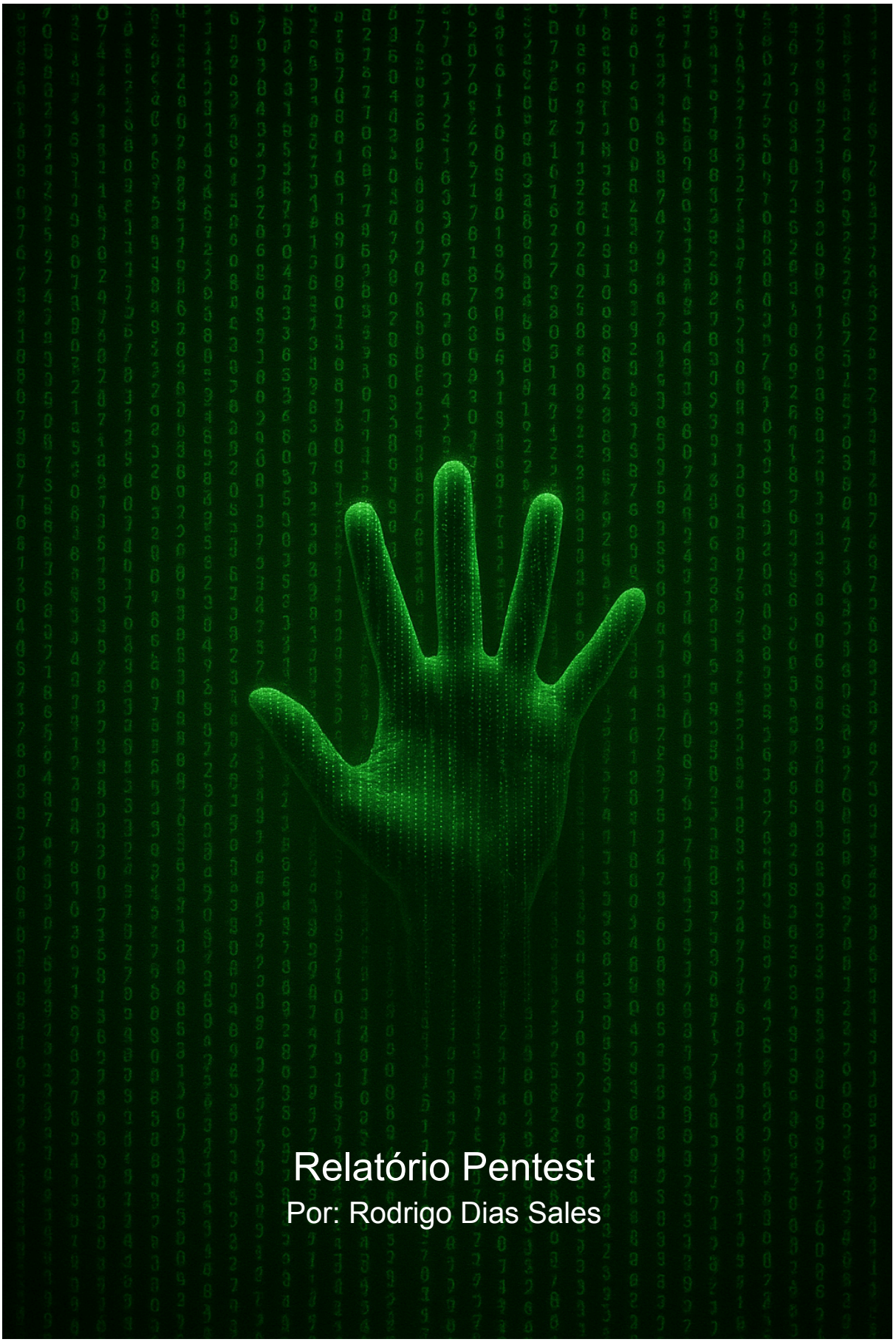

Relatório Pentest

Por: Rodrigo Dias Sales

1. Escopo

2. Objeto de análise

3. Data em que os testes foram feitos

4. Explicação sobre as evidências

5. Detalhamento dos dados e subdomínios

6.Detalhamento de softwares

7. Softwares usados, resultados e vulnerabilidades relatada

8. Quadro SWOT da segurança do site

9. Conclusão

10. Sugestões para o Contratante

1. Escopo

- Escopo técnico: Testes não-destrutivos de reconhecimentos e enumeração sobre o IP **98.95.207.28**.
- Não está dentro do escopo: ataques destrutivos, remoção/exclusão de dados, engenharia social e qualquer ação fora do IP definido.
- Ferramentas principais usadas na verificação inicial: **curl**, **dig**, **openssl**, **nmap**(escaneamento simples), **gobuster**, **nikto**, **whatweb**.

2. Objeto de análise

- Endereço IP alvo: **98.95.207.28**
- Reverse DNS: ec2-98-95-207-28.compute-1.amazonaws.com (obtido via dig -x).
- Provedor de hospedagem: Amazon Web Services (blocos identificados via whois).
- Tipo de sistema: Instância web pública (Apache + PHP identificados nos headers).

