Faculdade de Tecnologia de Americana – Ministro Ralph Biasi

Curso Superior de Tecnologia em Segurança da Informação

Cristiano Ferreira da Silva

Rafael Bradaschia Cortez

HARDENING: SEGURANÇA EM SERVIDORES LINUX

Boas Práticas

Americana, SP

2019

Faculdade de Tecnologia de Americana – Ministro Ralph Biasi

Curso Superior de Tecnologia em Segurança da Informação

Cristiano Ferreira da Silva

Rafael Bradaschia Cortez

HARDENING: SEGURANÇA EM SERVIDORES LINUX

Boas Práticas

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. Juliane Borsato Beckedorff Pinto
Área de concentração: Segurança Física

Americana, SP

2019

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

S579h SILVA, Cristiano Ferreira da.

Hardening: segurança em servidores Linux: boas práticas. / Cristiano Ferreira da Silva, Rafael Bradaschia Cortez.. – Americana, 2019.

54f.

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

Orientador: Profa. Esp. Juliane Borsato Beckedorf Pinto

1 Segurança em sistemas de informação 2. Linux – sistema operacional. I. CORTEZ, Rafael Bradaschia II. PINTO, Juliane Borsato Beckedorf III. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana

CDU: 681.519

681.3.066

Faculdade de Tecnologia de Americana

Cristiano Ferreira da Silva
Rafael Bradaschia Cortez

HARDENING: SEGURANÇA EM SERVIDORES LINUX
Boas Práticas

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

Americana, 06 de Dezembro de 2019.

Banca Examinadora:

Juliane Borsato Beckedorff Pinto (Orientador)
Especialista
FATEC - Americana

Kleber de Oliveira Andrade (Membro)
Doutor
FATEC - Americana

Renan Mercuri Pinto (Membro)
Doutor
FATEC - Americana

AGRADECIMENTOS

Em primeiro lugar agradecemos a Deus que permitiu que nós tivéssemos forças para concluir este curso. Os nossos familiares e amigos pelo incentivo e dedicação ao nosso sucesso.

Agradecemos também aos professores pela paciência e dedicação ao lecionar ao longo destes 3 anos.

DEDICATÓRIA

Dedicamos esta monografia a Deus que nos proporcionou a chance de nos superarmos a cada dia, e em especial aos brilhantes amigos: Fernando, Filipe, Jéssica, Tiago, e Pamela pelo excepcional apoio e incentivo que nos deram durante a pesquisa.

RESUMO

O presente texto conceitua o fator da segurança da informação no âmbito da instalação, configuração e manutenção de servidores que utilizam sistemas operacionais baseados em *Linux* ao longo do seu tempo de vida, tratando de pontos fundamentais para a garantia dos três pilares da segurança da informação e que, em alguns casos, não são percebidos pelos administradores de rede nos ambientes empresariais. O texto relata a quantidade de dispositivos vulneráveis expostos ao mundo, que além de colocarem em risco a segurança dos dados de suas próprias redes, ainda podem servir de *bots* em ataques *DDoS*. O texto também ressalta sobre a importância significativa que possuem as políticas, como a de *backup* e de recuperação de desastres no quesito de ajudarem a garantir que uma empresa seja capaz de dar continuidade ao seu negócio, mesmo após uma interrupção da sua entrega de serviço. Seja essa interrupção causada por algum desastre natural, ou até mesmo algum ato criminoso. Também será demonstrada algumas configurações, sejam elas na inicialização do Sistema Operacional (S.O.), quanto em ferramentas pré instaladas, a fim de tornar o sistema mais seguro para utilização.

Palavras Chave: Hardening, Segurança da Informação, Servidores Linux

ABSTRACT

The present text conceptualizes the factor of the information security in the scope of the installation, configuration and maintenance of servers that use Linux-based operating systems throughout their life time, dealing with fundamental points to guarantee the three pillars of information security and which in some cases are not perceived by network administrators in enterprise environments. The text reports the number of vulnerable devices exposed to the world, which in addition to jeopardizing the security of data from their own networks, can still serve as bots in DDoS attacks. The text also highlights the significant importance of policies such as backup and disaster recovery to help ensure that a business is able to continue business, even after a disruption of its service delivery. Whether this interruption is caused by some natural disaster, or even some criminal act. Also it will be show some configurations, be it Operational System initial settings or pre-installed applications. In order to increase security on system.

Keywords: *Hardening; Information Security; Linux Servers;*

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

INTRODUÇÃO

Hoje em dia a segurança da informação deve ser a base de qualquer projeto computacional, desde um pequeno *blog* hospedado em um servidor isolado, até os *data centers* das grandes corporações.

O principal objetivo da Segurança da Informação é proteger a informação e não os ativos tecnológicos por onde essa informação passa. Há três pilares fundamentais que garantem que uma informação está segura: a confidencialidade, a integridade e a disponibilidade. Esses três atributos da informação devem pautar todas as nossas ações e planejamentos para podermos obter um ambiente seguro.

A confidencialidade garante que a informação estará disponível somente a quem realmente necessita acessar. Para poder garantir a confidencialidade de uma informação, é necessário que haja um mapeamento de todos os usuários da rede e uma definição bem clara sobre os níveis de acesso que cada indivíduo terá na rede. Para que as informações trafeguem de forma confidencial, é necessário que todas as comunicações da rede sejam feitas de forma criptografada.

A integridade garante que a informação seja autêntica, sem corrupções ou adulterações em seu conteúdo. Para tal, é necessário que, além das medidas de segurança já citadas anteriormente, haja redundância no armazenamento das informações e uma política de *backup* bem estabelecida.

Disponibilidade deve garantir que a informação esteja disponível quando necessário a quem estiver autorizado a acessá-la. O conceito de disponibilidade deve assegurar que a informação esteja disponível independentemente do que aconteça, mesmo que ocorra desastres naturais, como terremotos, explosões, ou mesmo falhas sistêmicas, etc. Para tanto, além de medidas de segurança anteriormente citadas e de uma política forte de *backups*, é necessário que se defina um plano de recuperação de desastres.

O objetivo geral deste trabalho é demonstrar que um ambiente seguro depende não apenas do tipo do sistema utilizado, mas da forma como ele é configurado e como esse sistema interage com os outros dispositivos da rede.

Como objetivos específicos, pretende-se com esse trabalho demonstrar a confiabilidade, escalabilidade e versatilidade de um sistema operacional baseado no *kernel Linux* quanto a segurança da informação.

O trabalho foi estruturado em 10 capítulos, sendo que o primeiro conceitua o leitor quanto a importância da segurança da informação, as principais causas de incidentes e o *Hardening*. O segundo capítulo trata sobre o ambiente físico dentro de uma organização. Já o terceiro capítulo aborda os conceitos básicos sobre o *Linux* e de alguns quesitos técnicos quanto a instalação inicial. O quarto capítulo trata-se sobre o *LVM*, uma ferramenta utilizada para gerenciamento de volumes lógicos e físicos dentro de um sistema operacional. O quinto capítulo demonstra a necessidade de uma política de usuário dentro de uma organização, além de tratar o gerenciamento de permissões de usuários. O sexto capítulo aborda sobre a ferramenta *SELinux*, utilizada para gerenciamento de permissões de acessos. O sétimo capítulo aborda os conceitos sobre *backups* e suas diferenças e alguns pontos relacionados a planos de recuperação de desastres que deve-se atentar ao configurar e manter um servidor. Os capítulos oito e nove abordam a instalação e as configurações necessárias e demonstra os resultados referentes aos testes práticos e prova de conceito. No décimo e último capítulo, o texto conceitua a forma de se realizar a transição do sistema operacional, após o término de seu ciclo de vida, além das considerações finais.

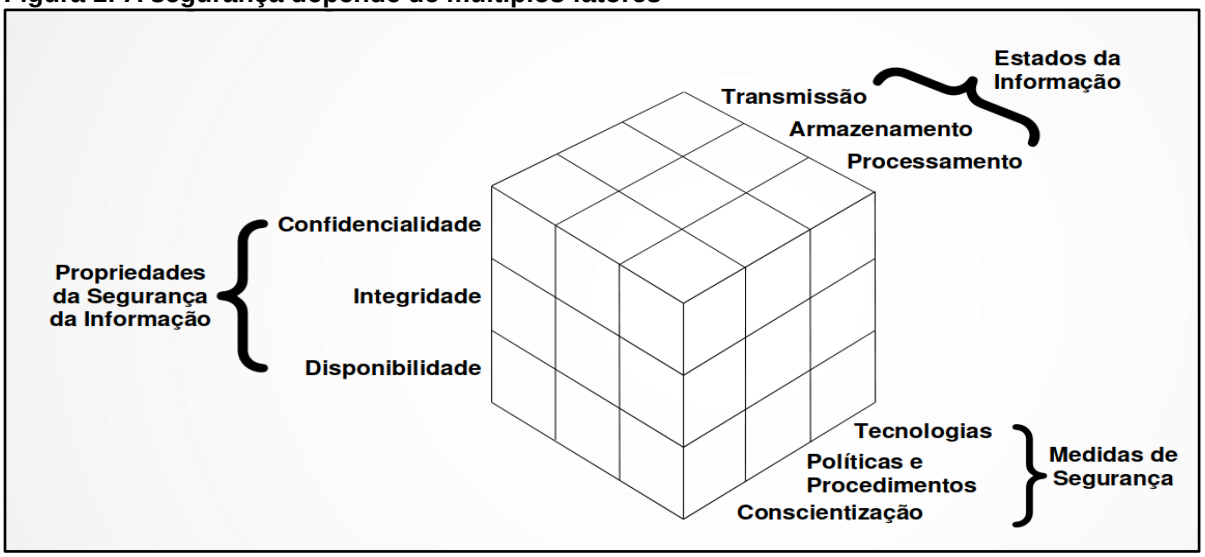
1. CONCEITOS BÁSICOS SOBRE A SEGURANÇA DA INFORMAÇÃO

Segundo Torres (1999) *apud* Lyra Rocha (2015, p. 9), o que diferencia uma informação de um dado é o contexto no qual esse dado é apresentado. Segundo Torres o dado isolado não tem nenhum valor, pois não se pode interpretá-lo, porém, se esse dado for exposto sob um contexto, ou seja, algo que faça com que esse dado tenha sentido ou um significado, ele passa a ser uma informação e necessita de cuidados quanto a sua segurança, pois, se obtido por algum atacante ou perdido em um desastre, pode ocasionar prejuízos ao dono da informação.

> Podemos definir a Segurança da Informação como uma área do conhecimento dedicada à proteção de ativos da informação contra acessos não autorizados, alterações indevidas ou sua indisponibilidade. (SÊMOLA, 2003, p. 43).

Torres relata que a segurança da informação tem por base três princípios básicos para a sua existência: a integridade, a confidencialidade e a disponibilidade (a segurança depende de múltiplos fatores). Esses atributos são mencionados por muitos autores como CID, conforme é apresentado na figura 1.

Figura 1: A segurança depende de múltiplos fatores



Fonte: CERT.br (a) (2014)

Confidencialidade – Garantir que as informações sejam entregues apenas àqueles que possuem autorização para obtê-las.

Integridade – Garantir que as informações estejam intactas e não corrompidas acidentalmente ou deliberadamente, e que elas estejam corretas e completas.

Disponibilidade – Garantir que as informações estejam disponíveis e acessíveis sempre que necessário aos que precisarem dela.

Existem alguns outros elementos complementares que ajudam a entender como funciona o processo de segurança da informação.

Autenticação – Garantir que um usuário é de fato quem alega ser.

Não repúdio – Capacidade de um sistema de provar que algum usuário foi o autor de uma ação.

Legalidade – Garantir que um sistema esteja de acordo com a legislação do país em que opera.

Privacidade – Capacidade de um sistema de ocultar informações sensíveis de um usuário, impossibilitando o rastreio deste usuário através de suas ações.

Auditoria – Capacidade de um sistema de auditar e fiscalizar tudo o que foi realizado pelos usuários, possibilitando a detecção de fraudes ou tentativas de ataque.

Ameaças – As ameaças, normalmente externas, são conhecidas por serem difíceis de ser controladas ou previstas. É caracterizada por acontecimentos externos como por exemplo uma catástrofe natural seja ela uma inundação, um incêndio, ou até algum evento envolvendo pessoas, como uma greve.

