UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
PROGRAMA DE RESIDÊNCIA EM TECNOLOGIA DA INFORMAÇÃO

Análise de Segurança Cibernética no Tribunal de Contas do Estado do Rio Grande do Norte: Aplicação dos Princípios OWASP na Identificação e Mitigação de Vulnerabilidades

José Edivandro de Sousa Júnior

Natal-RN, Brasil
2024

José Edivandro de Sousa Júnior

Análise de Segurança Cibernética no Tribunal de Contas do Estado do Rio Grande do Norte: Aplicação dos Princípios OWASP na Identificação e Mitigação de Vulnerabilidades

Trabalho de Conclusão de Curso apresentado ao Programa de Residência em Tecnologia da Informação do Instituto Metrópole Digital da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do título de Especialista em Tecnologia da Informação. Área de Concentração: Engenharia de Software

Orientador: Prof. Dr. Ramon dos Reis Fontes

Natal-RN, Brasil
2024

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Sousa Júnior, José Edivandro de.
Análise de segurança cibernética no Tribunal de Contas do Estado do Rio Grande do Norte: aplicação dos princípios OWASP na identificação e mitigação de vulnerabilidades / José Edivandro de Sousa Júnior. - 2024.
61f.: il.

Monografia (Especialização) - Universidade Federal do Rio Grande do Norte, Instituto Metrópole Digital, Programa de Residência em Tecnologia da Informação, Natal, 2024.
Orientação: Dr. Ramon dos Reis Fontes.

1. Segurança da informação - Monografia. 2. Vulnerabilidades - Monografia. 3. OWASP - Monografia. I. Fontes, Ramon dos Reis. II. Título.

RN/UF/BCZM CDU 004

Elaborado por Raimundo Muniz de Oliveira - CRB-15/429

José Edivandro de Sousa Júnior

Análise de Segurança Cibernética no Tribunal de Contas do Estado do Rio Grande do Norte: Aplicação dos Princípios OWASP na Identificação e Mitigação de Vulnerabilidades

Trabalho de Conclusão de Curso apresentado ao Programa de Residência em Tecnologia da Informação do Instituto Metrópole Digital da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do título de Especialista em Tecnologia da Informação. Área de Concentração: Engenharia de Software

Trabalho aprovado. Natal-RN, Brasil, 21 de fevereiro de 2024:

Prof. Dr. Ramon dos Reis Fontes
Orientador

Prof. Dr. Jean Mário Moreira de Lima
Examinador

Me. Frederico Nunes do Pranto Filho
Examinador

Natal-RN, Brasil
2024

Agradecimentos

Não tenho palavras para expressar o quanto este programa de residência transformou a minha vida. A forma como fui acolhido e orientado pela diretoria de informática do Tribunal de Contas do Estado do Rio Grande do Norte (DIN) é indescritível.

Quero agradecer, primeiramente, a Deus por esta oportunidade, e aos meus familiares e amigos que me apoiaram ao longo dessa jornada, em especial à minha mãe, Maria Dantas; ao meu pai, José Edivandro; e à minha namorada, Daniela Maia.

Também quero expressar minha gratidão de maneira geral a todos os servidores e terceirizados da DIN, em especial ao diretor Vinicius José, aos analistas Frederico Pranto e Guilherme Lucena e ao contador Cláudio Formiga. Lembrarei de cada ensinamento que tive com cada um de vocês.

Aos residentes que fizeram parte deste programa comigo, em especial a Renan Lima e Maria Eduarda, que foram meus mentores, minha eterna gratidão.

Por fim, ao meu orientador, Ramon Fontes, que me apoiou em todo o processo da produção deste trabalho.

"A segurança é tão forte quanto o elo mais fraco." (*Kevin Mitnick, 2000)*

Resumo

Muito se discorre acerca dos protocolos de segurança da informação, em virtude dos consideráveis incidentes de vazamentos de dados ao longo dos anos, fenômeno que, nas instituições públicas do Rio Grande do Norte, não destoa. Este trabalho de conclusão de curso (TCC) dedica-se ao Tribunal de Contas do Estado do Rio Grande do Norte (TCE/RN), explorando a aplicação dos princípios delineados pelo Open Web Application Security Project (OWASP) na análise de vulnerabilidades presentes nos sistemas desenvolvidos pelo referido tribunal. O OWASP, enquanto comunidade, concebe metodologias e ferramentas de forma gratuita, fundamentadas em incidentes pregressos, visando prevenir vulnerabilidades nas esferas pública e privada. O TCC em apreço avaliou a segurança do TCE/RN e a eficácia dos sistemas produzidos e mantidos pelos residentes, valendo-se da ferramenta Zed Attack Proxy, uma aplicação gratuita fundamentada nos protocolos preconizados pela OWASP, com o propósito de identificar e mitigar eventuais vulnerabilidades. Os objetivos específicos abrangem uma análise aprofundada da segurança cibernética, a catalogação das vulnerabilidades presentes nos sites do tribunal e a realização de análises minuciosas dessas vulnerabilidades no contexto de desenvolvimento. O desfecho da utilização da ferramenta de varredura de vulnerabilidades revelou falhas de segurança em todos os sistemas avaliados, conferindo validade à generalização da eficácia da ferramenta para todos os sistemas do tribunal. Tal constatação valida a importância intrínseca deste trabalho de conclusão de curso.

Palavras-chave: Segurança da Informação; vulnerabilidades; OWASP

Abstract

A great deal is discussed regarding information security protocols, due to the significant incidents of data leaks over the years, a phenomenon that is no different in public institutions in Rio Grande do Norte. This undergraduate thesis (TCC) is dedicated to the Court of Auditors of the State of Rio Grande do Norte (TCE/RN), exploring the application of principles outlined by the Open Web Application Security Project (OWASP) in analyzing vulnerabilities in systems developed by the aforementioned court. OWASP, as a community, develops methodologies and tools freely, based on past incidents, aiming to prevent vulnerabilities in both public and private spheres. The TCC in question assessed the security of TCE/RN and the effectiveness of systems produced and maintained by residents, using the Zed Attack Proxy tool, a free application based on protocols advocated by OWASP, with the purpose of identifying and mitigating potential vulnerabilities. Specific objectives include an in-depth analysis of cybersecurity, cataloging vulnerabilities on court websites, conducting detailed analyses of these vulnerabilities in a development context, and developing proactive plans to prevent new security vulnerabilities. The outcome of using the vulnerability scanning tool revealed security flaws in all assessed systems, validating the generalization of the tool's effectiveness for all court systems. Such findings affirm the intrinsic importance of this undergraduate thesis, which culminated in the development of a plan capable of mitigating these deficiencies.

Keywords: Information security; vulnerabilities; OWASP.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

IMD	Instituto Metrópole Digital
TCE/RN	Tribunal de Contas do Estado Rio Grande do Norte
UFRN	Universidade Federal do Rio Grande do Norte
OWASP	Open Web Application Security Project
ZAP	Zed Attack Proxy
LGPD	Lei Geral de Proteção de dados
ISO	International Organization for Standardization
ABNT	Associação Brasileira de Normas Técnicas
CORS	Cross-Origin Resource Sharing
URL	Uniform Resource Locator
NIST	National Institute of Standards and Technology
CWE	Common Weakness Enumeration

Sumário

1 Introdução

Na atual era de transformação digital, a rápida evolução das aplicações web tornou-se uma prioridade crítica para empresas, tanto do setor privado quanto instituições governamentais, a exemplo do Tribunal de Contas do Estado do Rio Grande do Norte (TCE/RN). A automação, a entrega contínua e a adoção da cultura *DevOps* prometem avanços significativos em termos de segurança de sistemas, mas também trazem consigo desafios de segurança à medida que novas aplicações são implantadas em ritmo acelerado.

Compreende-se que, interromper o processo de entrega de software ou recorrer à contratação de equipes especializadas em segurança pode acarretar em custos significativos para as organizações. No entanto, negligenciar a identificação precoce ou, a identificação tardia de vulnerabilidades, devido à velocidade frenética de desenvolvimento de software, pode resultar em prejuízos que superam em muito os investimentos necessários para aprimorar a segurança.

Neste contexto, a Fundação *Open Web Application Security Project* (OWASP), estabelecida em 2001, emerge como uma força crucial na aprimoração da segurança de software em escala global. Atuando como uma entidade sem fins lucrativos, a OWASP desempenha um papel central na conscientização sobre segurança cibernética e na elaboração de recursos e diretrizes essenciais para mitigar as ameaças inerentes às aplicações *web*.

Este trabalho de conclusão de curso teve como foco principal explorar a aplicação dos princípios orientadores da *Open Web Application Security Project* (OWASP) na análise das vulnerabilidades nos sistemas em produção do Tribunal de Contas do Rio Grande do Norte (TCE/RN) através da ferramenta *Zed Attack Proxy*, Avaliando a eficácia do sistema de segurança do TCE/RN

Espera-se que este trabalho represente uma contribuição significativa na incessante busca pela excelência em segurança de software e na defesa contra as ameaças cibernéticas em um mundo digital em constante evolução.

1.1 Motivação

A motivação subjacente a este trabalho de pesquisa está profundamente enraizada no desejo de contribuir para a proteção das informações do TCE/RN, tendo em vista que este é um órgão supervisor financeiro de todas as instituições públicas do estado e, portanto, possui informações e procedimentos sensíveis. Por meio deste estudo, busca-se uma contribuição significativa para o contínuo crescimento e segurança das informações mantidas pelo TCE/RN, ao mesmo tempo em que se reforça a conformidade com a Lei Geral de Proteção de Dados (LGPD), Lei nº 13.709/2018.

A importância da proteção de dados e da privacidade nunca foi tão crítica como nos dias atuais, em um mundo cada vez mais digital. A LGPD estabeleceu um marco legal para a proteção de dados pessoais no Brasil, exigindo que as organizações implementem medidas técnicas e organizacionais para garantir a segurança e a transparência no tratamento de informações sensíveis. Este contexto é enfatizado por incidentes de vazamento de dados de alta magnitude, como o ocorrido com o Ministério da Saúde em 2020, que expôs informações de 243 milhões de brasileiros devido a uma falha de segurança, e o incidente relatado pelo Banco Central em janeiro de 2021, onde dados pessoais vinculados a 160.147 chaves PIX foram potencialmente expostos pela empresa Acesso Soluções de Pagamento. Esses incidentes ressaltam a vulnerabilidade dos sistemas de informação e sublinham a importância de adotar práticas robustas de segurança cibernética.

O TCE/RN desempenha um papel fundamental na governança e na promoção da transparência nas ações governamentais, sendo a segurança de seus sistemas essencial para garantir a integridade das informações financeiras e operacionais. Diante desses exemplos alarmantes, esta pesquisa é motivada também pela necessidade de criar uma estrutura capaz de identificar e propor melhorias para minimizar as vulnerabilidades existentes. Assim, busca-se proporcionar ao TCE/RN uma maior confiança na proteção dos dados e, consequentemente, na prestação de serviços à sociedade, reforçando a segurança da informação como um pilar fundamental na era digital.

1.2 Objetivos

A seguir são apresentados o objetivo geral e os objetivos específicos que norteiam o desenvolvimento deste trabalho.

1.2.1 Objetivos gerais

Este trabalho tem como objetivo geral a implementação e a utilização da ferramenta *Zed Attack Proxy* (ZAP) para conduzir uma análise detalhada das vulnerabilidades de segurança cibernética nos sistemas INTRATCE, SisEscola e TCE Admin. Esses sistemas, desenvolvidos ou mantidos pela turma 04 da residência do Tribunal de Contas do Estado do Rio Grande do Norte (TCE/RN), serão explorados como estudos de caso detalhados no Capítulo 3.

1.2.2 Objetivos específicos

A fim de alcançar os objetivos gerais descritos acima, foram abordados os seguintes objetivos específicos:

- Realizar estudo do referencial teórico relacionado à *cyber segurança*;

- Buscar encontrar e catalogar vulnerabilidades nos sites do TCE/RN analisados pela ferramenta ZAP;
- Fazer uma análise das vulnerabilidades encontradas;
- Propor alternativas para dar seguimento no desenvolvimento deste trabalho.

1.3 Metodologia

Visando contribuir para o fomento da segurança da informação do Tribunal de Contas, conforme descrito acima, foi realizada uma busca por protocolos, ferramentas e estruturas atuais utilizadas pela comunidade, que possam ser integradas ao sistema de desenvolvimento e integração do TCE/RN. Nesta pesquisa, encontrou-se a organização sem fins lucrativos *Open Web Application Security Project* (OWASP), focada na melhoria da segurança de software.

A OWASP produz documentos, protocolos e fóruns de discussão com a comunidade sobre vulnerabilidades, além de disponibilizar ferramentas para a descoberta de vulnerabilidades de software. Dentre as ferramentas disponíveis, a *Zed Attack Proxy* (ZAP) destacou-se como uma possível candidata para atender aos objetivos do trabalho. Assim, foi realizada uma revisão bibliográfica baseada nos princípios de inclusão e exclusão para um melhor entendimento sobre a OWASP e a ZAP.

Após compreender a organização e a utilização da ZAP, iniciou-se um estudo mais aprofundado sobre como implementar e automatizar essa ferramenta nos sistemas produzidos pelo tribunal. Esse processo envolveu a análise detalhada das funcionalidades da ZAP, assim como o desenvolvimento de *scripts* e procedimentos para integrá-la eficientemente no ciclo de vida do desenvolvimento de software do TCE/RN. O objetivo era garantir que todas as aplicações desenvolvidas fossem submetidas a testes de segurança, utilizando as capacidades da ZAP para identificar e mitigar potenciais vulnerabilidades. Essa integração visou não apenas aprimorar a segurança das aplicações, mas também promover uma cultura de conscientização e proatividade em segurança de software entre os desenvolvedores e gestores do tribunal.

