DOI: 10.5748/20CONTECSI/PSE/SEC/7312

eLocator: e207312

TEACHING INFORMATION SECURITY: A PRACTICAL STUDY ON CROSSSITE SCRIPTING

Rafael Fernando Diorio – https://orcid.org/0000-0002-5574-0941
Instituto Federal De Educação, Ciência E Tecnologia De São Paulo - Ifsp

TEACHING INFORMATION SECURITY: A PRACTICAL STUDY ON CROSS-SITE SCRIPTING

ABSTRACT

Injection attacks are among the most dangerous attacks targeting Web applications. Among all types of injection attacks, Cross-Site Scripting (XSS), in whichmalicious scripts are injectedintoa legitimate andtrustedWeb application or website, is one of the most common attack vectors. For this reason, it is crucial that Information Technology students and professionals, especially those focused on information security, are prepared to deal with these attacks. In this context, a practical study on XSS attacks is discussed in this paper, exploring reflected XSS. To this end, based on a reference scenario, related attacks are reproduced for experimentation and discussion purposes. The experiments and discussion are based on a practical laboratory about Web application security, carried out with third-year students of an undergraduate computer science course to help them understand concepts related to the topic.

Keywords: Web Application Security, Cross-Site Scripting, Experimentation, Teaching and Learning, Information Security.

ENSINO DE SEGURANÇA DA INFORMAÇÃO: UM ESTUDO PRÁTICO SOBRE CROSS-SITE SCRIPTING

RESUMO

Os ataques de injeção estão entre os ataques mais perigosos direcionados a aplicações Web. Entre todos os tipos de ataques de injeção, o Cross-Site Scripting (XSS), no qual scripts maliciosos são injetados em umaaplicação Web ou website legítimo e confiável, é um dos vetores de ataque mais comuns. Por esse motivo, é crucial que estudantes e profissionais da área de Tecnologia da Informação, em especial, focados em segurança da informação, estejam preparados para lidar com tais ataques. Nesse contexto, um estudo prático sobre ataques de XSS é discutido neste trabalho, explorando o XSS refletido. Para tal, com base em um cenário de referência, ataques relacionados são reproduzidos para fins de experimentação e discussão. Os experimentos e discussão baseiam-se em um laboratório prático sobre segurança de aplicações Web, realizado com estudantes do terceiro ano de um curso de graduação em informática para auxiliá-los na compreensão de conceitos relacionados ao tema.

Palavras-chave: Segurança de Aplicações Web, Cross-Site Scripting, Experimentação, Ensino e Aprendizagem, Segurança da Informação.

1. Introdução

Entre os ataques direcionados a aplicações Web, o Cross-Site Scripting (XSS) é um dos mais comuns (En&Selvarajah, 2022; Hydaraet al., 2015; Yao et al., 2023). É considerado um tipo de ataque de injeção (OWASP Foundation, 2023a), listado entre os riscos de segurança mais críticos para aplicações Web pelo Open Web Application Security Project (OWASP) (OWASP Foundation, 2023b), tendo afetado diversas organizações ao longo dos anos (Liu et al., 2019).

Funcionalmente, de acordo com Yao et al. (2023), os ataques de XSS ocorrem, principalmente, quando as aplicações Web não tratam adequadamente as entradas dos usuários, resultando na injeção de scripts maliciosos pelos atacantes. Em termos de classificação, os ataques de XSS podem ser divididos/categorizados como XSS refletido, XSS armazenado e XSS baseado em DOM (DocumentObjectModel), executados no lado do servidor (XSS refletido, XSS armazenado) ou no lado do cliente (XSS baseado em DOM) (En&Selvarajah, 2022; Hydaraet al., 2015; Liu et al., 2019). Nesse sentido, no XSS refletido, o script injetado é refletido no servidor Web, como em uma mensagem de erro ou resultado de pesquisa, por exemplo (OWASP Foundation, 2023a). No XSS armazenado, o script injetado é armazenado permanentemente no servidor de destino, como em um banco de dados ou em um fórum de mensagens, por exemplo (Hydara et al., 2015; OWASP Foundation, 2023a). Por sua vez, no XSS baseado em DOM, que é causado por código inseguro no lado do cliente e não no lado do servidor, o código malicioso é executado no lado do cliente (Hydara et al., 2015; Liu et al., 2019).

