CENTRO PAULA SOUZA

Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Segurança da Informação

Ricardo de Paula Cardoso

GNU/Linux CentOS e Hardening:
Blindagem de um Sistema Operacional

Americana, SP
2015

CENTRO PAULA SOUZA

Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Segurança da Informação

Ricardo de Paula Cardoso

ricardopcardoso@outlook.com.br

GNU/Linux CentOS e Hardening:
Blindagem de um Sistema Operacional

Trabalho de Conclusão de Curso, desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Segurança da Informação da Fatec-Americana, sob orientação da Prof. Me. Alexandre Garcia Aguado.

Área de Concentração: Segurança da Informação

Americana, SP
2015

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

C266g

Cardoso, Ricardo de Paula
GNU/Linux CentOS E Hardening: blindagem de um sistema operacional. / Ricardo de Paula Cardoso. – Americana: 2015.
55f.

Monografia (Graduação em Tecnologia em Segurança da Informação). - - Faculdade de Tecnologia de Americana – Centro Estadual de Educação Tecnológica Paula Souza.
Orientador: Prof. Me. Alexandre Garcia Aguado

1. Segurança em sistemas de informação I. Aguado, Alexandre Garcia II. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana.

CDU: 681.518.5

Ricardo de Paula Cardoso

GNU/Linux CentOS e Hardening: Blindagem de um Sistema Operacional

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, 12 de dezembro de 2015.

Banca Examinadora:

Profº Me. Alexandre Garcia Aguado (Presidente)
Mestre
FATEC Americana

Profº Me. Eduardo Antonio Vicentini (Membro)
Mestre
FATEC Americana

Profº Me. Clerivaldo José Roccia (Membro)
Mestre
FATEC Americana

AGRADECIMENTOS

Em primeiro lugar, gostaria de agradecer a Deus, pelo dom da vida.

Aos meus pais, que desde cedo, me ensinaram o caminho do trabalho honesto.

À minha família e meus amigos, que tiveram paciência e abriram mão da minha companhia por três anos.

Ao meu orientador e amigo, Alexandre Garcia Aguado, pela ajuda e confiança.

À Cristina Luz, minha grande amiga que foi mais que uma simples professora, e sim uma irmã, ou melhor, mãe para todos os alunos.

À Ana Carolina por todas as dúvidas que pude tirar referentes à sua experiência com este tipo de trabalho.

A todas as pessoas que trabalham ou estudam na Fatec de Americana por colaborarem na minha formação.

Aos meus amigos sem exceção, que me ajudaram e me mantiveram motivado.

DEDICATÓRIA

À minha família, aos meus amigos e a todos os meus mestres dos Ensinos Fundamental e Médio, que acreditaram e me incentivaram para que continuasse estudando.

RESUMO

Nos dias de hoje cada vez menos se vê um microcomputador sem acesso à Internet. Isto se deve a popularidade da Internet ao longo dos anos, e também ao fácil acesso à rede. Esta alta acessibilidade acaba prejudicando em alguns aspectos. Um deles é segurança, principalmente na falta ou redução dela devido à alta exposição dos Servidores e Microcomputadores à rede mundial. Devido a estes fatores, foram criadas e desenvolvidas diversas técnicas para reforçar a segurança dos servidores e suas informações. Este trabalho tem por objetivo apresentar uma forma de se melhorar o nível de segurança de servidores e das informações contidas nos mesmos. Com isso sugere-se a implementação da técnica de *Hardening*, que ensina maneiras de se reforçar a segurança de servidores e microcomputadores através de bloqueios e restrições de acesso. A devida aplicação foi realizada em uma distribuição *GNU/Linux CentOS*, que além de ser um dos melhores sistemas para servidores devido a sua robustez e segurança, também é gratuito. Para tal finalidade serão usadas certas ferramentas como *iptables,* entre outras.

Palavras Chave: *Hardening*, Segurança, Servidores.

ABSTRACT

Today is getting hard to see a computer without internet access. This is due to the internet access popularity. This high accessibility ends damaging some aspects, one of those being security. Mainly the lack or reduction of it, due to the high exposure of servers and PC's to the world wide web, in response to this factors, were created and develop several techniques to reinforce the security level of these servers and the information stored on it. With this, it was sugested to implement a Hardening technique, that teaches ways to reinforce security of servers thru access blocking and restriction. The aplication was done be realized on a GNU/Linux CentOS, that is one of the best systems for servers due to it's resilience and security, and also it is free. For this purpose it will be used tools like iptables, and others.

Keywords: *Hardening, Security, Servers.*

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

Cada vez mais as pessoas estão armazenando, acessando ou manuseando informações via formato digital. Através da pesquisa realizada pelo CETIC, em 2014/2015, fica claro o aumento do número de domicílios com computadores, conforme mostra a **Quadro 1**(CETIC.br, 2015).

Quadro 1 - PROPORÇÃO DE DOMICÍLIOS COM COMPUTADORES
Percentual sobre o total de domicílios / Base: 65.129.753 domicílios. Dados coletados entre outubro de 2014 e março de 2015.

Percentual (%)		Sim	Não
TOTAL		50	49
Área	**Urbana**	55	45
	Rural	23	76
Região	**Sudeste**	59	40
	Nordeste	37	62
	Sul	57	43
	Norte	33	66
	Centro-Oeste	48	52
Renda familiar	**Até 1 Salário Mínimo**	17	83
	Mais de 1 até 2 Salários Mínimos	37	62
	Mais de 2 até 3 Salários Mínimos	61	38
	Mais de 3 até 5 Salários Mínimos	77	23
	Mais de 5 até 10 Salários Mínimos	91	9
	Mais de 10 Salários Mínimos	96	4
Classe social	**A**	99	1
	B	85	15
	C	49	51
	DE	12	87

Fonte: (NIC.br, 2015).

Novamente é possível visualizar que as pessoas estão tendo acesso à Internet, embora este número ainda seja pequeno. Através da pesquisa realizada pelo CETIC, em 2014/2015, fica claro que as regiões geográficas mais desenvolvidas economicamente têm acesso à Internet, conforme mostra a **Quadro 2**(CETIC.br, 2015).

Quadro 2 - PROPORÇÃO DE DOMICÍLIOS COM ACESSO À INTERNET

Percentual sobre o total de domicílios / Base: 65.129.753 domicílios. Dados coletados entre outubro de 2014 e março de 2015.

Percentual (%)		Sim	Não
TOTAL		50	50
Área	**Urbana**	54	46
	Rural	22	78
Região	**Sudeste**	60	40
	Nordeste	37	63
	Sul	51	49
	Norte	35	65
	Centro-Oeste	44	56
Renda familiar	**Até 1 Salário Mínimo**	17	83
	Mais de 1 até 2 Salários Mínimos	37	63
	Mais de 2 até 3 Salários Mínimos	59	41
	Mais de 3 até 5 Salários Mínimos	76	24
	Mais de 5 até 10 Salários Mínimos	89	11
	Mais de 10 Salários Mínimos	95	5
Classe social	**A**	98	2
	B	82	18
	C	48	52
	DE	14	85

Fonte: (NIC.br, 2015).

Não é possível obter-se a ideia deste trabalho sem entender o que é e como começou a Internet, pois depois da popularização dela, foi necessário desenvolver métodos de proteção à informação contidas nos computadores.

De acordo com Damski e Valente (1996) a Internet é um conjunto de redes que realizam a troca das informações de forma padronizada, ou seja, é a junção das redes afim de disponibilizar todas as informações para todos, da forma mais democrática possível.

O surgimento se deu no ano de 1969 com o estabelecimento do protocolo TCP/IP (*Transfer Control Protocol* / *Internet Protocol*) cuja finalidade era prover a comunicação entre computadores e redes de pesquisa do Departamento de Pesquisa (DOD) dos Estados Unidos da América. A principal característica da ARPANET era seu uso por Instituições de Ensino e Pesquisa, Militares, entre outros, onde a Internet somente existia para estes, ficando restritas somente a estes (DAMSKI; VALENTE, 1996).

Aos poucos as instituições de ensino e pesquisa foram ingressando, fazendo com que o conjunto de redes ficasse cada vez maior, e isso fez com que ela nascesse e crescesse. Sendo assim, não existe uma única entidade que gerencie e/ou se responsabilize por toda a rede, e isso que a faz democrática, pois tudo nela é feito através de comum acordo, inclusive seus padrões.

Após a primeira demonstração da ARPANET assim como ficou conhecida, ocorreu uma grande revolução onde após nove anos do lançamento da mesma haviam dezenas de *Hosts* interconectados. Devido a este crescimento a rede tornou-se o meio favorito de pesquisadores, professores e estudantes de se comunicarem por meio de computadores. Com o tempo a Internet tornou-se uma rede global, ou seja, conectando cada vez mais redes por todas as partes do globo. Uma particularidade importante é que a Internet nunca para, nunca desliga (DAMSKI; VALENTE, 1996).

Segundo Castells (2003) no início dos anos 90 a ARPANET foi retirada de operação devido ao fato de estar obsoleta. Este fato fez com que a Internet saísse do espectro militar, o que deu início a sua popularização.

Devido a este fato, diversos provedores de serviços de Internet montaram redes próprias e estabeleceram suas próprias portas de comunicação em bases comerciais. Isto fez com que a Internet crescesse rapidamente, ou seja, se tornasse global, e isso só foi possível através do projeto original da ARPANET.

O estopim da Internet no que se refere à abrangência no mundo todo ocorreu, após o desenvolvimento da www (World Wide Web) aplicação de compartilhamento de informação desenvolvida pelo inglês Tim Berners-Lee em 1990, e com a disponibilização do Navegador de Internet Netscape Navigator em 1995. E a partir

deste que ela nasceu para a maioria das pessoas, empresários e para a sociedade em geral.

A partir da popularização da Internet, desta alta disponibilidade se fez necessário aumentar a Segurança de nossos dados e informações. Parte dos dispositivos eletrônicos como *smartphones*, televisores, *tablets*, *notebooks* entre outros, tem capacidade de se conectar à internet, embora nem todas as classes possuam acesso à Internet conforme a **Quadro 2**, esta realidade vem mudando cada vez mais.

Atualmente, são muitas as pessoas que possuem acesso à internet, deve-se lembrar sempre, que a internet além de facilitar de várias maneiras para a realização de determinadas atividades, também viabiliza uma maior exposição aos dados e consequentemente uma maior exposição à alguns riscos. A exposição de servidores e microcomputadores a rede mundial causa vulnerabilidades que podem vir a ser exploradas por *crackers*[1] com o intuito de furtar informações importantes para práticas de crimes contra as vítimas. O crescimento continuo de pessoas com acesso à rede, fez crescer também a necessidade de se desenvolver técnicas de segurança que ajudam a prevenir tentativas de invasão e reforçar a segurança de computadores e servidores, principalmente as informações importantes contidas nos mesmos.

O panorama acima citado faz com que os administradores de redes, desenvolvedores de sistemas e outros profissionais da Computação desenvolvam, constantemente, novas ferramentas indispensáveis para bloquear acessos indevidos ao conteúdo existente nos computadores. Desta forma, nasce o *Hardening*, cujo tem sido cada vez mais implementado.

Segundo a definição de Skoudis (2012) *Hardening* é um processo de mapeamento de ameaças, atenuação de riscos e execução de atividades corretivas, com foco na infraestrutura e com o principal objetivo de preparar o ambiente alvo

[1] Cracker – Conhecidos também como “Hacker do mal” ou “Hacker sem ética” são pessoas que utilizam de seus conhecimentos para invadir sites e computadores sem autorização, com o intuito de causar vandalismo ou furto de informações (ULBRICH, 2011).

para enfrentar determinadas tentativas de ataque ou violação dos protocolos de segurança da informação.

Dentre as principais características desta técnica estão: Mitigação de Riscos com objetivo de manter a Integridade, Confiabilidade e Disponibilidade dos Sistemas Operacionais; Reduzir o Risco de Invasões; Desabilitar serviços desnecessários; Fortalecer senhas e serviços e preparar o ambiente caso ocorra um eventual ataque/invasão.

Este trabalho tem como intuito principal contribuir acerca da seguinte problemática:

Quais as vantagens, desvantagens e características da implementação de técnicas de Hardening em servidores GNU/Linux?

Neste trabalho, o objetivo é a apresentação da técnica Hardening, no sistema operacional *GNU/Linux CentOS*, que ensina maneiras de se reforçar a segurança de servidores e microcomputadores através de bloqueios e restrições de acesso, a devida aplicação foi realizada em uma distribuição *GNU/Linux*, será citado este nas seções posteriores.

Para compreender as vantagens, desvantagens e características da implementação de *Hardening*, serão realizados testes através de um estudo prático. Será apresentado um ambiente controlado onde a técnica será aplicada nos serviços mencionados através das seções deste trabalho, usando como ferramenta o *iptables*, onde ele será responsável pelo bloqueio e liberação das portas dos serviços em que serão realizados o *Hardening*.

Também será realizado o experimento (Teste da aplicação de *Hardening*), verificando se as alterações realizadas surtiram efeito.

