MÓDULO 02

Segurança *Blindando Seu Agente*

dmPolicy, firewall, fail2ban, credenciais seguras e proteção contra ataques

DURAÇÃO	FORMATO	KIT
15 min	**Live coding**	**2 arquivos**

Por Que Agora?

Servidores expostos na internet são alvos constantes de ataques. Seu servidor OpenClaw vai receber **1.000+ tentativas de brute force por dia**. Sem proteção, é questão de tempo até alguém entrar.

Log Real: 24 Horas de Ataques

1.015 tentativas de login SSH de 47 IPs diferentes. 3 tentativas de escalar privilégios. 12 scans de portas. Tudo em um único dia.

Este módulo não é opcional. **É o que separa "tô testando" de "tô em produção".**

As 9 Camadas de Proteção

1. **Telegram Allowlist (dmPolicy)** — só você comanda o bot
2. **Firewall UFW** — bloqueia portas desnecessárias
3. **Fail2ban** — 5 tentativas erradas de SSH = ban automático
4. **Credenciais no .env** — zero hardcode nos arquivos
5. **Portas em 127.0.0.1** — não expor apps na internet
6. **Cloudflare Tunnel** — acesso web seguro sem abrir portas
7. **SSH key-only** — desabilitar senha root
8. **Rotação de credenciais** — trocar chaves a cada 3 meses
9. **Audit periódico** — verificar vulnerabilidades toda semana

CRÍTICO

Se dmPolicy estiver "open", qualquer pessoa que encontrar seu bot pode comandar seu agente e acessar seus dados. Esse é o erro #1 que iniciantes cometem.

1. Telegram Allowlist (dmPolicy)

A primeira linha de defesa. Garante que apenas IDs autorizados podem usar o bot.

```bash
# Verificar config atual
cat ~/.openclaw/openclaw.json | grep -A5 dmPolicy
```

Procure por:

```json
{
  "dmPolicy": "allowlist",
  "allowedUsers": [123456789]
}
```

> ⚠ **Se estiver "open" — MUDE AGORA**
>
> Edite openclaw.json e mude pra "allowlist". Adicione seu Telegram ID (veja nos logs: `openclaw gateway logs`).

2. Firewall UFW

Bloqueia TODAS as portas de entrada, exceto as essenciais (SSH).

```bash
# Instalar e configurar
sudo apt install -y ufw
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow ssh
sudo ufw --force enable

# Verificar
sudo ufw status
```

Resultado esperado:

```text
Status: active

To                         Action      From
```

```
--                         ------      ----
22/tcp                     ALLOW       Anywhere
```

3. Fail2ban (Proteção SSH)

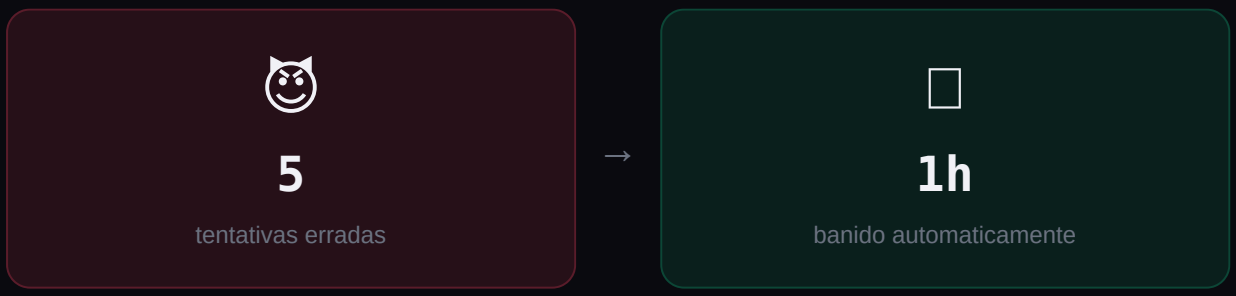
Detecta tentativas de brute force e bane IPs automaticamente.

```bash
# Instalar
sudo apt install -y fail2ban

# Configurar
sudo cat > /etc/fail2ban/jail.local << 'EOF'
[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 5
bantime = 3600
findtime = 600
EOF

# Ativar
sudo systemctl enable fail2ban
sudo systemctl restart fail2ban

# Verificar status
sudo fail2ban-client status sshd
```



4. Cloudflare Tunnel

Se você tiver apps web (Mission Control, dashboards), **NUNCA** exponha portas direto. Use Cloudflare Tunnel:

```bash
# Instalar cloudflared
curl -L https://github.com/cloudflare/cloudflared/releases/latest/download/cloudfl
sudo dpkg -i cloudflared.deb
```

```
# Autenticar
cloudflared tunnel login
```

5. Credenciais: Auditar e Corrigir

O problema mais comum: API keys hardcodadas nos arquivos. Se alguém acessar seu servidor, pega TUDO de uma vez.