1.4 Estrutura do trabalho

Este trabalho está dividido em seis capítulos, iniciando com esta introdução, que apresenta o resumo e expõe a metodologia adotada. O segundo capítulo aborda o referencial teórico, delineando o contexto atual da área de estudo envolvida neste Trabalho de Conclusão de Curso e destacando sua relevância. O terceiro capítulo contextualiza a proposta deste trabalho e os desafios enfrentados durante sua elaboração. O quarto capítulo valida a proposta apresentada anteriormente, exibindo os casos de uso e os resultados

obtidos. O quinto capítulo discute os resultados encontrados e propõe maneiras de mitigar os problemas identificados. Por fim, o sexto capítulo conclui o estudo, enfatizando os impactos e contribuições da utilização da ferramenta ZAP para o TCE/RN.

2 Referencial Teórico

Neste capítulo é apresentado o referencial teórico relativo à segurança da informação e segurança cibernética, enfatizando sua evolução e relevância crescente em um ambiente digital global. A segurança da informação é contextualizada como um pilar crucial na proteção de dados em uma era digital, enquanto a segurança cibernética é explorada através de suas estratégias e tecnologias inovadoras para salvaguardar infraestruturas digitais. O capítulo aprofunda ainda nos papíes dos frameworks NIST e OWASP, fundamentais no estabelecimento de padrões e práticas para a segurança cibernética e a segurança de aplicações web, respectivamente. A análise detalhada dessas estruturas é essencial para compreender as diretrizes de segurança contemporâneas e suas aplicações práticas, fornecendo uma base sólida para o estudo empírico proposto.

2.1 Segurança da Informação

Na era digital em que vivemos, a segurança da informação é uma coluna vertebral que sustenta a confiança e a estabilidade em diversos setores. Desde transações financeiras até dados sensíveis, a proteção desses ativos tornou vital para o funcionamento adequado da sociedade. A segurança da informação é um campo multifacetado que abrange uma gama de práticas e medidas destinadas a salvaguardar dados contra ameaças cada vez mais sofisticadas.

De acordo com a ISO/IEC 17799:2005 (recentemente atualizada para 27002:2022), a informação é um ativo crítico para o negócio de uma organização e, portanto, precisa ser adequadamente protegida, pois a conectividade e o advento da era digital fazem com que os dados estejam cada vez mais expostos e partilhados, sejam eles em diversas formas: papel impresso ou escrito, correio, e-mail, filme, gravação, ou mesmo por voz, simples conversa.

(LOURENÇO; DUARTE, 2020) oferecem uma definição abrangente ao descrever a Segurança da Informação como um conjunto de práticas voltadas para a proteção de informações, sistemas, recursos e demais ativos contra desastres, erros intencionais ou não, e manipulações não autorizadas. O objetivo primordial é reduzir a probabilidade e o impacto de incidentes da segurança, reconhecendo a necessidade de prevenção e resposta eficaz diante das ameaças.

A essência do conceito de segurança da informação reside na salvaguarda de um conjunto de dados, visando preservar o valor intrínseco que essas informações possuem para uma pessoa ou organização. Essa premissa é fundamentada nos princípios da Tríade da Confidencialidade, Integridade e Disponibilidade, conforme estabelecido por (HARRIS; MIAMI, 2019) e introduzida com mais detalhes abaixo:

- **Confidencialidade**: Um dos elementos cruciais da segurança da informação. uma vez que, se concentra na preservação da informação, garantindo que apenas as pessoas autorizadas e interessadas tenham acesso a determinados dados. De acordo com (BARCELOS et al., 2021) a Confidencialidade busca estabelecer mecanismos e controles que restrinjam o acesso não autorizado, protegendo informações sensíveis ou sigilosas.Visando criar uma camada de segurança em torno dos dados, assegurando que apenas aqueles que têm uma necessidade legítima de conhecimento possam obtê-lo.

- **Integridade**: A integridade é outro pilar dessa tríade. Nela, garantir que os dados permaneçam precisos, íntegros e livres de alterações maliciosas é essencial. Mecanismos como assinaturas digitais e controle de versões são empregados para rastrear e verificar mudanças, assegurando a integridade das informações. Portanto, a Integridade é um componente fundamental na segurança da informação, envolvendo a proteção das informações contra adulterações ou modificações não autorizadas. De acordo com (BARCELOS et al., 2021), esse princípio busca garantir que os dados permaneçam íntegros, ou seja, que mantenham sua precisão e consistência ao longo do tempo.

- **Disponibilidade**: Em um mundo onde a informação é um ativo instantâneo, é imperativo que os sistemas e dados estejam sempre acessíveis quando necessário. Estratégias de redundância, backups regulares e planos de recuperação de desastres desempenham um papel crucial nesse aspecto.(BARCELOS et al., 2021).

Conforme definida pela norma (BRASIL, 2005), a segurança vai além das propriedades clássicas de integridade, disponibilidade e confidencialidade, pois incorpora conceitos cruciais, tais como autenticidade, responsabilidade, não repúdio e confiabilidade. Esses elementos são fundamentais para assegurar a proteção dos ativos de informação contra diversas ameaças do ambiente.

O objetivo primordial da segurança da informação é proteger os ativos de informação contra ameaças, minimizando os riscos para os negócios. Para alcançar eficácia nesse sentido, é necessário criar políticas, procedimentos, processos, estruturas organizacionais e funções computacionais, tanto lógicas quanto físicas. A norma ABNT (BRASIL, 2005) ressalta a importância de estabelecer, implementar, monitorar, analisar criticamente e melhorar continuamente os controles, visando atender aos objetivos de negócio e segurança.

2.2 Segurança Cibernética

A ascensão tecnológica, marcada pela adoção generalizada da cibernética e da *internet*, transformou radicalmente a vida no século XXI. Enquanto revoluções comunica-

cionais anteriores, como o telégrafo e o telefone, datam do século XIX, a *internet* se destaca pela sua instantaneidade, sendo comparada a grandes transformações históricas, como a transição do nomadismo para a sedentarização e a revolução industrial (MANDARINO, 2010).

O surgimento da *Internet* remonta à década de 70, com a criação da ARPANET (*Advanced Research Projects Agency Net*), inicialmente concebida para conectar computadores isolados em diversas universidades nos Estados Unidos. Rapidamente, o interesse militar impulsionou seu desenvolvimento, culminando na privatização da *internet* na década de 1990, marcada pela ascensão da *World Wide Web* e ferramentas como o *E-mail*.

O rápido crescimento do número de usuários da *internet* não se limitou a indivíduos, uma vez que Estados nacionais também perceberam suas vantagens e buscaram modernizar suas estruturas administrativas e burocráticas, incorporando processos e armazenamento digitais. Contudo, essa revolução informacional trouxe consigo não apenas benefícios, mas também desafios, destacando-se a necessidade de defesa cibernética como um novo domínio estratégico, o "quinto domínio da defesa" (AGOSTINI, 2014).

Nesse cenário, a compreensão dos conceitos fundamentais da área cibernética torna-se crucial para a formulação de estratégias de defesa e segurança, visando proteger infraestruturas críticas e promover a soberania digital no ambiente digital do século XXI. Originada em 1961 com Wiener, refere-se ao controle e comunicação em humanos e máquinas, também associada à doutrina militar de defesa cibernética do Brasil (Ministério da Defesa, 2014). Por outro lado, a *Internet* é definida como uma rede global interligando inúmeras redes (BRASIL, 2019a). Embora cibernética e *internet* não sejam sinônimos, a *internet* é uma seção relevante do espaço cibernético, o qual abrange estruturas além da *internet*, como satélites e sistemas industrializados (VENTRE, 2011).

Ademais, é necessário compreender o conceito de espaço cibernético, definido como um sistema contingente hierárquico, engloba bases físicas, blocos lógicos, informações e atores. Sua territorialização, embora não siga padrões geográficos tradicionais, envolve a apropriação de pontos de interesse pelos Estados, configurando-se como um "quinto domínio" essencial para a defesa nacional. Além disso, a dificuldade de atribuição, o chamado "problema da atribuição", destaca-se no ciberespaço, tornando complexa a identificação dos autores de ataques cibernéticos (RID, 2013).

Ainda no âmbito conceitual, faz-se importante destacar que, a distinção entre segurança cibernética e defesa cibernética é um tema complexo, sujeito a interpretações variadas e falta de consenso. A diferenciação desses conceitos começou a se tornar mais clara no *Human Development Report* de 1994, quando a visão de segurança internacional passou a ser questionada, deixando de estar estritamente ligada aos Estados (UNIVERSITY, 1994).

Ao longo do tempo, a ideia de segurança cibernética ampliou-se para incluir não apenas questões relacionadas aos Estados, mas também à sociedade civil, setor privado,

segurança pública e atividades ilícitas (FILHO, 2014). Por outro lado, a defesa cibernética permanece vinculada à proteção dos interesses e soberania do Estado-nação, muitas vezes associada à perspectiva de guerra (OLIVEIRA, 2020).

No entanto, a diferenciação entre segurança e defesa cibernética nem sempre é clara, especialmente em situações em que a sabotagem e a espionagem cibernéticas podem afetar tanto o domínio privado quanto o público. Ambos os conceitos podem se sobrepor, tornando a interpretação e a aplicação dessas ideias desafiadoras. Autores como (GALOYAN, 2019) argumentam que a defesa cibernética está intrinsecamente ligada à guerra, enquanto (FILHO, 2014) observa que a diferenciação desses termos não é uniformemente adotada em todos os países.

Com a adesão da *Internet* ao longo dos últimos vinte anos, intensificou a comunicação global, mas tornou o mundo dependente da tecnologia. A rede possui cerca de 3 bilhões de usuários no mundo (LI; LIU, 2021) e é uma grande responsável por rodar a economia, independe do ramo econômico (cultural, social, artístico etc), todos aderiram a rede, sejam instituições governamentais ou não. Assim, sistemas vitais para o funcionamento dessas instituições adentraram na rede. Entretanto, sua adesão massiva também expôs vulnerabilidades nos sistemas web, tornando-se alvo de ameaças cibernéticas que variam desde sabotagens e espionagem até ataques não autorizados, representando desafios significativos (RID, 2013).

2.3 NIST

A *National Institute of Standards and Technology* (NIST) emerge como uma figura proeminente entre as principais instituições tecnológicas, desempenhando um papel crucial na definição de padrões de medição e diretrizes de segurança cibernética. Com uma missão dedicada à promoção da inovação e da competitividade econômica entre as empresas, o NIST contribui significativamente para a evolução e aprimoramento das práticas tecnológicas em escala global.(NIST, 1901)

Os padrões criados pelo NIST gozam de uma aceitação internacional notável, consolidando-se como referências confiáveis em diversos setores tecnológicos. A reputação destacada da instituição é resultado direto de suas contribuições abrangentes e contínuas nas áreas da tecnologia. Esses padrões não apenas estabelecem critérios rigorosos de qualidade e segurança, mas também refletem o compromisso do NIST com a excelência e a inovação.(NIST, 1901)

Ao adotar os padrões do NIST, a pesquisa se beneficia não apenas da expertise da instituição, mas também contribui para a harmonização e a interoperabilidade das práticas de segurança cibernética em um contexto global. A escolha desses padrões não apenas reforça a credibilidade da pesquisa, mas também demonstra um compromisso com a adoção de diretrizes robustas que atendam aos mais elevados padrões internacionais de

qualidade e segurança.(NIST, 1901)

O NIST desempenha um papel central no estabelecimento de diretrizes e padrões essenciais para a segurança cibernética, e suas Special Publications (SP) são referências fundamentais nesse cenário. Entre elas, a NIST SP 800-53 assume um papel crucial ao apresentar o "Controle de Segurança e Famílias de Controles". Este guia fornece uma estrutura minuciosa e abrangente para a seleção e implementação de controles de segurança em sistemas de informação. Ao abranger temas que vão desde autenticação e controle de acesso até criptografia e gestão de incidentes de segurança, o documento estabelece uma base sólida para a construção de ambientes seguros e resilientes.(NIST, 1901)

Outra publicação significativa é a NIST SP 800-61, o "Guia para Detecção de Incidentes e Resposta". Este documento fornece diretrizes detalhadas para a detecção e resposta a incidentes de segurança cibernética. Ao abordar as melhores práticas para identificar e mitigar ameaças, bem como ações eficazes de resposta a incidentes, a SP 800-61 é um recurso valioso para organizações que buscam fortalecer sua postura de segurança(NIST-800-61, 2023).

Para contratantes do governo dos EUA, a NIST SP 800-171 desempenha um papel essencial, estabelecendo diretrizes de segurança para proteger informações não classificadas, mas sensíveis. Definindo controles de segurança que as organizações devem implementar para proteger informações compartilhadas como governo, esta publicação é vital para garantir a segurança da informação em setores governamentais.(NIST-800-171, 2022)

A NIST SP 800-30, ou "Guia de Gerenciamento de Riscos", oferece uma estrutura abrangente para avaliar e gerenciar os riscos de segurança da informação em organizações. Isso inclui a identificação de ativos, avaliação de ameaças, análise de vulnerabilidades e a determinação de riscos aceitáveis. Essa abordagem sistemática é crucial para uma gestão eficaz da segurança da informação.(NIST-800-30, 2022)

Por fim, a NIST SP 800-37, o "Guia de Gestão de Riscos e Segurança da Informação", concentra-se em abordagens para gerenciar riscos de segurança da informação de forma mais ampla, integrando a segurança no ciclo de vida de sistemas e processos de negócios. Ao adotar uma perspectiva holística, este documento promove práticas que visam à construção de uma cultura organizacional voltada para a segurança. Em conjunto, essas publicações do NIST representam ferramentas essenciais para organizações que buscam fortalecer suas defesas cibernéticas e gerenciar riscos de forma eficaz. (NIST-800-37, 2022)

Além do renome da instituição que produz esses documentos, cada norma estabelecida pelo NIST é revisada periodicamente para refletir as mudanças dinâmicas no cenário de ameaças cibernéticas. Essa prática é essencial para garantir que os padrões e diretrizes permaneçam relevantes e eficazes ao longo do tempo. As revisões periódicas visam incorporar as lições aprendidas com incidentes de segurança, avanços tecnológicos e as evoluções nas táticas empregadas por cibercriminosos. (NIST, 1901)

O processo de revisão assegura a adaptabilidade das normas a cenários emergentes

e também promove a inovação e a atualização contínua das melhores práticas de segurança cibernética. À medida que as instituições participantes encontram novas ameaças ou desenvolvem abordagens mais eficazes para a proteção de sistemas de informação, essas experiências são consideradas no aprimoramento das normas existentes ou na criação de novas diretrizes. (NIST, 1901)

Dessa forma, os padrões estabelecidos pelo NIST representam um conjunto consolidado de diretrizes no momento de sua criação e evoluem em resposta às mudanças na paisagem de segurança cibernética. Essa abordagem dinâmica e proativa garante que as organizações que seguem esses padrões possam manter uma postura de segurança robusta, enfrentando os desafios emergentes de maneira eficiente.