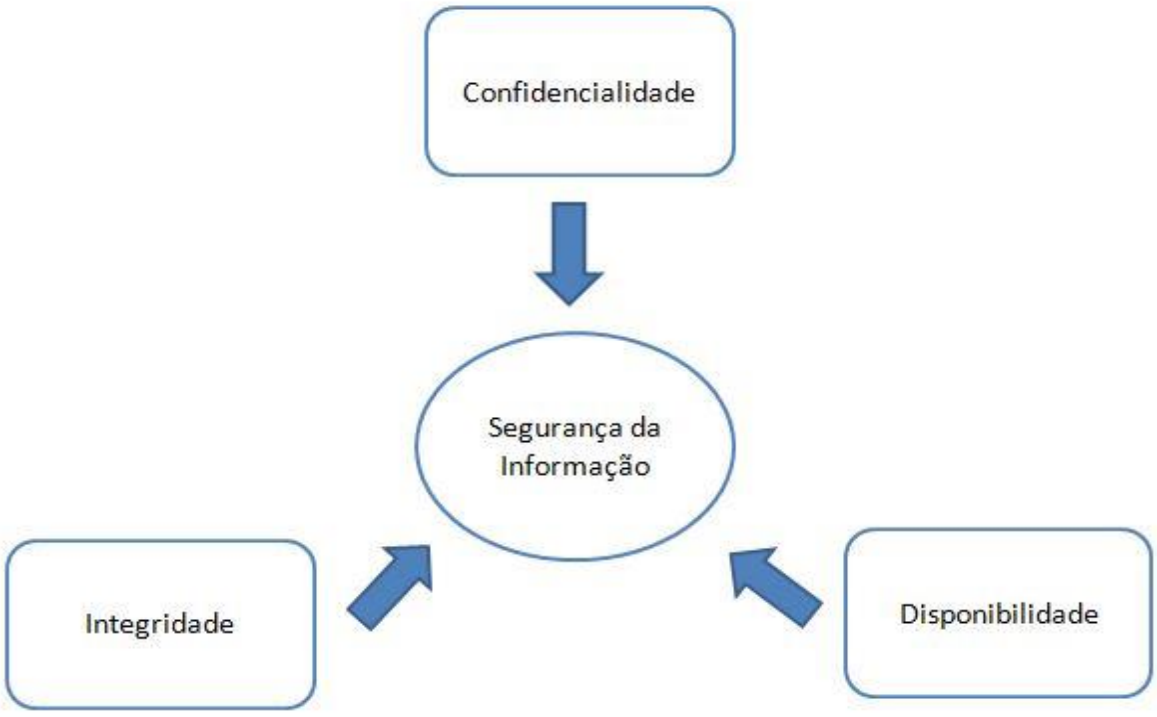
2 SEGURANÇA DA INFORMAÇÃO

Segundo Dantas (2011) as empresas dedicam atenção especial as suas informações, essas informações são caracterizadas como ativos para as empresas, devido ao alto valor que elas possuem para continuidade de negócios.

É fundamental que as empresas adotem medidas que garantam a segurança de suas informações. A Norma ABNT NBR ISO/IEC 27001 (2006) conceitua segurança da informação como a preservação da Confidencialidade, Integridade e Disponibilidade das informações.

Para que uma empresa possa garantir realmente a segurança de suas informações como dito anteriormente, há necessidade de que as mesmas adotem os princípios básicos de segurança, que garantem o sigilo, a veracidade e disponibilidade das informações. A **Figura 1** apresenta os três pilares da segurança da informação.

Figura 1 - Os três pilares que garantem a Segurança da Informação



Fonte: Baseado em Goodrich e Tamassia (2013)

Os três pilares da segurança da informação, são considerados os princípios básicos para se garantir a segurança de informações, Goodrich e Tamassia (2013) conceituam os mesmos da seguinte maneira:

- Confidencialidade: A confidencialidade tem como objetivo evitar a revelação de informações não autorizadas. A confidencialidade trabalha com a proteção de dados, ela garante acesso a pessoas autorizadas ao mesmo tempo em que não permite que usuários não autorizados tenham acesso a informações sigilosas. Manter as informações secretas normalmente é a essência da segurança da informação. Algumas das ferramentas utilizadas para garantir a confidencialidade atualmente são: Encriptação, controle de acesso, autenticação e autorização.

- Integridade: A integridade tem como objetivo garantir que as informações não foram alteradas por usuários não autorizados, ela também impede que os usuários que tem acesso a essas informações por ventura realizem modificações sem autorização. As ferramentas usadas para garantir confidencialidade também são eficientes para garantir integridade. Além delas existem ferramentas exclusivas para garantir integridade, algumas dessas são: Cópias de segurança, somas de verificação (*checksums*) e códigos de correção de dados.

- Disponibilidade: A disponibilidade visa garantir que os usuários devidamente autorizados possam acessar e modificar informações, a qualquer momento, ou seja, ela deve ser confidencial e integra, porém de fácil acesso aos usuários autorizados. As principais ferramentas que auxiliam na garantia da disponibilidade e da segurança das informações são: Proteções físicas e redundância computacional (GOODRICH; TAMASSIA 2013).

Além do tripé da segurança da informação existem também alguns outros princípios que ajudam a garantir a segurança destas, são eles: Autenticidade e Não-repúdio.

- Autenticidade: A autenticidade visa assegurar que um usuário é realmente quem diz ser.

- Não-repúdio: É a garantia de que um usuário não possa negar a autoria de algum documento. Porém, este princípio de segurança só é valido se trabalhar juntamente com a autenticidade e a integridade, pois a autenticidade mostra quem criou a mensagem e a integridade prova que a mesma não sofreu alterações.

2.1 HARDENING

Segundo a definição de Skoudis (2012) *Hardening* é um processo de mapeamento de ameaças, atenuação de riscos e execução de atividades corretivas, com foco na infraestrutura e com o principal objetivo de preparar o ambiente alvo para enfrentar determinadas tentativas de ataque ou violação dos protocolos de segurança da informação.

Dentre as principais características desta técnica estão: Mitigação de Riscos com objetivo de manter a Confidencialidade, Integridade e Disponibilidade dos Sistemas Operacionais; Reduzir o Risco de Invasões; Desabilitar serviços desnecessários; Fortalecer senhas e serviços e preparar o ambiente caso ocorra um eventual ataque/invasão.

De acordo com Julio, Reis e Verbena (2011) *Hardening* é uma técnica genérica, visto que esta pode ser aplicada em qualquer sistema operacional. Se tratando do *GNU/Linux* é possível obter um nível bastante satisfatório de segurança, através da implementação da técnica. Para tal deve-se considerar três grandezas: Segurança, Risco e Flexibilidade.

O profissional de Segurança deverá balancear de forma correta estas grandezas, a fim de que o sistema tenha uma alta produtividade e tenha segurança. Pois quanto maior a Segurança, menor o Risco e menor a Flexibilidade, e vice-versa.

De acordo com Julio, Reis e Verbena (2011), existem algumas utilizações de *Hardening* muito uteis na configuração de sistemas operacionais, como por exemplo, Particionamento de discos, Analisar e desabilitar serviços desnecessários e

inseguros, Analise de senhas fracas e implementação de senhas fortes, Desabilitar Shell/Remover usuários inválidos, Desconexão de usuários não autorizados, Implementação de senha criptografada no GRUB (no caso do Linux), Política de segurança para utilização de serviços de Rede, Gerenciamento de privilégios, Segurança no terminal/estação de trabalho (*Timeout* por inatividade), Busca/Bloqueio de portas abertas, entre outras.

Julio, Reis e Verbena (2011), mencionam algumas situações para qual o *Hardening* é recomendado:

> *Hardening* pode ser utilizado para evitar que usuários mal-intencionados aproveitem da ausência do administrador e implantem scripts maliciosos em servidores infectando toda a rede, bloquear que o usuário administrador faça *login* diretamente no terminal, efetuar *logout* por tempo de inatividade, remover pacotes que não são utilizados, remover permissões especiais de binários executáveis, dentre outras técnicas que serão apresentadas posteriormente. Julio, Reis e Verbena (2011).

2.2 VULNERABILIDADES, RISCOS, ATAQUES E OS TIPOS DE ATAQUES

De acordo com CERT.BR (2012) existem inúmeros motivos que levam um indivíduo a realizar ataques em um computador ou rede de computadores, esses motivos variam muito desde ataques por simples diversão, que não deixa de ser criminosos, até mesmo ataques por motivos criminosos, onde o intuito é o furto de informações, os motivos de ataque mais comuns são:

- Demonstração de poder: Neste tipo de ataque é comum que o atacante tente invadir uma rede de computadores com o objetivo de mostrar vulnerabilidades na mesma e assim oferecer através de ameaças serviços de proteção para que este tipo de invasão não volte a ocorrer.
- Prestigio: Este ataque pode ocorrer mais comumente em empresas com nível elevado de segurança, os atacantes fazem espécies de disputas para vangloriar se por ter conseguido realizar o ataque em uma empresa com nível elevado de segurança.

- Motivações financeiras: Em ataques com motivações financeiras, os atacantes têm como objetivo conseguir informações pessoais dos usuários para aplicar golpes.
- Motivações ideológicas: Neste tipo de ataque o atacante tem por objetivo tirar sites com conteúdo que vai contra a sua opinião fora do ar.
- Motivações comerciais: Esse ataque normalmente é realizado por empresas concorrentes, com o objetivo de derrubar a imagem da empresa perante os clientes.

Os atacantes utilizam inúmeras técnicas para realização de ataques como os descritos acima. Em sequência são apresentadas algumas destas técnicas.

A primeira a ser citada é uma das técnicas abordadas por este trabalho. Pois o *Hardening* visa mitigar as chances de uma possível Exploração de vulnerabilidades. Pode-se descrever vulnerabilidades como falhas em um sistema, que possam ter passado despercebidas durante a elaboração de um projeto. Essas falhas fazem com que o sistema fique mais vulnerável a ataques criminosos. Os atacantes exploram essas falhas com o objetivo de praticar crimes como, por exemplo: furto de informações sigilosas, ataques contra outros computadores ou para tirar serviços do ar.

Uma outra técnica que pode ser citada, é a técnica de Varreduras em redes (*Scan*). O *Scan* de redes é uma técnica comum de ataque, que consiste em varreduras minuciosas de redes para encontrar computadores ativos e fazer uma coleta de informações sobre os mesmos, como portas abertas, por exemplo, que facilitam a verificação de vulnerabilidades que possam ser exploradas. O *scan* de redes nem sempre é realizado por indivíduos com intenções maliciosas, pode ser feito também por pessoas autorizadas com o objetivo de encontrar possíveis vulnerabilidades e corrigi-las justamente para que as mesmas não sejam exploradas por atacantes.

Um outro exemplo a ser citado é Falsificação de *email* (*E-mail spoofing).* A falsificação de *e-mail* é uma técnica de ataque muito comum, a partir dela os atacantes são capazes de alterar os campos referentes ao cabeçalho do e-mail, isso faz com que pareça que um indivíduo enviou um *e-mail* quando na verdade esse *e-*

mail vem de outro lugar. Este ataque é possível porque os endereços de *e-mail* utilizam o protocolo SMTP[2], que permite a alteração dos dados do cabeçalho do *e-mail*. Estes *e-mails* normalmente são enviados contendo códigos maliciosos como o envio de mensagens de *Spam* e golpes de *phishing,* alguns exemplos comuns deste tipo de ataque são *emails* recebidos supostamente de entidades bancarias pedindo confirmação de dados, e *email* de conhecidos pedindo para clicar em *links* e baixar arquivos estranhos, em grande maioria esses podem ser considerados *emails* falsos contendo arquivos maliciosos, e tentativas de conseguir dados pessoais dos usuários.

Pode-se citar também interceptação de trafego (*Sniffing*). Com esta técnica é possível inspecionar os dados que trafegam na rede através de programas denominados *Sniffers*, os atacantes conseguem capturar senhas, dados de cartão de crédito e outras informações importantes que trafegam na rede sem criptografia.

Tem-se aqui a técnica principal deste trabalho, pois através desta técnica e para proteger desta foi desenvolvido todo este trabalho. Esta técnica é chamada de Força bruta (*Brute force*). Ataque de força bruta são ataques de tentativa e erro, onde o atacante tenta adivinhar o nome de usuário e a senha das vítimas, com isso o atacante pode executar tarefas, acessar sites, com todos os privilégios daquele usuário. Se um atacante consegue as informações de nome e senha de um usuário ele pode praticar diversas ações maliciosas como trocar a senha para impossibilitar o acesso do usuário, acessar e-mails e enviar conteúdo malicioso, e mensagens falsas, são inúmeras as possibilidades de ações que ele pode realizar. Mesmo que um atacante com consiga descobrir a senha do usuário isso pode causar danos, pois algumas contas costumam ser bloqueadas depois um número de tentativas de acesso sem sucesso.

Outros dois exemplos a serem citados são, Desfiguração de página (*Defacement*) e Negação de serviço (DOS e DDoS). Por meio da técnica Desfiguração de página os atacantes fazem alterações em sites, normalmente através de falhas no servidor web e erros de aplicação web. Na Negação de serviço, os ataques de negação de serviço não têm como objetivo coletar informações sigilosas, este é utilizado como técnica para tirar serviços, computadores ou até

[2] SMTP: Simple Mail Transfer Protocol é o protocolo padrão para envio de e-mails através da Internet.

mesmo uma rede de computadores ligadas na Internet fora do ar, quando vários computadores são utilizados para realizar um ataque ele recebe o nome de ataque de negação de serviço distribuído. Esse ataque pode ser realizado quando o computador atacante envia uma quantidade grande de requisições para um determinado serviço, tornando o mesmo temporariamente indisponível.

