



UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO  
INSTITUTO DE CIÊNCIAS EXATAS  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Luis Felipe Américo Fernandes

**MONITORAMENTO E VIRTUALIZAÇÃO DE SERVIDORES A PARTIR DA  
NORMA ISO 27002**

SEROPÉDICA

2018



LUIS FELIPE AMÉRICO FERNANDES

**MONITORAMENTO E VIRTUALIZAÇÃO DE SERVIDORES A PARTIR DA  
NORMA ISO 27002**

Monografia Apresentada a Banca Examinadora da UFRRJ, como requisito para obtenção do título de Graduado em Sistemas de Informação, sob a orientação do professor Luiz Maltar Castello Branco.

SEROPÉDICA  
2018

**UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO**

**INSTITUTO DE CIÊNCIAS EXATAS**

**DEPARTAMENTO DE COMPUTAÇÃO**

**COORDENAÇÃO DO CURSO DE GRADUAÇÃO EM SISTEMAS  
DE INFORMAÇÃO**

Monografia “MONITORAMENTO E  
VIRTUALIZAÇÃO DE SERVIDORES A  
PARTIR DA NORMA ISO 27002”  
apresentada e defendida por Luis Felipe  
Américo Fernandes matrícula 2012390151  
foi aprovada pela Banca Examinadora, com  
conceito “ ”, recebendo o número .

Seropédica , de de .

BANCA EXAMINADORA:

---

DsC Luiz Maltar Castello Branco

Orientador

---

MSc Nilton José Rizzo

---

MSc Tiago Cruz de França

SEROPÉDICA

2018

## AGRADECIMENTOS

Primeiramente, agradeço a Deus por todas as alegrias e oportunidades.

Agradeço à minha esposa; sem ela a minha vida não seria a mesma e eu não teria chegado até aqui.

Agradeço aos meus pais por todo apoio, carinho, incentivo e ajuda na minha trajetória.

Agradeço a todos os meus professores, em especial, ao professor Nilton José Rizzo que me orientou durante toda a graduação, transmitindo seus conhecimentos de uma forma peculiar, e ao professor Luiz Maltar Castello Branco por toda força e empenho na transmissão de seus ensinamentos e pelos pontos extras durante as aulas.

Agradeço ao amigo Hugo Rogério Borges de Freitas por todos os ensinamentos e oportunidades que contribuíram demasiadamente para o meu crescimento profissional.

Agradeço aos amigos que caminharam ao meu lado nessa jornada, auxiliando nos momentos difíceis e compartilhando experiências.

*Não tentes ser bem sucedido; tenta antes ser um homem de valor.*  
Albert Einstein

## RESUMO

Diversas mudanças técnicas, sociais e culturais foram provocadas pela globalização tecnológica e hoje a informação é o principal capital econômico. Encontramo-nos, portanto, em um panorama repleto de desafios que dizem respeito à Segurança da Informação, ao gerenciamento complexo, às soluções inovadoras e livres que possam ser utilizadas para potencializar e assegurar um Sistema de Informação em seu aspecto mais amplo. Esta pesquisa, portanto, tem como objetivo apresentar uma solução possível criada a partir da norma NBR ISO 27002, com o objetivo de mostrar a possibilidade de implementação de controles de segurança com software livre, incrementando os níveis de segurança e disponibilidade recomendáveis para servidores em organizações de qualquer porte e segmento.

*Palavras-chave: Segurança da Informação, NBR ISO 27002, Software livre*

## ABSTRACT

Several technical, social and cultural changes were brought about by technological globalization and today the information is the main economic capital. We find ourselves, therefore, in a panorama full of challenges related to Information Security, complex management, innovative and free solutions that can be used to enhance and secure an Information System in its broadest aspect. This research, therefore, aims to present a possible solution created from the NBR ISO 27002 standard, with the objective of showing the possibility of implementing security controls with free software, increasing the levels of security and availability recommended for servers in organizations of any size and segment.

*Keywords: Information Security, NBR ISO 27002, Free Software*

## LISTA DE FIGURAS

Figura 1. Normas de Gestão de Segurança da Informação .....	20
Figura 2: Síntese das principais temáticas abordadas pela norma NBR ISO 27002 .....	22
Figura 3. Arquiteturas de virtualização: total e paravirtualização .....	25
Figura 4. Funcionamento do <i>Hypervisor</i> .....	27
Figura 5: Etapas iniciais da instalação do <i>XenServer</i> .....	31
Figura 6. Configurações iniciais para instalação do <i>XenServer</i> .....	32
Figura 7. Finalização das configurações básicas e confirmação da instalação do <i>XenServer</i> .....	33
Figura 8. Finalização da Instalação do <i>XenServer</i> .....	33
Figura 9. Adicionando um servidor de virtualização no <i>XenCenter</i> .....	34
Figura 10. Seleção de <i>template CentOS</i> .....	36
Figura 11. Selecionando a Instalação a partir da ISO .....	36
Figura 12. Seleção do repositório .....	37
Figura 13. Configurações para os discos para um esquema especial de montagem .....	38
Figura 14. Conversão da máquina em <i>template</i> .....	52
Figura 15. Criação de uma nova máquina virtual .....	53
Figura 16. Escolhendo o template para a criação da nova máquina virtual .....	53
Figura 17. Verificação do certificado gerado na instalação .....	70
Figura 18. Tela inicial da instalação do <i>Zabbix</i> .....	70
Figura 19. Verificação dos pré-requisitos da instalação do <i>Zabbix</i> .....	71
Figura 20. Configuração do banco de dados .....	71



Figura 21. Configuração do servidor <i>Zabbix</i> .....	72
Figura 22. Verificação da configuração do <i>Zabbix Dashboard</i> .....	72
Figura 23. Configuração da regra de descoberta na rede .....	75
Figura 24. Aba de criação da ação de descoberta .....	75
Figura 25. Criação da <i>trigger</i> para a regra de descoberta .....	76
Figura 26. Associação dos <i>templates</i> para a regra de descoberta .....	76
Figura 27. <i>Hosts</i> encontrados pela regra de autobusca .....	77
Figura 28. Criação do <i>script</i> de verificação de atualizações para Linux .....	80
Figura 29. Criação da <i>trigger</i> para verificação de atualizações para Linux .....	81
Figura 30. Gráfico de interrupção na conexão de rede .....	82
Figura 31. Histórico de interrupções de rede .....	82
Figura 32. Gráfico de anomalia de utilização de recurso .....	83
Figura 33. Gráfico de saturação de uma maquina virtual .....	83
Figura 34. Tráfego de rede em <i>eth0</i> .....	84
Figura 35. Alertas no <i>Dashboard</i> .....	84

## LISTA DE TABELAS

Tabela 1. Comparação entre ferramentas de virtualização .....	26
Tabela 2. Comparação entre ferramentas de monitoramento.....	29

## LISTA DE SIGLAS E ABREVIACES

ABNT – Associao Brasileira de Normas Tcnicas

IEC – *International Engineering Consortium*

ISO – *International Organization for Standardization*

NBR – Norma Brasileira

CentOS - *Community Enterprise Operating System*

NMS – *Netware Management System*

IDC – *International Data Corporation*

## SUMÁRIO

<b>1. INTRODUÇÃO: SISTEMAS DE INFORMAÇÃO, VALOR E ESTRATÉGIA.....</b>	<b>12</b>
1.1 Segurança da Informação.....	12
1.2 A utilização de servidores virtualizados .....	13
1.3 Soluções livres .....	15
1.4 Motivação .....	16
1.5 Definição do problema .....	16
1.6 Hipótese .....	16
1.7 Objetivo .....	17
1.8 Justificativa .....	17
1.9 Trabalhos relacionados .....	18
1.10 Procedimentos metodológicos e Organização do trabalho	18
<b>2. CONCEITOS BÁSICOS .....</b>	<b>20</b>
2.1 A norma NBR ISO 27002.....	20
2.2 Virtualização .....	23
2.3 Monitoramento .....	28
<b>3. PREPARAÇÃO E CONFIGURAÇÕES UTILIZADAS .....</b>	<b>31</b>
3.1 Instalação do <i>Xenserver</i> .....	31
3.2 Instalação do <i>Xencenter</i> .....	34
3.3 Criação do <i>template</i> da máquina virtual .....	35
3.4 Configuração do <i>template</i> e adequação segundo a norma NBR ISO 27002 .....	39
3.4.1 Configuração inicial da rede .....	39
3.4.2 Instalações extras .....	39
3.4.3 Instalação do <i>zabbix-agent</i> .....	40

3.4.4 Modificação dos pontos de montagem do sistema operacional e remoção da permissão de execução dos arquivos binários .....	42
3.4.5 Configuração do <i>firewall</i> e do <i>SELinux</i> .....	44
3.4.6 Remoção do terminal padrão dos usuários .....	41
3.4.7 Modificação das portas padrão .....	46
3.4.8 Bloqueio do acesso ao <i>root</i> via SSH .....	47
3.4.9 Bloqueio de acesso aos <i>sockets</i> por <i>hosts</i> .....	48
3.4.10 Tempo máximo de acesso do usuário sem interação .....	50
3.5 Conversão da máquina virtual em um novo modelo .....	51
3.6 Criação de novas máquinas a partir do <i>template</i> de segurança.....	52
3.7 Configuração do servidor de banco de dados .....	53
3.7.1 Configuração da rede na máquina virtual .....	54
3.7.2 Instalação do servidor de banco de dados .....	54
3.7.3 Criação do usuário para a aplicação .....	55
3.7.4 Configuração dos arquivos de segurança do <i>PostgreSQL</i> .....	56
3.7.5 Liberação do <i>SELinux</i> e do <i>firewall</i> para a comunicação com o banco de dados .....	57
3.7.6 Verificação do funcionamento do serviço do <i>PostgreSQL</i> .....	58
3.8 Configuração do servidor <i>Zabbix</i> .....	59
3.8.1 Configuração inicial do servidor .....	59
3.8.2 Instalação das dependências .....	59
3.8.3 Compilando o servidor <i>Zabbix</i> .....	60
3.8.4 Configurações iniciais do servidor <i>Zabbix</i> .....	61

3.9 Configuração do <i>dashboard</i> do <i>Zabbix</i> .....	63
3.9.1 Configuração inicial <i>Dashboard</i> .....	63
3.9.2 Instalação dos pacotes para o servidor <i>web</i> .....	63
3.9.3 Extração dos arquivos do Dashbord para a pasta do <i>Apache</i> .....	64
3.9.4 Criação de certificados <i>SSL</i> para conexão segura com o servidor <i>Apache</i> .....	66
3.9.5 Configuração do <i>SSL</i> para páginas seguras .....	67
3.9.6 Inicialização do servidor e configuração do <i>firewall</i> .....	69
3.9.7 Configuração via <i>browser</i> para o <i>Dashboard</i> comunicar-se com o <i>Zabbix</i> .....	69
3.9.8 configuração dos agentes para o <i>auto-discovery</i> ...	73
3.10 Configuração do monitoramento com <i>Zabbix</i> .....	74
3.10.1 Configuração do <i>auto-discovery</i> do servidor <i>Zabbix</i> .....	74
3.10.2 Verificação da eficácia do <i>auto-discovery</i> na busca pelos ativos de rede .....	77
3.10.3 Verificação da atualização do sistema .....	77
<b>4. DISCUSSÃO DOS RESULTADOS</b> .....	82
<b>5. CONSIDERAÇÕES E PERSPECTIVAS</b> .....	86
<b>6.REFERÊNCIAS</b> .....	88
<b>7. ANEXOS</b> .....	96
<b>8. GLOSSÁRIO</b> .....	97

## 1. INTRODUÇÃO: SISTEMAS DE INFORMAÇÃO, VALOR E ESTRATÉGIA

Diversas mudanças técnicas, sociais e culturais foram provocadas pela globalização tecnológica, que ocorreu de forma acentuada a partir do período pós Guerra Fria. Hoje, a informação é o principal capital econômico, e a Tecnologia da Informação pode contribuir para o sucesso e competitividade das organizações, já que as decisões precisam ser tomadas com o máximo de informações válidas possíveis e de forma ágil.

Nesse contexto de agilidade, ubiquidade e valorização da informação, e tendências como *Big Data* e *Cloud Computing*, surgem grandes desafios, entre os quais pode-se citar:

### 1.1. SEGURANÇA DA INFORMAÇÃO

A segurança da Informação é uma preocupação cada vez mais crescente e por isso terá destaque neste trabalho.

Segundo a pesquisa global de segurança da informação do ano de 2016, os níveis executivos e os conselhos das empresas estão cada vez mais atentos aos riscos cibernéticos e seus impactos, e este fato ilustrou um aumento de 24% nos orçamentos destinados a esse tema no ano de 2015 (PWC, 2017).

Em 2017, o aumento detectado foi de 59% (EY, 2018). Apesar de os resultados serem favoráveis, ainda há necessidade de ações inovadoras e transformacionais para melhorar a segurança e privacidade da Informação em concomitância com a evolução das tecnologias de *hardware* e *software*.

Em 2017 um ataque de *Ransomware* denominado *WannaCry* afetou mais de 55.000 computadores em 77 países. Entre os afetados, destacam-se grandes Instituições, como a companhia global Telefônica, o Tribunal de Justiça do Estado de São Paulo e o Serviço de Saúde Nacional da França. Medidas simples como atualizações de sistema operacional e aplicações,

bem como a realização de *backups* poderiam evitar e reduzir impactos de ataques cibernéticos.

Cabe ressaltar que o conceito de segurança envolve questões diversas, como segurança física, infraestrutura tecnológica, conscientização organizacional, integridade de dados, gerenciamento de incidentes, entre outras, e por isso, as soluções para este problema devem abranger um conjunto de recursos que considerem o sistema de informação em seu aspecto mais amplo.

## 1.2 A UTILIZAÇÃO DE SERVIDORES VIRTUALIZADOS

Segundo pesquisa da *International Data Corporation* (IDC), desde o ano 2000 o número de máquinas virtuais vem superando, em vários países, o número de servidores físicos; na pesquisa mais recente sobre o uso de servidores virtualizados no Brasil, a consultoria IDC revelou que mais de 40% das empresas no Brasil utilizam servidores virtualizados. Já, Michael Warrilow, da Gartner, afirma que o mercado amadureceu rapidamente com muitas organizações com taxas de virtualização de servidores que ultrapassam 75%. (IT CHANNEL, 2016)

Apesar do surgimento de novas tecnologias, a virtualização de servidores permanece a plataforma de infraestrutura mais comum entre as empresas. A virtualização veio se consolidando e se expandindo no mercado principalmente devido ao avanço da computação em nuvem, segundo Luís Banhara, da Citrix, visto que os provedores de serviços virtuais necessitam de servidores virtuais para suprir a demanda de seus clientes. (VALOR ECONÔMICO, 2018)

A virtualização também é adotada por muitas empresas que buscam um alinhamento com a tendência da chamada *TI verde*, pois ela possibilita a diminuição da aquisição de *hardware* e uma redução do consumo de energia.



Diante de um cenário de grandes possibilidades e benefícios, é preciso fomentar uma gestão inteligente e segura de recursos. Com a possibilidade de um servidor não ser vinculado a um hardware específico, o gerenciamento se torna mais subjetivo. Além disso, as redes estão aumentando em termos de tamanho, complexidade e heterogeneidade. As organizações devem compreender essa realidade e buscar a adoção de medidas para mitigar os riscos iminentes. Para o analista do *Gartner* Neil MacDonald, “a segurança desses ambientes fica mais difícil. Há quebra das políticas de segurança ligadas à localização física. Por isso, precisamos de políticas de segurança independentemente do tipo de rede.” (COMPUTERWORLD, 2011)

A questão da Segurança ainda pode representar um risco para a criação e manutenção de ambientes virtualizados. Segundo Thompson (2014):

Com toda a visibilidade que a virtualização de servidores recebe, muitos podem pensar que todo mundo já a está utilizando em todos os lugares. No entanto, a realidade é que muitas empresas ainda estão no processo de adotar a virtualização pela primeira vez, para um novo grupo, departamento ou aplicativo. (THOMPSON, 2014)

Nesse sentido, a segurança pode representar um obstáculo para a virtualização, sobretudo pelo seu caráter de dinamicidade e pelo desafio de lidar com as camadas adicionais de complexidade que ela agrega à infraestrutura. Desse modo,

uma ameaça no ambiente virtual pode se espalhar para toda a rede? Preocupações como esta podem estar segurando inúmeras empresas a aderirem em maior proporção a tecnologia que será fator de sobrevivência em um futuro bem próximo, ou até atual. (LEANDRO, 2017)

É preciso, portanto, gerenciar os riscos e adotar medidas capazes de atenuá-los para que se possa utilizar o potencial da virtualização em prol das organizações.

### 1.3 SOLUÇÕES LIVRES

O software livre vem sendo adotado por um número cada vez maior de empresas, apresentando-se como uma alternativa à aquisição de licenças de softwares proprietários e sendo recomendado também em iniciativas do Governo Federal por ter a característica da robustez e por conferir maior autonomia e segurança de dados, "fator de incontestável relevância, até em razão de segurança nacional". (SENADO FEDERAL, 2012).

É um desafio, portanto, buscarmos soluções completas e seguras que estejam ancoradas nos princípios de *Software* Livre definidos pela *Free Software Foundation*<sup>1</sup>, através dos quais viabiliza-se a liberdade para:

- Executar um programa para qualquer propósito (liberdade 0);
- Estudar e adaptar um programa de acordo com as suas necessidades (liberdade 1);
- Redistribuir cópias de modo a compartilhar soluções (liberdade 2) e
- Liberar o código fonte para que toda a comunidade se beneficie das mudanças implementadas (liberdade 3).

---

1 <https://www.gnu.org/philosophy/free-sw.pt-br.html>

## 1.4 MOTIVAÇÃO

Uma situação gerencial real experimentada em uma Empresa no Estado do Rio de Janeiro durante a realização de estágio obrigatório para a disciplina AA393 coincidiu com o panorama exposto.

A empresa em questão possuía uma pequena parte de seus servidores virtualizados; porém, a performance da virtualização não correspondia ao esperado, o que pode ter acontecido por fatores como escolha incorreta dos *softwares*, má configuração, incompatibilidade de sistemas, falta de embasamento e conhecimento técnico para o tratamento da segurança, entre outros.

A empresa, portanto, tinha necessidade de renovar as tecnologias em seu ambiente organizacional, de forma a otimizar a utilização de recursos, não os deixando ociosos. Para atingir este objetivo, foi realizado um amplo estudo buscando-se uma implementação segura e bem-sucedida da virtualização, o que motivou a realização deste trabalho.

Por questões envolvendo ética, sigilo e privacidade, o nome da Empresa não será citado.

## 1.5 DEFINIÇÃO DO PROBLEMA

Como implantar boas práticas de Segurança na criação de ambientes virtualizados utilizando Software Livre?

## 1.6 HIPÓTESE

Este trabalho privilegiou a hipótese de que a combinação de tecnologias de monitoramento e virtualização através de *softwares* livres, ancorada nas premissas estabelecidas pela norma NBR ISO 27002, é capaz de otimizar um Sistema de Informação, favorecendo a criação de ambientes seguros.

Para a idealização da hipótese, consideramos o conceito mais amplo de Sistema de Informação, que o descreve como um conjunto de “recursos humanos, materiais, tecnológicos e financeiros agrupados segundo uma sequência lógica que permita o processamento de dados, e sua correspondente tradução em informação” (GIL, 1999)

## **1.7 OBJETIVO**

Este trabalho tem como objetivo, portanto, apresentar uma estratégia possível para implantação de segurança e otimização de Sistemas de Informação a partir da aplicação da norma NBR ISO 27002 em ambientes virtualizados utilizando Software Livre.

A norma NBR ISO 27002 estabelece diretrizes não só em termos de política e organização, mas também foca em ativos, controle de acesso, segurança física e de operações, aquisições, incidentes e até mesmo recursos humanos, constituindo diferentes seções que abordam os Sistemas de Informação em seu sentido mais amplo.

Acredita-se, portanto, que a utilização desta norma configure um embasamento legal para a implantação de medidas de segurança.

## **1.8 JUSTIFICATIVA**

Dado que nem todas as empresas possuem ambientes virtualizados e constatada a necessidade de um fundamento para a implantação e manutenção de ambientes virtualizados, o trabalho aqui apresentado justifica-se pela atualidade, abrangência e importância das temáticas envolvidas para as organizações. Por isso, tivemos a preocupação de generalização do estudo, onde a combinação sugerida está baseada em ferramentas e procedimentos de segurança que podem ser aplicados em qualquer organização, independente de porte, estrutura, segmento e outras

características administrativas, buscando a disponibilização dos procedimentos implantados como um instrumento de referência.

## 1.9 TRABALHOS RELACIONADOS

Como trabalho relacionado, cita-se a pesquisa de MELO, Sandro; Domingos, Cesar. CORREIA, Lucas, MARUYAMA, Tiago (2006), que consiste em um guia para decisões do uso de *software* livre em projetos de segurança destacando o valor computacional que ele tem a oferecer, mostrando um caminho a ser seguido da tática à prática em servidores.

A pesquisa aqui apresentada, contudo, tem como diferencial a combinação de tecnologias de virtualização e monitoramento, utilizando o melhor de cada uma para otimizar e assegurar o Sistema de Informação.

Também cita-se a pesquisa de TITON (2013), que documentou e analisou técnicas de *hardening* em servidores que utilizam Sistemas Operacionais Windows e Linux, apresentando um manual de boas práticas para a segurança de servidores no dia-a-dia. Titon, contudo, foca no comparativo entre os Sistemas Operacionais, e embora aplique regras de segurança, não aborda a temática do monitoramento, que é uma premissa da norma NBR ISO 27002. Por isso, esta pesquisa buscou a combinação das duas tecnologias, a fim de atender mais itens da norma.

