

Hardening de Sistema Operacional Linux Ubuntu

Éverton Victor S. Silva

CECyber Capacitação e Soluções de Tecnologias LTDA, São Paulo - São Paulo, evertonvictorpro@gmail.com

Pós - Segurança da Informação e Inteligência Defensiva

Alameda Rio Negro, 503 - Sala 2020, Alphaville Centro Industrial e Empresarial/Alphav

Barueri/SP, Brasil, 06454-000

Resumo – Este trabalho apresenta diretrizes para o processo de "hardening" de sistemas operacionais Linux, com ênfase na distribuição Ubuntu Server. O estudo baseia-se nos conteúdos abordados nas disciplinas da Pós-Graduação em Segurança da Informação e Inteligência Defensiva, com destaque para as aulas de Cloud & Mobile Security, ministradas pelo professor Eduardo R. Sant'Ana Popovici. Os tópicos aqui apresentados foram estruturados a partir da realização de procedimentos experimentais conduzidos em laboratório. O artigo aborda as principais práticas e recomendações para o fortalecimento da segurança em sistemas operacionais Linux, com foco específico no Ubuntu Server.

Palavras-chave — Hardening, Ubuntu Server, Segurança da Informação, Linux, Vulnerabilidades.

I. INTRODUÇÃO

A crescente dependência de servidores Linux no ambiente corporativo e em infraestruturas críticas torna essencial a implementação de medidas de segurança que garantam a integridade, disponibilidade e confidencialidade das informações. Entre as distribuições mais amplamente utilizadas, destaca-se o Ubuntu Server [1], amplamente adotado devido à sua estabilidade, flexibilidade e forte suporte da comunidade.

A prática de "hardening", ou endurecimento de sistemas, visa reduzir a superfície de ataque de um servidor por meio da aplicação de boas práticas de segurança, remoção de serviços desnecessários, configuração adequada de permissões e implementação de mecanismos de defesa, como firewall, controle de acesso e ferramentas de auditoria [2][3].

Este artigo apresenta um estudo prático sobre a identificação e mitigação de vulnerabilidades em uma instalação do Ubuntu Server. Inicialmente, configura-se um ambiente com falhas propositalmente, simulando um cenário real

de um servidor mal configurado ou negligenciado. Em seguida, são aplicadas técnicas de "hardening" para elevar o nível de segurança do sistema. Por fim, são realizados testes de validação para avaliar a eficácia das medidas adotadas.

O trabalho baseia-se em conceitos abordados na Pós-Graduação em Segurança da Informação e Inteligência Defensiva, com foco nas diretrizes discutidas na disciplina de Cloud & Mobile Security. Além disso, busca-se alinhar as práticas apresentadas às recomendações de segurança reconhecidas internacionalmente, como as diretrizes da norma ISO/IEC 27002 [4] e os benchmarks do Center for Internet Security (CIS) [5], fornecendo um guia didático e aplicável para profissionais da área de cibersegurança.

II. METODOLOGIA

Este trabalho foi desenvolvido em um ambiente de laboratório utilizando uma máquina virtual com configuração específica e propositalmente vulnerável, a fim de demonstrar falhas de segurança comuns em servidores.

A. Configuração Inicial do Ambiente

O ambiente foi configurado em uma Máquina Virtual (VM) utilizando a plataforma VMware. O sistema operacional instalado foi o Ubuntu Server 22.04 LTS, com as seguintes especificações:

- Processadores: 2 CPUs virtuais;
- Memória RAM: 4 GB;
- Armazenamento: 50 GB de disco virtual.
- Placa de Rede:
 - VMnet10, Host Only, 192.168.171.129/24
 - VMnet0, NAT, 10.180.20.100/24

Foram inseridas configurações intencionalmente inseguras para fins de análise, sendo elas:

- SSH com permissão de login direto como root;
- Firewall desativado;
- Serviços desnecessários ativos, como Apache e FTP;
- Contas de usuários com senhas fracas;
- Diretórios críticos com permissões inadequadas de acesso.

B. Identificação das Vulnerabilidades

O ambiente foi configurado em uma Máquina Virtual (VM) utilizando a plataforma VMware. O sistema operacional instalado foi o Kali Linux, com as seguintes especificações:

- Processadores: 2 CPUs virtuais;
- Memória RAM: 2 GB;
- Armazenamento: 25 GB de disco virtual.
- Placa de Rede:
 - VMnet10, Host Only, 192.168.171.130/24

A identificação das vulnerabilidades foi realizada a partir de um segundo ambiente de análise, utilizando uma máquina com Kali Linux [6]. Durante a varredura e análise, foram constatadas as seguintes fragilidades:

- Ausência de políticas de senha robustas;
- Serviços desnecessários e expostos à rede;
- Falta de atualizações de segurança do sistema e pacotes instalados.
- Firewall desativado;
- SSH com permissão de login direto como root;
- Diretórios críticos com permissões inadequadas de acesso.

As informações coletadas foram fundamentais para posterior etapa de mitigação e aplicação de boas práticas de segurança.

III. CONFIGURAÇÃO DAS VULNERABILIDADES

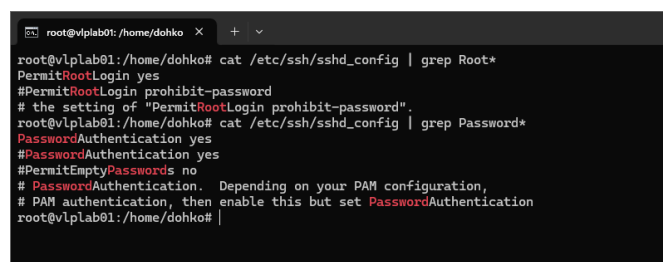
Esta etapa visa a criação de um ambiente propositalmente inseguro, a fim de servir como base para a aplicação posterior de medidas de segurança (hardening). O processo foi dividido em diversas ações que resultam em um servidor exposto a riscos comuns de ambientes mal configurados.

A. Ambiente e Objetivo

O objetivo foi configurar um Ubuntu Server 22.04 LTS vulnerável para fins educacionais, permitindo a análise de falhas e a implementação de práticas de correção. A instalação do sistema ocorreu em uma máquina virtual conforme descrito na Metodologia.

B. Configuração Insegura do SSH

O serviço SSH foi configurado para permitir acesso direto ao usuário root e autenticação via senha. As alterações foram realizadas no arquivo `/etc/ssh/sshd_config`, conforme a Fig. 1.



```

root@vlab01: /home/dohko
root@vlab01: /home/dohko# cat /etc/ssh/sshd_config | grep Root*
PermitRootLogin yes
#PermitRootLogin prohibit-password
# the setting of "PermitRootLogin prohibit-password".
root@vlab01: /home/dohko# cat /etc/ssh/sshd_config | grep Password*
PasswordAuthentication yes
#PasswordAuthentication yes
#PermitEmptyPasswords no
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication, then enable this but set PasswordAuthentication
root@vlab01: /home/dohko#
  
```

Fig. 1 - Configuração Insegura do SSH

Para fins de realização de testes, é necessário redefinir a senha do usuário root no sistema. O comando para realizar essa alteração é o seguinte: `sudo passwd root`.

Após executar este comando, será solicitada a definição de uma nova senha para o usuário root. Para simplificar o processo e garantir acesso rápido durante a execução de testes, recomenda-se o uso de uma senha simples. No contexto deste laboratório, a senha definida foi `P@ssword123`.

Essa prática permite o acesso facilitado para tarefas administrativas durante os testes, mas é importante ressaltar que o uso de senhas fortes e a alteração periódica das mesmas são essenciais para garantir a segurança do ambiente de produção.

C. Instalação de Pacotes e Serviços Desnecessários

Com o objetivo de criar uma superfície de ataque ampliada, foram instalados diversos serviços e pacotes que não são necessários para a função principal de um servidor, conforme a Fig. 2.

```

root@vlplab01:/home/dohko# dpkg -t | grep -E "(vsftpd|apache2|telnetd|ftplib|rsync-server|nmap|netcat-traditional|nmap)"
ii apache2 2.4.18-2ubuntu1.4 amd64 Apache HTTP Server (modules and other binary files)
ii apache2-bin 2.4.18-2ubuntu1.4 amd64 Apache HTTP Server (modules and other binary files)
ii apache2-data 2.4.18-2ubuntu1.4 amd64 Apache HTTP Server (common files)
ii apache2-utils 2.4.18-2ubuntu1.4 amd64 Apache HTTP Server (utility programs for web servers)
ii ftp 3.0.2-2ubuntu1 amd64 dummy transitional package for vsftpd
ii inetutils-telnetd 1.18-4b amd64 Telnet Server
ii netcat-traditional 1.10-4b amd64 TCP/IP Swiss Army Knife
ii nmap 7.90i120238897.3b01ef01d49g-3build2 amd64 The Network Mapper
ii nmap-common 7.90i120238897.3b01ef01d49g-3build2 all Architecture Independent files for nmap
ii openssh-sftp-server 1:9.6p1-3ubuntu1.5 amd64 Secure Shell (SSH) sftp server module, for SFTP access from
routers machines
ii openssh-server 8.4p1-4 amd64 Implementation of rshd and rlogind
ii rsh 8.17-15 amd64 Clients to query the rshd server
ii rsync 3.1.2-4ubuntu1 amd64 System status server
ii telnetd 0.17+2.5-3ubuntu1 amd64 transitional dummy package for inetutils-telnetd default ssi
ii vsftpd 3.0.2-2ubuntu1 amd64 enhanced ftp client
ii vsftpd 3.0.2-2ubuntu1 amd64 lightweight, efficient ftp server written for security

```

Fig. 2 - Instalação de Pacotes e Serviços Desnecessários

A seguir, detalhamos cada um deles e os riscos associados.