Risco – O conceito de risco, seja ele interno ou externo, tem como característica, explorar pontos de vulnerabilidades dentro de uma organização. Riscos externos podem ser definidos como softwares maliciosos ou *hackers* que queiram atacar a organização, já os riscos internos podem ser exemplificados como um servidor que fica em um local impróprio, ou até um mal controle de acesso a uma sala desse mesmo servidor.

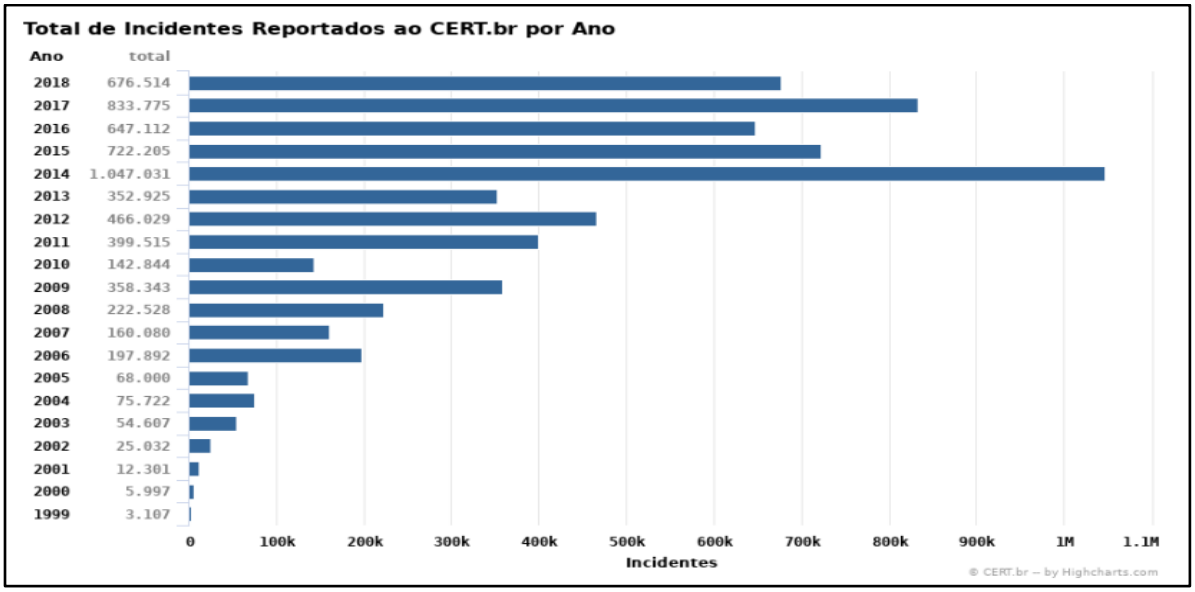
Vulnerabilidade – A vulnerabilidade pode ser definida como falha ou fraqueza e só existe quando a organização tem uma má configuração do ambiente de T.I., por isso é extremamente necessário que o administrador se mantenha sempre

atualizado em relação a *softwares*, políticas e controles, dessa forma o administrador irá mitigar essas falhas e fraquezas, e garantirá que seu ambiente esteja o menos vulnerável possível.

1.1 OS PRINCIPAIS ATAQUES

Quando é analisada a (Figura 2), percebe-se uma queda nos números de ataques cibernéticos em em comparação aos anos anteriores. Porém ainda assim os números são preocupantes segundo o CERT.br (Grupo de Resposta a Incidentes de Segurança para a Internet no Brasil). Visto que os números por mais que tenham diminuídos, ainda assim causam um grande impacto na segurança da informação.

Figura 2: Total de incidentes reportados ao CERT.br por ano

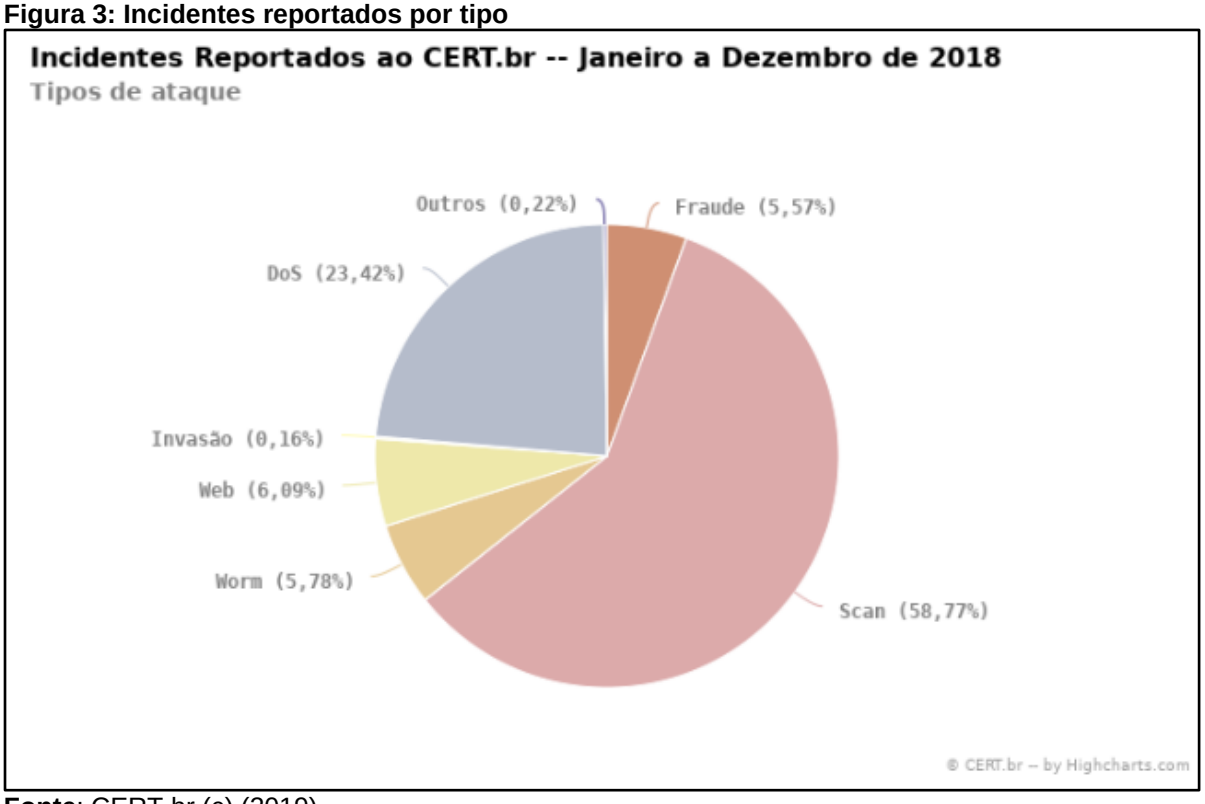


Fonte: CERT.br (b) (2019)

Segundo o CERT.br, mais da metade dos incidentes reportados no ano de 2018 são varreduras de redes (Figura 3), sondagem muito realizada por atacantes em busca de maiores informações sobre as suas possíveis vítimas e suas vulnerabilidades. O segundo tipo de incidente mais reportado é o de ataques *DDoS* (*Distributed Denial of Service*), ou ataque de negação de serviço distribuído, segundo *Kaspersky* (2018).

> Os ataques de rede distribuídos muitas vezes são chamados de ataques de negação de serviço distribuído (*DDoS*). Esse tipo de ataque aproveita os limites de capacidade específicos que se aplicam a todos os recursos de rede, como a infraestrutura que viabiliza o site de uma empresa. O ataque *DDoS* envia múltiplas solicitações para o recurso *Web* invadido com o objetivo de exceder a capacidade que o site tem de lidar com diversas solicitações, impedindo seu funcionamento correto. (KASPERSKY, 2018).

Figura 3: Incidentes reportados por tipo



Fonte: CERT.br (c) (2019)

Levando em consideração os itens citados acima, pode-se concluir que o cuidado com a segurança da informação deve ser tomado em todas as camadas por onde a informação trafega.

1.2 HARDENING

Para se ter um ambiente de baixo risco, ou seja, protegido contra as ameaças, internas ou externas e livre de vulnerabilidades é necessário que cada mínima parte de todo o ecossistema esteja bem configurada para que não haja brechas que possam ser exploradas por pessoas mal-intencionadas ou mal instruídas.

Esse processo de configuração é conhecido na área por *Hardening*. Durante o processo de *Hardening*, são executados diversos processos para aumentar a segurança de um sistema. A princípio, sempre deve-se ter em mente que um sistema que possua uma função única é, por via de regra, mais seguro do que um sistema multifunção. Por exemplo, se temos uma aplicação *web*, seria aconselhável que o banco de dados, a aplicação em si e o *storage* ficassem em servidores separados, pois em um eventual comprometimento de um desses serviços, ficaria muito mais complicado para o atacante obter acesso aos outros serviços.

> O processo de fortificação do sistema operacional aplicando técnicas específicas de controles para melhoramento das configurações com o objetivo de tornar mais seguro é conhecido pelo termo em inglês "*hardening*" em tradução literal "endurecimento". No contexto da segurança da informação, significa também o processo de proteger um sistema através da redução de suas possíveis vulnerabilidades. (TURNBULL, 2005, p.1).

O *Hardening* em sistemas envolve vários processos, dentre eles, o fechamento de portas de rede que não são necessárias para o funcionamento do servidor, a exclusão de pacotes desnecessários, a configuração correta dos serviços que rodam no sistema, a constante atualização do sistema operacional, bloqueio de usuários inativos, imposição de uma política de senha eficiente. Mas o processo de enrijecimento de um sistema operacional começa lá no planejamento. Como dito anteriormente, um sistema com um único propósito tende a ser mais seguro. Tendo isso em mente, é possível se planejar desde a distribuição do espaço em disco pelas partições, até quais serviços serão ou não mantidos.

2. O AMBIENTE FÍSICO

Quando se trata de um ambiente corporativo, a segurança deve ser trabalhada em todos os ambientes, desde a segurança física dos equipamentos, até o treinamento dos funcionários e estabelecimento de políticas que garantam um bom nível de segurança para as informações que trafegam dentro daquela rede. Fontes redundantes*, No-Breaks*, banco de baterias e geradores ajudam a garantir que as aplicações e os dados estarão disponíveis em caso de pane elétrica. *RAID* com espelhamento de discos garantem um retorno mais rápido e eficiente da operabilidade do sistema em casos de falhas nos discos.

> Tal como já mencionado, *RAID* é a sigla para *Redundant Array of Independent Disks* ou, em tradução livre, algo como "Matriz Redundante de Discos Independentes". Trata-se, basicamente, de uma solução computacional que combina vários discos rígidos (HDs) para formar uma única unidade lógica de armazenamento de dados. (INFOWESTER, 2018).

A proteção dos ambientes físicos é uma medida primordial para qualquer empresa que deseja ter o mínimo de segurança das suas informações e de seus clientes. Controles de acessos, câmeras de segurança, treinamento forte e contínuo dos empregados são medidas simples que não extinguem o risco de um ataque de um engenheiro social, mas dificultam bastante a jornada de um atacante. Através da engenharia social, um atacante pode facilmente adentrar em uma empresa que não se preocupa com esses pequenos detalhes. No livro “A arte de enganar”, de Kevin Mitnick, publicado em 2003, o autor conta diversos casos em que, com algumas ligações, foi possível se obter dados confidenciais.

> A engenharia social usa a influência e a persuasão para enganar as pessoas e convencê-las de que o engenheiro social é alguém que na verdade ele não e, ou pela manipulação. Como resultado, o engenheiro social pode aproveitar-se das pessoas para obter as informações com ou sem o uso da tecnologia. (MITNICK, 2003, p.6).

No capítulo (3. O sistema operacional), é abordado fatores que um administrador deve levar em consideração quando está escolhendo um sistema operacional para ser instalado em seu servidor.

