UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

Ricardo Henrique Rodrigues de Castro

Desenvolvimento de uma artefato para aprendizado sobre segurança da informação em APIs

Recife

Maio de 2023

Ricardo Henrique Rodrigues de Castro

Desenvolvimento de uma artefato para aprendizado sobre segurança da informação em APIs

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE
Departamento de Estatística e Informática
Curso de Bacharelado em Sistemas de Informação

Orientador: Rodrigo Elia Assad

Recife
Maio de 2023

RICARDO HENRIQUE RODRIGUES DE CASTRO

Desenvolvimento de uma artefato para aprendizado sobre segurança da informação em APIs

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em: 15 de Maio de 2023.

BANCA EXAMINADORA

Rodrigo Elia Assad (Orientador)
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Cleviton Vinicius Fonsêca Monteiro
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Desenvolvimento de uma artefato para aprendizado sobre segurança da informação em APIs

Ricardo H. R. de Castro [1], **Rodrigo Elia Assad** [1], **Cleviton V. F Monteiro** [1]

[1]Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco
Rua Dom Manuel de Medeiros, s/n, - CEP: 52171-900 – Recife – PE – Brasil

ricardo.castro@ufrpe.br, rodrigo.assad@ufrpe.br, cleviton.monteiro@ufrpe.br

Resumo. *Nos dias atuais, as Application Programming Interfaces (APIs) desempenham um papel fundamental nas aplicações, permitindo a integração entre diferentes sistemas. No entanto, devido à sensibilidade dos dados e informações pessoais que as APIs lidam, frequentemente elas são alvos de ataques realizados por agentes maliciosos. Para auxiliar desenvolvedores e analistas de segurança, o Open Web Application Security Project (OWASP) publicou uma lista dos dez problemas mais comuns em APIs, visando identificá-los e fornecer indicações de como resolvê-los. Neste contexto, este artigo propõe uma abordagem inovadora para o ensino-aprendizado sobre segurança da informação em APIs, utilizando o método de ensino-aprendizagem chamado Aprendizagem Baseada em Problemas (ABP), esta abordagem permitirá que os alunos se engajem ativamente na resolução de desafios reais relacionados à segurança em APIs, pois, eles serão expostos a problemas autênticos e desenvolverão habilidades práticas de análise, identificação de vulnerabilidades e aplicação de contramedidas.*

Abstract. *In today's world, Application Programming Interfaces (APIs) play a crucial role in applications by enabling integration between different systems. However, due to the sensitivity of the data and personal information that APIs handle, they are often targeted by malicious actors. To assist developers and security analysts, the Open Web Application Security Project (OWASP) has published a list of the top ten most common API security problems, aiming to identify and provide guidance on resolving them. In this context, this article proposes an innovative approach to learning security in APIs, utilizing the problem-based learning method known as Problem-Based Learning (PBL). This approach will actively engage students in solving real challenges related to API security, exposing them to authentic problems and developing practical skills in analysis, vulnerability identification, and countermeasure application.*

1. Introdução

Uma *Application Programming Interface* (API) pode ser definida como um conjunto de normas que permite a comunicação entre *softwares* de plataformas distintas através de protocolos e formatos específicos. Um dos principais benefícios em relação à utilização de uma API é o fato de não precisarmos escrever todo o código para consumir um determinado recurso, o que trás agilidade para o processo de desenvolvimento de *software*. Por

exemplo, um desenvolvedor que queira criar um aplicativo que mostre as fotos armazenadas no seu dispositivo, pode utilizar as APIs do próprio sistema operacional para realizar tal integração.

É comum que muitas destas APIs manipulem informações potencialmente sensíveis, como informações pessoais, financeiras e de saúde. Um exemplo deste tipo de API são as APIs de compartilhamento de dados bancários pessoais mais conhecida como *Open Banking* desenvolvidas e mantidas pelas instituições financeiras do Brasil.

Por este motivo, conforme demonstrado no estudo *State of API Security, Q1 2022* (Labs, 2022) promovido pela empresa *Salt Security*, o número de ataques realizados por agentes maliciosos nas APIs cresceu cerca de 681% entre março de 2021 e março de 2022. Neste mesmo estudo foi constatado que cerca de 34% das empresas entrevistadas não realizavam testes de segurança nas suas APIs, 85% delas consideravam as ferramentas de segurança que eles possuíam eram ineficientes contra ataques em APIs e que 35% consideravam que não possuíam a *expertise* necessária para implementar as proteções de segurança em suas APIs.

Diante deste cenário, o ensino-aprendizagem sobre segurança da informação em APIs desempenha um papel fundamental na capacitação de profissionais da área de tecnologia, para que estes sejam capazes de desenvolver APIs robustas, confiáveis e seguras.Alguns dos principais benefícios do ensino-aprendizagem em segurança da informação em APIs são:

- Conscientização dos riscos: O ensino-aprendizagem sobre segurança da informação em APIs ajuda os alunos a entenderem os riscos associados ao uso inadequado ou não seguro das APIs. Isso inclui a compreensão das vulnerabilidades comuns e das melhores práticas para mitigar esses riscos;
- Conhecimento de técnicas de proteção: Os alunos aprendem técnicas e estratégias eficazes para proteger as suas APIs contra ataques. Isso envolve a compreensão de métodos de autenticação segura, criptografia, validação de entrada, controle de acesso e monitoramento de atividades suspeitas;
- Habilidades de detecção e resposta a incidentes: O ensino-aprendizagem sobre segurança da informação em APIs capacita os alunos a identificarem e responderem rapidamente a incidentes de segurança. Eles aprendem a detectar atividades suspeitas, investigar possíveis violações e tomar medidas corretivas adequadas;
- Preparação para carreiras em segurança: O ensino-aprendizagem sobre segurança da informação em APIs prepara os alunos para carreiras em segurança cibernética, fornecendo as habilidades necessárias para enfrentar os desafios do mundo real. Eles se tornam profissionais capazes de proteger efetivamente as APIs que eles desenvolvem e garantir a integridade e confidencialidade dos dados;
- Proteção dos dados e reputação das organizações: A segurança da informação em APIs desempenha um papel crucial na proteção dos dados e na preservação da reputação das organizações. Os profissionais treinados nessa área ajudam a garantir que as APIs sejam implementadas e mantidas com os mais altos padrões de segurança, minimizando os riscos de violações de dados e outros incidentes de segurança.

Para auxiliar a comunidade de desenvolvimento de *software* e analistas de segurança da informação, a fundação *Open Web Application Security Project* (OWASP)

publicou em 2019 uma lista denominada *OWASP API Security Top 10*, que contém as dez vulnerabilidades mais encontradas em APIs bem como uma documentação de como identificá-las e corrigi-lás. Apesar de ter sido publicada em 2019, esta lista se mantém atual e relevante, pois, até o momento em que este artigo é escrito, nenhuma nova versão desta lista foi publicada.

O objetivo deste trabalho e a explicação sobre cada vulnerabilidade proposta pela OWASP serão elucidados nos próximos capítulos.

1.1. Objetivos gerais

O objetivo principal deste trabalho é propor uma nova abordagem para o aprendizado sobre segurança da informação em APIs, utilizando como base o método de ensino-aprendizagem a Aprendizagem Baseada em Problemas (ABP). Através dessa abordagem, os participantes serão capazes de desenvolver habilidades práticas na identificação, análise e resolução dos principais problemas de segurança enfrentados pelas APIs, utilizando como base, o OWASP API Security TOP 10.

