Centro Universitário de Brasília
Instituto CEUB de Pesquisa e Desenvolvimento - ICPD

UILLIAN DA SILVA SOUZA

SMTP RELAY COM IPV6 E IPSEC

Brasília
2016

UILLIAN DA SILVA SOUZA

SMTP RELAY COM IPV6 E IPSEC

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB/ICPD) como pré-requisito para obtenção de Certificado de Conclusão de Curso de Pós-graduação *Lato Sensu* em Redes de Computadores com Ênfase em Segurança.

Orientador: Prof. Me. Marco Antônio de Oliveira Araújo.

Brasília
2016

UILLIAN DA SILVA SOUZA

SMTP RELAY COM IPV6 E IPSEC

Trabalho apresentado ao Centro Universitário de Brasília (UniCEUB/ICPD) como pré-requisito para obtenção de Certificado de Conclusão de Curso de Pós-graduação *Lato Sensu* em Redes de Computadores com Ênfase em Segurança.

Orientador: Prof. Me. Marco Antônio de Oliveira Araújo.

Brasília, 29 de junho de 2016.

Banca Examinadora

_____

Prof. Dr. Gilson Ciarallo

_____

Prof. Esp. Sylas Rodrigues Mendes

AGRADECIMENTOS

A Deus por ter me dado saúde e força para superar as dificuldades.

Ao Banco do Brasil pelo apoio e investimento.

Ao professor Marco Antônio, pela orientação, apoio e confiança.

À minha namorada Sara, a minha família e a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

RESUMO

O presente estudo se propõe a demonstrar a aplicação de criptografia no nível de rede com a utilização IPsec sobre IPv6 para o serviço de envio de e-mail por SMTP e avaliar os ganhos e perdas no desempenho do serviço com a implementação desses protocolos. Os objetivos do presente trabalho são: Demonstrar a utilização do serviço de envio de e-mail utilizando o protocolo SMTP com IPv6 e IPsec, avaliar qual o impacto da utilização do IPsec e IPv6 no desempenho do serviço por meio da medição do tempo de respostas, comparando ao uso do SMTP com IPv4 e TLS. Para alcançar esses objetivos, procedeu-se da seguinte maneira: foi criado um ambiente de laboratório com dois servidores de SMTP onde foram testadas diferentes configurações de criptografia tanto para o IPv4 quanto para o IPv6. Neste ambiente foram realizados diversos testes com diferentes tamanhos de mensagens para demonstrar a utilização do SMTP em IPv6 com IPsec e criar estatísticas de performance dos servidores para cada situação. Ao final do trabalho foi possível concluir pela viabilidade do uso de SMTP com IPv6 e o IPSec. Verificamos também que o IPv6 apresenta uma performance próxima à do IPv4 e que o uso do IPSec proporciona um grande de segurança, mas pode causar perda de performance, principalmente nos casos de envio de arquivos grandes com tamanho acima de 20MB.

Palavras-chave: Smtp Relay IPv6 Ipsec TLS

ABSTRACT

This study aims to demonstrate the encryption application at the network level with IPsec use of IPv6 for mail delivery service for SMTP and evaluate the gains and losses in service performance with the implementation of these protocols. The objectives of this work are: To demonstrate the use of the sending mail service using the SMTP protocol IPv6 and IPsec, assess what impact the use of IPsec and IPv6 in service performance by measuring the response time compared to use of SMTP with IPv4 and TLS. To achieve these goals, it proceeded as follows: a laboratory environment was created with two SMTP servers where different encryption settings were tested for both IPv4 and IPv6. In this environment several tests with different sizes of messages were performed to demonstrate the use of SMTP IPv6 with IPsec and create performance statistics for servers for each situation. At the end of the work was concluded that SMTP works well with IPv6 and IPSec. We also note that IPv6 has a close performance to the IPv4 and IPSec use provides great security, but can cause loss of performance, especially in cases of sending large files with size up to 20MB.

Key words: **Smtp Relay IPv6 Ipsec TLS**

LISTA DE FIGURAS

LISTA DE QUADROS E TABELAS

SUMÁRIO

INTRODUÇÃO

O correio eletrônico, ou simplesmente e-mail (Eletronic Mail), como a internet tem se tornado cada vez mais importante tanto em ambientes corporativos quanto na vida de pessoas comuns. Este recurso acelerou o processo de comunicação, possibilitando a entrega de mensagens e correspondências eletrônicas formais quase que imediatamente ao envio. A internet tem evoluído muito, novos sistemas têm sido criados, mas o e-mail tem se consolidado como o meio principal para comunicação formal, sendo aceito inclusive como prova jurídica.

O e-mail foi um dos primeiros serviços disponibilizados no início da internet e, ainda hoje, é um dos mais utilizados. Qualquer pessoa pode obter uma conta de e-mail que tem um endereço eletrônico para enviar e receber mensagens através da internet. Um endereço, “e-mail *address*” ou “endereço de correio eletrônico”, possui a estrutura básica usuário@domínio, onde usuário representa o identificador do usuário e domínio é o nome do domínio na internet do provedor do e-mail.

Conforme Freitas (2004), além de facilitar, agilizar, dinamizar o processo de troca de ideias, informações, ordens, reclamações, etc., o correio eletrônico possibilitou tanto a empresas quanto a usuários domésticos, diminuírem custos (e tempo) na troca de informações.

Podemos verificar a importância do e-mail e consequentemente das informações trocadas pela rede através desta ferramenta. Logo, surge a necessidade de garantir que este serviço seja rápido e seguro.

O *Internet Protocol version* 6 (IPv6) foi criado em meados dos anos 90 para suprir a falta de endereços IPv4, realidade que ocorreu inicialmente na China e na Índia. Logo, esta nova versão do protocolo de internet (IP) substituirá a versão 4 que está no seu limite de endereços possíveis. O IPv6 possui muitas mudanças em relação ao Protocolo IPv4, sendo que a mais significante delas foi o aumento do tamanho do campo de endereços de rede, que foi de 32 para 128 bits. Assim, agora o limite de endereços é extremamente alto que atenderá tranquilamente a crescente demanda mundial. A grande expansão do uso da Internet, a explosão dos dispositivos móveis, a inclusão digital, entre outros, acabou esgotando a capacidade

de endereços únicos do IPv4. O IPv6 acaba com o limite de endereçamento além de apresentar novas especificações em relação a seu antecessor.

Temos o desenvolvimento da nova versão do IPv6, que possui novos recursos, inclusive de segurança e grande capacidade quanto em relação ao endereçamento, mas nada é totalmente seguro e ainda existem fragilidades e precisam ser eliminadas ou mitigadas.

Os sistemas de informações a cada dia são mais complexos e têm maior capacidade, com isso é agravado o problema o grande aumento do número de ameaças e vulnerabilidades, que são exploradas cada vez mais rápidas. Temos uma corrida entre o desenvolvimento de ferramentas de ataque e de defesa, que com a velocidade da internet, são disponibilizadas para o mundo, em um pequeno espaço de tempo.

Como grande aliada da segurança dos sistemas de comunicação para evitar ataques e garantir a confidencialidade da informação, temos a criptografia, que é o conjunto de princípios e técnicas empregadas para cifrar os dados, torná-la ininteligível para os que não tenham acesso às convenções combinadas.

Com a utilização de protocolos de criptografia como o TLS (*Transport Layer Secure*) e o IPsec (*Internet Protocol Security*), é possível criar uma VPN (*Virtual Private Network*) que é um túnel que oferece mecanismos que garantem a segurança da comunicação na internet que é um meio inseguro. Seu uso viabiliza a utilização da própria internet para realização de conexões confiáveis, outra opção de alto custo para estas conexões seria uma infraestrutura dedicada sem concorrência de terceiros indevidos.

Os principais protocolos de criptografia atualmente utilizados nas comunicações através da Internet são o TLS e o IPsec.

O presente estudo se propõe a demonstrar a aplicação de criptografia no nível de rede com a utilização IPsec sobre IPv6 para o serviço de envio de e-mail por SMTP e avaliar os ganhos e perdas no desempenho do serviço com a implementação desses protocolos.

Os objetivos do presente trabalho são: Demonstrar a utilização do serviço de envio de e-mail utilizando o protocolo SMTP com IPv6 e IPsec, avaliar qual o

impacto da utilização do IPsec e IPv6 no desempenho do serviço por meio da medição do tempo de respostas, comparando ao uso do SMTP com IPv4 e TLS.

Para alcançar esses objetivos, procedeu-se da seguinte maneira: Foi criado um ambiente de laboratório com dois servidores de SMTP onde foram testadas diferentes configurações de criptografia tanto para o IPv4 quanto para o IPv6. Neste ambiente foram realizados diversos testes com diferentes tamanhos de mensagens para demonstrar a utilização do SMTP em IPv6 com IPsec e criar estatísticas de performance dos servidores para cada situação.

O presente trabalho foi então estruturado em 4 capítulos.

No primeiro capítulo, apresenta-se o protocolo de envio de e-mails, o SMTP, e também o protocolo IPv6 que substituirá o modelo atual de endereçamento da internet; No segundo são descritos os protocolos TLS e IPsec, que são responsáveis por adicionar segurança às informações transmitidas na rede, criptografando os pacotes; no terceiro capítulo, temos a proposta de instalação e metodologia onde são apresentados os recursos utilizados, suas configurações e a explicação de como foram executados os testes; no quarto e último capítulo são apresentados os resultados e análise da pesquisa e testes realizadas no trabalho.

1 OS PROTOCOLOS SMTP E IPv6

1.1 SMTP

O correio eletrônico (e-mail) é um dos mais populares serviços da internet. A maioria dos sistemas de e-mail envia as mensagens através do protocolo SMTP - *Simple Mail Transfer Protocol* (RIABOV, 2005).

SMTP é o protocolo de aplicação da suíte de protocolos TCP/IP que permitem o envio de e-mail pela internet.

Um e-mail é enviado por uma série de transações de solicitação-resposta entre um cliente e um servidor. Essas transações entregam a mensagem, que é composta de cabeçalho, corpo, e o envelope, similarmente a uma carta tradicional (RIABOV, 2005).

O principal objetivo do SMTP, de acordo com a RFC 5321 (2008), é transferir e-mails com segurança e eficiência. Para isso, conta com diversas especificações, que se seguidas, criam uma poderosa ferramenta de entrega e recebimento de e-mails.

O SMTP é utilizado apenas para envio de mensagens. Para recuperar as mensagens do servidor, são utilizados outros protocolos como POP, IMAP e também o HTTP no caso de um webmail (RHOTON, 2000).

1.1.1 SMTP Relay

SMTP *Relay*, ou retransmissão de SMTP é um importante recurso do SMTP, é a possibilidade de enviar e-mails para outros domínios em várias redes diferentes (RFC 5321, 2008).

Através deste recurso podemos enviar e-mails entre servidores de domínios diferentes. Quando o servidor recebe uma mensagem destinada a um domínio de internet diferente do seu, se o SMTP *Relay* estiver disponível ele pesquisa via DNS qual o servidor MX responsável por aquele domínio e transfere a mensagem para o destino correto (RFC 5321, 2008).

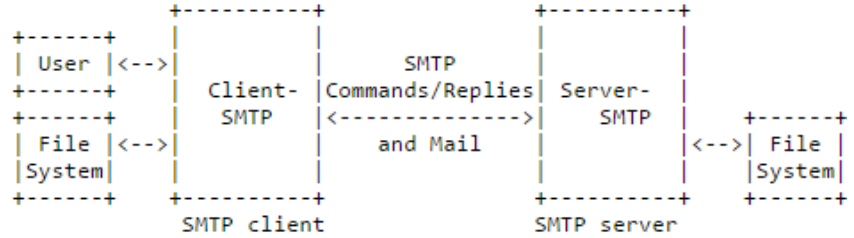
Um servidor SMTP pode ser o destino final de uma mensagem ou ser usado como relay, isto é, em algum momento pode assumir o papel de cliente e conectar-se com outro servidor (CONCEIÇÃO et al., 2014).

1.1.2 Estrutura do SMTP

O envio de mensagem por SMTP envolve um cliente e um servidor. Quando o cliente quer enviar uma mensagem, inicia uma conversa com o servidor em uma conexão de duas vias. O cliente é responsável por transferir a mensagem para um ou mais servidores ou avisar caso ocorra falhas (RFC 5321, 2008).

Na figura abaixo, temos a representação do design do SMTP conforme RFC 5321 (2008):

Figura 1 – Design do SMTP



Fonte: RFC 5321 (2008, p. 7)

A conversa entre um cliente SMTP e um servidor se dá por meio de comandos predefinidos, tais como: EHLO, MAIL, QUIT, etc. Quem envia os comandos é o cliente e quem gera as respostas é o servidor. Deste modo, uma transferência de e-mail pode ser dada por uma única conexão entre o remetente e o destinatário ou pode ocorrer através de uma série de “saltos” de servidor para servidor até o destino final. Em qualquer um dos casos, quando o cliente termina de enviar a mensagem para o servidor, sendo este o destino final ou não, o servidor deve se responsabilizar por entregar o e-mail com segurança ou informar ao cliente que ocorreu um erro (CONCEIÇÃO et al., 2014).

Os clientes de e-mail encontram os servidores por meio de DNS MX Records (*Mail Exchanger Record*), que são resolvidos em endereços IP.

O protocolo SMTP foi modificado de modo que o cliente e o servidor aceitem dividir funcionalidades além das já existentes no protocolo original. Essas

extensões permitem que versões estendidas de clientes e servidores possam se reconhecer e o servidor comunicar o cliente sobre quais serviços ele suporta.

Graças a essas mudanças, implementações antigas devem suportar até os mecanismos básicos de extensões. Por exemplo, servidores SMTP devem suportar o comando EHLO mesmo se não implementam nenhuma funcionalidade estendida e clientes devem usar o comando EHLO ao invés de HELO (Por motivos de compatibilidade com implementações mais antigas, ambos devem aceitar o comando HELO caso o EHLO não seja aceito). Estas extensões adicionadas ao protocolo SMTP existem, pois, alguns serviços que agora são importantes, não existiam quando o mesmo foi criado (CONCEIÇÃO et al., 2014).

