Segurança da Informação

Alexandre Neves, Felipe Fidelis

3 de novembro de 2025

1 Introdução

A segurança da informação tornou-se um dos pilares fundamentais para a operação de sistemas computacionais em ambientes acadêmicos, corporativos e governamentais. Com o aumento exponencial de ataques cibernéticos e a sofisticação das técnicas utilizadas por invasores, a implementação de medidas preventivas e defensivas deixou de ser opcional para se tornar essencial.

No contexto de infraestruturas de rede, o protocolo SSH (*Secure Shell*) é amplamente utilizado para acesso remoto seguro a servidores e dispositivos de rede. Entretanto, configurações padrão inadequadas, práticas inseguras de gerenciamento de credenciais e a ausência de proteções complementares tornam os serviços SSH vulneráveis a diversos vetores de ataque, incluindo força bruta, enumeração de usuários, roubo de chaves privadas e manipulação de logs.

Este trabalho apresenta um estudo abrangente sobre **hardening SSH** (enrijecimento de segurança), demonstrando através de testes práticos em ambiente virtualizado a diferença entre sistemas vulneráveis e sistemas protegidos. Utilizando a metodologia de *red team vs blue team*, foram desenvolvidos e executados oito ataques distintos que cobrem as principais fases da cadeia de ataque (*cyber kill chain*): reconhecimento, acesso inicial, descoberta de ativos, acesso a credenciais, movimento lateral e evasão de defesas.

1.1 Objetivos

Os objetivos deste trabalho são:

1. **Implementar técnicas de hardening SSH** em ambiente controlado, aplicando configurações de segurança recomendadas por organizações como NIST, CIS e OWASP;

2. **Simular ataques reais** utilizando ferramentas e scripts desenvolvidos especificamente para este estudo, cobrindo vetores de ataque como força bruta, enumeração de usuários, roubo de chaves SSH e manipulação de logs;

3. **Comparar quantitativamente** a eficácia das proteções implementadas, medindo taxas de sucesso de ataques em sistemas vulneráveis versus sistemas protegidos;

4. **Documentar boas práticas** de segurança aplicáveis a ambientes de produção, incluindo políticas de uso, configurações de firewall, detecção de intrusões e auditoria de logs;

5. **Contribuir para a formação de profissionais** de segurança da informação através da demonstração prática de técnicas defensivas e ofensivas.

1.2 Justificativa

Ataques direcionados a serviços SSH representam uma das principais ameaças à segurança de servidores corporativos, VPS (*Virtual Private Servers*) em nuvem e infraestruturas críticas. Segundo dados do SANS Institute, tentativas de força bruta SSH são uma das técnicas mais utilizadas por atacantes, com milhões de tentativas diárias detectadas em honeypots distribuídos globalmente.

A motivação para este estudo surge da necessidade de:

- **Reduzir a superfície de ataque** de sistemas que utilizam SSH como principal meio de acesso remoto;

- **Demonstrar de forma prática** o impacto de configurações inseguras através de testes de penetração controlados;

- **Estabelecer políticas de segurança** aplicáveis a laboratórios de informática em instituições de ensino;

- **Prevenir comprometimento de infraestruturas cloud** através de proteção de chaves SSH privadas.

1.3 Metodologia

A metodologia adotada neste trabalho consiste em:

1. **Construção de Ambiente Virtualizado:** Criação de três máquinas virtuais usando Vagrant e VirtualBox, simulando um cenário de ataque realista:
   - **VM Atacante** (192.168.56.20): origem dos ataques, equipada com ferramentas de pentest;
   - **VM Alvo - Vulnerável** (192.168.56.10): sistema com configurações padrão inseguras;
   - **VM Alvo - Hardened** (192.168.56.11): sistema com hardening SSH completo aplicado.
2. **Desenvolvimento de Scripts de Ataque:** Criação de cinco scripts Bash para simular ataques reais:
   - Reconhecimento SSH e mapeamento de rede;
   - Enumeração de usuários via SSH;
   - Força bruta e teste de proteções;
   - Manipulação e evasão de logs;
   - Roubo de chaves SSH privadas e movimento lateral.
3. **Implementação de Hardening:** Aplicação de configurações de segurança incluindo:
   - Hardening do serviço SSH (`sshd_config`);
   - Configuração de firewall UFW;
   - Instalação e configuração de Fail2ban;
   - Proteção de logs com atributo imutável (`chattr +i`);
   - Uso de chaves SSH protegidas por senha forte.

4. **Execução de Testes e Coleta de Métricas:** Execução dos ataques em ambos os alvos, documentando:
    - Taxa de sucesso/falha de cada ataque;
    - Quantidade de informações vazadas;
    - Tempo de detecção e bloqueio (fail2ban);
    - Logs gerados e evidências forenses.
5. **Análise Comparativa:** Elaboração de tabelas, gráficos e análises qualitativas demonstrando a eficácia das proteções implementadas.

1.4 Estrutura do Documento

Este documento está organizado da seguinte forma:

- **Seção 2:** Política de Segurança e Uso dos Computadores dos Laboratórios — estabelece diretrizes para uso seguro de recursos computacionais em ambiente acadêmico;
- **Seção 3:** Implementação de Hardening SSH — descreve configurações técnicas aplicadas no sistema protegido;
- **Seção 4:** Testes de Validação das Proteções — apresenta testes de varredura de portas, verificação de status e auditoria de configurações;
- **Seção 5:** Execução dos Ataques e Resultados — detalha oito ataques executados, resultados obtidos e análise de impacto;
- **Seção 6:** Conclusões Finais e Recomendações — resume lições aprendidas e melhores práticas para ambientes de produção.

2 Política de Segurança e Uso dos Computadores dos Laboratórios

2.1 Objetivo

Estabelecer diretrizes e responsabilidades para garantir a segurança, integridade, disponibilidade e uso adequado dos recursos de informática (hardware, software e rede) presentes nos laboratórios do Instituto Federal Goiano - Campus Ceres, prevenindo acessos não autorizados, danos e mau uso.

2.2 Âmbito de Aplicação

Esta política aplica-se a todos os usuários (estudantes, professores, pesquisadores e técnicos) que utilizam os computadores localizados nos laboratórios da instituição.

2.3 Diretrizes Gerais de Uso (Para Todos os Usuários)

2.3.1 Uso de Contas de Acesso

1. **Contas Individuais e Não Compartilháveis:** Cada usuário (aluno e professor) deve utilizar sua conta de acesso pessoal e intransferível. **É estritamente proibido compartilhar senhas ou utilizar contas de terceiros.**

2. **Senhas Fortes:** As senhas devem seguir os requisitos mínimos de complexidade definidos pela instituição (ex: mínimo de 8 caracteres, com letras maiúsculas, minúsculas, números e símbolos).

3. **Logout e Bloqueio:** Os usuários devem sempre efetuar *logout* ou bloquear a estação de trabalho ao se ausentarem, mesmo que por um breve período.

