Dos Estudos em Laboratório a Aplicação Prática em Análise Não Invasiva de Segurança

integração entre atividades formativas, estudo de caso e evidências coletadas

Matheus Clisman Mariano da Silva[1], Orientador: Prof. Dr. Carlos Alberto da Silva[2]

[1]Curso de Bacharelado em Engenharia de Computação

[2]Faculdade de Computação (FACOM)

Universidade Federal de Mato Grosso do Sul (UFMS)

Av. Costa e Silva, s/nº – Bairro Universitário, CEP: 79070-900 – Campo Grande – MS

matheus.clisman@ufms.br, carlos.silva@ufms.br

Resumo

Este artigo integra três frentes de trabalho desenvolvidas ao longo da atividade acadêmica: o relatório simplificado dos estudos práticos, o relatório técnico complementar do estudo de caso e o conjunto de evidências objetivas coletadas durante a análise. O objetivo é demonstrar não apenas quais tópicos foram estudados, mas principalmente o que foi acrescentado a partir desses estudos em uma análise não invasiva do sistema de certificados da UFMS. A etapa formativa envolveu reconhecimento de rede com Nmap, interceptação e análise de requisições com Burp Suite, estudo de XSS e IDOR, quebra de hash com John the Ripper e leitura de alertas e logs em ambiente de monitoramento. A partir desse repertório, foi estruturada uma coleta de evidências de baixo impacto sobre o SICERT, com foco em headers HTTP, cookies de sessão, comportamento do navegador, prova local de clickjacking e verificação de logout com sessão própria do avaliador. Os resultados mostraram que o principal acréscimo dos estudos foi a capacidade de transformar conteúdos de laboratório em um procedimento reprodutível de observação, classificação e recomendação técnica, sustentado por evidências reais e por limites éticos bem definidos. Os comandos e trechos de código mais relevantes foram reunidos ao final do documento, em anexos técnicos suplementares ao corpo principal do artigo.

Palavras-chave: segurança web; análise não invasiva; formação prática; evidências; SICERT.

1 Introdução

O desenvolvimento de competências em segurança da informação depende de uma combinação equilibrada entre estudo conceitual, experimentação em laboratórios controlados e capacidade de interpretar sinais técnicos em sistemas reais. Quando esse processo é bem feito, o estudante deixa de apenas reproduzir ferramentas e passa a compreender por que determinados comportamentos de rede, navegador e aplicação indicam maior ou menor maturidade de segurança.

No relatório simplificado, o percurso formativo foi documentado com evidências visuais dos estudos realizados em TryHackMe [1] e LetsDefend [2]. O relatório técnico complementar registra a etapa de maior densidade técnica, centrada em uma análise não invasiva do sistema de certificados da UFMS. O conjunto de evidências coletadas concentra os artefatos brutos, permitindo verificar como os achados descritos no texto se apoiam em comandos, saídas e registros concretos.

Este artigo une essas três camadas, mostrando a continuidade entre elas: quais conhecimentos foram adquiridos, como foram utilizados e o que foi efetivamente acrescentado. O texto valoriza tanto a etapa de formação quanto sua tradução em metodologia, evidências e recomendações técnicas. Os trechos de comando e código que sustentam essa passagem foram reproduzidos no material suplementar [11].

Este trabalho se insere em uma linha de pesquisa da Faculdade de Computação da UFMS sobre segurança de aplicações web. Fernandes e Silva [8], Vitorino e Silva [9] e Paião e Silva [10] conduziram investigações de vulnerabilidades em sistemas web reais, utilizando técnicas de pentest e ferramentas automatizadas. O presente artigo evolui nessa direção, aplicando uma metodologia não invasiva e eticamente delimitada sobre um sistema institucional real.