1.2. Objetivos específicos

Principais objetivos específicos:

1. Definir o método de ensino-aprendizagem que será utilizado;
2. Desenvolver uma API que utilize o método de ensino escolhido para abordar cada um dos problemas de segurança propostos pelo OWASP TOP 10 API Security, onde cada funcionalidade da API conterá um ou mais problemas de segurança. Para tal, a API deverá possuir as seguintes funcionalidades:
    - Uma funcionalidade de *login* que esteja vulnerável a ataques de força bruta e de credential stuffing;
    - Uma funcionalidade de exportar comprovante de pagamento que esteja vulnerável a injeção de código HTML/JavaScript em conversores de código HTML para PDF;
    - Implementar um *JSON Web Token* (JWT) que utilize um *secret* com baixa entropia e esteja passível a ataques de força bruta;
    - Uma funcionalidade de monitoramento que esteja vulnerável ao problema de subversão de filtragens baseadas em URL;
    - Uma funcionalidade de visualizar perfil que não possua nenhum mecanismo de controle de acesso atrelado;
    - Em cada funcionalidade, retornar uma mensagem de erro descritiva, contendo por exemplo, a pilha de erros da aplicação;
    - Uma funcionalidade de desativar conta que esteja vulnerável à atribuição de variáveis em massa;
    - Uma funcionalidade de consultar saldo que esteja vulnerável à injeção de código NoSQL;
    - Uma funcionalidade de transferir saldo que esteja vulnerável à escalação horizontal de privilégios;
3. Desenvolver uma documentação robusta que servirá como material de apoio, contendo um tutorial de como utilizar a API, detalhes sobre a sua estrutura e uma explicação sobre os problemas de segurança propostos;

4. Realizar experimentos com pessoas interessadas em aprender sobre segurança da informação em APIs, com o intuito de validar o artefato desenvolvido;
5. Realizar uma verificação de aprendizagem com os estudantes envolvidos nos experimentos;
6. Implementar um ciclo de melhoria contínua do artefato, cujas melhorias serão implementadas a partir dos *feedbacks* dos envolvidos nos ciclos de experimentação.

2. Referencial teórico

Esta seção trará, a partir de uma extensa revisão bibliográfica, os principais conceitos dos problemas de segurança que serão abordados pela API desenvolvida durante este projeto e uma breve reflexão sobre o tema de aprendizagem baseada em problemas.

2.1. Aprendizagem baseada em problemas

A Aprendizagem baseada em problemas (ABP) é uma abordagem educacional que utiliza problemas do mundo real como forma de facilitar o aprendizado. É um método de ensino centro no aluno, onde o mesmo é envolvido em um processo ativo de resolução de problemas do mundo real, cujo objetivo é promover o pensamento crítico, desenvolver a habilidade de solucionar problemas e aprofundar o conhecimento acerca de um determinado tema (Barrows, 1986).

As principais características da aprendizagem baseada em problemas são:

- Problemas reais - O ABP baseia-se na ideia de que os estudantes aprendem melhor ao tentarem encontrar soluções para problemas que sejam relevantes e significativos para eles. Esses problemas geralmente são complexos, o que exige dos estudantes o uso do pensamento crítico e e habilidades que possam auxiliar na resolução do problema;
- Centrado ao estudante - Uma abordagem de aprendizagem centrada no estudante significa dizer que o mesmo deve ser um participante ativo no processo de aprendizagem, buscando novas fontes de conhecimento para resolver o problema proposto;
- Colaboração - Os estudantes devem trabalhar juntos em equipes para encontrar soluções para os problemas apresentados, criando um ambiente de aprendizado favorável e inclusivo, onde os estudantes se sentem encorajados a compartilhar suas ideias e trabalhar em equipe;
- Aprendizado profundo - O ABP permite que os estudantes apliquem e integrem os conhecimentos e habilidades de maneira significativa durante a resolução dos problemas. Isso auxilia a aprofundar o conhecimento em um determinado tema, o que torna o aprendizado mais duradouro.

Um estudo de Barrows (1986) mostrou que os alunos que aprenderam através do ABP tiveram notas significativamente mais altas em testes de pensamento crítico e resoluções de problemas em comparação aos alunos que foram ensinados através de abordagens mais tradicionais. Além da melhoria no desempenho, o ABP também aumenta a motivação e o interesse dos alunos acerca de um determinado tema (Boud & Feletti, 1991).

Esta abordagem será aplicada em um artefato que será composto por uma documentação que servirá como material de apoio e uma API vulnerável, que conterá todos os dez problemas de segurança propostos pelo OWASP API Security TOP 10. Através

do material de apoio desenvolvido, os alunos serão capazes de entender os principais conceitos e como os problemas propostos pelo OWASP API Security TOP 10 são explorados no dia a dia.

Após este primeiro contato, a API vulnerável servirá como um laboratório para que os alunos apliquem o conhecimento adquirido para implementarem as melhores práticas de segurança para mitigar os riscos identificados. Tal abordagem permitirá que os alunos apliquem o aprendizado teórico de segurança da informação em um contexto prático e relevante e que eles desenvolvam habilidades essenciais de análise, identificação de vulnerabilidades e aplicação de contramedidas, contribuindo para o desenvolvimento de APIs mais seguras.

2.2. Vulnerabilidades

Conforme descrito no livro Foundations of Information Security: A Straightforward Introduction (Andress, 2019) uma vulnerabilidade pode ser uma fraqueza ou uma brecha passível de exploração por um agente malicioso cujo objetivo é causar dano ou prejuízo a empresa. Geralmente, vulnerabilidades estão atreladas a sistemas operacionais de dispositivos ou aplicações que a empresa utiliza.

Para auxiliar desenvolvedores e entusiastas em segurança da informação, algumas organizações como por exemplo, a Open Web Application Security Project (OWASP) categorizaram vulnerabilidades comumente encontradas em diversos tipos de aplicação. No contexto de APIs, as vulnerabilidades foram classificadas da seguinte forma:

Algumas dessas vulnerabilidades abordam o mecanismo de gerenciamento de sessão das aplicações, por exemplo, o *Broken Object Level Authorization* - também conhecido como BOLA - e o *Broken Function Level Authorization*. O primeiro refere-se a capacidade de um usuário conseguir manipular, visualizar ou executar ações em nome de outros usuários da aplicação (OWASP, 2019). Apesar de ser um pouco semelhante ao BOLA, o *Broken Function Level Authorization* possui como principal característica a possibilidade de um usuário realizar uma ação restringida a um perfil de acesso maior que o seu, por exemplo, um usuário comum utilizar uma funcionalidade administrativa (Gunathunga, 2019).

Outros problemas derivam da ausência de sanitização dos dados enviados pelo usuário, sendo estes, *Injection & Mass Assignment*. Falhas de injeção são comumente encontradas em aplicações que realizam alguma consulta em uma base de dados, este tipo de problema possui como causa raiz a ausência das validações citadas e a utilização de métodos inerentemente inseguros para a construção de consultas (Moraes, 2020).

Já o *Mass Assignment* ocorre quando uma funcionalidade responsável por atualizar alguma informação utilizada pela aplicação não restringe quais atributos um usuário pode modificar (Li, 2021). Através deste comportamento um agente malicioso poderia, por exemplo, alterar o seu próprio seu saldo numa aplicação de finanças.