O servidor SMTP obtém o conteúdo do e-mail pelo comando DATA, que deve estar de acordo a RFC 5322 (2008), que especifica o formato de mensagens da internet.

Durante uma transação entre o cliente e o servidor, este mantém três buffers de dados. O primeiro armazena o endereço de e-mail do remetente, o segundo armazena todos os destinatários para qual o e-mail deve ser entregue e o terceiro armazena o conteúdo do e-mail (RFC 5321, 2008).

1.1.3 Comandos do SMTP

Segundo as especificações da RFC 5321 (2008), o protocolo SMTP possui os seguintes comandos que são executados pelo cliente e recebem resposta do servidor:

- EHLO (*Extended Hello*) ou HELO (*Hello*)
- MAIL (*Mail*)
- RCPT (*Recipient*)
- DATA (*Data*)
- RSET (*Reset*)
- HELP (*Help*)
- QUIT (*Quit*)
- NOOP (*Noop*)
- VRFY (*Verify*)

Temos uma descrição dos comandos do SMTP realizada por Conceição (2014, p. 3), em seu trabalho “*Simple Mail Transfer Protocol*, uma implementação simplificada”:

> EHLO ou HELO
>
> Este comando é usado para identificar um cliente SMTP para o servidor. Tem como argumento um domínio válido do cliente, se o mesmo possuir um. Caso não possua, pode- se usar o endereço IP do cliente. O cliente SMTP identifica um servidor que implementa serviços de extinções por meio do comando EHLO. Caso o servidor aceite o comando, envia para o cliente uma mensagem de boas-vindas e uma lista com todos os serviços que implementa. Se o servidor não aceitar o comando, o cliente deve mandar o HELO por questões de compatibilidade.
>
> Com o EHLO (ou HELO), o cliente informa ao servidor que uma transação pode ser iniciada. Isto significa que deste ponto em diante, o cliente pode enviar comandos que, se forem na ordem correta, permitem o envio de um e-mail. Caso o EHLO seja enviado mais de uma vez, todas as tabelas, buffers e estados da transação tem de ser zerados.
>
> MAIL
>
> Comando inicial de uma transação de envio de e-mail entre o cliente e o servidor. Toma como argumento o endereço de e-mail do remetente e argumentos adicionais podem ser passados caso o servidor aceite serviços de extensão. O MAIL somente pode ser enviado caso nenhuma transação esteja em progresso.
>
> O envio limpa os três buffers que o servidor armazena (buffer do endereço de e-mail do remetente, buffer que armazena todos os endereços de e-mail dos destinatários e o que armazena o conteúdo do e-mail). O valor do endereço de e-mail do remetente é armazenado em um dos *buffers*.
>
> RCPT
>
> Vem após o MAIL e tem como argumento o endereço de e-mail do destinatário. Pode- se enviar mais de um RCPT durante a transação entre o cliente e o servidor. Cada uso especifica um destinatário diferente para o e-mail. Assim como o MAIL, este pode ter argumentos adicionais se o servidor suportar serviços de extensão. Os valores aqui passados também são armazenados no buffer de destinatários do servidor.
>
> DATA
>
> O DATA é o comando final em uma transação de e-mail. Quando o cliente envia o DATA, o servidor responde com o código 354, o que permite ele enviar qualquer caractere ASCII, e somente os caracteres de comandos: SP(*Space*), HT(*Halt*), CR(*Carriage Return*) e LF(*Line Feed*). Tudo o que o cliente enviou após o código 354, compõe o conteúdo do e-mail a ser enviado. O final da mensagem é indicado por: “<CR><LF>.<CR><LF>”.
>
> Assim que o servidor detecta que a mensagem terminou (Logo que recebe <CR><LF>.<CR><LF>), o mesmo armazena o conteúdo da mensagem no *buffers* de data. Feito isso, ele salva o objeto e-mail localmente em um arquivo independente para ser enviado e zera os buffers que foram preenchidos com os comandos MAIL, RCPT e DATA.

A partir do momento que o servidor aceita a mensagem do cliente, o mesmo se torna responsável por tudo o que possa acontecer com o e-mail recebido. Se ocorrer uma falha durante o envio do e-mail, o servidor deve informar o cliente sobre a falha.

RSET

É responsável por reiniciar toda transação de e-mail que esteja em andamento entre o cliente e o servidor, assim como todos os buffers que o servidor mantém. Pode ser enviado em qualquer momento da transação.

HELP

Este comando serve para o servidor enviar algumas informações de ajuda para o cliente. Não é obrigatório possuir argumentos e não deve interferir nos buffers do servidor.

QUIT

Comando responsável por finalizar a conexão entre o cliente e o servidor. O cliente não encerra a conexão imediatamente, ele somente envia o QUIT avisando que irá encerrar e espera o servidor responder com alguma mensagem. Pode ser enviado em qualquer momento da transação.

NOOP

Não toma ação nenhuma, somente pede para o servidor enviar resposta com código 250. O comando NOOP não interfere nos buffers do servidor.

VRFY

Tem a função de perguntar ao servidor se o argumento enviado é identificado como um usuário local ou endereço de e-mail. Caso o argumento seja identificado como um usuário local, informações sobre o usuário são enviadas para o cliente. O comando VRFY também não interfere nos buffers do servidor.

1.1.4 Processo de envio de e-mails

O envio de mensagens exige que os comandos sejam executados pelo cliente na sequência correta, caso contrário, ocorrerá falha no envio. Para cada comando realizado o servidor devolve uma resposta com um código que está definido no protocolo (RFC 5321, 2008).

Segue um exemplo de uma típica transação SMTP exemplificada na RFC 5321 (2008) em que uma mensagem é enviada por Smith que está no domínio bar.com para Jones, Green e Brown que estão no domínio foo.com. O cliente SMTP de bar.com se conectou diretamente ao servidor de foo.com. A mensagem foi aceita para Jones e Brown, porém Green não é um usuário válido em foo.com.

S: 220 foo.com *Simple Mail Transfer Service Ready*
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 *No such user here*
C: RCPT TO:<Brown@foo.com>
S: 250 OK
C: DATA
S: 354 *Start mail input; end with* <CRLF>.<CRLF>
C: Blah blah blah...
C: ...etc. etc. etc.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com *Service closing transmission channel*

Quando o e-mail é aceito para entrega pelo servidor, ele entra na fila de envio e assim que puder é entregue sendo armazenado na caixa de entrada do destinatário ou encaminhado para outro servidor nos casos em que o destinatário é de outro domínio.

1.2 IPv6

O IPv6 é a sexta versão do IP - Protocolo de Internet. Esta versão, substituirá seu antecessor, o IPv4. Sua criação teve início oficialmente em dezembro de 1993 quando a IETF formalizou, através da RFC 1550, as pesquisas para desenvolvimento da nova versão do protocolo IP, solicitando o envio de projetos e propostas para o novo protocolo. Só em janeiro de 1995 foi definida a recomendação final para o novo Protocolo Internet baseado 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 (SANTOS et al., 2015).

Esta versão oferece melhorias, comparada à versão anterior, os seus cabeçalhos foram alterados para um melhor aproveitamento e desempenho. A quantidade de endereços nesta versão é muito maior, pois ela, ao contrário da antiga versão já foi criada para suportar a demanda futura de endereços IP (BASSO, 2012).

Desde a criação do IPv6 foram feitas muitas modificações no protocolo, primeiramente foi testado em redes experimentais e após estar mais refinado, começou a ser utilizado em Provedores de Serviço, que passaram a utilizar o IPv6 em parte de suas redes. Grandes empresas da internet já utilizam o IPv6 e provedores como a Global Crossing, da CTBC, e da Telefônica já fornecem trânsito IPv6 comercialmente no Brasil. Devido à importância de implantação desta nova versão do IP os governos têm começado a apoiar esta implantação (SANTOS et al., 2015).

A principal motivação para a substituição do IPv4, foi a capacidade de endereçamento ter sido esgotada, uma vez que o IPv4 suporta apenas cerca de $4x10^9$, contra cerca de $3,4x10^{38}$ endereços da nova versão (SANTOS et al., 2015).

O esgotamento do IPv4 já é realidade, porém o IPv6 ainda não substituiu totalmente seu antecessor.

1.2.1 Transição

O IPv4 e o IPv6 não são diretamente compatíveis, já que o IPv6 foi projetado para substituir o IPv4, não para ser uma extensão, ou complemento, resolvendo o problema do esgotamento de endereços. Mas mesmo que não interoperem, os protocolos podem funcionar em paralelo nos mesmos equipamentos, possibilitando realizar a transição de forma gradual (SANTOS et al., 2015).

Diversas técnicas de transição foram criadas para permitir o funcionamento paralelo das duas versões. Seguem os principais tipos de técnicas de transição segundo Santos et al. (2015):

- Pilha dupla: consiste na convivência do IPv4 e do IPv6 nos mesmos equipamentos, de forma nativa, simultaneamente. Essa técnica é a técnica padrão escolhida para a transição para IPv6 na Internet e deve ser usada sempre que possível.
- Túneis: Permitem que diferentes redes IPv4 comuniquem-se através de uma rede IPv6, ou vice-versa.
- Tradução: Permitem que equipamentos usando IPv6 comuniquem-se com outros que usam IPv4, por meio da conversão dos pacotes.

1.2.2 Evoluções em relação ao IPv4

Na recomendação final para o novo Protocolo Internet, A RFC 1752 de janeiro de 1995, que especificou os requisitos para o IPv6, foram levantadas as limitações do IPv4 que deveriam ser superadas na nova versão. Seguem as principais limitações especificadas na RFC 1752 (1995) que deveriam ser superadas pelo IPv6:

- Prover um serviço de datagrama não confiável (como IPv4);
- Prover suporte *unicast* e *multicast*;
- Assegurar que o endereçamento é adequado além de um futuro previsível;
- Ser compatível com o IPv4, para que as redes existentes não precisem reinstaladas, enquanto ainda provê um caminho simples de migração do IPv4 para o IPv6;
- Prover suporte para autenticação e criptografia;
- A simplicidade arquitetônica deverá incorporar alguns recursos “adicionais” do IPv4 que foram acrescentados com o passar dos anos;
- Não fazer suposições sobre a topologia física, mídia ou capacidades da rede;
- Não fazer nada que afete o desempenho de um roteador encaminhando datagramas;
- O novo protocolo precisa ser extensível e capaz de evoluir para atender às necessidades de serviço futuras da Internet;
- É preciso haver suporte para hosts móveis, redes e interconexões

de redes;

- Permitir que os usuários criem interconexões de redes privadas em cima da infraestrutura básica da Internet.

O IPv6, atendendo aos requisitos especificadas, apresentou diversas melhorias em relação ao IPv4. Entre as evoluções observadas, podemos citar como principais os endereços quase ilimitados, aumento da mobilidade, melhor desempenho e também características de segurança superiores.

O quadro 1, abaixo, apresenta uma comparação entre IPv4 e IPv6.

Quadro 1 - Comparação entre IPv4 e IPv6

Característica	**IPv4**	**IPv6**
Tamanho do endereço	32bits	128bits
Suporte ao IPSec	Opcional	Obrigatório
Capacidade de QoS (*Quality of Service)*	Nenhuma referência	Introduz capacidades de QoS utilizando para isso o campo *Flow Label*
Fragmentação de pacotes	Processo de fragmentação realizada pelo *router*	A fragmentação deixa de ser realizada pelos *routers* e passa a ser processada pelos *hosts emissores*
Campos de opção	O cabeçalho inclui os campos de opção	Todos os campos de opção foram mudados para dentro do campo *extension header*
Descoberta de Endereços	O *Adress Resolution Protocol* (ARP), utiliza requisitos do tipo *Broadcast*	O ARP foi abandonado, sendo substituídos pelas mensagens *Neibhbor Discovery*
Multicast	*Internet Resolution Management Protocol* (IGMP) é utilizado para gerir relações locais de sub-redes	O IGMP fio substituído por mensagens *Multcast Listener Discovery*
Broadcast	Os Endereços de *Broadcast* são utilizados para enviar tráfego para todos os *hosts* de uma rede	Deixa de existir o endereço de *Broadcast*, para utilizar endereços *multicast*
Configuração dos Endereços	O endereço tem de ser configurado manualmente	Adição de funcionalidades de autoconfiguração
Tamanho dos pacotes	Suporta pacotes de 576 bytes, passiveis de serem fragmentados	Suporta pacotes de 1280 bytes, sem fragmentação

Fonte: Scimandar (2009)

1.2.3 Endereçamento no IPv6

Aumentar o limite de endereços de endereços disponíveis foi um dos principais motivos para iniciar a criação de uma nova versão do protocolo IP, sendo possível dizer que, de fato, foi o motivador da mudança, seguido por outros como melhoria do roteamento e segurança. Uma disponibilidade de endereços elevada permitirá atendimento total da demanda, uma vez que o IPv6 é capaz de endereçar todos os hosts existentes. Esta grande disponibilidade de endereços é, sem dúvida, um dos maiores benefícios da nova versão do protocolo, e atenderá a demanda crescente da internet (REGHINNI, 2014).

Santos et al. (2015) exemplifica quão é quantidade de endereços do IPv6:

> Assim, o IPv6 surgiu, com um espaço para endereçamento de 128 bits, podendo obter 340.282.366.920.938.463.463.374.607.431.768.211.456 endereços (2128). Este valor representa aproximadamente 79 octilhões (7,9x1028) de vezes a quantidade de endereços IPv4 e representa, também, mais de 56 octilhões (5,6x1028) de endereços por ser humano na Terra, considerando-se a população estimada em 6 bilhões de habitantes.

O endereço IPv6 é representado em um formato hexadecimal. Ele é dividido em oito blocos, sendo que cada bloco possui 16 bits (04 dígitos hexadecimais) separados por ":". Um endereço padrão IPv6 é dividido em três seções: *Global Routing Prefix, Subnet* e *Interface ID* (REGHINNI, 2014).

Exemplo de um endereço IPv6:
2001:0:5ef5:79fb:2c3f:2738:404f:36a

A representação dos prefixos de rede nos endereços IPv6 continua sendo escrita do mesmo modo que no IPv4, utilizando a notação CIDR, sendo representada seguindo a forma “endereço/tamanho_do_prefixo”, onde “tamanho_do_prefixo” é um valor decimal que especifica a quantidade de bits contíguos à esquerda do endereço que compreendem o prefixo (SANTOS et al., 2015).