1) Very Secure FTP Daemon

Very Secure FTP Daemon (vsftpd), embora seja um servidor FTP considerado seguro quando bem configurado, neste caso, foi instalado com a configuração padrão e com acesso anônimo permitido, o que expõe o servidor a riscos de upload e download não autorizados de arquivos, além da transmissão de dados em texto claro (sem criptografia).

2) Apache

O servidor web Apache foi instalado mesmo sem necessidade, aumentando a superfície de ataque e a chance de exploração de vulnerabilidades conhecidas, caso o serviço permaneça desatualizado ou mal configurado.

3) Telnet

O serviço Telnet permite conexões remotas sem qualquer forma de criptografia, expondo as credenciais e comandos em texto claro na rede. Seu uso é considerado obsoleto e altamente inseguro em ambientes de produção, sendo facilmente explorável por ataques de sniffing ou man-in-the-middle.

4) RSH

O RSH (Remote Shell) é um serviço de acesso remoto antigo e inseguro que não utiliza criptografia na comunicação, sendo suscetível a interceptações e spoofing. Sua instalação cria um ponto adicional de risco em redes que operam com confiança implícita.

5) RWHO

O rwho é um serviço utilizado para exibir informações de usuários logados em máquinas da rede. A exposição deste serviço pode auxiliar um atacante no reconhecimento de usuários ativos e topologia da rede, facilitando ataques direcionados.

6) Netcat

O Netcat é uma ferramenta poderosa para diagnósticos e transferências de dados em redes, mas também amplamente utilizada em atividades maliciosas, como abertura de backdoors e movimentação lateral. Sua presença em servidores produtivos sem controle adequado é um risco potencial.

7) Nmap

Embora o Nmap seja uma ferramenta essencial para auditoria de redes, em um ambiente vulnerável sua instalação pode ser explorada por usuários maliciosos para realizar varreduras internas, mapeando portas abertas e serviços ativos, auxiliando na identificação de vetores de ataque.

4) INETD

O inetd é um antigo gerenciador de diversos serviços de rede, como Telnet e FTP. Sua ativação contribui para a manutenção de serviços obsoletos e inseguros em execução, como o Telnet, dificultando o gerenciamento seguro do servidor.

D. Permissões Inseguras em Arquivos Sensíveis

Durante a configuração do ambiente vulnerável, foram aplicadas permissões incorretas a arquivos críticos do sistema, ampliando significativamente os riscos de segurança.

A Fig. 3 ilustra um exemplo da configuração incorreta de permissões aplicada no ambiente.

```

root@vlplab01:/home/dohko# ls -la /etc/shadow
-rwxrwxrwx 1 root shadow 1005 Mar 21 02:24 /etc/shadow
root@vlplab01:/home/dohko# ls -la /etc/passwd
-rw-r--r-- 1 root root 1897 Mar 21 02:24 /etc/passwd
root@vlplab01:/home/dohko# chmod 777 /etc/passwd
root@vlplab01:/home/dohko# ls -la /etc/passwd
-rwxrwxrwx 1 root root 1897 Mar 21 02:24 /etc/passwd
root@vlplab01:/home/dohko#

```

Fig. 3 - Permissões Inseguras em Arquivos Sensíveis

A seguir, são descritas as ações realizadas e as implicações de cada uma.

1) Arquivo /etc/shadow com permissão 777

O arquivo /etc/shadow armazena os hashes de senha dos usuários do sistema. O correto seria que apenas o usuário root tivesse permissão de leitura e escrita (modo 640 ou 400). No entanto, foi aplicada permissão 777, permitindo leitura, escrita e execução para todos os usuários do sistema. Esta

configuração facilita que qualquer usuário ou processo malicioso possa acessar, modificar ou até substituir os hashes de senha, possibilitando escalonamento de privilégios ou negação de acesso.

2) Arquivo `/etc/passwd` com permissão 777

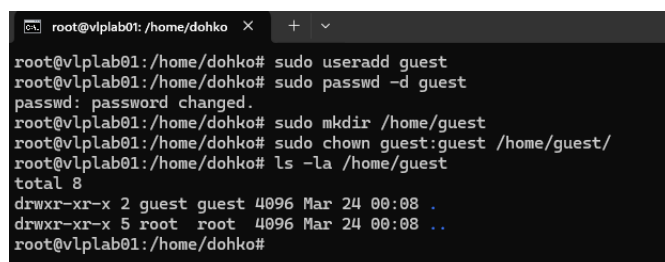
O `/etc/passwd` contém informações essenciais sobre as contas de usuário, como UID, GID e diretório home. Embora tradicionalmente possua permissão de leitura global (modo 644), ao configurar permissão 777, qualquer usuário pode alterar ou apagar informações deste arquivo, causando falhas graves de autenticação ou inserção de contas maliciosas.

E. Criação de Usuários sem Senha

Como parte da configuração de um ambiente deliberadamente inseguro, foi criada uma conta de usuário sem senha, prática que contraria diretamente as recomendações das políticas de segurança da informação.

Esse cenário foi elaborado para simular uma vulnerabilidade comum em ambientes mal configurados ou negligenciados, onde contas sem senha ou com credenciais padrão frequentemente permanecem habilitadas após a instalação inicial de sistemas ou dispositivos. Tais contas, como `admin`, `guest` ou `root`, são amplamente documentadas em manuais técnicos e fóruns públicos, facilitando a ação de atacantes que exploram essas configurações para obter acesso não autorizado e, em seguida, escalar privilégios no ambiente comprometido.

A Fig. 4 ilustra um exemplo da configuração incorreta de usuário `guest`.



```

root@vlplab01: /home/dohko X + v
root@vlplab01:/home/dohko# sudo useradd guest
root@vlplab01:/home/dohko# sudo passwd -d guest
passwd: password changed.
root@vlplab01:/home/dohko# sudo mkdir /home/guest
root@vlplab01:/home/dohko# sudo chown guest:guest /home/guest/
root@vlplab01:/home/dohko# ls -la /home/guest
total 8
drwxr-xr-x 2 guest guest 4096 Mar 24 00:08 .
drwxr-xr-x 5 root root 4096 Mar 24 00:08 ..
root@vlplab01:/home/dohko#

```

Fig. 4 - Configuração do Usuário Guest sem Senha

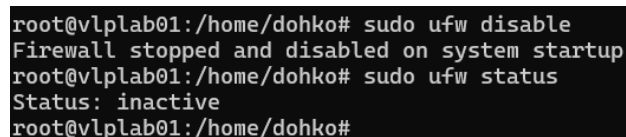
F. Desativação do Firewall

O firewall do sistema é uma das principais barreiras de defesa para controlar e filtrar o tráfego de rede. Neste ambiente de laboratório, o firewall foi desativado propositalmente para simular um cenário de rede exposta e

sem restrições, prática frequentemente observada em ambientes negligenciados ou mal configurados.

Assim, com essa ação, todas as portas e serviços que estejam escutando na rede permanecem acessíveis a qualquer origem externa, sem a aplicação de regras restritivas.

A Fig. 5 ilustra um exemplo de desativação do firewall padrão do Ubuntu Server, o UFW (Uncomplicated Firewall).



```

root@vlplab01:/home/dohko# sudo ufw disable
Firewall stopped and disabled on system startup
root@vlplab01:/home/dohko# sudo ufw status
Status: inactive
root@vlplab01:/home/dohko#

```

Fig. 5 - Desativação do Firewall UFW (Uncomplicated Firewall)

IV. RISCOS GLOBAIS DO AMBIENTE

A combinação das configurações inseguras realizadas durante a montagem do ambiente vulnerável gera um cenário de alto risco para qualquer sistema em operação real. A seguir, destacam-se os principais riscos associados ao conjunto de más práticas implementadas.

A. Comprometimento Total do Sistema

A presença de serviços desnecessários (como Telnet, FTP anônimo e rsh) somada à desativação do firewall e permissões excessivas em arquivos sensíveis cria uma superfície de ataque ampla. Isso facilita a exploração de vulnerabilidades conhecidas ou o abuso de funcionalidades para escalonamento de privilégios e execução remota de comandos.

B. Exposição de Dados Confidenciais

Permissões incorretas em arquivos críticos, como `/etc/shadow` e `/etc/passwd`, permitem que qualquer usuário local ou atacante com acesso remoto visualize ou copie informações sensíveis, como hashes de senhas. Isso facilita ataques de cracking offline e outras formas de comprometimento de credenciais.