Broken User Authentication - Este problema de segurança aborda o mecanismo de autenticação das aplicações (OWASP, 2019), desde o momento em que o usuário insere a combinação de login/senha até o gerenciamento da sessão. Caso exista uma má implementação deste mecanismo, é possível utilizá-lo como vetor para realizar ataques de dicionário, visando o comprometimento de credenciais ou como vetor para adulteração

do *token* de sessão recebido, com intuito de obter sessões válidas de outros usuários.

Excessive Data Exposure - Implementações genéricas tendem a expor dados potencialmente sensíveis dos usuários (Barahona, 2022) da aplicação pois não avaliam o grau de confidencialidade de cada informação ou atribuem ao *client-side* a tarefa de ofuscar tais informações.

Lack of Resources and Rate Limiting - A ausência de mecanismos anti automação fornecem os vetores necessários para que atacantes explorem problemas de segurança existentes na aplicação de maneira automatizada. A exploração deste problema tem impactos diferentes a depender da regra de negócio da aplicação, entretanto, o impacto mais comumente encontrado é a indisponibilidade da aplicação ou o comprometimento de credenciais de usuários.

Security Misconfiguration - Configurações inadequadas de segurança podem ocorrer em toda cadeia de desenvolvimento, abrangendo o nível de código até os softwares utilizados. Mensagens de erros extremamente descritivas, cabeçalhos HTTP que revelam informações potencialmente sensíveis e ausência de *patches* são vetores utilizados por atacantes para comprometer aplicações.

Improper Assets Manager - Aplicações que estejam nas versões de desenvolvimento ou descontinuadas podem servir como vetor para a exploração de problemas de segurança corrigidos em versões estáveis.

Insufficient Logging & Monitoring - Esta vulnerabilidade ocorre quando a aplicação não fornece para o time de monitoramento e detecção as informações necessárias para uma eventual resposta a incidentes (Kiprin, 2021), no contexto de APIs, são exemplos deste tipo de informação: o recurso que está sendo requisitado, o *token* de sessão do requisitante, os parâmetros enviados no corpo da requisição ou na URL.

Uma vez que a aplicação não implemente mecanismos de *log* e monitoramento, a mesma permite com que um atacante consiga explorar vulnerabilidades existentes na aplicação sem ser detectado.

A próxima seção descreverá trabalhos análogos ao proposto neste artigo e que servirão como inspiração para o desenvolvimento do mesmo.

3. Trabalhos Relacionados

O trabalho “Learn security through insecurity” (Sonntag, 2013) apresenta uma proposta de ensino-aprendizado sobre segurança da informação em aplicações *web*, para tal, o autor utiliza o método de ensino-aprendizagem a aprendizagem combinada, para desenvolver uma aplicação *web* na linguagem Java, denominada SecuritySampleServer. Nesta abordagem, os alunos serão apresentados aos problemas comumente encontrados em aplicações *web*, onde irão aprender sobre os conceito de cada um deles e depois irão aplicar as correções necessárias para que estes não sejam explorados.

Apesar de possuir uma proposta muito semelhante com a apresentada neste trabalho, o artefato proposto por Sonntag diferencia-se negativamente em dois grandes pontos: O primeiro ponto decorre do fato de que as vulnerabilidades escolhidas por ele não foram baseadas em uma lista referendada pela comunidade de segurança da informação, como por exemplo, o OWASP TOP 10 *Web Application Security Risks*. Já o segundo ponto é

que durante o desenvolvimento do *SecuritySampleServer*, nenhum experimento fora realizado com o intuito de avaliar se o artefato proposto por ele de fato seria capaz de ensinar sobre segurança da informação em aplicações *web*.

O artigo "Development of Training System and Practice Contents for Cybersecurity Education" (Shin et al, 2019) apresenta um sistema denonimado CyExec, este artefato tem como objetivo servir como uma plataforma de treinamento para que pequenas empresas e instituições de ensino capacitem-se nos tópicos de segurança da informação ofensiva e defensiva, com foco no aprendizado em detecção e respostas à incidentes. Para tal, este artefato foi dividido em duas partes:

1. Exercícios básicos - Esta parte do artefato trás consigo uma trilha de estudo que contém os conceitos dos problemas de segurança que serão abordados. O conteúdo desta trilha é baseado nas vulnerabilidades propostas por um aplicação vulnerável amplamente utilizada no mercado de segurança da informação, conhecida como WebGoat;
2. Exercícios aplicados - Os exercícios práticos fornecem um laboratório interativo, onde são abordadas as perspectivas do atacante e do defensor simultaneamente. O objetivo final é fazer com que as instituições que utilizem este artefato desenvolvam, através dos exercícios, as competências necessárias para identificar e responder corretamente à incidentes de segurança da informação.

O CyExec diferencia-se da abordagem proposta neste trabalho, uma vez que seu principal objetivo é ensinar como detectar e responder a incidentes de segurança, enquanto o objetivo deste trabalho é ensinar sobre problemas de segurança em APIs utilizando o método de ensino-aprendizagem ABP. Além disso, o CyExec não foi desenvolvido com base em um método de ensino-aprendizagem, mas sim utilizando um modelo cujo objetivo é avaliar a capacidade de uma empresa na detecção de ataques. Por fim, é importante ressaltar que este artefato não passou por um processo de validação, ao contrário da proposta apresentada neste trabalho.

Existem algumas ferramentas mais famosas no mercado, entre elas, a Damn Vulnerable Web Application (DVWA) que foi desenvolvida pelo DVWA Team, cujo objetivo é ensinar sobre as vulnerabilidades comumente encontradas em aplicações *web*. Para tal, a aplicação possui cerca de dez problemas de segurança e cada um desses problemas possuem 3 níveis de dificuldade e uma versão corrigida, além disso, fornece para o usuário a opção de visualizar o código fonte de cada funcionalidade para entender o comportamento de cada uma delas.

A aplicação WebGoat desenvolvida pelo Open Web Application Security Project (OWASP) funciona de maneira análoga ao DVWA, também tendo como ênfase as vulnerabilidades comumente encontradas em aplicações web, dividas em dez tópicos ou lições sobre cada problema proposto pelo OWASP TOP 10 *Web Application Security Risks* e contém cerca de dez desafios por funcionalidade.

Apesar de possuírem semelhanças com a proposta deste trabalho, os projetos anteriores abordam, em sua maioria, vulnerabilidades comumente encontradas em aplicações web e por este motivo, não abrangem todo o OWASP API Secyurity TOP 10, além disso, estes projetos não utilizam o método de ensino-aprendizagem aprendizagem baseada em problemas, que é o foco deste trabalho. Entretanto, os trabalhos citados nesta seção servirão de inspiração para o desenvolvimento da abordagem proposta neste trabalho, cujo

objetivo é fazer com que desenvolvedores e entusiastas em segurança da informação adquiram as habilidades e competências necessárias para desenvolver APIs seguras e resilientes.

4. Metodologia

Esta seção descreverá a metodologia utilizada para o desenvolvimento do artefato, também definirá a estrutura do experimento a ser realizado com estudantes de graduação e elucidará como o Aprendizado baseado em problemas (ABP) foi abordado pelo artefato proposto neste projeto.

Para melhor entendimento, o desenvolvimento deste projeto foi dividido em oito etapas, sendo elas:

1. Revisão bibliográfica;
2. Definição do método de ensino-aprendizagem;
3. Estudo do Design Science Research;
4. Construção do artefato: API vulnerável e documentação;
5. Definição e condução do experimento;
    - Execução do experimento;
    - Análise dos resultados obtidos;
    - Implementação de melhorias.

