Serviço Público Federal
Ministério da Educação
Fundação Universidade Federal de Mato Grosso do Sul

ATIVIDADE ORIENTADA A ENSINO

Segurança Computacional com foco em Pentest

Identificação

Atividade Orientada a Ensino

Acadêmico: Pedro Augusto Paião Figueiredo

RGA: 2016 1907 0260

Professor: Carlos Alberto da Silva

Atividade: Atividade Orientada a Ensino sobre Segurança computacional (Pentest + OSINT)

1. Introdução

Esta atividade orientada a ensino aplica, em ambiente controlado, conceitos do TCC “Superfície de ataque em SaaS: impacto do OSINT na autenticação”. O objetivo pedagógico é demonstrar como fontes abertas (OSINT) reduzem o espaço de busca de identificadores de conta e como controles de autenticação (mensagens, rate limiting, MFA) influenciam o risco prático.

2. Objetivos de Aprendizagem

- Empregar OSINT para inferir padrões de e-mail institucionais.
- Mapear e validar endpoints de autenticação com ferramentas de inspeção.
- Conduzir tentativas controladas e éticas de login com telemetria.
- Avaliar consistência de mensagens, presença de rate limiting e esforço temporal.
- Propor mitigações mensuráveis e alinhadas a boas práticas (OWASP/NIST/ANPD).

3. Escopo Ético e Autorização

Escopo restrito: 1 conta autorizada + 1 controle inexistente para observar mensagens/códigos.

Sem exaustão: amostras curtas; sem contornar WAF/CDN; sem dumps privados.

Registro: coleta de evidências (prints, CSV, fluxos) e descarte seguro pós-aula.

4. Ambiente de Laboratório

- **Virtualização:** Oracle VirtualBox; VM com Kali Linux.
- **Rede:** uso de proxy HTTPS local para inspeção; sem varreduras intrusivas no “origin”.
- **Reprodutibilidade:** scripts versionados e resultados em CSV.

5. Ferramentas Utilizadas

- Fiddler / Burp / mitmproxy: interceptação TLS e mapeamento de rotas de autenticação.
- Censys / CT logs / buscadores: OSINT para padrões de e-mail e superfície exposta.
- curl/httpie, jq, grep/awk: checagens rápidas, extração e sumarização.
- Python 3 + requests: script de tentativas controladas com concorrência por threads.
- Crunch: geração de wordlists determinísticas (amostras).
-

6. Metodologia

Análise do que funcionou bem, pontos de melhoria, e ajustes previstos para próximas turmas.

Passo 1 — Proxy HTTPS e mapeamento

Inspeção de chamadas do app e identificação de /api/auth/login e /api/auth/recovery-password (método, headers, corpo, padrões de resposta).

A Figura 1 mostra a captura de uma requisição para o endpoint de recuperação de senha /api/auth/recovery-password, destacando o método HTTP, a URL completa e o corpo JSON contendo apenas o campo "email"

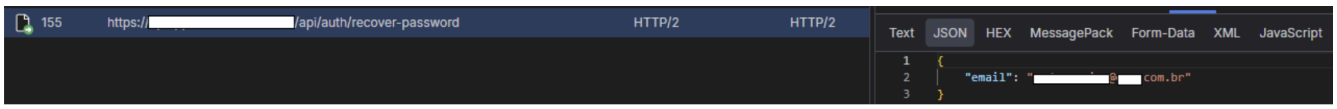

Figura 1. Rota de recuperar senha com o corpo enviado (/api/auth/recovery-password).

A Figura 2 apresenta a requisição para o endpoint de login /api/auth/login, na qual, além do "email", são enviados os demais parâmetros de autenticação (por exemplo, senha e tipo de grant). Essas capturas serviram de base para reproduzir o mesmo padrão de chamadas nos testes controlados com outras ferramentas.

controlados com outras ferramentas.

