CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Segurança da Informação

André Beraldi Chiosini

ANÁLISE COMPARATIVA ENTRE OS *FIREWALL IPTABLES E IPFW*

Americana, SP
2014

CENTRO PAULA SOUZA

FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Segurança da Informação

André Beraldi Chiosini

andre_chiosini@hotmail.com

ANÁLISE COMPARATIVA ENTRE OS *FIREWALL IPTABLES E IPFW*

Trabalho monográfico, desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Segurança da Informação da Fatec Americana, sob orientação da Prof. Me. Maria Cristina Luz Fraga Moreira Aranha.
Área de concentração: Segurança da Informação.

Americana, SP
2014

Chiosini, André Beraldi
C467a Análise comparativa entre os *firewall iptables* e IPFW. / André Beraldi Chiosini. – Americana: 2014.
69f.

Monografia (Graduação em Tecnologia em Segurança da Informação). - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza.
Orientador: Prof. Me. Maria Cristina da Luz Fraga Moreira Aranha

1. Segurança em sistemas de informação I. Aranha, Maria Cristina da Luz Fraga Moreira II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana.

CDU: 681.518.5

iii

André Beraldi Chiosini

ANÁLISE COMPARATIVA ENTRE OS *FIREWALL IPTABLES* E *IPFW*

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Segurança da Informação pelo CEETEPS/Faculdade de Tecnologia – FATEC/ Americana.
Área de concentração: Segurança da Informação

Americana, 02 de Dezembro de 2014.

Banca Examinadora:

Profa. Me. Maria Cristina Luz Fraga Moreira Aranha. (Presidente)
Fatec - Americana

Prof. Dr. Renato Kraide Soffner. (Membro)
Fatec - Americana

Prof. Me. Diogo Robles. (Membro)
Fatec - Americana

AGRADECIMENTOS

Inicialmente agradeço a minha família, a FATEC Americana, seus docentes, funcionários e todos que integram esta equipe, pois sem estes não existiria esta oportunidade. Agradeço a Deus por todas as oportunidades e por todas as pessoas que colocou em meu caminho, pois cada uma tem sua parcela em meu aprendizado e desenvolvimento.

Agradeço à orientadora Prof. Me. Maria Cristina Luz Fraga Moreira Aranha, pela paciência e dedicação no acompanhamento deste trabalho e a amizade dedicada. Agradeço ao amigo e Prof. Alexandre Garcia Aguado, pelos auxílios, conselhos e ensinamentos passados para tornar este trabalho possível.

Agradeço em especial minha Tia Juraci Beraldi, pois sem a sua grande parcela de auxílio este trabalho não teria sido concluído em tempo hábil.

DEDICATÓRIA

A Deus por tudo o que me proporciona na vida.

À minha esposa e filha amada que abriram mão de momento e horários de lazer e convivência para que este trabalho pudesse ser realizado e concluído.

Aos amigos e familiares que deram todo o apoio quando mais precisei.

RESUMO

Este trabalho conceitua, de forma detalhada, o que é um *firewall* e apresenta a comparação entre duas ferramentas de *firewall* para software livre, a saber:*iptables* e *ipfw*. Esta comparação tem por objetivo verificar se entre as duas versões de *firewall* há dados suficientes para alguma conclusão quanto à eficácia de uma delas em relação à outra, ou seja, pretende-se verificar se as ferramentas têm desempenhos similares ou se o desempenho de uma delas é melhor do que o desempenho da outra. Para a compreensão desta comparação, é feita uma explanação sobre o funcionamento de rede de computadores e seus protocolos. Também se apresenta a conceituação sobre as ameaças a estas redes, para que assim seja possível explanar sobre as questões que envolvem a segurança referente às redes de computadores. A realização da comparação entre as ferramentas será feita através de configurações em cada uma delas. Em seguida serão feitos testes em um ambiente controlado, analisando-se e comparando-se os resultados obtidos com cada uma delas. Após a análise e comparação dos dados coletados, é feita uma discussão sobre os resultados obtidos. Estes resultados mostrarão qual a conclusão obtida através do estudo de caso. Vale lembrar que o escopo deste trabalho diz respeito ao desempenho de dois tipos de *firewall* usados atualmente, porém medidas de segurança indicam que só o uso de *firewall* não deve ser o único método de proteção.

Palavras-chave: *Firewall*; *Software* Livre; Segurança da Informação;*Iptables; Ipfw.*

ABSTRACT

The main purpose of this paper is to give to concept, in detail what is firewall present and compare between two tools of firewall to a free software, known as iptables and ipfw. This basis of comparison has the goal to ascertain if both versions of firewall there is enough data to some conclusion about the efficiency of each one in relation to other, in other words, it is thus to verify if these tools has similar development or if the performance of each one is better than the other. To understand this comparison, it is made an explanation about the operation in the computer network and its protocols. Also shows the concepts about the threats to theses network, for this to be possible to outline about this questions which involve the safety relating to the computer network. The execution about this comparison between the tools will be done through tests and configuration in each one, and then it will be made tests in a controlled environment, where is possible to analyze and compare the results obtained in each of them. After acquire analyze and compare the data collect, an open discussion is raised about the results obtained. These results will show what is the conclusion which were gained through this results. It worth point out that the scope used in this work concerns to development about these two kinds of firewall which is currently used, however safety measures point to the only use of firewall must not be the only method of protection.

Keywords: *Firewall; Free Software; Information Security;Iptables; Ipfw.*

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

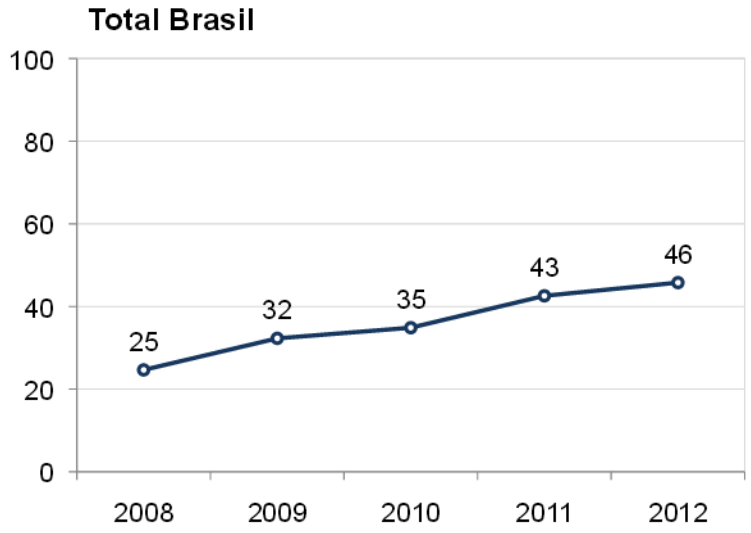
ARPANET	*Advanced Research Projects Agency Network*
CAN	*Campus Area Network*
CERT	Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil
CETIC	Centro de Estudos sobre as Tecnologias da Informação e da Comunicação
DOD	*Department of Defense*
FTP	*File Transfer Protocol*
GAN	*Global Area Network*
HTTP	*Hypertext Transfer Protocol*
HTTPS	*HyperText Transfer Protocol Secure*
IP	Internet *Protocol*
ISO	*International Organization for Standardization*
LAN	*Local Area Network*
MAN	*Metropolitan Area Network*
OSI	*Open Systems Interconnection*
PAN	*Personal Area Network*
SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Protocol*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
WAN	*Wide Area Network*
WWW	*World Wide Web*

SUMÁRIO

1. INTRODUÇÃO

É fato que cada vez mais as pessoas estejam armazenando, acessando ou manipulando as informações em formato digital. Isto se deve ao aumento do número de computadores e meios digitais utilizados, conforme é possível verificar em pesquisa realizada pelo CETIC, em 2013, na qual se pode observar o aumento da proporção de domicílios com computador, como mostra a Figura 1.

Figura 1 – Proporção de domicílios com computador em 2012.



Fonte: CETIC (2012)

Quando se torna necessário, em algum momento, acessar alguma dessas informações, estejam elas localizadas em um computador pessoal, em um servidor dentro de uma organização, em nuvem ou em qualquer outro meio de armazenamento e não sendo uma informação pública, realiza-se tanto a solicitação do acesso quanto a autorização a ela. Porém, da mesma forma que ocorre um maior número de manipulações dessas informações em meio digital, também ocorre um maior número de tentativas de acesso não autorizados, tornando necessária a criação de meios de proteção e controle dos acessos. Tentativas de manipulação não autorizadas a estas informações podem ser ocasionadas por diversos meios e fatores. Neste caso, há a necessidade de proteger tais informações das possíveis ameaças, sejam elas realizadas por *cracker*[1], vazamentos indevidos para empresas

[1] *Cracker* – Indivíduos que usam seus conhecimentos para invadir sites e computadores com objetivos ilícitos, como vandalismo ou roubo, chamado também de "*Hacker* do mal" ou "*Hacker* sem ética". (ULBRICH, 2011).

concorrentes ou quaisquer tipos de falhas. Fontes (2008) afirma que, dependendo do tipo da organização, a indisponibilidade da informação pode levar a organização a ter um impacto financeiro ou em sua imagem que pode fazer com que esta empresa não mais consiga manter suas atividades no mercado. Como a indisponibilidade da informação, para determinadas organizações, pode ser um fator crítico, nesses casos, quando realizada uma avaliação de riscos, o fator de proteção das informações torna-se um dos pontos com maior relevância e maior preocupação.

A informação, muitas vezes, é o bem mais precioso de uma empresa, podendo a mesma perder seus bens materiais e, mesmo assim, dar continuidade à realização dos seus trabalhos e obtenção de lucros. Porém, caso ocorra o vazamento ou perda de informações, certamente ocorrerão prejuízos imensuráveis e irrecuperáveis. Mitnick (2003, p.3) cita que “o fator humano é o elo mais fraco da segurança”. Considerando, então, que o elo mais fraco são as pessoas, é importante prover treinamentos, orientações e políticas em toda a organização. Para aquelas empresas que não possuem equipamentos que auxiliem nas ações de proteção, igualmente torna-se relevante alertar os responsáveis sobre a importância de se ter disponíveis os meios técnicos para cada situação que se apresente. Com base nessa fragilidade no recurso humano, se faz prudente a implementação de equipamentos de controle e monitoramento de rede como, por exemplo, o *firewall.* De acordo com Kurose (2010, p.535-536):

> Um *firewall* é uma combinação de hardware e software que isola a rede interna de uma organização da Internet em geral, permitindo que alguns pacotes passem bloqueando outros. Um *firewall* permite que um administrador de rede controle o acesso entre o mundo externo e os recursos da rede que administra, gerenciando o tráfego de e para esses recursos.

É neste contexto que se insere o problema apresentado neste trabalho, ou seja, escolher o *firewall* mais adequado para uma empresa. Portanto a pergunta que se faz é: Como decidir qual é o melhor *firewall,* entre os dois mais usados, *iptables* e *ipfw?* Sendo o *firewall* configurado e administrado por pessoas, vale lembrar o que foi mencionado por Mitnick (2003), que ao configurar um *firewall* deve-se sempre

planejar cada detalhe e cada configuração a ser realizada, com o intuito de prover a maior segurança e confiabilidade possíveis.

Os detalhes de configuração de um *firewall* são escolha particular de cada administrador, que é responsável por pensar cada possibilidade de ameaças, levando em consideração as ferramentas utilizadas, as portas padrões e os protocolos a serem bloqueados. Cabe ressaltar que o responsável pela configuração do *firewall* é também uma pessoa e que ao realizar o procedimento para a configuração deve evitar um padrão para as regras de configuração do *firewall*, utilizando para cada situação a configuração que julgar e compreender como a mais confiável e segura, dentro de suas necessidades e políticas. Com base nesses aspectos citados, este trabalho tem por objetivo geral identificar o *firewall* mais adequado entre dois deles: *iptables* e *ipfw.* Os objetivos específicos são:

- Fazer um estudo comparativo entre dois *firewalls* disponíveis no mercado: *iptables e ipfw,* considerando os aspectos técnicos e de usabilidade.

- Realizar um estudo de caso entre as duas ferramentas para determinar a eficiência de cada uma delas.

- Determinar as diferenças, vantagens e desvantagens de dois *firewalls* disponíveis no mercado: *iptables e ipfw*.

- Analisar os resultados obtidos.

- Discutir estes resultados para se chegar a alguma conclusão.

Há diversas hipóteses a se considerar e, entre elas:

a) As duas ferramentas têm a mesma eficiência.

b) Uma delas é mais eficiente que a outra.

c) A configuração de cada uma das ferramentas não é tão simples, exigindo treinamento e conhecimento do administrador da rede.

d) A comparação entre as ferramentas pode ser inconclusiva ou parcialmente inconclusiva.

A justificativa para a escolha do problema está relacionada à área de atuação do autor deste trabalho e à importância da escolha de um *firewall* adequado a uma organização. O método utilizado neste trabalho é com pesquisa aplicada, usando estudo de caso, pesquisa bibliográfica e documental, coletando dados de forma qualitativa.

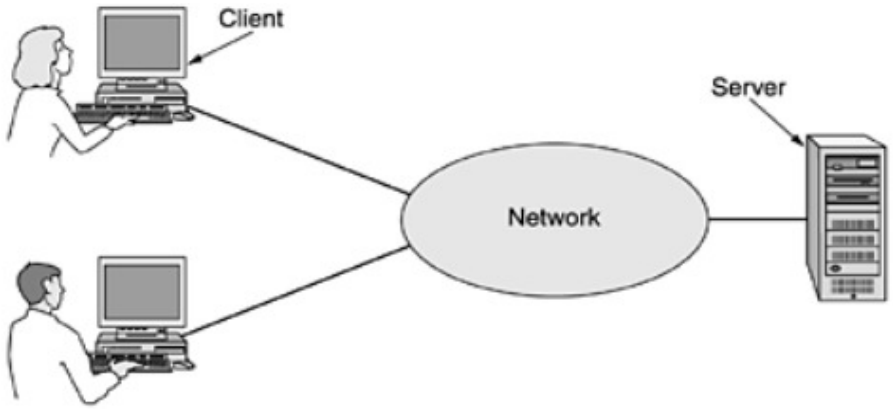
A organização deste trabalho é a seguinte: no próximo capítulo apresenta-se a pesquisa bibliográfica feita sobre redes de computadores. No Capítulo 3 apresentam-se os conceitos sobre Segurança da Informação, ameaças a redes e sobre *firewall*. No Capítulo 4 é feito o estudo comparativo entre as ferramentas escolhidas: *iptables* e *ipfw.* No Capítulo 5 apresentam-se as conclusões deste trabalho.

2. REDES DE COMPUTADORES

Rede de computador é um conjunto de computadores autônomos, interconectados por uma única tecnologia. Uma rede pode ser composta por dois computadores ou mais, interconectados, trocando informações entre si. Esta conexão física pode ser feita através de diversos meios, como fios de cobre, fibras ópticas, micro-ondas, ondas de infravermelho e, até mesmo, por satélites de comunicações (TANENBAUM, 2003).

A Internet é uma grande rede de computadores que interconecta milhares de dispositivos ao redor do mundo. Há pouco tempo, esses dispositivos eram basicamente computadores de mesa, estações de trabalhos ou os chamados servidores que armazenam e transmitem as informações. Atualmente verifica-se a existência de diferentes dispositivos como tvs, *laptops*, consoles para jogos, telefones celulares e automóveis que estão sendo conectados à Internet. A todos esses dispositivos, conectados, dá-se o nome de hospedeiros. (KUROSE, 2010). A Internet nos dias de hoje, segundo Comer (2006), tornou-se parte fundamental da vida cotidiana, uma vez que contém diversas informações, onde ocorrem trocas de correspondências, dados e arquivos de interesse comum. Mas, é importante entender que a Internet não é um novo tipo de rede física, e sim um método de interconexão de diversas redes físicas. A Figura 2 apresenta um exemplo de como é a rede de computadores entre dois clientes e o servidor.

Figura 2 – Exemplo de redes de computadores entre dois clientes e servidor.