Exemplo da representação do prefixo de uma sub-rede:

- Prefixo 2001:0:5ef5:79fb::/64

- Prefixo global 2001:0:5ef5::/48
- Identificador da sub-rede 79fb

Quadro 2 - Representação do prefixo de uma sub-rede

2001 : 0000 : 5ef5 :	79fb :	0000 : 0000 : 0000 : 0000
Prefixo global (48 bits)	Identificador da sub-rede (16 bits)	Disponível para endereçamento dos hosts (64 bits)

Fonte: Produzido pelo autor do trabalho

Conforme Santos et al. (2015), esta representação também possibilita a agregação dos endereços de forma hierárquica, identificando a topologia da rede através de parâmetros como posição geográfica, provedor de acesso, identificação da rede, divisão da sub-rede, etc. Com isso, é possível diminuir o tamanho da tabela de roteamento e agilizar o encaminhamento dos pacotes.

Ainda segundo Santos et al. (2015), temos algumas observações sobre o endereçamento do IPv6:

- Na representação de um endereço IPv6, é permitido utilizar tanto caracteres maiúsculos quanto minúsculos;
- São permitidas abreviações de endereços, onde podem ser omitidos os zeros a esquerda de um bloco: 2001:0:5ef5:79fb:0:0:0:1031;
- É permitido ainda a utilização de dois pontos, apenas uma única vez, para representar uma sequência consecutivas de zeros: 2001:0:5ef5:79fb::1031;
- Na utilização dos endereços IPv6 em URLs, esses passam a ser representados entre colchetes. Deste modo, não haverá ambiguidades caso seja necessário indicar o número de uma porta juntamente com a URL: http://[2001:0:5ef5:79fb::1031]:8080.

Existe a possibilidade de serem utilizados endereços que são formados por uma representação mista do IPv6 e IPv4, nos casos de ambientes que suportam as duas versões. Nestes casos o endereço IPv4 utiliza os quatro últimos bytes do IPv6 (HAGEN, 2014).

Exemplo de uma representação mista do IPv6 e IPv4:

0:0:0:0:0:0:192.168.0.2, ::192.168.0.2 ou ::C0A8:2

1.2.4 Tipos de Endereços

O protocolo IPv6 apresenta três tipos de endereço, sendo eles o *Unicast*, *Multicast* e *Anycast*. As novidades em relação ao IPv4 são, a implementação dos endereços *Anycast* e a eliminação dos endereços *Broadcast*, que foi considerado ineficiente, sendo então, substituído pelo uso dos endereços *Multicast* (REGHINI, 2014).

O endereço do tipo *Unicast* Identifica um host único e são utilizados para comunicação entre dois nós. Portanto, um pacote enviado a um endereço *unicast* será entregue a um único host ou interface. Os endereços *unicast* podem ser do tipo *Global Unicast, Unique-Local* ou *Link-Local.*

- *Global Unicast* são endereços públicos roteáveis na internet;
- *Unique-Local* são equivalentes aos endereços privados, utilizados em redes internas e não roteáveis na internet;
- *Link-Local* são endereços de interfaces locais usados em um mesmo host ou em uma sub rede.

Um endereço *multicast* é utilizado em comunicações onde é necessário enviar o mesmo pacote a vários hosts. São utilizados para identificar grupos de hosts. Os pacotes enviados para um endereço *multicast*, serão entregues a todos os que fazem parte do grupo (LOSHIN, 2004)

O *multicast* é parecido com o broadcast, já que permite o envio de um pacote hosts, diferenciando-se apenas pelo fato de que no broadcast o pacote é enviado a todos os hosts da rede, e no *multicast* apenas os hosts do grupo recebem o pacote.

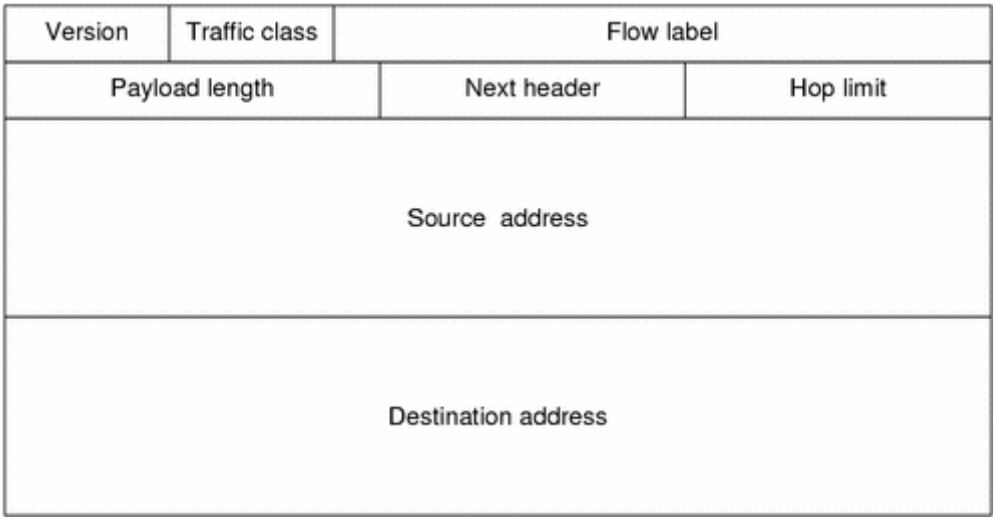
No endereço *Anycast*, muitos hosts poderão ter uma interface com o mesmo endereço. Porém, quando um pacote é enviado a um endereço *anycast* ele será entregue ao host mais próximo que possua uma interface com o endereço *anycast* de destino.

Usa-se o *anycast* para enviar um pacote a qualquer host dentro de um grupo. Este tipo de endereço pode ser utilizado para redundância de serviços e até balanceamento de carga em ambientes de alta disponibilidade.

1.2.5 Cabeçalho do IPv6

O cabeçalho do IPv6 possui um formato diferente do seu antecessor, os campos desta nova versão possuem tamanho fixo, sendo que seu tamanho total é de 64 bytes. Com um tamanho fixo, o processamento dos pacotes pelos roteadores é otimizado, visto que não há necessidade de calcular a extensão de certos campos, e nem o tamanho total do cabeçalho. Além disso, ocorreu uma redução do número de campos utilizados, por meio da exclusão de campos de pouca utilidade prática. O que também contribui para a diminuição do tempo gasto em processamento nos roteadores, foi a redução do número de campos com a eliminação de campos não essenciais e a utilização de cabeçalhos de extensão substituindo os campos de opção (ANTON, 1999).

Figura 2 – Cabeçalho do IPv6



Fonte: RFC 2460 (1998, p.4)

Smetana (2012), descreveu os campos que constituem então o cabeçalho IPv6 com base na RFC 2460:

- *Version* (4 bits) - Versão do protocolo IP utilizada, atualmente IPv6;
- *Traffic Class* (8 bits) - Permite diferenciar o tráfego utilizando classes e mecanismos de prioridade;

- *Flow label* (20 bits) - Identifica, juntamente com os campos *Source Address* e *Destination Address*, o fluxo ao qual o pacote pertence;
- *Payload Length* (16 bits) - Tamanho, em octetos, do restante do pacote, após o cabeçalho;
- *Next Header* (8 bits) - Indica qual o tipo do cabeçalho de extensão que segue o cabeçalho IPv6. Caso não esteja se utilizando cabeçalho de extensão, este campo indica a qual protocolo de transporte o pacote deve ser repassado;
- *Hop Limit* (8 bits) - Número máximo de saltos que o pacote pode realizar durante seu roteamento. Este valor é decrementado a cada salto e quando seu valor chega a zero o pacote é descartado;
- *Source Address* (128 bits) - Endereço do remetente;
- *Destination Address* (128 bits) - Endereço de destino.

1.2.6 Cabeçalhos de Extensão

Como o cabeçalho do IPv6 possui um valor fixo, informações adicionais necessárias para tarefas extras são especificadas nos cabeçalhos de extensão, que são inseridos após o cabeçalho IPv6. A utilização destes cabeçalhos permitiu simplificar o cabeçalho IPv6, com isso, campos de opções que existiam no cabeçalho IPv4 e que foram excluídos na nova versão, foram transformados em cabeçalhos de extensão (ANTON, 1999).

Uma grande vantagem dos cabeçalhos de extensão, é que, novos tipos de cabeçalho de extensão poderão ser criados conforme surgirem novas necessidades, permitindo uma maior flexibilidade para implementar novos recursos no futuro (ANTON, 1999).

A RFC 2460 (1998) estabelece seis tipos de cabeçalhos de extensão:

- *Hop-by-Hop options* - informações gerais a serem consideradas pelos roteadores;
- *Routing* - rota completa ou parcial a ser seguida;
- *Fragment* - gerenciamento de fragmentos de datagrama;
- *Authentication* - verificação da autenticidade do *payload* (utilizado no IPsec);

- *Encrypted security payload* - criptografia do *payload* (utilizado no IPsec);
- *Destination options* - informações adicionais a serem consideradas pelo destinatário.

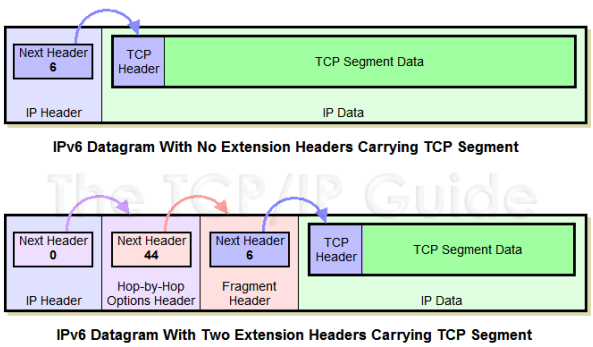
Conforme a RFC 2460 (1998), os cabeçalhos de extensão são utilizados entre o cabeçalho IPv6 e o *payload*. E quando mais do que um cabeçalho de extensão é usado no mesmo pacote, é recomendável que estes cabeçalhos aparecem na seguinte ordem:

- *IPv6 header;*
- *Hop-by-Hop Options header;*
- *Destination Options header;*
- *Routing header;*
- *Fragment header;*
- *Authentication header;*
- *Encapsulating Security Payload header;*
- *Destination Options header;*
- *Upper-layer header.*

O *Destination Options header* pode ser utilizado antes do *Routing header* ou antes do *Upper-layer header* dependendo da necessidade de processamento deste cabeçalho. O primeiro caso é utilizado para que as opções sejam processadas no primeiro destino, o segundo se utiliza para que as opções sejam processadas somente no destino final (RFC 2460, 1998).

A figura abaixo, apresenta exemplifica a utilização dos cabeçalhos de extensão:

Figura 3 - Cabeçalhos de Extensão



Fonte: Kozierok (2005, p. 408)

2 SEGURANÇA NA TRANSMISSÃO DE E-MAILS

Os dados enviados pela internet podem passar por muitas rotas diferentes e podem ser indevidamente capturados. Portanto existe a necessidade de proteger essas informações utilizando a criptografia. Assim, mesmo que os dados sejam capturados não poderão ser interpretados ou utilizados.

Na sequência, serão apresentados os protocolos que conferem segurança na rede durante a transmissão de e-mails via SMTP. Esses Protocolos são o TLS, *Transport Layer Security*, que adiciona criptografia na camada de aplicação, e o IPsec, *IP Security Protocol*, que provê criptografia na camada de rede.

2.1 TLS

O protocolo *Transport Layer Security* (TLS) atua na camada de transporte e provê segurança por criptografia para comunicação pela internet com o uso de certificados digitais. O TLS é o substituto do *Secure Socket Layer* (SSL), protocolo no qual foi baseado.

2.1.1 Características do TLS

O protocolo TLS tem a finalidade de prover os serviços de criptografia, autenticação e integridade de dados, fornecendo um canal seguro no meio inseguro. Ao criptografar as informações, pode-se garantir que somente o destino correto poderá ler o conteúdo enviado. Com a autenticação é possível verificar e garantir que as informações foram enviadas pelo remetente especificado. Com a integridade é verificado que os dados não foram alterados durante o processo de transferência. Esses serviços não precisam necessariamente serem utilizados sempre juntos no TLS, porém, visando garantir a segurança é recomendado que sejam utilizados (FERNANDES, 2014).

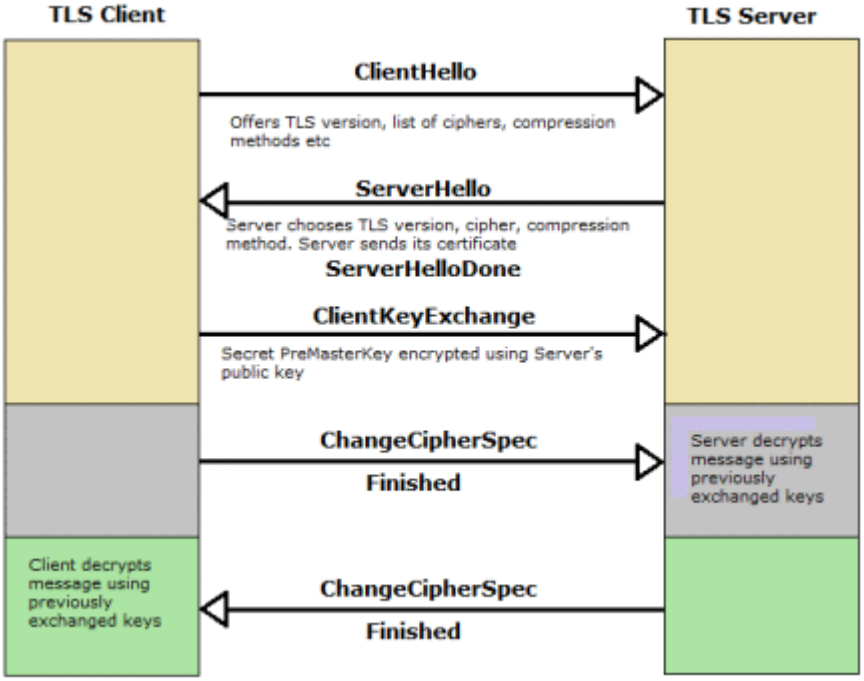
A proposta principal do TLS e dos protocolos de criptografia é estabelecer um canal seguro para transferência de dados pela internet. Para isso, as partes participantes da comunicação devem acordar sobre qual algoritmo de encriptação será usado e as chaves utilizadas. Para isso o TLS faz um *handshake* (negociação

ou aperto de mãos), onde ocorre a troca de chaves simétricas utilizando criptografia de chave pública (FERNANDES, 2014).