C. Quebra de Confidencialidade e Integridade

Serviços configurados sem criptografia, como FTP e Telnet, transmitem dados em texto claro, expondo credenciais e outras informações trocadas entre cliente e servidor. Um atacante com capacidade de interceptação de rede (MITM) pode capturar logins e senhas com facilidade.

D. Facilitação de Ataques Automatizados

A ausência de um firewall e de políticas de senha robustas torna o ambiente altamente suscetível a ataques automatizados, como varreduras com bots e worms, que podem comprometer o sistema rapidamente, explorando portas abertas e serviços inseguros.

E. Conformidade e Auditoria

O ambiente simulado viola práticas recomendadas de normas como ISO/IEC 27001, CIS Benchmarks e NIST SP 800-53 [7], comprometendo a postura de conformidade e segurança de uma organização caso essas falhas estivessem presentes em um ambiente real.

F. Risco de Movimentação Lateral

Usuários sem senha e permissões permissivas em diretórios e arquivos críticos podem facilitar a movimentação lateral em um ambiente corporativo, permitindo que um atacante, após comprometer esta máquina, avance para outros sistemas na rede interna.

G. Impacto na Disponibilidade

Serviços não monitorados ou configurados inadequadamente (como Apache, FTP e Telnet) podem ser explorados em ataques de negação de serviço (DoS) ou DDoS, afetando a disponibilidade dos sistemas e serviços em produção.

V. TESTANDO AS VULNERABILIDADES NO UBUNTU SERVER COM KALI LINUX

A segurança de servidores e sistemas operacionais é uma das principais preocupações na administração de redes e infraestrutura de TI. Configurações inadequadas e práticas de administração ineficazes podem resultar em vulnerabilidades que permitam a exploração de sistemas por atacantes.

O teste é realizado utilizando a ferramenta Kali Linux, um sistema operacional de testes de penetração amplamente utilizado por profissionais de segurança.

A. Varredura de Portas

O nmap foi utilizado para realizar uma varredura de portas no servidor Ubuntu, a fim de identificar quais serviços estavam expostos. O comando utilizado foi: `nmap -sS -p-192.168.171.1/24`. Este comando realiza uma varredura

"stealth" (SYN Scan) em todas as portas TCP de todos os IPs na sub-rede 192.168.171.1/24, ajudando a identificar portas abertas e serviços em execução sem completar a conexão, o que torna a varredura mais discreta.

Ao escanear o IP 192.168.171.129, foram encontradas as seguintes portas abertas:

1. Porta 21/tcp (FTP): Serviço de transferência de arquivos, vulnerável se mal configurado ou sem criptografia (FTPS).
2. Porta 22/tcp (SSH): Serviço de acesso remoto seguro, mas com riscos se configurado de forma inadequada.
3. Porta 23/tcp (Telnet): Serviço obsoleto e inseguro para acesso remoto, transmitindo dados em texto claro.
4. Porta 80/tcp (HTTP): Serviço de servidor web, vulnerável se não utilizar HTTPS.
5. Portas 513/tcp (Login) e 514/tcp (Shell): Usadas para login remoto e execução de comandos, inseguros por não criptografar a comunicação.

A Fig. 6 ilustra um exemplo de varredura de porta com nmap no endereço IP 192.168.171.129 do servidor Ubuntu (vlab01).

```
Nmap scan report for 192.168.171.129
Host is up (0.00024s latency).
Not shown: 65529 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
513/tcp   open  login
514/tcp   open  shell
MAC Address: 00:0C:29:C9:B5:E1 (VMware)
```

Fig. 6 - Varredura de Portas com Nmap

B. Exploração HTTP

A abertura da porta 80/tcp, utilizada para o serviço HTTP, indica que o servidor está expondo um serviço web sem criptografia, o que é uma prática insegura. Esse serviço, por ser transmitido em texto claro, facilita a interceptação e a manipulação de dados por atacantes, caso não haja a devida proteção [8]. O uso da porta 80 sem criptografia (HTTPS) expõe o servidor a diversos riscos, como ataques de man-in-the-middle, onde os dados podem ser capturados ou alterados durante a transmissão.

A Fig. 7 apresenta um exemplo de serviço HTTP em execução no servidor vulnerável 192.168.171.129 (vlplab01).

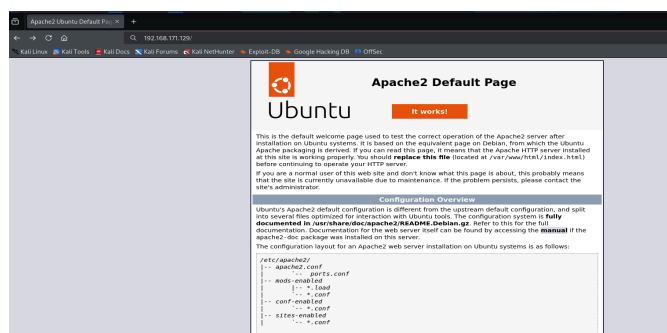


Fig. 7 - Execução de Serviço HTTP

Após a identificação do serviço HTTP na porta 80, um atacante pode realizar diversas explorações e varreduras para obter acesso ao ambiente interno, uma vez que o serviço pode ser vulnerável, especialmente por se tratar de um serviço desnecessário instalado no servidor.

C. Exploração de Conexão via Telnet com usuário Root

O Telnet é um protocolo de rede antigo e inseguro utilizado para acessar dispositivos remotamente, transmitindo dados em texto simples, o que o torna vulnerável a interceptações. Quando a porta Telnet está aberta, um atacante pode tentar se conectar diretamente ao serviço para obter acesso ao sistema. Para isso, pode-se utilizar o comando `telnet 192.168.171.129` em um terminal.

A Fig. 8 apresenta um exemplo de tentativa de conexão via Telnet no servidor vulnerável 192.168.171.129 (vlplab01).

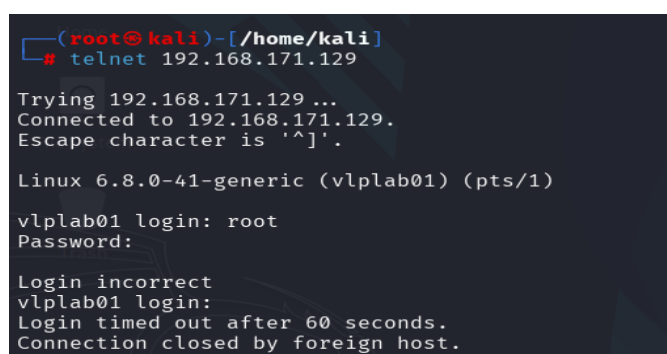


Fig. 8 - Tentativa de Conexão via Telnet

Ao estabelecer a conexão, o atacante será solicitado a fornecer credenciais de login. Se as credenciais forem fracas ou se o servidor permitir o login como root sem autenticação adequada, o atacante poderá obter acesso total ao sistema.

Neste caso, a senha do root no servidor 192.168.171.129 (vlplab01) é `P@ssword123`. Essa senha pode ser validada no Pwned Passwords, que a classifica como vulnerável, pois está presente na lista de vazamento de senhas, conforme mostrado na Fig. 9. A consulta pode ser realizada no site Have I Been Pwned [9].

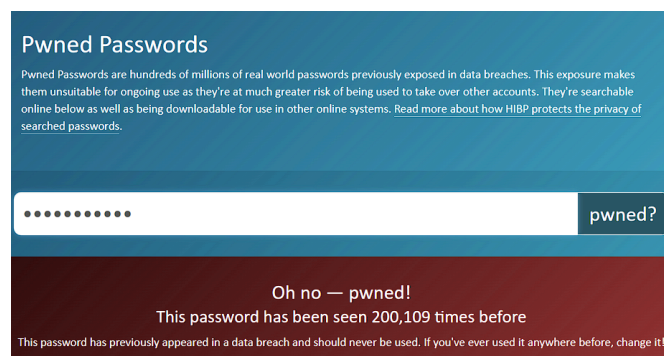


Fig. 9 - Consulta Realizada no Site Have I Been Pwned

Assim, o atacante pode obter acesso por meio de um ataque de força bruta. O ataque de força bruta consiste em tentar adivinhar a senha de acesso ao serviço, testando uma série de combinações possíveis até que a correta seja encontrada. No caso do Telnet, que não oferece criptografia e envia senhas em texto simples, esse tipo de ataque é relativamente fácil de realizar, especialmente se a senha for fraca ou simples.

