FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Segurança da Informação

Nathalia Peres

IPFW e SNORT:

Experimento aplicado a ataques de força bruta em servidores SSH.

Americana, SP

2017

FACULDADE DE TECNOLOGIA DE AMERICANA

Curso Superior de Tecnologia em Segurança da Informação

Nathalia Peres

IPFW e SNORT:

Experimento aplicado a ataques de força bruta em servidores SSH.

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Segurança da Informação, sob a orientação do (a) Prof.(a) Esp. Daniele Junqueira Frosoni.

Área de concentração: Segurança da Informação.

Americana, SP

2017

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana - CEETEPS
Dados Internacionais de Catalogação-na-fonte

V799e VIRE, Nathalia Peres

Experimento aplicado a ataques de força bruta em servidores SSH ./ Nathalia Peres Vire. – Americana: 2017.

90f.

Monografia (Curso de Tecnologia em Segurança da Informação) - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Profa. Esp. Daniele Junqueira Frosoni

1. Redes de computadores I. FROSONI, Daniele Junqueira II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.519

Nathalia Peres

IPFW e SNORT:
Experimento aplicado a ataques de força bruta em servidores SSH.

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Segurança da Informação, sob a orientação do (a) Prof.(a) Esp. Daniele Junqueira Frosoni.

Área de concentração: Segurança da Informação.

Americana, 30 de junho de 2017.

Banca Examinadora:

Daniele Junqueira Frosoni (Presidente)
Especialista
Faculdade de Tecnologia de Americana

Rodrigo Nogueira Tofani (Convidado 1)
Especialista
Faculdade de Tecnologia de Americana

Jeferson Wilian de Godoy Stenico (Convidado 2)
Doutor
Faculdade de Tecnologia de Americana

AGRADECIMENTOS

À minha mãe, pelo amor, incentivo e apoio incondicional.
Aos meus amigos que contribuíram direta e indiretamente para que eu continuasse.
Aos meus colegas de trabalho, que por muitas vezes me ampararam nas minhas crises de estresse e ausência na empresa para finalizar este trabalho.
À minha orientadora pela paciência.
Por último e mais importante, a Deus por ter me dado a benção de possuir todas essas pessoas em minha vida.

DEDICATÓRIA

Dedico este trabalho ao meu falecido avô Augusto Santini.

“Seja você quem for, seja qualquer posição que você tenha na vida, de um nível altíssimo ao mais baixo, tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá. De alguma maneira você chega lá.”

-Ayrton Senna da Silva.

RESUMO

Este trabalho tem por objetivo apresentar um experimento aplicado a ataques de força bruta em servidores SSH (*Secure Shell*). Para isso, foi realizada a integração de duas ferramentas de segurança da informação, um firewall e um IDS (*Intrusion Detection Systems*). Utilizado como primeira linha de defesa, o firewall tem por objetivo bloquear conexões não autorizadas enquanto o IDS desempenha a função de procurar comportamentos de conexões consideradas suspeitas e que podem comprometer a segurança e disponibilidade de um sistema dentro de uma rede. Quando empregadas em conjunto, essas ferramentas visam intensificar a segurança nos ambientes onde são aplicadas, podendo identificar e bloquear um ataque de força bruta em servidores SSH.

Palavras Chave: Redes de computadores**,** SSH, IDS.

ABSTRACT

This paper aims to present an experiment applied to brute-force attacks on SSH servers (Secure Shell). For that, it was used an integration between two information security tools: a Firewall and an IDS (Intrusion Detection Systems). Used as the first line of defense, the firewall acts in the network blocking unauthorized connections while the IDS looks for behaviors considered suspicious which can compromise the security and availability of a system. When used together, these tools aim to further enhance security in the environments where they are applied and can also identify and block a brute-force attack on SSH servers.

Keywords: Computer Network, SSH, IDS.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

VPN	Virtual Private Network
SSH	Secure Shell
RDP	Remote Desktop Protocol
IDS	Intrusion Detection System
OSI	Open System Interconnection
FTP	File Transfer Protocol
TCP	Transmission Control Protocol
IP	Internet Protocol
UDP	User Datagram Protocol
MPEG	Moving Picture Expert Group
GIF	Graphics Interchange Format
JPEG	Joint Photographics Experts Group
MP3	MP3 é uma abreviação de MPEG Layer 3
IANA	Internet Assigned Numbers Authority
DDOS	Distributed Denial of Service
IPS	Intrusion Prevention System

1 INTRODUÇÃO

Com o uso maciço e crescente da Internet para comunicação em inúmeros segmentos corporativos podendo englobar processos, departamentos, serviços, projetos, equipamentos e filiais que estão incessantemente compartilhando e acessando informações, tanto interna quanto externamente, vem aumentando drasticamente a necessidade de proteção e utilização de ferramentas de segurança física e lógica em ambientes empresariais.

Com o objetivo de centralizar informações, mas permitindo a mobilidade de funcionários, alta disponibilidade de serviços, redundância de dados e comunicação com diversas outras filiais, uma função muito utilizada por empresas é o acesso remoto a servidores e computadores localizados em ambientes geograficamente separados. Para realizar o acesso remoto, o SSH tem se tornado um serviço muito utilizado para conexões remotas a dispositivos em razão da maior segurança, facilidade de uso, suporte a diversos sistemas operacionais e possibilidade de diferentes formas de autenticação.

Entretanto, com tantas mudanças praticamente diárias na área de tecnologia, é fato que as empresas nem sempre estão preparadas para lidar com a quantidade de vulnerabilidades que as rodeiam, seja pela falta de profissionais qualificados para configuração e manutenção de sistemas de segurança, ausência de políticas internas de segurança, etc. A incidência de ataques cibernéticos tem se intensificado, visando principalmente o roubo de informações por meio de ataques de força bruta, exploração de vulnerabilidades, interceptação de tráfego e falsificação de e-mail. Portanto, é de extrema importância a empresa ser capaz de proteger e garantir o sigilo de seu ativo mais importante: A informação.

O **objetivo geral** deste trabalho é apresentar um estudo de caso integrando duas ferramentas de segurança da informação, um firewall e IDS, com o objetivo de agregar maior segurança dentro de uma rede. Utilizado como primeira linha de defesa, o firewall atua na rede bloqueando conexões não autorizadas. Por outro lado, o IDS desempenha a função de procurar comportamentos considerados suspeitos dentro de uma rede. O **objetivo específico** deste trabalho é apresentar

esses dois sistemas de segurança trabalhando em conjunto protegendo uma rede de um ataque de força bruta em um protocolo considerado seguro para acesso remoto: o SSH.

A **estrutura** deste trabalho está organizada da seguinte forma: O segundo capítulo possui conceitos importantes de redes de computadores que serão abordados posteriormente, o terceiro contém os princípios de segurança da informação e informações sobre as ferramentas utilizadas, o quarto expõe a criação de um ambiente de teste e configurações realizadas em cada ferramenta. No quinto capítulo são apresentadas as considerações finais.

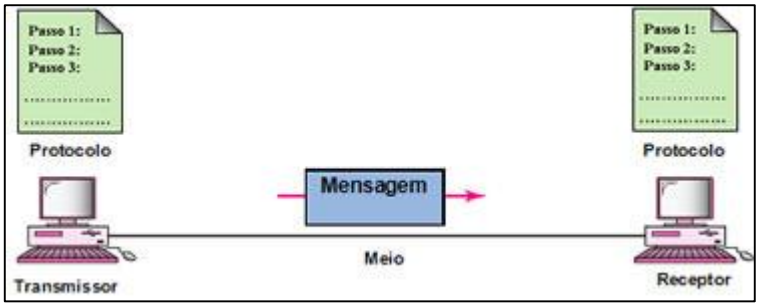
2 COMUNICAÇÃO E REDES DE COMPUTADORES

De acordo com Mendes (2015, p. 20): “As redes de computadores estabelecem a forma-padrão de interligar computadores para o compartilhamento de recursos físicos ou lógicos”. Para Tanenbaum (2003, p. 2) redes de computadores podem ser definidas como “Um conjunto de computadores autônomos interconectados por uma única tecnologia”. Além da conectividade física utilizando de meios como: cabos, fibra óptica, radiofrequência, etc., é necessário também que computadores interligados através de uma rede utilizem de alguma regra de comunicação, chamada de protocolo, para compartilhar recursos e informações.

2.1 PROTOCOLO

Um protocolo pode ser definido como uma forma de comunicação. Quando duas máquinas tentam se comunicar é necessário que uma regra entendida por ambas as partes seja definida para que a comunicação seja efetivada. Segundo Tanenbaum (2003, p. 29): “Um protocolo é um acordo entre as partes que se comunicam, estabelecendo como se dará a comunicação”. Como pode ser observado na figura 1, em uma comunicação existe um transmissor, quem emite a mensagem e um receptor, quem recebe a mensagem e entre eles, existe um meio de comunicação, podendo ser um cabo, por exemplo. Para a mensagem ser entendida por ambas as partes é necessário que um protocolo de comunicação seja estabelecido.

Figura 1 - Comunicação



Fonte: Brasilescola[1] ([s.d]).

Para Kurose (2010, p. 7): “Um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como as ações realizadas na transmissão e/ou no recebimento de uma mensagem ou outro evento.” Tanto a Internet quanto uma rede de computadores fazem o uso intenso de protocolos para comunicação para a interconexão de máquinas e sistemas. Aplicações de segurança, como firewall e IDS, baseiam suas regras de liberação, bloqueio ou monitoramento através de protocolos, sendo a única diferença a camada de atuação da ferramenta, onde os protocolos são especificados.

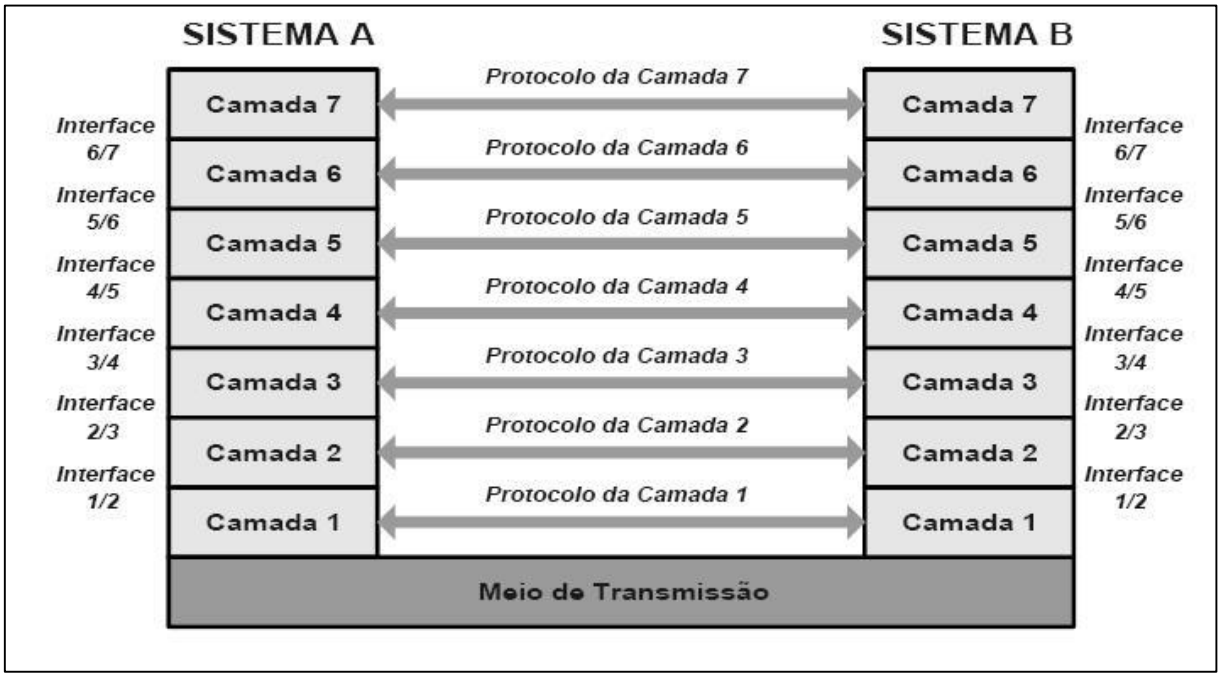
2.2 MODELO OSI E MODELO INTERNET

Para a melhor organização dos diversos serviços e protocolos que uma rede de computadores e a Internet possuem além de diminuir a complexidade de projetos aplicados a arquitetura de redes, fez-se necessário criar modelos estruturados em pilhas de camadas. O modelo OSI foi proposto em 1970 pela Organização Internacional para Padronização. Em consequência dos diversos protocolos que já existiam naquela época, fez-se necessário a criação de um modelo que desenvolvedores e fabricantes se apoiassem para criar aplicações e equipamentos, com o objetivo de garantir, independente da marca ou modelo do dispositivo, a comunicação entre sistemas computacionais. Conforme Peterson e Davie (2004, p.

[1] Disponível em: <http://brasilescola.uol.com.br/informatica/comunicacao-dados.htm>. Acesso em: 09 maio de 2017.

18) o modelo OSI: “[...] define um particionamento da funcionalidade da rede em sete camadas, onde um ou mais protocolos implementam a funcionalidade atribuída a determinada camada”. O objetivo de cada camada dentro da pilha é fornecer serviços a camadas superiores. A figura abaixo descreve o modelo de sete camadas:

Figura 2 - Modelo hierarquizado em 7 camadas



Fonte: Leofaragao[2] (2008)

A pilha TCP/IP segue o mesmo conceito do modelo OSI. Entretanto, nessa arquitetura a divisão é feita em quatro camadas: Aplicação, transporte, Internet e enlace. É possível observar uma comparação entre os dois modelos, OSI e Internet, na tabela 1:

[2] Disponível em:< https://leofaragao.wordpress.com/2008/09/02/um-pouco-mais-sobre-protocolos-hierarquia-e-camadas/>. Acesso em: 28 maio de 2017.

Tabela 1 - Comparação entre os modelos OSI e Internet

MODELO OSI	MODELO INTERNET
Aplicação	**Aplicação**
Apresentação	
Sessão	**Transporte**
Transporte	
Rede	**Internet**
Enlace	**Enlace**
Física	

Fonte: Próprio autor

As camadas podem ser descritas da seguinte forma:

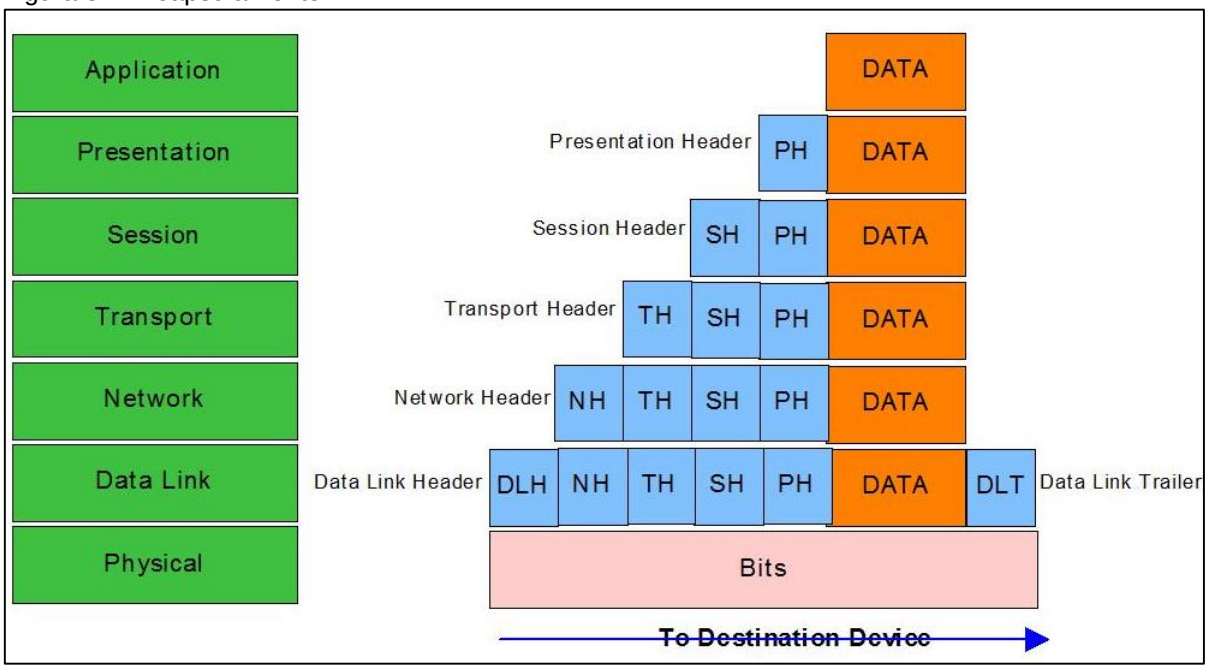
- Aplicação: Camada superior dos modelos OSI e TCP/IP, ela “oferece a interface com o usuário para a comunicação.” (BARRETT;KING, 2010, p. 39). Sendo a camada mais próxima do usuário, nela são oferecidos serviços para aplicações como transferência de arquivos, envio e recebimento de e-mail, acesso às páginas na Internet, resolução de nomes, acesso remoto, etc.
- Apresentação: “A camada de apresentação formata os dados para troca entre a camada de aplicação e a camada de sessão” (BARRETT;KING, 2010 p. 39). Ela é responsável pela entrega e formatação da informação para a camada de aplicação para posterior processamento ou apresentação. Formatos comumente utilizados são: MPEG, GIF, JPEG, MP3.
- Sessão: Esta camada provê serviços de gerência de conexão entre aplicações, fornecendo ainda mecanismos de segurança, autenticação e sincronismo entre as partes. “A camada de sessão gerencia a comunicação

entre as aplicações após uma conexão ter sido feita. Ela estabelece a sessão, gerencia as trocas de informações e depois desfaz, quando a sessão termina.” (BARRETT;KING, 2010, p. 39).

- Transporte: A camada de transporte “desempenha o papel fundamental de fornecer serviços de comunicação diretamente aos processos de aplicação que rodam em hospedeiros diferentes.” (STALLINGS, 2010, p. 140). Para ela, são atribuídas tarefas como a troca de dados, garantia de entrega da informação ao seu destino sem erros e na sequência correta, controle de fluxo e multiplexação de mensagens. Os dois protocolos utilizados nesta camada são TCP e UDP.
- Rede: “A camada de rede gerencia o roteamento dos pacotes que devem ser encaminhados para diferentes redes.” (BARRETT;KING, 2010, p. 38). O roteamento realizado nesta camada é baseado na escolha da melhor rota, dado que, o pacote enviado pode passar por diversos equipamentos intermediários durante o percurso até seu destino.
- Enlace: “A camada de enlace oferece controle de fluxo, controle de erro e sincronismo para a camada física.” (BARRETT;KING, 2010 p. 38). Utiliza de métodos de enquadramento, conversão e retransmissão de dados.
- Física: “A camada física trata de comunicações mecânicas e elétricas.” (BARRETT; KING, 2010, p. 37). Aqui, os dados são transformados em bits para envio, portanto, as comunicações através desta camada se dão por formas mecânicas e elétricas, por meio de cabos, sinais elétricos, fibra óptica e ondas de radiofrequência.