2.4 OWASP

Entre as principais prioridades no campo da Segurança da Informação, a salvaguarda dos sistemas web contra ataques cibernéticos assume um papel crucial. Essa tarefa é respaldada por recomendações estabelecidas pelo *Open Web Application Security Project* (OWASP), que se posiciona como uma referência essencial na garantia da segurança de sistemas online. OWASP surgiu em 2001 como uma organização global sem fins lucrativos dedicada à melhoria da segurança do software. Sua missão é reunir profissionais de segurança, desenvolvedores e organizações comprometidas em aprimorar as práticas de segurança, especialmente no contexto do desenvolvimento de aplicativos *web* e *software* em geral(OWASP, 2001)

A OWASP tem vários objetivos fundamentais e realiza diversas atividades para atingi-los. Em primeiro lugar, destaca-se o papel crucial desempenhado na educação e conscientização sobre segurança cibernética. A organização fornece uma ampla gama de recursos e materiais educativos, além de organizar eventos que servem como plataformas para disseminar boas práticas de segurança. Essa abordagem visa alcançar desenvolvedores, profissionais de segurança e tomadores de decisão, promovendo uma compreensão abrangente das ameaças e medidas preventivas.(OWASP, 2001)

Outro aspecto distintivo da OWASP é a manutenção de projetos de segurança de código aberto. Esses projetos englobam ferramentas, guias e *frameworks* destinados a abordar diferentes facetas da segurança de software. Ao oferecer soluções práticas e acessíveis, a OWASP capacita os desenvolvedores a criar sistemas mais seguros desde a fase inicial de desenvolvimento. (OWASP, 2001)

As recomendações essenciais fornecidas pela OWASP visam prevenir vulnerabilidades comuns encontradas em aplicativos da web. Entre as medidas prioritárias estão a proteção contra ataques de injeção de SQL, validação de entrada de dados, validação de sessão, e a implementação de autenticação e autorização adequadas. Por exemplo, a validação rigorosa da entrada de dados desempenha um papel crítico na prevenção de

ataques de injeção de SQL, os quais poderiam resultar na divulgação de dados sensíveis e na instabilidade do sistema. (OWASP, 2001)

A OWASP enfrenta efetivamente ameaças como *script* entre sites e falsificação de solicitações entre sites por meio de suas diretrizes. Os desenvolvedores podem substancialmente aumentar a segurança de seus aplicativos web ao aderir a essas diretrizes, incluindo práticas como o uso da Política de Segurança de Conteúdo(CSP) para evitar ataques XSS, representando uma excelente precaução.

Com a evolução das tecnologias, aplicativos de página única (SPAs) e *APIs RESTful* introduziram novos desafios aos protocolos de segurança da web. A OWASP aborda esses desafios fornecendo orientações atualizadas para lidar compre ocupações específicas desses ambientes. Um dos aspectos mais cruciais dessa orientação é a ênfase na autenticação fortalecida e na autorização baseada em funções, especialmente importante em cenários em que as aplicações web dependem fortemente de APIs para funcionar.

A organização é reconhecida por suas valiosas contribuições documentais, como o "OWASP Top 10", que destaca as principais ameaças à segurança de aplicativos web. Esses recursos são considerados guias essenciais pela comunidade de desenvolvimento, auxiliando na identificação e mitigação de vulnerabilidades comuns. A classificação dessas ameaças, numeradas em um ranking, visa fornecer uma perspectiva clara sobre sua criticidade, sendo a primeira a mais impactante e a última a apresentar menor risco. (OWASP, 2001)

Uma das categorias críticas é a Controle de acesso quebrado, que se manifesta quando um invasor obtém acesso não autorizado a uma página, manipulando parâmetros da URL. Isso inclui escalonamento horizontal e vertical de privilégios, com exemplos de falhas como falsificação de solicitação entre sites , Referência de objeto direto inseguro, Redirecionamento aberto e Travessia de caminho (FRANZESE, 2023).

A categoria injeção envolve a manipulação de parâmetros de entrada do usuário, resultando em ataques como injeção SQL/NoSQL, *Scripting* entre sites (XSS), injeção HTML e Injeção de comando. Os impactos abrangem roubo de dados, perda de dados e comprometimento do sistema.Falhas de Design Inseguro ocorrem quando desenvolvedores negligenciam padrões seguros, modelagem de ameaças ou arquiteturas de referência. *Common Weakness Enumeration* (CWEs) associados incluem atribuição de privilégio incorreta, gerenciamento inadequado de privilégios, violação do limite de confiança e credenciais insuficientemente protegidas (FRANZESE, 2023).

A categoria Configuração incorreta de segurança abrange vulnerabilidades decorrentes de erros de configuração, como cabeçalhos HTTP mal configurados e contas com login e senha padrão. Os impactos variam de acesso não autorizado a comprometimento total do sistema, com exemplos de CWEs como armazenamento de informações confidenciais em texto plano e armazenamento de senha em arquivos de configuração (FRANZESE, 2023).

Uma outra categoria é a Falhas de identificação e autenticação ocorrem quando

funções relacionadas à identidade, autenticação ou gerenciamento de sessão não são implementadas corretamente. CWEs associados incluem autenticação imprópria, requisitos de senha fracos e expiração de sessão insuficiente.Falhas em Integridade de software e dados resultam da dificuldade na atualização de arquiteturas complexas e no uso de *plug-ins* ou bibliotecas não confiáveis. CWEs associados incluem verificação insuficiente de autenticidade dedados, download de código sem verificação de integridade e falta de suporte para verificação de integridade (FRANZESE, 2023).

Por fim, temos a categoria de falhas insuficientes de registro e monitoramento, que ocorrem quando práticas inadequadas são seguidas para registrar e monitorar *logs* do sistema, com CWEs associados como neutralização inadequada de saída para *logs*, omissão de informações relevantes para a segurança e registro insuficiente.(FRANZESE, 2023).

Visando verificar essas classes de vulnerabilidades descritas acima, foi criada a ferramenta *Zed Attack Proxy* (ZAP), desenvolvida pela OWASP, destacando-se como uma solução útil para revisar a segurança de diversos aplicativos web. Sua flexibilidade e adaptabilidade permitem que profissionais de segurança personalizem e automatizem testes para identificar problemas como XSS, injeção de SQL, CSRF, entre outros. A integração perfeita do ZAP aos processos de desenvolvimento de software, graças aos seus relatórios detalhados e compatibilidade com pipelines de desenvolvimento, possibilita a identificação e correção de ameaças potenciais desde as fases iniciais do desenvolvimento, economizando tempo e recursos significativos.

A OWASP, portanto, desempenha um papel vital na promoção de práticas de segurança em toda a comunidade de desenvolvimento de software. Ao desafiar e capacitar organizações, ela contribui significativamente para a construção de sistemas mais robustos e resilientes contra as crescentes ameaças cibernéticas, que é o estudo principal deste trabalho, que será detalhado nos próximos capítulos[1].

[1] owasp.org/www-project-proactive-controls

3 Proposta

Este capítulo explora a configuração e o uso da *Zed Attack Proxy* (ZAP) para analisar sistemas selecionados com base na importância para a turma 04 do TCE/RN, destacando a adoção de uma abordagem sistemática para reforçar a segurança da informação. Também descreve o processo de automação dessa análise por meio da ferramenta Cypress. Além disso, o capítulo ressalta os desafios enfrentados durante o desenvolvimento deste trabalho.

Com o intuito de entregar uma contribuição significativa para o TCE/RN no que se refere a segurança da informação, foi escolhida ferramenta ZAP devido a sua grande aceitação da comunidade de segurança de informação e à sua capacidade de detectar uma ampla gama de vulnerabilidades. Além disso, a OWASP, organização sem fins lucrativos que produziu a ferramenta está ganhando cada vez mais relevância no ramo de segurança da informação, pois sempre atualiza as vulnerabilidades mais exploradas em ataques em organizações e como mitiga-las.

Vale salientar que entre as várias ferramentas produzidas e mantidas pela OWASP, a ZAP é a única projetada para encontrar vulnerabilidades em aplicações web, além disso, a mesma é amplamente utilizadas por profissões da área de segurança de informações e até mesmo por desenvolvedores que buscam validar a segurança de suas aplicações.

A *Zed Attack Proxy* possui vários modos de configuração: modo de segurança, modo protegido, modo *ATTACK* e modo padrão. A política de varredura utilizada foi a padrão, que possui os seguintes testes configurados para ataque:

1. Script entre sites (baseado em DOM)
2. Vazamento de informações
3. Divulgação de Código-Fonte
4. Execução Remota de Código
5. Hidden File Finder
6. Navegação no Diretório
7. Spring Actuator Information Leak
8. User Agent Fuzzer
9. Vulnerabilidade OpenSSL Heartbleed
10. GET for POST
11. Log4Shell
12. Oracle Padding Genérico
13. Redirecionamento Externo
14. Spoofing de Ação SOAP
15. Adulteração de parâmetros

16. Cross Site Scripting
17. Erro de Formato de String
18. Estouro de Buffer
19. Inclusão Lado do Servidor
20. Injeção CRLF
21. Injeção de Código no Lado do Servidor
22. Injeção de comando
23. Injeção SQL
24. Injeção XPath
25. Injeção XSLT
26. Metadados de nuvem potencialmente expostos
27. Server Side Template Injection
28. Spring4Shell
29. XML External Entity Attack
30. Inclusão de Arquivo Remoto
31. Travessia/Passagem de Caminho

Em geral, são realizados cada um dos testes listados acima a medida em que o *proxy* da ZAP recebe as URLs.

A escolha dos sites para varredura da ZAP foi baseada na relevância dos mesmos para a turma 04 da residência do TCE/RN (2022-2024). Foram selecionados três sistemas, nos quais cada um configurou um caso de uso, onde as varreduras realizadas dos mesmos são apresentadas no Capítulo 4. Diante disto, foram escolhidos os seguintes sistemas que são projetos base dos programas de residência: TCE Admin, INTRATCE e Escola de Contas.

O TCE Admin é um site que é constantemente utilizado pelos residentes. Ele é responsável por atribuir permissões para a criação de novos *sites*, rotas e usuários. O INTRATCE e a Escola de Contas são os projetos iniciais de desenvolvimento da turma 04 da residência, onde estão em processo de finalização no momento da escrita deste trabalho.

Inicialmente, as varreduras com a *Zed Attack Proxy* foram feitas de forma manual. Isso significa que as URLs dos sites escolhidos eram inseridos na ferramenta manualmente, assim como ilustrado na Figura 1.

Após a inserção da URL para varredura, é necessário escolher o navegador de acesso a URL. A ferramenta ZAP permite que o ataque seja feito em qualquer navegador da preferência do usuário. Caso o usuário deseje utilizar outros navegadores não listado pela ZAP, será necessário inseri-lo na raiz do projeto. Após a inserção da URL, a inicialização da varredura acontece após clicar em “Abrir o Navegador”, onde no exemplo da figura acima irá abrir a URL “http://intratcefeature.tce.govrn” no navegador *Firefox*, assim como ilustrado na Figura 2:

A ferramenta ZAP funciona da seguinte forma: é fornecido para a ferramenta uma

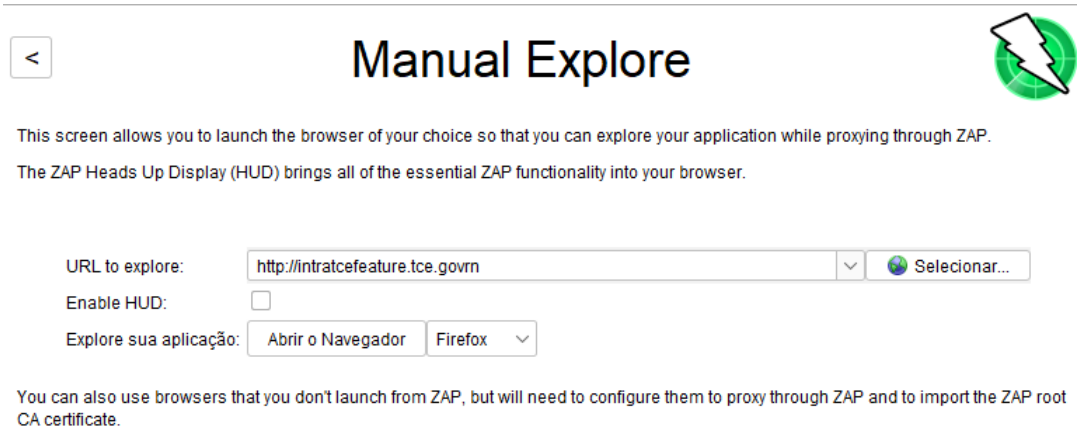
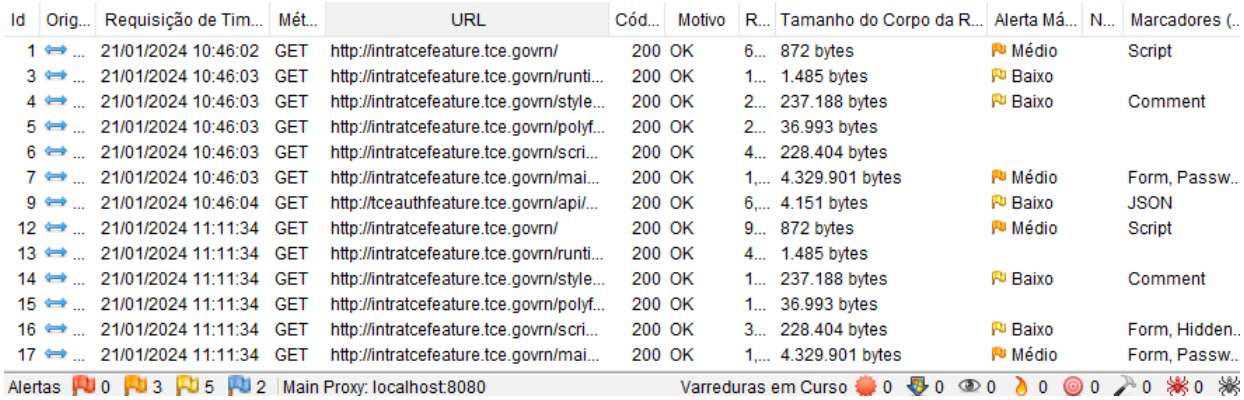


Figura 1 – Painel do *Zed Attack Proxy* para ataque manual.