Uma das ferramentas mais utilizadas para ataques de força bruta em serviços como Telnet é o Hydra. O Hydra permite realizar ataques de força bruta em diferentes protocolos, incluindo Telnet, de forma rápida e eficiente. Para realizar um ataque de força bruta com o Hydra, é necessário um arquivo de senhas (wordlist) e, opcionalmente, um arquivo de usuários. Caso queira atacar um usuário específico, como o root, pode-se utilizar a wordlist fornecida pelo Kali Linux, como o arquivo `rockyou.txt`, que contém uma lista extensa de senhas comuns [10]. O comando `hydra -l root -P /usr/share/wordlists/rockyou.txt telnet://192.168.171.129`, demonstra como realizar um ataque de força bruta com o Hydra. Onde:

1. `-l root`: Especifica o usuário alvo (nesse caso, root).
2. `-P /usr/share/wordlists/rockyou.txt`: Especifica a wordlist de senhas.
3. `telnet://192.168.171.129`: Define o serviço e o IP da máquina alvo.

Devido à simplicidade da senha definida para o usuário root, foi possível realizar a quebra da mesma utilizando um dicionário de senhas com a ferramenta Hydra, conforme ilustrado na Fig. 10.

```
(root@kali)~/home/kali
# hydra -l root -P /usr/share/wordlists/rockyou.txt telnet://192.168.171.129

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or se-
cret service organizations, or for illegal purposes (this is non-binding, these ** ignore l-
aws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-22 23:19:30
[WARNING] telnet is by its nature unreliable to analyze, if possible better choose FTP, SSH,
etc. if available
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from
a previous session found, to prevent overwriting. ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344402 login tries (l:1/p:14344402), ~
896526 tries per task
[DATA] attacking telnet://192.168.171.129:23/
[23][telnet] host: 192.168.171.129 login: root password: P@ssw0rd123
```

Fig. 10 - Brute force no Usuário Root

D. Exploração de Conexão via FTP

O File Transfer Protocol (FTP) é um protocolo utilizado para a transferência de arquivos entre sistemas em uma rede TCP/IP. No entanto, o FTP é um protocolo antigo e inseguro, pois transmite dados, incluindo credenciais de autenticação, em texto simples, tornando-o suscetível a ataques de interceptação e exploração [11].

Quando a porta padrão do FTP (21) está aberta em um servidor vulnerável, um atacante pode tentar estabelecer uma conexão direta para listar ou transferir arquivos sem a devida autenticação ou com credenciais fracas. Para isso, o atacante pode utilizar o cliente FTP nativo em sistemas Unix-like: *ftp 192.168.171.129*.

Após a tentativa de conexão, o servidor solicitará um nome de usuário e uma senha. Caso o serviço FTP permita o login anônimo (anonymous login), o atacante poderá ter acesso direto ao sistema de arquivos compartilhado, podendo listar, fazer download ou upload de arquivos críticos.

A Fig. 11 apresenta um exemplo de conexão via ftp com usuário anonymous no servidor ubuntu 192.168.171.129.

```
(root@kali)~/home/kali
# ftp 192.168.171.129
Connected to 192.168.171.129.
220 (vsFTPd 3.0.5)
Name (192.168.171.129:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Fig. 11 - Acesso com Usuário Anonymous

E. Exploração de Usuários sem Senha

A presença de contas de usuário sem senha configura uma falha crítica de segurança que pode ser facilmente explorada por atacantes. Essa situação é frequentemente observada em ambientes desatualizados ou mal gerenciados, nos quais administradores negligenciam a imposição de políticas mínimas de autenticação [12].

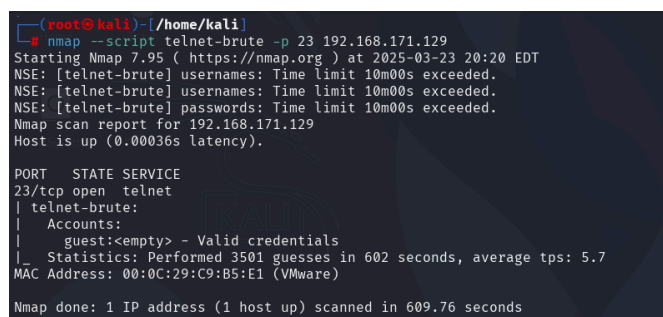
Contas sem senha reduzem drasticamente o nível de controle de acesso e permitem que qualquer agente malicioso com acesso à rede ou ao terminal consiga autenticar-se diretamente no sistema sem qualquer tipo de restrição. Em serviços de acesso remoto como Telnet, SSH ou FTP, a ausência de senha representa uma vulnerabilidade ainda mais grave, uma vez que tais serviços são frequentemente expostos à rede ou à internet.

Atacantes podem explorar essas contas utilizando scanners automatizados, como Nmap, que permitem detectar serviços acessíveis e listar contas sem autenticação. Uma vez dentro do sistema, o invasor poderá utilizar essas credenciais para realizar movimentações laterais, executar comandos arbitrários e buscar a elevação de privilégios, comprometendo todo o ambiente [8].

Além disso, contas sem senha são frequentemente associadas a ambientes de testes ou contas legadas esquecidas, o que demonstra a importância da manutenção de um processo contínuo de revisão de contas e permissões. Políticas de segurança recomendam o bloqueio ou remoção de contas sem senha, além da implementação de mecanismos de autenticação forte para prevenir esse tipo de exploração [12].

Uma técnica comum utilizada para descobrir usuários padrão sem senha em sistemas vulneráveis é a utilização de ferramentas de escaneamento de rede, como o Nmap. O Nmap pode ser configurado para realizar varreduras nos serviços de rede expostos e identificar potenciais vulnerabilidades, incluindo serviços que permitem a autenticação sem senha ou com credenciais padrão [8].

A Fig. 12 ilustra um exemplo de técnica que pode ser aplicada a serviços de rede, como o Telnet, para realizar uma tentativa de login utilizando um ataque de força bruta. O comando Nmap a seguir demonstra como o script `telnet-brute` pode ser utilizado para tentar várias combinações de senhas e usuários no serviço Telnet: `nmap --script telnet-brute -p 23 192.168.171.129`.



```
(root@kali) ~[/home/kali]
# nmap --script telnet-brute -p 23 192.168.171.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-23 20:20 EDT
NSE: [telnet-brute] usernames: Time limit 10m00s exceeded.
NSE: [telnet-brute] usernames: Time limit 10m00s exceeded.
NSE: [telnet-brute] passwords: Time limit 10m00s exceeded.
Nmap scan report for 192.168.171.129
Host is up (0.00036s latency).

PORT      STATE SERVICE
23/tcp    open  telnet
| telnet-brute:
|   Accounts:
|   | guest:<empty> - Valid credentials
|   | Statistics: Performed 3501 guesses in 602 seconds, average tps: 5.7
|   MAC Address: 00:0C:29:C9:B5:E1 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 609.76 seconds
```

Fig. 12 - Brute Force via Telnet com Nmap

Este script realiza um ataque de brute force simples, tentando senhas comuns para usuários padrão como root, admin, ou guest. Caso o sistema esteja configurado com uma conta padrão sem senha ou com senha fraca, o Nmap poderá identificar rapidamente a vulnerabilidade.

F. Exploração de Permissões Inseguras em Arquivos Críticos

A exploração de permissões inseguras em arquivos críticos é uma técnica comum utilizada por atacantes para obter acesso não autorizado a dados sensíveis e executar comandos prejudiciais. Arquivos críticos são aqueles que armazenam informações importantes ou funcionam como parte essencial do sistema, como arquivos de configuração, senhas e logs de sistema.

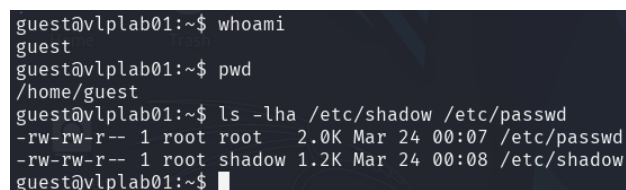
Quando as permissões de acesso a esses arquivos são configuradas de forma inadequada, qualquer usuário ou atacante com acesso ao sistema pode visualizar, modificar ou até mesmo excluir esses arquivos, comprometendo a segurança do ambiente. Esse tipo de falha de configuração é

frequentemente observado em sistemas mal administrados ou desatualizados, onde permissões excessivas ou inadequadas são concedidas a usuários comuns ou grupos não autorizados.

Permissões incorretas em arquivos críticos, como `/etc/shadow` e `/etc/passwd`, podem comprometer gravemente a segurança do sistema. Estes arquivos armazenam informações sensíveis, como os hashes das senhas dos usuários e detalhes relacionados à autenticação no sistema. Se as permissões desses arquivos não forem corretamente configuradas, qualquer usuário local ou atacante com acesso remoto poderá visualizar ou copiar esses dados.

Em um ambiente inseguro, onde a conta guest está presente e possui permissões inadequadas, um atacante pode explorar arquivos críticos do sistema, como `/etc/passwd` e `/etc/shadow`, para obter informações sensíveis, como hashes de senhas. explorar as permissões incorretas.

A Fig. 13 ilustra um exemplo da execução do comando `ls` para listar as permissões de arquivos críticos no sistema.



