INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-GRANDENSE - IFSUL, CÂMPUS PASSO FUNDO
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

MAURÍCIO CORVELLO DA SILVA

SEGURANÇA EM IPV6: INTERCEPTAÇÃO E NEUTRALIZAÇÃO DE ATAQUE NDP SPOOFING

Roberto Wiest

PASSO FUNDO, 2013

MAURÍCIO CORVELLO DA SILVA

SEGURANÇA EM IPV6: INTERCEPTAÇÃO E NEUTRALIZAÇÃO DE ATAQUE NDP SPOOFING

Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-Rio-Grandense, câmpus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Roberto Wiest.

PASSO FUNDO, 2013

MAURICIO CORVELLO DA SILVA

SEGURANÇA EM IPV6: INTERCEPTAÇÃO E NEUTRALIZAÇÃO DE ATAQUE NDP SPOOFING

Trabalho de Conclusão de Curso aprovado em ____/____/____ como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet

Banca Examinadora:

_____

Prof. Roberto Wiest

Orientador

_____

Prof. Me. Carlos Alberto Petry

Professor convidado

_____

Prof. Jorge Luis Boeira Bavaresco

Professor convidado

_____

Prof. Dr. Alexandre Lazzaretti

Coordenador do Curso

PASSO FUNDO, 2013

Aos meus pais, irmã, namorada e professores
pela compreensão, estímulo e apoio
em todos os momentos.

“Nossas dúvidas são traidoras e nos fazem perder o que, com frequência, poderíamos ganhar, por simples medo de arriscar.”

Shakespeare, William.

RESUMO

A popularização da informática e a necessidade de comunicação em uma escala global ocasionou um crescimento nas requisições de endereços IPv4 (*Internet Protocol verison 4)*, acarretando na escassez de endereços, o que promovendo o surgimento de um novo protocolo de endereçamento, o IPv6 (*Internet Protocol version 6*). Além disto, aInternet apresenta riscos aos usuários que cada vez mais realizam transações *online*. Devido a estas praticidades, estas informações tendem a ser alvo de inúmeros ataques provenientes da Internet e de redes locais. O surgimento do protocolo IPv6 trouxe novas vulnerabilidades. Muitas destas vulnerabilidades só serão corrigidas com o passar do tempo e/ou com a migração massiva de toda a rede para o endereçamento IPv6. O objetivo deste trabalho é explorar as vulnerabilidades do protocolo de descoberta de vizinhança do IPv6, buscando a sua neutralização através do uso da segurança IPSEC (IP *Security Protocol*) em sistemas operacionais Linux.

Palavras-chave: IPv6; IPSEC; Linux; Segurança; Modo Transporte; *Stateful*.

ABSTRACT

The popularization of computers and the need for communication on a global scale caused an increase in requests for IPv4 addresses (Internet Protocol version 4), resulting in the shortage of this addresses, providing the emergence of a new addressing protocol, the IPv6 (Internet Protocol version 6). Furthermore, although versatile, the Internet offers risks for the user, who increasingly makes online transactions, whether they are banking or not, due to the convenience and practical that the Internet provides, and often, this information tend to be the target of several attacks across the Internet and local networks. The appearance of IPv6 brought new vulnerabilities. Many of these vulnerabilities can only be corrected with the passage of time or with the mass migration of all network to the IPv6 addressing. The objective of this paper is to explore the vulnerabilities of Neighborhood Discovery Protocol IPv6's, seeking its neutralization through the use of security IPSEC (IP Security Protocol) on Linux operating systems .

Keywords: IPv6; IPSEC; Linux; Security; Transport Mode; Stateful.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

O endereçamento IP (*Internet Protocol*) é uma forma de “identificação” de um equipamento conectado em uma rede de computadores. Segundo Scrimger et. al. (2002, p.107), essa identificação pode ser análoga ao endereço de uma residência, no qual as ruas são representadas pelos cabos e casas por um número identificador nesta rua, fazendo referência ao número de identificação de uma máquina. Com a expansão e a popularização da informática e *smartphones,* um número maior de endereços foram distribuídos até que eles chegassem ao fim.

No contexto atual da tecnologia da informação, o protocolo de endereçamento IPv4 (*Internet Protocol version 4*), provê cerca de 4 bilhões (2^{32}) de combinações possíveis de endereços disponíveis. Estes 4 bilhões de endereços, segundo Florentino (2012, p. 20), tornaram-se insuficientes. Embora haja medidas para tentar amenizar o problema com a falta endereços IPv4, tais como, protocolo NAT (*Network Address Translation*), divisões de endereços públicos e privados, classes IP, mesmo assim os estoques de endereços acabaram se esgotando completamente no ano de 2011.

Para suprir esta falta de endereços e atender às demandas por endereçamentos futuros, uma nova versão do protocolo de endereço foi criada, o IPv6 (*Internet Protocol version 6*). Entre as principais diferenças entre os protocolos, pode-se citar, segundo Scrimger et al. (2002, p. 576):

- ✓ Capacidade de endereço: IPv6 aumenta a capacidade de endereçamento de 32 bits para 128 bits, suportando maiores níveis hierárquicos de endereçamento;
- ✓ Cabeçalho: alguns campos do IPv4 foram excluídos ou deixados como opcionais para reduzir o custo de processamento durante a manipulação dos pacotes, dando maior flexibilidade e velocidade no processo de transmissão de dados;
- ✓ Capacidades de autenticação e controle de privacidade: o IPv6 inclui suporte a extensões para suportar autenticação e controle de integridade.

A escassez de endereços IPv4 levou ao desenvolvimento da versão seis do protocolo IP (IPv6), surgindo como alternativa para a falta de endereços disponíveis, além de promover um leque maior para a ocorrência de novos ataques. Para garantir que o acesso a sites, encaminhamento de e-mail, compartilhamento de informações e dispositivos sejam seguros, faz-se necessário a elaboração de uma política de segurança que promova a confiabilidade da rede, autenticidade de usuários e garanta a integridade das informações compartilhadas em uma determinada rede.

Diante deste cenário, o presente projeto irá explorar as vulnerabilidades encontradas nas trocas de mensagem do protocolo NDP (*Neigborhood Discovery Protocol*) com o objetivo de executar um ataque *man-in-the-middle*, em específico, NDP *spoofing* em uma rede IPv6 local além de propor uma forma para neutralizar os ataques, fazendo uso do conjunto de protocolos IPSEC (*IP Security Protocol*) em modo transporte. O qual será detalhado posteriormente.

1.1 MOTIVAÇÃO

A Internet proporciona inúmeras multifuncionalidades. Através dela, é possível realizar transações bancárias, troca de dados, compras, enfim, inúmeras informações confidenciais trafegam diariamente na Internet. Neste sentido, a nova versão do IP provê o surgimento de novas vulnerabilidades, comprometendo a privacidade, a autenticidade e a integridade do protocolo. Diante destas vulnerabilidades, torna-se necessário a implementação de mecanismos de segurança compatíveis com o protocolo IPv6, garantindo, então, os princípios básicos de segurança no contexto da tecnologia da informação: confidencialidade, integridade, disponibilidade e autenticidade.

Assim como o protocolo de endereçamento IPv4 torna viável a execução de ataques, o IPv6 também possui suas falhas de segurança. Neste trabalho de conclusão de curso, algumas falhas nas trocas de mensagens ICMP (*Internet Control Message Protocol*) do protocolo de descoberta de vizinhança do IPv6 foram exploradas, resultado em um ataque do tipo MITM (*Man-in-the-Midle*). O ataque é conhecido como NDP *Spoofing* e é executado a partir das trocas de mensagens do protocolo de descoberta de vizinhança, protocolo NDP (*Neighborhood Discovery Protocol*).

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O novo protocolo de endereçamento IPv6 possui um conjunto de protocolos de segurança IP nativo da terceira camada da arquitetura de protocolos TCP/IP (*Transport Control Protocol/Internet Protocol*), a camada de rede. O IPSEC (*Internet Protocol Security*) visa a garantir segurança IP em nível da camada de rede através de duas modalidades: túnel e transporte. Em vista disso, o presente trabalho visa a testar métodos de ataque de interceptação de dados, tais como o ataque *Man in the Middle*, em específico NDP *Spoofing*, em uma LAN IPv6 protegida com a segurança IPSEC (*Internet Protocol Security*), operando em modo transporte e fazendo uso de sistema operacional Linux.

1.2.2 Objetivos específicos

- ✓ Explorar as vulnerabilidades encontradas no protocolo NDP;
- ✓ Simular ataque de interceptação de dados, em específico, NDP *Spoofing*, em uma rede IPv6 local;
- ✓ Implementar módulo de segurança IP em modo transporte para neutralização do ataque;
- ✓ Contabilizar e avaliar resultados.

2 REFERENCIAL TEÓRICO

2.1 INTRODUÇÃO A REDES

A Internet teve seu início entre os anos 1969 à 1972 com o projeto ARPANET[1]. Segundo Morimoto (2008), esta rede era composta por quatro nós[2] principais interligados entre si por cabeamento telefônico adaptado para uso de dados, com velocidade de 50kbps. Esta rede era constituída desprovida de um nó central, ou seja, mesmo com o rompimento de um nó específico da rede, ela seria capaz de operar normalmente.

No ano de 1983, a rede ARPANET atingiu a marca de 562 hosts. Todos os hosts passaram a adotar a arquitetura de protocolos que regem a comunicação da Internet até os dias atuais, a arquitetura de protocolos TCP/IP (*Transmission Control Protocol*/*Internet Protocol*). Esta arquitetura é formada por endereços com tamanho igual a 32 bits, totalizando um pouco mais de quatro bilhões de endereços disponíveis (2^{32} = 4294967296). Na década de 80, não se tinha ideia que os mais de quatro bilhões de endereços iriam se esgotar um dia e muito menos se tinha a projeção de crescimento e importância que a rede primitiva, até então implementada pelo ARPANET, iria ter futuramente.

Surge, em 1974, a arquitetura TCP/IP (*Transmission Control Procol*/*IP*). De 1974até 2013, passaram-se 41 anos e a arquitetura de protocolos continua sendo referência no que se diz respeito a conectividade da Internet.

2.1.1 Arquitetura TCP/IP

Segundo Scrimger et al.(2002, p.33), o modelo de arquitetura TCP/IP é um conjunto de camadas, onde cada camada representa um grupo de tarefas específicas e facetas da comunicação. As implementações de protocolo, que são

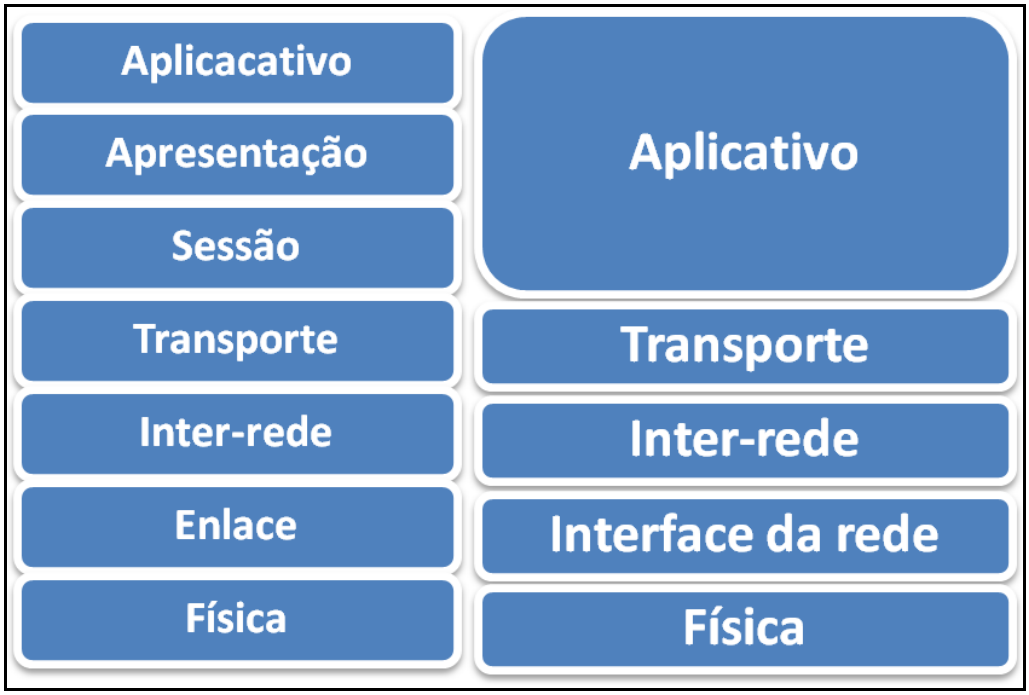
---

[1] *Advanced Research Projects Agency Net* - projeto precursor da Internet.
[2] Um nó ou nodos referem-se a quantidade de pontos interligados em uma rede de computadores.

uma combinação de *hardware* e *software*, na verdade realizam as funções associadas às camadas correspondentes.

A figura 1 ilustra a arquitetura de protocolos TCP/IP, composta por 5 camadas e sua relação com o modelo de referência OSI (*Open Systems Interconnection*), que é composto por 7 camadas.

Figura 1- Relação da arquitetura de protocolos TCP/IP e OSI.



Fonte: Scrimger et al., 2002, p.32

De acordo com a figura 1, pode-se observar que a arquitetura é composta pela camada de aplicativo, transporte, inter-rede, interface de rede e camada física. A seguir, estão descritas cada uma delas.

2.1.2 Camada aplicativo

A camada de aplicativo é a mais alta da arquitetura TCP/IP e, ao mesmo tempo, é de responsabilidade desta camada oferecer suporte aos protocolos necessários para fornecer os serviços de rede.

Como exemplo, pode-se citar os protocolos para transferências de arquivos FTP (*File Transfer Protocol*) e TFTP (*Trivial File Transfer Protocol*), protocolos de

troca de mensagem (Telnet[3]) e troca de e-mail (SMTP - *Simple Mail Transfer*), protocolo para gerenciamento de redes SNMP (*Simple Network Mangement Protocol*), entre outros.

2.1.2.1 Camada de transporte

A camada de transporte é responsável, principalmente, por fornecer uma interface entre as camadas mais baixas e a camada aplicativo. Além disso, a camada também provê a entrega dos dados do remetente ao seu respectivo destinatário.

Nessa camada, existem dois protocolos utilizados para realizar a transmissão dos dados, o protocolo TCP (*Transmission Control Protocol*) e protocolo UDP (*User Datagram Protocol*).

2.1.2.2 Camada de inter-rede

Ou interface da rede, é a camada responsável por identificar um dispositivo em uma rede local, fazendo uso de endereços físicos, gravados fisicamente em placas de rede, conhecidos como endereços MAC (*Media Access Control*). Para a transmissão de dados entre máquinas lotadas em redes diferentes, faz-se necessário anexar o endereço IP de destino, pois a identificação de *hosts* apenas pelo endereço MAC não é eficaz.

2.1.2.3 Camada interface de rede

As principais funcionalidades desta camada incluem:

- ✓ Identificar os dispositivos em uma rede local com o auxílio do endereço MAC;
- ✓ Organizar os *bits* recebidos a partir da camada física em quadros Ethernet[4];

---

[3] Protocolo de rede utilizado na Internet ou redes locais para facilitar a comunicação baseada em texto interativo usando uma conexão virtual.

