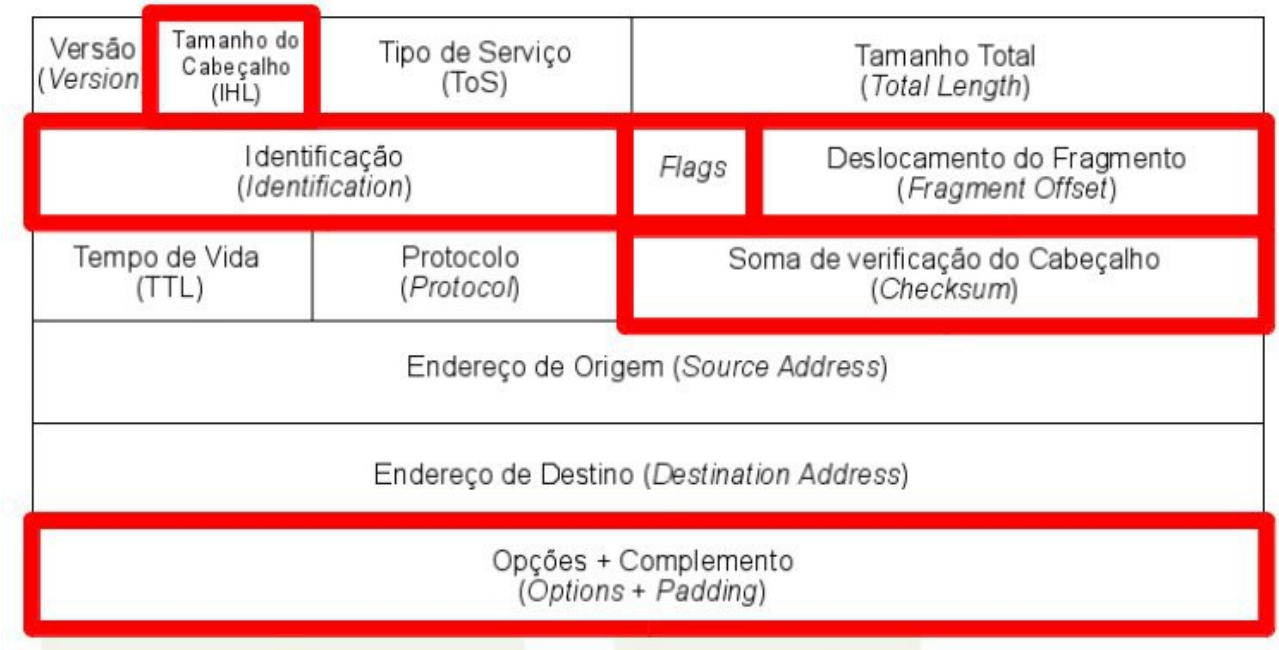

Figura 1: Campos do cabeçalho IPv4 removidos no IPv6. Fonte: (MOREIRAS et al., 2013)

Os campos *Type of Service (ToS), Total Length, Time to Live (TTL)* e *Protocol* do cabeçalho IPv4 foram renomeados, respectivamente, para *Traffic Class, Payload Length, Hop Limit* e *Next Header* no IPv6. Houve reposicionamento destes campos dentro do Cabeçalho para, assim como a renomeação, agilizar o processamento. Além disto, foi adicionado o campo *Flow Label* para viabilizar o funcionamento de um mecanismo a mais de Qualidade de Serviço (*Quality of Service* – QoS). Por fim, os campos *Version, Source Address* e *Destination Address* foram mantidos, tendo apenas seus tamanhos alterados. O cabeçalho IPv6 é mostrado na Figura 2.

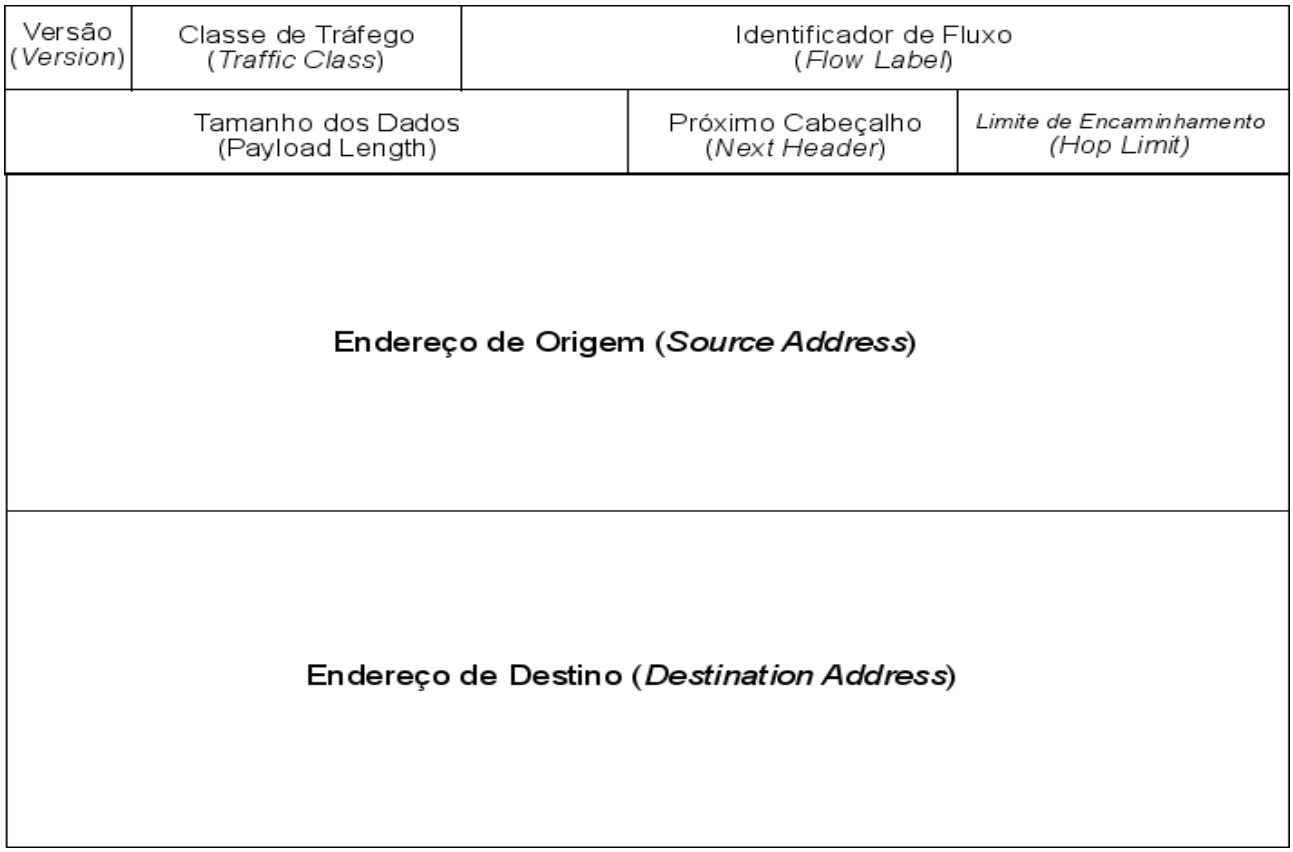

Figura 2: Cabeçalho IPv6. Fonte: (MOREIRAS et al., 2013)

Os campos do cabeçalho IPv6, conforme mostrados na Figura 2, são:

- **Version (4 bits) –** A versão do protocolo. Neste caso, o valor é sempre 6.
- **Traffic Class (8 bits) –** Identifica os pacotes por classe de serviço ou prioridade. Este campo provê as mesmas funcionalidades do campo *Tos* do IPv4, isto é, priorizar determinados tipos de pacotes.
- **Flow Label (20 bits) –** Identifica pacotes de um mesmo fluxo de comunicação (MOREIRAS et al, 2013.). Pacotes pertencentes a um mesmo fluxo são tratados de maneira similar pelos roteadores (DAS,2013).
- **Payload Length (16 bits) –** Indica o tamanho, em bytes, apenas dos dados enviados no pacote.
- **Next Header (8 bits) –** Indica o cabeçalho de extensão que segue o cabeçalho atual. Cabeçalhos de extensão serão detalhados na seção 1.3.
- **Hop Limit (8 bits) –** Indica o número máximo de nós na rede (geralmente roteadores) pelo qual o pacote pode ser encaminhado. O valor deste campo é decrementado em 1 a cada nó que encaminha o pacote.
- **Source Address (128 bits) –** O endereço de origem do pacote.
- **Destination Address (128 bits) –** O endereço de destino do pacote.

1.3 Cabeçalhos de extensão

No IPv6, informações opcionais são tratadas em cabeçalhos separados do cabeçalho base, chamados de cabeçalhos de extensão. Estes cabeçalhos localizam-se entre o cabeçalho base e o cabeçalho da camada de transporte e não possuem quantidade ou tamanho fixo. Caso existam múltiplos cabeçalhos de extensão em um mesmo pacote, estes cabeçalhos são adicionados em série, formando uma cadeia de cabeçalhos (MOREIRAS et al., 2013). A Figura 3 exemplifica o encadeamento de cabeçalhos de extensão. Nas especificações do IPv6, foram definidos seis cabeçalhos de extensão:

- **Hop-by-Hop –** Carrega informações que precisam ser processadas por todos os nós. Identificado pelo valor 00 no campo *Next Header.*
- **Routing –** Carrega uma lista de nós pelos quais o pacote deve passar antes de chegar ao destino. Identificado pelo valor 43 no campo *Next Header.*
- **Fragmentation –** Indica que o pacote é maior que a MTU do enlace e deve ser

fragmentado. Identificado pelo valor 44 no campo *Next Header.* No IPv6, a fragmentação de pacotes é realizada apenas na origem e a reagrupação, no destino final, diferente do IPv4, onde os pacotes podem ser fragmentados no caminho entre origem e destino.

- **Destination Options –** Carrega informações opcionais que devem ser processadas apenas pelo nó de destino. Identificado pelo valor 60 no campo *Next* Header (DEERING;HINDEN, 2013).
- **Authentication Header e Encapsulating Security Payload –** Fazem parte do cabeçalho do IPsec. Identificados, respectivamente, pelos valores 51 e 52 do campo *Next Header* (MOREIRAS et al., 2013).

.

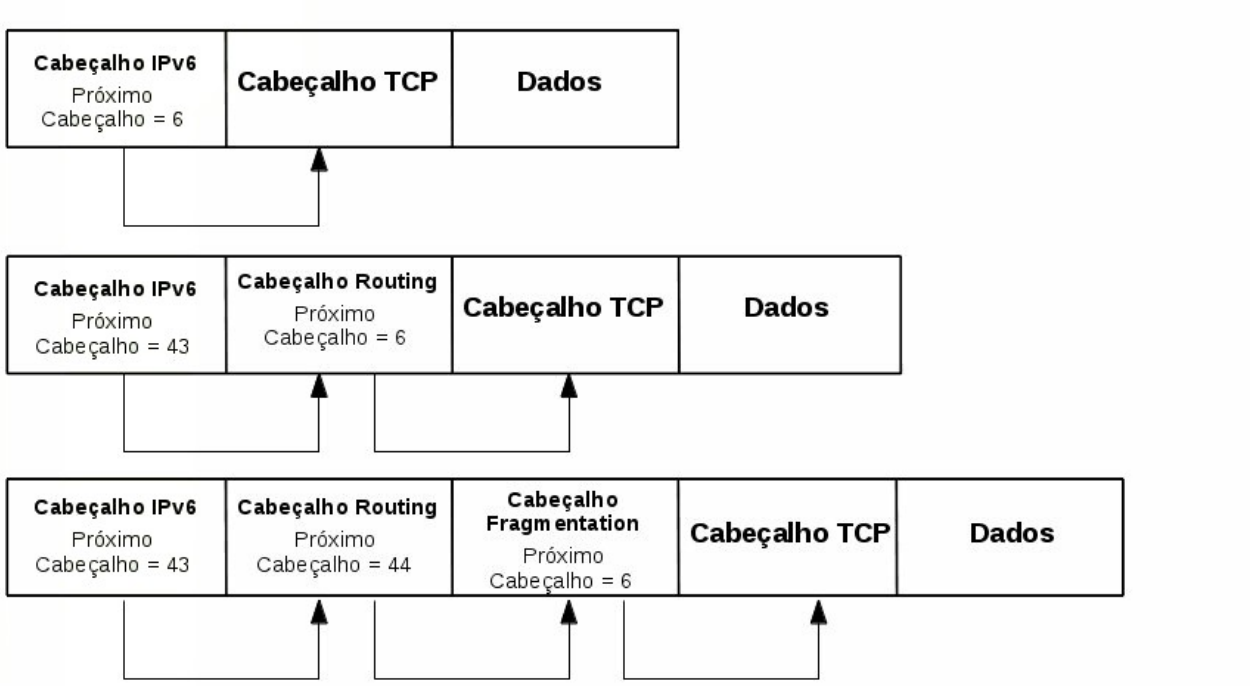

Figura 3: Formação de cadeias de cabeçalhos de extensão. Fonte: (MOREIRAS et al, 2013)

1.4 Endereçamento

A maior motivação para o desenvolvimento do IPv6 foi a necessidade de um maior espaço de endereçamento devido à iminente escassez de endereços IPv4. No IPv6, o espaço de endereçamento é de 128 bits, o que significa 2^{128} endereços possíveis. Este espaço de endereçamento representa cerca de $7{,}9\times10^{28}$ vezes a quantidade de endereços possíveis no IPv4, que possui um espaço de endereçamento de 32 bits e, portanto, 2^{32} endereços (MOREIRAS et al., 2013).

Os endereços IPv6 são representados por 8 grupos de 16 bits escritos em até 4 dígitos hexadecimais e separados por ":" (dois pontos), como nos exemplos em (1) e (2):

- `ABCD:EF01:2345:6789:ABCD:EF01:2345:6789 (1)`
- `2001:0DB8:0000:0000:0008:0800:200C:417A (2)`

Existem alguns mecanismos de abreviação que podem facilitar a representação dos endereços IPv6. Zeros à esquerda podem ser omitidos em cada um dos grupos, desta forma, o endereço 2001:0DB8:0000:0000:0008:0800:200C:417A, mostrado em (2), ficaria:

- `2001:DB8:0:0:8:800:200C:417A (3)`

Além disto, quando há uma grande sequência de zeros, esta sequência pode ser substituída pela notação "::". Esta forma de abreviação, no entanto, pode ser utilizada apenas uma vez e é importante ressaltar que ela se aplica apenas a sequências de zeros constituindo um ou mais grupos de 16 bits (HINDEN; DEERING, 2013). Aplicando esta forma de abreviação ao endereço 2001:DB8:0:0:8:800:200C:417A, mostrado em (3), o resultado seria:

- `2001:DB8::8:800:200C:417A (4)`

Existem três tipos de endereços definidos no IPv6: unicast, anycast e multicast.

1.4.1 Endereços unicast

Um endereço unicast é um endereço que identifica uma única interface. Desta forma, um pacote enviado a um endereço unicast será entregue a apenas uma interface. No IPv6, há diferentes tipos de endereços unicast, quais sejam:

- **Global unicast –** É um endereço globalmente roteável a acessível na Internet, assim como os endereços IPv4 públicos. Um endereço deste tipo é composto por um prefixo de roteamento global, que identifica o tamanho do bloco atribuído a uma rede; um identificador de sub-rede, utilizado para identificar um enlace em uma rede; e um identificador de

interface, que deve identificar de forma única uma interface de rede em um enlace. A estrutura destes endereços foi projetada de forma que os 64 bits mais à esquerda sejam utilizados para a identificação da sub-rede, e os 64 bits mais à direita, para a identificação da interface. Portanto, uma sub-rede IPv6 possui, tipicamente, o tamanho de prefixo /64. Atualmente, o prefixo reservado para a atribuição de endereços unicast globais é o 2000::/3. O formato de um endereço unicast é mostrado na Figura 4.

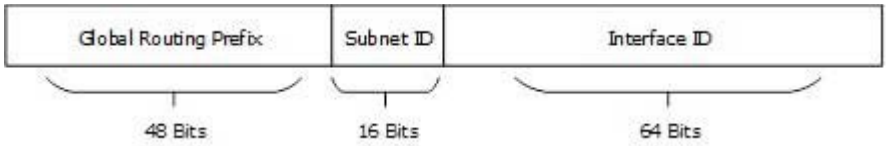

Figura 4: Endereço unicast global. Fonte: (TUTORIALSPOINT, 2013).

- **Link local –** Este tipo de endereço é utilizado apenas no enlace onde a interface de rede que o possui está conectada e é atribuído automaticamente juntando o prefixo reservado FE80::/64 a um identificador de interface. Estes endereços possuem escopo local, ou seja, não são roteáveis na Internet. A Figura 5 mostra o formato de um endereço link local. Os primeiros 64 bits representam o prefixo FE80::/64.

Figura 5: Endereço link-local. Fonte: (TUTORIALSPOINT, 2013).

- **Unique local address (ULA) –** É um endereço com grande probabilidade de ser globalmente único. Um endereço ULA é utilizado para comunicações locais, geralmente dentro de um enlace ou conjunto de enlaces. Este tipo de endereço não deve, portanto, ser roteado na Internet. O formato de um ULA é mostrado na Figura 6. Um endereço ULA é composto por:
  - Um prefixo, sendo reservado para este tipo de endereço o prefixo FC00::/7;
  - Um *flag* local – quando este *flag* tem valor 1, o prefixo é atribuído localmente. Se for 0, o prefixo é atrubído por uma organização central ainda não definida;
  - Um identificador global de 40 bits, utilizado para criar um prefixo globalmente único, gerado de forma pseudo-randômica.
  - Um identificador de interface de 64 bits (MOREIRAS et al., 2013).

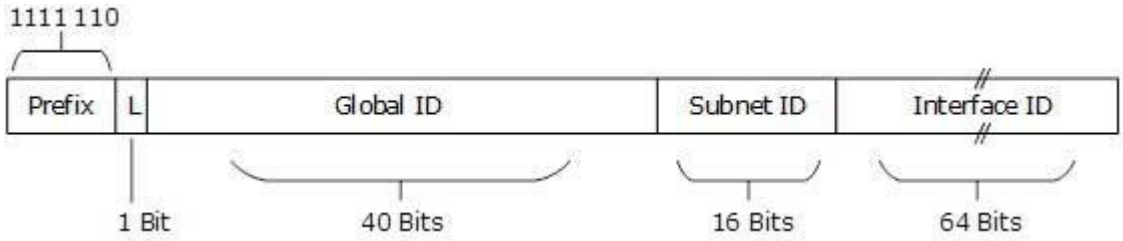

Figura 6: Unique Local Address - ULA. Fonte: (TUTORIALSPOINT, 2013)

1.4.2 Identificadores de interface

Em endereços IPv6 unicast, são utilizados identificadores de interface para a identificação de interfaces de rede dentro de um enlace. Estes identificadores devem ser únicos dentro de uma sub-rede e seu tamanho é, geralmente, 64 bits. É possível utilizar um mesmo identificador de interface para diferentes interfaces em um mesmo nó, desde que as interfaces estejam associadas a sub-redes diferentes. É possível também que um mesmo identificador de interface seja utilizado em diferentes nós dentro de um enlace, no entanto, isto não é recomendado. Existem diferentes formas de configurar um identificador de interface. O mais comum é que ele seja gerado a partir do endereço físico, ou endereço MAC, da interface (HINDEN; DEERING, 2013). Quando a interface possui um endereço MAC de 64 bits (padrão IEEE EUI-64), basta complementar o sétimo bit mais à esquerda. Caso a interface possua um endereço MAC de 48 bits (padrão IEEE 802), o primeiro passo é adicionar a sequência FF-FE entre o terceiro e o quarto *byte*. Após isto, é invertido o sétimo bit mais à esquerda (MOREIRAS et al., 2013). Este processo é demonstrado na Figura 7.

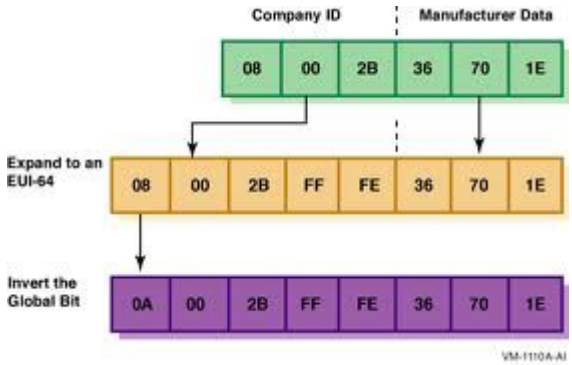

Figura 7: Identificador de interface gerado a partir de um endereço MAC de 48 bits. Fonte: (HP,2013)

1.4.3 Endereços especiais

Existem endereços IPv6 especiais utilizados para fins específicos:

- **Endereço não-especifidado –** É representado pelo endereço 0:0:0:0:0:0:0:0 ou ::0 (equivalente ao endereço IPv4 0.0.0.0). Este endereço é utilizado para indicar a ausência de endereço, não podendo ser atribuído a nenhum nó e nem utilizado como endereço de destino em nenhum pacote. Este endereço pode, no entanto, ser utilizado como endereço de origem em um pacote enviado a outro nó em um processo de atribuição de endereços.
- **Endereço de *loopback* –** É representado pelo endereço unicast ::1. Assim como o endereço IPv4 127.0.0.1, este endereço é utilizado para referenciar a própria máquina, não podendo ser atribuído a nenhum nó, utilizado como endereço de origem em um pacote IPv6 e nem roteado na Internet.
- **Endereço IPv4 mapeado –** Para a representação deste tipo de endereço é utilizado o formato ::FFFF:wxyz, onde wxyz são os bits do endereço IPv4 representados em dígitos decimais. O endereço 192.168.2.1, por exemplo, mapeado em um endereço IPv6, ficaria ::FFFF:192.168.2.1. Este tipo de endereço é utilizado em técnicas de transição.

Existem também faixas de endereços utilizadas para fins específicos, como a faixa 2002::/16, utilizada para a técnica de transição 6to4, e a faixa 2001:db8::/32, utilizado para representar endereços IPv6 em textos e documentações (MOREIRAS et al., 2013).

1.4.4 Endereços anycast

Um endereço anycast é atribuído a mais de uma interface, tipicamente de nós diferentes, com a propriedade de que um pacote enviado a um endereço anycast é enviado à interface mais próxima que contém este endereço, de acordo com a medida de distância do protocolo de roteamento. Endereços anycast são alocados a partir do espaço de endereçamento unicast e não há diferença sintática entre os dois tipos de endereços. Quando um endereço unicast é atribuído a mais de uma interface, ele se torna, portanto, um endereço anycast. No entanto, os nós que receberem este endereço anycast devem ser explicitamente configurados para indicar que possuem um endereço do tipo anycast (HINDEN; DEERING, 2013).

Um esquema de endereçamento anycast pode ser utilizado para a descoberta de serviços na

rede, como servidores DNS e *proxies* HTTP, garantindo a redundância destes serviços. Outros possíveis usos do endereçamento anycast é o balanceamento de carga entre múltiplos *hosts* e roteadores que provém o mesmo serviço e a localização de roteadores que forneçam acesso a uma sub-rede. É importante ressaltar que endereços anycast não podem ser utilizados como endereços de origem em pacotes IPv6.

1.4.5 Endereços multicast

Endereços multicast são utilizados para identificar um grupo de interfaces, sendo que cada interface pode pertencer a mais de um grupo. Pacotes enviados para um endereço multicast são entregues a todas as interfaces que compõe o grupo. O multicast já era suportado no IPv4, porém, de forma opcional. Já no IPv6, o suporte a multicast é obrigatório. O multicast funciona de forma similar ao broadcast, a diferença é que no broadcast os pacotes são entregues a todos os *hosts* da rede, sem exceção, enquanto no multicast os pacotes são entregues a um grupo de *hosts*, o que não necessariamente quer dizer todos. Através da utilização do multicast, a entrega de apenas uma cópia de dados a todos os *hosts* receptores pode reduzir a utilização de recurso da rede e otimizar a entrega de dados a estes *hosts*.

Os endereços multicast não devem ser usados como endereços de origem em um pacote. Endereços multicast derivam do bloco FF00::/8. Existem endereços multicast permanentes, alguns destes endereços são mostrados na Figura 8. O prefixo FF identifica um endereço multicast e é precedido por quatro *flags* e um valor de quatro bits que define o escopo do multicast (MOREIRAS et al., 2013).

Endereço	Escopo	Descrição
FF01::1	Interface	Todas as interfaces (*all-nodes*)
FF01::2	Interface	Todos os roteadores (*all-routers*)
FF02::1	Enlace	Todos os nós (*all-nodes*)
FF02::2	Enlace	Todos os roteadores (*all-routers*)
FF02::5	Enlace	Roteadores OSFP
FF02::6	Enlace	Roteadores OSPF designados
FF02::9	Enlace	Roteadores RIP
FF02::D	Enlace	Roteadores PIM
FF02::1:2	Enlace	Agentes DHCP
FF02::1:FFXX:XXXX	Enlace	*Solicited-node*
FF05::2	Site	Todos os roteadores (*all-routers*)
FF05::1:3	Site	Servidores DHCP em um site
FF05::1:4	Site	Agentes DHCP em um site
FF0X::101	Variado	NTP (*Network Time Protocol*)

Figura 8: Endereços multicast permanentes.
Fonte: (MOREIRAS et al., 2013)

1.5 Funcionalidades básicas do IPv6

As funcionalidades básicas do IPv6 estão associadas, principalmente, a dois protocolos: o *Internet Control Message Protocol version 6* (ICMPv6) e o *Neighbor Discovery Protocol* (NDP). Nesta seção, serão detalhados estes protocolos, bem como as funcionalidades associadas aos mesmos.

1.5.1 ICMPv6

O ICMPv6 é uma nova versão do ICMPv4. Esta nova versão incorpora as funcionalidades do ICMPv4 e algumas mudanças. Algumas das funções mais básicas do IPv6 estão associadas ao ICMPv6, portanto, este protocolo deve ser implementado por todos os nós IPv6 da Internet (CONTA; DEERING; GUPTA, 2013).

Além das funções já realizadas anteriormente pelo ICMPv4, o ICMPv6 desempenha uma série de novas funções. Uma das mudanças do ICMPv6 em relação à versão 4 é o fato do ICMPv6 agregar funções de três diferentes protocolos que, no IPv4, atuam isoladamente: o *Address Resolution Protocol* (ARP), responsável por associar endereços físicos a endereços lógicos, o *Reverse Address Resolution Protocol* (RARP), que faz o inverso do ARP, e o *Internet Group Management Protocol* (IGMP), responsável pelo gerenciamento de grupos de multicast. Estes protocolos deixam de existir no IPv6, uma vez que suas funções são desempenhadas pelo ICMPv6. Outra importante diferença do ICMPv6 para o ICMPv4 é o uso do ICMPv6 pelos seguintes protocolos e funcionalidades:

- **Multicast Listener Discovery (MLD) –** Atua no gerenciamento de grupos multicast;
- **Neighbor Discovery Protocol(NDP) –** Identifica características de vizinhos na rede. Este protocolo será detalhado na seção 1.5.2;
- **Path MTU Discovery –** Atua na descoberta da menor MTU entre dois nós;
- **Mobility Support –** Responsável pelo gerenciamento de endereços de origem dos hosts dinamicamente;

- **Autoconfiguração Stateless –** Permite a obtenção de endereços IPv6 globais sem o uso de DHCP (MOREIRAS et al., 2013).

Os pacotes ICMPv6 foram divididos em duas classes: mensagens de erro e de informação. O cabeçalho ICMPv6 possui uma estrutura simples, baseada em quatro campos básicos:

- **Type (8 bits) –** Especifica o tipo da mensagem e, portanto, o formato do corpo da mesma;
- **Code (8 bits) –** Apresenta informações adicionais sobre o motivo da mensagem;
- **Checksum (16 bits) –** Utilizado para verificar a integridade do cabeçalho ICMPv6 e de parte do cabeçalho IPv6;
- **Data (tamanho variável) –** Os dados relativos ao tipo da mensagem. O tamanho deste campo depende da mensagem, de forma que o tamanho máximo que este campo pode ter é igual tamanho de *Maximum Transmission Unit* (MTU) mínimo do IPv6 (MOREIRAS et al., 2013).

O formato do cabeçalho ICMPv6 é mostrado na Figura 9.

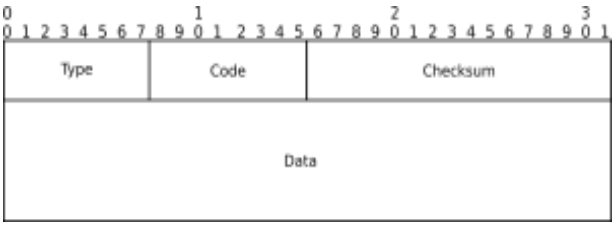

Figura 9: Cabeçalho ICMPv6. Fonte: (MOREIRAS et al., 2013)

1.5.2 NDP

O NDP foi definido para resolver problemas relacionados à comunicação entre nós vizinhos em uma rede. Para isto, o NDP atua sobre dois aspectos fundamentais do IPv6: autoconfiguração de nós

e transmissão de pacotes. No caso da autoconfiguração de endereços, o NDP fornece suporte para as seguintes funcionalidades:

- **Parameter Discovery –** Como um nó descobre informações sobre o enlace (como a MTU) ou sobre a Internet (como o limite de saltos (*hop limit*));
- **Address Autoconfiguration –** Mecanismo para viabilizar a autoconfiguração *stateless.*
- **Duplicate Address Detection –** Maneira como um nó descobre se o endereço que deseja atribuir a uma interface já está sendo usado por outro nó.

O funcionamento dos mecanismos de autoconfiguração de endereços será detalhado na seção 1.5.3. No caso da transmissão de pacotes, o NDP fornece suporte às seguintes funcionalidades:

- **Router Discovery –** Permite aos *hosts* a descoberta de roteadores na rede local, com a finalidade de determinar rotas padrão.
- **Prefix Discovery –** Descoberta de prefixos de rede do enlace com a finalidade de decidir para onde os pacotes serão enviados (i.e. Para um roteador ou diretamente para um nó da rede).
- **Address Resolution –** Determinação de um endereço físico através de um endereço lógico IPv6. Este processo é executado apenas em endereços IP da rede local para os quais o endereço físico ainda não é conhecido.
- **Neighbor Unreachability Detection –** Determina se um nó vizinho continua ou não alcançável. Esta é usada para todos os caminhos entre um *host* e nós vizinhos (sejam estes *hosts* ou roteadores). O procedimento para determinar um caminho alternativo para um destino depende do nó destino, ou seja, se este nó é o próprio destino, a resuloção de endereços (*Address Resolution*) deve ser realizada novamente. Se o nó destino for um roteador, é necessário que a rota padrão seja alterada para outro roteador.
- **Redirect –** Permite a um roteador informar a um nó sobre uma melhor rota para um determinado destino.
- **Next-Hop Determination –** Algoritmo para mapear o endereço IP de um destino em um endereço IP de um nó vizinho para onde o tráfego deve ser enviado. Este vizinho pode ser um roteador ou o próprio destino.

Para as funções do NDP, foram reservadas cinco tipos de mensagens ICMPv6:

- **Router Solicitation –** Pode ser enviada cada vez que uma interface é habilitada para solicitar que roteadores anunciem sua presença na rede;
- **Router Advertisement –** Enviada por roteadores periodicamente ou em resposta à mensagem Router Solicitation para anunciar sua presença na rede. Junto com esta mensagem, são enviadas informações como prefixos de rede, configurações de endereço, valor sugerido de *hop limit*, entre outras;
- **Neighbor Solicitation –** Enviada por um nó da rede para determinar o endereço físico de um nó vizinho ou para verificar se um nó vizinho está alcançável;
- **Neighbor Advertisement –** Enviada em resposta à Neighbor Solicitation ou para anunciar a mudança no endereço físico de um nó;
- **Redirect –** Utilizada por roteadores para redirecionar um *host* a uma melhor rota para um determinado destino ou para informar ao *host* que o destino é um nó vizinho (NARTEN et al., 2013).

1.5.3 Autoconfiguração de endereços

A autoconfiguração de endereços é um mecanismo que permite a uma interface obter um endereço IPv6 global automaticamente. A autoconfiguração pode ser *stateless*, isto é, sem guardar estado, ou *statefull*, ou seja, guardando estado. Apenas interfaces com suporte a multicast podem obter endereços através de autoconfiguração. O processo de autoconfiguração tem início quando uma interface com suporte a multicast é habilitada.

O primeiro passo da autoconfiguração é a atribuição de um endereço link-local à interface. Entretanto, antes da atribuição do endereço link-local, é necessário verificar se o endereço que se pretende atribuir à interface não está sendo usado por outra interface na rede. Para isto, é utilizada a funcionalidade *Duplicate Address Detection*, do NDP. O funcionamento deste mecanismo se dá através da troca de mensagens *Neighbor Solicitation* e *Neighbor Advertisement,* onde o nó que pretende atribuir um endereço link-local a uma de suas interfaces envia a todos os nós da rede uma

mensagem *Neighbor Solicitation* contendo o endereço a ser atribuído. Se um nó estiver utilizando este endereço, ele enviará em resposta uma mensagem *Neighbor Advertisement* informando este fato e o processo de autoconfiguração para, sendo requerida configuração manual. Uma vez que um nó obtém com sucesso um endereço link-local único, ele já possui conectividade IP com nós vizinhos.

A etapa seguinte envolve troca de mensagens *Router Solicitation* e *Router Advertisement.* Roteadores enviam mensagens *Router Advertisement* periodicamente, anunciando sua presença na rede. No entanto, para determinar a presença ou não de roteadores na rede com maior rapidez, o nó envia uma mensagem *Router Solicitation* para o grupo multicast *all-routers*. Os roteadores, assim que recebem esta mensagem, enviam uma mensagem *Router Advertisement* em resposta. As mensagens *Router Advertisement* contém zero ou mais campos chamados *Prefix Information.* No caso da autoconfiguração *stateless*, estes campos têm a função de fornecer informações necessárias para a configuração de um endereço global em uma interface. Neste caso, há campos *Prefix Information* contendo um prefixo global de sub-rede e o tempo pelo qual endereços criados a partir deste prefixo permanecerão válidos. O prefixo de sub-rede obtido desta maneira e o identificador da interface formarão o endereço IPv6 global (THOMSON; NARTEN; JINMEI ,2013).

1.6 Roteamento

O roteamento no IPv6 é bastante semelhante ao roteamento no IPv4, apenas com algumas modificações para suportar o formato e o tamanho dos endereços IPv6 e a adição de algumas novas funcionalidades. Desta forma, protocolos de roteamento do IPv4, como RIP e OSPF ainda podem ser usados. Através de extensões simplificadas, o IPv6 suporta as seguintes novas funcionalidades:

- **Provider Selection –** Seleção de provedores. O enlace de um determinado provedor é selecionado com base em políticas, custo, desempenho, etc.
- **Host Mobility –** Roteamento até a localização atual de um *host* considerando a possibilidade do mesmo de deslocar;
- **Auto-Readdressing –** Roteamento para um novo endereço.

Estas novas funcionalidades de roteamento são obtidas criando-se sequências de endereços através do cabeçalho de extensão Routing. Nós de origem utilizam as opções deste cabeçalho de extensão para listar um ou mais nós (ou grupos de nós) a serem visitados até um determinado destino.

Para fazer da sequência de endereços uma função geral do IPv6, os nós, na maioria dos casos, invertem as rotas de um pacote recebido (e autenticado) que contenha sequências de endereços, com a finalidade de retornar o pacote ao nó de origem. Isto é feito para que implementações de *hosts* IPv6 suportem, desde o princípio, o tratamento e inversão de rotas de origem, de forma a garantir a interoperabilidade entre estes *hosts* e aqueles que já possuem as novas funcionalidades de roteamento (TELECO, 2013).

2 Transição IPv4/IPv6

O IPv6 e o IPv4 não são compatíveis entre sí. No entanto, embora não interoperem, ambos os protocolos podem funcionar simultaneamente nos mesmos equipamentos. Com base nisto, pensou-se em realizar a transição entre IPv4 e IPv6 de forma gradual, isto é, terminada a implementação do IPv6, este seria implantado gradualmente na Internet de forma que funcionasse simultaneamente ao IPv4. Este funcionamento simultâneo do IPv4 e do IPv6 é chamado de pilha dupla. A idéia inicial era utilizar a pilha dupla até que o IPv6 estivesse implementado em todos os dispositivos e, portanto o IPv4 não seria mais necessário.

No período de transição seriam necessárias técnicas de transição auxiliares. Inicialmente para interconectar redes IPv6 a uma Internet majoritariamente IPv4 e, posteriormente, para fazer o contrário. Desta forma, a transição seria, tecnicamente, muito mais simples (MOREIRAS et al., 2013). No entanto, a maior parte das empresas não investiu na transição por considerar que os benefícios a curto prazo não justificavam os investimentos. Ao invés disto, muitas empresas continuaram investindo na implantação de *Network Address Translation* (NAT) como solução provisória para a escassez de endereços IP (HUSTON, 2013). Hoje existe a necessidade de implantar o IPv6 em uma Internet em constante crescimento. Para isto, novas técnicas de transição foram e continuam sendo desenvolvidas. As técnicas de transição podem ser classificadas em:

- **Pilha dupla –** IPv4 e IPv6 convivem de forma nativa nos mesmos equipamentos.
- **Túneis –** Permitem que diferentes redes IPv4 se comuniquem através de uma rede IPv6 ou vice-versa.
- **Tradução –** Permitem que equipamentos utilizando IPv6 se comuniquem com equipamentos utilizando IPv4 através da conversão de pacotes (MOREIRAS et al., 2013).

Nas seções a seguir, serão detalhadas as principais técnicas de transição IPv4/IPv6 atualmente em uso.

2.1 Pilha Dupla

A forma mais básica de transição entre IPv4 e IPv6 na Internet é manter o IPv4 funcionando de forma estável e, ao mesmo tempo, implantar o IPv6 nativamente, para que ambos os protocolos coexistam nos equipamentos. A técnica de Pilha Dupla (*Dual Stack* – DS) permite que dispositivos estejam equipados tanto com pilhas IPv4 quanto IPv6, tendo capacidade de enviar e receber os dois tipos de pacotes. Desta forma, um nó Pilha Dupla se comportará como um nó IPv6 quando estiver se comunicando com outro nó IPv6 e como um nó IPv4, quando a comunicação for com outro nó IPv4 (MOREIRAS et al., 2013). A comunicação via IPv4 e IPv6 utiliza rotas aprendidas através de protocolos de roteamento específicos de cada versão do IP. Quando ambos os protocolos estão ativos, as aplicações utilizam IPv4 ou IPv6 dependendo da resposta do servidor *Domain Name Service* (DNS). O endereço de destino é selecionado com base no tipo de tráfego IP e em requisitos específicos da comunicação. A atribuição de endereços nas interfaces dos dispositivos é feita também através de mecanismos específicos de cada versão do IP. Para a pilha IPv4, o endereço é atribuido manualmente ou via DHCP *version 4* (DHCPv4), enquanto para a pilha IPv6, o endereço é atribuído via DHCPv6 ou autoconfiguração *stateless* (GOMES; TRINDADE, 2012).

Alguma configurações, como roteamento e filtros de pacotes, são indepententes entre o IPv4 e o IPv6. Portanto, para utilizar a técnica de Pilha Dupla, necessita-se de protocolos que suportem tanto IPv4 quanto IPv6 nestas funcionalidades. Configurações independentes entre IPv4 e IPv6 são necessárias para diversos aspectos da rede, como informações em servidores DNS autoritativos, protocolos de roteamento, *firewalls* e gerenciamento das redes.

Utilizar Pilha Dupla pode não ser possível em todas as situações. Um exemplo de situação onde a Pilha Dupla não pode ser usada é quando não há mais endereços IPv4 disponíveis e o provedor precisa atender a usuários novos com IPv4 e IPv6. No entanto, isto não é um impedimento quando o provedor já utiliza NAT, uma vez que um endereço IPv6 nativo pode coexistir com um endereço IPv4 compartilhado. Outro impedimento do uso da Pilha Dupla é a existência de equipamentos obsoletos que não suportam IPv6 e não podem ser facilmente substituídos. Para contornar estas situações, foram desenvolvidas outras técnicas de transição (MOREIRAS et al., 2013).

2.2 Tunelamento

Na maior parte dos cenários, a infraestrutura de roteamento para o IPv6 será construída ao longo do tempo. Durante a implantação da infraestrutura do IPv6, a estrutura já existente para o IPv4 pode ser utilizada para o transporte do tráfego IPv6. Mecanismos para o transporte de tráfego IPv6 sobre uma infraestrutura IPv4 são fornecidos pelas técnicas de tunelamento. Túneis IPv6/IPV4 podem ser implementados de diferentes formas, entre elas:

- **Roteador-a-Roteador –** Roteadores com suporte a IPv6 e IPv4 interconectados por uma infraestrutura IPv4;
- **Host-a-Roteador –** Túneis para pacotes IPv6 entre um *host* IPv6/IPv4 e um roteador intermediário alcançável através de uma infraestrutura IPv4;
- **Host-a-Host –** Túneis entre *hosts* IPv6/IPv4 interconectados através de uma infraestrutura IPv4;
- **Roteador-a-Host –** Túneis entre roteradores IPv6/IPv4 e *hosts* IPv6/IPv4 de destino. Um túnel nesta configuração compreende apenas o último segmento de um caminho fim-a-fim (NORDMARK; GILLIGAN, 2013).

Técnicas de tunelamento utilizam encapsulamento de pacotes IPv6 em pacotes IPv4. Este encapsulamento é conhecido como 6in4 (IPv6-in-IPv4) e consiste em colocar o pacote IPv6 dentro do pacote IPv4, ajustando os endereços de origem e destino para o IPv4, e atribuir ao cabeçalho IPv4 o tipo 41. Quando o destino recebe um pacote deste tipo, o cabeçalho IPv4 é removido e o pacote é tratado como IPv6. O encapsulamento 6in4 é ilustrado na Figura 10 (MOREIRAS et al., 2013). Foram propostas diversas implementações de tunelamento IPv6/IPv4. As principais implementações são: 6over4, GRE e *Tunnel* Broker.

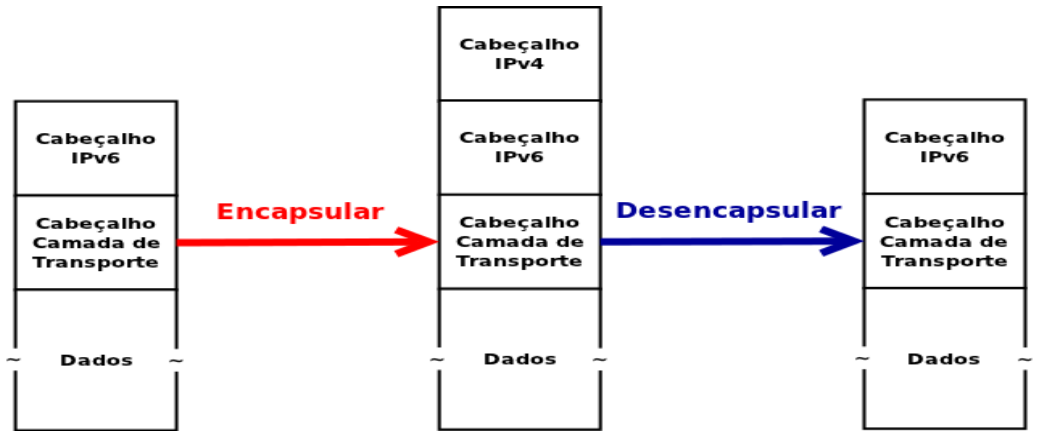

Figura 10: Encapsulamento 6in4. Fonte: (MOREIRAS et al., 2013)

2.2.1 **Tunelamento 6over4**

A forma mais básica de tunelamento é a técnica 6over4 (IPv6-over-IPv4). Esta técnica utiliza o encapsulamento 6in4, já mencionado na seção 2.2. Esta técnica utiliza um túnel manual estabelecido entre dois nós IPv4, onde a configuração manual define os enderços IPv4 de origem e destino em cada ponta do túnel. Túneis 6over4 podem ser utilizados para contornar um equipamento ou enlace sem suporte a IPv6 ou para criar túneis estáticos entre duas redes IPv6 em uma Internet IPv4 (MOREIRAS et al., 2013).

2.2.2 Túneis GRE

O *Generic Routing Encapsulation* (GRE) é uma técnica de tunelamento estático entre dois nós originalmente desenvolvida pela CISCO para o encapsulamento de diversos protocolos diferentes, entre eles o IPv6. O GRE é suportado pela maioria dos sistemas operacionais e roteadores e permite a criação de um enlace ponto-a-ponto. Assim como na técnica 6over4, a configuração dos túneis GRE é feita de forma manual, podendo gerar um esforço para manutenção proporcional à quantidade de túneis configurados.

O funcionamento do GRE consiste em adicionar um cabeçalho GRE e o cabeçalho IPv4 ao pacote original e enviá-lo ao nó destino. Uma vez que o pacote é recebido pelo nó destino (i.e. A outra ponta do túnel), os cabeçalhos IPv4 e GRE são removidos, restando apenas o pacote original, que é encaminhado normalmente (MOREIRAS et al., 2013).

2.2.3 Tunnel Broker

A técnica *Tunnel Broker* permite que dispositivos isolados ou uma rede IPv4 inteira obtenham conectividade IPv6 através do estabelecimento de um túnel junto a um provedor. Na prática, estes dispositivos ou redes funcionam como se implementassem pilha dupla.

Para utilizar a técnica *Tunnel Broker*, é necessário, primeiramente, realizar um cadastro junto a um provedor que ofereça este serviço. O provedor realizará a configuração do seu lado do túnel e fornecerá instruções (ou softwares ou scripts) para que o usuário realize a configuração do seu lado. Os provedores geralmente fornecem blocos fixos IPv6 variando de /64 a /48. Diferentes tecnologias podem ser utilizadas para prover os túneis, como 6over4, encapsulamento UDP, entre outras.

Recomenda-se a utilização de *Tunnel Brokers* para usários domésticos e corporativos que desejam se familiarizar com o IPv6 ou iniciar o processo de implantação em suas rede, porém seus provedores de acesso ainda não fornecem suporte ao IPv6. No Brasil, diversos usuários tem utilizado com sucesso túneis estabelecidos através da técnica *Tunnel Broker* para fins de testes (MOREIRAS et al., 2013).

2.3 Tradução NAT64

As técnicas de tradução foram criadas para a comunicação entre *hosts* IPv4 puros (i.e. Não possuem nenhum suporte a IPv6) e *hosts* IPv6 puros. Estas técnicas traduzem cabeçalhos IPv6 para cabeçalhos IPv4 e vice-versa e, além disto, traduzem endereços, APIs de programação e atuam na troca de tráfego TCP e UDP (DOMINGOS, 2006). Será abordada nesta seção uma das principais técnicas de tradução: o NAT64 .

O NAT64 é uma das principais técnicas de tradução. Esta técnica pode ser *statefull* (isto é, que guarda estado) ou *stateless* (isto é, que não guarda estado) de tradução de endereços IPv6 em endereços IPv4. Para a conversão do DNS, o NAT64 opera em conjunto com uma técnica conhecida como DNS64 (MOREIRAS et al., 2013). O DNS64 tem como função sintetizar um registro AAAA (quad-A) a partir de um registro A original. O nome de dono do registro AAAA sintético criado é o mesmo do registro A original. Porém, o registro AAAA possui um endereço IPv6 ao invés de um endereço IPv4, sendo que, neste caso, o endereço IPv6 deste registro é uma representação do endereço IPv4 contido no registro A original. Combinado com um mecanismo de tradução, como o NAT64, o DNS64 permite que um *host* IPv6 puro inicie uma conexão com um *host* IPv4 puro através de um nome de domínio qualificado (BAGNULO et al., 2013).

O NAT64 é configurado em equipamentos que possuem pelo menos duas interfaces de rede, uma conectada a uma rede IPv4 e a outra, a uma rede IPv6. A técnica NAT64 utiliza dois *pools* de endereços, um *pool* IPv6, para a representação de endereços IPv4 na rede IPv6, e um *pool* de

endereços IPv4, para a representação de endereços IPv6 na rede IPv4. Em função do grande espaço de endereçamento IPv6, é possível mapear cada endereço IPv4 em um endereço IPv6 diferente simplesmente concatenando um prefixo IPv6 (é recomendado o uso do prefixo 64:ff9b::/96) a um endereço IPv4 e um sufixo. Já para o caso do IPv4, em função da escassez de endereços, o mapeamento utilizando o *pool* de endereços IPv4 geralmente é feito dinamicamente. Também em função da escassez de endereços IPv4, é comum realizar o mapeamento de endereço e porta TCP entre as duas versões do protocolo IP, permitindo uma maior utilização do limitado *pool* de endereços IPv4.

Em função da natureza dinâmica do mapeamento IPv6 para IPv4, é mais simples permitir que a conexão seja iniciada em um nó IPv6. Portanto, tradução no NAT64 é feita através do mapeamento de um endereço IPv6 de origem em um endereço IPv4 de destino, juntamente com uma porta TCP. Através de um mecanismo como o NAT64, o cliente IPv6 obtém um endereço IPv6 contendo o endereço IPv4 do servidor encapsulado e envia o pacote a este endereço IPv6. O pacote é, então, interceptado pelo dispositivo que implementa o NAT64 que, por sua vez, associa ao pacote um endereço IPv4 de seu *pool* de endereços (BAGNULO; MATTHEWS; BEIJNUM, 2013). O funciomanento do NAT64 é ilustrado na Figura 11. No exemplo da Figura, o processo tem início com uma consulta do Cliente IPv6 ao servidor DNS64, buscando obter o endereço do Servidor IPv4 mapeado em um endereço IPv6. O resgistro AAAA do DNS64 apontará o nome do Servidor IPv4 para endereço do mesmo mapeado em um endereço IPv6. Para isto, o DNS64 fará uma consulta ao servidor DNS Autoritativo, obtendo o endereço IP do Servidor IPv4 a partir do registro A e sintetizará um registro AAAA apontando para o endereço do Servidor IPv4 mapeado em um endereço IPv6, que será enviado em resposta à consulta do Cliente IPv6. O Cliente IPv6 inicia a conexão com o Servidor IPv4 considerando o endereço do Servidor Ipv4 mapeado como endereço de destino. O pacote enviado pelo Cliente IPv6 é interceptado pelo dispositivo que implementa o NAT64. Este, por sua vez, mapeia o endereço do Cliente IPv6 em um endereço IPv4 de seu *pool* de endereços e uma porta TCP. Este endereço IPv4 passa a ser o endereço de origem do pacote e o endereço de destino passa a ser o próprio endereço do Servidor IPv4. Para a comunicação no sentido contrário (Servidor IPv4 – Cliente IPv6), o dispositivo que implementa o NAT64 intercepta o pacote e substitui os endereços IPv4 de origem e destino pelos respectivos endereços IPv6.

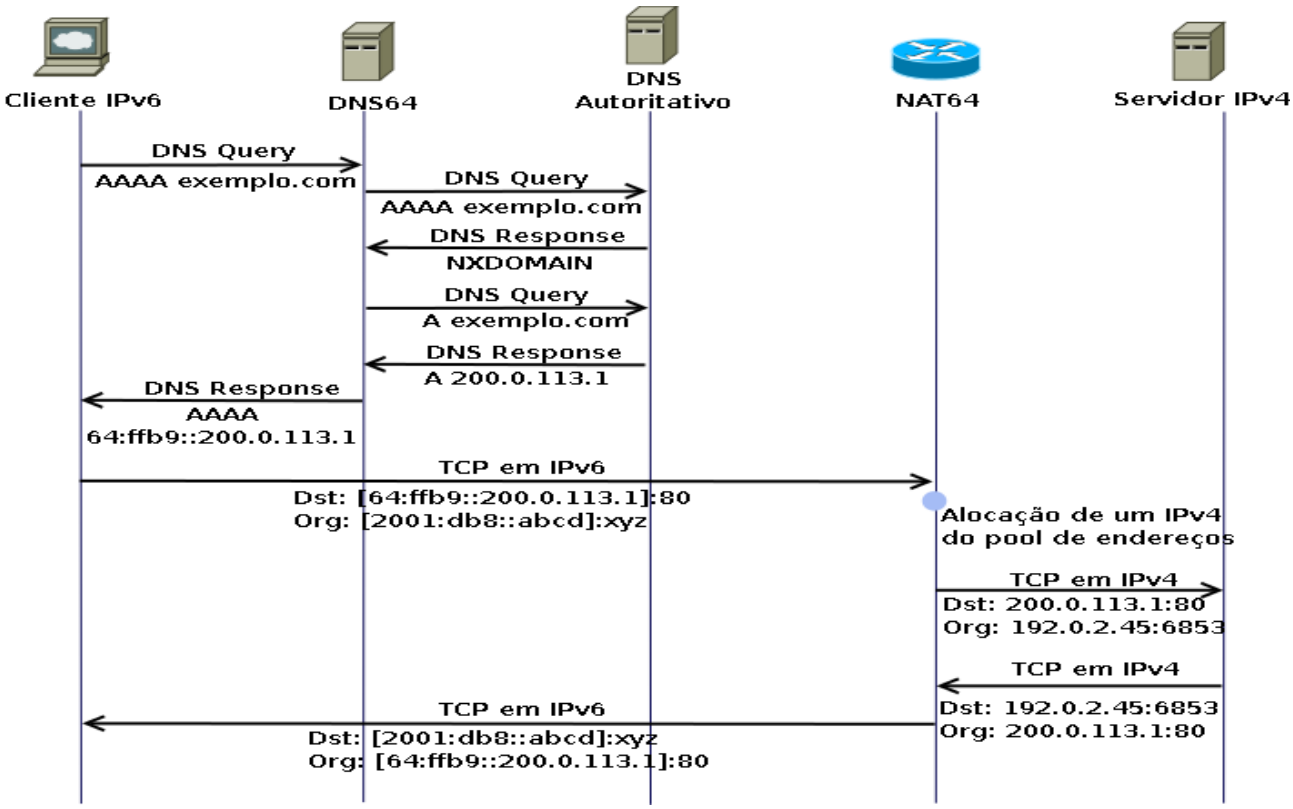

Figura 11: Funcionamento do NAT64. Fonte: (MOREIRAS et al., 2013)

2.4 DS-Lite

A técnica *Dual Stack Lite* (DS-Lite) é uma espécie de pilha dupla simplificada. Esta técnica pode ser empregada em situações onde o IPv6 já é oferecido de forma nativa pelos provedores. Para a implementação desta técnica é necessário um equipamento denominado *Address Family Transition Router* (AFTR), que implementa uma espécie de NAT chamada de *Carrier Grade NAT* (CGN). Para o transporte do tráfego IPv4 é estabelecido um túnel IPv4 sobre IPv6, sendo que nas extremidades deste túnel são utilizados endereços IPv4 da faixa reservada 192.0.0.0/29. Nos equipamentos de redes dos usuários são utilizados endereços IPv4 privados, atribuídos via DHCPv4. Um proxy DNS é utilizado para consultas via IPv4, sendo que estas consiltas são feitas recursivamente via IPv6 para evitar traduções desnecessárias no AFTR (MOREIRAS et al., 2013).

3 Segurança no IPv6

Ainda que o IPv6 seja um protocolo simplificado e aprimorado em relação ao IPv4, ele apresenta uma série de desafios em relação à segurança. Muitas ameaças presentes em redes IPv4 afetam também redes IPv6 e, com a maior utilização do IPv6, novas ameaças que atacam especificamente este protocolo vão sendo reveladas (CAICEDO; JOSHI; TULADHAR, 2009). Além disto, existem ataques que exploram a coexistência entre IPv4 e IPv6, o que significa que a transição pode implicar em vulnerabilidades na rede (MOREIRAS et al., 2013). Um aprimoramento na segurança do IPv6 foi o espaço de endereçamento que, por ter 96 bits a mais que o espaço de endereçamento do IPv4, dificulta a varredura na rede, o que no IPv4 é relativamente simples. Além disto, foi definido inicialmente que a inclusão do IPsec seria obrigatória em toda implementação do IPv6. No entanto, a inclusão do IPsec deixou de ser obrigatória posteriormente (SANKARAN, 2013).

3.1 IPsec

O IPsec consiste em um conjunto de protocolos criptográficos que possibilitam comunicação de dados e troca de chaves de forma segura. O IPsec utiliza os protocolos *Autehntication Header* (AH), que provê serviços de integridade de dados e autenticação, e o *Encapsulating Security Payload* (ESP), que além destes serviços, provê confidencialidade. Tanto o AH quanto o ESP são definidos como cabeçalhos de extensão no IPv6 (SANKARAN, 2013). Como os serviços de segurança fornecidos pelo IPsec utilizam chaves criptográficas, é necessário um mecanismo para a distribuição destas chaves. Especificamente para o IPsec, é definido um mecanismo denominado *Internet Key Exchange* (IKE), que provê as funcionalidades necessárias para a negociação de parâmetros de segurança entre dois nós. No entanto, outros mecanismos de distribuição de chaves podem ser utilizados para a distribuição de chaves no IPsec (KENT; ATKINSON, 2013).

3.2 Ameaças semelhantes em redes IPv6 e redes IPv4

Apesar das significativas mudanças que o IPv6 apresenta em relação ao IPv4, há ataques conhecidos em cenários IPv4 que podem afetar redes IPv6. Alguns destes são realizados de forma bastante similar em ambas as versões do protocolo IP. Além disto, há ataques que exploram a coexistência entre o IPv4 e o IPv6.

Comum em redes IPv4, o ataque de *sniffing* pode afetar igualmente redes IPv6. Este ataque consiste na captura de pacotes que trafegam em uma rede, possibilitando ao atacante a visualização de dados em texto claro contidos nos pacotes. Ataques de *sniffing* podem ser evitados através de uma implementação adequada do IPsec, cuja inclusão foi prevista para ser obrigatória em toda implementação do IPv6. No entanto, a inclusão do IPsec deixou de ser obrigatória e, portanto, ataques de *sniffing* podem afetar redes IPv6 praticamente da mesma forma que afetam redes IPv4.

Ataques visando a camada de aplicação, a exemplo de *buffer overflow*, ataques a aplicações *web*, vírus e descaracterização (*defacement*) de páginas, continuam passíveis de ocorrer da mesma forma. A implantação do IPv6 não traz novos mecanismos para a prevenção destes ataques, uma vez que eles ocorrem especificamente na camada de aplicação.

Ataques que envolvem a sobrecarga de um dispositivo com requisições, como o ataque de Negação de Serviço (*Denial of Service* – DoS), afetam redes IPv6 assim como afetam redes IPv4. Isto ocorre porque o princípio destes ataques continua o mesmo com a implantação do IPv6. Além disto, cabeçalhos de extensão, novas mensagens ICMPv6 e endereçamento multicast podem viabilizar outras formas de ataques de DoS.

Outro ataque comum às duas versões do protocolo IP é o ataque *Man-in-the-Middle*. No caso do IPv6, os riscos deste tipo de ataque estão diretamente relacionados a ataques de *Man-in-the-Middle* visando o IPsec, especificamente o mecanismo para troca de chaves IKE (DURDAGI; BULDU, 2010).

3.3 Questões de segurança específicas do IPv6

Com a especificação do IPv6, surgiram ataques que exploram funcionalidades e particularidades do IPv6. Cabeçalhos de extensão, NDP, ICMPv6 e autoconfiguração *stateless* são exemplos de características do IPv6 exploradas por ataques. Além disto, algumas ameaças presentes em redes IPv4 se adaptaram a redes IPv6.