```
guest@vplab01:~$ whoami
guest
guest@vplab01:~$ pwd
/home/guest
guest@vplab01:~$ ls -lha /etc/shadow /etc/passwd
-rw-rw-r-- 1 root root 2.0K Mar 24 00:07 /etc/passwd
-rw-rw-r-- 1 root shadow 1.2K Mar 24 00:08 /etc/shadow
guest@vplab01:~$
```

Fig. 13 - Exemplo de acesso a arquivos críticos de sistema

VI. HARDENIZAÇÃO DO SISTEMA OPERACIONAL

Ambientes Unix/Linux são amplamente utilizados em servidores e infraestruturas críticas, mas configurações inseguras podem expor esses sistemas a ataques cibernéticos. Medidas de "hardening" são essenciais para reduzir a superfície de ataque e fortalecer a segurança operacional.

A. Remoção de Permissão de Acesso Direto como Root via SSH

O acesso direto ao usuário root via SSH representa um risco significativo à segurança, pois facilita a exploração de credenciais comprometidas, possibilitando a obtenção de controle total do sistema por atacantes [13].

Para mitigar este risco, recomenda-se a desativação do login direto como root por meio da configuração do serviço SSH. Isso pode ser realizado editando o arquivo de

configuração `/etc/ssh/sshd_config` e modificando a seguinte diretiva: `PermitRootLogin no`. Após a alteração, o serviço SSH deve ser reiniciado para que as novas configurações entrem em vigor: `sudo systemctl restart sshd`, conforme ilustrado na Fig. 14.

```
root@vplab01:/home/dohko# nano /etc/ssh/sshd_config
root@vplab01:/home/dohko# cat /etc/ssh/sshd_config | grep Root*
PermitRootLogin no
#PermitRootLogin prohibit-password
# the setting of "PermitRootLogin prohibit-password".
root@vplab01:/home/dohko# sudo systemctl restart sshd*
root@vplab01:/home/dohko#
```

Fig. 14 - Remoção de Permissão de Acesso Direto como Root via SSH

Além disso, é uma prática recomendada criar um usuário administrativo para acesso remoto e utilizar o comando `sudo` para a execução de tarefas privilegiadas. Adicionalmente, a implementação da autenticação baseada em chaves SSH em substituição ao uso de senhas reforça a segurança do acesso remoto ao sistema [13].

B. Ativação do Firewall

A ausência de um firewall ativo expõe os serviços da máquina à rede sem qualquer controle de acesso, aumentando significativamente a superfície de ataque e facilitando explorações por agentes mal-intencionados. Para mitigar este risco, recomenda-se a ativação e configuração de um firewall, como o Uncomplicated Firewall (UFW). A ativação pode ser realizada com o seguinte comando: `sudo ufw enable`, conforme ilustrado na Fig. 15.

```
root@vplab01:/home/dohko# ufw status
Status: inactive
root@vplab01:/home/dohko# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y/n)? y
Firewall is active and enabled on system startup
root@vplab01:/home/dohko# ufw status
Status: active
root@vplab01:/home/dohko#
```

Fig. 15 - Ativação do Firewall

Além disso, é essencial definir regras de filtragem adequadas para restringir acessos indesejados, permitindo apenas conexões legítimas e necessárias para a operação do sistema.

C. Remoção de Serviços Desnecessários Ativos

A execução de serviços desnecessários aumenta a superfície de ataque do sistema, tornando-o mais suscetível a

vulnerabilidades conhecidas. Para mitigar este risco, recomenda-se a identificação e desativação de serviços não essenciais.

A listagem dos serviços em execução pode ser realizada com o seguinte comando: `sudo systemctl list-units --type=service`, conforme ilustrado na Fig. 16.

```
root@vplab01:/home/dohko# sudo systemctl list-units --type=service
UNIT
apache2.service
apparmor.service
apport.service
blk-availability.service
console-setup.service
cron.service
dbus.service
finalrd.service
getty@tty1.service
inetutils-inetd.service
keyboard-setup.service
kmod-static-nodes.service
lvm2-monitor.service
ModemManager.service
multipathd.service
open-vm-tools.service
plymouth-quit-wait.service
plymouth-quit.service
plymouth-read-write.service
polkit.service
rsyslog.service
rwhod.service
setvtrgb.service
snapd.apparmor.service
snapd.seeded.service
ssh.service
```

Fig. 16 - Listagem dos Serviços em Execução

Assim, serviços desnecessários podem ser desativados e interrompidos da seguinte forma: `sudo systemctl disable apache2 && sudo systemctl stop apache2`, conforme ilustrado na Fig. 17.

```
root@vplab01:/home/dohko# sudo systemctl disable apache2 && sudo systemctl stop apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install disable apache2
root@vplab01:/home/dohko# systemctl daemon-reload
root@vplab01:/home/dohko# systemctl status apache2
o apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; preset: enabled)
   Active: inactive (dead)
   Docs: https://httpd.apache.org/docs/2.4/

Mar 25 23:19:39 vplab01 systemd[1]: Starting apache2.service - The Apache HTTP Server:
Mar 25 23:21:12 vplab01 apachectl[950]: AH00558: apache2: Could not reliably determine
Mar 25 23:21:12 vplab01 systemd[1]: Started apache2.service - The Apache HTTP Server
Mar 25 23:33:12 vplab01 systemd[1]: Stopping apache2.service - The Apache HTTP Server
Mar 25 23:33:12 vplab01 apachectl[1891]: AH00558: apache2: Could not reliably determine
Mar 25 23:33:12 vplab01 systemd[1]: apache2.service: Deactivated successfully.
Mar 25 23:33:12 vplab01 systemd[1]: Stopped apache2.service - The Apache HTTP Server
[lines 1-12/12 (END)]
root@vplab01:/home/dohko#
```

Fig. 17 - Desativação de Serviços Desnecessários

Além disso, é recomendável remover completamente serviços não utilizados para reduzir potenciais vetores de ataque: `sudo apt purge apache2 -y`, conforme ilustrado na Fig. 18.

```

root@vplab01:/home/dohko# sudo apt purge apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libaprutil1t64
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  apache2*
0 upgraded, 0 newly installed, 1 to remove and 224 not upgraded.
After this operation, 465 kB disk space will be freed.
(Reading database ... 85298 files and directories currently installed.)
Removing apache2 (2.4.58-1ubuntu8.4) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for ufw (0.36.2-6) ...
(Reading database ... 85248 files and directories currently installed.)
Purging configuration files for apache2 (2.4.58-1ubuntu8.4) ...
Processing triggers for ufw (0.36.2-6) ...
root@vplab01:/home/dohko#

```

Fig. 18 - Remoção de Serviços Desnecessários

Para serviços essenciais, devem ser aplicadas regras de firewall restritivas para evitar acessos indevidos. Essas práticas seguem as diretrizes estabelecidas por benchmarks de segurança, como o CIS Benchmark para sistemas Linux [15].

D. Definição de Política de Senha

Contas com senhas fracas são alvos comuns de ataques de força bruta e dicionário, comprometendo a segurança do sistema. Para mitigar esse risco, recomenda-se a implementação de políticas de senha seguras utilizando Pluggable Authentication Modules (PAM).

Se o módulo “pam_pwquality” não está instalado no seu sistema, você pode instalá-lo para poder usar as configurações de complexidade de senha. O módulo pam_pwquality está disponível no pacote *libpam-pwquality* no Ubuntu. Para instalá-lo, execute o seguinte comando: *sudo apt install libpam-pwquality*.

A configuração pode ser realizada editando o arquivo de políticas de senha: *sudo nano /etc/pam.d/common-password*. Substitua ou adicione a linha existente para incluir as opções conforme o seu requisito: *password requisite pam_pwquality.so minlen=12 dcredit=-1 ucredit=-1 ocredit=-1 lcredit=-1 maxrepeat=3 maxclassrepeat=4 dictcheck=1 retry=3*, conforme ilustrado na Fig. 19.

```

root@vplab01:/etc/pam.d# cat /etc/pam.d/common-password | grep pam_pwquality.so
password requisite pam_pwquality.so minlen=12 dcredit=-1 ucredit=-1 ocredit=-1 lcredit=-1 maxrepeat=3 maxclassrepeat=4 dictcheck=1 retry=3
root@vplab01:/etc/pam.d#

```

Fig. 19 - Definição de Política de Senha

Os seguintes requisitos mínimos devem ser definidos para fortalecer a complexidade das senhas [16], conforme mostrado na Tabela I.

TABELA I – REQUISITOS DE COMPLEXIDADE DE SENHA CONFIGURADOS NO PAM

Parâmetro	Valor	Descrição
Minlen	Comprimento mínimo da senha	12
Dcredit	Requer pelo menos um dígito	-1
Ucredit	Requer pelo menos uma letra maiuscula	-1
Ocredit	Requer pelo menos um caractere especial	-1
Lcredit	Requer pelo menos uma letra minúscula	-1
Maxrepeat	Limita repetições do mesmo caractere a no máximo 3	3
Maxclassrepeat	Evita a repetição excessiva de tipos de caracteres	4
Dictcheck	Bloqueia senhas que contenham palavras comuns do dicionário	1
Retry	Permite que o usuário tente 3 vezes antes de falhar	3

E. Diretórios Críticos com Permissões Inadequadas

Permissões inadequadas em diretórios críticos podem permitir acesso não autorizado a dados sensíveis, representando uma séria vulnerabilidade para a segurança do sistema. Para mitigar esse risco, é fundamental realizar verificações regulares nas permissões dos diretórios críticos.

A verificação das permissões pode ser realizada com o comando: *ls -ld /etc/shadow /etc/passwd*. Após a verificação,

é necessário ajustar as permissões para garantir um acesso restrito, conforme ilustrado na Fig. 20.