Quando uma informação é enviada por um usuário, como um e-mail, por exemplo, o pacote em questão recebe cabeçalhos de cada camada que passa, em um processo conhecido como encapsulamento conforme ilustrado na figura 3. No destino, é feita a operação inversa, onde cada camada lê as suas respectivas informações conforme o pacote é repassado para camadas superiores.

Figura 3 - Encapsulamento



Fonte: Routemybrain[3] ([s.d]).

2.3 PROTOCOLO SSH

De acordo com Peterson e Davie (2004, p. 443): “O *Secure Shell* (SSH) oferece um serviço de login remoto e tem como finalidade substituir os programas Telnet e rlogin menos seguros [...]”. Ele é um protocolo de comunicação e conexão remota que atua na camada de aplicação adicionando uma camada de segurança se comparado a outros protocolos semelhantes, protegendo a comunicação com instrumentos que envolvem principalmente a criptografia de ponta a ponta em uma conexão. Utilizando de uma arquitetura cliente/servidor, os serviços e recursos para acesso se encontram em computadores designados como servidores e os requerentes dos serviços, são designados como clientes. Quando um cliente inicia uma conexão, é estabelecido um canal seguro de comunicação, posteriormente o servidor solicita uma senha para acesso. “Uma vez autenticado, cliente e servidor estabelecem uma

[3] Disponível em: https://www.routemybrain.com/intro-data-encapsulation-introduction-to-osi-layer-model-the-internetworking-part5/. Acesso em: 28 maio de 2017.

chave de sessão que eles usarão para criptografar quaisquer dados enviados pelo canal.” (PETERSON;DAVIE, 2004, p. 444).

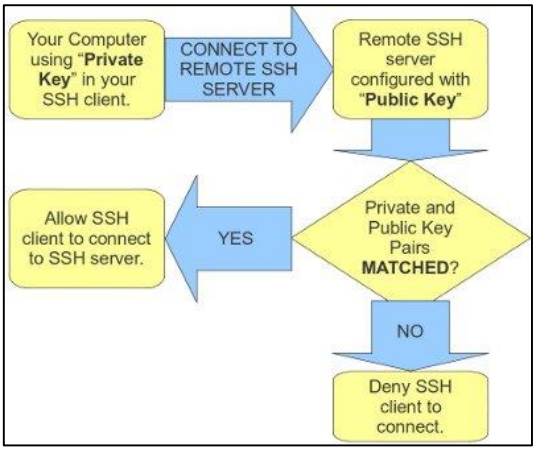
2.3.1 VERSÕES

Existem duas versões do protocolo SSH, a versão 1 e 2. A primeira versão, desenvolvida em 1995 possui muitas vulnerabilidades, tais como a interceptação da chave de acesso para decodificação das informações, portanto, não é mais utilizada. Em decorrência disso, foi desenvolvida a versão 2, que incorpora novos algoritmos e que corrige as falhas de segurança encontradas na versão 1.

2.3.2 FUNCIONAMENTO

Existem três elementos com qual o protocolo SSH trabalha: autenticação de usuários, criptografia dos dados e integridade da informação. Uma das funções da autenticação é exigir uma identidade do usuário, por meio de senhas ou chaves assimétricas. Já a criptografia utiliza de métodos e funções para “embaralhar” os dados e torná-los incompreensíveis para terceiros. A integridade, por sua vez, faz com que as informações transmitidas cheguem sem qualquer tipo de alteração ao destino. Na figura 4, é possível observar um fluxograma ilustrando o acesso via SSH e a troca de chaves entre ambas as partes.

Figura 4 - Protocolo SSH



Fonte: Stackexchange[4] (2013)

Para máquinas baseadas em Unix, existem utilitários que provêm a utilização do SSH. O OpenSSH "[...] admite um comando *ssh-keygen,* que pode ser usado para criar pares de chaves pública/privada. Essas chaves são então armazenadas em diversos arquivos no diretório *.ssh*, dentro do diretório home do usuário." (PETERSON;DAVIE, 2004, p. 444).

As configurações do SSH em um computador cliente e servidor utilizando o sistema operacional FreeBSD e Debian estão disponíveis no apêndice I.

2.4 TCP

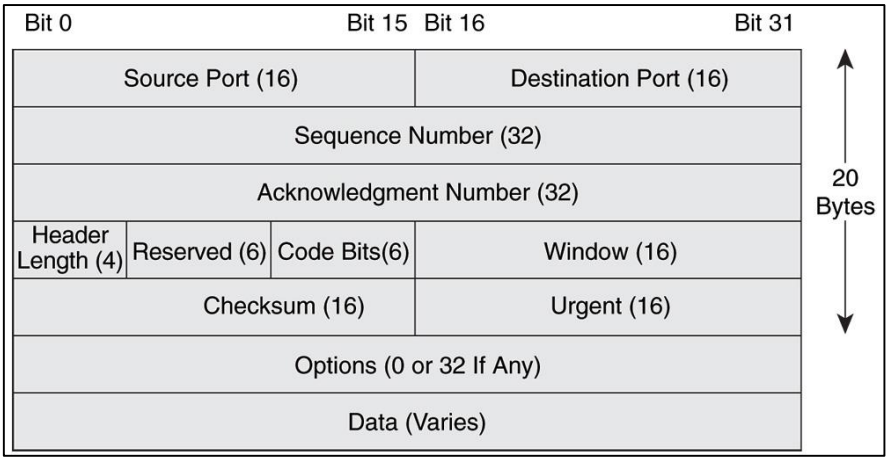
Sendo o protocolo mais utilizado na camada de transporte, o TCP quando aplicado juntamente com o protocolo IP, disponível na camada de rede, formam o sistema de envio de pacotes mais utilizado dentro de uma rede de computadores, o TCP/IP.

O TCP possui um sistema de numeração para garantir que os segmentos enviados da camada de transporte vão chegar ao destino na ordem correta. Se por ventura o destinatário não receber algum dos pacotes, a informação é enviada novamente. Outro ponto a se destacar é que existe também uma checagem de erros que atesta que nenhuma das informações foi corrompida durante o trajeto. Este protocolo é

[4] Disponível em:< https://security.stackexchange.com/questions/40050/what-is-the-best-practice-separate-ssh-key-per-host-and-user-vs-one-ssh-key-for>. Acesso em: 28 maio de 2017.

conhecido pela garantia de entrega agregada a ele. Caso um pacote não chegue ao seu destino o protocolo TCP se encarrega de reenviá-lo ao destinatário.

Figura 5 - Cabeçalho TCP



Fonte: Networking-forum[5] (2015)

Como mostrado na figura 5, o cabeçalho TCP possui alguns campos de informações importantes, como: Porta de origem e destino que são utilizados pela camada de aplicação para informar de qual serviço aquela mensagem tem origem e para qual porta deve ser encaminhada. O SSH, por exemplo, trabalha com o protocolo TCP atuando na porta 22.

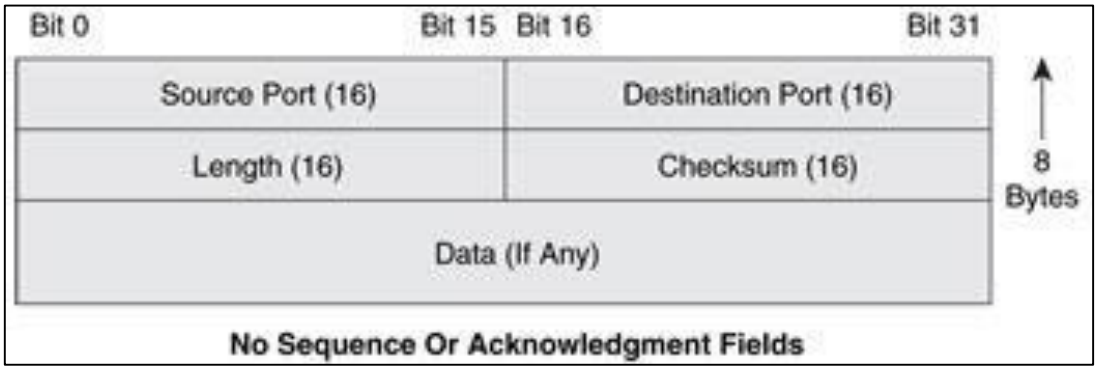
2.5 UDP

O protocolo UDP (User Datagram Protocol) trabalha de forma semelhante ao protocolo TCP. Entretanto, o protocolo UDP retira de sua operação toda a verificação de erros e checagem de recebimento que o TCP possui. Esse processo tem por objetivo agilizar o envio e recebimento de pacotes, uma vez que, todo o procedimento de verificação realizado pelo TCP, deixa a comunicação mais lenta. Quando o protocolo UDP é utilizado, ele envia sem a preocupação de que o mesmo

[5] Disponível em:< http://www.networking-forum.com/viewtopic.php?f=33&t=47667>. Acesso: 28 maio de 2017.

chegou a seu destino. Se caso houver algum erro nesta transmissão, serão enviados os pacotes programados em seguida, e os perdidos não poderão ser recuperados. Aplicações que utilizam esse protocolo possuem em sua característica a rapidez na entrega das informações, maior resistência a falhas e perda de dados na transmissão dos pacotes.

Figura 6 - Cabeçalho UDP



Fonte: Rumyittips[6] ([s.d])

2.6 PORTAS TCP E PORTAS UDP

Conforme abordado nas seções 2.4 e 2.5, protocolos da camada de transporte possuem em sua composição números de portas de origem e destino, independente de qual forma o envio é realizado. Aplicações fazem o uso constante dessas portas para especificar a qual serviço a informação enviada ou recebida pertence.

A IANA (*Internet Assigned Numbers Authority*) é o órgão responsável por pela atribuição de números na Internet, sejam eles de portas, IP, DNS, etc. No site da IANA é possível obter uma lista completa de portas TCP e UDP. Equipamentos de segurança como o firewall e IDS também fazem uso da combinação de portas para bloqueio e liberação de conexões. Com um firewall configurado na rede, é possível impedir que aplicações maliciosas façam uso de portas abertas ou inseguras no computador. É possível também realizar configurações para que determinadas

[6] Disponível em:< http://rumyittips.com/what-is-differences-between-tcp-and-udp/>. Acesso em: 28 maio de 2017.

portas permaneçam bloqueadas, impedindo a conexão de aplicativos que fazem uso destas, realizar o redirecionamento de portas evitando utilizar portas padrões de aplicativos, etc.

2.7 IPV4

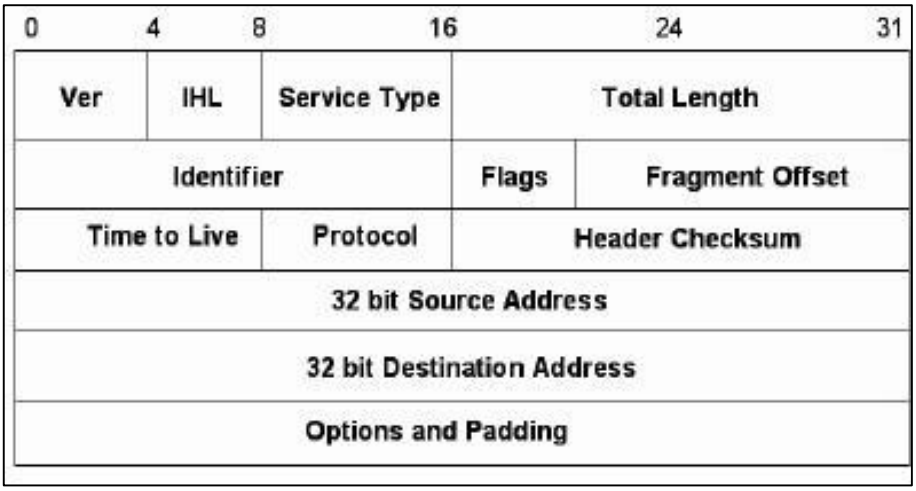
Dentro de uma rede de computadores cada dispositivo conectado possui um endereço atribuído chamado IP. Sendo o IP o principal protocolo de identificação, cabe a ele endereçar e encaminhar informações que transitam por uma rede de computadores. Os segmentos da camada de transporte são recebidos pela camada de rede onde recebem um endereço IP de origem e destino. Segundo Kurose (2010, p. 247) “Um pacote na camada de rede é denominado datagrama”. Os roteadores são os responsáveis por determinar “o caminho que um datagrama segue desde a origem até o destino”.

O IPv4 é a versão mais utilizada do protocolo IP. Ela possui 32 bits de endereçamento, portanto é possível que existam 4.294.967.296 de combinações possíveis. O endereçamento do IPv4 é dividido em quatro grupos de oito bits, chamados de octetos, sendo que nesses quatro grupos é possível encontrar números de até três algarismos cada, sendo eles de 0 até 255.

2.7.1 COMPONENTES DO CABEÇALHO IPV4

Apesar dos vários campos encontrados dentro de um datagrama, neste momento é importante focar apenas em três: Protocolo, endereço de origem da conexão e endereço de destino, que são as informações utilizadas pelas ferramentas que serão posteriormente configuradas neste trabalho.

Figura 7 - Datagrama



Fonte: Vivaolinux[7] (2012)

Conforme a figura 7, é possível observar que no campo de endereço de origem está o endereço de IP da origem do datagrama e respectivamente no campo de destino está o IP a qual o datagrama foi enviado.

[7] Disponível em:< https://www.vivaolinux.com.br/artigo/Datagramas>. Acesso em: 28 maio de 2017.

3 SEGURANÇA DA INFORMAÇÃO

A segurança da informação está diretamente relacionada com a proteção de um conjunto de informações a fim de preservar o seu valor para um usuário ou organização. Mesmo antigamente, a segurança da informação está presente na sociedade. Mesmo que não nesses termos, o conceito relacionado à proteção da informação era praticado em assuntos relacionados a guerra, diplomacia, economia etc. Com o único objetivo de garantir que a informação chegasse ao seu destino sem qualquer tipo de interceptação ou que ela não fosse lida por pessoas não autorizadas, recursos como a criptografia eram utilizados.

Dentro de uma organização, por exemplo, a proteção da informação podia ser feita apenas armazenando seus dados e informações em um ambiente restrito. Sendo assim, o acesso físico a ela só poderia ser feito por pessoas devidamente autorizadas. Entretanto, com os grandes avanços na área de tecnologia da informação, novos serviços de comunicação foram surgindo. Com a propagação do uso de dispositivos que possuem acesso à Internet, tanto dentro da organização quanto de usuários comuns, a segurança agora também abrange o acesso lógico a informação. Com quase tudo sendo feito através da Internet, tornou-se necessário o uso de ferramentas automatizadas para garantir a proteção de arquivos e dados que trafegam na rede.

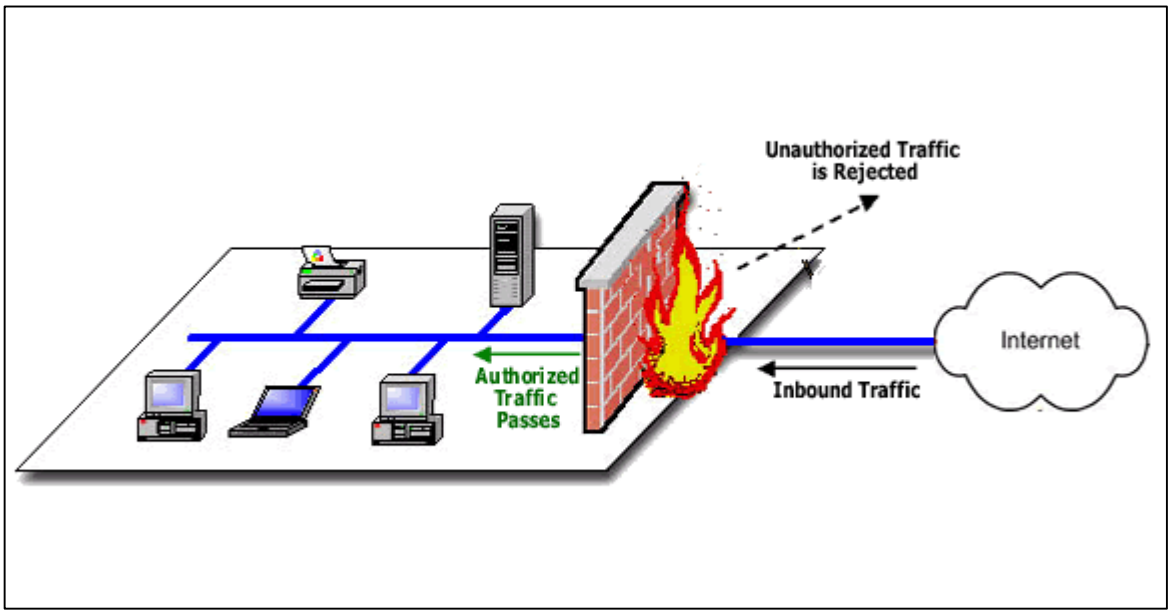
3.1 FERRAMENTAS DE APOIO A SEGURANÇA DA INFORMAÇÃO

Com a quantidade de informações que rodeiam uma rede, principalmente dentro de uma empresa, onde a vazão de dados, acesso indevido ou comprometimento de equipamentos, pode causar desde a perda de clientes até a falência da empresa, mecanismos que têm por função realizar a segurança lógica de uma rede devem ser essencialmente utilizados e configurados.

3.1.1 FIREWALL

Uma das primeiras linhas de defesa de uma rede é o firewall. Para Barrett e King (2010, p. 314): “Firewall é um componente entre computadores e redes para ajudar a eliminar o acesso indesejado ao mundo exterior”. Sendo uma ferramenta de segurança dentro de uma rede, ela pode ser encontrada em forma de software, *hardware* dedicado ou uma junção de ambos, que tem por objetivo bloquear o acesso indevido a informações provenientes da Internet ou mesmo da própria rede interna. Conforme Stallings (2008, p. 443) “Um firewall forma uma barreira através da qual o tráfego indo a cada direção precisa passar. Uma política de segurança de firewall dita qual tráfego tem autorização para passar em cada direção.”. Essa definição de Stallings pode ser observada na figura 8.