## 1.10 PROCEDIMENTOS METODOLÓGICOS E ORGANIZAÇÃO DO TRABALHO

A etapa inicial desta pesquisa tem um caráter essencialmente bibliográfico, em que se pretende, através dos pressupostos teóricos selecionados, apresentar a conceituação básica dos processos de virtualização e monitoramento, bem como apresentar a norma que fundamenta esta pesquisa.

Em seguida, apresenta-se as ferramentas selecionadas para a etapa prática. Nesta etapa, abordamos as ferramentas que ofereciam soluções mais completas relacionadas aos processos de monitoramento e virtualização.

A etapa seguinte contém a apresentação de procedimentos para implantação da norma NBR ISO 27002 na prática.

Para finalizar a pesquisa, apresentaremos uma breve discussão dos resultados, bem como considerações, limitações e perspectivas futuras.

## 2. CONCEITOS BÁSICOS

Para uma melhor compreensão da proposta descrita neste trabalho, apresenta-se a seguir a Norma NBR ISO 27002, bem como os conceitos de virtualização e monitoramento.

### 2.1 A NORMA NBR ISO 27002

A norma 27002 surgiu a partir da ABNT ISO/IEC 17799 elaborada no Comitê Brasileiro de Computadores e Processamento de Dados (ABNT/CB-21) pela Comissão de Estudo de Segurança Física em instalações de Informática (CE-21:204.01), sendo, portanto, equivalente à ISO/IEC 17799:2005. Deste modo, integra uma família de normas de sistemas de gestão de segurança da informação (SGSI), conforme apresentado abaixo:

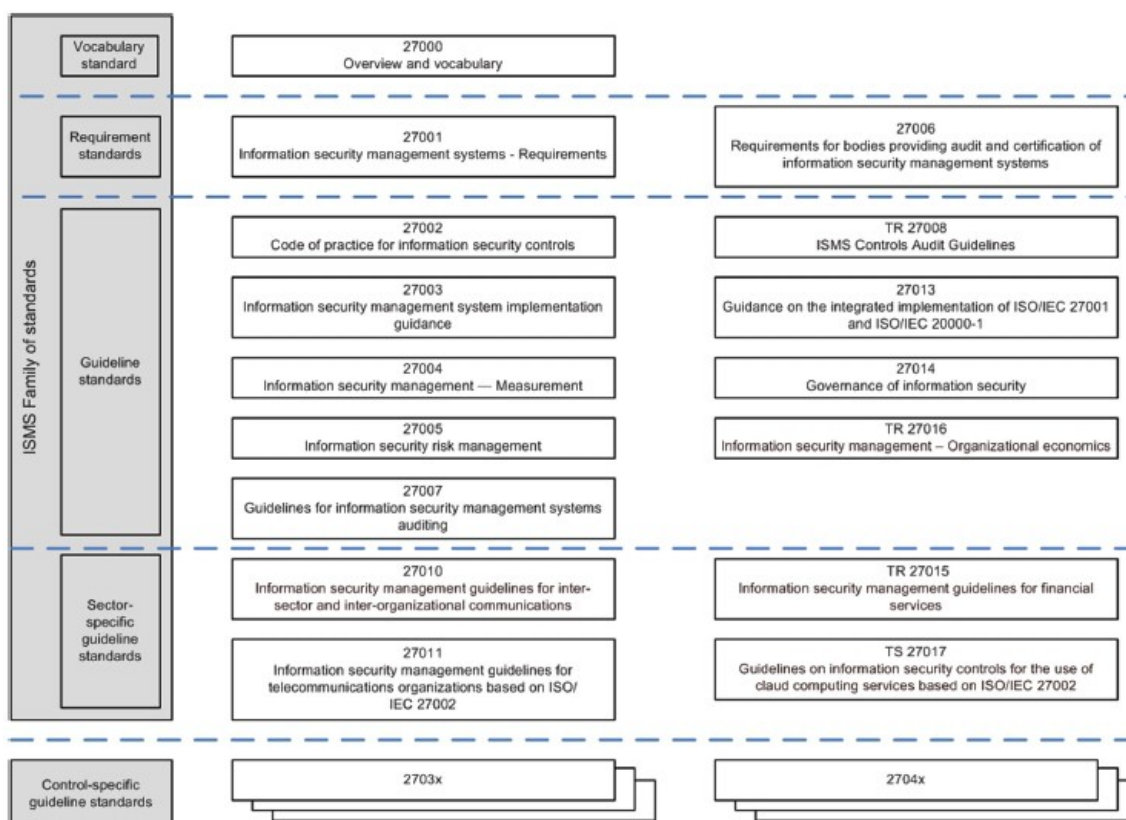


Figura 1. Normas de Gestão de Segurança da Informação<sup>2</sup>

<sup>2</sup> <https://s.profissionaisti.com.br/wp-content/uploads/2015/07/ISO-27000-relacionamentos1.jpg>

Vemos a importância da Segurança da Informação, que atualmente conta com 16 publicações voltadas para a implementação e operação de sua gestão, todas alinhadas com padrões internacionais e adequados a organizações de todos os portes e segmentos.

A norma NBR ISO 27002, portanto, tem como objetivo estabelecer um ponto de partida para o desenvolvimento de diretrizes específicas de segurança para as organizações, o que deve ser feito de acordo com o contexto e as particularidades de cada uma. Para tanto, é apresentado o conceito de segurança da Informação que rege o documento:

Segurança da Informação é a proteção da informação de vários tipos de ameaças para garantir a continuidade do negócio, minimizar o risco ao negócio, maximizar o retorno sobre os investimentos e as oportunidades de negócio. (NBR ISO 27002, 2005, p.10)

Sendo assim, a Segurança da Informação é considerada vital em nosso atual contexto sócio-histórico-cultural complexo e em constante evolução, pois é importante para os negócios tanto do setor público como do setor privado. Segundo pesquisas recentes, a segurança da informação é uma prioridade para todos os níveis hierárquicos e há a necessidade de se definir, alcançar, manter e melhorar a segurança para assegurar a competitividade e a inovação.

Cabe ressaltar que a NBR ISO 27002 atenta diversas vezes para o fato de que a segurança envolve ameaças diversas, como fraudes eletrônicas, espionagem, sabotagem, vandalismo, incêndio, inundação, entre outros. Por isso é necessária uma análise dos riscos, o que é imprescindível para direcionar e determinar as ações gerenciais apropriadas para combater cada um dos riscos. A análise também auxilia a definir a implementação dos





Dado que nenhum conjunto de controles pode conseguir a segurança completa, esta pesquisa limita-se a apresentar uma solução possível criada a partir de uma das normas de segurança mais relevantes, com o objetivo de mostrar a possibilidade de implementação de controles de segurança com *software* livre, solução esta que pode potencializar um Sistema de Informação através da aplicação de técnicas para chegar a um nível de segurança e disponibilidade recomendável para servidores.

As configurações realizadas no âmbito desta pesquisa focaram em mecanismos voltados para o controle de acesso, para a proteção contra elementos maliciosos e para a gestão dos ativos.

## 2.2. VIRTUALIZAÇÃO

Segundo Carissimi (2008), a virtualização apesar de não possuir uma definição consensual, pode ser considerada uma técnica que permite particionar um único sistema computacional em vários outros denominados de máquinas virtuais. Cada máquina virtual oferece um ambiente completo muito similar a uma máquina física. Este ambiente completo pode ser composto por sistema operacional, aplicativos, serviços de rede interconectados virtualmente através de ativos virtualizados ou não. Isto significa que a virtualização possibilita a integração de instâncias heterogêneas e oferece uma gama de aplicações, até mesmo em relação a implementação de *desktops* remotos, *clusters* e discos virtuais.

Como benefícios da virtualização, podemos citar:

- A possibilidade de hospedar muitos serviços numa única máquina física, com administração individual de cada uma, o que permite melhor gerenciamento com menor custo de *hardware*.
- A virtualização permite um balanço dinâmico de carga imperceptível pelos usuários, melhorando a disponibilidade.

- A virtualização pode se recuperar rapidamente de problemas de *hardware*, já que uma máquina virtual pode ser facilmente transferida para outro *hardware*.
- A virtualização permite otimizar o processamento de *hardware*, de forma a não deixar recursos ociosos.
- A virtualização diminui o tempo para implantação de novos sistemas, além de permitir a construção de ambientes de testes.
- A virtualização permite criar e retornar *snapshots*, que são registros de estado e dados de uma máquina virtual em determinado momento. Tais registros servem como um ponto de restauração em casos de erro.

Como desvantagens, podemos citar:

- Há a necessidade de instalação de diversos pacotes para um sistema operacional entender que ele está sendo virtualizado;
- Há a necessidade de mediação de um *hypervisor* nas chamadas ao sistema;
- Sistemas operacionais antigos e com código fechado não possuem extensão nativa para virtualização;
- Pode haver dificuldade no acesso direto a placas muito específicas ou alguns dispositivos de entrada e saída;

Apesar das desvantagens, a virtualização continua sendo uma opção viável para compor uma estrutura de Tecnologia da Informação. Segundo *Michael Warillow*, diretor de pesquisas da *Gartner*, “O mercado amadureceu rapidamente ao longo dos últimos anos, com grandes organizações alcançando taxas de virtualização de servidores que ultrapassam 75% – o que ilustra o alto nível de penetração” (GUERRA, 2016)

Singh (2008) apresenta a virtualização como um *framework* ou metodologia para dividir os recursos de um computador em múltiplos ambientes de execução, aplicando um, ou mais conceitos, e tecnologias

como particionamento de software ou hardware, tempo compartilhado, simulação completa ou parcial da máquina, emulação e qualidade de serviços.

A respeito da simulação completa ou parcial, estas são exemplos de arquiteturas para virtualização. A figura 3 representa de forma sintética as diferenças entre as duas arquiteturas:

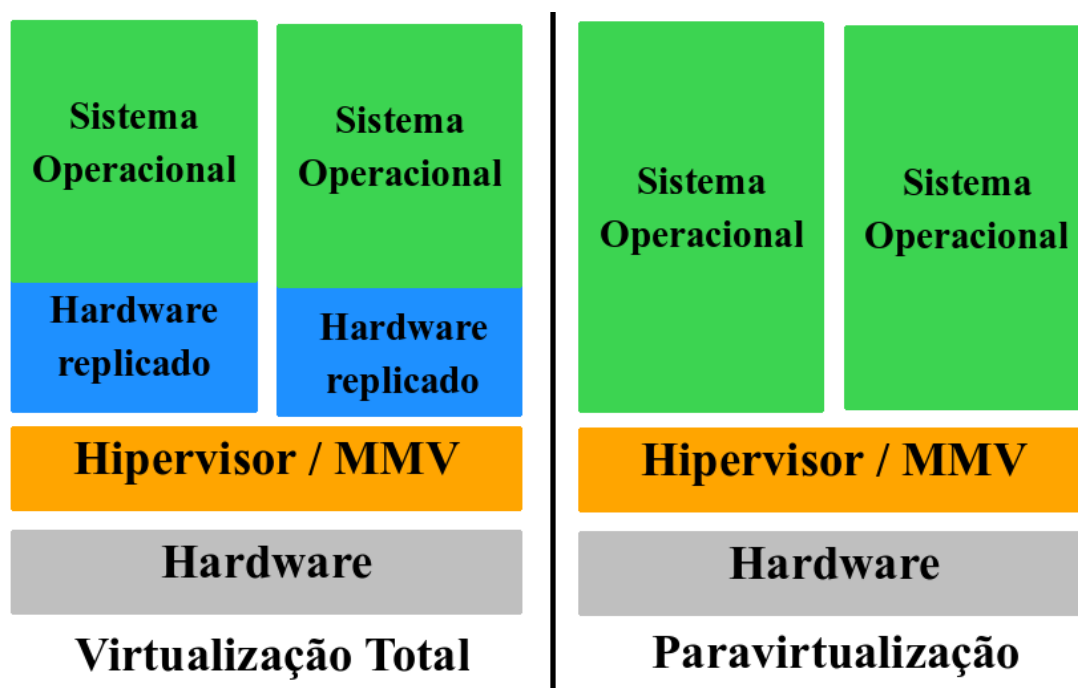


Figura 3. Arquiteturas de virtualização: total e paravirtualização<sup>3</sup>

No âmbito desta pesquisa, contemplamos o conceito de paravirtualização, pois esta é a arquitetura que possui mais benefícios para o escopo deste trabalho, tais como gerenciamento mais apurado, melhor controle de máquinas, melhor desempenho do *hardware*, entre outros.

A paravirtualização utiliza um *hypervisor*, que é um monitor de máquina virtual. Ele tem como objetivo prover e gerenciar recursos para a máquina virtual. Na paravirtualização, o *hypervisor* entrega uma simulação de *hardware* para o sistema operacional hospedeiro, o que permite que a

<sup>3</sup> Disponível em <<https://www.portalgsti.com.br/2016/11/virtualizacao-completa-e-paravirtualizacao.html>> Acesso em 18 nov 2018

virtualização se comunique com o *hardware*. Assim, os dispositivos de *hardware* são acessados por *drivers* da própria máquina virtual. Segundo Mathews et. al (2009), esta arquitetura de virtualização é leve e rápida, capaz de melhorar a entrada e saída e a alocação de recursos.

As ferramentas *Microsoft Windows Server 2008 R2 Hyper V*, *Xen Server*, *VM Ware* e *Red Hat Enterprise Virtualization* são as mais populares em levantamento realizado em documentos de repositórios acadêmicos e pesquisas de mercado sobre a virtualização. A seguir, apresenta-se uma síntese da análise comparativa:

<b>Ferramenta</b>	<b>Sistemas Operacionais Suportados</b>	<b>Sistema Operacional Hospedeiro</b>	<b>Licença</b>	<b>Tipo de Virtualização</b>	<b>Arquitetura das máquinas virtuais</b>
Hyper V	Windows, Linux, FreeBSD	Microsoft Windows Server	Comercial	Para-virtualização	X86, x86-64, AMD64
Xen Server	Windows, Linux, FreeBSD, Outros	Nenhum	GPL / Comercial	Para-virtualização	X86, x86-64, AMD64
VM Ware	Windows, Linux, FreeBSD, Outros	Nenhum	Comercial	Para-virtualização	X86, x86-64, AMD64
Red Hat Enterprise Virtualization	Windows, Linux, FreeBSD, Outros	Red Hat, Enterprise Linux	Comercial	Para-virtualização	X86, x86-64, AMD64

Tabela 1. Comparação entre ferramentas de virtualização

Após a análise, foi selecionada a plataforma *XenServer*, software livre, e campeã no *Server Virtualization Vendor Landscape* do *Info-Tech Research Group*. (Info-Tech Research Group Inc, 2016) Esta ferramenta teve uma performance significativamente melhor, pois constitui uma solução de

infraestrutura de rede virtual, *multi-tenant*, completa com um *hypervisor* de 64 bits, com um console intuitivo para gerenciamento de uma quantidade ilimitada de servidores e máquinas virtuais.

A figura a seguir retrata o funcionamento do *hypervisor XenServer* e mostra como o sistema hospedeiro emula o novo *hardware* fazendo a alocação dos recursos da máquina para as máquinas virtuais:

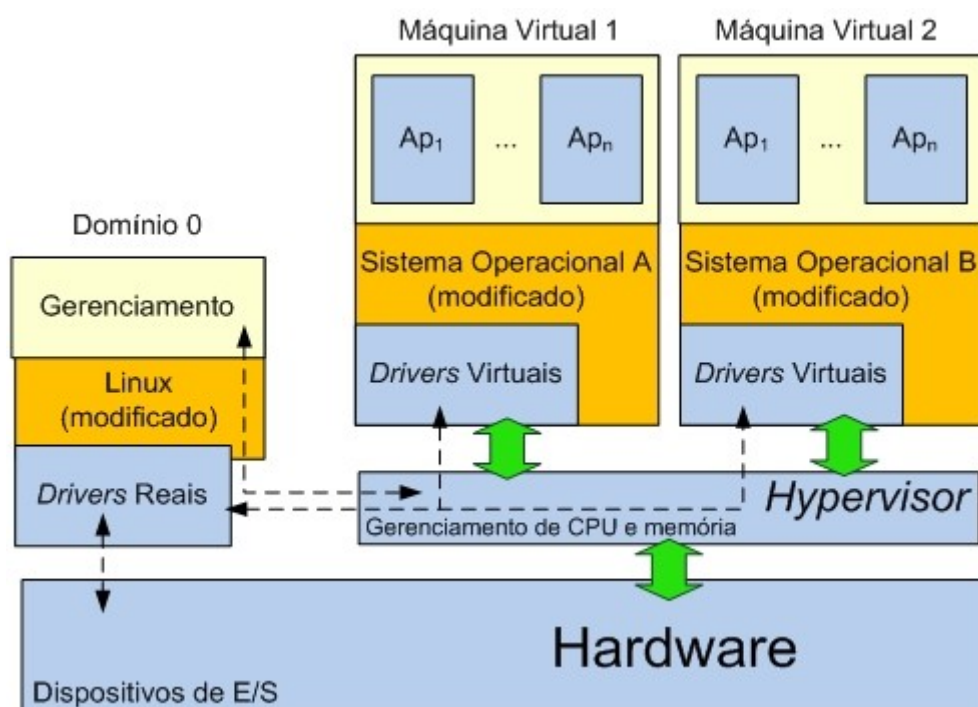


Figura 4. Funcionamento do Hypervisor<sup>4</sup>

A ferramenta apresenta-se como confiável, disponível, segura, de fácil configuração e de alta performance. Funcionalidades como balanceamento dinâmico da carga de trabalho, gerenciamento de energia, otimização de memória, serviços de provisionamento, recuperação e proteção de máquinas virtuais, entre outras, fazem com que seja uma ótima opção para automatização e gerenciamento avançado de recursos, já que reduz drasticamente custos com *datacenters* e oferece capacidades de gerenciamento avançadas.

4 Entenda o Xen. Disponível em <<http://deinfo.uepg.br/~alunoso/2017/XEN/conceitos-basicos.html>> Acesso em 18 nov 2018

## 2.3 MONITORAMENTO

De acordo com a norma ISO 27002, convém que uma estrutura de gerenciamento seja estabelecida para iniciar e controlar a implementação da segurança da informação dentro da organização (NBR ISO 27002, 2005, p. 10).

Sendo o monitoramento o processo de acompanhamento capaz de fornecer indicadores de operações com o objetivo de gerar relatórios de incidentes diversos, ele fornece uma visão geral de toda a infraestrutura tecnológica e apresenta resultados em painéis e gráficos, permitindo uma visão operacional, tática e estratégica, que favorecem análise de dados e a administração dos recursos disponíveis. Para tanto, faz-se necessária a utilização de um NMS (*NetWare Management System*), um sistema de gerenciamento de rede responsável por aplicações capazes de monitorar e controlar dispositivos em rede.

O NMS deve ser utilizado para monitorar dispositivos de forma automatizada, notificar a ocorrência de falhas, permitir o planejamento de investimentos em novos recursos, além de permitir avaliação da qualidade dos serviços, o que facilita a descoberta e resolução de falhas provocando o mínimo de impacto aos usuários, contribui para a visualização ordenada de dados estatísticos para apoiar os processos de tomada de decisão e auxilia o provisionamento dos recursos físicos e virtuais, entre outros benefícios.

Com o monitoramento é possível ver como os dados coletados de maneira correta podem gerar informação, e mais ainda, como dados sobre a tecnologia geram conhecimento para desenvolver a própria tecnologia.

Segundo FRAZÃO e BRAGA (2015),

Nenhum sistema computacional está livre de ataque de crackers. O melhor que pode ser feito é dificultar ao máximo suas ações e fazer uma monitoração permanente para procurar detectar

rapidamente as tentativas de invasão. (FRAZÃO e BRAGA, 2015, p. 142)

Sendo assim, pode-se dizer que as principais vantagens do monitoramento são a rápida detecção de problemas e invasões, o que contribui para a redução do tempo de inatividade e para garantir a disponibilidade dos recursos, além de prestar dados acerca da saúde dos ativos, bem como justificar necessidades de provisionamento.

As ferramentas *Cacti*, *HP Network*, *IBM Tivoli*, *Nagios* e *Zabbix* demonstraram-se as mais populares em levantamento realizado em documentos de repositórios acadêmicos e pesquisas de mercado. A seguir, apresenta-se uma síntese da análise realizada:

Ferramenta	Agente	Plugin	SNMP	Aplicação Web	Mapas	Alertas	Base de Dados	Licença
Cacti	Não	Sim	Sim	Controle total	Plugin	Sim	RRDtool, MySQL	GPL
HP Network	Não	Sim	Sim	Controle total	Sim	Sim	PostgreSQL Oracle Database	Comercial
IBM Tivoli	Não	Sim	Sim	Controle parcial	Sim	Sim	MySQL Oracle Database DB2	Comercial
Nagios	Suportado (NRPE)	Sim	Plugin	Controle parcial	Sim	Sim	Flat file, SQL	GPL
Zabbix	Sim	Sim	Sim	Controle total	Sim	Sim	Oracle, MySQL, PostgreSQL IBM DB2, SQLite	GPL

Tabela 2. Comparação entre ferramentas de monitoramento

A partir desta análise comparativa, foi possível a escolha do *Zabbix* como melhor opção para tratar da questão do monitoramento. O *Zabbix* se mostrou preparado para controle de grandes e pequenos ambientes distribuídos, pois além de possuir uma interface de fácil utilização é extremamente funcional e flexível. Armazena dados históricos, tendências e



configurações em bancos de dados e pode ser facilmente integrado aos bancos *MySQL*, *Oracle*, *PostgreSQL* e *SQLite*. Nesta ferramenta, a lógica está no servidor e os agentes funcionam apenas como coletores de dados. Está disponível para plataformas de 32 e 64 bits. Possui itens diversificados, como *templates*, *triggers*, cenários, telas, gráficos em tempo real, e possibilidade de *scripts* externos e integração com outras ferramentas através de plugins. Além disso, é um *software* livre e que possui suporte a diversos sistemas operacionais, tais como *Linux*, *Windows*, *Mac OS*, entre outros.