Figura 2 – Navegador aberto pelo ZAP para ataque.

URL de algum site. Então, a ZAP irá bombardear essa URL com requisições HTTP e verificar o retorno de cada requisição, analisando o conteúdo da resposta e se há uma possível vulnerabilidade na resposta da requisição fornecida. A Figura 3 ilustra um exemplo de algumas requisições que a ferramenta fez com a URL base “http://sisescolafeature.tce.govrn”.



Id	Orig...	Requisição de Tim...	Mét...	URL	Cód...	Motivo	R...	Tamanho do Corpo da R...	Alerta Má...	N...	Marcadores (..
1	...	21/01/2024 10:46:02	GET	http://intratcefeature.tce.govrn/	200	OK	6...	872 bytes	Médio		Script
3	...	21/01/2024 10:46:03	GET	http://intratcefeature.tce.govrn/runti...	200	OK	1...	1.485 bytes	Baixo		
4	...	21/01/2024 10:46:03	GET	http://intratcefeature.tce.govrn/style...	200	OK	2...	237.188 bytes	Baixo		Comment
5	...	21/01/2024 10:46:03	GET	http://intratcefeature.tce.govrn/polyf...	200	OK	2...	36.993 bytes			
6	...	21/01/2024 10:46:03	GET	http://intratcefeature.tce.govrn/scri...	200	OK	4...	228.404 bytes			
7	...	21/01/2024 10:46:03	GET	http://intratcefeature.tce.govrn/mai...	200	OK	1,...	4.329.901 bytes	Médio		Form, Passw..
9	...	21/01/2024 10:46:04	GET	http://tceauthfeature.tce.govrn/api/...	200	OK	6,...	4.151 bytes	Baixo		JSON
12	...	21/01/2024 11:11:34	GET	http://intratcefeature.tce.govrn/	200	OK	9...	872 bytes	Médio		Script
13	...	21/01/2024 11:11:34	GET	http://intratcefeature.tce.govrn/runti...	200	OK	4...	1.485 bytes			
14	...	21/01/2024 11:11:34	GET	http://intratcefeature.tce.govrn/style...	200	OK	1...	237.188 bytes	Baixo		Comment
15	...	21/01/2024 11:11:34	GET	http://intratcefeature.tce.govrn/polyf...	200	OK	1...	36.993 bytes			
16	...	21/01/2024 11:11:34	GET	http://intratcefeature.tce.govrn/scri...	200	OK	3...	228.404 bytes	Baixo		Form, Hidden..
17	...	21/01/2024 11:11:34	GET	http://intratcefeature.tce.govrn/mai...	200	OK	1,...	4.329.901 bytes	Médio		Form, Passw..

Figura 3 – listagem de requisições e alertas.

A Figura 2 também mostra que a URL da página está em vermelho, o que demostra algum tipo de risco. Essa questão será melhor abordada no capítulo 3.1, que fala sobre os desafios encontrados ao longo do desenvolvimento deste trabalho de conclusão de curso.

Após a injeção da URL na ferramenta e abrir o navegador, o ataque da ferramenta é iniciado. A ZAP intercepta e registra todas as solicitações de resposta HTTP/HTTPS entre seu navegador e o servidor web destino. A ferramenta, então, registra todas as requisições e as respostas do servidor, incluindo cabeçalhos, parâmetros, *cookies*, URL´s e métodos HTTP. Caso a ZAP identifique uma possível vulnerabilidade em sua varredura, a mesma emite um alerta na interface da sua aplicação. A Figura 3 ilustra as requisições realizadas e os alertas encontrados.

Posteriormente a toda a varredura do *proxy* do todas as rotas da página, a ferramenta gera um relatório no qual expõe as vulnerabilidades encontradas no ataque e quais foram as requisições que encontraram uma determinada vulnerabilidade. Segue abaixo uma demonstração do relatório expondo uma vulnerabilidade e requisição que expôs a mesma através da Figura 4.

Figura 4 – Requisição e vulnerabilidade encontrada

A execução de forma manual foi efetuada a fim de verificar o funcionamento da ferramenta ZAP. Porém, a execução manual pode se tornar inviável caso for necessário executar esse mesmo procedimento em todos os mais de 80 sistemas que o TCE/RN sustenta atualmente. Portanto, buscou-se uma alternativa para execução manual e encontrou-se a

ferramenta Cypress[1], que mostrou-se uma alternativa viável para executar a automação, tendo em vista que a mesma faz requisições a API e também simula a ação do usuário na interface. Atualmente, não existe nenhuma compatibilidade nativa entre o Cypress e a ZAP, porém, foi desenvolvido uma solução para que haja uma união entre as ferramentas.

A escolha da ferramenta Cypress também se mostra adequada para automação na ZAP, dada a sua recente adoção pelo TCE/RN. A turma 04 da residência iniciou os estudos e aplicou a ferramenta no INTRATCE para realizar testes de interface e testes de API. Na elaboração deste trabalho, o tribunal planeja integrar o Cypress em todos os seus sistemas, devido aos resultados positivos obtidos em sua aplicação pelos residentes.

A ideia por trás da integração do ZAP com o Cypress consiste no seguinte: o proxy da ZAP é integrado às requisições do Cypress e a medida em que são feitas as requisições à API e a interface do usuário, a ZAP executa os testes de vulnerabilidades nas URL´s chamadas. Para realização dessa integração na prática, foi injetado a porta onde o *proxy* da ZAP estava rodando no comando de inicialização do cypress, dessa forma, o cypress já iniciará suas ações com a ZAP sendo alimentada pelas requisições feitas pelos testes do mesmo. A figura Figura 5 ilustra como foi realizado esse procedimento.

```
"cy:open": "cross-env HTTP_PROXY=http://localhost:8080 NO_PROXY=\"<-loopback>\"
```

Figura 5 – Inserção do proxy da ZAP na inicialização do cypress

Esse procedimento foi validado colocando a API do ZAP em container docker rodando na porta 8080 e executando o projeto do Cypress que compõe a pipeline do TCE/RN com o proxy do ZAP em suas requisições e verificando a rede do seu *container*. A Figura 6 ilustra a rede da ZAP a medida em que recebe as requisições do Cypress.

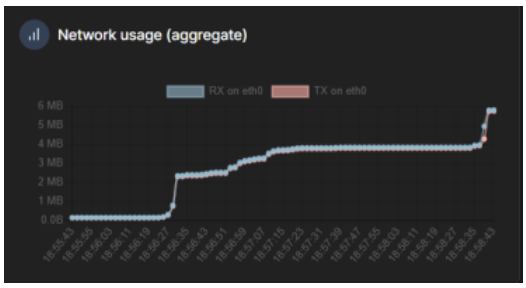


Figura 6 – Rede da ZAP recebendo as requisições do Cypress

Após execução dos testes do Cypress, é realizado uma requisição para API da ZAP para a gerar o relatório das vulnerabilidades mapeadas. Após isso, é feita análise do relatório gerado para identificar e classificar as vulnerabilidades com base em sua severidade e tipo. Cada vulnerabilidade encontrada foi catalogada e descrita em detalhes no próximo capítulo.

[1] https://www.cypress.io/

3.1 Desafios encontrados

Ao longo do desenvolvimento da proposta do presente projeto, foram entrevistados alguns terceirizados e servidores a fim de chegar em uma opinião comum para propor um plano de desenvolvimento de forma a mitigar as vulnerabilidades encontradas pela ferramenta nos projetos desenvolvidos. Entretanto, não há uma padronização de planejamento e execução entre as equipes de desenvolvimento do tribunal e não houve um senso comum sobre o assunto.

Ainda assim, os desenvolvedores consultados viram a importância de propor melhorias no plano de desenvolvimento das ferramentas, ressaltando a falta de organização nas mudanças feitas no sistema, visto que não há uma padronização para os desenvolvedores e que não há uma documentação mais elaborada sobre as mudanças, principalmente dos projetos que recebem sustentação. Sustentação é o termo utilizado no TCE/RN para o ato de manter, resolver bugs e aprimorar os sistemas em produção, em uma grande maioria os sistemas considerados legados.

Outro problema encontrado ao longo da produção do trabalho foi o fato de que o antivírus do TCE/RN considera a ferramenta ZAP como uma ameaça e bloqueia o seu funcionamento. Isso ocorreu pois a ZAP atua como um *man-in-the-middle* entre o navegador do usuário o site, interceptando e modificando o tráfego para testar a segurança da aplicação. Esse tipo de comportamento pode ser semelhante ao de *malwares* que tentam capturar ou alterar dados do usuário de forma maliciosa, por este motivo o antivírus do tribunal bloqueou a ZAP.Mas este problema foi resolvido rapidamente pelo departamento de infraestrutura.

Resolvido o problema do bloqueio da ferramenta, levantou-se o questionamento sobre a validade do estudo a respeito das vulnerabilidades serem falsos positivos, devido as varreduras utilizarem os ambientes de desenvolvimento para análise, pois o ambiente de produção possui *proxys* que mitigariam algumas vulnerabilidades. Entretanto, o *proxy* de produção do TCE/RN foi anexado aos ambientes de desenvolvimento e não houveram mudanças nas ameaças encontradas.

4 Validação

Este capítulo visa apresentar a validação da proposta apresentada no capítulo anterior. Foram escolhidos três sites para varredura, os quais foram separados em casos de uso e estão distribuídos neste capítulo em seções, cada seção expõe os resultados obtidos da varredura da ZAP. Segue abaixo uma breve descrição sobre os sistemas utilizados nos casos de uso:

- SisEscola: Projeto legado, desenvolvido em MVC ASP NET, que está sendo migrado de tecnologia (Angular e .NET) pelos residentes da equipe A da turma 04. com previsão de entrega para o final do ano de 2023. O SisEscola é um sistema que está no ar deste meados de 2004, que tem o intuito de capacitar e aperfeiçoar os servidores do quadro pessoal do TCE/RN, com a realização de treinamentos e eventos.

  Além da migração de sistema, o sistema está passando por evoluções para acompanhar as novas tecnologias que o sistema legado não suporta.

- INTRATCE: É o sistema interno do Tribunal de Contas, o qual atualmente possui o "nova de área restrita" como sistema interno, mas com a migração de tecnologias irá ser renomeado para INTRACE. A área restrita também é feita em MVC ASP NET e também será migrada para o Angular com .NET.

  O INTRATCE é um sistema mais robusto, os residentes da equipe B da turma 04 desenvolveram três módulos deste funcionalidade, que possuem data de entrega para o final do ano de 2023. Os módulos desenvolvidos foram: Comunicação social, Memorando eletrônico e Requerimento funcional. A ideia é que toda área restrita seja migrada ao longo do tempo, mas assim como feita na residência da turma 04, em módulos.

- TCE Admin: Embora não seja desenvolvido pelos residentes, esse sistema é usado cotidianamente para dar permissões e acessos as ferramentas do TCE/RN. Além disso, esse sistema recebeu melhorias e sustentação por um dos residentes da turma 04.

Como todas as aplicações ainda recebem novas funcionalidades, foi selecionado o dia 15 de dezembro de 2023 pra fazer todas as varreduras dos três casos de uso, para que não haja aumento de vulnerabilidades ao longo do desenvolvimento da proposta deste presente trabalho de conclusão de curso. Ademais, a discussão dos resultados obtidos nos casos de uso serão discutidas no próximo capítulo. Os casos de uso são apresentados logo abaixo:

4.1 Estudo de caso 1

O primeiro *site* analisado no projeto foi a escola de contas do Tribunal de Contas. Projeto desenvolvido pela turma 4 da residência do TCE/RN. A ferramenta ZAP foi utilizada no ambiente de *feature* (desenvolvimento). A Tabela 1 apresenta as vulnerabilidades encontradas na Escola de Contas.