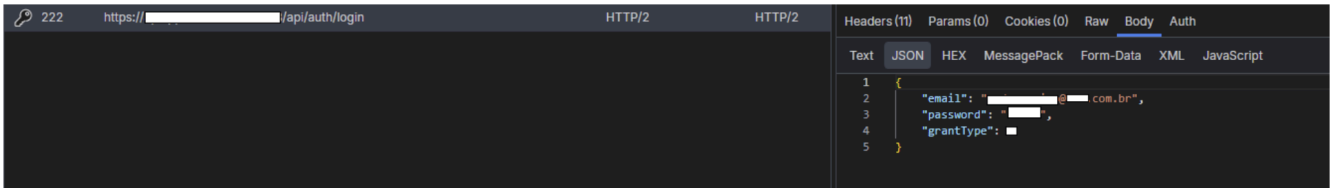

Figura 2. Rota de login com o corpo enviado (/api/auth/login).

Passo 2 — Reconhecimento (OSINT)

Navegação por buscadores, CT logs e Censys para inferir padrão institucional de e-mail, camada WAF/CDN e metadados públicos.

Após o mapeamento de endpoints, foi realizado um reconhecimento da camada de exposição para caracterizar o front do domínio e verificar a presença de WAF/CDN. A sequência adotada incluiu varredura com nmap (portas/serviços e metadados TLS/SNI), consulta a whois (propriedade/ASN) e busca no Censys (OSINT) para levantar possíveis IPs candidatos por correlação de certificado/hostname.

Procedimento.

1. **nmap:** confirmação de portas expostas e coleta de metadados.
2. **whois:** verificação de propriedade/ASN, com indicação de Cloudflare como WAF/CDN.
3. **Censys:** correlação por certificados/hostnames para identificação de IPs candidatos.

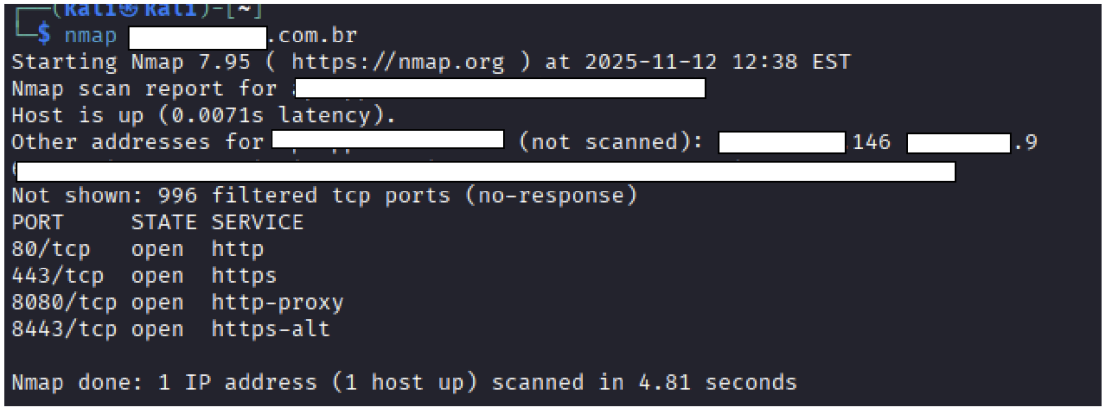

Figura 3. nmap: portas/serviços e metadados TLS/SNI observados para o domínio.