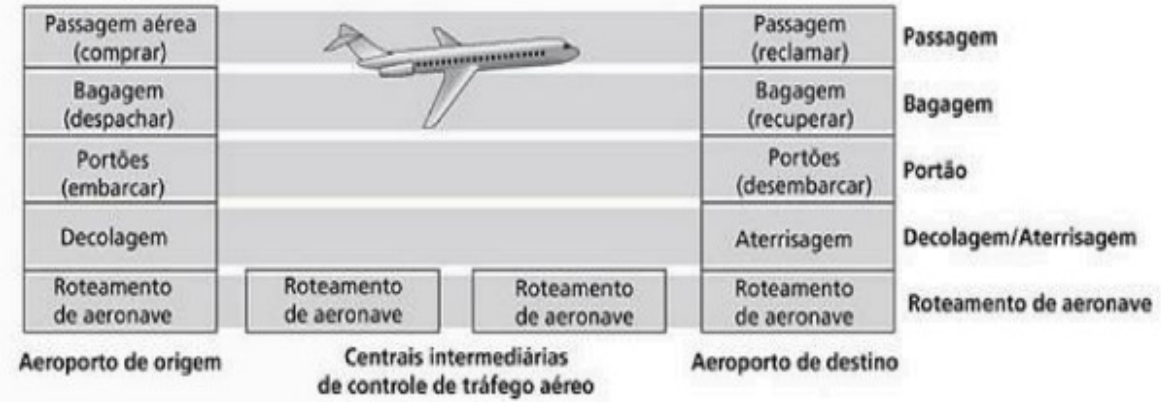
Fonte: Tanembaum (2003, p.20)

A rede pública mais importante e que está disponível para as organizações é a Internet, pois realiza interligações entre outras redes de computadores alocadas em partes distintas do mundo. Stallings (2005) ressalta que, além disso, a Internet não pertence e não é gerenciada por uma única entidade. A Internet e qualquer outra intranet privada consistem em várias redes separadas que são interconectadas por roteadores, cujos dados são transmitidos da origem, através de pacotes, para o seu destino, percorrendo um caminho que envolve outras redes e roteadores. Organizações, residências e qualquer meio que utilizem as redes de computadores podem ser catalogadas por tipos de redes. Edwards (2009) afirma que na atualidade existem diversos tipos de redes, cada uma delas utilizada conforme a necessidade e a arquitetura do local a ser implementada.

2.1 Modelos de Redes de Computadores

Ao definir modelos de redes de computadores, Kurose (2010) apresenta um modelo dividido em camadas e se utiliza de uma analogia entre o sistema de uma companhia aérea e redes de computadores, conforme Figura 3, para que se possa obter um melhor entendimento. O sistema de uma companhia aérea é iniciado no momento em que são adquiridas as passagens para o voo; tendo as passagens e o destino escolhido, despacha-se a bagagem. Em seguida embarca-se através do portão previamente determinado; após o embarque dos passageiros tem início a decolagem da aeronave e todo o procedimento de voo, também chamado de roteamento da aeronave. Chegando ao seu destino inicia-se o processo de aterrissagem da aeronave, para o desembarque dos passageiros no portão predestinado para aquele voo específico. Após o desembarque, a bagagem despachada fica disponível a quem de direito e, caso necessário, retorna-se ao processo de passagens para a reclamação de alguma ocorrência. (KUROSE, 2010).

Figura 3 – Modelo de um sistema de uma companhia aérea dividido em camadas.



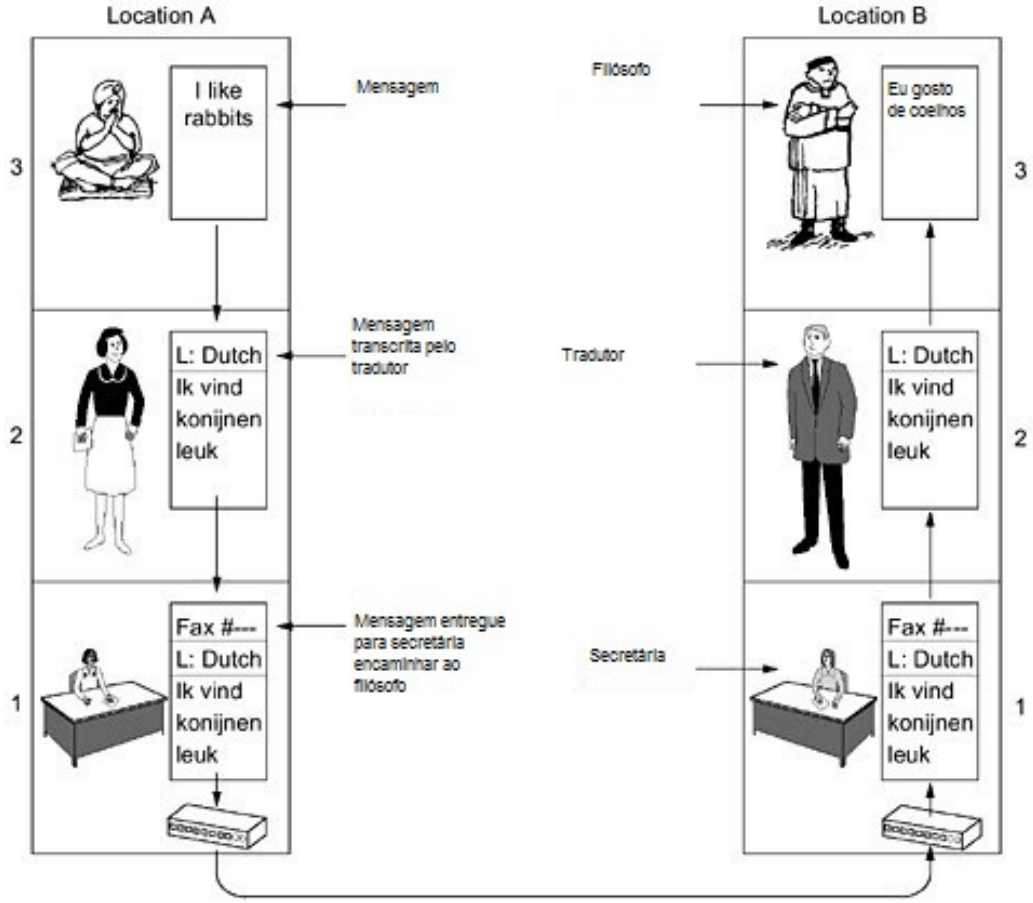
Fonte: Kurose (2010, p.36)

Por meio desta analogia Kurose (2010) explica a relação entre um modelo de sistema de companhia aérea e as redes de computadores, de forma que cada camada de origem se relacione com a camada de destino. Cada camada, combinada às camadas abaixo dela, implementará e dará continuidade à execução de um serviço.

A continuidade do serviço realizado entre as camadas, segundo Kurose (2010) tem, como o exemplo já mencionado, todo o procedimento relativo ao voo da aeronave, desde o embarque até o desembarque. Pode-se comparar este modelo de um sistema de voo ao despacho de um pacote encaminhado pelo *host* de origem e o seu encaminhamento através das portas utilizadas pelos protocolos até o momento em que as informações são recuperadas e o envio possa ser concluído no *host* de destino.

Para que dois dispositivos diferentes, com interfaces e aplicações distintas possam se comunicar e ter a total compreensão do que foi dito pelo remetente faz-se necessária a utilização de uma linguagem padrão. Ao utilizar um modelo de camadas similar ao funcionamento da rede de computadores é possível compreender melhor o funcionamento das redes, camadas e protocolos. A Figura 4 mostra a analogia de uma comunicação entre dois filósofos que não falam a mesma língua (TANENBAUM, 2003).

Figura 4 – Analogia da utilização de modelos de redes de computadores em camadas.



Fonte: Modificado de Tanenbaum (2003, p.39)

É possível usar esta analogia para explicar e demonstrar a comunicação entre duas redes distintas. Tanenbaum (2003) utiliza este modelo para demonstrar o encapsulamento dos pacotes no *host* de origem até o momento da interpretação pelo *host* de destino, sendo cada etapa uma camada. O entendimento da analogia apresentada na Figura 4 por meio da seguinte explicação. Dois filósofos um Inglês e um Brasileiro, que não falam a mesma língua e possuem cada um a sua secretária desejam se comunicar e, para isso, contratam um tradutor. O primeiro filósofo escreve a mensagem a ser enviada em inglês "*I like rabbits*" e encaminha ao seu tradutor. Até este momento é possível verificar que o filósofo e o tradutor se comunicam na mesma língua e se compreendem perfeitamente. O tradutor do primeiro filósofo realizará a tradução do texto em inglês para uma linguagem neutra que o outro filósofo compreenda que, neste caso, é o Holandês. Nesta parte é utilizada a linguagem compreendida pelos dois tradutores que repassarão a informação, de forma compreensível, ao filósofo de destino como o protocolo escolhido para a comunicação. É possível verificar por esse exemplo que os

filósofos são os *host*s que irão se comunicar e o texto como o pacote encapsulado. O tradutor do filósofo remetente encaminha para a secretária que utilizará o *fax* para encaminhar à secretária do filósofo destinatário e esta entregará ao filósofo o texto em Holandês. Ao receber a mensagem e compreender o seu conteúdo no idioma neutro, este último conseguirá reportar ao primeiro filósofo em uma linguagem que seja compreensível ao tradutor e ao filósofo, no caso o idioma em português. Portanto, através desta analogia de camadas e protocolos é possível verificar que a mensagem recebida pelo filósofo de destino será “Eu gosto de coelhos”, que será compreendida e assimilada pelo destinatário demonstrando que as informações estavam íntegras e compreensivas, da mesma forma em que é realizado o funcionamento de uma rede em camadas e o encapsulamento dos pacotes.

2.1.1 Modelo OSI

O modelo de referência OSI foi desenvolvido pela ISO como um modelo para arquitetura de protocolos e como estrutura básica para o desenvolvimento de padrões de protocolo. Este modelo teve suas funções de comunicação particionadas em uma hierarquia de camadas. Stallings (2005) afirma que o modelo definido como OSI, geralmente é composto por sete camadas (Aplicação, Apresentação, Sessão, Transporte, Rede, Enlace de dados, Física) conforme mostra o Quadro 1. Cada camada realiza funções necessárias para que a próxima camada possa continuar a realizar as funções, decompondo, assim, um problema em uma série de problemas mais administráveis. Sendo necessário o envio de uma mensagem entre duas entidades a comunicação se dará por meio da invocação da camada de aplicação (camada 7) onde será estabelecido o relacionamento com a camada 7 do destino usando o protocolo de aplicação. O protocolo de aplicação exigirá os serviços da camada 6 e assim por diante até a camada física (Camada 1) onde realmente serão transmitidos os *Bits* para o destino.

Quadro 1 – As camadas OSI

CAMADA	NOME
7	Aplicação
6	Apresentação
5	Sessão
4	Transporte
3	Rede
2	Enlace
1	Física

Fonte: Elaborado pelo Autor

A camada de número 1, denominada Física, lida com características mecânicas, elétricas, funcionais e de procedimento para acessar o meio físico; é nesta camada que se inicia o processo da transmissão e recepção do fluxo de *Bits* não estruturado. Após a camada física ter formatado os dados de forma que a camada de enlace de dados – também chamada de camada 2 – consiga interpretá-los, portanto, após o recebimento dos *Bits,* ela os converte de maneira inteligível transformando em unidades de dados e encaminhando para a camada de rede. A camada de rede (3) é a responsável pelo tráfego no processo cabendo a ela estabelecer, manter e terminar as conexões; é esta camada que decidirá qual o melhor caminho para os dados seguirem, assim como estabelecerá as rotas encaminhando os dados para a camada de transporte. A camada de transporte (4) tem a responsabilidade pela qualidade e confiabilidade na entrega e no recebimento dos dados, permitindo o controle de fluxo de ponta a ponta. A camada de sessão (5) é responsável por fornecer a estrutura de controle para a comunicação entre as aplicações, estabelecendo a sessão de conexão entre as aplicações e gerenciando para garantir que a conexão se mantenha ativa e possa ser encerrada quando necessário. A camada de apresentação (6) cuidará da formatação dos dados e da representação destes, oferecendo independência aos processos da aplicação, com relação as diferenças na representação dos dados para que o ambiente de aplicação consiga interpretar. A camada de aplicação (7) é a responsável por proporcionar acesso ao ambiente para os usuários; esta camada é a mais visível aos usuários, visto que ocorre a interação direta com ela por meio de softwares.

2.1.2 Modelo TCP/IP

O modelo TCP/IP é resultante dos estudos realizados na rede de pesquisas e desenvolvimentos denominada ARPANET, patrocinada pelo departamento de defesa dos Estados Unidos DOD. Tanenbaum (2003) explica que a sua implementação foi gradativa junto às universidades e repartições públicas, que foram conectadas pelo uso de linhas telefônicas dedicadas. A implantação gradativa da ARPANET permitiu que fosse possível verificar a sua fragilidade e assim motivasse a criação de um projeto inicial de pesquisa, com o objetivo de conectar várias redes de maneira uniforme. O projeto, desenvolvido para resolver os problemas da ARPANET através de uma nova arquitetura, mais tarde foi denominado de Modelo de referência TCP/IP. O modelo TCP/IP não possui hoje um padrão obrigatório a ser seguido, podendo se apresentado com 4 camadas ou com 5 camadas. Stallings (2005) define que, entretanto, com base nos padrões dos protocolos existentes e desenvolvidos, o TCP/IP deve ser dividido em 5 camadas conforme apresenta o Quadro 2. No modelo de 5 camadas Stallings (2005) afirma que a comunicação inicia-se pela camada de aplicação (camada 5) que solicitará os serviços da camada inferior, no caso a camada de transporte (camada 4), mantendo o fluxo até que se destine à camada física (camada 1), que realizará a comunicação direta com o *host*.

Quadro 2 – As camadas TCP/IP

Camada	Nome
5	Aplicação
4	Transporte
3	Inter-rede
2	Acesso à rede
1	Física

Fonte: Elaborado pelo Autor

A primeira camada, chamada de física, abrange a interface física entre um dispositivo de transmissão de dados e um meio de transmissão ou rede. Esta camada trata das características dos meios de transmissão, da natureza dos sinais, da taxa de dados e de questões relacionadas à comunicação, enquanto a segunda camada, chamada de camada de acesso à rede, trata do acesso e do roteamento de dados por uma rede para dois sistemas finais conectados à mesma rede. Caso a

comunicação se dê entre dois dispositivos conectados em redes diferentes será necessária a realização de procedimentos pela camada de inter-rede. A camada de inter-rede tem sua função em roteamentos para redes distintas; este protocolo tem uso não somente em sistemas finais, mas, também, em roteadores. Para a comunicação entre duas redes, comumente existe o requisito da confiabilidade visando garantir que todas as informações chegarão ao seu destino na mesma ordem em que serão enviadas. Assim, faz-se uso de uma camada comum, chamada de camada de transporte, que é compartilhada por todas as aplicações.

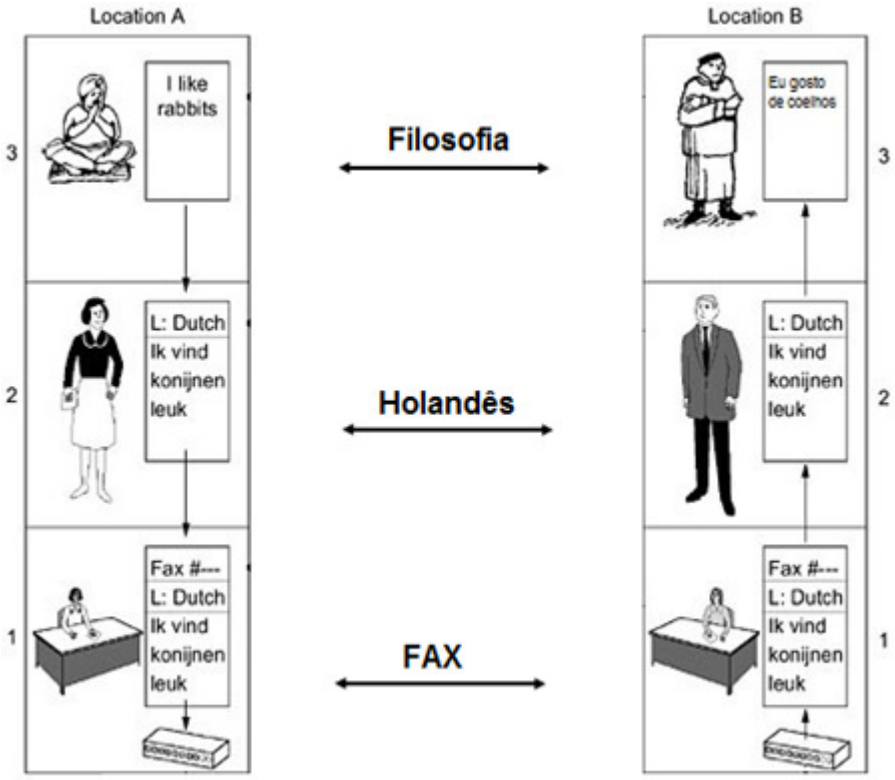
A camada de transporte visa permitir que as entidades pares dos *host*s de origem e de destino mantenham uma conversação; nesta é realizada a divisão entre dois protocolos: o protocolo TCP e o UDP. O protocolo TCP permite a entrega, sem erros e de forma fragmentada, de um fluxo de *bytes* originário de uma determinada máquina, conectada em qualquer lugar da inter-rede. Já o protocolo UDP não garante a entrega, a preservação de sequência ou a proteção contra a duplicação das informações. Algumas aplicações orientadas à transação utilizam o UDP como, por exemplo, o protocolo padrão de gerenciamento de rede SNMP. A camada de aplicação contém todos os protocolos de nível mais alto (TELNET, SMTP, FTP, entre outros) e, para cada tipo de aplicação diferente é utilizado um protocolo diferente; é nesta camada que ocorre uma maior interação com o usuário, visto que softwares trabalham diretamente sobre esta camada.