2.2.1 Roteiro Organizado de Ataque

De acordo com Macedo (2012) ao planejar um ataque, deve ser seguido um plano de estratégia, de acordo com o alvo desejado. Um cracker experiente nesta área, estando a fim de alcançar seu objetivo, sempre traça um roteiro a ser seguido religiosamente. Um bom exemplo de roteiro organizado para ataques, é exemplificado abaixo:

- Localizar o alvo desejado;
- Concentrar o máximo de informações possíveis sobre o alvo, geralmente utilizando alguns serviços da própria rede, ou até mesmo, ferramentas utilizadas na administração e gerenciamento da rede alvo;
- Disparar o ataque sobre o alvo, a fim de invadir o sistema, explorando a vulnerabilidade do sistema operacional, servidores e serviços oferecidos pela rede. O invasor pode até mesmo abusar um pouco da sorte tentando adivinhar uma senha na máquina alvo, fazendo combinações possíveis;
- Não deixar pistas da invasão, pois geralmente as ações realizadas pelos invasores são registradas no sistema alvo em arquivos de log, possibilitando que o administrador do sistema invadido possa vir a descobrir a invasão, a não ser que o invasor se preocupe em eliminar todos e quaisquer vestígios que o incriminem;
- O invasor deve conseguir não somente senhas de usuários comuns, pois os privilégios destes usuários são limitados, não dando acesso a recursos mais abrangentes no sistema. Com isso, é de grande importância que o invasor consiga uma senha de administrador, pois terá todos os recursos de gerenciamento do sistema disponíveis, para alterações e até mesmo gerar bug no sistema. Instalar ferramentas que façam a captura de senhas de forma clandestina aos olhos do administrador, para isso existem programas que conseguem rodar em segundo plano sem que a vítima perceba, podendo ser colocados na pasta usada pela vítima;

- Criar caminhos alternativos de invasão, logo que a administradora do sistema encontrar uma “porta aberta” que permita a invasão esta será fechada, mas se o invasor gerar outras maneiras de invadir o sistema, certamente terá outra chance de invasão, já que teve a preocupação de criar novas rotas alternativas;

- Utilizar a máquina invadida como “portão de entrada” para invasão de outras máquinas da rede e até mesmo do computador central.

Macedo (2012).

3 GNU/LINUX E GPL

Segundo Sabino (2011), GNU acrônimo de *GNU is Not Unix* (de forma traduzida GNU não é Unix) é o nome do projeto de Richard Stallman atualmente mantido pela Free Software Foundation (FSF), o projeto foi lançado em 27 de setembro de 1983. O objetivo era criar um sistema chamado GNU, este seria compatível ao Unix, porém completamente baseado em Software Livre, Stallman incluiu neste novo sistema alguns componentes como, por exemplo, o núcleo do sistema operacional, compilador C e editor de texto.

Foi criado pelo projeto a grande maioria dos programas livres atualmente utilizados principalmente nas Distribuições Linux. Em uma das subseções abaixo está sendo descrita a definição de distribuição Linux.

Em 1984 Stallman largou seu emprego para dedicar-se inteiramente ao seu projeto, e também para não correr o risco de perder os direitos de seu trabalho.

Sabino (2011) diz também que o grande mérito do projeto tem sido a criação dos programas citados acima e também a explanação da ideia de *Software* Livre, Stallman acredita que se ele gosta de um programa ele deve compartilhá-lo com outras pessoas que também gostem. Esta ideia concede aos usuários dos mesmos, quatro liberdades, sendo elas:

1 Liberdade para executar o programa como quiser;

2 Liberdade para realizar quantas cópias julgar necessário;

3 Liberdade de realizar modificações no programa a próprio gosto;

4 Liberdade para distribuir versões melhoradas.

Embora a FSF publique diversas licenças com a ideia de preservar a liberdade do software, a principal licença da fundação é a GNU GPL-GNU General Public License, visto que ela é usada pela maioria dos softwares do projeto GNU, assim como outro que não são parte dele.

A própria fundação sempre faz questão de mencionar que o livre se refere a liberdade e não a isenção do custo, ou seja, o acesso ao código fonte é liberado.

Linux é o núcleo ou *kernel* de um sistema operacional gratuito, desenvolvido e lançado ao mundo pela primeira vez por Linux Benedict Torvalds, em 1991 (BALL e DUFF, 2004). Torvalds era aluno da Universidade de Helsinki situada na Finlândia, e neste ano ele começou a desenvolver um Sistema Operacional (*kernel)* Livre, tendo como base o Minix. Sendo assim o Linux (*Kernel)* é um programa que é carregado no inicializar do computador, ou seja, é ele que provê a interface entre o que é exibido ao usuário e o *hardware.*

Segundo Ball e Duff (2004) o Linux é um Sistema Operacional livre, devido à autorização que há em se realizar as modificações desejadas no sistema, e gratuito devido ao fato de não poder ser cobrado nenhum valor em relação ao Sistema Operacional. Para tal, milhões de desenvolvedores situados no mundo todo colaboram na manutenção e evolução do Linux.

De acordo com Ball e Duff (2004) uma distribuição Linux é um conjunto de aplicativos e/ou utilitários rodando sobre um *kernel* Linux. A combinação do Linux (Kernel), com *drivers*, utilitários, interfaces com o usuário, entre outros softwares gera uma distribuição Linux, e é esta junção que faz com que ele fique da forma como se está habituado a ver. De maneira geral uma distribuição Linux é composta por alguns aplicativos como *player* de música, gravador de DVDs, suíte de escritório, entre outras.

Estas possuem características que refletem a necessidade e filosofia da comunidade, que conforme citado anteriormente é quem ajuda na manutenção e evolução das distribuições. Elas também são chamadas de *distros* ou “sabores” do Linux.

Alguns exemplos de distribuições não derivadas existentes são Red Hat[3], Debian[4], Slackware[5].

E alguns exemplos de distribuição derivada (que é baseada em um sistema operacional que já existe) são elas Fedora[6], Ubuntu[7].

---

[3] Disponível em: <https://www.redhat.com/pt-br>. Acesso em: 12 set. 2015
[4] Disponível em: <https://www.debian.org/index.pt.html>. Acesso em: 12 set. 2015
[5] Disponível em: <http://www.slackware.com>. Acesso em: 12 set. 2015
[6] Disponível em: <https://getfedora.org/pt_BR>. Acesso em: 12 set. 2015
[7] Disponível em: <http://www.ubuntu.com>. Acesso em: 12 set. 2015

Segundo Ball e Duff (2004) existem grandes vantagens em se utilizar Linux. Várias delas podem ser mencionadas. O Linux é um ótimo custo benefício, devido ao fato de o Linux fornece um excelente retorno sobre o investimento, isto tudo deve-se ao fato de o Linux ter pouco ou nenhum custo por máquina, além de não ter taxas de *royalties* ou licenciamento, sendo assim uma única cópia de uma distribuição Linux pode ser utilizada infinitamente. O Linux é um excelente Desktop, também é uma excelente plataforma de Servidor. Veloz, seguro (devido à maioria do vírus desenvolvidos serem para os Sistemas Windows), estável e robusto. A grande maioria dos Banco de Dados, tem versões disponíveis para Linux. O Linux aumenta a vida útil do *Hardware,* isto deve-se ao fato de que o Linux é desenvolvido para rodar em diversos tipos de computadores (CPUs), assim economizando dinheiro para usuários domésticos, pequenas e microempresas e até corporações. O Linux é uma boa plataforma de desenvolvimento, isto se deve ao baixo custo para realizar o desenvolvimento de softwares no Linux. O Linux possui uma plataforma isenta de *Royalties* para desenvolvimento de Softwares em diferentes linguagens. Assim o Linux torna-se uma boa opção para desenvolvedores iniciantes e novas empresas cujo foco é em tecnologia.

3.1.1 Por que usar CentOS?

De acordo com Morimoto (2008), o CentOS é uma distribuição gratuita, gerada a partir do código fonte disponibilizado pela Red Hat, Inc. Esta distribuição é praticamente uma versão gratuita do Red Hat Enterprise Linux (RHEL), onde esta possui um excelente histórico de segurança, um excelente suporte comunitário e atualizações pontuais de segurança. Esta é bastante utilizada por empresas de hospedagem e bastante indicada para quem necessita rodar softwares como Oracle, por exemplo, que somente são suportados nele, e não pode pagar caro pelo sistema.

Segundo Morimoto (2008), há pequenas diferenças entre o CentOS e o Red Hat Enterprise Linux (RHEL):

> A grande diferença entre o CentOS e o Red Hat Enterprise é a questão do suporte, já que, embora caro, o suporte oferecido pela Red Hat é bastante personalizado e os profissionais passam por um exame de certificação exigente (o RHCE) que mistura testes teóricos e práticos. Em servidores de missão crítica, usar o Red Hat Enterprise e pagar pelo suporte é geralmente uma boa opção, já que além de ajuda na implementação, você tem uma equipe pronta para agir em caso de problemas inesperados. Para os demais casos, você pode perfeitamente utilizar o CentOS contando com o suporte comunitário oferecido através dos fóruns do projeto. Morimoto (2008).

Visto que todos os materiais disponibilizados pela Red Hat, Inc. e por terceiros para o Red Hat Enterprise Linux (RHEL) e baseado nestes argumentos, foi feita a escolha desta distribuição para a realização deste trabalho.

4 SERVIÇOS E FUNCIONALIDADES DO LINUX

Neste capitulo serão abordados alguns serviços e ferramentas essenciais em ambientes GNU/Linux, os quais, posteriormente serão alvo de estudo no caso prático apresentado neste trabalho.

4.1 SSH E ATAQUES À SSH

De acordo com Red Hat, Inc. (2005) SSH (ou Secure SHell) é um protocolo que facilita a comunicação segura entre dois sistemas usando uma arquitetura cliente / servidor e permite que os usuários se conectem a um host remotamente. Ao contrário de outros protocolos de comunicação à distância, como FTP[8] ou Telnet[9], SSH criptografa a sessão de *login*, tornando impossível para intrusos para coletar senhas não criptografadas.

SSH foi projetado para substituir sistemas defasados, sistemas menos seguros para fazer *logon* remotamente em outro sistema através de um comando *shell* tais como métodos de telnet ou rsh. Um programa relacionado chamado scp substitui programas mais antigos projetados para copiar arquivos entre hosts, como o rcp. Como estes aplicativos mais antigos não criptografam senhas entre o cliente e o servidor, é bom evitá-los. Usando métodos seguros para fazer *login* em sistemas remotos diminui os riscos de segurança, tanto para o sistema cliente e no sistema remoto.

O protocolo SSH oferece as seguintes garantias:

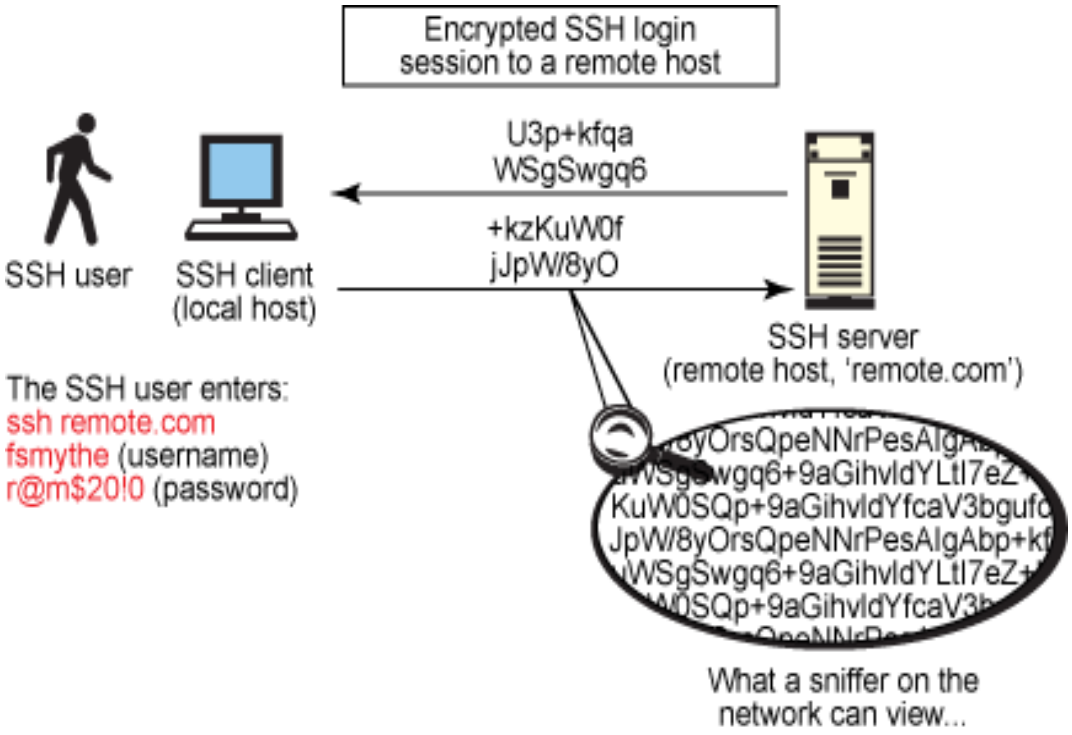
- Após a conexão inicial, o cliente pode verificar se ele está se conectando ao mesmo servidor conectado a ele anteriormente.
- O cliente transmite suas informações de autenticação para o servidor usando a criptografia forte de 128 bits.

[8] FTP é um protocolo seguro para transferência de arquivos em redes TCP/IP (MORIMOTO, 2005)
[9] Telnet é um programa para acesso remoto, cujo tráfego não é criptografado (MORIMOTO, 2005)

- Todos os dados enviados e recebidos durante uma sessão são transferidos usando criptografia de 128-bit, tornando as transmissões interceptadas extremamente difíceis de decifrar e ler.

Para melhor explanar como funciona uma conexão SSH, segue abaixo **Figura 2**.