2.1.2 Funcionamento do TLS

Numa comunicação TLS a primeira etapa é realizar o *handshke*, onde as partes estabelecem uma conexão TCP e trocam mensagens onde informam o tipo de criptografia suportado para definir qual será utilizado, trocam as chaves criptográficas. A sequência de mensagens enviadas no *handshake* é mostrada na figura abaixo.

Figura 4 - Esquema de *Handshake* do TLS



Fonte: Sidhpurwala (2013)

O *ClientHello* é enviado para o servidor, fornecendo informações sobre sua configuração, como a versão do TLS que ele suporta, os métodos de encriptação suportados (*CipherSuites*) e uma lista dos tipos de compressão suportados.

O servidor responde com um *ServerHello*, nesta resposta ele informa o tipo de criptografia escolhido, um número gerado aleatoriamente e seu certificado digital.

O cliente gera uma chave aleatória, a *PreMasterSecret*, criptografada com a chave pública do certificado do servidor, então envia para o servidor. O envio desta chave é chamado de *ClientKeyExchange*.

O cliente e o servidor geram então a chave de sessão usando o número aleatório gerado no início do *handshake* e a *PreMasterSecret*, concluindo o *handshake*. A partir deste ponto, são enviados os dados criptografados.

A verificação da integridade das mensagens no TLS é feita utilizando algoritmo MAC, designado no *ServerHello*. Cada chave possui sua chave MAC, quem está enviando calcula o MAC da mensagem com sua chave MAC, então envia a mensagem criptografada junto com o MAC calculado. O receptor *descriptografa* a mensagem e ele mesmo calcula o MAC da mensagem recebida. Se o MAC calculado pelo receptor for igual ao recebido, a integridade está garantida (FERNANDES, 2014).

2.2 IPsec

IP Security Protocol (IPsec) assim como o TLS, é um protocolo que fornece privacidade, criando um canal seguro para transferência de informações através de um meio inseguro, a internet. Uma das grandes diferenças em relação ao TLS é que ele atua em nível de rede, garantindo confidencialidade, autenticação e integridade através da utilização de criptografia. Assim, o IPsec disponibiliza um alto nível de segurança na transmissão das informações, protegendo inclusive as informações da camada de transporte e até mesmo da própria camada de rede.

O IPsec provê um tunelamento ponto a ponto com autenticação mútua. Ele normalmente é utilizado para a comunicação entre servidores ou redes, onde é necessário um nível de segurança maior.

O IPv6 já possui suporte nativo ao IPsec, enquanto no IPv4 ele é um componente opcional.

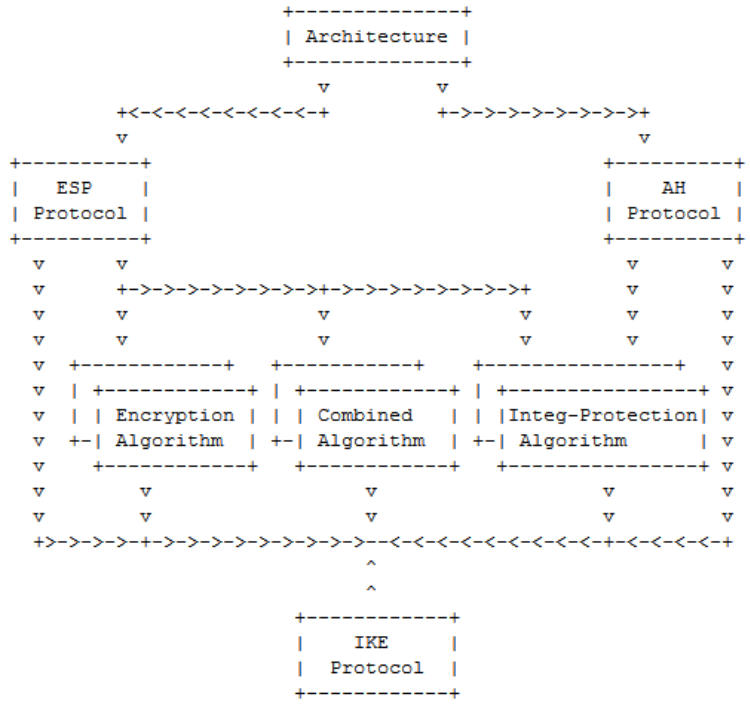
2.2.1 Arquitetura do IPSec

O IPsec foi projetado para fornecer segurança de alto nível nas comunicações, utilizando criptografia, sendo aplicado na camada de rede. Ele provê confidencialidade, autenticidade, integridade dos dados transmitidos.

Conforme Rotole e Rezende (2004), além de ser um padrão aberto IETF que está sendo adotado por todos os fabricantes de equipamentos de redes de computadores e desenvolvedores de sistemas, por definição o IPSec possui uma arquitetura aberta no sentido de possibilitar a inclusão de outros algoritmos de autenticação e criptografia.

Na RFC 6071 (2011), encontram-se as diretrizes para produção, organização e inter-relacionamento entre os documentos que são utilizados para descrever o conjunto de protocolos do IPSec. Na figura abaixo, podemos verificar o esquema da arquitetura geral do IPsec.

Figura 5 - Esquema da arquitetura geral do IPsec



Fonte: RFC 6071 (2011, p. 6)

No esquema da arquitetura apresentado acima temos o *roadmap a ser seguido por* especificações que descrevem o uso de algoritmos de autenticação e criptografia definidas pelo protocolo ESP (*Encapsulating Security Payload*), e de autenticação definidas pelo protocolo AH (*Authentication Header*) (ROTOLE; REZENDE, 2004).

2.2.2 Protocolos AH e ESP

Os serviços do IPsec funcionam com a aplicação de dois protocolos de segurança adicionando novos cabeçalhos nos pacotes IP, são eles o AH (*Authentication Header*) e o ESP (*Encapsulating Security Payload*). Esses protocolos são utilizados de forma complementar para potencializar a segurança proporcionada por cada um.

O AH é o cabeçalho responsável pela autenticação da origem do pacote, enquanto o ESP trata do encapsulamento criptográfico da mensagem (*payload)* e dos procedimentos e dos protocolos de gerenciamento de chaves.

Para garantir a interoperabilidade entre suas implementações, as especificações dos protocolos AH e ESP determinam que sejam suportados alguns algoritmos pré-definidos.

O protocolo AH (*Authentication Header*) precisa prover suporte aos algoritmos HMAC-MD5 e HMAC-SHA-1.

Já para o ESP (*Encapsulating Security Payload*), deve prover suporte aos algoritmos citados acima, bem como aos DES-CBC, *Null Authentication Algorithm* e *Null Encryption Algorithm*.

2.2.3 Gerenciamento de Chaves

O IPsec utiliza chaves criptográficas para criação dos túneis e o gerenciamento e autenticação destas chaves são fundamentais para o estabelecimento das conexões.

O gerenciamento de chaves do IPSec utilizando o protocolo IKE (*Internet Key Management*), que é uma associação do protocolo ISAKMP (*Internet Security Association and Key Management Protocol*) com o protocolo de Oakley.

O ISAKMP é um *framework* que especifica os serviços de autenticação e troca de chaves.

O Oakley é um protocolo que descreve o processo de negociação da troca de chaves. Este protocolo é baseado no algoritmo *Diffie-Hellman* em que as duas partes podem negociar uma chave mesmo sem utilizar a criptografia.

O IKE atua em duas fases. Primeiro, é estabelecido um canal seguro para realização das operações do ISAKMP entre as partes. Então, as partes negociam as associações de segurança (AS) para utilização pelo IPsec.

2.2.4 Modos de Funcionamento do IPSec

O IPsec disponibiliza dois modos de funcionamento, o modo de transporte e o modo túnel. No modo de transporte, apenas o conteúdo da mensagem é protegido, sendo que o cabeçalho IP não é alterado. Já no modo túnel, o pacote IP é totalmente protegido, sendo criptografado e recebendo um novo cabeçalho IP, assim só ficará visível o destino final do pacote.

O modo de transporte é utilizado na proteção das conexões entre apenas dois hosts, enquanto que o modo túnel pode ser configurado entre dispositivos de borda em redes distintas, e assim protege todo o tráfego entre estas redes.

Também é possível utilizar os dois modos de funcionamento combinados, podendo ser utilizado o modo túnel para autenticar o pacote da rede interna e depois utilizar o modo transporte para criptografar o pacote resultante (BRAGHETTO et al., 2003).

3 PROPOSTA DE INSTALAÇÃO E METODOLOGIA

Para realização dos testes foi criado um ambiente com duas máquinas virtuais utilizando os devidos recursos para implementação das tecnologias pesquisadas. Estes recursos, suas configurações e a metodologia utilizada na pesquisa serão detalhados neste capítulo.

Para alcançar o objetivo deste trabalho foram utilizadas duas máquinas virtuais do Oracle Virtual Box com 1 processador e 512MB de memória nos quais foram configurados IPv4 e IPv6 fixos; Depois foram instalados os servidores de SMTP Postfix, clientes de SMTP para envio e verificação do recebimento; Foi configurado o IPsec para IPv4 e IPv6 com o IPSec-Tools; E para concluir a preparação do ambiente de testes foi instalado o analisador de rede Wireshark para realizar a coleta de evidências.

Após a preparação do ambiente, foram realizados diversos testes para verificação do funcionamento, coleta de evidências e análise do desempenho/tempo de entrega de mensagens com diversos tamanhos.

3.1 Recursos Utilizados

Para realização dos testes foi criado um ambiente com duas máquinas virtuais utilizando os devidos recursos para implementação das tecnologias pesquisadas.

3.1.1 Virtualizador Oracle Virtual Box

O Virtual Box permite, através da virtualização, instalar e executar diferentes sistemas operacionais em um único computador, possibilitando criar um ambiente completo onde pode ser configurada, inclusive uma rede interna simulando ambientes físicos reais.

Porém, logicamente a criação de hosts e disponibilidade de recursos como CPU, memória RAM e disco, fica limitada aos recursos disponíveis na máquina hospedeira.

3.1.2 Sistema Operacional Linux Debian

Nos servidores virtuais utilizamos o sistema operacional Debian, que já possui suporte nativo ao IPv6.

O Debian foi escolhido por vários motivos, entre os principais podemos citar os seguintes: é um SO livre, suporta todas as tecnologias necessárias à esta pesquisa, está disponível para download na internet, disponibilidade e facilidade de instalar os pacotes, é muito utilizado em servidores corporativos.

Foi realizada instalação com as configurações padrão sem interface gráfica para otimizar os recursos. A configuração das interfaces de rede será detalhada em item específico.

3.1.3 Servidor SMTP Postfix

O Postfix é um agente de transferência de e-mails (MTA) livre e de código aberto que, encaminha e entrega e-mails implementando o protocolo SMTP, e foi desenvolvido para ser uma alternativa segura ao Sendmail.

Criado em 1997, ainda hoje continua a ser ativamente desenvolvido e é o MTA padrão de inúmeros sistemas operacionais.

Foi escolhido por ser simples, de fácil instalação e configuração. Além de ter suporte nativo ao IPv6 e TLS.

3.1.4 Cliente SMTP Mutt

O Mutt é um cliente de e-mail para Linux muito leve e com interface textual. Ele é útil para enviar as mensagens inclusive via linha de comando e também para verificar se as mensagens foram entregues.

Para enviar/conferir as mensagens é só acessar o servidor com seu usuário local e utilizar o comando “mutt” para entrar na interface do programa.

Usamos configuração padrão, é só instalar com o “*apt-get install mutt*”.

3.1.5 Gerenciador do IPsec IPsec-Tools

IPsec-Tools é um conjunto de ferramentas para a implementação do IPsec Linux 2.6.

Os seguintes utilitários estão disponíveis no IPsec-Tools:

- *libipsec*: Biblioteca com implementação PF_KEY;
- *setkey*: Ferramenta para manipular e despejar o banco de dados Política de núcleo de segurança (SPD) e banco de dados *Security Association* (SAD);
- *guaxinim: Internet Key Exchange* (IKE) *daemon* para introduzir automaticamente as conexões IPsec;
- *racoonctl*: Uma ferramenta de controle baseado em *shell de guaxinim*.

3.1.6 Analisador de protocolos de rede Wireshark

O Wireshark é programa que permite a análise dos pacotes transmitidos na rede, capturando todo o tráfego da rede e mostrando de forma organizada em uma interface gráfica. Entre os muitos recursos destaca-se a possibilidade e facilidade de aplicação de filtros para analisar pacotes específicos.

Usamos configuração padrão, instalando com o "*apt-get install wireshark*". Para capturar o tráfego da rede utilizando o Wireshark. é só iniciar o programa com "sudo wireshark" selecionar uma ou todas interfaces da máquina e iniciar a captura.

Se você estiver usando um servidor remoto em modo gráfico, deverá exportar o display da máquina para acessar a interface gráfica do Wireshark.

3.2 Configuração dos Recursos

Foi necessário a configuração dos recursos e diversas variáveis para prover o ambiente propício aos testes desejados.

Alguns dos recursos/softwares utilizados, não precisaram de configuração adicional. Portanto, neste item serão apresentadas a configuração dos recursos que precisaram ser personalizados.

3.2.1 Configuração das Interfaces de Rede dos Servidores

Foram utilizados endereços IP fixos para as máquinas. Para isso o arquivo /etc/network/interfaces foi alterado em cada máquina, inserindo os respectivos endereços de IPv4 e IPv6.

O quadro abaixo mostra a configuração da interface eth0 na máquina 1 que recebeu o IPv4 192.168.0.13 e o IPv6 fc00::1003.