3. O SISTEMA OPERACIONAL

Deve se levar em consideração diversos fatores ao se escolher um sistema operacional (S.O.) para se instalar em um servidor. Fatores como estabilidade, escalabilidade, confiabilidade, suporte e o mais importante, o custo. No universo da tecnologia, temos diversas opções de S.O., o *Windows Server* da *Microsoft*, as distribuições baseadas em *Linux* e as baseadas em *BSD*. Neste trabalho será utilizado o *CentOS Linux Distribution*, por ser uma distribuição extremamente estável e escalável, com um amplo suporte a ferramentas de segurança, atualizações e correções de erros constantes. Suportado pela comunidade e baseado no *Red Hat Enterprise Linux* (*RHEL*), o *CentOS* é um ótimo sistema para empresas que querem obter um sistema consistente, seguro e sem custos com licenciamento, visto que o projeto *CentOS* é *Open Source* e está escrito sob a licença *GPL3* e distribuído gratuitamente.

> As licenças para a maioria dos softwares e outros trabalhos práticos são projetadas para tirar sua liberdade de compartilhar e alterar os trabalhos. Por outro lado, a Licença Pública Geral GNU tem como objetivo garantir a sua liberdade de compartilhar e alterar todas as versões de um programa - para garantir que permaneça software livre para todos os seus usuários. (OPENSOURCE, 2018).

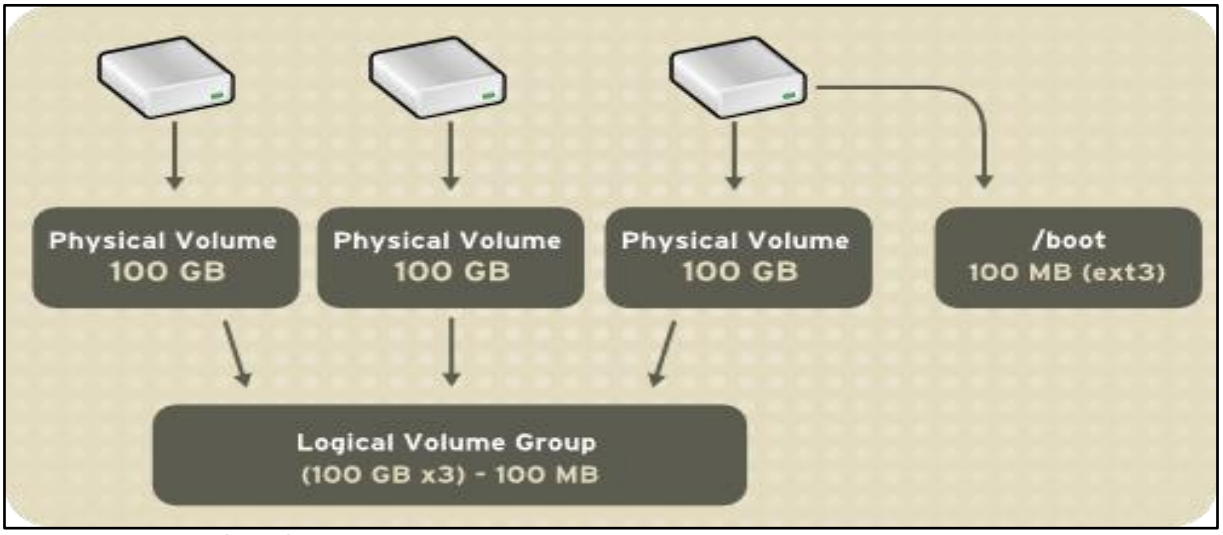
As características do *CentOS*, dão a impressão que ele é um sistema completamente invulnerável e que ao utilizá-lo, os dados estarão totalmente seguros, mas não é bem assim, se não houver consciência e conhecimento ao configurar corretamente o sistema operacional e a comunicação entre os demais nós da rede, deixa-se muitas falhas e vulnerabilidades expostas e que podem acarretar consequências catastróficas em uma corporação.

4. LVM - LOGICAL VOLUME MANAGER

LVM é uma ferramenta muito útil para realizar o gerenciamento de volumes lógicos, capaz de realizar alocação de discos, divisões, espelhamentos e redimensionamento de discos segundo a documentação oficial da *Red Hat (RED HAT, 2018)*.

O *LVM* divide o trabalho em duas etapas, primeiro ele mapeia os discos físicos em grupos de volumes físicos (PV). Com os volumes físicos criados, o *LVM* consegue os agrupar em volumes lógicos (*VG*), conforme é demonstrado na figura 4. A única partição do sistema que não pode ser trabalhada com o *LVM* é o */boot*, pois se trata da partição responsável por inicializar o GRUB e chamar o primeiro processo do sistema operacional, o *init*. A partição */boot* é inicializada pelo *boot-loader*, que por sua vez não é capaz de ler o *LVM*.

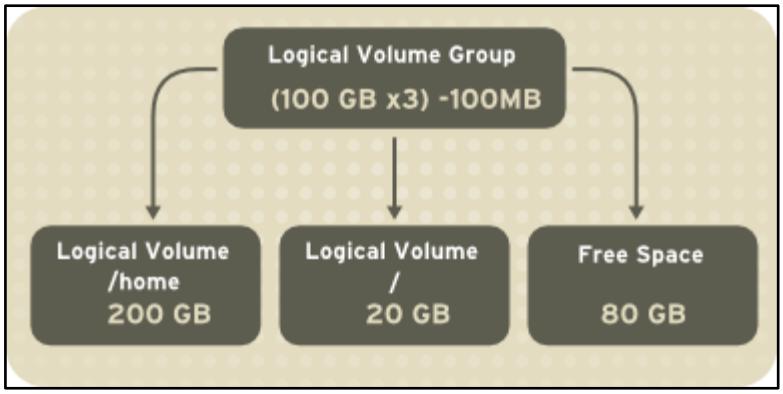
Figura 4: Criação do grupo de volumes



Fonte: RED HAT (2018)

Os *VGs* podem ser divididos em “partições”, chamadas volumes lógicos (*LV*). Essas partições podem ser formatadas e montadas no sistema conforme a necessidade, como o /, */home*, */var*, entre outros. (figura 5). O restante do espaço disponível no *VG* pode ser utilizado para expandir algum *LV* existente ou para criar um.

Figura 5: Distribuição do volume lógico



Fonte: RED HAT (2018)

É importante utilizar *LVM* nos servidores, pois é uma ferramenta que permite a escalabilidade do sistema, na segurança e na confiabilidade do mesmo. Atualmente os sistemas da *Red Hat* trabalham com o *LVM* versão 2, que funciona em versões mais atuais do *kernel Linux* (>2.6).

5. POLÍTICA DE CONTROLE DE USUÁRIOS

Ter controle sobre quem está acessando o sistema operacional é muito importante do ponto de vista de segurança, mas além do controle de quem tem acesso, também é extremamente importante o nível de privilégios que cada usuário irá obter no S.O..

No *linux*, o controle de usuários e grupos é realizado através do modelo de controle de acesso discricionário (*DAC*). O *DAC* define as permissões de leitura, escrita e execução de cada arquivo pelo dono do arquivo, o grupo do dono e o restante dos usuários.

5.1. PERMISSÕES

A transcrição a seguir demonstra a saída do comando *ls -l* (O comando *ls* é responsável por listar o conteúdo de um diretório, enquanto o parâmetro l, indica a listagem detalhada). Na parte em negrito, pode-se ver uma série de caracteres que determinam as permissões de cada arquivo. Esses caracteres se dispõe da seguinte forma:

```
[admin@localhost /]$ ls -l
total 20
lrwxrwxrwx.   1 root root    7 Out  9 16:39 bin -> usr/bin
dr-xr-xr-x.   5 root root 4096 Out 10 07:00 boot
drwxr-xr-x.  20 root root 3240 Nov 18 17:41 dev
drwxr-xr-x.  83 root root 8192 Nov 18 17:41 etc
drwxr-xr-x.   3 root root   19 Out  9 16:57 home
drwxr-xr-x.   2 root root    6 Abr 11  2018 media
drwxr-xr-x.   2 root root    6 Abr 11  2018 mnt
drwxr-xr-x.   2 root root    6 Abr 11  2018 opt
dr-xr-xr-x. 182 root root    0 Nov 18  2019 proc
[ … ]
[admin@localhost /]$
```

O primeiro caractere indica o tipo do arquivo, conforme é demonstrado na tabela 1 os tipos de arquivos e seus respectivos símbolos:

Tabela 1: Tipos de arquivos e seus símbolos

Tipo de Arquivo	Símbolo	Descrição
Arquivo regular	-	Arquivo de texto, de imagem, vídeo, programa executável, etc. etc
Arquivo de Diretório	d	Contém um conjunto de arquivos de quaisquer tipos, inclusive doutros diretórios.
Arquivos de dispositivo - dispositivo orientado a bloco	b	Operações de entrada e saída são realizadas byte a byte de modo sequencial.
Arquivos de dispositivo - dispositivo orientado a caractere	c	as operações de entrada e saída são realizadas em blocos de modo aleatório.
Socket	s	Usado para a comunicação bidirecional entre dois processos.
Named pipe (ou FIFO)	p	Permite a comunicação entre dois processos executados no mesmo Sistema Operacional.
Link	l	Hard link: dois arquivos apontando para o mesmo inode. Symbolic link: ponteiro para um arquivo existente.

Fonte: Próprio Autor

> Arquivo de dispositivo (orientado a bloco e orientado a caractere).
> São os arquivos utilizados para gerenciar os dispositivos de entrada e saída, e apresentam 02 subtipos:
> **Caractere**: as operações de entrada e saída são realizadas de modo sequencial, *byte* a *byte*. Normalmente, os dados são lidos e escritos diretamente no dispositivo, dispensando o uso de *buffers* (espaço em memória). Ex.: portas seriais.
> **Bloco**: as operações de entrada e saída são realizadas de modo aleatório, em blocos, fazendo o uso de *buffers* intermediários. Ex.: disco rígido. **(**GSTI, 2017).

Os demais caracteres são divididos em três grupos de três caracteres que representam as permissões do dono do arquivo, do grupo do dono e dos demais usuários respectivamente.

Os caracteres ‘rwx’ significam:

r = Leitura (*Read*)

w = Escrita (*Write*)

x = Execução (*Execution*)

Quando houver um ‘-’ no lugar de algum desses caracteres, significa que a negação dessa permissão.

5.2. SUID, SGID e STICKY BIT

Além das permissões padrão de leitura, escrita e execução, no modelo *DAC*, ainda é possível que um arquivo obtenha permissões especiais, o *SUID*, *SGID* e o *Sticky Bit*. Com essas permissões, é possível realizar o gerenciamento mais adequado ao ambiente da empresa administrada. Será demonstrado abaixo mais detalhadamente sobre cada um dos tipos.

5.2.1. SUID

Quando é listado os arquivos com um usuário não "*root*" para saber as permissões através do comando *ls -l*, depara-se com a letra 's' ou 'S'. Essa letra representa o *bit* do *SUID* (*Set User ID*), e é uma propriedade para arquivos executáveis. A letra s pode aparecer para o usuário como maiúscula ou minúscula. A diferença entre elas é que o 's' minúsculo representa a permissão de execução, e o 'S' maiúsculo não tem. Isso acontece devido a aparição desse caractere ocorrer no lugar onde deveria aparecer o caractere de execução. A seguir, será apresentado um exemplo de como executar o comando para definir a permissão de *SUID*.

```
[admin@localhost ~]$ chmod 4755 file_suid
[admin@localhost ~]$ ls -l file_suid
-rwsr-xr-x. 1 admin admin 0 Nov 18 17:50 file_suid
```

Quando um programa ou arquivo é executado ou solicitado respectivamente, o mesmo utilizará das permissões do usuário que está logado para saber que atividades o mesmo pode realizar.

O que o *SUID* faz, é definir permissões para que um usuário possa executar um determinado programa com permissões do proprietário, geralmente se aplica isso a arquivos pertencentes ao *root*. É extremamente recomendado que se utilize esse artifício de forma temporária.

> "O *SUID* é um recurso que faz com que qualquer usuário, ao executar um determinado arquivo, tenha

> os mesmos direitos e poderes do dono do arquivo, durante sua execução" (MOTA FILHO, 2012, p.437).

5.2.2. SGID

O *SGID* (*Set Group ID*) tem a funcionalidade muito parecida com a do *SUID* explicada anteriormente, diferenciando por ser aplicada em grupos ou diretórios.