4. **Monitoramento:** A instituição reserva-se o direito de monitorar o uso dos equipamentos para fins de manutenção e segurança, conforme a legislação vigente.

2.4 Software e Configurações

1. **Instalação e Configuração de Ambiente de Desenvolvimento (Sistemas Multiusuários):** É proibida a modificação ou remoção de qualquer software, aplicativo ou arquivo executável que faça parte do sistema operacional base. **EXCEÇÃO - Configuração de Ambiente:** Nos laboratórios dos cursos de Sistemas de Informação, Informática para Internet e Inteligência Artificial, é permitido que o aluno realize a configuração do seu ambiente de desenvolvimento pessoal (*personal environment*) para fins curriculares, **desde que**:

   (a) A instalação de bibliotecas, pacotes e *frameworks* seja feita utilizando **gerenciadores de pacotes com escopo local** (ex: `pip`

`-user`, `npm` em modo local, gerenciadores de ambientes virtuais como `conda` ou `venv`).

(b) A configuração de variáveis de ambiente, como o `PATH`, seja feita estritamente através dos arquivos de configuração da *shell* no diretório `home` do usuário.

(c) **Não seja utilizada a elevação de privilégios de administrador** (`sudo` ou `root`) para qualquer instalação ou configuração.

2. **Uso de Contêineres e Virtualização:** Para tarefas que exijam contêineres (*Docker, Podman*), o aluno deve, prioritariamente, utilizar soluções *rootless* (que não requerem acesso de administrador) ou ambientes de desenvolvimento pré-configurados e aprovados pela TI. A concessão de acesso ao grupo `docker` é proibida por representar risco de segurança à máquina hospedeira.

3. **Downloads Não Autorizados:** É proibido o download e/ou armazenamento de conteúdo ilegal, malicioso (*vírus, malware*), pornográfico ou que viole direitos autorais.

4. **Alteração de Configurações:** É proibida a alteração de configurações de sistema, rede, papel de parede, *screensaver* ou qualquer ajuste que comprometa o padrão operacional da máquina.

2.4.1 Uso da Rede e Internet

1. **Acesso Remoto (*SSH, VNC, RDP*):** O acesso remoto entre estações de alunos é estritamente proibido. O uso de protocolos de acesso remoto para fins acadêmicos ou de pesquisa deve ser formalmente solicitado e limitado a servidores específicos da instituição, conforme as regras de *firewall* e segurança.

2. **Comportamento Ético:** É proibido utilizar a rede para fins que violem a lei, promovam *hacking, phishing* ou qualquer atividade que cause prejuízo à instituição ou a terceiros.

2.5 Diretrizes Específicas para Alunos

1. **Uso Exclusivo para Fins Acadêmicos:** Os computadores dos laboratórios destinam-se primariamente a atividades de ensino, pesquisa

e extensão. O uso pessoal excessivo (jogos, redes sociais, *streaming*) pode ser restringido.

2. **Armazenamento Temporário:** Arquivos pessoais devem ser salvos em serviços de armazenamento em nuvem da instituição (se disponíveis) ou em mídias externas. A instituição não se responsabiliza por arquivos salvos no disco local, que podem ser apagados a qualquer momento (ex: no *reboot* da máquina).

2.6 Diretrizes Específicas para Professores/Docentes

2.6.1 Segurança da Estação Docente (Mesa do Professor)

1. **Desativação de Serviços Desnecessários:** O serviço **SSH (ou qualquer outro serviço de acesso remoto como VNC ou RDP)** deve ser **desativado** na estação do professor por padrão. Ele só poderá ser ativado temporariamente para propósitos didáticos específicos, e deve ser desativado imediatamente após o uso.

2. **Firewall Rigoroso:** O *firewall* da estação docente deve estar sempre ativo e configurado para **bloquear todas as conexões de entrada**, exceto aquelas absolutamente necessárias para o funcionamento em sala de aula (ex: projeção de tela).

3. **Contas de Usuário:** O professor deve utilizar uma **conta de usuário padrão (não administrador)** para as aulas, reservando a conta de administrador para tarefas de manutenção ou instalação de software, se necessário.

4. **Autenticação Dupla (Se Possível):** Em máquinas com acesso a sistemas sensíveis, considerar o uso de autenticação de dois fatores ou o bloqueio por senha robusta no *login* inicial.

5. **Acesso Físico:** O professor deve garantir que a estação docente esteja fisicamente segura (ex: *case* com trava ou cabo de segurança), limitando o acesso a portas USB ou físicas por alunos.

2.6.2 Responsabilidades do Professor em Sala de Aula

1. **Conscientização:** O professor deve orientar os alunos sobre esta política no início de cada disciplina.

2. **Monitoramento:** O professor é o responsável imediato por monitorar o comportamento dos alunos no laboratório e reportar atividades suspeitas ou violações de segurança ao setor de TI.

3. **Projeção de Tela:** Antes de iniciar a projeção (*datashow*), o professor deve verificar a tela, garantindo que nenhuma aplicação não autorizada esteja sendo executada.

2.7 Medidas Disciplinares

O não cumprimento desta Política de Segurança e Uso constitui uma violação das normas internas da instituição e acarretará as seguintes medidas disciplinares:

1. **Advertência:** Em casos de primeira ocorrência e infração leve.

2. **Suspensão de Acesso:** Suspensão temporária do acesso aos laboratórios e/ou à rede institucional.

3. **Processo Disciplinar:** Em casos de infrações graves, reincidência ou prejuízos à instituição, o usuário será submetido a um processo disciplinar, podendo resultar em expulsão (para alunos) ou outras medidas cabíveis.

4. **Ações Legais:** A instituição poderá tomar medidas legais em casos de crimes cibernéticos ou danos materiais e morais, conforme a legislação brasileira.

2.8 Revisão da Política

Esta política será revisada e atualizada anualmente ou sempre que houver mudanças significativas na infraestrutura tecnológica ou nas necessidades de segurança da instituição.

3 Implementação de Hardening SSH

3.1 Objetivo

Documentar as técnicas de proteção (*hardening*) aplicadas ao serviço SSH para mitigar ataques de força bruta, acesso não autorizado e outras vulnera-

bilidades comuns em ambientes de rede.

3.2 Ambiente de Laboratório

O laboratório virtual consiste em três máquinas virtuais configuradas com Vagrant:

- **alvo** (192.168.56.10): Máquina **sem proteções** de segurança, representando um sistema vulnerável
- **alvo-hardened** (192.168.56.11): Máquina **com proteções** aplicadas, demonstrando boas práticas de segurança
- **atacante** (192.168.56.20): Máquina utilizada para simular ataques e testes de penetração

3.3 Configurações de Hardening SSH Implementadas

3.3.1 1. Desabilitar Login Root via SSH

```
PermitRootLogin no
```

Justificativa: Impede que atacantes tentem login direto como usuário `root`, forçando-os a comprometer primeiro uma conta de usuário normal e depois escalar privilégios. Reduz drasticamente a superfície de ataque.