Quadro 3 - Arquivo /etc/network/interfaces máquina 1

```
#/etc/network/interfaces Configuração rede máquina 1
# Configuração de loopback
auto lo
iface lo inet loopback

# Configuração IPv4 fixo na eht0
auto eth0
iface eth0 inet static
address 192.168.0.13
netmask 255.255.255.0
broadcast 192.168.0.255
gateway 192.168.0.1

# Configuração IPv6 fixo na eht0
iface eth0 inet6 static
address fc00::1003
netmask 64
```

Fonte: Produzido pelo autor do trabalho

Na sequência, o próximo quadro mostra a configuração da interface eth0 na máquina 2 que recebeu o IPv4 192.168.0.14 e o IPv6 fc00::1004.

Quadro 4 - Arquivo /etc/network/interfaces máquina 2

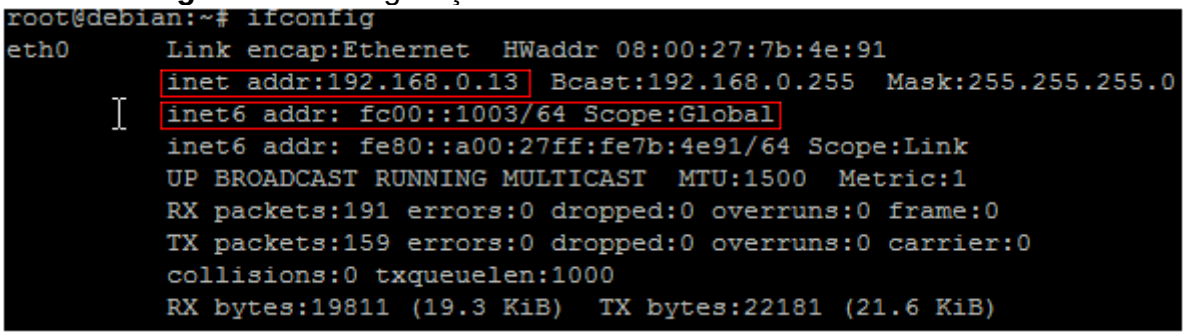
```
#/etc/network/interfaces Configuração rede máquina 2
# Configuração de loopback
auto lo
iface lo inet loopback

# Configuração IPv4 fixo na eht0
auto eth0
iface eth0 inet static
address 192.168.0.14
netmask 255.255.255.0
broadcast 192.168.0.255
gateway 192.168.0.1

# Configuração IPv6 fixo na eht0
```

```
iface eth0 inet6 static
address fc00::1004
netmask 64
```

Fonte: Produzido pelo autor do trabalho

Colocada a configuração, o serviço de rede foi reiniciado com o comando *service networking restart* e a configuração conferida com o *ifconfig:*

Figura 6 – Configuração de rede



Fonte: Produzido pelo autor do trabalho.

A conectividade entre os servidores foi testada utilizando o comando *ping* para IPv4 e *ping6* para IPv6.

3.2.2 Instalação e Configuração do Postfix

O Postfix foi instalado nas duas máquinas usando o apt-get do Debian: "*apt-get install postfix".* A instalação disponibiliza um assistente de configuração, mas foi utilizada a opção "*No configuration*" para realizar a configuração manualmente através do arquivo /etc/postfix/main.cf de cada servidor.

Como os testes de envio de mensagens serão feitos para contas de domínios cujos servidores do teste estão como responsáveis, não foi necessário a configuração de um DNS, adicionando os endereços dos hosts envolvidos nos arquivos "/ect/hosts" de cada máquina.

Para utilização do TLS devem ser configurados os parâmetros "smtp_tls" na configuração do Postfix.

Para garantir a criptografia da mensagem até que seja armazenada é necessário configurar também os parâmetros "smtpd_tls". O smtpd trata das

mensagens cujas caixas destinatárias estão no servidor, são mensagens para o domínio local.

Uma vez configurado, para ativar ou desativar o TLS, é só atribuir valor "*yes*" aos parâmetros "smtp_use_tls" e "smtpd_use_tls".

Para configuração do TLS, foram utilizados certificados auto assinados. Estes certificados devem ser inseridos como confiáveis nos servidores para o funcionamento do TLS.

O quadro abaixo apresenta a configuração do TLS:

Quadro 5 - Configuração TLS Postfix

```
#/etc/postfix/main.cf Configuração Postfix

# Parâmetros do TLS
smtpd_tls_CAfile = /etc/postfix/certs/ussca.pem
smtpd_tls_cert_file = /etc/postfix/certs/debian-cert.pem
smtpd_tls_key_file = /etc/postfix/certs/debian-key.pem
smtpd_tls_session_cache_database =
btree:${data_directory}/smtpd_scache
smtpd_tls_auth_only=no
smtpd use tls=yes

smtp_tls_CAfile = /etc/postfix/certs/ussca.pem
smtp_tls_cert_file = /etc/postfix/certs/debian-cert.pem
smtp_tls_key_file = /etc/postfix/certs/debian-key.pem
smtp_tls_session_cache_database =
btree:${data_directory}/smtp_scache
smtp_tls_auth_only=no
smtp_tls_received_header = yes
smtp_tls_security_level=may
smtp use tls=yes
```

Fonte: Produzido pelo autor do trabalho

Para ativar a utilização do IPv6, é necessário configurar acrescentar o valor "IPv6" ao campo "inet_protocols". Adicionalmente devem ser adicionados os endereços IPv6 aos campos "relayhost", "mynetworks" e smtp_bind_address6.

O quadro abaixo mostra a configuração do IPv6 no Postfix:

Quadro 6 - Configuração IPv6 Postfix

```
#/etc/postfix/main.cf Configuração Postfix

# Parâmetros do IPv6
```

```
relayhost = [FC00::1003] # IPv6 de destino do SMTP Relay

mynetworks   =   127.0.0.0/32   192.168.0.0/24   [fe80::]/64
[fc00::]/64 [::1]/128 # Redes confiáves

inet_protocols = IPv4, IPv6

smtp_bind_address6 = FC00::1004
```

Fonte: Produzido pelo autor do trabalho

Após concluída toda configuração, o Postfix deverá atender tanto no IPv4 quanto no IPv6. Uma observação é que o endereço IPv6 já deve ter sido configurado como estático no SO para o correto funcionamento do Postfix.

No quadro abaixo está demonstrada a configuração completa para que o Postfix atenda com IPv6 e IPv4.

Quadro 7 - Configuração Postfix

```
#/etc/postfix/main.cf Configuração Postfix máquina 2

# Parâmetros do TLS
smtpd_tls_CAfile = /etc/postfix/certs/ussca.pem
smtpd_tls_cert_file = /etc/postfix/certs/debian-cert.pem
smtpd_tls_key_file = /etc/postfix/certs/debian-key.pem
smtpd_tls_session_cache_database =
btree:${data_directory}/smtpd_scache
smtpd_tls_auth_only=no
smtpd_use_tls=yes

smtp_tls_CAfile = /etc/postfix/certs/ussca.pem
smtp_tls_cert_file = /etc/postfix/certs/debian-cert.pem
smtp_tls_key_file = /etc/postfix/certs/debian-key.pem
smtp_tls_session_cache_database =
btree:${data_directory}/smtp_scache
smtp_tls_auth_only=no
smtp_tls_received_header = yes
smtp_tls_security_level=may
smtp_use_tls=yes


# Parâmetros do IPv6
relayhost = [FC00::1003] #Endereço IPv6 do SMTP Relay
#relayhost = 192.168.0.13 #Endereço IPv4 do SMTP Relay
```

```
mynetworks = 127.0.0.0/32 192.168.0.0/24 [fe80::]/64
[fc00::]/64 [::1]/128
inet_interfaces = all
inet_protocols = IPv4, IPv6
smtp_bind_address6 = FC00::1004

# Parâmetros do domínio
myhostname = debian2.uss.com.br
myorigin = /etc/mailname
mydestination = uss.com.br, debian2.uss.com.br,
localhost.uss.com.br, localhost
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

# Demais Parâmetros
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no
append_dot_mydomain = no
readme_directory = no
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
message_size_limit = 0
recipient_delimiter = +
```

Fonte: Produzido pelo autor do trabalho

Após a configuração, é necessário reiniciar o Postfix com o comando "*service postfix restart*".

Pode-se verificar se o servidor está atendendo na porta 25 no IPv4 e IPv6 com o comando "*netstat -anp |grep :25*", o servidor deve estar *LISTEN*:

Figura 7 – Verificação do serviço SMTP

```
root@debian:~# netstat -anp |grep :25
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN
tcp6       0      0 :::25                   :::*                    LISTEN
```

Fonte: Produzido pelo autor do trabalho

3.2.3 Configuração do IPsec

O IPSec será utilizado no modo de transporte com os cabeçalhos de extensão AH e ESP. No modo de transporte, apenas os dados e cabeçalhos das camadas superiores serão criptografados, o cabeçalho IP é mantido. O cabeçalho AH garante a autenticidade e integridade do pacote e o ESP a confidencialidade, criptografando os dados.

Para configuração do IPsec foi utilizado o IPSec-Tools que também pode ser instalado com o apt-get: "*apt-get install ipsec-tools*". A configuração é realizada com a edição do arquivo "/etc/ipsec-tools.conf".

Antes de configurar o IPsec em cada máquina, precisamos gerar uma chave aleatória AH e ESP para cada máquina com os seguintes comandos:

- Chave AH: "`dd if=/dev/random count=16 bs=1| xxd -ps`"

Figura 8 – Exemplo da geração da chave AH

```
root@debian:~# dd if=/dev/random count=16 bs=1 | xxd -ps
c33e733e08ded27c2856cda87bc9b7b9
16+0 records in
16+0 records out
16 bytes (16 B) copied, 0.0223626 s, 0.7 kB/s
```

Fonte: Produzido pelo autor do trabalho

- Chave ESP: "`dd if=/dev/random count=24 bs=1| xxd -ps`"

Figura 9 – Exemplo da geração da chave ESP

```
root@debian:~# dd if=/dev/random count=24 bs=1 | xxd -ps
93985258323afaa6c5fc1a3152d056f32d2781c72dc9ccd4
24+0 records in
24+0 records out
24 bytes (24 B) copied, 0.0252916 s, 0.9 kB/s
```

Fonte: Produzido pelo autor do trabalho

Após geradas as chaves aleatórias AH e ESP de cada máquina foram colocadas no arquivo "/etc/ipsec-tools.conf" precedidos de "0x".

O quadro abaixo mostra a configuração do arquivo /etc/ipsec-tools.conf na máquina 1, onde as políticas estabelecem como obrigatórios os cabeçalhos de extensão AH e ESP. Observe o valor "0x" antecedendo as chaves AH e ESP:

Quadro 8: Configuração IPsec máquina 1 usando AH/ESP

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

# /etc/ipsec-tools.conf
```

```
# Configuração IPsec máquina 1 usando AH/ESP

flush;
spdflush;
#chave AH da máquina 1
add FC00::1003 FC00::1004 ah 0x200 -A hmac-md5
0x0119efa254d2e13eaece6d03eba36bbe;
#chave ESP da máquina 1
add FC00::1003 FC00::1004 esp 0x201 -E 3des-cbc
0xfbfc7ab536247dd210f8a1db5aeab46a4905ff9c42965312;
#chave AH da máquina 2
add FC00::1004 FC00::1003 ah 0x300 -A hmac-md5
0x41280a1f3a61d515a9d139ffb70caafd;
#chave ESP da máquina 2
add FC00::1004 FC00::1003 esp 0x301 -E 3des-cbc
0x947d47c4f89731c2f35edc8abee2855e64a905535929c253;

#Políticas de Segurança
spdadd FC00::1003 FC00::1004 any -P out ipsec
esp/transport//require
ah/transport//require;
spdadd FC00::1004 FC00::1003 any -P in ipsec
esp/transport//require
ah/transport//require;
```

Fonte: Produzido pelo autor do trabalho

O quadro abaixo mostra a configuração do arquivo /etc/ipsec-tools.conf na máquina 2, onde podemos observar que em relação à configuração da máquina 1 a diferença é sutil, sendo alteradas apenas as linhas referente às políticas de segurança.

Quadro 9: Configuração IPsec máquina 2 usando AH/ESP

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

# /etc/ipsec-tools.conf
# Configuração IPsec máquina 2 usando AH/ESP

flush;
spdflush;
#chave AH da máquina 1
add FC00::1003 FC00::1004 ah 0x200 -A hmac-md5
0x0119efa254d2e13eaece6d03eba36bbe;
#chave ESP da máquina 1
add FC00::1003 FC00::1004 esp 0x201 -E 3des-cbc
0xfbfc7ab536247dd210f8a1db5aeab46a4905ff9c42965312;
#chave AH da máquina 2
add FC00::1004 FC00::1003 ah 0x300 -A hmac-md5
0x41280a1f3a61d515a9d139ffb70caafd;
#chave ESP da máquina 2
```

```
add FC00::1004 FC00::1003 esp 0x301 -E 3des-cbc
0x947d47c4f89731c2f35edc8abee2855e64a905535929c253;

#Políticas de Segurança
spdadd FC00::1004 FC00::1003 any -P out ipsec
esp/transport//require
ah/transport//require;
spdadd FC00::1003 FC00::1004 any -P in ipsec
esp/transport//require
ah/transport//require;
```

Fonte: Produzido pelo autor do trabalho

Após a configuração, o serviço “setkey” deve ser iniciado usando o comando “*service setkey start*” e a comunicação entre as duas máquinas passa a utilizar o IPsec.

A configuração do IPsec para o IPv4 é a mesma, bastando trocar os endereços IPv6 pelos IPv4 nos arquivos de configuração e reiniciar o serviço "setkey" com o comando “*service setkey restart*”.

3.3 Realização dos Testes

Com o ambiente devidamente configurado, foram feitos testes para verificação do funcionamento e análise do desempenho/tempo de entrega de mensagens SMTP comparando o uso do IPv6 ao IPv4 com TLS e IPsec.

Foram realizados testes com o envio de mensagens de 1KB, 5MB, 10MB, 20MB e 40MB utilizando as seguintes configurações:

- IPv4 sem Criptografia;
- IPv4 com TLS;
- IPv4 com IPsec;
- IPv4 com IPsec e TLS;
- IPv6 sem Criptografia;
- IPv6 com TLS;
- IPv6 com IPsec;
- IPv6 com IPsec e TLS.