[4] Quadros são pacotes de dados que são transmitidos entre computadores. Esse pacote possui um cabeçalho, informando a origem, o destino, quais dados estão sendo transmitidos.

- ✓ Converter os endereços de IP em endereços de rede loca e vice-versa;
- ✓ Detectar e notificar os erros;
- ✓ Controlar o fluxo de dados.

Entre os dispositivos de rede que fazem parte desta camada encontram-se as placas de rede, pontes. Além disto, é de responsabilidade desta camada controlar a velocidade da transmissão de dados entre os dispositivos.

2.1.2.4 Camada física

Camada mais baixo nível da arquitetura de protocolos TCP/IP. Nesta camada, ocorre a transmissão física dos dados. O caminho físico pelo qual os dados são transmitidos é denominado meio de transmissão (fibra óptica, ondas de rádio e outros). Além disso, todos os dados provenientes das camadas superiores são convertidos em uma série de *bits*, podendo assim ser enviados pelo meio físico.

2.2 IPV4 E O PROTOCOLO ARP (*ADDRESS RESOLUTION PROTOCOL*)

2.2.1 IPv4

Para se entender o endereçamento IP, pode-se fazer a seguinte analogia: “Um endereço IP é semelhante ao endereço utilizado para enviar uma carta pelo correio. O endereço deve apresentar todas as informações requeridas para que ocorra a entrega. Caso contrário, a carta não irá chegar ao seu destino correto. Um endereço incompleto ou incorreto afeta o endereçamento IP da mesma maneira que como afeta a entrega de uma carta utilizado para se enviar uma carta. [...]” (SCRIMGER et al., 2002, p.107). Ou seja, um endereço IP é análogo ao endereço de uma residência. A partir deste endereço é possível que uma determinada máquina receba todas as configurações necessárias para que ocorra a conectividade em uma rede de computadores possibilitando, assim, a ocorrência de trocas de mensagens entre as máquinas integrantes. Assim como o endereço de uma residência afeta o correto destino de uma carta, um endereço IP incorreto ou incompleto compromete toda a operabilidade de uma rede, interrompendo toda a comunicação entre os equipamentos da rede.

Um endereço IP é formado por quatro blocos de oito bits cada, denominado octeto. Os octetos são subdivido em duas partes: a primeira parte é utilizada para identificar um. A segunda é utilizada para identificar um *host* dentro desta rede. Endereços IPv4 possui endereços de tamanho igual a 32 bits. Inicialmente, estes endereços foram divididos em três classes de tamanhos fixos da seguinte forma, segundo IPv6 Brasil (2012):

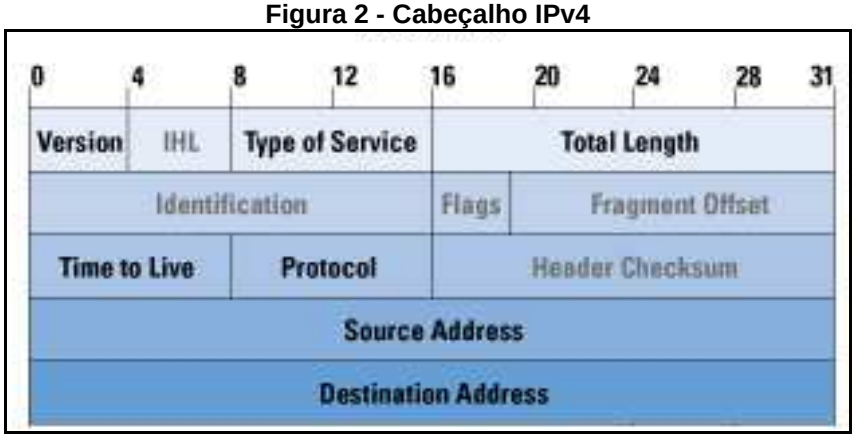
- ✓ Classe A: definia o bit mais significativo como 0, utilizava os 7 bits restantes do primeiro octeto para identificar a rede, e os 24 bits restantes para identificar o host. Esses endereços utilizam a faixa de 1.0.0.0 até 127.0.0.0;
- ✓ Classe B: definia os 2 bits mais significativo como 10, utilizava os 14 bits seguintes para identificar a rede, e os 16 bits restantes para identificar o host. Esses endereços utilizam a faixa de 128.0.0.0 até 191.255.0.0;
- ✓ Classe C: definia os 3 bits mais significativo como 110, utilizava os 21 bits seguintes para identificar a rede, e os 8 bits restantes para identificar o host. Esses endereços utilizam a faixa de 192.0.0.0 até 223.255.255.0;

A ideia de divisão dos endereços IPv4 em classes teve o intuito de prover uma distribuição de endereços mais flexível, atendendo redes de pequeno, médio e grande porte. Entretanto, essa divisão mostrou-se ineficiente com o passar do tempo.

Para mais informações sobre a arquitetura TCP/IP e endereçamento IPv4, consultar a bibliografia *TCP/IP a Bíblia*, Scrimger et al. (2002, p.49).

Os campos de um cabeçalho IPv4 irão definir o comportamento de um pacote IPv4 na transmissão dos dados. Este cabeçalho está representado pela figura 2.

Figura 2 - Cabeçalho IPv4



Fonte: Scrimger et al., 2002, p.117.

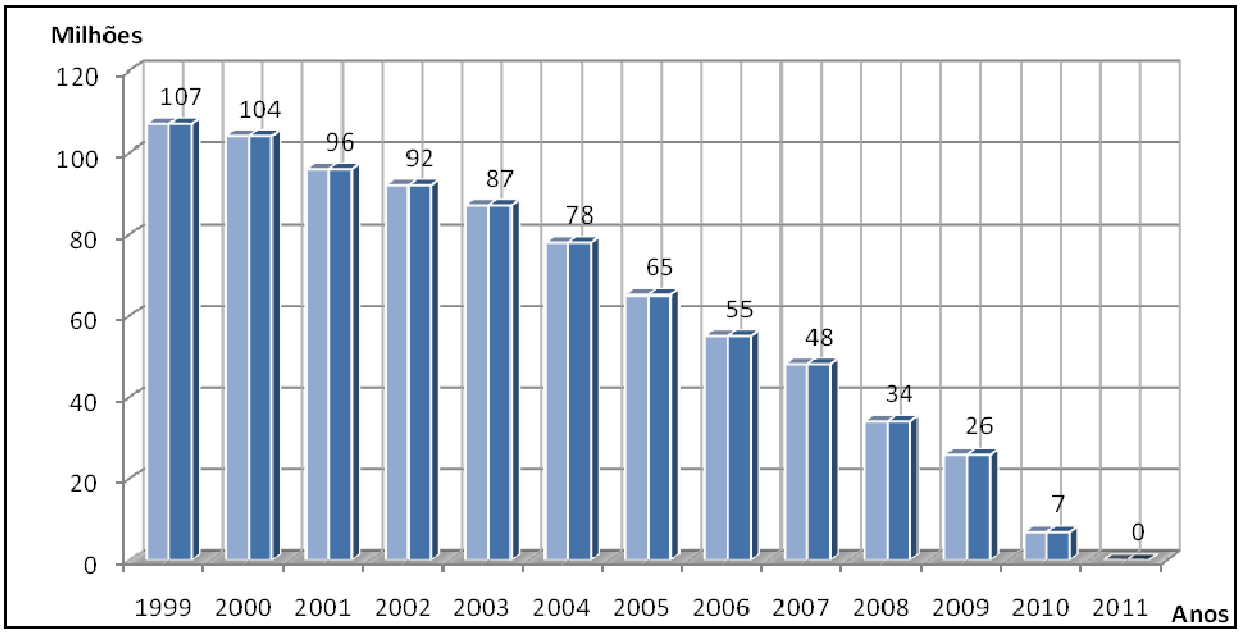
2.2.1.1 Esgotamento de endereços

O crescimento da Internet, com o passar dos anos, ocasionou uma divisão de três fases. Segundo Florentino (2012, p.17), a primeira fase foi marcada pela interconexão apenas de computadores à Internet. Atualmente, vive-se a segunda fase que, além de interconectar computadores, conecta pessoas através das redes sociais, em qualquer parte do mundo. A terceira fase é considerada a Internet das coisas, na qual, em um futuro próximo, aparelhos domésticos e a própria residência poderão ser interconectados e interagir entre si, necessitando de endereços IP para que ocorra a comunicação.

A figura 3 mostra a evolução do estoque de endereços IPv4 da IANA[5] (*Internet Assigned Number Authority*), na América Latina.

[5] Organização mundial que controla as concessões de endereços IPs no mundo, através de escritórios regionais, denominados RIR. Atualmente, existem cinco no mundo.

Figura 3 - Estoque de Endereços IPv4 IANA.



Fonte: Florentino, 2012, p. 21

Observa-se que, com o passar dos anos, a reserva de blocos de IPv4 foi decrescendo de forma gradativa, chegando ao esgotamento total no ano de 2011. Uma das principais causas do esgotamento do endereço IPv4 é dada pela má distribuição dos blocos IP entre os países. Conforme Florentino (2012), mais da metade dos endereços disponíveis foi destinado aos EUA, onde o *backbone*[6] da Internet foi criado e, a outra metade, foi dividida entre os demais países. Porém, mesmo se uma redistribuição de forma igualitária dos blocos de endereços entre os países fosse realizada, ainda existiram problemas referentes à falta de endereço. Neste mesmo contexto, outro grande motivo da escassez foi a aquisição, por empresas privadas, de blocos inteiros de endereços. Alguns blocos com o equivalente a 16 milhões de endereços disponíveis estão alocados para empresas que muitas vezes não os utilizam em sua totalidade.

De acordo com Medeiros (2010), uma pesquisa divulgada pelo IBGE (Instituto Brasileiro de Geografia Estatística), realizada pelo PNAD (Pesquisa Nacional por Amostra de Domicílio), mostra que no Brasil, entre os anos de 2005 e 2009, a Internet obteve um crescimento de 112%. Somente entre 2008 e 2009, o

[6] "Espinha dorsal" é uma rede principal por onde passam os dados dos clientes da internet. No Brasil, as empresas BrasilTelecom, Telecom Italia, Telefônica, Embratel, Global Crossing e a Rede Nacional de Ensino e Pesquisa (RNP) prestam esse serviço. Essa mesma rede também é responsável pelo envio e recebimento de dados entre grandes cidades e até entre Brasil e outros países.

crescimento chegou à casa de 21,5%, ou seja, 12 milhões de usuários, chegando a um total de 67,9 milhões de usuários. Mas, se o crescimento for analisado sob aspecto global, a rede Internet cresceu mais de 400% nos últimos anos.

2.2.2 Protocolo ARP

O protocolo ARP, segundo Scrimger et al.(2002, p.91), é um protocolo atuante na camada Inter-rede. Na realidade, o ARP separa o limite entre a camada de Enlace e a camada de Inter-rede. Este protocolo é utilizado para facilitar a substituição e a adição de novos dispositivos às redes, caso ocorra à troca de algum equipamento ou *hardware* de rede, sem ser necessária a alteração manual das tabelas de endereços existentes em cada máquina.

O funcionamento do protocolo ARP é bastante simples. Inicialmente, uma mensagem é encaminhada ao endereço de *broadcast*[7] da rede. Este envio tem por objetivo descobrir qual é o endereço físico (MAC) de um determinado *host* de rede a partir de um endereço IP. Tendo-se em vista que o pacote enviado é uma mensagem de *broadcast*, todos os dispositivos irão responder. Apesar disto, apenas o *host* para que a solicitação foi emitida reconhece seu endereço IP e responde com seu endereço IP acrescido do endereço físico atualizando, então, a *cache* de resolução de endereço. O ARP pode ser dividido em duas partes. Uma parte é utilizada para mapear um endereço de IP no endereço físico correspondente. A segunda é responsável por responder as solicitações de conversão de endereços recebida de outros *hosts.*

Segundo Scrimger et al. (2002, p.92), um endereço IP é armazenado em e, quando iniciado, o endereço é recuperado. Entretanto, pode-se encontrar sérios problemas em estações de trabalho que não possuem disco rígido, pois elas necessitam de um endereço IP para que seja possível recuperar seus arquivos de inicialização de sistema, a partir da rede, e operar normalmente. Para que isso seja possível, o protocolo RARP (*Reverse Address Resolution Protocol*) permite que estas máquinas sem disco rígido consigam obter endereços IP a partir de um servidor e, assim, possam recuperar seus arquivos de inicialização e, também,

[7] Broadcast é um endereço de rede que permite que a informação seja enviada para todos os nós de uma rede, em vez de um hospedeiro específico.

permite que as estações possam emitir solicitações para verificar os endereços IP de outros nós da mesma rede.

2.3 IPV6 E O PROTOCOLO NDP

Para solucionar o problema descrito na seção anterior e acompanhar o crescimento da Internet, o novo protocolo de endereçamento, IPv6, foi criado a partir de três projetos-base, de acordo com Centro de Estudos e Pesquisas em Tecnologias de Redes e Operações, segundo IPv6 Brasil (2012):

- ✓ CANTIP (*Common Architecture for the Internet*): foi concebido como um protocolo de convergência, para permitir a qualquer protocolo da camada de transporte ser executado sobre qualquer protocolo de camada de rede, criando um ambiente comum entre os protocolos da Internet;
- ✓ TUBA (*TCP and UDP with Bigger Addresses*): sua proposta era de aumentar o espaço para endereçamento do IPv4 e torná-lo mais hierárquico, buscando evitar a necessidade de se alterar os protocolos da camada de transporte e aplicação. Pretendia uma migração simples e a longo prazo, baseada na atualização dos hosts e servidores DNS, entretanto, sem a necessidade de encapsulamento ou tradução de pacotes, ou mapeamento de endereços;
- ✓ SIPP (*Simple Internet Protocol Plus*): concebido para ser uma etapa evolutiva do IPv4, sem mudanças radicais e mantendo a interoperabilidade com a versão 4 do protocolo IP, fornecia uma plataforma para novas funcionalidades da Internet, aumentava o espaço para endereçamento de 32 bits para 64 bits, apresentava um nível maior de hierarquia e era composto por um mecanismo que permitia “alargar o endereço”, chamado *cluster addresses*.