Portanto, a primeira etapa do desenvolvimento deste projeto consistiu na revisão bibliográfica sobre os temas ensino-aprendizagem e problemas de segurança da informação em APIs. Esta revisão teve como objetivo definir qual método de aprendizagem será abordado pelo artefato proposto e como abordar os problemas de segurança propostos dentro do contexto ensino-aprendizagem.

Na segunda etapa, o método de ensino escolhido foi a aprendizagem através de problemas (ABP). Desta maneira, para abordar o ABP, cada classe de vulnerabilidade proposta pelo OWASP API Security TOP 10 será transformada numa funcionalidade de uma aplicação financeira que estará disponível a partir da API desenvolvida neste projeto. Cada funcionalidade representará um problema no qual um banco fictício está tentando solucionar.

O OWASP API Security TOP 10 foi escolhido como base para este projeto pelos motivos a seguir:

- Relevância - O OWASP API Security TOP 10 trás os problemas de segurança mais críticos e comumente encontrados em APIs, o que o torna uma referência relevante e útil para pesquisadores e praticantes.
- Credibilidade - O OWASP é uma organização mundialmente conhecida e respeitada pelos pesquisadores em segurança da informação, também é amplamente aceita como uma organização de alta credibilidade e confiabilidade neste ramo. A utilização dos materiais do OWASP como base para estudos aumenta as chances de aceitação pela comunidade acadêmica e pelos profissionais do mercado de segurança da informação.
- Praticidade - O OWASP API Security TOP 10 é uma guia prático que pode ser utilizado durante o processo de *design* e desenvolvimento de um software, trazendo consigo exemplos e recomendações práticas sobre os problemas em questão.

- Aceitação - O OWASP TOP 10 API é amplamente aceito como um padrão na indústria de segurança da informação. Ele é utilizado por desenvolvedores, profissionais de segurança e organizações.
- Comparabilidade - Utilizando o OWASP API Security TOP 10 como base para a pesquisa, o estudo pode ser comparado com outros projetos que utilizam o mesmo padrão, o que facilita a compreensão dos resultados encontrados pela pesquisa.

A terceira etapa deste trabalho consistiu no estudo sobre a metodologia *Design Science Research* (DSR) para o desenvolvimento do artefato proposto neste projeto. Conforme descrito por Henvel (2004) o *Design Science Research* tem como propósito principal o ganho de conhecimento sobre um determinado tema a partir da construção de um artefato.

Para alcançar tal objetivo, Henvel definiu sete diretrizes que devem ser seguidas para a construção de um artefato a partir do DSR, sendo elas:

1. Design como um artefato - O objetivo final do DSR deve ser a produção de um artefato, seja ele um modelo ou um *software*;
2. Relevância do problema - O problema abordado pelo artefato deve ser um problema do cotidiano e que possua grande relevância;
3. Avaliação do Design - A utilidade, qualidade e eficácia do artefato desenvolvido deve ser classificada através de métodos rigorosos de avaliação;
4. Contribuição em pesquisa - O resultado obtido deve contribuir de maneira clara e objetiva para a área de conhecimento escolhida;
5. Rigor em pesquisa - A credibilidade do DSR está atrelada a utilização de métodos rigorosos na construção e na qualificação do artefato;
6. Design como um processo de pesquisa - O DSR visa garantir que os meios disponíveis foram utilizados de maneira eficiente e de fato resolvem o problema proposto;
7. Pesquisa como forma de comunicação - A proposta do artefato deve ser entendida por pessoas técnicas e pessoas de gestão.

A partir do estudo realizado anteriormente, a quarta etapa consistiu no desenvolvimento da API que irá compor o artefato. Esta API representará um banco digital, onde cada vulnerabilidade descrita pelo OWASP TOP 10 API Security será abordada, através das funcionalidades da aplicação, como um problema que o banco em questão está tentando solucionar.

A API foi desenvolvida utilizando a linguagem de programação Node.js com o apoio do *framework Express*. A arquitetura escolhida para ela foi a *Model-View-Controller (MVC)* em conjunto com o padrão *Representational State Transfer* (REST).

A arquitetura MVC foi escolhida pois separa com nitidez as responsabilidades atribuídas a cada um dos componentes da aplicação, o que facilita no entendimento, na resolução de problemas e manutenção do código fonte. Por sua vez, o padrão REST define a utilização de URLs amigáveis e operações uniformes, o que simplifica o processo de consumo da API desenvolvida.

O banco de dados escolhido foi um banco de dados não relacional, neste caso, o MongoDB Server na versão 6.0.1. Informações detalhadas sobre a estrutura da API e do banco de dados podem ser obtidas através da página API Reference (Henrique, 2023).

Ainda na quarta etapa, com o intuito de auxiliar os estudantes no aprendizado em segurança da informação através do artefato proposto, fora desenvolvido uma aplicação *web* cujo objetivo é servir como material de apoio para os estudantes que irão participar dos ciclos de experimentação. O material de apoio foi estruturado da seguinte maneira:

- Introdução - Esta seção contém uma breve descrição sobre a proposta desta API;
- Setup - Esta seção contém as instruções necessárias para a configuração e utilização da API;
- API Reference - Esta seção contém todos os detalhes sobre como a API está estruturada, fornecendo ao usuário uma descrição detalhada sobre a sua arquitetura e como seus componentes estão relacionados;
- Materiais de apoio - Esta seção contém dez subseções onde cada uma representará um problema de segurança abordado pela API. Cada subseção trará consigo uma descrição do problema, como explorá-lo e como corrigi-lo. Todo conteúdo da seção materiais de apoio foi desenvolvido com base em *papers* e livros referendados pela comunidade de segurança da informação.

Este material de apoio foi desenvolvido utilizando o *framework* Gitbook. O conteúdo desenvolvido encontra-se disponível na seção Material de apoio (Henrique, 2023).

Na quinta etapa, com o intuito de entender se de fato o artefato proposto auxilia no processo de aprendizagem de segurança da informação em APIs, um experimento foi realizado com 10 estudantes, majoritariamente do curso de Sistemas de Informação da Universidade Federal Rural de Pernambuco (UFRPE). O experimento foi estruturado da seguinte forma:

1. Cada estudante receberá o código fonte da aplicação juntamente com o material de apoio;
2. O experimento será realizado remotamente, os estudantes entrarão num *meet* com o facilitador e terão 2 horas para ler o material de apoio, entender os problemas de segurança abordados e aplicar as correções necessárias;
3. 5 problemas de serão abordados no experimento, os problemas foram previamente escolhidos pelo facilitador, neste caso, o primeiro autor deste artigo. Os problemas escolhidos para as rodadas do experimento foram os seguintes:
    - Improper Assets Management;
    - Broken Object Level Authorization;
    - Broken User Authentication;
    - Mass Assignment;
    - Insufficient Logging & Monitoring.
4. Durante o experimento, os alunos podem tirar dúvidas com o facilitador conceituais sobre os problemas de segurança apresentados;
5. Após o fim do tempo determinado, cada estudante responderá um questionário contendo cerca de 15 perguntas — disponíveis no apêndice A - Questionário — para avaliar os seguintes tópicos:
    - A qualidade do conteúdo do material de apoio de cada problema de segurança abordado durante o experimento;
    - Como o estudante avalia o processo de aprendizagem, em outras palavras, se ele considera que aprendeu sobre segurança da informação em APIs;

- O nível de dificuldade dos problemas propostos pela API;
- Verificação de aprendizagem;
- Avaliar se o interesse sobre segurança da informação aumentou após o contato com o artefato desenvolvido;
- Feedbacks gerais acerca do artefato desenvolvido.