2.2 Protocolos de Rede de Computadores

O protocolo de rede tem como função estabelecer as regras de comunicação entre dois sistemas finais. Esse conjunto de regras permite que a comunicação entre sistemas distintos e com interfaces distintas consigam se comunicar sem a ocorrência de problemas de interpretação. Cada protocolo originado de uma rede pertence a uma camada especifica e é utilizado em conjunto com protocolos de outras camadas, onde são agrupados em pilhas de protocolos. A segmentação dos pacotes é realizada pelos protocolos garantindo que dentro destes pacotes estejam contidas todas as informações necessárias para que o *host* de destino ao receber o pacote possa compreender seu conteúdo por completo (KUROSE, 2010).

Aproveitando o exemplo da comunicação entre os filósofos apresentado anteriormente, a Figura 5 mostra agora os protocolos de comunicação entre eles.

Figura 5 – Protocolos da comunicação entre Filósofos.

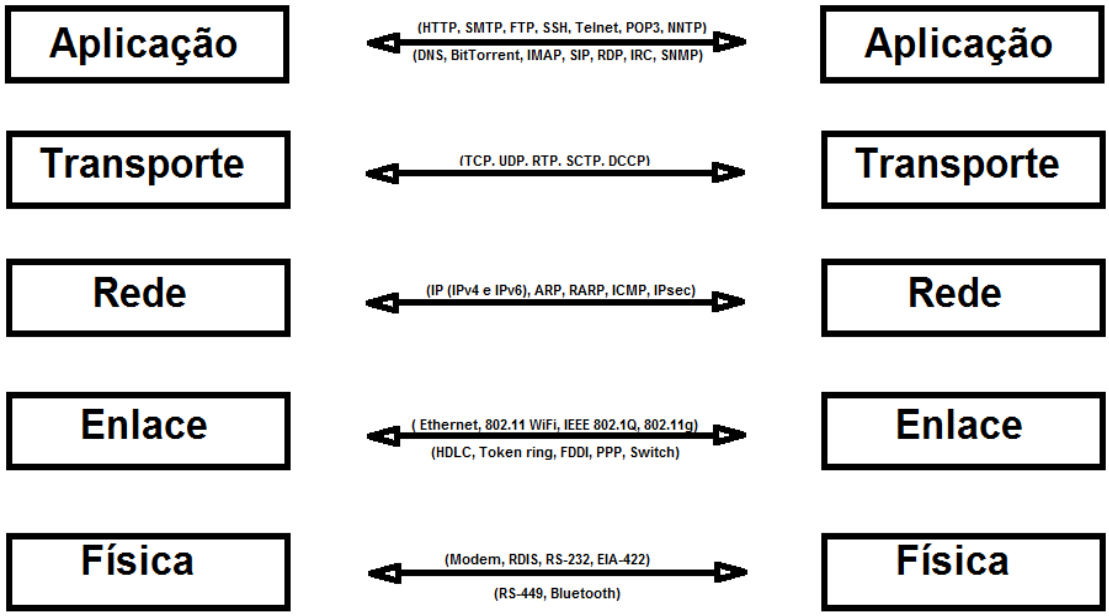


Fonte: Modificado de Tanenbaum (2003).

A Figura 5 ilustra um modelo de comunicação dividido em 3 camadas e seus respectivos protocolos; nesta analogia é possível verificar que cada camada possui um protocolo específico para sua comunicação. A primeira camada é constituída pelos filósofos. Os filósofos, talvez sem saber, “falam” o mesmo protocolo, afinal, ambos conhecem e devem conhecer a filosofia que é o protocolo utilizado nesta camada. A segunda camada é o momento em que os tradutores receberam a mensagem dos filósofos e estes a traduzirão para uma linguagem padrão, no caso o Holandês. O protocolo desta camada é o Holandês, pois é este o idioma de conhecimento dos dois tradutores. Ao receber a mensagem já escrita pelo filósofo do remetente e devidamente traduzida para o Holandês, a secretária do remetente encaminhará para a secretária do destinatário através do *fax.* O conhecimento da utilização do aparelho é obrigatório para que se possa enviar e receber a mensagem, que é o protocolo utilizado.

Como visto na analogia, cada camada do destinatário possui um protocolo responsável para a comunicação com a mesma camada do remetente. Com esta conclusão torna-se possível obter uma melhor compreensão da função dos protocolos e a comunicação junto às camadas, verificando que cada camada utiliza um protocolo específico. Da mesma forma que no exemplo de comunicação entre os filósofos, as redes de computadores também possuem para cada camada, protocolos específicos que são compreensíveis somente no âmbito daquela camada. A Figura 6 apresenta as camadas e seus respectivos protocolos.

Figura 6 – Protocolos e suas respectivas camadas



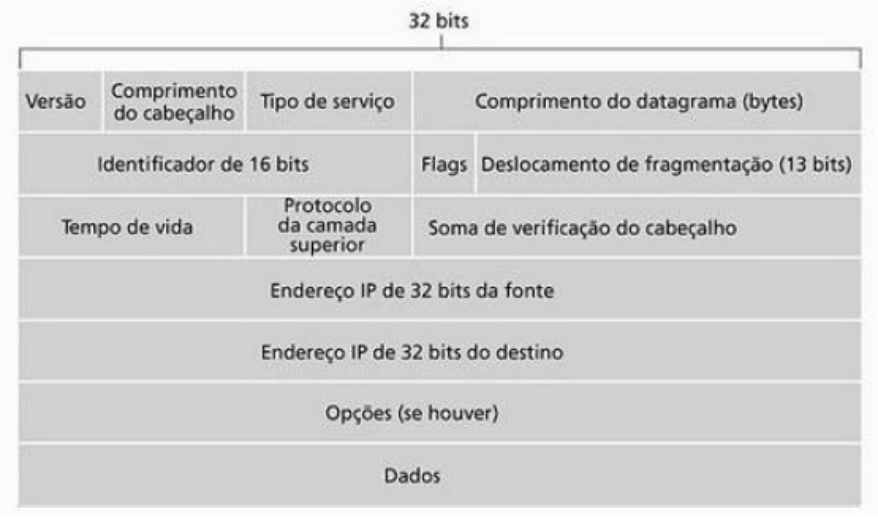
Fonte: Elaborada pelo Autor

Os protocolos de rede relacionados a este trabalho são os seguintes:

- Protocolo IP: Para que seja realizada a comunicação entre diferentes redes é preciso utilizar um protocolo responsável pelas funções de roteamento dos pacotes. Stallings (2005) afirma que este protocolo está localizado na camada de rede e é chamado de protocolo IP, existente tanto para os modelos de redes TCP/IP quanto para os modelos OSI.

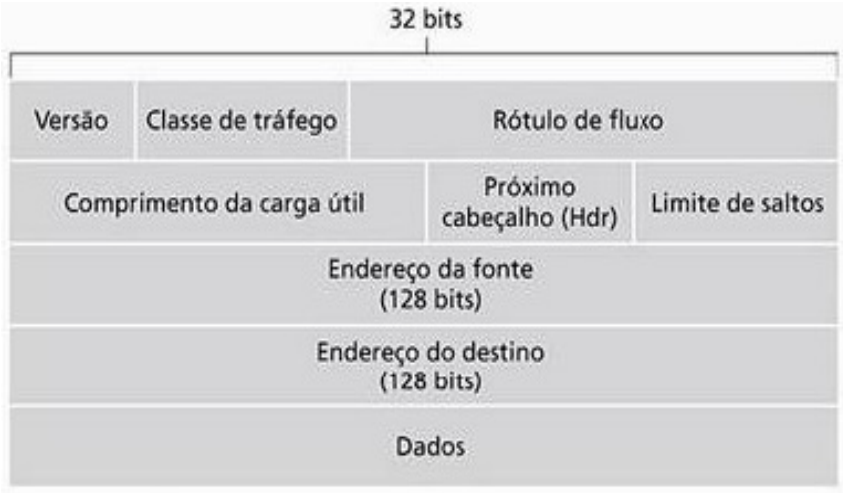
Com o grande crescimento das redes de computadores e seus *host*s notou-se que o protocolo IP possui limites quanto à quantidade de endereços que pode fornecer. Devido a esta limitação do IP e da grande quantidade de redes que utilizam os endereços existentes atualmente, foi elaborado um estudo referente a uma nova versão deste protocolo, versão esta que permite uma maior quantidade de endereços IP. Esta nova versão foi chamada de IPv6 e entre as duas versões existentes, hoje a utilizada em maior escala ainda é a antiga, chamada de IPv4; cada uma dessas versões possui um padrão diferente de datagrama (KUROSE, 2010). O datagrama IP é definido por Kurose (2010) como aquele que dispõe de um serviço não confiável, no qual o pacote pode ser recebido de forma desordenada, duplicada ou até mesmo não ser recebido por inteiro. Este protocolo é o responsável por todas as comunicações entre redes diferentes e é comumente encontrado em todas as redes residenciais, empresariais ou na Internet, possuindo um formato de endereçamento utilizado para a identificação de redes e computadores, denominado endereço IP, possuindo 32 *Bits* em sua versão IPv4 e 128 *Bits* na versão IPv6. A Figura 7 apresenta um datagrama de IPv4. Já a Figura 8 apresenta um datagrama de IPv6.

Figura 7 – Datagrama IPv4.



Fonte: (KUROSE, 2010, p.248)

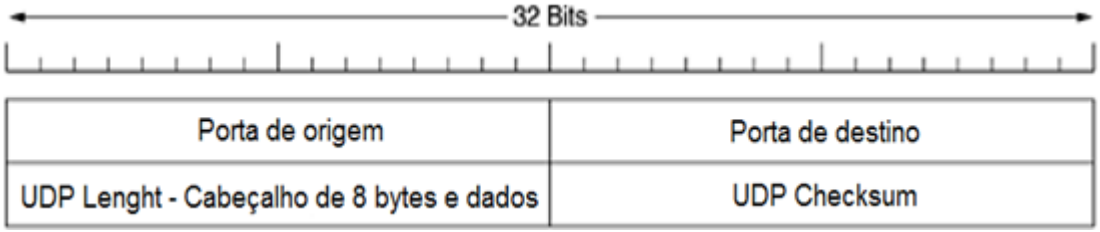
Figura 8 – Datagrama IPv6.



Fonte: (KUROSE, 2010, p.265)

- Protocolo UDP: Sendo o UDP e o TCP os dois protocolos da camada de transporte, Kurose (2010) ressalta que estes possuem a responsabilidade fundamental de ampliar o serviço de entrega IP entre dois sistemas finais, para um serviço de entrega processo a processo, e que esta ampliação é denominada de multiplexação e demultiplexação. A multiplexação é a função realizada para que se obtenha o conjunto com os dados e cabeçalhos no *host* de origem, para a criação dos segmentos e encaminhamento da camada de rede. A demultiplexação é a função realizada no receptor, no qual a camada de transporte examinará o segmento recebido da camada de rede, para identificar para qual porta correta deverá entregar os dados contidos no segmento. O UDP pode ser definido como, basicamente, o IP com um pequeno cabeçalho, como afirma Tanenbaum (2003). É um protocolo simples da camada de transporte, que permite à aplicação encaminhar o datagrama da origem, porém sem a garantia de que o pacote chegará ao *host* de destino e, igualmente, sem a garantia que este chegará na mesma ordem sequencial que foi enviado. A transmissão de segmentos do UDP consiste em cabeçalhos de 8 *bytes*, seguido pela carga útil; no cabeçalho UDP, como mostra a Figura 9, é possível visualizar a inclusão da porta de destino e de origem.

Figura 9 – Datagrama do UDP



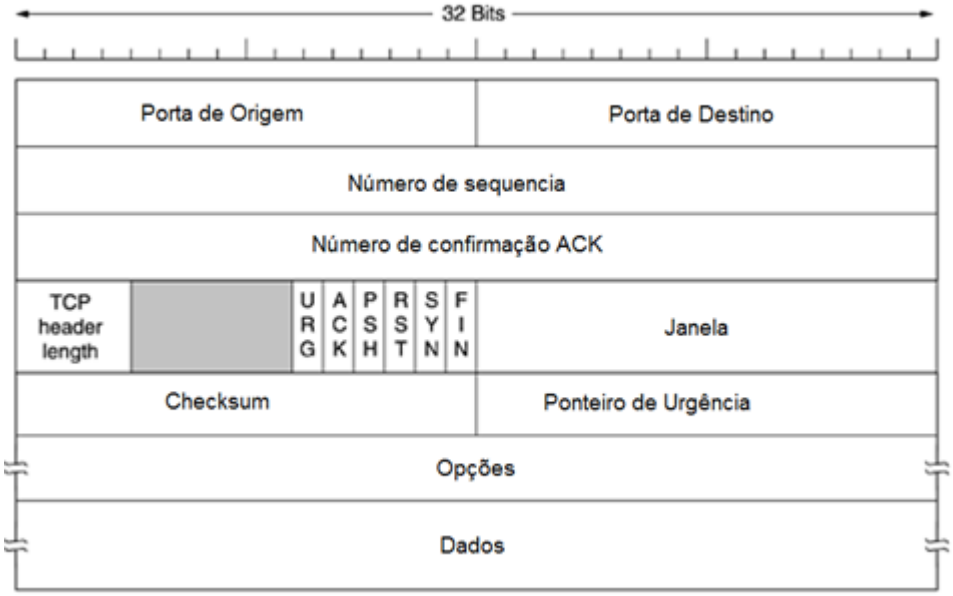
Fonte: Modificado de Tanenbaum (2003).

Sem os campos responsáveis pelas portas no datagrama, a camada de transporte não saberia o que fazer com o pacote recebido. Isso é devido às portas de origem e destino que a camada consegue entregar os segmentos corretamente, segundo Tanenbaum (2003). A grande importância do registro da porta de origem é para a situação em que uma resposta deve ser devolvida à origem, e a porta de destino possui a função de especificar qual processo deve receber o segmento. O protocolo UDP não realiza o controle de fluxo, de erros ou retransmissão após o recebimento de um segmento incorreto, ou seja, ele apenas fornece uma interface para o protocolo IP com o adicional de demultiplexação. (TANENBAUM, 2003).

- Protocolo TCP: Uma das funções mais importantes do TCP é prover a transferência confiável de dados. De acordo com Kurose (2010), o TCP consegue prover esta confiabilidade através dos controles de fluxo, números de sequência, reconhecimentos e temporizadores, podendo garantir que os dados sejam entregues ao destinatário na mesma ordem em que foram enviados e que estes estarão corretos provendo, também, o controle de congestionamento e evitando computadores com grandes quantidades de tráfego. Quando ocorrer uma falha no recebimento do datagrama, cabe ao TCP administrar os *timers* e retransmití-los quando necessário, podendo, até mesmo, reorganizá-los quando os datagramas forem recebidos fora de ordem. O serviço TCP, segundo Tanenbaum (2003), é obtido quando tanto a origem e o destino criam os chamados soquetes, sendo que um soquete possui um número de endereço constituído pelo endereço IP do *host* e um número de 16 *Bits* local para esse *host*, chamado de porta. Para que o serviço TCP funcione é necessário que uma conexão seja estabelecida entre um soquete da máquina transmissora e um soquete da máquina receptora, podendo ser utilizado o mesmo soquete para várias conexões simultâneas. Uma das características do TCP é ter todas as suas conexões *full-duplex*, o que significa que todo o tráfego pode ser

transmitido em ambas as direções e ao mesmo tempo, bem como, possuir conexões ponto a ponto, em que cada conexão possui exatamente dois pontos terminais. Todas as informações transmitidas e recebidas pelas entidades através do TCP são em formato de segmentos. Estes possuem o tamanho definido pelo software TCP. Nele podem estar contidos os dados de várias gravações ou apenas uma parte de uma gravação que foi dividida em mais de um segmento. O segmento do TCP se inicia com um cabeçalho fixo de 20 *bytes*, que se refere ao cabeçalho IP, sendo seguido pelo cabeçalho TCP, conforme Figura 10.

Figura 10 – Cabeçalho do UDP



Fonte: Modificado de Tanenbaum (2003).

- Protocolos da Camada de Aplicação: Para que dois sistemas finais possam se comunicar é preciso que ocorra a compreensão das mensagens e, para tanto, existem os protocolos da camada de aplicação. É de extrema importância que não sejam confundidos os protocolos da camada de aplicação com as aplicações em si. Ao ser executada uma aplicação e caso haja a necessidade de realizar a comunicação através da rede, será utilizado um protocolo da camada de aplicação. (KUROSE, 2010). Quando for preciso enviar as informações de um remetente para um *host* de destino, a aplicação utilizará para o envio o protocolo adequado, conforme padrão já estabelecido, para que ao receber as informações o *host* de destino possa, por meio desse protocolo utilizado, entender o padrão e o conteúdo que foi transmitido (KUROSE, 2010).

Atualmente os protocolos de aplicação estão presentes em quase todos os aplicativos utilizados na rede. Kurose (2010) explana sobre diversos protocolos e cita como um dos mais utilizados na atualidade o protocolo HTTP, responsável pelas comunicações na web e o HTTPS utilizado também para *web*, seguros devido à criptografia. Além do HTTP e HTTPS é possível listar diversos protocolos e suas portas utilizadas para a correta funcionalidade, como mostra o Quadro 3.