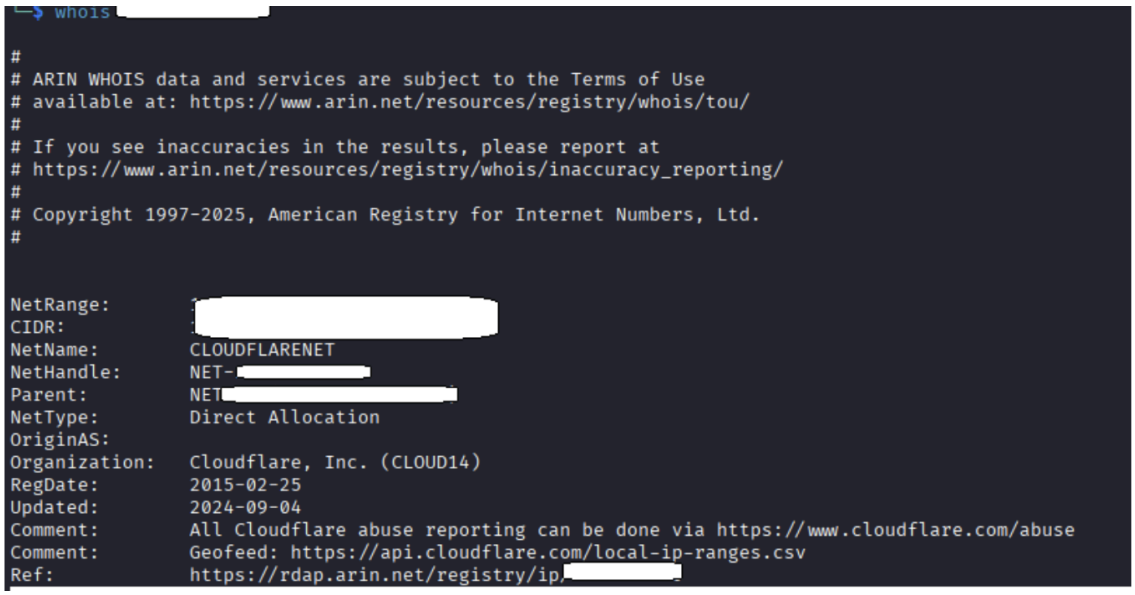

Figura 4. whois: evidência de CDN/WAF (Cloudflare) como camada de exposição.