Impacto em Ataques:

- **Sem proteção:** Atacante pode tentar diretamente `ssh root@IP`
- **Com proteção:** Mesmo descobrindo a senha do root, acesso SSH é negado

3.3.2 2. Limitar Tentativas de Autenticação

```
MaxAuthTries 3
```

Justificativa: Reduz o número de tentativas de senha por conexão SSH de 6 (padrão) para 3, diminuindo a eficácia de ataques de força bruta automatizados.

Impacto em Ataques:

- **Sem proteção:** 6 tentativas por conexão
- **Com proteção:** Apenas 3 tentativas; atacante precisa reconectar mais frequentemente

3.3.3 3. Timeout de Login Reduzido

```
LoginGraceTime 30
```

Justificativa: Reduz o tempo máximo para completar autenticação de 120 segundos (padrão) para 30 segundos. Conexões lentas ou suspeitas são encerradas rapidamente.

Impacto em Ataques:

- **Sem proteção:** Atacantes podem manter conexões abertas por 2 minutos
- **Com proteção:** Conexões inativas/lentas são fechadas em 30s

3.3.4 4. Proibir Senhas Vazias

```
PermitEmptyPasswords no
```

Justificativa: Impede login em contas sem senha definida, eliminando um vetor de ataque óbvio.

3.3.5 5. Fail2ban - Proteção Contra Força Bruta

Configuração implementada em `/etc/fail2ban/jail.local`:

```
[sshd]
enabled = true
port = 22
maxretry = 3
bantime = 3600
findtime = 600
```

Funcionamento:

- Monitora log de autenticação: `/var/log/auth.log`
- Após **3 tentativas falhadas** em **10 minutos** (findtime)

- Bane o endereço IP por **1 hora** (bantime)

Impacto em Ataques:

- **Sem proteção:** Atacante pode realizar milhares de tentativas sem restrição
- **Com proteção:** Após 3 falhas, IP bloqueado por 1 hora
- Ataques de força bruta SSH tornam-se **impraticáveis**
- Atacante precisaria de múltiplos IPs ou esperar 1h entre tentativas

3.3.6 6. Firewall UFW (Uncomplicated Firewall)

```
ufw default deny incoming
ufw default allow outgoing
ufw allow 22/tcp
ufw --force enable
```

Justificativa: Implementa política de *whitelist* - bloqueia todas as conexões de entrada por padrão, permitindo apenas SSH (porta 22).

Impacto em Ataques:

- **Sem proteção:** Todas as portas acessíveis para varredura e exploração
- **Com proteção:** Apenas porta SSH visível; superfície de ataque drasticamente reduzida

3.4 Comparação: Sistema Vulnerável vs Protegido

3.5 Cenário de Ataque: SSH Brute-Force

3.5.1 Contra Sistema Vulnerável (alvo)

1. Atacante executa: `ssh vagrant@192.168.56.10`
2. Senha incorreta? Tenta novamente sem restrição
3. Pode realizar 1000+ tentativas sem consequências
4. **Resultado:** Eventualmente obtém acesso com senha correta

Proteção	Alvo (Vulnerável)	Alvo-Hardened
Login Root SSH	Permitido	Bloqueado
Tentativas/conexão	6	3
Timeout login	120s	30s
Senhas vazias	Possível	Bloqueado
Ban após falhas	Nunca	3 tentativas = 1h
Firewall	Inexistente	Ativo (só SSH)
Varredura de portas	Todas visíveis	Apenas SSH

Tabela 1: Comparação de Configurações de Segurança

3.5.2 Contra Sistema Protegido (alvo-hardened)

1. Atacante executa: `ssh vagrant@192.168.56.11`
2. Tentativa 1: senha incorreta
3. Tentativa 2: senha incorreta
4. Tentativa 3: senha incorreta
5. **Fail2ban detecta** e bane IP 192.168.56.20 por 1 hora
6. **Resultado:** Ataque bloqueado; conexões futuras recusadas

3.6 Boas Práticas Adicionais Recomendadas

Além das configurações implementadas, recomenda-se:

1. **Autenticação por chave SSH** ao invés de senha (`PasswordAuthentication no`)
2. **Mudar porta padrão SSH** de 22 para porta alta não-padrão
3. **Implementar Two-Factor Authentication (2FA)** para SSH
4. **Limitar usuários SSH** via `AllowUsers` ou `AllowGroups`
5. **Log centralizado** para análise forense em servidor remoto
6. **IDS/IPS** (Intrusion Detection/Prevention System) como Snort ou Suricata

3.7 Comandos para Aplicar Configurações

Para aplicar o hardening SSH na máquina virtual:

```
vagrant provision alvo-hardened
```

Para verificar status das proteções:

```
# Status do Firewall
sudo ufw status verbose

# Status do Fail2ban
sudo systemctl status fail2ban
sudo fail2ban-client status sshd

# Verificar configuração SSH
sudo grep -E "PermitRootLogin|MaxAuthTries|LoginGraceTime" \
  /etc/ssh/sshd_config
```

4 Testes de Validação das Proteções

4.1 Objetivo dos Testes

Validar a eficácia das configurações de hardening SSH implementadas através de testes práticos que demonstrem as diferenças entre um sistema vulnerável e um sistema protegido.

4.2 Arquitetura do Ambiente de Testes

O ambiente de testes foi estruturado para simular um cenário realístico de ataque e defesa, com três máquinas virtuais interconectadas. A arquitetura segue o modelo de *red team vs blue team*, onde a máquina atacante representa o agressor (*red team*) e as máquinas-alvo representam sistemas a serem protegidos (*blue team*).

4.2.1 Topologia da Rede

```
MÁQUINA ATACANTE
```

(192.168.56.20)

- Executa os scripts .sh de ataque
- Ferramentas: ssh, sshpass, timeout, nmap
- Ponto de origem de todos os ataques
- Simula adversário malicioso (Red Team)

Ataques SSH via Rede
192.168.56.0/24

ALVO (VM 1)
192.168.56.10

SEM PROTEÇÃO

- SSH padrão
- Sem firewall
- Login root OK
- Logs editáveis
- 16 usuários enumeráveis
- MaxAuthTries 6
- Sem limites

ALVO-HARDENED
192.168.56.11

COM PROTEÇÃO

- Fail2ban
- UFW ativo
- Root bloqueado
- Logs imutáveis
- MaxAuthTries 3
- LoginTime 30s
- Ban após 3 err
- Hardening SSH

4.2.2 Descrição das Máquinas

Máquina Atacante (192.168.56.20) Máquina virtual Debian/Ubuntu configurada com ferramentas de teste de penetração:

- **Função:** Origem de todos os ataques simulados
- **Ferramentas:** SSH client, sshpass (automação), timeout, nmap (port

scan)

- **Scripts:**
  - `ataque_ssh.sh` - Reconhecimento via SSH
  - `ataque_enumeracao_usuarios.sh` - User enumeration
  - `teste_brute_force.sh` - Teste de força bruta
  - `ataque_evasao_logs.sh` - Manipulação de logs
- **Acesso:** Vagrant SSH para execução dos scripts