Realizamos os testes utilizando IPsec e TLS visando uma situação de extrema segurança. Essa configuração também é útil em casos onde não é possível

utilizar o IPsec fim a fim, como nos casos em que o IPsec só é aplicado aos dispositivos de borda entre duas redes.

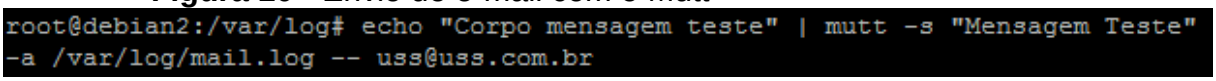
Para fins de referência, também foram realizados testes com o ping (Protocolo ICMP) entre as máquinas.

Utilizando o cliente de SMTP Mutt as mensagens são enviadas da máquina local do usuário para o servidor SMTP na máquina 1, uma vez que o servidor 1 não é o dono do domínio de destino, ele encaminha as mensagens para o servidor 2, que entregará a mensagem ao destino.

O envio de mensagens com o Mutt é realizado via linha de comando no terminal linux seguindo a seguinte sintaxe: *echo "Corpo da mensagem" | mutt -s "Assunto" -a /local/anexo.txt -- endereço_destino@domínio.destino.*

Na figura abaixo temos um exemplo de envio de e-mail utilizando o Mutt.

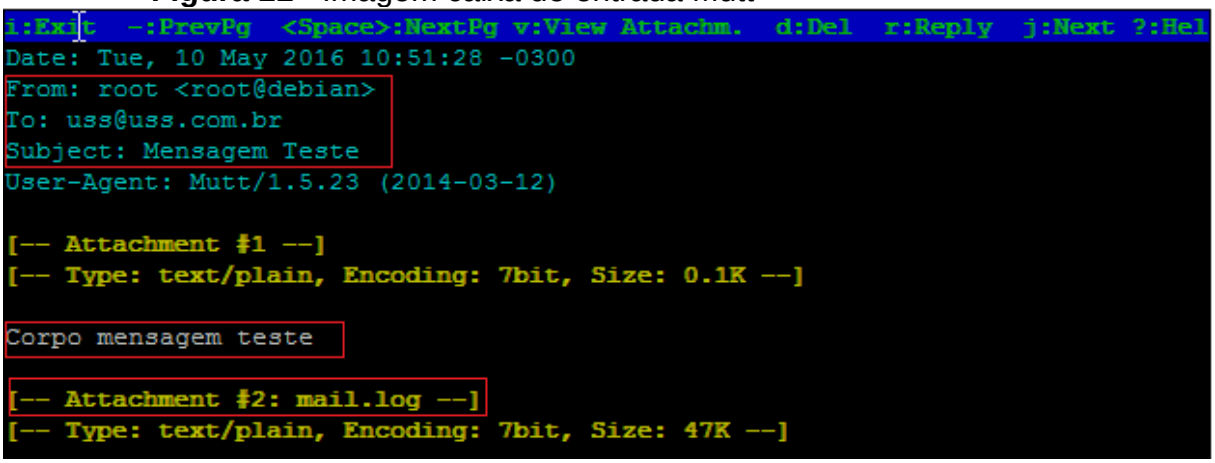
Figura 10 - Envio de e-mail com o Mutt



Fonte: Produzido pelo autor do trabalho

A mensagem pode ser conferida fazendo logon com o usuário de destino utilizando o próprio Mutt, como mostra a figura abaixo.

Figura 11 - Imagem caixa de entrada Mutt



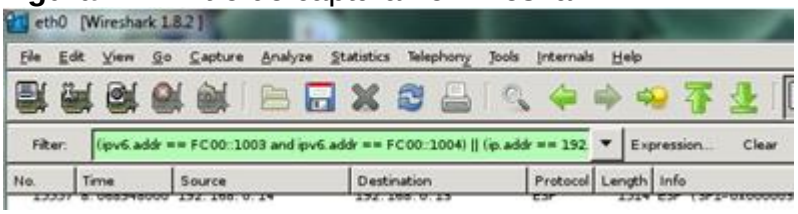
Fonte: Produzido pelo autor do trabalho

Para cada teste realizado foi feita a captura do tráfego de rede e análise utilizando o Wireshark.

Durante a coleta, foram aplicados filtros para mostrar apenas o tráfego entre as máquinas de teste. Exemplo: (IPv6.addr == FC00::1003 and IPv6.addr == FC00::1004) || (ip.addr == 192.168.0.13 and ip.addr == 192.168.0.14).

A figura abaixo ilustra a utilização do filtro na figura abaixo.

Figura 12 - Filtro de captura no Wireshark

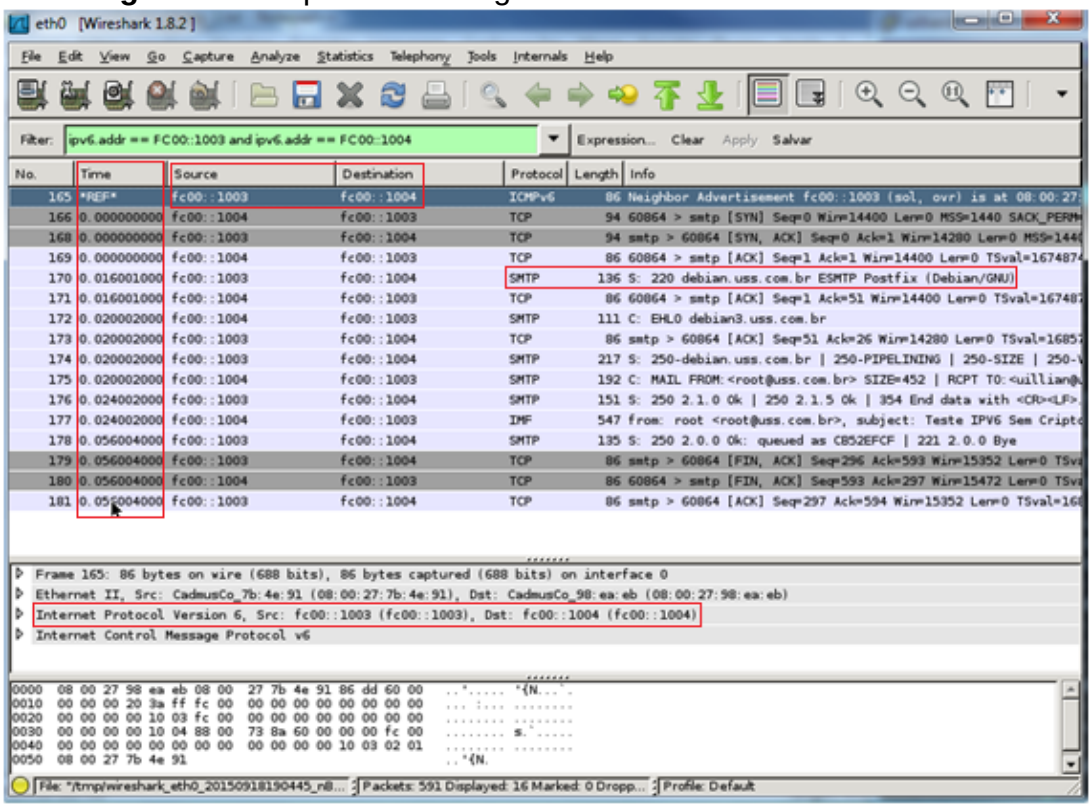


Fonte: Produzido pelo autor do trabalho

Os tempos foram computados desde abertura até o encerramento das conexões. As informações obtidas durante o teste serão apresentadas e analisadas na seção de análise e resultados.

A figura abaixo demonstra a captura do tráfego da rede durante o envio de uma mensagem. Podemos verificar neste exemplo, que foi utilizado IPv6, que estavam sendo transmitidos os pacotes SMTP, os dados não estavam criptografados e o tempo de duração da transferência foi de 0,05 segundos.

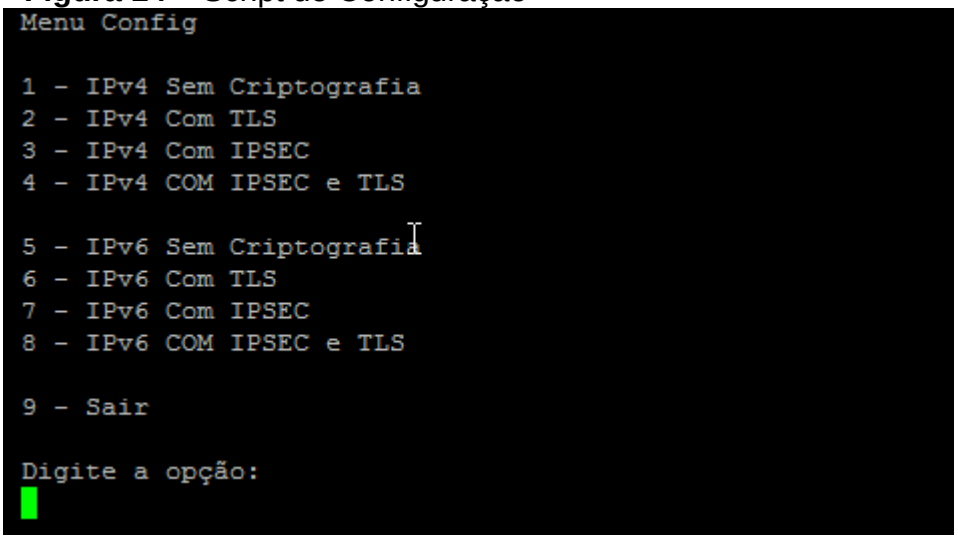
Figura 13 - Captura do tráfego da rede com Wireshark



Fonte: Produzido pelo autor do trabalho

Como facilitador para alternar entre as configurações durante a realização dos testes, foi criado um *shell-script* para automatizar este processo. Quando a configuração desejada é escolhida, ela será aplicada automaticamente nas máquinas do ambiente. A figura abaixo ilustra o script de configuração.

Figura 14 – Script de Configuração



Fonte: Produzido pelo autor do trabalho

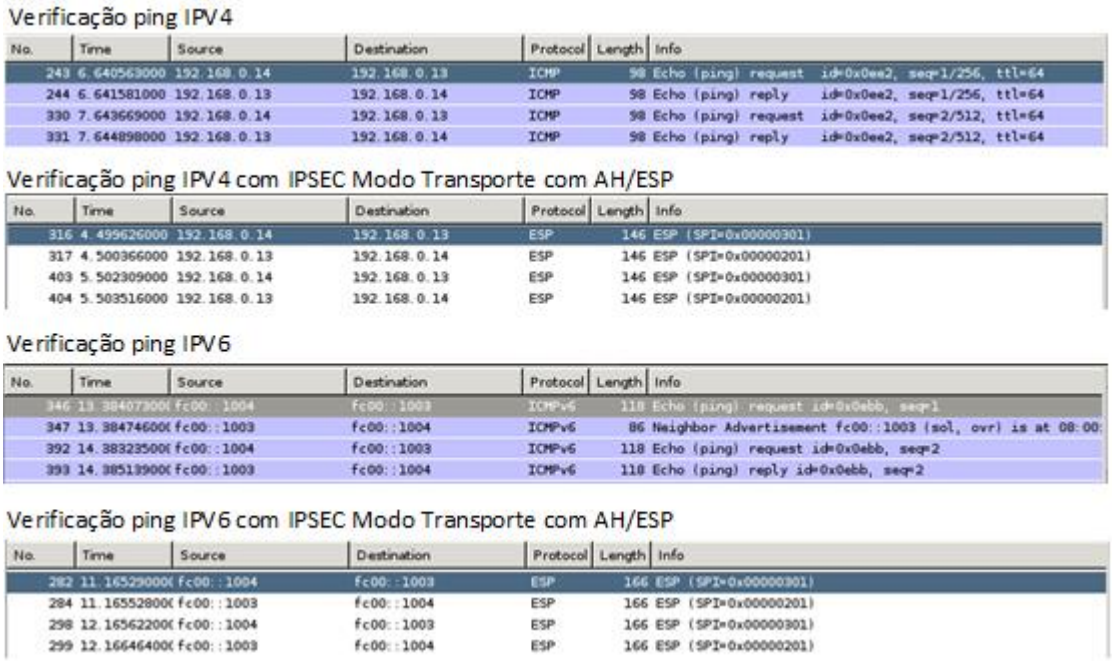
4 RESULTADOS E ANÁLISE FINAL

Esta seção apresenta os resultados verificados e as respectivas análises para as situações e comportamentos observados.

Neste primeiro momento será apresentado o resultado do teste de rede com o protocolo IPCMP (*Ping*), ou seja, desconsiderando a camada de aplicação.

A figura abaixo mostra o tráfego durante a transmissão do *ping* com IPv4 e IPv6.

Figura 15 – Tráfego durante teste de *ping*



Fonte: Produzido pelo autor do trabalho

A tabela abaixo mostra uma comparação dos resultados de testes com o ping (Protocolo ICMP) entre o IPv4 e IPv6 com e sem IPsec, onde podemos verificar que apesar dos pacotes serem maiores no IPv6, o tempo médio de resposta do IPv6 foi praticamente o mesmo do IPv4. O que podemos observar é que com o uso do IPsec os pacotes têm significativo aumento de tamanho e consequentemente maior tempo de resposta.