Tabela 1 – Contagens de alertas por risco e confiança da Escola de Contas

		Confiança				
		Confiança do usuário	**Alto**	**Médio**	**Baixo**	**Total**
Risco	**Alto**	0 (0,0%)	1 (7,1%)	0 (0,0%)	0 (0,0%)	1 (7,1%)
	Médio	0 (0,0%)	1 (7,1%)	2 (14,3%)	1 (7,1%)	4 (28,6%)
	Baixo	0 (0,0%)	2 (14,3%)	3 (21,4%)	1 (7,1%)	6 (42,9%)
	Informativo	0 (0,0%)	1 (7,1%)	1 (7,1%)	1 (7,1%)	3 (21,4%)
	Total	0 (0,0%)	5 (35,7%)	6 (42,6%)	3 (21,4%)	14 (100%)

Como é mostrado na tabela acima, o sistema possui vulnerabilidades do nível informativo até o nível alto, listando um total de 14 vulnerabilidades diferentes. A Tabela 2 lista os nomes das vulnerabilidades encontradas e a quantidade de ocorrências no sistema.

Tabela 2 – Contagens de alertas por tipo de alerta da Escola de Contas

Tipo de Alerta	Risco	Count
O servidor vaza informações por meio dos campos de cabeçalho (Apêndice H)	Baixo	38 (271,4%)
Cabeçalho X-Content-Type-Options ausente (Apêndice J)	Baixo	35 (250,0%)
Servidor vaza informações de versão (Apêndice G)	Baixo	9 (64,3%)
Configuração Incorreta Entre Domínios (Apêndice C)	Médio	6 (42,9%)
Divulgação de Informações - Comentários Suspeitos (Apêndice L)	Informativo	3 (21,4%)
Ausência de tokens Anti-CSRF (Apêndice E)	Médio	2 (14,3%)
Divulgação de informações de identificação pessoal (Apêndice A)	Alto	1 (7,1%)
Cabeçalho da Política de Segurança de Conteúdo (Apêndice B)	Médio	1 (7,1%)
Cabeçalho anti-clickjacking ausente (Apêndice D)	Médio	1 (7,1%)
Divulgação de Data e Hora - Unix (Apêndice K)	Baixo	1 (7,1%)
Divulgação de IP Privado (Apêndice I)	Baixo	1 (7,1%)
Cabeçalho de resposta da versão X-AspNet (Apêndice F)	Baixo	1 (7,1%)
Aplicativo web moderno	Informativo	1 (7,1%)
Resposta de gerenciamento de sessão identificada	Informativo	1 (7,1%)
Total de Vulnerabilidades Diferentes		**14**

Entre os 14 tipos de vulnerabilidades encontradas, foram registradas 101 ocorrências, destacando-se os alertas o servidor vaza informações por meio dos campos de cabeçalho e cabeçalho X-Content-Type-Options Ausente, com 38 e 35 ocorrências respectivamente.

4.2 Estudo de caso 2

O estudo de caso 2 analisou as vulnerabilidades do INTRATCE. Projeto desenvolvimento pela equipe B da turma 4 da residência do TCE/RN. Assim, como no estudo de

caso anterior, a varredura foi feita em ambiente de desenvolvimento. A tabela Tabela 3 apresenta as vulnerabilidades encontradas no INTRATCE.

Tabela 3 – Contagens de alertas por risco e confiança do INTRATCE

		Confiança				
	Confiança do usuário	**Alto**	**Médio**	**Baixo**	**Total**
Risco	**Alto**	0 (0,0%)	1(4,8%)	0 (0,0%)	0 (0,0%)	1 (4,8%)
	Médio	0 (0,0%)	2 (9,5%)	2 (9,5%)	1 (4,8%)	5 (23,8%)
	Baixo	0 (0,0%)	3 (14,3%)	6 (28,6%)	0 (0%)	9 (42,9%)
	Informativo	0 (0,0%)	1 (4,8%)	2 (9,5%)	3 (14,3%)	6 (28,6%)
	Total	0 (0,0%)	7 (33,3%)	10(47,6%)	4 (19,0%)	19 (100%)

O sistema apontou 19 vulnerabilidades, variadas de informativas até o nível alto. A Tabela 4 lista os nomes das vulnerabilidades encontradas e a quantidade de ocorrências no sistema.

Tabela 4 – Contagens de alertas por tipo de alerta do INTRATCE

Tipo de Alerta	**Risco**	**Count**
O servidor vaza informações por meio dos campos de cabeçalho (Apêndice H)	Baixo	48 (228,6%)
Cabeçalho X-Content-Type-Options ausente (Apêndice J)	Baixo	45 (214,3%)
Servidor vaza informações de versão por meio do campo de cabeçalho (Apêndice G)	Baixo	29 (138,1%)
Cabeçalho Strict-Transport-Security não definido (Apêndice N)	Baixo	19 (90,5%)
Cabeçalho de resposta da versão X-AspNet (Apêndice F)	Baixo	15 (71,4%)
Divulgação de Informações - Comentários Suspeitos (Apêndice L)	Informativo	14 (66,7%)
Atributo de elemento HTML controlável pelo usuário	Informativo	12 (57,1%)
Configuração Incorreta Entre Domínios (Apêndice C)	Médio	5 (23,8%)
Cabeçalho da Política de Segurança de Conteúdo (Apêndice B)	Médio	5 (23,8%)
Cabeçalho anti-clickjacking ausente (Apêndice D)	Médio	5 (23,8%)
Aplicativo *web* moderno	Informativo	5 (23,8%)
ID da sessão na reescrita de URL (Apêndice O)	Médio	4 (19,0%)
Reexamine as diretivas de controle de cache	Informativo	4 (19,0%)
Resposta de gerenciamento de sessão identificada	Informativo	3 (14,3%)
Divulgação de informações de identificação pessoal (Apêndice A)	Alto	2 (9,5%)
Ausência de tokens Anti-CSRF (Apêndice E)	Médio	1 (4,8%)
cookie com atributo SameSite Nenhum (Apêndice P)	Baixo	1 (4,8%)
Divulgação de IP Privado (Apêndice I)	Baixo	1 (4,8%)
Recuperado do cache	Informativo	1 (4,8%)
Total de Vulnerabilidades Diferentes		**19**

Entre os 19 tipos de vulnerabilidades encontradas, foram registradas 224 ocorrências, destacando-se os alertas o servidor vaza informações por meio dos campos de cabeçalho e cacabeçalho X-Content-Type-Options Ausente, com 48 e 45 ocorrências respectivamente.

4.3 Estudo de caso 3

O último estudo de caso analisou o TCE Admin, sistema amplamente utilizado pelos residentes, servidores e terceirados do Tribunal de Contas que está em constante

Tabela 5 – Contagens de alertas por risco e confiança do TCE Admin

		Confiança				
		Confiança do usuário	**Alto**	**Médio**	**Baixo**	**Total**
Risco	**Alto**	0 (0,0%)	1 (6,2%)	0 (0,0%)	0 (0,0%)	1 (6,2%)
	Médio	0 (0,0%)	1 (4,8%)	3 (18,8%)	1 (6,2%)	5 (31,2%)
	Baixo	0 (0,0%)	3 (18,8%)	2 (12,5%)	1 (6,2%)	6 (37,5%)
	Informativo	0 (0,0%)	1 (6,2%)	2 (12,5%)	1 (6,2%)	4 (25,0%)
	Total	0 (0,0%)	6 (37,5%)	7(43,8%)	3 (18,8%)	16 (100%)

evolução. A varredura foi feita em ambiente de desenvolvimento, a Tabela 5 apresenta as vulnerabilidades encontradas no TCE Admin.

O sistema apontou 16 vulnerabilidades, variadas de informativas até o nível alto. A Tabela 6 abaixo lista os nomes das vulnerabilidades encontradas e a quantidade de ocorrências no sistema.

Tabela 6 – Contagens de alertas por tipo de alerta do TCE Admin

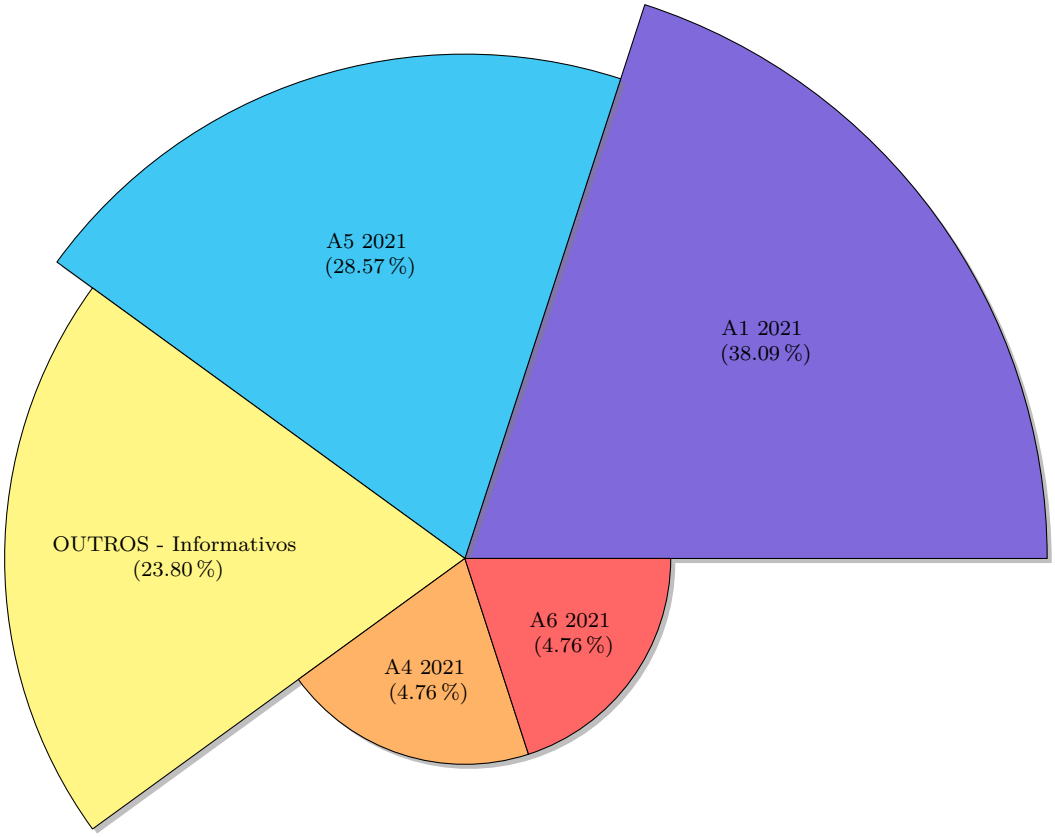
Tipo de Alerta	**Risco**	**Count**
Divulgação de Data e Hora - Unix	Baixo	40 (250,0%)
O servidor vaza informações por meio dos campos de cabeçalho	Baixo	26 (162,5%)
X-Content-Type-Options Header Missing	Baixo	26 (162,5%)
Servidor vaza informações de versão por meio do campo de cabeçalho	Baixo	8 (50,0%)
Divulgação de informações de identificação pessoal	Alto	3 (18,8%)
Configuração Incorreta Entre Domínios	Médio	3 (18,8%)
Divulgação de Informações Comentários Suspeitos	Informativo	3 (18,3%)
Resposta de gerenciamento de sessão identificada	Informativo	2 (12,5%)
Cabeçalho da Política de Segurança de Conteúdo	Médio	1 (6,2%)
Cabeçalho anti-clickjacking ausente	Médio	1 (6,2%)
Vulnerable JS Library	Médio	1 (6,2%)
Ausência de tokens Anti-CSRF	Médio	1 (6,2%)
Cabeçalho Strict-Transport-Security não definido (p18)	Baixo	1 (6,2%)
Cabeçalho Strict-Transport-Security não definido (p8)	Baixo	1 (6,2%)
Aplicativo web moderno	Informativo	1 (6,2%)
Recuperado do cache	Informativo	1 (7,1%)
Total de Vulnerabilidades Diferentes		**16**

Entre os 16 tipos de vulnerabilidades encontradas, foram registradas 109 ocorrências, destacando-se a Divulgação de Data e Hora - Unix com 40 ocorrências.

5 Discussão de resultados

Para análise dos resultados obtidos foi realizada uma quantificação das vulnerabilidades encontradas nos sistemas do Tribunal de Contas com as referências ao OWASP TOP 10 relatadas nos relatórios produzidos pela ZAP. Um resultado amplo está ilustrado na Figura 7.

Figura 7 – Divisão de vulnerabilidades em referência ao OWASP TOP 10.



De acordo com a figura acima foi possível observar que 76.20% das vulnerabilidades (A1, A4, A5 e A6) estão diretamente associadas ao OWASP 10 TOP, ou seja, relacionadas aos maiores alertas de ataque nas instituições no último estudo de 2021. No entanto, 23.80% das ocorrências que foram encontrados nos sistemas verificados são de vulnerabilidades não ligadas ao OWASP TOP 10 e possuem risco informativo, ou seja, vulnerabilidades com baixas chances de serem exploradas.

Nas seções subsequentes desse capítulo serão discutidas as vulnerabilidades encontradas nos casos de uso do capítulo anterior, onde serão apresentadas possíveis formas de solucionar ou mitigar as ameaças associadas. A ordem das seções está baseada na porcentagem que cada tópico do OWASP 10 TOP atingiu nos sistemas em ordem crescente.

5.1 A04:2021 – Design Inseguro

Com o menor índice de ocorrência em relação ao OWASP TOP 10 encontra-se o "A04:2021 - *Insecure design*", o OWASP TOP 4 de 2021. Essa é uma nova categoria da última revisão que foca em riscos relacionados a *design* e falhas de arquitetura de *software*. Vale ressaltar que essa categoria não está relacionada à implementação, mas sim, ao próprio *design* da aplicação.