### 3. PREPARAÇÃO E CONFIGURAÇÕES REALIZADAS

Nesta seção, veremos os passos que podem ser executados para a instalação e configuração do ambiente de aplicação da proposta apresentada.

#### 3.1 INSTALAÇÃO DO XENSERVER

A ISO de instalação do *XenServer* está disponível no endereço <https://xenserver.org>. Após o *download*, é necessário gravar a imagem em CD. A figura 5 representa os passos do início da instalação até a escolha da configuração do teclado:



Figura 5: Etapas iniciais da instalação do XenServer

Feito isso, os passos seguintes, presentes na figura 6, dizem respeito à configuração da senha de acesso ao servidor, passando pela configuração de rede, até a configuração do fuso horário. Para as configurações de rede é necessário informar o endereço IP, a máscara de sub-rede e o *gateway*.

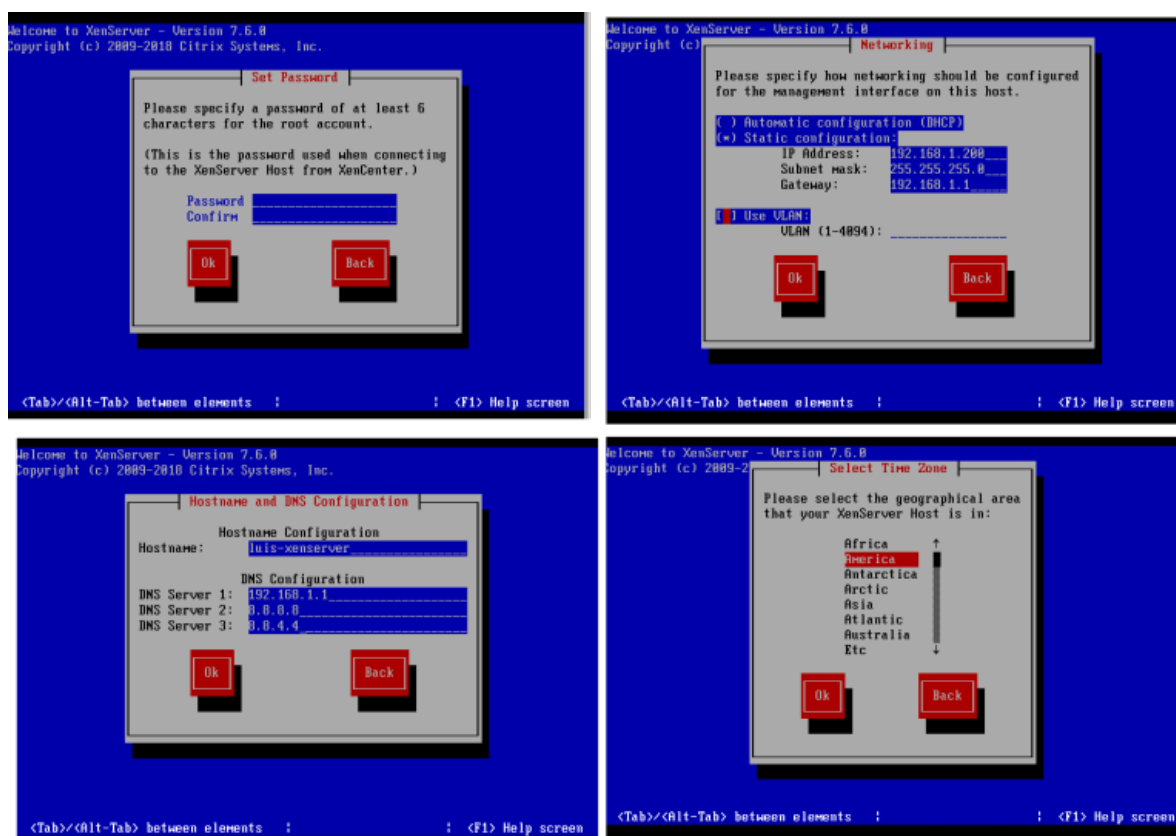


Figura 6. Configurações iniciais para instalação do *XenServer*

Nos passos seguintes, apresentados na figura 7, deve-se confirmar a instalação do sistema operacional, bem como realizar a instalação de pacotes extras opcionais.

Após a execução dessas ações, é recomendado realizar um *check-up* para verificar se a instalação foi bem-sucedida. A opção de *check-up* é fornecida pelo sistema após o término da instalação.

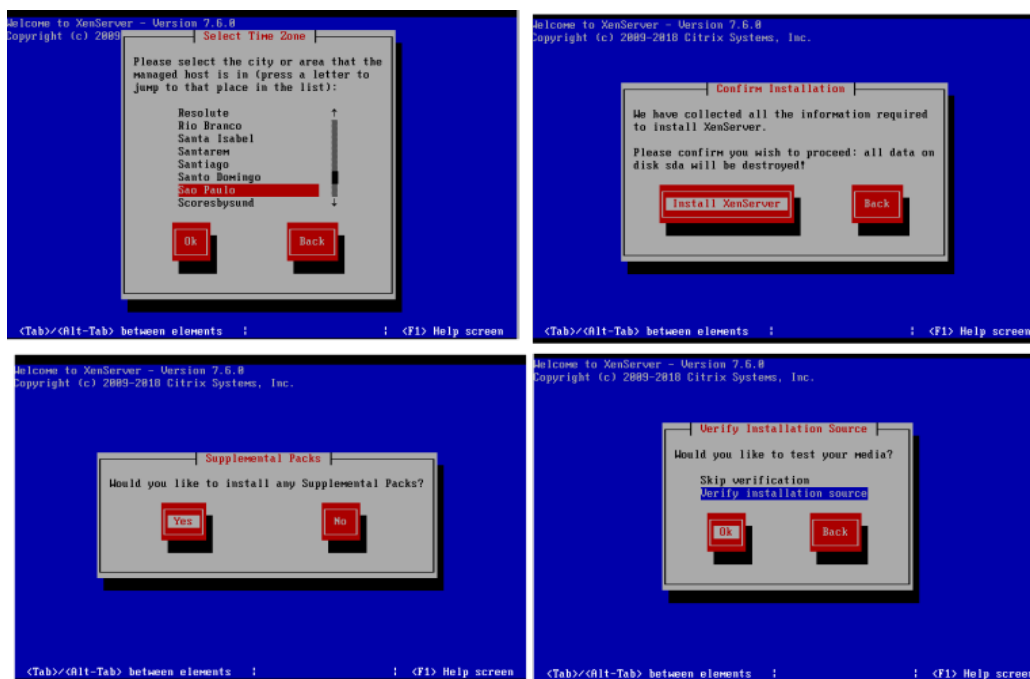


Figura 7. Finalização das configurações básicas e confirmação da instalação do *XenServer*

Feita a verificação da instalação e a instalação dos pacotes extras, deve-se finalizar a instalação e remover a mídia de instalação para inicializar o sistema operacional.

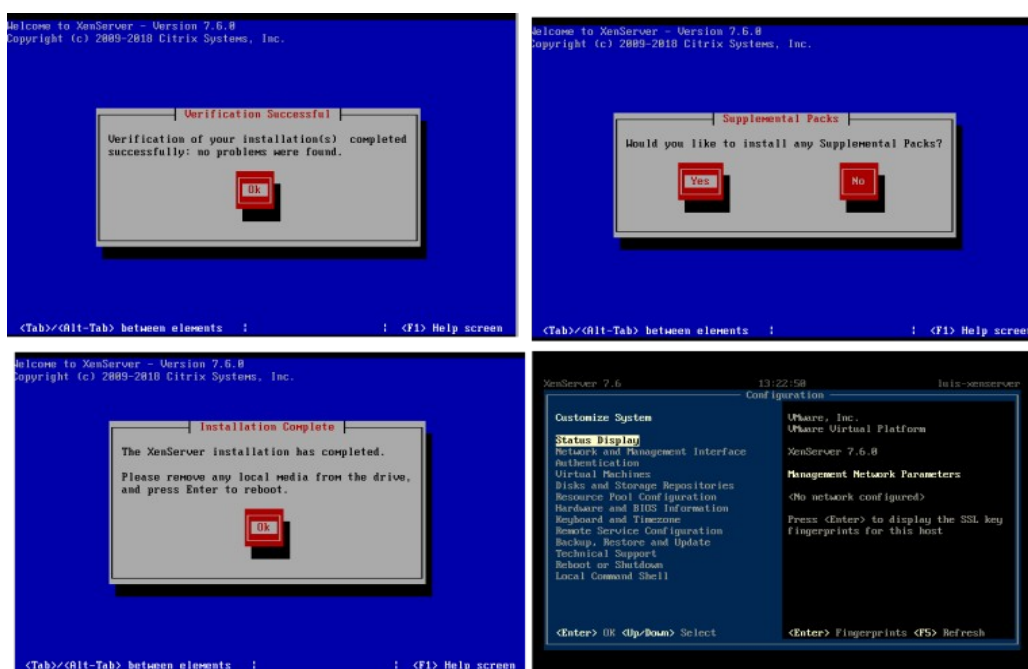


Figura 8. Finalização da Instalação do *XenServer*

## 5.2 INSTALAÇÃO DO XENCENTER

Para a configuração das máquinas virtuais de forma mais fácil através de interface gráfica, recomenda-se o programa *XenCenter*, também disponibilizado em <https://xenserver.org> na área de *downloads*. Esta ferramenta possibilita conexão com servidores do *XenServer*, o que permite instalar e gerenciar máquinas virtuais, configurando opções de hardware e segurança, entre outras.

Concluída a instalação, pode-se, entre as funcionalidades da ferramenta, adicionar diversos servidores ao console de gerenciamento, criar um *pool* de máquinas virtualizadas e realizar movimentação das máquinas entre os servidores.

É necessário, primeiramente, adicionar o servidor de virtualização, e para isto, basta um clique no botão *add new server*. Em seguida, basta adicionar o IP do servidor e a descrição conforme a figura 9:

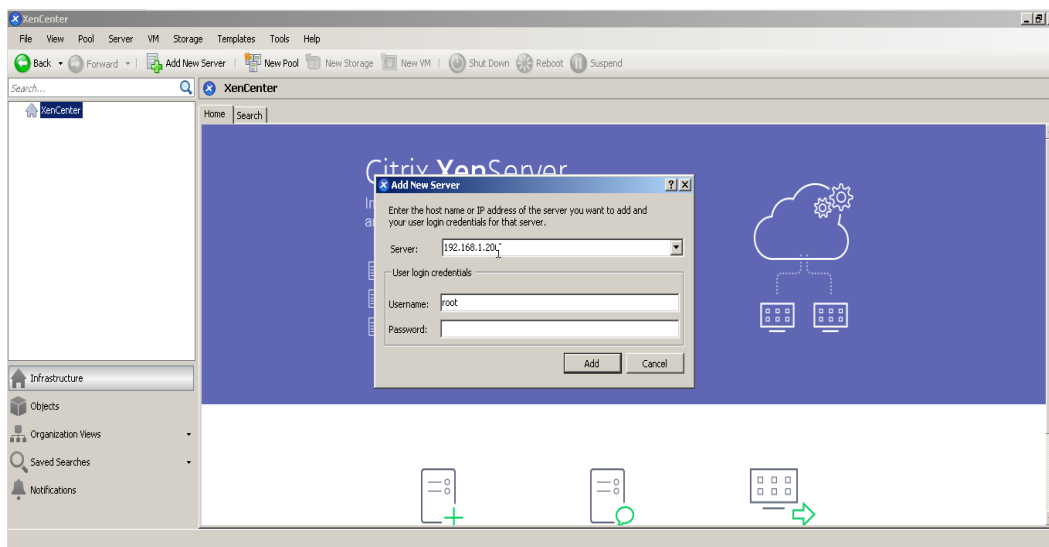


Figura 9. Adicionando um servidor de virtualização no XenCenter

### 5.3 CRIAÇÃO DO *TEMPLATE* DA MÁQUINA VIRTUAL

Para a criação da máquina virtual, é necessário fazer o *download* da imagem ISO *Netinstall* disponível em [www.centos.org](http://www.centos.org). Ao término do *download*, é necessário copiar a imagem ISO para o servidor *XenServer*.

Para copiar a imagem, é necessária a criação de um diretório que contemple as ISOs que serão utilizadas, o que pode ser feito através do seguinte comando:

```
mkdir -p /var/opt/xen/local-iso
```

Para que o *XenServer* seja capaz de identificar as ISOs no diretório mencionado, é necessário executar o seguinte comando:

```
xe sr-create name-label=LocalISO type=iso  
device-config:location=/var/opt/xen/local-iso device-config:legacy_mode=true  
content type=iso
```

Com isto, cria-se a máquina virtual com a ISO indicada na Figura 10:

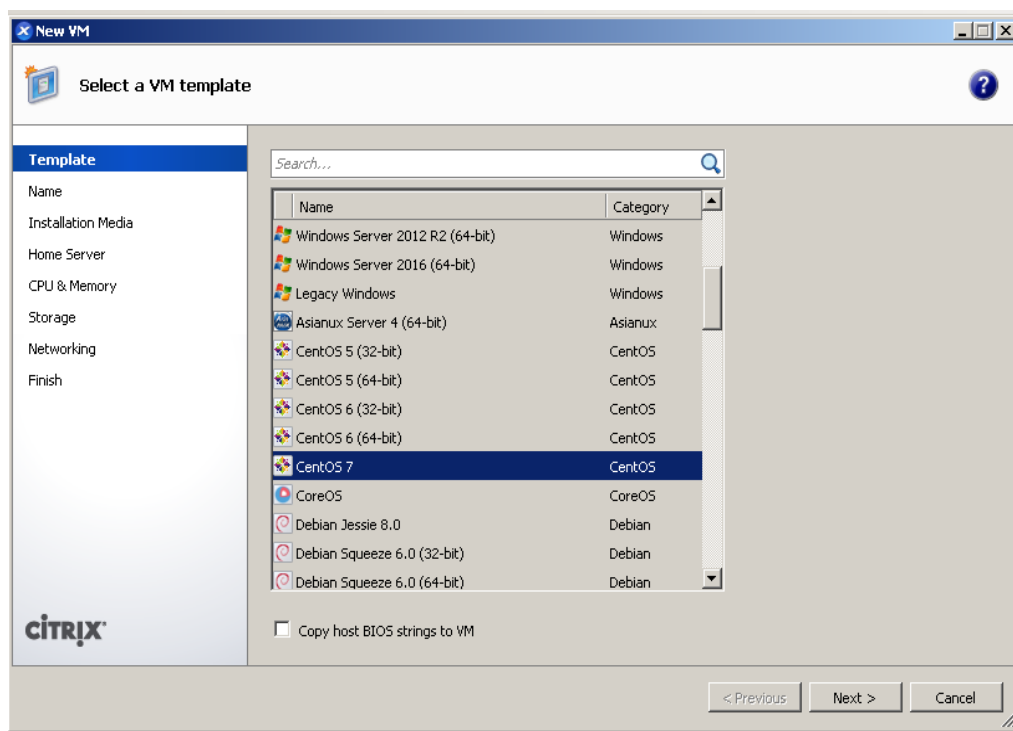


Figura 10. Seleção de *template CentOS*

Após a escolha do *template*, deve-se selecionar a ISO que foi adicionada ao servidor anteriormente.

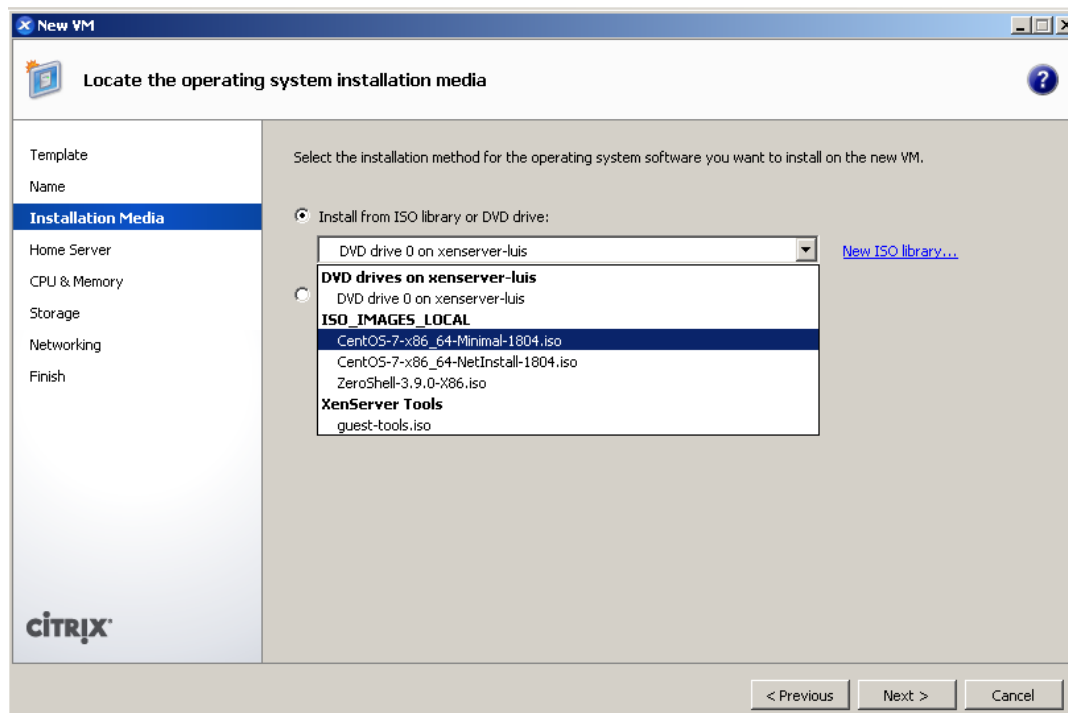


Figura 11. Selecionando a Instalação a partir da ISO

Em seguida, é necessário configurar o repositório para baixar os pacotes da instalação do servidor. De acordo com os critérios: confiabilidade, disponibilidade e velocidade no *download*, recomenda-se o repositório [mirror.globo.com](http://mirror.globo.com). Trata-se de um repositório brasileiro oficial de pacotes e ISOs de projetos *free* e *opensource*.