> "O *SGID* (*Set Group ID*) é similar ao *SUID*. No entanto, é voltado para o grupo do arquivo" (MOTA FILHO, 2012, p.438).

Se aplicada essa configuração a um diretório, por exemplo, qualquer usuário que for inserido dentro do grupo que tem o *bit SGID* ativo, herdará essa permissão. A transcrição a seguir, apresenta o comando para definir a respectiva permissão.

```
[admin@localhost ~]$ chmod 2755 file_sgid
[admin@localhost ~]$ ls -l file_sgid
-rwxr-sr-x. 1 admin admin 0 Nov 18 17:52 file_sgid
```

5.2.3. STICKY BIT

A função do *sticky Bit* é fazer com que o arquivo que tenha esse *bit* ativado, seja mantido na memória *swap* após a sua primeira execução. Isso era válido na época da sua criação, em 1974, devido a baixa capacidade das memórias da época. Hoje em dia, o *sticky bit* não tem função alguma, tendo uma função meramente decorativa. A transcrição a seguir demonstra o comando utilizado para aplicar a permissão do *sticky bit*.

```
[admin@localhost ~]$ chmod 1755 file_stickbit
[admin@localhost ~]$ ls -l file_stickbit
-rwxr-xr-t. 1 admin admin 0 Nov 18 17:53 file_stickbit
```

> Por uma questão de compatibilidade com sistemas antigos ou *sticky bit* ainda pode ser atribuído arquivos executáveis a única consideração é o fato de que nos tempos atuais ele não fala nada sendo uma mera figura decorativa essa compatibilidade está no fato de que um arquivo poderá ser trazido de um sistema operacional com stickit ativado e isso tem que

> ficar evidente até mesmo possibilitando a volta de si mesmo arquivo para o sistema de origem nas mesmas condições iniciais assim ou *sticky bit* não poderia simplesmente sumir da mesma forma é possível criar um arquivo no sistema moderno ativar o *sticky bit* enviar para o sistema antigo para que lá eles eles a sua função. (MOTA FILHO, 2012, p. 439).

Basicamente, somente usuário *root* e o proprietário da pasta em questão, pode realizar as alterações do conteúdo que ali está.

> "Os arquivos e diretórios que estiverem imediatamente abaixo de um diretório marcado com o *sticky bit* só poderão ser movidos, removidos ou renomeados pelo usuário que os criou (ou pelo *root*,é lógico)". (MOTA FILHO, 2012, p. 439).

Após dar o comando *ls -l* para verificar as permissões de acesso, o usuário irá se deparar com as letras t maiuscula ou minuscula. A diferença entre elas é que o ‘t’ minúsculo representa a permissão de execução, e o ‘T’ maiúsculo não tem. Isso acontece devido a aparição desse caractere ocorrer no lugar onde deveria aparecer o caractere de execução.

6. SELINUX - SECURITY ENHANCED LINUX

SELinux é uma arquitetura de segurança que utiliza o *Linux Security Modules* (*LSM*). Essa arquitetura de segurança foi desenvolvida pela *Red Hat* em parceria com a Agência de segurança americana (*NSA*) e incorporada ao *Linux* na versão 2.6 do *kernel* em 2003.

Por padrão, o *Linux* utiliza o controle de acesso discricionário (*DAC*). Mas além dele, o *Linux* disponibiliza um sistema de segurança chamado *SELinux*. O *SELinux* trabalha em cima do modelo de controle de acesso mandatório (*MAC*), que define uma camada a mais de segurança, permitindo restringir ainda mais o acesso. O *SELinux* pode definir políticas de segurança que restrinjam até mesmo o usuário *ROOT* do sistema, além de guardar logs de todos os eventos que ocorreram no sistema. O *SELinux* não desabilita o modelo *DAC*.

> *SELinux* define os direitos de acesso e transição de todos os usuários, aplicações, processos e arquivos no sistema. *SELinux* rege as interações dessas entidades usando a política de segurança que especifica o quão rigorosa ou branda uma determinada instalação do *Red Hat Enterprise Linux* deveria ser. (RED HAT, 2018).

No *CentOS 7*, o *SELinux* já vem, por padrão, habilitado e em modo '*Enforcing*'. Com o comando *sestatus* é possível descobrir se o *SELinux* está habilitado ou não no sistema conforme é apresentado na transcrição a seguir. Há três status possíveis para o *SELinux*:

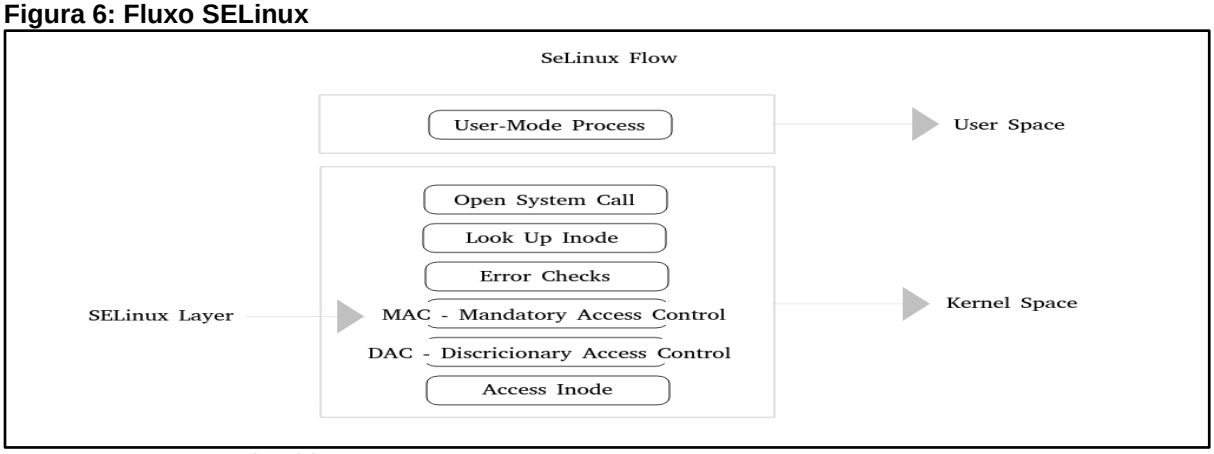
- O modo "*enforcing*", que significa que o *SELinux* irá monitorar o sistema e bloquear qualquer ação suspeita, além de gerar *logs* de todos os eventos.
- O modo "*permissive*", que faz com que o *SELinux* realize a auditoria do sistema, porém sem bloquear nenhuma ação.
- E é possível desabilitá-lo completamente no modo "*disable*". É sempre bom reforçar que desabilitar o *SELinux* não é recomendado em ambiente de produção.

```
[admin@localhost ~]$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
```

```
Max kernel policy version:      31
[admin@localhost ~]$
```

Mantendo-se o *SELinux* habilitado, garante-se uma maior restrição no acesso ao sistema. Por exemplo, se um servidor *web* com o *SELinux* habilitado é comprometido por um atacante, o mesmo somente conseguirá obter acesso ao diretório do *site*, pois o *SELinux* irá restringir o acesso do *user* do servidor *web* através da sua assinatura. Quando algum arquivo ou diretório recebe uma tentativa de acesso, o *shell* do sistema irá realizar uma chamada ao *kernel*, que por sua vez, irá verificar a existência do arquivo e suas respectivas permissões, vide figura 6.

Figura 6: Fluxo SELinux



Fonte: Autor Desconhecido

7. POLÍTICAS DE BACKUP E RECUPERAÇÃO DE DADOS

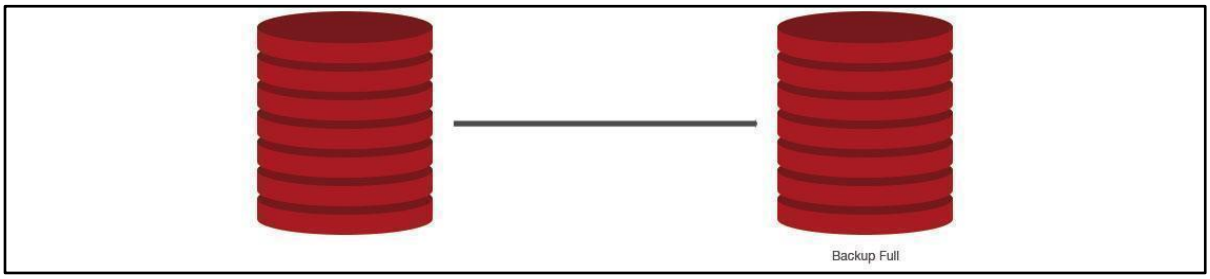
Ter uma política de *backup* bem definida é extremamente importante no tocante a garantir a preservação dos dados de uma empresa, visto que podem ocorrer desastres naturais, quebras de sistema, exclusões de arquivos ou bases de dados de forma acidental ou mesmo intencional. Políticas de *backup* garantem que a empresa consiga se recuperar de um desastre de forma rápida e com o mínimo de impacto possível aos *stakeholders*.

O primeiro passo ao elaborar uma política de *backup* é fazer a análise dos dados para que se possa mensurar a quantidade total dos dados, o tempo de vida dos dados e o método que será utilizado para realizar o *backup e* definir o que são dados comuns, dados críticos, lixo digital. Este alinhamento deve ser realizado pela equipe de T.I. em conjunto com área responsável pelo negócio da empresa, que deve elaborar, implantar e realizar a manutenção dessa política. Os arquivos que denominamos como lixo eletrônico, são arquivos que não são informações importantes para a continuidade do negócio.

Há três tipos de *backups* que podem ser realizados, o completo, o incremental e o diferencial. Cada um deles tem uma finalidade específica e devem ser utilizados em conjunto para que se possa construir uma política de *backup* forte e ao mesmo tempo objetiva.

- **Backup Completo** - O *backup* completo, como o próprio nome já revela, se trata de *backup* total dos arquivos de um servidor (Figura 7). Este tipo de *backup* deve ser realizado periodicamente, para que sirva como base para os demais tipos de *backup*.

Figura 7: Backup Completo

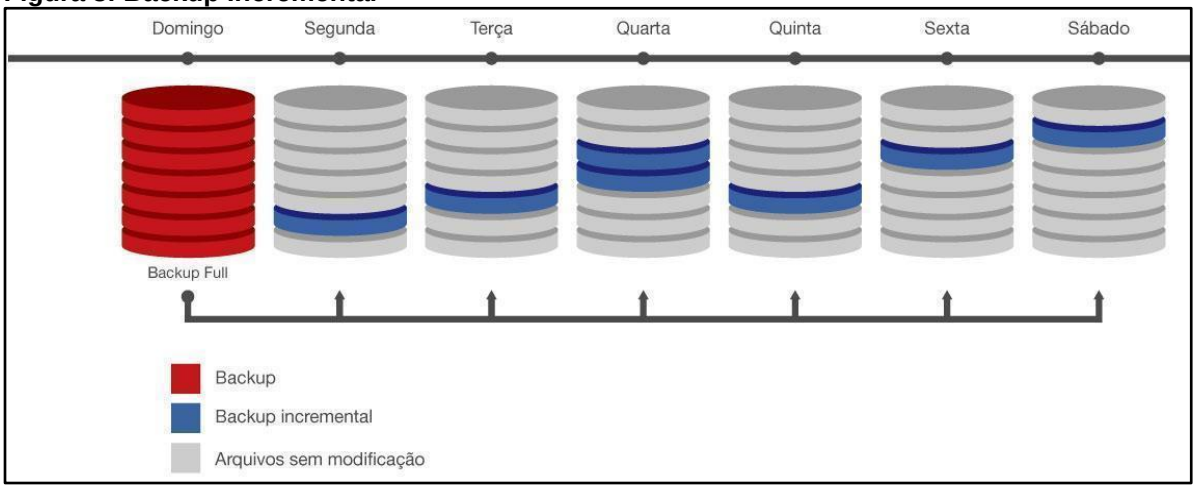


Fonte: CONTROLE.NET (2018)

- **Backup Incremental -** O *backup* incremental é o *backup* que busca armazenar apenas os arquivos e diretórios que foram criados ou modificados desde o último *backup* completo. É um tipo de *backup* menor, mais leve e rápido. Porém necessita de um *backup* completo para que seja utilizável.