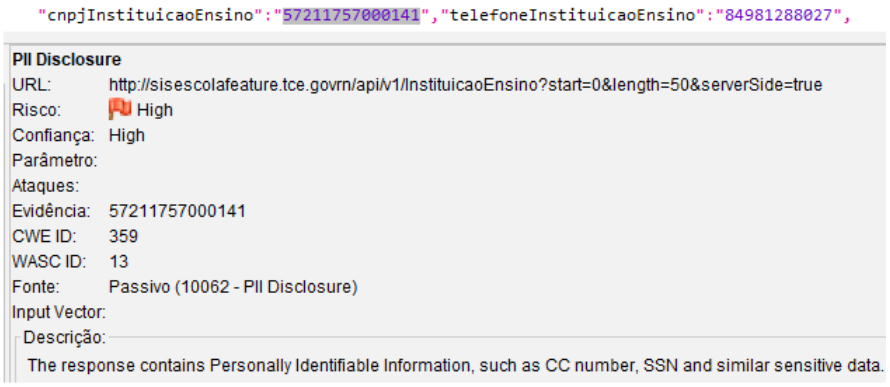
Vale evidenciar que uma aplicação com *design* seguro ainda poderá ter problemas de implementação que levam a vulnerabilidades, entretanto, um *design* inseguro não pode ser corrigido até mesmo com uma implementação perfeita.

Entre alguns problemas que podem levar a ocorrência dessa vulnerabilidade, pode-se listar:

- Má comunicação dentre as equipes de desenvolvimento;
- Ausência de padrões de *design*;
- Falta de conhecimento em práticas de *design* seguro;
- Complexidade do sistema;
- Falta de consideração de segurança nas fases iniciais do desenvolvimento;
- Configuração do servidor.

No estudo em questão, apenas uma ocorrência entre as vulnerabilidades identificadas nos casos de uso faz menção ao *Insecure design*, como listado no Apêndice A. Essa ocorrência, no entanto, destaca-se por ser a única vulnerabilidade classificada como de alto risco em todo o estudo. Ela indica a presença de uma suposta informação confidencial no sistema, mais especificamente um número de cartão de crédito, conforme mencionado também no Apêndice A.

Figura 8 – Evidência falso positivo.



Essa vulnerabilidade foi observada em todos os sistemas analisados e, após uma investigação mais detalhada, foi verificado que se tratava, na realidade, de um número de CNPJ, que não é considerado um dado pessoal no âmbito da LGPD. Assim, pode-se classificar como um falso positivo. A Figura 8 apresenta evidências que confirmam essa conclusão em relação ao alerta mencionado.

5.2 A06:2021 – Componentes vulneráveis e desatualizados

Na vulnerabilidade A06:2021 - Componentes vulneráveis e desatualizados houve apenas uma ocorrência, que consiste na utilização de códigos de terceiros, bibliotecas, que sejam vulneráveis ou desatualizados, podendo comprometer o sistema a depender da vulnerabilidade que a ocorrência impõe. A vulnerabilidade que faz menção ao Top 06 no estudo é a Biblioteca JS vulnerável, que relata que a biblioteca ckEditor 5 está desatualizada. A equipe que mantém as bibliotecas do TCE/RN foi notificada dessa vulnerabilidade e no momento de produção dos resultados desse trabalho essa vulnerabilidade já foi resolvida.

5.3 A05:2021 – Configuração incorreta de segurança

A segunda maior ocorrência de vulnerabilidades identificada está relacionada ao A05:2021 - Configuração Incorreta de Segurança, o qual ocupa a quinta posição no OWASP TOP 10 de 2021. Cerca de 28.57% das vulnerabilidades encontradas estão ligadas aos erros na configuração de ambientes de software, hardware ou rede, que podem deixar o sistema vulnerável a ataques.

De acordo com a documentação do OWASP TOP 10, essa vulnerabilidade pode ser encontrada em várias áreas, como por exemplo:

- **Configurações de Segurança e Atualizações**: Ambientes com atualizações desatualizadas ou recursos de segurança mal configurados reduzem a eficácia da proteção contra ameaças modernas.
- **Configurações de Segurança em Componentes de Software:** A falta de configurações seguras em servidores de aplicativos, *frameworks*, bibliotecas e bancos de dados pode levar a vulnerabilidades críticas, comprometendo a segurança do sistema.
- **Cabeçalhos de Segurança e Diretivas:** A ausência ou configuração inadequada de cabeçalhos de segurança e diretivas nos servidores pode expor o sistema a várias

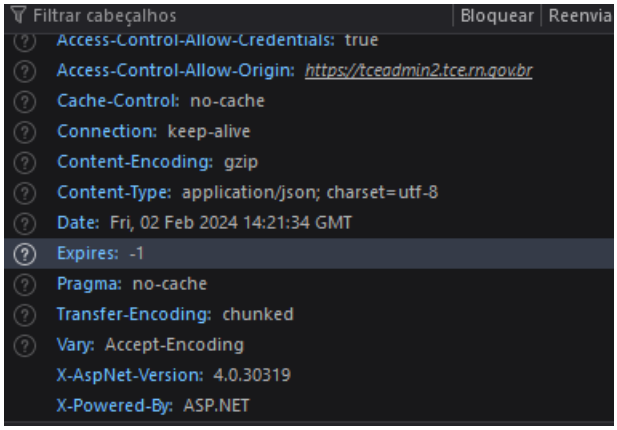
ameaças cibernéticas. [1]

- **Software Desatualizado ou Vulnerável:** O uso de *software* desatualizado ou conhecido por ser vulnerável, conforme destacado no A06:2021 do OWASP, é um risco grave que pode ser explorado por agentes mal-intencionados.

Na prática, a vulnerabilidade A05:2021 foi identificada em várias configurações de cabeçalho incorretas, como a ausência de cabeçalhos relacionados à política de segurança de conteúdo, proteção contra *clickjacking* e a exposição de informações de versão do servidor.

Em geral, todas as vulnerabilidades ligadas ao A05:2021 estão associadas a configuração incorreta de cabeçalho. A Figura 9, retirada do TCE Admin em ambiente de produção ilustra um exemplo de informações vazadas do servidor, a exemplo de sua versão e tipo.

Figura 9 – Evidência de versão do servidor exposta.



Essencialmente, a ZAP classifica esse tipo de informação como uma ameaça, pois alguns atacantes podem explorar alguma vulnerabilidade do servidor devido à versão que está desatualizada ou vulnerável. Dessa forma, foi realizado um estudo para edição de *proxys* no servidor do tribunal e, para atenuar essas vulnerabilidades, os seguintes *proxys* são apresentam-se como requisitos fundamentais:

- ***Proxy-pass-header Set-cookie***: Instrui o *Nginx* a repassar o cabeçalho *Set-cookie* do servidor *upstream* para o cliente. Isso é útil para manter a sessão do usuário entre o cliente e o servidor de aplicação.

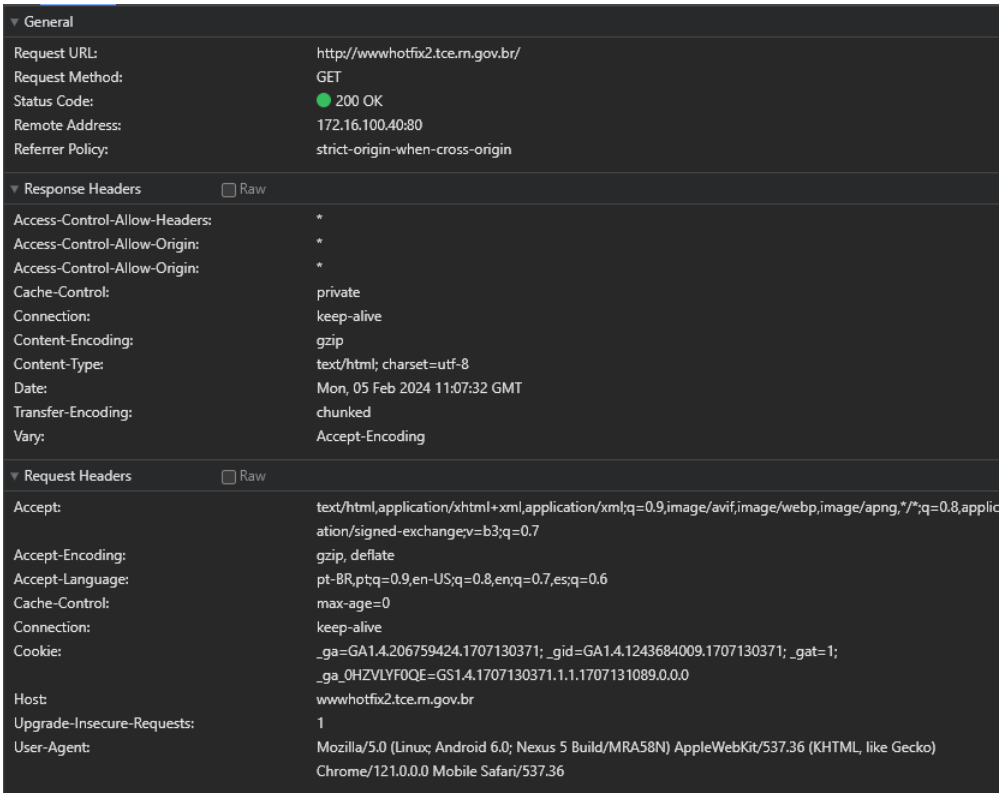
- ***Proxy-redirect off***: Desativa a reescrita de URLs nos cabeçalhos de redirecionamento do servidor *upstream*, significa que o *Nginx* não modificará os cabeçalhos de localização (Location) ou de atualização (Refresh) enviados pelo servidor *upstream*.

[1] https://owasp.org/Top10/A05-2021-Security-Misconfiguration/

- ***Proxy-set-header Accept-Encoding*** : Remove o cabeçalho *Accept-Encoding* do pedido enviado ao servidor *upstream*. Isso pode ser usado para evitar que o servidor *upstream* envie conteúdo codificado (como gzip) que o Nginx deveria descomprimir.
- ***Proxy-set-header Referer http-referer***: Define o cabeçalho Referer nos pedidos enviados ao servidor *upstream* para o mesmo valor que foi recebido do cliente, útil para aplicações que dependem do cabeçalho Referer para funcionar corretamente.
- ***Proxy-set-header Host host***: Define o cabeçalho *Host* nos pedidos enviados ao servidor *upstream* para o valor do *host* solicitado pelo cliente, ou seja, é crucial em situações onde o servidor *upstream* depende do cabeçalho Host para resolver a solicitação corretamente.
- ***Proxy-http-version 1.1***: Define a versão do HTTP para 1.1 nos pedidos ao servidor *upstream* HTTP/1.1.
- ***Remove Headers***: Essas diretivas (proxy-hide-header) instruem o Nginx a não repassar certos cabeçalhos do servidor *upstream* para o cliente. Isso é feito por motivos de segurança ou para ocultar informações sobre a tecnologia usada pelo servidor *upstream*.
- ***Proxy-set-header cookie http-cookie***: Passa os *cookies* recebidos do cliente para o servidor *upstream*, o que é essencial para manter a sessão do usuário.
- ***Proxy-set-header X-Real-IP remote-addr***: Passa o endereço IP real do cliente para o servidor *upstream*, útil para registros ou para aplicações que necessitam do IP real do cliente.
- ***Proxy-set-header X-Forwarded-For proxy-add-x-forwarded-for***: diciona o endereço IP do cliente ao cabeçalho X-Forwarded-For, usado para rastrear o endereço IP original do cliente através de múltiplos proxies.
- ***Proxy-no-cache http-pragma http-authorization***: Define condições sob as quais a resposta não será cacheada. Se os cabeçalhos *Pragma* ou Authorization estiverem presentes, o caching será desabilitado
- ***Proxy-cache-bypass http-pragma http-authorization***: Serve conteúdo não cacheado se os cabeçalhos *Pragma* ou *Authorization* estiverem presentes no pedido do cliente.
- ***Proxy-next-upstream error timeout invalid-header http-500 http-502 http-503 http-504 http-404***: Define em quais situações o *Nginx* tentará o próximo servidor *upstream* na lista. Essa é uma diretiva importante para a resiliência e alta disponibilidade, pois permite ao *Nginx* tentar outro servidor se o primeiro falhar.

A Figura 10 ilustra o ambiente do sistema TCE Admin em testes após a aplicação dos *proxys* acima listados.

Figura 10 – Evidência de correção de exposição de informações do servidor.



A implementação dessas diretrizes em um ambiente de correção demonstrou sua eficácia ao ocultar as informações de versão do servidor anteriormente expostas, como evidenciado pela comparação das figuras que ilustram a versão do servidor antes e depois das correções aplicadas. Essa abordagem não apenas melhora a segurança, mas também alinha o sistema às melhores práticas recomendadas pelo OWASP.

5.4 A01:2021 – Controle de acesso quebrado

Por fim, com 38.09% dos alertas está o OWASP TOP 1 (A01) - *broken Access Control*. Essa vulnerabilidade geralmente emerge quando desenvolvedores negligenciam o princípio do menor privilégio ao implementar sistemas, deixando brechas que podem ser exploradas de diversas formas, incluindo:

- Vazamento de informações;
- Modificação de dados;
- Destruição de dados.

Um exemplo clássico dessa falha ocorre quando, após fazer login em um sistema, um usuário consegue alterar a URL para acessar privilégios de outro tipo de usuário, como um aluno mudando sua URL para obter acesso aos privilégios de um professor, possibilitando, por exemplo, a alteração de notas.

Além disso, o Controle de Acesso Quebrado pode ser explorado através de referências diretas não seguras a objetos e técnicas como SQL *Injection*, onde invasores inserem comandos SQL maliciosos em sistemas que não validam adequadamente os dados de entrada. Os objetivos desses ataques incluem obter dados não autorizados, manipular ou destruir dados e até mesmo executar comandos do sistema.

Outra forma de exploração envolve o acesso indevido a métodos HTTP como POST, PUT, e DELETE, permitindo que usuários manipulem a aplicação de formas não intencionadas pelos desenvolvedores, como apagar ou alterar dados sensíveis.