O uso de cabeçalhos de extensão combinados com IPsec podem evitar ataques baseados em manipulação de cabeçalho. No entanto, os próprios cabeçalhos de extensão podem ser utilizados em ataques. Uma longa cadeia de cabeçalhos de extensão pode sobrecarregar um dispositivo ou mascarar um ataque (SANKARAN, 2013). O principal cabeçalho de extensão associado a ataques é o cabeçalho *Routing*, que pode ser utilizado para burlar controle de acesso, o que é feito através do envio de um pacote por um *host* a um roteador, acessível publicamente, contendo um cabeçalho *Routing* com o endereço de um servidor inacessível publicamente como destino . O roteador, ao receber este pacote, o encaminhará para o servidor. Em caso de sucesso, uma série de pacotes ilegítimos pode ser gerada e enviada desta forma para o servidor, o que resulta em um ataque de *DoS* (DAVIES; KRISHNAN; SAVOLA, 2013).

O protocolo NDP pode também ser explorado por atacantes. Um destes ataques é o ataque de *DoS* explorando a funcionalidade de *Duplicate Address Detection.* Este ataque consiste em enviar uma mensagem *Neighbor Advertisement* em resposta a cada mensagem *Neighbor Solicitation* recebida. Isto fará com que os endereços de tentativa sejam sempre considerados em uso, impedindo que novos dispositivos obtenham endereços IP válidos e se conectem à Internet. Outro ataque envolvendo o NDP é a falsificação de mensagens *Router Advertisement*, onde um dispositivo que não é um roteador envia esta mensagem possivelmente com a finalidade de se tornar roteador e interceptar o tráfego ou realizar um ataque de *DoS* através do anúncio de um roteador falso, criando um local para onde todo o tráfego é desviado (MOREIRAS et al., 2013).

Apesar de serem feitas com maior dificuldade devido ao maior espaço de endereçamento, varreduras ainda são possíveis em redes IPv6. Como seria inviável percorrer toda uma rede local IPv6 (tipicamente com 2^{64} possíveis hosts), as varreduras em redes IPv6 visam endereços multicast que identificam grupos de dispositivos de interesse dos atacantes (por exemplo, todos os roteadores)(DURDAGI; BULDU, 2010).

3.3.1 Questões de segurança relacionadas ao ICMPv6

O ICMPv6 é um protocolo fundamental para o funcionamento do IPv6, sendo utilizado em funcionalidades essenciais do mesmo, como a descoberta de vizinhos (via NDP) e a autoconfiguração de endereços. Devido à sua importância, é de enorme importância que a troca de mensagens ICMPv6 seja segura. Apesar de sua importância, o ICMPv6 é classificado como um protocolo simples e não é chamada muita atenção para os problemas de segurança a ele associados,

o que o torna vulnerável a ataques.

Um dos ataques que utiliza mensagens ICMPv6 é o ataque de *DoS*. Há várias maneiras de realizar este ataque através do ICMPv6, incluindo o envio de uma quantidade excessiva de pacotes ICMPv6 a um determinado *host* e o envio de mensagens de erro ilegítimas que interrompem conexões. Além disto, mensagens ICMPv6 podem ser utilizadas para impedir que dispositivos obtenham endereços IPv6 válidos e desabilitar intefaces.

O ataque de *flooding*, um tipo de ataque de *DoS* que consiste em sobrecarregar a banda de um enlace, impedindo a comunicação entre outros *hosts* através do mesmo, também ocorre em redes IPv6 através do ICMPv6. O princípio deste ataque é o mesmo que no IPv4, porém novos tipos de pacotes ICMP e a dependência de endereços multicast no IPv6 possibilitam a realização destes ataques de diferentes formas (SAAD et al., 2013).

3.4 Questões de segurança referentes a técnicas de transição

Estima-se que uma transição completa para o IPv6 ainda levara um longo tempo. Isto significa que ambas as versões do protocolo IP funcionarão em conjunto por um longo período através de técnicas de transição. Portanto, para manter uma rede segura, é necessário levar em consideração não só as ameaças particulares de cada versão do IP como também ameaças que exploram a coexistência dos protocolos (TAIB; BUDIARTO, 2007). Mesmo se o IPv6 não for nativo na rede, é necessário considerá-lo ao prever medidas de segurança, pois sistemas operacionais atuais possuem suporte ao IPv6 e o mesmo pode ser utilizado independente de ter sido implementado oficialmente na rede (MOREIRAS et al., 2013). A presente seção discute questões de segurança relativas a técnicas de transição IPv4/IPv6.

3.4.1 Segurança em pilha dupla

Dispositivos operando com pilha dupla terão tanto endereços IPv4 quanto endereços IPv6 configurados em suas interfaces. Uma das implicações do uso desta técnica é que ataques podem atingir o dispositivo tanto via IPv4 quanto via IPv6. Portanto, mecanismos de controle como *firewalls*, clientes VPN e sistemas de detecção de intrusões devem ser capazes de analisar tanto tráfego IPv4 quanto IPv6 e, quando necessário, bloquear tráfego de cada versão do IP

especificamente. Em um cenário de pilha dupla, é recomendável que as configurações de *firewall* sejam adaptadas de forma a suportar também o IPv6 e conter um conjunto de regras específico para esta versão do IP. Alternativamente, pode-se configurar um *firewall* específico para o IPv6, separado do *firewall* IPv4 (TAIB; BUDIARTO, 2007).

3.4.2 Segurança em técnicas de tunelamento

As diferentes formas de tunelamento propostas também podem apresentar vulnerabilidades. Algumas destas vulnerabilidades independem da forma de tunelamento utilizada, enquanto outras ocorrem especificamente em determinadas técnicas de tunelamento. Um risco envolvendo técnicas de tunelamento em geral é o encapsulamento de um pacote IPv6 que carrega conteúdo malicioso em um datagrama IPv4 legítimo e examinar apenas o pacote externo, ignorando o conteúdo do pacote encapsulado. Neste caso, quando o pacote IPv4 chegar à saída do túnel, o pacote IPv6 malicioso é desencapsulado sem ser examinado e consegue adentrar a rede IPv6, podendo causar danos (BILSKI, 2011).

O Tunnel Broker é uma técnica utilizada por usuários domésticos do IPv6. Esta técnica se torna um problema quando o administrador da rede desconhece a existência usuários dentro de sua rede que utilizam Tunnel Brokers. Sem este controle, o administrador não pode prever ataques que venham a explorar os túneis estabelecidos.

Em túneis configurados manualmente, o problema está na necessidade de configurar o *firewall* dos nós onde se encontram as pontas do túnel para permitir a entrada de pacotes IPv4 que encapsulam pacotes IPv6 e ICMPv6. Como não há um método de especificação próprio para técnicas de tunelamento, a verificação do pacote é feita através do endereço de origem do pacote IPv4 encapsulador. No entanto, tanto o endereço do pacote encapsulador quanto o do pacote encapsulado podem ser ilegítimos *(spoofing)*, fazendo com que conteúdos maliciosos possam ser injetados nas extremidades dos túneis ou passar por *firewalls* (TAIB; BUDIARTO, 2007). Roteadores 6to4 também podem ser afetador por *spoofing* (BILSKI, 2011).

3.4.3 Segurança em técnicas de tradução

As vulnerabilidades das técnicas de tradução estão principalmente relacionadas ao IPsec e a ataques de *DoS*. Mecanismos de tradução podem não suportar os esquemas de segurança fim-a-fim que dependem dos endereços de origem e destino, como o IPsec. Isto se deve ao fato destes mecanismos de transição modificarem os endereços de origem e destino no processo de tradução.

Técnicas de tradução necessitam que uma série de informações de estado sejam mantidas. Isto pode ser explorado para iniciar um ataque de *DoS* através do envio de pequenos fragmentos de dados indefinidamente ao dispositivo que implementa o mecanismo de tradução. Este dispositivo também pode ser alvo de um ataque *reflected DoS*, onde ele receberia pacotes alterados contendo um endereço multicast como endereço de origem(BI; WU; LENG, 2007).

No caso da técnica de tradução NAT64, que opera em conjunto com o DNS64, é preciso considerar ataques ao DNS, uma vez que mecanismo DNS64 está sucetível aos mesmos ataques que o DNS (BAGNULO et al., 2013). Um dos principais ataques visando o DNS é o redirecionamento de requisições DNS feito através de um servidor DNS falso na rede local. Quando o atacante consegue anunciar um servidor DNS falso em uma rede local, este atacante pode responder a resuisições associando um nome a um edereço IP não correspondente e levando *hosts* da rede local a acessarem um outro servidor, que pode ser utilizado para obter credenciais de acesso, por exemplo. Além disto, a associação de nomes a endereços IP não correspondentes pode acarretar em um ataque de *DoS* caso o endereço IP anunciado pelo servidor DNS falso não exista (JANBEGLOU; ZAMANI; IBRAHIM, 2010). Para atacar o DNS64 o atacante pode, além disto, alterar o prefixo utilizado para a tradução no NAT64, uma vez que este prefixo deve ser o mesmo utilizado pelo DNS64. A alteração do prefixo utilizado pelo NAT64 pode resultar em ataques de *DoS*, *flooding* e pode também possibilitar a captura de pacotes e vizualização de informações contidas nos mesmos pelo atacante (BAGNULO et al., 2013).

Possíveis defesas para ataques que tem como alvo mecanismos de tradução seriam a validação dos endereços de origem e destino dos pacotes, esquemas de autenticação e mecanismos que realizem a tradução de forma estática. No entanto, estes mecanismos de defesa poderiam acarretar em um grande consumo de recursos computacionais e tornar a implementação das técnicas de tradução muito mais complexas (BI; WU; LENG, 2007).

4 Cenários de testes e experimentos

A implantação do IPv6 traz uma série de implicações quanto à segurança das redes. Não apenas as ameaças já existentes em redes IPv4 continuam presentes, como há também uma série de ataques que exploram especificamente o IPv6 e a convivência entre IPv4 e IPv6 através das técnicas de transição.

Para que se possa fazer uma avaliação das questões de segurança relativas ao IPv6 e às técnicas de transição, visualizando as ameaças em experimentos práticos e elaborando estratégias de defesa, foi proposto para este trabalho a realização de testes em cenários de redes montados em um ambiente de laboratório que implementa ambas as versões do protocolo IP, bem como técnicas de transição. Foram definidos dois cenários de testes. O primeiro deles, descrito na Seção 4.2, funciona com a técnica de pilha dupla, onde todos os *hosts* possuem tanto endereços IPv4 quanto endereços IPv6 válidos (isto é, não é utilizado NAT). No segundo cenário, descrito na Seção 4.3, opera uma rede puramente IPv6 que se comunica com a Internet IPv4 através da técnica de tradução NAT64. A técnica de pilha dupla foi escolhida para os testes por ser a mais básica forma de transição IPv4/IPv6. O NAT64, além de sua importância, foi escolhido por ser uma técnica de mais fácil implantação na infraestrutura disponível para a realização de testes.

Para os cenários de testes, foi utilizada a infraestrutura física do Ponto de Presença (*Point of Presence*) da Rede Nacional de Ensino e Pesquisa em Santa Catarina (PoP-SC/RNP). Os cenários são constituídos de máquinas virtuais rodando o sistema operacional Linux, distribuição Debian 7.0 64 bits. As máquinas virtuais estão alocadas em um servidor de virtualização do PoP-SC/RNP, utilizando como ambiente de virtualização o VMware *vSphere ESXi Hypervisor*, detalhado na Seção 4.1.1. Algumas destas máquinas atuam como roteadores devido à indisponibilidade de roteadores reais para a realização dos experimentos, para isto, foi habilitada a função de roteamento do *Kernel* Linux, permitindo que as máquinas virtuais Linux encaminhassem pacotes. As configurações específicas dos cenários de teste com pilha dupla e NAT64 são descritas respectivamente nas seções 4.2 e 4.3. As alocações de blocos IP para cada cenário e de endereços IP para cada interface podem ser visualizados nas figuras 12 e 13.

Para cada cenário de testes, foram desenvolvidos experimentos visando atingir cada uma das redes IPv6 através de vulnerabilidades deste protocolo e das técnicas de transição IPv4/IPv6 . Em cada cenário, os ataques foram realizados pela máquina atacante através de utilitários específicos da ferramenta THC-IPV6, descrita na Seção 4.1.4. Todos os utilitários do THC-IPV6 são executados via Interface de Linha de Comando (*Command Line Interface* – CLI). O escopo dos ataques é a rede

local (LAN) dos cenários A Tabela 1 relaciona os experimentos realizados, ferramentas realizadas, cenários de teste onde foram realizados e efeitos esperados dos ataques. Os experimentos realizados em cada um dos cenários de testes são descritos nas Seções 4.3 e 4.5

Ataque	**Utilitário THC-IPv6**	**Cenário**	**Comparativo com IPv4**
DoS para novos endereços IPv6	dos_new_ip6	Pilha dupla e NAT64	Este ataque é específico do IPv6
Anúncio de um roteador falso	fake_router26	Pilha dupla e NAT64	Ataque possível, porém, no Ipv6, pode explorar funcionalidade específica
Flooding de mensagens *Neighbor Advertisement*	flood_advertise6	Pilha dupla e NAT64	Este ataque é específico do IPv6
Flooding de mensagens *Router Advertisement*	flood_router26	Pilha dupla e NAT64	Este ataque é específico do IPv6
Servidor DNS falso	parasite6 fake_dns6d	NAT64	Ataque possível, porém no IPv6 pode atingir DNS64

Tabela 1: Experimentos realizados

4.1 Ferramentas utilizadas

Para a realização dos experimentos, uma série de ferramentas tiveram de ser utilizadas para montagem dos cenários, configuração das máquinas virtuais e realização dos ataques, além da detecção dos mesmos. As principais ferramentas utilizadas nos cenários de testes são descritas nesta seção.

4.1.1 *VMware vSphere ESXi Hypervisor*

O *vSphere ESXi Hypervisor* é um ambiente de virtualização desenvolvido pela empresa norte

americana VMware. O *vSphere ESXi Hypervisor* é instalado diretamente no *hardware* do servidor, ou seja, não roda sobre outro sistema operacional. As máquinas virtuais compartilham os mesmos recursos físicos e todas podem rodar ao mesmo tempo. Além disto, todo o gerenciamento do *vSphere ESXi Hypervisor* pode ser feito remotamente através de ferramentas específicas (VMWARE, 2014).

O *vSphere ESXi Hypervisor* foi utilizado para este trabalho pois é a ferramenta de virtualização utilizada no PoP-SC/RNP e havia um servidor com o *vSphere ESXi Hypervisor* instalado disponível para a criação das máquinas virtuais dos cenários de testes.

4.1.2 *Router Advertisement Daemon* – RADVD

O RADVD é um utilitário para enviar mensagens *Router Advertisement* em sistemas operacionais Linux e BSD. O RADVD é utilizado em um *host* Linux ou BSD para que este atue como roteador IPv6, enviando periodicamente mensagens *Router Advertisement* para a rede local e possibilitando o funcionamento do mecanismo de autoconfiguração de endereços *stateless* (LITECH, 2014).

O RADVD foi utilizado neste trabalho devido ao fato de máquinas virtuais Linux serem utilizadas como roteadores nos cenários e também devido ao fato de ser a ferramenta mais utilizada para o envio de mensagens *Router Advertisement* a partir de *hosts* Linux, além de ser *software* livre e de código aberto.

4.1.3 Tayga

O Tayga é uma implementação de NAT64 *stateless* para Linux. O Tayga opera em espaço de usuário e utiliza uma interface TUN, uma interface de rede definida por *software*, para a troca de pacotes IPv4 e IPv6 com o *Kernel* Linux (LITECH, 2014). O Tayga foi utilizado para este trabalho devido ao fato do dispositivo que realiza a tradução ser uma máquina virtual Linux e por ser de fácil instalação e configuração, além de ser *software* livre.

4.1.4 *The Hacker's Choice* IPv6 – THC-IPv6

O THC-IPv6, desenvolvido pela comunidade *The Hacker's Choice*, e consiste em uma série de utilitários para a realização de testes, ataques e exploração de vulnerabilidades dos protocolos IPv6

e ICMPv6. O THC-IPv6 inclui ferramentas para a realização de ataques de *DoS, flooding, spoofing,* entre outros (THE HACKER'S CHOICE, 2014). O THC-IPv6 foi utilizado neste trabalho pela vasta gama de utilitários para ataques explorando vulnerabilidades do IPv6, o que o torna ideal para os testes realizados, além de ser uma ferramenta livre e desenvolvida de forma colaborativa.

4.1.5 *6Guard*

O *6Guard* é uma ferramenta para a detecção de ataques ao IPv6 baseada em *honeypot* projetada para detectar ataques em uma rede local. O *6Guard* é escrito em *Python*, utilizando a biblioteca *Scapy* e contém três módulos:

- **Módulo *honeypot* –** um *host* IPv6 de baixa interação que suporta as funcionalidades NDP e autoconfiguração *stateless*, responsável por detectar ataques unicast.
- **Módulo *globalpot* –** módulo que foca na detecção de ataques multicast.
- **Módulo de análise de eventos –** módulo responsável por analisar as mensagens recebidas na rede local e gerar as mensagens de alerta (THE HONEYNET PROJECT, 2014).

O *6Guard* foi utilizado neste trabalho por ser uma ferramenta livre e de fácil instalação, além de ser capaz de detectar um grande número de ataques passíveis de serem realizados em redes IPv6.

4.1.6 *Wireshark* e *tcpdump*

O Wireshark é uma ferramenta de análise de tráfego de rede via interface gráfica. O Wireshark possui funcionalidades de captura de tráfego em tempo real e também de análise de tráfego *offline*, possibilitando salvar e exportar arquivos de capturas. Além disto, o Wireshark suporta capturas em diversas arquiteturas de rede (WIRESHARK, 2014). Neste trabalho, o Wireshark foi utilizado para a análise de tráfego, devido à facilidade de análise proporcionada pela ferramenta, além do fato de ser *software* livre. No entanto, como as máquinas virtuais do cenário não dispunham de interface gráfica, optou-se por utilizar, para a captura de tráfego, o *tcpdump.*

O *tcpdump* é uma ferramenta de análise de tráfego que opera via CLI. Em conjunto com uma biblioteca denominada *libpcap*, o *tcpdump* suporta também captura de tráfego, permitindo redirecionar a saída da captura para um arquivo que pode ser aberto em outras ferramentas, como o Wireshark, para análise (TCPDUMP, 2014). Além de suas funcionalidades e do fato de operar via CLI, o *tcpdump* foi utilizado neste trabalho por também ser *software* livre.

4.2 Cenário de testes pilha dupla

A coexistência temporária entre IPv4 e IPv6 é a forma mais básica de transição entre as versões do protocolo IP. Esta forma de transição é implementada pela técnica de pilha dupla. Para os primeiros testes com a técnica de pilha dupla, foi elaborado um ambiente de laboratório onde todos os hosts possuem tanto endereços IPv4 quanto IPv6 configurados em suas interfaces, sendo os endereços IPv4 roteáveis na internet. Este cenário foi elaborado com o objetivo de explorar vulnerabilidades específicas do IPv6, que afetam diretamente uma rede IPv4/IPv6 com pilha dupla. A Figura 12 mostra um diagrama deste cenário de testes.

O cenário é composto por um roteador, um firewall, um servidor web e três *hosts*, sendo que um deles atua como atacante na rede local. O roteador possui três interfaces de rede, sendo duas interfaces *Wide Area Network* (WAN), eth0 e eth1, e uma interface *Local Area Network* (LAN). A interface eth0 está diretamente conectada ao roteador do PoP-SC via IPv4 e a interface eth1, diretamente conectada via IPv6. A interface eth2 está diretamente conectada ao firewall, tanto via IPv4 quanto via IPv6.

O firewall possui, também, três interfaces de rede. A interface eth0 atua como interface WAN, diretamente conectada ao roteador do cenário. A interface eth1 está conectada ao servidor web via IPv4 e IPv6. O servidor possui endereços IP estáticos. A interface eth2 é o *gateway* para os *hosts.* Os endereços IPv4 são distribuídos aos *hosts* via DHCPv4 e os endereços IPv6 são obtidos via autoconfiguração de endereços *stateless*. Para a divulgação das informações de roteamento através de mensagens *Router Avertisement*, é utilizada a ferramenta *Router Advertisement Daemon* (RADVD).

A máquina atacante está conectada à rede local junto aos *hosts* e obtém endereços IPv4 e IPv6 dinamicamente, assim como os demais *hosts*. Para que esta máquina realize ataques, foi utilizada a ferramenta *The Hacker's Choice IPv6* (THC-IPv6).

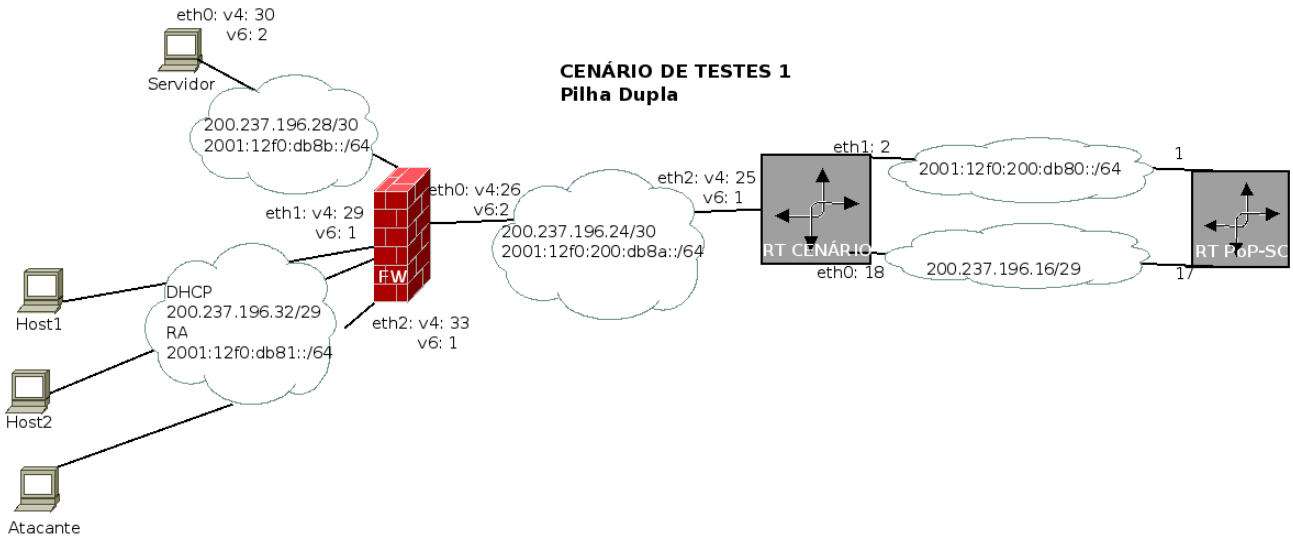

Figura 12: Cenário de testes pilha dupla. Fonte: própria

4.3 Experimentos realizados no cenário de testes pilha dupla

Há uma série de ataques que exploram mecanismos específicos do IPv6, a exemplo do NDP e da autoconfiguração de endereços. Ao ativar IPv6 em uma rede IPv4, tornando-a uma rede pilha dupla, as vulnerabilidades inerentes ao IPv6 passam a ameaçar esta rede. Os experimentos no cenário de testes pilha dupla foram elaborados visando demonstrar vulnerabilidades em uma rede decorrentes da implantação do IPv6. Neste cenário, foram realizados, a partir da rede local, ataques de *DoS* através da sobrecarga da rede (isto é, *flooding*) com pacotes ilegítimos e através do mascaramento de pacotes (*spoofing*). Os ataques realizados e as ferramentas utilizadas para os mesmos são descritos nas seções de 4.3.1 a 4.3.4.

4.3.1 *Experimento 1 – DoS para novos endereços Ipv6*

O ataque realizado neste primeiro experimento consiste em um ataque de *DoS* onde a máquina atacante explora a funcionalidade *Duplicate Address Detection*, enviando uma mensagem *Neighbor Advertisement* para cada mensagem *Neighbor Solicitaiton* recebida e fazendo com que dispositivos que se conectam à rede não consigam obter endereços IPv6.

Para este ataque, foi utilizado o utilitário *dos_new_ip6*, parte do THC-IPV6. Ao ser executado via CLI, o *dos_new_ip6* deve receber como parâmetro uma interface de rede. O *dos_new_ip6* passará a responder a todas as mensagens *Neighbor Solicitation* recebidas pela interface

especificada enviando uma mensagem *Neighbor Advertisement*. Este experimento envolveu a máquina atacante, executando o *dos_new_ip6*, e dois *hosts*, tentando obter endereços IPv6. A Figura 13 ilustra o ataque realizado neste experimento. Na figura, as setas pretas, rotuladas como NS, representam mensagens *Neighbor Solicitation* enviadas pelos *hosts* durante o procedimento de detecção de endereços duplicados, com a finalidade de verificar se os endereços IPv6 que desejam atribuir a suas interfaces de rede estão em uso. As setas vermelhas, rotuladas como NA, representam mensagens *Neighbor Advertisement* enviadas pela máquina atacante, anunciando que possui os endereços pretendidos pelos *hosts*.

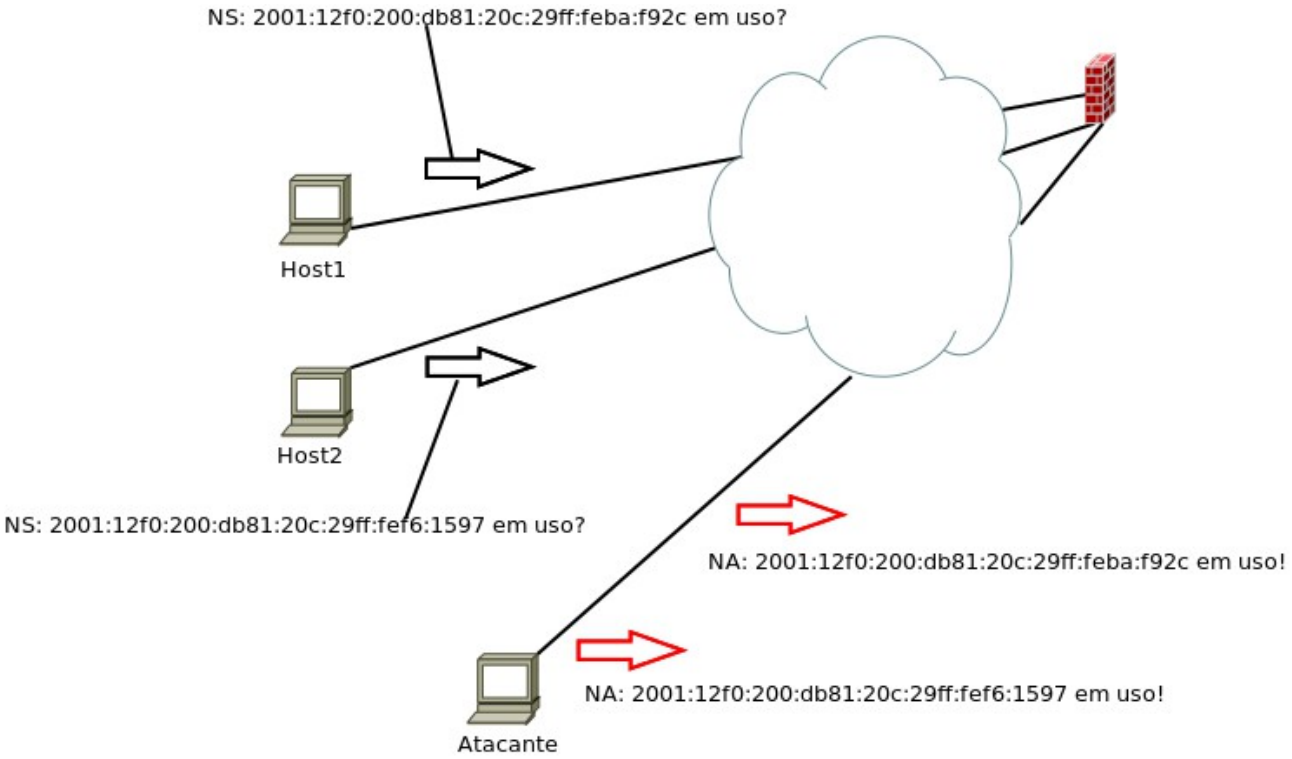

Figura 13: Representação do ataque de DoS para novos endereços IPv6. Fonte: própria

4.3.2 Experimento 2 – anúncio de um roteador falso

Neste segundo experimento, a máquina atacante envia mensagens *Router Advertisement* para todos os nós da rede local, anunciando a si mesma como roteador. Este ataque pode possibilitar à maquina atacante capturar o tráfego da rede local ou pode acarretar em um ataque de *DoS*, onde todo o tráfego é direcionado para um dispositivo que não é roteador e, portanto, não é encaminhado.

Para este ataque, foi utilizado o utilitário *fake_router26*. Este utilitário dispara mensagens *Router Advertisement* na rede local, anunciando a própria máquina onde é executado como roteador. Entre as opções deste utilitário estão a possibilidade de especificar um prefixo de rede a ser

divulgado, a possibilidade de especificar um endereço IP e/ou um endereço MAC de origem e até mesmo opções para burlar o mecanismo de segurança *Router Advertisement Guard* (RA Guard). Se executado sem opções, apenas tendo uma interface de rede como parâmetro, o que é obrigatório, o *fake_router26* enviará mensagens *Router Advertisement* tendo como endereços IP e MAC de origem os respectivos endereços da máquina onde é executado. Este experimento foi executado sob duas abordagens diferentes: execução do *fake_router26* sem opções, com o intuito de capturar o tráfego da rede local a partir da máquina atacante, e a execução do *fake_router26* especificando um endereço IPv6 link-local falso, com o intuito de realizar um ataque de *DoS*. Este experimento envolveu a máquina atacante, executando o *fake_router26,* e dois *hosts,* nos quais era testada a conectividade com a Internet. A Figura 14 ilustra a forma como o ataque ocorre neste experimento. As setas rotuladas como RA representam mensagens *Router Advertisement.* No entanto, as setas pretas representam mensagens legítimas, enviadas pelo roteador do cenário, enquanto as setas vermelhas representam mensagens ilegítimas, enviadas pela máquina atacante.

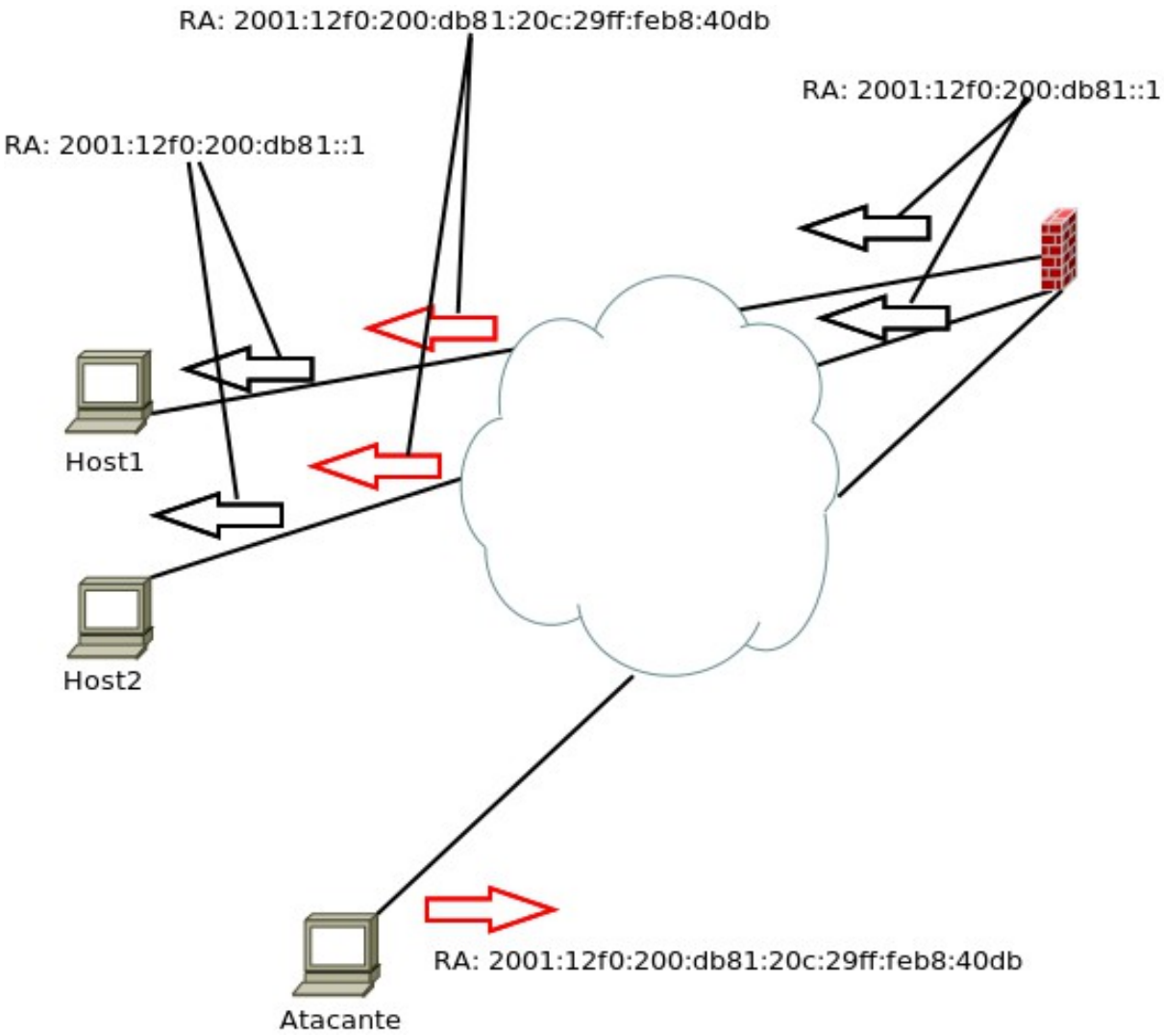

Figura 14: Representação do ataque de DoS através do anúncio de um roteador falso. Fonte: própria

4.3.3 Experimento 3 – *Neighbor Advertisement flooding*

Neste experimento, a máquina atacante realiza um ataque de *flooding* com mensagens *Neighbor Advertisement*, o que significa enviar uma grande quantidade de mensagens *Neighbor Advertisement* para a rede local, causando negação de serviço (*DoS*).

Para este ataque, foi utilizado o utilitário *flood_advertise6*. Quando executado, este utilitário dispara uma grande quantidade de mensagens *Neighbor Advertisement* na rede local e segue disparando mensagens até que seja parado manualmente. O *flood_advertise6* recebe como parâmetro uma interface de rede, por onde serão disparadas as mensagens, e pode receber, opcionalmente, o endereço IP de um alvo para o *flood*. Neste experimento, *flood_advertise6* foi executado sem opções. O experimento envolveu a máquina atacante, executando o *flood_advertise6,* dois hosts, nos quais era testada a conectividade com a Internet, e o servidor *web*, verificando-se a possibilidade de acessá-lo. A Figura 15 ilustra o funcionamento do ataque neste experimento. Na figura, as setas vermelhas representam o grande volume de mensagens *Neighbor Advertisement* enviadas para a rede local.

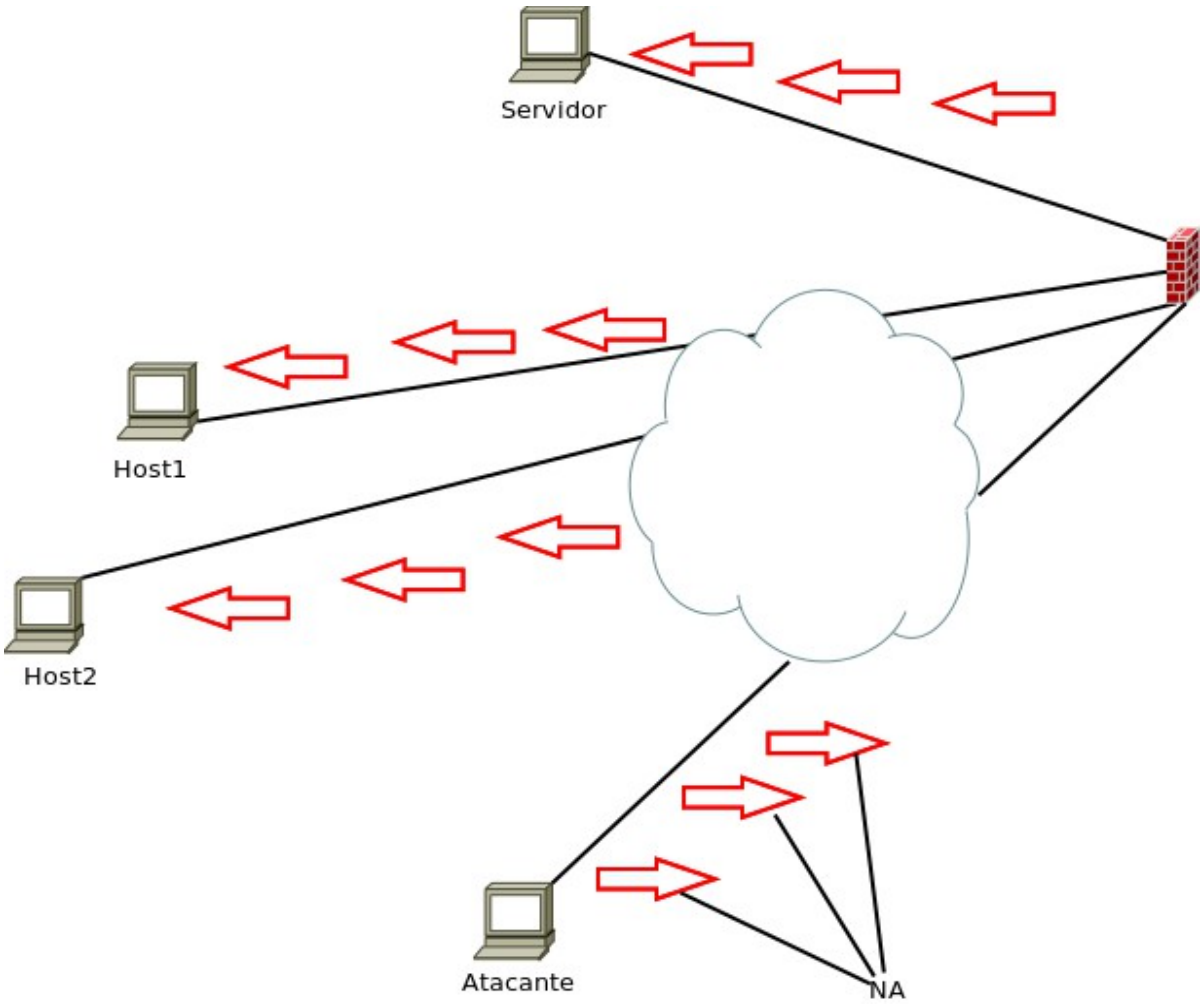

Figura 15: Representação do ataque de flooding de mensagens Neighbor Advertisement. Fonte: própria

4.3.4 Experimento 4 – *Router Advertisement flood*

Este experimento também consiste em um ataque de *flooding*. No entanto, o *flooding* foi feito com mensagens *Router Advertisement*. Neste caso, a máquina atacante disparava uma série de mensagens *Router Advertisement* falsas que, além de serem enviadas em grande quantidade devido ao *flooding*, continham prefixos de rede e informações de roteamento inválidas, o que faz com que os *hosts* que recebem estas mensagens adquiram um grande número de endereços IPv6 e rotas inválidas.

Para este experimento foi utilizado o utilitário *flood_router26*, que dispara uma grande quantidade de mensagens *Router Advertisement* falsas. Existem diferentes modos de execução para este utilitário: enviar apenas informações de roteamento, enviar apenas informações de prefixo de rede e desabilitar extensões de privacidade. Por padrão, o *flood_router26* envia tanto informações

de roteamento quanto prefixos de rede. Este utilitário inclui opções que permitem alterar parâmetros das mensagens RA (como tamanho da mensagem e TTL) e também opções para tentar burlar RA Guard. Neste experimento, o *flood_router26* no modo padrão, sem opções, tendo como parâmetro apenas a interface de rede, parâmetro obrigatório. Este experimento envolveu a máquina atacante, disparando mensagens *Router Advertisement*, os dois *hosts* e o servidor *web*, para os quais era testada a conectividade via ICMP e HTTP, respectivamente.A Figura 16 ilustra o funcionamento do ataque neste experimento. As setas vermelhas indicam uma grande quantidade de mensagens *Router Advertisement* enviadas pela máquina atacante para a rede local, conendo prefixos aleatórios. Estas mensagens levam os *hosts* a configurarem diversos endereços IPv6 inválidos em suas interfaces de rde conectadas à rede local, conforme representado na Figura 16.

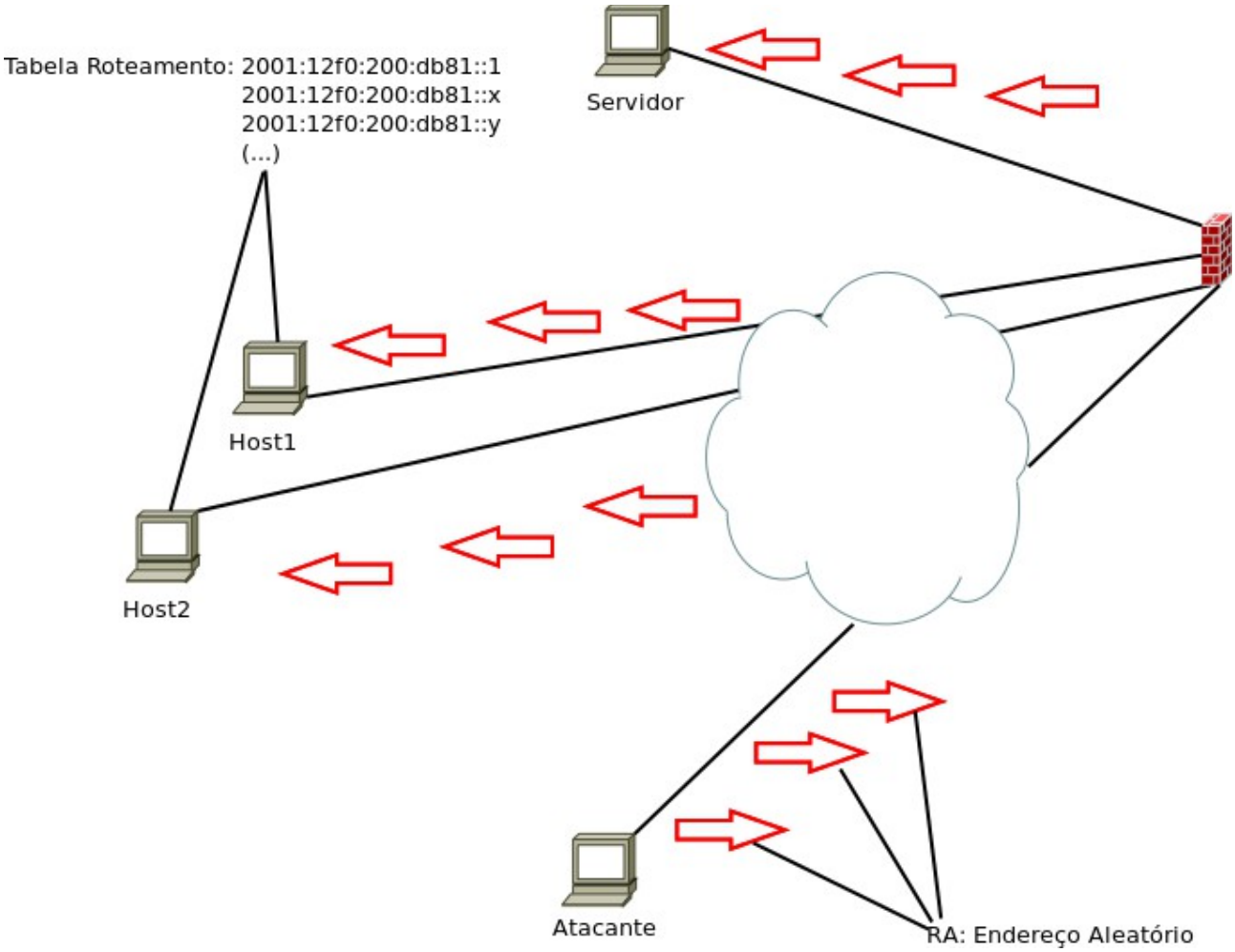

Figura 16: Representação do ataque de flooding de mensagens Router Advertisement. Fonte: própria

4.4 Cenário de testes NAT64

A técnica de tradução NAT64 permite que redes puramente IPv6 se comuniquem com *hosts* puramente IPv4, traduzindo cabeçalhos, endereços e outras informações entre as diferentes versões do protocolo IP. O uso de NAT64 implica em vulnerabilidades que afetam o dispositivo que implementa o NAT64, além das vulnerabilidades específicas do IPv6. Para que fosse possível estudar os efeitos de ataques a vulnerabilidades do protocolo IPv6 em uma rede puramente IPv6, bem como realizar testes de segurança com o próprio NAT64, foi elaborado um cenário de testes onde os *hosts* possuem apenas endereços IPv6 e se conectam à Internet através de um roteador com IPv6 e IPv4 configurados, onde roda um mecanismo de tradução NAT64/DNS64. A Figura 13 mostra um diagrama deste cenário de testes.

Este cenário é composto por um roteador, que também atua também como *firewall* e NAT64; um servidor web e três *hosts*, sendo que um dele atua como atacante na rede local. O roteador deste cenário possui três interfaces físicas de rede (eth0, eth1 e eth2) e uma interface TUN (nat64), utilizada pela ferramenta que realiza a tradução NAT64. As interfaces eth0 e eth1 estão diretamente conectadas ao roteador do PoP-SC respectivamente via IPv4 e IPv6. A interface eth2 atua como interface LAN e, portanto, possui apenas endereçamento IPv6. A interface nat64 possui endereçamento IPv4 e IPv6, uma vez que, por esta interface, passarão todos os pacotes traduzidos.

Para realizar a tradução via NAT64, foi utilizada a ferramenta Tayga. O Tayga é uma implementação de NAT64 *stateless* para Linux. Conforme mencionado anteriormente nesta seção, o Tayga utiliza uma interface TUN para interceptar e traduzir os pacotes. Como é necessário que o NAT64 opere em conjunto com DNS64, foi utilizada a ferramenta *Bind* para operar em conjunto com o Tayga, realizando a função de DNS64.

Aos *hosts*, conectados à interface eth2 do roteador, os endereços IP são atribuídos dinamicamente via RADVD. No atacante, conectado à rede local junto aos *hosts*, é utilizada a ferramenta THC-IPV6 para a realização dos ataques.

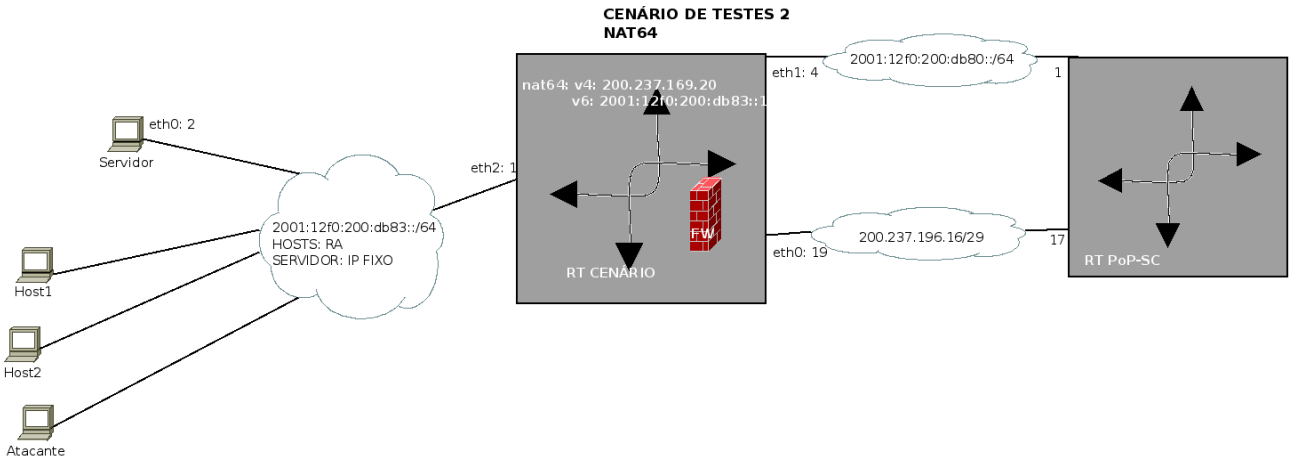

Figura 17: Cenário de testes NAT64. Fonte: própria

4.5 Experimentos realizados no cenário de testes NAT64

Neste cenário, a comunicação entre *hosts* IPv4 e *hosts* IPv6 é feita através da técnica de tradução NAT64. Tem-se, no cenário, uma rede puramente IPv6 conectada à Internet através de um dispositivo que possui ambos os protocolos nativamente e implementa a tradução NAT64, interceptando pacotes de conexões entre um *host* IPv6 da rede local do cenário e um servidor IPv4 externo e traduzindo endereços IPv6 para endereços IPv4 (e vice-versa).

Os experimentos neste cenário visavam demonstrar, além das já conhecidas vulnerabilidades específicas do IPv6, como uma rede IPv6 poderia estar vulnerável especificamente por utilizar a técnica de tradução NAT64 juntamente com o mecanismo DNS64. Além dos ataques já realizados no cenário anterior , houve também um caso de teste envolvendo ataque ao DNS, visando o mecanismo DNS64 e, consequentemente, a tradução de endereços via NAT64. Este caso de testes é descrito na Seção 4.5.1.

4.5.1 Experimento 1 – Servidor DNS falso

O DNS64 é um mecanismo muito importante para o funcionamento do NAT64, uma vez que o endereço de um servidor IPv4 a ser mapeado em um endereço IPv6 é obtido através de um registro A, a ser sintetizado em um registro AAAA. Sem a possibilidade de obter o endereço do servidor IPv4, não é possível realizar o mapeamento deste endereço em um endereço IPv6, impossibilitando a comunicação de *hosts* IPv6 com este servidor. O ataque deste experimento visa explorar esta

vulnerabilidade fazendo com que a máquina atacante atue como um servidor DNS falso, que responde a todas as consultas com um mesmo endereço IPv6.

Para o ataque deste experimento foram usados dois utilitários do THC-IPV6: o *parasite6*, que redireciona o tráfego da rede local para a máquina atacante interceptando e realizando *spoofing* de mensagens *Neighbor Solicitation*, e o *fake_dns6d*, um servidor DNS falso que responde a todas as consultas por um registro AAAA com o mesmo endereço IPv6. Optou-se por usar ambas as ferramentas conjuntamente pois desta forma o tráfego da rede local poderia ser desviado para o servidor DNS falso, o que torna o ataque mais eficaz.

A ferramenta *parasite6* redireciona o tráfego da rede local para a máquina atacante através da interceptação e *spoofing* de mensagens *Neighbor Solicitation*. Esta ferramenta recebe como parâmetro, obrigatoriamente, uma interface de rede, podendo receber opcionalmente um endereço MAC falso. Ao ser executado, este utilitário detecta mensagens *Neighbor Solicitation* na rede, intercepta estas mensagens e as envia como se fossem provenientes da máquina atacante ou de uma máquina inexistente caso seja especificado um endereço MAC falso, o que resulta em um ataque de DoS. Neste experimento, o *parasite6* foi executado recebendo uma interface de rede como parâmetro.

O utilitário *fake_dns6d* consiste em um servidor DNS falso, que responde a consultas por um registro AAAA com o mesmo endereço IPv6. Esta ferramenta recebe como parâmetro, obrigatoriamente, uma interface de rede e o endereço IPv6 utilizado nas respostas das consultas. Há ainda opções para burlar mecanismos de segurança do IPv6. Neste experimento, o *fake_dns6d* foi executado apenas com os parâmetros obrigatórios. Este endereço envolveu a máquina atacante, redirecionando o tráfego para si mesma e atuando como servidor DNS falso, e dois hosts, tentando conectividade com servidores externos. A Figura 18 ilustra o funcionamento do ataque neste experimento. Na figura, as setas pretas representam consultas ao DNS realizadas pelos *hosts*, destinadas ao servidor DNS real do cenário, enquanto as setas vermelhas representam as respostas da máquina atacante, atuando como servidor DNS.

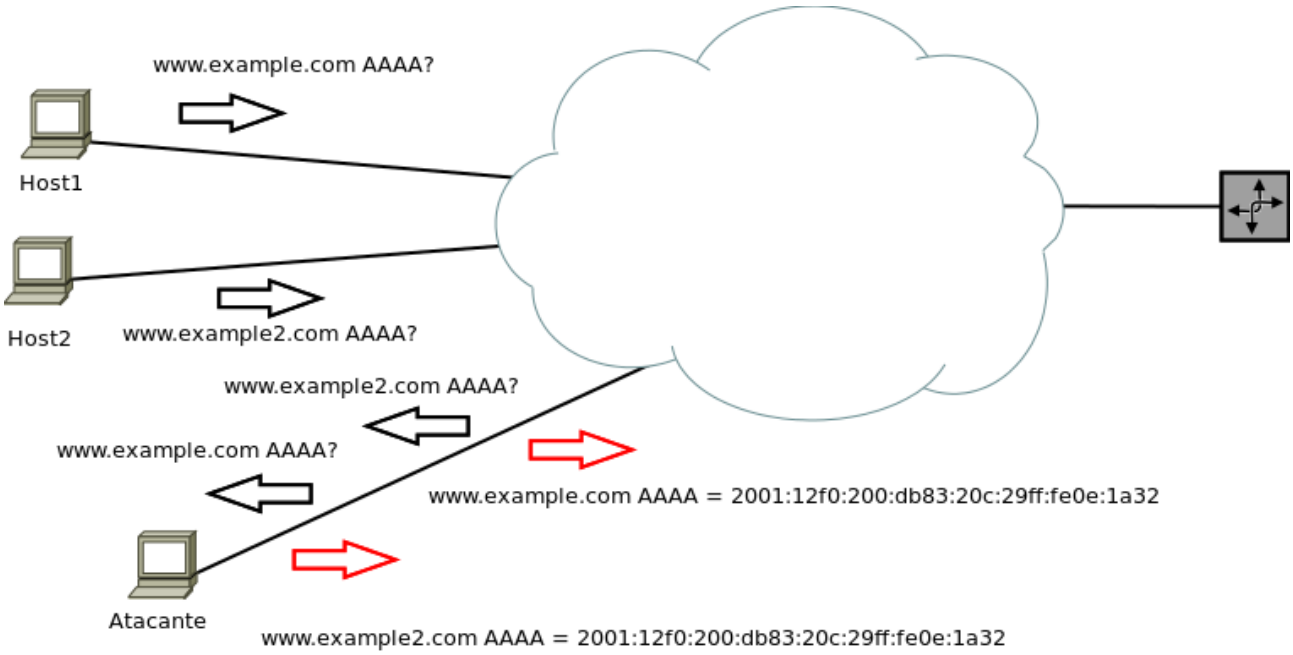

Figura 18: Representação do anúncio de um servidor DNS falso. Fonte: própria

5 Resultados

No Capítulo 5, são descritos os experimentos realizados em cada um dos cenário de testes, levando em conta os ataques realizados e ferramentas utilizadas. Cada um destes ataques foi documentado e suas consequências, analisadas. Além disto, foram levantadas formas de detecção e mitigação destes ataques. Neste capítulo, são apresentados os resultados e conclusões dos experimentos práticos e são descritas as formas de detecção e mitigação dos ataques. A Tabela 2 sumariza resultados obtidos. Na tabela, são mostrados, para cada ataque, os efeitos causados, forma de detecção e possível forma de defesa.

Ataque	**Efeito**	**Formas/Ferramentas para detecção**	**Possível forma de defesa**
DoS para novos endereços Ipv6	Impediu a obtenção de endereços Ipv6 globais por hosts em ambos os cenários	6Guard, captura de pacotes	SEND
Anúncio de um roteador falso	Acarretou em ataques de man-in-the-middle e DoS em ambos os cenários	6Guard, captura de pacotes, verificação de tabela de roteamento (cenário pilha dupla)	RA Guard
Flooding de Mensagens Neighbor Advertisement	Sobrecarregou a rede local com mensagens Neighbor Advertisement	6Guard, captura de pacotes	SEND
Flooding de Mensagens Router Advertisement	Sobrecarregou a rede local com mensagens Router Advertisement e comprometeu a configuração de interfaces e rotas dos hosts	6Guard, captura de pacotes	RA Guard
Servidor DNS falso	Comprometeu o funcionamento do DNS64 e, portanto, a tradução de endereços	Verificação do endereço associado a um nome	DNSSEC

Tabela 2: Resultados obtidos

5.1 *DoS* para novos endereços IPv6

O primeiro ataque testado nos cenários Pilha Dupla e NAT64 é o ataque de *DoS* para novos endereços IPv6. Este ataque explora a funcionalidade *Duplicate Address Detection* (DAD) fazendo com que todos os endereços IPv6 que um *host* tentar atrubuir a uma de suas interfaces de rede seja considerado em uso. As Seções 6.1.1 e 6.1.2 apresentam o resultado deste experimento nos cenários pilha-dupla e NAT64, respectivamente.

5.1.1 *DoS* para novos endereços IPv6 no cenário pilha-dupla

No cenário pilha-dupla, a ferramenta *dos_new_ip6* foi executada recebendo como parâmetro a interface eth0 da máquina atacante. Após isto, as interfaces de rede dos dois *hosts* do cenário foram reiniciadas para que eles tentassem atribuir novamente um endereço IPv6 a elas via autoconfiguração *stateless*. Os *hosts*, por sua vez, enviariam mensagens *Neighbor Solicitation* para todos os nós da rede para descobrir se o endereço IPv6 que pretendem atribuir à suas interfaces de rede já está sendo usado. Se algum outro *host* estiver usando um destes endereços IPv6, ele deve enviar uma mensagem *Neighbor Advertisement* ao *host* que pretende usar este endereço. Caso a autoconfiguração falhasse e as interfaces de rede dos *hosts* não obtivessem um endereço IPv6, o sucesso do ataque seria confirmado.

A Figura 19 mostra a execução do ataque na CLI da máquina atacante. Como pode ser visto na figura, o *dos_new_ip6* apresenta mensagens informando que foi realizado com sucesso o *spoofing* de pacotes ICMPv6 que, neste caso, são mensagens *Neighbor Advertisement* enviadas em resposta às mensagens *Neighbor Solicitation* dos *hosts* como se a máquina atacante possuísse seus endereços IPv6 pretendidos. Estas mensagens serão enviadas a qualquer *host* que tentar obter endereços IPv6 da rede local do cenário. O atacante, portanto, age como se possuísse todos os endereços IPv6 disponíveis.

