TECNOLOGIAS E MONITORAMENTO EM REDES DE COMPUTADORES: implementação de tecnologias e, exploração de recursos de monitoramento

Victor Renan Silva de Jesus
Graduando em Tecnologia em Redes de Computadores pela Fatec Bauru
victor.jesus7@fatec.sp.gov.br

Gabriel Hungaro Martins
Graduando em Tecnologia em Redes de Computadores pela Fatec Bauru
gabriel.martins35@fatec.sp.gov.br

Henrique Pachioni Martins
Docente na Fatec Bauru
henrique.martins01@fatec.sp.gov.br

RESUMO

Quando se fala em segurança de rede, pensamos apenas em softwares bloqueadores, entretanto, além da extrema importância destes softwares, a segurança de uma rede, não se trata apenas de prevenção de tentativas de invasões, mas também, do próprio funcionamento dela. A proposta principal deste artigo, é mostrar como funciona um ambiente de monitoramento, pontos críticos, e tecnologias em redes de computadores. Para a construção do ambiente, foi utilizado software de virtualização e conteinerização, máquinas físicas e, softwares de monitoramento de código aberto. A conclusão sobre o tema, é que com um ambiente de monitoramento harmônico, tivemos controle total sobre o que acontece com os dispositivos conectados à rede e sobre o que trafega por ela.

Palavras-chave: Virtualização. Conteinerização. Dispositivos. Rede. Ambiente. Código aberto.

1 INTRODUÇÃO

A tecnologia evoluiu muito, hoje, é imprescindível uma empresa não usufruir da tecnologia em seu negócio, seja para armazenamento e análise de dados, vendas online, suporte ao cliente, enfim, existem infinitos benefícios ao integrar o negócio a tecnologia. Para que seja possível essa integração, é necessário equipamentos, como por exemplo, racks com switches, hubs, controladora, roteadores, entre diversos outros equipamentos necessários para o fornecimento e segurança da rede. Além dos equipamentos, existem pessoas mal-intencionadas, por n motivos, por exemplo, ex-funcionários insatisfeitos com a empresa, hackers, concorrentes, entre outros.

Em 2021, segundo Galvani (2022), o Brasil teve um registro de mais de 9 bilhões de tentativas de ataques cibernéticos, sendo o terceiro país do mundo, dentre os que mais tiveram uma crescente nos ataques. Galvani, também prevê uma alta nos ataques de phishing, deepfake, ransomware, infoinstaler e webskinner. Dentre os tipos de ataques citados anteriormente, o phishing é o mais efetivo e o mais utilizado. Segundo Rodrigues (2018), em uma matéria no site da empresa de cyber segurança, a Kaspersky, em 2017 a solução de ponto a ponto fornecido pela empresa, bloqueou cerca de 37 milhões de ataques na América Latina, dados apenas dos sete primeiros meses. O phishing, possuí essa eficiência, porque consiste em engenharia social, seja um link malicioso enviado por e-mail, SMS, ou por aplicativo de mensagem, até mesmo pessoas se passando por algum conhecido ou membro da empresa. Fora os riscos de invasão, existem os problemas físicos com os equipamentos citados anteriormente, como por exemplo, queda ou oscilações de energia, mal funcionamento de um componente, rompimento de cabos e fios, equipamentos desligados, temperatura elevada, movimentação estranha nas portas dos equipamentos, entre outros. E, além dos equipamentos, é de extrema importância, acompanhar o funcionamento dos sistemas e serviços, se estão fora do ar, travados, apresentando lentidão, atividades suspeitas ou em mal uso, pois qualquer problema, pode impactar diretamente nos resultados e objetivos da empresa.

Visto todos os possíveis incidentes citados, é possível entender a necessidade de um monitoramento adequado. Este documento, se trata da implementação e de exploração de recursos para que seja mantido a integridade da rede, e em caso de incidentes, ter um rápido diagnostico e, maior agilidade para resolver a situação, e assim, não comprometa os dados, funcionalidade, e a rotina de uma empresa.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Virtualização

VMware Workstation, Oracle VirtualBox, Microsoft Hyper-V, entre outros, são exemplos de softwares de virtualização chamados de hipervisors. Com os hipervisors, temos controle sobre o provisionamento das máquinas hospedeiras, ou, máquinas virtuais. A virtualização nos permite criar um “computador dentro de um computador”, segundo Gogoni (2019), o hipervisor é um software capaz de simular um ambiente computacional, que é capaz de executar sistemas operacionais. Todos os recursos utilizados para a virtualização, depende inteiramente do hardware (componentes físicos) da máquina principal (denominado host) para o seu funcionamento, como por exemplo, núcleos do processador, memória RAM, memória de armazenamento, placa de rede, entre outros. Exemplificando, se a máquina host possui 20 gigabytes (GB) de memória RAM, e, para máquina virtual, foi definido 10 gigabytes, enquanto estiver ligada, os 10 GB de memória RAM serão utilizados da máquina host, assim funciona com todos os recursos da máquina virtual, se a máquina host tiver 10 núcleos de processamento, e, o hipervisor que está sendo utilizado permite criar dois núcleos virtuais (VCPUs) para cada núcleo físico, e, para máquina virtual fora definido 4 núcleos, enquanto ela estiver ligada, a máquina host, trabalhará apenas com 8 núcleos.

2.2 Monitoramento

Para monitorar a rede, é necessário softwares que tragam informações o suficiente para termos controle total, indicando a saúde e disponibilidade do hardware, espaço em disco, criação de alertas, disponibilidade da conexão prevenção e detecção de ataques, uso e disponibilidade de memória e processamento, organização de usuários em máquinas, permissões, armazenamento e controle de logs, atualizações, comunicação de fora da rede, e arquitetura para facilitar a visualização e monitoramento. Os softwares como o Zabbix e o Wazuh, nos permite criar uma interface totalmente personalizada, e, também, nos permite integrá-lo a outros sistemas, para incluirmos como por exemplo uma máscara, e dessa forma então, fazer com que a interface fique ainda melhor para obter um diagnóstico, que neste caso, seriam os gráficos e tabelas do Grafana, uma aplicação web de código aberto que podemos fazer o uso de gráficos e tabelas para melhorar a visibilidade dos gráficos e dashboards. E neste estudo, faremos uso do Zabbix, Wazuh, Grafana, Snort

