PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS ESCOLA POLITÉCNICA E DE ARTES GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

CONEXÃO VIA SSH PARA UM FIREWALL FÍSICO ADMINISTRADO REMOTAMENTE

NATAN BORGES NUNES

GOIÂNIA

2025

NATAN BORGES NUNES

CONEXÃO VIA SSH PARA UM FIREWALL FÍSICO ADMINISTRADO REMOTAMENTE

Trabalho de Conclusão de Curso apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Computação.

Orientador(a): Prof. M.E.E. Marcelo Antonio Adad de Araújo.

GOIÂNIA

2025

NATAN BORGES NUNES

CONEXÃO VIA SSH PARA UM FIREWALL FÍSICO ADMINISTRADO REMOTAMENTE

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel em Engenharia de Computação, e aprovado em sua forma final pela Escola Politécnica e de Artes da Pontifícia Universidade Católica de Goiás, em _____/_____/_____.

_____

Prof. Luiz Álvaro de Oliveira Junior

Coordenador de Trabalho de Conclusão de Curso

Banca Examinadora:

_____

Orientador(a): Prof. M.E.E. Marcelo Antonio Adad Araújo

_____

Prof. M.E.E. Carlos Alexandre Ferreira de Lima

_____

Prof. Dr. Nilson Cardoso Amaral

GOIÂNIA

2025

RESUMO

O presente trabalho é um estudo sobre como explorar a conexão SSH (*Secure Shell*) para uso e manutenção remota de um *firewall*, e usa como base um Raspberry Pi 4. Os *firewalls* são soluções de segurança de rede que atuam como um filtro, garantindo maior proteção para seu ambiente. Eles permitem a restrição de acesso a sites ou sistemas conforme as exigências da empresa, impedindo que usuários acessem conteúdos inapropriados, não relacionados ao trabalho ou maliciosos. Para configurar um *firewall* de forma eficiente, é essencial compreender suas regras e as necessidades específicas do ambiente, garantindo melhor desempenho e prevenindo possíveis ataques. Com o SSH, é possível acessar o *firewall* remotamente para aplicar regras, monitorar logs, atualizar *firmware* e diagnosticar problemas sem necessidade de contato físico. Outra funcionalidade útil é a criação de túneis SSH, que permite gerenciar outros serviços de rede de forma criptografada. Para garantir ainda mais segurança, o monitoramento e a auditoria das conexões SSH são essenciais, possibilitando o rastreamento de atividades e a identificação de tentativas de acessos indevidos. Dessa forma, o SSH se torna uma ferramenta essencial para a administração remota de *firewalls* físicos, proporcionando maior segurança e eficiência na gestão da rede.

Palavras-chave: *firewall*, SSH (*Secure Shell*), Raspberry Pi, segurança, ataques.

ABSTRACT

This paper is a study on how to leverage SSH (Secure Shell) for the remote use and maintenance of a firewall, using a Raspberry Pi 4 as a foundation. Firewalls are network security solutions that act as filters, ensuring greater protection for the environment. They allow access restrictions to websites or systems according to company requirements, preventing users from accessing inappropriate, non-work-related, or malicious content. To configure a firewall efficiently, it is essential to understand its rules and the specific needs of the environment, ensuring better performance and preventing potential attacks. With SSH, it is possible to remotely access the firewall to apply rules, monitor logs, update firmware, and diagnose issues without the need for physical contact. Another useful feature is SSH tunneling, which allows the management of other network services in an encrypted manner. To further enhance security, monitoring and auditing SSH connections are essential, enabling activity tracking and identifying unauthorized access attempts. Thus, SSH becomes an essential tool for the remote administration of physical firewalls, providing greater security and efficiency in network management.

Keywords: firewall, SSH (Secure Shell), Raspberry Pi, security, attacks.

LISTAS DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

2FA - *Two-Factor Authentication* (Autenticação em Dois Fatores)

AES - *Advanced Encryption Standard*

Dr. - Doutor

DPI - *Deep Packet Inspection* (Inspeção Profunda de Pacotes)

ECDSA - *Elliptic Curve Digital Signature Algorithm*

IDS - *Intrusion Detection System* (Sistema de Detecção de Intrusões)

IPS - *Intrusion Prevention System* (Sistema de Prevenção de Intrusões)

Me. - Mestre

M.E.E. - Mestre em Engenharia Elétrica

MFA - *Multi-Factor Authentication* (Autenticação Multifator)

MitM - *Man-in-the-Middle* (ataque de interceptação)

NAT - *Network Address Translation* (Tradução de Endereços de Rede)

NGFW - *Next-Generation Firewall* (Firewall de Próxima Geração)

OTP - *One-Time Password* (Senha de Uso Único)

Prof. - Professor

RDP - *Remote Desktop Protocol* (Protocolo de Área de Trabalho Remota)

RSA - *Rivest-Shamir-Adleman* (algoritmo de criptografia)

SIEM - *Security Information and Event Management* (Gestão de Eventos e Informações de Segurança)

SSH - *Secure Shell*

VPN - *Virtual Private Network* (Rede Privada Virtual)

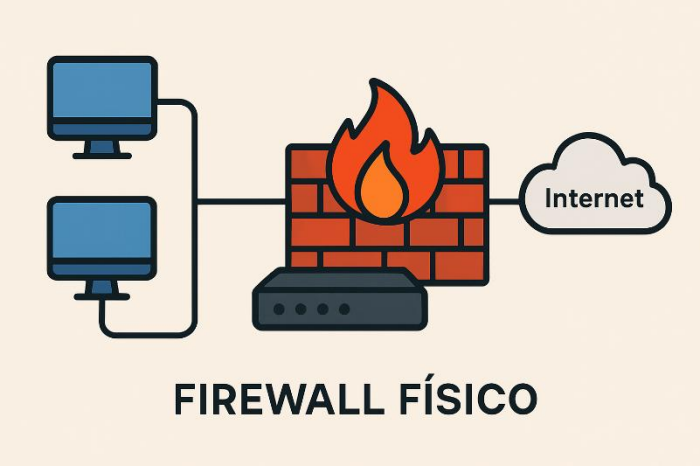
SUMÁRIO

Capítulo 1

1.1 Introdução

A segurança em redes de computadores é uma preocupação crescente diante da sofisticação dos ataques cibernéticos e da constante exposição de sistemas interconectados. Nesse cenário, os *firewalls* surgem como elementos centrais no controle de tráfego, como apresentado na figura 1, atuando como uma barreira entre redes confiáveis e redes externas potencialmente perigosas (STALLINGS, 2020). Em especial, a administração remota desses dispositivos exige métodos seguros de acesso, sendo o protocolo SSH (*Secure Shell*) uma das soluções mais eficazes para garantir confidencialidade e integridade na comunicação entre o administrador e o *firewall* físico.

Figura 1



Fonte: Autoria Própria

O protocolo SSH, segundo Barrett e Silverman (2001), utiliza criptografia assimétrica e autenticação por chave pública para impedir interceptações e ataques *man-in-the-middle*, oferecendo uma linha de comando protegida para administração remota. Esse tipo de conexão é especialmente útil em ambientes onde o acesso físico ao *firewall* é limitado ou inviável, como em centros de dados remotos ou infraestruturas distribuídas.

Ao aplicar essa técnica, é essencial que boas práticas de segurança sejam observadas, como o uso de autenticação multifator (MFA), limitação de IPs autorizados e o monitoramento constante de acessos. Como reforçam Anderson e Moore (2006), a segurança efetiva é

alcançada não apenas pela implementação de ferramentas, também pela forma como elas são configuradas e gerenciadas.

Dessa forma, a conexão via SSH para um *firewall* físico administrado remotamente representa uma solução prática e robusta frente aos desafios atuais de segurança em redes corporativas, equilibrando eficiência operacional com requisitos críticos de proteção da informação.