Figura 8 - Representação Firewall



Fonte: Westitsolutions[8] ([s.d])

3.1.2 TIPOS DE FIREWALL

Dependendo de fatores como estrutura da rede, quantidade de banda, dados trafegados ou necessidades específicas dos usuários, o firewall pode trabalhar de

[8] Disponível: < https://westitsolutions.com/Security.php>. Acesso em: 28 maio de 2017

diferentes formas, também podendo ser categorizado pela maneira como analisa e bloqueia o tráfego.

3.1.2.1 FILTRO DE PACOTES

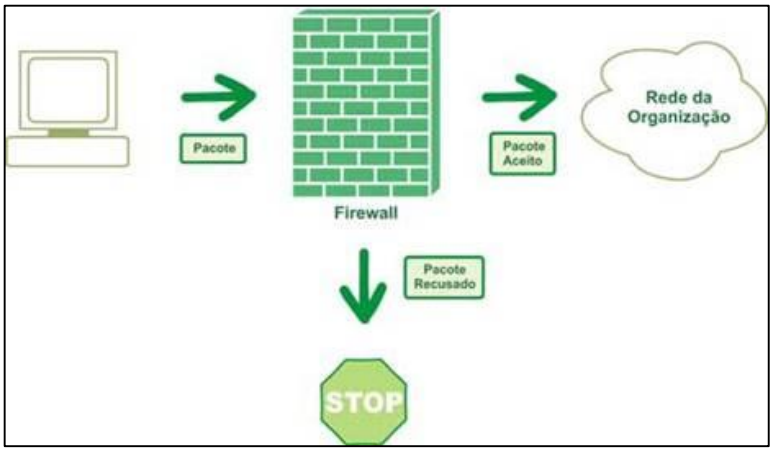
De acordo com Tanenbaum (2003, p. 826):

> “Os filtros de pacotes são baseados em tabelas configuradas pelo administrador do sistema. Essas tabelas listam as origens e os destinos aceitáveis, as origens e destinos bloqueados e as regras padrão que orientam o que deve ser feito com os pacotes recebidos de outras máquinas ou destinadas a elas.”

Para Kurose (2010, p. 536): “Um filtro de pacotes examina cada datagrama que está sozinho, determinando se o datagrama deve passar ou ficar baseado nas regras específicas do administrador”.

A filtragem de pacotes é uma das funcionalidades mais básicas que um firewall pode desempenhar. Normalmente, dispositivos que realizam roteamento possuem em sua configuração funções para realizar a filtragem de pacotes. A figura 9 ilustra o funcionamento da filtragem de pacotes.

Figura 9 - Filtro de pacotes



Fonte: Unipvirtual[9] ([s.d])

[9] Disponível em: < http://unipvirtual.com.br/material/MATERIAL_ANTIGO/redes_dados/html/mod_14.html>. Acesso em: 28 maio de 2017

Esse tipo de firewall opera na camada de rede, portanto, a inspeção na filtragem de pacotes foca na análise de parâmetros associados à interface, porta, protocolo de transporte, IP de origem e de destino do pacote.

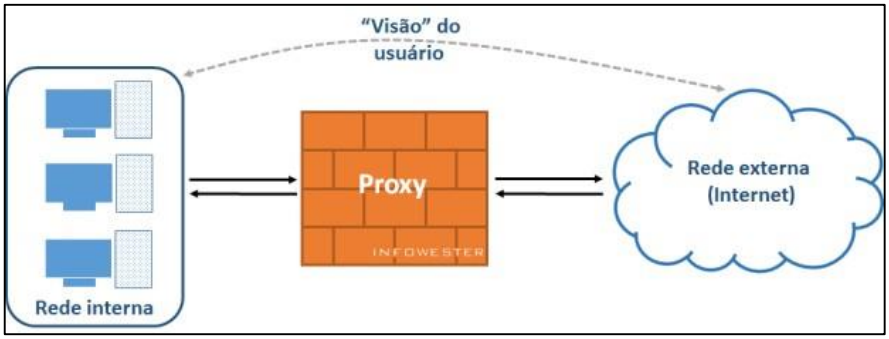
3.1.2.2 INSPEÇÃO COM ESTADO

Sendo uma evolução do filtro de pacotes, neste tipo de firewall existe uma tabela de estado juntamente com a tabela de regras configurada. O firewall trata de rastrear toda a conexão e aguarda a resposta, levando em consideração informações como porta e IP. Se as informações estiverem corretas, a entrada do pacote é autorizada, mesmo que não existam regras específicas previamente configuradas.

3.1.2.3 PROXY

Um firewall proxy pode executar diversas funcionalidades dentro de uma rede. Sendo representado na figura 10, o proxy atua como um intermediário entre hosts dentro de uma rede e a Internet, onde pode ser utilizado para melhorar o desempenho do acesso à Internet, uma vez que ele armazena cachê de conteúdos acessados, bloqueio a determinadas páginas na Internet com base em uma lista de endereços ou palavras específicas ou apenas compartilhar Internet com a rede.

Figura 10 - Proxy



Fonte: Infowester[10] ([s.d])

[10] Disponível em: < https://www.infowester.com/firewall.php>. Acesso em: 28 maio de 2017

3.1.3 IDS

Sistemas de detecção de intrusão são *softwares* que analisam comportamentos de conexões que podem comprometer a segurança e disponibilidade de um sistema dentro de uma rede. Comportamentos considerados como anormais são aqueles que visam explorar vulnerabilidades dentro de uma rede através de portas inseguras ou abertas, *logins* não autorizados, requisições de serviço em grande quantidade, etc.

Funcionando como uma segunda linha de defesa, ele trabalha monitorando e coletando registros e dados que trafegam na rede, sendo apenas acionado quando uma ameaça é detectada por meio de uma quantia considerável de informações em sua base que confirmam a maliciosidade da requisição.

Quando isso ocorre, o IDS dispõe de diversos que componentes e integrações com outros softwares e sistemas de segurança, podendo agir de forma passiva enviando avisos e relatórios a administradores de redes ou de forma ativa realizando uma ação de resposta a ameaça encontrada.

3.1.4 TIPOS DE IDS

O IDS pode trabalhar em diversas linhas de frente, o que pode depender do que ele está monitorando e como está fazendo isso.

3.1.4.1 BASEADO EM REDE

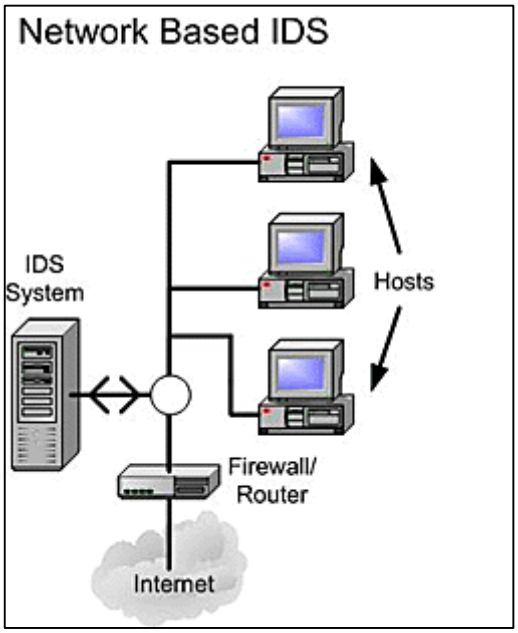
Neste tipo de monitoramento, o IDS avalia o tráfego de rede em um segmento ou dispositivo, e inspeciona a rede em busca de atividades consideradas suspeitas. Ele é capaz de ser configurado para examinar diversas características de eventos que podem ser prejudiciais a uma rede. De acordo com o blog Ostec (2015), que divulga informações relacionadas à segurança e redes de computadores, uma das características desse modelo é também:

> “[...] capaz de detectar inúmeros tipos de eventos de interesse, e geralmente é implantado em uma topologia de segurança como

fronteira entre duas redes, por onde o tráfego é afunilado. Por conta disso, em muitos casos, o próprio recurso de IDS acaba sendo integrado diretamente no firewall.

Na figura 11 é possível observar uma aplicação de um IDS baseado em rede, analisando o comportamento de conexões de todo um segmento de rede.

Figura 11 - IDS baseado em rede



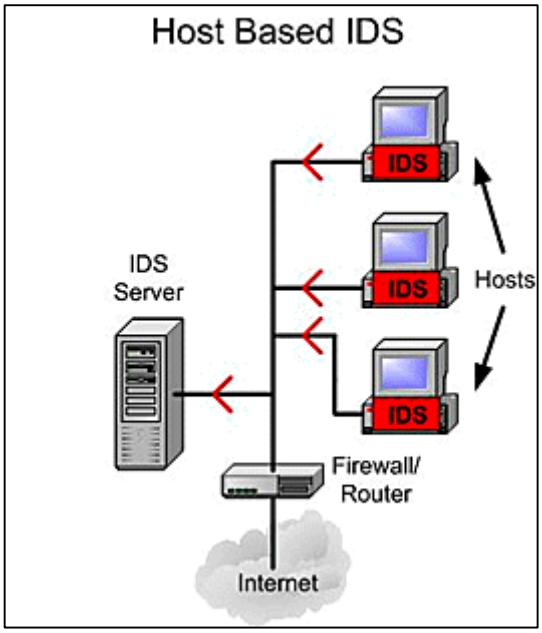
Fonte: Informit[11] (2002)

3.1.4.2 BASEADO EM HOST

Ainda conforme o blog Ostec (2016), nessa modalidade o IDS é configurado em um computador dentro da rede, sendo ele servidor ou não, onde fiscaliza eventos que acontecem nos hosts onde estão configurados. Como exemplificado na figura 12, o IDS foca no tráfego da rede para o dispositivo, os processos que estão em execução, *logs* do sistema, e o acesso e alteração em arquivos e aplicações.

[11] Disponível em: < http://www.informit.com/articles/article.aspx?p=29601>. Acesso em: 28 maio de 2017.

Figura 12 - IDS baseado em host



Fonte: Informit[12] (2002)

3.1.4.3 FORMAS DE DETECÇÃO

O IDS pode trabalhar com algumas técnicas de detecção, dependendo da forma como é configurado.

3.1.4.3.1 CONHECIMENTO

Detecções por assinatura fazem uso de um banco de dados que possuem registros de comportamentos suspeitos já conhecidos, auxiliando na identificação de tentativas de invasão. Neste tipo de estrutura, é de suma importância a constante atualização do banco de dados utilizado, uma vez que, novas vulnerabilidades e formas de ataque vão surgindo.

[12] Disponível em: < https://www.hackthis.co.uk/articles/basics-of-intrusion-detection-systems>. Acesso em: 28 maio de 2017.

3.1.4.3.2 COMPORTAMENTO

Neste tipo de busca, parte-se do princípio de que ataques são ações contrárias a que são rotineiramente executadas em uma rede. O IDS produz um perfil de atividades usualmente executadas dentro de uma rede ou host e quando dados monitorados estão fora do padrão uma ação é realizada.

3.1.4.4 FORMAS DE UTILIZAÇÃO

Quando uma ameaça é detectada, o IDS pode responder ao incidente de duas formas.

3.1.4.4.1 ATIVO

No modo ativo, um IDS pode tanto detectar comportamentos suspeitos dentro de uma rede e alertar seu administrador como também intervir na conexão bloqueando automaticamente a origem do problema.

3.1.4.4.2 PASSIVO

Ao contrário de um IDS ativo, o passivo trabalha de maneira silenciosa, apenas monitorando o tráfego e identificando potenciais ataques ou anormalidades. A partir dos *logs* gerados, ele envia um alerta ao administrador de rede sobre qual evento dentro da rede foi detectado.

3.2. ATAQUE DE FORÇA BRUTA

Conforme descrito pela Cartilha de Segurança para Internet (2016):

> “Ataques costumam ocorrer na Internet com diversos objetivos, visando diferentes alvos e usando variadas técnicas. Qualquer serviço, computador ou rede que

> seja acessível via Internet pode ser alvo de um ataque, assim como qualquer computador com acesso à Internet pode participar de um ataque.”

Para surtirem efeito, esses ataques utilizam de vulnerabilidades existentes dentro da rede, sendo elas físicas, lógicas ou humanas. Portanto, quanto mais protegida a rede estiver, menor será o impacto do ataque. A simulação proposta neste projeto executa um dos ataques mais conhecidos em segurança, que se vale de vulnerabilidades tanto humanas, quanto lógicas.

O ataque de força bruta é um termo utilizado para descrever uma técnica que visa descobrir uma senha através de incessantes tentativas de combinação, podendo até ser definido como um método de adivinhação pura. Ainda conforme cartilha publicada pelo Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil: “Um ataque de força bruta, ou brute force, consiste em adivinhar, por tentativa e erro, um nome de usuário e senha e, assim, executar processos e acessar sites, computadores e serviços em nome e com os mesmos privilégios deste usuário.” (CERT, 2016).

Softwares que executam este ataque fazem uso de dicionários, que nada mais são do que um banco de dados com palavras conhecidas e comumente utilizadas por usuários na definição de uma senha. Elas podem ser combinações como: Datas importantes, nomes de familiares, combinação de letras e números sequenciais, times de futebol, etc. Além disso, é comum encontrar computadores, servidores e serviços configurados com senhas já definidas por padrão.

Dependendo da dimensão do ataque de força bruta, o mesmo pode ser classificado como um Ataque de Negação de Serviço, onde o servidor recebe uma carga tão grande de requisições de determinado serviço, que devido ao esgotamento de recursos do sistema, o mesmo acaba se tornando instável.

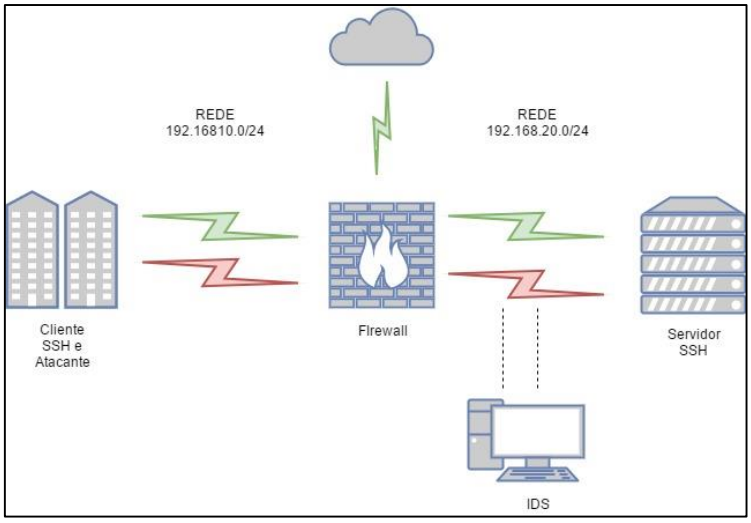
4 DESENVOLVIMENTO

Neste capítulo serão apresentadas as ferramentas, o cenário e sistemas utilizados para promover um ataque de força bruta a um servidor SSH.

4.1 CENÁRIO

O ambiente proposto neste estudo de caso contém cinco máquinas, das quais são constituídas por: Um servidor SSH utilizado como alvo do ataque, um firewall com regras pré-configuradas, um sistema de detecção de intrusão, uma máquina cliente SSH e uma máquina que realizará o ataque de força bruta. Para a devida comunicação das máquinas virtuais, as configurações de rede de cada uma delas encontram-se no apêndice A. Neste cenário, conforme mostrado na figura 13, existem duas redes: Rede externa (192.168.10.0/24) e a rede interna (192.168.20.0/24). Nesta simulação, a rede externa faz parte de um ambiente corporativo que possui acesso remoto a um servidor SSH. Já a rede interna contém o servidor SSH, o qual a rede externa acessa, um firewall e IDS.

Figura 13 - Cenário



Fonte: Próprio autor.

4.2 VIRTUALBOX

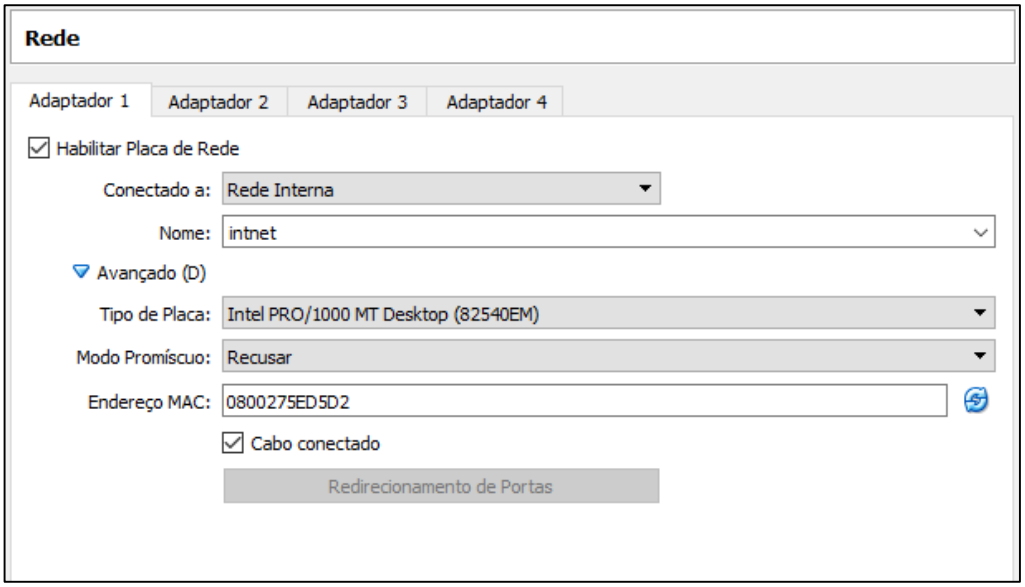
O VirtualBox permite a execução de diversos sistemas operacionais virtuais simultaneamente, utilizando como base o sistema operacional e arquitetura da própria máquina onde o software está instalado. Sendo de fácil utilização, esta ferramenta foi escolhida para criar as máquinas virtuais e executar o ambiente de teste. Atualmente ele está na versão 5.1.18.

Figura 14 - Máquinas virtuais

Fonte: Próprio autor

Com exceção da máquina Debian – Snort e FreeBSD - Firewall, todas as outras possuem adaptadores de rede iguais. A configuração desses adaptadores se faz necessária para emular uma rede dentro do sistema operacional após ele ser iniciado. As figuras abaixo ilustram os ajustes realizados para essas máquinas.