2.2.1 IDS

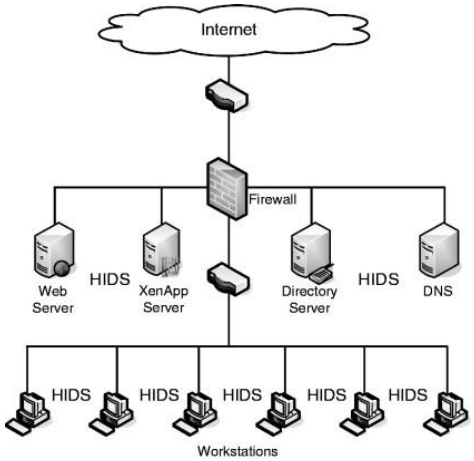
Sistema de detecção de intrusão (em inglês, Intrusion Detection System – IDS), é um sistema que é utilizado para monitorar a rede, ou, um host específico.

Quando se tratando de um IDS passivo, monitora o tráfego de dados na rede, através dos protocolos TCP/IP e UDP, detecta atividades fora do comum na rede, armazena logs, e, gera alertas para o administrador. Já o ativo, também conhecido como Sistema de Proteção de Intrusão (em inglês Intrusion Protection System – IPS), pode tomar decisões quando detecta atividades que por ele, seja reconhecido como incomum, tudo depende da configuração.

Tanto o IPS, quanto o IDS, podem ser configurados para detectar essas atividades.

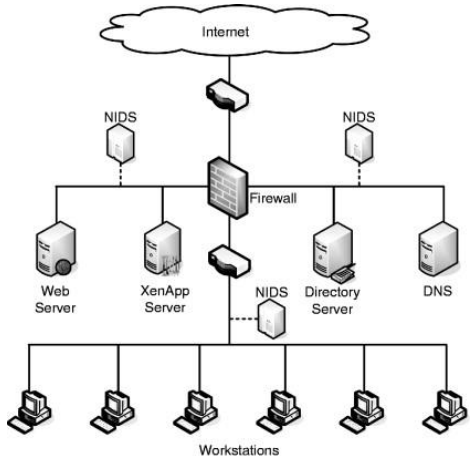
Para este artigo, utilizamos o Snort, para monitorar um computador pessoal, sendo dessa forma, o sistema pode ser reconhecido como Sistema Hospedeiro de Detecção de Intrusão (em inglês Host-based Intrusion Detection System - HIDS), que é o caso de quando o IDS ou IPS é configurado para monitorar um host ao invés da rede. No caso de ser configurado para monitorar a rede, pode ser chamado de Sistema de Rede de Detecção de Intrusão (em inglês Network-based Intrusion Detection System - NIDS). A imagem 2 ilustra como funciona um HIDS, e a imagem 2 como funciona um NIDS.

Imagem 1 – Exemplo de HIDS



Fonte: sciencedirect.com

Imagem 2 – Exemplo de NIDS



Fonte: sciencedirect.com

Existem muitas vantagens de se usar um IDS como NIDS, como por exemplo:

a) É quase independente das máquinas que estão contidas na rede;
b) É completamente eficaz contra varreduras nas portas;
c) Dificilmente é identificado por atacantes.

E as maiores desvantagens são:

a) Não é capaz de detectar ataques criptografados;
b) Se mal configurado, pode tomar decisões que afetam diretamente o desempenho da rede.

Já as vantagens de se usar um HIDS:

a) Monitora completamente o comportamento da máquina hospedeira, estado do SO, informações de armazenamento, controle do hardware e, um dos principais motivos, tráfego de rede.
b) Não gera gargalo na rede

Desvantagens:

a) É necessário a instalação em cada máquina;
b) Consome recursos de hardware;
c) Aumenta a latência e tráfego de rede.
d) São mais suscetíveis a ataques, por estarem diretamente no host e, na maioria dos casos, possuem privilégios de administrador.

2.2.3 Protocolo ICMP

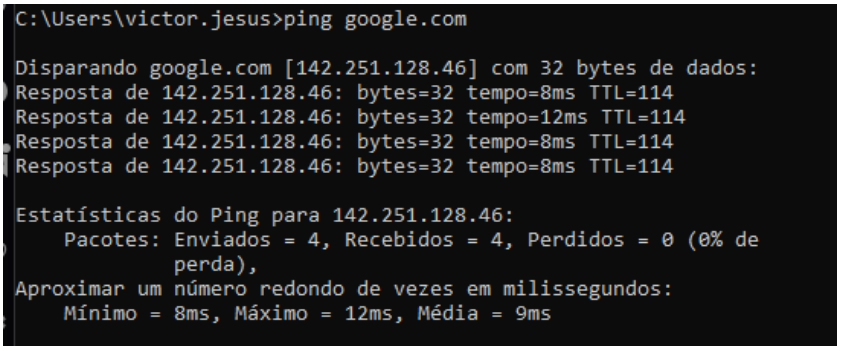
Internet Control Message Protocol (protocolo de mensagens de controle da internet), é um protocolo da camada 5 do modelo ISO/OSI (camada de Rede). É um protocolo utilizado para diagnósticos de problemas de comunicação de rede determinando a integridade e velocidade dos dados que estão chegando ao endereço destino.

Sendo um protocolo essencial para extrair relatórios e realizar testes de comunicação e disponibilidade. Mesmo com excelentes pontos para a segurança de uma rede, o ICMP também pode ser utilizado para ataques de negação de serviços (em inglês - Distributed Denial of Service - DDoS). O objetivo principal do ICMP, é relatório de erros. Quando a conexão entre dispositivos é feita, o protocolo ICMP acaba gerando erros e compartilhando com o dispositivo, e caso qualquer dado não chegar ao destinatário, o pacote de dados é descartado e é retornado uma mensagem ICMP para o dispositivo remetente.