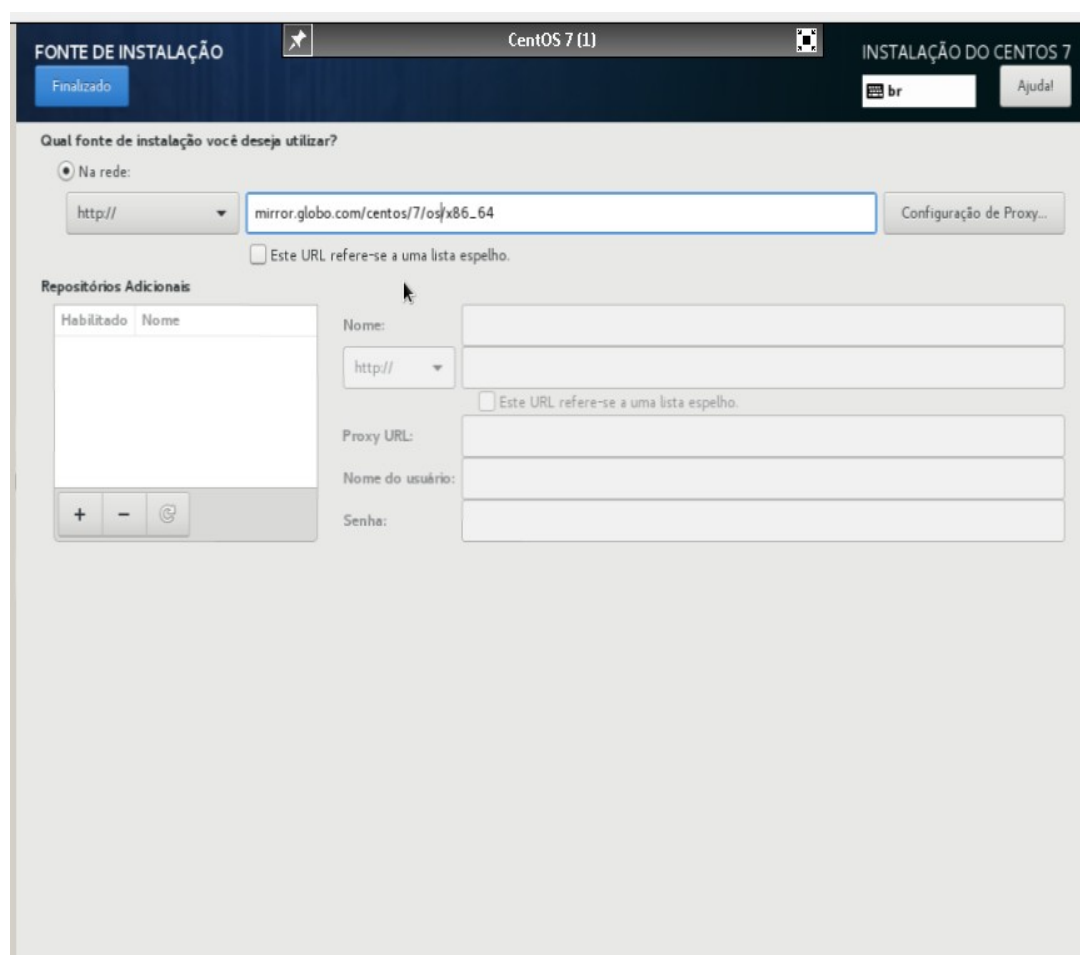


Figura 12. Seleção do repositório

Após configurar o repositório para fazer o *download* dos pacotes de instalação da máquina virtual, deve-se configurar os discos para um esquema especial de montagem em que possamos aplicar controles descritos na norma NBR ISO 27002. O esquema de montagem pode ser realizado conforme a Figura 13:



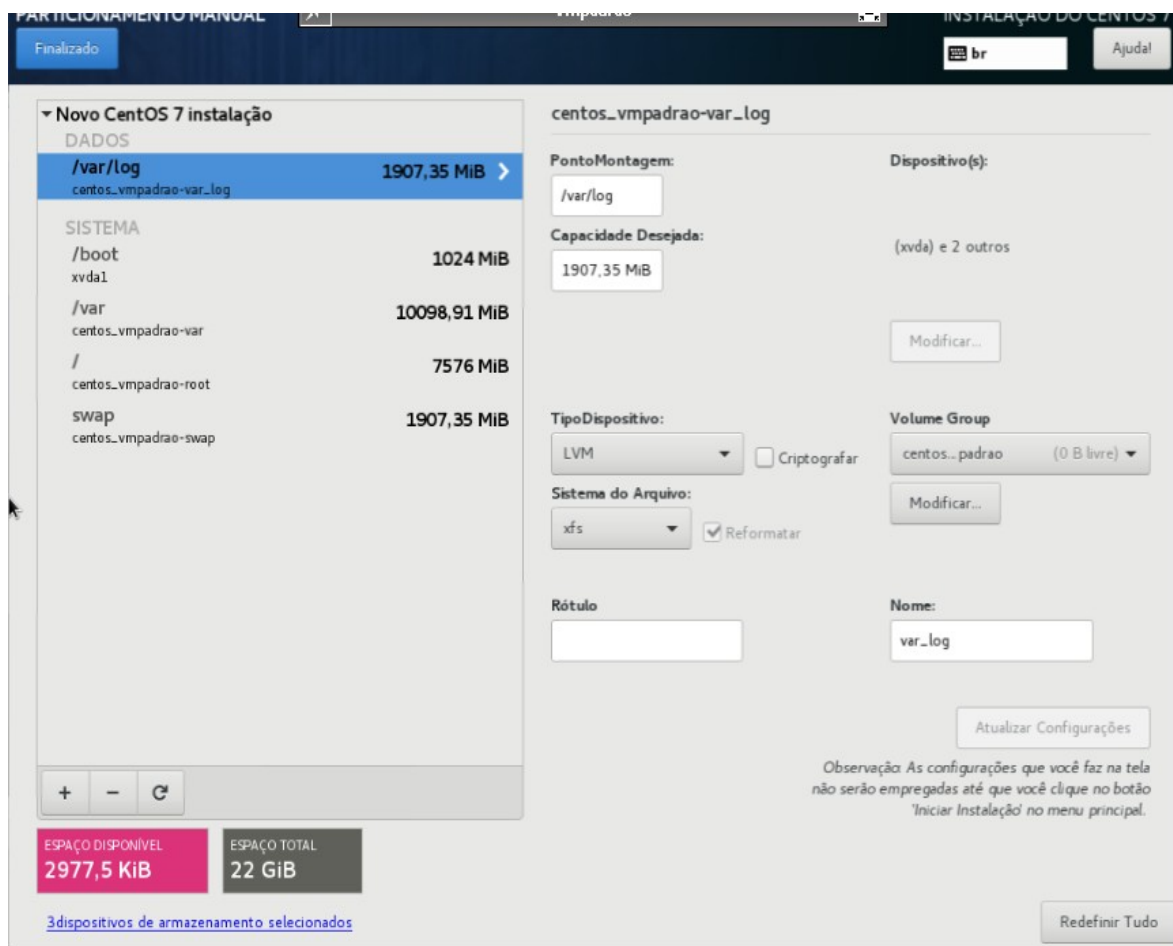


Figura 13. Configurações para os discos para um esquema especial de montagem

Executada esta etapa, a máquina virtual base já está pronta para a criação do *template*.

### 3.4 CONFIGURAÇÃO DO *TEMPLATE* E ADEQUAÇÃO SEGUNDO A NORMA NBR ISO 27002

Dado que a norma NBR ISO 27002 apresenta diretrizes gerais para a Segurança da Informação, e “nem todos os controles e diretrizes contidos nesta norma podem ser aplicados.” (NBR ISO 27002, p. 19) considerou-se, para esta etapa as recomendações passíveis de aplicação e teste para realizar referência cruzada com algumas seções da norma. Segue abaixo a demonstração da configuração, bem como a aplicação das recomendações na configuração do modelo da máquina virtual criada no item 5.3.

#### 3.4.1 Configuração inicial da rede

Para configuração da máquina virtual, é necessário a configuração da rede na qual a máquina está alocada, podendo ser feita da seguinte forma:

```
vi /etc/sysconfig/network  ## Comando para modificação do arquivo de rede
NETWORKING=yes            ## Habilitando a configuração de rede
HOSTNAME=vm-padrao.luistestes  ## Configurando o nome a máquina
GATEWAY=192.168.1.1        ## Host que interligará a rede interna com
a internet
hostnamectl set-hostname vm-padrao.luistestes ## Comando para setar o
nome padrão da máquina
```

#### 3.4.2 Instalações extras

Para a máquina funcionar de maneira satisfatória, e para possibilitar futuras intervenções, é necessária a adição de pacotes adicionais para o sistema *Enterprise Linux* através do pacote *epel*. Pacotes de compilação também podem ser necessários em muitos casos, pois é um pré-requisito para algumas aplicações, como o *XenServer*, que é compilado. Seguem os comandos da instalação:

```
rpm -Uvh http://mirror.globo.com/epel/epel-release-latest-7.noarch.rpm ##  
instalação do repositório para termos acesso a pacotes extras  
yum -y install gcc gcc-c++ make wget ntsysv setuptool openssh-clients rsync  
systemconfig-firewall net-tools ## instalando pacotes utilitários que são  
utilizados com frequência  
yum install bash-completion bash-completion-extras -y ## instalando pacotes  
que auxiliam a utilização do bash, que é a interface de utilização do sistema.
```

### 3.4.3 Instalação do *zabbix-agent*

Como essa máquina é um modelo, é necessário instalar o *zabbix-agent*, pois segundo a documentação fornecida pela Zabbix SIA©, o agente deve ser executado no *host* que se deseja monitorar. O agente coleta informações locais sobre o ativo monitorado e posteriormente envia essas informações para o servidor, que emite alertas em caso de falhas. O agente Zabbix é executado como um processo *daemon*.

Quando for necessário criar uma nova máquina virtual, o *zabbix-agent* já estará instalado, bastando apenas configurar o IP do servidor para o qual ele transmitirá os dados do ativo, otimizando assim, o tempo para implantação de um novo servidor virtual.

Com o agente rodando em todas as máquinas, será possível registrar tudo que acontece com elas, pois através dos dados obtidos via *zabbix-agent*, cria-se um histórico desse ativo. Assim, aplica-se o item 10.10 da norma NBR ISO 27002, em que “convém que registros (*log*) de operador e registros (*log*) de falhas sejam utilizados para assegurar que os problemas de sistema de informação são identificados.” (NBR ISO 27002, 2005, p.60) Desta forma, é possível realizar uma auditoria com base nas informações históricas.

A instalação do *zabbix-agent* também representa uma adequação ao item 7.1.1 da norma, em que:

Convém que a organização identifique todos os ativos e documente a importância destes ativos. Convém que o inventário do ativo inclua todas as informações necessárias que permitam recuperar de um desastre, incluindo o tipo do ativo, formato, localização, informações sobre cópias de segurança, informação sobre licenças e a importância do ativo para o negócio. Convém que o inventário não duplique outros inventários desnecessariamente; porém, ele deve assegurar que o seu conteúdo está coerente”. (NBR ISO 27002, 2005, p.21)

Isto é assegurado, uma vez que o *Zabbix server* guarda um conjunto de informações retiradas de seus agentes, e permite a inserção manual de informação, auxiliando os profissionais de tecnologia no monitoramento e inventário de ativos.

A instalação do *zabbix-agent* pode ser realizada através dos seguintes comandos:

```
rpm -Uvh http://repo.zabbix.com/zabbix/4.0/rhel/7/x86_64/zabbixrelease-4.0-1.el7.noarch.rpm ## instalação do repositório oficial do Zabbix, que utilizamos para instalar e manter atualizados os agentes
yum update ## atualização do sistema operacional.
yum install zabbix-agent -y ## instalação do agente Zabbix que irá coletar as informações do sistema

systemctl enable zabbix-agent ##Habilitar o serviço do agente Zabbix para carregar junto a inicialização do sistema.
```

### 3.4.4 Modificação dos pontos de montagem do sistema operacional e remoção da permissão de execução dos arquivos binários

O item 10.4 da norma NBR ISO 27002 trata da proteção contra códigos maliciosos e códigos moveis. A respeito disso,

“convém que sejam implantados controles de detecção, prevenção e recuperação para proteger contra códigos maliciosos, assim como procedimentos para a devida conscientização dos usuários.”  
(NBR ISO 27002, 2005, p. 46)

Nesse sentido, uma medida possível é bloquear a execução dos binários para proteger a máquina da execução indevida, cabendo ao administrador escolher quais podem ser executados para atribuir as permissões. Assim, o bloqueio pode evitar vulnerabilidades como a chamada *SambaCry*<sup>5</sup>, através da qual o invasor faz o *upload* de um arquivo binário contendo um código, fazendo com que o servidor o execute. Essa vulnerabilidade se comporta como o *Wannacry*, criptografando os arquivos do disco.

Outra vulnerabilidade descoberta recentemente é chamada de *Mutagen Astronomy*.<sup>6</sup> Ela é capaz de escalar privilégios para o *root* através de um binário de raiz da *SUID*.

Dados os riscos provenientes da execução de arquivos binários, recomenda-se o bloqueio através da modificação das permissões, que pode ser feita através do comando *chmod -s*. Já, a modificação do ponto de montagem com o bloqueio de execução dos binários pode ser realizada com os comandos a seguir:

---

5 CVE-2017-7494

6 CVE-2018-14634

vi /etc/fstab ## modificação do arquivo de montagem.

UUID=34a6447c-233c-469f-9a9a-48460935a0da /boot xfs defaults,nosuid 0 0 ## linha do ponto de montagem do diretório boot, removendo a permissão de execução dos binários.

/dev/mapper/centos\_vm--padrao-var /var xfs defaults,nosuid,noexec 0 0 ## linha do ponto de montagem do diretório var removendo a permissão de execução dos binários.

/dev/mapper/centos\_vm--padrao-var\_log /var/log xfs defaults,nosuid,noexec,noatime 0 0 ## Linha do ponto de montagem do diretório e log removendo a permissão de execução dos binários, e removendo o rótulo de acesso dos arquivos, assim arquivos que são acessados constantemente ficam mais rápidos.

chmod -s -Rv / ## removendo a permissão de execução de arquivos binários em todo o sistema.

chmod +s /usr/bin/passwd ## Permitindo a execução do binário de troca de senha.

chmod +s /usr/su ## Permitindo a execução do binário de troca de usuário

chmod +s /usr/bin/su ## Permitindo a execução do binário de troca de usuário

chmod +s /usr/sbin/zabbix\_agentd ## Permitindo a execução do binário do agente Zabbix

### 3.4.5 Configuração do *Firewall* e do *SELinux*

Segundo a norma NBR ISO 27002 no item 11.2, “convém que procedimentos formais sejam implementados para controlar a distribuição de direitos de acesso a sistemas de informação e serviços” com o objetivo de assegurar o acesso de usuário autorizado e prevenir acesso não autorizado a sistemas de informação. (NBR ISO 27002, 2005, p. 86)

No item 11.1.1 subitem b das informações adicionais, a orientação é “estabelecer regra baseada na premissa “tudo é proibido, a menos que expressamente permitido”, em lugar da regra mais fraca, “tudo é permitido, a menos que expressamente proibido””. Esta premissa remete-nos a recursos de segurança como o Firewall e o SELinux.

O 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 e 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 fluxo de tráfego de e para esses recursos. (KUROSE, 2010)

Já, o SELinux é um sistema que agrega uma camada adicional de segurança para os servidores. Infelizmente, “mesmo sendo conhecido por grande parte dos administradores de redes/sistemas, o uso mais comum consiste basicamente em “desabilitá-lo”” (BERNARDES, 2014).

Segundo a Equipe de Resposta e Tratamento de Incidentes de Segurança da Informação da Universidade Estadual de Campinas, mais de 90% dos incidentes com Servidores Web Linux poderiam ter sido evitados se o SELinux estivesse ativo (BERNARDES, 2016) .

As estatísticas da Universidade Estadual de Campinas exemplificam a importância de se manter o *Firewall* e o *SELinux* ativos, liberando apenas as portas essenciais para o funcionamento dos serviços que foram destinados para a máquina.

Nesse sentido, uma das portas principais em ambientes virtualizados é a que corresponde ao SSH, que é um protocolo utilizado para acesso remoto que envolve autenticação, criptografia e integridade dos dados transferidos via rede.

Os comandos abaixo podem ser utilizados para liberar o acesso permanente do *firewall* à porta SSH e para instalação do utilitário que permite a adição de novas regras ao *SELinux*, desbloqueando também no *SELinux* a porta do SSH.

```
firewall-cmd --permanent --add-port=2223/tcp ## Adicionando nova regra no
firewall permitindo comunicação na porta 2223
yum install policycoreutils-python ## Instalando programa para configuração
do SELinux
semanage port -a -t ssh_port_t -p tcp 2223 ## Adicionando nova regra de
execução no SELinux
semanage port -l | grep ssh ## Verificando no SELinux quais portas o SSH
pode executar
```

#### 3.4.6 Remoção do terminal padrão dos usuários

Segundo o item 11.2.2, “convém que a concessão e o uso de privilégios sejam restritos e controlados” (NBR ISO 27002, 2005, p. 87). Como orientação no item e, deve-se considerar que “desenvolvimento e uso de programas que não necessitem funcionar com privilégios sejam estimulados”. (NBR ISO 27002, 2005, p.87).



Dado que “usuários que não utilizam *shell* não têm necessidade de tê-la” (MELO et al. 2006), pode-se bloquear o acesso ao *shell* padrão do sistema, liberando o *shell* apenas para usuários que realmente precisem utilizá-lo. Para isso, pode-se editar o arquivo:

```
vi /etc/default/useradd      ## Utilização do editor de Linux para
modificação do arquivo de criação de usuários.
SHELL=/bin/false  ## trecho do arquivo onde encontramos qual terminal
padrão novos usuários vão utilizar
```

### 3.4.7 Modificação das portas padrão

Com base no item 11.4.4 da norma NBR ISO 27002 que diz: “convém que sejam controlados os acessos físico e lógico a portas de diagnóstico e configuração.” (NBR ISO 27002, 2005, p. 93), uma forma de aumentar o nível de proteção da rede é modificar a porta padrão dos serviços mais sensíveis a fim de dificultar a ação dos programas do tipo *portscanners* que procuram vulnerabilidades nas portas padrões.

Dado que “cada serviço disponibilizado por um servidor é uma possível porta de invasão para os crackers” (FRAZÃO e BRAGA, 2015, p.142), é altamente recomendável a alteração de porta padrão, que:

dificulta a ação de programas de enumeração (descoberta de dados) que vasculhem as redes, procurando acesso por essa porta. Outra forma de aprimorar a segurança dos sistemas que possuem um servidor SSH é determinar uma lista de hosts e usuários que terão autorização para a conexão. (SILVA e MARTINS, 2014)

Para realizar a modificação da porta padrão, pode-se editar o arquivo *sshd\_config*, conforme indicação a seguir:

`vi /etc/ssh/sshd_config` ## Utilização do editor do Linux para modificação do arquivo de configuração do servidor SSH

Port 2223 ## trecho do arquivo onde encontramos a porta que o nosso servidor SSH funciona, alterando a porta padrão de 22 para 2223

A identificação 2223 para a porta representa um padrão mais rigoroso de segurança por ser acima de 1024 segundo PERBONI (2013):

Algumas empresas adotam um padrão mais rigoroso de segurança e alteram essa porta para uma porta acima de 1024, pois a maioria dos scanners de portas é baseada em assinaturas de portas já conhecidas, ou seja, serviços que por padrão definem uma porta para execução. Dessa forma será dificultada a identificação do serviço que está em execução na porta do servidor. Portanto, ao conectar com o cliente no servidor SSH, é necessário usar a opção `-p` no cliente para indicar a porta de conexão. (PERBONI, 2013)

Desta forma, aplica-se mais um controle fundamentado na NBR ISO 27002.

### 3.4.8 Bloqueio do acesso ao *root* via SSH

O item 11.4 da norma NBR ISO 27002 diz: “Convém que os usuários com acesso às redes e aos serviços de rede não comprometam a segurança desses serviços.” (NBR ISO 27002, 2005, p.91) Nesse sentido, deve-se realizar alguns ajustes para limitar possíveis brechas de segurança, já que “o usuário *root* é o mais visado por crackers ou usuários mal intencionados” (REIS; JULIO; VERBENA, 2011). Por isso,

para dificultar a ação destas ameaças, desativar o login como usuário *root* nos terminais modo texto torna-se fundamental. Dessa forma, o administrador deverá efetuar o login como usuário

comum e quando for necessário executar uma tarefa administrativa tornar-se root com o comando su. (REIS; JULIO; VERBENA, 2011)

Visto que geralmente atacantes tentam, via força bruta, ter acesso ao usuário *root* em máquinas *Linux*, segundo SIQUEIRA (2010), é uma boa prática realizar o bloqueio de acesso ao root via SSH, pois:

ao bloquear o acesso direto ao usuário root, é acrescentada uma segunda camada de segurança, pois somente após um invasor ou mesmo um usuário legítimo conseguir entrar como um usuário comum é que se poderá fazer o login como root. (SIQUEIRA, 2010)

. Logo, ao bloquear o acesso direto ao *root*, aplica-se mais um controle de segurança à máquina virtual.

Uma das maneiras de realizar esse bloqueio é através do arquivo *sshd\_config*, conforme indicação abaixo:

vi /etc/ssh/sshd\_config ## Utilização do editor do Linux para modificação do arquivo de configuração do servidor SSH.

PermitRootLogin no ## trecho do arquivo onde bloqueamos o login via servidor SSH do usuário root.

### 3.4.9 Bloqueio de acesso aos *sockets* por *hosts*

O item 11.4.4 da norma NBR ISO 27002 diz: “convém que sejam controlados os acessos físico e lógico a portas de diagnóstico e configuração.” (NBR ISO 27002, 2005, p.93). Nesse sentido, “o uso do *TCP wrappers* é uma boa prática na implementação de segurança em redes, limitando o uso dos serviços de rede. (TITON, 2013, p. 37)

Este sistema de segurança possibilita que seja feito um controle de acesso:

Para entender seu funcionamento, imaginemos um sistema recebendo um pedido pela rede procurando um determinado serviço; este serviço passa antes pelo TCP\_Wrappers. O sistema de segurança faz o log do pedido e então confere as regras de acesso. Se não existirem regras de impedimento de um endereço IP ou de um host em particular, o TCP\_Wrappers devolve o controle ao serviço. Caso exista alguma regra de impedimento, a requisição será negada. A melhor definição para o TCP\_Wrappers seria a de um intermediário entre a requisição de uma conexão e o serviço. (JUNIOR, 2010)

Através desta ferramenta é possível aplicar ao template da máquina uma outra medida de proteção, permitindo que apenas alguns *hosts* tenham acesso àquele *socket* de comunicação. Para a utilização do *tcp\_wrappers*, pode-se utilizar os seguintes comandos:

```
yum -y install tcp_wrappers  ## Instalação de um programa para filtrar  
acesso à rede.
```

```
vi /etc/hosts.deny ## Utilização do editor do Linux para modificação do  
arquivo de configuração de hosts bloqueados.
```

```
sshd: ALL    ## trecho do arquivo onde bloqueamos o acesso via SSH ao  
servidor.
```

```
vi /etc/hosts.allow  ## Utilização do editor do Linux para modificação do  
arquivo de configuração de hosts permitidos.
```

```
Sshd: 192.168.1.64 ##Trecho do arquivo onde permitimos o acesso via SSH  
de uma única máquina ao servidor.
```

### 3.4.10 Tempo máximo de acesso do usuário sem interação

Conforme o item 11.5 da NBR ISO27002 que diz: “Convém que recursos de segurança da informação sejam usados para restringir o acesso aos sistemas operacionais para usuários autorizados.” (NBR ISO 27002, 2005, p. 95), tem-se a orientação no item f de permitir “restrição do tempo de conexão dos usuários, quando apropriado.” Esta restrição é um controle de segurança também em relação ao acesso físico e visa evitar que pessoas má intencionadas obtenham acesso ao servidor:

Imagine um administrador logado como root em um dos servidores executando uma tarefa administrativa. Logo em seguida é solicitado sua presença em outro setor, e seu terminal ficou logado como root. Um funcionário mal intencionado pode aproveitar dessa situação e danificar o sistema, ou roubar informações de acesso somente do root. Para REDUZIR, isso mesmo, reduzir essa possibilidade de ataque será utilizada a variável TMOUT. (REIS, 2010)

A variável TMOUT tem como objetivo executar um *logout* automático por tempo em segundos de inatividade no terminal. Esta variável deverá constar no arquivo `tmout.sh`, que pode ser editado através do programa `vi` conforme indicação abaixo:

```
vi /etc/profile.d/tmout.sh    ##Utilização do editor de texto do Linux para
modificação do arquivo de configuração de tempo de acesso ao sistema sem
utilização
TMOUT=900 ##Trecho do arquivo onde setamos o tempo que o usuário
pode ficar sem interação com o sistema
export PATH ##Trecho do arquivo onde forçamos que esse parâmetro seja
utilizado
```

*Deslogar* o terminal que não está em uso é, portanto, uma boa prática relacionada à segurança física, que por muitas vezes é reduzida à noção de acesso aos ativos. Segundo BERARDINELLI (2017),

Quando falamos em segurança física é muito comum imaginarmos apenas os aspectos da proteção no acesso aos ativos tecnológicos como, por exemplo, localização da sala de servidores, sistema de controle de acesso ao local, sistema de prevenção de incêndio, sistema de refrigeração do ambiente, etc. (BERARDINELLI, 2017)

Contudo, é fundamental pensarmos em práticas para:

“melhorar a segurança do sistema (e dos dados) quando o possível atacante já venceu todas as outras barreiras de segurança física e está ali, em frente ao servidor.” (BERARDINELLI, 2017)

Como exemplo de outras boas práticas para atender ao item 11.5, cita-se autenticação de usuários, registro de tentativas de autenticação e registro do uso de privilégios especiais do sistema, entre outras possibilidades.