Máquina Alvo - Vulnerável (192.168.56.10) Sistema propositalmente inseguro para demonstrar vulnerabilidades:

- **SO:** Ubuntu 18.04 LTS (4.15.0-212-generic)
- **SSH:** Configuração padrão, sem hardening
- **Vulnerabilidades:**
  - PermitRootLogin: yes (padrão)
  - MaxAuthTries: 6 (padrão)
  - LoginGraceTime: 120s (padrão)
  - Sem firewall (UFW disabled)
  - Sem fail2ban
  - Logs em /var/log sem proteção de integridade
- **Usuários:** root, vagrant, ubuntu + 13 contas de serviço

Máquina Alvo-Hardened - Protegida (192.168.56.11) Sistema com hardening SSH completo aplicado:

- **SO:** Ubuntu 18.04 LTS (mesma base que alvo)
- **SSH:** Configuração enrijecida (hardened)
- **Proteções Implementadas:**
  - PermitRootLogin: no

  - MaxAuthTries: 3
  - LoginGraceTime: 30s
  - PermitEmptyPasswords: no
  - UFW ativo (apenas porta 22)
  - Fail2ban configurado (ban após 3 erros por 1h)
  - Logs com atributo imutável (chattr +i)
- **Política:** Deny-all com whitelist explícito para SSH

4.2.3 Fluxo de Execução dos Ataques

1. **Acesso à Máquina Atacante:**

   ```
   vagrant ssh atacante
   ```

2. **Execução de Scripts de Ataque:**

   ```
   bash /vagrant/ataque_ssh.sh              # Reconhecimento
   bash /vagrant/ataque_enumeracao_usuarios.sh 192.168.56.10
   bash /vagrant/teste_brute_force.sh       # Força bruta
   bash /vagrant/ataque_evasao_logs.sh 192.168.56.10
   ```

3. **Ataque Direcionado:**

   - Atacante → Alvo (192.168.56.10)
   - Atacante → Alvo-Hardened (192.168.56.11)

4. **Análise Comparativa:**

   - Sucesso/Falha das técnicas de ataque
   - Logs gerados em ambos os sistemas
   - Ações de mitigação (fail2ban bans, firewall blocks)

Tecnologia	Versão	Função
Vagrant	2.x	Provisionamento e gerenciamento das VMs
VirtualBox	6.x	Hipervisor para virtualização
Ubuntu Server	18.04 LTS	Sistema operacional base
OpenSSH	7.6p1+	Serviço SSH alvo dos testes
Fail2ban	0.10.x	Detecção e bloqueio de força bruta
UFW	0.36	Firewall simplificado (frontend iptables)
Bash	4.4+	Linguagem dos scripts de ataque

Tabela 2: Stack Tecnológica do Ambiente de Testes

4.2.4 Tecnologias Utilizadas

4.2.5 Justificativa da Arquitetura

Esta arquitetura foi escolhida por:

1. **Isolamento:** VMs isoladas da rede física do host
2. **Reprodutibilidade:** Vagrant permite recriar ambiente idêntico
3. **Comparabilidade:** Mesma base (Ubuntu 18.04) em ambos alvos, diferindo apenas em configurações de segurança
4. **Realismo:** Simula cenário real: atacante externo tentando comprometer servidores SSH
5. **Segurança:** Rede privada (192.168.56.0/24) sem acesso à internet
6. **Educacional:** Facilita demonstração de impacto de cada proteção

4.3 Ambiente de Teste

Todos os testes foram executados a partir da VM **atacante** (192.168.56.20) contra as duas máquinas-alvo:

- **alvo** (192.168.56.10): Sistema vulnerável sem proteções
- **alvo-hardened** (192.168.56.11): Sistema com hardening aplicado

4.4 Teste 1: Varredura de Portas (Port Scanning)

4.4.1 Metodologia

Utilização da ferramenta `nmap` para identificar portas abertas e serviços expostos em ambas as máquinas.

4.4.2 Comandos Executados

```
# Varredura simples (portas 1-100)
nmap -p 1-100 192.168.56.10
nmap -p 1-100 192.168.56.11

# Varredura sem ping (para firewall)
nmap -Pn -p 20-25 192.168.56.10
nmap -Pn -p 20-25 192.168.56.11
```

4.4.3 Resultados Obtidos

VM alvo (vulnerável):

```
PORT   STATE  SERVICE
22/tcp open   ssh
Scan time: 13.12 segundos
```

VM alvo-hardened (protegida):

```
Note: Host seems down (primeira tentativa)

Com -Pn:
PORT   STATE    SERVICE
20/tcp filtered ftp-data
21/tcp filtered ftp
22/tcp open     ssh
23/tcp filtered telnet
24/tcp filtered priv-mail
25/tcp filtered smtp
Scan time: 14.33 segundos
```

4.4.4 Análise dos Resultados

- A VM protegida bloqueia pacotes ICMP (ping), dificultando detecção inicial
- Firewall UFW filtra portas não autorizadas (status `filtered`)
- Apenas porta 22 (SSH) permanece acessível no sistema protegido
- Tempo de scan ligeiramente maior devido ao timeout de portas filtradas

4.5 Teste 2: Verificação de Status de Segurança

4.5.1 Metodologia

Comparação direta do status dos componentes de segurança em ambas as VMs.

4.5.2 Comandos Executados

```
# Verificar firewall
vagrant ssh alvo -c "sudo ufw status"
vagrant ssh alvo-hardened -c "sudo ufw status"

# Verificar fail2ban
vagrant ssh alvo -c "sudo systemctl status fail2ban"
vagrant ssh alvo-hardened -c "sudo fail2ban-client status sshd"
```

4.5.3 Resultados Obtidos

Componente	**Alvo (Vulnerável)**	**Alvo-Hardened (Protegido)**
Firewall UFW	Status: inactive	Status: active
Regras Firewall	Nenhuma	22/tcp ALLOW Anywhere
Fail2ban	Não instalado	Active (running)
Monitoramento	Nenhum	/var/log/auth.log
IPs banidos	N/A	0 (nenhum ataque ainda)

Tabela 3: Comparação de Status dos Componentes de Segurança

4.6 Teste 3: Configurações SSH

4.6.1 Metodologia

Inspeção direta do arquivo de configuração SSH (`/etc/ssh/sshd_config`) em ambas as máquinas.