1.2 Objetivo

A proposta deste trabalho é analisar e demonstrar a viabilidade técnica e os benefícios da utilização do protocolo SSH na administração remota de um *firewall* físico. Diante da crescente demanda por acessos seguros a dispositivos de rede localizados em ambientes críticos ou de difícil acesso, torna-se essencial o emprego de ferramentas que assegurem confidencialidade, integridade e autenticidade dos dados trafegados. Conforme destacam Tanenbaum e Wetherall (2013), a proteção do canal de comunicação entre o administrador e o dispositivo é um dos pilares da segurança em redes. Nesse contexto, este estudo visa aplicar esse conhecimento ao caso específico de *firewalls* físicos, explorando práticas seguras de configuração, autenticação e monitoramento remoto.

1.3 Objetivo Geral

O objetivo geral deste projeto é implementar e validar uma conexão remota segura via SSH entre uma estação de administração e um *firewall* físico, promovendo uma abordagem prática de gerenciamento à distância com foco em segurança da informação. Essa implementação considerou aspectos técnicos essenciais, como a criação de pares de chaves, restrições de acesso por IP, uso de autenticação por chave pública, e definição de políticas de segurança. Segundo Stallings (2020), a adoção de conexões criptografadas para dispositivos de rede é uma exigência básica em qualquer arquitetura de segurança moderna.

1.4 Objetivos Específicos

- Estudar os fundamentos do protocolo SSH e suas vantagens sobre métodos legados como Telnet;
- Levantar os requisitos técnicos para a configuração de SSH em um *firewall* físico;
- Aplicar boas práticas de segurança, como autenticação por chave e MFA;

- Demonstrar, por meio de simulação ou ambiente real, a comunicação segura entre cliente e *firewall* via SSH;
- Avaliar riscos e propor soluções para possíveis vulnerabilidades na administração remota.

Estes objetivos seguem a abordagem recomendada por Anderson e Moore (2006), que enfatizam a necessidade de segurança contínua e integrada em sistemas críticos de rede.

1.5 Justificativa

A escolha do tema se justifica pela necessidade crescente de realizar a administração remota de dispositivos de rede com segurança e eficiência, especialmente em um cenário onde a descentralização de infraestruturas é cada vez mais comum. O protocolo SSH destaca-se como uma alternativa confiável, substituindo métodos obsoletos e vulneráveis como Telnet. Além disso, *firewalls* físicos continuam sendo amplamente utilizados em ambientes empresariais para a segmentação e proteção de redes locais, tornando relevante o domínio de técnicas que possibilitem seu gerenciamento remoto seguro. Como pontua Whitman e Mattord (2017), a segurança da informação deve estar presente em todos os níveis da operação de TI, incluindo a administração de dispositivos. Portanto, este trabalho visa não apenas contribuir com o desenvolvimento técnico, mas também com a adoção de práticas seguras no cotidiano da administração de redes.

1.6 Procedimentos Metodológicos

A abordagem metodológica deste trabalho combina pesquisa bibliográfica e experimental. A primeira etapa consiste na revisão da literatura especializada sobre o protocolo SSH, segurança de redes e administração de *firewalls*, utilizando livros, artigos acadêmicos e documentos técnicos de referência na área.

1.6.1 Pesquisa Bibliográfica

A pesquisa bibliográfica serviu de base teórica para o desenvolvimento do projeto, fundamentando conceitos como criptografia, autenticação, acesso remoto e protocolos de comunicação. Obras de autores como (Tanenbaum e Wetherall (2013), Stallings (2020) e Barrett e Silverman (2001)) foram utilizadas como referência para compreender os princípios e as boas práticas de segurança envolvidas na utilização do SSH.

1.6.2 Pesquisa Experimental

A fase experimental consiste na configuração prática de um ambiente de teste contendo um *firewall* físico (como o pfSense ou outro dispositivo equivalente) e uma estação cliente com acesso via SSH. Foram configuradas chaves de autenticação, controle de acesso por IP, logging de eventos e testes de conexão para validação da segurança e funcionalidade do sistema.

Capítulo 2

2.1 Fundamentos do SSH (*Secure Shell*)

O SSH (*Secure Shell*) é um protocolo amplamente utilizado para estabelecer conexões seguras em redes, possibilitando o acesso remoto a dispositivos de forma criptografada. Diferente de métodos mais antigos, como Telnet, o SSH protege a transmissão de dados contra interceptações, garantindo maior segurança ao tráfego de informações sensíveis (STALLINGS, 2017).

2.1.1 Como o SSH Funciona

O SSH opera através da arquitetura cliente-servidor, onde um usuário (cliente) se conecta a um dispositivo remoto (servidor) para executar comandos e gerenciar arquivos de maneira segura. A comunicação ocorre por meio da porta 22, utilizando protocolos criptográficos que asseguram a integridade e a confidencialidade dos dados (TANENBAUM & WETHERALL, 2019).

2.1.2 Protocolos de Criptografia Utilizados

Para garantir a segurança da conexão, o SSH emprega algoritmos de criptografia como AES (*Advanced Encryption Standard*), RSA (*Rivest-Shamir-Adleman*) e ECDSA (*Elliptic Curve Digital Signature Algorithm*). Esses métodos protegem contra-ataques de interceptação, impedindo que invasores acessem informações transmitidas entre cliente e servidor (KUROSE & ROSS, 2021).

2.1.3 Modos de Autenticação

A autenticação no SSH pode ser feita de diferentes formas, dependendo do nível de segurança desejado:

- Senha Tradicional: método mais simples, mas menos seguro, pois pode ser alvo de ataques de força bruta;
- Chave Pública e Privada: envolve a criação de um par de chaves criptográficas, em que a chave pública é armazenada no servidor e a privada no cliente, proporcionando um acesso mais seguro (FOROUZAN, 2018);

- Autenticação em Dois Fatores (2FA): adiciona uma camada extra de proteção ao exigir um código adicional gerado por um dispositivo confiável.

2.1.4 Segurança e Vulnerabilidades

Embora seja uma ferramenta altamente segura, o SSH pode ser alvo de ataques caso não seja devidamente configurado. Algumas vulnerabilidades comuns incluem:

- Ataques de Força Bruta: tentativas sucessivas de login utilizando combinações de senhas;

Exploração de Chaves Fracas: o uso de chaves criptográficas de baixo nível pode facilitar invasões;

- Ataques *Man-in-the-Middle* (MM): ocorre quando um invasor intercepta a comunicação entre cliente e servidor, caso não haja uma verificação adequada das chaves públicas (STALLINGS, 2017).

Para mitigar esses riscos, é recomendável utilizar autenticação por chaves, restringir acessos por IP e monitorar continuamente os logs de conexões SSH. Dessa forma, a utilização do SSH se torna uma solução essencial para acessos remotos seguros e administração de redes sem comprometer a integridade dos dados.

2.2 Fundamentos de *Firewalls*

Os *firewalls* desempenham um papel fundamental na segurança de redes, atuando como barreiras de proteção entre dispositivos internos e ameaças externas. Sendo monitorado e filtrado o tráfego de entrada e saída com base em regras predefinidas, garantindo que apenas comunicações autorizadas sejam permitidas (STALLINGS, 2017). Essa funcionalidade é essencial para prevenir acessos não autorizados, ataques cibernéticos e a propagação de *malware* dentro de uma rede corporativa ou residencial.

2.3 O *Firewall* e sua Importância na Segurança de Redes

Um *firewall* é um sistema de segurança que pode ser implementado tanto em *software* quanto em *hardware*. Seu principal objetivo é controlar o fluxo de informações entre redes diferentes, bloqueando pacotes de dados suspeitos ou potencialmente perigosos. Segundo Kurose & Ross (2021), os *firewalls* funcionam como “porteiros digitais”, analisando cada

pacote de dados antes de permitir sua entrada ou saída da rede. Esse processo protege sistemas contra invasões e impede a comunicação com servidores maliciosos.

