Java Web com Servlets e JSPs

Segurança em Java EE

Softblue
cursos online

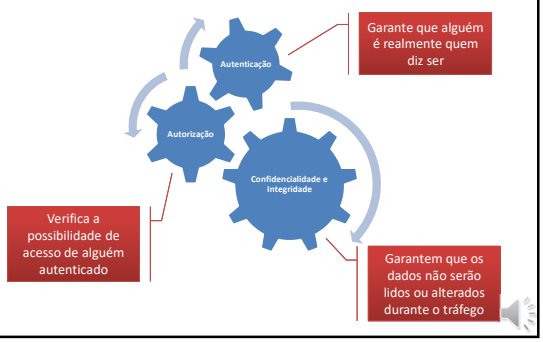
1

Tópicos Abordados

- Conceitos de segurança
  - Autenticação
  - Autorização
- Definindo roles de acesso
- Roles reais e roles lógicos
- Protegendo recursos
- Tipos de autenticação
- Confidencialidade e integridade
- HTTPS
- HTTPS e o Java EE

2

Os Principais Conceitos de Segurança



3

Autenticação

- Garante a identidade de alguém
  - Usuário e senha
  - Leitura biométrica
  - Certificado digital
- Em Java EE, o processo de autenticação não é padronizado
  - Cada container implementa da sua forma

4

Autenticação no Tomcat

- A forma mais fácil de autenticação é feita via arquivos texto

tomcat-users.xml

```
<tomcat-users>
  <role rolename="gerente"/>
  <role rolename="diretor"/>

  <user username="paulo" password="123" roles="gerente"/>
  <user username="pedro" password="123" roles="diretor"/>
</tomcat-users>
```

- Outras formas também são suportadas

5

Autorização

- Depois de autenticado, o usuário pode acessar o recurso protegido?
- Em Java EE, a autorização é padronizada na especificação
  - Todos os containers seguem o mesmo padrão

6

Definindo os Roles de Acesso

Softblue

- Um role representa um grupo de acesso
- Os roles da aplicação são definidos no arquivo *web.xml*

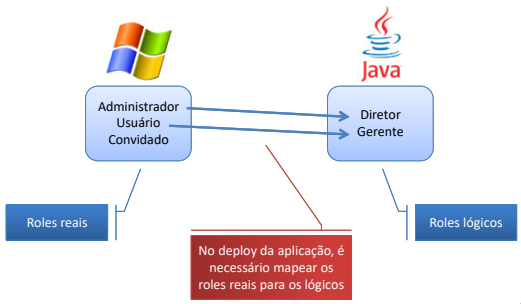
```
<security-role>
  <role-name>gerente</role-name>
</security-role>
<security-role>
  <role-name>diretor</role-name>
</security-role>
```

Estes roles são considerados lógicos

7

Roles Reais x Lógicos

Softblue



8

Protegendo Recursos

Softblue

- Recursos são protegidos em Java EE com base em mapeamentos de URL e HTTP method

```
<web-app>
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>admin pages</web-resource-name>
      <url-pattern>/admin/*</url-pattern>
      <http-method>GET</http-method>
    </web-resource-collection>

    <auth-constraint>
      <role-name>diretor</role-name>
    </auth-constraint>
  </security-constraint>
</web-app>
```

Mapeamento e HTTP method

9

Tipos de Autenticação

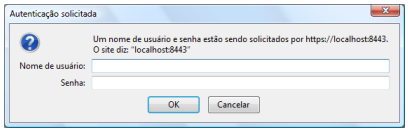
- Ao acessar um recurso protegido, o browser precisa solicitar as credenciais do usuário de alguma forma
- Três formas de autenticação são suportadas por todos os containers Java EE

Forma	Descrição
BASIC	O browser abrirá uma janela padrão solicitando usuário e senha
FORM	O usuário será redirecionado para uma página customizada para digitar o usuário e a senha
CLIENT-CERT	A autenticação é baseada em certificado digital

10

Autenticação BASIC

- O container cuida de todo o processo de autenticação
- Uma janela aberta pelo próprio browser solicita o usuário e a senha



11

Autenticação BASIC

- A configuração é feita no *web.xml*

```
<web-app>
  <login-config>
    <auth-method>BASIC</auth-method>
  </login-config>
</web-app>
```

12

Autenticação FORM

- Páginas customizadas para solicitar informações de login
  - Uma página solicitando usuário e senha
  - Uma página para que haja o redirecionamento caso os dados digitados sejam inválidos

```
<web-app>
  <login-config>
    <auth-method>FORM</auth-method>
    <form-login-config>
      <form-login-page>/login.jsp</form-login-page>
      <form-error-page>/erroLogin.jsp</form-error-page>
    </form-login-config>
  </login-config>
</web-app>
```

13

Autenticação FORM

- O formulário de login deve seguir algumas regras

login.jsp

```
<html>
<body>
<form method="POST" action="j_security_check">
Usuário: <input type="text" name="j_username">
<BR>
Senha:   <input type="password" name="j_password">
<BR>
<input type="submit" value="Login"></TD>
</form>
</body>
</html>
```

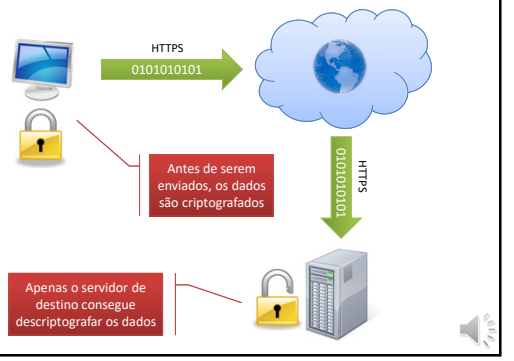
14

Confidencialidade e Integridade

- A confidencialidade previne que os dados sejam lidos durante o tráfego na rede
- A integridade garante que os dados não serão alterados enquanto trafegam pela rede
- O HTTPS dá essas garantias
  - **H**yper**t**ext **T**ransfer **P**rotocol **S**ecure

15

O HTTPS



16

HTTPS e Java EE

- Primeiramente é necessário configurar o servidor para que ele aceite conexões do tipo HTTPS
- Depois, no *web.xml*, basta definir que o acesso a determinado recurso necessita de HTTPS
- O container cuida do resto

17

HTTPS e Java EE

```
<web-app>
  <security-constraint>
    <web-resource-collection>
      <web-resource-name>admin pages</web-resource-name>
      <url-pattern>/admin/*</url-pattern>
    </web-resource-collection>

    <user-data-constraint>
      <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</web-app>
```

Garante o uso de HTTPS sempre que uma URL no padrão */admin/** for acessada

18