REPÚBLICA FEDERATIVA DO BRASIL
UNIVERSIDADE *HACKER*
INTRODUÇÃO A SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS
MÓDULO VII – DESENVOLVIMENTO *WEB*

Cross-Site Scripting

Prof. Alex Dias Camargo

alexcamargoweb@gmail.com

UNIHACKER.CLUB
PROGRAMA UNIVERSIDADE HACKER

UNIHACKER.CLUB
PROGRAMA UNIVERSIDADE HACKER

I. Plano de aula

Na aula anterior foi visto:

- **Conceito de autenticação quebrada**
- **Exemplo prático: PHP e *Python***

UNIHACKER.CLUB
PROGRAMA UNIVERSIDADE HACKER

I. Plano de aula

Nesta aula será apresentado:

- **Conceito de execução de comandos em sites cruzados**
- **Exemplo prático: PHP e *JavaScript***

UNIHACKER.CLUB
PROGRAMA UNIVERSIDADE HACKER

1. Introdução

XSS consiste em **modificar valores** que a aplicação *Web* usa para enviar variáveis entre duas páginas. Um exemplo é fazer com que através de uma página HTML sejam executados scripts JavaScript.

- **Cenário**: um sistema acadêmico possui vulnerabilidades XSS na exibição de anúncios dos um alunos (mural de recados).
- **Ataque**: Injeção de *script* de alerta JavaScript no formulário.
- **Ambiente**: *Apache* 2.4, PHP 5.6, *MySQL* 5.7.
- **Repositório**: https://github.com/alexcamargoweb/unihacker

GitHub

1. Introdução

aluno	titulo	descricao
Fulano	Oferta!	Vendo mesa de escritório em ótimo estado. Valor R...
Beltrana	Vendo ou troco	Estou negociando a minha geladeira. Valor: R$ 200,...
Ciclano	Aulas de francês online	Dou aulas de inglês online com horário marcado. Va...
Aluno Hacker	Invasão!	<script>alert('Site invadido!');</script>

Figura. Banco de dados: visão geral.

Registro a ser analisado!