2.3.1 Tipos de *Firewalls*

Os *firewalls* podem ser classificados em diferentes categorias, dependendo de como realizam o controle do tráfego de rede:

- *Firewall* de Filtragem de Pacotes: Analisa pacotes individuais de dados e os permite ou bloqueia com base em critérios como endereço IP, número da porta e protocolo de comunicação. É um dos métodos mais rápidos, mas não verifica o conteúdo das conexões (TANENBAUM & WETHERALL, 2019);
- *Firewall Stateful* (com estado): Esse tipo mantém um histórico das conexões estabelecidas, permitindo decisões mais inteligentes sobre o tráfego, pois reconhece pacotes pertencentes a sessões legítimas já estabelecidas. Essa abordagem melhora a segurança em relação ao *firewall* de filtragem de pacotes tradicional (FOROUZAN, 2018);
- *Firewall Proxy*: Atua como um intermediário entre os usuários e a internet. Ao invés de conectar diretamente os dispositivos à web, o proxy recebe as solicitações e as retransmite, ocultando informações da rede interna e filtrando conteúdos potencialmente perigosos (KUROSE & ROSS, 2021);
- *Firewall* de Próxima Geração (NGFW – *Next-Generation Firewall*): Além das funções tradicionais, esse tipo incorpora recursos como inspeção profunda de pacotes (DPI – *Deep Packet Inspection*), detecção de intrusos (IDS/IPS) e análise comportamental para identificar ameaças sofisticadas (STALLINGS, 2017);

2.3.2 Principais Regras e Políticas de *Firewall*

A eficácia de um *firewall* depende da configuração de regras precisas e de uma política de segurança bem definida. As regras podem ser configuradas para permitir ou negar acessos com base em diferentes critérios, como:

- Endereço IP de origem e destino;
- Protocolo de comunicação (TCP, UDP, ICMP);
- Portas utilizadas (HTTP - 80, HTTPS - 443, SSH - 22 etc.);

- Tempo de conexão permitido.

Uma política de *firewall* bem estruturada deve seguir o princípio do menor privilégio, ou seja, permitir apenas o tráfego essencial para o funcionamento da rede e bloquear todo o restante por padrão (FOROUZAN, 2018).

2.3.3 Exemplos de *Firewalls* Físicos Utilizados em Redes Corporativas

Em ambientes empresariais, os *firewalls* físicos são amplamente utilizados para garantir proteção avançada. Alguns dos principais fabricantes incluem:

- Cisco ASA (*Adaptive Security Appliance*): Amplamente utilizado em empresas de grande porte, possui recursos avançados de filtragem e segurança integrada;
- *pfSense*: Solução baseada em software livre, pode ser implementada em hardware dedicado para oferecer controle detalhado do tráfego de rede;
- *Fortinet FortiGate*: *Firewall* robusto com funcionalidades NGFW, incluindo proteção contra ameaças em tempo real.

O uso de *firewalls* físicos garante maior controle sobre a rede e melhora a segurança contra-ataques externos, tornando-se uma peça essencial na infraestrutura de TI de qualquer organização (KUROSE & ROSS, 2021).

Os *firewalls* são elementos indispensáveis para a segurança digital, atuando como a primeira linha de defesa contra invasões e acessos não autorizados. A escolha e configuração adequada de um *firewall*, seja ele baseado em *software* ou *hardware*, é essencial para garantir a proteção das informações e a integridade da rede. Para um funcionamento eficiente, é fundamental aplicar regras precisas, realizar auditorias frequentes e combinar o *firewall* com outras camadas de segurança, como criptografia e autenticação reforçada.

2.4 Configuração de um *Firewall* em um Raspberry Pi 4

A segurança de redes exige ferramentas eficientes para proteger dispositivos contra acessos não autorizados e tráfego malicioso. O Raspberry Pi 4, um minicomputador de baixo custo e alta flexibilidade, pode ser configurado como um *firewall*, oferecendo controle sobre conexões e filtragem de pacotes para garantir maior proteção (Kurose & Ross, 2021). Essa solução é especialmente útil em ambientes domésticos e pequenas empresas, onde a segurança da rede precisa ser reforçada sem investimentos em equipamentos de alto custo.

2.4.1 Escolha do Sistema Operacional

Para transformar o Raspberry Pi 4 em um *firewall* funcional, o primeiro passo é selecionar um sistema operacional adequado. Existem diversas opções disponíveis, cada uma com características específicas para segurança e filtragem de tráfego:

- Raspberry Pi OS com *IPTables*: O próprio sistema operacional oficial do Raspberry Pi suporta o uso do *IPTables*, um *firewall* baseado em regras para filtragem e controle de pacotes (STALLINGS, 2017);
- *OpenWRT*: Uma distribuição Linux otimizada para roteadores e *firewalls*, oferecendo um ambiente leve e altamente configurável;
- *pfSense*: Um dos *firewalls* mais populares, baseado em FreeBSD, amplamente utilizado em empresas para segurança avançada;
- *OPNsense*: Derivado do *pfSense*, oferece um painel gráfico intuitivo para gerenciamento simplificado das regras de *firewall*.

A escolha do sistema operacional depende do nível de controle e da interface desejada. Soluções como *pfSense* e *OPNsense* oferecem um ambiente gráfico mais amigável, enquanto *IPTables* e *OpenWRT* exigem configuração via linha de comando.

2.4.2 Instalação e Configuração do *Firewall*

Após definir o sistema operacional, é necessário instalar e configurar o *firewall* no Raspberry Pi 4. De acordo com Tanenbaum & Wetherall (2019), *firewalls* eficientes devem ser configurados seguindo políticas de segurança bem definidas para garantir que apenas tráfego autorizado seja permitido.

Os principais passos para configurar um *firewall* no Raspberry Pi incluem:

- Instalação do *software firewall*: Caso o sistema escolhido não tenha um *firewall* integrado, pode-se instalar o *IPTables* ou o UFW (*Uncomplicated Firewall*) para simplificar a configuração;
- Definição das regras de tráfego: É essencial criar regras para permitir ou bloquear acessos com base nos endereços IP, protocolos e portas utilizadas;
- Habilitação do NAT (*Network Address Translation*): O NAT permite que múltiplos dispositivos compartilhem um único endereço IP público, garantindo anonimato e segurança;

- Monitoramento dos logs de tráfego: Verificar continuamente os registros de acessos pode ajudar na detecção de tentativas de invasão ou tráfego suspeito.

Essas etapas garantem que o Raspberry Pi funcione como um *firewall* eficaz, protegendo a rede contra acessos indevidos.

2.4.3 Definição de Regras de Filtragem e Controle de Tráfego

Uma configuração eficiente de *firewall* exige a definição de regras precisas para filtrar e gerenciar o tráfego de rede. Segundo Forouzan (2018), *firewalls* devem operar seguindo o princípio do menor privilégio, ou seja, bloqueando todo o tráfego por padrão e liberando apenas o necessário.

As regras de filtragem podem incluir:

- Bloqueio de portas não utilizadas: Reduz a superfície de ataque ao desativar serviços desnecessários;
- Restrição de acessos por IP: Permite apenas conexões vindas de endereços IP confiáveis;
- Filtragem por protocolo: Define quais protocolos de comunicação (TCP, UDP, ICMP) são permitidos;
- Monitoramento de tráfego em tempo real: Identifica padrões de atividade suspeita para prevenir ataques.

Essas práticas aumentam a segurança da rede e evitam que dispositivos não autorizados acessem informações sensíveis.

2.4.4 Monitoramento e Registro de Logs de Segurança

O monitoramento contínuo é essencial para garantir que o *firewall* esteja operando corretamente e detectando possíveis ameaças. Segundo Stallings (2017), a análise de logs permite identificar atividades suspeitas e tomar medidas preventivas antes que ataques se concretizem.

Os logs podem ser armazenados localmente ou enviados para servidores externos para análise posterior. Ferramentas como *Fail2Ban* ajudam a bloquear automaticamente endereços IP que realizam tentativas repetidas de acesso indevido, protegendo o sistema de contra-ataques de força bruta.