  A utilização deste tipo de *backup* deve ser feita com cautela, pois é um tipo de *backup* que grava apenas uma parcela dos dados. Por exemplo, conforme a Figura 8, se o *backup* de segunda, terça e quarta se corromper, o *backup* de quinta se torna ineficaz, pois os dados de três dias serão perdidos.

Figura 8: Backup Incremental

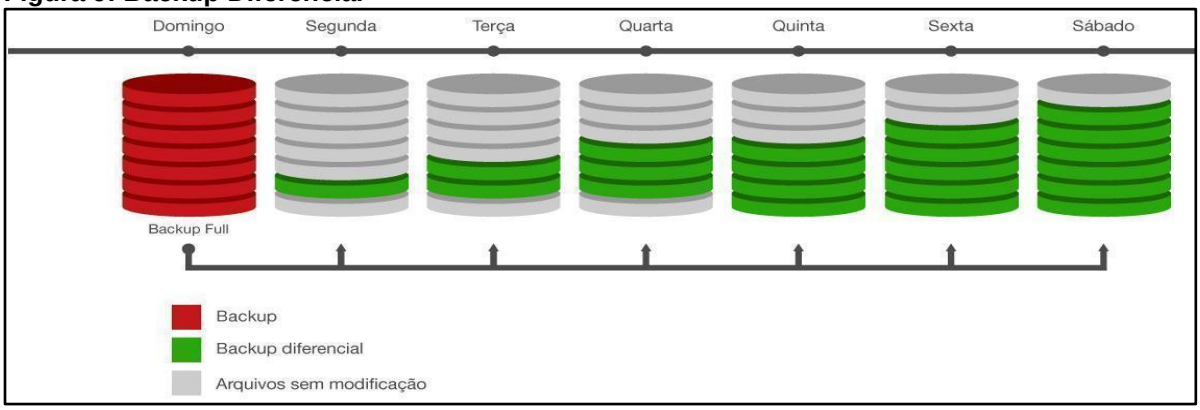


Fonte: CONTROLE.NET (2018)

- **Backup Diferencial -** O *backup* diferencial é um tipo de cópia de segurança que armazena, assim como o *backup* incremental, os dados criados ou modificados após o último *backup* total, porém armazena também os dados do último *backup* diferencial, o que o torna muito mais efetivo, se comparado com o *backup* incremental.

Assim como é demonstrado na figura 9, com a combinação do uso do *backup* total, com o *backup* diferencial, em casos de desastre, para a retomada da operação, basta que se restaure o último *backup* total e o último *backup* diferencial.

Figura 9: Backup Diferencial



Fonte: CONTROLE.NET (2018)

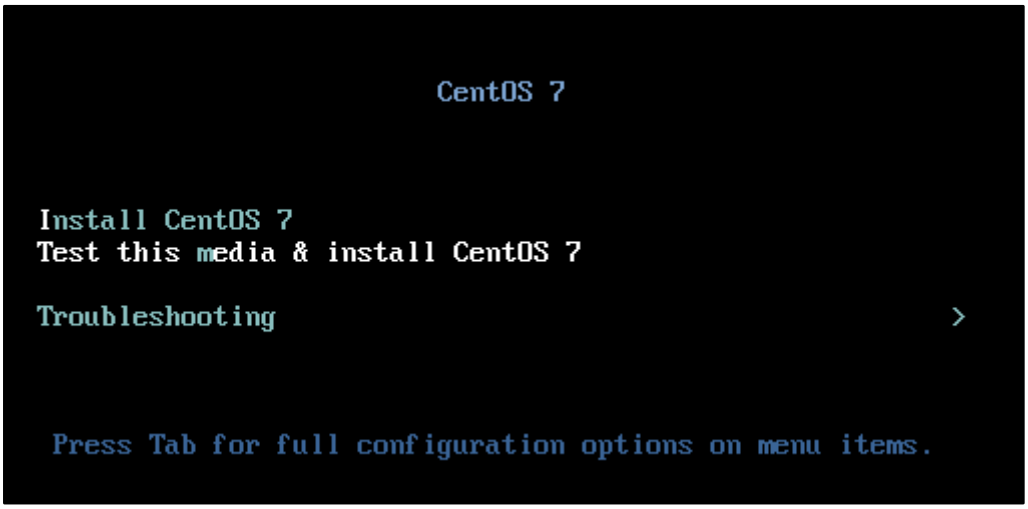
- **Plano de Recuperação de Desastres -** O PRD (Plano de Recuperação de Desastres) indispensável para que se garantir a continuidade do negócio. É necessário que haja um plano pré-estabelecido a respeito dos passos a serem tomados, caso ocorra algum desastre natural, como terremotos ou incêndios, para garantir a continuidade do negócio. Mesmo que a empresa possua um bom plano de *backup*, uma boa política de segurança, funcionários bem treinados, sistemas atualizados e bem configurados, se a empresa não tiver a capacidade de voltar a operar normalmente de forma rápida após um desastre, a mesma terá imensos prejuízos e pode até mesmo abrir estado de falência.

> A recuperação de desastres é a capacidade de reiniciar operações de TI completas dentro de um período de tempo ou objetivo de tempo de recuperação (RTO) específico e em determinado ponto no processo de TI ou objetivo de ponto de recuperação (RPO). Um RTO mais curto indica um tempo de recuperação mais rápido, enquanto um RPO menor indica que menos transações e informações se perderam durante a paralisação não planejada. (BRAZIL EMC, 2018).

8. INSTALAÇÃO E CONFIGURAÇÃO INICIAL

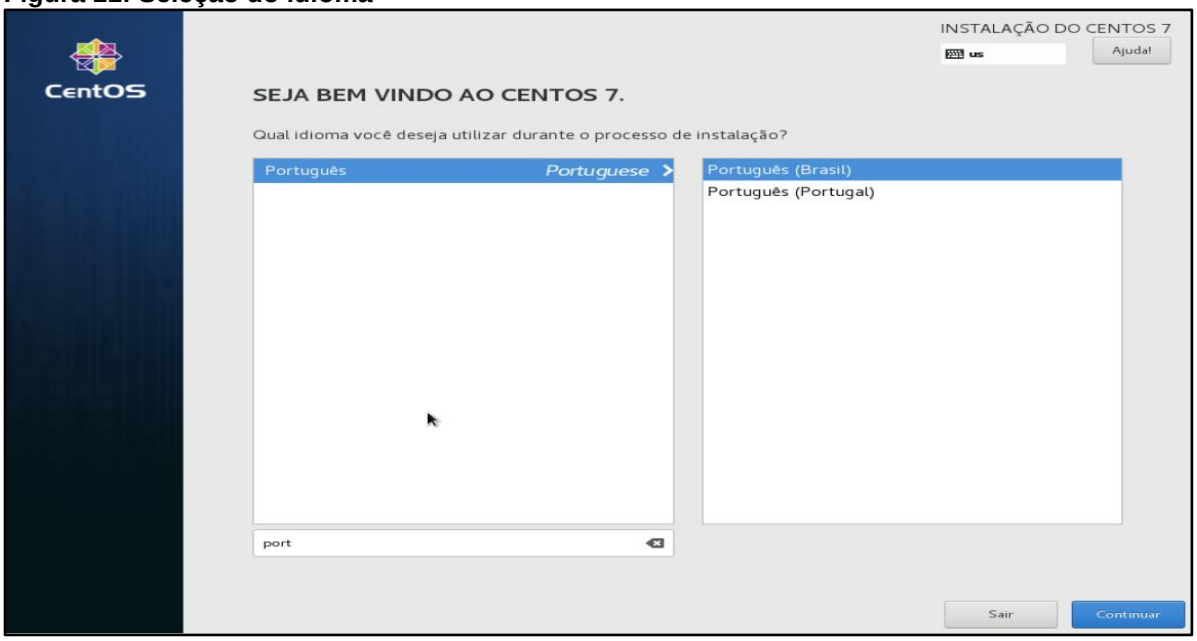
Há um longo caminho para que se obtenha um ambiente computacional consideravelmente seguro. Já na instalação do sistema operacional, há alguns pontos que temos que nos atentar para elevarmos a segurança do ambiente. Quando se pensa em servidores, independente da finalidade do servidor, temos que manter em foco a economia de recursos e o controle de acesso. Baseado nesses focos, neste trabalho será utilizado a versão *minimal* da distribuição *CentOS*. Após realizar o carregamento da mídia de instalação do sistema (Figura 10) e a configuração do idioma (Figura 11), o sistema exibe a tela principal da instalação (figura 12).

Figura 10: Tela inicial do disco de instalação do CentOS



Fonte: Próprio Autor

Figura 11: Seleção do idioma



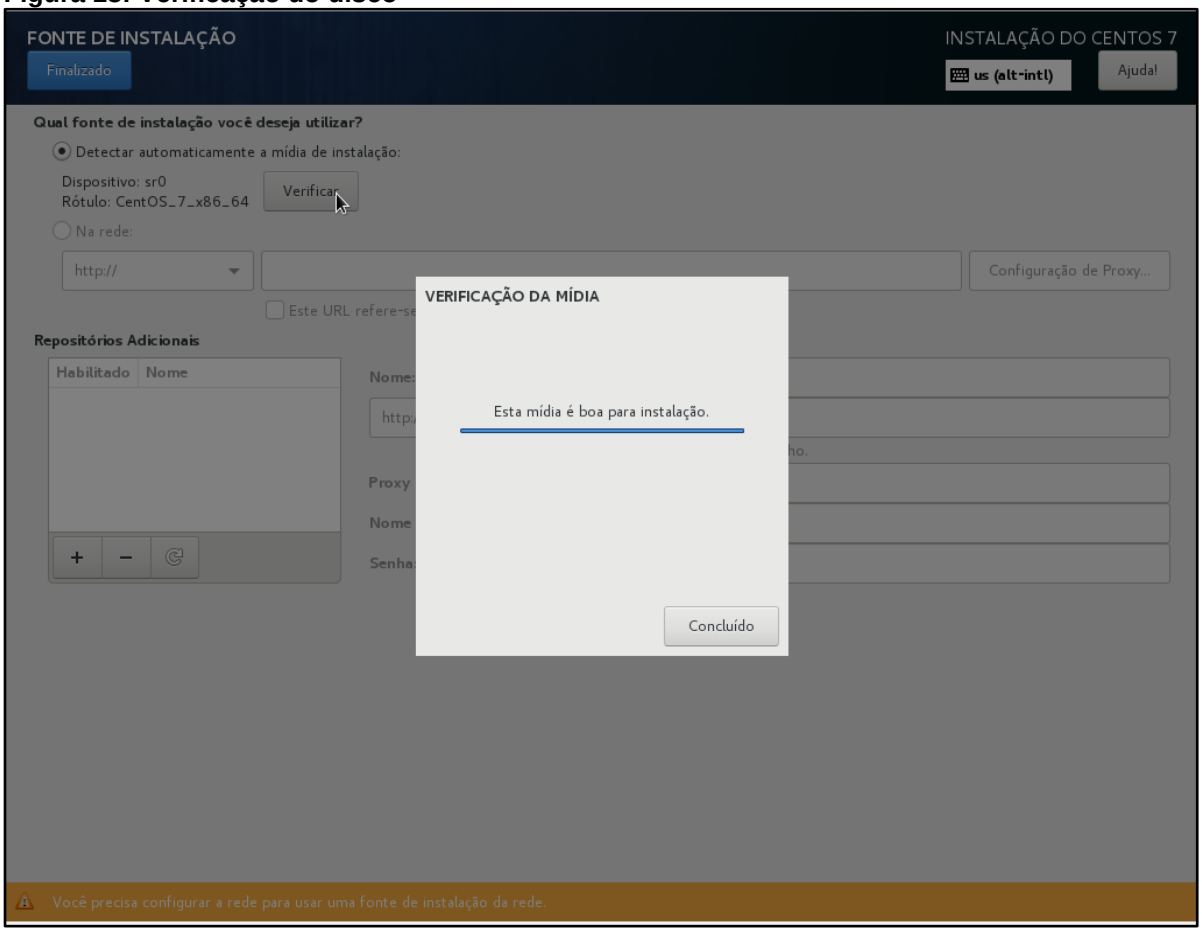
Fonte: Próprio Autor

Figura 12: Tela principal do disco de instalação

Fonte: Próprio Autor

Na tela principal da mídia de instalação, é apresentada as opções de configuração do sistema no momento da pré-instalação. Alguns deles são triviais, como data e hora, idioma e configuração do teclado, porém há alguns que merecem um pouco de atenção. Primeiramente será necessário verificar se a mídia de instalação não está corrompida, para isso é necessário ir até o menu “FONTE DE INSTALAÇÃO” (Figura 12) e clicar em verificar (Figura 13).