Existem ferramentas extremamente úteis que fazem uso do ICMP para diagnostico de comunicação, por exemplo o ping e o tracert, que são comandos utilizados para saber a quantidade de saltos (TTL – em inglês -Time To Live) entre dispositivos de um dispositivo para outro antes de ser descartado, sendo no máximo 255. O ping, testa a velocidade (latência) da conexão. Já o traceroute,

mostra os dispositivos nos quais os pacotes precisam trafegar para chegar ao endereço de destino. A proxima imagem mostra o comando ping sendo utilizado com o site da google;

Imagem 3 – Exemplo do comando ping



Fonte: Os autores

A imagem 3, é um exemplo do comando ping sendo utilizado através do terminal prompt de comando do Windows, para o endereço do Google, na imagem, podemos entender que foram enviados quatro pacotes e, foi recebido pelo destino, quatro pacotes de 32 bits, ou seja, a comunicação entre os dois endereços está sendo realizada.

Imagem 4 – Exemplo do comando tracert

Fonte: Os autores

Na imagem 4, temos um exemplo da rota realizada entre o dispositivo

pessoal até o servidor do Google, utilizando o comando tracert no terminal PowerShell do Windows

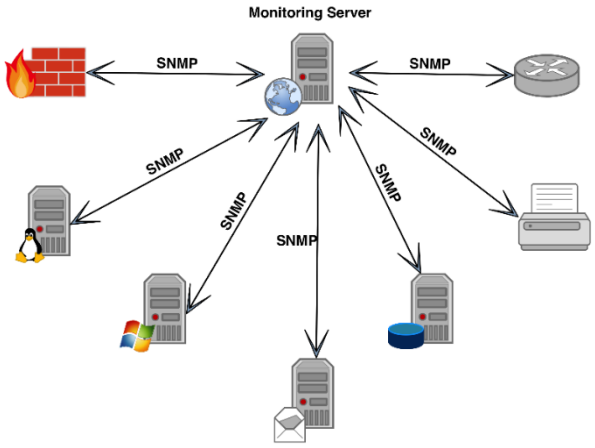
2.2.4 Protocolo SNMP

Protocolo Simples de Gerenciamento de redes (camada 2 do modelo ISO/OSI, em inglês - Simple Network Management Protocol - SNMP). Todo e qualquer dispositivo que possa e, esteja conectado a rede, possui a capacidade de se comunicar através do SNMP. Se trata de um protocolo padrão. Foi desenvolvido especialmente para facilitar o gerenciamento e monitoramento de rede. Permite que varias ou, uma máquina ligada a rede, possa ser definida como gerente (em inglês - manager) e, os dispositivos que enviam informações para o manager, são definidos como agentes (em inglês - agent).

Para o processamento das informações recebidas pelos agents, é comum o uso de uma árvore hierárquica, que é organizada pelo tipo de informação, chamada de base de informações de gerenciamento, ou MIB (Management Information Base). Nesta árvore hierárquica, são gravadas todas as informações que são necessárias para gerir cada dispositivo, usando todas as variáveis dos agents requeridas pelos dispositivos managers.

Os dispositivos agents, se diferenciam através de um identificador de objeto, ou OID (Object identifier). Os dispositivos manager reconhecem as informações fornecidas pelos agents para que possa ser realizado a consulta. A imagem a seguir (imagem 5), é um layout de exemplo, mostrando a comunicação entre os dispositivos, onde o “monitoring server” é o dispositivo gerenciador e, os demais sao os agentes.

Imagem 5 – Exemplo de comunicação entre agents e manager



Fonte: ozeki.hu

A imagem 5 ilustra a troca de informações entre agents e manager, onde, o servidor de monitoramento é o manager e, os demais equipamentos são os agents.

2.2.4 Contêiner

Com a conteinerização, é possível criar ambientes completamente isolados, sem a necessidade de criar uma máquina virtual. É possível isolar serviços e aplicações em contêineres, separando as aplicações e serviços, sem que tenha qualquer tipo de conflito com os demais. Existem diversos provedores de serviços de conteinerização, o que utilizamos para este estudo, foi o Docker.

2.2.5 Docker

A funcionalidade do Docker, consiste no Docker server, daemon, imagens, registros e por fim, os contêiners, seguimos para a explicação de cada um;

a) Docker server: responsável pelo armazenamento e ações relacionadas aos contêiners.
b) Docker daemon: recebe todos os comandos a partir da interface de linha de comando (ILC, em inglês - command-line interface - CLI), ou interface de programação de aplicação (em inglês – aplication programming interface - API). Possuí a capacidade de criar, iniciar, desligar e excluir contêiners.
c) Imagens: as imagens do Docker, são arquivos que contém toda a estrutura e conteúdo de sistemas operacionais. São a base de todos os contêiners, o Docker

server, utiliza AUFS, um sistema de arquivos que, possibilita a escrita de templates que possuem todas as configurações necessárias e desejadas para a construção do contêiner.

d) Registros: serve como repositório de imagens, permitindo ao administrador, construir, distribuir e salvar imagens com outros, através do site Docker Hub (hub.docker.com), onde é possível tanto depositar imagens, quanto obter, funciona como o Git Hub.
e) Contêiners: ambiente isolado de execução, possibilitado através de imagens do Docker, instanciados na camada de escrita da imagem, possibilitando isolar recursos

3 MATERIAIS E MÉTODO

3.1 Objetivo

Com o objetivo de mostrar como funciona o monitoramento da rede e a sua importância primordial para se tomar decisões ágeis em situações críticas e evitar futuros incidentes, criamos um pequeno ambiente, com alguns elementos utilizados familiares em empresas.

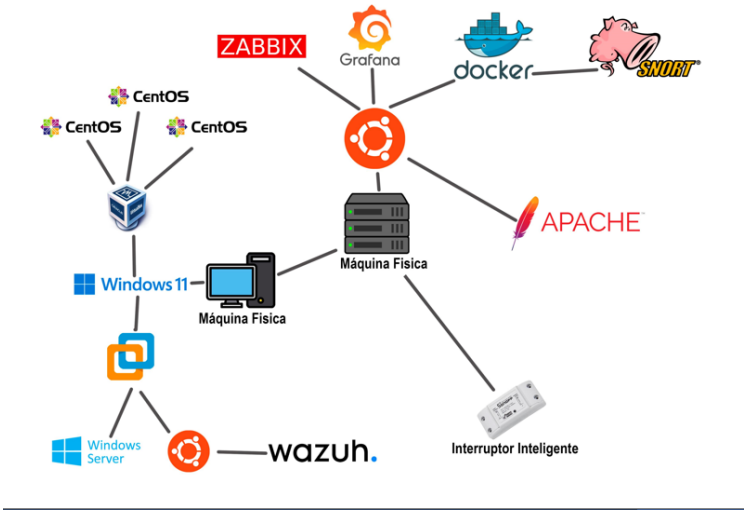
3.2 Recursos

Para cumprir com o objetivo de criar um ambiente de rede, utilizamos os seguintes recursos:

a) Para a virtualização, fizemos uso dos softwares VWware Player e Oracle Virtual Box. Virtualizando os sistemas operacionais; CentOS, Windows Server e Ubuntu Server.
b) Criação e gerenciamento de conteiners; Docker.
c) Softwares de monitoramento; Zabbix, Wazuh e Snort.
d) Dispositivos físicos; interruptor inteligente Sonoff, mini PC Evo e, um computador pessoal.
e) Software para personalização de gráficos; Grafana.
f) Servidores web; Apache.
g) Banco de dados; MySQL.