1 Auditar (descobrir chaves expostas)

```bash
grep -r -n -I \
  -e "sk-[a-zA-Z0-9]" \
  -e "ANTHROPIC_API_KEY.*=.*['\"]" \
  -e "TELEGRAM.*TOKEN.*=.*['\"]" \
  --include="*.json" --include="*.md" --include="*.js" \
  /root/.openclaw/ 2>/dev/null
```

Se aparecer qualquer resultado → tem chave exposta.

2 Mover tudo pro .env

```bash
nano /root/.openclaw/.env
```

Formato (sem aspas, sem espaços):

```env
ANTHROPIC_API_KEY=sk-ant-sua-chave-aqui
OPENAI_API_KEY=sk-sua-chave-aqui
TELEGRAM_BOT_TOKEN=123456:ABC-seu-token-aqui
```

3 Proteger o .env

```bash
# Só root pode ler
chmod 600 /root/.openclaw/.env

# Verificar (deve mostrar -rw-------)
ls -la /root/.openclaw/.env
```

4 Remover chaves dos outros arquivos

Edite openclaw.json, scripts, etc. e apague as linhas com chaves. O OpenClaw lê automaticamente do .env.

6. Sync systemd + .env (Armadilha Comum!)

⚠ REGRA INVIOLÁVEL

Ao trocar qualquer credencial, atualizar em AMBOS os lugares:

1. Editar `/root/.openclaw/.env`
2. Atualizar o systemd override: `sudo systemctl edit openclaw`
3. Recarregar: `sudo systemctl daemon-reload`
4. Reiniciar: `sudo systemctl restart openclaw`

Por quê: O override do systemd tem prioridade sobre o .env. Se trocar só o .env, o valor antigo continua valendo. Muita gente perde horas debugando isso.

7. Rotação Trimestral de Credenciais

A cada 3 meses:

- Gerar nova API key na Anthropic
- Gerar novo bot token no Telegram (ou manter o mesmo)
- Atualizar .env e systemd override
- Reiniciar o gateway

Agende um lembrete no calendário pra não esquecer.

8. SSH Hardening

```bash
# Verificar se root login com senha está ativo
grep "PermitRootLogin" /etc/ssh/sshd_config
```

Ideal: `PermitRootLogin prohibit-password` (só SSH key, sem senha).

Mais Seguro

Crie um usuário não-root com sudo, configure SSH key, e desabilite completamente root login. Isso previne ataques de força bruta no usuário root.

Checkpoint de Segurança

- [ ] dmPolicy = allowlist (só meu Telegram ID)
- [ ] UFW ativo (firewall bloqueando portas)
- [ ] Fail2ban ativo (proteção SSH)
- [ ] Credenciais no .env (audit passou limpo)
- [ ] systemd + .env sincronizados
- [ ] SSH hardened (key-only preferível)
- [ ] Rotação trimestral agendada no calendário

Prompt para o Agente

Cole este prompt + anexe o arquivo prds/security-hardening.md:

Acabei de assistir o Módulo 2 do curso sobre segurança. Leia o PRD de security hardening que estou anexando e me guie passo a passo. **Importante:** - Antes de CADA ação, me explique O QUE vai fazer e POR QUÊ - Me peça confirmação antes de executar qualquer mudança no servidor - Se algo já estiver configurado, me avise e pule pro próximo passo - No final, rode um audit de segurança e me mostre o resultado **O que espero que cubra:** 1. Telegram allowlist (dmPolicy) — pra ninguém comandar meu bot 2. Firewall UFW — bloquear portas desnecessárias 3. Fail2ban — proteger contra brute force SSH 4. Credenciais — como guardar API keys de forma segura 5. Portas de aplicação — não expor nada em 0.0.0.0 Vamos blindar meu servidor?

PRÓXIMO MÓDULO

Módulo 3 — Identidade & Personalidade