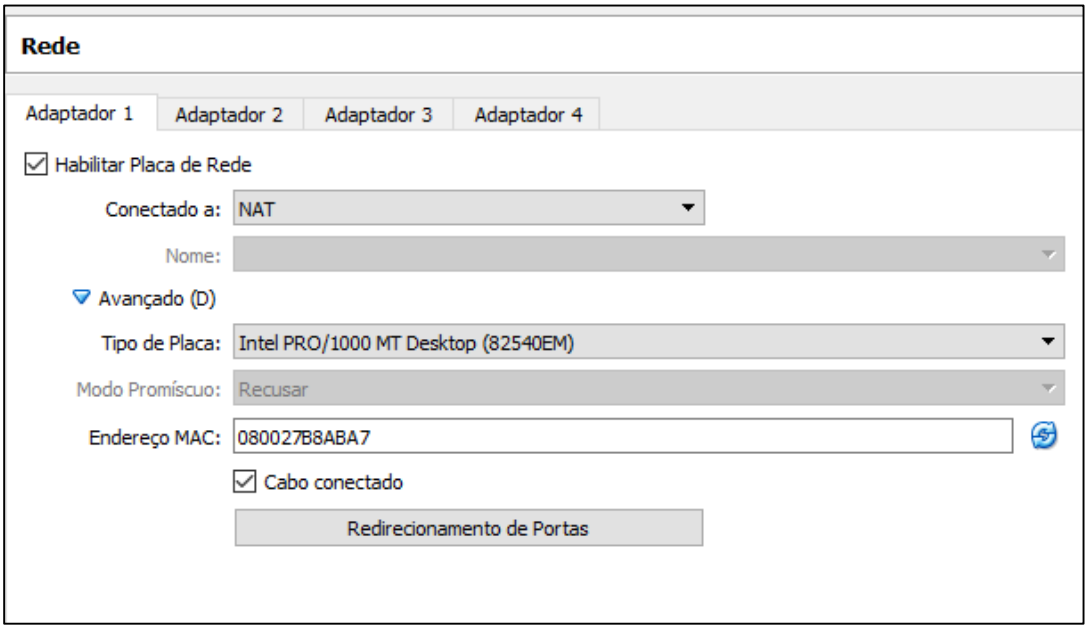
Figura 15 – Adaptador 1 das máquinas virtuais



Fonte: Próprio autor.

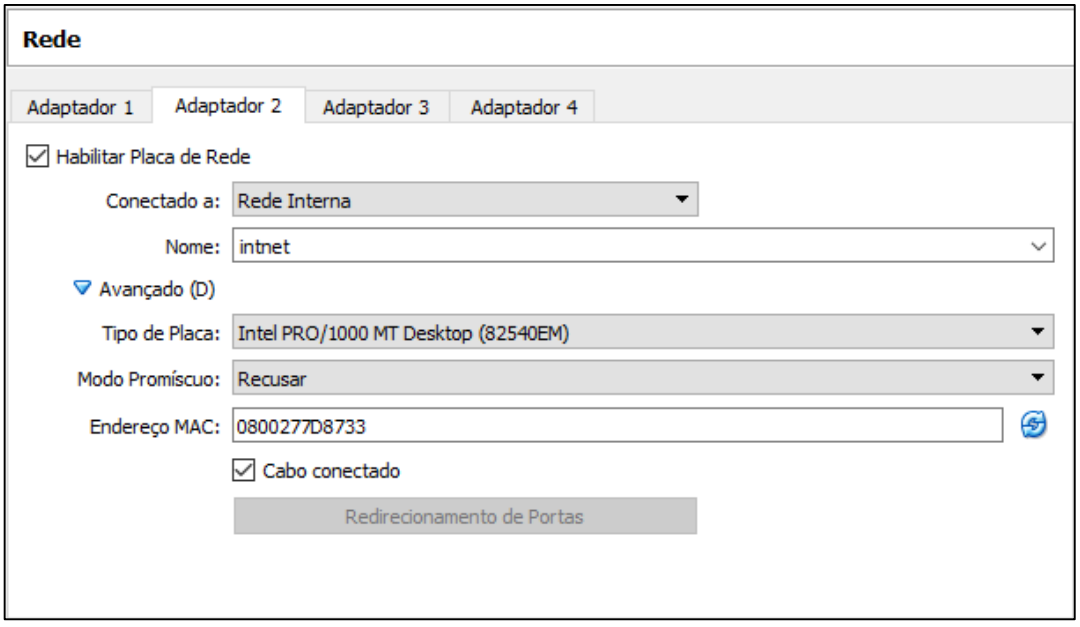
A máquina virtual FreeBSD – Firewall possui configuração diferente das demais pois é ele quem faz a ponte de comunicação entre as duas redes, interna e externa, funcionando como um roteador. Portanto, conforme ilustrado na figura 16, 17 e 18 o firewall possui três interfaces de rede: A primeira para conexão com a Internet, a segunda para comunicação com a rede externa e a terceira para comunicação com a rede interna.

Figura 16 - Adaptador 1 do FreeBSD Firewall



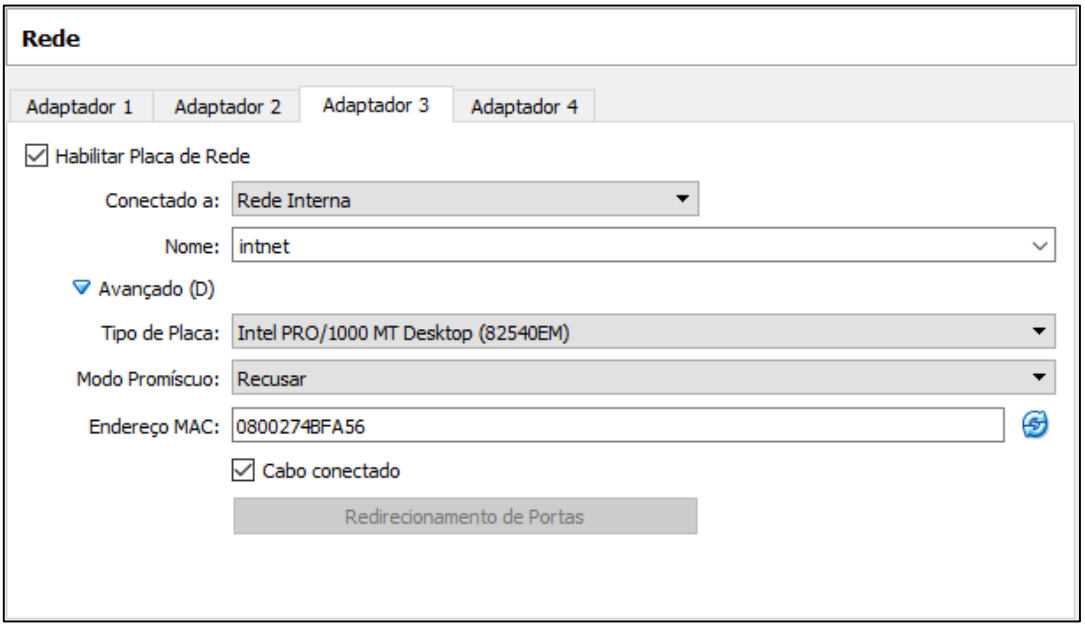
Fonte: Próprio autor

Figura 17 - Adaptador 2 do FreeBSD Firewall



Fonte: Próprio autor

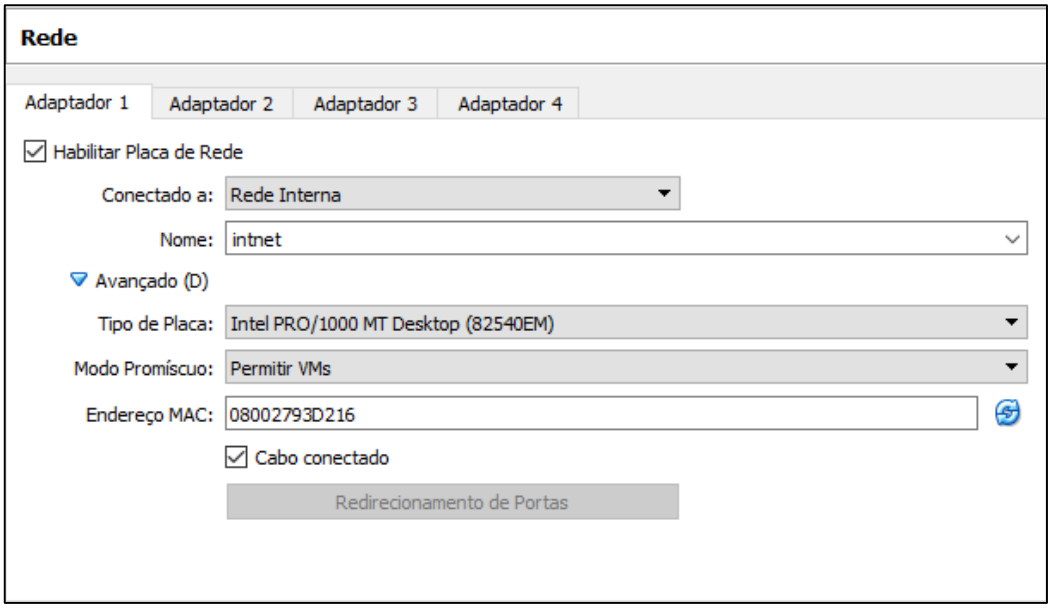
Figura 18 - Adaptador 3 do FreeBSD Firewall



Fonte: Próprio autor

Já o Debian – Snort atua na rede em modo Bridge, onde ele “escuta” todos os pacotes que trafegam por ela. Essa configuração é necessária pois o IDS deve procurar por comportamentos suspeitos não apenas de conexões destinadas a ele, mas sim para todos os hosts dentro da rede onde ele atua. A figura 19 mostra a configuração do adaptador de rede Snort.

Figura 19 - Adaptador 1 do Debian Snort



Fonte: Próprio autor.

4.3 SISTEMAS OPERACIONAIS E FERRAMENTAS

Para criação deste ambiente, foram utilizadas duas vertentes de sistemas operacionais baseados em Unix. A escolha desses sistemas não foi aleatória, mas sim com o propósito de utilizar o melhor de cada distribuição e adquirir maior leveza ao executar todas as máquinas virtuais ao mesmo tempo.

4.3.1 FREEBSD

O FreeBSD é um sistema operacional do tipo *Unix-like*, entretanto, por motivos judiciais não pode utilizar a marca Unix. O FreeBSD tem como principais objetivos a facilidade de uso de seu sistema e suporte a diferentes arquiteturas hardware. Apesar de possuir uma versão para usuários finais, seu uso é amplamente difundido em servidores de alta performance.

4.3.2 DEBIAN

O Debian foi lançado em 1993 por Ian Murdock e sua esposa Debra, inspiração para o nome *Debian*. Baseado em sistemas *Unix-like,* é tanto utilizado para servidores como para desktops, possuindo grande suporte a inúmeras interfaces gráficas. Sendo uma das distribuições mais estáveis dessa família, vários outros sistemas comerciais tomaram o Debian como base, sendo um dos mais conhecidos para utilização em usuário final o Ubuntu. Uma curiosidade atribuída a este sistema operacional é que, suas versões em fase Testing recebem nomes retirados da franquia de filmes Toy Story. Quando essas versões se tornam *Stable*, recebem um número de versão. Atualmente, o Debian está na sua versão *Stable* 8.5 Jessie.

4.4 IPFW

De acordo com Tracanelli e Goto (sem data) o IPFW é: “é o utilitário mais comum e mais popular pra implementar a filtragem de pacotes IP e controle de tráfego de rede no FreeBSD”. Existem diversas opções de ferramentas para implementação de firewall. Mesmo que alguns deles possuam certas particularidades, em um geral cumprem as mesmas funções de filtragem. Portanto, vai da escolha de cada administrador a qual usar. O IPFW foi escolhido especialmente pela facilidade de escrita das regras e por estar sendo ministrado nas aulas de Segurança em sistemas operacionais e redes de computadores, ministrada pelo professor Marcus Lahr. Sendo um filtro de pacotes, o IPFW só irá aceitar ou rejeitar um pacote dependendo das regras que foram previamente configuradas e de acordo com as informações que tem acesso.

O IPFW pode ser configurado de duas formas: *Open* ou *closed*. No modo open, todas as conexões são permitidas e apenas as definidas no arquivo de regras de configuração são bloqueadas. No modo closed, pelo contrário, todas as conexões são bloqueadas e apenas as regras de liberação definidas no arquivo de configuração são permitidas. No apêndice B é possível encontrar os procedimentos

para ativação do IPFW no FreeBSD e o arquivo de configuração com as regras de bloqueio utilizadas neste trabalho.

4.5 SNORT

O Snort é uma ferramenta de detecção de invasão baseado em rede. Por ser open source, possui constante atualização de software. Quando aplicado em uma rede, o Snort é responsável pela análise de diversos protocolos se baseando nas regras previamente configuradas. No contexto do Snort, regras são um conjunto de parâmetros que quando satisfeitos, geram um alerta. Além da dos próprios recursos que o Snort oferece, é possível integrá-lo com outras aplicações, como: Banco de dados, Snortsam, Guardian, Iptables, IPFW,
PHP, etc. No apêndice C encontram-se os comandos necessários para a instalação das bibliotecas e módulos que o Snort utiliza e posteriormente a instalação do Snort.

4.5.1 GUARDIAN

O Guardian é uma aplicação que funciona juntamente com o Snort. A partir dos logs gerados pela captura de eventos dentro da rede compatíveis com as regras de comportamento do Snort, o Guardian atua bloqueando as conexões de origem consideradas suspeitas através da integração com um firewall. A configuração do Guardian pode ser encontrada no apêndice D.

4.5.2 BARNYARD

Conforme artigo publicado no Vivaolinux.com.br (2014), o Barnyard pode ser definido como um interpretador de logs de Snort que, após a análise, armazena esses logs em um banco de dados. Sua instalação e configuração pode ser encontrada no apêndice E.

4.5.2.1 BASE

A fim de demonstrar maior riqueza e clareza de informações, foi feito uso de uma interface gráfica para o Snort, o utilitário BASE. O BASE “é uma ferramenta de navegação para análise de dados, construída utilizando-se da linguagem de programação PHP.” (VIVAOLINUX, 2017). Com ele, é possível acompanhar em tempo real os logs gerados pelo Snort e armazenados pelo Barnyard, através de gráficos. Neste trabalho foi utilizada a versão 1.4.5. Sua instalação está descrita no apêndice F.

4.6 HYDRA

A maioria dos usuários faz uso de senhas de baixa complexidade para acesso a sistemas e servidores. Palavras relacionadas à família, datas, esporte, sequência de caracteres ou nomes são as mais utilizadas. Uma justificativa para isso seria a fácil memorização. Entretanto, devido a frequência com que senhas que se enquadram nessas categorias, ataques de força bruta utilizam de dicionários que possuem em seu conteúdo senhas que seguem exatamente os mesmos padrões.

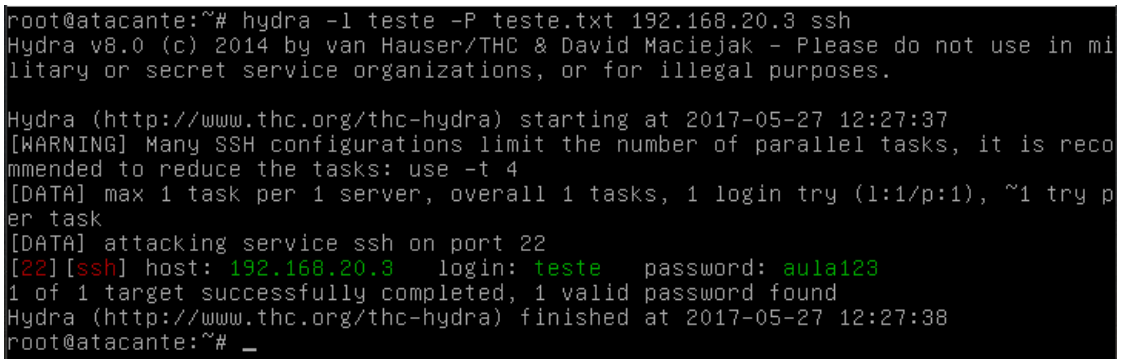
O Hydra é uma ferramenta para a quebra de logins e senhas dentro de uma rede. Fazendo uso do método de adivinhação, o Hydra utiliza de dicionários ou arquivos de possíveis credenciais de acesso para realizar o ataque. Ao ser iniciado, o software tenta todas as possíveis combinações de senhas até encontrar a correta.

Sendo um utilitário de linha de comando, ao iniciar um ataque pelo Hydra é possível definir alguns parâmetros, como: Definir o login do usuário ou arquivo de possíveis nomes, o arquivo de senha, endereço do servidor alvo, tipo de serviço a ser explorado. Ao obter sucesso, o Hydra informa a senha encontrada. Devido a grande quantidade de requisições de acesso que ele faz ao ser iniciado, o Hydra foi utilizado para simular um ataque de força bruta.

Na figura 20 é possível visualizar a utilização da ferramenta Hydra. A primeira linha de comando utiliza os seguintes parâmetros: O software utilizado, neste caso o

(hydra), (-l teste) é o usuário o qual o ataque visa descobrir a senha, (-P teste.txt) é o dicionário de senhas que o hydra irá utilizar para realizar o ataque de força bruta, (192.168.20.3) é o IP destino do ataque e (ssh) o protocolo que será explorado.

Figura 20- Representação de funcionamento do Hydra



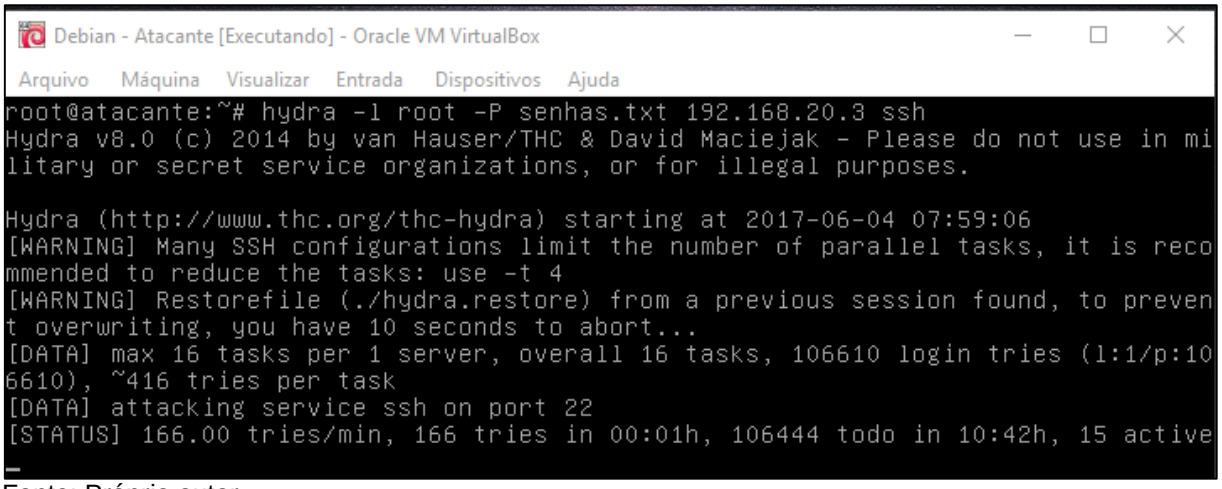
Fonte: Próprio autor.