Quadro 3 – Protocolos de aplicação e suas portas

Protocolos	Nome	Portas	Descrição
SSH	*Secure Shell*	22	Permite a conexão e execução remotamente
HTTP	*Hypertext Transfer Protocol*	80	Protocolos de transferência de hipertexto para WEB
SNMP	*Simple Network Management Protocol*	161	Protocolo de gerenciamento de rede simples
POP3	*Post Office Protocol*	110	Protocolo para recebimento de e-mail
SMTP	*Simple Mail Transfer Protocol*	25	Protocolo para envio de e-mail
FTP	*File Transfer Protocol*	21	Protocolo de transferência de arquivos
NTP	*Network Time Protocol*	123	Protocolo de hora para rede de computadores
DNS	*Domain Name System*	53	Protocolo para serviços de nomes de domínio

Fonte: Modificado de Kurose (2010)

Cada um destes protocolos do Quadro 3 possui a sua responsabilidade e função nas redes de computadores, o uso destes protocolos não implica em garantir a segurança contra ameaças a redes de computadores, onde para isso é necessário que se tenha o conhecimento sobre ameaças a redes de computadores.

3. SEGURANÇA DA INFORMAÇÃO

3.1 Ameaças a Redes de Computadores

Atualmente a Internet tornou-se algo essencial para instituições, empresas, universidades, órgãos do governo e para fins profissionais, sociais e pessoais. Kurose (2010) ressalta que, da mesma forma que existem todas as utilidades da Internet e das redes de computadores, igualmente existem as situações em que diversos tipos de ameaças tentam causar problemas no cotidiano das pessoas e organizações, danificando os dispositivos conectados na Internet. Ameaças a redes de computadores, para Wendt (2013), podem ser tratadas como uma ação criminosa e chamadas de ações de crimes cibernéticos. Entre todos os tipos de ameaças e invasões possíveis às redes, a engenharia social é a ação criminosa utilizada em conjunto com praticamente todas as outras ações que podem ser realizadas na rede. Uma invasão à segurança de uma empresa, quase sempre se inicia com a obtenção de uma informação ou documento que aparenta não ter importância para seu proprietário. Segundo afirma Mitnick (2003), isto ocorre porque o proprietário não protege estas informações e documentos, devido ao fato de não conseguir visualizar o motivo para o qual deva ser protegido e restrito; nestes casos, esta informação se transforma numa vulnerabilidade a ser explorada pela ameaça, podendo iniciar uma série de crimes cibernéticos.

Considera-se uma ameaça toda e qualquer atitude ou ato que venha comprometer, em qualquer aspecto, um ou mais recursos e seus ativos tecnológicos, físicos e humanos. Estas ameaças podem ser classificadas em ameaças naturais, acidentais ou intencionais, exemplificando as ameaças em geral como, vírus, *trojans*, roubo e furto de equipamentos, fenômenos atmosféricos, engenharia social, instalações físicas inadequadas, incêndio, desabamentos e inundações (COSTA, 2011). A classificação das ameaças entre naturais, acidentais ou intencionais é explicada por Sêmola (2003). As ameaças naturais são ocasionadas por fenômenos da natureza, como incêndios naturais, terremotos, tempestades eletromagnéticas, maremotos, aquecimento, poluição, enchentes podendo ser incluído nesta classificação qualquer outro incidente de âmbito natural.

As ameaças acidentais são ocasionadas, quase sempre, por acidentes e erros causados pela falta de conhecimento ou descuido. As ameaças intencionais são ocasionadas com a intenção que a falha realmente ocorra; procedimento este comumente realizado por agentes humanos, *crackers*, invasores, espiões, ladrões, incendiários, depredadores e, ainda, podendo ser incluída qualquer outra ameaça ocasionada pela má fé do praticante. Outro aspecto importante, associado a redes diz respeito à vulnerabilidade. Vulnerabilidades podem ser definidas, segundo Costa (2011), como condições a serem exploradas pelas ameaças, para assim comprometer os recursos disponíveis em um ambiente.

A falta de conhecimento dos funcionários sobre a política de segurança e suas regras pode ser classificada como uma vulnerabilidade que pode ser agravada quando ainda não há regras ou políticas de gestão de segurança da informação, quando há falta de infraestrutura de TI, ou estas infraestruturas sejam mal planejadas como, por exemplo, os pontos de acesso para redes sem fio disponíveis ao público e conectados na rede interna empresarial. Tais vulnerabilidades podem ser encontradas em processos, políticas, procedimentos, equipamentos e nos recursos humanos. Dantas (2001) afirma que vulnerabilidades por si só não provocam as ações criminosas e que para tanto se faz necessário um agente causador, considerando que estas vulnerabilidades estão relacionadas diretamente com as fragilidades. Por meio das informações obtidas em pesquisas realizadas pela Módulo Security S.A., nos anos de 2002 e 2003 (DANTAS, 2011), o Quadro 4 demonstra as principais vulnerabilidades.

Quadro 4 – Vulnerabilidades no período de 2002 e 2003

VULNERABILIDADES	2002	2003
Principais Pontos de Invasão		
Internet	55%	60%
Sistemas Internos	20%	23%
Principais Responsáveis		
Hackers	48%	32%
Funcionários	24%	23%
Prestadores de Serviços	12%	4%
Principais Obstáculos para Implementação da Segurança da Informação		
Falta de consciência dos executivos	33%	23%
Falta de consciência dos usuários	29%	14%

Fonte: Dantas (2011)

Segundo CERT (2012), ataques ocorrem a qualquer computador que possa estar conectado em uma rede ou que seja acessível pela Internet. É preciso destacar que estando qualquer computador conectado à Internet este pode participar de um ataque, de forma voluntária ou involuntária. Não existe um único motivo para que os ataques sejam realizados, porém, podem ser citados exemplos como: ataques ocasionados por demonstração de poder e conhecimento; motivações financeiras e ideológicas; motivações comerciais, bem como, qualquer outra forma pela qual os atacantes possam se expressar. Os ataques podem ser divididos em duas categorias, visando uma definição clara e uma melhor forma de catalogar estes ataques. Stallings (2005) define estas duas categorias como ataques passivos e ataques ativos. Ataques passivos são mais difíceis de detectar, pelo fato de não envolverem alteração nos dados; o objetivo deste ataque é obter informações desejadas através do monitoramento das transmissões internas na rede em tempo real, sendo que, para evitar uma série desses ataques faz-se o uso da criptografia. Ataques ativos possuem uma maior facilidade para detecção, visto que este tipo de ataque envolve uma modificação ou, em alguns casos, a criação de um fluxo falso de dados; este ataque pode ser dividido em quatro categorias: falsidade, repetição, modificação de mensagens e negação de serviço. O ataque ativo de falsidade ocorre geralmente, quando o atacante tenta se passar por um usuário autenticado da rede. Já o ataque de repetição envolve a captura passiva dos dados e é realizada a

retransmissão destes dados, a fim de simular um efeito de não autorizado. O ataque de modificação de mensagens consiste em o atacante capturar a mensagem a ser encaminhada e modificar seu conteúdo original, podendo assim, realizar as funções desejadas para as quais não possui autorização; os ataques de negação de serviço são responsáveis pela desativação temporária de funções, sistemas e redes; este ataque, quase sempre, é realizado com uma sobrecarga de mensagens ao alvo. Outro item importante, relacionado às redes são os riscos.

Risco pode ser definido como a consequência entre as ameaças e vulnerabilidades, sobre todos os ativos de uma organização. Geralmente, a classificação desses riscos ocorre em 3 níveis que são: baixo, médio e alto. O critério a ser estabelecido para classificar quais são os níveis dos riscos deve ser analisado pelos gestores da organização, uma vez que não existe uma forma padrão de se classificar. Duas empresas distintas podem estar sujeitas ao mesmo risco, porém, cada uma possuindo uma consequência no caso de concretização da ameaça (COSTA, 2011). Avaliações de riscos tornam-se necessárias para que sejam classificados os riscos das vulnerabilidades e ameaças dos ativos de uma organização, segundo Costa (2011). Conforme a realização destas avaliações para cada tipo possível de risco, Costa (2011) afirma que se deve tratar a classificação dos riscos como baixa para as ameaças que tenham fácil solução, que causem prejuízos pequenos e passem despercebidas pela maioria dos integrantes da equipe e do público externo. Os riscos de classificação considerados altos podem ser definidos para as ameaças e vulnerabilidades que possam oferecer uma grande perda financeira para a empresa, uma solução que demande muito tempo e recursos para sua correção e que possam ainda vir a comprometer o negócio ou a reputação da empresa. Dessa forma, o risco de classificação média coloca-se como o meio termo entre os dois já definidos. Para que esta classificação seja realizada é preciso determinar os limites em função dos custos, tempo e repercussão em caso de ocorrência para cada tipo de classificação. Após o levantamento de todas as atividades realizadas pela empresa como, por exemplo, a infraestrutura, sistemas, políticas e recursos existentes, a análise de riscos pode ser feita para que, dessa forma, sejam identificadas as ameaças e vulnerabilidades bem como classificados os riscos existentes (COSTA, 2011).

3.2 Segurança da Informação

A segurança da informação é a garantia da preservação da confidencialidade, integridade e disponibilidade das informações. A informação é um ativo que como qualquer outro ativo importante, é essencial para as organizações e, por essa razão, precisa ser protegida. Segurança da informação é a proteção destes ativos, dos mais diversos tipos de ameaças, para que se possa garantir a continuidade do negócio, minimizar os riscos e maximizar os retornos sobre o investimento. Esta segurança é obtida a partir da implementação de um conjunto de controles tais como: políticas, processos, procedimentos, estruturas organizacionais e funções de *software* e *hardware*. (NBR ISO/IEC 27002:2013).

Durante as últimas décadas, devido aos incidentes de segurança que a cada dia estão mais presentes nos meios de comunicação, a segurança da informação tem passado por diversas mudanças. Stallings (2005) ressalta que, anteriormente, as informações, em sua maioria, eram produzidas e armazenadas em papel, e quando consideradas valiosas ou de grande importância tinham sua proteção constituída principalmente por meios físicos, como o uso de armários com trancas e segredos para guardar os documentos importantes que deveriam ser protegidos dos acessos indevidos. Com a introdução dos computadores e as informações em formato digital, tornou-se evidente a necessidade de ferramentas automatizadas para proteger os dados e deter os *crackers*. Estas ações e o uso destas ferramentas foram denominados de “segurança de computadores” e, foram criadas medidas de segurança para que quando da utilização de sistemas distribuídos e uso de redes, seja possível proteger os dados durante sua transmissão. De acordo com Fontes (2006), a segurança da informação pode ser definida como um conjunto de orientações, normas, procedimentos, ações e políticas existentes para minimizar o risco do negócio em relação à dependência do uso dos recursos de informação. A segurança tem como objetivo proteger a informação, garantindo, assim, a possibilidade da continuidade dos negócios e permitindo que a missão seja alcançada. Salienta-se que o uso indevido da informação ou a falta da informação para o negócio pode ocasionar perdas que comprometam o seu funcionamento e obtenção de fundos. A segurança da informação existe para minimizar os riscos do negócio. Fontes (2006) define que proteger a informação significa garantir seis

objetivos: disponibilidade, integridade, confidencialidade, legalidade, auditabilidade e não repúdio à autoria. A disponibilidade visa garantir que a informação esteja sempre disponível para o funcionamento e o uso da organização, permitindo, assim, o alcance de seus objetivos e missões. A integridade tem a função de garantir que a informação está correta, que é verdadeira e que não está corrompida, evitando que ocorram falhas ao acessar esta informação. A confidencialidade garante que a informação somente deva ser acessada e utilizada por aqueles que dela necessitam e, nesse sentido, necessita-se do acesso quando classificada em níveis. A legalidade garante que a informação esteja de acordo com as leis, as licenças de uso e os contratos. A auditabilidade visa garantir que se verifique quem acessou a informação e o que realizou e, com isso, obter o registro de acessos e modificações. O não repúdio de autoria tem a função de garantir que o autor da informação, ou o responsável pela alteração da informação, não possa negar a autoria, pois existem meios de comprovar e garantir o autor das edições.

A proteção desses aspectos de segurança passa por um conjunto de escolhas, ações e ferramentas. Uma dessas ferramentas, que tem por objetivo proteger as informações que trafegam em uma rede de computador é o *firewall*.

O *firewall* permite que seja realizada uma filtragem no local a ser utilizado para obter apenas as informações que são desejadas e autorizadas, permitindo maior controle e mais segurança, por meio das configurações previamente realizadas.

3.3 Firewall

Atualmente a maioria das empresas e organizações possuem informações confidenciais que, se reveladas a um concorrente podem provocar sérias consequências para as proprietárias das informações. Estando estas empresas conectadas à rede é preciso que se tenha a preocupação em manter somente os *Bits* importantes e confiáveis entrando e saindo da rede e, com isso, descartando os *Bits* que não podem e não devem ser tratados. Para que seja realizado um tratamento dos *Bits* de uma maneira eficaz, podem ser realizadas a implementação

e a utilização das ferramentas chamadas de *firewall,* que irão tratar todo o tráfego de entrada ou de saída da rede fazendo com que cada pacote passe pelo seu filtro.

O *firewall* sendo o único computador conectado diretamente a Internet terá a função de garantir que somente os serviços autorizados cheguem de forma segura ao seu destino. Ao utilizar uma infraestrutura em que não se possui o *firewall* conectado diretamente à rede externa, é preciso que cada *host* da rede contenha meios próprios de garantir a sua segurança, não existindo, desse modo, um filtro preestabelecido com os serviços e protocolos a serem filtrados e permitidos. É importante salientar que os *firewalls* não evitam que os *host*s de uma rede sejam infectados por vírus, pois, a maioria das contaminações tende a ser causada pelo próprio usuário (D'OLIVEIRA NETO, 2004).

Um erro grave que foi cometido inicialmente e até hoje é cometido muitas vezes, é o de utilizar as funções do *firewall* somente para agentes externos. A utilização do *firewall* configurado apenas para verificar os pacotes da rede externa se deve, segundo D'Oliveira Neto (2004), ao fato de acreditar que as ameaças estão somente na Internet e não nos usuários internos da rede. A grande maioria dos casos de invasões à rede interna tem alguma parcela de culpa dos usuários internos, não sendo esta necessariamente intencional. O *firewall* pode ser dividido em três classes: filtro de pacotes, NAT e híbrido.

Filtro de pacotes é a classe de *firewall* mais utilizada na atualidade; este trabalha decidindo o destino de um pacote, podendo descartar ou aceitar o mesmo mediante uma comparação das regras adicionadas em sua configuração; o não uso deste *firewall* significa deixar todas as portas abertas permitindo a livre circulação de pacotes na rede. O *firewall* NAT é a classe de *firewall* responsável pela manipulação da rota padrão dos pacotes, manipulando o endereço de origem e destino dos pacotes, chamado por muitos de "tradução de endereçamento". O *firewall* híbrido é o conjunto dos dois *firewalls* citados anteriormente e que realizam a filtragem de pacotes e a tradução de endereços.

Como destacado o *firewall* é uma ferramenta de extrema importância para a segurança das redes de computadores, pois quando configurado corretamente conforme o estudo e as análises realizadas sobre a rede a ser protegida tem a

função de protegê-la. Visto a grande quantidade de *firewall* disponíveis no mercado é preciso obter o esclarecimento sobre qual o mais indicado e qual o ideal dentre todas as opções existentes no mercado. O próximo capítulo visa obter resultados através da comparação entre duas destas ferramentas – *Iptables* e *Ipfw*.

4. COMPARATIVO ENTRE FERRAMENTAS DE SOFTWARE

Conforme citado nos capítulos anteriores existem diversos tipos de ferramentas destinadas a proteção de rede. Este capítulo apresenta duas ferramentas de *firewall* disponíveis no mercado, comparando-as entre si. Estas ferramentas são: *iptables* e *ipfw*. Para realizar a comparação entre elas, serão projetados 2 ambientes. Cada ambiente possuirá 2 cenários, idênticos para os 2 ambientes. Inicialmente serão demonstradas algumas configurações e características a fim de elucidar sobre as ferramentas utilizadas, posteriormente a testes serão feitos a fim de se obter os resultados que serão apresentados. A seguir, apresenta-se a configuração utilizada para a criação dos ambientes através do *software VirtualBox.*

4.1 Ambientes e cenários

Para a realização do experimento serão utilizados dois ambientes, onde o “Ambiente 1” utilizará a ferramenta *iptables* enquanto o “Ambiente 2” usará o *ipfw*. Em cada ambiente serão realizados 2 cenários sendo que para a realização destes será mantido o mesmo padrão de testes e serão realizadas as configurações mais próximas possíveis entre os dois ambientes. O ambiente virtual utilizado foi o seguinte: Três máquinas virtuais e uma máquina física, que será responsável pela distribuição da rede. As especificações técnicas de *hardware* e de *software* utilizadas no cenário são para a máquina hospedeira o uso do *hardware*, processador *Core 2 Duo* de 2.00Ghz, 4GB de memória RAM DDR2 e disco rígido de 500GB. Para o *software* é utilizado do sistema operacional *Windows 8 Pro 64 Bits* possuindo instalado o *software* de virtualização *Oracle VM VirtualBox* versão 4.3.12 r93733.

