Análise de Vulnerabilidades da Plataforma Moodle com base no Top 10 da OWASP

Silvio E. Quincozes[1,2], Leonardo Frangello Franzese[2], Juliano F. Kazienko[3], Vagner E. Quincozes[4]

[1] Universidade Federal do Pampa (UNIPAMPA) – Alegrete, Brasil.
[2] FACOM – Universidade Federal de Uberlândia (UFU) – Uberlândia, Brasil.
[3] CTISM – Universidade Federal de Santa Maria (UFSM) – Santa Maria, Brasil.
[4] IC – Universidade Federal Fluminense (UFF) – Niterói, Brasil.

silvioquincozes@unipampa.edu.br, {sequincozes, leofrangello}@ufu.br
kazienko@redes.ufsm.br, vequincozes@midiacom.uff.br

Resumo. *Moodle é uma plataforma Web que contém informações acadêmicas e pessoais de mais 316 milhões de usuários. Em um cenário onde são registradas quantidades crescentes de ataques cibernéticos, cabe o questionamento: A plataforma Moodle está segura? A fim de investigar a resposta para essa pergunta, este trabalho apresenta um estudo de caso calcado nas principais vulnerabilidades reportadas pela lista Top 10 OWASP de 2021. Os resultados obtidos por meio da ferramenta OWASP Zed Attack Proxy (ZAP) revelaram 894 alertas de potenciais vulnerabilidade que podem ser potencialmente exploradas.*

1. Introdução

O Projeto Aberto de Segurança em Aplicações Web, do inglês, *Open Web Application Security Project* (OWASP) [OWASP 2022a] consiste em uma comunidade internacional sem fins lucrativos que tem por finalidade a produção de artigos, metodologias, documentações, ferramentas e tecnologias no campo de segurança de aplicativos web. Tal organização disponibiliza regularmente uma lista das dez principais vulnerabilidades, denominada *OWASP Top 10* [OWASP 2022a]. O *OWASP Top 10* serve para conscientizar os desenvolvedores sobre os riscos mais críticos em uma aplicação web [Santos et al. 2019].

A segurança cibernética é uma preocupação de nível mundial. Os dados recentes demonstram crescimento na quantidade de ocorrências de ataques cibernéticos. No 2° trimestre de 2022, houve um aumento de 32% nos ataques cibernéticos globais comparado ao 2° trimestre de 2021, onde ocorreu uma média de 1200 ataques semanais por organização. Já no Brasil, no mesmo período, os ataques aumentaram em 46% comparado ao período do 2° trimestre de 2021 [Schendes 2022]. Dentre os alvos desses ataques, destacam-se aqueles direcionados às aplicações web. De acordo com o relatório da [Kaspersky 2021] durante o ano de 2021, 15,45% dos computadores de usuários da internet global sofreram algum tipo de ataque.

Atualmente na literatura há trabalhos que apresentam esforços direcionados ao estudo das principais vulnerabilidades do *OWASP Top 10* [Priyawati et al. 2022] [Sampaio 2021] [Fredj et al. 2020]. No entanto, a maioria desses estudos concentram-se em versões anteriores, tal como o *OWASP Top 10* divulgado em 2017 —- que pode ser considerado desatualizado, uma vez que em setembro

de 2021 foi publicada uma relação atualizada das principais vulnerabilidades. Portanto, faltam estudos atualizados acerca dos grupos de vulnerabilidades mais frequentes em aplicações web atuais. Para preencher essa lacuna, este trabalho propõe uma investigação das principais vulnerabilidades encontradas na plataforma Moodle a partir da lista de vulnerabilidades *OWASP Top 10*, publicada em 2021. A principal contribuição deste trabalho consiste na aplicação da ferramenta de varredura de vulnerabilidades OWASP ZAP seguida da análise dos alertas identificados. Os resultados por meio desta ferramenta identificaram 894 alertas.