Quando bem-sucedidos, os ataques de XSS permitem ao atacante obter dados confidenciais do usuário, como tokens de sessão e cookies, por exemplo, e também podem ser explorados para reescrever o conteúdo de sites, espalhar malware, realizar phishing e causar negação de serviço, entre outros (En&Selvarajah, 2022; Hydara et al., 2015; Liu et al., 2019; OWASP Foundation, 2023a; Xu et al., 2020; Yao et al., 2023; Yusof&Pathan, 2016). Dessa forma, o XSS bem-sucedido pode resultar em graves violações de segurança (Hydaraet al., 2015), com efeitos relacionados podendo ser extremamente críticos para organizações e indivíduos. Por tais motivos, várias contribuições recentes abordam questões relacionadas aos ataques de XSS, tal como nos trabalhos de Alenzi&Abbase (2022), En&Selvarajah (2022), Liu et al. (2019), Xu et al. (2020) e Yao et al. (2023), entre outros. Ainda assim, os ataques de XSS estão entre os riscos de segurança mais críticos para aplicações Web na atualidade.

Diante desse cenário, objetivando contribuir com outros trabalhos relacionados ao tema e com ênfase no processo de ensino e aprendizagem em segurança da informação, este trabalho discute um estudo prático sobre ataques de XSS, abordando a reprodução do XSS refletido. Para tal, a partir de um cenário de referência, um ambiente computacional é utilizado para fins de experimentação e discussão. Os experimentos e discussão baseiam-se em um laboratório prático sobre segurança de aplicações Web, realizado com estudantes do terceiro ano de um curso de graduação em informática, em uma das aulas de segurança da informação, para auxiliá-los na compreensão de conceitos sobre XSS. Essa discussão é importante, por exemplo, para conciliar teoria e prática em atividades de ensino relacionadas ao tema, bem como para que novas pesquisas e contribuições sejam realizadas no âmbito da segurança da informação, tais como para a prevenção e para a detecção dos ataques de XSS.

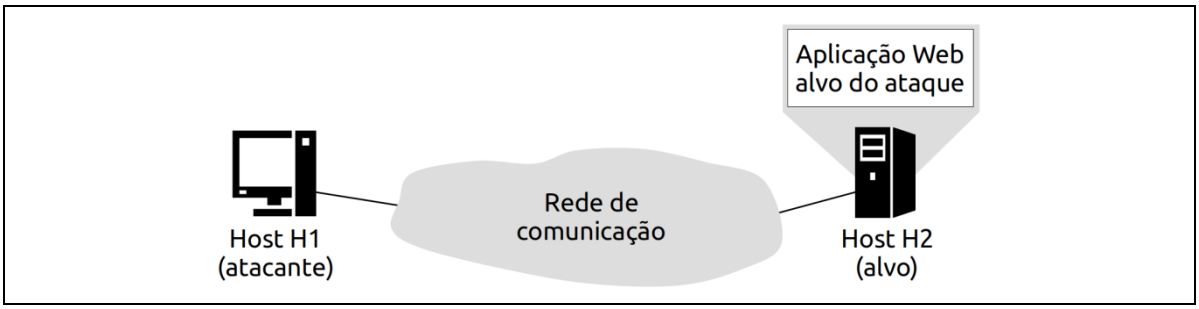
O restante deste trabalho está organizado da seguinte forma: o cenário de referência utilizado para fins de experimentação e discussão é descrito na Seção 2. Os materiais e métodos são descritos na Seção 3. Os resultados experimentais são descritos na Seção 4 e, por fim, a conclusão e os trabalhos futuros são descritos na Seção 5.

2. Cenário de Referência

O cenário de referência utilizado para fins de experimentação e discussão sobre ataques de XSS é ilustrado na Figura 1.

Figura 1

Cenário de referência utilizado para fins de experimentação e discussão sobre ataques de XSS.



Conforme ilustrado na Figura 1, dois hosts são utilizados no cenário de referência explorado neste trabalho: um host atacante (H1) e um host alvo (H2). Ambos os hosts estão interconectados por meio de uma rede de comunicação baseada em um ambiente computacional virtualizado, em que o host H1 atua como cliente e o host H2 atua como servidor para os serviços de hospedagem de páginas Web e de banco de dados, hospedando a aplicação Web alvo do ataque de XSS.