A imagem 6 retrata o layout do ambiente que criamos.

Imagem 6 – Layout do ambiente que criamos



Fonte: Os autores

No computador pessoal, com sistema operacional Windows 11, através do Virtual Box, foi virtualizado três máquinas com sistema CentOS e, utilizando Piranha para criar um Cluster. Através do VMware Player, virtualizamos o Ubuntu Server com o host do Wazuh e, uma máquina com Windows Server com o Active Directory (AD) instalado, porém, sem estar configurado.

No mini PC Evo, fizemos a instalação do host do Zabbix, Grafana, e o Docker, com o Docker, criamos um container compartilhando o diretório de logs, utilizando a imagem do ubuntu 22.04 LTS para hospedar o Snort.

3.2.1 Cluster

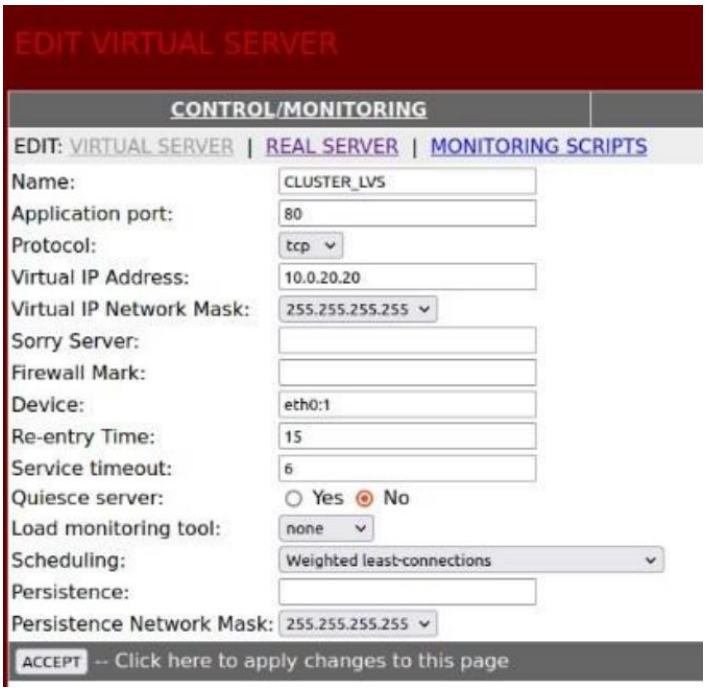
Através do computador pessoal realizamos a instalação do VirtualBox e, virtualizamos uma máquina com sistema operacional CentOS, com 4GB de memória RAM, 20GB de armazenamento e 2 VCPUs. Fizemos dois clones desta máquina, sendo a primeira com o nome LVS_Master, e as outras; LVS_Node1 e LVS_Node2. Determinamos os seguintes endereços IP para cada:

LVS_Master: 10.0.20.10/16
LVS_Node1: 10.0.20.11/16
LVS_Node2: 10.0.20.12/16

Criamos um endereço de rede virtual através do gerenciador do VirtualBox com o IP 10.0.20.20/24 sendo a sua máscara 255.255.255.255, para apenas se enxergar na rede. Atribuímos a cada uma das três máquinas virtuais, o IP 10.10.10.10/24 como novos adaptadores de rede para podermos configurar em um serviço de cluster que já vem nos repositórios do CentOS, o Piranha, conseguimos acessar tela de configuração dele através do navegador, colocando o IP da máquina juntamente a porta onde o serviço do Piranha está configurado para ser

acessado, no caso, a porta 3636. Como máquina principal, utilizamos a LVS_Master, então acessamos as configurações através do 10.0.20.10:3636

Imagem 7 - Configuração do servidor no Piranha



Fonte: Os autores

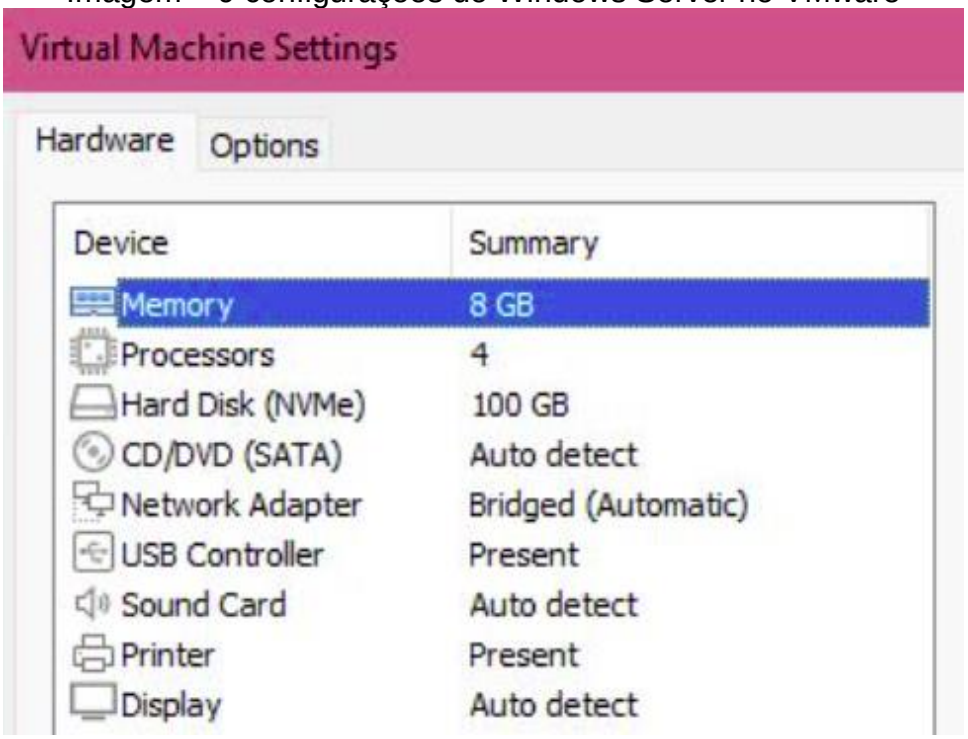
Imagem - 8 VirtualBox com as VMs; LVS_Master, LVS_Node1 e LVS_node2