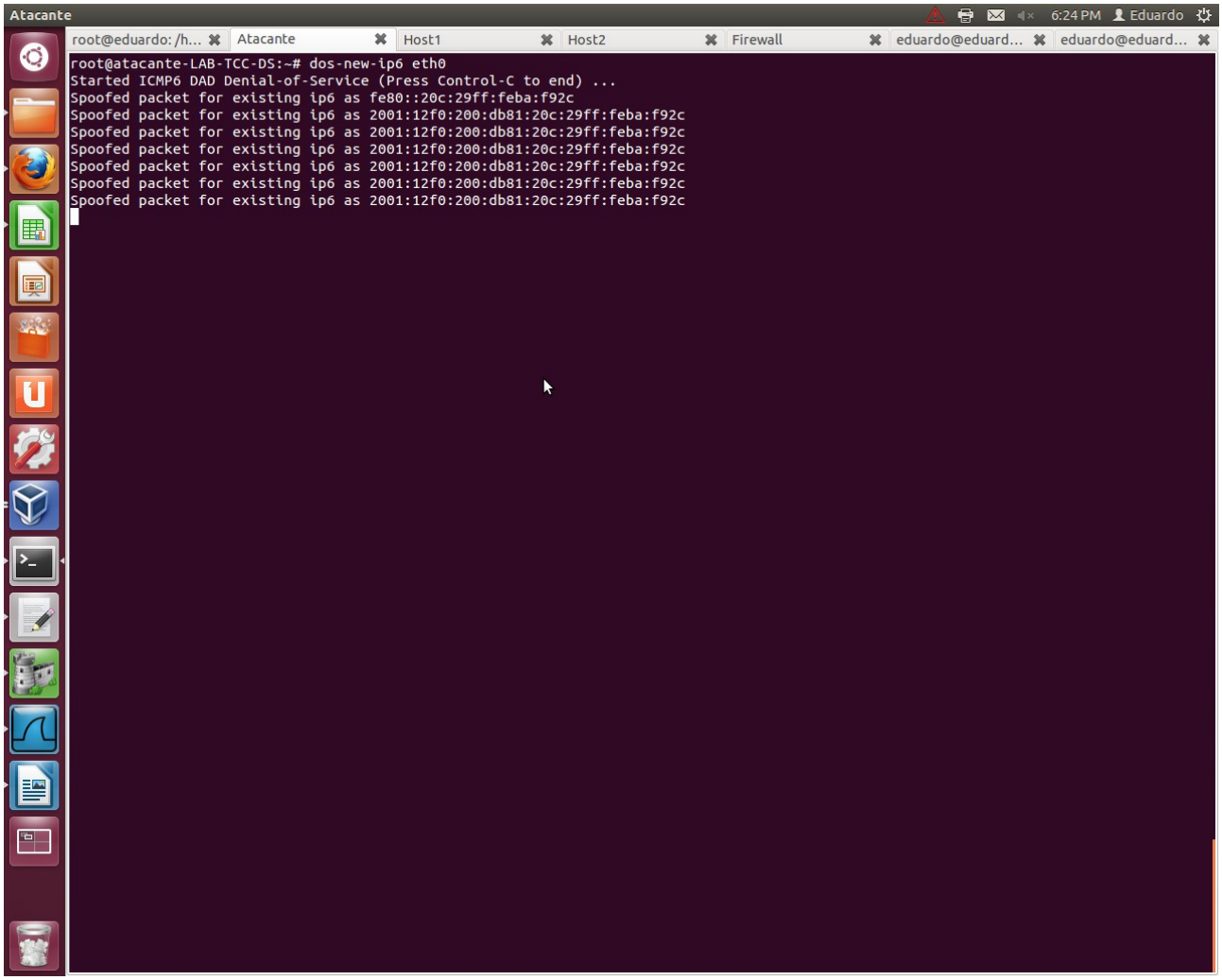

Figura 19: Máquina atacante executando ataque de DoS para endereços IPv6 novos. Fonte: própria

A Figura 20 mostra a saída de um comando *ifconfig*, utilizado para configurar e verificar a configuração de interfaces de rede no sistema operacional Linux, em um dos *hosts* do cenário, executado antes do ataque. Como pode-se observar na figura, a interface eth0 do *host* possui um endereço IPv6 link-local e um endereço IPv6 global, podendo, portanto, se conectar à Internet via IPv6. Já na Figura 21, é mostrada a saída do comando *ifconfig* após a realização do ataque. É possível observar na figura que o *host* possui apenas um endereço IPv6 link-local, o que o impossibilita de conectar-se à Internet via IPv6. Podemos observar, portanto, que a maquina atacante conseguiu impedir que as demais máquinas obsivessem endereços IPv6 globais e obteve êxito no ataque de *DoS.*

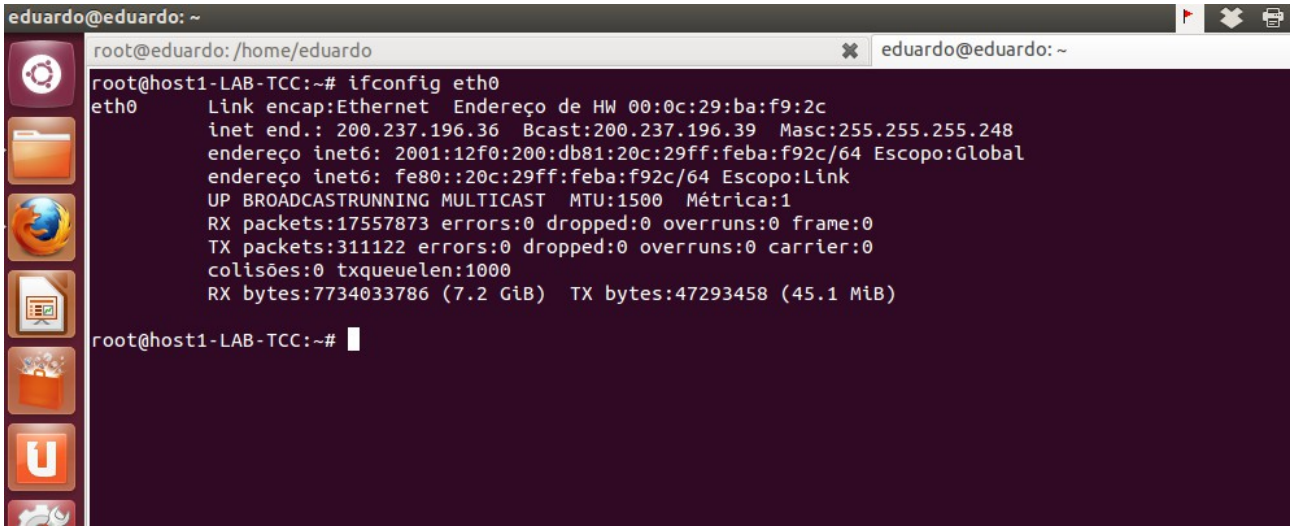

Figura 20: Configuração da interface eth0 de um host antes do ataque de DoS a endereços IPv6 novos. Fonte: própria

Figura 21: Configuração da interface eth0 de um host após o ataque de DoS a endereços IPv6 novos. Fonte: própria

5.1.2 *DoS* para novos endereços IPv6 no cenário NAT64

O ataque de *DoS* para novos endereços IPv6 foi executado no cenário NAT64 seguindo a mesma metodologia do cenário pilha-dupla, onde o atacante executou a ferramenta *dos_new_ip6* tendo sua interface eth0 como parâmetro e passou a responder todas as mensagens *Neighbor Solicitation* dos *hosts* que tentavam obter endereços IPv6 com mensagens *Neighbor Advertisement* informando que os endereços IPv6 pretendidos já estavam em uso. Assim como no cenário pilha-dupla, as interfaces de rede dos demais *hosts* do cenário foram reiniciadas para que os mesmos

tentassem obter endereços IPv6 globais novamente.

A Figura 22 mostra a execução da ferramenta *dos_new_ip6* na máquina atacante do cenário NAT64. É possível observar na figura as mensagens do *dos_new_ip6* indicando que foi feito o *spoofing* de mensagens *Neighbor Advertisement* enviadas aos *hosts* informando que os endereços IPv6 que pretendem configurar em suas interfaces estão supostamente em uso, enquanto não estão em uso de fato e as mensagens *Neighbor Advertisement* são ilegítimas.

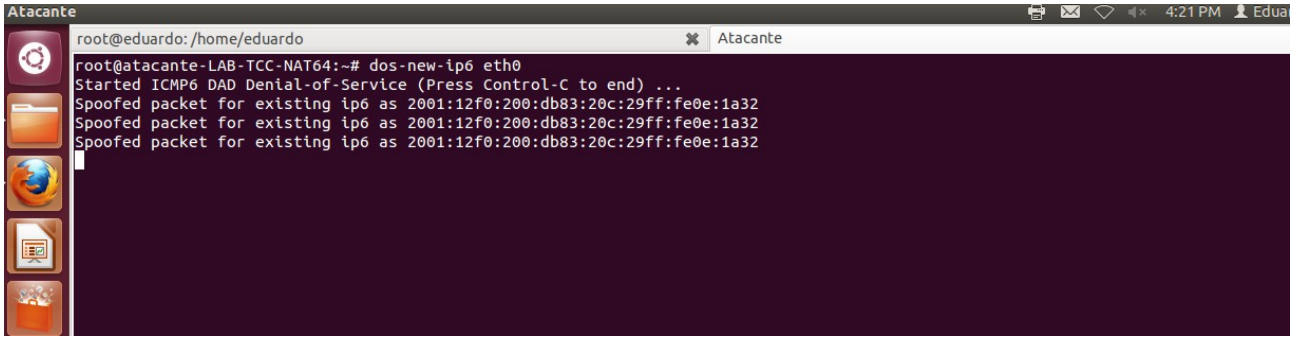

Figura 22: Execução da ferramenta dos_new_ip6 na máquina atacante do cenário NAT64. Fonte: própria

Na Figura 23, é mostrada a saída de um comando *ifconfig* em um dos *hosts* do cenário antes da execução do ataque. Pode-se verificar que esta interface possui um endereço IPv6 global, o que possibilida o *host* de se conectar à Internet. Ressalta-se que, neste caso, há apenas endereços IPv6 configurados na interface de rede deste *host*, pois o mesmo está em uma rede puramente IPv6. Na Figura 24 vemos a saída do comando *ifconfig* após a execução do ataque. Podemos observar na figura que, assim como occoreu no cenário pilha-dupla, o *host* não obteve um endereço IPv6 e, portanto, o que o impossibilita de conectar-se à Internet.

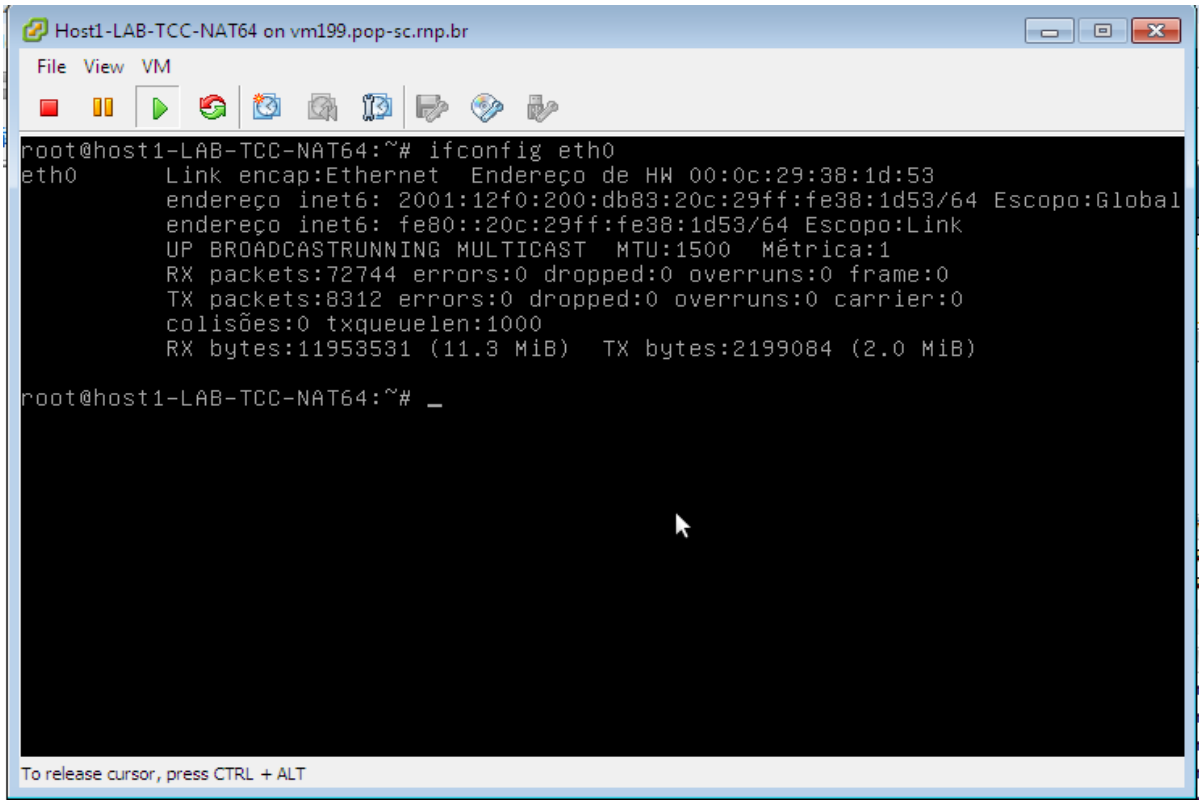

Figura 23: Configuração da interface eth0 de um host do cenário NAT64 antes da execução do ataque de DoS a endereços IPv6 novos. Fonte: própria

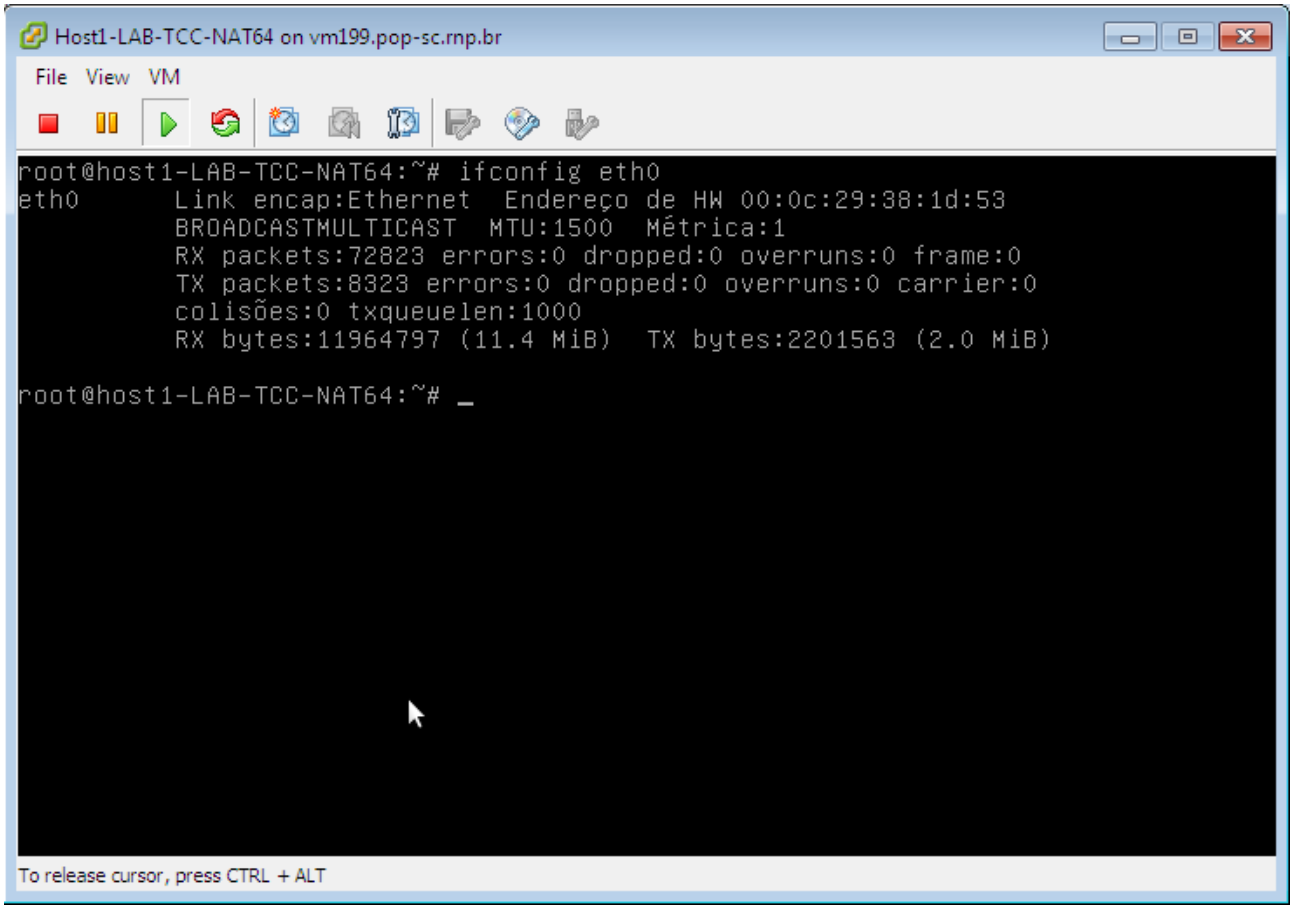

Figura 24: Configuração da interface eth0 de um host do cenário NAT64 após a execução do ataque de DoS a endereços IPv6 novos. Fonte: própria

5.2 Anúncio de um roteador falso

Este ataque, executado nos cenários pilha-dupla e NAT64, consiste em enviar mensagens *Router Advertisement* falsas, anunciando a máquina atacante como roteador e direcionando o tráfego da rede local para o atacante. Conforme mencionado na Seção 4.3.4, este ataque foi executado sob duas abordagens diferentes: redirecionar o tráfego da rede local para a máquina atacante (caractericando um ataque de *man-in-the-middle*) e o anúncio de um roteador com endereço IPv6 inexistente, acarretando em um ataque de *DoS*.

5.2.1 Anúncio de um roteador falso no cenário pilha-dupla – *man-in-the-middle*

Para executar um ataque de *man-in-the-middle* através do envio de mensagens *Router Advertisement* falsas, foi executada a ferramenta *fake_router26* sem opções, tendo como parâmetro apenas a interface eth0 da máquina atacante, o que faz com que as mensagens *Router Advertisement* enviadas anunciem a própria máquina atacante como roteador, isto é, as mensagens *Router Advertisement* tem como endereço IPv6 e MAC de origem os endereços IPv6 e MAC da máquina atacante. Com isto, o tráfego que parte da rede local do cenário é redirecionado para a máquina atacante, possibilitando à mesma a captura deste tráfego. A execução da ferramenta *fake_router26* para este ataque é mostrada na Figura 25.

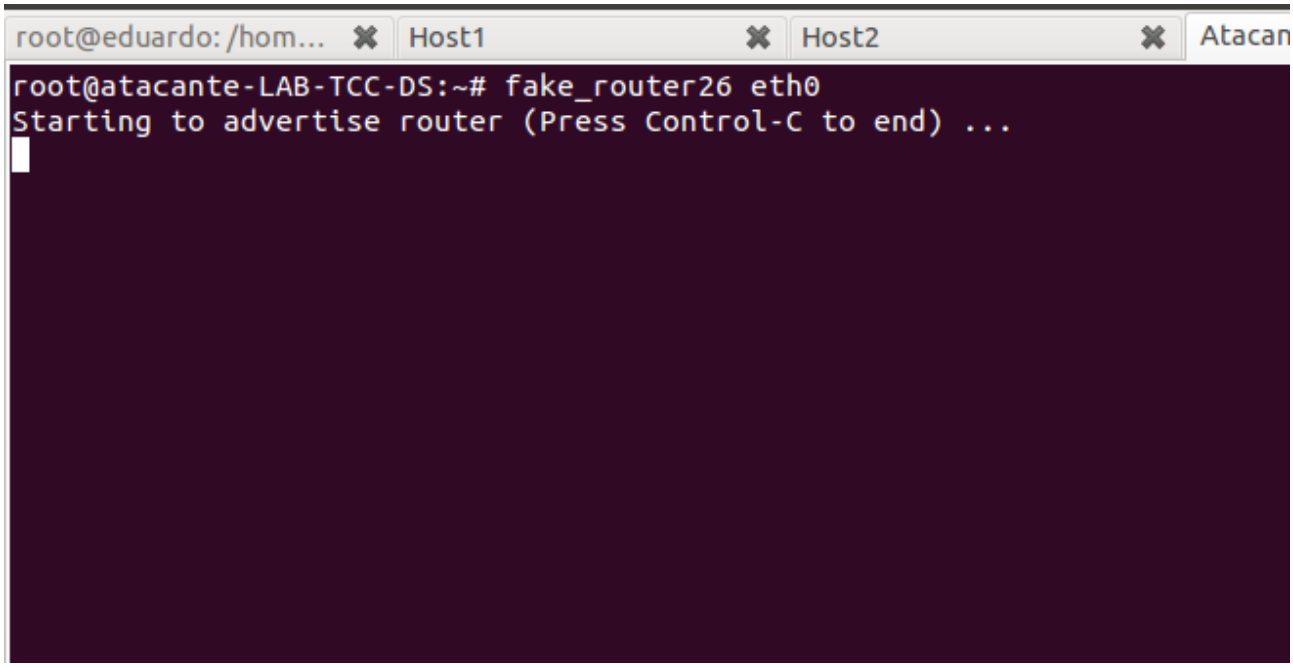

Figura 25: Execução da ferramenta fake_router26 no cenário pilha-dupla. Fonte: própria

Caso este ataque seja bem sucedido, ele alterará, primeiramente, as configurações de rota padrão dos *hosts*. A Figura 26 mostra a execução do comando *route -n6* em um dos *hosts* do cenário antes da execução do ataque. Este comando mostra a tabela de roteamento de um *host* Linux, sendo que o parâmetro *-n6* especifica que deve ser mostrada a tabela de roteamento IPv6. Na Figura 26, pode-se observar apenas um endereço IPv6 como rota padrão deste *host*, neste caso, o endereço do roteador real. Já a Figura 27 mostra a execução do comando *route -n6* após a execução do ataque. Pode-se observar na figura que há dois endereços IPv6 como rota padrão, sendo que o segundo endereço pertence à máquina atacante e foi configurado a partir das mensagens *Router Advertisement* falsas.

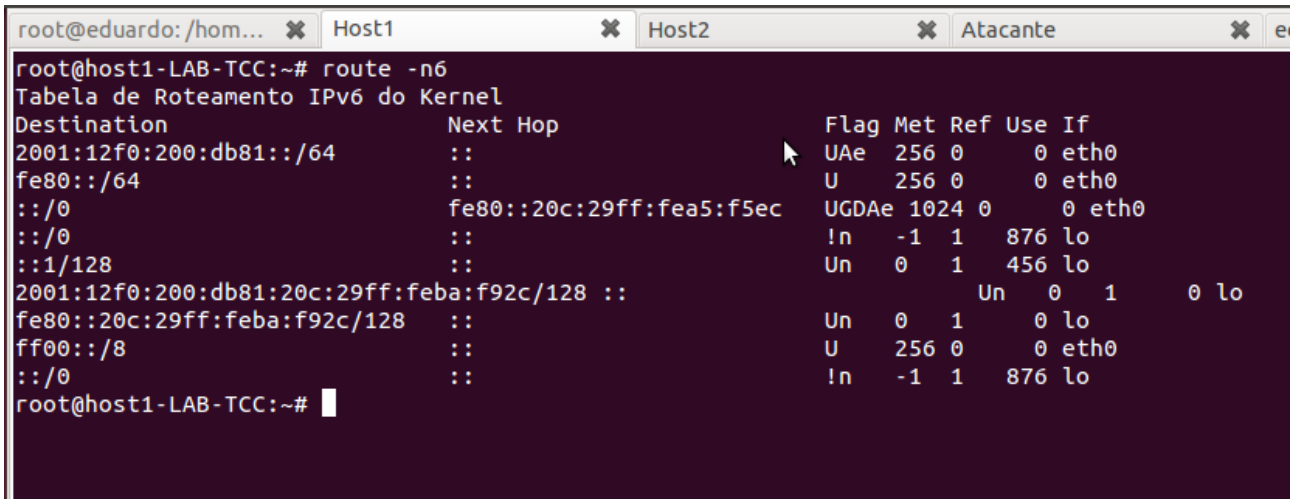

Figura 26: Saída do comando route -n6 antes do anúncio do roteador falso. Fonte: própria

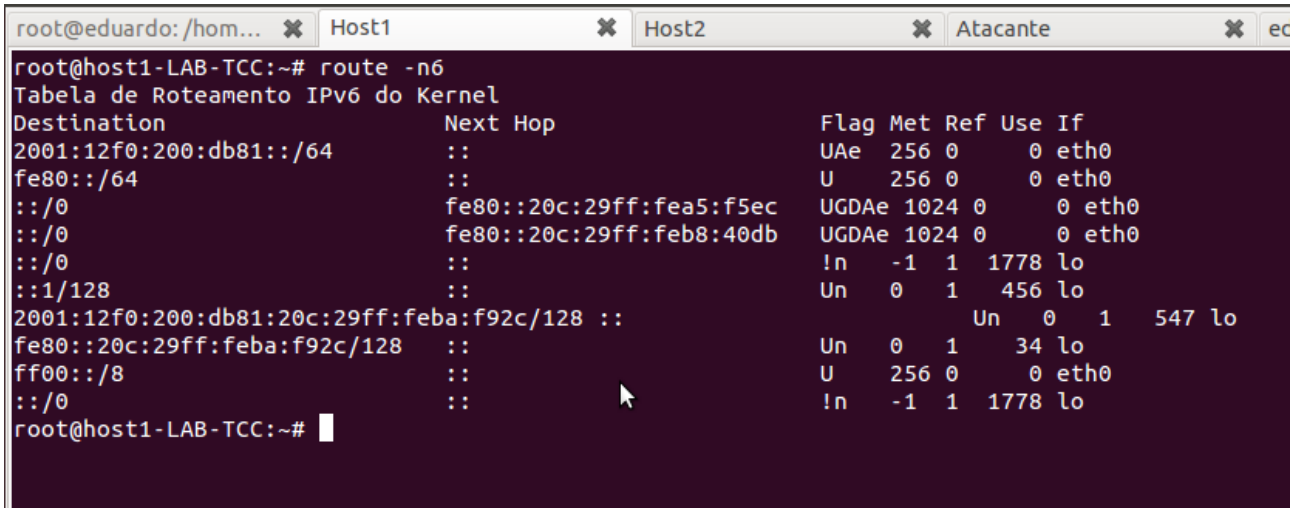

Figura 27: Saída do comando route -n6 após o anúncio do roteador falso. Fonte: própria

Após a confirmação de que as configurações de rota padrão dos *hosts* haviam sido alteradas em função do ataque, foram realizados testes de conectividade. Em um dos *hosts*, foi executado o comando *ping6* para um servidor externo, testando a conectividade via ICMPv6. Em outro *host*, foi testada a conectividade via HTTP no servidor web do cenário através do comando *wget*.

Na Figura 28 é mostrada a saída do comando *ping6* no primeiro *host*. Pode-se observar que as mensagens ICMPv6 *echo request* recebiam resposta (*echo reply*). Porém, a taxa de perdas de pacotes chegou a 94%, o que mostra que este ataque também teve o efeito de prejudicar bastante a conectividade com a Internet.

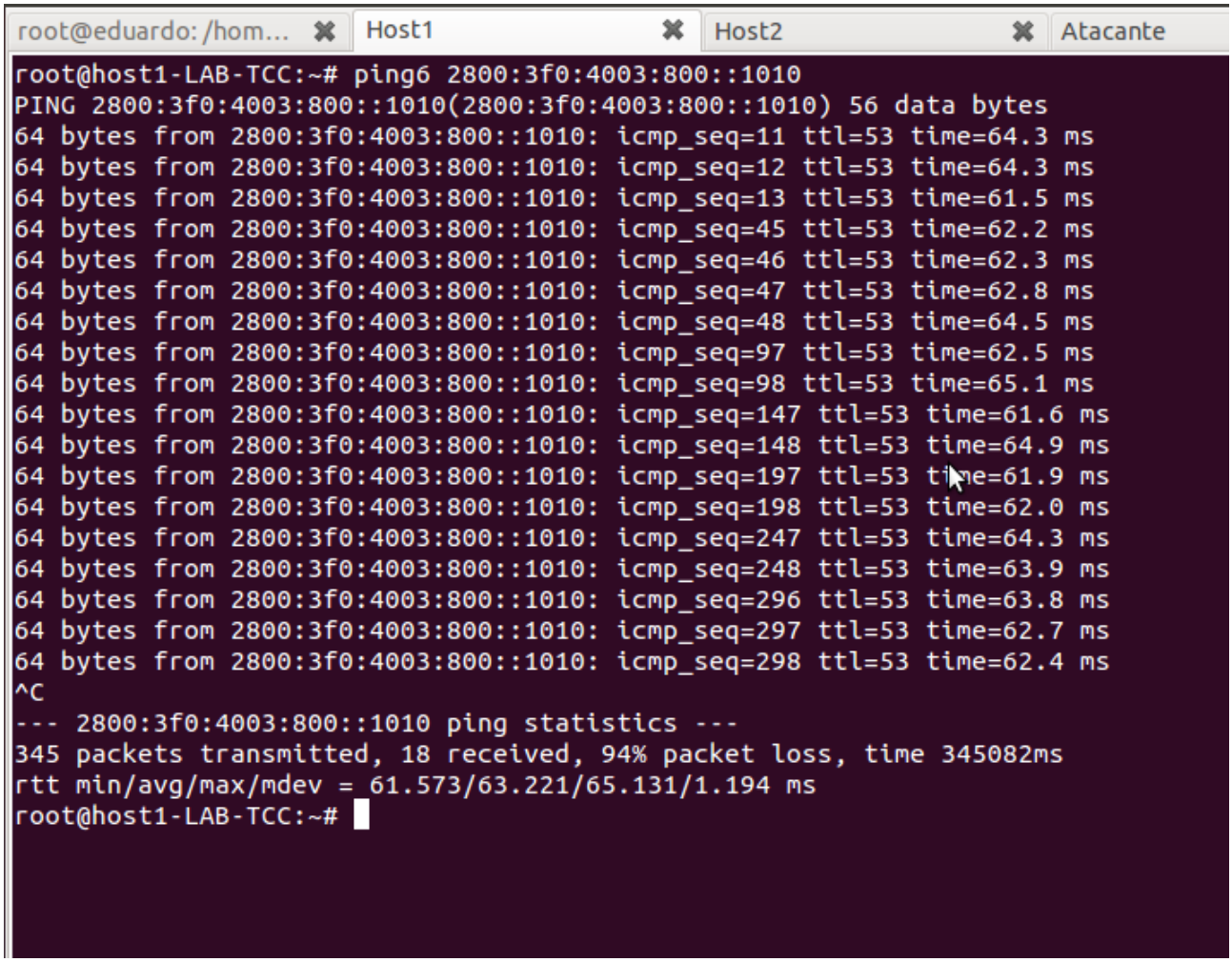

Figura 28: Teste de conectividade com um servidor externo via ping6. Fonte: própria

Na Figura 29 é mostrada a saída do comando *wget* no segundo *host*. Pode-se observar que a conexão do *host* com o servidor web do cenário é realizada com sucesso em quase todas as tentativas, falhando apenas em uma. Isto mostra que a conectividade com o servidor interno foi pouco prejudicada. No entanto, o tráfego entre os *hosts* e qualquer servidor, interno ou externo, pode ser fácilmente capturado. A Figura 30 mostra uma captura de pacotes realizada a partir da máquina atacante. Podemos ver na figura as mensagens *echo request* que o primeiro *host* envia ao servidor externo, com o objetivo de realizar um teste de conectividade. Isto mostra que o tráfego dos *hosts* está passando pela máquina atacante, o que significa que o atacante tem acesso à comunicação dos *hosts* com qualquer servidor e pode facilmente vizualizar dados não protegidos, caracterizando-se então um bem sucedido ataque de *man-in-the-middle* explorando o mascaramento de pacotes ICMPv6 do tipo *Router Advertisement*, uma vulnerabilidade não prevista em redes IPv4.

```
root@eduardo: /home/eduardo   Atacante   Host1   Host2
root@Host2-LAB-TCC-DS:~# wget http:///[2001:12f0:200:db8b::2]
--2014-06-03 17:04:38--  http:///[2001:12f0:200:db8b::2]/
Conectando-se a 2001:12f0:200:db8b::2:80... conectado.
A requisição HTTP foi enviada, aguardando resposta... 200 OK
Tamanho: 177 [text/html]
Salvando em: “index.html”

100%[======================================================================================>] 177         --.-K/s   em 0s

2014-06-03 17:04:45 (26,8 MB/s) - “index.html” salvo [177/177]

root@Host2-LAB-TCC-DS:~# wget http:///[2001:12f0:200:db8b::2]
--2014-06-03 17:04:50--  http:///[2001:12f0:200:db8b::2]/
Conectando-se a 2001:12f0:200:db8b::2:80... falhou: Tempo esgotado para conexão.
Tentando novamente.

--2014-06-03 17:05:54--  (tentativa: 2)  http:///[2001:12f0:200:db8b::2]/
Conectando-se a 2001:12f0:200:db8b::2:80... conectado.
A requisição HTTP foi enviada, aguardando resposta... 200 OK
Tamanho: 177 [text/html]
Salvando em: “index.html.1”

100%[======================================================================================>] 177         --.-K/s   em 0s

2014-06-03 17:06:01 (25,8 MB/s) - “index.html.1” salvo [177/177]

root@Host2-LAB-TCC-DS:~#
```

Figura 29: Teste de conectividade via HTTP com o servidor interno. Fonte: própria

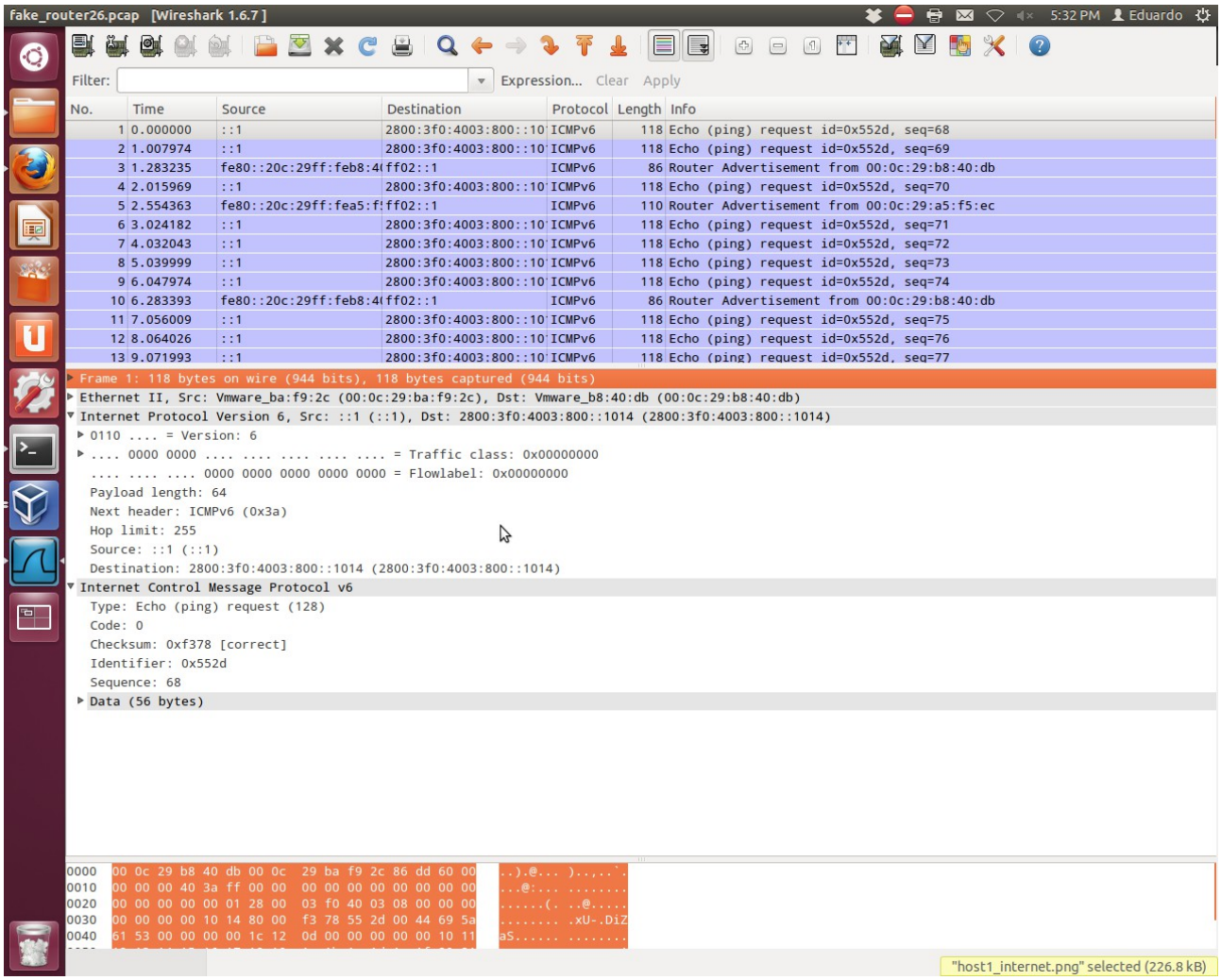

Figura 30: Captura de pacotes a partir da máquina atacante. Fonte: própria

5.2.2 Anúncio de um roteador falso no cenário pilha dupla – *DoS*

Outra abordagem para atacar a rede local anunciando um roteador falso é realizar um ataque de *DoS* através do anúncio de um roteador inexistente. Isto pode ser feito através do envio de mensagens *Router Advertisement* que tem como endereço IPv6 e MAC de origem endereços IPv6 e MAC que não estão atribuídos a nenhum outro *host.* Desta forma, os *hosts* que recebem estas mensagens *Router Advertisement* adicionam às suas tabelas de roteamento uma rota padrão para um roteador inexistente e, portanto, os pacotes enviados a este roteador são descartados. Para realizar este ataque, foi utilizada a ferramenta *fake_router26* executada com a opção *-s <ip-origem>*, espeficicando-se um endereço IPv6 inexistente no parâmetro *ip-origem.* A Figura 31 mostra a execução do *fake_router26* para este ataque.

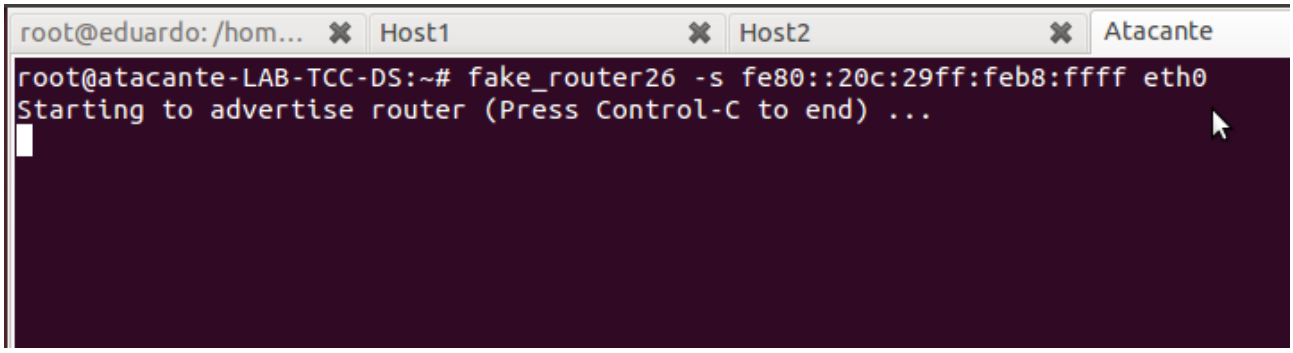

Figura 31: Execução da ferramenta fake_router26 para gerar um ataque de DoS através do envio de mensagens Router Advertisement falsas. Fonte: própria

Para verificar os efeitos do ataque, inicialmente verificou-se a alteração da configuração de rotas padrão nos *hosts*. A Figura 32 mostra a execução do comando *route -n6* em um dos *hosts* do cenário antes do ataque. Como é esperado, observa-se que a tabela de roteamento deste *host* possui apenas uma rota padrão, neste caso a rota para o roteador real do cenário. Já na Figura 33, que mostra a saída do comando *route -n6* após o ataque, observa-se a existência se uma segunda rota padrão, indicando que os *hosts* do cenário, ao receberem as mensagens *Router Advertisement* falsas, configuraram rotas para o roteador falso anunciado.

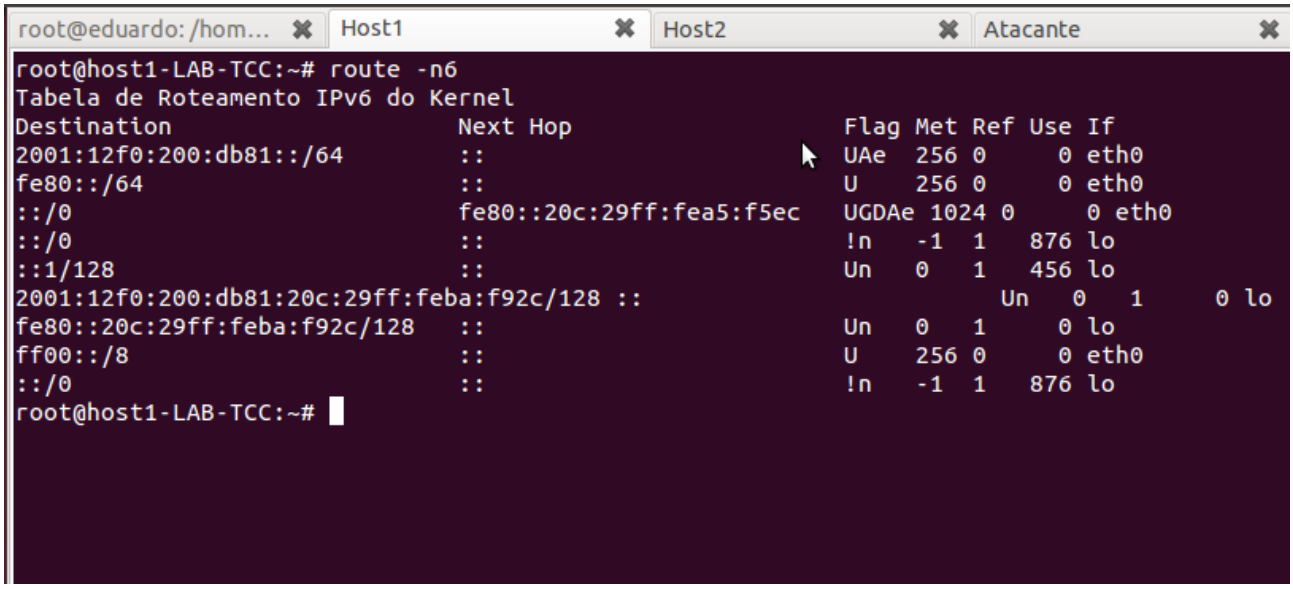

Figura 32: Saída do comando route -n6 antes do anúncio do roteador falso. Fonte: própria

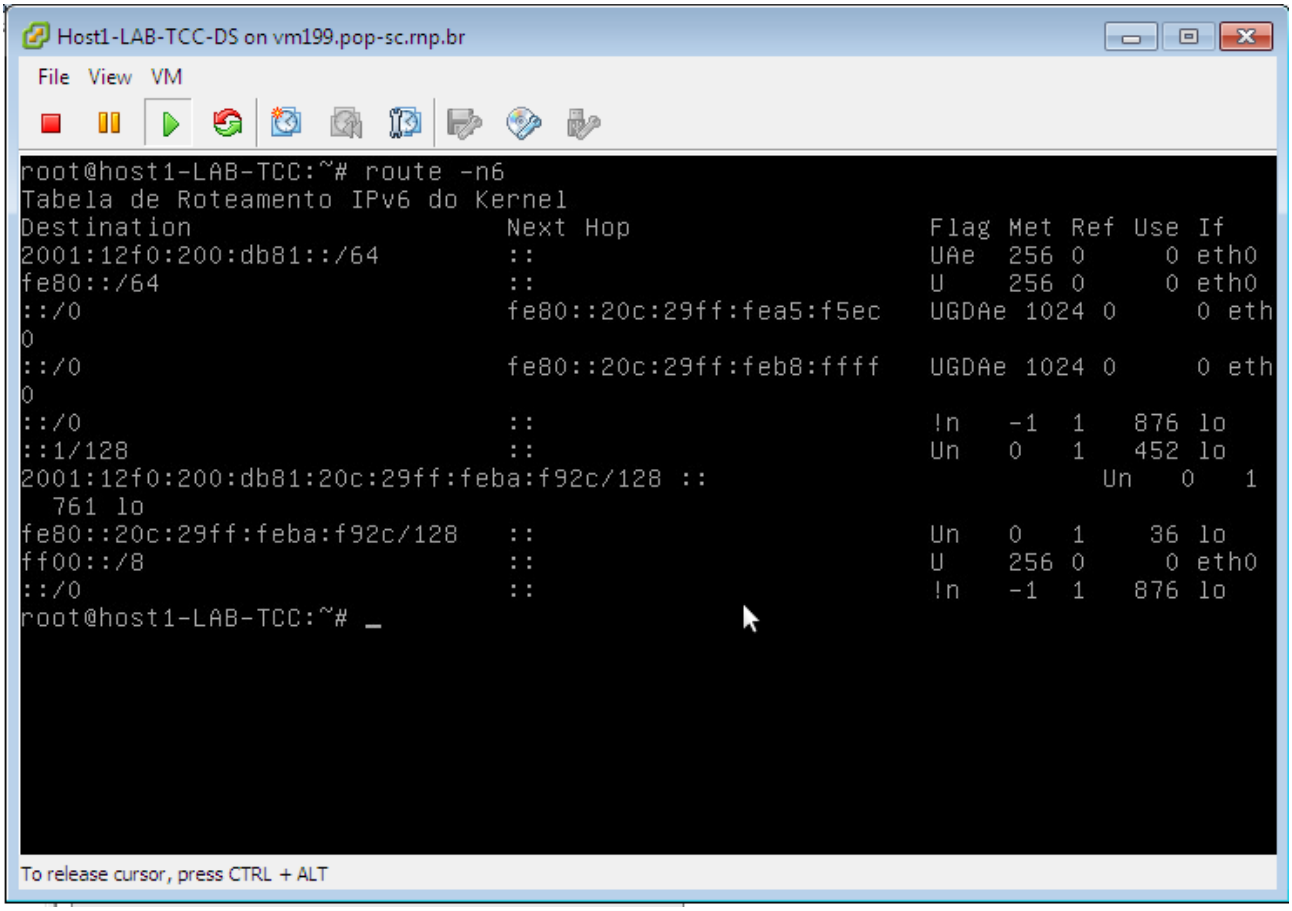

Figura 33: Saída do comando route -n6 após o anúncio do roteador falso. Fonte: própria

Após a verificação da alteração das configurações de rota padrão dos *hosts*, foram realizados testes de conectividade tanto para a Internet quanto para a própria rede do cenário. Primeiramente, foi executado um teste de conectividade via ICMPv6 através do comando *ping6* para um servidor externo. A saída do comando *ping6* executado é mostrada na Figura 34. como pode ser observado na figura, houve uma taxa de perda de pacotes de 100%. Esta taxa de perda de pacotes mostra que o *host* não consegue comunicação com o servidor externo e com a Internet pois, em função do ataque, os pacotes enviados pelo *host* são encaminhados para um roteador inexistente e, portanto, acabam sendo descartados.

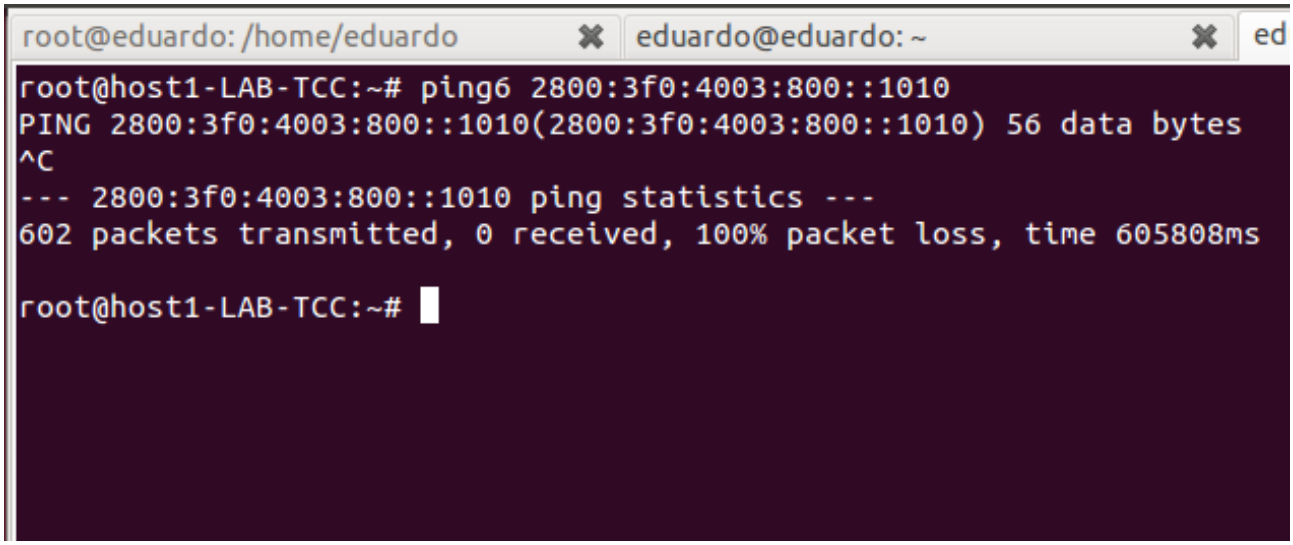

Figura 34: Teste de conectividade com servidor externo via ICMPv6. Fonte: própria

Após a verificação da conectividade via ICMPv6, foi realizado um teste de conectividade com o servidor web do cenário via HTTP utilizando o comando *wget*. A saída do comando é mostrada na Figura 35. Observa-se na figura que, ao contrário do ocorrido no ataque de *man-in-the-middle* através do anúncio de um roteador falso, não foi possível conectar-se ao servidor do cenário via HTTP em nenhuma das tentativas, também em função do desvio dos pacotes para um roteador inexistente.

Figura 35: Teste de conectividade com o servidor do cenário via HTTP. Fonte: própria

Os testes executados após o ataque mostram que a conectividade dos *hosts* interna e externamente ao cenário foi totalmente comprometida pelo anúncio de um roteador inexistente. Constata-se, portanto, que é possível utilizar mensagens *Router Advertisement* mascaradas para provocar com sucesso um ataque de *DoS* em uma rede local IPv6.

5.2.3 Anúncio de um roteador falso no cenário NAT64 – *man-in-the-middle*

Seguindo a metodologia do ataque de *man-in-the-middle* através do anúncio de um roteador falso no cenário pilha dupla, este ataque foi realizado também no cenário NAT64. Para a realização do ataque, foi executada a ferramenta *fake_router26* tendo como parâmetro apenas a interface eth0 da máquina atacante. Desta forma, a máquina atacante envia mensagens *Router Advertisement* com seus próprios endereços IPv6 e MAC como endereços de origem, fazendo com que o tráfego da rede local seja desviado para ela. A execução do *fake_router26* é mostrada na Figura 36.

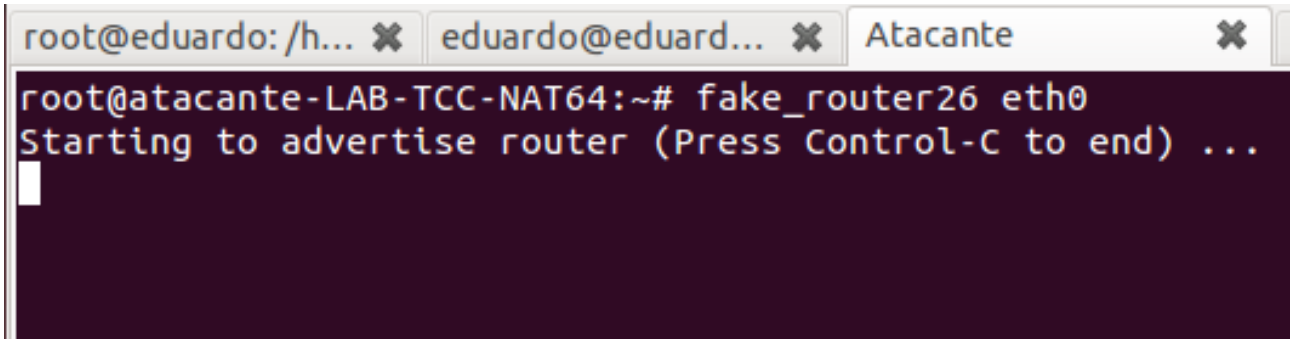

Figura 36: Execução do fake_router26 para realizar um ataque de man-in-the-middle na máquina atacante do cenário NAT64. Fonte: própria

O primeiro indício de sucesso deste ataque é a alteração das configurações de rota padrão dos *hosts* do cenário. A alteração das configurações de rota padrão foram verificadas através do comando *route -n6*. A Figura 37 mostra a saída deste comando em um dos *hosts* antes do início do ataque. Na figura, é mostrada a configuração esperada, com apenas uma rota padrão, ou seja, a do roteador do cenário. Já na Figura 38, pode ser vista a configuração de rota padrão deste mesmo *host* após o início do ataque. Percebe-se que há uma outra rota padrão na tabela de roteamento do *host*. Esta segunda rota foi configurada a partir das mensagens *Router Advertisement* enviadas pela máquina atacante e, portanto, destinará o tráfego da rede local para a mesma.

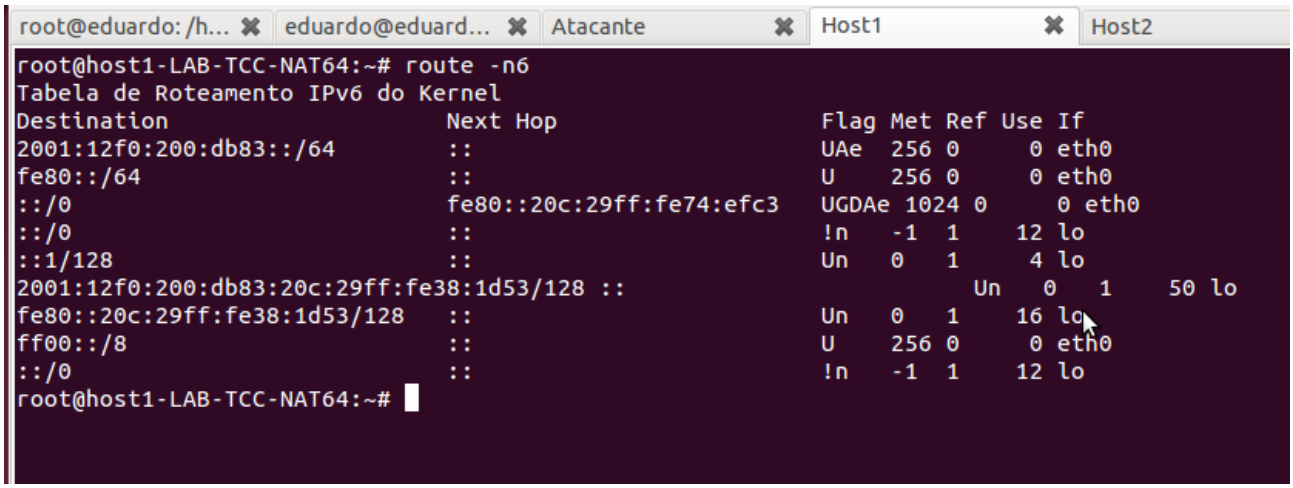

Figura 37: Saída do comando route -n6 antes do anúncio do roteador falso. Fonte: própria

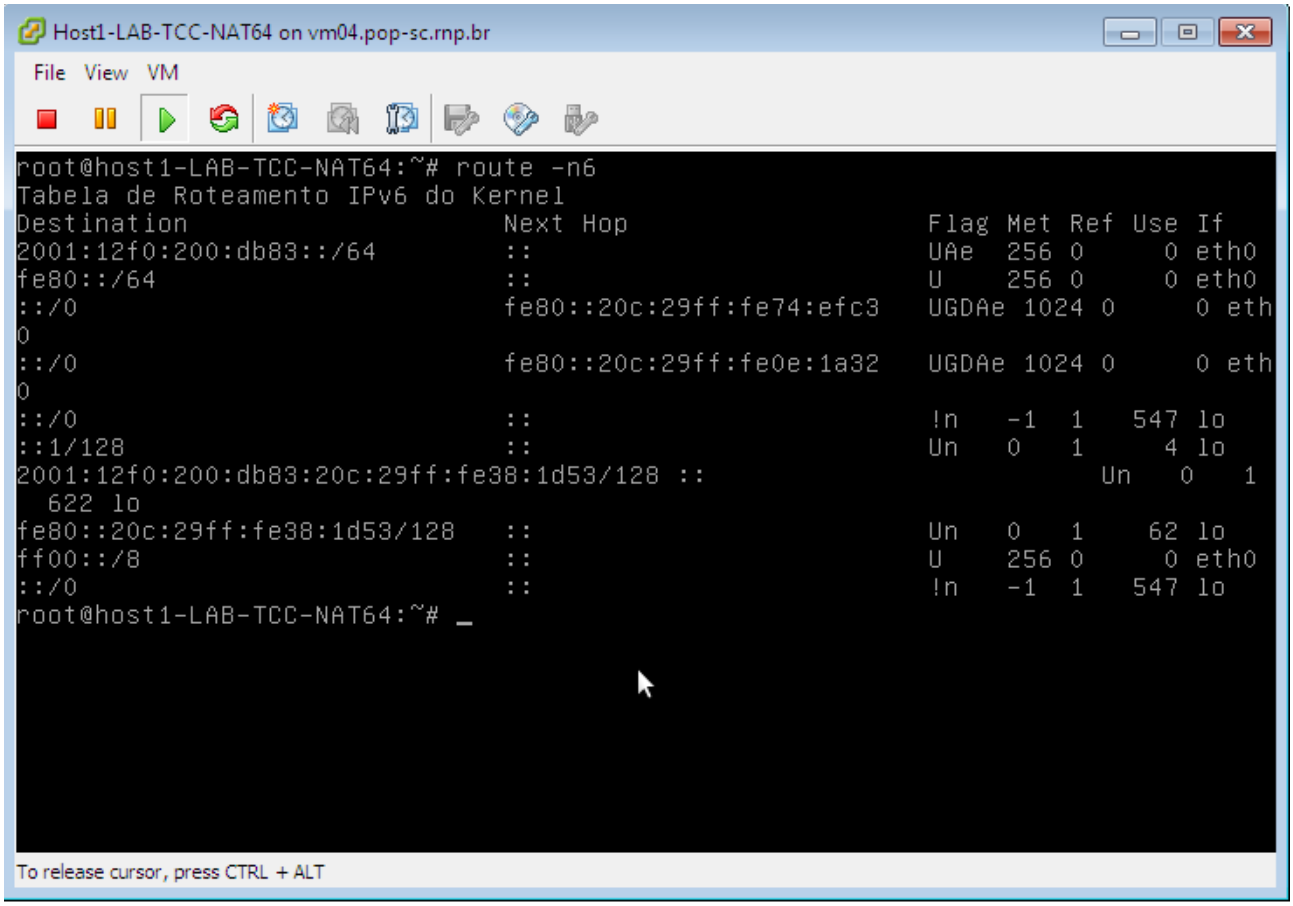

Figura 38: Saída do comando route -n6 após o anúncio do roteador falso. Fonte: própria

Assim como no cenário pilha dupla, após a verificação da alteração de configurações de rota padrão, foram executados testes de conectividade. Em um dos *hosts* do cenário realizou-se um teste de conectividade com um servidor externo via ICMPv6 através do comando *ping6*. No outro *host*,

foi realizado um teste de conectividade via HTTP através do comando *wget.*

A Figura 39 mostra a saída do comando *ping6*, utilizado no teste de conectividade realizado no primeiro *host*. Como pode ser observado na figura, algumas mensagens *echo-request* recebiam resposta (*echo-reply)*, porémo houve uma alta taxa de perda de pacotes, chegando a 92%.