Figura 13: Verificação do disco



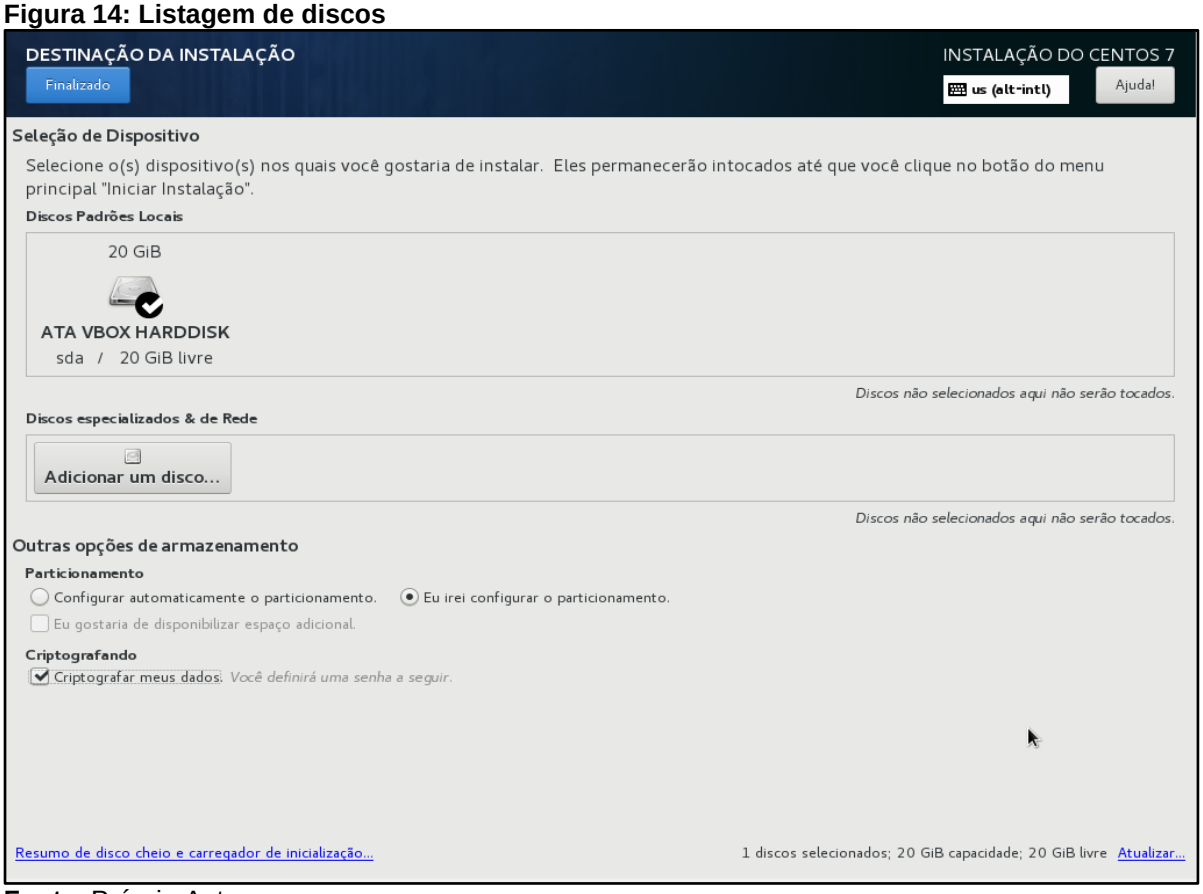
Fonte: Próprio Autor

Um dos pontos mais importantes ao instalar um sistema operacional é o particionamento do disco. Não é recomendado que se utilize o particionamento padrão para realizar a instalação do sistema, pois o mesmo irá separar apenas a */home* do /. Ao se realizar o particionamento manual, tem-se maior liberdade e maior controle sobre o sistema. Deve-se sempre trabalhar com o disco criptografado (Figura 14 e 15), se atentar muito na questão da divisão do espaço de armazenamento entre as partições pois este é um ponto crucial da gestão de recursos de um equipamento. Saber particionar o disco, pode auxiliar na recuperação em casos de quebras do sistema ou até mesmo evitar que uma

partição importante fique sem espaço, enquanto outra, que não precise tanto, tenha espaço de sobra. Conforme a Figura 16, é possível averiguar um exemplo de particionamento pensado para um servidor de aplicações *web*. Onde a partição */var* possui a maior parte do disco, mais até do que o */*, pois nela irá conter as aplicações que aquele servidor irá hospedar. Outra partição que é recomendável que se mantenha separada é a */var/log*, pois é o diretório que armazena os *logs* do sistema, segundo a documentação do *CentOS*.

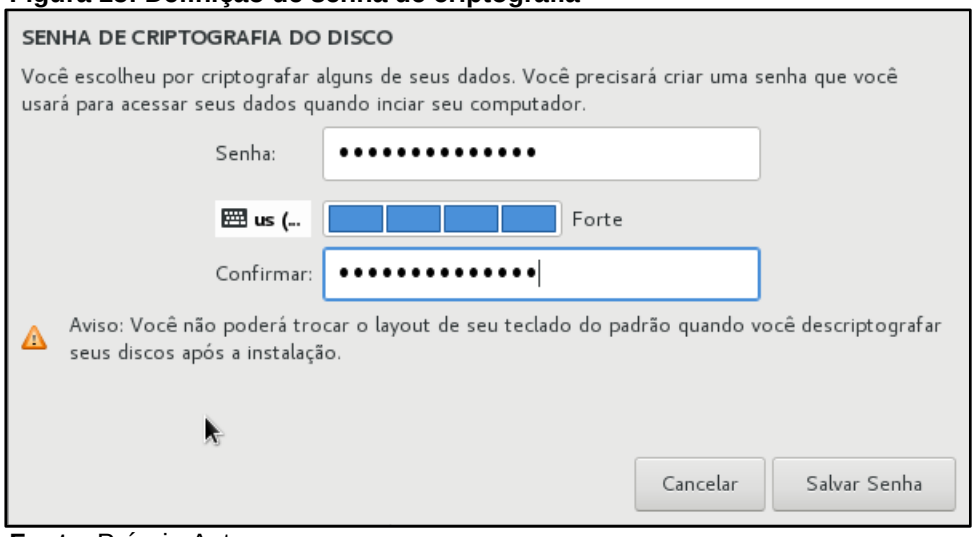
Além de auxiliar na escalabilidade do sistema, separar as partições de acordo com a necessidade ajuda bastante na hora de se recuperar de uma quebra inesperada do sistema, pois em muitos casos o conteúdo relativo ao negócio está separado em uma partição diferente do sistema.

Figura 14: Listagem de discos



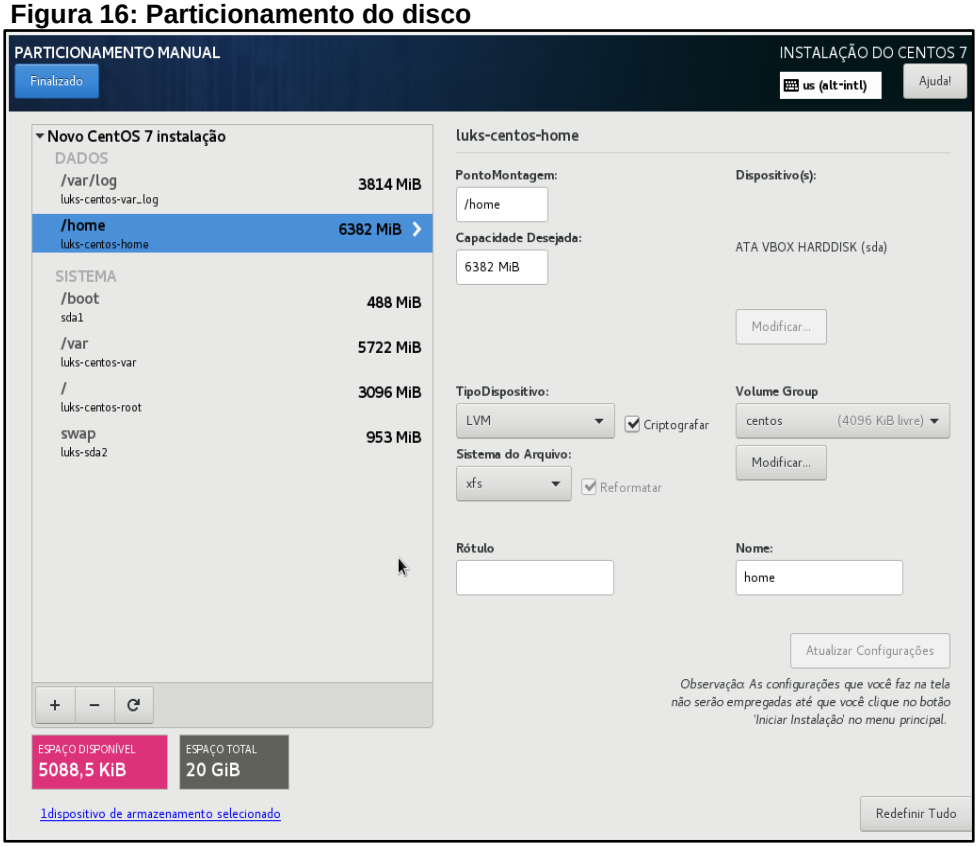
Fonte: Próprio Autor

Figura 15: Definição de senha de criptografia



Fonte: Próprio Autor

Figura 16: Particionamento do disco

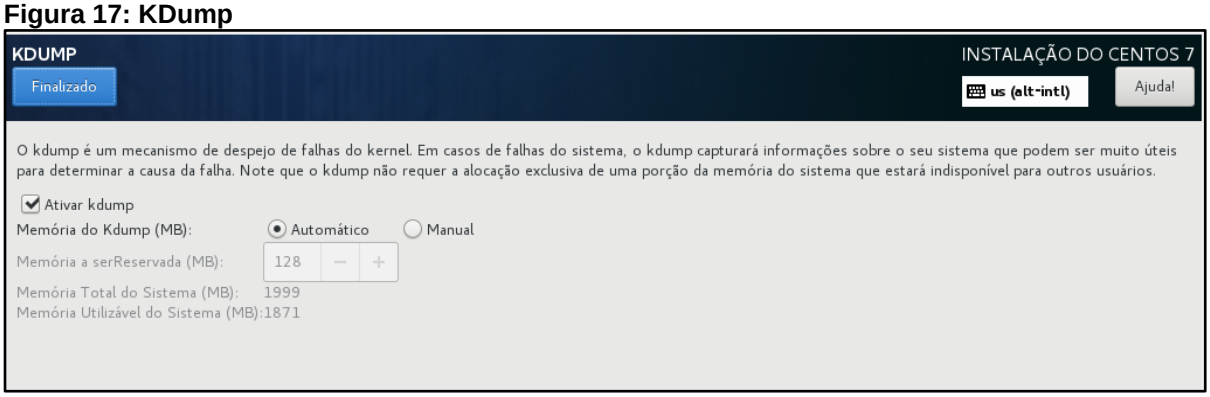


Fonte: Próprio Autor

Distribuições como o *CentOS* possuem uma ferramenta chamada *KDUMP* (Figura 17), que, segundo a própria *Red Hat*, é um mecanismo de despejo de falhas do *kernel*.

Caso aconteça um travamento de sistema, o *Kdump* captura informações do seu sistema que podem ser valiosas ao determinar a causa do travamento. Habilitar este recurso ajuda a auditar possíveis quebras do sistema.

Figura 17: KDump



Fonte: Próprio Autor

9. PRIMEIROS PASSOS NO SISTEMA

Antes de se instalar as aplicações e ferramentas necessárias para que o servidor possa operar na rede, deve-se realizar alguns pequenos passos para torná-lo um pouco mais seguro.