A configuração de um *firewall* no Raspberry Pi 4 é uma alternativa acessível e eficiente que reforça a segurança da rede. Escolher um sistema operacional adequado, definir regras de filtragem precisas e monitorar o tráfego regularmente são etapas essenciais para garantir a proteção contra ameaças. Com o devido planejamento, o Raspberry Pi pode se tornar um *firewall* robusto para ambientes domésticos e empresariais, oferecendo controle total sobre a segurança da conexão, já observado inclusive em trabalhos anteriores, do orientador do presente.

2.5 Configuração e Uso do SSH para Acesso Seguro ao *Firewall*

A administração remota de um *firewall* exige uma conexão segura para evitar que dados confidenciais sejam interceptados por atacantes. O *Secure Shell* (SSH) é uma das tecnologias mais eficazes para esse fim, pois fornece um canal criptografado para a comunicação entre cliente e servidor. Sua implementação permite o gerenciamento remoto de *firewalls* físicos, garantindo segurança e eficiência na administração da rede (STALLINGS, 2017).

2.5.1 O SSH e sua Importância na Segurança de Redes

O SSH (*Secure Shell*) é um protocolo de comunicação segura que permite acessar dispositivos remotos via linha de comando, protegendo os dados por meio de criptografia. Ele substitui protocolos inseguros como Telnet e FTP, que transmitem informações em texto plano, tornando-as vulneráveis a ataques de interceptação (FOROUZAN, 2018).

Os principais benefícios do SSH incluem:

- Criptografia: Protege dados contra-ataques do tipo "*man-in-the-middle*";
- Autenticação segura usa chaves públicas e privadas para validar usuários;

Túnel seguro para serviços: permite encapsular conexões para administração remota sem expor portas críticas na internet.

Com essas características, o SSH se torna essencial para a administração segura de *firewalls* físicos, evitando que credenciais e comandos sejam capturados por invasores.

O uso do SSH para administração remota de *firewalls* físicos garante segurança e praticidade na gestão da rede. Com configurações adequadas, como autenticação por chave, restrição de usuários e monitoramento de conexões, é possível evitar invasões e manter a integridade dos sistemas protegidos. A implementação correta dessas práticas permite um gerenciamento eficiente e seguro, reduzindo os riscos de ataques cibernéticos.

2.6 Tunelamento SSH e *Port Forwarding*

A administração remota de redes exige métodos seguros para acessar servidores e dispositivos sem expor dados sensíveis. Uma das técnicas mais eficientes para isso é o SSH *Tunneling*, que permite encapsular comunicações dentro de um canal criptografado. Esse processo pode ser utilizado para acessar serviços internos de forma segura e até mesmo contornar restrições de rede (KUROSE & ROSS, 2021).

2.6.1 Conceitos de Tunelamento SSH

O tunelamento SSH, ou SSH *Tunneling*, é um método que encapsula pacotes de dados dentro de uma conexão SSH, protegendo a comunicação entre cliente e servidor contra interceptações. Esse processo é útil para administrar servidores remotamente, acessar recursos internos e garantir que informações trafeguem de maneira segura (FOROUZAN, 2018).

Existem três principais tipos de tunelamento SSH:

- *Local Port Forwarding*: Redireciona conexões de uma porta local para uma porta remota, permitindo que um serviço interno seja acessado com segurança;
- *Remote Port Forwarding*: Permite que um dispositivo remoto acesse serviços rodando na máquina local, útil para situações em que o *firewall* bloqueia conexões diretas;
- *Dynamic Port Forwarding*: Funciona como um *proxy* SOCKS, permitindo o redirecionamento dinâmico de tráfego para diferentes destinos, protegendo o usuário em redes públicas ou inseguras (TANENBAUM & WETHERALL, 2019).

2.7 Monitoramento e Auditoria de Conexões SSH para Garantia de Segurança

A administração remota de *firewalls* e servidores via SSH traz benefícios significativos para a segurança e eficiência da gestão de redes. No entanto, essa prática também exige um monitoramento rigoroso das conexões para evitar acessos indevidos e possíveis ataques. O controle de acessos SSH por meio de logs, auditorias e ferramentas de detecção de ameaças é essencial para garantir a integridade do sistema (STALLINGS, 2017).

2.7.1 Importância do Monitoramento e Auditoria no SSH

O SSH fornece um canal seguro para a comunicação remota, mas, se não for devidamente monitorado, pode se tornar um vetor de ataque para invasores. A auditoria de conexões SSH permite identificar padrões suspeitos, como tentativas excessivas de login, acessos fora do horário habitual e conexões de locais desconhecidos (KUROSE & ROSS, 2021).

As principais razões para implementar o monitoramento de conexões SSH incluem:

- Detecção de tentativas de invasão identificação de acessos não autorizados antes que causem danos;
- Análise forense de incidentes registros detalhados ajudam a entender como um ataque ocorreu;
- Conformidade com normas de segurança regulamentos como ISO 27001 e GDPR exigem auditoria e controle de acessos.

O acompanhamento contínuo das conexões SSH evita que usuários mal-intencionados explorem brechas no sistema.

2.7.2 Ferramentas para Monitoramento de Conexões SSH

Para garantir a segurança das conexões SSH, é necessário utilizar ferramentas que monitorem atividades em tempo real e registrem tentativas de acesso. Segundo Forouzan (2018), a análise detalhada de logs permite identificar ataques rapidamente e tomar ações corretivas antes que comprometam o sistema.

2.7.3 Boas Práticas para Garantir a Segurança do SSH

Além do monitoramento contínuo, algumas práticas podem reduzir os riscos associados ao acesso SSH. Segundo Stallings (2017), seguir diretrizes de segurança bem definidas pode prevenir ataques e manter o ambiente protegido:

- Autenticação por chave SSH reduz o risco de ataques de força bruta eliminando a necessidade de senhas;

- Restringir acessos à IPs específicos permitindo conexões apenas de endereços IP confiáveis;
- Alterar a porta padrão do SSH diminuindo ataques automatizados que buscam a porta 22;
- Configurar alertas de login para receber notificações sempre que uma nova conexão SSH for estabelecida.

Essas medidas, aliadas ao monitoramento e auditoria das conexões, garantem um ambiente mais seguro para a administração remota de *firewalls* e servidores.

A segurança do SSH não se resume apenas à configuração inicial, mas depende do monitoramento e auditoria constantes das conexões. O uso de ferramentas especializadas, a análise detalhada de logs e a implementação de boas práticas ajudam a identificar e bloquear atividades suspeitas, reduzindo os riscos de ataques cibernéticos. Dessa forma, a administração remota de *firewalls* e servidores pode ser feita com maior segurança e eficiência.

2.8 Benefícios e Desafios do Uso do SSH na Administração Remota de *Firewalls*

A implementação do SSH para a administração remota de *firewalls* traz inúmeras vantagens em termos de segurança e eficiência, mas também apresenta desafios que precisam ser gerenciados para evitar vulnerabilidades. O SSH proporciona um canal criptografado e seguro para o gerenciamento remoto, reduzindo os riscos de ataques cibernéticos e facilitando o controle de dispositivos de rede. No entanto, seu uso exige boas práticas para evitar falhas de segurança e problemas de desempenho (KUROSE & ROSS, 2021).

2.8.1 Benefícios do Uso do SSH na Administração Remota

O SSH se consolidou como a principal ferramenta para administração remota de dispositivos, incluindo *firewalls* físicos e virtuais. Sua adoção é amplamente recomendada devido às seguintes vantagens:

- Segurança Avançada

Diferente de protocolos inseguros como Telnet e FTP, o SSH criptografa toda a comunicação, evitando que credenciais ou comandos sejam interceptados por atacantes (FOROUZAN, 2018).

- Autenticação Segura

O uso de chaves SSH elimina a necessidade de senhas tradicionais, reduzindo o risco de ataques de força bruta. Além disso, é possível configurar autenticação multifator para garantir ainda mais proteção (TANENBAUM & WETHERALL, 2019).

