Riscos de Segurança em Aplicações Web

Hugo Hadden
Universidade de Aveiro

21-02-2025

Contents

1 Introdução

Com a matéria lecionada na aula de Redes e Segurança Informática, desenvolvi um trabalho relacionado com armazenamento e edição de dados em ficheiros XML e JSON. Também foram vistas maneiras de minimizar e prevenir riscos de segurança.

2 Desenvolvimento

2.1 Ficheiro XML

Criação do ficheiro XML para guardar os dados de dois produtos exemplo.

```xml
<?xml version="1.0" encoding="UTF-8"?>

<products>
    <product>
        <name>Produto 1</name>
        <price>19.99</price>
        <img>product1.jpg</img>
        <category>Eletronics;</category>
    </product>
    <product>
        <name>Produto 2</name>
        <price>29.99</price>
        <img>product2.jpg</img>
        <category>Home Applience;</category>
    </product>
</products>
```

Figure 1: Ficheiro XML.

2.2 Leitura do ficheiro XML em Python

Programa em Python para ler o ficheiro XML (na imagem também está para ler ficheiros JSON pois desenvolvi no mesmo ficheiro)

```python
import json
import xml.etree.ElementTree as ET

with open('produtos.json', 'r') as file:
    data_json = json.load(file)

tree = ET.parse('produtos.xml')
root = tree.getroot()

option = 0
while (option != 3):
    print()
    print("1. JSON")
    print("2. XML")
    print("3. Sair")
    option = int(input("Escolha o formato do arquivo que deseja visualizar: "))
    if (option == 1):
        print("Produtos em JSON:")
        for product in data_json['products']:
            print("Nome: " + str(product['name']))
            print("Preço: " + str(product['price']))
            print("Categoria: " + str(product['category']))
            print("Imagem: " + str(product['img']))
            print("================================")
    elif (option == 2):
        print("Produtos em XML:")
        for product in root:
            print("Nome: " + product.find('name').text)
            print("Preço: " + product.find('price').text)
            print("Categoria: " + product.find('category').text)
            print("Imagem: " + product.find('img').text)
            print("================================")
    elif (option == 3):
        print("A sair...")
    else:
        print("Opção inválida")
```

Figure 2: Código Python para ler o ficheiro XML.

2.3 Modificação para uso de entidades internas

O ficheiro XML foi modificado para usar variáveis internas nas categorias dos produtos.

Figure 3: Uso de variáveis internas no XML.

2.4 Testes de segurança

Foram realizados testes para verificar as vulnerabilidades do processamento do ficheiro XML, como por exemplo o ataque "Billion Laughs" que faz com que o servidor ou computador fique a gastar memória RAM.

Figure 4: Ataque Billion Laughs.

2.5 Sugestões de segurança

Para prevenir riscos de segurança em XML, o sugerido é:

- O uso de bibliotecas seguras, como `lxml` ou `defusedxml`;
- Não permitir de entidades externas;
- Limitar o tamanho ou número das entidades XML.

2.6 Ficheiro JSON

Foi criado um ficheiro JSON para guardar os mesmo produtos.

Figure 5: Ficheiro JSON.

2.7 Leitura do ficheiro JSON

Reutilizei o código para ler ficheiros XML e adicionei o método para ler ficheiros JSON.

```python
import json
import xml.etree.ElementTree as ET

with open('produtos.json', 'r') as file:
    data_json = json.load(file)

tree = ET.parse('produtos.xml')
root = tree.getroot()

option = 0
while (option != 3):
    print()
    print("1. JSON")
    print("2. XML")
    print("3. Sair")
    option = int(input("Escolha o formato do arquivo que deseja visualizar: "))
    if (option == 1):
        print("Produtos em JSON:")
        for product in data_json['products']:
            print("Nome: " + str(product['name']))
            print("Preço: " + str(product['price']))
            print("Categoria: " + str(product['category']))
            print("Imagem: " + str(product['img']))
            print("================================")
    elif (option == 2):
        print("Produtos em XML:")
        for product in root:
            print("Nome: " + product.find('name').text)
            print("Preço: " + product.find('price').text)
            print("Categoria: " + product.find('category').text)
            print("Imagem: " + product.find('img').text)
            print("================================")
    elif (option == 3):
        print("A sair...")
    else:
        print("Opção inválida")
```

Figure 6: Código para ler o ficheiro JSON.

2.8 Criação do ficheiro JSON com dados do utilizador

Foi feito um programa que lê dados do utilizador e guarda num ficheiro JSON.

```python
import json

dados = {}

option = 0

while (option != 3):
    print()
    print("1. Escrever dados no json")
    print("3. Sair")
    option = int(input("Escolha a opção desejada: "))
    if (option == 1):
        print()
        new_campos = input("Digite o novo campo: ")
        new_data = input("Digite o novo dado: ")
        dados[new_campos] = new_data
        with open('dados.json', 'w') as file:
            json.dump(dados, file, indent=4)
        print("Dados salvos com sucesso!")
        print()

    elif (option == 3):
        print("A sair...")
    else:
        print("Opção inválida")
```

Figure 7: Código Python para criar o ficheiro JSON.

3 Conclusão

Neste relatório, foram abordados ficheiros JSON e XML para armazenar e editar dados, foram desenvolvidos códigos em Python para ler os ficheiros, foram analisados os riscos de segurança do processamento do ficheiro XML. No final foi feito um código para ler os dados que o utilizador insere e armazenar os mesmos num ficheiro JSON.