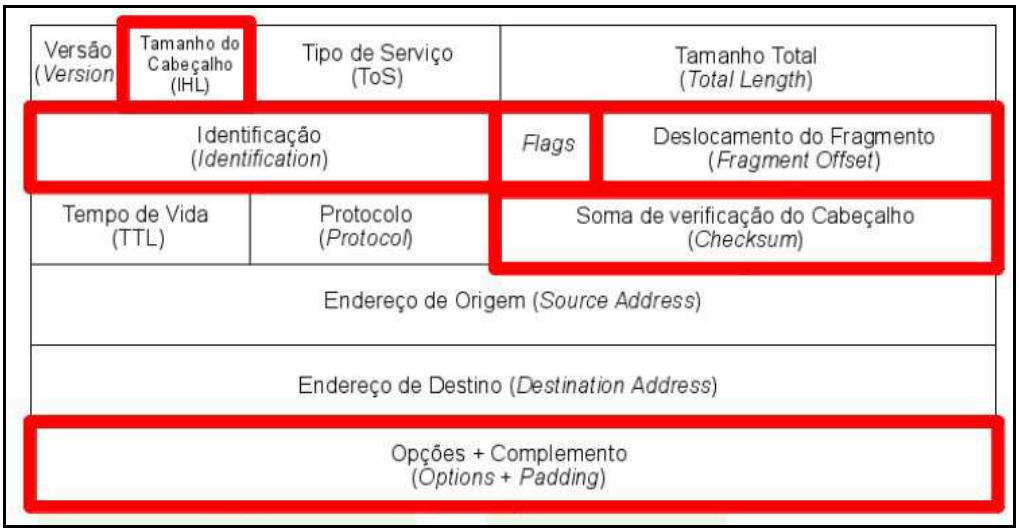
> As três propostas apresentavam problemas significativos. Portanto, a informação final para o novo protocolo baseou-se em uma versão revisada do SIPP, que passou a incorporar endereços de 128 bits, juntamente com os elementos de transição e autoconfiguração do TUBA, o endereçamento baseado no CIDR e os cabeçalhos de extensão. O CATNIP, por ser considerado muito incompleto, foi descartado.
> (IPv6 BRASIL, 2012).

O protocolo de endereçamento IPv6 possuí inúmeras diferenças em relação a sua versão anterior. Dentre elas, pode-se destacar:

- ✓ Novo formato de cabeçalho;
- ✓ Maior espaço de endereçamento;
- ✓ Autoconfiguração de endereços;
- ✓ Cabeçalho de expansão;
- ✓ Melhor suporte para QoS (*Quality of Service*);
- ✓ Novo protocolo para interação entre nós vizinhos;
- ✓ Suporte adequado para IPSEC;
- ✓ NAT (*Network Address Translation*) se torna desnecessário devido ao grande número de endereços disponíveis;

Um datagrama IPv6 é composto por um cabeçalho de 128 bits, o que possibilita, aproximadamente, cerca de 340 undecilhões (2^{128}) de combinações de endereços disponíveis, mudando-se completamente a forma de se planejar e implementar uma rede. O cabeçalho IPv6 possui um tamanho igual a 40 bytes. Embora o novo cabeçalho possua um tamanho quatro vezes maior em relação ao seu antecessor. Esse cabeçalho é considerado mais “enxuto” e com maior desempenho, pois muitos campos existentes no cabeçalho IPv4 foram removidos, de acordo com a figura 4.

Figura 4 - Remoção e/ou alteração cabeçalho IPv4.



Fonte: IPv6 Brasil, 2012.

A figura 4 faz a representação das alterações realizadas no cabeçalho IPv4. Os campos em destaques na figura representam as alterações e/ou remoções de

campos para dar origem ao cabeçalho IPv6. Segundo Florentino (2012, p. 22), o campo Tamanho do Cabeçalho foi removido, visto que o novo protocolo possui um tamanho fixo de 40 bytes. Já os campos de *Identification, Flags* e *Fragment Offset* foram removidos e deram lugar a um único campo, chamado *Fragmentation*. Ainda, o campo *Header Checksum* não está mais presente. Já os campos *Type of Service* cederam espaço aos novos campos *Traffic Class* e *Flow Lable*. O campo *Total Length*, foi substituído pelo campo *Payload Length* e o campo *Time To Life* (TTL), passou a se chamar *Hop Limit*. O campo Protocolo foi substituído pelo campo *Next Header.* Já os campos *Source Address* e *Destination Address* passaram de 32 para 128 bits.

Na figura 5, é ilustrado o cabeçalho IPv6.

Figura 5 - Cabeçalho IPv6.

Fonte: Florentino, 2012, p.27

O cabeçalho IPv6 possui a seguinte estrutura:

- ✓ *Version* (4 *bits*): indica a versão do protocolo IP;
- ✓ *Traffic Class* (8 *bits*): para fins de implementação de QoS;
- ✓ *Flow Label* (20 *bits*): prover serviços para aplicações que necessitem de alto desempenho;
- ✓ *Palyload Length* (16 *bits*) volume de dados em bytes que o pacote transporta;
- ✓ *Next Header* (8 *bits*): próximo protocolo encapsulado;
- ✓ *Hop Limit* (8 *bits*): semelhante ao TTL (*Time to live*) do IPv4, mas identifica o número de *hops* antes da transmissão do pacote IP*;*
- ✓ *Source Address* (128 *bits*): endereço IPv6 de origem;
- ✓ *Destination Address* (128 *bits*): endereço IPv6 de destino.

2.3.1 Configuração de endereços

Umas das grandes vantagens do protocolo IPv6 sobre o protocolo IPv4 está na característica de ser um protocolo *plug-and-play*[8], ou seja, o protocolo proporciona aos dispositivos a possibilidade de autoconfigurar seus endereços IP

A autoconfiguração destes endereços pode ser determinada a partir de dois mecanismos, segundo Thomson e Narten (1998).

2.3.1.1 Autoconfiguração *Stateful*

Neste mecanismo de autoconfiguração, os *hosts* obtêm as configurações de rede a partir de um servidor de DHCP (*Dynamic Host Configuration Protocol*) centralizado. Com a utilização de um serviço de DHCP, há uma base de dados (*leases*) com as informações dos endereços atribuídos aos *hosts* em uma determinada rede.

Este tipo de autoconfiguração permite, além de prover endereços, outras configurações, tais como *broadcast, gateway* de rede, servidores de DNS, etc. Percebe-se que este mecanimo de autoonfiguração é semelhante ao processo que ocorre em redes IPv4.

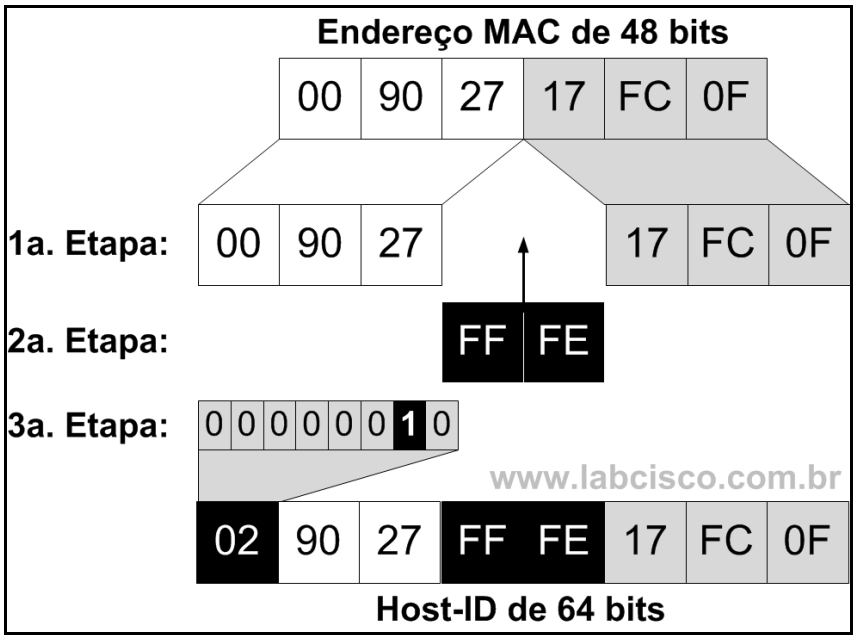
2.3.1.2 Autoconfiguração *Stateless*

Neste modo, um endereço é automaticamente gerado pelo próprio *host* através de uma combinação de informações locais e informações divulgadas pelos roteadores. Neste processo, os roteadores informam o prefixo de rede, responsável pela identificação da sub-rede, enquanto isso, os hosts fazem uso do endereço físico (MAC) e configuram seu endereço IP concatenado ao prefixo divulgado a seu endereço físico (MAC). Por ser um endereço global, é considerado único na Internet. Em outras palavras, este tipo de autoconfiguração é aquela em que o servidor não mantém um registro dos endereços atribuídos aos clientes. A figura 6 demonstra como é formado os endereços IPv6 em modo s*tateless*.

---

[8] Arquitetura *plug-and-play* tem por objetivo fazer com que o computador reconheça e configure automaticamente qualquer dispositivo ou protocolo, facilitando a expansão segura dos computadores e eliminando a configuração manual.

Figura 6 - Formação endereço *Stateless.*



Fonte: Brito, 2013.

Segundo Brito (2012), a figura exemplifica a formação de endereços IPv6 com a autoconfiguração s*tateless*. Pode-se observar que a formação destes endereços consiste na concatenação do endereço físico de 48 bits e pode-se separá-lo em dois blocos iguais de 24 bits. Após isso, o processo adiciona os algarismos hexadecimais FFFE (mais 16 bits) entre os dois blocos, expandido o endereço para 64 bits. Então, ocorre a inversão do sétimo bit para 1.

Uma *flag* indicando que o endereço é administrado localmente. Finalmente, após a expansão do endereço, ocorre a anexação do prefixo da rede ao identificador de *host* gerando, assim, um endereço global *unicast*.

2.3.2 ICMPv6 (*Internet Control Message Protocol Version* 6)

Devido à falta de um mecanismo facilitador de identificação de erros durante o processamento de pacotes no cabeçalho IP, o protocolo ICMP age para “cobrir” esta falha. Além de reportar erros no processamento de pacotes, realizar diagnósticos e enviar mensagens sobre as características da rede, a versão 6 do protocolo é incompatível com a versão do protocolo para redes IPv4.

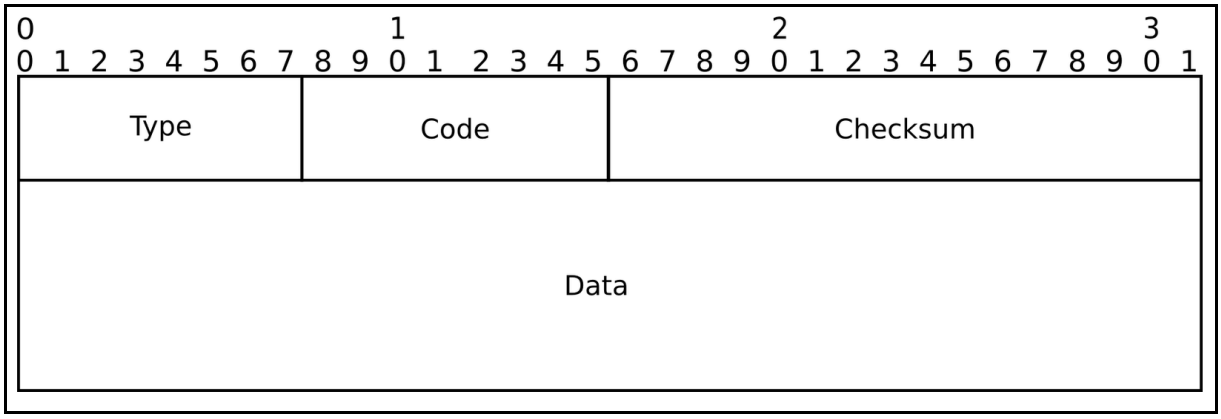
A incompatibilidade com o protocolo ICMPv6 em redes IPv4 é dada pela redução da multiplicidade de protocolos. Ou seja, segundo IPv6 Brasil (2012), alguns protocolos do endereçamento IPv4 foram removidos ou tiveram suas funcionalidades agregadas no ICMPv6, tais como:

- ✓ ARP (*Address Resolution Protocol*): objetiva mapear os endereços físicos através do endereço lógicos;
- ✓ RARP (*Reverse Address Resolution Protocol*): realiza o inverso do ARP, mapeando endereços lógicos para endereços físicos;
- ✓ IGMP (*Internet Group Management Protocol*): atua com o gerenciamento de membros de grupos *multicast*.

2.3.2.1 Formato ICMPV6

A estrutura de uma mensagem ICMP é bastante simples, composta por quatro campos básicos, conforme ilustrado na figura 6.

Figura 7 - Formato da mensagem ICMPv6.



Fonte: IPv6.br, 2012.

De acordo com a figura 6, segundo IPv6 Brasil (2012), a mensagem ICMPv6 é formada por:

- ✓ *Type* (8 bits): especifica o tipo da mensagem além de determinar o formato do corpo da mensagem . Um exemplo de seu uso é o valor 2, que representa uma mensagem "*Packet Too Big*";
- ✓ *Code* (8 bits): fornece informações adicionais sobre o motivo da mensagem. Como exemplo de uso, pode-se citar o envio de uma mensagem para informar a falha de conexão entre dois dispositivos, através da mensagem "*Destination Unreachable*", representado pelo valor 0; *Checksum* (16 bits): é utilizado para detectar dados corrompidos no cabeçalho ICMPv6 e em parte do cabeçalho IPv6.
- ✓ *Data*: faz referência ao tipo da mensagem, podendo ser desde diagnósticos de rede até erros. Seu tamanho é variável de acordo com a

mensagem, desde que não exceda o tamanho de MTU mínimo do IPv6 (1280 bits).

Devido ao grande número de informações que trafegam pelos nós de rede, as mensagens ICMP foram dividias em duas classes:

- ✓ Mensagens de erro: responsáveis por reportar erros durante a transmissão de um pacote tanto por nós destinam, como por nós intermediários.
- ✓ Mensagens de informação: proveem mensagens referentes a informações sobre os protocolos e suas funcionalidades aos nós de rede.

2.3.3 *Neighbor Discovery Protocol* (NPD)

O protocolo NDP concentra um conjunto de mensagens ICMPv6 e processos que irão determinar a forma como os nós de rede irão se intercomunicar em uma vizinhança. Além disso, o protocolo combina as funções de protocolos ARP, ICMP *Router Discovery* e ICMP *Redirect*. Segundo Santos et al. (2010), o protocolo é utilizado para:

- ✓ Determinar o endereço MAC dos nós da rede;
- ✓ Encontrar roteadores vizinhos;
- ✓ Determinar prefixos e outras informações de configuração da rede;
- ✓ Detectar endereços duplicados;
- ✓ Determinar a acessibilidades dos roteadores;
- ✓ Fazer o redirecionamento de pacotes;
- ✓ Efetuar a autoconfiguração de endereços.

É de responsabilidade do protocolo o processo de autoconfiguração de nós e a transmissão de pacotes. No caso da autoconfiguração, existem três funcionalidades importantes, segundo site IPv6.br (2013):

- ✓ *Parameter Discovery*: atua na descoberta por um nó de informações sobre o enlace (como MTU) e sobre a Internet (como *hop limit*);
- ✓ *Address Autoconfiguration*: trabalha com a autoconfiguração *stateless*[9] de endereços nas interfaces de um nó;

[9] Autoconfiguração que permite a aquisição de endereços globais sem o uso de DHCP.

- ✓ *Duplicate Address Detection*: utilizado para descobrir se o endereço que se deseja atribuir a uma interface já está sendo utilizado por outro nó na rede.