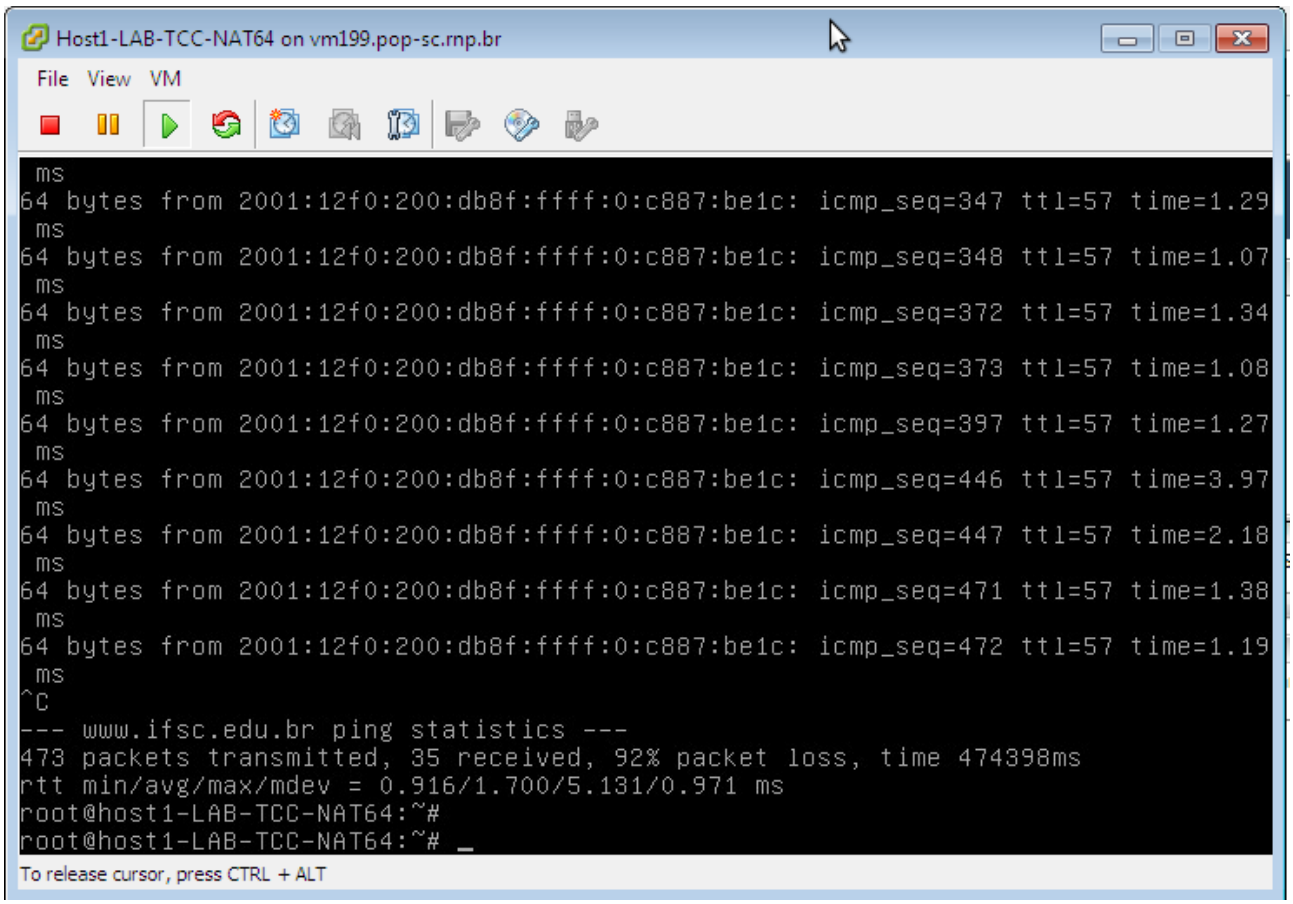

Figura 39: Teste de conectividade com um servidor externo via ICMPv6. Fonte: própria

Na Figura 40, pode ser vista a saída do comando *wget*. Como pode ser observado na figura, as conexões com o servidor via HTTP foram bem sucedidas em todas as tentativas, ou seja, neste cenário, a conectividade com o servidor interno não foi afetada. No entanto, assim como ocorreu no cenário pilha dupla, o tráfego entre da rede local pôde ser capturado pela máquina atacante. Na Figura 41 é mostrada uma captura de pacotes realizada pela máquina atacante. Pode-se observar, por exemplo, que a máquina atacante conseguiu capturar algumas das mensagens ICMPv6 destinadas ao servidor externo. Isto significa que, assim como no cenário pilha dupla, a máquina atacante consegue redirecionar para si e capturar o tráfego da rede local, vizualizando facilmente dados não protegidos. No cenário NAT64, o ataque de *man-in-the-middle* através do anúncio de um roteador falso também foi bem sucedido.

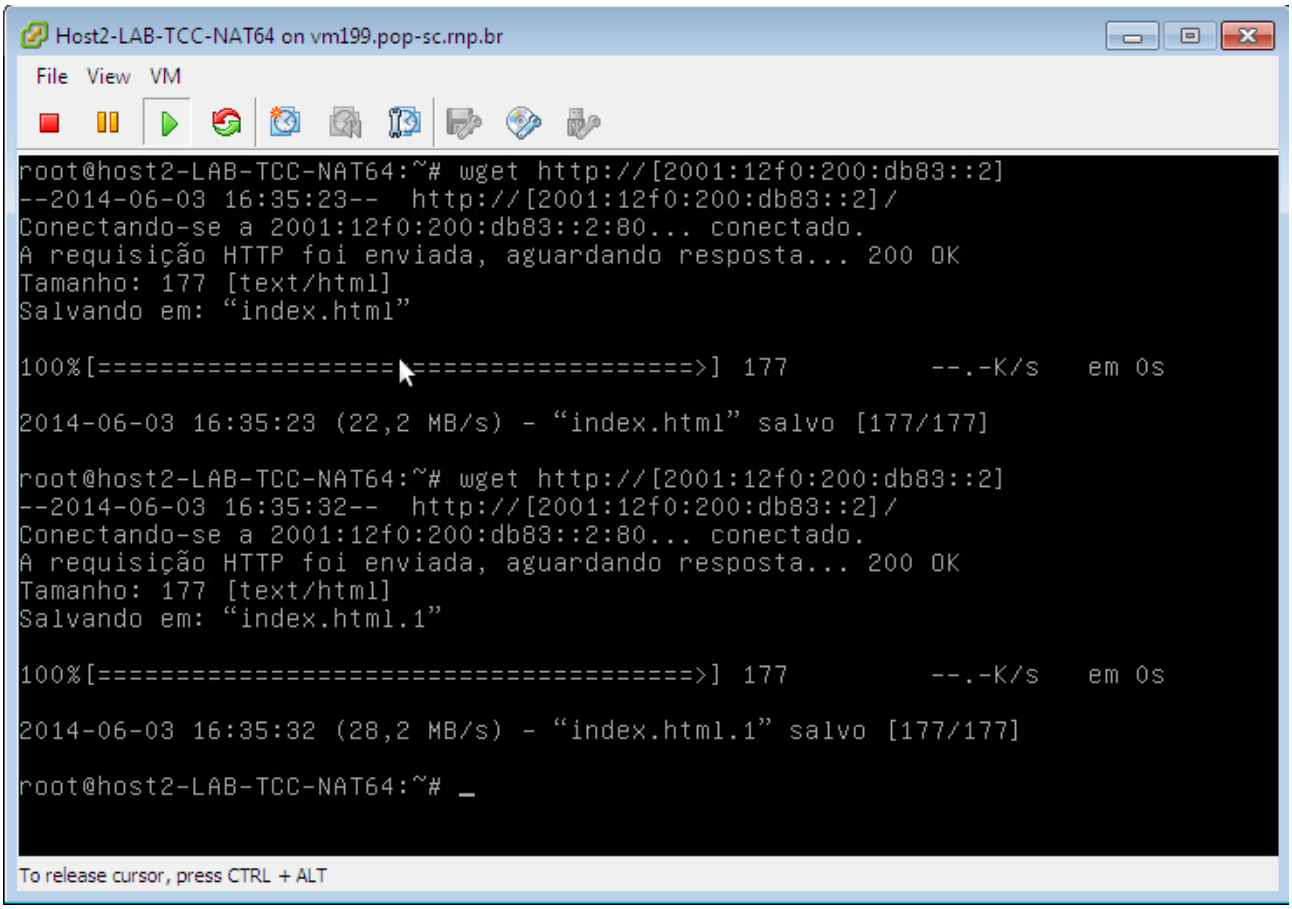

Figura 40: Teste de conectividade com o servidor interno via HTTP. Fonte: própria.

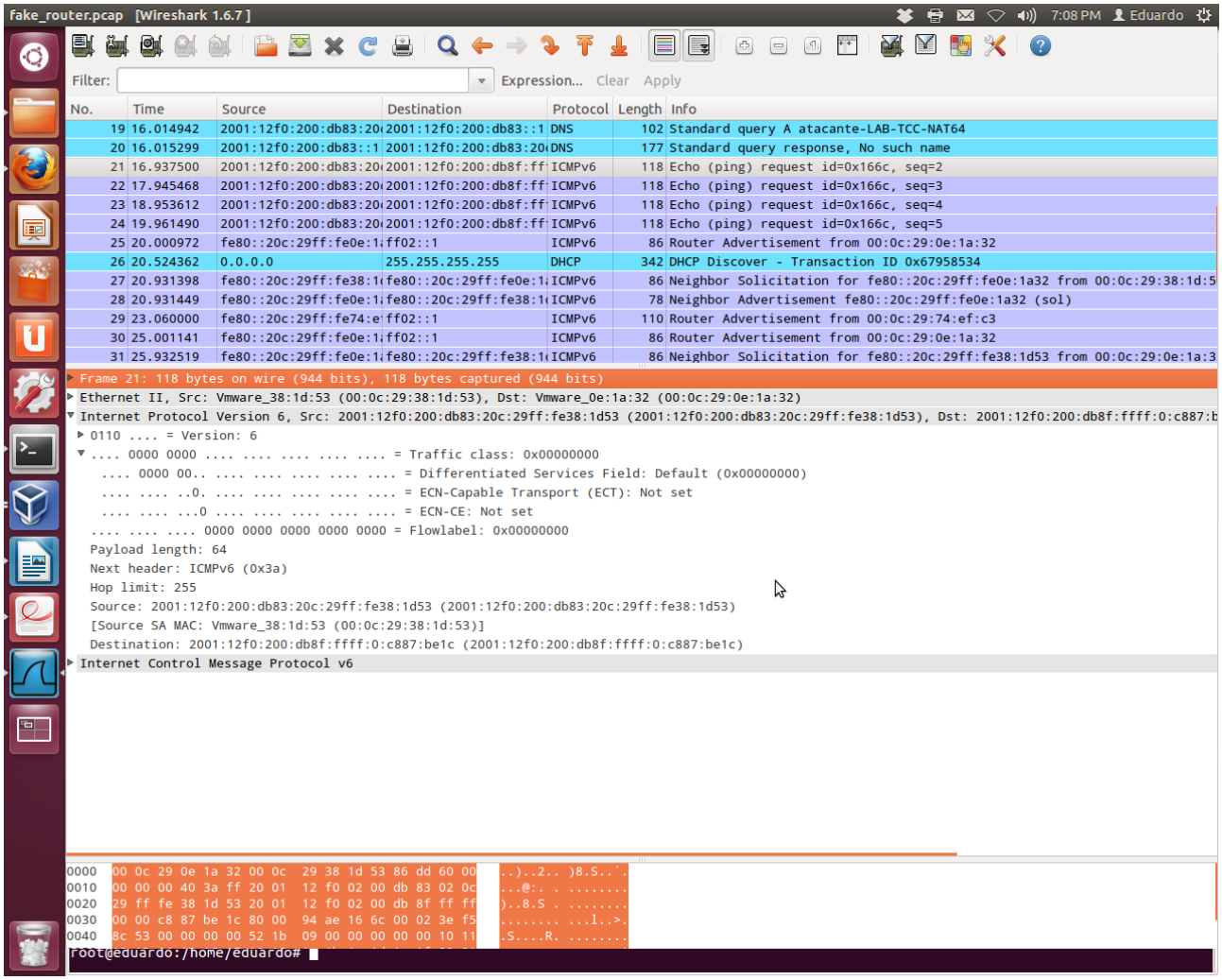

Figura 41: Captura de pacotes a partir da máquina atacante. Fonte: própria.

5.2.4 Anúncio de um roteador falso no cenário NAT64 – *DoS*

A abordagem do anúncio de um roteador falso para gerar um ataque de *DoS* também foi explorada no cenário NAT64. Assim como no cenário pilha dupla, para este ataque, foi executada a ferramenta *fake_router26* com a opção *-s <ip-origem>*, sendo que o parâmetro *<ip-origem>* corresponde a um endereço IPv6 inexistente. A Figura 42 mostra a execução do *fake_router26* para este ataque.

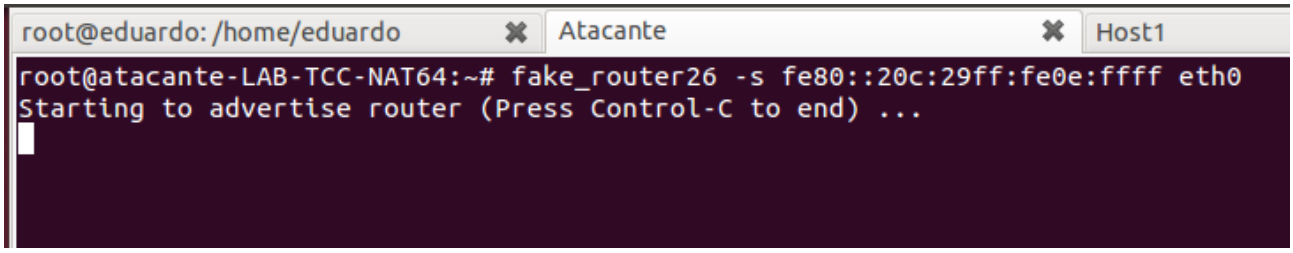

Figura 42: Execução do fake_router26 para realizar um ataque de DoS na máquina atacante do cenário NAT64. Fonte: própria

Para verificar os efeitos do ataque, é necessário, primeiramente, verificar as configurações de rota padrão dos *hosts* do cenário. Assim como no cenário pilha dupla, a verificação de configurações de rota padrão foi feita através do comando *route -n6*. A Figura 43 mostra a configuração de rotas padrão para um dos *hosts* do cenário antes do início do ataque. Na figura, nota-se que há apenas uma rota padrão na tabela de roteamento deste *host* e esta rota é, como o esperado, para o roteador do cenário. Já na Figura 44, é mostrada a configuração de rotas padrão deste mesmo *host* após o início do ataque. Pode-se observar que há uma segunda rota padrão na tabela de roteamento deste *host* que, neste caso, é a rota para um roteador inexistente configurada a partir das mensagens *Router Advertisement* enviadas pela máquina atacante.

Figura 43: Saída do comando route -n6 antes do anúncio do roteador falso. Fonte: própria

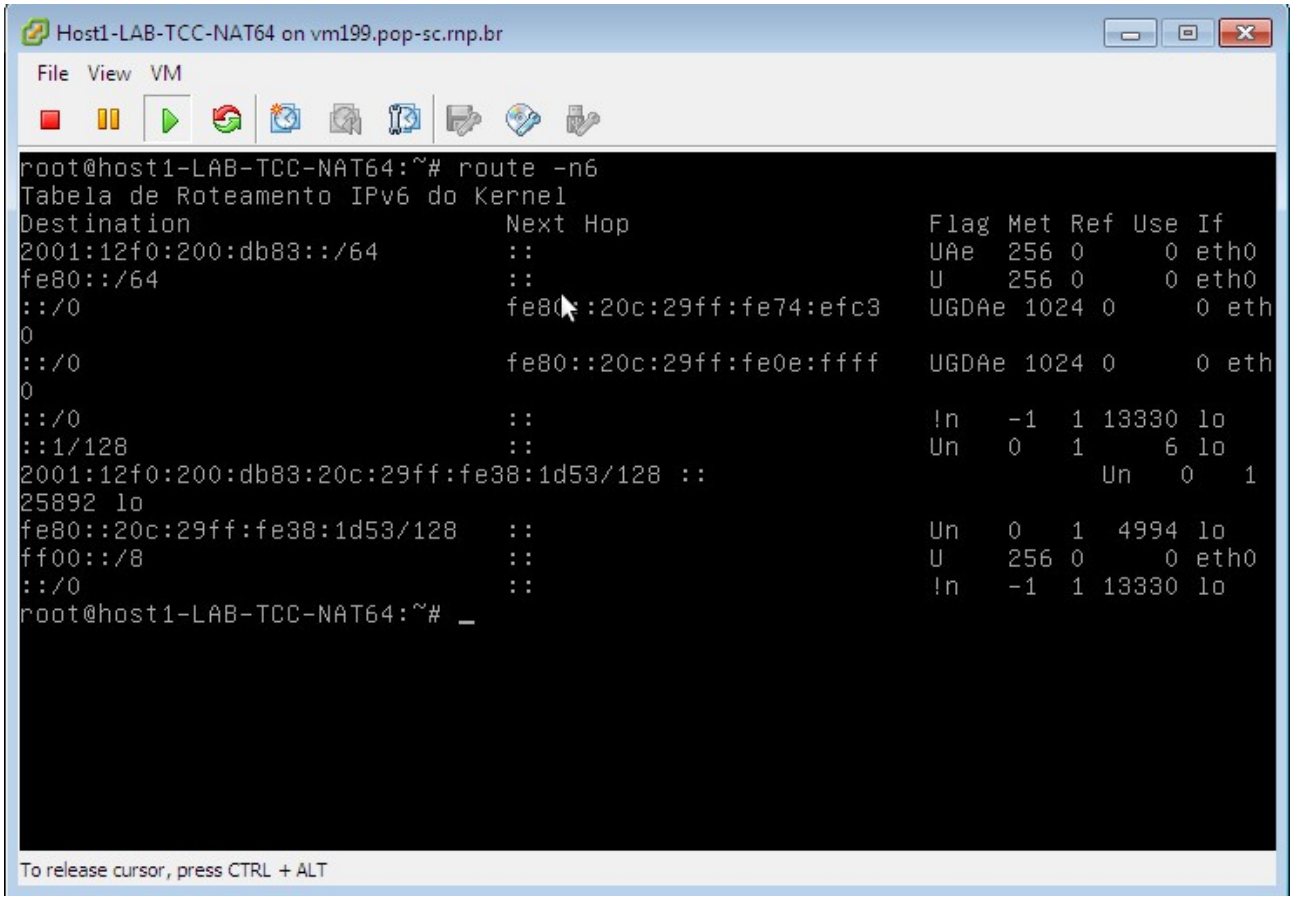

Figura 44: Saída do comando route -n6 após o anúncio do roteador falso. Fonte: própria

Após verificar a alteração nas configurações de rotas padrão dos *hosts*, foram realizados testes de conectividade com um servidor externo via ICMPv6 e com o servidor web do cenário via HTTP. O teste com o servidor externo via ICMPv6 foi executado através do comando *ping6.* Por se tratar de um cenário de tradução de endereços, foi escolhido para este teste um servidor externo puramente IPv4. A Figura 45 mostra a saída do comando *ping6* para este ataque. Como pode ser observado na figura, há uma taxa de perda de pacotes de 98%, o que praticamente impossibilita conectividade do *host* com a Internet.

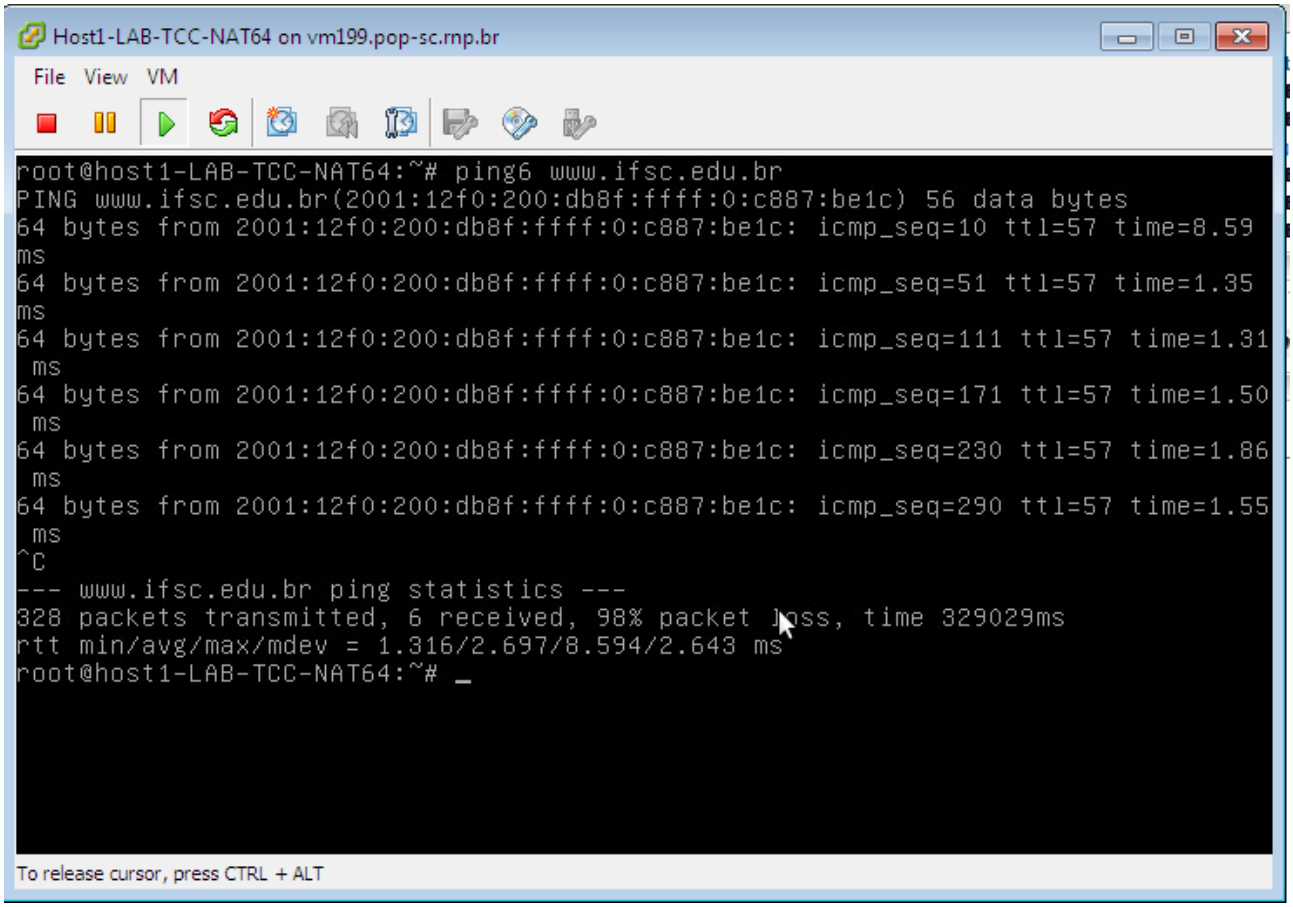

Figura 45: Teste de conectividade com um servidor externo via ICMPv6. Fonte: própria

Em um segundo *host* foi executado um teste de conectividade via HTTP com o servidor web do cenário através do comando *wget*. A saída do comando é mostrada na Figura 46. Como pode ser observado na figura, apesar do ataque, a conexão com o servidor ainda é bem sucedida.

Os resultados dos testes mostram que, no cenário NAT64, o anúncio de um roteador falso com endereço IPv6 inexistente comprometeu a conectividade com a Internet. No entanto,diferentemente do ocorrido no cenário pilha dupla, a conectividade com o servidor web do cenário não foi afetada. Apesar disto, o fato da conectividade com a Internet ter sido inviabilizada indica que foi realizado um ataque de *DoS* bem sucedido através do anúncio de um roteador falso e inexistente no cenário NAT64.

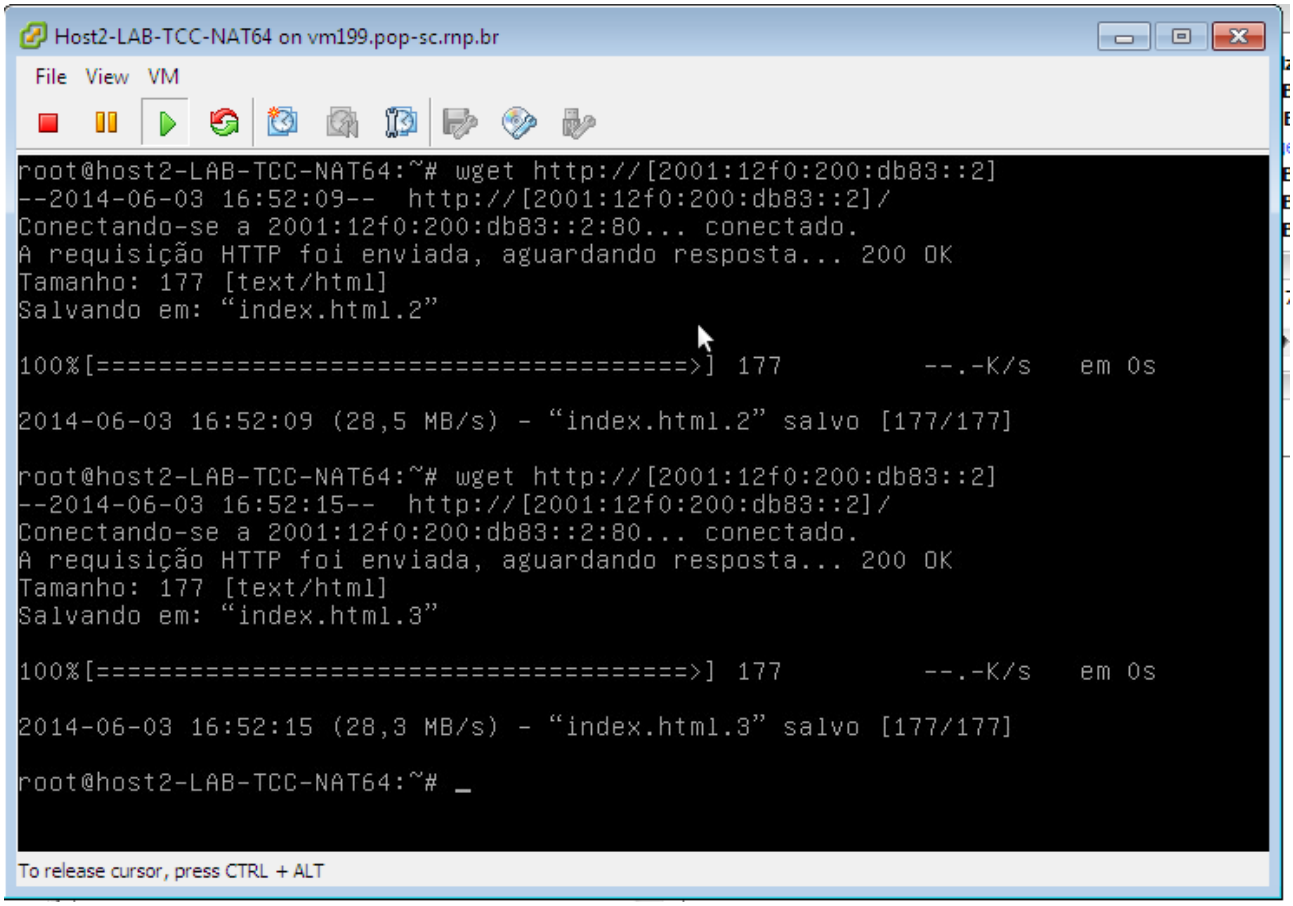

Figura 46: Teste de conectividade com o servidor web do cenário via HTTP. Fonte: própria

5.3 *Flooding* de mensagens *Neighbor Advertisement*

Neste experimento, executado nos cenário pilha dupla e NAT64, foi realizado *flooding* de mensagens *Neighbor Advertisement*, isto é, foi disparada uma grande quantidade de mensagens *Neighbor Advertisement* com o intuito de provocar um ataque de *DoS* sobrecarregando tanto a rede com o volume de tráfego gerado quanto o roteador local com o processamento de uma grande quantidade de mensagens *Neighbor Advertisement*. As seções 5.3.1 e 5.3.2 descrevem os experimentos com este ataque nos cenários pilha dupla e NAT64, respectivamente.

5.3.1 *Flooding* de mensagens *Neighbor Advertisement* no cenário pilha dupla

Neste cenário, o ataque de *flooding* de mensagens *Neighbor Advertisement* foi realizado por meio da ferramenta *flood_advertise6*, que dispara um grande número de mensagens *Neighbor*

Advertisement até que seja parado manualmente. Para este cenário, o *fake_advertise6* foi executado tendo como parâmetro a interface eth0 da máquina atacante. A Figura 47 mostra a execução do *flood_advertise6* na máquina atacante do cenário pilha dupla. Cada um dos pontos impressos na tela indicam mil mensagens *Neighbor Advertisement* disparadas pela ferramenta pela interface de rede especificada.

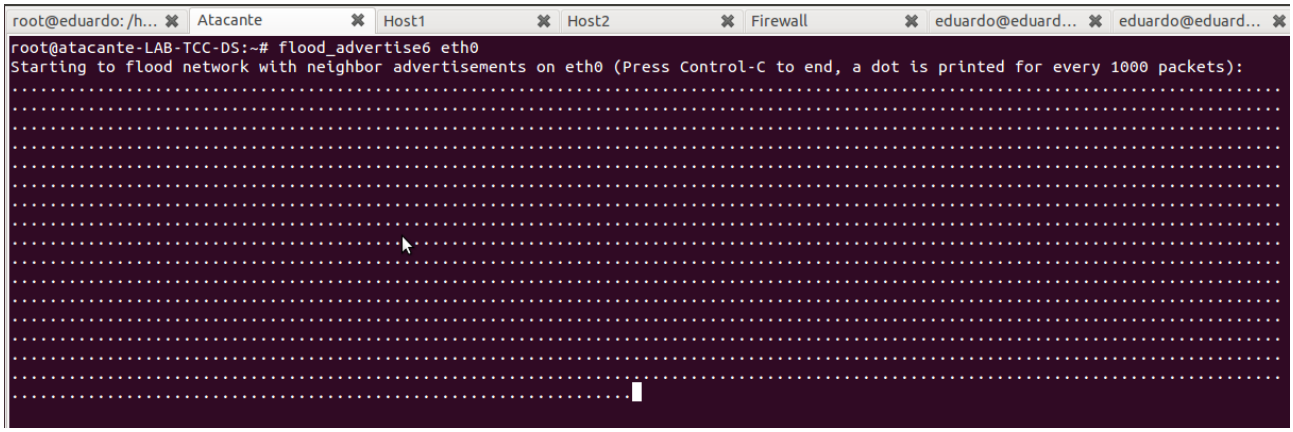

Figura 47: Execução do flood_advertise6 na máquina atacante do cenário pilha dupla. Fonte: própria

Para verificar a negação de serviço, foram executados testes de conectividade com os *hosts* e com o servidor web do cenário. O teste de conectividade com os *hosts* foi realizado via ICMPv6 através do comando *ping6*. Na figura 48 pode ser observada a saída do comando *ping6* para um dos *hosts* do cenário. Observa-se que o *host* respondiam às mensagens *echo-request* enviadas através do comando *ping6.* Porém, após o início da execução do *flood_advertise6*, era recebida uma mensagem de erro indicando que o *host* estava inalcançável.

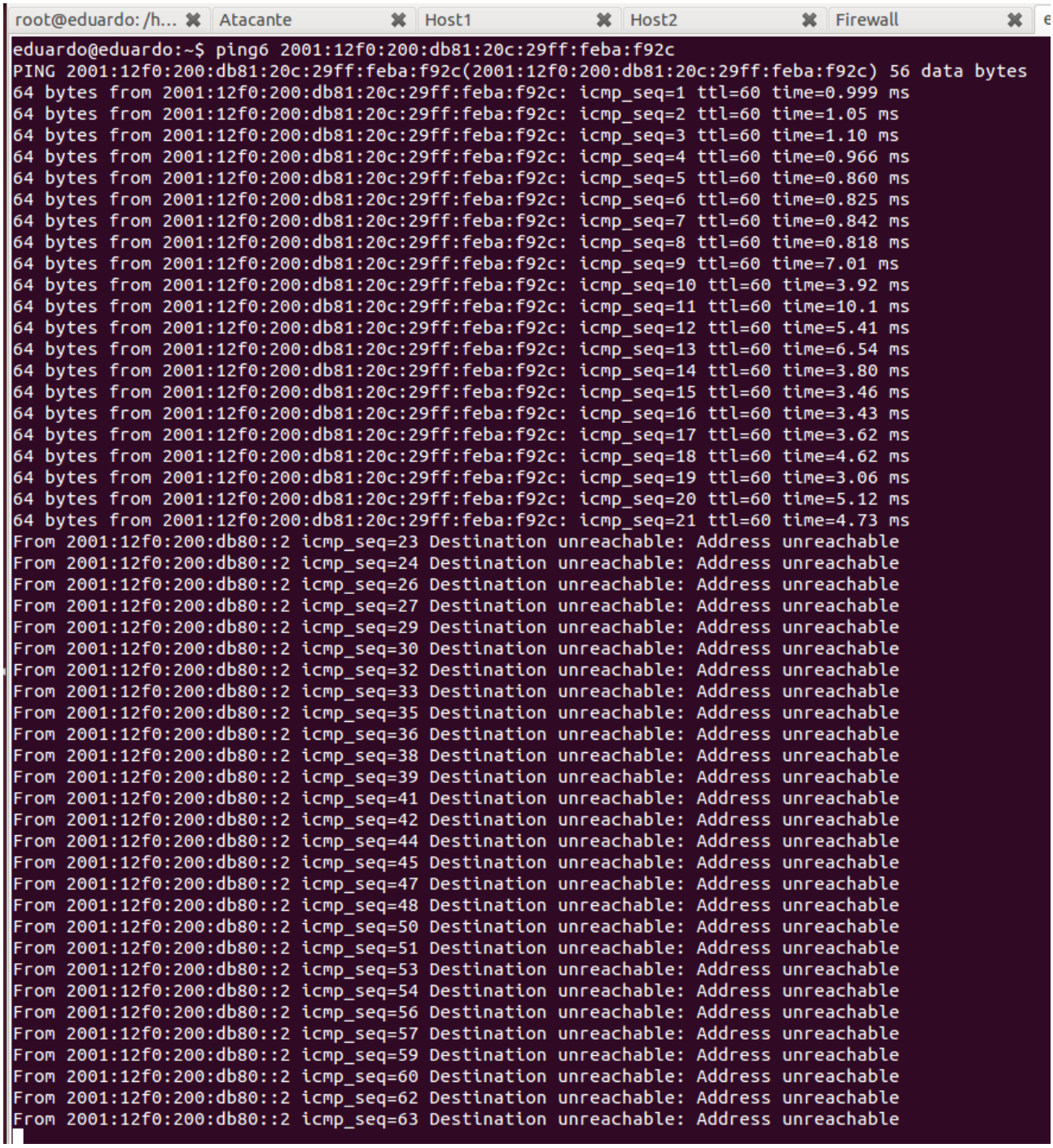

Figura 48: Teste de conectividade com um host do cenário através do comando ping6. Fonte: própria

No servidor web, foi testada a conectividade via HTTP utilizando-se um navegador web. A Figura 49 mostra o acesso ao servidor antes do ataque. Pode-se observar na figura que era possível a conectividade com o servidor. Já na Figura 50, é mostrada a tentativa de acesso ao servidor após o início do ataque. Pode-se observar que o navegador web apresenta uma mensagem de erro informando que a conexão com o servidor não pode ser estabelecida.

Ao analisar os resultados dos testes de conectividade realizados, verifica-se que o grande volume de mensagens *Neighbor Advertisement* disparadas na rede local afeta todos os *hosts* nela presentes e até mesmo o servidor web. Através da sobrecarga da rede com mensagens *Neighbor Advertisement*, que, por sua vez, sobrecarregam o roteador local com o processamento destas mensagens, realizou-se um bem sucedido ataque de *DoS* utilizando as mensagens *Neighbor Advertisement*, previstas apenas a partir do IPv6.

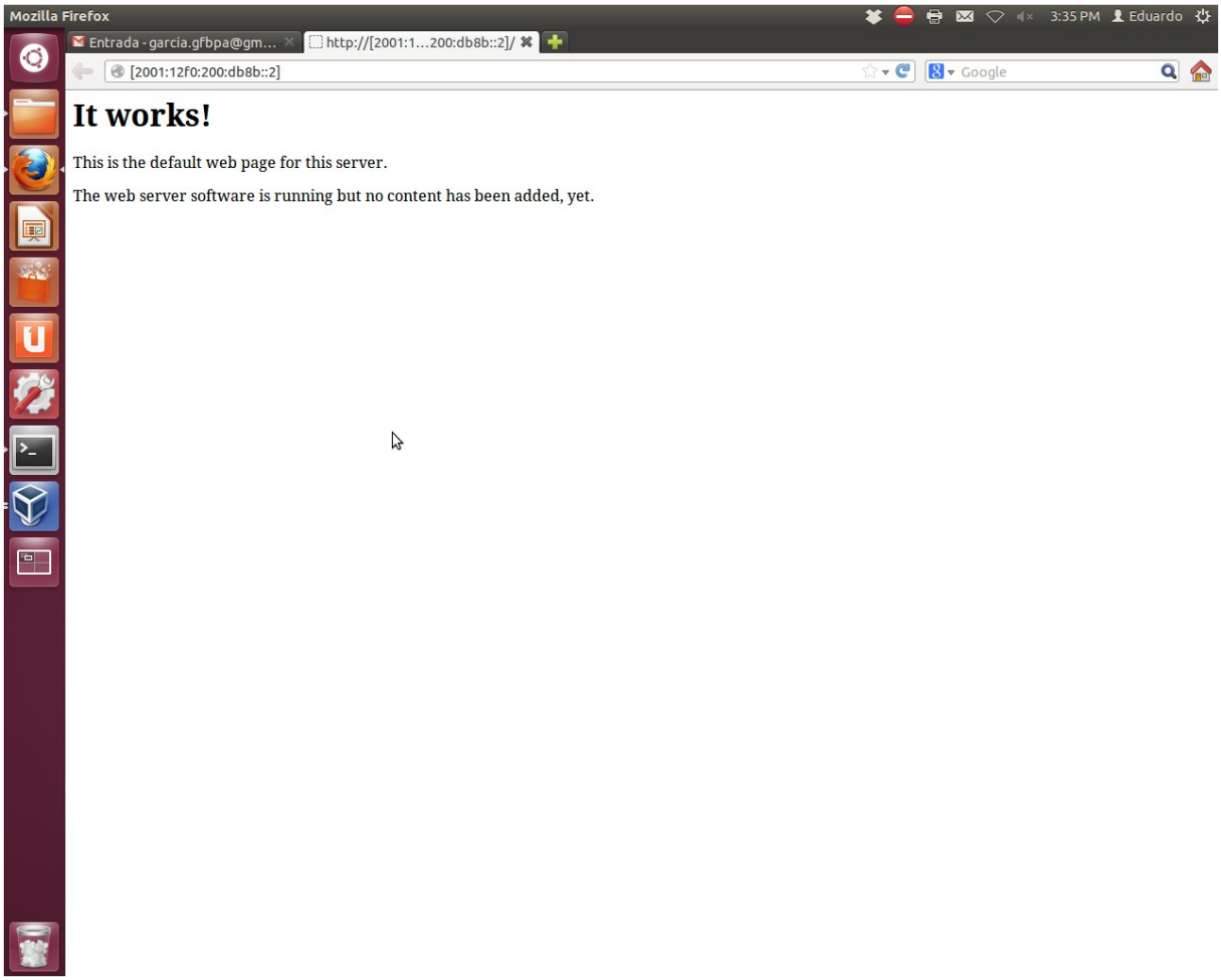

Figura 49: Teste de conectividade com o servidor web via navegador antes do início do ataque. Fonte: própria

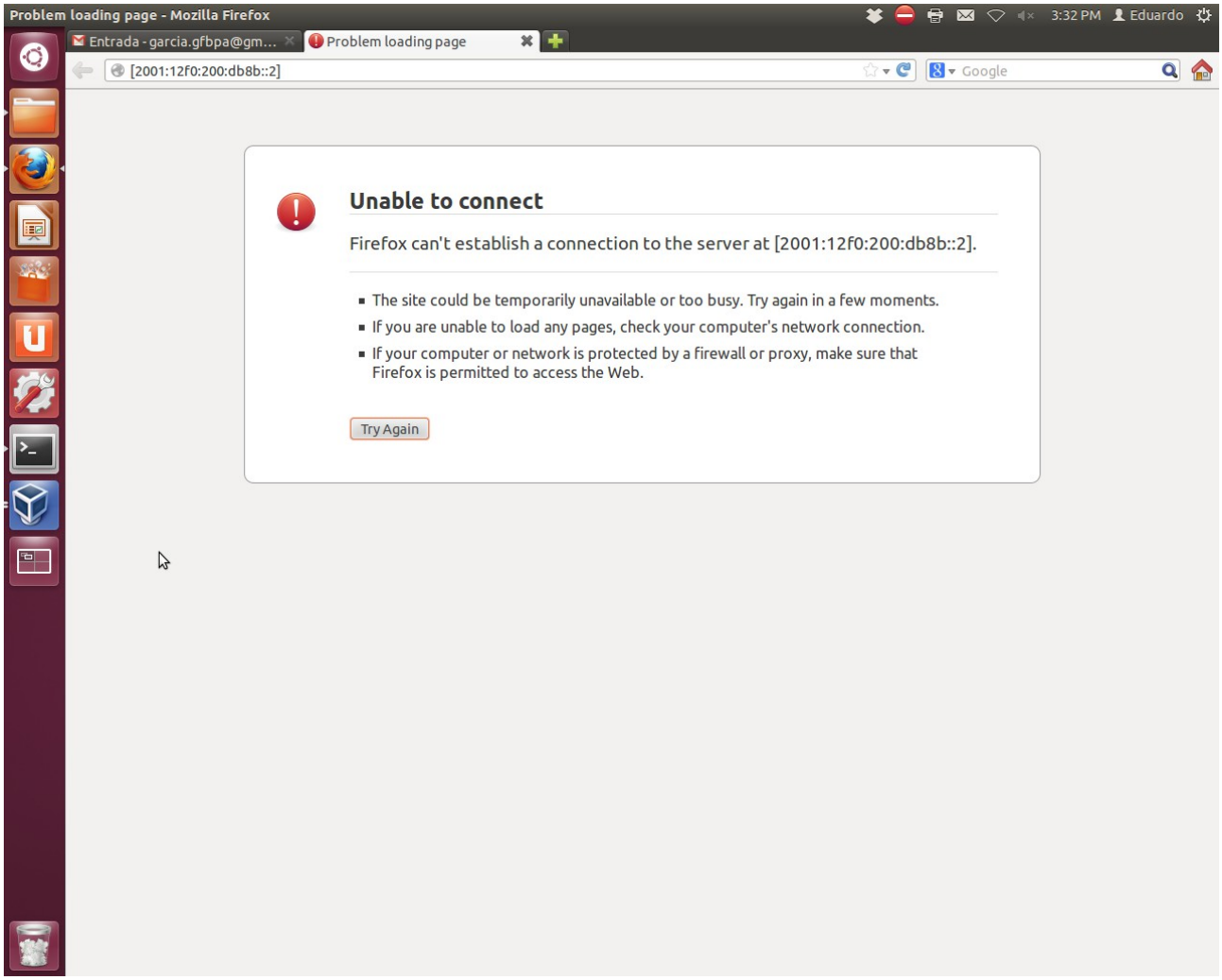

Figura 50: Teste de conectividade com o servidor web via navegador após o início do ataque. Fonte: própria

5.3.2 *Flooding* de mensagens *Neighbor Advertisement* no cenário NAT64

Seguindo a mesma metodologia do cenário pilha dupla, o ataque de *flooding* de mensagens *Neighbor Advertisement* foi realizado no cenário NAT64. Neste cenário, a ferramenta *flood_advertise6* também foi executada recebendo como parâmetro a interface eth0 da máquina atacante, disparando por esta interface um grande volume de mensagens *Neighbor Advertisement* na rede local do cenário. A execução do *flood_advertise6* é mostrada na Figura 51.

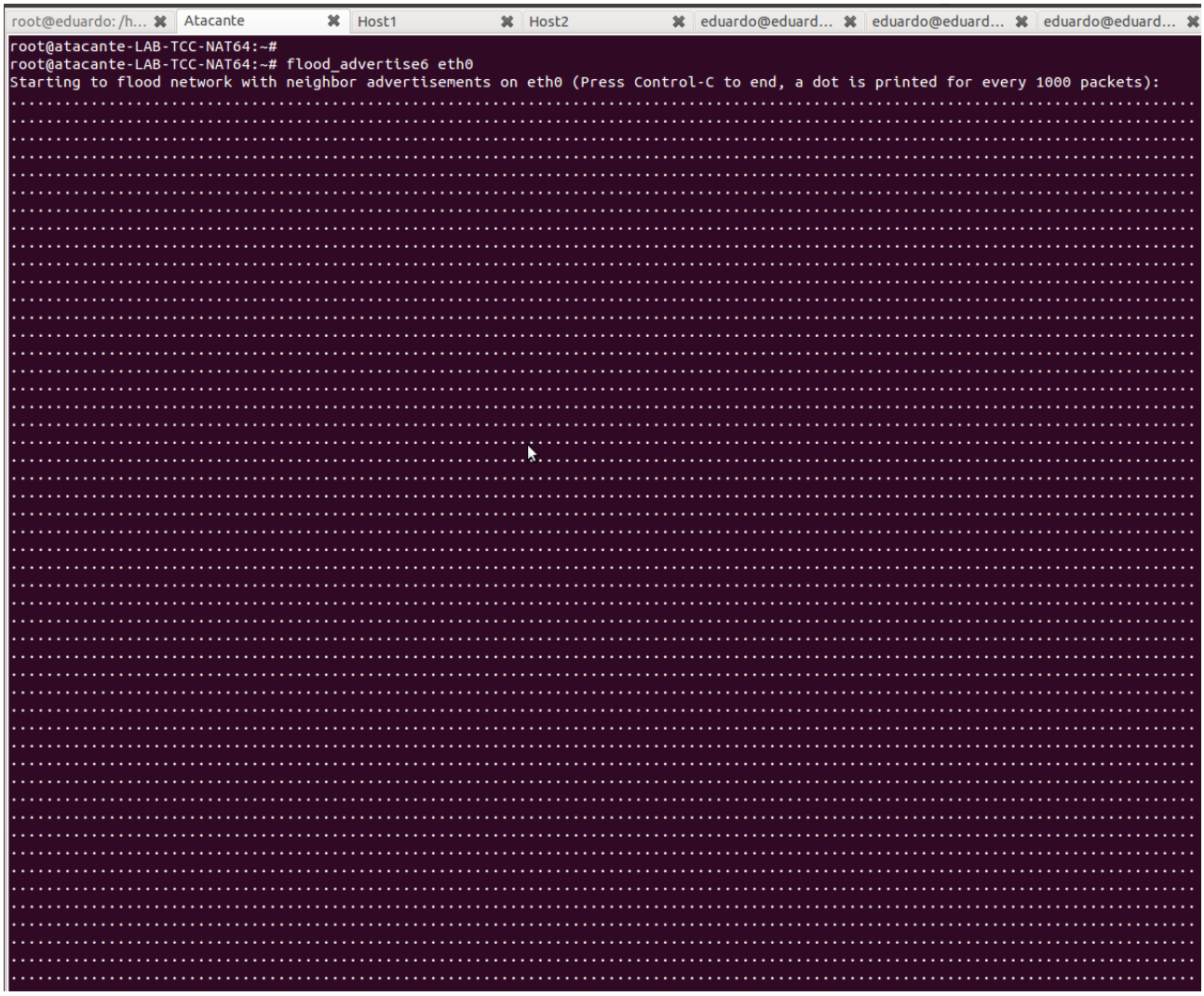

Figura 51: Execução do flood_advertise6 na máquina atacante do cenário NAT64. Fonte: própria

Assim como no cenário pilha dupla, foram realizados testes de conectividade nos *hosts* e no servidor web. Nos *hosts*, os testes de conectividade foram realizados via ICMPv6 através do comando *ping6*. A saída do comando *ping6* para um dos *hosts* é mostrada na Figura 52. Pode-se observar que, inicialmente, o *hosts* respondia às mensagens *echo-request* enviadas através do comando *ping6*. Após o início do ataque, o *host* para de responder a estas mensagens e é retornada uma mensagem de erro indicando que o *host* está inacessível.

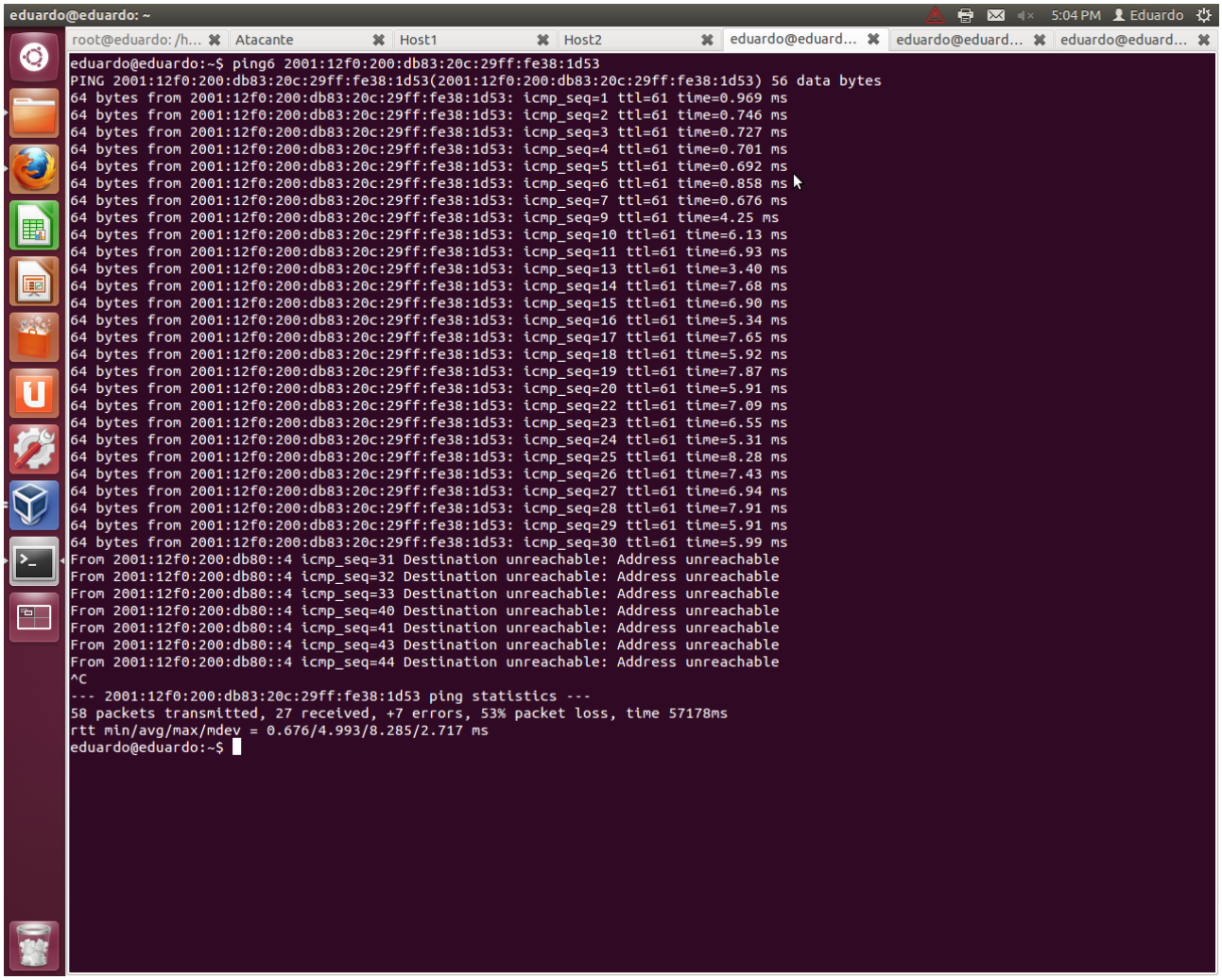

Figura 52: Teste de conectividade com um host do cenário através do comando ping6. Fonte: própria

Assim como no cenário pilha dupla, os testes de conectividade com o servidor web via HTTP foram realizados através de um navegador web. Na Figura 53 é mostrada uma tentativa de conexão com o servidor antes da realização do ataque. Pode-se observar que a tentativa de conexão é bem sucedida e o servidor mostra a página web corretamente. Já na Figura 54, que mostra uma tentativa de conexão com servidor após o início do ataque, pode-se observar que a conexão não é bem sucedida e o servidor apresenta uma mensagem de erro informando que não é possível estabelecer conexão com o servidor.

Após os testes de conectividade, pode-se observar que, assim como no cenário pilha dupla, a sobrecarga da rede local do cenário com mensagens *Neighbor Advertisement*, que, por consequência, sobrecarrega o roteador local com o processamento destas mensagens, provoca um bem sucedido ataque de *DoS* na rede local do cenário.

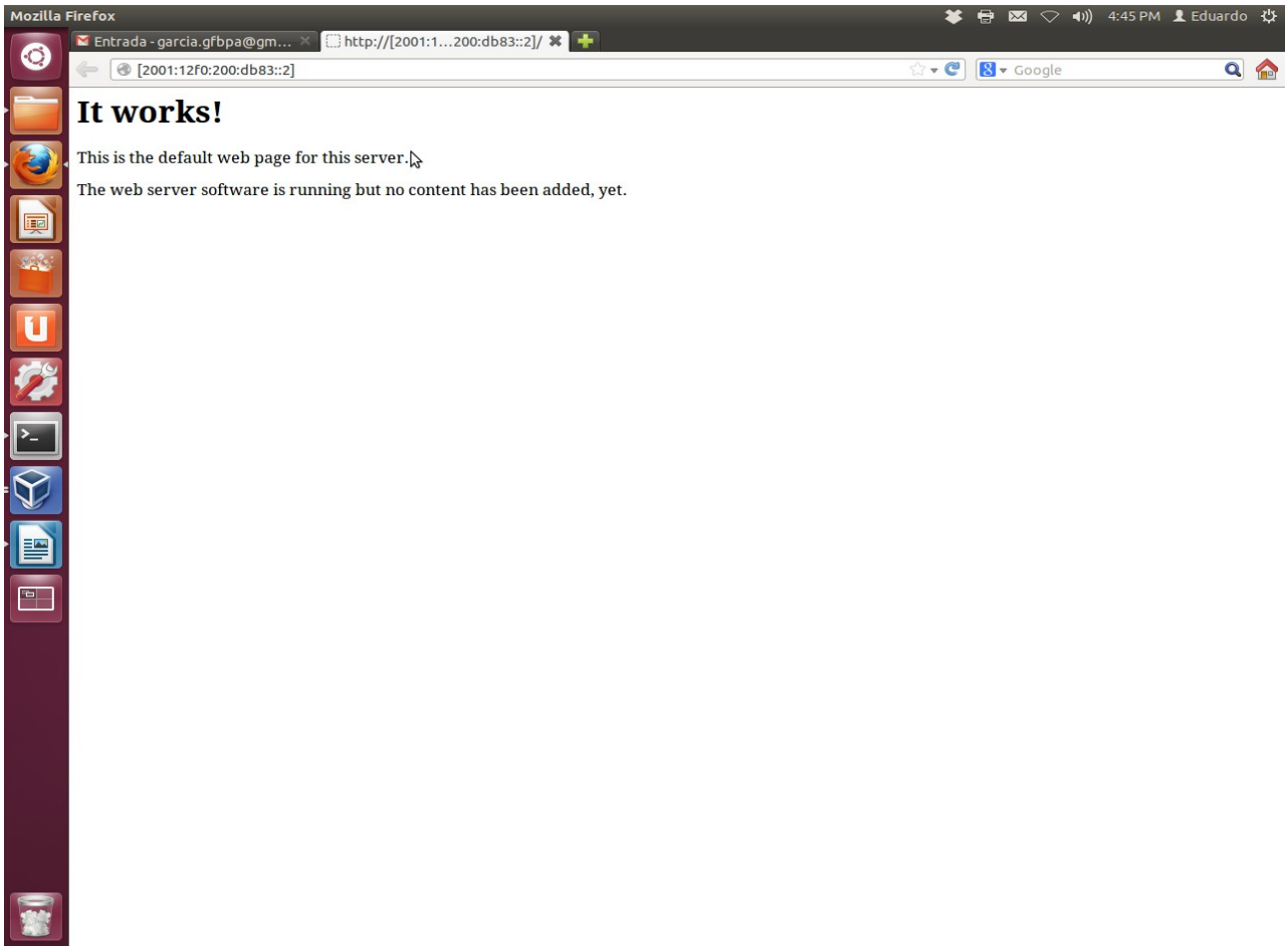

Figura 53: Teste de conectividade com o servidor web do cenário via HTTP antes do início do ataque. Fonte: própria

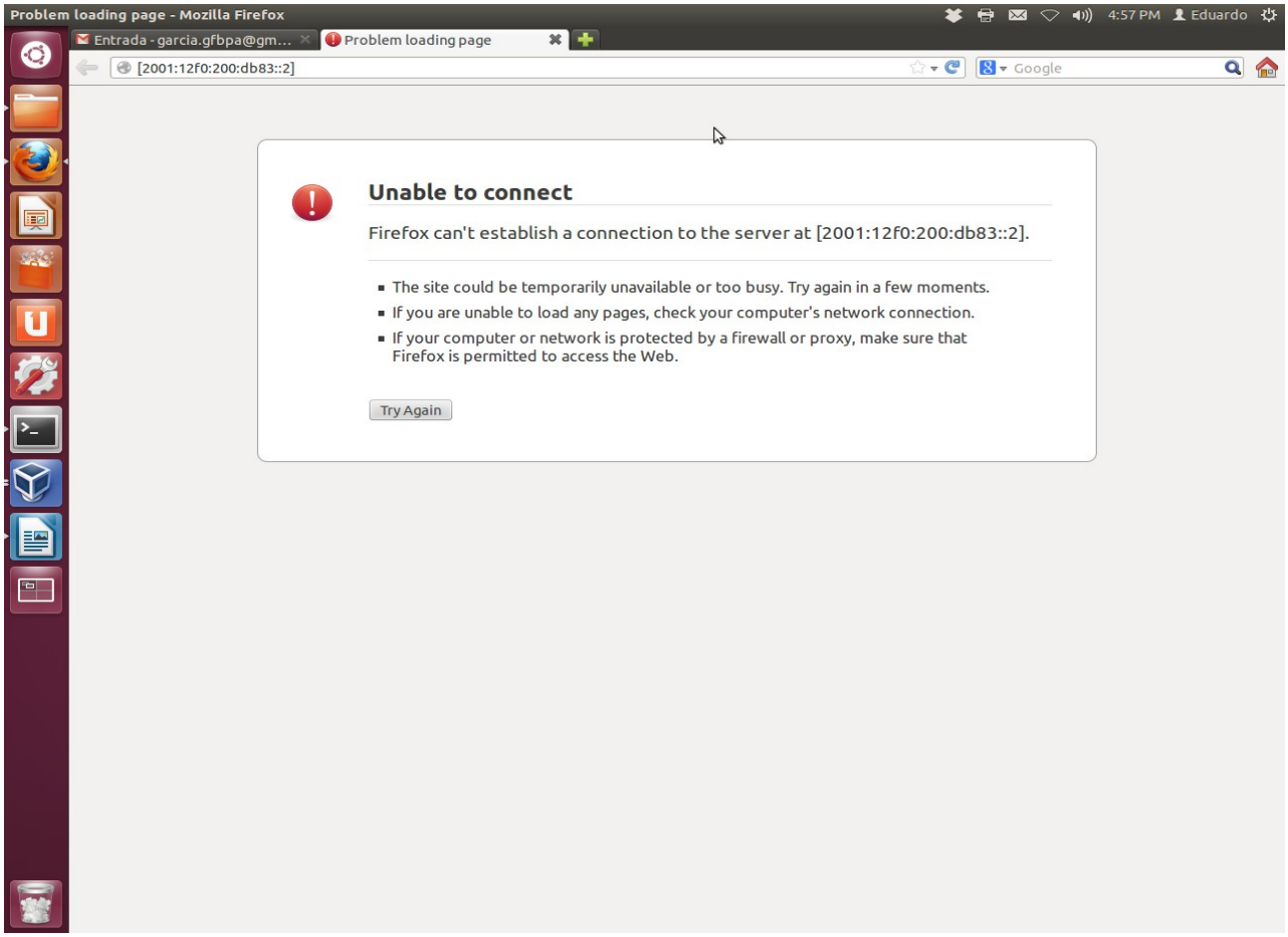

Figura 54: Teste de conectividade com o servidor web do cenário via HTTP após o início do ataque. Fonte: própria

5.4 *Flooding* de mensagens *Router Advertisement*

Além do ataque de *flooding* com mensagens *Neighbor Advertisement*, foi realizado, nos cenários pilha dupla e NAT64, um ataque de *flooding* com mensagens *Router Advertisement*. Este ataque, caso seja bem sucedido, não apenas sobrecarrega a rede local com o grande volume de tráfego como também faz com que os *hosts* que recebem as mensagens *Router Advertisement* disparadas na rede configurem em suas interfaces de rede endereços IPv6 inválidos, assim como faz com que os *hosts* recebam informações de roteamento inválidas. As seções 5.4.1 e 5.4.2 descrevem a realização deste ataque nos cenários pilha dupla e NAT64, respectivamente.