1. In

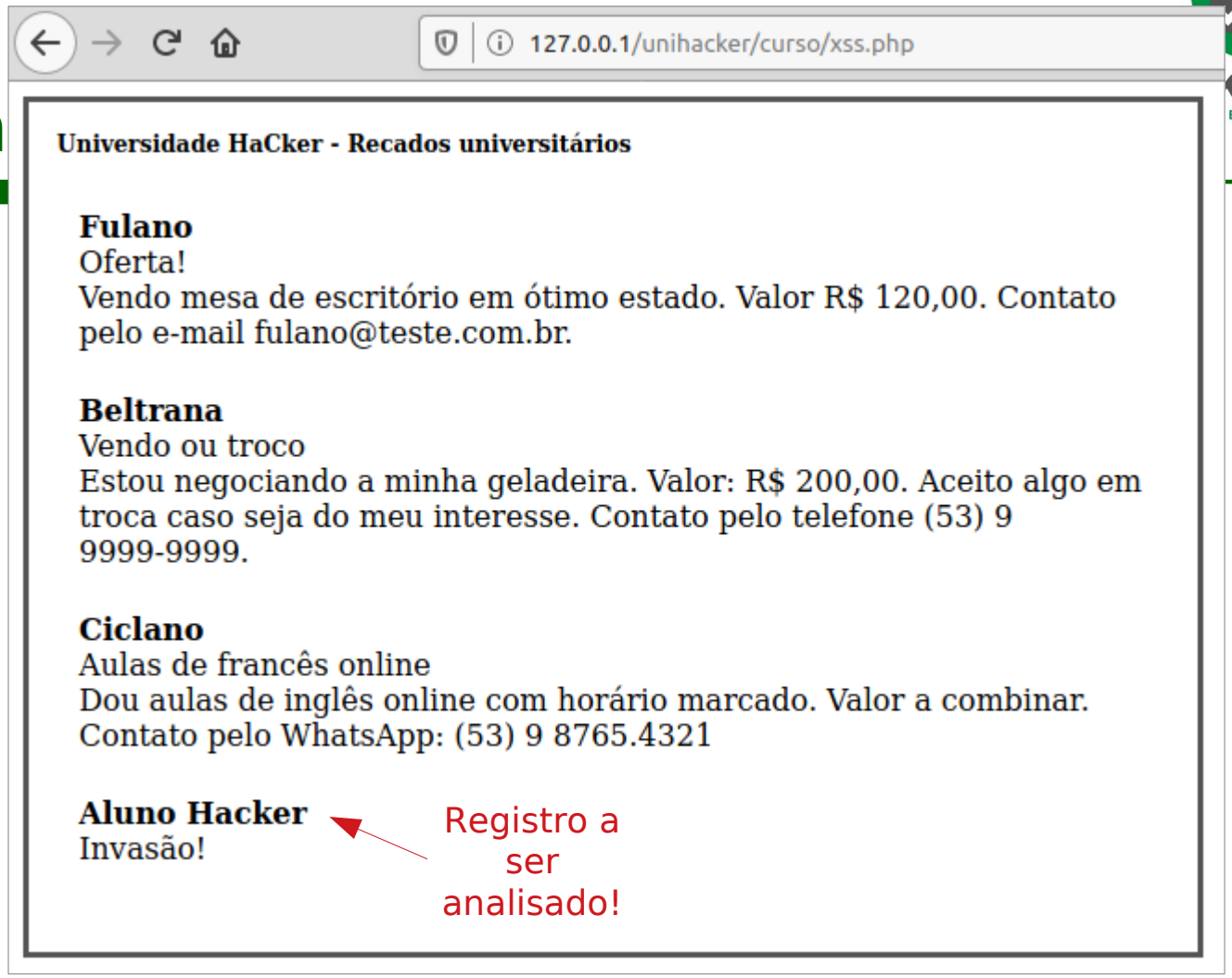

Figura. Aplicação: visão geral.

1 Introdução

```php
<?php

// Abre conexão com o banco de dados
$conn = mysqli_connect('127.0.0.1', 'unihacker', 'UnIh@CkEr', 'unihacker');
// Define o charset
mysqli_set_charset($conn, "utf8");

// Faz a consulta ao banco
$query = "SELECT * FROM recados";

// Executa a query
$result = mysqli_query($conn, $query);

?>

<style>
.dados{
    padding: 10px;
}
.negrito{
    font-weight: bold;
}
</style>
```

Figura. Aplicação: conexão com o banco de dados.

```
28  <html>
29  <head>
30    <title>Universidade Hacker - Cross-site scripting</title>
31  </head>
32  <body style="font-family: Tahoma; font-size:12px;">
33    <div style="border: 3px solid #555; width: 600px; padding:15px;">
34      <span style='font-weight:bold'>Universidade HaCker - Recados universitários</span>
35      <br /><br />
36      <table>
37          <?php
38              while($row = mysqli_fetch_assoc($result)){
39          ?>
40              <tr>
41                  <td class="dados">
42                      <strong><?php print($row['aluno']); ?></strong>
43                      <br />
44                      <?php print($row['titulo']); ?>
45                      <br />
46                      <?php
47                          // Exibir sem escapar a saída (COM VULNERABILIDADE)
48                          print($row['descricao']);
49                      ?>
50                      <br />
51                  </td>
52              </tr>
53          <?php
54              }
55          ?>
56      </table>
57  </body>
58  </html>
```

Figura. Ataque XSS: código-fonte do *script* (com vulnerabilidade).