Caso a utilização do protocolo venha a ser para a realização de transmissão de dados, tem-se, segundo site IPv6.br (2012):

- ✓ *Router Discovery*: atua na descoberta de roteadores pertencentes ao enlace;
- ✓ *Prefix Discovery*: encarregado pela busca de prefixos de redes, cuja finalidade é decidir para onde os pacotes serão enviados, se para um roteador ou direto para um nó;
- ✓ *Address Resolution*: descobre o endereço MAC através de um endereço lógico IPv6.
- ✓ *Neighbor Unreachability Detection*: permite que os nós descubram se um vizinho é ou se continua alcançável, uma vez que problemas podem acontecer tanto nos nós como na rede;
- ✓ *Redirect*: permite ao roteador informar ao nó uma rota melhor ao ser utilizado para enviar pacotes a determinado destino;
- ✓ *Next-Hop Determination*: algoritmo utilizado para mapear um endereço IP de destino em um endereço IP de um vizinho para onde o tráfego deve ser enviado.

2.3.3.1 Mensagens NDP

Dentre as várias mensagens ICMPv6 trocadas em um segmento de rede local, Florentino (2012, p.48) destaca cinco delas são responsáveis pelo funcionamento do protocolo NDP.

- ✓ RS (*Router Solicitation*);
- ✓ RA (*Router Advertisement*);
- ✓ NS (*Neighbor Solicitation*);
- ✓ NA (*Neighbor Advertisement*);
- ✓ *Redirect*.

2.3.3.2 RS

Mensagem enviada com o objetivo de descobrir a presença de um *router* IPv6 ativo no segmento de rede. Estas mensagens partem de uma estação de trabalho, com a finalidade de receber as configurações necessárias de rede para que ocorra a conexão. Estas informações podem ser DNS, prefixo de um roteador em uma rede local, etc. Com base nessas informações recebidas, o *host* terá conectividade com a Internet sem a necessidade de um servidor DHCPv6.

2.3.3.3 RA

Mensagens deste tipo podem ser enviadas em duas situações. A primeira pode ser apenas uma responder uma solicitação RS. A segunda ocorre quando está relacionada ao envio de anúncios de roteadores ativos. Mesmo que as solicitações RS sejam bloqueadas, o roteador pode “revelar sua existência” para as estações de trabalho por meio de RAs.

2.3.3.4 NS

Mensagem enviada por um determinado *host* com o objetivo de encontrar um endereço MAC correspondente a um endereço IPv6 e/ou podem ser utilizadas para mapear a unicidade de um determinado endereço. Pode-se perceber que mensagens deste tipo são semelhantes ao ARP *request* do IPv4.

2.3.3.5 NA

Mensagens deste tipo são enviadas em resposta à solicitação NS ou simplesmente para informar a mudança de endereço de um determinado *host*.

2.3.3.6 *Redirect*

Segundo ipv6.br(2013)Mensagens enviadas por roteadores para informar ao nó solicitante de uma comunicação uma melhor opção de caminho para ser utilizada. Em outras palavras, ele envia o endereço do próximo salto que deve ser usado para encaminhar pacotes quando se comunicar com aquele determinado destino.

2.4 SEGURANÇA DE REDES

No contexto da Tecnologia da Informação, segundo ISO (*International Standardization Organization*) apud Souza, pode-se entender por Segurança em Redes de Computadores todo o conjunto de ações e recursos utilizados para diminuir as vulnerabilidades em uma determinada rede e/ou sistemas computacionais.

Ao se projetar uma política de segurança em uma rede de computadores e/ou quaisquer sistemas que operem nesta mesma rede, deve-se levar em consideração alguns conceitos que, segundo Stallings (2008, p. 5) irão prover uma correta projeção de políticas de segurança e, consequentemente, diminuir as vulnerabilidades em um ambiente computacional:

- ✓ Ataques de Segurança: qualquer ação que comprometa a segurança da informação pertencente a uma organização;
- ✓ Mecanismo de Segurança: um processo que é projetado para detectar, impedir ou permitir a recuperação de um ataque à segurança;
- ✓ Serviço de segurança: um serviço de processamento ou comunicação que aumenta a segurança dos sistemas de processamento de dados e as transferências de informações de uma organização.

O CGI.br (2006, p.1) afirma que, para um computador e/ou sistema computacional ser seguro, deve apresentar confiabilidade, integridade e disponibilidade aos usuários. Além disso, a segurança deve ser implementada obedecendo às seguintes propriedades, segundo Kurose e Ross (2010, p. 493):

- ✓ Confiabilidade: somente o remetente e o destinatário pretendido devem poder entender o conteúdo da mensagem transmitida;
- ✓ Autenticação do ponto final: o remetente e o destinatário precisam confirmar a identidade da outra parte envolvida na comunicação – confirmar que a outra parte realmente é quem alega ser;
- ✓ Integridade da mensagem: mesmo que o remetente e o destinatário consigam se autenticar reciprocamente, eles também querem assegurar que o conteúdo de sua autenticação não seja alterado, por acidente ou por má intenção, durante a transmissão;

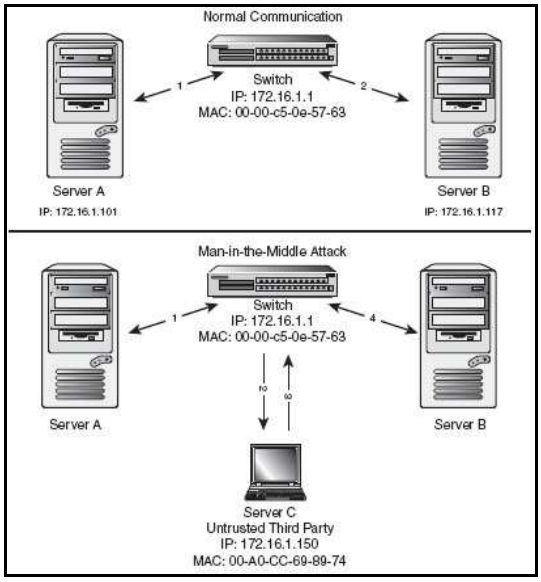
- ✓ Segurança operacional: hoje quase todas as organizações possuem redes conectadas à Internet pública. Essas redes podem ser comprometidas potencialmente por atacantes que ganham acesso a elas por meio de Internet pública.

Em redes IPv6, muitos autores o consideram menos seguro que a sua versão antecessora. Apesar do curto espaço de tempo de existência do protocolo IPv6, ataques às vulnerabilidades estão ocorrendo desde o ano 2000. As maiores vulnerabilidades encontradas no IPv6 são decorrentes do IPv4 que, durante o seu desenvolvimento, grande parte de seu cabeçalho foi mantida e agregadas às novas vulnerabilidades do IPv6. Sabe-se também que, segundo Florentino (2012, p.79), não há uma data limite para que o protocolo IPv4 caia em desuso. Portanto, há a necessidade de que ambos os protocolos, IPv4 e IPv6, coexistam e operem paralelamente, dando brechas a ataques, principalmente, do tipo DoS.

2.4.1 Ataque MITM (MAN IN THE MIDDLE)

Ataque *man-in-the-middle*, ou “homem de meio” consiste em um ataque de interceptação de dados, quando uma máquina estabelece uma conexão com outras duas ou mais máquinas (vítimas), repassando todas as mensagens. O atacante se insere no meio da comunicação entre as vítimas e intercepta todas as mensagens trocadas entre as mesmas, sendo capaz de rejeitar, retransmitir ou modificar as mensagens em tempo real. Na figura 7, pode-se observar um exemplo de como esse tipo de ataque acontece.

Figura 8 - Ataque MITM em uma rede local.



Fonte: Hackpittsburgh, 2011.

Em se tratando de ataques MITM, após o atacante armazenar as informações capturadas, elas são redirecionados automaticamente, em tempo real, ao seu destino, representado pelo *Server B*. Com isso, pode-se observar que além do "envenenamento" das informações durante a transmissão de dados, as ações ocorreram transparentes para as vítimas, representadas pelo *Server A* e *B*, logo, eles não saberão que, nesse meio de campo, as informações foram redirecionadas, automaticamente, para um terceiro computador, atacante, e repassadas para seu destino, como se nada tivesse ocorrido.

Pode-se citar dois exemplos deste ataque: o *Address Resolution Protocol Spoofing* (ARP *Spoofing*) e o *Neighbor Discovery Procolo Spoofing* (NPD *Spoofing*). Em redes IPv4, são exploradas as mensagens ARP *Request* e ARP *Reply* para a execução dos ataques. Da mesma forma ocorre em redes IPv6, em que são exploradas as falhas nas trocas de mensagens entres os nós de redes, através das mensagens NDP *Solicitation* e NDP *Advertisement*.

2.4.2 ARP Spoofing

Ataques deste tipo, em muitos casos, tendem no envio blocos de consultas e/ou respostas ARP falsos e, por consequência, acabam interceptando todo o tráfego de rede ou, ainda, estes ataques podem "personificar" suas máquinas e modificar o fluxo de dados da rede. Isto é conhecido como ARP *spoofing*. Segundo

Goding e Carnut (2003), ataques deste tipo são realizados a partir do “envenenamento” da *cache* de resolução de endereços utilizada pelo protocolo ARP.

Pelo fato de as mensagens ICMP (*Internet Control Message Protocol*) ARP *request* e ARP *replay* não terem nenhuma correlação entre si, os ataques ocorrem explorando estas trocas de mensagens ICMP e, por consequência, as máquinas aceitam qualquer mensagem do tipo ARP *replay*, com a finalidade de atualizar sua *cache* de endereços resolvidos. A partir de então, ao enviar uma mensagem ICMP do tipo ARP *replay* falsa e, em seguida, ocorrer a atualização da *cache* de endereços, pode-se redirecionar todo o fluxo de rede. Ou seja, através desta resposta falsa é possível, por exemplo, forçar um roteador, *switch* e/ou aplicação a desviar o tráfego para o atacante, podendo assim rejeitar mensagens, resultando em um ataque de DoS; retransmitir mensagens ao destino legítimo enquanto armazena as informações para posterior análise, resultando em um *sniffer* de ativo; ou modificá-las em tempo real com a finalidade de quebrar a criptografia de um determinado pacote IP, resultando em um ataque *man-in-the-middle*.

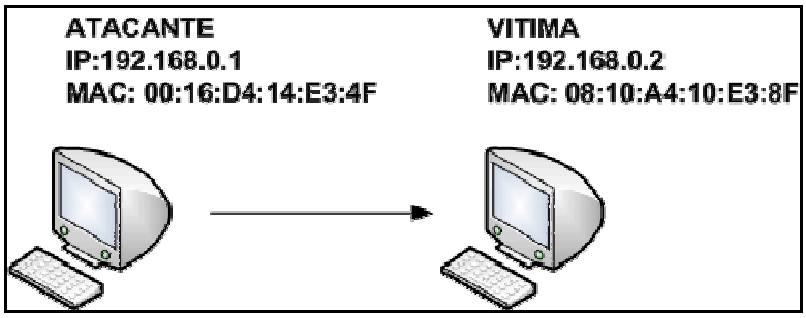
O ataque ARP *spoofing* pode ser executado de duas formas:

- ✓ Unidirecional: ataque é realizado enviando um ARP *replay* falso para uma das vítimas. O atacante, por sua vez, irá interceptar os dados em uma única direção, ou seja, irá interceptar os dados somente da vítima que possuir a ARP *cache* envenenada;
- ✓ Bidirecional: ataque realizado em todas as vítimas que possuírem a ARP *cache* envenenada, desta forma, o atacante poderá desviar todo e qualquer tráfego entre as vítimas.

Este tipo de ataque é bastante funcional em redes com padrão *ethernet*, pois o ataque ocorre de forma ativo, forçando as vítimas e enviarem os dados para o endereço MAC do atacante. A seguir, a sequência de figuras irá exemplificar a ocorrência de um ataque ARP *spoofing* em uma rede local.

Na figura 9, o atacante envia uma *echo request* com um endereço de um host qualquer, por exemplo 192.168.0.10, com a finalidade de atualização da *cache* de endereços.

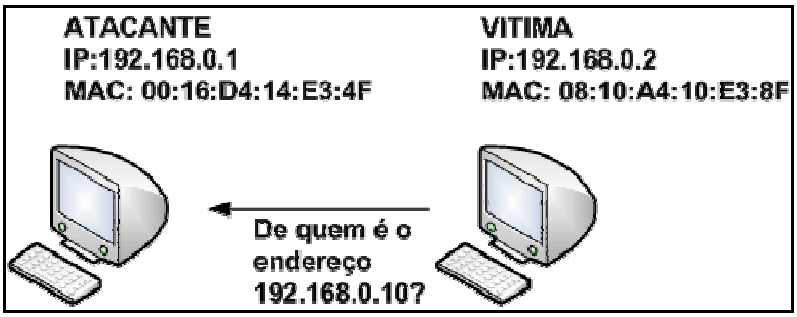
Figura 9 - Ataque *echo request* falso.



Fonte: do Autor.

A vítima, por sua vez, envia um ARP *request*, com finalidade de descobrir qual o endereço físico pertencente ao endereço lógico 192.168.0.10, ilustrado na figura 10.

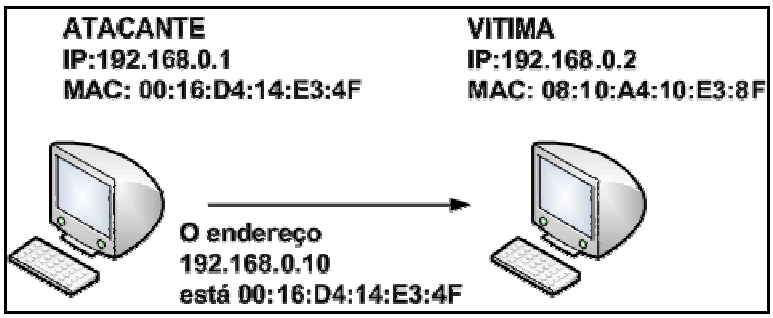
Figura 10 - Envio de ARP *request* forjado.



Fonte: do Autor.

Na figura 11, o atacante responde à vítima com uma ARP *replay*, informando que o endereço 192.168.0.10 pertence ao seu endereço físico. Sendo assim, o atacante serve como *gateway* para a vítima, recebendo, assim, todos os dados que trafegam pela rede.

Figura 11 - ARP replay.

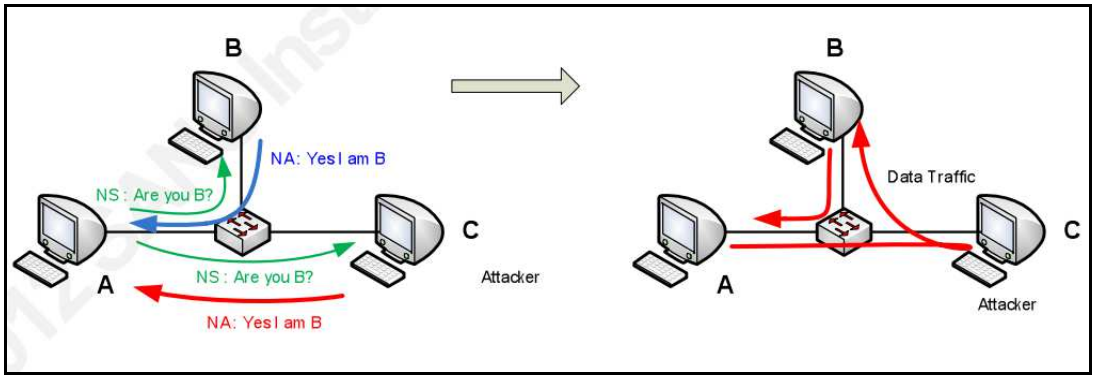


Fonte: do Autor.