2. Materiais e Métodos

A metodologia, ilustrada na Figura 1, envolve a execução da ferramenta *Zed Attack Proxy* (ZAP) [OWASP 2022b], que envia requisições à plataforma Moodle [Dougiamas 2022] (passo 1) e recebe respostas. A análise dessas respostas (passo 2) permite que a ferramenta gere um relatório de alertas de potenciais vulnerabilidades (passo 3).

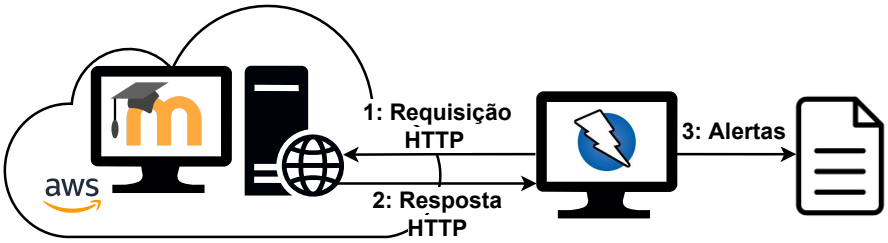

Figura 1. Cenário de experimentação adotado.

A escolha da plataforma Moodle como objeto de estudo se justifica pela preocupação existente com dados pessoais e acadêmicos de mais de 316 milhões de usuários no mundo [Dougiamas 2022]. Já a ferramenta OWASP ZAP foi escolhida devido às suas funcionalidades gratuitas e de código-aberto, com amplo reconhecimento pela comunidade. O cenário experimental envolve a instalação do OWASP ZAP (versão 2.12.0) em uma máquina virtual com Kali Linux, e do Moodle (versão 4.1) em uma máquina virtual na Amazon AWS com Ubuntu 22.04 e com o servidor Apache na versão 2.4.52. O OWASP ZAP executa um *scan* automático dividido em varredura passiva e ativa, registrando todas as requisições e gerando alertas de vulnerabilidades, destacando a quantidade, tipos, criticidade e URLs das vulnerabilidades encontradas.

3. Resultados e Discussão

A Tabela 1 sumariza os alertas reportados pela ferramenta OWASP ZAP. Em particular, foram identificados 894 alertas que são divididos de acordo com seu risco: *Informativo*, *baixo*, *médio* e *alto*. Embora existam dez vulnerabilidades no *OWASP Top 10*, os alertas reportados pelo OWASP ZAP para a plataforma Moodle encontram-se entre as cinco primeiras vulnerabilidades (A01, A02, A03, A04 e A05, conforme [OWASP 2022a]).

Note que muitos dos alertas reportados são originados da falta de configurações adicionais além da instalação padrão do Moodle. O objetivo desta análise consiste em revelar o quão vulnerável pode ser uma instalação do Moodle quando realizada por um usuário que não tem o perfil de especialista em segurança da informação e, portanto, não implementa medidas adicionais além daquelas que a própria plataforma oferece.

Tabela 1. Alertas reportados pela ferramenta OWASP ZAP na plataforma Moodle.

Nome da vulnerabilidade	Risco	Categoria no *OWASP Top 10*	Alertas
SQL Injection	Alto	A03 - Injection	5
Missing Anti-CSRF tokens	Médio	A05 - Security Misconfiguration	56
CSP Header Not Set	Médio	A05 - Security Misconfiguration	37
HTTP to HTTPS Insecure Transition in Form Post	Médio	A02 - Cryptographic Failures	4
Hidden File Found	Médio	A05 - Security Misconfiguration	1
Missing Anti-Clickjacking Header	Médio	A05 - Security Misconfiguration	11
.htaccess Information Leak	Médio	A05 - Security Misconfiguration	2
Big Redirect Detected[1]	Baixo	A04 - Insecure Design	12
Cookie No HTTPOnly Flag	Baixo	A05 - Security Misconfiguration	5
Cookie without SameSite Attribute	Baixo	A01 - Broken Access Control	5
Disclosure of Date and Time - Unix	Baixo	A01 - Broken Access Control	150
Server Leaks Version Information[2]	Baixo	A05 - Security Misconfiguration	68
Strict-Transport-Security Header Not Set	Baixo	A05 - Security Misconfiguration	11
X-Content-Type-Option Header Missing	Baixo	A05 - Security Misconfiguration	65
Information Disclosure - Suspicious Comments	Informativo	A01 - Broken Access Control	89
Information Disclosure - Sensitive Information in URL	Informativo	A01 - Broken Access Control	2
Modern Web Application	Informativo	Nenhum	55
Re-examine Cache-control Directives	Informativo	Nenhum	8
User Agent Fuzzer	Informativo	Nenhum	240
User Controllabe HTML Element Attribute (Potential XSS)	Informativo	A03 - Injection	68