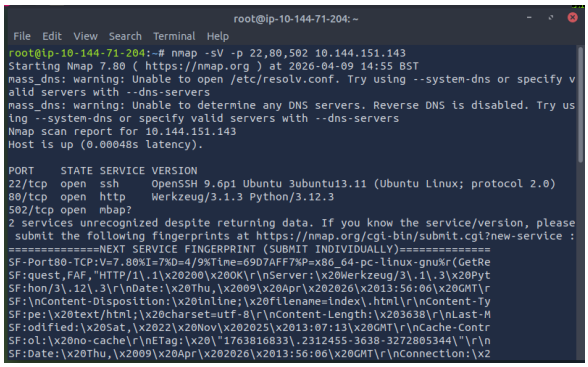
2 Base de Estudos Desenvolvida em Laboratórios Práticos

Os estudos iniciais se concentraram em cinco etapas complementares, relatados na Figura 1. A primeira foi o reconhecimento de rede com Nmap, importante para compreender identificação de serviços, superfícies expostas e a lógica de enumeração de portas e respostas. A segunda foi a análise de requisições HTTP com *Burp Suite*, utilizada para entender a estrutura de cabeçalhos, parâmetros, métodos e fluxos de comunicação entre cliente e servidor.

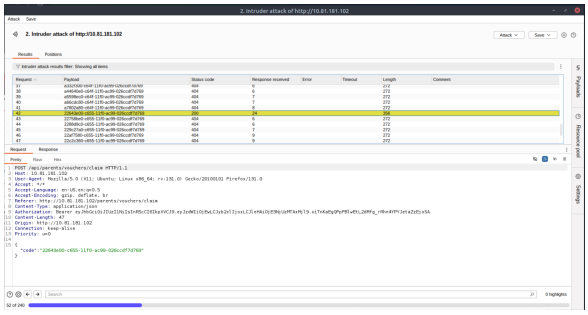
A terceira frente foi o estudo de vulnerabilidades *web*, especialmente *Cross-Site Scripting* (XSS) e *Insecure Direct Object Reference* (IDOR). Esses laboratórios não foram

aproveitados como roteiro de exploração no caso real; eles serviram como base para reconhecer a importância de validações do lado servidor, de controles de navegador e da diferença entre observação técnica e tentativa ativa de abuso. A quarta frente foi a quebra de *hash* com a ferramenta John the Ripper, que ampliou a compreensão sobre sensibilidade de credenciais, tratamento de segredo e impacto de exposições indevidas. A quinta frente foi a análise de alertas e logs em ambiente de monitoramento, com apoio da LetsDefend e do Splunk, o que contribuiu para uma postura mais sistemática diante de evidências técnicas.

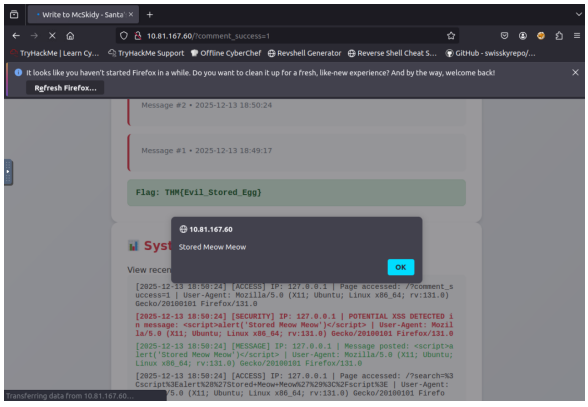
Em conjunto, essas atividades deram origem a um repertório prático que vai além do uso isolado de ferramentas. O principal ganho foi a capacidade de formular perguntas técnicas corretas diante de uma aplicação web: quais controles devem aparecer na resposta HTTP, que tipo de cookie está sendo emitido, como o navegador deve ser protegido contra enquadramento indevido, quais dados são realmente necessários no front-end e quais verificações podem ser feitas sem extrapolar limites éticos.



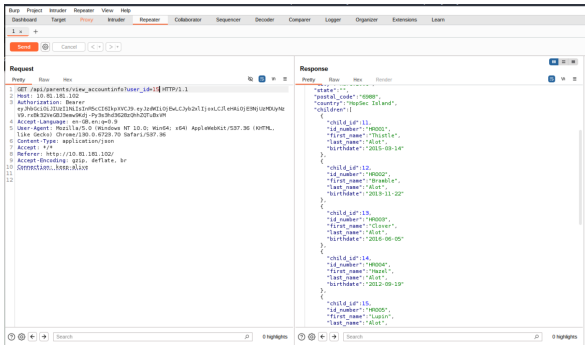
Reconhecimento de rede com Nmap



Análise de requisições com Burp Suite



Estudo de XSS em laboratório controlado



Estudo de IDOR com apoio do Burp Suite

Figura 1: Recorte das atividades formativas que serviram de base para a etapa aplicada do trabalho.