A exploração de tokens de acesso mal configurados e a injeção de *cookies* para elevar privilégios também são mencionadas, juntamente com a configuração incorreta de CORS (Cross-Origin Resource Sharing), o qual em situações de má configuração pode permitir que recursos sejam acessados de domínios não confiáveis, expondo dados sensíveis ou possibilitando ações maliciosas.

Durante a análise, várias vulnerabilidades específicas foram associadas ao A01:2021, em destaque:

- **Configuração incorreta entre domínios**: O uso indevido do cabeçalho *Access-Control-Allow-Origin* pode permitir solicitações cruzadas de qualquer site.
- **Ausência de tokens Anti-CSRF**: Falha em verificar se as ações foram intencionalmente iniciadas pelo usuário autenticado.
- **ID da sessão na reescrita de URL**: Exposição de identificadores de sessão na URL, que pode ser interceptada.
- **Exposição de informações pelo cabeçalho 'X-Powered-By'**: Revela detalhes sobre a tecnologia do servidor.
- **Divulgação de IP Privado e Data/Hora**: Exposição de informações internas da rede e *timestamps*.
- **Divulgação de Data e Hora - Unix** : data/hora foi divulgado pelo aplicativo/-servidor web - Unix
- ***cookie* com atributo SameSite definido como None**: Permite o envio de *cookies* em solicitações de origem cruzada.
- **Comentários suspeitos**: Falsos positivos identificados pela ferramenta de varredura como potenciais vulnerabilidades.

A configuração “Incorreta entre Domínios” em todos os casos de uso ocorrem pelo mesmo motivo, o proxy *Access-Control-Allow-Origin* define como os recursos de um site podem ser solicitados de um domínio diferente do site em questão, a ocorrência que a ZAP aponta como evidência é a utilização desse proxy da seguinte forma: *Access-Control-Allow-Origin: **; permitindo que qualquer site faça solicitações cruzadas *(cross-origin requests)* para o servidor, ou seja, qualquer site pode solicitar recursos do servidor, como dados de APIs, sem restrições.

Atualmente, todas as aplicações do TCE/RN necessitam de *tokens* de autenticação para acessar os recursos das APIs, que inclusive, é uma das soluções que a própria ZAP dá como sugestão, porém, pode-se apagar os cabeçalhos não utilizados nas aplicações como uma medida de proteção adicional para esta ameaça.

A segunda vulnerabilidade que aponta a A01:2021 é a ausência de *tokens* Anti-CSRF, que ocorre quando um atacante consegue forçar um usuário a executar ações indesejadas em um aplicativo *web* em que ele está autenticado, sem que o usuário esteja ciente ou tenha a intenção de realizá-las.

O motivo principal dessa vulnerabilidade existir nos sistemas do TCE/RN é que o aplicativo *web* não verifica se as solicitações recebidas de um usuário foram intencionalmente iniciadas por ele. Sendo assim, o aplicativo falha em assegurar que as ações são legítimas e não foram forjadas por um terceiro.

Para prevenir essa vulnerabilidade existem diversas possibilidades, destacando-se a utilização de um token Anti-CSRF, o qual deve ser único para cada sessão do usuário e verificado pelo servidor a cada solicitação, como também a geração de um alerta de solicitação de confirmação de ações do usuário.

A vulnerabilidade seguinte é a de “ID da sessão na reescrita de URL”, que ocorre quando o identificador único de uma sessão de usuário é passado como parte da URL, que pode ser uma prática adotada para manter o estado da sessão entre as solicitações HTTP.

Para solucionar problemas relacionados a essa vulnerabilidade é necessário estabelecer um *token* de sessão seguro e tempo de expiração para os mesmo. Os sistemas do TCE/RN já possuem essas validações nas aplicações, logo, essa vulnerabilidade não será facilmente explorada nos sistemas do tribunal.

A próxima vulnerabilidade é a “Exposição de Informações pelo Cabeçalho *X-Powered-By*” que em todas as suas evidências nos casos de uso relatam a seguinte informação> X-Powered-By: ASP.NET. Essa vulnerabilidade ocorre devido a exposição de informações sobre o sistemas e como mostra a sessão 5.3 deste mesmo capítulo, podemos resolver esses problema com a utilização de *proxys* no servidor.

Outra vulnerabilidade que aborda o controle de acesso quebrado é a "Divulgação de IP Privado", a qual representa a falha de segurança, pois fornece a atacantes informações sobre a estrutura da rede interna de uma organização. Como solução para os problemas

relacionados a essa vulnerabilidade é a remoção das referências diretas nos códigos das respostas enviadas ao cliente das aplicações.

A vulnerabilidade de Divulgação de Data e Hora - Unix é um falso positivo nos sistemas, dado que a evidência da ocorrência mostra uma numeração que não corresponde a uma data. A figura Figura 11 ilustra a evidência que a ZAP utilizou para justificar a vulnerabilidade nos sistemas do TCE/RN.

Figura 11 – Evidência coletada pela ZAP sobre exposição de Data e Hora

A última vulnerabilidade de nível baixo de risco no OWASP TOP 1 é a "*cookie* com atributo *SameSite none*", a qual ocorre devido à configuração do atributo *SameSite* dos *cookies* como *None*, permitindo que os *cookies* sejam enviados em solicitações de origem cruzada (*cross-site requests*). O atributo SameSite foi introduzido para dar um controle mais preciso sobre como os *cookies* são enviados em solicitações entre sites, oferecendo uma camada adicional de proteção. Para resolver esse problema, basta definir o atributo SameSite como Strict ou Lax.

Strict: O *cookie* só é enviado em solicitações dentro do mesmo site.

Lax: Permite que os *cookies* sejam enviados em solicitações *cross-site* quando o usuário está navegando para o site de origem.

Finalizando as vulnerabilidades que fazem menção ao A01:2021 temos a divulgação de comentários suspeitos. Foram verificadas todas as ocorrências dessa vulnerabilidade e pode-se afirmar que também é um falso positivo, pois a ferramenta seleciona palavras-chave na aplicação e evidencia como comentários suspeitos, como por exemplo a palavra "*select*" e "*debug*" . Vale ressaltar que na compilação do *TypeScript* para o *JavaScript* os comentários são todos removidos antes da chegada ao servidor.

Neste capítulo, observou-se que a maioria das vulnerabilidades está diretamente associada aos maiores riscos de ataque OWASP, com destaque para as categorias A01 (Controle de Acesso Quebrado), A05 (Configuração Incorreta de Segurança), A06 (Componentes Vulneráveis e Desatualizados) e A04 (Design Inseguro). As soluções e mitigações propostas para cada categoria de vulnerabilidade, incluindo a implementação de práticas de design seguro, atualização de componentes, configuração adequada de cabeçalhos de segurança e uso de *tokens* de autenticação, demonstram um caminho claro para a melhoria da segurança dos sistemas analisados, o próximo capítulo abordará a conclusão deste trabalho baseado nos resultados e discussões advindas dessa capítulo em questão.

6 Conclusão

Conforme abordado ao longo deste trabalho foi possível observar que os sistemas produzidos pelo TCE-RN possuem um nível de segurança satisfatório. Considerando um espaço de 22 tipos de vulnerabilidades identificadas nos casos de uso examinados, somente uma foi categorizada como de risco alto, porém, esta foi considerada um falso positivo após análise detalhada da evidência e analisada no capítulo 5.

Além disso, o uso da ferramenta ZAP trouxe contribuições significativas, dado que a adoção dessa ferramenta possibilitou a realização de melhorias nos servidores dos sistemas de produção do TCE/RN, como a ocultação de informações nos cabeçalhos das resposta e atualização da biblioteca *ckEditor* 5, reduzindo potenciais vulnerabilidades resultantes dessa condição.

Por outro lado, as diferentes formas de metodologias das equipes foi um obstáculo para a produção da ideia inicial do trabalho, uma vez que a criação de um plano de desenvolvimento unificado para as equipes de desenvolvimento e sustentação do TCE/RN seria uma alternativa inviável. Dessa forma, foi proposto aos desenvolvedores a realização de pequenas alterações de segurança em seus códigos como, por exemplo, a implementação de uma verificação de CORS que impede o deploy de uma alteração de um código vulnerável.

No entanto, apesar dos desafios encontrados, a utilização da ferramenta ZAP nos sistemas do Tribunal de Contas é uma alternativa viável, considerando-se que é gratuita e recebe atualizações constantes, contando com um forte apoio da comunidade de segurança da informação.

6.1 Trabalhos futuros

Como sugestão para trabalhos futuros, sugere-se a utilização da ferramenta ZAP como *proxy* acoplado às requisições do *Cypress*, visto que se mostrou ser uma alternativa rápida e prática para a varredura nos sistemas do Tribunal. Para isso, é fundamental que a ferramenta ZAP esteja integrada na *pipeline* de todos os sistemas do tribunal, condição que não ocorreu durante a confecção deste trabalho. Outra perspectiva a ser explorada consiste em realizar a padronização da ferramenta *Cypress* nos sistemas do TCE-RN através da integração da ferramenta ZAP na *pipeline* com as requisições do *Cypress*, assim como abordado no Capítulo 3.

Referências

AGOSTINI, M. A cibernÉtica sob a Ótica do fenÔmeno da guerra e da agenda de seguranÇa. Universidade Federal de Santa Catarina, ago 2014. Disponível em: <https://repositorio.ufsc.br/xmlui/handle/123456789/124695>. Citado na página 19.

BARCELOS, A. et al. *LEI GERAL DE PROTEÇÃO DE DADOS E O PAPEL DO DPO*. [S.l.]: Revista Projetos Extensionistas |Faculdade de Pará de Minas -FAPAM, 2021. v. 1. Citado na página 18.

BRASIL. Tecnologia da informação — técnicas de segurança — código de prática para a gestão da segurança da informação. *Diário Oficial [da] República Federativa do Brasil*, Brasília, DF, 2005. ISSN 1677-7042. Disponível em: <https://www.facom.ufu.br/~william/Disciplinas%202012-2/BSI%20-%20Auditoria%20e%20Seguranca/Material%20Adicional/NBR%20ISO-IEC%2017799-2005-PORTUGUES.pdf>. Citado na página 18.

FILHO, M. *Segurança e Defesa Cibernética: da fronteira física aos muros virtuais.* [S.l.]: Editora UFPE, 2014. Citado na página 20.

FRANZESE, L. Um estudo das vulnerabilidades do owasp top 10 na plataforma moodle. Universidade Federal de Uberlândia, jun. 2023. Disponível em: <https://repositorio.ufu.br/handle/123456789/38406>. Citado 2 vezes nas páginas 23 e 24.

GALOYAN, A. *Segurança cibernética no âmbito das relações internacionais.* Tese (Doutorado) — Biblioteca Central da UNB, 2019. Disponível em: <http://dx.doi.org/10.26512/2019.TCC.22386>. Citado na página 20.

HARRIS, S.; MIAMI, F. *All-in-One CISSP® All-in-One Exam Guide*. [S.l.]: Mcgraw Hill, 2019. v. 7. Citado na página 17.

LI, Y.; LIU, Q. A comprehensive review study of cyber-attacks and cyber security - emerging trends and recent developments. *Energy Reports*, Elsevier BV, v. 7, p. 8176–8186, nov. 2021. Disponível em: <https://doi.org/10.1016/j.egyr.2021.08.126>. Citado na página 20.

LOURENÇO, R. M.; DUARTE, R. P. Gestão de segurança da informação. 004, 2020. Citado na página 17.

MANDARINO, R. *Seguranca E Defesa Do Espaco Cibernetico Brasileiro.* [S.l.]: CUBZAC, 2010. v. 1. Citado na página 19.

Ministério da Defesa. *Portaria Normativa No 3.010/MD, de 18 de Novembro de 2014.* 2014. Diário Oficial da União, nº 224, 19 de novembro de 2014. Aprova a Doutrina Militar de Defesa Cibernética. Citado na página 19.

NIST. 1901. <https://www.nist.gov/>. Acessado em: data de acesso. Citado 3 vezes nas páginas 20, 21 e 22.

NIST-800-171. *NIST Special Publication 800-171, Computer Security Incident Handling Guide.* [S.l.], 2022. Acessado em: 17 de dezembro de 2023. Citado na página 21.

NIST-800-30. *NIST Special Publication 800-30, Guide for Conducting Risk Assessments*. [S.l.], 2022. Acessado em: 17 de dezembro de 2023. Citado na página 21.

NIST-800-37. *NIST Special Publication 800-37,Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy*. [S.l.], 2022. Acessado em: 17 de dezembro de 2023. Citado na página 21.

NIST-800-61. *NIST Special Publication 800-61, Computer Security Incident Handling Guide*. [S.l.], 2023. Acessado em: 17 de dezembro de 2023. Citado na página 21.

OLIVEIRA, M. A. *DEFESA CIBERNÉTICA E MOBILIZAÇÃO NACIONAL*. [S.l.]: Editora UFPE, 2020. Citado na página 20.

OWASP. 2001. <https://www.owasp.org/>. Acessado em: 2023. Citado 2 vezes nas páginas 22 e 23.

RID, T. *Cyber war will not take place. Oxford University Press*. [S.l.]: OXFORD University Press, 2013. v. 1. Citado 2 vezes nas páginas 19 e 20.

UNIVERSITY, O. Human development report 1994. Oxford University, 1994. Disponível em: <https://hdr.undp.org/sites/default/files/private/documents/ /hdr1994encompletenostatspdf.pdf>. Citado na página 19.

VENTRE, D. *XIX Curso Internacional de Defensa*: Seguridad global y potencias emergentes en un mundo multipolar. Jaca, 26 al 30 de septiembre de 2011: [s.n.], 2011. Academia General Militar – Universidad de Zaragoza. Citado na página 19.