A seguir, os alertas reportados pela ferramenta OWASP ZAP são apresentados e discutidos. Como contribuição adicional, são descritos os principais ataques que podem ser realizados através da exploração de tais vulnerabilidades, bem como as devidas contramedidas recomendadas para cada situação.

3.1. Vulnerabilidades de Risco Alto

Dentre as vulnerabilidades reportadas e categorizadas pela ferramenta OWASP ZAP [OWASP 2022b], apenas uma categoria foi classificada como tendo alto nível de risco. Foram gerados 5 alertas para a possibilidade de ataques de injeção de comandos através da vulnerabilidade *SQL Injection*. Essa vulnerabilidade pertence à categoria *Injection*, a qual ocupa a terceira posição no *OWASP Top 10*, conforme [OWASP 2022a].

As falhas de SQL Injection são introduzidas quando os desenvolvedores de software criam consultas de banco de dados dinâmicas construídas com concatenação de *strings* que inclui entrada fornecida pelo usuário. Assim, uma potencial forma de ataque à plataforma Moodle pode se dar manipulando o valor do parâmetro *logintoken* ao enviar uma requisição *POST* a partir da página inicial */moodle/login/index.php*.

De acordo com o relatório da ferramenta, os resultados da página foram manipulados com sucesso através de uma tentativa de ataque automatizado que usa as condições booleanas denotadas nas Equações 1 e 2.

$$[NeM7GmXnkNHvlEGtgNg5SIQ2rFw0EtCe' AND '1' =' 1' --] \quad (1)$$

$$[NeM7GmXnkNHvlEGtgNg5SIQ2rFw0EtCe' AND '1' =' 2' --] \quad (2)$$

Como principais contramedidas para a contenção de ataques de injeção, como o *SQL Injection*, recomenda-se verificar todos os dados no lado do servidor. Não deve-se assumir que as informações provindas da aplicação cliente são sempre confiáveis, mesmo que hajam validações implementadas na aplicação cliente. Ademais, o uso de funções como o `mysqli_real_escape_string` ou procedimentos (*procedures*) podem ser

usados como formas de mitigação da vulnerabilidade apresentada. Por fim, a adoção do princípio de privilégio mínimo é fortemente recomendada como forma de garantir que um atacante que explore a vulnerabilidade de *SQL Injection* esteja limitado às permissões de seu grupo de usuários.

3.2. Vulnerabilidades de Risco Médio

As vulnerabilidades reportadas pelo OWASP ZAP como risco médio constituem variações das vulnerabilidades *Cryptographic Failures* e *Security Misconfiguration*, as quais ocupam a segunda e quinta posição no *OWASP Top 10*, respectivamente. Foram gerados 111 alertas para seis tipos subcategorias dessas vulnerabilidades, conforme detalhado a seguir.