Existem alguns aspectos que podem ajudar a identificar as dificuldades que os participantes enfrentam, incluindo:

- Conhecimento prévio limitado - Os alunos envolvidos no experimento podem ter dificuldades se não possuírem conhecimento prévio básico sobre alguns conceitos que antecedem os problemas de segurança abordados;
- Falta de experiência prática - Os alunos que tiverem pouca experiência prática em desenvolvimento ou em resolução de problemas em APIs poderão enfrentar dificuldades no momento em que irão implementar as medidas de segurança adequadas para tornar a API segura;
- Má gestão do tempo - O experimento possui um *time-box* de 2 horas definido. Caso os alunos não gerenciem adequadamente quanto tempo irão utilizar para assimilar o conteúdo e quanto tempo irão utilizar para solucionar os problemas abordados, existe a possibilidade de que eles não consigam resolver todos os itens propostos.

Foi delineado que o estudante necessita ter os seguintes pré-requisitos para participar do experimento:

- Possuir conhecimento básico em sistemas operacionais Linux ou Windows;
- Possuir experiência no desenvolvimento e no consumo de APIs REST;
- Possuir conhecimento intermediário na linguagem de programação JavaScript;
- Possuir conhecimento básico sobre o protocolo HTTP e suas aplicações;
- Possuir conhecimento intermediário em banco de dados;
- Possuir conhecimento básico sobre os conceitos de autorização e autenticação.

Após o fim do experimento, o estudante deve enviar ao facilitador o código fonte da aplicação, para que seja possível avaliar se as correções adotadas pelo estudante de fato resolveram o problema de segurança em questão. O experimento foi dividido em três rodadas, a primeira foi realizada com três estudantes, a segunda com quatro estudantes e a última com três estudantes.

Os *feedbacks* e resultados obtidos nas rodadas de experimentação serão analisados para entender se o artefato proposto cumpriu o seu propósito de forma satisfatória e quais melhorias devem ser aplicadas. Uma vez que as melhorias indicas estejam implementadas, novas rodadas de experimentação serão realizadas para manter um ciclo de melhoria contínua do artefato.

Ademais, a metodologia utilizada no desenvolvimento deste trabalho possui algumas limitações, entre elas:

- Subjetividade nas avaliações - A avaliação dos estudantes por meio de um questionário pode ser subjetiva e variar entre os participantes. As respostas podem ser influenciadas por percepções individuais, o que pode dificultar uma avaliação objetiva e precisa do processo de aprendizagem;

- Limitações na verificação de aprendizagem - A verificação da aprendizagem por meio do questionário pode não ser um indicador completo e abrangente das habilidades adquiridas. Outras formas de avaliação podem fornecer uma visão mais completa do conhecimento e habilidades dos participantes;
- Restrições acerca do tempo - O experimento foi realizado em um período de 2 horas, o que pode ser considerado um fator limitante para a resolução completa dos problemas e aprofundamento nos conceitos de segurança. O tempo restrito pode afetar a compreensão e aplicação das correções necessárias;
- Limitações na comunicação - Embora os estudantes tenham a oportunidade de tirar dúvidas conceituais com o facilitador, a interação remota pode não ser tão efetiva quanto a comunicação presencial. A falta de interação face a face pode limitar a compreensão e a clareza nas explicações;
- Limite na quantidade de participantes - Para garantir que o facilitador consiga auxiliar todos os participantes do experimento, o número máximo de participantes por rodada é de 5 o que torna o processo de validação do artefato moroso.

É importante considerar essas limitações ao interpretar os resultados do experimento e ao analisar a eficácia do artefato desenvolvido, buscando formas de mitigar essas limitações e aprimorar a abordagem de ensino-aprendizagem.

5. Resultados

Esta seção descreverá os resultados obtidos em cada rodada do experimento, trazendo consigo o *feedback* dos alunos sobre cada uma das perguntas do questionário, o resultado da verificação de aprendizagem realizada pelo autor ao final de cada rodada e as melhorias implementadas no artefato após recebimento dos *feedbacks* de cada um dos alunos.

5.1. Primeira rodada do experimento

O principal objetivo da primeira rodada do experimento era receber um prognóstico inicial sobre alguns aspectos da API e da documentação dos cinco problemas de segurança que fizeram parte do experimento. Para tal, os alunos avaliaram em uma escala de 1 a 5 o quão útil foi o material de apoio de cada problema de segurança no aprendizado deles durante o experimento, as notas atribuídas por eles podem ser observadas na Tabela 1 a seguir:

Aluno	Improper Assets Management	Mass Assignment	Insufficient Logging & Monitoring	Broken Object Level Authorization	Broken User Authentication
VG	5	5	4	5	5
HC	5	4	5	5	5
TG	5	5	5	5	5

Tabela 1. *Feedbacks* da primeira rodada sobre os materiais de apoio.

Como podemos observar na Tabela 1, os *feedbacks* recebidos acerca do material de apoio desenvolvido foram positivos. A próxima pergunta do questionário tem como objetivo saber se os alunos corrigiram todos os problemas de segurança propostos, na rodada em questão, todos os alunos responderam que cumpriram esta etapa com êxito.

A partir desta informação o facilitador, neste caso, o autor deste artigo, avaliou cada código fonte submetido ao formulário com o intuito de verificar se os alunos de fato corrigiram com êxito os problemas de segurança abordados no experimento. A Tabela 2 a seguir ilustra o resultado desta verificação de aprendizagem:

Aluno	Improper Assets Management	Mass Assignment	Insufficient Logging & Monitoring	Broken Object Level Authorization	Broken User Authentication
VG	Corrigido	Corrigido	Corrigido	Corrigido	Corrigido
HC	Corrigido	Corrigido	Corrigido	Corrigido	Corrigido
TG	Corrigido	Corrigido	Corrigido	Corrigido	Corrigido

Tabela 2. Resultado da verificação de aprendizagem da primeira rodada.

Nesta rodada, os estudantes TG e VG resolveram todos os problemas de segurança em pouco mais de uma hora e o aluno HC utilizou todo o *time-box* do experimento. Também foi solicitado que os alunos avaliassem em uma escala de 1 a 5 a usabilidade da API e o quão desafiadores eram os problemas de segurança abordados nesta rodada do experimento. Conforme ilustrado na Tabela 3 abaixo, todos avaliaram como nota 4 ambos os tópicos.

Aluno	Em uma escala de 1 a 5, o quão fácil foi utilizar a API?	Em uma escala de 1 a 5, o quão desafiadores eram os problemas propostos pela API?
VG	4	4
HC	4	4
TG	4	4

Tabela 3. Avaliação sobre usabilidade da API e grau de dificuldade dos desafios.

Por último, cada aluno realizou uma autoavaliação, em uma escala de 1 a 5, no tocante ao quanto aprendeu sobre segurança de informação e se o interesse dele em tópicos de segurança da informação aumentou após o contato com a API desenvolvida neste projeto. A resposta dos alunos para cada uma destas perguntas estão representadas na Tabela 4 a seguir:

Aluno	Em uma escala de 1 a 5, o quanto você aprendeu sobre segurança em APIs?	Após o contato com esta API, o seu interesse por segurança da informação aumentou?
VG	5	Sim
HC	5	Sim
TG	5	Sim

Tabela 4. Autoavaliação sobre aprendizado e aumento de interesse em segurança da informação.