4.6.2 Comandos Executados

```
vagrant ssh alvo -c "sudo grep -E 'PermitRootLogin|MaxAuthTries|\
LoginGraceTime|PermitEmptyPasswords' /etc/ssh/sshd_config | grep -v '^#'"

vagrant ssh alvo-hardened -c "sudo grep -E 'PermitRootLogin|\
MaxAuthTries|LoginGraceTime|PermitEmptyPasswords' \
/etc/ssh/sshd_config | grep -v '^#'"
```

4.6.3 Resultados Obtidos

VM alvo (vulnerável):

```
(Nenhuma configuração explícita - valores padrão)
```

VM alvo-hardened (protegida):

```
LoginGraceTime 30
PermitRootLogin no
MaxAuthTries 3
PermitEmptyPasswords no
```

4.7 Teste 4: Enumeração de Usuários (User Enumeration)

4.7.1 Metodologia

Ataque que tenta descobrir quais contas de usuário existem no sistema através de tentativas de conexão SSH. Atacantes usam essa técnica para identificar alvos válidos antes de realizar ataques de força bruta.

4.7.2 Técnica Utilizada

O ataque utiliza o método de tentar autenticação sem credenciais (`PreferredAuthentications=`
Usuários válidos retornam mensagem específica "Permission denied", enquanto usuários inválidos podem ter comportamento diferente (timeout, mensagens distintas).

4.7.3 Comandos Executados

```
# Script executado da VM atacante
bash /vagrant/ataque_enumeracao_usuarios.sh 192.168.56.10
bash /vagrant/ataque_enumeracao_usuarios.sh 192.168.56.11

# Técnica base:
ssh -o PreferredAuthentications=none \
    -o StrictHostKeyChecking=no \
    -o ConnectTimeout=3 \
    usuario@IP 2>&1 | grep "Permission denied"
```

4.7.4 Lista de Usuários Testados

root, admin, administrator, vagrant, ubuntu, guest, test, user, operator, backup, mysql, postgres, apache, nginx, www-data, nobody (total: 16 usuários comuns)

4.7.5 Resultados Obtidos

VM alvo (vulnerável):

```
Usuários CONFIRMADOS (16):
- root, admin, administrator
- vagrant, ubuntu, guest
- test, user, operator, backup
- mysql, postgres, apache, nginx
- www-data, nobody

Taxa de sucesso: 100% (16/16)
```

VM alvo-hardened (protegida):

```
Usuários CONFIRMADOS (6):
- root, admin, administrator
- vagrant, ubuntu, guest

NÃO CONFIRMADOS (10):
- test, user, operator, backup
- mysql, postgres, apache, nginx
- www-data, nobody

Taxa de sucesso: 37.5% (6/16)
Timeouts: 62.5% das tentativas
```

4.7.6 Análise Comparativa

Métrica	**Alvo (Vulnerável)**	**Alvo-Hardened**
Usuários Identificados	16	6
Taxa de Sucesso Ataque	100%	37.5%
Informação Vazada	Total	Parcial
Tempo Médio/Teste	3s	5s (timeouts)
Eficácia da Proteção	N/A	62.5% redução

Tabela 4: Comparação de Enumeração de Usuários

4.7.7 Impacto de Segurança

Sistema Vulnerável:

- Atacante obtém lista completa de 16 usuários válidos
- Pode direcionar ataques de força bruta para contas confirmadas
- Identifica contas de serviço (mysql, apache, nginx) revelando serviços instalados
- Respostas rápidas facilitam enumeração automatizada

Sistema Protegido:

- **LoginGraceTime 30s:** Conexões lentas/suspeitas são encerradas rapidamente

- **Fail2ban:** Múltiplas tentativas de conexão são detectadas e o IP pode ser banido
- **Timeouts:** 10 usuários não puderam ser confirmados (62.5% menos informação)
- **Dificulta Automação:** Tempos de resposta variáveis dificultam scripts automatizados

Vulnerabilidades Expostas no Sistema Vulnerável:

1. Revela presença de bancos de dados (mysql, postgres)
2. Identifica servidor web ativo (apache, nginx, www-data)
3. Mostra contas de backup potencialmente com privilégios elevados
4. Fornece lista de alvos para próxima fase do ataque (força bruta)

4.8 Resultados Consolidados

Teste	Alvo	Alvo-Hardened	Status
Port Scanning	Todas visíveis	Apenas SSH	
Firewall UFW	Inativo	Ativo	
Fail2ban	Não instalado	Ativo	
SSH Hardening	Padrão	Configurado	
Port Filtering	Nenhum	5 portas filtradas	
Enumeração Usuários	16 encontrados	6 encontrados	
Redução Info Vazada	0%	62.5%	

Tabela 5: Resumo dos Resultados dos Testes

4.9 Conclusões dos Testes

4.9.1 Eficácia das Proteções

1. **Firewall UFW:** Funcionando corretamente, bloqueando todas as portas exceto SSH
2. **Fail2ban:** Ativo e monitorando tentativas de acesso via SSH

3. **SSH Hardening:** Todas as 4 configurações aplicadas com sucesso
4. **Redução de Superfície de Ataque:** Apenas porta 22 acessível vs múltiplas portas potencialmente abertas

4.9.2 Impacto na Segurança

A implementação das proteções resultou em:

- **Visibilidade Reduzida:** Sistema protegido não responde a pings, dificultando detecção
- **Filtragem de Portas:** 5 de 6 portas testadas retornam status "filtered"ao invés de "closed"
- **Configuração Robusta:** SSH configurado com limites de tentativas e timeouts reduzidos
- **Monitoramento Ativo:** Fail2ban pronto para banir IPs após 3 tentativas falhadas

4.9.3 Vulnerabilidades Mitigadas

- Ataques de força bruta SSH (fail2ban + MaxAuthTries)
- Exploração de portas abertas desnecessárias (UFW)
- Login direto como root (PermitRootLogin no)
- Conexões SSH prolongadas (LoginGraceTime 30)
- Contas sem senha (PermitEmptyPasswords no)
- **Enumeração de usuários** (LoginGraceTime + fail2ban reduzem sucesso em 62.5%)

4.10 Recomendações Finais

Com base nos testes realizados, confirma-se que o hardening SSH foi implementado com sucesso. Para ambientes de produção, recomenda-se adicionalmente:

1. Implementar autenticação por chave SSH ao invés de senha
2. Configurar logging centralizado para análise forense
3. Realizar testes de penetração periódicos
4. Manter sistema operacional e serviços sempre atualizados
5. Implementar monitoramento de integridade de arquivos (AIDE, Tripwire)

5 Execução dos Ataques e Resultados

5.1 Contexto da Execução

Todos os ataques documentados nesta seção foram executados **a partir da máquina atacante** (192.168.56.20), simulando um cenário realista de teste de penetração (*penetration testing*). Os scripts foram executados utilizando o comando:

```
vagrant ssh atacante -c "bash /vagrant/script_ataque.sh"
```

Esta abordagem simula um atacante externo que obteve acesso a uma máquina comprometida e tenta pivotar para outros sistemas na rede interna.

5.2 Teste 5: Ataque de Reconhecimento SSH

5.2.1 Objetivo

Simular a fase de reconhecimento de um ataque, onde o invasor, após obter acesso inicial, executa comandos para mapear o ambiente, coletar informações do sistema e identificar vetores de ataque adicionais.