3 O que foi Acrescentado a Partir dos Estudos

O ponto central deste trabalho não está apenas no fato de ter havido estudos prévios, mas em como esses estudos permitiram acrescentar procedimentos concretos a uma análise aplicada. A partir do que foi desenvolvido no relatório simplificado, a etapa aplicada passou a incorporar elementos que não apareciam no texto inicial, mas que resultam diretamente da maturação obtida nos experimentos laboratórios.

Os principais acréscimos foram os seguintes:

1. definição de um escopo ético explícito para análise não invasiva, sem exploração, sem fuzzing, sem brute force e sem acesso a dados de terceiros, em conformidade com os princípios de minimização e finalidade previstos na Lei Geral de Proteção de Dados (LGPD) [4];
2. elaboração de um roteiro de coleta reproduzível, com comandos shell, delimitação de escopo e registro de metadados, reproduzido integralmente nos Anexos A e B;
3. organização das evidências em diretório dedicado, com *hashes* (valores únicos gerados por função criptográfica, utilizados para verificar a integridade dos arquivos coletados), *headers* (cabeçalhos HTTP enviados pelo servidor em cada resposta), *fingerprint* (conjunto de características técnicas que permitem identificar tecnologias utilizadas pela aplicação) e arquivos auxiliares (registros complementares de apoio à análise, como metadados e logs de execução dos scripts);
4. avaliação técnica de cookies, headers de segurança, recursos externos e comportamento de navegação autenticada com sessão própria;
5. construção de prova de conceito local de carregamento em *iframe*, cujo código essencial é apresentado no Anexo D;
6. classificação dos achados com base em CWE [5] e OWASP [3], acompanhada de recomendações técnicas objetivas.

Esses acréscimos mostram uma mudança importante de nível. Enquanto os estudos em laboratório tiveram como foco a aprendizagem de técnicas e conceitos, a etapa de análise técnica estruturada passou a exigir disciplina metodológica, rastreabilidade de evidências, delimitação ética e capacidade de transformar observações pontuais em diagnóstico técnico sustentado.

4 Aplicação Prática no SICERT com Base em Evidências

4.1 Escopo, coleta e organização do material

A aplicação prática concentrou-se no domínio https://certificados.ufms.br/ e em rotas associadas, sem uso de técnicas agressivas – isto é, sem varreduras de força bruta, *fuzzing*, envio de *payloads* de exploração ou qualquer ação que pudesse alterar o estado da aplicação ou indisponibilizar o serviço. Foram empregadas exclusivamente técnicas de observação passiva e de baixo impacto, como requisições HTTP padrão e leitura de cabeçalhos de resposta. O material bruto foi organizado em um conjunto de evidências composto por dumps de *headers*, corpos HTML, metadados, resumo de verificação, fingerprint com `whatweb`, capturas suplementares de validação externa e informações complementares de TLS. Os trechos mais relevantes desse conjunto, assim como os scripts que os geraram, foram reproduzidos nos Anexos A a F do material suplementar[1], mantidos fora do corpo principal para preservar a extensão acadêmica do artigo.

Essa estrutura de coleta representa um dos principais ganhos metodológicos trazidos pelos estudos anteriores. Em vez de depender apenas de observação manual, o trabalho passou a contar com um conjunto de evidências reproduzíveis, com horário registrado, comandos verificáveis e possibilidade de conferência posterior. Isso aproxima o texto acadêmico de uma prática mais próxima de auditoria técnica, ainda que em escopo limitado e não invasivo.

4.2 Evidências técnicas principais

O resumo de verificação de *headers* registrou, entre outros pontos, a presença de HSTS (*HTTP Strict Transport Security*, cabeçalho que obriga o navegador a se comunicar apenas via HTTPS), política de *cache* restritiva (que impede o armazenamento local de respostas sensíveis pelo navegador) e cookie com o atributo `HttpOnly` (que impede o acesso ao cookie por meio de JavaScript no lado do cliente), bem como a ausência de cabeçalhos importantes de endurecimento do navegador. O trecho apresentado na Listagem 1 sintetiza a verificação executada durante a coleta, retomada com mais contexto no Anexo C:

Listing 1: Resultado da verificação dos cabeçalhos de segurança HTTP

```
[PRESENTE] Strict-Transport-Security
[AUSENTE] Content-Security-Policy
[AUSENTE] X-Frame-Options
```