Fonte: Os autores

3.2.2 Windows Server

A instalação do Windows server, serviu para complementar e ter mais variações de sistemas operacionais dentro da rede, pois não instalamos nenhum serviço no mesmo. Para instalação, usamos o computador pessoal e fizemos uso do VMware Player. As configurações da máquina virtual foram; 100GB armazenamento, 8GB de memória RAM e 4 VCPUs.

Imagem – 9 configurações do Windows Server no VMware



Fonte: Os autores

3.2.3 Interruptor inteligente

Para a configuração do dispositivo Sonoff, utilizamos o aplicativo para celular Smart Life, fornecido pela Volcano Technology limited. Através dele temos informações sobre o endereço IP e o controle de acesso de mídia (em inglês – Media Access Control - MAC). Para o dispositivo manter o endereço IP, foi necessário acessar remotamente o roteador e, acessar as configurações de host dinâmico (em inglês – Dynamic Host Configuration Protocol - DHCP), e através do endereço MAC do dispositivo Sonoff, dedicar um IP específico. Dessa forma, todas as vezes em que o dispositivo ficar fora da rede e voltar, será mantido o endereço IP. Dessa forma, para ser monitorado pelo Zabbix, configuramos o monitoramento via ping ICMP. Assim, nos mostrando a sua disponibilidade.

Imagem – 10 painel de configuração do aplicativo Smart Life

Fonte: Os autores

3.2.4 Wazuh

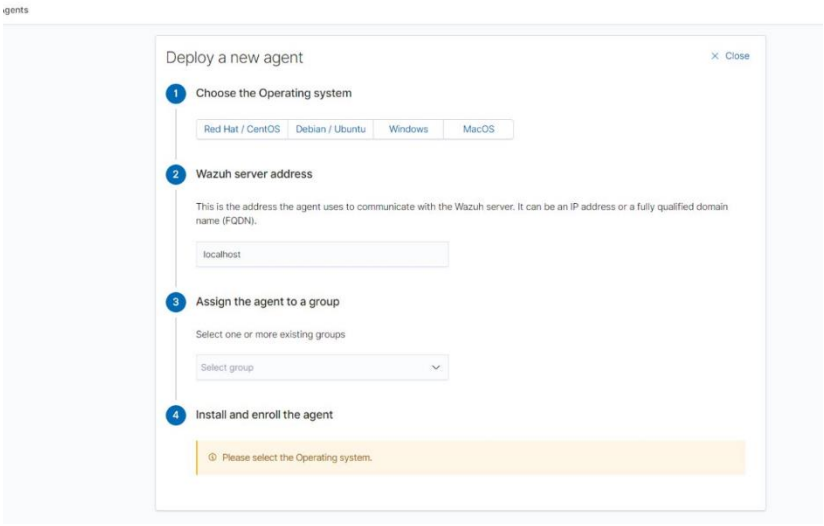
A instalação e configuração do Wazuh, foi feita seguindo os manuais disponibilizados no site oficial wazuh.com. O sistema operacional onde o gerenciador do Wazuh está instalado, é o Ubuntu 22.04 que está virtualizado pelo VMware Player, instalado no computador pessoal. Para a instalação, seguimos a documentação Quick Start do Wazuh, com o seguinte comando pelo terminal do linux; **curl -sO https://packages.wazuh.com/4.3/wazuh-install.sh && sudo bash ./wazuh-install.sh -a**. Após a instalação através deste comando, recebemos a senha de primeiro acesso e configuração. Para acessar o painel gerenciador do Wazuh, basta acessar através do navegador utilizando o IP da máquina onde fora feito o quick start.

Imagem – 11 tela inicial do Wazuh

Fonte: Os autores

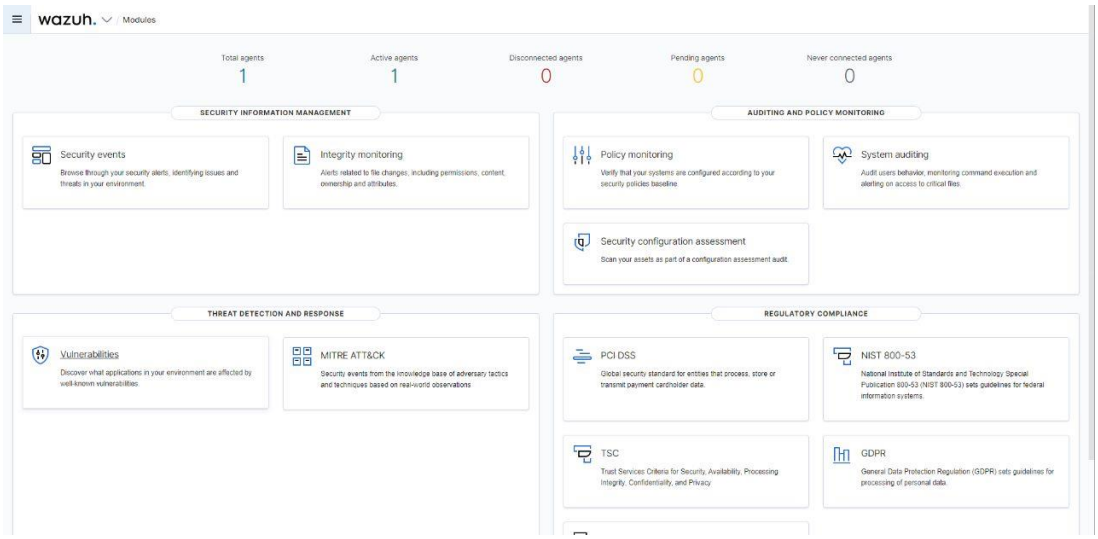
Para adicionarmos os dispositivos que serão monitorados através do Wazuh, precisamos adicionar em cada um deles o Wazuh agent, que é o responsável por enviar relatórios para o gerenciador, através do SNMP. É necessário seguir os passos da imagem 12. Assim, para cada situação, é gerado um código para ser executado via terminal, seja através do PowerShell (Windows) ou terminais linux.