2.4.3 NDP Spoofing

Semelhante ao ARP *spoofing*, o NDP *spoofing* é um ataque nativo de Inter-rede IPv6, em que ambos os tipos de ataques exploram as falhas existentes nas traduções de endereços. A principal diferença está relacionada na forma em que os dois protocolos tratam a tradução dos endereços. No protocolo IPv6, a tradução de endereços ocorre com o auxílio de trocas de mensagens NS e NA, conforme ilustrado na figura 12.

Figura 12 - NDP Spoofing.



Fonte: Pilihanto, 2011, p.56.

A estratégia de ataque é a mesma que a do ARP *Spoofing*, em que um atacante “envenena” a *cache* NDP, explorando as vulnerabilidades das mensagens ICMP trocadas entre emissor e receptor, em específico as mensagens NS e NA. Caso este “envenenamento” seja bem sucedido, o atacante, representado pelo computado C da figura 12, poderá capturar todo o tráfego da rede que seria encaminhado para do computador A para o computador B.

2.4.4 IPSEC (IP *Security Protocol*)

Um dos principais problemas relativos ao protocolo IP está relacionado à segurança, já que no momento de sua “concepção” este aspecto não foi considerado importante. Na década de 80, quando o protocolo foi criado, não se tinha a ideia da expansão que a Internet teria, muito menos se sabia a quantidade de problemas que a falta de segurança viria a causar devido a má projeção deste quesito. Os pacotes IPs podem ser facilmente capturados, interceptados, modificados e retransmitidos de forma transparente, ou seja, sem que o usuário perceba a ocorrência destes eventos em uma rede local e/ou na Internet.

Na prática, de acordo com Carissimi, Rochol e Granville (2009, p. 366), os campos de endereços IP origem e destino podem ser alterados facilmente por qualquer programa malicioso, comprometendo um dos três pontos já citados anteriormente ou, até mesmo, todos. Desta forma, não há uma garantia de que o pacote IP foi entregue de fato à máquina correspondente ao IP destino, o que compromete a autenticidade do pacote. Por outro lado, há o comprometimento da integridade do pacote IP, ocasionado pela modificação do cabeçalho e/ou conteúdo do pacote IP transmitido. Por último, há o comprometimento da confidencialidade do pacote pelo fato de o mesmo ser transmitido em texto claro, ou seja, sem criptografia, por uma rede local ou pela Internet.

O IPSEC, segundo Carissimi, Rochol e Granville (2009, p.366), “é um conjunto de protocolos definidos pela IETF (*Internet Engineering Task Force*) para prover segurança na camada de rede”. Em outras palavras, o IPSEC fornece segurança à camada de rede e às camadas superiores, fornecendo uma maior segurança aos pacotes IP durante a transmissão e, inclusive, aos seus dados. Além disso, o conjunto de protocolos IPSEC é utilizado para realizar a autenticação do emissor e receptor das mensagens trocadas em uma rede, o que garante a integridade e a privacidade das informações trafegadas entre as pontas.

O IPSEC estabelece uma conexão lógica, fazendo uso de um protocolo de sinalização para “negociar” as trocas de mensagens entre os pares, denominado SA (*Security Association*). A conexão SA é unidirecional, ou seja, uma SA para cada conexão. Caso uma conexão em que seja realizada a troca de dados e/ou arquivos, duas SAs são criadas, uma em cada direção. Conforme Carissimi, Rochol e Granville (2009, p. 367), uma conexão SA é identificada por três componentes: identificador de conexão, tipo de protocolo e endereço IP origem.

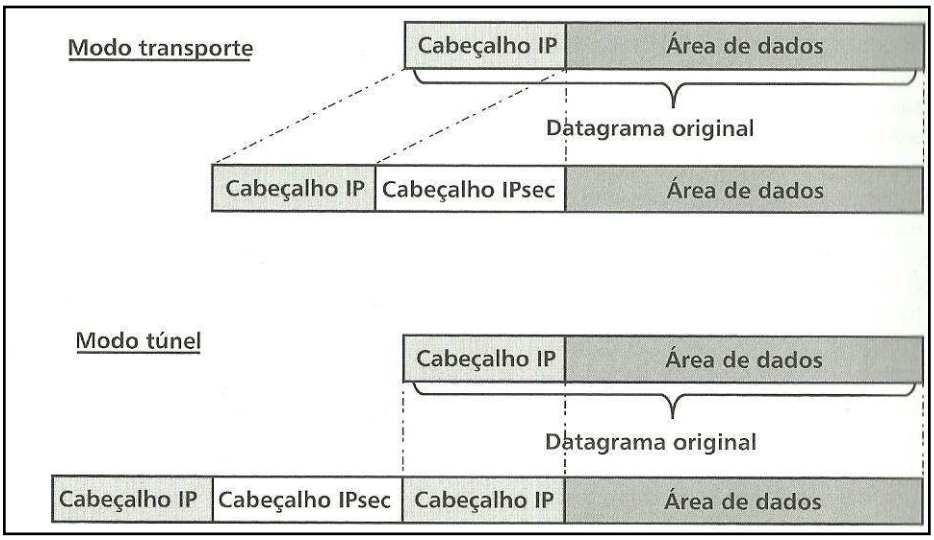
O primeiro, o identificador de conexão, denominado SPI (*Security Parameter Index*) é responsável pela identificação de uma SA, esse parâmetro é definido durante a negociação da conexão. Todos os membros de uma SA devem conhecer o SPI correspondente e usá-lo durante a comunicação.

O segundo componente é o tipo de protocolo utilizado para a obtenção da segurança, podendo ser referente à autenticação através do protocolo AH (*Authentication Header*) ou ao encapsulamento com o protocolo ESP (*Encapsulating*

Security Payload). Por fim, o componente endereço IP origem faz referência ao endereço IP de origem de uma conexão SA.

O protocolo IPSEC atua de duas formas: modo transporte e modo túnel, conforme representado na figura 13.

Figura 13 - IPSEC em modo transporte e túnel.



Fonte: CARISSIMI, ROCHOL e GRANVILLE, 2009, p. 368.

Pode-se perceber que na modalidade transporte do IPSEC, o cabeçalho IPSEC é inserido logo após o cabeçalho IP do pacote original. O campo *type header* é modificado para identificar que a área de dados deve ser tratada pelo IPSEC. Por sua vez, o cabeçalho IPSEC contém as informações referentes ao SPI.

Já no IPSEC em modo túnel o pacote IP original, cabeçalho e área de dados são tratados como dados e prefixado por um cabeçalho IPSEC. Esse novo datagrama, cabeçalho IPSEC seguido do pacote original, é encapsulado em um novo pacote IP. Em outras palavras, pode-se dizer que um pacote IP carrega um pacote IPSEC que, por sua vez, possui o pacote original.

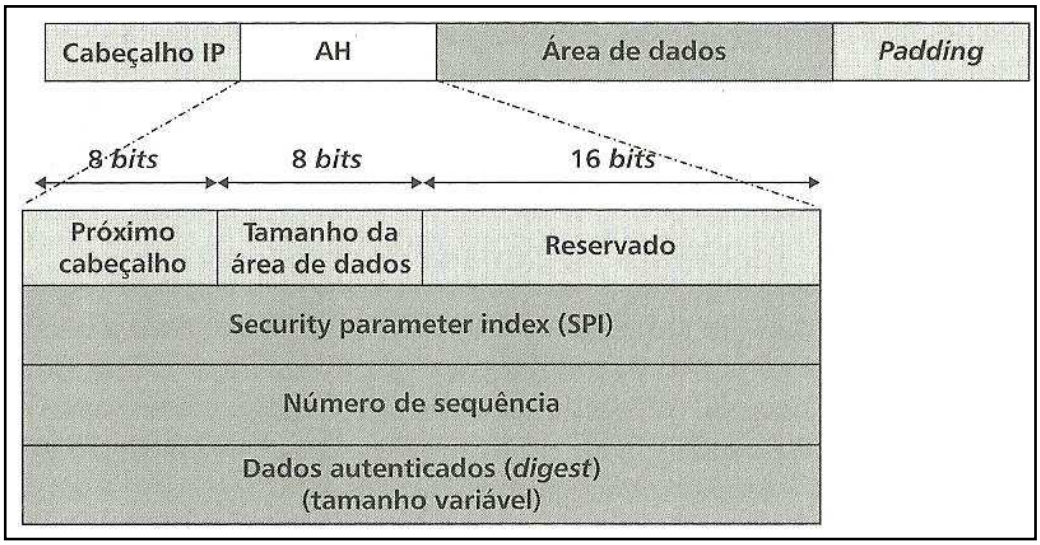
Além disso, o protocolo IPSEC anexa outros dois cabeçalhos de extensão a um pacote IP. Cada um destes cabeçalhos faz referência ao protocolo *Authentication Header* (AH), garantindo a autenticidade e a integridade, mas não garante a privacidade do pacote IP a ser transmitido. Por outro lado, o protocolo *Encapsulating Security Playload* (ESP) fornece autenticidade, integridade e privacidade.

Os protocolos AH e ESP serão descritos nas próximas seções.

2.4.4.1 AH (Authentication Header)

Segundo Carissimi, Rochol e Granville (2009, p. 368), o protocolo AH possui um cabeçalho composto por seis campos e, ao ser implementado, o mesmo se insere entre o cabeçalho IP e a área de dados do pacote IP original, conforme ilustrado pela figura 14.

Figura 14 - Protocolo AH no IPSEC modo transporte.



Fonte: CARISSIMI, ROCHOL e GRANVILLE, p.369, 2009.

Cada campo do cabeçalho AH possui as seguintes funções:

- ✓ Próximo cabeçalho: identifica o tipo de PDU[10] (*Protocol Data Unit)* transportada na área de dados, podendo ser destinada aos protocolos ICMP, UDP, TCP ou até mesmo a todos;
- ✓ Tamanho da área de dados (*playload*): fornece o tamanho do cabeçalho AH e não o tamanho da área destinada aos dados;
- ✓ Reservado: campo é destinado para uso futuro;
- ✓ *Security parameter index*: desempenha o papel de identificador de uma conexão SA;
- ✓ Número da sequência: identifica de forma única cada um dos pacotes IP enviados por uma conexão SA. Caso haja uma retransmissão desse pacote, um novo número é gerado e anexado a este campo;

[10] Unidade de Dados de Protocolo descreve um bloco de dados que é transmitido entre duas instâncias da mesma camada.

- ✓ Dados autenticados: destinado à assinatura digital gerada por meio da combinação da função *hash* com uma chave simétrica negociada entre as partes durante a conexão segura.

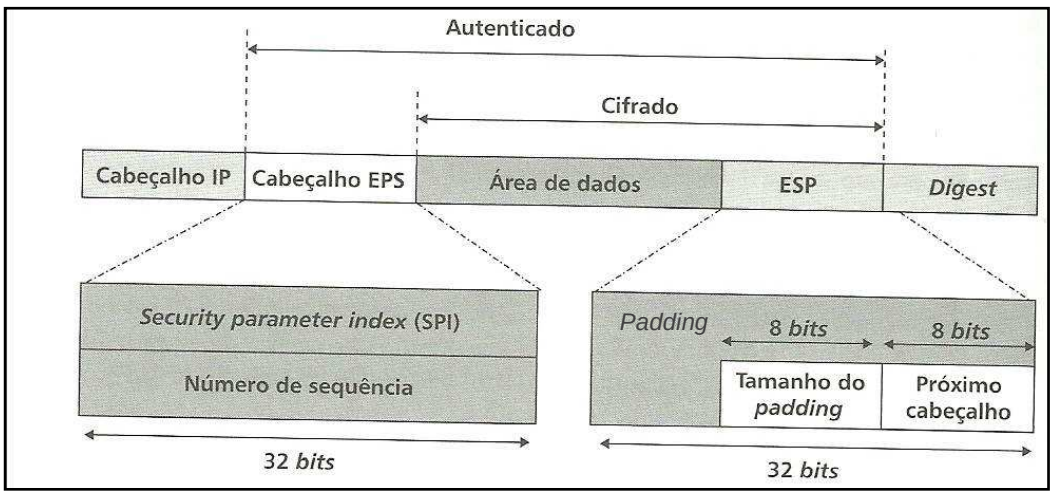
A partir do cabeçalho AH preenchido e anexado ao cabeçalho IP do pacote original, o AH obtém a integridade de seus pacotes IP transmitidos através da utilização de uma assinatura digital. Para que seja garantida a integridade dos pacotes após a transmissão dos mesmos, o destinatário calcula a assinatura digital e, se obtiver exatamente igual ao resultado recebido, tem-se a comprovação de que o pacote não foi alterado durante a transmissão.

2.4.4.2 ESP (*Encapsulating Security Payload*)

De acordo com Carissimi, Rochol e Granville (2009, p. 369), o protocolo ESP adiciona o seu cabeçalho entre o cabeçalho IP e a área de dados do pacote IP original e, ao final do mesmo, adiciona o rótulo *trailer* ESP e a autenticação HMAC (*Hash-based Message Authentication Code*). Toda a região delimitada pelo cabeçalho ESP e o HMAC serão cifrados.

A figura 15 traz os campos do cabeçalho ESP em um pacote IP a ser transmitido.

Figura 15 - Cabeçalho protocolo ESP em modo transporte.



Fonte: CARISSIMI, ROCHOL e GRANVILLE, p.370, 2009.

Carissimi, Rochol e Granville (2009, p. 369) esclarecem que o protocolo ESP possui seu cabeçalho formado por dois campos, cada um contendo 32 bits: SPI (*Security Parameter Index*) e número de sequência. Ambos os campos possuem exatamente a mesma função que no protocolo AH. O primeiro é responsável pela a

identificação dos pacotes enviados; e o segundo é responsável pela identificação única destes pacotes enviados.

Por sua vez, o rótulo *trailer* ESP possui outros três campos: *bits* de preenchimento (*padding*), tamanho de área de preenchimento e próximo cabeçalho. O campo *padding* é utilizado para fixar o tamanho do rótulo ESP igual e múltiplo de 4 *bits.* O tamanho deste campo *padding*.

O campo referente ao próximo cabeçalho tem sua função semelhante a do protocolo AH, sendo responsável por identificar o tipo da PDU transportada na área de dados. Por fim, o campo de dados autenticados, *Digest*, assim como no protocolo AH, é resultante da aplicação da função de *hash* sobre todas as partes do datagrama entre o cabeçalho ESP e o rótulo *trailer* ESP.

3 DESCRIÇÃO METODOLÓGICA E RESULTADOS

Este capítulo irá abordar o método utilizado para a realização das atividades, o ambiente utilizado para realização dos testes, as ferramentas utilizadas, bem como os resultados alcançados.