### 3.5 CONVERSÃO DA MÁQUINA VIRTUAL EM UM NOVO MODELO

Com as configurações realizadas e as políticas de segurança aplicadas à máquina virtual criada, esta pode ser utilizada como *template*. Deste modo, todas as políticas de segurança serão aplicadas a qualquer serviço que for criado no futuro, bastando apenas criar a nova máquina virtual utilizando o *template*. Desta forma, é possível otimizar o tempo para a implantação de novos sistemas.

Para a conversão da máquina em *template*, é necessário deixar a máquina desligada, clicar com o botão direito em cima da máquina que deseja converter em *template*, e clicar na opção *convert to template*, conforme a figura 14:

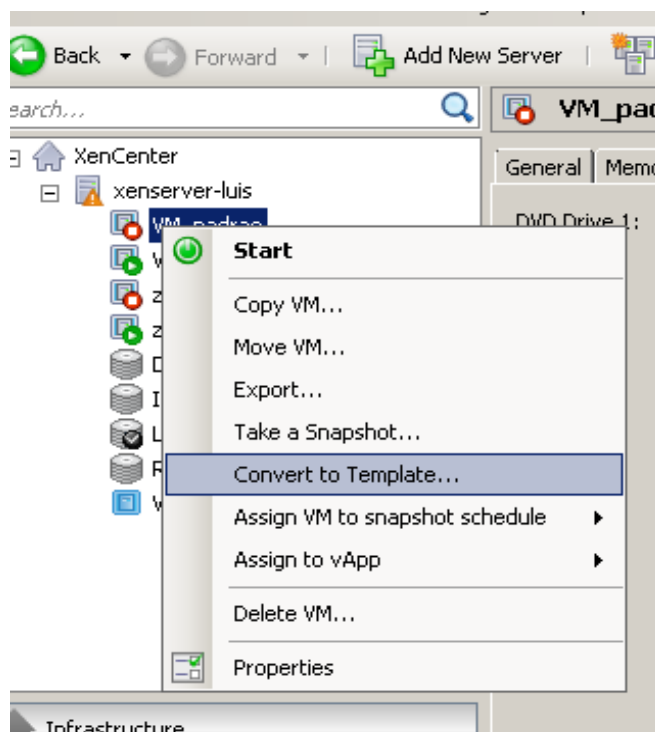


Figura 14. Conversão da máquina em *template*

### 3.6 CRIAÇÃO DE NOVAS MÁQUINAS A PARTIR DO *TEMPLATE* DE SEGURANÇA

Como descrito anteriormente, a partir desse *template* é possível aplicar as políticas de segurança a cada nova máquina virtual implantada. Para a criação de uma nova máquina, pode-se realizar o procedimento indicado a seguir:

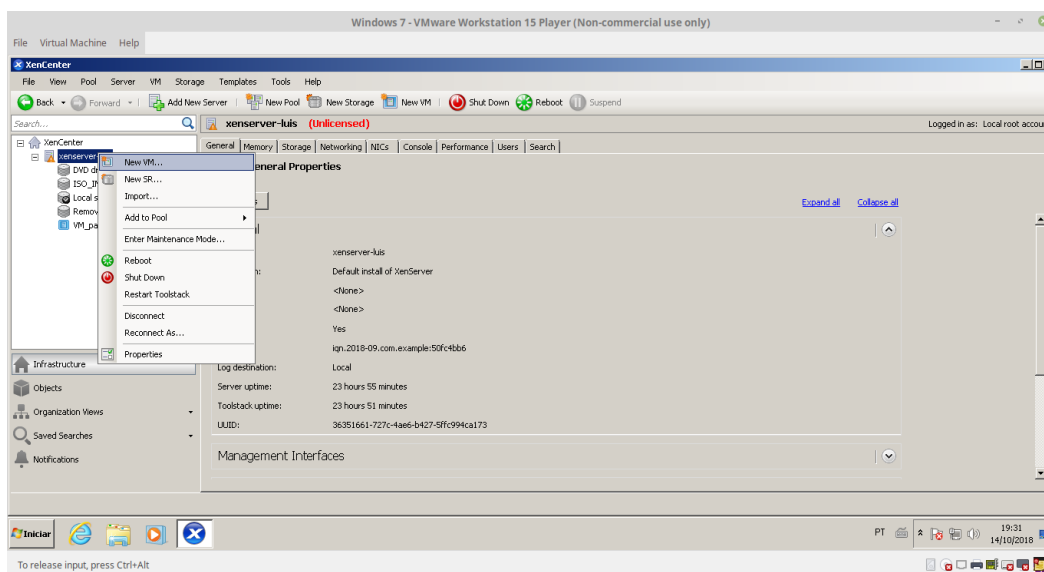


Figura 15. Criação de uma nova maquina virtual

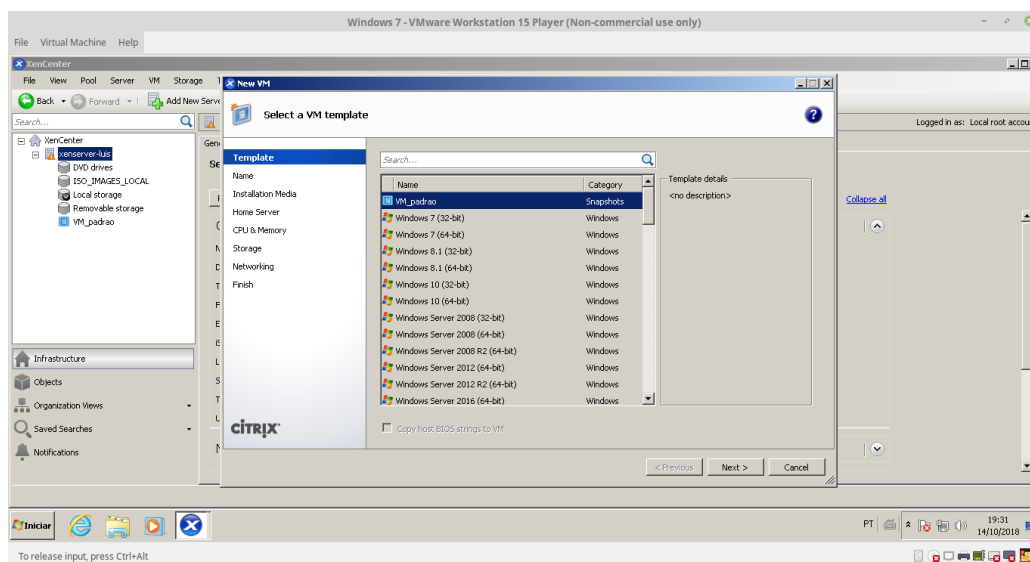


Figura 16. Escolhendo o template para a criação da nova maquina virtual

### 3.7 CONFIGURAÇÃO DO SERVIDOR DE BANCO DE DADOS

Para realizar o monitoramento com o *Zabbix*, é necessário um banco de dados para guardar o histórico de dados. O *Zabbix* oferece suporte a vários bancos.

Neste trabalho, utilizou-se o *PostgreSQL* pela robustez e por possuir várias diretrizes de segurança nativas. A escolha, portanto, baseou-se no



conhecimento do banco de dados, aliado com as políticas de segurança nativas do *PostgreSQL*. Nas seções abaixo descreve-se como é possível realizar a configuração da máquina do Banco de Dados.

### 3.7.1 Configuração da rede na máquina virtual

Como a máquina virtual vem com a configuração e o nome padrão do *template*, é necessário fazer a troca dessas informações. Deve-se atribuir um IP específico para a máquina na rede e deve-se também proceder com a troca do *hostname* para identificação.

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0          ##Utilização do editor
de texto do Linux para modificação do arquivo de configuração da interface
de rede.
BOOTPROTO="static"          ##Trecho do arquivo onde escolhemos o
método estático de endereçamento IP
IPADDR="192.168.1.51"        ##Trecho do arquivo onde se define o IP da
máquina
NETMASK="255.255.255.0"     ##Trecho do arquivo onde se define a
máscara de rede da máquina
hostnamectl set-hostname bd.luistestes                ##Comando para renomear a
máquina.
```

### 3.7.2 Instalação do servidor de banco de dados

Para a instalação do banco de dados, foi utilizado o repositório do *PostgreSQL*, uma vez que os repositórios padrão vêm com um instalador bem desatualizado. Essa abordagem de adição do repositório do próprio *PostgreSQL* é interessante, pois atualizações de segurança futuras poderão ser implementadas mais rapidamente. Seguem os comandos:

```
yum install https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-
x86_64/pgdg-redhat96-9.6-3.noarch.rpm -y  ##Instalação do repositório
padrão do banco de dados Postgres
```

```
yum install postgresql96 postgresql96-server postgresql96-contrib
##Instalação do servidor de banco de dados Postgres e as dependências
para o correto funcionamento do mesmo
postgresql96-libs postgresql96-devel -y
```

```
usr/pgsql-9.6/bin/postgresql96-setup initdb  ##Esse comando tem que ser
executado pois o postgres precisa criar os diretórios e arquivos de controle
do banco
```

```
systemctl enable postgresql-9.6.service  ##Habilitar o serviço do Postgres
para carregar junto a inicialização do sistema
```

```
systemctl start postgresql-9.6.service  ## Iniciar o servidor Postgres
```

### 3.7.3 Criação do usuário para a aplicação

Segundo a norma NBR ISO 27002, no item 11.2.2, “convém que a concessão e o uso de privilégios sejam restritos e controlados” (NBR ISO 27002, 2005, p. 87). Adicionalmente, no subitem f, tem-se a orientação de que “os privilégios sejam atribuídos para um identificador de usuário (ID de usuário) diferente daqueles usados normalmente para os negócios.” (NBR ISO 27002, 2005, p. 87). Para tanto, sugere-se separar os usuários de cada sistema para manter o isolamento e a segurança de cada aplicação. Por isso, criou-se um usuário específico para a aplicação *Zabbix* a fim de manter o banco de dados seguro, visto que muitos administradores costumam usar o usuário *Postgres* para acesso ao banco. Desta forma, o *Zabbix* não tem acesso a outras bases de dados dentro do banco. Seguem os comandos:

```

su – postgres          ## Logando no sistema com o usuário postgres
psql                   ## Usando o programa psql para interação com o
banco de dados
CREATE DATABASE zabbixdb; ## Comando SQL para criar o banco de
dados
CREATE ROLE zabbix LOGIN; ## Comando SQL para criar um usuário no
banco de dados
\password zabbix      ## Comando SQL para criar um usuário para setar a
senha no banco de dados
\q                    ## Comando para encerrar a conexão com o banco de dados

```

#### 3.7.4 Configuração dos arquivos de segurança do PostgreSQL

O PostgreSQL mantém a configuração de acesso ao banco somente localmente. Para os outros serviços que necessitam acessar a base de dados, como o *Zabbix*, é necessário liberar o acesso externo à porta do banco; por questões de segurança, deve-se setar no arquivo *postgresql.conf* uma porta diferente da porta padrão de acordo com a NBR ISO 27002 item 11.4.4 citada na seção 5.4.7, conforme os comandos abaixo:

```

vi /var/lib/pgsql/9.6/data/postgresql.conf ## Utilização do editor de texto do
Linux para modificação do arquivo de configuração dos parâmetros do banco
de dados
listen_addresses = '*' ## Trecho do arquivo onde permitimos que o banco de
dados seja acessado remotamente
Port = 5435           ## Trecho do arquivo onde escolhemos qual porta o
banco de dados vai utilizar

```

Além disso, ainda em conformidade com o item 11.4.4 deve-se bloquear o acesso dos serviços que não necessitam de acesso ao banco.

Neste caso, liberou-se o acesso apenas ao servidor *Zabbix* e ao servidor Web para utilização do *Dashboard*. A liberação de acesso foi feita através da modificação do arquivo de configuração *pg\_hba.conf*:

```
vi /var/lib/pgsql/9.6/data/pg_hba.conf      ##Utilização do editor de
texto no Linux para modificação do arquivo de configuração de acesso ao
banco de dados
host all all 127.0.0.1/32 md5              #Trecho do arquivo onde
declaramos: banco que pode ser acessado, usuário que pode acessar, qual
host tem acesso e qual o método de autenticação no banco
host zabbixdb zabbix 192.168.1.51/32 md5    #Trecho do arquivo
onde declaramos: banco que pode ser acessado, usuário que pode acessar,
qual host tem acesso e qual o método de autenticação no banco
host zabbixdb zabbix 192.168.1.53/32 md5    #Trecho do arquivo
onde declaramos: banco que pode ser acessado, usuário que pode acessar,
qual host tem acesso e qual o método de autenticação no banco.
```

### 3.7.5 Liberação do *SELinux* e do *Firewall* para a comunicação com o banco de dados

Como o servidor de banco de dados deve funcionar fora da porta padrão, é necessário configurar as regras de *firewall* para permitir a comunicação e o *SELinux* para permitir a execução do binário do banco. Utilizou-se o comando de configuração do firewall e o comando de configuração do *SELinux*, lembrando do item 11.2 da norma, citado na seção 5.4.5 deste capítulo.

```
firewall-cmd --permanent --add-port=5435/tcp      ##Adicionando nova
regra no firewall permitindo comunicação na porta 5434.
```

```
semanage port -a -t postgresql_port_t -p tcp 5435      ## Adicionando
nova regra de execução no SELinux.
```

### 3.7.6 Verificação do funcionamento do serviço do PostgreSQL

Para verificar o correto funcionamento da configuração da porta do *PostgreSQL*, uma possibilidade é utilizar o comando *netstat*, pois trata-se, segundo o *Linux Network Administrators Guide*, de “uma ferramenta útil para verificar configuração e atividade de rede”; sendo assim, utilizamos o *netstat* para verificar se as configurações de porta do banco de dados foram realizadas de maneira correta. Através dessa ferramenta é possível conferir se o servidor está “falando” naquela porta e qual é a faixa de endereços que a porta está “ouvindo”.

A saída do comando mostrou que o serviço carregou na porta correta e as configurações foram aceitas, conforme indicação a seguir:

```
systemctl restart postgresql-9.6.service      ##Comando para reiniciar o
servidor postgres
systemctl status postgresql-9.6.service      ##Comando para verificar se
o servidor carregou corretamente
netstat -ntul ## Comando para verificar se a porta do banco de dados está
correta.
```

Além disso, outra possibilidade para analisar processos, específica para o PostgreSQL, é a *pg\_activity*, que cruza informações de recursos do sistema operacional com dados do banco sobre cada processo, mostrando dados como qual a query em execução, há quanto tempo, se está bloqueada por outros processos e outras informações. (CAIUT, 2015)

### 3.8 CONFIGURAÇÃO DO SERVIDOR *ZABBIX*

Nesta seção, apresenta-se as configurações realizadas para execução do servidor *Zabbix*.

#### 3.8.1 Configuração Inicial do servidor

Para criar o servidor *Zabbix*, criou-se uma nova máquina virtual a partir do modelo e configurou-se a rede e o *hostname* da seguinte forma:

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0          ##Utilização do editor
de texto do Linux para modificação do arquivo de configuração da interface
de rede
BOOTPROTO="static"    ##Trecho do arquivo onde escolhemos o método
estático de endereçamento IP
IPADDR="192.168.1.51"  ##Trecho do arquivo onde se define o IP da
máquina
NETMASK="255.255.255.0" ##Trecho do arquivo onde se define a máscara
de rede da máquina
hostnamectl set-hostname zabbix.luistestes          ##Comando para
renomear a máquina
```

#### 3.8.2 Instalação das dependências

Foi necessário instalar algumas dependências para o correto funcionamento do servidor *Zabbix*. A instalação procedeu conforme indicação abaixo:

```
yum -y install curl-devel net-snmp-devel libxml2-devel libssh2-devel
OpenIPMI-devel openldap-devel unixODBC-devel fping net-snmp-utils
libevent-devel ## instalando os pacotes que serão usados pelo Zabbix
server para monitoramento do sistema
```

```
yum install https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-
x86_64/pgdg-redhat96-9.6-3.noarch.rpm -y ## Adicionando o repositório do
banco de dados
```

```
[root@zabbix]# yum install postgresql-devel -y ## Instalando os aplicativos
que serão utilizados para conexão com o banco de dados.
```

### 3.8.3 Compilando o servidor *Zabbix*

A fim de possibilitar a modificação das configurações extras do servidor *Zabbix* sem perder desempenho, optou-se por compilar o *Zabbix* através do código-fonte, recomendação indicada pelo manual fornecido pelo desenvolvedor. Seguem abaixo os comandos utilizados, neste caso, para compilar o *Zabbix*.

```
wget https://sourceforge.net/projects/zabbix/files/ZABBIX%20Latest
%20Stable/4.0.0/zabbix-4.0.0.tar.gz ## baixando os pacotes para a
compilação do Zabbix
cd /usr/src/ ##mudando de diretório para o diretório /usr/local/
tar -zxvf /tmp/zabbix-4.0.0.tar.gz ##descompactando os arquivos para
a compilação do Zabbix
./configure --prefix=/usr/local/zabbix-4.0.0
--sysconfdir=/etc/zabbix --mandir=/usr/share/man --disable-static --enable-
server --disable-proxy
--enable-agent --enable-ipv6 --with-postgresql --without-jabber --with-libxml2
--with-unixodbc
--with-net-snmp --with-ssh2 --with-openipmi --with-ldap --with-libcurl ##Pré
compilação do Zabbix passando os parâmetros conforme a documentação
de compilação do Zabbix
make install ## Compilação dos pacotes do zabbix
```

```
strip --strip-all /usr/local/zabbix-4.0.0/*/* ## Limpando os arquivos de
configuração
```

### 3.8.4 Configurações Iniciais do servidor *Zabbix*

Para o correto funcionamento do servidor e para atualizações futuras, é recomendada a criação de *links* simbólicos, pois com eles é possível compilar o novo *Zabbix* enquanto uma versão antiga está em funcionamento. Desse modo, assim que terminar a compilação, basta trocar o *link* simbólico para a nova versão e ela já estará em funcionamento. Caso o serviço na versão mais atual não inicie corretamente, pode-se criar o *link* simbólico, direcionando-o para a versão antiga, o que fará com que o *downtime* do serviço seja reduzido.

Esta prática de criação de *links* simbólicos é interessante, pois em caso de atualização, as configurações já realizadas apontarão para o *link* simbólico e não para uma versão específica do *Zabbix*, garantindo assim, que as configurações sejam mantidas.

A criação dos *links* simbólicos pode ser realizada conforme indicação abaixo:

```
/usr/local/zabbix-4.0.0/bin/zabbix_get -help ## verificando se o agente do
Zabbix via compilação está funcionando.
```

```
ln -svf /usr/local/zabbix-4.0.0 /usr/local/zabbix ## Link simbólico para a pasta
do servidor Zabbix. Em caso de atualização será necessário apenas mudar o
link, não será necessário reconfigurar tudo com o novo diretório
```

```
mkdir /var/{log,run}/zabbix -p ## Criando os diretórios de log do Zabbix
chown zabbix /var/{log,run}/zabbix ## Atribuindo permissão de acesso ao
servidor Zabbix
```



vi /etc/profile.d/zabbix-path.sh   ## Criando um link para quando necessário chamar o binário do Zabbix, não precisar explicitar o caminho completo.

./etc/profile           ## exportando o profile para podermos utilizar o link criado anteriormente

vi /etc/zabbix/zabbix\_server.conf       ##Utilização do editor de texto do Linux para modificação do arquivo de configuração do servidor Zabbix

ListenPort=10051       ## porta que o servidor Zabbix irá responder

SourceIP=192.168.1.51   ## IP do servidor Zabbix

LogFile=/var/log/zabbix/zabbix\_server.log   ## Diretório de log do servidor Zabbix

DBHost=192.168.1.50   ## IP do servidor de banco de dados onde o servidor vai salvar as informações.