Imagem – 12 tela de geração de códigos para instalação do agent wazuh



Fonte: Os autores

Imagem 13 – Painel do Wazuh onde podemos escolher qual informação queremos acessar



Fonte: Os autores

O Wazuh traz informações extremamente relevantes para segurança da rede, baseadas no Instituto Nacional de Padrões e Tecnologia (em inglês - National Institute of Standards and Technology - NIST). O Wazuh, indica o score de segurança da rede, apontando tudo que está correto e tudo que falta para atingir o padrão de segurança estipulado pelo NIST.

Imagem 14 – Score do NIST sobre o computador pessoal

Fonte: Os autores

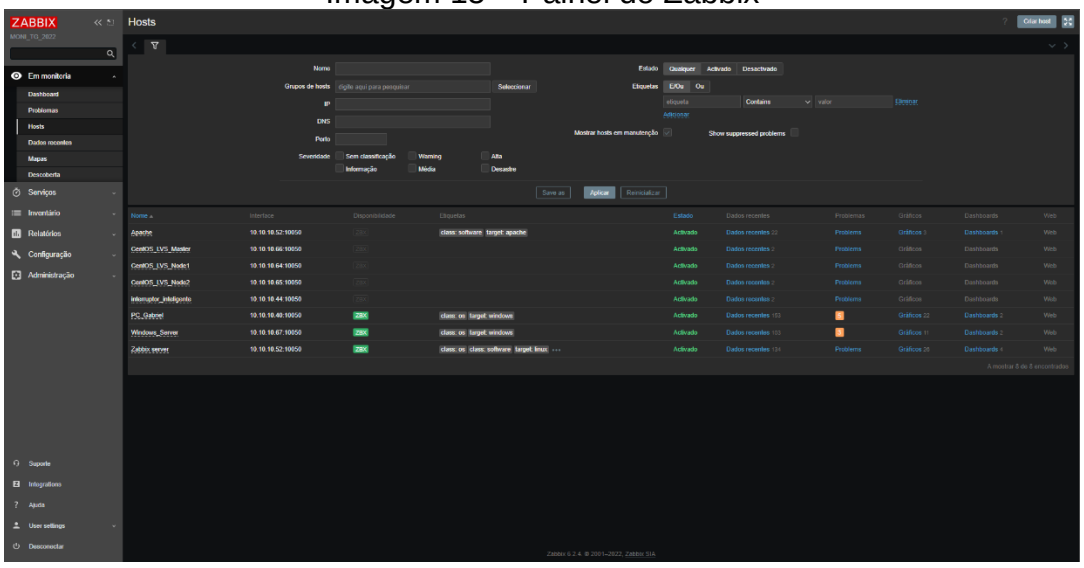
3.2.5 Zabbix

A instalação do host Zabbix 6.0, foi realizada no sistema operacional Ubuntu Server 22.04, que por sua vez, está instalado no mini PC Evo. A instalação e configuração do Zabbix, foi realizada seguindo o manual disponibilizado no site **zabbix.com**. Após o download do Zabbix server, foi necessário a criação de um banco de dados para armazenar as credenciais de administrador, então criamos no MariaDB.

Para definir as máquinas clientes, fizemos o download do zabbix agent através do site oficial. Para cada sistema operacional, existe a forma correta de baixar e instalar o zabbix agent. Para acessar as configurações, preferências e administrar o zabbix, basta acessar o IP da máquina hospedeira do software, através do navegador.

A forma que o Zabbix consegue monitorar os dispositivos clientes ligados a ele é bem simples, pode ser através de SNMP (fazendo uso do agent), ou através do Ping (ICMP). Através do SNMP, podemos obter muitas informações sobre muitos recursos, inclusive serviços e micro serviços. Já através do Ping, conseguimos apenas informações de conexão, por exemplo latência e disponibilidade.

Imagem 15 – Painel do Zabbix



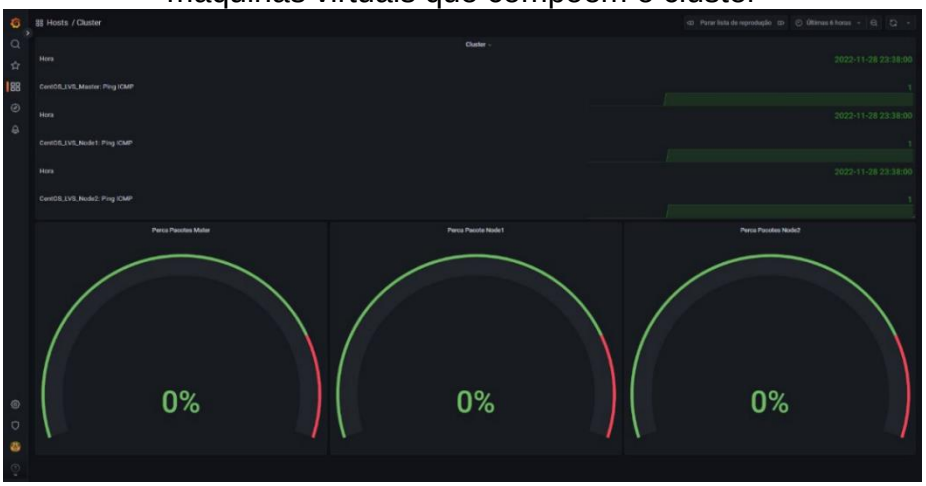
Fonte: Os autores

3.2.5 Grafana

Para deixar o Zabbix ainda mais completo, instalamos e configuramos o Grafana para integrá-lo ao Zabbix, seguindo o manual da documentação disponibilizada em

grafana.com. E assim, criando templates personalizados para melhor visão dos dispositivos monitorados.