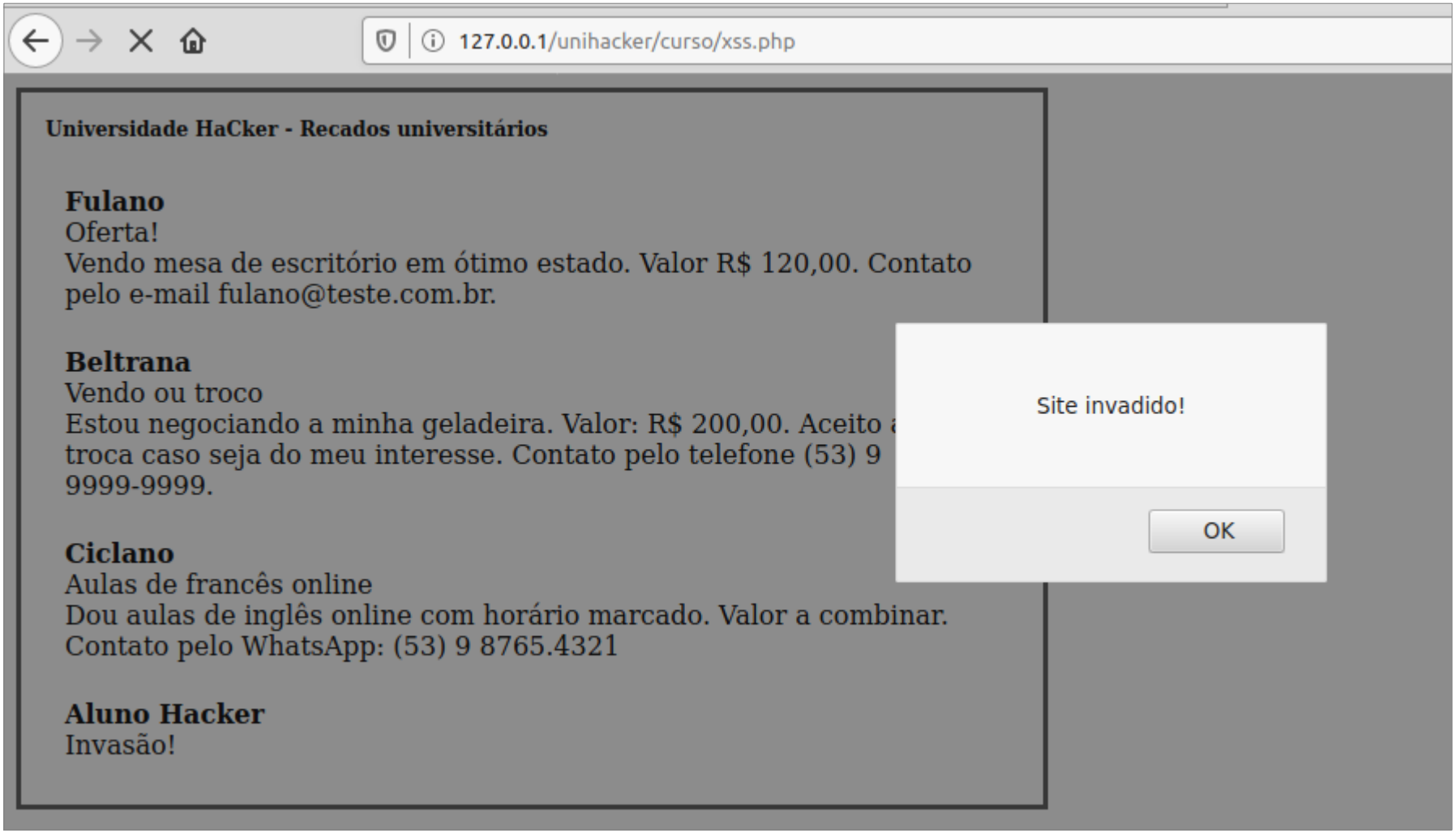

Figura. Ataque XSS: aplicação (com vulnerabilidade).

```
28  <html>
29  <head>
30   <title>Universidade Hacker - Cross-site scripting</title>
31  </head>
32  <body style="font-family: Tahoma; font-size:12px;">
33    <div style="border: 3px solid #555; width: 600px; padding:15px;">
34      <span style='font-weight:bold'>Universidade HaCker - Recados universitários</span>
35      <br /><br />
36      <table>
37          <?php
38              while($row = mysqli_fetch_assoc($result)){
39          ?>
40              <tr>
41                  <td class="dados">
42                      <strong><?php print($row['aluno']); ?></strong>
43                      <br />
44                      <?php print($row['titulo']); ?>
45                      <br />
46                      <?php
47                          // Exibe as tags HTML sem interpretá-las
48                          print(htmlspecialchars($row['descricao']));
49                      ?>
50                      <br />
51                  </td>
52              </tr>
53          <?php
54              }
55          ?>
56      </table>
57  </body>
58  </html>
```

Previne a entrada de caracteres especiais

Figura. Ataque XSS: código-fonte do *script* (sem vulnerabilidade).

1.