[1]Os anexos técnicos mencionados ao longo deste artigo estão reunidos no material suplementar, disponível em: https://github.com/cl1sman/material-suplementar-tcc/blob/main/material_suplementar.pdf.

```
[AUSENTE] X-Content-Type-Options
[AUSENTE] Referrer-Policy
[AUSENTE] Permissions-Policy

[PRESENTE] HttpOnly
[AUSENTE] Secure
[AUSENTE] SameSite
```

Como validação visual suplementar, duas capturas geradas no *SecurityHeaders.com* [7] foram incorporadas ao conjunto documental. A primeira resume a avaliação do domínio com nota D – classificação que, na escala de A+ a F adotada pela ferramenta, indica configuração parcial dos cabeçalhos de segurança, com lacunas relevantes [7] – e destaca a presença de `Strict-Transport-Security` e a ausência de `Content-Security-Policy`, `X-Frame-Options`, `X-Content-Type-Options`, `Referrer-Policy` e `Permissions-Policy`. A segunda reproduz a seção *Raw Headers* do serviço, com valores coerentes com os cabeçalhos obtidos pela coleta própria. Essas imagens, reunidas no Anexo E, não substituem a verificação por comandos; elas apenas reforçam visualmente a consistência dos resultados.

O fingerprint leve obtido com `whatweb -a 1` reforçou a identificação do cookie `CAKEPHP`, da presença de HSTS e do uso de PHPCake, conforme evidencia na Listagem 2 e na saída ampliada reproduzida no Anexo F:

Listing 2: Fingerprint da aplicação obtido com WhatWeb

```
https://certificados.ufms.br/ [200 OK] Cookies[CAKEPHP],
HttpOnly[CAKEPHP], PHPCake,
Strict-Transport-Security[max-age=31536000; includeSubDomains]
```

Também foi preservado o contexto TLS do serviço, com certificado emitido para `*.ufms.br` por `GlobalSign RSA OV SSL CA 2018`, válido até outubro de 2026. O resumo dessa evidência também foi reunido no Anexo F. Embora esse dado não elimine os achados de configuração HTTP, ele ajuda a situar o serviço em um ambiente real de produção e reforça a observação de que a postura de segurança não é uniforme: alguns controles estão corretamente presentes, enquanto outros ainda exigem ajuste.

4.3 Sessão própria, logout e prova de conceito local

Outro acréscimo relevante em relação ao relatório simplificado foi a capacidade de analisar o comportamento de uma sessão autenticada sem romper o critério ético do trabalho. Na etapa aplicada, a rota `/home` foi verificada com cookie obtido legitimamente em sessão própria do avaliador, o que permitiu confirmar que o `CAKEPHP` atua como identificador de sessão autenticada. Em seguida, o mesmo cookie foi retestado após logout, retornando HTTP `302`, sem evidência de permanência indevida da sessão. Os comandos centrais dessa

validação foram reunidos no Anexo C.

Em paralelo, a prova de conceito local de *clickjacking* – técnica de ataque em que uma página é carregada de forma oculta dentro de um *iframe* em outro site, induzindo o usuário a clicar em elementos sem perceber sua real função – mostrou que a página pública podia ser carregada dentro de um *iframe* sem bloqueio explícito do navegador. O código completo dessa prova de conceito foi deslocado para o Anexo D, juntamente com a captura de tela correspondente, mantendo no corpo do artigo apenas sua função metodológica: demonstrar, de forma controlada, a ausência de controles de enquadramento. Esse tipo de verificação exemplifica bem o que foi acrescentado a partir dos estudos: não se trata de explorar a aplicação, mas de usar conhecimento técnico para produzir evidências pequenas, objetivas e defensáveis sobre comportamentos observáveis.

5 Discussão dos Principais Achados

A Tabela 1 resume a relação entre as etapas estudadas em laboratório e os acréscimos observados na etapa aplicada, servindo de referência para a discussão apresentada nas subseções seguintes.