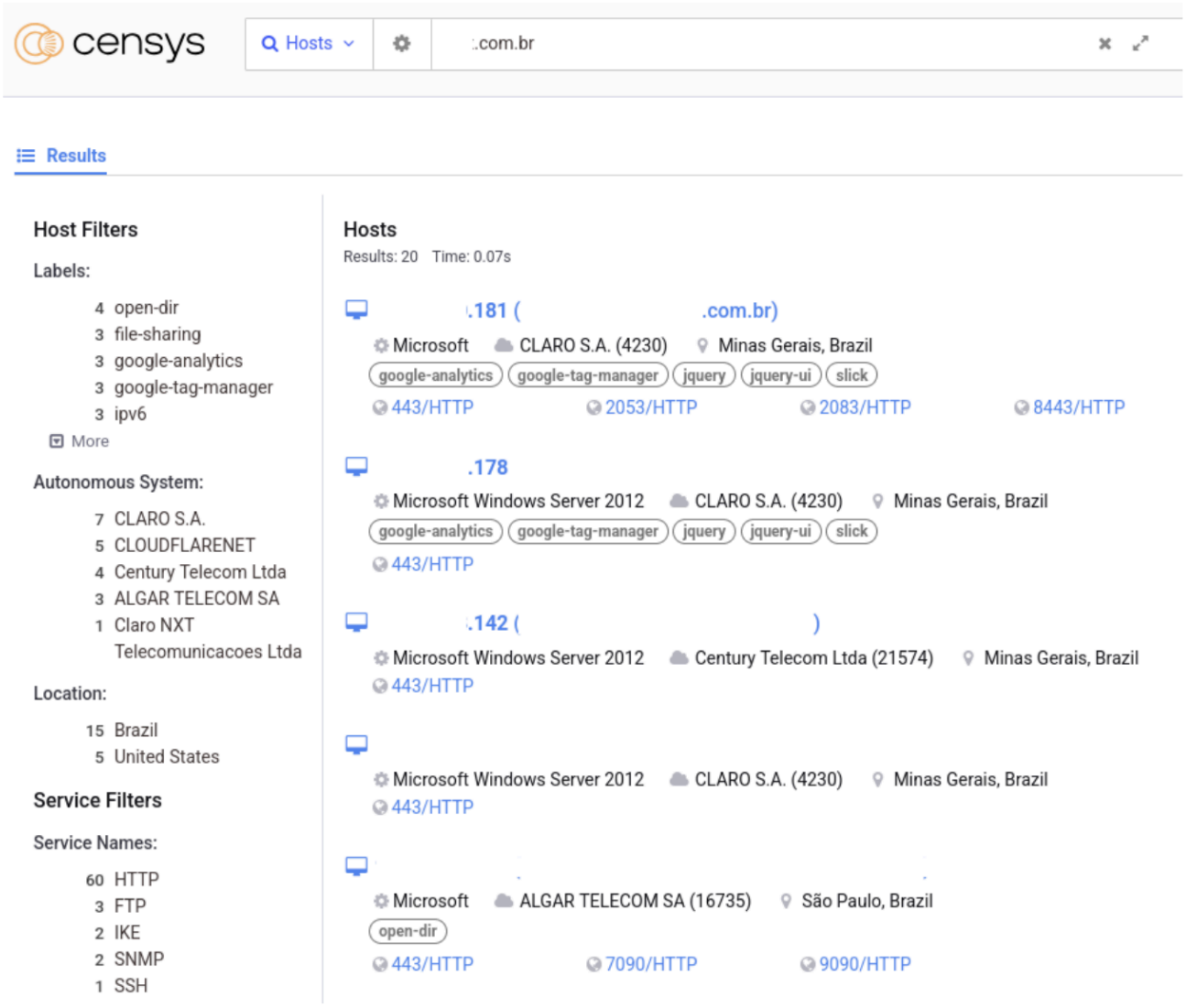

Figura 5. Censys: correlação por certificado/hostname indicando possíveis IPs candidatos

Passo 3 — Validação controlada

Reprodução da requisição de login via IP/hostname em cliente HTTP, comparando códigos/latências com o fluxo original.

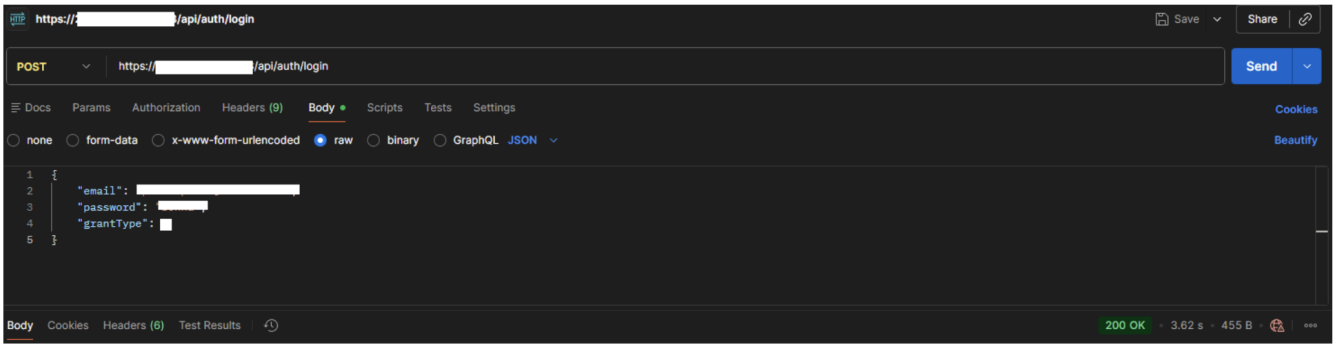

Figura 6. Postman — requisição POST para https://:/api/auth/login com Content-Type: application/json e corpo equivalente ao mapeado no Fiddler.

Passo 4 — Obtenção de identificadores (OSINT)

Coleta passiva de endereços e seleção de 1 e-mail autorizado para os testes; um e-mail inexistente serve de controle.