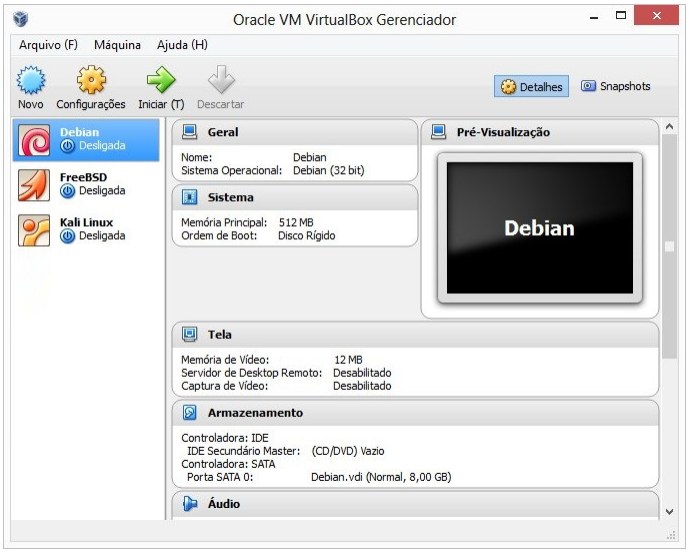
Para as máquinas virtuais serão configurados dois servidores sendo o utilizado no ambiente 1 com sistema operacional *Debian* 32*Bits* versão 7.5 com 512MB de memória RAM, e o utilizado para o segundo ambiente terá o sistema operacional *FreeBSD* 32 *Bits* versão 10.0 também com 512MB de memória RAM.

Ambas as versões depois de instaladas possuirão cada uma seu *firewall* nativo instalado, sendo que para o *Debian* é disponibilizado o *Iptables* e para o *FreeBSD* o *ipfw.* Para que se possa realizar o teste da aplicabilidade da regra aplicado no *firewall* será utilizada uma terceira máquina virtual, máquina esta com o sistema Operacional *Kali Linux* 32 *Bits* versão 3.14.5.

4.1.1 Configuração do *VirtualBox*

Para que seja possível manter o maior padrão entre os dois ambientes, as configurações de hardware disponibilizadas para as máquinas contendo o sistema operacional *Debian* e *FreeBSD* serão idênticas conforme é possível verificar nas figuras a seguir (Figuras 11, 12, 13, 14 e 15).

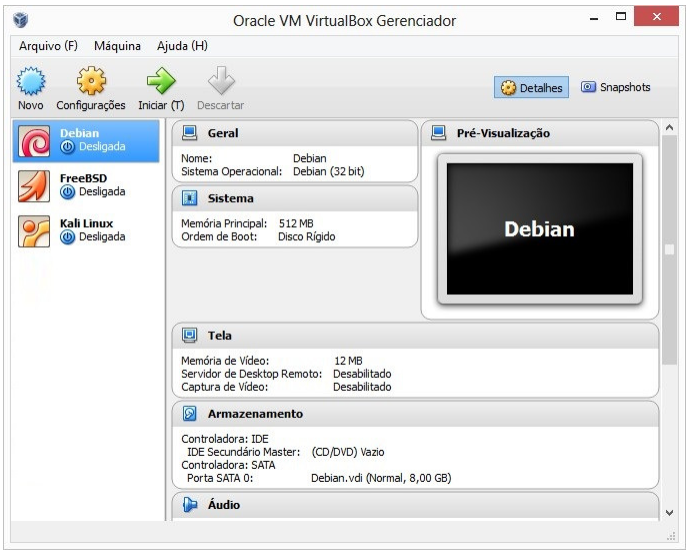
Figura 11 – Máquinas virtualizadas



Fonte: Elaborado pelo Autor

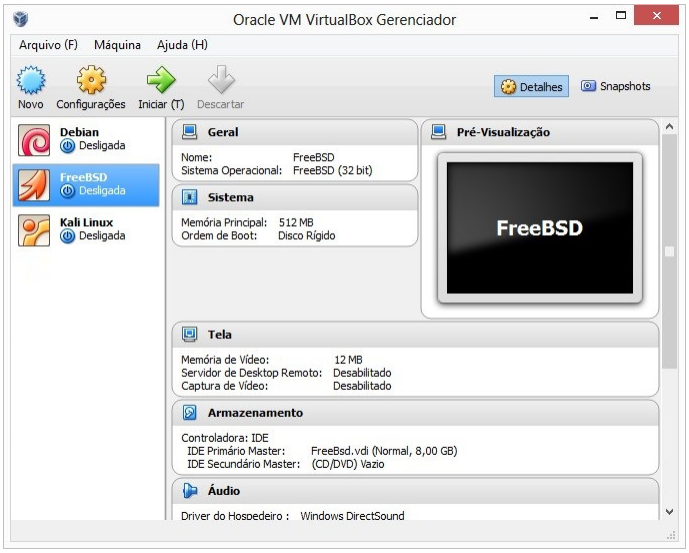
Através das Figuras 12 e 13 é possível verificar que tanto o servidor *Debian* quanto o servidor *FreeBSD*, dispõem de 512MB de memória RAM e 12MB disponíveis para vídeo.

Figura 12 – *Debian*



Fonte: Elaborado pelo Autor

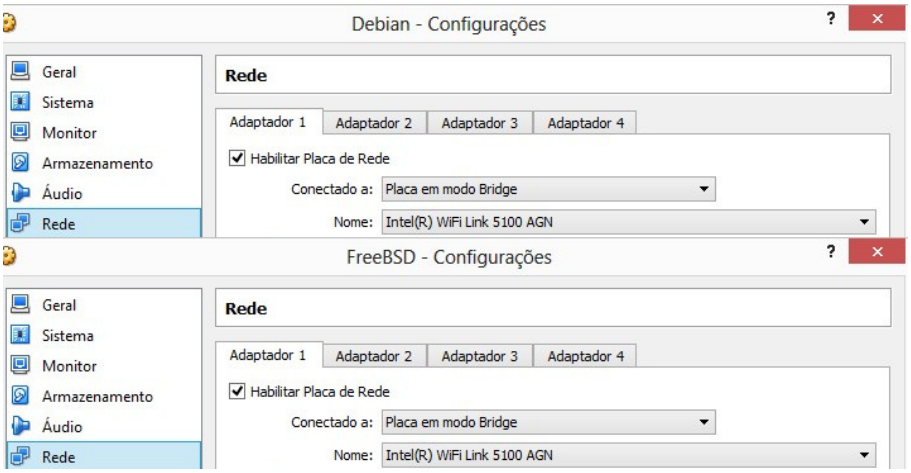
Figura 13 – *FreeBSD*



Fonte: Elaborado pelo Autor

Para a comunicação entre as máquinas virtuais e a máquina local que será responsável pela disponibilização da rede externa, os dois servidores irão dispor de uma placa de rede ativa em modo *bridge* conforme é possível verificar na Figura 14

Figura 14 – Interfaces de rede



Fonte: Elaborado pelo Autor

A máquina atacante irá dispor de um ambiente similar aos dois servidores, possuindo esta também 512 MB de memória RAM e 12MB disponíveis para vídeo, a unidade de rede também estará disponível no modo *bridge*, conforme é possível ser visualização na Figura 15.

Figura 15– *Kali Linux*

Fonte: Elaborado pelo Autor

Com estas configurações realizadas para cada máquina virtual será iniciada a realização dos testes nos ambientes e cenários.

4.2 Ambiente 1: Testes com *iptables*

O ambiente 1, conforme já foi explanado, disporá do servidor *Debian* contendo o *firewall iptables* e a máquina atacante contendo o sistema *Kali Linux*.

4.2.1 Características do *Iptables*

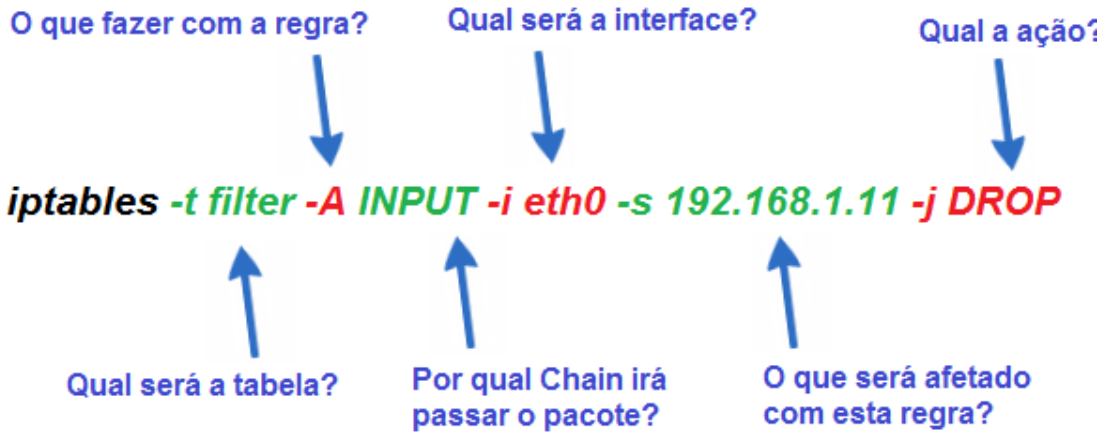
O *iptables* foi incluído no Linux através do *kernel* 2.4 em meados de 1999, substituindo o então utilizado *ipchans*. Chamado também por muitos de *netfilter*, porem *netfilter* é o nome do módulo que fornece ao Linux as funções de *firewall* e NAT, sendo assim é possível afirmar que o *iptables* é a ferramenta que controla o *netfilter,* visto que através dele é possível a criação de regras de *firewall* e NAT.

As regras criadas no *iptables* serão executadas exatamente na ordem em que estiverem dispostas, ao receber um pacote será realizada uma comparação com as regras para saber se este terá ou não permissão para passar pelo *firewall*.

Para que seja possível realizar as configurações na ferramenta de *firewall iptables* é preciso que se tenha o conhecimento da estrutura de suas regras, e o que são as tabelas, *chains*, comandos, parâmetros e ações.

Para isso é possível observar a Figura 16 que exemplifica graficamente uma regra.

Figura 16 – Regra do *iptables*



Fonte: Elaborado pelo Autor

Neste exemplo é descrita uma regra com o intuito de descartar toda a comunicação realizada pelo ip 192.168.1.11, para maior entendimento é possível ler a regra descrita da seguinte forma, “Tudo o que for recebido pelo *firewall* através da placa de rede *eth0*, proveniente do ip, 192.168.1.11 deve ser descartado”.

Como é observado na figura o comando *iptables* é responsável por executar as funções na ferramenta. Após isso tem-se a tabela onde será inserida a regra, neste caso é a tabela *filter* (-t *filter*). Após isso é informado o que será feito com esta regra, onde a letra A é responsável por informar que será adicionado dentro da lista de regras.

O comando input é a *chain* por qual passará essa informação, ou seja, tudo o que passar pela entrada, -s especifica que um IP será afetado e logo após tem-se qual é este IP. A última informação a ser preenchida é qual será a ação realizada por esta regra, neste caso é o *DROP*, ação esta que tem por finalidade descartar qualquer informação. Este exemplo de regra é utilizado com frequência após analises de *LOG's* onde se constate que o servidor está sendo alvo de ataque de um determinado IP.

4.2.2 Cenário 1: *DoS* a porta 80

O ataque *DoS* (*Denial of Services*) direcionado a uma porta especifica tem por objetivo sobrecarregar todos os recursos de uma porta a fim de tornar indisponíveis os serviços providos através dela. Para este teste foi instalada e configurada uma página simples através do apache2 no servidor *Debian* tornando disponível assim uma página na rede, conforme mostra a Figura 17.

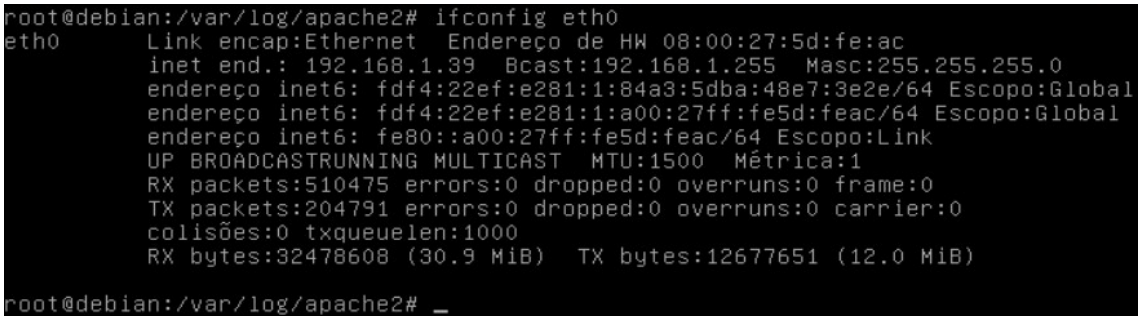
Figura 17 – Servidor *Debian* com Apache2

Fonte: Elaborado pelo Autor

Através da Figura 17 é possível verificar a disponibilidade do servidor na rede externa, é possível visualizar também que o IP disponível é 192.168.1.39,

conforme é confirmado através do comando *ipconfig eth0* realizado direto no servidor e visto na Figura 18.

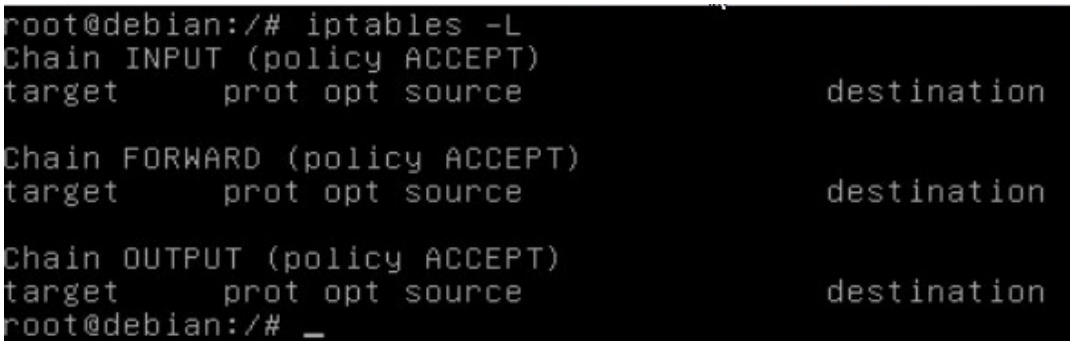
Figura 18 – IP verificado direto no servidor *Debian*



Fonte: Elaborado pelo Autor

Primeiro será simulado o ataque a esta porta sem que o servidor possua regra alguma configurada. Para isso é possível visualizar se existem e se sim quais são estas regras através do comando *iptables* -L, conforme Figura 19, onde se verifica a não existência de nenhuma regra.

Figura 19 – Listagem das regras no *iptables*

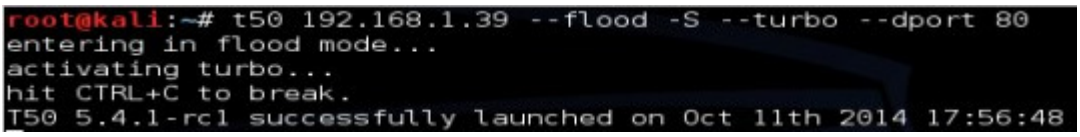


Fonte: Elaborado pelo Autor

Sabendo o IP ou endereço do servidor, é possível realizar a simulação do ataque através da máquina *Kali Linux* utilizando da ferramenta T50, ferramenta esta de ampla utilização em testes de vulnerabilidades em servidores, onde será realizado o comando t50 192.168.1.39 --*flood -S --turbo --dport 80*.

Para que se tenha uma maior compreensão deste comando é possível fazer a leitura dele da seguinte forma “A ferramenta t50 enviará informações ao IP alvo 192.168.1.39, através de uma inundação de TCP SYN, com grande desempenho direcionado a porta 80”. É possível visualizar este comando sendo realizado através da Figura 20.