Imagem 16 – Dashboard do Grafana, monitorando a disponibilidade das três máquinas virtuais que compõem o cluster

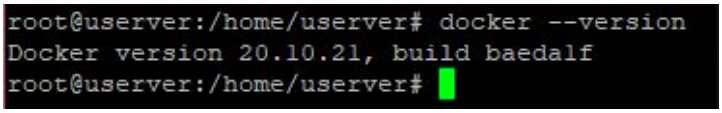


Fonte: Os autores

3.2.5 Docker

Fazendo acesso ao mini PC Evo, realizamos a instalação do Docker, através do repositório do Ubuntu apenas utilizando o comando; **sudo apt-get install docker.io**. Após isso, o Docker já ficou disponível para começarmos a utilizá-lo.

Imagem 17 – Comando docker **–verison** mostrando a versão do Docker que utilizamos



Fonte: Os autores

3.2.5.1 Snort

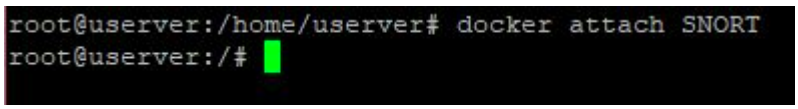
A instalação do Snort, foi realizada em um conteiner no Docker, que por usa vez, está instalado no mini PC Evo. O conteiner para a utilização do Snort, foi criado com o comando **docker run -it -d --net=host --name SNORT -v /home/victor/snort-logs:/var/log/snort ubuntu:latest** – Onde o paramentro **–net=host,** nos possíbilitou definir as configurações de rede do conteiner, exatamente igual à do host. Dessa forma, utilizando o mesmo endereço IP, funcionando como se estivéssemos na máquina host, só que sem modificar nenhum arquivo da máquina física. Para definirmos o nome do conteiner, utilizamos o parâmetro –**name SNORT**, definido assim, o nome do conteiner para SNORT. O paramentro –**v /home/victor/snort-logs:/var/log/snort**, fez com o que os arquivos de logs gerados pelo snort, cujos mesmos são criados no diretório **/var/log/snort**, sejam compartilhados com o diretório **/home/victor/snort-logs** localizado na máquina física. Por fim, o parâmetro **ubuntu:latest** nos permitiu utilizar a imagem mais atualizada do sistema operacional Ubuntu (versão 22.10). O comando **docker ps**, nos permite ter uma visão dos contêineres ativos no momento.

Imagem 18 – Comando **docker ps** mostrando o conteiner criado para o Snort, ativo

Fonte: Os autores

Para acessar o conteiner criado para hospedar o Snort, usamos o comando **docker attach SNORT**

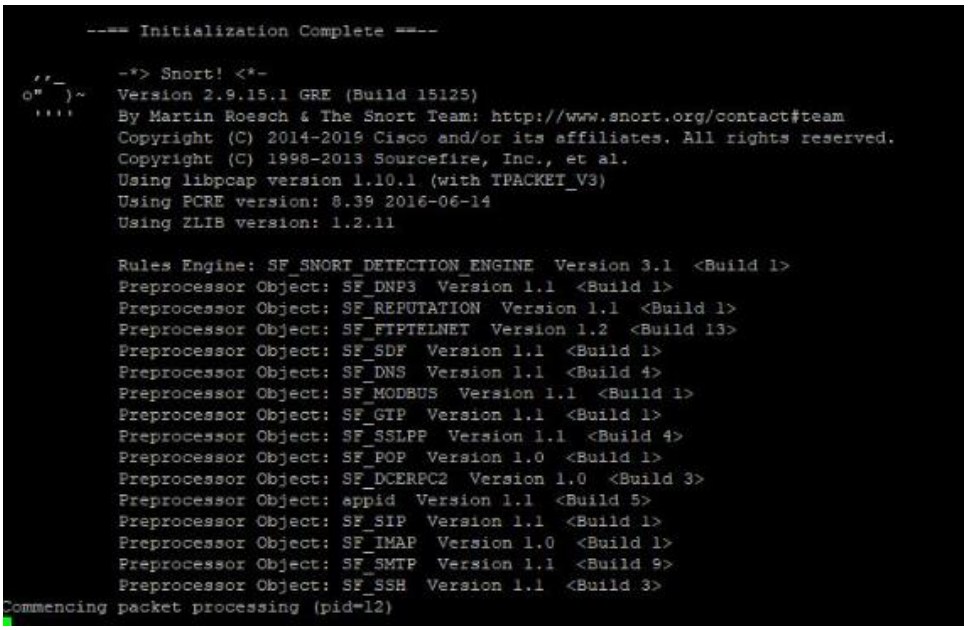
Imagem 19 – Acessando o conteiner SNORT através do comando **docker attach SNORT**



Fonte: Os autores

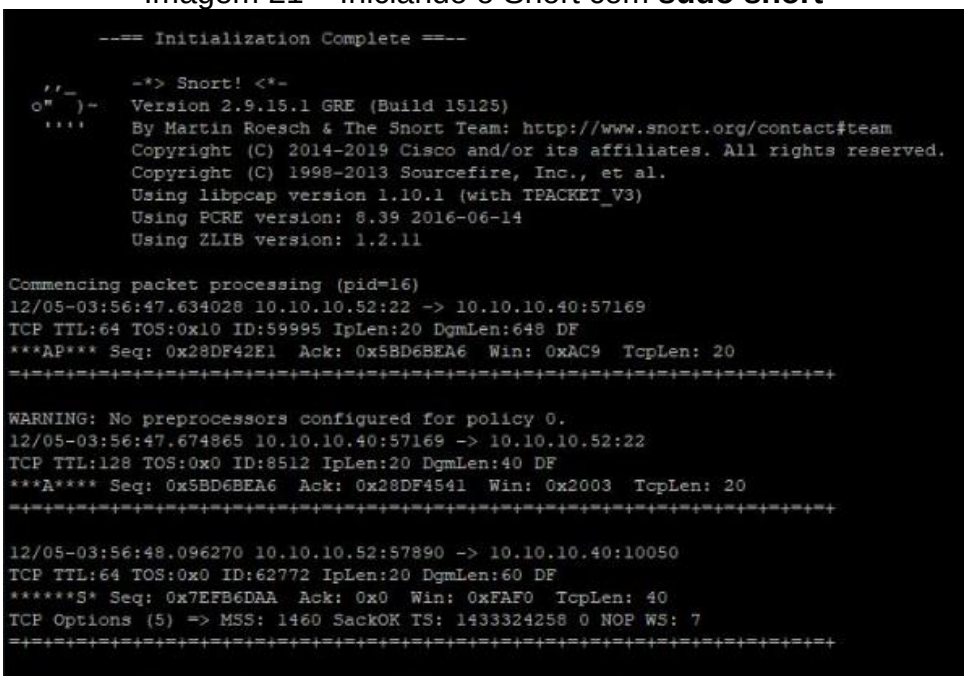
Dentro do conteiner, fizemos a instalação do Snort com o comando **apt-get install snort**. Com esta instalação simples do Snort, ele automaticamente fica configurado para monitorar o IP no qual a máquina onde foi instalado, possuí e, como definimos o conteiner para obter a mesma configuração do host, ao ativarmos o serviço do Snort, ele faz o monitoramento do tráfego realizado no host e no conteiner. Iniciamos os serviços do Snort com o comando snort -**d -l /var/log/snort/ -A console -c /etc/snort/snort.conf**. É possível iniciar o snort apenas com o comando **sudo snort**, o que faz com que podemos ver o monitoramento em tempo real, porém, os parâmetros que utilizamos para iniciar os serviços, definem que o Snort atue em segundo plano e apenas faça a geração de arquivos de logs, para o diretório **/var/log/snort/**.