Em tal cenário, a aplicação Web alvo do ataque de XSS é acessível por meio do endereço http://www.exemplo.com.br/app e tem como referência um sistema acadêmico, com interfaces de acesso público, acessíveis por qualquer usuário, e interfaces de acesso restrito, acessíveis por usuários autenticados. O comportamento esperado da aplicação, sem vulnerabilidades, é que seus recursos e funcionalidades não sejam comprometidos por entradas maliciosas dos usuários (isto é, entradas maliciosas dos atacantes).

3. Materiais e Métodos

Para a implementação do cenário de referência ilustrado na Figura 1, a solução Oracle VM VirtualBox[1] foi utilizada, com ambos os hosts (H1 e H2) baseados em máquinas virtuais. Em tal cenário, o host H1 foi configurado utilizando o sistema operacional Kali Linux[2] 2023.2 e o host H2 foi configurado utilizando o sistema operacional Linux CentOSStream[3] 9. No host H2, os serviços de hospedagem de páginas Web e de banco de dados foram configurados, respectivamente, por meio das soluções

[1]Oracle VM VirtualBox: https://www.virtualbox.org/
[2]Kali Linux: https://www.kali.org/
[3]Linux CentOSStream: https://www.centos.org/centos-stream/

Apache, com suporte ao PHP, e MySQL. A aplicação Web alvo do ataque, hospedada no host H2, foi desenvolvida utilizando as soluções PHP e MySQL.

Nesse cenário, os ataques foram realizados do host H1 para o host H2, explorando oXSS na aplicação Web hospedada junto ao mesmo de forma manual, sem a utilização de ferramentas automatizadas em tal processo. A exploração do XSS foi realizada de forma manual, em especial, para fins didáticos, objetivando apresentar possíveis técnicas e abordagens que podem ser utilizadas pelo atacante para a realização de tais ataques.

Na aplicação Web alvo do ataque, as explorações tiveram como referência uma de suas interfaces de acesso público, utilizada para fins de pesquisa de cursos (Figura 2). Em tal interface, o XSS foi explorado por meio do campo destinado ao nome do curso a ser pesquisado.Quanto ao código-fonte da interface ilustrada na Figura 2, a Figura 3 ilustra o trecho pertinente ao mesmo responsável por receber a entrada do usuário (linha 1) e por exibir tal entrada como parte do retorno da aplicação para a pesquisa de cursos (linha 2).

Figura 2

Interface da aplicação Web alvo do ataque de XSS, com destaque para o campo utilizado para fins de experimentação.