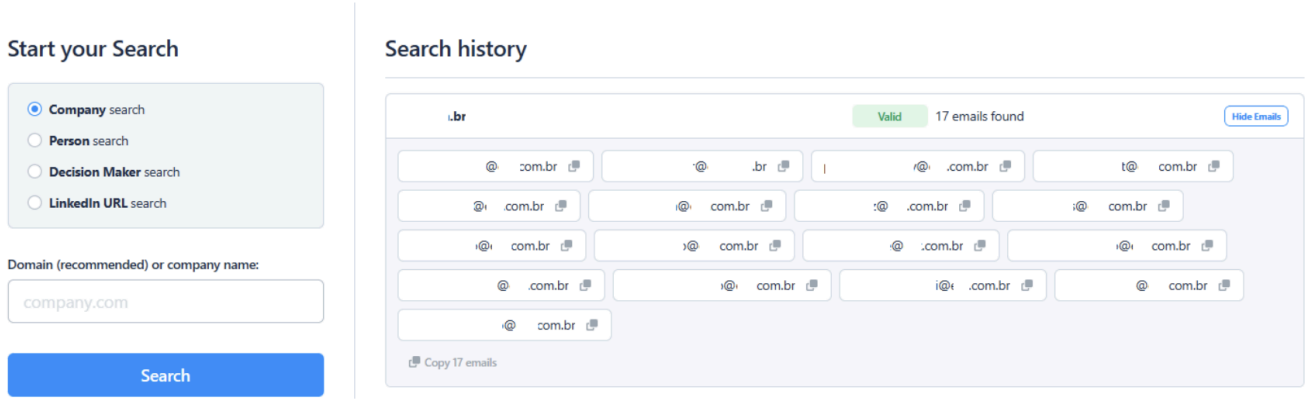

Figura 7. Exemplo de consulta OSINT para extração de padrões de e-mail no domínio de interesse.

Passo 5 — Recovery

Comparação de respostas do /recovery-password (válido vs. inexistente) para avaliar granularidade e potencial de enumeração.

A Figura 8 ilustra o teste controlado realizado sobre o endpoint de recuperação de senha /api/auth/recovery-password. Na parte superior da captura, observa-se a requisição HTTP contendo apenas o campo "email" no corpo JSON. Na parte inferior, são exibidas as respostas retornadas pela aplicação para diferentes cenários (endereço cadastrado e endereço inexistente), com variações tanto no conteúdo da mensagem quanto nos campos estruturados da resposta. Essa diferença de granularidade fornece ao atacante um canal de enumeração de contas, permitindo inferir se um determinado e-mail está ou não registrado no sistema a partir do texto de erro e dos códigos retornados.