Imagem 20 – Iniciando o serviço do Snort com **d -l /var/log/snort/ -A console -c /etc/snort/snort.conf**



Fonte: Os autores

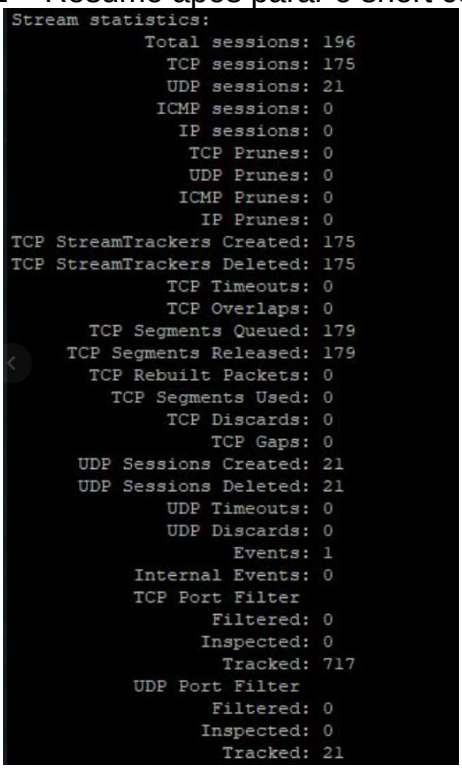
Imagem 21 – Iniciando o Snort com **sudo snort**



Fonte: Os autores

Ao pressionar **ctrl + c** é possível parar o funcionamento do software. Desta forma, o Snort traz um resumo do que foi detectado na rede, independente do modo que foi iniciado.

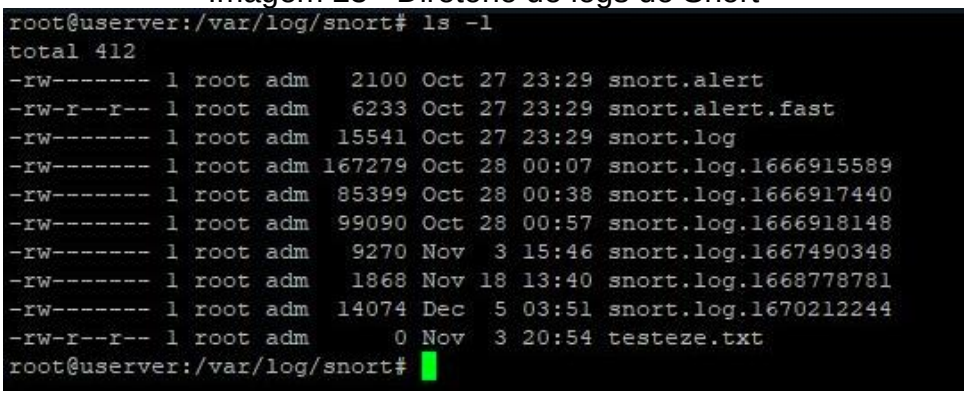
Imagem 22 – Resumo após parar o snort com **ctrl + c**



Fonte: Os autores

Dentro do diretório /var/log/snort, é possível ter acesso a todos os logs gerados pelo Snort todas as vezes em que foi iniciado e parado.

Imagem 23 - Diretório de logs do Snort



Fonte: Os autores

4 CONCLUSÃO

Com todos os recursos utilizados, tanto para a criação do ambiente de rede, quanto os utilizados para monitoramento, foi possível ter controle sobre tudo o que definimos para ser monitorado, desde a disponibilidade de conexão até desempenho. Com isso podemos afirmar que, com um ambiente de monitoramento bem-organizado, permite a tomada de ações ágeis de formas preventivas e corretivas, evitando incidentes e desastres como a indisponibilidade de sistemas, parada de operação, softwares mal-intencionados, tentativas de ataques e saúde do hardware.

5 REFERENCIAS

ANDRADE, M, 2021. Disponível em: marcoandrade.com.br/. Acesso em: 30 set 2021.

GALVANI, J 02 fev 2022. Disponível em: www.folhavitória.com.br. Acesso em: 01 abr 2022

GUGELMIN, F 05 nov 2020. Disponível em: canaltech.com.br. Acesso em 04 abr 2022

GRAFANA, 2021. Disponível em: grafana.com/grafana/dashboards.
Acesso em: 30 set 2021

GOGONI, R. Disponível em: tecnoblog.net. Acesso em: 11 nov 2019

PRUDENCIATO, R 23 ago 2017. Disponível em: linuxsemfronteiras.com.br. Acesso em: 30 set 2021

PINTO, P 12 dez 2020. Disponível em: pplware.sapo.pt. Acesso em 30 set 2021

RODRIGUES, R 20 nov 2018. Disponível em: www.kaspersky.com.br. Acesso em 26 abr 2022