5.4.1 *Flooding* de mensagens *Router Advertisement* no cenário pilha dupla

Neste cenário, o ataque foi realizado através da execução da ferramenta *flood_router26* recebendo como parâmetro a interface eth0 da máquina atacante. Ao ser executado o *flood_router26,* a máquina atacante passou a disparar uma grande quantidade de mensagens *Router Advertisement* contendo prefixos IPv6 aleatórios. A Figura 55 mostra a execução do *flood_router26* na máquina atacante do cenário. Cada ponto impresso na tela indica mil mensagens *Router Advertisement* enviadas.

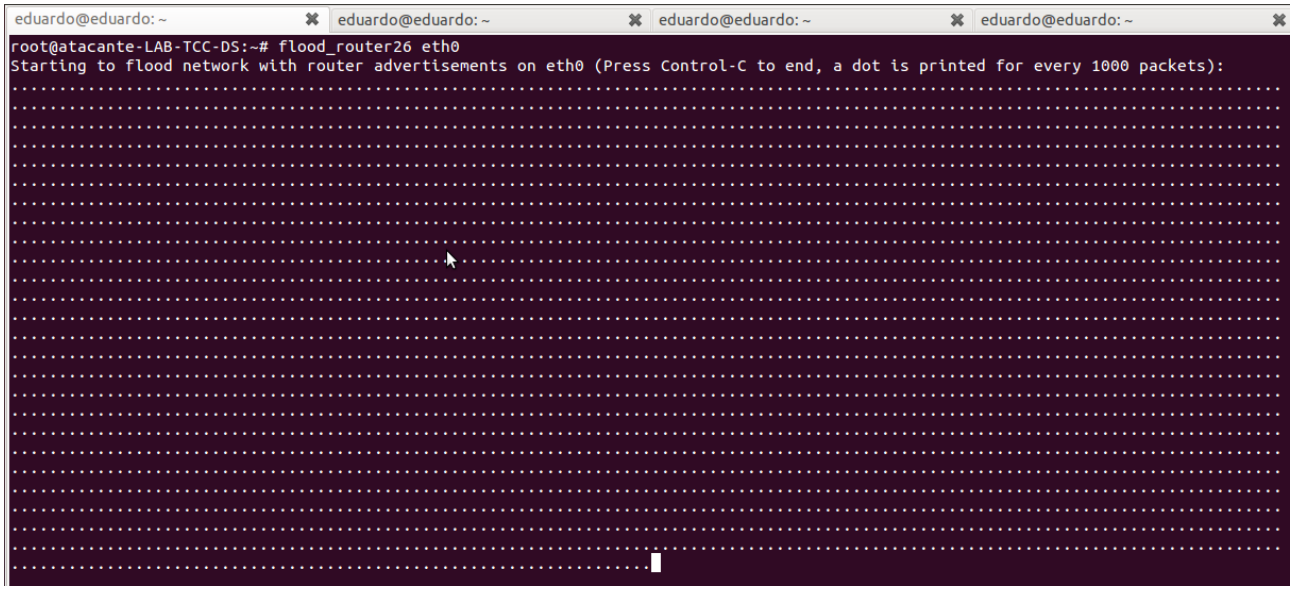

Figura 55: Execução do flood_router no cenário pilha dupla. Fonte: própria

Após o início do ataque, pode-se observar no em cada um dos *hosts*, diretamente no console do VMware, diversas mensagens de erro informando que não foi possível definir rota padrão. A impossibilidade de configurar rota, neste caso, se deve ao grande volume de mensagens *Router Advertisement* recebidas pelos *hosts* que, combinadas, contém um volume de informações de prefixo e roteamento maior do que os *hosts* são capazes de suportar. A Figura 56 mostra estas mensagens para um dos *hosts*.

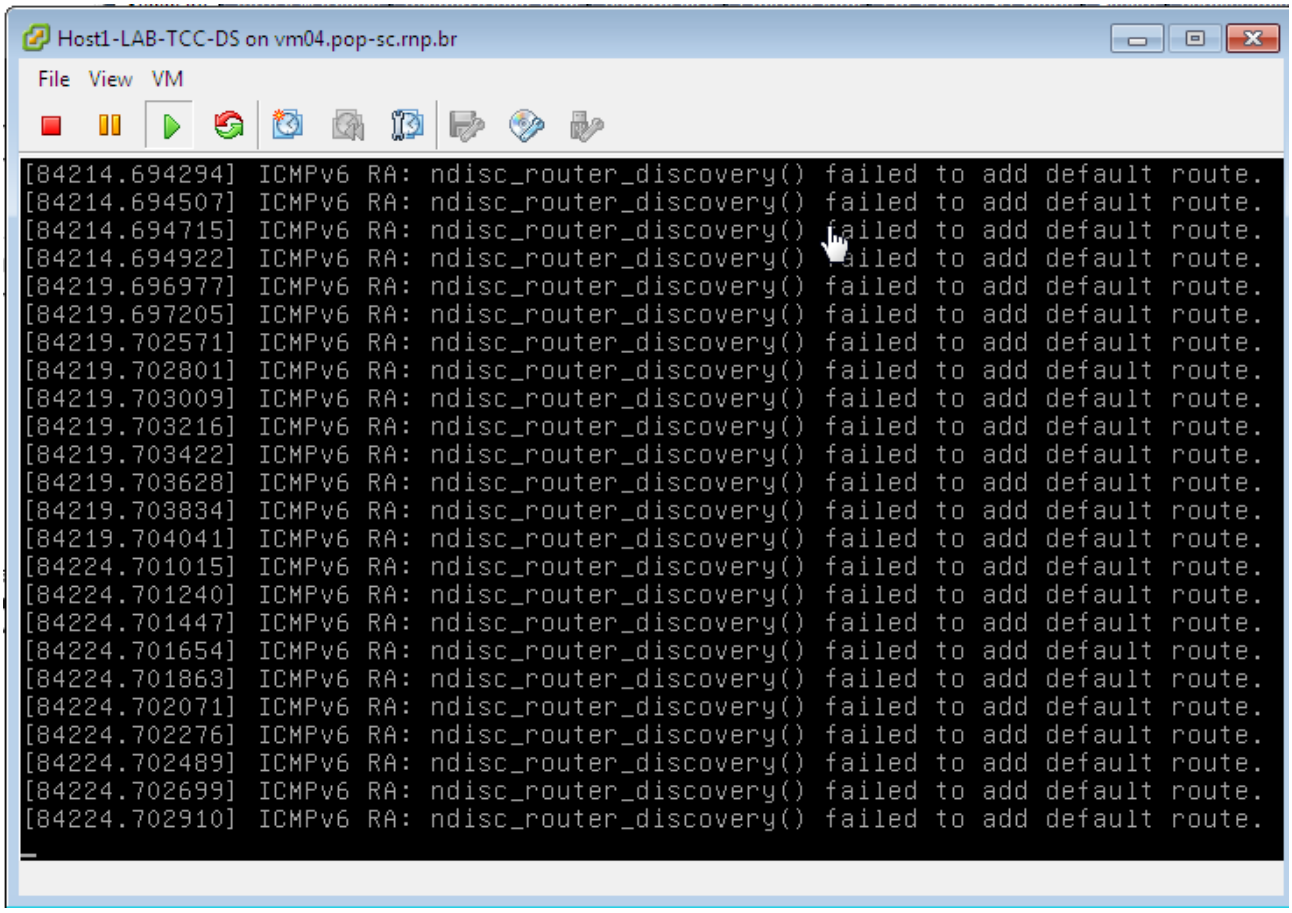

Figura 56: Mensagens de erro ao tentar configurar rota padrão em um dos hosts. Fonte: própria

As mensagens *Router Advertisement* carregam informações de prefixo e roteamento utilizadas na autoconfiguração de endereços. Como consequência deste ataque, os *hosts* do cenário deveriam configurar diversos endereços IPv6 em suas interfaces de rede conectadas à rede local do cenário. Na Figura 57, é mostrada a configuração da interface eth0 de um dos *hosts* do cenário antes do ataque, com apenas um endereço IPv6 global configurado. Já na Figura 58, é mostrada a configuração da mesma interface deste *host* após o ataque. Pode-se verificar que há diversos endereços IPv6 configurados e que estes outros endereços possuem um prefixo diferente do prefixo de rede do cenário. Verifica-se, portanto, que estes endereços foram configurados a partir das mensagens *Router Advertisement* ilegítimas enviadas pela máquina atacante.

Figura 57: Configuração da interface de rede de um dos hosts antes do início do ataque. Fonte: própria

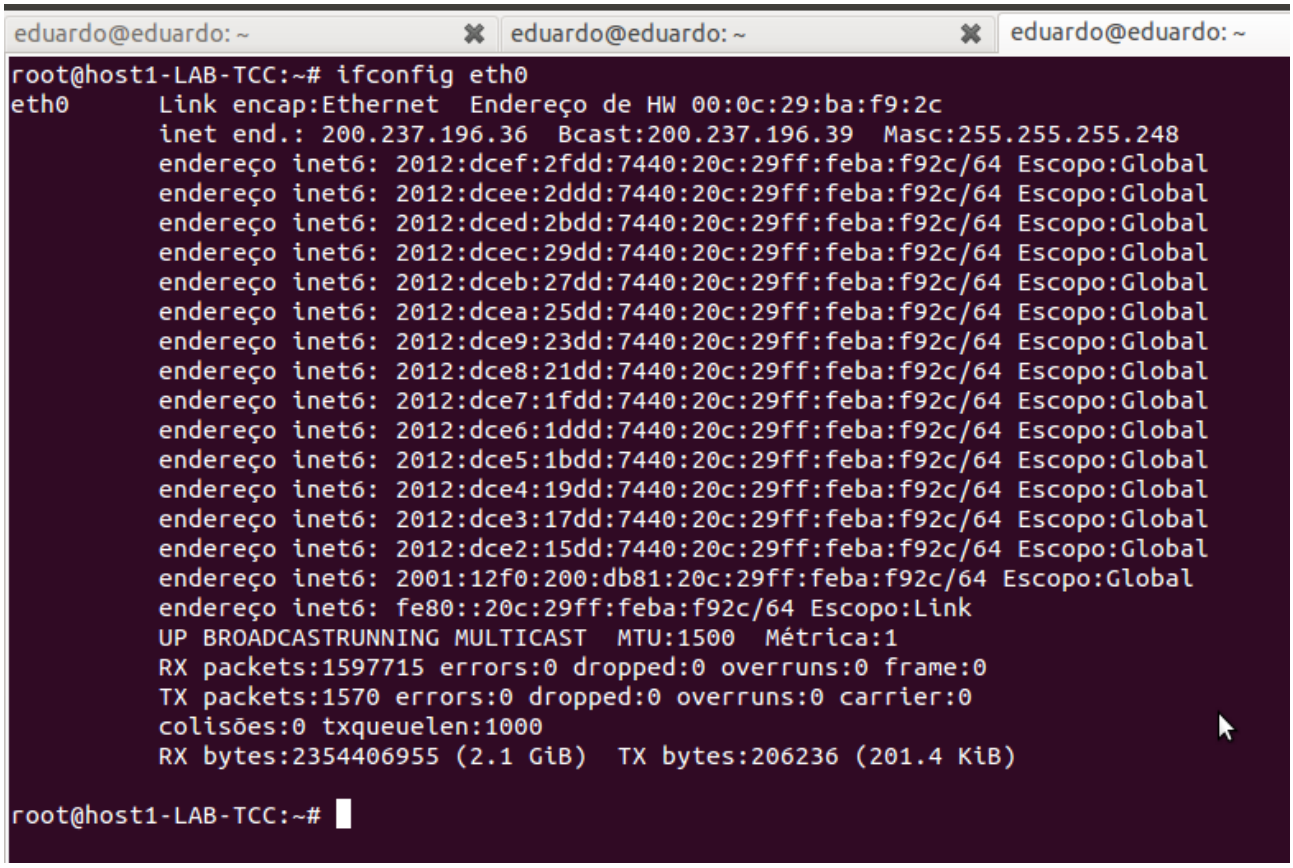

Figura 58: Configuração da interface de rede de um dos hosts após o início do ataque. Fonte: própria

Após o início do ataque, foram realizados testes de conectividade nos *hosts* e no servidor web. Nos *hosts*, o teste de conectividade foi realizado via ICMPv6 através do comando *ping6*. A Figura 59 mostra a saída do comando *ping6* para um dos *hosts*. É possível verificar que as mensagens

echo-request não recebiam resposta, mas sim uma mensagem de erro indicando que o *host* estava inalcançável.

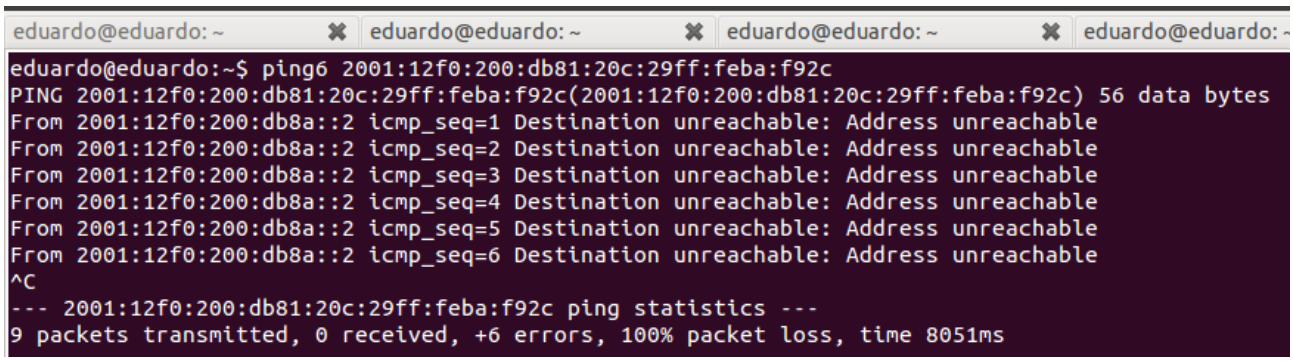

Figura 59: Teste de conectividade com um servidor externo via ICMPv6. Fonte: própria

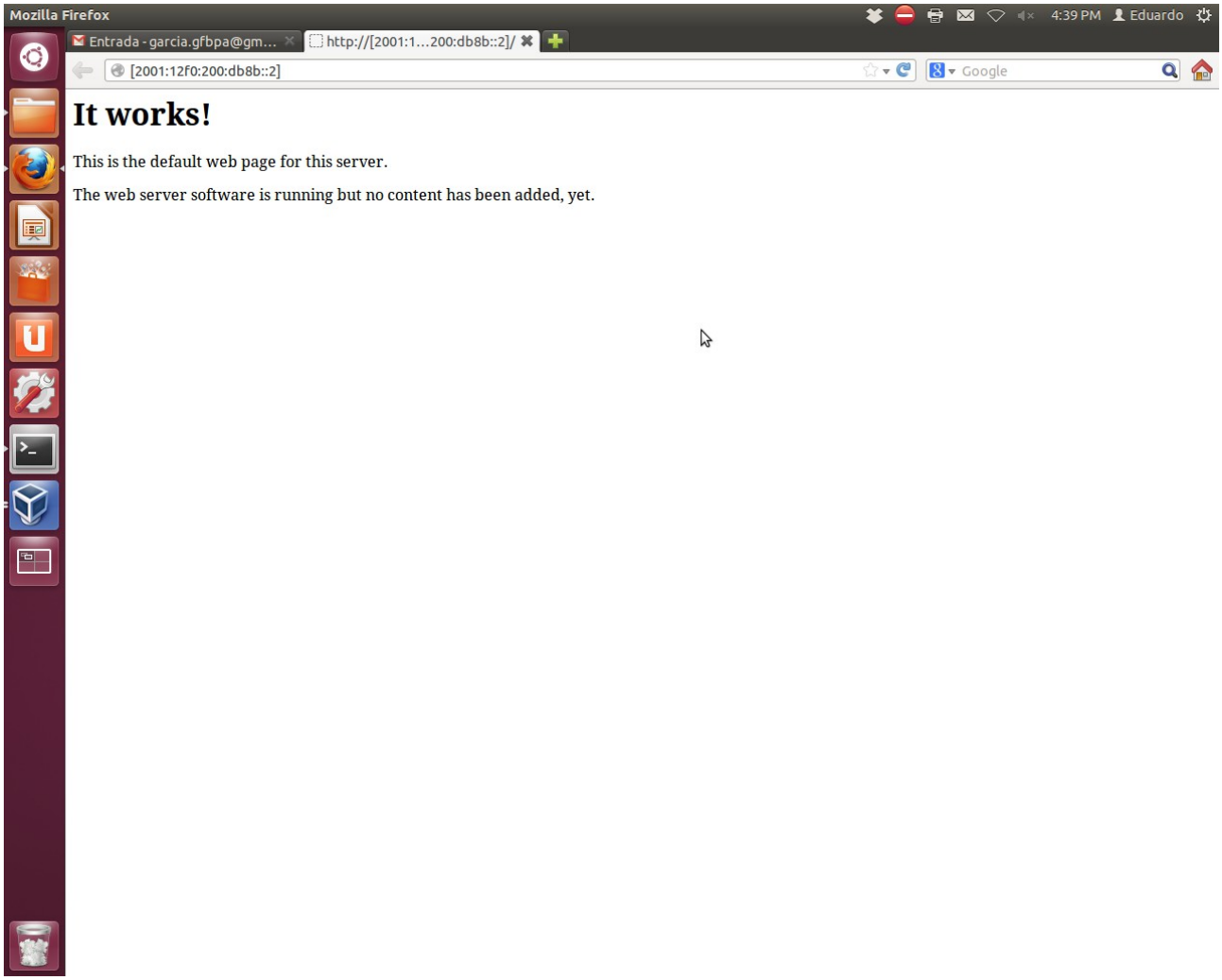

Figura 60: Teste de conectividade com o servidor do cenário via HTTP antes do início do ataque. Fonte: própria

No servidor web, os testes foram realizados via HTTP através de um navegador web. A Figura

60 mostra a conexão com o servidor via navegador web antes do ataque, exibindo a página web corretamente. Já na Figura 61, podemos verificar que a conexão não é estabelecida e o navegador web mostra uma mensagem de erro.

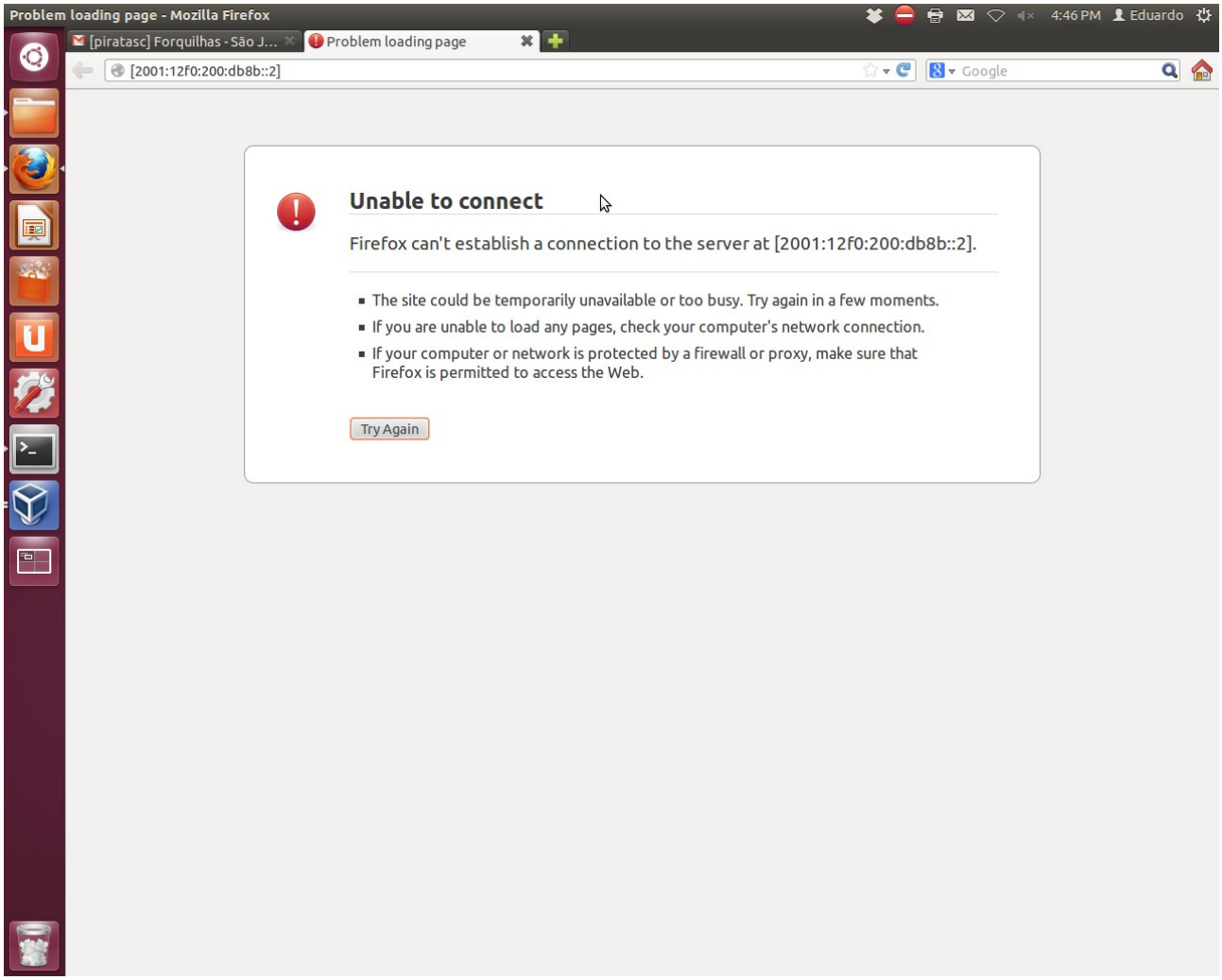

Figura 61: Teste de conectividade com o servidor do cenário via HTTP após o início do ataque. Fonte: própria

Verifica-se que as mensagens *Router Advertisement* enviadas pela máquina atacante, além de sobrecarregarem a rede local em função do grande volume, impedem a conectividade com os *hosts* em função dos endereços IPv6 e informações de roteamento inválidas. Obteve-se, portanto, mais um ataque de *DoS* bem sucedido explorando o mecanismo de autoconfiguração de endereços IPv6.

5.4.2 *Flooding* de mensagens *Router Advertisement* no cenário NAT64

Seguindo a metodologia do cenário pilha dupla, foi executado o ataque de *flooding* de mensagens *Router Advertisement* no cenário NAT64. Assim como no cenário pilha dupla, a ferramenta *flood_router26* foi executada recebendo como parâmetro a interface eth0 da máquina atacante. A Figura 62 mostra a execução do *flood_router* neste cenário. Cada ponto impresso na tela representa mil mensagens *Router Advertisement* enviadas.

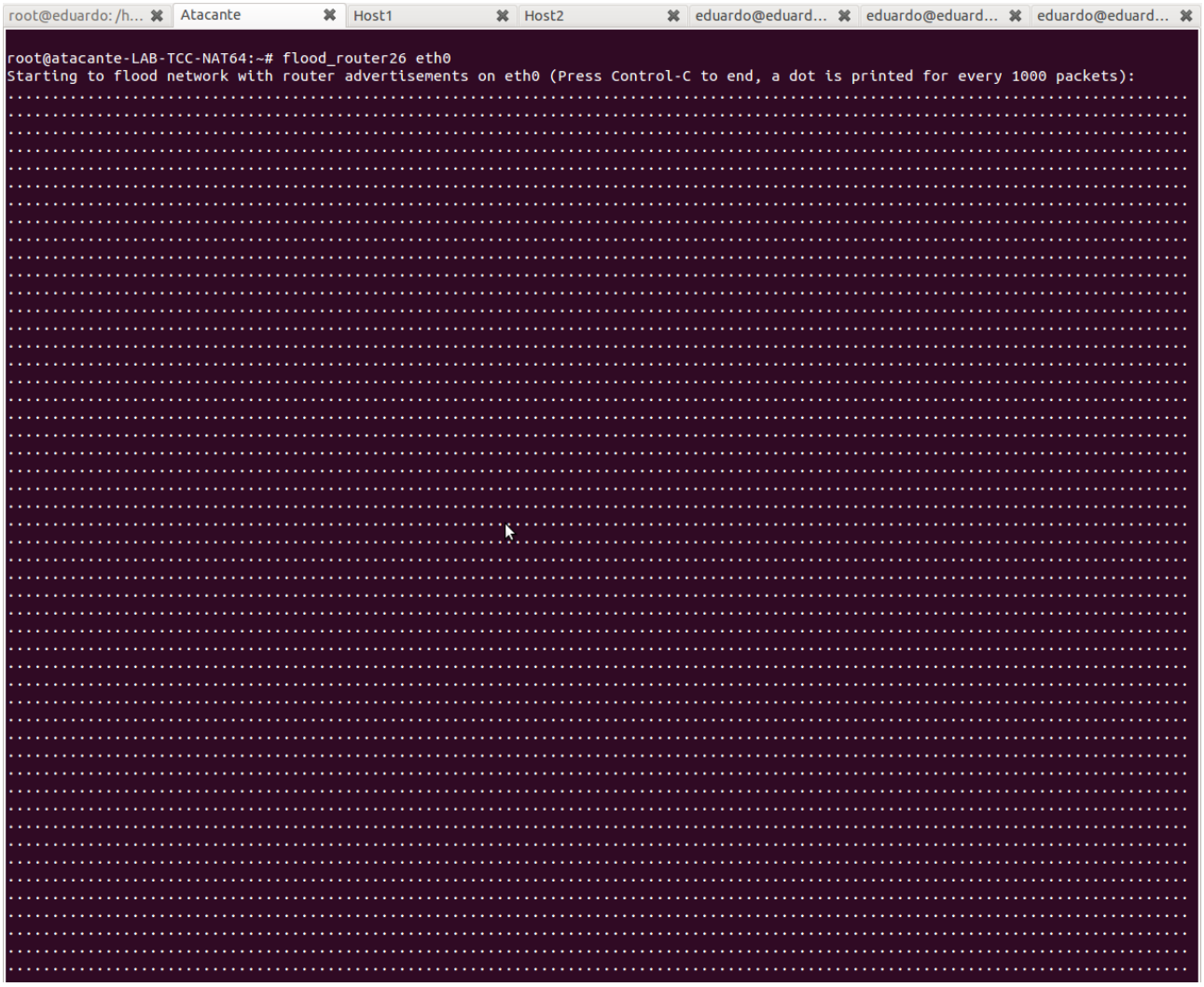

Figura 62: Execução do flood_router26 no cenário NAT64. Fonte: própria

Após o início do ataque, pode-se verificar nos *hosts*, observando-se diretamente o console do VMware, as mensagens de erro informando que não é possível definir rota padrão. Assim como no cenário pilha dupla, o grande volume de mensagens *Router Advertisement* carrega um volume de informações de prefixo e roteamento maior que o suportado pelos *hosts* e a autoconfiguração de

endereços falha em muitos casos. A Figura 63 mostra estas mensagens de falha de configuração para um dos *hosts*.

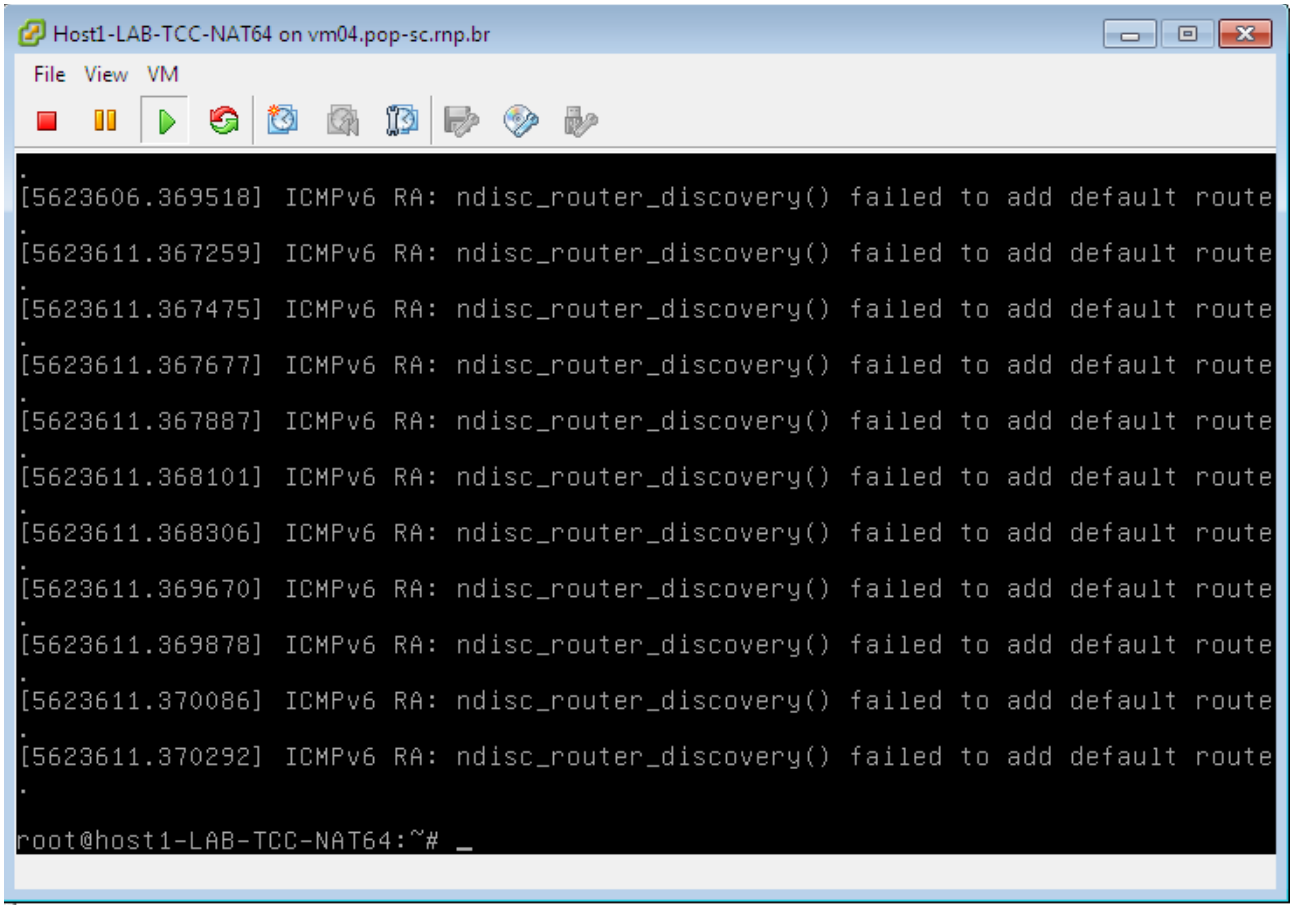

Figura 63: Mensagens de erro ao tentar configurar rota padrão em um dos hosts. Fonte: própria

Uma vez constatado que o grande volume de mensagens *Router Advertisement* já surte efeito nos *hosts* da rede local do cenário, é esperado que os mesmo configurem em suas interfaces de rede conectadas à rede local um grande número de endereços IPv6. A Figura 64 mostra a configuração da interface eth0 de um dos *hosts* do cenário antes do ataque, contendo apenas um endereço IPv6 global configurado. Já na Figura 65, que mostra a configuração da mesma interface de rede do mesmo *host* após o ataque, pode-se observar que há vários endereços IPv6 globais configurados, sendo que estes outros endereços possuem um prefixo de rede diferente do prefixo da rede local do cenário.

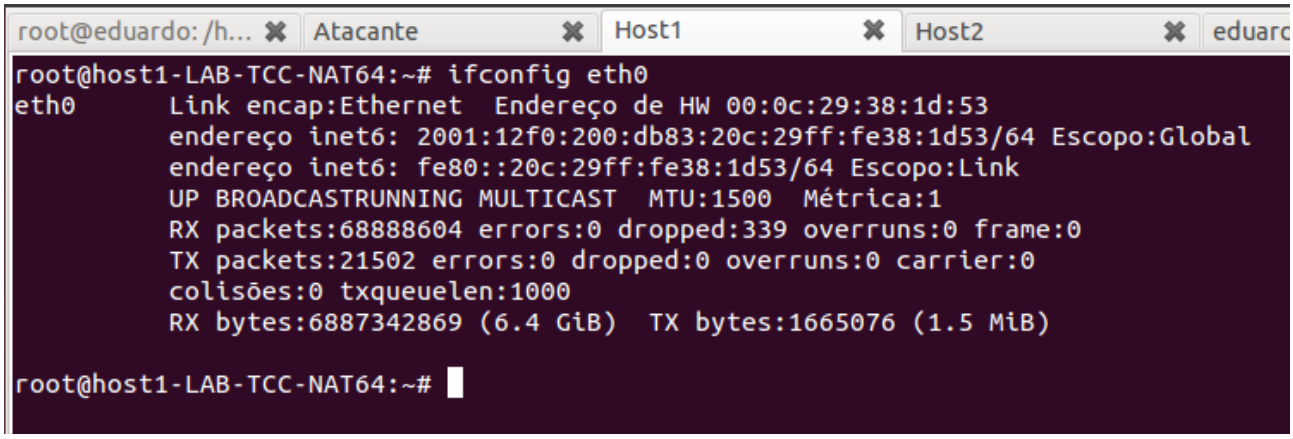

Figura 64: Configuração da interface de rede de um dos hosts antes do início do ataque. Fonte: própria

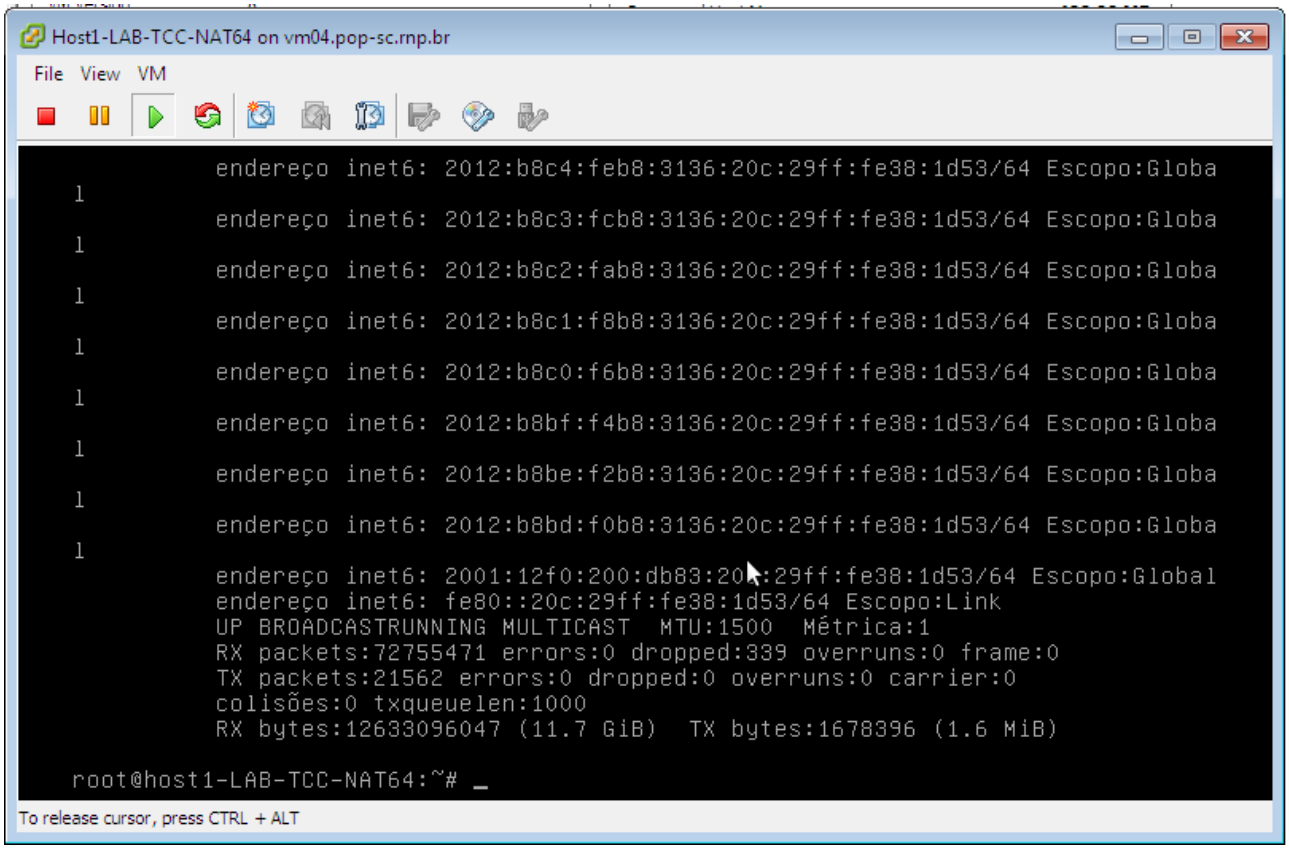

Figura 65: Configuração da interface de rede de um dos hosts após o início do ataque. Fonte: própria

Após o início do ataque, foram realizados testes de conectividade nos *hosts* e no servidor web. Nos *hosts*, o teste de conectividade foi realizado via ICMPv6 através do comando *ping6*. A Figura 66 mostra a saída do comando *ping6* para um dos *hosts*. Assim como no cenário pilha dupla, é possível verificar que as mensagens *echo-request* não recebiam resposta, mas sim uma mensagem de erro indicando que o *host* estava inalcançável.

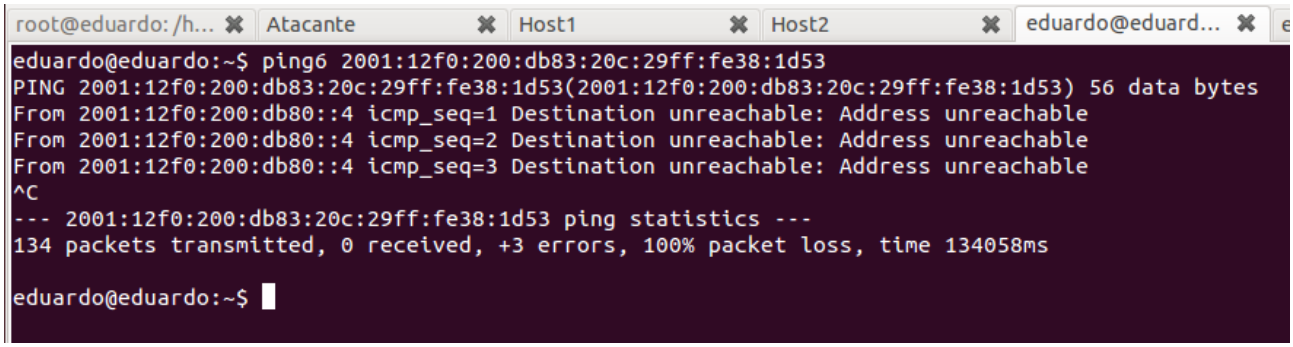

Figura 66: Teste de conectividade com um servidor externo via ICMPv6. Fonte própria

No servidor web, assim como no cenário pilha dupla, os testes foram realizados via HTTP através de um navegador web. A Figura 67 mostra a conexão com o servidor via navegador web antes do ataque, exibindo a página web corretamente. Já na Figura 68, podemos verificar que a conexão não é estabelecida pois o tempo limite para o estabelecimento da mesma é excedido.

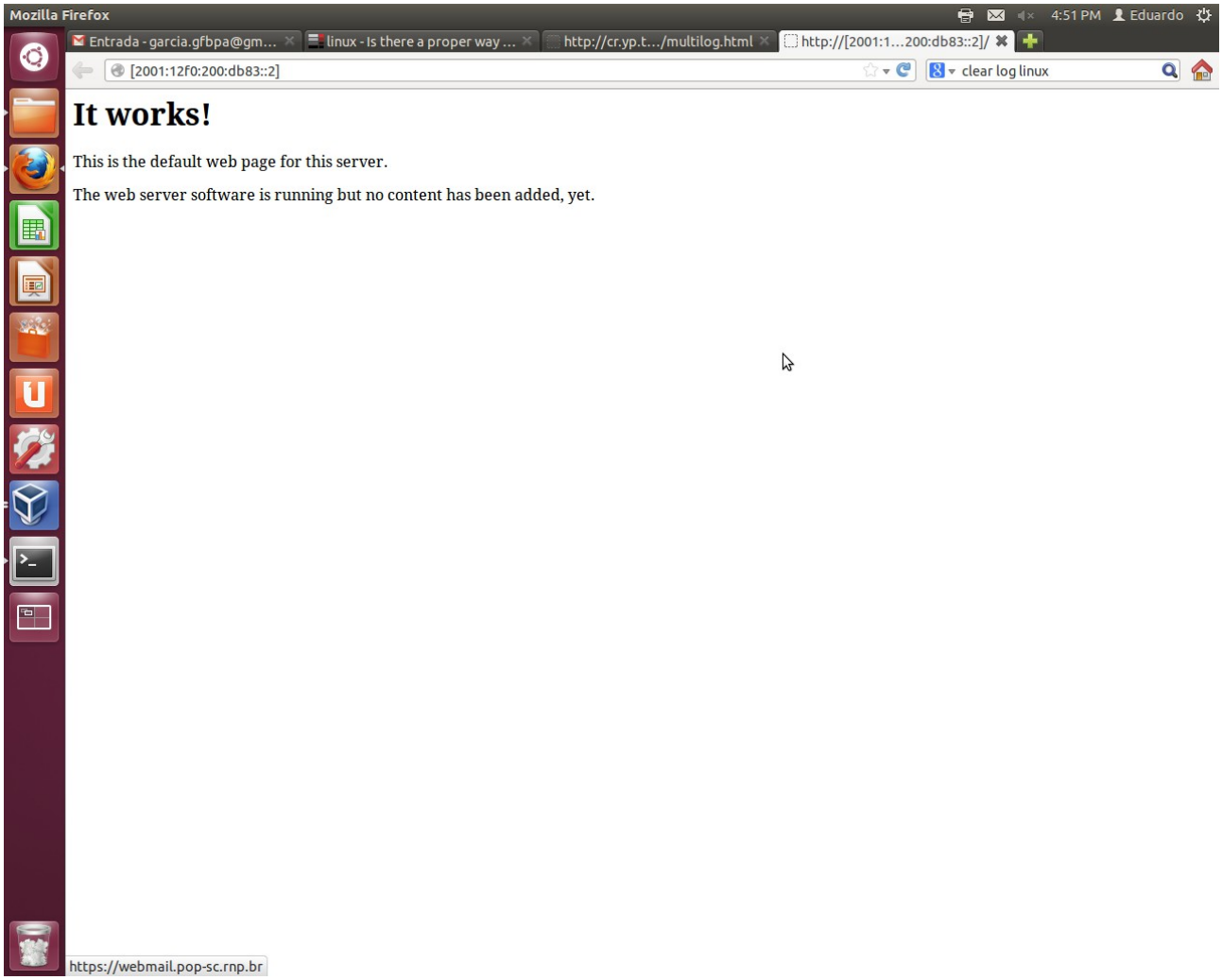

Figura 67: Teste de conectividade com o servidor do cenário via HTTP antes do início do ataque. Fonte: própria

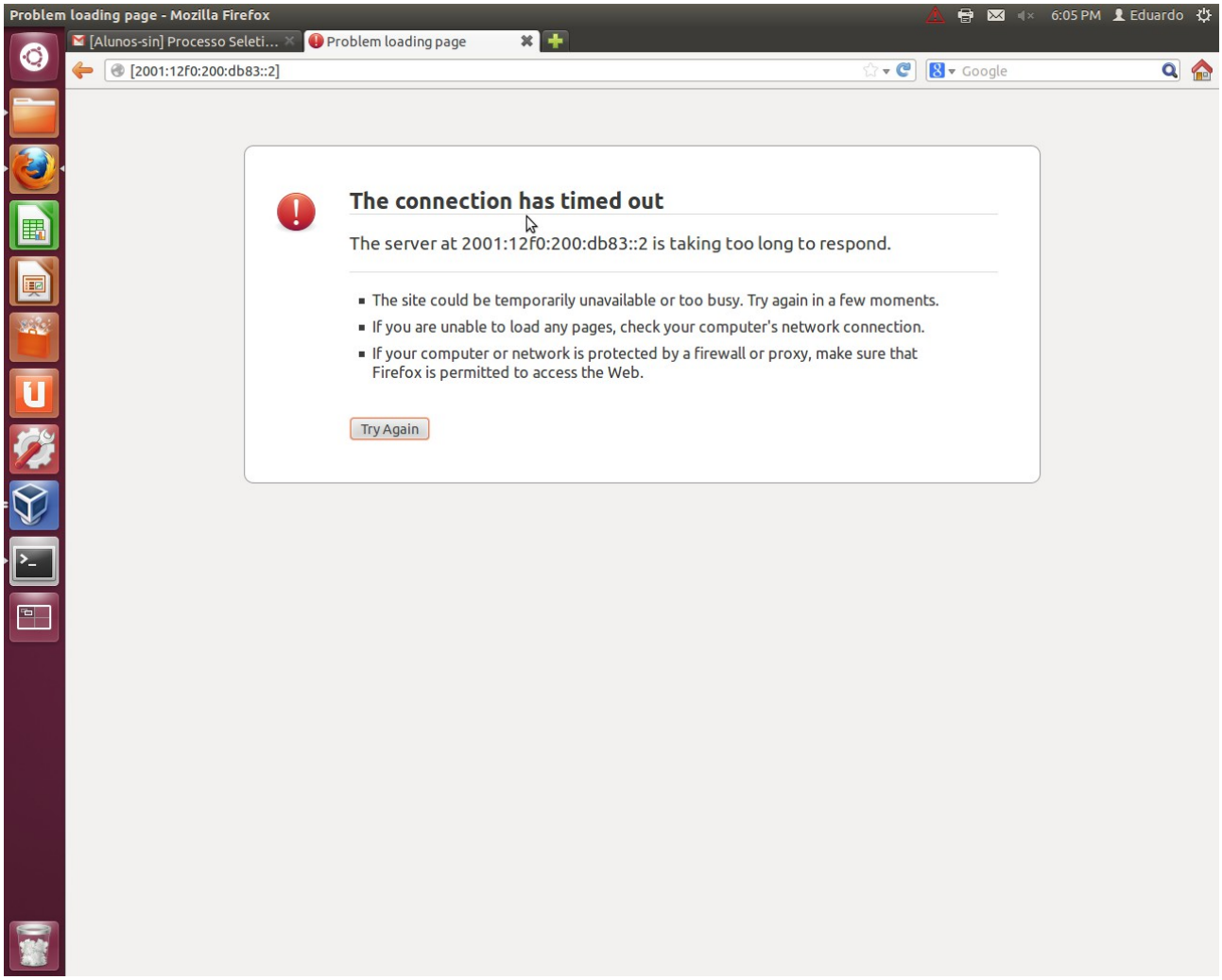

Figura 68: Teste de conectividade com o servidor do cenário via HTTP após o início do ataque. Fonte: própria

Assim como no cenário pilha dupla, o grande volume de mensagens *Router Advertisement* disparadas provocou um ataque de *DoS* sobrecarregando a rede e levando os *hosts* a configurar endereços IPv6 inválidos em suas interfaces de rede.

5.5 Servidor DNS falso

Uma maneira eficaz de atingir especificamente redes IPv6 que se utilizam o mecanismo de tradução NAT64 para se comunicarem com redes puramente IPv4 é realizando um ataque visando o mecanismo DNS64, que é de grande importância para a tradução de endereços uma vez que os endereços IPv4 a serem mapeados em endereços IPv6 são obtidos através de consultas DNS, sendo os registros A convertidos em registros AAAA através do DNS64. Neste experimento, realizado apenas no cenário NAT64, a máquina atacante atuou como um servidor DNS falso que respondia a

todas as consultas por um registro AAAA com o mesmo endereço IPv6. Este ataque causa dois grandes impactos. Primeiramente, a máquina atacante pode capturar tráfego dos *hosts*, uma vez que atuará como servidor DNS na rede local e responderá a consultas dentro da mesma. Além disto, o fato do servidor DNS falso reponder a todas as consultas com o mesmo endereço IPv6 pode simplesmente inviabilizar a comunicação com o servidor IPv4 real.

Este ataque envolveu duas etapas. Primeiramente, foi executada a ferramenta *parasite6* recebendo como parâmetro a interface eth0 da máquina atacante, para que todo o tráfego da rede local fosse desviado para a máquina atacante. Após isto, foi executada aferramenta *fake_dns6d*, ferramenta que atua como o servidor DNS falso, recebendo como parâmetro o endereço IPv6 que seria utilizado para responder às consultas. A Figura 69 mostra a execução destas ferramentas.

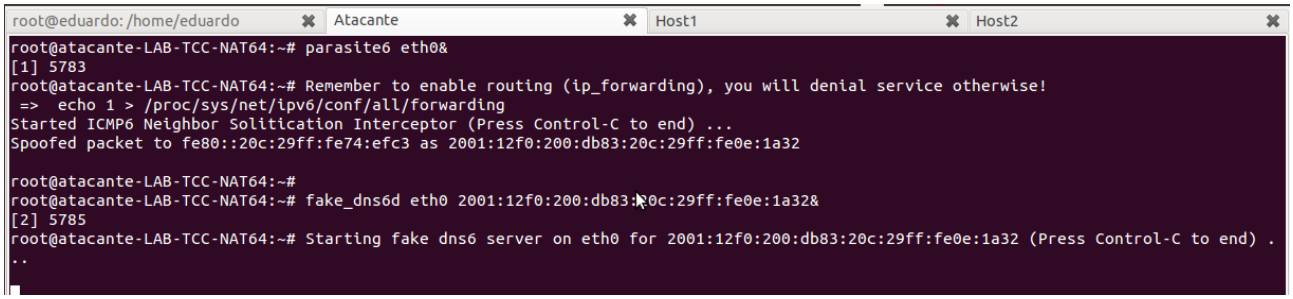

Figura 69: Execução das ferramentas parasite6 e fake_dns6d. Fonte: própria

Para verificar a atuação do servidor DNS falso, foi executado, primeiramente, o comando *ping6* para um servidor puramente IPv4 externo através do nome deste servidor. A Figura 70 mostra a saída do comando *ping6* antes do ataque, é possível verificar que o endereço IPv6 que responde às mensagens *echo-request* enviadas pelo comando *ping6* e o endereço IPv4 do servidor traduzido via NAT64. Já a Figura 71 mostra a saída do comando *ping6* após o ataque. Verifica-se que, desta vez, o endereço IPv6 presente nas respostas é o mesmo endereço passado à ferramenta *fake_dns6d* como parâmetro. Isto significa que, após o início do ataque, as consultas DNS estão sendo respondidas pelo servidor falso. Além disto, após o ataque, foi feita uma tentativa de estabelecer uma conexão HTTP com o servidor externo através do comando *wget*. A Figura 72 mostra a saída deste comando. Pode-se verificar na figura que a conexão não foi bem sucedida pois o endereço IP associado ao nome do servidor externo é o endereço IP falso fornecido pela máquina atacante enquanto servidor DNS falso.

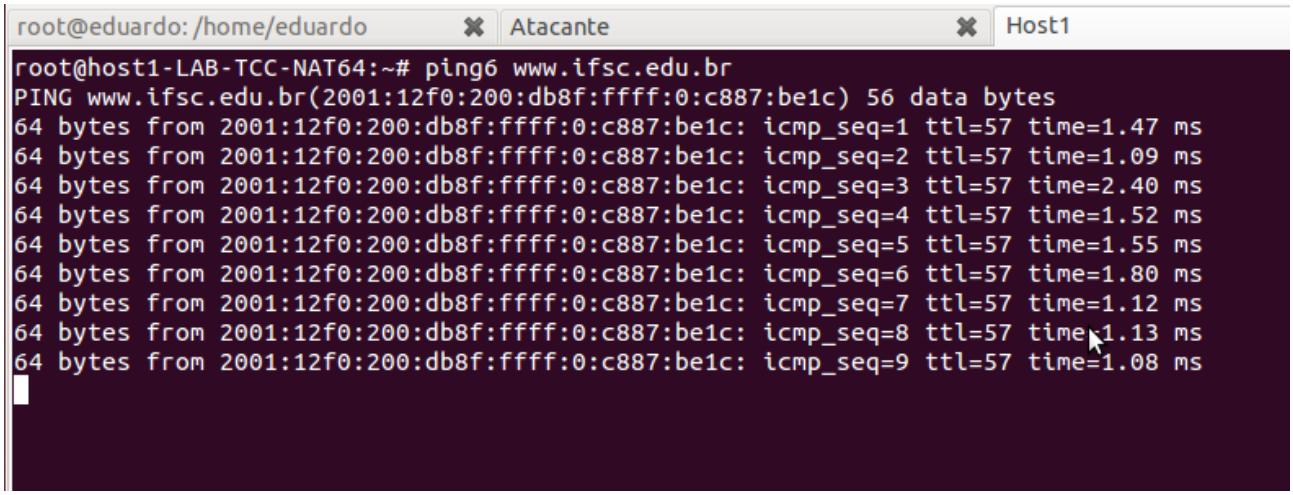

Figura 70: Teste de conectividade via ICMPv6 antes do início do ataque. Fonte: própria

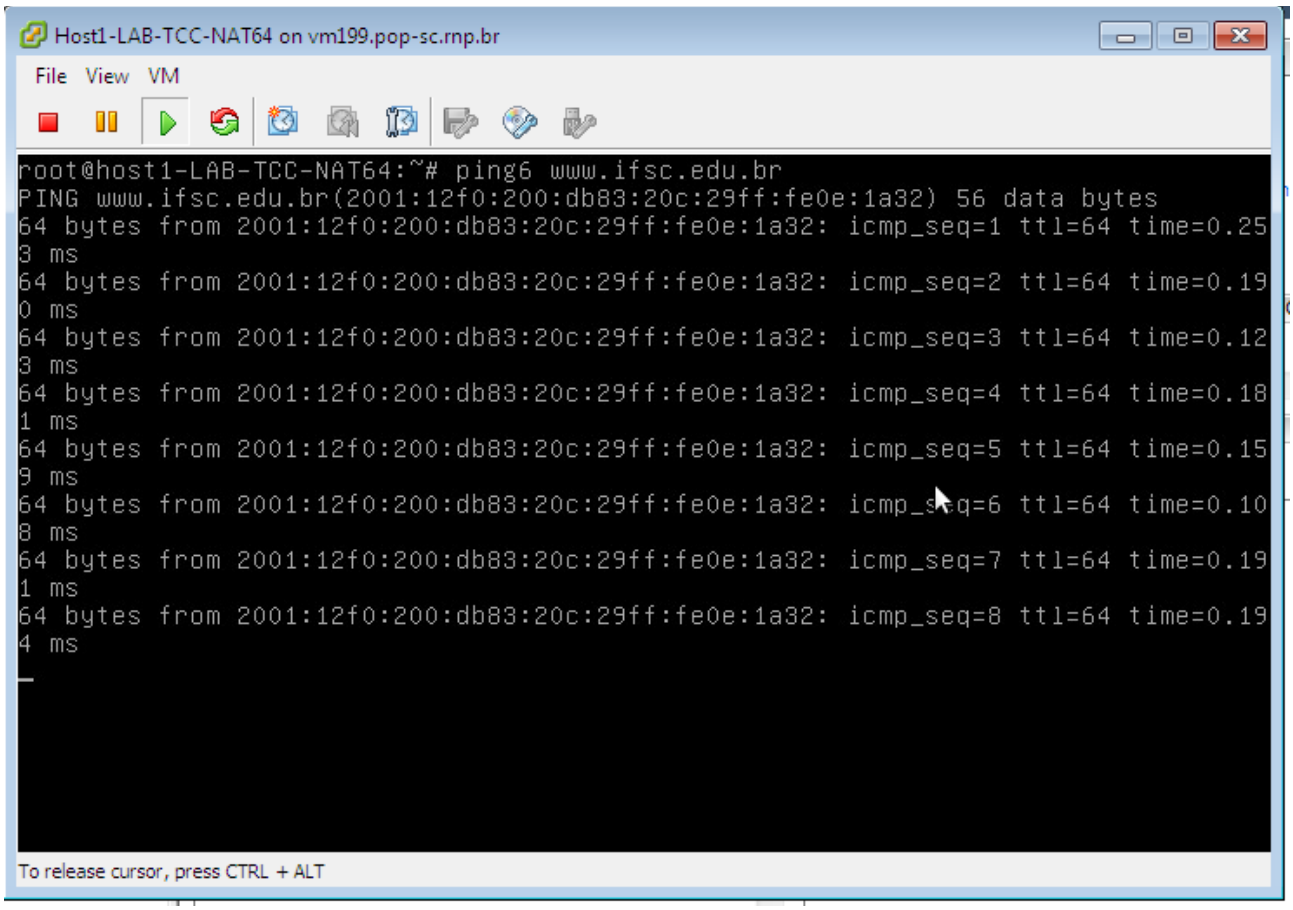

Figura 71: Teste de conectividade via ICMPv6 após o início do ataque. Fonte: própria

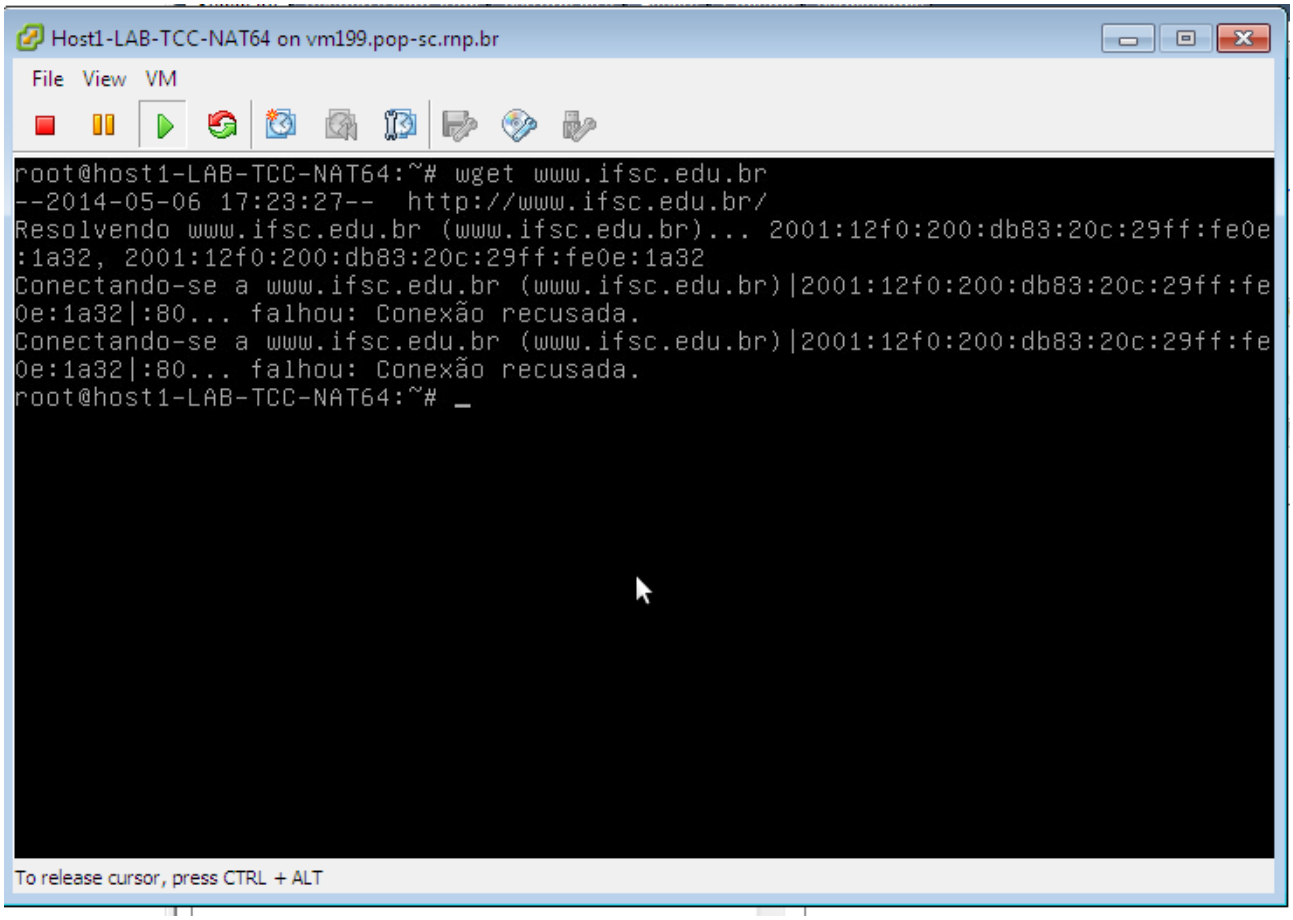

Figura 72: Teste de conectividade com um servidor externo via HTTP. Fonte: própria

Verificou-se neste ataque que a máquina atacante conseguiu atuar como servidor DNS e responder às consultas da rede local, comprometendo de forma crítica a tradução de endereços via NAT64. Este ataque se mostra particularmente crítico pois é difícil implementar uma forma de defesa para o mesmo e redes puramente IPv6 podem ficar totalmente isoladas da Internet IPv4 em função deste ataque.