```
root@vplab01:/etc/security# ls -ld /etc/shadow /etc/passwd
-rw-rw-r-- 1 root root 1890 Mar 25 23:43 /etc/passwd
-rw-rw-r-- 1 root shadow 1096 Mar 25 23:43 /etc/shadow
root@vplab01:/etc/security# sudo chmod 600 /etc/shadow
root@vplab01:/etc/security# sudo chmod 644 /etc/passwd
root@vplab01:/etc/security# ls -ld /etc/shadow /etc/passwd
-rw-rw-r-- 1 root root 1890 Mar 25 23:43 /etc/passwd
-rw----- 1 root shadow 1096 Mar 25 23:43 /etc/shadow
root@vplab01:/etc/security#
```

Fig. 20 - Desativação de Serviços Desnecessários

Além disso, deve-se implementar o princípio do menor privilégio, garantindo que apenas usuários e serviços autorizados tenham permissões adequadas para acessar ou modificar diretórios críticos.

F. Remoção do Usuário Anonymous FTP

A presença do usuário *anonymous* no serviço FTP representa um risco de segurança, uma vez que pode permitir acessos não autenticados ao sistema. Para mitigar essa vulnerabilidade, recomenda-se a remoção ou desativação desta conta. No caso de servidores que utilizam o *vsftpd*, a desativação pode ser realizada editando o arquivo de configuração: *sudo nano /etc/vsftpd.conf*. No arquivo, deve-se localizar e modificar a seguinte linha: *anonymous_enable=NO*. Após a alteração, é necessário reiniciar o serviço para que as configurações tenham efeito: *sudo systemctl restart vsftpd*, conforme ilustrado na Fig. 21.

```
root@vplab01:/etc/security# sudo nano /etc/vsftpd.conf
root@vplab01:/etc/security# sudo systemctl restart vsftpd
root@vplab01:/etc/security# cat /etc/vsftpd.conf | grep anonymous*
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
# Uncomment this to allow the anonymous FTP user to upload files. This only
# Uncomment this if you want the anonymous FTP user to be able to create
# If you want, you can arrange for uploaded anonymous files to be owned by
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
root@vplab01:/etc/security#
```

Fig. 21 - Remoção do Usuário Anonymous FTP

Caso o servidor utilize outra implementação de FTP, recomenda-se consultar a documentação específica para desativar o acesso anônimo. Essa medida reduz significativamente a superfície de ataque, prevenindo acessos indevidos e potenciais exposições de arquivos sensíveis.

G. Atualizar o Sistema

Manter o sistema operacional atualizado é uma medida essencial para a segurança e estabilidade do ambiente. A aplicação regular de atualizações garante a correção de

vulnerabilidades, melhorias de desempenho e compatibilidade com novas tecnologias, conforme ilustrado na Fig. 22.

```
root@vplab01:/etc/security# sudo apt update && sudo apt upgrade -y
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libevent-2.1-7t64 tcpd
Use 'sudo apt autoremove' to remove them.
The following upgrades have been deferred due to phasing:
  ubuntu-drivers-common
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
root@vplab01:/etc/security#
```

Fig. 22 - Atualização do Sistema

Em distribuições baseadas em Debian, como Ubuntu, a atualização do sistema pode ser realizada com o seguinte comando: *sudo apt update && sudo apt upgrade -y*. O comando executa duas etapas fundamentais:

- *apt update*: Atualiza a lista de pacotes disponíveis, garantindo que o sistema tenha conhecimento das versões mais recentes dos repositórios.
- *apt upgrade -y*: Instala automaticamente todas as atualizações disponíveis para os pacotes já instalados.

É recomendável que essas atualizações sejam realizadas periodicamente, preferencialmente com verificações automatizadas ou mediante políticas de manutenção planejadas. Além disso, em ambientes críticos, a aplicação de patches deve ser precedida por testes para evitar incompatibilidades ou interrupções inesperadas nos serviços.

H. Definir Senha em usuários sem senha

Contas de usuário sem senha representam um risco significativo para a segurança do sistema, pois podem permitir acessos não autorizados. Para mitigar essa vulnerabilidade, é essencial identificar e definir senhas para essas contas.

A verificação de usuários sem senha pode ser realizada com o seguinte comando: *sudo cat /etc/shadow | awk -F: '(\$2=="") {print \$1}'*. O comando extrai os nomes de usuários cujas senhas não estão definidas no arquivo */etc/shadow*, conforme ilustrado na Fig. 23.

```

root@vlplab01:/etc/security# sudo cat /etc/shadow | awk -F: '{\$2==\"\"}{print \$1}'
teste_inseguro
guest
root@vlplab01:/etc/security#

```

Fig. 23 - Listagem de Usuários sem Senha

Caso sejam identificadas contas sem senha, recomenda-se definir uma senha segura utilizando o comando: *sudo passwd usuário*. Onde o *usuário* deve ser substituído pelo nome da conta correspondente. Após a execução, o sistema solicitará a definição de uma nova senha para o usuário especificado, conforme ilustrado na Fig. 24.

```

root@vlplab01:/home# sudo cat /etc/shadow | awk -F: '{\$2==\"\"}{print \$1}'
teste_inseguro
guest
root@vlplab01:/home# sudo passwd guest
New password:
BAD PASSWORD: The password is shorter than 12 characters
Retype new password:
Sorry, passwords do not match.
New password:
Retype new password:
passwd: password updated successfully
root@vlplab01:/home# sudo passwd teste_inseguro
New password:
Retype new password:
passwd: password updated successfully
root@vlplab01:/home# sudo cat /etc/shadow | awk -F: '{\$2==\"\"}{print \$1}'
root@vlplab01:/home#

```

Fig. 24 - Definição de Senha

Além disso, é recomendável configurar políticas de senha robustas, garantindo que novas credenciais atendam a requisitos mínimos de complexidade e expirem periodicamente para reforçar a segurança do ambiente.

I. Remoção de Ferramentas Administrativas Desnecessárias

A presença de ferramentas administrativas desnecessárias em um sistema operacional pode representar riscos de segurança, pois podem ser exploradas por atacantes para reconhecimento, movimentação lateral ou escalonamento de privilégios. Portanto, é recomendável remover ou restringir o uso dessas ferramentas sempre que não forem essenciais para o funcionamento do ambiente.

Dentre as ferramentas que devem ser avaliadas e removidas quando não forem estritamente necessárias, destacam-se:

- *nmap* – Ferramenta de varredura de rede que pode ser utilizada para mapear portas abertas e serviços expostos.
- *netcat* – Utilitário que permite conexões arbitrárias via TCP e UDP, podendo ser usado para tunneling ou execução remota de comandos.

- *tcpdump* – Capturador de pacotes de rede que pode expor tráfego sensível a usuários não autorizados.
- *inetd* – Super servidor que gerencia serviços baseados em rede; o uso de alternativas mais seguras, como *systemd* ou *xinetd*, é preferível.
- *telnet* – Protocolo inseguro de comunicação remota que transmite credenciais em texto claro. Deve ser substituído por SSH (Secure Shell).
- *rwho* – Serviço que exibe informações sobre usuários logados na rede, podendo expor dados sensíveis sobre a estrutura do sistema.

A remoção dessas ferramentas pode ser realizada com o seguinte comando: *sudo apt remove --purge nmap netcat tcpdump inetutils-inetd telnet rwho -y*. Após a remoção, recomenda-se a execução de uma verificação para garantir que os pacotes foram completamente desinstalados: *dpkg -l | grep -E 'nmap|netcat|tcpdump|inetd|telnet|rwho'*, conforme ilustrado na Fig. 25.

```

root@vlplab01:/home/dohko# dpkg -l | grep -E 'nmap|netcat|tcpdump|inetd|inetutils-telnet|telnet|rwho'
root@vlplab01:/home/dohko# dpkg -l | grep -E 'nmap|netcat|tcpdump|inetd|inetutils-telnet|telnet|rwho'
root@vlplab01:/home/dohko#

```

Fig. 25 - Remoção de Ferramentas Administrativas Desnecessárias

Além da remoção, é importante reforçar as configurações de controle de acesso para evitar a instalação não autorizada dessas ferramentas. A aplicação de políticas como "whitelisting" e "hardening" do sistema reduz a superfície de ataque e fortalece a segurança do ambiente.

J. Configuração do SSH

O SSH é um protocolo de rede amplamente utilizado para garantir comunicações seguras entre sistemas em redes não confiáveis. Sua principal aplicação é o acesso remoto a servidores, proporcionando um canal seguro para a execução de comandos e a transferência de arquivos.

No contexto do sistema operacional Ubuntu, a instalação e a configuração do SSH são processos simples, mas cruciais para manter a segurança da infraestrutura de TI. O OpenSSH Server pode ser instalado com o seguinte comando: *sudo apt install openssh-server*. A instalação do OpenSSH Server permite que a máquina Ubuntu aceite conexões SSH.

Após a instalação, é essencial verificar se o serviço SSH está em funcionamento. Isso pode ser feito utilizando o

comando: `sudo systemctl status ssh`. Se o serviço não estiver em execução, ele pode ser iniciado manualmente com: `sudo systemctl start ssh`. Além disso, para garantir que o SSH inicie automaticamente após a reinicialização do sistema, o comando a ser utilizado é: `sudo systemctl enable ssh`, conforme ilustrado na Fig. 26.