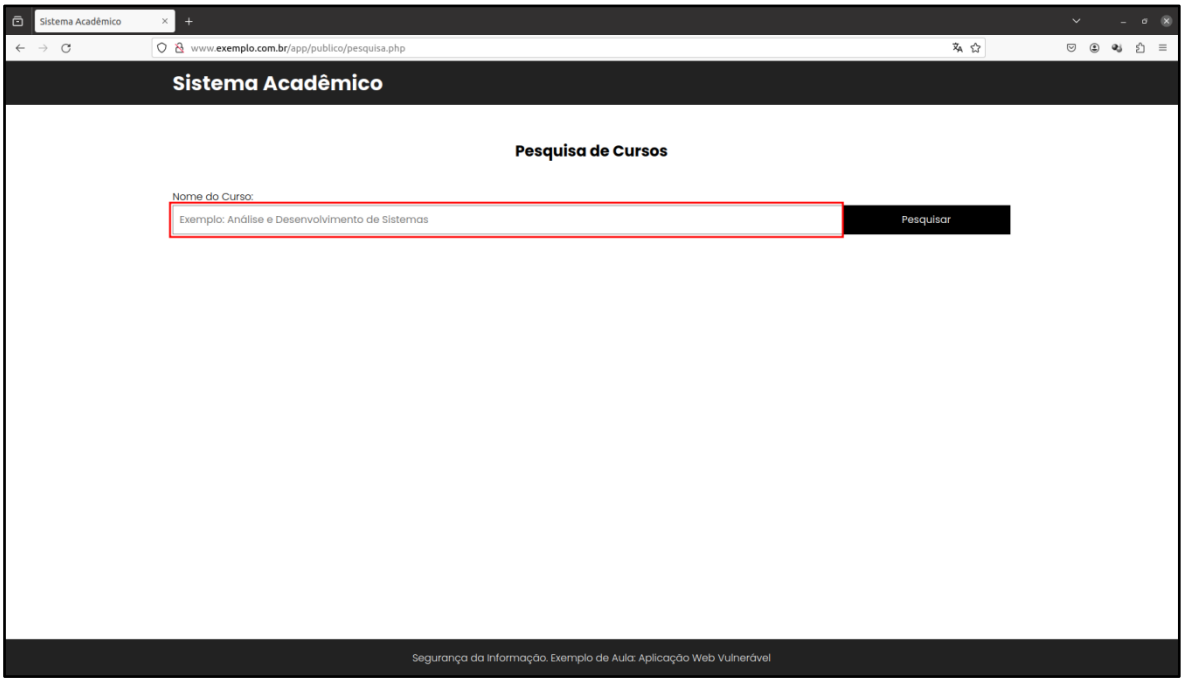


Figura 3

Trecho do código-fonte da interface ilustrada na Figura 2 responsável por receber a entrada do usuário (linha 1) e por exibir tal entrada como parte do retorno da aplicação para a pesquisa de cursos (linha 2).

```
1   $search = $_GET["search"];
2   <p style="margin: 0;"><strong>Resultado da pesquisa por:
    </strong><?phpecho $search; ?></p>
```

Para fins de experimentação, considerou-se que o atacante tinha como objetivo aobtenção de cookies dos usuários, que poderiam levar ao sequestro de suas sessões quando autenticados na aplicação Web alvo do ataque,bem como a manipulação de conteúdos apresentados pela aplicaçãono resultado da pesquisa de cursos disponibilizada pela mesma, via exibição de cursos inexistentes em seu banco de dados, que poderiam afetar a reputaçãoda organização responsável pela aplicação.

4. Resultados e Discussão

Considerando o cenário de referência descrito na Seção 2 e os materiais e métodos descritos na Seção 3, uma possível abordagem a ser utilizada pelo atacante para detectar se a aplicação Web alvo do ataque é vulnerável ao XSS se dá pela entrada de um script, como para exibição de uma mensagem qualquer, por exemplo, enquanto entrada para a pesquisa de cursos. Nesse sentido, se a aplicação for vulnerável ao XSS, o script será executado e a mensagem será exibida ao atacante. Caso contrário, se a aplicação não for vulnerável ao XSS, o script não será executado e a mensagem não será exibida ao atacante. Dessa forma, para efeitos de experimentação, a Figura 4 ilustra um exemplo de script para exibição da mensagem“XSS– Exemplo de Aula” e a Figura 5 ilustra o retorno da aplicação ao receber como entrada para a pesquisa de cursos o script ilustrado na Figura 4.