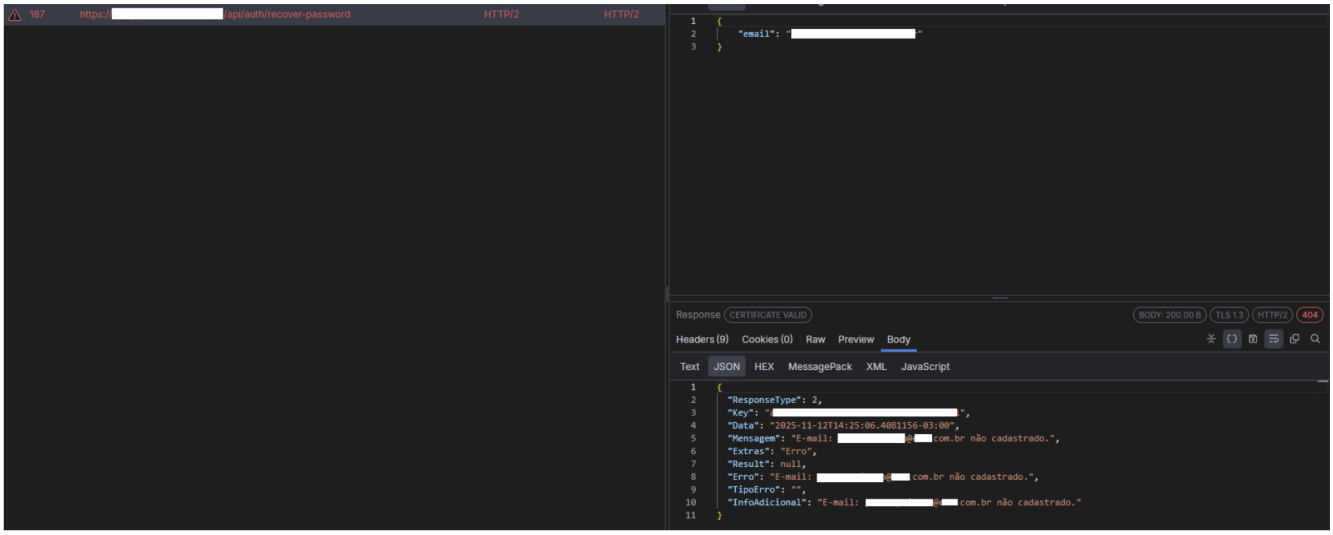

Figura 8. Evidência de granularidade de mensagens no endpoint /api/auth/recovery-password (exemplo sanitizado).

Passo 6 — Login com script fixo (threads)

Execução de amostra: threads concorrentes lendo wordlist curta; registro por tentativa: timestamp, código, latência e snippet em CSV; resumo por categorias (200/401/403/429/5xx).

```
CONST URL      = "https://<host>/api/auth/login"
CONST EMAIL    = "usuario_autorizado@dominio"
CONST WORDLIST = "wordlist_8digitos.txt"
CONST THREADS  = 8
CONST OUT_CSV  = "login_results.csv"

fila_senhas = carregar_linhas(WORDLIST)
iniciar_csv(OUT_CSV, "timestamp,attempt,http_status,elapsed_ms,snippet")

contador = 0
função worker():
  enquanto houver senha em fila_senhas:
    senha = puxar_proxima()
    ini = agora()
    resp = POST(URL, json={email: EMAIL, password: senha})
    fim = agora()
    escrever_csv(OUT_CSV, [ini, ++contador, resp.status, millis(fim-ini), trecho(resp.mensagem)])
    se resp.status == 200: sinalizar_sucesso_e_parar_todos()

lançar THREADS workers
aguardar_todas
resumir_por_codigo(OUT_CSV)
```

7. Resultados (resumo didático)

- **Recovery:** respostas distintas (ex.: 200 para e-mail válido vs. 401 para inexistente) → canal de enumeração.
- **Login (amostra):** predominância de 401; 0 ocorrências de 429 na janela observada; 1 retorno 200 isolado (não conclusivo).
- **Esforço prático:** sem reação defensiva explícita na amostra curta; latência estável.

8. Evidências

- Capturas do Fiddler/Burp (requisições e respostas sanitizadas).

- Fluxograma do procedimento do script.
- Saída do terminal com resumo por código HTTP e caminho do CSV.
- Excertos de CSV (5–10 linhas) e gráfico simples (barra) de códigos.

9. Discussão

- Impacto do OSINT: reduzir o espaço de busca de identificadores encurta o caminho até o ponto de autenticação.
- Superfície de login: mensagens não uniformes e ausência de limites claros elevam o risco de enumeração/tentativas repetidas.
- Medibilidade: métricas como tempo até 429, tentativas até bloqueio, variação de latência e taxa de indistinguibilidade das respostas permitem acompanhar evolução.

9. Mitigações Recomendadas

- Mensagens uniformes em login/recovery (indistinguíveis para falhas).
- Rate limiting com janela deslizante por conta e origem (IP/ASN), atraso progressivo e bloqueio temporário.
- MFA por risco e checagem contra senhas comprometidas.
- Higiene de exposição pública: reduzir previsibilidade de e-mails e remover artefatos acessíveis.
- Telemetria de abuso: dashboards de códigos/latências, bursts por ASN, alertas de spraying/stuffing.

7. Assinaturas

Documento assinado digitalmente
PEDRO AUGUSTO PAIAO FIGUEIREDO
Data: 25/11/2025 19:48:03-0300
Verifique em https://validar.iti.gov.br

Acadêmico(a): _____

Professor(a): _____