No apêndice G é possível encontrar a instalação e testes realizados com o Hydra.

4.7 TESTES

Como pode ser observado na figura 21, primeiramente foi iniciado o ataque através da máquina Debian – Atacante, sendo seu IP 192.168.10.2. Com o utilitário Hydra, através de linha de comando, foram definidos os seguintes parâmetros: -l (nome do usuário), -P (arquivo de senha), IP da máquina alvo (192.168.20.3) e protocolo a ser explorado. A saída foi a seguinte:
hydra –l root –P senhas.txt 192.168.20.3 ssh.

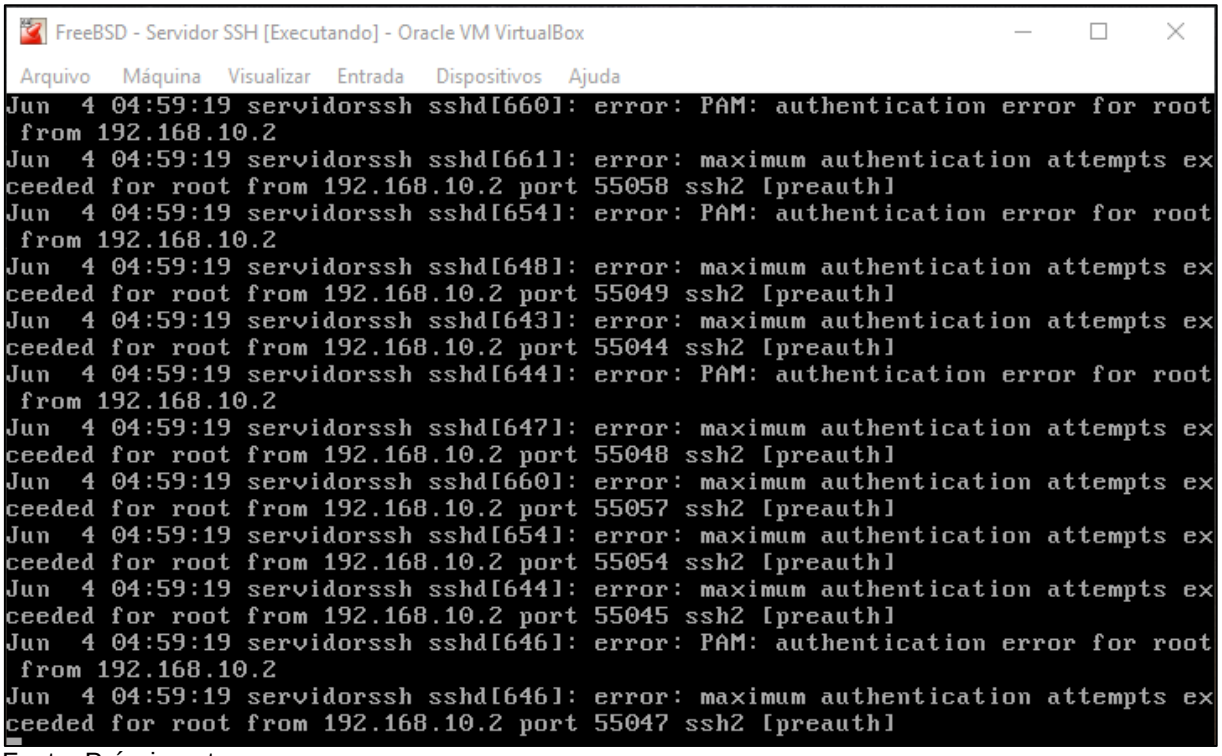
Figura 21 - Ataque com Hydra



Fonte: Próprio autor.

Logo após o período de 10 segundos informado pelo Hydra, tem início o ataque. Na máquina alvo, o Debian - Servidor SSH, já é possível encontrar tentativas de acesso sem sucesso. Essas tentativas de acesso podem ser encontradas na figura 22.

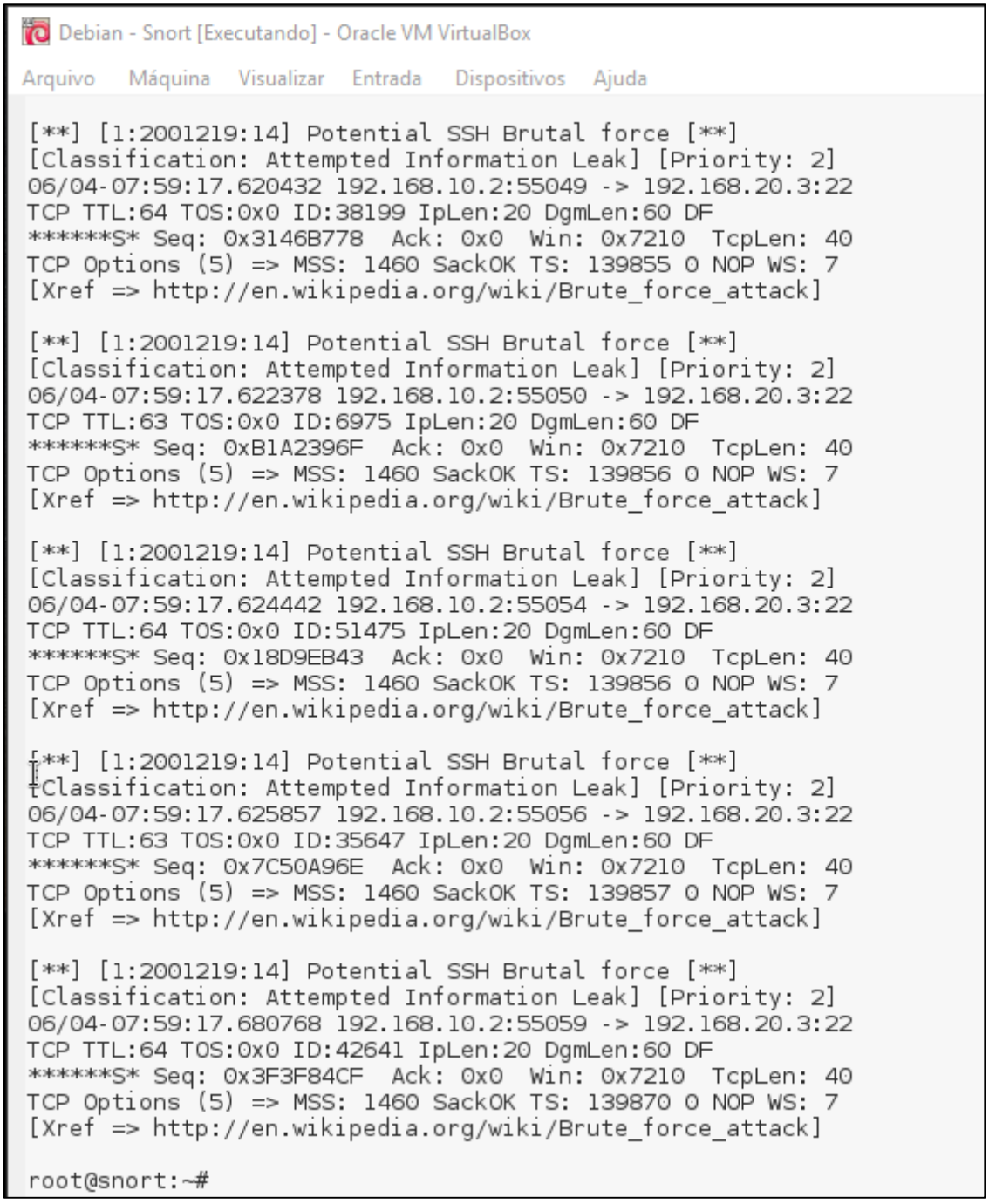
Figura 22 - Mensagens de tentativa de acesso no servidor SSH



Fonte: Próprio autor.

Na máquina Debian Snort, tomando como base suas regras de detecção, é possível encontrar os logs gerados pela ferramenta, que identificaram uma anomalia na rede. Os logs do Snort podem ser encontrados em /var/log/snort/alert e sua saída é ilustrada na figura 23.

Figura 23 - Logs gerados pelo Snort

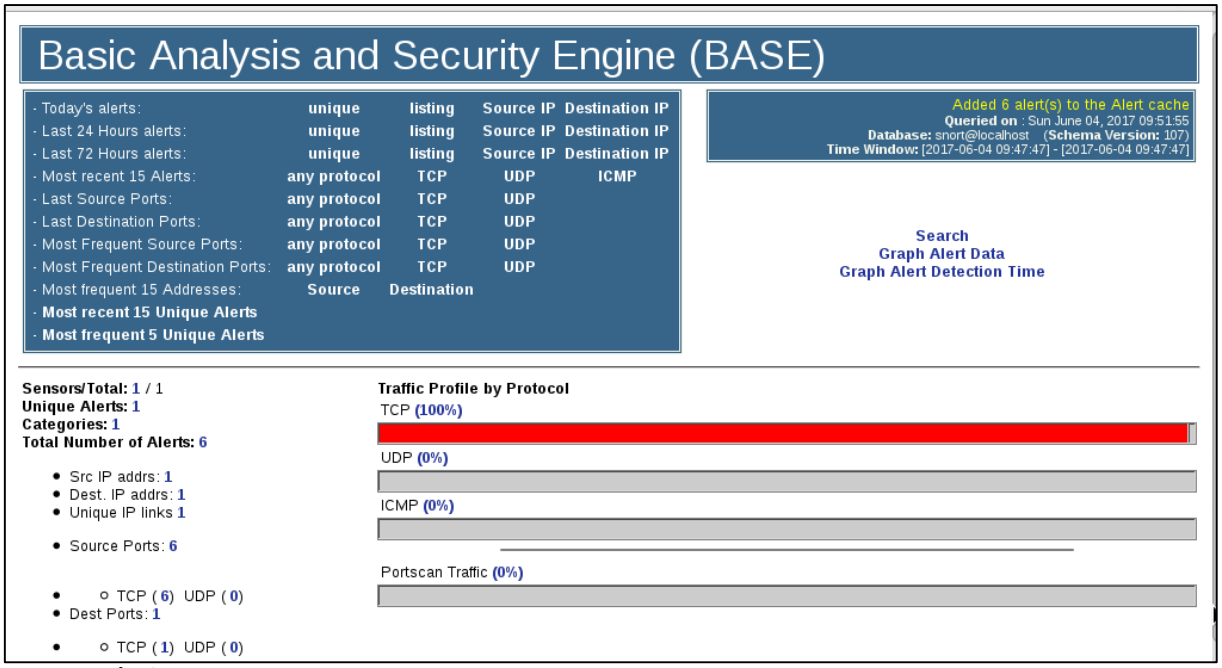


Fonte: Próprio autor.

O Barnyard, salva as informações contidas no arquivo de log do Snort em um banco de dados. De acordo com a figura 24, por meio do utilitário BASE é possível analisar

essas informações pelo navegador, onde são destacados o protocolo explorado e a quantidade de alertas gerados.

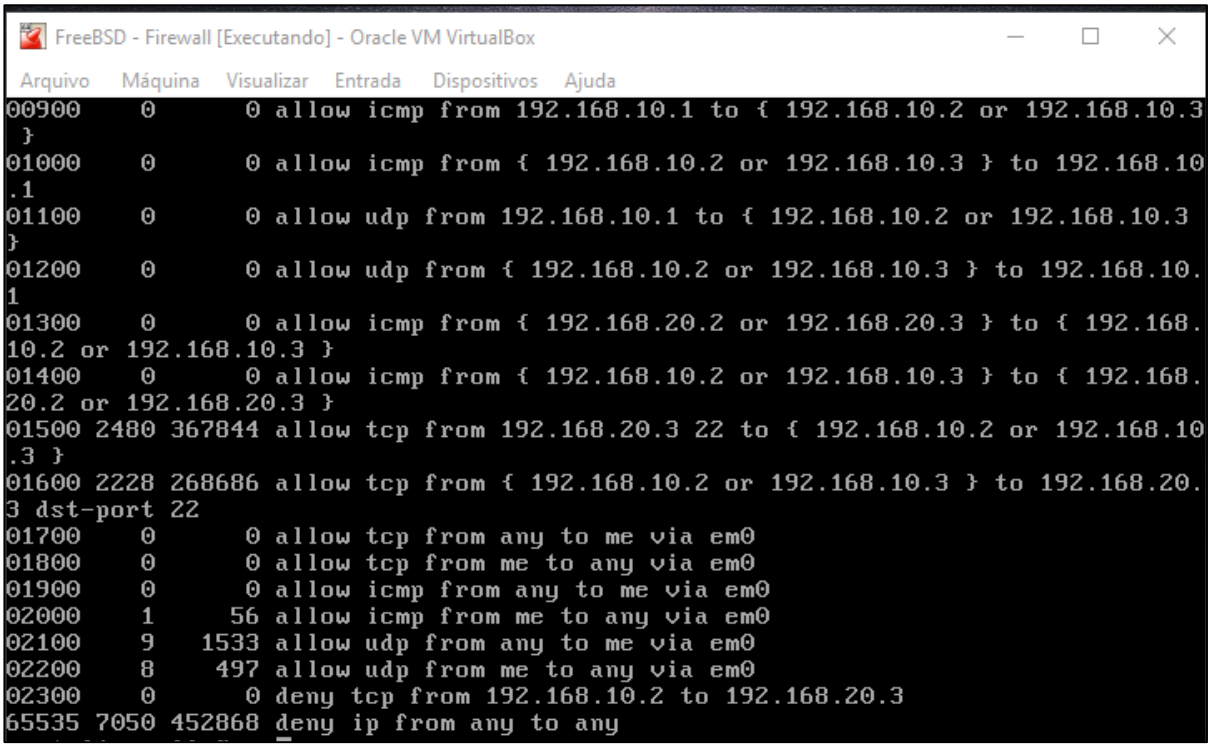
Figura 24 - Acesso as informações através do navegador



Fonte: Próprio autor.

Neste intervalo de tempo, o Guardian realiza uma conexão SSH juntamente com o firewall enviando uma regra de bloqueio de acordo com os parâmetros da conexão maliciosa. Na figura 25 é possível visualizar essa regra, sendo ela a última a ser incluída.

Figura 25 - Nova regra de firewall no IPFW



Fonte: Próprio autor.

5 CONCLUSÃO

Ainda com toda segurança envolvida, servidores SSH podem ser alvos de ataques de força bruta. Dependendo da dimensão do ataque ele pode vir a ser considerado um ataque de negação de serviço. Portanto, é imprescindível que, após a instalação de um servidor SSH, suas regras definidas por padrão sejam modificadas. Parâmetros como: porta, tentativas de acesso simultâneo, acesso remoto do root, permissão de acesso sem senha, etc.

Elencando com qualquer outro tipo de serviço ou aplicação, ainda assim é de extrema importância que mais de uma forma de prevenção e detecção seja aplicada dentro de uma rede. O Snort, por exemplo, sem o Guardian ou o firewall, funciona apenas como um sistema de alarme. Individualmente, ele não bloqueia ou toma qualquer tipo de atitude para parar uma ação, apenas avisa que algo está ocorrendo. Portanto, essas ferramentas não devem só ser aplicadas, também devem estar trabalhando em conjunto, diminuindo o impacto de uma tentativa de ataque em uma rede por meio de uma vulnerabilidade encontrada.

É oportuno frisar também que o motivo da mitigação das ferramentas de segurança, por mais que estivessem trabalhando em conjunto, mas em máquinas separadas, é não sobrecarregar o sistema operacional de cada uma e consequentemente causar uma má performance. Além de que, quando a conexão é bloqueada na primeira linha de defesa da rede, os recursos de banda dessa rede não são consumidos, não causando lentidão ou interoperabilidade dos seus serviços e sistemas.

6. CONSIDERAÇÕES FINAIS

Como possibilidade para futuros trabalhos acadêmicos, baseando nesta mesma abordagem, existe a oportunidade de configurar o Snort trabalhando como um IPS (*Intrusion Prevention System*), onde ele é a primeira barreira de segurança de uma rede. Outra possibilidade, é utilizar a função do Snort como *Host-based* em que ele verifica conexões e comportamentos suspeitos apenas na máquina onde está instalado.

APÊNDICE A – CONFIGURAÇÃO DE REDE DAS MÁQUINAS VIRTUAIS

1. Firewall:

Figura 26 - Configuração de rede do FreeBSD Firewall

```
#nome da maquina
hostname="firewall"

#teclado
keymap="/root/teclado.kbd"

#rede
ifconfig_em0="DHCP"
ifconfig_em1="inet 192.168.10.1 netmask 255.255.255.0"
ifconfig_em2="inet 192.168.20.1 netmask 255.255.255.0"
gateway_enable="YES"
```

Fonte: Próprio autor.

2. Snort:

Figura 27 - Configuração de rede do Debian Snort

```
root@snort:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.20.2
network 192.168.20.0
gateway 192.168.20.1
broadcast 192.168.20.255
root@snort:~#
```

Fonte: Próprio autor.

3. Servidor SSH:

Figura 28 - Configuração de rede do Debian Servidor SSH

```
root@servidorssh:~ # cat /etc/rc.conf
hostname="servidorssh"
keymap="br275.iso.acc.kbd"
ifconfig_em0="inet 192.168.20.3 netmask 255.255.255.0"
defaultrouter="192.168.20.1"
ipv6_enable="NO"
```

Fonte: Próprio autor.

4. Cliente SSH

Figura 29 - Configuração de rede do Debian Cliente SSH

```
root@clientessh:~ # cat /etc/rc.conf
hostname="clientessh"
ifconfig_em0="inet 192.168.10.3 netmask 255.255.255.0"
defaultrouter="192.168.10.1"
ipv6_enable="NO"
```

Fonte: Próprio autor.

4. Atacante

Figura 30 - Configuração de rede do Debian Atacante

```
root@atacante:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface

allow-hotplug eth0
iface eth0 inet static
address 192.168.10.2
network 192.168.10.0
netmask 255.255.255.0
gateway 192.168.10.1
broadcast 192.168.10.255
```

Fonte; Próprio autor.

APÊNDICE B – CONFIGURAÇÃO DO IPFW

Após as configurações de rede, já é possível habilitar o IPFW no FreeBSD.