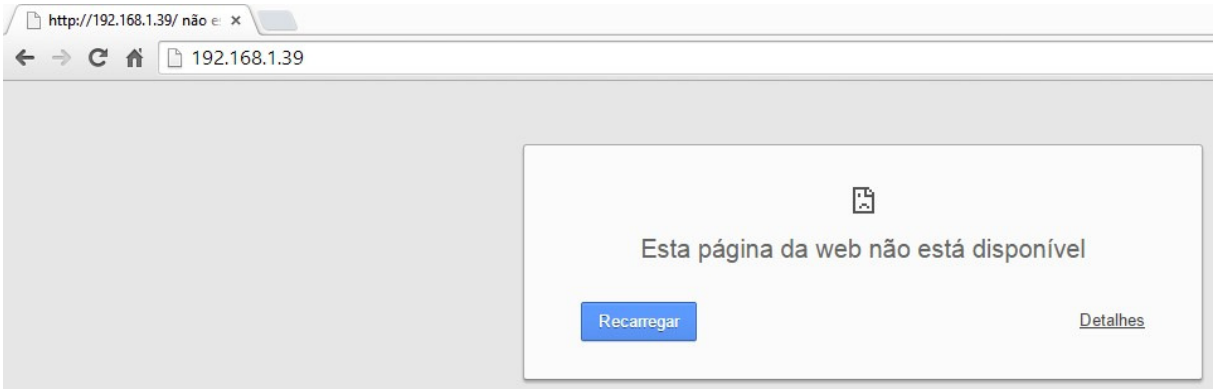
Figura 20 – Ferramenta de simulação de ataque



Fonte: Elaborado pelo Autor

Após a realização deste comando foi cronometrado o tempo aproximado de 5 minutos até que o servidor *Debian* estivesse totalmente inativo não sendo mais possível obter o acesso à página disponível, conforme Figura 21.

Figura 21 – Página do servidor inacessível.



Fonte: Elaborado pelo Autor

Após a realização desta simulação torna-se necessária a tentativa de correção do problema através de uma ou mais regras na ferramenta de *firewall* existente no servidor *Debian*, o *iptables*.

Para a realização da tentativa de proteção contra o DOS na porta 80 serão aplicadas duas regras, *iptables -A FORWARD -p tcp --syn -m limit --limit 10/s -j ACCEPT* e *iptables -A FORWARD -p tcp --syn -j DROP*, onde é possível ler estas regras da seguinte maneira, "Adicionar uma regra no *iptables* para tudo o que entrar no servidor e for um *tcp syn* ter um limite máximo de 10 entradas aceitas por segundos. Caso ultrapasse a 11 ou mais por segundo deve ser descartado."

A Figura 22 demonstra a execução da listagem de regras já configuradas através do comando “iptables –L” e após isso a configuração das duas regras já descritas.

Figura 22 – Regras aplicadas no *iptables*

```
root@debian:/var/log/apache2# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@debian:/var/log/apache2# iptables -A FORWARD -p tcp --syn -m limit --limit
10/s -j ACCEPT
root@debian:/var/log/apache2# iptables -A FORWARD -p tcp --syn -j DROP
root@debian:/var/log/apache2# _
```

Fonte: Elaborado pelo Autor

Após a aplicação destas regras foi realizada uma nova simulação do mesmo ataque direcionado à porta 80 do servidor, conforme observa-se na Figura 23.

Figura 23 – Segunda tentativa de ataque ao servidor

```
root@kali:~# t50 192.168.1.39 --flood -S --turbo --dport 80
entering in flood mode...
activating turbo...
hit CTRL+C to break.
T50 5.4.1-rc1 successfully launched on Oct 11th 2014 18:31:01
```

Fonte: Elaborado pelo Autor

Com as regras configuradas da maneira que foram detalhadas é possível observar, na Figura 24, que a página se manteve acessível durante toda a simulação, simulação esta que teve um tempo cronometrado de 30 minutos. Notou-se que durante todo o teste o acesso a página tornou-se lenta.

Figura 24 – Página disponível durante simulação de ataque DOS

Fonte: Elaborado pelo Autor

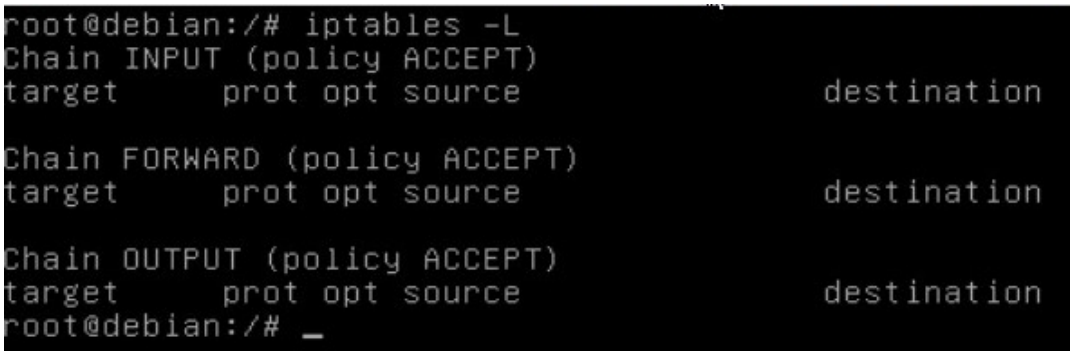
4.2.3 Cenário 2: *Brute force* ao serviço SSH

O ataque *Brute force* (Força Bruta), consiste em realizar repetidas tentativas automatizadas de possibilidades de senhas e usuários. Para este tipo de ataque é utilizada em sua maioria grandes listas incluindo diversas possibilidades de usuários e senhas, onde o objetivo é que uma dessas seja válida, sendo assim possível obter o usuário ou a senha desejada.

Para a realização deste teste o *brute force* foi direcionado ao serviço SSH, serviço este que por padrão utiliza da porta 22 e está disponível em quase todos os servidores Linux configurados.

A realização da primeira simulação será realizada sem que exista uma regra configurada na ferramenta de *firewall iptables*, visualizando as regras através do comando *iptables* –L, conforme se vê na figura 25.

Figura 25 – Listagem das regras do *iptables* no servidor *Debian*

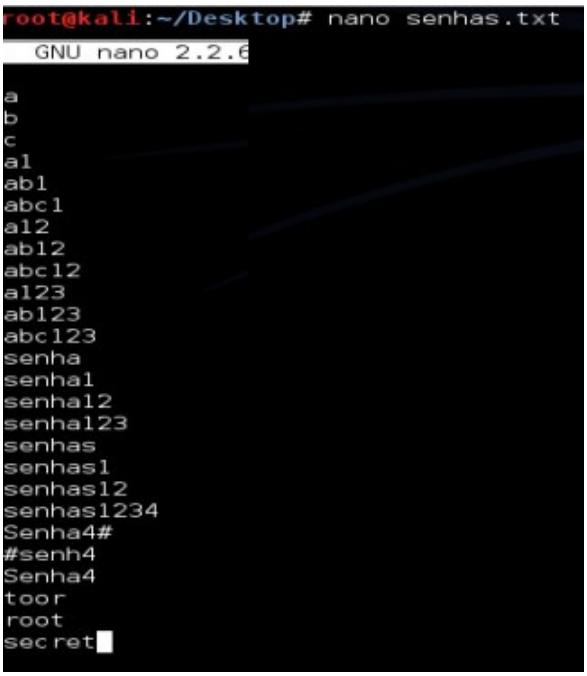


Fonte: Elaborado pelo Autor

Para realizar esta simulação através da máquina virtual *Kali Linux* será utilizada a ferramenta *hydra*, usada para a maioria dos testes de vulnerabilidades a força bruta em servidores.

Para realizar este teste será criada uma lista com algumas possibilidades de senhas conforme apresentado na Figura 26, onde será inserida a senha válida para o usuário “cliente” e além desta, conterá também outras possibilidades de senhas.

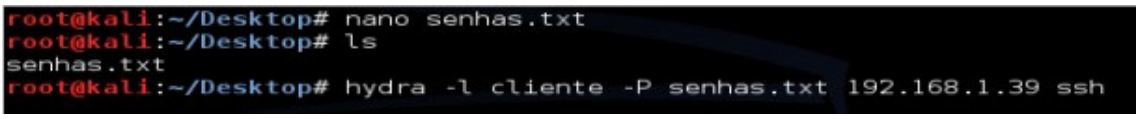
Figura 26 – Arquivo contendo as possibilidades de senhas



Fonte: Elaborado pelo Autor

Após a criação do arquivo com senhas aplica-se o comando *hydra* -l cliente -P senhas.txt 192.168.1.39 ssh a fim de iniciar o teste. Para que se tenha uma maior compreensão é possível ler este comando da seguinte forma “A ferramenta *hydra* trabalhará com o usuário cliente e a lista de senhas, senhas.txt, direcionada ao ip 192.168.1.39 ao serviço ssh”, conforme se observa na Figura 27.

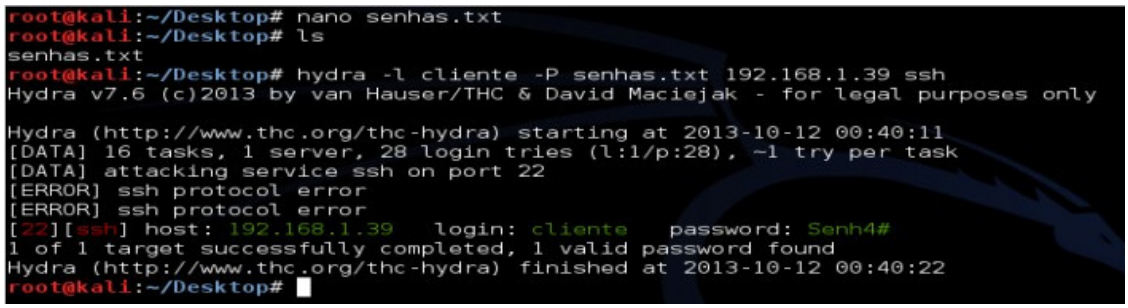
Figura 27 – Ferramenta *hydra* de simulação de ataque



Fonte: Elaborado pelo Autor

Após a realização do comando é possível observar a resposta informada pela ferramenta *hydra* conforme Figura 28, onde se apresentam as informações obtidas com as tentativas de acessos.

Figura 28 – Resposta da ferramenta *hydra*



Fonte: Elaborado pelo Autor

Após obter confirmação da existência do usuário e a senhas para este usuário é necessário confirmar se realmente é possível acessar com estas informações conforme mostra a Figura 29.

Figura 29 – Teste do acesso SSH

Fonte: Elaborado pelo Autor

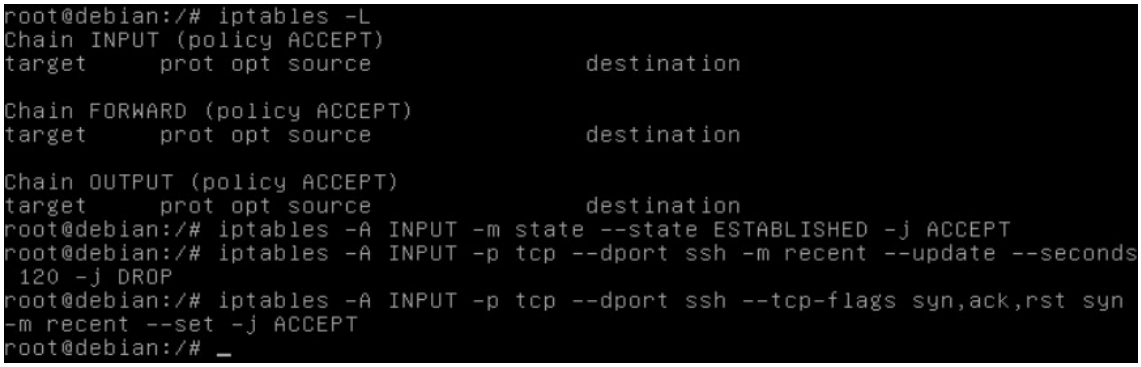
Após verificar a vulnerabilidade é necessária a realização da tentativa de proteção contra o *brute force* direcionado ao SSH. Para isso serão aplicadas três regras.

a) *iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT*
b) *iptables -A INPUT -p tcp --dport ssh -m recent --update --seconds 120 -j DROP*
c) *iptables -A INPUT -p tcp --dport ssh --tcp-flags syn,ack,rst syn -m recent --set -j ACCEPT*

É possível ler estas regras da seguinte forma, (a) "*iptables* adicione uma regra para tudo o que for entrada e já tiver alguma vez estabelecido conexão com o servidor uma permissão de entrada". Para a regra (b) é possível ler da seguinte forma, "Adicionar no *iptables* uma regra para todas as entradas com o protocolo tcp direcionados a porta ssh que estiverem com tentativas recentes de comunicação uma negação de acesso por 120 segundos". E para a regra (c) é possível fazer a interpretação da seguinte maneira, "Adicionar ao *iptables,* para tudo o que for entrada com o protocolo tcp direcionado a porta ssh que contenha as *flas syn, ack,* e *rst syn* uma permissão de entrada"

É possível observar estas configurações sendo realizadas na Figura 30.

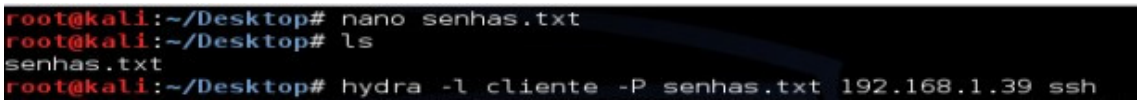
Figura 30 – Regras aplicadas no *iptables*



Fonte: Elaborado pelo Autor

Após a aplicação destas regras foi realizada uma nova simulação do mesmo ataque direcionado ao serviço SSH do servidor, conforme se observa na Figura 31.

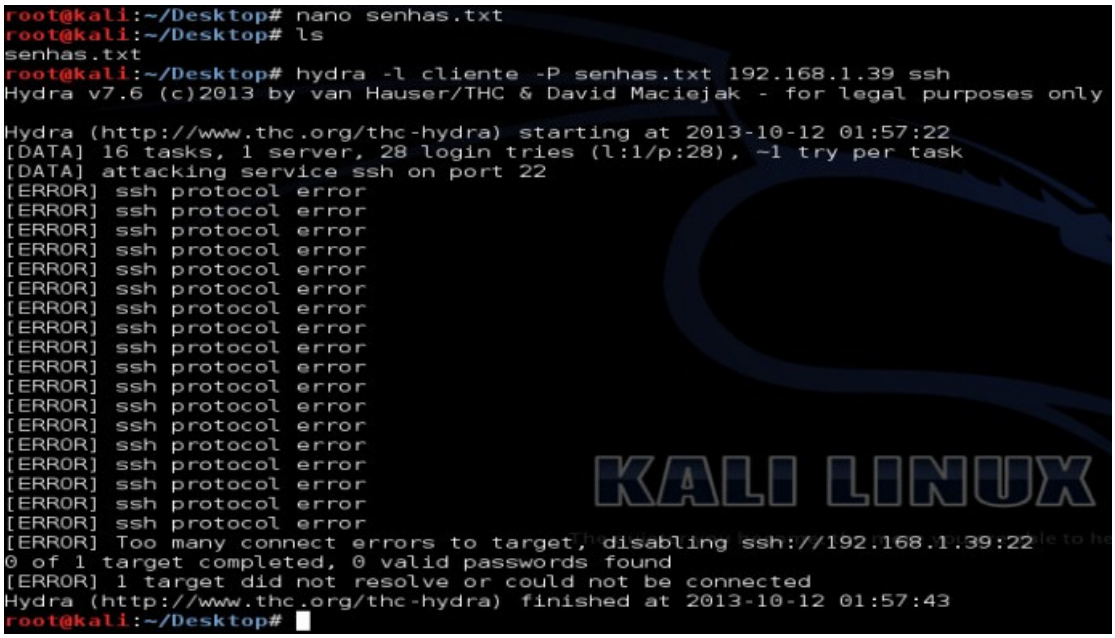
Figura 31 – Segunda tentativa de ataque ao servidor



Fonte: Elaborado pelo Autor

Após a execução do comando estando as regras configuradas dentro da ferramenta *iptables* no servidor é possível observar através da própria ferramenta que não foi possível obter a senha do usuário cliente, conforme ilustrado através da Figura 32.

Figura 32 – Não obtenção da senha através da ferramenta *hydra*.



Fonte: Elaborado pelo Autor

4.3 Ambiente 2: Testes com *ipfw*

O ambiente 2 descrito, disporá do servidor com sistema *FreeBSD* contendo a ferramenta de *firewall ipfw* e a máquina atacante contendo o sistema *Kali Linux*.

4.3.1 Características do *Ipfw*

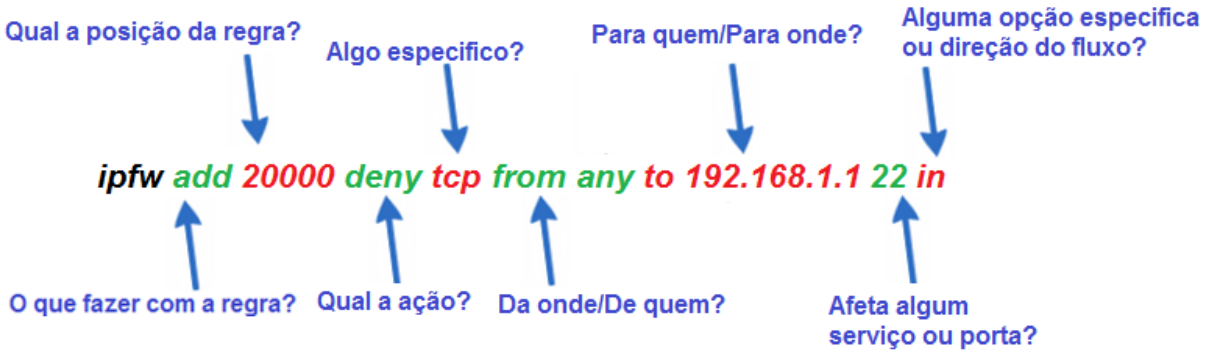
O *ipfw* é o *firewall* nativo para o sistema operacional *FreeBSD*, apesar de ser um *firewall* nativo esta ferramenta por padrão vem desabilitada sendo necessário habilitar. O *ipfw* na realidade é o nome da interface de comando do *IPFIREWALL*.