5.6 Detecção dos ataques

Nas seções de 5.1 a 5.5 deste capítulo, são mostrados diferentes ataques que podem ser realizados com sucesso em uma rede IPv6 levando em consideração técnicas de transição IPv4/IPv6. Além de testar ataques e apontar vulnerabilidades em cenários de transição IPv4/IPv6, está previsto no escopo deste trabalho o estudo da possibilidade de propor uma solução de segurança para estes cenários. Tendo isto em vista, foram estudadas formas de detecção e proteção contra os ataques realizados. Nesta seção, são abordadas as formas de detecção de cada um dos ataques.

5.6.1 Detecção do ataque de *DoS* para novos endereços IPv6

Para detectar este ataque, é necessário verificar o tráfego de mensagens *Neighbor Advertisement* ilegítimas na rede, enviadas pela máquina atacante como se esta fosse algum dos *hosts* do cenário tentando obter um endereço IPv6. Isto pode ser feito sob duas abordagens. Primeiramente, pode-se fazer uma captura de pacotes na interface do *firewall* conectada à rede local do cenário pois, como as mensagens *Neighbor Advertisement* são enviadas ao grupo multicast *All-nodes*, o *firewall* também as receberia. Alternativamente, pode ser utilizada uma ferramenta específica para a detecção deste ataque, como a ferramenta *6Guard*, utilizada neste caso.

A primeira tentativa de detecção deste ataque foi realizada através de captura de pacotes na interface interna do *firewall* do cenário. Para a captura de pacotes, foi utilizado o comando *tcpdump*. Já para a análise foi utilizada, em uma estação de trabalho externa ao cenário, a ferramenta Wireshark, devido ao fato desta prover maior facilidade para a análise da captura de pacotes e também pelo fato do sistema operacional instalado nas máquinas virtuais do cenário não possuir interface gráfica, apenas CLI. A análise da captura de pacotes através do Wireshark é mostrada na Figura 73. Na figura, é possível ver que uma série de mensagens *Neighbor Advertisement*, com diferentes endereços IPv6 de origem, foram capturadas. Neste ataque, a ferramenta *dos_new_ip6* altera o endereço IPv6 de origem das mensagens *Neighbor Advertisement* enviadas, logo, não seria possível detectar as mensagens ilegítimas através do endereço IPv6 de origem. No entanto, o endereço MAC de origem da mensagem é gerado aleatoriamente, logo, a mensagem *Neighbor Advertisement* falsa, apesar de conter o endereço mesmo endereço IPv6 de origem de um dos *hosts* do cenário, não conterá o mesmo endereço MAC. Em destaque na cor verde na Figura 73, está a opção endereço MAC alvo da mensagem *Neighbor Advertisement*, que contem como valor o endereço MAC do *host* que envia a mensagem, neste caso, o endereço MAC falso gerado pela ferramenta *dos_new_ip6*. Na Figura 74, é mostrada a configuração da interface de rede eth0 do *host* que tenta se conectar à rede e receber o endereço IPv6 anunciado pela máquina atacante na mensagem mostrada em detalhes na Figura 73. Pode-se verificar que o endereço MAC de origem da mensagem não corresponde ao endereço MAC do *host*, em destaque na cor verde na Figura 74. Portanto, pode-se, agora, concluir que esta mensagem *Neighbor Advertisement* se trata de uma mensagem falsa e proveniente de um ataque.

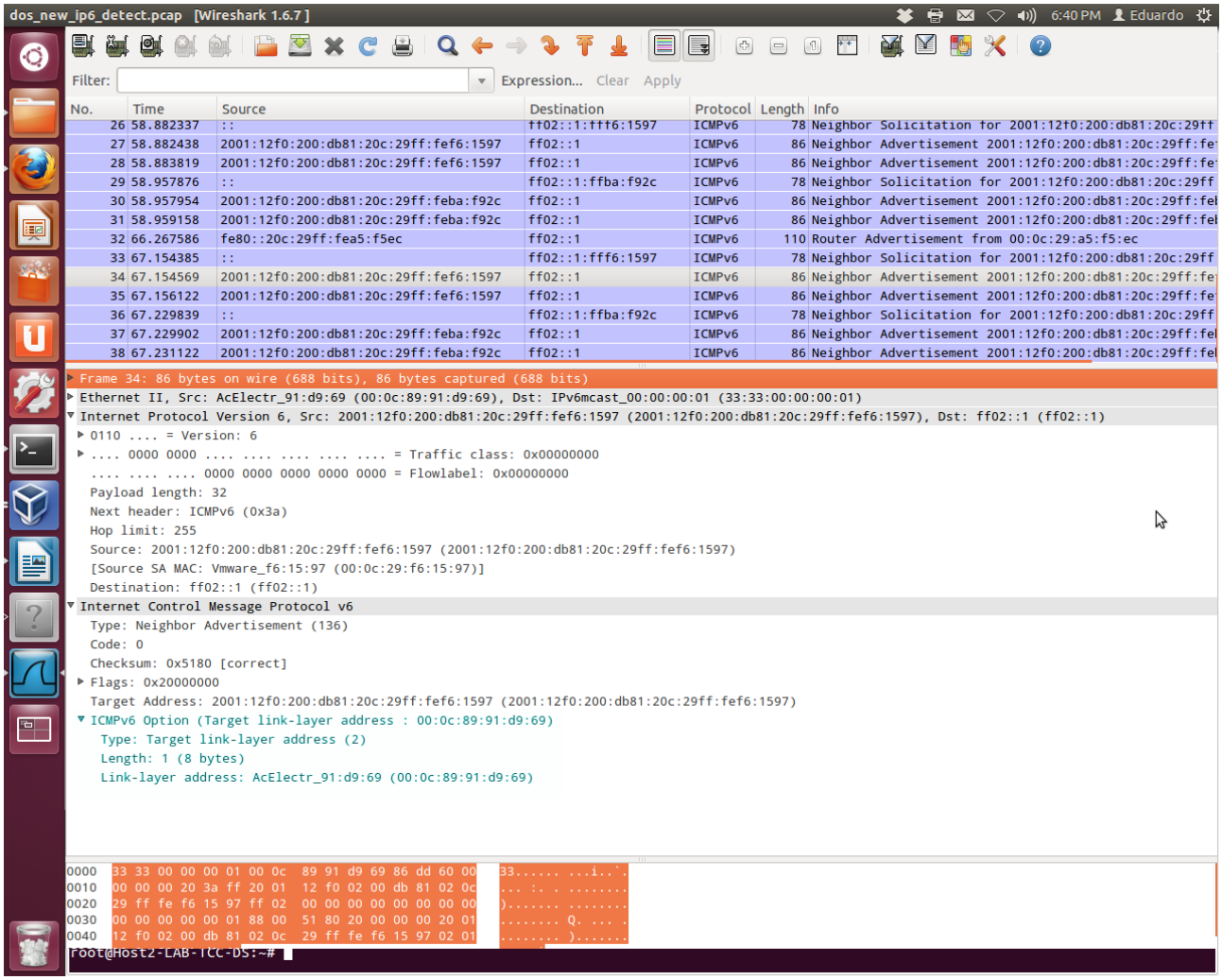

Figura 73: Captura de pacotes para detecção do ataque de DoS para novos endereços IPv6. Fonte: própria

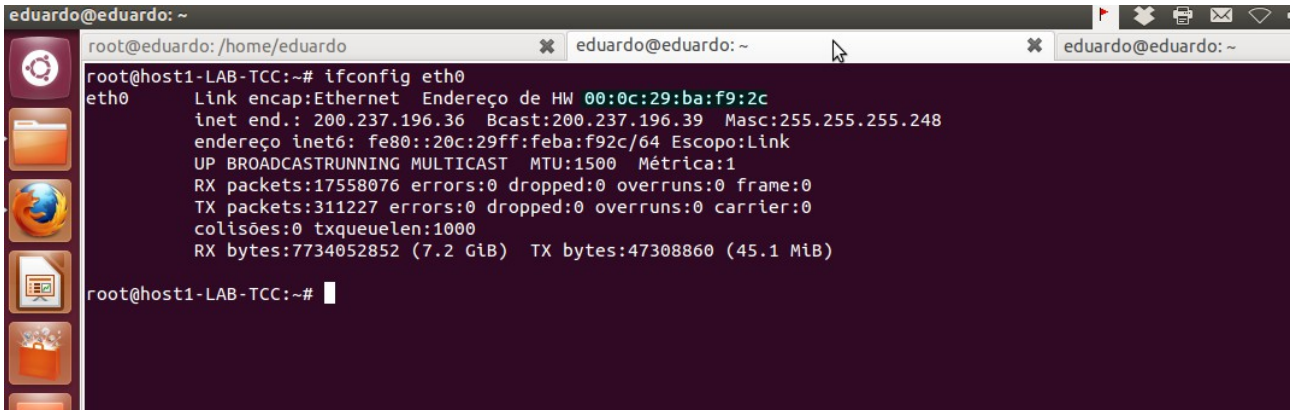

Figura 74: Configuração da interface eth0 de um dos hosts do cenário após o ataque de DoS para novos endereços IPv6. Fonte: própria

Alternativamente à captura de pacotes, pode ser utilizada a ferramenta *6Guard*. Ao ser executado via CLI, ao analisar as mensagens *Neighbor Advertisement* recebidas, *6Guard* consegue detectar os pacotes ilegítimos oriundos da máquina atacante e gerar mensagens de alerta indicando que a rede local está sob ataque. A Figura 75 mostra as mensagens de alerta geradas pelo *6Guard*. Para cada mensagem *Neighbor Advertisement* proveniente da máquina atacante, é gerada uma mensagem de alerta mostrando informações como tipo do ataque, ferramenta utilizada para o ataque, endereço MAC do atacante, entre outras. Nota-se que nas mensagens de alerta geradas são apresentados diversos endereços MAC diferentes que, supostamente, pertencem a diversos fabricantes. Isto deixa claro que, na verdade, trata-se de endereços MAC forjados.

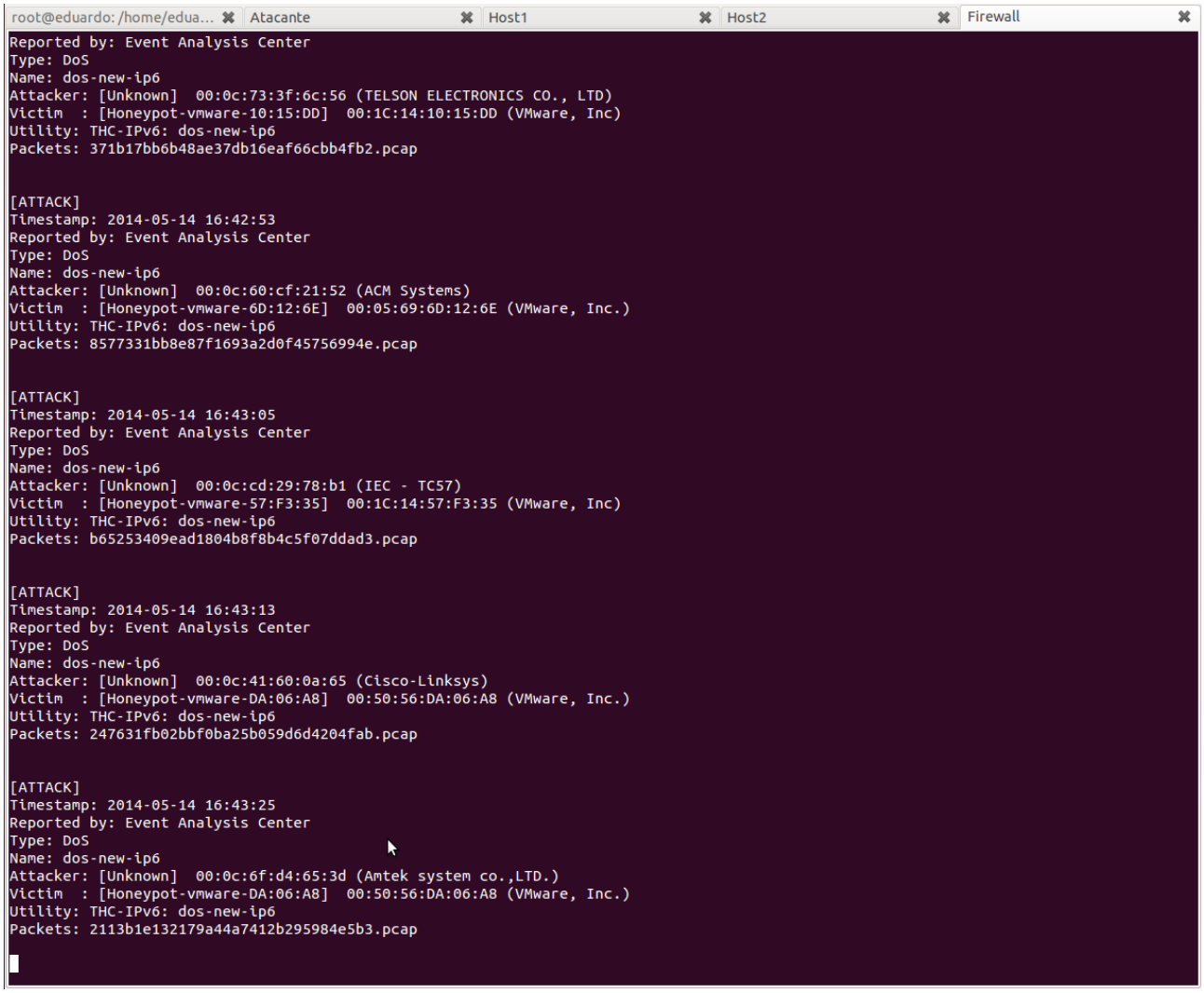

Figura 75: Detecção do ataque de DoS para novos endereços IPv6 através da ferramenta 6Guard. Fonte: própria

5.6.2 Detecção do ataque de anúncio de um roteador falso

Este ataque pode ser detectado de forma similar ao ataque de *DoS* para novos endereços IPv6. Neste caso, é necessário atentar para o tráfego de mensagens *Router Advertisement* ilegítimas, ou seja, que não sejam provenientes do roteador, na rede local. As duas abordagens de detecção utilizadas para o ataque de *DoS* para novos endereços IPv6 se aplicam também para este ataque. Como as mensagens *Router Advertisement* são enviadas para o endereço multicast *all-nodes*, uma captura de pacotes na interface do *firewall* conectada à rede local do cenário capturará mensagens *Router Advertisement* provenientes desta rede, como é o caso da máquina atacante. Além disto, a ferramenta *6Guard* também é capaz de detectar este ataque.

A primeira tentativa de detecção deste ataque foi realizada através de captura de pacotes na interface interna do *firewall* do cenário. Pelas mesmas razões expostas na Seção 5.6.1, foram utilizadas as ferramentas *tcpdump* e Wireshark para captura e análise da captura de pacotes, respectivamente. A Figura 76 mostra a análise da captura de pacotes através do Wireshark. Pode-se observar que foram capturadas diversas mensagens *Router Advertisement* provenientes de diferentes origens. Sabendo se que há apenas uma máquina enviando mensagens *Router Advertisement* para a rede local (o *firewall*, no caso do cenário pilha dupla, ou o próprio roteador, no caso do cenário NAT64), pode-se verificar na captura de pacotes as mensagens *Router Advertisement* cujo endereço IPv6 de origem não corresponde ao da máquina que atua como roteador no cenário. Na Figura 76, a mensagem *Router Advertisement* destacada na cor marrom corresponde a uma mensagem *Router Advertisement* ilegítima.

No cenário pilha dupla, existe a alternativa ainda mais simples de acessar um dos *hosts* remotamente via IPv4 e verificar a tabela de roteamento deste *host* através do comando *route -n6*, como é mostrado na Figura 77. Uma rota padrão diferente daquela anunciada pela máquina que atua como roteador no cenário indica que a rede local está sob ataque.

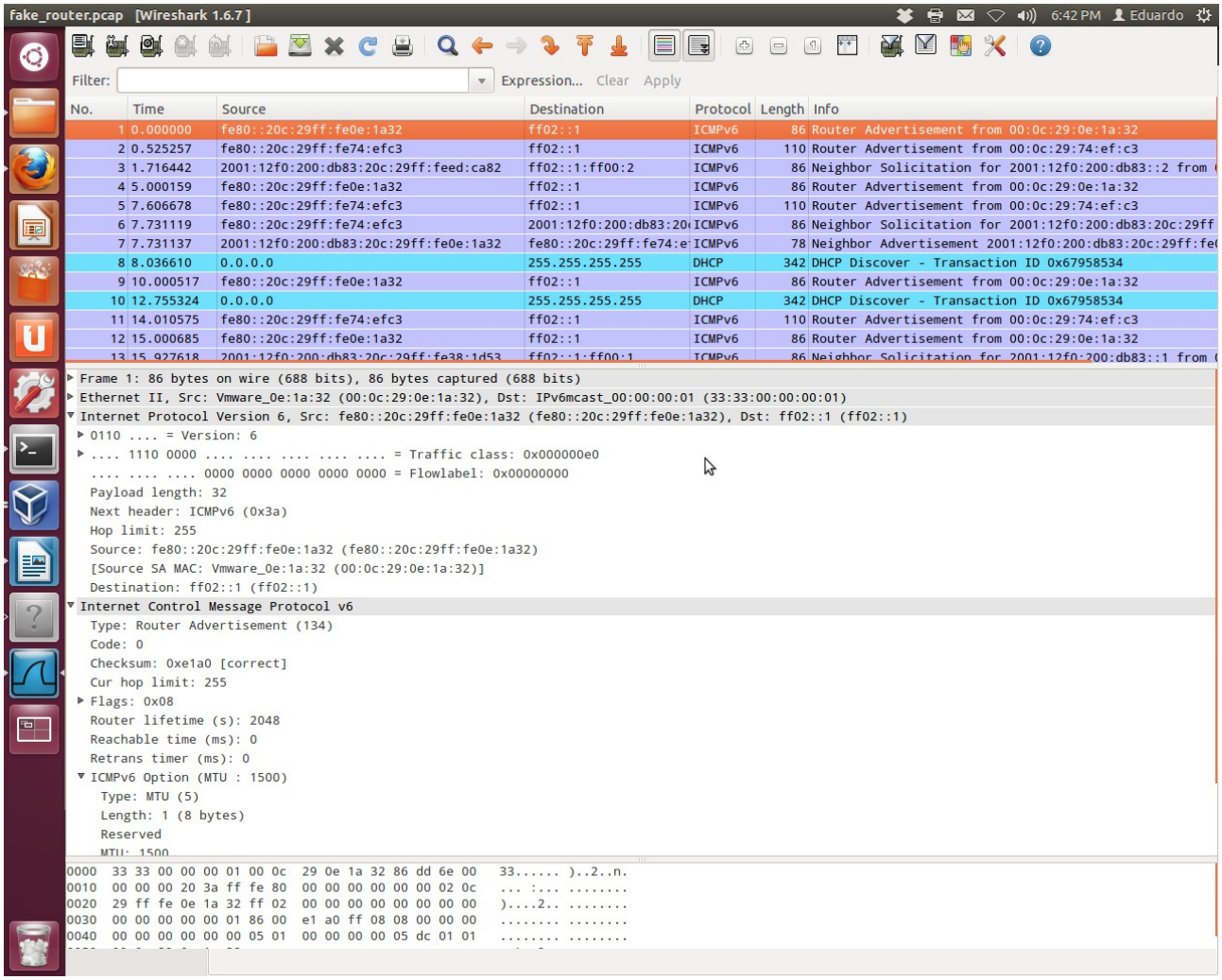

Figura 76: Captura de pacotes para detecção do ataque de DoS através do anúncio de um roteador falso. Fonte: própria

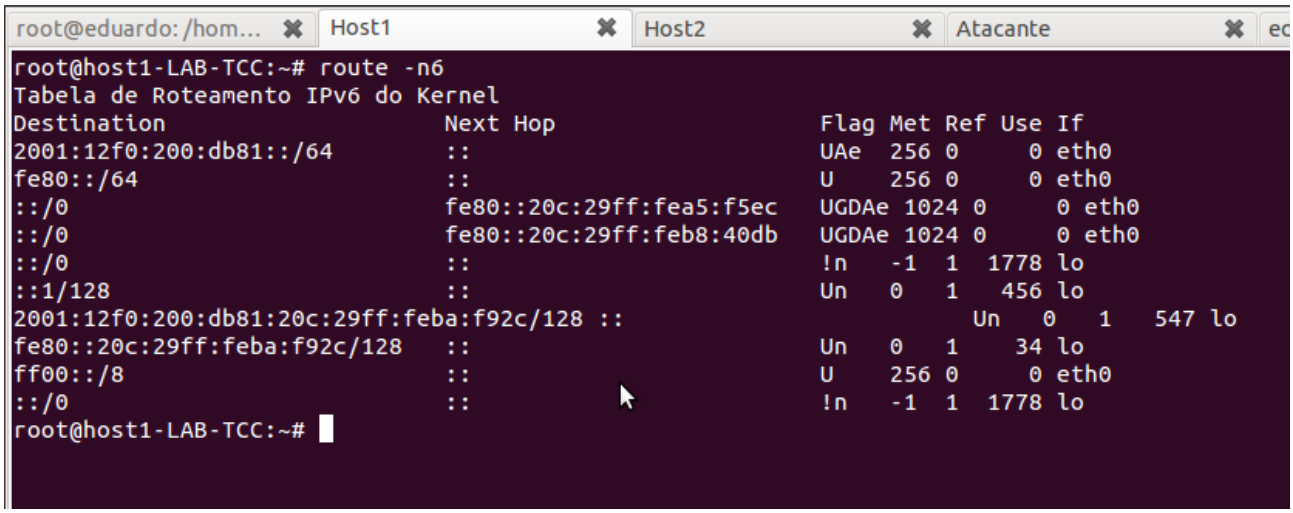

Figura 77: Detecção do ataque de DoS através do anúncio de um roteador falso via route -n6 no cenário pilha dupla. Fonte: própria

Alternativamente à captura de pacotes, pode ser utilizada a ferramenta *6Guard*. Assim como as mensagens *Neighbor Advertisement*, o *6Guard* é capaz de analisar as mensagens *Router Advertisement* recebidas, detectar mensagens ilegítimas e gerar mensagens de alerta indicando que a rede está sob ataque. A Figura 78 mostra as mensagens de alerta geradas pelo *6Guard*. Para cada mensagem *Router Advertisement* ilegítima detectada, é gerada uma mensagem de alerta contendo informações como o tipo do ataque, ferramenta utilizada para o ataque, endereço MAC do atacante, entre outras. Nota-se que todas as mensagens de alerta apontam o mesmo endereço IPv6 como origem das mensagens *Router Advertisement* detectadas, isto é, trata-se do endereço IPv6 da máquina atacante.

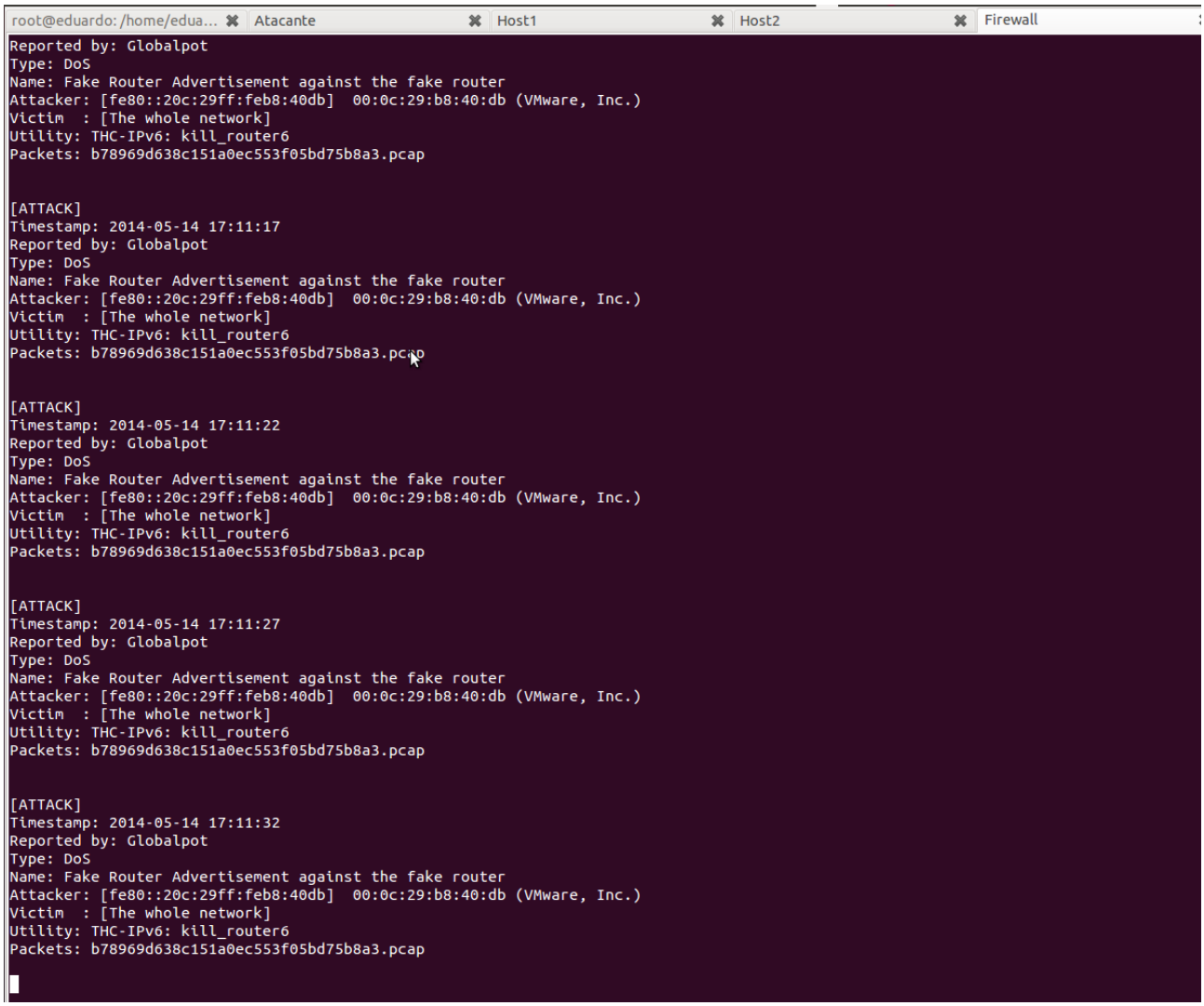

Figura 78: Detecção do ataque de DoS através do anúncio de um roteador falso via 6Guard.
Fonte: própria

5.6.3 Detecção do ataque de *flooding* de mensagens *Neighbor Advertisement*

A forma de detectar este ataque é bastante semelhante à mencionada na Seção 5.6.1, visto que este ataque também envolve *spoofing* e envio de mensagens *Neighbor Advertisement* para a rede local. Porém, apesar da ferramenta *6Guard* também ser capaz de detectar este ataque, a captura de pacotes se mostrou mais eficiente.

Ao ser executado para a detecção deste ataque, o *6Guard* analisa as mensagens *Neighbor Adverrtisement* recebidas, detecta as mensagens oriundas da máquina atacante e gera mensagens de alerta indicando que a rede está sob ataque, contendo informações como o tipo do ataque, ferramenta utilizada para o ataque, endereço MAC do atacante, entre outras. A Figura 79 mostra as mensagens de alerta geradas pelo *6Guard* para este ataque. Observa-se que as mensagens *Neighbor Advertisement* ilegítimas são detectadas, porém são geradas mensagens indicando diferentes tipos de ataques, deixando dúbia a análise.

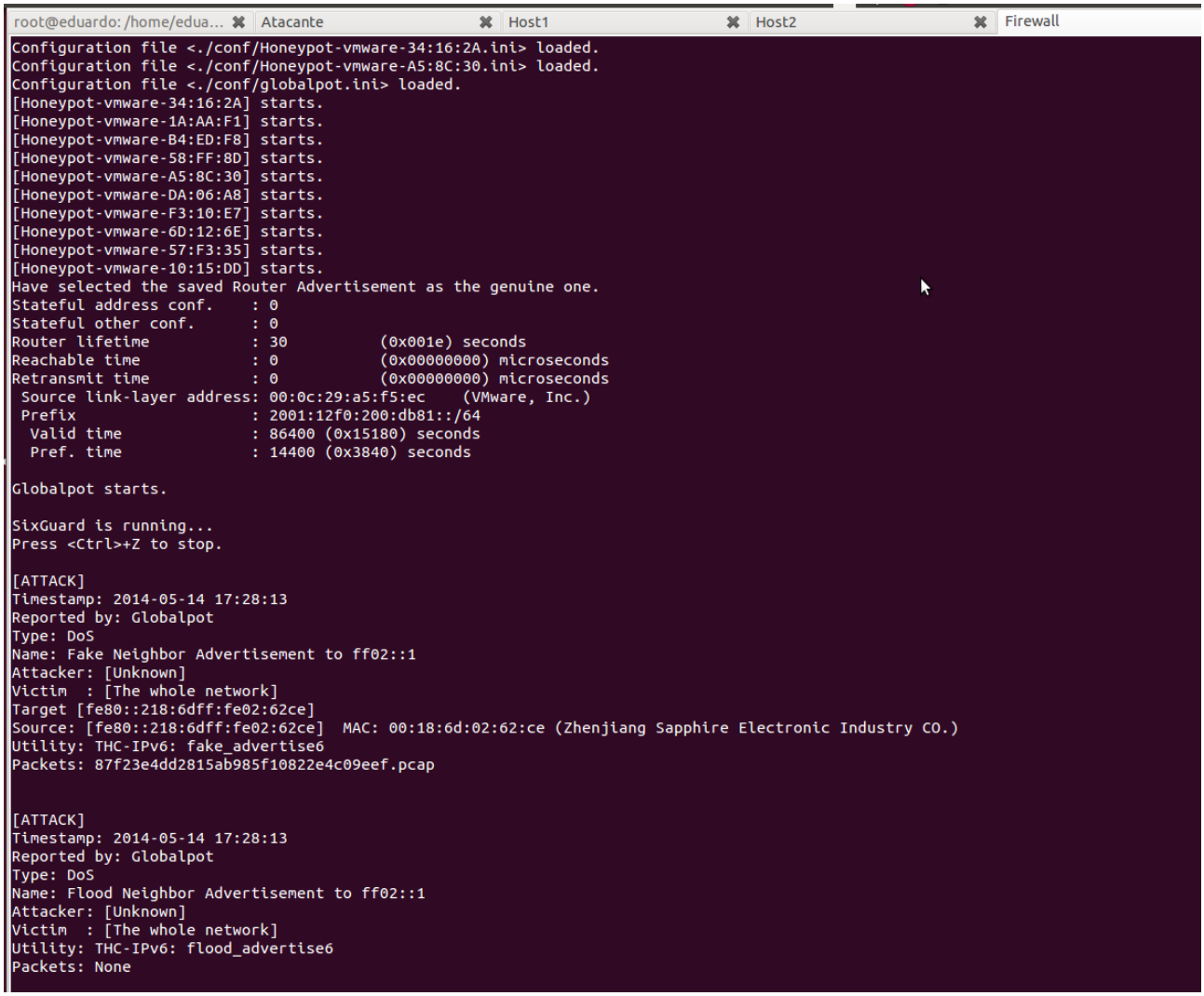

Figura 79: Detecção do ataque de flooding de mensagens Neighbor Advertisement via 6Guard. Fonte: própria

O *6Guard* consegue detectar o ataque de *flooding* de mensagens *Neighbor Advertisement*, porém, não deixa claro qual tipo de ataque está ocorrendo na rede. Ao realizar uma captura de pacotes na interface interna do *firewall* dos cenários, verificou-se que em um curto espaço de tempo foi capturado um grande número de pacotes. Ao analisar a captura de pacotes, verificou-se que um grande número de mensagens *Neighbor Advertisement* foi capturado e que cada uma destas mensagens *Neighbor Advertisement* possuía um endereço IPv6 de origem diferente, caracterizando os endereços aleatórios gerados pela ferramenta *flood_advertise6*. A captura de pacotes, portanto, deu uma maior certeza de que o ataque se tratava de um ataque de *flooding*. A Figura 80 mostra a análise da captura de pacotes para a detecção deste ataque. Pelas mesmas razões expostas na Seção 6.6.1, foram utilizadas as ferramentas *tcpdump* e Wireshark para a captura e análise da captura de

pacotes, respectivamente.

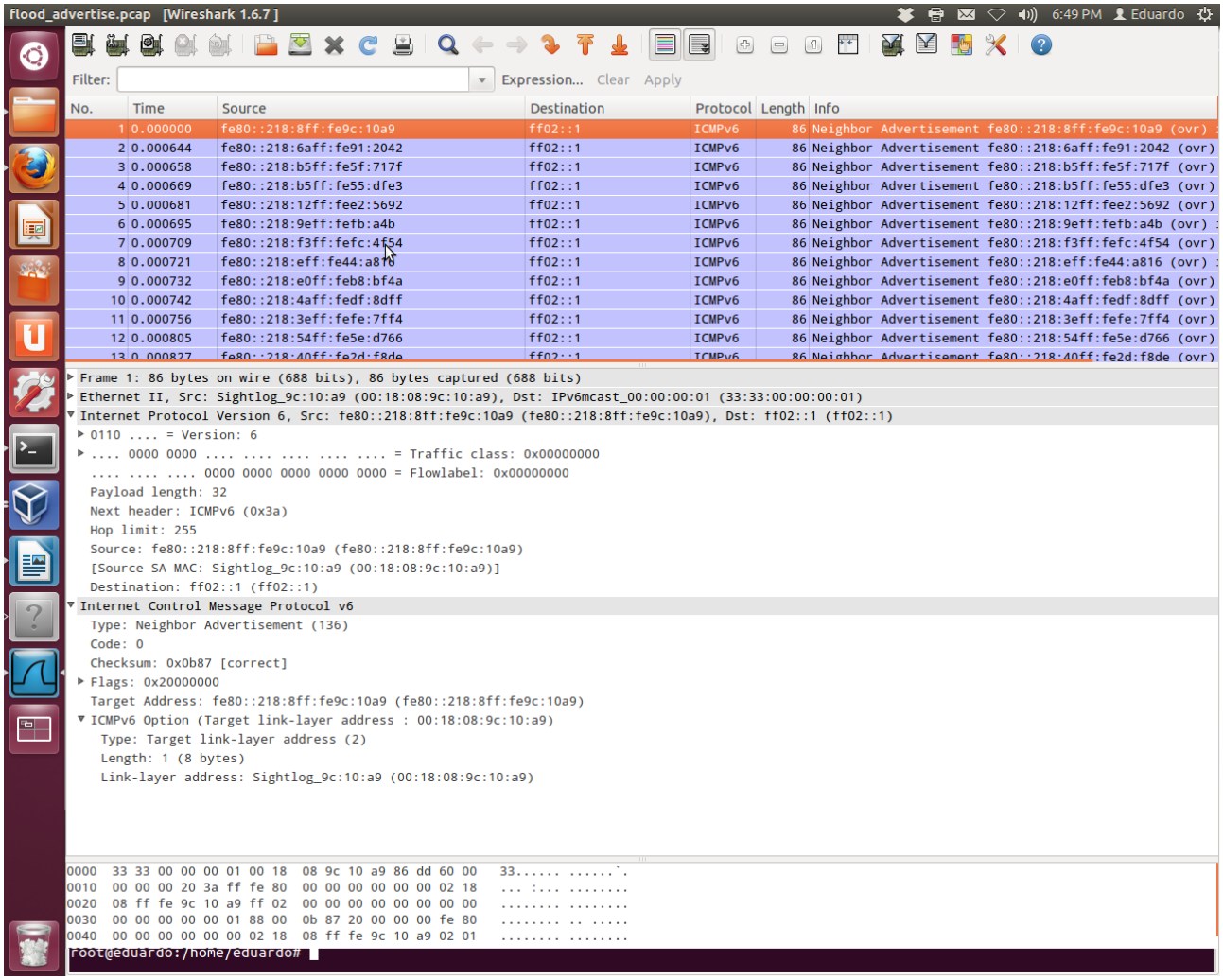

Figura 80: Análise da captura de pacotes para detecção do ataque de flooding de mensagens Neighbor Advertisement via 6Guard. Fonte: própria

5.6.4 Detecção do ataque de *flooding* de mensagens *Router Advertisement*

Por se tratar de um ataque que envolve *spoofing* e envio de mensagens *Router Advertisement* para a rede local, foi utilizada uma abordagem semelhante à apresentada na Seção 5.6.2 para a detecção deste ataque. No cenário pilha dupla, existe a opção de acessar remotamente um dos *hosts* via IPv4 e verificar configuração da interface de rede deste *host* através do comando *ifconfig*. Para ambos os cenários, a ferramenta *6Guard* é capaz de detectar as mensagens *Router Advertisement* falsas, assim como é possível capturá-las através da interface interna do *firewall*.

Uma verificação simples, mas que pode sinalizar a ocorrência do ataque, é a verificação da

configuração da interface de rede dos *hosts*, possível no cenário pilha dupla pois neste cenário ainda é possível acessar os *hosts* remotamente com o ataque em andamento. A Figura 81 mostra a verificação da configuração da interface de rede eth0 de um dos *hosts* através do comando *ifconfig*. Pode-se verificar a existência de múltiplos endereços IPv6, ou seja, múltiplas máquinas enviando mensagens *Router Advertisement* no cenário, enquanto apenas uma máquina deve enviar mensagens *Router Advertisement*. Isto indica que há mensagens *Router Advertisement* ilegítimas trafegando na rede local.

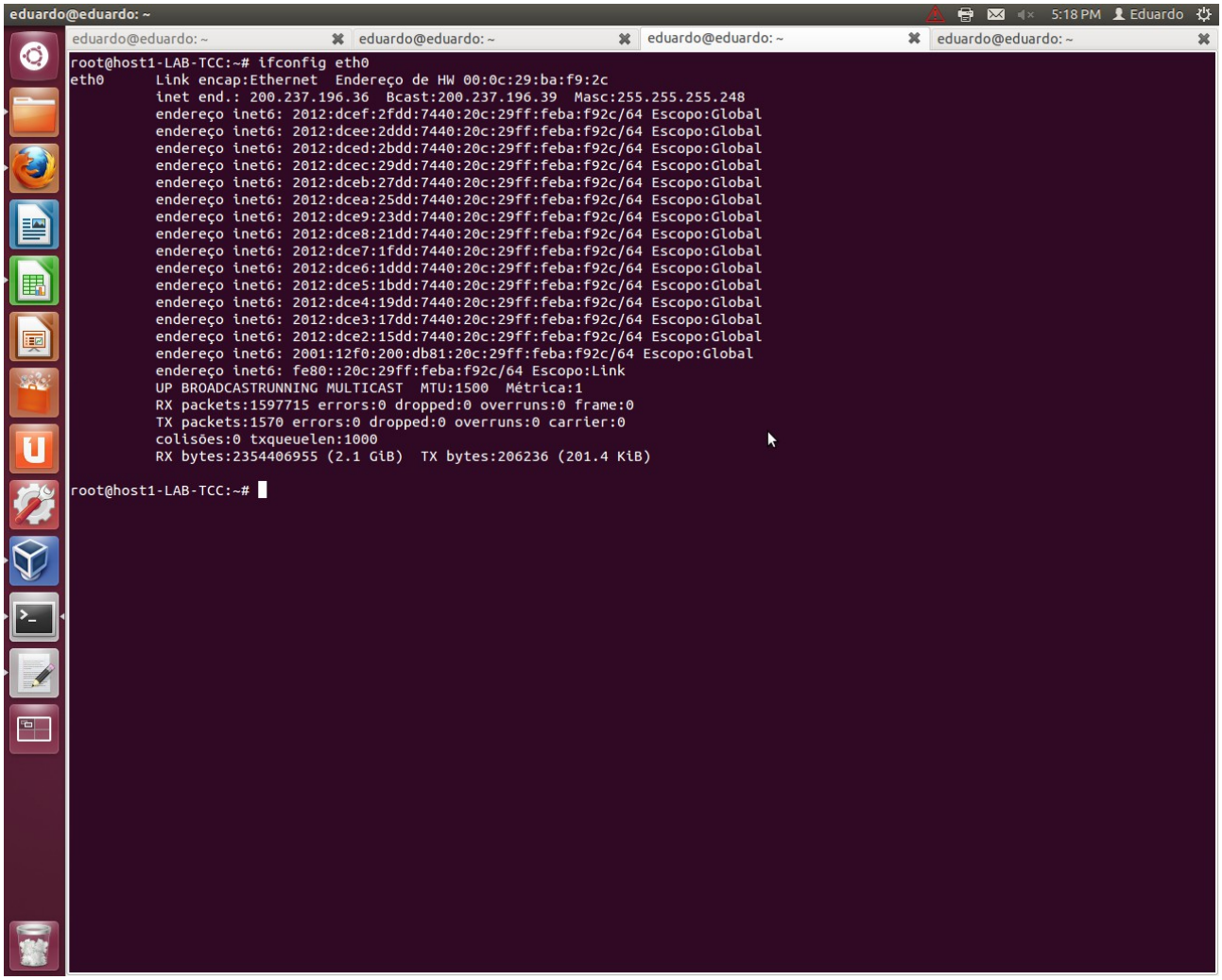

Figura 81: Verificação da configuração da interface eth0 de um host para detecção do ataque de flooding de mensagens Router Advertisement. Fonte: prórpria

De forma mais precisa, o *6Guard* consegue detectar o ataque analisando as mensagens *Router Advertisement* recebidas. Para os casos de mensagens ilegítimas detectadas, serão geradas mensagens de alerta contendo informações como o tipo do ataque, ferramenta utilizada para o ataque, endereço MAC do atacante, entre outras. A Figura 82 mostra a detecção deste ataque através

do *6Guard*. As mensagens de alerta mostram que diversas mensagens *Router Advertisement* foram detectadas e cada uma delas contém um endereço IPv6 de origem diferente, o que é característico da ferramenta *flood_router26*, que gera pacotes com endereços de origem aleatórios. A única imprecisão na detecção é o fato de o *6Guard* apontar o ataque como sendo anúncio de um roteador falso. No entanto, ainda há um forte indício de *flooding* em função do grande volume de mensagens capturadas.

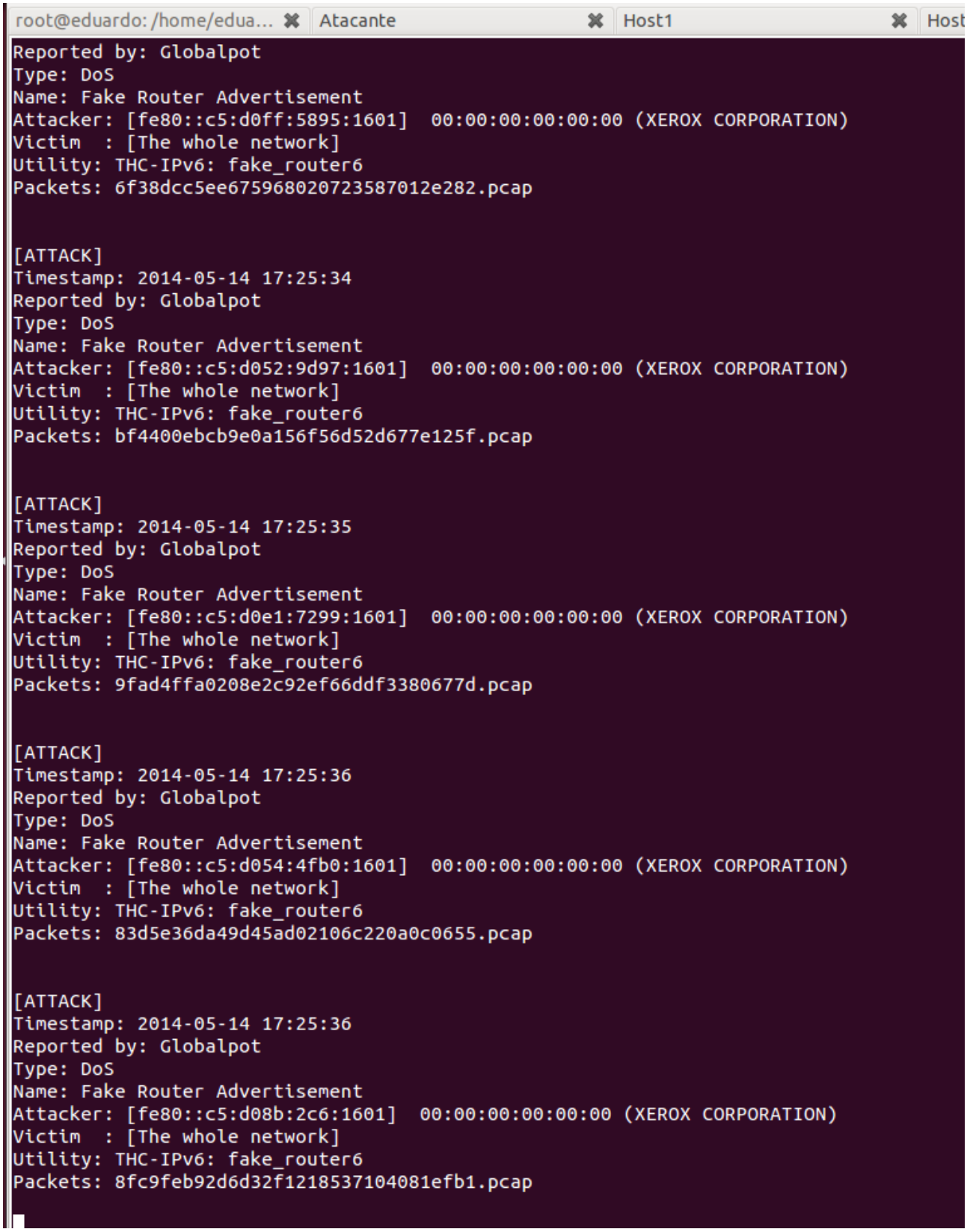

Figura 82: Detecção do ataque de flooding de mensagens router advertisement via 6Guard. Fonte: própria

A captura de pacotes também pode ser utilizada para a detecção deste ataque. Em um curto espaço de tempo, foi possível capturar um grande número de pacotes. Ao analisar a captura de pacotes, percebe-se a captura de um grande número de mensagens *Router Advertisement* contendo endereços IPv6 de origem diferentes, o que caracteriza o funcionamento da ferramenta *flood_router26,* que envia um grande número de mensagens *Router Advertisement* para a rede local. A Figura 83 mostra a captura de pacotes realizada. Pelas mesmas razões expostas na Seção 5.6.1, foram utilizadas as ferramentas *tcpdump* e Wireshark para a captura e análise da captura de pacotes, respectivamente.

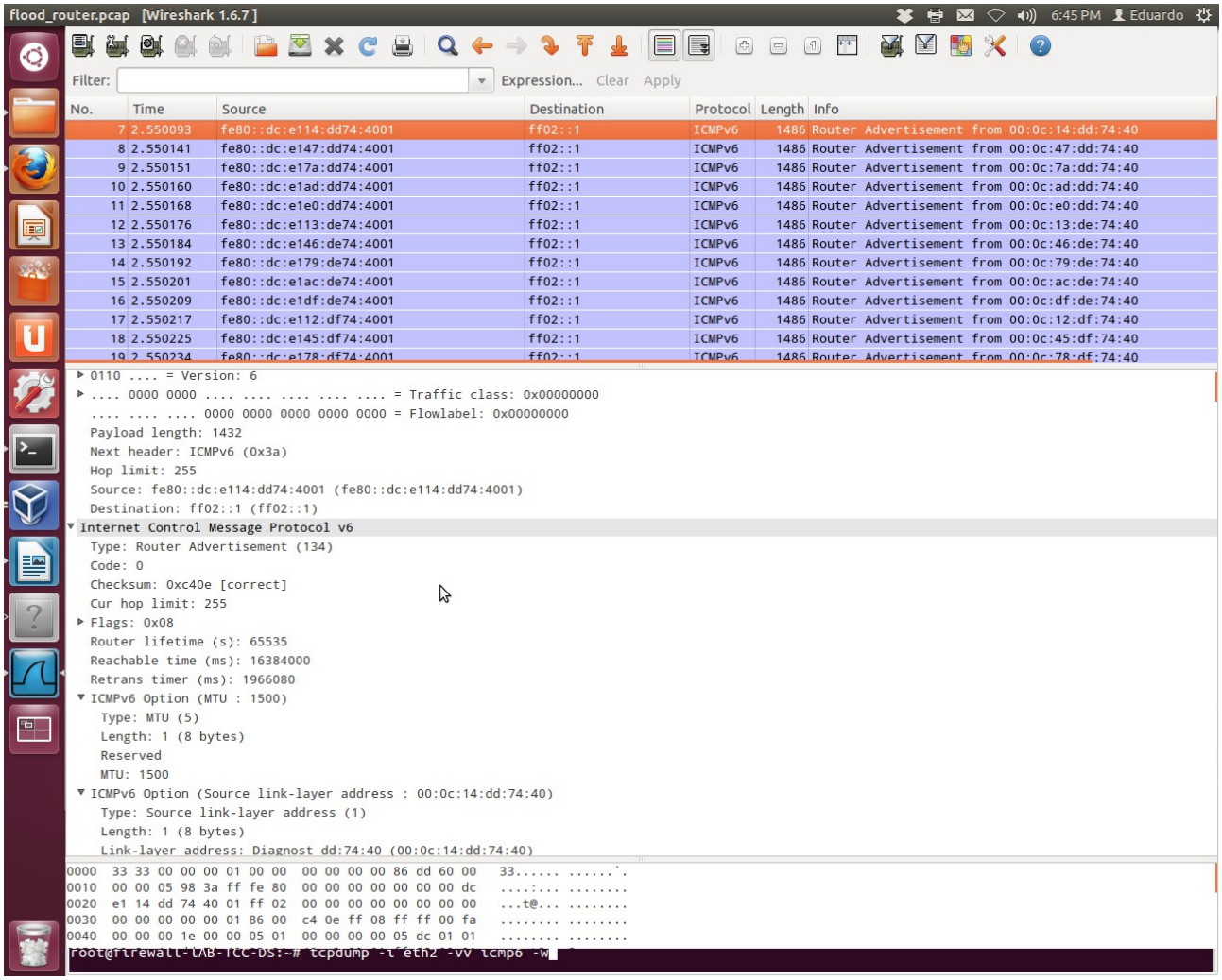

Figura 83: Captura de pacotes para a detecção do ataque de flooding de mensagens Router Advertisement. Fonte: própria

5.6.5 Detecção do ataque de anúncio de um servidor DNS falso

Este ataque se revelou o de mais difícil detecção, pois o *6Guard* não é capaz de detectá-lo e realizar uma captura de pacotes a partir do *firewall* do cenário não é uma medida eficaz para a detecção deste ataque, uma vez que o tráfego da rede local é desviado para a máquina atacante. Uma possibilidade de detecção deste ataque é através da própria tentiva de acesso a um servidor externo a partir de um dos *hosts* do cenário, como é mostrado na Figura 84. Ao tentar acessar o servidor externo e verificar que a tentativa de conexão foi mal sucedida, pode-se verificar que o nome do servidor externo foi resolvido em um endereço IPv6 que está contido na faixa de endereçamento da rede local. Pode-se concluir que há um provável atacante dentro da própria rede local direcionando seu ataque ao DNS.

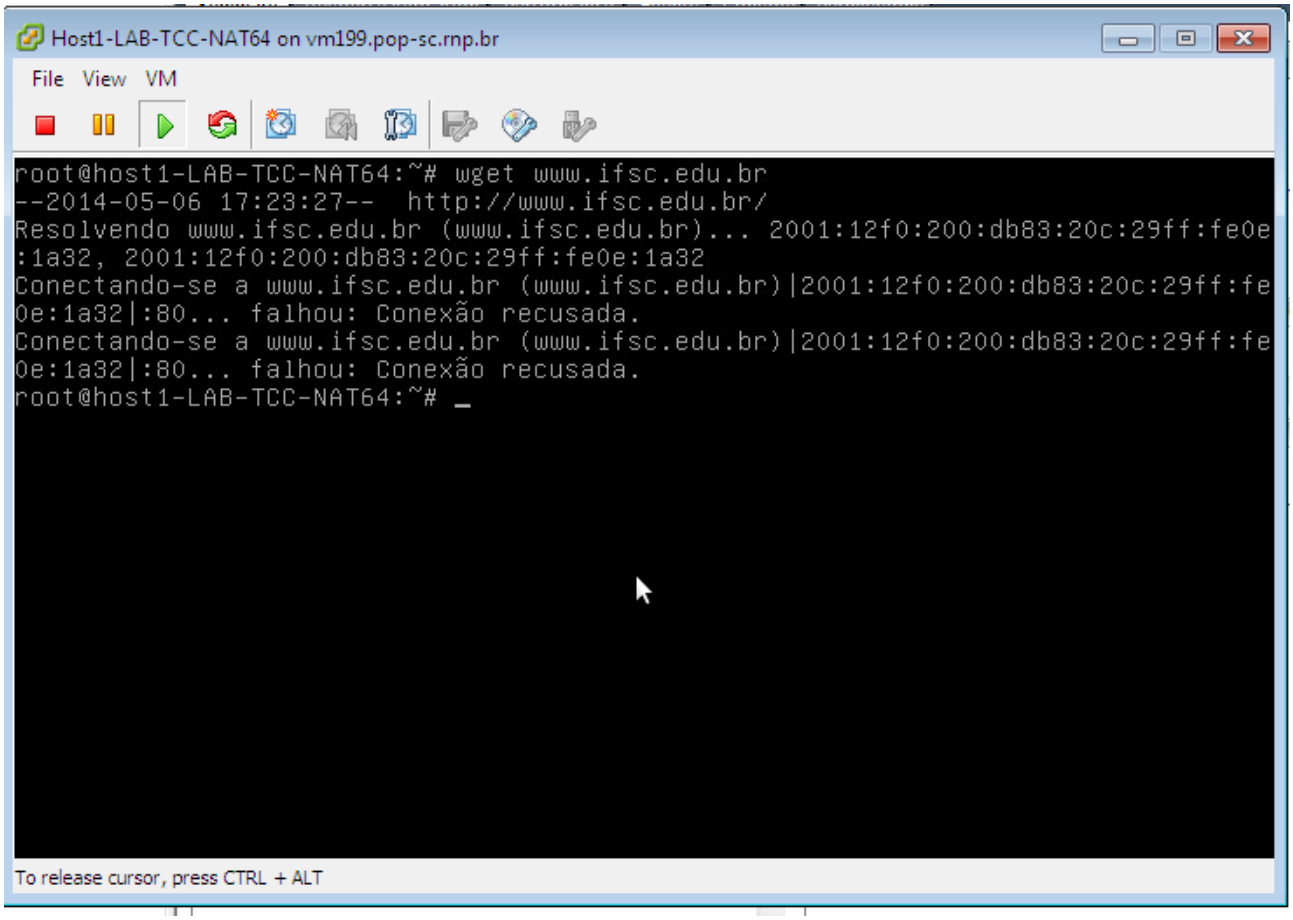

Figura 84: Detecção do ataque do anúncio de um servidor DNS falso através de tentativa de acesso a um servidor externo a partir de um host do cenário. Fonte: própria

5.7 Possíveis formas de defesa

Após a verificação de vulnerabilidades nos cenários de testes, propôs-se que neste trabalho seria buscada uma solução para as vulnerabilidades apontadas. A busca desta solução envolve levantar formas de detecção dos ataques, mostradas na Seção 5.6, e formas de defesa contra os mesmos. De forma geral, os ataques realizados envolvem NDP, autoconfiguração de endereços e DNS64. Tendo isto em vista, foram levantadas três formas de defesa que podem cobrir este espectro de ataques: *Secure Neighbor Discovery* (SEND), *Router Advertisement Guard* (RA Guard) e *DNS Security* (DNSSEC). Estas formas de defesa, bem como possibilidades de implantação das mesmas, são detalhadas nas seções 5.7.1, 5.7.2 e 5.7.3.