5.1 Cookie de sessão sem `Secure` e `SameSite`

O achado mais relevante da análise foi a emissão do cookie `CAKEPHP` apenas com `HttpOnly`, sem os atributos `Secure` e `SameSite`, conforme evidenciado na Listagem 1. Esse resultado importa porque o mesmo cookie foi associado a conteúdo autenticado quando usado em sessão própria válida. A ausência de `Secure` enfraquece a proteção explícita do cookie no contexto HTTPS, enquanto a falta de `SameSite` reduz a defesa padrão contra envio automático em cenários *cross-site*. Em classificação, o caso se relaciona a *CWE-614*, *CWE-1275* e OWASP Top 10 2021 A05.

5.2 Ausência de políticas de proteção do navegador

As evidências também mostraram ausência de cinco políticas importantes de proteção do navegador: `Content-Security-Policy`, `X-Frame-Options`, `X-Content-Type-Options`, `Referrer-Policy` e `Permissions-Policy`. Em termos práticos, isso significa que o navegador recebe menos orientações defensivas do que poderia. A ausência de CSP não comprova XSS, mas remove uma camada importante de mitigação. Já a falta de `X-Frame-Options` e de `frame-ancestors` foi coerente com a prova local de enquadramento em *iframe*, reforçando o risco de clickjacking.

Os demais headers ausentes compoem um bloco de endurecimento complementar.

O header `X-Content-Type-Options` reduz interpretação indevida de tipos MIME. O `Referrer-Policy` controla informações de referência enviadas a outros domínios. Já o `Permissions-Policy` restringe APIs sensíveis do navegador. Isoladamente, alguns desses itens podem parecer secundários, mas em conjunto eles indicam oportunidade clara de revisão de configuração em camada comum da aplicação.

5.3 Pontos positivos e maturidade parcial

Apesar dos achados, a coleta também evidenciou controles positivos. O sistema utiliza HSTS com `includeSubDomains`, políticas restritivas de cache e cookie com `HttpOnly`. Além disso, não foram observados os headers `Server` e `X-Powered-By`, o que reduz exposição desnecessária de tecnologia. O teste de logout com retorno HTTP `302` também sugere comportamento adequado de invalidação de sessão no fluxo observado.

Essa combinação de ausências e presenças é importante para a interpretação do estudo. O sistema não se apresenta como um ambiente sem controles; ao contrário, ele demonstra maturidade parcial. Justamente por isso, as recomendações propostas são pontuais e realistas: tratam de complementar uma base que já existe, e não de reconstruir a segurança da aplicação do zero. A Tabela 1 sintetiza essa progressão entre o que foi estudado e o que foi efetivamente acrescentado.

Tabela 1: Síntese do que foi estudado e do que foi acrescentado

Base no relatório formativo	**Acréscimo observado no estudo de caso e nas evidências coletadas**
Reconhecimento e enumeração em laboratório	delimitação de alvos, fingerprint leve e leitura de superfície exposta sem varredura agressiva
Análise de requisições com Burp Suite	verificação de headers HTTP, comportamento de cookies e interpretação de respostas do SICERT
Estudos de XSS e IDOR	valorização de controles preventivos como CSP, frame protection e cuidado ético para não explorar a aplicação
Quebra de hash e sensibilidade de credenciais	maior atenção ao papel do cookie de sessão como credencial sensível
Análise de alertas e logs	organização de evidências, rastreabilidade e leitura sistemática dos artefatos coletados

6 Recomendações Técnicas

Com base nas evidências coletadas, as recomendações principais são as seguintes:

- configurar o cookie de sessão com `Secure; HttpOnly; SameSite=Lax`, avaliando `SameSite=Strict` quando compatível;
- implementar `Content-Security-Policy` inicialmente em modo de relatório, para posterior endurecimento em produção;
- bloquear enquadramento por terceiros com `frame-ancestors` e `X-Frame-Options`;
- habilitar `X-Content-Type-Options: nosniff`;
- definir `Referrer-Policy: strict-origin-when-cross-origin`;
- restringir APIs não utilizadas por meio de `Permissions-Policy`;
- revisar a quantidade de dados renderizados no JavaScript da área autenticada e consolidar os controles em camada comum da aplicação.

7 Conclusão

O trabalho integrado entre o relatório formativo, o estudo de caso técnico e o conjunto de evidências coletadas permite uma leitura mais completa do processo acadêmico desenvolvido. A etapa inicial mostrou o que foi estudado em laboratórios práticos. A etapa seguinte revelou o que esses estudos tornaram possível acrescentar: definição de escopo ético, coleta reproduzível, organização de evidências, leitura técnica de cookies e headers, validação de comportamento autenticado com sessão própria e formulação de recomendações objetivas.