1. Carregar o ipfw:

```
# kldload ipfw
```

2. Criar script com as regras de bloqueio:

```
# ee /etc/rc.firewall.rules
```

3. Adicionar as regras:

Figura 31 - Regras de firewall 1

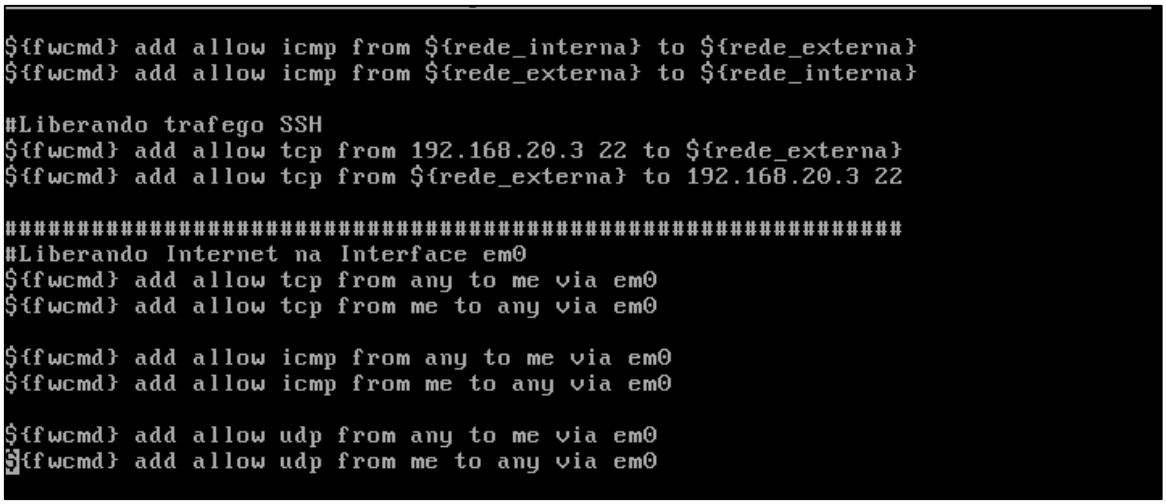
```
#!/bin/sh

#Script de firewall
#Criado em 26/04/2017
#Autor Nathalia Peres

fwcmd="/sbin/ipfw"
rede_interna="{192.168.20.2 or 192.168.20.3 }"
rede_externa="{192.168.10.2 or 192.168.10.3 }"

${fwcmd} add check-state
${fwcmd} -q flush

#####################################################################
#Liberando trafego na rede interna
${fwcmd} add allow tcp from 192.168.20.1 to ${rede_interna}
${fwcmd} add allow tcp from ${rede_interna} to 192.168.20.1
```

Fonte: Próprio autor.

Figura 32 - Regras de firewall 2

Fonte: Próprio autor.

Figura 33 - Regras de firewall 3



Fonte: Próprio autor.

4. Abrir arquivo de configuração

```
# ee /etc/rc.conf
```

5. Adicionar a linha abaixo para habilitar o firewall após a inicialização da máquina:

```
# firewall_enable="YES"
```

6. Adicionar o arquivo de regras como padrão para o ipfw:

```
# firewall_script="/etc/rc.firewall.rules"
```

7. Configuração final no arquivo */etc/rc.conf*:

Figura 34 - Configuração do arquivo rc.conf do FreeBSD Firewall:

Fonte: Próprio autor.

APÊNDICE C – INSTALAÇÃO E CONFIGURAÇÃO DO SNORT

1. Primeiramente, é necessário realizar a instalação de algumas bibliotecas:

```
# apt install build-essential libpcap-dev libpcre3-dev
libdumpnet-dev bison flex -y zlib1g-dev liblzma-dev openssl
libssl-dev autoconf libtool pkg-config
```

2. Para agrupar todos os módulos necessários, foi criada a pasta IDS:

```
# mkdir IDS
```

3. Entrar na pasta

```
# cd IDS
```

4. Baixar a aplicação de apoio ao Snort:

```
# wget https://snort.org/downloads/snort/daq-2.0.6.tar.gz
```

5. Descompactar o arquivo:

```
# tar -xvzf daq-2.0.6.tar.gz
```

6. Entrar na pasta:

```
# cd daq-2.0.6
```

7. Gerar arquivos de instalação:

```
# ./configure
```

8. Compilar:

```
# make
```

9. Instalar:

```
# make install
```

10. Voltar a pasta anterior

```
# cd ..
```

11. Baixar a aplicação segunda aplicação de apoio ao Snort:

```
# wget
https://github.com/nghttp2/nghttp2/releases/download/v1.17.0/n
ghttp2-1.17.0.tar.gz
```

12. Descompactar a aplicação:

```
# tar -xzvf nghttp2-1.17.0.tar.gz
```

13. Entrar na pasta

```
# cd nghttp2-1.17.0
```

14. Realizar atualização de ficheiros:

```
# autoreconf –i –force
```

15. Compilar:

```
# automake
```

16. Gerar *script* de configuração:

```
# autoconf
```

17. Gerar arquivos de instalação e bibliotecas:

```
# ./configure --enable-lib-only
```

18. Compilar:

```
# make
```

19. Instalar a aplicação:

```
# make install
```

20. Voltar um diretório:

```
# cd ..
```

21. Realizar o download do Snort:

```
# wget https://snort.org/downloads/snort/snort-
2.9.9.0.tar.gz
```

22. Descompactar o arquivo:

```
# tar -xvzf snort-2.9.9.0.tar.gz
```

23. Entrar na pasta:

```
# cd snort-2.9.9.0
```

24. Gerar arquivos de instalação:

```
# ./configure --enable-sourcefire
```

25. Compilar:

```
# make
```

26. Instalar a aplicação:

```
# make install
```

27. Atualizar o cache de bibliotecas:

```
# ldconfig
```

28. Criar link simbólico do arquivo:

```
# ln -s /usr/local/bin/snort /usr/sbin/snort
```

29. Verificar se a instalação ocorreu com sucesso:

```
# snort -V
```

Figura 35 - Versão do Snort

```
root@snort:~# snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.9.0 GRE (Build 56)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2016 Cisco and/or its affiliates. All rights reser
ved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.6.2
           Using PCRE version: 8.35 2014-04-04
           Using ZLIB version: 1.2.8
```

Fonte: Próprio autor.

21. Criar um grupo e usuário para o Snort:

```
# groupadd snort

# useradd snort -r -s /sbin/nologin -c SNORT_IDS -g snort
```

22. Realizar a criação de diretórios para o snort, as quais são especificadas no arquivo de configuração:

```
# mkir /etc/snort sudo /etc/snort/rules
/etc/snort/rules/iplists /etc/snort/preproc_rules
/usr/local/lib/snort_dynamicrules /etc/snort/so_rules
```

23. Criar os arquivos de regras:

```
# touch /etc/snort/rules/iplists/black_list.rules
/etc/snort/rules/iplists/white_list.rules
/etc/snort/rules/local.rules /etc/snort/sid-msg.map
```

24. Criar os diretórios para armazenamento dos logs:

```
# mkdir /var/log/snort /var/log/snort/archived_logs
```

25. Inserir permissões de acesso:

```
# chmod -R 5775 /etc/snort sudo

# chmod -R 5775 /var/log/snort

# chmod -R 5775 /var/log/snort/archived_logs

# chmod -R 5775 /etc/snort/so_rules

# chmod -R 5775 /usr/local/lib/snort_dynamicrules
```

26. Mudar proprietário dos arquivos:

```
# chown -R snort:snort /etc/snort

# chown -R snort:snort /var/log/snort

# chown -R snort:snort /usr/local/lib/snort_dynamicrules
```

27. Entrar na pasta /IDS/snort-2.9.9.0/etc e copiar os arquivos para a pasta /etc do sistema:

```
# cp *.conf* /etc/snort
# cp *.map /etc/snort
# cp *.dtd /etc/snort
```

28. Entrar na pasta:

```
# cd IDS/snort-2.9.9.0/src/dynamic-preprocessors/build/usr/local/lib/snort_dynamicpreprocessor/
```

29. Copiar o arquivo:

```
# cp * /usr/local/lib/snort_dynamicpreprocessor/
```

30. Comentar todos os arquivos:

```
# sed -i "s/include \$RULE\_PATH/#include \$RULE\_PATH/" /etc/snort/snort.conf
```

31. Editar o arquivo de configuração do Snort:

```
# nano /etc/snort/snort.conf
```

32. Na linha HOME_NET inserir o IP que o Snort irá monitorar:

```
HOME_NET 192.168.20.0/24
```

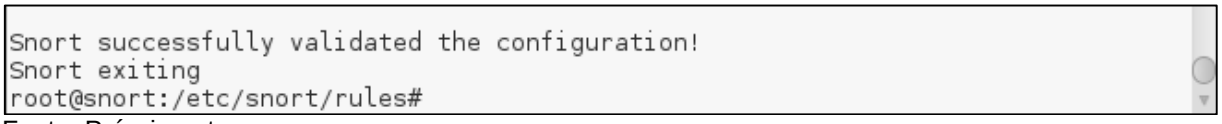
33. Definir os caminhos dos diretórios as variáveis e posteriormente salvar o arquivo:

```
RULE_PATH /etc/snort/rules

SO_RULE_PATH /etc/snort/so_rules

PREPROC_RULE_PATH /etc/snort/preproc_rules

WHITE_LIST_PATH /etc/snort/rules/iplists

BLACK_LIST_PATH /etc/snort/rules/iplists

include $RULE_PATH/local.rules
```

34. Verificar funcionamento do Snort:

```
sudo snort -T -i eth0 -c /etc/snort/snort.conf
```

Figura 36 - Verificação de funcionamento do Snort:



Fonte: Próprio autor.

35. Criar um script de inicialização para o Snort:

```
# nano /lib/systemd/system/snort.service
```

36. Inserir as linhas abaixo e posteriormente salvar o arquivo:

```
[Unit] Description=Snort NIDS Daemon
After=syslog.target network.target

[Service] Type=simple
ExecStart=/usr/local/bin/snort -q -u snort -g snort -c
/etc/snort/snort.conf -i ens160

[Install] WantedBy=multi-user.target
```

37. Ativar o Snort para iniciar no *boot* da máquina:

```
# sudo systemctl enable snort
```

38. Iniciar o serviço:

```
# sudo systemctl start snort
```

39. Verificar seu funcionamento:

```
# systemctl status snort
```

Figura 37 - Serviço do Snort em funcionamento

Fonte: Próprio autor.

APÊNDICE D – INSTALAÇÃO E CONFIGURAÇÃO DO GUARDIAN

1. Entrar na pasta do IDS:

```
# cd IDS
```

2. Baixar a aplicação:

```
# wget http://www.chaotic.org/guardian/guardian-
1.7.tar.gz
```

3. Descompactar o arquivo:

```
# tar -xvzf guardian-1.7.tar.gz
```

4. Entrar na pasta:

```
# cd guardian-1.7
```

5. Abrir o arquivo:

```
# nano guardian.conf
```

6. Alterar e descomentar as linhas:

```
HostIpAddr          192.168.20.2

Interface           eth0

AlertFile           /var/log/snort/alert
```

7. Salvar o arquivo e copiar para a pasta */etc*:

```
# cp guardian.conf /etc/
```

8. Criar o arquivo */etc/guardian.ignore:*

```
# nano /etc/guardian.ignore
```

9. Inserir o IP do servidor o qual o Guardian vai ignorar, neste caso :

```
192.168.20.2
```

10. Entrar na pasta *scripts*:

```
# cd scripts
```

11. Copiar os scripts de bloqueio e desbloqueio:

```
# cp iptables_block.sh /sbin/guardian_block.sh

# cp iptables_unblock.sh /sbin/guardin_unblock.sh
```

12. Voltar uma pasta:

```
# cd ..
```

13. Editar o arquivo guardian.pl:

```
# nano guardian.pl
```

14. Alterar a linha que contém “inet addr” para:

```
# inet end
```

15. Salvar o arquivo e copiar para o diretório /sbin:

```
# cp guardian.pl /sbin
```

16. Criar arquivo de log do Guardian:

```
# touch /var/log/guardian.log
```

17. Criar script de para inicialização do Guardian:

```
# nano /etc/init.d/guardian
```

18. Inserir as seguintes configurações:

```
#!/bin/bash

test -f /sbin/guardian.pl || exit 0
case "$1" in
    start)
          guardian.pl -c /etc/guardian.conf
          ;;
    stop)
          kill -9 $(pgrep guardian.pl)
          ;;
```

```
    *)
            echo "Opção invalida. Use start ou stop."
            exit 2
            ;;
esac
exit 0
```

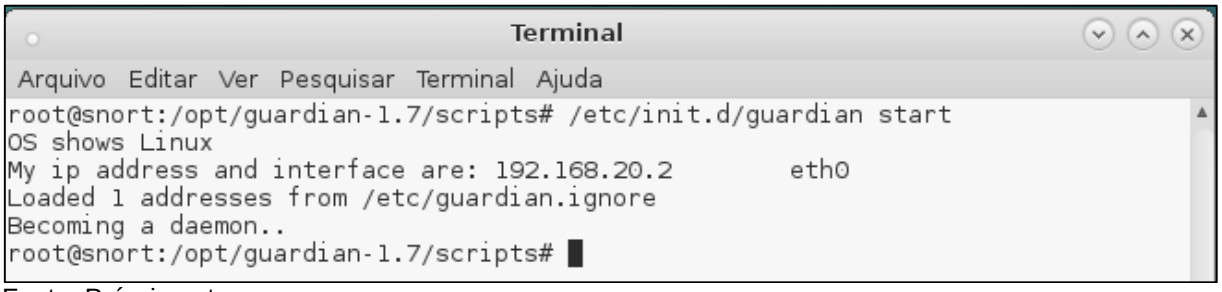
19. Dar permissão de execução para o script:

```
# chmod 755 /etc/init.d/guardian
```

20. Iniciar o Guardian:

```
# /etc/init.d/guardian start
```

Figura 38 - Serviço do Guardian em funcionamento



Fonte: Próprio autor.

APÊNDICE E – INSTALAÇÃO E CONFIGURAÇÃO DO BARNYARD

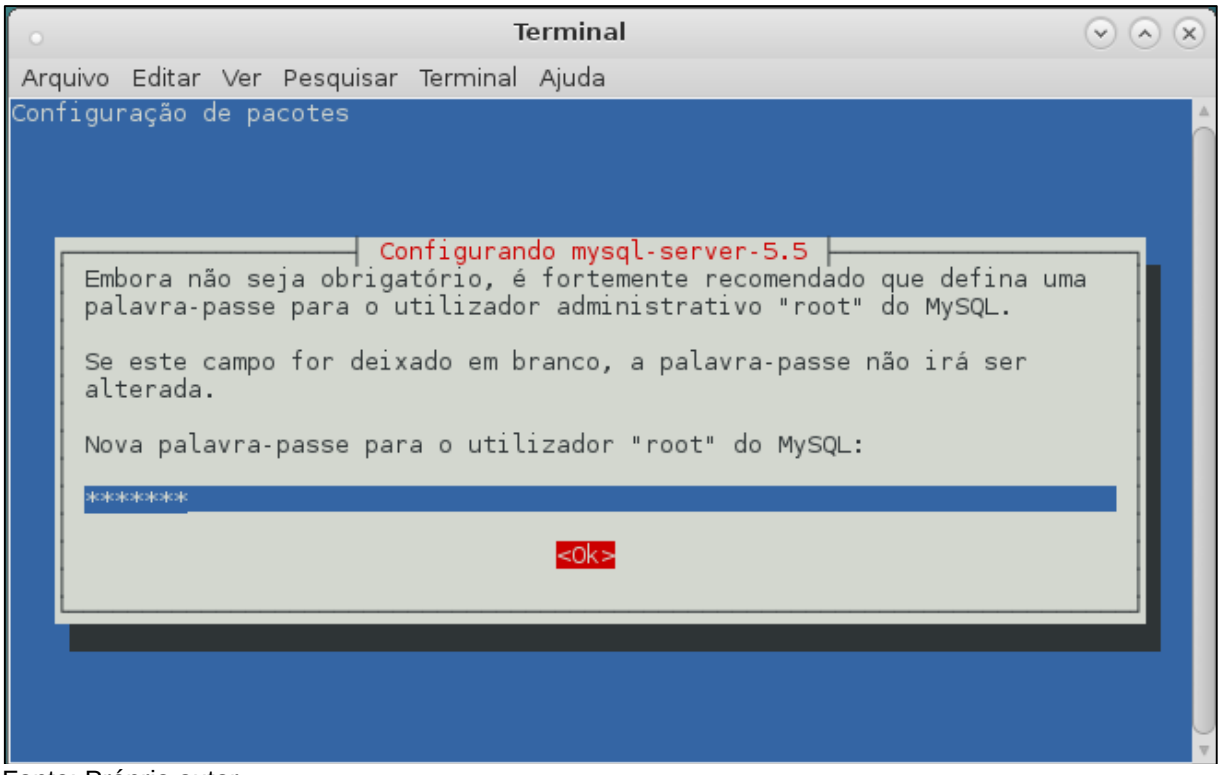
Antes de realizar o download da aplicação, é necessário configurar um banco de dados na máquina:

1. Instalar os pacotes necessários:

```
# apt-get install -y mysql-server libmysqlclient-dev
mysql-client autoconf libtool
```

2. Configurar uma senha para o usuário root do mysql, neste caso “aula123”. Em seguida, será solicitado a senha novamente para confirmação:

Figura 39 - Senha do usuário root do mysql



Fonte: Próprio autor.