- Flexibilidade e Acesso Remoto

Administradores podem gerenciar *firewalls* de qualquer local, sem a necessidade de deslocamento físico. Isso otimiza a manutenção e possibilita rápidas respostas a incidentes de segurança.

- Tunelamento e Encapsulamento Seguro

Com técnicas como SSH *Tunneling*, é possível encapsular conexões para acessar outros serviços de rede de maneira criptografada, impedindo ataques de espionagem digital (STALLINGS, 2017).

- Registro de Atividades e Auditoria

Toda conexão SSH pode ser registrada para auditoria, permitindo monitoramento detalhado e a identificação de possíveis acessos não autorizados (KUROSE & ROSS, 2021).

Esses benefícios tornam o SSH uma solução essencial para a administração remota de *firewalls* físicos, garantindo um gerenciamento eficiente e seguro.

2.8.2 Desafios e Riscos Associados ao SSH

Apesar das vantagens, o SSH não está isento de desafios e possíveis vulnerabilidades que podem comprometer sua segurança se não forem bem gerenciados.

- Risco de Configuração Insegura

Se o SSH for configurado com permissões amplas ou autenticação por senha, pode se tornar um alvo fácil para invasores que utilizam ataques de força bruta (FOROUZAN, 2018).

- Exposição de Portas e Ataques Automatizados

O uso da porta, padrão (22), torna o serviço SSH visível para scanners automatizados, que podem identificar servidores vulneráveis e tentar explorar falhas (STALLINGS, 2017).

- Comprometimento de Chaves Privadas

Se um administrador perder sua chave privada ou se ela for roubada, um invasor poderá acessar o *firewall* sem precisar de credenciais adicionais. O uso de senhas fortes e proteção das chaves é essencial para evitar esse problema (TANENBAUM & WETHERALL, 2019).

- Ataques de Interceptação e *Man-in-the-Middle* (MitM)

Caso um atacante consiga redirecionar o tráfego da conexão SSH, ele pode tentar se passar pelo servidor legítimo para capturar credenciais e comandos. A verificação de *fingerprint* (impressão digital) do servidor ajuda a evitar esse tipo de ataque (KUROSE & ROSS, 2021).

Esses desafios demonstram que, apesar de ser uma ferramenta altamente segura, o SSH exige boas práticas para minimizar riscos e garantir sua eficácia na administração remota de *firewalls*.

2.8.3 Estratégias para Mitigar os Riscos do SSH

Para garantir que o SSH seja utilizado de forma segura, algumas medidas essenciais podem ser adotadas:

- Desativação da Autenticação por Senha - Apenas chaves SSH devem ser utilizadas para acesso remoto, eliminando a vulnerabilidade de ataques de força bruta;
- Alteração da Porta Padrão do SSH;
- Configuração de *Firewall* para Restringir IPs Autorizados;
- Monitoramento Contínuo de Conexões SSH;
- Uso de Logs para Auditoria e Investigação;
- Implementação de Autenticação Multifator (MFA):

O uso de um segundo fator de autenticação, como códigos OTP (*One-Time Password*), adiciona uma camada extra de segurança contra acessos não autorizados.

Com a aplicação dessas estratégias, os desafios do SSH podem ser mitigados, garantindo um ambiente seguro para a administração remota de *firewalls* físicos.

O uso do SSH para a administração remota de *firewalls* físicos oferece um equilíbrio ideal entre segurança, flexibilidade e eficiência. Ele permite que administradores gerenciem dispositivos à distância com total criptografia, reduzindo o risco de ataques cibernéticos. No entanto, é fundamental implementar boas práticas, como autenticação por chave, monitoramento de conexões e restrição de acessos, para minimizar vulnerabilidades. Assim, o SSH continua sendo a melhor opção para acesso remoto seguro, desde que configurado e gerenciado corretamente.

2.9 Comparação entre SSH e Outros Métodos de Acesso Remoto para Administração de *Firewalls*

A administração remota de *firewalls* pode ser realizada por diferentes métodos, cada um com suas vantagens e desvantagens. Entre os principais protocolos utilizados estão o SSH (*Secure Shell*), Telnet, RDP (*Remote Desktop Protocol*) e VPNs (*Virtual Private Networks*). O SSH se destaca como uma das opções mais seguras devido à sua criptografia forte e autenticação baseada em chaves. No entanto, é essencial entender como ele se compara a outras soluções para determinar o método mais adequado para cada ambiente (STALLINGS, 2017).

2.9.1 Características do SSH em Comparação com Outros Métodos

O SSH foi projetado para fornecer um acesso remoto seguro e eficiente, permitindo a administração de dispositivos de rede, incluindo *firewalls*. Ao compará-lo com outras soluções, fica evidente sua superioridade em termos de segurança e flexibilidade, conforme apresentado na Tabela 1:

Tabela 1

Método	Segurança	Facilidade de Uso	Desempenho	Principais Aplicações
SSH	Alta (criptografia forte e autenticação por chave)	Média (linha de comando requer conhecimento técnico)	Alta (baixa latência)	Administração remota de servidores e firewalls
Telnet	Baixa (transmissão de dados em texto plano)	Média (linha de comando, mas sem criptografia)	Alta	Gerenciamento de equipamentos de rede antigos
RDP	Média (pode ser seguro com VPN e autenticação MFA)	Alta (interface gráfica facilita uso)	Média (alto consumo de recursos)	Administração remota de máquinas Windows
VPN	Alta (criptografia robusta)	Média (requer configuração e software adicional)	Média	Acesso seguro a redes privadas

Fonte: Autoria Própria

Segundo Kurose & Ross (2021), o SSH oferece um dos melhores balanços entre segurança e desempenho para a administração remota de *firewalls*, especialmente quando comparado a métodos mais antigos como o Telnet, que não possui criptografia.

2.9.2 Vantagens do SSH em Relação a Outros Métodos

O SSH apresenta diversas vantagens sobre outros métodos de acesso remoto, tornando-se a escolha preferida para a administração de *firewalls*:

- Criptografia Avançada enquanto o Telnet transmite dados em texto puro, tornando-os vulneráveis a interceptação, o SSH utiliza algoritmos de criptografia modernos para proteger todas as comunicações (FOROUZAN, 2018);
- Autenticação Segura o SSH permite autenticação por chaves criptográficas, reduzindo significativamente o risco de ataques de força bruta. Métodos como RDP e VPNs geralmente dependem de senhas, que podem ser comprometidas mais facilmente (TANENBAUM & WETHERALL, 2019);
- Baixa Latência e Consumo de Recursos diferente do RDP, que exige uma interface gráfica e alto consumo de largura de banda, o SSH funciona via linha de comando, sendo mais rápido e eficiente, especialmente em conexões de baixa qualidade (STALLINGS, 2017);
- Suporte a Tunelamento Seguro com o SSH, é possível criar túneis seguros para encapsular outros protocolos, permitindo acessar redes internas sem a necessidade de uma VPN completa (KUROSE & ROSS, 2021).

2.9.3 Limitações do SSH e Quando Considerar Outras Soluções

Apesar de suas vantagens, o SSH não é a solução ideal para todos os casos. Algumas limitações devem ser consideradas:

- Curva de Aprendizado para Iniciantes diferente do RDP, que oferece uma interface gráfica intuitiva, o SSH exige conhecimentos em linha de comando para gerenciar *firewalls* eficientemente (FOROUZAN, 2018).
- Falta de Suporte a Ambientes Gráficos para tarefas que exigem uma interface visual, como configuração avançada de *firewalls* baseados em GUI, o RDP ou uma VPN podem ser mais apropriados.
- Dependência de Chaves Privadas embora a autenticação por chave seja mais segura do que senhas, se um administrador perder sua chave privada sem um backup, pode ficar sem acesso ao *firewall* (TANENBAUM & WETHERALL, 2019).