As regras existentes no *ipfw* serão executadas exatamente na ordem em que estiverem dispostas. Quando o servidor recebe através da interface de entrada um pacote será realizada uma comparação com as regras existente no *ipfw* para saber se este terá ou não permissão para passar pelo *firewall*.

Para que seja possível realizar as configurações das regras através do *ipfw* é preciso que se tenha o conhecimento da estrutura de suas regras e sua forma de

trabalhar. Para isso é possível observar a Figura 33 que exemplifica graficamente uma regra.

Figura 33 – Regra do *ipfw*



Fonte: Elaborado pelo Autor

Neste exemplo é descrita uma regra com o intuito de descartar toda a comunicação realizada por qualquer pessoa para o IP 192.168.1.11 a fim de usar a porta 22. Para maior entendimento é possível ler a regra descrita da seguinte forma, “Adicionar um regra onde na posição 20000 para que qualquer pessoa que tentar se comunicar com o IP 192.168.1.1 através do protocolo tcp com o intuito de acessar a porta 22 deve ser negado”.

Como é observado na figura o comando *ipfw* é responsável por executar as funções na ferramenta, após ele é incluído o comando que ira informar o que fazer com a regra e logo após em qual posição devera inserir esta regra.

Após definir o que fazer e em que posição fazer é descrito o que será feito e neste exemplo é negar. Caso seja preciso é possível especificar um protocolo como foi usado para o tcp, sendo necessário apenas informar a origem e o destino que será afetado com esta regra. Para algumas regras é possível incluir uma opção no final como, por exemplo, qual o fluxo que será afetado, se será de entrada ou de saída.

4.3.2 Cenário 1: DoS a porta 80

O ataque DoS (*Denial of Services*) direcionado a uma porta específica tem por objetivo sobrecarregar todos os recursos de uma porta a fim de tornar indisponíveis os serviços providos através dela. Para este teste foi instalada e

configurada uma página simples através do apache2 no servidor *FreeBSD* tornando disponível assim uma página na rede. A Figura 34 mostra a explicação dada.

Figura 34 – Servidor *FreeBSD* com página disponível no Apache2

Fonte: Elaborado pelo Autor

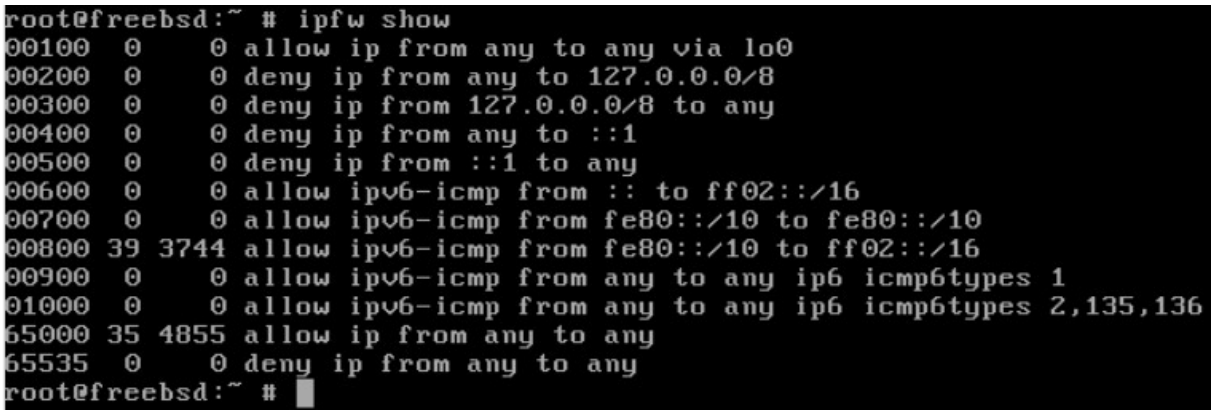
Através da Figura 34 é possível verificar a disponibilidade do servidor na rede externa, é possível visualizar também que o endereço disponível externamente é 192.168.1.37, conforme é confirmado através do comando *ipconfig* em0 realizado direto no servidor e apresentado na Figura 35.

Figura 35 – IP verificado direto no servidor *FreeBSD*

Fonte: Elaborado pelo Autor

Primeiro será simulado o ataque direcionado a esta porta sem que exista no servidor regra alguma configurada, para isso é possível visualizar se existem e caso existirem quais são as regras através do comando *ipfw show* ou *ipfw list*. A Figura 36 mostra a execução do comando *ipfw* show, verificando-se a não existência de regras além das regras padrão para a configuração do servidor.

Figura 36 – Listagem das regras no *ipfw*

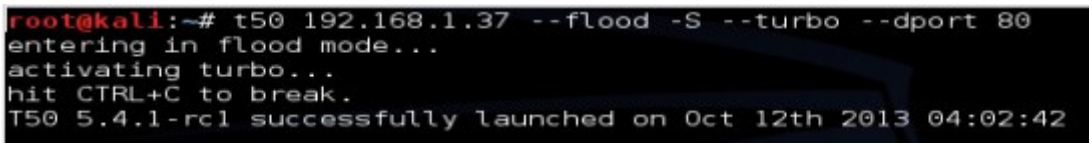


Fonte: Elaborado pelo Autor

Sabendo o IP ou endereço do servidor, é possível realizar a simulação do ataque através da máquina virtual com o sistema *Kali Linux* utilizando da ferramenta T50, ferramenta esta que foi utilizada para os mesmo testes no ambiente 1 através do comando *t50 192.168.1.39 --flood -S --turbo --dport 80*.

Este comando foi explicado no ambiente 1 de tal uma forma que se possa ter uma maior compreensão deste comando, realizando a leitura do mesmo da seguinte forma “A ferramenta t50 enviará informações ao IP alvo 192.168.1.39, através de uma inundação de TCP SYN, com grande desempenho direcionado a porta 80”. É possível visualizar este comando sendo realizado através da Figura 37.

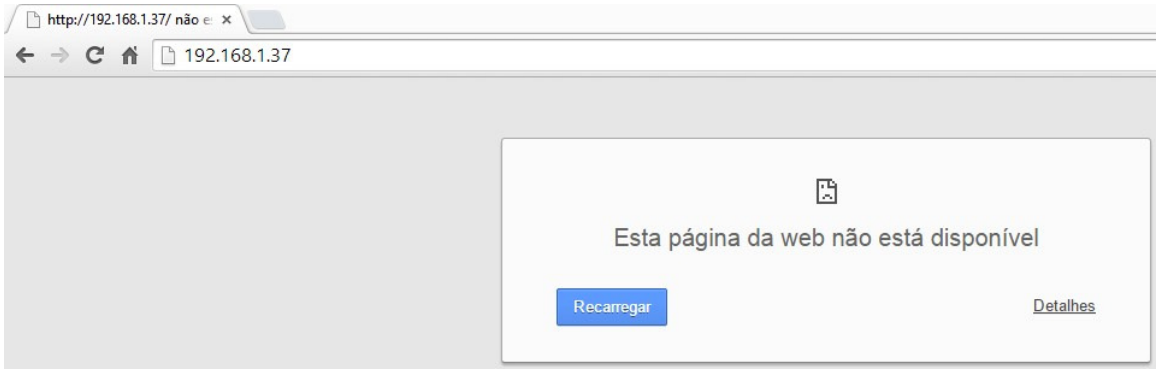
Figura 37 – Ferramenta T50 de simulação de ataque



Fonte: Elaborado pelo Autor

Após a realização deste comando foi cronometrado o tempo aproximado de 3 minutos até que o servidor *FreeBSD* estivesse totalmente inativo não sendo mais possível obter o acesso a página disponível, conforme Figura 38.

Figura 38 – Página do servidor inacessível.



Fonte: Elaborado pelo Autor

Após a realização desta simulação será a tentativa de correção da vulnerabilidade através de uma ou mais regras na ferramenta de *firewall* existente no servidor *FreeBSD*, o *ipfw*.

Para a realização da tentativa de proteção contra o ataque DOS na porta 80 serão aplicadas duas regras, *ipfw add deny all from any to any in frag* e *ipfw add deny log tcp from any to any in tcpflags syn, fin recv em0*, onde é possível ler estas regras da seguinte maneira, "Adicionar uma regra no *iptables* para negar o que entrar no servidor de qualquer IP para qualquer IP e for um pacote vazio com as tcp *flags syn, fin, recv* através da interface *em0*". A Figura 39 mostra a aplicação das regras.

Figura 39 – Regras aplicadas no *ipfw*

```
root@freebsd:~ # ipfw add deny all from any to any in frag
65100 deny ip from any to any in frag
root@freebsd:~ # ipfw add deny log tcp from any to any in tcpflags syn,fin recv
em0
65200 deny log tcp from any to any in tcpflags syn,fin recv em0
root@freebsd:~ #
```

Fonte: Elaborado pelo Autor

Após a aplicação destas regras foi realizada uma nova simulação do mesmo ataque direcionado a porta 80 do servidor, conforme é observado na Figura 40.

Figura 40 – Segunda tentativa de ataque ao servidor

```
root@kali:~# t50 192.168.1.37 --flood -S --turbo --dport 80
entering in flood mode...
activating turbo...
hit CTRL+C to break.
T50 5.4.1-rc1 successfully launched on Oct 12th 2013 04:11:41
```

Fonte: Elaborado pelo Autor

Com estas duas regras configuradas da maneira como detalhado, foi cronometrado o mesmo tempo do ambiente 1 – 30 minutos - porém com estas regras não foi possível garantir a total estabilidade do servidor ocorrendo momentos em que a página ficava inacessível e logo após estava novamente acessível.

4.3.3 Cenário 2: *Brute force* ao serviço SSH

O ataque *Brute force* (Força Bruta), consiste em realizar repetidas tentativas automatizadas de possibilidades de senhas e usuários a fim de se obter uma autenticação. Para este tipo de ataque são utilizadas, em sua maioria, grandes listas incluindo diversas possibilidades de usuários e senhas, onde o objetivo é que uma dessas seja válida, sendo assim possível obter o usuário ou a senha desejada.

Para a realização deste teste o *brute force* foi direcionado ao serviço SSH, serviço este que por padrão utiliza da porta 22 e está disponível em quase todos os servidores *FreeBSD* configurados.

A realização da primeira simulação será realizada sem que exista alguma regra configurada na ferramenta disponível no *FreeBSD* o *ipfw*, onde se pode visualizar as regras através do comando *ipfw show*, conforme é possível visualizar na Figura 41.

Figura 41 – Listagem das regras do *ipfw* no servidor *FreeBSD*.

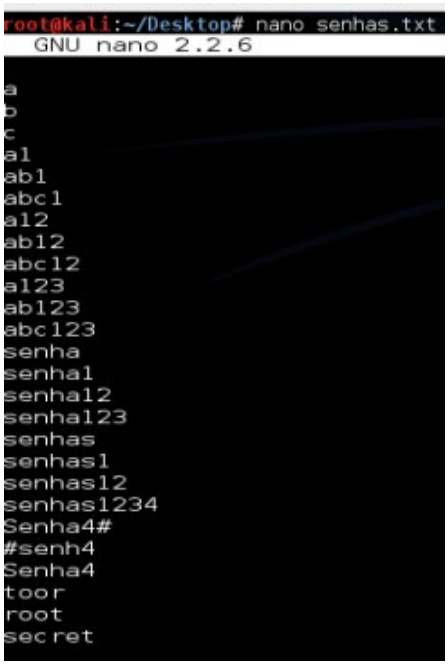
```
root@freebsd:~ # ipfw show
00100     4      192 allow ip from any to any via lo0
00200     0        0 deny ip from any to 127.0.0.0/8
00300     0        0 deny ip from 127.0.0.0/8 to any
00400     0        0 deny ip from any to ::1
00500     0        0 deny ip from ::1 to any
00600     0        0 allow ipv6-icmp from :: to ff02::/16
00700     0        0 allow ipv6-icmp from fe80::/10 to fe80::/10
00800    69     6624 allow ipv6-icmp from fe80::/10 to ff02::/16
00900     0        0 allow ipv6-icmp from any to any ip6 icmp6types 1
01000     0        0 allow ipv6-icmp from any to any ip6 icmp6types 2,135,136
65000  9037  1166799 allow ip from any to any
65535     0        0 deny ip from any to any
```

Fonte: Elaborado pelo Autor

Para realizar esta simulação através da máquina virtual *Kali Linux* será utilizada a ferramenta *hydra.* Esta ferramenta foi utilizada para a maioria dos testes de vulnerabilidades a força bruta em servidores.

Para realizar este teste será escrita a mesma lista, realizado no ambiente um com as mesmas possibilidades de senhas onde será inserida a senha válida para o usuário “cliente”. Além desta irá conter também todas as outras possibilidades de senhas incluídas no primeiro ambiente. A Figura 42 mostra a criação deste arquivo de senhas.

Figura 42 – Arquivo contendo as possibilidades de senhas



Fonte: Elaborado pelo Autor

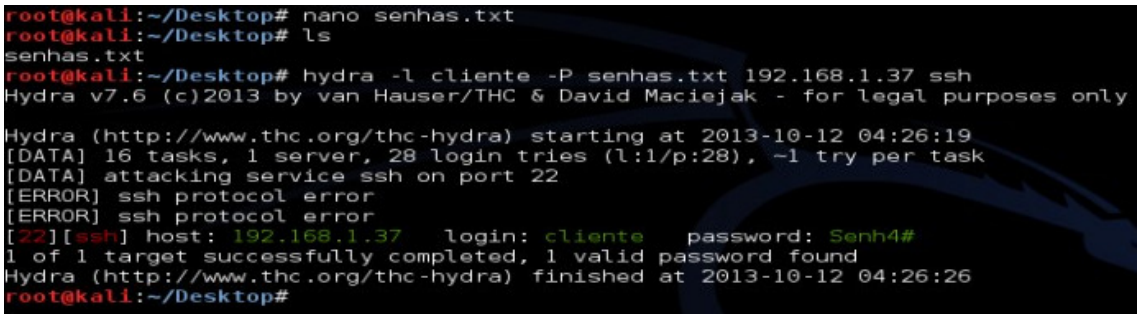
Após a criação do arquivo com todas as senhas a serem verificadas através da simulação aplica-se o mesmo comando já utilizado no ambiente anterior, *hydra* -l cliente -P senhas.txt 192.168.1.39 ssh a fim de iniciar o teste, conforme é possível observar na Figura 43.

Figura 43 – Ferramenta *hydra* para simulação de ataque

Fonte: Elaborado pelo Autor

Após a realização do comando é possível observar a resposta informada pela ferramenta *hydra*, onde se apresentam as informações obtidas com as tentativas de acessos e a confirmação da senha do usuário cliente, mostradas na Figura 44.

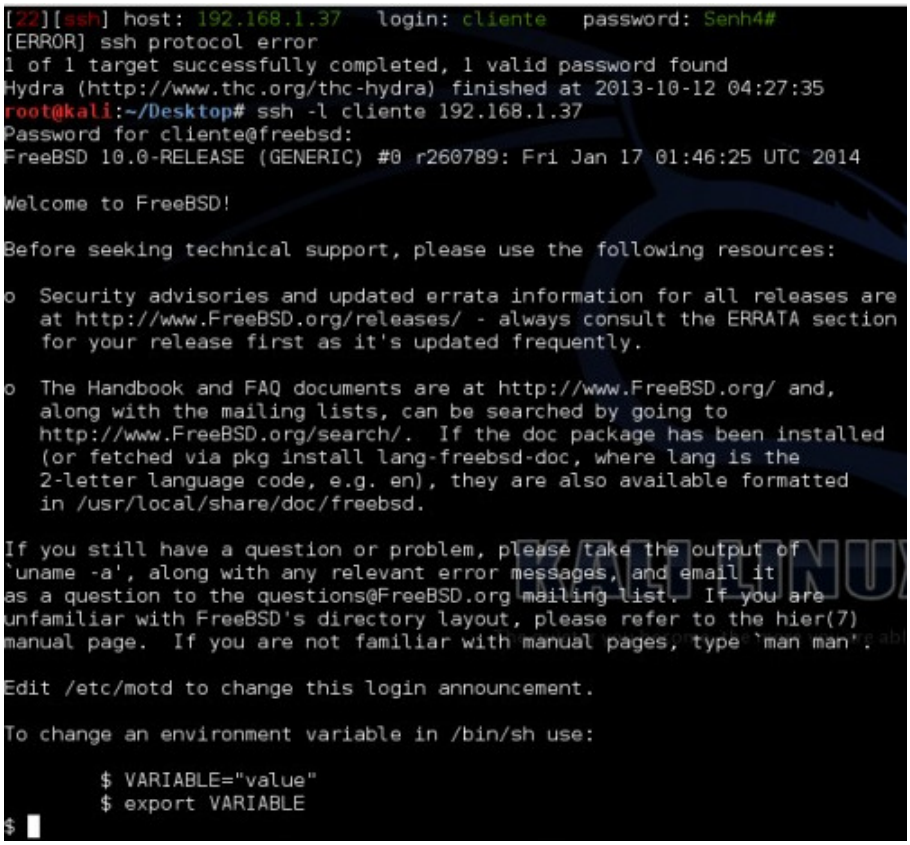
Figura 44 – Resposta da ferramenta *hydra*



Fonte: Elaborado pelo Autor

Após obter confirmação da existência do usuário e a senha para este usuário é necessário confirmar se realmente é possível acessar através do SSH com as informações obtidas conforme Figura 45.