O primeiro passo que se deve realizar sempre que se instalar um sistema operacional em uma máquina, é atualizá-lo. Parece óbvio, mas pontos como esse passam despercebidos muitas vezes e podem acarretar consequências significativas sob o contexto de segurança, como vulnerabilidades de *zero day* por exemplo.

> *Zero-day ou 0day* é uma expressão recorrente quando o assunto é vulnerabilidade grave em *softwares* e sistemas operacionais. O termo costuma ser aplicado em duas situações: quando brechas graves de segurança são encontradas e quando ataques de *hackers* explorando essas brechas são identificados. (TECHTUDO, 2017).

Deve-se sempre manter o sistema atualizado e com as últimas versões dos *softwares* instalados para que se possa garantir que haja o mínimo de vulnerabilidades no sistema, conforme é demonstrado na transcrição a seguir.

```
[root@localhost admin]# yum update
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: ftp.unicamp.br
 * extras: ftp.unicamp.br
 * updates: ftp.unicamp.br
No packages marked for update
[root@localhost admin]#
```

A segunda coisa a ser feita é bloquear o *firewall* do sistema. Independentemente de sua rede interna estar protegida por um *firewall* na borda da rede, é importante que cada máquina tenha o *firewall* configurado e permitindo o tráfego de dados somente do que for necessário. Continuando o exemplo de um servidor *WEB*, inicialmente pode-se deixar apenas as portas relativas à conexão *FTP*, *HTTP*, *HTTPS* e de *DATABASE* liberadas e o resto das portas bloqueadas. Por se tratar de um servidor *WEB*, a princípio não há necessidade de se liberar outras portas, evitando assim vulnerabilidades desnecessárias. Por padrão, o *CentOS 7* utiliza o

FirewallD para gerir o as configurações de *firewall* substituindo os antigos *scripts* do *iptables*. O *FirewallD* trabalha com o conceito de zonas, permitindo uma maior flexibilidade na configuração do *firewall*. O *FirewallD* facilita a configuração do *firewall* e torna mais simples o seu entendimento, conforme a transcrição a seguir.

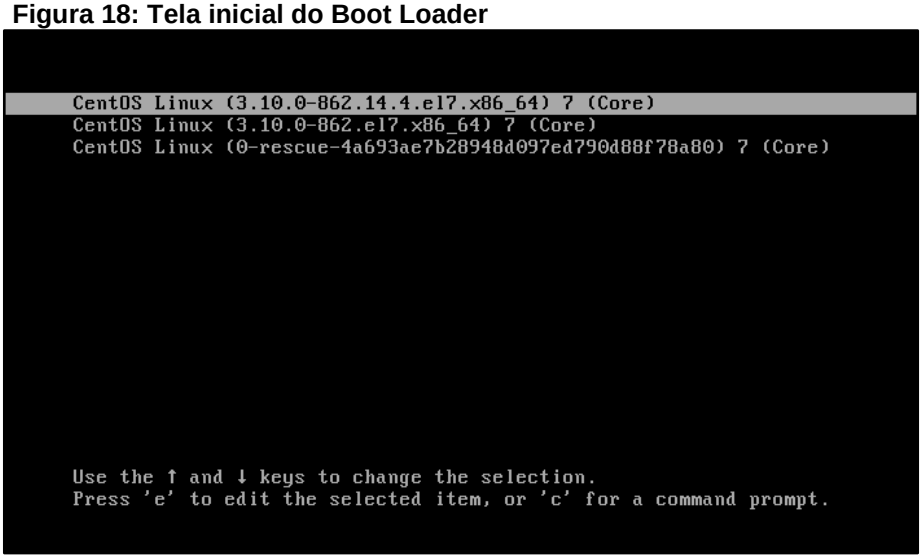
```
[root@localhost admin]# firewall-cmd --get-default-zone
drop
[root@localhost admin]# firewall-cmd --zone=drop --add-service=http --
permanent
success
[root@localhost admin]# firewall-cmd --zone=drop --add-service=https --
permanent
success
[root@localhost admin]# firewall-cmd --zone=drop --add-service=ftp --
permanent
success
[root@localhost admin]# firewall-cmd --zone=drop --add-service=mysql --
permanent
success
[root@localhost admin]# firewall-cmd --reload
success
[root@localhost admin]# firewall-cmd --zone=drop --list-all
drop (active)
  target: DROP
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: ftp http https mysql
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost admin]#
```

Deve-se configurar uma senha para o acesso ao *Boot Loader* do sistema, conforme é demonstrado na transcrição abaixo, a fim de que, se alguém obtiver acesso físico, de forma indevida, ao servidor, não consiga manipular os parâmetros passados ao *kernel* em tempo de execução (Figura 18 e 19). Caso os parâmetros de inicialização do *kernel* estejam desprotegidos, o atacante poderá manipular a inicialização do sistema, permitindo que ele obtenha acesso ao sistema sem a necessidade de um usuário.

```
[root@localhost admin]# grub2-setpassword
```

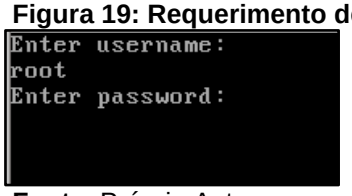
```
Enter password:
Confirm password:
[root@localhost admin]# grub2-mkconfig
```

Figura 18: Tela inicial do Boot Loader



Fonte: Próprio Autor

Figura 19: Requerimento de senha do GRUB



Fonte: Próprio Autor

Figura 20: Requerimento da senha de descriptografia de disco

Fonte: Próprio Autor

Ao definir a criptografia de disco, o usuário que tiver acesso físico a máquina, terá uma barreira a mais, caso tenha o intuito de mudar algum parâmetro na inicialização do S.O.

Caso um usuário mal intencionado, tenha acesso físico a máquina, e tente fazer qualquer mudança nos parâmetros de inicialização do S.O., será solicitado a senha para descriptografia de disco (Figura 20). Essa senha será uma barreira a mais de segura.

9.1. TUNNING DO SSH

O primeiro passo é alterar o arquivo de configuração do *SSH*, o */etc/ssh/sshd_config*. Nele, pode-se bloquear o acesso remoto pelo usuário *root* e pode-se também alterar a porta padrão do *SSH*, o que pode ajudar a aumentar a segurança nos servidores, pois dificulta um possível ataque, visto que a porta padrão é a 22, o atacante primeiramente tentará o ataque por ela. No caso, a porta será alterada para a 10022, conforme é demonstrado na transcrição a seguir.

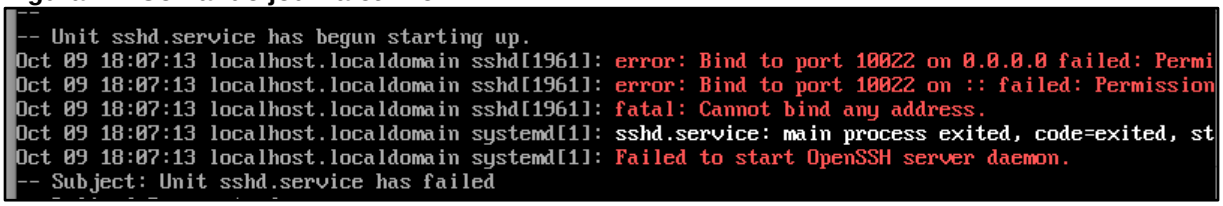
```
# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
Port 10022
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
[ … ]
# Authentication:
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

Alterando a porta padrão do serviço, o S.O. apresentará uma mensagem de erro do *SELinux* ao tentar reiniciar o serviço do *SSH* para aplicar as alterações na configuração conforme é evidenciado na transcrição a seguir.

```
[root@localhost admin]# systemctl restart sshd
Job for sshd.service failed because the control process exited with error
code. See "systemctl status sshd.service" and "journalctl -xe" for
details.
[root@localhost admin]#
```

Através do comando *journalctl -xe,* pode-se verificar com mais detalhes o que ocasionou o erro. No caso do exemplo da figura 21, o erro é '*...Bind to port 10022…'.* Através dos *logs*, pode-se verificar que o *SELinux* bloqueou a inicialização do serviço, por não reconhecer a porta.

Figura 21: Comando journalctl -xe



Fonte: Próprio Autor

Conforme mostrado na transcrição do arquivo de configuração do *SSH* no começo da seção 9.1, é indicado o procedimento necessário para a alteração da porta no *SELinux*. O *semanage* é uma ferramenta que pertence ao pacote *setools* (Pacote com ferramentas para o gerenciamento do *SELinux*). Por padrão, o *setools e o setroubleshoot (*ferramenta para o auxílio no *troubleshoot)* não vem instalados no sistema, conforme mostrado a seguir.

```
[root@localhost admin]# semanage
bash: semanage: command not found
```

```
[root@localhost admin]# yum search setools
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: ftp.unicamp.br
 * extras: ftp.unicamp.br
 * updates: ftp.unicamp.br
================================================================================
=========================================     N/S     matched:     setools
================================================================================
=========================================
setools.x86_64 : Policy analysis tools for SELinux
setools-console.x86_64 : Policy analysis command-line tools for SELinux
setools-devel.i686 : Policy analysis development files for SELinux
setools-devel.x86_64 : Policy analysis development files for SELinux
```

```
setools-gui.x86_64 : Policy analysis graphical tools for SELinux
setools-libs.i686 : Policy analysis support libraries for SELinux
setools-libs.x86_64 : Policy analysis support libraries for SELinux
setools-libs-tcl.x86_64 : Tcl bindings for SELinux policy analysis

  Name and summary matches only, use "search all" for everything.


[root@localhost admin]# yum search setroubleshoot
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: ftp.unicamp.br
 * extras: ftp.unicamp.br
 * updates: ftp.unicamp.br
============================================================================
=====================================    N/S    matched:    setroubleshoot
============================================================================
=====================================
setroubleshoot-plugins.noarch   :   Analysis   plugins   for   use   with
setroubleshoot
setroubleshoot.x86_64 : Helps troubleshoot SELinux problems
setroubleshoot-server.x86_64 : SELinux troubleshoot server

  Name and summary matches only, use "search all" for everything.
```

Após a instalação dos pacotes mostrados nas transcrições anteriores (*setools* e *setroubleshoot*), é possível executar o comando para alteração da porta do *SSH*. Com isso, já será possível inicializar o serviço conforme é apresentado no exemplo abaixo.

```
[root@localhost admin]# semanage port -a -t ssh_port_t -p tcp 10022
[root@localhost admin]# systemctl restart sshd
[root@localhost admin]# systemctl status sshd
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor
preset: enabled)
   Active: active (running) since Seg 2019-11-18 18:55:36 EST; 5s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 3324 (sshd)
   CGroup: /system.slice/sshd.service
           └─3324 /usr/sbin/sshd -D

Nov 18 18:55:36 localhost.localdomain systemd[1]: Starting OpenSSH server
daemon...
Nov 18 18:55:36 localhost.localdomain sshd[3324]: Server listening on
0.0.0.0 port 10022.
Nov 18 18:55:36 localhost.localdomain sshd[3324]: Server listening on ::
port 1022.
Nov 18 18:55:36 localhost.localdomain systemd[1]: Started OpenSSH server
daemon.
[root@localhost admin]#
```

Porém, para que se possa acessar o serviço do *SSH*, é necessário realizar a liberação da porta 10022 no *firewall* e bloquear a porta padrão que foi liberada anteriormente como é possível ver na transcrição a seguir.

```
[root@localhost admin]# firewall-cmd --zone=drop --add-port=10022/tcp --
permanent
success
[root@localhost admin]# firewall-cmd --complete-reload
success
[root@localhost admin]# firewall-cmd --zone=drop --list-all
drop (active)
  target: DROP
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: ftp http https mysql
  ports: 10022/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@localhost admin]#
```

Após todas essas configurações, é possível se conectar remotamente através da nova porta e não será mais possível a conexão com o usuário *root*. A transcrição a seguir demonstra a conexão via *SSH* através da porta definida anteriormente.