O método realizado será embasado na utilização de quatro máquinas Linux, sendo um servidor, um atacante e dois clientes. Os testes de interceptação de dados serão executados através da ferramenta Parasite6, explorando as conexões FTP (*File Transport Protocol*) entre as máquinas clientes com a máquina servidora.

As ferramentas fazem referência às tecnologias utilizadas nos testes de laboratório, possibilitando a análise e comparação dos resultados obtidos durante a realização dos testes.

Como ambiente de teste, optou-se pela arquitetura cliente/servidor em um ambiente virtualizado. A arquitetura escolhida provê uma estrutura de aplicação distribuída, ou seja, ocorre a distribuição de serviços (máquinas servidoras) entre as demais máquinas em uma rede de computadores. Além disto, as máquinas que irão requerer os serviços são designadas clientes.

A utilização da virtualização das máquinas foi escolhida pelo fato de que é possível permitir a execução de vários sistemas operacionais e seus respectivos softwares a partir de uma única máquina, seja ela um desktop convencional ou um potente servidor, oferecendo resultados como qualquer outro computador. A virtualização do cenário de teste tem por finalidade a simulação de um ambiente real, o que irá prover a capacidade de utilizar diversos sistemas e aplicativos sem a necessidade de acesso físico à máquina na qual estão hospedados, promovendo praticidade para realização de testes.

Segundo Alecrim (2012), dentre as vantagens da virtualização, estão:

- ✓ Gerenciamento centralizado: dependendo da solução de virtualização utilizada, fica mais fácil monitorar os serviços em execução, já que o seu gerenciamento é feito de maneira centralizada;

- ✓ Implementação mais rápida: dependendo da aplicação, a virtualização pode permitir sua implementação mais rápida, uma vez que a infraestrutura já está instalada;
- ✓ Uso de sistemas legados: pode-se manter em uso um sistema legado, isso é antigo, mas ainda essencial às atividades da companhia, bastando destinar a ele uma máquina virtual compatível com o seu ambiente;
- ✓ Diversidade de plataformas: pode-se ter uma grande diversidade de plataformas e, assim, realizar testes de desempenho de determinada aplicação em cada uma delas, por exemplo;
- ✓ Ambiente de testes: é possível avaliar um novo sistema ou uma atualização antes de efetivamente implementá-la, diminuindo significativamente os riscos inerentes a procedimentos do tipo;
- ✓ Segurança e confiabilidade: como cada máquina virtual funciona de maneira independente das outras, um problema que surgir em uma delas – como uma vulnerabilidade de segurança – não afetará as demais;
- ✓ Migração e ampliação mais fácil: mudar o serviço de ambiente de virtualização é uma tarefa que pode ser feita rapidamente, assim como a ampliação da infraestrutura.

A tabela 1 tem por finalidade detalhar o ambiente de teste composto por 4 máquinas virtualizado pelo software Virtualbox.

Tabela 1 - Ambiente de teste virtualizado detalhado.

Máquina	SO	Memória	ETH0	ETH1
debianSRV	Debian 7	1024 MB		2001:888:db8:1::a
debianCliente	Debian 7	1024 MB	2001:888:db8:1::2000	*
debianAtacante	Debian 7	1024 MB	2001:888:db8:1::1fc3	*
ubuntuCliente	Ubuntu 13.10	1024 MB	2001:888:db8:1::1228	*

Fonte: do Autor.

A não utilização de máquinas com a plataforma Windows no ambiente de teste foi tomada pela incompatibilidade dos sistemas operacionais com o modo IPSEC implementado. Segundo o suporte Microsoft, não é possível estabelecer uma ligação IPSEC entre um sistema operacional Linux e um sistema operacional Windows. Esta incompatibilidade é dada, segundo o suporte Microsoft, devido à

negociação das chaves IPSEC, no qual o protocolo AH e o *payload* de segurança de ESP são utilizados para proteger o mesmo pacote, logo, ocorre a utilização de ambos os protocolos, AH e ESP. Em sistemas Windows, a ordem destes protocolos mudam e este comportamento interrompe a negociação das chaves. Neste caso, quando inicia a conexão de IPSEC a partir de um computador baseado em Linux, o sistema operacional Linux propõe que o formato de segurança do IPsec é AH + ESP. Por conseguinte, não é possível estabelecer a ligação.

A tabela 2 traz as relações de endereços físicos (MAC) das máquinas integrantes do ambiente virtual de testes.

Tabela 2 - Relação endereços MAC ambiente de teste.

Máquina	MAC ETH0	MAC ETH1
debianSRV	08:00:27:7f:7b:46	08:00:27:58:8b:39
debianAtacante	08:00:27:8e:49:87	*
debainCliente	08:00:27:bc:e8:9a	*
ubuntuCliente	08:00:27:51:1c:3d	*

Fonte: do Autor.

A tabela 1 traz, de forma detalhada, a configuração de hardware virtual das máquinas que compõem o ambiente de teste. Neste cenário, a máquina debianSRV fará o papel de servidor da arquitetura cliente servidor, provendo serviços de FTP pelo pacote proftpd, DHCP e DHCPv6, através do pacote dhcp3-server. Ambos os serviços, FTP e DHCP, foram instalados e configurados em sua forma básica de funcionamento destinada a testes.

Além disso, pode-se observar a existência de duas interfaces de rede na máquina debianSRV, ETH0 e ETH1. A primeira, ETH0, é responsável pela conexão com a Internet via protocolo IPv4; A segunda, ETH1, é responsável pela distribuição de endereços IPv4 e IPv6 pelo serviço de DHCP.

A máquina debianAtacante fará o papel de um cliente, mas dotado de software mal-intencionado para a execução dos ataques. Já os outros dois clientes, debianCliente e ubuntuCliente, irão representar clientes em uma rede local, fazendo uso dos serviços disponibilizados pela máquina servidora.

3.1 FERRAMENTA DE ATAQUE

O desenvolvimento de ferramentas para exploração das vulnerabilidades do IPv6 é relativamente lento. Atualmente, são poucas as alternativas existentes para a execução de ataques em redes IPv6. Visando à exploração das vulnerabilidades existentes nas traduções de mensagens do protocolo NDP citadas na subseção 2.4.3, fez-se uso do *toolkit* THC-IPv6, um “kit” de ferramentas de ataque específica para redes IPv6. Este toolkit desenvolvido pela *The Hackers Choice* (THC) provê fácil utilização para a realização de ataques em redes IPv6, permitindo explorar as vulnerabilidades com a utilização de poucas linhas de código.

Entre as diversas estratégias de ataques que têm por objetivo explorar as vulnerabilidades do protocolo IPv6, segundo Hauser (2013), este *toolkit* fornece:

- ✓ Alive6: um *host scan* para redes IPv6. Utiliza mensagens *echo request* para realizar o *scan*;
- ✓ Parasite6: ferramenta para ataques MITM. O programa realiza uma varredura na rede e responde a todos os pacotes NS com o seu próprio endereço MAC;
- ✓ Fake-advertiser6: anuncia uma máquina na rede com um pacote NA;
- ✓ DoS-new-ip6: ferramenta de negação de serviço em resposta a mensagens DAD (*Duplicated Address Detection*), informando que o endereço desejado pela vítima já está em uso por outra máquina.

Além disso, um *sniffer* de rede irá ser utilizado para auxiliar nos testes desenvolvidos. Um *sniffer* de rede tem por objetivo interceptar e registrar o tráfego de dados em uma rede de computadores, capturando os pacotes transmitidos, decodificando-os e facilitando a análise do seu conteúdo pelo usuário de acordo com o protocolo definido, entre outros possíveis critérios de filtragem.

Por outro lado, o *sniffer* pode ser utilizado para fins maliciosos, pois através da utilização desta ferramenta, é possível capturar o tráfego da rede, obter cópias de arquivos importantes durante sua transmissão e obter senhas de acesso, ampliando ainda mais o leque de intrusão em um ambiente invadido. Sendo assim, através do *software* Wireshark, segundo Brito (2012), é possível capturar todo o tráfego de uma rede de computadores em tempo de execução através de uma interface de rede específica.

3.2 EXECUÇÃO DO ATAQUE

Após a instalação e configuração do *toolkit* THC-IPv6 na máquina debianAtacante, primeiramente, foi realizado testes de conectividade entre os *hosts* de rede, através do Ping[11]. Todas as máquinas presentes no ambiente de teste responderam aos testes de conexão, o próximo passo foi realizado o ataque.

O ataque foi executado pela ferramenta Parasite6 que, segundo Hauser (2013), consiste em um ARP *spoofing* para redes IPv6, no caso, em um ataque de NDP *spoofing*, redirecionando todo o tráfego local para a máquina atacante. Além disso, é necessário ativar o roteamento de pacotes na máquina atacante, caso contrário, o ataque será do tipo *Denial of Service* (DoS).

Para iniciar o ataque, executamos o comando parasite6, conforme ilustrado na figura 16.

Figura 16 - Ataque Parasite6.

```
Sex Nov  1 15:39:31 BRST 2013
root@debianAtacante:~# parasite6 -RF eth0
Remember to enable routing (ip_forwarding), you will denial service otherwise!
 =>  echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
Started ICMP6 Neighbor Solitication Interceptor (Press Control-C to end) ...
```

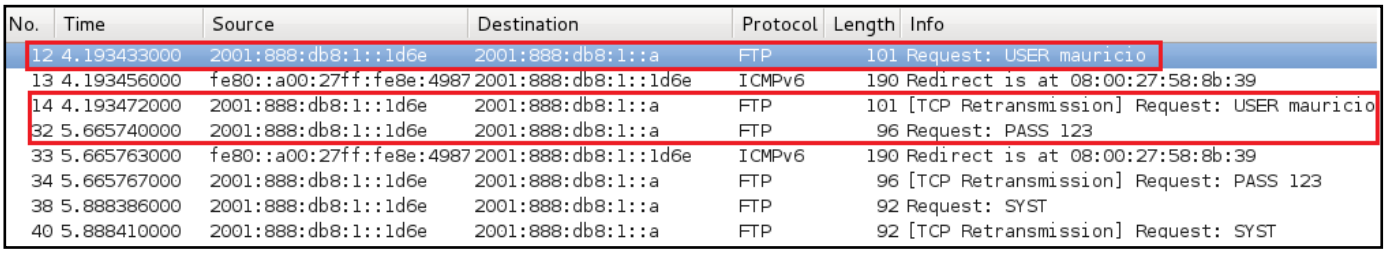
Fonte: do Autor.

A execução do comando da figura 16 dá início ao ataque NDP *spoofing*. Para a execução do ataque, fez-se uso da ferramenta Parasite6, através do comando “parasite6”, com a utilização dos parâmetros RF, onde R faz referência à “injeção” do endereço da máquina debianAtacante como destino padrão e F faz a fragmentação dos pacotes para burlar a segurança de rede. Paralelo ao ataque, executou-se a ferramenta Wireshark para auxiliar na interpretação dos pacotes interceptados.

Para testar a eficiência do ataque e analisar os resultados obtidos, na máquina debianCliente, foi executado uma conexão FTP com a máquina debianServidor. Fazendo uso da ferramenta Wireshark, observam-se os resultados obtidos na figura 17 e 18 pela máquina debianAtacante.

[11] Utilitário de rede que faz uso do protocolo ICMP para testar a conectividade entre os equipamentos de rede.

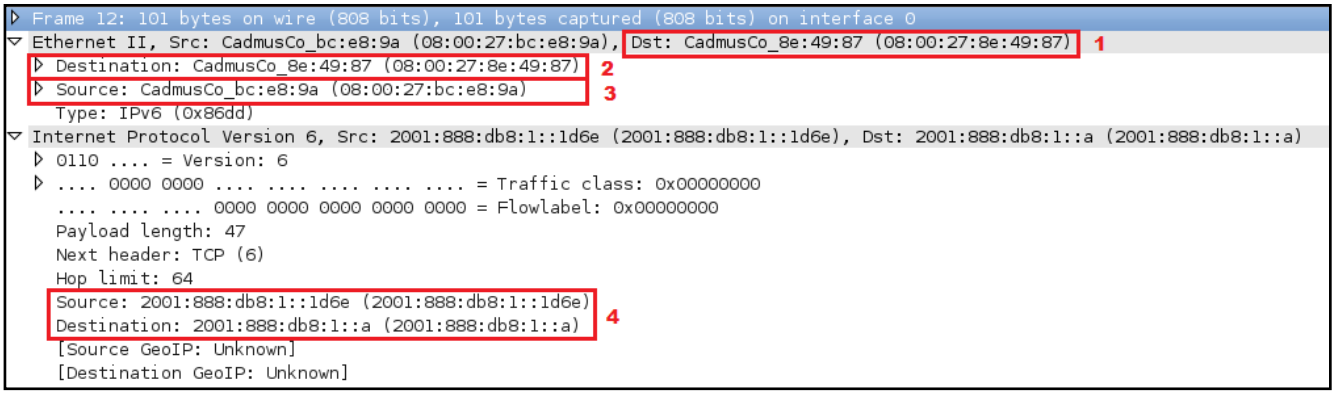
Figura 17 - Ataque parasite6 em execução.



No.	Time	Source	Destination	Protocol	Length	Info
12	4.193433000	2001:888:db8:1::1d6e	2001:888:db8:1::a	FTP	101	Request: USER mauricio
13	4.193456000	fe80::a00:27ff:fe8e:4987	2001:888:db8:1::1d6e	ICMPv6	190	Redirect is at 08:00:27:58:8b:39
14	4.193472000	2001:888:db8:1::1d6e	2001:888:db8:1::a	FTP	101	[TCP Retransmission] Request: USER mauricio
32	5.665740000	2001:888:db8:1::1d6e	2001:888:db8:1::a	FTP	96	Request: PASS 123
33	5.665763000	fe80::a00:27ff:fe8e:4987	2001:888:db8:1::1d6e	ICMPv6	190	Redirect is at 08:00:27:58:8b:39
34	5.665767000	2001:888:db8:1::1d6e	2001:888:db8:1::a	FTP	96	[TCP Retransmission] Request: PASS 123
38	5.888386000	2001:888:db8:1::1d6e	2001:888:db8:1::a	FTP	92	Request: SYST
40	5.888410000	2001:888:db8:1::1d6e	2001:888:db8:1::a	FTP	92	[TCP Retransmission] Request: SYST

Fonte: do Autor.

Figura 18 - Pacote interceptado.



Fonte: do Autor.

A figura 17 apresenta os pacotes interceptados pela máquina debianAtacante, de endereço IP 2001:888:db8:1::1fc3. No momento da conexão FTP entre a máquina debianCliente, endereço IP 2001:888:db8:1::1d6e e debianSRV, endereço 2001:888:db9:1::a, pode-se observar que a máquina debianAtacante, durante a ocorrência do ataque, torna-se o *gateway* da rede. Ou seja, a máquina irá se comportar como um nó central de rede, responsável pelo roteamento dos pacotes na rede virtual. A ocorrência deste evento é clara, pois todas as requisições FTP ocorridas entre cliente e servidor foram interceptadas, inclusive os dados referentes a usuário e senha no momento em que a conexão FTP foi estabelecida, conforme destacado na figura 17.