Figura 2 – Representação de uma Conexão via SSH



Fonte: IBM

Como o protocolo SSH criptografa tudo que envia e recebe, ele pode ser usado para proteger protocolos inseguros. O servidor SSH pode tornar-se um canal para garantir protocolos inseguros usando uma técnica chamada de encaminhamento de porta, aumentando a segurança geral do sistema e de dados.

Segundo Murilo (2007) devido ao SSH proporcionar uma maior segurança os administradores de redes têm sido negligentes no que se refere à escolha de senhas fortes, ou seja, elas têm sido escolhidas de forma que não são suficientemente fortes. Este fato faz com que estas senhas estejam sujeitas a adivinhação e ataques de força bruta (*Brute Force*). Segundo ainda Murilo (2007) este tem sido o objetivo dos ataques de dicionário de senhas, ou seja, descobrir senhas fracas.

De acordo ainda com Murilo (2007) tem sido observado que tem ocorrido com mais frequência tentativas de ataque à SSH via dicionário de senhas, e estes além de estarem disponibilizados na Internet, se encontram disponíveis em diversos idiomas, inclusive em Português do Brasil.

Murilo (2007) cita diversas formas para se defender contra estes ataques. De acordo com este autor, são elas:

- Mover o serviço para uma porta não padrão:
  Embora não seja considerada uma medida corretiva, a "segurança por obscuridade" é viável devido ao fato de que os ataques ao *ssh* tendem a buscar somente a porta padrão (22/*tcp*).
- Limitação de Recursos:
  A limitação da utilização dos recursos do serviço no caso do OpenSSH, consiste em alterar as cláusulas MaxStartups e MaxAuthTries, que são, respectivamente, a quantidade de instâncias não autenticadas simultâneas (por padrão são 10), e quantas tentativas sem sucesso são permitidas por usuário (por padrão são 6).
- Limitar início de conexão (SYN):
  Algumas ferramentas de ataque, para ganhar eficiência, disparam várias cópias de si mesmas, sendo assim se faz necessário configurar no firewall[10] alguns limitadores para estes tipos de situações.

[10] 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. Kurose (2010, p.535-536)

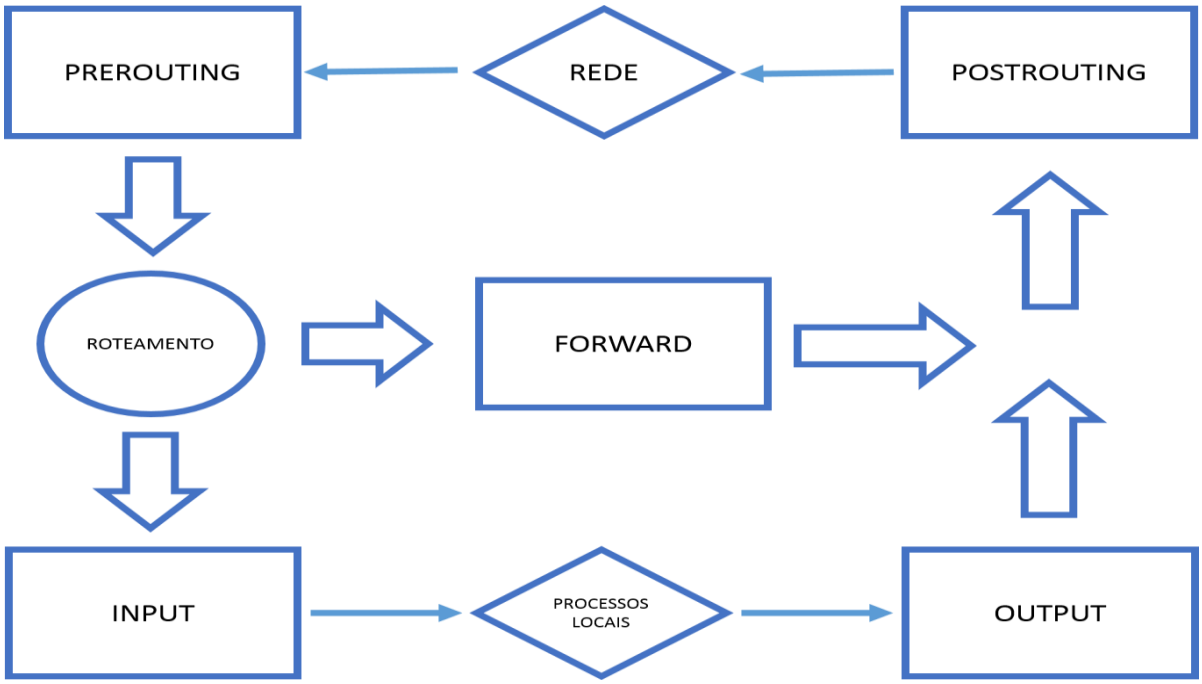
4.2 *IPTABLES/NETFILTER*

De acordo com D'Oliveira Neto (2004) *Netfilter* nada mais é que uma ferramenta agregada ao Linux para que este pudesse controlar seu próprio fluxo, este provê ao Linux as funções de *firewall* e NAT. O *Netfilter* contém 3 tabelas padrões (FILTER, NAT, MANGLE) e suas *chains* (INPUT, FORWARD, OUTPUT e POSTROUTING e PREROUTING), também chamadas de situações de fluxo, são elas:

- FILTER: É a tabela padrão do *Netfilter,* sua função é realizar o filtro dos pacotes.
    - INPUT: Tudo que entra no *Host*.
    - FORWARD: Tudo que chega ao *Host*, mas que deve ser redirecionado a outro.
    - OUTPUT: Tudo que sai do *Host*.
- NAT: Implementa funções de tradução de endereços de Rede
    - PREROUTING: Realiza alterações nos pacotes antes do roteamento dos mesmos.
    - OUTPUT: Trata os pacotes emitidos pelo *host Firewall*.
    - POSTROUTING: Realiza alterações nos pacotes após o roteamento dos mesmos.
- MANGLE: Implementa alterações especiais em pacotes em um nível mais complexo.
    - PREROUTING: Realiza alterações especiais nos pacotes antes do roteamento dos mesmos.
    - OUTPUT: Trata os pacotes emitidos pelo host Firewall, antes do roteamento dos mesmos.

Para um maior entendimento de como funciona as *chains* do *iptables*, segue abaixo a **Figura 3**.

Figura 3 – Representação das *Chains iptables*



Fonte: Elaborado pelo Autor

De acordo com D'Oliveira Neto (2004) o *iptables* é uma ferramenta de *Front-End* que controla o módulo *netfilter.* O *iptables* é o sucessor do *ipchains*, este foi sucedido através do *kernel* 2.4 no começo de 1999, lançado por Rusty Russell (também participante do projeto *Netfilter*). É possível realizar a afirmação de que o *iptables* é a ferramenta que controla o *netfilter,* pois através dele é possível a criação das regras de *firewall* e NAT.

Este é composto por 4 aplicativos, são eles:

- ***iptables*:** Aplicativo principal para IPv4.
- ***ip6tables*:** Aplicativo principal para IPv6.
- ***iptables-save*:** Aplicativo que salva todas as regras inseridas na sessão ativa.
- ***iptables-restore*:** Aplicativo que restaura todas as regras salvas.

Uma importante observação a ser feita é referente à execução das regras criadas, as regras criadas são executadas na ordem em que estiverem dispostas, ao receber um pacote será realizada uma verificação nas regras, a fim de saber se este pacote terá ou não permissão para passar pelo *firewall*.

É necessário ter conhecimento da estrutura do *iptables*, para que seja possível realizar as configurações nesta ferramenta. Pode-se citar tabelas, *chains*, comandos, parâmetros e ações. Segue abaixo um exemplo de regra do *iptables:*

iptables –t filter –A INPUT –I eth0 –s 192.168.2.200 –j DROP

No exemplo acima é descrita uma regra cujo intuito é descartar toda a comunicação realizada pelo *IP* 192.168.2.200, falando de uma forma mais clara, “Tudo o que for recebido pelo *firewall* através da interface de rede eth0, oriundo do ip, 192.168.2.200 deve ser descartado”.

Conforme pode ser notado na figura o comando *iptables* é responsável por executar as funções na ferramenta. Após isso se tem a tabela indicada por –t, 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 -A indica que será adicionado dentro da lista de regras.

O comando input é a *chain* por qual passará essa informação, ou seja, tudo o que trafegar pela entrada, -s que significa origem 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 geralmente é utilizado quando for constatado que o servidor está sendo alvo de ataques de um determinado IP.

5 EXPERIMENTO REALIZADO

Conforme antecipado nos capítulos anteriores, existem várias formas de se evitar estes tipos de ataques a estes serviços, como “Mover o serviço para uma porta não Padrão”, “Limitação de Recursos” e “Limitar início de conexão (SYN)”. Este capítulo apresenta a aplicação das técnicas citadas e seus resultados, ou seja, os pontos positivos e negativos. Para tal foi projetado um ambiente, onde este ambiente possui duas máquinas virtuais, onde uma máquina virtual será o Atacante (invasor) e a outra será a que receberá a tentativa de invasão (Invadido).

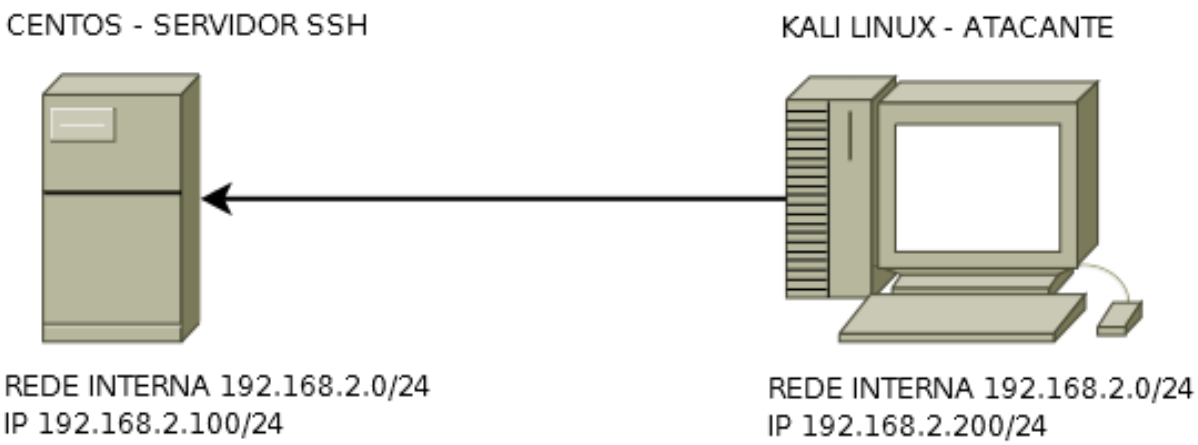
Através destas máquinas serão simulados dois cenários, onde um será sem a aplicação de *Hardening* e o outro com a aplicação da técnica. Inicialmente a fim de explanar como funcionam as ferramentas utilizadas, serão demonstradas algumas configurações e algumas características, posteriormente serão realizados os testes a fim de se obter os resultados que serão apresentados.

5.1 DESCRIÇÃO DO AMBIENTE

Para a realização do experimento foi utilizado um ambiente, contendo duas máquinas virtuais onde a “1ª” será quem sofrerá a tentativa de invasão (Invadido) e a “2ª” será o Invasor. O Ambiente utilizado foi: Duas máquinas Virtuais e uma máquina física, denominada hospedeiro, onde esta foi responsável pela distribuição da rede. As especificações técnicas de hardware e de software utilizadas no hospedeiro são: Processador Intel Pentium Dual Core de 2.16Ghz, 4GB de memória RAM DDR2 e disco rígido de 500GB. Para o software é utilizado do sistema operacional GNU/Linux Mint 17.2 64 Bits possuindo instalado o software de virtualização Oracle VM Virtual Box versão 4.3.28 r100309.

Para as máquinas virtuais foram configurados dois servidores sendo a máquina atacada com sistema operacional GNU/Linux CentOS 32Bits versão 6.5 com 768MB de memória RAM, e a utilizada como máquina atacante possui o sistema operacional Kali Linux 32 Bits versão também com 768MB de memória RAM. Lembrando que isto ocorre tanto no Cenário 1, quanto no Cenário 2. Tudo isto que foi dito acima por ser melhor compreendido através da **Figura 4**.