```
root@vplab01: /home# sudo systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-03-26 00:14:26 UTC; 26min ago
   TriggeredBy: ● ssh.socket
   Docs: man:sshd(8)
         man:sshd_config(5)
   Main PID: 1548 (sshd)
     Tasks: 1 (limit: 4552)
    Memory: 3.1M (peak: 4.3M)
       CPU: 46ms
   CGroup: /system.slice/ssh.service
           └─1548 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

Fig. 26 - Execução do Serviço SSH

Se o sistema estiver utilizando o firewall `ufw` (Uncomplicated Firewall), será necessário configurar as regras para permitir conexões SSH. Para liberar a porta padrão do SSH (porta 22), o seguinte comando deve ser executado: `sudo ufw allow ssh`, conforme ilustrado na Fig. 27.

```
root@vplab01: /home# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To Action From
--
22/tcp ALLOW IN Anywhere
22/tcp (v6) ALLOW IN Anywhere (v6)

root@vplab01: /home#
```

Fig. 27 - Regras de Firewall para Conexão via SSH

Caso o servidor utilize uma porta personalizada para SSH, a regra pode ser ajustada para refletir essa alteração. Por exemplo, se o SSH estiver configurado para a porta 2222, o comando seria: `sudo ufw allow 2222/tcp`. Após configurar as regras do firewall, o comando abaixo pode ser utilizado para ativar o firewall, caso ainda não tenha sido feito: `sudo ufw enable`.

VII. VALIDAÇÃO DO "HARDENING" APLICADO

A validação do "hardening" aplicado visa garantir que as medidas de segurança implementadas no sistema estejam sendo eficazes na proteção contra vulnerabilidades. O processo de validação envolve a execução de uma série de testes e verificações para garantir que as configurações e práticas de segurança estão devidamente aplicadas. A seguir, são apresentadas as principais áreas a serem verificadas

durante o processo de validação do "hardening" em um sistema Ubuntu.

A. Varredura de Portas

A varredura de portas é uma técnica fundamental para identificar quais portas de rede estão abertas em um sistema. Esta técnica ajuda a garantir que apenas as portas necessárias estão acessíveis, minimizando a exposição a ataques de rede. A ferramenta Nmap é comumente utilizada para realizar a varredura de portas, por meio do comando: `nmap -sS -O 192.168.171.129`.

Após a execução da varredura, é importante que apenas as portas essenciais, como a porta 22 para SSH, estejam abertas. Outras portas, que não são necessárias, devem ser fechadas, conforme ilustrado na Fig. 28.

```
root@kali: ~# nmap -sS -O 192.168.171.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-25 20:57 EDT
Nmap scan report for 192.168.171.129
Host is up (0.00036s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|router|storage-misc
Running (JUST GUESSING): Linux 4.X|5.X|6.X|2.6.X|3.X (97%), MikroTik RouterOS 7.X (97%), Synology DiskStation Manager 5.X (89%)
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3 cpe:/o:linux:linux_kernel:6.0 cpe:/o:linux:linux_kernel:2.6.32 cpe:/o:linux:linux_kernel:3 cpe:/o:synology:diskstation_manager:5.2
Aggressive OS guesses: Linux 4.15 - 5.19 (97%), Linux 4.19 (97%), Linux 5.0 - 5.14 (97%), OpenWrt 21.02 (Linux 5.4) (97%), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3) (97%), Linux 6.0 (94%), Linux 5.4 - 5.10 (91%), Linux 2.6.32 (91%), Linux 2.6.32 - 3.13 (91%), Linux 3.10 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.24 seconds
```

Fig. 28 - Varredura de Portas com Nmap

B. Exploração HTTP

A exploração HTTP visa garantir que o servidor web esteja adequadamente configurado para evitar vulnerabilidades e a exposição de informações sensíveis. Em um ambiente de hardening, o ideal é minimizar a superfície de ataque removendo pacotes e serviços desnecessários. Neste caso, o pacote Apache foi removido, uma vez que não era necessário, reduzindo a superfície de ataque ao desabilitar a exposição de serviços HTTP desnecessários.

Para verificar se as portas HTTP (porta 80) e HTTPS (porta 443) estão fechadas, você pode utilizar o Nmap com o seguinte comando: `nmap -p 80,443 192.168.17.129`. Esse comando realiza a varredura nas portas 80 e 443, permitindo verificar se os serviços HTTP e HTTPS estão abertos ou fechados no endereço IP especificado, conforme ilustrado na Fig. 29.


```
(root@kali)-[/home/kali]
# nmap -p 80,443 192.168.17.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-25 21:03 EDT
Nmap scan report for 192.168.17.129
Host is up (0.00079s latency).

PORT      STATE      SERVICE
80/tcp    filtered  http
443/tcp   filtered  https

Nmap done: 1 IP address (1 host up) scanned in 7.83 seconds

(root@kali)-[/home/kali]
```

Fig. 29 - Varredura de Porta HTTP(s) com Nmap

C. Exploração de Conexão via Telnet com Usuário Root

A exploração de conexões via Telnet com o usuário root busca verificar se o sistema permite o acesso remoto de forma insegura e com privilégios elevados, o que pode representar um risco significativo. No entanto, após a implementação de “hardening”, a porta Telnet foi fechada, uma vez que o Telnet transmite informações em texto claro, tornando-se uma prática insegura para conexões remotas. A recomendação é utilizar o SSH (Secure Shell), que oferece criptografia e maior segurança.

Porém, é importante garantir que o SSH esteja configurado de forma segura, impedindo o acesso direto ao usuário root. Em um ambiente de hardening, a prática recomendada é desabilitar o login direto via SSH para o usuário root, forçando a autenticação por um usuário não privilegiado e escalonamento de privilégios apenas quando necessário.

```
Connection to 192.168.171.129 closed.
PS C:\Users\event> ssh root@192.168.171.129
root@192.168.171.129's password:
Permission denied, please try again.
root@192.168.171.129's password:
Permission denied, please try again.
root@192.168.171.129's password:
root@192.168.171.129: Permission denied (publickey,password).
PS C:\Users\event>
```

Fig. 30 - Tentativa de Conexão via Ssh com Root Direto

Para verificar se o acesso SSH está permitindo o login com o usuário root, você pode consultar o arquivo de configuração do SSH: `cat /etc/ssh/sshd_config | grep PermitRootLogin`. Se a diretiva `PermitRootLogin` estiver configurada como `No`, significa que o acesso ao SSH com o usuário root não está permitido, conforme ilustrado na Fig. 31.

```
root@vplab01:/home/dohko# nano /etc/ssh/sshd_config
root@vplab01:/home/dohko# cat /etc/ssh/sshd_config | grep Root*
PermitRootLogin no
#PermitRootLogin prohibit-password
# the setting of "PermitRootLogin prohibit-password".
root@vplab01:/home/dohko# sudo systemctl restart sshd*
root@vplab01:/home/dohko#
```

Fig. 31 - Tentativa de Conexão via Ssh com Root Direto

D. Exploração de Conexão via FTP com Anonymous

A exploração de conexões via FTP com acesso anônimo visa verificar se o serviço FTP está configurado de maneira insegura, permitindo que qualquer usuário se conecte ao servidor sem autenticação. O FTP (File Transfer Protocol) transmite dados em texto claro, o que aumenta o risco de interceptação de informações sensíveis, como credenciais de login, se o serviço não for adequadamente configurado.