Já na figura 18, observa-se a ocorrência do ataque NDP *spoofing* com mais detalhes. É visto que, os campo *destination*, representados pelos números 2 e 3, fazem referência ao endereços MAC da máquina debianAtacante. Já nos campos representados pelo número 4, tem-se os endereços IP das máquinas debianCliente e debianSRV, envolvidos na conexão FTP. No momento em se inicia o ataque, a ferramenta Parasite6 anexa o endereço MAC da máquina debianAtacante ao campo *destination*. Desta forma, o endereço IP de destino, representado pelo número 4,

está relacionado ao endereço MAC da máquina debianAtacante. Desta forma, a máquina atacante torna-se o nó central da rede, fazendo com que todos os pacotes sejam redirecionados para ela e, posteriormente, entregues ao seu real destino, máquina debianCliente. Para a máquina clienteUbuntu, o mesmo processo ocorre.

3.3 NEUTRALIZAÇÃO DO ATAQUE

Como descrito na subseção 2.4.4, o IPSEC possui duas formas de implementação: modo túnel e modo transporte. Neste trabalho, foi utilizado o IPSEC em modo transporte, implementação *default* do IPSEC. Esta modalidade é utilizada para comunicações *host* a *host*. Nesta modalidade ocorre a criptografia da mensagem *playload* enviada por um pacote IP, fazendo utilização dos protocolos de segurança AH e/ou ESP, conforme expecificado nas páginas (41 e 42).

A configuração do IPSEC foi realizada a partir da instalação do pacote IPSec-tools, um utilitário para configuração IPSEC desenvolvido pela The KAME Projects. Segundo a The KAME Projects, ferramenta contém componentes para manipular e suportar conexões IPSEC, tais como:

- ✓ libipsec: biblioteca com a implementação PF_KEY;
- ✓ setkey: ferramenta para manipulação do kernel, *Security Policy, Database* (SPD) e *Security Association Database* (SAD);
- ✓ racoon: *Internet Key Exchange Daemon* (IKE) automático para conexões com chave IPSEC;
- ✓ racoonctl: ferramenta de controle baseado em *shell*[12] para racoon.

Há outras ferramentas para configuração e gerenciamento do IPSEC, tais como FreeS/WAN e QuickSec. Porém, o IPSec-tools é a ferramenta com maior referencial teórico e de fácil configuração.

Para que o IPSEC seja funcional no ambiente de teste proposto, todas as configurações presentes neste trabalho foram realizadas nas máquinas, conforme a tabela 1. As configurações do IPSEC iniciam-se com a criação das chaves de autenticação utilizadas pelos protocolos AH e ESP, em todas as máquinas que compõem o ambiente de teste, conforme mostra as figuras 19 e 20.

---

[12] Módulo que atua como interface como usuário em um sistema operacional, possuindo diversos comandos internos que permitem ao usuário solicitar serviços do sistema operacional.

Figura 19 - Criação chave de autenticação AH.

```
dd if=/dev/random count=16 bs=1 | xxd –ps
```

Fonte: do Autor.

Figura 20 - Criação chave de autenticação ESP

```
dd if=/dev/random count=24 bs=1 | xxd –ps
```

Fonte: do Autor.

Nas figuras 19 e 20, tem-se a representação da criação das chaves utilizadas pelo IPSEC. Para a criação das chaves, fez-se uso de um gerador de números aleatórios localizado em “/dev/random”. Os comandos retornaram duas chaves: uma com tamanho igual a 16 bytes (128 bits), para o algoritmo hmac-md5 utilizado pelo protocolo AH. Uma segunda chave, com tamanho de 24 bytes (192 bits), para algoritmos 3des-cbc, utilizado pelo protocolo ESP.

Cada chave criada deverá ser única para cada máquina e anexada ao arquivo de configuração do “ipsec-tools.conf”, localizado nodo diretório “/etc”. A figura 21 ilustra o arquivo de configuração ipsec-tools.conf da máquinaSRV .

Figura 21 - ipsec-tools.conf debianSRV.

```
1  ##Flush the SAD and SPD
2  flush;
3  spdflush;
4  ## IPSEC-TOOLS SERVIDOR ##
5
6  ## CHAVE SERVIDOR ##
7  add 2001:888:0db8:1::a 2001:888:db8:1::2000 ah 0x200 -A hmac-md5
8  0xb927357fd1cf4941b374296d955bdd26;
9  add 2001:888:0db8:1::a 2001:888:db8:1::2000 esp 0x201 -E 3des-cbc
10 0xa53b9d6f14983585977dac92ed4131895a8a24db39334cee;
11 add 2001:888:0db8:1::a 2001:888:db8:1::1228 ah 0x400 -A hmac-md5
12 0x123ccb014529690735e4459206afbac1;
13 add 2001:888:0db8:1::a 2001:888:db8:1::1228 esp 0x401 -E 3des-cbc
14 0x6c5ad3d6092c347b73ad3bbdffa5568c9d14345a8a501a23;
15
16 ## CHAVE CLIENTE DEBIAN ##
17 add 2001:888:0db8:1::2000 2001:888:db8:1::a ah 0x300 -A hmac-md5
18 0x2fcb44080e8d7f777491fa13c36f4544;
19 add 2001:888:0db8:1::2000 2001:888:db8:1::a esp 0x301 -E 3des-cbc
20 0x55685548cdd7c0dc7d5d8270dcde60510234ff8fcd6006bc;
21
22 ## CAHVE CLIENTE UBUNTU ##
23 add 2001:888:0db8:1::1228 2001:888:db8:1::a ah 0x500 -A hmac-md5
24 0xb563a94f5654e73aaf753c9fbed70536;
25 add 2001:888:0db8:1::1228 2001:888:db8:1::a esp 0x501 -E 3des-cbc
26 0x689626051ea84897478dff22fc5eb82db943f60ce5672ff3;
27
28 ## POLITICA DE SEGURANCA DE SAIDA ##
29 spdadd 2001:888:0db8:1::a 2001:888:db8:1::2000 any -P out ipsec
30       esp/transport//require
31       ah/transport//require;
32 spdadd 2001:888:0db8:1::a 2001:888:db8:1::1228 any -P out ipsec
33       esp/transport//require
34       ah/transport//require;
35
36 ## POLITICA DE SEGURANCA DE ENTRADA ##
37 spdadd 2001:888:0db8:1::2000 2001:888:db8:1::a any -P in ipsec
38       esp/transport//require
39       ah/transport//require;
40 spdadd 2001:888:0db8:1::1228 2001:888:db8:1::a any -P in ipsec
41       esp/transport//require
42       ah/transport//require;
```

Fonte: do Autor.

O arquivo de configuração do IPSEC da máquina debianSRV centraliza todas as configurações necessárias para que o modo transporte do IPSEC seja funcional entre as máquinas clienteDebian/debianServidor e entre ubuntuCliente/debianSRV.

Nas linhas 2 e 3, tem-se as diretivas responsáveis por limpar o banco de dados utilizados para armazenar as políticas de segurança, em especifico, o comando “spdsflush” remove todas e quaisquer entradas criadas.

Nas linhas 7 e 9, é adicionado os endereços IPs das máquinas debianSRV e debianCliente, respectivamente, para que seja possível a conexão entre as mesmas. Nas linhas 8 e 10, encontram-se as chaves AH e ESP, respectivamente. Ambas as chaves criadas serão utilizadas para a conexão IPSEC entre as máquinas debianSRV e debianCliente. O caractere 0x, indicando que a mesma é uma chave expressa em hexadecimal.

Nas linhas 11 e 13 ocorre o mesmo processo descrito anteriormente, mas agora para a adição da conexão entre as máquina debianSRV e ubuntuCliente. Nas linhas 12 e 14, encontram-se as chaves AH e ESP, respectivamente. Vale lembrar que para cada máquina adicionada pelo parâmetro “add” é necessária a criação de uma nova chave.

A partir da linha 28, ocorrem as políticas de segurança para conexões de entrada e saída. Nas linhas 29, 32, 37 e 40, tem-se o comando “spdadd” que, primeiramente, irá liberar conexões de saída (*out*), entrada (*in*) de qualquer protocolo (*any*) do IP de origem ao IP destino. Nas linhas 30, 31, 33, 34, 38, 39, 41 e 42, o comando aplica a utilização dos protocolos ESP e AH, respectivamente, para garantir a integridade e autenticidade da conexão *host* a *host*. Nas mesmas linhas, o parâmetro “*require*” determina que a associação de segurança IPSEC seja obrigatória para que ocorra a troca de pacotes entre as pontas envolvidas.

Nas figuras 22 e 23, há a configuração realizada na máquina debianCliente e ubuntuCliente. Em suma, os arquivos de configurações de ambas as máquinas são semelhantes, alterando-se apenas as linhas referentes às chaves de conexões IPSEC de cada máquina e as políticas de segurança de entrada e saída.

Figura 22 - ipsec-tools.conf debianCliente.

```
#!/usr/sbin/setkey -f

##### CONFIGURAÇÃO DEBIAN CLIENTE #####

## Flush the SAD and SPD

flush;
spdflush;

## CHAVE DE SEGURANCA SERVIDOR ##
add 2001:888:0db8:1::a 2001:888:db8:1::2000 ah 0x200 -A hmac-md5
0xb927357fd1cf4941b374296d955bdd26;
add 2001:888:0db8:1::a 2001:888:db8:1::2000 esp 0x201 -E 3des-cbc
0xa53b9d6f14983585977dac92ed4131895a8a24db39334cee;

## CHAVE DE SEGURANCA CLIENTE ##
add 2001:888:0db8:1::2000 2001:888:db8:1::a ah 0x300 -A hmac-md5
0x2fcb44080e8d7f777491fa13c36f4544;
add 2001:888:0db8:1::2000 2001:888:db8:1::a esp 0x301 -E 3des-cbc
0x55685548cdd7c0dc7d5d8270dcde60510234ff8fcd6006bc;

### POLITICAS DE SEGURANÇA ###
spdadd 2001:888:0db8:1::2000 2001:888:db8:1::a any -P out ipsec
     esp/transport//require
     ah/transport//require;

spdadd 2001:888:0db8:1::a 2001:888:db8:1::2000 any -P in ipsec
     esp/transport//require
     ah/transport//require;
```

Fonte: do Autor.

Figura 23 - ipsec-tools ubuntuCliente.

```
#!/usr/sbin/setkey -f

## Flush the SAD and SPD

flush;
spdflush;

## IPSEC-TOOLS CLIENTE UBUNTU ##

## CHAVE SERVIDOR ##
add 2001:888:0db8:1::a 2001:888:db8:1::1228 ah 0x400 -A hmac-md5
0x123ccb014529690735e4459206afbac1;
add 2001:888:0db8:1::a 2001:888:db8:1::1228 esp 0x401 -E 3des-cbc
0x6c5ad3d6092c347b73ad3bbdffa5568c9d14345a8a501a23;

## CAHVE CLIENTE UBUNTU ##
add 2001:888:0db8:1::1228 2001:888:db8:1::a ah 0x500 -A hmac-md5
0xb563a94f5654e73aaf753c9fbed70536;
add 2001:888:0db8:1::1228 2001:888:db8:1::a esp 0x501 -E 3des-cbc
0x689626051ea84897478dff22fc5eb82db943f60ce5672ff3;

## POLITICA DE SEGURANCA DE SAIDA ##
spdadd 2001:888:0db8:1::1228 2001:888:db8:1::a any -P out ipsec
     esp/transport//require
     ah/transport//require;

## POLITICA DE SEGURANCA DE ENTRADA ##
spdadd 2001:888:0db8:1::1228 2001:888:db8:1::a any -P in ipsec
     esp/transport//require
     ah/transport//require;
```

Fonte: do Autor.

Realizadas as configurações nas máquinas debianSRV, debianCliente e ubuntuCliente, é necessário que as configurações sejam carregadas no SPD (*Security Parameter Database*) para que o IPSEC tenha seu correto funcionamento. Para isso, o comando “setkey –f /etc/ipsec-tools.conf”, apontando para o arquivo de configuração do IPSEC, irá carregar todas as diretivas de configuração em seu conteúdo no SPD. Já o comando “setkey –DP” irá listar as configurações carregadas no SPD. A situação descrita anteriormente está representada na figura 24. Nela encontram-se todas as informações referentes à utilização das chaves, data de criação, etc.

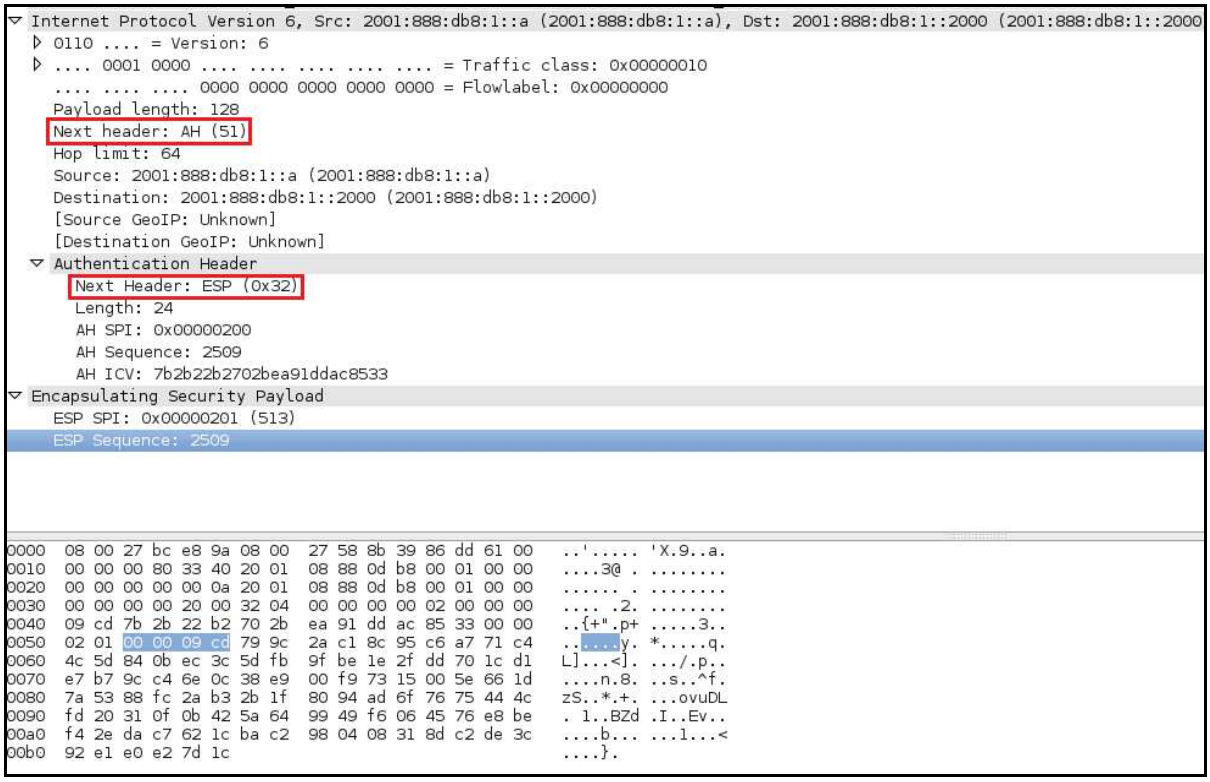
Figura 24 - Lista de diretivas de segurança debianServidor.

```
1  Sex Nov  8 11:02:38 BRST 2013
2  root@debianSRV:/home/mauricio# setkey -DP
3  2001:888:db8:1::1d6e[any] 2001:888:db8:1::a[any] 255
4       fwd prio def ipsec
5       esp/transport//require
6       ah/transport//require
7       created: Nov  8 02:08:37 2013  lastused:
8       lifetime: 0(s) validtime: 0(s)
9       spid=450 seq=1 pid=3667
10      refcnt=1
11 2001:888:db8:1::1d6e[any] 2001:888:db8:1::a[any] 255
12      in prio def ipsec
13      esp/transport//require
14      ah/transport//require
15      created: Nov  8 02:08:37 2013  lastused: Nov  8 02:14:09 2013
16      lifetime: 0(s) validtime: 0(s)
17      spid=440 seq=2 pid=3667
18      refcnt=1
19 2001:888:db8:1::a[any] 2001:888:db8:1::1d6e[any] 255
20      out prio def ipsec
21      esp/transport//require
22      ah/transport//require
23      created: Nov  8 02:08:37 2013  lastused: Nov  8 02:14:35 2013
24      lifetime: 0(s) validtime: 0(s)
25      spid=433 seq=0 pid=3667
26      refcnt=1
```

Fonte: do Autor.