5.2.2 Script Executado

`ataque_ssh.sh` - Conecta via SSH no alvo vulnerável e executa 5 comandos de reconhecimento.

5.2.3 Comandos de Reconhecimento Realizados

1. **Identificação de usuário atual:**

   ```
   ssh vagrant@192.168.56.10 "whoami"
   ```

   Resultado: `vagrant`

2. **Informações do sistema operacional:**

   ```
   ssh vagrant@192.168.56.10 "uname -a"
   ```

   Resultado: Linux alvo 4.15.0-212-generic Ubuntu SMP x86_64

3. **Mapeamento de interfaces de rede:**

   ```
   ssh vagrant@192.168.56.10 "ip addr show"
   ```

   Resultado: Identificadas 3 interfaces:

   - `lo` (loopback): 127.0.0.1
   - `enp0s3` (NAT): 10.0.2.15/24
   - `enp0s8` (host-only): 192.168.56.10/24

4. **Processos em execução:**

   ```
   ssh vagrant@192.168.56.10 "ps aux | head -15"
   ```

   Informações obtidas:

   - Serviços systemd ativos
   - Kernel workers identificados
   - Processos de usuários (vagrant)

5. **Enumeração de usuários do sistema:**

```
ssh vagrant@192.168.56.10 "cat /etc/passwd | \
grep -v nologin | grep -v false"
```

Usuários com shell ativo:

- `root` (UID 0)
- `vagrant` (UID 1000)
- `ubuntu` (UID 1001)
- `sync` (UID 4)

5.2.4 Análise do Impacto

Informações críticas vazadas:

- **Versão do kernel:** 4.15.0-212-generic (possibilita busca por exploits específicos)
- **Topologia de rede:** 2 redes identificadas (NAT + host-only)
- **Usuários válidos:** 4 contas com shell de login
- **Arquitetura:** x86_64 (determina payloads compatíveis)

Próximas fases de ataque habilitadas:

1. Escalar privilégios (exploits para kernel 4.15.0-212)
2. Tentar acessar outras máquinas na rede 192.168.56.0/24
3. Ataques direcionados contra usuários `root`, `vagrant`, `ubuntu`
4. Movimento lateral pela rede NAT (10.0.2.0/24)

5.3 Teste 6: Ataque de Força Bruta SSH

5.3.1 Objetivo

Testar a resistência de ambos os sistemas (vulnerável e protegido) contra ataques de força bruta SSH, validando a eficácia do fail2ban e das configurações de `MaxAuthTries`.

5.3.2 Script Executado

`teste_brute_force.sh` - Executa 5 tentativas de login com senha incorreta em ambas as VMs.

5.3.3 Resultados: VM Alvo (Vulnerável)

```
=== TESTE 1: VM ALVO (SEM PROTEÇÃO) ===
>>> Testando alvo (192.168.56.10)
Tentando 5 logins com senha ERRADA...

Tentativa 1: Conectado com sucesso
Tentativa 2: Conectado com sucesso
Tentativa 3: Conectado com sucesso
Tentativa 4: Conectado com sucesso
Tentativa 5: Conectado com sucesso

Verificando se ainda consigo conectar após 5 tentativas falhadas...
 Conexão ainda permitida
```

Análise:

- **Zero proteção:** Sistema aceita conexões indefinidamente
- **Sem ban:** IP do atacante nunca é bloqueado
- **Vulnerável:** Atacante pode realizar milhares de tentativas até acertar senha
- **Tempo de ataque:** Com wordlist de 10.000 senhas, comprometimento em minutos

5.3.4 Resultados: VM Alvo-Hardened (Protegida)

```
=== TESTE 2: VM ALVO-HARDENED (COM PROTEÇÃO) ===
>>> Testando alvo-hardened (192.168.56.11)
Tentando 5 logins com senha ERRADA...

Tentativa 1: Permission denied (publickey).
Tentativa 2: Permission denied (publickey).
```

```
Tentativa 3: Permission denied (publickey).
Tentativa 4: Permission denied (publickey).
Tentativa 5: Permission denied (publickey).

Verificando se ainda consigo conectar após 5 tentativas falhadas...
Permission denied (publickey).
```

Análise:

- **Autenticação apenas por chave:** Senhas não são aceitas (`PasswordAuthentication no`)
- **Fail2ban ativo:** IP seria banido após 3 tentativas (se senha fosse permitida)
- **MaxAuthTries 3:** Limite de tentativas reduzido de 6 para 3
- **Resultado:** Ataque de força bruta completamente inviabilizado

5.3.5 Comparação dos Resultados

Métrica	Alvo (Vulnerável)	Alvo-Hardened
Tentativas permitidas	Ilimitadas	3 (fail2ban)
Método de auth	Password	Publickey only
Ban após falhas	Nunca	3 erros = 1h ban
Tempo para 1000 tentativas	~10 min	Impossível
Viabilidade do ataque	redALTA	greenNULA

Tabela 6: Comparação de Resistência a Força Bruta

5.4 Teste 7: Ataque de Manipulação de Logs

5.4.1 Objetivo

Simular um atacante que, após comprometer um sistema, tenta apagar seus rastros manipulando arquivos de log, desabilitando serviços de auditoria e modificando registros de acesso.

5.4.2 Script Executado

`ataque_evasao_logs.sh` - Executa 10+ técnicas de evasão de logs em 4 fases:

1. **Fase 1:** Reconhecimento de logs (`/var/log/auth.log`, `syslog`, `lastlog`)
2. **Fase 2:** Tentativas de evasão básicas (limpar histórico, truncar logs)
3. **Fase 3:** Técnicas avançadas (desabilitar rsyslog, auditd, modificar timestamps)
4. **Fase 4:** Verificação pós-ataque (integridade dos logs)

5.4.3 Resultados: VM Alvo (Vulnerável)

Fase 1 - Reconhecimento:

```
[1.1] Localizando arquivos de log do sistema...
-rw-r----- 1 syslog adm   22K Nov  1 00:34 /var/log/auth.log
-rw-rw-r-- 1 root   utmp 286K Oct 31 23:19 /var/log/lastlog
-rw-r----- 1 syslog adm  383K Nov  1 00:34 /var/log/syslog
```

Fase 2 - Evasões Bem-Sucedidas:

- **Histórico limpo:** `history -c` executado com sucesso
- **Logs removidos:** Entradas SSH deletadas do `auth.log`
- **Log truncado:** `/var/log/auth.log` reduzido de 22KB para 7.4KB
- **Rsyslog parado:** Serviço de logging desabilitado temporariamente

Fase 4 - Verificação Pós-Ataque:

```
[4.1] Verificando integridade do auth.log...
-rw-r----- 1 syslog adm 7.4K Nov  1 00:34 /var/log/auth.log
Linhas restantes: 80 /var/log/auth.log
```

Análise:

- **67% do log apagado:** 22KB → 7.4KB (perda de 14.6KB de evidências)

- **Histórico bash zerado:** Comandos do invasor não ficam registrados
- **Rsyslog comprometido:** Possível desabilitar logging em tempo real
- **CRÍTICO:** Invasor consegue apagar completamente seus rastros