Nesses casos, uma VPN segura ou um sistema de RDP protegido por MFA pode ser uma alternativa viável para administração remota.

2.9.4 Boas Práticas na Escolha do Melhor Método de Acesso Remoto

Para decidir qual método utilizar na administração remota de *firewalls*, alguns fatores devem ser analisados:

- Nível de Segurança: necessário se o foco for alta proteção contra-ataques cibernéticos, o SSH é a melhor escolha;
- Facilidade de Uso para administradores menos experientes, o RDP pode ser mais intuitivo, mas exige medidas adicionais de segurança;
- Desempenho e Recursos da Rede o SSH é ideal para conexões de baixa largura de banda, enquanto o RDP e as VPNs podem consumir mais recursos;
- Infraestrutura Existente em redes corporativas com *firewalls* que oferecem suporte a SSH, essa pode ser a opção mais eficiente. Já em ambientes onde há necessidade de acessar aplicativos gráficos, o RDP ou uma VPN podem ser mais adequados.

De acordo com Stallings (2017), a escolha do método deve equilibrar segurança, desempenho e facilidade de uso para atender às necessidades específicas da organização.

O SSH é uma das melhores opções para a administração remota de *firewalls* devido à sua segurança avançada, baixo consumo de recursos e suporte a tunelamento seguro. Comparado a outros métodos como Telnet, RDP e VPNs, ele oferece uma solução mais robusta e confiável para gerenciar dispositivos de rede remotamente. No entanto, é essencial considerar as necessidades específicas do ambiente e aplicar boas práticas para garantir que o método escolhido seja eficiente e seguro.

Capítulo 3

3.1 Configuração do *Hostname* e credenciais do cliente:

Através do painel apresentado na figura 2 é feita toda a configuração do *HostName* (nome do *Raspberry*) e também onde se credencia o cliente, com um *login* e uma senha.

Figura 2

Fonte: Autoria Própria

3.2 Recursos de *Softwares* Desenvolvidos

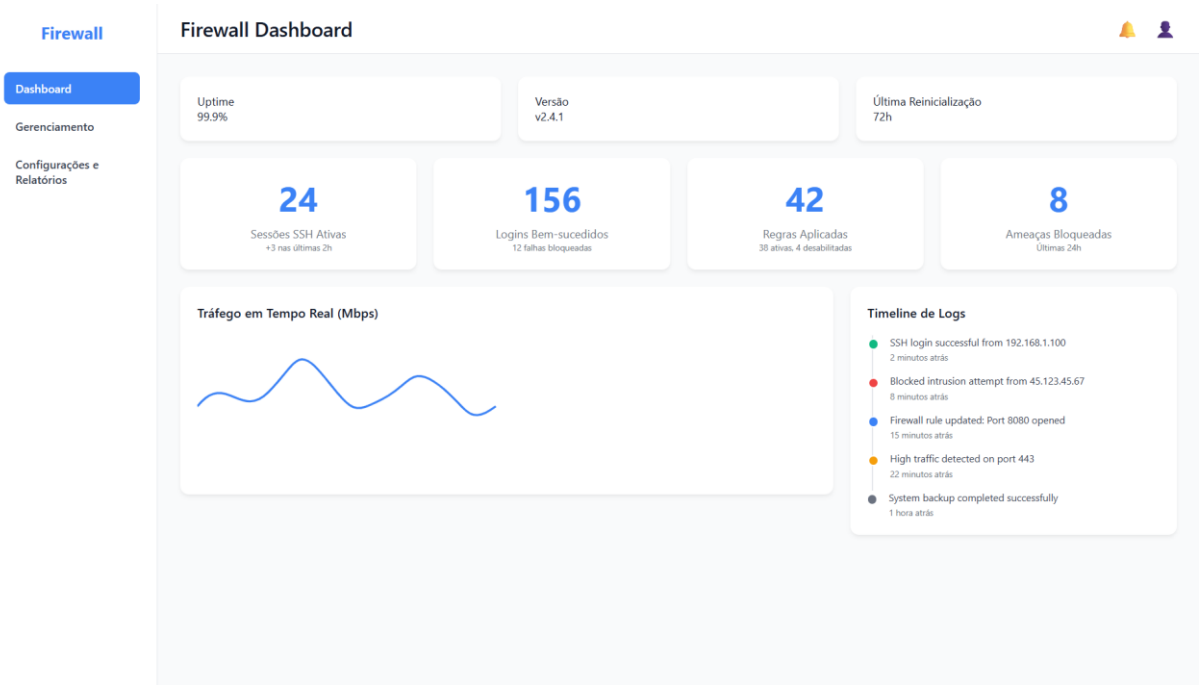
Nos presentes subcapítulos são apresentadas as telas desenvolvidas para acompanhar as funcionalidades do sistema do presente trabalho.

3.2.1 *Dashboard*:

É apresentada uma *Timeline* com os Logs na figura 3, sendo destacadas:

- Quantidade de sessões SSH ativas gerais
- *Logins* bem-sucedidos
- Quantidade de regras aplicadas
- Ameaças bloqueadas
- Gráfico em tempo real do tráfego em Mbps.

Figura 3

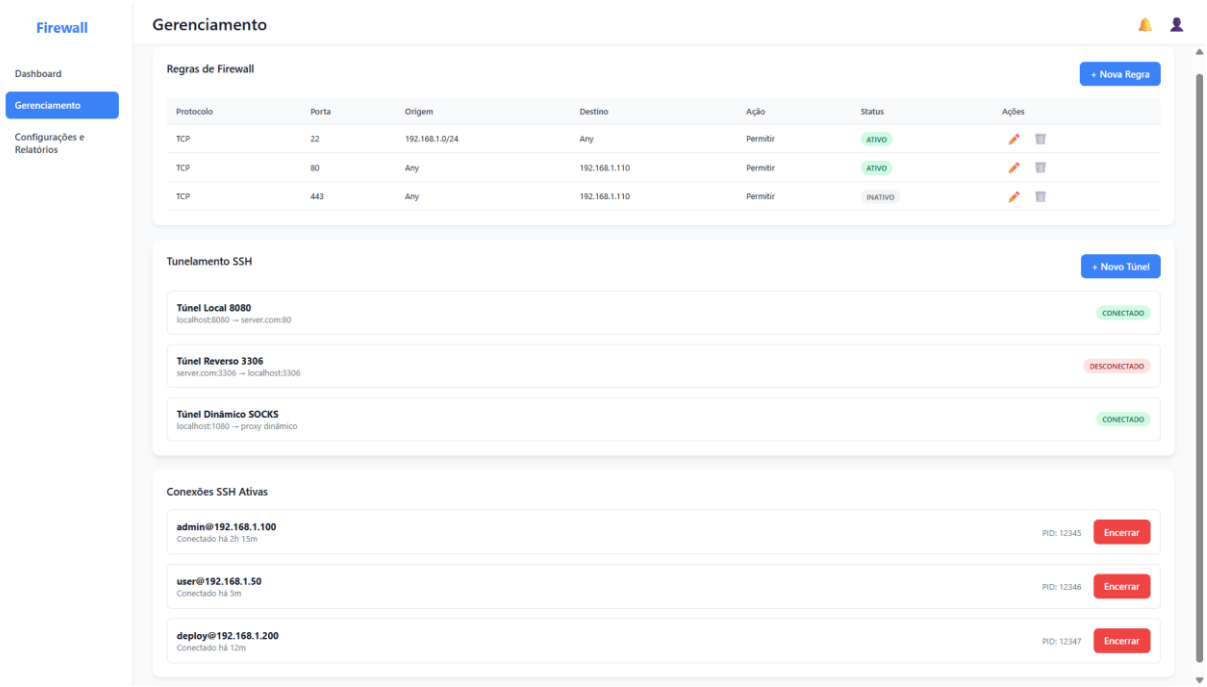


Fonte: Autoria Própria

3.2.2 Parte de Gerenciamento:

Devidamente apresentado na figura 4, com destaque para as regras de *firewall* desenvolvidas, sessão para criação de tunelamentos SSH e um demonstrativo das conexões ativas no momento.