3. Data em que os testes foram realizados

- Data/Hora: **01/12/2025**

4. Explicação sobre as evidências

- Todas as evidências estão na pasta “Evidencias”, não diretamente no relatório, a fim de evitar que o relatório fique muito extenso.
- A partir daqui, sempre que for citar alguma evidência, vou indicar o nome da imagem que vai poder ser verificada separadamente do relatório.

5. Detalhamento dos dados e subdomínios

5.1 Tecnologias identificadas

- Servidor HTTP: Apache/2.4.54 (Debian), evidenciado em curl -I.png e nikto.png.
- Linguagem: PHP/7.4.33, evidenciado em curl -I.png.
- HTTPS: porta 443 aparentemente fechada/recusando conexões, evidenciado em openssl.png, tráfego apenas em HTTP no momento.

5.2 Diretórios e itens sensíveis identificados

- Identificados via robots.txt : /admin/, /backup/, /.git/, /config/ (e linha indicando /backup/database_backup_2024.sql), evidenciado em curl robots.txt.png.

5.3 DNS

- Alocação em blocos AWS (AMAZO-4 / ADSN-1), evidenciado em whois-pt1.png, whois-pt2.png, whois-pt3.png.

- dig reverse DNS confirma nome de host do EC2 (Amazon), evidenciado em dig -x +short.png.

5.4 Enumeração de conteúdo e endpoints

- Diretórios vhosts encontrados, evidenciados em gobuster.png e gobuster vhost.png.
- Na evidência grep - curl.png, existem pontos a investigar em relação a buscas/extrações no HTML.

5.5 Portas e serviços detectados

Porta	Situação	Serviço
21	OPEN	ftp
2222	OPEN	EtherNetIP-1
80	OPEN	http
3306	OPEN	mysql
8080	OPEN	http-proxy
22	FILTERED	ssh

- Evidenciado em nmaps 1.png e nmaps 2.png.

5.6 Impactos

- Serviços expostos ao público aumentam a superfície de ataque. Banco de dados MySQL acessível externamente (porta 3306) pode permitir enumeração ou acessos indevidos se não houver firewall/ACLs; portas HTTP/8080 indicam aplicação web e possíveis endpoints a serem avaliados; porta 2222 aberta com serviço não-SSH padrão merece investigação (padrão não convencional e potencialmente mal configurado). A porta 22 marcada como FILTERED indica que um filtro está presente (firewall/IDS) mas não necessariamente bloqueia todo o acesso.

5.7 Achados iniciais de segurança

- Ausência de cabeçalhos X-Frame-Options e X-Content-Type-Options.
- Cookie PHPPSESSID criado sem HttpOnly flag.
- Entrada em robots.txt referenciando arquivos sensíveis, risco de exposição.

Todos evidenciados por nikto.png.

6.Detalhamento de softwares

- O alvo responde como uma instância web Apache/2.4.54 sobre Debian, com aplicações em PHP (X-Powered-By: PHP/7.4.33), evidenciado por curl -I.png e whatweb.png.
- Foram detectadas aplicações web respondendo nas portas 80 e 8080 (evidências: nmap port 80,8080 pt1.png, nmap port 80,8080 pt2.png) que sugerem múltiplos serviços HTTP ou proxies reversos.
- O servidor expõe também serviços tradicionalmente sensíveis: FTP na porta 21 (nmap ftp.png), MySQL na porta 3306 (nmap port 3306.png) e um serviço na porta 2222 (nmaps 1.png / nmaps 2.png) cuja identificação precisa ser avaliada, pode ser um serviço não padronizado ou um serviço com porta custom.
- A porta 22 está marcada como FILTERED (nmaps 2.png), indicando algum filtro/firewall em frente ao SSH. A tentativa de conexão TLS em 443 retornou Connection refused (openssl.png), indicando que HTTPS não está ativo no momento.

7. Softwares usados, resultados e vulnerabilidades relatada

7.1 Principais resultados e vulnerabilidades observadas

- Divulgação de caminhos sensíveis via robots.txt: /admin/, /backup/, /.git/, /config/ e indicação explícita de backup /backup/database_backup_2024.sql (curl robots.txt.png). Severidade: **Alta** (informação que facilita descoberta de recursos sensíveis).
- Repositório .git potencialmente exposto (diretório listado em robots.txt; confirmar via .git/HEAD). Severidade: **Alta**.
- Ausência de TLS (porta 443 recusando conexões), tráfego em texto claro, cookies/credenciais transitando sem proteção (openssl.png). Severidade: **Média/Alta**.
- Falta de cabeçalhos de segurança essenciais (X-Frame-Options, X-Content-Type-Options, Content-Security-Policy ausentes) e cookie PHPPSESSID sem HttpOnly flag, evidenciado por nikto.png. Severidade: **Média**.
- Exposição de serviços de banco de dados (MySQL 3306 aberto) e FTP (porta 21) que ampliam a superfície de ataque. Severidade: **Alta** para MySQL público; **média/alta** para FTP dependendo da configuração, evidenciado por nmap ftp.png, nmap port 3306.png.
- Resultados de fuzzing identificaram endpoints adicionais e possíveis pontos de administração/backup, evidenciado por gobuster.png, grep - curl.png. Severidade: **varia por endpoint; recomenda-se revisão caso-a-caso.**
- Detecção de possível vulnerabilidade de injeção SQL, evidenciado por sqlmap.png. Severidade: **Crítica**.