Os 56 alertas de *Cross-Site Request Forgery* (CSRF) foram causadas devido à ausência de *tokens* que são tipicamente usados para prevenir ataques que explorem essa vulnerabilidade. O *token anti-CSRF* deve ser representado por um valor de tamanho grande e aleatório para dificultar a sua descoberta e também ser exclusivo por sessão de usuário. Ao explorar essa vulnerabilidade, o atacante força a vítima a enviar solicitações HTTP para um destino sem seu conhecimento ou intenção. Nas fases de arquitetura e design, as contramedidas cabíveis incluem a geração e verificação de *nonces* exclusivos para formulários. Já na fase de implementação, uma contramedida que pode ser adotada consiste na verificação do cabeçalho HTTP *Referer* de modo a confirmar se a solicitação se originou de uma página esperada. Note que a segunda solução pode ser ineficiente em casos em que usuários ou *proxies* legítimos desabilitam o envio deste cabeçalho por razões de privacidade [OWASP 2022c].

Os 37 alertas de *Content Security Policy* (CSP) *Header Not Set* ocorreram devido a falta de configuração deste cabeçalho HTTP. É importante observar que tal configuração não faz parte de um procedimento previsto durante a instalação padrão da plataforma Moodle. Portanto, instalações da plataforma que não contemplarem tais medidas adicionais podem ficar vulneráveis. Um atacante pode explorar tais vulnerabilidades para executar ataques de injeção de dados, por exemplo. Como forma de mitigação, deve-se assegurar que o servidor web esteja devidamente configurado para suportar a definição deste cabeçalho pelo cliente. As definições de cabeçalhos CSP podem variar de acordo com o navegador e suas versões [OWASP 2022c].

Os 4 alertas *HTTP to HTTPS Insecure Transition in Form Post* revelam que a plataforma Moodle não foi instalada em um ambiente seguro que contemple o uso do protocolo *Hyper Text Transfer Protocol Secure* (HTTPS). Por conta disso a aplicação fica vulnerável à ataques *Man-In-The-Middle* (MITM), possibilitando que um invasor consiga interceptar a troca de dados entre duas partes e roubar suas informações. Como forma de mitigação, recomenda-se fortemente a utilização do protocolo HTTPS ao invés do HTTP.

O alerta de *Hidden File Found* revela que há um arquivo potencialmente confidencial que pode estar exposto a usuários não autorizados. Isso pode permitir que usuários mal-intencionados possam obter informações administrativas, obter conhecimento da configuração da aplicação e conseguir credenciais de acesso, possibilitando que ele consiga expandir os seus métodos de ataques na aplicação. Uma outra possibilidade consiste em realizar engenharia social usando as informações obtidas. Recomenda-se fortemente desativar da produção qualquer componente que seja desnecessário ou então garantir que o seu acesso seja restrito a usuários autenticados.

Os 11 alertas de *Missing Anti-Clickjacking Header* ocorreram pois o cabeçalho *X-Frame-Options* não foi configurado no cabeçalho de resposta HTTP. Com isso, o invasor pode realizar ataque de *Clickjacking*, onde ocorre uma sobreposição de elementos invisíveis ou semi-transparentes em cima de áreas clicáveis da página. Para mitigar essa vulnerabilidade, é necessário certificar-se que o cabeçalho*X-Frame-Options* esteja configurado para todas as páginas web utilizadas pela aplicação.

Os 2 alertas de *.htaccess Information Leak* ocorrem devido ao acesso indevido ao arquivo *.htaccess* da aplicação. Esse arquivo pode ser utilizado para realizar alterações nas configurações do servidor Apache. Portanto, um invasor ao conseguir acesso a este arquivo, pode realizar a ativação e desativação de funcionalidades que o servidor Apache pode oferecer. Portanto, a melhor maneira para remover esse alerta consiste em desativar qualquer tipo de acesso a este arquivo.

3.3. Vulnerabilidades de Risco Baixo

As vulnerabilidades reportadas pelo OWASP ZAP como risco baixo constituem variações das categorias de *Security Misconfiguration*, *Broken Access Control* e *Insecure Design*. No total, foram gerados 316 alertas para sete tipos de subcategorias dessas vulnerabilidades, conforme detalhado a seguir.