Tabela 1 - Testes ICMP (*ping*)

	Testes ICMP (*ping*)	
	Tamanho do pacote enviado	**Tempo médio de resposta**
IPv4	98	0,837

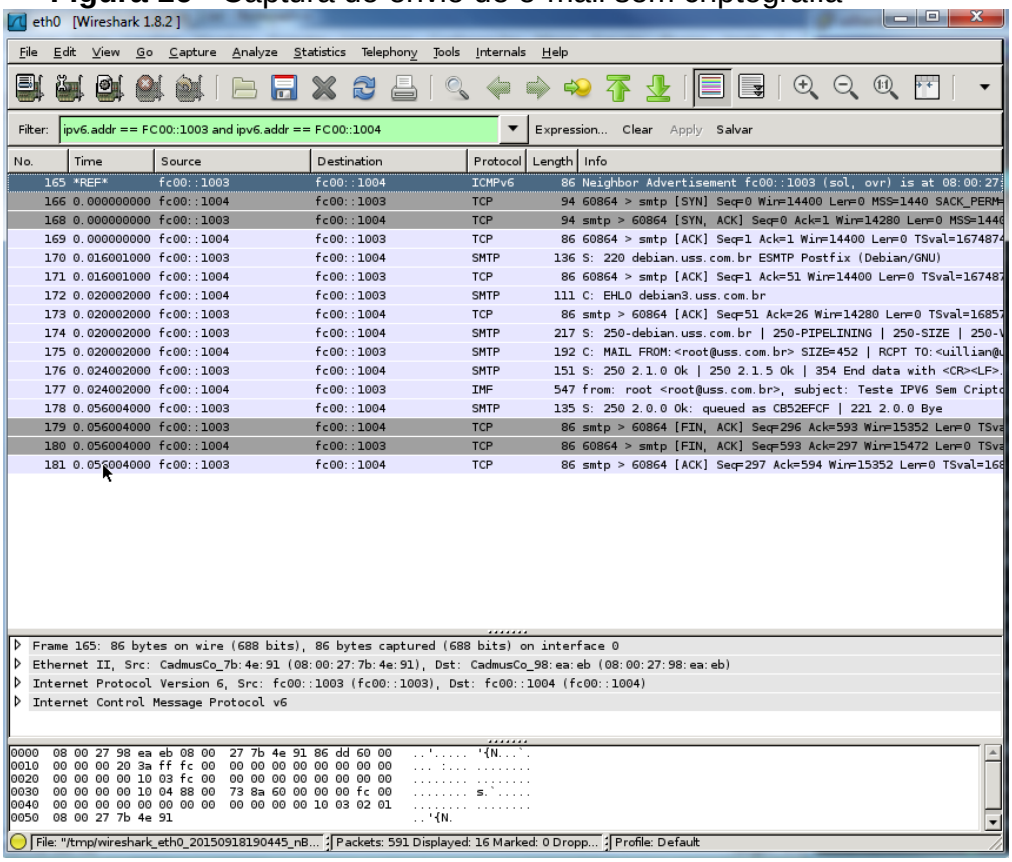
IPv4 com IPsec	146	0,938
IPv6	118	0,835
IPv4 com IPsec	166	0,926

Fonte: Produzido pelo autor do trabalho

Conforme demonstrado no quadro acima, com o IPsec foram acrescidos 48 bytes ao pacote enviado, sendo que neste caso são 24 bytes do cabeçalho AH e 24 bytes do cabeçalho ESP. Conforme explicado anteriormente, o IPSec foi utilizado no modo de transporte com os cabeçalhos de extensão AH e ESP.

Na figura abaixo podemos observar a captura do tráfego de rede durante o envio de e-mail (SMTP) com IPv6 e sem a utilização criptografia. Neste caso, os dados enviados e recebidos podem ser facilmente explorados. Ainda na figura abaixo também é possível validar claramente a sequência de comandos do SMTP.

Figura 16 - Captura do envio de e-mail sem criptografia



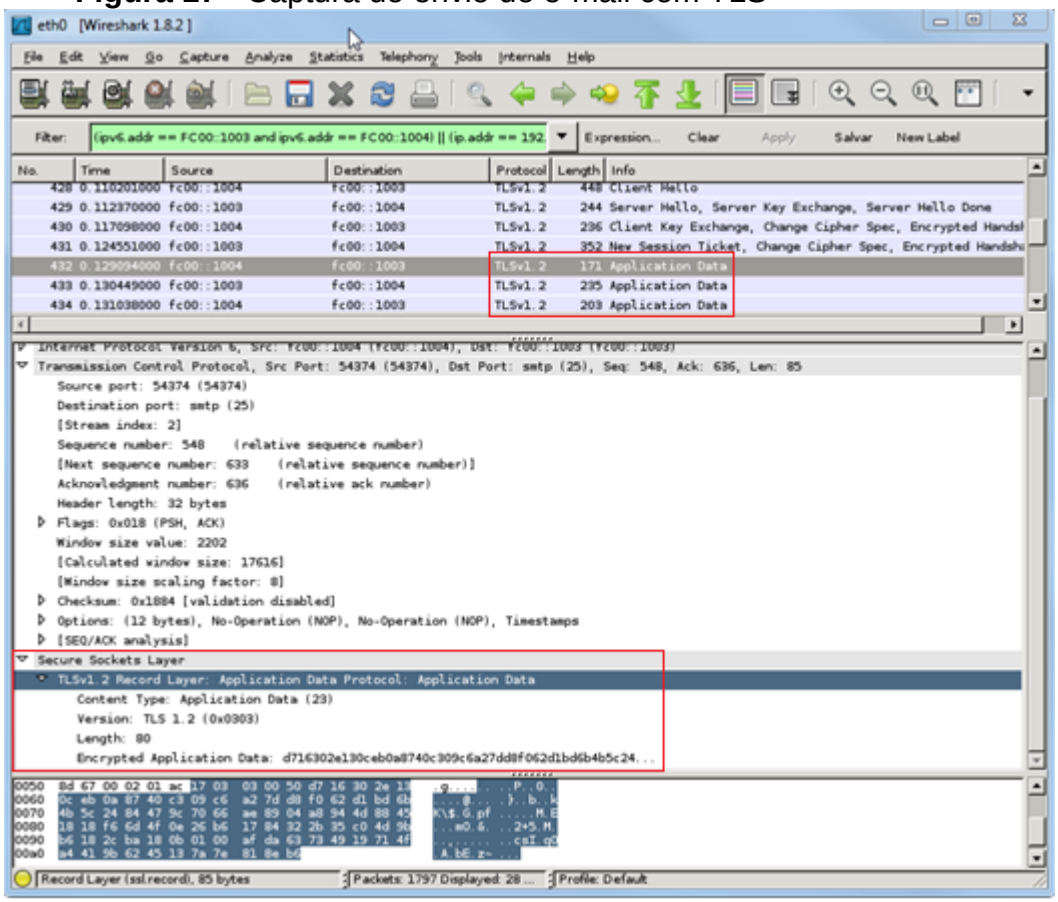
Fonte: Produzido pelo autor do trabalho

Conforme anteriormente explicado, os dados enviados pela internet podem passar por muitas rotas diferentes e podem ser capturados indevidamente,

portanto existe a necessidade de proteger essas informações utilizando a criptografia.

Nesta próxima figura, está um exemplo de envio usando TLS, criptografia na camada de aplicação. Neste caso, a conexão é aberta na porta do SMTP, o comando “EHLO” é enviado e o servidor responde que tem suporte para o TLS “STARTTLS”, o cliente inicia uma sessão TLS com o comando “STARTTLS” e só a partir deste ponto os dados serão criptografados. Então, na criptografia com TLS, é possível ver algumas informações, como o tipo de pacote que está passando, versão do TLS utilizado, domínio de origem e destino, mas não é mais possível ver a sequência do SMTP ou mesmo os dados enviados no e-mail.

Figura 17 - Captura do envio de e-mail com TLS



Fonte: Produzido pelo autor do trabalho

O exemplo acima, a criptografia utilizada é provida no nível de aplicação e ainda foi possível observar que tipo de aplicação enviou os pacotes e até a negociação antes de se estabelecer a conexão criptografada.

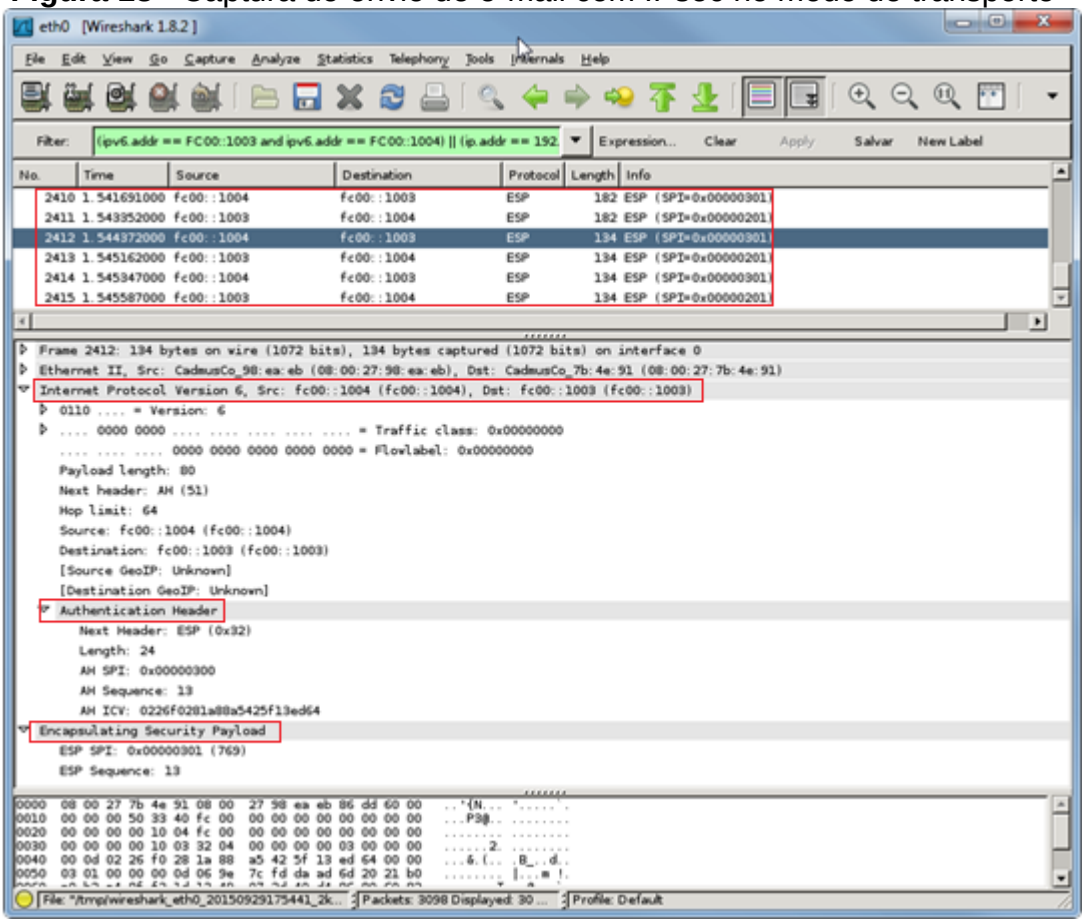
No próximo exemplo, é demonstrada a criptografia na camada de rede onde toda a transmissão, desde o início, será criptografada.

A próxima figura apresenta a captura durante o envio de e-mail utilizando o IPsec com IPv6. Devido à criptografia em nível de rede, não são mostradas informações do pacote como o tipo de aplicação ou mesmo o protocolo de transporte usado. Com a utilização do IPsec o aumento de segurança é muito grande, um ataque fica muito difícil quando o atacante não sabe sequer o que está sendo transmitido.

Podemos verificar no exemplo da figura abaixo em uma captura só é apresentado o protocolo ESP, que é um cabeçalho do IPsec.

Como estamos utilizando o modo de transporte, o cabeçalho ip não foi criptografado, o que permite identificar o ip de origem e destino do pacote. Também é possível ver o cabeçalho de autenticação do IPsec, o AH, utilizado para verificação da autenticidade do pacote.

Figura 18 - Captura do envio de e-mail com IPsec no modo de transporte

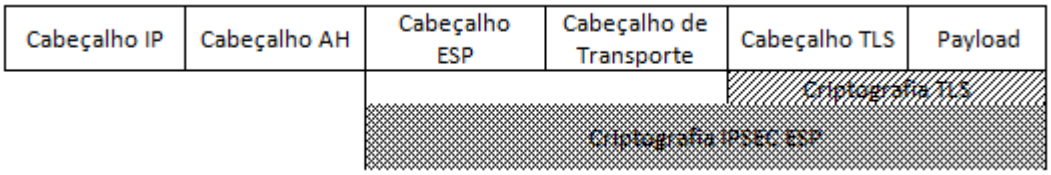


Fonte: Produzido pelo autor do trabalho

Os testes mostraram a possibilidade de utilizar TLS e IPsec juntos, uma vez que esses protocolos atuam em camadas diferentes. Esta configuração possibilita aumentar ainda mais a segurança da transmissão. Essa é uma opção em casos onde não é possível utilizar o IPsec fim a fim, como nos casos em que o IPsec só é aplicado aos dispositivos de borda entre duas redes.

Como na captura só é possível observar até o cabeçalho ESP, segue na figura abaixo o esquema de um pacote quando utilizado TLS e IPsec.

Figura 19 - TLS com IPsec Modo de transporte



Fonte: Produzido pelo autor do trabalho

Observamos uma alta de garantia de segurança proporcionada pelas opções de criptografia demonstradas. Porém, se a segurança aumenta com o uso da criptografia, o processamento e o tempo de resposta também aumentam.