Apesar de grande parte dos resultados obtidos nesta rodada serem positivos, a partir dos *feedbacks* dos participantes, foram identificadas melhorias a serem aplicadas tanto a API vulnerável quanto ao material de apoio. A Tabela 5 representa os *feedbacks* em questão:

Feedback	Melhorias a serem implementadas
”A documentação do gitbook está com erro nas rostas. Por exemplo, a rota correta é /api/users/login e não /login. Além disso, estão com barras duplicadas, ficando //login.”	1. Ajustar as seções do material de apoio que fazem referência a uma determinada rota da aplicação.
”Sempre que houver um print ou snippet do código, falar o nome da classe para facilitar a visualização.”	2. Alterar a documentação para deixar mais explícito ao usuário quais classes/arquivos devem ser modificados para a resolução do problema.
”Criar um docker-compose para facilitar no setup inicial.” ”Poderia adicionar um script docker-compose para facilitar a execução do projeto, aumentando a rapidez na configuração, sem a necessidade de instalar as ferramentas na máquina local.”	3. Criar um script docker para facilitar o setup da aplicação.

Tabela 5. *Feedback* de melhorias que devem ser implementadas após o término da primeira rodada.

Além dos *feedbacks* anteriores, foi possível observar que todos utilizaram a ferramenta Postman para consumir a API desenvolvida, portanto, com o intuito de facilitar a utilização da API durante o experimento, fora desenvolvida uma *collection* para a ferramenta em questão, contendo todos os *endpoints* e os dados necessários para consumir a API desenvolvida neste projeto.

Após a análise destes *feedbacks*, todas as sugestões de melhorias indicadas pelos estudantes foram implementadas no artefato e uma nova rodada do experimento fora realizada.

5.2. Segunda rodada do experimento

A segunda rodada tinha como principal objetivo a verificação do impacto das melhorias implementadas a partir dos *feedbacks* recebidos na rodada anterior e também se a avaliação de cada aluno sobre os tópicos abordados no questionário se manteriam semelhantes ao da rodada anterior.

A avaliação dos participantes desta rodada acerca do conteúdo do material de apoio não variou significativamente, tendo resultado muito semelhante ao obtido na rodada anterior. As notas atribuídas pelos alunos em cada tópico está descrita na Tabela 6 a seguir:

Aluno	Improper Assets Management	Mass Assignment	Insufficient Logging & Monitoring	Broken Object Level Authorization	Broken User Authentication
VM	5	4	5	5	4
BG	4	5	5	5	4
AL	5	5	5	5	5
AL	5	5	5	4	5

Tabela 6. *Feedbacks* dos alunos da segunda rodada sobre o material de apoio.

Entretanto, nesta rodada, três dos quatro alunos que participaram do experimento não conseguiram resolver todos os desafios propostos. Ao serem questionados qual(is) problema(s) de segurança não foi corrigido com êxito, todos os três informaram que não conseguiram aplicar uma solução para o problema *Broken Object Level Authorization*. Ressaltamos o fato de que todos os alunos que não conseguiram solucionar o problema de segurança *Broken Object Level Authorization* utilizaram todo o *time-box* definido para o experimento. Apenas o aluno AL forneceu *feedback* acerca da ausência de tempo para a resolução dos problemas, o mesmo argumentou que acabou perdendo tempo resolvendo com outras coisas que julgou relevante e que se não fosse por isto, teria resolvido o problema em questão.

O resultado da verificação de aprendizagem desta rodada está descrito na Tabela 7 a seguir:

Aluno	Improper Assets Management	Mass Assignment	Insufficient Logging & Monitoring	Broken Object Level Authorization	Broken User Authentication
VM	Corrigido	Corrigido	Corrigido	Corrigido	Corrigido
BG	Corrigido	Corrigido	Corrigido	Não corrigido	Corrigido
AL	Corrigido	Corrigido	Corrigido	Não corrigido	Corrigido
HM	Corrigido	Corrigido	Corrigido	Não corrigido	Corrigido

Tabela 7. Resultado da verificação de aprendizagem da segunda rodada.

Foi possível constatar uma melhoria sutil na avaliação dos alunos acerca da usabilidade da API após as melhorias implementadas a partir dos *feedbacks* recebidos na rodada anterior. Nesta rodada, dois dos quatro alunos forneceram a nota máxima (5) neste quesito, entretanto, um dos participantes avaliou como 3 a usabilidade da API.

Na avaliação dos alunos acerca do grau de dificuldade dos problemas de segurança apresentados, dois dos alunos atribuíram a nota 3 a este tópico, onde um deles informou no *feedback* que já os conhecia, entretanto, os outros dois alunos restantes atribuíram a nota máxima (5) a este tópico. As notas atribuídas por cada um dos alunos aos tópicos de usabilidade e grau de dificuldade dos problemas abordados, estão representadas na Tabela 8 abaixo:

Aluno	Em uma escala de 1 a 5, o quão fácil foi utilizar a API?	Em uma escala de 1 a 5, o quão desafiadores eram os problemas propostos pela API?
VM	4	3
BG	3	5
AL	5	4
HM	5	5

Tabela 8. Avaliação sobre usabilidade da API e grau de dificuldade dos desafios.

No quesito de autoavaliação de aprendizagem, os resultados obtidos nesta rodada foram menores que os da anterior, tendo dois dos quatro alunos avaliando este quesito com a nota três e somente um aluno atribuindo a nota máxima. Ademais, apenas um aluno informou que o seu interesse sobre segurança da informação não aumentou após o contato com o artefato desenvolvido neste projeto e durante o experimento, o que é interessante, pois, apesar de grande parte deles não terem corrigido todos os problemas, o artefato foi capaz de despertar interesse sobre o tema.

A Tabela 9 seguir ilustra a resposta de cada aluno sobre os tópicos citados anteriormente:

Aluno	Em uma escala de 1 a 5, o quanto você aprendeu sobre segurança em APIs?	Após o contato com esta API, o seu interesse por segurança da informação aumentou?
VM	3	Sim
BG	3	Sim
AL	4	Não
HM	5	Sim

Tabela 9. Autoavaliação dos alunos da segunda rodada sobre aprendizado e aumento de interesse em segurança da informação.

As melhorias identificadas após o final da segunda rodada estão descritas na Tabela 10 a seguir:

Feedback	Melhorias a serem implementadas
”Existe um erro na documentação do Improper Assets Management, o arquivo apontado para correção é o personRoute.js, entretanto, o arquivo correto é o personController.js.”	1. Corrigir a documentação do material de apoio em questão, informando o arquivo correto.
”Algumas rotas do postman estão com o cabeçalho de autorização informado incorretamente.”	2. Ajustar as rotas do postman onde o cabeçalho X-Acces-Token está informado erroneamente.
”É necessário alterar a ordem dos problemas na documentação, pois, ao utilizar o botão de next, sou redirecionado para um problema de segurança que não faz parte do escopo do experimento.”	3. Alterar a ordem da documentação no Gitbook, para que não haja desvios no fluxo do experimento.

Tabela 10. Segunda rodada de*feedbacks* de melhorias.