Assim, o valor do artigo não está apenas em listar achados do SICERT, mas em demonstrar uma progressão de aprendizagem. Os estudos anteriores não ficaram isolados como exercícios de plataforma; eles foram convertidos em critério técnico para observar uma aplicação real com responsabilidade, baixo impacto e sustentação documental. Esse é, em última análise, o principal resultado acrescentado a partir da base de estudos feita anteriormente.

Referências

[1] TRYHACKME. *TryHackMe*. Disponível em: https://tryhackme.com/. Acesso em: 1 dez. 2025.

[2] LETSDEFEND. *LetsDefend*. Disponível em: https://letsdefend.io/. Acesso em: 20 dez. 2025.

[3] OWASP FOUNDATION. *OWASP Top 10: The Ten Most Critical Web Application Security Risks*. 2021. Disponível em: https://owasp.org/www-project-top-ten/. Acesso em: 3 jun. 2026.

[4] BRASIL. *Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD)*. Diário Oficial da União, Brasília, DF, 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm. Acesso em: 10 jun. 2026.

[5] MITRE. *Common Weakness Enumeration*. Disponível em: https://cwe.mitre.org/. Acesso em: 3 maio 2026.

[6] MOZILLA. *HTTP security headers*. Disponível em: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers. Acesso em: 3 jun. 2026.

[7] HELME, Scott. *SecurityHeaders.com*. Disponível em: https://securityheaders.com/. Acesso em: 3 maio 2026.

[8] FERNANDES, Hatanael Lima; SILVA, Carlos Alberto da. *Investigação de Vulnerabilidades em Aplicações Web Utilizadas por Empresas de Leilões Online e Instituições de Ensino a Distância (EAD)*. Trabalho de Conclusão de Curso – Universidade Federal de Mato Grosso do Sul, Campo Grande, 2025. Disponível em: https://repositorio.ufms.br/handle/123456789/13210. Acesso em: 10 jun. 2026.

[9] VITORINO, Gabriel Augusto Ocampos; SILVA, Carlos Alberto da. *Análise de Vulnerabilidades em Domínios WordPress de Instituições de Ensino*. Trabalho de Conclusão de Curso – Universidade Federal de Mato Grosso do Sul, Campo Grande, 2025. Disponível em: https://repositorio.ufms.br/handle/123456789/12178. Acesso em: 10 jun. 2026.

[10] PAIÃO, Pedro Augusto; SILVA, Carlos Alberto da. *Superfície de ataque em SaaS: Impacto do OSINT na autenticação e mitigações*. Trabalho de Conclusão de Curso – Universidade Federal de Mato Grosso do Sul, Campo Grande, 2025. Disponível em: https://repositorio.ufms.br/retrieve/51c473d3-42a7-4230-b483-72ac6491c9a7/31967.pdf. Acesso em: 10 jun. 2026.

[11] SILVA, Matheus Clisman Mariano da. *Material Suplementar: Anexos Técnicos – Dos Estudos em Laboratório a Aplicação Prática em Análise Não Invasiva de Segurança*. 2026. Disponível em: https://github.com/cl1sman/material-suplementar-tcc/blob/main/material_suplementar.pdf. Acesso em: 10 jun. 2026.

Material Suplementar

Os anexos técnicos deste trabalho — incluindo os scripts de coleta, saídas de comandos, código da prova de conceito de clickjacking e capturas complementares de TLS e fingerprint — foram organziados em um documento separado, disponível publicamente no seguinte endereço:

https://github.com/cl1sman/material-suplementar-tcc/blob/main/material_suplementar.pdf

- **Anexo A** – Script completo de coleta não invasiva (`coleta_nao_invasiva_certificados_ufms.sh`);
- **Anexo B** – Script de verificação rápida de headers (`verificar_headers.sh`);
- **Anexo C** – Comandos e saídas centrais da verificação de headers, cookies e logout;
- **Anexo D** – Código da prova de conceito de clickjacking e captura visual do resultado;
- **Anexo E** – Capturas complementares do SecurityHeaders.com;
- **Anexo F** – Trechos de fingerprint com WhatWeb e verificação de certificado TLS.