3. Entrar no arquivo de configuração do Snort:

```
# /etc/snort/snort.conf
```

4. Descomentar a seguinte linha:

```
output unified2: filename snort.u2, limit 128
```

5. Entrar na pasta /IDS:

```
# cd /IDS
```

6. Realizar o download da aplicação:

```
# wget
https://github.com/firnsy/barnyard2/archive/master.tar.gz -O
barnyard2-Master.tar.gz
```

7. Descompactar a pasta:

```
# tar zxvf barnyard2-Master.tar.gz
```

8. Entrar na pasta da aplicação:

```
# cd barnyard2-master
```

9. Atualizar ficheiros de configuração:

```
# autoreconf -fvi -I ./m4
```

10. Criar link simbólico:

```
# ln -s /usr/include/dumbnet.h /usr/include/dnet.h
```

11. Atualização das bibliotecas:

```
# ldconfig
```

12. Instalação da biblioteca do mysql de acordo com o sistema operacional:

```
# ./configure --with-mysql --with-mysql-
libraries=/usr/lib/x86_64-linux-gnu
```

13. Compilar:

```
# make
```

14. Instalar a aplicação:

```
# make install
```

15. Verificar se a aplicação foi instalada corretamente:

```
# barbyard2 –V
```

Figura 40 - Versão do Barnyard2

Fonte: Próprio autor

16. Criar e copiar os arquivos de configuração do Barnyard:

```
# cp ~/snort_src/barnyard2-master/etc/barnyard2.conf
/etc/snort/

# mkdir /var/log/barnyard2

# chown snort.snort /var/log/barnyard2

# touch /var/log/snort/barnyard2.waldo

# chown snort.snort /var/log/snort/barnyard2.waldo
```

17. Entrar no mysql:

```
# mysql –u root –p
Enter password: aula123
```

18.Criar uma base de dados para o Snort:

```
mysql> create database snort;
```

19. Entrar na base de dados criada:

```
mysql> use snort;
```

20. Copiar o template do Barnyard para a base de dados:

```
mysql> source ~/snort_src/barnyard2-
master/schemas/create_mysql
```

21. Criar um usuário para acesso ao novo banco:

```
mysql> CREATE USER 'snort'@'localhost' IDENTIFIED BY
'aula123';

mysql> CREATE USER 'snort'@'192.168.20.2' IDENTIFIED BY
'aula123';
```

22. Dar permissões de acesso ao usuário:

```
mysql> grant create, insert, select, delete, update on
snort.* to 'snort'@'192.268.20.2';

mysql> grant create, insert, select, delete, update on
snort.* to 'snort'@'localhost';
```

23. Sair do mysql:

```
mysql> exit;
```

24. Entrar no arquivo de configuração do Barnyard:

```
# nano /etc/snort/barnyard2.conf
```

25. Editar o arquivo com as configurações para acesso ao mysql:

```
output database: log, mysql, user=snort
password=MySqlSNORTpassword dbname=snort host=localhost sensor
name=sensor01
```

26. Editar as permissões de acesso ao arquivo:

```
# chmod o-r /etc/snort/barnyard2.conf
```

27. Criar um arquivo de serviço para o Barnyard:

```
# nano /lib/systemd/system/barnyard2.service
```

28. Inserir as seguintes configurações:

```
[Unit]
Description=Barnyard2 Daemon
After=syslog.target network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/barnyard2 -c
/etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -q -w
/var/log/ snort/barnyard2.waldo -g snort -u snort -D -a
/var/log/snort/archived_logs

[Install]
WantedBy=multi-user.target
```

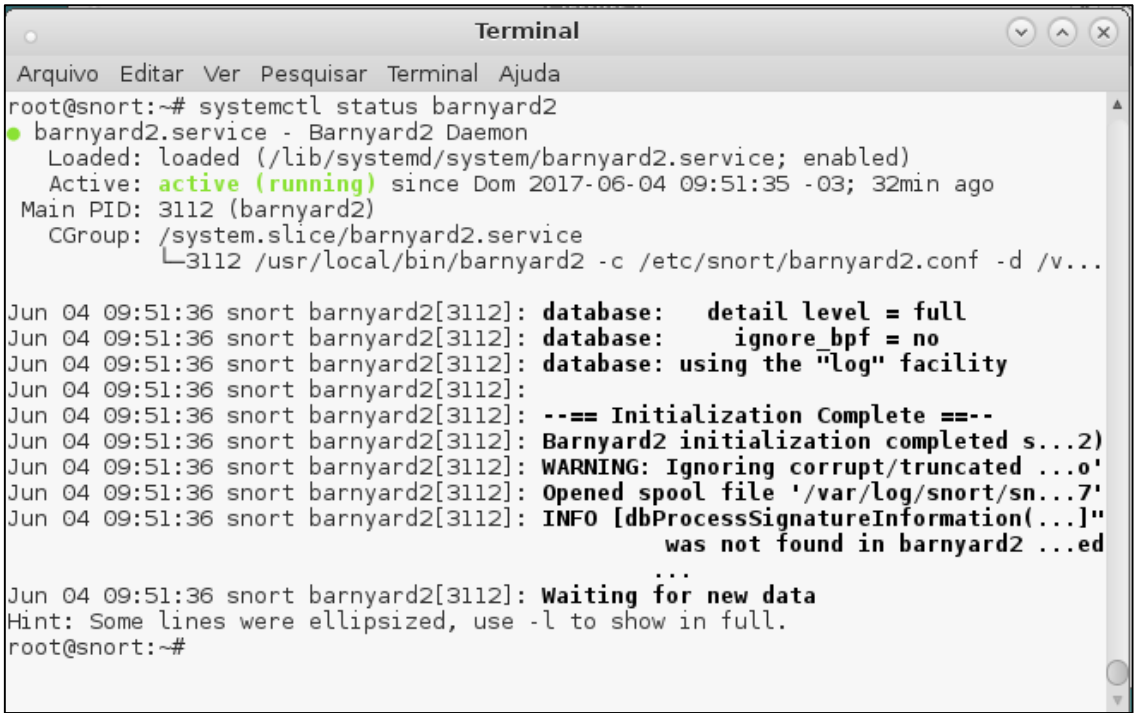
29. Iniciar o serviço no boot:

```
# systemctl enable barnyard2
```

30. Iniciar o serviço:

```
# systemctl start barnyard2
```

Figura 41 - Serviço do Barnyard2 em funcionamento



Fonte: Próprio autor.

APÊNDICE F – INSTALAÇÃO E CONFIGURAÇÃO DO BASE

1. Instalar as bibliotecas necessárias:

```
# apt install -y apache2 libapache2-mod-php5 php5 php5-mysql php5-common php5-gd php5-cli php-pear
```

2. Instalar gráficos:

```
# pear install -f --alldeps Image_Graph
```

3. Entrar na pasta /IDS:

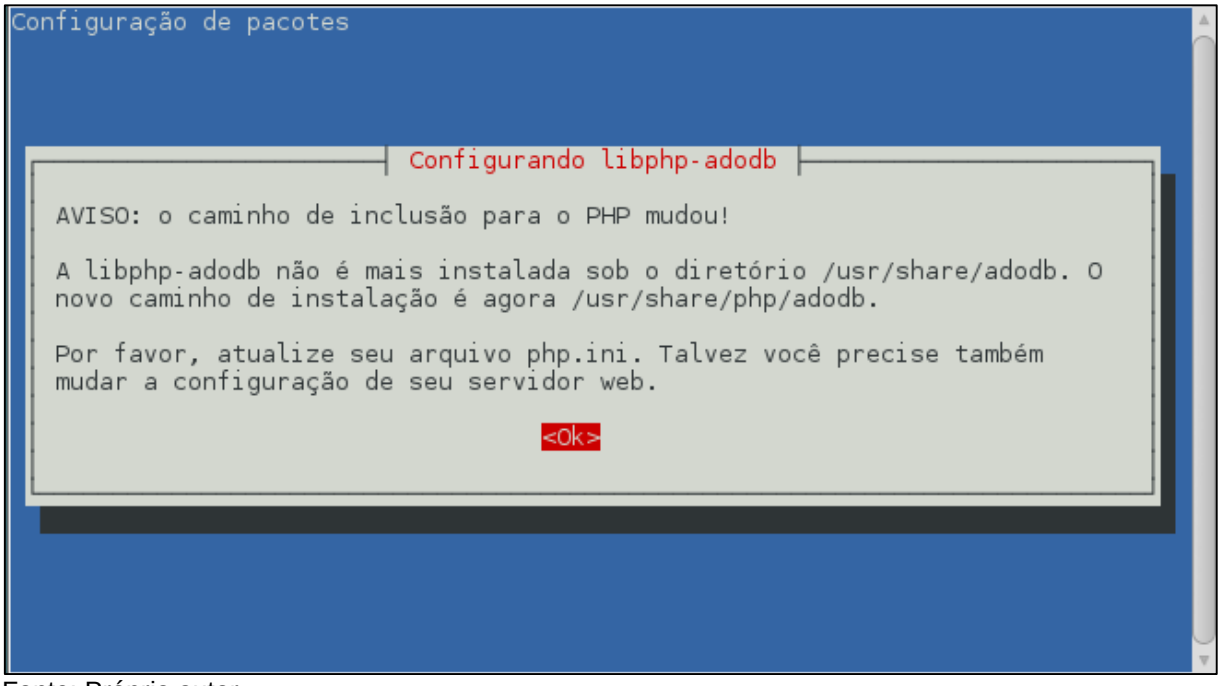
```
# cd /IDS
```

4. Baixar a aplicação ADODB:

```
# wget https://sourceforge.net/projects/adodb/files/adodb-php5-only/adodb-520-for-php5/adodb-5.20.8.tar.gz
```

5. Clicar em ok:

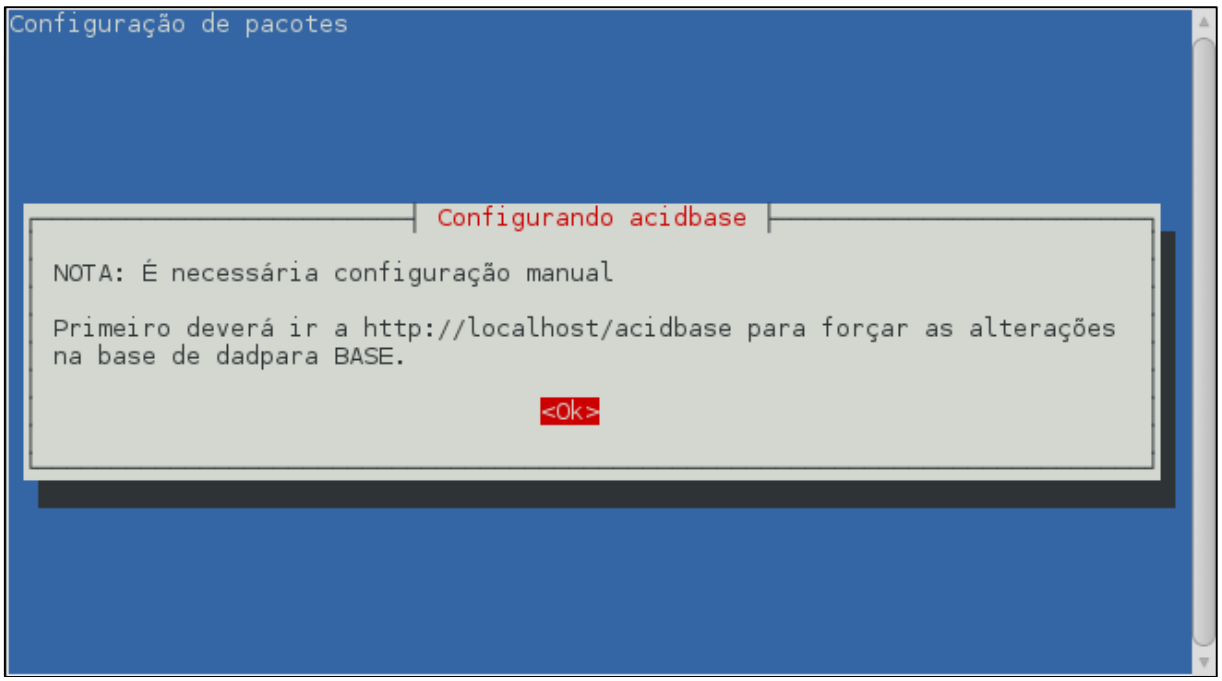
Figura 42 - Configuração adodb



Fonte: Próprio autor

6. Clicar em OK:

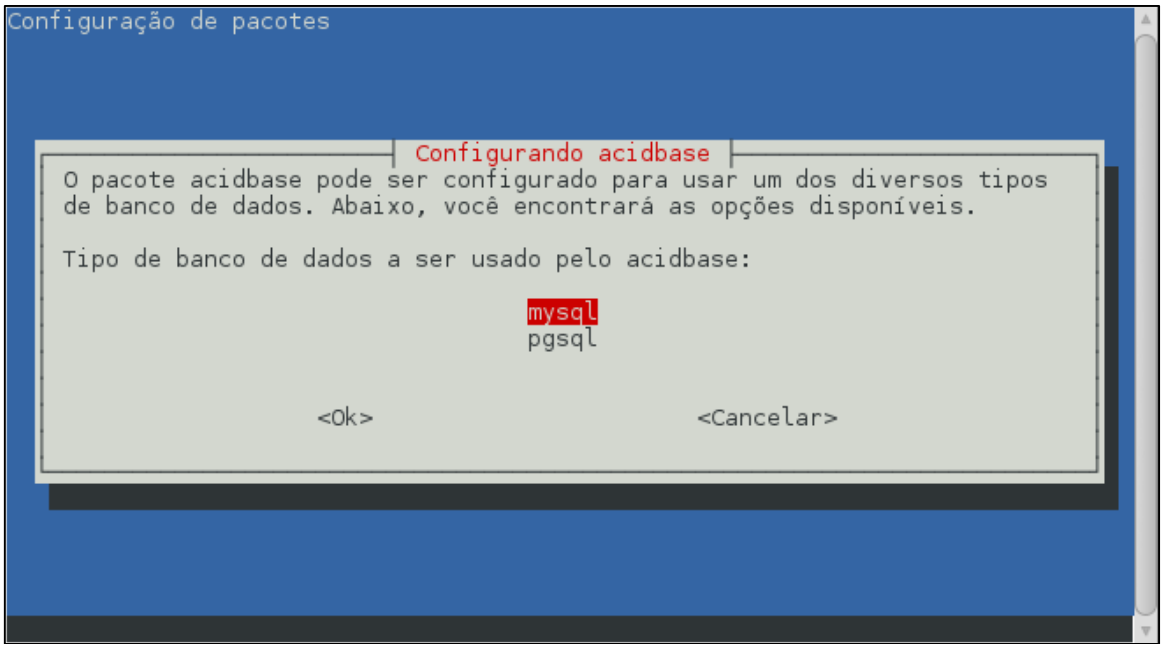
Figura 43 - Configuração do acidbase



Fonte: Próprio autor

7. Selecionar mysql:

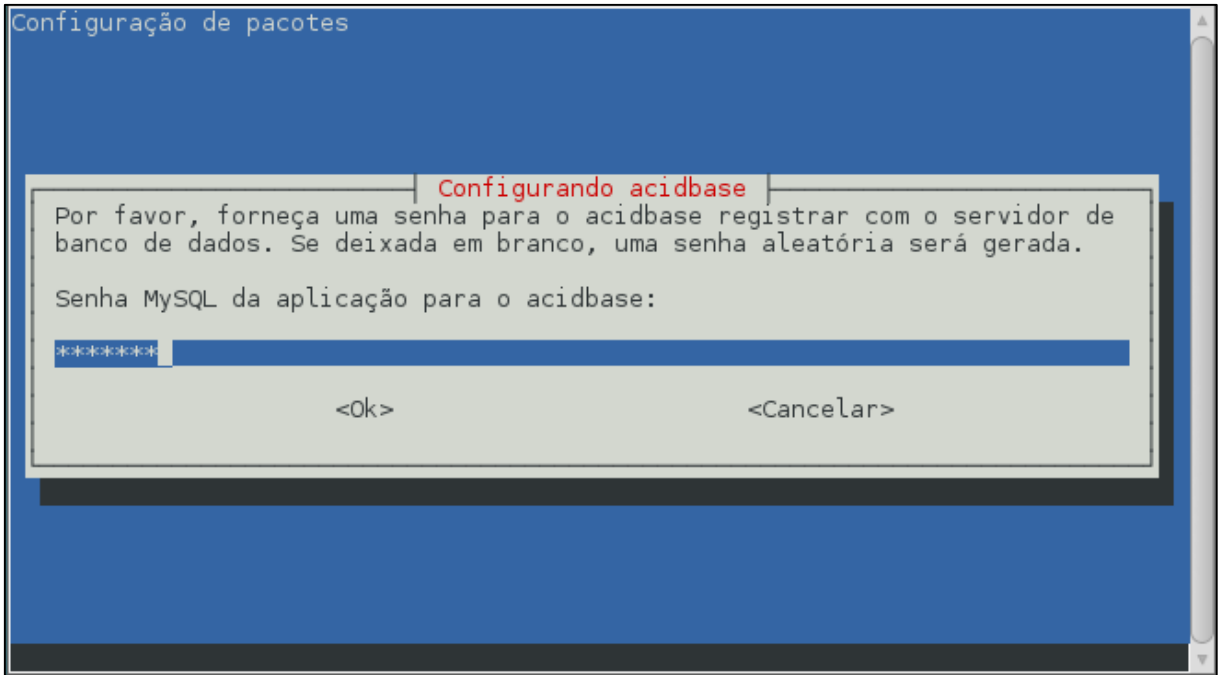
Figura 44 - Selecionar banco de dados



Fonte: Próprio autor

8. Inserir uma senha para o Acidbase. Foi inserida a senha aula123:

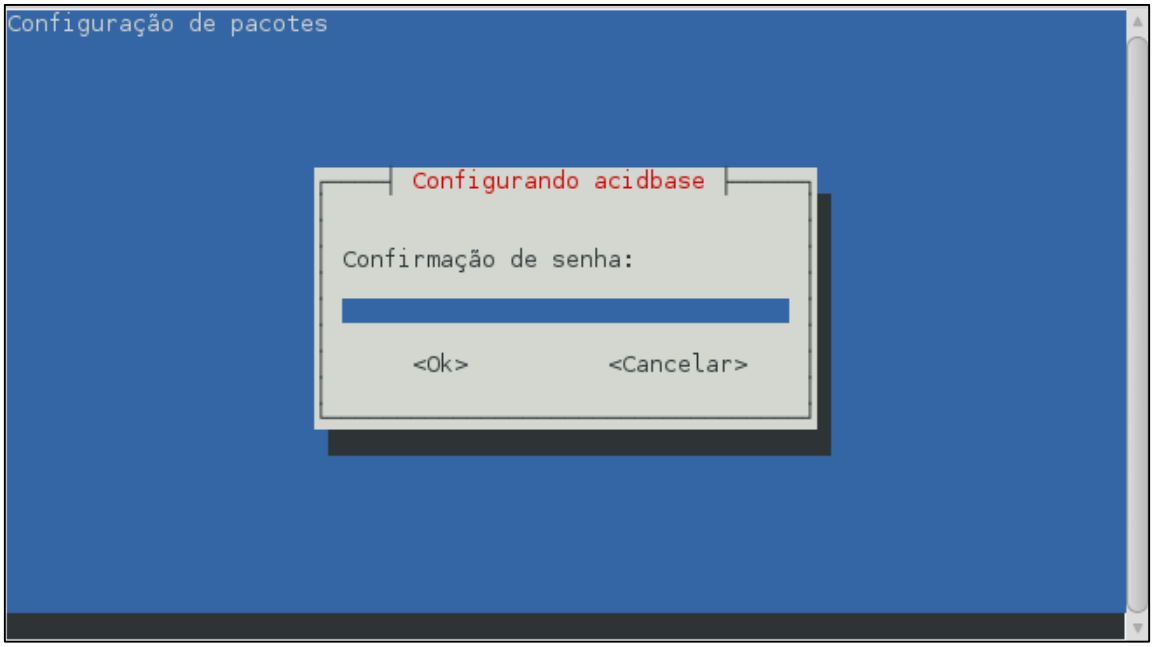
Figura 45 - Definir senha de acesso para o acidbase