A lista das configurações por parte dos clientes é a mesma, diferindo apenas na ordem dos endereços IPs referente às linhas 3,11 e 19. Para verificar se o IPSEC modo transporte entre as máquinas foi estabelecido, o teste de conectividade Ping6 foi executado entre as máquinas debianSRV e debianCliente e, paralelo a isso, a ferramenta Whireshark foi executada na máquina debianCliente para analisar as trocas de pacotes entre os pares, conforme ilustrado na figura 25.

Figura 25 - Trocas de mensagens ICMP via IPSec.



Fonte: do Autor.

Observa-se na figura 25 que o IPSEC em modo transporte foi implementado. A comprovação disto está na adição do campo AH no cabeçalho IPv6 original. Além disso, é visto que o campo *Next Header* referente ao cabeçalho AH faz referência ao protocolo ESP. Este, por sua vez, foi adicionando após o cabeçalho IPv6 original. Como descrito na subseção 2.4.4.2, a utilização dos protocolos AH e ESP na implementação do IPSEC em modo transporte proporciona a segurança apenas aos cabeçalhos superiores em relação ao cabeçalho IPSEC, pois o mesmo é adicionado após o cabeçalho IP original. Em relação à máquina ubuntuCliente, o resultado do teste de conectividade com a máquina debianSRV retornou o mesmo resultado mas, com seus respectivos endereços IPs.

3.4 RESULTADOS

De acordo com os resultados obtidos pelos testes na subseção 3.2, o ataque MITM executado pela ferramenta Parasite6 provou que os pacotes de conexão FTP trocados entre as máquinas debianCliente e debianFTP foram interceptados. A partir de então, a implementação das políticas de segurança IP em modo transportes

foram realizadas e os testes de interceptação de dados executados novamente. O objetivo da reexecução do teste é demostrar que realmente o IPSEC implementado irá garantir a segurança dos pacotes IP que trafegam na rede IPv6.

Na figura 26, tem-se a execução da ferramenta Parasite6 com a finalidade de interceptar o tráfego com o IPSEC em modo transporte implementado.

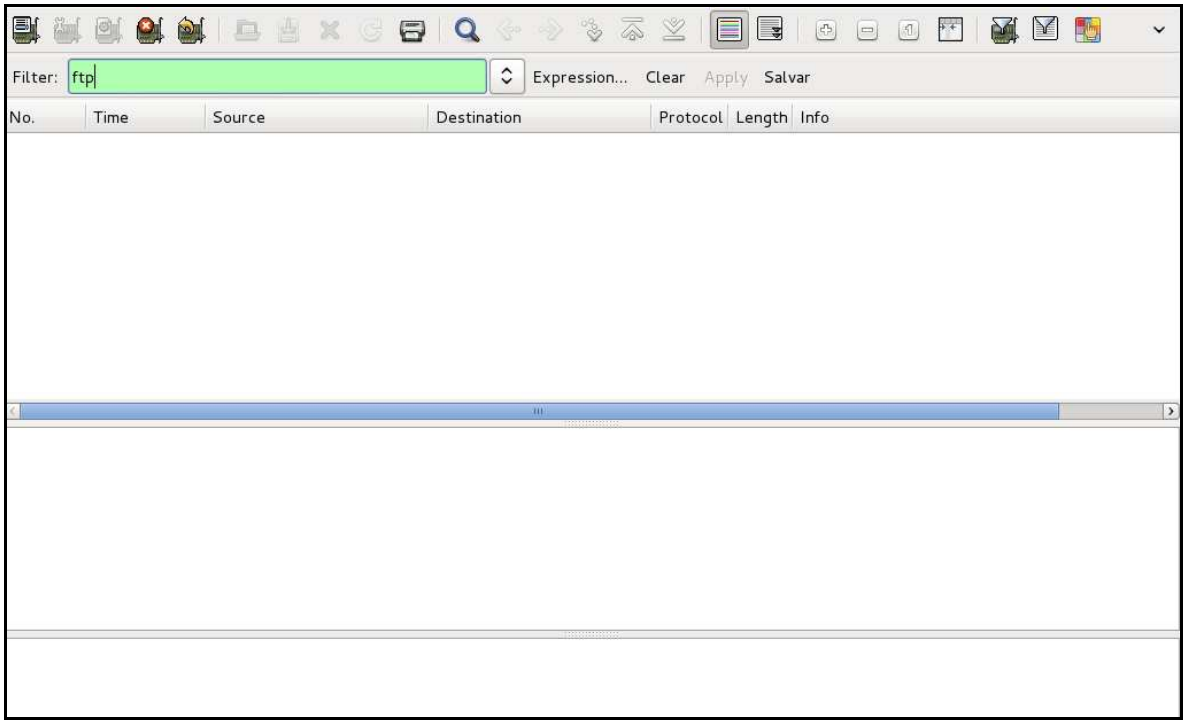
Figura 26 - Parasite6 sem captura.

```
Ter Nov 12 15:25:42 BRST 2013
root@debianAtacante:~# parasite6 -RF eth0
Remember to enable routing (ip_forwarding), you will denial service otherwise!
 =>  echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
Started ICMP6 Neighbor Solitication Interceptor (Press Control-C to end) ...
```

Fonte: do Autor.

Feita a conexão FTP entre as máquinas envolvidas nos testes, a figura 28 traz o resultado dos testes obtidos pela máquina debianAtacante, responsável pela intercepção do tráfego de rede. Após a implementação das políticas de segurança IPSEC nas máquinas debianSRV, debianCliente e ubuntuCliente, os testes de conexão FTP foram reexecutados e, a partir do resultado obtido, filtrou-se as trocas de pacotes referente ao FTP através da ferramenta Wireshark do atacante.

Figura 27 - Wireshark debianAtacante.



Fonte: do Autor.

Pode-se observar, na figura 28, que as tentativas de interceptação do tráfego entre as máquinas falharam. No período de tempo em que se executou o teste de conexão e transferência de arquivos FTP, em nenhum momento a ferramenta Parasite6 efetivou uma interceptação. Desta forma, a implementação do módulo de segurança IPSEC implementado pela ferramenta ipsec-tools se mostrou eficiente. Através de uma configuração simples e objetiva, conseguiu-se atingir o objetivo proposto, neutralização de ataque NPD *spoofing* em redes Linux, com êxito.

4 CONSIDERAÇÕES FINAIS

No decorrer da realização do trabalho, percebeu-se que uma das principais mudanças do protocolo IPv6 em relação ao protocolo IPv4 está no uso nativo do módulo de segurança IP, o IPSEC. A utilização deste módulo visou prover a segurança necessária para neutralizar os ataques de interceptação de dados, NDP *Spoofing*, utilizados no ambiente de teste, trazendo resultados satisfatórios.

Durante a ocorrência dos testes, verificou-se que o tráfego de dados em uma rede sem segurança é um grande risco, pois os mesmos dados podem ser interceptados por um invasor, de forma transparente, comprometendo a privacidade, a integridade e a autenticidade dos dados.

Através da utilização do módulo IPSEC, aumentou-se consideradamente a segurança das conexões entre as máquinas envolvidas nos testes durante as trocas de mensagens, bloqueando qualquer tentativa de interceptação do tráfego de rede por parte do atacante. Além disso, os testes mostraram que, ao ser ativado a segurança IPSEC com as regras de conexão, os *hosts* apenas respondem aos testes e aos outros *hosts* que estão na sua lista de confiança, descartando automaticamente os hosts que não possuem a chave para uma SA para conexão e, com isso, o ataque MITM não terá efeito.

Em suma, o método proposto para neutralização do ataque NDP *Spoofing* através do IPSEC em modo transporte mostrou-se viável. Através de uma configuração simples, é possível fornecer segurança à camada de rede e a todas as outras superiores, garantindo a autenticidade, integridade e privacidade necessárias aos datagramas IPs transmitidos.

4.1 TRABALHOS FUTUROS

Para melhorar o trabalho desenvolvido, algumas indicações referentes a implementações futuras do módulo de segurança IP, IPSEC, visando o aperfeiçoamento da pesquisa, são os indicados abaixo:

- ✓ Automatização do processo de adição de endereços IPs e criação das chaves pré-compartilhadas, utilizadas pelos protocolos AH e ESP, junto ao arquivo “ipsec-tools.conf”;
- ✓ Estudar formas de implementações do IPSEC em modo transporte envolvendo conexões entre máquinas Windows e Linux;
- ✓ Estudar o IPSEC integrado com IPv6 em Modo Túnel, fazendo uso da ferramenta Racoon (Ipsec-tools, 2009) para esta configuração;
- ✓ Elaborar cenários e testes mais complexos para verificar o desempenho do IPSEC como um ambiente onde se trabalhe com *Active Directory*, no sentido de se testar o desempenho e segurança;

5 REFERÊNCIAS

ALECRIN, Emerson. O que é virtualização e para que serve?. 2012. Disponível em: <http://www.infowester.com/virtualizacao.php>. Acesso em 30 OUT 2013.

BRITO, Edivaldo. Tech tudo - Como usar o Wireshark. 2012. Disponível em <http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2012/09/como-usar-o-wireshark.html>. Acesso em: 30 OUT 2013.

CARISSIMI, Alexandre da Silva; ROCHOL, Juergem; GRANVILLE, Lisandro Zambenedetti. *Redes de computadores*. Porto Alegre. Bookman, 2009.

CGI.BR. Comitê Gestor da Internet no Brasil. *Cartilha de Segurança de Rede*.2006. Disponível em: <http://cartilha.cert.br/sobre/old/cartilha_seguranca_3.1.pdf>. Acesso em JUN 2013.

CHESWICK, William R. [et al.]. *Firewall e segurança na Internet*: repelindo o hacker ardiloso. 2. ed. Porto Alegre: Bookman, 2005.

FLORENTINO, Adilson Aparecido. *IPv6 na Prática*. São Paulo: Linux New Media Editora Ltda, 2012.

GONDIM, João J. C.; CARNUT, Marco Antônio. *ARP spoofing detection on switched ethernet networks*: afeasibility study. 2003. Disponível em: <http://www.postcogito.org/PublicationsInEnglish/arpspoof-detection-v10final2.pdf>. Acesso em: 26 JUN 2013.

HACKPITTSBURGH. *Network Securtiy*: Man-In-The-Middle Attacks. 2011. Disponível em:< http://www.hackpittsburgh.org/wp-content/uploads/2011/03/Man-In-The-Middle.jpg>. Acesso em: 24 JUN 2013.

HAUSER, Van. *THC-IPV6-Attack-Toolkit*. Disponível em: <http://www.thc.org/thc-ipv6/README>. Acesso em: 25 JUN 2013.

IPSec-tools. *RACOON*. 2009. Disponível em: <http://ipsec-tools.sourceforge.net/>. Acesso em: 02 DEZ 2013.

IPv6 Brasil. CEPTRO. Centro de Estudos e Pesquisas em Tecnologias de Redes e Operações. *IPv6*. 2012. Disponível em: <http://IPv6.br/entenda/>. Acesso em: 05 abr. 2013.

KUROSE, James F; ROSS, Keith W. *Redes de computadores e a Internet*: uma abordagem top-dow. 5. ed. São Paulo: Addilson Wesley, 2010.

MEDEIROS, Plínio. *IBGE divulga que o crescimento da internet é de 112% entre o período de 2005 a 2009*. 2010. Disponível em: <http://www.tecnocratadigital.com.br/ibge-diz-crescimento-internet-112-2005-2009/>. Acesso em: 12 ABR 2013.

MORIMOTO, Carlos Eduardo*. Redes Guia Prático*. Porto Alegre: Sul Editores, 2008.

PILIHANTO, Atik. *A Complete Guide on IPv6 Attack and Defense*. 2011. Disponível em: <http://www.sans.org/reading_room/whitepapers/detection/complete-guide-ipv6-attack-defense_33904>. Acesso em: 26 JUN 2013.

SANTOS, Rodrigo Regis; MOREIRAS, Antônio M.; REIS, Eduardo Ascenço; ROCHA, Ailton Soares. *Curso IPv6 básico*. 2010. Disponível em: <http://ipv6.br/wordpress/wp-content/plugins/download-monitor/download.php?id=IPv6-apostila.pdf>. Acesso em: 25 JUN 2013.

SOUZA, Wendley. Segurança em redes de computadores. Disponível em: <http://www.brasilescola.com/informatica/seguranca-redes.htm>. Acesso em 12 NOV 2013.

SCRIMGER, Rob; LASALLE, Paul; PARIHAR, Mridula; GUPTA, Meeta. *TCP/IP, a Bíblia*. Rio de Janeiro: Elsevier, 2002.

STALLINGS, William. *Criptografia e segurança de redes*. 4. ed. São Paulo: Pearson Prentice Hall, 2008.

Suporte Microsoft. Não é possível estabelecer uma ligação IPsec entre um sistema operativo do Linux e um sistema operativo do Windows quando inicia a ligação do sistema operativo Linux. Disponível em: <http://support.microsoft.com/kb/950826/pt>. Aceso em 01 NOV 2013.

The Hackers Choice. THC-IPV6 - *Attacking the IPV6 protocol suite*. 2013. Disponível em: <https://www.thc.org/thc-ipv6/>. Acesso em 10 OUT 2013.

The KAME Project. Disponível em: <http://www.kame.net/>. Acesso em 15 OUT 2013.

THOMSON, S. Bellcore; NARTEN, T. *IPv6 Stateless Address Autoconfiguration*. 1998. Disponível em: <http://www.ietf.org/rfc/rfc2462.txt>. Acesso em 14 SET 2013.

Virtualbox. Oracle VM VirtualBox. Disponível em: <https://www.virtualbox.org/>. Acesso em: 20 AGO. 2013.