DBName=zabbixdb ## Nome do banco de dados que o servidor Zabbix vai utilizar

DBUser=zabbix       ##Usuário de acesso do servidor Zabbix ao banco de dados

DBPassword=zabbixpw   ## Senha de aceso do servidor Zabbix ao banco de dados

DBPort=5435       ## Porta de comunicação com o servidor de banco de dados

Timeout=30       ## Tempo de espera de conexão com o banco de dados.

### 3.9 CONFIGURAÇÃO DO *DASHBOARD* DO *ZABBIX*

Nesta seção, descreve-se a instalação e configuração do *Dashboard* do *Zabbix*, que é necessário para a visualização dos *hosts*, bem como configurações e monitoramento dos ativos de rede.

#### 3.9.1 Configuração Inicial Dashbord

O *Dashboard* do *Zabbix* utiliza linguagem PHP. Desta forma, é necessário criar um servidor web com o PHP instalado para utilização do *Dashboard*. A criação de uma nova máquina virtual para o servidor *web* foi feita a partir do *template*; nesta nova máquina, configurou-se a rede e o *hostname* da seguinte forma:

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0    ##Utilização do editor de texto
do Linux para modificação do arquivo de configuração da interface de rede
BOOTPROTO="static"    ##Trecho do arquivo onde escolhemos o método
estático de endereçamento IP
IPADDR="192.168.1.53"  ##Trecho do arquivo onde se define o IP da
máquina
NETMASK="255.255.255.0"    ##Trecho do arquivo
aonde define a mascara de rede da maquina.
```

```
hostnamectl set-hostname pache.luistestes    ##Comando para renomear a
máquina
```

#### 3.9.2 Instalação dos pacotes para o servidor *web*

Como o *Dashboard* necessita da linguagem PHP para montar as telas, foi necessária a instalação dos seguintes pacotes:

```
yum -y install php httpd php-gd php-pgsql php-ldap php-bcmath phpmbstring  
php-xml php-xmlwriter php-xmlreader php-ctype php-session php-gettext  
mod_ssl
```

op ## Instalação dos pacotes necessários para a utilização do Dashbord

### 3.9.3 Extração dos arquivos do *Dashboard* para a pasta do Apache

Nesta etapa, propõe-se o isolamento das aplicações, que é um dos maiores benefícios da virtualização, segundo Silva (2007):

Com a virtualização, podemos isolar aplicações de alto risco de aplicações potencialmente vulneráveis. O isolamento eleva a proteção contra aplicações maliciosas, aumentando a dificuldade das mesmas acessarem dados, ou afetar processos que estejam executando em outras máquinas virtuais. (SILVA, 2007, p. 30)

Desse modo, o isolamento pode funcionar como mais uma medida de segurança, prevenindo não só de ataques, mas também como medida de tolerância à falhas:

isolando falhas, prevenimos que uma aplicação em mau funcionamento comprometa todo o sistema. Por exemplo, aplicações que fazem conexões externas, como servidores de e-mail, podem ser confinadas em máquinas virtuais próprias, ou seja, mantendo-as separadas do sistema operacional anfitrião. Além do mais, podemos definir nesta máquina virtual, privilégios para acesso a dados e serviços vitais. Ainda neste contexto, uma máquina virtual pode ser usada para dar privilégios limitados a usuários ou aplicações convidados em um ambiente que pode ser seguramente “deletado” quando o serviço for concluído. Se algumas destas máquinas virtuais estiverem comprometidas, somente seria necessário descartar as máquinas virtuais

corrompidas e iniciar novas outras, ou recuperar cópias de seguranças (backup). (SILVA, 2007, p. 30)

No ambiente utilizado, para isolar as aplicações é necessário copiar os arquivos do *Dashboard* da máquina do servidor *Zabbix* para a máquina do Apache. Desse modo, caso o Apache seja comprometido, não comprometeria também os demais serviços nesta rede.

Foi feita a compactação dos arquivos da máquina do *Zabbix* e em seguida, foi realizada a cópia e a descompactação dos arquivos para a máquina do Apache no diretório *temp*; posteriormente, copiou-se o conteúdo descompactado para o diretório do Apache. As permissões necessárias em cada subdiretório foram feitas da seguinte maneira:

```
cd /tmp/                ##Mudando para o diretório temp
tar -xf frontends.tar   ##Descompactando os arquivos do Dashbord
mv -fv frontends/php/* /var/www/html/  ##Movendo os arquivos
descompactados para a pasta do apache
find html -type d -exec chmod 0750 {} \;  ##Atribuindo as permissões
necessárias para acesso do apache
find html -type f -exec chmod 0640 {} \;  ##Atribuindo as permissões
necessárias para acesso do apache
chown root.apache -R html/  ##Atribuindo as permissões de dono
para acesso do apache
chmod 0770 html/conf      #Atribuindo as permissões
necessárias para acesso do apache.
```

### 3.9.4 Criação de certificados *SSL* para conexão segura com o servidor *Apache*

Como o enfoque desta pesquisa é a segurança, não utilizou-se páginas do tipo *http*, dada a facilidade de sua interceptação; a configuração realizada para o servidor *Apache* exhibe as páginas em *https*. Embora os certificados sejam auto assinados, sabe-se que a comunicação entre o *browser* e o servidor está em modo seguro, evitando que pessoas mal intencionadas interceptem a senha de login do servidor *Zabbix*, graças à criptografia de chave pública-privada transmitida pelo *Apache*. Os comandos abaixo demonstram como as chaves foram geradas.

```
# openssl genrsa -out ca.key 1024
```

```
Generating RSA private key, 1024 bit long modulus
```

```
.....++++++
```

```
e is 65537 (0x10001)
```

```
# openssl genrsa -out ca.key 1024
```

```
Generating RSA private key, 1024 bit long modulus
```

```
.....++++++
```

```
e is 65537 (0x10001)
```

```
# openssl req -new -key ca.key -out ca.csr
```

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [XX]:BR

State or Province Name (full name) []:Rio de Janeiro

Locality Name (eg, city) [Default City]:Rio de Janeiro

Organization Name (eg, company) [Default Company Ltd]: Luis Felipe  
Americo Fernandes

Organizational Unit Name (eg, section) []:TCC

Common Name (eg, your name or your server's hostname)  
[]:apache.luistestes

Email Address []:luisfelipeamerico@gmail.com

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:

An optional company name []:

```
# openssl x509 -req -days 3650 -in ca.csr -signkey ca.key -out ca.crt
```

Signature ok

subject=/C=BR/ST=Rio de Janeiro/L=Rio de Janeiro/O= Luis Felipe Americo

Fernandes/OU=TCC/CN=apache.luistestes/

emailAddress=luisfelipeamerico@gmail.com

Getting Private key

### 3.9.5 Configuração do *SSL* para páginas seguras

Nesta etapa, é necessário configurar o *Apache* para reconhecer os certificados. Para isso, copiou-se cada chave para seus devidos lugares, e setou-se a configuração do certificado no arquivo de configuração do

*Apache*. A configuração desses parâmetros do Apache é pré requisito para o *Dashbord* funcionar.

Os comandos a seguir demonstram a cópia do certificado e a configuração do servidor *Apache*, bem como apresentam os parâmetros que devem ser setados:

```
cp ca.crt /etc/pki/tls/certs ; cp ca.key /etc/pki/tls/private/ca.key ; cp ca.csr/etc/
pki/tls/private/ca.csr ## Cópia das chaves de segurança para as pastas
correspondentes
```

```
vi /etc/php.ini      ##Utilização do editor de texto do Linux para
modificação do arquivo de configuração do servidor PHP
```

```
short_open_tag = On      ##Configuração de permissão de utilização
das tags PHP curtas
```

```
default_charset = "iso-8859-1" ##Configuração do charset correta para
nosso idioma.
```

```
date.timezone = America/Sao_Paulo ## Configuração da hora
correspondente
```

```
post_max_size = 256M      ## Modificação na memória
conforme necessário para o servidor Zabbix
```

```
upload_max_filesize = 256M ## Modificação do tamanho
do arquivo conforme necessário para o servidor Zabbix
```

```
vi /etc/httpd/conf.d/ssl.conf ##Utilização do editor de texto do Linux para
modificação do arquivo de configuração do servidor apache com módulo
SSL
```

```
SSLCertificateFile /etc/pki/tls/certs/ca.crt ##Adição do certificado
gerado
```

```
SSLCertificateKeyFile /etc/pki/tls/private/ca.key    ##Adição da chave publica gerada para autenticação
```

### 3.9.6 Inicialização do servidor e configuração do *Firewall*

Nesta etapa, deve-se iniciar o servidor e configurar o serviço para subir na inicialização do sistema e adicionar a regra no *Firewall* para permitir acesso à página somente na porta segura do servidor. Isso pode ser feito através dos comandos abaixo:

```
systemctl enable httpd    ##Habilitar o serviço do apache para carregar junto a inicialização do sistema.
systemctl start httpd     ##Inicializar o servidor apache.
firewall-cmd --permanent --add-port=443/tcp    ##Adicionando nova regra no firewall permitindo a comunicação do servidor PHP somente na porta segura
firewall-cmd --reload     ## recarregar o firewall do Linux
```

### 3.9.7 Configuração via *browser* para o *Dashbord* comunicar-se com o *Zabbix*

Nesta etapa, deve-se configurar o *Dashboard* para acessar a base de dados e visualizar o *Zabbix server* funcionando. Para isto, foram realizadas as configurações conforme os passos indicados a seguir:

Aqui vemos o nosso certificado SSL criado, validando que essa pagina é o nosso servidor.



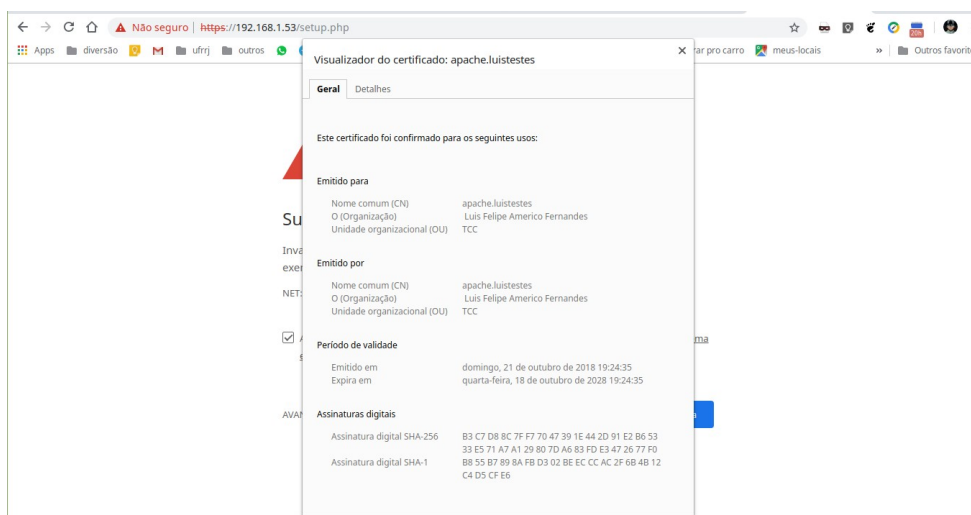


Figura 17. Verificação do certificado gerado na instalação

Na Figura 18, já temos a tela Inicial do servidor *Zabbix*.

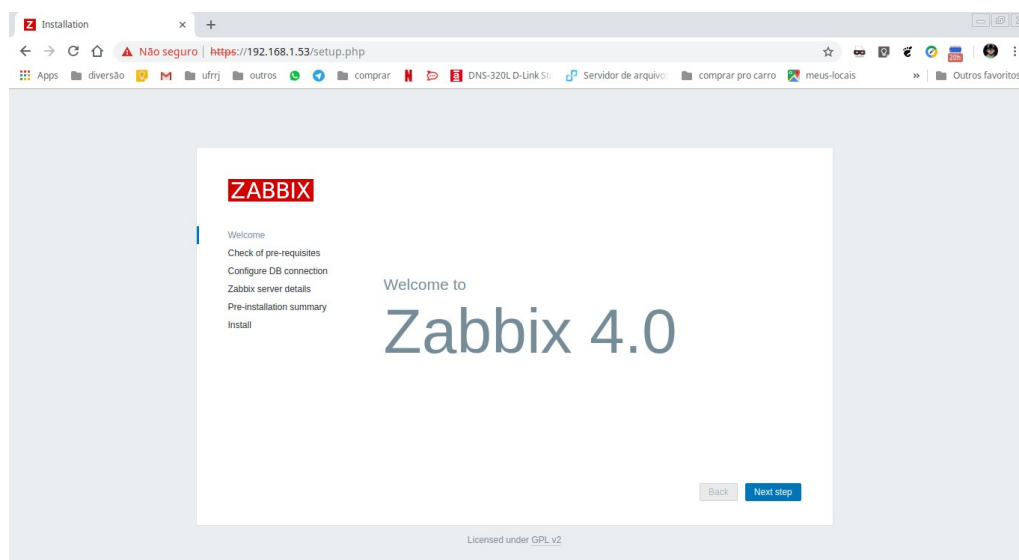
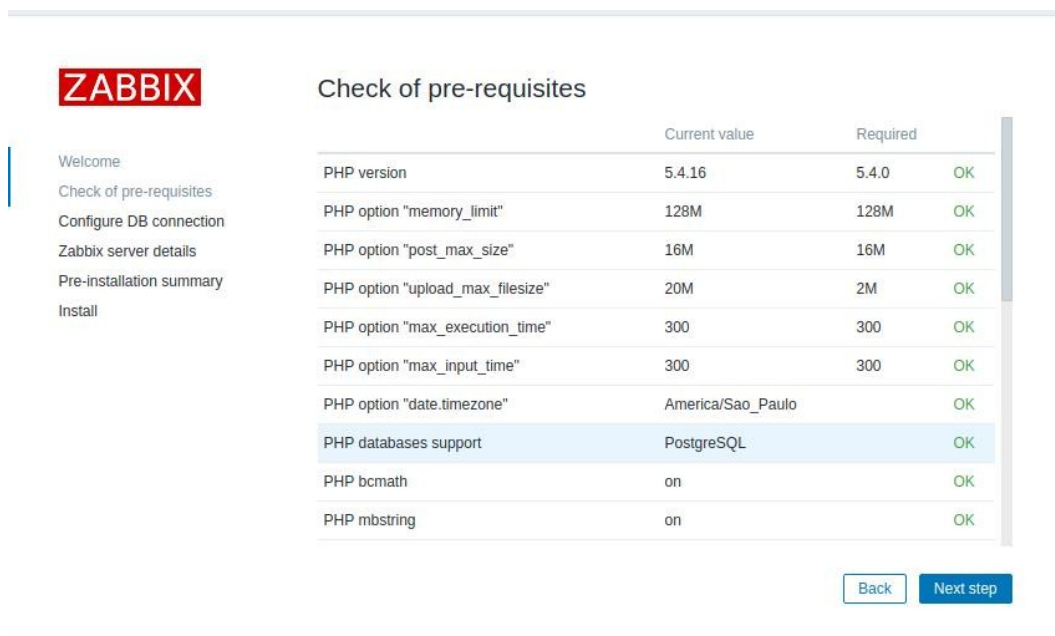


Figura 18. Tela inicial da instalação do Zabbix

Na figura a seguir é possível ver a lista com os pré-requisitos para o completo funcionamento do *dashboard*.



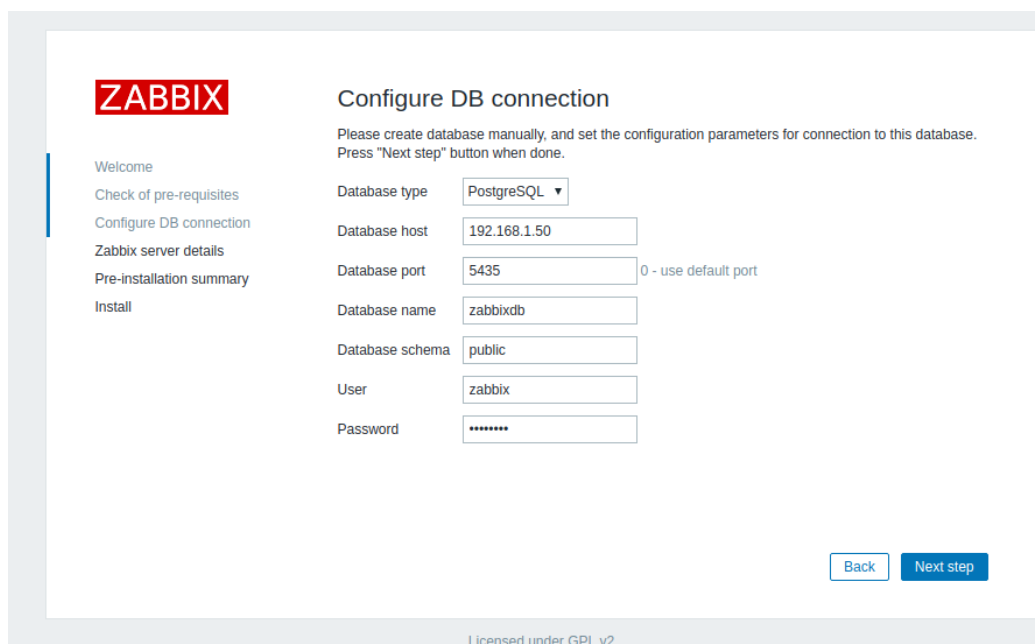
The screenshot shows the ZABBIX installation wizard's 'Check of pre-requisites' step. On the left is a sidebar with navigation links: Welcome, Check of pre-requisites (active), Configure DB connection, Zabbix server details, Pre-installation summary, and Install. The main area displays a table of system requirements.

	Current value	Required	
PHP version	5.4.16	5.4.0	OK
PHP option "memory_limit"	128M	128M	OK
PHP option "post_max_size"	16M	16M	OK
PHP option "upload_max_filesize"	20M	2M	OK
PHP option "max_execution_time"	300	300	OK
PHP option "max_input_time"	300	300	OK
PHP option "date.timezone"	America/Sao_Paulo		OK
PHP databases support	PostgreSQL		OK
PHP bcmath	on		OK
PHP mbstring	on		OK

At the bottom right are 'Back' and 'Next step' buttons.

Figura 19. Verificação dos pré-requisitos da instalação do Zabbix

Em seguida, é possível configurar a conexão com o banco de dados.



The screenshot shows the 'Configure DB connection' step of the ZABBIX installation wizard. The sidebar on the left has 'Configure DB connection' as the active step. The main area contains instructions and form fields for database configuration.

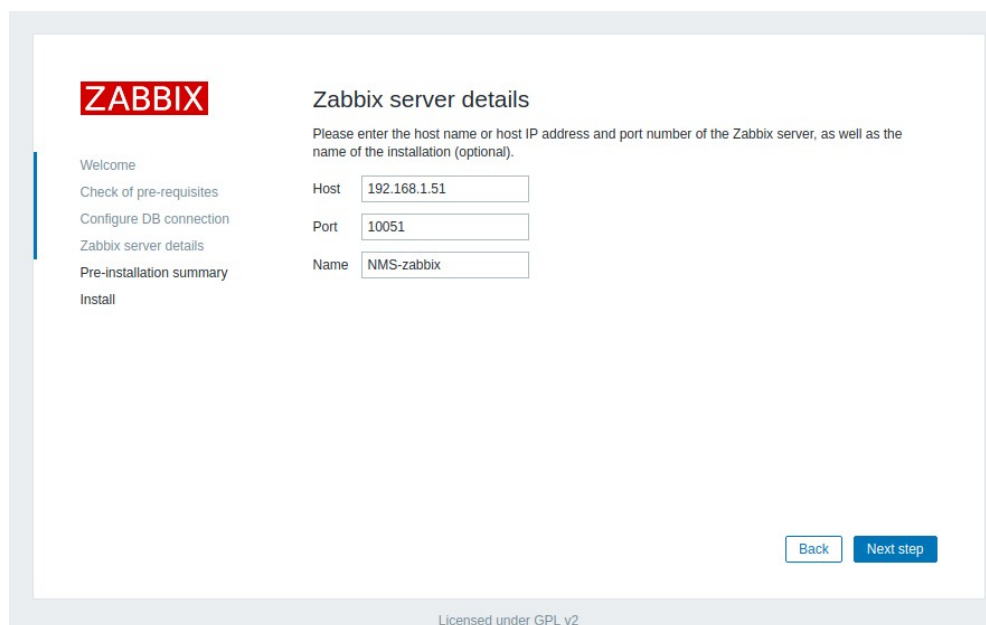
Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Database type	PostgreSQL ▼
Database host	192.168.1.50
Database port	5435 0 - use default port
Database name	zabbixdb
Database schema	public
User	zabbix
Password	*****

At the bottom right are 'Back' and 'Next step' buttons. At the very bottom, it says 'Licensed under GPL v2'.

Figura 20. Configuração do banco de dados

No próximo passo, deve-se configurar o IP e a porta do servidor Zabbix.

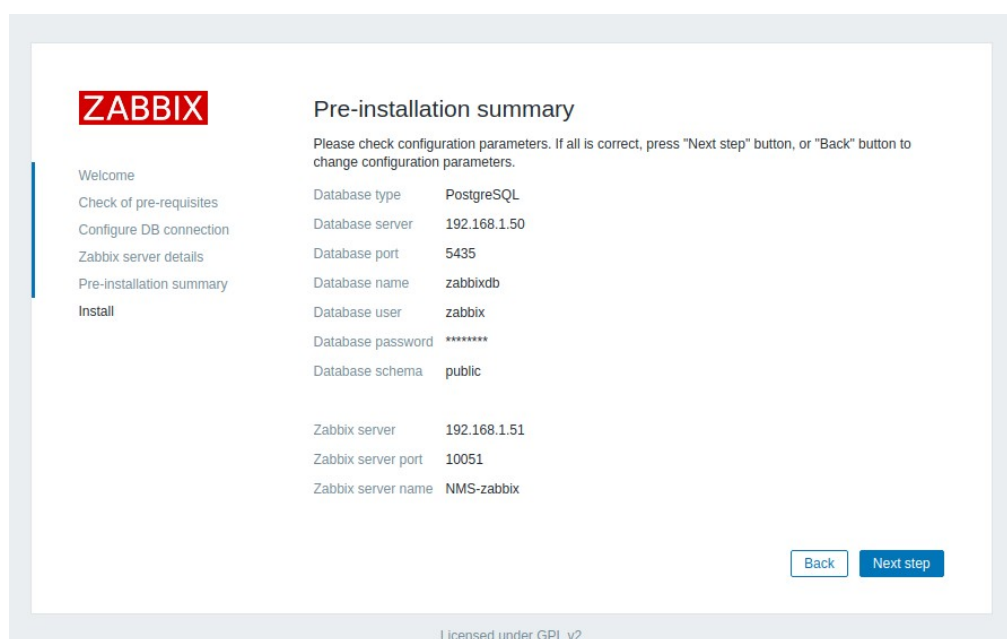


The screenshot shows the 'Zabbix server details' configuration step in the Zabbix installation wizard. On the left, a sidebar lists the installation steps: Welcome, Check of pre-requisites, Configure DB connection, Zabbix server details (highlighted), Pre-installation summary, and Install. The main area is titled 'Zabbix server details' and includes a sub-header 'Please enter the host name or host IP address and port number of the Zabbix server, as well as the name of the installation (optional)'. Below this, there are three input fields: 'Host' with the value '192.168.1.51', 'Port' with the value '10051', and 'Name' with the value 'NMS-zabbix'. At the bottom right, there are 'Back' and 'Next step' buttons. The footer indicates 'Licensed under GPL v2'.

Field	Value
Host	192.168.1.51
Port	10051
Name	NMS-zabbix

Figura 21. Configuração do servidor Zabbix

Em seguida, temos acesso à lista com os parâmetros configurados nas etapas anteriores, mostrando o resumo do que foi realizado.



The screenshot shows the 'Pre-installation summary' screen in the Zabbix installation wizard. The sidebar on the left highlights the 'Pre-installation summary' step. The main area is titled 'Pre-installation summary' and includes a sub-header 'Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.' Below this, there are two sections of configuration parameters. The first section lists database-related parameters, and the second section lists Zabbix server parameters. At the bottom right, there are 'Back' and 'Next step' buttons. The footer indicates 'Licensed under GPL v2'.

Parameter	Value
Database type	PostgreSQL
Database server	192.168.1.50
Database port	5435
Database name	zabbixdb
Database user	zabbix
Database password	*****
Database schema	public
Zabbix server	192.168.1.51
Zabbix server port	10051
Zabbix server name	NMS-zabbix

Figura 22. Verificação da configuração do *Zabbix Dashboard*

### 3.9.8 Configuração dos agentes para o *auto-discovery*

Com o servidor pronto e em execução, prosseguiu-se para a etapa de configuração dos agentes para o *zabbix-server* fazer a busca na rede e encontrar os servidores, iniciando o monitoramento. Deve-se atentar para o fato de que todas as máquinas virtuais deverão ser configuradas do mesmo jeito para que a comunicação com o servidor seja possível.

O item 10.10 recomenda que os sistemas sejam monitorados e eventos de segurança da informação sejam registrados. (NBR ISO 27002, 2005, p. 60). Desta forma, é essencial que se configure todos os agentes para que eles se comuniquem com os servidores, enviando informações sobre todas as informações relevantes da máquina. Para isso, configura-se o arquivo *zabbix\_agentd.conf* para comunicação com o servidor conforme indicação a seguir:

```
[root@zabbix ~]# vi /etc/zabbix/zabbix_agentd.conf      ##Utilização do
editor de texto do Linux para modificação do arquivo de configuração do
agente Zabbix
ListenIP=192.168.150.5      ## IP do host que o Zabbix agent está
rodando
ListenPort=10050            ## Porta de comunicação com o servidor
Zabbix
Server=192.168.1.51         ## IP do servidor Zabbix
Hostname=zabbix.luistestes  ## Mudando o hostname da máquina
[root@zabbix ~]# systemctl restart zabbix-agent.service. ## reiniciar o
serviço do agent Zabbix
```

### 3.10 CONFIGURAÇÃO DO MONITORAMENTO COM *ZABBIX*

Nesta seção, apresenta-se as configurações necessárias para que o servidor efetue a varredura de rede a fim de encontrar todos os ativos que nela estão.

#### 3.10.1 Configuração do *auto-discovery* do servidor *Zabbix*

Como todos os *zabbix-clients* já estão configurados, precisamos configurar o *zabbix-server* para buscar na rede os ativos disponíveis para monitoramento. Existe, no Zabbix, um recurso denominado auto-discovery que procura novos hosts em uma rede. Assim, deve-se configurar o *auto-discovery* na opção *Descoberta* do menu *Configuração*. Cria-se, deste modo, um novo padrão de descoberta, conforme indicado na imagem a seguir:

The screenshot displays the Zabbix web interface for configuring discovery rules. The top navigation bar includes 'ZABBIX', 'Monitoramento', 'Inventário', 'Relatórios', 'Configuração', and 'Administração'. The 'Configuração' menu is expanded, showing 'Grupos de hosts', 'Templates', 'Hosts', 'Manutenção', 'Ações', 'Correlacionamento de eventos', 'Descoberta', and 'Serviços'. The 'Descoberta' sub-menu is selected, leading to the 'Regras de descoberta' page.

The configuration form for a new discovery rule is shown with the following fields:

- \* Nome:** busca servidores
- Descoberta por proxy:** Nenhum proxy
- \* Intervalo de IPs:** 192.168.1.1-254
- \* Intervalo de atualização:** 5m
- \* Checagens:**
  - Agente SNMPv2 "1.3.6.1.2.1.2.2.1.2" (Edit, Remove)
  - Agente Zabbix "agent.ping" (Edit, Remove)
  - Agente Zabbix "system.uname" (Edit, Remove)
  - ICMP ping (Edit, Remove)
  - [Nova](#)
- Critério único do dispositivo:**
  - ☒ Endereço IP
  - ☐ Agente SNMPv2 "1.3.6.1.2.1.2.2.1.2"
  - ☐ Agente Zabbix "agent.ping"
  - ☐ Agente Zabbix "system.uname"
- Ativo:** ☒

At the bottom of the form are four buttons: 'Atualizar', 'Clonar', 'Excluir', and 'Cancelar'.

Figura 23. Configuração da regra de descoberta na rede

Após criar a regra de *descoberta*, configura-se a *trigger* para, assim que um novo ativo de rede for encontrado, o servidor realizar a sua associação com o *template* de monitoramento correto, conforme indicado a seguir:

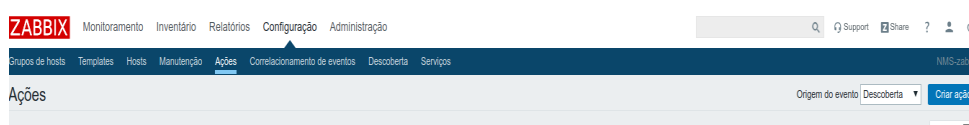


Figura 24. Aba de criação da ação de descoberta

Na tela seguinte, vemos a criação de *triggers* no Zabbix:

**ZABBIX** Monitoramento Inventário Relatórios Configuração Administração

Grupos de hosts Templates Hosts Manutenção **Ações** Correlacionamento de eventos Descoberta Serviços

Ações

Origem do evento: Descoberta Criar ação

Filtrar

---

**ZABBIX** Monitoramento Inventário Relatórios Configuração Administração

Grupos de hosts Templates Hosts Manutenção **Ações** Correlacionamento de eventos Descoberta Serviços

Ações

Ação Operações

\* Nome: Auto discovery. Linux servers.

Tipo do cálculo: E/OU A and B and C

Condições	Texto	Nome	Ação
A	Valor recebido contém	Linux	Remover
B	Status da descoberta de rede igual	Ligado	Remover
C	Tipo de serviço igual	Agente Zabbix	Remover

Nova condição: IP do Host igual 192.168.0.1-127,192.168.2.1

Adicionar

Ativo ☒

\* Ao menos uma operação deve existir.

Atualizar Clonar Excluir Cancelar

Figura 25. Criação da *trigger* para a regra de descoberta

Em seguida, associamos os templates do *autodiscovery*:

Figura 26. Associação dos *templates* para a regra de descoberta

### 3.10.2 Verificação da eficácia do *auto-discovery* na busca pelos ativos de rede

Após a conclusão da etapa anterior, verifica-se se o ativo de rede foi encontrado pelo servidor *Zabbix* e se o mesmo associou o *template* correto para a verificação do ativo de rede. Esse processo pode ser visto na opção Descoberta do menu Monitoramento, conforme a imagem abaixo:

Status das regras de autobusca			Regra de descoberta: todos		
Dispositivo descoberto	Host monitorado	Uptime/Downtime	Agente Zabbix: agent.ping	Agente Zabbix: system.uname	ICMP: ping
<b>busca servidores (8 dispositivos)</b>					
192.168.1.1 (dsdevice.lan)	switch externo	2 dias, 22:59:56			2d 22h 59m
192.168.1.6		1 dia, 09:13:31			1d 9h 13m
192.168.1.50		2 dias, 22:48:32			2d 22h 48m
192.168.1.51 (zabbix.luistestes)	zabbix.luistestes	2 dias, 22:48:22	2d 22h 48m	2d 22h 48m	2d 22h 48m
192.168.1.53	192.168.1.53	2 dias, 22:47:57	2d 22h 47m	2d 22h 47m	2d 22h 47m
192.168.1.64 (raspberrylan)	raspberrylan	2 dias, 22:45:27	2d 9h 30m	2d 9h 30m	2d 22h 45m
192.168.1.200	xenserver	2 dias, 22:13:42			2d 22h 13m
192.168.1.251	switch dmz-interna	2 dias, 22:01:50			2d 22h 1m

Figura 27. *Hosts* encontrados pela regra de autobusca

### 3.10.3 Verificação da atualização do sistema

A atualização de software também pode ser considerada uma medida de segurança. Por isso,

os softwares instalados devem ser atualizados regularmente, visando a expansão de funcionalidades e a correção de falhas de funcionamento detectadas nas versões em uso. Os desenvolvedores também utilizam as atualizações para corrigir falhas de segurança encontradas nos seus pacotes de softwares, que comprometem a segurança do sistema. (FRAZÃO e BRAGA, 2015, p.145)

Tal constatação converge com a NBR ISO 27002, onde encontra-se premissas que incitam a atualização segura. Segundo a norma, “convém que pacotes de correções de software sejam aplicados quando puderem remover ou reduzir as vulnerabilidades de segurança.” (NBR ISO 27002, p. 111)

Essa atualização, ainda segundo a norma, deve ser executada somente por administradores treinados a fim de minimizar os riscos. Nesse sentido,

O administrador deve verificar, com frequência, na página da distribuição utilizada, o lançamento de novas versões de pacotes. Na verdade, os sistemas baseados em pacotes RPM, como Red Hat, CentOS e Fedora, possuem um mecanismo ainda mais poderoso para manter o sistema sempre atualizado. Trata-se do YUM, que é um gerenciador de pacotes baseado no RPM. (FRAZÃO e BRAGA, 2015, p.146)



Uma forma de realizar o *check* de atualizações em diversos servidores sem perder o controle no ambiente, é buscar por *updates* utilizando *scripts*. Isto pode ser feito de forma automatizada através do *Zabbix*.

O *Zabbix* é maleável o suficiente para permitir a execução de *scripts* em *shell*, *python*, *perl* entre outros. Para implementar essa medida, foi criado o “*script sh: /etc/zabbix/externalscripts/packages.sh*” conforme indicado abaixo:

```
echo "UserParameter=packages.status,/etc/zabbix/externalscripts/
packages.sh" >> /etc/zabbix/zabbix_agentd.d/packagegers.conf ##Criação do
arquivo de configuração para o zabbix agent utilizar o script de atualização.
```

```
root@zabbix ~]# vi /etc/zabbix/externalscripts/packages.sh ## script de
busca de pacotes desatualizados
#!/bin/bash ##Marcação do script.
if test -n "$1"
then
echo -e "Opção inválida."
exit 1
fi
rm -Rf /var/tmp/yum-zabbix-* 2>&1 /dev/null ## limpa o temporário aonde o
yum guarda as informações dos pacotes.
RELEASE=`cat /etc/*release 2> /dev/null` ## verifica qual a versão do
sistema operacional
if echo $RELEASE | grep Ubuntu &> /dev/null ##Verifica se o sistema é
base Ubuntu
then
```