Fonte: Próprio autor

9. Confirmar a senha:

Figura 46 - Confirmação de senha



Fonte: Próprio autor

10. Descompactar a pasta:

```
# tar -xvzf adodb-5.20.8.tar.gz
```

11. Mover a pasta da aplicação:

```
# mv adodb5 /var/adodb
```

12. Mudar as permissões de acesso a pasta:

```
# chmod -R 755 /var/adodb
```

13. Baixar a aplicação BASE:

```
# wget
http://sourceforge.net/projects/secureideas/files/BASE/base-
1.4.5/base-1.4.5.tar.gz
```

14. Descompactar a pasta:

```
# tar xzvf base-1.4.5.tar.gz
```

15. Mover a pasta da aplicação:

```
# mv base-1.4.5 /var/www/html/base/
```

16. Entrar na pasta do BASE:

```
# cd /var/www/html/base
```

17. Criar um novo arquivo:

```
# cp base_conf.php.dist base_conf.php
```

18. Editar arquivo de configuração:

```
# nano /var/www/html/base/base_conf.php
```

19. Alterar as seguintes linhas do arquivo:

```
$BASE_urlpath = '/base';

$DBlib_path = '/var/adodb/';

$alert_dbname = 'snort';
```

```
$alert_host = 'localhost';

$alert_port = '';

$alert_user = 'snort';

$alert_password = 'aula123';
```

20. Alterar as permissões de acesso:

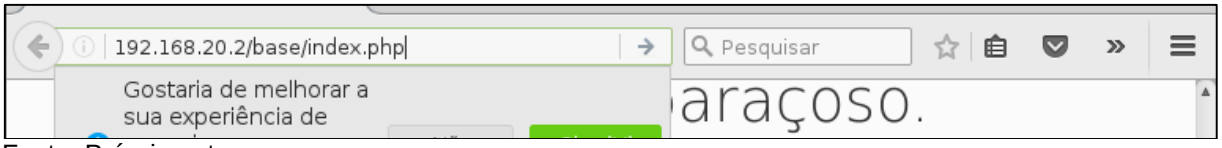
```
# chown -R www-data:www-data /var/www/html/base

# chmod o-r /var/www/html/base/base_conf.php
```

21. Reiniciar o serviço do apache:

```
# service apache2 restart
```

22. Pelo navegador, acessar o endereço http://192.168.20.2/base/index:

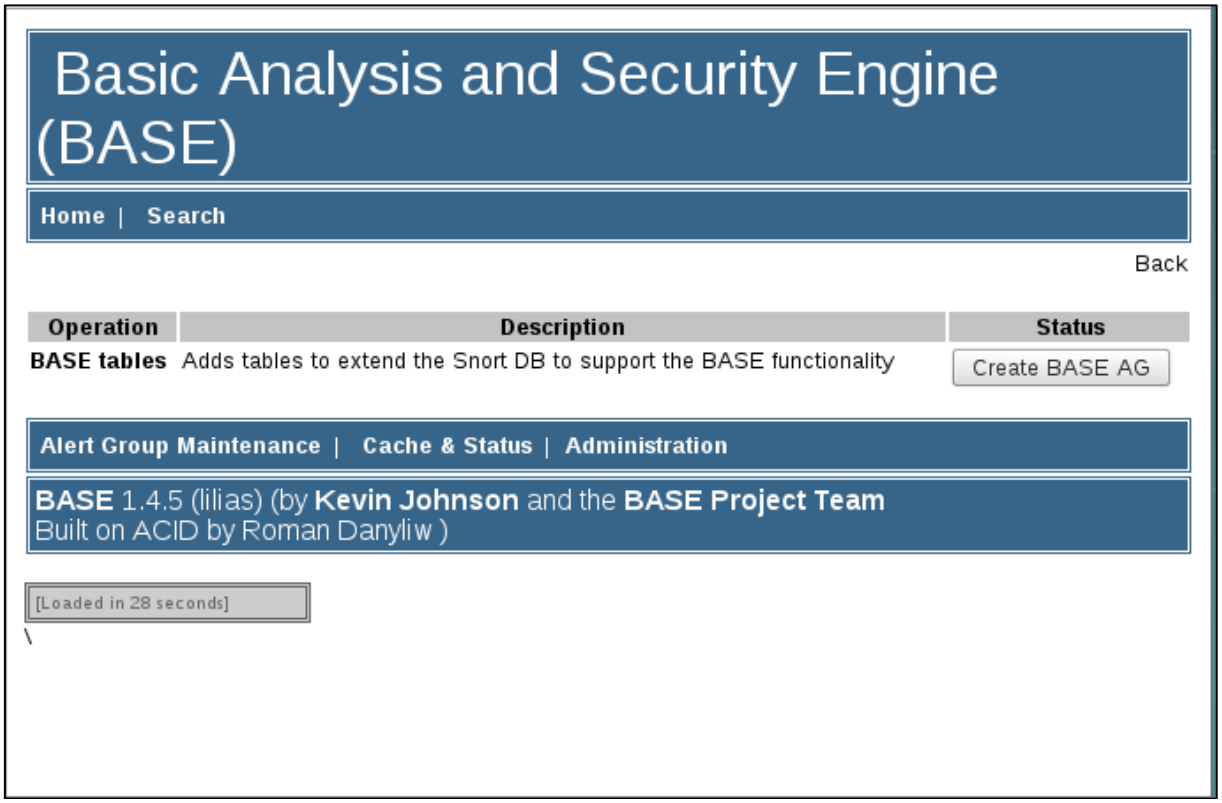
Figura 47 - Endereço de acesso no navegador



Fonte: Próprio autor

23. A seguinte tela será apresentada. Deve clicar em Create BASE AG:

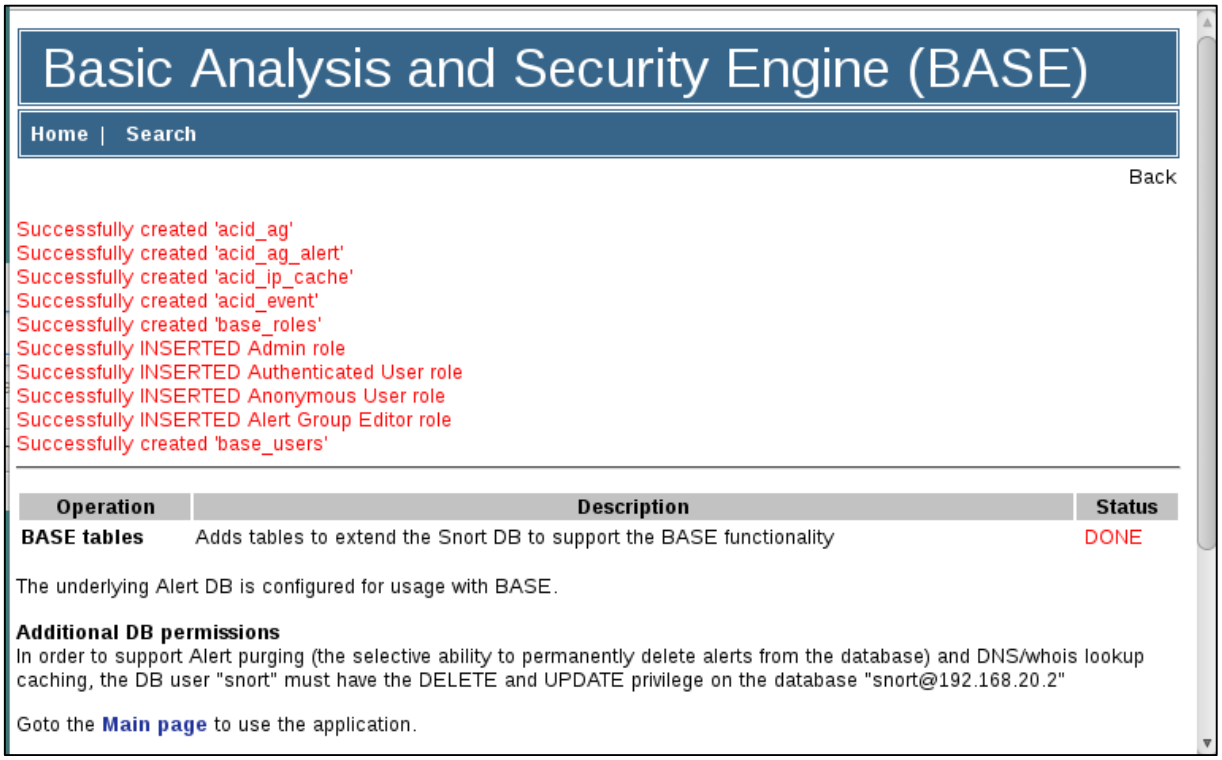
Figura 48 – Criação de tabelas do BASE



Fonte: Próprio autor

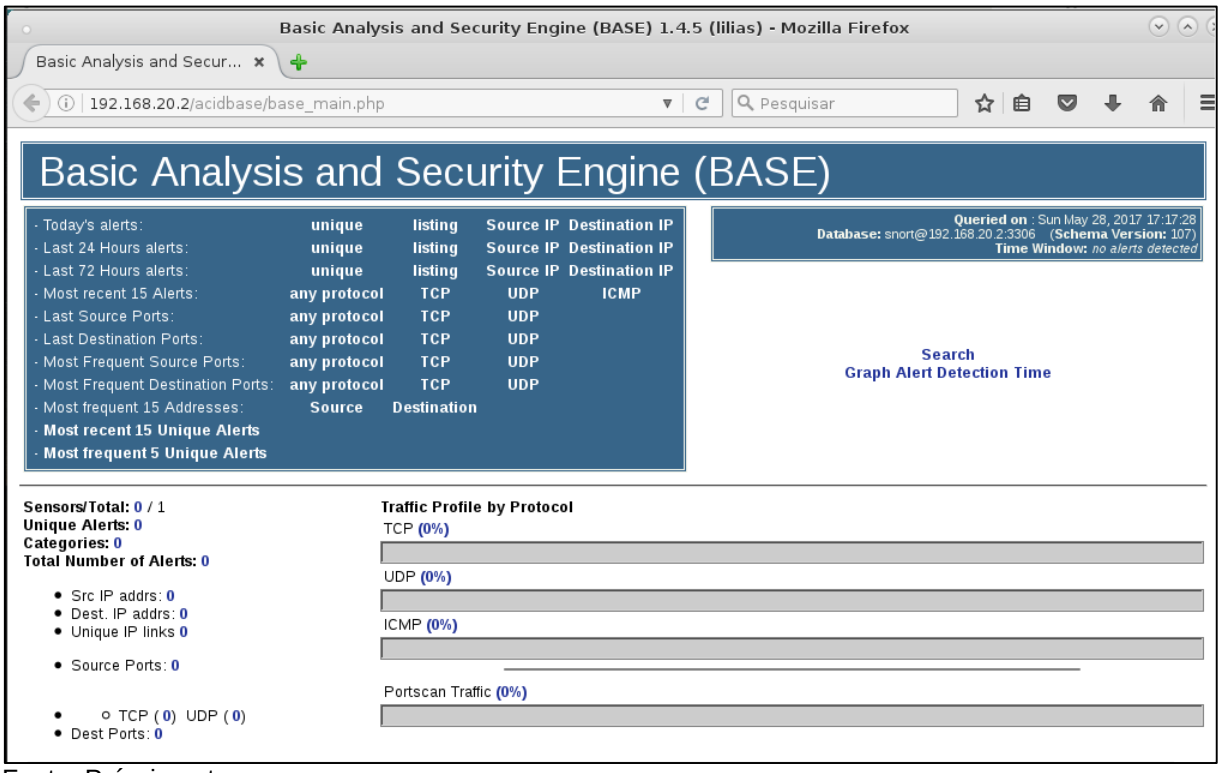
24. A seguir, a confirmação da criação. Deve clicar em Main page para seguir para a página principal:

Figura 49 - Confirmação de criação das tabelas



Fonte: Próprio autor

25. Página inicial do BASE:



Fonte: Próprio autor

APÊNDICE G – INSTALAÇÃO E CONFIGURAÇÃO DO HYDRA

1. Instalar o pacote do Hydra:

```
# apt install hydra
```

2. Realizar o download de um dicionário de senhas:

```
# wget
http://downloads.skullsecurity.org/passwords/cain.txt.bz2
```

3. Descompactar o arquivo:

```
# bunzip cain.txt.bz2
```

APÊNDICE H – COMUNICAÇÃO ENTRE IDS E IPFW

1. Instalar o pacote OpenSSH no Debian – SNORT:

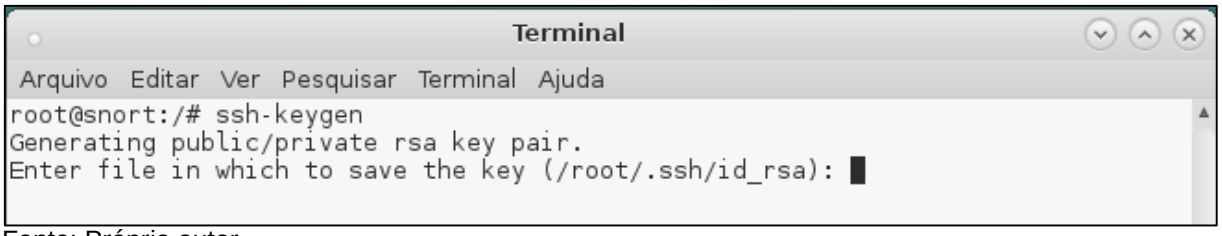
```
# apt install openssh-server openssh-client
```

2. Gerar chaves:

```
# ssh-keygen
```

3. Manter o caminho padrão para geração de chaves:

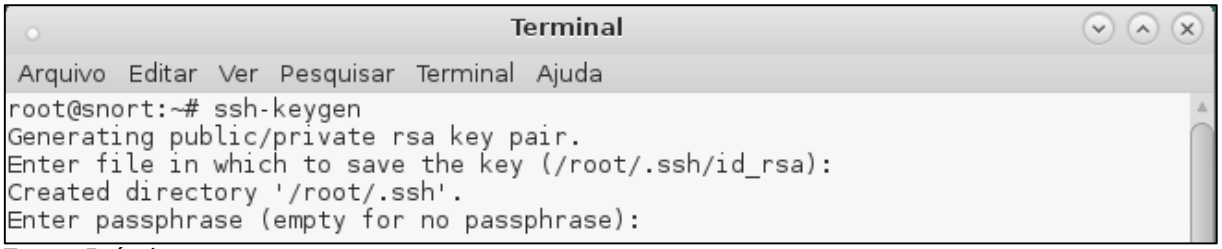
Figura 50 - Confirmação do caminho das chaves



Fonte: Próprio autor.

4. Manter sem senha:

Figura 51 - Senha para geração das chaves



Fonte: Próprio autor.

APÊNDICE I – CONFIGURAÇÂO SSH NO FREEBSD

Essa configuração foi realizada na máquina FreeBSD – Firewall. Ela é necessária para comunicação entre o Guardian e o Firewall. Por mais que algumas regras apresentem insegurança, o firewall só permite essa comunicação dentro da rede onde o Snort está localizado.

1. Verificar o status do serviço SSH na máquina:

```
# service sshd status
```

2. Caso o serviço esteja inativo, ele deve ser habilitado.

```
# service sshd enable
```

3. Abrir o arquivo rc.conf:

```
# ee /etc/rc.conf
```

4. Adicionar a linha abaixo para habilitar o serviço do SSH:

```
# ssh_enable = "YES"
```

5. Iniciar o serviço SSH:

```
# service sshd start
```

6. Abrir o arquivo de configuração do SSH:

```
# ee /etc/ssh/sshd_config
```

7. Descomentar as linhas:

```
Port 22

PermitRootLogin

PermitEmptyPasswords yes
```

8. Reiniciar serviço SSH

```
# service sshd restart
```

Posteriormente, foi configurado o SSH no FreeBSD – Servidor SSH.

1. Verificar o status do serviço SSH na máquina:

```
# service sshd status
```

2. Caso o serviço esteja inativo, ele deve ser habilitado.

```
# service sshd enable
```

3. Abrir o arquivo rc.conf:

```
# ee /etc/rc.conf
```

4. Adicionar a linha abaixo para habilitar o serviço do SSH:

```
# ssh_enable = "YES"
```

5. Iniciar o serviço SSH:

```
# service sshd start
```

6. Abrir o arquivo de configuração do SSH:

```
# ee /etc/ssh/sshd_config
```

7. Descomentar as linhas:

```
Port 22

PermitRootLogin no

PermitEmptyPasswords no
```

8. Reiniciar serviço SSH

```
# service sshd restart
```

REFERÊNCIAS BIBLIOGRÁFICAS

MENDES, Douglas Rocha. **Redes de computadores, teoria e prática**. 1º ed. São Paulo: Novatec, 2015.

TANENBAUM, Andrew S. **Redes de computadores**. 4º ed. Rio de Janeiro: Elsevier, 2003.

KUROSE, James F. e Keith W. Ross. **Redes de computadores e a Internet:** uma abordagem top down. 5º ed. São Paulo: Pearson Prentice Hall, 2010.

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

CERT.br, Cartilha de segurança para a Internet. Disponível em:< https://cartilha.cert.br/ataques/>. Acesso em: 02 de maio de 2017.

TRACANELLI, Patrick; GOTO Mauricio, IPFW FREBSD. Disponível em:< ftp://ftp.ige.unicamp.br/pub/documentos/IPFW.pdf>. Acesso em: 02 de maio de 2017

MONTE, Silvio do, SDI (IDS) com o SNORT, MySQL, PHP e BASE em 15 minutos. Disponível em:< https://www.vivaolinux.com.br/artigo/SDI-(IDS)-com-o-SNORT-MySQL-PHP-e-BASE-em-15-minutos>. Acesso em: 02 de junho de 2017.

PETERSON, Larry L; DAVIE Bruce S. **Redes de computadores**: Uma abordagem de sistemas. 2004. 3º ed. Rio de Janeiro: Elsiever.

KRUM, Snort + BarnYard2 + Snorby no Slackware 14.1. Disponível em:< https://www.vivaolinux.com.br/artigo/Snort-BarnYard2-Snorby-no-Slackware-141>. Acesso em: 03 de junho de 2017.

Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil, Ataques na Internet. Disponível em:< https://cartilha.cert.br/ataques/>. Acesso em: 04 de junho de 2017.

PANDINI, William, IDS: História, conceito e terminologia. Disponível em:< https://blog.ostec.com.br/seguranca-perimetro/ids-o-que-e-e-principais-conceitos>. Acesso em: 04 de junho de 2016.