5.7.1 *Secure Neighbor Discovery* – SEND

O SEND foi definido para proteger o NDP de ameaças que visam este mecanismo. O SEND acrescenta ao NDP um conjunto de novas funcionalidades e opções para torná-lo mais seguro, protegendo as mensagens trocadas no processo de descoberta de vizinhos. A principal funcionalidade de segurança do SEND é a utilização de endereços gerados criptograficamente (*Cryptographically Generated Address* – CGA). Estes endereços são utilizados para garantir que um nó que envia uma mensagem *Neighbor Advertisement* informando que já possui um determinado IPv6 realmente possui aquele endereço. Antes que os nós de uma rede possam enviar mensagens informando que possuem um determinado endereço IPv6, estes nós geram um par de chaves pública e privada. Os nós da rede trocam chaves públicas através da opção CGA, inserida nas mensagens do NDP com o SEND. Outras funcionalidades importantes do SEND são a certificação de roteadores, isto é, um *host* deve ser confiável ao roteador antes que possa defení-lo como roteador padrão, e a autenticação de mensagens via assinatura digital (ARKKO et al., 2014).

O SEND pode ser eficaz na defesa contra os ataques de *DoS* para novos endereço IPv6 e de *flooding* de mensagens *Neighbor Advertisement*. Devido à certificação de roteadores e à autenticação de mensagens via assinatura digital, ele pode ser utilizado também na proteção contra o ataque de *flooding* de mensagens *Router Advertisement*, já que esta funcionalidade poderia evitar a configuração de endereços e rotas ilegítimas nos *hosts*.

Atualmente, existem implementações do SEND para roteadores CISCO (CISCO, 2014) e Juniper (JUNIPER, 2014), além de implementações para Linux, como o NDProtector (AMNESIAK, 2014), Easy-SEND (SOURCEFORGE, 2014) e ipv6-send-cga (GOOGLE, 2014).

Nos experimentos, não foi possível testar estas soluções devido à indisponibilidade de roteadores reais para experimentos e, no caso das implementações para Linux, devido à dificuldades de instalação e pouca documentação ou suporte.

5.7.2 *Router Advertisement Guard* – RA Guard

O RA Guard é uma solução desenvolvida para detectar e bloquear mensagens *Router Advertisement* falsas em redes IPv6 onde não há suporte completo ao SEND e onde todo o tráfego IPv6 passa por dispositivos de camada 2 gerenciáveis, como *switches*, capazes de bloquear as mensagens *Router Advertisement* falsas. O RA Guard pode ser *stateless* ou *statefull*. Uma implementação *stateless* do RA Guard examina os pacotes recebidos e decide se os mesmos serão encaminhados ou descartados. Para decidir se encaminhará ou não as mensagens, o RA Guard *stateless* examina parâmetros como endereço MAC de origem, porta do equipamento onde a mensagem foi recebida, endereço IPv6 de origem e lista de prefixos contida na mensagem. Estes parâmetros são comparados com as configurações do equipamento, onde é definido quais os endereços MAC de origem permitidos, a portas onde podem ser recebidas mensagens *Router Advertisement*, endereços IPv6 e listas de prefixos permitidas. Já o RA Guard *stateful* aprende dinamicamente quais mensagens *Router Advertisement* devem ser permitidas e quais devem ser bloqueadas, armazenando informações sobre as mensagens durante um período de tempo para definir como estas mensagens serão tratadas posteriormente. Há ainda o RA Guard *stateful* baseado em SEND, que aplica as opções e funcionalidades do SEND a mensagens *Router Advertisement* (LEVY-ABEGNOLY et al., 2014).

O RAGuard pode ser eficaz contra os ataques de *DoS* através do anúncio de um roteador falso e de *flooding* de mensagens *Router Advertisement*. Com o RA Guard, seria possível definir políticas para identificar e bloquear as mensagens *Router Advertisement* ilegítimas geradas pela máquina atacante. Foram encontradas implementações do RA Guard para *switches* CISCO (CISCO, 2014) e Brocade (BROCADE, 2014). O RA Guard não pôde ser testado nos experimentos devido ao fato de não haver *switches* disponíveis para experimentos e também ao fato de não ter sido encontrada nenhuma solução para Linux.

5.7.3 *DNS Security* – DNSSEC

O DNSSEC consiste em especificações de segurança para algumas informações fornecidas pelo DNS. O DNSSEC permite a conferência de nomes de domínio e endereços IP correspondentes através de assinatura digital e criptografia de chaves públicas. Para o armazenamento das chaves, o DNSSEC prevê um registro específico chamado de DNSKEY. O uso do DNSSEC prevê que as respostas das consultas DNS contenham, além do registro consultado, um registro de assinatura digital chamado de RRSIG. O RRSIG é uma assinatura digital da resposta a uma consulta DNS, verificada através do registro DNSKEY (ARENDS et al., 2013).

O DNSSEC pode tornar mais seguro o mecanismo DNS64 em determinados casos, se for amplamente utilizado. No entanto, este mecanismo não se mostrou eficaz ara ataques como o realizado no cenário de testes NAT64, devido ao fato de todo o tráfego ser redirecionado à máquina atacante e também devido ao fato do DNSSEC não prover confidencialidade dos dados, permitindo que a máquina atacante ainda tenha acesso a dados desprotegidos capturados.

6 Conclusões e trabalhos futuros

No início deste trabalho foram definidos como objetivos específicos:

- Revisão bibliográfica do protocolo IPv6, focando na relação entre transição IPv4/IPv6 e segurança; (1)
- Realização de testes por meio da infraestrutura implantada e avaliação dos resultados; (2)
- Avaliação da possibilidade de encontrar possíveis soluções de segurança tendo em vista os testes realizados. (3)

Tendo em vista os objetivos definidos, chegou-se às seguintes conclusões:

Entende-se que o primeiro objetivo específico (1) foi atingido totalmente, pois este travalho apresenta, nos capítulos 1, 2 e 3, uma completa revisão bibliográfica acerca do protocolo Ipv6, das principais técnicas de transição existentes e das questões de segurança envolvendo a implantação e transição para o Ipv6.

Entende-se que o segundo objetivo específico (2) foi atingido totalmente, pois neste trabalho foi implantada e mantida em produção uma uma infraestrutura para testes e, nesta infraestrutura, os testesforam realizados com sucesso, gerando os resultados apresentados no Capítulo 5.

Entende-se que o terceiro objetivo específico (3) foi atingido parcialmente. Foram encontradas e testadas formas de detecção dos ataques realizados. Foram, também, levantadas possíveis formas de defesa, no entanto, conforme exposto na Seção 5.7, não foi possível realizar testes com as soluções encontradas. Acredita-se que, para que a avaliação da possibilidade de encontrar possíveis soluções de segurança tendo em vista os testes realizados, as soluções de segurança encontradas devem ser validadas.

6.1 Considerações acerca dos resultados

A transição para o IPv6 avança. Praticamente todos os computadores e equipamentos de rede fabricados atualmente suportam tanto IPv4 quanto IPv6. O maior suporte ao IPv6 e a crescente escassez de endereços IPv4 contribui para que haja cada vez mais redes onde IPv4 e IPv6 coexistem. A preocupação com o período de transição é bastante visível em trabalhos acadêmicos (SANKARAN, 2013) (CAICEDO; JOSHI; TULADHAR, 2009) (TAIB; BUDIARTO, 2007). Não

só continuam a existir antigas ameaças do IPv4, como surgem novas ameaças específicas do IPv6 e dos mecanismos de transição. No entanto, não se vêem muitas experiências práticas que levam em consideração as ameaças específicas do IPv6 e dos mecanismos de transição, testando a eficácia de estratégias de defesa para estas ameaças (SANKARAN, 2013) (TAIB; BUDIARTO, 2007) . O que motivou o desenvolvimento deste trabalho foi abordar de forma prática as questões de segurança no período de transição do IPv4 para o IPv6, buscando, também, fornecer soluções para os problemas de segurança testados.

Os efeitos dos ataques em cada um dos cenários se mostraram bastante parecidos nos cenários de testes. Ataques de *DoS* obtiveram êxito em negar serviço à rede local, nos ataques de *man-in-the-middle*, a máquina atacante interceptou com sucesso o tráfego dos *hosts* da rede local e os ataques de *flooding* sobrecarregaram a rede e os *hosts* da rede local. Deve-se considerar, no entanto que, apesar dos efeitos dos ataques serem semelhantes, suas consequências em cada um dos cenários apresentam uma grande diferença. No cenário pilha-dupla, apesar dos ataques comprometerem criticamente a rede IPv6, ainda há a conectividade via IPv4, que não é afetada. No momento em que se encontra a transição atualmente, estes ataques não causariam grande impacto, uma vez que a maior parte dos *sites* e serviços da Internet ainda operam com IPv4. Mas, à medida que a transição avança e mais servidores passam a operar apenas com IPv6, os ataques realizados podem ser muito mais perigosos, comprometendo completamente a conectividade. No cenário NAT64, os ataques já causam um impacto muito maior, uma vez que a rede local é puramente IPv6, portanto, ataques de *DoS* comprometeriam totalmente a conectividade dos *hosts* com a Internet e ataques de *man-in-the-middle* capturam todo o tráfego dos *hosts*. Os efeitos destes tipos de ataques explorando vulnerabilidades específicas do IPv6 serão de grande impacto tanto no estado atual quanto em estágios mais avançados da transição para o IPv6. No caso de ataques que visam especificamente o mecanismo de transição NAT64 e o DNS64, haverá impactos maiores a curto prazo, enquanto o IPv4 ainda é amplamente utilizado e comprometer a tradução de endereços tem maior efeito sobre a conectividade. No entanto, à medida que a transição avança, os impactos serão cada vez menores, uma vez que cada vez mais serviços e *sites* na Internet passam a operar apenas em IPv6 e a tradução passa a ser menos necessária.

Ressalta-se que, apesar do êxito dos ataques e apesar dos resultados dos experimentos mostrarem que a transição do IPv4 para o IPv6 e, posteriormente, a implantação do IPv6 por completo implicarão em novas vulnerabilidades na rede, em nenhum momento este trabalho conclui que o IPv6 é um protocolo inseguro e que sua adoção deve ser desestimulada. Acredita-se, com base nos resultados deste trabalho e nas formas de defesa pesquisadas, que a maior adoção do IPv6 tende a torná-lo mais seguro, uma vez que deve haver maior preocupação com as vulnerabilidades deste

protocolo e, portanto, as soluções de segurança existentes, ainda recentes e oferecidas por um pequeno número de fabricantes de equipamentos de rede, tendem a ser aprimoradas, assim como a oferta das mesmas tende a aumentar. Os resultados deste trabalho devem servir para alertar sobre as vulnerabilidades inerentes ao IPv6 e às técnicas de transição e estimular, em conjunto com a adoção do IPv6, o desenvolvimento de soluções de segurança prevendo as vulnerabilidades inerentes ao IPv6 e às técnicas de transição.

Entende-se que o primeiro objetivo específico (1) foi atingido totalmente, pois este travalho apresenta, nos capítulos 1, 2 e 3, uma completa revisão bibliográfica acerca do protocolo Ipv6, das principais técnicas de transição existentes e das questões de segurança envolvendo a implantação e transição para o IPv6

6.2 Propostas de trabalhos futuros

A partir dos resultados obtidos com este trabalho, propõe-se como trabalhos futuros:

- **Realização de experimentos com técnicas de tunelamento e a técnica DS-Lite –** estas técnicas de transição não foram contempladas com experimentos práticos neste trabalho. Propõe-se que sejam realizados experimentos contemplando as mesmas, uma vez que estas técnicas ainda são utilizadas em cenários de transição reais.
- **Realização de experimentos com formas de defesa em equipamentos de rede reais –** em função da indisponibilidade, no âmbito deste trabalho, de equipamentos reais para experimentos, como roteadores e *switches*, propõe-se um trabalho que viabilize a realização de experimentos em equipamentos de rede reais e teste estratégias e políticas de segurança em cenários de transição IPv4/IPv6.
- **Realização dos experimentos deste trabalho explorando diferentes opções das ferramentas do THC-IPv6 –** Os utilitários do THC-IPv6 utilizados neste trabalho oferecem diferentes opções além daquelas aqui utilizadas, que podem ser exploradas em novos experimentos em diferentes cenários.
- **Desenvolvimento de uma ferramenta que implemente os mecanismos SEND e RA GUARD para Linux –** Há um pequeno número de implementações do SEND e não foi encontrada nenhuma implementação do RA Guard para Linux. Propõe-se o desenvolvimento de ferramentas para suprir esta demanda.

Referências:

AMNESIAK. **NDProtector**. Disponível em: <http://amnesiak.org/NDprotector/>. Acesso em: 6 jun. 2014.

ARENDS, R. et al. **Resource Records for the DNS Security Extensions.** RFC 4034. Disponível em: <http://www.ietf.org/rfc/rfc4034.txt>. Acesso em: 6 jun. 2014.

ARKKO, J et al. **SEcure Neighbor Discovery (SEND)**. RFC 3971. Disponível em: <http://www.hjp.at/doc/rfc/rfc3971.html>. Acesso em: 6 jun. 2014.

BAGNULO, M et al. **DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers.** RFC 6147 Disponível em: <http://tools.ietf.org/html/rfc6147>. Acesso em: 15 set. 2013.

BAGNULO, M; MATTHEWS, P; BEIJNUM, I Van. **Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers.** RFC 6146. IETF. Disponível em: <http://tools.ietf.org/html/rfc6146>. Acesso em: 15 set. 2013.

BI, Jun; WU, Jianping; LENG, Xiaoxiang. IPv4/IPv6 Transition Technologies and Univer6 Architecture. **International Journal Of Computer Science And Network Security**, Beijing, n. , p.232-243, jan. 2007.

BILSKI, Tomasz. Security-Functionality Tradeoffs in IP Transition Phase. **6th International Conference On Internet Technology And Secured Transactions**, Abu Dhabi, n. , p.632-638, dez. 2011.

BROCADE. **Example of configuring IPv6 RA guard.** Disponível em: <http://www.brocade.com/downloads/documents/html_product_manuals/FI_08000a_SECURITY/wwhelp/wwhimpl/common/html/wwhelp.htm#href=FI_ipv6_ra_guard.15.6.html&single=true>. Acesso em: 6 jun. 2014.

CAICEDO, Carlos; JOSHI, James; TULADHAR, Summit. IPv6 Security Challenges. **Ieee Computer Society,** [s.l.], v. 1, n. 1, p.36-42, fev. 2009.

CISCO (Eua). **IPv6 Secure Neighbor Discovery.** Disponível em: <http://www.cisco.com/en/US/docs/ios-xml/ios/sec_data_acl/configuration/15-2mt/ip6-send.html#GUID-DCB20ADF-1F8E-434B-AE97-54802879F34F>. Acesso em: 6 jun. 2014.

CISCO. **IPv6 RA Guard.** Disponível em: <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6/configuration/15-2s/ip6-15-2s-book/ip6-ra-guard.html>. Acesso em: 6 jun. 2014.

CONTA, A; DEERING, S; GUPTA, M. **Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.** RFC 4443. IETF. Disponível em: <http://tools.ietf.org/html/rfc4443>. Acesso em: 03 set. 2013.

DAS, Kaushik. **IPv6 Header Deconstructed.** Disponível em: <http://www.ipv6.com/articles/general/IPv6-Header.htm>. Acesso em: 02 set. 2013.

DAVIES, E; KRISHNAN, S; SAVOLA, P. **IPv6 Transition/Coexistence Security Considerations.** RFC 2401. IETF. Disponível em: <http://www.ietf.org/rfc/rfc2401.txt>. Acesso em: 14 out. 2013.

DEERING, S; HINDEN, R. **Internet Protocol, Version 6 (IPv6) Specification.** RFC 2460. IETF. Disponível em: <http://xml2rfc.tools.ietf.org/html/rfc2460>. Acesso em: 01 set. 2013.

DOMINGOS, Fabiano. TECNICA DE TRANSIÇÃO ENTRE REDES IPV4/IPV6. **Revista de Ciências Exatas e Tecnologia**, [s. L.], n. , p.1-22, jan. 2006.

DURDAGI, Emre; BULDU, Ali. IPV4/IPV6 security and threat comparisons. **Procedia Social And Behavioral Sciences**, Istanbul, n. , p.5285-5291, 25 jan. 2010.

GOMES, Alexandre; TRINDADE, Carlos. **Rede IP I:Melhores Práticas de Migração de Rede IPv4 para IPv6.** [s. L.]: Teleco, 2012.

GOOGLE. **Ipv6-send-cga.** Disponível em: <http://code.google.com/p/ipv6-send-cga/>. Acesso em: 6 jun. 2014.

HINDEN, R; DEERING, S. **Internet Protocol Version 6 (IPv6) Addressing Architecture.** RFC 3513. IETF. Disponível em: <http://tools.ietf.org/html/rfc3513>. Acesso em: 28 out. 2013.

HUSTON, Geoff **IPv6 Transitional Uncertainties.** Disponível em: <http://www.circleid.com/posts/ipv6_transitional_uncertainties/>. Acesso em: 05 set. 2013.

HP (Usa). **HP TCP/IP Services for OpenVMS Guide to IPv6.** Disponível em: <http://h71000.www7.hp.com/doc/732final/6645/6645pro.html>. Acesso em: 30 out. 2013.

JANBEGLOU, Maziar; ZAMANI, Mazdak; IBRAHIM, Suhaimi. Redirecting Network Traffic toward a Fake DNS Server on a LAN. **3º Ieee International Conference On 3rd Computer Science**

And Information Technology. Chengdu, p. 429-433. jul. 2010.

JUNIPER (Eua). **Example: Configuring Secure IPv6 Neighbor Discovery.** Disponível em: <http://www.juniper.net/techpubs/en_US/junos13.3/topics/topic-map/ipv6-secure-neighbor.html>. Acesso em: 6 jun. 2014.

KENT, S; ATKINSON, R. **Security Architecture for the Internet Protocol.** RFC 2401. IETF. Disponível em: <http://www.ietf.org/rfc/rfc2401.txt>. Acesso em: 13 out. 2013.

LEVY-ABEGNOLI, E. et al. **IPv6 Router Advertisement Guard.** RFC 6105. Disponível em: <http://www.hjp.at/doc/rfc/rfc6105.html>. Acesso em: 6 jun. 2014.

LITECH. **Linux IPv6 Router Advertisement Daemon (radvd).** Disponível em: <http://www.litech.org/radvd/>. Acesso em: 11 jun. 2014.

LITECH. **README for TAYGA v0.9.2.** Disponível em: <http://www.litech.org/tayga/README-0.9.2>. Acesso em: 11 jun. 2014.

MOREIRAS, Antônio et al. **Cabeçalho.** Disponível em: <http://ipv6.br/entenda/cabecalho/>. Acesso em: 01 set. 2013.

MOREIRAS, Antônio et al. **Endereçamento.** Disponível em: <http://ipv6.br/entenda/enderecamento/>. Acesso em: 01 set. 2013.

MOREIRAS, Antônio et al. **Funcionalidades.** Disponível em: <http://ipv6.br/entenda/funcionalidades/>. Acesso em: 01 set. 2013.

MOREIRAS, Antônio et al. **Transição.** Disponível em: <http://ipv6.br/entenda/transicao/>. Acesso em: 01 set. 2013.

NARTEN, T et al. **Neighbor Discovery for IP version 6 (IPv6).** RFC 4861. IETF. Disponível em: <http://tools.ietf.org/html/rfc4861>. Acesso em: 03 set. 2013.

NORDMARK, E; GILLIGAN, R. **Basic Transition Mechanisms for IPv6 Hosts and Routers.** RFC 4213. IETF. Disponível em: <http://tools.ietf.org/html/rfc4213>. Acesso em: 17 set. 2013.

SAAD, Redhwan; RAMADASS, Sureswaran; MANICKAM, Selvakumar. A Study on Detecting ICMPv6 Flooding Attack based on IDS. **Australian Journal Of Basic And Applied Sciences**, Penang, n. , p.175-181, jan. 2013.

SANKARAN, R. Migration of IPv6 - Security Issues. **International Journal Of Computer Trends And Technology**, Chennai, n. , p.567-572, abr. 2013.

SOURCEFORGE. **Easy-SEND.** Disponível em: <http://sourceforge.net/projects/easy-send/>. Acesso em: 6 jun. 2014.

TAIB, Abidah; BUDIARTO, Rahmat. Security Mechanisms for the IPv4 to IPv6 Transition. **The 5 Student Conference On Research And Development**, [s. L.], n. , p.1-6, dez. 2007.

TANENBAUM, Andrew. **Redes de Computadores.** 4. ed. Amsterdam: Campus, 2003.

TCPDUMP. **TCPDUMP & LIBPCAP.** Disponível em: <http://www.tcpdump.org/>. Acesso em: 11 jun. 2014.

TELECO (Brasil). **IPv6: Endereço e Roteamento.** Disponível em: <http://www.teleco.com.br/tutoriais/tutorialipv6/pagina_3.asp>. Acesso em: 03 set. 2013.

THE HACKER'S CHOICE. **THC-IPV6.** Disponível em: <https://www.thc.org/thc-ipv6/>. Acesso em: 11 jun. 2014.

THE HONEYNET PROJECT. **6Guard: a honeypot-based IPv6 attack detector.** Disponível em: <https://www.honeynet.org/node/944>. Acesso em: 11 jun. 2014.

THOMSON, S; NARTEN, T; JINMEI, J. **IPv6 Stateless Address Autoconfiguration.** RFC 4862. IETF. Disponível em: <http://tools.ietf.org/html/rfc4862.html>. Acesso em: 01 nov. 2013.

VMWARE. **VSphere ESXi Hypervisor.** Disponível em: <http://www.vmware.com/products/vsphere/features/esxi-hypervisor.html>. Acesso em: 11 jun. 2014.

WIRESHARK. **About Wireshark.** Disponível em: <http://www.wireshark.org/about.html>. Acesso em: 11 jun. 2014.

Anexo 1 – Artigo: Avaliação dos Aspectos de Segurança em Um Cenário de Transição IPv4/IPv6

Abstract. *The depletion of the available Ipv4 addresses is approaching, wich increases the urgency of the migration to Ipv6. However, Ipv4 and Ipv6 are not compatible, therefore the transition must take place gradually. For that reason, transition techniques were created. Ipv4/Ipv6 transition did not occur as expected and it's urgency brings up a critical factor: security. The objective of this work is to evaluate the security aspects em Ipv4/Ipv6 transition techniques, contributing to the identification and minimizantion of threats and to a safer transition. The results show that the tested transition scenarios are susceptible to attacks, but there are ways to detect this attacks and there are possible ways to deffend against them.*

Resumo. *O esgotamento dos endereços IPv4 disponíveis está cada vez mais próximo e, com isto, cresce a urgência da migração para o Ipv6. No entanto, Ipv4 e Ipv6 não são compatíveis entre sí e, portanto, a transição deve ser feita gradativamente. Por isto, foram criadas técnicas de transição. A transição Ipv4/Ipv6 não ocorreu conforme o esperado e a urgência desta transição traz à tona um fator crucial: a segurança. Este artigo tem como objetivo avaliar os aspectos de segurança em técnicas de transição IPv4/IPv6, contribuindo para a identificação e minimização de ameaças e uma transição mais segura. Os resultados mostram que os cenários de transição testados estão suscetíveis a ataques, mas há formas de detecção e possíveis formas de defesa.*

1. Introdução

O IPv6 é a nova versão do Protocolo Internet (*Internet Protocol* – IP) e foi projetado para substituir a versão 4 deste protocolo (IPv4) tendo em vista o esgotamento do espaço de endereçamento do mesmo. Entre as principais mudanças do Ipv4 para o Ipv6, destaca-se o maior espaço de endereçamento, simplificação do cabeçalho, maior suporte a extensões e controle de fluxo, incluindo suporte a extensões de segurança (DEERING; HINDEN, 2013). O esgotamento de endereços IPv4 está cada vez mais próximo, o que aumenta a urgência da transição para o IPv6. No entanto, IPv4 e IPv6 não são compatíveis entre sí e, portanto, não é possível realizar a transição apenas desativando o IPv4 e passando a usar somente IPv6. É necessário que a transição seja feita gradativamente, o que significa que IPv4 e IPv6 coexistirão por um tempo, até que a transição esteja completa. Tendo em vista a necessidade de coexistência entre IPv4 e IPv6 durante este período de transição, foram desenvolvidas técnicas de transição (MOREIRAS et al., 2013).

Um problema encontrado na implantação do IPv6 e na transição IPv4/IPv6 é a questão de segurança. Existem vulnerabilidades associadas específicamente ao IPv6, que vão sendo reveladas à medida que a adoção deste protocolo aumenta. Estas vulnerabilidades podem afetar redes IPv6 mesmo no período de transição. Além disto, as próprias técnicas de transição podem ser exploradas por atacantes. O desafio que se encontra é fazer com que a transição do IPv4 para o IPv6 seja feita de forma segura, prevendo ameaças e vulnerabilidades desde os estágios da transição e considerando o uso de diferentes técnicas. Tendo em vista o fato de vulnerabilidades serem reveladas à medida que o uso do IPv6 aumenta, a realização de testes em cenários de transição pode contribuir para elucidação de problemas de segurança na transição IPv4/IPv6, assim como pode ajudar a encontrar soluções ou aprimorar soluções já existentes.

Este artigo propões a elaboração de cenários de testes utilizando duas das principais técnicas de transição IPv4/IPv6 existentes: pilha dupla e NAT64. Propõe-se também que sejam feitos testes nestes cenários através da realização de ataques explorando vulnerabilidades do IPv6 e das técnicas de transição. Além disto, propõe-se o estudo de possíveis formas de defesa, tendo em vista os cenários de teste, os ataques realizados e os efeitos de cada ataque nos cenários.

Atualmente, há pesquisas acadêmicas e recomendações técnicas abordando questões de segurança específicas do IPv6 (SANKARAN, 2013) (DAVIES; KRISHNAN; SAVOLA, 2013) (DURDAGI; BULDU, 2010). Há também pesquisas acadêmicas e recomendações técnicas abordando questões de segurança referentes às técnicas de transição (TAIB; BUDIARTO, 2007) (BILSKI, 2011) (BI; WU; LENG, 2007) (BAGNULO et al., 2013).

O texto do artigo está organizado em oito seções. A Seção 2 apresenta os principais trabalhos relacionados. A Seção 3 apresenta os conceitos básicos e principais funcionalidades do IPv6. A Seção 4 apresenta algumas das principais técnicas de transição IPv4/IPv6. A Seção 5 apresenta as principais ameaças e vulnerabilidades relativas ao IPv6 e às técnicas de transição apresentadas na Seção 4. A Seção 6 descreve os cenários de testes e experimentos realizados. A Seção 7 apresenta os resultados dos testes, bem como formas de detecção dos ataques e possíveis formas de defesa. A última seção apresenta as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

O trabalho de (SANKARAN, 2013) tem como objetivo uma revisão dos aprimoramentos do IPv6 em relação ao IPv4, focando nas funcionalidades do IPv6 relacionadas à segurança. O trabalho de (SANKARAN, 2013) é estudo teórico que aponta os principais aprimoramentos do IPv6 em relação ao IPv4, no que diz respeito à segurança, apontando também alguns problemas de segurança relacionados ao IPv6 e à transição IPv4/IPv6. O trabalho de (SANKARAN, 2013) conclui que, apesar do IPv6 apresentar aprimoramentos em funcionalidades de segurança e resolver antigos problemas encontrados no IPv4, novos problemas surgem com a nova versão do protocolo. Além disto, (SANKARAN, 2013) conclui também que a própria transição pode trazer uma série de novos desafios com relação à segurança. O trabalho de (SANKARAN, 2013) se assemelha a este trabalho por tratar de questões de segurança do IPv6 e da transição IPv4/IPv6. A principal diferença está no fato deste trabalho apresentar, além do estudo teórico, experimentos em laboratório.

No trabalho de (DURDAGI; BULDU, 2010), o objetivo é analisar e comparar ameaças existentes no IPv4 e no IPv6. O trabalho de (DURDAGI; BULDU, 2010) é um estudo teórico que aborda questões de segurança similares no IPv4 e no IPv6 e também questões de segurança específicas do IPv6. O trabalho de (DURDAGI; BULDU, 2010) conclui que apesar dos aprimeramentos do IPv6 em relação ao IPv4, certas vulnerabilidades já conhecidas no IPv4 continuam existindo no IPv6 e novas vulnerabilidades surgem com a implantação do IPv6. O trabalho de (DURDAGI; BULDU, 2010) conclui também que a solução dos novos problemas de seurança encontrados no IPv6 contribuirão para uma maior aceitação do protocolo. O trabalho de (DURDAGI; BULDU, 2010) se assemelha a este trabalho por estudar questões de segurança relacionadas à implantação do IPv6. As principais diferenças encontradas são o foco deste trabalho apresentar um foco maior nas técnicas de transição IPv4/IPv6 e apresentar também experimentos em laboratórios, ausentes no trabalho de (DURDAGI; BULDU, 2010).

O trabalho de (TAIB; BUDIARTO, 2007) tem como objetivo classificar potenciais problemas de segurança no período de transição IPv4/IPv6 e identificar mecanismos de prevenção a ataques. No trabalho de (TAIB; BUDIARTO, 2007) são apontados possíveis ameaças e vulnerabilidades devido a implantação do IPv6 e relacionadas às tecnicas de transição. O trabalho de (TAIB; BUDIARTO, 2007) aponta também possíveis formas de solução. O trabalho de (TAIB; BUDIARTO, 2007) conclui que a transição do IPv4 para o IPv6 demanda diversas considerações a respeito da segurança e destaca as contribuições do trabalho para estas considerações. O trabalho de (TAIB; BUDIARTO, 2007) se assemelha a este trabalho por elencar problemas de segurança relativos à implantação do IPv6 e focar no período de transição. A principal diferença do trabalho de (TAIB; BUDIARTO, 2007) para este trabalho é o fato deste trabalho apresentar experimentos em laboratório.

O trabalho de (BI; WU; LENG, 2007) tem como objetivo propor uma arquitetura para um estágio futuro da transição IPv4/IPv6, chamada de univer6. O trabalho de (BI; WU; LENG, 2007) sumariza e compara técnicas e cenários de transição, abordando também aspectos de segurança, e apresenta a proposta da arquitetura univer6. O trabalho de (BI; WU; LENG, 2007) conclui apontando uma série de desafios na transição IPv4/IPv6, sendo um dos principais pontos levantados a segurança. O trabalho de (BI; WU; LENG, 2007) se assemelha a este trabalho por abordar questões de segurança na transição IPv4/IPv6. A principal diferença está no fato do trabalho de (BI; WU; LENG, 2007) focar em propor uma solução para a transição IPv4/IPv6, enquanto este trabalho foca em questões de segurança.

3. O protocolo IPv6

O Protocolo Internet versão 6 (*Internet Protocol version 6* – IPv6) foi desenvolvido com o objetivo de substituir o IPv4. Entre as principais motivações para o desenvolvimento do IPv6 está a crescente demanda por endereços IP em função de um grande aumento no número de dispositivos pessoais capazes de acessar a Internet disponíveis no mercado. Esta demanda por endereços IP não era prevista quando o IPv4 foi projetado, o que tem como consequência uma iminente escassez de endereços IP (TANENBAUM, 2003) (MOREIRAS et al., 2013).

Do ponto de vista da arquitetura TCP/IP, o IPv6 é, conceitualmente, bastante semelhante ao IPv4. No entanto, foram realizadas mudanças em sua estrutura para proporcinar aprimoramentos no protocolo. Primeiramente, o IPv6 possui um espaço de endereçamento de 128 bits, contra 32 bits do IPv4. Este aumento no espaço de endereçamento, além de possibilitar a conexão de um número muito maior de dispositivos na rede, permite níveis mais específicos de agregação de endereços e a implementação de mecanismos de autoconfiguração, que possibilitam a obtenção de endereços IPv6 globais automaticamente sem o uso de *Dynamic Host Configuration Protocol* (DHCP) (MOREIRAS et al., 2013). Os endereços IPv6 são representados por 8 grupos de 16 bits escritos em até 4 dígitos hexadecimais e separados por ":" (dois pontos), Como no exemplo mostrado em (1). Há mecanismos de abreviação que permitem uma representação simplificada de endereços IPv6. É possível omitir zeros à esquerda nos grupos de 4 dígitos hexadecimais e uma grande sequência de zeros pode ser substituída pela notação “::”, como no exemplo em (2).

- 2001:0DB8:0000:0000:0008:0800:200C:417A (1)
- 2001:DB8::8:800:200C:417A (2)

Foram definidos três tipos de endereços no IPv6:

- **Endereços unicast –** identificam uma única interface, de modo que pacotes enviados a endereços unicast são entregues a somente uma interface.
- **Endereços anycast –** identificam um conjunto de interfaces, de modo que um pacote enviado a um endereço anycast é entregue à interface pertencente a este conjunto mais próxima da origem.
- **Endereços multicast –** identificam um conjunto de interfaces, sendo que, neste caso, um pacote entregue a um endereço multicast é entregue a todas as interfaces pertencentes ao grupo (MOREIRAS et al., 2013).

A simplificação no formato do cabeçalho é outro aprimoramento do IPv6 em relação ao IPv4. Alguns campos do cabeçalho IPv4 foram removidos ou tornados opcionais. Desta forma, é possível reduzir o custo de processamento dos pacotes pelos roteadores (MOREIRAS et al., 2013) e, portanto, melhorar a vazão em suas interfaces de rede (TANEMBAUM, 2003).

Além da simplificação do cabeçalho, o IPv6 suporta cabeçalhos de extensão. Os cabeçalhos de extensão comportam as opções que não fazem mais parte do cabeçalho base. Estes cabeçalhos localizam-se entre o cabeçalho base e o cabeçalho da camada de transporte e não possuem quantidade ou tamanho fixo. Caso existam múltiplos cabeçalhos de extensão em um mesmo pacote, estes cabeçalhos são adicionados em série, formando uma cadeia de cabeçalhos. Com cabeçalhos de extensão, o roteamento se torna mais eficaz, há limitações menos rigorosas para o tamanho e a quantidade de opções e há também uma maior flexibilidade para a introdução de novas opções no futuro (MOREIRAS et al., 2013).

Dando maior atenção à qualidade de serviço, o IPv6 permite a identificação de um fluxo de dados. Desta forma, é possível determinar se pacotes pertencentes a um determinado fluxo devem

ser tratados de forma diferenciada (MOREIRAS et al., 2013). Como exemplo de tráfego diferenciado, pode-se citar aplicações de tempo real (DEERING; HINDEN, 2013).

Com relação à segurança, foram especificados cabeçalhos de extensão capazes de fornecer mecanismos de autenticação, além de garantir a integridade e confidencialidade dos dados transmitidos (MOREIRAS et al.2013).

3.1. ICMPv6

O ICMPv6 é a nova versão do ICMPv4. Esta nova versão incorpora as funcionalidades do ICMPv4 e algumas mudanças. Algumas das funções mais básicas do IPv6 estão associadas ao ICMPv6, portanto, este protocolo deve ser implementado por todos os nós IPv6 da Internet (CONTA; DEERING; GUPTA, 2013).

Além das funções já desempenhadas pelo ICMPv4, o ICMPv6 desempenha uma série de novas funções. Uma das mudanças do ICMPv6 em relação ao ICMPv4 é que o ICMpv6 passa a desempenhar funções dos protocolos ARP, RARP e IGMP que, por sua vez, deixam de existir com o uso do IPv6. Outra importante diferença é o fato das mensagens ICMPv6 serem utilizadas nas principais funcionalidades do IPv6. Dentre estas funcionalidades, pode-se destacar o *Neighbor Discovery Protocol* (NDP) e a autoconfiguração de endereços (MOREIRAS et al., 2013).

3.2. NDP

O NDP foi definido para resolver problemas relacionados à comunicação entre nós vizinhos em uma rede. Para isto, o NDP atua sobre dois aspectos fundamentais do IPv6: autoconfiguração de nós e transmissão de pacotes. No caso da autoconfiguração de endereços, o NDP fornece suporte para as seguintes funcionalidades:

- **Parameter Discovery** – Como um nó descobre informações sobre o enlace (como a MTU) ou sobre a Internet (como o limite de saltos (*hop limit*));
- **Address Autoconfiguration** – Mecanismo para viabilizar a autoconfiguração *stateless.*
- **Duplicate Address Detection** – Maneira como um nó descobre se o endereço que deseja atribuir a uma interface já está sendo usado por outro nó.

No caso da transmissão de pacotes, o NDP fornece suporte às seguintes funcionalidades:

- **Router Discovery** – Permite aos *hosts* a descoberta de roteadores na rede local, com a finalidade de determinar rotas padrão.
- **Prefix Discovery** – Descoberta de prefixos de rede do enlace com a finalidade de decidir para onde os pacotes serão enviados (i.e. Para um roteador ou diretamente para um nó da rede).
- **Address Resolution** – Determinação de um endereço físico através de um endereço lógico IPv6. Este processo é executado apenas em endereços IP da rede local para os quais o endereço físico ainda não é conhecido.
- **Neighbor Unreachability Detection** – Determina se um nó vizinho continua ou não alcançável. Esta é usada para todos os caminhos entre um *host* e nós vizinhos (sejam estes *hosts* ou roteadores). O procedimento para determinar um caminho alternativo para um destino depende do nó destino, ou seja, se este nó é o próprio destino, a resuloção de endereços (*Address Resolution*) deve ser realizada novamente. Se o nó destino for um

roteador, é necessário que a rota padrão seja alterada para outro roteador.

- **Redirect –** Permite a um roteador informar a um nó sobre uma melhor rota para um determinado destino.
- **Next-Hop Determination** – Algoritmo para mapear o endereço IP de um destino em um endereço IP de um nó vizinho para onde o tráfego deve ser enviado. Este vizinho pode ser um roteador ou o próprio destino.

Para as funções do NDP, foram reservadas cinco tipos de mensagens ICMPv6:

- **Router Solicitation** – Pode ser enviada cada vez que uma interface é habilitada para solicitar que roteadores anunciem sua presença na rede;
- **Router Advertisement** – Enviada por roteadores periodicamente ou em resposta à mensagem Router Solicitation para anunciar sua presença na rede. Junto com esta mensagem, são enviadas informações como prefixos de rede, configurações de endereço, valor sugerido de *hop limit*, entre outras;
- **Neighbor Solicitation** – Enviada por um nó da rede para determinar o endereço físico de um nó vizinho ou para verificar se um nó vizinho está alcançável;
- **Neighbor Advertisement** – Enviada em resposta à Neighbor Solicitation ou para anunciar a mudança no endereço físico de um nó;
- **Redirect –** Utilizada por roteadores para redirecionar um *host* a uma melhor rota para um determinado destino ou para informar ao *host* que o destino é um nó vizinho (NARTEN et al., 2013).

3.3. Autoconfiguração de endereços

A autoconfiguração de endereços é um mecanismo que permite a uma interface obter um endereço IPv6 global automaticamente. A autoconfiguração pode ser *stateless*, isto é, sem guardar estado, ou *statefull*, ou seja, guardando estado. Apenas interfaces com suporte a multicast podem obter endereços através de autoconfiguração. O processo de autoconfiguração tem início quando uma interface com suporte a multicast é habilitada.

O primeiro passo da autoconfiguração é a atribuição de um endereço link-local à interface. Entretanto, antes da atribuição do endereço link-local, é necessário verificar se o endereço que se pretende atribuir à interface não está sendo usado por outra interface na rede. Para isto, é utilizada a funcionalidade *Duplicate Address Detection,* do NDP. O funcionamento deste mecanismo se dá através da troca de mensagens *Neighbor Solicitation* e *Neighbor Advertisement,* onde o nó que pretende atribuir um endereço link-local a uma de suas interfaces envia a todos os nós da rede uma mensagem *Neighbor Solicitation* contendo o endereço a ser atribuído. Se um nó estiver utilizando este endereço, ele enviará em resposta uma mensagem *Neighbor Advertisement* informando este fato e o processo de autoconfiguração para, sendo requerida configuração manual. Uma vez que um nó obtém com sucesso um endereço link-local único, ele já possui conectividade IP com nós vizinhos.

A etapa seguinte envolve troca de mensagens *Router Solicitation* e *Router Advertisement.* Roteadores enviam mensagens *Router Advertisement* periodicamente, anunciando sua presença na rede. No entanto, para determinar a presença ou não de roteadores na rede com maior rapidez, o nó envia uma mensagem *Router Solicitation* para o grupo multicast *all-routers*. Os roteadores, assim que recebem esta mensagem, enviam uma mensagem *Router Advertisement* em resposta. As mensagens *Router Advertisement* contém zero ou mais campos chamados *Prefix Information.* No caso da autoconfiguração *stateless,* estes campos têm a função de fornecer informações necessárias

para a configuração de um endereço global em uma interface. Neste caso, há campos *Prefix Information* contendo um prefixo global de sub-rede e o tempo pelo qual endereços criados a partir deste prefixo permanecerão válidos. O prefixo de sub-rede obtido desta maneira e o identificador da interface formarão o endereço IPv6 global (THOMSON; NARTEN; JINMEI, 2013).

4. Transição IPv4/IPv6

O IPv6 e o IPv4 não são compatíveis entre sí. No entanto, ambas as versões do protocolo IP podem funcionar simultaneamente nos equipamentos. Com base nisto, pensou-se em fazer a transição do IPv4 para o IPv6 de forma gradual, onde IPv6 e IPv4 coexistirião durante um determinado período nos equipamentos e, posteriormente, o IPv4 seria abandonado paulatinamente. Esta coexistência entre IPv6 e IPv4 nos equipamentos é chamado de pilha dupla, a mais básica técnica de transição IPv4/IPv6. Além da pilha dupla, seriam necessárias técnicas auxiliares para interconectar, inicialmente, redes IPv6 a uma Internet majoritariamente IPv4 e, posteriormente, para fazer o contrário. No entanto, a transição IPv4/IPv6 está ocorrendo muito mais lentamente do que o esperado e isto demandou a criação de novas técnicas de transição (MOREIRAS et al., 2013). As seções 4.1 e 4.2 descrevem duas das principais técnicas de transição IPv4/IPv6: pilha dupla e NAT64. A técnica de pilha dupla foi escolhida para este trabalho por ser a mais básica forma de transição IPv4/IPv6 e o NAT64, além de sua importância, foi escolhido por ser uma técnica de mais fácil implantação na infraestrutura disponível para a realização de testes.

4.1. Pilha dupla

A técnica de transição pilha dupla é a forma mais básica de transição IPv4/IPv6, esta técnica consiste em manter o IPv4 funcionando de forma estável e, ao mesmo tempo, implantar o IPv6 nativamente nos equipamentos. Desta forma, os dispositivos são capazes de enviar e receber pacotes de ambas as versões do protocolo IP, isto significa que este dispositivo se comportará como um nó IPv4 quando se comunicar com outro nó IPv4 e como um nó IPv6 quando se comunicar com outro nó IPv6 (MOREIRAS et al., 2013).

4.2. NAT64

As técnicas de tradução foram desenvolvidas para viabilizar a comunicação entre *hosts* puramente IPv4 (ou seja, que não possuem suporte ao IPv6) e *hosts* puramente IPv6 (ou seja, que não possuem suporte ao IPv4).Estas técnicas traduzem endereços e cabeçalhos IPv6 para IPv4 e vice-versa (DOMINGOS, 2006).

Uma das principais técnicas de tradução é o NAT64. O NAT64 é configurado em um equipamento que possui pelo menos duas interfaces de rede, sendo uma das interfaces conectada a uma rede IPv6 e a outra, a uma rede IPv4. Esta técnica trabalha com dois *pools* de endereços, um *pool* IPv4 e um *pool* IPv6. Em função da grande quantidade de endereços IPv6 disponíveis, é possível mapear cada endereço IPv4 em um endereço IPv6 diferente simplesmente concatenando um prefixo IPv6 a um endereço IPv4. Já para o caso do IPv4, em função da escassez de endereços, o mapeamento utilizando o pool de endereços IPv4 geralmente é feito dinamicamente. Também em função da escassez de endereços IPv4, é comum realizar o mapeamento de endereço e porta TCP entre as duas versões do protocolo IP. Em função da natureza dinâmica do mapeamento de IPv6 para IPv4, é mais fácil permitir que a conexão inicie na rede IPv6. Portanto, tradução no NAT64 é feita através do mapeamento de um endereço IPv6 de origem em um endereço IPv4 de destino, juntamente com uma porta TCP. Através de um mecanismo como o NAT64, o cliente IPv6 obtém um endereço IPv6 contendo o endereço IPv4 do servidor encapsulado e envia o pacote a este endereço IPv6. O pacote é, então, interceptado pelo dispositivo que implementa o NAT64 que, por sua vez, associa ao pacote um endereço IPv4 de seu pool de endereços (BAGNULO; MATTHEWS; BEIJNUM, 2013).

Para a conversão do DNS, o NAT64 opera em conjunto com uma técnica chamada de DNS64. O DNS64 tem como função sintetizar um registro AAAA (quad-A) a partir de um registro A original. O nome de dono do registro AAAA sintético criado é o mesmo do registro A original. Porém, o registro AAAA possui um endereço IPv6 ao invés de um endereço IPv4, sendo que, neste caso, o endereço IPv6 deste registro é uma representação do endereço IPv4 contido no registro A original. Combinado com um mecanismo de tradução, como o NAT64, o DNS64 permite que um host IPv6 puro inicie uma conexão com um host IPv4 puro através de um nome de domínio qualificado (BAGNULO et al., 2013).

5. Segurança no IPv6

Ainda que o IPv6 seja um protocolo simplificado e aprimorado em relação ao IPv4, ele apresenta uma série de desafios em relação à segurança. Muitas ameaças presentes em redes IPv4 afetam também redes IPv6 e, com a maior utilização do IPv6, novas ameaças que atacam especificamente este protocolo vão sendo reveladas (CAICEDO; JOSHI; TULADHAR, 2009). Além disto, existem ataques que exploram a coexistência entre IPv4 e IPv6, o que significa que a transição pode implicar em vulnerabilidades na rede (MOREIRAS et al., 2013). Um aprimoramento na segurança do IPv6 foi o espaço de endereçamento que, por ter 96 bits a mais que o espaço de endereçamento do IPv4, dificulta a varredura na rede, o que no IPv4 é relativamente simples. Além disto, foi definido inicialmente que a inclusão do IPsec seria obrigatória em toda implementação do IPv6. No entanto, a inclusão do IPsec deixou de ser obrigatória posteriormente (SANKARAN, 2013).

5.1. Ameaças semelhantes em redes IPv6 e redes IPv4

Apesar das significativas mudanças que o IPv6 apresenta em relação ao IPv4, há ataques conhecidos em redes IPv4 que podem afetar redes IPv6 de forma bastante similar. Ataques que visam a camada de aplicação e *sniffers* afetam redes IPv6 exatamente da mesma forma que afetavam redes IPv4. Ataques como negação de serviço (*Denial of Service* – DoS) e *man-in-the-middle* também afetam redes IPv6, uma vez que o princípio destes ataques continua o mesmo. No entanto, em redes IPv6 há novas forma de executar estes ataques (DURDAGI; BULDU, 2010).

5.2. Questões de segurança específicas do IPv6

Com a especificação do IPv6, surgiram ataques que exploram particularidades do IPv6. Estes ataques estão relacionados principalmente a funcionalidades específicas do IPv6 como NDP, autoconfiguração de endereços e novas mensagens especificadas no ICMPv6 (SANKARAN, 2013).

Um dos principais ataques relacionados ao NDP é o ataque de DoS explorando a funcionalidade *Duplicate Address Detection*. Este ataque consiste em enviar uma mensagem *Neighbor Advertisement* em resposta a cada mensagem *Neighbor Solicitation* recebida. Isto fará com que os endereços de tentativa sejam sempre considerados em uso, impedindo que novos dispositivos obtenham endereços IP válidos e se conectem à Internet. Outro ataque envolvendo o NDP, além de afetar a autoconfiguração de endereços, é a falsificação de mensagens *Router Advertisement*, onde um dispositivo que não é um roteador envia esta mensagem possivelmente com a finalidade de se tornar roteador e interceptar o tráfego ou realizar um ataque de DoS através do anúncio de um roteador falso, criando um local para onde todo o tráfego é desviado (MOREIRAS et al., 2013).

Além dos ataques envolvendo o NDP e a autoconfiguração de endereços, mensagens ICMPv6 específicas como *Neighbor Advertisement* e *Router Advertisement* podem ser utilizadas para ataques de *flooding*. Este ataque segue o mesmo princípio aplicado em redes IPv4, porém as novas mensagens especificadas e a dependência de endereços multicast possibilitam diferentes formas de realizar este ataque em redes IPv6 (SAAD et al., 2013).

5.3. Questões de segurança em cenários de transição pilha dupla

Dispositivos operando com pilha dupla terão tanto endereços IPv4 quanto endereços IPv6 configurados em suas interfaces. Uma das implicações do uso desta técnica é que ataques podem

atingir o dispositivo tanto via IPv4 quanto via IPv6, inclusive ataques que exploram vulnerabilidades específicas do IPv6. Portanto, mecanismos de controle como *firewalls*, clientes VPN e sistemas de detecção de intrusões devem ser capazes de analisar tanto tráfego IPv4 quanto IPv6 e, quando necessário, bloquear tráfego de cada versão do IP especificamente. Em um cenário de pilha dupla, é recomendável que as configurações de *firewall* sejam adaptadas de forma a suportar também o IPv6 e conter um conjunto de regras específico para esta versão do IP. Alternativamente, pode-se configurar um *firewall* específico para o IPv6, separado do *firewall* IPv4 (TAIB; BUDIARTO, 2007).

5.4. Questões de segurança em cenários de transição NAT64

No caso da técnica de tradução NAT64, que opera em conjunto com o DNS64, é preciso considerar ataques ao DNS, uma vez que mecanismo DNS64 está sucetível aos mesmos ataques que o DNS (BAGNULO et al., 2013). Um dos principais ataques visando o DNS é o redirecionamento de requisições DNS feito através de um servidor DNS falso na rede local. Quando o atacante consegue anunciar um servidor DNS falso em uma rede local, este atacante pode responder a resuisições associando um nome a um edereço IP não correspondente e levando hosts da rede local a acessarem um outro servidor, que pode ser utilizado para obter credenciais de acesso, por exemplo. Além disto, a associação de nomes a endereços IP não correspondentes pode acarretar em um ataque de DoS caso o endereço IP anunciado pelo servidor DNS falso não exista (JANBEGLOU; ZAMANI; IBRAHIM, 2010). Para atacar o DNS64 o atacante pode, além disto, alterar o prefixo utilizado para a tradução no NAT64, uma vez que este prefixo deve ser o mesmo utilizado pelo DNS64. A alteração do prefixo utilizado pelo NAT64 pode resultar em ataques de DoS, flooding e pode também possibilitar a captura de pacotes e vizualização de informações contidas nos mesmos pelo atacante (BAGNULO et al., 2013).

6. Cenários de testes e experimentos

Para que se possa fazer uma avaliação das questões de segurança relativas ao IPv6 e às técnicas de transição, visualizando as ameaças em experimentos práticos e estudando possíveis formas de defesa, foram definidos dois cenários de testes. O primeiro deles funciona através da técnica de transição pilha dupla, isto é, IPv4 e IPv6 funcionam simultaneamente em todos os dispositivos da rede. Este cenário é detalhado na Seção 6.1. Os experimentos realizados neste cenário são detalhados na Seção 6.2.

O segundo cenário de testes funciona através da técnica de tradução NAT64, isto é, o cenário é formado por uma rede puramente IPv6 que se comunica com a Internet IPv4 através de NAT64. Este cenário é detalhado na Seção 6.3. Os experimentos realizados neste cenário são detalhados na Seção 6.4.

Para os cenários de testes, foi utilizada a infraestrutura física do Ponto de Presença (*Point of Presence*) da Rede Nacional de Ensino e Pesquisa em Santa Catarina (PoP-SC/RNP). Os cenários são constituídos de máquinas virtuais rodando o sistema operacional Linux, distribuição Debian 7.0 64 bits. As máquinas virtuais estão alocadas em um servidor de virtualização do PoP-SC/RNP, utilizando como ambiente de virtualização o VMware vSphere ESXi Hypervisor. Algumas destas máquinas atuam como roteadores devido à indisponibilidade de roteadores reais para a realização dos experimentos, para isto, foi habilitada a função de roteamento do Kernel Linux, permitindo que as máquinas virtuais Linux encaminhassem pacotes. Cada um destes cenários possui uma máquina atacante. Para a realização dos ataques, foi utilizada a ferramenta *The Hacker's Choice-IPv6* (THC-IPv6), que consiste em uma série de utilitários para a realização de testes, ataques e exploração de vulnerabilidades dos protocolos IPv6 e ICMPv6. A Tabela 1 sumariza os experimentos realizados. Na tabela, são mostrados os ataques realizados, ferramentas do THC-IPv6 utilizadas, cenários afetados e um comparativo com o IPv4.

Ataque	**Utilitário THC-IPv6**	**Cenário**	**Comparativo com IPv4**
DoS para novos endereços IPv6	dos_new_ip6	Pilha dupla e NAT64	Este ataque é específico do IPv6
Anúncio de um roteador falso	fake_router26	Pilha dupla e NAT64	Ataque possível, porém, no Ipv6, pode explorar funcionalidade específica
Flooding de mensagens *Neighbor Advertisement*	flood_advertise6	Pilha dupla e NAT64	Este ataque é específico do IPv6
Flooding de mensagens *Router Advertisement*	flood_router26	Pilha dupla e NAT64	Este ataque é específico do IPv6
Servidor DNS falso	parasite6 fake_dns6d	NAT64	Ataque possível, porém no IPv6 pode atingir DNS64

Tabela 1: Experimentos realizados

6.1. Cenário de testes pilha dupla

O cenário de testes pilha dupla foi elaborado com o objetivo de explorar vulnerabilidades específicas do IPv6, que afetam diretamente uma rede IPv4/IPv6 com pilha dupla. Este cenário é composto por um roteador, um firewall, um servidor web e três hosts, sendo que um deles atua como atacante na rede local. O roteador possui três interfaces de rede, sendo duas interfaces Wide Area Network (WAN), eth0 e eth1, e uma interface Local Area Network (LAN). A interface eth0 está diretamente conectada ao roteador do PoP-SC via IPv4 e a interface eth1, diretamente conectada via IPv6. A interface eth2 está diretamente conectada ao firewall, tanto via IPv4 quanto via IPv6.

O firewall possui, também, três interfaces de rede. A interface eth0 atua como interface WAN, diretamente conectada ao roteador do cenário. A interface eth1 está conectada ao servidor web via IPv4 e IPv6. O servidor possui endereços IP estáticos. A interface eth2 é o gateway para os hosts. Os endereços IPv4 são distribuídos aos hosts via DHCPv4 e os endereços IPv6 são obtidos via autoconfiguração de endereços stateless. Para a divulgação das informações de roteamento através de mensagens Router Avertisement, é utilizada a ferramenta Router Advertisement Daemon (RADVD). A máquina atacante está conectada à rede local junto aos hosts e obtém endereços IPv4 e IPv6 dinamicamente, assim como os demais hosts. A Figura 1 mostra um diagrama deste cenário.

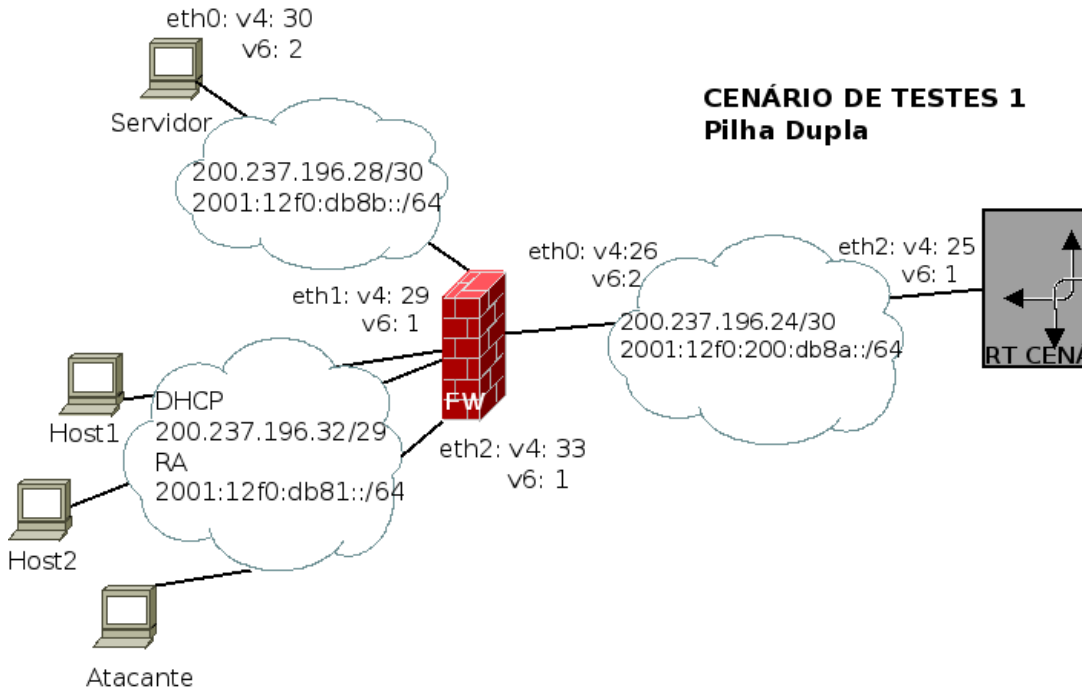

Figura 1: Cenário de testes pilha dupla.

6.2. Experimentos realizados no cenário de testes pilha dupla

Nos experimentos feitos neste cenário, foram realizados os seguintes ataques:

- **DoS para novos endereços IPv6 –** atacante explora a funcionalidade *Duplicate Address Detection* para impedir que *hosts* obtenham endereços IPv6;
- **Anúncio de um roteador falso –** atacante anuncia a sí mesmo como roteador;
- ***Flooding* de mensagens *Neighbor Advertisement*** – atacante dispara grande número de mensagens *Neighbor Advertisement*;
- ***Flooding* de mensagens *Router Advertisement*** – atacante dispara grande número de mensagens *Router Advertisement*.

6.2.1. DoS para novos endereços IPv6

O ataque realizado neste primeiro experimento consiste em um ataque de DoS onde a máquina atacante explora a funcionalidade Duplicate Address Detection, enviando uma mensagem Neighbor Advertisement para cada mensagem Neighbor Solicitaiton recebida e fazendo com que dispositivos que se conectam à rede não consigam obter endereços IPv6.