Figura 4 - Visualização do Ambiente

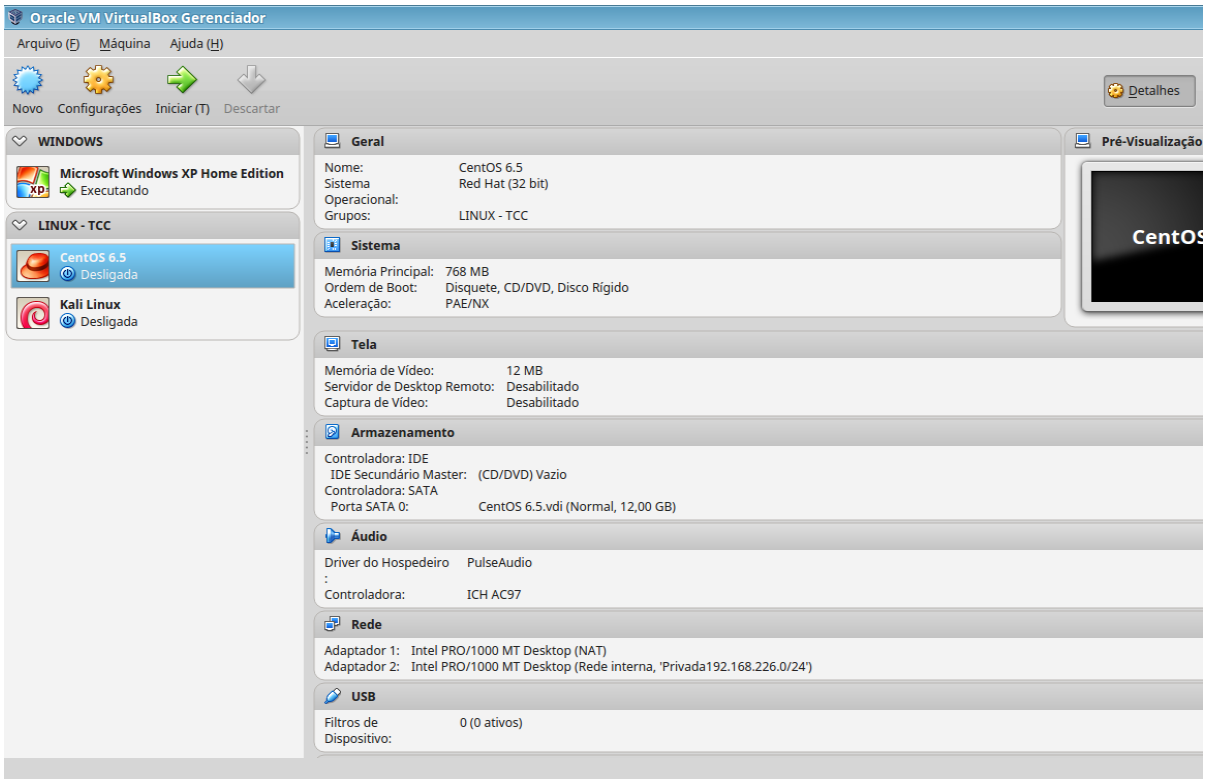


Fonte: Elaborado pelo Autor

5.1.1 Configuração do Virtual Box

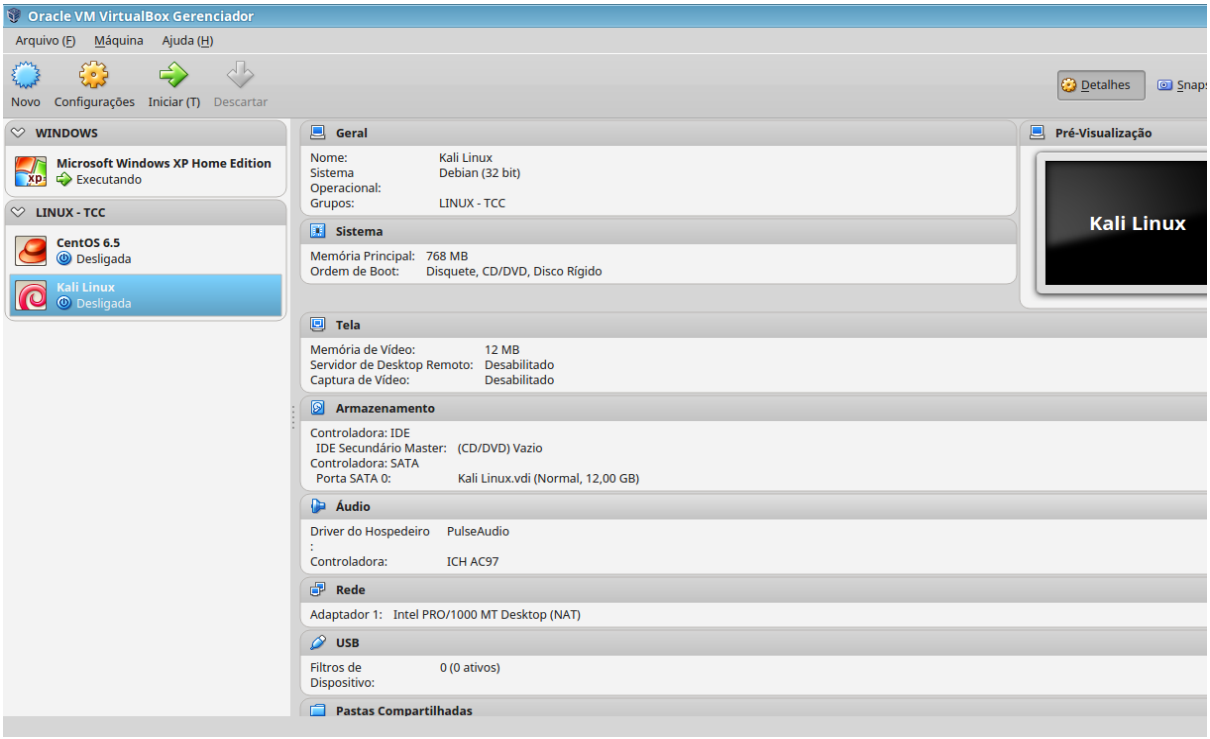
Para que seja possível manter uma padronização no ambiente, as configurações de hardware disponibilizadas para as máquinas contendo o sistema operacional CentOS e Kali serão idênticas conforme é possível verificar nas figuras a seguir (Figuras 5, 6, 7, 8 e 9).

Figura 5 – Máquina Virtualizada CentOS



Fonte: Elaborado pelo Autor

Figura 6 – Máquina Virtualizada KALI



Fonte: Elaborado pelo Autor

Através das figuras anteriores é possível visualizar que ambas máquinas, dispõem de 768MB de memória RAM e 12MB de memória de vídeo.

Para a comunicação entre as máquinas virtuais e a máquina local que será responsável pela disponibilização da rede externa, o CentOS (Máquina Atacada) irá dispor de:

- 01 Placa de Rede ativa em modo NAT para comunicação com a Internet

- 01 Placa de Rede ativa em modo Rede Interna, cujo nome definido foi ‘Privada 192.168.2.0/24’, isto tudo é possível verificar na **Figura 7**.

Figura 7 – Máquina Atacada Interfaces de Rede CentOS

Fonte: Elaborado pelo Autor

Já o Kali (Máquina Atacante) irá dispor de:

- 01 Placa de Rede ativa em modo Rede Interna, cujo nome definido foi ‘Privada 192.168.2.0/24’, isto tudo é possível verificar na **Figura 8**.

Figura 8 – Máquina Atacante Interfaces de Rede KALI

Fonte: Elaborado pelo Autor

Partindo do ponto em que as máquinas já estão instaladas e configuradas, agora será descrito o experimento.

5.2 DESCRIÇÃO DO EXPERIMENTO

Conforme dito anteriormente, o experimento será a aplicação das técnicas citadas anteriormente, onde o foco será obter a resposta para a problemática deste trabalho que é: Quais as vantagens e desvantagens de se aplicar a técnica chamada *Hardening* no *GNU/Linux CentOS.* As vulnerabilidades serão exploradas através de ataques de força bruta que consistem em realizar tentativas automatizadas de ataque repetitivamente onde estas usam usuários e senhas. Geralmente este tipo de ataque utiliza de grandes listas onde estas incluem diversas possibilidades de *logins* (usuários) e senhas, cujo objetivo é de que uma delas seja válida, e assim obter-se o *login* e senha desejado. Para tal será utilizada a ferramenta *hydra*[11], lembrando que existem outras ferramentas de ataque a SSH, como por exemplo a ferramenta BruteSSH2.

Serão utilizados dois cenários para se obter a resposta para a problemática, no primeiro serão realizados testes sem uso de *Hardening*. No segundo serão realizados os testes implementando a técnica.

Para a realização deste experimento foi escolhido um serviço, o SSH cuja porta padrão é a porta 22 e este se encontra disponível em boa parte dos Servidores Linux.

Além da configuração que será realizada no Arquivo de Configuração do serviço, será utilizado o *iptables* a fim de realizar os bloqueios e restrições que forem necessárias.

[11] Ferramenta bastante utilizada para testes de invasão, principalmente quebra de senha (*Password Attack*) por força bruta (*Brute Force)* usando palavras chaves através de wordlist (lista de palavras). Suporta uma vasta quantidade de protocolos para atacar como: HTTP, SMTP, IMAP, POP3, entre outros. (THC-HYDRA, 2015).

Após esta breve explicação sobre como serão realizados os testes, será iniciada a realização do experimento.

5.3 REALIZAÇÃO DO EXPERIMENTO E RESULTADOS OBTIDOS

Conforme já dito anteriormente ataques de força bruta consistem em realizar tentativas automatizadas de ataque repetitivamente onde estas usam usuários e senhas. Geralmente este tipo de ataque utiliza de grandes listas onde estas incluem diversas possibilidades de *logins* (usuários) e senhas, cujo objetivo de que uma delas sejam válidas, e assim obter-se o *login* e senha desejado.

O experimento começa pelo protocolo SSH, onde a primeira simulação será realizada sem que exista qualquer alteração no arquivo de configuração do serviço ou qualquer regra configurada na ferramenta de *firewall iptables*. Na **Figura 9** é possível visualizar que não existem regras configuradas através do comando *iptables* –L.

Figura 9 – Listagem das Regras iptables - CentOS

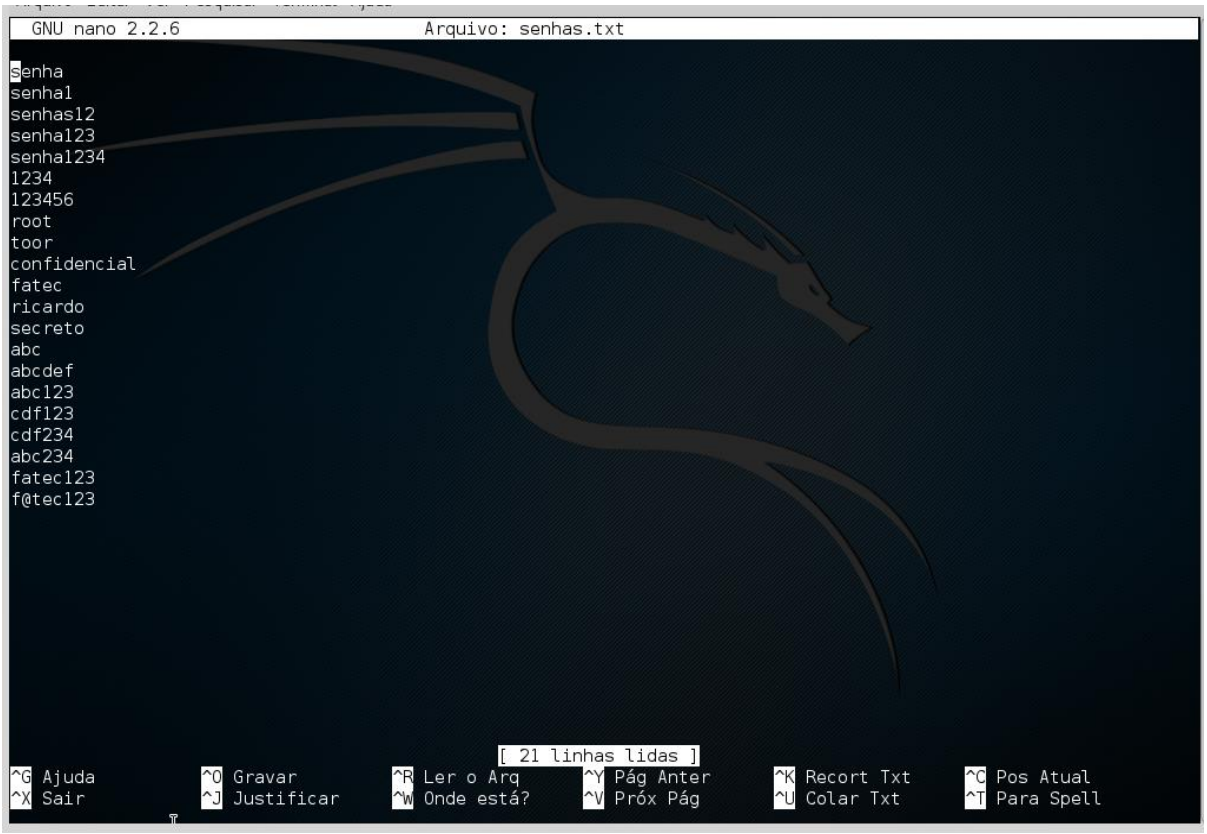
Fonte: Elaborado pelo Autor

Para realizar esta simulação através da Máquina Atacante será utilizada a ferramenta *hydra*, esta é utilizada para a maioria dos testes de vulnerabilidades via força bruta em servidores. Esta faz uso de palavras chaves através de *wordlist* para

testar todas as combinações de senhas possíveis na listagem, sendo assim, caso a senha seja encontrada a ferramenta mostra o resultado na tela.

Para tal será criada uma lista de senhas com algumas possibilidades conforme apresentado na **Figura 10**, onde será inserida a senha válida para o usuário “root” e além desta, conterá também outras possibilidades de senhas.

Figura 10 – Possibilidades de senhas válidas



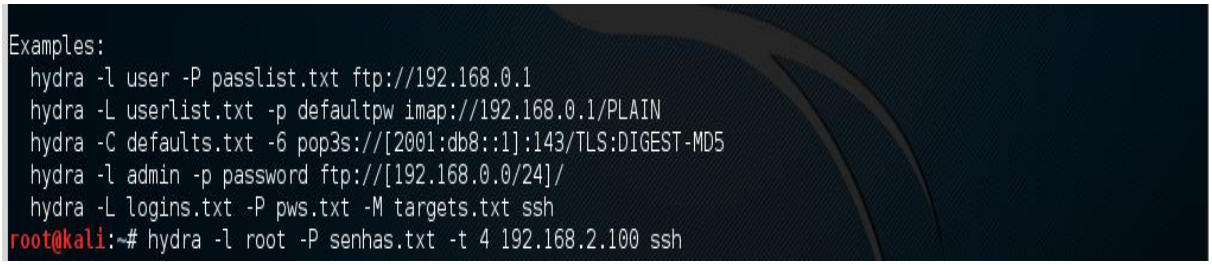
Fonte: Elaborado pelo Autor

Após a criação do arquivo com senhas, o primeiro cenário será apresentado.