8. Quadro SWOT da segurança do site

8.1 Forças

- Presença aparente de algum filtro na porta 22, evidenciado por nmaps 2.png, indicando controle de acesso básico ao SSH.
- O site responde de forma consistente e sem erros graves de indisponibilidade nos testes de verificação, evidenciado por curl -I.png.

8.2 Fraquezas

- Divulgação de caminhos sensíveis em robots.txt e indicação de arquivo de backup , evidenciado por curl robots.txt.png.
- Ausência de TLS, evidenciado por openssl.png, e cabeçalhos de segurança essenciais ausentes, evidenciado por nikto.png, tráfego e sessões potencialmente expostas.
- Serviços sensíveis expostos publicamente: MySQL (3306), FTP (21), e um serviço não-padrão na porta 2222, aumenta a superfície de ataque.
- Cookie de sessão sem HttpOnly flag, evidenciado por nikto.png, risco de roubo via XSS.
- Possível indexação de diretório em /config, evidenciado por nikto.png, vazamento de arquivos de configuração.

8.3 Oportunidades

- Implementar HTTPS com Let’s Encrypt para proteger tráfego e habilitar HSTS.
- Remover repositório .git do webroot e mover backups para armazenamento seguro (S3 com políticas restritas ou volumes não expostos).
- Implementar cabeçalhos de segurança (CSP, X-Frame-Options, X-Content-Type-Options, Referrer-Policy).
- Configurar firewall/SG para restringir acesso a MySQL apenas a redes/hosts autorizados.
- Adotar WAF leve e monitoramento/alertas para detectar varreduras massivas e acessos a backup/.git.

8.4 Ameaças

- Scanners automatizados e agentes maliciosos que exploram caminhos expostos em robots.txt e endpoints encontrados.
- Vazamento de credenciais/sensíveis se backups ou .git estiverem acessíveis.
- Exploração de vulnerabilidades conhecidas em versões desatualizadas de PHP/Apache (identificadas via headers).
- Exposição de bases de dados e serviços sem restrição de IP que permitem exploração remota.

9. Conclusão

- A verificação técnica inicial revela uma superfície de ataque significativa para o alvo 98.95.207.28. Os achados mais críticos são a exposição de caminhos sensíveis (robots.txt apontando para backup/.git) e a presença de serviços sensíveis acessíveis publicamente (MySQL, FTP). A ausência de HTTPS e a falta de cabeçalhos de segurança aumentam o risco de interceptação e abuso de sessões. Algumas configurações mostram controle inicial (porta 22 filtrada), mas não são suficientes para mitigar os riscos identificados. Recomenda-se tratar de forma prioritária a remoção de backups e repositórios expostos, restringir e isolar o acesso aos serviços de banco de dados, habilitar TLS e corrigir headers e flags de cookies. Se houver evidência confirmada de SQLi (sqlmap.png), isso requer ação imediata e investigação do endpoint afetado.

10. Sugestões para o Contratante

Prioridade Imediata

- Remover arquivos de backup e diretórios sensíveis do webroot público.
- Retirar o diretório .git do diretório público do servidor; posicionar repositório fora do webroot e utilizar deploys que não deixem .git acessível.

Alta prioridade

- Habilitar HTTPS (certificados válidos, p.ex. Let’s Encrypt) e forçar redirecionamento HTTP→HTTPS; aplicar HSTS.
- Restringir acesso ao MySQL (porta 3306) via firewall / Security Group (permitir apenas hosts/serviços internos autorizados) ou mover banco para rede privada.
- Remover ou configurar FTP apropriadamente (usar SFTP/SSH com autenticação forte) ou desativar se não for necessário.
- Corrigir cabeçalhos de segurança: adicionar X-Frame-Options, X-Content-Type-Options, Content-Security-Policy, Referrer-Policy; marcar cookies com HttpOnly e Secure.

Médio prazo

- Atualizar PHP e componentes do servidor (migrar de PHP 7.4 para uma versão suportada com correções de segurança); atualizar Apache/OS conforme necessidade.
- Implementar WAF e regras básicas para identificar e bloquear varreduras/fuzzing automatizadas (mitigar ferramentas como gobuster).

Recomendações operacionais contínuas

- Implementar política de backups segura (armazenamento cifrado, acesso controlado, não expor via HTTP).
- Integrar logs com SIEM/central de logs e habilitar alertas (acessos anômalos a /backup, /admin, .git, tentativas de SQLi).
- Treinamento de desenvolvedores sobre não commitar segredos e usar mecanismos de secrets management.
- Cronograma de patching e vulnerabilidade com prioridade para serviços expostos.