Um exemplo de falha que pode ser explorada para revelar o *cookie* de um usuário para terceiros é o *Cross-Site Scripting*. Pode-se mitigar o acesso de *cookies* por terceiros ao incluir o sinalizador *HTTPOnly* no cabeçalho de resposta HTTP [OWASP 2022c]. Os 5 alertas de *Cookie No HTTPOnly* foram gerados devido a ausência do atributo *HTTPOnly* no cabeçalho HTTP *Set-Cookie*. Este atributo tem por finalidade impedir que os *cookies* sejam acessados através de *scripts*. Dessa forma, um tipo de ataque que pode explorar essa falta de configuração consiste no *Cross-Site Scripting*.

Os 5 alertas de *Cookie without SameSite Attribute* foram gerados devido à ausência do atributo *SameSite* no cabeçalho HTTP *Set-Cookie*. Tal atributo tem por finalidade evitar que um *cookie* seja enviado como resultado de uma solicitação entre sites. Dessa forma, um tipo de ataque que pode explorar essa falta de configuração consiste no CSRF, já mencionado anteriormente. Então, para mitigar a possibilidade de um atacante falsificar solicitações entre sites através do envio de *cookies*, deve-se habilitar o atributo *SameSite* no cabeçalho de resposta [OWASP 2022c],

Os 12 alertas de *Big Redirect Detected (Potential Sensitive Information Leak)* ocorreram devido ao tamanho da resposta recebida pelo servidor no momento em que o mesmo efetua redirecionamentos, o qual foi considerado longo pela ferramenta OWASP ZAP. O potencial risco dessa vulnerabilidade consiste nos casos em que a resposta do redirecionamento contém informações confidenciais ou dados pessoais. Ademais, com base em tais informações, novos ataques podem ser explorados. Portanto, o jeito mais eficaz de mitigar essa vulnerabilidade é configurar o servidor para que o mesmo não exiba conteúdos confidenciais ou privados no redirecionamento [OWASP 2022c].

Os 150 alertas de *Disclosure of Date and Time - Unix* ocorreram devido ao envio das informações de data e hora do servidor na resposta HTTP. O principal problema do vazamento dessa informação consiste em facilitar a um atacante a obtenção de informações internas para a criação de padrões de exploração. Para mitigar esse tipo de vazamento de informação, deve-se verificar se o uso dessa informação por parte de um atacante pode ser

agregada com outras informações para o mapeamento de padrões por atacantes. Em caso positivo, deve-se remover o campo *timestamp* do cabeçalho resposta [OWASP 2022c].

Os 68 alertas de *Server Leaks Version Information* ocorreram devido ao vazamento da versão do servidor *web* na resposta HTTP. Esse tipo de informação pode ser usada pelo atacante para identificar outras vulnerabilidades que são conhecidas naquela versão em que o servidor se encontra. Como forma de mitigação, pode-se remover o campo *"Server"* do cabeçalho de resposta HTTP.

Os 11 alertas gerados por *Strict-Transport-Security Header Not Set* ocorreram devido a configuração deste cabeçalho HTTP não ser um procedimento que faz parte da instalação padrão da plataforma Moodle. O HTTP *Strict-Transport-Security* (HSTS) consiste em um aprimoramento de segurança opcional que redireciona solicitações HTTP não seguras para HTTPS. A falta da utilização desse cabeçalho possibilita que ataques como *Man-In-The-Middle* ocorra. Portanto, como forma de mitigação, deve-se realizar a configuração deste cabeçalho no servidor web.

Os 65 alertas de *X-Content-Type-Option Header Missing* ocorreram devido a falta de configuração do cabeçalho *X-Content-Type-Option*, o qual faz parte do mecanismo de *Anti-Mime-Sniffing*. Sem essa configuração um invasor pode disfarçar um arquivo, fazendo com que ele se pareça outro, conseguindo assim ataques do tipo *Cross-Site-Scripting*. Portanto, para mitigar essa vulnerabilidade, recomenda-se a configuração do cabeçalho *X-Content-Type-Options* como *nosniff* para todas as páginas web.