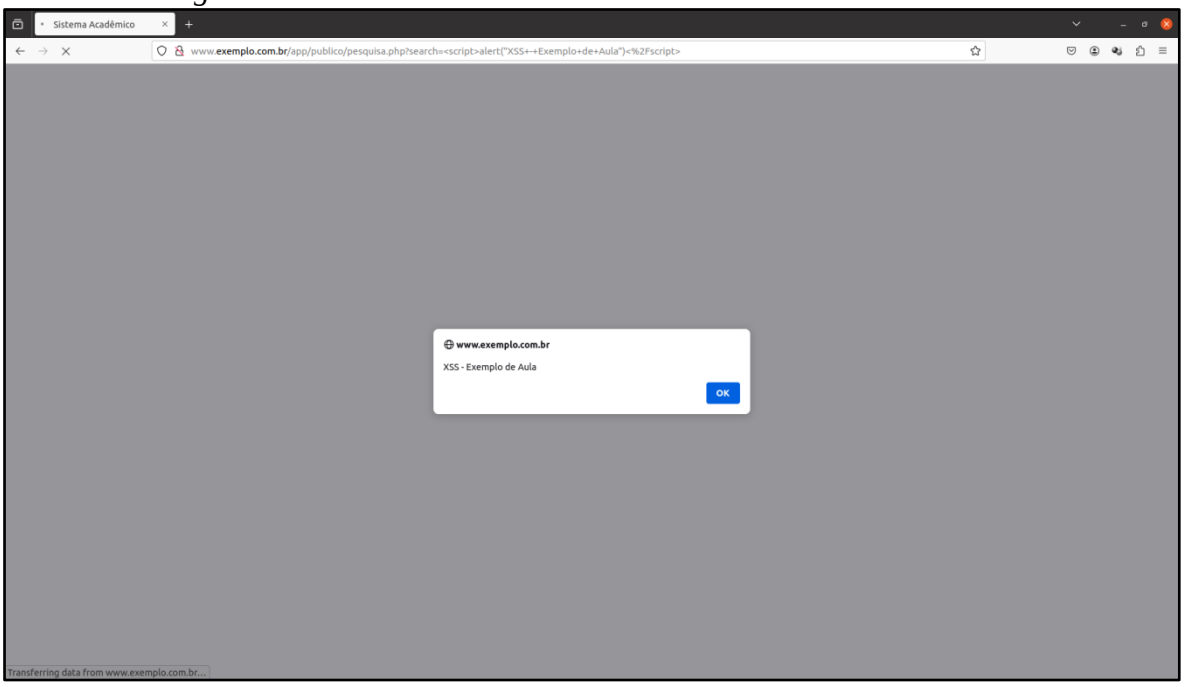
Figura 4

Exemplo de script para exibição da mensagem “XSS – Exemplo de Aula”.

```
<script>alert("XSS - Exemplo de Aula")</script>
```

Figura 5

Retorno da aplicação ao receber como entrada para a pesquisa de cursos o script ilustrado na Figura 4.



Por meio do retorno da aplicação ilustrado na Figura 5, é possível observar que o script ilustrado na Figura 4 foi executado pela mesma, exibindo a mensagem “XSS – Exemplo de Aula”.De modo complementar, a Figura 6 ilustra o código-fonte da página da aplicação após a execução do script ilustrado na Figura 4, com destaque para o código pertinente ao mesmo como parte do código-fonte da página. Por meio de tal ilustração, pode-se observar que o script ilustrado na Figura 4, enquanto entrada do atacante para a pesquisa de cursos,foi incorporado ao código-fonte da página de modo integral, sem qualquer tipo de tratamento quanto ao XSS. Diante desse cenário, pode-se observar que a aplicação Web alvo do ataque é vulnerável ao XSS.

Figura 6

Código-fonte da página da aplicação após a execução do script ilustrado na Figura 4, com destaque para o código pertinente ao mesmo como parte do código-fonte da página.

Após detectar que a aplicação Web é vulnerável ao XSS, o atacante pode explorá-lano âmbito do objetivo descrito na Seção 3, ou seja, para aobtenção de cookies dos usuários, bem como para a manipulação de conteúdos apresentados pela aplicação no resultado da pesquisa de cursos disponibilizada pela mesma.

Nesse contexto, para efeitos de experimentação, a Figura 7 ilustra um exemplo de script para a obtenção do cookie do usuário autenticado na aplicação Web, e a Figura 8 ilustra um exemplo de código HTML para a manipulação de conteúdos apresentados pela aplicação enquanto resultado da pesquisa de cursos disponibilizada pela mesma, nesse caso, apresentando informações sobre dois cursos inexistentes no banco de dados da aplicação: “Tecnologia em Assuntos Aleatórios” e “Especialização em Assuntos Aleatórios”. Na Figura 7, “http://www.atacante.com.br” refere-se ao endereço do atacante, utilizado para receber o cookie da vítima após a execução do script exemplificadona figura em questão.

Figura 7

Exemplo de script para obter o cookie do usuário autenticado na aplicação Web.

```
<script>window.location="http://www.atacante.com.br/?cookie=" +
document.cookie</script>
```

Figura 8

Exemplo de código HTML para apresentar informações sobre dois cursos inexistentes no banco de dados da aplicação: "Tecnologia em Assuntos Aleatórios" e "Especialização em Assuntos Aleatórios".

```
<br><br>
<divclass="row">
<divclass="column cl-100 column-content">
<table id="tabela">
<tr>
<th>Nome</th>
<th>Nível</th>
<th>Duração</th>
<th>Período</th>
</tr>
<tr>
<td>Tecnologia em Assuntos Aleatórios</td>
<td>Graduação</td>
<td>2 anos</td>
<td>Noturno</td>
</tr>
<tr>
<td>Especialização em Assuntos Aleatórios</td>
<td>Pós-graduação</td>
<td>1 ano</td>
<td>Noturno</td>
</tr>
</table>
</div>
</div>
```

De modo complementar ao ilustrado na Figura 7, para receber o cookie da vítima, o atacante deve ter um servidor Web em execução no endereço especificado, no exemplo, acessível via "http://www.atacante.com.br". Para fins de experimentação, a Figura 9 ilustra o comando utilizado no host atacante para iniciar o servidor Web utilizado para esse propósito, nesse caso, via Python.