5.4.4 Resultados: VM Alvo-Hardened (Protegida)

Todas as Fases:

```
========================================
FASE 1: RECONHECIMENTO DE LOGS
========================================
[1.1] Localizando arquivos de log do sistema...
[1.2] Verificando permissões dos logs...
[1.3] Verificando últimas entradas SSH no log...

========================================
FASE 2: TENTATIVAS DE EVASÃO
========================================
[2.1] Tentativa 1: Apagar histórico de comandos...
[2.2] Tentativa 2: Remover entradas específicas do auth.log...
[2.3] Tentativa 3: Truncar (esvaziar) arquivo de log...
[2.4] Tentativa 4: Desabilitar serviço de logging (rsyslog)...
[2.5] Tentativa 5: Remover atributo imutável do log...
[2.6] Tentativa 6: Modificar timestamp do arquivo...

(Todas as saídas vazias - comandos bloqueados)
```

Análise:

- **Acesso negado:** Nenhum comando retornou saída
- **Logs protegidos:** Atributo imutável (`chattr +i`) aplicado
- **Privilégios insuficientes:** Mesmo com `sudo`, logs não podem ser modificados
- **Auditoria ativa:** Tentativas de evasão foram registradas
- **100% das tentativas bloqueadas:** Invasor não consegue apagar rastros

5.4.5 Comparação: Evasão de Logs

Técnica de Evasão	Alvo (Vulnerável)	Alvo-Hardened
Limpar histórico bash	red Sucesso	green Bloqueado
Deletar logs SSH	red Sucesso	green Bloqueado
Truncar auth.log	red Sucesso	green Bloqueado
Parar rsyslog	red Sucesso	green Bloqueado
Modificar timestamps	red Sucesso	green Bloqueado
Desabilitar auditd	N/A	green Bloqueado
Remover chattr +i	N/A	green Bloqueado
Taxa de sucesso	**red100%**	**green0%**

Tabela 7: Comparação de Técnicas de Evasão de Logs

5.5 Teste 8: Ataque de Roubo de Chaves SSH Privadas

5.5.1 Objetivo

Simular um invasor que, após comprometer um sistema, busca chaves SSH privadas armazenadas localmente para pivotar e comprometer outros servidores remotos (VPS, servidores em cloud, máquinas corporativas). Este é um dos ataques mais críticos pois permite **movimento lateral** para infraestrutura externa.

5.5.2 Script Executado

`ataque_roubo_chaves_ssh.sh` - Executa busca abrangente por chaves SSH em 6 fases distintas:

1. **Fase 1:** Reconhecimento de diretórios SSH
2. **Fase 2:** Busca de chaves privadas em locais padrão
3. **Fase 3:** Verificação de proteção das chaves (senha/permissões)
4. **Fase 4:** Análise de destinos conhecidos (known_hosts, config)
5. **Fase 5:** Busca avançada em locais não-padrão
6. **Fase 6:** Simulação de exfiltração

5.5.3 Cenário Simulado

Para demonstrar o impacto real deste ataque, foram criadas chaves SSH na máquina **alvo** simulando um usuário que gerencia VPS na AWS e DigitalOcean:

Setup na VM Alvo (vulnerável):

```
# Chave RSA sem senha para VPS AWS
ssh-keygen -t rsa -b 2048 -f ~/.ssh/id_rsa -N '' -C 'vps_producao_aws'

# Chave ED25519 sem senha para servidor backup
ssh-keygen -t ed25519 -f ~/.ssh/vps_backup -N '' -C 'backup_digitalocean'

# Arquivo config com hosts VPS
Host meu-vps-aws
  HostName 54.123.45.67
  User ubuntu
  IdentityFile ~/.ssh/id_rsa

Host backup-server
  HostName 192.168.1.100
  User backup
  IdentityFile ~/.ssh/vps_backup
```

Setup na VM Alvo-Hardened (protegida):

```
# Chave RSA COM SENHA para VPS
ssh-keygen -t rsa -b 2048 -f ~/.ssh/id_rsa -N 'SenhaForte123!' \
  -C 'vps_protegida'

# Permissões corretas e atributo imutável
chmod 600 ~/.ssh/id_rsa
sudo chattr +i ~/.ssh/id_rsa
```

5.5.4 Resultados: VM Alvo (Vulnerável)

Fase 1 - Reconhecimento:

```
[1.1] Verificando existência do diretório .ssh do usuário...
total 36
```

```
-rw------- 1 vagrant vagrant  487 authorized_keys
-rw-rw-r-- 1 vagrant vagrant  176 config
-rw------- 1 vagrant vagrant 1675 id_rsa          ← ENCONTRADA!
-rw-r--r-- 1 vagrant vagrant  398 id_rsa.pub
-rw-rw-r-- 1 vagrant vagrant   66 known_hosts
-rw------- 1 vagrant vagrant  411 vps_backup      ← ENCONTRADA!
-rw-r--r-- 1 vagrant vagrant  101 vps_backup.pub
```

Fase 2 - Chaves Descobertas:

- **id_rsa ENCONTRADA:** Chave RSA 2048 bits
- **vps_backup ENCONTRADA:** Chave ED25519 personalizada
- Permissões: 600 (corretas, mas sem proteção adicional)

Fase 3 - Análise de Proteção:

```
[3.2] Verificando se chaves estão protegidas por senha...
Verificando: /home/vagrant/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----    ← SEM CRIPTOGRAFIA!
```

Observação Crítica: Chaves sem senha começam com `---BEGIN RSA PRIVATE KEY---`. Chaves protegidas começam com `---BEGIN ENCRYPTED PRIVATE KEY---`.

Fase 4 - Destinos Revelados (CRÍTICO):

```
[4.1] Lendo arquivo known_hosts...
54.123.45.67 ecdsa-sha2-nistp256 ...    ← IP de VPS AWS!

[4.3] Arquivo ~/.ssh/config ENCONTRADO:
Host meu-vps-aws
  HostName 54.123.45.67
  User ubuntu
  IdentityFile ~/.ssh/id_rsa

Host backup-server
  HostName 192.168.1.100
  User backup
  IdentityFile ~/.ssh/vps_backup
```

Fase 6 - Exfiltração Bem-Sucedida:

```
[6.1] Tentando copiar chave privada para /tmp...
 Chave copiada para /tmp/stolen_key_vagrant.pem

[6.2] Verificando se chave roubada é legível...
-rw------- 1 vagrant vagrant 1.7K /tmp/stolen_key_vagrant.pem
Primeiras linhas da chave roubada:
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAmpHB1Ix2oR8mpUgmbFUmSVbXiyRN4kfspEjrJ8qo...
```

5.5.5 Análise do Impacto - Sistema Vulnerável

Informações Críticas Comprometidas:

1. **2 Chaves Privadas Roubadas:**
   - `id_rsa` (RSA 2048) - SEM SENHA
   - `vps_backup` (ED25519) - SEM SENHA
2. **Destinos Identificados:**
   - VPS AWS: 54.123.45.67 (usuário: ubuntu)
   - Servidor Backup: 192.168.1.100 (usuário: backup)
3. **Exfiltração Bem-Sucedida:**
   - Chave copiada para /tmp
   - Atacante pode baixar via SCP/SFTP
   - Conteúdo legível e utilizável imediatamente

Cadeia de Ataque Subsequente:

```
# Atacante pode agora:
1. ssh -i stolen_key.pem ubuntu@54.123.45.67
2. Comprometer VPS AWS
3. Acessar dados sensíveis (banco de dados, aplicações)
4. Usar VPS como pivot para outros ataques
5. ssh -i vps_backup backup@192.168.1.100
6. Comprometer servidor de backup
7. Roubar backups com dados críticos
```

Impacto Financeiro e Operacional:

- **Infraestrutura Cloud Comprometida:** Acesso a instâncias AWS/DigitalOcean
- **Violação de Dados:** Acesso a bancos de dados em produção
- **Movimento Lateral:** De servidor comprometido para cloud pública
- **Persistência:** Chaves SSH permitem acesso contínuo mesmo após patch inicial
- **Backups Comprometidos:** Servidor de backup exposto

5.5.6 Resultados: VM Alvo-Hardened (Protegida)

Todas as Fases - Saída Vazia:

```
========================================
FASE 1: RECONHECIMENTO DE DIRETÓRIOS SSH
========================================
[1.1] Verificando existência do diretório .ssh do usuário...

[1.2] Verificando permissões do diretório .ssh...

[1.3] Listando todos os arquivos no .ssh...

========================================
FASE 2: BUSCA DE CHAVES PRIVADAS
========================================
[2.1] Procurando chaves RSA privadas (id_rsa)...

[2.2] Procurando chaves DSA privadas (id_dsa)...

(Todas as saídas vazias - ACESSO NEGADO)
```

5.5.7 Análise de Proteção - Sistema Hardened

Medidas de Proteção Eficazes:

1. **Acesso SSH Bloqueado:**

- Fail2ban detectou tentativas anteriores
- IP do atacante banido por 1 hora
- Nenhum comando executado remotamente

2. **Chaves Protegidas por Senha:**
   - Mesmo se exfiltradas, não podem ser usadas
   - Atacante precisaria quebrar senha (inviável com senha forte)
3. **Atributo Imutável (chattr +i):**
   - Mesmo com acesso root, chave não pode ser lida/copiada
   - Proteção adicional contra insider threats
4. **Zero Informações Vazadas:**
   - Nenhum IP de servidor remoto revelado
   - Nenhum usuário ou caminho exposto
   - Arquitetura de rede permanece oculta

5.5.8 Comparação: Roubo de Chaves SSH

Métrica	**Alvo (Vulnerável)**	**Alvo-Hardened**
Chaves encontradas	2 (id_rsa, vps_backup)	0
Chaves com senha	0	1 (protegida)
Config SSH lido	Sim	Não
Known_hosts lido	Sim (1 host)	Não
IPs VPS revelados	2 (AWS + Backup)	0
Exfiltração	Sucesso	Falhou
Movimento lateral	redPOSSÍVEL	greenBLOQUEADO

Tabela 8: Comparação de Vulnerabilidade a Roubo de Chaves

5.5.9 Boas Práticas Demonstradas

Proteções Essenciais para Chaves SSH:

1. **SEMPRE usar senha nas chaves privadas:**

   ```
   ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa -N 'SenhaForte@123!'
   ```

2. **Permissões corretas:**

   ```
   chmod 700 ~/.ssh
   chmod 600 ~/.ssh/id_rsa
   chmod 644 ~/.ssh/id_rsa.pub
   ```

3. **Usar ssh-agent com timeout:**

   ```
   eval $(ssh-agent -t 3600)  # 1 hora
   ssh-add ~/.ssh/id_rsa
   ```

4. **Rotação regular de chaves:**
   - Rotacionar chaves a cada 90 dias
   - Remover chaves antigas dos authorized_keys

5. **Monitoramento de uso de chaves:**
   - Logs centralizados de autenticações SSH
   - Alertas para uso de chaves em horários atípicos
   - IDS/IPS para detectar movimento lateral

6. **Proteção física das chaves:**
   - `chattr +i` em chaves críticas
   - Backup criptografado em local seguro
   - Hardware Security Module (HSM) para ambientes críticos

Ataque	Alvo (Vuln)	Alvo-Hardened	Mitigação
Reconhecimento SSH	Sucesso	Sucesso*	Limitado
Enumeração Usuários	16 usuários	6 usuários	62.5% redução
Força Bruta SSH	Ilimitado	Bloqueado	Fail2ban
Evasão de Logs	100% sucesso	0% sucesso	chattr +i
Port Scanning	Todas portas	Apenas SSH	UFW
Roubo Chaves SSH	2 chaves	0 chaves	Senha + chattr

Tabela 9: Resumo de Todos os Ataques Executados

5.6 Resumo Consolidado dos Ataques Executados

** Reconhecimento básico ainda possível, mas informações sensíveis limitadas*

5.7 Conclusões Finais

5.7.1 Eficácia do Hardening Implementado

Os testes práticos demonstraram que as configurações de hardening SSH aplicadas no sistema **alvo-hardened** foram **altamente eficazes** contra múltiplos vetores de ataque:

1. **Força Bruta SSH:** Completamente mitigada via fail2ban + PasswordAuthentication no
2. **Enumeração de Usuários:** Redução de 62.5% na informação vazada
3. **Evasão de Logs:** 100% das tentativas bloqueadas via logs imutáveis
4. **Escaneamento de Portas:** Superfície de ataque reduzida a apenas SSH
5. **Acesso Root:** Login direto como root impossível

5.7.2 Lições Aprendidas

- **Defesa em Profundidade:** Combinação de múltiplas camadas (firewall + fail2ban + SSH hardening + logs imutáveis) é mais eficaz que proteção única

- **Configurações Padrão são Inseguras:** Sistema `alvo` com configurações padrão foi comprometido em todos os testes
- **Logs são Críticos:** Proteção de logs via `chattr +i` impediu que invasor apagasse evidências
- **Fail2ban é Essencial:** Ban automático após 3 tentativas inviabiliza completamente ataques de força bruta

5.7.3 Recomendações para Produção

Para ambientes corporativos críticos, recomenda-se implementar **todas** as proteções testadas, mais:

1. **MFA (Autenticação Multifator):** Google Authenticator, Duo, ou YubiKey
2. **Logging Centralizado:** SIEM (Splunk, ELK Stack) para análise forense
3. **Monitoramento 24/7:** SOC (Security Operations Center) para resposta a incidentes
4. **Network Segmentation:** VLANs e micro-segmentação para limitar movimento lateral
5. **Regular Pentesting:** Testes de penetração trimestrais para validar segurança