Neste caso, para realizar o teste de acesso anônimo, a abordagem será tentar realizar a conexão via SFTP, que é mais seguro e também pode ser configurado para permitir o acesso anônimo, se desejado. No entanto, no ambiente de hardening, o serviço FTP foi fechado para minimizar a superfície de ataque, conforme ilustrado na Fig. 32.

```
(root@kali)-[/home/kali]
# nmap -p 21 192.168.17.129
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-25 21:16 EDT
Nmap scan report for 192.168.17.129
Host is up (0.00055s latency).

PORT      STATE      SERVICE
21/tcp    filtered  ftp

Nmap done: 1 IP address (1 host up) scanned in 6.86 seconds

(root@kali)-[/home/kali]
```

Fig. 32 - Varredura de Porta FTP com Nmap

Embora o serviço FTP esteja desabilitado, o SFTP, que é criptografado e mais seguro, pode ser utilizado para transferências de arquivos. A ideia é verificar se o acesso anônimo está habilitado no SFTP, o que também pode ser um risco de segurança caso mal configurado. Para acessar o servidor de forma anônima. No terminal, execute o comando: `sftp anonymous@192.168.17.129`. Se o servidor permitir a conexão sem exigir credenciais, significa que o acesso anônimo está habilitado no SFTP. Caso contrário, o servidor solicitará autenticação, o que indica que o acesso anônimo foi desabilitado, conforme ilustrado na Fig. 33.

```
PS C:\Users\event> sftp anonymous@192.168.171.129
anonymous@192.168.171.129's password:
Permission denied, please try again.
anonymous@192.168.171.129's password:
Connection closed
PS C:\Users\event>
```

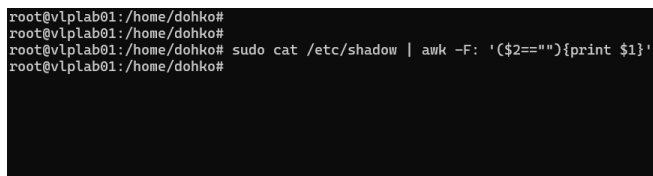
Fig. 33 - Tentativa de Conexão via SFTP com Anonymous

E. Exploração de Usuários sem Senha

A exploração de usuários sem senha visa identificar se existem contas no sistema que não exigem autenticação para acesso, o que representa um risco significativo de segurança.

Contas sem senha podem ser utilizadas por atacantes para obter acesso não autorizado ao sistema, caso encontrem uma forma de explorar essas contas.

No ambiente de "hardening" aplicado, todas as contas sem senha foram removidas ou configuradas para exigir autenticação adequada. Assim, não existem mais usuários sem senha no sistema, garantindo que o acesso a qualquer conta seja feito de forma controlada e com a devida validação de credenciais, conforme ilustrado na Fig. 34.



```

root@vplab01:/home/dohko#
root@vplab01:/home/dohko#
root@vplab01:/home/dohko# sudo cat /etc/shadow | awk -F: '($2=="") {print $1}'
root@vplab01:/home/dohko#

```

Fig. 34 - Listagem de Usuários sem Senha

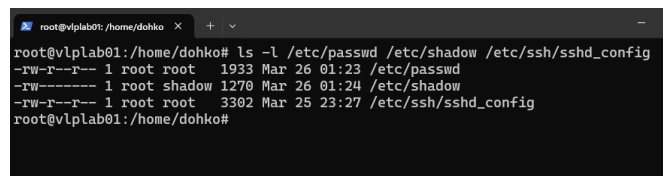
A aplicação de políticas rigorosas de segurança durante o processo de "hardening" foi crucial para a eliminação de contas sem senha e para a minimização da superfície de ataque, conforme as boas práticas de segurança.

F. Exploração de Permissões Inseguras em Arquivos Críticos

A exploração de permissões inseguras em arquivos críticos refere-se à verificação de arquivos sensíveis ou essenciais do sistema que possuem permissões excessivas, permitindo acesso ou modificação não autorizada. Arquivos críticos incluem configurações do sistema, arquivos de log, chaves privadas, senhas e outros dados sensíveis. Permissões inadequadas nestes arquivos podem ser exploradas por atacantes, comprometendo a segurança do sistema.

Após a aplicação do "hardening" no sistema, todas as permissões inseguras em arquivos críticos foram removidas ou ajustadas para garantir o princípio do mínimo privilégio. Arquivos sensíveis, como os de configuração e chaves privadas, passaram a ter permissões restritas, permitindo acesso apenas a usuários e grupos específicos, como o root.

Essas ações minimizam a superfície de ataque, impedindo acessos não autorizados e protegendo a integridade do sistema. Como resultado, não há mais permissões inseguras em arquivos críticos, conforme ilustrado na Fig. 35.



```

root@vplab01:/home/dohko#
root@vplab01:/home/dohko# ls -l /etc/passwd /etc/shadow /etc/ssh/sshd_config
-rw-r--r-- 1 root root 1933 Mar 26 01:23 /etc/passwd
-rw----- 1 root shadow 1270 Mar 26 01:24 /etc/shadow
-rw-r--r-- 1 root root 3302 Mar 25 23:27 /etc/ssh/sshd_config
root@vplab01:/home/dohko#

```

Fig. 35 - Listagem de Arquivos Críticos com Permissões Inseguras

VIII. CONCLUSÃO

Este trabalho apresentou um estudo detalhado sobre o processo de "hardening" em sistemas operacionais Linux, com foco no Ubuntu Server. Foram simuladas vulnerabilidades comuns em ambientes não protegidos, como a ativação de serviços desnecessários, permissões inadequadas em arquivos críticos e a desativação de mecanismos de defesa, evidenciando as falhas de segurança que poderiam ser exploradas por um atacante. O estudo abordou, de forma prática, as técnicas de "hardening" aplicadas a essas falhas, com base em diretrizes reconhecidas internacionalmente, como as normas ISO/IEC 27002 [4] e benchmarks do CIS [5].

A aplicação dessas técnicas mostrou-se eficaz na mitigação dos riscos, destacando a importância de medidas como a remoção de serviços desnecessários, a configuração segura do SSH, a ativação do firewall, e a implementação de políticas de senha e permissões adequadas. A validação das melhorias, por meio de varreduras e tentativas de exploração, comprovou que o sistema, antes vulnerável, passou a atender a padrões elevados de segurança, protegendo os dados quanto à sua integridade, confidencialidade e disponibilidade.

A pesquisa reforça a importância do "hardening" como uma prática contínua e fundamental para a segurança dos sistemas, alinhando-os às melhores práticas de conformidade e auditoria. A implementação dessas medidas oferece uma defesa sólida contra ameaças cibernéticas e pode ser expandida por meio de técnicas avançadas de monitoramento e resposta a incidentes, visando um ciclo de segurança completo.

AGRADECIMENTOS

Gostaria de expressar meus sinceros agradecimentos a todos os docentes da Pós-Graduação em Segurança da Informação e Inteligência Defensiva da CECyber, cuja

contribuição foi essencial para a realização deste trabalho. Em especial, dedico minha profunda gratidão ao professor Eduardo R. Sant'Ana Popovici, cuja orientação nas disciplinas de Cloud & Mobile Security foi crucial para o desenvolvimento desta pesquisa. Seus conhecimentos e ensinamentos não só embasaram este estudo, mas também enriqueceram minha trajetória acadêmica.

Este trabalho só foi possível graças à dedicação e ao empenho de todos os professores envolvidos, aos quais sou imensamente grato.

REFERÊNCIAS

- [1] Canonical Ltd., "Ubuntu Server Documentation," [Online]. Disponível em: <https://ubuntu.com/server/docs>.
- [2] Red Hat, "Security Guide - Red Hat Enterprise Linux," [Online]. Disponível em: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/security_hardening.
- [3] J. R. Vacca, Computer and Information Security Handbook, 3rd ed., Academic Press, 2017.
- [4] International Organization for Standardization, "ISO/IEC 27002:2022 - Information security, cybersecurity and privacy protection — Controls," 2022.
- [5] Center for Internet Security (CIS), "CIS Benchmark for Linux," 2022. Disponível em: <https://www.cisecurity.org>. Acesso em: mar. 2025.
- [6] Kali Linux, "Kali Linux Documentation," Offensive Security, 2025. [Online]. Disponível em: <https://www.kali.org/docs/>. Acesso em: mar. 2025.
- [7] NIST, "Security and Privacy Controls for Information Systems and Organizations," NIST Special Publication 800-53 Revision 5, Update 1, National Institute of Standards and Technology, 2025. [Online]. Disponível em: <https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final>. Acesso em: mar. 2025.
- [8] Nmap Project, "Nmap Scripting Engine," Nmap Documentation, 2023. [Online]. Disponível em: <https://nmap.org/book/nse.html>. Acesso em: mar. 2025.
- [9] "Pwned Passwords," Have I Been Pwned, [Online]. Disponível em: <https://haveibeenpwned.com/Passwords>. Acesso em: mar. 2025.
- [10] "Hydra - Network Logon Cracker," Kali Linux Tools, [Online]. Disponível em: <https://tools.kali.org/password-attacks/hydra>. Acesso em: mar. 2025.
- [11] J. Postel and J. Reynolds, "File Transfer Protocol (FTP)," RFC 959, Oct. 1985. [Online]. Disponível em: <https://www.rfc-editor.org/rfc/rfc959.txt>. Acesso em: mar. 2025.
- [12] OWASP, "Authentication Cheat Sheet," OWASP Foundation, 2023. [Online]. Disponível em: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html. Acesso em: mar. 2025.
- [13] T. Ylonen, and C. Lonvick, "Secure Shell (SSH) Protocol Architecture," RFC 4251, Network Working Group, Cisco Systems, Inc., SSH Communications Security Corp., Jan. 2006. [Online]. Disponível em: <https://doi.org/10.17487/RFC4251>.