Figura 9

Comando utilizado no host atacante para iniciar o servidor Web utilizado para receber o cookie da vítima após a execução do script ilustrado na Figura 7.

```
sudo python3 -m http.server 80
```

Diante desse cenário, a Figura 10 ilustra o retorno da aplicação ao receber como entrada para a pesquisa de cursos o script ilustrado na Figura 7, a Figura 11 ilustra o recebimento do cookie da vítima pelo atacante e a Figura 12 ilustra o retorno da aplicação ao receber como entrada para a pesquisa de cursos o código HTML ilustrado na Figura 8.

Figura 10

Retorno da aplicação ao receber como entrada para a pesquisa de cursos o script ilustrado na Figura 7.

Figura 11

Recebimento do cookie da vítima pelo atacante.

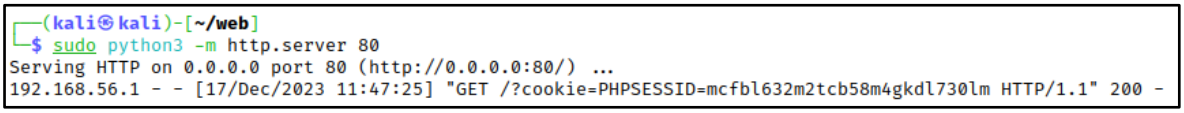
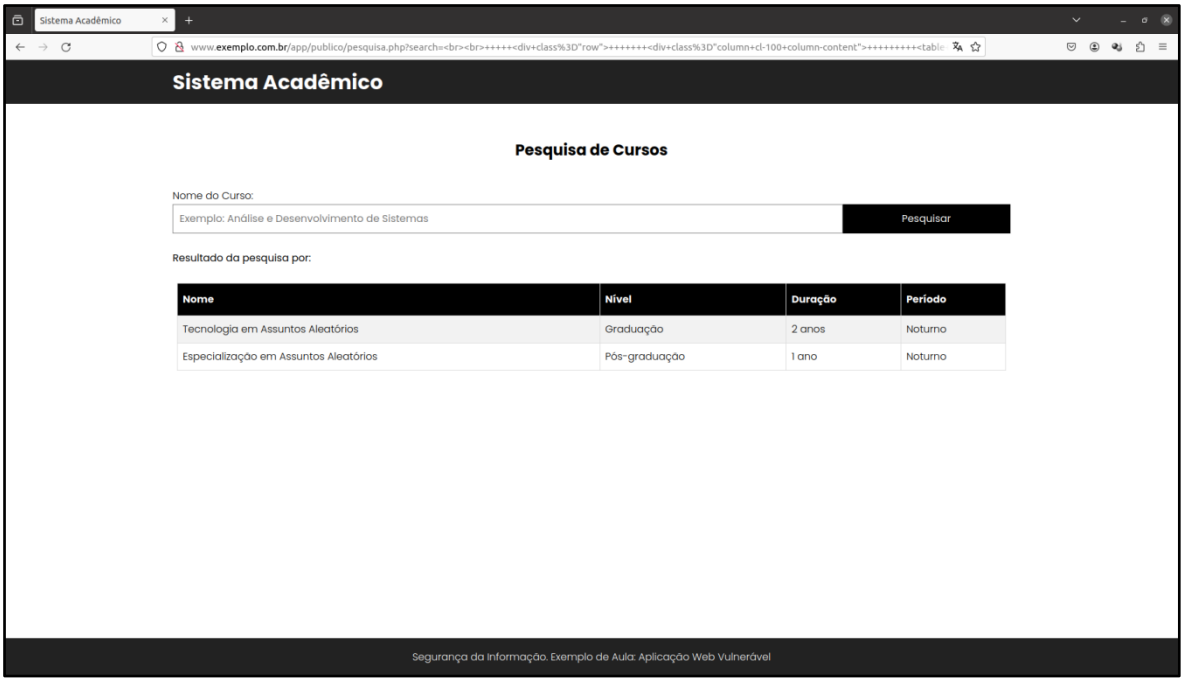


Figura 12

Retorno da aplicação ao receber como entrada para a pesquisa de cursos o código HTML ilustrado na Figura 8.