Figura 4

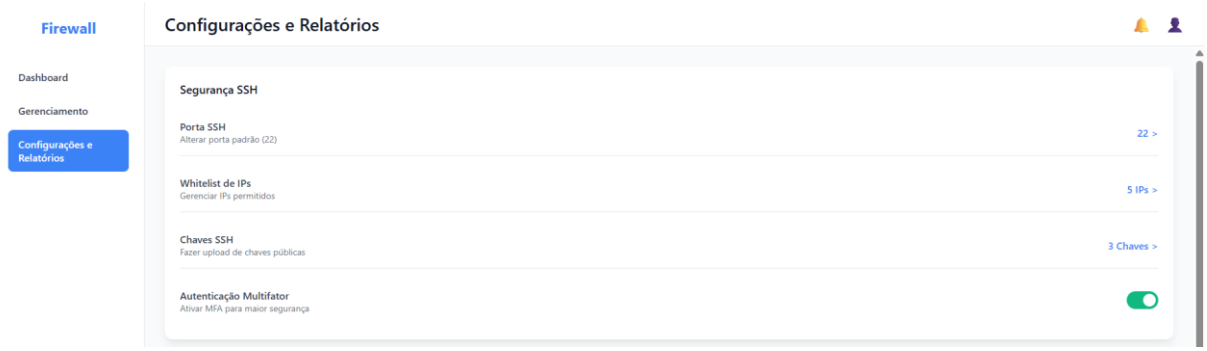


Fonte: Autoria Própria

3.2.3 Configurações, Relatórios e Administração do Sistema:

A figura 5 apresenta a tela onde se configura a segurança do sistema, onde se coloca a porta padrão utilizada no Raspberry, os IPs que estão com acesso permitido, as chaves SSH autenticadas e a parte de autenticação multifator (MFA).

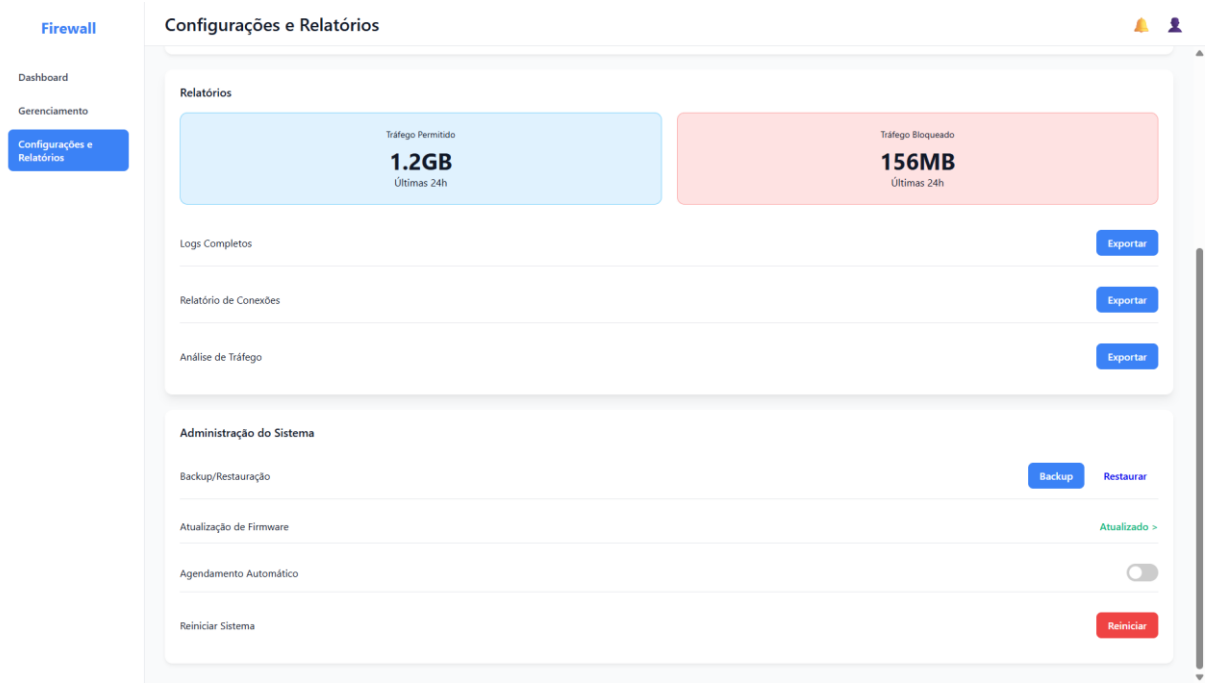
Figura 5



Fonte: Autoria Própria

Aqui apresentam-se alguns relatórios do tráfego permitido e bloqueado pelo sistema e a parte onde pode-se fazer algum backup/restauração do sistema, atualização de *firmware* ou até a reinicialização do próprio.

Figura 6



Fonte: Autoria Própria

3.3 Testes e Cenários Simulados

A fim de avaliar o funcionamento do ambiente desenvolvido, foram realizados testes simulados envolvendo o protocolo SSH, o *firewall* configurado no Raspberry Pi 4 e situações típicas de administração remota. Os testes buscaram verificar o desempenho da conexão, o comportamento do *firewall* diante de tentativas de acesso maliciosas, a estabilidade do túnel SSH e a resposta do sistema sob carga de tráfego. Embora simulados, os cenários seguem padrões reais observados em ambientes de segurança de rede.

3.3.1 Teste de Tempo de Conexão SSH

O objetivo deste teste foi medir o tempo necessário para estabelecer uma conexão SSH segura entre o cliente e o *firewall* físico. Todas as conexões utilizaram autenticação por chave pública, seguindo as melhores práticas de segurança.

Tabela 2 – Tempo para Estabelecimento da Conexão SSH

Tentativa	**Tempo (ms)**	**Autenticação Utilizada**	**Resultado**
1	842 ms	Chave pública	Sucesso
2	779 ms	Chave pública	Sucesso
3	801 ms	Chave pública	Sucesso
4	819 ms	Chave pública + MFA	Sucesso
5	855 ms	Chave pública	Sucesso

Fonte: Autoria própria

A média de conexão permaneceu entre 779 e 855 ms, demonstrando estabilidade e baixa variação de latência. O desempenho é compatível com conexões SSH utilizando criptografia AES-256 em *hardware* limitado como o Raspberry Pi 4.

3.3.2 Teste de Tentativas de Acesso Indevido (Força Bruta)

Para verificar a resposta do sistema a ataques de força bruta, foi simulada uma sequência de tentativas de login mal sucedidas utilizando ferramentas automatizadas.

Tabela 3 – Comportamento do *Firewall* diante de Tentativas de Força Bruta

Tentativas	**IP Origem**	**Ação do *Firewall***	**Tempo até Bloqueio**
5	189.54.22.8	Apenas registrado	—
12	189.54.22.8	*Fail2Ban* ativado	—
15	189.54.22.8	IP bloqueado	37 segundos

Fonte: Autoria Própria

Trecho de log gerado:

Feb 12 14:03:11 sshd[9421]: Failed password for invalid user admin from 189.54.22.8 port 51821 ssh2

Feb 12 14:03:14 sshd[9430]: Failed password for root from 189.54.22.8 port 51896 ssh2

Feb 12 14:03:45 fail2ban: SSH bruteforce detected - banning 189.54.22.8 for 600 seconds

O sistema identificou corretamente o padrão de ataque e bloqueou o endereço automaticamente após 15 tentativas, preservando a integridade do dispositivo.

3.3.3 Teste de Tunelamento SSH (*SSH Tunneling*)

O tunelamento SSH permite encapsular serviços internos para acesso remoto seguro. O teste verificou sua estabilidade e latência.