5.3.1 Cenário 1: Antes da aplicação da técnica *Hardening*

Para começar aplica-se o comando hydra -l root–Psenhas.txt –t 4 192.168.2.100 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 root e a lista de senhas indicada pelo arquivo senhas.txt onde estas são as senhas possíveis para o usuário root, com no máximo 4 tentativas simultâneas, tudo isto direcionado ao ip 192.168.2.100 ao serviço ssh”, conforme se observa na **Figura 11**.

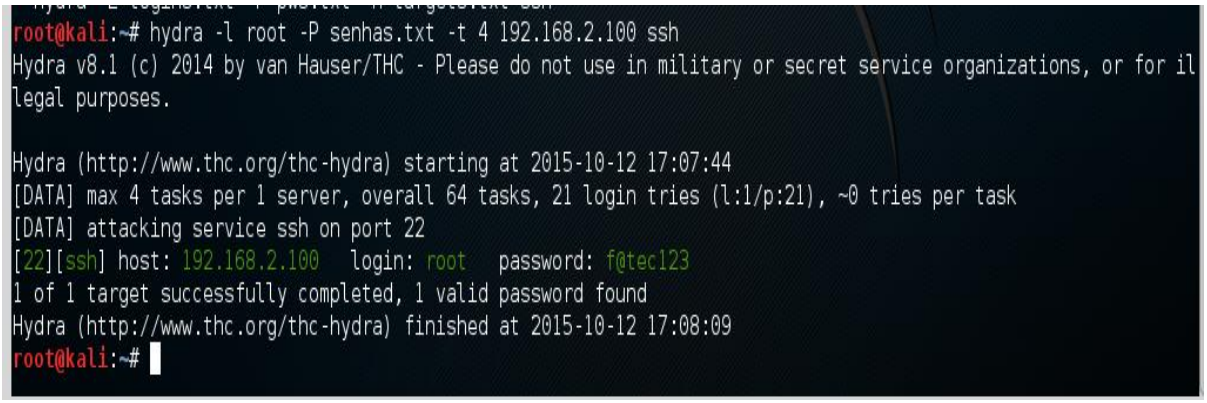
Figura 11 – Ferramenta de Simulação de Ataque



Fonte: Elaborado pelo Autor

Após a realização do comando é possível observar a resposta informada pela ferramenta *hydra* conforme **Figura 12**, onde se apresentam as informações obtidas com as tentativas de acessos.

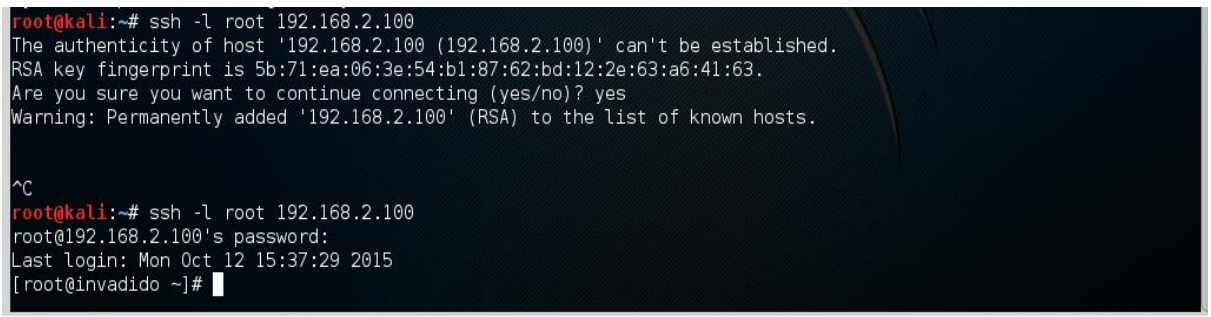
Figura 12 – Resposta da Ferramenta



Fonte: Elaborado pelo Autor

É possível ver na **Figura 12** que após o comando, a ferramenta *hydra* retorna uma possível senha para o usuário root. 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 13**.

Figura 13 – Teste de Acesso SSH



Fonte: Elaborado pelo Autor

Percebe-se então na **Figura 13** que a senha descoberta de fato permite o acesso a Máquina Atacada.

Após perceber que de fato o sistema sem nenhuma proteção é vulnerável a este tipo de ataque, apresenta-se agora o segundo cenário, onde serão aplicadas as técnicas de *Hardening* e os mesmos testes serão repetidos.

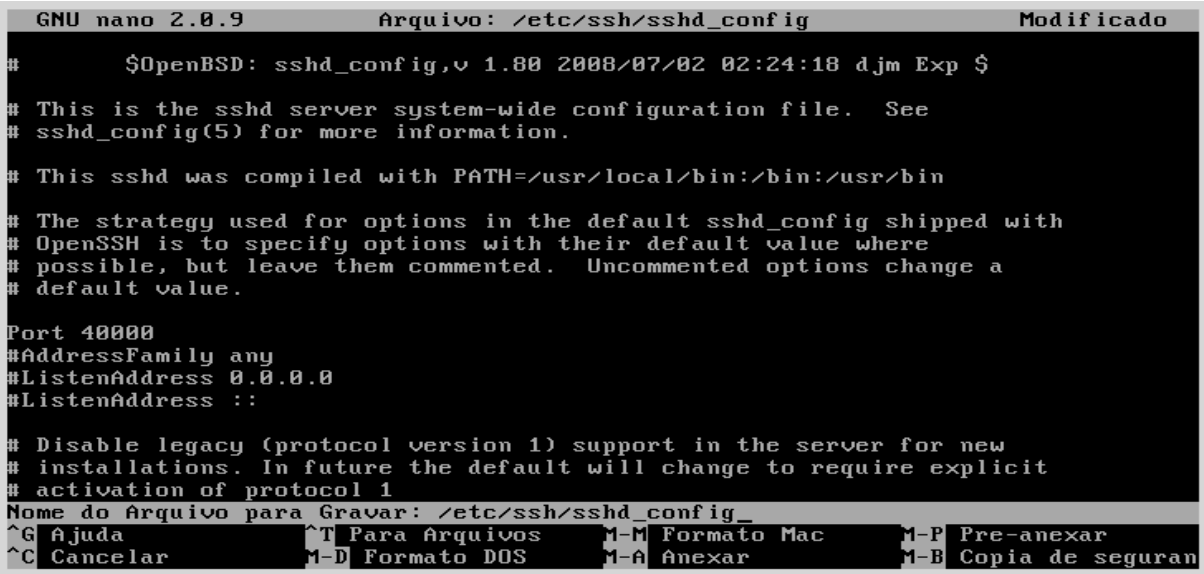
5.3.2 Cenário 2: Aplicação da Técnica *Hardening*

No cenário 2 será realizada a aplicação da técnica *Hardening*, onde serão realizadas alterações no arquivo de configuração do SSH, além de bloqueios via *iptables.*

5.3.2.1 Alteração da Porta Padrão

Esta prática é de simples implementação. Se tratando do OpenSSH basta editar o arquivo sshd_config e substituir, na cláusula Port, o número 22 por outro valor. Lembrando que esta alteração será realizada no Servidor, ou seja, no CentOS, conforme mostra a **Figura 14**.

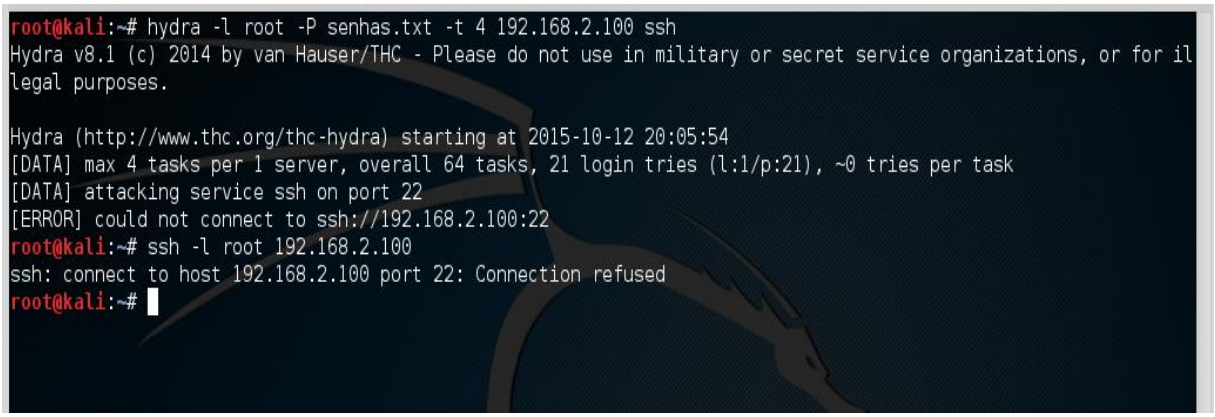
Figura 14 – Alteração da Porta Padrão SSH



Fonte: Elaborado pelo Autor

Após a alteração da porta foi realizado o reinicio do serviço. Posteriormente foi realizada uma nova tentativa de captura da senha focando a porta padrão, onde não se obteve êxito. Somente para fins de confirmação foi realizada uma tentativa de acesso ao *ssh* utilizando o mesmo comando usado anteriormente, novamente não se obteve êxito, conforme mostra a **Figura 15**.

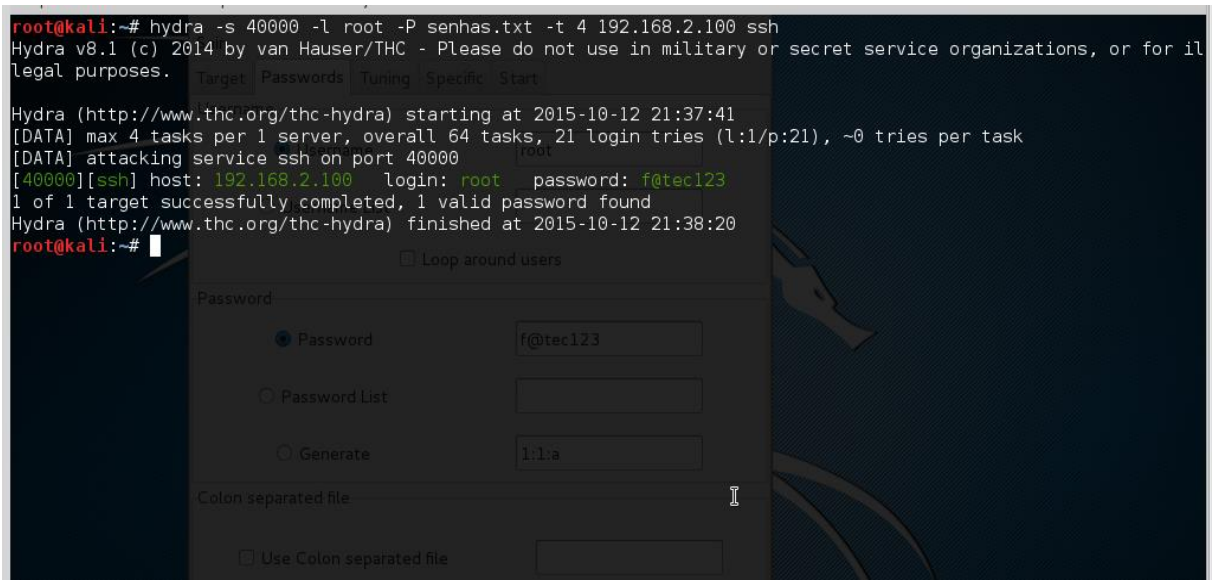
Figura 15 – Alteração da Porta Padrão SSH/Acesso Negado



Fonte: Elaborado pelo Autor

Esta implementação conforme dito anteriormente não soluciona o problema, afinal se o invasor souber a porta do *ssh* ele conseguirá o ataque e o acesso normalmente, conforme **Figura 16**.