```
rcortez@blackwolf:~$ ssh admin@192.168.100.10 -p 10022

admin@192.168.100.10's password:

Last login: Mon Nov 18 17:43:01 2019 from 192.168.100.3
[admin@localhost ~]$
```

9.2. AJUSTES FINOS

Além das etapas anteriormente apresentadas, existem mais alguns ajustes possíveis para tornar o ambiente mais seguro.

- **REMOVER PACOTES DESNECESSÁRIOS**

Por mais que a instalação do *CentOS 7* no modo *minimal* seja bem enxuta, ainda há aplicativos que podem ser considerados desnecessários para um servidor, dependendo da aplicação ao qual ele for submetido.

Com o comando *rpm -qa,* conforme mostrado a seguir, pode-se listar todos os pacotes instalados no sistema.

```
[root@localhost admin]# rpm -qa
kexec-tools-2.0.15-33.el7.x86_64
setup-2.8.71-10.el7.noarch
authconfig-6.2.8-30.el7.x86_64
kbd-misc-1.15.5-15.el7.noarch
kbd-1.15.5-15.el7.x86_64
nss-softokn-freebl-3.44.0-5.el7.x86_64
nss-util-3.44.0-3.el7.x86_64
cryptsetup-2.0.3-5.el7.x86_64
libsepol-2.5-10.el7.x86_64
passwd-0.79-5.el7.x86_64
[ … ]
```

Utilizando o comando *grep*, conforme mostrado a seguir, pode-se filtrar essa pesquisa para facilitar as buscas.

```
[root@localhost admin]# rpm -qa | grep <package>
[ … ]
```

E então, mandar a saída do comando *grep*, contendo os arquivos que deseja-se excluir, para o comando de remoção (*yum remove pacote) através do comando xargs,* conforme exemplificado a seguir.

```
[root@localhost admin]# rpm -qa | grep <package> | sudo xargs yum remove
-y
[ … ]
```

- **DESABILITAR DETECÇÃO DE USB**

Caso a segurança física da empresa seja violada e o atacante obtiver acesso a sala de servidores, é importante que dispositivos *USB*, como *pendrives* ou HD externas, não sejam detectados automaticamente, pois isso dificulta a ação do invasor. A transcrição a seguir demonstra o comando utilizado para realizar a configuração para a não detecção de dispositivos *USB*.

```
[root@localhost admin]# echo "install usb-storage /bin/true" >
/etc/modprobe.d/no-usb
[root@localhost admin]# cat /etc/modprobe.d/no-usb
install usb-storage /bin/true
[root@localhost admin]#
```

- **DESABILITAR CRON**

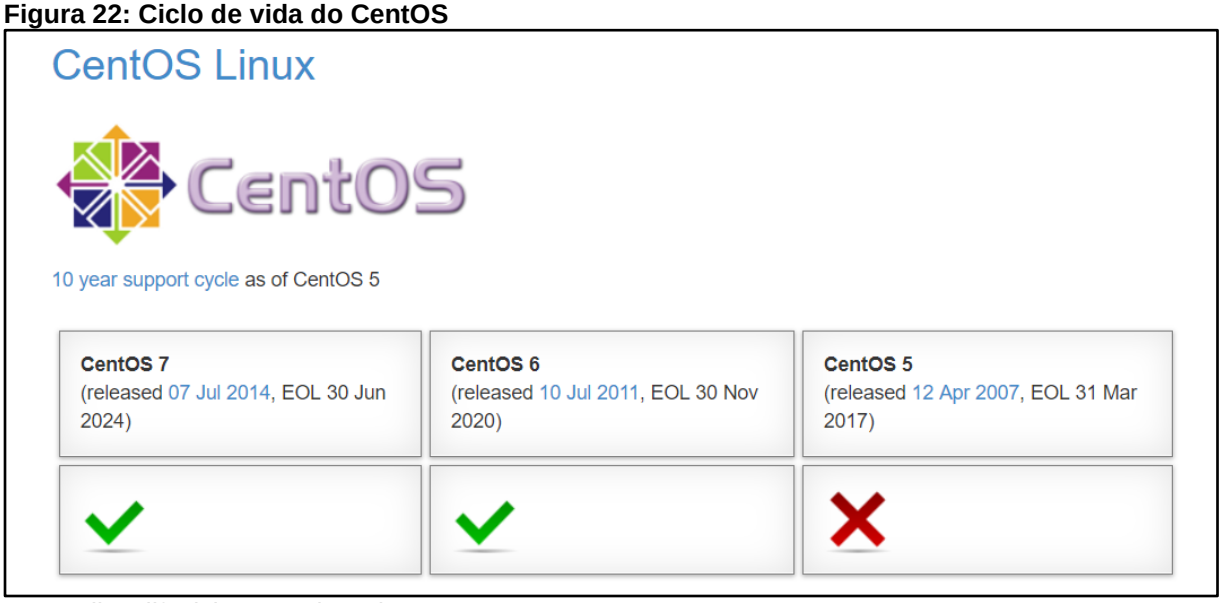
Da mesma maneira, em um ambiente produtivo, é importante que se desabilite o *CRON* (Agendador de tarefas no Linux) da máquina, pois o atacante, caso consiga obter acesso físico ao servidor e consiga conectar um teclado ao mesmo, poderá agendar tarefas no *CRON* que facilite o seu acesso remoto no sistema posteriormente, visto que tarefas agendadas no *CRON*, não irão alertar sistemas de segurança, como o *SELinux*. A transcrição a seguir demonstra o comando para desabilitar a *CRON* da máquina.

```
[root@localhost admin]# cat /etc/cron.deny
ALL
[root@localhost admin]# echo ALL > /etc/cron.deny
[root@localhost admin]#
```

10. CONSIDERAÇÕES FINAIS

O mundo da tecnologia está constantemente sofrendo alterações, e com isso as empresas, de tempos em tempos lançam novas versões de seus sistemas e gradativamente param de manter os sistemas anteriores. O ciclo de vida do *CentOS* é de 10 anos, o que significa que o *CentOS 7* (Figura 22) que foi lançado em 07 de julho de 2014, terá atualizações, melhorias em seu sistema e *patches* de segurança até julho de 2024.

Figura 22: Ciclo de vida do CentOS



Fonte: linuxlifeciclye.com (2018)

Mesmo com um ciclo de vida tão longo, em uma determinada data, o sistema deixará de receber as atualizações de segurança e as correções sistêmicas necessárias para que o S.O. seja seguro para o uso em um ambiente empresarial.

O processo de transição entre um sistema que será descontinuado para o seu sucessor deve ser planejado com antecedência, para que não haja impactos para nenhum dos *stakeholders.*

Apesar do sistema operacional oferecer ferramentas para que o sistema seja atualizado para a nova versão sem a necessidade de uma reinstalação, alguns pacotes podem se corromper, outros pacotes podem ser instalados de forma

desnecessária, enfim, para manter um servidor enxuto e configurado da forma correta, o melhor a se fazer é preparar a transição, e realizar a instalação do novo sistema desde o início. Deve-se verificar quais pacotes serão descontinuados na versão nova do sistema, quais pacotes manterão as versões atuais, e quais virão com versões superiores a que estão instaladas atualmente.

Deve-se comunicar com antecedência todos os desenvolvedores e os donos das aplicações, para que eles tenham tempo o suficiente para adequar os seus sistemas às novas versões dos pacotes que virão com a nova versão do S.O.. O planejamento quanto a execução desta tarefa deve ocorrer, de forma que o impacto para com os usuários seja o menor possível.

A partir da apresentação e análise dos dados, observa-se que, com o decorrer do tempo, manter um ambiente empresarial seguro, requer uma atenção redobrada aos detalhes, pois pequenas vulnerabilidades são suficientes para que estragos enormes aconteçam.

Uma questão importante a se ressaltar neste trabalho é o fato de que nenhum ambiente, por mais seguro que seja, é realmente 100% seguro. Afirmar que um ambiente ou um sistema é totalmente seguro é uma grande falácia e que pode levar corporações inteiras a ruína.

É de inteira responsabilidade do administrador de rede e da equipe de T.I. manter um ambiente seguro. Para tanto, é necessário que o administrador saiba cuidar de todo o ciclo de vida de um servidor, desde o momento de sua instalação, até o dia de sua descontinuação.

É necessário também que toda equipe de T.I. se mantenha sempre atualizada referente a *Softwares*, *Hardwares* e novas tecnologias que possam lhe garantir mais segurança, ganho em desempenho e performance na empresa a fim de manter a melhoria constantemente.

Assim como é citado por Kevin Mitnick no livro A arte de enganar (2003), e com base em dados estatísticos publicados pelo CERT.br, a segurança da informação é um ponto extremamente importante e que deve ser levado em consideração por todos, pois ao colocar um servidor vulnerável na internet, além de expor as informações da empresa, ainda coloca em risco a segurança de toda a internet.

REFERÊNCIAS BIBLIOGRÁFICAS

BRAZIL EMC. **Disaster Recovery**. Dell EMC, 2018, Disponível em: <https://brazil.emc.com/corporate/glossary/disaster-recovery.htm> Acesso em: 10 nov. 2018.

CERT.br (a). **Fundamentos da segurança da informação**. CERT.br, 2014, Disponível em: <https://www.cert.br/docs/palestras/certbr-egi2014.pdf> Acesso em: 31 out. 2018.

CERT.br (b). **Incidentes reportados ao CERT.br**. CERT.br, 2019, Disponível em: <https://www.cert.br/stats/incidentes/2018-jan-dec/tipos-ataque.html> Acesso em: 23 out. 2019.

CERT.br (c). **Estatísticas dos incidentes reportados ao CERT.br**. CERT.br, 2019, Disponível em: <https://www.cert.br/stats/incidentes/> Acesso em: 23 out. 2019.

CONTROLE.NET. **Tipos de backup**: completo ou full, incremental e diferencial. Controle.net, 2018, Disponível em: <https://www.controle.net/faq/tipos-de-backup-o-que-e-backup-full-incremental-e-diferencial> Acesso em: 11 nov. 2018.

GSTI. **Tipos de arquivos em linux**. Portalgsti, 2017, Disponível em: <https://www.portalgsti.com.br/2017/04/tipos-de-arquivos-em-linux.html> Acesso em: 15 out. 2019.

INFOWESTER. **Sistemas RAID** (Redundant Array of Independent Disks), 2012, Disponível em: <https://www.infowester.com/raid.php> Acesso em: 31 out. 2018.

KASPERSKY. **O que são ataques de DDoS?**. Kaspersky, 2018, Disponível em: <https://www.kaspersky.com.br/resource-center/threats/ddos-attacks> Acesso em: 31 out. 2018.

LINUX LIFE CYCLE. **Support life cycles for enterprise linux distributions**. Linux Life Cycle, 2019, Disponível em: <https://linuxlifecycle.com/> Acesso em: 11 nov. 2018.

LYRA, Maurício Rocha. **Governança da segurança da informação**. Brasília: Edição do Autor, 2015.

MOTA FILHO, João Eriberto. **Descobrindo o Linux:** entenda o sistema operacional GNU/Linux 3º Edição. São Paulo: Novatec Editora, 2012.

MITNICK, Kevin D.; SIMON, William L. **A arte de enganar:** Ataques de hackers: controlando o fator humano na segurança da informação. São Paulo: Pearson, 2003.

OPENSOURCE. **GNU general public license version 3**. Opensource.org, 2018, Disponível em: <https://opensource.org/licenses/GPL-3.0> Acesso em: 31 out. 2018.

RED HAT. **Introduction to SELinux**. Chapter 49.2, 2018. Disponível em: <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/ch-selinux> “Traduzido pelo autor” Acesso em: 31 out. 2018.

RED HAT. **LVM** (logical volume manager). Chapter 11, 2018. Disponivel em: <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/c-lvm> Acesso em: 31 out. 2018.

SÊMOLA, Marcos. **Gestão da segurança da informação**: visão executiva da segurança da informação: Rio de Janeiro – Elsevier, 2003.

TECHTUDO. **O que é uma falha zero day?**. Techtudo, 2017, Disponível em: <https://www.techtudo.com.br/noticias/2017/08/o-que-e-uma-falha-zero-day.ghtml>Acesso em: 20 nov. 2018.

TURNBULL, James. **Hardening Linux**. Austrália: Apress, 1º Edição, 2005.