Para este ataque, foi utilizado o utilitário dos_new_ip6, parte do THC-IPV6. Ao ser executado via CLI, o dos_new_ip6 deve receber como parâmetro uma interface de rede. O dos_new_ip6 passará a responder a todas as mensagens Neighbor Solicitation recebidas pela interface especificada enviando uma mensagem Neighbor Advertisement. Este experimento envolveu a máquina atacante, executando o dos_new_ip6, e dois hosts, tentando obter endereços IPv6.

6.2.2. Anúncio de um roteador falso

Neste segundo experimento, a máquina atacante envia mensagens Router Advertisement para todos os nós da rede local, anunciando a si mesma como roteador. Este ataque pode possibilitar à maquina atacante capturar o tráfego da rede local ou pode acarretar em um ataque de DoS, onde todo o tráfego é direcionado para um dispositivo que não é roteador e, portanto, não é encaminhado.

Para este ataque, foi utilizado o utilitário fake_router26. Este utilitário dispara mensagens *Router Advertisement* na rede local, anunciando a própria máquina onde é executado como roteador. Entre as opções deste utilitário estão a possibilidade de especificar um prefixo de rede a ser divulgado, a possibilidade de especificar um endereço IP e/ou um endereço MAC de origem e até mesmo opções para burlar o mecanismo de segurança *Router Advertisement Guard* (RA Guard). Se executado sem opções, apenas tendo uma interface de rede como parâmetro, o que é obrigatório, o fake_router26 enviará mensagens *Router Advertisement* tendo como endereços IP e MAC de origem os respectivos endereços da máquina onde é executado. Este experimento foi executado sob duas abordagens diferentes: execução do fake_router26 sem opções, com o intuito de capturar o tráfego da rede local a partir da máquina atacante, e a execução do fake_router26 especificando um endereço IPv6 link-local falso, com o intuito de realizar um ataque de DoS. Este experimento envolveu a máquina atacante, executando o fake_router26, e dois hosts, nos quais era testada a conectividade com a Internet.

6.2.3. *Flooding* de mensagens *Neighbor Advertisement*

Neste experimento, a máquina atacante realiza um ataque de flooding com mensagens *Neighbor*

Advertisement, o que significa enviar uma grande quantidade de mensagens *Neighbor Advertisement* para a rede local, causando negação de serviço (DoS).

Para este ataque, foi utilizado o utilitário flood_advertise6. Quando executado, este utilitário dispara uma grande quantidade de mensagens *Neighbor Advertisement* na rede local e segue disparando mensagens até que seja parado manualmente. O flood_advertise6 recebe como parâmetro uma interface de rede, por onde serão disparadas as mensagens, e pode receber, opcionalmente, o endereço IP de um alvo para
o flood. Neste experimento, flood_advertise6 foi executado sem opções. O experimento envolveu a máquina atacante, executando o flood_advertise6, dois hosts, nos quais era testada a conectividade com a Internet, e o servidor web, verificando-se a possibilidade de acessá-lo.

6.2.4. *Flooding* de mensagens *Router Advertisement*

Este experimento também consiste em um ataque de flooding. No entanto, o flooding foi feito com mensagens Router Advertisement. Neste caso, a máquina atacante disparava uma série de mensagens Router Advertisement falsas que, além de serem enviadas em grande quantidade devido ao flooding, continham prefixos de rede e informações de roteamento inválidas, o que faz com que os hosts que recebem estas mensagens adquiram um grande número de endereços IPv6 e rotas inválidas.

Para este experimento foi utilizado o utilitário flood_router26, que dispara uma grande quantidade de mensagens Router Advertisement falsas. Existem diferentes modos de execução para este utilitário: enviar apenas informações de roteamento, enviar apenas informações de prefixo de rede e desabilitar extensões de privacidade. Por padrão, o flood_router26 envia tanto informações de roteamento quanto prefixos de rede. Este utilitário inclui opções que permitem alterar parâmetros das mensagens RA (como tamanho da mensagem e TTL) e também opções para tentar burlar RA Guard. Neste experimento, o flood_router26 no modo padrão, sem opções, tendo como parâmetro apenas a interface de rede, parâmetro obrigatório. Este experimento envolveu a máquina atacante, disparando mensagens Router Advertisement, os dois hosts e o servidor web, para os quais era testada a conectividade via ICMP e HTTP, respectivamente.

6.3. Cenário de testes NAT64

O cenário de testes NAT64 foi elaborado com o objetivo de realizar testes envolvendo vulnerabilidades específicas do mecanismo de tradução NAT64 e do DNS64, além de realizar testes com vulnerabilidades específicas do IPv6. Este cenário é composto por um roteador, que também atua também como firewall e NAT64; um servidor web e três hosts, sendo que um dele atua como atacante na rede local. O roteador deste cenário possui três interfaces físicas de rede (eth0, eth1 e eth2) e uma interface TUN (nat64), utilizada pela ferramenta que realiza a tradução NAT64. As interfaces eth0 e eth1 estão diretamente conectadas ao roteador do PoP-SC respectivamente via IPv4 e IPv6. A interface eth2 atua como interface LAN e, portanto, possui apenas endereçamento IPv6. A interface nat64 possui endereçamento IPv4 e IPv6, uma vez que, por esta interface, passarão todos os pacotes traduzidos.

Para realizar a tradução via NAT64, foi utilizada a ferramenta Tayga. O Tayga é uma implementação de NAT64 *stateless* para Linux. Conforme mencionado anteriormente nesta seção, o Tayga utiliza uma interface TUN para interceptar e traduzir os pacotes. Como é necessário que o NAT64 opere em conjunto com DNS64, foi utilizada a ferramenta Bind para operar em conjunto com o Tayga, realizando a função de DNS64. Aos hosts, conectados à interface eth2 do roteador, os endereços IP são atribuídos dinamicamente via RADVD. A Figura 2 mostra um diagrama deste cenário.

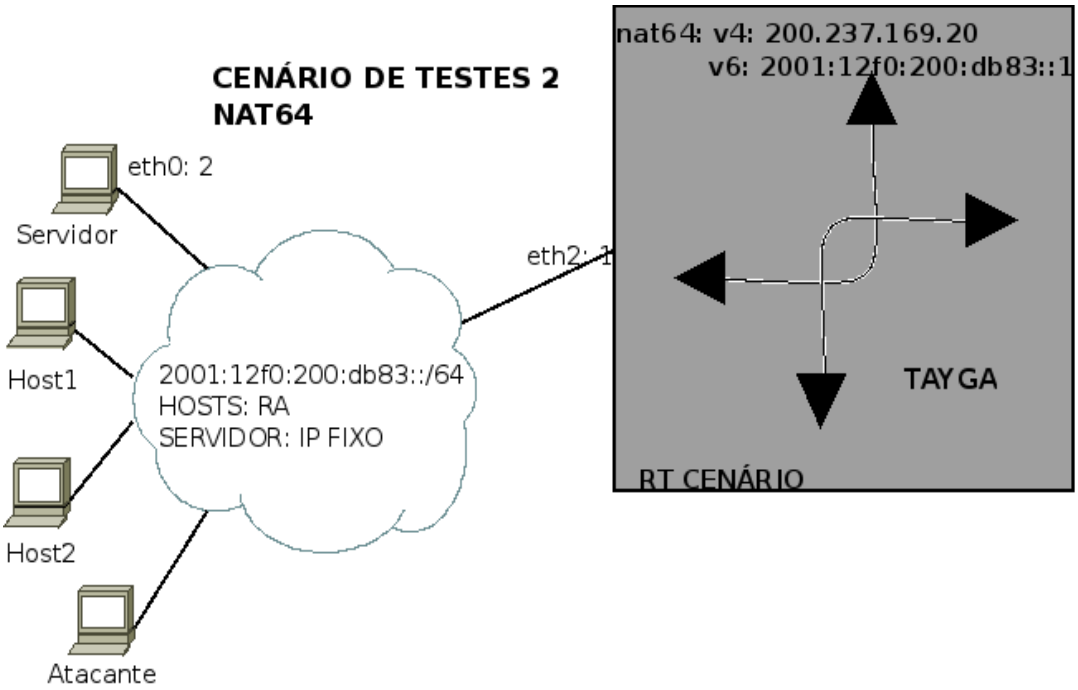

Figura 2: Cenário de testes NAT64

6.4. Experimentos realizados no cenário de testes NAT64

Nos experimentos feitos neste cenário, foram repetidos os ataques realizados no cenário pilha dupla e, além disto, foi realizado o seguinte ataque:

- **Servidor DNS falso –** atacante atua como servidor DNS, respondendo a consultas com um endereço falso.

6.4.1. Servidor DNS falso

O DNS64 é um mecanismo muito importante para o funcionamento do NAT64, uma vez que o endereço de um servidor IPv4 a ser mapeado em um endereço IPv6 é obtido através de um registro A, a ser sintetizado em um registro AAAA. Sem a possibilidade de obter o endereço do servidor IPv4, não é possível realizar o mapeamento deste endereço em um endereço IPv6, impossibilitando a comunicação de hosts IPv6 com este servidor. O ataque deste experimento visa explorar esta vulnerabilidade fazendo com que a máquina atacante atue como um servidor DNS falso, que responde a todas as consultas com um mesmo endereço IPv6.

Para o ataque deste experimento foram usados dois utilitários do THC-IPV6: o parasite6, que redireciona o tráfego da rede local para a máquina atacante interceptando e realizando spoofing de mensagens Neighbor Solicitation, e o fake_dns6d, um servidor DNS falso que responde a todas as consultas por um registro AAAA com o mesmo endereço IPv6. Optou-se por usar ambas as ferramentas conjuntamente pois desta forma o tráfego da rede local poderia ser desviado para o servidor DNS falso, o que torna o ataque mais eficaz.

A ferramenta parasite6 redireciona o tráfego da rede local para a máquina atacante através da interceptação e spoofing de mensagens Neighbor Solicitation. Esta ferramenta recebe como parâmetro, obrigatoriamente, uma interface de rede, podendo receber opcionalmente um endereço MAC falso. Ao ser executado, este utilitário detecta mensagens Neighbor Solicitation na rede, intercepta estas mensagens e as envia como se fossem provenientes da máquina atacante ou de uma máquina inexistente caso seja especificado um endereço MAC falso, o que resulta em um ataque de DoS. Neste experimento, o parasite6 foi executado recebendo uma interface de rede como parâmetro.

O utilitário fake_dns6d consiste em um servidor DNS falso, que responde a consultas por um registro AAAA com o mesmo endereço IPv6. Esta ferramenta recebe como parâmetro, obrigatoriamente, uma interface de rede e o endereço IPv6 utilizado nas respostas das consultas. Há ainda opções para burlar mecanismos de segurança do IPv6. Neste experimento, o fake_dns6d foi executado apenas com os parâmetros obrigatórios. Este endereço envolveu a máquina atacante, redirecionando o tráfego para si mesma e atuando como servidor DNS falso, e dois hosts, tentando conectividade com servidores externos.

7. Resultados obtidos

Na Seção 6, são descritos os experimentos realizados em cada um dos cenário de testes, levando em conta os ataques realizados e ferramentas utilizadas. Cada um destes ataques foi documentado e suas consequências, analisadas. Além disto, foram levantadas formas de detecção e mitigação destes ataques. Esta seção descreve os resultados obtidos após os testes.

7.1. *DoS* para novos endereços IPv6

Este experimento foi executado nos cenários de teste pilha dupla e NAT64 e envolveu a execução do ataque de *DoS* para novos endereços IPv6. Este ataque explora a funcionalidade Duplicate Address Detection (DAD) fazendo com que todos os endereços IPv6 que um host tentar atribuir a uma de suas interfaces de rede seja considerado em uso. Para a realização deste ataque foi utilizada a ferramenta dos-new-ip6. Esta ferramenta foi executada recebendo como parâmetro a interface eth0 da máquina atacante. Após isto, as interfaces de rede dos dois hosts do cenário foram reiniciadas para que eles tentassem atribuir novamente um endereço IPv6 a elas via autoconfiguração *stateless*. Os *hosts*, por sua vez, enviariam mensagens Neighbor Solicitation para todos os nós da rede para descobrir se o endereço IPv6 que pretendem atribuir à suas interfaces de

rede já está sendo usado. Se algum outro host estiver usando um destes endereços IPv6, ele deve enviar uma mensagem *Neighbor Advertisement* ao host que pretende usar este endereço. Caso a autoconfiguração falhasse e as interfaces de rede dos hosts não obtivessem um endereço IPv6, o sucesso do ataque seria confirmado.

A Figura 3 mostra a execução do ataque na CLI da máquina atacante. Como pode ser visto na figura, o dos_new_ip6 apresenta mensagens informando que foi realizado com sucesso o spoofing de pacotes ICMPv6 que, neste caso, são mensagens *Neighbor Advertisement* enviadas em resposta às mensagens *Neighbor Solicitation* dos *hosts* como se a máquina atacante possuísse seus endereços IPv6 pretendidos. Estas mensagens serão enviadas a qualquer *host* que tentar obter endereços IPv6 da rede local do cenário. O atacante, portanto, age como se possuísse todos os endereços IPv6 disponíveis.

A

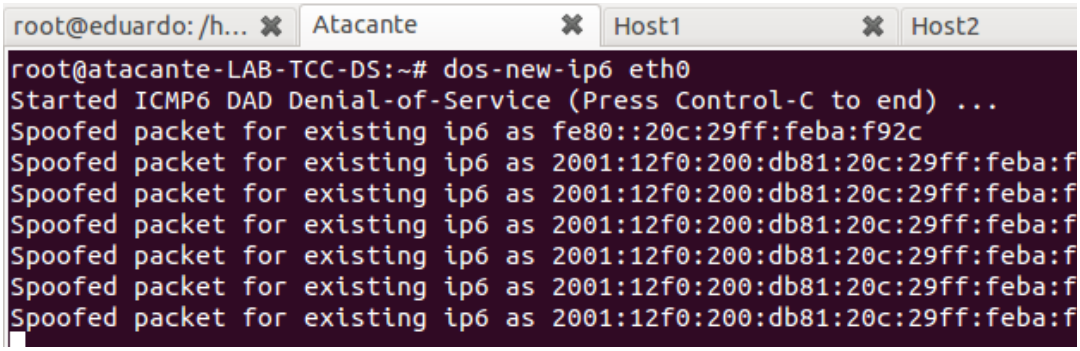

Figura 3: Máquina atacante executando ataque de DoS para endereços IPv6 novos

Figura 4 mostra a saída de um comando *ifconfig*, utilizado para configurar e verificar a configuração de interfaces de rede no sistema operacional Linux, em um dos *hosts* do cenário, executado antes do ataque. Como pode-se observar na figura, a interface eth0 do *host* possui um endereço IPv6 link-local e um endereço IPv6 global, podendo, portanto, se conectar à Internet via IPv6. Já na Figura 5, é mostrada a saída do comando *ifconfig* após a realização do ataque. É possível observar na figura que o *host* possui apenas um endereço IPv6 link-local, o que o impossibilita de conectar-se à Internet via IPv6. Podemos observar, portanto, que a maquina atacante conseguiu impedir que as demais máquinas obsivessem endereços IPv6 globais e obteve êxito no ataque de *DoS.*

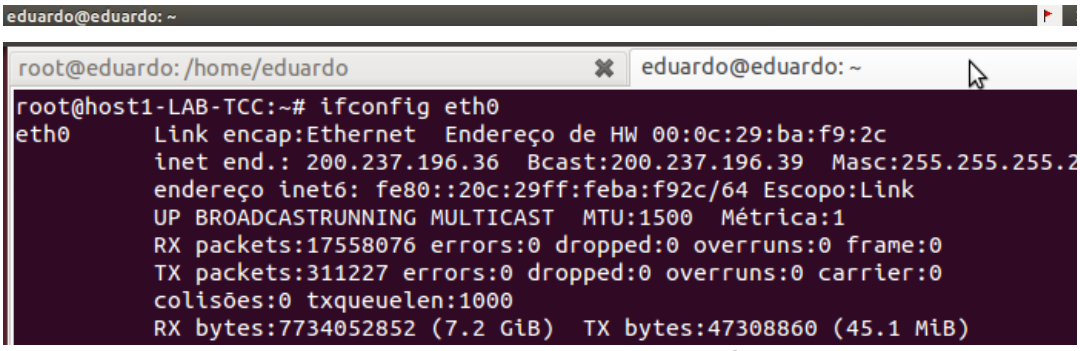

Figura 5: Configuração da interface eth0 de um host após o ataque de DoS a endereços IPv6 novos.

Figura 4: Configuração da interface eth0 de um host antes do ataque de DoS a endereços IPv6 novos.

7.2. Anúncio de um roteador falso

Este experimento foi executado nos cenários de testes pilha dupla e NAT64 e envolve o anúncio de um roteador falso na rede local. Este ataque consiste em enviar mensagens *Router Advertisement* falsas, anunciando a máquina atacante como roteador e direcionando o tráfego da rede local para o atacante. Este ataque foi executado sob duas abordagens diferentes: redirecionar o tráfego da rede local para a máquina atacante (caractericando um ataque de *man-in-the-middle*) e o anúncio de um roteador com endereço IPv6 inexistente, acarretando em um ataque de *DoS*.

7.2.1 Anúncio de um roteador falso – *man-in-the-middle*

Para executar um ataque de *man-in-the-middle* através do envio de mensagens *Router Advertisement* falsas, foi executada a ferramenta *fake_router26* sem opções, tendo como parâmetro apenas a interface eth0 da máquina atacante, o que faz com que as mensagens *Router Advertisement* enviadas anunciem a própria máquina atacante como roteador, isto é, as mensagens *Router Advertisement* tem como endereço IPv6 e MAC de origem os endereços IPv6 e MAC da máquina atacante. Com isto, o tráfego que parte da rede local do cenário é redirecionado para a máquina atacante, possibilitando à mesma a captura deste tráfego.

Caso este ataque seja bem sucedido, ele alterará, primeiramente, as configurações de rota padrão dos *hosts*. A Figura 6 mostra a execução do comando *route -n6* após a execução do ataque. Pode-se observar na figura que há dois endereços IPv6 como rota padrão, sendo que o segundo endereço pertence à máquina atacante e foi configurado a partir das mensagens *Router Advertisement* falsas.

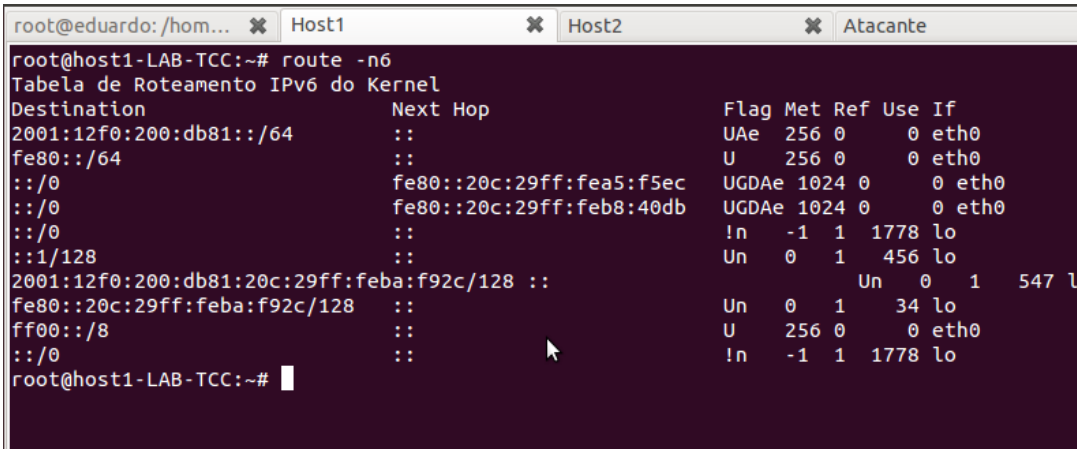

Figura 6: Saída do comando route -n6 após o anúncio do roteador falso.

Após a confirmação de que as configurações de rota padrão dos *hosts* haviam sido alteradas em função do ataque, foram realizados testes de conectividade. Em um dos *hosts*, foi executado o comando *ping6* para um servidor externo, testando a conectividade via ICMPv6. Em outro *host*, foi testada a conectividade via HTTP no servidor web do cenário através do comando *wget*.

Na Figura 7 é mostrada parte da saída do comando *ping6* no primeiro *host*. Pode-se observar que as mensagens ICMPv6 *echo request* recebiam resposta (*echo reply*). Porém, a taxa de perdas de pacotes chegou a 94%, o que mostra que este ataque também teve o efeito de prejudicar bastante a conectividade com a Internet.

Na

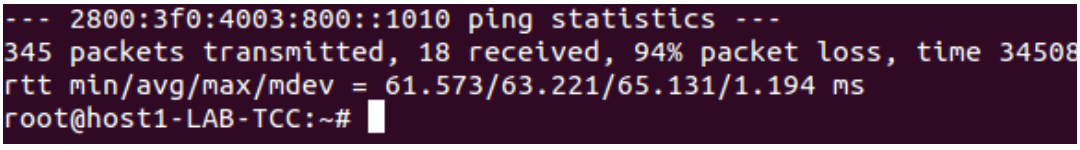

Figura 7: Teste de conectividade com um servidor externo via ping6.

Figura 8 é mostrada a saída do comando *wget* no segundo *host*. Pode-se observar que a conexão do *host* com o servidor web do cenário é realizada com sucesso em quase todas as tentativas, falhando apenas em uma. Isto mostra que a conectividade com o servidor interno foi pouco prejudicada. No entanto, o tráfego entre os *hosts* e qualquer servidor, interno ou externo, pode ser fácilmente capturado. A Figura 9 mostra uma captura de pacotes realizada a partir da máquina atacante. Podemos ver na figura as mensagens *echo request* que o primeiro *host* envia ao servidor externo, com o objetivo de realizar um teste de conectividade. Isto mostra que o tráfego dos *hosts* está passando pela máquina atacante, o que significa que o atacante tem acesso à comunicação dos *hosts* com qualquer servidor e pode facilmente vizualizar dados não protegidos, caracterizando-se então um bem sucedido ataque de *man-in-the-middle* explorando o mascaramento de pacotes ICMPv6 do tipo *Router Advertisement*, uma vulnerabilidade não prevista em redes IPv4.

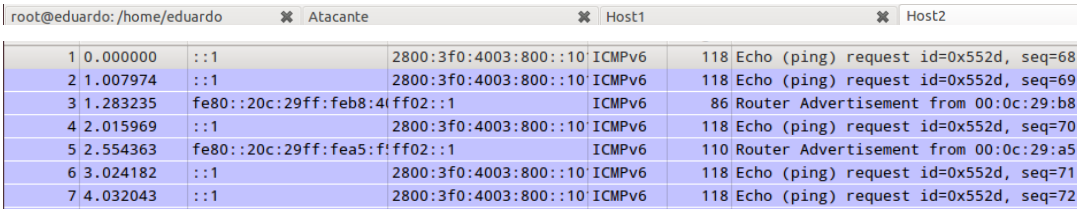

Figura 9: Captura de pacotes a partir da máquina atacante.

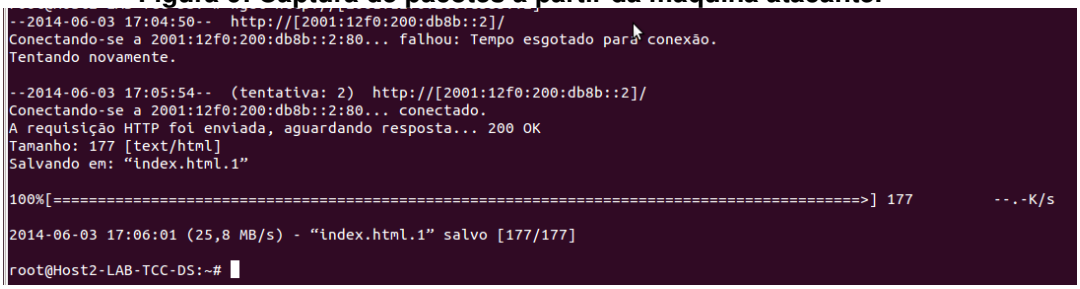

Figura 8: Teste de conectividade via HTTP com o servidor interno.

7.2.2. Anúncio de um roteador falso – *DoS*

Outra abordagem para atacar a rede local anunciando um roteador falso é realizar um ataque de *DoS* através do anúncio de um roteador inexistente. Isto pode ser feito através do envio de mensagens *Router Advertisement* que tem como endereço IPv6 e MAC de origem endereços IPv6 e MAC que não estão atribuídos a nenhum outro *host.* Desta forma, os *hosts* que recebem estas mensagens *Router Advertisement* adicionam às suas tabelas de roteamento uma rota padrão para um roteador inexistente e, portanto, os pacotes enviados a este roteador são descartados. Para realizar este ataque, foi utilizada a ferramenta *fake_router26* executada com a opção *-s <ip-origem>*, espeficicando-se um endereço IPv6 inexistente no parâmetro *ip-origem*, como é mostrado na Figura 10.

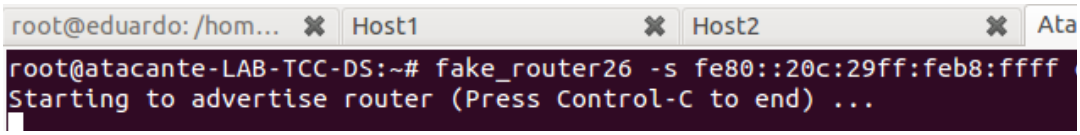

Figura 10: Execução da ferramenta fake_router26 para gerar um ataque de DoS através do envio de mensagens Router Advertisement falsas.

Para verificar os efeitos do ataque, inicialmente verificou-se a alteração da configuração de rotas padrão nos *hosts* através do comando *route -n6*. Assim como no ataque descrito na Seção 7.2.1, ao executar o comando *route -n6* após o início do ataque, verificou-se o surgimento de uma segunda rota padrão dos *host*, indicando que os *hosts* do cenário, ao receberem as mensagens *Router*

Advertisement falsas, configuraram rotas para o roteador falso anunciado.

Após a verificação da alteração das configurações de rota padrão dos *hosts*, foram realizados testes de conectividade tanto para a Internet quanto para a própria rede do cenário. Primeiramente, foi executado um teste de conectividade via ICMPv6 através do comando *ping6* para um servidor externo. Parte da saída do comando *ping6* executado é mostrada na Figura 11. como pode ser observado na figura, houve uma taxa de perda de pacotes de 100%. Esta taxa de perda de pacotes mostra que o *host* não consegue comunicação com o servidor externo e com a Internet pois, em função do ataque, os pacotes enviados pelo *host* são encaminhados para um roteador inexistente e, portanto, acabam sendo descartados.

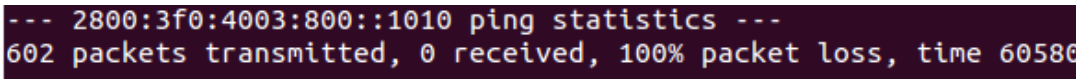

Figura 11: Teste de conectividade com servidor externo via ICMPv6.

Após a verificação da conectividade via ICMPv6, foi realizado um teste de conectividade com o servidor web do cenário via HTTP utilizando o comando *wget*. A saída do comando é mostrada na Figura 12. Observa-se na figura que, ao contrário do ocorrido no ataque de *man-in-the-middle* através do anúncio de um roteador falso, não foi possível conectar-se ao servidor do cenário via HTTP em nenhuma das tentativas, também em função do desvio dos pacotes para um roteador inexistente.

Figura 12: Teste de conectividade com o servidor do cenário via HTTP.

Os testes executados após o ataque mostram que a conectividade dos *hosts* interna e externamente ao cenário foi totalmente comprometida pelo anúncio de um roteador inexistente. Constata-se, portanto, que é possível utilizar mensagens *Router Advertisement* mascaradas para provocar com sucesso um ataque de *DoS* em uma rede local IPv6.

7.3. *Flooding* de mensagens *Neighbor Advertisement*

Este experimento foi executado nos cenários pilha dupla e NAT64 e envolve o ataque de *flooding* de mensagens *Neighbor Advertisement.* Este ataque consiste em disparar uma grande quantidade de mensagens *Neighbor Advertisement* com o intuito de provocar um ataque de *DoS* sobrecarregando tanto a rede com o volume de tráfego gerado quanto o roteador local com o processamento de uma grande quantidade de mensagens *Neighbor Advertisement*. Este ataque foi realizado por meio da ferramenta *flood_advertise6*, que dispara um grande número de mensagens *Neighbor Advertisement* até que seja parado manualmente. Para este cenário, o *fake_advertise6* foi executado tendo como parâmetro a interface eth0 da máquina atacante. A Figura 13 mostra a execução do *flood_advertise6* na máquina atacante do cenário pilha dupla. Cada um dos pontos impressos na tela indicam mil mensagens *Neighbor Advertisement* disparadas pela ferramenta pela interface de rede especificada.

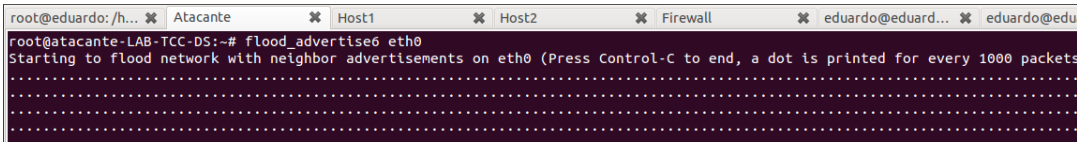

Figura 13: Execução do flood_advertise6 na máquina atacante

Para verificar a negação de serviço, foram executados testes de conectividade com os *hosts* e com o servidor web do cenário. O teste de conectividade com os *hosts* foi realizado via ICMPv6 através do comando *ping6*. No servidor web, a conectividade foi testada via HTTP utilizando-se um navegador web. Verificou-se nos testes que, após o início do ataque, tanto a tentativa de conectividade com o servidor externo quanto com o servidor interno foram mal sucedidas, portanto ao analisar os resultados dos testes de conectividade realizados, verifica-se que o grande volume de mensagens *Neighbor Advertisement* disparadas na rede local afeta todos os *hosts* nela presentes e até mesmo o servidor web. Através da sobrecarga da rede com mensagens *Neighbor Advertisement*, que, por sua vez, sobrecarregam o roteador local com o processamento destas mensagens, realizou-se um bem sucedido ataque de *DoS* utilizando as mensagens *Neighbor Advertisement*, previstas apenas a partir do IPv6.

7.4. *Flooding* de mensagens *Router Advertisement*

Além do ataque de *flooding* com mensagens *Neighbor Advertisement*, foi realizado, nos cenários pilha dupla e NAT64, um ataque de *flooding* com mensagens *Router Advertisement*. Este ataque, caso seja bem sucedido, não apenas sobrecarrega a rede local com o grande volume de tráfego como também faz com que os *hosts* que recebem as mensagens *Router Advertisement* disparadas na rede configurem em suas interfaces de rede endereços IPv6 inválidos, assim como faz com que os *hosts* recebam informações de roteamento inválidas. Este ataque foi realizado através da execução da ferramenta *flood_router26* recebendo como parâmetro a interface eth0 da máquina atacante. Ao ser executado o *flood_router26*, a máquina atacante passou a disparar uma grande quantidade de mensagens *Router Advertisement* contendo prefixos IPv6 aleatórios.

As mensagens *Router Advertisement* carregam informações de prefixo e roteamento utilizadas na autoconfiguração de endereços. Como consequência deste ataque, os *hosts* deveriam configurar diversos endereços IPv6 em suas interfaces de rede conectadas à rede local. Na Figura 14, é mostrada a configuração da mesma interface deste *host* após o ataque. Pode-se verificar que há diversos endereços IPv6 configurados e que estes outros endereços possuem um prefixo diferente do prefixo de rede do cenário. Verifica-se, portanto, que estes endereços foram configurados a partir das mensagens *Router Advertisement* ilegítimas enviadas pela máquina atacante.

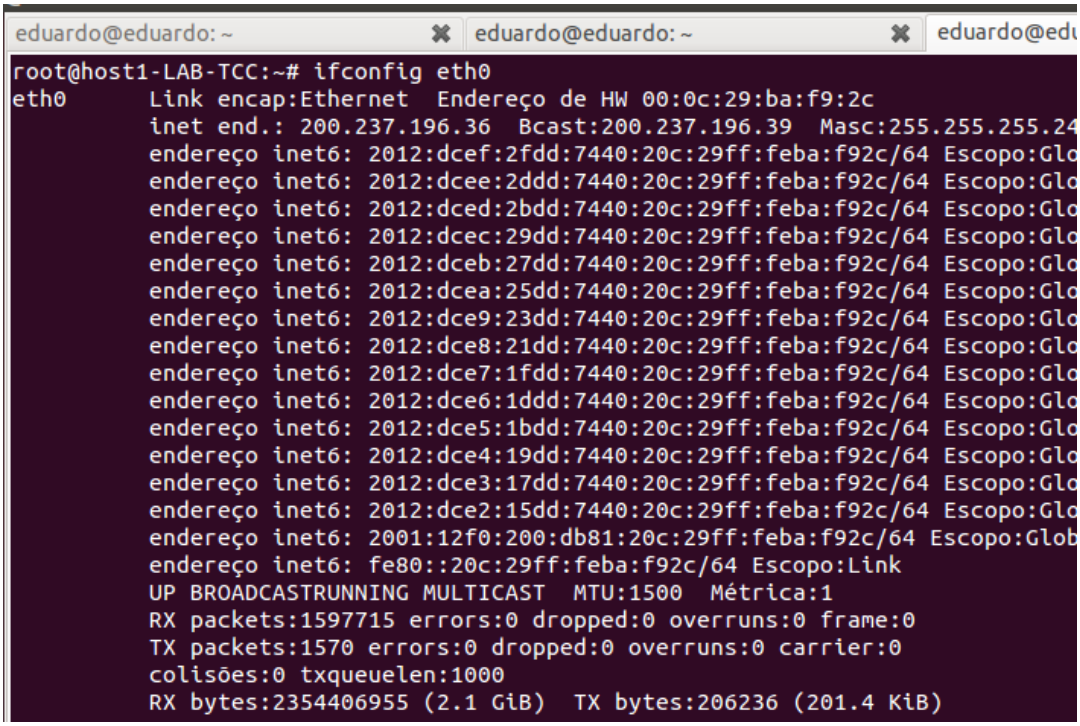

Figura 14: Configuração da interface de rede de um dos hosts após o início do ataque.

Após o início do ataque, foram realizados testes de conectividade nos *hosts* e no servidor web. Nos *hosts*, o teste de conectividade foi realizado via ICMPv6 através do comando *ping6*, enquanto no servidor web, o teste foi realizado via HTTP através de um navegador web. Verificou-se que, após o início do ataque, ambos os testes resultaram em tentativas de conexão mal sucedidas.

Ao analisar os resultados dos testes, verifica-se que as mensagens *Router Advertisement* enviadas pela máquina atacante, além de sobrecarregarem a rede local em função do grande volume, impedem a conectividade com os *hosts* em função dos endereços IPv6 e informações de roteamento inválidas. Obteve-se, portanto, mais um ataque de *DoS* bem sucedido explorando o mecanismo de autoconfiguração de endereços IPv6.

7.5. Servidor DNS falso

Uma maneira eficaz de atingir especificamente redes IPv6 que se utilizam o mecanismo de tradução NAT64 para se comunicarem com redes puramente IPv4 é realizando um ataque visando o mecanismo DNS64, que é de grande importância para a tradução de endereços uma vez que os endereços IPv4 a serem mapeados em endereços IPv6 são obtidos através de consultas DNS, sendo os registros A convertidos em registros AAAA através do DNS64. Neste experimento, realizado apenas no cenário NAT64, a máquina atacante atuou como um servidor DNS falso que respondia a todas as consultas por um registro AAAA com o mesmo endereço IPv6. Este ataque causa dois grandes impactos. Primeiramente, a máquina atacante pode capturar tráfego dos *hosts*, uma vez que atuará como servidor DNS na rede local e responderá a consultas dentro da mesma. Além disto, o fato do servidor DNS falso reponder a todas as consultas com o mesmo endereço IPv6 pode simplesmente inviabilizar a comunicação com o servidor IPv4 real.

Este ataque envolveu duas etapas. Primeiramente, foi executada a ferramenta *parasite6* recebendo como parâmetro a interface eth0 da máquina atacante, para que todo o tráfego da rede local fosse desviado para a máquina atacante. Após isto, foi executada aferramenta *fake_dns6d*, ferramenta que atua como o servidor DNS falso, recebendo como parâmetro o endereço IPv6 que seria utilizado para responder às consultas. A Figura 15 mostra a execução destas ferramentas.

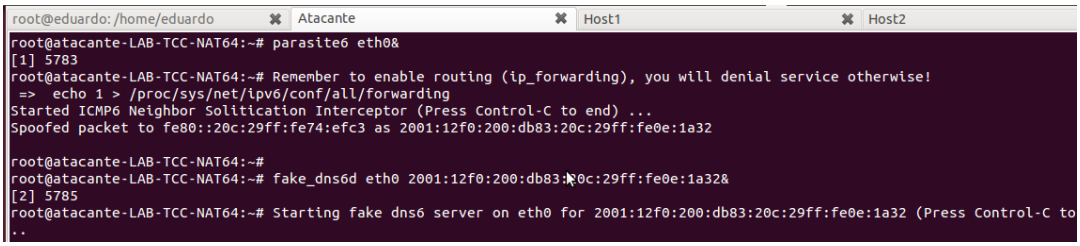

Figura 15: Execução das ferramentas parasite6 e fake_dns6d.

Para verificar a atuação do servidor DNS falso, foi executado, primeiramente, o comando *ping6* para um servidor puramente IPv4 externo através do nome deste servidor. A Figura 16 mostra a saída do comando *ping6* após o ataque. Verifica-se que, desta vez, o endereço IPv6 presente nas respostas é o mesmo endereço passado à ferramenta *fake_dns6d* como parâmetro. Isto significa que, após o início do ataque, as consultas DNS estão sendo respondidas pelo servidor falso. Além disto, após o ataque, foi feita uma tentativa de estabelecer uma conexão HTTP com o servidor externo através do comando *wget*. Foi possível verificar que a conexão não foi bem sucedida pois o endereço IP associado ao nome do servidor externo é o endereço IP falso fornecido pela máquina atacante enquanto servidor DNS falso.

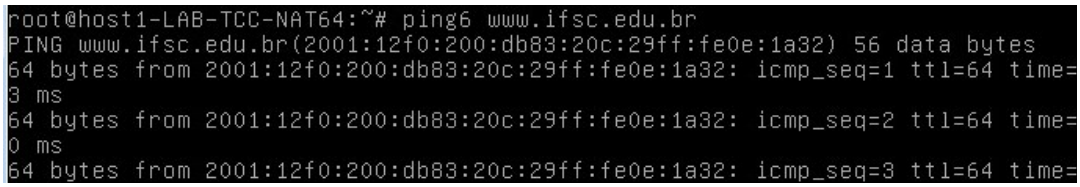

Figura 16: Teste de conectividade via ICMPv6 após o início do ataque.

Verificou-se neste ataque que a máquina atacante conseguiu atuar como servidor DNS e responder às consultas da rede local, comprometendo de forma crítica a tradução de endereços via NAT64. Este ataque se mostra particularmente crítico pois é difícil implementar uma forma de defesa para o mesmo e redes puramente IPv6 podem ficar totalmente isoladas da Internet IPv4 em função deste ataque.

7.6. Detecção dos ataques

Além da realização de ataques nos experimentos realizados, foram levantadas formas de detecção dos ataques. Tendo em vista os ataques realizados, foram identificadas duas maneiras principais de detectá-los: via captura de pacotes e através da ferramenta 6Guard.

Uma vez que a maioria dos ataques realizados envolvem mascaramento (*spoofing)* de pacotes, uma maneira de detectá-los é através da captura e análise de tráfego. Analisando os pacotes capturados, foi possível diferenciar os pacotes legítimos dos pacotes gerados pela máquina atacante e, portanto, foi possível constatar a ocorrência de um ataque. Através da captura e análise de tráfego foi possível detectar os ataques de *DoS* para novos endereços IPv6, anúncio de um roteador falso, *flooding* de mensagens *Neighbor Advertisement* e *flooding* de mensagens *Router Advertisement*. Para a captura de pacotes foi utilizada a ferramenta *tcpdump* e para a análise, a ferramenta Wireshark.

Outra maneira de detectar os ataques que envolvem mascaramento de pacotes é através da ferramenta *6Guard*. O *6Guard* é uma ferramenta para a detecção de ataques ao IPv6 baseada em *honeypot* projetada para detectar ataques em uma rede local. O *6Guard* é escrito em *Python*, utilizando a biblioteca *Scapy* e contém três módulos:

- **Módulo *honeypot* –** um *host* IPv6 de baixa interação que suporta as funcionalidades NDP e autoconfiguração *stateless*, responsável por detectar ataques unicast.
- **Módulo *globalpot* –** módulo que foca na detecção de ataques multicast.
- **Módulo de análise de eventos –** módulo responsável por analisar as mensagens recebidas na rede local e gerar as mensagens de alerta (THE HONEYNET PROJECT, 2014).

O *6Guard* é uma forma automatizada de detecção de ataques pois, ao detectar tráfego proveniente de um ataque, o *6Guard* emite mensagens de alerta informando do ataque, como a mostrada na Figura 17. Através do *6Guard* foi possível detectar os ataques de *DoS* para novos endereços IPv6, anúncio de um roteador falso, *flooding* de mensagens *Neighbor Advertisement* e *flooding* de mensagens *Router Advertisement*.

O

```
[ATTACK]
Timestamp: 2014-05-14 17:11:22
Reported by: Globalpot
Type: DoS
Name: Fake Router Advertisement against the fake router
Attacker: [fe80::20c:29ff:feb8:40db]  00:0c:29:b8:40:db (VMware, In
Victim  : [The whole network]
Utility: THC-IPv6: kill_router6
Packets: b78969d638c151a0ec553f05bd75b8a3.pcap
```

Figura 17: Mensagem de alerta do 6Guard.

ataque envolvendo um servidor DNS falso é um caso particular, pois não é possível detectá-lo via captura de pacotes ou via *6Guard*. No entanto é possível detectar este ataque através da própria tentiva de acesso a um servidor externo a partir de um dos *hosts*, como é mostrado na Figura 16. Ao tentar acessar o servidor externo e verificar que a tentativa de conexão foi mal sucedida, pode-se verificar que o nome do servidor externo foi resolvido em um endereço IPv6 que está contido na faixa de endereçamento da rede local. Pode-se concluir que há um provável atacante dentro da própria rede local direcionando seu ataque ao DNS.

7.7. Possíveis formas de defesa

Após a verificação de vulnerabilidades nos cenários de testes, buscou-se uma solução para as vulnerabilidades apontadas. A busca desta solução envolve levantar formas de detecção dos ataques, e formas de defesa contra os mesmos. De forma geral, os ataques realizados envolvem NDP, autoconfiguração de endereços e DNS64. Tendo isto em vista, foram levantadas três formas de defesa que podem cobrir este espectro de ataques: *Secure Neighbor Discovery* (SEND), *Router Advertisement Guard* (RA Guard) e *DNS Security* (DNSSEC).

7.7.1. Secure Neighbor Discovery – SEND

O SEND foi definido para proteger o NDP de ameaças que visam este mecanismo. O SEND acrescenta ao NDP um conjunto de novas funcionalidades e opções para torná-lo mais seguro, protegendo as mensagens trocadas no processo de descoberta de vizinhos. A principal funcionalidade de segurança do SEND é a utilização de endereços gerados criptograficamente (*Cryptographically Generated Address* – CGA). Estes endereços são utilizados para garantir que um nó que envia uma mensagem *Neighbor Advertisement* informando que já possui um determinado IPv6 realmente possui aquele endereço. Antes que os nós de uma rede possam enviar mensagens informando que possuem um determinado endereço IPv6, estes nós geram um par de chaves pública e privada. Os nós da rede trocam chaves públicas através da opção CGA, inserida nas mensagens do NDP com o SEND. Outras funcionalidades importantes do SEND são a certificação de roteadores, isto é, um *host* deve ser confiável ao roteador antes que possa defení-lo como roteador padrão, e a autenticação de mensagens via assinatura digital (ARKKO et al., 2014).

O SEND pode ser eficaz na defesa contra os ataques de *DoS* para novos endereço IPv6 e de *flooding* de mensagens *Neighbor Advertisement*. Devido à certificação de roteadores e à autenticação de mensagens via assinatura digital, ele pode ser utilizado também na proteção contra o ataque de *flooding* de mensagens *Router Advertisement*, já que esta funcionalidade poderia evitar a configuração de endereços e rotas ilegítimas nos *hosts*.

Atualmente, existem implementações do SEND para roteadores CISCO (CISCO, 2014) e Juniper (JUNIPER, 2014), além de implementações para Linux, como o NDProtector (AMNESIAK, 2014), Easy-SEND (SOURCEFORGE, 2014) e ipv6-send-cga (GOOGLE, 2014). Nos experimentos, não foi possível testar estas soluções devido à indisponibilidade de roteadores reais para experimentos e, no caso das implementações para Linux, devido à dificuldades de instalação e pouca documentação ou suporte.

7.7.2. Router Advertisement Guard – RA Guard

O RA Guard é uma solução desenvolvida para detectar e bloquear mensagens *Router Advertisement* falsas em redes IPv6 onde não há suporte completo ao SEND e onde todo o tráfego IPv6 passa por dispositivos de camada 2 gerenciáveis, como *switches*, capazes de bloquear as mensagens *Router Advertisement* falsas. O RA Guard pode ser *stateless* ou *statefull*. Uma implementação *stateless* do RA Guard examina os pacotes recebidos e decide se os mesmos serão encaminhados ou descartados. Para decidir se encaminhará ou não as mensagens, o RA Guard *stateless* examina parâmetros como endereço MAC de origem, porta do equipamento onde a mensagem foi recebida, endereço IPv6 de origem e lista de prefixos contida na mensagem. Estes parâmetros são comparados com as configurações do equipamento, onde é definido quais os endereços MAC de origem permitidos, a portas onde podem ser recebidas mensagens *Router Advertisement*, endereços IPv6 e listas de prefixos permitidas. Já o RA Guard *stateful* aprende dinamicamente quais mensagens *Router Advertisement* devem ser permitidas e quais devem ser bloqueadas, armazenando informações sobre as mensagens durante um período de tempo para definir como estas mensagens serão tratadas posteriormente. Há ainda o RA Guard *stateful* baseado em SEND, que aplica as opções e funcionalidades do SEND a mensagens *Router Advertisement* (LEVY-ABEGNOLY et al., 2014).

O RAGuard pode ser eficaz contra os ataques de *DoS* através do anúncio de um roteador falso e de *flooding* de mensagens *Router Advertisement*. Com o RA Guard, seria possível definir políticas para identificar e bloquear as mensagens *Router Advertisement* ilegítimas geradas pela máquina atacante. Foram encontradas implementações do RA Guard para *switches* CISCO (CISCO, 2014) e Brocade (BROCADE, 2014). O RA Guard não pôde ser testado nos experimentos devido ao fato de não haver *switches* disponíveis para experimentos e também ao fato de não ter sido encontrada nenhuma solução para Linux.

7.7.2. Router Advertisement Guard – RA Guard

O DNSSEC consiste em especificações de segurança para algumas informações fornecidas pelo DNS. O DNSSEC permite a conferência de nomes de domínio e endereços IP correspondentes através de assinatura digital e criptografia de chaves públicas. Para o armazenamento das chaves, o DNSSEC prevê um registro específico chamado de DNSKEY. O uso do DNSSEC prevê que as respostas das consultas DNS contenham, além do registro consultado, um registro de assinatura digital chamado de RRSIG. O RRSIG é uma assinatura digital da resposta a uma consulta DNS, verificada através do registro DNSKEY (ARENDS et al., 2013).

O DNSSEC pode tornar mais seguro o mecanismo DNS64 em determinados casos, se for amplamente utilizado. No entanto, este mecanismo não se mostrou eficaz ara ataques como o realizado no cenário de testes NAT64, devido ao fato de todo o tráfego ser redirecionado à máquina atacante e também devido ao fato do DNSSEC não prover confidencialidade dos dados, permitindo que a máquina atacante ainda tenha acesso a dados desprotegidos capturados.

8. Conclusão e trabalhos futuros

A preocupação com o período de transição é bastante visível em trabalhos acadêmicos (SANKARAN, 2013) (BI; WU; LENG, 2007) (TAIB; BUDIARTO, 2007). Não só continuam a existir antigas ameaças do IPv4, como surgem novas ameaças específicas do IPv6 e dos

mecanismos de transição. No entanto, não se vêem muitas experiências práticas que levam em consideração as ameaças específicas do IPv6 e dos mecanismos de transição, testando a eficácia de estratégias de defesa para estas ameaças (SANKARAN, 2013) (TAIB; BUDIARTO, 2007). Este trabalho apresenta experimentos explorando vulnerabilidades do IPv6, levando em consideração as técnicas de transição. Com isto, contribui-se para a verificação, na prática, dos efeitos de ataques explorando vulnerabilidades específicas do IPv6 e das técnicas de transição, possivelmente estimulando a realização de mais testes.

Os efeitos dos ataques em cada um dos cenários se mostraram bastante parecidos nos cenários de testes. Ataques de *DoS* obtiveram êxito em negar serviço à rede local, nos ataques de *man-in-the-middle*, a máquina atacante interceptou com sucesso o tráfego dos *hosts* da rede local e os ataques de *flooding* sobrecarregaram a rede e os *hosts* da rede local. Deve-se considerar, no entanto que, apesar dos efeitos dos ataques serem semelhantes, suas consequências em cada um dos cenários apresentam uma grande diferença. No cenário pilha-dupla, apesar dos ataques comprometerem criticamente a rede IPv6, ainda há a conectividade via IPv4, que não é afetada. No momento em que se encontra a transição atualmente, estes ataques não causariam grande impacto, uma vez que a maior parte dos *sites* e serviços da Internet ainda operam com IPv4. Mas, à medida que a transição avança e mais servidores passam a operar apenas com IPv6, os ataques realizados podem ser muito mais perigosos, comprometendo completamente a conectividade. No cenário NAT64, os ataques já causam um impacto muito maior, uma vez que a rede local é puramente IPv6, portanto, ataques de *DoS* comprometeriam totalmente a conectividade dos *hosts* com a Internet e ataques de *man-in-the-middle* capturam todo o tráfego dos *hosts*. Os efeitos destes tipos de ataques explorando vulnerabilidades específicas do IPv6 serão de grande impacto tanto no estado atual quanto em estágios mais avançados da transição para o IPv6. No caso de ataques que visam especificamente o mecanismo de transição NAT64 e o DNS64, haverá impactos maiores a curto prazo, enquanto o IPv4 ainda é amplamente utilizado e comprometer a tradução de endereços tem maior efeito sobre a conectividade. No entanto, à medida que a transição avança, os impactos serão cada vez menores, uma vez que cada vez mais serviços e *sites* na Internet passam a operar apenas em IPv6 e a tradução passa a ser menos necessária.

Ressalta-se que, apesar do êxito dos ataques e apesar dos resultados dos experimentos mostrarem que a transição do IPv4 para o IPv6 e, posteriormente, a implantação do IPv6 por completo implicarão em novas vulnerabilidades na rede, em nenhum momento este trabalho conclui que o IPv6 é um protocolo inseguro e que sua adoção deve ser desestimulada. Acredita-se, com base nos resultados deste trabalho e nas formas de defesa pesquisadas, que a maior adoção do IPv6 tende a torná-lo mais seguro, uma vez que deve haver maior preocupação com as vulnerabilidades deste protocolo e, portanto, as soluções de segurança existentes, ainda recentes e oferecidas por um pequeno número de fabricantes de equipamentos de rede, tendem a ser aprimoradas, assim como a oferta das mesmas tende a aumentar. Os resultados deste trabalho devem servir para alertar sobre as vulnerabilidades inerentes ao IPv6 e às técnicas de transição e estimular, em conjunto com a adoção do IPv6, o desenvolvimento de soluções de segurança prevendo as vulnerabilidades inerentes ao IPv6 e às técnicas de transição.

8.1. Trabalhos futuros

- **Realização de experimentos com técnicas de tunelamento e a técnica DS-Lite** – estas técnicas de transição não foram contempladas com experimentos práticos neste trabalho. Propõe-se que sejam realizados experimentos contemplando as mesmas, uma vez que estas técnicas ainda são utilizadas em cenários de transição reais.
- **Realização de experimentos com formas de defesa em equipamentos de rede reais** – em função da indisponibilidade, no âmbito deste trabalho, de equipamentos reais para experimentos, como roteadores e *switches*, propõe-se um trabalho que viabilize a realização de experimentos em equipamentos de rede reais e teste estratégias e políticas de segurança

em cenários de transição IPv4/IPv6.

- **Realização dos experimentos deste trabalho explorando diferentes opções das ferramentas do THC-IPv6** – Os utilitários do THC-IPv6 utilizados neste trabalho oferecem diferentes opções além daquelas aqui utilizadas, que podem ser exploradas em novos experimentos em diferentes cenários.
- **Desenvolvimento de uma ferramenta que implemente os mecanismos SEND e RA GUARD para Linux –** Há um pequeno número de implementações do SEND e não foi encontrada nenhuma implementação do RA Guard para Linux. Propõe-se o desenvolvimento de ferramentas para suprir esta demanda.

Referências

AMNESIAK. **NDProtector**. Disponível em: <http://amnesiak.org/NDprotector/>. Acesso em: 6 jun. 2014.

ARENDS, R. et al. **Resource Records for the DNS Security Extensions.** RFC 4034. Disponível em: <http://www.ietf.org/rfc/rfc4034.txt>. Acesso em: 6 jun. 2014.

ARKKO, J et al. **SEcure Neighbor Discovery (SEND)**. RFC 3971. Disponível em: <http://www.hjp.at/doc/rfc/rfc3971.html>. Acesso em: 6 jun. 2014.

BAGNULO, M et al. **DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers.** RFC 6147 Disponível em: <http://tools.ietf.org/html/rfc6147>. Acesso em: 15 set. 2013.

BAGNULO, M; MATTHEWS, P; BEIJNUM, I Van. **Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers.** RFC 6146. IETF. Disponível em: <http://tools.ietf.org/html/rfc6146>. Acesso em: 15 set. 2013.

BI, Jun; WU, Jianping; LENG, Xiaoxiang. IPv4/IPv6 Transition Technologies and Univer6 Architecture. **International Journal Of Computer Science And Network Security**, Beijing, n. , p.232-243, jan. 2007.

BILSKI, Tomasz. Security-Functionality Tradeoffs in IP Transition Phase. **6th International Conference On Internet Technology And Secured Transactions**, Abu Dhabi, n. , p.632-638, dez. 2011.

BROCADE. **Example of configuring IPv6 RA guard.** Disponível em: <http://www.brocade.com/downloads/documents/html_product_manuals/FI_08000a_SECURITY/wwhelp/wwhimpl/common/html/wwhelp.htm#href=FI_ipv6_ra_guard.15.6.html&single=true>. Acesso em: 6 jun. 2014.

CAICEDO, Carlos; JOSHI, James; TULADHAR, Summit. IPv6 Security Challenges. **Ieee Computer Society,** [s.l.], v. 1, n. 1, p.36-42, fev. 2009.

CISCO (Eua). **IPv6 Secure Neighbor Discovery.** Disponível em: <http://www.cisco.com/en/US/docs/ios-xml/ios/sec_data_acl/configuration/15-2mt/ip6-send.html#GUID-DCB20ADF-1F8E-434B-AE97-54802879F34F>. Acesso em: 6 jun. 2014.

CISCO. **IPv6 RA Guard.** Disponível em: <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6/configuration/15-2s/ip6-15-2s-book/ip6-ra-guard.html>. Acesso em: 6 jun. 2014.

CONTA, A; DEERING, S; GUPTA, M. **Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.** RFC 4443. IETF. Disponível em: <http://tools.ietf.org/html/rfc4443>. Acesso em: 03 set. 2013.

DEERING, S; HINDEN, R. **Internet Protocol, Version 6 (IPv6) Specification.** RFC 2460. IETF. Disponível em: <http://xml2rfc.tools.ietf.org/html/rfc2460>. Acesso em: 01 set. 2013.

DOMINGOS, Fabiano. TECNICA DE TRANSIÇÃO ENTRE REDES IPV4/IPV6. **Revista de Ciências Exatas e Tecnologia**, [s. L.], n. , p.1-22, jan. 2006.

DURDAGI, Emre; BULDU, Ali. IPV4/IPV6 security and threat comparisons. **Procedia Social And Behavioral Sciences**, Istanbul, n. , p.5285-5291, 25 jan. 2010.

GOOGLE. **Ipv6-send-cga.** Disponível em: <http://code.google.com/p/ipv6-send-cga/>. Acesso em: 6 jun. 2014.

JANBEGLOU, Maziar; ZAMANI, Mazdak; IBRAHIM, Suhaimi. Redirecting Network Traffic toward a Fake DNS Server on a LAN. **3º Ieee International Conference On 3rd Computer Science And Information Technology**. Chengdu, p. 429-433. jul. 2010.

JUNIPER (Eua). **Example: Configuring Secure IPv6 Neighbor Discovery.** Disponível em: <http://www.juniper.net/techpubs/en_US/junos13.3/topics/topic-map/ipv6-secure-neighbor.html>. Acesso em: 6 jun. 2014.

LEVY-ABEGNOLI, E. et al. **IPv6 Router Advertisement Guard.** RFC 6105. Disponível em:

<http://www.hjp.at/doc/rfc/rfc6105.html>. Acesso em: 6 jun. 2014.
MOREIRAS, Antônio et al. **Cabeçalho.** Disponível em: <http://ipv6.br/entenda/cabecalho/>. Acesso em: 01 set. 2013.
MOREIRAS, Antônio et al. **Endereçamento.** Disponível em: <http://ipv6.br/entenda/enderecamento/>. Acesso em: 01 set. 2013.
MOREIRAS, Antônio et al. **Funcionalidades.** Disponível em: <http://ipv6.br/entenda/funcionalidades/>. Acesso em: 01 set. 2013.
MOREIRAS, Antônio et al. **Transição.** Disponível em: <http://ipv6.br/entenda/transicao/>. Acesso em: 01 set. 2013.
NARTEN, T et al. **Neighbor Discovery for IP version 6 (IPv6).** RFC 4861. IETF. Disponível em: <http://tools.ietf.org/html/rfc4861>. Acesso em: 03 set. 2013.
SAAD, Redhwan; RAMADASS, Sureswaran; MANICKAM, Selvakumar. A Study on Detecting ICMPv6 Flooding Attack based on IDS. **Australian Journal Of Basic And Applied Sciences**, Penang, n. , p.175-181, jan. 2013.
SANKARAN, R. Migration of IPv6 - Security Issues. **International Journal Of Computer Trends And Technology**, Chennai, n. , p.567-572, abr. 2013.
SOURCEFORGE. **Easy-SEND.** Disponível em: <http://sourceforge.net/projects/easy-send/>. Acesso em: 6 jun. 2014.
TAIB, Abidah; BUDIARTO, Rahmat. Security Mechanisms for the IPv4 to IPv6 Transition. **The 5 Student Conference On Research And Development**, [s. L.], n. , p.1-6, dez. 2007.
TANENBAUM, Andrew. **Redes de Computadores.** 4. ed. Amsterdam: Campus, 2003.
THE HONEYNET PROJECT. **6Guard: a honeypot-based IPv6 attack detector.** Disponível em: <https://www.honeynet.org/node/944>. Acesso em: 11 jun. 2014.
THOMSON, S; NARTEN, T; JINMEI, J. **IPv6 Stateless Address Autoconfiguration.** RFC 4862. IETF. Disponível em: <http://tools.ietf.org/html/rfc4862.html>. Acesso em: 01 nov. 2013.