Figura 16 – Acesso Efetuado com Informações da Porta



Fonte: Elaborado pelo Autor

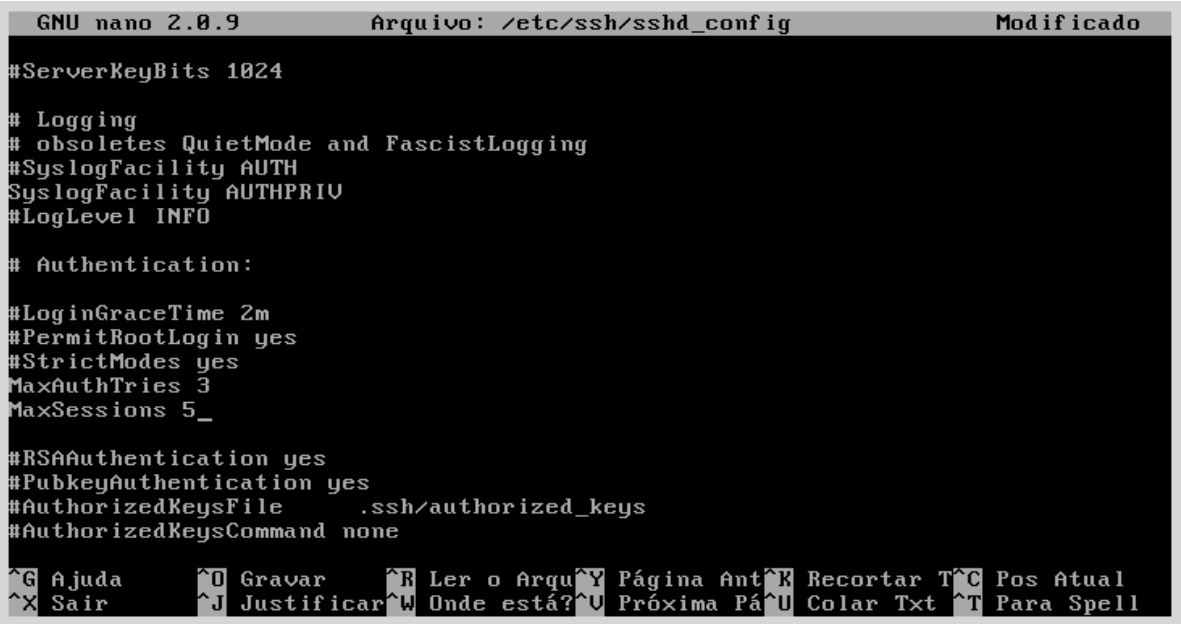
Para solucionar este problema serão aplicadas algumas regras no Firewall do servidor CentOS ao decorrer do experimento.

5.3.2.2 Limitação de Recursos

Esta prática também é de simples implementação. Se tratando do OpenSSH basta editar o arquivo sshd_config, alterar as cláusulas MaxStartups e MaxAuthTries, que são, respectivamente, a quantidade de instâncias não

autenticadas simultâneas (por padrão são 10), e quantas tentativas sem sucesso são permitidas por usuário (por padrão são 6), serão reduzidos os valores de ambas pela metade. Lembrando que esta alteração será realizada no Servidor, ou seja, no CentOS, conforme mostra a **Figura 17**.

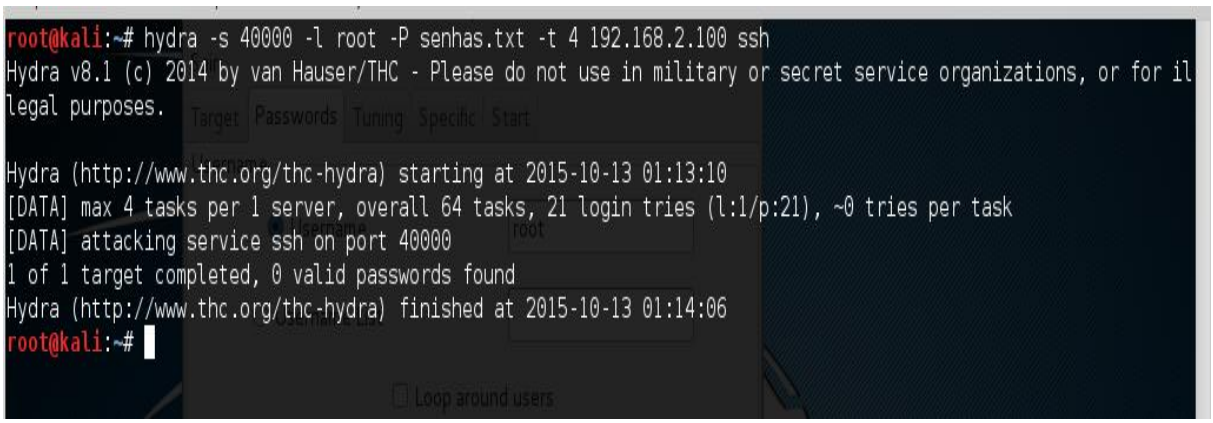
Figura 17 – Alteração no Limite de Autenticações



Fonte: Elaborado pelo Autor

Após a alteração das cláusulas foi realizado o reinicio do serviço. Posteriormente foi realizada uma nova tentativa de captura da senha simulando que o serviço opera na porta 40000, onde também não se obteve êxito no ataque, conforme mostra a **Figura 18**.

Figura 18 – Alteração no Limite de Autenticações (Descoberta da Senha sem êxito)



Fonte: Elaborado pelo Autor

5.3.2.3 Limitar início de conexão (SYN)

Após verificar a vulnerabilidade no 5.3.1 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 40000 -m recent --update --seconds 120 -j DROP

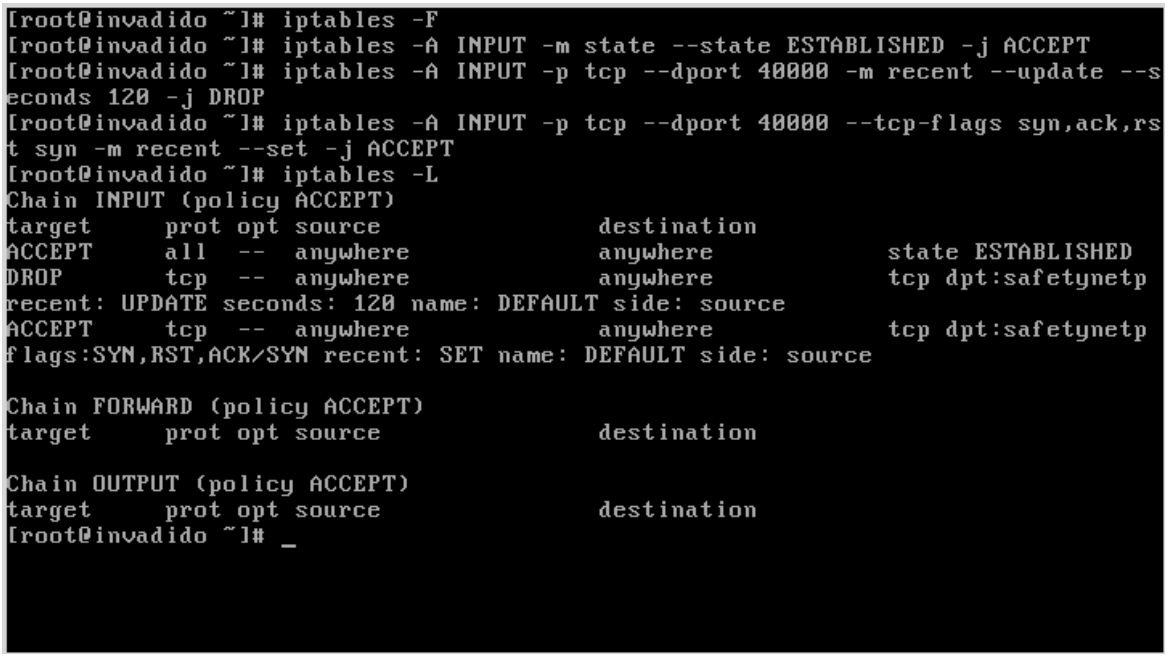
c) iptables -A INPUT -p tcp --dport 40000 --tcp-flags syn,ack,rst syn -m recent --set -j ACCEPT

Para uma maior explanação do que está sendo realizado é possível ler estas regras da seguinte forma:

(a) “iptables, adicione uma regra para tudo o que for com destino entrada e já tiver alguma vez estabelecido conexão com o servidor, uma permissão de entrada”.
(b) “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”.
(c) “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”.

Na **Figura 19**, é possível verificar as regras sendo criadas e também é possível visualizá-las através do comando iptables –L.

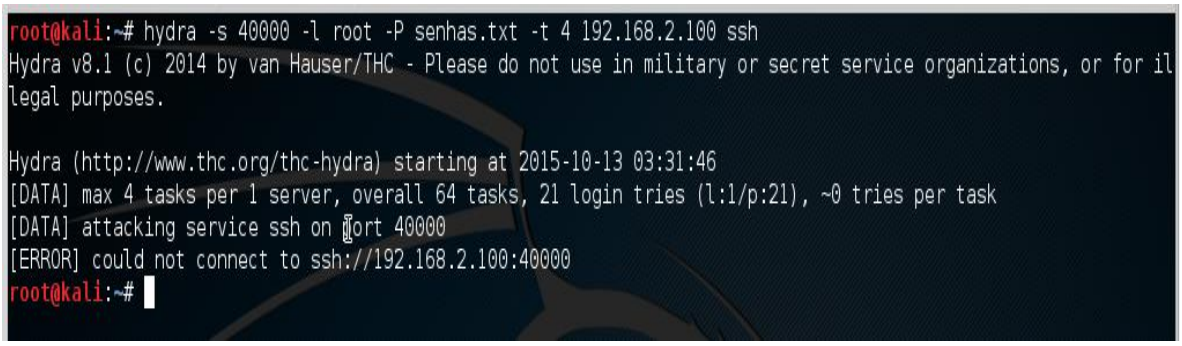
Figura 19 – Regras Implementadas e Listadas



Fonte: Elaborado pelo Autor

Após a aplicação das regras foi realizada uma nova simulação do mesmo ataque direcionado ao serviço SSH do servidor simulando que se saiba que o mesmo opera na porta 40000, onde esta também foi negada e não foi possível obter a senha, conforme se observa na **Figura 20**.

Figura 20 – Senha não obtida após Regras Implementadas e Listadas



Fonte: Elaborado pelo Autor

5.3.2.4 Bloquear *login* acesso com o Root

Esta prática também é de simples implementação. Se tratando do OpenSSH basta editar o arquivo sshd_config, alterar a cláusula PermitRootLogin cujo valor padrão é Yes. Foi alterado para no, conforme **Figura 21**.

Figura 21 – Não permitir Login com o usuário Root

Fonte: Elaborado pelo Autor

Após a aplicação das regras foi realizada uma nova simulação do mesmo ataque direcionado ao serviço SSH do servidor, simulando o fato de que o mesmo opera na porta 40000 e que o usuário é “root” e a senha “f@tec123”, onde este foi negado, conforme se observa na **Figura 22**.

Figura 22 – Login com o usuário Root não Autorizado

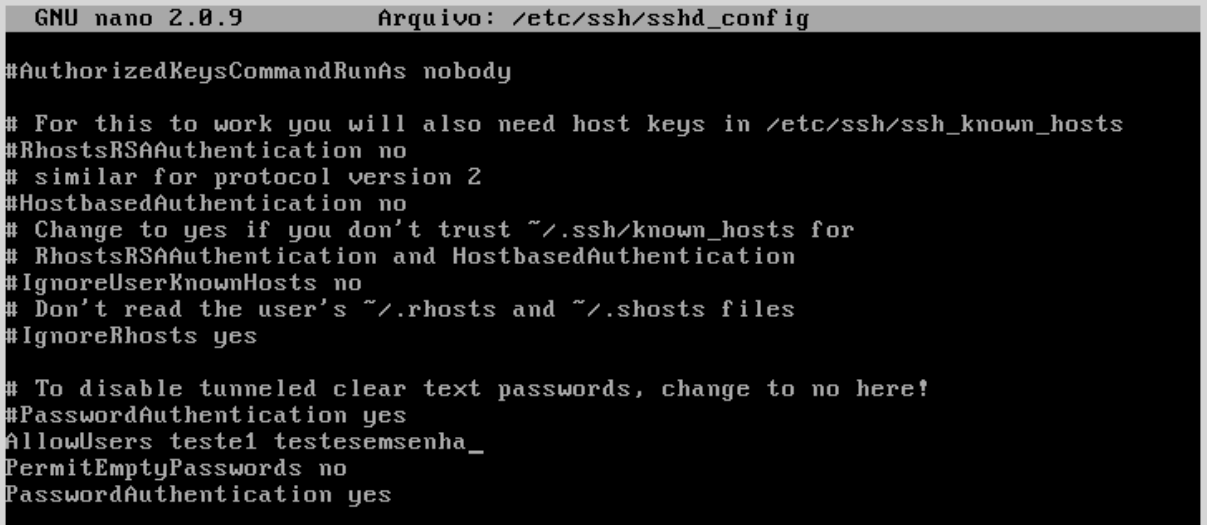
Fonte: Elaborado pelo Autor

5.3.2.5 *Login* somente com usuários autorizados e não permitir senhas em branco

Esta prática também é de simples implementação. Se tratando do OpenSSH basta editar o arquivo sshd_config, inserir a cláusula AllowUsersem seguida o nome de usuário, isto tudo partindo do ponto que já foram criados três usuarios (teste1 ‘f@tec1234’, teste2 ‘f@m123’, testesemsenha ’’), onde o ultimo usuário será utilizado para o teste de não permitir senhas em branco cuja configuração é realizada alterando a cláusula PermitEmptyPasswords, somente o usuário “teste1” e

“testesemsenha” possuem autorização para acessar o SSH do servidor, segue alteração conforme **Figura 23**.