Diante desse cenário, por meio do XSS bem-sucedido, o atacante poderia, por exemplo, utilizar o cookie recebido do usuário/vítima(Figura 11), quando autenticado na aplicação Web alvo do ataque,para sequestrarsua sessão e interagir com a aplicação se passando pelo mesmo, bem comoutilizar a manipulação de conteúdos apresentados pela aplicação no resultado da pesquisa de cursos (Figura 12) para afetar a reputação da organização responsável pela aplicação, bem como o relacionamento da mesma com seus clientes. Para tal, o atacante poderia combinar o XSS com técnicas de engenharia social (Zhenget al., 2019) e explorar a realização dephishing (Diorio et al., 2019a), por exemplo. Além disso, o atacante poderia explorar o XSS para espalhar malware(Diorioet al., 2018)e causar negação de serviço(Diorio et al., 2019b), entre outros, com efeitos relacionados podendo ser extremamente críticos para organizações e indivíduos.

5. Conclusão e Trabalhos Futuros

Este trabalho apresentou um estudo prático sobre ataques de XSS. Para tal, com base em um cenário de referência, ataques relacionados foram reproduzidos para fins de experimentação e discussão.

Os experimentos e discussão foram baseados em um laboratório prático sobre segurança de aplicações Web, realizado com estudantes do terceiro ano de um curso de graduação em informática, em uma das aulas de segurança da informação, para auxiliá-los na compreensão de conceitos sobre XSS, explorando o XSS refletido. Essa discussão é importante, por exemplo, para conciliar teoria e prática em atividades de ensino relacionadas ao tema, bem como para que novas pesquisas e contribuições sejam realizadas no âmbito da segurança da informação, tais como para a prevenção e para a detecção dos ataques de XSS.

Trabalhos futuros serão realizados para avaliar a percepção dos estudantes sobre o estudo prático descrito neste trabalho. Além disso, outros estudos práticos serão propostos, explorados e avaliados.

Referências

Alenzi, K. F., &Abbase, O. A. B. (2022). A Defensive Framework for Reflected XSS in Client-SideApplications. *Journalof Web Engineering*, *21*(7), 2209-2229.

Diorio, R. F., Serafim, E., Alves, K. R., & Meira, M. C. (2018). Segurança da informação e de sistemas computacionais: Um estudo prático sobre ataques utilizando malwares. In *9º Congresso Sul Brasileiro de Computação (SULCOMP 2018)*.

Diorio, R. F., Serafim, E., Alves, K. R., & Meira, M. C. (2019a). A PracticalStudyonPhishingAttacks. In *16th CONTECSI - InternationalConferenceonInformation Systems and Technology Management*.

Diorio, R. F., Serafim, E., Alves, K. R., & Meira, M. C. (2019b). A PracticalStudyonFlooding-BasedDistributedDenialof Service Attacks. In *16th CONTECSI - InternationalConferenceonInformation Systems and Technology Management*.

En, V. T., &Selvarajah, V. (2022, December). Cross-Site Scripting (XSS). In *2022 IEEE 2nd InternationalConferenceon Mobile Networks and Wireless Communications (ICMNWC)* (pp. 1-5). IEEE.

Hydara, I.,Sultan, A. B. M., Zulzalil, H., &Admodisastro, N. (2015). Currentstateofresearchoncross-site scripting (XSS) – A systematicliteraturereview. Informationand Software Technology, 58, 170-186.

Liu, M., Zhang, B., Chen, W.,& Zhang, X. (2019). A surveyofexploitationanddetectionmethodsof XSS vulnerabilities. *IEEE access*, *7*, 182004-182016.

OWASP Foundation. (2023a). Cross Site Scripting (XSS). https://owasp.org/www-community/attacks/xss/.

OWASP Foundation. (2023b). OWASP Top Ten. https://owasp.org/www-project-top-ten/.

Xu, G., Xie, X., Huang, S., Zhang, J., Pan, L., Lou, W., &Liang, K. (2020). JSCSP: a novel policy-based XSS defensemechanism for browsers. *IEEE transactionsondependableandsecurecomputing*, *19*(2), 862-878.

Yao, Y., He, J., Li, T., Wang, Y., Lan, X., & Li, Y. (2023). AnAutomatic XSS Attack Vector GenerationMethodBasedontheImprovedDueling DDQN Algorithm. *IEEE TransactionsonDependableandSecureComputing*.

Yusof, I.,&Pathan, A. S. K. (2016). Mitigatingcross-site scriptingattackswith a contentsecuritypolicy. *Computer*, *49*(3), 56-63.

Zheng, K., Wu, T., Wang, X., Wu, B., & Wu, C. (2019). A sessionand dialogue-based social engineering framework. *IEEE Access*, *7*, 67781-67794.