Baseado nas evidências coletadas nos testes, as informações foram agrupadas na tabela a seguir, que apresenta a consolidação dos resultados dos testes de uso do SMTP, onde são apresentados os tempos em segundos (s) da abertura ao encerramento das conexões para envio de e-mails com SMTP:

Tabela 2 - Consolidação dos resultados dos testes

Testes SMTP - Tempo para entrega de E-mails (s)					
	1KB	**5MB**	**10MB**	**20MB**	**40MB**
IPv4 sem criptografia	0,05	1,79	3,22	6,89	16,48
IPv4 com TLS	0,10	3,53	5,87	11,80	28,60
IPv4 com IPsec	0,09	4,56	8,17	17,64	38,54
IPv4 com IPsec e TLS	0,11	5,35	10,48	22,97	53,39
IPv6 sem criptografia	0,05	1,84	3,51	7,81	21,20
IPv6 com TLS	0,13	4,08	8,30	17,37	29,33
IPv6 com IPsec	0,16	4,31	8,28	17,65	38,65
IPv6 com IPsec e TLS	0,17	5,77	12,47	22,69	54,71

Fonte: Produzido pelo autor do trabalho

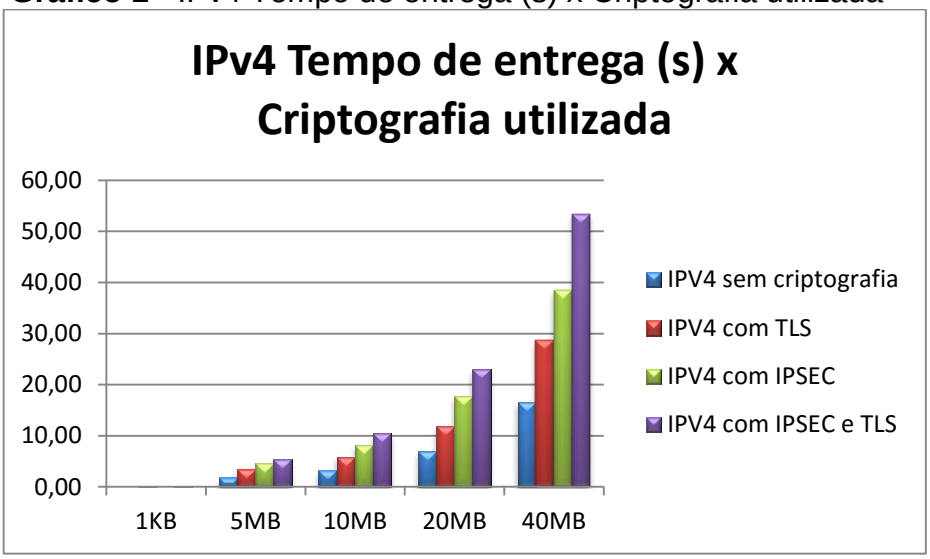
O uso do IPv6 em relação ao IPv4 levou um tempo um pouco maior para concluir a transferência, inclusive, a diferença ficou maior quando foram enviados e-

mails com 20M e 40M. Essa diferença pode representar a necessidade de aumento de recursos de rede em caso de redes com grande tráfego.

Em relação ao uso dos protocolos criptográficos, o tempo de resposta tem considerável aumento tanto no IPv4 quanto no IPv6, acentuando-se esta diferença quando do envio de e-mails transportando arquivos maiores.

O gráfico 1 mostra o desempenho com IPv4, apresentando o tempo gasto para entregar mensagens de diferentes tamanhos de acordo com o protocolo de criptografia utilizado. Podemos ver o aumento do tempo chegando a 54 segundos quando enviado um e-mail com um anexo de 40M utilizando juntos o IPsec e TLS.

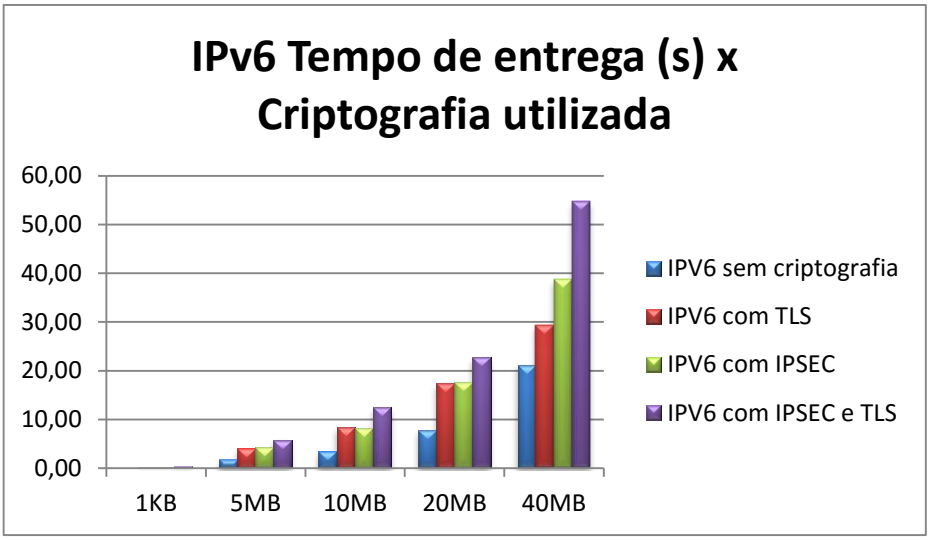
Gráfico 1 - IPv4 Tempo de entrega (s) x Criptografia utilizada



Fonte: Produzido pelo autor do trabalho

Semelhante ao anterior, o próximo gráfico mostra o desempenho obtido com IPv6. Neste podemos ver que o tempo chega a 55 segundos quando enviado um e-mail com um anexo de 40M utilizando juntos o IPsec e TLS.

Gráfico 2 - IPv6 Tempo de entrega (s) x Criptografia utilizada

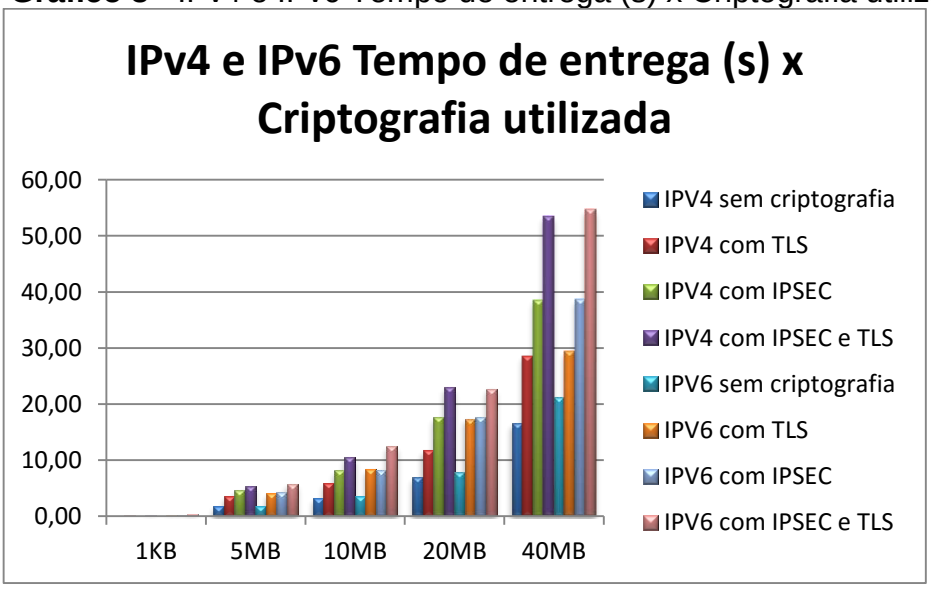


Fonte: Produzido pelo autor do trabalho

Encerramos apresentando um gráfico que compara IPv4 e IPv6 nas situações de testes propostas.

Fica ilustrado, permitindo reafirmar que, em nossos testes, nas diversas configurações de criptografia utilizadas, o IPv6 apresentou desempenho pouco pior que o IPv4, mostrando-se viável sua utilização com um alerta para possíveis perdas de performance para implementações em redes de grande porte.

Gráfico 3 - IPv4 e IPv6 Tempo de entrega (s) x Criptografia utilizada



Fonte: Produzido pelo autor do trabalho

CONCLUSÃO

O estudo permitiu demonstrar a utilização do serviço de envio de e-mail utilizando o protocolo SMTP com IPv6 e IPsec, demonstrando desde a configuração da rede para IPv6 e IPsec à configuração dos servidores de e-mail.

Verificamos que o uso do IPv6 em relação ao IPv4 apresenta um aumento pequeno no tempo de entrega das mensagens.

Com o uso do IPsec, temos um ganho claro na segurança das informações trafegadas na rede, pois podemos garantir a autenticidade e confidencialidade dos pacotes IP, ficando sigiloso até mesmo os tipos de dados contidos nos pacotes. Porém, para garantir a autenticidade e confidencialidade, é necessário acrescentar dois cabeçalhos extras aos pacotes.

O uso da criptografia tanto com TLS quanto com IPsec pode gerar um aumento de duas a três vezes no tempo de entrega em relação ao envio sem criptografia. Porém, o IPsec apresenta uma performance pouco pior em relação ao TLS em troca do ganho no nível de segurança. O uso do IPsec em conjunto com o TLS, como esperado, tem o maior custo devido à utilização de dois protocolos criptográficos.

Podemos afirmar que mais de 90% das mensagens SMTP estão delimitadas dentro de uma faixa de tamanho de até 5MB (CASTILHO et al., 2010). Assim, tomamos as mensagens de 5MB como referência para comparação do desempenho em relação ao tamanho das mensagens.

Tendo por base as mensagens de 5 MB vimos que esse valor foi em média 2 vezes maior nas mensagens com 10 MB, 4 vezes para 20MB e chegando a uma média de 9 vezes para mensagens de 40 MB.

Portanto, o uso do IPv6 não apresenta uma performance muito aquém do IPv4, mostrando-se que sua utilização é totalmente viável seu uso em servidores SMTP, com ressalva para possíveis perdas de performance nas implementações em redes de grande porte. O uso do IPsec proporciona maior segurança à comunicação, mas causa perda de performance que fica mais explícita nos casos de envio de arquivos grandes como os de 20 e 40 MB. Assim, ao implantar o IPsec na comunicação entre servidores SMTP, devem ser levados em consideração aspectos

como: o tamanho da rede, intensidade da utilização e tamanho dos arquivos trafegados.

Este trabalho poderá ser útil como referência para futuros trabalhos e implementações envolvendo a utilização do SMTP com IPv6 e IPsec, possivelmente IPSec no modo de túnel.

REFERÊNCIAS

ANTON, Eric Ricardo. Protocolo IPv6. 1999. Disponível em: http://www. gta. ufrj. br/grad/99_2/eric/index. htm#IPv6. Acesso em: 01 de setembro de 2015.

BASSO, Cristina. Implementação de IPsec integrado com o IPv6. 2012. Disponível em http://repositorio.roca.utfpr.edu.br/jspui/handle/1/198. Acesso em: 02 de outubro de 2015.

BRAGHETTO, Luis Fernando B. et al. IPSec Segurança de Redes – INF542. 2003. Disponível em: http:// www.braghetto.eti.br /files/ipsec%20-%20Versao%20Final.pdf. Acesso em: 05 de abril de 2016.

CASTILHO, Luis Henrique D. et al. Caracterização de tráfego SMTP na Rede de Origem. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2010.

CONCEIÇÃO, Vinícius V. et al. *Simple Mail Transfer Protocol*, uma implementação simplificada. 2014. Disponível em: http://www. facom. ufms. br/ gestor/ titan .php?target=openFile&fileId=1686. Acesso em: 01 de setembro de 2015.

FERNANDES, Eder Leão. Segurança na implementação do Protocolo TLS. 2014. Disponível em: http://www. dca. fee. unicamp. br/ ~marco/ cursos/ ia012_14_1/ trabalhos_finais/ tf_12_artigo. pdf. Acesso em: 15 de setembro de 2015.

FREITAS, H. et al. O correio eletrônico, a comunicação e a decisão: um paralelo Brasil-França. Porto Alegre: SBSI – Simpósio Brasileiro de Sistemas de Informação, PUCRS-UFRGS-UNISINOS-UCS. Outubro, 2004.

HAGEN, Silvia. *IPv6 Essentials. O'Reilly Media, Sebastopol-USA*, 2014.

KOZIEROK, Charles M. *The TCP/IP guide: a comprehensive, illustrated Internet protocols reference. No Starch Press*, 2005.

LOSHIN, Pete. *IPv6: Theory, protocol, and Practice. The Morgan Kaufmann Series in Networking*. 2.ed. São Francisco: Morgan Kaufmann, 2004.

REGHINI, Edivania Cardoso. Técnicas de tunelamento para redes hibridas-IP & IPv6. 2014. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/handle/1/1756. Acesso em: 15 de setembro de 2015.

RHOTON, John. Programmer's guide to internet mail: SMTP, POP, IMAP, and LDAP. Digital Press, 2000.

RIABOV, Vladimir V. SMTP (Simple Mail Transfer Protocol). Handbook of Computer Networks: LANs, MANs, WANs, the Internet, and Global, Cellular, and Wireless Networks, Volume 2, p. 388-406, 2005. Disponível em: http://onlinelibrary.wiley.com/doi/10.1002/9781118256114.ch26/summary. Acesso em: 01 de setembro de 2015.

ROTOLE, Erick Dantas; REZENDE, Pedro Antônio D. Arquitetura ip security. 2004. Disponível em http://www. cic. unb. br/docentes/pedro/trabs/IPsec. rtf. Acesso em: 20 de setembro de 2015.

RFC 1752. BRADNER, S.; MANKIN, A. The Recommendation for the IP Next Generation Protocol. Internet Engineering Task Force, 1995. Disponível em: https:// tools. ietf. org/ html/ rfc1752. Acesso em: 01 de setembro de 2015.

RFC 2460. DEERING, S.; HINDEN, R. Internet Protocol, version 6 (IPv6). Internet Engineering Task Force, RFC, 1998. Disponível em: https:// tools. ietf. org/ html/ rfc2460. Acesso em: 01 de setembro de 2015.

RFC 5321. KLENSIN, J. 2008. Simple Mail Transfer Protocol. Disponível em: http://tools.ietf.org/pdf/rfc5321.pdf. Acesso em: 01 de agosto de 2015.

RFC 5322. RESNICK, P. Internet Message Format. Internet Engineering Task Force, 2008. Disponível em http://tools. ietf. org/rfc/rfc5322. Acesso em: 20 de setembro de 2015.

RFC 6071. FRANKEL, S.; KRISHNAN, S. IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap. Internet Engineering Task Force, RFC, 2011. Disponível em http://tools. ietf. org/rfc/rfc6071. Acesso em: 20 de setembro de 2015.

SANTOS, Rodrigo R. et al. IPv6.br – Curso IPv6 Básico. 2015. Disponível em: http://www.ipv6.br/pagina/downloads. Acesso em: 05 de abril de 2016.

SCIMANDAR. Differences IPv4 Vs IPv6. 2009. Disponível em: http:// www. techsutram. com /2009/03/differences-IPv4-vs-IPv6.html. Acesso em: 15 de setembro de 2015.

SIDHPURWALA, Huzaifa. Transport Layer Security. 2013. Disponível em: https:// rhsecurity. wordpress. com /tag/tls/. Acesso em: 07 de abril de 2016.

SMETANA, George Marcel MA. IPv4 e IPv6. Laboratório de Arquitetura e Redes de Computadores, Escola Politécnica da Universidade de São Paulo (USP), 2012. Disponível em: http://www.abusar.org.br/ftp/pitanga/Redes/ArtigoIP.pdf. Acesso em: 15 de setembro de 2015.