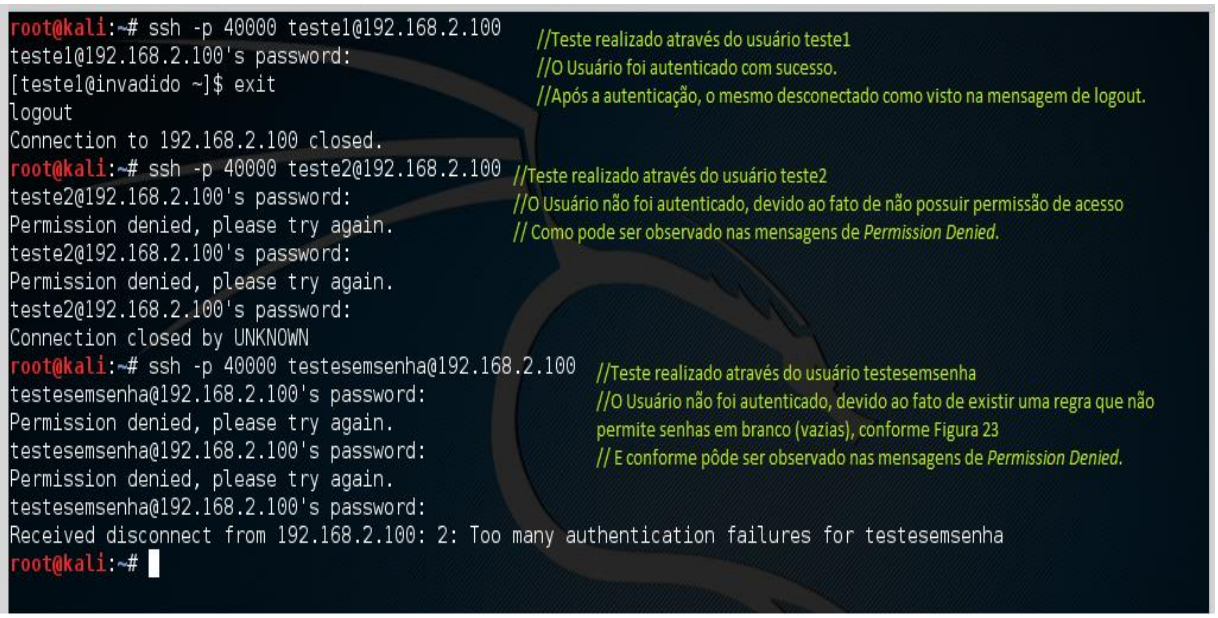
Figura 23 – Login somente com usuários autorizados e não permitir senhas em branco – Configuração



Fonte: Elaborado pelo Autor

Após a aplicação das regras foi realizada uma nova simulação do mesmo ataque, direcionado ao serviço SSH do servidor, simulando o fato de que o mesmo opera na porta 40000, verificando se somente foi possível logar com o teste1, pois o teste2 não é autorizado e o testesemsenha não poderia devido ao fato de não possuir senha, os testes saíram conforme esperado, de acordo com que se observa na **Figura 24**.

Figura 24 – Login com o usuário não permitido e sem senha não Autorizado



Fonte: Elaborado pelo Autor

5.3.2.6 Outras possíveis técnicas de *Hardening*

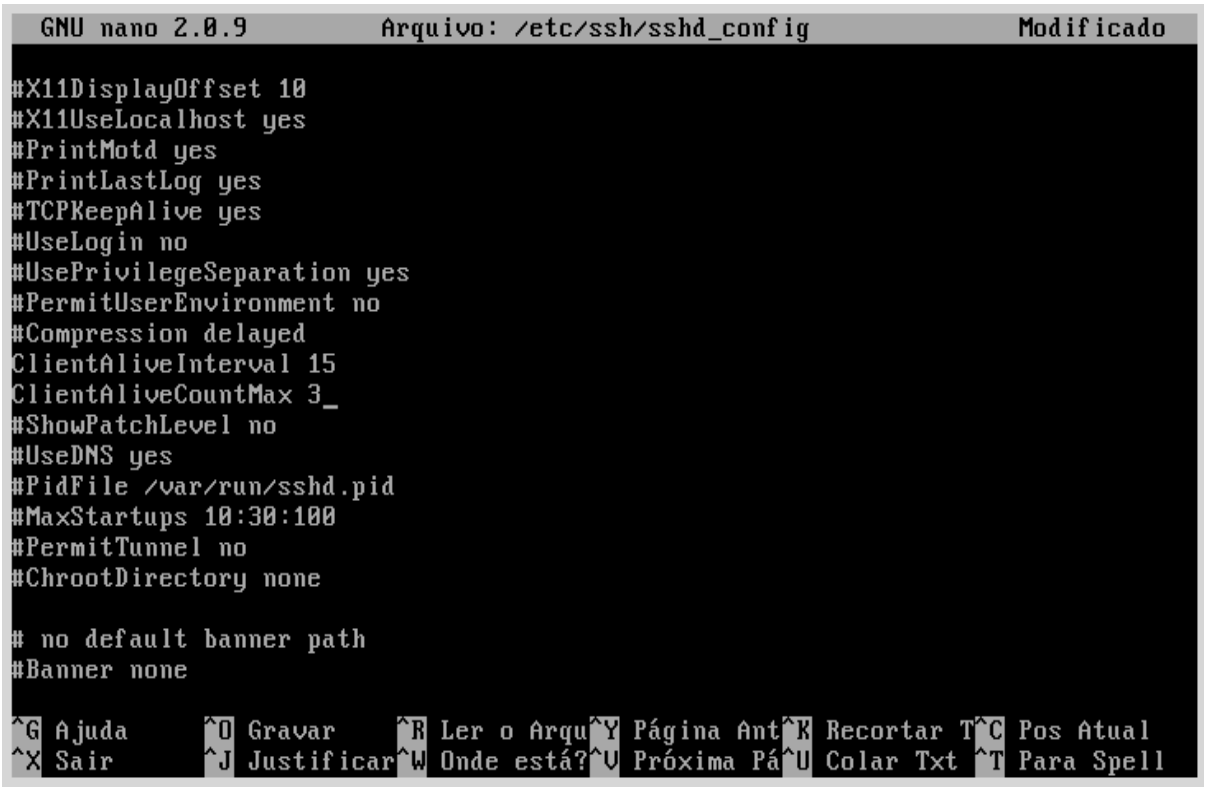
Uma delas é “Ignorar o arquivo .rhosts. O arquivo .rhosts é utilizado principalmente para configurar o conjunto de ferramentas “rtools”: rsh, rcp, rlogin, rwho e rexec, onde este indica quais máquina poderiam fazer a conexão ambas se encontram em desuso devido à falta de segurança das mesmas. Este servia para identificar hosts nos quais o servidor confia. Esta prática também é de simples implementação. Se tratando do OpenSSH basta editar o arquivo sshd_config, alterar as cláusulas IgnoreRhosts para “Yes”, RhostsAuthentication para “no”, e RhostsRSAAuthentication para “no”. Na versão utilizada isto já vem por padrão.

Tem-se também como uma outra possível técnica, a implementação de *Timeouts* por inatividade. É sempre interessante que se configure um timeout que irá desconectar do servidor depois de um tempo de inatividade. Esta prática também é de simples implementação. Se tratando do OpenSSH basta editar o arquivo sshd_config alterar as cláusulas ClientAliveInterval para 15 e ClientAliveCountMax 3.

Para uma maior explanação do que está sendo realizado é possível ler esta aplicação da seguinte forma:

“Clientes SSH que não responderem serão desconectados em aproximadamente 45 segundos”. Segue alteração conforme **Figura 25**.

Figura 25 – Configuração Timeout por Inatividade



Fonte: Elaborado pelo Autor

Após isto os testes foram finalizados e os resultados avaliados.

6 CONSIDERAÇÕES FINAIS

Com a popularização da Internet, a exposição das máquinas se tornou cada vez mais comum. Tem-se então o *firewall* como um elemento imprescindível diante da necessidade ou objetivo de se aumentar a segurança. Quando se tem um sistema Linux no papel de *firewall*, a configuração deste é muito flexível e versátil se ocorrer via *iptables*, conforme pode ser visto.

Seguindo o que diz a definição de *Hardening* as possibilidades de se aumentar a segurança são inúmeras, quando se usa o *iptables* vão das mais simples até as mais complexas; neste trabalho, também foi mostrado que o *iptables* é uma ferramenta poderosa e essencial na proteção de servidores, que pode ser utilizada com sucesso contra os ataques à servidores SSH.

Através do ambiente e do Cenário 1 foi possível visualizar quão desprotegido é um servidor cujo serviço SSH está totalmente mal configurado, e sem a presença de um *firewall* realizando os devidos bloqueios.

Já no Cenário 2 foi possível perceber que através do próprio SSH é possível fortalecer a sua segurança, onde de todas que puderam ser testadas, todas retornaram um bom resultado, ou seja, realmente houve aumento na segurança. Também pode ser visualizado que não basta apenas alterar uma cláusula, ou apenas realizar os bloqueios no *Firewall* e sim que para uma maior garantia é necessário a junção de diversas alterações.

Em se tratando de dificuldades, foi possível notar que todas as configurações tiveram que ser executadas manualmente, o que implica num alto conhecimento ao Sistema Operacional onde a regra está sendo aplicada.

Este trabalho reforça que sempre devem ser realizadas simulações de ataques a vulnerabilidades para que se possa checar se realmente a determinada alteração surtiu o resultado esperado, caso não, estes deverão ser revistos até que estejam devidamente configurados e protegidos, garantindo um pouco mais que segurança estará provida, visto que a mesma nunca chegará a 100%.

Vale citar dois pontos importantes, neste contexto, que não foram explorados e poderiam ser pesquisados como trabalhos futuros: a utilização da ferramenta i*ptables* como auxiliar na execução de ataques contra uma rede com a finalidade de encontrar vulnerabilidades de maneira mais especifica juntamente do NMAP, e a aplicação da técnica chamada *Port Knocking* cuja finalidade é ocultar a porta do determinado serviço para que ela não seja exibida como ‘open’ no NMAP, onde a mesma só será exibida após determinadas portas forem solicitadas antes desta, por exemplo, sendo assim o *iptables* acabará novamente sendo um importante aliado para proteção dos serviços existentes no Linux.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT NBR ISO/IEC 27001. **Tecnologia da informação: técnicas de segurança: sistemas de gestão de segurança da informação: requisitos.** Disponível em: <https://qualidadenapratica.files.wordpress.com/2011/05/iso_iec-27001.pdf>, 2006. Acesso em: 28 ago. 2015.

BALL, B.; DUFF, H. **Dominando Linux Red Hat e Fedora**: Conhecimento, Soluções, Especialização. São Paulo: Makron Books. 2004.

CASTELLS. M. **A Galáxia da Internet.** 1ª. Ed. Rio de Janeiro: Zahar. 2003.

CERT.BR. **Cartilha de Segurança para Internet.** 2ª. Ed. São Paulo: Comitê Gestor da Internet no Brasil. 2012.

CETIC.BR. **Proporção de domicílios com computador.** Disponível em: < http://www.cetic.br/tics/usuarios/2014/total-brasil/A1/expandido>, 2015. Acesso em: 12 set. 2015.

CETIC.BR. **Proporção de domicílios com acesso à internet.** Disponível em: <http://www.cetic.br/tics/usuarios/2014/total-brasil/A4/expandido>, 2015. Acesso em: 12 set. 2015.

DAMSKI, J. C. B; VALENTE, A. S. M. **Internet: Guia do Usuário Brasileiro**. São Paulo: Makron Books. 1995.

DANTAS, M. L. **Segurança da Informação: uma abordagem focada em gestão de riscos.** Olinda-PE: Livro Rápido, 2011.

D'OLIVEIRA NETO, U. **Dominando Linux firewall Iptables**. Rio de Janeiro: Ciência Moderna. 2004.

GOOGRICH, M. T; TAMASSIA, R. **Introdução à segurança de computadores.** Porto Alegre: Bookman. 2013.

HILL, R. **Getting started with SSH security and configuration.** Disponível em: < https://www.ibm.com/developerworks/aix/library/au-sshsecurity/>, 2011. Acesso em: 15 set. 2015.

KUROSE, J. F; ROSS, K. W. **Redes de Computadores e a Internet:** Uma Abordagem Top-Down. 5ªed. São Paulo: Pearson Education do Brasil. 2010.

MACÊDO, D. **Segurança de redes de computadores.** Disponível em: <http://www.diegomacedo.com.br/seguranca-de-redes-de-computadores/#more-1659>, 2012. Acesso em: 13 dez. 2015.

MORIMOTO, C. E. **FTP**. Disponível em: <http://www.hardware.com.br/termos/ftp>, 2005. Acesso em: 15 set. 2015.

MORIMOTO, C. E. **Telnet**. Disponível em: <http://www.hardware.com.br/termos/telnet>, 2005. Acesso em: 15 set. 2015.

MORIMOTO, C. E. **Tutorial do CentOS**. Disponível em: <http://www.hardware.com.br/tutoriais/centos/>, 2008. Acesso em: 17 dez. 2015.

MURILO, N. **Sugestões para defesa contra-ataques de força bruta para SSH.** Disponível em: <http://www.cert.br/docs/whitepapers/defesa-forca-bruta-ssh/#2>, 2007. Acesso em: 15 set. 2015.

RED HAT, INC. **Red Hat Enterprise Linux 4:** manual de referencia**.** Disponível em: <http://www.gb.nrao.edu/pubcomputing/redhatELWS4/RH-DOCS/rhel-rg-es-4/ch-ssh.html>, 2005. Acesso em: 15 set. 2015.

REIS, F. A; VERBENA, M. F; JULIO, E. P. **Hardening.** Disponível em: <http://www.devmedia.com.br/hardening-artigo-revista-infra-magazine-1/20818>, 2011. Acesso em: 16 dez. 2015.

SABINO, V. C. **Um estudo sistemático de licenças de software livre.** Dissertação apresentada ao instituto de matemática e estatística da Universidade de São Paulo (USP). São Paulo, 2011.

SKOUDIS, E. **Step-by-step guide to computer attacks and effective defenses**. 2ªed.Canada: CRAM101. 2012.

THC.ORG. **THC-Hydra**. Disponível em: <https://www.thc.org/thc-hydra/>, 2015. Acesso em: 15 set. 2015.

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