Durante esta rodada, um dos alunos cogitou desistir do experimento, argumentando que não possuía proficiência suficiente na linguagem de programação JavaScript e que possuía uma expectativa de que o experimento funcionasse de maneira análoga à um *workshop* mas que sentiu que estava numa prova técnica.

O material de apoio do conteúdo *Broken Object Level Authorization* deverá ser incrementado, pois, apesar de terem atribuído a nota máxima a este tópico, três dos quatro estudantes não conseguiram resolver este problema de segurança, o que indica que este conteúdo precisa ser remodelado para que haja um melhor entendimento de como corrigí-lo.

Também foi possível constatar que nenhum dos envolvidos nas duas rodadas optaram por solucionar o problema de segurança *Broken User Authentication* através da substituição do algoritmo criptográfico responsável pela assinatura do *token* de sessão gerado pela aplicação, portanto, este texto será removido do material de apoio em questão.

5.3. Terceira rodada do experimento

Após as divergências apresentadas na rodada anterior, a terceira rodada teve como objetivo validar se as melhorias implementadas seriam capazes de minimizar a quantidade de alunos que não conseguiram resolver todos os problemas de segurança e se os resultados obtidos serão semelhantes aos da primeira rodada, onde o tópico principal — aprendizado através do artefato proposto — foi avaliado com nota máxima (5) pelos estudantes.

Nesta rodada, os estudantes atribuíram a nota máxima a todos os tópicos referentes ao material de apoio, conforme demonstrado na Tabela 11 a seguir:

Aluno	Improper Assets Management	Mass Assignment	Insufficient Logging & Monitoring	Broken Object Level Authorization	Broken User Authentication
MD	5	5	5	5	5
DA	5	5	5	5	5
RN	5	5	5	5	5

Tabela 11. *Feedbacks* dos alunos da terceira rodada sobre os materiais de apoio.

Nenhum dos estudantes que participaram desta rodada do experimento utilizaram o *timebox* completo. Entretanto, um estudante conseguiu resolver o *Broken Object Level Authorization*, o mesmo argumentou durante o experimento que não foi capaz de corrigir o problema em questão, pois, o artefato desenvolvido apresentou problemas para ser executado dentro de um Subsistema Windows para Linux (WSL) somente nesta funcionalidade em específico.

O resultado da verificação de aprendizagem está representado na Tabela 12 ilustrada a seguir:

Aluno	Improper Assets Management	Mass Assignment	Insufficient Logging & Monitoring	Broken Object Level Authorization	Broken User Authentication
MD	Corrigido	Corrigido	Corrigido	Corrigido	Corrigido
DA	Corrigido	Corrigido	Corrigido	Não corrigido	Corrigido
RN	Corrigido	Corrigido	Corrigido	Corrigido	Corrigido

Tabela 12. Resultado da verificação de aprendizagem da terceira rodada.

No tópico de usabilidade, apenas um estudante atribuiu a nota máxima (5), os outros dois estudantes atribuíram a nota 4 a este tópico, resultado semelhante ao da primeira rodada. Já no quesito grau de dificuldade dos problemas de segurança abordados, dois dos três estudantes atribuíram a nota máxima (5) a este tópico e um estudante atribuiu a nota 4, o que representa uma melhora sutil em comparação ao resultado da segunda rodada. A Tabela 13 a seguir ilustra os resultados em questão:

Aluno	Em uma escala de 1 a 5, o quão fácil foi utilizar a API?	Em uma escala de 1 a 5, o quão desafiadores eram os problemas propostos pela API?
MD	5	4
DA	4	4
RN	4	4

Tabela 13. Avaliação dos alunos da terceira rodada sobre usabilidade da API e grau de dificuldade dos desafios.

O objetivo de fazer com que a maioria dos alunos voltassem a atribuir a nota máxima ao tópico de autoavaliação de aprendizagem foi concluído com êxito, nesta rodada, dois dos três alunos participantes atribuíram a nota máxima (5) a este tópico. No tocante ao aumento de interesse após o contato com a API desenvolvida neste trabalho, todos os alunos da terceira rodada atribuíram a nota máxima (5) a este tópico. A Tabela 14 trás a resposta de cada estudante acerca dos tópicos citados anteriormente:

Aluno	Em uma escala de 1 a 5, o quanto você aprendeu sobre segurança em APIs?	Após o contato com esta API, o seu interesse por segurança da informação aumentou?
MD	4	Sim
DA	5	Sim
RN	5	Sim

Tabela 14. Autoavaliação dos alunos da terceira rodada sobre aprendizado e aumento de interesse em segurança da informação.

A Tabela 15 representa o *feedback* dos estudantes acerca de melhorias que devem ser implementadas no artefato:

Feedback	Melhorias a serem implementadas
”Ao executar a aplicação no Windows Subsystem Linux (WSL) A rota para a funcionalidade que aborda o problema do Broken Object Level Authorization não funcionou corretamente.”	1. Investigar e corrigir o bug em questão, para que a API funcione corretamente em instâncias WSL.
”Existe um problema de dessincronização na funcionalidade de Mass Assignment, apesar de atualizar o objeto corretamente, o MongoDB trás por padrão a instância anterior, o que faz com que seja necessário utilizar a funcionalidade de visualizar saldo para verificar as alterações em tempo real.”	2. Ajustar o método findOne da funcionalidade em questão, adicionando o operador new à variável accountDisabled.

Tabela 15. *Feedbacks* de melhorias que devem ser implementadas após o término da terceira rodada.

6. Conclusão

O objetivo principal deste trabalho consiste no desenvolvimento de uma nova abordagem para ensinar sobre segurança da informação em APIs, utilizando o método de ensino-aprendizagem a aprendizagem baseada em problemas. Para alcançar tal objetivo, fora desenvolvida um artefato composto por uma API vulnerável à todos os problemas propostos pelo OWASP TOP 10 API Security e uma documentação que serviria como material de apoio e conseguiriam aprender sobre os problemas de segurança e aplicar as correções necessárias.

Para a construção do artefato, fora adotada uma metodologia de desenvolvimento de artefato conhecida como *Design Science Research* (DSR) em conjunto com o método de ensino-aprendizagem citado anteriormente. A união destes fatores teve como objetivo principal avaliar se de fato os estudantes conseguiriam aprender sobre segurança da informação em APIs através da abordagem proposta.

Analisando os resultados obtidos de uma maneira geral, 6 dos 10 estudantes que participaram dos experimentos atribuíram nota máxima (5) e 2 dos 10 atribuíram a nota 4 ao tópico que verifica se eles conseguiram aprender sobre segurança da informação através da abordagem proposta, o que indica que em um primeiro momento, o propósito deste trabalho foi cumprido.

Além disso, foi possível constatar a importância do *Design Science Research* (DSR) no processo de melhoria contínua do artefato desenvolvido, ao todo foram implementadas cerca de dez melhorias a partir dos *feedbacks* obtidos e especialmente na rodada 3, foi possível constatar que as melhorias trouxeram resultados positivos no que tange ao aprendizado dos estudantes.

Também devemos destacar o sucesso na utilização do aprendizado baseado em problemas (ABP) na construção do material de apoio, pois, conforme demonstrado no resultado obtido nas rodadas realizadas, na maioria das vezes os alunos atribuíram a nota máxima ao conteúdo desenvolvido. 9 dos 10 alunos avaliaram com nota máxima o material de apoio dos problemas *Broken Object Level Authorization*, *Improper Assets Management* e *Insufficient Logging & Monitoring*. 8 dos 10 alunos avaliaram com nota máxima o material de apoio dos problemas *Mass Assignment* e *Broken User Authentication*.