3.4. Motivos Informativos

Foram gerados 89 alertas de *Suspicious Comments* devido a comentários suspeitos que podem expor informações sobre o funcionamento do sistema, além de 2 alertas de *Information Disclosure* por suspeita de vazamento de informações confidenciais em URLs. Houve também 55 alertas informativos sobre recursos modernos do Moodle, 8 alertas relacionados à má configuração de controle de cache, 240 alertas de *User Agent Fuzzer* indicando possíveis vulnerabilidades ao modificar o *User Agent*, e 68 alertas de Potencial XSS devido à modificação de atributos HTML via parâmetros na URL, todos requerendo ações específicas para mitigação.

4. Conclusão e Trabalhos Futuros

Este trabalho apresentou um estudo de caso acerca das potenciais vulnerabilidades da lista *OWASP Top 10* atualizada presente no ambiente virtual de ensino e aprendizagem Moodle por meio da ferramenta OWASP ZAP. A plataforma Moodle é utilizada por mais de 316 milhões de usuários e com o aumento de ataques cibernéticos é de extrema importância verificar o nível de segurança da plataforma ao ser instalada por um usuário comum. Portanto esse estudo mostrou que a plataforma Moodle, ao ser instalada sem qualquer tipo de configuração de segurança adicional está sujeita a diversas vulnerabilidades, através das quais um invasor pode se aproveitar para efetuar ataques na aplicação. Tal análise foi realizada utilizando o *scan* automático da ferramenta OWASP ZAP, o qual gerou um total de 894 alertas divididos entre 20 vulnerabilidades. Como trabalhos futuros, pretende-se realizar a configuração dos métodos de mitigação sugeridos ao longo do trabalho e realizar novamente o *scan* da ferramenta OWASP ZAP para verificar a eficácia desses métodos.

Referências

Dougiamas, M. (2022). Aprendizado on-line com o LMS mais popular do mundo - Moodle. Disponível em: https://moodle.org/.

Fredj, O. B., Cheikhrouhou, O., Krichen, M., Hamam, H., and Derhab, A. (2020). An owasp top ten driven survey on web application protection methods. In *International Conference on Risks and Security of Internet and Systems*, pages 235–252, New York, NY, USA. Springer, Springer, Cham.

Kaspersky (2021). Kaspersky security bulletin 2021. https://go.kaspersky.com/rs/802-IJN-240/images/KSB_statistics_2021_eng.pdf.

OWASP (2022a). Owasp top 10:2021. Disponível em: https://owasp.org/www-project-top-ten/. Acesso em: 25 set. 2022.

OWASP (2022b). Owasp zed attack proxy (zap). Disponível em: https://www.zaproxy.org/.

OWASP (2022c). Owasp zed attack proxy (zap) - alerts documentation. Disponível em: https://www.zaproxy.org/docs/alerts/. Acesso em: 25 set. 2022.

Priyawati, D., Rokhmah, S., and Utomo, I. C. (2022). Website vulnerability testing and analysis of website application using owasp. *International Journal of Computer and Information System*, 3(3):142–147.

Sampaio, F. F. (2021). *Uma análise prática das principais vulnerabilidades em aplicações web baseado no top 10 OWASP*. Bacharelado, Universidade Federal do Paraná (UFC), Paraná.

Santos, L. C. M. C., Prado, E. P. V., and Chaim, M. L. (2019). Vulnerability Detection Techniques and Tools and Their Relationship to Agile Methods and Software Quality and Service Models. In *Proceedings of the XV Brazilian Symposium on Information Systems*, SBSI'19. Association for Computing Machinery.

Schendes, W. (2022). Ataques cibernéticos no brasil cresceram 46% no segundo trimestre. https://olhardigital.com.br/2022/08/09/seguranca/ataques-ciberneticos-brasil-cresce-46/.