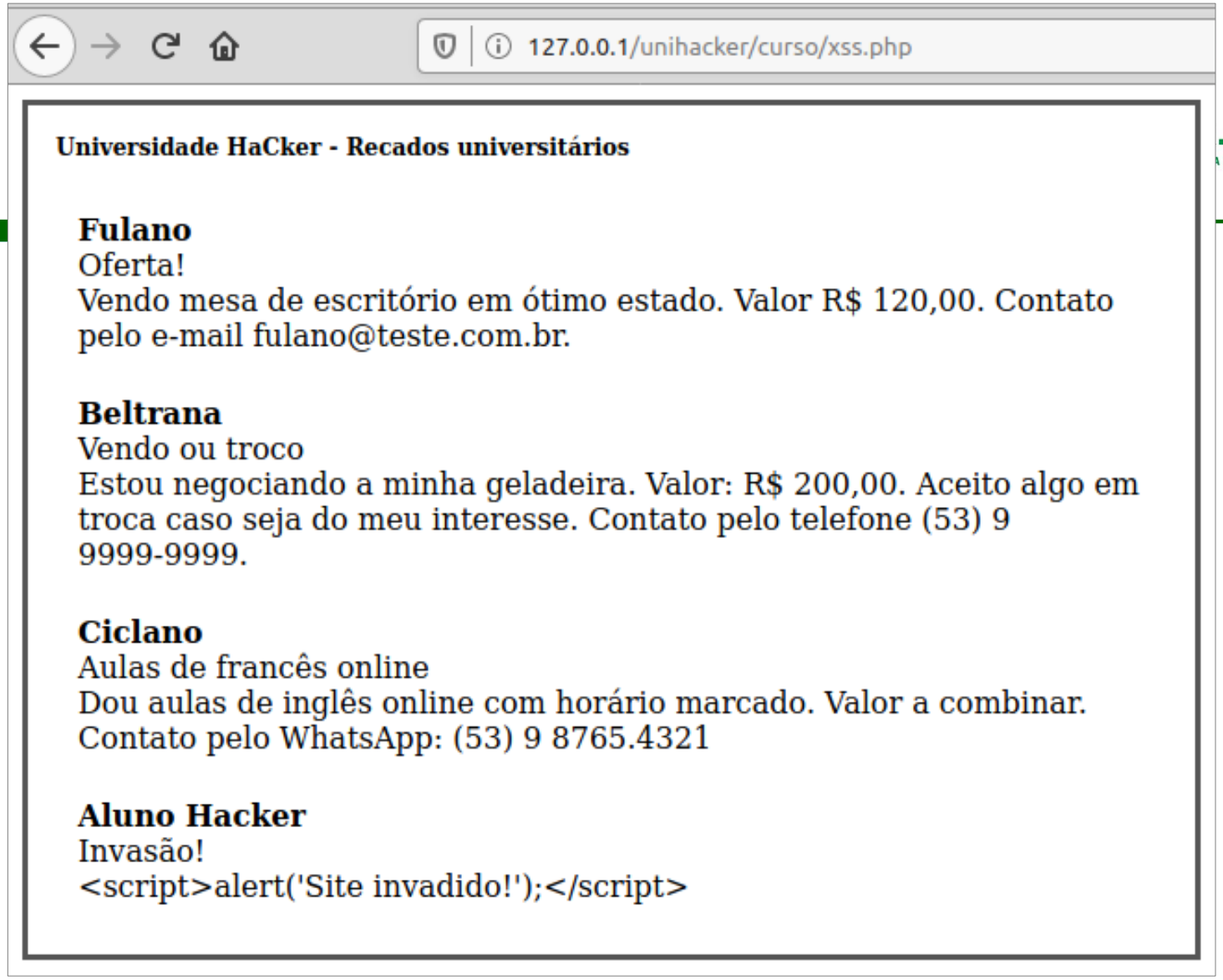

Figura. Ataque XSS: aplicação (sem vulnerabilidade).

```
28  <html>
29  <head>
30   <title>Universidade Hacker - Cross-site scripting</title>
31  </head>
32  <body style="font-family: Tahoma; font-size:12px;">
33    <div style="border: 3px solid #555; width: 600px; padding:15px;">
34      <span style='font-weight:bold'>Universidade HaCker - Recados universitários</span>
35      <br /><br />
36      <table>
37          <?php
38              while($row = mysqli_fetch_assoc($result)){
39          ?>
40              <tr>
41                  <td class="dados">
42                      <strong><?php print($row['aluno']); ?></strong>
43                      <br />
44                      <?php print($row['titulo']); ?>
45                      <br />
46                      <?php
47                          // Exibe sem as tags HTML
48                          print(strip_tags($row['descricao']));
49                      ?>
50                      <br />
51                  </td>
52              </tr>
53          <?php
54              }
55          ?>
56      </table>
57  </body>
58  </html>
```

Remove tags HTML e PHP da string

Figura. Ataque XSS: código-fonte do *script* (sem vulnerabilidade).