```
QUANTITY=`/usr/lib/update-notifier/apt-check 2>&1 | cut -d ";" -f 1` ##Verifica
quantos pacotes existem para atualização usando a ferramenta de
gerenciamento de pacotes do Ubuntu
echo $QUANTITY
exit 0
else
if echo $RELEASE | grep CentOS &> /dev/null ##Verifica se o sistema é
CentOS
then
QUANTITY=`yum check-update -q | grep -v ^$ | wc -l` ##Verifica quantos
pacotes existem para atualização usando a ferramenta de gerenciamento de
pacotes do CentOS
echo $QUANTITY
exit 0
else
echo -e "Opção inválida."
exit 1
fi
fi
```

É necessário setar no *Zabbix* a configuração de disparo desse item, configurando em *administração, scripts* selecionando a opção *criar script*, conforme indicado na imagem abaixo:

**ZABBIX** Monitoramento Inventário Relatórios Configuração Administração

Geral Proxies Autenticação Grupos de usuários Usuários Tipos de mídias **Scripts** Fila

### Scripts

\* Nome

Tipo ☐ IPMI ☒ Script

Executar em ☒ Agente Zabbix ☐ Servidor Zabbix (proxy) ☐ Servidor Zabbix

\* Comandos

Descrição

Grupo de usuários

Grupo de hosts

Permissões de host necessárias ☒ Leitura ☐ Escrita

Ativar confirmação ☐

Mensagem de confirmação

Testar mensagem de confirmação

Figura 28. Criação do script de verificação de atualizações para Linux

Depois, aplica-se esse novo *script* no *template* para monitorar o sistema operacional. No menu configuração, deve-se selecionar o *template* de monitoramento de sistemas operacionais *Linux*, em seguida selecionar a opção *Triggers* e realizar a configuração conforme a Figura 29:

Todos os templates / Template EMPRESA - S.O. Linux   Aplicações 9   Itens 50   Triggers 25   Gráficos 6   Telas 1   Regras de descoberta

Trigger   Dependências

\* Nome

Severidade Não classificada Informação Atenção Média Alta Desastre

\* Expressão  [Adicionar](#)

[Construtor de expressão](#)

Geração de eventos OK Expressão Expressão de recuperação Nenhum

Modo de geração de eventos de INCIDENTE Simple Múltiplo

Fechamentos de eventos OK Todos os incidentes Todos os incidentes que o valor da etiqueta combine

Etiquetas   [Remover](#)

[Adicionar](#)

Permitir fechamento manual ☐

URL

Descrição

Ativo ☒

[Atualizar](#) [Clonar](#) [Excluir](#) [Cancelar](#)

Zabbix 4.0.0

Figura 29. Criação da *trigger* para verificação de atualizações para *Linux*

#### 4. DISCUSSÃO DOS RESULTADOS

Com a implementação dos procedimentos indicados como forma de atendimento à norma NBR ISO 27002, pode-se considerar que a combinação de virtualização com monitoramento realmente é capaz de otimizar o Sistema de Informação, tornando-o mais ágil, disponível e, principalmente, seguro.

A partir do monitoramento da latência, foi possível, entre outros indicadores, identificar instabilidade na rede. O gráfico abaixo exemplifica uma interrupção identificada no serviço de rede.

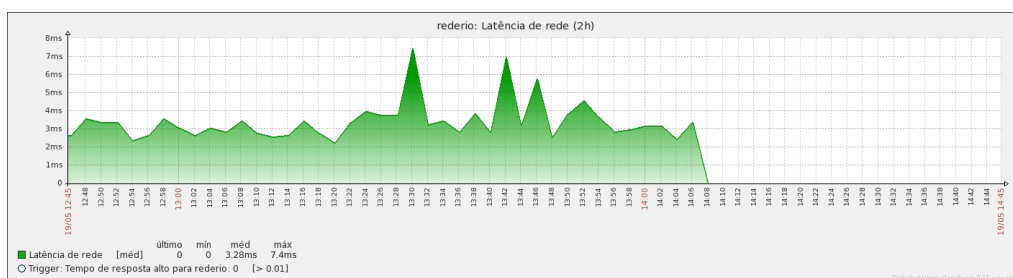


Figura 30. Gráfico de interrupção na conexão de rede

Nesse contexto, ao analisar os demais gráficos de monitoramento, identificou-se instabilidades por parte de um ativo, com momentos de interrupção de comunicação entre o servidor e a rede. A detecção dessas interrupções auxilia a Administração dos serviços, pois nesse caso, foi possível provar e comunicar à prestadora a ocorrência das interrupções. Assim, a prestadora prosseguiu com a análise e solucionou o problema.

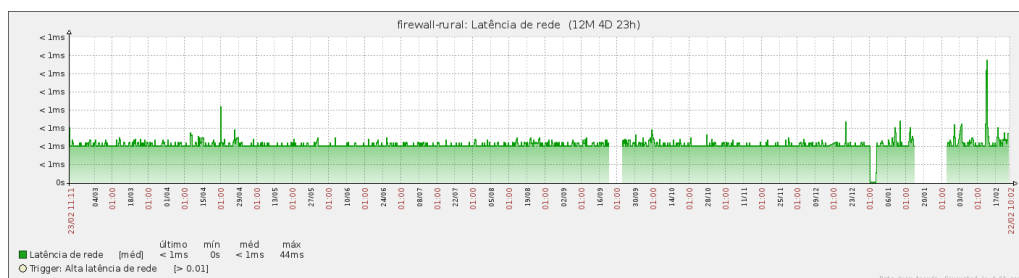


Figura 31. Histórico de interrupções de rede

Com base no monitoramento em tempo real, verificou-se também que é possível ver qualquer ação fora do padrão, visto que há um histórico do comportamento normal do ativo de rede. Assim, qualquer ação fora do padrão pode ser interpretada como potencial ataque, sendo passível de investigação.

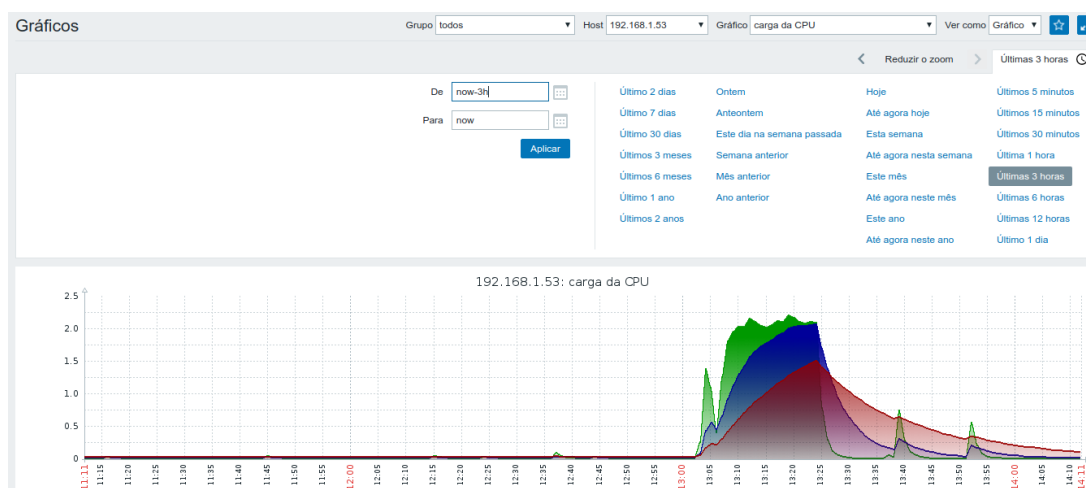


Figura 32. Gráfico de anomalia de utilização de recurso

Na figura 33 há a representação de uma sobrecarga na CPU, mostrando que o recurso já está saturado. Esse resultado foi importante para a gestão do ativo. Dados como esse podem ser utilizados como justificativa, previsão e provisão de novas aquisições.

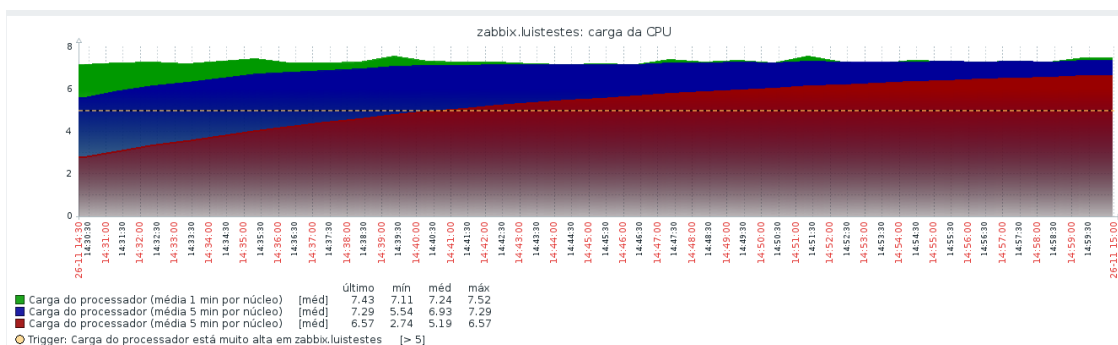


Figura 33. Gráfico de saturação de uma máquina virtual

Na figura 34, observa-se a latência de utilização da placa de rede, sendo isso o trafego de rede. Este dado é importante, pois mostra a utilização da máquina e um possível gargalo no futuro.

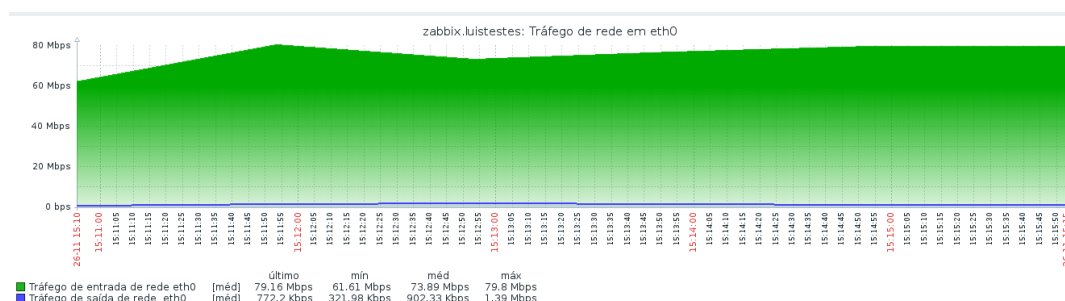


Figura 34. Tráfego de rede em eth0

Essas sobrecargas podem ser vistas na área de visão geral do Zabbix.

Incidentes								
Hora	Hora da recuperação	Status	Informação	Host	Incidente + Severidade	Duração	Reconhecido	Ações
15:30:05	15:32:05	RESOLVIDO		switch externo	Tempo de resposta alto para switch externo: 14ms	2m	Não	1
15:00								
14:27:24		INCIDENTE		zabbix.kuistestes	Carga do processador está muito alta em zabbix.kuistestes	1h 5m 33s	Não	1

Figura 35. Alertas no Dashboard

Segundo a NBR ISO 27002, a segurança da informação é obtida a partir da implementação de um conjunto de controles adequados. Neste sentido, fica clara a importância do monitoramento para garantir que os objetivos do negócio e de segurança da organização sejam atendidos. Esta pesquisa, portanto, buscou, nesse contexto, uma interconexão prática entre os conceitos de monitoramento e virtualização.

A virtualização tem fundamental importância nessa proposta, pois a aplicação das regras de segurança em um ambiente não virtualizado demandaria um grande tempo de implantação, uma vez que cada regra

deveria ser aplicada em  $n$  máquinas. A virtualização possibilita que a aplicação das políticas de segurança seja realizado uma única vez, uma vez que todas as outras máquinas partiriam de um template com todas as políticas de segurança já aplicadas.



## 5. CONSIDERAÇÕES E PERSPECTIVAS

Sendo a informação um ativo que, “como qualquer outro ativo importante, é essencial para os negócios de uma organização e conseqüentemente necessita ser adequadamente protegida” (NBR ISO 27002, 2005, p. 16), esta pesquisa é de grande relevância para o contexto sócio-econômico atual, onde preza-se cada vez mais pela agilidade das soluções e pela segurança.

Desta forma, apresentou-se um *guideline* para atendimento das indicações de segurança presentes na NBR ISO 270002 em ambientes virtualizados, de modo que os procedimentos realizados convergem com os princípios de segurança da informação.

A principal contribuição desta pesquisa foi o estabelecimento de um *guideline* de implantação de técnicas de monitoramento e virtualização com base na norma ISO 27002 utilizando software livre. Este *guideline* pode ser facilmente utilizado para implantação das tecnologias citadas em qualquer ambiente organizacional, independentemente do porte.

Cabe ressaltar que por questões de sigilo e privacidade demandadas pela empresa onde este trabalho foi inicialmente realizado, nem todos os controles implementados, bem como os resultados obtidos, puderam ser apresentados; porém, o conteúdo apresentado demonstra que a hipótese idealizada foi capaz de atender ao problema proposto.

Na execução deste trabalho, uma das limitações foi de caráter financeiro, o que implicou na falta de ativos para testes. Isto comprometeu a investigação envolvendo mais de um servidor e um *storage* de rede, o que forneceria uma importante contribuição referente ao *Work Balancing* (Balanceamento de carga) no *XenServer*.

Para isso, como trabalhos futuros, sugere-se um estudo sobre o recurso *XenMotion*, que utiliza os conceitos de HA (*High Availability*), WLB (*Work Load Balancing*) e *Rolling Pool Upgrade*. Estes recursos, juntos,

possibilitam a movimentação de máquinas virtuais entre *hosts* entre um mesmo pool, com *downtime* quase imperceptível, com perda de poucos pacotes. Esta técnica poderia ser utilizada para aplicar *patches* de segurança, pois neste caso, é necessário reiniciar o servidor para o *patch* entrar em funcionamento. Com este recurso, as máquinas poderiam ser movidas à quente para outro servidor de virtualização, enquanto os *patches* de correção são aplicados. Com isso, as máquinas virtuais continuariam funcionando sem interrupções.

Sugere-se também a implantação de mais controles e o monitoramento de mais serviços e ativos da infraestrutura utilizada. Pode-se também, utilizar ferramentas de teste para detectar diferentes tipos de vulnerabilidade na rede.

Há também a possibilidade de se fazer um estudo comparativo entre políticas de segurança, verificando a relação entre as medidas de segurança e a performance da máquina.

Outra possibilidade seria uma investigação a respeito da aplicação das medidas de segurança da norma NBR ISO 27002 com *Zabbix* em ambientes de contêiner.

Assim, espera-se contribuir cada vez mais para a proteção e valorização do nosso bem maior: a informação.

## 6. REFERÊNCIAS

BERARDINELLI, Mario Luiz. **Segurança física**. 2017. Disponível em <<http://www.mariolb.com.br/LinuxDocs/physical-security.html>> Acesso em 11 dez 2018

BERNARDES, Gesiel. SELinux: Um importante aliado na Segurança de Servidores Linux (Parte 1). Disponível em <<https://www.security.unicamp.br/blog/54-selinux-um-importante-aliado-na-seguranca-de-servidores-linux-parte-1/>> Acesso em 12 dez 2018

BERNARDES, Gesiel. **Dicas para manter seu ambiente web seguro**. Disponível em <<https://www.security.unicamp.br/blog/31-dicas-para-manter-seu-ambiente-web-seguro/>> Acesso em 12 dez 2018

BRAGA, Ascensão. **A Gestão da Informação**. Millenium. N.º 19, junho de 2000. Disponível em: <http://repositorio.ipv.pt/bitstream/10400.19/903/1/A%20GEST%C3%83O%20DA%20INFORMA%C3%87%C3%83O.pdf>. Acesso em 20 de fevereiro de 2010.

BRAGA, Ascensão. **Modernização do Sector Vitivinícola: Modelo de Sistema de Informação**. 1996. Tese (Mestrado em Gestão) – Universidade da Beira Interior, Covilhã/Portugal.

CAIUT, Fabio. **Administração de banco de dados**. / Fábio Caiut. - Rio de Janeiro, Rede Nacional de Ensino e Pesquisa – Escola Superior de Redes, 2015

CAMPOS, Ronaldo; CAZARINI, Edson. **Integrando o sistema de informação gerencial à organização: aspectos da modelagem organizacional segundo a metodologia EKD.** Disponível em <<http://legacy.unifacef.com.br/quartocbs/arquivos/18.pdf>> Acesso em 10 Jan. 2015

CARISSIMI, Alexandre. **Virtualização: da teoria a soluções.** 26° Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos. Rio de Janeiro, 2008

CARVALHO, Rodrigo; FERREIRA, Marta. **Tecnologia da Informação Aplicada à Gestão do conhecimento.** Disponível em <<http://enancib.ibict.br/index.php/enancib/ivenancib/paper/viewFile/2589/1719>> Acesso em 20 nov 2018

CYTRIX SYSTEMS, INC. **Vendor Landscape: Server Virtualization.** Disponível em <[https://www.citrix.com/content/dam/citrix/en\\_us/documents/products-solutions/vendorlandscape-server-virtualization.pdf](https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/vendorlandscape-server-virtualization.pdf)> Acesso em 15. fev. 2015

CITRIX SYSTEMS, INC. **XenServer 6.5.0 Administrator's Guide.** Disponível em <<http://support.citrix.com/article/CTX141500>> Acesso em 08 fev. 2015

CITRIX SYSTEMS, INC. **XenServer 6.5.0 Installation Guide.** Disponível em <<http://support.citrix.com/article/CTX141501>> Acesso em 08 fev. 2015

CITRIX SYSTEMS, INC. **XenServer 6.5.0 Virtual Machine User's Guide**. Disponível em <<http://support.citrix.com/article/CTX141502>> Acesso em 08 fev. 2015

COMPUTERWORLD. **Gartner: virtualização e cloud exigem reforço na segurança**. Disponível em <<https://computerworld.com.br/2011/06/30/gartner-virtualizacao-e-cloud-exigem-reforco-na-seguranca/>> Acesso em 10 dez 2018

DAMASCO, Miguel. **Conceitos de Sistemas de Informação**. Disponível em <<http://www.profdamasco.site.br.com/SlidesFundamentosSI.pdf>> Acesso em 10 Jan. 2015

EY. **20ª Pesquisa Global sobre Segurança da Informação 2017-2018**. Disponível em <<https://www.ey.com/Publication/vwLUAssets/EY-GISS-2017/%24File/GISS-2017-Port.pdf>> Acesso em 23 mar 2018

FRAZÃO, Junior Ari; BRAGA, Marcelo. **Administração de Sistemas Linux**. 2 ed. Rio de Janeiro, Rede Nacional de Ensino e Pesquisa - Escola Superior de Redes, 2015

GIL, A.L. **Sistemas de informações contábil/financeiros**. 3.ed. São Paulo: Atlas. 1999.

GUERRA, Ana Rita. **Mercado de virtualização de servidores cresce 5,7% em 2016**. Disponível em <<https://www.bit.pt/mercado-virtualizacao-servidores-cresce-2016/>> Acesso em 18 fev 2017

HORST, Adail Spínola; PIRES, Aécio dos Santos; DÉO, André Luis Boni; **De A a Zabbix**, NOVATEC, 1ª Edição, 2015.

Info-Tech Research Group Inc. (2016) **Vendor landscape: Server virtualization**. Disponível em < <https://www.infotech.com/research/ss/it-vendor-landscape-server-virtualization> > Acesso em 15 abr 2017

IT Channel. **Mercado de virtualização de servidores atinge o pico da maturidade, segundo a Gartner**. 2016. Disponível em <<https://www.itchannel.pt/news/software/mercado-de-virtualizacao-de-servidores-atinge-o-pico-da-maturidade-segundo-a-gartner>> Acesso em 10 dez 2018

JUNIOR, Marcelino. **Segurança da informação com TCP\_Wrappers**. Linux Magazine outubro de 2010. Disponível em <[http://www.linuxmag.com.br/images/uploads/pdf\\_aberto/LM\\_71\\_68\\_71\\_03\\_seg\\_tcpwrap.pdf](http://www.linuxmag.com.br/images/uploads/pdf_aberto/LM_71_68_71_03_seg_tcpwrap.pdf)> Acesso em 13 dez 2018

KUROSE, James F. **Redes de Computadores e a Internet: uma abordagem top-down** / James F. Kurose e Kith W. Ross; tradução Opportunity translations; revisão técnica Wagner Zucchi. -- 5. ed. -- São Paulo: Addison Wesley, 2010

LEANDRO, Sergio. **Por que os gestores ainda temem a virtualização?** Disponível em <<https://computerworld.com.br/2017/06/28/por-que-os-gestores-ainda-temem-virtualizacao/>> Acesso em 11 dez 2018

LIMA, Gercina Ângela et al. **Tecnologia da Informação: Impactos na sociedade**. <<http://www.uel.br/revistas/uel/index.php/informacao/article/view/1699/1450>> Acesso em 15 Fev. 2015

LIMA, Janssen dos Reis.; **Monitoramento de Redes com Zabbix**, Editora Brasport, 1ª Edição, 2014.

MATHEWS, Jeanna N. **Executando o Xen - Um guia prático para a Arte da Virtualização**. Rio de Janeiro, Editora Alta Books, 2009.

MELO, Sandro. DOMINGOS, Cesar. CORREIA, Lucas. MARUYAMA, Tiago. **BS7799: Da Tática à Prática em Servidores Linux**. Rio de Janeiro: Editora Alta Books, 2006. 232p.

NBR ISO/IEC 27002. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. – **Tecnologia da informação – Técnicas de segurança – Código de prática para a gestão da segurança da informação**. Rio de Janeiro, ABNT, 2005.

NUNES, Rubens; SILVA, Carlos Eduardo de M. Viegas. **A identificação de domínios tecnológicos: o caso da Engenharia de Biossistemas**. Pirassununga, 2013; Disponível em: <http://www.sef.usp.br/wp-content/uploads/sites/9/2014/01/a-identificacao-de-paradigmas-tecnologicos.pdf> Acesso em 15 fev 2017

PERBONI, Marcos. **SSH: Protocolo seguro para acesso remoto**. 2013. Disponível em: <<http://marcosvperboni.wordpress.com/2013/02/15/ssh-protocolo-seguro-para-acesso-remoto/>>. Acesso em: 27 nov. 2018

PORTAL TECNOLOGIA UOL. **Ransomware: metade das empresas brasileiras já sofreu sequestro de sistemas.** Disponível em <<https://tecnologia.uol.com.br/noticias/reuters/2017/03/13/metade-de-empresas-do-brasil-ja-sofreu-sequestro-de-sistemas-digitais-diz-pesquisa.htm>> Acesso em 16 jul 2017

PWC. **Principais conclusões da Pesquisa Global de Segurança da Informação 2016.** Disponível em <<https://www.ey.com/Publication/vwLUAssets/EY-GISS-2017/%24File/GISS-2017-Port.pdf>> Acesso em 04 nov 2017

REIS, Flavio A; JULIO Eduardo P; VERBENA, Marcos F. **Hardening.** 2011. Disponível em <<https://www.devmedia.com.br/hardening-artigo-revista-infra-magazine-1/20818>> Acesso em 11 dez 2018

SENADO FEDERAL. **Política pública de Utilização de Software Livre.** Caderno 3. Brasil, Junho de 2012. Disponível em <<http://www2.senado.leg.br/bdsf/bitstream/handle/id/243078/Caderno3.pdf>> Acesso em 10 dez 2018

SILVA, Rodrigo Ferreira. **Virtualização de Sistemas Operacionais.** Petrópolis, RJ. 2007. Disponível em <<http://www.Incc.br/~borges/doc/Virtualizacao%20de%20Sistemas%20Operacionais.TCC.pdf>> Acesso em 10 dez 2018

SILVA, Pablo Rezende e MARTINS, Márcio Silva. **O uso da Ferramenta de Honeypot kippo para a segurança de redes.** Caderno de Estudos em Sistemas de Informação, CES/JF, 2014.



Singh, A. **An introduction to virtualization.** Disponível em <<http://www.kernelthread.com/publications>> acesso em 15 fev 2017

SIQUEIRA, Luciano Antonio. **Infraestrutura de redes.** / Luciano Antonio Siqueira. - São Paulo: Linux New Media do Brasil Editora Ltda, 2010.

THOMPSON, Michael. **Seis armadilhas a serem evitadas na virtualização e na adoção em nuvem.** Disponível em <<http://tiinside.com.br/tiinside/17/07/2014/seis-armadilhas-serem-evitadas-na-virtualizacao-e-na-adocao-da-nuvem/>> Acesso em 10 dez 2018

TITON, Regis. **Segurança da Informação aplicada a servidores utilizando técnicas de Hardening.** Disponível em <[https://repositorio.ufsm.br/bitstream/handle/1/249/Titon\\_Regis.pdf?sequence=1&isAllowed=y](https://repositorio.ufsm.br/bitstream/handle/1/249/Titon_Regis.pdf?sequence=1&isAllowed=y)> Acesso em 01 dez 2018

VALOR ECONÔMICO. **Virtualização chega a dispositivos de rede e ao desktop.** Disponível em <<https://www.pressreader.com/brazil/valor-econ%C3%B4mico/20180320/282230896225194>> Acesso em 10 dez 2018

Vieira, Claudia S.; Mereilles, Fernando de Souza; and Cunha, Maria Alexandra, **"Fatores que influenciam o indivíduo na utilização da Computação em Nuvem"** (2015). CONF-IRM 2015 Proceedings.28. Disponível em <<http://aisel.aisnet.org/confirm2015/28>>. Acesso em 09 dez 2018

ZABBIX SIA. **Zabbix Documentation 2.2.** Disponível em  
<<https://www.zabbix.com/documentation/2.2/>> Acesso em 10 fev. 2015

## 7. ANEXOS

[https://github.com/luisfelipeamerico/TCC\\_Luis\\_Felipe](https://github.com/luisfelipeamerico/TCC_Luis_Felipe)

## 8. GLOSSÁRIO

*Apache* – É um servidor web livre.

*Backup* – Cópia de dados de um dispositivo de armazenamento a outro para que possam ser restaurados em caso da perda dos dados originais, o que pode envolver apagamentos acidentais ou corrupção de dados.

*Big Data* – Conjuntos de dados muito grandes ou complexos, que os aplicativos de processamento de dados tradicionais ainda não conseguem lidar. Os desafios desta área incluem: análise, captura, curadoria de dados, pesquisa, compartilhamento, armazenamento, transferência, visualização e informações sobre privacidade dos dados.

*Cloud Computing (Computação em nuvem)* – Refere-se à utilização da memória e da capacidade de armazenamento e cálculo de computadores e servidores compartilhados e interligados por meio da internet.

*Cluster* – Computadores interligados que trabalham em conjunto para que, em muitos aspectos, eles possam ser vistos como um único sistema.

*Dashboard* – Os *dashboards* são painéis visuais que centralizam informações importantes para o negócios. Permitem, desta forma, entender o cenário em tempo real e tomar decisões baseadas em informações reais e que estão ocorrendo agora, monitoradas minuto a minuto.

*Firewall* – O *firewall* é um dispositivo de uma rede de computadores que tem por objetivo aplicar uma política de segurança a um determinado ponto da rede.

*Firewalld* – *Firewalld* é uma ferramenta de gerenciamento de *firewall* para sistemas operacionais Linux utilizado amplamente nos sistemas *Red Hat Like*.

*Firewall\_cmd* – *Comando do firewalld*

*Hardening* – É um processo de mapeamento das ameaças, mitigação dos riscos e execução das atividades corretivas e com o foco na infraestrutura, e objetivo principal de torná-la preparada para enfrentar tentativas de ataques.

*Hypervisor* – *Hypervisor* é uma camada de *software* entre o *hardware* e o sistema operacional. O *Hypervisor* é responsável por fornecer ao sistema operacional visitante a abstração da máquina virtual. E é ele que controla o acesso dos sistemas operacionais visitantes aos dispositivos de *hardware*.

*Host* – *Host* é qualquer computador ou máquina conectado a uma rede, que conta com número de IP e nome definidos. Essas máquinas são responsáveis por oferecer recursos, informações e serviços aos usuários ou clientes.

*HTTP* - (*HyperText Transfer Protocol*) Protocolo usado para transferir páginas *Web* entre um servidor e um cliente.

*IP* - Sequência de números associada a cada computador conectado à Internet.

*Linux* – É um sistema operacional que utiliza o *Kernel Linux*

*Opensource* – (Código aberto) é um modelo de desenvolvimento que promove um licenciamento livre para o design ou esquematização de um

produto, e a redistribuição universal desse design ou esquema, dando a possibilidade para que qualquer um consulte, examine ou modifique o produto.vas de ataques.

*Portscanners* – Scanner de Porta. É um software malicioso que tem como objetivo mapear as portas TCP e UDP.

*Ransomware* – *Software* malicioso que restringe o acesso ao sistema infectado e cobra um resgate.

*Root* – Termo usado para caracterizar o processo que permite o uso de um sistema operacional baseado no Unix/Linux como superadministrador.

*SELinux* – *Mecanismo de Segurança MAC (Mandatory Access Control) do Kernel do Linux*

*Shell* - Interface de usuário para acessar os serviços de um sistema operacional. Serve de ponte entre o usuário e o sistema.

*Snapshot* – Retrato do estado de um sistema em um estágio específico

*Sockets* – Abstração para endereços de comunicação através dos quais processos se comunicam.

*SSH* – O SSH (*Secure Shell*) é um protocolo que permite a você acessar virtualmente o servidor como se você estivesse em um terminal.

*SSL* – O *SSL (Secure Socket Layer)* é um protocolo que fornece confidencialidade e integridade nas comunicações entre um cliente e um servidor, podendo ser também usado para prover autenticação.

*VI* – Editor de texto dos sistemas operacionais baseados em Unix

Zabbix\_server – Servidor que gerencia a coleta e recebimento de dados, calcula o estado das triggers e envia notificações aos usuários.

Zabbix\_agent – Ferramenta de monitoramento de redes, servidores e serviços, pensada para monitorar a disponibilidade, experiência de usuário e qualidade de serviços.