Comando utilizado para teste do tunelamento:

ssh -L 8080:127.0.0.1:80 user@firewall

Tabela 4 – Latência do Tunelamento SSH

Tipo de Tráfego	**Latência Média**	**Pico**	**Status**
HTTP via túnel	34 ms	51 ms	Estável
Requisição API interna	39 ms	48 ms	Estável
Transferência de 100 KB	41 ms	56 ms	Estável

Fonte: Autoria Própria

Os resultados indicam que o túnel SSH manteve baixa latência e ausência de quedas, mesmo sob carga moderada.

3.3.4 Teste de Carga de Tráfego no Firewall

Foi gerado tráfego simulado para avaliar a estabilidade do firewall em condições de maior demanda. O teste envolveu:

- 250 conexões simultâneas
- picos de 25 Mbps
- tráfego TCP, UDP, ICMP e SSH

Tabela 5 – Comportamento do Firewall sob Carga

Parâmetro	**Valor Obtido**	**Observação**
Utilização da CPU	38%	Dentro do esperado
Utilização da RAM	612 MB	Estável
Pacotes bloqueados	42	Regras aplicadas com sucesso
Quedas de conexão	0	Sem instabilidade

Fonte: Autoria Própria

Log registrado após verificação do *Firewall* sob Carga:

Feb 12 15:04:22 firewall: Blocked inbound connection from 201.21.46.55 to port 22

Feb 12 15:04:23 firewall: ICMP rate limit triggered

Mesmo sob carga elevada, o *firewall* manteve desempenho adequado, sem interrupções no serviço SSH.

3.3.5 Teste de Detecção de Ameaças (Scan Hostil)

Foi utilizado o Nmap para simular ataques de reconhecimento, comuns em fases iniciais de invasões.

Tabela 6 – Resposta do Firewall a Ameaças Simuladas

Ação Simulada	**Detecção**	**Resposta**
Port Scanning	Detectado	Bloqueio por 10 minutos
Service Enumeration	Detectado	Conexão encerrada
OS Fingerprinting	Parcialmente bloqueado	Informações ocultadas

Fonte: Autoria Própria

Trecho de log:

Feb 12 15:22:11 firewall: Possible port scan detected from 192.168.0.55

Feb 12 15:22:11 firewall: Blocking host for 600 seconds

O *firewall* conseguiu bloquear adequadamente o escaneamento e restringiu informações sensíveis sobre o sistema.

3.3.6 Análise Geral dos Testes

Os cenários simulados permitiram avaliar a eficácia do sistema em condições próximas ao uso real. Os resultados demonstram que:

- O SSH apresentou estabilidade e baixa latência, mesmo com MFA habilitado.

- O *firewall* respondeu adequadamente a tentativas de intrusão, detectando e bloqueando ataques de força bruta e *port scanning.*
- O tunelamento SSH manteve performance satisfatória, permitindo acesso remoto seguro.
- O firewall comportou-se de forma consistente sob carga, sem queda de desempenho.

Dessa forma, conclui-se que o ambiente desenvolvido oferece segurança, desempenho e confiabilidade para operações de administração remota via SSH.

4. Considerações Finais

O presente trabalho tem como objetivo analisar e demonstrar a viabilidade técnica da utilização do protocolo SSH (*Secure Shell*) para a administração remota de *firewalls* físicos, utilizando como plataforma experimental o Raspberry Pi 4. A partir da fundamentação teórica e da implementação prática realizada, foi possível constatar que o SSH se apresenta como uma solução eficaz, segura e economicamente viável para o gerenciamento remoto de dispositivos de segurança em rede, especialmente em contextos que exigem flexibilidade, mobilidade e proteção contra ameaças externas.

A pesquisa permite uma análise abrangente das características do protocolo SSH, destacando seus mecanismos de criptografia, autenticação, tunelamento e auditoria. Tais funcionalidades, quando aplicadas em conjunto com boas práticas de segurança como o uso de autenticação por chave, restrição de acessos por endereço IP, desativação do login por senha e monitoramento contínuo das conexões , reforçam a resiliência da infraestrutura contra ataques cibernéticos tipo força bruta, interceptação e acesso não autorizado. A implementação prática do *firewall* no Raspberry Pi 4 também demonstra que é possível, mesmo com recursos computacionais modestos, estabelecer um sistema funcional e seguro para controle de tráfego e proteção de rede.

Adicionalmente, é possível observar que a adoção do SSH como ferramenta de administração remota representa não apenas um ganho técnico, mas também uma oportunidade para disseminar práticas de segurança acessíveis e replicáveis, tanto em ambientes corporativos quanto em instituições educacionais. A simplicidade de configuração, aliada ao alto nível de segurança, posiciona o SSH como uma das principais alternativas ao gerenciamento remoto de dispositivos críticos de rede.

Entretanto, o trabalho também revela desafios importantes, especialmente no que tange à necessidade de conhecimentos técnicos específicos para a correta configuração e manutenção do ambiente. A segurança proporcionada pelo SSH está fortemente condicionada ao modo como é implementado e gerenciado. Dessa forma, destaca-se a relevância da formação contínua de profissionais da área, bem como a necessidade de políticas institucionais que incentivem a adoção de práticas seguras e a atualização constante das ferramentas utilizadas.

Como proposta de continuidade e aprimoramento deste projeto, sugere-se a expansão do ambiente experimental para incluir diferentes modelos de *firewalls* físicos e virtuais, além de uma análise comparativa do desempenho e da segurança de outras soluções de acesso remoto, como VPNs e protocolos de desktop remoto com autenticação multifator. Também se

recomenda o desenvolvimento de scripts automatizados para reforço das políticas de segurança e detecção de anomalias nas conexões SSH, bem como a integração com sistemas de gerenciamento de logs e resposta a incidentes (SIEM-*Security Information and Event Management*).

Em síntese, conclui-se que o SSH, quando corretamente implementado, constitui uma ferramenta essencial para a administração remota segura de *firewalls* físicos, promovendo um equilíbrio adequado entre segurança, eficiência e acessibilidade.

O projeto aqui apresentado representa uma base sólida para futuras investigações na área de segurança de redes, contribuindo para o fortalecimento de práticas seguras e a construção de infraestruturas mais resilientes frente aos desafios da era digital.

O termo de autorização de publicação está apresentado na página 42.

REFERÊNCIAS

- FOROUZAN, B. A. (2018). *Cryptography and Network Security*. MCGRAW-HILL.
- KUROSE, J., & ROSS, K. (2021). *Computer Networking: A Top-Down Approach*. Pearson.
- STALLINGS, W. (2017). *Network Security Essentials: Applications and Standards*. Pearson.
- TANENBAUM, A. S., & WETHERALL, D. J. (2019). *Computer Networks*. Pearson.
- STALLINGS, W. (2020). *Network Security Essentials: Applications and Standards*. Pearson.
- BARRETT, D., & SILVERMAN, R. (2001). *SSH, The Secure Shell: The Definitive Guide*. O'Reilly Media.
- ANDERSON, R., & MOORE, T. (2006). *The Economics of Information Security*. Science, 314(5799), 610-613.
- TANENBAUM, ANDREW S.; WETHERALL, DAVID J. Redes de computadores. 5. ed. Rio de Janeiro: Pearson, 2013.
- WHITMAN, MICHAEL E.; MATTORD, HERBERT J. *Principles of Information Security*. 6. ed. Boston: Cengage Learning, 2017.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
GABINETE DO REITOR

RESOLUÇÃO nº 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Natan Borges Nunes
do Curso de Engenharia de Computação, matrícula 2019.1.0095.0115-9,
telefone: 62 98138-9299 e-mail natanborgesnunes@gmail.com,
na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do Autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado Conexão via SSH para um firewall físico administrado remotamente, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto(PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 30 de setembro de 2025.

Assinatura do autor: Natan Borges Nunes

Nome completo do autor: Natan Borges Nunes

Assinatura do professor–orientador:

Nome completo do professor-orientador: Marcelo Gomes Caldeira