APÊNDICE A – Divulgação de informações de identificação pessoal

- **Alert tags**: OWASP 2021 A04, OWASP 2017 A03
- **Descrição**: A resposta contém informações de identificação pessoal, como número CC, SSN e dados confidenciais semelhantes.

  Tipo de cartão de crédito detectado: Maestro

  Número de Identificação Bancária: 572117

  Marca: MAESTRO

  Categoria: PADRÃO

  Evidência: 57211757000141
- **Solução**:Verifique a resposta quanto à presença potencial de informações de identificação pessoal (PII) e garanta que nada confidencial seja vazado pelo aplicativo.

APÊNDICE B – Cabeçalho da Política de Segurança de Conteúdo (CSP) não definido

- **Alert tags**: OWASP 2021 A05, OWASP 2017 A06
- **Descrição**: A Política de Segurança de Conteúdo (CSP) é uma camada adicional de segurança que ajuda a detectar e mitigar certos tipos de ataques, incluindo Cross Site Scripting (XSS) e ataques de injeção de dados. Esses ataques são usados para tudo, desde roubo de dados até destruição de sites ou distribuição de malware. O CSP fornece um conjunto de cabeçalhos HTTP padrão que permitem aos proprietários de sites declarar fontes aprovadas de conteúdo que os navegadores devem ter permissão para carregar naquela página - os tipos cobertos são JavaScript, CSS, quadros HTML, fontes, imagens e objetos incorporáveis, como miniaplicativos Java, ActiveX, arquivos de áudio e vídeo.
- **Solução**:Certifique-se de que seu servidor web, servidor de aplicativos, balanceador de carga, etc. esteja configurado para definir o cabeçalho Content-Security-Policy.

APÊNDICE C – Configuração Incorreta Entre Domínios

- **Alert tags**: OWASP 2021 A01, OWASP 2017 A05
- **Descrição**: A configuração incorreta do CORS no servidor da web permite solicitações de leitura entre domínios de domínios arbitrários de terceiros, usando APIs não autenticadas neste domínio. No entanto, as implementações do navegador da Web não permitem que terceiros arbitrários leiam a resposta de APIs autenticadas. Isso reduz o risco um pouco. Essa configuração incorreta pode ser usada por um invasor para acessar dados que estão disponíveis de maneira não autenticada, mas que usam alguma outra forma de segurança, como lista de permissões de endereços IP.
- **Solução**:Certifique-se de que dados confidenciais não estejam disponíveis de maneira não autenticada (usando uma lista branca/de permissões de endereços IP, por exemplo). Configure o cabeçalho HTTP "Access-Control-Allow-Origin"para um conjunto mais restritivo de domínios ou remova todos os cabeçalhos CORS inteiramente, para permitir que o navegador web aplique a Same Origin Policy (SOP) de uma maneira mais restritiva.

APÊNDICE D – Cabeçalho anti-clickjacking ausente

- **Alert tags**: OWASP 2021 A05, OWASP 2017 A06, WSTG-V42-CLNT-09
- **Descrição**: A resposta não inclui Content-Security-Policy com diretiva 'frame-ancestors' ou X-Frame-Options para proteção contra ataques de 'ClickJacking'.
- **Solução**:Os navegadores modernos suportam os cabeçalhos HTTP Content-Security-Policy e X-Frame-Options. Certifique-se de que um deles esteja definido em todas as páginas da web retornadas pelo seu site/aplicativo.

  Se você espera que a página seja enquadrada apenas pelas páginas do seu servidor (por exemplo, faz parte de um FRAMESET), você deve usar SAMEORIGIN; caso contrário, se você nunca espera que a página seja enquadrada, você deve usar DENY. Como alternativa, considere implementar a diretiva "frame-ancestors"da Política de Segurança de Conteúdo.

APÊNDICE E – Ausência de tokens Anti-CSRF

- **Alert tags**: OWASP 2021 A01, OWASP 2017 A05, WSTG-V42-SESS-05
- **Descrição**: Não foram localizados tokens Anti-CSRF no formulário de submissão HTML.

  Uma falsificação de solicitação entre sites (Cross-Site Request Forgery ou simplesmente CSRF) é um ataque que envolve forçar a vítima a enviar uma solicitação HTTP a um destino alvo sem seu conhecimento ou intenção, a fim de realizar uma ação como a vítima. A causa implícita é a funcionalidade do aplicativo usando ações previsíveis em URLs/formulários, de maneira repetível. A natureza do ataque é que o CSRF explora a confiança que um site tem em um usuário. Em contrapartida, um ataque do tipo Cross-Site Scripting (XSS) explora a confiança que um usuário tem em um site. Como o XSS, os ataques CSRF não são necessariamente entre sites, mas também podem ser. A falsificação de solicitação entre sites também é conhecida por "CSRF", "XSRF", "one-click attack", "session riding", "confused deputy", e "sea surf".

  Os ataques CSRF são efetivos em várias situações, incluindo:

  * - A vítima tem uma sessão ativa no site de destino;

  * - A vítima está autenticada por meio de autenticação HTTP no site de destino;

  * - A vítima está na mesma rede local do site de destino.

  O CSRF era usado principalmente para executar ações contra um site-alvo usando os privilégios da vítima, mas técnicas recentes foram descobertas para vazamento de informações obtendo acesso às respostas. O risco de vazamento/divulgação não autorizada de informações aumenta drasticamente quando o site de destino é vulnerável a XSS, porque o XSS pode ser usado como uma plataforma para CSRF, permitindo que o ataque opere dentro dos limites da política de mesma origem.

- **Solução**:Fase: Arquitetura e Design.

  Use uma biblioteca verificada ou framework que não permita que essa vulnerabilidade ocorra, ou forneça construções/implementações que tornem essa vulnerabilidade mais fácil de evitar.

  Por exemplo, use pacotes anti-CSRF, como o OWASP CSRFGuard.

  Fase: Implementação.

Certifique-se de que seu aplicativo esteja livre de problemas de cross-site scripting (XSS), porque a maioria das defesas CSRF pode ser contornada usando script controlado por invasor.

Fase: Arquitetura e Design.

Gere um número arbitrário de uso único e exclusivo (ou Nonce = "N"de "number-número em inglês - e "once"de "uma vez"também em inglês) para cada formulário, coloque o nonce no formulário e verifique-o ao receber o formulário. Certifique-se de que o nonce não seja previsível (CWE-330).

Observe que isso pode ser contornado usando XSS.

Identifique operações especialmente perigosas. Quando o usuário realizar uma operação perigosa, envie uma solicitação de confirmação separada para garantir que o usuário pretendia realizar aquela operação.

Observe que isso pode ser contornado usando XSS.

Utilize o controle ESAPI Session Management.

Este controle inclui um componente para CSRF.

Não use o método GET para qualquer solicitação que acione uma mudança de estado.

Fase: Implementação.

Verifique o cabeçalho HTTP Referer para ver se a solicitação foi originada de uma página esperada. Isso pode interromper funcionalidades legítimas, porque os usuários ou proxies podem ter desativado o envio do Referer por motivos de privacidade."frame-ancestors"directive.

APÊNDICE F – Cabeçalho de resposta da versão X-AspNet

- **Alert tags**: OWASP 2021 A05, OWASP 2017 A06, WSTG-V42-INFO-08
- **Descrição**: O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP "X-AspNet-Version"/"X-AspNetMvc-Version".
- **Solução**:Configure o servidor para que ele não retorne esses cabeçalhos.

APÊNDICE G – Servidor vaza informações de versão por meio do campo de cabeçalho de resposta HTTP “Servidor”

- **Alert tags**: OWASP 2021 A05, OWASP 2017 A06, WSTG-V42-INFO-08
- **Descrição**:O servidor web/aplicativo está vazando informações de versão por meio do cabeçalho de resposta HTTP "Servidor". O acesso a essas informações pode facilitar que invasores identifiquem outras vulnerabilidades às quais seu servidor web/aplicativo está sujeito.
- **Solução**:Certifique-se de que seu servidor web, servidor de aplicativos, balanceador de carga, etc. esteja configurado para suprimir o cabeçalho "Servidor"ou fornecer detalhes genéricos.

APÊNDICE H – O servidor vaza informações por meio dos campos de cabeçalho de resposta HTTP 'X-Powered-By'

- **Alert tags**: OWASP 2021 A01, OWASP 2017 A03, WSTG-V42-INFO-08
- **Descrição**:O servidor da web/aplicativo está vazando informações por meio de um ou mais cabeçalhos de resposta HTTP "X-Powered-By". O acesso a essas informações pode facilitar que os invasores identifiquem outras estruturas/componentes dos quais seu aplicativo da web depende e as vulnerabilidades às quais esses componentes podem estar sujeitos.
- **Solução**:Certifique-se de que seu servidor web, servidor de aplicativos, balanceador de carga, etc. esteja configurado para suprimir cabeçalhos "X-Powered-By".

APÊNDICE I – Divulgação de IP Privado

- **Alert tags**: OWASP 2021 A01, OWASP 2017 A03
- **Descrição**:Um IP privado (como 10.x.x.x, 172.x.x.x, 192.168.x.x) ou um nome de host privado do Amazon EC2 (por exemplo, ip-10-0-56-78) foi encontrado no corpo da resposta HTTP. Esta informação pode ser útil para futuros ataques direcionados a sistemas internos.
- **Solução**:Remova o endereço IP privado do corpo da resposta HTTP. Para comentários, use comentários JSP/ASP/PHP em vez de comentários HTML/JavaScript que podem ser vistos pelos navegadores clientes.

APÊNDICE J – Cabeçalho X-Content-Type-Options ausente

- **Alert tags**: OWASP 2021 A05, OWASP 2017 A06
- **Descrição**:O cabeçalho X-Content-Type-Options do Anti-MIME-Sniffing não foi definido como 'nosniff'. Isso permite que versões mais antigas do Internet Explorer e do Chrome executem detecção MIME no corpo da resposta, potencialmente fazendo com que o corpo da resposta seja interpretado e exibido como um tipo de conteúdo diferente do tipo de conteúdo declarado. As versões atuais (início de 2014) e legadas do Firefox usarão o tipo de conteúdo declarado (se houver algum definido), em vez de executar a detecção de MIME.
- **Solução**:Certifique-se de que o aplicativo/servidor web defina o cabeçalho Content-Type adequadamente e que defina o cabeçalho X-Content-Type-Options como 'nosniff' para todas as páginas da web.

  Se possível, certifique-se de que o usuário final use um navegador da Web moderno e compatível com padrões que não execute detecção de MIME ou que possa ser direcionado pelo aplicativo da Web/servidor da Web para não executar detecção de MIME

APÊNDICE K – Divulgação de Data e Hora - Unix

- **Alert tags**: OWASP 2021 A01, OWASP 2017 A03
- **Descrição**:Um carimbo de data/hora foi divulgado pela aplicação/servidor web - Unix
- **Solução**:Confirme manualmente se os dados do carimbo de data/hora não são confidenciais e se os dados não podem ser agregados para divulgar padrões exploráveis.

APÊNDICE L – Divulgação de Informações - Comentários Suspeitos

- **Alert tags**: OWASP 2021 A01, OWASP 2017 A03, WSTG-v42-INFO-05
- **Descrição**:A resposta parece conter comentários suspeitos que podem ajudar um invasor. Observação: as correspondências feitas em blocos de script ou arquivos são referentes a todo o conteúdo e não apenas aos comentários.
- **Solução**:Remova todos os comentários que retornam informações que podem ajudar um invasor e corrigir quaisquer problemas subjacentes aos quais eles se referem.

APÊNDICE M – Biblioteca JS vulnerável

- **Alert tags**: OWASP 2017 A09, OWASP 2021 A06, CVE-2021-21254,CVE-2021-21391, CVE-2022-31175
- **Descrição**:A biblioteca identificadackeditor5, versão 23.0.0 é vulnerável.
- **Solução**:Atualize para a versão mais recente do ckeditor5.

APÊNDICE N – Cabeçalho Strict-Transport-Security não definido

- **Alert tags**: OWASP 2021 A05, OWASP 2017 A06
- **Descrição**:HTTP Strict Transport Security (HSTS) é um mecanismo de política de segurança da web por meio do qual um servidor da web declara que os agentes de usuário em conformidade (como um navegador da web) devem interagir com ele usando apenas conexões HTTPS seguras (ou seja, HTTP em camadas sobre TLS/SSL). HSTS é um protocolo de rastreamento de padrões IETF e é especificado na RFC 6797.
- **Solução**:Certifique-se de que seu servidor web, servidor de aplicativos, balanceador de carga, etc. esteja configurado para impor Strict-Transport-Security.

APÊNDICE O – ID da sessão na reescrita de URL

- **Alert tags**: OWASP 2021 A01, WSTG-v42-SESS-04, OWASP 2017 A03
- **Descrição**:A reescrita de URL é usada para rastrear o ID da sessão do usuário. O ID da sessão pode ser divulgado através do cabeçalho do referenciador entre sites. Além disso, o ID da sessão pode ser armazenado no histórico do navegador ou nos logs do servidor.
- **Solução**:Para conteúdo seguro, coloque o ID da sessão em um cookie. Para ser ainda mais seguro, considere usar uma combinação de cookie e reescrita de URL.

APÊNDICE P – Cookie com atributo SameSite Nenhum

- **Alert tags**: OWASP 2021 A01, WSTG-v42-SESS-02, OWASP 2017 A05
- **Descrição**:Um cookie foi definido com seu atributo SameSite definido como "none", o que significa que o cookie pode ser enviado como resultado de uma solicitação 'entre sites'. O atributo SameSite é uma contramedida eficaz para falsificação de solicitação entre sites, inclusão de script entre sites e ataques de temporização.
- **Solução**:Certifique-se de que o atributo SameSite esteja definido como 'lax' ou, de preferência, 'strict' para todos os cookies.