Figura 45 – Teste de SSH após verificado senha com *Hydra*

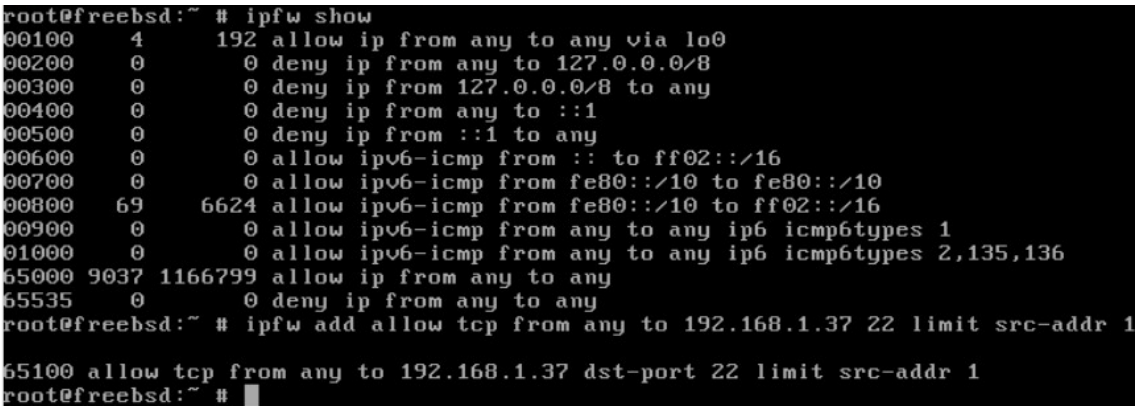


Fonte: Elaborado pelo Autor

Após verificada a vulnerabilidade é necessário a realização da aplicação de uma regra com a intenção de realizar a proteção contra o *Brute force* direcionado ao SSH, para isso será aplicada uma regra, "*ipfw add allow tcp from any to* 192.168.1.37 *22 limit src-addr* 1".

Lê-se esta regra da seguinte forma, “adicionar no *ipfw* um regra para permitir que qualquer pessoa tenha somente uma entrada na porta 22 do IP 192.168.37” A criação desta regra tem a intenção de permitir que somente uma tentativa por vez seja permitida por um IP, não permitindo múltiplas tentativas visto que esta é a forma que o *brute force* trabalha. É possível observar esta configuração sendo realizada na Figura 46.

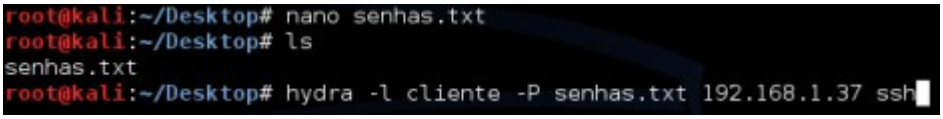
Figura 46 – Regras aplicadas no *ipfw*



Fonte: Elaborado pelo Autor

Após a aplicação destas regras foi realizada uma nova simulação do mesmo ataque direcionado ao serviço SSH do servidor, conforme se observa na Figura 47.

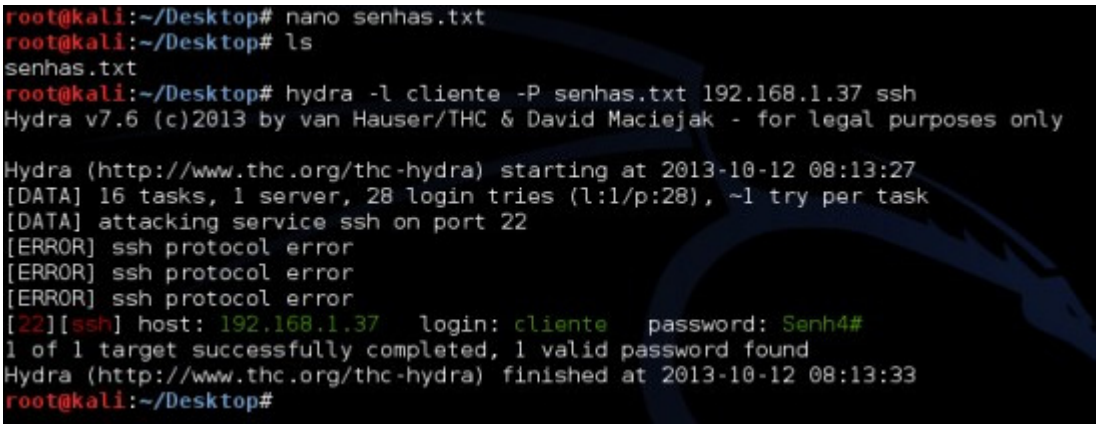
Figura 47 – Segunda tentativa de ataque ao servidor



Fonte: Elaborado pelo Autor

Após a execução do comando estando a regra configurada dentro da ferramenta *ipfw* no *FreeBSD* é possível observar através da ferramenta *hydra* que mesmo assim foi possível obter a senha do usuário cliente, conforme ilustrado através da Figura 48.

Figura 48 – Nova obtenção da senha através da ferramenta *hydra*.



Fonte: Elaborado pelo Autor

4.4 Resultados e Discussões

Qualquer comparação entre duas ferramentas distintas precisa ser imparcial e tentar reproduzir as regras com a maior igualdade possível, mesmo sabendo que é muito difícil reproduzir as mesmas regras entre duas ferramentas distintas, porém com a mesma função.

É possível notar, através dos testes e de seus resultados, em cada ambiente e cada cenário, que cada ferramenta possui suas particularidades. Também é possível notar que cada uma possui sua estrutura de regras e aplicações e que cada cenário, para a ferramenta *iptables* e a ferramenta *ipfw,* apresentou um resultado diferente do outro cenário equivalente. Para que pudesse ser mantido o maior padrão possível não foram utilizados ferramentas, aplicativos ou outro software além da aplicação da execução das regras.

Ao realizar o primeiro cenário e comparar os resultados do ambiente 1 com o ambiente 2 notou-se que na ferramenta *iptables* foi realizada uma configuração que conseguiu quase que totalmente mitigar o ataque permanecendo apenas uma lentidão ao acesso à página, enquanto no segundo ambiente após a realização da aplicação das regras além da lentidão ocorreu a inacessibilidade da página, porém a mesma, logo após, já estava disponível. A estrutura de regras na ferramenta *ipfw* permitiu uma maior facilidade para compreensão da estrutura e elaboração da regra na ferramenta *ipfw*, enquanto a estrutura do *iptables* apesar de equivalente fez com

que fosse necessário um maior estudo sobre as possibilidades e aplicações disponíveis.

O teste visava apenas a configuração de regras diretamente na ferramenta de *firewall*, não realizando configurações disponíveis ao associar uma ferramenta ou aplicação que possa trabalhar em conjunto a *firewall*, pois estas permitiriam uma infinidade de regras e configurações a fim de garantir uma maior segurança a rede.

A proteção ao ataque DoS não deve ser realizada somente através desta regra aplicada nos testes, podendo ser amplamente estudada e desenvolvida através de outras associações de regras e aplicações equivalentes para as duas ferramentas de *firewall*.

Para o segundo cenário foi possível constatar uma grande dificuldade de encontrar regras equivalentes entre o *ipfw* e o *iptables*. Como é possível notar, a regra aplicada na ferramenta *ipfw* demonstrou certa dificuldade de prover a proteção da mesma forma que o *iptables*. Esta regra especificamente na ferramenta *iptables* permitiu o uso da função *recent* que consegue criar uma lista onde é possível gravar e checar quais foram os endereços que fizeram tentativas recentes de acesso permitindo assim a criação de uma regra que cheque estas informações.

Da mesma forma que é realizada a função *recent* no *iptables* é possível realizar na ferramenta *ipfw,* porém para que seja possível realizar desta forma é necessária a utilização de aplicações que trabalham em conjunto com o *ipfw* como a ferramenta “SSHGUARD”, ou outras que podem auxiliar na mitigação deste problema.

Com estas informações foi elaborada a Tabela 1 contendo a descrição esquemática dos ambientes e cenários.

Tabela 1 – Descrição esquemática de ambientes e cenários.

Ambiente 1 (*iptables)*		**Ambiente 2** *(ipfw)*	
Cenário 1 *DoS* à porta 80	**Cenário 2** *Brute Force* ao serviço SSH	**Cenário 1** *DoS* à porta 80	**Cenário 2** *Brute Force* ao Serviço SSH
Sem nenhuma regra configurada	Sem nenhuma regra configurada	Sem nenhuma regra configurada	Sem nenhuma regra configurada
Regra 1a e 2a **1a:** *iptables -A FORWARD -p tcp --syn -m limit --limit 10/s -j ACCEPT* **2a:** *iptables -A FORWARD -p tcp --syn -j DROP*	Regra 1b, 2b e 3b **1b:** *iptables A INPUT -m state --state ESTABLISHED -j ACCEPT* **2b:** *iptables -A INPUT -p tcp --dport ssh -m recent --update --seconds 120 -j DROP* **3b:** *iptables - A INPUT -p tcp --dport ssh --tcp-flags syn,ack,rst syn -m recent --set -j ACCEPT*	Regra 1c e 2c **1c:** *ipfw add deny all from any to any in frag* **2c:** *ipfw add deny log tcp from any to any in tcpflags syn, fin recv em0*	Regra 1d **1d:** *ipfw add allow tcp from any to 192.168.1.37 22 limit src-addr 1*

Fonte: Elaborado pelo Autor

5. CONCLUSÕES

Retomando os resultados obtidos durante o teste e apresentados na Tabela 2, para melhor compreensão das questões relacionadas as conclusões do trabalho. Vale lembrar que os resultados foram apresentados da seguinte forma: Para cada ferramenta, apresenta-se primeiro o resultado sem aplicações das regras e em segundo o resultado após aplicações das regras.

Tabela 2 – Descrição esquemática dos resultados nos ambientes e cenários.

Resultados			
Ambiente 1 *(iptables)*		**Ambiente 2** *(ipfw)*	
Cenário 1 *DoS* à porta 80	**Cenário 2** *Brute Force* ao serviço SSH	**Cenário 1** *DoS* à porta 80	**Cenário 2** *Brute Force* ao Serviço SSH
Sem regras Obteve sucesso na simulação do ataque	Sem regras Obteve sucesso na simulação do ataque	Sem regras Obteve sucesso na simulação do ataque	Sem regras Obteve sucesso na simulação do ataque
Regras aplicadas Reteve o ataque sem problemas	Regras aplicadas Reteve o ataque sem problemas	Regras aplicadas Reteve o ataque com instabilidades na página *online* disponível no servidor.	Regras aplicadas Não reteve o ataque, faltou um aplicativo associado para concluir o teste com sucesso.

Fonte: Elaborado pelo Autor

Os resultados mostram que as ferramentas tem desempenho semelhantes no ambiente 1, cenários 1 e 2. Os resultados do ambiente 2 cenário 1 mostram pouquíssimas diferenças entre as duas ferramentas. A instabilidade apresentada no servidor com o firewall iptables deve ser mais explorada para se concluir alguma coisa relacionada ao tipo de aplicação (no caso a página *online*). Os resultados obtidos no ambiente 2, cenário 2 são inconclusivos, pois sem um aplicativo associado não é possível concluir os testes. A percepção da necessidade deste aplicativo surgiu durante a realização do experimento neste ambiente e cenário.

Com os resultados obtidos, verificou-se que a hipótese d, apresentada no capítulo que trata da introdução deste trabalho foi confirmada.

Além disso, a comparação dos resultados dos testes de vulnerabilidade realizados neste trabalho, nos dois servidores tentando manter o mesmo padrão e o máximo de igualdade, permitiu concluir que as duas ferramentas de *firewall* são eficientes e podem prover uma segurança adequada, quando configuradas corretamente e quando se possui um amplo conhecimento em relação a qual delas será usada como ferramenta de trabalho. Esta conclusão reforça uma das hipóteses apresentadas no Capitulo 1 deste trabalho, a saber: a comparação entre as ferramentas pode ser inconclusiva ou parcialmente inconclusiva.

É necessário um estudo mais detalhado sobre a real necessidade de uso de qualquer uma delas. Vale lembrar que se recomenda verificar com qual ferramenta o responsável pela configuração e manutenção das regras tem maior afinidade. Sabe-se, que mesmo assim, não se consegue 100% de segurança (STALLINGS, 2005).

Recomenda-se, como um estudo futuro, a associação de uma ou mais ferramentas, software ou aplicações trabalhando em conjunto ao *firewall* escolhido, auxiliando no monitoramento, manutenção, proteção e segurança da estrutura e da rede, para verificar se o seu desempenho é mais eficiente.

Este trabalho reforça que sempre devem ser realizadas simulações de ataques a vulnerabilidades para que se possa checar se a regra aplicada ao *firewall*, configuração, aplicações, software ou equipamento estão devidamente configurados e protegidos, garantindo assim maior certeza de que a segurança estará provida.

Maiores estudos sobre as vulnerabilidades podem ser feitos como sugestão para trabalhos futuros, como por exemplo teste para a porta 21 (FTP). Justifica-se esta sugestão visando a obtenção de um padrão mais adequado na configuração de ferramentas quando o administrador executa configurações por diversas vezes seguidas em redes distintas.

REFERÊNCIAS

CENTRO DE ESTUDOS SOBRE AS TECNOLOGIAS DA INFORMAÇÃO E DA COMUNICAÇÃO (CETIC).
Disponível em: <http://www.cetic.br/usuarios/tic/2012/ apresentacao-tic-domicilios-2012.pdf> Acesso em: 17/04/2014

CENTRO DE ESTUDOS, RESPOSTA E TRATAMENTO DE INCIDENTES DE SEGURANÇA NO BRASIL (CERT**). Cartilha de Segurança para Internet**. versão 4.0. São Paulo: Comitê Gestor da Internet no Brasil, 2012. Disponível em: < http://cartilha.cert.br/livro/cartilha-seguranca-Internet.pdf Acesso em: 21/09/2014.

COMER, Douglas E. **Interligação de redes com TCP/IP**: Volume 1 Princípios, protocolos e arquitetura. Rio De Janeiro: Campus, 2006.

COSTA, Marcelo A. S. L. **Computação Forense**: A análise forense no contexto da resposta a incidentes computacionais. Campinas: Milennium, 2011.

DANTAS, Marcus Leal. **Segurança da informação**: uma abordagem focada em gestão de riscos. Olinda: Livro Rápido, 2011.

D'OLIVEIRA NETO, Urubatan. **Dominando Linux *Firewall Iptables***. Rio de Janeiro: Moderna, 2004.

EDWARDS, James; BRAMANTE, Richard. **Networking Self-Teaching Guide**: OSI, TCP/IP,LANs, MANs, WANs, Implementation, Management And Maintenance. Indiana, Wiley, 2009.

FONTES, Edison Luiz Gonçalves. **Segurança da Informação**: O Usuário Faz a Diferença. Rio de Janeiro: Saraiva, 2006.

FONTES, Edison Luiz Gonçalves. **Praticando a Segurança da Informação**. Rio de Janeiro: Brasport, 2008.

KUROSE, J. F.; Ross, K. W. **Redes de Computadores e a Internet**: uma abordagem Top Down. São Paulo: Pearson, 2010.

MITNICK, Kevin D.; SIMON, William L. A. **A arte de enganar**: Ataques de hackers. São Paulo: Pearson, 2003.

NBR ISO/IEC 27002:2013. **Tecnologia da informação** — Técnicas de segurança — Código de prática para controles de segurança da informação. 08/11/2013

STALLINGS, William. **Redes e Sistemas de Comunicação de Dados**: Teoria e aplicações Corporativas. São Paulo, 2005.

SÊMOLA, Marcos. **Gestão da segurança da informação: Uma visão executiva.** 8° Ed. Rio de Janeiro, Editora Campus, 2003.

TANENBAUM, Andrew. **Redes de Computadores**. Rio de Janeiro: Campus, 2003.

ULBRICH, Henrique; DELLA VALLE, James. **Universidade H4CK3R**: Desvende o submundo hacker. São Paulo: Digerati, 2011.

WENDT, Emerson. **Crimes Cibernéticos**: ameaças e procedimentos de investigação. Rio de Janeiro: Brasport, 2013.