Apesar de grande parte dos *feedbacks* recebidos sobre a abordagem proposta serem positivos, é muito cedo para afirmar que ela possui, de fato, a capacidade de ensinar sobre segurança da informação, pois, é necessário que mais iterações sejam realizadas, abordando todos os problemas propostos, com o maior número de pessoas possível. Ademais, uma investigação mais profunda sobre o problema de segurança *Broken Object Level Authorization*, pois, 4 dos 10 alunos não conseguiram resolvê-lo.

Apesar de grande parte dos *feedbacks* recebidos sobre a metodologia utilizada serem positivos, é muito cedo para afirmar que ela possui, de fato, a capacidade de ensinar sobre segurança da informação, pois, apenas cinco problemas de segurança foram avaliados e houveram apenas três iterações, totalizando um número de dez estudantes. Para que esta metodologia seja considerada válida, faz-se necessário que todos os problemas de segurança sejam validados por, no mínimo, 50 pessoas e que os resultados acerca da autoavaliação sobre aprendizado sejam, em sua maioria, a nota máxima (5) e que a grande maioria dos estudantes consigam resolver todos os problemas de segurança propostos.

Por fim, é possível concluir de que apesar da abordagem de ensino não estar devidamente validada, os resultados obtidos demonstram que a metodologia utilizada para o desenvolvimento deste trabalho são válidas.

Uma das próximas etapas para este trabalho consiste na realização do experimento abordando problemas de seguranças distintos, pois, apenas cinco dos problemas de segurança propostos pelo artefato foram avaliados durante o desenvolvimento deste trabalho. Outro trabalho futuro pode ser a realização de um estudo cujo objetivo é verificar se uma mudança na combinação de problemas de segurança abordados no experimento possui influência no desempenho dos alunos no tópico de verificação de aprendizagem.

Além disso, sugere-se que os problemas de segurança propostos pelo artefato sejam atualizados conforme novas edições do OWASP API Security TOP 10 sejam publicadas, visando garantir que o artefato aborde problemas de segurança atuais e relevantes para a comunidade de segurança da informação.

Referências

BARROWS, H. S. (1986). A taxonomy of problem-based learning methods. Medical Education, 20(6), 481-486.

BOUD, D., & FELETTI, G. (1991). The challenge of problem-based learning. London: Kogan Page.

SONNTAG, M. "Learning security through insecurity,"2013 Second International Conference on E-Learning and E-Technologies in Education (ICEEE), 2013, pp. 143-148, doi: 10.1109/ICeLeTE.2013.6644363.

SHIN ET AL, S. "Development of Training System and Practice Contents for Cybersecurity Education,"2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI), 2019, pp. 172-177, doi: 10.1109/IIAI-AAI.2019.00043.

LI, Vickie. API Security 101: Mass Assignment. Disponível em https://blog.shiftleft.io/api-security-101-mass-assignment-31060f7ee80e. Acesso em 25 de março de 2022.

KIPRIN, Borislav. Comprehensive Guide to Insufficient Logging & Monitoring and How to Prevent It. Disponível em https://crashtest-security.com/insufficient-logging-monitoring-guide/. Acesso em 21 de abril de 2022.

PAXTON-FEAR, Katie. Improper Assets Management. Disponível em https://www.traceable.ai/owasp-api/improper-assets-management. Acesso em 21 de abril de 2022.

MORAES, Vinícius. Há muito tempo, numa web distante, nascia o SQL Injection. Disponível em https://sidechannel.blog/ha-muito-tempo-numa-web-distante-nascia-o-sql-injection/. Acesso em 20 de abril de 2022.

OWASP. API7:2019 Security Misconfiguration. Disponível em https://github.com/OWASP/API-Security/blob/master/2019/en/src/0xa7-security-misconfiguration.md. Acesso em 20 de abril de 2022.

GUNATHUNGA, Sagara. API Security: How to avoid Broken Object Level Authorization & Broken Function Level Authorization. Disponível em https://sagarag.medium.com/api-security-how-to-avoid-broken-object-level-authorization-broken-function-level-authorization-f93b9b5ac333. Acesso em 20 de abril de 2022.

PAXTON-FEAR, Katie. Lack of Resources & Rate Limiting. Disponível em https://www.traceable.ai/owasp-api/lack-of-resources-rate-limiting. Acesso em 30 de março de 2022.

BARAHONA, Dan. Drilling Down Into Excessive Data Exposure: How to Protect Your APIs Sensitive Data. Disponível em https://www.apisec.ai/blog/excessive-data-exposure. Acesso em 30 de março de 2022.

OWASP. API1:2019 Broken Object Level Authorization. Disponível em https://github.com/OWASP/API-Security/blob/master/2019/en/src/0xa1-broken-object-level-authorization.md. Acesso em 20 de abril de 2022.

OWASP. API2:2019 Broken User Authentication. Disponível em

https://github.com/OWASP/API-Security/blob/master/2019/en/src/0xa2-broken-user-authentication.md. Acesso em 20 de abril de 2022.

LABS, Salt. Salt Security State of API Security Report Reveals API Attacks Increased 681% in the Last 12 Months. Disponível em: https://salt.security/press-releases/salt-security-state-of-api-security-report-reveals-api-attacks-increased-681-in-the-last-12-months. Acesso em 22 de abril de 2022.

ANDRESS, Jason. Foundations of information security: A Straightforward Introduction. No Starch Press. 2019.

HENRIQUE, Ricardo. Material de apoio. Disponível em https://tcc-bsi-ufpre.gitbook.io/tcc-api-reference/materiais-de-apoio. Acesso em 17 de abril de 2023.

HENRIQUE, Ricardo. API Reference. Disponível em https://tcc-bsi-ufpre.gitbook.io/tcc-api-reference/api-reference/api-reference. Acesso em 17 de abril de 2023.

OWASP. API Security TOP 10 2019. Disponível em https://owasp.org/www-project-api-security/. Acesso em 26 de novembro de 2022.

Apêndice A - Questionário

As 15 perguntas contidas no questionário de avaliação do artefato foram as seguintes:

1. Qual o seu nome?
2. Você possuía interesse em aprender mais sobre segurança da informação?
3. Avalie o material de apoio sobre o problema de segurança Broken Object Level Authorization;
4. Avalie o material de apoio sobre o problema de segurança Broken User Authentication;
5. Avalie o material de apoio sobre o problema de segurança Mass Assignment;
6. Avalie o material de apoio sobre o problema de segurança Improper Assets Management;
7. Avalie o material de apoio sobre o problema de segurança Insufficient Logging & Monitoring;
8. Você conseguiu corrigir todos os problemas propostos pelo experimento?
9. Caso a resposta da pergunta anterior tenha sido não, especifique os problemas que você não corrigiu;
10. Em uma escala de 1 a 5, o quão fácil foi utilizar a API?
11. Em uma escala de 1 a 5 o quão desafiadores eram os problemas propostos pela API?
12. Em uma escala de 1 a 5, o quanto você aprendeu sobre segurança em APIs?
13. Após o contato com esta API, o seu interesse por segurança da informação aumentou?
14. Envie o código fonte da aplicação com as correções realizadas;
15. *Feedback* livre.