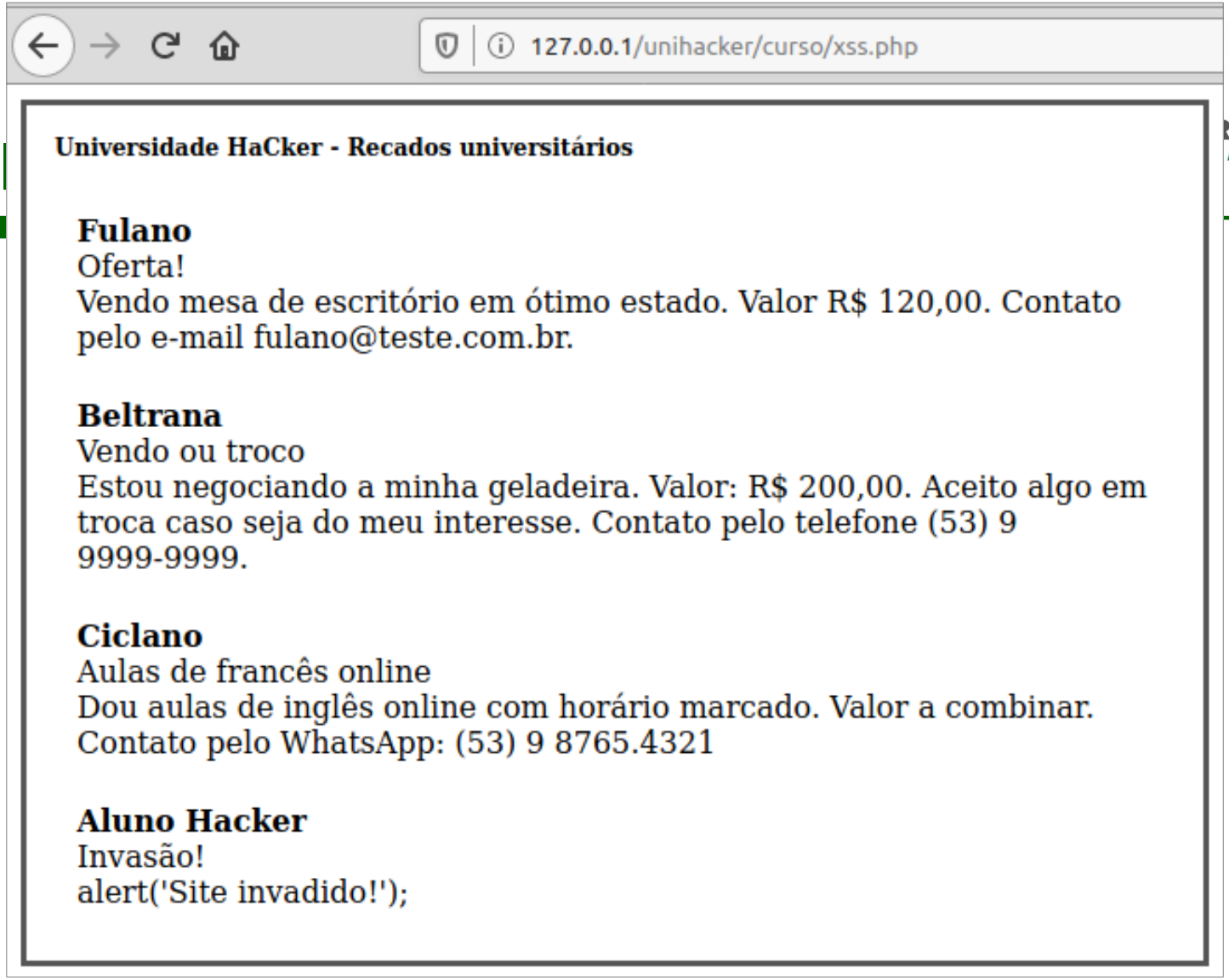

Figura. Ataque XSS: aplicação (sem vulnerabilidade).

2. Exercícios

1. Responda o *quiz* sobre *Cross-Site-Scripting*:

Understanding Cross Site Scripting - (ProProfs Quizzes)
https://moodle.unihacker.club/mod/url/view.php?id=189

Referências básicas

OWASP, Top. Top 10-2017. **The Ten Most Critical Web Application Security Risks. OWASP™ Foundation. The free and open software security community.** URL: https://www.owasp.org/index.php/Top_10-2017_Top_10, 2017.